

**ИНФОРКОМ**

**ПЕРСОНАЛЬНЫЙ КОМПЬЮТЕР ZX-SPECTRUM  
Ю. ПОМОРЦЕВ**



**TR-DOS**

**для профессионалов и любителей**



**Москва - 1994**

**PDF version by Deny (Денисенко Д.А.)**

**e-mail: [DenyDA@mail.ru](mailto:DenyDA@mail.ru)**

**2007**

Книга содержит углубленное описание операционной системы TR-DOS и предназначена для тех, кто пишет программы в машинном коде для компьютеров ZX-Spectrum.

Для первоначального знакомства с TR-DOS читателю целесообразно изучить фирменное "Руководство Пользователя", в котором подробно описана система команд и другая аналогичная информация.

Некоторые сведения из этого руководства приведены ниже в виде краткой справки.

**АВТОР**

## СОДЕРЖАНИЕ

<b>ВВЕДЕНИЕ .....</b>	<b>4</b>
КРАТКИЙ ПЕРЕЧЕНЬ КОМАНД TR-DOS .....	4
КОДЫ ОШИБОК ПРИ ВЫЗОВЕ ИЗ BASIC .....	5
ДИСКИ И ДИСКОВОДЫ .....	5
СТРУКТУРА ДИСКЕТЫ TR-DOS .....	6
СТРУКТУРА ЗАГОЛОВКА ФАЙЛА В КАТАЛОГЕ .....	6
СТРУКТУРА СИСТЕМНОГО СЕКТОРА .....	7
ЗАМЕЧАНИЯ О ДРУГИХ ВЕРСИЯХ TR-DOS .....	8
ВХОДНЫЕ ТОЧКИ TR-DOS .....	8
НЕКОТОРЫЕ СИСТЕМНЫЕ ПЕРЕМЕННЫЕ TR-DOS v5.01 .....	15
ОБЛАСТЬ СВЯЗИ С ДИСКОВЫМ ФАЙЛОМ .....	18
ЗАПИСЬ ПРИ ПОМОЩИ "MAGIC KEY" .....	18
ОСОБЕННОСТИ ФОРМАТИРОВАНИЯ TR-DOS .....	20
ДИСКОВЫЕ ПРОГРАММЫ .....	21
SPECTRUM И IBM PC .....	23
<b>ПРИЛОЖЕНИЯ .....</b>	<b>24</b>
АДРЕСА ПОРТОВ ДИСКОВОГО КОНТРОЛЛЕРА "БЕТА" .....	24
ОПИСАНИЕ БИС КОНТРОЛЛЕРА НГМД 1818 ВГ 93 .....	24
НАЗНАЧЕНИЕ ВЫВОДОВ МИКРОСХЕМЫ .....	25
СИСТЕМА КОМАНД БИС КОНТРОЛЛЕРА 1818 ВГ 93 .....	26
СТРУКТУРА РЕГИСТРА СОСТОЯНИЯ 1818 ВГ 93 .....	28
ФОРМАТ ЗАПИСИ НА ДИСК .....	29
ПРИМЕР ФОРМАТА МАССИВА ДАННЫХ 1-го СЕКТОРА ДЛЯ ДВОЙНОЙ ПЛОТНОСТИ .....	29
ПРИМЕР ФОРМАТА МАССИВА ДАННЫХ 1-го СЕКТОРА ДЛЯ ОДИНАРНОЙ ПЛОТНОСТИ .....	30
СЛУЖЕБНЫЕ БАЙТЫ ИНДЕКСНОЙ ОБЛАСТИ .....	30
КОНТАКТЫ СОЕДИНИТЕЛЯ ДИСКОВОДА .....	31

## ВВЕДЕНИЕ

В последнее время с компьютерами ZX SPECTRUM получил широкое распространение BETA-диск интерфейс с дисковой операционной системой TR-DOS (версии 5.0x) фирмы Technology Research Ltd. Эта система коренным образом изменяет возможности старого доброго SPECTRUMa, превращая его в весьма эффективный инструмент.

Самые краткие характеристики TR-DOS сводятся к следующим:

ЛЕГКИЙ И БЫСТРЫЙ ДОСТУП	– загрузка любой программы, имеющейся на диске, происходит за несколько секунд;
ШИРОКИЙ ВЫБОР ДИСКОВОДОВ	– 5.25" или 3.5", 40 или 80 дорожек, одно- или двухсторонние (см. далее);
ЕМКОСТЬ	– до 4 дисководов одновременно. Данные хранятся с двойной плотностью, что дает суммарную емкость до 2.5 МВ;
ГИБКОСТЬ	– интерфейс независим, Вы можете использовать дисководы с другими компьютерами;
СОВМЕСТИМОСТЬ	– интерфейс можно использовать с компьютерами SPECTRUM, SPECTRUM+ и SPECTRUM 128;
АВТОСТАРТ	– автоматически запускает BASIC-программу, "boot" после включения питания или сброса (только в SPECTRUM 48);
АВТОМАТИЧЕСКАЯ ПРОВЕРКА	– автоматически проверяет и согласовывает систему с параметрами дисковода;
"ВОЛШЕБНАЯ" КЛАВИША	– благодаря ей вы можете сбросить на дискету текущую программу (см. далее);
СИСТЕМНЫЙ ПЕРЕКЛЮЧАТЕЛЬ	– дает возможность отключать интерфейс для стандартной конфигурации или используется для сброса;
РАБОТА С ФАЙЛАМИ	– TR-DOS работает со всеми типами файлов, имеющимися у SPECTRUMa - программами (B), кодами (C), числовыми и знаковыми массивами (D), и, кроме того, с дисковыми файлами с последовательным и произвольным доступом, тип которых обозначается как (#);
СИНТАКСИС	– самый простой, с использованием ключевых слов;

TR-DOS содержится в ППЗУ и использует только 128 байтов ОЗУ.

## КРАТКИЙ ПЕРЕЧЕНЬ КОМАНД TR-DOS

*"A : "	COPY B	OPEN#
*"B : "	FORMAT	PEEK
*"C : "	GO TO	POKE
*"D : "	ERASE	PRINT#
40	LIST	RANDOMIZE USR 15616
80	LIST#3	RANDOMIZE USR 15619:REM:
CAT	LOAD	RETURN
CAT#3	INPUT#	RUN
CLOSE#	MERGE	SAVE
COPY	MOVE	VERIFY
COPY s	NEW	

Примеры:

**FORMAT "\$NAME"** – форматирование диска, как одностороннего.

**COPY "новый", "старый" CODE** – копирование на одном диске.

**COPY s "имя"** – копировать на другой диск на одном дисководе.

**COPY "A:имя", "B:имя"** – копирование файла с дисковода B: на A:.

**COPY "B:\*", "A:\*"** – копирование всех файлов с дисковода A: на B:.

**NEW "новый", "старый"** – переименование файла (только в DOS).

**GO TO "\$имя"** – запуск MAGIC-CODE – файла с прерываниями IM2.

**PEEK "имя" адрес, сектор** – чтение сектора N из файла в память.

**POKE "имя" адрес, сектор** – запись из памяти в файл сектора N (N от единицы до числа, указанного в CAT для данного файла).

**OPEN #N, "имя", W** – открыть последовательный дисковый файл для записи (WRITE), с номером канала N от 4 до 15.

**OPEN #4, "имя", R** – то же, но для чтения (READ).

**PRINT #4; "ТЕКСТ"** – вывод в дисковый файл с номером канала 4.

**INPUT #4; T\$** – ввод из файла в строковую переменную.

**OPEN #N, "имя", RND, длина\_записи** – открыть дисковый файл с произвольным доступом (запись и чтение).

**PRINT #4; номер\_записи, T\$** – вывод T\$ в файл с произвольным доступом.

**INPUT #N; (номер\_записи), T\$** – ввод записи файла в переменную.

**CLOSE #N** – закрыть дисковый файл с номером N (обязательно!).

## ***КОДЫ ОШИБОК ПРИ ВЫЗОВЕ ИЗ BASICa***

Коды ошибок можно получить из DOS - переменной 23823 или через BASIC-переменную (например, err) следующим образом:

```
LET err=USR 15619 : REM : команда DOS
```

Значение кодов ошибки:

- |    |  |
|----|--|
| 0  | - нет ошибок                                   |
| 1  | - нет файла                                    |
| 2  | - файл уже существует                          |
| 3  | - нет места на диске                           |
| 4  | - каталог переполнен (уже записано 128 файлов) |
| 5  | - переполнение номера записи                   |
| 6  | - нет диска                                    |
| 7  | - ошибка на диске                              |
| 8  | - ошибка синтаксиса                            |
| 10 | - канал уже открыт                             |
| 11 | - диск не форматирован                         |
| 12 | - канал не открыт                              |

Иногда полезной бывает конструкция, имеющая следующий вид INPUT #9\*USR 15619:..., которая при ошибке загрузки вызывает остановку программы.

## ***ДИСКИ И ДИСКОВОДЫ***

Чаще всего с контроллером ВЕТА используется 5.25 дисковод двойной плотности. Дисководы высокой плотности (1.2, 1.44MB) использовать не имеет смысла, т.к. данный режим не поддерживается. Если у Вас применяется 40-дорожечный дисковод, у Вас будут проблемы с перезаписью, при обмене программами, т.к. 80-дорожечные диски на таком дисковом не читаются, а 40-дорожечные на 80-дорожечном - читаются, но не пишутся. Напомню, что режим "40" на 80-дорожечном дисковом не обеспечивает стандартную 40-дорожечную запись (см. фирменную инструкцию)!

Дисковод с двойной плотностью называют 80-дорожечным, хотя практически он обеспечивает доступ к 84...86 дорожкам (в зависимости от конкретного дисковода и дискеты). В нумерации дорожек имеется некоторая путаница, так как иногда используются номера цилиндров (пар дорожек, 0...85) и номер стороны (0,1; 0-верхняя), а иногда - сквозная нумерация (0...171; четные номера - верхние). При использовании TR-DOS встречаются оба варианта. Доступ к дорожкам 80...85 возможен при вызове процедур из ПЗУ; стандартными командами DOS он не поддерживается.

TR-DOS дает 16 секторов на дорожке и 256 байтов в секторе. Уменьшенный (в MS-DOS, например, сектор=512) размер сектора имеет преимущество при хранении большого числа коротких файлов, т.к. более экономно расходуется дисковое пространство (если файл имеет, к примеру, длину всего 1 байт, на диске он все равно займет целый сектор). Кроме того, при использовании файлов произвольного доступа увеличивается скорость операций.

Сказанное дает возможность подсчитать количество секторов и вместимость форматированного диска (учитывая, что TR-DOS использует внешнюю [0] дорожку для хранения системной информации о диске):

40 дорожек односторонний =  $38 \cdot 16 = 624$  сектора  $\cdot 256 = 156$  KB  
 40 дорожек двухсторонний =  $79 \cdot 16 = 1264$  сектора  $\cdot 256 = 316$  KB  
 80 дорожек односторонний =  $79 \cdot 16 = 1264$  сектора  $\cdot 256 = 316$  KB  
 80 дорожек двухсторонний =  $159 \cdot 16 = 2544$  сектора  $\cdot 256 = 636$  KB  
 86 дорожек (максим. кол.) =  $171 \cdot 16 = 2736$  сектора  $\cdot 256 = 684$  KB

Это означает, что на 1 дорожке помещается 4 KB, или 4 сектора на 1 KB. Таким образом, разделив количество свободных секторов на 4, можно узнать, сколько килобайт свободного пространства осталось на диске (напомню, 1KB=1024 байт).

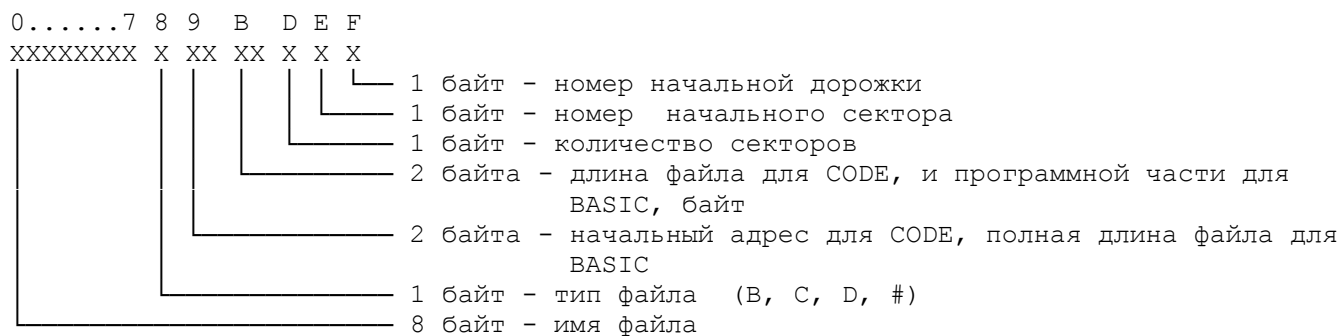
TR-DOS использует двойную плотность записи, поэтому получаем 80 дорожек и 16 секторов по 256 байт на дорожке.

