

МИНИСТЕРСТВО ПУТЕЙ СООБЩЕНИЯ СССР  
МОСКОВСКИЙ ОРДЕНА ЛЕНИНА  
И ОРДЕНА ТРУДОВОГО КРАСНОГО ЗНАМЕНИ  
ИНСТИТУТ ИНЖЕНЕРОВ ЖЕЛЕЗНОДОРОЖНОГО ТРАНСПОРТА

---

Кафедра математического обеспечения  
автоматизированных систем управления

## РАБОТА ЗА ПУЛЬТОМ ЭВМ «ИСКРА-226»

Методические указания

по дисциплине

«ПРОГРАММИРОВАНИЕ НА ЭВМ «ИСКРА-226»

Москва — 1987

353

У т в е р ж д е н о  
редакционно-издательским  
советом института

РАБОТА ЗА ПУЛЬТОМ ЭВМ "ИСКРА-226"  
Методические указания по дисциплине  
"Программирование на ЭВМ ИСКРА-226"  
для слушателей факультета повышения квалификации  
преподавателей и студентов

Москва - 1987

Методические указания составил преподаватель МИИТа  
доц. Г.М. Курбатов

Р е ц е н з е н т ы: канд. физ.-мат. наук А.Р. Ротенберг  
(ВНИИНС), канд. техн. наук Г.А.Шейкина  
(МИИТ)

## ВВЕДЕНИЕ

ЭВМ "ИСКРА-226" представляет собой клавишную мини-ЭВМ, предназначенную для решения в диалоговом режиме широкого круга инженерных, научно-технических, планово-экономических, оптимизационных, информационно-поисковых и учетно-статистических задач.

Выпуск машины предусмотрен в шести исполнениях, различающихся типом управляющей памяти процессора и составом устройств ввода/вывода [1,2] .

ЭВМ "ИСКРА-226" работает со встроенными и загружаемыми языками программирования. Основным языком ЭВМ "ИСКРА-226" является БЭЙСИК-интерпретатор. Вместе с ЭВМ вплоть до 1985 г. поставлялась версия языка БЭЙСИК 01. Однако уже разработаны и эксплуатируются версии языка БЭЙСИК 02, которые обладают более широкими возможностями по сравнению с языком БЭЙСИК 01. В дальнейшем для ЭВМ "ИСКРА-226" предполагается разработать следующие системы программирования: БЭЙСИК-компилятор, БЭЙСИК-графическую версию, РПГ-компилятор, ПАСКАЛЬ-интерпретатор, систему обработки структурированных данных [1] .

### 1. СОСТАВ И ХАРАКТЕРИСТИКИ ЭВМ "ИСКРА-226"

В состав ЭВМ "ИСКРА-226" входит процессор интерпретирующий диалоговый (ПИД) , состоящий из процессора интерпретирующего (ПИ) , блока отображения символично-графической информации (БОСГИ) , и устройства клавишного (УК) . Кроме ПИД в состав ЭВМ "ИСКРА-226" I-го исполнения входят: накопитель на гибких магнитных дисках (НГМД) и печатающее устройство (ПУ).

Другие исполнения ЭВМ "ИСКРА-226" отличаются от машины

исполнения I наличием накопителя на жестком магнитном диске (исполнение 2 и 3), накопителя на широкой магнитной ленте (исполнение 3), графопостроителя типа НЗ06 (исполнение 3, 4 и 6), указателя графической информации, аналого-цифрового и цифро-аналогового преобразователей, блока связи с СМ ЭВМ (исполнение 6).

С каждым из указанных выше внешних устройств ПИД взаимодействует через специальный согласующий блок - блок интерфейсный функциональный (БИФ). Замена в комплекте ЭВМ "ИСКРА-226" одного внешнего устройства на другое однотипное внешнее устройство связано с заменой соответствующего БИФа.

#### I.I. Процессор интерпретирующий диалоговый (ПИД)

Процессор интерпретирующий диалоговый [3] предназначен для исполнения загружаемых программ пользователя, управления устройствами ввода/вывода, а также осуществления режима диалога. Конструктивно процессор интерпретирующий и блок отображения символьно-графической информации выполнены в одном корпусе. Устройство клавишное подключается к процессору с помощью кабеля.

#### I.I.I. Процессор интерпретирующий (ПИ)

Процессор интерпретирующий имеет следующие технические характеристики:

емкость оперативной памяти - 64 К байт;

разрядность чисел - 13 десятичных разрядов;

форма представления чисел:

целые в диапазоне  $0 \leq |N| \leq 7999$ ;

вещественные в диапазоне  $10^{-99} \leq |N| \leq 10^{99}$ ;

среднее время выполнение операций:

арифметические - 0,001 с. ;

извлечение квадратного корня - 0,002 с. ;

элементарные функции - 0,05 с.

ПИ имеет загружаемую управляющую память емкостью 32К x 16 бит и постоянную управляющую память емкостью 8К x 16 бит. Одновременно к процессору может быть подключено до 7 устройств ввода/вывода.

### 1.1.2. Блок отображения символьно-графической информации (БОСТИ)

БОСТИ служит для индикации на экране текста программы или данных, осуществления функций редактирования, выдачи сообщений об ошибках и их локализации, индикации результатов счета по программе, выделения отдельных символов и текста в противоположной контрастности (черные символы на белом фоне), отображения графических построений [4]. Технические характеристики блока:

размер экрана по диагонали - 31 см ;

формат экрана: символьный канал - 24 строки по 80 символов в строке;

графический канал - 256 строк по 560 точек в строке .

БОСТИ ЭМ "ИСКРА-226" позволяет выводить на экран как символьную, так и графическую информацию. Для этих целей в машине организованы два канала связи с магистралью ввода/вывода: канал символьной информации и канал графической информации.

Канал символьной информации. При включении машины и загрузке ее внешним математическим обеспечением машина автоматически устанавливает связь с каналом символьной информации. На экране для указания места вывода очередного символа высвечивается специальный курсор: "\_" - подстрочная черточка. За начало экрана при выводе символьной информации принимается левый верхний угол - первая позиция первой строки.

В табл. I. I. приведены коды управления курсором и экраном при выводе символьной информации. Эти коды можно использовать совместно с оператором PRINT непосредственно в программе. Например, при выполнении строки PRINT HEX(03); TAB(30); HEX(12); "ВВОДИТЕ ДАННЫЕ"; HEX(11) очищается экран и курсор устанавливается на первое знакоместо. С 30 позиции первой строки в инверсном свечении выводится текст "вводите данные" и осуществляется переход к нормальному свечению экрана.

Установить курсор в заданную позицию экрана со стиранием части символов в строке можно также с помощью специального оператора PRINT AT . Форма записи оператора:

PRINT AT (P1,P2,P3) ,

где: P1 - номер строки, в которую перемещается курсор;

P2 - номер позиции в строке, в которой устанавливается курсор;

P3 - число стираемых в строке символов.

Пример:

PRINT AT(10,15,6)

При выполнении оператора PRINT AT курсор устанавливается в 15 позицию 10 строки экрана и в позициях с 15 по 20 стирается 6 символов.

Таблица I.I

Коды ВОСГИ (символьный канал)

№ п/п	НАИМЕНОВАНИЕ КОМАНДЫ	HEX-КОД
1	УСТАНОВКА КУРСОРА НА ПЕРВОЕ ЗНАКОМЕСТО ЭЛТ (В ЛЕВЫЙ УГОЛ) БЕЗ ОЧИСТКИ ЭКРАНА	01
2	ОЧИСТКА ЭКРАНА И УСТАНОВКА КУРСОРА НА ПЕРВОЕ ЗНАКОМЕСТО	03
3	РАНЕЕ ПОГАШЕННЫЙ КУРСОР ДОЛЖЕН СВЕТИТЬСЯ	05
4	КУРСОР ДОЛЖЕН СОХРАНЯТЬ СВОЕ МЕСТОПОЛОЖЕНИЕ НА ЭКРАНЕ, НО НЕ СВЕТИТЬСЯ	06
5	ЗВОНК	07
6	СДВИГ КУРСОРА НА ОДНО ЗНАКОМЕСТО ВЛЕВО ПО СТРОКЕ	08
7	СДВИГ КУРСОРА НА ОДНО ЗНАКОМЕСТО ВПРАВО ПО СТРОКЕ	09
8	ПЕРЕВОД КУРСОРА ВНИЗ	0A
9	ПЕРЕВОД КУРСОРА ВВЕРХ	0C
10	ПЕРЕВОД КУРСОРА НА ПЕРВОЕ ЗНАКОМЕСТО В СТРОКЕ	0D
11	ИНДИКАЦИЯ БЕЛЫХ СИМВОЛОВ НА ТЕМНОМ ФОНЕ	11
12	ИНДИКАЦИЯ ТЕМНЫХ СИМВОЛОВ НА БЕЛОМ ФОНЕ	12
13	ПЕРЕВОД КУРСОРА ВНИЗ В НАЧАЛО СЛЕДУЮЩЕЙ СТРОКИ, НА ПЕРВОЕ ЗНАКОМЕСТО	85



Канал графической информации. Для работы с графической информацией необходимо ввести команду `SELECT PRINT 10` и выполнить оператор `PRINT HEX(0D0F06)` ,

где: 10 - физический адрес экрана с графикой; 0D - код очистки экрана и установки курсора в начало координат; 0F - код снятия блокировки экрана; 06 - код засветки экрана.

Для указания вывода новой точки графического изображения на экране высвечивается курсор в виде символа "Г" . За начало координат принимается левый нижний угол экрана с координатами  $X_0Y_0$ , где:  $X_0$  - ось абсцисс, ориентированная по строке;  $Y_0$  - ось ординат, ориентированная по кадру.

В табл. I.1.2. приведены коды управления курсором и экраном при выводе графической информации.

Для перехода от вывода графической информации к выводу символьной информации необходимо выполнить команду `SELECT PRINT 05` и очистить экран оператором `PRINT HEX(03)` .

### I.1.3. Устройство клавишное (УК)

Устройство клавишное "ИСКРА-007-31" служит для формирования кодов символов, нанесенных на клавиши, и передачи этих кодов на вход процессора. На клавиатуре УК, изображенной на рис. I.1 можно условно выделить 8 функциональных зон [5] .

