

Министерство радиопромышленности СССР

Утвержден

ИП.00045-ОТ 34 ОТ ЛУ

ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ МВЦ  
"ПАРТНЕР ОI.OI"

Руководство оператора

ИП.00045-ОI 34 ОI

Листов 68

АНОТАЦИЯ

Данное руководство описывает работу с программным обеспечением МВЦ 'Партнер 01.01'.

Программное обеспечение (ПО) включает системныерезидентные программы Монитор, Ассемблер, Редактор и Бейсик, а также загружаемые с магнитофона игровые программы.

СОДЕРЖАНИЕ

	Лист
1. Назначение	4
2. Запуск программы	4
3. Директивы МОНИТОРА	5
4. Запросы МОНИТОРА	12
5. Работа с видеоЗУ	17
6. Директивы РЕДАКТОРА и АССЕМБЛЕРА	19
7. Интерпретатор языка Бейсик	23
8. Описание программного обеспечения, поставляемого в комплекте с МВЦ	38
9. Дополнительные средства монитора для работы с магнитофоном.	42
Приложение 1	50
Приложение 2	51
Приложение 3	54
Приложение 4	55
Лист регистрации изменений	56

## 1. НАЗНАЧЕНИЕ

### 1.1. Назначение системных программ

Программа МОНИТОР выполняет функции минимальной операционной системы, т.е. обеспечивает управление устройствами машины (контроллерами клавиатуры, магнитофона, дисплея), предоставляет пользователю возможность загрузки программ с магнитофона (МГ) или из ПЗУ, запуска их и записи на МГ, а также необходимые средства для отладки программ, написанных на языке Ассемблера. Данная версия программы Монитор в основном совместима с аналогичной программой ЭВМ типа 'Радио 86 РК' (Микроша, Криста), что позволяет использовать ПО этих машин.

Программы РЕДАКТОР и АССЕМБЛЕР, работающие совместно, обеспечивают ввод с МГ или клавиатуры программ на языке Ассемблера, трансляцию с получением листинга и загрузочного текста, а также запись на МГ загрузочных и исходных текстов программ.

С помощью РЕДАКТОРА можно вводить, редактировать и записывать на МГ произвольные тексты.

Интерпретатор языка Бейсик служит для выполнения директив и программ языка Бейсик.

Язык высокого уровня Бейсик предназначается для использования профессиональными программистами при разработке широкого класса программ, а также для применения неподготовленными пользователями при решении простых вычислительных задач на уровне калькулятора и при эксплуатации ими готовых программных изделий в качестве языка управления заданиями.

Логической основой языка Бейсик служат стандартные средства языка Бейсик.

5  
1100045-01 5401

## 1.2. Распределение адресного пространства.

Распределение адресного пространства дано в приложении 5.

Оператор имеет возможность работать с десятью картами памяти (с десятью вариантами распределения адресного пространства). Для переключения карт используется регистр конфигурации, который находится по адресу DA00H. По умолчанию при нажатии клавиши "сброс" устанавливается восьмая карта. Для установки МВЦ в карту NX, необходимо записать в регистр конфигурации код X0H. Например:

карта	код
0	00
1	10
2	20
3	30
4	40
5	50
6	60
7	70
8	80
9	90
10	A0

Пример переключения карт по директиве МОНИТОРА:

```
MDA00 <BK>
DA00 FF 00 (карта 0)
DA01 FF .
```

Внимание пользователю: Работать рекомендуется в 1, 2, 8 картах. Остальные карты используются для технологических целей.

ОЗУ МВЦ разбито на две части - "ОЗУ1" с 0 до 32K и "ОЗУ2" с 32K до 64K.

В качестве примера рассмотрим распределение адресного

6  
1100045-01 34 01

пространства восьмой карты.

Обозначение сигналов : N контакт разъема X1...X4 :

MASK RAML	:	B10	:
MASK ROML	:	B21	:
PORT1L	:	B20	:
PORT2L	:	B24	:
SEL.SLOTL	:	B25	:
SEL.INTL	:	B29	:

Эти сигналы предназначены для управления ресурсами (порты ввода-вывода, ОЗУ, ПЗУ) расположеннымми в дополнительных модулях, подключаемых к МВЦ.

С 0 до 7FFFH (32K) располагается "ОЗУ1".

С 8000H до 9FFFH (8K) и с E000H до E7FFFH (2K) находится "OKHO1", через которое осуществляется подключение дополнительного ПЗУ, при обращении по этим адресам активизируется сигнал MASK ROML.

С A000H до BFFFFH (8K) располагается "ПЗУ1" - Бейсик.

При запуске Бейсика по директиве 'B' он из ПЗУ загружается в ОЗУ с 0 по 2000H.

С C000H до C7FFFH (2K) и с E800H до FFFFFH (6K) располагается "ПЗУ2" - Ассемблер, Редактор, Монитор+тест.

С C800H до D7FFFH (4K) находится "OKHO2", через которое осуществляется подключение дополнительного ОЗУ, при обращении по этим адресам активизируется сигнал MASK RAML.

С D800H до DFFFFH (2K) во всех десяти картах располагаются устройства. Устройства занимают 8 областей. Четыре области отданы резидентным портам МВЦ, четыре - пользователю для подключения модулей.

Резидентные порты МВЦ:

D800H - D801H - адрес контроллера ЭЛТ KP580BГ75.

D900H - D903H - адрес контроллера клавиатуры и магнитофона KP580ББ55.

*7  
1100045-01 34 01*

**DAO0H** - адрес регистра конфигурации.

**DVO0H** - **DVOFH** - адрес контроллера ПДП КР580ВТ57.

**Пользовательские порты МВЦ:**

**DCO0H** - **DCFFFH** - адрес порта ввода-вывода общего назначения, при обращении по этому адресу активизируется сигнал **P0RT1L**.

**DDO0H** - **DDFFFH** - адрес порта ввода-вывода общего назначения, при обращении по этому адресу активизируется сигнал **P0RT2L**.

**DEO0H** - **DEFFFH** - адрес порта вывода, разрешает обращение к ресурсам модуля, при обращении по этому адресу активизируется сигнал **SEL.SLOTL**.

**DFO0H** - **DFFFFH** - адрес порта вывода, разрешение/запрещение прерывания от модуля, при обращении по этому адресу активизируется сигнал **SEL.INTL**.

## 2. ЗАПУСК ПРОГРАММ

Запуск программы МОНИТОР производится при включении машины после нажатия на клавишу сброса "С". После выполнения подготовительных операций программа выводит на экран сообщение: **ПАРТНЕР 01.01**, свидетельствующее о готовности к работе.

Дальнейшие действия пользователя с программой МОНИТОР

сводятся к заданию и выполнению директив и описаны в разделе 3.

Запуск программ АССЕМБЛЕР и РЕДАКТОР производится специальной директивой 'A', а запуск программы БЕЙСИК - директивой 'B'. Описание этих директив , а также сообщения программ приведены в разделе 3.

### 3. ДИРЕКТИВЫ МОНИТОРА

С помощью директив МОНИТОРА пользователь может управлять работой машины , просматривать и изменять содержимое ячеек памяти и регистров процессора , отлаживать программу , загружать программы с МГ или ПЗУ.

#### 3.1. Пуск МОНИТОРА.

Запуск программы МОНИТОР производится при включении машины после нажатия на клавишу сброса "С". Через некоторое время на экране появляется сообщение: ПАРТНЕР 01.01(В.02) ,свидетельствующее о готовности к работе.

Дальнейшие действия пользователя с программой МОНИТОР сводятся к заданию и выполнению директив.

#### 3.2. Общая форма записи директив.

«Символ»[Параметр 1],[Параметр 2],[Параметр 3] «ВК»

Имена всех директив состоят из одной латинской буквы , непосредственно за которой могут следовать не более трех параметров , представляющие собой шестнадцатиричные числа.

Параметры отделяются друг от друга запятой. Запятую ставят и в тех случаях , когда один из параметров ( в том числе и первый) отсутствует. Директивы заканчиваются нажатием клавиши ВК, в дальнейшем это будет подразумеваться.

При вводе директив возможны следующие ошибки : длинная строка, неправильное имя директивы , неправильный формат числа. Если эти ошибки произошли , то машина отвечает на них звуковым сигналом и на экране появляется знак вопроса. После чего необходимо повторить ввод директивы.

### 3.3. Работа с магнитофоном.

В работе с магнитофоном используются две директивы 'I' и 'O'. Подробнее эти директивы будут описаны в п.п. 3.3.1 и 3.3.2.

При работе с магнитофоном необходимо произвести следующие действия :

- 1) включить магнитофон ;
- 2) вставить кассету в МГ ;
- 3) если нужно считать программу , то ввести директиву 'I'.

Если нужно записать на магнитофон программу , ввести директиву 'O' .

4) нажать соответствующие клавиши магнитофона ( для записи или считывания ) и удостовериться , что место на ленте выбрано правильно;

5) при появлении тонального сигнала из магнитофона нажать на МВЦ клавишу ВК ;

6) при появлении звуковой паузы нажать на магнитофоне клавишу останова.

Воспроизводимый магнитофоном звуковой сигнал должен иметь наиболее чистый тон , для этого необходимо предварительно подкрутить винт универсальной головки воспроизведения магнитофона.

При загрузке информации в МВЦ и при записи на МГ изображение на экране ТВ пропадает ( до первой звуковой паузы).

#### 3.3.1. Директива 'O'.

По директиве 'O' на магнитофон выводится содержимое области памяти.

Форма директивы :

<O> [начальный адрес], [конечный адрес], [скорость]

Один , два или три параметра могут быть опущены. Первые два параметра в этом случае принимаются равными нулю. Если опущен третий параметр , то принимается его системное значение , установленное в МОНИТОРЕ.

При выводе на МГ содержимого области памяти вычисляется контрольная сумма. Начальный адрес , конечный адрес и контрольная сумма записываются вместе с данными. После окончания вывода на дисплей выводятся начальный, конечный адреса и контрольная сумма.

Пример: 0100,300 ВК

После вывода: 0100.

Запись на МГ содержимого 0300 ячеек памяти с адреса 100 653A по 300.

### 3.3.2. Директива 'I'.

По директиве 'I' с МГ вводятся данные и записываются в область памяти , начальный и конечный адреса которой также вводятся с МГ. Форма директивы :

<I> [смещение],[скорость]

Если задан первый параметр , то данные будут записаны в память , начиная с адреса , равного сумме прочитанного адреса и первого параметра.Если опущен второй параметр , то принимается его системное значение , установленное в МОНИТОРЕ.

