

681.3

Т 38



ВСЕСОЮЗНЫЙ  
НАУЧНО-ИССЛЕДОВАТЕЛЬСКИЙ ИНСТИТУТ  
ИНФОРМАЦИИ И ЭКОНОМИКИ МИНПРИБОРА СССР  
(ИНФОРМПРИБОР)

КАТАЛОГ

# ТЕХНИЧЕСКИЕ СРЕДСТВА АСУТП

Средства вычислительной техники

ВЫЧИСЛИТЕЛЬНЫЙ КОМПЛЕКС СМ 1425

# ПСИ

Москва 1989

**В. С. Громов, В. Г. Мельниченко,  
Б. И. Панферов, Л. М. Плахов,  
Н. А. Сирота**

ИНФОРМПРИБОР выпускает третье издание каталога Государственной системы промышленных приборов и средств автоматизации (ГСП) под общей редакцией канд. техн. наук В. А. Рухадзе.

Каталог издается в виде отдельных томов. Каждый том состоит из нескольких выпусков, содержащих описание технических средств ГСП, объединенных по отдельным измеряемым физическим величинам или выполняемым функциям в составе АСУТП.

Структура каталога включает следующие тома:

Том 1. «Общее описание ГСП».

Том 2. «Средства получения информации о параметрах технологических процессов».

Том 3. «Средства локального контроля и автоматизации».

Том 4. «Средства централизованного контроля и регулирования».

Том 5. «Средства вычислительной техники».

Том 6. «Средства воздействия на процесс».

Том 7. «Типовые конструкции и элементы».

Материалы каталога предназначены для проектировщиков автоматизированных систем управления технологическими процессами, разработчиков средств автоматизации контроля и служб эксплуатации этих средств.

В настоящем выпуске дано общее описание вычислительных комплексов СМ 1425 и их назначение, перечислены области применения, рассмотрены принципы построения и приведены краткие описания и характеристики технических и программных средств, входящих в ВК СМ 1425.

По вопросам, касающимся издания каталога, просим обращаться по адресу: 125877, ГСП, Москва, А-252, Чапаевский пер., 14, ИНФОРМПРИБОР.

Ответственный за выпуск И. Н. Морозова

## Средства вычислительной техники

# ВЫЧИСЛИТЕЛЬНЫЙ КОМПЛЕКС СМ 1425

## ГОСУДАРСТВЕННАЯ СИСТЕМА ПРОМЫШЛЕННЫХ ПРИБОРОВ И СРЕДСТВ АВТОМАТИЗАЦИИ

Москва 1989

### НАЗНАЧЕНИЕ И ОБЛАСТЬ ПРИМЕНЕНИЯ

Вычислительный комплекс СМ 1425 (рис. 1) является новой моделью семейства 16-разрядных малых ЭВМ. Программно совместимый с СМ 1420,

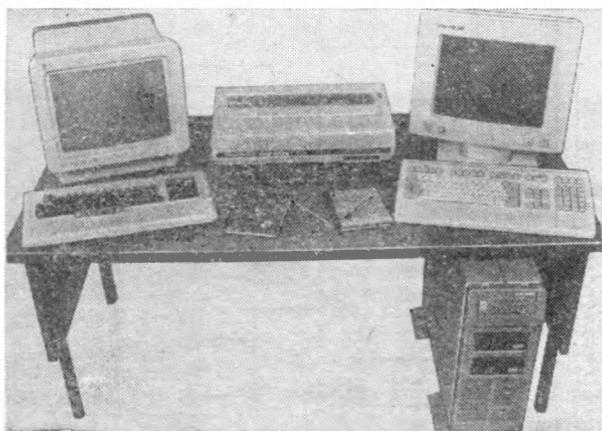


Рис. 1. Вычислительный комплекс СМ 1425

комплекс СМ 1425 обладает большей производительностью и надежностью, имеет существенно меньшие габариты, массу и потребляемую мощность. СМ 1425 оснащен современной периферией с высокими техническими характеристиками. Широкое применение больших интегральных схем по-

зволило резко уменьшить размеры электронных устройств. Компактность и низкая стоимость комплекса приближают его к классу микроЭВМ.

Комплекс предназначен для использования в системах научных исследований, автоматизированных системах управления производством, информационно-справочных и других системах. Вместо «Общей шины» (ОШ) — системного интерфейса предыдущих моделей — в комплексе СМ 1425 применен 22-разрядный магистральный параллельный интерфейс (МПИ). Введение блочной передачи данных между устройствами прямого доступа и оперативной памятью позволяет более эффективно использовать этот интерфейс. Для подключения к СМ 1425 устройств с ОШ имеется специальный модуль согласования системных интерфейсов.

Система команд СМ 1425 включает в себя помимо команд СМ 1420 команды для организации дополнительного режима работы «Супервизор». Наличие встроенных тестов, аппаратного загрузчика и микропрограммного эмулятора пульта управления обеспечивает удобство обслуживания комплекса. Система команд, время их выполнения и представление перерабатываемой информации определяются параметрами микропроцессора. Его система команд содержит набор команд СМ 1420, в том числе команды с плавающей запятой (за исключением команды «Диагностика»), а также дополнительные команды SPL, CSM, TSTSET, WRTLCK, MFPT.

Комплекс обеспечивает мультипрограммную работу в режиме реального времени по временным меткам таймера при частоте импульсов  $(50 \pm 1)$  Гц, а также в режиме разделения времени и имеет следующие дополнительные возможности: три режима работы процессора («Ядро», «Супервизор», «Пользователь»); два набора регистров диспетчера памяти для каждого из режимов работы, что позволяет в два раза увеличить объем адресуемой виртуальной памяти; два набора регистров общего назначения, которые могут использоваться программными средствами для уменьшения задержки переключения с одной программы на другую; блочную передачу данных устройствами прямого доступа к оперативной памяти; аппаратный механизм программных запросов на прерывание.

Наличие в СМ 1425 большого объема хранимых в ПЗУ процессора встроенных тестов и введение средств резервирования улучшают эксплуатационные характеристики комплекса. Система контроля и диагностики дает возможность осуществлять контроль хранения информации с исправлением одиночных и обнаружением двойных ошибок в оперативной памяти, первичную проверку работоспособности основного оборудования комплекса встроенными тестами, проверку работоспособности и диагностику основных устройств комплекса при загрузке операционной системы.

#### ТЕХНИЧЕСКИЕ ДАННЫЕ

Представление информации:  
 беззнаковой . . . . . в поразрядном  
 двоичном коде

с фиксированной запятой . . . . .	в дополнительном коде со знаком
с плавающей запятой . . . . .	в прямом коде
Формат данных при выполнении команд, двоичные разряды:	
беззнаковой информации . . . . .	8; 16
с фиксированной запятой . . . . .	8; 16; 32
с плавающей запятой . . . . .	32; 64
Адресность команд . . . . .	безадресные, одно- и двухадресные
Быстродействие комплекса, млн. коротких команд/с . . . . .	не менее 3
Емкость ПЗУ для программ-загрузчиков и встроенных тестов, Кслов . . . . .	8
Емкость оперативной памяти, Мбайт:	
минимальная . . . . .	0,5
максимальная . . . . .	4
Питание:	
от сети переменного тока напряжением, В . . . . .	220 (50 Гц)
от источника стабилизированного напряжения комплекса В260 . . . . .	5 В, 40 А; 12 В, 10 А; —12 В, 1 А
Потребляемая мощность, кВ·А . . . . .	1,2
Габаритные размеры базового монтажного блока (тумбы), мм . . . . .	560×200×720
Масса комплекса, кг . . . . .	не более 100
Условия эксплуатации:	
температура окружающего воздуха, °С . . . . .	10—35
относительная влажность при 30°С, % . . . . .	40—80
атмосферное давление, кПа . . . . .	84—107
вибрация:	
частота, Гц . . . . .	25
амплитуда, мм . . . . .	не более 0,1

## СТРУКТУРА И СОСТАВ

СМ 1425 имеет агрегатный принцип построения (рис. 2), позволяющий создавать разнообразные по набору и количеству технических средств варианты комплекса. Все функциональные блоки выполнены в виде конструктивно законченных устройств (модулей), связь между которыми осуществляется по МПИ.

### Процессор

Центральным звеном комплекса является процессор. Он управляет всей работой комплекса, обеспечивает обмен информацией между внешними устройствами и оперативной памятью (ОП), выполняет арифметическую и логическую обработку информации. К основным элементам процессора, определяющим его архитектуру, относятся регистры общего назначения, регистры управления системой, организация памяти и система команд.

### Регистры общего назначения

Всего в процессоре имеется 16 шестнадцатирядных регистров общего назначения, доступных программисту, и только восемь из них доступны пользователю в любой момент времени. Они могут использоваться как накопители, индексные регистры, указатели адресов или указатели области памяти для временного хранения данных (стековые указатели). Конкретное использование регистров зависит от выбранного режима адресации.

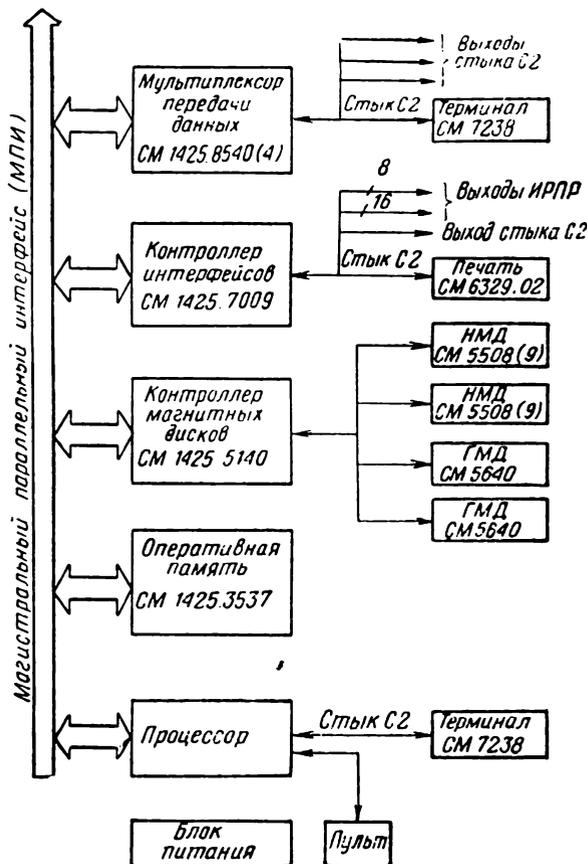


Рис. 2. Структурная схема типового комплекса СМ 1425

Существует две группы регистров, обозначенных R0...R5 и R0'...R5'. Группа, выбранная для применения на текущем этапе выполнения программы, указывается разрядом 11 в слове состояния процессора. Некоторые команды используют регистр R6 в качестве указателя стека, не указывая данный регистр в адресном поле команды. Поэтому R6 называется указателем аппаратного стека и обозначается как SP. Аппаратный стек применяется также в процедурах обработки прерываний. Он используется только для хранения слов, поэтому содержимое указателя стека изменяется всегда на 2. Из трех имеющихся указателей стека: KSP, SSP, USP, в каждый конкретный момент времени только один доступен пользователю. Выбор стекового указателя осуществляется согласно кодам в разрядах режима слова состояния процессора:

00 (режим «Ядро») — указатель стека KSP;

01 (режим «Супервизор») — указатель стека SSP;

11 (режим «Пользователь») — указатель стека USP.

В большинстве случаев выбор регистра SP зависит от кода в разрядах слова состояния процессора PSW [15, 14] (текущий режим). В командах MFPI, MFPD, MTP1 и MTPD при использовании регистра R6 в качестве приемника выбор SP зависит от кода в разрядах PSW [13, 12] (предыдущий режим). Коды выбора регистра указателя стека одни и те же для текущего и предыдущего режимов.

Регистр R7 используется в качестве программно-счетчика команд (PC). Он содержит 16 разрядов виртуального адреса следующего слова потока команд, которое должно быть выбрано из оперативной памяти. Обращение к ОП производится автоматически при выполнении команд программы. В то же время R7 является регистром общего назначения. Однако использование его для других целей может привести к непредсказуемым результатам.

## Регистры управления системой

К регистрам управления системой относятся: слово состояния процессора PSW, регистр ошибок процессора CPUERR, регистр запроса программно-прерывания PIRQ, регистр таймера LTC, регистр обслуживания MR.

Регистр PSW (адрес 17777776) содержит информацию о текущем состоянии процессора и определяет режим его работы. Формат регистра PSW показан на рис. 3. Содержимое разрядов следующее.



Рис. 3. Формат регистра PSW

Разряды 15 и 14 — информация о текущем режиме работы процессора, разряды 13, 12 — информация о предыдущем режиме его работы. Рабочий режим кодируется в обоих случаях аналогичным образом и определяется согласно табл. 1.

Значение разрядов регистра PSW		Режим работы процессора
15, 14	13, 12	
0 0	0 0	Режим «Ядро»; все команды разрешены
0 1	0 1	Режим «Супервизор»; запрещены команды HALT, SPL, RESET
1 0	1 0	Запрещенный режим
1 1	1 1	Режим «Пользователь»; запрещены команды HALT, SPL, RESET

Разряд 11 выбирает группу регистров общего назначения. При PSW [11]=0 выбирается группа R0...R5, при PSW [11]=1 — группа R0'...R5'.

Разряды 10 и 9 в CM 1425 не используются, читаются нулями.

Разряд 8 доступен для записи и чтения; на работу комплекса влияния не оказывает.

Разряды 7...5 определяют текущий уровень приоритета процессора, который может быть присвоен ему согласно табл. 2.

Таблица 2

Значение разрядов регистра PSW			Уровень приоритета
7	6	5	
1	1	1	7
1	1	0	6
1	0	1	5
1	0	0	4
0	1	1	3
0	1	0	2
0	0	1	1
0	0	0	0

Разряд 4 (T) используется для отладки программ. Если перед выполнением какой-либо команды этот разряд установлен в единицу, то после выполнения команды происходит внутреннее прерывание по вектору с адресом 14. Разряд 4 не может быть установлен непосредственной записью в регистр PSW. Он изменяется только при прерываниях и с помощью команд RTI и RTT.

Разряды 3...0 содержат признаки, которые характеризуют результат выполнения последней команды. Способ установки признаков зависит от выполняемой команды и приводится в ее описании. При выполнении большинства команд устанавливаются стандартные признаки:

N=1, результат отрицательный;

Z=1, результат равен нулю;

V=1, при получении результата произошло арифметическое переполнение;

C=1, при получении результата произошел перенос из старшего разряда.

Регистр PSW сбрасывается в нулевое состояние при включении питания, пуске с консольного терминала по команде G и нажатии на клавишу «Пуск» панели управления.

Регистр CPUERR (адрес 1777766) содержит информацию об источнике возникновения ошибочных ситуаций, вызывающих прерывание по вектору с адресом 4. Формат регистра приведен на рис. 4. Содержимое разрядов следующее.

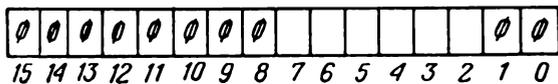


Рис. 4. Формат регистра CPUERR

Разряды 15... 8, 1, 0 в СМ 1425 не используются и всегда читаются нулями.

Разряд 7 устанавливается в единицу в режиме «Ядро», если выполнение команды HALT запрещено ( $MR[3]=1$ ), а также при попытке выполнить команду HALT в режимах «Пользователь» или «Супервизор».

Разряд 6 устанавливается в единицу при обращении к памяти по нечетному адресу в случае чтения команды, чтения и записи двухбайтной информации, а также попытки выбрать команду из регистра общего назначения.

Разряд 5 устанавливается в единицу при обращении к оперативной памяти по несуществующему адресу (тайм-аут).

Разряд 4 устанавливается в единицу по таймауту при обращении к странице ввода-вывода.

Разряд 3 устанавливается в единицу при нарушении «желтой зоны» стека (обращение к стеку в режиме «Ядро» по адресу в пределах от 340 до 377).

Разряд 2 устанавливается в единицу при нарушении «красной зоны» стека (адреса от 0 до 377) — прекращение записи в стек в режиме «Ядро» при системном прерывании, аварийном завершении.

Регистр CPUERR только читается. Он сбрасывается в нулевое состояние при включении питания, пуске с консольного терминала по команде G, нажатии клавиши «Пуск» панели управления, любой записи в регистр.

Регистр PIRQ (адрес 1777772) является средством прерывания программы по вектору с адресом 240. Формат регистра приведен на рис. 5. Содержимое разрядов следующее.

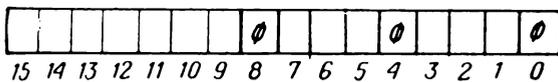


Рис. 5. Формат регистра PIRQ

Каждый из разрядов регистра с 15 по 9 устанавливает запрос на прерывание с уровнем соответственно 7... 1. В регистре может быть установлено более одного запроса.

В разрядах 7... 5 и 3... 1 аппаратно формируется закодированное значение самого приоритетного из установленных запросов.

Разряды 8, 4, 0 не используются и читаются нулями.

При выполнении прерывания обслуживающая программа обрабатывает прерывание и сбрасывает в регистре PIRQ разряд, соответствующий запросу на данное прерывание. К разрядам 15... 9 возможен программный доступ с операциями записи и

чтения, а к разрядам 7... 5 и 3... 1 — только с операциями чтения. Регистр PIRQ сбрасывается в нуль при пуске программы и по команде RESET.

Регистр LTC (адрес 17777546) управляет реакцией на сигнал интерфейса BEVNTL. В регистре используется только разряд 6, к которому возможен программный доступ с операциями записи и чтения. Остальные разряды читаются нулями.

Если  $LTC[6]=1$ , сигнал BEVNTL вызывает формирование запроса на прерывание с уровнем приоритета 6. Адрес вектора прерывания 100. Если  $LTC[6]=0$ , сигнал BEVNTL не воспринимается. Разряд 6 сбрасывается в нуль при включении питания, пуске с пульта (в режиме эмулятора пульта) и по команде RESET.

Регистр MR (адрес 17777750) обеспечивает задание варианта завершения процедуры включения питания и варианта реакции на команду HALT. Кроме того, регистр отражает состояние источника питания и наличие ускорителя операций с плавающей запятой. Формат регистра показан на рис. 6. Содержимое разрядов следующее.

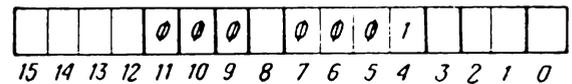


Рис. 6. Формат регистра MR

Разряды 15... 12 определяют старшие разряды адреса начальной загрузки и устанавливаются пользователем на блоке элементов процессора.

Разряды 11... 9 в СМ 1425 не используются.

Разряд 8 устанавливается в единицу, если на плате процессора установлена интегральная микросхема (ИМС) ускорителя команд с плавающей запятой.

Код 0001 в разрядах 7... 4 определяет тип процессора СМ 1425. Устанавливается распайкой на блоке элементов (БЭ) процессора. Используется системным матобеспечением.

Разряд 3 определяет вариант выполнения команды HALT в режиме «Ядро». Если  $MR[3]=0$ , процессор по команде HALT входит в режим эмулятора пульта. Если  $MR[3]=1$ , по команде HALT значение указателя стека устанавливается равным 4 и происходит прерывание по вектору с адресом 4. Значение разряда устанавливается переключением на блоке элементов процессора.

Разряды 2, 1 определяют вариант включения питания. Значения этих разрядов опрашиваются в конце процедуры включения питания и по коду в них определяется вариант завершения этой процедуры. Варианты включения питания приведены в табл. 3.

Таблица 3

Значение разрядов регистра MR		Действие
2	1	
0	0	Значение PC ← (24), PSW ← (26)
0	1	Переход в режим эмулятора пульта, PSW: = 0
1	0	Значение PC = 173000, PSW: = 340
1	1	Переход по адресу загрузчика (см. MR[15...12]), PSW: = 340

Разряд 0 определяет состояние источника питания; MR[0]=1 говорит о нормальном рабочем состоянии питания.

## Организация памяти

Минимально адресуемой единицей памяти является байт (восемь двоичных разрядов). Память представляет собой линейную последовательность байтов, начиная с нулевого адреса. Кроме того, память можно представить в виде последовательности двухбайтных слов. Младшему байту в слове соответствует четный адрес, а старшему — нечетный. Обращения к словам выполняются по четырем адресам. Адреса векторов прерываний, ячеек ОП, регистров периферийных устройств и регистров процессора, если это не оговорено особо, указываются в восьмеричной системе исчисления.

Процессор работает с 16-разрядным виртуальным адресом, позволяющим обращаться к 64 Кбайт (32 Кслов) памяти. Виртуальный адрес преобразуется диспетчером памяти процессора в 18-разрядный или 22-разрядный физический адрес. При этом реализуется возможность динамического перемещения адресов и защиты памяти от несанкционированного доступа.

## Стек

Стеком называется такой способ организации массива элементов памяти, при котором их запись и выборка производятся по принципу: последний записанный элемент выбирается из массива первым. Адрес, по которому выполняется выборка или запись элемента, называется вершиной стека.

Архитектура комплекса позволяет организовывать стеки со скользящей вершиной. В качестве указателя вершины стека можно использовать любой из регистров общего назначения, кроме счетчика команд РС. Режимы адресации с автоувеличением и автоуменьшением обеспечивают автоматическую регулировку положения вершины стека.

При организации стека программист должен установить в указателе стека первоначальное значение адреса его вершины. При записи в стек процессор вначале уменьшает содержимое указателя стека на 2 (при стеке, состоящем из слов) или на 1 (при стеке, состоящем из байтов), а затем записывает новый элемент по адресу, содержащемуся в указателе стека. При выборке из стека процессор выбирает элемент по адресу, содержащемуся в указателе стека, и затем увеличивает содержимое указателя на 2 или 1.

В режиме «Ядро» предусмотрена аппаратная защита зоны векторов прерываний. Если при обращении к памяти по указателю стека КSP формируется виртуальный адрес меньше 400, фиксируется нарушение границы стека. Существует «желтая зона» из 16 слов, расположенных ниже границы стека. При этом выполнение команды завершается и выполняется прерывание по вектору с адресом 4. Нарушение «красной зоны» — фатальная ошибка при обращении к стеку. Выполнение команды, вызвавшей нарушение, прекращается и выполняется прерывание по вектору с адресом 4. «Красная» и «желтая» зоны занимают ячейки с адресами 0...377 и 340...377 соответственно.

## Система команд и представление данных

Команды комплекса СМ 1425 отличаются друг от друга количеством обрабатываемых операндов. По этому признаку команды делятся на безадресные, одно- и двухадресные. Формат безадресных команд состоит только из поля кода операции. В формате одно- и двухадресных команд кроме поля кода операции имеются, соответственно, одно или два поля адреса. 16-разрядный формат команды не позволяет задавать в поле адреса непосредственно полный адрес ячейки памяти, поэтому используется косвенная адресация через регистры общего назначения. В поле адреса указываются номер выбранного регистра и режим адресации (способ использования регистра). Всего имеется восемь режимов адресации. Более подробно система команд описана в приложениях 1 и 2.

Числа для обработки могут быть представлены в двух видах: со знаком и без него. Беззнаковое представление чисел имеет диапазон 0... (2<sup>16</sup>—1) при операциях со словами и 0... (2<sup>8</sup>—1) при операциях с байтами. Если во время выполнения команды операнд выходит за этот диапазон, то в регистре PSW устанавливается разряд переноса С. При знаковом представлении чисел в операциях со словами знаком является разряд 15, в операциях с байтами — разряд 7. Числа со знаком представлены в дополнительном коде. Диапазон представления чисел в операциях со словами —2<sup>15</sup>... (2<sup>15</sup>—1), в операциях с байтами —2<sup>7</sup>... (2<sup>7</sup>—1). Если при выполнении команд операнд выходит за диапазон знакового представления чисел, устанавливается разряд переполнения V в регистре PSW. В табл. 4 даны коды некоторых чисел, в том числе максимальных положительного и отрицательного.

Таблица 4

Десятичное число	Восьмеричное число	
	Знаковый разряд 15	Разряды числа 14...0
+32767	0	77777
+32766	0	77776
+1	0	00001
0	0	00000
—1	1	77777
—2	1	77776
—32767	1	00001
—32768	1	00000

## Быстродействующая буферная память

Статистика выполнения программ наглядно показывает, что большую часть времени программа выполняется в пределах сравнительно небольшого набора адресов. Организация быстродействующей буферной памяти (ББП) процессора основана на этой особенности поведения программ. ББП с небольшим объемом и высоким быстродействием используется для хранения содержимого ячеек ОП, обращение к которым непосредственно предшест-

вует текущему состоянию программы. Программа, хранящаяся в ББП, выполняется значительно быстрее, чем программа, находящаяся в ОП, из-за большой разницы между временем доступа к ББП и ОП, подключенной к МПИ. Ниже поясняется структура ББП процессора. ББП — это память с прямым отображением емкостью 8 Кбайт.

### Конструкция физического адреса и слова ББП

Физический адрес ББП может быть представлен в виде трех составляющих: девяти разрядов поля метки LABEL, двенадцати разрядов индекса INDEX и одного разряда адресации байта в слове. Формат адреса ББП приведен на рис. 7. Поле индекса содержит 12 младших разрядов адреса и служит для адресации слов ББП объемом 4096 слов, поле метки — 9 старших разрядов физического адреса.

Каждое слово ББП содержит: девятиразрядное поле признаков TAG, разряд контроля поля при-

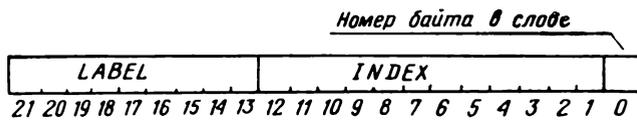


Рис. 7. Формат адреса ББП

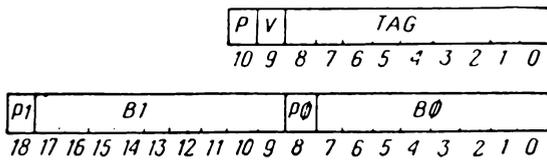


Рис. 8. Формат слова ББП

разряд достоверности V равен единице и все контролируемые порции информации слова ББП прочитаны без ошибок. Если условия попадания выполняются при операции чтения, то содержимое байтов B0 и B1 служат источником информации; если же условия попадания не выполняются (промах), то информация читается из ОП. При этом заполняется соответствующее слово ББП.

Содержимое ячейки ОП запоминается в ББП следующим образом: в поле признаков TAG слова ББП, адресуемого полем индекса физического адреса ячейки, записывается содержимое поля метки LABEL физического адреса, в разряд достоверности V записывается 1, в поля байтов B0 и B1 — информация, прочитанная из ОП; для всех контролируемых порций информации формируются значения контрольных разрядов, которые записываются соответственно в разряды P, P0 и P1. Последующее обращение процессора по тому же адресу вызовет попадание, и информация поступит из ББП. Надо помнить, что заполнение ББП обычно разрушает предыдущее достоверное значение слова ББП. ББП заполняется при промахах в операциях чтения и записи слова.

Операция записи может быть операцией записи слова или операцией записи байта. Основная память обновляется при любой из них. При операции записи байта и попадании в ББП запись соответствующего байта будет производиться и в ОП и в ББП. Такой тип записи называется сквозной записью. При операции записи слова и промахе будет выполняться заполнение слова ББП и запись двух байтов в ОП. При операции записи байта и промахе запись байта будет производиться только в ОП. Все варианты работы с ББП приведены в табл. 5. Содержимое старших 8 Кбайт адресного пространства никогда не запоминается в ББП и,

Таблица 5

Операции	Устройство прямого доступа		Процессор	
	Попадание	Промах	Попадание	Промах
Чтение	Чтение из ОП; ББП не изменяется	Чтение из ОП; ББП не изменяется	Чтение из ББП	Чтение из ОП; заполнение слова ББП
Запись слова	Запись в ОП; обнуление V	Запись в ОП; ББП не изменяется	Запись в ОП; обновление в ББП	Запись в ОП; заполнение слова ББП
Запись байта	То же	То же	То же	Запись в ОП; ББП не изменяется
Чтение; обход (CCR[9] = 1)	—	—	Чтение из ОП; обнуление V	Чтение из ОП; ББП не изменяется
Запись; обход (CCR[9] = 1)	—	—	Запись в ОП; обнуление V	Запись в ОП; ББП не изменяется
Чтение; принудительный промах (CCR[2] ∨ CCR[3] = 1)	—	—	Чтение из ОП; ББП не изменяется	Чтение из ОП; ББП не изменяется
Запись; принудительный промах (CCR[2] ∨ CCR[3] = 1)	—	—	Запись в ОП; ББП не изменяется	Запись в ОП; ББП не изменяется

знаков P, разрядность достоверности поля признаков V, два байта данных ББП B0 и B1, два контрольных разряда байтов P0 и P1. Формат слова ББП представлен на рис. 8. Содержимое ячейки ОП находится в ББП (попадание), если содержимое поля признаков слова TAG, адресуемое полем индекса физического адреса ячейки, равно содержимому поля метки LABEL физического адреса,

следовательно, всегда вызывает промах. Сделано это потому, что в указанную страницу адресного пространства входят регистры состояния устройств, динамически изменяющиеся и всегда содержащие самую свежую информацию.

При включении питания в каждое слово ББП записываются правильные контрольные разряды. Данное действие будем называть очисткой ББП.

В случаях, когда устройство прямого доступа (УПД) выполняет запись в ячейки ОП, копии которых находятся в ББП, возникает проблема несоответствия информации в ББП и ОП (проблема потенциально устаревшей информации). При выполнении УПД операции записи в ОП в соответствующих словах ББП необходимо обнулять признаки достоверности. Данную процедуру реализует оборудование ББП, отслеживающее все операции УПД. Эта процедура включает в себя и блочные операции УПД.

Для диагностирования и обеспечения высокой готовности комплекса используется программное отключение ББП. Она блокируется установкой в единицу любого из разрядов CCR[3] или CCR[2] регистра управления ББП. Если ББП заблокирована, при всех обращениях формируется промах. Это означает, что все обращения будут обрабатываться в ОП, реакция на ошибки по паритету в ББП заблокирована и заполнение ББП не выполняется. Такое состояние соответствует исключению ББП из комплекса. В режиме обхода (CCR[9]=1) все обращения адресуются к ОП, производится контроль по паритету и в заполненных словах ББП, по адресам которых выполняется обращение, гасится разряд достоверности V. Операции чтения в режиме обхода выполняют контроль ошибок по паритету в ББП и отменяют все попадания.

## Реакция на ошибки

Ошибки ОП, воспринимаемые процессором, указывают на некорректируемое искажение информации. Они всегда вызывают фатальный исход, так как информация в ОП не может быть восстановлена. Реакция процессора на некорректируемую ошибку при выполнении чтения из ОП состоит в прекращении операции и прерывании программы по вектору с адресом 114 в пространстве данных режима «Ядро».

В ББП используется контроль по четности для четного (нулевого) байта информации и поля признаков и контроль по нечетности для нечетного байта информации. Ошибки в ББП никогда не рассматриваются как фатальные, ибо искажения информации может быть восстановлена чтением правильной информации из ОП. Процессор реагирует на ошибки ББП или прекращением операции с последующим прерыванием, или прерыванием в конце текущей команды. Адрес вектора прерывания тот же, что и для ошибок ОП. Реакция на ошибки в ББП может быть заблокирована. Конкретный вариант реакции определяется состоянием разрядов CCR[7] и CCR[0] регистра управления ББП и поясняется табл. 6.

Таблица 6

Значения разрядов CCR		Выполняемые действия
7	0	
0	0	Обновление ББП; прерывание по вектору с адресом 114
0	1	Только обновление ББП
1	X	Обновление ББП; прекращение операции и прерывание по вектору с адресом 114

Если в ББП возникает ошибка по паритету в то время, как содержимое регистра ошибок системы памяти MSER еще не очищено от информации, зафиксированной по предыдущей ошибке, то новая ошибка не вызовет никакой реакции. Обращение будет обрабатываться в ОП и на место искаженных данных в ББП будет записана верная информация. Такая реакция на многократные ошибки предотвращает генерацию неопределенной серии прерываний или циклов с прекращением операций и прерываниями в ответ на неисправности оборудования в ББП.

## Регистры управления и состояния

Регистры управления и состояния включают в себя: регистр управления ББП CCR, регистр ошибок системы памяти MSER и регистр попаданий/промахов HMR. Эти регистры используются для управления системой памяти и фиксации информации об обнаруженных ошибках.

**Регистр CCR** (адрес 1777746) управляет операциями ББП. С его помощью могут быть программно установлены режим обхода ББП, режим прекращения операции при обнаружении ошибок и режим принудительного промаха. Формат регистра CCR показан на рис. 9. Содержимое разрядов следующее.

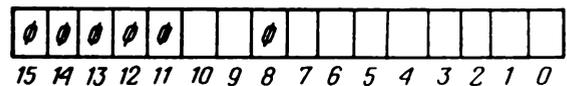


Рис. 9. Формат регистра CCR

Разряды 15...11 в СМ 1425 не используются.

Разряд 10 в единичном состоянии вызывает запись неверного контрольного разряда поля признаков TAG при всех циклах записи, что приведет к ошибке паритета TAG при последующем обращении по этому адресу.

Если разряд 9 — безусловный обход — установлен в единичное состояние, все обращения к памяти процессор отсылает к ОП.

Разряд 8 в единичном состоянии вызывает очистку всего содержимого ББП (содержимое становится недостоверным). Запись нуля в разряд не оказывает влияния на работу ББП. Очистка ББП длится приблизительно 1 мс. В течение этого времени не обслуживаются УПД и запросы на прерывание и не выполняются вычисления.

Разряд 7 — прекращение операции при ошибках ББП — совместно с разрядом 0 определяют вариант реакции на ошибки по паритету. Разряд используется только с целью диагностирования ББП.

Разряд 6 в единичном состоянии вызывает запись неверных контрольных разрядов данных при всех циклах записи, что приведет к ошибкам по паритету данных при последующем обращении по этому адресу.

Разряды 5, 4 могут быть установлены в единицу или сброшены в нуль программно. В СМ 1425 они не используются.

Разряды 3, 2 — принудительный промах. При установке в единицу любого из разрядов 3 или 2 все обращения процессора к памяти отсылаются к ОП, разряды признаков и данные в ББП не изменяются, контроль по паритету не выполняется.

Разряд 1 — режим диагностирования. При еди-

ничном значении разряда все операции записи слова, если не установлены режимы обхода и принудительного промаха, будут заполнять ячейки ББП независимо от того, что обращение адресуется к несуществующей памяти. При этом прерывание происходить не будет.

Разряд 0 совместно с разрядом 7 определяют вариант реакции на ошибки ББП. При коде 00 в этих разрядах ошибка ББП вызовет прерывание через вектор с адресом 114, при коде 01 ошибки ББП не вызывают никакой реакции. Во всех случаях любая ошибка ББП отсылает обращение к ОП и вызывает обновление данных ББП.

Все используемые разряды, кроме разряда 8, пишутся и читаются. Разряд 8 только пишется, читается всегда нулем. Регистр сбрасывается в нуль при включении питания или пуске программы. Команда RESET на него не воздействует.

**Регистр HMR** (адрес 1777752) фиксирует информацию о последних шести обращениях к ББП и выполнен в виде регистра сдвига, который запоминает попадание как единицу, а промах как нуль для последних операций чтения. Попадание означает, что данные извлекались из ББП, а промах — что они были получены из ОП. Разряд 0 отражает самое последнее обращение к памяти, затем его значение сдвигается влево при каждом последующем обращении к ОП. Регистр доступен только для чтения. Формат регистра HMR показан на рис. 10.

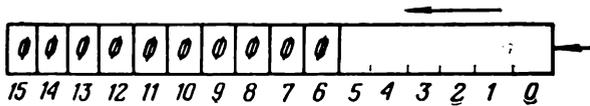


Рис. 10. Формат регистра HMR

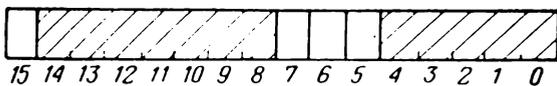


Рис. 11. Формат регистра MSER

**Регистр MSER** (адрес 1777744) фиксирует прекращение операции и тип обнаруженной ошибки. Формат регистра приведен на рис. 11. Содержимое разрядов следующее.

Разряд 15 устанавливается в единицу, если ошибка ББП или ОП прекратила выполнение операции. Прекращение операции происходит при всех некорректируемых ошибках ОП и ошибках ББП, если разряд 7 регистра CCR установлен в единицу. Ошибки паритета ББП обнаруживаются на циклах шины, которые не являются циклами предварительной выборки.

Разряды 14...8, 4...0 в СМ 1425 не используются.