## СТРУКТУРА ДИСКЕТЫ TR-DOS

Вся служебная информация о дискете размещается на нулевой дорожке. Первые восемь секторов (№ 0...7) содержат каталог файлов, следующий сектор - № 8 - системный, в нем записаны ее параметры. С дорожки номер 1 начинается основное пространство дискеты, где собственно и хранятся файлы. Как видите, организация диска очень проста, по сравнению с MS-DOS, поэтому и возникают сложности с удалением файлов, требуется уплотнение дискового пространства командой MOVE.

Сектора нулевой дорожки с номерами от 9 по 15 обычно не используются и при помощи рассмотренных ниже процедур DOS в них можно разместить какую-либо собственную информацию. Существует, например, оригинальный "boot", занимающий в основной области диска всего один сектор, который при записи размещает свои коды в свободных секторах нулевой дорожки (автор Ю. Власов). Однако, пользуясь этими секторами, не записывайте на такой диск при помощи "Magic" - данная процедура испортит сектора с номерами 10, 11 о "Magic" см. далее).

## СТРУКТУРА ЗАГОЛОВКА ФАЙЛА В КАТАЛОГЕ



Процедура удаления файла заключается в замене первого байта его имени кодом 01, после чего, файл считается удаленным и не отображается в каталоге. Вся информация на дискете сохраняется, пока не будет выполнена команда MOVE. Если удален файл в конце каталога, первый байт имени может получить нулевое значение (признак свободного пространства).

Для файлов типа "DATA", поле длины (Length) также представляет собой длину массива (переменной), а поле адреса (Start)-начальный адрес этого массива в момент записи, с учетом 257-байтного сдвига на величину буфера обмена (см. далее). Для дисковых файлов (#), поле длины содержит длину заполненной части блока, от 0 до 4096 (напомню, что эти файлы записываются блоками по 4096 байт, т.е. 16 секторов, под одинаковым именем). Старший байт поля адреса для дискового файла равен 32, а младший - номеру блока в файле (0, 1, 2,...), что в каталоге отображается как 8192, 8193, 8194...

На дискете заголовки файлов записаны в первых 8 секторах (с номерами от 0 до 7) нулевой дорожки. Кроме того, следует заметить, что порядковый номер заголовка файла в каталоге имеет следующее свойство: старшая половина байта соответствует номеру сектора, в котором записан заголовок, а младшая - номеру заголовка в секторе. Это бывает полезно при работе с диском.

Как видим, номер строки автостарта BASIC- программы не входит в данные заголовка. Он записывается вместе с программой, 5 байт после конца программы или области переменных, если они есть – в поле ELINE: после разделителя полей #80 следует байт #AA и 2 байта номера строки (младший, старший). В отличие от кассеты, автостарт с нулевой строки не выполняется.

Поле "число секторов" используется при копировании файла, а поле "длина файла" - при его загрузке. Обнулив число секторов, можно сделать файл не копируемым (хотя эта защита так же легко и снимается).

Каталог дискеты вмещает максимум 128 файлов, считая и стертые; а так как подкаталоги в TR-DOS не создаются, то при большом числе коротких файлов каталог переполняется, и часть пространства на дискете остается неиспользованной.

### **СТРУКТУРА СИСТЕМНОГО СЕКТОРА**

Системную информацию о дискете содержит 8-й сектор нулевой дорожки, точнее, конец этого сектора, начиная с 225-го байта:

ИМЯ	АДРЕС			НАЗНАЧЕНИЕ
BUFF_ADR	0	#00	#5025	Начало системного сектора, загруженного в динамически выделяемый буфер для дисковых операций.
DCU_SEC	223 224	#DF		Слово. Содержит число секторов на диске, расформатированном при помощи DCU (вначале равно значению в #E5). На диске расформатированном TR-DOS равно 0.
FR_S_NEXT	225	#E1	#5E06	Байт. Номер следующего свободного сектора. Инициализируется нулем.
FR_T_NEXT	226	#E2	#5E07	Байт. Номер следующей свободной дорожки. Инициализируется единицей.
TYPE DISC	227	#E3	#5E08	Байт. Тип дискеты: #16 - 80 track, 2 side #17 - 40 track, 2 side #18 - 80 track, 1 side #19 - 40 track, 1 side
N_FILES	228	#E4	#5E09	Байт. Количество файлов на диске, включая удаленные. Инициализируется нулем.
N_FRE_SEC	229 230	#E5	#5E0A	Слово. Количество свободных секторов. Инициализируется в зависимости от типа диска: #09F0 - 80 track, 2 side #04F0 - 40 track, 2 side #04F0 - 80 track, 1 side #0270 - 40 track, 1 side При форматировании программой DCU на максимальное число дорожек: #0A70 - 84 track, 2 side #0AB0 - 86 track, 2 side
MAIN_BYTE	231	#E7	#5E0C	Байт. Количество секторов на дорожке. Должно быть #10, иначе диск признается неформатированным.
ZERO	232	#E8		Два байта нулей.
BLANK9	234	#EA		9 байтов пробелов (код #20).
ZERO	243	#F3		1 нулевой байт.
N_DEL_FIL	244	#F4	#5E19	Байт. Количество удаленных файлов. Инициализируется нулем.
DISC_TITL	245	#F5	#5E1A	Поле из 8 байт. Имя диска после форматирования.
ZERO	253	#FD		3 байта нулей. Конец главного сектора.

\* ПРИМЕЧАНИЕ: адрес дан от начала сектора в десятичном и HEX виде, а также во временном буфере DOS, начиная с 23845 (#5D25), куда считывается системный сектор, что

необходимо при анализе ПЗУ. Для шестнадцатеричного представления вместо стандартной записи, скажем, 5D25h, используется более привычная для пользователей SPECTRUMа - #5D25.

### **ЗАМЕЧАНИЯ О ДРУГИХ ВЕРСИЯХ TR-DOS**

Версия 5.03 TR-DOS является более быстрой. Особенно это заметно при выполнении команд LIST, MOVE, COPY B. Однако для повышения быстродействия пришлось отказаться от тестирования дисководов, в связи, с чем некоторые старые типы дисководов с низкой скоростью перемещения головки с этой версией работать не будут. На уровне команд отличий между версиями нет, однако структура ПЗУ версий 5.01 и 5.03 сильно отличается; 5.04 по адресам более похожа на 5.03. Далее, в основном речь идет о версии 5.01, которая используется автором; некоторые сведения о 5.03 приводятся из литературы (например, сборник "ZX-Ревю" № 1 за 93 г). Хотелось бы подчеркнуть, что автор не является экспертом по всем версиям TR-DOS. В основном приведенная информация достоверна и для других версий, хотя не исключены и какие-то отличия.

### **ВХОДНЫЕ ТОЧКИ TR-DOS**

Способ работы со стандартными командами TR-DOS подробно приведен в её фирменной инструкции. При программировании на BASICе пользоваться ими легко и удобно. Однако при более сложной работе со SPECTRUMом возникает необходимость обращаться к операционной системе, минуя BASIC и интерпретатор TR-DOS, например, из программы, написанной в машинных кодах. Предлагаемый в фирменном руководстве к TR-DOS способ (имитация строки BASICа в кодах) громоздок и неудобен. Гораздо лучше обращаться непосредственно к операционной системе, находящейся в "теновом" ПЗУ. Для этого в TR-DOS существуют специальные возможности, и на некоторых из них я остановлюсь.

Включение DOS происходит при переходе на один из адресов, находящийся в диапазоне 15616-15871 (3D00-3DFF), т.е. там, где в основном ПЗУ находится таблица символов и, следовательно, при нормальной работе управление туда никогда не передается. Схема контроллера BETA сделана так, что при выборке команды в этом диапазоне адресов срабатывает триггер, который заменяет основное ПЗУ теновым - DOS, одновременно становятся доступными порты контроллера (см. приложение). Обратное переключение происходит при переходе на любой адрес в ОЗУ, так что доступ к DOS весьма ограничен - Вы не можете непосредственно из своей программы обращаться ни к его портам, ни к ПЗУ, за исключением указанного диапазона адресов. Однако в этом диапазоне находятся все входные точки для вызова большинства необходимых функций DOS.

Фактически имеется 23 точки входа, однако применяются только некоторые из них, т.к. не все они известны. Основные входные точки начинаются с команды NOP (00), по-видимому, для компенсации задержки при переключении ПЗУ (возможен и запуск по следующему адресу):

DEC	HEX	
15616	3D00	— вход в DOS из BASIC.
15619	3D03	— вызов команды DOS из BASIC.
15622	3D06	— канал связи с дисковым файлом - программа ввода.
15629	3D0D	— канал связи с дисковым файлом - программа вывода (в описании канала используется 3D0E).
15635	3D13	— выполнение команды, заданной в регистре "C" процессора (см. далее).
15652	3D24	— связана с инициализацией системной области DOS - если при вызове включен флаг "C", выделяется 112 байт для системной области DOS, и регистр HL устанавливается на начало этой области (5CC2). При обращении по адресу #3021 предварительно проверяется наличие системной области.
15663	3D2F	— переход на любой адрес DOS.

Используя последнюю из указанных точек, можно получить полный доступ к TR-DOS ПЗУ. По адресу 15663 (#3D2F) вслед за командой NOP следует единственная команда RET. Она позволяет осуществлять переход по любому адресу DOS, предварительно помещенному на вершину стека, например, следующим образом:



```

LD      HL,ADR_DOS      ; адрес в DOS ПЗУ
CALL    DOS              ; чтобы получить на стеке адрес возврата
...      ...              ; продолжение программы
DOS      PUSH    HL      ; на стеке-адрес подпрограммы DOS
        JP      #3D2F    ; вызов подпрограммы через точку входа

```

Этот способ хорош в том случае, когда есть дизассемблированный текст программы TR-DOS с отмеченными адресами соответствующих подпрограмм. Любопытным читателям можно рекомендовать самостоятельно получить такой текст и разобраться в нем (необходимые для этого исходные коды можно записать: SAVE "DOS" CODE 0,16383). Правда, следует предупредить, что интерпретатор DOS написан крайне сложно и невразумительно. Процесс анализа дизассемблированного текста весьма напоминает решение кроссворда, только пользы от этого занятия несколько больше. Для облегчения задачи расшифровки кодов отметим участки, которые занимают некоторые процедуры и области данных:

v5.01	v5.03	ПРОЦЕДУРЫ И ДАННЫЕ
0000-0007	0000-0007	холодный старт; полная инициализация системы
0008-000F		RST8, инициализация, в DE задана вершина памяти
0010-0012		RST10, вывод символа из регистра A, как в SOS
0018-001A		RST18, вывод строки символов, начиная с (HL); последний символ должен быть #00 или >#7F
0020-0022		RST20, вызов программы из основного ПЗУ; адрес задается двумя байтами, следующими после RST20
0028-002A		RST28, возвращает в HL адрес текущего канала с учетом сдвига, заданного в регистре C
0066-0068	0066-0069	переход на адрес обработки NMI от клавиши "Magic"
0069-		копирует стандартные литеры A...U в область UDG устанавливает стандартные значения системных переменных
016C-01D2		вызов DOS из BASICA (через #3D03, т.е. 15619)
01C7-		LD A,(HL) и устанавливает флаг Z, если A=#0D или #80
01D3-	01D3-	выход из DOS после выполнения команды; литеры #0D,#0F,#10,#11 подпрограммы #3D13 (резерв)
0239-		вход в DOS из BASICA (через #3D00, т.е. 15616)
0355-03A0	0360-03AB	название версии TR-DOS
03FB-	0405-	выполнение литеры #18 подпрограммы #3D13
0429-		команда CAT
0530-		команда NEW
0686-		команда COPY
077D-		команда ERASE
07F6-0FFF	0800-0FFF	свободные ячейки памяти (около 2 KB #FF)
1000-		команда "*"
1003-1017		текст "Подсоединен интерфейс 1"
108D-1138	10A5-1150	текст системной информации для LIST
115D-11A7		печать LIST (регистр HL) в десятичной форме
11B6-		команда LIST
1348-		команда COPY s
1514-		команда COPY B
1644-	165C-	выполнение литеры #08 подпрограммы #3D13
164C-	1664-	выполнение литеры #09 подпрограммы #3D13
1693-		команда MOVE
17DD-17DE		Пересылка - LDIR, RET (нужно задать HL,BC,DE)
17E0-		команда VERIFY
17E5-		команда LOAD
1971-		команда PEEK
1975-		команда POKE
197D-		команда MERGE

