

**МАШИНА ВЫЧИСЛИТЕЛЬНАЯ ЭЛЕКТРОННАЯ ПЕРСОНАЛЬНАЯ
"МИКРОША"**

Руководство по эксплуатации

ЯБЗ.031.092 РЭ

1. ОБЩИЕ УКАЗАНИЯ	3
2. КОМПЛЕКТ ПОСТАВКИ	3
3. ТЕХНИЧЕСКИЕ ДАННЫЕ	4
4. ТРЕБОВАНИЯ ПО ТЕХНИКЕ БЕЗОПАСНОСТИ	5
5. УСТРОЙСТВО ИЗДЕЛИЯ	5
6. ПОДГОТОВКА К РАБОТЕ	7
6.1. Порядок установки и включения	7
6.2. Подключение ПЭВМ к телевизору	8
6.3. Клавиатура ПЭВМ	9
6.4. Подключение ПЭВМ к магнитофону	14
7. ПОРЯДОК РАБОТЫ	18
7.1. "СИСТЕМНЫЙ МОНИТОР"	18
7.2. Система команд микропроцессора	32
7.3. Распределение адресного пространства ПЭВМ	50
7.4. Общие принципы программирования БИС контроллеров	52
7.5. Краткое описание языка Бейсик	67
7.6. Загрузчик	86
7.7. Редактор и Ассемблер	87
7.8. Дизассемблер	91
7.9. Отладчик	94
7.10. Редактор текстов	104
7.11. Графический редактор	108
7.12. Игровая программа "Быстрый счет"	114
7.13. Игровая программа "Удав"	115
7.14. Игровая программа "XONIX"	115
7.15. Игровая программа "TETRIS"	115
7.16. Игровая программа "РАСМАН"	116
7.17. Игровая программа "Лестница"	117
7.18. Игровая программа "Клад"	117
7.19. Игровая программа "Цирк"	117
8. ТЕХНИЧЕСКОЕ ОБСЛУЖИВАНИЕ	117
9. ГАРАНТИЙНЫЕ ОБЯЗАТЕЛЬСТВА	119
10. ГАРАНТИЙНЫЙ ТАЛОН	121
11. ОТРЫВНОЙ ТАЛОН НА ТЕХНИЧЕСКОЕ ОБСЛУЖИВАНИЕ	122
12. ПЕРЕЧЕНЬ ЭЛЕМЕНТОВ	123
13. СХЕМА ЭЛЕКТРИЧЕСКАЯ ПРИНЦИПИАЛЬНАЯ	-

1. ОБЩИЕ УКАЗАНИЯ

При покупке персональной ЭВМ (ПЭВМ) требуйте проверки ее работоспособности путем демонстрации выполнения одной из программ из состава программного обеспечения, входящего в комплект поставки.

Убедитесь в наличии на гарантийном талоне и отрывных талонах на гарантийное обслуживание штампа магазина, разборчивой подписи или штампа продавца и даты продажи. Проверьте сохранность пломб на ПЭВМ и ее комплектность.

После хранения в холодном помещении или после перевозки в зимних условиях включайте ПЭВМ в сеть не раньше, чем через 3 часа пребывания при температуре $(20 \pm 5)^\circ\text{C}$.

ПЭВМ предназначена для работы при температуре от 10 до 35°C и относительной влажности от 40 до 80%.

Не допускается хранить и эксплуатировать ПЭВМ в помещениях с повышенной влажностью. При работе с ПЭВМ не закрывайте отверстия на блоке питания и системном блоке посторонними предметами.

Кассету с программным обеспечением, входящую в комплект поставки ПЭВМ, рекомендуется хранить в футляре на расстоянии не менее 1 м от нагревательных приборов, в месте, защищенном от воздействия прямых солнечных лучей и резких перепадов температуры. Не рекомендуется оставлять кассету на длительное время на включенном магнитофоне.

Не включайте ПЭВМ при снятой с разъема "Внутренний интерфейс" крышке.

Не рекомендуется подключать ПЭВМ к сети 220В вблизи бытовых устройств, потребляющих от сети большой ток, например, холодильника или стиральной машины. Это может привести к сбоям при работе ПЭВМ.

Прежде, чем включать ПЭВМ, ознакомьтесь с настоящим руководством по эксплуатации, с элементами управления и надписями на системном блоке, блоке питания и модуляторе.

2. КОМПЛЕКТ ПОСТАВКИ

Системный блок	1 шт.
Блок питания	1 шт.
Модулятор	1 шт.
Кассета с программным обеспечением	1 шт.
Руководство по эксплуатации	1 книга
Запасная вставка плавкая	1 шт.
Упаковка	1 шт.

3. ТЕХНИЧЕСКИЕ ДАННЫЕ

Разрядность	8 бит
Объем оперативной памяти	32 Кбайт
Формат изображения при выводе информации на экран телевизионного приемника:	
- алфавитно-цифровой	64X25 знаков
- псевдографической	128x50 точек
Тактовая частота микропроцессора	1,8 МГц
Время выполнения операции типа регистр-регистр или арифметической операции	3 мкс
Питание от сети переменного тока	220 В, 50 Гц
Допустимые колебания напряжения сети от	187 до 242 В
Потребляемая мощность, не более	20 Вт
Условия эксплуатации:	
- температура окружающего воздуха	от 10 до 35°C
- относительная влажность воздуха	от 40 до 80%
Масса и габаритные размеры приведены в табл. 1	

Таблица 1

Наименование	Масса, кг	Габаритные размеры, мм
Блок системный	1,4	390x230x55
Блок питания	1,3	160x100x100
Модулятор	0,2	100x30x40

Сведения о содержании драгоценных металлов:

- 1) золото - 4,2989 г.;
- 2) серебро - 0,8892 г.;
- 3) палладий - 0,0011 г.

4. ТРЕБОВАНИЯ ПО ТЕХНИКЕ БЕЗОПАСНОСТИ

БУДЬТЕ ОСТОРОЖНЫ!

В блоке питания ПЭВМ имеется опасное для жизни напряжение (220В). Во избежание несчастных случаев категорически запрещается включать в сеть блок питания при снятых деталях корпуса.

Замена вставки плавкой производится только при выключенном из сети блоке питания. Перед включением ПЭВМ убедитесь в отсутствии повреждений изоляции сетевого шнура блока питания, повреждений корпусов блока питания и системного блока. Не подключайте ПЭВМ к неисправным телевизору и магнитофону.

5. УСТРОЙСТВО ИЗДЕЛИЯ

ПЭВМ "Микроша" предназначена для массового применения в быту при изучении основ вычислительной техники и программирования, составления прикладных программ пользователя, обработки и хранения информации, организации досуга.

Для обеспечения работы с ПЭВМ необходимо правильное объединение в единую систему системного блока, блока питания, модулятора, телевизионного приемника и бытового кассетного магнитофона.

Телевизионный приемник и магнитофон не входят в комплект поставки ПЭВМ, предполагается, что эти устройства имеются у покупателя.

ПЭВМ выполнена на основе микропроцессора КР580ВМ80А. Оперативное запоминающее устройство (ОЗУ) динамического типа имеет емкость 32 КБайт.

Системный блок ПЭВМ содержит клавиатуру, предназначенную для управления ПЭВМ, а также разъемы для подключения блока питания ("Питание"), магнитофона ("Магнитофон"), модулятора ("Видео"), внешних устройств ("Интерфейс-1", "Интерфейс-2", "Внутренний интерфейс").

Клавиатура ПЭВМ выполнена с использованием контактных переключателей. На каждый переключатель установлена клавиша с нанесенными на ней знаками.

Блок питания ПЭВМ состоит из силового трансформатора, выпрямителей и стабилизаторов напряжения +5В, -5В, +12В.

Модулятор предназначен для подключения ПЭВМ к телевизионному приемнику через антенный вход без каких-либо доработок телевизора. С выхода модулятора в телевизионный приемник подается сигнал с частотой четвертого или пятого телевизионного канала, модулированный по амплитуде видеосигналом, поступающим из системного блока.

В ПЭВМ предусмотрена звуковая индикация (встроенная в системный блок динамическая головка) и световая индикация текущего регистра ("РУС"). ПЭВМ обеспечивает отображение на экране телевизора букв русского алфавита

(строчных и прописных), латинского алфавита, цифр, знаков препинания и псевдографических символов.

Программное обеспечение ПЭВМ включает программу "СИСТЕМНЫЙ МОНИТОР", размещенную в постоянном запоминающем устройстве (ПЗУ) системного блока, и набор программ на магнитофонной кассете, входящей в комплект поставки ПЭВМ. В ПЭВМ предусмотрена возможность ввода программ в ОЗУ с магнитной ленты с помощью бытового кассетного магнитофона, а также запись программ на магнитофон.

В последующих разделах Вы можете более подробно познакомиться с устройством ПЭВМ.

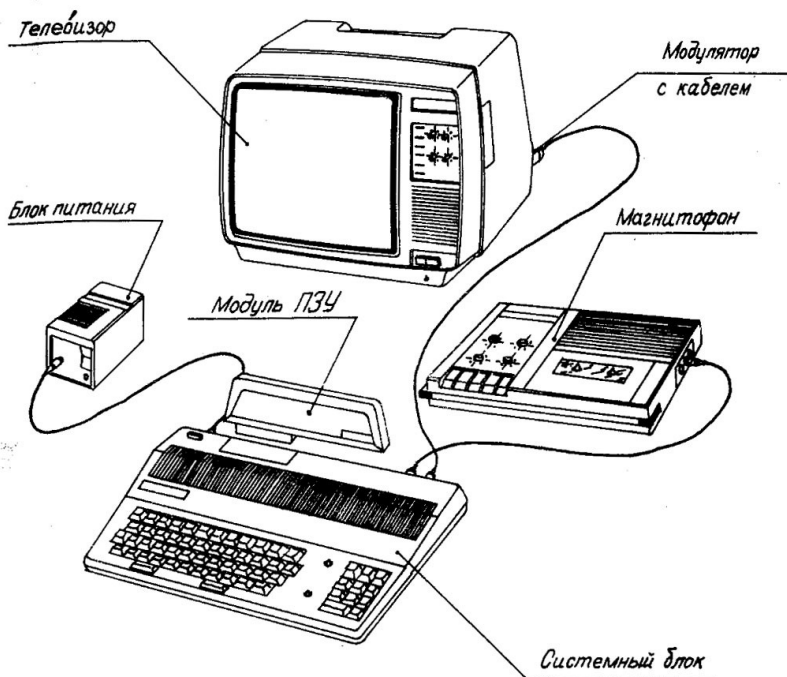


Рис. 1

6. ПОДГОТОВКА К РАБОТЕ

6.1. Порядок установки и включения

Перед началом работы ознакомьтесь с настоящим руководством по эксплуатации ПЭВМ. Проверьте комплектность ПЭВМ в соответствии с разделом 2 настоящего руководства по эксплуатации. При работе Вам понадобятся телевизор и бытовой кассетный магнитофон.

Перед подключением к ПЭВМ убедитесь в исправности телевизора и магнитофона.

Соедините составные части ПЭВМ в единую систему в соответствии с рис.1. Расположение разъемов для подключения внешних устройств показано на рис.2.

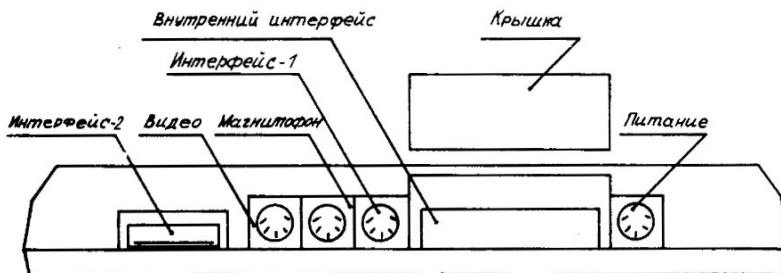


Рис. 2

Разъемы "Интерфейс-1", "Интерфейс-2", "Внутренний интерфейс" предназначены для подключения внешних устройств, разработанных специально для ПЭВМ "Микроша".

Расположите составные части ПЭВМ таким образом, чтобы при работе клавиатура системного блока находилась перед Вами. Телевизор можно установить за системным блоком. Магнитофон расположите рядом с системным блоком: Вам понадобится вводить программы с магнитофона в ПЭВМ и записывать программы на магнитофон. Блок питания можно установить в менее доступном месте, насколько позволяет длина кабеля, соединяющего блок питания с системным блоком. Подключите блок питания к разъему "Питание", кабель модулятора — к разъему "Видео", кабель магнитофона — к разъему "Магнитофон" системного блока. При подключении к ПЭВМ магнитофона используйте кабель, входящий в комплект магнитофона. Более подробно о подключении ПЭВМ к магнитофону Вы можете прочитать в конце данного раздела.

Модулятор подключите к антенному входу телевизора, предварительно отключив телевизор от антенны. При подключении используйте, в зависимости от типа телевизора, антенный вход 1:1 или MB. Следите, чтобы

кабель модулятора не был натянут, иначе контакт с антенным гнездом телевизора может нарушиться.

Включите питание телевизора и магнитофона. Переключатель сети на блоке питания переведите в положение "ВКЛ". При этом должен загореться светодиод на блоке питания. Нажмите на кнопку "СБРОС" на системном блоке.

6.2. Подключение ПЭВМ к телевизору

Модулятор ПЭВМ можно настроить на частоту четвертого или пятого телевизионного канала (от 85,25 до 93,25 МГц) вращением ротора подстроенного конденсатора "ЧАСТОТА" (гравировка "Ч" на корпусе модулятора).. Переключатель селектора каналов телевизора установите на четвертый или пятый канал. При выборе телевизионного канала следует убедиться в том, что на этом канале отсутствуют телевизионные передачи. Если модулятор настроен на частоту, близкую к частоте телевизионной передачи, изображение на экране телевизора ухудшается.

Тумблер "Настройка ручная-автоматическая" телевизора поставьте в положение "Автоматическая". Когда телевизор прогреется, настройте модулятор по наилучшему изображению на экране вращением - ротора подстроенного конденсатора "Ч". В верхней левой части экрана должна появиться надпись "МИКРОША", символ "->" и мигающая черта - курсор.

При плохом изображении тумблер "Настройка ручная-автоматическая" поставьте в положение "Ручная" и плавным вращением ручной настройки добейтесь устойчивого изображения.

Если Ваш телевизор имеет плавную настройку на телевизионный канал, настройтесь на частоту модулятора с помощью селектора каналов телевизора.

С помощью регулировки строчной и кадровой развертки добейтесь устойчивой синхронизации изображения.

Для более точной настройки телевизора наберите на клавиатуре системного блока:

D, FF

и нажмите на клавишу "BK". На экране должна появиться таблица, состоящая из столбцов цифр и букв. С помощью смещения по горизонтали (при наличии) установите таблицу таким образом, чтобы все ее элементы были видны на экране. Регулированием размера по вертикали установите оптимальную величину знаков. Изменяя яркость и контрастность, добейтесь наилучшего изображения на экране телевизора.

6.3. Клавиатура ПЭВМ

Прежде чем начать работу с ПЭВМ, необходимо ознакомиться с клавиатурой. Клавиатура ПЭВМ состоит из трех видов клавиш:

- алфавитно-цифровые клавиши;
- функциональные клавиши;
- клавиши управления.

Алфавитно-цифровые клавиши позволяют использовать при работе с ПЭВМ:

- 32 символа букв русского алфавита;
- 26 символов букв латинского алфавита;
- 10 символов арабских цифр;
- 27 символов специальных знаков (арифметические действия, знаки препинания и т.д.).

Как уже отмечалось выше, после нажатия кнопки "СБРОС" на экране появляется символ и мигающая черта (курсор). При нажатии любой алфавитно-цифровой клавиши курсор сдвигается на одну позицию (знакоместо) вправо, а на его месте появляется символ, изображенный на нажатой клавише. В некоторых случаях курсор расположен на одну строку выше появляющегося на экране символа, что не влияет на работу ПЭВМ.

Клавиатура может работать в латинском или русском наборе букв (нижний или верхний регистр). После нажатия кнопки "СБРОС" устанавливается нижний регистр (латинские буквы). При нажатии алфавитной клавиши на экране появляется символ, изображенный в нижней части нажатой клавиши. Это не относится к цифровым клавишам и к шести клавишам, содержащим специальные знаки-

"," "- ":" ". ", "/"
+ , = , * , > , < , ? .

При нажатии на клавишу "РУС/LAT" загорается светодиод "РУС", появляется звуковой сигнал, и устанавливается верхний регистр. Если Вы нажмете на любую алфавитно-цифровую клавишу, звуковой сигнал прекращается, и на экране появляется символ, расположенный в верхней части нажатой клавиши. При повторном нажатии клавиши "РУС/LAT" светодиод "РУС" гаснет и включается нижний (латинский) регистр.

Регистр можно также переключить с помощью клавиши "НР". Если нажать и удерживать эту клавишу, регистр переключается на противоположный. При отпускании клавиши регистр переходит в исходное состояние. Необходимо отметить, что вывести на экран символы, расположенные в нижней части цифровых клавиш и перечисленных выше шести клавиш со специальными знаками можно только при нажатой клавише "НР". Клавишей "НР" удобно пользоваться, когда при работе с клавиатурой необходимо часто менять регистр.

С помощью функциональной клавиши "F1" на экран можно выводить не только прописные, но и строчные буквы русского алфавита. При однократном нажатии "F1" все выведенные на экран латинские буквы или знаки "[", "]", "^", "@", "_" превращаются в строчные буквы русского алфавита, расположенные на той же клавише, что и соответствующие латинские буквы или знаки (например, перечисленные выше знаки превращаются в строчные буквы "ш", "щ", "ч", "ю", "ъ"). При повторном нажатии клавиши "F1" происходит обратное преобразование строчных букв русского алфавита в прописные латинские буквы и знаки. Возможность вывода на экран строчных латинских букв в ПЭВМ не предусмотрена.

К алфавитно-цифровым клавишам относятся две клавиши "ПРОБЕЛ" (" "), расположенные в нижней части клавиатуры. При нажатии на любую из этих клавиш курсор сдвигается на одну позицию вправо, оставляя пустое знакоместо.

К функциональным клавишам относятся клавиши "F1", "F2", "F3", "F4", "F5", расположенные в правой части клавиатуры.

Функция, выполняемая клавишей "F1", описана выше. Клавиши "F2"-"F5" могут выполнять различные функции, которые задаются конкретной программой. Например, в программе "Редактор текстов" клавиши "F2" и "F4" выполняют функции редактирования. Кроме того, с помощью клавиши "F4" можно остановить выполнение ряда директив "СИСТЕМНОГО МОНИТОРА".

К клавишам управления относятся клавиши "СВРОС", "НР", "УС", "ПС", "ВК", "ГТ", "АР2", "РУС/LAT", "СТР", "↖" (Стрелка вверх-влево), "←", "→", "↑", "↓". Действие клавиш "СВРОС", "НР" и "РУС/LAT" описано выше. Клавиша "УС" - управляющий символ, -

используется совместно с другими клавишами. При одновременном нажатии клавиши "УС" и определенных алфавитных клавиш выполняется та или иная функция. Перечень функций, выполняемых ПЭВМ при нажатии "УС" и других клавиш, приведен в табл. 2.

Клавиша "ПС" - перевод строки, сдвигает курсор на одну, строку вниз. Используется в некоторых программах при редактировании.

Клавиша "ВК" - возврат каретки, - обеспечивает ввод директив, набранных на клавиатуре. Обычно эта клавиша нажимается после набора какого-либо выражения, состоящего из алфавитно-цифровых символов, после чего ПЭВМ воспринимает данное выражение. Если выражение не имеет смысла, курсор переходит в начало следующей строки, а перед набранным выражением появляется вопросительный знак.

Клавиша "ГТ" - горизонтальная табуляция, - приобретает управляющую функцию только в конкретных программах. Например, в программе "Редактор текстов" при нажатии "ГТ" курсор перемещается на 8 знакомест вправо. Без

специальной программы с помощью "ГТ" на экран выводится символическое изображение человека, используемое в игровых программах.

Клавиша "AP2", как и "ГТ", используется для выполнения функций управления в различных прикладных программах. Например, в программе "Редактор текстов" с помощью "AP2" происходит переключение режимов редактирования.

С помощью клавиши "СТР" происходит очищение экрана, курсор перемещается в левый верхний угол экрана.

Клавиша "↖" (Стрелка вверх-влево) перемещает курсор в левый верхний угол без очищения экрана.

Клавиши "→", "↑", "↓", "←" управляют положением курсора.











































Если какую-либо клавишу удерживать в нажатом состоянии более 1 секунды, происходит повторение выполняемых этой клавишей функций. Например, при нажатии алфавитно-цифровой клавиши происходит вывод на экран соответствующих символов до тех пор, пока нажатая клавиша не будет отпущена.

Для удобства работы с клавиатурой в ПЭВМ введена звуковая индикация нажатия клавиши. Каждое замыкание контактов любой клавиши, кроме "НР", "УС" и "РУС/LAT" сопровождается щелчком.

Каждому алфавитно-цифровому и псевдографическому символу, выводимому на экран телевизора, а также функциям, выполняемым при нажатии функциональных клавиш и клавиш управления (кроме "УС", "НР", "РУС/LAT") соответствуют в ПЭВМ управляющие, псевдографические и алфавитно-цифровые коды дисплея. Методы использования кодов дисплея при составлении программ пользователя можно рассмотреть на примерах, приведенных в разделе 7.1.3. В настоящем разделе приводятся таблицы алфавитно-цифровых, псевдографических и управляющих кодов дисплея (табл. 3,4,5).

функции, выполняемые ПЭВМ и символы, выводимые на экран при
одновременном нажатии клавиши "УС" и других клавиш

Таблица 2

"УС"+ КЛАВИША (РУС/ LAT)	ВЫПОЛНЯЕМАЯ ФУНКЦИЯ	КЛАВИША, ВЫПОЛНЯЮЩАЯ ТУ ЖЕ ФУНКЦИЮ (БЕЗ"УС")	СИМВОЛ НА ЭКРАНЕ	
			БЕЗ "F1"	ПОСЛЕ "F1"
Я/Ј	Перевод строки	ПС	-	-
Ц/С	Аналогично "F4"	F4		
У/Ў	-	-		
К/К	-	-		
Е/Е	-	-		
Н/Н	-	-		
Г/Г	Звуковой сигнал	-	-	-
Ш/[Аналогично "AP2"	AP2	-	-
Щ/]	-	-		
З/З	Курсор вниз	↓	-	-
Х/Н	Курсор влево	←	-	-
Ф/Ф	-	-		
Ы/У	Курсор вверх	↑	-	-
В/В	-	-		
А/А	Аналогично "F2"	F2		
П/Р	-	-		
Р/Р	-	-		
О/О	-	-		
Л/Л	Курсор в левый верхний угол экрана	↖	-	-
Д/Д	Аналогично "F5"	F5		
Ж/У	-	-		
Э/\	-	-		
Ъ/_	Очистка экрана	СТР	-	-
Я/О	-	-		
Ч/^	-	-		
С/С	-	-		
М/М	Возврат каретки	ВК	-	-
И/И	Аналогично "ГТ"	ГТ		
Т/Т	-	-		
Ь/Х	Курсор вправо	→	-	-
Б/В	Аналогично "F3"	F3		
Ю/@	Переключение знакогенератора	F1	-	-

Примечание. Прямоугольником обозначено знакоместо.











































Управляющие коды дисплея

Таблица 3

КОД	КЛАВИША	ФУНКЦИЯ
07	УС+Г/G	ЗВУКОВОЙ СИГНАЛ
08	←	ПЕРЕМЕЩЕНИЕ КУРСОРА ВЛЕВО НА 1 ПОЗИЦИЮ
0A	ПС	ПЕРЕВОД СТРОКИ
0C	↖	УСТАНОВКА КУРСОРА В НУЛЕВУЮ ПОЗИЦИЮ В ЛЕВЫЙ ВЕРХНИЙ УГОЛ ЭКРАНА
0D	ВОЗВРАТ КАРЕТКИ	УСТАНОВКА КУРСОРА В НАЧАЛО СТРОКИ
18	→	ПЕРЕМЕЩЕНИЕ КУРСОРА ВПРАВО НА 1 ПОЗИЦИЮ
19	↑	ПЕРЕМЕЩЕНИЕ КУРСОРА ВВЕРХ НА 1 ПОЗИЦИЮ
1A	↓	ПЕРЕМЕЩЕНИЕ КУРСОРА ВНИЗ НА 1 ПОЗИЦИЮ
1B	AP2	ОПРЕДЕЛЯЕТСЯ ПРОГРАММОЙ ПОЛЬЗОВАТЕЛЯ
1F	СТР	ОЧИСТКА ЭКРАНА С УСТАНОВКОЙ КУРСОРА В НУЛЕВУЮ ПОЗИЦИЮ

Псевдографические коды дисплея

Таблица 4

КОД	КЛАВИША	СИМВОЛ	
		БЕЗ "F1"	ПОСЛЕ "F1"
01	F2, УС+А/А		
02	F3, УС+В/В		
03	F4, УС+Ц/С		
04	F5, УС+Д/Д		
05	УС+Е/Е		
06	УС+Ф/Ф		
09	ГТ, УС+И/И		
0B	УС+К/К		
0E	УС+Н/Н		
0F	УС+О/О		
10	УС+П/Р		
11	УС+Я/Q		
12	УС+Р/R		
13	УС+С/S		
14	УС+Т/T		
15	УС+У/U		
16	УС+Ж/V		
17	УС+В/W		
1C	УС+Э/\		
1D	УС+Щ/]		
1E	УС+Ч/^		

6.4. Подключение ПЭВМ к магнитофону

Для ввода программ с магнитофона рекомендуется подключать ПЭВМ к линейному выходу магнитофона ("⊕"). как показано на схеме электрической принципиальной (лист " "), при вводе программ сигнал поступает на контакт 3 разъема "Магнитофон".

Уровень входного сигнала должен быть не менее 0,1В в случае, если уровень сигнала низкий и программа загружается с ошибками, можно подать сигнал в ПЭВМ с розетки для подключения головных телефонов ("⊖") при максимальном уровне громкости. При записи программ на магнитную ленту подключайте ПЭВМ к входу магнитофона "Запись с электрофона" ("⊖"), "Запись с магнитофона" ("⊖") или "Запись с радиотрансляционной сети" ("⊖"). Сигнал на запись поступает с контакта 1 разъема "Магнитофон".

Для того, чтобы убедиться в правильности подключения магнитофона к ПЭВМ попробуйте ввести в ПЭВМ одну из программ. Для этого установите в магнитофон кассету с программным обеспечением, перемотайте магнитную ленту влево и найдите начало первой программы (программа "Загрузчик"). Каждая программа начинается тональным сигналом длительностью около 3 с. Перед началом программы голосом объявляется ее название.

Нажмите на клавиатуре ПЭВМ клавишу "I" (латинская буква). Включите режим "Воспроизведение" магнитофона. Одновременно с началом тонального сигнала нажмите клавишу "BK". Допускается нажатие "BK" не более чем за 2с до и не более чем через 2с после начала тонального сигнала. Если время между моментом нажатия "BK" и началом тонального сигнала превысит 2с, программа может загрузиться с ошибкой.

После нажатия клавиши "BK" изображение на экране телевизора пропадает, остается только мигающий курсор. Прослушивается тональный сигнал, а затем шум цифровой записи (программы). После окончания тонального сигнала и начала программы курсор должен исчезнуть. Если сигнал из магнитофона в ПЭВМ не поступает из-за неправильного подключения, неисправности магнитофона или по какой-либо другой причине, через 2-3 сек. после нажатия клавиши "BK" на экране появится сообщение:

?!I
-МИКРОША-
->
и курсор.

В случае успешной загрузки программы на экране появится сообщение:

7400
7598
851A

где 7400, 7598 — начальный и конечный адреса области оперативной памяти, в которую была введена программа "Загрузчик", 851A — контрольная сумма программы. Эти данные вводятся в программу при записи на магнитофон и для данной программы остаются неизменными.

Если по какой-либо причине программа загрузилась с ошибкой, на экране появляется сообщение:

7400
7598
9621 ?!
851A

где 9621 — неправильная контрольная сумма, полученная в результате загрузки, 851A — правильная контрольная сумма программы "Загрузчик", 7400, 7598 начальный и конечный адреса программы. При большом количестве ошибок при загрузке возможно изменение контрольной суммы, начального и конечного адресов или появления сообщения:

?!I
-МИКРОША-
->

Причиной появления ошибок при загрузке программ может быть неисправность магнитофона, дефект записи на магнитной ленте или неисправность ПЭВМ.

Существенное влияние на качество загрузки программ оказывают даже небольшие отклонения технических параметров магнитофона от номинальных значений. Обычно такие отклонения параметров не выявляются при прослушивании фонограмм и определяются с помощью специального оборудования. Далее приводятся основные виды неисправностей магнитофона и методы повышения надежности загрузки программ при наличии неисправностей.

1) Скорость движения ленты не соответствует номинальной. В этом случае влияние неисправности полностью устраняется с помощью программы "Загрузчик". Руководство по применению этой программы приведено в разделе 7.

2) Неправильное положение магнитной головки, вследствие чего уменьшается уровень выходного сигнала. Как было указано выше, можно увеличить уровень выходного сигнала, подавая сигнал в ПЭВМ с выходного разъема для подключения головных телефонов при максимальном уровне

громкости.

3) Низкий уровень сигнала на линейном выходе, неправильно отрегулирован усилитель воспроизведения магнитофона. В этом случае, как и в предыдущем, рекомендуется подавать сигнал в ПЭВМ с разъема для подключения головных телефонов.

4) Коэффициент детонации магнитофона превышает допустимое значение. При этом возможно появление ошибок при загрузке, однако предыдущие методы повышения надежности загрузки в данном случае неэффективны. При появлении ошибки необходимо повторить операцию загрузки программы.

Причиной ошибки может быть загрязнение магнитной головки, прижимного ролика и других элементов лентопротяжного тракта магнитофона.

При неустойчивой загрузке программы рекомендуется добиться правильной загрузки с помощью одного из перечисленных выше методов, а затем записать программу на свой магнитофон из оперативной памяти ПЭВМ.

Порядок записи программ из ПЭВМ на магнитофон следующий. Перед началом записи программы рекомендуется с помощью микрофона записать название данной программы. Затем отключите микрофон, подключите ПЭВМ ко входу магнитофона "Запись с электрофона" ("МАГЭЛЕКТОРОФОН"), "Запись с магнитофона" ("МАГЗАПИСЬ"). "Запись с радиотрансляционной линии" ("МАГТРАНСЛЯЦИЯ"). Нажмите клавишу "О" на клавиатуре ПЭВМ, наберите начальный и конечный адрес программы, которую Вы хотите записать, например:

01100,1CFF

Включите магнитофон в режим записи. Нажмите клавишу "ВК". Увеличивая регулятором уровень записи магнитофона, добейтесь, чтобы индикатор "УРОВЕНЬ ЗАПИСИ" магнитофона показывал максимальное значение. Однако устанавливать регулятор уровня записи в крайнее положение (при наличии запаса регулировки) не следует, так как при этом могут возникнуть искажения сигнала и качество записи программы ухудшается. Установив один раз оптимальный уровень записи, Вы можете и в дальнейшем производить запись с данным уровнем.

