В.А. Павлов

Подсистема дисковой памяти ПК

Учебное пособие для вузов

Рекомендовано
учебно-методическим объединением вузов
по университетскому политехническому образованию
в качестве учебного пособия
для студентов, обучающихся по специальности 220100 -
"Вычислительные машины, комплексы, системы и сети"

Саров
СарФТИ
2002

УДК 681.327.634
ББК 32.973
П12

Рецензенты:
д-р техн. наук, профессор кафедры МО ЭВМ Нижегородского государственного университета Ю.Л. Кетков;
к-т техн. наук, доцент кафедры "Компьютерные системы и сети" МГТУ им. Н.Э. Баумана И.В. Баскаков;
кафедра "Вычислительная техника" Нижегородского государственного технического университета

Павлов Виктор Александрович
П12. Подсистема дисковой памяти ПК. Учебное пособие. СарФТИ, Саров, 2002. 270с.: ил.

В учебном пособии рассмотрены основные характеристики устройств внешней памяти, логическая структура дисков, системная поддержка внешней памяти, память на гибких и жестких дисках, особенности работы с устройствами, подключенными к шине PCI, устройства массовой памяти на сменных дисках и проблемы, связанные с установкой, обслуживанием и тестированием дисков. Пособие предназначено для студентов СарФТИ, обучающихся по специальности 220100 - "Вычислительные машины, комплексы, системы и сети" и родственным ей специальностям..

©В.А. Павлов, 2002г.

Содержание

Предисловие -- -- -- -- -- -- -- -- -- 6
1. Внешняя память компьютера -- -- -- -- -- -- 7
1.1. Основные характеристики внешней памяти -- -- -- -- -- 7
1.2. Конструктивы и питание -- -- -- -- -- -- -- 9
1.3. Компоненты накопителей -- -- -- -- -- -- -- 10
1.4. Принципы хранения информации -- -- -- -- -- -- 13
1.5. Хранение информации на магнитных дисках-- -- -- -- -- -- 14
1.6. Контроллеры дисковых накопителей -- -- -- -- -- -- 16
1.7. Параметры дисковых накопителей -- -- -- -- -- -- 18
Контрольные вопросы -- -- -- -- -- -- -- -- 20
2. Логическая структура диска -- -- -- -- -- -- -- 21
2.1. Разделы и логические диски -- -- -- -- -- -- -- 21
2.2. Файловые системы FAT-12, FAT-16 и FAT-32 -- -- -- -- -- 26
2.2.1. Размещение информации на логических дисках -- -- -- -- -- 26
2.2.2. Назначение и внутренняя организация таблиц размещения файлов -- -- 29
2.3. Каталоги файлов -- -- -- -- -- -- -- -- 31
2.4. Возможности системы FAT -- -- -- -- -- -- -- 34
2.5. Сжатие дисков -- -- -- -- -- -- -- -- -- 35
Контрольные вопросы -- -- -- -- -- -- -- -- 37
3. Системная поддержка внешней памяти-- -- -- -- -- -- 38
3.1. Сервисы BIOS -- -- -- -- -- -- -- -- -- 38
3.1.1. Традиционный сервис BIOS -- -- -- -- -- -- -- 40
3.1.2. Преодоление предела 528 Мбайт -- -- -- -- -- -- 46
3.1.3. Расширенный сервис BIOS -- -- -- -- -- -- -- 48
3.2. Взаимодействие ОС и прикладных программ с дисками -- -- -- -- 52
3.2.1. Общие сведения -- -- -- -- -- -- -- -- 52
3.2.2. Группа дисковых функций MS DOS -- -- -- -- -- -- 56
3.2.2.1. Классические функции взаимодействия с дисками -- -- -- -- 56
3.2.2.2. Улучшенные функции для работы с дисками -- -- -- -- -- 57
3.2.2.3. Низкоуровневые дисковые функции DOS -- -- -- -- -- 64
3.2.2.4. Примеры использования функций DOS -- -- -- -- -- 65
Контрольные вопросы -- -- -- -- -- -- -- -- 71
4. Память на гибких магнитных дисках -- -- -- -- -- 73
4.1. Дискеты: плотность и форматы -- -- -- -- -- -- -- 73
4.2. Накопители на гибких магнитных дисках-- -- -- -- -- -- 75
4.3. Контроллеры НГМД -- -- -- -- -- -- -- -- 77
4.3.1. БИС контроллера накопителя на гибком магнитном диске К580ВГ72 (i8272). -- 77
4.3.2. Регистры контроллера НГМД PS/2 и PC/AT. -- -- -- -- -- 81
4.3.3. Система команд контроллера -- -- -- -- -- -- -- 86
4.4. Рекомендации по программированию на уровне регистров -- -- -- 92
4.5. Пример программирования контроллера гибких дисков на уровне регистров -- 94
4.6. Прерывания BIOS для работы с дисками на низком уровне -- -- -- 108
4.7. Пример использования функций BIOS для работы с гибкими дисками -- -- 111
Контрольные вопросы -- -- -- -- -- -- -- -- 115
5. Память на жестких дисках -- -- -- -- -- -- -- 116
5.1. Накопители с интерфейсами ST-506 (ST-412) и ESDI -- -- -- -- 116
Контрольные вопросы -- -- -- -- -- -- -- -- 118
5.2. Накопители с интерфейсом ATA (IDE) -- -- -- -- -- -- 120
5.2.1. Интерфейс ATA -- -- -- -- -- -- -- -- 120
5.2.1.1. Электрический интерфейс -- -- -- -- -- -- -- 122
5.2.1.2. Регистры устройств -- -- -- -- -- -- -- -- 128
5.2.1.3. Адресация данных -- -- -- -- -- -- -- -- 131
5.2.1.4. Система команд -- -- -- -- -- -- -- -- 133
5.2.1.4.1. Основные команды -- -- -- -- -- -- -- 136
5.2.1.4.2. Инициализация, идентификация и конфигурирование устройств -- -- 137
5.2.1.4.3. Работа со сменными носителями -- -- -- -- -- -- 144
5.2.1.4.4. Команды для флэш-памяти-- -- -- -- -- -- -- 146
5.2.1.4.5. Управление энергопотреблением и шумом -- -- -- -- -- 146
5.2.1.4.6. Защита данных -- -- -- -- -- -- -- -- 147
5.2.1.4.7. Мониторинг состояния — SMART -- -- -- -- -- -- 149

5.2.1.4.8. Потоковое расширение команд ATA -- -- -- -- -- 150
5.2.1.5. Протоколы взаимодействия хоста и устройства -- -- -- -- 150
5.2.1.6. Протоколы и режимы передачи данных -- -- -- -- -- 151
5.2.1.7. Средства многозадачности (ATA-4) -- -- -- -- -- -- 154
5.2.1.8. Пакетный интерфейс ATAPI -- -- -- -- -- -- -- 154
5.2.1.9. Адаптеры и контроллеры шины ATA -- -- -- -- -- -- 159
5.2.1.10. Категории устройств IDE -- -- -- -- -- -- -- 160
5.2.1.11. Конфигурирование устройств -- -- -- -- -- -- 161
5.2.2. Примеры программ, непосредственно работающих с контроллером жесткого диска ATA 162
5.2.3. Риск потери информации, связанный с выполнением операций форматирования
и записи данных -- -- -- -- -- -- -- -- -- 166
Контрольные вопросы -- -- -- -- -- -- -- -- 167
5.3. Диски с интерфейсом SCSI -- -- -- -- -- -- -- 170
5.3.1. SCSI -- -- -- -- -- -- -- -- -- -- 171
5.3.1.1. Параллельный интерфейс SCSI -- -- -- -- -- -- 173
5.3.1.1.1. Фазы шины -- -- -- -- -- -- -- -- 181
5.3.1.1.2. Управление интерфейсом -- -- -- -- -- -- -- 184
5.3.1.2. Типы периферийных устройств -- -- -- -- -- -- 186
5.3.1.3. Адресация и система команд -- -- -- -- -- -- -- 189
5.3.1.4. Выполнение команд -- -- -- -- -- -- -- -- 193
5.3.1.5. Конфигурирование устройств SCSI -- -- -- -- -- -- 194
5.3.1.5.1. Хост-адаптер SCSI-- -- -- -- -- -- -- -- 196
5.3.1.5.2. Подключение устройств к шине -- -- -- -- -- -- 196
5.3.1.5.3. Терминаторы -- -- -- -- -- -- -- -- 197
Контрольные вопросы -- -- -- -- -- -- -- -- 200
6. Особенности работы с устройствами, подключенными к шине PCI-- -- 202
6.1.Конфигурационное пространство устройства PCI -- -- -- -- -- 202
6.2. Функции PCI BIOS -- -- -- -- -- -- -- -- -- 203
6.3. Поиск устройства PCI по коду класса -- -- -- -- -- -- 206
6.3.1. Пример программы поиска -- -- -- -- -- -- -- 209
6.4. Особенности реализации режима DMA на системных платах с шиной PCI -- -- 212
6.4.1. Пример программы -- -- -- -- -- -- -- -- 216
Контрольные вопросы -- -- -- -- -- -- -- -- 221
7. Устройства массовой памяти на сменных носителях-- -- -- -- 222
7.1. Магнитные диски -- -- -- -- -- -- -- -- 222
7.2. Магнитооптические диски -- -- -- -- -- -- -- 224
7.3. Оптические диски — CD, DVD, PD -- -- -- -- -- -- 227
7.3.1. Носители информации CD, CD-R, CD-RW -- -- -- -- -- 227
7.3.2. Устройство приводов CD-ROM, CD-R и CD-RW -- -- -- -- 229
7.3.3. Организация дисков CD -- -- -- -- -- -- -- 231
7.3.3.1 CD-DA -- -- -- -- -- -- -- -- -- 232
7.3.3.2. CD-ROM -- -- -- -- -- -- -- -- -- 234
7.3.3.3. CD-R и CD-RW -- -- -- -- -- -- -- -- 235
7.3.3.4. CD-I -- -- -- -- -- -- -- -- -- 237
7.3.3.5. CD-ROM XA -- -- -- -- -- -- -- -- 237
7.3.3.6. Многосеансовые диски -- -- -- -- -- -- -- 238
7.3.4. Файловые системы -- -- -- -- -- -- -- -- 239
7.3.5. Разновидности компакт-дисков -- -- -- -- -- -- 241
7.3.6. Запись на оптические диски -- -- -- -- -- -- -- 242
7.3.7. Системная поддержка CD-ROM -- -- -- -- -- -- 245
7.3.8. Загружаемые CD-ROM -- -- -- -- -- -- -- 246
7.3.9. Оптические диски с прямым доступом -- -- -- -- -- -- 250
7.3.10. Диски DVD -- -- -- -- -- -- -- -- -- 250
Контрольные вопросы -- -- -- -- -- -- -- -- 253
8. Установка, обслуживание и тестирование дисков -- -- -- -- 255
8.1. Интерфейс подключения устройств -- -- -- -- -- -- 255
8.2. Установка новых устройств -- -- -- -- -- -- -- 256
8.3. Проблемы использования больших дисков -- -- -- -- -- 258
8.3.1. Ограничения BIOS и ATA -- -- -- -- -- -- -- 259
8.3.2. Преодоление ограничений BIOS -- -- -- -- -- -- 261
8.3.3. «Трехмерная геометрия» дисков SCSI -- -- -- -- -- -- 273
8.3.4. Ограничения операционных систем -- -- -- -- -- -- 264
8.4. Конфигурирование и форматирование дисков -- -- -- -- -- 265

8.5. УТИЛИТЫ обслуживания дисков -- -- -- -- -- -- -- 268
8.5.1. Системные утилиты -- -- -- -- -- -- -- 268
8.5.2. УТИЛИТЫ от производителей дисков -- -- -- -- -- -- 269
8.5.3. УТИЛИТЫ «третьих фирм» -- -- -- -- -- -- -- 271
8.6. Основные причины отказов дисков -- -- -- -- -- -- 272
8.7. Повышение надежности и производительности дисков (RAID-массивы) -- -- 273
Контрольные вопросы -- -- -- -- -- -- -- -- 274
Рекомендуемая литература -- -- -- -- -- -- -- 276

Предисловие

Это пособие посвящено одной из важнейших подсистем персональных (и не только персональных) компьютеров, под которыми подразумеваются самые распространенные — IBM PC-совместимые. В нем в основном рассматриваются вопросы, которые не нашли своего отражения в пособие: "Павлов В.А. Учебные материалы по курсу «Периферийные устройства ЭВМ»: Часть 2. СарФТИ, Саров. 2001. - 271с.: ил.".

В главе 1 дается общая характеристика внешней памяти и ее место в архитектуре ПК. В главе 2 рассмотрены вопросы логической организации дисковых накопителей, а в главе 3 - общие вопросы их системной поддержки. Довольно широкий спектр вопросов, связанных с подсистемой гибких дисков, рассмотрен в главе 4. Подсистеме жестких дисков посвящена глава 5. Особое внимание в ней уделено рассмотрению интерфейсов ATA и SCSI. В главе 6 рассмотрены особенности работы с устройствами, подключенными к шине PCI и режиму Ultra DMA. Глава 7 посвящена массовой памяти на сменных носителях. При этом особое внимание уделяется накопителям на оптических дисках. В последней восьмой главе обсуждаются вопросы, связанные с установкой, обслуживанием и тестированием накопителей на жестких дисках.

В главах, посвященных гибким и жестким дискам, особое внимание уделяется вопросам программирования на уровнях регистров и с использованием BIOS. В них приведены примеры программ, на писанных на ассемблере и реализующих взаимодействие с дисками через регистры контроллеров и через функции BIOS.

Данное пособие является информационной поддержкой раздела курса лекций "Адаптеры и контроллеры ЭВМ", читаемого автором в Саровском физико-техническом институте (СарФТИ) в рамках специальности "Вычислительные машины, комплексы, системы и сети" и других родственных специальностей. Оно также может быть использовано для информационной поддержки и других курсов лекций, например, "Системное программирование" и "Периферийные устройства ЭВМ".

1. Внешняя память компьютера

К внешней памяти относятся устройства, позволяющие сохранять информацию для последующего ее использования независимо от состояния (включен или выключен) компьютера. Устройства хранения данных могут использовать различные физические принципы хранения информации — магнитный, оптический, электронный в любых их сочетаниях. Внешняя память принципиально отличается от внутренней (оперативной, постоянной и специальной) памяти способом доступа процессора (исполняемой программы) к ее содержимому. Процессор способен одной командой обратиться (считать, записать) к любому биту, байту, слову, двойному слову и более крупным структурам данных, расположенным во *внутренней* памяти. Но что еще важнее — он выбирает инструкции (команды) для исполнения именно из внутренней (оперативной или постоянной) памяти. Характерной особенностью *внешней* памяти является то, что ее устройства оперируют *блоками* информации, но никак не байтами или словами, как это позволяет оперативная память. Эти блоки обычно имеют фиксированный размер, кратный степени числа 2. Блок может быть переписан из внутренней памяти во внешнюю или обратно только целиком, и для выполнения любой операции обмена с внешней памятью требуется специальная процедура (подпрограмма). Процедуры обмена с устройствами внешней памяти привязаны к типу устройства, его контроллеру и способу подключения устройства к системе (интерфейсу).

По *методу доступа* к информации устройства внешней памяти разделяются на устройства с прямым (или непосредственным) и последовательным доступом. *Прямой доступ* (direct access) подразумевает возможность обращения к блокам по их адресам в произвольном порядке. Традиционными устройствами с прямым доступом являются дисковые накопители, и часто в понятие «диск» или «дисковое устройство» (disk device) вкладывают значение «устройство внешней памяти прямого доступа». Так, например, виртуальный диск в ОЗУ и электронный диск на флэш-памяти отнюдь не имеют круглых, а тем более вращающихся деталей. Традиционными устройствами с *последовательным доступом* являются накопители на магнитной ленте (tape device), они же стриммеры. Здесь каждый блок информации тоже может иметь свой адрес, но для обращения к нему устройство хранения должно сначала найти некоторый маркер начала ленты (тома), после чего последовательным холостым чтением блока за блоком дойти до требуемого места и только тогда производить сами операции обмена данными. Конечно, каждый раз возвращаться на начало ленты необязательно, однако необходимость последовательного сканирования блоков (вперед или назад) — неотъемлемое свойство устройств последовательного доступа. Несмотря на очевидный проигрыш во времени доступа к требуемым данным, ленточные устройства последовательного доступа находят применение для хранения очень больших массивов информации. Устройства прямого доступа — диски самой различной природы — являются обязательной принадлежностью подавляющего большинства компьютеров.

1.1. Основные характеристики устройств внешней памяти

Главная характеристика устройств — *емкость* хранения (capacity), измеряемая в килобайтах, мегабайтах, гигабайтах и терабайтах (Кбайт, Мбайт, Гбайт, Тбайт или KB, MB, GB, TB, или, еще короче — K, M, G, T). Здесь, как правило, приставки кило-, мега-, гига-, тера- имеют *десятичные* значения — 10^3, 10^6, 10^9 и 10^{12} соответственно. В других подсистемах компьютера, например при определении объема ОЗУ, ПЗУ и другой внутренней памяти, эти же приставки чаще применяют в *двоичных* значениях 2^{10}, 2^{20}, 2^{30} и 2^{40} соответственно, при этом 1 Кбайт = 1024 байт, 1 Мбайт = 1024 Кбайт, 1 Гбайт -1024 Мбайт, 1 Тбайт - 1024 Гбайт. Этими разночтениями объясняются различия значений емкости одного и того же устройства, полученных из разных источников. «Двоичные» кило-, мега-, гига-, тера более «увесисты», поэтому емкость устройства, выраженная в десятичных единицах, будет выглядеть внушительнее. Так, например, один и тот же былой порог «беспроблемного» объема жесткого диска составлял 528 Мбайт (десятичных) или 504 Мбайт (двоичных).

Устройства внешней памяти могут иметь *сменные* или *фиксированные носители* информации. Применение сменных носителей (removable media) позволяет хранить неограниченный объем информации, а если носитель и формат записи стандартизованы, то они позволяют и обмениваться информацией между компьютерами. Существуют устройства с автоматической сменой носителя — ленточные карусели, дисковые устройства JukeBox. Эти достаточно дорогие устройства применяют в мощных файл-серверах. Для настольных машин имеются накопители CD-ROM с несколькими дисками (CD-changer), сменяемыми автоматически.

Важнейшими общими параметрами устройств являются время доступа, скорость передачи данных и удельная стоимость хранения информации.

Время доступа (access time) определяется как усредненный интервал от выдачи запроса на передачу блока данных до фактического начала передачи. Дисковые устройства имеют время доступа от единиц до сотен миллисекунд. Для электронных устройств внешней памяти время доступа определяется быстродействием используемых микросхем памяти и при чтении составляет доли микросекунд, причем запись может ожидаться значительно дольше, что объясняется природой энергонезависимой электронной памяти. Для устройств с подвижными носителями основной расход времени происходит в

процессе *позиционирования* головок (seek time — время поиска) и *ожидания* подхода к ним требуемого участка носителей (latency— скрытый период). Для дисковых и ленточных устройств принципы позиционирования различны, и различные составляющие процесса поиска будут подробнее рассмотрены в описании соответствующих устройств.

Скорость записи и считывания определяется как отношение объема записываемых или считываемых данных ко времени, затрачиваемому на эту операцию. В затраты времени входит и время доступа, и время, затрачиваемое на передачу данных. При этом оговаривается характер запросов — линейный или случайный, что сильно сказывается на величине скорости из-за влияния времени доступа. При определении скорости линейного чтения/записи (linear transfer rate read/write) производится обращение к длинной цепочке блоков с последовательным нарастанием адреса. При определении скорости случайных запросов чтения/записи (random transfer rate read/write) соседние запросы разбросаны по всему носителю. Для современных многозадачных ОС характерно чередующееся выполнение нескольких потоков запросов, и в каждом потоке высока вероятность последовательного нарастания адреса.

Скорость передачи данных (Transfer Speed, Transfer Rate или сокращенно XFER) определяется как производительность обмена данными, измеряемая после выполнения поиска данных. Однако в способе измерения этого параметра возможны разночтения, поскольку современные устройства имеют в своем составе буферную память существенных размеров. Скорости обмена буферной памяти с собственно носителем (внутренняя скорость) и с внешним интерфейсом могут существенно различаться. Если *скорость* работы *внешнего интерфейса* ограничивается быстродействием электронных схем и достижимой частотой передаваемых сигналов, то *внутренняя скорость* более жестко ограничивается возможностями электромеханических устройств (скоростью движения носителя и плотностью записи). При измерениях скорости передачи на небольших объемах пересылок проявится ограничение внешнего интерфейса буферной памяти, при средних объемах — ограничение внутренней скорости, а при больших объемах проявится еще и время поиска последующих блоков информации. Бывает, что в качестве скорости передачи данных указывают лишь максимальную скорость интерфейса, а о внутренней скорости можно судить по частоте вращения дисковых носителей и числу секторов на треке, но об этих понятиях будет сказано чуть позже.

Определение *удельной стоимости хранения* информации для накопителей с фиксированными носителями пояснения не требует. В случае сменных носителей этот показатель интересен для собственно носителей, но не следует забывать и о цене самих накопителей, которую тоже можно приводить к их емкости.

В табл. 1.1 приведены основные параметры распространенных устройств внешней памяти. Поскольку эти компоненты развиваются весьма динамично, данные таблицы, фиксирующие состояние на осень-зиму 2000 года, не стоит рассматривать с коммерческой точки зрения. Они позволяют сопоставить возможности различных решений задачи хранения и переноса данных. А о динамике развития можно сказать, что за последние годы, например, «модный» объем винчестера ежегодно удваивался даже при некотором снижении цены, так что удельная стоимость хранения снижалась более чем вдвое. Поскольку емкость и скорость, постоянно растут, в таблице; эти возможности роста обозначены символом. "+".

Таблица 1.1. Характеристики устройств внешней памяти

Тип и размер диска	Емкость диска, байт	Время доступа, мс	Скорость, Мбайт/с	Цена ($) устройства/ 1 Мбайт носителя
FDD 5" 360K	360K	100*	0,027	-
FDD 5" 1,2M	1,2M	100*	0,045	-
FDD 3,5"	1.44M	100*	0,055	20/0,3
HDD IDE	10-30Г+	7,5-10	2-20+	100/(20Гбайт)
HDD SCSI	10-30Г+	7,5-10	2-40+	120/(20Гбайт)
CD-ROM 1x	650M	240-500	0,15	не выпускаются
CD-ROM 48x	650M	75	7,2 (макс.)	50/менее0,01
CD-RW 4/4/32 (IDE)	650M	150	Чтение: 4,8 (маке.) Запись:0,6	120/0,17(CD-RW), 0,1(CD-R)
CD-RW 8/8/32 (SCSI)	650M	150	Чтение: 4,8 (макс.) Запись:1,2	200/0,17(CD-RW), 0,1(CD-R)
DVD-ROM12X	4,7-17,081"	200	130(макс.)	100
MOD 3,5"	230M	50	Чтение: 1,2-2 Запись: 0,6-2	/0,035
MOD 3,5"	540/640M	28	Чтение: до 3,67 Запись: до 1,2	230/0,008

MOD 3,5"	1,3Г	28	Чтение: до 4,5 Запись: до 1,5	350/0,017
LS-120	120М	70	0,1-0,5	130/0,1
IomegaZip 100	100М	29	1,4	50-100/0,1
IomegaZip250	250М	29	2,4	100-170/0,07
IomegaJaz	1Г	16	7,5	200/0,1
IomegaJaz	2Г	16	7,5	300/0,063
SyQuestEZ135	135М	21	1,4	
SyQuest EZFIyer	230М	13,5	до 2,4	100/0,22
SyQuest SparQ	1Г	12	3,7-6,9	230-250/0,06
SyQuest SyJet	1,5Г	12	3,7-6,9	300/0,1

* Время доступа без учета разгона диска.

1.2. Конструктивы и питание устройств

Все устройства внешней памяти, применяемые в PC, *имеют унифицированные конструктивные исполнения* (конечно, за исключением крупногабаритных устройств с автоматической сменой носителя или бобинных накопителей на магнитной ленте). Их типоразмеры стандартизованы: наиболее жестю задается ширина и высота устройств (поскольку их лицевые панели могут «выглядывать» из лицевой панели компьютера), глубина ограничена только максимально допустимым значением. Единообразное расположение резьбовых крепежных отверстий позволяет унифицировать конструкцию *отсеков* (bay) корпусов PC, предназначенных для установки накопителей. Первые накопители на гибких магнитных дисках диаметром 5,25", применяемые в PC, имели размеры лицевой панели 146,1×82,6 мм (5,75"×3,25") и глубину около 203 мм (8"). Этот формат называется полновысотным пятидюймовым форматом (доли дюйма для краткости опускают) — 5" Full-Height Form-Factor. Такого же размера были и первые винчестеры. Вскоре высоту накопителя удалось уменьшить вдвое — до 41,4 мм, и этот формат (5" Half-Height — половинная высота) до сих пор используется как стандартный для многих типов устройств: НГМД с 5"-дискетами, приводы CD и DVD, стримеры, магнитооптика, накопители на жестких дисках (старые и новые значительной емкости) и др. НГМД с дискетами диаметром 3,5" имеют ширину и высоту 101,6×25,4 мм (4"x1") и длину (глубину) около 146 мм (5,74"). Этот формат широко используется и для современных трехдюймовых винчестеров, хотя среди них встречаются и «тонкие» модели высотой 20 мм (0,75"), и «толстые» — высотой 41 мм (1,6"). Для портативных компьютеров используют формат 2,75" с размерами 70,1×12,7×100,5 мм (2,76"×0,5"×3,95"), есть и накопители формата 1,8".

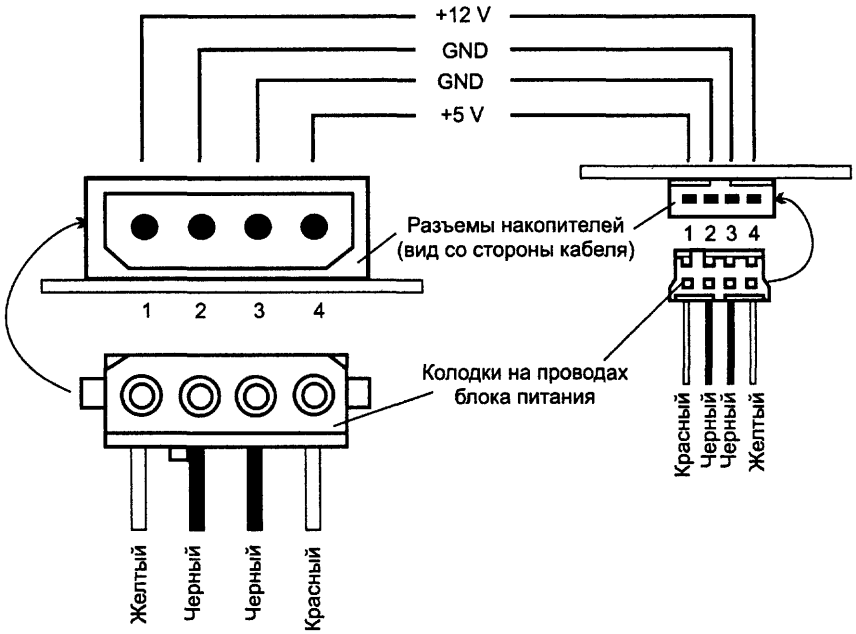

Рис. 1.1. Разъемы питания накопителей

Стандартизованы также *и разъемы подключения питания* (рис. 1.1). Миниатюрные разъемы используются только для питания трехдюймовых НГМД, практически все остальные устройства трех- и пятидюймовых форматов используют большие разъемы. Напряжение +5 В используется для питания электронных схем, напряжение +12 В — для питания двигателей, хотя в некоторых накопителях при-

воды могут питаться и от шины +5 В.

ВНИМАНИЕ

Редко, но попадаются блоки питания с ошибочно собранными колодками — у них шины +5 В и +12 В перепутаны, что легко заметить по цветовой маркировке проводов. Подключение накопителя к такой колодке чревато выгоранием его электроники — она рассчитана на питание 5В,ане12В.

В момент запуска двигателей ток потребления по цепи +12 В может превышать установившееся значение в несколько раз. Если мощности блока питания окажется недостаточно и напряжение «просядет», накопитель может и остановить двигатель, а после возврата питания к норме снова попытаться его запустить. Хорошо заметный на слух процесс многократных безуспешных попыток «завести» мотор указывает на необходимость замены блока питания на более мощный. В компьютерах с большим числом накопителей моменты их запуска стараются разнести по времени, что позволяет снизить пиковую нагрузку на источник питания.

По отношению к корпусу компьютера устройства могут быть внутренними (internal) и внешними (external). *Внутренние устройства* помещаются в специальные трех- или пятидюймовые отсеки корпуса компьютера и питаются от его же блока питания. В описании корпусов компьютеров отсеки также подразделяются на внешние и внутренние, но они различаются лишь тем, может ли передняя панель устройства, установленного в отсек, выходить на лицевую панель корпуса или нет. *Внешние устройства* помещают в отдельный корпус, а питаются они от собственного блока питания или перехватывают питание +5 В от разъема клавиатуры компьютера. Внешнее исполнение имеют как малогабаритные портативные устройства, так и особо крупные дисковые массивы. Сами приводы для внешних и внутренних устройств обычно имеют одинаковый конструктив одного из распространенных форматов.

1.3. Компоненты накопителей на магнитных дисках

Рассказав в общих чертах о работе дисковых накопителей, остановимся подробнее на отдельных деталях накопителей на магнитных дисках.

Пластины (Platter) дисков могут быть гибкими или жесткими. В любом случае их материал не должен сильно изменять свой размер со временем и под действием перепадов температур. Для пластин гибких дисков используют майлар или лавсан, для жестких — обычно алюминий. На поверхность пластины наносят *рабочий магнитный слой,* который чаще всего основан на окиси железа. Хранимую информацию представляет состояние намагниченности отдельных участков рабочей поверхности. От качества материала рабочего слоя (коэрцитивной силы, размера зерна) зависит допустимая плотность записи информации. Более высокой плотности хранения информации позволяет добиться применение металлического носителя — Plated Media. Механическая прочность и устойчивость рабочего слоя определяет долговечность носителя.

Традиционно для записи и считывания информации используются *магнитные головки,* представляющие собой миниатюрные катушки индуктивности, намотанные на магнитном сердечнике с зазором. При записи головка намагничивает участок магнитного слоя, проходящий под рабочим зазором, в направлении, определяемом направлением протекающего тока. При проходе намагниченных участков поверхности около индуктивной головки считывания в момент смены направления намагниченности в ней наводятся импульсы э.д.с., полярность которых определяется знаком смены направления магнитного поля. Таким образом, сигнал считывания в индуктивной головке по форме представляет собой производную от записанного сигнала. Это свойство обязательно учитывается при выборе способа модуляции, о которой речь пойдет ниже. Требования к оптимальной конструкции головок для процессов записи и считывания различаются, так что универсальная головка представляет собой некоторый компромисс. Первые индуктивные головки содержали проволочные обмотки, их сменили головки, выполненные по *тонкопленочной технологии TF* (Think Film — тонкая пленка). В современных накопителях для считывания все чаще применяют *магниторезистивные головки,* основанные на эффекте анизотропии сопротивления полупроводников в магнитном поле (AMR — anisotropic magnetoresistance). В них через магниторезистивный датчик пропускают измерительный ток, и величина падения напряжения на нем пропорциональна намагниченности находящегося под головкой участка магнитной поверхности. В отличие от индуктивной магниторезистивная головка вырабатывает сигнал не по производной записанного сигнала, а повторяет его форму. Магниторезистивная головка считывания хорошо уживается с индуктивной головкой записи, что позволяет достигать высокой плотности записи информации на магнитный носитель. Однако по технологии изготовления она сложнее тонкопленочной индуктивной, поскольку в ней сочетаются разнородные компоненты.

Скорость вращения дисков у разных накопителей различна. У накопителей на гибких дисках она составляет 300 или 360 об/мин, у распространенных накопителях на жестких дисках — 3600 об/мин. Чем выше скорость вращения, тем больше скорость обмена информацией с диском. У наиболее быстрых дисков скорость достигает 5400, 7200 и даже 10000 об/мин. Однако высокие скорости вращения порождают проблемы, связанные с балансировкой, гироскопическим эффектом и аэродинамикой головок. Из-за

гироскопического эффекта не рекомендуется перемещение (точнее, смена ориентации оси шпинделя) включенных накопителей с вращающимся шпинделем. Накопители для портативных компьютеров разрабатываются с учетом этих эффектов.

Для магнитных головок весьма критично расстояние от головки до поверхности магнитного слоя носителя. В накопителях на гибких дисках в нерабочем положении головка поднята над поверхностью диска на несколько миллиметров, а в рабочем прижимается к поверхности диска. Однако непосредственный контакт головки с поверхностью допустим лишь при малых скоростях движения носителя. В накопителях с высокой скоростью вращения головки поддерживаются на микроскопическом расстоянии от рабочей поверхности аэродинамической подъемной силой. «Падение» головки на рабочую поверхность, которое произойдет, если диск остановится, может повредить как головку, так и поверхность диска. Чтобы этого не происходило, в нерабочем положении головки *паркуются* (Park) — отводятся в нерабочую зону, где допустимо их «приземление». Старые винчестеры требовали выполнения операции парковки, инициируемой программным обеспечением. Для ее выполнения в параметрах жестких дисков, хранимых в CMOS, присутствовал номер цилиндра для парковки (Landing Zone или LZone). Парковка выполнялась запуском утилиты PARK или других утилит. Однако, если после выполнения парковки появлялось обращение к жесткому диску, головки его, естественно, выходили из «зоны приземления», и парковка теряла смысл. В современных накопителях парковка осуществляется автоматически при снижении напряжения питания или же по снижении скорости вращения шпинделя ниже допустимого значения. Для таких накопителей указанное в BIOS Setup значение параметра LZone игнорируется. Контроллеры современных дисков к тому же не выпустят головок из зоны парковки, пока шпиндель не наберет заданных оборотов. Высота «полета» головки должна выдерживаться довольно строго, иначе магнитные поля головок будут «промахиваться» мимо рабочего слоя. Высота определяется тем положением, когда подъемная сила, определяемая скоростью вращения, формой «крыла» головки и плотностью воздуха, уравновесит давление прижимающей головку пружины. Плотность воздуха меняется в зависимости от атмосферного давления, и по этой причине в параметрах накопителей указывают максимальную высоту над уровнем моря, на которой может работать накопитель (или минимальное атмосферное давление). Конечно же, должна быть и верхняя граница допустимого давления (для подводных аппаратов), но этот параметр для устройств широкого применения не указывают, поскольку с такими условиями сталкиваются редко.

В качестве *привода для позиционирования* головок на нужный цилиндр в накопителях на гибких дисках и старых винчестерах применяют *шаговые двигатели.* Эти двигатели под действием серии импульсов, подаваемых на их обмотки, способны поворачивать свой вал на определенный угол. Этот угол может быть только кратен минимальному шагу, определяемому конструкцией двигателя. Вращательное движение вала шагового двигателя преобразуется в поступательное с помощью червячного механизма или металлической ленты, намотанной на вал. Таким образом, поворот вала двигателя на один шаг приводит к перемещению блока головок на один цилиндр. Червячная передача должна быть высококачественной — люфты в ней будут приводить к погрешности позиционирования. Ленточная передача в принципе свободна от люфтов, она обеспечивает более высокую точность и быстродействие позиционирования. Однако под действием изменения температуры и со временем из-за износа положение треков на носителе, задаваемое только шаговым двигателем, будет меняться, и ранее записанная информация может перестать считываться. Требования к точности позиционирования растут с повышением плотности хранения информации (количества цилиндров на диске), а быстродействие позиционирования определяет время доступа, которое, естественно, стремятся уменьшить. С точки зрения теории автоматического управления привод с шаговым двигателем является *разомкнутой системой* (то есть системами без обратной связи). Такая система не позволяет корректировать ошибки позиционирования, вызванные, например, температурным изменением размеров дисков. Конечно, при всех операциях обмена проверяется адресный маркер цилиндра и в случае его несовпадения делается повторная попытка позиционирования — возврат к нулевому цилиндру и подача требуемого количества шаговых импульсов. Выход на нулевую дорожку определяется по *датчику нулевого цилиндра,* в качестве которого обычно применяется оптоэлектронная пара с флажком, связанным с блоком головок. Точное положение датчика нулевого цилиндра для несменных дисков не так уж существенно — главное, чтобы оно не менялось после процедуры низкоуровневого форматирования, о которой речь пойдет ниже. Для накопителей со сменными носителями положение нулевого цилиндра существенно — для обеспечения совместимости накопителей у всех устройств его положение должно совпадать. Однако датчик задает положение нулевого цилиндра лишь грубо — он определяет только номер шага привода, на котором головки находятся напротив нулевого цилиндра. Более точно положение можно отрегулировать вращением корпуса двигателя в пределах нескольких градусов (не больше, чем угловой шаг двигателя). В накопителях с шаговым приводом довольно сложно осуществить автоматическую парковку головок, поскольку для этого необходимо сформировать серию импульсов управления шаговым двигателем. В случае внезапного отключения питания для автопарковки должна быть запасена достаточная энергия. Иногда в качестве хранилища энергии используется сам пакет дисков: при отключении питания шпиндельный двигатель начинает работать генератором, обеспечивая питание схемы управления и

шагового двигателя на время автопарковки.

В современных накопителях применяют привод головок с *подвижной катушкой* (Voice Coil Actuator), работающий по принципу звуковой катушки динамика. В таком приводе блок головок связан с катушкой индуктивности, помещенной в магнитное поле постоянного магнита. При протекании тока через катушку на нее начинает действовать сила, пропорциональная силе тока, которая вызовет перемещение катушки, а следовательно, и блока головок. Привод может быть линейным или поворотным. В накопителе с линейным приводом катушка с блоком головок перемещается по радиусу дисков. Такой привод применялся в накопителях больших машин. В накопителе с поворотным приводом блок головок с катушкой размещены на поворотной рамке (рис. 1.2), и траектория головок отличается от радиальной. При этом азимут головки относительно трека меняется при перемещении головки, и эта азимутальная погрешность нежелательна для работы головок записи и считывания. Тем не менее с этой неприятностью, отсутствующей в случае линейного привода, мирятся из-за относительной простоты исполнения, меньших габаритов, а следовательно, и меньшей инерционности поворотного привода. В большинстве современных накопителей на жестких дисках применяется поворотный привод. Управляя направлением и силой тока, можно быстро перевести блок головок в любое положение — произвольное, а не по фиксированным шагам. Но в такой системе позиционирования необходима обратная связь — информация о текущем положении головок, по которой контроллер может управлять приводом. Привод, обеспечивающий точное позиционирование по сигналу обратной связи, называется *сервоприводом.* Управление сервоприводом может быть оптимизировано по времени установления головок на требуемую позицию: когда отклонение от заданного положения велико, можно подавать больший ток, вызывающий большое ускорение блока. По мере приближения ток уменьшается, а для компенсации инерции в конце позиционирования ток может и поменять направление (активное торможение). Такая система привода позволяет сократить время доступа до единиц миллисекунд против сотен миллисекунд, характерных для шагового привода. Остается только решить вопрос об источнике сигнала обратной связи для сервопривода, который, с точки зрения теории автоматического управления, является *замкнутой системой.*

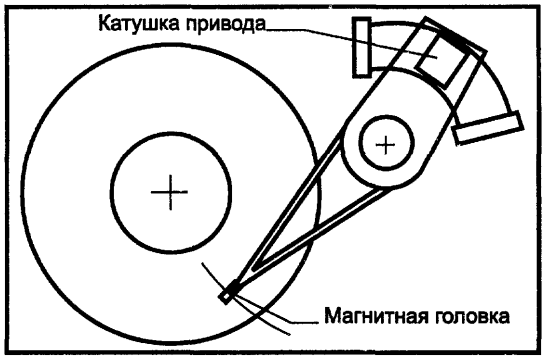

Рис. 1.2. Поворотный привод головки

В первых накопителях с линейным приводом использовалась специальная зубчатая рейка и магнитный датчик, по сигналу которого отсчитывался номер трека. Однако по отношению к диску такая система привода все равно оставалась разомкнутой — привод позиционировал головки по своим соображениям о координатах. Замкнуть систему управления позволило размещение сервометок — вспомогательной информации для «системы наведения» — прямо на диске. В таком случае изменения размеров диска и привода под действием температуры и других факторов перестает существенно влиять на точность позиционирования, поскольку сервометки располагаются на тех же искомых треках. Сервометки записываются при сборке накопителя, когда для позиционирования используется внешний «прицел» специального технологического оборудования. В процессе эксплуатации сервометки только считываются, а серводорожки построены таким образом, что по считываемым сигналам обеспечивается быстрая и точная идентификация положения головки для поиска и слежение за положением головок относительно найденного трека. По месту размещения сервометок различают накопители с *выделенной сервоповерхностью* (Dedicated Servo) и со встроенными сервометками (Embeded Servo). В первом случае в пакете дисков выделяется одна поверхность, используемая исключительно для хранения сервометок, и соответствующая ей головка является сервоголовкой. Ошибка позиционирования в такой системе может возникать вследствие изменения взаимного положения (перекоса) головок в блоке. Сервоголовка для следящей системы дает информацию практически непрерывно, что улучшает качество процесса поиска и слежения. Верным признаком выделенной сервоповерхности является нечетное количество головок (рабочих), указанное для данного накопителя.

В накопителях со *встроенными сервометками* информация для сервопривода записывается на рабочих поверхностях между секторами с данными. Она может размещаться в начале каждого трека —

при этом на диске появляется клиновидная область сервометок. Недостатком такого размещения является то, что сервоинформация (сигнал обратной связи) доступна дискретно с периодичностью в один оборот диска, который для скорости 3600 об/мин занимает 16,6 мс. До точного позиционирования приходится выжидать нескольких оборотов диска. Более быстродействующий вариант — размещение сервометок перед каждым сектором, что позволяет выйти на заданный трек даже за доли оборота шпинделя. Преимущество встроенных сервометок в том, что он позволяет компенсировать любые изменения геометрии, поскольку система наводит головки именно по тому треку, к которому выполняется доступ. Существуют накопители и с гибридной сервосистемой, где кроме выделенной сервоповерхности используются и сервометки, размещенные на рабочих поверхностях.

Для накопителей с подвижной катушкой проблема автопарковки решается легко: блок головок при обесточивании сервопривода может быть легко установлен в зону парковки с помощью пружины. В старых накопителях с линейным приводом для возврата головок в исходное состояние использовали энергию, запасенную в конденсаторе.

Вся электромеханическая часть накопителя — пакет дисков со шпиндельным двигателем и блок головок с приводом, сокращенно называемая *HDA* (Head Disk Assembly), заключается в защитный кожух. Совершенно очевидно, что в корпусах накопителей воздух должен быть чистым — мелкая частичка, попавшая под головку, под которой пролетает носитель со скоростью в несколько десятков километров в час, может поцарапать и головку и диск. Обычно кожух не герметичен — в нем имеется отверстие, закрытое фильтром, обеспечивающее выравнивание давления внутри сборки с атмосферным. Кроме этого фильтра, называемого барометрическим, имеется еще и внутренний рециркуляционный фильтр. Этот фильтр устанавливается на пути потока воздуха, увлекаемого вращающимся пакетом дисков. Он улавливает частички, которые могут выбиваться из поверхности дисков при «взлете» и «посадке» головок. Разборку блока HDA можно производить только в специальном помещении или боксе с воздухом, очищенным от пыли (если предполагается, что после переборки накопитель должен будет снова работать). Особенно это касается накопителей с большой емкостью (переборка старинного 20-мегабайтного ST-225 в обычной комнате автору «сошла с рук»). Кроме пыли, в корпусе накопителя могут образоваться капельки конденсата влаги. Если накопитель находился на морозе, то в теплом помещении перед включением он должен быть выдержан несколько часов, пока конденсат не испарится с отогревшихся частей. Если запустить не отогревшийся накопитель, последствия для рабочих поверхностей могут быть фатальными, особенно если головки примерзнут к пластинам. Существуют, конечно, и специальные накопители для работы в особых климатических условиях. Они могут иметь герметический корпус, который выдерживает разность внутреннего и наружного давления. В PC, как правило, применяют накопители обычного исполнения.

Кроме механики, дисковый накопитель должен иметь и *блок электроники,* управляющий приводами шпинделя и головок, а также обслуживающий сигналы рабочих головок записи-считывания. *Контроллером накопителя* называют электронное устройство, на одной (интерфейсной) стороне которого обмен идет байтами команд, состояния и, конечно же, записываемой и считываемой информации, а другая его сторона связывается с блоком HDA. В современных накопителях на жестких дисках с интерфейсом ATA и SCSI контроллер расположен на плате электроники, смонтированной вместе с блоком HDA. В накопителях на гибких дисках на плате электроники накопителя смонтированы только схемы управления двигателями, усилители-формирователи сигналов записи и считывания, датчика индекса и некоторых других. Контроллер гибких дисков вынесен на специальную плату адаптера или размещается на системной плате. Интерфейс между НГМД и его контроллером чисто логический с уровнями ТТЛ. В накопителях на жестких дисках с интерфейсами ST506/412 и ESDI контроллер также внешний. Объединение контроллера с блоком HDA позволяет решить многие проблемы оптимизации накопителей, о которых речь пойдет ниже.

1.4. Принципы хранения информации

Энергонезависимое хранение информации может осуществляться на различных физических принципах. Раньше всех начали применять *магнитный способ хранения,* где запись «0» или «1» изменяет направление намагниченности элементарной хранящей ячейки. Устройства хранения на магнитных сердечниках состояли из матрицы ферритовых колец (по кольцу на каждый хранящийся бит), пронизанных обмотками (адреса, записи и считывания). Считывание выполнялось импульсом тока, пытающимся намагнитить ячейку в определенном направлении. Если ячейка была в противоположном состоянии, то эта попытка наводила импульс в обмотке считывания. Устройства ферритовой памяти были громоздкими, но сугубо статическими — в них не было движущихся частей (кроме поворотных рам блоков, которые нередко приходилось вскрывать для ремонтных работ). В устройствах с подвижным носителем хранящие ячейки движутся относительно головок записи/считывания, и, в зависимости от направления намагниченности, они вызывают в головке считывания импульс определенной полярности. На таком принципе строились и магнитные барабаны первых ЭВМ, и магнитные диски, и накопители на магнитной ленте. В оптических устройствах хранения используют изменение отражающей или пропускающей способности участков носителей. Первые *оптические устройства* использовали в

качестве носителя фотопленку (может, кто и помнит), перфоленту (до сих пор используется в старых станках с ЧПУ), перфокарты. Теперь оптические устройства хранят информацию на дисках с ячейками микроскопических размеров, считываемых лазерным лучом. Подробнее оптические, а также магнитооптические устройства рассмотрены в главе 6. В 2000 году появилось сообщение о новом типе оптических дисков FMD (Fluorescent Multilayer Disk, флюоресцентный многослойный диск), разработанном компанией Constellation 3D Inc.(C3D). В этих дисках информацию несут частички флюоресцирующего вещества, вкрапленные в слои прозрачного пластика. В отличие от CD/DVD, где информативна степень отражения лазерного луча от текущей точки поверхности, здесь воспринимается флюоресцентное свечение, вызванное этим лучом. Оптическая система привода позволяет фокусироваться лишь на требуемом слое. Поскольку слои прозрачны, их число может быть значительно увеличено без ощутимых потерь сигнала. Для начала предлагается 12-слойный диск емкостью 50 Гбайт, со скоростью считывания до 1 Гбит/с. Пока что разработана технология печати дисков с матриц (ROM), но уже прорабатывается и технология однократно записываемых дисков. Первые сообщения о проекте «трехмерных дисков» появились еще в 1997 году, но теперь уже они вот-вот выйдут на рынок.

Из *электронных устройств* распространение получила флэш-память, сочетающая довольно высокую плотность хранения с теперь уже приемлемой ценой. Флэш-память является статической и имеет очень высокое быстродействие считывания, правда, не очень быструю процедуру записи. Для флэш-памяти характерно то, что для перезаписи должен предварительно стираться целый блок ячеек: современные микросхемы состоят из набора блоков. В режиме хранения на флэш-память питание можно не подавать — энергопотребление нулевое. В режиме чтения потребление достаточно малое, но стирание и запись, естественно, требуют энергозатрат.

Устройства хранения на флэш-памяти выпускаются в разнообразных конструктивных исполнениях. Первые «статические диски» выполнялись в виде устройства 3,5" формата с интерфейсом ATA. Затем появились флэш-карты с интерфейсом PC Card (PCMCIA), Card Bus, которые используются в блокнотных ПК, а также в ряде бытовых электронных устройств, например в цифровых фотокамерах.

Для настольных ПК, не оборудованных слотом PC Card, компания SmartDisk разработала оригинальное устройство FlashPath™ (выпускается фирмами FujiFilm и Toshiba). Оно выполнено в виде дискеты (не дисковода!) 3,5" формата и имеет слот, в который вставляется флэш-карта SSFDC (Solid State Floppy Disk Card), она же SmartMedia, производства фирм Toshiba и Samsung. В устройстве имеется электронная схема, передающая информацию с карты в компьютер (и обратно) через магнитные головки дисковода. Питается эта «дискета» от батареек-«таблеток». Устройство совместимо со стандартными дисководами на 1,44 Мбайт (не 2,88 и не LS120), но для работы, естественно, требует специальный драйвер. Скорость обмена выше, чем у COM-порта: 4 Мбайт передаются примерно за 2 минуты. Устройство поддерживает карты SmartMedia объемом 2, 4, 8 и 16 Мбайт с питанием 3,3 или 5 В. Однако карты Compact Flash, применяемые в ряде моделей цифровых фотокамер, вставить в FlashPath нельзя — они толще дискеты.