ЗОНА I содержит алфавитно-цифровую клавиатуру с дополнительно нанесенными на ней операторами языка БЭЙСИК. Эта часть клавиатуры управляется переключателем "рус/лат" и клавишей SHIFT. В положении переключателя регистров "лат" (латинские буквы) выводятся символы, изображенные в верхней части клавиш (как бы по верхнему регистру) . В положении пе-

Таблица 1.2

КОДЫ БОСГИ (графический канал)

№ п/п	НАИМЕНОВАНИЕ КОМАНДЫ	HEX-КОД
1	УСТАНОВКА КУРСОРА В НАЧАЛО КООРДИНАТ БЕЗ ОЧИСТКИ ЭКРАНА	01
2	ПЕРЕМЕЩЕНИЕ КУРСОРА $-\Delta X$ ; $-\Delta Y$	02
3	ПЕРЕМЕЩЕНИЕ КУРСОРА $+\Delta X$ ; $-\Delta Y$	03
4	ПЕРЕМЕЩЕНИЕ КУРСОРА $-\Delta X$ ; $+\Delta Y$	04
5	ПЕРЕМЕЩЕНИЕ КУРСОРА $+\Delta X$ ; $+\Delta Y$	05
6	ЗАСВЕТ	06
7	СТИРАНИЕ ЗАСВЕТА	07
8	ПЕРЕМЕЩЕНИЕ КУРСОРА $-\Delta X$	08
9	ПЕРЕМЕЩЕНИЕ КУРСОРА $+\Delta X$	09
10	СОХРАНЕНИЕ ИНФОРМАЦИИ	0B
11	ПЕРЕМЕЩЕНИЕ КУРСОРА $-\Delta Y$	0A
12	ПЕРЕМЕЩЕНИЕ КУРСОРА $+\Delta Y$	0C
13	ОЧИСТКА ЭКРАНА, УСТАНОВКА КУРСОРА В НАЧАЛО КООРДИНАТ	00
14	БЛОКИРОВКА ЭКРАНА	0E
15	СНЯТИЕ БЛОКИРОВКИ	0F

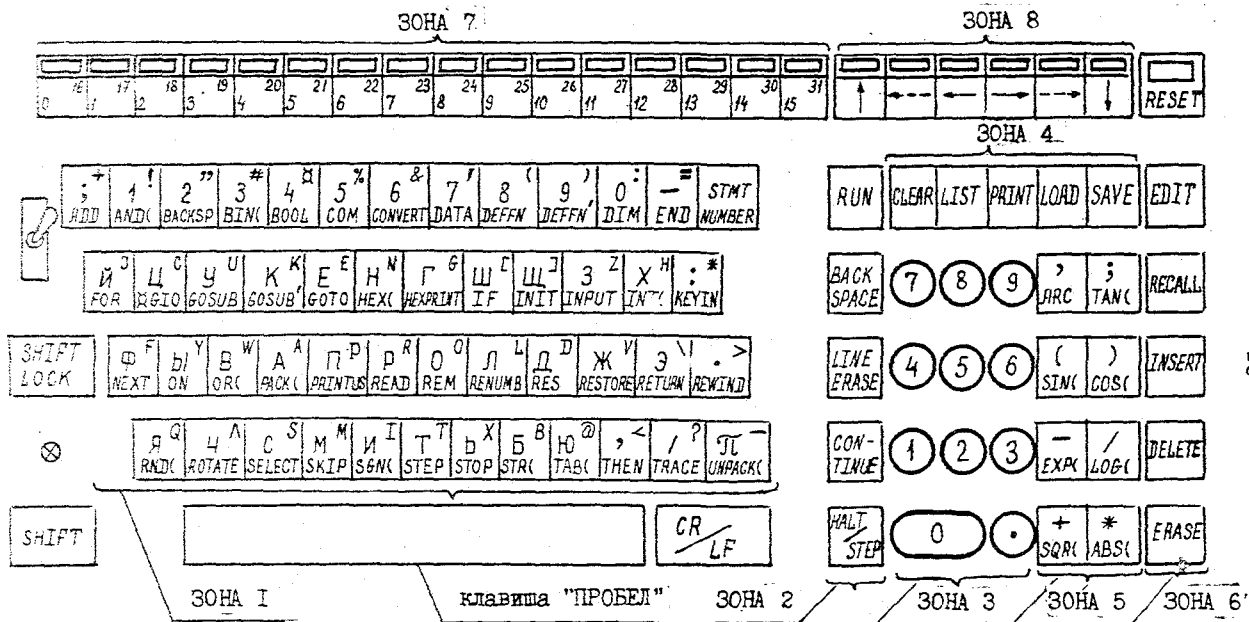


Рис.1.1 КЛАВИАТУРА МИКРО-ЭВМ "ИСКРА-226"

реключателя "рус" (русские буквы) вводятся символы, обозначенные в середине клавиш (как бы по среднему регистру). Нижний регистр включается и выключается соответственно при однократном нажатии клавиши SHIFT LOCK (загорается сигнальная лампочка) и SHIFT (сигнальная лампочка гасится). По нижнему регистру осуществляется ввод операторов языка, изображенных в нижней части клавиш (вводятся целые слова). Ввод операторов по нижнему регистру будет осуществляться также в том случае, если клавиша SHIFT постоянно нажата (зафиксирована пальцем). Операторы языка БЭЙСИК можно вводить также посимвольно по верхнему регистру (в положении переключателя "лат").

ЗОНА 2 содержит клавиши управления счетом и стиранием символов на экране:

**RUN** - вызывает исполнение программы;

**BACK SPACE** - служит для посимвольного стирания текста в процессе набора строки;

**LINE ERASE** - стирает текст всей строки вместе с ее номером (до посылки строки в память машины);

**HALT/STEP** - служит для пошагового исполнения программы "оператор за оператором";

**CONTINUE** - служит для выхода из режима пошагового исполнения и перехода к нормальному выполнению программы.

ЗОНА 3 содержит стандартную цифровую клавиатуру и десятичную точку: "0", "1", "2", "3", "4", "5", "6", "7", "8", "9", "." .

ЗОНА 4 содержит часто используемые операторы языка БЭЙСИК:

- CLEAR - очищает память машины от ранее введенных программ и данных;
- LIST - выводит текст программы, находящейся в памяти машины, на указанное устройство;
- PRINT - служит для индикации или распечатки цифровой и символьной информации;
- LOAD - обеспечивает чтение программы с магнитного диска и загрузку ее в память машины;
- SAVE - служит для записи программы, хранящейся в памяти машины, на магнитный диск;

ЗОНА 5 содержит клавиши ввода стандартных математических функций, знаков арифметических действий, круглых скобок и разделителей.

При нажатой клавише SHIFT вводятся идентификаторы элементарных математических функций с открывающейся левой скобкой: ARC, TAN, SIN, COS, EXP, LOG, SQR, ABS .

При отжатой клавише SHIFT (положение переключателя "рус/лат" любое) вводятся: "+" (ПЛЮС), "-" (МИНУС), "\*" (УМНОЖИТЬ), "/" (РАЗДЕЛИТЬ), "(" , ")", " , " ; " .

ЗОНА 6 содержит клавиши редактирования текста программы:

- EDIT - переводит программную строку с указанным номером в состояние редактирования;
- RECALL - вызывает текст редактируемой строки на экран;
- INSERT - раздвигает строку от местоположения курсора для вставки нового символа;
- DELTE - стирает символ на местоположении курсора, а часть текста, расположенного справа от курсора.

сора, сдвигается на одну позицию влево;

ERASE - стирает часть строки от местоположения курсора вправо и до конца строки.

ЗОНА 7 содержит 16 клавиш, с помощью которых пользователь может реализовать 32 специальные функции. В положении переключателя "лат" реализуются функции с номерами от 0 до 15. В положении переключателя "рус" - функции с номерами от 16 до 31.

ЗОНА 8 содержит 6 клавиш управления курсором в режиме редактирования при нажатых клавишах EDIT и RECALL :

- ↑ - перемещение курсора на одну строку вверх;
- ←←←← - перемещение курсора на пять позиций влево по строке;
- ← - перемещение курсора на одну позицию влево по строке;
- - перемещение курсора на одну позицию вправо по строке;
- - перемещение курсора на пять позиций вправо по строке;
- ↓ - перемещение курсора на одну строку вниз.

Остальные клавиши выполняют следующие функции. Клавиша CR/LF служит для передачи строки программы или оператора управления в память машины. Клавиша RESET используется для передачи команды аварийного останова. По этой команде прекращается выполнение программы, действие устройств ввода/вывода, на экране гасится изображение и управление передается пользователю. Клавиша ПРОБЕЛ служит для перемещения курсора на следующую позицию в строке.

### 1.2. Накопитель на гибких магнитных дисках (НГМД)

НГМД "ИСКРА 005-50" является внешней памятью ЭВМ "ИСКРА-226" [6]. НГМД имеет два кармана: "R" - маркировка верхнего дисководов и "F" - маркировка нижнего дисководов. На передней панели НГМД размещены органы управления и индикации. Внизу расположен выключатель питания с встроенной лампочкой, сигнализирующей о готовности НГМД к работе. Рядом с выключателем расположена клавиша для восстановления начального состояния НГМД (перевод магнитной головки на дорожку с адресом 00, установка защиты от записи и т.п.). Выше выключателя расположены индикаторы выбора диска (зеленые лампочки), защиты записи (красные лампочки) и клавиши сброса защиты записи соответственно для накопителей "F" и "R".

Для НГМД носителями информации является два сменных магнитных диска типа ЕС-5074 с двумя рабочими поверхностями на каждом. Поверхность диска имеет 77 дорожек. Дорожка содержит 13 секторов по 256 байт. Таким образом информационная емкость одной поверхности гибкого диска составляет порядка 256 К байт (1001 сектор x 256 байт).

ЭВМ "ИСКРА-226" может также работать с НГМД "ИСКРА 005-51", который обеспечивает доступ только к одной стороне гибкого диска.

### 1.3. Печатающее устройство (ПУ)

ПУ предназначено для вывода текстов программ и результатов счета на бумажный носитель. ЭВМ "ИСКРА-226" комплектуется ПУ типа РОБОТРОН 1154. Его технические характеристики: скорость печати - до 50 знаков в секунду, максимальная длина

строки - 132 знака, структура знака - точечная матрица 7x5, высота знака - 2,6 мм, ширина знака - 2,0 мм, расстояние между знаками - 2,54 мм.

РОБОТРОН 1154 в комплекте ЭВМ "ИСКРА-226" может быть заменен другими печатающими устройствами типа РОБОТРОН 1156M ДЗМ 180 .

ЭВМ "ИСКРА-226" 2-го и 3-го исполнения имеют в комплекте накопитель на магнитном диске (НМД) "ИСКРА 005-71" с двумя дисками - сменным и фиксированным. Емкость каждого диска - 2,5 М байт.

ЭВМ "ИСКРА-226" 3-го исполнения имеет накопитель на магнитной ленте (НМЛ) "ИСКРА 005-61". В накопителе используется 9-ти дорожечная лента. Емкость накопителя -  $10^8$  бит. Время перемотки ленты не более 300 с. Скорость обмена информацией ~ 10 К байт/с.

ЭВМ "ИСКРА-226" 3-го, 4-го и 6-го исполнения имеют в комплекте графопостроитель зависимостей  $X = f(t)$  и  $Y = f(t)$  типа Н 306 . Размеры рабочего поля записи графика: по каналу  $X$  - 30 см. , по каналу  $Y$  - 20 см . Максимальная скорость регистрации - 75 см/с.



## 2. СИСТЕМНЫЕ КОМАНДЫ И РЕЖИМЫ РАБОТЫ ЭВМ "ИСКРА-226"

### 2.1. Адресация устройств ввода/вывода

При решении задач на ЭВМ "ИСКРА-226" пользователь осуществляет операции ввода/вывода с применением различных внешних (консольных) устройств машины. Основным консольным устройством для ввода данных CI(CONSOLE INPUT) является устройство клавишное с физическим адресом (ФАУ) 01. Основным консольным устройством для вывода данных CO(CONSOLE OUTPUT) служит блок отображения символьно-графической информации с ФАУ 05. Физические адреса этих устройств выбираются автоматически при включении машины и образуют ее пульт управления [1]. Физические адреса основных консольных устройств ЭВМ "ИСКРА-226" приведены в табл.2.1.