Установив уровень записи, перематывайте ленту на то место, где Вы записали название программы, нажмите клавишу "СБРОС", "О", наберите начальный и конечный адрес программы и повторите запись при оптимальном уровне. При записи программы после нажатия "ВК" экран гаснет, в динамике магнитофона в течение 3 с. прослушивается тональный сигнал, затем в динамиках ПЭВМ и магнитофона - сигнал записываемой программы, светодиод "РУС" на системном блоке загорается в такт записываемому сигналу. После окончания записи на экране появляется

набранное перед началом записи выражение и контрольная сумма записанной программы, например:

01100,1CFF
9911

Чтобы убедиться в том, что программа записана правильно, перемотайте ленту до начала программы и произведите загрузку в соответствии с указаниями настоящего раздела. В случае возникновения ошибки при загрузке проверьте правильность соединения ПЭВМ с магнитофоном и повторите запись программы при другом уровне или на другом участке ленты (возможен дефект на небольшом участке магнитной ленты).

Если характерный шум, сопровождающий запись информации на магнитофон, является нежелательным, то перед выполнением директивы "O" необходимо выполнить следующие действия:

- 1) нажать клавишу "СБРОС";
- 2) набрать на клавиатуре директиву "MD803" и нажать "BK";
- 3) после появления на экране строки "D803 82" последовательно нажать клавиши "A", "O", "BK",
- 4) в результате ПЭВМ возвратится в режим "СИСТЕМНОГО МОНИТОРА", о чем свидетельствует появление группы символов "->".

Звуковое сопровождение выводимой информации будет отключено до тех пор, пока не будет нажата клавиша "СБРОС".

7. ПОРЯДОК РАБОТЫ

7.1. "СИСТЕМНЫЙ МОНИТОР"

7.1.1. Общие сведения

Если Вы подключили ПЭВМ к телевизору и магнитофону в соответствии с п.п. 6.2, 6.4 и изучили клавиатуру в соответствии с п. 6.3 настоящего руководства, Вы можете приступить к работе с ПЭВМ. Нажмите кнопку "СБРОС". В левом верхнем углу экрана появляется надпись "МИКРОША" и символ "->". Это означает, что работает программа "СИСТЕМНЫЙ МОНИТОР", входящая в состав программного обеспечения ПЭВМ. В отличие от остальных программ эта программа не загружается в ПЭВМ с магнитофона, а находится в постоянном запоминающем устройстве (ПЗУ) системного блока. Для запуска этой программы достаточно нажать кнопку "СБРОС". С помощью программы "СИСТЕМНЫЙ МОНИТОР" производится загрузка с магнитной ленты и запуск программ, входящих в комплект ПЭВМ, составление и отладка программ пользователя. Поэтому, прежде чем приступать к работе с этими программами, необходимо изучить порядок работы в "СИСТЕМНОМ МОНИТОРЕ".

7.1.2. Директивы "СИСТЕМНОГО МОНИТОРА"

Программа "СИСТЕМНЫЙ МОНИТОР" выполняет следующие функции.

1) При каждом нажатии клавиши "СБРОС" программа осуществляет старт и начальную установку ПЭВМ. При этом очищается экран, в левом верхнем углу экрана появляется надпись "МИКРОША", знак "->" и мигающий курсор. В ячейки экранной области оперативной памяти записываются нули. Содержимое ячеек с номерами от 0 до 75FF не изменяется, то есть при нажатии кнопки "СБРОС" ранее введенные в ОЗУ программы сохраняются, но выполнение их прекращается.

2) Программа обеспечивает обмен информацией с устройствами ввода-вывода (магнитофоном, телевизором, клавиатурой).

3) Программа представляет пользователю возможности для просмотра, сравнения и изменения содержимого оперативной памяти и регистров процессора, запуск и отладку находящихся в оперативной памяти программ (диалог в "СИСТЕМНОМ МОНИТОРЕ").

4) В состав "СИСТЕМНОГО МОНИТОРА" входят подпрограммы, которые можно использовать при составлении программ пользователя.

Для организации диалога в "СИСТЕМНОМ МОНИТОРЕ" имеется 11 типов директив. Директивы обозначаются буквами латинского алфавита. При необходимости после директивы указываются адреса и данные в шестнадцатеричном виде. Данные и адреса разделяются запятыми. Адреса задаются

четырёхзначными числами, данные - двухзначными числами. Увеличение количества знаков при записи адресов и данных приводит к неправильному выполнению директивы. Уменьшение указанного количества знаков воспринимается "СИСТЕМНЫМ МОНИТОРОМ" как наличие нулей в старших разрядах (слева от записанных значений адреса или данных). Например, директива "D100,1000" воспринимается как "D0100,1000". Отсутствие знаков воспринимается "СИСТЕМНЫМ МОНИТОРОМ" как соответствующее количество нулей, кроме особых случаев, которые будут рассмотрены в дальнейшем. Например, директива "F,1000" соответствует "F0000,1000".

Для выполнения директивы после ее набора нажмите "BK".

Признаком окончания выполнения любой из директив "СИСТЕМОГО МОНИТОРА" является появление на экране знака "->" и мигающего курсора.

В случае, если при наборе директивы произошла ошибка, можно отредактировать набранное выражение, переместив курсор влево по строке с помощью клавиши "←" (стрелка влево) и повторив набор. Следует иметь в виду, что при редактировании директив "МОНИТОРА" не допускается перемещать курсор вправо, вверх и вниз клавишами "→", "↑", "↓". При редактировании с помощью этих клавиш директива не выполняется.

При наборе директив "СИСТЕМОГО МОНИТОРА" нельзя вставлять символ "ПРОБЕЛ" ни перед директивой, ни между директивой и параметрами, не допускается пользоваться русскими буквами вместо латинских.

В случае ошибки при наборе директивы слева от набранного выражения появляется вопросительный знак, например:

```
?>D,10000  
->
```

Далее рассмотрим директивы "МОНИТОРА" и их особенности.

1) Директива "I" (латинская буква) - ввод данных в ПЭВМ с магнитофона. Позволяет вводить в ПЭВМ программы, записанные в кодах микропроцессора. Применение директивы "I" описано в разделе 6.4. Остается добавить, что директива позволяет загружать программы со смещением адресов, для чего после "I" необходимо набрать на клавиатуре адрес, на который необходимо сместить программу при загрузке.

Например, программа записана с адресами от 100 до 200. Если при загрузке набрать на клавиатуре

```
I1000
```

то программа загрузится в область памяти от 1100 до

1200. При этом после загрузки на экране появятся новые адреса:

1100
1200

и контрольная сумма.

Однако таким образом целесообразно загружать файлы данных или программы при отладке, так как при загрузке со смещением адресация команд становится неправильной и программа не выполняется.

2) Директива "D" - просмотр интервала памяти. После "D" наберите адрес начала интервала памяти, затем запятую и адрес конца интервала памяти, который Вы хотите просмотреть, после чего нажмите "BK". На экране появляются 16 столбцов двухзначных шестнадцатеричных чисел - данные, в дефисной части экрана - столбец четырехзначных чисел - адреса, соответствующие левому столбцу данных. Адреса ячеек, в которых находятся данные из других столбцов, можно определить, прибавив к адресу строки число, соответствующее адресу столбца. Например, если набрать директиву:

DF800,F820 BK

на экране появится

F800 C3 1B F8 C3 D0 FE C3 0D FC C3 25 FD C3 AB FC C3
F810 25 FD C3 7F FE C3 42 FA C3 AD F9 3E 98 32 03 C0
F820 3E

Число 25 в верхней строке расположено по адресу F80A, в средней строке - по адресу F810, число 7F в средней строке - по адресу F813 и т.д.

При выполнении директивы "D" на экран выводится одновременно не более 24 строк, или 383 адреса (в шестнадцатеричной системе - 180 адресов). Если Вы просматриваете большой интервал памяти, таблица данных и адресов на экран поднимается снизу вверх, и начальная часть таблицы уходит с экрана. Чтобы остановить на время просмотра движение таблицы, нажмите и удерживайте в нажатом состоянии клавишу "РУС/LAT". При отпускании клавиши движение таблицы возобновится. Чтобы не включался звуковой сигнал при переключении на русский регистр, можно удерживать в нажатом состоянии клавишу "НР" и нажимать клавишу "РУС/LAT". В этом случае при первом нажатии на "РУС/LAT" движение таблицы прекращается, при втором - возобновляется, и т.д. Для полной остановки таблицы нажмите на клавишу "F4".

3) Директива "I" - просмотр интервала памяти в символьном виде. Эта директива аналогична предыдущей, но данные выводятся на экран не в шестнадцатеричной форме,

а в виде алфавитно-цифровых символов знакогенератора, соответствующих кодам, записанным в просматриваемых адресах (см. табл. 5). При этом отображаются в виде символов только коды от 20 до 7F (алфавитно-цифровые). Вместо кодов от 0 до 1F (псевдографические символы и управляющие коды) и кодов от 80, не отображающих символов знакогенератора, в таблице ставятся точки. Остановка движения таблицы при выполнении директивы "L" осуществляется аналогично директиве "D".

Пример выполнения директивы монитора "L".

При наборе

"LF800,F820" ВК

на экране появляется таблица

```
F800 .....%.....  
F810 SS..П..В....>.2..  
F820 >
```

4) Директива "М" - последовательный просмотр и изменение данных в ячейках памяти. С помощью этой директивы можно исправлять программы, составленные в кодах микропроцессора, или составлять небольшие программы, а также управлять режимами работы ПЭВМ. Нажимите клавишу "М", наберите адрес ячейки, содержимое которой Вы хотите просмотреть или изменить, нажимите "ВК". На экране появляется адрес, рядом-двухзначное шестнадцатеричное число, представляющее собой содержимое ячейки с указанным адресом. Если Вы нажмете "ВК", появляется следующий по порядку адрес и следующее число. При этом содержимое ячейки с предыдущим адресом не меняется. Если набрать на клавиатуре двухзначное число и нажать "ВК", это число запишется по указанному слева адресу взамен предыдущего. При наборе однозначного числа оно воспринимается как двухзначное с нулем в старшем разряде. Выполнение директивы прекращается нажатием клавиши с точкой.

Пример выполнения директивы монитора "М".

При наборе

M1000 ВК

на экране появляется

1000 00

при нажатии "ВК"

```
1000 00  
1001 FF
```

Изменим содержимое ячеек с адресами от 1001 до 1009:

```
1000 00
1001 FF 55 BK
1002 00 55 BK
1003 FF 55 BK
1004 00 55 BK
1005 FF 55 BK
1006 00 55 BK
1007 FF 55 BK
1008 00 55 BK
1009 FF 55 BK
100A 00.
->
```

Выполнение директивы "M" проверим с помощью директивы "D":

```
D1000,100A BK
1000 00 55 55 55 55 55 55 55 55 00
```

5) Директива "F" - запись числа во все ячейки интервала памяти. Нажмите клавишу "F", наберите адрес начала интервала памяти, адрес конца интервала и двухзначное шестнадцатеричное число, которое необходимо записать в ячейки данного интервала. Адреса и число (данные) разделяются запятыми, например:

```
F1000,1100,AA BK
```

В результате выполнения этой директивы в каждую ячейку с адресом от 1000 до 1100 запишется число AA. Выполнение директивы можно проверить с помощью директивы "D". Если после набрать запятую, а затем набрать адрес и число, данное число запишется во все ячейки памяти от нуля до набранного числа. Если не набирать число, во все ячейки заданного интервала памяти запишутся нули.

6) Директива "T" - копирование содержимого интервала памяти (пересылка содержимого одной области памяти я другую): Нажмите клавишу "T", наберите адрес начала копируемого интервала, адрес конца копируемого интервала и адрес начала интервала памяти, в который необходимо переслать содержимое первого интервала. Адреса разделяются запятыми. Например:

```
T100,1FF,1500 BK
->
```

В результате выполнения этой директивы содержимое интервала памяти от 100 до 1FF пересылается в область памяти от 1500 до 15FF. Выполнение директивы можно проверить с помощью директивы "D", просмотрев оба

интервала памяти и сравним их. Копирование осуществляется побайтно, начиная с младшего адреса. В случае, если начало второго интервала памяти находится внутри первого интервала, копирование происходит правильно только в пределах первого интервала, а затем повторяется начало первого интервала. Например, в области памяти от 100 до 104 записаны числа 1, 2, 3, 4, 5. (Можно записать эти числа с помощью директивы "М"). Если набрать директиву

T100,104,102 BK

а затем проверить содержимое памяти от 100 до 106 с помощью директивы "D", получим:

0100 01 02 01 02 01 02 01
->

Если начальный адрес второго интервала памяти меньше начального адреса первого интервала, а конечный адрес второго интервала находится внутри первого интервала, то начальная часть первого интервала стирается. Например

->T100,104,FD BK

В результате выполнения директивы получим

00FD 01 02 03
0100 04 05 03 04 05
->

7) Директива "С" — сравнение двух интервалов памяти. Формат записи этой директивы аналогичен формату директивы "Т".

Пример.

->C100,1FF,1500 BK

В случае совпадения содержимого интервалов 100- 1FF и 1500-15FF после выполнения директивы на экране появляется знак "->" и курсор.

При несовпадении данных на экране появляется адрес ячейки из первого интервала, ее содержимое и содержимое ячейки из второго интервала.

Пример.

С помощью директивы "F" запишем во все ячейки от 0 до 75FF число 55.

F,75FF,55 BK

Сравним интервалы от 0 до 3AFF и от 3B00 до 75FF.

C,3AFF,3B00 BK

Если предыдущая директива выполнена правильно, на экране появится:

->
и курсор.

С помощью директивы "M" изменим содержимое нескольких ячеек памяти в одном из сравниваемых интервалов:

M1000 BK
1000 55 66 BK
1001 55 66 BK
1002 55 00 BK
1003 55.
->

Выполнив указанные действия, мы изменим содержимое ячеек с адресами 1000, 1001, 1002.

Теперь повторим сравнение интервалов с помощью директивы "C".

C,3AFF,3B00 BK

В результате выполнения директивы на этот раз получим сообщение о несовпадении содержимого ячеек 1000, 1001, 1002 с содержимым соответствующих ячеек второго интервала (4B00, 4B01, 4B02):

1000 66 55
1001 66 55
1002 00 55
->

8) Директива "S" - поиск кода в заданной области памяти. Наберите "S", адреса начала и конца области памяти, в которой Вы хотите найти заданный код, затем сам код. Адреса и код необходимо разделить запятыми.

Пример.

Рассмотрим предыдущий пример с записью кода 55 во все ячейки от 0 до 75FF и изменением кодов в ячейках 1000, 1001, 1002. С помощью директивы "S" Вы можете найти, в каких ячейках памяти в интервале от 0 до 75FF имеются коды 66.

S,75FF,66 BK

В результате выполнения директивы получим

1000
1001
->

Таким образом, выявлены адреса ячеек, в которых содержатся коды 66.

Если заданного кода нет в проверяемом интервале памяти, после выполнения директивы появляется знак -> и курсор.

9) Директива "O" - вывод содержимого области памяти на магнитную ленту. Выполнение директивы описано в разделе 6.4 настоящего руководства по эксплуатации.

Приведем еще один пример.

Запишите в интервал памяти от 0 до 100 коды 55:

F,100,55 BK

Запишите содержимое интервала памяти от 0 до 100 на магнитную ленту, для чего подключите магнитофон к ПЭВМ в соответствии с разделом 6.4. Наберите на клавиатуре O,100

Установите в магнитофон кассету без записей, включите магнитофон в режим записи и нажмите "BK".

Экран гаснет, прослушивается тональный сигнал в динамике магнитофона, а затем тональный сигнал другой частоты (коды 55) в динамиках магнитофона и ПЭВМ.

Начальная часть тонального сигнала (синхросигнал) имеет место при записи любой программы. Последующий сигнал звучит как тональный из-за однородной информации, записанной в интервале от 0 до 100.

После окончания записи на экране появляется набранная директива и контрольная сумма записанного набора данных (файла)

0,100
0055
->

Правильность записи можно проверить с помощью директивы "I" в соответствии с разделом 6.4.

10) Директива "G" - запуск программ, составленных в кодах микропроцессора. С помощью директивы "G" можно запустить в работу любую отложенную программу, составленную в кодах микропроцессора и хранящуюся в оперативной памяти ПЭВМ. Для этого необходимо знать начальный адрес этой программы, т.е. Адрес команды, которая исполняется в программе первой. Обычно начальный адрес программы совпадает с начальным адресом области памяти, в которой расположена эта программа. Начальный адрес набирается после "G".

Пример.

Загрузите с помощью директивы "I" программу "Тетрис". В случае успешной загрузки программы наберите на клавиатуре директиву

G0000 BK

После этого на экране появляется название программы "Тетрис" и указания по дальнейшим действиям. Это означает, что программа запущена. Если начальный адрес равен нулю, его можно не набирать:

G BK

При отладке программ в кодах с помощью директивы "G" можно задать не только адрес запуска, но и адрес остановки программы.

Пример.

Если необходимо запустить программу с адреса 100 и остановить на адресе 110, наберите на клавиатуре

G100,110 BK

(адрес останова записывается через запятую после адреса запуска).

С помощью директивы "M" запишем код 00 в ячейки с адресами от 100 до 110. Код 00 соответствует команде NOP микропроцессора - отсутствие операции (система команд микропроцессора описана далее). В этом случае после выполнения директивы получим

G100,110
0110
->

Во второй строке выражения появляется код, содержащийся в регистре PC микропроцессора (0110). Если в запущенной программе нет переходов по адресам вне заданного интервала (в данном случае от 100 до 110), то содержимое регистра PC совпадает с заданным в директиве адреса останова.

Программы, составленные на языке программирования Бейсик, с помощью директивы "G" не запускаются.

11) Директива "X" - просмотр и изменение содержимого регистров микропроцессора. О регистрах микропроцессора можно будет прочитать в разделе "Структурная схема ПЭВМ".

Для выполнения директивы необходимо набрать на клавиатуре

X BK

На экране появляется перечень регистров микропроцессора и содержимое регистра РС, например:

```
PC-FFFF
HL-
BC-
DE-
AF-
SP-
```

Набирая на клавиатуре четырехзначные шестнадцатеричные числа, можно изменять содержимое регистров. После набора каждого числа нажимайте "BK". Пропущенные знаки воспринимаются как нули в старших разрядах. После каждого нажатия "BK" появляется содержимое следующего регистра. Если необходимо проверить содержимое регистра, не изменяя его, нажмите "BK". После шести нажатий "BK" выполнение директивы прекращается.

Пример:

```
PC-FFFF 0000 BK
HL-0000 BK
BC-0030 BK
DE-0000 1111 BK
AF-6006 BK
SP-76CF BK
->
```

После выполнения указанной директивы можно убедиться в том, что содержимое регистров РС и DE изменилось, повторив выполнение директивы "X".

7.1.3. Подпрограммы "СИСТЕМНОГО МОНИТОРА"

В состав "СИСТЕМНОГО МОНИТОРА" входят стандартные подпрограммы, к которым может обращаться программа пользователя. Далее приведены адреса в шестнадцатеричном виде и краткие характеристики подпрограмм монитора.

После описания некоторых подпрограмм приведены примеры небольших программ, в которых используются подпрограммы "СИСТЕМНОГО МОНИТОРА". Набрать указанные программы можно с помощью директивы "СИСТЕМНОГО МОНИТОРА" - "M". В правой части примеров приведены мнемонические обозначений команд. Эти примеры Вы сможете разобрать после изучения системы команд микропроцессора.

1) Старт "СИСТЕМНОГО МОНИТОРА" - F800. Программа производит начальную установку ПЭВМ. Выполняется при нажатии кнопки "СБРОС" или выполнении директивы

"GF800" BK

2) Ввод символа с клавиатуры - F803. Код нажатой клавиши помещается в регистр А (аккумулятор).

Пример 1. Наберите с помощью директивы "M" программу

```
0100 CD CALL F803
0101 03
0102 F8
```

Нажмите кнопку "СБРОС", затем наберите директиву запуска программы:

```
G100,103 BK
```

В результате перехода в подпрограмму "СИСТЕМНОГО МОНИТОРА" (CALL F803) ПЭВМ находится в режиме ожидания. После нажатия "BK" курсор перемещается в начало новой строки. Нажмите любую клавишу, например "U". После нажатия клавиши курсор перемещается на следующую строку и появляется символ ПЭВМ находится в режиме "СИСТЕМНОГО МОНИТОРА".

С помощью директивы "X" проверьте содержимое регистра А. В регистре А должен быть записан код 55, соответствующий нажатой клавише "U" (см. табл.5). Можно повторить пример с другой клавишей, предварительно нажав кнопку "СБРОС".

Пример 2

```
0100 CD CALL F803  обращение к подпрограмме
0101 03
0102 F8
0103 FE CPI 08      сравнение содержимого
0104 08             аккумулятора с кодом 08
0105 C2 JNZ F800    переход по ненулевому
0106 00             результату в "МОНИТОР"
0107 F8
0108 3E MVI 7F      запись в аккумулятор кода 7F
0109 7F
010A 32 STA 7A00     запись содержимого
010B 00             аккумулятора в ячейку
010C 7A             7A00 (экранная область)
010D C3 JMP F89D     переход в "МОНИТОР"
010E 9D             без сброса
010F F8
```

Наберите директиву запуска программы

```
G100 BK
```

курсор перемещается в начало следующей строки. Если нажать клавишу "-" с кодом 08 (см. табл.3), код 7F

записывается в аккумулятор, а из аккумулятора - в ячейку 7A00, находящуюся в экранной области памяти. При этом на экране появляется квадрат (код 7F). Если нажать на любую другую клавишу, происходит выход в "СИСТЕМНЫЙ МОНИТОР" со сбросом (соответствует нажатию кнопки

"СБРОС").

3) Ввод одного байта с магнитофона - F806.

Входные параметры: если в аккумуляторе перед обращением к подпрограмме содержится код 08, программа выполняется без поиска синхробайта, если F0 - с поиском синхробайта. Вводимый байт записывается в аккумулятор.

4) Вывод символа на экран - F809 или F80F. Код выводимого на экран символа должен быть записан в регистре С.

Пример.

```
0100 CD CALL F809 обращение к подпрограмме
0101 09
0102 F8
0103 C3 JMP F89D переход в монитор
0104 9D
0105 F8
```

С помощью директивы "X" запишите в регистр С код какого-либо символа, например 7F. При запуске программы по директиве "G100" ВК, на экране слева от набранной директивы появится квадрат (код 7F).

5) Запись байта на магнитофон - F80C. Выводимый байт должен находиться в регистре С.

Пример:

```
0100 CD CALL F80C обращение к подпрограмме
0101 0C
0102 F8
0103 C3 JMP 0100 переход к началу программы
0104 00
0105 01
```

С помощью директивы "X" запишите в регистр С какой-либо код, запустите программу по G100 ВК и запишите код из регистра С на магнитофон. Запись будет происходить до нажатия кнопки "СБРОС".

6) Опрос состояния клавиатуры - F812.

Если клавиша не была нажата, в аккумуляторе 0, если нажата - FF.

Пример.

```
0100 21 LXI 7800->NL запись адреса 7800
0101 00 в регистровую пару NL
0102 78
0103 CD CALL F812 обращение к подпрограмме
0104 12
0105 F8
0106 FE CPI FF сравнение содержимого
0107 FF аккумулятора с FF
0108 CA JZ F89D переход в "МОНИТОР" по
0109 9D нулевому результату 010A F8
```

```

010B 71 MOV C->M запись содержимого регистра C
                        в память (адрес - в HL)
010C 23 INX H         увеличение содержимого HL на 1
010D C3 JMP 0103      переход по адресу 0103
010E 03
010F 01

```

С помощью директивы "X" запишите в регистр C какой-либо код, например 44, и осуществите запуск программы директивой "G100" ВК. Экран начнет заполняться буквами D (код 44). Если нажать любую клавишу, заполнение прекратится, т.к. при нажатии клавиши подпрограмма опроса клавиатуры (F812) записывает в аккумулятор код FF, в результате выполнения команды CPI FF получается нулевой результат и ПЭВМ переходит в "СИСТЕМНЫЙ МОНИТОР" (F89D) по команде JZ.

7) Вывод байта на экран в шестнадцатеричном виде - F815. Выводимый код помещается в аккумулятор.

Пример 1:

```

0100 CD CALL F815 обращение к подпрограмме
0101 15
0102 F8
0103 C3 JMP F89D выход в "СИСТЕМНЫЙ МОНИТОР"
0104 9D
0105 F8

```

С помощью директивы "X" запишите в аккумулятор любой код и запустите подпрограмму директивой "G100" ВК. Слева от набранной директивы на экране появится записанный в аккумулятор код.

Пример 2:

В этом примере используются сразу четыре подпрограммы. С помощью данной программы можно получить код любого символа при нажатии соответствующей клавиши.

```

0100 CD CALL F812 обращение к подпрограмме
0101 12 опроса клавиатуры
0102 F8
0103 B7 ORA A проверка наличия нажатой клавиши
0104 CA JZ 0100 переход к началу программы по
0105 00 нулевому результату
0106 01 (клавиша не нажата)
0107 CD CALL F803 обращение к подпрограмме ввода
0108 03 символа с клавиатуры
0109 F8
010A CD CALL F815 обращение к подпрограмме вывода
010B 15 байта на экран в
010C F8 шестнадцатеричном коде
010D 0E MVI 0A->C запись в регистр C кода 0A (перевод
010E 0A строки)
010F CD CALL F809 обращение к подпрограмме вывода

```

0110 09	символа на экран
0111 F8	
0112 0E MVI 0D->C	запись в регистр C кода 0D (возврат
0113 0D	каретки)
0114 CD CALL F809	обращение к подпрограмме вывода
0115 09	символа на экран
0116 F8	
0117 C3 JMP 0100	переход к началу программы
0118 00	
0119 01	

Запуск программы - по директиве "G100" BK. После запуска при нажатии любой клавиши слева от директивы запуска появляется код нажатой клавиши. При нажатии другой клавиши ее код появляется под предыдущим кодом. Это достигается в программе с помощью двух обращений к подпрограмме вывода символа на экран (F809). При первом обращении происходит переход на следующую строку (код 0A соответствует PC), при втором - переход к началу строки (код 0D соответствует BK). Если исключить первое обращение в F809, новые коды будут появляться на месте старых, без смещения. При исключении второго обращения в F809 коды будут выводиться на экран "по диагонали", со смещением на следующую строку и на следующую позицию по горизонтали.

8) Подпрограмма вывода на экран сообщения - F818. Выводит на экран символы, коды которых содержатся в области памяти до тех пор, пока не встретится код 00. Адрес начала области памяти должен быть в регистровой паре HL.

Пример.

С помощью директивы "F" запишите в область памяти от 0 до FF коды 55:

F,FF,55 BK

В ячейку 100 с помощью директивы "M" запишите код 00, в регистровую пару HL с помощью директивы "X" - коды 0000 (адрес 54 начала области памяти);
наберите директиву

GF818 BK

При этом на экране появятся 4 строки букв "U" (код 55).

9) Вход в "СИСТЕМНЫЙ МОНИТОР" без сброса - F89D.

10) Ввод кода нажатой клавиши - FEEA. При выполнении этой подпрограммы в аккумулятор заносится код клавиши, если последняя была нажата в момент обращения, или код FF если никакая из клавиш не была нажата. Отличие этой подпрограммы от F803 заключается в том, что ввод кода производится без ожидания.

7.2. Система команд микропроцессора

Система команд микропроцессора КР580ВМ80 является основой, на которой построено все программное обеспечение ПЭВМ. С помощью системы команд можно составлять различные программы пользователя, используя директивы "СИСТЕМНОГО МОНИТОРА". Особенностью ПЭВМ является взаимосвязь между ее устройством и программным обеспечением, поэтому прежде чем перейти к изучению системы команд и рассмотрению примеров программ, составленных в машинных кодах, необходимо кратко ознакомиться со структурной схемой ПЭВМ.

7.2.1 Структурная схема ПЭВМ

Структурная схема ПЭВМ "Микроша" приведена на рис.3. В составе компьютера можно выделить следующие функциональные части:

- 1) Системные шины
- 2) Тактовый генератор и формирователь сигнала "СЕРОС"
- 3) Микропроцессор
- 4) Узел начального запуска
- 5) Узел выбора памяти или устройства ввода/вывода
- 6) Постоянное запоминающее устройство (ПЗУ) "МОНИТОРА"
- 7) Оперативное запоминающее устройство (ОЗУ)
- 8) Контроллер прямого доступа в память (ПДП)
- 9) Контроллер дисплея
- 10) Контроллер клавиатуры
- 11) Клавиатура
- 12) Таймер
- 13) Программируемый параллельный интерфейс
- 14) Схема сопряжения с магнитофоном

Системные шины микропроцессора функционально разделены на три группы: шина адреса, шина данных, шина управления. По шине адреса передаются шестнадцатиразрядные адреса памяти или устройств ввода-вывода. Младший бит адреса передается по линии А0, старший бит - по линии А15. По двунаправленной восьмиразрядной шине данных происходит обмен информацией между отдельными частями ПЭВМ. Младший бит данных передается по линии D0, а старший - по линии D7. Шина управления служит для передачи управляющих и синхронизирующих сигналов.

Тактовый генератор, собранный на микросхеме КР580ГФ24 (D1), синхронизирует работу процессора и других частей ПЭВМ. На элементах R3, C2, S1 собран формирователь сигнала "СЕРОС". Этот сигнал необходим для начальной установки регистров микропроцессора.

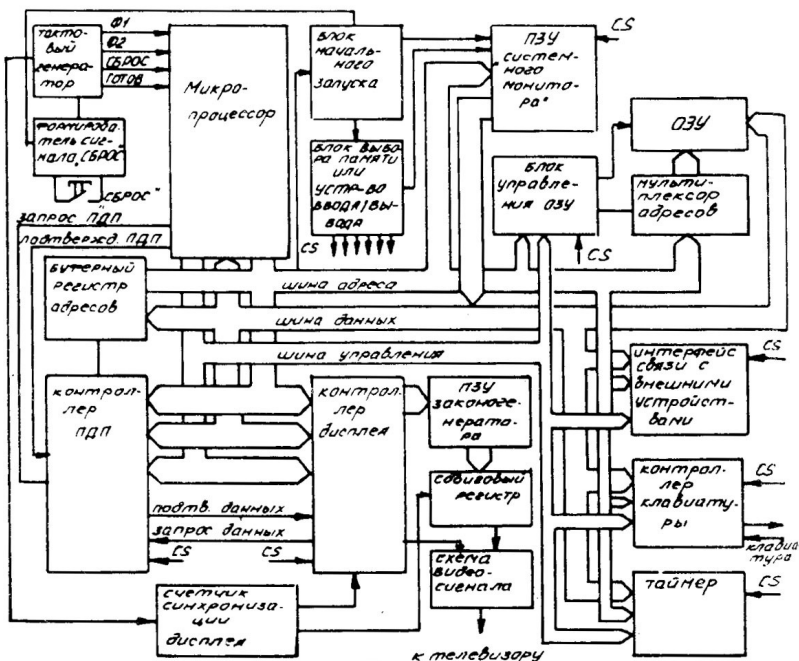


Рис. 3

Микропроцессор КР580ВМ80А (D5) осуществляет адресацию исполнительных устройств, обмен с ними информацией по шине данных, логическую и арифметическую обработку данных, выполнение программ.