Разряд 7 устанавливается в единицу при ошибке в старшем байте данных.

Разряд 6 устанавливается в единицу при ошибке в младшем байте данных.

Разряд 5 устанавливается в единицу при ошибке в поле признаков.

Разряды 7, 6, 5 устанавливаются индивидуально по ошибкам ББП, если CCR[7]=1. Если CCR[7]=0, то все три разряда устанавливаются в единицу по любой ошибке ББП. Регистр MSER доступен только для чтения и сбрасывается в нуль при любом обращении к нему с операцией записи,

включении питания и пуске с пульта управления. Команда RESET на него не воздействует.

## Средства преобразования адресов

Прежде чем описывать принципы и аппаратные средства преобразования адресов, определим основные понятия: виртуальное адресное пространство, физическое адресное пространство, область регистров периферийных устройств.

Поскольку для адресации команд и операндов процессор использует 16-разрядные регистры (PC, SP, R0 и т. д.), программа может использовать адреса между 0 и 177777, т. е. адресовать 32 Кслов памяти, которые определены как виртуальное адресное пространство программы. С целью увеличения объема виртуального адресного пространства введена возможность преобразования адресов команд и операндов в два независимых виртуальных пространства по 32 Кслов каждое: пространство команд — I и пространство команд — D. Таким образом, программа может использовать 64 Кслов памяти.

Физическое адресное пространство памяти — область ОП, адресуемая при помощи аппаратуры преобразования адресов. Объем физического адресного пространства зависит от разрядности сформированного диспетчером адреса. В СМ 1425 адрес после преобразования содержит 22 разряда и, следовательно, позволяет адресовать до 2048 Кслов памяти. Верхние 4 Кслов этого пространства (адреса 17760000...17777777) используются для адресации регистров периферийных устройств и внутренних регистров процессора.

В СМ 1425 реализовано динамическое преобразование адресов памяти, обеспечиваемое аппаратными и программными средствами. К аппаратным средствам относится диспетчер памяти, который выполняет преобразование из одного адресного пространства в другое, позволяя расширить емкость произвольно адресуемой памяти. Программные средства преобразования — это входящие в состав операционных систем программы. Виртуальные программные адреса процессора преобразуются в физические путем 16-, 18- или 22-разрядного адресного преобразования. Операции адресного преобразования могут быть использованы для обеспечения программной совместимости вычислительного комплекса СМ 1425 с другими моделями ЭВМ ряда СМ-3...СМ 1420.

16-разрядное преобразование — это непосредственное перемещение из виртуальных адресов в физические. Нижние 28 Кслов виртуальных адресов соответствуют таким же физическим адресам. Область старших 4 Кслов отображается в область адресов регистров периферийных устройств (17760000...17777777). 16-разрядное преобразование используется процессором после включения питания, пуска программы с пульта управления и после команды RESET, когда отключается режим преобразования в диспетчере памяти. Режим преобразования может быть также отключен программной перезагрузкой регистра диспетчера MMR0. Данный режим адресного преобразования определяется состоянием разрядов MMR0[0]=1 и MMR3[4]=0 регистров диспетчера памяти.

18-разрядное преобразование виртуальных адресов в физические выполняется диспетчером памяти следующим образом. Виртуальные адреса 32 Кслов для каждого из режимов «Ядро», «Супервизор» и «Пользователь» отображаются в физические адреса 128 Кслов. Физические адреса старших 4 Кслов преобразуются в адреса регистров периферийных устройств.

22-разрядное преобразование виртуальных адресов при включенном диспетчере ( $MMR0[0]=1$ ) определяется единичным значением регистра  $MMR3[4]$ . При этом преобразовании используются полные 22-разрядные адреса доступа к физической памяти. Старшие 4 Кслов — это та же область регистров периферийных устройств.

## Диспетчер памяти

Диспетчер памяти (ДП) содержит оборудование для диспетчеризации и защиты памяти. ДП процессора имеет емкость доступной памяти 4 Мбайт, три режима управления выборкой памяти — «Ядро», «Супервизор» и «Пользователь», 16 страниц в каждом режиме (по 8 страниц для команд и данных). Длина страницы 64 ... 8192 байт (1 ... 128 блоков). Каждая страница обеспечивается полной защитой и возможностью перемещения.

## Регистры диспетчера памяти

В ДП входят три набора регистров, используемых для преобразования адресов в режимах «Ядро», «Супервизор» и «Пользователь». Каждый из этих наборов содержит две группы по 8 пар регистров адреса страницы (PAR) и регистров описания страницы (PDR). Одна группа регистров используется при обращении в пространство D, другая группа — при обращении в пространство I.

Адреса регистров в режиме «Ядро»: пространство данных — PAR 17772360 ... 17772376, PDR 17772320 ... 17772336; пространство команд — PAR 17772340 ... 17772356, PDR 17772300 ... 17772316.

Адреса регистров в режиме «Супервизор»: пространство данных — PAR 17772260 ... 17772276, PDR 17772220 ... 17772236; пространство команд — PAR 17772240 ... 17772256, PDR 17772200 ... 17772216.

Адреса регистров в режиме «Пользователь»: пространство данных — PAR 17777660 ... 17777676, PDR 17777620 ... 17777636; пространство команд — PAR 17777640 ... 17777656, PDR 17777600 ... 17777616.

Регистр PAR определяет начальный адрес страницы как номер блока в физическом адресном пространстве.

Формат регистра PDR показан на рис. 12. Содержимое разрядов следующее.

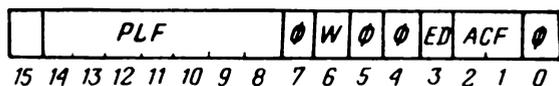


Рис. 12. Формат регистра PDR

Разряд 15 в единичном состоянии вызывает обход ББП, т. е. обращение адресуется к ОП. При этом в случае попадания в операциях чтения или записи устанавливается в нуль разряд достоверности для данного слова ББП.

Разряды 14 ... 8 (поле длины страницы PLF) за-

дают номер блока, определяющий границу разрешения области страницы. При обращении к ОП номер блока в виртуальном адресе сравнивается с кодом в поле длины страницы. Ошибка длины страницы фиксируется, если при расширении страницы вверх номер блока в виртуальном адресе больше кода в поле длины страницы или при расширении страницы вниз номер блока в виртуальном адресе меньше кода в поле длины страницы.

Разряды 7, 5, 4, 0 не используются.

Разряд 6 служит признаком записи в страницу (W). В единичном состоянии он показывает, что информация в странице изменена, т. е. с момента загрузки в ОП в страницу была внесена запись. Разряд автоматически сбрасывается в нулевое состояние при записи в PAR или в PDR данной страницы.

Разряд 3 (ED) определяет направление расширения страницы. Значение  $ED=0$  устанавливает направление расширения вверх, при котором разрешенными являются блоки с номерами от нулевого до номера, установленного в поле PLF, включительно. Значение  $ED=1$  устанавливает направление расширения страницы вниз, при котором разрешенными являются блоки с номерами от 127 (177) до номера, установленного в поле PLF, включительно.

Разряды 2 и 1 содержат код доступа (ACF) к странице ОП. Этот код определяет разрешенные операции обращения к данной странице. Обращение к странице с операциями, которые запрошены кодом в поле ACF, вызовет прекращение выполнения операции и прерывание по вектору с адресом 250. Кодировка поля ACF следующая: 00 — страница недоступна (прерывание при любом обращении); 01 — страница доступна для чтения (прерывание при операциях записи); 10 — не используется (прерывание при любом обращении); 11 — страница доступна для операций чтения и записи.

К разряду 6 можно обратиться только с операцией чтения, к остальным разрядам — с операциями чтения и записи.

Регистр  $MMR0$  (адрес 17777572) обеспечивает управление ДП и фиксирует информацию при обнаружении ошибок. Формат регистра показан на рис. 13. Содержимое разрядов следующее.

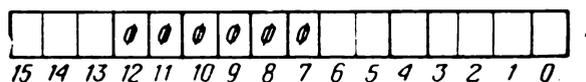


Рис. 13. Формат регистра  $MMR0$

В разрядах 15 ... 13 фиксируется информация, указывающая на ошибки, обнаруженные при работе ДП. Разряд 15 устанавливается в единицу при попытке обратиться в недоступную страницу ( $ACF=0$  или  $ACF=2$ ) или при попытке использовать перемещение адресов в запрещенном режиме работы процессора (режим 10). Разряд 14 устанавливается в единицу при попытке обращения к ячейке с адресом, выходящим за границы разрешенной области, указанной в поле PLF регистра PDR. Разряд 13 устанавливается в единицу при попытке произвести запись в страницу, доступную только для чтения.

Разряды 12 ... 7 не используются, читаются нулями.

В разрядах 6 и 5 фиксируется код режима ра-

боты процессора («Ядро» — 00, «Супервизор» — 01, «Пользователь» — 11, запрещенный — 10), при котором выполнилось обращение, вызвавшее ошибку. Если указывается запрещенный режим, генерируется прерывание и устанавливается разряд 15.

В разряде 4 фиксируется признак адресного пространства, при обращении к которому была обнаружена ошибка: если этот разряд установлен в единицу — пространство D, если сброшен в нуль — пространство I.

В разрядах 3...1 фиксируется номер страницы, при обращении к которой была обнаружена ошибка.

Состояние разряда 0 управляет работой ДП. Если он установлен в единицу, разрешено преобразование адресов и выполняются все действия, связанные с защитой памяти. Если он сброшен в нуль, преобразование адресов и действия по защите памяти не выполняются.

К разрядам 6...1 возможен доступ только с операциями чтения, к остальным разрядам — с операциями чтения и записи. Разряды 15...13 могут устанавливаться непосредственной записью, однако этот способ установки не приведет к прерыванию. Независимо от того, установлены разряды путем непосредственной записи или вследствие прерывания, установка в единицу любого из разрядов 15...13 ведет к тому, что ДП «замораживает» содержимое MMR0[6...1], MMR1, MMR2. Содержимое разрядов сохраняется, пока разряды MMR0[15...13] не будут сброшены в нуль непосредственной записью. Разряды MMR0[15...13, 0] сбрасываются в нуль также при включении питания и пуске программы.

**Регистр MMR1** (адрес 1777574) фиксирует величину любого автоувеличения или автоуменьшения регистра общего назначения, включая использование регистра R7 (PC). Величина, на которую увеличивается или уменьшается регистр общего назначения, записывается в дополнительном коде. Младший байт регистра MMR1 используется для всех команд с операндами-источниками, а операнды-приемники могут использовать любой байт в зависимости от режима и типа команды. Формат регистра показан на рис. 14. Содержимое разрядов следующее.

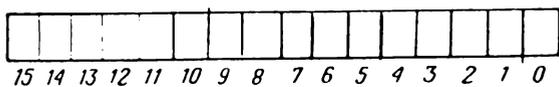


Рис. 14. Формат регистра MMR1

В разрядах 15...11 фиксируется величина увеличения или уменьшения в дополнительном коде для регистра, определяемого в разрядах 10...8, которые указывают один из восьми регистров общего назначения.

В разрядах 7...3 фиксируется величина увеличения или уменьшения в дополнительном коде для регистра общего назначения, определяемого в разрядах 2...0.

Регистр MMR1 доступен только для чтения и сбрасывается в нуль в начале выборки каждой команды.

**Регистр MMR2** (адрес 1777576). В него загружается счетчик программы при выборке текущей команды и «замораживается» по любому прерыванию, которое регистрируется в MMR0.

**Регистр MMR3** (адрес 1777516) разрешает обращения к пространствам данных для режимов «Ядро», «Супервизор» и «Пользователь», также устанавливает 22-разрядное преобразование адреса при включении ДП и разрешает выполнение команды «Вызов супервизора» (CSM). Формат регистра приведен на рис. 15. Содержимое разрядов следующее.

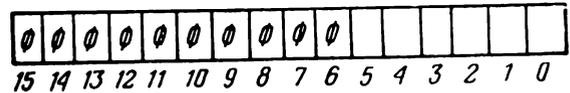


Рис. 15. Формат регистра MMR3

Разряды 15...6 не используются, читаются нулями.

Разряд 5 устанавливается в единицу или сбрасывается в нуль под управлением программы, но процессором не анализируется.

Разряд 4 разрешает 22-разрядное преобразование адреса (при MMR3[4]=0 — 18-разрядное преобразование адреса).

Разряд 3, если он установлен в единицу, разрешает выполнение команды CSM. При MMR3[3]=0 выполнение этой команды вызовет прерывание по несуществующему коду команды.

Если один из разрядов 2...0 установлен в единицу, то разрешается преобразование адресов через I- и D-пространства соответствующего режима. Если разряд сброшен в нуль, то все обращения идут только через пространство I соответствующего режима. Разряд 2 разрешает использование пространства D для режима «Ядро», разряд 1 — для рабочего режима «Супервизор», разряд 0 — для рабочего режима «Пользователь».

Регистр MMR3 сбрасывается в нуль во время включения питания, пуска программы или по команде RESET.

## Формирование физического адреса

При каждом преобразовании адреса происходит обращение в пространство I или пространство D. Пространство I используется для выборки всех команд, индексных слов, абсолютных адресов и непосредственных операндов; пространство D — для всех других обращений. Информация, необходимая для формирования физического адреса, берется из виртуального адреса и регистра PAR. Виртуальный адрес может быть разбит на следующие составные части:

поле активной страницы (разряды 15...13). Определяет, какой из восьми регистров адреса страницы из набора PAR (PAR0...PAR7), заданного кодом режима («Ядро», «Супервизор», «Пользователь») и типом пространства (I или D), будет использоваться для формирования физического адреса;

номер блока (разряды 12...6). Определяет номер блока объемом в 32 слова относительно начала страницы.

смещение в блоке (разряды 5...0). Определяет адрес ОП в пределах блока.

Содержимое выбранного регистра PAR всегда определяет начальный адрес страницы. Формирование физического адреса состоит в суммировании содержимого PAR, выбранного по коду в разрядах 15...13 виртуального адреса, с номером блока вир-

туального адреса и подстановке в качестве младших разрядов физического адреса смещения в блоке из виртуального адреса. Схема формирования физического адреса приведена на рис. 16. Как видно из рисунка, положение страницы в ОП можно задавать с дискретностью в 32 слова.



Рис. 16. Схема формирования физического адреса ДП

Прерывания, вызываемые оборудованием ДП, производятся по вектору с адресом 250 в режиме «Ядро». Регистры MMR0, MMR1, MMR2, MMR3 используются для уточнения причины прерывания и перезапуска программы, вызвавшей его. При выявлении оборудованием ДП причины прерывания информация, связанная с прерыванием, фиксируется в перечисленных регистрах и «замораживается» в них до очистки MMR0[15...13] в конце выполнения программы обработки прерывания. В течение этого интервала времени информация о новых ошибках, обнаруженных ДП, фиксироваться не будет.

## Системный интерфейс

Связь процессора с оперативной памятью и другими устройствами комплекса осуществляется через магистральный параллельный интерфейс (рис. 17). МПИ реализуется на основе магистрали и логических узлов, входящих в каждое подключаемое к ней устройство.

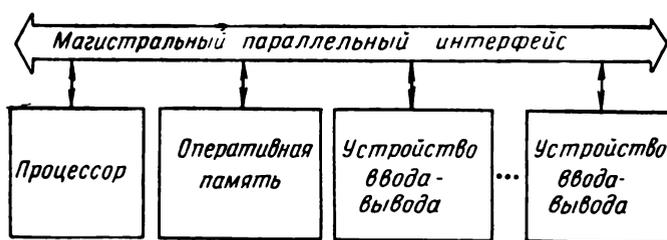


Рис. 17. Обобщенная структурная схема подключения устройств к МПИ

Магистраль состоит из линий, объединенных в группы по функциональному назначению: шины обмена информацией, шины управления обменом, шины передачи управления, шины прерывания, вспомогательной шины. Большинство линий магистрали осуществляют двустороннюю передачу сигналов. К таким линиям все устройства комплекса подключены параллельно. Небольшая группа линий предназначена для односторонней передачи.

В любой операции обмена всегда участвуют два устройства, связанные между собой как задатчик (управляющее устройство) и исполнитель (управляемое устройство). Одновременно двух и более за-

датчиков на МПИ быть не должно. В каждый момент времени на магистрали может выполняться один из трех видов взаимодействия между подключенными к ней устройствами: обмен информацией, передача управления магистралью, прерывание.

Интерфейс используется процессором и всеми внешними устройствами с разделением во времени и в соответствии с установленным приоритетом. По отношению к интерфейсу процессор рассматривается как внешнее устройство с изменяемым приоритетом. Этот приоритет задается в слове состояния процессора PSW и может изменяться программно. Остальные устройства на интерфейсе имеют фиксированные уровни приоритетов.

Каждое подключаемое к магистрали устройство имеет в своем составе один или более адресуемых по магистрали регистров. Регистры устройств ввода-вывода, программно доступные регистры процессора и все ячейки оперативной памяти составляют единое адресное пространство МПИ. Это позволяет процессору обрабатывать данные из регистров устройств без пересылки их в ячейки памяти или в свои регистры.

Вся область адресов МПИ может быть разбита на три зоны: зона адресов регистров периферийных устройств и регистров процессора — 17777776...17760000; зона адресов ячеек памяти — 17757776...00001000; зона постоянно распределенных адресов памяти 00000776...00000000. Подробное описание МПИ приведено в приложении 3.

## Система прерываний

Прерывания служат для организации ввода-вывода информации, а также для сообщения системе о возникновении необычных и запланированных программных ситуаций в самом процессоре. По принципу организации ввода-вывода устройства можно разделить на программно управляемые и устройства с прямым доступом в ОП (УПД).

В простейшем случае ввод-вывод информации для устройств первого типа производится по следующему алгоритму. Процессор с помощью обычных команд читает или записывает информацию в соответствующий регистр данных контроллера периферийного устройства. Достоверность информации при выводе или готовность принять следующий байт при вводе определяются разрядом готовности в управляющем регистре. Ожидание достоверной информации может осуществляться программным опросом разряда готовности или через прерывание. В первом случае процессор постоянно остается на программе обслуживания периферийного устройства, во втором — процессор, записав очередной байт в регистр данных, может перейти на выполнение другой программы. После завершения операции периферийным устройством контроллер установит запрос на прерывание.

Для УПД процессор заносит в соответствующие регистры информацию об объеме массива данных, адресе в ОП, месте расположения на носителе, а также команду. При получении команды типа «чтение/запись» внешнее устройство осуществляет обмен данными с ОП без участия процессора (прямой доступ в память). Запрос на прямой доступ устанавливается по линии BDMRL на МПИ. Устройство, установившее запрос на линии, получит от ар-

бита процессора разрешение по линии BDMGL, не ожидая выполнения очередной команды процессора. Далее, если МПИ не занят другим задатчиком, задатчиком становится УПД. Процессор при этом, если ему нужен МПИ, приостанавливает выполнение команды, а после освобождения МПИ продолжает ее выполнение. УПД может занимать МПИ на время передачи от 1 до 16 слов данных. По завершении операции ввода-вывода УПД устанавливает запрос на прерывание. Для этого УПД должно быть подключено к одной из четырех линий программных прерываний BIRQ7L...BIRQ4L, определяющих уровень приоритета его прерывания. При одном уровне более высокий приоритет имеет устройство, которое физически ближе подключено к процессору.

Процессор рассматривается как устройство с изменяемым уровнем приоритета, задаваемым программно с помощью разрядов PSW[7...5]. В табл. 7

Таблица 7

Причина прерывания	Адрес вектора прерывания	Уровень приоритета
RWRF	24	Н/М
FPE	244	Н/М
PARITY	114	Н/М
BIRQ7	Устанавливается на устройстве ввода-вывода	7
BEVNT	100	6
BIRQ6	Устанавливается на устройстве ввода-вывода	6
BIRQ5	То же	5
BIRQ4	»	4
HALT	—	См. примечания

Примечания. 1. Н/М — запрос не маскируется. 2. Сигнал, поступающий по линии BHALTL, обычно имеет самый низкий приоритет. Однако при чтении вектора прерывания он получает наивысший приоритет. Это дает возможность пользователю прерывать бесконечное закликивание программы, которое возникает, если при чтении вектора обнаруживаются условия прерывания программы.

приведены внешние по отношению к процессору запросы на прерывание. Их можно разделить на маскируемые и немаскируемые. Маскируемые запросы приходят по линиям BEVNTL (запрос на прерывание от таймера) и BIRQ7L...BIRQ4L. При получении запроса по одной из этих линий процессор сравнивает приоритет запроса с собственным приоритетом. В случае, если приоритет процессора выше или равен приоритету запроса, прерывание не происходит. Если же приоритет запроса выше, процессор выдает сигнал разрешения передачи по линии BIAKL. После выполнения текущей команды процессор начинает выполнять процедуру прерывания, считывая с шин МПИ адрес вектора прерывания, установленный устройством.

К немаскируемым запросам относятся следующие:

BHALTL — запрос на останов, переводит процессор в режим эмулятора пульта;

PWRFL — нарушение в подаче питания, вызывает прерывание по вектору с адресом 24;

FPE — исключительная ситуация в ускорителе

команд с плавающей запятой (ПЗ), вызывает прерывание по вектору с адресом 244;

PARITY — ошибка при обращении к памяти, вызывает прерывание по вектору с адресом 114.

Для сообщения системе о возникновении необычных или запланированных программных ситуаций служат внутренние прерывания (табл. 8). При обработке внутренних прерываний анализируется состояние следующих регистров процессора:

CPUErr, фиксирующего ошибочные ситуации в программном обеспечении;

PIRQ[15...9], обеспечивающего семь уровней запросов на программное прерывание;

MMR0, фиксирующего ошибки диспетчера памяти;

FPS и FEC, фиксирующих ошибочные ситуации при выполнении команд с плавающей запятой.

Реакция процессора при возникновении запроса на прерывание может быть двойкой: выполнение команды завершается или выполнение команды подавляется (см. табл. 8).

Таблица 8

Адрес вектора прерывания	Причина прерывания	Выполнение текущей команды
4	Нечетная адресация, тайм-аут Нарушение «желтой зоны» стека Нарушение «красной зоны» стека Команда HALT	Подавляется Завершается Подавляется См. примечание
10	Несуществующий код команды Несуществующий режим адресации Команда «Вызов супервизора» CSM	Подавляется » Завершается
14	Команда BPT Прерывание по T разряду PSW	» »
20	Команда IOT	»
24	Нарушение в системе питания	»
30	Команда EMT	»
34	Команда TRAP	»
114	Ошибка по паритету при обращении к памяти	Подавляется
240	Маскируемые программные прерывания уровней 7...1 по содержимому регистра PIRQ [15...9]	Завершается
244	Исключительные ситуации команд с ПЗ: несуществующий код команды; несуществующий режим адресации; деление на нуль; ошибка преобразования числа с ПЗ в число с фиксированной запятой (ФЗ); переполнение порядка; исчезновение порядка; неверные данные	Подавляется » » Завершается » » Завершается или подавляется
250	Ошибка диспетчера памяти	Подавляется

Примечание. Команда HALT может выполняться по-разному в зависимости от режима процессора и режима выполнения самой команды HALT. В режиме «Ядро» при MR[3]=0 процессор переходит в режим эмулятора пульта, а при MR[3]=1 происходит прерывание по вектору с адресом 4. В режимах «Пользователь» и «Супервизор» команда HALT вызывает прерывание по вектору с адресом 4.

Стандартная процедура программного прерывания состоит из записи в ОП информации о прерываемой программе и перехода с помощью вектора прерывания на программу, обслуживающую данный тип прерывания. Информация о прерванной программе записывается в стек обслуживающей программы в виде двух слов: первое слово (старший адрес) — содержимое PSW, второе (младший адрес) — содержимое PC, указывающее адрес команды, на которую надо вернуться после завершения обслуживающей программы. Вектор прерывания представляет собой два смежных слова ОП: первое слово (младший адрес) определяет содержимое PC, второе — PSW обслуживающей программы. При записи в регистр PSW все используемые разряды, кроме разрядов 13 и 12, устанавливаются по информации из ОП. В разряды 13, 12 переписывается значение разрядов 15, 14.

Программа, обслуживающая прерывание, может в свою очередь быть прервана более приоритетным прерыванием. Это наслаивание приоритетных прерываний может продолжаться до любого уровня и ограничено только объемом памяти, введенной под стек. Обычно программа обслуживания прерывания выполняет переход на прерванную программу с помощью команды RTI или RTT, которая извлекает два слова из стека и загружает их обратно в регистры PSW и PC. Если во время процедуры перехода на программу обработки прерываний возникает прерывание по нечетной адресации или тайм-ауту, процессор выполняет прерывание по вектору с адресом 4.

## Архитектура дисковой памяти

Комплекс СМ 1425 использует новый подход к организации взаимодействия центрального процессора, оперативной памяти и контроллеров внешних устройств с прямым доступом типа дисковой памяти. Сущность подхода состоит в упорядочении аппаратных и программных средств, образующих типовую подсистему массовой памяти или, иначе, архитектуру массовой памяти (АМП). Данная архитектура предполагает три независимых слоя (рис. 18), связанных между собой стандартными интерфейсами. Это позволяет производить изменения внутри одного слоя без воздействия на другие, что дает возможность объединять и согласовывать различные аппаратные и программные ресурсы.

В рамках АМП команды и данные передаются через интерфейсы от слоя к слою в форме пакетов сообщений. В главном слое центральный процессор выполняет прикладные пользовательские программы, которым требуется массовая память. Различные функции слоя контроллера обеспечивают главному слою возможность безошибочной записи или чтения данных по мере необходимости с его собственной скоростью. Главный слой использует для выполнения операций ввода-вывода два подслоя программного обеспечения. Подслой, называемый драйвером массовой памяти, в ответ на требования от прикладных программ пользователей создает пакеты сообщений в соответствии с протоколом управления массовой памятью для того, чтобы выполнять функции ввода-вывода общего характера, такие, как чтение и запись. В этом же подслое драйвер диагностики строит пакеты сообщений в соответствии с протоколом диагностики. Другой под-

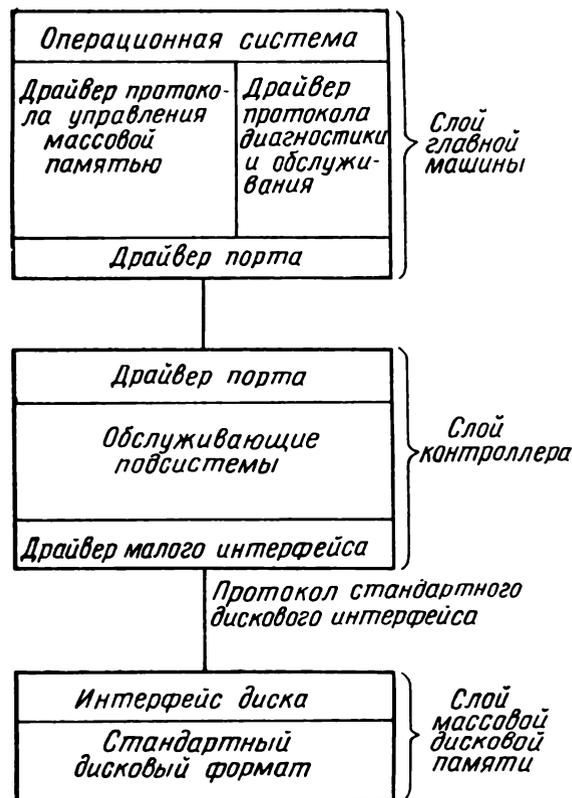


Рис. 18. Компоненты архитектуры массовой памяти

слой, драйвер портов, постоянно находящийся в главном слое, передаст пакеты по шине между главной машиной и контроллером.

Слой контроллера включает в себя драйвер портов, который получает сообщения от драйвера портов главной машины или передает ему сообщения. Две обслуживающие подсистемы — одна для сообщений протокола массовой памяти и другая для сообщений протокола диагностики — представляют собой программируемую часть контроллера и как таковые определяют функциональные особенности подсистемы. Контроллер содержит также программную часть, управляющую графиком сообщений между контроллером и накопителем.

Существенным отличием принятой архитектуры от архитектуры СМ 1420 является то, что используется единственный драйвер массовой памяти, управляющий всем обменом сообщений между заданной операционной системой и любым устройством массовой памяти определенного класса — НМД или НМЛ. Драйвер массовой памяти не содержит никакой информации, относящейся к конкретному контроллеру или НМД (НМЛ). Таким образом, для любого числа типов контроллеров или накопителей, работающих в определенной вычислительной системе, нужен драйвер только одного класса.

Драйвер, содержащийся в главной машине, освобождается от хранения данных о диске. Контроллер и диск, работая совместно, передают в главную машину только данные. Все необходимые действия по обнаружению ошибок, их исправлению и восстановлению работоспособности системы выполняются в контроллере.

## Состав комплекса

На начало производства запланирован выпуск двух типовых комплексов СМ 1425.01 и СМ 1425.02,

отличающихся друг от друга только объемом оперативной и внешней памяти. По мере развития технических и программных средств, а также исходя из опыта эксплуатации СМ 1425, могут быть разработаны новые конфигурации.

Основой типового комплекса является базовый блок. Он выполнен в виде небольшой стойки (тумбы), которую рекомендуется устанавливать рядом с рабочим местом оператора. Другие составные части комплекса — видеотерминалы, печатающие устройства — имеют настольное исполнение и могут располагаться на удалении от базового блока. Электронные устройства выполнены в виде одноплатных модулей, представляющих собой блоки элементов на плате типа Е2 размером 220 × 233,4 мм. В базовом блоке имеется монтажный блок для установки восьми БЭ.

В типовых комплексах, составы которых приведены в табл. 9, занято шесть мест. Два свободных места могут быть использованы для расширения комплексов. Предусмотрена возможность подключения к базовому блоку аналогичного ему по конструкции дополнительного блока, что позволяет строить комплексы с применением более восьми БЭ. Устройствами расширения базового комплекса являются: устройство согласования системных интерфейсов — МПИ и ОШ СМ 1425.4511, контроллер внешней памяти на МЛ бобинного типа СМ 1425.5021, контроллер внешней памяти на МЛ

потокowego типа СМ 1425.5020, адаптер дистанционной связи синхронный.

Таблица 9

Наименование составных частей типовых комплексов	Количество на исполнение	
	СМ 1425.С1	СМ 1425.02
Процессор	1	1
Модуль оперативный запоминающий СМ 1425.3537.02	1	1
Накопитель на магнитном диске СМ 5508	2	—
Накопитель на магнитном диске СМ 5509	—	2
Накопитель на гибком магнитном диске СМ 5640	2	2
Контроллер НМД и НГМД СМ 1425.5140	1	1
Видеотерминал СМ 7238 (или СМ 6329.02)	2	2
Печатающее устройство СМ 6329.02	1	1
Контроллер интерфейсов групповой (КГИ) СМ 1425.7009	1	1
Мультиплексор передачи данных СМ 1425.8540 (СМ 1425.8544)	1	1
Операционная система ОС РВМ	1	1

## КОНСТРУКТИВНО-ЭЛЕМЕНТНАЯ БАЗА

### Конструкция

Конструктивно комплекс выполнен в виде тумбы, устанавливаемой под столом или рядом с ним, и внешних устройств, располагаемых на столах. Основными конструктивными элементами комплекса являются: блок элементов, блок монтажный (БМ), корпус, тумба. БЭ — многослойная печатная плата с расположенными на ней электро- и соединительными элементами. На плате закреплен экстрактор для установки и фиксации БЭ в монтажной панели БМ. БМ — несущая конструкция, на каркасе которой размещаются блок электропитания, системный пульт, монтажная панель и направляющие для установки восьми БЭ, два накопителя на жестких магнитных дисках, два накопителя на гибких магнитных дисках и распределительная плата для подключения накопителей. Внешние устройства (видеотерминалы, печатающие устройства) подключаются через распределительную панель. Межблочные соединения БМ выполнены кабелями и перемычками. Для организации вентиляции в БМ имеются три вентилятора, два из которых установлены непосредственно в блоке электропитания.

БМ, помещенный в корпус, представляет собой тумбу. Корпус состоит из двух жестко соединенных стенок, передней и задней панели, фиксирующихся с помощью защелок. Для установки изделий на пол на корпусе закреплена подставка. В нижней и верхней частях корпуса выполнены прорезы для обеспечения вентиляции устройств и блоков. Схема вентиляции вертикальная. Для подключения внешних устройств снимается задняя крышка и обеспечивается доступ к распределительной панели.

### Элементная база

Вычислительный комплекс СМ 1425 использует современную элементную базу, основу которой составляют микропроцессоры, созданные по К/МОП- и n — МОП-технологии, полужаказные схемы на базовых матричных кристаллах, элементы памяти различного уровня интеграции и быстродействия.

### Центральный микропроцессор

Центральный микропроцессор (ЦМП) используется в блоке элементов процессора и представляет собой гибридную микросборку на керамической многослойной плате с 60 выводами. На плате размещены две микросхемы: КН1831ВМ1 (обрабатывающая часть) и КН1831ВУ1 (управляющая часть) с 84 выводами каждая. Микросхема КН1831ВМ1 выполняет все арифметические и логические функции, преобразование адресов и генерирует сигналы синхронизации ввода-вывода. Помимо непосредственно обрабатывающей части эта микросхема содержит диспетчер памяти, управление вводом-выводом, управляющие регистры для арифметики с плавающей запятой и оперативного буферного запоминающего устройства. Микросхема КН1831ВУ1 управляет работой КН1831ВМ1 по микрокомандам. Основная составляющая часть КН1831ВУ1 — устройство микропрограммного управления и логика управления чередованием микрокоманд. Максимальная частота работы схемы 15 МГц.

Значительная доля быстродействия ЦМП достигается благодаря механизму предвыборки и предварительного декодирования. При этом обращения к памяти перекрываются с внутренними операциями и необходимость явной выборки инструкции и циклов декодирования делается минимальной. Операции предвыборки и преддекодирования выполняются в ЦМП автоматически и не могут быть изменены пользователем. Главная функция механизма предвыборки — наполнение четырех регистров информацией и дальнейшее их соответственное пополнение. Эти четыре регистра: виртуальный счетчик программ (VPC), физический счетчик программ (PPC), буфер предвыборки (PB) и регистр инструкций (IR), имеют коллективное название — конвейер предвыборки. Содержимое регистров в начале конвейера используется для определения содержимого регистров далее по конвейеру. Когда конвейер заполнен, говорится, что механизм предвыборки находится в устойчивом состоянии. Процесс накопления конвейера следующий:

Микроцикл 1 Микроцикл 2 Микроцикл 3 Микроцикл 4  
 VPC ← PC PPC ← MMU PB ← M[PPC] IR ← PB  
 (VPC) PPC ← MMU PB ← M[PPC]  
 VPC ← (VPC) PPC ← MMU  
 ← VPC + 2 VPC ← (VPC)  
 ← VPC + 2 VPC ←  
 ← VPC + 2  
 PC ← PC + 2  
 MMR2 ← PC

В микроцикле 1 VPC просто устанавливается равным PC. В микроцикле 2 VPC посылается через MMU (диспетчер памяти), и результирующий физический адрес загружается в PC. Затем VPC увеличивается на 2. В этот момент мы имеем правильные VPC и PPC и говорится, что конвейер синхронизирован. Иногда при выполнении макроинструкции конвейер является синхронизированным, но не заполненным. В таком случае нужно выполнить микроциклы 3 и 4 для следующей макроинструкции. В микроцикле 3 слово в памяти, адресуемое PPC, выбирается и поступает в PB. В PPC поступает перемещенный (отображенный) VPC и затем VPC снова получает приращение. В микроцикле 4 PB посылается в IR и декодируется как следующая макроинструкция (заметим, что ЦМП взводит PRDC в этот момент). Новое содержимое PB выбирается из ячейки памяти, адресуемой PPC. В PPC снова поступает перемещенный (отображенный) VPC, и VPC снова получает приращение. В течение того же микроцикла 4 прежнее PC загружается в MMR2 (если MMR0[5...13] = 000) и затем увеличивается на 2.

В устойчивом состоянии, т. е. когда микроцикл 4 завершен, IR содержит выполняемую макроинструкцию, PB — данные из ячейки памяти, указываемой PC, PPC — физический адрес следующего слова, которое должно быть предвыбрано, и VPC — приращенное значение PC. После входа в устойчивое состояние поток макроинструкций, работающих только с регистрами, может выполняться со скоростью одна инструкция на микроцикл (т. е. микроцикл 4). В то время, как одна инструкция выполняется, следующая декодируется и следующая предвыбирается в PB. В течение микроцикла 4 содержимое буфера предвыборки загружается в IR, а слово, адресуемое PPC, — в PB, VPC перемещается и загружается в PPC и VPC приращивается на 2. Таким образом поддерживается устойчивое состояние, что позволяет в следующем микроцикле выполнить следующую макроинструкцию. Заметим также, что шина ЦМП остается занятой 100% времени.