1.5. Хранение информации на магнитных дисках

Дисковые накопители имеют своей основой механизм, схематически представленный на рис. 1.3.

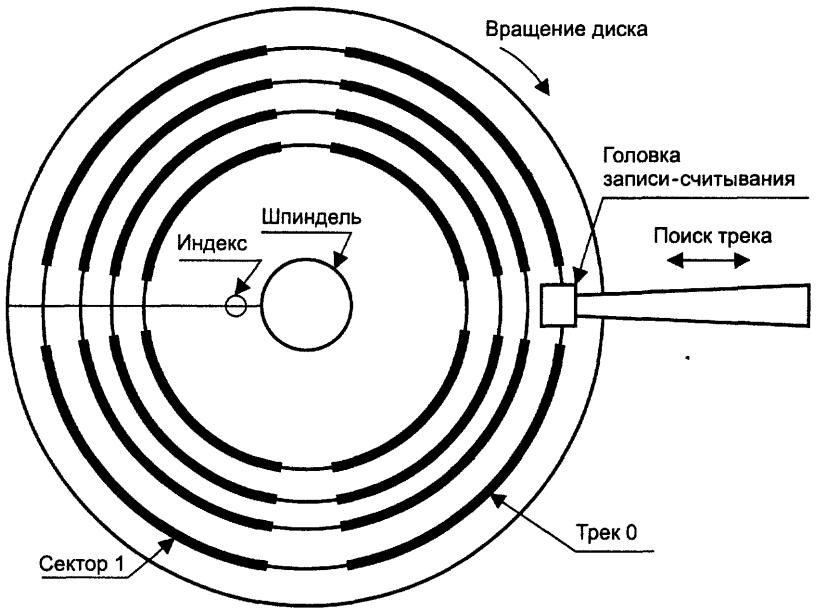

Рис 1.3.Устройство дискового накопителя

Носителем информации является диск (один или несколько), на который нанесен слой вещества, способного намагничиваться (чаще всего ферромагнитный). Хранимую информацию представляет состояние намагниченности отдельных участков рабочей поверхности. Диски вращаются с помощью *двигателя шпинделя* (spindle motor), обеспечивающего требуемую частоту вращения в рабочем режиме. На диске имеется *индексный маркер,* который, проходя мимо специального датчика, отмечает начало каждого оборота диска. Информация на диске располагается на концентрических *треках* (дорожках), нумерация которых начинается с внешнего трека (track 00). Каждый трек разбит на *секторы* (sector) фиксированного размера. Сектор и является минимальным блоком информации, который может быть записан на диск или считан с него. Нумерация секторов начинается с единицы и привязывается к индексному маркеру. Каждый сектор имеет служебную область, содержащую адресную информацию, контрольные коды и некоторую другую информацию, и область данных, размер которой традиционно составляет 512 байт. Если накопитель имеет несколько рабочих поверхностей (на шпинделе может быть размещен пакет дисков, а у каждого диска могут использоваться обе поверхности), то совокупность всех треков с одинаковыми номерами составляет *цилиндр* (cylinder). Для каждой рабочей поверхности в накопителе имеется своя *головка* (head), обеспечивающая запись и считывание информации. Головки нумеруются, начиная с нуля. Для того чтобы произвести элементарную операцию обмена — запись или чтение сектора, шпиндель должен вращаться с заданной скоростью, блок головок должен быть подведен к требуемому цилиндру, и только когда требуемый сектор подойдет к выбранной головке, начнется физическая операция обмена «полезными» данными между головкой и блоком электроники накопителя. Кроме того, головки считывают служебную информацию (адресную и сервисную), позволяющую определить и установить их текущее местоположение. Для записи информации на носитель используются различные *методы частотной модуляции,* позволяющие кодировать двоичную информацию, намагничивая зоны магнитного слоя, проходящие под головкой. Перемагничивание зоны происходит лишь в том случае, если магнитное поле в ней преодолеет некоторый порог (коэрцитивную силу), свойственный данному носителю. При считывании намагниченные зоны наводят в головке электрический сигнал, из которого декодируется ранее записанная информация. Контроллер накопителя выполняет сборку и разборку блоков информации (секторов или целых треков), включая формирование и проверку контрольных кодов, осуществляет модуляцию и демодуляцию сигналов головок и управляет всеми механизмами накопителя.

Несмотря на кажущуюся простоту конструкции, записать и потом достоверно считать информацию с диска не так-то просто. Для записи данных необходимо сформировать последовательный код, который должен быть самосинхронизирующим: при последующем считывании из него должны извлекаться и данные, и синхросигнал, что позволяет восстановить записанную цепочку битов (этим занимается сепаратор данных — узел дискового контроллера). Кроме того, напомним, что индуктивные считывающие головки воспринимают только факты изменения намагниченности участков трека. Также учтем, что физическое исполнение — магнитные свойства носителя, конструкция головок, скорость движения, «высота полета» головок и т. п. — задает предельно достижимую плотность изменения состояния намагниченности (flux density), которую хотелось бы использовать максимально эффективно. Эта плотность измеряется в количестве зон с различным состоянием намагниченности на дюйм длины трека — FCI(Flux Changes per Inch — изменений потока на дюйм) и в современных накопителях достигает десятков тысяч. Для записи на диск используются различные *схемы кодирования* (data encoding scheme), отличающиеся по сложности реализации и эффективности работы. В первых моделях накопителей использовалась *частотная модуляция FM* (Frequency Modulation). Здесь для каждого бита данных на треке отводится ячейка с окнами для представления бита и синхросигнала, что весьма неэффективно расходует предел FCI. *Более эффективна модифицированная частотная модуля-ция* MFM(Modified Frequency Modulation), в которой синхросигнал вводится только при кодировании следующих подряд нулевых битов, что позволяет удвоить плотность записи при той же плотности изменения потока. И FM, и MFM являются схемами с побитным кодированием. Более эффективны схемы группового кодирования, при которых цепочка байтов данных (сектор) предварительно разбивается на группы по несколько битов, которые кодируются по определенным правилам. Схема кодирования *RLL* (Run Length Limited), как это следует из названия, построена на ограничении длины неперемагничиваемых участков трека. Наиболее популярна схема RLL 2.7 — в ней число неперемагничиваемых ячеек лежит в диапазоне от 2 до 7. Для накопителей с высокой плотностью используется схема RLL 1.7, обеспечивающая большую надежность считывания. Существует и схема ARLL (Advanced RLL) — малораспространенная схема RLL 3.9. Схемы RLL стали работоспособными только при определенном уровне качества (стабильности характеристик), достигнутом в области технологии создания магнитных накопителей. По этим схемам происходит упаковка данных и исключение избыточных синхросигналов. Кстати сказать, FM и MFM являются разновидностями RLL: FM эквивалентна RLL 0.1; MFM — RLL 1.3. Соотношение полезной плотности записи *BPI* (Bit Per Inch — битов на дюйм) при одинаковой плотности FCI при популярных схемах кодирования следующее:

FM : MFM : RLL 1.7 : RLL 2.7=1 : 2 : 2,54 : 3.

Из-за того что линейная скорость носителя относительно головки на внутренних цилиндрах меньше, чем на внешних, для обеспечения нормальной записи при меньшей скорости приходится применять предкомпенсацию записи. Для жестких дисков в CMOS Setup имеется параметр WPcom (Write Precompensation) — номер цилиндра, начиная с которого контроллер должен вырабатывать сигнал предкомпенсации. Для накопителей со встроенным контроллером этот параметр игнорируется, поскольку они сами «знают», как работать со своими дисками.

Как уже говорилось, информация на дисках записывается и считывается посекторно, и каждый сектор имеет определенную *структуру* (формат). Не вдаваясь в глубокие подробности, отметим, что в начале каждого сектора имеется заголовок, за которым следует поле данных и поле контрольного кода. В *заголовке* имеется поле идентификатора, включающее номер цилиндра, головки и собственно сектора. В этом же идентификаторе может содержаться и пометка о дефектности сектора, служащая указанием на невозможность его использования для хранения данных. Достоверность поля идентификатора проверяется с помощью контрольного кода заголовка. Заголовки секторов записываются только во время операций низкоуровневого форматирования, причем для всего трека сразу. При обращении к сектору по чтению или записи заголовок только считывается. *Поле данных* сектора отделено от заголовка небольшим зазором (gap), необходимым для того, чтобы при операции записи головка (точнее, обслуживающая ее схема) могла успеть-переключиться из режима чтения (заголовка) в режим записи (данных). Сектор завершается *контрольным кодом поля данных — CRC* (Cyclic Redundancy Check — контроль с помощью циклического избыточного кода) или *ECC* (Error Checking and Correcting — обнаружение и коррекция ошибок). CRC-код позволяет только обнаруживать ошибки, а ECC-код позволяет и исправлять ошибки небольшой кратности. В межсекторных промежутках может размещаться сервоинформация, служащая для точного наведения головки на трек.

Современные жесткие диски внутренне могут быть организованы несколько иначе, чем в вышеописанной схеме. Индексные датчики теперь не используются — начало трека определяется из считываемого сигнала. Физическая разбивка на секторы (по 512 байт данных, которым предшествует идентификатор) может отсутствовать — группа секторов трека представляет собой единый битовый поток, защищенный избыточным кодированием, из которого вычисляется блок данных, находящийся в требуемой позиции (так называемый ID-less format). Для коррекции данных применяются избыточные коды Рида-Соломона, позволяющие большинство ошибок исправлять налету, не требуя повторного считывания блока данных (и дополнительного оборота диска).

Для того чтобы диск можно было использовать для записи и считывания информации, он должен быть *отформатирован.* Форматирование может разделяться на два уровня:

- Низкоуровневое форматирование (LLF - Low Level Formatting) — формирование заголовков и пустых (расписанных заполнителем) полей данных всех секторов всех треков. При форматировании выполняется и верификация (проверка читаемости) каждого сектора, и в случае обнаружения неисправимых ошибок считывания в заголовке сектора делается пометка о его дефектности.
- Форматирование верхнего уровня заключается в формировании логической структуры диска (таблиц размещения файлов, корневого каталога и т. п.), соответствующее файловой системе применяемой ОС. Эта процедура выполнима только после низкоуровневого форматирования.

Итак, структура трека — последовательность секторов — задается при его форматировании, а начало трека определяется контроллером по сигналу от индексного датчика или иным способом. Нумерация секторов, которая задается контроллеру при форматировании, может быть достаточно произвольной — важно лишь, чтобы все секторы трека имели уникальные номера в пределах допустимого диапазона. При обращении к сектору он ищется по идентификатору, а если за оборот диска (или за несколько оборотов) сектор с указанным номером не будет найден, контроллер зафиксирует ошибку Sector Not Found — сектор не найден. Забота о поиске сектора по его заголовку, помещение в его поле данных записываемой информации, снабженной контрольным кодом, а также считывание этой информации и ее проверка с помощью CRC- или ECC-кода лежит на контроллере накопителя. И, конечно же, контроллер управляет поиском затребованного цилиндра и коммутацией головок, выбирая нужный трек.

1.6. Контроллеры дисковых накопителей

Контроллеры современных накопителей представляют собой сложные устройства, построенные на основе микропроцессоров. В состав контроллера входят узлы, непосредственно связанные с обработкой и формированием сигналов блока HDA (Head Disk Assembly), — коммутатор головок, схемы кодирования-декодирования данных с усилителями сигналов, схемы управления позиционированием, шпинделем и, если надо, сменой носителя. Кроме того, контроллер может содержать буферную память (кэш), объем которой может достигать сотен килобайт, хотя и это не предел. Рассмотрим, на что расходуется «интеллект» контроллера.

Хранение данных на магнитном носителе всегда сопровождается появлением *ошибок,* причин у

которых может быть множество: дефект поверхности носителя, случайное перемагничивание участка носителя, попадание посторонней частицы под головку, неточность позиционирования головки над треком, колебание головки по высоте, вызванное внешней вибрацией (ударом) корпуса накопителя, уходом различных параметров (из-за старения, изменения температуры, давления и т. п.). Независимо от причин, ошибки должны быть выявлены и по возможности исправлены. Для контроля достоверности хранения применяется CRC-код для поля данных, который позволяет фиксировать ошибки некоторой кратности, а ECC-код при большей избыточности даже исправлять ошибки. Если сектор считывается с ошибкой, контроллер автоматически выполнит повторное считывание, и при случайности ошибки велик шанс правильного считывания сектора. Однако, если ошибка вызвана, например, неточностью позиционирования головки на середину трека, связанной с уходом параметров, повторное считывание может и не дать положительного эффекта. Накопитель с шаговым приводом головки в этой ситуации может только повторить позиционирование — вернуться на нулевой трек и снова «дошагать» до нужного — иногда это помогает. У привода с подвижной катушкой больше возможностей для поиска положения головки, оптимального для считывания данных. Для этого следящая система может покачать головку относительно центрального положения, заданного сервометками, и найти точку, где данные читаются верно. Если данные так и не удается считать верно, контроллер обязан сигнализировать об этом установкой бита ошибки контрольного кода в байте состояния, на что программа может отреагировать сообщением вида «CRC Data Error».

Если контроллеру никак не удается достоверно прочитать записанные в сектор данные, такой сектор должен быть исключен из дальнейшего использования. Если этого не сделать, бесчисленные повторные попытки обращения к этому сектору будут отнимать массу времени, а результата все равно не будет. На уровне накопителя отметка о дефектности блока делается в заголовке сектора, запись в который, как известно, производится только во время низкоуровневого форматирования. Однако современные встроенные контроллеры дисков сами обрабатывают обнаружение дефектных секторов и вместо них подставляют резервные, так что для пользователя дефектные секторы у дисков ATA и SCSI не видны. Контроллер помечает дефектный сектор специальной пометкой в заголовке, а вместо него использует сектор из резервной области, которая имеется в конце каждого трека. Однако простая подстановка номера отказавшего сектора в заголовок сектора из резервной области приведет к нарушению последовательности секторов и, следовательно, к падению производительности при обращении к длинному блоку данных. Более «хитрые» контроллеры производят сдвиг данных во всех секторах, следующих за дефектным, на один сектор назад (Defective Sector Slipping), после чего обращение к последовательной цепочке секторов снова станет «гладким» (без скачков вперед-назад) и быстрым. Конечно, со временем настанет момент, когда все резервные блоки будут использованы, и тогда появление видимого дефектного блока будет сигналом к замене накопителя.

Надежность считывания в большой степени зависит от точности позиционирования головок относительно продольной оси трека. Позиционирование, обеспечиваемое сервоприводом (особенно с выделенной сервоповерхностью), может быть и не оптимальным для каждой головки и требовать коррекции. Главным образом, эта коррекция необходима из-за изменения рабочей температуры. «Умный» контроллер хранит карту отклонений для цилиндров и головок, которую он создает и периодически корректирует в процессе работы. Процесс автоматической *термокалибровки* накопителя (Thermal Calibration, или T-Cal) запускается встроенным контроллером автоматически, и для системы, в которую он установлен, момент запуска этого процесса случаен. Процесс автоматической термокалибровки заметен пользователю: винчестер, к которому нет обращений, вдруг «начинает жить своей жизнью», выполняя серию позиционирований (слышно по характерным звукам). Во время рекалибровки доступ к данным накопителя приостанавливается, что не всегда допустимо. Накопители, предназначенные для мультимедийных целей, а также используемые как мастер-диски для записи данных на лазерный диск, должны обеспечивать довольно длительный непрерывный поток передачи данных. Их контроллеры не должны позволять себе «вольности» термокалибровкой приостанавливать сеансы передачи данных.

Кроме термокалибровки, есть еще один процесс, асинхронно запускаемый контроллером диска, — так называемое *свипирование* (Sweeping): если к диску долгое время отсутствуют обращения, он перемещает головки в новое, случайным образом определенное положение. Этот процесс напоминает «хранитель экрана» монитора (Screen Saver) и предназначен для выравнивания степени износа поверхности диска.

Современные контроллеры имеют встроенную *кэш-память,* от размера которой и алгоритма использования сильно зависит производительность обмена данными. Общепринятой технологией кэширования диска является *упреждающее считывание* (Read Ahead). Суть его проста: если контроллер получает запрос на чтение сектора, то он автоматически считает в буфер и секторы, следующие за запрошенным. Весьма вероятный запрос к следующему сектору будет обслужен из буфера без задержки (Latency), которая возможна из-за естественной несинхронизируемости действий операционной системы и приложений с вращением диска. Более «ловкие» контроллеры идут дальше: они считывают в буфер весь трек, как только выполнится команда позиционирования, а когда придет следующая за ней команда

чтения, данные уже будут в буфере. Такие хитрости (Zero Latency — нулевая задержка) позволяет система команд интерфейса SCSI. Как и для всякой кэш-памяти, для эффективности встроенного кэша накопителя существенным фактором является алгоритм выделения памяти и замещения старых записей. Как обычно, замене подлежат наиболее старые записи. Вопрос о размере выделяемых областей для упреждающего чтения может решаться исходя из текущей статистики обращений. Контроллер с *адаптивным кэшированием,* заметив, что последние запросы чтения являются одиночными, перестанет выделять большие области под упреждающее чтение. Если характер запросов изменится, адаптивный контроллер примет соответствующие решения. Кроме того, отпечаток на алгоритм кэширования накладывает и многозадачный характер современных операционных систем и их дисковых запросов. Таким образом, многозадачность проникает и во встроенные контроллеры дисков.

Дисковые накопители являются, пожалуй, той частью компьютера, отказ которой оборачивается самыми крупными убытками (если потерянные данные не имеют дублей). Естественно, *надежность* их пытаются повышать всеми возможными способами, но отказы все-таки случаются. Отказы разделяются на предсказуемые и непредсказуемые. *Предсказуемые отказы* (Predictable Failure) появляются в результате постепенного ухода каких-либо параметров от номинальных значений, когда этот уход перейдет некоторый порог. Если наблюдать за такими параметрами накопителей, как время разгона до заданной скорости, время позиционирования, процент ошибок позиционирования, высота полета головок, производительность (зависящей и от числа вынужденных повторов для успешного выполнения операций), количество использованных резервных секторов и других параметров, то становится возможным предсказание отказов. Сообщение об их приближении операционной системе и (или) пользователю позволяет предпринять необходимые меры и предотвратить крупный ущерб. Целям предупреждения отказов служит технология *S.M.A.R..T.* (Self-Monitoring, Analysis and Reporting Technology — технология самонаблюдения, анализа и сообщения), применяемая в современных накопителях. Эта технология, разработанная фирмой Seagate, имеет корни в технологии IntelliSafe фирмы Compaq и PFA (Predictive Failure Analysis — анализ предсказуемых отказов) фирмы IBM.

Задачи слежения за параметрами накопителя возлагаются на контроллер, а программному обеспечению компьютера остается только периодически интересоваться, все ли в порядке в накопителе, или грядет беда. Спецификации S.M.A.R.T. существуют в двух версиях — для интерфейсов ATA и SCSI, которые различаются как по системам команд, так и по способам сообщений состояния. Конечно, остаются и *непредсказуемые отказы* (Non-Predictable Failure), которые случаются внезапно. Они чаще всего вызываются разрушениями электронных схем под действием импульсных помех, механические части страдают от ударов. Снижать вероятность непредсказуемых отказов позволяет совершенствование технологий производства компонентов.

1.7. Параметры дисковых накопителей

После рассмотрения устройства и работы дисковых накопителей должен быть понятным смысл их основных параметров.

Форматированная емкость (Formatted Capacity) представляет собой объем хранимой полезной информации — то есть сумму полей данных всех доступных секторов. Неформатированная емкость (Unformatted Capacity) представляет собой максимальное количество бит, записываемых на всех треках диска, включая и служебную информацию (заголовки секторов, контрольные коды полей данных). Соотношение форматированной и неформатированной емкости определяется форматом трека (размером сектора), но, поскольку для рядового пользователя свободы выбора формата нет, практический интерес представляет только форматированная емкость диска, которая указывается для стандартного размера сектора (512 байт). Напомним, что мегабайт — это больше, чем миллион байт, а гигабайт — больше чем миллиард, поэтому вольности в применении этих единиц могут приводить в недоумение.

Интерфейс (Interface) определяет способ подключения накопителя. Для накопителей со встроенным контроллером распространены интерфейсы ATA, он же IDE (все их разновидности будут рассмотрены ниже) и SCSI. Сравнение интерфейсов ATA и SCSI будет сделано ниже. Из накопителей с внешним контроллером в настоящее время актуальны только НГМД со своим собственным интерфейсом. Устаревшие жесткие диски имели интерфейсы ST-506/412 и ESDI, напоминающие интерфейс НГМД.

Быстродействие и производительность характеризуется следующими параметрами:

Внутренняя скорость передачи данных (Internal Transfer Rate), измеряемая в количестве бит (мегабит) в секунду, передаваемых между носителем и буферной памятью контроллера, задает физический предел производительности накопителя. Здесь сознательно объем передаваемой информации измеряют в битах, а не байтах: из длинной цепочки бит сектора «полезными» являются только байты поля данных, а остальные — «накладные расходы». Так что получать среднюю скорость передачи полезной информации (Кбайт/с) простым делением внутренней скорости на 8 (число бит в байте) некорректно. Для НГМД скорость обмена составляет 250 Кбит/с при дискетах 360 Кбайт, 500 Кбит/с для дискет 1,2 и 1,44 Мбайт. У современных винчестеров средней производительности скорость передачи составляет 60—80 Мбит/с, а у достаточно современных (с частотой вращения 7200 об/мин и PRML-кодированием) приближается к 200 Мбит/с.

Внешняя скорость передачи данных (External Transfer Rate), измеряемая в количестве килобайт (мегабайт) полезных данных в секунду, передаваемых по шине внешнего интерфейса, зависит от быстродействия электроники контроллера, типа интерфейсной шины и режима обмена. Для интерфейса ATA в режиме обмена PIO Mode 0 скорость составляет 3,3 Мбайт/с, PIO Mode 4 — 16,6 Мбайт/с, Ultra-DMA в режиме 2 — 33 Мбайт/с. Для шин SCSI ограничения скорости в зависимости от типа электрического интерфейса составляют 5, 10, 20 или 40 Мбайт/с, а для оптического канала и 100 Мбайт/с. Рядом с внешней скоростью обычно указывают и объем внутренней кэш-памяти, а также особенности ее организации (многосегментность, адаптивность). По характеристикам кэша можно судить о том, в какой степени может быть интересна внешняя скорость.

Время перехода на соседний трек (Track-to-Track Seek), измеряемое в миллисекундах, характеризует быстродействие системы позиционирования. Для современных жестких дисков характерно время перехода в 0,8-2 мс, причем для записи оно несколько больше, чем для считывания (записывать лучше при более точном позиционировании).

Среднее время поиска (Average Seek Time) определяется по смеси обращений к случайным цилиндрам и для современных жестких дисков составляет около 8-10 мс. Чем больше объем накопителя, тем больше и время поиска: большее число головок труднее быстро перемещать; большее число цилиндров или увеличивает длину перемещения головок, или повышает требования к точности позиционирования.

Максимальное время поиска (Maximum Seek Time) примерно в два раза превышает среднее и относится к самым дальним переходам между крайними цилиндрами.

Скорость вращения шпинделя (Spindle Speed), измеряемая в оборотах в минуту (RPM — Rotates Per Minutes), позволяет косвенно судить о производительности (внутренней скорости). Для ранних жестких дисков значение 3600 об/мин является обычной, (5400 - 7200) — обычна для современных накопителей, 10000 об/мин является высокой скоростью вращения.

Среднее ожидание сектора при одиночном обращении (Average Latency) обычно составляет половину времени полного оборота (для 3600 об/мин — 8 мс). Новейшие контроллеры ухитряются уменьшить это значение чуть ли не до нуля (Zero Latency), начиная считывание всего трека сразу после позиционирования (не дожидаясь команды).

Программы, определяющие производительность накопителей, сознательно или неумышленно могут измерять различные параметры производительности. Если программа измеряет время передачи данных, находящихся в кэш-буфере контроллера, она покажет внешнюю скорость (предел возможности интерфейса или накопителя). Этот результат можно получить при зацикленном чтении блока данных, размер которого заведомо меньше объема кэш-памяти. Заставить программу достоверно измерить внутреннюю скорость затруднительно — для этого нужно ухитриться исключить влияние как буферизации, так и процесса поиска. Интерес представляет длительно выдерживаемая скорость передачи данных (Sustained Transfer Rate), на которую влияют все составляющие: внутренняя и внешняя скорости, время позиционирования, задержка подхода сектора, количество ошибок позиционирования и чтения. Минимальное гарантированное значение этой скорости определяет возможность применения накопителя для мультимедийных приложений реального времени.

К параметрам *организации* и *конфигурации* относятся следующие:

- *Количество физических дисков* (Disks) или рабочих поверхностей (Data Surfaces), используемых для хранения данных. Современные накопители с небольшой высотой имеют малое (1-2) количество дисков для облегчения блока головок. Большее число дисков (и высоту) имеют старые накопители и современные накопители большой емкости.
- *Количество физических головок чтения-записи* (Read/Write Heads), естественно, совпадающее с числом рабочих поверхностей.
- *Физическое количество цилиндров* (Cylinders) с нескольких сотен, характерных для первых винчестеров, возросло до нескольких тысяч.
- *Размер сектора* (Bytes Per sector) обычно составляет 512 байт.
- *Расположение сервометок* или сервоголовок (Servo Head) может быть на выделенной поверхности (Dedicated Servo), на рабочих поверхностях (Embeded Servo) или гибридным (Hybrid Servo).
- *Метод кодирования-декодирования* данных (Recording Method или Data Encoding Scheme) может быть MFM (FM почти и не применяли), RLL (ARLL), PRML. Последний является наиболее прогрессивным.

Для дисков со встроенными контроллерами ATA физическая организация маскируется *логической конфигурацией* (геометрией) — числом *логических цилиндров, головок и секторов* (Logical Cylinders, Heads, Sectors). Эти параметры могут отличаться от физической организации, важно лишь, чтобы общее число секторов, полученное перемножением этих трех логических параметров, не превышало числа секторов, физически существующих на всех рабочих поверхностях накопителя. Контроллеры дисков с интерфейсом SCSI, а также современные ATA-контроллеры используют *линейную логическую адресацию*

LBA (Logical Block Addressing) секторов вместо традиционной *трехмерной геометрии CHS* (Cylinder-Head- Sector). Соображения, по которым выбирается способ трансляции логических параметров, передаваемых контроллеру по внешнему интерфейсу, в физические, по которым управляется HDA, будут изложены ниже.

Надежность устройства (Reliability) и *достоверность* хранения данных (Data Integrity) характеризуются следующими параметрами:

- *Ожидаемое время до отказа MTBF* (Mean Time Before Failure), измеряемое в сотнях тысяч часов, является, естественно, среднестатистическим показателем для данного изделия. Реально столько часов (100 000 часов — это более 10 лет) испытания проводить, естественно, невозможно. На самом деле делается выборка из большой группы устройств, из которых за вполне обозримое время испытаний какая-то часть выйдет из строя. По зафиксированному потоку отказов теория вероятностей позволяет вычислить это условное ожидаемое время безотказной работы. Значение MTBF, равное 100 000 часов, считается малым; 200 000-400 000 часов — нормальным, а 1 000 000 часов — высоким показателем надежности. Но, как говорится, «столько не живут», а возможный отказ винчестера в течение года (всего-то 8 760 часов) вполне уложится в статистический показатель (если у вас за год не отказала значительная партия устройств). Иногда указывают и ожидаемое время наработки на отказ *POH* (Power On Hours), которое учитывает только время работы устройства (MTBF не различает, включено устройство или нет).
- Более ценным для пользователя является *гарантийный срок* (Limited Warranty), в течение которого изготовитель (или поставщик) обеспечивает ремонт или замену отказавшего устройства. Примечательно, что даже при MTBF=800000 часов (91 год) изготовитель дает гарантию всего на 3-5 лет.
- *Вероятность неисправимых ошибок чтения* (Nonrecoverable Read Errors per Bits Read) для современных винчестеров имеет порядок одной ошибки на 10^{14} считанных бит. Оценить, много это или мало, можно следующим образом. Пусть винчестер постоянно находится в работе и к нему непрерывно идут обращения со средней производительностью чтения, которую «на глазок» можно оценить в 1 Мбайт/с (это соответствует, например, сильно загруженному диску сервера). Тогда простая арифметика показывает, что раз в 115 дней будут возникать ошибки, не восстанавливаемые (но обнаруженные!) схемами ECC-контроля. Вполне вероятно, что повторное считывание позволит считать сектор без ошибок.
- *Вероятность исправимых ошибок* (Recoverable Read Errors per Bits Read) имеет порядок единицы на 10 считанных бит. Если бы не было ECC-контроля (или если схема контроля неисправна, с чем доводилось сталкиваться на практике), этот поток ошибок (в том же примере — чаще, чем раз в три часа) сделал бы работу с таким накопителем просто невыносимой.
- *Вероятность ошибок поиска* (Seek Errors per seek) характеризует качество сервосистемы. Для современных винчестеров характерна вероятность одной ошибки на 10 операций поиска. Эти ошибки (при малом их числе) вполне безобидны, поскольку наличие номера цилиндра в заголовке каждого сектора не позволяет «промахнуться» при выполнении операций чтения или записи. Повторение операции поиски только слегка снижает среднее время доступа.

Физические параметры включают *ширину* (Width), *высоту* (Heigth) и *глубину* (Depth) корпуса накопителя, измеряемые в дюймах (inches) или миллиметрах, и *массу* (Weight), измеряемую в фунтах (lЬ) или килограммах.

Потребляемая мощность определяется номинальными и пиковыми токами, потребляемыми по цепям +5 В и +12 В.

Условия эксплуатации и хранения определяют возможные *диапазоны температур, атмосферного давления, влажности и силы допустимых ударов.* Вполне понятно, что условия эксплуатации (Operating Conditions) несколько жестче, чем условия хранения (Non-Operating Conditions).

Контрольные вопросы

1. Что относится к внешней памяти и каково её отличие от внутренней памяти?
2. В каких устрорйствах внешней памяти используется прямой, а в каких - последовательный доступ к информации?
3. Перечислите основные характеристики внешней памяти.
4. Из чего делают пластины магнитных дисков и какие магнитные покрытия в них используются?
5. Опишите основные типы магнитных головок, используемых в наконителях на жестких магнитных дисках.
6. Какие приводы магнитных головок используются в накопителях на магнитных дисках? Каковы принципы их позиционирования?
7. Опишите основные принципы хранения информации на внешних носителях.
8. Каковы особенности записи и считывания информации на магнитные диски?
9. Что такое физическое и логическое форматирование магнитных дисков?

10. Что входит в функции контроллеров современных дисковых накопителей?
11. Перечислите основные параметры дисковых накопителей.

В данной главе использованы материалы [2, 3, 4, 6]. Более подробную информацию по общим вопросам организации внешней памяти, ее интерфейсам, конструктивным особенностям накопителей и носителей, их параметрам и т.п. можно найти в [2, 3, 6, 7, 8].

2.Логическая структура диска

С аппаратной точки зрения любой диск можно представить как совокупность секторов, адресуемых тем или иным способом (CHS или LBA), и каждый сектор может быть записан и считан (только целиком) независимо от других. Однако для большинства прикладных программ интерес представляет не обращение к отдельным секторам, а возможность обращения к файлам, которые могут занимать произвольное, причем, возможно, и не целое количество секторов. Для облегчения обращения к файлам и упорядочения использования пространства секторов диска в состав любой операционной системы входит *файловая система,* тесно связанная с логической структурой диска.

2.1. Разделы и логические диски

Операционная система представляет дисковую память в виде набора *логических дисков,* или приводов (logical drive). Каждому логическому диску присваивается свое логическое имя: A:, B: — для дискет, C:, D:, E: и следующие буквы — для жестких дисков, CD-ROM и прочих устройств. Рассмотрим логическую организацию физических дисков, чтобы понять, каким образом несколько логических дисков (например, C:, D: и E:) могут оказаться расположенными на одном винчестере.

Логический диск — это совокупность секторов с последовательно нарастающими номерами. Самый первый сектор логического диска называется *загрузочным* (boot sector). В этом секторе всегда хранится описатель параметров диска и файловой системы и дополнительно может содержаться программа загрузки операционной системы (загрузчик). Если на диске с загрузчиком присутствуют и соответствующие файлы ОС и эта ОС может быть загружена с него на компьютер, такой диск называется *системным.*

Поскольку жесткий диск (hard drive) в общем случае не является съемным (иное его название — fixed disk — фиксированный диск), а операционных систем, которые хочется использоваться на одном компьютере; может быть несколько, и каждая из них претендует на свою логическую организацию диска, договорились о возможности разбиения жесткого диска на несколько *независимых разделов.* Дискеты в таком разбиении не нуждаются, поскольку их легко сменить, да и маленький объем не располагает к делению. Кроме обеспечения «плюрализма» операционных систем, разбиение диска на разделы позволяет уменьшить размеры логических дисков, что бывает выгодно из-за ограничений файловых систем. Разбиение винчестера на логические диски используют и для упорядочивания использования дискового пространства. Структуру жесткого диска поясняет рис. 2.1.

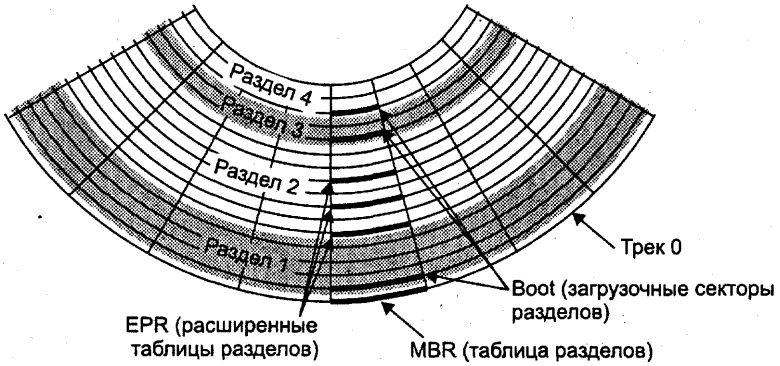

Рис. 2.1. Структура жесткого диска

Физический жесткий диск может быть разбит на несколько *разделов* (partition). Информация о структуре диска — *таблица разделов* (partition table) — хранится в *главной загрузочной записи MBR* (Master Boot Record), находящейся в общеизвестном месте — цилиндр 0, головка 0, сектор 1. В начале этого сектора расположена программа *главного загрузчика* (master boot), а за ней имеется таблица разделов, содержащая четыре описателя разделов. Каждый описатель задает границы разделов, причем в двух системах: CHS (координаты начала и конца) и LBA (начало и длина). Разделы, как правило, начинаются точно по границе цилиндра (координаты N, 0, 1), кроме первого, начинающегося обычно с первой головки нулевого цилиндра (0, 1, 1), поскольку под нулевой головкой расположен сектор с MBR. Заканчиваться разделы должны на границе цилиндра, что позволяет через номера конечной головки и конечного сектора определить число головок и секторов на треке.

Описатель задает и атрибуты раздела — системный код и флаг активности. *Флаг активности* указывает главному загрузчику, какой раздел ему следует загружать. Флаг активности может быть установлен только для одного раздела диска (или ни у одного). *Системный код* определяет тип раздела; операционная система для своей файловой системы может использовать разделы только известных ей типов. Таблица разделов может заполняться как с начала, так и с конца. Если разделов меньше четырех, то свободные описатели обнулены. Свободные описатели, равно как и используемые, могут располагаться в любом месте таблицы (в начале, середине, конце). Формирование таблицы разделов —

конфигурирование жесткого диска — как правило, выполняется утилитой FDISK.

Структура *MBR* следующая:

• байты 000h-lBDh — код загрузки Boot-сектора активного раздела;

• байты 1BEh-lCDh, 1CEh-lDDh, 1DEh-lEDh, lEEh-lFDh-описатели разделов *(Partition Entry),* 16-байтные структуры (табл. 2.1);

• байты 1FE=55h, 1FF=AAh — сигнатура *MBR* (слово AA55h).

Таблица 2.1. Структура описателя раздела

Смещение	Длина, байт	Адрес в секторе для раздела №				Назначение
		1	2	3	4	
00h	1	1BEh	1CEh	1DEh	1EEh	*Boot Flag* — флаг активности раздела: 80h=активный (загружаемый, помещается в регистр *DL),* 0=нет
01h	1	1BFh	1CFh	1DFh	1EFh	*Begin Head* — номер начальной головки, помещается в регистр *DH*
02h	2	1C0h	1D0h	1E0h	1F0h	*Begin SecCyl* — номер начального сектора и цилиндра в формате загрузки регистра *CX* в BIOS *INT 13h*
04h	1	1C2h	1D2h	1E2h	1F2h	*System Code* — системный код (см. табл. 2.2)
05h	1	1C3h	1D3h	1E3h	1F3h	*Ending Head* — номер конечной головки
06h	2	1C4h	1D4h	1E4h	1F4h	*Ending SecCyl* — номер конечного сектора и цилиндра
08h	4	1C6h	1D6h	1E6h	1F6h	*Starting* Sector[1] — номер начального сектора раздела (адрес в системе LBA)
0Ch	4	1CAh	1DAh	1EAh	1FAh	*Num Sectors* — количество секторов в разделе

[1] Иное название поля — *Relative Sector* — относительный номер сектора.

Для главного загрузчика загрузить активный раздел означает найти положение первого (загрузочного) сектора этого раздела, загрузить сектор в память и передать управление на его начало. Дальнейшие действия выполняются уже загрузчиком активного раздела. Байт, содержащий флаг активности раздела, после отработки главного загрузчика будет использоваться как физический номер диска, с которого производится загрузка. Из-за этого появляется ограничение на номер физического загрузочного диска - поскольку признаком активного раздела является значение 80h, то загрузочным может быть только первый физический диск. Перед загрузкой активного раздела главный загрузчик проверяет, является ли найденный раздел единственным активным; и если нет, то останавливается с сообщением «Invalid partition table». Если активный раздел не найден, главный загрузчик останавливается с сообщением «Missing operating System» (раньше он через BIOS *INT 18h* пытался запустить интерпретатор BASIC, зашитый в ПЗУ).

При исполнении прерывания BIOS INT 19h (вызов начального загрузчика, bootstrap loader) сектор с MBR загружается в память по адресу 0000:7000h и, если в его конце обнаружена сигнатура загрузочного сектора (AA55h), управление передается на его начало. При исполнении начальный загрузчик первым делом перемещает (копирует) свой код (и таблицу разделов) по адресу 0000:0600h и продолжает свое исполнение уже из новой области. Найдя описатель активного раздела, главный загрузчик загружает в память по адресу 0000:7C00h его первый сектор, при этом регистр *SI* указывает на описатель активного раздела (таблица разделов оказывается в памяти по адресам 0000:07B0-07FDh). Первый сектор загружаемого раздела ищется просто: в регистр *DX* заносится слово 0, а в *CX* — слово 2 из описателя активного раздела (см. табл. 2.1). После этого остается лишь задать адрес буфера в памяти (в *ES:BX),* функцию чтения одного сектора (AX=0201h) и вызвать дисковый сервис BIOS *INT 13h.* Если считать сектор без ошибок не удается (делается до 5 попыток), главный загрузчик останавливается с сообщением «Error loading operating system». Главный загрузчик инвариантен по отношению к загружаемым операционным системам и дискам. Однако традиционный загрузчик в принципе не способен загрузить раздел, находящийся дальше, чем через 8,4 Гбайт от начала диска, поскольку пользуется исключительно CHS-описанием границ разделов. Для больших дисков главный загрузчик должен использовать линейные описания разделов.

Структура раздела зависит от его типа. Некоторые системные коды (типы разделов) приведены в табл. 2.2. Далее ограничимся описанием разделов и логических дисков для DOS/Windows 9x.

Таблица 2.2. Коды и типы разделов жесткого диска

Код	Раздел; ОС, с которой введен; файловая система; объем
00	Неизвестный (неформатированный) раздел
01	DOS FAT12; MS-DOS 2.0; до 15 Мбайт
02	XENIX root
03	XENIX usr
04	DOS FAT16; MS-DOS 3.0; до 32 Мбайт
05	(DOS) Extended; MS-DOS 3.3; до 2 Гбайт
06	DOS FAT16 (Big DOS); MS-DOS 4.0; до 2 Гбайт
07	OS/2 HPFS или Windows NT NTFS
08	AIX
09	AIX bootable
0A	OS/2 Boot Manage
08	Win95 FAT32; Windows 95 OSR2; 512 Мбайт-2 Тбайт
0C	Win95 FAT32 (LBA); Windows 95 OSR2; 512 Мбайт-2 Тбайт
0E	Win95 FAT16 (LBA); Windows 95 OSR2; 32 Мбайт-2 Гбайт
0F	(Win95) Extended (LBA); Windows 95 OSR2
11	Hidden DOS FAT12 .
14	Hidden DOS FAT16
16	Hidden DOS FAT16
40	Venix 80286
51	Novell?
52	Microport
63	GNU HURD
64	Novell Netware 2
65	Novell Netware 3
75	PC/IX
80	Old MINIX
81	Linux/MINIX
82	Linux Swap
83	Linux
85	Linux extended
93	Amoeba
94	Amoeba BBT
A5	BSD/386
B7	BSDI fs
B8	BSDI swap
C7	Syrinx
DB	CP/M
E1	DOS access
E3	DOS R/O
F2	DOS secondary
FF	BBT

Разделы с кодами (01, 04, 06, 0B, 0C, 0E) являются *первичными разделами* (primary partition) DOS/Windows. Утилита FDISK из MS DOS и Windows 9x позволяет создавать не более одного первичного раздела, хотя в принципе их может быть и больше. Первичный раздел содержит один *логический диск* (logical drive). В стандартном случае, когда на диске имеется один первичный раздел, для первого винчестера на нем будет диск C:, для второго — D: и т.д. Операционные системы MS DOS и Windows 9x «не любят», когда на одном диске более одного первичного раздела, а также когда первичный раздел не является первым в таблице разделов. Другие ОС (например, Linux) более лояльны к количеству и расположению разделов. В первом секторе логического диска (boot sector) находится загрузчик и описатель типа файловой системы (FAT12, FAT16, FAT32) и структуры диска. Этот загрузчик загружает ОС, расположенную на данном диске (если он системный). За загрузчиком располагается несколько копий таблицы размещения файлов FAT (File Allocation Table), корневой каталог и собственно область данных диска. Разные коды первичных разделов указывают на различную разрядность FAT (см. ниже), новые типы вводились по мере роста размеров винчестера. С Windows 95 OSR2 появились новые типы разделов для FAT32 и FAT16 (0Ch, 0Eh), специально для дисков, поддерживающих адресацию LBA. Заметим, что в каждом описателе разделов задаются как трехмерные границы раздела (начальные и конечные номера цилиндра, головки и сектора), так и линейные (номер

начального сектора и их количество), но долгое время по старинке использовали только трехмерные описатели.

Среди разделов DOS/Windows 9x активным может быть только первичный раздел (только его загрузчик может быть исполнен).

Расширенный раздел (extended partition, код 05 или 0F) служит для организации произвольного количества логических дисков (на рисунке 2.1 раздел 2 является расширенным). Первый сектор расширенного раздела аналогичен MBR (но загрузчик отсутствует) и *содержит расширенную таблицу разделов EPR* (Extended Partition Record), той же структуры, но с некоторыми оговорками. Первый описатель задает *вторичный* (secondary) раздел, отведенный под очередной логический диск; в нем указывается код раздела с файловой системой (для DOS/Windows это FAT с кодами 04h, 06h, 0Bh, 0Ch или 0Eh, для других ОС — свои). В этом описателе, как обычно, задаются координаты начала и конца раздела с логическим диском (трехмерные и линейные). Если этот логический диск занимает не весь объем расширенного раздела, то второй описатель тоже имеет код 05 или 0F и указывает на положение сектора со следующей расширенной таблицей разделов. Остальные описатели не используются (их коды нулевые). Если свободного места в разделе уже нет, то и второй описатель не используется. В следующей расширенной таблице разделов действуют те же правила. Эта цепочка заканчивается на расширенной таблице, у которой во втором описателе стоит нулевой код раздела. Заметим, что второй описатель в расширенных таблицах может только указывать на положение следующей расширенной таблицы. Часть пространства расширенного раздела может оставаться не распределенной, в дальнейшем она может быть использована под логические диски. Цепочка расширенных таблиц разделов должна быть непрерывной, неветвящейся (используются только два описателя, и только второй может указывать на следующую таблицу) и незацикленной (второй описатель не должен ссылаться на ту же таблицу или предыдущую в цепочке). Несоблюдение первых двух условий ведет «только» к потере логических дисков (их система не найдет). Несоблюдение последнего условия может привести к «зависанию» ОС при загрузке: она зациклится на бесконечном определении повторяющихся логических дисков. Код (05 или 0F) расширенного раздела не несет никакой информации о файловой системе, и данный тип раздела используется как указатель на расширенную таблицу рядом ОС, и отличных от DOS/Windows. Координаты расширенных таблиц разделов обычно имеют вид N, 0, 1.

По расположению на физическом диске расширенные разделы являются вложенными друг в друга; все они располагаются в области, описанной в главной таблице разделов как расширенный раздел. В главной таблице может быть описан лишь один расширенный раздел.

В трактовке описателей расширенных разделов, к сожалению, возможны разночтения. Например, FDISK из MS DOS 6.22 честно расставляет трехмерные описатели (в логической геометрии CHS), но это возможно лишь для дисков не более 8,4 Гбайт- При этом в расширенных таблицах линейные описатели логических дисков (вторичных разделов) ставятся не относительно начала физического диска (сектора с MBR), а относительно сектора с EPR. Здесь же линейные ссылки *(starting sector)* на очередную EPR (с кодом 05) ставятся относительно первой в цепочке (линейного адреса расширенного раздела, описанного в MBR). Поле длины в описателе логического диска (вторичного раздела) относится именно к нему. В ссылке на очередную EPR поле длины определяет расстояние до EPR, следующей за ней. Сборка цепочки логических дисков именно по этому полю позволит удалять логические диски из середины цепочки, не разрывая ее.

Если расширенные разделы имеют код 0Fh, то линейные адреса всех элементов таблиц будут указываться относительно начала физического диска (так поступает новый FDISK, и это более естественно, поскольку при этом описатель LBA является эквивалентом описателя CHS).

Каждый логический диск из расширенного раздела имеет ту же структуру, что и вышеописанный первичный раздел. Он так же начинается с загрузочного сектора (только загрузчик никогда не исполняется), в котором имеется описание структуры логического диска. Координаты загрузочных секторов логических дисков обычно имеют вид N, 1, 1. Операционная система назначает логическим дискам расширенных разделов имена (буквы), остающиеся после дисков первичных разделов. Так, если имеется один жесткий диск, и у него имеется первичный раздел и вторичный, разбитый на два логических диска, то мы увидим следующее:

- C: — первичный раздел;
- D: — первый логический диск расширенного раздела;
- E: — второй логический диск расширенного раздела.

Теперь если добавить второй жесткий диск (всего с одним первичным разделом), то картина изменится:

- C: — первичный раздел первого диска (остался на месте);
- D: — первичный раздел второго диска (новый);
- E: — первый логический диск расширенного раздела первого диска (тот, что был D:);
- F: — второй логический диск расширенного раздела первого диска (тот, что был E:).

Если у нового диска был бы расширенный раздел со своими логическими дисками, то они бы заняли следующие буквы (G:, H: ....). О механизме присвоения логических имен следует помнить, устанавливая программы на компьютер, к которому эпизодически подключают дополнительные винчестеры. Незыблемое имя (C:) будет только у первичного раздела винчестера, подключенного ведущим к первому контроллеру ATA (если используется SCSI, то все немного сложнее).

2.2. Файловые системы FAT12, FAT16 и FAT32

Операционная система выбирает способ организации хранения информации на носителе в зависимости от его типа и объема, а также пожеланий пользователя. Гибкие диски для AT-совместимых компьютеров всегда организованы в виде одного логического диска со структурой FAT12. Жесткий диск может содержать один или несколько разделов, предназначенных для одной или нескольких различных операционных систем, а разделы в свою очередь могут состоять из одного или нескольких логических дисков.

2.2.1. Размещение информации на логических дисках

Логический диск (том) файловой системы типа FAT, состоит из четырех основных областей (рис. 2.2), расположенных в следующем порядке:

- резервная область;
- область таблиц размещения файлов (FAT1 и FAT2);
- область корневого каталога (не существует в FAT32);
- область файлов и каталогов.

Рис. 2.2. Организация данных на логических дисках

В первом секторе логического диска с системой FAT располагаются загрузочный сектор и блок параметров BIOS. В документации Microsoft они обозначаются как Boot Sector (BS) и BIOS Parameter Block (BPB) соответственно. Начальный участок данного блока для всех типов FAT идентичен; описание этого участка приведено в табл. 2.3.