Так как после получения сигнала "СБРОС" микропроцессор начинает выполнять программу, расположенную с адреса 0000, а для программы "СИСТЕМНЫЙ МОНИТОР" отведена область памяти с адреса F800, в компьютер введен узел начального запуска (элементы D9.2, D10.2, D14.1). Узел начального запуска обеспечивает автоматическую передачу управления программе "СИСТЕМНЫЙ МОНИТОР" (D15) при сбросе микропроцессора и влияния на дальнейшую работу МП не оказывает.

Основным элементом блока выбора памяти или устройства ввода-вывода является дешифратор D11. В зависимости от состояния линий A11-A15 шины адреса происходит формирование сигнала CS (выбор микросхемы) на соответствующих входах устройств, к которым происходит обращение в текущий момент времени.

ПЗУ "СИСТЕМНОГО МОНИТОРА", выполненное на микросхеме К573РФ2, предназначено для хранения кодов управляющей программы.

ОЗУ компьютера выполнено на микросхемах памяти динамического типа КР565РУ6 (D23-D38). Код адреса

заносятся в адресный регистр микросхем через входы A0-A6 последовательно: сначала поступают коды семи младших разрядов адреса, а затем семи старших, сопровождаемые соответственно сигналами выборки строки RAS и столбца CAS. Временное мультиплексирование адресов выполняется микросхемами D19, D20. Для формирования сигналов, управляющих работой динамической памяти, служит узел, собранный на микросхемах D17, D18 и элементах D4.4, D6.4, D10.2, D10.3.

Контроллер ПДП предназначен для пересылки данных из ОЗУ в контроллер дисплея. В состав контроллера входит микросхема KP580BT57 (D2) и многорежимный регистр (D7), предназначенный для временного хранения восьми старших разрядов кода адреса.

Контроллер дисплея предназначен для формирования комплексного телевизионного видеосигнала. Контроллер состоит из БИС KP580BT75 (D8), ПЗУ знакогенератора K573P42 (D13), счетчика-формирователя сигнала синхронизации дисплея K155IE4 (D3), сдвигового регистра K155IP13 (D16) и схемы формирования комплексного видеосигнала, собранной на дискретных элементах.

Контроллер клавиатуры предназначен для формирования сигналов обмена информацией с клавиатурой и магнитофоном. Контроллер выполнен на БИС KP580BB55 (D39). Кроме этого на контроллер клавиатуры возложены функции управления динамической головкой и светодиодом "РУС/LAT".

Блок клавиатуры подробно описан в разделе 6.3 настоящего руководства.

Программируемый таймер KP580BI53 (D22) предназначен для формирования временных интервалов.

Программируемый параллельный интерфейс D12 выполнен на микросхеме KP580BB55 и обеспечивает подключение к ПЭВМ различных внешних устройств, осуществляющих обмен данными в параллельном коде: АЦП, ЦАП, печатающего устройства и т.д. Кроме этого через микросхему D12 осуществляется и последовательный ввод-вывод.

Схема сопряжения с магнитофоном выполнена на дискретных элементах R36-R40, C5-C10, D40, VD2. Ее назначение - обеспечить уверенную запись и считывание цифрового сигнала с носителя на магнитной ленте.

В схеме ПЭВМ предусмотрена возможность расширения функциональных характеристик системы путем подключения дополнительных устройств к внутренним системным шинам. Подключение производится с помощью разъема "Внутренний интерфейс", для дополнительных устройств, в качестве которых могут выступать ячейки дополнительного ОЗУ или ПЗУ, отведены особые области памяти 8000-BFFF, E000-F7FF.

7.2.2. Общие сведения о внутренней структуре микропроцессора KP580BM80A

Микропроцессор KP580BM80A имеет сложную внутреннюю структуру, однако с точки зрения пользователя он состоит только из семи восьмиразрядных регистров, получивших условные названия А, В, С, D, Е, H, L, регистра признаков результата F и двух шестнадцатиразрядных регистров SP и PC.

Регистр А, называемый также аккумулятором, используется для хранения операнда, с которым работает арифметико-логическое устройство (АЛУ) микропроцессора. Результат обработки также помещается в аккумулятор. Принципы организации и функционирования АЛУ и других недоступных пользователю элементов МП в настоящем описании не рассматриваются.

Шесть регистров В, С, D, Е, H, L предназначены для хранения промежуточных данных. При исполнении некоторых команд регистры могут попарно объединяться в так называемые регистровые пары ВС, DE, HL. Это делает возможным обработку шестнадцатиразрядных данных. Регистр признаков F необходим для принятия решения по результату выполнения той или иной команды. В регистре F вырабатываются признаки состояния, при этом устанавливаются в единицу следующие биты регистра F:

1) бит Z - признак нуля, устанавливается, если результат выполнения команды равен 0;

2) бит S - признак отрицательного результата, устанавливается, если результат выполнения команды отрицателен;

3) бит P - признак четности, устанавливается, если количество единиц в двоичном коде результата команды четное;

4) бит C - признак переноса, устанавливается, если в результате сложения двух восьмиразрядных чисел возникает перенос из старшего разряда или в результате вычитания возникает заем;

5) бит AC - признак вспомогательного переноса, устанавливается, когда перенос возникает из четвертого разряда двоичного числа. Этот признак используется при операциях с четырехразрядными операндами.

Формат регистра F приведен ниже.

S	Z	0	AC	0	P	1	C
---	---	---	----	---	---	---	---

При обращении к памяти для чтения кода очередной команды из микропроцессора на шину адреса поступает шестнадцатиразрядное содержимое регистра PC, называемого счетчиком команд. В этом регистре к моменту окончания выполнения текущей команды всегда подготавливается адрес очередной команды программы. В процессе выполнения

программы МП часто приходится обращаться к ячейкам ОЗУ для записи к чтению промежуточных данных. В системе команд имеются команды, с помощью которых можно задать адрес обращения к памяти непосредственно, то есть произвести непосредственную адресацию. Они имеют трехбайтовый формат и, следовательно, занимают три последовательно расположенных в памяти ячейки, в первом байте команды хранится код операции, а во втором и третьем записан шестнадцатиразрядный адрес обращения к памяти. При выполнении такой команды МП последовательно считывает значения второго и третьего байтов во внутренние буферные регистры и затем передает их на шину адреса. Например, для записи содержимого аккумулятора в ячейку памяти с адресом 5000 можно использовать команду с непосредственной адресацией STA 5000.

В системе команд есть также команды, использующие косвенную регистровую адресацию. При их выполнении адресация осуществляется по содержимому одной из регистровых пар BC, DE, HL, куда предварительно помещается адрес требуемой ячейки памяти и откуда он поступает на шину адреса. Записать содержимое аккумулятора в ячейку 5000 можно с помощью команды STAX D, предварительно занеся в регистровую пару DE число 5000.

Кроме описанных двух способов адресации, возможна адресация к ячейкам памяти по содержимому шестнадцатиразрядного регистра SP, называемого указателем стека.

Под стеком в микропроцессоре KP580BM80A понимается любая область ОЗУ, служащая для хранения адресов и промежуточных данных, адресация к ячейкам которой осуществляется с помощью указателя стека SP. Запись и чтение из стека производится по принципу "первый вошел - последний вышел". Перед выполнением команды, использующей регистр SP, в него предварительно должен быть записан код начала стековой области (код вершины стека). В ПЭВМ "Микроша" адрес начала стека (76CF) заносится в указатель стека автоматически при поступлении сигнала СБРОС. С помощью стековой адресации в стек может быть записано содержимое любой регистровой пары или программного счетчика PC. Запись в стек происходит побайтно: сначала записывается старший байт в ячейку памяти с адресом, на единицу меньшим содержимого указателя стека (т.е. с адресом SP-1), затем - младший байт в ячейку с адресом SP-2. Таким образом при записи указатель стека смещается в область меньших адресов. При чтении данных из стека сначала производится загрузка младшего регистра пары младшим байтом данных, а затем в старший регистр заносится старший байт данных. Одновременно содержимое регистра SP смещается область старших адресов.

7.2.3. Описание системы команд микропроцессора КР580ВМ80А

Система команд это полный перечень элементарных действий, которые способен производить микропроцессор. Управляемый этими командами МП выполняет очень простые действия, такие как элементарные арифметические и логические операции, операции сравнения и пересылки данных. С помощью этих простых команд можно запрограммировать выполнение алгоритма любой сложности.

Микропроцессор КР580ВМ80А имеет фиксированную систему команд, причем каждой команде соответствует определенный машинный код (см. табл.6). Код операции каждой команды в шестнадцатеричном виде приведен в левом столбце (старшие разряды) и верхней строке (младшие разряды). Например команде вызова подпрограммы CALL соответствует машинный код CD. Для облегчения восприятия текстов программ, команды микропроцессора представлены в мнемонической форме (мнемокоде), то есть мнемокод команды отражает выполняемые действия. Например, команда загрузки регистра вторым байтом команды имеет мнемонику MVI (MOVE IMMEDIATE), а команда записи содержимого аккумулятора в память — LDA (LOAD DIRECT). Мнемокоды команд и их краткие характеристики приведены в табл.7.

СИСТЕМА КОМАНД МИКРОПРОЦЕССОРА КР580ВМ80А

Таблица 6

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	NOP	LXI B, &	STAX B	INX B	INR B	DCR B	MVI B, #	RLC	-	DAD B	LDAX B	DCX B	INR C	DCR C	MVI C, #	RRC
1	-	LXI D, #	STAX D	INX D	INR D	DCR D	MVI D, #	RAL	-	DAD D	LDAX D	DCX D	INR E	DCR E	MVI E, #	RAR
2	-	LXI H	SHLD *	INX H	INR H	DCR H	MVI H, #	DAA	-	DAD H	LHLD H	DCX H	INR L	DCR L	MVI L, #	CMA
3	-	LXI SP, #	STA *	INX SP	INR M	DCR M	MVI M, #	STC	-	DAD SP	LHLD *	DCX SP	INR A	DCR A	MVI A, #	CMC
4	MOV B, B	MOV B, C	MOV B, D	MOV B, E	MOV B, H	MOV B, L	MOV B, M	MOV B, A	MOV C, B	MOV C, C	MOV C, D	MOV C, E	MOV C, H	MOV C, L	MOV C, M	MOV C, A
5	MOV D, B	MOV D, C	MOV D, D	MOV D, E	MOV D, H	MOV D, L	MOV D, M	MOV D, A	MOV E, B	MOV E, C	MOV E, D	MOV E, E	MOV E, H	MOV E, L	MOV E, M	MOV E, A
6	MOV H, B	MOV H, C	MOV H, D	MOV H, E	MOV H, H	MOV H, L	MOV H, M	MOV H, A	MOV L, B	MOV L, C	MOV L, D	MOV L, E	MOV L, H	MOV L, L	MOV L, M	MOV L, A
7	MOV M, B	MOV M, C	MOV M, D	MOV M, E	MOV M, H	MOV M, L	HLT	MOV M, A	MOV A, B	MOV A, C	MOV A, D	MOV A, E	MOV A, H	MOV A, L	MOV A, M	MOV A, A
8	ADD B	ADD C	ADD D	ADD E	ADD H	ADD L	ADD M	ADD A	ADC B	ADC C	ADC D	ADC E	ADC H	ADC L	ADC M	ADC A
9	SUB B	SUB C	SUB D	SUB E	SUB H	SUB L	SUB M	SUB A	SBB B	SBB C	SBB D	SBB E	SBB H	SBB L	SBB M	SBB A
A	ANA B	ANA C	ANA D	ANA E	ANA H	ANA L	ANA M	ANA A	XRA B	XRA C	XRA D	XRA E	XRA H	XRA L	XRA M	XRA A
B	ORA B	ORA C	ORA D	ORA E	ORA H	ORA L	ORA M	ORA A	CMP B	CMP C	CMP D	CMP E	CMP H	CMP L	CMP M	CMP A
C	RNZ	POP B	JNZ *	JMP *	CNZ *	PUSH B	ADI #	RST 0	RZ	RET	JZ *	-	CZ *	CALL *	ACI #	RST 1
D	RNC	POP D	JNC *	-	CNC *	PUSH D	SUI #	RST 2	RC	-	JC *	-	CC *	-	SBI #	RST 3
E	RPO	POP H	JPO *	XTHL	CPO *	PUSH H	ANI #	RST 4	RPE	PCHL	JPE *	XCHG	CPE *	-	XRI #	RST 5
F	RP	POP PSW	JP *	-	CP *	PUSH PSW	DRI #	RST 6	RM	SPHL	JM *	-	CM *	-	CPI #	RST 7

Система команд микропроцессора КР580ВМ80А

Таблица 7

КОМАНДА	ДЛИНА В БАЙТАХ	ФУНКЦИЯ	ПРИЗНАКИ				
			Z	S	C	AC	P
MOV	1	Передача содержимого одного регистра в другой.	-	-	-	-	-
MVI	2	Загрузка регистра вторым байтом команды.	-	-	-	-	-
INR	1	Увеличить содержимое регистра на 1.	+	+	-	+	+
DCR	1	Уменьшить содержимое регистра на 1.	+	+	-	+	+
ADD	1	К содержимому аккумулятора прибавить содержимое регистра.	+	+	+	+	+
ADC	1	К содержимому аккумулятора прибавить содержимое регистра с учетом флага переноса.	+	+	+	+	+
SUB	1	Вычитать из содержимого аккумулятора содержимое регистра.	+	+	+	+	+
SBB	1	Вычитать из аккумулятора содержимое регистра и флаг переноса	+	+	+	+	+
ANA	1	Выполнить логическое умножение аккумулятора и регистра.	+	+	0	+	+
XRA	1	Выполнить операцию неравнозначности аккумулятора и регистра.	+	+	0	0	+
ORA	1	Выполнить операцию логического сложения аккумулятора и регистра.	+	+	0	0	+
CMP	1	Сравнить содержимое аккумулятора и регистра.	+	+	+	+	+
ADI	2	К содержимому аккумулятора прибавить второй байт команды.	+	+	+	+	+
ACI	2	К содержимому аккумулятора прибавить второй байт команды с учетом флага переноса.	+	+	+	+	+
SUI	2	Из содержимого аккумулятора вычитать второй байт команды.	+	+	+	+	+
SBI	2	Из содержимого аккумулятора вычитать второй байт команды с учетом флага переноса.	+	+	+	+	+
ANI	2	Выполнить операцию логического умножения аккумулятора и второго байта команды.	+	+	0	0	+
XRI	2	Выполнить операцию неравнозначности аккумулятора и второго байта команды.	+	+	0	0	+
ORI	2	Выполнить логическое сложение аккумулятора и второго байта команды.	+	+	0	0	+
CPI	2	Сравнить содержимое аккумулятора и второго байта команды.	+	+	+	+	+
RLC	1	Сдвинуть циклически влево содержимое аккумулятора.	-	-	+	-	-
RRC	1	Сдвинуть циклически вправо содержимое аккумулятора.	-	-	+	-	-

RAL	1	Сдвинуть циклически влево содержимое аккумулятора через флаг переноса.	-	-	+	-	-
RAR	1	Сдвинуть циклически вправо содержимое аккумулятора через флаг переноса.	-	-	+	-	-
JMP	3	Безусловный переход по указанному адресу.	-	-	-	-	-
JC	3	Условный переход по единичному состоянию флага переноса.	-	-	-	-	-
JNC	3	Условный переход по нулевому состоянию флага переноса.	-	-	-	-	-
JZ	3	Условный переход по нулевому значению результата.	-	-	-	-	-
JNZ	3	Условный переход по ненулевому значению результата.	-	-	-	-	-
JP	3	Условный переход по положительному значению результата.	-	-	-	-	-
JM	3	Условный переход по отрицательному значению результата.	-	-	-	-	-
JPE	3	Условный переход по четности кода результата.	-	-	-	-	-
JPO	3	Условный переход по нечетности кода результата.	-	-	-	-	-
CALL	3	Безусловный переход к подпрограмме.	-	-	-	-	-
CC	3	Переход к подпрограмме по единичному состоянию флага переноса.	-	-	-	-	-
CP	3	Переход к подпрограмме по положительному значению результата.	-	-	-	-	-
CZ	3	Переход к подпрограмме по нулевому значению результата.	-	-	-	-	-
CNZ	3	Переход к подпрограмме по ненулевому значению результата.	-	-	-	-	-
CM	3	Переход к подпрограмме по отрицательному значению результата.	-	-	-	-	-
CPE	3	Переход к подпрограмме по четности кода результата.	-	-	-	-	-
CPO	3	Переход к подпрограмме по нечетности кода результата.	-	-	-	-	-
RET	1	Возврат из подпрограммы.	-	-	-	-	-
RC	1	Возврат из подпрограммы по единичному состоянию флага переноса.	-	-	-	-	-
RNC	1	Возврат из подпрограммы по ненулевому состоянию флага переноса.	-	-	-	-	-
RZ	1	Возврат из подпрограммы по нулевому значению результата.	-	-	-	-	-
RNZ	1	Возврат из подпрограммы по ненулевому значению результата.	-	-	-	-	-
RP	1	Возврат из подпрограммы по положительному значению результата.	-	-	-	-	-

RM	1	Возврат из подпрограммы по отрицательному значению результата.	-	-	-	-	-
RPE	1	Возврат из подпрограммы по четности кода результата.	-	-	-	-	-
RPO	1	Возврат из подпрограммы по нечетности кода результата.	-	-	-	-	-
RST	1	Начальный запуск превышающей программы.	-	-	-	-	-
LXI	3	Загрузка регистровой пары или указателя стека вторым и третьим байтами команды.	-	-	-	-	-
PUSH	1	Запись пары регистров в стек.	-	-	-	-	-
POP	1	Загрузка пары регистров из стека.	-	-	-	-	-
POP PSW	1	Загрузка аккумулятора и регистра признаков из стека.	+	+	+	+	+
STA	3	Запись содержимого аккумулятора по указанному адресу.	-	-	-	-	-
LDA	3	Загрузка аккумулятора содержимым ячейки с указанным адресом.	-	-	-	-	-
XCHG	1	Поменять местами содержимое регистровых пар HL и DE.	-	-	-	-	-
XTHL	1	Поменять местами содержимое верхней ячейки стека и регистровой пары HL.	-	-	-	-	-
SPHL	1	Передать содержимое пары HL в указатель стека.	-	-	-	-	-
PCHL	1	Передать содержимое пары HL в программный счетчик.	-	-	-	-	-
DAD	1	К содержимому пары HL прибавить содержимое любой другой регистровой пары или содержимое указателя стека.	-	-	+	-	-
STAX	1	Записать содержимое аккумулятора по адресу, указанному в регистровой паре.	-	-	-	-	-
LDAX	1	Загрузка аккумулятора содержимым ячейки с адресом, указанным в регистровой паре.	-	-	-	-	-
INX	1	Увеличить на 1 содержимое регистровой пары или указателя стека.	-	-	-	-	-
DCX	1	Уменьшить на 1 содержимое регистровой пары или указателя стека.	-	-	-	-	-
CMA	1	Инвертировать аккумулятор.	-	-	-	-	-
STC	1	Установить 1 в флаге переноса.	-	-	1	-	-
CMC	1	Инвертировать флаг переноса.	-	-	+	-	-
DAA	1	Десятичная коррекция.	+	+	+	+	+
SHLD	3	Записать содержимое пары HL в ячейку с указанным адресом.	-	-	-	-	-
LHLD	3	Загрузить пару HL содержимым ячейки с указанным адресом.	-	-	-	-	-
NOP	1	Пустая операция.	-	-	-	-	-
HLT	1	Останов процессора.	-	-	-	-	-

Примечание. В графе "ПРИЗНАКИ" приняты обозначения:

- "+"- команда изменяет состояние признака;
- "-"- команда не изменяет состояние признака;
- "0"- команда устанавливает бит признака в состояние 0;
- "1"- команда устанавливает бит признака в состояние 1.

Названия регистровых пар в мнемонике команд даются в сокращенном виде с помощью первых букв их (названия: так вместо BC, DE, HL записывается соответственно B, D, H.

Например, команда увеличения на единицу содержимого регистровой пары HL записывается как INX H.

В табл. 6 приняты следующие обозначения:

- & - двухбайтовый операнд;
- * - адрес;
- # - однобайтовый операнд;

Например, команда безусловного перехода (код C3) в таблице записана в виде JMP. Это означает, что вместо символа "*" необходимо указать реальный адрес перехода.

Например, при составлении программы возникла необходимость перейти к началу программы "СИСТЕМНЫЙ МОНИТОР". Команда безусловного перехода в этом случае будет иметь вид JMP F800.

Для записи в регистровую пару HL числа FFFF можно воспользоваться командой LXI H,& (код 21), указав вместо символа необходимый двухбайтовый операнд: LXI H,FFFF.

Аналогично, в случае записи в регистр C числа 10 возможно использование команды MVI C,# (код 0E) в виде MVI C,10.

Из анализа таблицы следует, что команды микропроцессора занимают от одного до трех байтов памяти, причем первый байт любой команды содержит код операции. Он определяет формат команды и те действия, которые должен проделать микропроцессор над данными в процессе ее выполнения. Так команда безусловного возврата из подпрограммы RET - однобайтовая, команда MVI C,# - двухбайтовая, команды LXI H,& и CALL * - трехбайтовые.

Среди всей совокупности команд микропроцессора можно выделить несколько групп однотипных команд:

1) Команды пересылок

Эта группа команд передает данные из одного регистра микропроцессора в другой. При написании команды в мнемокоде сначала указывается регистр-приемник данных, а затем регистр-передатчик. Например, при передаче данных из аккумулятора в регистр B используется команда MOV B,A. В качестве регистра-приемника или регистра-передатчика может быть определена ячейка памяти, адрес которой находится в регистровой паре HL. Такая ячейка памяти является особым регистром M. Например, в

результате выполнения фрагмента программы:

```
LXI H,5000  
MVI M,00
```

в ячейку памяти с адресом 5000 будет записано число 00.

Примеры однобайтовых пересылок с использованием регистра М: MOV A,M , MOV M,C.

Для пересылок данных между аккумулятором и ячейками памяти в качестве адреса ячейки памяти может быть использовано содержимое регистровых пар BC и DE. По командам STAX B, STAX D произойдет запись содержимого аккумулятора в ячейки памяти с адресами, определяющимися содержимым пар BC и DE соответственно. Запись в аккумулятор данных из ячейки производится по команде LDAX B или LDAX D.

Адрес ячейки памяти для записи аккумулятора можно задать и непосредственно. При этом используется команда STA *. Для обратной пересылки - LDA *.

С помощью двухбайтовой команды MVI R,# (R - обозначение одного из регистров) можно записать операнд в любой из регистров микропроцессора.

Команда SPHL переносит содержимое регистровой пары HL в указатель стека SP.

Трехбайтовые команды LXI B,& , LXI D,& , LXI H,& служат для непосредственной записи в соответствующие регистровые пары шестнадцатиразрядного операнда.

Используя команды SHLD *, LHLD * можно организовать пересылки данных между регистровой парой HL и ячейкой памяти, адресуемой по содержимому второго и третьего байтов команды. Например, для записи содержимого регистровой пары HL в ячейки памяти 5000 и 5001 можно использовать команду SHLD 5000. При этом содержимое регистра L будет записано в ячейке 5000, а H - в ячейке с адресом 5001.

С помощью команд PUSH B, PUSH D, PUSH H содержимое соответствующих регистровых пар заносится в стек. По команде PUSH PSW в стек засылаются данные из аккумулятора и регистра признаков. Команды POP B, POP D, POP H производят обратную пересылку.

2) Команды обмена

Данная группа содержит две команды. Одна из них, XCHG, инициирует обмен содержимым регистровых пар HL и DE. Обмен данными между парой HL и ячейкой памяти, адрес которой находится в указателе стека SP, производится по команде XTHL.

3) Команды арифметических операций

Команды этой группы обеспечивают выполнение арифметических операций над содержимым аккумулятора и вторым операндом, источник которого указывается в коде операции. Результат операции помещается в аккумулятор,

уничтожая находившийся ранее там операнд. Например, команда прибавления к содержимому аккумулятора константы 15 будет иметь вид ADI 15. Сложение байта, находящегося в аккумуляторе, с содержимым регистра D производится по команде ADD D.

Сложение аккумулятора с константой и признаком переноса производится по команде ACI, а с содержимым какого-либо регистра и признаком переноса - по команде ADC R (R - наименование регистра). Например, сложение аккумулятора с содержимым регистра E и признаком переноса происходит по команде ADC E.

Таким же образом осуществляется вычитание из аккумулятора константы (команда SUI), константы и признака переноса (команда SBI), содержимого регистра (команда SUB R), содержимого регистра и признака переноса (команда SBB R). При выполнении операции вычитания уменьшаемое должно находиться в аккумуляторе. Результат операции также помещается в аккумулятор.

Сложение двухбайтовых операндов реализуется командами DAD B, DAD D, DAD H, DAD SP, при этом содержимое регистровой пары HL суммируется с содержимым указанной регистровой пары и результат помещается в пару HL,

Очень часто при написании программ используются команды INR R, DCR R, INX RP, DCX RP (R - наименование регистра, RP - наименование регистровой пары) с помощью этих команд производится увеличение и уменьшение на единицу содержимого регистра или регистровой пары. Например, если в регистровой паре DE содержалось число ABCD, то после выполнения команды INX D регистровая пара будет содержать ABCE.

Кроме этого к группе команд арифметических операций относится и DAA - команда двоично-десятичного сложения.

4) Команды логических операций

Операции поразрядного логического умножения аккумулятора с содержимым регистра или вторым операндом происходит по командам ANA R и ANI # соответственно. Результатом операции является восьмиразрядное двоичное число, отдельные разряды которого равны 1 только тогда, когда соответствующие разряды у обоих операндов также равны 1. Например, если в аккумуляторе содержалось шестнадцатеричное число 25 (в двоичной форме 00100101), то после логического умножения с шестнадцатеричным числом AC (в двоичной форме 10101100) в аккумулятор будет занесен результат: шестнадцатеричное число 24 (00100100).

При выполнении поразрядного логического сложения помощью команд ORA R и ORI # образуется двоичное число, отдельные разряды которого равны 1 в том случае, когда соответствующие разряды какого-либо одного из операндов равны 1. После выполнения операции логического сложения шестнадцатеричных чисел 25 и AC, аккумулятор будет содержать шестнадцатеричное число AD (10101101).

Результатом выполнения операции ИСКЛЮЧАЮЩЕЕ ИЛИ командами XRA R или XRI # является байт, отдельные разряды которого равны 1 только в том случае, когда соответствующие разряды операндов имеют противоположные значения. Так, если в аккумуляторе содержится шестнадцатеричное число 25, то после XRI AC, его содержимым будет 89 (10001001).

Команды CMP R и CPI # позволяют сравнить два операнда. Сравнение производится вычитанием из первого операнда хранящегося в аккумуляторе, второго, в качестве которого может выступать содержимое регистра или восьмиразрядная константа. Если в результате операции вычитания окажется, что операнды равны, то признак нуля Z устанавливается в 1, если же значение операнда, хранящегося в аккумуляторе меньше значения второго операнда, то устанавливается в 1 признак переноса C. Необходимо заметить, что команды CMP и CPI не изменяют содержимое аккумулятора. Результаты операций сравнения не запоминаются, а используются лишь для установки признаков.

С помощью команды CMC можно изменить значение бита переноса на противоположное, а команда STC позволяет установить этот признак в 1. Значения всех битов аккумулятора можно изменить по команде CMA.

5) Команды сдвига содержимого аккумулятора

На рис.4 схематически показано, как происходит сдвиг содержимого аккумулятора влево или вправо командами сдвига RAL, RAR и командами циклического сдвига RLC, RRC. В операциях сдвига участвует бит переноса C регистра признаков F. Под воздействием каждой из этих команд происходит сдвиг содержимого аккумулятора на один разряд. Если необходимо сдвинуть содержимое аккумулятора на большее число разрядов, то команду сдвига необходимо повторить требуемое число раз.

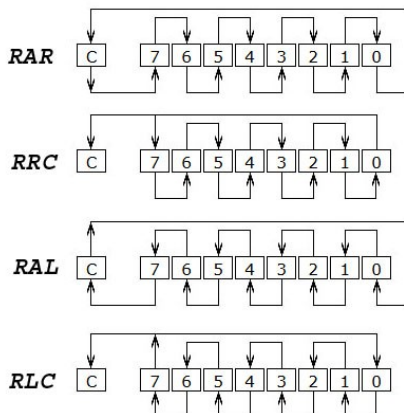


Рис. 4

6) Команды передачи управления

Пока в программе не встречаются команды этой группы, счетчик команд РС постоянно увеличивает свое значение, и микропроцессор выполняет команду за командой в порядке их расположения в памяти.

Порядок выполнения программы может быть изменен, если занести в регистр счетчика команд адрес, отличающийся адреса очередной команды. Такая передача управления может быть выполнена с помощью трехбайтовой команды безусловного перехода JMP *. Как только такая команда встретится о программе, в регистр счетчика команд запишется адрес указанный в команде, и выполнение программы будет продолжено с операции, код которой расположен по этому адресу.

Безусловную передачу управления можно произвести также по команде PCNL, в результате выполнения которой произойдет передача управления по адресу, хранящемуся в регистровой паре HL.

Кроме команды безусловного перехода микропроцессор имеет восемь трехбайтовых команд условного перехода. При встрече команды условного перехода передача управления по указанному в команде адресу происходит только в случае выполнения определенного условия. Если условие не удовлетворяется, то выполняется команда, следующая непосредственно за командой условного перехода. Условия, с которыми оперируют такие команды, определяются состоянием битов регистра признаков F:

NZ - ненулевой результат выполнения предыдущей команды, бит Z установлен в 0;

Z - нулевой результат выполнения предыдущей команды, бит Z установлен в 1;

NC - отсутствие переноса в результате выполнения предыдущей команды, бит C установлен в 0;

C - наличие переноса в результате выполнения предыдущей команды, бит C установлен в 1.

PO - нечетный результат предыдущей операции, бит P установлен в 0;

PE - четный результат предыдущей операции, бит P установлен в 1;

P - неотрицательный результат предыдущей операции, бит S установлен в 0;

M - отрицательный результат предыдущей операции, бит S установлен в 1.

Эти условия проверяются соответствующими командами: JNZ *, JZ *, JNC *, JC *, JPO *, JPE *, JP *, JM *.

Рассмотрим фрагмент программы:

```
MVI A,10
CMP B
JZ 5000
MOV B,A
```

Первой командой в аккумулятор заносится число 10. Затем содержимое аккумулятора сравнивается с содержимым регистра В. После этого, в случае нулевого результата, программа переходит к выполнению операции, расположенной по адресу 5000. При ненулевом результате сравнения содержимое аккумулятора пересылается в регистр В.