Инструкции, работающие с непосредственными данными и регистром, также используют механизм предвыборки максимально. В устойчивом состоянии поток этих макроинструкций выполняется в два микроцикла (микроциклы 3 и 4). В течение микроцикла 3 данные PB передвигаются в служебный регистр. В течение микроцикла 4 операция выполняется. В обоих циклах устойчивое состояние механизма предвыборки поддерживается предвыборкой следующего слова инструкции. Шина ЦМП снова занята 100% времени.

Конвейер предвыборки снова заполняется после последовательности восстановления питания или если происходит ошибка предвыборки. Последнее случается при записи регистров PS, CCR, PC или любых регистров диспетчера памяти. Ошибка предвыборки нарушает правильность только PB. Это означает, что конвейер остается синхронизированным и может быть снова заполнен за два микроцикла.

Рассмотрим следующую программу в качестве примера потока в конвейере:

Виртуальный адрес	Символическое представление	Восьмеричный код
1000	MOV R2, R3	010203
1002	BIS #1, R3	052703
		000001
1004	ADD R1, R3	060105
1006	CLR R0	005000
1012	ADD R3, R0	060300

Поток информации через конвейер происходит, как показано в табл. 10.

Таблица 10

Регистр конвейера	Микроциклы					
	N	N+1	N+2	N+3	N+4	N+5
PC	1002	1004	1006	1010	1012	1014
IR	MOV (010203)	BIS (052703)	BIS (052703)	ADD (060105)	CLR (005000)	ADD (060300)
PB	BIS (052703)	000001	ADD (060105)	CLR (005000)	ADD (060300)	*
PPC	MMU (1004)	MMU (1006)	MMU (1010)	MMU (1012)	MMU (1014)	MMU (1016)
VPC	1006	1010	1012	1014	1016	1020

\* Инструкция в ячейке 1014.

Заметим, что этот пример начинается с микроцикла N, ко времени которого конвейер предвыборки уже заполнен, т. е. находится в устойчивом состоянии. Все инструкции в примере выполняются за один микроцикл, за исключением инструкции BIS, выполняемой за два микроцикла.

Отличительной особенностью ЦМП является также встроенный микропрограммный отладчик, который позволяет эмулировать функции пульта с системного терминала.

### Периферийный микропроцессор

Периферийный микропроцессор K1807BH1 представляет собой микросхему с 40-выводным двухрядным корпусом, реализует базовую систему команд СМ ЭВМ и используется в периферийных интеллектуальных контроллерах (например, в контроллере памяти на магнитных дисках). Он имеет встроенный контроллер динамической и статической памяти, систему прерываний по четырем уровням, встроенное управление прямым доступом. Максимальная частота синхросигналов 7,5 МГц.

### Микросхемы памяти

Микросхема K565PY7 используется в качестве основного элемента оперативной памяти динамиче-

ского типа комплекса СМ 1425. Емкость микросхемы 256К×1 бит в корпусе, время выборки относительно сигнала выбора адреса столбцов не более 125 нс. Управление накопителем памяти осуществляется от 16-разрядной схемы обнаружения и коррекции ошибок КМ1804ВЖ1.

Микросхемы КМ132Р45В 4К×1 бит и КР132РУ12А 4К×1 бит используются для организации буферной статической памяти типа КЭШ. Время выборки у этих микросхем не более 55 нс. Схемы обеспечивают синхронный обмен данными с центральным микропроцессором.

В комплексе СМ 1425 широко применяются схемы электрически программируемой постоянной памяти, программируемых логических матриц и схемы программируемой матричной логики. Встроенные тесты и программа загрузки используют две микросхемы постоянной памяти с ультрафиолетовым стиранием типа К573РФ6 емкостью 8К×8 бит. Элементная база схем интерфейса включает в себя схему приемопередатчика с регистром типа К1804ВА2, а также набор микросхем серии К559. Характерной особенностью ВК СМ 1425 является применение быстродействующих микросхем серии К1531 с малым потреблением.

## СИСТЕМА ЭЛЕКТРОПИТАНИЯ

Электропитание комплекса осуществляется от однофазной сети переменного тока напряжением  $220 \text{ В}_{-15\%}^{+10\%}$  и частотой  $(50 \pm 1) \text{ Гц}$ . Напряжение 220 В подается на базовый блок, видеотерминалы, печатающие устройства. В базовом блоке данное напряжение поступает на источник электропитания В260 и три вентилятора. Оборудование электропитания базового блока обеспечивает: подключение базового блока к сети переменного тока напряжением 220 В с током потребления не более 3,5 А; централизованное включение питания; принудительную вентиляцию всех устройств, размещенных в блоке; определенную последовательность включения и выключения стабилизированных напряжений; возможность подключения цепи централизованного включения питания при расширении комплекса; индикацию наличия напряжения переменного тока 220 В и напряжений постоянного тока; фильтрацию радиопомех.

Все устройства, установленные в базовом блоке, питаются от вторичных напряжений источника В260. Токи, потребляемые отдельными устройствами от источника В260, приведены в табл. 11. Блок В260 вырабатывает также управляющие сигналы, выдаваемые на МПИ комплекса: ВЕРАСL — авария сети питания, ВЕРDCL — авария источника питания. При снятии задней крышки корпуса базового блока обеспечивается доступ к передней панели В260, где расположены ввод сетевого питания, сетевые предохранители, светодиоды «Вкл» и «Авария», шлиц потенциометра для регулировки выходных напряжений, розетка для подключения цепи

Таблица 11

Устройства ВК СМ 1425	Ток, потребляемый от источника В260, А		
	5 В, 40 А	+12 В, 10 А	-12 В, 1Аг
Процессор	6,5	0,02	0,05
СМ 1425.3537	4,5	—	—
СМ 1425.3537.01	5,0	—	—
СМ 1425.3537.02	6,0	—	—
СМ 5508	1	2	—
СМ 5509	1	2	—
СМ 5640	1	1,8	—
СМ 1425.5140	6,5	—	—
СМ 1425.7009	2,5	0,04	0,1
СМ 1425.8540	1,5	0,1	0,15
СМ 1425.8544	1,5	0,15	0,1
СМ 1425.5021	6,5	—	—
СМ 1425.4511	3	—	—

централизованного включения питания при расширении комплекса, шильдик. На задней панели В260 расположены разъемы для разводки выходных напряжений и подключения вентилятора. На пульте управления базового блока находятся переключатель включения-выключения сети «1 — 0», светодиод «Сеть» индикации наличия 220 В, светодиод «Питание» индикации наличия всех номиналов напряжений постоянного тока.

# ПРОЕКТИРОВОЧНОЕ ОБЕСПЕЧЕНИЕ

## Модуль оперативный запоминающий (МОЗ) СМ 1425.3537

Таблица 12

Место установки модуля	Адресное пространство модуля	Адрес регистра состояния
7	00000000 ... 07777777	17772104
6	10000000 ... 17777777	17772106

Предназначен для использования в составе вычислительного комплекса СМ 1425 и имеет два исполнения в зависимости от емкости памяти. Сопряжение устройства с комплексом осуществляется в соответствии с МПИ по ГОСТ 26765.51—86.

Адресное пространство комплекса имеет емкость 4 Мбайта и допускает установку одного или двух МОЗ согласно табл. 12. Причем МОЗ № 0 всегда устанавливается на место 7, а МОЗ № 1 — на место 6 (номера МОЗ присвоены условно). Фактически старшим адресом памяти является 17757777, так как старшие 4 Кбайт являются зоной регистров. Номер МОЗ определяется разрядом адреса [21], а выбор регистра состояния — разрядом адреса [1]. Адрес [21...0] и данные [15...0] пере-

даются по мультиплексированным интерфейсным шинам адреса и данных BDAL [21...0]L. Разводка сигналов МПИ на разъеме приведена в приложении 3.

МОЗ может выполнять передачу данных блоками. Максимальный размер блока 16 слов при условии, что разряды [4...0] начального адреса блока равны нулю. В этом режиме адрес выставляется на шины один раз, а затем передается последовательность данных до 16 слов.

В МОЗ предусмотрен регистр состояния CSR, имеющий три формата согласно табл. 13. В табли-

Таблица 13

Бит CSR	Формат 1	Формат 2	Формат 3	Читается с МПИ	Пишется с МПИ, сбрасывается по INITL	Формируется при ошибке	Отменяется при защите
	CSR [2] = 0		CSR [2] = 1				
	CSR [14] = 0	CSR [14] = 1					
0	Разрешение прерывания при некорректируемой ошибке			+	+	—	—
1	Запрет коррекция одиночных ошибок			+	+	—	+
2	Запрет записи контрольных разрядов. Чтение формата 3			+	+	—	+
3	Номер защищенной зоны			+	+	—	—
4	Признак любой ошибки			+	+	+	—
5	АНО 11	АНО 18	СЛО 0	+	—	+	—
6	АНО 12	АНО 19	СЛО 1	+	—	+	—
7	АНО 13	АНО 20	СЛО 2	+	—	+	—
8	АНО 14	АНО 21	СЛО 3	+	—	+	—
9	АНО 15	0	СЛО 4	+	—	+	—
10	АНО 16	0	СЛО 5	+	—	+	—
11	АНО 17	0	АЛО 19	+	—	+	—
12	0	0	АЛО 20	+	—	+	—
13	Разрешение защиты			+	+	—	—
14	Чтение формата 2			+	+	—	—
15	Признак некорректируемой ошибки			+	+	+	—

Примечание. «+» — реализация типа управления разрядом CSR.

Таблица 14

Код CSR		Биты CSR								Примечание
14	2	12	11	10	9	8	7	6	5	
0	0	0	A17	A16	A15	A14	A13	A12	A11	Номер банка 1 Кслов при некорректируемой ошибке Номер страницы 128 Кслов при некорректируемой ошибке Номер линейки и синдром при любой ошибке
1	0	0	0	0	0	A21	A20	A19	A18	
X	1	A20	A19	C5	C4	C3	C2	C1	C0	

це приняты следующие обозначения: АНО — адрес некорректируемой ошибки, АЛО — адрес любой ошибки, СЛО — синдром любой ошибки. Для выбора формата необходимо записать соответствующий код в разряды [2, 14] регистра CSR согласно табл. 14. При установке в единицу бита CSR [2] в операциях записи отменяется запись контрольных разрядов, т. е. записываются информационные разряды, а контрольные сохраняют свое предыдущее состояние. При установке в единицу бита CSR [1] отменяется коррекция одиночных ошибок.

Предусмотрена возможность отмены действий бит [1, 2] CSR в зоне защиты. Выбор зоны защиты определяется кодом CSR [13, 3], коррекция ошибок выполняется согласно табл. 15, а запись контрольных разрядов — согласно табл. 16. Расположение зон приведено в табл. 17, причем разряд адреса [21] определяет номер модуля, а разряд [20] — номер зоны. Объем зоны защиты 512 Кбайт.

Таблица 15

Код CSR			Режим работы МОЗ
13	3	1	
X	X	0	Разрешение коррекции по всему объему
0	X	1	Запрет коррекции по всему объему
1	0	1	Запрет коррекции по всему объему, кроме первых 512 Кбайт
1	1	1	Запрет коррекции по всему объему, кроме вторых 512 Кбайт

Таблица 16

Код CSR			Режим работы МОЗ
13	3	2	
X	X	0	Разрешение записи контрольных разрядов по всему объему
0	X	1	Запрет записи контрольных разрядов по всему объему
1	0	1	Запрет записи контрольных разрядов по всему объему, кроме первых 512 Кбайт
1	1	1	Запрет записи контрольных разрядов по всему объему, кроме вторых 512 Кбайт

Таблица 17

Место	Зона защиты	Адрес
7	Первые 512 Кбайт	00000000...01777777
	Вторые 512 Кбайт	02000000...03777777
6	Первые 512 Кбайт	10000000...11777777
	Вторые 512 Кбайт	12000000...13777777

В регистре состояния хранятся признаки одиночной и двойной ошибки, а также код синдрома и адрес ошибки. На шину BDAL[17]L выдается сигнал разрешения прерывания по ошибке, а на шину BDAL[16]L — сигнал некорректируемой ошибки.

МОЗ транслирует сигналы BIAKIL — BIAKOL, BDMGIL — BDMGOL.

В состав модуля СМ 1425.3537 (рис. 19) входят следующие функциональные узлы: узел управления

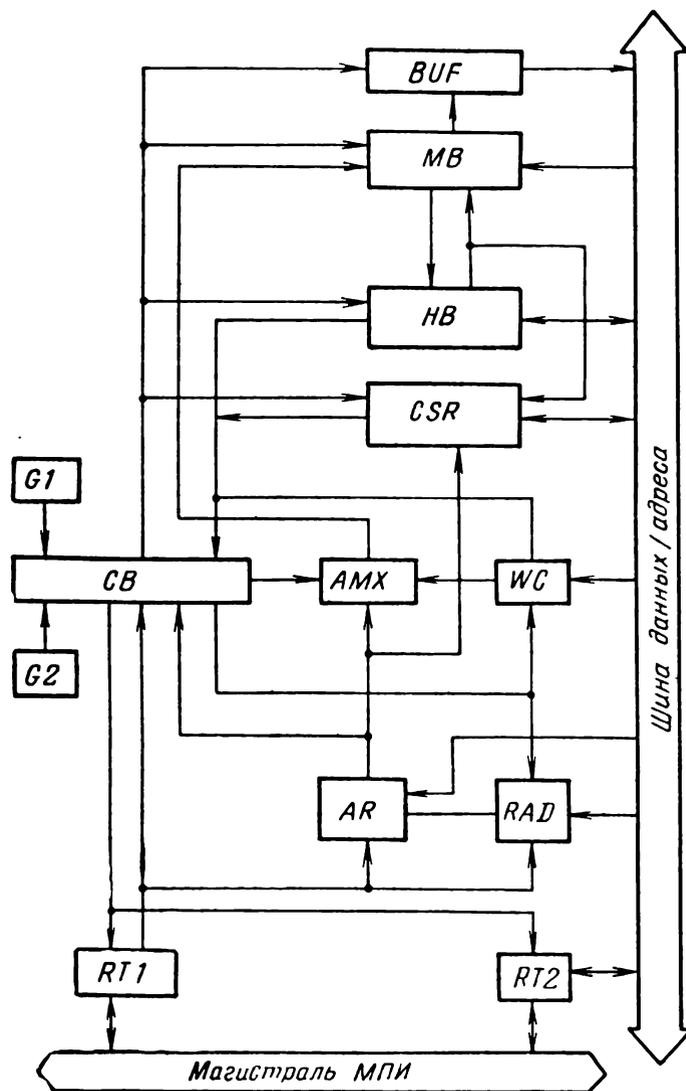


Рис. 19. Структурная схема МОЗ

CB, узел хранения MB, мультиплексор адреса AMX, регистр адреса AR, счетчик слов WC, дешифратор адреса RAD, узел контроля HB, регистр состояния CSR, узлы приемопередатчиков RT1 и RT2, буфер считанных данных BUF, кварцевый генератор G1, мультивибратор G2. Узел CB выдает управляющие сигналы в остальные узлы модуля, управляет внутренними потоками информации, обеспечивает периодическую регенерацию запоминающих ИМС. Узел хранения MB содержит 88 запоминающих ИМС K565PY7 емкостью 256 Кбит каждая, размещенных в четырех линейках по 22 разряда, из которых 16 информационных и 6 контрольных. Мультиплексор адреса AMX преобразует 18-разрядный параллельный код адреса в 9-разрядный мультиплексированный код адреса — две посылки по 9 разрядов: адрес строки и адрес столбца. Регистр AR предназначен для хранения адреса, принятого с интерфейса. Счетчик слов WC формирует адреса слов в режиме блочной передачи данных. Дешифратор адреса RAD служит для определения адреса регистра состояния согласно табл. 12. Узел контроля HB обеспечивает формирование контрольных разрядов по коду Хэмминга при записи данных

в узел хранения, а также коррекцию одиночных и обнаружение двойных ошибок при чтении данных из узла МВ. Формирование контрольных разрядов приведено в табл. 18, расшифровка синдромов при чтении — в табл. 19. В таблицах приняты следующие

Таблица 18

Разряд данных	Функция					
	Исключающее «ИЛИ»		Исключающее «ИЛИ-НЕ»		Исключающее «ИЛИ»	
	C5	C4	C3	C2	C1	C0
0	—	—	X	X	X	—
1	—	—	X	—	X	X
2	—	X	—	—	X	X
3	—	X	—	X	—	X
4	—	X	—	X	X	—
5	—	X	X	—	—	X
6	—	X	X	—	X	—
7	—	X	X	X	—	—
8	X	—	—	—	X	X
9	X	—	—	X	—	X
10	X	—	—	X	X	—
11	X	—	X	—	—	X
12	X	—	X	—	X	—
13	X	—	X	X	—	—
14	X	X	—	—	—	X
15	X	X	—	X	—	—

Таблица 19

Старшие разряды синдрома	Младшие разряды синдрома			S0	S1	S2	Тип ошибки													
	S5	S4	S3																	
0	0	0	0	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	
0	0	0	1	0	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1
0	1	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1
0	1	1	1	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1
1	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1
1	0	1	1	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1
1	1	0	1	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1
1	1	1	1	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1

щие условные обозначения: X — входит в функцию, \* — нет ошибок, — цифра — место одиночной ошибки, D — двойная ошибка, M — три и более ошибок. Регистр состояния CSR служит для диагностирования неисправностей узлов модуля и позволяет выполнять такие диагностические операции, как запрет записи контрольных разрядов, запрет коррекции одиночных ошибок, отмена запретов в защищенной зоне, разрешение прерывания по ошибке, чтение адреса некорректируемой ошибки, чтение адреса и синдрома любой ошибки. Узел передатчика RT2 выполняет прием адреса и обмен данными, а узел RT1 — обмен управляющими сигналами МОЗ с интерфейсом. Кварцевый генератор G1 работает на частоте 20 МГц и обеспечивает син-

хронизацию работы узла управления СВ. Мульти-вibrator G2 работает на частоте около 125 кГц и вырабатывает запросы на регенерацию запоминающих ИМС. Потенциал на входе X2:C17 определяет номер МОЗ и его адресное пространство согласно табл. 12. Вход X2:C16 не используется.

ТЕХНИЧЕСКИЕ ДАННЫЕ

Информационная емкость, Мбайт:	
СМ 1425.3537 . . . . .	0,5
СМ 1425.3537.02 . . . . .	2
Разрядность ячейки памяти, бит . . . . .	16+6 контрольных
Операции, выполняемые модулем . . . . .	чтение слова (ЧТС), запись слова (ЗПС), чтение — модификация — запись, запись байта, блочная передача (блочная запись, блочное чтение)
Цикл обращения при операциях ЧТС, ЗПС без учета времени обмена по шине и при отсутствии регенерации, мкс . . . . .	не более 0,7
Время выборки при операции ЧТС от сигнала BDINL до сигнала BRPLYL при отсутствии регенерации, мкс . . . . .	не более 0,5
Питание от источника постоянного тока напряжением, В . . . . .	5±0,25
Потребляемый ток, А . . . . .	7
Габаритные размеры, мм . . . . .	248×240×16
Масса, кг . . . . .	0,7

Контроллер НМД и НГМД  
СМ 1425.5140

Предназначен для управления малогабаритными внешними запоминающими устройствами (ВЗУ) на гибких и несменных жестких магнитных дисках (НГМД типа СМ 5640 и НМД типов СМ 5508 и СМ 5509) с диаметром носителей 133 мм, используемых в составе микроЭВМ или ВК, системная магистраль которых обеспечивает функциональную и конструктивную совместимость с СМ 1425. Контроллер имеет регистр адреса и состояния РАС — 772152 и регистр синхронизации обмена РСО — 772150, а также встроенные средства самодиагностики и визуальное отображение ее результатов. Обмен данными осуществляется в режиме прямого доступа к памяти 16-разрядными словами. Обеспечена возможность изменения логических адресов ВЗУ с помощью перемычек.

Контроллер является микропрограммно управляемым устройством с пакетно-ориентированным протоколом обмена. Выполнен на базе однокристалльного 16-разрядного микропроцессора КР1807ВМ1. Функционирование контроллера обеспечивается программой-монитором и микропрограммами блока микропрограммного управления (БМУ). Монитор записан в РПЗУ, а микропрограммы БМУ — в быстродействующих ППЗУ. Монитор осуществляет управление основными функциональными блоками контроллера и реализует часть протокола пакетного обмена между контроллером и ЭВМ. Под пакетом здесь следует понимать информацию определенной структуры, заключающую в себе сведения о содержании и результате обмена. Все операции, связанные с циклами обмена через магистраль МПИ или с чтением/записью информа-

ции на носитель, реализуются с помощью микропрограмм БМУ, которые в свою очередь подчинены монитору.

Пакетный обмен происходит в режиме прямого доступа к памяти. В пакетно-ориентированном протоколе обмена используются три типа пакетов: пакет команд, пакет ответов, пакет коммуникаций. Пакет команд содержит исходную информацию о команде и других параметрах обмена, пакет ответов — информацию о состоянии ВЗУ, ошибках и признаках обмена. Адреса пакетов команд и пакетов ответов описываются в пакете коммуникаций, адрес которого сообщается контроллеру в начальном диалоге через регистр РАС. Пакеты формируются в буферном ОЗУ под управлением монитора контроллера и в ЭВМ под управлением управляющих программ.

Контроллер состоит из следующих основных узлов (рис. 20): управляющего микропроцессора

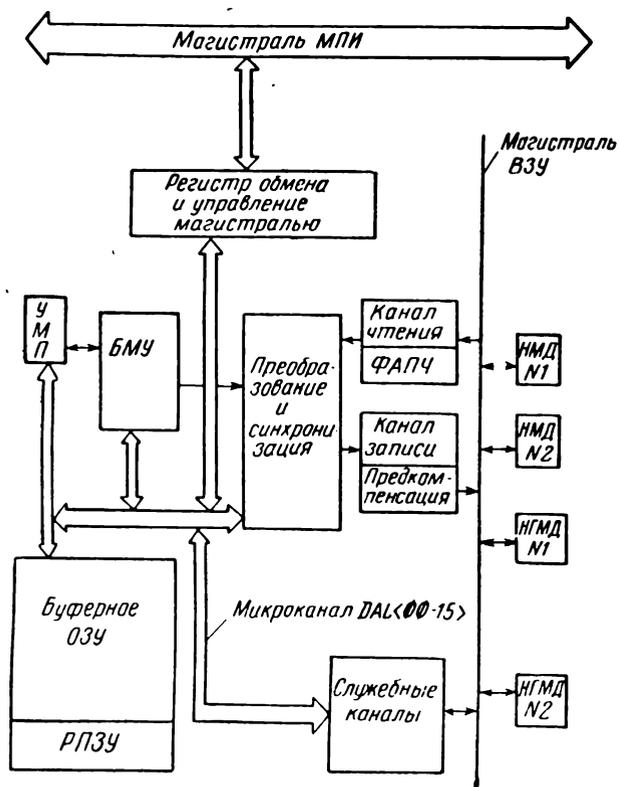


Рис. 20. Структурная схема контроллера SM 1425.5140 (ФАПЧ — схема фазовой подстройки частоты)

(УМП) со схемой внутренних прерываний и прямого доступа, предназначенного для управления всеми функциональными блоками контроллера согласно логике монитора; буферного ОЗУ и РПЗУ со схемами стробирования и адресации, предназначенных для хранения команд монитора, параметров обмена, программных регистров управления и состояния; регистров обмена и приемопередатчиков, включающих в себя регистры чтения/записи РАС и РСО, регистр вектора, буфер обмена данными; БМУ, содержащего высокоскоростное регистровое ПЗУ с мультиплексорами ветвлений и многофункциональные схемы прямого доступа (МСПД); служебных регистров и схемы управления магистралью ВЗУ; канала чтения со схемами выделения информации и схемой фазовой автоподстройки частоты; канала записи со схемами синхронизации и предкомпенсации; схем параллельно-

последовательного и последовательно-параллельного преобразования.

Протокол обмена осуществляется следующим образом. Для того чтобы начать работу с контроллером, необходимо выполнить процедуру инициализации. Целью инициализации является: сообщение контроллеру параметров расположенных в области коммуникации ЭВМ, сообщение ЭВМ параметров контроллера, проверка работоспособности контроллера и, если последняя имеет место, перевод контроллера в состояние «Включено». В результате загрузки произвольного кода в регистр РСО (772150) контроллер стартует и засылает в регистр РАС (чтение, 772152) слово исходного состояния. Далее следует диалог из четырех управляющих слов, которые записываются ЭВМ в регистр РАС (запись). На эти слова контроллер должен сформировать через регистр РАС (чтение) соответствующие ответы. В процессе начального диалога контроллер осуществляет тестирование области пакета коммуникаций и внутреннего буферного ОЗУ. В случае ошибок он выходит из диалога и заносит код ошибки в регистр РАС. После успешного завершения диалога регистр РАС обнулен.

По окончании начального диалога необходимо подготовить пакет коммуникаций (рис. 21), в

15	8	7	0	
Прерывание по команде				
Прерывание по ответу				
Виртуальный адрес пакета ответов 1				Область N описателей пакетов ответов
Флаг обмена пакетом	Физическое расширение адреса пакета			
Виртуальный адрес пакета ответов 2				
Флаг обмена пакетом	Физическое расширение адреса пакета			
⋮	⋮	⋮	⋮	
Виртуальный адрес пакета ответов N				Область N описателей пакетов команд
Флаг обмена пакетом	Физическое расширение адреса пакета			
Виртуальный адрес пакета команд 1				
Флаг обмена пакетом	Физическое расширение адреса пакета			
⋮	⋮	⋮	⋮	
Виртуальный адрес пакета команд N				
Флаг обмена пакетом	Физическое расширение адреса пакета			

Рис. 21. Структура пакета коммуникаций

структуре которого содержится информация об адресах пакетов команд, пакетов ответов и флагах обмена каждым пакетом. Пакеты команд и ответов резервируются для каждого виртуального пользователя при организации мультипрограммного режима доступа к ВЗУ. Каждому пакету соответствует двухсловный описатель. Первое слово описателя содержит виртуальный адрес пакета, второе — флаг пакета и физическое расширение адреса. После заполнения пакета коммуникаций необходимо

заполнить пакет команд и подготовить буфер для пакета ответов. Сигналом готовности пакета коммуникаций и пакета команд со стороны ЭВМ служит сигнал чтения регистра синхронизации обмена РСО. После этого сигнала контроллер читает параметры команды, которую ему необходимо выполнить, выполняет команду и заполняет буфер ответов. При обмене контроллер устанавливает два флага: флаг обмена пакетом команд и флаг обмена пакетом ответа. Если задать значение вектора, не равное нулю, то имеется возможность инициализировать прерывание по завершению обмена пакетами. После завершения обмена контроллеру можно передавать очередной пакет команд и сигнал синхронизации; при этом должны быть установлены флаги обменов в пакете коммуникаций. Для приведения контроллера в исходное состояние необходимо провести операцию записи в регистр синхронизации РСО. При чтении и записи в РСО передачи информации не происходит. Перечень команд контроллера приведен в табл. 20.

Таблица 20

Код команды (восьмеричный)	Функция
1	Отвержение. Команда завершается по истечении периода синхронизации
20	Выборка. Чтение данных из выбранного устройства в буфер данных
4	Установить характеристики контроллера
12	Установить характеристики устройства
13	Определить выбор пути
24	Произвести замену
41	Чтение. Чтение адресованного логического блока выбранного устройства во внешнюю память
42	Запись. Запись информации из внешней памяти в адресованный логический блок выбранного устройства
11	Включение. Логическое включение выбранного устройства
10	Выключение. Перевод выбранного устройства в свободное состояние
22	Очистка. Запись нулей в адресованный блок выбранного устройства
40	Сравнение с внешними данными. Чтение данных с выбранного устройства и сравнение с данными внешнего ОЗУ
2	Получить статус команды. Сообщение статуса выбранной команды возвратом числа, отображающего выполнение команды
3	Получить статус устройства. Сообщение статуса выбранного устройства

Критерием правильности выполнения команды служит равенство нулю регистра состояния обмена в пакете ответов и сохранение нулевого значения в регистре РАС. Коды регистра состояния содержат информацию о состоянии устройства, ошибках выполнения команды и приведены в табл. 21. Коды ошибок и отказов контроллера, возникающих при выполнении команд, заносятся в регистр РАС.

Основной набор команд задается идентификатором, равным нулю. При идентификаторе команд, равном 2, контроллер выполняет дополнительные и диагностические программы. Некоторые диагности-

Код байта (восьмеричный)	Состояние завершения обмена
0	Нормальное окончание
1	Неверная команда
2	Команда отвергнута
3	Выбранное устройство находится в автономном режиме
4	Устройство свободно
5	Ошибка формата
6	Защита записи
7	Ошибка сравнения
10	Ошибка данных
11	Ошибка обращения к внешнему ОЗУ
12	Ошибка контроллера
13	Ошибка устройства
37	Сообщение внутренней самодиагностики

ческие программы выбирают диагностический блок на НМД. Это позволяет упрощать диагностику чтения и записи без использования специальных тестовых программ. В состав диагностических и дополнительных программ включены программа записи и передачи диагностического блока (только НМД), программа чтения диагностического блока (только НМД), программа форматирования, сравнения и исключения плохих блоков (только НМД), программа чтения сектора без передачи данных в магистраль, программа позиционирования. Список дополнительных и диагностических команд приведен в табл. 22.

Таблица 22

Код команды	Содержание команды
1	Установить статус дополнительных и диагностических программ
2	Выполнить программы обеспечения
3	Выполнить локальные программы (дополнительные и диагностические)
4	Принять данные
5	Передать данные
6	Прекратить выполнение

Схема подключения контроллера к ВЗУ приведена на рис. 22.

#### ТЕХНИЧЕСКИЕ ДАННЫЕ

Принцип работы контроллера . . . . .	микропрограммный
Число подключаемых ВЗУ . . . . .	более 4 (в том числе не более 2 НМД)
Емкость, Кбайт:	
буферного ОЗУ . . . . .	4
РПЗУ . . . . .	32
ППЗУ . . . . .	9
Метод записи информации на НМД и НГМД . . . . .	модифицированная фазовая модуляция

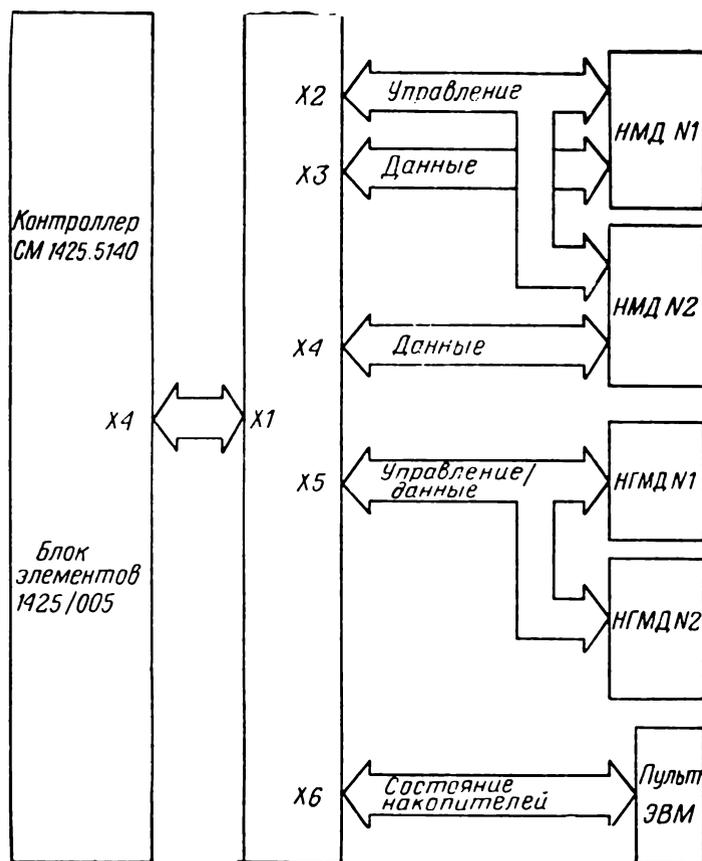


Рис. 22. Схема подключения контроллера к ВЗУ

Скорость обмена, Кбайт/с:	
для НГМД . . . . .	31,25
для НМД . . . . .	625
Адрес вектора прерывания . . . . .	154
Разрядность данных обмена в режиме прямого доступа, бит . . . . .	16
Питание от источника постоянного тока напряжением, В . . . . .	5; 12
Ток, потребляемый от источника, А:	
5 В . . . . .	7
12 В . . . . .	0,35
Габаритные размеры, мм . . . . .	250×242×14
Масса, кг . . . . .	1,5

## Мультиплексоры передачи данных СМ 1425.8540 и СМ 1425.8544

Предназначены для работы в составе ВК СМ 1425, обеспечивают подключение к комплексу видеотерминалов, а также сопряжение двух вычислительных комплексов СМ 1425 через каналы передачи данных. Обмен информацией с ВК СМ 1425 обеспечивается в соответствии с требованиями МПИ, прием и передача информации по четырем независимым каналам передачи данных — через стык С2 или ИРПС (СМ 1425.8544). Программируемыми параметрами передачи являются: формат символа; контроль данных на четность, нечетность или без контроля; скорость передачи данных; длина стоповой посылки. Функционирование мультиплексора определяется программируемыми регистрами, перечень которых приведен в табл. 23.

Регистр CSR предназначен для задания режимов работы мультиплексора и управления прерыванием.

Обозначение	Наименование	Базовый адрес	Операция с регистром
CSR	Регистр управления и состояния	760100	Чтение/запись
RBUF	Регистр принимаемых данных	760102	Чтение
LPR	Регистр параметров линий	760102	Запись
TCR	Регистр управления передатчиками	760104	Чтение/запись
MSR	Регистр состояния модемов	760106	Чтение
TDR	Регистр передаваемых данных	760106	Запись

ванием. Содержимое разрядов регистра следующее:

0...2, 10, 11 — не используются;

3 — контроль. Служит для организации внутреннего шлейфа цепей С103, С104. Записывается и читается;

4 — селективный сброс. Предназначен для программного сброса мультиплексора, кроме регистра MSR и старшего байта регистра TDR. В регистре RBUF сбрасывается только RBUF[15]. Записывается и читается;

5 — разрешение сканирования; записывается и читается;

6 — разрешение прерывания приемника; записывается и читается;

7 — прерывание приемника. Устанавливается, когда символ доступен чтению из RBUF. Только читается;

8, 9 — номер передатчика, вызвавшего прерывание. Только читается;

12 — разрешение прерывания по уровню тревожной сигнализации. Разрешает прерывание по CSR[13], если разрешено прерывание приемника CSR[6]=1. Записывается и читается;

13 — прерывание по уровню тревожной сигнализации. Вырабатывается после приема 16 символов. Только читается;

14 — разрешение прерывания передатчика; записывается и читается;

15 — прерывание передатчика. Устанавливается, если обнаружен канал передачи данных, у которого разрешена передача регистром TCR и есть пустой буферный регистр передатчика. Только читается.

Регистр RBUF предназначен для хранения принимаемых данных и информации о них. Является выходным регистром буферной памяти принимаемых данных. Только читается. Содержимое разрядов следующее:

7...0 — код символа принимаемых данных;

9, 8 — номер канала передачи данных, от которого получен символ;

10, 11 — не используются;

12 — ошибка по паритету. Устанавливается, если разрешен контроль по паритету и получен бит с неправильным паритетом;

13 — ошибка формата. Устанавливается, если получен символ с неправильной полярностью стоповой посылки;

14 — потеря символа (переполнение). Устанавливается, если один или больше символов были

потерины в канале, по которому принят символ, из-за того, что буферная память принимаемых данных переполнена;

15—данные действительны. Устанавливается, если символ, находящийся в RBUF, действителен.

Регистр LPR предназначен для загрузки параметров передачи. Только записывается. Содержимое разрядов следующее:

1, 0—номер канала передачи данных, к которому относятся загружаемые параметры;

2, 13, 14, 15—не используются;

4, 3—определяют формат символа: 00—5 бит, 01—6 бит, 10—7 бит, 11—8 бит;

5—длина стоповой посылки: 0—однократная стоповая посылка, 1—двухкратная стоповая посылка;

6—разрешение контроля по паритету;

7—контроль: 0—на четность; 1—на нечетность;

11...8—определяют код скорости передачи данных, бит/с:

0000—50; 0001—75; 0010—100; 0011—200; 0100—150; 0101—300; 0110—600; 0111—1200; 1000—1200\*; 1001—2400\*; 1010—2400; 1011—4800\*; 1100—4800; 1101—9600\*; 1110—9600; 1111—19200.