Таблица 2.3. Начальный участок загрузочного сектора

Наименование элемента	Смещение	Размер, байт	Описание
BS_jmpBoot	00h	3	Инструкция перехода (jmp) на загрузочный код
BS_OEMName	03h	8	Текстовая строка с аббревиатурой фирмы изготовителя и номером версии операционной системы
BPB_BytsPerSec	0Bh	2	Число байтов в секторе (всегда 512)
BPB_SecPerClus	0Dh	1	Число секторов в кластере
BPB_RsvdSecCnt	0Eh	2	Число резервных секторов в резервной области раздела, начиная с первого сектора раздела
BPB_NumFATs	10h	1	Число таблиц (копий) FAT в разделе (всегда равно 2)
BPB_RootEntCnt	11h	2	Для FAT12 и FAT16 - количество 32-байтных дескрипторов файлов в корневом каталоге (при использовании FAT16 равно 512); для FAT32 это поле имеет значение 0
BPB_TotSecl6	13h	2	Общее число секторов в разделе (если данное поле имеет значение 0, то число секторов задается полем BPB_TotSec32)
BPB_Media	15h	1	Тип носителя информации (см. табл. 2.6)

BPB_FATSzl6	16h	2	Для FAT12 и FAT16 - количество секторов, занимаемых одной копией FAT; для FAT32 поле имеет значение 0
BPB_SecPerTrk	18h	2	Число секторов на дорожке (для прерывания 13h)
BPB_NumHeads	1Ah	2	Число головок (для прерывания 13h)
BPB_HiddSec	1Ch	4	Число скрытых секторов, предшествующих разделу, содержащему данный том. Значение равно 0 для носителей, не подлежащих разбиению на разделы
BPB_TotSec32	20h	4	Общее число секторов в разделе (поле используется вместо BPB_TotSec16, если в разделе свыше 65 535 секторов; в противном случае поле содержит значение 0)

Информацию, приведенную в табл. 2.3, нужно дополнить следующими замечаниями.

- Поле безусловного перехода BS_jmpBoot может иметь два формата, соответствующих двум разным типам инструкций перехода процессора x86. Первый вариант начинается с кода EBh, второй — с кода E9h.
- Поле BS_OEMName, вообще говоря, может содержать любое значение, однако современные операционные системы Microsoft заносят в него код MSWIN4.1.
- Поле BPB_SecPerClus задает число секторов в кластере и может содержать значения, равные 2^N: 1, 2, 4, 8, 16, 32, 64.
- Поле BPB_RsvdSecCnt задает количество секторов резервной области. Для FAT12 и FAT16 данное поле всегда имеет значение 1, а для FAT32 — обычно значение 32.
- Поля BPB_TotSecl6 и BPB_TotSec32 задают число секторов в разделе, причем это число заносится только в одно из указанных полей, а второе должно содержать значение 0. Если в разделе не более 65 535 секторов, то используется поле BPB_TotSecl6, иначе — поле BPB_TotSec32.
- Поле BPB_Media описывает тип установленного в дисковод носителя информации. Перечень допустимых кодов носителей приведен в табл. 2.6.

Различия в структуре загрузочных секторов для разных типов FAT начинаются со смещения 24h. Для FAT12 и FAT16 структура имеет вид, показанный в табл. 2.4, а для FAT32 — в табл. 2.5. Рассмотрим отдельные поля более подробно.

- Поле BS_FilSysType содержит аббревиатуру файловой системы (FAT12, FAT16 или FAT32), но не должно использоваться для идентификации ее типа.
- Бит 7 поля BPB_ExtFlags содержит признак активности FAT: если он равен нулю, то в процессе работы изменения отражаются во всех FAT, а в противном случае активной является только одна копия, номер которой записан в битах 0-3 поля BPB_ExtFlags (счет номеров ведется с 0). Остальные разряды поля BPB_ExtFlags зарезервированы.

Таблица 2.4. Структура загрузочного сектора для FAT12 и FAT16, начиная со смещения 24h

Наименование элемента	Смещение	Размер, байт	Описание
BS_DrvNum	24h	1	Номер дисковода для прерывания 13h (для дисководов гибких дисков - от 0 до 3; для жестких дисков счет номеров ведется с 80h)
BS_Reservedl	25h	1	Зарезервировано для Windows NT. имеет значение 0
BS_BootSig	26h	1	Признак расширенной загрузочной записи (29h). Показывает, что следующие три поля присутствуют
BS_VolID	27h	4	Номер логического диска (формируется при форматировании как комбинация времени и даты создания)
BS_VolLab	2Bh	11	Метка диска (текстовая строка)
BS_FilSysType	36h	8	Текстовая строка с аббревиатурой типа файловой системы

Таблица 2.5. Структура загрузочного сектора для FAT32, начиная со смещения 24h

Наименование элемента	Смещение	Размер, байт	Описание
BPB_FATSz32	24h	4	Количество секторов, занимаемых одной копией FAT
BPB_ExtFlags	28h	2	Номер активной FAT

BPB_FSVer	2Ah	2	Номер версии FAT32: старший байт - номер версии, младший - номер ревизии. В настоящее время используется значение 0:0
BPB_RootClus	2Ch	4	Номер кластера для первого кластера корневого каталога (обычно имеет значение 2)
BPB_FSInfo	30h	2	Номер сектора структуры FSINFO в резервной области логического диска (обычно 1)
BPB_BkBootSec	32h	2	Номер сектора (в резервной области логического диска), используемого для хранения резервной копии загрузочного сектора (обычно 6)
BPB_Reserved	34h	12	Область, зарезервированная для дальнейших расширений (должна содержать нули)
BS_DrvNum	40h	1	Номер дисковода для прерывания 13 h (для дисководов гибких дисков - от 0 до 3; для жестких дисков счет номеров ведется с 80h)
BS_Reservedl	41h	1	Зарезервировано для Windows NT, имеет значение 0
BS_BootSig	42h	1	Признак расширенной загрузочной записи (29h). Показывает, что следующие три поля присутствуют
BS_VolID	43h	4	Номер логического диска (формируется при форматировании как комбинация времени и даты создания)
BS_VolLab	47h	11	Метка диска (текстовая строка)
BS_FilSysType	52h	8	Текстовая строка с аббревиатурой типа файловой системы

Таблица 2.6. Типы носителей информации

Код типа	Тип носителя информации
F0h	Гибкий диск, 2 стороны, 18 секторов на дорожке
F8h	Жесткий диск
F9h	Гибкий диск, 2 стороны, 15 секторов на дорожке
FCh	Гибкий диск, 1 сторона, 9 секторов на дорожке
FDh	Гибкий диск, 2 стороны, 9 секторов на дорожке
FEh	Гибкий диск, 1 сторона, 8 секторов на дорожке
FFh	Гибкий диск, 2 стороны, 8 секторов на дорожке

Кроме перечисленных в табл. 2.3-2.5 полей нулевой сектор логического диска должен содержать в байте со смещением 1FEh код 55h, а в следующем байте (со смещением 1FFh) — код AAh. Указанные два байта являются сигнатурой — признаком загрузочного сектора логического диска. У логических дисков первичных разделов в нулевом секторе находятся так же коды программы загрузчика, на начало которых указывает инструкция перехода (jmp), записанная в первых трех байтах сектора (см. табл. 2.3)

Таким образом, загрузочный сектор выполняет две важные функции: описывает структуру данных на диске, а также позволяет осуществить загрузку операционной системы (если она установлена на диске и диск является активным). В случае возникновения ошибки в момент записи информации в загрузочный сектор (например, из-за внезапного выключения электропитания) может быть утрачена вся информация на логическом диске, поэтому в системе FAT32 сектор 6 резервной области используется для хранения копии загрузочного сектора (Backup Boot Sector).

Поскольку размер таблицы FAT на логическом диске с организацией FAT32 может быть очень велик, для ускорения выполнения операций с FAT была введена структура FSInfo, размещаемая в секторе 1 резервной области. Эта структура содержит информацию о количестве свободных кластеров на диске и о номере первого свободного кластера в таблице FAT. Формат структуры описан в табл. 2.7.

Таблица 2.7. Структура сектора FSInfo и резервного загрузочного сектора FAT32

Наименование элемента	Смещение	Размер, байт	Описание
FSI_LeadSig	000h	4	Значение 41615252h- сигнатура, которая служит признаком того, что данный сектор содержит структуру FSInfo
FSI_Reserved1	004h	480	Поле зарезервировано для дальнейших расширений (все байты должны быть заполнены нулями)
FSI_StrucSig	lE4h	4	Значение 61417272h (сигнатура)

FSI_FreeCount	1E8h	4	Содержит текущее число свободных кластеров на диске. Если в поле записано значение 0FFFFFFFFh,То количество свободных кластеров неизвестно (его нужно вычислять)
FSI_NxtFree	1ECh	4	Содержит номер кластера, с которого дисковый драйвер должен начинать поиск свободных кластеров. Если в поле записано значение 0FFFFFFFFh, то поиск свободных кластеров нужно начинать с кластера №2
FSI_Reserved2	1F0h	12	Поле зарезервировано для дальнейших расширений (все байты должны быть заполнены нулями)
FSI_TrailSig	1FCh	4	Значение AA550000h - сигнатура, которая служит признаком конца структуры FSInfo

Хотя относительные номера секторов структур Backup Boot Sector и FSInfo являются константами, значения этих констант зачем-то хранятся в полях BPB_BkBootSec и BPB_FSInfo загрузочного сектора.

Чтобы прочесть значение некоторого элемента любой структуры данных, нужно вначале получить доступ к самой структуре (то есть определить ее местоположение на диске), а затем прочитать с диска в буфер оперативной памяти либо всю структуру, либо только тот сектор, в котором находится интересующий нас элемент. Местоположение основных структур данных на физическом диске, то есть абсолютные номера начальных секторов, можно вычислить по следующим формулам:

BS_StartSect = LDisk_StartSect

FSInfo_StartSect = BS_StartSect + BPB_FSIfo

FAT1_StartSect = BS_StartSect + RsvdCecCnt

FAT2_StartSect = FAT1_StartSect + BPB_FATSz

Rdir_StartSect = FAT1_StartSect + BPB_FATSz × BPB_NumFATs

Data_ StartSect = Rdir_ StartSect + (32 × BPB_RootEntCnt) / 512

В приведенных формулах выделены переменные, значения которых необходимо извлечь из структур данных, расположенных в загрузочном секторе логического диска, и использованы следующие обозначения:

- BS_Start Sect — номер загрузочного сектора логического диска;
- LDisk_StartSect — номер начального сектора диска;
- FATl_StartSect — номер начального сектора первой FAT;
- BPB_RsvdSecCnt — число секторов в резервной области (32 для системы FAT32 и 1 для остальных систем);
- FAT2_StartSect — номер начального сектора второй FAT;
- BPB_FATSz - размер одной копии FAT (BPB_FATSz32 для FAT32 и BPB_FATSzl6 для остальных систем);
- BPB_NumFATs — число копий FAT на диске;
- RDir_StartSect — номер начального сектора корневого каталога;
- Data_StartSect— номер начального сектора области данных;
- BPB_RootEntCnt — число записей в корневом каталоге (0 для FAT32 и 512 для остальных систем).

В разделе 4.7 приведен листинг программы ShowFDDSector, позволяющей при помощи функций BIOS осуществлять последовательный просмотр секторов гибкого диска в ASCII-кодах. Она считывает загрузочный сектор и извлекает из него информацию о формате гибкого диска (то есть о количестве рабочих поверхностей диска и числе секторов на дорожке). Эта программа достаточно универсальна и ее можно запускать на любом персональном компьютере - даже на древних машинах типа XT с пятидюймовыми дисководами.

2.2.2. Назначение и внутренняя организация таблиц размещения файлов

Размер файла, вообще говоря, не является постоянной величиной и может изменяться (обычно — увеличиваться) со временем. Если допускается хранение файла только в смежных (последовательно расположенных) секторах, то при увеличении размера файла операционная система должна полностью перезаписывать его в другую (свободную) область диска подходящего объема. Чтобы упростить и ускорить выполнение операции добавления новых данных в файл, в современных операционных системах применяются таблицы размещения файлов (File Allocation Table, сокращенно FAT), позволяющие хранить файл в виде нескольких несмежных участков.

При использовании FAT область данных логического диска разделена на участки одинакового размера — кластеры. Кластер может состоять из одного или нескольких последовательно

расположенных на диске секторов. Число секторов в кластере должно быть кратно 2^N и может принимать значения от 1 до 64 (размер кластера зависит от типа используемой системы FAT и объема логического диска).

Каждому кластеру поставлен в соответствие собственный элемент таблицы FAT. Первые два элемента FAT являются резервными — если на диске имеется CountOfClusters (счетчик кластеров данных), то число элементов таблицы будет равно CountOfClusters + 2. Тип FAT определяется значением CountOfClusters:

- если CountOfClusters < 4085, то используется система FAT 12;
- если 4084 < CountOfClusters < 65525, то используется система FAT16;
- если 65524 < CountOfClusters — используется система FAT32.

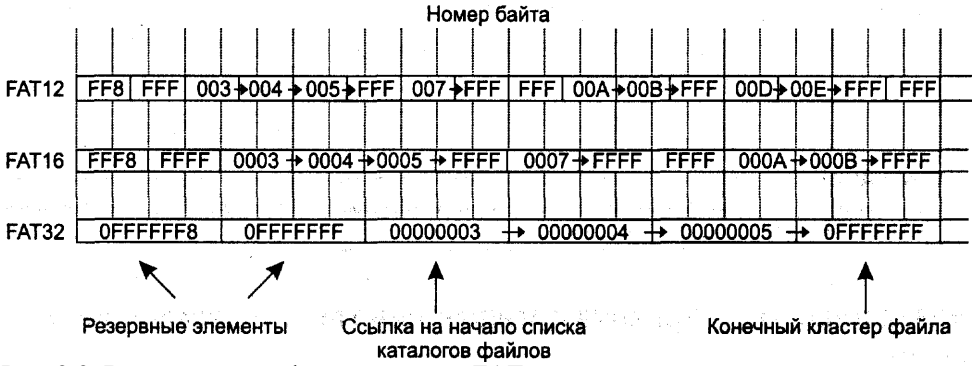

Рис. 2.3. Вид начальных фрагментов для FAT различных типов

Названия типов FAT ведут свое происхождение от размера элемента: элемент FAT12 имеет размер 12 бит (1,5 байта), FAT16 — 16 бит (2 байта), FAT32 — 32 бита (4 байта). Следует учитывать, что в FAT32 четыре старших двоичных разряда зарезервированы и игнорируются в процессе работы операционной системы (то есть значащими являются только семь младших шестнадцатеричных разрядов элемента).

Каждому файлу, находящемуся в области данных диска, соответствует цепочка элементов FAT — упорядоченный однонаправленный список (рис. 2.3). Поскольку список однонаправленный, он обеспечивает только движение вперед — если вы захотите вернуться к предыдущему кластеру, вам придется снова проводить поиск с самого начала списка.

В каталоге файлов для каждого файла приводится номер начального элемента в таблице FAT, соответствующий первому кластеру файла. Если файл содержит более оного кластера, то указанный элемент содержит номер элемента FAT, соответствующего следующему кластеру файла. Последний элемент списка содержит признак конца файла (End Of Clasterchain, сокращенно EOC), который в FAT12 может принимать значение от FF8h до FFFh, в FAT16 — значение FFF8h-FFFFh, в FAT32 — значение FFFFFF8h-FFFFFFFh (табл. 2.8). Операционные системы Microsoft всегда применяют для EOC значения FFFh, FFFFh и FFFFFFFh соответственно, однако дисковые утилиты других фирм могут использовать и иные допустимые значения.

Таблица 2.8. Значения специальных кодов элементов FAT

Значение кода	FAT12	FAT16	FAT32
Свободный кластер	0	0	0
Номер следующего элемента в цепочке	002 - FEFh	0002 - FFEFh	0000002 - FFFFFEF
Дефектный кластер	FF7h	FFF7h	FFFFFF7h
Последний кластер в списке	FF8h-FFFh	FFF8h-FFFFh	FFFFFF8h-FFFFFFFh

Кроме признака EOC определены также специальные коды для свободного кластера (имеет значение 0 во всех типах FAT) и дефектного кластера (имеет значение FF7h в FAT12, значение FFF7h в FAT16 и значение FFFFFF7h в FAT32).

В документации Microsoft для элементов FAT используется принятое в языке программирования C обозначение элементов массивов. Значение, хранящееся в элементе FAT[0] (первом резервном элементе таблицы FAT) является сигнатурой. Для FAT12 оно всегда равно FF8h, для FAT16- FFF8h, для FAT32 - FFFFFF8h.

В FAT [1] (то есть второй резервный элемент) при форматировании диска записывается код EOC. Кроме того, системы FAT16 и FAT32 могут использовать два старших значащих разряда указанного элемента в качестве флагов. Флаг CInShutBitMask занимает в системе FAT16 двоичный разряд 15, а в

системе FAT32 разряд 27. Если флаг ClnShutBitMask установлен в 1, то логический диск (том) является «чистым» (clean), если сброшен в 0 — «грязным» (dirty). Термин «грязный» означает, что работа с диском не была завершена надлежащим образом (например, по причине внезапного отключения электропитания) и при загрузке операционной системы должна быть выполнена процедура восстановления диска.

Флаг HrdErrBitMask служит признаком наличия сбоев при выполнении операций ввода/вывода. В системе FAT16 он занимает двоичный разряд 14, а в системе FAT32 разряд 26. При загрузке операционной системы (точнее, в момент монтирования тома) HrdErrBitMask устанавливается в 1, но в случае возникновения сбоя при записи или считывании информации флаг сбрасывается в 0.

Выполнять операции с элементами таблиц типов FAT16 и FAT32 очень легко, поскольку эти таблицы представляют собой массивы 16-разрядных и 32-разрядных слов соответственно. Работать с элементами FAT12 менее удобно, так как для доступа к элементу массива приходится выполнять ряд вспомогательных действий. При извлечении элемента из FAT12 порядок действий следующий:

1. Умножить номер элемента на 3.
2. Разделить результат на 2.
3. Извлечь из FAT 16-разрядное слово, используя в качестве адреса результат предыдущей операции.
4. Если номер элемента четный, выполнить операцию AND над считанным словом и маской 0FFFh. Если номер нечетный, сдвинуть считанное слово вправо на 4 разряда. В результате получаем искомое значение элемента FAT.

При записи элемента в FAT 12 порядок действий таков:

1. Умножить номер элемента на 3.
2. Разделить результат на 2.
3. Извлечь из FAT 16-разрядное слово, используя в качестве адреса результат предыдущей операции (адрес слова запомнить).
4. Если номер элемента четный, выполнить операцию AND над считанным словом и маской 0F000h, а затем операцию OR над полученным результатом и значением записываемого элемента. Если номер нечетный, выполнить операцию AND над считанным словом и маской 0F000h, а затем сдвинуть значение элемента влево на 4 разряда и выполнить OR с результатом предыдущей операции.
5. Записать полученное 16-разрядное слово обратно в FAT.

В программах, написанных на ассемблере, для выполнения умножения на 3 вместо команды MUL часто применяется алгоритм «сдвиг и сложение»: исходное число копируется, над копией числа выполняется сдвиг влево на один разряд (умножение на 2), а затем оба числа складываются (x + 2x = 3x). Вместо команды DIV при делении на 2 используется сдвиг вправо на один разряд.

Элемент FAT содержит номер кластера, но при работе с дисками на низком уровне адресуемой единицей данных является сектор, а не кластер. Номер начального сектора кластера SectorNum связан с номером кластера ClusterNum следующей формулой:

$$SectorNum=Data_StartSect+(ClusterNum-2)\times BPB_SecPerClus$$

SectorNum = StartSect + (ClasterNum-2) × BPB_SecPerClas

В документации Microsoft приводятся также следующие примечания относительно FAT.

- Данные в последнем секторе FAT не обязательно полностью принадлежат FAT - конец сектора, не относящийся к FAT, может содержать произвольные данные. Последний элемент FAT имеет номер, равный CountOfClusters + 1 (как сказано выше, нумерация ведется с нуля и в начало таблицы добавлены два резервных элемента).
- Последний сектор FAT должен вычисляться по значению CountOfClusters + 1. Использовать с этой целью значения BPB_FATSzl6 и BPB_FATSz32 нельзя.

2.3.Каталоги файлов

Каталог файлов представляет собой массив 32-байтных элементов — описателей файлов. С точки зрения операционной системы все каталоги (кроме корневого каталога в системах FAT 12 и FAT 16) выглядят как файлы и могут содержать произвольное количество записей.

Корневой каталог (Root Directory) — это главный каталог диска, с которого начинается дерево подкаталогов. Для корневого каталога в FAT12 и FAT16 выделено в системной области логического диска специальное место фиксированного размера (16 Кбайт), рассчитанное на хранение 512 элементов. В системе FAT32 корневой каталог является файлом произвольного размера.

Структура элемента каталога файлов приведена в табл. 2.9. Элемент начинается с 11-байтного поля DIR_Name, содержащего так называемое короткое имя (Short name) файла, по которому операционная система обычно осуществляет поиск файла в каталоге. Короткое имя состоит из двух полей: 8-байтного

поля, содержащего собственно имя файла, и 3-байтного поля, содержащего расширение. Если введенное пользователем имя файла короче восьми символов, то оно дополняется пробелами (код пробела — 20h); если введенное расширение короче трех байт, то оно также дополняется пробелами. Разделительная точка между именем и расширением файла не хранится в структуре данных; она подставляется программами операционной системы после имени файла только при выполнении операций, требующих взаимодействия с пользователем (задание имени файла, вывод списка файлов на экран и т. д.). Кроме того, следует учитывать, что в коротком имени все текстовые символы преобразуются операционной системой в верхний регистр.

Таблица 2.9. Структура элемента каталога

Наименование поля	Смещение	Размер, байт	Описание
DIR_Name	00h	11	Короткое имя файла
DIR_Attr	0Bh	1	Атрибуты файла
DIR_NTRes[1]	0Ch	1	Поле зарезервировано для Windows NT (должно содержать значение 0)
DIR_CrtTimeTenth[1]	0Dh	1	Поле, уточняющее время создания файла (содержит десятки миллисекунд). Значение поля может находиться в пределах от 0 до 199
DIR_CrtTime[1]	0Eh	2	Время создания файла
DIR_CrtDate[1]	10h	2	Дата создания файла
DIR_LstAccDate[1]	12h	2	Дата последнего обращения к файлу для записи или считывания данных
DIR_FstClasHI[1]	14h	2	Старшее слово номера первого кластера файла
DIR_WrtTime	16h	2	Время выполнения последней операции записи в файл
DIR_WrtDate	18h	2	Дата выполнения последней операции записи в файл
DIR_FstClusLO	1Ah	2	Младшее слово номера первого кластера файла
DIR_FileSize	1Ch	4	Размер файла в байтах (32-разрядное число)

[1] Поле обрабатывается только в файловой системе FAT32. В системах FAT12 и FAT16 поле считается зарезервированным и содержит значение 0.

Первый байт короткого имени (DIR_Name[0]) выполняет функции признака занятости элемента каталога:

- если DIR_Name [0] = E5h, то элемент каталога свободен (то есть его можно использовать при создании нового файла);
- если DIR_Name[0] = 00h, то элемент каталога свободен и является началом чистой области каталога (после него нет ни одного использованного элемента);
- если DIR_Name [0] = 05h, то следует считать, что в этом байте находится ASCII-символ с кодом 0E5h (символ KANJI японского алфавита).

На использование ASCII-символов в коротком имени накладывается ряд ограничений:

- нельзя использовать символы с кодами меньше 20h (за исключением кода 05h в DIR_Name[0]);
- нельзя использовать символы с кодами 22h, 2Ah, 2Bh, 2Ch, 2Eh, 2Fh, 3Ah, 3Bh, 3Ch, 3Dh, 3Eh, 3Fh, 5Bh, 5Ch, 5Dh, 7Ch;
- нельзя использовать символ пробела (код 20h) в DIRName [0].

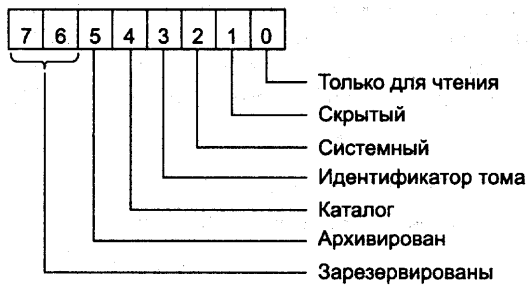

Рис. 2.4. Формат байта атрибутов файла

При задании пользователем имени файла допускается отсутствие расширения, но имя должно

содержать по крайней мере один символ.

Формат байта атрибутов файла DIR_Attr показан на рис. 2.4. Разряды байта атрибутов устанавливаются в 1 в том случае, если у файла имеется соответствующее свойство:

- бит 0 — файл только для чтения;
- бит 1 — скрытый файл;
- бит 2 — системный файл;
- бит 3 — идентификатор тома;
- бит 4 — каталог;
- бит 5 — архивированный файл;
- биты 6 и 7 — зарезервированы, должны быть установлены в 0.

Признаком того, что свободный элемент каталога используется для хранения участка длинного имени файла, является наличие единиц в разрядах 0-3 байта атрибутов (для описателей файлов и каталогов такое сочетание невозможно).

Поле времени создания файла DIR_CrtTime и поле времени выполнения последней операции записи в файл DIR_WrtTime имеют формат, показанный на рис. 2.5. Назначение разрядов полей времени следующее:

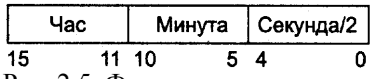

Рис. 2.5. Формат поля времени

- биты 0-4 — двухсекундный отсчет (допустимо значение от 0 до 29);
- биты 5-10 — минута (допустимо значение от 0 до 59);
- биты 11-15 — час (допустимо значение от 0 до 23).

При создании файлов отсчет дат ведется от начала эпохи MS-DOS, то есть от 01.01.1980. Поля даты создания файла DIR_CrtDate, даты последнего обращения к файлу DIR_LstAccDate и даты выполнения последней операции записи в файл DIR_WrtDate имеют формат, показанный на рис. 2.6. Назначение разрядов указанных полей:

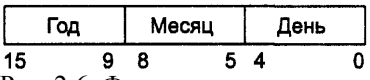

Рис. 2.6. Формат поля даты

- биты 0-4 — день месяца (допустимо значение от 1 до 31);
- биты 5-8 — номер месяца в году (допустимо значение от 1 до 12);
- биты 9-15 — номер года минус 1980 (допустимо значение от 0 до 127).

Номер первого кластера файла (и точка входа в таблицу FAT) задаются полями DIR_FstClusLO и DIR_FstClusHI, содержащими младшее слово и старшее слово номера кластера соответственно. Для FAT12 и FAT16 определено только поле DIR_FstClusLO, а DIR_FstClusHI является зарезервированным и должно содержать нулевое значение.

Поле размера файла DIR_FileSize является 32-разрядным числом, что позволяет задавать размер файла до 4 Гбайт (4 294 967 295 байт). Хотя каталоги по сути также являются файлами, значение этого поля для них устанавливается в ноль. Ограничение на размер каталога с этим полем никак не связано и вызвано тем, что дисковые утилиты операционной системы используют в качестве счетчика элементов каталога 16-разрядное слово (каталог может содержать до 65 536 32-байтных элементов и занимает, таким образом, не более 2 097 152 байт).

Новые версии операционной системы Windows (начиная с Windows 95) позволяют присваивать файлу (в дополнение к короткому имени) так называемое длинное имя (Long name), используя для его хранения пустые элементы каталога, смежные с основным элементом — описателем файла. Короткое и длинное имена файла являются уникальными, то есть не должны встречаться дважды в одном каталоге.

Длинное имя записывается не ASCII-символами, а в формате Unicode, где каждому национальному алфавиту соответствует собственный набор кодов. Расплатой за универсальность Unicode является снижение плотности хранения информации — каждый символ занимает два байта (16-разрядное слово). В пустые элементы каталога длинное имя записывается в разрезанном на кусочки виде, как показано в табл. 2.10.

В одном элементе каталога можно сохранить фрагмент длиной до 13 символов Unicode (поскольку в трех участках имеется в сумме 26 байт). Неиспользованный участок последнего фрагмента заполняется

кодами FFFFh.

Таблица 2.10. Структура элемента каталога, хранящего фрагмент длинного имени файла

Наименование поля	Смещение	Размер, байт	Описание
DIR_Counter	00h	1	Номер фрагмента
DIR_Lnamel	0lh	10	Первый участок фрагмента имени
DIR_Attr	0Bh	1	Атрибуты файла
DIR_Flags	0Ch	1	Байт флагов
DIR_ChkSum	0Dh	1	Контрольная сумма короткого имени
DIR_LName2	0Eh	12	Второй участок фрагмента имени
DIR_First	lAh	2	Номер первого кластера (должен быть равен 0)
DIR_Lname3	1Ch	4	Третий участок фрагмента имени

Длинное имя записывается в каталог первым, причем фрагменты размещены в обратном порядке, начиная с последнего — как показано на рис. 2.7. Вслед за длинным (полным) именем размещается стандартный описатель файла, содержащий укороченный по специальному алгоритму вариант этого имени. Фрагменты длинного имени пронумерованы. В младших разрядах байта номера фрагмента хранится собственно номер, разряд 6, вероятно, служит признаком последнего фрагмента длинного имени (имеет значение 1 для последнего фрагмента, у остальных фрагментов — 0), а разряд 7 зарезервирован (равен 0).

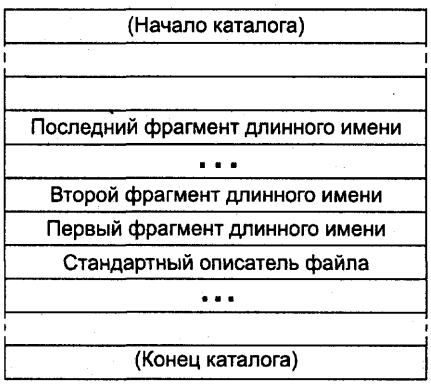

Рис. 2.7. Порядок записи в каталог описателя файла с длинным именем

Все каталоги, за исключением корневого, содержат в двух первых элементах вместо описателей файлов специальные ссылки. В элементе с номером 0 размещается указатель на сам каталог, а в поле имени находится одна точка («.»). В элементе с номером 1 размещается указатель на родительский каталог (каталог более высокого уровня), а в поле имени находится две точки («. .»). Если ссылка на таблицу FAT у элемента № 1 имеет нулевое значение, то текущий каталог находится в корневом каталоге.

2.4. Возможности системы FAT

Как видно, файловая система MS DOS и Windows 9x, построенная на основе FAT, довольно проста. Ограничения на объем диска, размеры и количество файлов с Переходом от FAT 12 через FAT 16 к FAT32 далеко отодвинулись от реальных потребностей обозримого будущего. Но на больших объемах дисков, а главное, на больших количествах файлов в каталогах простота системы приводит к низкой производительности. Дело в том, что поиск файла в каталоге ведется перебором элементов каталога до совпадения с искомым (или до конца каталога) — файлы в каталоге никак не упорядочены. Получается, что для того, чтобы найти файл в каталоге, в среднем необходимо просмотреть половину записей данного каталога. Правда, записи каталогов весьма компактные (по сравнению с другими системами), а для современных ОС разработаны еще и программные методы ускорения этих процедур (хэширование). Однако при количестве записей в каталогах, исчисляемом сотнями и тысячами, да еще и при большом объеме раздела с малым размером кластера работа с файлами, особенно фрагментированными, сильно замедляется.

Файловые системы FAT не имеют никаких средств защиты данных от несанкционированного доступа (атрибуты серьезной защитой не назовешь). Критичный компонент — FAT, от которого зависит возможность восстановления файлов в случае аварии, защищен дублированием. Правда, копии FAT, сколько бы их ни было, расположены друг за другом, так что при физическом повреждении поверхности

диска они могут потеряться и все.

Файловая система FAT32 отличается от FAT 16 не только размером элементов таблицы и допустимым размером логического диска. Загрузочная запись значительно расширена, в ней имеются указатели на резервные копии критических элементов диска. Здесь корневой каталог представляется, как и обычный, — цепочкой кластеров произвольной длины, так что число вхождений в него явно не ограничено. Корневой каталог может быть перемещен, возможно также переключение между используемыми копиями таблицы размещения файлов. В системе предполагается возможность динамического изменения размера диска, но пока что эта возможность в ОС еще не реализована. Систему FAT32 «понимает» только Windows 95 OSR2, Windows 98, ME и Windows 2000; дисковые утилиты Format, FDISK, Defrag и ScanDisk, входящие в их комплект, должным образом модифицированы. Том с FAT32 не может быть сжат компрессорами Microsoft DriveSpace или DriveSpace 3. Система Windows NT не поддерживает FAT32.

Отдельно отметим соглашения об именах файлов. В файловой системе FAT имена в формате 8.3 не делается различий между символами, набранными в разных регистрах. Для Windows 9x расширенные имена позволяют использовать и отображать любой регистр символов, но файлы не считаются разными, если символы в их именах отличаются только регистром. «Имя MS DOS» строится из длинного имени без учета регистра. С одной стороны, это удобно, с другой — может приводить к проблемам при переносе файлов на диски с иной файловой системой, где имеется чувствительность к регистру (case sensitivity). Тут-то и проявятся «мягкие» ошибки, когда ссылка на файл описана в одном регистре, а само имя — в другом. С этой проблемой часто сталкиваются при переносе Web-сайтов, созданных в среде Windows, на платформу, чувствительную к регистру (например, UNIX). Неприятностей добавляют и некоторые приложения Windows, которые «подправляют» имена файлов по своему вкусу (например, первая буква прописная, а за ней строчные).

Конечно же, есть и другие файловые системы, более сложные, защищенные и эффективные на больших объемах дисков и при большом количестве файлов. К ним можно отнести HPFS (OS/2), NTFS (Windows NT), Nowell NetWare, Unix, Linux и некоторые другие. Они построены иначе, используют другие механизмы распределения дискового пространства, но здесь мы их рассматривать не будем. Отметим лишь, что за полезные свойства часто приходится расплачиваться большими системными ресурсами (особенно оперативной памятью). Существуют средства для взаимных преобразований файловых систем как в составе ОС, так и от сторонних производителей.

ВНИМАНИЕ

Преобразование файловых систем в сторону усложнения (например, FAT в NTFS) не всегда дает оптимальную структуру новых дисков, и с «более эффективной файловой системой» работа может оказаться медленней (особенно, если мало оперативной памяти и «тихоходный» диск).

В процессе конверсии возможны ошибки с фатальным исходом. Перед выполнением преобразований скопируйте, по крайней мере, важные данные на внешний носитель.

Возможность работы ОС с диском, созданным не ее средствами, зависит от типа файловой системы, типа раздела, на котором он расположен, и размера раздела. Так, например, диск с файловой системой FAT 16 в Windows NT поддерживается на разделах типа 04, 05 и 06, но не поддерживается на разделах типов 0E и 0F, а FAT32 не поддерживается ни на каких.

2.5. Сжатие дисков

Для повышения эффективности использования дискового пространства хранимые файлы можно подвергать сжатию (компрессии) — обработке по специальному алгоритму, уменьшающему длину файла без потери информации. Соответственно, должен быть обратный алгоритм декомпрессии, абсолютно точно восстанавливающий исходный файл из сжатого. Существуют различные методы, различающиеся достижимой степенью сжатия и необходимыми для этого вычислительными ресурсами (объем используемой оперативной памяти и количество команд процессора). По общим соображениям очевидно, что большее сжатие требует большего процессорного времени. Достижимая степень сжатия зависит и от характера данных — некоторые файлы при попытке сжатия даже увеличиваются в объеме. Хорошо, например, сжимаются текстовые файлы (просто символьные или файлы MS WORD). Практически не сжимаются файлы типа .pdf (формат Adobe Acrobat), даже если у них внутри просто текст (их объем при этом во много раз превышает число символов). С ростом мощности компьютеров становились приемлемыми алгоритмы, обеспечивающие все большую степень сжатия. Так, первые архиваторы для файлов ARC, PKXARC и PACK сменились более мощными PKZIP, ARJ, RAR и другими. Хранение файлов в виде упакованных архивов дает экономию места на диске сразу по двум статьям: во-первых, уменьшается длина файла (число байтов). Во-вторых, если архивируется группа файлов, то уменьшаются потери на недоиспользованных кластерах (в среднем по половине кластера на файл). Современные архиваторы совместно с операционной системой позволяют работать (по крайней мере в режиме чтения) с файлами прямо из архива — выбранный файл распаковывается в каталог для

временного хранения и оттуда передается требуемому приложению. Для этого достаточно открыть архив архиватором (например, WinZip) и щелкнуть мышкой по требуемому файлу (с известным расширением).

Кроме сжатия отдельных файлов применяют и дисковые компрессоры — программные средства, сжимающие данные на диске «прозрачно» для приложений (и пользователя). Каждый раз при записи файла (или его фрагмента) выполняется компрессия, а при чтении — декомпрессия. Конечно, для исполнения в реальном времени пригодны не всякие алгоритмы компрессии, и ради экономии времени жертвуют достижимой степенью сжатия. Возможность сжатия заложена в такие сложные файловые системы, как, например, Novell NetWare (начиная с версий 4.x) и NTFS. Для файловой системы FAT (MS DOS и Windows 9x) встроенных компрессоров не предусмотрено, но с ними широко используются загружаемые компрессоры типа Stacker, DoubleSpace и DriveSpace. Компрессор DoubleSpace был введен в комплект поставки MS DOS версии 6, впоследствии в результате судебных разбирательств в него внесли некоторые изменения и он был переименован в DriveSpace. Идея этих компрессоров заключается в следующем. На обычном логическом диске, называемом *несущим,* размещается большой файл-образ сжатого диска CVF (Compressed Volume File). В этом образе есть своя система каталогов и таблица размещения файлов. Во время загрузки (до исполнения команд файла CONFIG.SYS) в оперативную память помещается специальный резидентный драйвер из файла DBLSPACE.BIN (или DRVSPACE.BIN), находящегося в корневом каталоге загрузочного диска. Этот драйвер эмулирует обращения к реальному диску операциями доступа к файлу-образу, на ходу осуществляя компрессию/декомпрессию. Для ОС эмулируемый диск выглядит, как обычный логический диск, и для удобства пользователя ему может назначаться логическое имя (буква), ранее принадлежащее несущему диску. Несущий диск при этом получает новое (ранее неиспользованное) имя, и к нему, в принципе, тоже можно обращаться обычным способом. Несущий диск можно и скрыть от приложений и пользователя (чтобы не было поползновений удалить «никому не нужный» громадный файл-образ). На несущем диске должны оставаться файлы, необходимые для загрузки ОС (для DOS это IO.SYS и MSDOS.SYS), драйвер DBLSPACE.BIN или DRVSPACE.BIN и собственно файл-образ. На одном несущем диске может размещаться и несколько файлов-образов, каждый из которых будет представлять свой сжатый логический диск (том). К компрессору относится еще и небольшой файл DBLSPACE.SYS (DRVSPACE.SYS), который, будучи загруженным из файла CONFIG.SYS командой DEVICEHIGH=DBLSPACE.SYS (или DEVICEHIGH=DRVSPACE.SYS), находит программный код DBLSPACE.BIN (DRVSPACE.BIN) и по возможности перемещает его из стандартной памяти в верхнюю память, освобождая дефицитное место в стандартной памяти. Кроме того, имеется и утилита обслуживания сжатых дисков DBLSPACE.EXE (DRVSPACE.EXE), с помощью которой создаются тома сжатых дисков и выполняются служебные операции. Утилита позволяет преобразовать «населенный» диск в сжатый, сжимая файлы и перемещая их в образ сжатого диска. Можно и не трогать существующие файлы, а создать новый сжатый диск. на свободном месте несущего. При желании (и наличии достаточного свободного места) можно выполнить и обратное перемещение («разжать» диск). Утилита создает в корневом каталоге диска С: файл DBLSPACE.INI (DRVSPACE.INI), в котором в текстовом виде описана конфигурация компрессора и перечислены списки активируемых сжатых томов. Например, строка AutoMount=l означает, что сжатые диски должны монтироваться автоматически, а ActivatedDrive=K,Il указывает, что сжатый диск К: будет эмулироваться файлом-образом DBLSPACE.001 на несущем диске I:. Утилита выполняет и форматирование сжатого диска — «зачистку» файла-образа, приводящую к потере всех данных.

Сжимать можно как жесткие диски, так и дискеты, но целесообразность сжатия дискет сомнительна по ряду причин (с дискетами удобнее пользоваться файловыми компрессорами).

Применение дискового компрессора имеет свои плюсы и минусы. С одной стороны, на диск в среднем удается поместить данных в 1,5-2 раза больше, чем на несжатый диск. При этом скорость обращения к файлам может даже и не упасть (при мощном процессоре), поскольку реальный объем обменов данными с физическим диском несколько сокращается. Однако компрессор является дополнительным звеном, повышающим уязвимость данных в случае сбоев, аварий и вирусных атак. Пользователям сжатых дисков особенно рекомендуется регулярно архивировать важные данные; на внешних носителях. Поскольку файл-образ на несущем диске расположен в единой цепочке кластеров (не фрагментирован, о чем заботится утилита установки сжатого диска), файлы, хранящиеся на сжатом диске, оказываются потенциально спасаемыми при такой аварии несущего диска, как потеря FAT. Используя сжатые диски, довольно просто переносить сложные структуры файлов и каталогов — достаточно скопировать один сжатый файл и подключить его в качестве сжатого диска там, где надо. При использовании сжатого диска объем свободного пространства точно не известен, поскольку компрессор ориентируется на ожидаемую степень сжатия, а реальная может заметно отличаться от нее в любую сторону. Файлы-архивы (ZIP, ARJ, RAR и т. п.), а также файлы со сложными данными (например, PDF) дисковым компрессором уже не сожмутся.

В Windows 95 утилита DriveSpace из стандартной поставки имеет ограничение на размер сжимаемого диска в ,500 Мбайт. При желании использовать компрессор можно поставить более мощный из комплекта MSPlus!, не имеющий этого ограничения (такой же компрессор входит и в состав Windows 98).

Контрольные вопросы.

1. Что понимается под разделами диска и логическими дисками?
2. Где размещается таблица разделов, что в ней содержится, какова структура описателя раздела?
3. Как осуществляется загрузка активного раздела?
4. Чем определяется структура раздела?
5. Какие разделы являются первичными разделами DOS/Windows9x?
6. Где хранится тип файловой системы и таблицы FAT для разделов DOS/Windows9x&
7. Какой раздел DOS/Windows9x может быть активным?
8. Для чего используют расширенный раздел, что содержится в его первом секторе?
9. Как распределяются имена логических дисков, расположенных на двух физических дисках, подключенных к первичному контроллеру ATA?
10. Из каких областей состоит логический диск с файловой системой типа FAT?
11. Что находится в первом секторе логического диска FAT?
12. Для чего используется сектор FSInfo и какова его структура?
13. Как определяется местонахождение структур на диске?
14. Что означает число, стоящее после аббревиатуры "FAT"?
15. Каково назначение и внутренняя организация таблиц размещения файлов?
16. Что такое кластер?
17. Как связан тип FAT со значением счетчика кластеров (CountOfClusters)?
18. Каковы значения специальных кодов элементов FAT?
19. Что такое "Грязный" (dirty) диск и как он метится?
20. Как производится извлечение и запись элементов FAT12?
21. Как связан номер начального сектора кластера и номер кластера?
22. Что представляет собой каталог файлов и какой размер имеет корневой каталог в FAT12, FAT16 и FAT32?
23. Опишите структуру элемента каталога файлов с короткими и длинными именами.
24. Каковы требования к короткому имени файла и его первому байту?
25. Какие атрибуты могут иметь файлы?
26. Как ведется отсчет времени и даты создания и корректировки файлов?
27. Чему равен максимальный размер файла и каталога?
28. Опишите порядок записи в каталог описателя файла с длинным именем.
29. Какова структура первых двух элементов некорневых каталогов?
30. Какие ОС поддерживают FAT32?
31. Какова идея работы компрессоров DoubleSpace и DriveSpace?

В данном разделе использованы материалы из [2, 3, 4]. В этих источниках в той или иной мере освещены разные аспекты логической организации дисков.

3. Системная поддержка дисковой памяти

Дисковая память имеет стандартную поддержку на уровне BIOS и операционной системы. Поддержка дисков со стороны BIOS заключается в предоставлении вышестоящим уровням программного обеспечения возможности чтения и записи секторов диска, форматирования трека и выполнения вспомогательных функций. Эти возможности предоставляются программным вызовом прерывания BIOS *INT 13h* — дискового сервиса. Дисковый сервис BIOS предназначен для изоляции вышестоящего ПО (ОС и приложений) от подробностей реализации дисковой системы.

Дисковый сервис *INT 13h* работает на уровне *физических устройств,* называемых также *физическими приводами* (physical drive). Традиционный дисковый сервис имеет программный интерфейс, сохранившийся еще со времен IBM PC/XT. Он позволяет оперировать с дисками размером до 528 Мбайт (см. п. 5.1.2.). Когда появились диски большего размера, в традиционный дисковый сервис ввели расширение (алгоритмы преобразования), позволяющее преодолеть этот барьер и, теоретически, работать с дисками объемом до 8,4 Гбайт. Однако на практике по пути к объему 8,4 Гбайт (уже не кажущемуся большим) пришлось преодолевать еще несколько барьеров, вызванных просчетами разработчиков ОС и BIOS. Для работы с большими дисками введены новые функции BIOS *INT 13h с* интерфейсом, совершенно не похожим на традиционный.

Операционная система предоставляет прикладным программам сервисы более сложные, чем функции BIOS, обеспечивая доступ к файловой системе диска. Самый низкий уровень обращения, допускаемый операционной системой, оперирует с секторами *логических дисков,* или приводов (logical drive), ассоциированными с именем устройств (A:, B:, C:,..., Z:). Один *физический диск* (жесткий) может содержать несколько логических дисков (см. п. 4.2). На физическом диске могут присутствовать системные области (MBR и резервированные секторы, см. п. 4.1), доступ к которым операционная система не предоставляет.

Для работы утилит и приложений с дисковой памятью имеется несколько уровней сервиса:

- Обращение к контроллеру дисков на уровне регистров, задавая адрес блока и количество секторов, требуемых для передачи. Это самый производительный способ обмена данными, но он требует знания как организации разделов диска и файловой системы, так и контроллера диска (его системы команд). Доступ обеспечивается ко всем элементам, кроме блоков, закрытых командой *Set Max Address* (для дисков ATA). При этом обходится фаза трансляции геометрии, выполняемая, в общем, по усмотрению BIOS.
- Использование сервисов BIOS INT 13h также дает неограниченный доступ к диску, но если используются традиционные вызовы (функции с номерами ниже 40h), то доступны лишь диски объемом до 8,4 Гбайт (при трансляции геометрии) или до 528 Мбайт (без трансляции). За абстрагирование от системы команд контроллера приходится расплачиваться некоторым снижением производительности, связанным с накладными расходами программного интерфейса.
- Используя интерфейс функций DOS, можно получить доступ к любому сектору логического диска, указав лишь его логическое имя, по функциям чтения/записи абсолютных секторов. Но при этом первым доступным сектором (логический адрес 0) является загрузочный сектор логического диска, а секторы с таблицами разделов и те, что не попадают в разделы логических дисков, оказываются недоступны. Для осмысленного доступа к данным необходимо знать структуры FAT и каталогов, и риск «наломать дров», вплоть до полной потери данных, довольно велик.
- И, наконец, интерфейс файловой системы ОС позволяет довольно легко создавать, искать и удалять файлы и каталоги, выполнять чтение и запись файлов целиком или частично и выполнять некоторые другие вспомогательные функции. При этом от пользователя данного интерфейса скрываются все тонкости работы с FAT и элементами каталогов (и, конечно же, все премудрости нижних уровней).

Прикладные программы и утилиты изолируют пользователя от всех этих тонкостей, что позволяет ему сконцентрировать внимание на решении своих смысловых задач. Уровень, на котором программа обращается к дискам, выбирается программистом из соображений минимальной достаточности: чем выше уровень сервиса, тем легче достичь совместимости с остальным ПО (предоставляется меньше возможностей для совершения ошибок). Чем ниже уровень сервиса, тем больше знаний нужно вложить в программу, чтобы она общалась с диском не наперекор общим правилам.

3.1. Сервисы BIOS

Функции дискового сервиса вызываются программным прерыванием *INT 13h.* При этом требуется задать номер функции, логический адрес устройства (не путать с именем логического диска!), адрес сектора и число передаваемых секторов и указать, в каком месте оперативной памяти находится буфер для обмена данными с диском. Все хлопоты по взаимодействию с контроллером требуемого диска берет на себя BIOS. Драйверы традиционных устройств — НГМД и винчестеров с интерфейсом ATA — находятся в системной BIOS. Для других устройств, в том числе и винчестеров с интерфейсом SCSI,

используются специальные собственные драйверы. Для дисков SCSI они располагаются в модуле дополнительной BIOS, находящемся на карте контроллера SCSI (или в системной BIOS, если контроллер SCSI расположен на системной плате), или в загружаемых программных модулях.

Исторически сложилось так, что в дисковом сервисе используется адресация непосредственно к цилиндрам, головкам и секторам устройства — CHS, хотя в большинстве случаев для программ более удобной была бы линейная адресация LBA. Однако, как известно, в случае дисков с интерфейсом ATA к реальной физике накопителей координаты сектора, передаваемые по интерфейсу, имеют только косвенное отношение. Для дисков с интерфейсом SCSI, у которых адресация секторов линейная, BIOS контроллера SCSI эмулирует трехмерную геометрию. Программный интерфейс с линейной адресацией логических блоков предоставляют только новые расширения *INT 13h* (функций с номерами 4xh).

Традиционно дисковый сервис подразделяет физические диски на *дискеты* (diskette) и *фиксированные диски* (fixed disk). Набор функций (табл. 5.1) для этих классов устройств несколько различается как по составу, так и по реализации совпадающих функций. Классы различаются по диапазонам номеров физических устройств: для дискет отводятся номера 0-7Fh (реально только 0-3), а для фиксированных дисков — 80h-FFh. Таким образом, признаком, по которому BIOS определяет, какой процедуре передать обслуживание запроса, является бит 7 в байте номера устройства. По смыслу бит 7 номера диска разделяет устройства на «дискеты» и «не дискеты». Если бит 7=0, выполняется процедура для контроллера НГМД, а если бит 7=1, выполняется процедура для контроллера жестких дисков. Контроллеры дисковых интерфейсов, имеющие в своем составе дополнительные модули BIOS, перехватывают вектор *INT 13h,* беря на себя обслуживание своих устройств. Когда в IBM PC/XT появились жесткие диски со своим контроллером, его модуль BIOS, инициализирующийся во время POST, вставал на место *INT 13h,* а указатель на исходный обработчик дискового сервиса (драйвер НГМД из системной BIOS) сохранялся на месте *INT40h.* Хотя поддержка жестких дисков давно уже включена в системную BIOS, ради совместимости возможность использования прерывания *INT40h* для вызова драйвера гибких дисков сохраняется. Интерфейс этого вызова совпадает с *INT 13h,* но номер устройства (в регистре DL) не должен превышать 7Fh.