При вводе вычисляется контрольная сумма и сравнивается с введенной. По окончании ввода на экран выводятся : начальный адрес, конечный адрес и вычисленная контрольная сумма. Если вычисленная и введенная контрольные суммы не совпадают, то на дисплей выводится введенная контрольная сумма со знаком вопроса - '?'.

Пример: I,40

0100

На экране высвечивается :

0300

начальный и конечный адреса и

6534

вычисленная контрольная сумма.

### 3.4. Пуск резидентных программ.

#### 3.3.1. Директива 'A'.

По директиве 'A' из ПЗУ загружаются в ОЗУ программы РЕДАКТОР и АССЕМБЛЕР , на экран выводится сообщение :

РЕДАКТОР

Форма директивы:

<A>

Специальная форма директивы:

<A>1234

Она используется для запуска технологического теста и пользователь не должен ее применять.

### 3.4.2. Директива 'B'.

По директиве 'B' из ПЗУ загружается в ОЗУ интерпретатор языка Бейсик, на экран выводится сообщение :

BASIC

==>

Форма директивы:

<B>

3.5. Директивы работы с памятью , редактирование ЗУ, регистров.

### 3.5.1. Директива 'D'.

По директиве 'D' на дисплей выводится содержимое заданной области памяти в шестнадцатиричном коде.

Форма директивы:

<D> [начальный адрес] , [конечный адрес]

Один или оба параметра директивы могут быть опущены. В этом случае они принимаются равными нулю.

Пример: D100,200

Эта директива выводит на дисплей содержимое области памяти от 100Н до 200Н в шестнадцатиричном коде.

### 3.5.2. Директива 'L'.

По директиве 'L' на дисплей выводится содержимое области памяти в символьном виде. Неотображаемые символы выводятся в виде точек.

Форма директивы:

<L> [начальный адрес] , [конечный адрес]

Один или оба параметра могут быть опущены. В этом случае они принимаются равными нулю.

Пример: L0,5

По директиве 'C' на экран выводятся 5 символов с указанием начального адреса, например:

0000 A. @MK

### 3.5.3. Директива 'F'.

По директиве 'F' содержимое области памяти, определенное первыми двумя параметрами, заполняется значением 3-го параметра.

Форма директивы:

<F> [начальный адрес], [конечный адрес], [код]

Один, два или три параметра могут быть опущены. В этом случае они принимаются равными нулю.

Пример: F100,200,45

В ячейках памяти от 100 до 200 записан код 45.

### 3.5.4. Директива 'M'.

По директиве M на дисплей выводятся адрес и содержимое ячейки памяти в шестнадцатиричном коде. Можно осуществить просмотр и (или) изменение содержимого ячеек.

Форма директивы:

<M> [адрес]

Нажатием клавиши BK отображается содержимое следующей ячейки.

Изменение ячейки производится после вывода на экран ее содержимого вводом шестнадцатиричного числа, заканчивающегося BK. Просмотр и(или) изменение заканчивается после ввода символа '.'. Параметр может быть опущен. В этом случае он принимается равным нулю.

Пример: M8AC

08AC 6A

В ячейке 08AC произведено изменение

08AD 77

содержимого. Последняя просмотренная

08AE 00 3A

ячейка 08AF.

08AF 44

### 3.5.5. Директива 'T'.

По директиве 'T' производится пересылка содержимого области памяти, определенной 2-мя параметрами в область памяти, начиная с адреса, заданного 3-м параметром.

Форма директивы:

<T> [начальный адрес], [конечный адрес], [адрес пересылки]

Один, два или три параметра могут быть опущены. В этом случае они равны нулю.

Пример: T100,300,1000

Содержимое ячеек памяти, начиная с 100 до 300, пересылается в область памяти, начиная с 1000-й ячейки.

### 3.5.6. Директива 'C'.

По директиве 'C' производится сравнение областей памяти, заданных параметрами.

Форма директивы:

<C> [начальный адрес], [конечный адрес], [адрес области сравнения]

Размер 1-й области сравнения принимается равным размеру области, заданной первыми двумя параметрами. 2-я область сравнения начинается с адреса, определяемого третьим параметром. Если содержимое сравниваемых ячеек не совпадает, то на экран выводятся адрес несовпадающей ячейки из 1-ой области, ее содержимое и соответствующая ячейка из 2-ой области.

Если все сравниваемые ячейки равны, выводим на стрелку. Один, два или три параметра могут быть опущены. В этом случае они принимаются равными нулю.

Пример:

1) правильное сравнение

:-->C200,300,2000

:-->

2) неправильное сравнение

:-->C200,300,2000

023A 4A 7C

02CF 78 B3

02EF 00 7F

:-->

### 3.5.7. Директива 'S'.

По директиве 'S' производится поиск значения, заданного 3-м параметром , в области памяти, определяемой первыми 2-мя параметрами .

Форма директивы:

<S> [начальный адрес],[конечный адрес],[искомый код]

На дисплей выводятся адреса ячеек памяти, содержимое которых равно искомому коду. Один, два или три параметра могут быть опущены. В этом случае они принимаются равными нулю.

Пример: S100,300,45

0220

В этих ячейках записан код 45.

02AC

02FE

:-->

### 3.5.8. Директива 'X'.

По директиве 'X' на дисплей выводится содержимое регистров процессора в шестнадцатиричном коде.

Синтаксис директивы: <Х>

Первым на дисплей выводится содержимое регистра РС. Просмотр регистровых пар HL, DE, BC, SP, AF производится по нажатию клавиши ВК.

Пример: X

РС 013A

При нажатии клавиши ВК поочередно будет выводиться содержимое регистров. Это содержимое можно изменить, записав справа новые значения.

HL

DE

BC

SP

AF

## 3.6. Запуск программ.

### 3.6.1. Директива 'G'.

По директиве 'G' управление передается программе по заданному

адресу.

Форма директивы:

<G> [адрес запуска], [адрес останова]

По достижении адреса , заданного вторым параметром , управление получает МОНИТОР и на дисплей выводится этот адрес. Один или оба параметра могут быть опущены . В этом случае первый параметр принимается равным нулю.

Пример: G -запуск с адреса 0

G100 -запуск с адреса 100

G100,103A -выполнение программы с адреса  
100 по адрес 103A

#### 4. ЗАПРОСЫ МОНИТОРА

Запросы предназначены для использования в программах, требующих действий МОНИТОРА по управлению устройствами машины. Запросы представляют собой команды вызова процедур (подпрограмм) : CALL <адрес> . Запросы могут передавать в МОНИТОР аргументы и (или) возвращать результаты.

Шестнадцатиричный код адреса необходим для работы с АССЕМБЛЕРОМ.

В скобках рядом с шестнадцатиричным кодом указан десятиричный код адреса. Этот код необходим для работы с БЕМСИКОМ.

##### 4.1. Старт МОНИТОРА.

Адрес: 0F800H (2048D)

Входные параметры : нет

Возвращаемые значения : нет

Пример : GF800

ПАРТНЕР 01.01(B.02)

:-->

##### 4.2. Ввод символа с клавиатуры.

Адрес: 0F803H (2048D)

Входные параметры : нет.

Возвращаемые значения :

Регистр А : код нажатой клавиши.

Пример: CALL 0F803H

Если нажата клавиша  , в аккумуляторе код единицы 31.

4.3. Ввод байта с МГ.

Адрес: 0F306H (2042D)

Входные параметры: нет.

Возвращаемые значения :

Регистр А : введенный байт.

4.4. Вывод символа на экран.

Адрес: 0F809H (2039D)

Входные параметры:

Регистр С - код выводимого символа.

Возвращаемые значения : нет.

Пример : MVI C,60H

CALL 0F809H

На экране появится символ Ю.

4.5. Опрос состояния клавиатуры.

Адрес: 0F812H (2030D)

Входные параметры: нет.

Возвращаемые значения :

регистр А = код 00 , если клавиша не нажата

регистр А = код OFFH , если клавиша нажата.

Пример: CALL 0F812H

СPI 0 ; А=0

JZ НЕТ 1 ; А=0      Клавиша не нажата

;                    Клавиша нажата

## 4.6. Запись байта на МГ.

**Адрес:** 0F80CH (2036D)

**Входные параметры:**

регистр С – выводимый байт.

**Возвращаемые значения :** нет.

## 4.7. Вывод байта на экране в шестнадцатиричном коде.

**Адрес:** 0F815H (2027D)

**Входные параметры :**

регистр А – выводимый код.

**Возвращаемые значения :** нет.

**Пример :** MVI A,8FH

CALL 0F815H

На экране появится : 8F

## 4.8. Вывод на экран сообщения.

**Адрес:** 0F818H (2024D)

**Входные параметры :**

регистровая пара (РП) Н – адрес начала текста.

**Возвращаемые значения :** нет.

**Пример :** LXI H,TEXT0

CALL 0F818H

TEXT0: DB "ПРОГРАММА", 0

По адресу , записанному в РП Н по запросу 0F818H выводится сообщение: ПРОГРАММА

## 4.9. Ввод кода нажатой клавиши.

**Адрес:** 0F81BH (2021D)

**Входные параметры :** нет.

**Возвращаемые значения :**

регистр А – код OFFH – клавиша не нажата

регистр А – код OFEH – клавиша ЛАТ

регистр А – код клавиши.

Пример : CALL 0F81EH

если нажата



, в аккумуляторе код : 31.

#### 4.10. Запрос положения курсора.

Адресс: 0F81EH (2018D).

Входные параметры : нет.

Возвращаемые значения :

регистр H – номер строки

регистр L – номер позиции.

Пример : CALL 0F81EH

По этой команде в HL записывается номер строки и столбца.

#### 4.11. Запрос байта из экранного буфера.

Адресс: 0F821H (2015D)

Входные параметры : нет.

Возвращаемые значения :

регистр A – код из буфера.

#### 4.12. Ввод блока с магнитофона.

Адресс: 0F824H (2012D)

Входные параметры :

регистровая пара H – смещение.

Возвращаемые значения :

регистровая пара H – адрес начала

регистровая пара D – адрес конца

регистровая пара B – контрольная сумма.

#### 4.13. Вывод блока на магнитофон.

Адресс: 0F827H (2009D)

Входные параметры :

регистровая пара H – адрес начала

регистровая пара D – адрес конца

регистровая пара B – контрольная сумма.