Таблица 2.1

Физические адреса консольных устройств

ФАУ(десятичный)	ФАУ(шестнадцатичный)	Тип консольного устройства
1	01	УК
5	05	БССГИ(символьный канал)
12	0C	ПУ
16	10	БССГИ(графический канал)
20	14	Графопостроитель
24	18	НГМД
27	1B	НМЛ
28	1C	НМД
32	20	АШУ

## 2.2. Выбор режима работы машины (команда SELECT)

Системная команда SELECT задает различные режимы работы ЭМ "ИСКРА-226".

Переопределение адреса консольного устройства. С помощью команды SELECT можно переопределять адреса устройств ввода/вывода:

SELECT CI<ФАУ> - переопределяет адрес консольного устройства ввода, например: SELECT CI 03 ;

SELECT CO<ФАУ> - переопределяет адрес консольного устройства вывода, например: SELECT CO 0C (64) ;

Переопределение адреса консольного устройства, ориентированного на работу с указанным устройством:

SELECT DISK<ФАУ> - задает адрес консольного устройства, ориентированного на работу с дисковыми операторами, например: SELECT DISK 18 ;

SELECT PLOT<ФАУ> - задает адрес консольного устройства, ориентированного на работу с операторами PLOT, например SELECT PLOT 05 (60) ;

SELECT LIST<ФАУ> - задает консольное устройство для вывода текста программы, например: SELECT LIST 05 (60) ;

SELECT PRINT<ФАУ> - задает консольное устройство для вывода информации по операторам PRINT, PRINTUSING, HEXPRINT, например: SELECT PRINT 0C (120) ;

SELECT INPUT<ФАУ> - задает консольное устройство ввода данных по операторам INPUT, KEYIN, например: SELECT INPUT 02 ;

SELECT #<логический номер><ФАУ> - назначает логический номер устройству с заданным ФАУ, например: SELECT # 2 0C .

Если в команде SELECT параметр <длина строки> отсутствует, то по умолчанию строка принимается равно 80 символам.

Замедление выполнения программы. Команда SELECT позволяет вводить задержку при выполнении программы. Форма задания команды: SELECT P <цифра>, где параметр <цифра> может принимать целые значения от 0 до 9 и определяет соответственно паузы от 0 до 1,5 с. с шагом 1/6 с.

Примеры:

- А. SELECT P6 - выполнение строк программы и их индикация будет осуществляться с интервалом в 1 с. ;
- В. SELECT P0 - переход к нормальному темпу исполнения программы.

Установление единицы измерения аргумента тригонометрических функций. Командой SELECT можно задавать градусы, радианы или грады ( $2\pi = 400^\circ$ ), как единицу измерения тригонометрических функций.

Примеры:

- А. SELECT R - аргументы тригонометрических функций в программе воспринимаются в радианах.
- В. SELECT D - аргументы тригонометрических функций в программе воспринимаются в градусах.
- С. SELECT G - аргументы тригонометрических функций в программе воспринимаются в градах.

Заданный командой SELECT режим действует до тех пор, пока: не задан с помощью этой же команды другой режим работы, не поступила команда CLEAR, не нажата кнопка начальной установки.

### 2.3. Очистка памяти машины (команда CLEAR)

Системная команда CLEAR служит для очистки памяти машины от ранее введенных программ и данных.

Команда CLEAR без параметров стирает в памяти машины как программу, так и данные. Чтобы убедиться в этом, необходимо после выполнения команды CLEAR задать команду LIST. Появление на экране сообщения об ошибке (ERR 23) свидетельствует об отсутствии информации в памяти машины.

Команда CLEAR P [`<строка 1>` [, `<строка 2>`]] позволяет стирать в памяти машины фрагмент программы, начинающийся с номера, заданного параметром `<строка 1>` и заканчивающийся номером, заданным параметром `<строка 2>`.

Примеры:

CLEAR P - стирает в памяти машины текст программы, данные сохраняются ;

CLEAR P 10,100 - стирает текст программы с 10 по 100-ю строку включительно, данные сохраняются ;

CLEAR P 50 - стирает текст программы с 50 строки до конца, данные сохраняются ;

CLEAR P,50 - стирает текст программы с начала и по 50 строку включительно, данные сохраняются.

Команда CLEAR V очищает память машины от ранее введенных данных, текст программы сохраняется.

Команда CLEAR N очищает память машины от ранее введенных данных, кроме данных, определенных в операторе CCM, текст программы сохраняется.

## 2.4. Вывод текста программы (команда LIST)

Системная команда LIST служит для вывода текста программы на заданное устройство.

Команда LIST без параметров индицирует полный текст программы.

Команда LIST [<строка 1> [, <строка 2>]] позволяет вывести из оперативной памяти машины фрагмент программы, начинающийся с номера, заданного параметром <строка 1> и заканчивающийся номером, заданным параметром <строка 2> .

Примеры:

LIST 100 - индицируется 100-я строка программы ;

LIST 100,150 - индицируется фрагмент программы с 100-й по 150 строку включительно ;

LIST ,200 - индицируется программа с начала и до 200-й строки включительно.

Команда LIST S - позволяет выводить текст программы страницами по 23 строки. Для вывода следующей страницы необходимо нажимать на клавишу CR/LF .

Команда LIST R - индицирует номера строк, содержащих переходы по операторам IF, GOTO, COSUB, ON ERROR и др. к указанной строке или диапазону строк. Пример записи команды: LIST R 100,350 .

Команда LIST V - индицирует номера строк, содержащих указанную переменную, диапазон переменных или все переменные. Пример записи команды: LIST V T, T1, T2 .

Команда LIST ' - индицирует номера строк, содержащих операторы GOSUB и DEFFN с указанными номерами (например LIST ' 24) или всех операторов GOSUB и DEFFN (например LIST

## 2.5. Автоматическая нумерация строк программы (команда STMT NUMBER)

Команда STMT NUMBER служит для автоматической нумерации строк программы при ее наборе. После очистки памяти машины при первом нажатии на клавишу STMT NUMBER первой строке программы присваивается номер 10. При последующих нажатиях — очередной строке программы присваивается номер на 10 больше.

## 2.6. Перенумерация строк программы (команда RENUMBER)

Команда RENUMBER служит для перенумерации строк программы. Форма записи команды:

RENUMBER <строка 1>[, <строка 2>[, <шаг>]],

где: <строка 1> — номер первой строки, с которой осуществляется перенумерация ;

<строка 2> — задает новый номер строке 1 ;

<шаг> — шаг новой нумерации (число от 0 до 99, по умолчанию принимается равным 10).

Примеры:

RENUMBER — перенумеровывает всю программу, начиная с 10 строки с шагом 10 ;

RENUMBER 100,150,5 — строка с номером 100 получает новый номер 150, остальные строки нумеруются с шагом 5 ;

RENUMBER ,5,5 — первая строка получает номер 5, остальные строки нумеруются с шагом 5 ;

RENUMBER 5 — строка 5 получает номер 10, остальные строки нумеруются с шагом 10 .

Все обращения к номерам строк по операторам GOTO, IF, GOSUB, ON, ON ERROR, PPINTUSING и др. автоматически получают новые номера.

### 2.7. Исполнение программы (команда RUN)

Команда *RUN* служит для выполнения счета по программе. Форма записи команды: *RUN* <номер строки>, где <номер строки> — строка, с которой начинается выполнение программы. При посылке в машину команды *RUN* осуществляется следующее: проверяется правильность загруженной программы, цифровым переменным присваивается значение ноль, а символьным переменным — значение пробел, указатель текущего значения, заданного оператором *READ*, возвращается к первому значению *DATA*, программа выполняется со строки с наименьшим номером или с заданного номера строки.

Примеры:

*RUN* — программа выполняется сначала;

*RUN* 100 — программа выполняется с 100-й строки.

### 2.8. Прерывание выполнения программы (команда RESET)

Команда *RESET* служит для аварийного прерывания выполнения программы. При нажатии на клавишу *RESET* прекращается выполнение текущей строки программы, действия устройств ввода/вывода, очищается экран дисплея и управление передается пользователю. Программа при этом в памяти машины не стирается.

Основные системные команды ЭМ "ИСКРА-226" приведены в табл. I приложения [1, 9].

### 3. ВЫПОЛНЕНИЕ ПРОГРАММЫ НА ЭВМ "ИСКРА-226"

#### 3.1. Структура программы

Основной программной единицей языка БЭЙСИК является строка. Строка бывает двух типов: строка непосредственного счета и программная строка.

Строка непосредственного счета служит для выполнения несложных математических расчетов, а также для вывода значе- ний цифровых выражений и промежуточных результатов в про- цессе отладки программы. Строка непосредственного счета на- бирается без номера. Она может содержать несколько простых операторов типа присваивания и печати, разделенных двоето- чием. Строка после ее набора вводится в машину нажатием на клавишу CR/LF, где анализируется на отсутствие ошибок. Если в строке обнаружена ошибка, то сообщение об этом выдается на экран. При отсутствии ошибок строка сразу же выполняется.

Пример: `SELECT D:X=30:Y=45:PRINT (SIN(X)+COS(Y))`

1.207106781187

Строка вычисляет значение выражения  $\sin(30)+\cos(45)$ . Аргументы заданы в градусах.

Программная строка служит для набора текста программы. Программа на языке БЭЙСИК состоит из пронумерованных програм- мных строк. Каждая строка программы имеет номер — целое число от 1 до 9999. Он необходим для: идентификации строки, уста- новления очередности выполнения строк в программе, обращения к требуемой строке в процессе выполнения программы. Строки в программе принято нумеровать с шагом 10. Это позволяет в дальнейшем при отладке или изменении программы вставлять



новые строки между уже имеющимися. В каждой строке программы можно записывать один или несколько операторов, разделенных знаком ":" . Количество символов в строке не должно превышать 240.

### 3.2. Набор текста программы

Перед набором программы память машины необходимо очистить от ранее введенных программ и данных. Информацию, хранящуюся в машине, стирают командой CLEAR . Ввод текста программы начинают с ввода номера первой программной строки — числа 10 . Его можно ввести двумя способами:

набрать номер 10 на цифровой клавиатуре;

нажать на клавишу STMT NUMBER .

При первом нажатии на клавишу STMT NUMBER первой программной строке приписывается номер 10, при каждом последующем нажатии — на 10 больше, т.е. 20, 30 и т.д. Требуемый номер строки можно также набирать с помощью цифровых клавиш. Строка программы набирается и визуальнo контролируется пользователем на экране. Если в строке обнаружена ошибка до ввода ее в машину, ошибку можно исправить с помощью клавиш BACK SPACE и LINE ERASE (см.п.3.4) . Набор строки завершается записью ее в память машины нажатием на клавишу CR/LF. При этом проверяется ее синтаксическая правильность и, если нет ошибок, строка записывается в память машины, а курсор устанавливается в начало следующей строки экрана. Пользователь после этого может приступить к набору следующей строки. В случае обнаружения ошибки машина записывает ошибочную строку в память, но выдает на экран сообщение об ошибке. Пользователь может повторить ввод неправильно набранной строки под



к системному сбою. В этом случае необходимо нажать клавишу RESET (или SR и RUN 2) и продолжить работу.

Ошибки исполнения возникают при попытке выполнить недопустимые математические операции, переходы, циклы и т.п.

Например, при выполнении программы:

```
10 X=5
20 Y=SQR(SIN(X))
30 PRINT Y
```

в 20 строке возникает сообщение об ошибке

```
20 Y=SQR(SIN(X))
^ERR03
```

Ошибка заключается в попытке извлечь квадратный корень из отрицательного числа.