Кроме функций дискового сервиса *(INT 13h), с* дисковыми устройствами связаны еще и векторы, обслуживающие аппаратные прерывания от контроллера НГМД — *INT0Eh* (линия IRQ 6) и от контроллера жестких дисков — *INT76h* (линия IRQ 14). При наличии двухканального порта ATA второй канал обычно использует линию-IRQ 15 (вектор 77h). В XT контроллер жестких дисков использовал линию IRQ 5 (вектор 0Dh). Дополнительные контроллеры дисков могут использовать и другие прерывания. Аппаратные прерывания вырабатываются контроллерами по завершении внутренних операций (нормальном и аварийном). На эти прерывания BIOS не реагирует, а при инициализации их векторы направляются на программную заглушку (инструкцию *IRET).*

Стандартные драйверы дисковых функций BIOS (включая и расширенный сервис) имеют однозадачное происхождение (наследие «тяжелого прошлого» с процессором 8088). Во время выполнения функции значительное процессорное время может затрачиваться на ожидание завершения операции относительно медленным (по своей электромеханической природе) устройством. Драйверы многозаданного режима построены иначе: у них есть вызывающая часть, инициализирующая начало операции, и обработчик аппаратного прерывания от контроллера, сообщающий операционной системе о выполнении операции и результате. Таким образом, время, необходимое контроллеру и накопителю для выполнения операции, для процессора остается свободным. Естественно, что многозадачный драйвер операционной системы должен иметь точное представление об аппаратной реализации контроллера, поскольку общение с ним идет на уровне регистров, а не функций BIOS. Это является одной из причин того, что «серьезную» операционную систему нужно устанавливать на компьютер, на котором она будет работать, в конкретном аппаратном окружении. Простое копирование операционной системы с соседнего компьютера может быть чревато серьезными проблемами и даже потерей данных.

Сервисы BIOS работают в реальном (16-разрядном) режиме процессора. Для современных ОС, работающих в 32-разрядном защищенном режиме, вызовы функций BIOS неэффективны еще и по причине необходимости согласования режимов, сопровождаемого большими накладными расходами. Своей функцией 48h расширенный дисковый сервис дает операционной системе все необходимые сведения, необходимые для реализации обеих частей дисковых драйверов, работающих в защищенном режиме процессора.

Интерфейс дискового сервиса допускает многосекторные операции чтения и записи, но для дискет запрашиваемая цепочка секторов не должна переходить за границу адресованного трека. В зависимости от того, поддерживает ли конкретный адресуемый контроллер и привод многосекторные передачи (и разрешена ли эта возможность в BIOS Setup), многосекторный вызов будет транслироваться на физическое устройство либо цепочкой одиночных запросов, либо как многосекторный. Заметим, что некоторые старые накопители могут некорректно отрабатывать многосекторные запросы на физическом уровне. Если имеются подозрения на эту неисправность, многосекторный режим, часто даже для каждого накопителя индивидуально, может быть запрещен опциями BIOS Setup. Конечно, при этом теряется производительность из-за увеличения доли «накладных расходов» на формирование запросов.

3.1.1. Традиционный сервис BIOS

Традиционный дисковый сервис работает в 16-разрядном режиме процессора, все параметры вызова передаются через регистры процессора. Адрес сектора задается в системе CHS и размещен весьма специфично (рис. 3.1). Сервис вызывается *программным прерыванием INT 13h,* при вызове принимаются следующие соглашения:

- *Номер функции* задается в регистре *AH.* Номер функции не должен превышать 3Fh.
- *Логический номер диска* (drive number) — в регистре *DL* (бит 7=0 — признак обращения к НГМД). В каждом классе устройства нумеруются с нуля: DL=00h — дисковод A:, DL=01h — дисковод B:, DL=80h — первый жесткий диск (диск C:), DL=81h — второй жесткий диск (диск D:) и т. д.
- *Номер цилиндра* (cylinder number) ограничен 10 битами (!), два старших бита — в регистре *CL[7,6],* младшие 8 бит — в регистре *CH.* Диапазон номеров цилиндра - 0-1023.
- *Номер головки* (head number), 8 бит — в регистре *DH* (диапазон 0-255).
- *Номер начального сектора* (sector number), 6 бит — в регистре *CL[5:0]*. В отличие от всех других параметров, нумерация секторов начинается с единицы (диапазон 1-63).
- *Количество секторов,* участвующих в операции, 8 бит — в регистре *AL* (0-255).
- Указатель на *начало буфера оперативной памяти* для считываемых и записываемых данных (address of buffer) — в регистрах *ES:BX.*
- *Результат выполнения* операции определяется по флагу переноса: CF=0 — успешное выполнение операции, CF=1 — обнаружены ошибки. *Код состояния* возвращается в регистре *AH* (табл. 3.5 и 3.6). Код завершения последней операции с дискетами хранится по адресу 40:4lh, с жесткими дисками — 40:74h.
- *Таблица параметров диска* для дискет (DPT, см. табл. 3.2) задана указателем в памяти по адресу 0:78h, для жестких дисков (HDPT, см. табл. 3.3) — 0:104h или 0:118h.

Регистр	Бит 7	6	5	4	3	2	1	0
DL	D7	d6	d5	d4	d3	d2	d1	d0
DH	H7	h6	h5	h4	h3	h2	h1	h0
CL	C9	c8	S5	s4	s3	s2	s1	s0
CH	c7	c6	c5	c4	c3	c2	c1	c0

Рис. 3.1. Размещение адреса в регистрах процессора: D7...D0 — логический номер диска, C9...c0 — номер цилиндра, H7...h0 — номер головки, S5...s0 — номер сектора

Список функций традиционного сервиса приведен в табл. 3.1. Устройства могут не поддерживать некоторые функции, о чем драйверы должны честно сообщить кодом возврата 0lh.

Таблица 3.1. Функции традиционного дискового сервиса

Номер функции	Назначение	Использование регистров, указателей и таблиц параметров				
AH		DL	DH, CL, CH	AL	ES:BX	DPT/HDPT
00h	Reset Disk System — сброс дисковой системы (всех контроллеров и устройств), позиционирование на нулевой цилиндр	-	-	-	-	-
01h	Read Status of Last Operation — чтение состояния последней операции	+	-	-	-	-
02h	Read Sectors into Memory — чтение секторов с диска в память	+	+	+	+	+
03h	Write Sectors from Memory — запись секторов из памяти на диск	+	+	+	+	+
04h	Verify Sectors — верификация секторов (холостое чтение без записи в память и проверка CRC/ECC)	+	+	+	-	+
05h	Format Desired Track — форматирование трека	+	+	+	+	+

Таблица 3.1. Функции традиционного дискового сервиса (продолжение)

Номер функции	Назначение	Использование регистров, указателей и таблиц параметров				
AH		DL	DH, CL, CH	AL	ES:BX	DPT/HDPT
08h	Get Drive Parameters — получение параметров диска	+[3]	+[3]	+[3]	-	+[3]
09h[1]	Initialize Drive Parameters — инициализация таблиц параметров диска	+	-	-	-	+
0Ah[1]	Read Long — «длинное» чтение (сектор и поле ECC)	+	+	+	+	+
0Bh[1]	Write Long — «длинная» запись (сектор и поле ECC)	+	+	+	+	+
0Ch[1]	Seek — поиск цилиндра	+	+	-	-	-
0Dh[1]	Alternative Disk Reset — альтернативный сброс (не затрагивая контроллера дискет)	+	-	-	-	-
10h[1]	Test Drive Ready — проверка готовности	+	-	-	-	-
11h[1]	Recalibrate — рекалибровка (позиционирование на нулевой цилиндр)	+	-	-	-	-
14h[1]	Controller Internal Diagnostics — диагностика контроллера жестких дисков	-	-	-	-	-
15h	Read DASD Type — получение типа диска: *AH=0*—нет диска; *AH=1* —дискета, без датчика смены диска; *AH=2* — дискета, с датчиком смены диска; *AH=3* — жесткий диск; иные значения — код ошибки. *CX:DX* содержат число 512-байтных секторов на диске	+	-	-	-	-
16h[2]	Diskette Change Line Status — проверка статуса смены дискеты: При *CF=0, AH=0* - смены носителя не было. При *CF=1, AH=1* — недопустимый номер диска; *AH=6* — была смена диска или определение смены не поддерживается; *AH=80h* — дисковод не готов или не установлен; иные значения — код ошибки	+	-	-	-	-
17h[2]	Set Diskette Type for Format — установка типа дискеты для форматирования (перед форматированием)	+	-	+[3]	-	-
18h[2]	Set Media Type for Format — установка типа носителя (для форматирования)	+	+[3]	-	-	-
20h[2]	Get Media Type — получение типа установленного носителя	+	-	-	-	-
24h[1]	Set Multiple Mode — установка параметров режима многосекторного обращения (в *AL* — число секторов за операцию)	+	-	+	-	-
25h[1]	Identify Drive ATA — идентификация накопителя (только для ATA-дисков)	+	-	-	+	-

[1] Только для фиксированных дисков.
[2] Только для дискет и других сменных носителей.
[3] См. описание функции.

Функции 00h - 04h, 0Dh, 10h и 14h особых пояснений не требуют. Для обычных операций считывания и записи данных в буфере находятся поля данных секторов, участвующих в обмене.

Функция 05h выполняет форматирование трека согласно структуре, описанной в буфере. Для *дискет* каждому сектору соответствует 4-байтная структура, содержащая (по порядку байтов) номер трека, номер головки, номер сектора и описатель размера сектора (0lh — сектор 256 байт, 02h — 512 байт, 03h — 1024 байт). Варьируя номера (и их последовательность) секторов, а иногда и их размер, можно формировать нестандартные дискеты, используемые как ключи защиты программного обеспечения. Если программа, считывающая ключ, способна определить еще и последовательность секторов (это можно сделать по задержкам между считыванием секторов), то подобрать такой ключ будет нелегким делом. Для *жестких дисков* структура данных проще — на каждый сектор отводится пара байтов, первый является признаком дефектности (80h) или годности (00h) сектора, второй —

номером сектора. Для дисков ESDI, SCSI и большинства дисков ATA функция форматирования не поддерживается.

Функция 08h позволяет определить параметры привода и число устройств на контроллере. В регистрах *CX* и *DH* возвращаются максимальные значения номера цилиндра, головки и сектора (формат см. на рис. 3.1). Для *дискет* в *DL* сообщается количество установленных дисководов, в *BL* — тип привода (значение из CMOS: 01 - 360 Кб, 02 - 1,2 Мб, 03 - 720 Кб, 04 - 1,44 Мб, 06 - 2,88 Мб); (ES:DI) указывает на 14-байтную таблицу параметров (табл. 5.2), ассоциированную с носителем максимального объема, допустимого для данного устройства. Для *жестких дисков в DL*— количество дисков на первом контроллере, а в *ES:DI* возвращается указатель на таблицу HDPT (табл. 3.3 или 3.4), в которой содержится подробное описание диска. Из-за интерфейсных ограничений на разрядность номера головки и цилиндра параметры жесткого диска, сообщаемые через регистры, могут отличаться от параметров, указанных в HDPT. Через регистры можно описать диск в пределах 1024×256×63 (8,4 Гбайт), через HDPT - 65 536×256×63 (33,8 Гбайт).

Функция 09h инициализирует выбранный жесткий диск в соответствии с параметрами, хранящимися в его HDPT. Для дисков ESDI и SCSI никаких действий не выполняется (они свои параметры устанавливают сами, не пользуясь таблицами).

Функции 0Ah и **0Bh** «длинного» чтения и «длинной» записи. В буфере находятся данные сектора и относящиеся к нему байты CRC или ECC (по умолчанию 4 байта, но производитель может изменить длину этого поля). Если операция чтения завершилась с обнаружением исправимой ошибки ECC, то в буфере находятся корректные данные. «Длинные» операции возможны лишь с одним сектором. Кстати, сформировав в буфере блок данных с неверным значением полей CRC/ ECC и выполнив «длинную» запись, можно записать на диск сектор, всегда читаемый с ошибкой. Если есть подозрения на неработоспособность схем контроля диска, этим приемом можно их проверить — контроллер с неисправными (или отсутствующими) схемами контроля ошибку не обнаружит. Для дисков ESDI и SCSI функция не поддерживается.

Функция 0Ch выполняет только поиск заданного цилиндра (головки и сектора), но не вызывает обмена данными. Она полезна для ускорения обмена: ее выполняют, если в буфере (в ОЗУ) еще нет подходящего места для данных, а адрес следующего запроса уже известен. Тогда последующая команда чтения/записи или форматирования выполнится быстрее за счет экономии времени на позиционировании.

Функция 11h «встряхивает» механизм позиционирования, ее обычно используют после неудачной попытки обращения к диску.

Функция 15h определяет тип диска. Некоторые версии BIOS могут сообщать число 512-байтных блоков, хранимых устройством (DX— младшее слово, *CX*— старшее).

Функция 16h определяет факт смены носителя; до применения этой функции следует убедиться в поддержке датчика смены по функции 15h.

Функция 17h задает тип форматируемой дискеты (настраивает скорость передачи данных контроллера), при ее вызове в *AL* заносится значение: 01 —360 Кб дискета в 360 Кб дисководе, 02 — 360 Кб дискета в 1,2 Мб дисководе, 03 — 1,2 Мб дискета в 1,2 Мб дисководе, 04 — 720 Кб дискета в 720 Кб дисководе. Для дискет 1,44 и 2,88 Мб используют функцию 18.

Функция 18h служит тем же целям, но в регистрах *CH* и *CL* задает количество треков и секторов на трек. Согласно спецификации SurePath BIOS (от IBM) в ES:DI возвращает указатель на 14-байтную структуру параметров (табл. 3.2), ассоциированную с выбранным типом носителя.

Функция 20h (необязательная) позволяет определить возможности накопителя со сменными носителями. Значения *AH* при возврате: 30h — нет датчика смены носителя, 31h — в устройстве нет носителя, 32h — привод не сообщает тип носителя. В *AL* возвращается тип установленного носителя: 06h — дискеты для режима 4M (2,88 Мбайт), 04h - 2M (1,44 или 1,2 Мбайт), 03h - 1 M (720,360 или 180 Кбайт), остальные значения зарезервированы.

Функция 24h (необязательная) позволяет заказать число секторов, передаваемых в режиме многосекторных передач за одну команду ATA.

Функция 25h используется для считывания в буфер 512-байтного «паспорта устройства» ATA/ATAPI (см. п. 5.2.1.4.2.).

Таблица параметров дискет (DPT — Diskette Parameter Table) содержит набор 11-байтных структур (табл. 3.2), описывающих свойства дисковых устройств с логическими адресами 0-3. Управляя этими параметрами, ОС и прикладные программы могут манипулировать скоростными характеристиками дисководов (в пределах возможности устройств), что заметно влияет на производительность дисковых операций. Подробнее смысл параметров пояснен в п. 4.6. Указатель на начало таблицы (таблица для нулевого устройства) расположен по адресу 0:78h — на месте вектора прерывания *INT 1Eh,* таблицы для следующих устройств начинаются со смещением 11, 22 и 33 байта соответственно. В спецификации SurePath BIOS (от IBM) предложена более длинная структура (14 байт), ссылку на начало для каждого дисковода можно получить по функции BIOS *INT 13h(18h).*

Таблица 3.2. Таблица параметров дискет (DPT)

Смещение	Назначение
0	Биты 4-7 — время перемещения головок на один шаг (SRT, step rate time), для приводов 1,44 Мбайт по умолчанию Dh. Биты 0-3 — время разгрузки головок (HUT, Head Unload Time), для приводов 1,44 Мбайт по умолчанию Fh
1	Бит 0: 1 =DMA не используется (всегда 0) Биты 1-7 — время успокоения после загрузки головок (HLT, Head Load Time):, для приводов 1,44 Мбайт по умолчанию 0000001.
2	Время выключения мотора после последней операции (в интервалах по 55мс), по умолчанию 25h (2,03 с)
3	Размер сектора: 0=128 байт или в соответствии с байтом 6, 1=256 байт, 2=512 (стандартный размер), 3=1024
4	Номер последнего сектора на треке (EOT), при тривиальной нумерации соответствует числу секторов на треке
5	Временной зазор для операций чтения/записи (GPL1, 2Ah для 5", 1Bh для 3,5")
6	Длина сектора в байтах, если байт 3=0
7	Временной зазор для операции форматирования трека (GPL2, 50h для 5", 54h для 3,5")
8	Символ-заполнитель для форматирования (обычно F6h)
9	Время успокоения головок после позиционирования (Head Settle Time), мс; по умолчанию 0Fh (15 мс)
0Ah	Время разгона двигателя (в интервалах по 125 мс), по умолчанию 8 (1 с)
0Bh[1]	Максимальный номер трека
0Ch[1]	Скорость передачи данных
0Dh[1]	Тип дисковода в CMOS

[1] Для SurePath BIOS (IBM)

Таблицы параметров жестких дисков (HDPT — Hard Disk Parameter Table), в отличие от предыдущих, очень сильно «морально устарели». Стандартная таблица (табл. 3.3) не позволяет различать физическую и логическую геометрию (см. п. 3.1.2), зато в ней есть параметры предкомпенсации записи и парковки, которые уже очень давно не используются. Указатель на таблицу первого жесткого диска (HDD 0, логический адрес 80h) расположен по адресу 0:104h (на месте вектора прерывания *INT41h),* второго (HDD 1, логический адрес 81 h) — 0:118h (INT46h), а для других дисков стандартного места не предусмотрено. Адрес таблицы параметров для любого жесткого диска можно получить в регистрах *ES:DI* по функции 8 (получение параметров диска). Для того чтобы вступили в силу изменения, внесенные программой в таблицы параметров первого или второго дисков, следует выполнить *INT 13h (9),* указав только номер диска (80 или 81h). При этом предполагается, что указатель на соответствующую таблицу находится на штатном месте. В расширенной BIOS, в которой используется трансляция геометрии (см. п. 3.1.2), таблица параметров имеет вид, приведенный в табл. 3.4. Признаком данного формата таблицы является байт сигнатуры Axh (смещение 3), значение которого зависит от способа трансляции.

Таблица 3.3. Стандартная таблица параметров жестких дисков (HDPT)

Смещение	Длина	Назначение
0	2	Число цилиндров
2	1	Число головок
3	2	Цилиндр, начиная с которого снижается ток записи (только для XT)[1]
5	2	Цилиндр, начиная с которого включается предкомпенсация записи[1]
7	1	Длина поля ECC
8	1	Флаги операций: Биты 0-2 — опции привода (только для XT) Бит 3:1— число головок больше 8 (только для AT) Бит 4=0 Бит 5:1— карта дефектов записана на цилиндре, следующем за максимальным Бит 6 — запрет исправления ошибок с помощью ECC Бит 7 — запрет повторов в случае ошибок
9	1	Тайм-аут для обычных операций (только для XT)
0Ah	1	Тайм-аут для операции форматирования (только для XT)
0Bh	1	Тайм-аут для операции проверки устройства (только для XT)
0Ch	2	Номер цилиндра зоны парковки (только для XT)
0Eh	1	Число секторов на треке (для AT)
0Fh	1	Резерв

[1]Устарело, не используется.

Параметры жестких дисков заносятся в таблицы во время начального тестирования компьютера (во время POST). Они задаются утилитой BIOS Setup или определяются автоматически по фактически подключенным устройствам (в зависимости от настройки Setup). Таблицы параметров жестких дисков не позволяют описывать диски размером более 33,8 Гбайт, поскольку предел физической геометрии — 65 536×16×63. Однако для современных ОС, использующих большие диски, это не проблема — они попросту данными таблицами не пользуются, а работают в режиме LBA.

Итак, традиционный дисковый сервис предоставляет набор базовых функций (чтение, запись, форматирование, определение параметров), достаточный для работы ОС и приложений. Однако применительно к жестким дискам у него есть ряд ограничений:

- малый (по нынешним меркам) размер обслуживаемого диска — 8,4 Гбайт (общее ограничение), а для дисков ATA (самых популярных) — 528 Мбайт;
- в системе зарезервировано место под таблицы параметров только для двух жестких дисков, в то время как к двухканальному контроллеру ATA можно подключить до четырех;
- интерфейс взаимодействия с вызывающей программой — 16-разрядный, что не позволяет в полной мере использовать возможности современных процессоров.

Таблица 3.4. Транслированная таблица параметров жестких дисков

Смещение	Длина	Назначение
0	2	Логическое число цилиндров
2	1	Логическое число головок (не более 256,0 означает 256)
3	1	Сигнатура таблицы (Axh)
4	1	Физическое число секторов на треке (не более 63)
5	2	Цилиндр, начиная с которого включается предкомпенсация записи[1]
7	1	Резерв
8	1	Флаги операций: Биты 0-2 — опции привода (только для XT) Бит 3:1 — число головок больше 8 (только для AT) Бит 4=0 Бит 5:1— карта дефектов записана на цилиндре, следующем за максимальным Бит 6 — запрет исправления ошибок с помощью ECC Бит 7 — запрет повторов в случае ошибок
9	2	Число физических цилиндров, не более 65 536 (0 означает 65 536)
0Bh	1	Число физических головок, не более 16 (0 означает 16)
0Ch	2	Номер цилиндра зоны парковки (только для XT)
0Eh	1	Логическое число секторов на треке (для AT)
0Fh	1	Контрольная сумма (двоичное дополнение байтов 0-14)

[1] Устарело, не используется.

Таблица 3.5. Коды ошибок обращения к НГМД

Код (AH)	Ошибка
00h	No error — нет ошибок
01h	Invalid diskette parameter or function not supported — неверные параметры дискеты или неподдерживаемая функция
02h	Address mark not found — не найден адресный маркер
03h	Write-protect error — попытка записи на защищенный диск
04h	Requested sector not found — сектор не найден
06h	Diskette change line active — обнаружен сигнал смены носителя
08h	DMA overrun on operation — переполнение операции DMA
09h	DMA attempt across a 64KB boundary — попытка пересечь границу 64-килобайтных блоков при операции DMA
0Ch	Media type not found — не определен тип носителя
10h	Cyclic redundancy check (CRC) error on diskette read — ошибка CRC-кода при чтении
20h	General controller failure — общий отказ контроллера
30h	Drive does not support media sense — накопителем не поддерживается датчик типа носителя
31h	No media in drive — нет носителя в приводе
32h	Media type not supported by drive — тип носителя не поддерживается приводом
40h	Seek operation failed — ошибка операции поиска
80h	Diskette drive not ready — неготовность дисковода

Таблица 3.6. Коды ошибок обращения к фиксированным дискам

Код (AH)	Ошибка
00h	No error — нет ошибок
01h	Unsupported function or parameter — неподдерживаемая функция или параметр
02h	Address mark not found — не найден адресный маркер
04h	Sector not found — сектор не найден
05h	Reset failed — ошибка при сбросе
07h	Drive parameter activity failed — ошибка активации параметров диска
09h	Data-boundary error — нарушение границы данных
0Ah	Bad sector flag detected — обнаружен сектор, помеченный как дефектный
0Bh	Bad cylinder detected — обнаружен дефектный цилиндр
0Dh	Invalid number of sectors on format — недопустимое для формата количество секторов
0Eh	Control data address mark detected — обнаружен адресный маркер управляющих данных
0Fh	DMA arbitration level out of range — арбитражный уровень DMA вне диапазона
10h	Uncorrectable error checking and correction (ECC) or cyclic redundancy check (CRC) error - ошибка, неисправимая с помощью ECC или обнаруженная с помощью CRC
11h	ECC corrected data error — ошибка, исправленная ECC (данные в памяти корректны)
20h	General controller failure — общая ошибка контроллера
40h	Seek operation failed — ошибка операции поиска
80h	Recal drive or controller time-out — тайм-аут операции рекалибровки
AAh	Drive not ready — привод не готов
BBh	Undefined error occurred — неопределенная ошибка
CCh	Write fault on selected drive — ошибка записи
E0h	Status error/error register=0
FFh	Sense operation failed — ошибка операции определения носителя

Поскольку на использовании сервиса BIOS *INT 13h* основана загрузка всех ОС, а в MS DOS (и не только) на этом сервисе держатся вообще все обращения к дискам (многозадачные системы, в том числе и Windows 9x, будучи загруженными, пользуются собственными драйверами), возникла необходимость продлить ему жизнь, сохраняя обратную совместимость.

3.1.2. Преодоление барьера 528 Мбайт (ECHS и LBA)

Формально традиционный сервис позволяет работать с дисками, имеющими до 1024×256×63= 16 515 072 секторов (около 8,4. Гбайт). Ряд операционных систем имеет ошибку, не позволяющую использовать полный объем, допустимый данным сервисом. Для дисков объемом более 15 481 935 секторов следует пользоваться только функциями расширенного сервиса (см. п. 3.1.3). Однако при работе с устройствами ATA имеется еще и барьер в 528 Мбайт. Дело в том, что контроллер жесткого диска ATA (и ST-506/412), на который ориентированы драйверы *INT13h,* имеет только 4-битный регистр номера головки (а в BIOS — 6 бит). Правда, этот же контроллер способен принимать 16-битный номер цилиндра (в BIOS — 10 бит). Понятно, что непосредственно без искажений через эти два фильтра (формат вызова и формат регистров контроллера) может пройти только вызов с самыми жесткими ограничениями по каждой координате. Тогда ограничение, полученное тем же перемножением диапазонов координат, получается около 528 миллионов байт:

(2^{10}=1024 цилиндра) × (2^4=16 головок) × (2^6-l=63 сектора) ×512 байт = 528 482 304 байт.

Для преодоления 528-мегабайтного барьера у дисков ATA, не трогая программного интерфейса, в BIOS ввели расширение традиционного дискового сервиса. Интерфейс ATA в трехмерной геометрии позволяет реализовать довольно большой (но уже не запредельный) объем диска:

(2^{16}=65 536 цилиндров) × (2^4=16 головок) × (2^8-1=255 сектора) × 512 байт = 136,9 Гбайт.

Чтобы достичь хотя бы интерфейсного ограничения BIOS (8,4 Гбайт), стали применять *трансляцию параметров вызова* функций *INT 13h,* которые будем теперь называть *логическими,* в *физические* параметры, передаваемые контроллерам ATA-дисков. (Зная устройство современных дисковых накопителей, здесь и далее мы не будем добираться до истинно физических параметров — реального номера цилиндра, головки и сектора) В функции, которая сообщает параметры диска (функция 8), производится обратная трансляция, так что *на стороне вызова программного интерфейса INT13h присутствуют только логические параметры.* Естественно, логический объем диска не может превышать физического: $(C\times H\times S)_{лог} \le (C\times H\times S)_{физ}$

Есть два основных пути трансляции: оставаясь в пределах трехмерной геометрии *CHS* (Cylinder-Head-Sector) и с переходом на логическую адресацию блоков *LBA,* если таковую поддерживает

накопитель. Возможности выбора пути трансляции определяются конкретной версией BIOS.

При использовании режимов, называемых *Large Disk* или *ECHS* (Extended CHS), общение с накопителем производится в трехмерной системе, но с преобразованными номерами цилиндров и головок. Главная задача преобразования — уменьшить число логических цилиндров, которое физически у больших дисков превышает предельные 1024, и увеличить логическое число головок при ограничении в 16 физических. Для того чтобы программы могли определить физическую геометрию диска, таблицу параметров HDPT модернизировали и на место устаревших параметров ввели описание физической геометрии. Там, где раньше было описание геометрии, теперь находятся логические параметры, используемые при вызовах. Попутно в расширении увеличили и число поддерживаемых таблиц (по крайней мере для четырех дисков).

Способов трансляции, к сожалению, может быть несколько. Можно, например, физическое количество цилиндров диска разделить на К (К=2, 4, 8,16..., при этом параметр преобразования К определяется так, чтобы результат не превышал 1024), а физическое число головок умножить на К (это произведение не должно превышать 255). Умножение и деление на степень числа 2 легко выполняется с помощью сдвигов, программный код трансляции получается компактным и работает быстро. Эта схема преобразования, называемая bit shift translation, рекомендуется фирмой Phoenix для включения в расширенный дисковый сервис BIOS. Она работает на любых дисках ATA, как поддерживающих LBA, так и не поддерживающих. Количество секторов на треке при трансляции не изменяется. Логические (параметры *INT 13h)* и физические (передаваемые устройствам ATA) параметры дисков для данной схемы приведены в табл. 3.7. Специально под эту схему в приложениях к ATA-2 и более поздним стандартам приводятся правила сообщения геометрии устройствами разной емкости, позволяющие наиболее эффективно использовать пространство диска (минимизируя число потерянных секторов). К сожалению, это не единственный способ трансляции геометрии, но здесь другие рассматривать не будем. Нам важно знать лишь цель преобразования и тот факт, что *из-за несовпадения методов трансляции в режимах Large Disk (ECHS) возможна несовместимость дисков, размеченных при различных версиях BIOS.*

Таблица 3.7. Трансляция геометрии с помощью сдвигов битов

Параметры устройства ATA		Параметры, сообщаемые INT13h		Теоретический предел
Число цилиндров C	Число головок H	Число цилиндров	Число головок	
1 - 1024	1-16	C	H	528.4Мбайт
1024 - 2048	1-16	C/2	H×2	1,057Гбайт
2048 - 4096	1-16	C/4	H×4	2,114Гбайт
4096 - 8192	1-16	C/8	H×8	4,228 Гбайт
8192 - 16384	1-16	C/16	H×16	8,456 Гбайт
16384-32768	1-8	C/32	H×32	8,456 Гбайт
32768 - 65536	1-4	C/64	H×64	8,456 Гбайт

В режиме *LBA* параметры стандартных вызовов транслируются в *линейный адрес,* который вычисляется однозначно в «естественном» порядке счета секторов. Сектору с нулевым логическим адресом соответствует первый сектор нулевой головки нулевого цилиндра. Общая формула вычисления логического адреса выглядит так:

LBA = (CYL × HDS+HD) × SPT + SEC - 1,

где CYL, HD и SEC — логические номера цилиндра, головки и сектора в пространстве CHS; HDS — количество головок; SPT— количество секторов на треке. Трехмерную геометрию в ответ на вызов функции 8 BIOS *INT 13h* сообщает следующим образом:

- количество секторов на треке *(SPT)* — всегда 63;
- количество головок *(HDS)* — 16, 32, 64, 128 или 255, в зависимости от объема диска;
- количество цилиндров определяется делением общего числа секторов на произведение *SPT×HDS.*

Число головок выбирается так, чтобы полученное число цилиндров не превышало 1024 (табл. 3.8). Недостатком данного метода являются более сложные вычисления при преобразованиях (здесь при умножении и делении уже не обойтись просто сдвигами).

Когда в CMOS Setup выбирается режим LBA, это означает использование схемы трансляции LBA assisted translation, отвечающей табл. 3.8. При этом вовсе не обязательно использование линейной адресации в обращениях к физическим устройствам ATA — если диск способен установить геометрию из данной таблицы (по ATA-команде *Initialize Device Parameters),* то обращения могут производиться и в режиме CHS. Обращение к диску в режиме LBA, безусловно, выгодно, лишь когда BIOS обслуживает

вызов расширенной функции 4x (см. ниже), поскольку при этом не требуется никакой трансляции (экономится процессорное время). Для того чтобы определить, в каком режиме используется диск ATA, можно прочитать байт из регистра *D/H* (1F6 для первичного контроллера или 176 для вторичного). Значение Ex или Fx указывает на то, что в последней операции использовался режим LBA (для ведущего или ведомого устройства соответственно), Ax или Bx — режим CHS. Прочитать порт можно, например, с помощью команды «I 1F6» (I 176 для вторичного контроллера) отладчика DEBUG (в среде DOS).

Таблица 3.8. Трансляция геометрии с использованием LBA

Число секторов в устройстве ATA, N	Параметры, сообщаемые INT 13h (в регистрах)			Теоретический предел
	Число секторов на трек	Число головок	Число цилиндров	
1 - 1 032 192	63	16	N/(63x16)	528,4Мбайт
1 032 192 - 2 064 384	63	32	N/(63x32)	1,057Гбайт
2 064 384 - 4 128 768	63	64	N/(63x64)	2,114Гбайт
4 128 768 - 8 257 536	63	128	N/(63x128)	4,228 Гбайт
8 257 536 - 16 450 560	63	255	N/(63x255)	8,422Гбайт

Итак, для дисков размером более 528 Мбайт BIOS определяет фиктивные параметры геометрии (если большие диски поддерживаются). Операционной системе и прикладным программам они будут сообщаться как реальные, а BIOS при вызовах дисковых сервисов будет преобразовывать параметры вызова так, что номера цилиндров и головок будут принадлежать пространству «внешних координат» ATA-диска.

Еще раз подчеркнем, что 528-мегабайтный барьер для дисков ATA имеет чисто *программное происхождение.* Электрический интерфейс и регистры контроллера накопителя ограничивают объем до 136,9 Гбайт в трехмерной адресации и 0,5 Тбайт в линейной. Средства преодоления этого барьера носят чисто программный характер и должны быть реализованы на уровне BIOS. Здесь могут быть следующие варианты:

- Современные версии BIOS имеют встроенный механизм трансляции, который включается для дисков объемом более 528 Мб. Трансляция может использовать режим LBA, если его поддерживает диск, или ECHS (Large Disk), который может работать с любыми ATA-дисками. Выбор режима трансляции выполняется при конфигурировании диска в BIOS Setup. Ряд версий BIOS не поддерживают LBA, некоторые не поддерживают ECHS. Если в компьютере применяется флэш-BIOS, но поддержки трансляции нет, то имеет смысл обновить версию BIOS, чтобы проблема преодоления барьера решалась самым лаконичным способом.
- Для компьютеров со старыми версиями BIOS, не имеющими встроенной трансляции и не допускающими легкого обновления, выпускались платы интерфейса E-IDE, снабженные микросхемой расширения ROM BIOS. Эта микросхема хранит программный код драйвера *INT 13h,* реализующего тот или иной тип трансляции (обычно ECHS). В остальном плата могла и не отличаться от традиционного адаптера ATA, хотя могла иметь интерфейс шины VLB (реже PCI) и более сложный контроллер с буферами и управлением шиной (Bus-Master).
- Если BIOS трансляцию не поддерживает, а плата с расширением ROM BIOS не применяется, то остается программная загрузка драйвера *INT 13h.* Однако этот драйвер должен быть загружен до загрузки операционной системы, которая уже опирается на BIOS. Такой драйвер может быть загружен через главный загрузчик — MBR (Master Boot Record). Код этого загрузчика исполняется в конце POST, и если он до загрузки загрузчика операционной системы (Boot Record из активного раздела диска) поместит в память новый драйвер *INT 13h,* то операционная система воспримет его как часть BIOS. На таком принципе работают драйверы типа Disk Manager и EZ-Drive. Такое решение проблемы имеет ряд неудобств. Во-первых, загружаемый драйвер использует часть области стандартной памяти, которую бывает жалко. Во-вторых, процедура конфигурирования диска с таким драйвером сложнее, чем обычного, поскольку требует инсталляции дополнительного программного обеспечения, да еще и размещаемого в системной области диска. И в-третьих, в нештатных ситуациях (например, под действием вируса) обслуживание диска становится более тонкой задачей. Например, универсальное (хотя и не общеизвестное) «лечение» главной загрузочной записи командой FDISK /MBR затрет загрузчик драйвера стандартным загрузчиком, и процедуру конфигурирования диска придется повторять. Конечно, в Disk Manager имеются соответствующие утилиты, но о необходимости их использования нужно помнить, а сами утилиты хранить на дискете. Однако ради большого объема диска приходится идти и на такие издержки.

Поскольку преобразования в LBA отличаются от ECHS, диски, размеченные в режиме LBA, не будут работать в режиме ECHS, и наоборот. Тот же результат возможен при переносе диска,

размеченного в режиме ECHS, на другой компьютер, у которого применен другой алгоритм, трансляции. Если повезет, то эта неработоспособность проявится сразу невозможностью загрузки операционной системы (сообщение «Missing Operation System»).

Значительно хуже, когда ОС загрузилась и кажется, что все хорошо. А в это время, возможно, все записи на диск идут по неправильным адресам, и крупная потеря данных приближается неумолимо. По той же причине *нельзя пытаться решать проблему больших дисков, форматируя их на компьютере с расширенной BIOS, а эксплуатируя на машине со стандартной версией, имеющей 528-мегабайтный барьер.* Сначала все будет хорошо, но когда начальная 528-мегабайтная область заполнится, следующая запись пойдет по нулевому цилиндру. А там были и Master Boot, *и* таблица разделов, и копии FAT. До области данных эта перекрывающая запись дойти, скорее всего, не успеет, но легче от этого не станет — данные будут потеряны. Сказанное не очень актуально для операционных систем, не использующих BIOS при обращении к дискам, — если такую ОС удается загрузить с диска, то проблему большого диска можно считать решенной.

3.1.3. Расширенный сервис BIOS

В предыдущей части мы говорили о способе преодоления первого барьера емкости диска (528 Мбайт), вызванного несогласованностью ограничений параметров вызовов сервиса BIOS и устройств ATA. Однако давно уже актуально преодоление ограничений интерфейса самой BIOS, которое, как было показано выше, составляет около 8,4 Гбайт- Чтобы получить возможность работы через BIOS с дисками большего объема, потребовалось ввести новые функции дискового сервиса.

Расширенный дисковый сервис BIOS Enhanced Disk Drive Services (EDD), продвигаемый фирмой Phoenix Technologies LTD, реализуется многими разработчиками BIOS и устройств массовой памяти. Он позволяет работать с устройствами, имеющими объем до 2^{64} секторов, эффективно используя архитектуру процессоров IA-32 и IA-64. Сервис оперирует линейным логическим адресом сектора (LBA). Вместо традиционных таблиц параметров дисков в нем используются новые, дающие исчерпывающую информацию об устройствах, их физической организации и интерфейсе. Устройства могут иметь сменные носители и сами быть съемными в процессе работы компьютера (например, подключенные к шине USB или IEEE 1394), так что понятие «сменяемость носителя» несколько размывается. Такие устройства должны поддерживать механизм уведомления о смене носителя и программное блокирование смены носителя. По прогнозам, емкости данного интерфейса должно хватить на 15-20 лет.

Расширения BIOS *INT 13h* используют ОС Windows 95, Windows 98, Windows 2000. Правда, это использование ограничено лишь начальной загрузкой и процессом инсталляции (FDISK, FORMAT), поскольку в регулярной работе используются собственные 32-разрядные драйверы. Их не используют DOS (все версии), Windows 3.1x, Windows NT, Novell NetWare, OS/2 Warp, Linux, Unix.

В настоящее время определены три набора функций:

- доступ к фиксированным дискам (fixed disk access subset) — функции 41-44h, 47h и 48h;
- блокировка и смена носителя (device locking and ejecting subset) — функции 41h, 45h, 46h, 48h и 49h;
- поддержка расширенных дисков (enhanced disk drive (EDD) support subset) — функции 41h и 48h.

Для эмуляции дисков на загружаемых CD-ROM имеются расширенные функции 4A-4Dh, описанные в п. 7.3.8.

Расширенный сервис, как и традиционный, вызывается *программным прерыванием INT 13h* с номерами функций свыше 3Fh (регистр *AH),* номер устройства (регистр DL) допустим в диапазоне 80h-FFh. Основные параметры вызова — начальный адрес блока, число секторов для передачи и адрес буфера — передаются через *адресный пакет* (device address packet). Формат пакета, по сравнению с передачей параметров традиционного сервиса через регистры процессора, довольно просторный (табл. 3.9).

Поскольку расширение BIOS может и отсутствовать, имеется *функция проверки его наличия* (номер 41h). Расширение может действовать избирательно (не для всех устройств), так что проверку надо производить для конкретного устройства, интересующего программу. Проверка дает номер версии расширения и карту поддерживаемых наборов функций. *Функции расширенного чтения, записи, верификации и поиска* (42h, 43h, 44h и 47h) по смыслу не отличаются от их аналогов из традиционного сервиса. Для работы со сменными носителями введены *функции отпирания/запирания, извлечения и проверки факта смены носителя* (45h, 46h и 49h). От идеологии традиционного сервиса сильно отличается *функция получения параметров устройства* (48h). Она возвращает в ОЗУ буфер с набором параметров и детальным описанием устройства, позволяющим ОС и приложениям работать с ним, минуя BIOS.

На диске адресуется до 2^{64} секторов. Адрес буфера передачи может задаваться либо в 16-битном формате реального режима (Seg:Offset), либо 64-битным линейным адресом. Число блоков, участвующих в операции, может задаваться двумя способами (см. табл.) и достигать 2^{32}. После выполнения операции в пакете будет число блоков, с которыми операция выполнена успешно (при

ошибке оно будет отличаться от указанного при входе).

Таблица 3.9. Формат адресного пакета

Смещение	Длина	Назначение
0	1	Длина пакета в байтах (не менее 16. При меньшей длине запрос отвергается с CF=1 и AH=01h)
1	1	Резерв(0)
2	1	Число блоков (секторов), подлежащих передаче, в пределах 0-127 (0-7Fh). 0 означает отсутствие передачи. Запрос со значениями 80h-FEh отвергается с CF=1 AH=01h. При значении поля FFh число блоков берется из поля со смещением 18h, при этом адрес буфера берется из поля со смещением 10h
3	1	Резерв(0)
4	4	Адрес буфера передачи в форме *Seg:Offset.* FFFF:FFFFh указывает на использование поля со смещением 10h для задания адреса.
8	8	Логический адрес начального блока (LBA), 64 бит.
10h	8	64-битный линейный адрес буфера передачи; используется, если поле 2=FFh или поле 4=FFFF:FFFFh
18h	4	Число блоков (секторов), подлежащих передаче; используется, если поле 2=FFh
1Ch	4	Резерв(0)

В описании функций подразумевается, что при вызове в регистрах *AH* и *DL* задается номер функции и номер устройства соответственно. При успешном выполнении функция возвращает CF=0, при ошибке — CF=1 и код ошибки в *AH,* Остальные регистры используются в зависимости от функции (см. ниже).

Функция 41h — проверка присутствия расширения (check extensions present). При вызове в *BX* заносится 55AAh. Если расширение присутствует, то при возврате установится CF=0 и BX=AA55h, в *AH* — номер версии расширения (30h для данного описания), в CX — битовая карта поддержки функций:

Бит 0: 1= присутствует поддержка доступа к фиксированным дискам.

Бит 1: 1= присутствует поддержка блокировки и смены носителя.

Бит 2: 1= присутствует поддержка расширенных дисков.

Бит 3: 1= присутствует 64-битное расширение (действительны поля 10h и далее в адресном пакете).

Биты 4-15 — резерв.

Если для заданного устройства расширение не подерживается, устанавливается CF=1 и в *AH* — код ошибки (01).

Функция 42h — расширенное чтение (extended read), *DS:SI* — адресный пакет.

Функция 43h — расширенная запись (extended write), *DS:SI* — адресный пакет. При вызове *cAL=0* или 1 — запись без верификации, *AL=2 — с* верификацией. Если заказанная верификация не поддерживается устройством, запрос отвергается (CF=1, AH=01h*).*

Функция 44h — верификация (verify sectors), DS:SI — адресный пакет.

Функция 45h — запирание/отпирание носителя (lock/unlock media). В *AL* — подкоманда: 0 — запереть носитель, 1 — отпереть, 2 — сообщить состояние. При успешном возврате состояние сообщается в AL: 1 — носитель заперт, 0 — нет. Для каждого устройства должен быть счетчик запираний (до 255); носитель открывается, если каждое запирание будет отперто (счетчик обнулен). Если делается лишнее отпирание, вызов возвращается с CF=1 и *AH* = B0h. Если число запирании превышает допустимое, вызов возвращается с CF=1 и *AH* = B4h. При включении питания и системном сбросе устройства считаются открытыми. Реализация запирания-отпирания зависит от устройства, но ОС и приложения полагают, что смена носителя в запертом устройстве невозможна.

Функция 46h — извлечение носителя (eject removable media), при вызове *AL=0.* Если носитель не сменный, вызов возвращается с CF= 1 и *AH* = B2h. Если носитель заперт, вызов возвращается с CF=1 и *AH* = Blh. Если носителя нет, вызов возвращается с CF=1 и *AH* = 31h. Если носитель отперт, то возможность извлечения определяется через вызов BIOS INT *15h* (функция 52h) и в случае отказа запрос отвергается. Если извлечение разрешено, но при его выполнении возникает ошибка, вызов возвращается с CF=1 и *AH* = B5h.

Функция 52h BIOS *INT 15h* определяет возможность извлечения носителя с точки зрения системы кэширования. При ее вызове *AH=52h* и в *DL* задается номер устройства. Возврат с CF=0 и *AH=0* означает возможность изъятия носителя, при CF=1 и *AH=B1h* или B3h изъятие не разрешается.

Функция 47h — расширенный поиск (extended seek), DS:SI — адресный пакет. Вызов уведомляет устройство о возможном последующем доступе к блокам, указанным в пакете. К моменту возврата поиск может и не быть реально завершен.

Функция 48h — получение параметров устройства (get device parameters). DS:SI — адрес буфера результата. Длина буфера указывается в первом слове буфера (см. табл. 3.10). Если при вызове

указывается длина менее 26, вызов отвергается. Если длина указывается менее 30, то указатель на таблицу параметров диска не возвращается.

В буфере возвращается геометрия, сообщаемая устройством (физическая); логическую геометрию можно получить по *INT 13h(8)*

Таблица 3.10. Содержимое буфера параметров устройства

Смещение	Длина	Назначение
0	2	Длина буфера
2	2	Флаги свойств: Бит 0 — «прозрачная» обработка пересечения границ страниц DMA Бит 1 — геометрия, сообщаемая в словах 4, 8 и 12, действительна Бит 2 — носитель сменяем, биты 4-6 действительны Бит 3 — поддержка записи с верификацией Бит 4 — поддержка уведомления о смене носителя Бит 5 — носитель запираемый Бит 6 — указана максимально возможная геометрия, но носителя в устройстве нет Бит 7 — для доступа к устройству используется функция 50h BIOS *INT 13h* Биты 8-15 — резерв
4	4	Число физических цилиндров
8	4	Число физических головок
12	4	Число физических секторов на треке
16	8	Число секторов. Если оно больше 15 482 880, в слове 2 бит 1 нулевой.
24	2	Число байтов в секторе
26	4	Указатель на расширенную таблицу параметров диска (DPTE, см. табл. 5.11), в формате Seg:Offset. FFFF.FFFFh означает, что DPTE отсутствует.
30	2	BEDDh — ключ присутствия информации о пути к устройству (device path information)
32	1	Длина информации, о пути к устройству, включая ключ (44)
33	1	Резерв(0)
34	2	Резерв(0)
36	4	Тип шины расширения (символы ASCII, дополненные пробелами): PCI - 50h, 43h, 49h, 20h; ISA — 49h, 53h, 41h, 20h
40	8	Тип интерфейса устройства (символы ASCII, дополненные пробелами): ATA, ATAPI, SCSI, USB, 1394, FIBRE, I^2O
48	8	Путь к интерфейсу
56	8	Путь к устройству
64	1	Резерв(0)
65	1	Контрольная сумма информации о пути (сумма байтов 30-73 должна быть нулевой)

Путь к интерфейсу позволяет адресоваться к микросхемам контроллера интерфейса устройства памяти. Он зависит от типа шины расширения:

- Для шины ISA в слове 48 находится базовый адрес портов ввода-вывода (16 бит), байты 50-55 зарезервированы (нулевые).
- Для шины PCI байт 48 содержит номер шины, 49 — номер слота, 50 — функция, 51 — номер канала, байты 52-55 (двойное слово) зарезервированы (нулевые).

Путь к устройству определяется адресацией, характерной для его типа интерфейса:

- ATA — байт 56 определяет номер устройства: 0 — устройство 0 (ведущее), 1 — устройство 1 (ведомое).
- ATAPI — байт 56 определяет номер устройства (аналогично ATA), байт 57 — номер логического устройства (LUN).
- SCSI — слово 56 определяет адрес устройства (SCSI ID), учетверенное слово 58 (8 байт) — номер логического устройства (LUN).
- USB — учетверенное слово 56 (8 байт) задает 64-битный серийный номер устройства, определенный спецификацией USB для устройств памяти.
- 1394 — учетверенное слово 56 (8 байт) задает 64-битный расширенный уникальный идентификатор EUI-64 (Extended Unique Identifier).
- FIBRE — учетверенное слово 56 (8 байт) задает 64-битный всемирный идентификатор WWID (Worldwide Identifier), учетверенное слово 64 (8 байт) — номер логического устройства (LUN).
- I^2O — учетверенное слово 56 (8 байт) задает 64-битный тег идентификации IdentifyTag.

Остальные элементы пути к устройству зарезервированы (нулевые).

Расширенная таблица параметров диска DPTE (Device Parameter Table Extension) позволяет обращаться к дискам ATA в обход BIOS *INT 13h.* Ее формат приведен в табл. 3.11.