1A9C-		команда SAVE
1CBC-	1CF0-	выполнение литеры #0A подпрограммы #3D13
1CC7-		команда RETURN
1D19-		команда RUN
1E01-	1E3D-	выполнение литеры #05 подпрограммы #3D13
1E11-	1E4D-	выполнение литеры #06 подпрограммы #3D13
1E86-		команда FORMAT
1F7D-1F8D	1FB9-1FC9	таблица
1FAF-	1FEB-	выполнение литеры #16 подпрограммы #3D13
1FBA-	1FF6-	выполнение литеры #17 подпрограммы #3D13
1FC1-		форматирование дорожки
2051-		проверка дорожки
208A-		форматирование дорожки, N-1 в регистре E
213C-		команда OPEN#
260C-		команда CLOSE
26EF-	2739-	выполнение литеры #15 подпрограммы #3D13
271C-27F1	2766-283B	тексты сообщений TR-DOS, начиная с "O.K."
27F2-	283C-	переход с адреса #3D13, определение литеры команды
2842-288D	288C-28D7	таблица адресов подпрограмм, адресуемых регистром C при обращении через #3D13
288E-	28D8-	выполнение литеры #07 подпрограммы #3D13
2896-	28E0-	выполнение литеры #13 подпрограммы #3D13
2899-	28E3-	выполнение литеры #14 подпрограммы #3D13
28E5-		при A=0 - перенос 16 байт заголовка из адреса HL в #5CDD, при A<>0 - из #5CDD по адресу HL
28A8-	28F2-	выполнение литеры #0C подпрограммы #3D13
28B1-	28FB-	выполнение литеры #0B подпрограммы #3D13
28C5-	290F-	выполнение литеры #0E подпрограммы #3D13
28DC-	2926-	выполнение литеры #12 подпрограммы #3D13
294D-		команда "40"
2964-		команда "80"
2968-29EA	29B1-2A34	тексты сообщений TR-DOS, начиная с "*ERROR*"
2A09-2A0B		вывод в порт - OUT (C),A и RET
2A0C-	2A56-2A06	обслуживание запроса прерывания NMI от клавиши "Magic"
2D57-		команда GOTO
2F79-2F7B		запись команды из регистра A в BГ93
2FA9-2FE7		таблица
30B3-317B	30FD-31F2	тексты ключевых слов TR-DOS, начиная с "SAVE"
317E-	2FF3-3031	таблица
31B3-3C00		свободные ячейки памяти (2637 байт #FF)
3D00-3D2F	3D00-3D2F	область стандартных точек входа в TR-DOS
3D8E-	3D98-	выполнение литеры #00 подпрограммы #3D13
3DC1-	3DCB-	выполнение литеры #01 подпрограммы #3D13
3E3A-	3E44-	выполнение команды BГ93 из регистра B,
		данные в регистре A
3E59-	3E63-	выполнение литеры #02 подпрограммы #3D13
3EF8-	3F02-	выполнение литеры #03 подпрограммы #3D13
3EFC-	3F06-	выполнение литеры #04 подпрограммы #3D13
3F09-		чтение дорожки (со всей служебной информацией)
3FB0-		процедура блочной записи - OUTI и RET
3FE6-3FFF	31FD-3C00	свободные ячейки

Обратите внимание, что ПЗУ версии 5.01 содержит более 4кб свободного пространства, открывая широкие перспективы по его доработке, пока слабо освоенные.

Нетрудно заметить, что создатели TR-DOS не придерживаются концепции сэра Синклера о неизменности ПЗУ, поэтому в различных версиях TR-DOS адреса процедур различны. Желательно обходиться стандартными точками входа (как это делается в большинстве порядочных систем, например, на IBM PC). Если Вы все же используете прямое обращение к ПЗУ, не поленитесь вставить в программу проверку версии DOS, например, при помощи стандартной процедуры #3D13 (см. далее):

```
LD    HL,#0361      ; проверяемый адрес в DOS ПЗУ для v5.01
LD    C,#13         ; литера процедуры #3D13 (см. далее)
CALL  #3D13         ; перенос 16 байт с адреса в HL в #5CDD
CALL  comp          ; процедура сравнения 8 байт в #5CDD и в
                   ; строке EXAMPL, если совпадают, флаг Z=0
JP     NZ,EXIT      ; не совпадают - версия не 5.01 - выход
                   ; или проверка на другую версию
...    ...          ; продолжение - можно работать с DOS ПЗУ
EXAMPL DEFM 'Ver 5.01' ; последовательность байт, характерная
                   ; для данной версии (любая известная)
```

Если у Вас нет желания написать подпрограмму сравнения "comp", версии 5.01 и 5.03 можно отличить более простым способом:

```
LD    HL,#026A      ; проверяемый адрес в DOS ПЗУ
CALL  #01C7         ; возвращает в регистре А байт по (HL)
CP    #AA           ; в регистре А байт #AA - версия 5.01
JP    Z,V501
CP    #F4           ; в регистре А байт #F4 - версия 5.03
JP    Z,V503
JP    EXIT          ; не #AA и не #F4 - неизвестная версия, выход
```

Однако, весьма вероятно, что с другими версиями DOS данная программа застрянет в #01C7, так и не достигнув "EXIT", поэтому первая проверка более универсальна.

К сожалению, в последнее время стали появляться программы, создатели которых разобрались со своей версией DOS и считают, что она единственная в мире. Запуск таких программ на другом компьютере, как правило, приводит к порче диска.

Наиболее важной для вызова TR-DOS из программы в машинных кодах является точка входа с адресом 15635 (#3D13). Разработчики возложили на данную процедуру функции, которые в серьезных машинах (например, IBM PC) выполняют векторы прерываний системы ввода-вывода. С адреса #3D13 в конечном итоге происходит переход к подпрограмме, которая, в зависимости от кода, содержащегося в регистре С, при помощи таблицы #2842 [10306 dec] (#288C [10380] в v5.03), передает управление соответствующей подпрограмме. Параметры для этих подпрограмм передаются при помощи регистров микропроцессора либо находятся в области системных переменных TR-DOS (например, заголовок файла). Ниже приводятся краткие описания процедур, вызываемых в зависимости от литеры (содержимого регистра С):

- С=#00 — выполняется команда "Восстановление" ВГ93: головка отводится на нулевую дорожку и ожидает сигнала INTRQ. Ожидание можно прервать нажатием клавиши BREAK.
- С=#01 — инициализация дисководов, заданного в регистре А. Если для указанного номера не инициализирована соответствующая переменная 23802...23805 (= #FF), то она устанавливается равной #08, затем определяются число дорожек и заносится в 23752...23755 (#00-для 40 дорожек, #80-для 80). Затем заданный номер из регистра А заносится в 23798 (временный дисковод), а в 23830 заносится копия системного регистра.
- С=#02 — установка головки на дорожку, номер которой задан в регистре А (0...159, т.е. 1-му треку, и сторонам 0 и 1, соответствуют номера дорожек 0 и 1).
- С=#03 — установка номера сектора, номер которого задан в регистре А (1...16) - номер помещается в #5CFF (23807).
- С=#04 — установка адреса буфера, заданного в регистре HL, в системную переменную #5D00 (23808).

- C=#05 – чтение блока секторов: в буфер по адресу HL считывается В секторов, первый из которых имеет номер Е на дорожке D (если В=0, считываются только заголовки секторов).
- C=#06 – запись блока секторов (параметры аналогичны предыдущей команде): из буфера по адресу HL на диск (дорожка D, сектор Е) записывается В секторов.
- C=#07 – вывод каталога диска в канал, заданный в регистре А (2-экран, 3-принтер). Номер диска берется из переменной #5CF6 (23798). Необходимо продублировать этот номер в ячейках 23800, 23801. Предварительно выполняется процедура #18.
- C=#08 – считывается заголовок файла в область системных переменных по адресу #5CDD-#5CEC. Номер требуемого файла (от 0 до 127, включая и удаленные) задается в регистре А.
- C=#09 – запись заголовка файла из системной области на диск, на место файла с номером, указанным в регистре А (переименование).
- C=#0A – поиск файла в каталоге по имени и типу, которые заданы в области системных переменных #5CDD-#5CE5; количество проверяемых байтов указывается в переменной 23814 (обычно равное 9). Если файл найден, его номер возвращается в регистре С и ячейках 23838 (#5D1E), 23823 (#5D0F); если нет - в регистре С содержится #FF, 23838 не изменяется, в 23823 - код ошибки #FF.
- C=#0B – запись на диск файла типа CODE с адреса HL длиной DE; имя (и тип?) файла должно находиться в системной области #5CDD-#5CE4. Проверка на наличие такого имени не производится. Предварительно выполняется процедура #18.
- C=#0C – запись на диск BASIC-программы; имя файла задается в системной области. Проверка имени также не производится. В переменной #5CD1 задается номер строки автостарта (младший и старший байт). Иногда, встречается замечание, что если тип не равен "B", файл записывается как "boot"; возможно, для версии 5.03? Предварительно выполняется процедура #18.
- C=#0D – выход из TR-DOS (резерв для расширения впоследствии).
- C=#0E – загрузка файлов или VERIFY; при этом заголовок и тип должен быть помещен в область #5CDD-#5CE5. При системной переменной 23801 (#5CF9) равной #00 выполняется загрузка, при #FF - проверка. Для загрузки BASIC - программы переменная 23824 (#5D10) должна быть равна #00; если же она равна #FF, то при загрузке новой, старая программа удаляется, но новая не считывается. В зависимости от содержимого регистра А подпрограмма будет работать по-разному:
- А=#00 – адрес загрузки и длина берется из каталога;
- А=#03 – коды загружаются с адреса HL, длина загружаемого файла определяется значением DE;
- А=#FF – коды загружаются с адреса HL, но длина загружаемого файла равна длине записанного на диск.
- C=#0F...11 – выход из TR-DOS (общий адрес для процедур #0D, #0F, #10, #11), (в версиях 5.01 и 5.03 резерв для расширения).
- C=#12 – стирается файл, имя и тип которого заданы в #5CDD-#5CE5, причем удаляются все файлы с такими параметрами. В #5D08 сохраняется первый байт имени, которым был заменен на #01; в #5D07 - число удаленных файлов.
- C=#13 – перенос информации о файле (16 байт) с адреса в HL, в #5CDD-#5CEC.
- C=#14 – перенос информации о файле с #5CDD-#5CEC по адресу в HL.
- C=#15 – проверка дорожки, номер которой задан в регистре D. Если обнаружены дефектные сектора, в #5D0F будет помещен код ошибки равный #07, а в системной переменной #5CD6 – число дефектных секторов.
- C=#16 – устанавливает текущей сторону диска 0 (верхнюю), например, для предыдущей процедуры. Изменяет системный регистр #FF (значением из #5D16 маскированным OR #3C, т.е. 00111100). Иначе говоря, загрузка системного регистра значением равным А OR #3C.

C=#17 – устанавливает текущей сторону диска 1 (нижнюю), (значение из #5D16 маскируется AND #6F). Иначе говоря, загрузка системного регистра значением равным A AND #6F.

C=#18 – считывает системный сектор и производит настройку контроллера на тип дискеты.

Таким образом, через процедуру #3D13 можно выполнить практически все необходимые операции с диском, которые могут понадобиться обычной прикладной программе. Только при написании специальных дисковых программ (например, для форматирования) требуется прямое обращение к DOS ПЗУ, однако без крайней необходимости это делать нецелесообразно, так как возникнет проблема совместимости.

В качестве примера приведем часто встречающийся загрузчик для программы-моноблока (все части программы записаны последовательно и объединены с BASICом):

```
LD HL,40000 ; адрес загрузки 1-го блока
LD B,17 ; число загружаемых секторов для него
CALL DOS ; вызов процедуры загрузки
CALL 40000 ; например, декомпрессия экрана
LD HL,24500 ; адрес загрузки 2-го (основного) блока
LD B,115 ; число загружаемых секторов
CALL DOS ;
CALL 24500 ; например, декодирование программы
JP START ; запуск загруженной программы
DOS LD DE, (#5CF4) ; адрес на диске из системной переменной
LD C,5 ; загрузка блока секторов
JP #3D13 ; вызов функции DOS
```

Загрузчик использует литеру #05 процедуры #3D13, т.е. грузит число секторов, заданное в регистре B, по адресу, заданному в HL (для разных программ, естественно, они будут различны). Чтобы определить адрес на диске, используется системная переменная #5CF4, где после загрузки каждого блока (в т.ч. начальной BASIC-части) находится адрес следующего сектора, дорожки. Таким образом, можно загружать любое число последовательных блоков, причем загрузка происходит значительно быстрее, потому что головка дисководов не ездит на нулевую дорожку за параметрами каждого файла; соответственно, уменьшается износ дискеты. Как правило, блоки записывают сразу после BASIC-загрузчика, и затем при помощи любого "Дискового доктора" исправляют число секторов этого загрузчика на суммарное, а бывшие блоки помечают как удаленные. Кроме прочих преимуществ, программу-моноблок труднее взломать.