Скорости, отмеченные «\*», при программировании не используются;

12—разрешение работы приемника.

Регистр TCR предназначен для управления передачей. Записывается и читается. Содержимое разрядов следующее:

3...0—регистр разрешения передачи. Каждый бит соответствует каналу с тем же номером;

4...7, 12...15—не используются;

11...8—регистр управления цепями C108 стыка C2. Соответствие между разрядом и номером канала: 8—0-й канал, 9—1-й канал, 10—2-й канал, 11—3-й канал.

Регистр MSR предназначен для анализа состояния цепей C109 и C125 стыка C2. Только читается. Содержимое разрядов следующее:

3...0—регистр состояния цепей C125 стыка C2. Каждый бит соответствует каналу с тем же номером;

4...7, 12...15—не используются;

11...8—регистр состояния цепей C109 стыка C2. Соответствие между разрядом и номером канала: 8—0-й канал, 9—1-й канал, 10—2-й канал, 11—3-й канал;

Регистр TDR предназначен для временного хранения передаваемых данных и управления цепями C103. Только записывается. Содержимое разрядов следующее:

7...0—буферный регистр передатчика. Загружается символом, который нужно передать в канал;

11...8—регистр управления цепями C103. Соответствие разрядов номеру канала: 8—0-й канал, 9—1-й канал, 10—2-й канал, 11—3-й канал;

15...12—не используются.

В мультиплексоре применена многоуровневая позиционно независимая система прерывания с фиксированным уровнем прерывания 5. Количество векторов прерывания 2—по вводу и выводу. Адреса регистров и вектора прерывания по вводу задаются перемычками. Перемычки размещены на колодках БЭ мультиплексора. Адрес вектора прерывания по выводу формируется увеличением адреса вектора прерывания по вводу на 4. Состав сигналов стыка C2, используемых в устройстве, приведен в табл. 24.

Таблица 24

Номер цепи стыка	Наименование цепи
102	Сигнальное заземление или общий обратный провод
102 (А)	Общий обратный провод оконечного оборудования данных (ООД)
102 (Б)	Общий обратный провод аппаратуры передачи данных
103	Передаваемые данные
104	Принимаемые данные
108.2	Оконечное оборудование данных готово
109	Детектор принимаемого линейного сигнала канала данных
125	Индикатор вызова

### ТЕХНИЧЕСКИЕ ДАННЫЕ

Система кодирования информации . . . . .	двоичная
Формат символа, бит . . . . .	5; 6; 7; 8
Единица данных . . . . .	бит, байт, слово (два байта)
Способ передачи цифрового сигнала данных . . . . .	последовательный, асинхронный, стартстопный
Скорость передачи данных, бит/с . . . . .	50; 75; 100; 150; 200; 300; 600; 1200; 2400; 4800; 9600; 19200
Режим работы . . . . .	одновременный, двухсторонний
Емкость буферной памяти принимаемых данных, байт . . . . .	128
Питание от источника постоянного тока напряжением, В . . . . .	+5, +12, -12
Ток, потребляемый от источника, А:	
+5 В . . . . .	1,6
+12 В . . . . .	0,1
-12 В . . . . .	0,15
Потребляемая мультиплексором мощность, Вт . . . . .	15
Габаритные размеры БЭ, мм . . . . .	250×245×16
Масса, кг . . . . .	0,7

## Контроллер магнитной ленты СМ 1425.5021

Предназначен для обеспечения управления работой накопителей на магнитных лентах потокового типа (НМЛ-П) в системах обработки информации, построенных на базе ВК СМ 1425. Для подключения контроллера к комплексу используется МПИ, для подключения накопителей НМЛ-П к контроллеру—интерфейс для малогабаритных накопителей на магнитной ленте ИНМЛ-П. Контроллер обеспечивает переключение метода записи БВН1 с плотностью записи в НМЛ-П 32 бит/мм и метода записи ФК с плотностью записи 63 бит/мм; переключение режимов работы НМЛ-П—стартстопный и потоковый; максимальную скорость обмена 160 Кбайт/с; номиналь-

ную емкость на один НМЛ-П 40 Мбайт. Связь контроллера с вычислительным комплексом осуществляется через разъемы X1 и X2 БЭ СМ 1425/007, связь контроллера с накопителем — через разъемы X3 и X4 БЭ СМ 1425/007.

Контроллер позволяет выполнять следующие функции: дешифрацию адреса НМЛ-П и логическое подключение его к ВК СМ 1425, а также логическое отключение НМЛ-П после завершения операции и по команде от процессора; адресацию программно-доступных регистров; синхронизацию пересылок информации; передачу в процессор информации о состоянии контроллера и НМЛ-П; прием и передачу информации НМЛ-П; контроль считанной информации при выполнении команд записи и чтения; управление работой НМЛ; дешифрацию кодов команд ввода-вывода, принятых из процессора; прием и передачу информации в режиме прямого доступа. В контроллере имеются два программно-доступных регистра: регистр адреса шины/буфера данных TSBA/TSDB и регистр состояния TSSR. TSBA только читаемый регистр, содержащий адрес последнего слова, считываемого из памяти вычислительного комплекса или записываемого в его память. TSDB предназначен только для записи и не очищается при сбросе от МПИ, обеспечивает контроллер начальным адресом пакета команд и может быть загружен побайтно. Побайтная загрузка TSDB используется в диагностических целях. TSSR содержит информацию о состоянии контроллера после завершения каждой команды, а также биты 16 и 17 адреса МПИ. Запись по адресу TSSR в старший байт рассматривается как запись в расширенный регистр буфера данных TSDBX. Запись в TSDBX возможна при установленном в 1 бите TSSR [7].

Базовый адрес контроллера задается переключками узла 7. Значение адреса контроллера может быть изменено с помощью переключек узла 7 S1.1, S1.2 на колодке S1. Базовый вариант контроллера имеет 22-разрядный адрес. Переход к 18-разрядному адресу осуществляется установкой соответствующей переключки S8.4 узла 18 на колодке S8. Имеется до четырех логических номеров устройств, которым соответствуют определенные узлом 24 физические номера накопителей и режим работы накопителей. Адреса программно-доступных регистров в зависимости от логического номера устройства и установки переключек S1.1, S1.2 приведены в табл. 25. Количество подключаемых логических устройств ограничивается переключками на колодке S1 узла 7 в соответствии с табл. 26.

Соответствие между номерами логических устройств, номерами подключенных накопителей, плотностью записи и режимом работы накопителей задается состоянием переключек S4, S5, S6 и коммутацией контактов колодок S4, S5, S6 между собой. Номера логических устройств 0, 1, 2, 3 соответствуют номерам накопителей при установленных переключках S4.1, S4.2, S5.1 и отсутствии коммутации между контактами колодок. При этом задается также старт-стопный режим работы для всех накопителей. Базовый вариант коммутации соответствует табл. 27.

При работе с накопителем СМ 5316 подключается только один накопитель, что обусловлено его конструкцией.

Таблица 25

Адреса регистров		Логический номер устройства	Наличие переключки	
TSBA/TSDA	TSSR		S1.2	S1.1
772520	772522	0	Есть	Есть
772524	772526	1		
772530	772532	2		
772534	772536	3		
772720	772722	0	Нет	»
772724	772726	1		
772730	772732	2		
772734	772736	3		
777360	777362	0	Есть	Нет
777364	777366	1		
777370	777372	2		
777374	777376	3		
777420	777422	0	Нет	»
777424	777426	1		
777430	777432	2		
777434	777436	3		

Таблица 26

Количество устройств	Логические номера устройств	Наличие переключки	
		S1.3	S1.4
1	00	Есть	Есть
2	00	»	Нет
3	01		
	00	Нет	Есть
4	01		
	10		
	00	»	Нет
	01		
	10		
	11		

Таблица 27

Логический номер устройства	Номер накопителя	Плотность записи, бит/мм	Режим работы
0	0	32	Старт-стопный
1	0	63	Потоковый
2	1	32	Старт-стопный
3	1	63	Потоковый

Уровень приоритета контроллера на разрешение прерывания процессора задается переключками S2.1, S2.2, S3.1 на колодках S2, S3 узла 4. Базовый уровень приоритета 5; для изменения уровня приоритета следует пользоваться табл. 28.

Таблица 28

Уровень приоритета	Наличие перемычек		
	S3.1	S2.2	S2.1
4	Есть	Есть	Есть
5	Нет	»	»
6	Есть	Нет	»
7	»	»	Нет

Базовый адрес вектора прерывания контроллера равен 224. Адрес вектора прерывания логического устройства 1 задается перемычками S7.1...S7.4, S8.1...S8.3 на колодках S7, S8 узла 18 согласно табл. 29. При этом соответствующий бит адреса устанавливается в единицу при отсутствии перемычки или в нуль при ее наличии. Для логических устройств, имеющих номера 2 и 3, адрес вектора прерывания увеличивается на 4 и 8 соответственно по отношению к определенному перемычками согласно табл. 29.

Таблица 29

Бит адреса	8	7	6	5	4	3	2
Перемычка	7.1	7.2	7.3	7.4	8.1	8.2	8.3

При включении электропитания в контроллере выполняется микропрограмма самодиагностики, после завершения которой контроллер готов к работе. Светится индикатор БЭ СМ 1425/007. Команды, данные и состояние передаются из памяти ВК СМ 1425 в контроллер и из контроллера в память ВК группами байтов, называемыми пакетами (буферами). Имеются четыре типа пакетов: пакет команд, пакет данных, пакет характеристик, пакет сообщений (или конечный пакет). Кроме того, пакеты подразделяются на пакеты (буферы) данных и пакеты (буферы) управления. Каждый из буферов может управляться или процессором, или контроллером. Содержимое буфера и источника информации представлены в табл. 30.

Таблица 30

	Буфер		Размер	Источник
	данных	управления		
—	Команда	—	8 байт	Процессор
Данные с ленты	—	—	1 байт...65 Кбайт	Контроллер
Данные на ленту	—	—	1 байт...65 Кбайт	Процессор
—	Характеристики	—	6...10 байт	»
—	Сообщение	—	14...16 байт	Контроллер

Передача данных и пакетов управления осуществляется по прямому доступу.

Пакет команды представляет собой, как правило, четыре 16-разрядных слова, хотя не все команды используют все четыре слова пакета. Первое слово в таких пакетах называется словом

заголовка. Пакет команд из двух слов состоит из слова команды и счетчика. По этой команде пропускаются зоны в направлении движения (вперед) или в обратном направлении (назад), а также маркеры группы зон (вперед или назад), или же осуществляется перематка ленты в начало.

В пакете команды «Установить характеристики» второе и третье слова содержат адрес буфера характеристики. Это обязательно четный адрес в оперативной памяти ВК СМ 1425. Четвертое слово пакета данной команды указывает количество байтов буфера характеристики (размер буфера): 6, 8 или 10 байт. Кроме того, команда «Установить характеристики» имеет второй пакет — пакет характеристики длиной четыре или пять слов. Первые два слова пакета характеристики содержат адрес пакета в ОП СМ 1425. Это обязательно четный адрес. Третье слово указывает количество байтов в буфере сообщения (размер буфера) — 14 или 16 байт.

Пакет сообщения формируется в контроллере и выдается в конце выполнения команды или по прерыванию «Внимание». Пакет сообщения состоит из слова заголовка (слово 1), слова длины поля данных (слово 2), слова счетчика байтов (зон) маркеров (слово 3) и четырех или пяти слов регистров расширенного состояния.

Обычная последовательность выполнения команды чтения или записи выглядит следующим образом:

процессор читает регистр состояния TSSR;

процессор загружает в регистр буфера данных TSDB начальный адрес пакета команды «Установить характеристики»;

контроллер выбирает пакет команды «Установить характеристики», содержащий от трех до пяти последовательных ячеек памяти (6...10 байт). Основной целью выполнения этой команды является получение начального адреса буфера сообщения. После завершения операции чтения или записи состояние загружается в буфер сообщения контроллером;

процессор загружает в регистр буфера данных TSDB начальный адрес пакета команды чтения или записи;

контроллер выбирает пакет команды;

контроллер как затчик начинает передачу данных между оперативной памятью ВК СМ 1425 и выбранным накопителем;

чтение или запись данных продолжается до тех пор, пока не будет исчерпан счетчик данных или не будет обнаружен конец зоны (при чтении);

информация о состоянии загружается в регистр TSSR контроллера и пакет сообщения, определенный последней командой «Установить характеристики»;

если требуется, контроллер вызывает прерывание как сигнал о конце команды.

Пакет команды и пакет сообщения в каждый момент времени доступны и могут использоваться или процессором, или контроллером. При нормальной обработке команды использование обоих пакетов передается одновременно: сначала процессором контроллеру (в начале обработки команды, когда процессор записывает адрес пакета команд в регистр TSDB; бит ACK в заголовке пакета команд устанавливается в единицу), а затем контроллером процессору (при завершении команды, когда контроллер устанавливает TSSR [7] в еди-

ницу; бит АСК в заголовке пакета сообщений устанавливается в единицу, кроме случая прерывания по «Вниманию»).

Таблица 31

**ТЕХНИЧЕСКИЕ ДАННЫЕ**

Базовый адрес вектора прерывания . . .	224
Уровень приоритета . . . . .	4...7
Передача данных . . . . .	3ПД
Количество адресуемых регистров . . . . .	2
Количество бит информации в слове . . .	16
Базовый адрес в восьмеричной системе для 22-разрядного адреса . . . . .	17772520
Количество подключаемых накопителей . .	2
Питание от источника постоянного тока напряжением, В . . . . .	5
Потребляемая мощность, В·А . . . . .	40
Габаритные размеры, мм . . . . .	250×242×14
Масса, кг . . . . .	1,0

**Контроллер интерфейсов групповой СМ 1425.7009**

Предназначен для работы в составе ВК СМ 1425 — подключения внешних устройств типа алфавитно-цифровые видеотерминалы, печатающие и другие аналогичные устройства, а также для сопряжения двух ВК СМ 1425. Контроллер обеспечивает асинхронную передачу данных между МПИ и каналами передачи данных (ПД). Функции контроллера определяются входящими в его состав четырьмя автономными блоками управления: адаптерами А1.1 ... А1.4, имеющими общий выход на МПИ и подключающими к ВК СМ 1425 каналы ПД.

Контроллер сопрягает ВК СМ 1425 с четырьмя каналами ПД:

- с первым каналом ПД через стык С2 с цепями модемного управления (адаптер А1.1);
- со вторым каналом ПД через стык С2 без цепей модемного управления (адаптер А1.2);
- с третьим каналом ПД через интерфейс ИРПР с 8-разрядной шиной данных (адаптер А1.3);
- с четвертым каналом ПД через интерфейс ИРПР с 16-разрядной шиной данных (адаптер А1.4).

Допускается установка нескольких контроллеров в один ВК СМ 1425. Каждому контроллеру с помощью перемычек на колодках S1, S6 присваивается индивидуальный номер, при этом каждый адаптер контроллера имеет свой базовый адрес регистров и адреса вектора прерывания (табл. 31).

Адаптеры А1.1 и А1.2 выполняют преобразование данных, принимаемых из МПИ в виде параллельных байтов, в последовательные старт-стопные символы, выдаваемые в канал ПД, и обратное преобразование. Содержат по четыре 16-разрядных программно-управляемых регистра (табл. 32). Адрес регистра определяется как сумма значений базового адреса регистра и смещения. Смещение — двоичный код в разрядах 2, 1, 0 адреса регистра. Неиспользуемые разряды могут принимать произвольные значения.

Содержимое разрядов регистра управления и состояния приемника для адаптеров А1.1 и А1.2 приведено в табл. 33.

Назначение разрядов буферных регистров принимаемых данных RBUF1, RBUF2 для адаптеров А1.1 и А1.2:

Номер контроллера	Адаптер	Базовый адрес регистров	Адрес вектора	
			приема	передачи
0	A1.1	175610	610	614
	A1.2	177510	174	200
	A1.3	164000	164	170
	A1.4	167770	770	774
1	A1.1	175610	610	614
	A1.2	176500	500	504
	A1.3	177510	174	200
	A1.4	167770	770	774
2	A1.1	175610	610	614
	A1.2	177510	174	200
	A1.3	177550	70	74
	A1.4	167770	770	774
3	A1.1	175620	620	624
	A1.2	176510	510	514
	A1.3	177550	70	74
	A1.4	167760	760	764
4	A1.1	175630	630	634
	A1.2	176520	520	524
	A1.3	164010	550	554
	A1.4	167750	750	754
5	A1.1	175640	640	644
	A1.2	176530	530	534
	A1.3	164020	560	564
	A1.4	167740	740	744
6	A1.1	175650	650	654
	A1.2	176540	540	544
	A1.3	164030	570	574
	A1.4	167730	730	734
7	A1.1	175660	660	664
	A1.2	176500	500	504
	A1.3	164040	600	604
	A1.4	167720	720	724

Таблица 32

Наименование регистров	Мнемоника регистров адаптера		Смещение адреса	Операция на МПИ
	A1.1	A1.2		
Регистр управления и состояния приемника	RCSR1	RCSR2	0	Чтение, запись
Буферный регистр принимаемых данных	RBUF1	RBUF2	2	Чтение
Регистр управления и состояния передатчика	TCSR1	TCSR2	4	Чтение, запись
Буферный регистр передаваемых данных	TBUF1	TBUF2	6	Запись

Разряд	Назначение разрядов регистра	
	RCSR1	RCSR2
15	Устанавливается по любому изменению состояния разрядов RCSR1 [12], RCSR1 [13], а также при появлении единицы в разряде RCSR1 [14]. Программно только читается. Сбрасывается при чтении регистра RCSR1 или сигналом BINITL	Не используется
14	Отражает состояние цепи 125. Программно только читается	»
13	Отражает состояние цепи 106. Программно только читается	»
12	Отражает состояние цепи 109. Программно только читается	»
11...8	Не используются	»
7	Устанавливается аппаратно по завершению приема символа. RCSR1 [7]=1 формирует запрос на прерывание от приемника. Программно только читается. Сбрасывается при чтении регистра RBUF1 или сигналом BINITL	Устанавливается аппаратно по завершению приема символа. RCSR2 [7]=1 формирует запрос на прерывание от приемника. Программно только читается. Сбрасывается при чтении регистра RBUF2 или сигналом BINITL
6	Разрешает выдачу запроса на прерывание от приемника при установленном разряде RCSR1 [7]. Программно читается и пишется. Сбрасывается сигналом BINITL	Разрешает выдачу запроса на прерывание от приемника при установленном разряде RCSR2 [7]. Программно читается и пишется. Сбрасывается сигналом BINITL
5	Разрешает выдачу запроса на прерывание от приемника при установленном разряде RCSR1 [15]. Программно читается и пишется. Сбрасывается сигналом BINITL	Не используется
4, 3	Не используются	»
2	Управляет состоянием цепи 105. Программно читается и пишется. Сбрасывается сигналом BINITL	»
1	Управляет состоянием цепи 108.2. Программно читается и пишется. Сбрасывается сигналом BINITL	»
0	Не используется	»

разряд 15 — устанавливается аппаратно, если один из разрядов RBUF [14...12] установлен в единицу. Сбрасывается при приеме очередного байта данных в регистр RBUF [7...0] или сигналом BINITL;

разряд 14 — устанавливается при переполнении регистра RBUF [7...0]. Сбрасывается при приеме очередного байта данных в регистр RBUF [7...0] или сигналом BINITL;

разряд 13 — устанавливается при обнаружении ошибки формата в принятом символе. Сбрасывается при приеме очередного байта данных в регистр RBUF [7...0] или сигналом BINITL;

разряд 12 — устанавливается при обнаружении

ошибки паритета в принятом символе. Сбрасывается при приеме очередного байта данных в регистр RBUF [7...0] или сигналом BINITL;  
разряды 11...8 — не используются;  
разряды 7...0 — образуют буфер принимаемых из канала последовательных данных. Сбрасываются при приеме очередного байта данных или сигналом BINITL.

Назначение разрядов регистров управления и состояния передатчика TCSR1 и TCSR2 для адаптеров A1.1 и A1.2:

разряды 15...13, 5...3, 1 — не используются;

разряд 12 — используется для имитации тактовых сигналов генератора. Устанавливается и сбрасывается программно. Сбрасывается также сигналом BINITL;

разряд 7 — устанавливается аппаратно после переписи байта данных из регистра TBUF [7...0] в сдвигающий регистр передатчика или по сигналу BINITL. TCSR [7]=1 формирует запрос на прерывание от передатчика. Программно только читается. Сбрасывается при записи байта в регистр TBUF [7...0];

разряд 6 — разрешает выдачу запроса на прерывание от передатчика при установленном разряде TCSR [7]. Программно читается и пишется. Сбрасывается сигналом BINITL;

разряд 2 — устанавливается и сбрасывается программно. TCSR[2]=1 позволяет проверить передачу данных приемника стыка С2 в шлейфном режиме, минуя схемы передатчика. Сбрасывается также сигналом BINITL;

разряд 0 — устанавливается и сбрасывается программно. При TCSR[0]=1 в цепи 103 стыка С2 устанавливается потенциал логического нуля. Сбрасывается также сигналом BINITL.

Назначение разрядов буферных регистров передаваемых данных TBUF1 и TBUF2 для адаптеров A1.1 и A1.2:

разряды 15...8 — не используются;

разряды 7...0 — образуют буфер, предназначенный для хранения передаваемых данных. Загружаются параллельными байтами программно. Сбрасываются сигналом BINITL.

Адаптер A1.3 содержит четыре 16-разрядных программно-управляемых регистра, указанных в табл. 34.

Таблица 34

Наименование регистра	Мнемоника	Смещение адреса	Операция на МПИ
Регистр управления и состояния приемника	RCSR3	0	Чтение, запись
Буферный регистр принимаемых данных	RBUF3	2	Чтение
Регистр управления и состояния передатчика	TCSR3	4	Чтение, запись
Буферный регистр передаваемых данных	TBUF3	6	Запись

Назначение разрядов регистра управления и состояния приемника RCSR3:

разряд 15 — указывает на ошибку при приеме данных от внешнего устройства (отсутствие готовности ВУ). Программно только читается;

разряды 14...8, 5...1 — не используются;

разряд 7 — устанавливается приемом данных от ВУ. RCSR3[7]=1 формирует запрос на прерывание от приемника. Программно только читается. Сбрасывается при чтении регистра RBUF3 [7...0], сигналом BINITL или сигналом установки RCSR3[0];

разряд 6 — разрешает выдачу запроса на прерывание от приемника, если RCSR3[7]=1 или RCSR3[15]=1. Программно читается и пишется. Сбрасывается сигналом BINITL;

разряд 0 — формирует сигнал запроса на ввод символа от ВУ. Программно только пишется. Читается всегда нулем. Сбрасывается при приеме данных в регистр RBUF3 [7...0] или сигналом BINITL.

Назначение разрядов буферного регистра принимаемых данных RBUF3:

разряды 15...8 — не используются;

разряды 7...0 — являются буфером данных, принимаемых от ВУ из параллельного 8-разрядного канала.

Назначение разрядов регистра управления и состояния передатчика TCSR3:

разряд 15 — указывает на ошибку при передаче. Отражает отсутствие готовности ВУ. Программно только читается;

разряды 14...8, 5...0 — не используются;

разряд 7 — устанавливается аппаратно снятием сигнала запроса от ВУ из интерфейса ИРПР или сигналом BINITL. TCSR3[7] формирует запрос на прерывание от передатчика. Программно только читается. Сбрасывается при занесении данных в регистр TBUF3[7...0];

разряд 6 — разрешает выдачу запроса на прерывание от передатчика, если TCSR3[7]=1 или TCSR3[15]=1. Программно читается и пишется. Сбрасывается сигналом BINITL.

Назначение разрядов буферного регистра передаваемых данных TBUF3:

разряды 15...8 — не используются;

разряды 7...0 — являются буфером данных, передаваемых в параллельный 8-разрядный канал. Загружаются параллельными байтами программно.

Адаптер А1.4 содержит три 16-разрядных программно-управляемых регистра (табл. 35). Назна-

разряды 14...8, 4...0 — не используются;

разряд 7 — готовность к приему слова данных из интерфейса ИРПР и передаче его в МПИ. Устанавливается аппаратно перед приемом очередного слова в регистр RBUF4. RCSR4[7]=1 формирует запрос на прерывание от приемника. Программно только читается. Сбрасывается при чтении регистра RBUF4 или сигналом BINITL;

разряд 6 — разрешает выдачу запроса на прерывание от приемника при установленном разряде RCSR4[7]. Программно читается и пишется. Сбрасывается сигналом BINITL;

разряд 5 — разрешает выдачу запроса на прерывание от передатчика при установленном разряде RCSR4[15]. Программно читается и пишется. Сбрасывается сигналом BINITL.

Разряды регистра RBUF4 образуют буфер данных, принимаемых из параллельного 16-разрядного канала ПД, разряды регистра TBUF4 — буфер данных, передаваемых в параллельный 16-разрядный канал ПД.

Помимо адаптеров А1.1...А1.4 в состав контроллера СМ 1425.7009 входят узел приемопередатчиков МПИ, узел сопряжения с МПИ и узел синхронизации. Контроллер осуществляет обмен данными между МПИ и каналами ПД под управлением программы. Адаптеры могут работать одновременно. Перед началом обмена регистры RCSR, TCSR загружаются управляющей информацией. Далее в операциях вывода программно анализируется состояние разряда готовности TCSR. При наличии 1 в буферный регистр вывода TBUF адаптеров А1.1, А1.2, А1.3 записывается байт данных, в адаптере А1.4 — слово данных. При этом разряд готовности TCSR сбрасывается в нуль. Адаптеры А1.1 и А1.2 преобразуют параллельные байты в последовательные коды и побитно выдают их в каналы ПД. Адаптеры А1.3 и А1.4 передают в интерфейс ИРПР параллельные коды. При освобождении буферного регистра разряд готовности снова устанавливается в единицу для получения следующего байта. В операциях ввода программно анализируется разряд готовности регистра RCSR. При наличии 1 выполняется считывание байта данных из буферного регистра ввода в память ЭВМ через МПИ. Управление обменом может выполняться с помощью прерывания. Адаптеры А1.1, А1.2, А1.3 подключены к четвертому уровню запроса, А1.4 — к пятому. Разрешение прерывания текущей программы при возникновении соответствующих условий в схемах адаптеров управляется занесением 1 или 0 в определенные разряды регистра RCSR.

Узел сопряжения с МПИ выполняет арбитраж и обслуживание возникающих запросов. Узел синхронизации обеспечивает тактировку передачи, присла данных в адаптерах А1.1 и А1.2. Адаптеры А1.3 и А1.4 выполняют обмен асинхронно.

Таблица 35

Наименование регистра	Мнемоника	Смещение адреса	Операция на МПИ
Регистр управления и состояния приемника-передатчика	RCSR4	0	Чтение, запись
Буферный регистр принимаемых данных	RBUF4	4	Чтение
Буферный регистр передаваемых данных	TBUF4	2	Чтение, запись

чение разрядов регистра управления и состояния приемника-передатчика RCSR4:

разряд 15 — готовность к передаче следующего слова данных из МПИ в интерфейс ИРПР. Устанавливается аппаратно по снятию сигнала запроса от ВУ или сигналом BINITL. RCSR4[15]=1 формирует запрос на прерывание от передатчика. Программно только читается. Сбрасывается при записи данных в регистр TBUF4;

## ТЕХНИЧЕСКИЕ ДАННЫЕ

Тип интерфейса:

для адаптера А1.1 . . . . . С2 с МУ  
 для адаптера А1.2 . . . . . С2 без МУ  
 для адаптеров А1.3 и А1.4 . . . . . ИРПР

Скорость передачи данных для адаптеров

А1.1 и А1.2, бит/с . . . . . 150, 300, 600,  
 1200, 2400, 4800,  
 9600, 19 200

Дальность передачи, м

рекомендовано  
 до 15

Формат данных, бит:		
для адаптеров А1.1 и А1.2 . . .	7 или 8	
для адаптера А1.3 . . . . .	8	
для адаптера А1.4 . . . . .	16	
Способ передачи цифрового сигнала:		
в адаптерах А1.1 и А1.2 . . .	последовательный асинхронный	
	параллельный асинхронный	
в адаптерах А1.3 и А1.4	одновременный или поочередный двусторонний	
Режим организации связи с ВУ:		
в адаптерах А1.1, А1.2 и А1.4	поочередный двусторонний	
в адаптере А1.3	двусторонний	

## Устройство согласования системных интерфейсов СМ 1425.4511

Предназначено для работы в составе ВК СМ 1425 — подключения устройств с интерфейсом ОШ и обеспечивает обмен информацией в режиме прямого доступа и режиме программного управления. Типичное время задержки при прямом доступе со стороны ОШ: при чтении между сигналами BUSMSYNL и BDINL 350 нс, между сигналами BRPLYL и BUSSSYNL 300 нс; при записи между сигналами BUSMSYNL и BDOUPL 450 нс. Напряжение электропитания не более +5 В; потребляемая мощность не более 15 Вт.

Конструктивно УССИ представляет собой один блок элементов. При работе устройства прямого доступа (УПД), подключенного к ОШ, УССИ выполняет преобразование 18-разрядного адреса интерфейса ОШ в 22-разрядный адрес МПИ. Обычно УССИ работает в «прозрачном» режиме: 18-разрядная адресная зона ОШ отображается в младшие 256 Кбайт адресной зоны МПИ. Возможна работа на ОШ со всем адресным пространством МПИ, т. е. адресная зона ОШ может соответствовать различным участкам адресной зоны МПИ. Для организации такого режима работы в УССИ имеется 6-разрядный адресуемый регистр расширения адреса RA. Старшие разряды адреса [21...16] должны задаваться только в регистр RA (при этом разряды расширения адреса 17...16 на ОШ должны обнуляться). Адрес регистра RA может выбираться в пределах от 17760000 до 17777776 и задается перемычками на адресном наборном поле УССИ согласно табл. 36 с учетом распределения

Таблица 36

Колодка	S3				S2				S1			
	7-8	5-6	3-4	1-2	7-8	5-6	3-4	1-2	7-8	5-6	3-4	1-2
Разряды адреса BDAL [12 . . . 01]	12	11	10	9	8	7	6	5	4	3	2	1

Примечания. 1. Если в соответствующем разряде адреса 1, перемычка отсутствует; если в соответствующем разряде адреса 0, перемычка устанавливается. 2. При автономной поставке УССИ перемычки устанавливаются с базовым технологическим адресом 17777774; установлена перемычка S1 1—2.

адресов в комплекс. Преобразование 18-разрядного адреса ОШ в 22-разрядный адрес МПИ с использованием регистра RA показано на рис. 23.

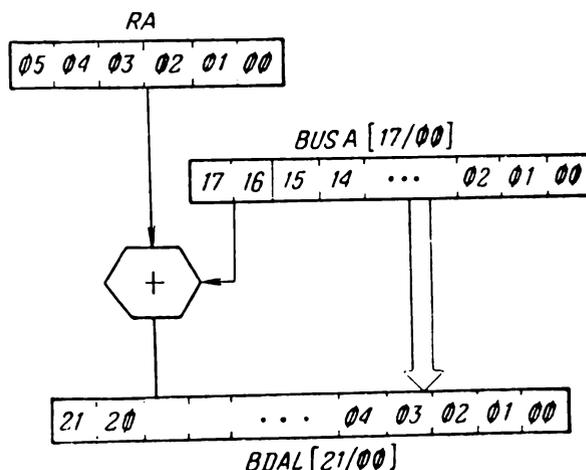


Рис. 23. Схема формирования в УССИ 22-разрядного адреса МПИ из 18-разрядного адреса ОШ

Допускается подключать к УССИ со стороны ОШ не более одного УПД. В случае подключения большего количества необходимо обеспечить их поочередную работу.

При поставке в составе комплекса УССИ устанавливается в монтажном блоке ВК на свободное место, но так, чтобы оно было самым крайним устройством, считая от процессора, из всех установленных. Для подключения устройств к УССИ следует использовать кабель ОШ, поставляемый комплектно с УССИ. Кабель одним концом подключается к разъему X3 УССИ, а другим — ко входному разъему ОШ первого подключаемого устройства.

## Накопители на несменных магнитных дисках

Накопители на несменных жестких магнитных дисках диаметром 130 мм с подвижными магнитными головками состоят из герметизированного корпуса, устройства управления, а также механической и электромеханической частей. К контроллеру комплекса накопители подключаются через интерфейс ИМД-М, соответствующий зарубежному аналогу ST506. Конструктивно подключение осуществляется через два плоских кабеля, предназначенных соответственно для передачи данных и управляющих сигналов. Состав и назначение цепей сигналов на разъемах интерфейса приведены в табл. 37, 38. Линии управления, состояния и обме-

Таблица 37

Номер контакта	Наименование сигнала	Условное обозначение	Источник сигнала
1	Накопитель выбран	DR SELEC-TED-L	Накопитель
2	Земля	⊥	То же
3	Запасной	RESERVED	»
4	Земля	⊥	»

Номер контакта	Наименование сигнала	Условное обозначение	Источник сигнала
5	Запасной	RESERVED	Накопитель
6	Земля	⊥	То же
7	Кабель установлен	OP CAB DET-H	»
8	Земля	⊥	»
9	Резервный	RESERVED	»
10	»	RESERVED	»
11	Земля	⊥	»
12	»	⊥	»
13	+ Данные записи	MFM WRT DATA+	Контроллер
14	- Данные записи	MFM WRT DATA-	То же
15	Земля	⊥	»
16	»	⊥	»
17	+ Данные воспроизведения	MFM RD DATA+	Накопитель
18	- Данные воспроизведения	MFM RD DATA-	То же
19	Земля	⊥	»
20	»	⊥	»

Таблица 38

Номер контакта	Наименование сигнала	Условное обозначение	Источник сигнала
02	Уменьшенный ток записи	RED WRT CUR-L	Контроллер
04	Выбор головки 2 <sup>2</sup>	HD SEL2-L	То же
06	Запись	WRT GATE-L	»
08	Поиск закончен	SEEK COMP-LETE-L	Накопитель
10	Дорожка 000	TRACK0-L	То же
12	Ошибка записи	WRITE FAULT-L	»
14	Выбор головки 2 <sup>9</sup>	HD SEL0-L	Контроллер
16	Кабель установлен	OP CAB DET-H	То же
18	Выбор головки 2 <sup>1</sup>	HD SEL1-L	»
20	Индекс	INDEX-L	Накопитель
22	Накопитель готов	READY-L	То же
24	Шаг	STEP-L	Контроллер
26	Выбор накопителя 1	DR SEL1-L	То же
28	Выбор накопителя 2	DR SEL2-L	»
30	Выбор накопителя 3	DR SEL3-L	»
32	Выбор накопителя 4	DR SEL4-L	»
34	Направление	DIR IN-L	»

на данными выполнены согласованным кабелем из витых пар или плоского кабеля.

В ВК СМ 1425 используется последовательный способ подключения накопителей. Схема накопителей имеет логику выбора для их адресации с по-

мощью контактных перемычек. Формат дорожки НМД представлен на рис. 24.

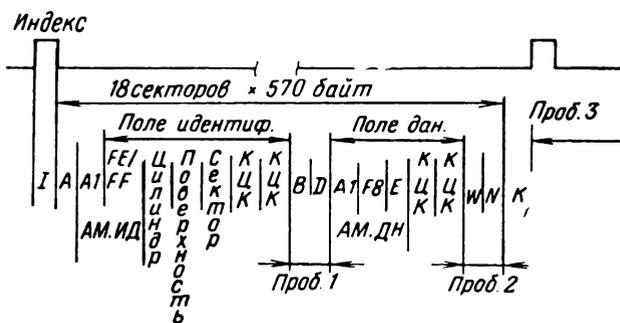


Рис. 24. Формат дорожки НМД:

I — индексная зона 30\*4E; A — синхронизация 13\*00; AI — специальный синхромаркер; B — 1\*00; D — синхронизация 13\*00; E — данные 512\*NN; W — 1\*00; N — 16\*4E; K — 352\*4E номинал; AM.ИД — адресный маркер зоны заголовка; AM.ДН — адресный маркер зоны данных; Проб. 1, Проб. 2, Проб. 3 — промежутки синхронизации; КЦК — контрольный циклический код

**ТЕХНИЧЕСКИЕ ДАННЫЕ**

	СМ 5508	СМ5514.02
Информационная емкость, Мбайт:		
неформатированная	12,76	26,66
форматированная	10/11	23,6
Скорость обмена, Мбит/с	5	
Метод записи	MFM	
Среднее время доступа, мс	85	48,33
Плотность записи, не менее:		
продольная (линейная), бит/мм	357,2	360
поперечная, дорожек/мм	13,6	15
Число дорожек на поверхности	306	320
Время готовности к работе, с	30	20
Питание от источника постоянного тока напряжением, В		5; 12
Номинальный/максимальный потребляемый ток от источника, А:		
5 В	2/4,5	2/4,6
12 В	1,2/1,5	1/1,2
Потребляемая мощность, Вт	30	29
Габаритные размеры, мм	86×150×208	82,5×146×240
Масса, кг	2,5	3,3

**Накопитель на гибком магнитном диске СМ 5640**

Является накопителем на миниатюрном гибком магнитном диске диаметром 130 мм с максимальной емкостью памяти 400 Кбайт при способе записи MFM. Электропитание накопителя обеспечивается от вторичного источника постоянного стабилизированного напряжения ВК. В качестве носителя данных применяется мини-дискета диаметром 130 мм (5,25"), которая предназначена для работы с двойной плотностью записи, двойной плотностью дорожек (96 tpi) и количеством дорожек 80.