Если останов при ошибке в процессе счета нежелателен, то в программе необходимо предусмотреть оператор ON ERROR, осуществляющий переход к заданному номеру строки в случае обнаружения ошибки.

Программные ошибки являются следствием некорректно составленной программы. Они могут быть вызваны неправильным использованием данных, операторов и т.п. Например, при исполнении программы

```
10 X=5
20 Y=2
30 Z=2^X+3*Y
40 PRINT X,Z
50 Y=Y+1
60 IF Y<=10THEN20
```

происходит "зацикливание", т.к. 60-я строка каждый раз ошибочно передает управление 20 строке, а не 30 строке.

Программные ошибки трудно сразу локализовать. Быстрой локализации и устранению ошибок способствует режим пошаговой отладки (см. п.3.6) .

### 3.4. Исправление ошибок в программе

В процессе набора текста программы пользователь допускает ошибки и обнаруживает их на экране. Если при этом еще не нажата клавиша CR/LF (строка не записана в память машины), то ее можно отредактировать с помощью клавиш BACK SPACE и LINE ERASE. При нажатии клавиши BACK SPACE курсор на экране смещается на одну позицию влево и стирает символ в этой позиции. Клавиша LINE ERASE стирает всю строку вместе с ее номером. Новая строка или часть текста строки вводятся обычным способом. Кроме того пользователь при наборе текста программы может выполнять следующие операции.

**З а м е н у** одной строки в тексте программы другой строкой. Для этого необходимо набрать новый текст строки под старым номером и нажать на клавишу CR/LF.

**И с к л ю ч е н и е** строки из текста программы. Для этого необходимо набрать номер исключаемой строки и нажать на клавишу CR/LF.

**В с т а в л е н и е** новой строки в текст программы. Для этого необходимо новой строке присвоить номер  $K (P \leq K \leq M)$ , набрать ее текст и нажать на клавишу CR/LF. P и M соответственно номера верхней и нижней строк программы, между которыми вставляется строка с номером K.

Пример исправления программы.

```
10 INPUT X,Y
20 Z=X+Y
30 PRINT Z
40 STOP
50 END
```

<u>Выполняемая функция</u>	<u>Действия пользователя</u>
Замена строки	20 Z=X <sup>2</sup> -Y CR/LF
Исключение строки	40 CR/LF
Вставление строки	15 PRINT X,Y CR/LF

В результате внесенных исправлений получим измененный текст программы:

```

10 INPUT X,Y
15 PRINT X,Y
20 Z=X-Y
30 PRINT Z
50 END .

```

### 3.5. Редактирование текста программы

Строка, набранная на экране дисплея и посланная в машину, записывается в ее память даже в том случае, если в ней обнаружена ошибка. Ошибочную или неверно набранную строку можно снова вызвать на экран и отредактировать. Вызов и редактирование строки осуществляется клавишами редактирования EDIT, RECALL, INSERT, DELETE, ERASE, клавишами BACK SPACE, LINE ERASE и шестью клавишами управления курсором ↑, ←, →, ↓, ←→, →←. Функции, выполняемые этими клавишами, описаны в разделе п.1.3. Клавиша LINE ERASE в режиме редактирования стирает часть строки от местоположения курсора влево и до конца строки. Клавиша BACK SPACE действует обычным образом. Клавиши редактирования и ключи управления курсором действуют только в пределах редактируемой строки.

Пример редактирования строки программы:

$Y=X/\cos(X)+3*X$       ←→       $Y=X+2/\cos(X)$  .

<u>ДЕЙСТВИЯ</u> <u>ПОЛЬЗОВАТЕЛЯ</u>	<u>ИНФОРМАЦИЯ НА ДИСПЛЕЕ</u>	<u>ПОЯСНЕНИЕ</u>
НАБРАТЬ НОМЕР РЕДАКТИРУЕМОЙ СТРОКИ И НАЖАТЬ КЛАВИШУ EDIT	*80	80-Я СТРОКА ПЕРЕВОДИТСЯ В РЕЖИМ РЕДАКТИРОВАНИЯ
НАЖАТЬ КЛАВИШУ RECALL	*80 $Y=X/(\cos(X)+3*X)$	ТЕКСТ РЕДАКТИРУЕМОЙ СТРОКИ ВЫЗЫВАЕТСЯ НА ДИСПЛЕИ
НАЖАТЬ КЛЮЧ ←---- ДВА РАЗА И КЛЮЧ ←- ТРИ РАЗА	*80 $Y=X/(\cos(X)+3*X)$	КУРСОР СОВМЕЩАЕТСЯ С СИМВОЛОМ /
НАЖАТЬ КЛАВИШУ INSERT ДВА РАЗА	*80 $Y=X/(\cos(X)+3*X)$	ЧАСТЬ СТРОКИ ОТ МЕСТОПОЛОЖЕНИЯ КУРСОРА СМЕЩАЕТСЯ ВПРАВО НА ДВЕ ПОЗИЦИИ
ВПЕЧАТАТЬ СИМВОЛЫ Λ2	*80 $Y=X\Lambda2/(\cos(X)+3*X)$	ВНОСИТСЯ ИЗМЕНЕНИЕ В ТЕКСТ СТРОКИ
НАЖАТЬ КЛЮЧ → ОДИН РАЗ	*80 $Y=X\Lambda2/(\cos(X)+3*X)$	КУРСОР СОВМЕЩАЕТСЯ С ПЕРВОЙ ПРАВОЙ СКОБКОЙ
НАЖАТЬ КЛАВИШУ DELETE ОДИН РАЗ	*80 $Y=X\Lambda2/\cos(X)+3*X)$	СТИРАЕТСЯ СКОБКА НА МЕСТОПОЛОЖЕНИИ КУРСОРА, ЧАСТЬ ТЕКСТА, РАСПОЛОЖЕННАЯ СПРАВА ОТ КУРСОРА, СМЕЩАЕТСЯ НА ОДНУ ПОЗИЦИЮ ВЛЕВО
НАЖАТЬ КЛЮЧ ---→ ОДИН РАЗ И КЛЮЧ → ОДИН РАЗ	*80 $Y=X\Lambda2/\cos(X)+3*X)$	КУРСОР СОВМЕЩАЕТСЯ СО ЗНАКОМ "+"
НАЖАТЬ КЛАВИШУ ERASE	*80 $Y=X\Lambda2/\cos(X)$	СТИРАЕТСЯ ЧАСТЬ СТРОКИ ОТ МЕСТОПОЛОЖЕНИЯ КУРСОРА ВПРАВО И ДО КОНЦА СТРОКИ
НАЖАТЬ КЛАВИШУ CR/LF	*80 $Y=X\Lambda2/\cos(X)$ : _	СТРЕДАКТИРОВАННАЯ СТРОКА ВВОДИТСЯ В ПАМЯТЬ МАШИНЫ

### 3.6. Отладка программы

Ф

Наиболее ответственным и трудоемким моментом при программировании является отладка программы. Начинать отладку программы можно уже на этапе ее составления, где рекомендуется:

проверить возможность решения задачи при ручном счете на конкретном примере, обратив особое внимание на предельные значения исходных параметров;

проследить ход решения задачи с теми же, что и при ручном счете, исходными данными по блок-схеме, а также по тексту программы.

На этапе набора текста программы рекомендуется:

расчленить программу на логически законченные блоки, в конце каждого блока предусмотреть выдачу промежуточных результатов и оператор STOP для программного прерывания выполнения программы;

проверить работу отдельных блоков, задав программе исходные данные, результаты счета с которыми заранее известны;

проверить правильность записи всех строк программы, обращая особое внимание на операторы управления типа IF, GOTO, GOSUB, и возможность выполнения заданных условий;

при работе с индексированными переменными убедиться, что строки и столбцы не перепутаны (например, данные в блоке DATA записаны по столбцам, а их ввод осуществляется по строкам), индексы изменяют свои значения и не превосходят предельно допустимые, количество и тип данных соответствует требуемому и т.п.

Если при счете по программе ошибка все-таки возникает, то

необходимо использовать специальные режимы отладки программы: TRACE, HALT/STEP, а также TRACE + HALT/STEP .

### 3.6.1. Режим слежения TRACE

Режим TRACE позволяет пользователю следить за ходом выполнения программы. Если в машину введена команда TRACE, то на экран выводятся результаты выполнения операций (по операторам типа LET, INPUT, READ) и указания переходов (по операторам типа IF, GOTO, GOSUB и т.п.).

В режиме TRACE для удобства наблюдения за результатами вычислений рекомендуется либо вводить задержку (см. п.2.), либо переадресовывать консольное устройство вывода и получать результаты на печатающем устройстве. В этом случае необходимо выполнить команды:

```
SELECT CO XC CR/LF
TRACE CR/LF
GOTO I0 CR/LF
RUN CR/LF
```

Для выхода из режима TRACE необходимо набрать команду TRACE OFF и нажать на клавишу CR/LF .

Пример исполнения программы в режиме TRACE.

Программа:

```
10 INPUT A,B
20 C=A+B
30 GOTO 50
40 END
50 D=B-A
60 PRINT C,D
70 GOTO 40
```



Исполнение программы:

SELECT P4	CR/LF	- ввод задержки;
TRACE	CR/LF	- задание режима TRACE;
:RUN	CR/LF	- запуск программы;
? 2,1	CR/LF	- ввод исходных данных;
C=3		- печать значения C
TRANSFER TO 50,		- переход к 50 строке;
D=-1		- печать значения D;
3	-1	- печать значений C и D;
TRANSFER TO 40		- переход к 40 строке;
END PROGRAMM		- конец программы.
FREE SPACE -64470		
:		
TRACE OFF	CR/LF	

### 3.6.2. Режим пошаговой отладки программы HALT/STEP

Возможны два способа использования режима HALT/STEP

В первом случае нажатие на клавишу HALT/STEP в процессе выполнения программы вызывает "прерывание" после исполнения очередной строки. На экране высвечивается строка останова и результат выполнения последнего оператора. Пользователь при этом может проанализировать полученные на момент прерывания результаты, поправить на печатающем устройстве бумагу и т.п. Для продолжения счета по программе необходимо нажать на клавишу CONTINUE.

Во втором случае клавиша HALT/STEP используется для пошагового исполнения программы, "оператор за оператором".

Выход на начало программы осуществляется командой:

GOTO <номер строки> .

К нормальному исполнению программы можно вернуться, нажав на клавишу CONTINUE .

Пример исполнения программы в режиме HALT/STEP.

Программа:	Исполнение программы:	
10 INPUT A,B	60 TO 10	CR/LF
20 C=A+B	:	
30 GOTO 50	:	HALT/STEP
40 END	10 INPUT A,B	
50 D=B-A	? 2,1	CR/LF
60 PRINT G,D	:	
70 GOTO 40	20 C=A+B	HALT/STEP
	:	
	:	HALT/STEP
	30 GOTO 50	
	:	
	50 D=B-A	HALT/STEP
	:	
	60 PRINT G,D	HALT/STEP
	3	-1
	:	
	70 GOTO 40	HALT/STEP
	:	
	40 END	HALT/STEP
	:	
	END PROGRAM	
	FREE SPACE-64468	
	:	

### 3.6.3. Совместное использование команды TRACE и клавиши HALT/STEP при отладке программы

Команда TRACE и клавиша HALT/STEP используются совместно в тех случаях, когда в процессе отладки программы необходимо проследить за ходом выполнения операторов и расширить объем оперативной выдачи результатов.

Пример исполнения программы в режиме TRACE + HALT/STEP.