Учтите, что если длина блока не кратна длине сектора (256 байт), с последним сектором, после Вашего блока кодов будет загружаться некоторый "мусор". Поэтому, когда требуется загрузить заданное число байтов, нужно пользоваться литерой #0E процедуры #3D13:

```
LD HL,NAME ; имя 1-го блока
LD C,#13 ; перенос имени из (HL) в область переменных DOS
CALL #3D13 ; вызов функции DOS
XOR A ; адрес и длина блока - из каталога
LD (#5CF9),A ; для LOAD
LD C,#0E ; загрузка файла по его имени
CALL #3D13 ;
LD HL,#5CE4 ; последний байт имени файла
INC (HL) ; следующий номер блока
LD HL,ADR2 ; рабочий адрес 2-го блока
XOR A ;
LD (#5CF9),A ;
LD A,#FF ; адрес блока - в HL, длина -
LD C,#0E ; из каталога
CALL #3D13 ;
JP START ; запуск загруженной программы
NAME DEFM "block_01C" ; имя и тип загружаемого файла
```

В приведенных примерах предполагается, что загружаемые файлы на диске, безусловно, присутствуют. Такой метод можно применять только для загрузки программ. Если же требуется загрузка из прикладной программы, когда имя файла задается пользователем, должна проводиться проверка наличия запрашиваемого файла. Желательно также выполнить настройку на тип диска (так как дискету могли заменить после запуска программы). Это можно выполнить, например, так:

```
LD C,#18 ; настройки на тип дискеты
CALL #3D13 ; вызов функции DOS
```

```

LD     HL,NAME           ; имя запрашиваемого файла
LD     C,#13             ; перенос имени из (HL) в области DOS
CALL  #3D13
LD     C,#0A             ; поиска файла по его имени
CALL  #3D13
LD     A,C               ; в "C"-номер файла в каталоге
INC    C
JR     Z,EXIT            ; если C=#FF, файл не найден - выход
LD     C,#08             ; чтение параметров файла в переменной DOS
CALL  #3D13
XOR    A
LD     (#5CF9),A         ; для LOAD
LD     HL,ADR            ; адрес загрузки блока
LD     A,#FF             ; адрес блока в HL, длина - из каталога
LD     C,#0E             ; загрузка файла по его имени
CALL  #3D13
RET
NAME   DEFM "textnameC"  ; имя и тип загружаемого файла

```

КАРТА ПАМЯТИ TR-DOS

При инициализации TR-DOS для нее выделяется 112 байт, соответственно область памяти, начиная с CHANS, сдвигается вверх:

Стандартное начало BASIC-области:	23755	#5CCB
Начало BASIC-области с TR-DOS:	23867	#5D3B
Начало BASIC-области с TR-DOS и интерфейсом 1:	23925	#5D75

Кроме того, во время чтения или записи сектора, между областью переменных DOS и CHANS создается буфер сектора - 257 байт (256 байт данных + контрольный байт). При этом область памяти, начиная с #5D25 (CHANS-1), сдвигается вверх еще на 257 байт, таким образом, что BASIC-область начинается с #5E3C. Если RAMTOP установлен слишком низко, то при вызове команды DOS может быть выдано сообщение "Out of memory", хотя в остальном BASIC работает нормально; это говорит о недостатке памяти для выделения 257-байтного буфера. Выделение буфера необходимо учитывать также при чтении кодов в область BASIC-программы: например, для загрузки в нулевую строку после REM, необходимо указывать адрес не PROG+5, а PROG+5+257, т.е. #5E41.

При переносе кассетных программ на диск, коды в нулевой строке оказываются сдвинуты на 112 байт, что требует соответствующей коррекции адресов. Проще убрать область DOS при помощи процедуры из основного ПЗУ:

```

LD     BC,112           ; величина сдвига
LD     HL,23755         ; адрес = стандартному началу BASIC-области
CALL  #19E8            ; удаление блока, начиная от HL, длиной BC

```

Указанная процедура сдвигает всю BASIC-область, корректируя значения системных переменных.

Следует отметить, что 0-я строка с кодами может неправильно обрабатываться DOS, не рекомендуется давать команды CAT, LIST.

Указанный способ выделения буфера перед областью CHANS объясняет странное поведение TR-DOS при обращении к MS-DOS дискете: любое обращение к диску начинается с чтения 8-го сектора во временный буфер, а так как сектор MS-DOS вдвое длиннее, таблица CHANS оказывается затертой - с соответствующими последствиями...

Адрес таблицы канальной информации указывается системной переменной CHANS (по адресу 23631, #5C4F); начало BASIC-программы определяется переменной PROG (см. 23635, #5C53).

## НЕКОТОРЫЕ СИСТЕМНЫЕ ПЕРЕМЕННЫЕ TR-DOS v5.01

В скобках [ ] даны значения при инициализации, соответственно, если DOS стартовала без диска в дисковом / с диском / и в процессе работы.

DEC	HEX	ЗНАЧЕНИЯ
23734	#5CB6	15 байт - стандартная область каналов CHANS K,S,R: по 2 байта - адреса процедуры вывода из системы, ввода в систему и байт-имя канала (K, S, R). В отличие от стандартных, изменен адрес в канале R: (23746) =#C9, так что адрес обработки ошибок канала R=#15C9, вместо #15C4 (аналогично в области DOS CHANS - 23858). Замена значения (#5CC2) с #C9 (RET) на #C3 (JP ?), [2 байта адреса] ведет к автостарту (по адресу) файла CODE после загрузки. При подключённом INTRFACE1 байт 23734=#00
23734	#5CB6	[2548,4264 (#09F4,#10A8)] K
23739	#5CBB	[2548,5572 (#09F4,#15C4)] S
23744	#5CC0	[3969,5577 (#0F81,#15C9)] R
23746	#5CC2	байт=#C9.Используется для вызова подпрограмм из DOS
23752	#5CC8	байт - режим работы диска "A" [#FF/#??]
23753	#5CC9	байт - режим работы диска "B" [#FF/#??]
23754	#5CCA	байт - режим работы диска "C" [#FF/#??]
23755	#5CCB	байт - режим работы диска "D" [#FF/#??] Ячейки инициализируются только в DOS 5.01 [#FF, если дисковод не инициализирован; иначе %?00000??: бит 7 = 1 для DD, = 0 для SD; бит 1 = 1 для DS, = 0 для SS; бит 0 = 0, если DD используется в режиме SD]
23756	#5CCC	байт - Номер сектора, выводимого командой CAT [#0D]
23757	#5CCD	байт - готовности дисковода (регистр состояния BG93) [#80/#00]
23758	#5CCE	байт - #00 операция чтения, #FF оператор записи [#00]
23759	#5CCF	2 байта - адрес WORKSP при работе MOVE, COPY, LIST
23761	#5CD1	2 байта - номер строки автостарта при SAVE BASIC
23764	#5CD4	байт - номер стертого файла при работе MOVE
23765	#5CD5	байт - номер сектора стертого файла при MOVE
23766	#5CD6	байт - номер дорожки стертого файла при MOVE. Для форматирования обнулить, по окончании будет число дефектных секторов (аналогично при проверке дорожки процедурой #15 команды #3D13). Способ загрузки для литеры #0E (регистр A). Равен #FF, если команда не выполнена
23767	#5CD7	2 байта - адрес. После инициализации указывает на счетчик [23771], после записи CODE - на начальный адрес в памяти. При вызове из BASICа содержит адрес строки, из которой произошел вызов. Кроме того, после проверки типа дисковода #5CD7 содержит число дорожек (?). При выполнении форматирования #5CD7 содержит число дорожек для форматирования, а #5CD8- флаг VERIFY: при #5CD8, не равным 0, форматированная дорожка не проверяется
23769	#5CD9	2 байта - адрес символа для интерпретации, аналогично CH_ADD. После инициализации указывает на [23869]. После записи файла CODE содержит длину записанного файла. При форматировании переменная #5CDA=#80, если дискета двусторонняя (иначе форматирована только SIDE1). Для литеры #0E хранит адрес загрузки файла
23771	#5CDB	2 байта - счетчик загрузки; после операций с файлами содержит длину загруженной/записанной части файла (?). При MOVE содержит текущие номера сектора и дорожки. При LIST байт #5CDB содержит номер текущего канала вывода
23773	#5CDD	16 байт заголовок файла; в той же форме, как и в каталоге

23773	#5CDD	8 байт - имя файла
23781	#5CE5	1 байт - тип файла (B, C, D, #)
23782	#5CE6	2 байта - адрес START для CODE или полная длина для BASIC
23784	#5CE8	2 байта - длина файла или программной части для BASIC
23786	#5CEA	1 байт - объем файла в секторах
23787	#5CEB	1 байт - номер начального сектора файла (#00...#0F)
23788	#5CEC	1 байт - номер начального трека файла (#00...#9F)
23789	#5CED	7 байт - параметров файла при копировании с дисковод на дисковод
23789	#5CED	2 байта - адрес START для CODE или полная длина для BASIC
23791	#5CEF	2 байта - длина файла или программной части для BASIC; при наличии INTERFACE 1 байт=#01
23793	#5CF1	1 байт - объем файла в секторах
23794	#5CF2	1 байт - номер начального сектора файла (#00...#0F)
23795	#5CF3	1 байт - номер начального трека файла (#00...#9F)
23796	#5CF4	байт - счетчик загружаемых секторов (номер сектора, обрабатываемый контроллером). После операций с файлами содержит номер следующего сектора, что часто используется при создании загрузчиков
23797	#5CF5	байт - счетчик загружаемых дорожек; после операций с файлами содержит номер следующей дорожки
23798	#5CF6	байт - дисковод для временной операции (0-3) [#00] (т.е. когда задается имя, например, "B: name"). Для адресации DOS использует двухбайтное число, поэтому байт #5CF7=#00
23799	#5CF7	байт - при возврате из DOS (15616) обнуляется
23800	#5CF8	байт - дисковод-источник при операции с двумя файлами (#00-#03) [#00]; #5CF9=#FF, если канал открыт (?)
23801	#5CF9	байт - дисковод-приемник при операции с двумя файлами (#00-#03) [#00]; тип операции: LOAD=#00, VERIFY=#FF [#00] (при #FF вместо команд загрузки будет выполняться сравнение)
23802	#5CFA	байт - время перемещения головки дисковода "A"
23803	#5CFB	байт - время перемещения головки дисковода "B"
23804	#5CFC	байт - время перемещения головки дисковода "C"
23805	#5CFD	байт - время перемещения головки дисковода "D" [#FF, если дисковод не инициализирован; иначе #08-#0B; #08 - максимальная скорость]. Используются только в DOS 5.01, в других версиях скорость не изменяется (только быстрая), но значение #08 используется как признак инициализации дисковода
23806	#5CFE	байт - команда контроллера ВГ93, задаваемая DOS (см.#3F18). Для команд чтения #5CFE=#80(%1000 0000), для команд записи #5CFE=#A0 (%1010 0000) [#00]
23807	#5CFF	байт - номер сектора+1 для ВГ93, задаваемый DOS (см.#3F0D)
23808	#5D00	2 байта - адрес буфера DOS, в который грузится сектор. Указывает на адрес DOS-CHANS; эта область каналов вместе с программой при загрузке временно сдвигается на 257 ячеек памяти. При открытом дисковом файле указывает на начало BASIC-области (#5E5F) [#00/#00/#5D25]
23810	#5D02	2 байта - временно сохраняется регистр HL. После инициализации указывает на (E_LINE+1 ?) [#5D3D] (сохраненный регистр HL?)
23812	#5D04	2 байта - временно сохраняется регистр DE [#317C]
23814	#5D06	байт - число байт, по которому происходит поиск имени файла в каталоге (8-имя + 1-тип) [#09]
23815	#5D07	байт - результат MOVE: #00 - no files, <>#00 - O.K.; счетчик удаленных файлов в процедуре #12 подпрограммы #3D13
23816	#5D08	байт - сохраняется 1-я буква имени файла при MOVE (и ERASE в процедуре