Накопитель подключается с помощью 26-полюсного разъема (вилки) 102-26 ТГЛ2931/04 производства ГДР. На кабеле используется розетка 222-26 ТГЛ29331/04-7. Для подключения питания применяется 10-полюсный разъем ТГЛ29331/04. Перечень сигналов интерфейса приведен в табл. 39. Формат дорожки НГМД представлен на рис. 25.

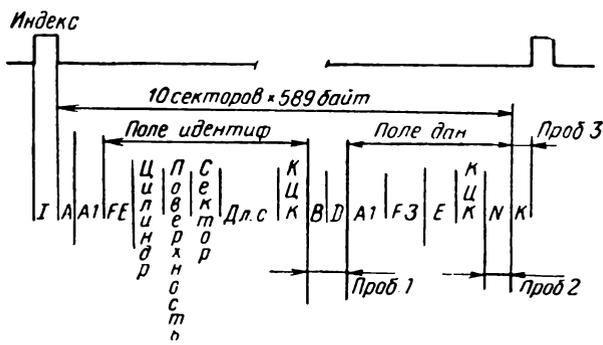


Рис. 25. Формат дорожки НГМД:

I — индексная зона 32\*4; A — синхронизация 12\*00; A1 — специальный синхромаркер 3\*4; D — синхронизация 12\*00; N — 36\*4E; K — 118\*4E номинал; Цилиндр — номер дорожки от 00 до 01, 1 байт; Поверхность — сторона ГМД от 00 до 01, 1 байт; Сектор — адрес сектора от 01 до 0A, 1 байт; Дл. с — признак длины сектора, 1 байт 02; КЦК — контрольный циклический код 2 байта, начиная с A1 (остальные обозначения см. в рис. 24)

Таблица 39

Наименование сигнала	Обозначение	Номер контакта	Источник сигнала
Включение двигателя	/MO	B6	Контроллер
Выбор накопителя 0	/SE	A4	То же
Выбор накопителя 1	/SE	B5	»
Выбор накопителя 2	/SE	A5	»
Выбор накопителя 3	/SE	A2	»
Шаговое позиционирование магнитной головки	/ST	A7	»
Разрешение записи	/WE	A9	»
Направление позиционирования головки	/SD	B7	»
Индекс	/IX	A3	Накопитель
Дорожка 00	/TO	B10	То же
Блокировка записи	/WP	B11	»
Готовность	/RDY	A13	»
Данные записи	/WD	B11	Контроллер
Данные воспроизведения	/RD	B12	То же

### ТЕХНИЧЕСКИЕ ДАННЫЕ

Количество рабочих поверхностей	1
Плотность записи, бит/мм	220
Плотность дорожек, мм <sup>-1</sup>	3,78
Частота вращения диска, мин <sup>-1</sup>	300±6
Количество дорожек	80
Скорость передачи данных, Мбит/с	250
Питание от источника постоянного тока напряжением, В	5; 12
Ток, потребляемый от источника, А:	
5 В	0,9—1
12 В	1—2,2
Габаритные размеры, мм	146,1×82,6×210
Масса, кг	1,5

## Видеотерминал растровый СМ 7238

Видеотерминал растровый СМ 7238 выпускается трех модификаций. Основное исполнение СМ 7238 предназначено для отображения алфавитно-цифровой и псевдографической информации с воз-

можностью вывода ее на внешнее устройство печати. Исполнение СМ 7238.01 дополнительно имеет возможность отображения графической информации с системой команд ТЕКТРОНИХ 4010/4014 и отображения информации на внешнем цветном мониторе. Исполнение СМ 7238.02 не имеет вывода на печать, но имеет возможность отображения графической информации с системой команд REGIS и отображения информации на внешнем цветном видеомониторе.

Видеотерминал снабжен портом связи с ЭВМ и обеспечивает работу с ней по стыку С2 в соответствии с требованиями ГОСТ 18145—81 и ГОСТ 23675—79. Видеотерминалы выпускаются также в варианте с интерфейсом ИРПС в соответствии с НМ МПК по ВТ 10—78. Подключение печатающего устройства к видеотерминалу идет по стыку С2 со скоростью обмена 150...9600 бит/с. Обмен информацией с комплексом производится в 7-битном коде (КОИ-7) и 8-битном коде (КОИ-8) согласно ГОСТ 19768—74. Скорость обмена информацией с внешним печатающим устройством до 9600 бит/с.

Видеотерминал СМ 7238 имеет в своем составе средства проверки функционирования узлов с помощью набора встроенных тестов диагностики.

### ТЕХНИЧЕСКИЕ ДАННЫЕ

Количество отображаемых символов	1920
Емкость буферной памяти, Кслов	3,3
Максимальное число строк	24
Число разновидностей символов в наборе	332
Максимальное число символов в строке	80 или 132
Формат матрицы:	
символа	7×8
изображения программируемых символов	8×10
Число адресуемых точек графического изображения	512×250
Размер поля изображения растровой графической информации, мм	210×190
Количество градаций яркости	4
Количество одновременно отображаемых цветов на внешнем цветном видеомониторе	4
Скорость обмена информацией, бит/с	19 200
Питание от сети переменного тока:	
напряжение, В	220
частота, Гц	50
Потребляемая мощность, В·А	140
Габаритные размеры, мм:	
модуля дисплейного	415×360×400
клавиатуры	505×53×240
Масса, кг	19,5

## Устройство последовательной печати СМ 6329.02

СМ 6329 представляет собой экономичное печатающее устройство со средней скоростью печати и игловой системой получения изображения. Благодаря кассетному принципу подключения к интерфейсу вид его определяется по заказу:

стык С2 (RS232C V.24) со скоростью обмена 150, 200, 300, 1200, 2400, 4800, 9600 бит/с;

ИРПС (токовая петля 20 мА) со скоростью 150, 200, 300, 1200, 2400, 4800, 9600 бит/с;

ИРПР;

параллельный интерфейс CENTRONICS.

Устройство поддерживает стандарт на передачу информации EPSON ESC/-P80. или ESC/-P81. Обеспечивает работу как в текстовом, так и в графическом режиме.

## ТЕХНИЧЕСКИЕ ДАННЫЕ

Принцип печати . . . . .	иглопечатающее матричное уст- ройство после- довательного действия
Скорость печати, знаков/с . . . . .	100
Направление печати . . . . .	в двух направ- лениях с логи- ческой имитаци- ей движения го- ловки
Количество иглол . . . . .	9
Ширина формуляров, мм: непрерывного сфальцованного с кра- сочной перфорацией . . . . .	130—420

рулонного диаметром не более 128 мм	85—420
Количество печатаемых экземпляров . . . . .	оригинал и 2 ко- пии
Растр символов, точек . . . . .	9×9
Число символов в наборе . . . . .	95 и 3 различ- ных набора
Размеры символов . . . . .	6 вариантов шрифтов, выби- раемых переключателями или программно
Питание от сети переменного тока:	
напряжение, В . . . . .	220
частота, Гц . . . . .	50
Потребляемая мощность, В·А . . . . .	70
Габаритные размеры, мм . . . . .	540×290×140
Масса, кг . . . . .	10

## ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ

### Встроенная программа загрузки и диагностики

Процессор имеет ПЗУ емкостью 8К 16-разрядных слов (две ИМС 8К×8 бит) для размещения программ встроенных тестов, программ-загрузчиков и вспомогательных программ. Назначение встроенных тестов — самотестирование процессора и памяти при включении питания, дополнительное тестирование связей устройств внешней памяти и консольного терминала с процессором и оперативной памятью, а также организация самотестирования процессора в прогонном режиме. Программы-загрузчики обеспечивают начальную загрузку программ с устройств внешней памяти и линий связи, вспомогательные программы — диалог с оператором, копирование, сохранение и восстановление дисковой памяти.

Для связи с оператором встроенные тесты используют консольный видеотерминал, а также регистры BCR и BDR. Регистр BCR только читается, регистр BDR только пишется. Оба они имеют на шине один и тот же адрес 17777524. BCR — 16-разрядный регистр управления загрузчиком. Разряды 8...15 не используются. Значение BCR[0...7] определяется состоянием 8 переключателей на плате пульта диагностики, задающих режимы работы тестов. BDR — регистр индикации загрузчика. Значение разрядов 0...5 высвечивается 6 светодиодами на плате пульта диагностики.

ПЗУ емкостью 32 К слов адресуется через две области адресов по 256 слов каждая: 17773000...17773777 и 17765000...17765777, которые называются соответственно «окно 173000» и «окно 165000». Для доступа ко всему объему ПЗУ через эти два окна используется регистр управления выбором слоя PCR. Регистр только записывается, его адрес на шине 17777522, формат представлен на рис. 26.

При обращении к области адресов шины от 17773000 до 17773777 адрес ячейки ПЗУ формиру-

ется из битов PCR[9...15] (старшие разряды адреса ПЗУ) и битов [8...1] адреса шины (младшие разряды адреса ПЗУ). При обращении к области адресов шины 17765000...17765777 адрес ячейки ПЗУ формируется из битов PCR[7...1] и битов [8...1] адреса шины. Разряд 0 адреса шины используется при чтении ПЗУ в байтовых операциях. Физически разряд 0 соответствует выбору одной из двух ИМС. Старшие разряды адреса ПЗУ, формируемые по PCR, называются номером слоя (семь разрядов позволяют адресовать 128 слоев), а младшие — адресом ячейки в слое. Так как регистр PCR читается всегда нулями, определить содержимое разрядов PCR[9...15] и PCR[1...7] можно, прочитав ячейки 17773774 и 17765774 соответственно в каждом слое.

### Тестовое обеспечение

Тестовое программное обеспечение ВК СМ 1425—ТОС 1425 позволяет осуществить проверку функционирования как СМ 1425 в целом, так и его отдельных устройств, представляет собой комплекс испытательных (тестовых) программ и содержит управляющую часть, средства мультипрограммной проверки, тестовые программы и справочно-информационные файлы. Как программная система с широким набором выполняемых функций по организации программной проверки ТОС 1425 обладает следующими свойствами:

- расширяемостью по составу проверяемых устройств и тестовых программ;

- унифицированностью требований к структуре и оформлению тестовых программ и их интерфейсу с управляющими программами;

- возможностью получения справочной информации по системе.

ТОС 1425 позволяет хранить все тестовое обеспечение на магнитных носителях, причем допускается копирование и проверка текстовых файлов. Управляющая часть системы—это мониторы, драйверы, служебные программы, обеспечивающие подготовку и управление процессом программной проверки. Средства мультипрограммной проверки обеспечивают создание и выполнение комплексных тестов для проверки ВК СМ 1425 в мультипрограммном режиме. Справочно-информационные файлы служат для облегчения пользования ТОС 1425 и

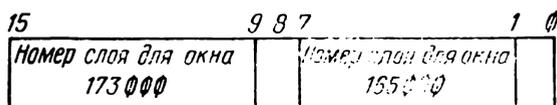


Рис. 26. Формат регистра PCR

предусматривают возможность получения справочной информации. В состав ТОС 1425 входят тесты процессора, оперативной памяти и внешних устройств.

ТОС 1425 включена в вычислительный комплекс СМ 1425 в качестве составной части и функционирует на ВК СМ 1425 с емкостью оперативной памяти не менее 512 Кбайт при следующем минимальном составе внешних устройств: комбинированный контроллер НГМД и НМД, консольный терминал, накопитель на ГМД. ТОС 1425 программно совместима с тестовой диагностической операционной системой ТЕДОС. Носителями данных ТОС 1425 являются гибкие диски типа ГМД-130.

## Системное программное обеспечение

ОС РВМ — многофункциональная операционная система реального времени, предназначенная для ЭВМ типа СМ 1425 и СМ 1420. ОС РВМ используется для обеспечения быстрого ответа на события реального времени, управления технологическими и научными экспериментами в реальном времени, расчетов научно-технического характера с повышенными требованиями к емкости памяти, автоматизированного проектирования с использованием мультипрограммного режима, разработки и отладки программ с применением языков программирования высокого уровня в интерактивном режиме. ОС РВМ может иметь целевое назначение или быть универсальной многотерминальной системой, используемой многими пользователями. Применяется в качестве базовой операционной системы в локальных многомашинных и территориально-распределенных управляющих вычислительных комплексах для организации работы распределенных систем.

ОС РВМ обеспечивает: эффективное использование ресурсов комплекса; удобство работы пользователя; мультипрограммную работу в режиме реального времени, разделения времени и в пакетном режиме; пакетное управление выполнением задач; управление разделением оперативной памяти на память инструкций и память данных; управление выполнением системных и пользовательских программ в трех режимах работы процессора; управление файлами с последовательной, относительной и индексно-последовательной организацией; обслуживание операций ввода-вывода в реальном масштабе времени; возможность организации сетей

ЭВМ и файловой структуры на дисках и магнитных лентах разных типов; использование набора внешних устройств из номенклатуры СМ ЭВМ; подключение пользовательских драйверов для дополнительных внешних устройств; динамическое распределение и уплотнение памяти объемом до 4 Мбайт; динамическую загрузку задач с диска и выгрузку их на диск. В ОС РВМ реализована совместимость по основным параметрам с операционной системой ОС РВЗ и возможность передачи файлов между томами с файловой структурой ДОС, РАФОС, РАФОС-2.

ОС РВМ рассчитана на работу с разнообразным оборудованием. В зависимости от конкретного применения ОС РВМ генерируются ее различные версии: от небольших систем для лабораторных исследований до больших систем, включающих средства для межмашинных взаимодействий локальных и распределенных многомашинных управляющих вычислительных комплексов.

В ОС РВМ включены следующие языки программирования: МАКРОАСSEMBЛЕР, ФОРТРАН-77, КОБОЛ, БЕЙСИК, ПАСКАЛЬ.

Язык МАКРОАСSEMBЛЕР является стандартным языком программирования для ОС РВМ. Имеет полный набор средств символического кодирования инструкций процессора, средств распределения памяти, секционирования программ; позволяет использовать как системные макрокоманды, так и макрокоманды, написанные пользователем.

Язык ФОРТРАН-77 в ОС РВМ используется для программирования вычислительных задач, а также задач, работающих в реальном времени.

Язык КОБОЛ ориентирован на решение экономических задач. Он позволяет писать программы преимущественно для экономических приложений на языке, достаточно близком к обычному.

БЕЙСИК — один из наиболее простых языков программирования, использующий простые операторы и хорошо известные математические выражения для выполнения различных операций. Он позволяет пользователю для написания простых программ ограничиваться элементарными операторами и имеет средства для написания и отладки сложных программ.

ПАСКАЛЬ — это язык высокого уровня, позволяющий использовать принципы структурного программирования.

При разработке программ пользователю следует иметь в виду, что для представления символов ОС РВМ использует кодовую таблицу КОИ-8 с упорядоченным русским алфавитом.

## Приложение 1

### СИСТЕМА КОМАНД

Все команды с учетом выполняемых действий и используемых форматов разбиты на одноадресные, двухадресные, команды ветвлений, команды перехода и работы с подпрограммой, команды прерываний, команды изменения признаков, прочие команды управления программой.

При описании команд и режимов адресации используются следующие обозначения:

OP — код команды;

OFF — смещение в командах перехода;

DM — режим адресации приемника;

DR — номер регистра общего назначения приемника;

SM — режим адресации источника;

SR — номер регистра общего назначения источника;

R — номер регистра общего назначения;

(R) — содержимое регистра общего назначения;

(XXX) — содержимое XXX;

DD или DST — адрес приемника;

SS или SRC — адрес источника;  
*a* — признак косвенного режима адресации;  
 X — содержимое следующего (второго или третьего слова) команды при индексной адресации;  
 $\wedge$  — функция «И» (логическое умножение);  
 $\vee$  — функция «ИЛИ» (логическое сложение);  
 + — функция «Исключающее ИЛИ» (сложение по модулю 2);  
 $\neg$  — функция «НЕ» (инверсия);  
 $\leftarrow$  — «Становится»;  
 (SP)+ — выборка из стека;  
 -(SP) — запись в стек;  
 N — разряд признака отрицательного числа в слове состояния;  
 Z — разряд признака нуля в слове состояния;  
 V — разряд признака переполнения в слове состояния;  
 C — разряд признака переноса в слове состояния;  
 TEMP — внутренний регистр процессора, используемый для временного хранения информации.

### РЕЖИМЫ АДРЕСАЦИИ

В процессоре используются 8 режимов адресации. Код режима адресации указывается в команде. Можно выделить два вида адресации — прямую и косвенную. В первом случае адресным полем команды определяется адрес операнда, во втором — адрес адреса операнда. Режимы адресации приведены в табл. 40.

Таблица 40

Код	Мнемоника	Название	Режим
000	R	Регистровый	Содержимое регистра является операндом
001	<i>a</i> R или (R)	Косвенный регистровый	Содержимое регистра является адресом операнда
010	(R)+	Автоувеличение	Содержимое регистра используется как адрес операнда, а затем увеличивается на 1 или 2
011	<i>a</i> (R)+	Косвенный с автоувеличением	Содержимое регистра используется как адрес адреса операнда, а затем увеличивается на 2
100	-(R)	Автоуменьшение	Содержимое регистра уменьшается на 1 или 2, а затем используется как адрес операнда
101	<i>a</i> -(R)	Косвенный с автоуменьшением	Содержимое регистра уменьшается на 2, затем используется как адрес адреса операнда
110	X(R)	Индексный	Содержимое регистра суммируется со значением индекса X, находящимся в следующем слове команды. Полученная сумма является адресом операнда
111	<i>a</i> X(R)	Косвенный индексный	Содержимое регистра суммируется со значением X, находящимся в следующем слове команды. Сумма является адресом адреса операнда

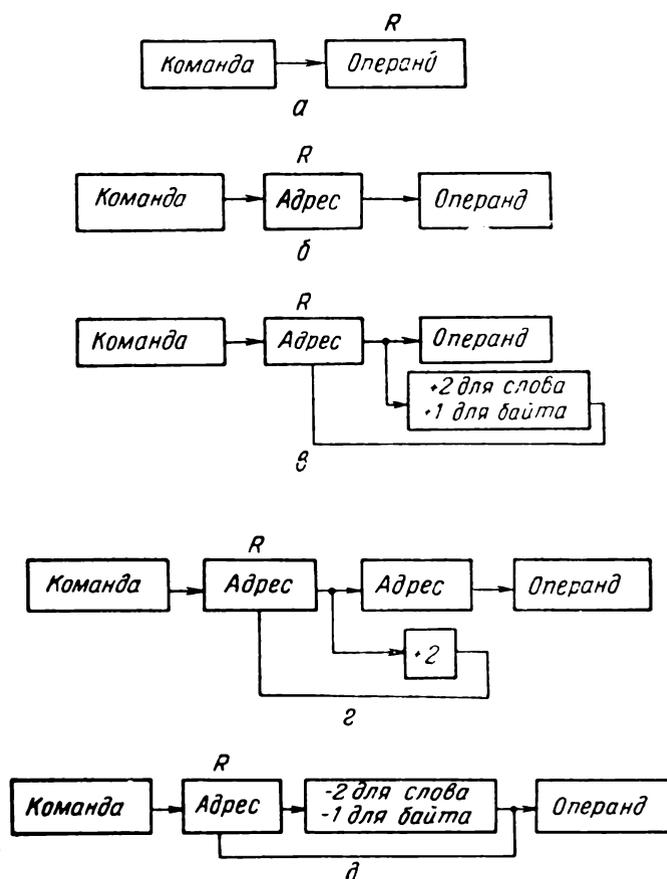
Регистр R7 (PC) также может быть использован в любом режиме адресации. Но на практике имеет смысл использовать его только в четырех режимах, указанных в табл. 41.

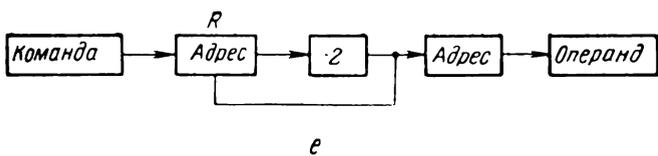
Таблица 41

Код	Мнемоника	Название	Режим
010	#	Непосредственный	Содержимое ячейки, следующей за первым словом команды, является операндом
011	<i>a</i> #A	Абсолютный	Содержимое ячейки A, следующей за первым словом команды, является адресом операнда
110	A	Относительный	Содержимое ячейки, следующей за первым словом команды, суммируется с содержимым PC. Полученная сумма A является адресом операнда
111	<i>a</i> A	Косвенно относительный	Содержимое ячейки, следующей за первым словом команды, суммируется с содержимым PC. Полученная сумма A является адресом адреса операнда

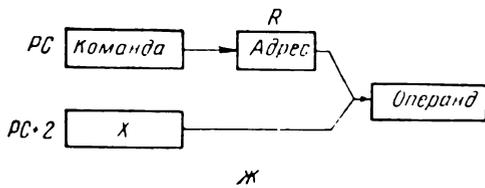
Режимы адресации 6 и 7 называются относительными потому, что адрес вычисляется относительно текущего содержимого PC. Содержимое второго слова команды указывает расстояние (в байтах) между адресом или адресом адреса операнда и текущим адресом в PC.

Все вышеупомянутые способы адресации показаны на рис. 27.

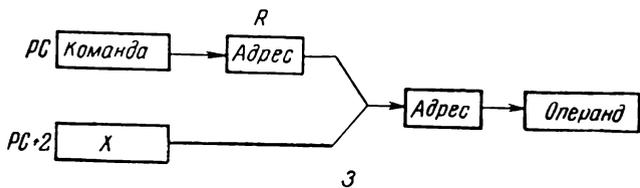




е



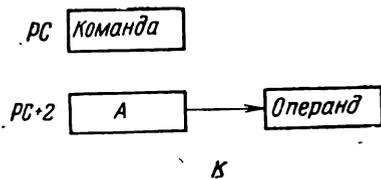
ж



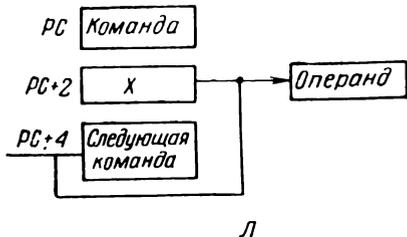
з



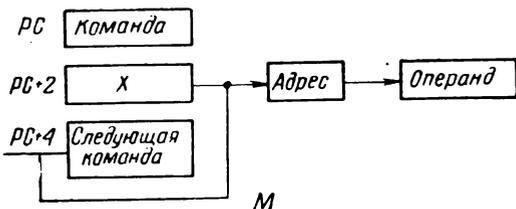
и



к



л



м

Рис. 27. Режимы адресации:

а — прямой регистровый R; б — косвенный регистровый (R); в — прямой с автоувеличением (R)+; г — косвенный с автоувеличением  $a(R)+$ ; д — прямой с автоуменьшением  $-(R)$ ; е — косвенный с автоуменьшением  $a-(R)$ ; ж — прямой индексный X(R); з — косвенный индексный  $aX(R)$ ; и — непосредственная адресация #; к — абсолютная адресация  $a::A$ ; л — относительная адресация A; м — косвенно относительная адресация  $aA$

### БАЙТОВЫЕ КОМАНДЫ

Байтовые команды имеют такие же формат, алгоритм, описание, установку признаков, что и соответствующие команды, оперирующие со словами. В байтовых командах с регистровой адресацией обрабатывается только младший байт адресуемого

регистра. Исключение составляют команды MOV, MFPS, при выполнении которых знак операнда расширяется в старшие (15...8) разряды регистра-приемника.

### ФОРМАТЫ КОМАНД

Одноадресные команды (рис. 28) выполняют действия над одним операндом; в разрядах 5...0 задается адрес приемника (DM, DR) или адрес источника (SM, SR). Двухадресные команды обычно выполняют действия над двумя операндами и адресуют источник и приемник. В формате на рис. 29, а поля DR, DM адресуют приемник, а поля SM, SR — источник. На рис. 29, б показан формат двухадресной команды, в котором разряды 5...0 задают адрес источника или приемника, а второй операнд всегда находится в регистре общего назначения, задаваемом полем R.

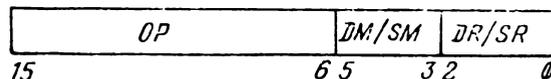
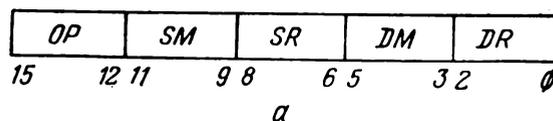
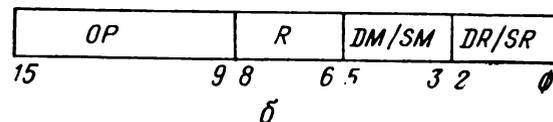


Рис. 28. Формат одноадресных команд и команды JMP



а



б

Рис. 29. Формат двухадресных команд: а — общий; б — команд с регистрами-приемниками и команд JSR, XOR

В командах ветвления (рис. 30, а) в поле OFF задается смещение со знаком, определяющее адрес ветвления относительно значения счетчика команд PC при выполнении условия ветвления. В формате команд EMT, TRAP (рис. 30, б) в поле UC записывается код пользователя, определяющий подпрограмму перехода. Безадресные команды (рис. 30, в) содержат в поле команды только код операции. В командах изменения признаков (рис. 30, г) в разрядах 3...0 указывается, какие из разрядов текущего слова состояния PSW, содержащие признаки, необходимо сбросить в нуль, если четвертый разряд команды равен нулю, или установить в единицу, если четвертый разряд команды равен единице.

В команде RTS (рис. 30, д) в разрядах 2...0 указывается адрес регистра связи, содержащего адрес возврата. В команде MARK (рис. 30, е) в поле NN указывается величина, на которую должен быть сокращен стек. В команде SOB (рис. 30, ж) в шести младших разрядах (NN) указывается величина смещения для определения адреса при переходе по счетчику, а в поле R — регистр, в котором организован счетчик.

В описание каждой команды входят ее наименование, мнемоническое обозначение на языке МАКРОАССЕМБЛЕР, условное обозначение, включающее восьмеричный код операции, алгоритм

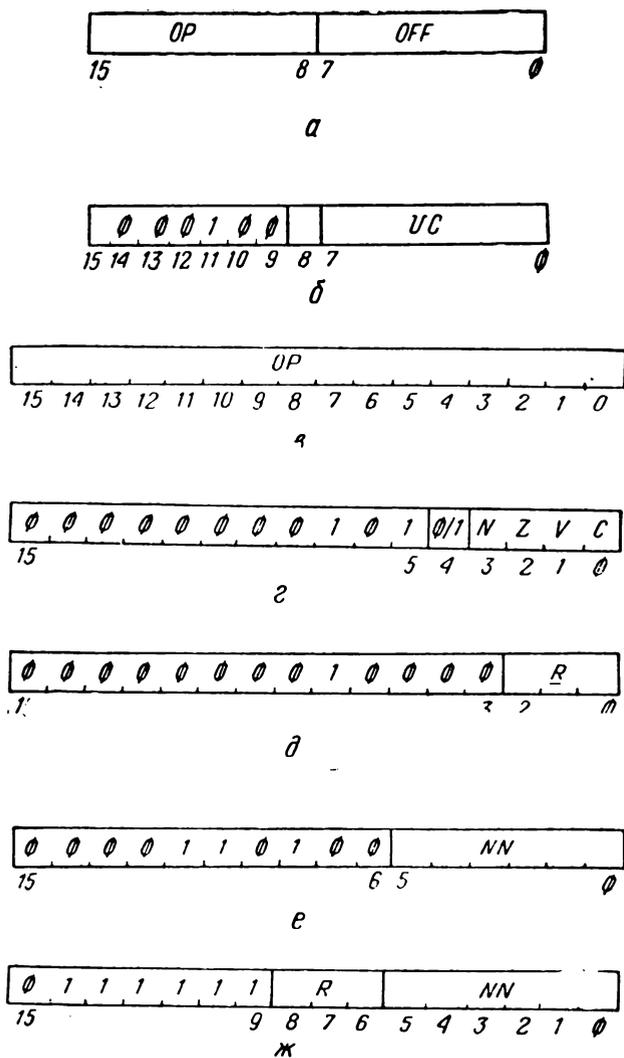


Рис. 30. Форматы команд:

а — ветвлений; б — EMT, TRAP; в — безадресных RTI, BPT, IOT, RTT, MFPT; г — изменения признаков; д — RTS; е — MARK; ж — SOB

выполнения команды, правила установки признаков и краткое пояснение действий, выполняемых командой.

### Одноадресные команды

#### Перестановка байтов:

SWAB 0003DD.

Действие: байт 1 ← байт 0.

Признаки: N = 1, если старший (седьмой) разряд младшего байта результата равен 1, в противном случае N = 0;

Z = 1, если младший байт результата равен 0, в противном случае Z = 0;

V = 0;

C = 0.

Описание: старший и младший байты содержимого приемника меняются местами. Адресация производится по полному слову.

#### Очистка:

CLR 0050DD — очистка слова;

CLRB 1050DD — очистка байта.

Действие: (DST) ← 0.

Признаки: N = 0;

Z = 1;

V = 0;

C = 0.

Описание: все разряды приемника сбрасываются в нуль.

#### Инвертирование:

COM 0051DD — инвертирование слова;

COMB 1051DD — инвертирование байта.

Действие: (DST) ← ¬(DST).

Признаки: N = 1, если результат меньше 0, в противном случае N = 0;

Z = 1, если результат равен 0, иначе Z = 0;

V = 0;

C = 1.

Описание: все разряды приемника инвертируются.

#### Прибавление единицы:

INC 0052DD — прибавление единицы к слову;

INCB 1052DD — прибавление единицы к байту.

Действие: (DST) ← (DST) + 1.

Признаки: N = 1, если результат меньше 0, иначе N = 0;

Z = 1, если результат равен 0, иначе Z = 0;

V = 1, если исходное (DST) = 077777 (слово) или (DST) = 177 (байт), в противном случае V = 0;

C не изменяется.

Описание: к содержимому приемника прибавляется 1.

#### Вычитание единицы:

DEC 0053DD — вычитание единицы из слова;

DECB 1053DD — вычитание единицы из байта.

Действие: (DST) ← (DST) - 1.

Признаки: N = 1, если результат меньше 0, иначе N = 0;

Z = 1, если результат равен 0, иначе Z = 0;

V = 1, если исходное (DST) = 100000 (слово) или (DST) = 200 (байт), в противном случае V = 0;

C не изменяется.

Описание: из содержимого приемника вычитается 1.

#### Изменение знака:

NEG 0054DD — изменение знака слова;

NEGB 1054DD — изменение знака байта.

Действие: (DST) ← -(DST).

Признаки: N = 1, если результат меньше 0, иначе N = 0;

Z = 1, если результат равен 0, иначе Z = 0;

V = 1, если результат равен 100000 (слово) или (DST) = 200 (байт), в противном случае V = 0;

C = 0, если результат равен 0, иначе C = 1.

Описание: содержимое приемника заменяется его дополнением до 2. Число 100000 (слово) или 200 (байт) не меняется, так как наибольшее отрицательное число не имеет положительного аналога в дополнительном коде.

#### Прибавление переноса:

ADC 0055DD — прибавление переноса к слову;

ADCB 1055DD — прибавление переноса к байту.

Действие: (DST) ← (DST) + (C).

Признаки: N = 1, если результат меньше 0, иначе N = 0;

Z = 1, если результат равен 0, иначе Z = 0;

$V:=1$ , если исходное  $(DST)=077777$  (слово) или  $(DST)=177$  (байт), а значение  $C$  было равно 1, в противном случае  $V:=0$ ;  
 $C:=1$ , если исходное  $(DST)=377$  (байт), а значение  $C$  было равно 1, иначе  $C:=0$ .

Описание: Содержимое разряда  $C$  прибавляется к содержимому приемника. Эта команда нужна в многословных операциях. Она дает возможность, например, использовать перенос от сложения двух младших слов при сложении двух старших слов.

#### Вычитание переноса:

$SBC\ 0056DD$  — вычитание переноса из слова;  
 $SBCB\ 1056DD$  — вычитание переноса из байта.  
 Действие:  $(DST) \leftarrow (DST) - (C)$ .

Признаки:  $N:=1$ , если результат меньше 0, иначе  $N:=0$ ;

$Z:=1$ , если результат равен 0, иначе  $Z:=0$ ;

$V:=1$ , если исходное  $(DST)=100000$  (слово) или  $(DST)=200$  (байт), а значение  $C$  было равно 1, иначе  $V:=0$ ;

$C:=1$ , если исходное  $(DST)=0$  и значение  $C$  было равно 1, в противном случае  $C:=0$ .

Описание: из содержимого приемника вычитается содержимое разряда  $C$ . Эта команда позволяет в многословных операциях использовать перенос от вычитания младших слов при вычитании старших слов.

#### Проверка:

$TST\ 0057DD$  — проверка слова;  
 $TSTB\ 1057DD$  — проверка байта.  
 Действие:  $(DST) \leftarrow (DST)$ .

Признаки:  $N:=1$ , если результат меньше 0, иначе  $N:=0$ ;

$Z:=1$ , если результат равен 0, иначе  $Z:=0$ ;

$V:=0$ ;

$C:=0$ .

Описание: в результате выполнения команды содержимое приемника не меняется, а признаки  $N$  и  $Z$  устанавливаются в зависимости от содержимого приемника.

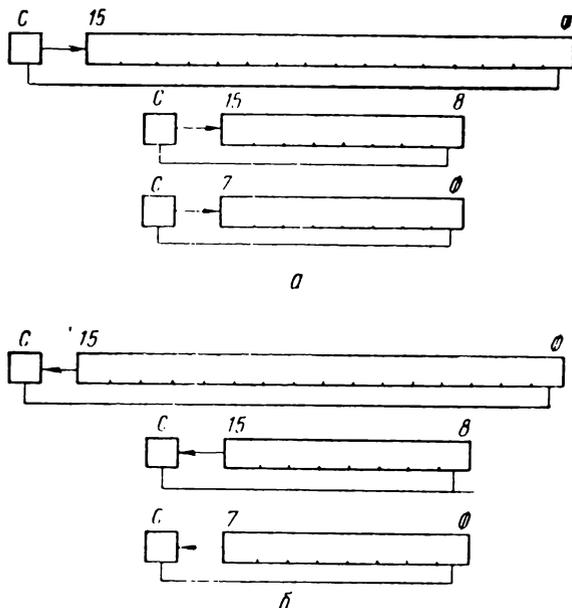


Рис. 31. Циклический сдвиг:  
 а — вправо ROR (B); б — влево ROL (B)

#### Циклический сдвиг вправо (рис. 31, а):

$ROR\ 0060DD$  — циклический сдвиг вправо слова;

$RORB\ 1060DD$  — циклический сдвиг вправо байта.

Действие:  $(DST) \leftarrow (DST)$ , циклически сдвинутое на одну позицию вправо.

Признаки:  $N:=1$ , если старший разряд результата установлен в единицу, иначе  $N:=0$ ;

$Z:=1$ , если результат равен 0, иначе  $Z:=0$ ;

$V$  — загружается значением функции  $N+C$ , значения  $N$  и  $C$  после окончания операции сдвига;

$C$  загружается содержимым младшего разряда приемника.

Описание: производится циклический сдвиг всех разрядов содержимого приемника вправо на один разряд. Нулевой разряд загружается в разряд  $C$  слова состояния, а предыдущее значение  $C$  — в старший разряд приемника.

#### Циклический сдвиг влево (рис. 31, б):

$ROL\ 0061DD$  — циклический сдвиг влево слова;  
 $ROLB\ 1061DD$  — циклический сдвиг влево байта.

Действие:  $(DST) \leftarrow (DST)$ , циклически сдвинутое на одну позицию влево.