Возвращаемые значения : нет.

4.14. Подсчет контрольной суммы.

Адрес: OF82AH (2006D)

Входные параметры :

регистровая пара Н - адрес начала

регистровая пара D - адрес конца.

Возвращаемые значения :

регистровая пара В - контрольная сумма.

Пример : LXI H,ADRН

LXI D,ADRCON

CALL OF82AH

В ВС - контрольная сумма.

4.15. Запуск индикации на экране.

Адрес: OF82DH (2003D)

Входные параметры : нет.

Возвращаемые значения : нет.

4.16. Передача адреса верхней границы свободной памяти программе пользователя.

Адрес: OF830H (2000D)

Входные параметры : нет.

Возвращаемые значения :

регистровая пара Н - адрес границы .

4.17. Установка адреса верхней границы свободной памяти программы пользователя.

Адрес: OF833H (1997D)

Входные параметры :

регистровая пара Н - адрес границы.

Возвращаемые значения : нет.

## 5. РАБОТА С ВИДЕОЗУ

### 5.1 Структура видеозу.

Видеозу имеет формат 78\*30. Адрес первой ячейки 76D0H, адрес последней 7FF3H.

Видимая часть видеозу имеет формат 64\*25, т.е. невидимыми остаются 3 верхние строки, 2 нижние, 11 символов справа и 3 символа слева.

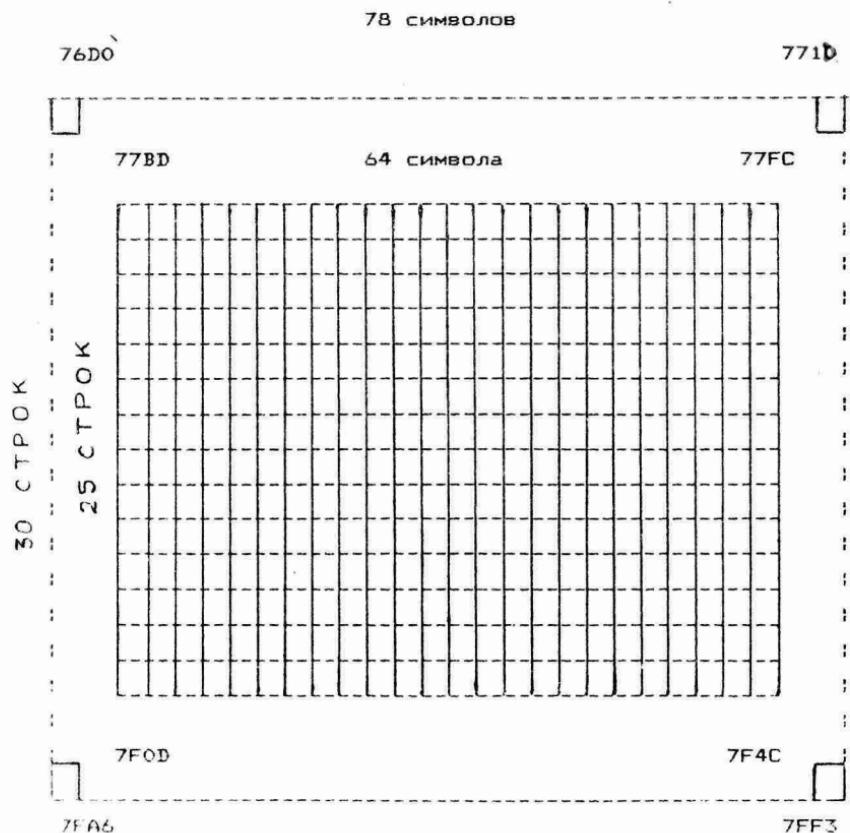


Рис.1.

Начальные и конечные адреса видимой части видеозу см. в  
приложении.

## 5.2 Работа со знакогенератором.

МВЦ "ПАРТНЕР 01.01" дает возможность работать с восемью наборами символов и тремя атрибутами изображения этих символов : подчеркивание, негатив и мерцание. Работа с нужным набором символов и атрибутами , осуществляется через запись в видеоЗУ соответствующего кода, биты управления которого показаны на рис.2. Биты управления П, Н и М , установленные в 1 включают соответствующие атрибуты , а установленные в 0 - выключают . Выбор набора символов осуществляется тремя битами БН1, БН2, БН3 согласно табл. 1.

Действия кода управления распространяется до появления следующего кода ( в границах видеоЗУ ). В видимой части видеоЗУ код отображается пробелом .

Пример :

```
5 PRINT CHR$(31)
10 POKE 30887,142: POKE 30888,77: POKE 30889,69: POKE 30890,140
15 GOTO 10
```

В данном примере оператором POKE 30887,142 в первую ячейку четвертой строки видимой части видеоЗУ записан управляющий код 142 ( побитно 10001110 ) , означающий , что выбран третий набор символов (БН1=1, БН2=1, БН3=0 ) и включен атрибут мерцания (M=1). Затем выведены два символа 77(M) и 69(E), управляемый код, выключающий мерцание 140 ( побитно 10001100 , M=0 ).

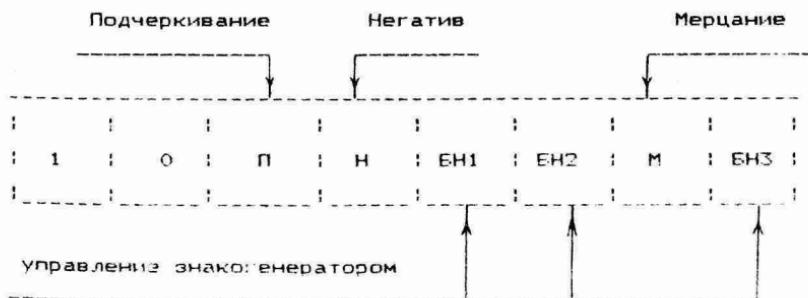


Рис. 2.

Таблица 1

Переключение знакогенератора.

номер набора	БН1			БН2			БН3		
	1	2	3	1	2	3	1	2	3
0	0			0			0		
1	0			1			0		
2	1			0			0		
3	1			1			0		
4	0			0			1		
5	0			1			1		
6	1			0			1		
7	1			1			1		

## 6. ДИРЕКТИВЫ РЕДАКТОРА И АССЕМБЛЕРА.

6.1. РЕДАКТОР предназначен для редактирования любого текста непосредственно на экране дисплея : исправление ошибок , удаление и ввод новых символов и фрагментов текста. РЕДАКТОР записывает готовые части текста на магнитную ленту , и при необходимости компонует из отдельных частей полный текст.

### 6.2. Пуск РЕДАКТОРА.

Запуск РЕДАКТОРА производится директивой 'A' МОНИТОРА, на экране появляется надпись : РЕДАКТОР. Задание дают поочередным нажатием на клавиши AP2 и N (нажатие клавиши AP2 и какой-либо буквы латинского алфавита или символа будем именовать директивой и обозначать алгебраической суммой , например, AP2+N).

По этой директиве очистится экран , и на запрос NEW? можно ответить : Y, в этом случае текстовый буфер в памяти компьютера очистится , и РЕДАКТОР перейдет в режим ввода строки ( в начале первой строки появится символ "►" ), в противном случае установится режим редактирования ( в начале строки - символ "\*" ).

### 6.2.1. Режим ввода строки .

Этот режим используется для ввода текста с клавиатуры компьютера. Стока может состоять не более чем из 63 символов. Набор строки завершает нажатием клавиши ВК , после чего она пересыпается в текстовый буфер. Допущенную при вводе ошибку можно исправить, сместив курсор назад до нужного места. Устранив ошибку, его можно вернуть в любую позицию в строке. После нажатия ВК в текстовый буфер заносится символ , под которым находился курсор, и все символы , находящиеся слева от него.

### 6.2.2. Режим редактирования.

Чтобы перейти из режима ввода строки в режим редактирования необходимо ввести директиву "СТР".

СТР – клавиша



Режим редактирования позволяет просмотреть введенный текст построчно или фрагментами в 24 строки. Каждый фрагмент выводится директивой AP2+↓ . Для просмотра текста в обратном порядке используется директива AP2+↑ . К началу и концу текста можно вернуться соответственно по директивам AP2+В или AP2+Е. Для перемещения курсора к началу первой строки фрагмента пользуются клавишой "↖" , а внутри него – клавишами "←" , "→" , "↑" , "↓" .

Для перехода в режим ввода строки достаточно нажать на клавиши AP2+Е+"↓" .

Чтобы найти какой-либо фрагмент текста , вводят директиву AP2+Л и символы , которые необходимо отыскать , и нажимают "ВК" – на экране появится фрагмент текста, начиная со строки, в которой они впервые встретились.

С помощью директивы AP2+R можно найти все фрагменты с заданными символами. Если в тексте этих символов нет, то звучит звуковой сигнал и восстанавливается режим редактирования. Если символы не сообщены РЕДАКТОРУ , на экране отобразится последняя страница текста.

Символы можно удалить из строки установкой курсора под соответствующим знакоместом и нажатием на клавишу Ф2, а освобождают для пропущенного – клавишей Ф4.

Чтобы вставить новые строки , к началу следующей за ними строки подводят курсор и используют директиву AP2+A. Если их нужно вставить перед первой строкой текста , то сначала нажимают  , а потом AP2+A.

Для удаления фрагмента текста курсор помещают в начало его первой строки и нажимают AP2+D. Затем перемещают курсор до строки, перед которой заканчивается удаляемый фрагмент текста , и вновь вводят директиву AP2+D. Если этот фрагмент решено оставить, вместо вторичного ввода директивы AP2+D нажимают на клавишу "СТР".

#### 6.2.3. Работа с магнитофоном.

Текст из памяти компьютера можно записать на магнитофон с помощью директивы AP2+G. РЕДАКТОР запрашивает имя текста , под которым он будет записан на ленту. Указав имя и включив магнитофон в режим записи ,нажимают "ВК".

Для ввода текста с магнитофона используют директиву AP2+I, и в ответ на запрос РЕДАКТОРа - имя нужного текста.Затем включают магнитофон на воспроизведение и нажимают "ВК".

Для сравнения текста из памяти машины с текстом с ленты используется директива AP2+V. Если тексты не идентичны, на экране появляется сообщение : "ОШИБКА" , а если совпадают – их начальный фрагмент.