При написании программ обычно можно выделить одинаковые последовательности команд, часто встречающиеся в разных частях программы. В целях экономии памяти их объединяют в подпрограммы. В любой части основной программы можно поставить трехбайтовую команду безусловного вызова подпрограммы CALL *, во втором и третьем байтах которой содержится адрес вызываемой подпрограммы. Выполнение команды CALL * начинается с засылки в стек адреса следующей команды. Этот адрес называется адресом возврата из подпрограммы. Он необходим для того, чтобы по окончании выполнения подпрограммы вернуться к продолжению выполнения основной программы. Затем в стек заносится адрес, указанный в команде CALL, и управление передается по этому адресу.

Выполнение подпрограммы всегда заканчивается командой безусловного возврата из подпрограммы RET. При этом адрес возврата из стека переносится в регистр счетчика команд PC и управление вновь передается основной программе.

Кроме трехбайтовой команды безусловного вызова CALL *, в системе команд микропроцессора имеется восемь однобайтовых команд RST0-RST7 вызова подпрограмм, расположенных по фиксированному адресу. Появление в основной программе любой из этих команд вызывает запись в стек адреса возврата из подпрограммы и передачу управления на соответствующую ячейку памяти, где расположена первая команда подпрограммы. В табл.8 дано соответствие между командами RST0-RST7 и шестнадцатеричными адресами ячеек памяти, куда передается управление.

Таблица 8

КОМАНДЫ	АДРЕС НАЧАЛА ПОДПРОГРАММЫ
RST 0	0000
RST 1	0008
RST 2	0010
RST 3	0018
RST 4	0020
RST 5	0028
RST 6	0030
RST 7	0038

Кроме команд безусловного вызова и возврата из подпрограмм, в системе команд микропроцессора имеется восемь команд условного вызова подпрограмм и восемь команд условного возврата, действие которых определяется, так же как и у команд условного перехода, состоянием отдельных битов регистра признаков F. Если условие, необходимое для выполнения команды, отсутствует, то вызов и возврат из подпрограммы не происходит.

7) Специальные команды

Данная группа содержит две команды. Команда NOP не производит никаких действий, но так как на ее выполнение расходуется определенное время, то она часто используется для задания временных интервалов. По команде HLT происходит останов микропроцессора.

7.2.4. Примеры простейших программ

Рассмотрим фрагмент программы:

```
LDA 0C00
CMA
STA 0C01
CALL F89D
```

Данная программа обеспечивает извлечение числа из ячейки памяти с адресом 0C00, его инвертирование и запись в ячейку памяти с адресом 0C01. Далее следует выход в программу "СИСТЕМНЫЙ МОНИТОР".

При составлении программ необходимо все числа и коды команд приводить к шестнадцатеричной системе счисления. При работе с двухбайтовыми величинами нужно также помнить о том, что в системе команд микропроцессора КР580ВМ80А принят следующий формат записи данных в память. Сначала в память заносится младший байт числа, а затем старший. Например, если в регистровой паре HL записано двухбайтовое число FEEA, то после команды SHLD 1000 в ячейке с адресом 1000 будет записан младший байт EA, а в ячейке 1001 - старший FE.

Для записи программы в память ЭВМ необходимо перевести мнемокоды команд в машинные коды при помощи табл.6:

```
LDA 0C00 3A 00 0C
CMA      2F
STA 0C01 32 01 0C
CALL F89D CD 9D F8
```

Перед вводом любой программы рекомендуется очистить память компьютера с помощью директивы "F" "СИСТЕМНОГО МОНИТОРА":

```
->F0000,75FF,00
```


Теперь можно вводить программу в память. Для этого нужно с помощью директивы "М" "СИСТЕМНОГО МОНИТОРА" ввести с клавиатуры машинные коды:

```
->M0000
0000 00 3A
0001 00 00
0002 00 0C
0003 00 2F
0004 00 32
0005 00 01
0006 00 0C
0007 00 CD
0008 00 9D
0009 00 F8
```

Теперь можно выполнить программу. Для этого, также с помощью директивы "М", в ячейку 0C00 нужно занести какое-либо число, например 46:

```
->M0C00
0C00 00 46
```

Запуск программы производится директивой "G" "СИСТЕМНОГО МОНИТОРА":

```
->G0000
```

После вывода на экран стрелки, директивой "М" можно просмотреть результат выполнения программы: в ячейке с адресом 0C01 должно содержаться инверсное значение числа 46.

```
->M0C00
0C00 46
0C01 B9
```

Рассмотрим пример составления программы с использованием подпрограмм "СИСТЕМНОГО МОНИТОРА". Пусть возникла необходимость вывести на экран код нажатой клавиши. Для этого могут быть использованы две подпрограммы "СИСТЕМНОГО МОНИТОРА": подпрограмма ввода в аккумулятор кода нажатой клавиши, расположенная по адресу F803, и подпрограмма вывода на экран шестнадцатеричного кода — F815.

```
CALL F803 CD 03 F8
CALL F815 CD 15 F8
JMP 0000 C3 00 00
```

Третья команда (JMP 0000) "зацикливает" программу, то есть при каждом нажатии на любую клавишу на экран будет

выводиться ее код.

```
->M0000
0000 00 CD
0001 00 03
0002 00 F8
0003 00 CD
0004 00 15
0005 00 F8
0006 00 C3
0007 00
0008 00
```

После запуска программы директивой

```
->G0000
```

можно нажать любую клавишу на клавиатуре. При этом на экране появится ее код. Например, при нажатии клавиши "F4" на экране появится код 03.

7.3. Распределение адресного пространства ПЭВМ

Процессор КР580ВМ80А имеет 16 адресных линий А0-А15 и может непосредственно адресовать 65536 ячеек памяти (64 Кбайт). Каждая ячейка памяти имеет свой порядковый номер, называемый адресом. Адресное пространство процессора в ПЭВМ "Микроша" разделено следующим образом:

- 1) оперативная память, 32 Кбайт, адреса 0000-7FFF;
- 2) постоянная память, 2 Кбайт, адреса F800-FFFF;
- 3) ячейки ввода-вывода, 8 Кбайт, адреса C000-DFFF;
- 4) свободное адресное пространство, 22 Кбайт, адреса 8000 - BFFF и E000 - F7FF.

7.3.1. Оперативная память

Адресное пространство, занятое оперативной памятью 32 Кбайт, предназначено для хранения текущих данных и программ пользователя. Все ячейки памяти с адресами 0000-7FFF являются доступными для пользователя и их содержимое можно изменять программно или с помощью директив "СИСТЕМНОГО МОНИТОРА". Однако, существуют некоторые области ОЗУ, которые не могут быть использованы произвольно. Одна из этих областей - стек.

Стек располагается в интервале адресов 7600-76CF. Более подробно ознакомиться с его организацией можно в разделе 7.2.2. Часть адресного пространства стека занимают рабочие ячейки "СИСТЕМНОГО МОНИТОРА".

Экранная страница занимает интервал 76D0-7FFF и используется в качестве видеоОЗУ. Каждой ячейке в этой области соответствует определенное знакоместо на экране. Содержимое ячейки является кодом КОИ-7 символа,

выведенного на экран в соответствующей позиции. Например, с помощью директивы "М" "СИСТЕМНОГО МОНИТОРА" в ячейку 7A55 записан код 09. При этом на экране в позиции, соответствующей ячейке 7A55, будет выведено изображение человечка. В некоторых ячейках экранной страницы постоянно записан код 00. Это необходимо для улучшения качества изображения на правой и левой границах экрана. При нажатии клавиши "СБРОС" во все ячейки экранной страницы записывается код 00.

7.3.2. Постоянная память

В адресном пространстве процессора самая старшая область адресов F800-FFFF отводится под постоянную память системы, в которой хранится программа СИСТЕМНЫЙ МОНИТОР".

7.3.3. Ячейки ввода-вывода

Организация ввода-вывода в ПЭВМ основывается на способе реализации ввода-вывода как части адресного пространства. При этом обмен информацией с контроллерами и интерфейсами устройств ввода-вывода осуществляется аналогично обмену при работе процессора с ячейками памяти. То есть любое устройство ввода-вывода (входные и выходные регистры, регистры управления) трактуется процессором как ячейки памяти, доступные стандартным операциям чтения и записи. Это делает возможным программировать режимы работы контроллеров ПЭВМ. Адреса контроллеров распределены следующим образом:

C000-C7FF - область, занимаемая адресами контроллера клавиатуры KP580BB55A (D39). Адрес C000 - адрес порта А микросхемы, C001 - порта В, C002 - порта С, C003 - регистра управляющего слова, остальная часть этой области памяти C004 - CFFF) повторяет первые 4 ячейки C000-C003. Через порты А и В микросхемы производится обмен информацией между МП и клавиатурой. Отдельные биты порта С выполняют следующие функции:

- а) бит 0 - вывод на магнитофон;
- б) бит 1 - управление динамической головкой;
- в) бит 2 - управление таймером, также управляющего громкоговорителем;
- г) бит 3 - управление светодиодом "РУС/LAT";
- д) бит 4 - ввод с магнитофона;
- е) биты 5, 6, 7 - опрос клавиш "РУС/LAT", "УС", "НР";

C800-CFFF - область адресов БИС периферийного адаптера KP580BB55A (D12). Адреса C800, C801, C802, C803 соответствуют портам А, В, С и регистру управляющего слова БИС. Остальной интервал циклически повторяет первые четыре ячейки. Через биты 5 и 6 порта В производится вывод информации в последовательном виде через разъем "Интерфейс-1" (X3), бит 7 этого порта

служит для переключения знакогенератора из режима "РУС" в "LAT" и обратно. Через бит 0 порта С осуществляется ввод последовательной информации во внутренние регистры ПЭВМ;

D000-D7FF — адреса, отведенные для БИС дисплейного контроллера КР580ВГ75 (D8). Адрес D000 — адрес регистра команд, D001 — адрес регистра признаков команд;

D800-DFFF — адреса отведенные для БИС таймера КР580ВИ53 (D22). Адрес D800 — адрес счетчика 0, D801 — адрес счетчика 1, D802 — адрес счетчика 2, D803 — адрес регистра управляющего слова. Остальные ячейки интервала циклически повторяют первые четыре;

F800-FFFF — особая область памяти, при операции "Чтение" адресуется к программе "СИСТЕМНЫЙ МОНИТОР", при операции "Запись" к БИС контроллера ПДП КР580ВТ57 (D2). Адрес F808 — адрес регистра управления, F804 — адрес регистра адреса канала номер 2 контроллера ПДП, а F805 — регистра конца счета этого канала.

7.3.4. Свободное адресное пространство

В свободное адресное пространство входят интервалы памяти 8000-BFFF, E000-F7FF. Эти интервалы можно использовать для подключения внешних устройств к разьему "Внутренний интерфейс". В этот же разъем выведены сигналы выбора "32K" — для интервала 8000-BFFF, "CS1" — для интервала E000-E7FF, "CS2" — для интервала E800-EFFF, "CS3" — для интервала F000-F7FF. При необходимости подключить устройство с полем адресов большим, чем свободное адресное пространство, это устройство должно быть реализовано по страничному типу.

7.4. Общие принципы программирования БИС контроллеров

Применение в ПЭВМ "Микроша" программируемых БИС КР580ВИ53, КР580ВВ55А, КР580ВТ57 и КР580ВГ75 дает широкие возможности для расширения функциональных возможностей компьютера и для его использования в качестве центрального устройства в различных системах управления. В данном разделе приводятся сведения о том, каким образом с помощью выбора режимов работы микросхем пользователь может расширить возможности ПЭВМ.

Программирование режимов БИС в общем случае производится путем записи управляющих слов во внутренние регистры микросхем. Как уже отмечалось, доступ к этим регистрам возможен с помощью стандартных команд записи и чтения из памяти (например LDA *, STA * и т.п.). Форматы управляющих слов, а также особенности программирования каждой БИС приведены ниже. Необходимо заметить, что первоначальное программирование БИС осуществляется программой "СИСТЕМНЫЙ МОНИТОР".

7.4.1. Программирование БИС таймера КР580ВИ53

Микросхема КР580ВИ53 представляет собой программируемый трехканальный таймер-счетчик, вырабатывающий временные интервалы, управляемые программой. Таймер состоит из трех независимых идентичных программируемых шестнадцатиразрядных вычитающих счетчиков (обозначенных 0, 1, 2), работающих в двоичном или двоично-десятичном коде. Максимальное значение счета в двоичном коде 2 в степени 16, а в двоично-десятичном 10 в степени 4. Конечное число счета - 0. Работа счетчиков синхронизируется частотой до 2 МГц.

Программирование БИС заключается в записи управляющего слова в регистр управляющего слова (адрес D803) и числа пересчета n в регистр выбранного счетчика (адреса D800, D801, D802). Формат управляющего слова приведен на рис.5. Бит D0 управляющего слова информирует таймер о том, в каком коде будет загружено число пересчета в один из счетчиков. Биты D1-D3 определяют режим работы выбранного счетчика, D4-D5 - характеризует порядок загрузки числа пересчета (загружается только младший байт счетчика, только старший байт, сначала младший, а затем старший байты). Биты D7-D8 - определяют номер выбранного канала.

Для каждого счетчика имеется возможность задания одного из шести режимов работы:

1) режим 0 - выдача сигнала прерывания по конечному числу. При работе по этому режиму на выходе счетчика "OUT" появляется уровень "0" сразу же после установления режима работы. После загрузки числа n в счетчик выход остается в том же состоянии и счетчик начинает считать, если на входе разрешения работы счетчика "GATE" установлен уровень "1". В ПЭВМ Микроша1 входы "GATE0" и "GATE1" соединены с шиной питания +5В. При достижении конечного числа на выходе счетчика устанавливается уровень "1" и остается до тех пор, пока счетчик не будет перезагружен режимом работы или новым числом пересчета;

Канал		Чтение/Запись		Режим работы			Код
D7	D6	D5	D4	D3	D2	D1	D0
00 - канал 0		00 - "зашелкивание"		000 - режим 0			0 - двоичный
01 - канал 1		01 - мл. байт		001 - режим 1			1 - двоично-
10 - канал 2		10 - ст. байт		010 - режим 2			десятичный
11 - запрет		11 - мл., ст. байт		011 - режим 3			
				X00 - режим 4			
				101 - режим 5			
				110 - режим 6			
				111 - режим 7			

X - безразличное состояние

Рис. 5

2) режим 1 - программируемый ждущий мультивибратор. В этом режиме выход счетчика после загрузки числа n устанавливается в уровень "0" после первого тактового сигнала, следующего за передним фронтом на управляющем входе. Одновременно начинается счет, а при достижении конечного числа на выходе устанавливается "1". Таким образом, в этом режиме счетчик представляет собой ждущий мультивибратор с программно-устанавливаемой длительностью сигналов;

3) режим 2 - генератор тактовых сигналов. В этом режиме на выходе счетчика через заданное число периодов тактовой частоты появляется уровень "0" длительностью в один период тактовой частоты. Скважность импульсов определяется числом n ;

4) режим 3 - генератор прямоугольных сигналов. В этом режиме на выходе счетчика будет высокий уровень в течение половины интервала времени, заданного числом n и уровень "0" в течение другой половины;

5) режим 4 - программно-управляемый строб. После установления режима на выходе канала появляется уровень "1". Когда число полностью загружено в счетчик канала и на управляющий вход подан сигнал "1", начинается счет и при достижении конечного числа на выходе появляется импульс уровня "0", длительностью в один период тактовой частоты.

6) режим 5 - схемотехнически управляемый строб, в режиме 5 счетчик является перезапускаемым, т.е. каждое нарастание сигнала на входе "GATE" запускает счетчик или перезапускает его, если счет не окончен.

Рассмотрим один из вариантов программирования таймера. Допустим возникла необходимость получить на выходе "OUT2" счетчика сигнал прямоугольной формы. Для этого необходимо перевести счетчик 2 в режим работы номер 3. Формат управляющего слова следующий: 10110110 (в шестнадцатеричной форме B6). Бит D0 информирует о том, что последующая загрузка числа пересчета будет производиться в двоичной форме. Биты D1, D2, D3 определяют третий режим работы счетчика. Биты D4 и D5 определяют порядок загрузки числа пересчета: сначала в счетчик будет загружен младший байт числа, а затем старший. Битами D6, D7 производится выбор счетчика номер 2. После записи управляющего слова в регистр управления таймера, имеющего адрес D803, необходимо в регистр счетчика 2 (адрес D802) записать число пересчета. Например, шестнадцатеричное 5EA8. Все вышеперечисленные действия выполняет фрагмент программы:

```
LXI H,D803
MVI M,B6
DCX H
MVI M,A8
MVI M,5E
```

Если в процессе счета возникла необходимость в считывании величины текущего счета, то в регистр управления таймера записывается специальное слово, биты D4 и D5 которого содержат 0. Биты D6 и D7 как и ранее служат для выбора счетчика. Биты D0-D3 - произвольные. При получении такого управляющего слова счетчик приостанавливает работу. Затем производят считывание двух байтов из регистра счетчика.

В заключение следует заметить, что счетчик 2 таймера в ПЭВМ "Микроша" используется для управления динамической головкой, что позволяет создавать различные звуковые эффекты в программах пользователя. Счетчики 0 и 1 используются для управления внешними устройствами, подключаемыми к ПЭВМ через разъемы "Внутренний интерфейс" и "Интерфейс-2" соответственно.

7.4.2. Программирование БИС КР580ВВ55А

Микросхема КР580ВВ55А представляет собой программируемое устройство, используемое для ввода-вывода параллельной информации. Схема позволяет осуществлять обмен восьмизначными данными по трем каналам: А, В, С, причем канал С может быть подразделен на два независимых подканала. Направление обмена и режим работы для каждого канала задается программно. Каналы служат для передачи как данных, так и управляющих сигналов. С точки зрения программирования БИС состоит из регистра управления и регистров каналов каждому из которых соответствует определенный адрес. В ПЭВМ "Микроша" применены две микросхемы КР580ВВ55А D12 и D39. Все последующие примеры будут рассматриваться применительно к БИС периферийного адаптера D12. При этом регистр управления имеет адрес С803, адрес канала А - С800, В - С801, С - С802. Программирование контроллера клавиатур производится аналогично.

Режим канала А и 4-7 разряда канала С			Канал А	4-7 разряды канала С	Режим канала В и 0-3 разряда канала С	Канал В	4-7 разряды канала С
D7	D6	D5	D4	D3	D2	D1	D0
=1	00 - режим 0 01 - режим 1 1X - режим 2		1 - ввод 0 - вывод	1 - ввод 0 - вывод	0 - режим 0 1 - режим 1	1 - ввод 0 - вывод	1 - ввод 0 - вывод

Х - безразличное состояние

Рис. 6

Режим работы каналов программируют в управляющем слове. Формат управляющего слова, определяющего режим работ адаптера, приведен на рис.6. Режим работы каналов можно изменять в процессе выполнения программы пользователя, что позволяет обслуживать различные периферийные устройства одной БИС. Микросхеме КР580ВВ55А может работать в одном из трех режимов:

1) режим 0 - простой ввод-вывод. В этом режиме производится обмен информацией по магистрали данных ПЭВМ между МП и устройством ввода-вывода через любой из трех каналов без сигналов управления.

2) режим 1 - стробируемый ввод-вывод. В режиме осуществляется ввод или вывод данных по магистрали данных ПЭВМ через канал А или В по стробирующим сигналам. Линии канала С используются для приема или выдачи сигналов управления.

В режиме ввода данных из внешнего устройства через канал А (В) по линиям канала С передаются следующие управляющие сигналы:

а) инверсный сигнал "Строб приема", формируемый внешним устройством. Этот сигнал инициирует запись данных из внешнего устройства во внутренний регистр канала А (В). Передается по линии РС4 (РС2);

б) сигнал "Подтверждение приема", формируемый БИС.

Устанавливается в "1" и поддерживается на протяжении всего времени от записи данных в регистр канала до их чтения микропроцессором. Передается по линии РС5 (РС1);

в) сигнал "Запрос на прерывание", формируемый БИС.

Информирует микропроцессор о готовности данных к вводу. Передается по линии РС3 (РС0).

При работе канала А (В) в режиме вывода информации по линиям канала С передаются следующие управляющие сигналы:

а) инверсный сигнал "Строб записи", формируемый БИС.

Указывает на готовность данных к передаче во внешнее устройство. Передается по линии РС7 (РС1);

б) инверсный сигнал "Подтверждение записи", формируемый внешним устройством. По этому сигналу производится запись данных во внешнее устройство. Передается по линии РС6 (РС2);

в) сигнал "запрос прерывания", формируемый БИС. Информирует МП о готовности к приему новых данных. Передается по линии РС3 (РС0).

3) режим 2 - двунаправленный обмен информацией по каналу А. При обмене в этом режиме внешнее устройство ввода-вывода должно подключаться к порту канала А через восьмиразрядный трехстабильный буфер приема-передачи данных. Сигналы управления вводом-выводом передаются по линиям канала С. Управляющие сигналы являются комбинацией управляющих сигналов для канала А в режиме 1.

Линии канала С, не использующиеся для передачи управляющих сигналов в режимах 1 и 2, могут быть использованы для ввода-вывода.

Контроль состояния внешнего устройства, подключенного к периферийному адаптеру, в режимах 1 и 2 можно осуществлять чтением слова состояния канала С. Внешнее устройство должно производить обмен информацией с ПЭВМ сигналами, совместимыми по уровню с ТТЛ логикой.

Рассмотрим пример программирования БИС КР580ВВ55А.

Допустим, к ПЭВМ через разъем "Интерфейс-2" подключена восьмисегментная индикаторная матрица. В этом случае выводом световой информации можно управлять программно, путем выдачи матрице определенного параллельного восьмиразрядного кода. Бит, содержащий "1", зажигает сегмент, в то время как бит "0" - гасит. Вывод информации будет производиться по каналу В периферийного адаптера. Каждый сегмент индикатора подключен к одной из линий канала. Для перевода БИС КР580ВВ55А в необходимый режим работы в регистр управления (адрес С803) записывается следующее управляющее слово: 10000000 (в шестнадцатеричном виде 80). В соответствии с рис.6 это управляющее слово переводит микросхему в нулевой режим работы, при этом канал В настроен на вывод информации.

Далее в регистр канала В (адрес С801) последовательно записываются коды, определяющие порядок индикации матричных сегментов. Фрагмент программы, выполняющий указанные действия, выглядит следующим образом:

```
MVI A,80
STA C803
MVI A,01
STA C801
MVI A,02
STA C801
.
.
.
MVI A,FF
STA C801
```

Рассмотрим пример программирования контроллера клавиатуры D39. Как уже отмечалось, с помощью БИС таймера КР580ВИ53 можно управлять работой динамической головки ПЭВМ. Однако для запуска счетчика 2 таймера необходимо подать сигнал высокого уровня на вход разрешения "GATE2" и, кроме этого, разрешить работу логических элементов D21.3 и D21.4. Эти условия выполняются при появлении в разрядах PC1 и PC2 контроллера клавиатуры сигналов высокого уровня. Следовательно, для получения звуковых эффектов необходима программная настройка как таймера, так и контроллера клавиатуры. Следующая программа позволяет

получить тональный сигнал:

```
LXI H,D803
MVI M,B6
DCX H
MVI M,10
MVI M,20
LXI H,C003
MVI M,80
DCX H
MVI M,06
```

Последняя строка определяет перевод разрядов PC1 и PC2 канала C в единичный уровень. Перевод этих разрядов в нулевой уровень останавливает работу счетчика и динамической головки. Частота тонального сигнала определяется числом пересчета, записанным в регистр счетчика 2 таймера.

7.4.3. Программирование БИС KP580BT57

Режим прямого доступа к памяти (ПДП) используется в ПЭВМ для быстрого обмена массивами данных между ОЗУ компьютера и быстродействующим внешним устройством. При работе в режиме ПДП микропроцессор отключается от микроЭВМ и функцию управления магистралями адреса и управления берет на себя контроллер ПДП.

Микросхема KP680BT57 представляет собой четырехканальный программируемый контроллер ПДП. Устройство позволяет осуществлять двунаправленный обмен массивами данных емкостью до 16 Кбайт между ОЗУ и любым из четырех внешних устройств по каналам ПДП (номера 0...3). При одновременном поступлении запросов на ПДП от нескольких устройств программируемая логика схемы позволяет выбирать наивысший по приоритету канал ПДП. Направление обмена данными, а также режим работы устройства задаются с помощью программы начальной установки. БИС содержит восьмизначный регистр управления, а также шестнадцатиразрядные регистры адреса и конца счета каждого из каналов. В ПЭВМ "Микроша" используется лишь один канал ПДП, имеющий второй порядковый номер. По нему осуществляется передача данных из ОЗУ в контроллер дисплея.

В процессе программирования в регистр управления (адрес F808) заносится управляющее слово, определяющее режим работы контроллера ПДП. Формат управляющего слова приведен на рис.7. В регистр адреса (адрес F804) записывается адрес начала той области ОЗУ, в которой располагаются выводимые данные. В младшие четырнадцать разрядов регистра конца счета (адрес F805) записывается число, на единицу меньшее длины передаваемого массива. После передачи очередного байта содержимое регистра

автоматически уменьшается на единицу. При достижении нуля выставляется сигнал "Конец счета". Старшие два разряда регистра конца счета используются для задания направления обмена данными по каналу. Единица в разряде D14 определяет запись данных из ОЗУ в контроллер дисплея.

D0	1	Разрешение работы канала 0
D1	1	Разрешение работы канала 1
D2	1	Разрешение работы канала 2
D3	1	Разрешение работы канала 3
D4	1	Установка циклического сдвига приоритета
D5	1	Установка удлиненной записи
D6	1	Установка КС-СТОП
D7	1	Установка автозагрузки

Рис. 7

Микросхема КР580ВТ57 может работать в следующих режимах:

1) автозагрузка. В этом режиме работает канал 2 с помощью внутренних регистров канала 3. При этом программируется только канал 2. Этот режим позволяет каналу 2 многократно осуществлять обмен массивами данных. По сигналу "Конец счета", начальный адрес массива, количество циклов и направление передачи из регистров канала автоматически загружается в регистры канала 2 с целью продолжения обмена массивом данных;

2) режим остановки по сигналу "Конец счета". В этом режиме запрещается работа канала ПДП после формирования сигнала "Конец счета",

3) режим удлиненной записи. В этом режиме увеличивается время записи данных из ОЗУ во внешнее устройство. Используется в тех случаях, когда быстроедействие ОЗУ и внешнего устройства различно.

4) режим циклического сдвига приоритетов используется в том случае, если в обмене данными по ПДП участвуют 2, 3 или 4 устройства. После каждого цикла обмена приоритеты каналов меняются в кольцевой последовательности так, что обслуживаемому каналу присваивается самый низкий приоритет. При фиксированном приоритете нулевой канал имеет высший приоритет.

В программе "СИСТЕМНЫЙ МОНИТОР" в ячейках F843-F853 и F85B-F85F располагается программа начальной установки контроллера ПДП. Она имеет вид:

```
MVI A,80
STA F808
LXI H,F804
```

```

MVI M,D0
MVI M,76
INX H
MVI M,23
MVI M,49
MVI A,A4
STA F808

```

Первые две строки программы производят запись управляющего слова в регистр управления. Формат слова – 10000000 (80 в шестнадцатеричном виде), что устанавливает режим автозагрузки, однако работа каналов пока не разрешена (нулевые биты в разрядах D0-D3 управляющего слова). Затем в регистр канала 2 записывается адрес начала области памяти, отведенной под массив данных: 76D0. В регистр конца счета канала 2 записана длина выводимого массива: 0923 и направление передачи: единица в разряде D4. Последние две строки программы разрешают работу канала 2 контроллера ПДП.

7.4.4. Программирование БИС KP580BG75

Интегральная микросхема KP580BG75 представляет собой однокристальный контроллер электронно-лучевой трубки. Ее основные функции: вывод информации из памяти ПЭВМ на экран ЭЛТ, промежуточное хранение информации, управление синхронизацией, параметрами развертки и изображения, управление курсором. С помощью программной установки микросхемы можно изменять следующие параметры: число знаков в знакоряде (количество символов в одной строке), число строк раstra в одном знакоряде, число знакорядов в кадре, длительность обратного хода строчной и кадровой разверток, форма курсора и т.д.

БИС видеоконтроллера в ПЭВМ "Микроша" предназначена для сопряжения контроллера ПДП типа KP580BT57 с генератором стандартных знаков – ПЗУ, декодирующим точечную матрицу. Поэтому режим работы контроллера дисплея зависит от режима функционирования контроллера ПДП. Видеоконтроллер через контроллер ПДП получает доступ к экранной области ОЗУ, считывает информацию и ряд за рядом отображает ее на экране. В БИС предусмотрены два буферных запоминающих устройства (ЗУ) на один знакоряд. Пока одно буферное ЗУ используется для отображения, второе заполняется знаками следующего знакоряда. Затем ЗУ меняются местами. Количество знаков в ряде и количество знакорядов на экране задается программно, что позволяет сопрягать KP580BG75 с большинством типов электронно-лучевых индикаторов. Для заполнения ЗУ на один знакоряд запрашивается цикл ПДП. Продолжительность цикла ПДП и интервалы между циклами программируются, что позволяет использовать в ОЗУ экранной страницы динамические ЗУ с различным временем

регенерации.

Для программирования микросхемы используются два регистра: восьмизначный регистр команд (адрес D001) и восьмизначный регистр признаков (адрес D000). В регистр команд записывается сама команда, а в регистр признаков - ее параметры. Число параметров колеблется от 0 до 4 в зависимости от типа команды.

Микросхема КР580ВГ75 имеет набор из восьми команд:

1) Команда "Сброс". Формат команды - 00000000 (в шестнадцатеричном виде 00). Команда имеет четыре параметра. С помощью команды "Сброс" задаются следующие характеристики изображения: количество знаков в знакоряде, количество знакорядов в кадре, количество строк растра в знакоряде, длительность обратного хода кадровой и строчной развертки, форма курсора, чередование отображения знакорядов, а также режимы работы внутреннего счетчика строк БИС. Параметры команды "Сброс" приведены в табл.9.

Таблица 9

КОМАНДА/ПАРАМЕТР	ОПИСАНИЕ	ШИНА ДАННЫХ
Команда	"Сброс"	00000000
Параметр	Компановка кадра	SHNNNNNN
Параметр	---"---	VVRRRRRR
Параметр	---"---	UUUULLLL
Параметр	---"---	MFCCZZZZ

Параметры команды "Сброс" определяют нижеперечисленные характеристики кадра:

а) параметр S - чередование знакорядов. Параметром S микросхема программируется на бланкирование (гашение) чередующихся знакорядов. При единичном значении бита S первый знакоряд отображается на экране, второй бланкируется, третий отображается и т.д.

Таблица 10

S	ФУНКЦИЯ
0	Нормальные знакоряды
1	Чередующиеся знакоряды

б) параметр NNNNNN - количество знаков в знакоряде.

Таблица 11

ННННННН	КОЛИЧЕСТВО ЗНАКОВ В РЯДУ
0000000	1
0000001	2
0000010	3
.....	...
1001111	80
1010000	Не используется
.....	...
1111111	Не используется

в) параметр VV – длительность обратного хода кадровой развертки.

Таблица 12

VV	КОЛИЧЕСТВО ЗНАКОРЯДОВ НА ОБРАТНОМ ХОДЕ КАДРОВОЙ РАЗВЕРТКИ
00	1
01	2
10	3
11	4

г) параметр RRRRRR – количество знаковрядов в кадре.