Таблица 3.11. Расширенная таблица параметров диска (DPTE)

Смещение	Длина	Назначение
0	2	Базовый адрес блока командных регистров (адрес регистра данных)
2	2	Базовый адрес блока управляющих регистров
4	1	Старшая тетрада для регистра номера устройства и головки: Биты 0-3=0 Бит 4 — выбор устройства ATA (бит DEV) Бит 5=1 Бит 6 — режим LBA Бит 7=1
5	1	Резерв для BIOS
6	1	Биты 0-3 — номер IRQ Биты 4-7=0
7	1	Счетчик блоков для команд множественного чтения
8	1	Биты 0-3 — номер канала DMA Биты 4-7 — тип DMA (согласно ATA-2)
9	1	Биты 0-3 — режим PIO Биты 4-7=0
10	2	Аппаратно-специфические опции: Бит 0 — разрешение скоростных режимов PIO (действителен байт 9) Бит 1 — разрешение использования DMA Бит 2 — разрешение мультисекторных операций Бит 3 — разрешена трансляция CHS (устройство имеет более 1024 цилиндров), тип трансляции в битах 9-10 Бит 4 — разрешена трансляция LBA (адрес из пакета расширенного сервиса непосредственно передается в устройство) Бит 5 — сменный носитель Бит 6 — устройство ATAPI Бит 7 — разрешен 32-битный режим обмена Бит 8 — сообщение о готовности принять пакет через запрос прерывания (для ATAPI) Биты 9-10 —тип трансляции: 00—с помощью сдвигов, 01 — с помощью LBA, 10 — резерв, 11 — специфическая схема Бит 11 — использование Ultra DMA Биты 12-15=0
12	2	Резерв(0)
14	1	Версия таблицы (11h)
15	1	Контрольная сумма (двоичное дополнение суммы байтов 0-14)

Функция 49h — получение информации о факте смены носителя (get extended media change status). При возврате CF=0 и AH=0 означает, что носитель не менялся; *CF*= 1 и *AH*=06 означает активность сигнала смены носителя (хотя носитель мог и не меняться, просто открыли и закрыли дверку).

Функция 4Eh — установка аппаратной конфигурации (set hardware configuration), в *AL* — подфункция:

0 — разрешение предвыборки (поиска);

1 — запрет предвыборки (поиска);

2 — установка максимально допустимого режима PIO;

3 — установка режима PIO 0;

4 — установка режима PIO, заданного в BIOS Setup;

5 — разрешение использования максимально возможного режима DMA для /Л/Т *13h;*

6 — запрет использования DMA для *INT 13h.*

При нормальном возврате *CF*=0, *AH*=0*, AL*=0*,* если команда выполнена, не затрагивая других устройств, *AL=1,* если команда влияет на другие устройства.

Функция 50h — посылка командного пакета (send packet command), AL=D7h, ES:SI указывает на пакет. Пакет начинается со слова, задающего его длину, за которым следует тело пакета, специфичное для данного типа интерфейса.

Возможные коды ошибок (при CF=1):

01 — функция не реализована;

80h — ошибка выполнения команды;

97h — подфункция D7h не поддерживается для данного устройства;
C3h — пакет слишком короткий.

3.2. Взаимодействие ОС и прикладных программ с дисковыми устройствами

3.2.1. Общие сведения.

Прямое назначение дисковых устройств в компьютере — хранение файлов. Прикладной программе могут потребоваться различные файловые функции: создать новый файл; записывать в него информацию, добавляя данные в конец или замещая в середине; читать данные, последовательно с начала или произвольно; определить размер файла; прочитать или изменить его атрибуты; удалить файл; найти в каталоге файлы, удовлетворяющие определенным условиям поиска; создать, удалить или переименовать каталог. Все эти функции оперируют элементами файловой системы — в случае FAT (см. п..2) это элементы каталогов и элементы таблицы размещения файлов. Поскольку файловая система является частью ОС, желательно, чтобы все программы при обращении к файлам пользовались сервисами, предоставляемыми ОС. Тогда ОС будет полновластной хозяйкой дискового пространства, и она сможет (если написана без ошибок) предоставлять различным программам доступ к файлам, корректно разрешая конфликтные ситуации.

Операционная система предоставляет прикладным программам *сервисы* различного уровня — вызовы API (Application Program Interface — интерфейс прикладных программ). В ранних версиях MS DOS файловые сервисы были очень неудобными—с файлами приходилось работать через 37-байтные структуры FCB (File Control Block — блок управления файлом). Эти сервисы вызывались по прерыванию *INT21h с* номерами функций (в регистре *AH)* 0F-16h, 21h, 22h, 24h, 27-29h. В FCB указывается номер диска, имя файла, его длина, текущая позиция в файле, дата и время создания или последней модификации (в специфическом формате). Блок FCB формировался операционной системой из текстовой строки вида D:FILENAME.TXT (по функции 29h). Файлы были доступны только из текущего каталога, для смены которого требовалось выполнение отдельной функции. Начиная с MS DOS 2.0 были введены новые, UNIX-подобные функции унифицированного доступа к файлам через 16-битные индексы открытых файлов (file handle). Когда приложению требуется начать работу с файлом, оно выдает запрос на открытие файла (функция 3Dh), указав на строку, содержащую полное имя файла вида [D:\][PATH\]FILENAME.EXT. В ответ ОС возвращает индекс, на который приложение будет ссылаться при дальнейших вызовах сервисов для этого файла. Унификация заключается в том, что приложения (и ОС) могут работать одинаково с файлами и последовательными устройствами ввода-вывода. Имеется пять стандартных устройств:

- 0000h — стандартный ввод (клавиатура);
- 000lh — стандартный вывод (экран в телетайпном режиме);
- 0002h — стандартный вывод сообщения об ошибках (экран);
- 0003h — стандартное внешнее устройство (AUX, по умолчанию COM1);
- 0004h — стандартная печать (PRN, по умолчанию LPT1).

Остальные значения индексов ОС назначает для открываемых файлов или специальных устройств ввода-вывода. Манипулирование номерами позволяет легко перенаправлять потоки данных. Например, консольная команда DIR > DIRLIST.TXT направит поток данных (список файлов в текущем каталоге) в файл DIRLIST.TXT вместо стандартного устройства вывода.

Данные сервисы вызываются тоже через прерывание *INT21h* и включают следующие функции:

- 36h — получение информации о свободном пространстве на логическом диске с указанным номером (0 — текущий диск, 1 — диск A:, 2 — диск B:...) в виде числа свободных кластеров, размера кластера (Число секторов) и размера сектора (в байтах);
- 39h — создание каталога (действие, выполняемое по консольной команде MKDIR или MD);
- 3Ah — удаление каталога (RMDIR или RD);
- 3Bh — установка текущего каталога по умолчанию (CHDIR или CD);
- 3Ch — создание файла (если файл уже существует, то все данные удаляются);
- 3Dh — открытие файла;
- 3Eh — закрытие файла;
- 3Fh — чтение из файла (или устройства);
- 40h — запись в файл (или устройство);
- 41h — удаление файла;
- 42h — перемещение указателя позиции чтения/записи (а также определение размера файла);
- 43h — чтение/установка атрибутов файла;
- 4400h — определение типа (файл или устройство);
- 4406h — проверка читаемого файла (входного устройства) на окончание файла (EOF);
- 4407h — проверка выходного устройства (записываемого файла) на занятость;

- 440Ah — проверка, находится ли файл на сетевом устройстве;
- 45h — дублирование индекса;
- 46h — перенаправление ввода-вывода;
- 4Eh — поиск в каталоге первого элемента, удовлетворяющего условиям поиска (с использованием метасимволов «*» и «?»);
- 4Fh — поиск следующего элемента, удовлетворяющего условиям поиска;
- 56h — переименование файла или/и перемещение в другой каталог того же диска, а также переименование (но не перемещение) каталога;
- 57h — чтение/установка даты и времени создания файла;
- 5Ah — создание файла с уникальным именем (например, как временный);
- 5Bh — создание нового файла;
- 5Ch — блокировка/разрешение доступа к области файла;
- 67h — установка (увеличение) максимального числа индексов;
- 68h — выгрузка буферов указанного файла на диск;
- 6Ch — расширенное создание и открытие файла.

Одновременно ОС может использовать ограниченное число индексов, из которых 5 заняты стандартными устройствами. Если требуется возможность одновременного открытия большего числа файлов, то в файле CONFIG.SYS указывают параметр FILES=N (N=8-255, по умолчанию 8). Если приложению требуется большее число индексов (их может использоваться и более одного для каждого открытого файла), то с помощью функции 67h оно (для данного приложения) может быть увеличено до 65 535, но при наличии свободной памяти. Хотя эта функция появилась еще в MS DOS 3.3, не все приложения умеют ею пользоваться, так что задавать параметр FILES приходится для многих приложений MS DOS. Заметим, что для очень старых приложений, работающих через FCB, для увеличения числа одновременно открытых файлов требуется в файле CONFIG.SYS задавать параметр FCBS=N (N= 1-255, по умолчанию 4).

Данные файловые функции поддерживаются и в Windows для приложений MS DOS. Заметим, что они работают только с «короткими» именами (по схеме «8.3»).

Для приложений Windows имеются свои вызовы API, работающие и с «длинными» именами, но здесь мы их рассматривать не будем.

Кроме файловых функций, ОС может предоставлять сервисы и более низкого уровня. Например, в MS DOS через *INT21h* можно вызвать набор функций, включающих установку/чтение параметров геометрии блочного устройства, чтение, запись, верификацию и форматирование трека, определение сменяемости носителя и некоторые другие функции. При вызове AX=440Dh, в *CL* указывается номер подфункции.

Кроме того, есть и функции абсолютного чтения (INT 25h) и записи (INT 26h) секторов логических дисков. Для этих функций доступны все секторы логического диска, начиная с загрузочного (его номер 0). Секторы с таблицами разделов, как и секторы, выходящие за пределы логического диска, по этим функциям недоступны.

Работа программы с файлами диска в обход сервисов ОС по чтению не криминальна, поскольку риск заключается лишь в том, что программа может «промахнуться» мимо требуемых данных, которые ОС у нее «из-под носа» сотрет или переместит в соответствии с легальными запросами. Легко понять, что запись на диск в обход ОС допустима далеко не всегда: распределение свободных кластеров, как и занятие свободных элементов каталогов, должно выполняться «из одних рук», чтобы два процесса не пытались одновременно выполнять запись в один и тот же элемент — это приведет к потере данных. Особенно остро эти проблемы встают в многозадачных ОС, и по этой причине попытки доступа к диску в обход ОС блокируются средствами защиты. Но и в однозадачных ОС, когда к файлам компьютера обеспечивается доступ по сети, возникают аналогичные проблемы. Многозадачная ОС должна отслеживать попытки приложений обращаться к дискам непосредственно — путем обращения к портам ввода-вывода дисковых контроллеров или вызовов сервисов BIOS. Процессоры x86, применяемые в IBM PC-совместимых компьютерах, начиная с 386-го, имеют аппаратные средства защиты, достаточные для такого контроля. Операционная система может фильтровать перехватываемые попытки доступа в обход ее интерфейсов API, позволяя исполнять безопасные запросы и блокируя, на ее взгляд, опасные, снимая приложения с сообщением «Приложение выполнило недопустимую операцию и будет закрыто».

Файловые сервисы операционной системы выполняют довольно сложные логические функции. Так, например, для простого чтения произвольного фрагмента файла, заданного при вызове смещением требуемого фрагмента и его длиной, требуется:

- найти начало цепочки кластеров (прочитав элемент каталога);
- «прошагать» по этой цепочке, отсчитав требуемое число пропускаемых кластеров (читая с диска и обрабатывая элементы FAT);
- найти сектор кластера, начиная с которого производить чтение;

- считать требуемое количество секторов (но не далее конца кластера);
- если требуется, продолжить чтение секторов следующих кластеров в цепочке;
- передать в буфер, указанный приложением, требуемый фрагмент, отсекая лишние данные в начале и конце, поскольку требуемый фрагмент может начинаться и заканчиваться в произвольном месте, а не на границе секторов.

ВНИМАНИЕ ———————————————————————————————

Попутно заметим, что когда выполняется запись произвольного фрагмента в конец файла, в «хвост» до границы сектора попадает «мусор», оставшийся в памяти, возможно, от предыдущих дисковых операций. Секторы до конца кластера обычно не переписываются. Так что в «хвостах» файлов (между «официальным» концом, определенным его длиной, и концом последнего сектора его последнего кластера) может оставаться информация, не всегда предназначенная для посторонних глаз. Посмотреть (и затереть) ее можно утилитами редактирования дисков, например, DISKEDIT.

Как видно, для выполнения одного файлового запроса операционной системе приходится выполнять серию различных обращений к диску. Операции записи еще сложнее, поскольку ОС приходится читать FAT для определения положения файла и для поиска свободных кластеров, чтобы записать продолжение файла и записывать модифицированные фрагменты FAT на диск. Во время каждого из обращений могут возникнуть ошибки, и в обязанности ОС входит обработка этих ошибок. При ошибках чтения делается некоторое число повторов, возможно, и со «встряхиванием диска» (поиском нулевого трека), и если они оказываются безуспешными, может появляться стандартное сообщение вида «Error reading file... Abort, Retry, Ignore?» со всем известными вытекающими последствиями. Ошибка при записи может вызвать и более сложные действия — пометку кластера со сбойным сектором в FAT и попытку записи блока на другое место (правда, чаще ОС ограничивается сообщением об ошибке и отказом выполнения запроса, а для пометки кластера как сбойного требуется выполнение специальной утилиты, например ScanDisk).

Операционная система в той или иной мере старается оптимизировать доступ к диску, минимизируя (в разумных пределах) фактические обращения к диску, поскольку они занимают значительное время. Проблема быстродействия дисковой памяти решается за счет оперативной памяти — применяется *кэширование дисков* — хранение образов последних из использованных блоков дисковой памяти в оперативной в надежде на то, что к ним вскоре будет следующий запрос, который удастся удовлетворить из памяти. Первым делом, конечно же, кэшируется таблица размещения файлов (FAT), поскольку обращение к ее элементам требуется для всех файловых операций. В идеале хорошо бы держать в памяти всю таблицу, но для больших дисков это потребовало бы слишком большого расхода оперативной памяти. Далее кэшируют области пользовательских данных, а также каталоги. В MS DOS «неглубокое» кэширование выполняет операционная система, которая имеет буферы для хранения считываемых и записываемых блоков. Если затребованный блок с диска уже находится в одном из буферов, ОС не будет «беспокоить» диск, а удовлетворит запрос из буфера. При записи ОС накапливает данные в буфере до тех пор, пока не заполнится сектор. Однако при закрытии файла, а также по функции 68h (см. выше) выгрузка буфера на диск производится немедленно. Процессом кэширования в буферах DOS можно управлять с помощью строки «BUFFERS=n[,m]» файла CONFIG.SYS. Параметр n (1-99) задает число буферов, которые может держать ОС в оперативной памяти. Необязательный параметр m (0-8, по умолчанию 0) задает число буферов, используемых для упреждающего чтения блоков, следующих за затребованными. Область памяти для буферов, естественно, уменьшает объем памяти, доступной программам. Значение n по умолчанию задается в зависимости от объема ОЗУ (для систем с 640К и более памяти n=15). Более глубоким кэшированием дисков занимается загружаемый драйвер SMARTDRV.EXE, без которого работа приложений с интенсивным дисковым обменом, а также оболочки Windows 3.x в среде MS DOS оказывалась невыносимо медленной. В Windows 9x/NT мощное кэширование встроено в операционную систему. Параметрами кэширования можно управлять: например, в Windows 9x можно управлять упреждающим чтением, а также оптимизировать систему для роли настольного компьютера, портативного компьютера или файл-сервера. В операционных системах, предназначенных для файл-серверов (Novell, Windows NT и др.), применяют мощные системы кэширования различных элементов диска. Они, как правило, являются основными потребителями оперативной памяти, отводимой под нужды ОС. Кроме кэширования средствами ОС (и дополнительных драйверов), в дисковой системе применяется автономное кэширование устройств, реализуемое в самих накопителях и их высокопроизводительных контроллерах.

На производительность исполнения файловых запросов в значительной мере влияет расположение файлов на диске. В первую очередь заметно влияние фрагментации файлов: если кластеры файла хаотично раскиданы по диску, то ОС для прохода по цепочке кластеров вынуждена считывать множество секторов FAT, и если не вся таблица отображается в ОЗУ, то только для определения текущей позиции может потребоваться множество обращений к диску. Для последовательного доступа к данным такого файла головкам диска придется «скакать» по всему диску, а время поиска дает самую большую

задержку во всем процессе обращения (в среднем 10 мс на случайном поиске). Нефрагментированный файл может быть считан всего за одно-два обращения к диску: считывание фрагмента FAT (если его нет в памяти) и одна многосекторная передача для считывания всех данных (если позволяет размер буфера). Реально, конечно же, большой файл зачастую считывается за несколько обращений к диску, но потери времени на позиционирование будут минимальными. Так что целесообразность регулярной дефрагментации файлов вполне очевидна. Следующий фактор, влияющий на производительность, — взаимное расположение файлов, которыми одновременно пользуется одно приложение. Даже если они и не фрагментированы, но расположены в разных областях диска, при чередующихся запросах на позиционирование головок будет тратиться значительное время. Оптимальным будет компактное расположение файлов, используемых каждым приложением. Но для такой оптимизации нужна статистика использования файлов различными приложениями, данными которой должна пользоваться оптимизирующая утилита. Этим занимается новый дефрагментатор из комплекта Windows 98.

Для нормальной работы кэширования система должна точно знать, возможна ли смена носителя во время сеанса работы ОС, и если возможна, корректно отрабатывать факт смены. Если носитель сменился, то данные из кэша недействительны и требуется считывание носителя. До изъятия носителя данные из буферов записи должны быть выгружены на физический носитель, иначе они будут утеряны. Именно из этих соображений введена классификация дисков на фиксированные и сменяемые, а у сменных дисков крайне желателен механизм уведомления о смене, а еще лучше управление механизмом смены по командам от ОС. Для дисководов, не поддерживающих смену носителя (5" 360 и 720 Кбайт), ОС при отработке каждого файлового запроса вынуждена пересчитывать FAT и требуемую область диска. В этом легко было убедиться (когда такие дисководы еще не были раритетом), если два раза подряд выдавать команду DIR для дисководов 360 и 1,44 (1,2) Мбайт, предварительно дожидаясь остановки двигателя. В первом случае вывод на экран будет задерживаться до раскрутки мотора и выполнения считываний оба раза. Во втором случае задержка будет только после первого обращения, а на второй раз вывод начнется одновременно с раскруткой мотора: ОС считает данные из буфера, но запустит мотор, чтобы возможная последующая операция не ждала раскрутки. Если вынуть и вставить дискету, то та же команда снова вызовет задержку — ОС снова будет читать FAT и требуемый каталог. Другой пример на ту же тему — неработающий датчик смены носителя у дисковода, для которого система подразумевает наличие датчика . В этом случае система считает FAT с первого установленного диска и будет ею пользоваться для всех дисков, которые будут установлены далее в этом сеансе работы. Кроме того, система будет показывать «фантомные» каталоги первого диска. Реально перечитать диск в этом сеансе ее заставит только комбинация Ctrl-Break и, возможно, закрытие, приложения, обратившегося к диску.

ВНИМАНИЕ ———————————————————————————————————

Если при работе в Windows возникло желание проверить на читаемость файлы, только что записанные на дискеты (или сравнить их с исходными), следует предварительно вынуть и вставить дискету. Иначе «читаться» будут данные из буфера в ОЗУ — это гораздо быстрее, но абсолютно бессмысленно.

Перед извлечением сменного носителя нужно позволить приложению, пишущему на этот диск, завершить операции записи — иначе данные будут потеряны или же приложение («устами» ОС) будет просить вставить диск с серийным номером xxxx-xxxx.

Перед выключением компьютера приложения (по крайней мере, пишущие на диск), а также операционная система (если это не MS DOS) должны быть закрыты, чтобы все данные из кэша записи были записаны на диск. Иначе возможна потеря данных, появление «бесхозных» Кластеров, потеря (несохранение) настроек и конфигураций.

С точки зрения надежности, ОС старается не задерживать физическую запись в FAT, поскольку здесь экономия времени может обернуться крупными потерями (1 сектор FAT при размере кластера в 32 Кбайт «стоит» восьми Мбайт пользовательских данных).

ОС взаимодействует с дисками посредством программных драйверов. MS DOS работает исключительно через сервисы BIOS, что обеспечивает хорошую совместимость ПО с «железом» компьютера. Диск с MS DOS и приложениями можно переносить с машины на машину без каких-либо изменений или настройки драйверов. Однако, как было показано выше, драйверы BIOS работают в 16-разрядном режиме процессора и недостаточно эффективны (зато универсальны). Многозадачные ОС (Windows 9x/NT, Unix, OS/2 и другие) используют собственные драйверы, которые привязаны к конкретным моделям контроллеров интерфейсов дисковых устройств. Расплатой за эффективность стала аппаратная зависимость конкретной инсталляции ОС, так что простой перенос винчестера с системой на машину с другим контроллером дискового интерфейса может потребовать перенастройки или смены драйверов. Правда, для интерфейса ATA в базовом варианте использования (без DMA и прямого управления шиной) все контроллеры имеют одинаковую программную модель (поскольку в этом режиме контроллер ATA «прозрачен»). Для эффективной работы дисковой системы следует устанавливать специальные драйверы, работающие в режиме UltraDMA (или просто DMA) с прямым управлением

шиной (bus mastering).

При запуске приложений ОС тоже обращается к файлам на диске: она загружает в память требуемый файл (.COM или .EXE) и передает управление в точку входа, определяемую типом файла. Запуск приложений через командные файлы (.BAT) или ярлыки добавляет дополнительную ступень обращения: чтение этого файла и определение имени файла (а в командном файле их последовательности), который следует загрузить для выполнения. Напомним, что файл для исполнения может задаваться как с полным путем, начинающимся с буквы — имени диска, так и с относительным или же вообще без пути. Если указано только имя, ОС ищет файл сначала в текущем каталоге, а затем в каталогах, перечисленных в переменной PATH файла AUTOEXEC.BAT. Если приложение задает имя файла без полного пути, то он ищется в текущем каталоге или относительно него, если в пути используются символы «..» (на один уровень вверх) и «\» (разделитель имен подкаталогов).

Кроме обеспечения хранения файлов данных и приложений, в чем диску помогает оперативная память, диски используются и как средство расширения оперативной памяти. Основной недостаток оперативной памяти заключается в том, что конструктивно достижимый ее объем во много раз меньше, чем у дисковой (пока что это было справедливо на всех ступенях технического прогресса). Решить проблему увеличения объема оперативной памяти за счет дисковой позволяет *виртуальная память,* которую можно считать кэшированием оперативной памяти на диске. Суть ее заключается в том, что программам предоставляется виртуальное пространство оперативной памяти, по размерам превышающее объем физически установленной оперативной памяти. Это виртуальное пространство разбито на страницы фиксированного размера, а в физической оперативной памяти в каждый момент времени присутствует только часть из них. Остальные страницы хранятся на диске, откуда операционная система может их «подкачать» в физическую на место предварительно выгруженных на диск страниц. Для прикладной программы этот процесс прозрачен (если только она не критична ко времени обращения к памяти). Для пользователя этот процесс заметен по работе диска даже в тот момент, когда не требуется обращение к файлам. Расплатой за почти безмерное увеличение объема доступной оперативной памяти является снижение средней производительности памяти и некоторый расход дисковой памяти на так называемый файл подкачки (swap file). Файлы подкачки могут располагаться на одном или нескольких дисках, их положением и размерами управляет ОС, и пользователю в этот процесс вмешиваться, как правило, не приходится. Естественно, размер виртуальной памяти не может превышать размера дисковой памяти. Виртуальная память реализуется операционными системами (и оболочками) защищенного режима (например, OS/2, MS Windows) на основе аппаратных средств процессоров класса не ниже 286, а наиболее эффективно — 32-разрядных процессоров 386 и старше.

3.2.2 Группа дисковых функций MS-DOS

Рассмотрим более подробно группу дисковых функций MS DOS, вызываемых с помощью программных прерываний INT 21h, INT 25h и INT 26h.

В эту группу входят прерывания, предназначенные для выполнения основных функций операционной системы, в том числе для выполнения операций с логическими дисками, файлами и каталогами. Дисковые функции DOS обладают достаточной полнотой и универсальностью для решения любых задач в реальном режиме DOS. Они могут применяться и в режиме линейной адресации памяти, но информацию в расширенную память приходится пересылать через промежуточный буфер в первом мегабайте адресного пространства процессора. Впрочем, дополнительные пересылки не особенно замедляют работу: поиск данных на диске и передача информации между диском и процессором занимает гораздо больше времени, чем копирование такого же объема данных с одного участка оперативной памяти в другой.

Ниже описаны функции DOS, выполняющие основные операции над логическими дисками, каталогами и файлами. При описании используются следующие термины:

- строка ASCIIZ — текстовая строка в ASCII-коде, которая завершается нулевым значением;
- дескриптор файла — уникальный номер, который операционная система присваивает создаваемому или открываемому файлу в качестве идентификатора (чтобы потом обращаться к файлу по этому номеру — вплоть до его закрытия).

3.2.2.1.Классические функции для работы с дисками

К этой группе относятся функции, появившиеся в ранних версиях операционной системы MS-DOS и сохранившиеся с тех пор практически без изменений. Такие функции отличаются крайне примитивной обработкой ошибок:

- в случае успешного завершения операции флаг CF сбрасывается в 0;
- в случае ошибки флаг CF устанавливается в 1.

Для обращения к дисковым функциям DOS используется прерывание Int 21h.

Прерывание Int 21h, функция 0Eh: сменить текущий логический диск

Функция позволяет выбрать логический диск. Перед вызовом прерывания требуется записать в регистры следующие значения:

- в AH — значение 0Eh;
- в AL — код логического диска (0 - A:, 1 - B: и т. д.).

После завершения операции функция возвращает в регистре AL максимально возможный в данной системе номер логического дисковода (определяется параметром LASTDRIVE в файле CONFIG.SYS).

Прерывание Int 21 h, функция 19h: определить номер текущего дисковода

Функция определяет номер дисковода, который в данный момент считается текущим, то есть используется по умолчанию.

Перед вызовом прерывания требуется записать в регистр AH значение 19h.

После завершения операции функция возвращает в регистре AL код логического диска (0 - A:, 1 - B: и т. д.).

Прерывание Int 21 h, функция 1Ah: изменить адрес области обмена с диском

Функция устанавливает адрес буфера, используемого в операциях ввода/вывода и поиска в каталогах.

Перед вызовом прерывания требуется записать в регистры следующие значения:

- в AH — значение 1Ah;
- в DS : DX — указатель на новый адрес буфера обмена DTA.

ПРИМЕЧАНИЕ

При запуске программы ее область DTA первоначально установлена по адресу PSP: 0080 h.

Прерывание Int 21 h, функция 2Fh: получить адрес области обмена с диском

Функция определяет текущий адрес буфера, используемого в операциях ввода/вывода и поиска в каталогах.

Перед вызовом прерывания требуется записать в регистр AH значение 2Fh.

После завершения операции функция возвращает в ES:BX - указатель на адрес буфера обмена DTA.

Прерывание Int 21 h, функция 36h: определить объем свободного места на диске

Функция определяет объем свободного места на заданном логическом диске. Перед вызовом прерывания требуется записать в регистры следующие значения:

- в AH — значение 36h;
- в AL — код логического диска (0 - A:, 1 - B: и т. д.).

В случае ошибки в регистре AX будет возвращен код 0FFFFh (недопустимый код логического диска).

В случае успешного завершения операции функция возвращает:

- в AX — число секторов в кластере;
- в BX — число свободных кластеров;
- в CX — число байтов в секторе;
- в DX — полное число кластеров на диске.

Объем свободного пространства определяется произведением содержимого регистров AX, BX и CX, а полный объем диска в байтах — произведением AX, CX и DX.

3.2.2.2. Улучшенные функции для работы с дисками

По мере развития MS-DOS в набор функций постоянно вносились дополнения, упрощающие выполнение тех или иных операций и улучшающие контроль за их выполнением. Для вызова фнункций данной группы также используется прерывание Int 21h.

Перечисленные ниже функции DOS имеют усовершенствованные средства контроля: в случае ошибки кроме установки флага CF выдают в регистре AX код ошибки, по которому можно определить причину ее возникновения. Возможные значения кодов ошибок приведены в табл. 3.12. Однако следует учитывать, что содержимое регистра AX в случае успешного завершения данных функций не сохраняется.

Прерывание Int 21 h, функция 39h: создать подкаталог

Функция создает в текущем дереве каталогов новый подкаталог. Перед вызовом прерывания требуется записать в регистры следующие значения:

- в AH — значение 39h;
- в DS: DX — указатель на полную спецификацию каталога в виде строки ASCIIZ (должны существовать все каталоги на заданном пути, кроме последнего; вызов функции завершается -ошибкой, если родительский каталог заполнен и является корневым).

Возможные коды ошибки: 03h, 05h.

Прерывание Int 21 h, функция 3Ah: удалить подкаталог

Функция удаляет указанный подкаталог. Перед вызовом прерывания требуется записать в регистры следующие значения:

- в AH — значение 3Ah;
- в DS: DX — указатель на полную спецификацию удаляемого каталога в виде строки ASCIIZ.

Возможные коды ошибки: 03h, 05 h, 06h, l0h.

ПРИМЕЧАНИЕ Каталог должен быть пустым, иначе выдается сообщение об ошибке. Поэтому перед удалением каталога нужно его очистить, удалив все имеющиеся в нем файлы и каталоги.

Таблица 3.12. Значения расширенных кодов ошибки

Код ошибки	Расшифровка кода
00h	Нет ошибки
01h	Неверный номер функции
02h	Файл не найден
03h	Путь не найден
04h	Открыто слишком много файлов (нет свободных дескрипторов)
05h	Доступ запрещен
06h	Недопустимый дескриптор
07h	Разрушен блок управления памятью
08h	Недостаточно памяти
09h	Недопустимый адрес блока памяти
0Ah	Ошибка окружения (длина больше 32 Кбайт)
0Bh	Недопустимый формат
0Ch	Недопустимый код доступа
0Dh	Недопустимые данные
0Eh	Неизвестное устройство
0Fh	Недопустимый дисковод
10h	Попытка удалить текущий каталог
11h	Устройство не то же самое
12h	Больше нет файлов
13h	Диск защищен от записи
14h	Неизвестное устройство
15h	Дисковод не готов
16h	Неизвестная команда
17h	При проверке CRC обнаружена ошибка данных
18h	Неверная длина структуры запроса
19h	Ошибка поиска (позиционирования)
1Ah	Неизвестный тип носителя (не DOS-диск)
1Bh	Сектор не найден
1Ch	В принтере нет бумаги
1Dh	Ошибка записи
1Eh	Ошибка чтения
1Fh	Общий сбой
20h	Нарушение разделения
21h	Нарушение запирания
22h	Недопустимая смена диска (в ES:DI будет находиться указатель на метку нужного тома в виде строки ASCIIZ)
23h	FCB недоступен
24h	Разделяемый буфер переполнен
25h	Несоответствие кодовой страницы
26h	Невозможно завершить операцию с файлом
27h	Недостаточно места на диске
28h - 31h	Зарезервированы
32h	Неподдерживаемый сетевой запрос
33h	Удаленный компьютер не слушает
34h	Дублирование имени в сети
35h	Не найдено сетевое имя
36h	Сеть занята
37h	Сетевое устройство больше не существует
38h	Исчерпан лимит команд NetBIOS

Таблица 3.12. Значения расширенных кодов ошибки (продолжение)

Код ошибки	Расшифровка кода
39h 3Ah 3Bh 3Ch 3Dh 3Eh 3Fh 40h	Аппаратная ошибка сетевого адаптера Неверный отклик из сети Неожиданная ошибка сети Несовместимый удаленный адаптер Очередь на печать заполнена Нет места для файла печати Файл печати удален Сетевое имя удалено
41h 42h 43h 44h 45h 46h 47h 48h	Доступ к сети невозможен Неверный тип сетевого устройства Сетевое имя не найдено Исчерпан лимит сетевых имен Исчерпан лимит сеанса работы NetBIOS Временная пауза Сетевой запрос не принят Приостановлено переназначение принтера или диска
49h - 4Fh 50h 51h 52h 53h 54h 55h	Зарезервированы Файл уже существует Зарезервирован Каталог не может быть создан Отказ по прерыванию Int24h (критическая ошибка) Слишком много переназначений Двойное перенаправление
56h 57h 58h 59h 5Ah	Недопустимый пароль Недопустимый параметр Ошибка сетевой записи Функция не поддерживается в сети Требуемый компонент системы не установлен

Прерывание Int 21 h, функция 3Bh: перейти в другой каталог

Функция создает в текущем дереве каталогов новый подкаталог. Перед вызовом прерывания требуется записать в регистры следующие значения:

- в AH — значение 3Bh;
- в DS: DX — указатель на полную спецификацию заданного каталога в виде строки ASCIIZ.

Возможный код ошибки: 03h.

Прерывание Int 21 h, функция 3Ch: создать файл

Функция создает файл для записи. Если файл уже существует, то его размер усекается до 0.

Перед вызовом прерывания требуется записать в регистры следующие значения:

- в AH — значение 3Ch;
- в CX — атрибуты создаваемого файла (табл. 3.13);
- в DS: DX — указатель на имя файла в виде строки ASCIIZ.

В случае успешного завершения операции функция возвращает в регистре AX дескриптор файла.

Возможные коды ошибки: 03 h, 04h, 05 h.

Прерывание Int 21 h, функция 3Dh: открыть существующий файл

Функция открывает файл для чтения, записи или дозаписи информации. Указатель при этом устанавливается в начало файла.

Перед вызовом прерывания требуется записать в регистры следующие значения:

- в AH — значение 3Dh;
- в AL — режимы доступа (табл. 3.14);
- в DS : DX — указатель на имя файла в виде строки ASCIIZ.

В случае успешного завершения операции функция возвращает в регистре AX дескриптор файла.

Возможные коды ошибки: 0lh, 02h, 03h, 04h, 05h, 0Ch.

Таблица 3 13.Формат слова атрибутов файла

Номер	Описание
0	Только для чтения
1	Скрытый
2	Системный
3	Метка тома (разряд может быть установлен только при считывании атрибутов; при создании файла или изменении атрибутов значение разряда должно быть равно 0)
4	Каталог (разряд может быть установлен только при считывании атрибутов; при создании файла или изменении атрибутов значение разряда должно быть равно 0)
5	Признак архивации
6-15	Зарезервированы, должны быть равны 0

Таблица 3 14. Формат байта режимов доступа

Номер	Описание
0-2	Режим доступа: 000 - только для чтения; 001 - только для записи; 010 - для чтения и записи
3	Зарезервирован, должен быть равен 0
4-6	Режим разделения файлов: 000 - режим совместимости; 001 - другим программам запрещен любой доступ к файлу; 010 - другим программам запрещена запись в файл; 011 - другим программам запрещено чтение из файла; 001 - другим программам разрешен полный доступ к файлу
7	Флаг наследования: 0 - дочерний процесс наследует дескриптор, 1 - не наследует

Прерывание Int 21 h, функция 3Eh: закрыть файл

Функция сбрасывает на диск содержимое всех буферов, обновляет информацию в каталоге, а затем освобождает дескриптор файла (после этого дескриптор может быть присвоен другому файлу).

Перед вызовом прерывания требуется записать в регистры следующие значения:

- в AH — значение 3Eh;
- в BX — дескриптор файла.

Возможный код ошибки: 06h.

Прерывание Int 21 h, функция 3Fh: чтение информации из файла

Функция считывает данные из файла в указанный буфер. Считывание начинается с текущей позиции указателя файла.

Перед вызовом прерывания требуется записать в регистры следующие значения:

- в AH — значение 3Fh;
- в BX — дескриптор файла;
- в CX — число байтов, подлежащих считыванию;
- в DS: DX — указатель на буфер, в который должна быть занесена считанная информация.

В случае успешного завершения операции функция возвращает в регистре AX число реально считанных байт (оно может быть меньше значения, указанного в регистре CX при вызове функции, если в процессе считывания достигнут конец файла).

Возможные коды ошибки: 05h, 06h.

Прерывание Int 21 h, функция 40h: запись информации в файл

Функция записывает данные из указанного буфера в файл. Запись начинается с текущей позиции указателя файла.

Перед вызовом прерывания требуется записать в регистры следующие значения:

- в AH — значение 40h;
- в BX — дескриптор файла;
- в CX — число байтов, подлежащих записи;
- в DS: DX — указатель на буфер, информация из которого записывается в файл.

В случае успешного завершения операции функция возвращает в регистре AX число реально записанных байт (оно может быть меньше значения, указанного в регистре CX при вызове функции, если на диске недостаточно свободного места, то есть произошло переполнение).

Возможные коды ошибки: 05h, 06h.

Прерывание Int 21 h, функция 41 h: удалить файл

Удаляет указанный файл. Перед вызовом прерывания требуется записать в регистры следующие значения:

- в AH — значение 41h;
- в DS: DX — указатель на имя файла в виде строки ASCIIZ.

Возможные коды ошибки: 02h,03h,05h.

ПРИМЕЧАНИЕ Функция не допускает групповых операций и не удаляет файлы, имеющие атрибут «только для чтения» (этот атрибут нужно изменить при помощи функции 4 3 h, чтобы файл можно было удалить).

Прерывание Int 21 h, функция 42h: изменить положение указателя файла

Функция смещает указатель на заданное число байтов.

Перед вызовом прерывания требуется записать в регистры следующие значения:

- в AH — значение 42h;
- в AL — код точки отсчета положения указателя (0 — отсчет ведется от начала файла, 1 — от текущего положения указателя, 2 — от конца файла);
- в BX — дескриптор файла;
- в CX — старшая часть смещения;
- в DX — младшая часть смещения.

В случае успешного завершения операции функция возвращает в регистрах DX и AX новое положение указателя относительно начала файла:

- в DX — старшая часть значения положения указателя;
- в AX — младшая часть значения. Возможные коды ошибки: 0lh, 06h.

Прерывание Int 21 h, функция 43h, подфункция 00h: получить атрибуты файла

Подфункция определяет атрибуты указанного файла. Перед вызовом прерывания требуется записать в регистры следующие значения:

- в AX — значение 4300h;
- в DS : DX — указатель на имя файла в виде строки ASCIIZ.

В случае успешного завершения операции функция возвращает в регистре CX слово атрибутов файла (см. табл. 3.13).

Возможные коды ошибки: 0lh, 02h, 03h, 65h.

Прерывание Int 21 h, функция 43h, подфункция 01h: изменить атрибуты файла

Подфункция изменяет атрибуты указанного файла. Перед вызовом прерывания требуется записать в регистры следующие значения:

- в AX — значение 4301h;
- в CX — новые значения атрибутов файла (см. табл. 3.13);
- в DS : DX — указатель на имя файла в виде строки ASCIIZ.

Возможные коды ошибки: 0lh, 02h, 03h, 05h.

ПРИМЕЧАНИЕ Функция не может изменить атрибуты метки тома или каталога.

Прерывание Int 21 h, функция 47h: определить имя текущего каталога на указанном устройстве

Выдает имя текущего (рабочего) каталога на указанном логическом диске. Перед вызовом прерывания требуется записать в регистры следующие значения:

- в AH — значение 47h;
- в DL — код логического диска (0 — диск, используемый по умолчанию (текущий), 1 — диск «A:», 2 — диск «B:» и т. д.);
- в DS:SI — указатель на буфер размером 64 байта, выделенный для записи имени каталога.

В случае успешного выполнения функции в буфер будет записан путь от корневого каталога до текущего в виде строки ASCIIZ. Описание пути не включает в себя идентификатор диска и начальный обратный слэш «\».

Возможный код ошибки: 0Fh.

Прерывание Int 21 h, функция 4Eh: найти первый файл заданного типа

Функция ищет в указанном каталоге первый файл, соответствующий заданной спецификации.

Перед вызовом прерывания требуется записать в регистры следующие значения:

- в AH — значение 4Eh;
- в CX — маску атрибутов файла (см. табл. 3.13);
- в DS : DX — указатель на спецификацию файла в виде строки ASCIIZ (спецификация может включать в себя путь и шаблоны).

В случае успешного завершения функция возвращает в области DTA блок данных для первого найденного файла. Формат блока данных приведен в табл. 3.15..

Возможные коды ошибки: 02h, 03h, 12h.

ПРИМЕЧАНИЕ Получить адрес области обмена с диском DTA можно при помощи функции 2F h.

Таблица 3.15. Формат блока данных файла для функций поиска

Смещение	Размер элемента	Описание
00h	BYTE	Буква логического диска
0lh	11 байт	Шаблон для поиска
0Ch	BYTE	Атрибуты поиска
0Dh	WORD	Счетчик элементов внутри каталога
0Fh	WORD	Номер кластера начала родительского каталога
11h	4 байта	Зарезервировано
15h	BYTE	Атрибуты найденного файла
16h	WORD	Время создания файла:
		биты 0-4 - двухсекундные приращения:
		биты 5-10-минуты;
		биты 11-15-часы
18h	WORD	Дата создания файла:
		биты 0-4 - день;
		биты 5-8 - месяц;
		биты 9-15 - номер года (относительно 1980 г.)
lAh	DWORD	Размер файла в байтах
lEh	3 байтов	Имя и расширение файла в виде строки ASCIIZ

Прерывание Int 21 h, функция 4Fh: найти следующий файл

Функция следующие файлы, соответствующие спецификации, заданной при предшествующем вызове функции 4Eh.

Перед вызовом прерывания требуется записать в регистр AH значение 4Fh. В области DTA должен находиться блок данных от предыдущего вызова функции 4Eh или 4Fh.

В случае успешного завершения функция возвращает в области DTA блок данных очередного найденного файла (см. табл. 3.15).

Возможный код ошибки: 12h.

Прерывание Int 21 h, функция 56h: переименовать или переместить файл

Функция изменяет текущее имя файла на заданное и может выполнить при этом перемещение файла из одного каталога в другой (в пределах одного логического диска). Возможно также переименование (но не перемещение) каталогов. Перед вызовом прерывания требуется записать в регистры следующие значения:

- в AH — значение 56h;
- в DS: DX — указатель на спецификацию существующего файла в виде строки ASCIIZ (спецификация может включать в себя путь, но не должна включать шаблоны групповых операций);
- в ES: BX — новое имя файла в виде строки ASCIIZ (может включать в себя путь, но не должна включать шаблоны).

Возможные коды ошибки: 02h, 03h, 05h, 11h.

Прерывание Int 21 h, функция 57h, подфункция 00h: получить время и дату создания файла

Считывает время и дату создания файла с заданным дескриптором. Перед вызовом прерывания требуется записать в регистры следующие значения:

- в AX — значение 5780h;
- в BX — дескриптор файла.

В случае успешного завершения операции функция возвращает в регистрах следующие значения:

- в CX — время создания файла:
  - биты 0-4 — двухсекундные приращения;
  - биты 5-10 — минуты;
  - биты 11-15 — часы;
- в DX — дату создания файла:
  - биты 0-4 — день;
  - биты 5-8 — месяц;
  - биты 9-15 — номер года (относительно 1980 г.).

Возможные коды ошибки: 0lh, 06h.

Прерывание Int 21 h, функция 57h, подфункция 01 h: изменить время и дату создания файла

Изменяет время и дату создания файла с заданным дескриптором. Перед вызовом прерывания требуется записать в регистры следующие значения:

- в AX — значение 5701h;
- в BX — дескриптор файла;
- в CX — время создания *файла:*
  - биты 0-4 — двухсекундные приращения;

- биты 5-10 — минуты;
- биты 11-15 — часы;

- в DX — дату создания файла:
  - биты 0-4 — день;
  - биты 5-8 — месяц;
  - биты 9-15 — номер года (относительно 1980 г.).

Возможные коды ошибки: 01h, 06h.

Прерывание Int 21 h, функция 59h: получить дополнительную информацию об ошибке

Эту функцию (при необходимости) следует вызывать сразу после получения сообщения о возникновении ошибки при выполнении какой-либо операции по прерыванию Int 21h. Функция выдает дополнительную информацию о причинах возникновения ошибки и рекомендации по ее устранению.

ПРИМЕЧАНИЕ При выдаче кода ошибки обычно он упрощается до старого набора кодов DOS 2.x (коды 01h-12h),а данная функция выдает уточненный (расширенный) код.

Перед вызовом прерывания требуется записать в регистры следующие значения:

- в AH — значение 59h;
- в BX — значение 0.

Функция возвращает в регистрах следующие значения:

- в AX — расширенный код ошибки (см. табл. 3.12.);
- в BH — класс ошибки (табл. 3.16.);
- в BL — код рекомендуемого действия (табл. 3.17.);
- в CH — местоположение ошибки (табл. 3.18.);

ВНИМАНИЕ

В результате выполнения функции будут разрушены регистры CL,DX,SI,DI,BP,DSHES, поэтому их содержимое перед вызовом данной функции следует сохранить.

Таблица 3.16. Значения кодов класса ошибок

Код класса	Значение
0lh	Нехватка ресурсов (недостаточно места на диске или нет свободных каналов ввода/вывода)
02h	Доступ к файлу временно запрещен (с ним работает другая программа)
03h	Доступ запрещен - у программы нет необходимых полномочий для работы
04h	Ошибка системной программы
05h	Отказ оборудования
06h	Отказ системы (разрушен файл конфигурации)
07h	Ошибка прикладной программы
08h	Не найдено
09h	Неверный формат
0Ah	Заблокировано
0Bh	Ошибка носителя информации
0Ch	Уже существует
0Dh	Неизвестно

Таблица 3.17. Значения кодов рекомендуемых действий

Код действия	Рекомендуемая реакция на ошибку
01h	Повторить несколько раз и в случае неудачи предложить пользователю либо прервать операцию, либо проигнорировать ошибку
02h	Повторить несколько раз с задержкой между повторами и в случае неудачи предложить пользователю либо прервать операцию, либо проигнорировать ошибку
03h	Запрос пользователю для повторного ввода информации (обычно необходимо в случае некорректно заданного имени файла или диска)
04h	Прервать программу после освобождения используемых ресурсов
05h	Немедленно прервать программу
06h	Игнорировать ошибку
07h	Повтор после вмешательства пользователя, который должен устранить причину ошибки

Таблица 3.18. Значения кодов местоположения ошибок

Код	Местоположение ошибки
01h	Неизвестно
02h	Диск
03h	Сеть
04h	Устройство последовательного доступа
05h	Память

Прерывание Int 21 h, функция 5Ah: открыть существующий файл

Открывает файл для чтения, записи или дозаписи информации. Указатель при этом устанавливается в начало файла.

Перед вызовом прерывания требуется записать в регистры следующие значения:

- в AH — значение 5Ah;
- в CX — атрибуты файла (см. табл. 3.13);
- в DS: DX — указатель на путь в виде строки ASCIIZ (в конце строки должно находиться на менее 13 байт, заполненных нулями для получения сгенерированного имени).

В случае успешного завершения операции функция возвращает в регистре AX дескриптор файла, открытого для чтения и записи в режиме совместимости; DS: DX при этом будет указывать на путь, дополненный сгенерированным именем файла.

Возможные коды ошибки: 03h, 04h, 05h.

Прерывание Int 21h, функция 5Bh: создать новый файл

Создает файл для записи. От функции 3Ch отличается тем, что завершается ошибкой, если файл с таким именем уже существует.

Перед вызовом прерывания требуется записать в регистры следующие значения:

- в AH — значение 5Bh;
- в CX — атрибуты создаваемого файла (см. табл. 3.13);
- в DS: DX — указатель на имя файла в виде строки ASCIIZ.

В случае успешного завершения операции функция возвращает в регистре AX дескриптор файла.

Возможные коды ошибки: 03h, 04h, 05h, 50h.

3.2.2.3. Низкоуровневые дисковые функции DOS

К этой группе относятся функции, обеспечивающие непосредственную работу с секторами данных на логических дисках. Применяются они либо для ускорения ввода/вывода данных в системах управления реального времени, либо в специальных программах, предназначенных для восстановления удаленных файлов или исправления повреждений в файловой структуре диска.

Каждая из функций данной группы имеет собственный номер прерывания. При выполнении этих прерываний разрушается содержимое всех регистров общего назначения, сохраняется только содержимое сегментных регистров. После выполнения функции необходимо также извлечь из стека содержимое регистра флагов, помещенное туда при вызове прерывания.

В случае ошибки все функции этой группы устанавливают флаг CF и возвращают в регистре AL код ошибки (см. табл. 3.12), а в регистре AH — код состояния устройства (табл. 3.19).

Таблица 3.19. Коды состояния устройства

Код	Состояние
01h	Неверная команда
02h	Дефектная адресная метка
03h	Диск защищен от записи
04h	Сектор не найден
08h	Сбой DMA
10h	Ошибка данных - неправильная контрольная сумма (CRC)
20h	Сбой контроллера
40h	Сбой при выполнении поиска
80h	Устройство не отвечает

Прерывание int 25h: абсолютное чтение секторов из разделов малого объема

Прерывание используется для чтения секторов с диска по заданным логическим номерам. Объем раздела не должен превышать 32 Мбайт.

Перед вызовом прерывания требуется записать в регистры следующие значения:

- в AL — код логического диска (0 — «A:», 1 — «B:» и т. д.);
- в CX — число считываемых секторов;
- в DX — номер начального сектора;

- в DS: BX — адрес буфера для размещения данных.

В случае успешного выполнения операции в буфер будут записаны данные, прочитанные с диска.

Прерывание Int 25h, функция FFFFh: абсолютное чтение секторов из разделов большого объема

Прерывание используется для чтения секторов с диска по заданным логическим номерам. Читает из разделов, имеющих объем более 32 Мбайт.