Программа:	Исполнение программы:
10 INPUT A,B	:TRACE CR/LF
20 C=A+B	
30 GOTO 50	:GOTO 10 CR/LF
40 END	TRANSFER TO 10
50 D=B-A	:
60 PRINT C,D	10 INPUT A,B HALT/STEP
70 GOTO 40	? 2,1 CR/LF
	:
	20 C=A+B HALT/STEP
	C=3
	:
	30 GOTO 50 HALT/STEP
	TRANSFER TO 50
	:
	50 D=B-A HALT/STEP
	D=-1
	:
	60 PRINT C,D HALT/STEP
	3 -1
	:
	70 GOTO 40 HALT/STEP
	TRANSFER TO 40
	:
	HALT/STEP
	40 END
	END PROGRAMM
	FREE SPACE- 64468
	:TRACE OFF CR/LF

В режимах TRACE и HALT/STEP+TRACE пользователь может на каждом шаге дополнительно вызывать на экран значения любых переменных, уже известных в программе (например, PRINT A;B;X), выполнять несложные расчеты (например PRINT  $A \times B^2 + B$ ) или производить редактирование строк программы.

### 3.7. Выполнение счета по программе

Запуск программы на исполнение осуществляется командой RUN . Программа выполняется со строки, имеющей наименьший номер, и ее исполнение продолжается в порядке возрастания номеров строк. В случае, если счет по программе с первой строки нежелателен, например, при повторном счете, когда данные уже введены, в операторе RUN указывают номер строки, с которого необходимо начать исполнение программы.

Программа может иметь один или несколько операторов STOP. В этом случае при счете осуществляется запланированный останов, который позволяет пользователю оценить промежуточные результаты, подготовить печатающее устройство, сменить диск в накопителе и т.п. Продолжить счет по программе можно нажав на клавишу CONTINUE.

Выбор устройства для вывода результатов счета по программе осуществляется командой SELECT см.п.2.2. .

### 3.8. Определение времени счета по программе

Время, затрачиваемое машиной на выполнение программы, определяется с помощью таймера. Для организации таймера служит оператор INPUT \*.

Форма записи оператора:

$$\text{INPUT } * \left\{ \begin{array}{l} \langle \text{простая цифровая переменная} \rangle \\ \langle \text{элемент цифрового массива} \rangle \end{array} \right\} ,$$

где  $\langle \text{простая цифровая переменная} \rangle$  - переменная, в которую записывается текущее значение таймера .

Для вычисления отрезка времени, затрачиваемого машиной на выполнение программы, необходимо предусмотреть операторы INPUT\* в начале и в конце программы, а затем вычислить раз

ность показаний, зафиксированных в указанных цифровых переменных. Запуск таймера осуществляется автоматически. дискретность таймера - 2 мс.

Пример организации таймера в программе:

```
10 INPUT *T1
20 S, K=1 '
30 FOR N=1 TO 100
40 S=S+SIN(N*K)
50 NEXT N
60 PRINT "S="; S
70 INPUT *T2
80 PRINT "ВРЕМЯ СЧЕТА РАВНО"; (T1-T2)/2000; "С."
```

```
S= .8720289863463
ВРЕМЯ СЧЕТА РАВНО 3.506 С.
```

#### 4. ДИСКОВЫЕ ОПЕРАЦИИ И РАБОТА С ГИБКИМИ МАГНИТНЫМИ ДИСКАМИ

В качестве внешней памяти ЭВМ "ИСКРА-226" (исполнение I) служит накопитель на гибких магнитных дисках [6]. Информация на гибких магнитных дисках (ГМД) хранится в виде файлов (наборов программ или наборов данных). Для работы с ГМД в языке БЭЙСИК предусмотрены дисковые операции [8, 10], которые позволяют работать с программами в двух режимах: режиме каталога файлов (18 операторов) и режиме адресации секторов (6 операторов).

В режиме каталога файлов (режим DC) можно создавать файлы программ и файлы данных на диске, а также обращаться к ним по имени файла. Режим DC обеспечивает преобразование данных из внутреннего формата, в котором они хранятся в памяти машины, во внешний формат для их хранения на гибком магнитном диске, запись и чтение данных в заданной последовательности, автоматический контроль заданной структуры файлов на диске.

В режиме адресации секторов используются два типа операций - DA и BA, которые обеспечивают прямой доступ к заданным секторам диска. Режим DA имеет те же форматы для хранения данных, что и режим DC, но пользователь при обращении к файлу каждый раз должен указывать номера секторов, где размещены считываемые данные. Режим BA позволяет работать с содержимым одного сектора.

В данном разделе рассматриваются дисковые операции режима каталога как основного режима работы с дисками. Сведения об операторах режимов DC, DA и BA приведены в табл.3. приложения.

#### 4.1. Подготовка диска к работе (оператор SCRATCH DISK)

Перед записью информации на новый диск или при появлении на уже бывшем в употреблении диске дефектных секторов диск необходимо форматировать, т.е. записать на нем чистые сектора и служебную информацию для контроля.

Для установки ИМД в дисковомод необходимо:

включить питание ИМД;

аккуратно вставить ИМД наклейкой кверху и к себе в свободный R или F карман дисковода;

закрывать дверцу дисковода, нажав ее ручку вниз.

Защита диска от записи устанавливается автоматически при включении питания или при нажатии на кнопку восстановления начального состояния.

Форматизация диска осуществляется командой:

```
PRINT HEX (1B000100000000) .
```

где в 6 позиции HEX-кода ставится 1, если ИМД установлен в кармане R, и 0 - если ИМД находится в кармане F.

После форматизации диска необходимо создать оглавление для размещения имен и адресов для всех файлов программ и файлов данных. Эта процедура выполняется оператором **SCRATCH DISK**.

Форма записи оператора:

```
SCRATCH DISK <тип диска> [ <устройство> , ] [ L S = <арифметическое  
выражение 1 > , ] E M P = <арифметическое выражение 2 > ,
```

где: <арифметическое выражение 1> - число секторов, которое отводится под зону указателя каталога (обычно от 5 до 10, по умолчанию - 24) ;

<арифметическое выражение 2> - число секторов, отводимое под зону каталога.

Для форматизации диска необходимо выполнить одну из программ.

Для верхнего R накопителя:

```
10 SELECT PRINT18  
20 PRINT HEX<1B000100000400>  
30 SCRATCH DISK RLS=10,END=999
```

Для нижнего F накопителя

```
10 SELECT PRINT18  
20 PRINT HEX<1B000000000400>  
30 SCRATCH DISK FLS=10,END=999
```

Программа выбирает в качестве консольного устройства ИМД (10-я строка) , производит форматизацию диска (20-я строка) , создает в секторах диска с 0 по 10 зону указателя каталога и до 999 сектора включительно - зону для размещения файлов программ и файлов данных (30-я строка) . Перед выполнением программы необходимо нажать на кнопку сброса защиты записи.

#### 4.2. Запись программы на диск (оператор `SAVE DC`)

Для записи программы, находящейся в памяти машины, на ИМД служит оператор `SAVE DC`.

Форма записи оператора:

`SAVE DC<тип диска>[n][<устройство>][(<арифметическое  
выражение>)] <"имя">` ,

где целая часть арифметического выражения определяет количество резервируемых секторов на случай последующего расширения программы; <"имя"> программы не должно превышать 8 символов.

Примеры:

`SAVE DCR(4)"GRAPHIC" ;`

`SAVE DCF(6)"СИСТЕМА" - запись с контролем .`

Перед посылкой команды в машину необходимо нажать на кнопку сорока защиты записи. Загорание на лицевой панели дисководов зеленой лампочки рядом с карманом, где находится диск, легкое пощелкивание дисководов в процессе записи и появление на экране знаков : свидетельствует о благополучной записи программы на диск. Для того, чтобы убедиться в этом, необходимо вызвать на экран оглавление библиотеки программ, хранящихся на магнитном диске, и найти среди них имя только что записанной программы.

#### 4.3. Вывод содержимого зоны указателя каталога (оператор `LIST DC`)

Оглавление содержимого диска выводится на экран или на печатающее устройство с помощью оператора `LIST DC`.



Форма записи оператора:

<тип диска>[<устройство>,) { {<символьная переменная>}  
{<символьная константа>} } },

где <символьная переменная> или <символьная константа> позволяют просматривать указатель каталога по "маске", т.е. по первой букве имени программы или по заданному сочетанию первых букв.

Примеры:

A. LIST DCF ;

B. LIST DCF "S" .

При выполнении первой команды (пример А) на заданное устройство выводится:

```
FIXED CATALOG
INDEX SECTORS = 00020
END CAT.AREA = 01000
CURRENT END = 00307

NAME      TYPE  START  END  USED
VETV1    P   00178 00183 00003
ЧЕРТЕЖ   P   00299 00307 00004
СИСТЕМ   P   00036 00049 00004
ТАБЛБОУЛ P   00354 00365 00006
СТАТИС   P   00111 00129 00009
. . . . . ,
```

где: **FIXED CATALOG** — каталог фиксированной длины;  
**INDEX SECTORS** — длина оглавления в секторах;  
**END CAT-AREA** — длина зоны каталога на диске;  
**CURRENT END** — текущий конец информации на диске;  
**NAME** — имя файла;  
**TYPE** — тип файла (P-программа, D-данные) ;  
**START** — номер сектора начала файла;  
**END** — номер сектора конца файла;  
**USED** — количество занятых файлом секторов.

При выполнении второй команды (пример В) на заданное устройство выводится информация о программах, имена которых начинаются на букву "S" :

NAME	TYPE	START	END	USED
SISTEM	P	00036	00049	00004
STATIS	P	00111	00129	00009
SUMMAC	P	00251	00259	00004
SUMMAB	P	00243	00250	00004
SORT	P	00050	00064	00005
SUMMAA	P	00235	00242	00004

#### 4.4. Чтение программы с диска (оператор LOAD DC )

Для загрузки программы или ее сегмента с ИМД в память машины служит оператор LOAD DC.

Форма записи оператора:

LOAD DC <тип диска>{устройство .} <имя>  
[<номер строки>] [, <номер строки>] ,

где параметры <номер строки> определяет начало и конец загружаемого сегмента программы.

Для загрузки программы необходимо:

вставить ИМД в свободный R или F карман дисководы;

очистить память машины от ранее введенных программ и данных (выполнить команду CLEAR) ;

набрать и выполнить команду загрузки, например:

```
LOAD DCF"ЧЕРТЕЖ"          CR/LF
```

Текст загруженной программы можно вызвать на экран командой LIST .

#### 4.5 . Повторная запись программы на диск ( оператор SCRATCH)

После редактирования, изменения или дополнения вызванной с ИМД программы ее снова можно записать на диск под но-

вым или тем же самым старым именем. Для этого предварительно необходимо ликвидировать старую программу, а на ее месте записать новую. Присвоение программе с указанным именем статуса "ликвидируемой" осуществляется оператором SCRATCH .

Форма записи оператора:

```
SCRATCH <тип диска> [<устройство> ,] <"СТАРое Имя  
ПРОГРАММЫ"> .
```

Однако эта команда не стирает программу на диске, она делает ее недоступной пользователю при последующих к ней обращениях.

Запись на месте ликвидируемой программы новой программы осуществляется оператором SAVE DC .