РЕДАКТОР позволяет компоновать текст из нескольких фрагментов, которые вводят директивой AP2+M. Любую директиву работы с магнитофоном можно отменить, нажав на клавишу "СТР".

#### 6.3. Назначение АССЕМБЛЕРа для МВЦ "ПАРТНЕР 01.01".

6.3.1. АССЕМБЛЕР - программа - транслятор для разработки программ на языке Ассемблера микропроцессора KP580BM80A. Текст программы с клавиатуры или магнитной ленты с помощью РЕДАКТОРА заносят в область текстового буфера ОЗУ, а затем с помощью АССЕМБЛЕРа транслируют. Результат трансляции - машинные коды располагаются в другой области ОЗУ, называемой областью трансляции. Вход в АССЕМБЛЕР из РЕДАКТОРА производится по нажатию клавиши "СТР".

#### 6.3.2. Требования к программам :

1) Программа на языке ассемблера, предназначенная для трансляции, должна состоять из строк, которые имеют следующий формат записи : "метка : оператор operand ; комментарий ". Пример:

МЕТ1: XRA A ;ОБНУЛЕНИЕ АККУМУЛЯТОРА

2) Метки должны состоять из последовательности латинских или русских букв или цифр и начинаться с буквы или символов. В качестве меток нельзя использовать имена регистров А, В, С, Д, Е, Н.

3) В программах можно использовать псевдооператор ORG, END, DB, DW, DS, EQU. Имена меток перед псевдооператорами должны заканчиваться ":".

4) В качестве operandов можно использовать имена регистров, константы, имена меток и выражения, десятичные шестнадцатиричные числа. После шестнадцатиричного числа ставят латинскую букву Н, а если оно начинается с буквы , впереди добавляют цифру 0. Operandами могут быть выражения с арифметическими знаками "+", "-".

6.3.3. Трансляция начинается при нажатии на одну из следующих клавиш :

1) "1" - трансляция с выводом на экран протокола трансляции, представляющего собой строки исходного текста программы, перед которыми в шестнадцатиричной системе выводятся коды ошибок, адреса размещения команд и данных в ОЗУ и машинные коды транслируемой программы ;

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

3) "3" - текст программы транслируется в машинные коды и выводится сообщение о результатах трансляции : число ошибок и два шестнадцатиричных числа : старший адрес программы в области загрузки и в области трансляции ( ограниченный символами "/" ).

Трансляция программы прерывается одновременным нажатием на клавиши "УС" и "С". Выйти из АССЕМБЛЕРа в МОНИТОР можно при нажатии на клавиши "УС" и "Е". Если при трансляции объем ОЗУ окажется недостаточным, она прекратится и на экране появится

сообщение "МАЛО ПАМЯТИ".

При трансляции АССЕМБЛЕР анализирует синтаксис программы и выводит на печать информацию об ошибках в виде кодов :

- 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.

#### ПРИМЕР ОТТРАНСЛИРОВАННОЙ ПРГРАММЫ :

```
1100 31 00 60 START : LXI SP,6000H
1103 78           MOV A,B
1104 CD 15 F8     CALL 0F815H ; вывод кода
```

Директивы РЕДАКТОРа и АССЕМБЛЕРа приведены в приложениях 2 и 3.

### 7. ИНТЕРПРЕТАТОР ЯЗЫКА БЕЙСИК.

7.1. Программа на языке Бейсик состоит из строк , которые могут содержать операторы и команды. Программа может вводиться и выполняться в одном из двух режимов : косвенном и непосредственном.

#### 7.2. Косвенный режим.

Чаще всего программа на языке Бейсик записывается в косвенном режиме , где каждая строка начинается с номера строки. Номер строки включает от одной до пяти цифр в диапазоне от 1 до 65535. Номер строки выполняет две функции. Во-первых, он служит меткой оператора и может быть использован для ссылки на данный оператор, во-вторых, номер строки определяет порядок выполнения операторов.

При составлении программы целесообразно нумеровать строки номерами кратными 5 или 10, чтобы иметь возможность вставлять при отладке программы забытые или дополнительные строки.

За номером строки следует оператор или группа операторов .

Завершается строка нажатием клавиши "VK" (возврат каретки).  
Операторы в одной строке разделяются символом ":".

Например:

10 INPUT A,B,C <VK> - строка с одним оператором

20 X = 5 : PRINT X,Y,Z : IF X = A THEN 10 <VK> - строка с  
несколькоими операторами.

При передаче управления на строку, содержащую несколько  
операторов, управление передается первому оператору строки.

### 7.3. Непосредственный режим.

Интерпретатор языка Бейсик отличает строки, вводимые для последующего выполнения, от строк, вводимых для непосредственного выполнения в зависимости от наличия номера строки. Операторы, которые начинаются с номера строки, запоминаются; операторы без номера строки выполняются по мере их ввода в систему.

Такая особенность языка Бейсик позволяет использовать МЭЦ как калькулятор.

Например: PRINT 2+(89\*2/100)+SIN(23) <VK>

На экране будет выведено : 2.9337

Операторы Бейсика могут быть 2-х типов:

- 1) команды, начинающиеся с ключевого слова;
- 2) команды сравнения, начинающиеся с имени переменной.

### 7.4. Элементы языка.

Данными могут быть как константы, так и переменные.

Форма записи чисел близка к естественной. Однако, вместо запятой используется точка, например:

123.456

0.053

Для записи очень больших и очень маленьких чисел разрешается использование экспоненциальной формы. При десятичном порядке числа, стоящее после буквы Е, может быть только целым, например:

1.234E-5 = 0,00001234

-1.234E5 = 123400

Диапазон вещественных констант: от -10 до 10

Шестнадцатиричное число определяется оператором "&", например:

A = &FFFF

Для перевода десятичного числа в шестнадцатиричное предусмотрен оператор "@", который используется совместно с оператором "PRINT" и позволяет выводить на экран число или результат вычисления в шестнадцатиричном виде, при условии, что это число не превышает 65535 (FFFF). Если число дробное, то в шестнадцатиричном виде будет представлена только его целая часть, например:

PRINT @10,@(4000+96) <ВК>

0A.....1000

Все числа, начиная от 8000H до FFFFH, представляются отрицательными и представляют собой разность между представлением этого числа в десятичном виде и числом 65536, например:

PRINT &7FFF,&8000,&FFFF <ВК>

32767.....-32768.....-1

Строковые константы имеют длину не более 80 символов и могут содержать любые символы, кроме возврата каретки и кавычек. Присваивать строковые константы можно только строковым

переменным. Переменными называют величины, значения которых в процессе выполнения программы могут меняться. Переменные также могут быть вещественными и строковыми. Переменные представлены в таблице:

Тип			Имя
Простые	Массивы		
Вещественные			А
Строчные			АХ
	Вещественные		А(3,12,2)
	Строчные		АХ(3,12,2)

Имя переменной может состоять не более чем из двух символов. Первым всегда латинская буква, вторым может быть как буква, так и цифра.

Размер массива ограничивается памятью. Идентификация переменных массивов проводится по всем символам имени и по типу переменной.

#### 7.4.2. Представление операций.

При вычислении выражений интерпретатор Бейсика выполняет арифметические операции согласно приоритетам:

1 - возведение в степень

2 - умножение, деление

3 - сложение, вычитание

С помощью скобок можно изменить последовательность выполнения арифметических операций. В первую очередь вычисляются части выражений, заключенные в круглые скобки. Вычисление групп

с вложенными скобками начинается с внутренних скобок и идет в направлении к внешним скобкам.

Например:

$$A * [(B^2 * C) / 2]$$

вычисляется в следующем порядке:

$$B^2, B^2 * C, [(B^2 * C) / 2], [(B^2 * C) / 2] * A$$

Обозначение арифметических операций:

" $^2$ " – возведение в степень

"\*" – умножение

"/" – деление

"+" – сложение

"–" – вычитание

Обозначение операций отношения:

">" – больше

"<" – меньше

"=" – равно

"<>" "><" – не равно

"=>" ">=" – больше или равно

"=<" "<=" – меньше или равно

Обозначение логических операций:

"NOT" – логическое "НЕ"

"AND" – логическое "И"

"OR" – логическое "ИЛИ"

Логическое "истина" тождественно арифметической единице,

31  
1700045-013401

"ложь" - нули.

Для строковых констант применимы отношения "=" и "<>".

#### 7.4.3. Директивы Бейсика.

CLOAD - загрузить программу с магнитофона (МГ). Например:

CLOAD"ПРОГР1"<ВК> - загрузка с МГ программы с именем ПРОГР1 .

CLOAD" " - загрузка с МГ первой встретившейся программы.

Для считывания программы, записанной на МВЦ с версией монитора младше ОЗ (01.02) необходимо набрать CLOAD и нажать <ВК> при появлении второго тонального сигнала.

CSAVE - записать программу на МГ. Например:

CSAVE"ПРОГР1" - запись программы с именем ПРОГР1.

CSAVE" " - запись программы без имени.

NEW - устанавливает начальное состояние памяти, стирает программы.

Директиву NEW необходимо набрать сразу после запуска Бейсика.

RUN - запуск программы со строки с наименьшим номером, для запуска программы с промежуточной точки нужно указать номер строки, например:

RUN100

STOP - останавливает выполнение программы с выдачей номера строки и номера оператора, где произошел останов.

CONT - продолжает выполнение программы, остановленной оператором STOP или клавишей F4.

MERGE - соединить программу в ОЗУ с программой на МГ. При этом необходимо иметь в виду, что номера строк подгружаемой

программы должны быть больше максимального номера строки программы, находящейся в ОЗУ.

VERIFY - проверить правильность записи программы на МГ. При проверке происходит сравнение программы, находящейся в ОЗУ компьютера, с записью на МГ.

#### 7.4.4. Редактирование, текстовый ввод.

При вводе программы с клавиатуры вводимую информацию можно сразу редактировать, вставляя пропущенные или удаляя ошибочно введенные символы. Для вставки курсор подводят к символу, перед которым предполагается вставка, и вводят пропущенный символ. Удаление символа происходит при нажатии клавиши "F2" после установки курсора под этим символом. Ввод заканчивается нажатием клавиши "ВК". Нажатие клавиши "F4" приводит к отмене вводимой информации.

Максимальная длина строки 128 символов.

Для оперативного набора программ в Бейсике предусмотрена возможность ввода служебных слов последовательным нажатием клавиши "AP2" и клавиши, соответствующей этому служебному слову. Таблица соответствия символьных клавиш служебным словам приведена в приложении 6.