Таблица 3.20. Блок параметров операции абсолютного чтения

Смещение	Размер	Описание
00h	DWORD	Номер сектора
04h	WORD	Число считываемых секторов
06h	DWORD	Дальний (far) указатель на буфер данных

Перед вызовом прерывания требуется записать в регистры следующие значения:

- в AL — код логического диска (0 — A:, 1 — B: и т. д.);
- в CX — значение 0FFFFh;
- в DS: BX — адрес блока параметров операции абсолютного чтения (табл. 3.20).

В случае успешного выполнения операции в буфер, адрес которого указан в блоке параметров, будут записаны данные, прочитанные с диска.

Прерывание Int 26h: абсолютная запись секторов в разделы малого объема

Используется для записи секторов на диск по заданным логическим номерам. Объем раздела не должен превышать 32 Мбайт.

Перед вызовом прерывания требуется записать в регистры следующие значения:

- в AL — код логического диска (0 — A:, 1 — B: и т. д.);
- в CX -- число записываемых секторов;
- в DX — номер начального сектора;
- в DS: BX — адрес буфера, содержащего записываемые данные.

Прерывание Int 26h, функция FFFFh: абсолютная запись секторов в разделы большого объема

Используется для записи секторов на диск по заданным логическим номерам. Читает из разделов, имеющих объем более 32 Мбайт.

Перед вызовом прерывания требуется записать в регистры следующие значения:

- в AL — код логического диска (0 — A:, 1 — B: и т. д.);
- в CX — значение 0FFFFh;
- в DS: BX — адрес блока параметров операции абсолютной записи (формат блока приведен в табл. 3.21).

Таблица 3.21. Блок параметров операции абсолютной записи

Смещение	Размер	Описание
00h	DWORD	Номер сектора
04h	WORD	Число записываемых секторов
06h	DWORD	Дальний (far) указатель на буфер данных

3.2.2.4. Примеры использования функций DOS

В прикладных программах чаще всего применяются функции записи и считывания файлов. Программа SaveRusFont, приведенная на листинге 3.1, записывает в текущий каталог диска текущий шрифт MS-DOS, полученный из памяти видеоконтроллера, в двоичном представлении (файл fontO.fnt) и в виде инверсного изображения в формате BMP (файл fontO.bmp). Она является упрощенным вариантом программы формирования изображения стандартных американского и русского шрифтов MS-DOS.

Кроме описанных в предшествующих главах процедур общего назначения (их описание дано в разделе "Кпавиатура"), в программе SaveRusFont используются приведенные в том же листинге 3.1 вспомогательные подпрограммы, выполняющие следующие функции:

• процедура GrabRusFont выполняет операцию считывания шрифта из памяти видеоконтроллера (она уже была описана в главе 4 «Видеоконтроллеры»);

• процедура ShowRusFont отображает шрифт на экран в стандартном 256-цвет-ном режиме VGA с разрешением 320x200;

• процедура CreateBMPFile создает на диске файл и записывает в него заголовок формата BMP;

• процедура WnteRasterStnng служит для записи в файл очередной строки изображения в формате BMP RGB;

• процедура CloseBMPF-ile завершает операцию записи файла изображения на. диск.

Листинг 3.1. Запись русского шрифта на диск в двоичном формате и в виде инверсного изображения в формате BMP RGB

```
IDEAL
P386
LOCALS
MODEL MEDIUM

; Подключить файл мнемонических обозначений
; кодов управляющих клавиш
include "lst_1_03.inc"
; Подключить файл мнемонических обозначений цветов
include "lst_1_05.inc"

SEGMENT sseg para stack 'STACK'
        DB 400h DUP(?)
ENDS

DATASEG
; Текстовые сообщения
Txt1 DB LIGHTCYAN,0,23
     DB "ЗАПИСЬ НА ДИСК РУССКОГО ШРИФТА DOS",0
     DB LIGHTGREEN,12,20
     DB "Запись шрифта в двоичный файл FONT0.FNT",0
     DB YELLOW,24,35,"Ждите ...",0
Txt2 DB LIGHTGREEN,11,20
     DB "Запись шрифта в файл BMP-тила FONT0.BMP",0
     DB LIGHTGREEN,13,18
     DB "Будет произведен переход в графический режим",0
     DB YELLOW,24,24,"Нажмите любую клавишу и ждите",0
Txt3 DB LIGHTGREEN,12,28,"Запись шрифта завершена",0
     DB YELLOW,24,29,"Нажмите любую клавишу",0
; Буфер для сохранения шрифта (16x256 байт)
Font8x16 DB 4096 DUP(?)
; Позиция отображаемого символа
FontString DW ? ;номер строки шрифта
FontColumn DW ? ;номер колонки шрифта
; Имя для файла BMP
BMPFileName DB 'font0.bmp',0 ;имя для двоичного файла
; Кодовый номер файла
FileHandle DW 0
; Заголовок файла BMP
BMPStruct DB 'BM'    ;сигнатура
          DD 153654 ;размер файла в байтах
          DW 0      ;резерв
          DW 0      ;резерв
          DD 54     ;смещение области данных
          DD 40     ;размер описателя изображения
          DD 320    ;ширина изображения в пикселях
          DD 160    ;высота изображения в пикселях
          DW 1      ;число битовых плоскостей
          DW 24     ;число битов на пиксель
          DD 0      ;метод сжатия
          DD 153600 ;размер изображения в байтах
          DD 0      ;разрешение по горизонтали
          DD 0      ;разрешение по вертикали
          DD 0      ;число цветов в изображении
          DD 0      ;число важных цветов изображения
; Одна строка изображения в BMP-формате
BMPData    DB 960 DUP(?)
; Номер выводимой строки изображения
CurrentString DW ?
ENDS

CODESEG
PROC SaveRusFont
        mov     AX,DGROUP
        mov     DS,AX
; Считать шрифт из видеопамяти
        call    GrabRusFont
; Установить текстовый режим и очистить экран
        mov     AX,3
        int     10h
; Скрыть курсор – убрать за нижнюю границу экрана
        mov     [ScreenString],25
        mov     [ScreenColumn],0
        call    SetCursorPosition
```

```
; Предупредить пользователя о начале продолжительной
; по времени операции записи файла BMP-типа на диск
        call    ClearScreen
        mov     SI,offset Txt1
        call    ShowColorString
        mov     SI,offset Txt2
        call    ShowColorString
        call    ShowColorString
        call    ShowColorString
        call    GetChar
; Установить видеорежим VGA 320x200, 256 цветов
        mov     AX,13h
        int     10h
; Для наглядности отобразить шрифт на экране
; (8 строк по 32 символа)
        call    ShowRusFont

; Создать файл font0.bmp и записать заголовок файла
        call    CreateBMPFile
; Вывести изображение на диск построчно
        ; Вывод начинается с НИЖНЕЙ строки рисунка
        mov     [CurrentString],160
        mov     AX,0A000h

; Записать шрифт на диск в двоичном формате
        ; Вывести текстовые сообщения на экран
        mov     SI,offset Txt1
        call    ShowColorString
        call    ShowColorString
        call    ShowColorString
        ; Записать шрифт в файл font0.fnt
        mov     DX,offset Font8x16
        call    WriteFontFile

        mov     ES,AX
; Цикл вывода строк
@@P0:   ; Вычислить координаты начала экранной строки
        mov     AX,320
        mov     DX,[CurrentString]
        dec     DX
        mul     DX
        ; В SI записать указатель на строку экрана
        mov     SI,AX
        ; В DI записать указатель на строку BMP
        mov     DI,offset BMPData
; Сформировать строку BMP с инверсией цвета точек
        mov     CX,320      ;счетчик пикселей
@@P1:   mov     AL,[ES:SI]
        cmp     AL,0
        je      @@P2        ;на экране черная точка?
        ; Записать в строку BMP черную точку
        mov     [word ptr DI],0
        add     DI,2
        mov     [byte ptr DI],0
        inc     DI
        jmp short @@P3
        ; Записать в строку BMP белую точку
@@P2:   mov     [word ptr DI],0FFFFh
        add     DI,2
        mov     [byte ptr DI],0FFh
        inc     DI
@@P3:   inc     SI
        loop    @@P1
        ; Записать строку BMP-файла на диск
        call    WriteRasterString
        ; Перейти на строку вверх
        dec     [CurrentString]
        jnz     @@P0
; Завершить запись BMP-файла
        call    CloseBMPFile

; Переустановить текстовый режим
        mov     ax,3
        int     10h
; Скрыть курсор – убрать за нижнюю границу экрана
        call    SetCursorPosition
```

```
; Вывести сообщение о завершении записи шрифта
        mov     SI,offset Txt1
        call    ShowColorString
        mov     SI,offset Txt3
        call    ShowColorString
        call    ShowColorString
        call    GetChar
@@End:  ; Переустановить текстовый режим
        mov     ax,3
        int     10h
        ; Выход в DOS
        mov     AH,4Ch
        int     21h
ENDP SaveRusFont

;*************************************************
;* СЧИТЫВАНИЕ РУССКОГО ШРИФТА ИЗ ВИДЕОКОНТРОЛЛЕРА *
;*************************************************
PROC GrabRusFont near
        pushad
; Перепрограммировать синхронизатор
        cli
        mov     DX,3C4h
        ; Установить последовательную адресацию
        ; ячеек видеопамяти
        mov     AX,0704h
        out     DX,AX
        sti
; Перепрограммировать графический контроллер
        mov     DX,3CEh
        ; Выбрать для считывания плоскость 2
;*************************************************
;* ОТОБРАЗИТЬ ШРИФТ НА ЭКРАНЕ В РЕЖИМЕ 320 X 200 *
;*************************************************
PROC ShowRusFont near
        pusha
        push    ES
        mov     AX,0A000h
        mov     ES,AX
        mov     SI,offset Font8x16
        xor     DI,DI
        mov     [FontString],0
        mov     DL,0    ;цвет фона символа
        mov     DH,15   ;цвет символа
@@m0:   mov     [FontColumn],0
        push    DI
@@m1:   ; Отобразить очередной символ
        mov     AH,16   ;число строк в маске символа
@@m2:   ; Отобразить строку изображения символа
        mov     AL,[SI] ;загрузить очередной байт маски
        mov     CX,8
@@m3:   ; Вывести на экран очередную точку изображения
        mov     AX,0204h
        out     DX,AX
        ; Запретить четную-нечетную адресацию
        mov     AX,0005h
        out     DX,AX
        ; Установить окно доступа по адресу A0000h
        mov     AX,0006h
        out     DX,AX

; Скопировать шрифт в буфер Font8x16
        mov     AX,0A000h
        mov     ES,AX
        mov     SI,0
        mov     BX,offset Font8x16
        mov     DX,256
@@M0:   mov     CX,16
@@M1:   mov     AL,[ES:SI]
        mov     [BX],AL
        inc     BX
        inc     SI
        loop    @@M1
        add     SI,16
        dec     DX
        jnz     @@M0
        popad
        ret
ENDP GrabRusFont
```

```
        rol     AL,1
        jc      @@m4
        mov     [ES:DI],DL
        jmp short @@m5
@@m4:   mov     [ES:DI],DH
@@m5:   inc     DI
        loop    @@m3
        inc     SI
        add     DI,320-8
        dec     AH
        jnz     @@m2
        sub     DI,320*16-8-2
        inc     [FontColumn]
        cmp     [FontColumn],32
        jb      @@m1
        pop     DI
        add     DI,320*(16+4)
        inc     [FontString]
        cmp     [FontString],8
        jb      @@m0
        pop     ES
        popa
        ret
ENDP ShowRusFont

;***********************************************
;* ОТКРЫТЬ BMP-ФАЙЛ И ЗАПИСАТЬ ЕГО ЗАГОЛОВОК *
;***********************************************
PROC CreateBMPFile NEAR
        pusha
        ; Создать файл для записи
        mov     AH,3Ch
        mov     CX,0
        mov     DX,offset BMPFileName
        int     21h
        mov     [FileHandle],AX
        ; Записать заголовок BMP-файла
        mov     BX,[FileHandle]
        mov     CX,54
        mov     DX,offset BMPStruct
        mov     AH,40h
        int     21h
        popa
        ret
ENDP CreateBMPFile

;*****************************************************
;* ЗАПИСАТЬ СТРОКУ РАСТРОВОГО ИЗОБРАЖЕНИЯ В BMP-ФАЙЛ *
;*****************************************************
PROC WriteRasterString NEAR
        pusha
        ; Записать блок данных
        mov     BX,[FileHandle]
        mov     CX,960
        mov     DX,offset BMPData
        mov     AH,40h
        int     21h
        popa
        ret
ENDP WriteRasterString
;*********************
;* ЗАКРЫТЬ BMP-ФАЙЛ *
;*********************
PROC CloseBMPFile NEAR
        pusha
        ; Закрыть файл
        mov     AH,3Eh
        mov     BX,[FileHandle]
        int     21H
        popa
        ret
```

```
ENDP CloseBMPFile
ENDS

; Подключить процедуры вывода данных на экран
include "lst_1_02.inc"
; Подключить процедуры чтения/записи шрифта на диск
include "lst_6_02.inc"

END
```

Стандартный графический режим VGA с номером 13 h используется в программе для большей наглядности. На самом деле изображение может быть сформировано в любом свободном участке оперативной памяти и отображать его на экран совершенно не обязательно, хотя и полезно при отладке программы. Формирование и вывод в файл строк изображения выполняется поочередно: это простой, экономичный (с точки зрения использования оперативной памяти), но довольно неэффективный способ работы: запись на диск осуществляется крайне медленно. При формировании строки изображения цвет инвертируется (черный цвет заменяется белым и наоборот), а затем вместо кода цвета подставляется реальное значение яркостей цветовых компонент режима RGB: 0, 0, 0 для черного и 0FFh, 0FFh, 0FFh для белого цвета.

Процедура записи шрифта в файл в простом двоичном коде ReadFontFile и процедура считывания шрифта из файла WriteFontFile будут применяться в последующих примерах, поэтому они выделены в отдельный модуль, приведенный в листинге 3.2. Программа SaveRusFont из листинга 3.1 использует только процедуру записи шрифта.

Процедуры из листинга 3.2 записывают и считывают файл целиком, за один этап. Такой способ гораздо эффективнее, чем работа по кусочкам, однако он не всегда возможен: размер файла передается через регистр CX и потому ограничен предельным значением 16-разрядного числа (65 535). Если необходимо при помощи функций DOS обрабатывать файл большого объема, то делать это неизбежно приходится по частям. Поскольку при работе с дисками минимальной адресуемой единицей для операционной системы является кластер, то размер считываемого или передаваемого за одну операцию участка файла с целью ускорения работы системы выравнивается на максимально возможный размер кластера (32 768 байт).

Листинг 3.2. Запись шрифта в файл и считывание шрифта из файла в двоичном формате

```
DATASEG
; Текстовые сообщения
WrtFontErr DB LIGHTRED,12,23
           DB "Ошибка при записи файла font0.fnt",0
NoFontFile DB LIGHTRED,12,16,"Ошибка: не удается "
           DB "открыть файл шрифта font0.fnt",0
PrKbButton DB YELLOW,24,29,"Нажмите любую клавишу",0
; Имя для двоичного файла, содержащего шрифт
BinFileName DB 'font0.fnt',0
ENDS

CODESEG
;****************************************
;*   ЗАПИСАТЬ ШРИФТ В ДВОИЧНЫЙ ФАЙЛ     *
;* Параметры:                           *
;* DS:DX — указатель на массив шрифта.  *
;****************************************
PROC WriteFontFile NEAR
        pusha
        ; Создать файл для записи
        push    DX
        mov     AH,3Ch
        mov     CX,0    ;доступ без ограничений
        mov     DX,offset BinFileName ;имя файла
        int     21h
        jc      @@Err
        mov     BX,AX   ;запомнить Handle в BX
        pop     DX
        ; Записать данные в файл
        mov     CX,4096 ;размер файла в байтах
        mov     AH,40h
        int     21h
        jc      @@Err
        ; Закрыть файл
        mov     AH,3Eh
        int     21H
        jc      @@Err
        popa
        ret
```

```
; Аварийный выход – ошибка при записи файла
@@Err:   ; Установить текстовый режим
         mov     AX,3
         int     10h
         ; Скрыть курсор
         mov     [ScreenString],25
         mov     [ScreenColumn],0
         call    SetCursorPosition
         ; Выдать сообщение об ошибке
         mov     SI,offset WrtFontErr
         call    ShowColorString
         mov     SI,offset PrKbButton
         call    ShowColorString
         call    GetChar
         ; Выход в DOS
         mov     AH,4Ch
         int     21h
ENDP WriteFontFile

;**************************************
;*      ПРОЧИТАТЬ ШРИФТ ИЗ ФАЙЛА      *
;* Параметры:                         *
;* DS:DX – указатель на массив шрифта. *
;**************************************
PROC ReadFontFile NEAR
         pushad
         ; Открыть файл для чтения
         push    DX
         mov     AH,3Dh
         mov     AL,0     ;доступ "только для чтения"
         mov     DX,offset BinFileName ;имя файла
         int     21h
         jc      @@Err
         mov     BX,AX    ;запомнить Handle в BX
         pop     DX
         ; Прочитать данные из файла
         mov     CX,4096 ;размер файла в байтах
         mov     AH,3Fh
         int     21h
         jc      @@Err
         ; Закрыть файл
         mov     AH,3Eh
         int     21H
         jc      @@Err
         popad
         ret
; Аварийный выход – ошибка при чтении файла
@@Err:   ; Установить текстовый режим
         mov     AX,3
         int     10h
         ; Скрыть курсор
         mov     [ScreenString],25
         mov     [ScreenColumn],0
         call    SetCursorPosition
         ; Выдать сообщение об ошибке
         mov     SI,offset NoFontFile
         call    ShowColorString
         mov     SI,offset PrKbButton
         call    ShowColorString
         call    GetChar
         ; Выход в DOS
         mov     AH,4Ch
         int     21h
ENDP ReadFontFile
ENDS
```

ПРИМЕЧАНИЕ

Запускать пример из листинга 3,1 можно на любом AT-совместимом компьютере как с жесткого, так и с гибкого диска. Запись выполняется медленно - операция занимает несколько секунд даже при использовании жесткого диска (слишком медленно работают функции DOS). При запуске программы с гибкого диска снимите защиту записи.

Контрольные вопросы.

1. На каких уровнях имеет системную поддержку дисковая память?
2. Опишите характеристики уровней сервиса, доступные для утилит и прикладных программ при работе с дисковой памятью.
3. Где находятся драйверы традиционных и нетрадиционных дисковых устройств, вызываемых

программным прерыванием INT13h?
4. Какие прерывания, помимо INT13h, используются при работе с дисками?
5. Какова основная особенность драйверов многозадачных ОС?
6. Когда используется традиционный сервис BIOS?
7. Какие соглашения должны выполняться при использовании прерывания INT13h?
8. Перечислите функции прерывания INT13h традиционного дискового сервиса?
9. Каковы особенности функций 05h и 08h?
10. Что содержит таблица параметров дискет и как ей пользуются?
11. В чем отличия между стандартной и транслированной таблицами параметров жестких дисков?
12. Какие ошибки могут фиксироваться при работе с дискетами через INT13h?
13. В чем барьера 528 Мбайт и как он преодолевается?
14. Каковы особенности трансляции ECDH и LBA параметров вызова INT 13h в физические параметры, передаваемые в контроллеры ATA?
15. Какой объем диска допускает физический интерфейс и регистры контроллера ATA при трехмерной и логической адресации?
16. Каковы варианта преодоления программных барьеров?
17. Где используется расширенный скрвис BIOS?
18. Каков набор функций расширенного сервиса?
19. Опишите формат адресного пакета.
20. Охарактеризуйте функции 41h - 48h.
21. Опишите структуру буфера параметров устройств.
22. Что содержится в расширенной таблице параметров диска (DPTE) и для чего ее используют?
23. Какие сервисы предоставляет ОС прикладным программам для работы с дисками?
24. Какой файловый сервис был в ранних MS DOS?
25. Какой сервис был принят начиная с версии MS DOS 2.0?
26. Какие стандартные устройства поддерживались этим сервисом?
27. Как вызывались сервисы и какие функции ими поддерживались?
28. В чем заключается оптимизация доступа к дискам в рамках ОС?
29. Что представляет собой группа дисковых функций MS DOS?
30. Охарактеризуйте классические функции для работы с дисками через прерывание INT21h.
31. Охарактеризуйте улучшенные функции для работы с дисками.
32. Охарактеризуйте низкоуровневые функции DOS.

В данном разделе использованы материалы из [1, 2, 3, 4]. Более подробные сведения по функциям BIOS приведены в[4] и в разделах 4 - 7 данной разработки.

4. Память на гибких магнитных дисках

Накопители на гибких магнитных дисках (НГМД), или дискетах, применялись с первых моделей PC, у которых они были единственным средством хранения и переноса информации. С тех пор эти устройства претерпели относительно небольшие изменения — размер дискеты уменьшился почти в два раза, а емкость возросла всего на порядок. По сравнению с другими компонентами прогресс невелик. Английское сокращенное название НГМД — *FDD* (Flexible или Floppy Disk Drive).

Первые накопители использовали дискеты диаметром 5,25" (133 мм), для краткости обозначаемые как 5", заключенные в бумажные конверты. Первые накопители использовали 40 треков на рабочей поверхности, позже появились устройства с удвоенной поперечной плотностью, имеющие 80 треков. Дискеты имели одну *SS* (Single Side) или две *DS* (Double Side) рабочие поверхности, а накопители, соответственно, одну или две головки. По продольной плотности записи различают устройства *обычной* и *высокой* плотности записи, позволяющие записывать на треке 9 или 15 секторов соответственно (360 Кбайт и 1,2 Мбайт). Более высокая плотность записи данных сопровождается и повышенной скоростью их передачи.

Вместе с машинами класса AT появились накопители для дискет диаметром 3,5", также для краткости обозначаемые как 3". Эти компактные дискеты (говорят, что их диаметр определялся по размеру нагрудного кармана рубашки) заключены в жесткий пластмассовый конверт. Кроме более высокой плотности хранения информации, чем у 5" дискеты, они лучше защищены от внешних воздействий (пыли и деформации). Все 3" накопители используют 80 треков. По продольной плотности существуют три градации, обеспечивающие хранение 9, 18 или 36 секторов на треке в стандартном режиме форматирования с емкостью 720 Кбайт, 1,44 и 2,88 Мб соответственно.

4.1. Дискеты: плотность и форматы

Дискеты различаются по диаметру диска и плотности хранения информации, их параметры обычно входят в обозначение типа.

По *плотности записи* (Density) различают следующие типы:

- *SD* (Single Density) — одинарная плотность (давно устаревший тип);
- *DD* (Double-Density) — двойная продольная плотность (стандартные дискеты 360 Кбайт);
- *QD* (Quadra-Density) — двойная продольная плотность с удвоенным количеством треков (720 Кбайт);
- *HD* (High-Density) — высокая плотность (стандартные дискеты 1,2 и 1,44 Мбайт);
- *ED* (Extra-High Density) — сверхвысокая плотность (2,88 Мбайт),

Поперечная плотность измеряется количеством треков на дюйм TPI (Track Per Inch), хотя пользователю интереснее знать количество дорожек — именно его можно задавать в качестве одного из параметров команды FORMAT. Соотношение плотностей записи и форматированной емкости (в стандартном для PC режиме форматирования) приведено в табл. 4.1.

Таблица 4.1. Размеры/ плотность и емкость дискет

Размер	Число треков	Поперечная плотность,	Обозначение	Емкость
5"	40	48	DD	360 Кбайт
5"	80	96	QD	720 Кбайт
5"	80	96	HD	1,2 Мбайт
3.5"	80	135	OD	720 Кбайт
3.5"	80	135	HD	1.44 Мбайт
3,5"	80	135	ED	2,88 Мбайт

Стандартизация форматов записи на дискеты обеспечивает переносимость данных между компьютерами. Кроме стандартных форматов, для повышения надежности использующих поверхность диска не совсем полностью, распространены и некоторые другие форматы, доступные после загрузки драйверов типа 800.COM и им подобных. Современные версии BIOS позволяют использовать нестандартные параметры (иногда только при чтении) и без дополнительных драйверов. Наиболее употребимые форматы дискет и ключи команды FORMAT, их обеспечивающие, приведены в табл. 4.2.

Для дискет IBM PC существует стандартный формат дорожки, который в документации называется IBM System 34 Format. Каждая дорожка состоит из записей трех типов:

- начало дорожки;
- сектор на дорожке;
- конец дорожки.

На отформатированной дорожке содержится по одной записи начала и конца дорожки, а также одна или несколько записей секторов, как это показано на рис. 4.1.

Таблица 4.2. Наиболее употребимые форматы дисков

Емкость	Плотность	Дисковод					Параметры команды DOS FORMAT
		5"		3"			
		360	1,2	720	1,44	2,88	
360KB	DD	S+	+	*	*	*	/T:40 /N:9
400KB	DD	*	*	*	«	*	/T:40 /N:10
720KB	DD	-	*	S+	+	+	/T:80 /N:9
800KB	DD	-	*	4-	*	*	/T:80 /N:10
1,2MB	HD	-	S+	-	*	*	/T:80 /N:15
1,36MB	HD	-	*	-	*	*	/T:80 /N:17
1,44MB	HD	-	-	-	S+	+	/T:80 /N:18
1,6MB	HD	-	-	-	*	*	/T:80 /N:20
2,88MB	ED	-	-	-	-	S+	/T:80 /N:36

S — стандартные форматы по умолчанию.
+ — стандартные DOS-форматы.
* — доступны через драйверы типа 800.COM.

Начало дорожки содержит следующие поля:

GAP4A - промежуток, записываемый контроллером при форматировании дорожки. Содержит 80 байт заполненных значением 4Eh;

SYNC - промежуток, записываемый контроллером при форматировании дорожки. Содержит 12 байт заполненных значением 00h;

IAM - адресный маркер начала дорожки, записываемый контроллером при форматировании дорожки. Содержит три байта, заполненных значением C2h со специальным нарушением последовательности бит синхронизации, и байт со значением FCh;

GAP1 - промежуток, записываемый контроллером при форматировании дорожки. Содержит 50 байт, заполненных значением 4Eh. .

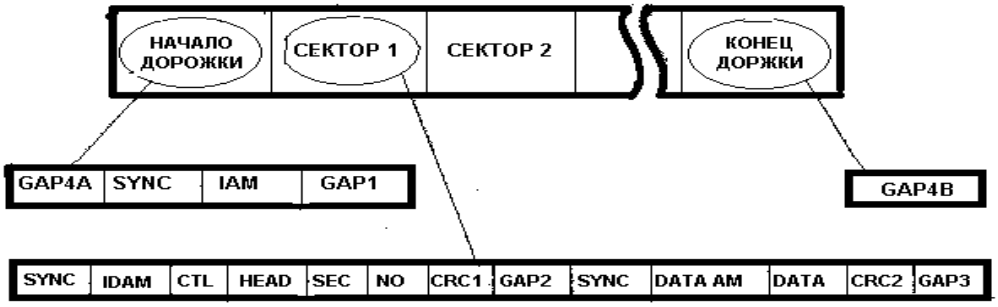

Рис. 4.1. Формат дорожки.

Запись сектора на дорожке включает такие поля:

SYNC - промежуток, записываемый контроллером при форматировании дорожки. Содержит 12 байт, заполненных значением 00h;

IDAM - адресный маркер идентификатора сектора, записываемый контроллером при форматировании дорожки. Содержит три байта со значением Alh и специальным нарушением последовательности бит синхронизации, а также значение FEh - признак маркера идентификатора;

CYL - номер цилиндра (один байт). Записывается при форматировании дорожки;

HEAD - номер головки (дорожки). Занимает один байт и записывается при форматировании дорожки;

SEC - номер сектора (один байт). Записывается при форматировании дорожки;

NO - код размера сектора (один байт). Записывается при форматировании дорожки и определяет размер поля данных в секторе. Длина поля данных определяется по формуле $128*2^{NO}$;

CRC1 - контрольная сумма идентификатора сектора (два байта);

GAP2 - промежуток, записываемый контрадлером при форматировании дорожки. Содержит 22 байта, заполненных значением 4Eh;

DATA AM - адресный маркер данных. Записывается контроллером при форматировании дорожки или при записи данных. Содержит три байта, заполненных шестнадцатсричным значением Alh со

специальным нарушением последовательности бит синхронизации и значение FBh (обычные данные) или F8h (стертые данные);

DATA - поле данных. Длина поля определяется параметром NO, как описано выше;

CRC2 - контрольная сумма данных (два байта).

GAP3 - промежуток, записываемый при форматировании дорожки. Длина промежутка задается в операции форматирования.

Конец дорожки представляет собой единственное поле **GAP4B** - промежуток, записываемый контроллером при форматировании дорожки. Содержит значения 4Eh. Длина поля определяется остатком дорожки до индексного маркера.

В IBM PC приняты стандартные параметры формата дискет, приводимые в таблице 4.3. Значение GAP3 и количество секторов приводится в десятичном формате.

Таблица 4.3. Стандартные параметры формата дискет

Тип дискеты	Размер сектора	GAP3	Секторов на дорожке
5.25"/360	NO-2 (512)	80	9
5.25"/1.2	NO-2 (512)	84	15
3.5"/720	NO-2 (512)	80	9
3.5"/1.44	NO-2 (512)	108	18

Для надежного хранения данных следует использовать дискеты с плотностью, однозначно соответствующей используемому накопителю и формату, как указано в табл. 8.10. У дискет 3" имеются ключевые отверстия, по которым датчик накопителя может определить тип носителя Дискеты имеют *ключ защиты записи.* У дискет 5" заклеенное непрозрачной наклейкой окно не позволяет накопителю (точнее, контроллеру) выполнять запись. У дискет 3" имеется выдвижная шторка, но здесь, наоборот, перекрытое окно разрешает запись. Защита дискет может и не сработать, если провод 28-интерфейсного шлейфа оборван.

Дискеты применяются как для оперативного переноса информации, так и для хранения архивов.

4.2. Накопители на гибких магнитных дисках

Накопители на гибких дисках устроены практически одинаково. Для позиционирования головок применяются шаговые двигатели. Шпиндельные двигатели современных накопителей совместно с платой электроники автоматически поддерживают требуемую *частоту вращения*. В старых накопителях иногда предусматривалась возможность подстройки частоты вращения с помощью регулировочного потенциометра.

Среди 5" устройств были наиболее распространены *40-дорожечные дисководы DD* и *80-дорожечные HD.* 80-дорожечные накопители QD, широко используемые в компьютерах семейства ZX-Spectrum, в PC широкого распространения не получили, хотя и встречались в некоторых моделях XT-подобных машин: Olivetti, EC-1841 и некоторых других. Для их использования в 40-дорожечном режиме BIOS должна сообщать контроллеру удвоенный номер трека. По количеству проводов, подходящих к шпиндельному двигателю 5" дисковода, можно определить тип накопителя: если их 4 — то дисковод режим HD не поддерживает, если их 5 — поддерживает.

При записи данных на дискету, которые не хотелось бы потерять, есть смысл включать *верификацию записи.* И в MS-DOS и в Windows 95 работает команда VERIFY ON, включающая контрольное считывание после каждой записи. Конечно, при этом запись происходит несколько дольше, но риск больших потерь становится несколько меньше. Чтобы впоследствии верификация не тормозила процесс записи на жесткий диск, имеется обратная команда VERIFY OFF, которую вводят после записи файлов на дискету.

Все НГМД, применяемые в PC, независимо от типа и размера, имеют одинаковый ***интерфейс*** и ***унифицированные 34-контактные разъемы*** двух типов: с печатными двусторонними ламелями у 5"-устройств и двухрядными штырьковыми контактами у 3"-устройств. Используемый в PC кабель-шлейф имеет перевернутый фрагмент из 7 проводов с номерами 10-16. Этот поворот позволяет подключать к контроллеру одним шлейфом до двух НГМД, причем адрес накопителя определяется его положением на шлейфе: для привода A: фрагмент перевернут, для B: — нет. Универсальный шлейф с пятью разъемами позволяет подключать пару любых дисководов, которые должны располагаться в разных зонах.

Состав сигналов интерфейса приведен в табл. 4.4, где показано, как эти сигналы приходят на разные накопители. Все сигналы интерфейса являются цифровыми (ТТЛ) с низким активным уровнем.

Интерфейс дисководов подразумевает наличие терминаторов на устройствах. Теоретически их предполагалось включать только на последнем дисководе в шлейфе, практически же их никогда и не отключают. Современные трехдюймовые накопители используют «распределенный терминатор».

Таблица 4.4. Кабель интерфейса НГМД

Контроллер		Дисковод B:		Дисковод A:	
Контакт	Сигнал	Контакт	Сигнал	Контакт	Сигнал
2	Reduce Write	2	Low Cur.	2	Low Cur.
4	Reserved	4	Reserved	4	Reserved
6	Reserved	6	DS3	6	DS3
8	Index	8	Index	8	Index
10	Motor On A	10	DS0	16	Motor*
12	Drive Sel 1	12	DS1*	14	DS2
14	Drive Sel 0	14	DS2	12	DS1*
16	Motor On B	16	Motor*	10	DS0
18	Direction	18	Direction	18	Direction
20	Step	20	Step	20	Step
22	Write Data	22	WData	22	Wdata
24	Write Gate	24	WGate	24	Wgate
26	Track 00	26	TR00	26	TR00
28	Write Protect	28	WProt	28	Wprot
30	Read Data	30	RData	30	Rdata
32	Side 1	32	Side 1	32	Side 1
34**	Disk Changed	34	DC	34	DC

Нечетные контакты 1-33 — земля. Для 5" дисководов ключ между контактами 4-5 и 6-7.

* Пара сигналов, обеспечивающая выборку FDD (Motor On A и Drive Sel 0 для дисковода A: и Motor On B и Drive Sel 1 для дисковода B).

** Контакт 34 используется только в AT.

Логически интерфейс довольно прост. Для перемещения головок на один шаг контроллер должен подать импульс Step, направление перемещения определяется уровнем сигнала Direction: при низком уровне (сигнал активен) перемещение происходит в сторону центра диска (номер трека увеличивается). Нулевой трек контроллер находит, перемещая головки от центра до появления сигнала Track 00. Выбор номера головки производится сигналом Side 1. Начало трека накопитель отмечает импульсом Index, который вырабатывается при прохождении индексного отверстия вращающейся дискеты мимо датчика. Считываемые данные в закодированном виде (но усиленные и сформированные в ТТЛ-сигнал) поступают от накопителя по линии Read Data. Для включения режима записи служит сигнал Write Gate, закодированные данные в цифровом виде поступают от контроллера по линии Write Data. Если установлена дискета, защищенная от записи, накопитель сообщит об этом сигналом Write Protect. Для снижения тока записи, которое требуется при работе накопителей HD с дискетами DD и QD, предназначен сигнал Reduce Write. Накопители HD при *смене дискеты* устанавливают сигнал Disk Changed, который сбрасывается после обращения к этому накопителю. Этот сигнал заслуживает особого внимания. Он имеется только у дисководов HD и ED, причем использование этого контакта может определяться джамперами дисковода. В PC соответствующий джампер устанавливается в положение «DC» (Disk Change). Альтернативное использование этой линии — сигнализация готовности устройства, что может обозначаться как RY, RDY или SR. Такая установка джампера приводит к постоянной ошибке «Drive not ready». При обрыве цепи 34 возникают «фантомные директории»: сменив дискету, система видит каталог предыдущей и при записи, естественно, испортит данные. Это явление объясняется тем, что, не получив сигнала смены носителя, операционная система воспользуется копией каталога предыдущей дискеты, хранящейся в буфере (в ОЗУ). От таких «призраков», часто принимаемых за действие вируса, в среде DOS спасает нажатие комбинации CTRL+C (CTRL+BREAK) при каждой смене диска, что приводит к принудительной очистке буферов.

Для того чтобы заставить накопитель работать, нужно включить его мотор сигналом Motor On и выбрать накопитель сигналом Drive Set. Для выборки накопитель имеет четыре сигнала DS0...DS3, но отзывается только на один из них, определенный установкой джамперов. Только выбранный накопитель воспринимает управляющие сигналы от контроллера и передает контроллеру свои выходные сигналы. О том, что накопитель выбран, свидетельствует светодиодный индикатор на его лицевой панели.

Контроллер НГМД и интерфейсный кабель, принятые в PC, позволяют адресоваться к одному из двух накопителей и включать мотор сигналами Drive Sel 0 и Motor On A для накопителя A: и сигналами

Drive Sel 1 и Motor On B для накопителя B. При этом на обоих накопителях джамперы устанавливаются так, что они отзываются на сигнал Drive Sel 1 (контакт 12 разъема). Обычно джамперы на дисководе обозначаются DS0/DS1/DS2/DS3, и следует установить джампер DS1. Если джамперы обозначаются как DS1/DS2/DS3/DS4, что встречается не часто, то следует установить DS2. Принятая система выборки позволяет все дисководы конфигурировать однотипно, а адрес задавать положением на шлейфе. Современные версии BIOS Setup имеют опцию *Swap Floppy,* при включении которой дисководы A: и B: меняются местами без переключения кабеля.

Некоторые специфические клоны PC используют иную систему выборки накопителей и «прямой» кабель-шлейф. При этом используется выборка устройства сигналом DS0, но переключение выборки на эту линию некоторыми накопителями не поддерживается, в результате чего замена накопителей в этих «фирменных» машинах может стать хлопотным делом, особенно при отсутствии технической документации.

Правильность конфигурирования и подключения накопителей легко проверить при начальной загрузке. Если обращение к дискетами при загрузке не запрещено опциями BIOS Setup, то должны мигнуть индикаторы сначала устройства A:, затем — B:. Если они светятся постоянно, скорее всего перевернуты разъемы шлейфа. Если они мигают одновременно, то оба устройства оказались на одном адресе.

Типы установленных дисководов задаются стандартными опциями BIOS Setup и хранятся в памяти CMOS. Установка неверного типа приводит к ошибкам обращения к дисководам.

4.3. Контроллеры НГМД

Контроллер накопителей на гибких дисках FDC (Floppy Drive Controller) является всегда внешним по отношению к накопителю и обычно располагается на одной плате с контроллером или адаптером жестких дисков. Причина этой взаимной привязанности будет пояснена ниже. Программная модель этого контроллера в PC является элементом стандартной спецификации. Все контроллеры, включая и i82077, обеспечивают совместимость с «исторической» микросхемой контроллера NEC PD765, которая является аналогом i8272 (см. ниже).

Контроллер *FDC XT* поддерживает до четырех накопителей (FDD), хотя многие контроллеры имеют интерфейсные схемы только для одного шлейфа, то есть для двух накопителей. Эти контроллеры обеспечивают скорость передачи данных 250 и 300 Кбит/с, что соответствует плотностям записи SD и DD или QD. FDC XT занимает порты с адресами 3F0h-3F7h.

Контроллер *FDC AT* поддерживает только два накопителя (стало ясно, что за счет гибких дисков объем дисковой памяти увеличивать не эффективно), но появилась новая скорость 500 Кбит/с для плотности HD (1,2 и 1,44 Мб). Контроллеры, поддерживающие и накопители ED (2,88 Мб), должны обеспечивать и скорость 1000 Кбит/с. В карте ресурсов AT имеется место под два контроллера НГМД:

- FDC AT#1 (стандартный или основной) занимает порты с адресами 3F0h-3F7h (как и FDC XT).
- FDC AT#2 (дополнительный) занимает порты с адресами 370h-377h.

Контроллеры вырабатывают запрос *аппаратного прерывания* IRQ6 (BIOS INT 0Eh) по окончании выполнения внутренних операций.

Для обмена данными может использоваться канал DMA2.

Контроллер НГМД поддерживается дисковым сервисом BIOS INT 13h. С контроллером общается и обработчик аппаратного прерывания от таймера IRQ0 (BIOS INT 08h), который декрементирует счетчик времени работы мотора НГМД (BIOS Data Area, ячейка 0:0440) и по его обнулению отключает мотор. Адрес дисковода, мотор которого включен, в позиционном коде содержит ячейка 0:043F. При каждом обращении к дискете в регистре контроллера устанавливается бит включения мотора и в счетчик времени заносится константа, соответствующая выдержке на отключение (по умолчанию — 2 с). Таким образом, если в течение этого интервала нет следующих обращений, мотор автоматически выключается (если BIOS отрабатывает аппаратные прерывания от таймера). Не отключающийся мотор может быть косвенным признаком «зависания» компьютера.

4.3.1. БИС контроллера накопителя на гибком магнитном диске К580ВГ72 (i8272).

Контроллер накопителя на гибком магнитном диске (КНГМД) КР580ВГ72 (Floppy Drive Controller FDC i8272) реализует функцию управления четырьмя накопителями на гибких магнитных дисках, обеспечивая работу в формате с одинарной FM и с двойной MFM плотностью, включая двустороннюю запись на дискету. Он имеет схему сопряжения с процессором, ориентированную на системную шину микропроцессоров серий К580, К1810, К1821 (i8080, i8086 i80286 и т.п.); обеспечивает многосекторную и многоканальную передачу объемов данных, задаваемых программно как в обычном режиме, так и в режиме ПДП (DMA); имеет встроенный генератор и схему, упрощающую построение контура фазовой автоподстройки.

Условное графическое обозначение и структурная схема КНГМД приведены на рисунках 4.2 и 4.3 соответственно.

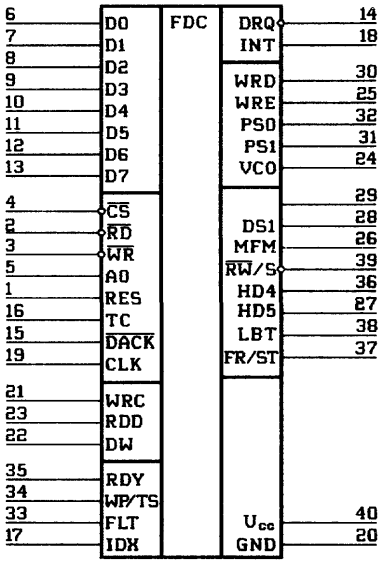

Рис. 4.2. Условное графическое обозначение КНГМД.

Назначение выводов.

RESET — сброс. Выходной сигнал, устанавливающий контроллер в исходное состояние.

-RD—чтение. Сигнал -RD=0 определяет операцию чтения данных из контроллера.

-WR—запись. Сигнал -WR=0 определяет операцию записи данных в контроллер.

-CS — выбор кристалла. Разрешение обращения к контроллеру. Сигнал -CS=0 разрешает действие сигналов -RD и -WR.

A0 — выходной сигнал, разрешающий обращение либо к регистру состояний (A0=0), либо к регистру данных (A0=1).

DB7 — DB0 — двунаправленная шина данных.

DRQ — запрос на ПДП. Сигнал DRQ=1 определяет запрос на ПДП ЦП.

-DACK — подтверждение ПДП. Сигнал от ЦП, сообщающий контроллеру о том, что шины ЦП находятся в z-состоянии.

TC — окончание ПДП. Сигнал TC= 1 сообщает контроллеру об окончании циклов ПДП.

INDEX — индекс, признак обнаружения начала дорожки.

INT — сигнал запроса прерывания ЦП от контроллера.

CLK—вход, подключаемый к генератору (4 или 8 МГц).

WRCLK — синхроимпульсы записи. Вход, подключаемый к генератору частотой F=500 КГц при одинарной плотности и F=1 МГц при двойной, с длительностью положительного полупериода 250 нс в обоих случаях. Сигналы должны быть инициированы для режимов как записи, так и чтения.

DW — информационное окно, вырабатывается схемой фазовой автоподстройки и используется для выбора данных с дисковода.

RD DATA — линия приема входных данных с дисковода в последовательном коде.

VCO — синхронизация, выходной сигнал контроллера, участвующий в формировании «окна» в схеме фазовой автоподстройки.

WE — разрешение записи, сигнал записи данных на дискету.

MFM—выбор режима плотности записи. Сигнал MFM=1 определяет двойную плотность, MFM=0—одинарную.

HDSEL—выбор головки. Сигнал HD SEL=1 определяет работу с головкой 1; HD SEL=0—работу с головкой 0.

DS1, DS0 — выбор устройства, выходные сигналы, обеспечивающие адресацию к одному из четырех дисководов.

WR DATA — линия вывода данных в последовательном коде.

PS1, PS0—предкомпенсация, выходные линии, передающие код предварительного сдвига в режиме MFM.

FLT/TRK0 — отказ/дорожка 0, указывает на сбой при операциях обмена или выбора дорожки 0 в режиме поиска.

WP/TS — защита записи/двусторонний, входной сигнал, определяющий режим записи при операциях обмена или режим поиска информации с двух сторон дискеты.

READY — сигнал готовности дисковода.

HDL — загрузка головки, выходной сигнал начальной установки головки дисковода.

FR/STP — сброс отказа/шаг, осуществляет сброс ошибки в режиме обмена и обеспечивает переход

головки на следующий цилиндр.

LCT/DIR — малый ток/направление, определяет направление движения головки.

-RW/SEEK—запись/чтение/поиск, определяет направление движения головки в режиме поиска, единичный сигнал означает увеличение, нулевой — уменьшение.

Ucc— шина питания (+5 В).

GND — общий.

Структурная схема контроллера (рис. 4.3) включает три функциональных блока: буфер шины данных (БШД), обеспечивающий связь контроллера с ЦП и вырабатывающий запросы на прерывание и ПДП; блок управления накопителями на НГМД (БУН), принимающий и вырабатывающий сигналы для управления накопителями, и блок управления контроллером (БУК).

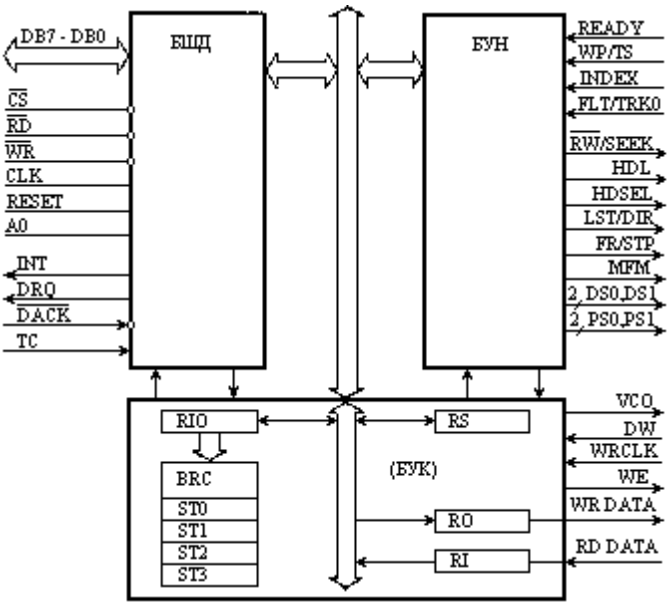

Рис. 4.3. Структурная схема контроллера НГМД ВГ72.

Блок управления контроллером включает несколько регистров специального назначения.

Регистр входных/выходных данных **RIO** адресуется при A0=1 и доступен для чтения и записи со стороны ЦП.

Основной регистр состояния **RS** доступен только для операций чтения и содержит разряды, определяющие состояние контроллера по взаимодействию с НГМД и ЦП.

Входной регистр **RI** *и выходной регистр* **RO** — регистры приема/передачи данных в последовательном коде. Программно они недоступны.

Кроме перечисленных регистров контроллер имеет блок регистров BRC для хранения кода команды и служебной информации (атрибутов), необходимой для выполнения команд. В блоке BRC можно выделить четыре регистра (ST3 — ST0), несущие информацию о состоянии контроллера и дисковода при выполнении команд.

Прием/передача данных может осуществляться контроллером в двух режимах: ПДП и программном режиме. В режиме ПДП необходимо дополнительно использовать контроллер ПДП К1810ВГ37 (i8237), обрабатывающий сигнал запроса на ПДП DRQ и формирующий сигналы подтверждения (DACK) и конца ПДП (TC). В программном режиме контроллер формирует сигналы запроса на прерывание INT при пересылке каждого байта между контроллером и ЦП, предоставляя возможность управления обменом подпрограмме ЦП.

Кроме того, контроллер имеет схему обнаружения адресного маркера, что упрощает реализацию контура фазовой автоподстройки. Время перехода с дорожки на дорожку и время загрузки/разгрузки головки (позиционирование) задается программно. Контроллер имеет много дополнительных возможностей, таких, как многократное выполнение операций чтения/записи с помощью одной команды, режимы одинарной и двойной плотности записи.

Схема подключения контроллера в режиме ПДП для одного накопителя представлена на рис. 4.4. На рис. 4.5 приведена схема МПС с КНГМД, в которой используются также три БИС универсального приемника/передатчика USART К580ВВ51А (i8251) для приема/передачи данных в последовательном коде, программируемый таймер PIT К1810ВИ54 (i8254) для управления синхронизацией приема /передачи USART и контроллер прерывания PIC для управления режимом прерывания. Схема позволяет

использовать КНГМД вместе с клавиатурой и другими внешними устройствами, для обслуживания которых нужны указанные БИС.

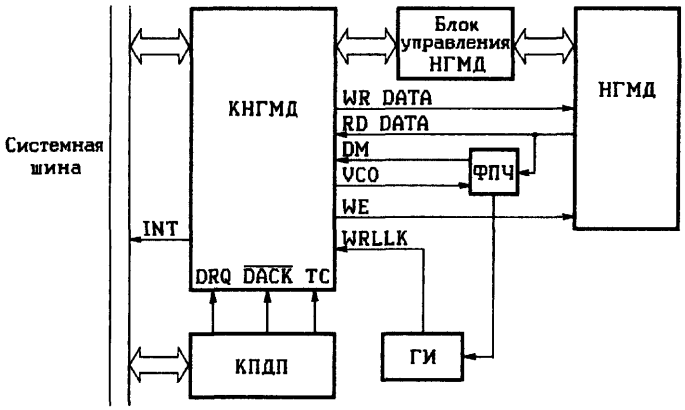

Рис. 4.4. Схема подключения контроллера к накопителю.