Форма записи оператора:

```
SAVE DC <тип диска> [<устройство> ,] (<"СТАРое Имя  
ПРОГРАММЫ">) <"НОВОЕ ИЛИ СТАРОЕ Имя ПРОГРАММЫ"> .
```

При повторной записи программы необходимо:

вставить ГМД в свободный R или F карман дисковода;

выполнить команду SCRATCH;

нажать на клавишу сброса защиты записи и

выполнить команду SAVE .

```
Пример: SCRATCH F"СИСТЕМА"          CR/LF  
         SAVE DCF("<СИСТЕМА">)"СИСТЕМА"  CR/LF
```

На месте ликвидируемой программы с именем "СИСТЕМА" записывается программа с тем же самым именем.

```
Пример: SCRATCH R"СИСТЕМА"          CR/LF  
         SAVE DCR("<СИСТЕМА">)"СИСТЕМА"  CR/LF
```

На месте ликвидируемой программы с именем "СИСТЕМА" записывается программа с именем "SISTEM".

При повторной записи программы на диск следует помнить,

что дисководы разных машин могут быть несовместимы, поэтому отладку программ и повторную их запись на ГМД желательно производить на одной и той же машине.

#### 4.6. Копирование информации с диска на диск (оператор COPY)

Копирование информации с диска на диск осуществляется оператором COPY.

Форма записи оператора:

$COPY \{ \langle \text{устройство} \rangle \} \left\{ \begin{array}{l} FR \\ RF \end{array} \right\} \langle \text{арифметическое выражение 1} \rangle ,$   
 $\langle \text{арифметическое выражение 2} \rangle$

где:  $\langle \text{арифметическое выражение 1} \rangle$  - задает адрес первого сектора копируемой области;

$\langle \text{арифметическое выражение 2} \rangle$  - задает адрес последнего сектора копируемой области.

Оба диска вставляются в верхний и нижний карманы дисковода, нажимается кнопка сброса защиты записи и задается одна из команд.

Для БЭЙСИК 01:

$COPY RF(0,1000)$  - копирование ведется с верхнего диска на нижний и копируется весь диск ;

$COPY FR(200,600)$  - копирование ведется с нижнего диска на верхний и копируются сектора с 200 по 600 .

Для БЭЙСИК 02:

$COPY R(0,1000) TO F$  - при копировании всего диска;

$COPY F(200,600) TO R$  - при копировании части диска.

При копировании информации с диска на диск файлы, отмеченные как ликвидируемые, не удаляются.

#### 4.7. Копирование каталога с диска на диск с удалением ликвидируемых файлов (оператор MOVE)

Для копирования каталога с диска на диск с удалением всех ликвидируемых файлов из зоны каталога и их имен из зоны указателя каталога со сжатием оставшихся файлов служит оператор **MOVE**.

Форма записи оператора:

$$\text{MOVE} \left[ \langle \text{устройство} \rangle \right] \left\{ \begin{array}{l} \text{FR} \\ \text{RF} \end{array} \right\} .$$

Примеры записи оператора.

Для языка БЭЙСИК 01:

**MOVE RF** — копирование ведется с верхнего диска на нижний;

**MOVE FR** — копирование ведется с нижнего диска на верхний.

Для языка БЭЙСИК 02:

**MOVE R TO F** — копирование ведется с верхнего диска на нижний;

**MOVE F TO R** — копирование ведется с нижнего диска на верхний.

#### 4.8. Изменение размера зоны каталога на диске (оператор MOVE END)

Для изменения ранее установленного конца зоны каталога на диске служит оператор **MOVE END**.

Форма записи оператора:

**MOVE END** <тип диска> [ <устройство> ] = <арифметическое выражение>, где <арифметическое выражение> — определяет адрес нового конечного сектора зоны каталога.

Пример:

```
MOVE END R 360
```

где 360 - адрес нового конечного сектора зоны каталога для диска R .

#### 4.9. Открытие файлов данных на запись (оператор DATA SAVE DC OPEN)

Для резервирования секторов на вновь создаваемый файл данных и занесения информации об этом файле в зону указателя каталога служит оператор DATA SAVE DC OPEN. Оператор может быть применен для повторного использования секторов файла, отмеченного как ликвидируемый.

Форма записи оператора:

```
DATA SAVE DC OPEN <тип диска> [n] [<номер файла>,  
  { <имя I>  
    <арифметическое выражение> } <имя> ,
```

где: <арифметическое выражение>-задает количество резервируемых под файл секторов;

<имя I> - ликвидируемый файл;

<имя> ~ новый файл.

Пример:

```
DATA SAVE DC OPEN R(150)"дан1" .
```

В результате выполнения команды на диске R будет создан новый файл с именем "дан1" размером в 150 секторов.

Если необходимо создать новый файл с именем "дан2" на месте старого файла с именем "дан1", то требуется выполнить две команды:

```
SCRATCH DISK R"дан1" ;
```

```
DATA SAVE DC OPEN R("дан1")"дан2"
```

На одном диске можно иметь несколько открытых файлов. В этом случае для каждого открытого файла во всех операторах режима DC нужно иметь свой номер вида #P (где P - число от 0 до 7).

#### 4.10. Запись данных на диск (оператор DATA SAVE DC)

Для записи данных из списка аргументов в ранее открытый файл служит оператор DATA SAVE DC .

Форма записи оператора:

$$\text{DATA SAVE DC} [\text{M}] [\text{<номер файла>}] \left\{ \begin{array}{l} \text{END} \\ \text{<список аргументов>} \end{array} \right\}$$

Наличие знака M в операторе указывает на то, что запись ведется с проверкой на считывание. Использование END в конце оператора позволяет формировать специальную запись конца файла с указанием количества занятых файлом секторов.

Примеры: DATA SAVE DC M, LK, KK) ;  
DATA SAVE DC #2, END .

#### 4.11. Закрытие файлов данных (оператор DATA SAVE DC CLOSE)

Для закрытия файла с номером 0 (по умолчанию), всех открытых файлов (ALL) или одного указанного файла служит оператор DATA SAVE DC CLOSE .

Форма записи оператора:

$$\text{DATA SAVE DC ' CLOSE} \left\{ \left[ \begin{array}{l} \text{ALL} \\ \text{<номер файла>} \end{array} \right] \right\}$$

где <номер файла> - # <логический номер> .

Примеры: DATA SAVE DC CLOSE #2

DATA SAVE DC CLOSE ALL

4.12. Открытие файлов данных на чтение  
(оператор DATA LOAD DC OPEN)

Для открытия файлов данных, записанных ранее в каталог файлов, а также для повторного открытия файлов служит оператор DATA LOAD DC OPEN

Форма записи оператора:

```
DATA LOAD DC OPEN <тип диска> [<номер файла>] {TEMP,  
<адрес начального сектора>, <адрес конечного сектора>}  
<имя>
```

Параметр TEMP используется для открытия временного рабочего файла.

Примеры: DATA LOAD DC OPEN R "ДАННЫЕ" ;

```
DATA LOAD DC OPEN F TEMP 20,200 .
```

4.13. Чтение данных с диска (оператор DATA LOAD DC)

Для считывания логических записей из файла каталога на диске с последующим присвоением считанных значений переменным или массивам из списка аргументов служит оператор DATA LOAD DC .

Форма записи оператора :

```
DATA LOAD DC <номер файла>, <список аргументов>  
<номер файла> ,
```

где <номер файла> - ~~№~~ <логический номер> .

Примеры:

```
DATA LOAD DCR A, B, C <> ;
```

```
DATA LOAD DC #1, "ДАННЫЕ" .
```



#### 4.14. Переход по концу записи файла (оператор `IF END THEN` )

Для анализа состояния текущего файла данных в процессе считывания служит оператор `IF END THEN`.

Форма записи оператора:

`IF END THEN <номер строки> .`

Пример: `IF END THEN 460 .`

При считывании закрывающей записи файла управление будет передано 460 строке программы.

#### 4.15. Получение информации о файле (оператор `LIMITS` )

Для получения информации о местоположении файла и количестве занимаемых им секторов служит оператор `LIMITS`.

Форма записи оператора:

`LIMITS <тип диска> [<номер файла> ,] [<имя> , <цифровая  
переменная 1> , <цифровая переменная 2> ,  
<цифровая переменная 3> ,`

где: <цифровая переменная 1> - переменная, куда заносится начальный адрес сектора в файле;

<цифровая переменная 2> - переменная, куда заносится конечный адрес сектора в файле;

<цифровая переменная 3> - переменная, куда заносится количество занимаемых файлом секторов .

Пример:

`LIMITS R"DAN1", S1, S2, S3 .`

#### 4.16. Перемещение вперед в файле (оператор `DSKIP` )

Для пропуска определенного количества секторов или логических записей в файле служит оператор `DSKIP`.



<арифметическое выражение 2> - номер конечного сектора, которым заканчивается проверка.

Если в операторе параметры <арифметическое выражение> не заданы, то проводится проверка всех секторов в диапазоне  $0 + 1000$ .

Примеры:

VERIFY R - проводится проверка всех секторов диска R;

VERIFY F(370,640) - проверяются сектора диска F с 370 по 640.

При обнаружении ошибки в процессе проверки на экран выдается сообщение, например ERROR IN SECTOR 510, где 510 - номер ошибочного сектора.

Ниже приводится программа копирования информации с верхнего диска на нижний с контролем и выводом на печать копируемых секторов:

```
FOR I=10 TO 500: COPY R(I,I) TO F(I):  
PRINT I: NEXT I: VERIFY F(10,500)
```

Сведения о дисковых операциях режима адресации секторов DA и BA приведены в табл.3 приложения и в [8,10].

## Системные команды ЭМ "ИСКРА-226"

№ п/п	Имя оператора	Назначение оператора	Примеры
1	CLEAR	Очистка памяти машины	CLEAR CLEAR P CLEAR P 50,200 CLEAR V CLEAR N
2	CONTINUE	Продолжение работы прерванной прог- раммы	CONTINUE
3	HALT/STEP	Пошаговое выполне- ние программы	HALT/STEP
4	LIST	Вывод текста прог- раммы	LIST LIST 60,400 LIST S LIST 50,200 LISTY T1,T2,T3 LIST 24
5	RENUMBER	Перенумерация строк прог- раммы	RENUMBER RENUMBER 50,150,5
6	RESET	Аварийный останов	RESET
7	RUN	Исполнение прог- раммы	RUN RUN 100
8	TRACE	Выполнение режима слежения за перехо- дами в программе	TRACE
9	TRACE OFF	Выход из режима слежения	TRACE OFF
10	SELECT	Выбор режима работы	SELECT CO 00 120 SELECT LIST 05 SELECT PRINT 00 (60) SELECT P6 SELECT R SELECT D
11	STMT NUMBER	Автоматическая нуме- рация строк прог- раммы	STMT NUMBER