Таблица 13

RRRRRR	КОЛИЧЕСТВО ЗНАКОРЯДОВ В КАДРЕ
000000	1
000001	2
.....	...
111111	64

д) параметр UUUU – местоположение подчеркивания. Если номер строки подчеркивания больше 7, то верхняя и нижняя строки раstra в знаковряде будут гаситься. Если же номер строки подчеркивания меньше или равен 7, то бланкирование этих строк производиться не будет.

Таблица 14

UUUU	НОМЕР СТРОКИ ПОДЧЕРКИВАНИЯ В ЗНАКОРЯДЕ
0000	1
0001	2
....	...
1111	16

е) параметр LLLL – количество строк раstra в знакоряде.

Таблица 15

LLLL	КОЛИЧЕСТВО СТРОК РАСТРА В ЗНАКОРЯДЕ
0000	1
0001	2
....	...
1111	16

ж) параметр М – режим счетчика строк.

Таблица 16

М	РЕЖИМ СЧЕТЧИКА СТРОК
0	Режим 0
1	Режим 1

В режиме "0" код на выходе внутреннего счетчика строк микросхемы соответствует номеру строки раstra в знакоряде. В режиме "1" счетчик строк сдвинут на единицу относительно номера строки раstra в знакоряде. Режим "0" используется для работы со знакогенераторами, бланкирующими нулевой адрес. Режим "1" используется для знакогенераторов, запускающихся с нулевого адреса.

з) параметр F – режим атрибутов поля.

Таблица 17

F	РЕЖИМ АТТРИБУТОВ ПОЛЯ
0	Непрозрачный
1	Прозрачный

Атрибуты поля используются для изменения визуальных характеристик поля знака: мерцание, подчеркивание и т.д.

и) параметр CC – тип курсора, в ПЭВМ "Микроша" курсор формируется только при значениях 01 и 51 битов CC.

Таблица 18

CC	ТИП КУРСОРА
00	Мерцающий негативный видеоблок
01	Мерцающее подчеркивание
10	Немерцающий негативный видеоблок
11	Немерцающее подчеркивание

к) параметр ZZZZ - длительность обратного хода строчной развертки.

Таблица 19

ZZZZ	КОЛИЧЕСТВО ЗНАКОВ ПРИ ОБРАТНОМ ХОДЕ СТРОЧНОЙ РАЗВЕРТКИ
0000	2
0001	4
0010	5
....	...
1111	16

2) Команда "Начало воспроизведения". По этой команде начинаются запросы ПДП для отображения информации, а также устанавливаются внутренние флаги состояния. Микросхема КР580ВГ75 программируется для получения информации пакетными посылками от одного до восьми знаков в пакете и с интервалами между пакетами от 0 до 55 синхроимпульсов знака. Команда "Начало воспроизведения" не имеет параметров.

Таблица 20

КОМАНДА/ПАРАМЕТР	ОПИСАНИЕ	ШИНА ДАННЫХ
КОМАНДА	НАЧАЛО ВОСПРОИЗВЕДЕНИЯ	001SSSBV

Отдельные биты слова команды определяют режим запросов ПДП:

а) биты SSS - интервал между пакетными запросами ПДП.

Таблица 21

SSS	КОЛИЧЕСТВО СИНХРОИМПУЛЬСОВ ЗНАКА МЕЖДУ ЗАПРОСАМИ ПДП
000	0
001	7
010	16
011	23
...	...
111	55

б) биты BV - количество запросов ПДП в пакете.

Таблица 22

ВВ	КОЛИЧЕСТВО ЗАПРОСОВ ПДП В ПАКЕТЕ
00	1
01	2
10	4
11	8

3) Команда "Прекращение воспроизведения". Формат командного слова: 01000000 (в шестнадцатеричной форме 40). По команде "Прекращение воспроизведения" происходит запрет отображения и установка внутренних флагов состояния БИС. Для возобновления изображения необходимо подать команду "Начало воспроизведения". Параметры отсутствуют.

4) Команде "Чтение регистров светового пера". Эта команда в ПЭВМ "Микроша не используется.

5) Команда "Загрузка курсора", формат командного слова: 10000000 (шестнадцатеричное 80). Обеспечивает помещение последующих байтов в регистры микросхемы, определяющие положение курсора. Команда имеет два параметра. Первый – номер знакоместа в знакоряду, второй – номер знакоряда.

6) Команда "Разрешение прерываний". В ПЭВМ "Микроша" не используется.

7) Команда "Запрет прерываний". В ПЭВМ "Микроша" не используется.

8) Команда "Начальная установка счетчиков". Формат командного слова: 11100000 (шестнадцатеричное E0). По этой команде внутренние счетчики микросхемы устанавливаются в начальное положение, соответствующее отображению в левом верхнем углу экрана.

Рассмотрим пример программирования БИС КР580ВГ75. Ячейки F835–F842, F854–F85A "СИСТЕМНОГО МОНИТОРА" занимает программа начальной установки контроллера дисплея:

```
LXI H,D001
MVI M,00
DCX H
MVI M,4D
MVI M,1D
MVI M,99
MVI M,93
LXI H,D001
MVI M,27
MVI M,E0
```

Первые две строки программы производят запись в регистр командного слова числа 00, что соответствует

выдаче команды "Сброс". Эта команда имеет четыре параметра, которые должны быть записаны в регистр признаков. Занесение параметров (шестнадцатеричные числа 4D, 1D, 99, 93) производится при выполнении 3...7 строк программы.

Первый параметр команды "Сброс" имеет формат 4D. В двоичной форме — 01001101. При этом S = 0, ННННННН = 1001101. Это означает, что отображаться будут все знакоряды по 77 знаков в каждом знакоряде.

Второй параметр — 1D (00011101). VV = 00, RRRRRS = 011101. Длительность обратного хода кадровой развертки равна времени отображения одного знакоряда. Количество знакорядов в кадре равно 29.

Третий параметр — 99 (10011001). В этом слове UUUU — 1001, LLLL = 1001. При этом линия подчеркивания имеет девятый порядковый номер, а знакоряд формируется девятью строками раstra.

Четвертый параметр имеет формат 93 (10010011). Отдельные биты имеют значения: M = 1, F = 0, CC = 01, ZZZZ = 0011.

Такой формат определяет первый режим работы счетчика строк, "непрозрачный" режим отображения атрибутов поля, форму курсора в виде мигающей линии подчеркивания. Время обратного хода строчной развертки выбрано равным времени отображения восьми знаков.

После ввода параметров команды "Сброс" в регистр параметров команд БИС, в регистр командного слова заносится команда "Начало воспроизведения". Ее формат 27 (00100111). В этой команде программируется количество передаваемых знаков в пакете ПДП (BB = 11) и временной интервал между пакетами (SSS = 001).

Программирование БИС завершается подачей команды "Начальная установка счетчиков". Ее формат — E0 (11100000).

Для изменения формата изображения в программе пользователя можно использовать следующую программу:

```
LXI H,D001
MVI M,00
LXI H,D000
MVI M,4D
MVI M,1D
MVI M,99
MVI M,93
MVI A,80
STA F808
LXI H,F804
MVI M,D0
MVI M,76
INX H
MVI M,23
MVI M,49
```

```
LXI H,D001
MVI M,27
MVI M,E0
MVI A,A4
STA F808
```

Например, в шестой строке программы можно изменить параметр с 99 на 77. при этом строки псевдографических символов будут отображаться на экране дисплея без межстрочных интервалов.

Изменив третий параметр команды "Сброс" в строке номер 7 с 93 на В3, можно изменить форму курсора. В этом случае он будет представлен немигающей линией подчеркивания.

Необходимо отметить, что для получения устойчивого изображения на экране при изменении какой-либо одной характеристики изображения, приходится подбирать практически все параметры команды "Сброс".

7.5. Краткое описание языка Бейсик

7.5.1. Общие положения

Язык высокого уровня Бейсик является наиболее простым и доступным для освоения пользователем, практически не имеющим опыта работы с вычислительной техникой. Несмотря на простоту, на нем можно создавать достаточно сложные программы: от игровых до программ управления внешними объектами.

Язык Бейсик в ПЭВМ "Микроша" реализован в виде интерпретатора. Это означает, что в памяти компьютера одновременно располагается как сам интерпретатор, так и программа, написанная на Бейсике. Выполнение программы в этом случае производится следующим образом. Интерпретатор языка считывает первую строку программы, транслирует ее в машинные коды и выполняет действия, соответствующие этим кодам. Затем следует переход ко второй строке и процесс повторяется. Такое построение обуславливает относительно малое быстродействие программ на Бейсике, так как машинные коды строк программы не сохраняются и при выполнении каждой новой строки вновь следует обращение к интерпретатору. Кроме обеспечения выполнения программы пользователя на интерпретатор языка Бейсик возложены и некоторые Дополнительные функции.

Для того, чтобы успешно составлять программы на Бейсике, пользователь должен ясно представлять себе весь процесс разработки программ, начиная с ввода текста с клавиатуры, заканчивая сохранением программы на магнитной ленте и ее выполнением.

Программа интерпретатор Бейсика составлена в машинных кодах и записана на кассете с программным обеспечением, входящей в комплект ПЭВМ. Ее загрузка в память "Микроши"

производится с помощью директивы "I" "СИСТЕМНОГО МОНИТОРА" Интерпретатор занимает в памяти интервал адресов 0000-19FF Таким образом программа пользователя в любом случае (при вводе с клавиатуры или магнитофона) располагается с адреса 1A00.

Для того, чтобы работать с программами на Бейсике необходимо сперва загрузить в память машины интерпретатор (директива "I" "СИСТЕМНОГО МОНИТОРА"), а затем запустить его в работу директивой "G". При этом в левом верхнем углу экрана появляется надпись "*МИКРОША* BASIC" и символ "=>". Это означает, что интерпретатор готов к приему указаний пользователя. Перед вводом какой-либо информации рекомендуется выполнить директиву NEW. Для этого ее набирают на клавиатуре и нажимают "BK".

Программа на Бейсике состоит из последовательности пронумерованных строк. Строкам принято присваивать номера с интервалом 10. В дальнейшем это может оказаться полезным, если потребуется вставить в программу несколько новых строк.

Пример:

наберите на клавиатуре:

```
10 REM
20 REM
40 REM
```

Набор каждой строки завершается нажатием клавиши "BK". проверьте текст набранной программы, для чего наберите на клавиатуре слово LIST и нажмите клавишу "BK". На экране должна появиться набранная программа. Теперь наберите

```
30 REM
```

Повторно проверив текст программы директивой LIST, можно убедиться, что строка с номером 30 расположена между строками с номерами 20 и 40.

Диапазон номеров строк - от 0 до 65535. Каждая строка программы состоит из операторов, содержание которых предписывает интерпретатору те или иные действия. Если операторов несколько, то их в строке отделяют друг от друга символом ":".

Представленная версия интерпретатора Бейсик имеет встроенный экраный редактор. Это позволяет редактировать текст программы, выведенный на экран директивой LIST, а также строку или директиву в процессе набора. Для редактирования, если курсор расположен не в исправляемой строке, клавишами "↑" или "↓" подведите курсор к исправляемой строке. Затем клавишами "→" или "←" переведите курсор в исправляемую позицию. В случае, если был пропущен символ, нажмите клавишу с этим

символом. Недостающий символ появляется на том месте, где был установлен курсор, а часть строки справа от курсора переместится на одну позицию вправо. Если необходимо удалить лишний символ, подведите курсор под этот символ, а затем нажмите клавишу "F2". Символ исчезнет, а часть строки справа от курсора переместится на одну позицию влево. После того, как строка отредактирована, с помощью клавиши переместите курсор до конца строки и нажмите "BK".

Некоторые другие функции клавиатуры:

1) клавиша "F4" останавливает выполнение программы выдачей на экран номера строки, в которой произошел останов;

2) клавиша "СТР" обеспечивает выход в "СИСТЕМНЫЙ МОНИТОР" со сбросом;

3) клавиша "O" - при одновременном нажатии и удерживании совместно с клавишей "УС" выключает вывод на экран информации;

4) клавиша "F1" переключает коды знакогенератора на экране. При однократном нажатии клавиши "F1" латинские буквы превращаются в прописные буквы русского алфавита. Изменяются также и псевдографические символы. При повторном нажатии "F1" происходит обратное переключение. Произвести переключение знакогенератора можно с помощью директив РОКЕ -14335,128 и РОКЕ -14335,0.

Язык Бейсик оперирует двумя видами данных: константами и переменными. Существует два типа констант: числовые и символьные. Числовые константы - это любые десятичные числа в интервале от -1,7E38 до 1.7E38. Буква Е указывает на экспоненциальную форму представления числа. То есть константа 2E3 соответствует числу 2000, а 2E-2 - 0,02. Числовые константы могут быть представлены и в естественном формате. При этом целая часть числа отделяется от дробной символом ".". Например, 12.345 или -0.99. Ноль перед точкой можно не ставить. Точность задания констант - шесть значащих цифр.

Символьные константы представляют собой последовательность любых символов, заключенных в кавычки. Например, "БЕЙСИК", "ПЛЮС" - символьные константы.

В программах на Бейсике используются переменные двух типов: числовые и символьные. Каждой переменной присваивается имя, состоящее из одного или двух символов. Первый из них обязательно должен быть буквой латинского алфавита. Вторым - также буквой латинского алфавита либо цифрой. Символьные переменные после имени содержат знак "\$". Например: М, М4 - допустимые имена числовых переменных; А\$, АМ\$ - символьных.

Группе переменных одного типа может быть присвоено общее имя, и их в этом случае называют переменными с индексами или массивами. Для обращения к какой-либо переменной внутри массива используются один или

несколько индексов. Наименьшее значение индекса равно 0, а наибольшее определяется размером массива. Имена массивов подчиняются тем же правилам, что и имена переменных. Индексы, указывающие номер элемента массива, располагаются в круглых скобках после имени массива.

Например, М\$(3) – третий элемент одномерного массива символьных переменных, имеющего имя М\$; АМ(1,4) – элемент, расположенный на пересечении первой строки и четвертого столбца двумерного числового массива с именем АМ. При работе программы для каждого массива в памяти ПЭВМ резервируется соответствующая область. Перед использованием массив должен быть описан при помощи оператора DIM, о чем будет сказано ниже.

Кроме переменных и констант в выражения языка Бейсик входят знаки операций, скобки и имена функций. Все выражения можно разделить на четыре типа: арифметические, символьные, выражения отношения и логические. В табл.23 приведены знаки операций, принятые в Бейсике: Числовые константы и переменные могут принимать участие в выражениях любого типа. Для символьных переменных и констант разрешены только операции отношения и слияния (+). При вычислении результата выражения все операции имеют определенный приоритет, определяющий порядок вычислений. Чем выше в табл.23 находится знак той или иной операции, тем выше ее приоритет.

Таблица 23

ЗНАК ОПЕРАЦИИ	ОПЕРАЦИЯ
АРИФМЕТИЧЕСКИЕ ОПЕРАЦИИ	
^	ВОЗВЕДЕНИЕ В СТЕПЕНЬ
*	УМНОЖЕНИЕ
/	ДЕЛЕНИЕ
+	СЛОЖЕНИЕ
-	ВЫЧИТАНИЕ
ОПЕРАЦИИ ОТНОШЕНИЯ	
>	БОЛЬШЕ
<	МЕНЬШЕ
=	РАВНО
<>	НЕ РАВНО
>=	БОЛЬШЕ ИЛИ РАВНО
<=	МЕНЬШЕ ИЛИ РАВНО
ЛОГИЧЕСКИЕ ОПЕРАЦИИ	
NOT	ОТРИЦАНИЕ ("НЕ")
AND	ЛОГИЧЕСКОЕ УМНОЖЕНИЕ ("И")
OR	ЛОГИЧЕСКОЕ СЛОЖЕНИЕ ("ИЛИ")

7.5.1.1. Директивы языка Бейсик

Директивы языка Бейсик позволяют оператору производить просмотр и отладку программы, а также вводить и выводить исходные тексты программ с магнитной ленты.

Директива *NEW* подготавливает интерпретатор для ввода новой программы с клавиатуры. Текст программы, набранный ранее, стирается.

Директива *RUN N* служит для запуска программы со строки с номером *N*. Если номер строки отсутствует, то работа программы начинается со строки с наименьшим номером.

Директива *LIST N* выполняет распечатку на экране текста программы, находящегося в ОЗУ, начиная со строки с номером *N*. Если номер отсутствует, то распечатка начинается с первой строки программы.

Директива *CONT* позволяет продолжить выполнение программы с того места, где она была прервана оператором Бейсика *STOP*.

Директива *MSAVE (ИМЯ)* позволяет переписать программу на Бейсике из ОЗУ на магнитную ленту. Имя программы должно состоять из одного символа латинского алфавита и указывается без скобок. Указание имени не является обязательным. Если имя программы при записи не указано, то при последующем вводе имя программы также не указывается.

Директива *MLOAD (ИМЯ)* предназначена для загрузки программ на Бейсике с магнитной ленты в ОЗУ. Если имя программы указано при записи, то происходит загрузка данной программы только при совпадении имен, если имя не указано - в ОЗУ загружается первая встретившаяся программа, написанная на Бейсике.

7.5.2. Операторы языка Бейсик

В Бейсике имеется ряд операторов, которые можно разделить на две группы: выполняемые и невыполняемые (операторы описания).

Выполняемые операторы, в свою очередь, также могут быть разделены на ряд подгрупп: ввод-вывод, управление выполнением программы, организация циклов, построение псевдографических изображений, связь с машинными ресурсами, переходы.

7.5.2.1. Невыполняемые операторы

Оператор *REM* служит для вставки в текст программы на Бейсике комментариев, поясняющих те или иные действия, выполняемые в данной части программы. Этот оператор не влияет на выполнение программы, так как все символы, расположенные в строке за оператором, игнорируются интерпретатором.

Пример:

```
10 REM ИГРОВАЯ ПРОГРАММА "ОТЕЛЛО"
```

Оператор *DIM* предназначен для описания массивов, используемых в программе. При выполнении оператора в ПЭВМ под размещение массива отводится определенная область памяти, размер которой определяется размером массива. Массив можно не описывать, если его размерность не превышает 10. Одним оператором *DIM* можно описать сразу несколько массивов.

Пример:

```
10 DIM A(20), P(15,4)
20 DIM M$(10,10)
```

Оператор *DATA* служит для описания данных непосредственно в программе. Данные любого типа располагаются после оператора и перечисляются через запятую. Программа может содержать любое количество операторов *DATA* в любом месте программы. Все данные, описанные оператором *DATA*, образуют блок данных. Считывание данных из блока производится оператором *READ*. При этом данные будут вводиться последовательно, начиная с первого, описанного оператором *DATA*, заканчивая последним. После каждого обращения к блоку происходит перемещение на одну позицию внутреннего указателя данных интерпретатора.

Пример:

```
10 DATA 12.345, "PLUS", "IBM"
20 DATA -4.34, 0.008
```

Оператор *DEF* дает возможность вводить в текст программы определение новых функций Бейсика (см. раздел 7.5.3). Оператор *DEF* можно использовать только в программе. Определяемая функция должна иметь только один аргумент. Имена вновь определяемых функций должны начинаться с символов *FN*, за которыми могут стоять один или два любых символа. Например, *FNA*, *FNX1* – допустимые имена, а *FNES2*, *FNEKA* – недопустимые. В примере, приведенном ниже, описана функция *FNCT(X)* – тригонометрическая функция котангенс, не реализованная в виде встроенной в интерпретаторе.

Пример:

```
10 DEFFNCT(X)=COS(X)/SIN(X)
20 INPUT X
30 A=FNCT(X)
40 PRINT A
```

С помощью представленной программы можно составить

таблицу котангенсов от различных аргументов.

7.5.2.2 Выполняемые операторы

Оператор *READ* предназначен для чтения данных из блока, описанных оператором *DATA*. Формат оператора в программе следующий: *READ X1,X2,...,XN*, где *XN* - имена числовых или символьных переменных, которым присваиваются значения перечисленные в блоке данных. При считывании данных необходимо следить за соответствием типов переменных и данных.

Пример:

```
10 DATA "МЕТРО", 4, 29
20 READ A$, B, C
```

После выполнения этого фрагмента символьной переменной *A\$* будет присвоено значение "МЕТРО", а числовым переменным *B* и *C* - значения 4 и 29 соответственно. Чтобы убедиться в этом, наберите на клавиатуре:

```
?A$,B,C
```

После нажатия "BK" на экран будут выведены значения указанных переменных.

Оператор *RESTORE* служит для перемещения внутреннего указателя данных в первую позицию, обеспечивая тем самым повторное считывание блока данных.

Пример:

```
10 DATA 1,2,3
20 READ A,B,C
30 PRINT A,B,C
40 RESTORE
50 GOTO 10
```

После запуска программы директивой *RUN* на экране появятся три столбца цифр 1,2,3.

Оператор *INPUT* позволяет вводить данные в программу на Бейсике непосредственно с клавиатуры компьютера. Значения введенных данных присваиваются переменным, имена которых перечислены вслед за оператором. Вводить с клавиатуры можно как числовые, так и символьные константы. При выполнении оператора *INPUT* на экран дисплея выводится символ означающий готовность программы к вводу данных. Если оператором *INPUT* необходимо ввести несколько данных, то после ввода очередного значения ставится запятая и набор данных продолжается. Ввод данных завершается нажатием клавиши "BK".

Пример:

```
10 INPUT M, AM$
```

После высвечивания на экране знака вопроса необходимо ввести сначала числовую, а затем символьную константу.

После оператора INPUT может стоять строка символов, заключенная в кавычки. Имя переменной указывается вслед за строкой символов и отделяется символом ";". В этом случае при выполнении оператора перед знаком вопроса будет выведено соответствующее сообщение. При вводе данных, пока не нажата клавиша "BK", для внесения исправлений можно пользоваться клавишей "_".

Пример:

```
20 INPUT "ВАШЕ ИМЯ"; S$
```

Оператор PRINT предназначен для вывода на экран значений переменных, сообщений и результатов вычислений. Если оператор применен без операнда, то происходит печать строки пробелов. Операндов, расположенных вслед за оператором PRINT, может быть несколько, и тогда их отделяют друг от друга разделителями "," или ";". При этом символ "," определяет, что под каждое выводимое значение отводится 14 позиций в строке, а ";" - столько, сколько необходимо. Числовые константы при печати могут быть представлены в виде целого числа, числа с десятичной точкой или в экспоненциальной форме. В любой форме печатается не более шести цифр.

При вводе программы с клавиатуры компьютера вместо набора слова PRINT можно использовать символ "?". При последующем просмотре текста программы по директиве LIST знаки вопроса будут заменены словом PRINT.

Если после последнего операнда в операторе PRINT стоит один из разделителей (";" или ","), то при выполнении следующего оператора PRINT печать будет продолжена в той же строке. Если разделителя нет, то печать начнется с новой строки.

Оператор CUR X,Y служит для перемещения курсора в позицию с координатой X по горизонтали и координатой Y по вертикали. Диапазон изменения координат курсора по горизонтали 0-63, по вертикали 0-23. Если после оператора CUR X,Y сразу следует оператор печати PRINT, то вывод информации начнется с позиции с координатами X,Y.

Пример:

```
10 CUR 20,15  
20 PRINT "ПРИВЕТ !"
```

Оператор GOTO N служит для безусловной передачи управления на строку с номером N. Это означает, что как только в программе встретится оператор GOTO, естественный ход выполнения программы (строка за строкой) прервется и выполнение программы будет

продолжено со строки с номером, указанным в операторе.

Пример:

```
10 GOTO 150
```

Операторы *GOSUB N* и *RETURN* предназначены для организации подпрограмм. Оператор *GOSUB N* передает управление на первую строку подпрограммы. Подпрограмма должна заканчиваться оператором *RETURN*. После его выполнения происходит возврат в то место основной программы, откуда произошел вход в подпрограмму. Допускается многократная вложенность подпрограмм.

Операторы *ON X GOTO N1,N2,...,NM* и *ON X GOSUB N1,N2,...,NM* в зависимости от результата выражения *X* реализуют условную передачу управления на одну из строк программы, номер которой указан в списке *N1,...,NM*, следующем за оператором. При выполнении оператора сначала вычисляется значение выражения *X*, от результата берется целая часть, которая и указывает на номер строки в списке. Если результат выражения меньше 1 или больше, чем количество номеров в списке, то выполняется оператор, непосредственно следующий за оператором *ON-GOTO* или *ON-GOSUB*.

Пример:

```
10 INPUT A
20 ON A GOTO 40,50,60
30 PRINT "ВВЕДЕНО ЧИСЛО МЕНЬШЕ 1 ИЛИ БОЛЬШЕ 3":GOTO 10
40 PRINT "ВВЕДЕНО ЧИСЛО 1":GOTO 10
50 PRINT "ВВЕДЕНО ЧИСЛО 2":GOTO 10
60 PRINT "ВВЕДЕНО ЧИСЛО 3":GOTO 10
```

Работает программа следующим образом. При выполнении оператора *INPUT* на экран выводится знак вопроса, после чего программа переходит в режим ожидания ввода данных с клавиатуры. В ответ оператор нажимает любую цифровую клавишу, после ввода программа анализирует, какая клавиша была нажата и, в соответствии с результатом анализа, выводит на экран то или иное сообщение. Предположим, пользователь ввел число 2. Оператором *INPUT* переменной *A* присваивается значение 2. В свою очередь оператор *ON-GOTO* передает управление на строку, номер которой в списке указан вторым, т.е. на строку 50. Это вызывает на экране сообщение "ВВЕДЕНО ЧИСЛО 2". Допустим, что пользователь ввел число 7. В этом случае значение переменной *A* больше количества элементов в списке и, следовательно, управление передается на оператор, расположенный непосредственно за *ON-GOTO*. Это оператор *PRINT "ВВЕДЕНО ЧИСЛО МЕНЬШЕ 1 ИЛИ БОЛЬШЕ 3"*.

Оператор *STOP* предназначен для прекращения выполнения программ. Его применение позволяет создавать в программе контрольный останов. Выполнение прерванной программы

можно продолжить с помощью директивы CONT. После выполнения оператора STOP на экран выводится сообщение "СТОП В N", где N - номер строки, содержащей этот оператор.

Оператор FOR X TO Y STEP Z представляет собой оператор начала цикла. Конечным оператором является оператор NEXT. В операторе начала цикла X - выражение, определяющее имя переменной и присвоенное ей первоначальное значение, Y - выражение, определяющее конечное значение переменной цикла, а Z значение, на которое должна изменяться переменная цикла после каждого выполнения оператора NEXT (шаг цикла). Если шаг равен 1, то выражение STEP Z не указывается. Шаг цикла может иметь и отрицательное значение. Порядок выполнения цикла следующий:

- 1) задается имя переменной и ее начальное значение;
- 2) выполняются операторы, находящиеся внутри цикла (операторы тела цикла);
- 3) производится проверка достижения переменной цикла конечного значения;
- 4) изменяется значение переменной цикла на величину шага цикла;
- 5) если конечное значение не достигнуто, то все приведенные операции повторяются;
- 6) если конечное значение достигнуто, то управление передается оператору, следующему за NEXT.

Пример:

```
10 FOR I=1 TO 10 STEP 2
20 PRINT I
30 NEXT
```

Эта программа производит распечатку на экране числового ряда от 1 до 10 с шагом 2.

Условный оператор IF X THEN Y является оператором условной передачи управления. Его работа состоит в следующем:

- а) проверяется истинность выражения X;
- б) если X - истинно, то выполняется оператор или группа операторов (выражение Y), расположенная после слова THEN;
- в) если X - ложно, то управление будет передано следующему оператору.

Выражение X может включать проверку различных условий с использованием как операций отношения, так и логических операций. Операторы, стоящие после THEN, также могут быть различными. В частности, если необходимо выполнить оператор GOTO, то его имя можно не указывать. В этом случае запись IF X THEN GOTO N аналогична IF X THEN N.

Пример:

```
10 IF A+B<0 THEN 100
20 IF A=30 THEN STOP
30 IF A=0 OR B=0 THEN PRINT "КОНЕЦ"
```

Оператор *CLS* предназначен для стирания информации с экрана.

Оператор *PLOT X,Y,Z* позволяет высветить или погасить точку с координатами $0 < X < 127$ по горизонтали и $0 < Y < 49$ по вертикали. Если $Z = 1$, то соответствующая точка высвечивается, если $Z = 0$ - гаснет.

Пример:

```
10 CLS
20 PLOT 64,25,1
```

Оператор *LINE X,Y* позволяет вычерчивать отрезки прямых линий. Операнды X и Y задают координату конечной точки отрезка. Ограничения на значения X и Y такие же, как и в операторе *PLOT*. Для задания координат начала отрезка, а также его вида (засветка или гашение), служит оператор *PLOT*. Начальными координатами отрезка могут выступать значения X и Y , предыдущего оператора *LINE*.

Пример:

```
10 CLS
20 PLOT 0,0,1
30 LINE 40,0
40 LINE 40,40
```

Эта программа высвечивает два отрезка в левой части экрана.

Оператор *POKE X,Y* позволяет записать в ячейку памяти, адрес которой задан выражением X , величину, равную результату выражения Y . Необходимо учесть, что интерпретатор Бейсика оперирует лишь с десятичными числами и, следовательно, для определения ячейки записи ее шестнадцатеричный адрес нужно перевести в десятичную форму (см. описание функции *USR(X)*).

Оператор *CLEAR X* предназначен для очистки памяти от переменных. Этим оператором всем числовым переменным присваивается значение 0, а всем символьным - "пустая строка". Если параметр X указывается, то в памяти выделяется Область памяти размером X байт под размещение символьных переменных. Пример:

```
10 CLEAR 300
```

Этот оператор отводит 300 байт под буфер символьных переменных. По умолчанию размер буфера принимается равному 200 байт.

7.5.3 Функции языка Бейсик

В языке Бейсик имеется ряд встроенных функций, которые позволяют значительно упростить написание программ. Определение "встроенная" означает, что интерпретатор содержит подпрограмму вычисления или обработки данной функции. Существуют функции, работающие с числовыми и символьными данными, функции преобразования типов данных, а также функции Обращения к машинным ресурсам.

Для обращения к встроенной функции необходимо лишь указать ее имя и аргумент. Аргумент всегда заключается в круглые скобки. В качестве аргумента любой функции могут выступать константы, переменные, а также выражения содержащие в свою очередь обращения к встроенным функциям. Пример:

```
10 A=LOG(45)
20 B=SQR(LOG(X)+SIN(0.5))
```

функция $SQR(X)$ вычисляет корень квадратный из выражения X .

функция $EXP(X)$ определяет значение экспоненциальной функции аргумента, заданного выражением X .

функция $LOG(X)$ вычисляет значение натурального логарифма от выражения X .

функция $ABS(X)$ вычисляет абсолютную величину выражения X . При этом значения функции следующие:

X , если $X > 0$

0, если $X = 0$

$-X$, если $X < 0$

функция $SGN(X)$ определяет знак числа. Значения функции $SGN(X)$ следующие:

1, если $X > 0$

0, если $X = 0$

-1, если $X < 0$

функция $SIN(X)$ вычисляет значение синуса выражения X (X - в радианах)

функция $COS(X)$ вычисляет косинус выражения X (X - в радианах).