		#12 подпрограммы 3D13)
23820	#5D0C	байт - признак выделения буфера ввода / вывода: #00 - выделен, #FF - нет [#FF] (см. #5D00)
23821	#5D0D	байт - номер файла при копировании с двумя дисковыми
23822	#5D0E	байт - при #FE сообщения DOS на экран не выводить, при #FF - признак работы BASICA, при другом - TR-DOSa
23823	#5D0F	байт - код ошибки TR-DOS при вызове из BASICA (см. ранее), а также номер файла в каталоге при его поиске литерой #0A процедуры #3D13 [#00]
23824	#5D10	старший байт ошибки при вызове 15616 или процедуры (например, литеры чтения файла #0E процедуры #3D13); необходимо предварительно записать #00 [#00/#FF]
23825	#5D11	2 байта - адрес строки команды TR-DOS: при вызове из BASICA указывает на CH_ADD, при работе в 15616 - на буфер редактора E_LINE [#5D3C]; адрес строки команды TR-DOS: при вызове из DOS = E-LINE; из BASIC = CH-ADD
23827	#5D13	2 байта - адрес, указывает на старший адрес стека, содержащий адрес процедуры обработки ошибок ERR_SP [#FF54]; копия ERR-SR; если старший байт = #AA, выполняется команда RUN "boot", а в #5D19 код #FE
23829	#5D15	байт - режим DOS: при #FF - выполнение команды DOS из BASICA, при #00 - работа в DOS
23830	#5D16	байт - копия системного регистра (режим работы контроллера, обычно номер дискового + #3C/#2C см. далее описание порта #FF)
23831	#5D17	байт - признак инициализации DOS: если = #00, при входе в DOS выводится заставка и выполняется команда RUN "boot", при = #AA - нет (повторный вход) [#00/#AA]; при = #FF, не дает ошибку при чтении неверного адреса
23832	#5D18	используется если подключен INTERFACE 1, при этом #5D18=#FF [#00]
23833	#5D19	байт - номер дискового по умолчанию(#00...#03)[#00]
23834	#5D1A	2 байта - адрес подпрограммы возврата из TR-DOS [#02C0] (встречается также #0201 ?)
23836	#5D1C	2 байта - хранят текущее значение стека SP [#FF3E]
23838	#5D1E	байт - номер файла в каталоге, если он найден [#00]
23840	#5D20	3 начальных байта последней командной строки
23843	#5D23	байт - объем памяти для MOVE (в блоках, min 4KB)
23845	#5D25	байт - начало временного буфера DOS [#FF]
23846	#5D26	20 байт - область каналов CHANS-DOS K, S, R, P; аналогичны стандартным по адресу #5CB6 (+ канал принтера):
23846	#5D26	[2548,4264 (#09F4,#10A8)] K
23851	#5D2B	[2548,5572 (#09F4,#15C4)] S
23856	#5D30	[3969,5577 (#0F81,#15C9)] R *
23861	#5D35	[2548,5572 (#09F4,#15C4)] P
23866	#5D3A	байт-разделитель [#80]
23867	#5D3B	Начало BASIC-программы, если не открывались дисковые файлы (см. далее)

\* ПРИМЕЧАНИЕ: область CHANS при инициализации DOS из BASICA (после USR 0) несколько отличается от инициализации DOS по сбросу: адрес процедуры ввода канала R имеет такое же значение, как без DOS - #15C4, т.е. байт (#5D32)=#C4, хотя (#5CC2) все же = #C9.

При инициализации стандартной BASIC-области из DOS переменные NSPPC (#5C44) ... SUBPPC (#5C47), NXTLIN (#5C55), MEM (#5C68), T\_ADR (#5C74) не инициализируются (= #00); PIP (#5C39)=#1E, а NMIADR (#5CB0)=#AA вместо #00.

Следует отметить, что многие адреса используются DOS в разные моменты для разных целей, например, сохранения регистров и др. Более подробно описаны переменные, содержащие полезную для программиста информацию.

## **ОБЛАСТЬ СВЯЗИ С ДИСКОВЫМ ФАЙЛОМ**

Если открыт дисковый файл, дополнительно выделяется следующая область:

DEC	HEX	ЗНАЧЕНИЯ
23866	#5D3A	292 байта - область канала связи с дисковым файлом:
23866	#5D3A	2 байта - адрес процедуры записи в канал [15630 (#3D0E)]
23868	#5D3C	2 байта - адрес процедуры чтения из канала [15622 (#3D06)]
23870	#5D3E	1 байт - имя канала - "D"
23871	#5D3F	8 байт - #F4, #32, #33, #38, #24, #01, #00, #4C (поврежденная строка РОКЕ 238??,76 (?))
23879	#5D47	3 байта - тип файла: 1B, 00, FF - последовательный, для записи; 00, 00, 00 - последовательный, для чтения; 1B, 00, 7F - с произвольным доступом
23882	#5D4A	16 байт - параметры файла из каталога (см. #5CDD: имя, тип, адрес, длина, количество секторов, местоположение на диске)
23898	#5D5A	4 байта - #3A, #EC, #36, #34 или #11, #00, #00, #00 (?)
23902	#5D5E	256 байт - буфер связи с файлом; содержит текущий фрагмент текста
24158	#5E5E	байт-разделитель [#80]
24159	#5E5F	Начало BASIC-программы, если был открыт один дисковый файл

Вообще говоря, довольно мощная система дисковых файлов (#) слабо используется программистами (мне, например, не попадалось ни одной программы с её применением). В качестве иллюстрации открывающихся возможностей приведу один фокус с дисковыми файлами, позволяющий просто решить проблему распечатки (принтеры у нас пока слабо распространены в народе). Итак, попробуйте запустить следующую программу:

```
10 CLEAR 28000
20 RANDOMIZE USR 15619: REM: OPEN #4,"printer",W
30 POKE 23580, PEEK 23582 : POKE 23581, PEEK 23583
40 RANDOMIZE USR 15619: REM: LOAD "mons3" CODE 40000
50 RANDOMIZE USR 40000
```

Естественно, на дискете должны присутствовать загружаемые коды и достаточное свободное пространство для открытия дискового файла. После запуска MONS'a введите команду "T", на вопрос "Printer?" ответьте "?", и Вы получите на дискете некоторое число 16-секторных блоков "printer #", содержащих дизассемблированный текст в ASCII формате (подчеркну, в отличие от записи в формате GENS, непригодном для распечатки). Некоторое неудобство представляет необходимость после окончания вывода выйти из программы и закрыть дисковый файл (CLOSE #4), чтобы записался остаток информации из буфера. Однако, без этого можно обойтись, если допечатать в конце порядка 255 лишних байт.

Данный метод использует переключение потоков в поле STRMS, с канала принтера #3 ("P") на открытый канал дискового файла, в нашем примере #4. Аналогичным образом можно переключить большинство программ, осуществляющих печать через канал #3. Для графики, естественно, программа должна быть сконфигурирована под нужный тип принтера. Затем остается только скопировать полученные файлы на IBM PC (как это сделать, рассмотрим позже), где их можно объединить, с удобствами отредактировать и распечатать.

## **ЗАПИСЬ ПРИ ПОМОЩИ "MAGIC KEY"**

При помощи кнопки MAGIC практически любая программа может быть записана на дискету в любой момент времени, и затем запущена с того же места. Неважно, как она загружалась первоначально – с магнитной ленты или дисковода. На диск записывается полностью все ОЗУ, а также содержимое регистров процессора. Естественно, что дисковое пространство при этом расходуется не самым экономным образом, хотя для многих игрушек со сложными загрузчиками или кодовыми блоками длиной более 41 KB, использование MAGIC вполне оправдано (тем более, что существуют программы для приведения MAGIC-файлов в нормальный вид).

Работа процедуры MAGIC основана на немаскируемом прерывании, которое в основном ПЗУ не используется из-за ошибки. В контроллере BETA нажатие кнопки MAGIC переключает ПЗУ на DOS и вырабатывает импульс немаскируемого прерывания. Получив сигнал NMI, процессор немедленно откладывает все остальные дела, заносит на стек адрес следующей команды (для возврата) и переходит на адрес #0066, и далее на подпрограмму обработки (см. адреса ПЗУ).

Чтобы сохранить значение регистров (AF, BC, DE, HL, IX, IY, BC', DE', HL', AF', I, R), их заносят в стек. Значение триггера прерывания IFF запоминается вместе с регистром I как флаг P/V. Глубина стека составляет 13 слов, а с учетом работы самой процедуры записи - 21 слово (от значения SP в момент нажатия MAGIC). Запись начинается с сохранения первых 512 байт экранной области в 2-х секторах нулевой дорожки (#0A, #0B). Затем в эту область экрана считывается системный сектор дискеты и организуется рабочая область DOS (стек). Далее определяется начало свободной области на диске и происходит запись всего ОЗУ, начиная с экрана; наличие свободного места на диске не проверяется. В каталоге записывается файл с именем "@", типом "C"; в поле адреса указывается адрес стека (значение SP). После записи (если она прошла успешно, и если был включен режим DOS), состояние программы восстанавливается так же, как при запуске MAGIC-файла. Процедура MAGIC записывает в последнюю ячейку буфера принтера 23551 (#5BFF) - #C9 (RET), а в системные ячейки сканирования клавиатуры 23552 (#5C00) и 23553 (#5C01) соответственно #16 и #10.

В фирменной инструкции указывается, что для некоторых программ имя файла должно начинаться с "\$". Этот символ означает, что при запуске будет установлен режим прерываний IM2. Программа не может самостоятельно определить, какой тип прерываний был до записи, поэтому необходимо проверить, работает ли программа с обычным именем, и если нет - поставить в начале имени символ "\$".

Для записи при помощи MAGIC желательно использовать чистый диск, так как в случае аппаратного сбоя или особенностей программы, диск может быть испорчен. Некоторые программы не имеют стека достаточной глубины (42 байта), и при записи стек затирает саму программу. Иногда успешная запись зависит от момента нажатия MAGIC, ведь при работе программы глубина стека меняется; так что при неудаче стоит попробовать повторно. Проверку области стека или появления байта #C9 по адресу #5BFF можно использовать специально для защиты от применения MAGIC (попробуйте, например, записать MAGICом программу DCU 2.02).

При загрузке MAGIC-файла на экране так же, как при записи, организуется рабочая область для DOS. Из каталога диска считываются параметры файла. Значение из поля адреса (байты #09, #0A) заносится в регистр SP. В зависимости от первого байта имени устанавливается тип прерываний IM1 или IM2, если имя начинается с "\$" (прерывания пока запрещены). Далее из стека восстанавливаются значения регистров в порядке, обратном записи (см. выше). Упрощенно процедура восстановления выглядит следующим образом:

```

POP    AF
LD     R,A
POP    AF           ; вектор прерываний I и триггер прерывания
LD     I,A
EX     AF,AF'       ; IFF (как значение флага P/V) сохраняется в AF'
POP    AF           ; восстановление альтернативных регистров
POP    HL
POP    DE
POP    BC
EXX
POP    IY           ; восстановление основных регистров
POP    IX
POP    HL
POP    DE
POP    BC
EX     AF,AF'       ; восстановление флага P/V, содержащего
                    ; значение IFF
JP     PO,ADR
EI           ; разрешить прерывания, если P/V=1
POP    AF
RET           ; переход по адресу, который был в момент NMI

```

Существуют программы, которые преобразуют MAGIC-файл в обычный, с BASIC - загрузчиком для ленты или диска. В них для запуска используется аналогичная процедура.

## **ОСОБЕННОСТИ ФОРМАТИРОВАНИЯ TR-DOS**

Форматирование, осуществляемое TR-DOS, является упрощенным вариантом описанного в приложении стандартного. Основным отличием является полное отсутствие индексного поля, т.е. дорожка начинается сразу с первого пробела (12 байт) и идентификатора 1-го сектора. Кроме того, в идентификаторе сектора не устанавливается байт стороны диска - он всегда равен #00, а сторона определяется аппаратно. Отсутствие индексного поля - основная причина, по которой диск со SPECTRUMа невозможно прочесть на других ЭВМ, например, IBM PC. Существующие на IBM программы для форматирования дисков в формате TR-DOS (например, HOFORMAT) создают индексное поле, которое никак не влияет на работу с диском на SPECTRUM, но позволяет читать и записывать их на IBM (при помощи известной программы HOBETA). На SPECTRUMе определить, что диск отформатирован при помощи HOFORMAT, можно по тому, что чистые сектора заполнены не нулевым байтом, а #F6.

Чередование секторов как правило стандартное (2:1), т.е. расположение секторов на дорожке 1-9-2-10-...-16, однако при помощи программы DCU можно отформатировать диск в режиме "fast", когда сектора следуют по порядку. При мультисекторных операциях это ускоряет работу с диском, но если программа обрабатывает сектора по одному, может существенно её замедлить.