Таблица 2

Сообщения об ошибках

ОШИБКА	ПРИЧИНА ВОЗНИКНОВЕНИЯ ОШИБКИ	
СБОЙ СИСТЕМЫ	ПЕРЕХОД ЧЕРЕЗ НУЛЕВОЙ АДРЕС	
СБОЙ MO	СБОЙ МАТОВЕСПЕЧЕНИЯ	
СБОЙ ОЗУ	СБОЙ КОНТРОЛЬНОГО РАЗРЯДА ОЗУ	
СБОЙ УП	СБОЙ КОНТРОЛЬНОГО РАЗРЯДА УП	
ERR 01	ТЕКСТОВОЕ ПЕРЕПОЛНЕНИЕ	
ERR 02	ПЕРЕПОЛНЕНИЕ ТАБЛИЦ	
ERR 03	МАТЕМАТИЧЕСКАЯ ОШИБКА	
ERR 04	ПЕРЕМЕННАЯ НЕ ОПРЕДЕЛЕНА	
ERR 06	СИНТАКСИЧЕСКАЯ ОШИБКА ПРИ ВВОДЕ С КЛАВИАТУРЫ	
ERR 07	СИНТАКСИЧЕСКАЯ ОШИБКА ПРИ ВВОДЕ С ДИСКА/ЛЕНТЫ	
ERR 08	ФУНКЦИЯ FN НЕ ОПРЕДЕЛЕНА	
ERR 09	НЕСООТВЕТСТВИЕ ПАРАМЕТРОВ DEFFN 'И GOSUB '	
ERR 10	СИНТАКСИЧЕСКАЯ ОШИБКА ПРИ ВВОДЕ ПРОГРАММЫ ИЗ ПЕРЕМЕННОЙ	
ERR 11	НЕСУЩЕСТВУЮЩИЙ НОМЕР СТРОКИ	
ERR 12	НЕПРАВИЛЬНЫЙ ОПЕРАТОР	
ERR 13	В СТРОКЕ ОТСУТСТВУЕТ DATA ДЛЯ RESTORE	
ERR 14	ОШИБКА В ТРАНСЛЯЦИИ ПРИ ВЫПОЛНЕНИИ ОПЕРАТОРА ASMB:	
	НЕПРАВИЛЬНАЯ ИНСТРУКЦИЯ	01
	НЕВЕРНО ЗАДАН ТИП КОНСТАНТЫ	02
	МЕТКА ОПИСАНА ДВАЖДЫ	04
	ОШИБКА В ЗАДАНИИ КОНСТАНТЫ	05
	ИСПОЛЬЗОВАН НЕОПИСАННЫЙ ИДЕНТИФИКАТОР	06
	ПЕРЕПОЛНЕНИЕ ОЗУ ИЛИ ОП	07
	ИСПОЛЬЗОВАНИЕ КОНСТАНТЫ K > 15	08
	КОНСТАНТА ИЛИ НОМЕР ЯЧЕЙКИ > 255	09
	НЕДОПУСТИМЫЙ АДРЕС БЕЗУСЛОВНОГО ПЕРЕХОДА	10
	НЕДОПУСТИМЫЙ АДРЕС УСЛОВНОГО ПЕРЕХОДА	11
	КОНСТАНТА > B17777	12
	АВТОКОДНАЯ ПРОГРАММА НЕ ЗАКАНЧИВАЕТСЯ END	17
	ОБЪЕМ ВЫДЕЛЕННОЙ РАБОЧЕЙ ПАМЯТИ НЕДОСТАТОЧЕН ДЛЯ ТРАНСЛЯЦИИ ПРОГРАММЫ	20
	ОБЪЕМ ВЫДЕЛЕННОЙ ОБЛАСТИ ПОД ОБЪЕКТНУЮ ПРОГРАММУ НЕДОСТАТОЧЕН	21
ERR 15	НЕПРАВИЛЬНЫЙ НОМЕР СТРОКИ ИЛИ ДИАПАЗОНА СТРОК	
ERR 16	НЕТ АССЕМБЛЕРА	
ERR 18	НЕДОПУСТИМАЯ ВЕЛИЧИНА (ЧИСЛА) РАЗМЕРНОСТИ	

	ИНДЕКСА)
ERR 20	НЕДОПУСТИМЫЙ ФОРМАТ ЧИСЛА (ПОРЯДОК БОЛЬШЕ 99)
ERR 22	НЕОПРЕДЕЛЕННЫЙ МАССИВ
ERR 23	ОТСУТСТВИЕ ПРОГРАММЫ В ПАМЯТИ
ERR 24	НЕПОРЯДОЧЕННЫЙ СПИСОК ДАННЫХ
ERR 25	НЕДОПУСТИМОЕ ИСПОЛЬЗОВАНИЕ ОПЕРАТОРОВ GOSUB/RETURN, FOR/NEXT
ERR 27	ДААННЕ ИСЧЕРПАНЫ ИЛИ НАХОДЯТСЯ ЗА ДОПУСТИМЫМИ ПРЕДЕЛАМИ
ERR 29	НЕДОПУСТИМЫЙ ФОРМАТ ДАННЫХ В ОПЕРАТОРЕ INPUT
ERR 31	НОВЫЙ НОМЕР СТРОКИ ПО ОПЕРАТОРУ RENUMBER > 9999
ERR 32	ДЛИНА ВЫГРУЖАЕМОЙ ПРОГРАММЫ БОЛЬШЕ ДЛИНЫ ПЕРЕМЕННОЙ
ERR 36	ОШИБОЧНЫЙ ФОРМАТ В ОПЕРАТОРАХ CONVERT, PACK, UNPACK
ERR 37	ДЛЯ ОПЕРАТОРА PRINTUSING НЕТ ОПЕРАТОРА IMAGE
ERR 40	НАРУШЕНА ВОЗРАСТАЮЩАЯ ПОСЛЕДОВАТЕЛЬНОСТЬ ИДЕНТИФИКАТОРОВ ПЕРЕМЕННЫХ В ОПЕРАТОРЕ LIST V
ERR 41	НЕДОПУСТИМЫЙ АРГУМЕНТ STR
ERR 43	НЕДОПУСТИМОЕ ПРИСВОЕНИЕ
ERR 44	ПРОГРАММА ЗАЩИЩЕНА
ERR 45	ДЛИНА ОПЕРАТОРА > 255
ERR 46	НОВЫЙ НАЧАЛЬНЫЙ НОМЕР СТРОКИ В ОПЕРАТОРЕ RENUMBER МАЛ
ERR 47	НЕДОПУСТИМОЕ ЗАДАНИЕ ФАУ
ERR 48	ОТСУТСТВУЕТ ОПЕРАТОР DEFFN / С УКАЗАННЫМ НОМЕРОМ
ERR 49	НЕКВАДРАТНАЯ МАТРИЦА
ERR 50	НЕПРАВИЛЬНЫЕ ПАРАМЕТРЫ МАТРИЦ
ERR 51	НЕВЕРНЫЙ МАТРИЧНЫЙ АРГУМЕНТ В ОПЕРАТОРАХ УМНОЖЕНИЯ МАТРИЦ
ERR 52	ИНВЕРТИРОВАНИЕ СИНГУЛЯРНОЙ МАТРИЦЫ
ERR 53	ДЛИНА РАСПАКОВЫВАЕМОГО МАССИВА МАЛА
ERR 54	ОШИБОЧНАЯ ДЛИНА СТРОКИ УСТРОЙСТВА В ОПЕРАТОРЕ SELECT
ERR 55	ОШИБОЧНЫЙ АДРЕС СЕКТОРА В ОПЕРАТОРАХ COPY, VERIFY
ERR 56	ЧИСЛО ПРЕВЫШАЕТ ФОРМАТ В ОПЕРАТОРАХ PACK, CONVERT
ERR 58	ДЛЯ ОПЕРАТОРА LOAD DA ТРЕБУЕТСЯ ОБЩЕНАЯ ПЕРЕМЕННАЯ
ERR 59	ДЛИНА ПЕРЕМЕННОЙ МАЛА
ERR 60	ФАЙЛ НЕ ОТКРЫТ
ERR 66	ОШИБКА ПО К.С. ПРИ СЧИТЫВАНИИ
ERR 67	НЕПРАВИЛЬНЫЙ ТИП ЗАПИСИ ПРОГРАММА/ДААННЕ

ERR 68	БУФЕР ВВОДА/ВЫВОДА МАЛ	
ERR 69	ПЕРЕМЕННАЯ/МАССИВ НЕ ПОМЕЩАЕТСЯ В БУФЕР ВВОДА/ВЫВОДА	
ERR 70	В ОПЕРАТОРЕ PRINT USING НЕТ ФОРМАТА	
ERR 71	ФАЙЛ УЖЕ ЕСТЬ В КАТАЛОГЕ	
ERR 72	ФАЙЛ ВЫЧЕРКНУТ	
ERR 73	ФАЙЛ ОТСУТСТВУЕТ В КАТАЛОГЕ	
ERR 74	АДРЕС СЕКТОРА ВНЕ ФАЙЛА	
ERR 75	КОНЕЦ КАТАЛОГА	
ERR 76	ФАЙЛ ЗАПОЛНЕН	
ERR 77	ВРЕМЕННЫЙ ФАЙЛ НАХОДИТСЯ НЕ ЗА ОБЛАСТЬЮ КАТАЛОГА	
ERR 78	УКАЗАТЕЛЬ КАТАЛОГА ЗАПОЛНЕН	
ERR 80	ОШИБКА ВВОДА/ВЫВОДА:	
	- АВАРИЯ ВВУ (ДИСК, ЛЕНТА)	01
	- НАРУШЕНИЕ ПОСЛЕДОВАТЕЛЬНОСТИ ПРОЦЕДУР	02
	- УВВ НЕ ГОТОВО К ОБМЕНУ	03
	- ЗАЩИТА ОТ ЗАПИСИ	05
	- МАРКЕР НАЧАЛА (КОНЦА) ЛЕНТЫ	06
	- ОШИБКА ПО К.С. КОНТРОЛЬНОГО ЧТЕНИЯ ПОСЛЕ ЗАПИСИ	07
	- АВАРИЯ БИФ	08
	- ОШИБКА ПО К.С. ПРИ СЧИТЫВАНИИ	0A
	- ОШИБКА ПРОЦЕДУРЫ (КОДИРОВАНИЕ, ФОРМАТ, К.С. ОБМЕНА ПИД И БИФ)	0B
	- СЕКТОР С УКАЗАННЫМ АДРЕСОМ НЕ НАЙДЕН	0C
	- НАРУШЕНИЕ ПОСЛЕДОВАТЕЛЬНОСТИ ИНТЕРФЕЙСНЫХ КОМАНД В ПРОЦЕДУРЕ	0D
	- НЕДОПУСТИМЫЙ АДРЕС СЕКТОРА	0E
	- ОШИБОЧНАЯ ДЛИНА МАССИВА	0F