Признаки:  $N:=1$ , если старший разряд результата установлен в единицу, иначе  $N:=0$ ;

$Z:=1$ , если результат равен 0, иначе  $Z:=0$ ;

$V$  загружается значением функции  $N+C$ , значения  $N$  и  $C$  после окончания операции сдвига;

$C$  загружается содержимым старшего разряда приемника.

Описание: производится циклический сдвиг всех разрядов содержимого приемника влево на один разряд. Старший разряд загружается в разряд  $C$  слова состояния, а предыдущее значение  $C$  — в нулевой разряд приемника.

#### Арифметический сдвиг вправо (рис. 32, а):

$ASR\ 0062DD$  — арифметический сдвиг вправо слова;

$ASRB\ 1062DD$  — арифметический сдвиг вправо байта.

Действие:  $(DST) \leftarrow (DST)$ , сдвинутое арифметически на одну позицию вправо.

Признаки:  $N:=1$ , если старший разряд результата установлен в единицу, иначе  $N:=0$ ;

$Z:=1$ , если результат равен 0, иначе  $Z:=0$ ;

$V$  загружается значением функции  $N+C$ , значения  $N$  и  $C$  после окончания операции сдвига;

$C$  загружается содержимым младшего разряда приемника.

Описание: все разряды содержимого приемника сдвигаются вправо на один разряд. Знаковый разряд повторяется. Разряд  $C$  слова состояния загружается содержимым нулевого разряда приемника. Команда выполняет деление (со знаком) содержимого приемника на 2.

#### Арифметический сдвиг влево (рис. 32, б):

$ASL\ 0063DD$  — арифметический сдвиг влево слова;

$ASLB\ 1063DD$  — арифметический сдвиг влево байта.

Действие:  $(DST) \leftarrow (DST)$ , сдвинутое арифметически на одну позицию влево.

Признаки:  $N := 1$ , если старший разряд результата установлен в единицу, иначе  $N := 0$ ;

$Z := 1$ , если результат равен 0, иначе  $Z := 0$ ;

$V$  загружается результатом операции  $N + C$ , значения  $N$  и  $C$  после окончания операции сдвига;

$C$  загружается содержимым старшего разряда приемника.

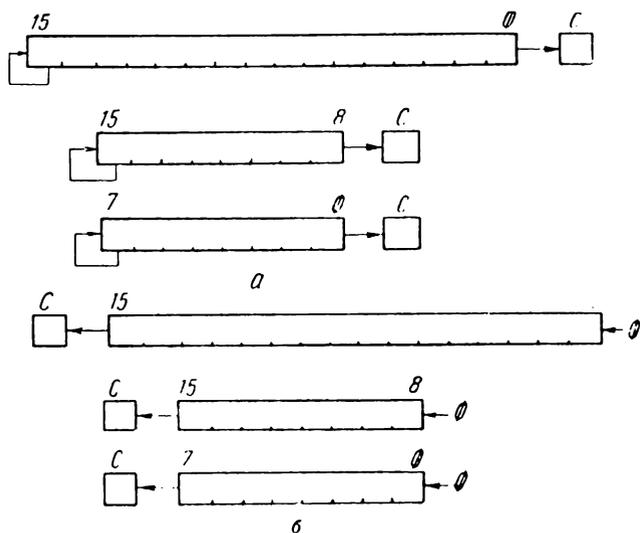


Рис. 32. Арифметический сдвиг (одноадресные команды):  
а — вправо ASR(B); б — влево ASL(B)

Описание: все разряды приемника сдвигаются на одну позицию влево. В младший разряд операнда записывается 0. Разряд  $C$  слова состояния загружается содержимым старшего разряда приемника. Команда выполняет умножение (со знаком) содержимого приемника на 2 с индикацией переполнения.

#### Запись PSW:

MTPS 1064SS.

Действие:  $PSW[7...0] \leftarrow (SRC)$ .

Признаки: устанавливаются в соответствии с разрядами 3...0 операнда источника.

Описание: байт, указанный адресом источника, записывается в  $PSW[7...0]$ . В режиме «Ядро» разряды  $PSW[7...5]$  записываются из операнда источника, а в режимах «Пользователь» и «Супервизор» разряды  $PSW[7...5]$  не меняются. Разряд  $PSW[4]$  не может быть установлен этой командой.

#### Пересылка из пространства данных или команд предыдущего режима:

MFPI 0065SS — пересылка из пространства команд предыдущего режима:

MFPD 1065SS — пересылка из пространства данных предыдущего режима.

Действие:  $TEMP \leftarrow (SRC)$ ;  $-(SP) \leftarrow (TEMP)$ .

Признаки:  $N := 1$ , если результат меньше 0, иначе  $N := 0$ ;

$Z := 1$ , если результат равен 0, иначе  $Z := 0$ ;

$V := 0$ ;

$C$  не изменяется.

Описание: команда выбирает слово по адресу источника, определяемому предыдущим режимом ( $PSW[13, 12]$ ), и записывает его в стек текущего

режима. Адрес источника вычисляется в текущем режиме. Данные команды можно использовать для связи текущего стека со стеком предыдущего режима. При нулевом режиме адресации и регистре  $R6$  в качестве операнда источника выбирается указатель стека предыдущего режима.

#### Пересылка в пространство данных или команд предыдущего режима:

MTP1 0066DD — пересылка в пространство команд предыдущего режима;

MTPD 1066DD — пересылка в пространство данных предыдущего режима.

Действие  $TEMP \leftarrow (SP) +$ ;  $(DST) \leftarrow (TEMP)$ .

Признаки:  $N := 1$ , если результат меньше 0, иначе  $N := 0$ ;

$Z := 1$ , если результат равен 0, иначе  $Z := 0$ ;

$V := 0$ ;

$C$  не изменяется.

Описание: команда выбирает слово из текущего стека, определяемого  $PSW[15, 14]$  и записывает его по адресу приемника, определяемому предыдущим режимом ( $PSW[13, 12]$ ). Адрес приемника вычисляется в текущем режиме. При нулевом режиме адресации и регистре  $R6$  адресом приемника является указатель стека предыдущего режима.

#### Расширение знака:

SXT 0067DD.

Действие:  $(DST) \leftarrow 0$ , если  $N = 0$ ;  $(DST) \leftarrow \leftarrow 177777$ , если  $N = 1$ .

Признаки:  $N$  не изменяется;

$Z$  устанавливается в единицу, если  $N = 0$ ;

$V := 0$ ;

$C$  не изменяется.

Описание: команда SXT заполняет все разряды приемника единицами или нулями в зависимости от разряда  $N$ , установленного в PSW предшествующей командой. Эта команда используется в вычислениях с двойной точностью. Она позволяет преобразовать число, содержащееся в одном слове, в эквивалентное число с учетом знака, состоящее из двух слов.

#### Чтение PSW:

MFPS 1067DD.

Действие:  $(DST) \leftarrow PSW[7...0]$ .

Признаки:  $N := 1$ , если старший разряд результата установлен в единицу, иначе  $N := 0$ ;

$Z := 1$ , если результат равен 0, иначе  $Z := 0$ ;

$V := 0$ ;

$C$  не изменяется.

Описание: пересылаются младшие 8 разрядов слова состояния по адресу приемника, который является адресом байта. При нулевом режиме адресации разряд  $PSW[7]$  расширяется на старший байт регистра-приемника.

#### Проверка приемника и установка младшего разряда:

TSTSET 0072DD.

Действие:  $RO \leftarrow (DST)$ ;  $(DST) \leftarrow (DST) \vee 000001$ .

Признаки:  $N := 1$ , если содержимое регистра  $R0$  меньше 0, иначе  $N := 0$ ;

$Z := 1$ , если содержимое регистра  $R0$  равно 0, иначе  $Z := 0$ ;

$V := 0$ ;

$C$  принимает исходное значение  $(DST[0])$ .

Описание: читается с блокировкой шины слово приемника и запоминается в R0. Записывается в приемник его старое содержимое с установкой в единицу младшего разряда и блокируется шина.

Прерывание: при регистровой адресации прерывание по вектору 10.

**Запись с блокировкой шины:**

WRTLCK 0073DD.

Действие:  $(DST) \leftarrow (R0)$ .

Признаки:  $N := 1$ , если результат меньше 0, иначе  $N := 0$ ;

$Z := 1$ , если результат равен 0, иначе  $Z := 0$ ;

$V := 0$ ;

C не изменяется.

Описание: выполняются чтение приемника с блокировкой шины и запись по адресу приемника содержимого R0 с последующей разблокировкой шины.

Прерывание: при регистровой адресации прерывание по вектору 10.

### Двухадресные команды

**Пересылка:**

MOV 01SSDD — пересылка слова;

MOVB 11SSDD — пересылка байта.

Действие:  $(DST) \leftarrow (SRC)$ .

Признаки:  $N := 1$ , если содержимое источника меньше 0, иначе  $N := 0$ ;

$Z := 1$ , если содержимое источника равно 0, иначе  $Z := 0$ ;

$V := 0$ ;

C не изменяется.

Описание: операнд источника пересылается по адресу приемника. Предыдущее содержимое приемника теряется. Содержимое ячейки (SRC) не изменяется. Команда MOVB с использованием регистрового режима адресации (единственная среди байтовых команд) расширяет старший разряд младшего байта на старший байт (расширение знака).

**Сравнение:**

CMP 02SSDD — сравнение слова;

CMPB 12SSDD — сравнение байта.

Действие:  $(SRC) - (DST)$ .

Признаки:  $N := 1$ , если результат меньше 0, иначе  $N := 0$ ;

$Z := 1$ , если результат операции равен 0, иначе  $Z := 0$ ;

$V := 1$ , если произошло арифметическое переполнение, т. е. если операнды имели разные знаки, а знаки результата и приемника совпадают, в противном случае  $V := 0$ ;

$C := 0$ , если был перенос из старшего разряда результата, иначе  $C := 1$ .

Описание: производится арифметическое сравнение операндов приемника и источника, операнды не изменяются. В результате операции устанавливаются только соответствующие признаки.

**Проверка разрядов:**

BIT 03SSDD — проверка разрядов слова;

BITB 13SSDD — проверка разрядов байта.

Действие:  $(SRC) \wedge (DST)$ .

Признаки:  $N := 1$ , если старший разряд результата установлен в единицу, иначе  $N := 0$ ;

$Z := 1$ , если результат равен 0, иначе  $Z := 0$ ;

$V := 0$ ;

C не изменяется.

Описание: выполняется операция «И» (логиче-

ское умножение) над операндами источника и приемника. Операнды не изменяются. В результате операции устанавливаются только соответствующие признаки.

**Очистка разрядов:**

BIC 04SSDD — очистка разрядов слова;

BICB 14SSDD — очистка разрядов байта.

Действие:  $(DST) \leftarrow \neg (SRC) \wedge (DST)$ .

Признаки:  $N := 1$ , если старший разряд результата установлен в единицу, иначе  $N := 0$ ;

$Z := 1$ , если результат равен 0, иначе  $Z := 0$ ;

$V := 0$ ;

C не изменяется.

Описание: выполняется операция логического умножения над содержимым операнда приемника и инвертированным операндом источника. Разряды источника, содержащие 1, сбрасывают соответствующие разряды приемника. Первоначальное содержимое приемника теряется. Этой командой сбрасываются в нуль выбранные разряды приемника.

**Логическое сложение:**

BIS 05SSDD — логическое сложение слов;

BISB 15SSDD — логическое сложение байтов.

Действие:  $(DST) \leftarrow (SRC) \vee (DST)$ .

Признаки:  $N := 1$ , если старший разряд результата установлен в единицу, иначе  $N := 0$ ;

$Z := 1$ , если результат равен 0, иначе  $Z := 0$ ;

$V := 0$ ;

C не изменяется.

Описание: выполняется операция «ИЛИ» (логическое сложение) над операндами источника и приемника: результат запоминается по адресу приемника. Первоначальное содержимое приемника теряется. Этой командой устанавливаются в единицу выбранные разряды приемника.

**Сложение:**

ADD 06SSDD.

Действие:  $(DST) \leftarrow (DST) + (SRC)$ .

Признаки:  $N := 1$ , если результат меньше 0, иначе  $N := 0$ ;

$Z := 1$ , если результат операции равен 0, иначе  $Z := 0$ ;

$V := 1$ , если произошло арифметическое переполнение, т. е. если знаки операндов совпали, а результат имеет противоположный знак, иначе  $V := 0$ .

$C := 1$ , если был перенос из старшего разряда результата, иначе  $C := 0$ .

Описание: производится сложение операнда-источника и операнда-приемника. Результат запоминается по адресу приемника. Первоначальное содержимое приемника теряется.

**Вычитание:**

SUB 16SSDD.

Действие:  $(DST) \leftarrow (DST) - (SRC)$ .

Признаки:  $N := 1$ , если результат меньше 0, иначе  $N := 0$ ;

$Z := 1$ , если результат операции равен 0, иначе  $Z := 0$ ;

$V := 1$ , если произошло арифметическое переполнение, т. е. если операнды имели разные знаки, а знаки результата и источника совпадают, иначе  $V := 0$ ;

$C := 0$ , если был перенос из старшего разряда результата, иначе  $C := 1$ .

Описание: производится вычитание операнда-источника из операнда-приемника. Результат запоми-

нается по адресу приемника. Первоначальное содержимое приемника теряется.

**Умножение:**

MUL 070RSS.

Действие:  $R, RV1 \leftarrow (R) * (SRC)$ .

Признаки:  $N := 1$ , если произведение меньше 0, иначе  $N := 0$ ;

$Z := 1$ , если произведение равно 0, иначе  $Z := 0$ ;

$V := 0$ ;

$C := 1$ , если произведение меньше -2 или больше 1/2, иначе  $C := 0$ .

Описание: содержимое источника и регистра приемника перемножаются как целые числа в дополнительном коде. Результат запоминается в регистре приемника R (старшая часть) и следующем за ним регистре RV1 (младшая часть). Если номер регистра приемника нечетный, то в нем запоминается только младшая часть произведения.

**Деление:**

DIV 071RSS.

Действие:  $R, RV1 \leftarrow (R, RV1) / (SRC)$ .

Признаки:  $N := 1$ , если частное меньше 0, иначе  $N := 0$ ;

$Z := 1$ , если частное равно 0, иначе  $Z := 0$ ;

$V := 1$ , если  $(SRC) = 0$  или абсолютное значение содержимого регистра больше абсолютного значения содержимого источника; в этом случае выполнение команды прекращается, так как для представления частного требуется свыше 16 разрядов;

$C := 1$ , если делается попытка деления на 0, иначе  $C := 0$ .

Описание: 32-разрядное делимое в регистрах R (старшая часть) и следующем за ним RV1 (младшая часть) делится на 16-разрядный делитель (операнд-источник). Результат содержит 16-разрядное частное, запоминаемое в R, и 16-разрядный остаток, запоминаемый в RV1. Деление выполняется так, чтобы остаток имел такой же знак, как и делимое. Номер регистра R должен быть четным.

**Арифметический сдвиг (рис. 33, а):**

ASH 072RSS.

Действие:  $R \leftarrow (R)$ , сдвинутое арифметически на

NN разрядов вправо или влево, где NN — 6 младших разрядов операнда-источника.

Признаки:  $N := 1$ , если результат меньше 0, иначе  $N := 0$ ;

$Z := 1$ , если результат равен 0, иначе  $Z := 0$ ;

$V := 1$ , если знак операнда в процессе сдвига меняется, иначе  $V := 0$ ;

C загружается содержимым последнего разряда, выдвинутого из регистра.

Описание: содержимое выбранного регистра сдвигается вправо или влево на число разрядов, заданное в дополнительном коде в 6 младших значащих разрядах операнда источника (NN). Если число NN положительное, сдвиг производится влево, если отрицательное — вправо. Нулевое значение NN сдвигов не вызывает. Таким образом, при выполнении одной команды ASH имеется возможность передвинуть содержимое регистра на 32 позиции вправо или на 31 позицию влево.

**Арифметический сдвиг двойного слова (рис. 33, б):**

ASHC 073RSS.

Действие:  $R, RV1 \leftarrow (R, RV1)$ , сдвинутое арифметически на NN позиций вправо или влево, где NN — 6 младших разрядов операнда-источника.

Признаки:  $N := 1$ , если результат меньше 0, иначе  $N := 0$ ;

$Z := 1$ , если результат равен 0, иначе  $Z := 0$ ;

$V := 1$ , если знаковый разряд в процессе сдвига меняется, иначе  $V := 0$ ;

C загружается содержимым последнего разряда, выдвинутого из регистра.

Описание: содержимое регистров R с четным номером и следующего за ним RV1 обрабатывается, как 32-разрядное слово. Причем младшая часть слова (разряды 15..0) содержится в RV1, а старшая часть (разряды 31..16) — в R. Объединенное таким образом число сдвигается вправо или влево на количество позиций, определяемое числом в 6 младших разрядах операнда-источника (NN). Значение NN может изменяться в пределах от -32 до +31. Отрицательное значение NN определяет сдвиг вправо, положительное — влево. Если в команде указывается регистр с нечетным номером, то R и RV1 совпадают, т. е. обрабатывается содержимое одного регистра. В этом случае при указании сдвига вправо осуществляется циклический (кольцевой) сдвиг максимум на 16 разрядов.

**Исключающее «ИЛИ»:**

XOR 074RDD.

Действие:  $(DST) \leftarrow R + (DST)$ .

Признаки:  $N := 1$ , если результат меньше 0, иначе  $N := 0$ ;

$Z := 1$ , если результат равен 0, иначе  $Z := 0$ ;

$V := 0$ ;

C не изменяется.

Описание: производится операция «исключающее ИЛИ» (поразрядное сложение по модулю 2) над содержимым регистра и приемника, указанных в команде. Результат помещается по адресу приемника. Содержимое регистра R не изменяется.

**Команды ветвлений**

Старший байт команды ветвления содержит код операций, младший — 8-разрядное смещение со знаком (XXX), определяющее адрес ветвления от-

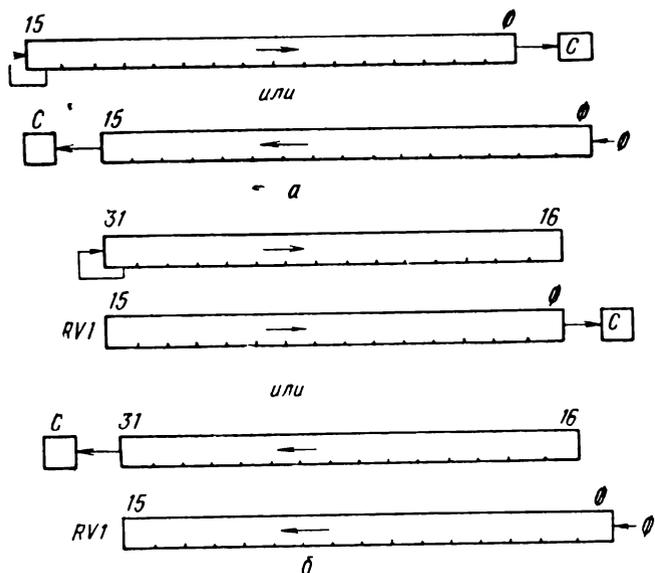


Рис. 33. Арифметический сдвиг (двухадресные команды): а — ASH; б — двойного слова ASHC

носителем счетчика команд РС. Ветвление выполняется в команде безусловного ветвления, причем состояние анализируемых признаков должно соответствовать условию, заданному в команде. Если условие не выполняется, то управление передается следующей по порядку команде.

При выполнении команды ветвления процессором анализируется логическое выражение, в котором в качестве переменных используются разряды признаков PSW. Команда вызывает ветвление по адресу, являющемуся суммой смещения, умноженного на 2, и текущего содержимого счетчика команд РС. Счетчик команд при вычислении адреса ветвления указывает на слово, следующее за командой ветвления.

Смещение показывает, на сколько ячеек нужно перейти относительно текущего содержимого РС в ту или другую сторону. Старший разряд смещения (разряд 7) является знаковым. Если он установлен в единицу (смещение отрицательное), ветвление происходит в сторону уменьшения адреса, в обратном направлении; если в седьмом разряде содержится нуль (смещение положительное), ветвление происходит в сторону увеличения адресов, в прямом направлении.

Так как смещение задано в словах, то для получения адреса ветвления смещение необходимо умножить на 2 перед прибавлением к РС, который определяет адрес ОП с учетом разряда, указывающего адрес байта в слове.

Восьмиразрядное смещение позволяет производить ветвление в обратном направлении максимально на 200 слов от слова, на которое указывает текущее содержимое РС, и на 177 слов в прямом направлении.

Список команд и условия ветвления приведены в табл. 42. Команды BHIS и BLO идентичны командам BCC и BCS соответственно. Дополнительная мнемоника введена для удобства программирования.

Действие:  $(PC) \leftarrow (PC) + 2 * XXX$ .

Признаки: не изменяются.

*Команды перехода и работы с подпрограммой*

**Безусловный переход** (см. рис. 28):

JMP 0001DD.

Действие:  $PC \leftarrow (DST)$ .

Признаки: не изменяются.

Описание: передает управление по адресу приемника. Формат команды аналогичен формату одноадресных команд. Адрес передачи управления определяется по тем же правилам, что и адрес операнда в одноадресных командах.

Прерывание: использование регистровой адресации вызывает прерывание программы через вектор 10 в пространстве данных режима «Ядро».

**Обращение к подпрограмме:**

JSR 004RDD.

Действие:  $TEMP \leftarrow (DST)$ ;  $-(SP) \leftarrow (R)$ ;  $R \leftarrow (PC)$ ;  $PC \leftarrow (TEMP)$ .

Описание: содержимое регистра связи R записывается в стек, в регистр связи записывается адрес возврата, выполняется переход по адресу приемника DST.

Прерывание: использование регистровой адресации вызывает прерывание программы через вектор 10 в пространстве данных режима «Ядро».

**Возврат из подпрограммы:**

RTS 00020R.

Мнемоника	Код операции	Название команды	Условия ветвления
BR	000400+XXX	Ветвление безусловное	—
BNE	001000+XXX	Ветвление, если не равно (нулю)	$Z = 0$
BEQ	001400+XXX	Ветвление, если равно (нулю)	$Z = 1$
BGE	002000+XXX	Ветвление, если больше или равно (нулю)	$N+V=0$
BLT	002400+XXX	Ветвление, если меньше (нуля)	$N+V=1$
BGT	003000+XXX	Ветвление, если больше (нуля)	$Z \vee (N+V)=0$
BLE	003400+XXX	Ветвление, если меньше или равно (нулю)	$Z \vee (N+V)=1$
BPL	100000+XXX	Ветвление, если плюс	$N=0$
BMI	100400+XXX	Ветвление, если минус	$N=1$
BHI	101000+XXX	Ветвление, если больше	$C \vee Z=1$
BLOS	101400+XXX	Ветвление, если меньше или равно	$C \vee Z=0$
BVC	102000+XXX	Ветвление, если нет арифметического переполнения	$V=0$
BVS	102400+XXX	Ветвление, если есть арифметическое переполнение	$V=1$
BCC	103000+XXX	Ветвление, если нет переноса	$C=0$
BHIS	103000+XXX	Ветвление, если больше или равно	$C=0$
BCS	103400+XXX	Ветвление, если перенос	$C=1$
BLO	103400+XXX	Ветвление, если меньше	$C=1$

Действие:  $PC \leftarrow (R)$ ;  $R \leftarrow -(SP)$ .

Признаки: не изменяются.

Описание: содержимое регистра связи (адрес возврата) передается в счетчик команд, из стека восстанавливается первоначальное содержимое этого регистра, выполняется переход по адресу возврата.

**Восстановление SP:**

MARK 0064NN.

Действие:  $SP \leftarrow (PC) + 2 * NN$ ;  $PC \leftarrow (R5)$ ;  $R5 \leftarrow (SP) +$ , где NN — число параметров.

Признаки: не изменяются.

Описание: сокращается стек на число  $2 * NN$  (где NN — число, записанное в коде команды) с выходом из подпрограммы. Команда используется в стандартной процедуре возврата из подпрограммы с очисткой стека. Действия, выполняемые командой, становятся понятными при рассмотрении следующего примера:

MOV R5, —(SP) — занести старое значение R5 в стек.

MOV P1, —(SP) — поместить NN параметров, передаваемых подпрограмме, в стек.

MOV P2, —(SP)

.

.

MOV PN, —(SP)

MOV#MARKNN, —(SP) — поместить команду MARKNN в стек.

MOV SP, R5 — записать адрес команды MARKNN в R5.

JSR PC, SUB — переход на подпрограмму.

В этот момент стек заполнен:

Старое значение R5

P1

P2

.

.

.

PNN

MARKNN

Старое значение PC.

Программа переходит на подпрограмму (на адрес SUB). В конце подпрограммы выполняется команда RTS R5 — возврат из подпрограммы; содержимое R5 пересылается в PC, который затем адресуется команду MARKNN. Старое содержимое PC пересылается в R5.

Команда MARKNN вызывает следующие действия:

указатель стека устанавливается на адресацию старого содержимого R5;

текущее содержимое R5 (старое значение PC) пересылается в PC;

старое содержимое R5 из стека переписывается в R5; этим завершается возврат из подпрограммы.

**Примечание.** При включенном диспетчере для выполнения команды MARK необходимо преобразовывать адреса стека как в пространстве команд, так и в пространстве данных.

**Вычитание единицы и ветвление:**

SOB 077RNN.

Действие:  $R \leftarrow (R) - 1$ ;  $PC \leftarrow (PC) - 2 * NN$ , если результат не равен 0;  $PC \leftarrow (PC)$ , если результат равен 0.

Признаки: не изменяются.

Описание: содержимое указанного в команде регистра уменьшается на 1. Если результат не равен 0, то из PC, который указывает адрес следующей команды, вычитается удвоенное смещение. Смещение интерпретируется как 6-разрядное положительное число и поэтому переход вперед невозможен. Команда удобна для организации циклов. Для этого программисту достаточно занести число повторений цикла в регистр, а в конце цикла поставить команду SOB.

### Команды прерываний

**Командное прерывание для системных программ:**

EMT 104000—104377.

Действие:  $-(SP) \leftarrow (PSW)$ ;  $-(SP) \leftarrow (PC)$ ;  $PC \leftarrow (30)$ ;  $PSW \leftarrow (32)$ .

Признаки: загружаются из вектора прерывания.

Описание: выполняется процедура прерывания через вектор с адресом 30. Код операции занимает старший байт кода команды. Младший байт может быть использован программистом для занесения любого кода. С ячейки, на которую командой передается управление, помещается программа, которая может анализировать младший байт команды. Эта программа в свою очередь может передавать управление одной из 256 возможных подпрограмм.

**Командное прерывание:**

TRAP 104400—104777.

Действие:  $-(SP) \leftarrow (PSW)$ ;  $-(SP) \leftarrow (PC)$ ;  $PC \leftarrow (34)$ ;  $PSW \leftarrow (36)$ .

Признаки: загружаются из вектора прерывания.

Описание: выполняется процедура прерывания через вектор с адресом 34. Команда аналогична EMT.

**Командное прерывание для отладки:**

BPT 000003.

Действие:  $-(SP) \leftarrow (PSW)$ ;  $-(SP) \leftarrow (PC)$ ;  $PC \leftarrow (14)$ ;  $PSW \leftarrow (16)$ .

Признаки: загружаются из вектора прерывания.

Описание: выполняется процедура прерывания через вектор с адресом 14. Команда состоит только из кода операции и используется при обращении к одной подпрограмме.

**Командное прерывание для ввода-вывода:**

IOT 000004.

Действие:  $-(SP) \leftarrow (PSW)$ ;  $-(SP) \leftarrow (PC)$ ;  $PC \leftarrow (20)$ ;  $PSW \leftarrow (22)$ .

Признаки: загружаются из вектора прерывания.

Описание: выполняется процедура прерывания через вектор с адресом 20. Команда состоит только из кода операции и используется при обращении к одной подпрограмме.

**Возврат из прерывания:**

RTI 000002.

Действие:  $PC \leftarrow (SP) +$ ;  $PSW \leftarrow (SP) +$ .

Признаки: загружаются из стека.

Описание: значения счетчика команд и регистра слова состояния восстанавливаются чтением из стека. В режимах «Пользователь» и «Супервизор» разряды PSW[7..5] не меняются. Если разряд PSW[4] установлен в единицу, после выполнения команды RTI происходит прерывание. Команда используется для возврата из программ, обслуживающих прерывания.

**Возврат из прерывания с запретом слежения:**

RTT 000006.

Действия:  $PC \leftarrow (SP) +$ ;  $PSW \leftarrow (SP) +$ .

Признаки: загружаются из стека.

Описание: эта команда по своему действию идентична команде RTI за исключением того, что при установке разряда PSW[4] прерывание происходит после выполнения первой команды, следующей за RTT.

### Команды изменения признаков

Разряды 3..0 команды изменения признаков указывают, какие разряды PSW должны быть установлены в единицу или сброшены в нуль. Если разряд 4 команды равен 1, производится установка в единицу разряда признака; если разряд 4 равен 0, разряд признака устанавливается в нуль.

NOP 000240 — нет операции.

CLC 000241 — очистка C.

Признаки: N, Z, V не изменяются, C := 0.

CLV 000242 — очистка V.

Признаки: N, Z, C не изменяются, V := 0.

CLZ 000244 — очистка Z.

Признаки: N, V, C не изменяются, Z := 0.

CLN 000250 — очистка N.

Признаки: Z, V, C не изменяются, N := 0.

CCC 000257 — очистка всех признаков (N, Z, V, C).

Признаки: N, Z, V, C := 0.

SEC 000261 — установка C.

Признаки: N, Z, V не изменяются, C := 1.

SEV 000262 — установка V.

Признаки: N, Z, C не изменяются, V := 1.

SEZ 000264 — установка Z.

Признаки: N, V, C не изменяются, Z := 1.

SEN 000270 — установка N.

Признаки: Z, V, C не изменяются, N := 1.

SCC 000277 — установка всех признаков (N, Z, V, C).

Признаки: N, Z, V, C: =1.

*Остальные команды управления программой*

**Останов:**

HALT 000000.

Признаки: не изменяются.

Описание: выполнение команды зависит от текущего режима работы процессора и установленного в регистре MR варианта отработки команды HALT. По данной команде в режиме «Ядро» при MR[3]=0 процессор переходит в режим эмулятора пульта, при MR[3]=1 происходит прерывание по вектору с адресом 4 (при этом устанавливается в единицу разряд 7 регистра CPUERR). В режимах «Пользователь» и «Супервизор» команда вызывает прерывание по вектору с адресом 4 (в регистре CPUERR также устанавливается в единицу разряд 7).

Прерывание: в режиме «Ядро» при MR[3]=1, а также в режимах «Пользователь» и «Супервизор» команда HALT вызывает прерывание по вектору с адресом 4.

**Ожидание:**

WAIT 000001.

Признаки: не изменяются.

Описание: выполнение команды WAIT зависит от режима работы процессора. В режиме «Ядро» прекращается извлечение команды из памяти. МПИ свободен и процессор ожидает прерывания от внешних устройств. При прерывании запоминается адрес следующей за WAIT команды и после окончания работы по прерывающей программе продолжается основная программа. При выполнении этой команды достигается максимальная скорость обмена информацией между памятью и внешними устройствами на МПИ, свободном от запросов процессора. В режимах «Пользователь» и «Супервизор» WAIT выполняется как NOP.

**Сброс внешних устройств:**

RESET 000005.

Описание: в режиме «Ядро» все устройства на МПИ устанавливаются в исходное состояние. В режимах «Пользователь» и «Супервизор» RESET выполняется как NOP.

**Запись типа процессора:**

MFPT 000007.

Действие: R0←N.

Признаки: не изменяются.

Описание: в результате выполнения команды в регистр R0 записывается число N. Команда используется для сообщения системному матобеспечению типа используемого микропроцессора. Для процессора CM 1425 N=5.

**Установка уровня приоритета:**

SPL 00023N.

Действие: PSW[7...5]←N.

Признаки: не изменяются.

Описание: в режиме «Ядро» три младших разряда команды записываются в разряды приоритета PSW. В режимах «Супервизор» и «Пользователь» эта команда выполняется как NOP.

**Вызов супервизора:**

CSM 0070DD.

Действие: если MMR3[03]=1 и текущий режим не режим «Ядро», то SP режима «Супервизор»←(SP) текущего режима; TEMP[15...4]←PSW[15...4]; TEMP[3...0]←0; PSW[13, 12]←PSW[15, 14]; PSW[15, 14]←01; PSW[4]←0; -(SP)←(TEMP); -(SP)←(PC); -(SP)←(DST); PC←(10).

Признаки: не изменяются.

Описание: команда выполняется в режимах «Пользователь» или «Супервизор». В режиме «Ядро» это несуществующая команда. По команде CSM указатель стека текущего режима записывается в указатель стека режима «Супервизор». Производятся переключение в режим «Супервизор», запоминание трех слов (PSW с очищенными разрядами признаков, PC и содержимого адреса приемника) в стеке режима «Супервизор», обнуление разряда PSW[4], запись в PC содержимого ячейки 10 (в режиме «Супервизор»).

Вызванная программа в режиме «Супервизор» может вернуться к вызывающей программе, выбрав операнд из стека и выполнив команду RTI. При возврате признаки будут определяться содержимым регистра PSW в стеке. Следовательно, вызванная программа в режиме «Супервизор» может управлять значением признаков при ее завершении.

Прерывание: в режиме «Ядро» это несуществующая команда с обычным прерыванием по вектору 10. При MMR3[3]=0—это несуществующая команда для всех режимов.

Приложение 2

## СИСТЕМА КОМАНД С ПЛАВАЮЩЕЙ ЗАПЯТОЙ

Система команд с плавающей запятой (ПЗ) включает полный набор команд с ПЗ, а также команды преобразования целых чисел с ПЗ. Команды с ПЗ используют шесть универсальных накопителей — аккумуляторы (AC0...AC5) для вычисления, хранения и передачи данных в универсальные регистры или ОП. Каждый аккумулятор в зависимости от типа команд и слова состояния может рассматриваться как 64- или 32-разрядный. Если выполняются команды с 32-разрядными операндами, то используются только старшие 32 разряда, младшие разряды остаются без изменения. Команды с ПЗ могут адресовать аккумуляторы ПЗ, регистры общего назначения или ячейки оперативной памяти. Аккумуляторы доступны только командам с ПЗ.

Обозначения, применяемые в описании команд с ПЗ:

(XXXX) — содержимое XXXX;

A := (B) — A присваивается значение содержимого B;

<= — меньше или равно;

< — меньше;

> — больше;

E(XXXX) — характеристика (экспонента) содержимого XXXX;

∨ — функция «ИЛИ»;

AB(XXXX) — абсолютное значение содержимого XXXX;

\*\* — возведение в степень;

\* — умножение.

## ФОРМАТЫ ДАННЫХ С ПЗ

Математические числа с ПЗ определяются формулой  $(2^{**k}) * F$ ,  $k$  — порядок, являющийся целым числом;  $F$  — мантисса, являющаяся дробным числом. Значение мантиссы определяется условием  $1/2 \leq F < 1$ . Форматы данных с ПЗ являются производными этого математического представления. Имеются два формата данных с ПЗ (рис. 34).

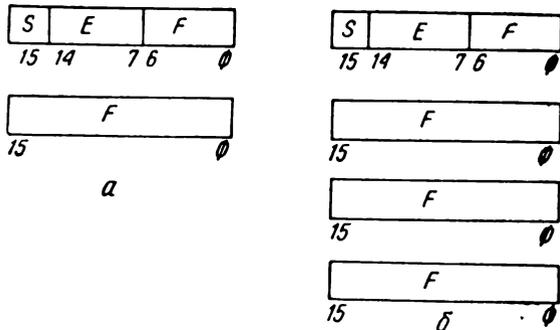


Рис. 34. Формат данных с ПЗ:  
а — F, обычная точность; б — D, высокая точность

Число формата обычной точности (F) содержит 32 двоичных разряда, а число формата высокой точности (D) — 64 разряда. Форматы различаются только длиной мантиссы. Старший разряд в обоих форматах отведен под знак числа (разряд S).

Для указания порядка служит характеристика, которая формируется путем сложения порядка со знаком и числа 200, т. е. представляет порядок с избытком 200. Характеристика, равная 0, отведена для представления нуля ПЗ, остальные значения характеристики от 1 до 377 представляют значения порядка от  $-177$  до  $+177$  (рис. 35).