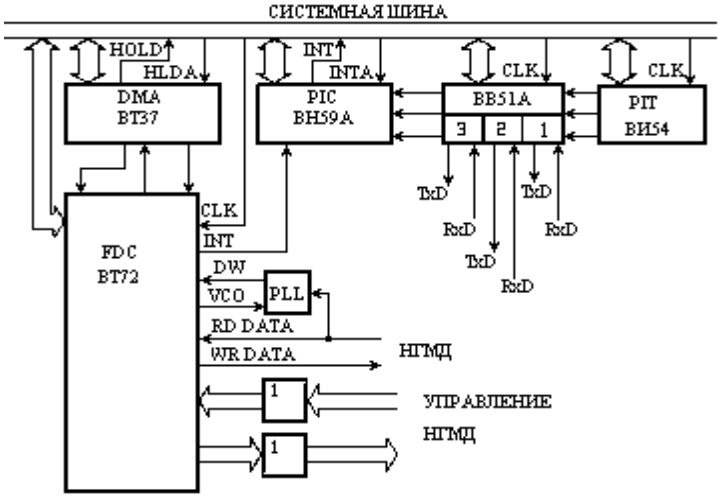

Рис. 4.5. Схема подключения КНГМД к шине МПС.

На рис. 4.6—4.10 представлены временные диаграммы работы контроллера в различных режимах.

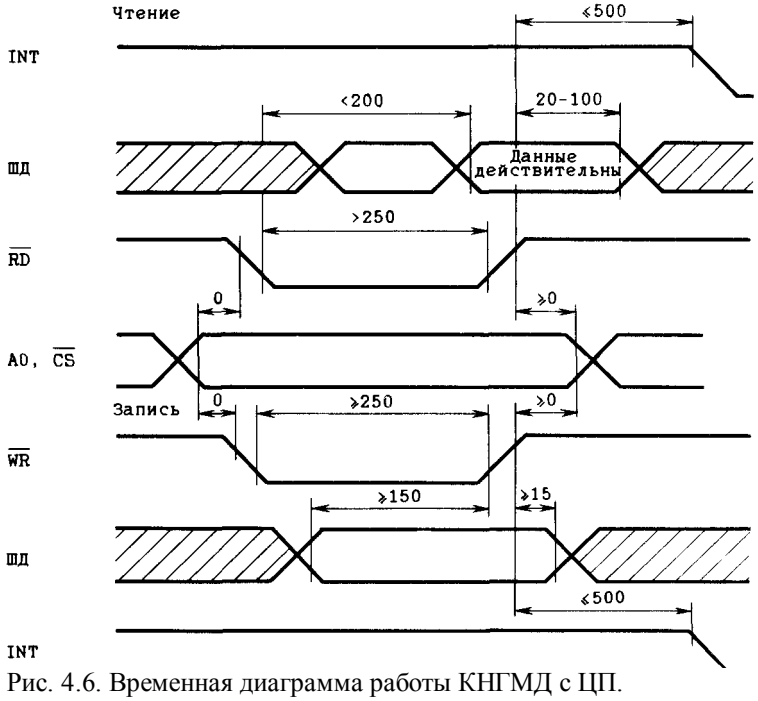

Рис. 4.6. Временная диаграмма работы КНГМД с ЦП.

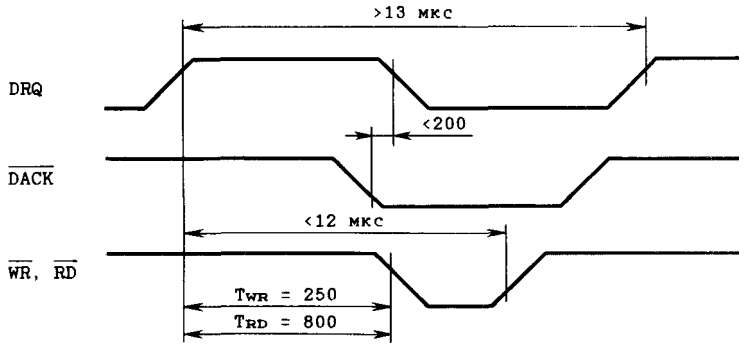

Рис. 4.7. Временная диаграмма работы в режиме ПДП.

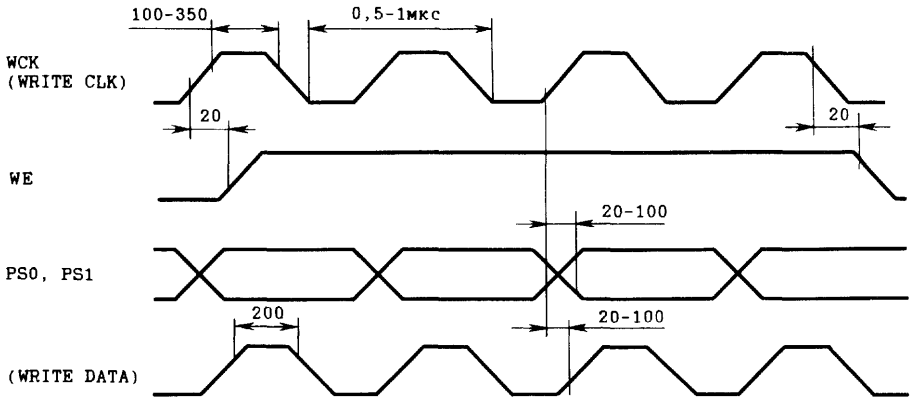

Рис. 4.8. Временная диаграмма работы КНГМД в режиме записи на дискету.

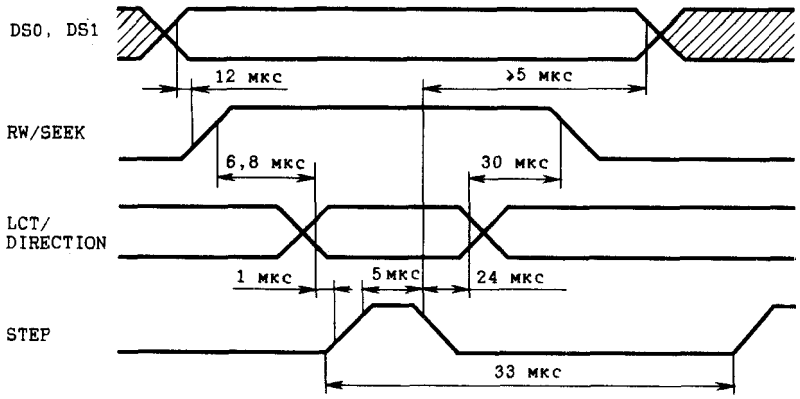

Рис. 4.9. Временная диаграмма выполнения операции Поиск (FCLK = 8 МГц).

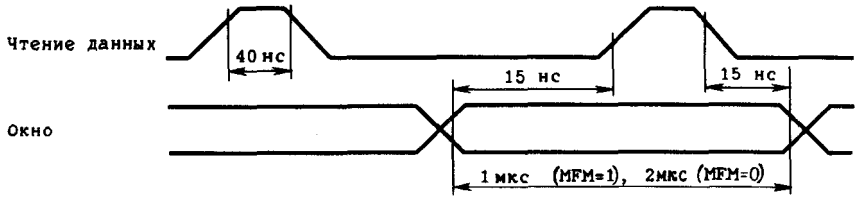

Рис. 4.10. Временная диаграмма работы КНГНД в режиме чтения данных с дискеты.

КНГМД может выполнять 15 различных команд, каждая из которых загружается в контроллер многобайтовой пересылкой из ЦП. Выполнение команды можно разделить на три фазы. В фазе 1 контроллер получает от ЦП всю информацию, необходимую для выполнения команды; в фазе 2 он выполняет команду; в фазе 3 — фазе результата — ЦП получает информацию о состоянии котроллера.

4.3.2. Регистры контроллера НГМД PS/2 и PC/AT.

Адреса регистров контроллера находятся в диапазоне от 3F0h до 3F7h. В таблице 4.4 приводятся относительные адреса и условные наименования регистров. Адрес порта ввода-вывода регистра вычисляется путем сложения базового адреса (3F0h) с относительным адресом. Ряд регистров, имеющихся в контроллере i82077, не описывается, так как эти регистры не используются в компьютерах

фирмы IBM.

Таблица 4.4. Регистры контроллера

Адрес	Режим доступа	Наименование
0	R	Регистр состояния A (SRA)
1	R	Регистр состояния B (SRB)
2	W	Регистр управления (DOR)
4	R	Основной регистр состояния (MSR)
5	R/W	Регистр данных (DR)
7	R	Регистр сигналов входа (DIR)
7	W	Регистр конфигурации (CCR)

Регистр состояния A (SRA). Данный регистр используется только на компьютерах серии PS/2, оборудованных шиной MCA. На других компьютерах этот регистр не используется. Формат регистра приведен ниже.

7	6	5	4	3	2	1	0
INT	DRV2	STEP	TRK0	HDS	INDX	WP	DIR

INT указывает на состояние сигнала прерывание: 0 - нет прерывания; 1 - контроллер хранит прерывание.

DRV2 указывает наличие второго дисковода: 0 - второй дисковод подключен; 1 - второго дисковода нет.

STEP показывает состояние сигнала установки головок (направление установки зависит от значения бита DIR): 0 - нет сигнала установки головок, 1 - в данный момент выполняется перемещение головок на одну дорожку.

TRK0 указывает, находятся ли головки на дорожке 0: 0 - головки на нулевой дорожке; 1 - головки не на нулевой дорожке.

HDS определяет, какая головка выбрана в настоящий момент: 0 - выбрана головка 0; 1 - выбрана головка 1.

INDX определяет состояние сигнала обнаружения начала дорожки (индекса): 0 - индекс обнаружен; 1 - нет индекса.

WP указывает состояние линии защиты записи: 0 - есть защита записи; 1 - запись разрешена.

DIR показывает направление перемещения головок: 0 - перемещение к младшим номерам цилиндров (к наружной поверхности); 1 - перемещение к старшим номерам цилиндров (к внутренней поверхности).

Биты 6, 4, 2 и 1 отображают состояние сигналов от накопителя. После сброса биты 7, 5, 3 и 0 устанавливаются в 0.

Регистр состояния B (SRB). Данный регистр используется только на компьютерах серии PS/2, оборудованных шиной MCA. На других компьютерах этот регистр не используется. Формат регистра и значения битов описаны ниже.

7	6	5	4	3	2	1	0
Резерв		DSEL	WRD	RDD	WE	MOT1	MOT0

DSEL указывают выбранный накопитель: 0 - выбран накопитель 0; 1 - выбран накопитель 1.

WRD показывает состояние сигнала «запись данных» со стороны контроллера.

Состояние бита **RDD** изменяется в соответствии с сигналом «чтение данных» со стороны накопителя.

WE определяет, выполняется ли запись. Указывает состояние соответствующего сигнала контроллера, который включает цепь записи на накопителе.

MOT1 и **MOT0** указывают на состояние мотора накопителей 1 и 0 соответственно: 0 - мотор выключен; 1 - мотор включен.

После сброса значения битов 4 и 3 обнуляются. Биты 5, 1 и 0 обнуляются только при аппаратном сбросе.

Регистр управления (DOR). Данный регистр используется на компьютерах всех классов. Регистр предназначен для управления накопителями. Формат регистра приведен ниже.

7	6	5	4	3	2	1	0
Резерв		MOT1	MOT0	DMA	Reset	0	DSEL

MOT1 и **MOT0** управляют включением моторов накопителей 1 и 0 соответственно: 0 - выключить мотор; 1 - включить мотор.

DMA задает режим работы с DMA и прерываниями: 0 - работа с DMA и прерываниями запрещена; 1 - работа с DMA и прерываниями разрешена.

В режиме работы с прерываниями контроллер гибких дисков вырабатывает прерывание по линии 6 контроллера прерываний (IRQ 6), что соответствует вектору прерываний 0Eh а в режиме DMA - канал 2 подсистемы прерывания.

Reset указывает, требуется ли сброс контроллера: 0 - выполнить сброс; 1 - контролер доступен.

DSEL выбирает накопитель для работы: 0 - накопитель 0; 1 - накопитель 1.

Биты 6-7 и 1 в регистре управления в PC/AT зарезервированы (устанавливаются в 0).

На ряде контроллеров, например, на машинах класса PS/2, бит DMA не имеет никакого значения, так как аппаратное включение контроллера всегда устанавливает режим работы с передачей прерываний в систему и использование DMA. На этих машинах работа без DMA и прерываний невозможна. Помимо этого, режим работы с DMA определяется в команде SPECIFY (см. ниже).

Значения, загружаемые в регистр управления для выбора соответствующего накопителя и включения мотора, приведены в таблице 4.5.

Таблица 4.5. Значения регистра DOR

Номер накопителя	Значение регистра управления
0	1Ch
1	2Dh

Основной регистр состояния (MSR). Данный регистр предназначен для определения разрешения и направления передачи данных между микропроцессором и контроллером.

7	6	5	4	3	2	1	0
RQM	DIO	NDMA	CBSY	Резерв		BSY1	BSY0

RQM показывает, разрешена ли передача данных: 0 - передача данных запрещена; 1 - передача данных разрешена.

DIO указывает направление передачи данных, когда бит RQM установлен в 1: 0 - передача данных от микропроцессора к контроллеру; 1 - передача данных от контроллера к микропроцессору.

Бит **NDMA** действителен, когда в команде SPECIFY указан режим работы без DMA. Бит устанавливается в 1 во время фазы выполнения команды. Позволяет дифференцировать фазу выполнения и фазу чтения результатов.

Бит **CBSY** устанавливается в 1 во время выполнения команды сразу после передачи в контроллер байта команды и сбрасывается после завершения фазы чтения результата. Если команда не имеет фазы чтения результата, то этот бит сбрасывается в 0 после передачи последнего байта команды.

BSY1 и **BSY0** устанавливаются в 1 для накопителей 1 и 0 соответственно, когда накопитель находится в режиме установки головок

Регистр данных (DR). Через этот регистр передается байт данных между микропроцессором и контроллером. Разрешение на передачу данных и направление передачи определяется состоянием битов 7-6 основного регистра состояния (MSR).

Регистр сигналов входов (DIR). Регистр определен только на контроллерах, поддерживающих накопители с несколькими плотностями записи и чтения. Таким образом, регистр отсутствует на контроллерах базовых моделей IBM PC/XT. Формат регистра представлен ниже.

7	6	5	4	3	2	1	0
DCH	Резерв						HSEL

Данный регистр определяет смену дискеты в накопителе. Это позволяет установить момент, когда необходимо выполнение процедуры выбора плотности записи-чтения (выбирается двойная или повышенная плотность).

Единичное значение бита **DCH** указывает, что в накопитель установлена новая дискета. Значение бита сбрасывается при выполнении команды «Установить» (SEEK) при ненулевом перемещении головок.

Бит **HSEL** поддерживается только на компьютерах класса PS/2. Он определяет текущую скорость передачи данных между контроллером и накопителем. Единичное значение указывает, что выбрана скорость передачи данных 250 килобит/сек (двойная плотность), в противном случае - скорость 500 килобит/сек (высокая плотность). В контроллерах современных PC биты 6 - 0 имеют следующие значения:

Бит 6: 0 - нет записи; 1 - запись.

Биты (2 - 5) определяют выбранную головку 0 - 3 (позиционный код).

Бит 1: 1 - выбран привод 1.

Бит 0: 1 - выбран привод 0.

Регистр конфигурации (CCR). Регистр определен только на контроллерах, поддерживающих накопители с несколькими плотностями записи и чтения. Этот регистр отсутствует на контроллерах базовых моделей IBM PC/XT.

Регистр определяет скорость передачи данных между контроллером и накопителем.

7	6	5	4	3	2	1	0
Резерв						SSEL	

Биты **SSEL** определяют скорость передачи данных между накопителем и контроллером:

- 00 - 500 килобит/сек;
- 01 - 300 килобит/сек;
- 10 - 250 килобит/сек;
- 11 - зарезервировано.

В контроллерах современных PC биты 7 - 0 имеют следующие значения:

- Бит 7: 1 - сброс контроллера.
- Бит 6: 1 - отключение питания дисководов.
- Бит 5: 1 - разрешение внешней фильтрации данных.
- Биты (4 - 2): - выбор времени предкомпенсации.
- Биты (1,0) - выбор скорости обмена: 00 - 500 Кбит/с; 01 - 300 Кбит/с; 10 - 250 Кбит/с; 11 - 1 Мбит/с.

Регистры состояния

Кроме перечисленных регистров контроллер имеет блок регистров для хранения кода команды и служебной информации (атрибутов), необходимой для выполнения команд. Среди них можно выделить четыре регистра (ST3 — ST0), несущие информацию о состоянии контроллера и дисковода при выполнении команд.

Значения регистров состояния ST0-ST2 передаются контроллером после завершения операций. Содержимое этих регистров указывает, произошла ли ошибка во время выполнения операции, а также позволяет уточнить причину ошибки. Регистр состояния ST3 непосредственно отображает состояние линий со стороны накопителя. Чтение этого регистра выполняется специальной командой.

Регистр состояния 0 (ST0) *имеет следующий формат:*

7	6	5	4	3	2	1	0
IC		SE	EC	R3	HD	DS	

Биты **IC** (Interrupt Code) определяют код прерывания:

- 00 - нормальное завершение команды;
- 01 - ненормальное завершение команды;
- 10 - недействительная команда;
- 11 - аварийное завершение.

SE (Seek End) определяет завершение команды установки. Единичное значение данного бита указывает на завершение команд «Рскалибровать» или «Установить».

Бит **EC** (Equipment Check) устанавливается в 1, если при выполнении команды «Рекалибровать» после 77-ми сигналов управления перемещением головок не получен сигнал «Дорожка 0".

Бит **R3** зарезервирован.

Бит **HD** (Head Address) указывает номер головки, для которого выполнялась операция.

Поле **DS** (Drive Select) задают номер выбранного накопителя:

- 00 - накопитель 0;
- 01 - накопитель 1;
- 10 - накопитель 2;
- 11 - накопитель 3.

Регистр состояния 1 (ST1) *имеет следующий формат.*

7	6	5	4	3	2	1	0
EN	R6	DE	OR	R3.	ND	NW	MA

EN (End of Cylinder) устанавливается в 1 при попытке получить доступ к сектору за последним сектором на цилиндре.

Биты **R6** и **R3** зарезервированы.

Единичное значение бита **DE** (Data Error) указывает на ошибку контрольной суммы (CRC) в поле данных или в адресном маркере сектора.

OR (Overrun) установлен в 1, если запрос контроллера не обслужен со стороны CPU или DMA за определенный период времени.

ND (No Data) равен 1 в следующих случаях;

- не найден запрошенный сектор в командах чтения;
- адресный маркер сектора не может быть прочитан без ошибки в команде «Чтение идентификатора»;
- не найден сектор, заданный в команде «Чтение дорожки».

NW (Not Writable) равен 1, если активен сигнал «Защита записи» со стороны накопителя при выполнении команд записи или форматирования.

MA (Missing Adress Mark) установлен в 1 в следующих случаях:

- не найден адресный маркер сектора при повторном обнаружении индекса;
- не найден адресный маркер данных, при этом бит MD в ST2 также устанавливается в 1.

Регистр состояния 2 (ST2) *имеет следующий формат.*

7	6	5	4	3	2	1	0
R7	CM	DD	WC	R3	R2	BC	MD

Биты **R7, R3** и **R2** зарезервированы.

CM (Control Mark) установлен в 1 при попытке получить доступ к сектору с отметкой «стертые» данные по команде «Чтение данных» или по команде «Чтение стертых данных» к сектору с обычными данными.

DD (Data Error in Data Field) устанавливается в 1 при фиксации ошибки контрольной суммы (CRC) в поле данных.

WC (Wrong Cylinder) принимает значение 1, если внутренний номер цилиндра контроллера не совпал с номером цилиндра из идентификатора сектора.

BC (Bad Cylinder) установлен в 1, если внутренний номер цилиндра контроллера не совпал с номером цилиндра из идентификатора сектора и равен FFh. При этом установлен в 1 бит ND.

MD (Missing Data Address Mark) равен 1, если во время выполнения команд чтения не найден адресный маркер данных.

Регистр состояния 3 (ST3) *имеет следующий формат:*

7	6	5	4	3	2	1	0
R7	WP	R5	T0	R3	HD	R1	DSEL

Биты **R7, R5** и **R1** зарезервированы.

WP (Write Protect) отображает состояние сигнала «Защита записи» со стороны накопителя (1 - установлена защита записи).

T0 (Track 0) установлен в 1, когда головки находятся на дорожке 0.

HD (Head Address) указывает номер выбранной головки со стороны накопителя.

DSEL определяет номер выбранного накопителя.

4.3.3. Система команд контроллера

Любая команда контроллера гибких дисков включает до трех фаз (в некоторых командах отдельные фазы могут отсутствовать) :

• *командная фаза.* В командной фазе контроллеру от процессора передаются данные, необходимые для выполнения команды, путем последовательной записи в регистр данных кода команды и передаваемых параметров.

• *фаза выполнения.* В фазе выполнения контроллер выполняет запрошенную команду и, возможно, получает дополнительные данные от системы.

• *фаза получения результата.* В этой фазе контроллер передает процессору результат выполнения команды. Результат считывается путем последовательного выполнения команд чтения регистра данных.

В дальнейшем описании принимаются следующие традиционные для технической документации сокращения:

C - номер цилиндра;

H - номер головки;

R - номер сектора;

N - код длины сектора;

EOT - конец дорожки. Определяет номер последнего сектора на дорожке (количество секторов на дорожке);

GPL - длина межсекторного промежутка (GAP3);

DTL - длина передаваемых данных. Действительна для сектора длиной 128 байт (код длины 0). В остальных случаях значение равно FFh;

ST0 - регистр состояния 0;

ST1 - регистр состояния 1;

ST2 - регистр состояния 2;

ST3 - регистр состояния 3;

HDS - выбор номера головки;

DS - выбор номера накопителя;

HLT - время загрузки головок;

HUT - время разгрузки головок;

MT - многодорожечная операция. Обе дорожки цилиндра рассматриваются как непрерывный цилиндр;

NCN - новый номер цилиндра;

PCN - текущий номер цилиндра;

SK - признак пропуска стертых данных;

SRT - время между последовательными сигналами на перемещение головок;

TC - конец счетчика данных.

При описании команд применяются таблицы. В колонке «Фаза» показывается фаза команды. В колонке «Чтение/запись» указывается направление передачи данных (W - запись в регистр данных контроллера, R - чтение из регистра данных). В колонке «Данные» приведены значения, последовательно передаваемые между контроллером и системой. Далее приводятся основные команды контроллера гибких дисков.

Чтение данных (Read Data). Эта команда предназначена для чтения данных сектора в память машины. Команда обычно применяется для чтения данных с дискеты. По команде читаются данные, содержащие признак FBh в адресном маркере данных.

Сразу после получения команды чтения контроллер начинает сканировать данные адресных маркеров. При обнаружении сектора, идентификатор которого совпадает с заданным в команде (C,H,R,N), начинается передача данных из сектора. Затем номер сектора увеличивается на 1 и выполняется чтение данных следующего сектора. Команда завершается при получении сигнала TC (конец счетчика данных). Этот сигнал обычно генерируется средствами DMA при исчерпании счетчика

	ЧТЕНИЕ/ ЗАПИСЬ	ДАННЫЕ D7 D6 D5 D4 D3 D2 D1 D0	ПРИМЕЧАНИЕ
ФАЗА КОМАНДЫ.	W	MT 1 SK 0 0 1 1 0	КОД КОМАНДЫ
	W	0 0 0 0 0 HDS 0 DS	
	W	C	ДАННЫЕ СЕКТОРА (АТРИБУТЫ КОМАНДЫ)
	W	H	
	W	R	
	W	N	
	W	EOT	
	W	GPL	
	W	DTL	

ВЫП.	R		ПЕРЕДАЧА ДАННЫХ ОТ КОНТРОЛЛЕРА К СИСТЕМЕ
РЕЗ.	R	ST0	ИНФОРМАЦИЯ СОСТОЯНИЯ
	R	ST1	
	R	ST2	
	R	C	ДАННЫЕ СЕКТОРА
	R	H	
	R	R	
	R	N	

передаваемых данных.

При работе без DMA система должна обеспечить передачу сигнала TC. Если сигнал TC получен, а сектор не прочитан до конца, то контроллер продолжает чтение сектора до конца без передачи данных в систему и выполняет проверку контрольной суммы (CRC) и после этого завершает команду. Команда завершается аварийно, если затребованный сектор не обнаружен или произошел сбой при чтении идентификатора или данных сектора.

При работе с секторами длиной 128 байт (со значением N=0) имеет смысл поле DTL (для всех остальных размеров секторов значение DTL должно быть равно 0FFh). Если DTL меньше 128, то выполняется чтение количества байт, заданное параметром DTL. После этого контроллер считывает оставшиеся байты и выполняет проверку контрольной суммы.

Чтение стертых данных (Read Deleted Data). Эта команда предназначена для чтения данных, отмеченных как стертые, в память машины. По команде читаются данные, содержащие признак F8h в адресном маркере данных.

Выполнение команды полностью аналогично команде «Чтение данных».

	ЧТЕНИЕ/ ЗАПИСЬ	ДАННЫЕ D7 D6 D5 D4 D3 D2 D1 D0	ПРИМЕЧАНИЕ
ФАЗА КОМАНДЫ.	W	MT 1 SK 0 1 1 0 0	КОД КОМАНДЫ
	W	0 0 0 0 0 HDS 0 DS	
	W	C	ДАННЫЕ СЕКТОРА (АТРИБУТЫ КОМАНДЫ)
	W	H	
	W	R	
	W	N	
	W	EOT	
	W	GPL	
	W	DTL	
ВЫП.	R		ПЕРЕДАЧА ДАННЫХ ОТ КОНТРОЛЛЕРА К СИСТЕМЕ
РЕЗ.	R	ST0	ИНФОРМАЦИЯ СОСТОЯНИЯ
	R	ST1	
	R	ST2	
	R	C	ДАННЫЕ СЕКТОРА
	R	H	
	R	R	
	R	N	

Запись данных (Write Data). Эта команда предназначена для записи данных сектора из памяти машины. Команда обычно применяется для записи данных на дискету. По команде записываются данные, содержащие признак FBh в адресном маркере данных.

	ЧТЕНИЕ/ ЗАПИСЬ	ДАННЫЕ D7 D6 D5 D4 D3 D2 D1 D0	ПРИМЕЧАНИЕ
ФАЗА КОМАНДЫ.	W	MT 1 0 0 0 1 0 1	КОД КОМАНДЫ
	W	0 0 0 0 0 HDS 0 DS	
	W	C	ДАННЫЕ СЕКТОРА (АТРИБУТЫ КОМАНДЫ)
	W	H	
	W	R	
	W	N	
	W	EOT	
	W	GPL	
	W	DTL	
ВЫП.	R		ПЕРЕДАЧА ДАННЫХ ОТ СИСТЕМЫ К КОНТРОЛЛЕРУ

РЕЗ.	R	ST0	ИНФОРМАЦИЯ СОСТОЯНИЯ
	R	ST1	
	R	ST2	
	R	C	ДАННЫЕ СЕКТОРА
	R	H	
	R	R	
	R	N	

Команда выполняется аналогично команде «Чтение данных». Сразу после получения команды записи контроллер начинает сканировать данные адресных маркеров. При обнаружении сектора начинается запись данных в сектор. Затем номер сектора увеличивается на 1 и выполняется запись следующего сектора. Команда завершается при получении сигнала TC. Команда завершается аварийно, если затребованный сектор не обнаружен или произошел сбой при чтении идентификатора.

При работе с секторами длиной 128 байт (со значением N-0) имеет смысл поле DTL. Если DTL меньше 128, то выполняется запись количества байт, заданное параметром DTL. После этого контроллер записывает оставшиеся байты равными нулю и записывает контрольную сумму.

Запись стертых данных (Write Deleted Data). Данная команда предназначена для записи данных, помеченных как стертые, из памяти машины на дискету. По команде записываются данные, содержащие признак F8h в адресном маркере данных. Команда выполняется аналогично команде «Запись данных».

	ЧТЕНИЕ/ ЗАПИСЬ	ДАННЫЕ D7 D6 D5 D4 D3 D2 D1 D0	ПРИМЕЧАНИЕ
ФАЗА КОМАНДЫ.	W	MT 1 0 0 1 0 0 1	КОД КОМАНДЫ
	W	0 0 0 0 0 HDS 0 DS	
	W	C	ДАННЫЕ СЕКТОРА (АТРИБУТЫ КОМАНДЫ)
	W	H	
	W	R	
	W	N	
	W	EOT	
	W	GPL	
	W	DTL	
ВЫП.	R		ПЕРЕДАЧА ДАННЫХ ОТ СИСТЕМЫ К КОНТРОЛЛЕРУ
РЕЗ.	R	ST0	ИНФОРМАЦИЯ СОСТОЯНИЯ
	R	ST1	
	R	ST2	
	R	C	ДАННЫЕ СЕКТОРА
	R	H	
	R	R	
	R	N	

Чтение дорожки (Read Track). Команда предназначена для чтения данных со всей дорожки. По этой команде контроллер сразу после обнаружения индекса начинает читать данные из секторов в непрерывный буфер.

Если обнаруживается ошибка в поле адресного маркера идентификатора или поле данных, то контроллер продолжает выполнение команды, устанавливая соответствующие признаки ошибки при завершении команды. Контроллер сравнивает значение идентификатора сектора, заданное в команде, с прочитанными адресными маркерами секторов и, если сектор не найден, то устанавливает признак ND в ST1.

	ЧТЕНИЕ/ ЗАПИСЬ	ДАННЫЕ D7 D6 D5 D4 D3 D2 D1 D0	ПРИМЕЧАНИЕ
ФАЗА КОМАНДЫ.	W	0 1 0 0 0 0 1 0	КОД КОМАНДЫ
	W	0 0 0 0 0 HDS 0 DS	
	W	C	ДАННЫЕ СЕКТОРА (АТРИБУТЫ КОМАНДЫ)
	W	H	
	W	R	
	W	N	
	W	EOT	
	W	GPL	
	W	DTL	
ВЫП.	R		ПЕРЕДАЧА ДАННЫХ ОТ КОНТРОЛЛЕРА К СИСТЕМЕ

РЕЗ.	R	ST0	ИНФОРМАЦИЯ СОСТОЯНИЯ
	R	ST1	
	R	ST2	
	R	C	ДАННЫЕ СЕКТОРА
	R	H	
	R	R	
	R	N	

Команда завершается, когда прочитано количество секторов, заданное в поле EOT команды. Если ни один адресный маркер идентификатора на дорожке не найден, то признак IC в ST0 устанавливается в 01 и признак MA в ST1 устанавливается в 1.

Форматирование дорожки (Format Track). Данная команда предназначена для форматирования дорожки.

Параметр N определяет код размера сектора. SC задает количество секторов, которое требуется записать на дорожку, GPL - размер GAP3, а D - значение, записываемое в поле данных каждого сектора.

Команда начинает выполняться при обнаружении индекса. Сначала записываются данные начала дорожки.

Во время фазы выполнения от системы к контроллеру для каждого сектора должны передаваться значения C, H, R и N. Эти значения заносятся в адресный маркер идентификатора. Затем записываются адресный маркер данных и поле данных, заполненное символом, который определяется параметром D. Длина поля данных задается параметром N, определенным в командной фазе. Наконец, записывается контрольная сумма данных и поле GAP3, длина которого определяется параметром GPL, заданным в командной фазе. Этот цикл записи сектора повторяется для всех секторов, количество

	ЧТЕНИЕ/ ЗАПИСЬ	ДАННЫЕ D7 D6 D5 D4 D3 D2 D1 D0	ПРИМЕЧАНИЕ
ФАЗА КОМАНДЫ.	W	0 1 0 0 1 1 0 1	КОД КОМАНДЫ
	W	0 0 0 0 0 HDS 0 DS	
	W	N	БАЙТ В СЕКТОРЕ
	W	SC	СЕКТОРОВ НА ДОРОЖКУ
	W	GPL	GAP3
	W	D	ЗАПОЛНИТЕЛЬ
ВЫП.	W	C	ПЕРЕДАЧА ИДЕНТИФИКАТОРОВ ДЛЯ КАЖДОГО СЕКТОРА
		H	
		R	
		N	
РЕЗ.	R	ST0	ИНФОРМАЦИЯ СОСТОЯНИЯ
	R	ST1	
	R	ST2	
	R	C (НЕОПРЕДЕЛЕНО)	ДАННЫЕ СЕКТОРА ПОСЛЕ ВЫПОЛНЕНИЯ КОМАНДЫ
	R	H (НЕОПРЕДЕЛЕНО)	
	R	R (НЕОПРЕДЕЛЕНО)	
	R	N (НЕОПРЕДЕЛЕНО)	

которых задастся параметром SC. В конце дорожки записывается поле GAP4B. Запись GAP4B выполняется до тех пор, пока не будет обнаружен индекс. На этом выполнение команды форматирования завершается.

Как и все команды, операция форматирования прекращается после получения сигнала TC. В этом случае контроллер прекращает форматирование секторов и начинает запись поля GAP4B, независимо от значения, заданного в поле SC (количество секторов).

Чтение идентификатора (Read ID). Данная команда предназначена для определения местоположения головок. По этой команде считывается первый обнаруженный на дорожке идентификатор сектора. Однако, если идентификатор сектора содержит ошибку в поле контрольной суммы, то контроллер игнорирует этот сектор и продолжает сканирование дорожки. Команда завершается, когда идентификатор сектора считан без ошибок.

Если при повторном обнаружении индекса ни один идентификатор сектора не. был прочитан без ошибок, то команда завершается аварийно.

При этом поле IC регистра состояния 0 принимает значение 01, а бит MA в регистре состояния 1 устанавливается в единичное значение. В результирующей фазе передаются данные прочитанного идентификатора сектора.

	ЧТЕНИЕ/ ЗАПИСЬ	ДАННЫЕ D7 D6 D5 D4 D3 D2 D1 D0	ПРИМЕЧАНИЕ
ФАЗА КОМ.	W W	0 1 0 0 1 0 1 0 0 0 0 0 0 HDS 0 DS	КОД КОМАНДЫ
ВЫП.	R		ЧИТАЕТ ПЕРВЫЙ ВСТРЕЧЕННЫЙ ИДЕНТИФИКАТОР СЕКТОРА НЕ СОДЕРЖАЩИЙ ОШИБОК
РЕЗ.	R R R R R R R	ST0 ST1 ST2 C H R N	ИНФОРМАЦИЯ СОСТОЯНИЯ ДАННЫЕ СЕКТОРА

Рекалибровать (Recalibrate). Эта команда предназначена для установки начального положения головок (позиционирования на нулевой цилиндр). Она обязательно должна выполняться после сброса контроллера. По команде контроллер начинает выдавать сигналы управления перемещением головок.

Команда завершается при обнаружении сигнала «Дорожка О» со стороны дисковода, либо после выдачи 77-ми сигналов перемещения головок. Она не имеет обычной фазы результата. Результат выполнения команды должен быть получен по команде «Получить результат прерывания».

ФАЗА	ЧТЕНИЕ/ ЗАПИСЬ	ДАННЫЕ D7 D6 D5 D4 D3 D2 D1 D0	ПРИМЕЧАНИЕ
ФВЗА КОМ.	W W	0 0 0 0 0 1 1 1 0 0 0 0 0 0 0 DS	КОД КОМАНДЫ
ВЫП.	R		ГОЛОВКИ ПЕРЕМЕЩАЮТСЯ К ЦИЛИНДРУ 0

Установить (Seek). Эта команда предназначена для установки головок на требуемый цилиндр. По команде контроллер начинает выдавать сигналы управления перемещением головок. В зависимости от текущего положения головок сигналы выдаются для перемещения в прямом или в обратном направлении. Количество сигналов определяется разностью между текущим и требуемым номером цилиндра. Команда не имеет фазы результата. Результат выполнения команды должен быть получен по команде «Получить результат прерывания».

ФАЗА	ЧТЕНИЕ/ ЗАПИСЬ	ДАННЫЕ D7 D6 D5 D4 D3 D2 D1 D0	ПРИМЕЧАНИЕ
ФАЗА КОМАНДЫ.	W W W	0 0 0 0 0 1 1 1 0 0 0 0 0 0 0 DS NCN	КОД КОМАНДЫ НОВЫЙ НОМЕР ЦИЛИНДРА
ВЫП.	R		ГОЛОВКИ ПЕРЕМЕЩАЮТСЯ К ЗАДАННОМУ ЦИЛИНДРУ

Следующая после «Установить» команда работы с носителем (чтение/запись/формат) должна выдаваться не ранее истечения определенного временного интервала, необходимого для успокоения головок. Время успокоения головок составляет не менее 15-20 мсек.

Получить результат прерывания (Sense Interrupt Status).

Данная команда предназначена для получения результата прерывания после выполнения команд «Рекалибровать» и «Установить». Также команда используется для снятия результата прерывания после программного сброса контроллера.

	ЧТЕНИЕ/ ЗАПИСЬ	ДАННЫЕ D7 D6 D5 D4 D3 D2 D1 D0	ПРИМЕЧАНИЕ
ФАЗА КОМ.	W W	0 0 0 0 1 0 0 0 0 0 0 0 0 HDS 0 DS	КОД КОМАНДЫ
РЕЗ.	R R	ST0 PCN	ИНФОРМАЦИЯ СОСТОЯНИЯ

Результат завершения команды передается в признаках IC и SE в ST0. Признак SE устанавливается

равным 1 после команд «Рекалибровать» и «Установить». Признак IC = 00 при успешном завершении команды «Рекалибровать» или «Установить». Признак IC = 01 при неуспешном завершении команды «Рекалибровать» или «Установить». Команда не имеет фазы выполнения и данные результата читаются сразу после выдачи команды.

Получить состояние устройства (Sense Drive Status). Данная команда предназначена для определения ST3, содержащего значения сигналов по некоторым линиям контроллер - накопитель. Команда не имеет фазы выполнения и данные результата читаются сразу после выдачи команды.

	ЧТЕНИЕ/ ЗАПИСЬ	ДАННЫЕ D7 D6 D5 D4 D3 D2 D1 D0	ПРИМЕЧАНИЕ
ФАЗА КОМ.	W W	0 0 0 0 0 1 0 0 0 0 0 0 0 HDS 0 DS	КОД КОМАНДЫ
РЕЗ.	R	ST3	ИНФОРМАЦИЯ СОСТОЯНИЯ

Определить (Specify). Данная команда предназначена для программирования внутренних таймеров контроллера. SRT определяет время между сигналами управления перемещением головок с дорожки на дорожку. Этот интервал должен быть не меньше, чем время перемещения головок, определенное для конкретного дисковода. HUT определяет интервал между завершением операций чтения и записи и разгрузкой головок. HLT определяет время загрузки головок перед началом операции чтения и записи. ND определяет режим работы контроллера с DMA. Единичное значение этого бита определяет режим работы без DMA. Команда не имеет фазы выполнения и фазы результата.

Для контроллеров, совместимых с 8272A (т. е. не имеющих встроенного регистра конфигурации), временные интервалы, задаваемые значениями HUT, SRT и HLT, постоянны и не за-

	ЧТЕНИЕ/ ЗАПИСЬ	ДАННЫЕ D7 D6 D5 D4 D3 D2 D1 D0	ПРИМЕЧАНИЕ
ФАЗА КОМАНДЫ.	W W W	0 0 0 0 0 1 0 0 ====SRT==== ====HUT==== ==========HLT========= ND	КОД КОМАНДЫ АТРИБУТЫ КОМАНДЫ

висят от текущей скорости передачи данных. Временные интервалы, задаваемые значениями HUT, SRT и HLT, изменяются следующим образом:

- HUT - с шагом 16 мс в диапазоне 16-240 мс (1 = 16 мс, 2=32мс,...,0Fh=240мс);
- SRT - с шагом 1 мс в диапазоне 1-16 мс (0Fh = 1 мс, 0Eh = 2 мс,...1 = 16 мс);
- HLT - с шагом 2 мс в интервале 2-254 мс (1 = 2 мс, 2 = 4 мс,... 7Fh=254мс).

У современных контроллеров (например: Intel 82077) временные интервалы, задаваемые значениями SRT, HUT и HLT зависят от значения скорости передачи данных, установленной в регистре конфигурации. Эти значения в миллисекундах для соответствующих величин скоростей приведены в таблицах 4.6, 4.7 и 4.8.

Таблица 4.6. Значения HUT

Значение HUT	Временной интервал (мс)		
	Скорость 500К	Скорость 300К	Скорость 250К
0	256	426	512
1	16	26.7	32
. . .	. . .	. . .	. . .
0Eh	224	373	448
0Fh	240	400	480

Таблица 4.7. Значения SRT

Значение SRT	Временной интервал (мс)		
	Скорость 500К	Скорость 300К	Скорость 250К
0	16	26.7	32
1	15	25	30
. . .	. . .	. . .	. . .
0Eh	2	3.33	4
0Fh	1	1.67	2

Таблица 4.8. Значения HLT

Значение HLT	Временной интервал (мс)		
	Скорость 500К	Скорость 300К	Скорость 250К
00h	256	426	512
0lh	2	3.3	4
02h	4	6.7	8
. . .	. . .	. . .	. . .
7Eh	252	420	504
7Fh	254	423	508

Для многих команд контроллера требуется передача ряда параметров. Рекомендуемые значения ряда параметров команд приведены в таблице 4.9.

Таблица 4.9. Рекомендуемые параметры команд

Тип накопителя	Тип диска	SRT+ HUT	HLT	SSEL	GPL	
					R/W	F
5.25"/360	360К	DFh	02h	02h	2Ah	50h
5.25"/1.2	360К	DFh	02h	0lh	2Ah	50h
5.25"/1.2	1.2M	DFh	02h	00h	IBh	54h
3.57720	720К	DFh	02h	02h	2Ah	50h
3.5"/1.44	720К	DFh	02h	02h	2Ah	50h
3.571.44	1.44M	AFh	02h	00h	1Bh	6Ch

В таблице 4.9 используются следующие обозначения:

- SSEL - скорость передачи данных между накопителем и контроллером, определяемая в регистре конфигурации (CCR);
- R/W - значение GPL для чтения и записи;
- F - значение GPL для форматирования.

4.4. Рекомендации по программированию на уровне регистров

Режим работы с DMA или без DMA

Контроллер гибких дисков допускает два режима работы: с контроллером DMA или без него. В режиме работы без DMA допускается работа с прерываниями или без прерываний.

Следует отметить, что на ряде современных компьютеров контроллер подключен таким образом, что не допускает работу без DMA.

Для задания режима работы с DMA следует установить в регистре управления бит DMA в состояние 1, а бит ND в команде «Определить» в состояние 0. При работе в режиме DMA перед выдачей команды, требующей передачу данных во время фазы исполнения, необходимо запрограммировать второй канал DMA. Завершение команды будет передаваться через прерывание от контроллера гибких дисков.

Чтобы определить режим работы без DMA с прерываниями, следует установить в регистре управления бит DMA в состояние 1, а бит ND в команде «Определить» в состояние 1. При работе в этом режиме контроллер будет генерировать прерывание при необходимости передачи каждого байта во время фазы выполнения команды. Для завершения команды программа должна принудительно обеспечить передачу сигнала TC.

Для задания режима работы без DMA и без прерываний следует установить в регистре управления бит DMA в состояние 0, а бит ND в команде «Определить» в состояние 1. При работе в данном режиме для передачи данных во время фазы выполнения программа должна контролировать состояние бит RQM и DIO в основном регистре состояния и, соответственно, передавать или принимать данные.

Для определения различия между фазой выполнения и фазой результата необходимо контролировать бит NDMA в основном регистре состояния. Если контроллер находится в режиме выполнения команды, то этот бит устанавливается в 1. Для завершения команды программа должна принудительно обеспечить передачу сигнала TC.

Следует отметить, что работа в режиме без DMA возможна только на достаточно быстрых машинах. Реально, работа без DMA применялась только на IBM PC Jr, когда во время работы с гибкими

дисками закрывались все прерывания. Практически, работать без DMA рекомендуется в режиме без прерывании, т. е. в режиме опроса. Это объясняется тем, что процедура обработки прерывания более длительная, чем процедура опроса состояния контроллера.

Особенности работы с регистром данных

Запись данных в контроллер через регистр данных возможна только в случае, когда в основном регистре состояния бит RQM установлен в 1, а бит DIO - в 0.

Чтение данных из контроллера через регистр данных возможно только в случае, когда в основном регистре состояния бит RQM установлен в 1, а бит DIO - в 1.

Таким образом, сначала надо проверить эти условия и только при **их** выполнении осуществлять передачу данных.

Например, для чтения каждого байта из контроллера необходимо выполнить следующую последовательность команд.

```
;
; Ждать готовность контроллера
;
            mov     dx,3f4h         ;Загрузить адрес основного регистра состояния
InOataWait: in      al.dx           ;Считать состояние
            and     a1,11000000Ь    ;Выделить биты готовности
            cmp     a1,11000000Ь    ;Контроллер даст данные?
            Jne     InDataWait      ;Нет, ждать состояние
;
;Прочитать байт
;
            inc     dx              ;Получить адрес регистра данных
            in      al,dx           ;Прочитать данные из регистра данных
```

Между операциями чтения и записи регистра данных во время командной фазы и фазы результата необходима временная задержка для выполнения внутренних операций контроллера. Минимально допустимая задержка называется временем восстановления БИС контроллера и составляет 23 мксек.

Особенности обработки прерываний для компьютеров PS/2 с шиной MCA

На компьютерах класса PS/2 с шиной MCA контроллер прерываний работает в режиме «уровня», т. е. прерывание возникает не по изменению уровня сигнала на соответствующей линии контроллера прерываний, а условие прерывания имеет место всегда, пока установлен высокий уровень на соответствующей линии контроллера прерываний.

Поэтому обработка прерывания от контроллера гибких дисков не должна завершаться командой завершения прерывания (EOI), так как контроллер гибких дисков сохраняет запрос на прерывание после того, как оно произошло. Запрос на прерывание обычно снимается контроллером гибких дисков после выдачи команд IN/OUT в порты контроллера. Команда завершения прерывания (EOI) в контроллер прерываний должна выдаваться только тогда, когда бит INT в регистре состояния A контроллера гибких дисков станет равным 0.

Начальный программный сброс

Начальный программный сброс необходимо выполнять в случае возникновения ошибок, а также перед началом работы, чтобы гарантировать начальное состояние контроллера. При сбросе происходит очистка всех внутренних регистров контроллера, кроме регистров, заданных командой «Определить».

Сброс контроллера осуществляется в следующей последовательности:

1. Устанавливается в 0 бит 2 регистра DOR (остальные биты устанавливаются в зависимости от выбранного номера накопителя).
2. Выполняется задержка на время восстановления БИС контроллера (не менее 23 мксек).
3. Устанавливается в 1 бит 2 регистра DOR (остальные биты устанавливаются в зависимости от выбранного номера накопителя).
4. Выполняется ожидание прерывания от контроллера.
5. Четыре раза выполняется команда «Получить результат прерывания». Четырехкратное выполнение команды объясняется тем, что контроллер дает состояние сброса для четырех устройств. После каждой команды «Получить результат прерывания» в регистре ST0 должно содержаться двоичное значение 110000xx, где xx принимает последовательно значения 00, 01, 10 и 11, т. с. указывает номер устройства. Наличие других значений указывает на неисправность контроллера.

Если производилось включение мотора, то необходимо обеспечить ожидание времени раскрутки.

Для накопителей, используемых в компьютерах фирмы IBM, это время составляет не менее 500 мс.

После начального сброса всегда необходимо выполнить команду «Рекалибровать» для установки начального положения головок, так как после сброса внутренний регистр контроллера, содержащий номер текущего цилиндра, находится в неопределенном состоянии. При использовании накопителя с 80 цилиндрами первая команда «Рекалибовать» может дать ошибку, если головки находились за цилиндром 77 (контроллер выдает 77 сигналов управления перемещением головок). В этом случае следует повторить команду «Рекалибровать». Повторная ошибка указывает на неисправность накопителя или контроллера.

Особенности использования других команд.

При практическом программировании следует учитывать особенность управления перемещением головок при работе с дискетами 360К в дисководе 1.2М. Дисковод 1.2М обеспечивает 80 цилиндров, а на дискете 360К используются только 40. Поэтому при выполнении команды «Установить» следует выполнять установку с учетом пропуска одного цилиндра.

Этот механизм получил название «двойной шаг». Например, для работы с цилиндром 1 дискеты 360К необходимо выполнить установку на цилиндр 2. Таким образом номер реального цилиндра для установки вычисляется по формуле:

(номер цилиндра для работы) - 2

Следует также напомнить, что после команд «Установить» или «Рекалибровать» перед командами чтения/записи необходимо обеспечить задержку длительностью 15-20 мс для успокоения головок после движения.

После завершения сброса, а также команд «Установить» и «Рекалибровaты» результат завершения команды должен быть получен по команде «Снять состояние прерывания». После выполнения других команд, имеющих фазу результата, должно быть прочитано 7 байтов состояния (в соответствии с описаниями команд).