При вводе и редактировании текста используются следующие операторы:

AUTO - автоматическая нумерация строк с заданием начального номера строки и приращения, например:

AUTO 140,5

По умолчанию каждый из параметров равен 10. Для выхода из режима AUTO нужно нажать клавишу "F4" (УС + С).

RENUM - изменение номеров строк с заданием начального номера и приращения. По умолчанию каждый из параметров равен 10.

Пример:

RENUM 5,3

Этот оператор устанавливает номер первой строки - 5, второй - 8 и т. д.

LIST - вывести часть или всю программу на экран дисплея, например:

LIST - вывод всего текста программы;

LIST N1,N2 - вывод части текста программы от строки N1 до строки N2 включительно;

LIST N1, - вывод текста программы от строки с номером N1 до конца программы;

LIST,N2 - вывод текста программы с начала до N2.

EDIT - вывод строки для редактирования, например:

EDIT N -редактирование строки N

Редактирование производится также, как и при вводе строки. Выход из режима происходит при нажатии клавиши "ВК". Клавиша "F4" отменяет режим редактирования и в программе остается старая строка.

DELETE - удалить строки из программы, например:

DELETE N1,N2 - удаляет фрагмент программы с номерами строк от N1 до N2 включительно.

REM - определить комментарий. При выполнении программы игнорируется.

PAUSE - останавливает выполнение программы на заданное время (в секундах), например:

PAUSE T

Диапазон возможных изменений Т от 0,0015 до 65 с. При Т=0 интерпретатор остановит выполнение программы и возобновит его только после нажатия любой клавиши.

#### 7.4.5. Ввод/вывод данных. Массивы.

В Бейсике могут использоваться простые переменные и массивы. Массивы при этом должны быть описаны. Если размерность массива не указана, то ему по умолчанию присваивается размерность 10.

DIM <список массивов> – описывает размерности массивов.

Например:

40 DIM X(8),T\$ (2,7)

DIM описывает одномерный массив X из 8 элементов и двухмерный массив T\$, состоящий из двух строк и семи столбцов. Массив T\$ – символьный.

Ввод данных с клавиатуры осуществляется оператором "INPUT", например:

10 INPUT "ЖДУ ВВОДА";A,B,C...

При выполнении этого оператора на экран будут выведены сообщение "ЖДУ ВВОДА" и знак вопроса. Компьютер будет ожидать ввода данных. После завершения ввода переменным будут присвоены соответствующие значения. Сообщение в кавычках при вводе может отсутствовать, а переменные могут быть как числовыми, так и символьными. Вывод знака вопроса можно отменить, для этого в операторе "INPUT" после сообщения в кавычках надо поставить ",", вместо ";".

Если все переменные числовые, то вводимые данные могут быть как числами, так и результатами арифметических выражений.

Пример:

10 INPUT "ВВЕДИТЕ ЧИСЛО:",X:PRINT X: GOTO 10

RUN

ВВЕДИТЕ ЧИСЛО:2\*2 <VK>

4

ВВЕДИТЕ ЧИСЛО:&A <VK>

10

ВВЕДИТЕ ЧИСЛО:<F4>

&gt;

При не полностью введенных данных выход из режима ввода по нажатию клавиши "BK" блокирован. В этом случае при нажатии "BK" выводится вопросительный знак и ожидается ввод недостающих данных. Однако, если нажать клавишу "F4" (УС+С), то интерпретатор перейдет в непосредственный режим. Лишние введенные данные интерпретатором игнорируются.

**DATA** - задает значения, которые будут считаны оператором **READ**.

**READ** - присваивает переменным значения из списка при операторе **DATA**.

**RESTORE** - устанавливает указатель для чтения из оператора **DATA**.

Считывание данных с помощью оператора "READ" может осуществляться в любой последовательности, для чего в операторе "RESTORE" предусмотрен параметр, определяющий номер строки, на которую настраивается указатель при работе оператора "READ". При отсутствии у оператора "RESTORE" параметра указатель настраивается на первый блок данных, определенных оператором "DATA".

Пример:

```
10 RESTORE 80
20 READ A,B,C
30 PRINT A,B,C
40 RESTORE 70
50 READ A,B,C
60 PRINT A,B,C
70 DATA 10,20,30
80 DATA 1,2,3
```

После запуска этой программы на экран будут выведены числа 1,2,3, а на следующей строке 10,20,30.

**PRINT** - вывод на дисплей.

**AT** - используется при **PRINT** для вывода в заданную позицию

экрана, например:

PRINT AT 5,5; "A"

TAB - используется для пропуска заданного числа позиций, например:

PRINT TAB(X); "A"

X - число позиций.

@ - используется при PRINT для вывода чисел в шестнадцатиричном виде, например:

PRINT @255

SPC - используется для печати заданного числа пробелов, например:

PRINT SPC(X); "A"

CUR - используется для задания координат при выводе на дисплей, например:

CUR X,Y: PRINT "A"

#### 7.4.6. Команды передачи управления.

GOTO - переход к строке с заданным номером, например:

GOTO 200

GOSUB - выполнить подпрограмму, например:

GOSUB 300

RETURN - конец подпрограммы.

IF...THEN - выполнить оператор (операторы) после THEN, если выражение истинно. Иначе перейти к следующей строке.

ON - выполнить подпрограмму или переход по результату выражения.

Пример:

10 INPUT I,J

```
20 ON I GOSUB 40,50,60
30 IF I=0 THEN PRINT "КОНЕЦ": STOP
35 GOTO 10
40 PRINT J*j:RETURN
50 PRINT J*j*j:RETURN
60 PRINT J*j*j*j:RETURN
```

Если I=1, то вызывается подпрограмма с номером 40, если I=2, то с номером 50 и т.д. Если задать I=0, происходит останов в строке 30.

FOR...TO...STEP...NEXT – создание цикла.

FOR I=N TO S – оператор цикла. Выполняются все операторы между FOR и соответствующим NEXT S раз.

STEP X – элемент оператора цикла, задает шаг цикла, если X=1, то его применение не обязательно.

NEXT – оператор конца цикла.

Пример:

```
10 FOR I=0 TO 100 STEP 5
20 PRINT I
30 NEXT I
```

В данном примере будут выводиться числа от 0 до 100 с шагом 5: 0,5,10,15...100.

#### 7.4.7. Псевдографика.

CLS – очищает экран дисплея и сохраняет курсор в месте очистки.

HOME – очищает экран дисплея и устанавливает курсор в левый верхний угол экрана.

PLOT – высвечивает (гасит) точку в заданной позиции,  
например:

PLOT X,Y,Z

$Z = 0$  – погасить точку,  $Z = 1$  – высветить.

LINE – проводит линию до указанной позиции, например:  
LINE X,Y

Чтобы избежать ошибок при построении псевдографических изображений, следует выбирать X и Y исходя из формата экрана.  
 $1 \leq X \leq 25$ ,  $1 \leq Y \leq 64$ .

#### 7.4.8. Работа с памятью.

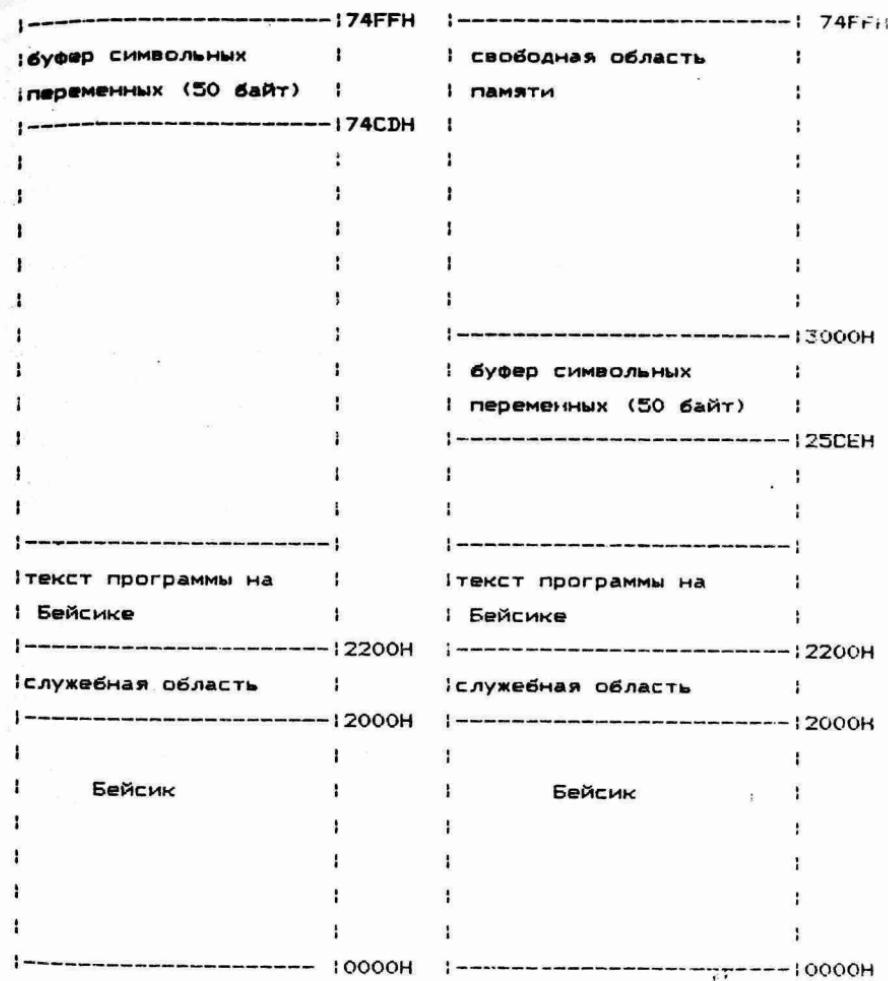
CLEAR – очистить переменные с заданием (или без) размера буфера символьных переменных, например:

CLEAR 1000

Оператор очищает переменные и задает размер буфера символьных переменных 1000 байт.

HIMEM – установить верхнюю границу, используемой Бейсиком оперативной памяти. Границу можно установить в пределах от 2400H до стека. На рисунке показано два распределения памяти:  
а – после запуска Бейсика;  
б – после выполнения директивы HIMEM &3000

<sup>39</sup>  
1100045-01 3401



а

б

Рис.