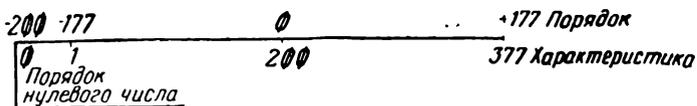


Рис. 35. Представление порядка чисел с ПЗ

Дробная часть нулевого числа ПЗ всегда нормализована, т. е. ее старший двоичный разряд равен 1. Этот разряд называется «скрытым» и при хранении в памяти чисел с ПЗ всегда опускается. Мантисса без «скрытого» разряда расположена в оставшихся разрядах числа ПЗ — 23 разрядах формата F и 55 разрядах формата D. При вычислениях «скрытый» разряд восстанавливается в качестве старшего разряда мантиссы, т. е. мантисса в арифметических операциях имеет разрядность 24 и 56 разрядов соответственно.

Как для положительных, так и для отрицательных значений чисел мантисса представляется в прямом коде. Различие в знаке числа отражается на значении знакового разряда S:  $S=0$  соответствует положительным числам,  $S=1$  — отрицательным.

Так как числа с ПЗ хранятся в памяти без «скрытого» разряда, невозможно различить числа, мантиссы которых равна 0 и  $1/2$ . Поэтому для представления нулевого числа с ПЗ используется характеристика, равная 0. Она может возникнуть в арифметических операциях как особый случай переполнения или исчезновения порядка. Порядок ре-

зультата считается переполненным, если его значение превышает 177. Исчезновение порядка характеризуется значением его, меньшим  $-177$ . Если порядок результата равен  $-200$ , что является особым случаем исчезновения порядка, то характеристика результата равна 0. Характеристика результата, равная 0, получается также, если при переполнении порядка полная характеристика равна 400 (порядок равен 200). При этом в качестве результата записываются младшие 8 разрядов характеристики, равные 0.

Неопределенным числом называется отрицательный ноль, т. е. число, знаковый разряд которого равен 2, характеристика — 0, а мантисса представлена любым кодом. Если в арифметической операции в качестве операнда из оперативной памяти выбирается неопределенное число, то оно или обрабатывается как истинный ноль, или вызывает прерывание программы. Выборка неопределенного числа из аккумулятора в качестве операнда никогда не вызывает прерывания.

## ФОРМАТЫ ЦЕЛЫХ ЧИСЕЛ

Система команд с ПЗ обеспечивает преобразование чисел с плавающей запятой в числа с фиксированной запятой и наоборот. Система команд с ПЗ различает числа с ФЗ короткого формата, содержащие 16 разрядов (формат I), и числа длинного формата, содержащие 32 разряда (формат L).

На рис. 36 показаны форматы чисел с ФЗ, используемых при работе с командами ПЗ (S — знак числа).

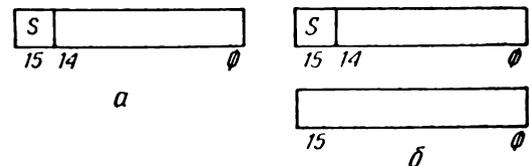


Рис. 36. Формат чисел с ФЗ, используемых в командах с ПЗ:

а — короткий I; б — длинный L

## СЛОВО СОСТОЯНИЯ ПЗ (FPSW)

FPSW представлено 16-разрядным словом и обеспечивает режим выполнения операции с ПЗ, управление прерыванием программы с ПЗ, сохранение признаков, формируемых командами с ПЗ. На рис. 37 показан формат FPSW. Содержимое разрядов следующее.

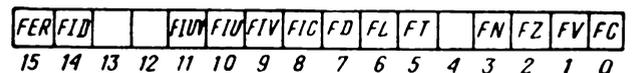


Рис. 37. Формат FPSW

FC — признак переноса ПЗ. FC устанавливается в единицу, если команда формирует перенос из старшего значащего разряда результата, иначе — сбрасывается. FC может быть установлен только в командах преобразования чисел с ПЗ в целые числа с ФЗ.

FV — признак переполнения порядка результата ПЗ. FV устанавливается в единицу, если команда вызывает переполнение порядка, иначе — сбрасывается.

FZ — признак нулевого результата ПЗ. FZ устанавливается в единицу, если характеристика результата равна 0, иначе — сбрасывается.

FN — признак отрицательного результата ПЗ. FN устанавливается в единицу, если знак результата отрицательный, иначе — сбрасывается.

FT — признак запрета округления результата ПЗ. Если FT установлен, то результат не округляется, иначе — округляется.

FL — формат чисел с ФЗ. Если FL установлен, задается длинный формат, если сброшен — короткий.

FD — формат чисел с ПЗ. Если FD установлен, то используется высокая точность, иначе — обычная.

FIC — маска прерывания по ошибке преобразования числа с ПЗ в число с ФЗ. Если ошибка имеет место, то при FIC=1 происходит прерывание программы. При FIC=0 прерывание отсутствует. В обоих случаях результату присваивается значение истинного нуля. Команда преобразования дает ошибку, если она формирует число с большим количеством разрядов, чем может поместиться в результате, формат которого определяется разрядом FL.

FIV — маска прерывания по переполнению порядка результата ПЗ. Если имеет место переполнение порядка и FIV=1, то происходит прерывание программы. Записанная характеристика результата на 400 меньше истинной. Мантисса результата является правильной. Если переполнение порядка возникает при FIV=0, то прерывание отсутствует, а в качестве результата записывается истинный нуль. Особые случаи переполнения рассмотрены при описании команд MOD и LDEXP.

FIU — маска прерывания по исчезновению порядка результата ПЗ. Если имеет место исчезновение порядка и FIU=1, то происходит прерывание программы. Записанная характеристика результата на 400 больше истинной, исключая случай равенства истинной характеристики 0 (записываемая характеристика также равна 0, т. е. правильна). Мантисса результата является правильной. Если исчезновение порядка возникает при FIU=0, то прерывание отсутствует, а в качестве результата записывается истинный нуль.

FIUV — маска прерывания по неверным данным. Если из оперативной памяти в качестве операнда команд сложения, вычитания, сравнения, деления, умножения, умножения и выделения целого числа, загрузки с преобразованием формата числа с ПЗ, загрузки, проверки и установки, записи положительной, записи дополнения принимается неопределенное число, т. е. отрицательный нуль, то при FIUV=1 возникает прерывание программы. Прерывание происходит перед выполнением перечисленных выше команд, исключая команды проверки и установки, записи положительной и записи дополнения, для которых прерывание происходит после выполнения команд. Заметим, что прерывание никогда не возникает, если неопределенное число принимается из аккумулятора, т. е. прерывание никогда не возникает в режиме 0. Если FIUV=0, то прерывание отсутствует, а в качестве операнда вместо неопределенного числа используется истинный нуль.

FID — разряд запрета прерывания. Если FID=1, то все прерывания ПЗ запрещены. Однако для тех программных нарушений, у которых

индивидуальная маска отсутствует или равна 1, только собственно прерывание подавляется, т. е. отсутствует переход на программу обработки прерываний. Все другие действия, связанные с прерыванием, выполняются. FID используется в основном для целей наладки. Нормально FID находится в нулевом состоянии. При описании обработки программных нарушений подразумевается, что FID=0.

FER — признак ошибки ПЗ. FER устанавливается в единицу, если имеет место программное нарушение, а разряд индивидуальной маски для этого нарушения или отсутствует, или разрешает прерывание, т. е. равен 1. Установка разряда FER не зависит от состояния FID. Будучи установлен какой-либо командой с ПЗ, FER может быть очищен только командами LDFPS (загрузка FPSW) и RESET. Установка признаков в разрядах FPSW [3...0] и признака ошибки производится командами ПЗ в процессе их выполнения. Любой из разрядов режима (FPSW[7...5]) и управления прерыванием (FPSW[11...8]) может быть установлен пользователем командой LDFPS.

#### ОБРАБОТКА ПРОГРАММНЫХ ПРЕРЫВАНИЙ ПЗ

Необычные ситуации, возникающие при выполнении команд с ПЗ, называются программными нарушениями. Их причиной являются неправильное указание или использование операндов и команд, а также получение особых результатов.

Существует шесть программных нарушений при выполнении команд с ПЗ: неверные данные, т. е. наличие в памяти неопределенного числа; исчезновение порядка; переполнение порядка; ошибка преобразования из ПЗ в ФЗ; попытка деления на 0; несуществующие коды операций или несуществующий режим адресации ПЗ.

Программные нарушения вызывают прерывания программы. Прерывания по первым четырем программным нарушениям могут быть индивидуально замаскированы, т. е. запрещены, наличием 0 в соответствующем разряде 4-разрядной программной маски, содержащейся в FPSW[11...8]. Для двух остальных программных нарушений индивидуальной маски нет. Единичное значение разряда FID в FPSW запрещает прерывание по всем шести программным нарушениям. Такое маскирование всех прерываний применяется в основном для целей отладки.

Если соответствующее программное нарушение не замаскировано индивидуальной маской или не имеет программной маски, то устанавливается в единицу признак ошибки — разряд FER FPSW и код причины прерывания заносится в 4-разрядный регистр кода прерываний FEC. Кроме того, адрес команды, вызвавшей прерывание, заносится в регистр адреса прерывания FEA. Перечисленные действия выполняются даже тогда, когда переход на программу обработки прерываний, т. е. собственно прерывание, подавляется единичным значением разряда FID FPSW. Регистры FEC и FEA командой STST (запись состояния регистров прерываний) переписываются в оперативную память. Заметим, что нет специальных команд для занесения в эти регистры заранее сформированной информации. В табл. 43 даны коды программных прерываний, заносимые в FEC.

Таблица 43

Программное нарушение	Восьмеричный код прерывания
Несуществующий код команды или несуществующий режим адресации	2
Деление на нуль	4
Ошибка преобразования из ПЗ в ФЗ	6
Переполнение порядка	10
Исчезновение порядка	12
Неверные данные	14

Обработка всех программных прерываний, вызываемых командами с ПЗ, производится через вектор с адресом 244. Далее при описании команд с ПЗ программные прерывания будут описаны только для тех команд, где они могут иметь место.

#### ОКРУГЛЕНИЕ И ПОГРЕШНОСТЬ РЕЗУЛЬТАТА

Для команд с ПЗ, выполняющих действия над операндами с ПЗ, мантисса промежуточного результата в общем случае имеет большую разрядность, чем мантисса конечного результата, разрядность которой определяется FD. Младшие разряды нормализованной мантиссы промежуточного результата, расположенные правее правой границы мантиссы конечного результата, при формировании последней теряются, если FT=1. Если FT=0, то производится округление мантиссы результата, что позволяет учесть значение старшего теряемого разряда.

Округление состоит в прибавлении 1 к старшему теряемому разряду, называемому разрядом округления, и распространении переноса. Если разряд округления равен 1, то мантисса конечного результата после округления оказывается больше на единицу младшего значащего разряда, чем мантисса неокругленного результата. Если разряд округления равен 0, мантиссы округленного и неокругленного результатов равны.

Погрешность результата не превышает значения наименьшего значащего разряда без округления и 1/2 наименьшего значащего разряда с округлением. Команда считается точной, если при формировании результата команды отсутствует потеря значащих разрядов. Далее при описании команд с ПЗ точность результата будет описана только для тех команд, где может иметь место погрешность результата.

#### ФОРМАТЫ КОМАНД С ПЗ

Команды с ПЗ имеют длину в одно слово и представляются одним из пяти форматов F1...F5, приведенных на рис. 38.

Поля форматов команд интерпретируются следующим образом:

OC — код команд с ПЗ, равен 17;

FOC — код команды, определяющий конкретную команду с ПЗ;

SRC — код, определяющий источник числа с ФЗ, интерпретируется как в системе команд с ФЗ;

DST — код, определяющий приемник числа с ФЗ, интерпретируется как в системе команд с ФЗ;

AC — код, определяющий один из четырех аккумуляторов ПЗ AC0...AC3;

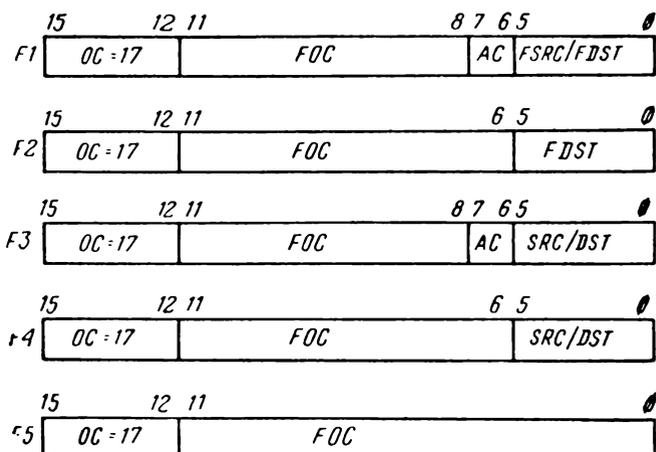


Рис. 38. Форматы команд с ПЗ

FSRC — код, определяющий источник операнда с ПЗ;

FDST — код, определяющий приемник числа с ПЗ.

Использование кодов FSRC и FDST аналогично использованию кодов SRC или DST соответственно, за исключением режима 0. В режиме 0 разряды 2...0 полей FSRC и FDST определяют один из шести аккумуляторов AC0...AC5, а не регистров общего назначения. Наличие в этих разрядах кодов 6 и 7 определяет обращение к несуществующим аккумуляторам и в этом случае выполняется прерывание по несуществующему режиму адресации. В режимах автоувеличения и автоуменьшения содержимое регистра изменяется на 4, если используются данные формата F, и на 10 для данных формата D. При непосредственной адресации (режим 2, RD=7) только 16 разрядов запоминаются или загружаются.

Команды с ПЗ бывают двухадресными, одноадресными и безадресными. В двухадресных командах формата F1, использующих один операнд, адрес его определяется одним из полей AC или FSRC, а адрес результата — оставшимся полем. В командах формата F1, использующих два операнда, результат записывается в аккумулятор. Команды формата F2 используют всегда один операнд, результат записывается по тому же адресу. В двухадресных командах формата F3, использующих один операнд, адрес операнда определяется одним из полей AC или PC, а результат записывается по адресу, определяемому другим полем. В командах формата F4 поле SRC (DST) задает адрес источника или приемника операнда с ФЗ. В этих командах содержимое источника помещается в один из регистров ПЗ или содержимое регистров ПЗ записывается по адресу приемника. Команды формата F5 или задают режим работы ПЗ путем установки соответствующих разрядов FPSW, или осуществляют перепись признаков ПЗ в соответствующие разряды слова состояния процессора (PSW).

Имеется 46 команд с ПЗ. Команда определяется кодом полей OC и FOC и разрядами форматов данных FD и FL. Так, например, один и тот же код полей OC и FOC, равный 1111 001 0, задает две команды умножения ПЗ: MULF — умножение обычной точности при FD=0 и MULD — умножение высокой точности при FD=1 (индекс F в мнемонике команд ПЗ означает обычную точность, индекс D — высокую точность). Аналогично для команд,

выполняющих действия над числами с ФЗ, разряд FL определяет короткий формат (индекс I в мнемонике команды) и длинный формат (индекс L).

При восьмеричном изображении команды в общем виде конкретные коды полей AC, FSRC/FDST, SRC/DST не приводятся. Например, команда умножения обычной точности MULF изображается как 171 AC FSRC. Единица восьмого разряда в командах форматов F1 и F3 изображается совместно с двухразрядным полем AC следующим образом: AC+4. Так, команда загрузка экспоненты LDEXP имеет изображение 176 (AC+4) SRC.

#### Команды загрузки:

LDF — загрузка обычной точности;

LDD — загрузка высокой точности.

Код команды: 172 (AC+4) FSRC, формат F1.

Действие:  $AC := (FSRC)$ .

Признаки: FC:=0;

FV:=0;

FZ:=1, если результат равен 0, ина-

че FZ:=0;

FN:=1, если результат меньше 0,

иначе FZ:=0;

FZ:=1, FN:=1, если (FSRC)=-0.

Описание: содержимое FSRC загружается в AC.

Прерывание: несуществующий режим адресации. Если из оперативной памяти выбран неопределенный операнд —0 и FIUV=1, то выполняется прерывание по неверным данным перед выполнением команды. Содержимое AC не меняется. Если FIUV=0, то прерывание не выполняется и —0 записывается в AC. В этом случае признаки всегда соответствуют неопределенному операнду —0 независимо от значения FIUV.

#### Команды загрузки с преобразованием формата числа с ПЗ:

LDCDF — загрузка с преобразованием из высокой точности в обычную;

LDCFD — загрузка с преобразованием из обычной точности в высокую.

Код команды: 177 (AC+4) FSRC, формат F1.

Действие:  $AC := CXY(FSRC)$ , где CXY означает преобразование из формата X в формат Y, причем  $X=D$ ,  $Y=F$ , если  $FD=0$ , и  $X=F$ ,  $Y=D$ , если  $FD=1$ ; если  $E(FSRC)=0$ , то  $AC:=0$ ; если  $FD$ ,  $FT$  и  $FIV$  — нулевые и округление в команде LDCDF вызвало переполнение порядка, то  $AC:=0$ .

Признаки: FC:=0;

FV:=1, если произошло переполнение, иначе FV:=0;

FZ:=1, если результат равен 0, ина-

че FZ:=0;

FN:=1, если результат меньше 0,

иначе FN:=0;

FZ:=1, FN:=1, если (FSRC)=-0.

Описание: если  $FD=0$ , то операнд высокой точности, содержащийся по адресу, определяемому FSRC, преобразуется в операнд обычной точности. Если  $FT=0$ , то результат преобразования перед записью в AC округляется. При  $FD=1$  операнд обычной точности загружается в старшую часть AC, а младшая часть AC очищается. Если преобразуемый операнд — неопределенное число —0, то признаки устанавливаются по его значению независимо от FIUV.

Прерывание: несуществующий режим адресации. Прерывание по неверным данным, которое

происходит перед выполнением команды. Переполнение порядка в команде LDCDF, если выполняется округление и  $FIV=1$ .

Точность: LDCFD — точная команда.

#### Команды сложения:

ADDF — сложение обычной точности;

ADDD — сложение высокой точности.

Код команды: 172 AC FSRC, формат F1.

Действие:  $(AC) + ((FSRC)) = SUM$ , где SUM — сумма. Если произошло исчезновение порядка и  $FIV=0$ , то  $AC:=0$ . Если произошло переполнение порядка и  $FIV=0$ , то  $AC:=0$ . Во всех остальных случаях  $AC:=SUM$ .

Признаки: FC:=0;

FV:=1, если произошло переполнение порядка, иначе FV:=0;

FZ:=1, если результат равен 0, иначе FZ:=0;

FN:=1, если результат меньше 0, иначе FN:=0.

Описание: содержимое FSRC складывается с содержимым AC и результат помещается в AC. В зависимости от FD операция выполняется над числами обычной ( $FD=0$ ) или высокой ( $FD=1$ ) точности. Результат нормализуется. Если  $FT=0$ , то результат округляется. Если произошло переполнение или исчезновение порядка и эти нарушения замаскированы соответствующими разрядами FPSW, то в AC записывается абсолютный нуль. Если прерывания разрешены, то в AC записывается полученный результат. При этом в случае исчезновения порядка записанная характеристика на 400 больше истинной, в случае переполнения — на 400 меньше истинной, за исключением случая нулевой характеристики, которая является истинной. Мантисса имеет правильное значение.

Прерывание: несуществующий режим адресации; прерывание по неверным данным перед выполнением команды; переполнение порядка; исчезновение порядка.

Точность: погрешность результата, получаемая при переполнении или исчезновении порядка, определена в описании команды. Для операндов с противоположными знаками и разностью порядков, равной 0, результат получается точным.

#### Команды вычитания:

SUBF — вычитание обычной точности;

SUBD — вычитание высокой точности.

Код команды: 173 AC FSRC, формат F1.

Действие:  $(AC) - (FSRC) = DIFF$ , где DIFF — разность. Если произошло переполнение порядка и  $FIV=0$ , то  $AC:=0$ . Если произошло исчезновение порядка и  $FIV=0$ , то  $AC:=0$ . Во всех остальных случаях  $AC:=DIFF$ .

Признаки: FC:=0;

FV:=1, если произошло переполнение порядка, иначе FV:=0;

FZ:=1, если результат равен 0, иначе FZ:=0;

FN:=1, если результат меньше 0, иначе FN:=0.

Описание: содержимое FSRC вычитается из содержимого AC. Вычитание выполняется обычной ( $FD=0$ ) или высокой точности ( $FD=1$ ), с округлением ( $FT=0$ ) или без него ( $FT=1$ ). Нормализованный результат записывается в AC. Если имеет место переполнение или исчезновение порядка и соответствующее прерывание разрешено программной

маской FPSW, то записываемая в АС характеристика больше истинной на 400 для случая исчезновения и меньше истинной на 400 для случая переполнения порядка. Нулевая характеристика записывается в АС точно. Если при исчезновении или переполнении порядка прерывания запрещены нулевым значением программной маски, то в АС записывается абсолютный нуль.

Прерывание: несуществующий режим адресации; прерывание по неверным данным перед выполнением команды; переполнение порядка; исчезновение порядка.

Точность: погрешность результата при переполнении или исчезновении порядка определена в описании команды. Для операндов с одинаковыми знаками и разностью порядков, равной 0, результат получается точным.

#### Команды умножения:

MULF — умножение обычной точности;

MULD — умножение высокой точности.

Код команды: 171 АС FSRC, формат F1.

Действие:  $(AC) * (FSRC) = PROD$ , где PROD — произведение. Если произошло переполнение порядка и FIV=0, то AC:=0. Если произошло исчезновение порядка и FIU=0, то AC:=0. Во всех остальных случаях AC:=PROD.

Признаки: FC:=0;

FV:=1, если произошло переполнение порядка, иначе FV:=0;

FZ:=1, если результат равен 0, иначе FZ:=0;

FN:=1, если результат меньше 0, иначе FN:=0.

Описание: содержимое АС умножается на содержимое FSRC и результат записывается в АС. Если характеристика любого операнда равна 0, то в АС записывается абсолютный нуль. Умножение заключается в сложении характеристик и умножении мантисс. Сумма характеристик первого и второго операндов, уменьшенная на 200, является характеристикой промежуточного результата. Так как мантиссы операндов нормализованы, то мантисса промежуточного результата нормализована или денормализована на один разряд. Мантисса промежуточного результата содержит 48 разрядов для формата F и 56 разрядов для формата D. Денормализованная промежуточная мантисса нормализуется, а характеристика промежуточного результата в этом случае уменьшается на 1. Мантисса результата формируется длиной 24 и 56 разрядов для форматов F и D соответственно. При FT=0 результат округляется. Если произошло переполнение или исчезновение порядка и соответствующее нарушение замаскировано, то в АС записывается абсолютный нуль. Если прерывание разрешено, то при переполнении порядка характеристика результата становится меньше истинной на 400, а при исчезновении порядка — больше истинной на 400, за исключением нулевой характеристики, которая является истинной. Результат, равный неопределенному числу, может получиться только в результате переполнения или исчезновения порядка. Он записывается в АС только если соответствующее прерывание разрешено.

Прерывание: несуществующий режим адресации; прерывание по неверным данным перед выполнением команды; переполнение порядка; исчезновение порядка.

Точность: погрешность результата при перепол-

нении или исчезновении порядка определена в описании команды.

#### Команды умножения и выделения целого числа:

MODF — умножение и выделение числа обычной точности;

MODD — умножение и выделение числа высокой точности.

Код команды: 171 (AC+4) FSRC, формат F1.

Действие:  $PROD = (AC) * (FSRC) = N + G$ , где PROD — произведение. Результат умножения двух чисел с ПЗ получается в виде  $ABS(PROD) = 2^k * F$ , где  $1/2 \leq F < 1$  — мантисса результата;  $(200+k)$  — характеристика результата. Части N и G записываются как числа с плавающей запятой.

Признаки: FC:=0;

FV:=1, если произошло переполнение порядка, иначе FV:=0;

FZ:=1, если результат равен 0, иначе FZ:=0;

FN:=1, если результат меньше 0, иначе FN:=0.

Описание: команда выполняет умножение двух чисел с ПЗ, разделяет результат на целую (N) и дробную (G) части и записывает одну или обе части как числа с ПЗ. N и G имеют тот же знак, что и произведение. N записывается в нечетный аккумулятор ACV1, а G — в аккумулятор АС. Таким образом, если в поле АС указан нечетный аккумулятор, то запоминается только G, а N терется. Если в поле АС указан четный аккумулятор, то запоминаются и N и G. При выполнении команды возможны следующие случаи:

если  $N > 2^L$  ( $L$  — разрядность мантиссы операндов) и переполнения нет, то в ACV1 записывается N, обрезанное до L разрядов, а в АС записывается абсолютный нуль;

если  $1 \leq N \leq 2^L$ , то в ACV1 записывается N, а в АС — G после нормализации и округления согласно значению FT;

если  $N < 1$  и нет исчезновения, то в ACV1 записывается абсолютный нуль, а в АС — G;

если есть переполнение порядка и FIV=1, то N обрезается до L разрядов, а в АС записывается абсолютный нуль. Если FIV=0, то в ACV1 и в АС записывается абсолютный нуль;

если есть исчезновение порядка, то при FIU=1 в ACV1 записывается абсолютный нуль, а в АС — G. Если FIU=0, то при исчезновении порядка в ACV1 и АС записывается абсолютный нуль.

Прерывание: несуществующий режим адресации; прерывание по неверным данным перед выполнением команды; переполнение порядка; исчезновение порядка.

Точность: если есть исчезновение или переполнение порядка и соответствующее прерывание разрешено, то при записи характеристика результата оказывается увеличенной на 400 при исчезновении и уменьшенной на 400 при переполнении. Если  $N > 2^L$ , то ошибка целой части равна наименьшему значащему разряду. Если  $1 \leq N \leq 2^L$  или  $N < 1$  и нет исчезновения порядка, то погрешность результата не превышает значения наименьшего значащего разряда без округления и  $1/2$  наименьшего значащего разряда с округлением.

#### Команды деления:

DIVF — деление обычной точности;

DIVD — деление высокой точности.

Код команды: 174 (AC+4) FSRC, формат F1.

Действие:  $(AC)/(FSRC) = QUOT$ , где QUOT —

частное. Если  $E(FSRC) = 0$ , то  $AC := (AC)$ ; команда не выполняется. Если  $E(AC) = 0$ , то  $AC := 0$ . Если произошло исчезновение порядка и  $FIU = 0$ , то  $AC := 0$ . Если произошло переполнение порядка и  $FIV = 0$ , то  $AC := 0$ . Во всех остальных случаях  $AC := QUOT$ .

Признаки:  $FC := 0$ ;  
 $FV := 1$ , если произошло переполнение, иначе  $FV := 0$ ;  
 $FZ := 1$ , если  $E(AC) = 0$ , иначе  $FZ := 0$ ;  
 $FN := 1$ , если результат меньше 0, иначе  $FN := 0$ .

Описание: содержимое  $AC$  делится на содержимое  $FSRC$  и результат записывается в  $AC$ . В зависимости от состояния  $FD$  операция выполняется над числами обычной точности ( $FD = 0$ ) или высокой точности ( $FD = 1$ ). Деление заключается в вычитании характеристик и делении мантисс. Сумма числа 200 и разности характеристик делимого и делителя является характеристикой промежуточного результата. Так как мантиссы исходных операндов всегда нормализованы, то мантисса частного оказывается всегда нормализованной или переполненной. В последнем случае мантисса результата нормализуется, а характеристика промежуточного результата увеличивается на 1. Если  $FT = 0$ , то результат округляется. Полученный результат всегда записывается в  $AC$ . При переполнении и исчезновении порядка, когда соответствующие прерывания запрещены ( $FIV = 0$ ,  $FIU = 0$ ), в  $AC$  записывается абсолютный нуль. Если содержимое  $AC$  равно 0, то в  $AC$  заносится абсолютный нуль. В случае, если содержимое  $FSRC$  равно 0, команда не выполняется, в  $FEC$  заносится код 4 и выполняется прерывание по попытке деления на 0.

Прерывание: несуществующий режим адресации; прерывание по неверным данным перед выполнением команды; деление на 0; переполнение порядка; исчезновение порядка.

Точность: погрешность результата, получаемая при переполнении или исчезновении порядка, определена в описании команды. Во всех остальных случаях погрешность результата определяется как значение наименьшего значащего разряда без округления и  $1/2$  наименьшего значащего разряда с округлением.

#### Команды сравнения:

$CMPI$  — сравнение обычной точности;

$CMPIH$  — сравнение высокой точности.

Код команды: 173 ( $AC+4$ )  $FSRC$ , формат  $F1$ .

Действие:  $(FSRC) - (AC)$ .

Признаки:  $FC := 0$ ;  
 $FV := 0$ ;  
 $FZ := 1$ , если  $(FSRC) - (AC) = 0$ , иначе  $FZ := 0$ ;  
 $FN := 1$ , если  $(FSRC) - (AC) < 0$ , иначе  $FN := 0$ .

Описание: сравнивается содержимое  $FSRC$  и  $AC$  и устанавливаются соответствующие признаки. Операнды не изменяются. Если знаки операндов не равны, то признаки устанавливаются по значению знаков. Если знаки равны, то сравнение производится путем вычитания содержимого  $AC$  из содержимого  $FSRC$ , анализа полученного результата и установки признаков. В случае, когда оба операнда нулевые, в  $AC$  записывается абсолютный нуль.

Прерывание: прерывание по неверным данным перед выполнением команды; несуществующий режим адресации.

#### Команды очистки:

$CLR$  — очистка обычной точности;

$CLRH$  — очистка высокой точности.

Код команды: 1704  $FDST$ , формат  $F2$ .

Действие:  $FDST := 0$ .

Признаки:  $FC := 0$ ;

$FV := 0$ ;

$FZ := 1$ ;

$FN := 0$ .

Описание: в зависимости от состояния разряда  $FD$  производится запись нулей по адресу  $FDST$ . Если  $FD = 1$ , обнуляются 64 разряда. В случае использования непосредственного режима адресации обнуляются только 16 разрядов.

Прерывание: несуществующий режим адресации.

#### Команды проверки и установки:

$TST$  — проверка и установка обычной точности;

$TSTD$  — проверка и установка высокой точности.

Код команды: 1705  $FDST$ , формат  $F2$ .

Действие: установка в  $FPSW$  признаков в соответствии с содержимым  $FDST$ .

Признаки:  $FC := 0$ ;

$FV := 0$ ;

$FZ := 1$ , если  $E(FDST) = 0$ , иначе

$FZ := 0$ ;

$FN := 1$ , если  $(FDST) < 0$ , иначе

$FN := 0$ .

Описание: осуществляется анализ содержимого  $FDST$ , по результатам этого анализа устанавливаются признаки в  $FPSW$ . Содержимое  $FDST$  не изменяется.

Прерывание: несуществующий режим адресации; прерывание по неверным данным после выполнения команды.

#### Команды записи положительной:

$ABS$  — запись положительная обычной точности;

$ABSD$  — запись положительная высокой точности.

Код команды: 1706  $FDST$ , формат  $F2$ .

Действие: если  $(FDST) < 0$ , то  $FDST := \neg (FDST)$ ; если  $(FDST) > 0$ , то  $FDST := (FDST)$ ; если  $E(FDST) = 0$ , то  $FDST := 0$ .

Признаки:  $FC := 0$ ;

$FV := 0$ ;

$FZ := 1$ , если  $E(FDST) = 0$ , иначе

$FZ := 0$ ;

$FN := 0$ .

Описание: выполнение команды заключается в присвоении знака «+» содержимому  $FDST$ . Если характеристика  $FDST$  равна 0, то по адресу  $FDST$  записывается абсолютный нуль.

Прерывания: несуществующий режим адресации; прерывание по неверным данным после выполнения команды.

#### Команды записи дополнения:

$NEG$  — запись дополнения обычной точности;

$NEGD$  — запись дополнения высокой точности.

Код команды: 1707  $FDST$ , формат  $F2$ .

Действие:  $FDST := -(FDST)$ ; если  $E(FDST) = 0$ , то  $FDST := 0$ .

Признаки:  $FC := 0$ ;

$FV := 0$ ;

$FZ := 1$ , если  $E(FDST) = 0$ , иначе

$FZ := 0$ ;

FN:=1, если результат меньше 0, иначе FN:=0.

Описание: производится инвертирование знака содержимого FDST и установка признаков. В случае, если характеристика содержимого FDST равна 0, в FDST записывается абсолютный нуль.

Прерывание: несуществующий режим адресации; прерывание по неверным данным после выполнения команды.

#### Команды записи:

STF — запись обычной точности;

STD — запись высокой точности.

Код команды: 174 AC FDST, формат F1.

Действие: FDST:=(AC).

Признаки: не меняются.

Описание: число с ПЗ, содержащееся по адресу источника AC, помещается по адресу приемника FDST. Число с ПЗ может быть представлено в формате F или D в зависимости от значения FD. В результате выполнения этой команды неопределенное число с ПЗ —0 может быть помещено в память, если хранилось в AC как результат выполнения предыдущей команды.

Прерывание: несуществующий режим адресации.

#### Команды записи с преобразованием числа с ПЗ в число с ФЗ:

STCFI — запись с преобразованием числа с ПЗ формата F в число с ФЗ формата I;

STCFL — запись с преобразованием числа с ПЗ формата F в число с ФЗ формата L;

STCDI — запись с преобразованием числа с ПЗ формата D в число с ФЗ формата I;

STCDL — запись с преобразованием числа с ПЗ формата D в число с ФЗ формата L.

Код команды: 174 (AC+4) DST, формат F3.

Действие: DST:=CXJ(AC), если  $-(JL+1) \leq CXJ(AC) \leq JL$ , иначе DST:=0, где CXJ означает преобразование числа с ПЗ формата X в число с ФЗ формата J, причем:

J=I, если FL=0; J=L, если FL=1;

X=F, если FD=0; X=D, если FD=1;

JL — наибольшее число с ФЗ, равное  $2^{15}-1$  в формате I и  $2^{31}-1$  в формате L.

Признаки: C:=FC:=0, если выполняется неравенство  $-(JL+1) \leq CXJ(AC) \leq JL$ , иначе C:=FC:=1;

V:=FV:=0;

Z:=FZ:=1, если (DST)=0, иначе

Z:=FZ:=0;

N:=FN:=1, если (DST)<0, иначе

N:=FN:=0.

Описание: число с ПЗ, содержащееся в AC и имеющее формат F или D, в зависимости от значения FD представляется в виде целой и дробной части. Целая часть преобразуется в число с ФЗ формата I или L в зависимости от значения FL. Если число с ПЗ было отрицательным, то для представления числа с ФЗ используется дополнительный код. Полученное после преобразования целой части число с ФЗ помещается по адресу приемника DST. Дробная часть отбрасывается независимо от FT. Если задан регистровый или непосредственный режимы адресации приемника DST, а FL=1, то записывается только старшая часть числа с ФЗ, причем прерывание не происходит.

Прерывание: если значение целой части числа с ПЗ после преобразования превышает значение наибольшего числа с ФЗ с учетом значения FL, то

возникает программное нарушение по ошибке преобразования из ПЗ в ФЗ (наибольшее число с ФЗ равно JL для положительных чисел и JL+1 — для отрицательных). В этом случае по адресу приемника DST записывается 0 независимо от значения FIC. Если FIC=1, то происходит прерывание.

Точность: результат команды является точным только тогда, когда целая часть числа с ПЗ после преобразования не превышает значения наибольшего числа с ФЗ, определенного FL, а дробная часть равна 0.

#### Команды загрузки с преобразованием числа с ФЗ в число с ПЗ:

LDCIF — загрузка с преобразованием числа с ФЗ формата I в число с ПЗ формата F;

LDCID — загрузка с преобразованием числа с ФЗ формата I в число с ПЗ формата D;

LDCLF — загрузка с преобразованием числа с ФЗ формата L в число с ПЗ формата F;

LDCLD — загрузка с преобразованием числа с ФЗ формата L в число с ПЗ формата D.

Код команды: 177 AC SRC, формат F3.

Действие: AC:=CJX(SRC), где CJX означает преобразование числа с ФЗ формата J в число с ПЗ формата X, причем:

J=I, если FL=0; J=L, если FL=1;

X=F, если FD=0; X=D, если FD=1.

Признаки: FC:=0;

FV:=0;

FZ:=1, если (AC)=0, иначе

FZ:=0;

FN:=1, если (AC)<0, иначе

FN:=0.

Описание: число с ФЗ, содержащееся по адресу источника SRC и имеющее формат I или L, в зависимости от значения FL преобразуется в число с ПЗ формата F или D (в зависимости от значения FD). Полученное после преобразования число с ПЗ помещается по адресу приемника AC. Если число с ФЗ было отрицательное, то перед преобразованием оно переводится в прямой код. Если задан регистровый или непосредственный режим адресации источника SRC, а FL=1, то перед преобразованием число с ФЗ дополняется справа 16 нулями.