функция $TAN(X)$ определяет значение тангенса выражения X (X - в радианах).

функция $ATN(X)$ вычисляет значение арктангенса выражения - X . Значение функции представляется в радианах.

функция $INT(X)$ предназначена для выделения целой части выражения X .

функция $RND(1)$ генерирует случайное число в диапазоне от 0 до 1. Аргумент функции всегда равен единице.

функция $LEN(X$)$ позволяет осуществить переход от Символьной переменной к числовой. Результатом ее работы является число, равное количеству символов (длине) символьной переменной $X$$.

Пример:

```
10 X$ = "ЛОГАРИФМ"  
20 A = LEN(X$)
```

В этом случае переменной A будет присвоено значение 8.

функция `LEFT$(X$,Y)` присваивает новой символьной переменной значение, равное первым Y символам переменной X\$.

Пример:

```
10 X$ = "ЛОГАРИФМ"  
20 A$ = LEFT$(X$,3)
```

В этом случае символьной переменной A\$ будет присвоено значение "ЛОГ".

функция `RIGHT$(X$,Y)` присваивает новой символьной переменной значение, равное Y крайним правым символам переменной X\$.

Пример:

```
10 X$ = "ЛОГАРИФМ"  
20 A$ = RIGHT$(X$,4)
```

В этом случае символьной переменной A\$ будет присвоено значение "РИФМ".

функция `MID$(X$,Y,Z)` присваивает новой символьной переменной значение, равное строке из Z символов, расположенных в символьной переменной X\$ с позиции номер Y.

Пример:

```
10 X$ = "ЛОГАРИФМ"  
20 A$ = MID$(X$,3,4)
```

В этом случае переменной A\$ присваивается значение "ГАРИ".

функция `STR$(X)` служит для преобразования числовых величин в символьный вид. Аргумент X - число или арифметическое выражение, а результат работы функции строка, являющаяся символьным представлением этого числа.

Пример:

```
10 A = 258  
20 X$ = STR$(A)
```

Переменной X\$ присваивается значение "258".

функция `VAL(X$)` предназначена для обратного преобразования данных из символьного вида в числовой, начиная с крайнего левого символа переменной X\$. Если в

строке встречается недопустимый символ (не цифра и не знак числа, а, например, буква), то преобразование прекращается. Если уже первый символ является недопустимым, то значение функции равно 0.

Пример:

```
10 X$ = "СТРОКА 3"  
20 Y$ = "3 СТРОКА"  
30 A = VAL (X$)  
40 B = VAL (Y$)
```

В этом случае числовой переменной A будет присвоено значение 0, а B - значение 3.

Функция *ASC(X\$)* переводит код ASCII первого символа переменной X\$ в десятичный вид.

Пример

```
10 X$ = "D"  
20 A = ASC(X$)
```

Переменной A присваивается значение 68.

Функция *CHR\$(X)* присваивает новой символьной переменной значение в виде символа, код ASCII которого в десятичной форме равен X (значение X лежит в пределах от 0 до 255).

Пример:

```
10 X = 68  
20 A$ = CHR$(X)
```

Переменной A\$ присвоено значение "D".

Функция *SPC(X)* позволяет вставлять в печатаемую строку пробелы. Значение аргумента X не должно превышать 255.

Пример:

```
10 PRINT "IBM"; SPC (10); "SYSTEM"
```

Слова IBM и SYSTEM на экране будут разделены десятью пробелами.

Функция *TAB(X)* перемещает курсор в нужную позицию строки. Аргумент X - десятичное число в интервале от 0 до 255. Он указывает, сколько позиций от левого края необходимо отступить для начала печати.

Пример:

```
10 PRINT TAB (5); "A"; TAB (10); "B"
```

Символ "A" будет выведен в шестой позиции строки, символ "B" - в одиннадцатой.

Функция *POS(I)*. Результатом выполнения функции является целое число, равное номеру позиции последнего

символа в текущей строке.

Пример:

```
10 PRINT "АБВГД"
```

В этом случае

POS(1) = 5.

Функция *FRE(X)* предназначена для выдачи пользователю информации об объеме свободной памяти ПЭВМ. Если аргумент символьное выражение, то результат — количество свободных байтов в буфере символьных переменных. Если аргумент число, то результат — количество свободных байтов в памяти.

Пример:

```
10 PRINT FRE(0)
20 PRINT FRE(A$)
```

В первом случае на экран будет выведено сообщение о размере свободной памяти, во втором — о размере свободной области в буфере символьных переменных.

Функция *PEEK(X)* присваивает числовой переменной значение, равное десятичной форме числа, содержащегося в ячейке памяти ПЭВМ с десятичным адресом X.

Пример:

```
10 A=PEEK (0)
```

Функция *USR(X)* предназначена для организации связи программ, написанных на Бейсике с программами в машинных кодах. Такая связь необходима прежде всего для повышения быстродействия программ на Бейсике. Аргумент X — десятичный адрес ячейки памяти, начиная с которой записана программа в машинных кодах. Поэтому при встрече в программе оператор: *USR(X)* интерпретатор передаст управление программе, расположенной с адреса X. Аргумент X задается в десятичной форме в интервале от -32768 до 32767. При X=0 произойдет обращение к ячейке 0000, при X=32767 — к ячейке 7FFF, при X=-32768 к ячейке 8000, при X=-1 — к ячейке FFFF. В конце программы, к которой происходит обращение в операторе *USR(X)*, должна находиться команда RET (машинный код C9) Для передачи параметров и данных в программу на Бейсике или из нее можно воспользоваться операторами *POKE X,Y* и *PEEK (X)*. Десятичные адреса подпрограмм "СИСТЕМНОГО МОНИТОРА", а также портов контроллеров, используемых в ПЭВМ "Микроша" приведены в табл. 24.

7.5.4. Дополнительные сведения об интерпретаторе Бейсика

Интерпретатор имеет два основных режима работы: непосредственный и программный. Непосредственный режим позволяет использовать ПЭВМ в качестве калькулятора.

Например, необходимо определить площадь прямоугольника по известной формуле: $S=a*b$. Если значения a и b заданы (например $a=5$, $b=3$), то нет необходимости составлять программу, а достаточно лишь набрать на клавиатуре

```
"?73*5"
```

и нажать "BK". При этом сразу же высвечивается ответ: "15".

Однако в том случае, если требуется определить площадь нескольких прямоугольников, удобнее воспользоваться программным режимом работы интерпретатора. В этом случае все вычисления будут производиться по специальной программе. Она может иметь следующий вид:

```
10 CLS
20 INPUT "ЗНАЧЕНИЕ A";A
30 INPUT "ЗНАЧЕНИЕ B";B
40 S=A*B:PRINT "ПЛОЩАДЬ = ";S
```

После набора программы на клавиатуре, можно перейти к ее выполнению. Для этого надо ввести директиву RUN. При этом происходит очистка экрана и выводится вопрос "ЗНАЧЕНИЕ A?". В ответ нужно ввести значение длины одной из сторон прямоугольника. Затем следует запрос о длине второй стороны прямоугольника и, после ввода, на экране появляется сообщение "ПЛОЩАДЬ = ...". Вместо точек будет выведено значение площади.

Если набранная программа будет необходима пользователю для дальнейших вычислений, то ее рекомендуется сохранить на магнитной ленте и вводить в память ПЭВМ в любой нужный момент. Вывод программы на ленту производится по директиве MSAVE U интерпретатора. Буквой U обозначено имя программы. Указание имени программы, как уже отмечалось, не является обязательным. Порядок действий пользователя при выводе информации следующий:

- 1) набрать на клавиатуре MSAVE и, если необходимо, указать имя программы;
- 2) вставить компакт-кассету в магнитофон, установить уровень записи несколько больше номинального и включить магнитофон на запись;
- 3) нажать клавишу "BK".

После нажатия "BK" на ленту будет записан сначала тональный сигнал, а затем и программа, сопровождаемая характерным шумовым сигналом. При выводе информации экран дисплея гаснет, а по окончании выгрузки на экране

появляется сообщение "ОШИБКА В ...". Вместо точек выводится произвольный номер (0, 13003, 65367 и т.п.). Это сообщения не является признаком неправильной записи программы на ленту, а лишь информирует пользователя об окончании выгрузки.

Ввод программы на Бейсике в память компьютер производится в следующем порядке:

1) ввести в ПЭВМ интерпретатор Бейсика, если он не был введен ранее;

2) запустить интерпретатор в работу директивой "G"
"BK";

3) если интерпретатор был загружен ранее, то нужно выполнить директиву NEW;

4) набрать на клавиатуре слово MLOAD и имя программы если оно было указано при записи этой программы на ленту;

5) найти на компакт-кассете начало нужной программы;

6) включить магнитофон на воспроизведение и по началу тонального сигнала нажать BK".

При загрузке программы экран гаснет, а по ее окончании на дисплей выводится символ, означающий готовность интерпретатора. Необходимо заметить, что текст предыдущей программы теряется. Следовательно, интерпретатор всегда работает лишь с одной программой на Бейсике.

7.5.5. Сообщения об ошибках

Интерпретатор языка Бейсик позволяет в процессе выполнения программы обнаруживать и анализировать ошибки. Интерпретатор, конечно, не может обнаружить логические ошибки в построении программы, однако выявление ошибок синтаксиса, переполнения, ошибок, связанных с неправильным применением функций и т.п. значительно сокращает время отладки программы.

Если ошибка обнаружена в непосредственном режиме интерпретатора то на экран выводится сообщение

"? XX ОШИБКА", где XX - двухзначный код ошибки. Если ошибка обнаружена в программном режиме, то выполнение программы прекращается и на экран выдается сообщение

"? XX ОШИБКА В N", где XX - двухзначный код ошибки, а N - номер строки, в которой данная ошибка была обнаружена. После сообщения об ошибке на дисплее появляется символ предлагающий оператору просмотреть строку с ошибкой по директиве LIST и внести соответствующие изменения.

Ниже приведен перечень ошибок, анализируемых интерпретатором, и их двухзначные коды:

ошибка 01 - в программе встретился оператор NEXT, для которого не был выполнен соответствующий оператор FOR;

ошибка 02 - неверный синтаксис;

ошибка 03 - в программе встретился оператор RETURN без

предварительного выполнения оператора GOSUB;

ошибка 04 - данных, описанных оператором DATA, меньше, чем переменных оператора READ;

ошибка 05 - значение аргумента не соответствует области определения функции;

ошибка 06 - переполнение при выполнении арифметических операторов;

ошибка 07 - недостаточен объем памяти. Возможные причины: велик текст программы, вложенность подпрограмм и циклов больше нормы, слишком много переменных;

ошибка 08 - отсутствие строки с номером, указанным в операторе перехода;

ошибка 09 - индекс переменной не соответствует размерности массива;

ошибка 10 - выполнение оператора DIM или DEF для описания массивов или функций, описанных ранее;

ошибка 11 - деление на ноль;

ошибка 12 - попытка выполнить операторы INPUT или DEF в непосредственном режиме работы интерпретатора;

ошибка 13 - несоответствие типов данных;

ошибка 14 - переполнение буферной области памяти, отведенной для символьных переменных;

ошибка 15 - слишком большая длина символьной переменной;

ошибка 16 - выражение, содержащее символьные переменные, слишком сложное для интерпретатора;

ошибка 17 - невозможность продолжения выполнения программы по директиве CONT;

ошибка 18 - обращение к функции, не описанной оператором DEF.

Кроме описанных, интерпретатор выдает еще три сообщения об ошибках в случае неправильного ввода данных при выполнении оператора INPUT:

"?ПОВТОРИТЕ ВВОД" - указывает на несоответствие типов вводимых данных и переменных, описанных оператором INPUT;

"?ЛИШНИЕ ДАННЫЕ" - данных набрано больше, чем переменных, в операторе INPUT;

"??" - данных набрано меньше, чем переменных в операторе INPUT.

Соответствие шестнадцатеричных и десятичных адресов

Таблица 24

ШЕШНАДЦАТЕ- РИЧНЫЙ АДРЕС	ПОДПРОГРАММА ИЛИ ПОРТ ВВОДА-ВЫВОДА	ДЕСЯТИЧНЫЙ АДРЕС
F800	Старт "СИСТЕМНОГО МОНИТОРА"	-2048
F803	Ввод символа с клавиатуры	-2045
F806	Ввод байта с магнитофона	-2042
F809	Вывод символа на экран	-2039
F80C	Вывод байта на магнитофон	-2036
F812	Опрос состояния клавиатуры	-2030
F815	Вывод байта на экран в шестнадцатеричном виде	-2027
F818	Вывод слова на экран	-2024
F89D	Выход в "системный монитор" без сброса	-1891
FEEA	Ввод кода нажатой клавиши	-278
C000	Порт а контроллера клавиатуры	-16384
C001	Порт б контроллера клавиатуры	-16383
C002	Порт с контроллера клавиатуры	-16382
C003	Регистр управляющего слова контроллера клавиатуры	-16381
C800	Порт А периферийного адаптера	-14336
C801	Порт В периферийного адаптера	-14335
C802	Порт С периферийного адаптера	-14334
C803	Регистр управляющего слова периферийного адаптера	-14333
D000	Регистр команд контроллера дисплея	-12288
D001	Регистр признаков команд контроллера дисплея	-12267
D800	Счетчик 0 таймера	-10240
D801	Счетчик 1 таймера	-10239
D802	Счетчик 2 таймера	-10238
D803	Регистр управляющего слова таймера	-10237
F804	Регистр адреса канала 2 контроллера ПДП	-2044
F805	Регистр конца счета канала 2 контроллера ПДП	-2043
F808	Регистр управляющего слова контроллера ПДП	-2040

7.6 Загрузчик

Программа составлена в кодах микропроцессора, загрузка ее производится по директиве "I" "СИСТЕМНОГО МОНИТОРА". Адрес запуска программы - 7400.

Программа "Загрузчик" предназначена для ввода программ с магнитной ленты, записанных в формате "СИСТЕМНОГО МОНИТОРА" ПЭВМ "Микроша" с произвольной или неизвестной скоростью. С помощью "Загрузчика" в память ПЭВМ могут вводиться лишь программы, составленные в кодах микропроцессора КР580ВМ80А. Применение "Загрузчика" позволяет исключить ошибки считывания программ в тех случаях, когда скорость воспроизведения Вашего магнитофона отлична от номинальной.

Порядок работы с "Загрузчиком" следующий. После запуска программы командой "G7400" и "BK" производится очистка экрана и в его центральной части появляется сообщение "***ВВОД**" и запрос "СМЕЩЕНИЕ:". В ответ на него нужно набрать на клавиатуре четырехзначное значение смещения (аналогично директиве "I" "СИСТЕМНОГО монитора") и нажать "BK". В том случае, если программа будет загружаться в ОЗУ без смещения, необходимо просто нажать "BK": в этом случае смещение полагается равным нулю.

После нажатия "BK" программа выдает сообщение "ПО ГУДКУ НАЖМИТЕ "BK"". После этого Вам следует установить в магнитофон кассету с необходимой программой, перемотать ленту до начала этой программы, включить магнитофон на воспроизведение и по началу тонального сигнала нажать клавишу "BK".

После выполнения указанных операций экран погаснет и программа начнет вычислять частоту импульсов серии нулевых байтов, составляющих тональный сигнал. Затем в соответствии с этой частотой "Загрузчик" устанавливает необходимые константы задержки для подпрограммы ввода "СИСТЕМНОГО МОНИТОРА" и передает управление этой подпрограмме. По окончании ввода управление передается "СИСТЕМНОМУ МОНИТОРУ" и на экране появляется сообщение "В "О"- ...". Вместо точек будет выведено число, которое можно использовать в качестве третьего параметра директивы "О" "СИСТЕМНОГО МОНИТОРА", для того, чтобы произвести запись программы с той же скоростью, с которой была записана считанная программа. Кроме этого выводится начальный адрес расположения программы, конечный адрес и контрольная сумма. При ошибке считывания выводится ошибочная контрольная сумма и символы "?!".

7.7 Редактор и Ассемблер

Программа составлена в кодах процессора и загружается в память ПЭВМ по директиве I" "СИСТЕМНОГО МОНИТОРА". Стартовый адрес - 0000.

Предлагаемый пакет программ состоит из Редактора и Ассемблера.

Редактор предназначен для набора и редактирования любого текста непосредственно на экране дисплея: исправления ошибок, удаления и ввода символов и целых фрагментов. Все исходные тексты с помощью Редактора могут быть сохранены на магнитной ленте.

Ассемблер - программа-транслятор для разработки программ на языке ассемблера микропроцессора KP580BM80A. Текст программы под управлением Редактора с клавиатуры или магнитофона заносят в область текстового буфера ОЗУ, а затем с помощью Ассемблера транслируют. Результат трансляции - машинные коды разрабатываемой программы - располагаются в другой области ОЗУ, называемой областью трансляции.

Работа с пакетом программ начинается с запуска Редактора директивой "G" "СИСТЕМНОГО МОНИТОРА", в результате чего на экране появляется надпись "ED*МИКРОН*" (при первом запуске выводится сообщение об ошибке, однако Редактор самостоятельно восстанавливает нормальный режим работы).

Директива Редактора "AP2"+"N" очищает экран и выводит вопрос "NEW?". При ответе "Y" происходит очистка текстового буфера и Редактор переходит в режим набора строки (в начале первой строки появится псевдографический символ "стрелка вправо").

Режим набора используют для ввода текста с клавиатуры компьютера, причем строка может содержать не более 64 символов. За 8 позиций до конца строки генерируется звуковой сигнал, предупреждающий о том, что для продолжения набора необходимо перейти на новую строку. Набор строки завершается нажатием клавиши "BK", в результате чего она заносится в текстовый буфер. Допущенную ошибку легко исправить, сместив курсор клавишей до нужного места и набрав нужный символ. Далее можно возвратиться к любому месту набираемой строки. Для перехода к редактированию нужно нажать клавишу "СТР".

Режим редактирования позволяет оперативно просматривать введенный текст построчно, либо фрагментами по 24 строки. Очередной фрагмент выводится последовательным нажатием клавиш "AP2" и "↓" или "AP2" и "↑". Вернуться к началу или концу текста можно с помощью директив "AP2"+"B" или "AP2"+"E" соответственно. Для редактирования внутри отображаемого фрагмента нужно пользоваться клавишами перемещения курсора. Переход из режима редактирования в режим ввода строки происходит при попытке смещения курсора за пределы нижней границы

текста клавишей "**↓**". Для полного стирания ранее набранного текста необходимо с помощью директивы "**AP2**"+"**E**" или "**AP2**" + "**↓**" перевести курсор в последнюю строку, а затем набрать "**AP2**"+"**N**" и "**Y**".

В Редакторе предусмотрена возможность автоматического поиска группы символов. Для этого после задания директивы "**AP2**"+"**I**" вводят искомую группу и нажимают "**BK**" - на экране появится фрагмент текста, в первой строке которого содержатся нужные символы. Воспользовавшись директивой "**AP2**"+"**R**", можно найти последующие фрагменты с искомой группой символов.

Отдельные символы удаляют из строки установкой курсора под соответствующим знакоместом и нажатием на клавишу "**F2**", а освобождают место для пропущенного символа клавишей "**F4**".

В режиме редактирования в текст можно вставлять отдельные псевдографические символы, нажав предварительно клавиши "**AP2**"+"****".

Чтобы вставить в текст одну или несколько строк, к началу следующей за ними строки подводят курсор и выполняют директиву "**AP2**"+"**A**". Если надо вставить строки перед первой строкой текста, то необходимо перевести курсор в левый верхний угол экрана с помощью директивы "**AP2**"+"**↑**", а затем выполнить "**AP2**"+"**A**". В результате весь текст, следующий за помеченной строкой, удалается с экрана и Редактор переходит в режим ввода строк. Выйти из этого режима можно нажав клавишу "**STR**".

Для удаления фрагмента текста курсор перемещают в начало его первой строки и выполняют директиву "**AP2**"+"**D**". Затем перемещают курсор до строки, перед которой заканчивается удаляемый фрагмент, и вновь нажимают "**AP2**"+"**D**". Если данный фрагмент решено оставить, то нажимают клавишу "**STR**".

Имеющийся в памяти компьютера текст можно записать на магнитную ленту по директиве "**AP2**"+"**O**". В ответ на нее Редактор запрашивает имя текста, под которым он будет сохранен на ленте. Указав имя (что не является обязательным) и, включив магнитофон на запись, нажимают "**BK**".

Для приема текста с магнитной ленты вводят директиву "**AP2**"+"**I**", а затем, в ответ на запрос Редактора - имя вводимого текста. После этого включают магнитофон на воспроизведение и нажимают "**BK**". Если имя не указано, то произойдет загрузка первого встреченного Редактором текста. По окончании ввода на экране появится начальный фрагмент введенного текста.

Редактор может самостоятельно сравнить записанный на ленту текст с текстом в памяти компьютера. Для этого надо нажать клавиши "**AP2**" + "**V**" и ввести текст с магнитофона. Если тексты не идентичны, на экран выводится сообщение "**ОШИБКА**", а если совпадают - их начальный фрагмент. Редактор позволяет компоновать текст

из нескольких фрагментов, вводимых в этом случае директивой "AP2"+"M". Любую директиву работы с магнитофоном можно отменить нажатием клавиши "СТР".

Введенную программу можно сразу же оттранслировать передав управление Ассемблеру директивой "СТР", в ответ на которую на экране появится надпись "ASSM*МИКРОН*". При необходимости из Редактора можно выйти в "СИСТЕМНЫЙ МОНИТОР" нажав "УС"+"Е".

Необходимо учесть, что программы на языке ассемблера, предназначенные для трансляции, должны иметь определенный формат записи. В состав строки программы входят поля метки, оператора, операнда и комментария. В поле метки содержатся символические имена данных или команд, этим символическим именам при трансляции программы в машинные коды присваиваются значения, равные адресу ячейки, в которой располагаются отмеченные данные или команды. Поле метки всегда отделяется от поля оператора символом ":". Для удобства чтения программ между полями допускается вводить пробелы клавишей "ГТ". В поле оператора содержится команда микропроцессора (см. 7.2) или один из псевдооператоров "ORG", "END", "EQU", "DB", "DW", "DS". Псевдооператор "ORG" помещает значение выражения, используемого в качестве операнда, в счетчик адреса. Псевдооператор "END" информирует Ассемблер о достижении физического конца команды. Этот псевдооператор содержит последняя строка программы. Псевдооператор "EQU" присваивает метке значение, указанное в Поле операнда. Псевдооператор "DB" служит для размещения в последовательных ячейках памяти элементов списка байтовой длины, указанных в поле операнда. Псевдооператор "DW" аналогичен "DB", но оперирует с элементами списка двухбайтовой длины. Псевдооператор "DS" резервирует в памяти ячейки памяти в количестве, определяемом значением выражения, указанного в поле операнда.

Пример:

```
ORG      0000H
M1:      EQU      0F89DH
M2:      EQU      0F818H
          LXI      H,BUF
          CALL     M2
          CALL     M1
BUF:      DB      1FH,'МИКРОША',00H
          END
```

Используемые в программе метки должны состоять из последовательностей латинских или русских букв или цифр и обязательно начинаться с буквы или символа "@" или "?". Длина метки не более 6 символов. В качестве метки нельзя использовать

имена регистров процессора: A, B, C, D, E, H, L, SP, PSW. Имена меток с оператором присвоения "EQU" должны быть определены перед использованием.

Псевдооператор "ORG" должен быть описан только один раз.

В качестве операндов можно использовать имена регистров микропроцессора, имена меток, константы, выражения десятичные и шестнадцатеричные числа. После шестнадцатеричного числа ставится буква H, а если он начинается с буквы, то перед ним ставится 0.

Операндами могут быть выражения с арифметическими знакам "+" и "-". В них можно использовать константы, метки, и также текущий адрес трансляции, обозначаемый "\$".

Трансляция программы в машинные коды начинается после нажатия одной из клавиш:

"1" - программа транслируется с одновременным выводом на экран протокола трансляции, представляющего собой строки исходного текста программы, перед которыми выводятся шестнадцатеричные коды ошибок, адреса размещения команд и данных в ОЗУ компьютера и машинные коды транслируемой программы;

"2" - после трансляции программы на дисплее отображается перечень встретившихся в ней имен меток в алфавитном порядке и их шестнадцатеричные адреса;

"3" - исходный текст программы транслируется в машинные коды, и выводится результат трансляции: количество ошибок и два шестнадцатеричных числа.

Первое из них - адрес конца оттранслированной программы в той области, где она должна работать. Второе, ограниченное символами "/", - в области трансляции. Это связано с тем, что в результате трансляции машинные коды программы всегда располагаются в области оттранслированных программ независимо от адресов, в которых она должна работать. Поэтому, если начальный адрес транслируемой программы отличен от адреса начала области трансляции (1100H-18FFH), то перед отладкой или запуском ее необходимо переместить в рабочую область, определенную псевдооператором "ORG" в начале программы. Для перемещения можно воспользоваться директивой "T" "СИСТЕМОГО МОНИТОРА".

Трансляцию программы можно прервать нажатием клавиш "УС" и "С". Выйти из Ассемблера в "СИСТЕМНЫЙ МОНИТОР" можно, нажав клавиши "УС"+"Е" или любую другую клавишу с кодом, меньшим 31H. При нажатии "СТР" управление передается Редактору.

Если при трансляции объем ОЗУ компьютера окажется недостаточным, она прекратится и на экране появится сообщение "МАЛО ОЗУ".

Во время трансляции Ассемблер анализирует синтаксис исходного текста и, при обнаружении ошибок, выводит

следующее сообщение на экран:

- 01 - двойное определение метки;
- 02 - метка не была определена ранее;
- 04 - использована несуществующая мнемоника команды;
- 08 - неправильно определен операнд;
- 10 - в имени метки применен недопустимый символ.

Если в одной строке обнаружено несколько ошибок, то на экран выводится результирующий код, равный сумме кодов отдельных ошибок.

7.8 Дизассемблер

Программа составлена в кодах микропроцессора. Загрузка программы в память ПЭВМ производится по директиве "I" "СИСТЕМНОГО МОНИТОРА".

Дизассемблер позволяет с минимальными затратами времени переводить тексты программ с языка машинных кодов на язык Ассемблера.

Программа Дизассемблер используется вместе с Редактором, описание которого приводится в разделе 7.7 настоящего руководства. Дизассемблер занимает область памяти 0800H-0FFFH, Редактор — 0000H-07FFFH. Переход из Редактора в Дизассемблер и наоборот осуществляется нажатием клавиши "СТР". В "СИСТЕМНЫЙ МОНИТОР" можно выйти только из редактора, нажав клавиши "УС" и "Е" (в дальнейшем "УС"+"Е"). Прежде чем приступить к дизассемблированию программы, ее коды необходимо поместить в область ОЗУ с адреса 1100H, для чего можно использовать директиву "Т" "СИСТЕМНОГО МОНИТОРА". При входе в Дизассемблер (из "СИСТЕМНОГО МОНИТОРА" по директиве "G800", а из Редактора нажатием клавиши "СТР") на экране появляется сообщение:

DIS.*МИКРОН*
КОНЕЦ ТЕКСТА ...

Вместо точек будет выведено шестнадцатеричное число, соответствующее адресу конца текста в буфере. Эта информация необходима для определения свободного места в области ОЗУ, отводимой под текст и определяемой как разность адреса настройки указателя стека и адреса конца текста. Затем появляется символ "*" приглашающий оператора ввести директиву Дизассемблера. Их всего четыре:

- D - дизассемблирование;
- M - расстановка меток в дизассемблированном тексте;
- A - удаление из текста адресов;
- СТР-выход в Редактор.

После задания директивы "D" на экран выводится запрос "РАБОЧИЙ АДРЕС", на который нужно ответить вводом полного шестнадцатеричного адреса (со всеми незначащими нулями) той области ОЗУ, в которой работает

дизассемблируемая программа. Если при вводе адреса будет нажата клавиша, не соответствующая шестнадцатеричной цифре, то запрос рабочего адреса будет повторен, а если клавиша "СТР", то Дизассемблер перейдет в режим ожидания ввода директив.

Далее на экране дисплея появится запрос "ТЕКСТ (Y/N) ?". Если в дизассемблируемой программе имеются текстовые фрагменты, то необходимо нажать клавишу "Y". На последующие запросы "НАЧАЛО:" и "КОНЕЦ:" надо ответить вводом адреса начала и конца области данных (возможно определение до 32 областей данных).

После определения всех областей нужно на очередной запрос "НАЧАЛО:" ответить нажатием клавиши "BK". На экране появится вопрос "NEW (Y/N) ?". Ответ "Y" приведет к очистке буфера, и текст будет создаваться сначала. Любой другой ответ сохранит имеющийся в буфере текст, а вновь создаваемый пристыкует к нему.

После всех этих процедур, остается на запрос "АДРЕС:" ввести полный адрес начала дизассемблируемого фрагмента. Адрес вводят абсолютный, т.е. соответствующий той области памяти, в которой работает дизассемблируемая программа. Как только адрес будет введен, на экране появятся 24 строки дизассемблированного фрагмента, причем формат строки будет таким: сначала выводятся четыре символа адреса кода операции, затем четыре пробела (на их место в дальнейшем могут быть поставлены метки), следующие четыре позиции отводятся под мнемонику команды и одна позиция на разделительный пробел, за которым располагаются операнды (если в данной команде они имеются). О таком позиционировании следует помнить при редактировании текста. В случае его нарушения директивы "А" и "М" Дизассемблера могут выполняться не полностью.

Вывод фрагмента всегда заканчивается вопросом "ТЕКСТ (Y/N) ?". Нажатие клавиши "Y" приведет к записи этого фрагмента в буфер текста и дизассемблированию следующей части программы. При нажатии клавиши "N" или любой другой (кроме "СТР") фрагмент в буфер записан не будет, но дизассемблирование следующего фрагмента произойдет. Если нажать клавишу "СТР", то дизассемблированный фрагмент в буфер текста записан не будет и последовательное дизассемблирование будет прекращено. На экране появится запрос "АДРЕС?". Можно ввести новое значение адреса, с которого следует вести дизассемблирование, или нажать еще один раз клавишу "СТР" для завершения работы директивы "D".

После ввода директивы "М" Дизассемблер выдает запрос "СИМВОЛ:", требующий ввода символа (буквы латинского алфавита), который будет использован в качестве первого в именах всех расставляемых в тексте меток, два других символа - двухразрядное шестнадцатеричное число от 00H до FFH. Таким образом, максимальное число меток, которое

может быть расставлено Дизассемблером, - 256. Если при выполнении директивы "М" количество меток, превысит эту величину, то будет выведено сообщение "МНОГО МЕТОК" и работа директивы - "М" прекратится. В этом случае можно разбить программу на части и вновь провести дизассемблирование.

После окончания работы директивы "М" будут выведены начальное сообщение и адрес конца текста в буфере, а Дизассемблер перейдет в режим ввода директив.

Теперь можно ввести директиву А, результат работы которой, удаление всех адресов, расположенных в первых четырех позициях строки. После проведенных операций будет создан текст программы, который может быть отредактирован и вновь оттранслирован с помощью Ассемблера.