FRE - число байт свободного ОЗУ, например:  
PRINT FRE(X)

40  
1700045-01 34 01

21169

т.е. 21169 байт свободно.

POKE - записать байт в указанное место в ОЗУ.

PEEK - значение байта из ОЗУ по заданному адресу.

Пример:

10 POKE 21024,83

20 S=PEEK(21024)

30 PRINT S

Оператор POKE помещает значение 83 в ячейку ОЗУ с адресом 21024. Затем переменной S присваивается значение функции PEEK и оператором PRINT на экран будет выведено значение 83.

USR - обратиться к подпрограмме в машинных кодах по заданному адресу, например:

A = USR(&4000)

ADDR - адрес переменной. Этот оператор возвращает число, которое является адресом ячейки памяти, начиная с которой хранится переменная. Переменная может быть как числовой, так и символьной. Под числовые переменные в ОЗУ отводится 6 байт. Значения переменных хранятся в логарифмическом виде (по основанию 2), последний старший байт - характеристика, а три младших - мантисса переменной. Старший бит третьего байта знаковый. Для отрицательных чисел он равен единице.

Пример:

10 INPUT "ЧИСЛО"; X:PRINT "ЕГО ПРЕДСТАВЛЕНИЕ:";

20 PRINT @PEEK(ADDR(X)+3);:REM ВЫВОД ЧЕТВЕРТОГО БАЙТА (СТАРШЕГО)

30 PRINT @PEEK(ADDR(X)+2);:REM ВЫВОД ТРЕТЬЕГО БАЙТА

40 PRINT @PEEK(ADDR(X)+1);:REM ВЫВОД ВТОРОГО БАЙТА

50 PRINT @PEEK(ADDR(X));:REM ВЫВОД ПЕРВОГО БАЙТА (МЛАДШЕГО)

60 GOTO 10

После запуска этой программы и ввода числа на экран будет выводится его представление в виде четырех шестнадцатиричных чисел:

ЧИСЛО ? 1

ЕГО ПРЕДСТАВЛЕНИЕ: 81 00 00 00

ЧИСЛО ? -1

ЕГО ПРЕДСТАВЛЕНИЕ: 81 80 00 00

ЧИСЛО ? <F4>

=>

Символьные переменные хранятся следующим образом: в первой ячейке памяти хранится длина символьной переменной, вторая не используется, а в третьей и четвертой размещены соответственно младший и старший байты адреса начала последовательности символов этой переменной в буфере символьных переменных.

Необходимо помнить, что размер буфера символьных переменных должен быть, как минимум, в два раза больше общего числа символов всех символьных переменных.

#### 7.4.9. Функции Бейсика.

DEF FNX (параметр) выражение – оператор для определения функции, заданной пользователем. Здесь X – имя функции, в круглых скобках задается формальный параметр, выражение – любое допустимое в Бейсике арифметическое выражение, например:

10 DEF FNX (S) = S+S^2

20 P = FNX (2)

30 C = FNX (4)

40 PRINT "P=";P;"C=";C

На экран будет выведен следующий результат:

P=6 C=20

42  
1700045-013401

Для обозначения стандартных функций используются двухбуквенные наименования.

Аргументами математических функций могут быть константы, переменные и выражения, заключенные в круглые скобки. Аргументы тригонометрических функций должны задаваться в радианах.

SIN(X) - значение синуса

COS(X) - значение косинуса

TAN(X) - значение тангенса

ASN(X) - значение арксинуса

ACS(X) - значение арккосинуса

ATN(X) - значение арктангенса

ABS(X) - модуль X

EXP(X) - экспонента X

INT(X) - значение целой части X

RND(X) - генерация случайного числа, при X>0 значение функции является элементом случайной последовательности, при X=0 повторяется последний результат.

SGN(X) - знак аргумента: -1 при X<0; 0 при X=0; 1 при X>0.

SQR(X) - извлечение квадратного корня из X.

LOG(X) - натуральный логарифм X.

LB(X) - десятичный логарифм X.

#### Встроенные функции.

Для работы с символьными данными в Бейсике имеются следующие встроенные функции.

ASC(X) - значение кода символа, например:

X = ASC("E")

В результате X = 69

CHR\$(X) - символ, соответствующий значению аргумента, например:

AN = CHR\$(69)

В результате  $A\alpha = E$

INKEY $\alpha$  - символ нажатой клавиши (программа не останавливается).

пример:

```
10 PRINT INKEY $\alpha$ ;: BEEP 0.2,0
```

```
20 GOTO 10
```

При работе этой программы на экран выводятся символы, соответствующие нажимаемым клавишам и подается звуковой сигнал. Если клавиша не нажата выдается только звуковой сигнал.

LEFT $\alpha$  ( $A\alpha, X$ ) - заданное число символов с начала (левой части) символьной переменной.

MID $\alpha$  ( $A\alpha, X, Y$ ) - заданное число символов -  $Y$  символьной переменной, начиная с  $X$ .

RIGHT $\alpha$  ( $A\alpha, X$ ) - заданное число символов с конца (правой части) символьной переменной.

Пример:

```
10  $A\alpha = "МВЦ ПАРТНЕР"$ 
```

```
20 PRINT LEFT $\alpha(A\alpha, 3)$ 
```

```
30 PRINT RIGHT $\alpha(A\alpha, 7)$ 
```

```
40 PRINT MID $\alpha(A\alpha, 3, 3)$ 
```

После выполнения этой программы на экране появится:

МВЦ

ПАРТНЕР

Ц П

LEN ( $A\alpha$ ) - длина символьной переменной, например:

```
PRINT LEN("ПАРТНЕР")
```

В результате на экране появится:

6

POS( $X$ ) - номер позиции последнего выведенного символа на

дисплей.

SCREEN $\alpha$ (X,Y) - символ из заданной позиции экрана дисплея.

STR $\alpha$ (X) - преобразование числа в строку символов.

VAL(A $\alpha$ ) - преобразование строки символов в число.

#### Управляющие и алфавитно-цифровые коды дисплея.

Управляющие коды дисплея управляют отображением информации на экране. Их можно выдавать как с клавиатуры, так и используя подпрограмму Монитора "вывод символа". Управляющие коды приведены в таблице 2.

Алфавитно-цифровые коды дисплея соответствуют кодам КОИ-7, и приведены в таблице 3.

Таблица 2

код	обозначение	функция
07	-	Звуковой сигнал
08	<-	Перемещение курсора влево на 1 позицию
0A	ПС	Перевод строки
0C	\	Установка курсора в левый верхний угол
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	Функция прямой адресации курсора
1F	СTP	Чистка экрана с установкой курсора в левый верхний угол экрана

код	символ	код	символ	код	символ
20	пробел	40	@	60	ю
21	!	41	А	61	А
22	"	42	В	62	Б
23	#	43	С	63	С
24	¤	44	Д	64	д
25	%	45	Е	65	е
26	&	46	Ф	66	ф
27	,	47	Г	67	г
28	(	48	Н	68	х
29	)	49	И	69	и
2A	*	4A	Ј	6A	и
2B	+	4B	К	6B	к
2C	,	4C	Л	6C	л
2D	-	4D	М	6D	м
2E	.	4E	Н	6E	н
2F	/	4F	О	6F	о
30	0	50	Р	70	п
31	1	51	Q	71	я
32	2	52	Р	72	р
33	3	53	С	73	с
34	4	54	Т	74	т
35	5	55	У	75	у
36	6	56	В	76	ж
37	7	57	W	77	в
38	8	58	Х	78	ъ
39	9	59	Y	79	ы
3A	:	5A	Z	7A	з
3B	;	5B	Ѐ	7B	ш
3C	<	5C	\	7C	э
3D	=	5D	Ѡ	7D	щ
3E	>	5E	^	7E	ч
3F	?	5F	-	7F	зб

47  
1700045-01 34 01

Псевдографические символы.

Псевдографические изображения в ПЭВМ формируются из 16 различных символов. Эти символы и их шестнадцатиричные коды приведены в таблице 4.

Таблица 4

КОД	СИМВОЛ	КОД	СИМВОЛ
0		10	
1		11	
2		12	
3		13	
4		14	
5		15	
6		16	
7		17	

## 8. ОПИСАНИЕ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ, ПОСТАВЛЯЕМОГО В КОМПЛЕКТЕ С МВЦ

### 8.1. Состав программного обеспечения.

В комплект МВЦ входит следующее программное обеспечение:

- 1) игровая программа "Цирк";
- 2) игровая программа "Тетрис";
- 3) игровая программа "Шахматы";
- 4) игровая программа "Рикошет";
- 5) игровая программа "Вулкан".

### 8.2. Запуск любой игровой программы производится следующим образом:

- 1) найти на магнитной ленте начало программы;
- 2). нажать клавишу сброса "С";
- 3) нажать клавишу "R". В ответ на запрос ИМЯ ФАЙЛА[Р]: набрать имя программы и ключ контроля - /Р. Нажать клавишу "ВК";
- 4) включить магнитофон на воспроизведение;
- 5) по окончании считывания магнитофон выключить;
- 6) выполнить инструкцию к игровой программе.

### 8.3. Игровая программа "Цирк" (Имя -CIRK).

После запуска программы на дисплее появляется заставка "Цирк", надпись рядом с курсором предлагает ввести одну из команд:  
С - изменение скорости движения человечка;  
З - запуск игры.

После запуска игры на экране появляется новая картинка: сверху слева на помосте - человечек, под ним - батут. Движение батута управляется клавишами: "-->" (вправо), "<--" (влево), любой

другой (останов). Задача : двигать батут так, чтобы человечек, отскакивая от него, сбил все висящие знаки "&" в потолке, не упав на пол. Движение человечка начинается нажатием клавиши "-->" до писка биппера. Имеются пять попыток, использовав которые, игра останавливается и для ее продолжения нужно начать сначала.

#### 8.4. Игровая программа "Тетрис" (Имя - TETR).

После запуска программы на экране высвечивается картинка, где справа расположен СТАКАН, а слева - пояснительная информация. Игра начинается ответом на запрос "ваш класс 1-8", при котором надо нажать нужную клавишу от "1" до "8". После этого в стакан начинают падать различные фигуры.

Смысл игры в том, чтобы, управляя фигурами с помощью указанных клавиш, поместить их как можно больше в стакан.

Полностью заполненный ряд из фигур исчезает, освобождая место. Фигуры можно "уронить" на дно стакана с большей скоростью, используя клавишу "F".

В процессе игры набираются очки за каждую опущенную в стакан фигуру, фиксируется лучший результат.