Особенности форматирования могут использоваться для создания защищенных от копирования дисков. Например, DISTRIBUTOR переформатирует 159-ю дорожку, создавая в начале сектора под номером #06 длиной 128 байт и с номером головки равным 1; затем остальные сектора без чередования (1,2,3...), для 16-го сектора также задан номер головки равный 1. Программа DCU 2.12 подобным образом переформатирует нулевую дорожку, создавая в начале сектор с номером #AD, длиной 128 байт, а длину последнего (16-го) уменьшает также до 128.

Стандартным образом любая дорожка может быть отформатирована при помощи процедур из DOS ПЗУ, например, так:

```
XOR    A
LD      (#5CD8),A      ; для проверки дорожки после форматирования
LD      (#5CDA),A      ; для одностороннего форматирования
LD      (#5CD6),A      ; обнулить число дефектных секторов
LD      E,NTRK         ; номер дорожки-1 (для нулевой NTRK=#FF)
LD      B,1            ; число формируемых дорожек
LD      HL,#208A       ; адрес процедуры форматирования
CALL    DOS
LD      A, (#5CD6)     ; В регистре A - число дефектных секторов
.....
DOS     PUSH    HL      ; обращение к DOS ПЗУ
        JP      #3D2F
```

Данная процедура, для выборочного форматирования, может быть использована в программе, например, для восстановления дефектных дисков (естественно, что до форматирования дорожки нужно считать с неё всю доступную информацию, а после - записать её обратно). Представляет интерес также форматирование дополнительных дорожек за пределами обычных 80-ти (как правило, удастся отформатировать еще 2-3 дорожки).

Нестандартное форматирование осуществить значительно сложнее. Для этого используются процедуры #2F79 (выдача команды на ВГ93) и #3FB0 (блочная запись). На контроллер дается команда "запись дорожки", однако дополнительно потребуется ещё ряд процедур по проверке готовности диска, контроль результатов и др. Вся записываемая информация для целой дорожки, включая индексы, пробелы, области данных, должна быть предварительно сформирована в памяти (о структуре служебной информации - см. приложение).

Дорожка может быть считана при помощи процедуры #3F09 (также требуется ряд предварительных операций). В память кроме данных считывается вся служебная информация, которая подробно описана в приложении; метод форматирования не играет роли, поэтому можно прочитать не только TR-DOS, но и любые дискеты, в т.ч. защищённые от копирования. В

принципе, с использованием команд "чтение дорожки" и "запись дорожки" возможно, создать универсальный копировщик.

При чтении дорожки необходимо учесть следующее:

- ✓ при выполнении команды "чтение дорожки" БИС ВГ93 не проверяет контрольные суммы, поэтому дискеты, которые в нормальном режиме читаются не с первого раза, могут быть считаны неправильно;

- ✓ пробелы между полями не синхронизируются, поэтому могут читаться каждый раз по-разному, но все равно в виде последовательности одинаковых байт; число байт в пробелах может также незначительно отличаться;

- ✓ синхроимпульс записан специфическим образом, поэтому первый байт может читаться по-разному (#14, #C2), но два последующих - всегда #A1, #A1; поэтому идентификатор легко обнаружить по последовательности #A1, #A1, #FE, а данные - по #A1, #A1, FBh;

- ✓ неформатированная дорожка читается в виде беспорядочной последовательности байт, в которой отсутствуют указанные выше метки.

## ***ДИСКОВЫЕ ПРОГРАММЫ***

Благодаря использованию изложенных в заметке принципов, появляется все больше программ для SPECTRUMа, адаптированных к работе с TR DOS. Приведем список наиболее известных:

### **СИСТЕМНЫЕ**

ARTIST 2 Версия для DOS - Н. Родионов. Графический редактор; имеет больше возможностей, чем STUDIO, но менее удобен.

ART STUDIO v1.5 Разработчик James Hutchby (ОСР) 85 г. Версии для DOS - Н.Родионов; Kordial 91 г. (с инсталляцией). Удобный графический редактор; позволяет записать файл на прежнее место.

BDJTEST (Beta-disc's test) Разработчик Starocoltsev Eugen, 90 г. Показывает номера дорожек на диске, которые не читаются.

BETA COMMANDER v1.4. Разработчик Trubinov I.U. 91 г. BOOT - оболочка для работы с файлами: запуск, копирование и др. Удачно оформлены функции переименования (в т.ч. название диска) и копирования, но отсутствует информация о файлах.

CC (Conver commander v3.02R). Разработчик Ю. Ковалевский, 92 г. Оболочка для работы с файлами, имеет много функций, в т.ч. встроенный "диск-доктор", русифицированная, но великовата.

COPY Разработчик Lebedev S.M. 91 г. Копировщик дисков  $A \leftrightarrow B$  и  $A \rightarrow A$ ; копирует блок около 41 KB за раз, благодаря размещению программы в области экрана.

GENS 4.51 Разработчик HiSoft v4.1 87 г., beta-disk версия MOA 90. Кроме загрузки/записи программ, позволяет считывать исходный текст частями и осуществлять ассемблирование прямо на диск!

DCU 2.12 (Disc control utility V2.12). Разработчик Piter Ltd, Н. Родионов 91 г. Форматирование (в т.ч. на 86 дорожек) и контроль дисков; установка меток дистрибутива.

DENCOPY Разработчик Densoft 89 г. Копирование на ленту файлов типа "MAGIC", с загрузчиком.

IBM Разработчик Lebedev S.M., Krotov V.V. (for "Spark"). Одна из (программ для копирования данных с IBM дискет на TR-DOS; читает по дорожкам; поддерживает режимы 2D, 1D, 2S, 1S.

MASTER-KEY Разработчик Kotlarov D.V. (DVK-soft), TRTI v3.2 90 г. Одна из лучших программ типа "disk doctor" просмотр и редактирование данных на диске, имеет функцию поиска).

MOA v1.6. Разработчик MOA Ленинград 90 г. BOOT-оболочка для работы с файлами: запуск, копирование, переименование, просмотр.

MON7 (Advanced monitor v7.05). Разработчик Wolf. Дизассемблер программ непосредственно на диске.

MONS 4 beta. Разработчик HiSoft 87 г., адаптация IC' 90. Предоставляет великолепную возможность-дизассемблирование длинных программ прямо на диск.

OPENER Разработчик VAJ, Львов, 94г. Программа для загрузки, просмотра и редактирования защищенных BASIC-загрузчиков.

PASCAL'D (Паскаль версии HP4TM16) Разработчик HiSoft 84 г., адаптация Dereschuck.

PCOPIER2 Разработчик Н. Родионов 90 г. Копировщик "disc-tape" с выбором файлов.

RAMPRESS Разработчик Ю. Власов, Львов, 94г. Преобразует MAGIC - файлы в программу с загрузчиком для ленты и диска. В отличие от DENCOPY, записывает только один блок кодов, компрессируя его. Это экономит не менее 20% дискового пространства, а преобразованная таким образом программа при загрузке выглядит вполне приемлемо.

TLW2 (The Last Word 2) Разработчик Saga System & Myrmidon 86 Русская версия Д.Г.С., Кордиал 90 г. Текстовый редактор с широким набором команд. Связь с DOS через строку BASICa.

TREK A>B Разработчик Kordial 90 г. Удобный копировщик для тех, кто сумел приобрести два дисковода.

WHAM (Music Box). Разработчик MarkTime 85 г., дисковая версия Н. Родионов 90 г. Музыкальный редактор. Кроме записи/загрузки исходных файлов, записывает скомпилированную музыку на диск.

ZX-WORD v1.01. Разработчик Si-soft & Bg-soft 92 г., Харьков. Пакет программ для работы с текстовыми файлами формата IBM. Включает редактор ZX-WORD, имеющий привычный для пользователей IBM вид, но слишком мелкий шрифт (64 символа в строке). Для настройки на версию DOS требуется инсталляция. Включает также:

- ✓ GENSTEXT, TEXTGENS - перекодировщики из формата GENS4 в текстовый и обратно; имеются также для ZEUS и другие;

- ✓ SCRTEXT - перекодировщик файлов SCREEN\$ в графический формат принтера EPSON, позволяет включать в печатаемый текст рисунки;

- ✓ FORMAT - форматирование дискет в формате MS-DOS 3.0;

- ✓ IBM-COPY - обмен файлами между IBM и ZX. Позволяет, как считывать, так и записывать на IBM дискеты.

Незаменимым инструментом для программиста являются дисковые версии MONS4 и GENS4. Работа с ними сразу проявляет все преимущества дисковой системы - процесс разработки программы становится простым и удобным, в отличие от возни с магнитофоном. При помощи команды вставки (\*F, например, \*F 1:text\_ass) можно создавать новую программу из стандартных, отлаженных блоков, записанных на диске. Вместе с директивой ассемблирования на диск (например, A16,5000,1:object) это позволяет создавать большие программы. Монитор MONS4B осуществляет дизассемблирование на диск, если в команде "T" на запрос "Text:" ввести имя в виде [номер диска:название\_файла]. Файл записывается блоками по 16 секторов под заданным именем, после которого дописаны номера 00, 01 и т.д., причем дизассемблируемая программа может быть довольно длинной.

## ИГРОВЫЕ

ACADEMY	- Pete Cooke	86,	disk by	Andy Chernikov	
BALLGAME	- Pete Cooke	88,	disk by	N.Rodionov	
BRAIN STORM	- Pete Cooke	87,	disk by	N.Rodionov	90 г.
COLUMNS	- Piter	91,	разработчик	Красковский&Родионов	
DRAGON BREED	- Activision	90,	disk by	Paul Nosikov	91 г.
DRAGON SPIRIT	- Tengen	89,	disk by	Andy Chernikov	91 г.
EARTHLIGHT	- Pete Cooke	88,	disk by	N.Rodionov	90 г.
F-16 COMBATPILOT	- Digital int.	91,	disk by	Vasilyev Anton	
FIGHTER BOMBER	- Activision		disk by	V.Belyaev	91 г.
GHOSTBUSTERS II	- Activision	89,	disk by	AVF Moscov	
GOLDEN AXE	- Sega&Virgin	90,	disk by	Vasilyev Anton	
HEROES OF LANCE	- U.S. Gold	88,	disk by	Walsoft	92 г.
INDIANA JONES2	- Tiertex		disk by	Vasilyev Anton	91 г.
KLAX	- Tengen	90,	disk by	J.U.Trubinov	91 г.
LEMMINGS	- DMA design	91,	disk by	Andy Chernikov	92 г.
MICRONAUT 1	- Pete Cooke	88,	disk by	N.Rodionov	90 г.
MYTH	- Systems3		disk by	Andy Chernikov	
OPER.THUNDERBOLT	- Ocean&Taito		disk by	Vasilyev Anton	91 г.
R-TYPE	- Irem corp.	87,	disk by	N.Rodionov	90 г.
ROBOCOP	- Ocean	87,	disk by	AVF Moscov	
SIM CITY	- Probe	89,	disk by	DEN corp.	93 г.

SKATE CRAZY	- Gremlin	disk by	Andy Chernikov	91 г.
SUPER SCRAMBLE	- Gremlin	disk by	Vasilyev Anton	91 г.
WINTER EDITION	- Epyx	88, disk by	S. Skorobogatov	90 г.
ZOLYX	- Pete Cooke	87, disk by	N.Rodionov	90 г.

Под дисковыми версиями игр подразумеваются не просто программы, которые грузятся с диска, даже если при этом используется машинно-кодовый загрузчик. Адаптация предполагает наличие хотя бы некоторых из следующих признаков:

1. Выбор конфигурации игры и загрузка соответствующих блоков (например, разных персонажей или обстановки);
2. Дозагрузка последующих уровней (встречается довольно часто);
3. Запись таблицы рекордов на диск (имеется в большинстве программ в адаптации Н.Родионова);
4. Редактирование и сохранение на диске своих вариантов игровой обстановки (например, BALLGAME);
5. Запись отложенной игры для продолжения в дальнейшем (к сожалению, эта возможность, достаточно распространенная в кассетных версиях, в дисковых реализуется редко, например, в ACADEMY и SIM CITY; видимо, предполагается использовать MAGIC).

Нельзя не отметить вклад Николая Родионова из Санкт-Петербурга - как по количеству, так и качеству программ.

## ***SPECTRUM И IBM PC***

Вопрос обмена информацией между SPECTRUMом и IBM интересует многих. В первую очередь это, конечно, различные тексты, которые хотелось бы вводить, просматривать и печатать на обоих компьютерах. Данная проблема решается двумя путями: при помощи программ, работающих с дисками формата MS-DOS на SPECTRUMе (перечислены выше), и программ, работающих на IBM PC с дисками формата BETA TR-DOS.