Операторы и Команды работы с дисковыми устройствами

№	ИМЯ ОПЕРАТОРА	НАЗНАЧЕНИЕ ОПЕРАТОРА	ПРИМЕРЫ ЗАПИСИ
1	COPY	КОПИРОВАНИЕ ИНФОРМАЦИИ С ДИСКА НА ДИСК	COPY FR(0,500) (ДЛЯ БЭИСИК 01) COPY F(0,500) TO R1 (ДЛЯ БЭИСИК 02)
2	DATA LOAD BA	СЧИТЫВАНИЕ ИНФОРМАЦИИ ИЗ СЕКТОРА ДИСКА В СИМВОЛЬНУЮ ПЕРЕМЕННУЮ (МАССИВ)	DATA LOAD BAF("50" ",S)A*
3	DATA LOAD DA	СЧИТЫВАНИЕ ЛОГИЧЕСКИХ ЗАПИСЕЙ, НАЧИНАЯ С УКАЗАННОГО СЕКТОРА КАТАЛОГА	DATA LOAD DAR(100 ,L)A(),B*()
4	DATA LOAD DC	ЧТЕНИЕ ЛОГИЧЕСКИХ ЗАПИСЕЙ ИЗ ФАЙЛА КАТАЛОГА	DATA LOAD DCR "ДАННЕ"
5	DATA LOAD DC OPEN	ОТКРЫТИЕ ФАЙЛОВ ДАННЫХ, ЗАПИСАННЫХ РАНЕЕ В КАТАЛОГ	DATA LOAD DC OPEN F"DATA"
6	DATA SAVE BA	ЗАПИСЬ СИМВОЛЬНОЙ ИНФОРМАЦИИ В СЕКТОР ДИСКА	DATA SAVE BA F*/ IC,(K,K)A*()
7	DATA SAVE DA	ЗАПИСЬ ДАННЫХ ИЗ СПИСКА АРГУМЕНТОВ В ЗАДАННЫЙ СЕКТОР	DATA SAVE DA R/10 ,(C*,C*),A,T
8	DATA SAVE DC	ЗАПИСЬ ДАННЫХ ИЗ СПИСКА АРГУМЕНТОВ НА ДИСК	DATA SAVE DC M, L*,K()
9	DATA SAVE DC OPEN	РЕЗЕРВИРОВАНИЕ СЕКТОРОВ ДЛЯ НОВОГО ФАЙЛА	DATA SAVE DC OPEN R(150)"DAN1" DATA SAVE DC OPEN F("DAN1")"DAN2"
10	DATA SAVE DC CLOSE	ЗАКРЫТИЕ ФАЙЛА	DATA SAVE DC CLOSE #2
11	DSPACE	ВОЗВРАТ НАЗАД НА ЗАДАННОЕ КОЛИЧЕСТВО СЕКТОРОВ	DSPACE BEG DSPACE KZ S DSPACE Y
12	DSKIP	ПРОПУСК ЗАДАННОГО КОЛИЧЕСТВА СЕКТОРОВ	DSKIP 50 DSKIP ERD DSKIP 50S



13	IF END THEN	АНАЛИЗ СОСТОЯНИЯ ТЕКУЩЕГО ФАЙЛА ДААННЫХ	IF END THEN 520
14	LIMITS	ПОЛУЧЕНИЕ ИНФОР- МАЦИИ О ФАЙЛЕ (МЕСТОПОЛОЖЕНИЕ, КОЛИЧЕСТВО ЗАНИ- МАЕМЫХ СЕКТОРОВ)	LIMITS F"ДААННЫЕ", S1,S2,S3
15	LIST DC	РАСПЕЧАТКА СОДЕР- ЖИМОГО ЗОНЫ УКА- ЗАТЕЛЯ КАТАЛОГА	LIST DCF LIST DCR A*
16	LOAD DA	ЗАГРУЗКА ПРОГРАМ- МЫ ИЛИ ЕЕ ЧАСТИ В ПАМЯТЬ МАШИНЫ	LOAD DA F(40,A) LOAD DA R#4,(K*, C)300,500 LOAD DA R#5,(A, A),400
17	LOAD DC	ЗАГРУЗКА ПРОГ- РАММЫ ИЛИ ЕЕ СЕГ- МЕНТА С ЗАДАНЫМ ИМЕНЕМ В ПАМЯТЬ МАШИНЫ	LOAD DCF"STATIS" LOAD DCR"ПРОГР1" LOAD F#1
18	MOVE	КОПИРОВАНИЕ КАТА- ЛОГА С ДИСКА НА ДИСК	MOVE FR(БЭИСИК-1) MOVE F TO R (БЭИСИК-2)
19	MOVE END	ИЗМЕНЕНИЕ РАЗМЕРА ЗОНЫ КАТАЛОГА	MOVE END F=500 MOVE END R=400
20	SAVE DA	ЗАПИСЬ ПРОГРАММЫ (СЕКМЕНТА) НА УКАЗАННОЕ МЕСТО ДИСКА	SAVE DAR*(A*,A*) 100
21	SAVE DC	ЗАПИСЬ ПРОГРАММЫ (СЕКМЕНТА) НА ЗА- ДАНЫМ ДИСК	SAVE DCF*(5)"FR1" SAVE DCR*(5)"FR2"
22	SCRATCH	ПРИСВОЕНИЕ ФАЙЛАМ ПРОГРАММЫ СТАТУСА ЛИКВИДИРУЕМЫХ	SCRATCH F"FR1" SCRATCH R"FR2"
23	SCRATCH DISC	РЕЗЕРВИРОВАНИЕ СЕКТОРОВ ДИСКА ДЛЯ ЗОНЫ УКАЗАТЕ- ЛЯ КАТАЛОГА И ЗО- НЫ КАТАЛОГА	SCRATCH DISK FLS=5,END=999
24	VERIFY	ПРОВЕРКА СОСТОЯ- НИЯ СЕКТОРОВ ДИС- КА	VERIFY F VERIFY R(200,450)

ЛИТЕРАТУРА

1. Машина вычислительная электронная клавишная программно-управляемая "ИСКРА-226". Инструкция по программированию. Базовый объем. I.320.136 ДИ4-1. Часть I. Курск: з-д Счетмаш, 1983.
2. Можара В.Д. Конструкция и ремонт ЭКЭМ "ИСКРА-226". М.: Машиностроение, 1986. 224 с.
3. Процессор интерпретирующий диалоговый "ИСКРА-226". Техническое описание и инструкция по эксплуатации. 3.050.206 ТО. Курск: з-д Счетмаш, 1983.
4. Блок отображения символьно-графической информации БОСГИ-1920. Техническое описание и инструкция по эксплуатации. 5.104.166 ТО. Курск: з-д Счетмаш, 1983.
5. Устройство клавишное "Искра 007-31". Техническое описание и инструкция по эксплуатации. 3.040.313 ТО. Курск: з-д Счетмаш, 1983.
6. Накопитель на гибком магнитном диске "ИСКРА 005-51". Техническое описание и инструкция по эксплуатации. 3.060.000 ТО. Курск: з-д Счетмаш, 1983.
7. Машина вычислительная электронная клавишная программно-управляемая "ИСКРА-226". Инструкция по программированию. Базовый объем. I.320.136 ДИ4-1. Часть II. Курск: з-д Счетмаш, 1983.
8. Диски. Инструкция по программированию. I.320.136 ДИ4-2. Курск: з-д Счетмаш, 1983.
9. Нагинаев В.Н., Чернухин С.И., Шахуняц Т.Г. ЭМ "ИСКРА-226". Основные режимы работы. Методические указания к лабораторным работам. М.: МИИТ, 1985. -32 с.
10. Справочное руководство по языку БЭЙСИК 02. ВШ. -ИД-36640. - 788 с. с ил. Материал фирмы WANG LABORATORIES INC., США /1979, 451 р./.

ОГЛАВЛЕНИЕ

ВВЕДЕНИЕ . . . . .	3
1. СОСТАВ И ХАРАКТЕРИСТИКИ ЭВМ "ИСКРА-226". . . . .	3
1.1. Процессор интерпретирующий диалоговый (ПИД) . . . . .	4
1.1.1. Процессор интерпретирующий (ПИ) . . . . .	4
1.1.2. Блок отображения символично-графической информации (БОСГИ) . . . . .	5
1.1.3. Устройство клавишное (УК) . . . . .	8
1.2. Накопитель на гибких магнитных дисках (НГМД) . . . . .	14
1.3. Печатающее устройство (ПУ) . . . . .	14
2. СИСТЕМНЫЕ КОМАНДЫ И РЕЖИМЫ РАБОТЫ ЭВМ "ИСКРА-226" . . . . .	16
2.1. Адрессация устройств ввода/вывода . . . . .	16
2.2. Выбор режима работы машины (команда SELECT) . . . . .	17
2.3. Очистка памяти машины (команда CLEAR) . . . . .	19
2.4. Вывод текста программы (команда LIST) . . . . .	20
2.5. Автоматическая нумерация строк программы (команда STMT NUMBER) . . . . .	21
2.6. Перенумерация строк программы (команда RENUMBER) . . . . .	21
2.7. Исполнение программы (команда RUN) . . . . .	22
3. ВЫПОЛНЕНИЕ ПРОГРАММЫ НА ЭВМ "ИСКРА-226" . . . . .	23
3.1. Структура программы . . . . .	23
3.2. Набор текста программы . . . . .	24
3.3. Классификация ошибок . . . . .	25
3.4. Исправление ошибок в программе . . . . .	27
3.5. Редактирование текста программы . . . . .	28
3.6. Отладка программы . . . . .	30
3.6.1. Режим слежения TRACE . . . . .	31
3.6.2. Режим пошаговой отладки программы HALT/STEP . . . . .	32

3.6.3. Совместное использование команды TRACE и клавиши HALT/STEP при отладке программы . . .	33
3.7. Выполнение счета по программе . . . . .	35
4. ДИСКОВЫЕ ОПЕРАЦИИ И РАБОТА С ГИБКИМИ МАГНИТНЫМИ ДИСКАМИ . . . . .	36
4.1. Подготовка диска к работе(оператор SCRATCH DISK)	37
4.2. Запись программы на диск (оператор SAVE DC) . .	39
4.3. Вывод содержимого зоны указателя каталога (оператор LIST DC) . . . . .	39
4.4. Чтение программы с диска (оператор LOAD DC) .	41
4.5. Повторная запись программы на диск (оператор SCRATCH) . . . . .	41
4.6. Копирование информации с диска на диск (оператор COPY) . . . . .	43
4.7. Копирование каталога с диска на диск с удалением ликвидируемых файлов (оператор MOVE)	44
4.8. Изменение размера зоны каталога на диске (оператор MOVE END) . . . . .	44
4.9. Открытие файлов данных на запись (оператор DATA SAVE DC OPEN) . . . . .	45
4.10. Запись данных на диск (оператор DATA SAVE DC)	46
4.11. Закрытие файлов данных (оператор DATA SAVE DC CLOSE) . . . . .	46
4.12. Открытие файлов данных на чтение (оператор DATA LOAD DC OPEN) . . . . .	47
4.13. Чтение данных с диска (оператор DATA LOAD DC)	47
4.14. Переход по концу записи файла (оператор IF END THEN) . . . . .	48
4.15. Получение информации о файле (оператор LIMITS)	48
4.16. Перемещение вперед в файле (оператор BSkip)	48

- 4.17. Перемещение назад в файле  
(оператор **BACK SPACE**) . . . . . 49
- 4.18. Контроль за состоянием записей на диске  
(оператор **VERIFY**) . . . . . 49

ПРИЛОЖЕНИЕ

- Таблица 1. Системные команды ЭВМ "ИСКРА-226". . . . . 51
- Таблица 2. Сообщения об ошибках . . . . . 52
- Таблица 3. Операторы и команды работы с  
дисковыми устройствами . . . . . 55

ЛИТЕРАТУРА . . . . . 57

ГЕОРГИЙ МИТРОФАНОВИЧ КУРБАТОВ

РАБОТА ЗА ПУЛЬТОМ ЭВМ "ИСКРА-226"

Методические указания по дисциплине

"Программирование на ЭВМ ИСКРА-226"

для

слушателей факультета повышения квалификации,  
преподавателей и студентов

Редактор И.С. Громыкина

Технический редактор Н.Н. Васильева

Корректор М.Б. Остапович

---

Печать офсетная

Подписано к печати **7.04.87**

Формат 60x90 1/16. Усл.печ.л. 3,75 Уч.-изд.л. 1,5 Тираж 300 экз.

Изд. № **101**

Заказ № **639**

Бесплатно.

---

Редакционно-издательский отдел МИИТа

101475, Москва, А-55, ул. Обрезцова, 15

Типография МИИТа