Останов игры производится нажатием клавиши "Пробел".

#### 8.5. Игровая программа "Шахматы" (Имя - CHAX).

Предлагается программа игры в шахматы. Программа имеет два уровня сложности : '0' и '1'.

В зависимости от выбранного уровня машина затрачивает на поиск ответного хода 1,5 мин. (при уровне сложности '0') или 10 мин. (при сложности '1').

Ходы задаются указанием начального и конечного поля хода, например : E2-E4, взятие : 7:5 , короткая рокировка задается : 0-0, длинная: 0-00. В ходе игры можно предложить машине ничью ; для начала новой партии - следует ввести : NEW.

Программа не предназначена для решения задач и анализа окончания партии, поэтому возможны неточности в определении ситуаций пата и мата, что можно считать ее недостатком. Однако это компенсируется более сильной игрой в начале и середине партии (по сравнению с CHESS - 03, версия ЕК-0010).

1П.00045-01 34 01

8.6. Игровая программа "Рикошет" (Имя - RIKT).

После запуска программы появляется заставка "Рикошет".

После вторичного нажатия на клавишу "ВК" появляется картинка, где вверху расположены цели в виде знака "Х", внизу заряды - 1, справа пять пушек. Игра начинается нажатием клавиши "<-->". При этом пушка перемещается на игровую площадку и начинает двигаться. Передвижение пушки вправо и влево осуществляется клавишами "-->", "<--". Стрельба из пушки производится нажатием клавиши "Пробел", останов пушки - любой другой клавишей.

Игра заключается в том, чтобы выстрелом из пушки сбить все цели "Х", которые охраняются движущимися площадками. При попадании в эту площадку выстрел рикошетом бьет вниз по пушке и, если ее вовремя не убрать из-под выстрела, она будет уничтожена.

Делается пять попыток - по количеству пушек. В процессе стрельбы количество зарядов в ряду уменьшается. Надо сбить все цели имеющимися зарядами. По ходу игры фиксируется время, очки, заряды.

8.7. Игровая программа "Вулкан" (Имя - WULK).

Смысл игры в том, чтобы провести вертолет через извержение вулкана и переправить людей на площадку вертолета, управляя запрограммированными клавишами.

После запуска программы появляется заставка с надписью ВУЛКАН, таблица команд управления - "меню" - в левом нижнем углу заставки, таблица состояний - в правом нижнем углу. Курсор расположен после запроса "введите директиву". По этому запросу вводят нужную директиву из "Меню":

1) директива 'K' - "Программирование клавиш". После нажатия на клавишу "K" курсор перемещается в таблицу состояний и указывает на слово "влево". Нажатие после этого какой-либо клавиши будет означать, что эта клавиша управляет движением вертолета влево. Последующее нажатие остальных трех любых клавиш будет означать соответственно движение вертолета вправо, вверх и вниз. После этого курсор перемещается на прежнее место;

2) директива 'S' - "Звук".

Предназначена для выбора игрового режима со звуком или без него. При нажатии на клавишу "S" в таблице состояний появляется

звук - ВКЛ, при следующем нажатии : звук - ВЫКЛ ;

3) директива 'E' - "Выход".

При нажатии на клавишу "E" осуществляется выход в монитор. На экране появляется сообщение: Партнер 01.01(В.02)

:-->

4) директива 'P' - "Игра".

Нажатием на клавишу "P" выходим в игровой режим.

При этом на экране появляется вулкан, на левом склоне которого расположены люди в виде знаков "I", "Y" и т.д., а на правом склоне - площадка с тремя вертолетами. Лава вулкана изображается символами "\*".

При соприкосновении с символом "\*" вертолет взрывается и вылетает следующий. "\*" можно уничтожить выстрелами, которые производятся нажатием любой клавиши.

Взятие людей на борт вертолета производится путем подлета к ним сверху и касания корпусом вертолета. На борт берется только один человек, который может продержаться не больше определенного времени.

Примечание. Инструкция к игровой программе появляется на экране телевизора не сразу после считывания, а после задержки в 5-30 секунд, во время которой подсчитывается контрольная сумма программы и которая по времени зависит от объема прочитанной программы.

## 9. ДОПОЛНИТЕЛЬНЫЕ СРЕДСТВА МОНИТОРА ДЛЯ РАБОТЫ С МАГНИТОФОНОМ

В Монитор дополнительно введен способ записи файлов на магнитную ленту методом частотной модуляции.

Данный метод использует скорость записи 1200 бод. Логические уровни записываются в следующем виде: логический нуль – один период частоты 1200 Гц, логическая единица – два периода частоты 2400 Гц.

Каждый файл на ленте состоит из заголовка и байтов данных. Различают два типа заголовков: длинный, используемый как заголовок файла, и короткий, применяемый как разделитель внутри тела файла. По заголовкам драйвер МГ производит настройку временных констант для чтения.

Длинный заголовок 16000 \* 2400 Гц.

Короткий заголовок 4000 \* 2400 Гц.

Каждый байт данных состоит из одного '0' (стартовый бит), за которым следует 8-битный поток данных, после которого следуют два бита '1' (стоповые биты). В байте первым передается наименее значащий бит (0-й), а последним наиболее значащий бит (7-й).

В процессе чтения кассеты программное обеспечение считает число переходов сигналов через 0 за время, равное 3/4 номинальной длительности бита, и предполагает бит = 0, если обнаружен один переход, или = 1, если число переходов равно двум или трем.

Логический формат файлов на ленте зависит от назначения файла. В системе возможны три типа файлов, описываемых заголовком файла: текстовые файлы для интерпретатора Бейсика, текстовые файлы в формате КОИ-7 и файлы машинных кодов (двоичные файлы).

### 9.1. Форматы файлов.

#### 9.1.1. Текстовые файлы Бейсика.

[Заголовок файла]

Длинный заголовок

10 байтов ОДЗН

Имя файла (6 символов КОИ-7)

[Тело файла]

Короткий заголовок

Запись текста программы въ внутреннем

представлении интерпретатора

7 байтов ООН

9.1.2. Текстовые файлы КОИ-7.

[Заголовок файла]

Длинный заголовок

10 байтов ОЕАН

Имя файла (6 символов КОИ-7)

[Тело файла]

Короткий заголовок

256 байтов данных

Короткий заголовок

256 байтов данных

.....

Короткий заголовок

256 байтов данных

среди которых имеется маркер конца (код СУ/2 = 1АН)

9.1.3. Двоичный файл.

[Заголовок файла]

Длинный заголовок

10 байтов ОДОН

Имя файла (6 символов КОИ-7)

[Тело файла]

Короткий заголовок

Адрес начала загрузки (2 байта)

Адрес конца загрузки (2 байта)

Стартовый адрес программы (2 байта)

Коды программы

9.2.Запросы Монитора.

9.2.1.Чтение заголовка файла.

Адрес: 0F839H

Входные параметры:

Регистровая пара Н – адрес буфера имени

Возвращаемые значения:

Регистр А – тип файла

Буфер – имя файла

9.2.2.Чтение короткого заголовка.

Адрес: 0F83FH

Входные параметры: нет

Возвращаемые значения: нет

9.2.3.Чтение блока.

Адрес: 0F845H

Входные параметры:

Регистровая пара Н – начальный адрес области памяти

Регистровая пара Д – конечный адрес блока

Возвращаемые значения:

Прочитанный блок данных в указанной области памяти

9.2.4.Чтение байта.

Адрес: 0F84BH

Входные параметры: нет

Возвращаемые значения:

Регистр А – прочитанный байт

9.2.5.Запись заголовка файла.

Адрес: 0F83CH

Входные параметры:

Регистровая пара Н – адрес буфера имени

Регистр С – тип файла

Возвращаемые значения: нет

## 9.2.6. Запись короткого заголовка.

Адрес: 0F842H

Входные параметры: нет

Возвращаемые значения: нет

## 9.2.7. Запись блока.

Адрес: 0F848H

Входные параметры:

Регистровая пара Н - начальный адрес области памяти

Регистровая пара Д - конечный адрес области памяти

Возвращаемые значения: нет

## 9.2.8. Запись байта.

Адрес: 0F84EH

Входные параметры:

Регистр С - байт

Возвращаемые значения: нет

## 9.3. Директивы Монитора.

## 9.3.1. Директива 'W'

По директиве 'W' на МГ выводится содержимое области памяти в формате двоичного файла.

Форма директивы:

**W [начальный адрес], [конечный адрес], [стартовый адрес]**

Первый и третий параметры могут быть опущены. В этом случае первый параметр принимается равными нулю, а третий - FFFF.

После ввода этой директивы на экране появляется вопрос:

**ИМЯ ФАЙЛА [/P]:**

Оператор должен ввести имя файла. Имя файла должно содержать не больше 6 символов. Если символов больше, то воспринимается

только первые 6 символов. Имя файла может отсутствовать вообще. 'Р' - ключ контроля. Он вводится после имени файла, после символа '/'. При наличии ключа выполняется подсчет контрольной суммы, которая анализируется при считывании файла с МГ в память машины.

Запись файла на ленту производится следующим образом:

- 1) нажать на магнитофоне клавишу - "Пауза", затем одновременно клавиши - "Запись" и "Воспроизведение".
- 2) ввести директиву 'W', указать начальный, конечный и стартовый адреса. Нажать "ВК".
- 3) в ответ на запрос указать имя файла.
- 4) отжать паузу, нажать клавишу "ВК".

Появляется тональный сигнал, на экране ТВ изображение пропадает на время записи длинного заголовка и имени файла. Затем изображение восстанавливается и выводится имя файла. Следует пауза, во время которой подсчитывается контрольная сумма файла и которая по времени впрямую зависит от объема записываемого файла. Она может длиться 5-30 секунд. Затем изображение опять пропадает и производится запись тела файла.

- 5) по окончании записи нажать на магнитофоне клавишу "Останов". На экран будет выведено имя записанного файла.

Пример:

- 1) W100,300,1D0  
ИМЯ ФАЙЛА[ /Р]: ABCDEFJ/P

Имя принимается равным ABCDEF, последний символ 'J' игнорируется

После вывода:

ABCDEF

:-->

- 2) W100,300,1D0

ИМЯ ФАЙЛА[ /Р]: ABC

После вывода:

ABC

:-->

- 3) W100,300,1D0

ИМЯ ФАЙЛА[ /Р]: /P

После вывода:

:-->

### 9.3.2. Директива 'R'.

По директиве 'R' с МГ вводятся данные в формате двоичного файла, записываются в область памяти, начальный и конечный адреса которой также вводятся с МГ.