При запуске Редактора по директиве, "G" "СИСТЕМНОГО МОНИТОРА на экран выводится запрос "NEW?". Если ответить "Y", то будет очищен текстовый буфер и Редактор перейдет в режим ввода текста. При нажатии любой другой клавиши Редактор произведет анализ содержимого буфера на на ли признака конца текста и, если не обнаружит его, выведет сообщение "МАЛО ОЗУ" и запрос "NEW?", на который опять следует ответить "Y". Если признак конца текста будет найден, то на экран выводится его начальный фрагмент (при наличии в буфере текста произвольной информации заканчивающейся признаком конца текста, на экран будет выведена бессмысленная информация: ее не следует пытаться редактировать, т.к. это приведет к нарушению работы Редактора. Выйти из этой ситуации поможет директива "AP2"+"N" и нажатие клавиши "Y").

Используемая версия Редактора имеет некоторые отличия от описанной в разделе 7.7. Для удаления фрагмента текста необходимо пометить его начало нажатием клавиш "AP2"+"D", переместить курсор на строку, следующую за удаляемым фрагментом, после чего ввести директиву "AP2"+"E".

"AP2"+"S" — занесение в промежуточный буфер помеченного фрагмента текста. Начало запоминаемого фрагмента помечают директивой "AP2"+"S". Курсор перемещают на строку, следующую за последней строкой запоминаемого фрагмента, и вводят директиву "AP2"+"E". Фрагмент текста будет запомнен в буфере, в качестве которого используется область трансляции. Пользуясь директивой "AP2"+"S", Вы уничтожаете дизассемблируемую программу. При попытке занесения в буфер слишком большого фрагмента будет выведено сообщение "МАЛО ОЗУ".

"AP2"—"T" — вставка запомненного фрагмента в редактируемый текст перед строкой, на которой расположен курсор.

"AP2"+"L"X=Y — поиск и замена последовательности символов X на последовательность Y.

При необходимости разбиения строки текста на две,

курсор устанавливают на символ, с которого должна начинаться новая строка и нажимают "BK". Для слияния двух строк курсор подводят к первой из объединяемых строк и нажимают "PC".

В режиме ввода текста можно использовать возможность дублирования предыдущей строки или ее части нажатием клавиши в результате которого будут последовательно повторены символы предыдущей строки.

7.9. Отладчик

Программа составлена в кодах микропроцессора и загружается в память ПЭВМ по директиве "I" "СИСТЕМНОГО МОНИТОРА". Стартовый адрес - 0100H.

Символический отладчик (SID) обеспечивает отладку программ, включая в себя выполнение отлаживаемых программ в реальном времени и пошаговом режиме, дизассемблирование, ассемблирование, отображение содержимого памяти и достаточное ряд функций.

При работе SID отлаживаемая программа должна располагаться в области памяти 1EFF-5A00

7.9.1. Команды SID

Команда отладчика представляет собой предложение длиной до 64 символов. Первая буква определяет тип команды, остальные знаки являются параметрами. В некоторых командах применяются особые знаки или "W", что указывает на специальный режим выполнения команды. Особенности специального режима будут определены для каждого типа команд. В операнде команды может содержаться до трех параметров, разделяемых запятыми или пробелами. Ввод команды заканчивается нажатием клавиши "BK".

До нажатия "BK" команда может быть отредактирована с помощью клавиш "стрелка влево" и "стрелка вправо". При нажатии последней происходит стирание всей строки команды. Список команд отладчика приведен ниже:

- A - запись в память с использованием мнемкода;
- C - вызов подпрограмм отладки, вводимых пользователем;
- D - вывод содержимого области памяти в шестнадцатеричном виде и кодах ASCII;
- F - заполнение памяти константой;
- G - запуск программ на выполнение;
- H - вспомогательная команда;
- L - вывод содержимого области памяти в мнемкоде ассемблера;
- M - перемещение блоков памяти;
- P - установка и сброс точек прохода;
- S - изменение содержимого памяти;
- T - трассировка выполнения программы;
- U - выполнение программы без трассировки;

X - проверка и изменение состояния флажков и регистров процессора.

Подробное описание команд приведено ниже.

7.9.2. Параметры команд

В качестве параметров в командах SID могут быть следующие выражения:

- 1) шестнадцатеричные числа;
- 2) десятичные числа;
- 3) содержимое стека любой глубины;
- 4) арифметические выражения.

Независимо от того в каком виде представлен параметр, его максимальное значение в десятичном виде должно находиться в пределах 0-65535 (0-FFFF - в шестнадцатеричном).

Шестнадцатеричные числа идентифицируются по последним четырем символам.

Пример:

ВВОДИМОЕ	ШЕШНАДЦАТЕРИЧНОЕ
ЗНАЧЕНИЕ	ЗНАЧЕНИЕ
1	0001
100	0100
FAA0	FAA0
10000	0000
34567	4567

Десятичные параметры вводятся с предшествующим символом "#". Десятичные значения дополняются или отбрасываются согласно правилу, приведенному для шестнадцатеричных значений. Одновременно происходит перевод десятичного числа в соответствующее шестнадцатеричное.

Пример:

ВВОДИМОЕ	ШЕШНАДЦАТЕРИЧНОЕ
ЗНАЧЕНИЕ	ЗНАЧЕНИЕ
#8	0008
#255	00FF
#65545	0009

Преобразование десятичных параметров осуществляется по модулю 65536, т.е. при вводе больших значений получается параметр, равный разности между введенным значением и 65536.

Для удобства оператора в SID допускается возможность ввода одного или нескольких алфавитно-цифровых символов в коде ASCII (см. табл. 5), заключенных в апострофы. В этом случае каждый символ переводится в две шестнадцатеричные цифры согласно коду ASCII.

Подобно параметрам, приведенным выше, ряды знаков

одиночной длины дополняются слева нулями, в то время как ряды большей чем два знака длины воспринимаются по двум крайним правым знакам. Апострофы не включаются в ряд знаков при счете символов не учитываются.

Пример:

ВВОДИМОЕ	ШЕСНАДЦАТЕРИЧНОЕ
ЗНАЧЕНИЕ	ЗНАЧЕНИЕ
'A'	0041
'AB'	4142
'ABC'	4243
' A'	2041

Числа, а также ряды символов в коде ASCII могут быть скомбинированы в выражения с помощью арифметических знаков "+" и "-". Запись параметров в этом случае не допускает пробелов.

Знак "+" между двумя операндами означает сложение их значений. Если "+" стоит перед одиночным параметром, тогда а качестве исходного для суммирования берется значение, вычисленное на предыдущем шаге.

Пример:

DFF00+#128,+5

Первый параметр команды представляет собой выражение "FF00+0128". Его шестнадцатеричное значение - FF80. Второй параметр "+5" начинается со знака "+" который означает, что этот параметр будет сложен с предыдущим значением (FF80). Таким образом указанная выше команда аналогична DFF80,FF85.

Знак "-" заставляет SID вычитать значение последующего выражения из накопленного.

Пример;

Команда DFF00-200,-#512 эквивалентна DFD00,FE00.

С помощью специального оператора осуществляется пересылка содержимого стека в программный счетчик микропроцессора. Количество операторов указывает глубину стека, с которой выбирается параметр. Особенностью оператора является то, что при выборке данных из стека его содержимое не меняется.

7.9.3. Команда A

При помощи команды A производится ввод в память операндов в кодах микропроцессора KP580BM80A с использованием мнемокода (см. табл. 7).

Форматы команды A:

- 1) AFFFF
- 2) A
- 3) -A

Формат (1) начинает построчную трансляцию с адреса, условно обозначенного FFFF. Формат (2) аналогичен (1), за исключением того, что начальный адрес для трансляции принимается с последнего адреса трансляции.

SID выводит на дисплей адрес команды и ожидает ввода мнемокода, принимает его, преобразует в двоичную форму и заносит в память, после чего на экран выводится адрес следующей команды. Возврат в режим команд SID производится при нажатии "BK" или вводе в память символа ".".

Недействительная мнемоника или неверный формат полей вызывают сообщение об ошибке "?" и SID приглашает к повторному вводу мнемокода.

Выполнение команды формата (3) приводит к тому, что из памяти ПЭВМ удаляется модуль ассемблер/дисассемблер. Это увеличивает на 1,5 Кбайт область памяти, используемую для отладки программ, но команды A и L при этом становятся недействительными и команды T и U отображают только шестнадцатеричные коды.

Пример:

A5000

5000 MOV A,B

5001 INH A

7.9.4. Команда C

По команде C происходит вызов подпрограммы с указанного адреса без нарушения состояния регистров отлаживаемой программы.

Форматы команды C:

- 1) CFFFF
- 2) CFFFF,BBBB
- 3) CFFFF,BBBB,DDDD

Формат (1) выполняет вызов подпрограммы по адресу обозначенному FFFF. Во время вызова подпрограммы командой C этого формата содержимое регистровых пар BC=0000 и DE=0000. Выход из подпрограммы производится по команде RET, которая возвращает управление на командный уровень SID.

Формат (2) эквивалентен (1), но содержимое регистровой пары BC устанавливается по содержимому выражения BBBB, в

то время как DE-0000.

В формате (3) регистровая пара BC устанавливается по значению BBBB, а DE - по DDDD.

7.9.5. Команда D

Команда D используется для отображения отдельных областей памяти как в байтах, так и в формате слов. На экран выводится шестнадцатеричные значения и коды ASCII.

Форматы команды D:

- | | |
|---------------|----------------|
| 1) DFFFF | 5) DWFFFF |
| 2) DFFFF,SSSS | 6) DWFFFF,SSSS |
| 3) D | 7) DW |
| 4) D,SSSS | 8) DW,SSSS |

Формат команды (1) отображает память в байтном формате, начиная с адреса, заданного как FFFF. На экран выводятся 12 строк.

Формат (2) определяет конечный адрес отображаемого интервала памяти. При выполнении команды производится вывод на экран содержимого интервала памяти FFFF-SSSS.

Формат (3) отображает 12 строк, при этом адрес запуска берется из последнего отображенного адреса.

Форматы (5) - (8) подобны (1) - (4) за исключением того, что содержимое интервала памяти выводится в формате слов.

Выполнение команды может быть прекращено нажатием любой клавиши. Если перед командой D вводится знак то содержимое памяти в кодах ASCII не выводится.

7.9.6. Команда F

Команда F производит заполнение памяти константой.

Формат команды:

FFFFF,SSSS,NN

В формате приняты обозначения: FFFF - начальный адрес заполнения, SSSS - конечный адрес, NN - значение константы.

7.9.7. Команда G

Команда G используется для запуска отлаживаемых программ. Выполнение происходит в реальном времени с адреса заданного в команде G. По этой команде управление процессором передается от SID программе пользователя и не может быть возвращено до тех пор, пока не достигнет точки прерывания или точки прохода (см. команду P).

Форматы команды G:

- | | |
|--------------------|----------------------|
| 1) G | 7) -G |
| 2) GPPPP | 8) -GPPPP |
| 3) G,FFFF | 9) -G.FFFF |
| 4) GPPPP,FFFF | 10) -GPPPP,FFFF |
| 5) G,FFFF,SSSS | 11) -G,FFFF,SSSS |
| 6) GPPPP,FFFF,SSSS | 12) -GPPPP,FFFF,SSSS |

формат (1) запускает выполнение программы пользователя с адреса, содержащегося в программном счетчике, значение которого можно изменять по команде X. В этом случае точки прерываний не установлены.

формат (2) разрешает выполнение программы пользователя с адреса, заданного параметром команды.

формат (3) аналогичен (2), но устанавливает точку прерывания программы по адресу, обозначенному FFFF. В этом случае отлаживаемая программа получает управление и выполняется до тех пор, пока не встретится адрес прерывания. По достижении этой точки на экран выводится соответствующий адрес. Временная точка прерывания автоматически удаляется при переходе SID в командный режим. Инструкция, расположенная по адресу прерывания, не выполняется.

формат (4) объединяет функции (2) и (3).

формат (5) расширяет возможности команды, определяя две точки временного прерывания по адресам, обозначенным FFFF, SSSS. Выполнение программы начинается с текущего значения программного счетчика и продолжается до тех пор, пока не встретится один из адресов прерывания.

формат (6) аналогичен (5), но предполагает начальную становку программного счетчика.

форматы команды G (7)-(12) соответствуют (1)-(6), за исключением того, что информация в точках прохода не отображается до тех пор, пока соответствующий счетчик см. команду P) не достигнет 1.

7.9.8. Команда H

С помощью команды H можно осуществлять преобразование шестнадцатеричных чисел в десятичные и обратно, производить сложение и вычитание двух параметров.

форматы команды H:

- 1) HAAAA,BBBB
- 2) HAAAA
- 3) HAA

Формат (1) вычисляет сумму и разность двух операндов и отображает результат в виде:

SSSS RRRR

Приняты обозначения: SSSS - сумма, RRRR - разность.

Формат (2) используется для перевода чисел из шестнадцатеричного вида в десятичный и обратно. При выводе на экран результатов выполнения команды первое число - шестнадцатеричное, второе - десятичное.

Формат (3) используется для представления двухзначного числа во всех присущих ему видах. Результат представляется в виде:

NNNN #DDDD 'S'

где NNNN - шестнадцатеричная форма записи числа, DDDD - десятичная, S - соответствующий символ в коде ASCII. В форме S отображаются только алфавитно-цифровые символы с шестнадцатеричными кодами от 20 до 7F.

7.9.9. Команда L

Команда L выводит содержимое памяти с использованием мнемокодов ассемблера.

Форматы команды L:

- 1) LFFFF
- 2) LFFFF,SSSS
- 3) L
- 4) L,SSSS

Формат (1) выводит список дизассемблированных машинных кодов, начиная с адреса, обозначенного FFFF.

Формат (2) задает точные пределы дизассемблирования: FFFF-начальный адрес, SSSS-конечный.

Формат (3) начинает дизассемблирование с адреса, последнего в списке, ассемблированного (см. команду A),
трассированного (см. команды T и U), с адреса прерывания

(см. команды G и P).

Формат (4) выводит дизассемблированные коды с текущего адреса по адрес SSSS.

Если код операции не имеет действительной мнемоники на языке ассемблера KP580BM80A, то на экран выводится сообщение: ??=NN, где NN - шестнадцатеричное значение неверного кода операции.

7.9.10. Команда M

Команда M позволяет оператору перемещать блоки данных

из одной области памяти в другую.

Формат команды M:MFFFF,SSSS,RRRB

Приняты следующие обозначения: FFFF - начальный адрес пересылаемого блока, SSSS - конечный адрес, RRRR - начальный адрес области памяти, служащей приемником данных.

7.9.11. Команда P

С помощью команды P производится установка, сброс и отображение точек прохода. Действие этих точек похоже на действие временных точек прерывания с тем отличием, что точка прохода не удаляется во время выполнения программы и инструкция, соответствующая адресу точки прохода, выполняется. После отображения точки прохода оператор может продолжить выполнение отлаживаемой программы командой G до тех пор, пока не встретится новая точка прохода или временная точка прерывания.

Предусмотрена возможность задавать для каждой точки прохода определенное количество проходов, после которого наступает останов программы.

Каждый раз, когда при выполнении программы встречается точка прохода, на экран выдается сообщение:

NN PASS FFFF

где NN - число оставшихся проходов через данную точку, FFFF - адрес точки прохода. Кроме того, выводится содержимое всех регистров микропроцессора (см. команду X).

При достижении точки прохода число проходов через нее уменьшается на 1. Процесс выполнения программы продолжается до тех пор, пока число проходов не достигнет единицы. При этом точка прохода становится точкой прерывания и выполнение программы прекращается. Всего в программе может быть определено не более 8 точек прохода.

Форматы команды P:

- 1) PFFFF
- 2) PFFFF,NN
- 3) P
- 4) -PFFFF
- 5) -P

Формат (1) устанавливает точку прохода в адресе, обозначенном FFFF, с числом проходов равным 1, заставляя адрес FFFF становиться точкой прерывания.

Формат (2) устанавливает точку прохода с числом проходов NN. Число проходов не должно превышать FFH.

Формат (3) отображает все установленные точки прохода с указанием количества проходов по каждой точке.

Формат (4) сбрасывает точку прохода по указанному

адресу, в то время как (5) сбрасывает все точки.

7.9.12. Команда S

При помощи команды S производится проверка и изменение содержимого памяти.

Форматы команды S:

- 1) SFFFF
- 2) SWFFFF
- 3) S
- 4) SW

Формат (1) позволяет вводить в память ПЭВМ с адреса, обозначенного FFFF данные байтовой длины, в то время как (2) - двухбайтовой. В любом случае SID выводит на дисплей последовательные адреса, начиная с адреса, обозначенного FFFF, вместе с данными, хранящимися в ячейке с указанным адресом. После этого SID ожидает ввода данных с клавиатуры. Если нажимается клавиша "BK", то данные не изменяются и печатается следующая строка. Для изменения содержимого памяти в адресе оператор набирает на клавиатуре нужное значение (с последующим нажатием "BK"), которое и становится новым элементом данных. Выход из режима производится по вводу символа и нажатию "BK".

Формат (1) позволяет также заносить в память коды алфавитно-цифровых символов. Для этого они заключаются в кавычки.

Форматы (3) и (4) подобны (1) и (2) соответственно. В качестве адреса ячейки для вывода и изменения кодов берется адрес, находящийся в программном счетчике.

7.9.13. Команда T

Команда T позволяет проследить по шагам выполнение отлаживаемой программы с возможностью наблюдения за изменением состояния регистров.

Форматы команды T:

- 1) TN
- 2) T
- 3) TWN
- 4) TW

Формат (1) проследит выполнение программы с текущего содержимого программного счетчика, значение которого можно изменять по команде X. Параметр N определяет количество шагов трассировки. Формат (2) - частный случай формата (1) с N = 1.

При выполнении каждого шага трассировки на экран выводится состояние регистров в выполняемой программе перед выполнением соответствующей операции. На последнем

шаге трассировки выводится адрес прерывания и управление передается SID.

Форматы (3) и (4) позволяют облегчить отладку основного командного цикла. При использовании этих форматов осуществляется трассировка основной программы, а выполнение подпрограмм осуществляется в реальном времени.

В команде T учитываются точки прохода, установленное командой P. При этом трассировка прекращается, если во время выполнения очередного шага счетчик проходов установился в 1.

7.9.14. Команда U

Команда U сходна с командой T, описанной выше, за исключением того, что состояние регистров процессора при выполнении UN и UWN не отображается на каждом шаге, а на экран выводится лишь конечный результат. Прогон отлаживаемой программы может быть прерван в любой момент.

Форматы команды U:

- 1) UN
- 2) U
- 3) UWN
- 4) UW
- 5) -U

Форматы (2) и (4) выполняют функции, аналогичные форматам (1) и (2) директивы T, но без отображения состояния регистров на каждом шаге.

Форматы (3) и (4) полностью аналогичны соответствующим форматам команды T.

Формат (5) делает невозможным отображение точек проходов (см. команду P) до тех пор, пока число проходов не достигнет 1.

Подобно команде T выполнение программы может быть прервано нажатием любой клавиши на клавиатуре. В этом случае отображается адрес прерывания и управление переходит к SID.

7.9.15. Команда X

Команда X позволяет оператору проверять и изменять состояние регистров центрального процессора.

Форматы команды X:

- 1) X
- 2) XF
- 3) XR

Формат (1) отображает состояние процессора в данный

момент отладки программы в формате: CZMEI A=A B=BC D=DE H=HL S=SP P=PC OP,

где C, Z, M, E, I представляют собой состояния флагов переноса, нуля, минуса, четности и внутреннего переноса соответственно. Если позиция содержит "-", то соответствующий флаг сброшен, если букву - установлен.

Далее отображено содержимое аккумулятора A, регистровых пар BC, DE, HL, указателя стека SP и программного счетчика PC. Поле, помеченное OP, содержит мнемонику инструкции, соответствующую содержимому программного счетчика. В том случае, когда удален модуль ассемблер/дисассемблер, в этом поле печатается шестнадцатеричный код операции.

Формат (2) позволяет оператору изменять состояние флажка. Через F обозначен один из флагов: C, Z, M, E, I. Оператор может оставить флажок в его настоящем значении, нажав "BK", либо установить его вводом "1", либо сбросить вводом "0".

Формат (3) разрешает производить изменение содержимого регистров процессора. Буквой R обозначены имена регистра A, регистровых пар B, D, H, S (SP), P (PC). При работе этого формата отображается текущее содержимое регистров, после чего отладчик ожидает ответа оператора. Если оператор нажимает "BK", значение данных не изменяется. В противном случае набор символов, введенных с клавиатуры становится новым содержанием регистров. При использовании формата XA вводимые данные должны иметь размерность байта, в остальных случаях - два байта.

7.10 Редактор текстов

Программа загружается с помощью директивы "X" "СИСТЕМНОГО МОНИТОРА". Запуск производится нажатием клавиш "G" и "BK".

Предлагаемый Редактор предназначен для создания новых текстов, редактирования уже имеющихся, загрузки текстовых файлов с магнитофона и записи их на магнитную ленту.

Редактор имеет следующие основные режимы работы:

- 1) режим набора новой информации (режим набора строк);
- 2) режим редактирования исходного текста;
- 3) режим обмена с носителем информации (магнитофоном);

После загрузки Редактора и нажатия клавиш "G" и "BK" в левом верхнем углу экрана появляется надпись "ED.*МИКРОН*", символ и курсор в виде мигающей линии подчеркивания. Это свидетельствует о готовности Редактора к дальнейшей работе. Если в результате выполнения директив "G" и "BK" на экран было выведено сообщение "ОШИБКА. МАЛО ОЗУ", Вы не должны огорчаться, так как Редактор сам восстановит нормальный режим работы. После того как Редактор сообщит о своей готовности, Вам необходимо ввести команду, определяющую

в каком из режимов Вы будете работать.

7.10.1 Режим набора строк

Вход в режим набора строк осуществляется последовательным нажатием клавиш "AP2" и "H" (далее в описании поочередное или одновременное нажатие каких-либо клавиш будет, именоваться командой и обозначаться арифметической суммой, "например, "AP2"+"H" или "УС"+"Е". Обозначение клавиш соответствует русскому алфавиту). При этом происходит очистка экрана и выводится вопрос "NEW?". Если Вы отвечаете "Y" (сокращенное YES), то очищается текстовый буфер компьютера и Редактор переходит в режим набора строк.

В начале первой строки экрана появится мигающий курсор в виде псевдографического символа "->". Если Вы попытаетесь набирать какой-либо текст, то увидите, что ввод информации осуществляется в буквах латинского алфавита. Для перехода в русский алфавит необходимо два раза нажать клавишу "F1", при этом изменяется форма курсора: теперь это псевдографический символ "↓". Нажав на какую-нибудь алфавитно-цифровую клавишу, Вы получите изображение русской строчной буквы; нажав эту же клавишу совместно с клавишей "HP", Вы получите изображение русской заглавной буквы.

Этот режим работы дает возможность ввода текста с клавиатуры компьютера, причем строка может состоять не более чем из 63 символов. За 8 позиций до конца строки генерируется звуковой сигнал, предупреждающий о том, что для продолжения ввода необходимо перейти на новую строку. Такой же сигнал будет звучать и в тех случаях, когда Вы попытаетесь совершить действия, "неверные" с точки зрения ПЭВМ. Набор строки завершается нажатием клавиши "BK". При этом в память заносится текст, расположенный слева от курсора. После нажатия "BK" в конце строки остается метка в виде псевдографического символа "↓", однако в распечатке текста ее не окажется. Если Вы при наборе строки допустили ошибку, то ее легко исправить, сместив курсор назад до места ошибки клавишей и набрав нужный символ. Затем клавишей можно перевести курсор к тому месту, с которого Вы продолжите набор текста. Следует учесть, что указанная методика исправления ошибок в строке возможна лишь до тех пор, пока не нажата клавиша "BK", то есть до ввода строки в буфер компьютера.

После набора исходного текста можно переходить к его редактированию. Вход в режим редактирования из режима ввода строки осуществляется нажатием клавиши "STR".

7.10.2 Режим редактирования текста

Программа "Редактор текстов" обладает широкими возможностями для редактирования. С ее помощью можно вводить новые строки в уже имеющийся текст, удалять ненужные фрагменты, исправлять ошибки, оперативно просматривать исходный текст и т.д. Вход в режим редактирования возможен двумя путями:

- 1) из режима строки нажатием клавиши "СТР";
- 2) из режима обмена с магнитофоном. В этом случае после ввода текста с магнитофона Редактор автоматически переходит в режим редактирования.

Визуально определить то, что текстовый Редактор находится в режиме редактирования можно по изменению формы курсора и метки в конце строки. Они принимают вид засвеченного знакоместа и соответственно.

Режим редактирования позволяет оперативно просматривать текст построчно или фрагментами по 24 строки. Очередной фрагмент выводится на экран командой "AP2"+"↓". Для удобства восприятия текста в начале каждого нового фрагмента отображаются две строки предыдущего. Команда "AP2"+"↑" служит для просмотра текста в обратном порядке. К началу или концу текста Вы можете вернуться при помощи команд "AP2"+"Б" "AP2"+"Е" соответственно. Для перемещения курсора к началу первой строки фрагмента Вы можете использовать клавишу "↶".

Для перемещения курсора внутри фрагмента используйте клавиши "←", "→", "↓", "↑". попытка сместить курсор за пределы нижней границы текста приводит к выходу редактора в режим ввода строк, о чем свидетельствует изменение формы курсора. Чтобы найти какой - либо фрагмент текста, совсем не обязательно просматривать его от начала до конца; на этот случай предусмотрена команда "AP2"+"Д" — команда автоматического поиска группы символов. После набора команды Вам необходимо ввести искомую группу символов и нажать "ВК". После этого на экране отобразится фрагмент, в первой строке которого содержатся искомые символы. Последующие фрагменты с необходимой Вам группой символов определяются по команде "AP2"+"Р".

Отдельные символы удаляются установкой курсора в позицию соответствующего знакоместа и нажатием клавиши "F2". Освобождение места для пропущенного символа производится командой "F4".

В режиме редактирования в текст можно вставлять символы латинского алфавита. Для этого необходимо подвести курсор в соответствующую позицию и последовательно выполнить команды "AP2"+"Э" и "УС"+"Е". Затем ввести нужный символ и выполнить команды "AP2"+"Э" и "УС"+"Р".

Чтобы вставить в текст одну или несколько строк, к

началу следующей за ними строки подводят курсор и выполняют команду "AP2"+"A". Если эти строки Вам желательно вставить перед первой строкой текста, то вначале нужно нажать клавишу "\", а затем "AP2"+"A". В результате весь текст, следующий за помеченной строкой удаляется с экрана (но не из памяти компьютера) и программа переходит в режим ввода строк. Набрав нужные строки, Вы можете вновь перейти в режим редактирования с помощью клавиши "СТР".

Операция удаления не устраивающей Вас части текста производится по следующей методике. К началу первой строки удаляемого фрагмента подводится курсор и выполняется команда "AP2"+"D". При этом вместо первого символа строки появится псевдографический символ "↓". Затем клавишей "↓" перемещают курсор до строки, перед которой заканчивается удаляемый фрагмент и снова выполняется команда "AP2"+"D". Отмена команды удаления части текста производится нажатием клавиши "СТР".

После того как Вы набрали и отредактировали текст, его можно сохранить на магнитной ленте или сразу же распечатать на печатающем устройстве.

7.10.3 Режим организации обмена с магнитофоном

Имеющийся в памяти ПЭВМ текст выводится на магнитофон только из режима редактирования по команде "AP2"+"O". После выполнения команды произойдет очистка экрана и "Микроша" запросит у Вас имя файла, под которым он будет сохранен на магнитной ленте. Присвоение имени не является обязательным. После ввода имени Вам нужно включить магнитофон на запись и нажать клавишу "BK". Выгрузка информации сопровождается характерным звуковым сигналом.

Загрузка текста в буфер ПЭВМ производится по команде "AP2"+"I". После набора команды "Микроша" выводит на экран запрос имени вводимого текста. Если Вы не указываете имя, то происходит загрузка первого встреченного машиной файла. После ввода имени Вам нужно включить магнитофон на воспроизведение и по началу двухтонального сигнала нажать клавишу "BK". По окончании загрузки на экране отобразится первый фрагмент введенного текста и программа перейдет в режим редактирования.

Редактор может самостоятельно сравнить записанный на ленту текст с имеющимся в памяти машины. Для этого надо ввести команду "AP2"+"J" и загрузить текст с магнитной ленты. Если тексты не совпадают, на экране появится сообщение "ОШИБКА", а если полностью совпадают - их начальный фрагмент.

С помощью редактора можно компоновать текст из нескольких фрагментов, которые в данном случае вводятся с магнитофона командой "AP2"+"M"

Любую команду в режиме обмена Вы можете отменить, нажав клавишу "СТР".

7.11. Графический редактор

Программа составлена в кодах микропроцессора и загружается в память ПЭВМ по директиве "I" "СИСТЕМНОГО МОНИТОРА". Адрес запуска программы - 0000.

Графический редактор предназначен для оперативного создания различных иллюстративных материалов: графиков, схем, заставок и т.д.

При работе с редактором экран содержит 128 точек по горизонтали и 64 точки по вертикали. В этом случае вертикальные линии, так же как и горизонтальные, отображаются как непрерывные. Такой формат отображения достигается соответствующей программной настройкой контроллера KP580BG75. Графический редактор позволяет организовать в памяти ПЭВМ 9 экранных буферов с номерами 1...9 для хранения графической информации.

Работа с редактором заключается в следующем. После выполнения директивы "G" и "BK" на экран выводится список директив. Выдав соответствующую директиву, пользователь может создавать новые изображения в ручном или автоматическом режиме, сохранять их на магнитной ленте, загружать с магнитной ленты ранее подготовленные изображения и вносить в них изменения, в автоматическом режиме выводить изображения на экран в произвольном порядке и сопровождать их звуковыми эффектами.

Редактор позволяет выводить на экран и алфавитно-цифровую информацию. Символы при этом формируются в режиме псевдографики с использованием матрицы 6x8 точек. Возможен вывод до семи строк по 20 символов в каждой (используется тот же формат отображения и тот же набор символов, что и ПЗУ знакогенератора).

7.11.1. Ручной режим создания изображений

После выдачи директивы "PN" на экране дисплея отображается текущее содержимое буфера, номер которого определен параметром N (от 1 до 9). При первоначальном запуске редактора проводится очистка всех буферов. При последующих запусках редактора содержимое буферов не стирается. При вводе директив и их параметров последний набранный символ может быть удален клавишей "_". Ввод директивы заканчивается нажатием клавиши "BK". Директивы, набранные с какими-либо ошибками, просто игнорируются, при этом выдается звуковой сигнал ошибки (высокий тон, затем низкий).

В качестве графического курсора используется мигающий квадратик, равный по размерам одной графической точке. С помощью команд, перечисленных табл. 25, его можно перемещать в любое место экрана.