Полноценная работа с дискетами IBM на SPECTRUMе сопряжена с рядом трудностей. Необходимо, в частности, потратить несколько килобайт ОЗУ для таблицы размещения файлов (FAT), и проводить сложные манипуляции с номерами кластеров в этой таблице. Имеющиеся программы этого класса позволяют форматировать на SPECTRUMе MS-DOS дискеты, копировать на них файлы с обычных дискет и обратно, а также работать с текстами в альтернативной кодировке кириллицы (наиболее распространенная). Однако, часто функции копирования осуществляются по упрощенному алгоритму, в результате могут возникать проблемы с фрагментированными файлами и различными версиями MS-DOS (даже лучшая из программ этого класса IBM-COPY разработки Si-soft записывает файл таким образом, что в MS-DOS v5.0 его можно прочитать только после обработки "Norton Disk Doctor"). Следует отметить, что на физическом уровне дискеты MS-DOS читаются без проблем (при помощи обычной функции "чтение сектора" процедуры #3D13); сложности возникают именно с более сложной файловой системой.

Как показывает практика, более удобно пользоваться программами-эмуляторами TR-DOS на IBM PC. В дистрибутив "ZX-SPECTRUM на IBM PC" входят:

- fst.exe – работа с BETA-дискетой на IBM PC (TYPE, RENAME ...);
- hobeta.exe – работает как NORTON между BETA-дискетами и IBM PC;
- hofrmt.exe – форматирует BETA-дискеты на IBM PC;
- pctobeta.exe – преобразует файлы IBM PC в BETA-формат;
- spectrum.exe – эмулятор ZX-SPECTRUM (с дополнительными файлами);
- tapemast.exe – записывает BETA-файлы в образ ленты (tape\_zx.spc) для эмулятора;
- teledisk.exe – программа копирования любых дискет на IBM PC, разработка Sydex; включает несколько вспомогательных файлов.

Из перечисленных программ наибольший интерес представляет программа - оболочка "НОВЕТА" фирмы InterComplex. По внешнему виду и выполняемым функциям она напоминает известный "Norton Commander", однако работает с дискетами TR-DOS. Для копирования IBM - файлов их необходимо предварительно обработать программой "PCTOBETA", которая дописывает им стандартный BETA-заголовок. Для копирования дискет довольно удобен "TELEDISK", копирующий всю дискету целиком в один файл и наоборот. Фанатики SPECTRUMа

могут даже полюбоваться привычной картинкой на экране IBM, используя программу-эмулятор "SPECTRUM" (трудно сказать, кому это может понадобиться, учитывая, что запускаются в нем только кассетные программы, причем далеко не все, а переписать их можно только через дискету)...

Существенным недостатком пакета является то, что ни "HOBETA", ни "TELEDISK" не читают дискет, которые сформатированы обычным образом на SPECTRUMе, а только созданные программой "HOFORMAT" (о причине этого можно прочесть в разделе, посвященном форматированию). В данное время разрабатывается программа, позволяющая форматировать дискеты на SPECTRUMе подобно "HOFORMAT".

## ПРИЛОЖЕНИЯ

Данная информация необходима при анализе DOS ПЗУ, чтобы разобраться с обращениями к портам и работой БИС K1818BG93, а также может быть полезна при настройке платы контроллера.

### АДРЕСА ПОРТОВ ДИСКОВОГО КОНТРОЛЛЕРА "ВЕТА"

#1F запись - регистр команд, чтение - регистр состояния ВГ93;

#3F регистр дорожки ВГ93;

#5F регистр сектора ВГ93;

#7F регистр данных ВГ93;

#FF запись - системный регистр контроллера (ТМ9); режим:

7 6 5 4 3 2 1 0

- X - X X X X X

номер дисководов - DS (0...3);  
сброс ВГ93 - CLR (сброс = 0);  
загрузка головки - HLT (готовность = 1);  
сторона диска - SIDE (SIDE1, нижняя = 0);  
метод/плотность записи (МЧМ = 0/ЧМ = 1).

#FF чтение-готовность ВГ 93:

бит 6 - DRQ (запрос данных = 1);

бит 7 - INTRQ (выполняется команда = 0).

#7FFD блокировка включения DOS:

(#FD) бит 4 = 0 - блокировка, бит 4 = 1 - разрешение;

(ПРИМЕЧАНИЕ: адрес дешифруется линиями A01+A15, однако из BASICа OUT 253,0 (16) дает тот же эффект). В SPECTRUM-128 данный порт предназначен для управления банками памяти, а 4-й бит - сменой ПЗУ. Для SPECTRUM-48 этот порт в принципе бесполезен, т.к. блокировка просто приводит к зависанию системы при обращении к DOS. Его легко можно приспособить для своих целей, например, управления дополнительным банком ОЗУ (не забывая о блокировке DOS).

Напомню, что порты (кроме #FD) доступны только в режиме DOS, т.е. для процедур из DOS ПЗУ, что весьма затрудняет их использование. Например, вывод в порт, номер которого задан регистром C, может быть осуществлен при помощи подпрограммы по адресу #2A09 (для TR-DOS v5.01) через точку входа #3D2F (см. выше).

### ОПИСАНИЕ БИС КОНТРОЛЛЕРА НГМД 1818 ВГ 93

Микросхема обеспечивает связь НГМД с ЭВМ: установку головки на заданную дорожку, чтение/запись сектора или дорожки с вычислением контрольных сумм (CRC), управляет скоростью перемещения головки и предкомпенсацией записи. Поддерживаются два формата записи: одинарной (ЧМ/FM) и двойной плотности (МЧМ/MFM) и различные типы НГМД. БИС K1818BG93 является аналогом WD1793 фирмы "Western Digital".

#### ТЕХНИЧЕСКИЕ ХАРАКТЕРИСТИКИ 1818 ВГ 93 6K0.348.877-08 ТУ:

Технология

NMOP

Рабочая температура

+10...+70 °C

Напряжение питания

5 В; 12 В ±5%



Потребляемый ток, соответственно	60 мА; 20 мА
Напряжение сигналов: U <sub>вх.</sub> "1"	>2.6 В
U <sub>вх.</sub> "0"	<0.8 В
U <sub>вых.</sub> "1"	>2.8 В
U <sub>вых.</sub> "0"	<0.45 В
Максимальная емкость нагрузки	100 пФ
Потребляемая мощность, не более	500 мВт
Скорость обмена информацией:	
- при одинарной плотности записи	250 кБит/с
- при двойной плотности записи	500 кБит/с
Тактовая частота:	
- для 133 мм НГМД	1 МГц
- для 208 мм НГМД	2 МГц
Корпус (40-выводный DIP)	2.123.40-2

### **НАЗНАЧЕНИЕ ВЫВОДОВ МИКРОСХЕМЫ**

№	ОБОЗНАЧЕНИЕ*	НАЗНАЧЕНИЕ
1	BS	подложка (контроль напряжения смещения)
2	/W (/WE,/WR)	вход - разрешение записи информации с шины данных в выбранный регистр
3	/CS	вход - выбор микросхемы
4	/R (/MR,/RD)	вход - разрешение чтения информации из регистра на шину данных
5,6	A0,A1	адресная шина - выбор регистра:
	A1 A0	название регистра      назначение
	0 0	чтение регистр статуса      текущее состояние
		запись регистр команд      текущая команда
	0 1	регистр дорожки      текущий номер дорожки (00-79)
	1 0	регистр сектора      текущий номер сектора (01-16)
	1 1	регистр данных      данные записи/чтения
7-14	DB0-DB7 (D0-D7)	8-разрядная двунаправленная шина данных
15	STEP	выходной импульс перемещения головки на 1 шаг
16	DIRC	направление перемещения головки (1-к центру)
17	SL (EARLY)	выходной сигнал управления предкомпенсацией (указывает, что импульс данных WD должен быть сдвинут влево)
18	SR (LATE)	выходной сигнал управления предкомпенсацией (указывает, что импульс данных WD должен быть сдвинут вправо)
19	/CLR (/RST,/RES)	вход - сброс; в регистр команд записывается %0000 0011, на время действия сигнала выход INTRQ = 0, затем выполняется команда "восстановление" независимо от готовности НГМД; в регистр сектора записывается "#01", в регистр дорожки - "#00"
20	GND	корпус
21	U1	напряжение питания +5 В
22	/TEST	понижение скорости перемещения головки, если TEST = 0, до 400 мс/шаг (см.команды ВГ93)
23	HRDY (HLT)	вход - готовность головки (головка в рабочем положении, т.е. загружена)
24	CLK	вход - сигнал тактовой частоты (для дисководов 133 мм = 1 МГц)
25	RSTB (RG)	выход - строб чтения, подтверждающий прием данных от НГМД - признак преамбулы (RSTB = 1 после приема 2-х байтов "0" при одинарной плотности и 4-х "0" или "1" - при двойной плотности)
26	S (RC, RCLK)	вход - синхроимпульсы воспроизведения, выработанные из RAWR
27	/RAWR (/RA)	вход - импульсы данных чтения от НГМД

28	HLD	выход - загрузка магнитной головки
29	TR43	выход - указывает, что головка находится на дорожке с номером > 43 (44...79)
30	WSTB (WG)	выход - строб разрешения записи (= 1 во время записи на диск)
31	WD	выход - импульсы данных, записываемых на НГМД
32	CPRDY (READY)	вход - готовность дисководов выполнять запись или чтение (пока дисковод не выбран и не набрал обороты его сигнал READY = 0; при этом микросхема вырабатывает сигнал INTRQ)
33	WF/DE	двухнаправленная шина: при записи на диск (WSTB = 1) работает как вход WF - "ошибка записи", и при WF=0 запись прекращается; при чтении с диска (WSTB = 0) работает как выход DE - "разрешение данных", на котором после загрузки головки и установки HRDY = 1 будет "0". Выход с открытым стоком: требует Rнагр. Около 10 кОм подключенного к +5 В
34	/TR00	вход - сигнал "0" поступает от дисководов, когда головка установлена на нулевую дорожку
35	/IP (/INDEX)	вход - индексный импульс "0" от дисководов (указывает на начало дорожки)
36	/WPRT	вход - сигнал "0" от дисководов поступает, если диск защищен от записи
37	/DDEN	вход - указание плотности записи: "0" - двойная, "1" - одинарная
38	DRQ	выход - запрос данных; в режиме чтения указывает, что микросхема готова выдать очередной байт данных чтения; в режиме записи - готовность принять очередной байт для записи. Выход с открытым стоком
39	INTRQ	выход - запрос прерывания; готовность микросхемы (INTRQ = 0, когда закончено выполнение команды или считан регистр состояния). Выход с открытым стоком
40	U2	напряжение питания +12 В

\* ПРИМЕЧАНИЕ: Условное обозначение выводов в разных источниках отличается.

### ***СИСТЕМА КОМАНД БИС КОНТРОЛЛЕРА 1818 ВГ 93***

В БИТА-интерфейсе регистр команд/состояния имеет адрес #1F.

Микросхема обеспечивает выполнение 11 команд:

BIN	HEX	
0000 hvrr	00...0F	восстановление
0001 hvrr	10...1F	поиск
001t hvrr	20...3F	шаг головки (в том же направлении)
010t hvrr	40...5F	вперед
011t hvrr	60...7F	назад
100m seca	80...9F	чтение сектора
101m sec0	A0...BF	запись сектора
1100 0e00	C0,C4	чтение адреса
1110 0e00	E0,E4	чтение дорожки
1111 0e00	F0,F4	запись дорожки (при форматировании)
1101 iiii	D0...DF	принудительное прерывание

Значение битов команды:

гг - скорость позиционирования головки:

r1	r0	T шаг
0	0	6 ms
0	1	12 ms
1	0	20 ms
1	1	30 ms

(для Fclk=1Mrq. При сигнале TEST (= 0) на соответствующем входе период составляет около 400ms и не меняется)

v - проверка номера дорожки после позиционирования

h - загрузка головки

t - изменение номера дорожки (в рег. дорожки) после каждого шага

a - тип адресной метки (защита данных) (0-#FB, 1-#F8)

c - проверка номера стороны диска при идент. индексной области

e - задержка после загрузки головки (сигнала HLD) на 30 мс

s - сторона диска

m - мультисекторная операция (обрабатываются несколько секторов)

i - условие прерывания:

i0 - по переходу привода в состояние "Готов"

i1 - по переходу привода в состояние "Не готов"

i2 - по индексному импульсу

i3 - немедленное прерывание

В TR-DOS как правило для команд используется маска OR #18.