Точность: команды LDCIF, LDCID, LDCLD являются точными. Команда LDCLF вносит ошибку по наименьшему значащему разряду.

#### Команды записи с преобразованием формата числа с ПЗ:

STCFD — запись с преобразованием из обычной точности в высокую;

STCDF — запись с преобразованием из высокой точности в обычную.

Код команды: 176 AC FDST, формат F1.

Действие: FDST:=CXY(AC), где CXY означает преобразование числа с ПЗ формата X в число с ПЗ формата Y, причем:

X=F и Y=D, если FD=0;

X=D и Y=F, если FD=1.

Если E(AC)=0, то FDST:=0. Если FD=1, FT=0, FIV=0 и округление в команде STCDF вызвало переполнение порядка, то FDST:=0.

Признаки: FC:=0;

FV:=1, если округление в команде STCDF вызвало переполнение порядка, иначе FV:=0;

FZ:=1, если (AC)=0, иначе

FZ:=0;

FN:=1, если (AC)<0, иначе  
FN:=0.

Описание: число с ПЗ, содержащееся по адресу источника AC и представленное в формате F или D, в зависимости от значения FD преобразуется соответственно в формат D или F. Преобразование числа с ПЗ формата F в формат D выполняется путем дополнения недостающих разрядов мантииссы справа нулями. Преобразование числа с ПЗ формата D в формат F выполняется с учетом разряда FT. Полученный результат помещается по адресу приемника FDST.

Прерывание: несуществующий режим адресации; прерывание по переполнению порядка, может возникнуть при выполнении команды STCDF с округлением (FT=0).

Точность: команда STCDF вносит ошибку результата, определяемую выше.

#### Команда загрузки экспоненты:

LDEXP.

Код команды: 176 (AC+4) SRC, формат F3.

Действие:  $E(AC) := (SRC) + 200$ , если  $-200 < (SRC) < 200$ , знак и мантиисса числа не изменяются. Если  $(SRC) > 177$  и  $FIV=1$ , то  $E(AC) := ((SRC) + 200)$ . Если  $(SRC) > 177$  и  $FIV=0$ , то  $AC:=0$ . Если  $(SRC) < -177$  и  $FIU=1$ , то  $E(AC) := ((SRC) + 200)$ . Если  $(SRC) < -177$  и  $FIU=0$ , то  $AC:=0$ .

Признаки: FC:=0;

FV:=1, если (SRC)>177, иначе

FV:=0;

FZ:=1, если E(AC)=0, иначе

FZ:=0;

FN:=1, если (AC)<0, иначе

FN:=0.

Описание: числу с ПЗ, содержащемуся по адресу приемника AC, присваивается новая характеристика. Знак и мантиисса не изменяются. К числу с ФЗ, содержащемуся по адресу источника SRC и имеющему формат I, независимо от значения FL прибавляется 200 и значение становится новой характеристикой.

Прерывание: если значение числа с ФЗ превышает значение наибольшего порядка, равного 177, то возникает программное нарушение по переполнению порядка. При этом, если FIV=1, то полученная новая характеристика записывается по адресу приемника AC и происходит прерывание по переполнению порядка. Если FIV=0, то AC:=0 и прерывание отсутствует. Если значение числа с ФЗ меньше наименьшего отрицательного порядка, равного -177, то возникает программное нарушение по исчезновению порядка. При этом, если FIU=1, то полученная новая характеристика записывается по адресу приемника AC и происходит прерывание по исчезновению порядка. Если FIU=0, то AC:=0 и прерывание отсутствует.

#### Команда записи экспоненты:

STEXP.

Код команды: 175 AC DST, формат F3.

Действие:  $DST := E(AC) - 200$ .

Признаки: C:=FC:=0;

V:=FV:=0;

Z:=FZ:=1, если (DST)=0, иначе

Z:=FZ:=0;

N:=FN:=1, если (DST)<0, иначе

N:=FN:=0.

Описание: характеристика числа с ПЗ, содержащегося по адресу источника AC, которая является порядком числа с ПЗ, увеличенным на 200, преобразуется в число с ФЗ формата I независимо от состояния FL. Преобразование заключается в вычитании числа 200 из характеристики. Результат преобразования помещается по адресу приемника DST.

#### Команда загрузки слова состояния:

LDFPS.

Код команды: 1701 SRC, формат F4.

Действие:  $FPSW := (SRC)$ .

Описание: читается 16-разрядный операнд по адресу источника SRC и загружается в качестве текущего слова состояния программы ПЗ.

#### Команда записи слова состояния:

STFPS.

Код команды: 1702 DST, формат F4.

Действие:  $DST := (FPSW)$ .

Описание: текущее слово состояния программы ПЗ помещается по адресу приемника DST, причем разряды 13, 12 и 4 обнуляются.

#### Команда записи регистров прерывания:

STST.

Код команды: 1703 DST, формат F4.

Действие:  $DST := (FEC)$ ;  $DST+2 := (FEA)$ .

Описание: содержимое регистра кода прерывания FEC и регистра адреса прерывания FEA помещается соответственно по адресу приемника DST и адресу приемника, увеличенному на два (DST+2). Если задан регистровый или непосредственный режим адресации приемника, то записывается только содержимое FEC.

#### Команда установки формата высокой точности:

SETD.

Код команды: 170011, формат F5.

Действие:  $FD := 1$ .

Описание: устанавливается формат высокой точности путем установки в единицу разряда FD.

#### Команда установки формата обычной точности:

SETF.

Код команды: 17001, формат F5.

Действие:  $FD := 0$ .

Описание: устанавливается формат обычной точности путем очистки разряда FD.

#### Команда установки длинного формата целых чисел:

SETL.

Код команды: 170012, формат F5.

Действие:  $FL := 1$ .

Описание: признаки переписываются из слова целых чисел путем установки в единицу разряда FL.

#### Команда установки короткого формата целых чисел:

SETI.

Код команды: 170002, формат F5.

Действие: FL := 0.

Описание: устанавливается короткий формат целых чисел путем очистки разряда FL.

Команда переписи признаков:  
CFCC.

Код команды: 170000, формат F5.

Действие: C := FC; V := FV; Z := FZ; N := FN.

Описание: признаки переписываются из слова состояния FPSW в соответствующие разряды слова состояния (PSW).

Приложение 3

## ОРГАНИЗАЦИЯ ВВОДА-ВЫВОДА

Магистральный параллельный интерфейс (МПИ) предназначен для передачи данных между процессором, оперативной памятью и внешними устройствами, входящими в комплекс. МПИ соответствует ГОСТ 26765.51—86 и реализуется на основе магистрали и логических узлов подключаемых к ней устройств. При организации ввода-вывода информации всегда одно из взаимодействующих устройств управляет магистралью и называется задатчиком (ведущим) по отношению к связанному с ним устройству, называемому исполнителем (ведомым). В каждый момент времени на магистрали выполняется лишь один из трех видов взаимодействий подключенных к ней устройств: обмен информацией, передача управления магистралью и прерывание.

**Принцип работы.** Интерфейс использует временное мультиплексирование линий «адрес — данные». В определенный момент времени по этим линиям могут передаваться либо адрес, либо данные.

Принцип работы интерфейса при передаче адреса синхронный, а при передаче данных асинхронный. Синхронный принцип передачи заключается в том, что задатчик, передавая адрес, устанавливает синхронизирующий сигнал, по которому исполнитель принимает (стробирует) адрес, т. е. передача адреса производится по принципу «передача — прием». Асинхронный принцип передачи заключается в том, что на определенные сигналы, выданные задатчиком, должны быть ответы от исполнителя в порядке завершения передачи, т. е. обмен информацией производится по принципу «запрос — ответ — передача».

В связи с тем, что для завершения задатчиком цикла магистрали необходим ответ от исполнителя, каждый задатчик имеет схему выявления ошибки по тайм-ауту. Тайм-аут — это интервал времени, по истечении которого операция аварийно завершается, если исполнитель не отреагирует на операцию в течение данного интервала.

Таблица 44

Наименование линии (сигнала)	Мнемоника линии (сигнала)		Число линий	Функция	Источник сигнала	Присемник сигнала	Категория линии
	русская	английская					
<b>Шина обмена информацией</b>							
Адрес — данные	АД [15...0]	BDAL [15...0]L	16	1. Передача адреса устройства-исполнителя и/или ячейки памяти	Задатчик	Исполнитель	Двунаправленная
Расширение адреса	АР [21...16]	BDAL [21...16]L	6	2. Передача информации	Задатчик (исполнитель)	Исполнитель (здатчик)	То же
				1. Расширение адресного пространства при передаче адреса	Задатчик	Исполнитель	
				2. Передача ошибки паритета (BDAL [17, 16]L) при передаче данных	Исполнитель	Задатчик	
<b>Шина управления обменом</b>							
Синхронизация обмена	ОБМ	BSYNCL	1	Управление циклом магистрали	Задатчик	Все устройства	»
Чтение данных	ДЧТ	BDINL	1	Признак приема данных задатчиком	»	Исполнитель	»
Запись данных	ДЗП	BDOUTL	1	Признак выдачи данных задатчиком	»	»	»
Ответ устройства	ОТВ	BRPLYL	1	Подтверждение выполнения действия исполнителем	Исполнитель	Задатчик	»
Признак «запись — байт»	ПЗП	BWTBTL	1	1. Признак операции записи (при передаче адреса) 2. Признак операции записи байта (при передаче данных)	Задатчик	Исполнитель	»

Наименование линии (сигнала)	Мнемоника линии (сигнала)		Число линий	Функция	Источник сигнала	Приемник сигнала	Категория линии
	русская	английская					
Признак «запись байт»	ПЗП	BWTBTL	1	3. Признак записи слова в течение всего цикла записи блока	Задатчик	Исполнитель	Двунаправленная
Выбор устройства	ВУ	BBS7L	1	Признак обращения к адресуемым регистрам периферийных устройств или признак блочного чтения	»	»	То же
Признак «блочный обмен — регенерация»	РГН	BREFL	1	Подтверждение работы в блочном режиме, регенерация оперативной памяти	Исполнитель — активное устройство	Задатчик — динамическая память	»
<b>Шина передачи управления</b>							
Запрос магистрали	ЗМ	BDMRL	1	Запрос использования магистрали для передачи данных	Запрашивающее устройство	Процессор	»
Разрешение на захват магистрали	РЗМ, (РЗМП, РЗМИ)	BDMGL (BDMGIL, BDMGOL)	1	Разрешение использования магистрали для передачи данных	Процессор	Запрашивающее устройство	Однонаправленная
Подтверждение запроса	ПЗ	BSACKL	1	Подтверждение захвата магистрали	Задатчик	Процессор	Двунаправленная
<b>Шина прерывания</b>							
Запрос на прерывание	ЗПР	BIRQ [7..4]L	4	Запросы на использование шины для прерывания	Запрашивающее устройство	»	То же
Разрешение прерывания	ПРР (ПРРП, ПРРИ)	BIAKL (BIAKIL, BIAKOL)	1	Разрешение на использование шины для прерывания	Процессор	Запрашивающее устройство	Однонаправленная
Прерывание по внешнему событию	ПВС	BEVNTL	1	Запрос на прерывание по внешнему событию	Устройство	Процессор	Двунаправленная
<b>Вспомогательная шина</b>							
Установка	УСТ	BINITL	1	Сброс системы в исходное состояние	Пульт, процессор	Все устройства	То же
Останов	ОСТ	BHALTL	1	Установка в консольный режим	Пульт	Процессор	»
Авария: сетевого питания источника питания	АСП АИП	BERACL BERDCL	1 1	Контроль питания системы с целью принятия мер по защите информации от искажений	Блок питания	Процессор Все устройства	»

В случае необходимости обмена большими массивами данных с оперативной памятью внешние устройства используют блочный режим обмена информацией. При передаче блока данных выдается адрес только первого слова в блоке. Передача адреса для следующих слов исключается.

**Структура и состав интерфейса.** Магистраль включает в себя пять групп сигнальных линий и связанные с ними магистральные усилители-приемники и магистральные усилители-передатчики. По своему функциональному назначению сигнальные линии объединяются в следующие группы: шина обмена информацией, шина управления обменом, шина передачи управления, шина прерывания, вспомогательная шина. Линии и сигналы интерфейса приведены в табл. 44.

**Требования к функционально-временным характеристикам.** Все сигналы интерфейса в процессе обмена обязательно проходят через интерфейсные магистральные усилители — источник ИСТ (он же передатчик) и приемник ПРМ, прежде чем могут быть использованы каким-либо подключенным к магистрали устройством. Следует обратить внимание на то, что внутри устройства сигнал появляется в двух физически различных точках: на входе источника и на выходе приемника. Входы передатчиков и выходы приемников обозначены с учетом того, что они инвертируют сигналы (т. е. активный уровень сигнала на временной диаграмме высокий). Сигналы будут отмечаться следующим образом (вместо буквы В в мнемонике сигнала на шине): Т — передаваемый сигнал (на входе источника); R — принимаемый сигнал (на выходе приемника).

Когда два различных сигнала передаются от одного устройства к другому, переключаясь в одно и то же время, при их приеме может возникнуть разница во времени прихода (рассогласование) из-за разброса физических параметров приемопередатчиков и линий. Эта разница, или неопределенность в распространении сигнала по реальным цепям связи, называется перекосом. Для МПИ перекося сигнала гарантируется не более 75 нс. Операции передачи данных по магистрали приведены в табл. 45.

**Протокол цикла магистрали.** Устройство, ставшее задатчиком, начинает цикл магистрали, при этом предыдущая передача должна быть закончена (BSYNCL сброшен). Цикл магистрали можно разделить на следующие части: подцикл адресации и один или несколько подциклов передачи данных. Во время адресного подцикла задатчик выдает адрес нужного исполнителя. Все исполнители отвечают запоминанием битов адреса и сохранением

минимум через 150 нс устанавливает сигнал на линии BSYNCL. При этом адрес, BBS7L, BWTBTL устанавливаются на выходах приемника адресуемого устройства минимум за 75 нс до установки BSYNCL (время, необходимое для декодирования адреса исполнителем).

Время удержания и компенсации перекося начинается после установки BSYNCL: спустя минимум 100 нс задатчик снимает адрес, сбрасывает сигнал BBS7L и, если предстоит запись слова, сигнал BWTBTL. Эти сигналы должны сохраняться на приемниках исполнителя не менее 25 нс после установки BSYNCL. BSYNCL остается установленным в продолжении всего цикла магистрали. Опознавшее адрес устройство после появления сигнала BSYNCL становится исполнителем. Исполнитель назначен. Задатчик и исполнитель готовы к выполнению непосредственно процедуры передачи данных.

Память и периферийные устройства адресуются аналогично, за исключением использования BBS7L и старших разрядов адреса. Внешние устройства игнорируют 9 старших битов адреса BDAL[21...13]L, вместо них декодируется BBS7L совместно с 13 младшими битами адреса.

**Операция чтения.** При операции чтения (DATI) выполняется следующая последовательность событий:

здатчик производит адресацию устройства или памяти, выдавая адрес на линии BDAL[21...0]L и одновременно устанавливая сигнал BBS7L, а затем спустя не менее чем 150 нс выставляя сигнал BSYNCL;

исполнитель декодирует адрес и запоминает состояние «Устройство выбрано»;

здатчик снимает адрес на BDAL [21...0]L и сбрасывает BBS7L не менее чем через 100 нс после установки BSYNCL;

минимум через 100 нс (максимум 8 мкс, чтобы избежать тайм-аута; рекомендуемое время для тайм-аута 15...20 мкс, но не менее 8 мкс) после установки BSYNCL задатчик устанавливает сигнал BDINL;

исполнитель минимум через 0 нс после приема BDINL (максимум 8 мкс, чтобы избежать тайм-аута) и не более чем за 125 нс перед выдачей данных на передатчики BDAL устанавливает BRPLYL;

исполнитель минимум через 0 нс после приема BDINL и максимум через 125 нс после выдачи BRPLYL выдает данные на BDAL [15...0]L и информацию об ошибках на BDAL [17, 16]L (информация об ошибках выдается оперативной памятью, причем BDAL [16]L трактуется как ошибка паритета, а BDAL [17]L — разрешение ошибки паритета);

здатчик минимум через 200 нс, максимум через 2 мкс после установки BRPLYL принимает данные с BDAL [17...0]L и сбрасывает BDINL;

исполнитель отвечает на сброс BDINL сбросом BRPLYL и минимум через 0 нс, а максимум через 100 нс снимает данные с BDAL [17...0]L;

здатчик на сброс BRPLYL отвечает сбросом BSYNCL.

До следующей установки BSYNCL должны быть выполнены два условия: BSYNCL должен оставаться сброшенным минимум 200 нс и не должен устанавливаться минимум 300 нс после предыдущего сброса BRPLYL.

Таблица 45

Наименование операции	Мнемоника	Функция
<b>Режимы одиночного обмена</b>		
Чтение слова	DATI	Передача слова от исполнителя к задатчику
Запись слова	DATO	Передача слова от задатчика к исполнителю
Запись байта	DATOB	Передача байта от задатчика к исполнителю
Чтение — модификация — запись слова	DATIO	Передача слова от исполнителя к задатчику, обработка его задатчиком и передача результата обработки от задатчика к исполнителю по первоначальному адресу
Чтение — модификация — запись байта	DATIOB	Передача слова от исполнителя к задатчику, обработка его задатчиком и передача результата обработки (байта) от задатчика к исполнителю по первоначальному адресу
<b>Режимы блочного обмена</b>		
Чтение блока	DATBI	Передача блока данных от исполнителя к задатчику
Запись блока	DATBO	Передача блока данных от задатчика к исполнителю

этого состояния в течение всего цикла, пока не будет сброшен BSYNCL. Во время подцикла передачи данных выполняется собственно передача.

**Адресация устройств.** Адресный подцикл передачи данных по интерфейсу включает в себя время установки и компенсации перекося и время удержания и компенсации перекося. При установке и компенсации перекося задатчик выполняет следующее:

выдает на линии BDAL[21...0]L адрес, определяющий исполнителя, устанавливает сигнал BWTBTL при операциях DATO(B) или DATBO и сигнал BBS7L при обращении к устройству ввода-вывода. Установленный вместе с адресом BWTBTL указывает, что последующая операция будет DATO(B) или DATBO, а не установленный — DATI, DATBI или DATIO(B);

**Примечание.** Если BSYNCL остается установленным, то управление магистралью продолжает тот же задатчик и остается выбранным тот же исполнитель. Это используется в циклах DATIO(B), когда DATO или DATOB следует за DATI без сброса BSYNCL и без повторения подцикла адресации. Кроме того, медленно работающий исполнитель может удерживать передачу данных, поддерживая установленным BRPLYL, что заставит задатчик удерживать установленным BSYNCL.

**Операция записи.** При операции записи DATO(B) выполняется следующая последовательность событий:

задатчик производит адресацию устройства или памяти, выдавая адрес на линии BDAL[21...0]L и устанавливая сигналы BBS7L, если адрес принадлежит странице ввода-вывода, BWTBTL как признак записи не менее чем за 150 нс до установки сигнала BSYNCL и сам сигнал BSYNCL;

исполнитель декодирует адрес и запоминает состояние «Устройство выбрано»;

задатчик снимает адрес на BDAL[21...0]L, сбрасывает BBS7L и BWTBTL (если имеет место операция записи слова; при записи байта BWTBTL остается установленным до окончания цикла записи) не менее чем через 100 нс после установки BSYNCL и выдает данные на BDAL [15 ... 0]L;

минимум через 10 нс после установки данных и BWTBTL на своих передатчиках (максимум через 8 мкс после установки BSYNCL) задатчик устанавливает BDOUTL;

исполнитель принимает данные с линий BDAL, сигнал с линии BDOUTL и не позже чем через 8 мкс во избежание тайм-аута устанавливает сигнал на линии BRPLYL;

задатчик минимум через 150 нс после приема BRPLYL сбрасывает BDOUTL и минимум через 100 нс снимает данные с линий BDAL[15...0]L и BWTBTL, если выполнялась операция;

исполнитель воспринимает сброс BDOUTL. Данные принимаются и исполнитель сбрасывает BRPLYL;

задатчик отвечает сбросом BSYNCL, но не ранее чем через 175 нс после сброса BDOUTL и снятия данных с BDAL.

До следующей установки BSYNCL должны быть выполнены два условия: BSYNCL должен оставаться сброшенным минимум 200 нс и не должен устанавливаться минимум 300 нс после предыдущего сброса BRPLYL.

**Чтение с модификацией.** При чтении с модификацией DATIO(B) выполняется последовательность событий, аналогичная подциклам адресации и передачи данных для DATI и DATO(B). После адресации устройства подцикл чтения выполняется так, как было описано ранее для DATI, однако BSYNCL не сбрасывается. Сигнал BSYNCL остается установленным для записи слова или байта аналогично DATO(B). После завершения подцикла чтения задатчик выдерживает минимум 200 нс. от сброса BRPLYL до установки BDOUTL подцикла записи. Цикл заканчивается, когда задатчик сбрасывает BSYNCL, как это описано для DATO (B).

**Чтение блока.** При чтении блока DATBI подцикл адресации такой же, как описано ранее для других операций интерфейса. Задатчик выдает адрес первого слова в блоке на линиях BDAL[21...0]L, сбрасывает BBS7L и BWTBTL. Минимум через

150 нс после установки адреса устанавливается BSYNCL и сохраняется в течение всей процедуры блочного чтения. Минимум через 100 нс после установки BSYNCL устанавливается первый BDINL, а максимум через 50 нс после его установки задатчик устанавливает BBS7L, который используется в блочном обмене как признак блочного чтения и снимается перед чтением последнего слова.

Исполнитель устанавливает BRPLYL и выдает данные на BDAL [17...0]L после приема BDINL, как при обычном чтении. Одновременно с BRPLYL исполнитель устанавливает BREFL, если он может работать в блочном режиме и отвечать на следующие BDINL. Сигнал BREFL используется задатчиком в блочном обмене для подсчета считанных или записанных слов. При совпадении BREFL и BRPLYL задатчик устанавливает сигнал на линии BDINL для чтения следующего слова. Число выдаваемых сигналов BREFL должно быть на единицу меньше числа слов в считываемом блоке.

Если установлены сигналы BBS7L и BREFL, исполнитель готовится к повторному циклу чтения (прибавляет 2 к содержимому регистра адреса). Задатчик минимум через 200 нс, максимум через 2 мкс после установки BRPLYL сбрасывает BDINL и принимает данные с BDAL[17...0]L. При совпадении сигналов BRPLYL и BREFL задатчик минимум через 150 нс после сброса сигнала BDINL устанавливает его повторно. Исполнитель на сброс сигнала BDINL отвечает сбросом сигналов BRPLYL и BREFL, а получив новый сигнал BDINL, выставляет сигналы на линиях BRPLYL и BREFL и выдает данные на линии BDAL. Далее операция чтения многократно повторяется.

При заполнении счетчика адреса исполнителя, т. е. достижении им готовности к выдаче последнего в передаваемом блоке данных слова, исполнитель, получив сигнал BDINL, не устанавливает сигнал BREFL. Сигнал BREFL на выходе приемника задатчика сохраняет установившееся значение от максимум 75 нс после приема установки BRPLYL до минимум 20 нс после сброса BDINL. Задатчик может проверить состояние BREFL по заднему фронту BDINL. Если нарушается совпадение сигналов BRPLYL и BREFL, то задатчик не вырабатывает следующий сигнал BDINL. Если нет совпадения BBS7L и BREFL, исполнитель не готовит адрес следующего слова. Одновременно с выдачей последнего сигнала BDINL (но максимум через 50 нс) задатчик снимает сигнал BBS7L, а максимум через 250 нс после получения последнего сигнала BRPLYL — сигнал BSYNCL. На этом процедура блочного чтения завершается.

Если число слов в считываемом блоке меньше максимально возможного для исполнителя, процедура блочного чтения завершается снятием задатчиком сигнала BSYNCL.

**Запись блока.** При записи блока выполняется следующая последовательность событий (подцикл адресации такой же, как описано ранее для других операций интерфейса):

задатчик выдает адрес первого слова на линии BDAL[21...0]L, устанавливает сигнал BWTBTL и сбрасывает BBS7L. Минимум через 150 нс после установки адреса он устанавливает сигнал BSYNCL, минимум через 100 нс после установки BSYNCL выдает данные на BDAL[15...0]L и сбрасывает сигнал BWTBTL, так как при записи блока информа-

ция передается только словами; минимум через 100 нс после выдачи данных устанавливает сигнал BDOUTL;

исполнитель принимает установившееся значение данных не ранее чем за 25 нс до установки и не позже чем через 25 нс после сброса BDOUTL; устанавливает BRPLYL минимум через 0 нс после приема BDOUTL и одновременно BREFL, если является устройством с блочным режимом и способен обслужить следующие BDOUTL после принятого. Если исполнитель не может принять следующее слово (заполнение счетчика адреса), он не выставляет сигнал BREFL;

здатчик сбрасывает BDOUTL минимум через 150 нс после установки BRPLYL. При наличии сигналов BREFL и необходимости продолжения передачи данных задатчик минимум через 100 нс после сброса BDOUTL выдает на линии BDAL новые данные; минимум через 100 нс после выдачи новых данных и минимум через 150 нс после сброса BRPLYL устанавливает сигнал BDOUTL;

исполнитель сбрасывает сигнал после сброса BDOUTL. Если во время сброса сигнала BDOUTL сигнал BREFL не был установлен (BREFL сохраняет установившееся значение от максимум 75 нс после установки BRPLYL до минимум 20 нс после сброса BDOUTL) или если передача блока данных задатчиком закончена, данные снимаются задатчиком с линий BDAL минимум через 100 нс после сброса BDOUTL. Здатчик сбрасывает BSYNCL минимум через 175 нс после сброса BDOUTL и минимум через 0 нс после сброса BRPLYL. Процедура блочной записи завершается.

**Передача управления магистралью.** Процесс передачи управления магистралью можно разделить на три фазы: захват управления магистралью, передача данных и освобождение магистрали.

Во время захвата управления магистралью выполняется следующая последовательность событий: каждое активное устройство, готовое стать задатчиком, запрашивает магистраль, устанавливая сигнал BDMRL;

процессор параллельно с выполнением текущей программы анализирует состояние линии BDMRL и при появлении на ней сигнала, а также после окончания текущего обмена (при отсутствии сигнала BSACKL) или другого взаимодействия на магистрали начинает передачу управления магистралью, устанавливая сигнал BDMGOL, последовательно проходящий подключенные к магистрали устройства;

активное устройство, не готовое стать задатчиком (не выставившее сигналы BDMRL), после получения входного сигнала BDMGIL транслирует его далее — выдает сигнал BDMGOL, который сбрасывает после снятия сигнала BDMGIL;

ближайшее по линии BDMRL активное устройство, установившее сигнал BDMRL, при получении сигнала BDMGIL блокирует его дальнейшее распространение и, при отсутствии или после сброса сигналов от предыдущего цикла на линиях BRPLYL и BSYNCL, устанавливает сигнал на линии BSACKL и сбрасывает сигнал BDMRL.

Устройство, установившее сигнал BSACKL, становится задатчиком, и начинается фаза передачи данных. Во время этой фазы устройство с ПД продолжает держать установленным BSACKL. Собственно передача данных выполняется так, как описано ранее. Устройство с ПД может установить

BSYNCL для передачи данных минимум через 250 нс после приема BDMGIL и сброса предыдущего BSYNCL и BRPLYL на приемнике.

Во время фазы освобождения магистрали устройство с ПД освобождает шину, сбрасывая BSACKL. Это происходит после завершения или подавления последнего цикла передачи данных (BRPLYL сброшен). BSACKL может быть сброшен максимум за 300 нс до сброса BSYNCL.

**Прерывание.** Алгоритм прерывания на магистрали имеет три фазы: фазу запроса прерывания, фазу разрешения прерывания и арбитража приоритетов, фазу передачи вектора прерывания.

Во время запроса прерывания, разрешения прерывания и арбитража приоритетов выполняется следующая последовательность событий:

все устройства, подключенные к магистрали и требующие обработки прерывания, устанавливают запрос на прерывание не только на линии своего уровня прерывания, но и обязательно на линии BIRQ4L;

процессор параллельно с выполнением текущей программы анализирует состояние линий BIRQL и при наличии незамаскированных запросов, а также после окончания текущего обмена или другого взаимодействия на магистрали устанавливает сигнал BDINL и минимум через 150 нс — BIAKOL. Устройство, электрически ближайшее к процессору, принимает на свой приемник разрешение прерывания — сигнал BIAKIL;

если внутри этого устройства нет запроса прерывания, оно устанавливает BIAKOL и разрешение распространяется на следующее устройство магистрали; если устройство запрашивало прерывание, оно должно проверить, не запрашивает ли в этот момент прерывание какое-либо другое устройство с более высоким приоритетом, а для исключения гоночной ситуации должно запоминать по приему BDINL наличие как своего запроса, так и запросов на контролируемых линиях, ибо запросы на линии BIRQL могут выдаваться разными устройствами по мере возникновения соответствующих условий внутри этих устройств. Линии, которые требуется проверить на каждом уровне приоритета, указаны в табл. 46.

Таблица 46

Уровень приоритета устройства	Контролируемые линии
BIRQ4	BIRQ5, BIRQ6
BIRQ5	BIRQ6
BIRQ6	BIRQ7
BIRQ7	—

В дополнение к установке уровня 7 и 4 устройства седьмого уровня должны устанавливать запрос уровня 6. Это сделано для упрощения контроля линий в устройствах четвертого и пятого уровней. В соответствии с приведенным алгоритмом данные устройства могут не контролировать уровень 7, так как устройства седьмого уровня выставляют запрос уровня 6. Устройства четвертого и пятого уровней узнают о наличии запроса уровня 7, контролируя линию уровня 6. Этот алгоритм оптимизирован для устройств четвертого, пятого и шестого уровней,

поскольку устройства седьмого уровня требуются очень редко;

если ни одно устройство более высокого уровня не запрашивает прерывание, устройство, принявшее ВИАКІL, блокирует разрешение ВИАКОL и становится задатчиком. Начинается фаза передачи вектора прерывания;

если установлен запрос более высокого уровня, то устройство отказывается от разрешения прерывания и устанавливает ВИАКОL для распространения разрешения к следующему устройству на магистрали.

Фаза передачи вектора прерывания разрешается установкой сигнала BDINL и минимум через 150 нс сигнала ВИАКL. Устройство отвечает установкой BRPLYL и выдачей адреса вектора на BDAL[15...0]L. Сигналы на входах передатчиков BDAL должны установиться максимум через 125 нс после установки BRPLYL. Затем процессор принимает адрес вектора и сбрасывает BDINL и ВИАКОL. После этого устройство сбрасывает BRPLYL и максимум через 100 нс убирает биты адреса вектора.

Процессор производит обработку прерывания: сохранение PC и PSW прерванной программы; загрузку PC и PSW из адресуемых вектором ячеек; выполнение подпрограммы обработки прерывания.

**Прерывание по внешнему событию.** При наличии определенных условий в сетевом таймере разрешается обработка прерывания по внешнему сигналу BEVNTL. Обработка производится по шестому уровню приоритета. Процессор автоматически переходит к обслуживанию этого прерывания путем перехода по вектору с восьмеричным адресом 100. Так как адрес вектора внутренний, процессор не выполняет протокол магистрали для ввода адреса вектора прерывания, как в случае других внешних запросов прерывания.

**Включение, нарушение и восстановление питания.** Поскольку достоверность работы системы определяется состоянием питания, процессор, управляющий работой всех устройств, должен быть своевременно предупрежден о возможности нарушения питания, в том числе и питания периферийных и запоминающих устройств. Во время включения, выключения или нарушения питания имеет место последовательность операций, выполняемых процессором и устройствами и инициируемых в зависимости от состояния линий BERACL и BERDCL

на магистрали. Установка сигнала BERACL — первое событие в последовательности нарушения питания — указывает, что произошло нарушение и остается только 4 мс резерва питания постоянного тока. Установка сигнала BERDCL предупреждает, что остается только 5 мкс резерва питания постоянного тока, а сброс этого сигнала свидетельствует о стабильности питания постоянного тока минимум 3 мс. Сигналы BERACL и BERDCL должны оставаться сброшенными до тех пор, пока не произойдет нарушение питания.

Последовательность включения питания начинается, когда источник питания обеспечивает необходимый уровень напряжения постоянного тока с установленными сигналами BERDCL и BERACL. Это заставляет процессор устанавливать сигнал BINITL. После того, как питание постоянного тока установится, источник питания минимум через 3 мс сбрасывает сигнал BERDCL, минимум через 0 нс после его сброса процессор сбрасывает сигнал BINITL и минимум через 70 мс после сброса BERDCL источник питания сбрасывает сигнал BERACL. Процессор выполняет свою последовательность включения питания.

При нарушении питания предусмотрена следующая последовательность событий:

источник питания максимум через 10 мс устанавливает сигнал BERACL;

процессор переходит на выполнение программы отключения питания, которую должен выполнить максимум за 3 мс. При этом на 4...20 мкс устанавливается сигнал BINITL. Программа заканчивается инструкцией HALT, чтобы избежать возможного разрушения информации в оперативной памяти;

источник питания минимум через 4 мс после установки сигнала BERACL устанавливает сигнал BERDCL и минимум через 5 мкс после этого снимает постоянные стабилизированные напряжения.

**Примечание.** При любых нарушениях сети после установки сигнала BERACL обязательно должен быть установлен сигнал BERDCL. Здесь действует правило: раз последовательность нарушения питания начата, она должна быть доведена до конца.

Восстановление питания проводится после нормализации напряжения первичной системы электропитания по процедуре, аналогичной включению питания.

## Содержание

	Стр.
Назначение и область применения . . . . .	1
Структура и состав . . . . .	2
Процессор : : : . . . . .	—
Регистры общего назначения . . . . .	—
Регистры управления системой . . . . .	3
Организация памяти . . . . .	5
Стек . . . . .	—
Система команд и представление данных . . . . .	—
Быстродействующая буферная память . . . . .	—
Конструкция физического адреса и слова ББП . . . . .	6
Реакция на ошибки . . . . .	7
Регистры управления и состояния . . . . .	—
Средства преобразования адресов . . . . .	8
Диспетчер памяти . . . . .	9
Регистры диспетчера памяти . . . . .	—
Формирование физического адреса . . . . .	10
Системный интерфейс : . . . . .	11
Система прерываний . . . . .	—
Архитектура дисковой памяти . . . . .	13
Состав комплекса . . . . .	—
Конструктивно-элементная база . . . . .	14
Конструкция . . . . .	—
Элементная база . . . . .	—
Центральный микропроцессор . . . . .	—
Периферийный микропроцессор . . . . .	16
Микросхемы памяти . . . . .	—
Система электропитания . . . . .	—
Проектировочное обеспечение . . . . .	17
Модуль оперативный запоминающий (МОЗ) СМ 1425.3537 . . . . .	—
Контроллер НМД и НГМД СМ 1425.5140 . . . . .	19
Мультиплексоры передачи данных СМ 1425.8540 и СМ 1425.8544 . . . . .	22
Контроллер магнитной ленты СМ 1425.5021 . . . . .	23
Контроллер интерфейсов групповой СМ 1425.7009 . . . . .	26
Устройство согласования системных интерфейсов СМ 1425.4511 . . . . .	29
Накопители на несменных магнитных дисках . . . . .	—
Накопитель на гибком магнитном диске СМ 5640 . . . . .	30
Видеотерминал растровый СМ 7238 . . . . .	31
Устройство последовательной печати СМ 6329.02 . . . . .	—
Программное обеспечение . . . . .	32
Встроенная программа загрузки и диагностики . . . . .	—
Тестовое обеспечение . . . . .	—
Системное программное обеспечение . . . . .	33
Приложение 1. Система команд . . . . .	—
Приложение 2. Система команд с плавающей запятой . . . . .	43
Приложение 3. Организация ввода-вывода . . . . .	52

Редактор А. А. Черемкина  
Техн. редактор О. А. Овечкина  
Корректор В. А. Агеева

Сдано в набор 09.01.89. Подп. в печать 11.05.89. Формат 60×90<sup>1</sup>/<sub>8</sub>. Бумага типографская  
№ 1. Гарнитура литературная. Печать высокая. Усл. печ. л. 7,0. Уч.-изд. л. 8,5.  
Тираж 8904 экз. Заказ 13. Изд. № ГСП-529. Цена 1 р. 30 к.

Всесоюзный научно-исследовательский институт информации  
и экономики (ИНФОРМПРИБОР)  
125877, ГСП, Москва, А-252, Чапаевский пер., 14

Типография № 2 Госкомиздата РСФСР, 152901, г. Рыбинск, ул. Чкалова, 8.