Форма директивы:

**<R> [смещение]**

Если задано смещение, то данные будут записаны в память, начиная с адреса, равного сумме прочитанного начального адреса и смещения. После ввода этой директивы на экране появляется вопрос: ИМЯ ФАЙЛА[Р]:

Оператор должен ввести имя файла с контролем или без контроля, нажать на клавиатуре "ВК" и включить МГ на воспроизведение. Контрольная сумма анализируется только в том случае когда запись и считывание программы производится с контролем. Анализ контрольной суммы осуществляется методом генерации и проверки 16-разрядного кода контроля по избыточности. При этом методе правильная контрольная сумма при считывании файла должна равняться нулю.

Время, затраченное на подсчет контрольной суммы при данном методе впрямую зависит от объема проверяемого файла. Поэтому после считывания больших по объему программ информация о считывании появляется на экран ТВ не сразу, а после задержки в 20-40 секунд.

На экран выводятся в столбец: имя файла, начальный адрес, конечный адрес, стартовый адрес равный FFFF и если был указан ключ контроля при записи и считывании – контрольная сумма. Если при записи программ на ленту стартовый адрес был указан, то по окончании считывания она запускается автоматически.

Пример: R100

ИМЯ ФАЙЛА[Р]:ARD45/P

После считывания: ARD45

0200

0500

FFFF

КОНТР.СУММА:

0000

– правильное считывание.

Существуют три вида ошибок при считывании файла.

1) Ошибка контрольной суммы (не равна 0).

При этом на экран выводятся: имя файла, начальный, конечный и стартовый адреса программы, неправильная контрольная сумма и слово "Ошибка".

Ошибка контрольной суммы возникает и в том случае, когда запись на ленту производится без контроля, а считывание - с контролем.

Пример: R

ИМЯ ФАЙЛА[ /Р]:ARD45/P

После считывания: ARD45

0100

0300

0120

КОНТР.СУММА:

A0B5 \*ОШИБКА\*

2) Ошибка чтения тела файла.

При этом на экран выводится имя файла и слово "ошибка".

Пример: R

ИМЯ ФАЙЛА[ /Р]:ARD45/P

После считывания: ARD45 \*ОШИБКА\*

3) Ошибка чтения заголовка файла.

При этом на экран выводится только слово "ошибка".

Пример:

R

ИМЯ ФАЙЛА[ /Р]:ARD45/P

После считывания: \*ОШИБКА\*

Если имя файла не задано, то будет прочитан первый встречененный на ленте файл.

Директива 'R' позволяет производить поиск файла с указанным именем на ленте. В этом случае имена файлов, предшествующие искомому, выводятся на экран.

-59-

1П.00045-01 34 01

Пример:

R

ИМЯ ФАЙЛА[Р]:TST

T1

T2

T3

TST

0100

0200

FFFF

Если возникла ошибка в процессе чтения файлов, сообщение выводится после имени файла, прочитанного последним.

Пример:

R

ИМЯ ФАЙЛА[Р]:TST

T1

T2

T3 \*ОШИБКА\*

Аварийный выход из директивы 'R' - по клавише "УС".

ПОД	Очищает Текст	0000	УС
АСТ	Изменяет	0000	УС
ВУТ	Изменяет	0000	УС
ВСХ	Изменяет	0000	УС
ВЛХ	Изменяет	0000	УС
ВАУ	Изменяет	0000	УС
ВСР	Изменяет	0000	УС
ВРУ	Изменяет	0000	УС
ВАР	Изменяет	0000	УС

Выход в меню дальнейшего действия.

Помощь в работе с текстом для каждого из строк.

Помощь в работе с строками.

НАЧАЛЬНЫЕ И КОНЕЧНЫЕ АДРЕСА ВИДИМОЙ ЧАСТИ ВИДЕОЗУ

Номер строки	Начальный адрес	Конечный адрес
1	77BD	77FC
2	780B	784A
3	7859	788B
4	78A7	78E6
5	78F5	7934
6	7943	7982
7	7991	79D0
8	79DF	7A1E
9	7A2D	7A6C
10	7A7B	7ABA
11	7AC9	7B08
12	7B17	7B56
13	7B65	7BA4
14	7BB3	7BF2
15	7C01	7C40
16	7C4F	7C8E
17	7C9D	7CDC
18	7CEB	7D2A
19	7D39	7D78
20	7D87	7DC6
21	7DD5	7E14
22	7E23	7E62
23	7E71	7EB5
24	7EBF	7EFE
25	7F0D	7F4C

ДИРЕКТИВЫ РЕДАКТОРА

Директива | Выполняемое действие

A

Запуск РЕДАКТОРА по директиве МОНИТОРА

НАЧАЛО РАБОТЫ.

Очистка экрана, при ответе Y на запрос NEW? – очистка текстового буфера и установка режима ввода строки.

СТР

Выход в АССЕМБЛЕР.

УС и Е

Выход в МОНИТОР.

РАБОТА С МАГНИТОФОНОМ.

AP2+0

Вывод текста на магнитофон.

AP2+1

Ввод текста с магнитофона.

AP2+V

Сравнение текста, хранимого в ОЗУ, с введённым с магнитофона.

AP2+M

Ввод дополнительного фрагмента текста к уже имеющемуся в ОЗУ.

СТР

Отмена любой директивы работы с магнитофоном.

РЕЖИМ ВВОДА СТРОК.

ВК

Ввод в память набранной строки.

→, ←

Перемещение курсора для исправления строки.

СТР

Завершение ввода строк.

Директива : Выполняемое действие

**РЕДАКТИРОВАНИЕ ТЕКСТА.**

AP2+B Переход к началу текста

AP2+E Переход концу текста

AP2+A Ввод новой строки

AP2+↓ Просмотр фрагментов текста.

AP2+↑ Просмотр фрагментов текста в обратном порядке.

→ ↓ ↑ Перемещение курсора к началу первой строки.

→ ↓ ← ↑ Перемещение курсора к месту редактирования.

**ПОИСК ГРУППЫ СИМВОЛОВ.**

AP2+L Ввод группы символов.

VK Отображение фрагмента текста, в котором впервые встретились эти символы.

AP2+R Отображение следующего фрагмента текста с заданной группой символов.

**ИСПРАВЛЕНИЕ ОШИБОК.**

AP2+Ф4 Выключение автораздвижки символов.

AP2+Ф2 Выключение автораздвижки символов.

Директива : Выполняемое действие

02 Удаление символа.

04 Освобождение места для пропущенного символа.

AP2+1 Ввод одного псевдографического символа.

**УДАЛЕНИЕ ФРАГМЕНТА ТЕКСТА.**

↓,↑,AP2+↑ Перемещение курсора к началу первой строки удаляемого фрагмента.

AP2+D Маркировка первой строки удаляемого фрагмента.

↓,↑,AP2+↓ Перемещение курсора к началу строки, предшествующей последней удаляемой.

AP2+D Удаление фрагмента текста.

STR Отмена любой директивы удаления.

ДИРЕКТИВЫ АССЕМБЛЕРА

Директива	:	Выполняемое действие
1		Трансляция программы с отображением на экране протокола трансляции.
2		Трансляция программы с отображением на экране таблицы меток.
3		Трансляция программы с отображением на экране количества ошибок и адресов транслированной программы.
СТР		Выход в РЕДАКТОР.
УС и Е		Выход в МОНИТОР.

ПЕРЕЧЕНЬ ЛИТЕРАТУРЫ

1. Литература по программированию на Бейсике :

1.1. Дьяконов В.П Справочник по алгоритмам и программам на языке Бейсик для персональных ЭВМ-М., "Наука", 1987г.

1.2. Нико Дж. Бейсик : решение производственных задач. М.: "Машиностроение", 1987г.

1.3. Пул Л. Работа на персональном компьютере. -М., "Мир", 1986г.

1.4. Уорт Т. Программирование на языке Бейсик.-М., "Машиностроение", 1981г.

1.5. Эберт К., Эдерэр Х. Компьютеры. Применение в химии.-М., "Мир", 1988г.

1.6. Уолш Б. Программирование на Бейсике.-М., "Радио и связь", 1987г

2. Литература по программированию на Ассемблере.

2.1. Григорьев В.Л. Программное обеспечение микропроцессорных систем. : Энергоатомиздат, 1983 г.

2.2. Левенталь Л., Сейвилл У. Программирование на языке Ассемблера для микропроцессоров 8080 и 8085. М.: "Радио и связь", 1987 г.



ПРИЛОЖЕНИЕ 6

67  
1100045-013405

Лист регистрации изменений

Изм: Номера листов страниц	Всего:	N	Входящий	Подпись	Дата
----------------------------	--------	---	----------	---------	------

изме- нен- ных	заме- нен- ных	новых лиро- в иных	титулу- нице	документов и докум. и докум.	сопро- водитель- ного и докум. и и дата
1	-	бс2	-	-	48 111030-88
2.	2,6-11, 3-5,12, 42-49 13,16-37, 14,15,38-41 42-49	тиг.и. 5-7 39-55, 28-44	25-37, 4,24,38 66,64	-	56 111437-88
3.	8-23, тиг.и.	5-7	68	1111039-89	Бригада 11.12.88. МУЗ 24.11.88.

В РЕДАКТОР: 214:20  
40C:5F

ВНИМАНИЕ ПОТРЕБИТЕЛЯ !

ВНЕСЕНЫ ИЗМЕНЕНИЯ в ИЛ.00045-01 34 ОI :

Стр. 8. I4-я строка сверху. Следует читать :

... Партнер ОI.ОI (В.03)

Стр.I5. 4-я строка снизу. Следует читать :

... Партнер ОI.ОI (В.03)

Стр.I7. I4-я строка снизу. Следует читать :

Регистровая пара (РП) Н - адрес начала текста

Стр. 27. 4-я строка сверху. Следует читать :

IO I P TA: I P T B:I P TC ВК - строка с тремя опера-

торами

Стр.28. I2-я строка сверху. Следует читать :

... от  $-10^{-37}$  до  $10^{+37}$

Стр. 34. I3-я строка сверху. Следует читать :

IO I P T "ЖДУ ВВОДА"; A:I P T B:I P TC

Стр.36. I-я строка сверху. Следует читать :

IO I P T I:I P T