Команда "ВОССТАНОВЛЕНИЕ" осуществляет установку головки на нулевую дорожку магнитного диска. Если через 256 шагов не появится сигнал TR00, команда заканчивается. Всегда выполняется при сбросе контроллера независимо от готовности накопителя.

Команда "ПОИСК" - в регистре дорожки находится текущий номер дорожки, а в регистр данных записывается требуемый. Перемещение осуществляется до их совпадения. Должен быть установлен флаг v (бит D2 = 1).

Команда "ШАГ" обеспечивает выдачу импульса STEP на перемещение магнитной головки на один шаг в том же направлении, а "ШАГ ВПЕРЕД", "ШАГ НАЗАД" - изменение сигнала DIRC - направление перемещения.

Команда "ЧТЕНИЕ СЕКТОРА" обеспечивает чтение сектора, номер которого на текущей дорожке записан в регистр сектора. Сторона диска задается флагом s (0,1). При установленном флаге m - мультисекторная операция (номер сектора автоматически увеличивается, пока не будет достигнут конец дорожки, и операция повторяется). Длина сектора определяется при форматировании (см. далее). Флаг a – тип адресной метки защиты данных. При a=1 записывается байт #F8, разрешающий стирание сектора, при a=0 - #FB, запрещающий. Вначале считывается идентификатор сектора (см. далее); если адресная метка не обнаружена, в регистре состояния устанавливается признак "массив не найден". Если совпали номера дорожки, сектора, стороны и код CRC, происходит чтение данных: очередной байт выдается в регистр данных и сопровождается сигналом DRQ (готовность данных). Регистр должен быть считан до приема следующего байта; в противном случае в регистре состояния устанавливается флаг "потеря данных". В конце чтения сравнивается контрольная сумма CRC; если не совпадает, устанавливается флаг "ошибка в контрольном коде", мультисекторная операция прекращается.

Команда "ЗАПИСЬ СЕКТОРА" выполняется подобно предыдущей в части идентификации сектора. Сигнал DRQ генерируется, запрашивая первый байт данных. Затем вычисляются 22 байта для двойной (11 - для одинарной) плотности - пробел между индексной областью и данными. После этого, если DRQ обслужен и регистр данных получил байт, выдается строб WSTB и записываются данные, начиная с нулевых байт и адресной метки (см. формат). Регистр данных должен получать очередной байт на каждый сигнал запроса данных DRQ, со скоростью записи. Если байт не получен, в регистре состояния устанавливается флаг "потеря данных", а на диск записывается байт "#00". После данных записываются два байта контрольной суммы CRC,

вычисленной контроллером, как циклическая сумма  $A=X^{15}+X^{12}+X^5+1$ , и байт - пробел. Сигнал WSTB снимается (устанавливается в 0).

Команда "ЧТЕНИЕ АДРЕСА" выполняет считывание 6 байт индексной области (включая CRC), и передачу их компьютеру. Если CRC не совпадает, устанавливается бит состояния "ошибка CRC", и чтение продолжается. При выполнении этой команды байт из регистра дорожки запоминается в регистре сектора. По окончании как обычно, устанавливается сигнал готовности INTRQ и сбрасывается бит состояния "занято".

Команда "ЧТЕНИЕ ДОРОЖКИ" выполняет считывание всей информации на дорожке, включая индексный массив, CRC, идентификаторы, пробелы и данные. При этом не выдается строб чтения и не проверяется CRC, что позволяет использовать данную команду в диагностических целях.

Команда "ЗАПИСЬ ДОРОЖКИ" предназначена для форматирования ГМД. Вся информация для этой процедуры должна быть сформирована ЭВМ, включая пробелы и поля индексов и данных со всеми метками. Любая последовательность данных записывается, за исключением байт #F5-#FE, которые интерпретируются как управляющие адресные метки данных. В момент приема в регистр данных байтов #F8-#FE для режима ЧМ (FM) или байта #F5 для режима МЧМ (MFM), генерируется контрольный код. При появлении байта #F7 код CRC (2 байта) записывается на диск. Таким образом, при форматировании эти байты не могут присутствовать в области пробелов, индексных массивов и данных (см. формат).

Команда "ПРИНУДИТЕЛЬНОЕ ПРЕРЫВАНИЕ" задается для завершения любой выполняемой команды. В отличие от других команд она может быть записана в регистр команд в любой момент времени. Условие выполнения зависит от младших битов, команды (i0...i3): если они равны "0", выполнение текущей команды прекращается и INTRQ не вырабатывается. При i0=1 прерывание выполняется после перехода сигнала CPRDY из низкого уровня в высокий, при i1=1 - наоборот, из "1" в "0". При i2=1 - по поступлению индексного импульса (IP), т.е. началу дорожки. При i3=1 происходит немедленное прерывание команды. После выполнения этих условий формируется сигнал INTRQ.

### **СТРУКТУРА РЕГИСТРА СОСТОЯНИЯ 1818 ВГ 93**

После выполнения команды в регистре состояния записывается байт состояния контроллера, зависящий от результата выполнения команды:

КОМАНДА	РАЗРЯД РЕГИСТРА СОСТОЯНИЯ							
	7	6	5	4	3	2	1	0
вспомогательная	R	P	H	F	C	T	I	Q
чтение адреса	R	0	0	N	C	W	D	Q
чтение сектора	R	0	A	N	C	W	D	Q
чтение дорожки	R	0	0	0	0	W	D	Q
запись сектора	R	P	E	N	C	W	D	Q
запись дорожки	R	P	E	0	0	W	D	Q

Значение флаговых битов:

- R - готовность накопителя (1=не готов)
- P - защита записи (1=диск защищен)
- H - загрузка головки (1=загружена)
- E - ошибка записи (1=ошибка)
- A - тип АМ (1=запись со стиранием)
- F - ошибка поиска (1=позиция не верна)
- N - массив не найден (1=не найден)
- C - ошибка в контрольном коде CRC (1=CRC не верен)
- T - головка в исходном положении (1=нулевая дорожка)
- W - потеря данных (1=данные потеряны)
- I - индексный импульс (1=импульс)
- D - запрос данных (1=запрос данных)
- Q - занято (1=идет выполнение команды)

## ФОРМАТ ЗАПИСИ НА ДИСК



### ПРИМЕР ФОРМАТА МАССИВА ДАННЫХ 1-го СЕКТОРА ДЛЯ ДВОЙНОЙ ПЛОТНОСТИ

ИНДЕКСНОЕ ПОЛЕ	
80 байт #4E	= послеиндексный промежуток (5-й пробел)
12 байт #00	= синхропромежуток
3 байт #F6	= запись C2 - синхронизирующий АМ
1 байт #FC	= индексная метка (маркер индекса)
50 байт #4E	= первый пробел
1-Й СЕКТОР	
идентификатор 1-го сектора	
12 байт #00	= синхропромежуток
3 байт #F5	= запись A1 - синхроимпульс
1 байт #FE	= адресная метка идентификатора
1 байт nn	= номер дорожки, или цилиндр (#00-#4F)
1 байт nn	= номер головки (сторона диска-#00,#01)
1 байт nn	= номер сектора (#01-#10) (не с нулевого!)
1 байт xx	= длина сектора #00-#03:
	#00 - 128 байт
	#01 - 256 байт
	#02 - 512 байт
	#03 - 1024 байт
1 байт #F7	= запись 2-х байтов вычисленной ранее контрольной суммы (CRC) идентификатора
22 байт #4E	= второй пробел
поле данных 1-го сектора	
12 байт #00	= синхропромежуток
3 байт #F5	= запись A1 - синхроимпульс
1 байт #FB	= адресная метка данных
NN байт ??	= данные в соответствии с длиной сектора
1 байт #F7	= запись 2-х байтов вычисленной ранее контрольной суммы (CRC) данных
54 байт #4E (#00)	= третий пробел
.....	продолжение записи до прерывания (4-й пробел до начала индексного импульса)

## **ПРИМЕР ФОРМАТА МАССИВА ДАННЫХ 1-го СЕКТОРА ДЛЯ ОДИНАРНОЙ ПЛОТНОСТИ**

<b>ИНДЕКСНОЕ ПОЛЕ</b>	
40 байт #FF (00)	
+6 байт #00	= пятый пробел от начала индексного импульса
1 байт #FC	= индексная метка (1-й индексный массив)
26 байт #FF (#00)	= первый пробел
<b>1-Й СЕКТОР</b>	
<b>идентификатор 1-го сектора</b>	
6 байт #00	= пробел
1 байт #FE	= адресная метка индексных данных
1 байт nn	= номер дорожки (#00-#4F)
1 байт nn	= номер головки (сторона диска-#00,#01)
1 байт nn	= номер сектора (#01-#1A)
1 байт xx	= длина сектора #00-#03: #00 - 128 байт #01 - 256 байт #02 - 512 байт #03 - 1024 байт
1 байт #F7	= запись 2-х байтов вычисленной ранее контрольной суммы (CRC) идентификатора
<b>поле данных 1-го сектора</b>	
11 байт #FF (#00)	
+6 байт #00	= второй пробел
1 байт #FB	= адресная метка данных
NN байт ??	= данные в соответствии с длиной сектора
1 байт #F7	= запись 2-х байтов вычисленной ранее контрольной суммы (CRC) данных
27 байт #FF (#00)	= третий пробел
.....	продолжение записи до прерывания (4-й пробел до начала индексного импульса)

## **СЛУЖЕБНЫЕ БАЙТЫ ИНДЕКСНОЙ ОБЛАСТИ**

<b>Байт</b>	<b>Назначение</b>	
	<b>В режиме ЧМ</b>	<b>В режиме МЧМ</b>
#F5	не допускается	запись метки #A1 вычисляется CRC
#F6	не допускается	запись метки #C2
#F7	записывается 2 байта вычисленной ранее CRC	записывается 2 байта вычисленной ранее CRC
#F8-#FB	запись #F8-#FB с CLK=#C7, вычисляется CRC	запись #F8-#FB
#FC	запись #FC с CLK=#D7 (индексная метка перед первым индексным массивом)	запись #FC (индексная метка перед первым индексным массивом)
#FD	запись #FD с CLK=#FF	запись #FD
#FE	запись #FE с CLK=#C7 (индексная метка данных в начале индексного массива) вычисляется CRC	запись #FE (индексная метка данных в начале индексного массива)
#FF	запись #FF с CLK=#FF	запись #FF

Подробнее см. "В.А. Коваленко и др. БИС контроллера KP1818BG93 для НГМД./ Микропроцессорные средства и системы - 1986 № 3, с.3-8".

**Пример 0-й дорожки TR-DOS - диска:**

		#0C #E4 (12 байт)	первый пробел (несинхронизирован)
П Е Р В Ы Й  С Е К Т О Р	И Д Е Н Т И Ф И К А Т О Р	#E0 #00 (11 байт)	Синхропромежуток (несинхронизирован)
		#14 #A1 #A1	синхроимпульс
		#FE #00 #00 #01 #01	АМ индекса номер цилиндра номер головки номер сектора длина сектора
		#FA #0C	контрольная сумма
		#4E (22 байт)	
		второй пробел	
		#00 (12 байт)	
	П О Л Е  Д А Н Н Ы Х	#14 #A1 #A1	синхроимпульс
		#FB #00 (256 байт)	АМ данных данные
		#E1 #22	контрольная сумма
		#4E (60 байт)	
		третий пробел	
		#00 (12 байт) .....	следующие сектора (#09, #02, ... #10) размечаются аналогично

**КОНТАКТЫ СОЕДИНИТЕЛЯ ДИСКОВОДА**

Соединитель дисководов совместим с SHUGART:

2	4																	34
○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○
○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○
1	3																	33

- контакты "GND"

- 2 - Не подключен (NC)
- 4 - Не подключен (NC)
- 6 - Выбор дисководов "D:" (CS3)
- 8 - Индекс (IP)
- 10 - Выбор дисководов "A:" (CS0)
- 12 - Выбор дисководов "B:" (CS1)
- 14 - Выбор дисководов "C:" (CS2)
- 16 - Загрузка головки (LOAD HEAD или MOTOR ON)
- 18 - Направление (DIRC)
- 20 - Шаг (STEP)
- 22 - Запись данных (WR DATA)
- 24 - Разрешение записи (WR GATE)
- 26 - Нулевая дорожка (TRACK 00)
- 28 - Защита записи на диске (WPRT)
- 30 - Считывание данных (RD DATA)

32 - Сторона 1 (SIDE1)

34 - Готовность привода (RDY)

Ю. Поморцев /Львов 1994 г./

Отпечатано в ИПК "Московская правда",  
Москва, ул. 1905 года, дом 7.  
Тир. 10000. Заказ 4259