Команды ручного режима формирования изображения программы "Графический редактор"

Таблица 25

КЛАВИША	КОМАНДА	АНАЛОГ
←	Курсор влево	
→	Курсор вправо	УС+Н
	Курсор вверх	УС+У
	Курсор вниз	УС+З
F2	Курсор вверх-вправо	УС+А
	Курсор вверх-влево	УС+Л
F5	Курсор вниз-вправо	УС+D
F3	Курсор вниз-влево	УС+В
УС+Р	"прозрачный" курсор	
УС+К	Курсор - "карандаш"	
УС+R	Курсор - "ластик"	
УС+F	Пометить начало отрезка	
УС+S	Провести отрезок	
УС+M	Пометить начало блока	ВК
УС+J	Пометить конец блока	ПС
УС+N	Переслать блок в позицию курсора	
УС+I	Проинвертировать содержимое блока	ГТ
УС+E	Стереть блок	
УС+U	Снять определение блока	
УС+]	Сдвинуть экран вверх на 2 точки	
УС+T	Сдвинуть экран вниз на 2 точки	
УС+O	Сдвинуть экран влево на 2 точки	
УС+V	Сдвинуть экран вправо на 2 точки	
AP2	Включить текстовый экран	УС+]

При перемещении курсора по экрану возможны три режима:

- 1) курсор перемещается по экрану, не изменяя его содержимого ("прозрачный" курсор; команда "УС"+"Р");
- 2) курсор оставляет след в виде светлой линии ("карандаш"; команда "УС"+"К");
- 3) курсор стирает точки на экране ("ластик"; команда "УС"+"R").

Кроме получения изображений на экране с помощью курсора - "карандаша", оператор может формировать отрезки прямых линий следующим образом. Курсор перемещается в то место экрана, где будет начало отображаемого отрезка и пользователь выдает команду "УС"+"F". Затем курсор помещается в то место, где будет конец отрезка (предварительно следует включить режим

"прозрачного" курсора) и выдается команда "УС"+"S". На экране сразу появляется отрезок, соединяющий отмеченные точки. Установленные с помощью команды "УС"+"Г" координаты начала отрезка сохраняются до того момента, когда эта команда будет выдана вновь. Таким образом можно построить несколько отрезков с общим началом, повторяя команду "УС"+"S" при различных положениях курсора.

При формировании изображений часто возникает необходимость переноса части изображения из одного места экрана в другое или стирания его. Для этого служат команды работы с блоком. Под блоком понимается прямоугольная область на экране, заданная координатами левого нижнего и правого верхнего углов. В любой момент работы редактора может быть определен только один блок.

Для того, чтобы инициализировать блок, оператор помещает курсор в позицию, соответствующую левому нижнему углу блока, и выдает команду "УС"+"М". Затем курсор перемещается в позицию правого верхнего угла, после чего необходима команда "УС"+"J". Для того, чтобы получить копию блока в другом месте экрана, пользователь должен переместить курсор в то место, где будет располагаться левый нижний угол копии, и ввести команду "УС"+"N". Команда копирования сохраняет определение блока и поэтому его можно копировать неограниченное число раз.

Команда "УС"+"U" отменяет заданное ранее определение блока. Команда "УС"+"E" стирает все, что находится внутри помеченного блока. После ее выполнения определение блока снимается. Команда "УС"+"I" инвертирует изображение, находящиеся внутри блока.

Манипуляции с блоком можно проводить не только в пределах одного экрана. Блок, помеченный на одном из экранов, можно скопировать на любом другом. Для этого служит директива копирования экранов "KN1,N2", где N1-экран источник, N2-экран получатель. Нажатие клавиши "AP2" в ручном режиме всегда приводит к выводу на экран списка директив.

Для того чтобы оператор мог легко ориентироваться в установленных режимах работы, последняя строка экрана содержит так называемую строку состояния. Назначение отдельных полей строки следующее:

К= - вид курсора (Л - "ластик", К - "карандаш", П - "прозрачный") ;

Э= - текущий номер графического экрана;

Х= - позиция курсора по горизонтали;

У= - позиция курсора по вертикали;

XL= - координата начала отрезка по горизонтали;

YL= - координаты начала отрезка по вертикали;

В= - номер графического экрана на котором помечен блок;

X1= - координата левого нижнего угла по горизонтали;

Y1= - координата левого нижнего угла по вертикали;
X2= - координата правого верхнего угла по горизонтали;
Y2= - координата правого верхнего угла по вертикали.

Директива "У" разрешает или запрещает отображение строки состояния. Первоначально отображение строки разрешено, о чем свидетельствует буква "Д" в круглых скобках. При каждой выдаче команды это условие меняется на противоположное.

Для стирания всего содержимого экрана служит директива SN, где N определяет номер экрана.

Выход из ручного режима осуществляется нажатием клавиши "AP2".

7.11.2. Работа с магнитофоном

Подготовленные в ручном режиме изображения могут быть сохранены на магнитной ленте и в дальнейшем вновь введены в память ПЭВМ. Для записи служит директива 3N1,N2. Если N2 не указан, то на ленте записывается только содержимое экрана с номером N1. В противном случае на ленте будет сохранено содержимое экранов с номерами от N1 до N2 включительно.

Для чтения с магнитофона предусмотрена директива "Ч", по которой информация вводится в то место памяти, откуда она была записана.

7.11.3. Автоматический режим формирования изображений

Одна из особенностей описываемого редактора - возможность автоматического формирования изображений и управление последовательностью их отображения на экране. Для этого оператор должен подготовить специальную макрокоманду, описывающую требуемые действия.

Макрокоманда может вводиться как с клавиатуры, так и с магнитной ленты. На экране (поле макрокоманды) для нее отведены 17 строк. Таким образом, максимальная длина макрокоманды 1088 символов (17х64).

Ввод макрокоманды с клавиатуры начинается с выдачи директивы "Ф". При этом текстовый курсор помещается в левом верхнем углу поля макрокоманды. Клавиша "_" служит для стирания символа, стоящего слева от курсора. Вся информация в поле макрокоманды стирается путем двухкратного нажатия клавиши "СТР". Набрав текст макрокоманды, необходимо нажать клавишу "ВК", после чего на экране появится светлый прямоугольник - признак конца макрокоманды. Выполнение директивы "Ф" заканчивается нажатием клавиши "AP2". Запуск макрокоманды - по директиве "Г".

Операторы макрокоманды программы "Графический редактор"

Таблица 26

ОПЕРАТОР	НАЗНАЧЕНИЕ	ОПЕРАНДЫ
DN	Назначить буфер отображения	N - номер буфера
WN	Назначить буфер для записи	--" "--
Y	Очистить текущий буфер	Нет
CN1,N2	Копирование содержимого буфера	N1 - источник N2 - получатель
UN	Сдвинуть экран вверх на nx2 точки	N-число
NN	Сдвинуть экран вниз на nx2 точки	--" "--
EN	Сдвинуть экран влево на nx2 точки	--" "--
HN	Сдвинуть экран вправо на nx2 точки	--" "--
PX,Y,Z	Высветить точку	X,Y - координаты Z - контраст
LX,Y	Провести отрезок	X,Y - координаты конца отрезка
BX1,Y1,X2,Y2	Определить блок в текущем буфере	X1,Y1 - координаты левого нижнего X2,Y2 - правого верхнего углов
I	Проинвертировать изображение блока	Нет
J	Стереть блок	--" "--
M	Переслать блок в позицию курсора	--" "--
Z	Снять определение блока	--" "--
SN	Начало цикла 1	N - число повторений
Q	Конец цикла 1	Нет
FN	Начало цикла 2	N - число повторений
O	Конец цикла 2	Нет
T"СТРОКА"	Выдача на экран строки	Строка в кавычках
K	Ожидание ввода любого символа	Нет
GN1,N2	Выдача звукового сигнала	N1-высота тона N2-длительность
XN	Временная задержка	N - число

С помощью директивы "Т" макрокоманду можно записать на магнитную ленту, а директивы "Л" - считать с ленты в ОЗУ ПЭВМ. Следует учесть, что так как текст макрокоманды хранится только в области текстового экрана, то при случайном нажатии клавиши "СБРОС" он будет утерян. Поэтому рекомендуется сохранять макрокоманды на магнитной ленте.

Макрокоманда состоит из последовательности операторов

и операндов. Операнды разделяются с помощью символов ",", ". Между отдельными операторами допускается помещать любое число пробелов. Между операторами и операндами не должно быть других символов.

Перечень операторов макрокоманды приведен в табл. 26. Большинство действий по созданию и изменению графических изображений в ручном режиме могут быть описаны операторами макрокоманды. Выполнение оператора "DN" приводит к выдаче на дисплей содержимого экрана с номером N. При этом он автоматически становится буфером для чтения и записи, т.е. текущим буфером.

Пример:

Макрокоманда D1T"МИКРОША"K выведет на первый экран надпись "МИКРОША" и будет ожидать нажатия любой клавиши, после чего на экране вновь появится список директив.

Чтобы подготовить изображение в "невидимом" в данный момент буфере, необходимо назначить его буфером для записи с помощью директивы "WN".

Пример:

Макрокоманда D1T"ПЭВМ"W2T"МИКРОША"KD2K выводит на экран содержимое первого буфера - надпись "ПЭВМ", формирует на втором экране надпись "МИКРОША" и ожидает нажатия любой клавиши, после чего выводит на экран содержимое второго буфера и вновь ожидает нажатия клавиши.

Таким образом модификация содержимого буфера возможна только в том случае, если этот буфер назначен буфером для записи директивой "WN".

Оператором PX,Y,Z меняется позиция графического курсора. Третий операнд этого оператора задает контраст операторов "P" и "L". При Z=1 точка высвечивается, а при Z=0 - гаснет.

Графический редактор обладает возможностью организовывать циклы, что особенно важно при создании движущихся изображений. Циклы обозначаются соответственно SN-Q и FN-O. SN и FN - признаки начала цикла, Q и O - признаки конца цикла, N - параметр, определяющий число повторений. Допускается организация вложенных циклов: S-F-O-Q и F-S-Q-O. Вложения S-F-Q-O и F-S-O-Q являются недопустимыми.

Пример:

Макрокоманда D1P20,20,1T"МИКРОША"B18,18,20,28S10IQ выведет на экран сообщение "МИКРОША" и 10 раз проинвертирует его.

Редактирование текста макрокоманды производится с помощью функциональных клавиш, перечисленных в табл. 27.

Таблица 27

КЛАВИША	НАЗНАЧЕНИЕ	АНАЛОГ
←	Курсор влево на одну позицию	УС+H
→	Курсор вправо на одну позицию	УС+X
↑	Курсор вверх на одну позицию	УС+Y
↓	Курсор вниз на одну позицию	УС+Z
↖	Курсор в левый верхний угол поля	УС+L
" _ "	Удаление символа слева от курсора	
СТР (2 РАЗА)	Стереть всю макрокоманду	
AP2	Закончить ввод (редактирование) макрокоманды	

После запуска макрокоманды с помощью директивы "Г" она начинает выполняться. Если обнаруживается какая-либо ошибка, то выполнение макрокоманды заканчивается и выдается звуковой сигнал ошибки. Остановить выполнение макрокоманды можно нажатием клавиши "F4".

Макрокоманды можно запустить и без выхода в основной список директив. Для этого, находясь в режиме формирования макрокоманды, необходимо поместить текстовый курсор под тот оператор, начиная с которого будет выполняться макрокоманда, и нажать "УС+G". Эту возможность используют для отладки макрокоманд.

7.12. Игровая программа "Быстрый счет"

Программа предназначена для развития навыков быстрого счета.

Программа загружается директивой "Г" "СИСТЕМОГО МОНИТОРА". Запуск осуществляется нажатием клавиш "G" и "BK". Программа имеет встроенный Бейсик, поэтому после выполнения команды "G" и "BK" на экране появится надпись "МИКРОША BASIC". Перейти к игре можно по директиве Бейсика RUN.

В игре участвуют два "бегуна", один из которых (нижний) бежит с одинаковой скоростью, не останавливаясь. Верхний "бегун" бежит чуть быстрее нижнего, но на его пути появляются препятствия в виде арифметических задач на сложение (+), вычитание (-), умножение (x), деление (:), возведение в степень и извлечение корня. При появлении в верхней части экрана задачи Ваш "бегун" останавливается и ждет Вашего решения. Чем быстрее Вы введете правильный ответ, тем больше у него шансов первым достичь финиша. Если Вы неправильно подсчитали или ввели результат, на экране появится надпись "НЕВЕРНЫЙ ОТВЕТ" и задача снова появится на экране. В начале игры появляется вопрос

"УРОВЕНЬ СЛОЖНОСТИ?", на который Вам необходимо ответить вводом числа от 1 до 6.

7.13. Игровая программа "Удав"

Программа написана в машинных кодах микропроцессора и загружается в ПЭВМ по директиве "I" программы "СИСТЕМНЫЙ МОНИТОР". Запуск программы производится нажатием клавиш "G" и "BK". На экране появляется название программы. Нажмите клавишу "ПРОБЕЛ", после чего выберите скорость и сложность игры. Запуск игры производится нажатием одной из клавиш: "F4", "←", "→". Этими же клавишами и клавишей "F1" Вы управляете удавом. Задача Вашего удава - съесть как можно больше мышей (периодически появляющийся символ "\$"), не наткнуться на препятствие, не наступить себе на хвост и благополучно выбраться за пределы игрового поля через окно в верхней границе экрана. Если эта задача Вами выполнена, следует переход на более сложный экран.

7.14. Игровая программа "XONIX"

Программа загружается с помощью директивы "I" "СИСТЕМНОГО МОНИТОРА". Запуск программы производится нажатием клавиш "G" и "BK", переход от заставки к игре - нажатием клавиши "ПРОБЕЛ". Управление осуществляется клавишами "↑", "↓", "←", "→".

Перед Вами игровое поле, по которому двигаются символы "0", по рамке поля передвигаются темные квадраты.

Нажмите клавишу "→". В верхней части рамки поля Вы увидите "себя" - засвеченный квадрат. Ваша задача состоит в том, чтобы захватить как можно большую часть игрового поля путем отсекаания от него отдельных участков. При этом Вам следует избегать встреч с "0" и темными квадратами. Такая встреча приводит к потере попытки. К такому же результату приводит и намерение двигаться назад по своей траектории. Если Вам удалось занять достаточно большую часть игрового поля, то следует переход на следующий экран с увеличением числа "0", темных квадратов и попыток. Если при переходе на следующий экран Вы не потеряли ни одной попытки, на экране вместо цифр в графе попыток будут выводиться символы ":", ";" и т.д., что соответствует количеству количеству попыток 10, 11 и т.д.

7.15. Игровая программа "TETRIS"

Программа загружается с помощью директивы "I" "СИСТЕМНОГО МОНИТОРА". Запуск программы производится нажатием клавиш "G" и "BK".

Перед Вами заставка к игре и просьба о вводе уровня сложности. Нажав одну из клавиш "0"... "5" и "BK", Вы можете перейти непосредственно к игре.

Игра заключается в следующем. В колодец одна за другой

падают фигурки различной формы. Управлять траекторией падения можно с помощью клавиш "←", "→" и кроме этого имеется возможность клавишей "↓" поворачивать падающую фигурку. Нажатием клавиши "ПРОБЕЛ" можно ускорить движение фигурки. Ваша задача - уложить в колодце как можно большее число "кирпичиков". Если Вам удалось полностью заполнить одну из строк, то она исчезает, освобождая тем самым место для размещения новых фигурок. За каждую уложенную фигурку начисляется очко. Остановка игры происходит в том случае, если в колодце не осталось места для размещения очередной фигурки. При этом на экран выводится сообщение об окончании игры. Переход к началу осуществляется нажатием клавиши "BK".

7.18. Игровая программа "РАСМАН"

Программа загружается с помощью директивы "I" "СИСТЕМНОГО МОНИТОРА". Запуск осуществляется командой "G6000" и нажатием клавиши "BK".

После выполнения указанных операций на экране появится надпись "YOUR RANG (0...9)?". Нажав одну из указанных клавиш, Вы проинформируете "Микрошу" об уровне Вашей подготовки.

Перед Вами лабиринт, по которому передвигаются символы #, @, &, \$. Это людоеды, и встреча с ними приводит к потере одной попытки. В центре лабиринта Вы видите символ "O", движением которого можно управлять с помощью клавиш "↑", "↓", "←", "→". Передвигаясь по экрану, "O" уничтожает встречающиеся на его пути точки, за каждую "съеденную" точку получая одно очко. По углам лабиринта расположены четыре символа S. Если при движении "O" достигнет одного из них, то людоеды теряют силу (о чем свидетельствует их превращение в буквы A, B, V, Г). В этом случае нолику выгодна встреча с людоедами, так как за каждого уничтоженного неприятеля начисляется 100 очков. Однако такое "расслабленное" состояние людоедов продолжается очень недолго, поэтому в погоне за буквами будьте внимательны. В центре лабиринта периодически появляются призы в виде букв латинского алфавита. За "соединение" приза начисляется 200 очков. На прохождение лабиринта Вам предоставляется три попытки. Если все они неудачны, то на экране вновь появляется надпись "YOUR RANG (0...9)?" и, после ввода уровня сложности, происходит автоматический перезапуск игры. Сообщение о набранных очках и об оставшихся попытках выводится в правом верхнем углу экрана. Если Вам удалось пройти весь лабиринт, уничтожив при этом все точки, то игра переходит на более сложный уровень с одновременным увеличением на единицу числа попыток.

7.17. Игровая программа "Лестница"

Программа составлена в кодах микропроцессора и загружается в память по директиве "I" "СИСТЕМНОГО МОНИТОРА".

После выполнения директивы "G" и "BK" на экране появятся правила игры и другая необходимая информация. Задача играющего - пройти с нижнего уровня лестницы до ее верха, не встретившись при этом с падающими камнями, не попав в ловушку. Если задача выполнена, следует переход на следующий экран, если нет - возвращение на место старта.

7.18. Игровая программа "Клад"

Программа составлена в кодах микропроцессора и выводится в память ПЭВМ по директиве "I" "СИСТЕМНОГО МОНИТОРА". Адрес запуска - 0000.

Правила игры и задача играющего приведены на заставке к игре. Нажатием клавиши можно переключать игровые экраны.

7.19. Игровая программа "Цирк"

Игровая программа "Цирк" составлена в машинных кодах и загружается в память ПЭВМ с помощью директивы "I" "СИСТЕМНОГО МРНИТОРА". Запуск программы производится директивой "G100".

После запуска программы на экран выводится заставка к игре. Нажатием клавиши "S" производится выбор скорости. При нажатии клавиши "Z" следует переход от заставки к игровому циклу.

Игра имитирует выступление циркового гимнаста-акробата. Под куполом цирка расположены призы: символы "&". По арене перемещается батут. Управление его движением осуществляется клавишами "←" и "→". Задача играющего заключается в том, чтобы не дать, разбиться акробату. Добиться этого можно путем перемещения батута. Отталкиваясь от него, акробат взлетает под купол цирка, причем высота прыжка определяется правилом: чем ближе к краю батута было произведено отталкивание, тем выше прыжок. По пути гимнаст собирает призы, получая за каждый определенное количество очков. Играющему предоставляется пять попыток, и игра будет продолжаться до тех пор, пока все они не будут израсходованы.

8. ТЕХНИЧЕСКОЕ ОБСЛУЖИВАНИЕ

При эксплуатации ПЭВМ "Микроша" обычно не требуется каких-либо дополнительных мер по техническому обслуживанию, однако через определенное время может возникнуть такой дефект, как дребезг контактов

клавиатуры. Дефект возникает в результате изменения положения крепящей пластины нижнего контакта клавиши (см. рис. 8). Конструкция клавиши предусматривает возможность устранения дребезга.

Для этого необходимо проделать следующие операции:

- 1) отключить питание ПЭВМ;
- 2) снять клавишу с основания для чего приложить к ней некоторое усилие в вертикальном направлении;
- 3) увеличить зазор между верхним и нижним контактами путем отгибания вниз пластины, установленной над нижним контактом, изгибать сами контакты не допускается, так как они могут сломаться при деформации;
- 4) при наличии оксидной пленки на рабочих поверхностях контактов аккуратно протереть их ластиком;
- 5) установить клавишу на место.

Остальные работы по техническому обслуживанию ПЭВМ должны производиться только специалистами ремонтных предприятий.

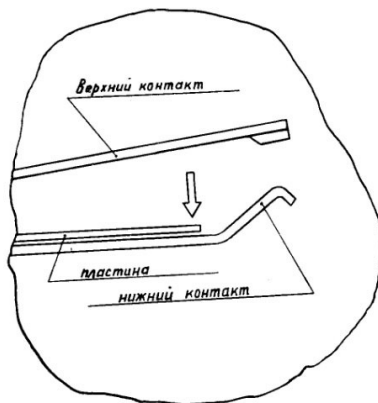


Рис. 8

9. ГАРАНТИЙНЫЕ ОБЯЗАТЕЛЬСТВА

Изготовитель гарантирует соответствие качества ПЭВМ требованиям технических условий при соблюдении владельцем правил Эксплуатации, изложенных в настоящем руководстве.

Изготовитель не гарантирует работоспособность ПЭВМ в случае подключения к разъемам системного блока устройств, не предусмотренных руководством по эксплуатации.

Гарантийный срок эксплуатации - 12 месяцев со дня продажи через розничную торговую сеть. При отсутствии в гарантийном талоне и отрывных талонах на гарантийный ремонт даты продажи и штампа магазина, подписи или штампа продавца гарантийный срок исчисляется с даты выпуска ПЭВМ заводом.

В течение гарантийного срока эксплуатации владелец имеет право на бесплатное техническое обслуживание и бесплатный ремонт по предъявлении гарантийного талона. При этом за соответствующий вид технического обслуживания вырезается один отрывной талон. Без предъявления гарантийного и отрывных талонов и при нарушении сохранности пломб претензии к качеству работы не принимаются и гарантийный ремонт не производится.

В течение гарантийного срока эксплуатации ремонт производится за счет владельца в том случае, если эксплуатация ПЭВМ производилась с нарушением правил, изложенных в настоящем руководстве.

Обмен неисправных ПЭВМ осуществляется через торговую сеть по предъявлении справки ремонтного предприятия и гарантийного талона в соответствии с действующими республиканскими правилами обмена промышленных товаров, купленных в розничной торговой сети государственной и кооперативной торговли.

Гарантийный ремонт ПЭВМ "Микроша" производится в следующих предприятиях:

1. Гарантийная мастерская ЛЭМЗ.

127411, г. Москва, Дмитровское шоссе, д. 115. Тел. 485-15-27.

2. Новгородское ПО "Экран".

173002, г.Новгород, ул. Германа, д. 22. Тел. 7-73-30.

Зона обслуживания: г. Новгород и Новгородская обл.

3. Научно-производственный кооператив "Импульс".

480083, г. Алма-Ата, ул. Шагабутдинова, 31-21.

Зона обслуживания: г. Алма-Ата и Алма-Атинская обл.

4. Казахское ПО ВТИ.

480012, г. Алма-Ата, ул. Комсомольская, д. 101.

Зона обслуживания: Казахская ССР.

5. Горьковское ПО ВТИ.

603109, г. Горький, ул. Краснофлотская, д. 56.

Зона обслуживания: Горьковская, Кировская обл.,
Марийская, Мордовская, Чувашская АССР.

6. Латвийское ПО ВТИ.

226063, г. Рига, ул. Кенгарага, д. 10. 215

Зона обслуживания: Латвийская ССР.

7. Белорусское ПО ВТИ.

220029, г. Минск, ул. Островского, д. 12а.

Зона обслуживания: Белорусская ССР.

8. Узбекское ПО ВТИ.

700096, г. Ташкент, ул. Муками, д. 43.

Зона обслуживания: Узбекская ССР.

9. Куйбышевское ПО ВТИ.

443090 г. Куйбышев, ул. Антонова-Овсеенко, д. 44

Зона обслуживания: Куйбышевская и Саратовская обл.

10. Московское ПО ВТИ.

103030, г. Москва, Перуновский пер., д. 3 строение 2.

Зона обслуживания: Брянская, Владимирская,
Ивановская, Калужская, Калининская, Костромская,
Московская, Орловская, Рязанская, Смоленская,
Тульская, Ярославская области.

11. Казанские ПО ВТИ.

420044, г. Казань, пр. Ямашева, д. 36.

Зона обслуживания: Татарская АССР. 4

12. Армянское ПО ВТИ.

375038, г. Ереван, ул. Абеяна, д. 6/1.

Зона обслуживания: Армянская ССР.

13.Новосибирское ПО ВТИ.

630038 г. Новосибирск, ул. Петухова, 16/1.

Зона обслуживания:Кемеровская, Новосибирская, Омская,
Томская, Тюменская области, Алтайский край.

14.Запорожское предприятие ВТИ.

330089 г. Запорожье, ул. Песчаная,3

Зона обслуживания: Украинская ССР.

Лианозовский электромеханический завод

Цена 500 руб.

Дополнительный прейскурант N 139-1977/105 п. 868

ГАРАНТИЙНЫЙ ТАЛОН

Заполняется заводом — изготовителем

Машина вычислительная электронная персональная "Микроша"

заводской номер _____ дата выпуска _____

заводской номер блока питания _____

заводской номер модулятора _____

Представитель ОТК завода-изготовителя _____
(штамп ОТК)

Адрес для предъявления претензий к качеству:
127411, г. Москва, ЛЭМЗ

Заполняется торговым предприятием

Дата продажи _____ Продавец _____

Штамп магазина

Заполняется ремонтным предприятием

Поставлен на гарантийное обслуживание _____

(наименование ремонтного предприятия, число, месяц, год)

Гарантийный номер _____

!

!

! Действителен по заполнении

!

! Лианозовский электромеханический завод

к т !

о е ! ОТРЫВНОЙ ТАЛОН НА ТЕХНИЧЕСКОЕ ОБСЛУЖИВАНИЕ

р х л !

е н и ! Заполняется заводом-изготовителем

ш и н !

о ч и ! Машина электронная вычислительная персональная

к е я ! "Микроша"

с !

о к ! Заводской номер _____ дата выпуска _____

т о ! Представитель ОТК завода-изготовителя _____

р е ! (штамп ОТК)

ы о !

в о т ! Заполняется торговым предприятием

н б р !

о с е ! Дата продажи _____

г л з ! (число, месяц, год)

о у а !

ж ! Продавец _____

т и ! (подпись или штамп)

а в ! Штамп магазина

л а !

о н ! Заполняется ремонтным предприятием

н и !

а я ! Регистрационный номер ПЭВМ _____

! Содержание работ по техническому обслуживанию _____

! _____

! _____

! _____

! _____

! Дата выполнения работ _____

! (число, месяц, год)

! Подпись лица, выполнявшего работу _____

! Подпись владельца ПЭВМ, подтверждающая техни-

! ческое обслуживание _____

! Штамп ремонтного предприятия _____

!

!

BA1	Головка громкоговорителя динамическая 0,25-ГД-19 ГОСТ 9010-84
BQ1	Резонатор кварцевый РК 169МА-14БВ- 16000кГц-В ОДО.338.017 ТУ

Конденсаторы

C1	КМ-66-Н90-0,47 мкф ОЖО.460.061 ТУ
C2	К50-16А-25В-5 мкф ОЖО.464.111 ТУ
C3	К50-16А-16В-10 мкф ---"
C4	КМ-56-Н90-0,22 мкф ОЖО.460.043 ТУ
C5	КМ-56-Н90-0,1 мкф ---"
C6, C7	КМ-56-Н90-0,022 мкф ---"
C8	КМ-56-М1500-3300 пф ---"
C9, C10	КМ-56-Н90-0,1 мкф ---"
C11...C34	КМ-56-Н90-0,068 мкф ---"
C35...C38	КИ-56-Н90-0,1 мкф ---"
C39	КМ-56-М1500-3300 пф ---"
C40...C42	КМ-56-Н90-0,022 мкф ---"

Микросхемы

D1	КР580ГФ24	БКО.348.745-14 ТУ
D2	КР580ВТ57	БКО.348.745-09 ТУ
D3	К155ИЕ4	БКО.348.006 ТУ
D4	К555ЛИ1	БКО.348.289-01 ТУ
D5	КР580ВМ80А	БКО.348.745-08 ТУ
D6	К555ЛП5	БКО.348.289-10 ТУ
D7	К589ИР12	БКО.348.319 ТУ 4
D8	КР580ВГ75	БКО.348.693 ТУ
D9	К555ЛН1	БКО.348.289-01 ТУ
D10	К555ЛА3	БКО.348.289-01 ТУ
D11	К555ИД7	БКО.348.289-02 ТУ
D12	КР580ВВ55А	БКО.348.745-01 ТУ
D13	К573РФ2	БКО.348.422-02 ТУ
D14	К155ТМ2	БКО.348.006-01 ТУ
D15	К573РФ2	БКО.348.422-02 ТУ
D16	К155ИР13	БКО.348.006-38 ТУ
D17	К555ИД4	БКО.348.289-08 ТУ
D18	К155ИР1	БКО.348.005-05 ТУ
D19, D20	К555КП11	БКО.348.289-14 ТУ
D21	К155ЛА12	БКО.348.006-37 ТУ
D22	КР580ВИ53	БКО.348.745-10 ТУ
D23...D38	КР565РУ6Д	БКО.348.731 ТУ
D39	КР580ВВ55А	БКО.348.745-01 ТУ
D40	КР140УД608	БКО.348.095-03 ТУ

L1...L3	Дроссель	ЯВ4.752.064
---------	----------	-------------

Резисторы МЛТ-0,125 ГОСТ7113-77

R1	МЛТ-0,125-2 кОм	±10%
R2	МЛТ-0,125-1 кОм	-"-
R3...R5	МЛТ-0,125-10 кОм	-"-
R6	МЛТ-0,125-1 кОм	-"-
R7...R12	МЛТ-0,125-10 кОм	-"-
R13...R15	МЛТ-0,125-1 кОм	-"-
R16	МЛТ-0,125-20 кОм	-"-
R17	МЛТ-0,125-1 кОм	-"-
R18	МЛТ-0,125-20 кОм	-"-
R19	МЛТ-0,125-51 Ом	-"-
R20	МЛТ-0,125-150 Ом	-"-
R21.R22	МЛТ-0,125-1 кОм	-"-
R23...R34	МЛТ-0,125-33 Ом	-"-
R35	МЛТ-0,125-27 Ом	-"-
R36.R37	МЛТ-0,125-1 кОм	-"-
R38	МЛТ-0,125-12 кОм	-"-
R39	МЛТ-0,125-2,4 кОм	-"-
R40	МЛТ-0,125-1 кОм	-"-
R41...R43	МЛТ-0,125-4,7 кОм	-"-
R44...R45	МЛТ-0,125-300 Ом	-"-
R46...R53	МЛТ-0,125-4,7 кОм	-"-

S1...S69	ВМ16-4 ОЮО.305.008 ТУ
	ВМ16-1 ОЮО.305.008 ТУ

VD1...VD2	Диод КД522Б	ДРЗ.362.029 ТУ
VD3...VD10	Диод Д9Б	СМЗ.362.015 ТУ
VD11.VD12	Светодиод АЛ307БМ	аАО.336.076 ТУ

VT1	Транзистор КТ315Б	ЖКЗ.365.200 ТУ
-----	-------------------	----------------

XI	Розетка ЯБ6.604.039
X2...X5	Соединитель Онц-кГ-4-5/16-Р ГОСТ 12368-78 X6
	Розетка СНП34С-30/27х9,4Р-22-В
	НЩО.364.052 ТУ