4.5.Пример программирования контроллера гибких дисков на уровне регистров

```
;Демонстрируются следующие функции:
;       1. Инициализация контроллера и установка параметров для работы с дискетой 5.257360К в
;       накопителе с номером 0, обеспечивающим максимальную емкость 1.2М.
;       2. Форматирование поверхности 0 цилиндра 1 на 9 секторов.
;       3. Запись двух первых секторов.
;       4. Чтение трех первых секторов.
code                    segment
                assume cs:code,ds:code
; ===========================================================================
;       область данных
; ===========================================================================
;
;    рабочая область для форматирования, чтения, записи
;
WorkArea        db      512*3*2-1 dup(0) ; память для трех секторов и выравнивания в пределах
                                         ; страницы DMA
WorkOff         dw      0                ; смещение начала рабочей области, выровненной в
                                         ; пределах страницы. DMA
FDCResult       label   byte             ; область данных для результата выполнения операций
                                         ; контроллера
ST0             db      0                ; регистр состояния 0
ST1             db      0                ; регистр состояния 1
ST2             db      0                ; регистр состояния 2
ResCyI          db      0                ; цилиндр
ResHead         db      0                ; голова
ResNum          db      0                ; номер сектора
ResSize         db      0                ; m код длины сектора
FDCResultLen    equ     $-FDCResult      ; длина поля результата
FDCOpCode       db      0                ; код команды контроллера
FDCErrCode      db      0                ; код ошибки
FDCOutAreaLen   equ     $-FDCResult      ; длина поля
```

```
;
; значения в поле FDCErrCode
;
FDCTimeOut equ     1                ; нет прерывания
FOCTimeFail equ    2                ; не ответил за положенное время
FDCBadData equ     3                ; остались данные в контроллере
FDCComErr  equ     4                ; другая ошибка
;
; области сохранения прерываний
;
Save08     label   dword            ; прерывание от таймера
Save08Off  dw      0                ; смещение
Save08Seg  dw      0                ; сегмент
;
Save0E     label   dword            ; прерывание от контроллера  гибких дисков
Save0EOff  dw      0                ; смещение
Save0ESeg  dw      0                 ; сегмент
;
IntFlags   db      0                ; признаки прерываний
IntPres    equ     80h              ; есть прерывание от контроллера
IntMCA     equ     40h              ; требуется команда завершения прерывания для шины MCA
PresMCA    equ     20h              ; шина MCA установлена
ResuttOut  label   byte             ; сообщение о результате
           db      'ST0'
FiretOut   db      ' '
ToMakeSkipLen db   0dh.0ah
           db      'ST1'
SkipOverTWe equ    $-ToMakeSkipLen ; длина до следующей переменной
           db      ' ',0dh,0ah,'ST2     '
           db      ' '.0dh.0ah.'C       '
           db      ' ',0dh,0ah,'H       '
           db      ' ',0dh,0ah,'R       '
           db      ' ',0dh,0ah,'N       '
           db      ' ',0dh,0ah,'OP      '
           db      ' ',0dh.0ah'RES     ',' ',0dh,0ah,'$'
GoodMsg    db      'Успешное завершение'0dh,0ah,'$'
BadMsg     db      'Неуспешное завершение',0dh,0ah,'$'
RtCounter  dw      0                ; счетчик интервалов таймера
;
; ============================================
;  основная процедура
; ============================================
main       proc
           push    cs               ; установим адресацию
           pop     ds
;
; проверим наличие шины MCA (микроканал)
;
           mov     ah.0c0h          ; получить характеристики машины
           int     15h
           jc      NonMCA           ; нет микроканала
;
; ES:BX - указывают на список характеристик машины; бит 1 байта по смещению 5 содержит признак
; наличия микроканала
           test    byte ptr es:[bx][5],00000010b ; есть микроканал?
           jz      NonMCA           ; нет, выход
           or      IntFlags,PresMca ; установим признак наличия MCA
; ================================================================================
;       определим адрес начала буфера в пределах страницы DMA
; ================================================================================
NonMCA:    mov     ax,offset WorkArea ; смещение рабочей области
           mov     bx,cs            ; сегмент для определения составляющей смещения
           mov     cl,4
           shl     bx,cl            ; составляющая смещения в сегменте
           add     bx,ax            ; полное смещение
           jc      WorkNotCross     ; переехали страницу -не беспокоимся
           mov     cx,bx            ; запомним полное смещение
           add     bx,512*3-1       ; размер буфера для трех секторов
```

```
                jnc     WorkNotCross    ; не пересекли страницу
                sub     ax,cx           ; добавим к смещению сколько до пересечения границы
                                        ; страницы
WorkNotCross:
                mov     WorkOff,ax      ; запомним смещение
; ================================================================================
;       установим обработчики прерываний
; ================================================================================
                sub     ax,ax           ; адресация векторов прерываний
                mov     es,ax
;
; запомним адреса обработчиков
;
                mov     ax,word ptr es:[08h*4]      ; смещение для таймера
                mov     bx,word ptr es:[08h*4+2]    ; сегмент для таймера
                mov     cx,word ptr es:[0eh*4]      ; смещение для гибкого диска
                mov     dx,word ptr es:[0eh*4+2]    ; сегмент для гибкого диска
                mov     Save08Off,ax    ; запомним смещение для таймера
                mov     Save08Seg,bx    ; запомним сегмент для таймера
                mov     Save0Eoff,cx    ; запомним смещение для гибкого диска
                mov     Save0ESeg,dx    ; запомним сегмент для гибкого диска
                cli                     ;запрет прерываний для установки новых векторов
                mov     word ptr es:[08h*4], offset New08
                mov     word ptr es:[08h*4][2],cs
                mov     word ptr es:[0eh*4],offset New0E
                mov     word ptr es:[0eh*4][2],cs
                sti                     ; разрешить прерывания
                push    cs              ; адресация по ES
                pop     es
; ------------------------------------------------------------------------------------------------------------
;       установим скорость передачи данных для дискеты
; ------------------------------------------------------------------------------------------------------------
                mov     a1,1            ; скорость передачи данных 300
                mov     dx,03f7h        ; регистр скорости
                out     dx,al           ; установим скорость
                jmp     short $+2
; ================================================================================
;       выполним начальный программный сброс контроллера
; ================================================================================
                mov     FDCOpCode,0     ; выполняем сброс
                cli                     ; закроем маски
;
; сначала сброс
;
                mov     al,00011000b    ; разрешение DMA-прерываний сброс контроллера включить
                                        ; мотор устройства 0 выбрать устройство 0
                mov     dx,03f2h        ; регистр управления
                out     dx,al           ; передадим данные в контроллер
                jmp     short $+2
                mov     bx,29           ; время ожидания в циклах таймера не менее 23 мкс
                call    WaitTimer2      ; ждем не менее 23 мкс
                mov     a1,00011100Ь    ; разрешение DMA-прерываний работа контроллера включить
                                        ; мотор устройства 0 выбрать устройство 0
                out     dx,al           ; передадим данные в контроллер
                jmp     short $+2
                and     IntFlags,255-IntPres-IntMCA         ; сбросим состояние прерывания
                sti                     ; откроем маски
                call    Waitint         ; ждем прерывание
                jc      FatalErr        ; не дождались
; ================================================================================
;       снимем прерывание и проверим завершение для 4-х устройств
; ================================================================================
                mov     cx,4            ; от четырех устройств
                mov     dl,11000000b    ; маска статуса прерывания
ResetSens:      call    Sensl           ; снимем состояние прерывания
                jc      FatalErr        ; неуспешно
                mov     ah,ST0          ; статус регистр
                cmp     ah,dl           ; сброс завершен?
```

```
                jne     FatalErr        ; нет, что-то другое
                inc     dl              ; следующее устройство
                loop    ResetSens       ; повторим снятие состояния
    ;
================================================================================
    ;       ждем раскрутки мотора
    ;
================================================================================
                mov     cx, 1000/55     ; время раскрутки мотора
                call    WaitTime        ; ждем
    ;
================================================================================
    ;       выполним команду Specify
    ;
================================================================================
                mov     ah,03h          ; команда SPECIFY
                mov     FOCOpCode,03h   ; запомним код команды SPECIFY
                call    OutFDC          ; передадим в контроллер
                jc      FatalErr        ; неуспешно
                mov     ah,0DFh         ; SRT + HUT для 360 в 1.2
                call    OutFDC          ; передадим в контроллер
                jc      FatalErr        ; неуспешно
                mov     ah,02h          ; HLT + ND
                call    OutFDC          ;  передадим в контроллер
                jc      FatalErr        ; неуспешно
    ; ============================================================================
    ;       выполним команду Recalibrate (обязательно после сброса)
    ; ============================================================================
                mov     FDCOpCode,07h   ; запомним код команды Recalibrate
                mov     cx,2            ; количество повторов ; Примечание;
                                        ; !! для дисководов с 80 цилиндрами
                                        ; !! за одну операцию головки могут ;
                                        ; !! не встать на цилиндр 0
    RecalRestart:
                cli                     ; закроем маски
                mov     h,07h           ; команда Recalibrate
                call    OutFDC          ; передадим в контроллер
                jc      FatalErr        ; неуспешно
                sub     ah,ah           ; устройство 0
                call    OutFDC          ; передадим в контроллер номер устройства
                jc      FatalErr        ; неуспешно
                and     IntFlags,255-IntPres-IntMCA ; сбросим состояние  прерывания
                sti                     ; откроем маски
                call    Waitint         ; ждем прерывание
                jc      FatalErr        ; не дождались
    ;
    ; снимем состояние прерываний
    ;
                call    Sensl           ; опросим прерывание
                jc      FatalErr        ; неуспешно
    ;
    ; проверим результат
    ;
                mov     ah,ST0          ; регистр состояния 0
                test    ah,00100000b    ; операция завершена?
                jz      RecalTry        ; нет, повторим операцию
                and     ah,11000000b    ; выделим биты типа завершения
                test    ah,11000000b    `; успешно?
                jz      RecalCompl      ; да, на выход
    RecalTry:   loop    RecalRestart    ; повторим
                jmp     FatalErr        ;все же неуспешно
    ;успешно сделали; подождем успокоения головок (для привода 1.2 время успокоения составляет 15
    ;мс)
    ;
    RecalCompI:
                mov     bx, 18750       ; время успокоения в циклах таймера 15000 мкс/0.8 мкс
```

```
                call    WaitTimer2      ; подождем
; ============================================================================
;       подготовим таблицу формата:
;       9 секторов длиной 2, цилиндр 1, поверхность 0
; ============================================================================
                mov     si,WorkOff      ; адрес рабочей области
                mov     cx,9            ; количество секторов
;
; подготовим заголовок
;
                mov     byte ptr ds:[si],2      ; размер сектора 2
                mov     byte ptr ds:[si+1],9 ; количество секторов 9
                mov     byte ptr ds:[si+2],80           ; размер GAP 3
                mov     byte ptr ds:[si+3],0F6h         ; символ-заполнитель
                sub     al,al                   ; для номеров секторов
PrepForFormat:
                inc     al                      ; увеличим номер сектора
                add     si,4                    ; увеличим адрес в таблице формата
                mov     byte ptr ds;[si],1      ; цилиндр 1
                mov     byte ptr ds:[si+1],0 ; поверхность 0
                mov     byte ptr ds:[si+2],al           ; номер сектора
                mov     byte ptr ds:[si+3],2            ; код длины сектора
                loop    PrepForFormat   ; заполним таблицу для секторов
; ============================================================================
;       установим головки для форматирования цилиндра 1; для цилиндра 1 и ; дискеты 360/1.2
;       необходимо  реально установиться на цилиндр 2
; ============================================================================
                cli
                mov     ah,0fh          ; команда SEEK
                mov     FDCOpCode,0fh   ; запомним код команды SEEK
                call    OutFDC          ; передадим в контроллер
                jc      FatalErr        ; неуспешно
                sub     ah, ah          ;устройство 0 голова 0
                call    OutFDC          ; передадим в контроллер
                jc      FatalErr        ; неуспешно
                mov     ah,2            ; цилиндр 2
                call    OutFDC          ; передадим в контроллер
                jc      FatalErr        ; неуспешно
                and     IntFlags,255-IntPres-IntMCA ; сбросим прерывание
                sti                     ; откроем маски
                call    Waitint         ; ждем прерывание
                jc      FatalErr        ; не дождались
;
; получим результат
;
                call    Sensl           ; опросим прерывание
                jc      FatalErr        ; неуспешно
;
; проверим результат
;
                mov     ah,ST0          ; регистр состояния 0
                test    ah,00100000b    ; операция завершена?
                jz      FatalErr        ; нет, ошибка
                and     ah,11000000b    ; выделим биты типа завершения
                test    ah,11000000b    ; успешно?
                jnz     FatalErr        ; нет, ошибка
                cmp     ST1,2           ; подходящий цилиндр?
                jne     FatalErr        ; нет, ошибка
;
; успешно сделали; подождем успокоения головок (для привода 1.2 время ; успокоения составляет 15 мс
;
                mov     bx, 18750       ; время успокоения в циклах таймера 15000 мкс/0.8 мкс
                call    WaitTimer2      ; подождем
; ============================================================================
;       форматируем цилиндр
; ============================================================================
;
; установить DMA на таблицу формата
```

```
;
                mov     di,WorkOff          ; смещение рабочей области
                add     di,4                ; начало таблицы формата
                mov     cx,9*4              ; длина таблицы формата:
                                            ; 9 секторов * описатель сектора  (4 байта)
                mov     al,04Ah             ; код оперции DMA - передаем в FDC
                call    SetDMA              ; инициируем DMA
                mov     FDCOpCode,04dh ; запомним код команды Format
;
; передадим все команды формата
;
                cli                         ; закроем маски
                mov     ah,04dh             ; команда форматирования
                call    OutFDC              ; передадим в контроллер
                jc      FatalErr            ; неуспешно
                sub     ah,ah               ; накопитель 0 поверхность 0
                call    OutFDC              ; передадим в контроллер
                jc      FatalErr            ; неуспешно
                mov     cx,4                ; количество данных в заголовке таблицы формата
                mov     si,WorkOff          ; заголовок таблицы формата
                eld
FormatTableHeader:
                lodsb                       ; байт из заголовка
                mov     ah,al               ; для передачи в контроллер
                call    OutFDC              ; передадим в контроллер
                jc      FatalErr            ; неуспешно
                loop    FormatTableHeader           ; повторим для всех параметров формата
                and     IntFlags,255-IntPres-IntMCA ; сбросим состояние прерывания
                sti                         ; откроем маски
                call    Waitint             ; ждем прерывание
                jc      FatalErr            ; не дождались
;
;получим результат
;
                call    InFDC               ; прочитаем данные из FDC
                jc      FatalErr            ; неуспешно
                mov     ah,ST0              ; регистр состояния 0
                and     ah,11000000b        ; выделим состояние завершения
                or      ah,ah               ; успешно?
                jnz     FatalErr            ; нет, ошибка
; ===========================================================================
;       запишем два сектора, начиная с сектора 1
; ===========================================================================
;
; заполним буфер значениями для записи
;
                mov     ax,055aah           ; значение для записи в два сектора
                mov     cx,512
                mov     di,WorkOff          ; адрес области записи
                cld
                rep stosw                   ; заполним область записи
;
;установить DMA на рабочую область
;
                mov     di,WorkOff          ; смещение рабочей области
                mov     cx,512*2            ; длина двух секторов
                mov     al,04Ah             ; код оперции DMA - передаем в FDC
                call    SetDMA              ; инициируем DMA
                mov     FDCOpCode,045h ; запомним код команды Запись
;
; передадим все команды записи
;
                cli                         ; закроем маски
                mov     ah,045h             ; команда записи
                call    OutFDC              ; передадим в контроллер
                jc      FatalErr            ; неуспешно
                sub     ah,ah               ; накопитель 0 поверхность 0
                call    OutFDC              ; передадим в контроллер
                jc      FatalErr            ; неуспешно
```

```
;
; перешлем данные сектора
;
                mov     ah,1                ; цилиндр 1
                call    OutFDC              ; передадим в контроллер
                jc      FatalErr            ; неуспешно
                mov     ah,0                ; поверхность 0
                call    OutFDC              ; передадим в контроллер
                jc      FatalErr            ; неуспешно
                mov     ah,1                ; сектор 1
                call    OutFDC              ; передадим в контроллер
                jc      FatalErr            ; неуспешно
                mov     ah,2                ; код длины 2
                call    OutFDC              ; передадим в контроллер
                jc      FatalErr            ; неуспешно
                mov     ah,9                ; количество секторов 9
                call    OutFDC              ; передадим в контроллер
                jc      FatalErr            ; неуспешно
                mov     ah,02Ah             ; GAP чтения/записи
                call    OutFDC              ; передадим в контроллер
                jc      FatalErr            ; неуспешно
                mov     ah,0FFh             ; длина передаваемых данных FF
                call    OutFDC              ; передадим в контроллер
                jc      FatalErr            ; неуспешно
                and     IntFlags,255-IntPres-IntMCA ; сбросим состояние прерывания
                sti                         ; откроем маски
                call    Waitint             ; ждем прерывание
                jc      FatalErr            ; не дождались
;
; получим результат
;
                call    InFDC               ; прочитаем данные из FDC
                jc      FatalErr            ; неуспешно
                mov     ah,ST0              ; регистр состояния 0
                and     ah,11000000b        ;выделим состояние завершения
                or      ah,ah               ; успешно?
                jnz     FatalErr            ;нет, ошибка
; ===================================================================
;       прочитаем три сектора, начиная с сектора 1
; ===================================================================
;
;установить DMA на рабочую область
;
                mov     di,WorkOff          ; смещение рабочей области
                mov     cx,512*3            ; длина трех секторов
                mov     al,046h             ; код операции DMA - передаем из FDC
                call    SetDMA              ; инициируем DMA
                mov     FDCOpCode,046n ; запомним код команды чтения
;
;передадим все команды чтения
;
                cli                         ; закроем маски
                mov     ah,046h             ; команда чтение
                call    OutFDC              ; передадим в контроллер
                jc      FatalErr            ; неуспешно
                sub     an,ah               ; накопитель 0 поверхность 0
                call    OutFDC              ; передадим в контроллер
                jc      FatalErr            ; неуспешно
;
; перешлем данные сектора
;
                mov     ah,1                ; цилиндр 1
                call    OutFDC              ; передадим в контроллер
                jc      FatalErr            ; неуспешно
                mov     ah,0                ; поверхность 0
                call    OutFDC              ; передадим в контроллер
                jc      FatalErr            ; неуспешно
                mov     ah,1                ; сектор 1
                call    OutFDC              ; передадим в контроллер
```

```
                jc      FatalErr        ; неуспешно
                mov     ah,2            ; код длины 2
                call    OutFDC          ; передадим в контроллер
                jc      FatalErr        ; неуспешно
                mov     ah,9            ; количество секторов 9
                call    OutFDC          ; передадим в контроллер
                jc      FatalErr        ; неуспешно
                mov     ah,02Ah         ; GAP чтения/записи
                call    OutFDC          ; передадим в контроллер
                jc      FatalErr        ; неуспешно
                mov     ah,0FFh         ; длина передаваемых данных
                call    OutFDC          ; передадим в контроллер
                jc      FatalErr        ; неуспешно
                and     IntFlags,255-IntPres-IntMCA ; сбросим прерывание
                sti                     ; откроем маски
                call    Waitint         ; ждем прерывание
                jc      FatalErr        ; не дождались
    ;
    ;получим результат
    ;
                call    InFDC           ; прочитаем данные из FDC
                jc      FatalErr        ; неуспешно
                mov     ah.ST0          ; регистр состояния 0
                and     ah,11000000b    ; выделим состояние завершения
                or      ah,ah           ; успешно?
                jz      GoodExrt        ; да, на выход
; ==========================================================================
    ;       ошибка при выполнении команды
    ; ======================================================================
    FatalErr:   sti                     ; возможно маски были закрыты
                cmp     FDCErrCode,0
                jne     GoodExit
                mov     FDCErrCode,FDCComErr
    ; ========================================================================
    ;       восстановим состояние контроллера и вектора прерываний

    GoodExrt:   mov     al,00001100Ь    ; разрешение DMA-прерываний ; работа контроллера
                                        ; выключить мотор устройства 0 ; выбрать устройство 0
                mov     dx,03f2h        ; регистр управления
                out     dx,al           ; передадим в контроллер
                jmp     short $+2
    ;
    ; восстановим адреса обработчиков
    ;
                sub     ax,ax           ; адресация вектора прерываний
                mov     es,ax
                cli
                mov     ax,Save08Off    ; возьмем смещение для таймера
                mov     bx,Save08Seg    ; возьмем сегмент для таймера
                mov     cx.Save0EOff    ; возьмем смещение для гибкого диска
                mov     dx.Save0ESeg    ; возьмем сегмент для гибкого диска
                mov     word ptr es:[08h*4],ax      ; смещение для таймера
                mov     word ptr es:[0eh*4+2],bx    ; сегмент для таймера
                mov     word ptr es:[0eh*4],cx      ; смещение для гибкого диска
                mov     word ptr es:[0eh*4+2],dx    ; сегмент для гибкого диска
                sti
                push    cs              ; адресация по ES
                pop     es
    ; ======================================================================
    ; выведем результат
    ; ======================================================================
                mov     si.offset FDCResult         ; результат
                mov     di,offset FirstOut          ; куда складывать
                mov     cx,FDCOutAreaLen            ; сколько выводить
                cld
    HexToCharLoop:
                lodsb
                call    HexChar                     ; преобразовать в симв.форму
```

```
                stosw
                add     di,SkipOverTitle        ; следующее поле результата
                loop    HexToCharLoop           ; заполним весь результат
                mov     dx,offset ResultOut     ; что выводим
                mov     ah,09h                  ; функция вывода строки
                int     21h                     ; выполним
                mov     dx,offset GoodMsg       ; сообщение об успехе
                cmp     FDCErrCode,0            ; есть ошибка?
                je      ShowMsg                 ; нет, выводим сообщение
                mov     dx,offset BadMsg        , сообщение о неуспехе
ShowMsg:        mov     ah,09h
                int     21h
                mov     ax,4c00h
                int     21h
main            endp
;
; ***********************************Процедуры*************************************
;               Ожидание целого числа циклов таймера
;       Вход:   CX - количество циклов таймера
; =============================================================================
WaitTime        proc
                push    ax
                push    bx
                push    cx
;
; установка таймера
;
                in      al,61 h         ; остановим 2-й канал таймера
                and     al,0ffh-03h
                out     61h,al
                jmp     short $+2
                mov     al,10110100b    ; маска таймера
                out     43h,al          ; установим
                mov     al,0ffh         ; новый счетчик
                out     42h,al          ; запишем все 16 бит
                jmp     short $+2
                out     42h,al
                jmp     short $+2
                in      al,61 h         ; запустим 2-й канал таймера
                or      al,01 h
                out     61h,al
                jmp     short $+2
;
; цикл ожидания
;
                mov     bx,0ffffh       ; начальное значение для сравнения
WaitTimeTest:
                mov     al,84h          ; запомним счетчик канала 2
                out     43h,al
                in      al,42h          ; прочитаем все 16 бит
                xchg    ah,al
                in      al,42h
                xchg    ah,al
                cmp     ax,bx           ; перешли через 0?
                mov     bx,ax           ; значение счетчика для сравнения
                jb      WaitTimeTest    ; не перешли через 0 - ждем
                loop    WaitTimeTest    ; повторим для всех циклов
                pop     cx
                pop     Ьх
                pop     ax
                ret
WaitTime        endp
; =============================================================================
;       Обработчик прерывания от FDC
; =============================================================================
New0E           proc
                push    ax
                or      cs:IntFlags,IntPres     ; укажем прерывание
```

```
                test    cs:IntFlags,PresMCA     ; есть микроканал?
                jz      New0ENormalEOI          ; нет, обычное завершение
                or      cs:IntFlags,IntMCA      ; требуется завершение  прерывания для микроканала
                jmp     New0EIRet               ; на выход
;
;завершим прерывание для обычной шины
;
New0ENormalEOI:
                mov     al,20h                  ; сообщим контроллеру
                out     20h,al                  ; прерываний EOI
                jmp     short $+2
New0EIRet:      pop     ax
                iret
New0E           endp
; ============================================================================
;       Обработчик прерывания от таймера
; ============================================================================
New08           proc
                push    ax
                cmp     cs:RtCounter,0      ;  требуется уменьшать счетчик ; времени?
                je      New08IRet           ; нет, на выход
                dec     cs.RtCounter        ; уменьшим счетчик времени
New08IRet:
;
; отобразим время в область данных BIOS
;
                push    ds
                mov     ax,40h              ; область данных BIOS
                mov     ds,ax
                add     word ptr ds:[6ch],1 ; увеличим счетчик времени
                adc     word ptr ds:[6eh],0
                cmp     word ptr ds:[6eh],0018h ; есть переход через сутки?
                jne     New08Time           ; нет, переход
                cmp     word ptr ds:[6ch],00b0h ; есть переход через сутки?
                jne     New08Time           ; нет, переход
                mov     byte ptr ds:[70h],1 ; полночь
                mov     word ptr ds:[6ch],0 ; новый счетчик времени
                mov     word ptr ds:[6eh],0
New08Time:
                pop     ds
                mov     al,20h              ; пошлем контроллеру EOI
                out     20h,al
                jmp     short $+2
                pop     ax
                iret
New08           endp
; ============================================================================
;       Передача данных в контроллер
;       Вход:  AH - байт для передачи
;       Выход: CF=0 - успешно, CF=1 - неуспешно
; ============================================================================
OutFDC          proc
                push    ax                  ; запомним регистры
                push    bx
                push    ex
                push    dx
                mov     FDCErrCode,0        ; сбросим код ошибки
;
; ждем готовность контроллера к приему
;
                mov     dx,3f4h             ; адрес статус регистра
                mov     bl,2                ; счетчик циклов
                xor     cx,cx               ; каждый цикл
OutFDC1:        in      al,dx               ; прочитаем статус регистр
                and     al,0c0h             ; выделим направление и состояние передачи данных
                cmp     al,80h              ; контроллер готов принять данные?
                je      OutFDC2             ; да, передадим данные
                loop    OutFDC1             ; ждем состояние
                dec     bl                  ; уменьшим счетчик повторов
```

```
                jnz     OutFDC1             ; ждем состояние
                mov     FDCErrCode,FDCTimeFail  ; не готов принять данные за полный цикл
                stc                         ; укажем ошибку
                jmp     short OutFDCRet     ; на выход
;
; передадим байт в контроллер
;
OutFDC2:        inc     dx                  ; адрес регистра данных
                mov     al,ah               ; данные в AL
                out     dx,al               ; передадим
                jmp     Short $+2
;
; подождем время восстановления
;
                mov     bx,29               ; время ожидания в циклах.таймера не менее 23 мкс
                call    WaitTimer2          ; ждем не менее 23 мкс
                c1c                         ; все в порядке
OutFDCRet:
                call    TestEOIMCA          ; проверим EOI для MCA
                pop     dx                  ; восстановим регистры
                pop     cx
                pop     bx
                pop     ax
                ret
OutFDC          endp
; ============================================================================
;       Прием данных из контроллера
;       Выход: CF=0 - успешно, CF=1 - неуспешно
;
InFDC           proc
                push    ax                  ; запомним регистры
                push    bx
                push    ex
                push    dx
                push    di
                mov     di,offset FDCResult         ; адрес для данных от контроллера
                mov     FDCErrCode,0        ; сбросим код ошибки
                mov     dx,3f4h             ; адрес статус-регистра
                mov     bh,7                ; максимальное количество данных из контроллера
InFDCO:         mov     bl,2                ; количество циклов ожидания данных
                xor     cx,cx               ; каждый цикл
InFDd:          in      al,dx               ; читаем статус-регистр
                and     AL,11000000b        ; выделим направление и состояние  передачи данных
                cmp     AL,11000000b        ; контролер готов дать данные?
                je      InFDC2              ; да, заберем данные
                loop    InFDC1              ; ждем состяние
                dec     bl                  ; уменьшим счетчик повторов
                jnz     InFDC1              ; ждем состяние
                mov     FDCErrCode,FDCTimeFail ; не готов принять данные  за полный цикл
                stc                         ; укажем ошибку
                jmp     short InFDCRet      ; на выход
;
; прочитаем байт и положим на место
;
InFDC2:         inc     dx                  ; адрес регистра данных
                in      al,dx               ; данные в AL
                stosb                       ; запомним данные
                dec     dx                  ; адрес статус-регистра
;
; подождем время восстановления
;
                push    bx
                mov     bx,29               ; время ожидания в циклах таймера  не менее 23 мкс
                call    WaitTimer2          ; ждем не менее 23 мкс
                pop     bx
                in      al,dx               ; прочитаем статус-регистр
                test    al,00010000b        ; есть еще данные?
                jz      InFDCF0             ; нет, на выход
                dec     bh                  ; уменьшим количество данных
```

```
                jnz     InFDC0          ; еще можно читать данные
                stc                     ; ошибка - остались данные в контроллере, хотя счетчик равен 0
                mov     FDCErrCode,FDCBadData   ; остались данные
                jmp     short InFDCRet  ; на выход
InFDCF0:        clc                     ; все в порядке
InFDCRet:       call    TestEOIMCA      ; проверим, не требуется ли EOI для MCA
                pop     di              ; восстановим регистры
                pop     dx
                pop     cx
                pop     bx
                pop     ax
                ret
InFDC           endp
; ==========================================================================
;               Выдача EOI для микроканала
;       Примечание: флажки не меняются
; ==========================================================================
TestEOIMCA      proc
                push    ax
                push    dx
                pushf
;
; проверим необходимость EOI для MicroChannel
;
                test    IntFlags,IntMCA ; требуется EOI для MCA?
                jz      TestEOIMCARet   ; нет, переход
;
; проверим возможность завершения прерывания
;
                mov     dx,3f0h         ; регистр прерываний
                in      al,dx           ; прочитаем
                test    al,80h          ; хранит прерывание?
                jnz     TestEOIMCARet   ; да - нельзя выдавать EOI
;
; завершим предыдущее прерывание для микроканала
;
                mov     al,66h          ; выполним команду EOI
                out     20h,al
                jmp     Short $+2
                and     IntFlags,255-IntMCA     ; отменим выдачу EOI для MCA
TestEOIMCARet:
                popf
                pop     dx
                pop     ax
                ret
TestEOIMCA      endp
; ==========================================================================
;               Ожидание короткого интервала времени
; Вход:         BX - количество единиц таймера (0.8 мксек) для ожидания
; ==========================================================================
WaitTimer2      proc
                push    ax
                push    bx
                neg     bx              ; определим остат.счетчик в BX
                in      al,61h          ; остановим 2-й канал таймера
                and     al,0ffh-03h
                out     61h,al
                jmp     short $+2
                mov     al,10110100b    ; установим маску таймера
                out     43h,al
                mov     al,0ffh         ; новый счетчик
                out     42h,al          ; запишем все 16 бит
                jmp     short $+2
                out     42h,al
                jmp     short $+2
                in      al,61h          ; запустим канал 2 таймера
                or      al,01h
                out     61h,al
                jmp     short $+2
```

```
;
; ждем завершения времени
;
WaitTimer2Loop:
                mov     al,84h              ; запомним счетчик
                out     43h,al
                jmp     short $+2
                in      al,42h              ; прочитаем все 16 бит
                xchg    ah,al
                in      al,42h
                cmp     bx,ax               ; перешли через требуемое время?
                jb      WaitTimer2Loop      ; нет, ждем
                pop     bx
                pop     ax
                ret
WaitTimer2      endp
; ===========================================================================
;Выполнение команды SENSL
; ===========================================================================
Sensl           proc
                push    ax
                mov     ah,08h              ; команда SENSL
                call    OutFDC              ; передадим в контроллер
                jc      SenslRet            ; ошибка
                call    InFDC               ; снимем состояние
SenslRet:       pop     ax
                ret
Sensl           endp
; ===========================================================================
;       Ожидание прерывания от FDC
; ===========================================================================
Waitint         proc
                mov     RtCounter,2000/55 ; ждем 2 сек
                mov     FDCErrCode,0        ; очистим поле ошибки
WaitIntLoop:
                test    IntFlags,IntPres    ; есть прерывание?
                jnz     WaitIntCF0          ; да, на выход
                cmp     RtCounter,0         ; исчерпано время?
                jne     WaitIntLoop         ; нет, ждем
                mov     FDCErrCode,FDCTimeOut ; не дождались прерывания
                stc                         ; укажем ошибку
                jmp     short WaitIntRet    ; на выход
WaitIntCF0:     clc                         ; все в порядке
WaitIntRet:     ret
Waitint         endp
; ===========================================================================
;                       Процедура инициализации DMA
;       Вход:   AL = код операции DMA (4ah - передавать в FDC, 46h -передавать из FDC)
;                       ES:DI - адрес буфера
;                       CX - длина данных
; ===========================================================================

SetDMA          proc
                push    ax                  ; запомним регистры
                push    cx
                cli
;
; остановим второй канал
;
                push    ax
                mov     al,00000110b        ; остановим второй канал
                out     0ah,al
                jmp     short $+2
                pop     ax
;
; установим направление передачи DMA no AL и сбросим указатель последовательности байт
;
                out     0Ch,al              ; сбросим указатель
```

```
                jmp     short $+2
                out     0Bh,al          ; установим режим работы, переданный в AL
                jmp     short $+2
;
; вычислим адрес буфера
;
                push    cx              ; запомним длину данных
                mov     ax,es           ; адрес сегмента для сдвига
                mov     cl,4
                rol     ax,cl           ; сдвинем
                mov     ch,al           ; остаток от сдвига и данные
                and     al,0f0h         ; сбросим остаток сдвига
                add     ax,di           ; прибавим смещение
                adc     ch,0            ; добавим потерянную 1 передадим адрес
                out     4,al            ; передать младшую часть
                jmp     short $+2
                mov     al,ah           ;старшая часть
                out     4,al            ; передать старшую часть
                jmp     short $+2
                mov     al,ch           ; старшие четыре бита
                and     al.0fh          ; оставим только их
                out     81h,al          ; запишем регистр страницы
                jmp     short $+2
;
; запишем длину данных
;
                pop     ax              ; восстановим длину
                dec     ax              ; для DMA должно быть минус 1
                out     5,al            ; младший байт длины
                jmp     short $+2
                mov     al,ah           ; перенесем старший байт
                out     5,al            ; старший байт длины
                jmp     short $+2
;
; запустить канал 2 DMA
;
                mov     al,2            ; запустим второй канал
                out     0ah,al
                jmp     short $+2
                sti
                pop     cx              ; восстановим регистры
                pop     ax
                ret
SetDMA          endp
; ===========================================================================
;               Процедура преобразования двоичного числа в печатную форму
;       Вход:   AL - число в двоичном виде
;       Выход:  AX - число в печатной форме
; ===========================================================================
HexChar         proc
                push    bx
                xor      ah,ah
                mov      BL,16
                div     bl
                add     al,30h
                cmp     al,39h
                jna     Hex1
                add     al,7
Hex1:           add     ah,30h
                cmp     ah,39h
                jna     Hex2
                add     ah,7
 Hex2:          pop     bx
                ret
HexChar         endp
code                    ends
end                     main
```

4.6. Прерывания BIOS для работы с дисками на низком уровне.

Перечень функций прерывания BIOS для работы с дисками на низком уровне и описание общих принципов их использования были приведены в предыдущем разделе. В данном разделе приводится более подробное описание основных функций, используемых при работе в основном с гибкими дисками, и ряд дополнительных сведений по их использованию.

Прерывания BIOS нужны программисту в тех случаях, когда невозможно использовать функции DOS: для работы с гибкими дисками нестандартного формата и для восстановления поврежденной файловой структуры или стертых файлов. В отличие от DOS, функции BIOS предназначены для работы с физическими дисками, а не с их разделами (логическими дисками) — не забудьте об этом, задавая номер диска.

Для вызова дисковых функций BIOS используется прерывание Int 13h. После выполнения операции:

- в случае успешного завершения флаг переноса CF сбрасывается, в регистр AH заносится значение 0;
- в случае неудачи флаг CF устанавливается, а в регистр AH заносится код состояния дисковода, возможные значения которого приведены в таблицах 3.5 и 3.6 для гибких и жестких дисков соответственно).

ПРИМЕЧАНИЕ При выполнении операций с гибкими дисками ошибки могут возникать из-за того, что в момент обращения к диску мотор дисковода выключен либо не успел разогнаться до номинальной скорости (функции BIOS не имеют встроенного контроля скорости вращения диска). Операции чтения, записи, поиска и форматирования для гибкого диска в случае ошибки следует повторить трижды, каждый раз выполняя сброс. Если ошибка не устранена, то нужно выдать пользователю предупреждение и прервать выполнение операции.

Прерывание Int 13h, функция 00h: сброс дисковой системы

Функция предназначена для выполнения сброса (перевода в исходное состояние) контроллера диска и подключенных к нему дисководов. Ее необходимо вызывать после сбоев при выполнении операций чтения, записи, верификации или форматирования на гибком диске — перед повторением операции.

Перед вызовом прерывания требуется записать в регистры следующие значения:

- в AH ~ значение 00h;
- в DL — номер дисковода (00h-7Fh— гибкий диск, 80h-FFh— жесткий диск).

После завершения операции функция возвращает в регистре AH состояние дисковой системы.

ПРИМЕЧАНИЕ: При установленном бите 7 регистра DL выполняется общий сброс всех гибких и жестких дисков.

Прерывание Int 13h, функция 01 h: определить текущее состояние дисковой системы

Функция возвращает код завершения последней операции, выполненной на указанном дисководе. Перед вызовом прерывания требуется записать в регистры следующие значения:

- в AH — значение 0lh;
- в DL — номер дисковода (08h-7Fh — гибкий диск, 80h-FFh — жесткий диск).

После завершения операции функция возвращает в регистре AH состояние дисковой системы.

Прерывание Int 13h, функция 02h: читать сектор

Функция выполняет операцию считывания в заданную область оперативной памяти информации из одного или нескольких секторов диска.

Перед вызовом прерывания требуется записать в регистры следующие значения:

- в AH — значение 02h;
- в AL — число считываемых секторов (должно быть больше 0);
- в CH — номер цилиндра;
- в CL — номер начального сектора;
- в DH — номер головки;
- в DL — номер дисковода (08h-7Fh — гибкий диск, 80h-FFh — жесткий диск);
- в ES:BX — указатель на адрес буфера, в который производится считывание информации.

ПРИМЕЧАНИЕ: При использовании жестких дисков старшие два бита 10-разрядного номера цилиндра помещаются в старшие два бита регистра CL.

После завершения операции функция возвращает в регистре AH состояние дисковой системы. В случае успешного завершения операции будет возвращена следующая информация:

- в AL — число прочитанных секторов;
- в буфере — прочитанная с диска информация.

Прерывание Int 13h, функция 03h: записать сектор

Функция переписывает данные из заданной области оперативной памяти в один или несколько указанных секторов диска.

Перед вызовом прерывания требуется записать в регистры следующие значения:

- в AH — значение 03h;
- в AL — число записываемых секторов (должно быть больше 0);
- в CH — номер цилиндра;
- в CL — номер начального сектора;
- в DH — номер головки;
- в DL — номер дисковода (00h-7Fh — гибкий диск, 80h-FFh — жесткий диск);
- в E S: BX — указатель на адрес буфера, из которого производится считывание информации.

ПРИМЕЧАНИЕ: При использовании жестких дисков старшие два бита 10-разрядного номера цилиндра помещаются в старшие два бита регистра CL.

После завершения операции функция возвращает в регистре AH состояние дисковой системы. В случае успешного завершения операции в AL будет находиться число записанных секторов.

Прерывание Int 13h, функция 04h: проверить правильность записи

Функция предназначена для контроля правильности выполнения записи данных на диск путем считывания данных и проверки контрольного кода CRC.

Перед вызовом прерывания требуется записать в регистры следующие значения:

- в AH — значение 04h;
- в AL — число проверяемых секторов (должно быть больше 0);
- в CH — номер цилиндра;
- в CL — номер начального сектора;
- в DH — номер головки;
- в DL — номер дисковода (00h-7Fh — гибкий диск, 80h-FFh — жесткий диск);
- в ES:BX — указатель на адрес буфера, в который производится считывание информации.

ПРИМЕЧАНИЕ: При использовании жестки/дисков старшие два бита 10-разрядного номера цилиндра помещаются в старшие два бита регистра CL.

После завершения операции функция возвращает в регистре AH состояние дисковой системы, а в регистре AL — число проверенных секторов.

Прерывание Int 13h, функция 05h: форматировать дорожку гибкого диска

Функция предназначена для форматирования дорожек, то есть подготовки поверхности диска к выполнению операций чтения/записи данных.

Перед вызовом прерывания требуется записать в регистры следующие значения:

- в AH — значение 05h;
- в CH — номер цилиндра (дорожки);
- в DH — номер головки;
- в DL — номер дисковода (90h-7Fh);
- в ES: BX — указатель на адрес буфера, в котором содержится список адресных полей. Список состоит из 4-байтных элементов (формат которых описан в табл. 3.5) — по одному элементу на каждый создаваемый сектор.

После завершения операции функция возвращает в регистре AH состояние дисковой системы.

ВНИМАНИЕ

Нельзя применять функцию форматирования при работе с жесткими дисками: она устарела и ее использование может либо привести к повреждению поверхности диска (часть дорожек станет нечитаемой), либо вообще полностью вывести диск из строя.

Таблица 4.10 Формат элемента списка адресных полей

Смещение	Размер	Описание
00h	BYTE	Номер дорожки
0lh	BYTE	Номер головки
02h	BYTE	Номер сектора
03h	BYTE	Размер сектора (байт): 0 - 128; 1 - 256; 2 - 512; 3 - 1024

Прерывание Int 13h, функция 08h: получить параметры дисковода

Функция предназначена для определения параметров дисководов. Перед вызовом прерывания требуется записать в регистры следующие значения:

- в AH — значение 08h;
- в DL — номер дисковода (00h-7Fh — гибкий диск, 80h-FFh - жесткий диск).

После завершения операции функция возвращает в регистре AH состояние дисковой системы.

В случае успешного завершения операции функция возвращает в регистрах следующую информацию:

- в BL — код типа дисковода (код выдается только для гибких дисков, возможные значения кода описаны в табл. 4.11);
- в CH — младшие 8 разрядов максимального номера цилиндра;
- в CL — максимальный номер сектора (разряды 0-5) и два старших бита максимального номера цилиндра (разряды 6-7);
- в DH — максимальный номер головки;
- в DL — общее число дисководов в системе;
- в ES:DI — указатель на таблицу параметров гибкого диска (выдается только для гибких дисков).

Таблица 4.11. Значения кодов типа дисковода

Код	Тип дисковода
1	5,25", 360 Кбайт, 40 дорожек
2	5,25", 1,2Мбайт, 80 дорожек
3	3,5", 720 Кбайт, 80 дорожек
4	3,5", 1,44 Мбайт, 80 дорожек

Прерывание Int 13h, функция 0Dh: сброс контроллера жесткого диска

Функция выполняет инициализацию контроллера жесткого диска, не затрагивая (в отличие от функции 00h) контроллер гибких дисков. Головки заданного дисковода перемещаются при этом на нулевую дорожку.

Перед вызовом прерывания требуется записать в регистры следующие значения:

- в AH — значение 0Dh;
- в DL — номер жесткого диска (80h-FFh).

После завершения операции функция возвращает в регистре AH состояние дисковой системы.

Вызывать данную функцию следует при возникновении ошибок в работе контроллера (см. табл. 3.6) кодов состояния системы).

Прерывание Int 13h, функция 10h: проверить готовность жесткого диска к работе

Функция определяет готовность диска к выполнению операций ввода/вывода. Перед вызовом прерывания требуется записать в регистры следующие значения:

- в AH — значение 10h;
- в DL — номер жесткого диска (80h-FFh).

После завершения операции функция возвращает в регистре AH состояние дисковой системы.

Прерывание Int 13h, функция 11h: рекалибровка жесткого диска

Функция выполняет перемещение головок заданного дисковода на нулевую дорожку.

Перед вызовом прерывания требуется записать в регистры следующие значения:

- в AH — значение 11h;
- в DL — номер жесткого диска (80h-FFh).

После завершения операции функция возвращает в регистре AH состояние дисковой системы.

Вызывать данную функцию следует в случае возникновении сбоев в работе диска (см. табл.3.6).

Прерывание Int 13h, функция 16h: проконтролировать смену гибкого диска

Функция предназначена для обеспечения целостности данных на диске, то есть для защиты прикладной программы от внезапной смены пользователем гибкого диска. В дисковых дупликаторах (специальных программах, предназначенных для копирования гибких дисков) и инсталляторах (программах, обеспечивающих установку на компьютер прикладных пакетов) данная функция, наоборот, обеспечивает замену уже использованного гибкого диска на следующий.

Перед вызовом прерывания требуется записать в регистры следующие значения:

- в AH — значение 16h;
- в DL — номер дисковода гибких дисков (00h-7Fh).

После завершения операции функция возвращает в регистре AH результат проверки (значение 0 — диск не менялся, значение 06h — диск был заменен).

Прерывание Int 13h, функция 18h: установить тип носителя для форматирования

Функция позволяет установить основные параметры, используемые при форматировании диска. Эту функцию необходимо вызвать перед началом работы с функцией 05h, чтобы система BIOS могла установить корректное значение скорости передачи данных для используемого дисковода.

Перед вызовом прерывания требуется записать в регистры следующие значения:

- в AH — значение 18h;
- в CH — число цилиндров;
- в CL — число секторов на дорожке;
- в DL — номер дисковода (00h-7Fh).

После завершения операции функция возвращает в регистре AH состояние дисковой системы.

В случае успешного завершения операции функция возвращает в ES:DI указатель на таблицу параметров гибкого диска.

Векторы параметров дисководов

Несколько ячеек в таблице векторов прерываний хранят указатели на таблицы параметров дисководов (вместо адресов процедур обработки прерываний). Вектор 1Eh (то есть 32-разрядное слово, расположенное в оперативной памяти по адресу 0000:0078h) хранит таблицу параметров гибких дисков для дисковода (табл. 3.2). Векторы 41h (адрес 0000:0104h) и 46h (адрес 0000:0124h) хранят таблицы параметров жестких дисков (для диска 0 и диска 1 соответственно), формат которых показан в табл. 3.3.

ВНИМАНИЕ

Значения в этих таблицах устанавливаются функциями BIOS в процессе работы. Во избежание сбоев в работе компьютера не рекомендуется что-либо изменять в них непосредственно. Информацию из таблиц можно только читать, причем для современных жестких дисков эта информация вообще бесполезна (они работают в режиме линейной адресации секторов).

4.7. Пример использования функций BIOS для работы с гибкими дисками.

В листинге 4.1 приведена программа ShowFDDSector, позволяющая при помощи функций BIOS осуществлять последовательный просмотр секторов гибкого диска в ASCII-кодах. Перед тем как начать просмотр, программа просит пользователя установить гибкий диск в дисковод A:, считывает загрузочный сектор и извлекает из него информацию о формате гибкого диска (то есть о количестве рабочих поверхностей диска и числе секторов на дорожке). Пролистывание секторов осуществляется при помощи клавиш ↓ и ↑; для выхода из программы нужно нажать клавишу Esc. Для считывания секторов используется вспомогательная процедура ReadFDDSector.

Листинг 4.1. Просмотр секторов гибкого диска на дисководе A: при помощи функций BIOS

```
IDEAL
P386
LOCALS
MODEL MEDIUM

; Подключить файл мнемонических обозначений
; кодов управляющих клавиш
include "lst_1_03.inc"
; Подключить файл мнемонических обозначений цветов
include "lst_1_05.inc"

DATASEG
; Текстовые сообщения
Txt0 DB LIGHTCYAN,0,7,"ПРИМЕР ИСПОЛЬЗОВАНИЯ ФУНКЦИЙ "
     DB "BIOS ДЛЯ ЧТЕНИЯ ИНФОРМАЦИИ С ДИСКЕТЫ",0
     DB LIGHTGREEN,12,24
     DB "Установите дискету в дисковод A:",0
     DB YELLOW,24,29,"Нажмите любую клавишу",0
Txt1 DB LIGHTCYAN,0,10,"Просмотр секторов "
     DB "гибкого диска, находящегося в дисководе A:",0
     DB LIGHTGREEN,2,14,"Диск имеет   поверхность(и),"
     DB "    секторов на дорожке",0
```

```
     DB LIGHTCYAN,17,8,"Управляющие клавиши:",0
     DB YELLOW,24,27,"Нажмите управляющую клавишу",0
Txt2 DB 19,8,"Стрелка вниз – следующий сектор;",0
     DB 20,8,"Стрелка вверх – предыдущий сектор;",0
     DB 21,8,"Esc – выход.",0
Txt3 DB LIGHTGREEN,5,8
     DB "Дорожка N    Головка N    Сектор N    ",0
TErr DB LIGHTRED,12,20
     DB "Ошибка считывания сектора с дисковода A:",0
     DB YELLOW,24,29,"Нажмите любую клавишу",0

; Адрес считываемого сектора в режиме CHS
Cylinder    DB ?  ;номер цилиндра
Head        DB ?  ;номер головки
Sector      DB ?  ;номер сектора
; Предельные значения координат
MaxCylinder DB 79 ;?  ;максимальный номер цилиндра
MaxHead     DB 1  ;?  ;максимальный номер головки
MaxSector   DB 18 ;?  ;максимальный номер сектора

; Область памяти для хранения прочитанного сектора
SectorDataBuffer DB 512 DUP (?)
ENDS

SEGMENT sseg para stack 'STACK'
        DB 400h DUP(?)
ENDS

CODESEG
;*****************************
;* Основной модуль программы *
;*****************************
PROC ShowFDDSector
        mov     AX,DGROUP
        mov     DS,AX
; Установить текстовый режим и очистить экран
        mov     AX,3
        int     10h
; Скрыть курсор – убрать за нижнюю границу экрана
        mov     [ScreenString],25
        mov     [ScreenColumn],0
        call    SetCursorPosition
; Вывести на экран сообщение "Вставьте дискету"
        mov     SI,offset Txt0
        call    ShowColorString
        call    ShowColorString
        call    ShowColorString
        ; Ожидать нажатия клавиши
        call    GetChar
; Прочитать BOOT-сектор дискеты
        mov     [Cylinder],0
        mov     [Head],0
        mov     [Sector],1
        call    ReadFDDSector
; Определить и запомнить параметры дискеты
        mov     AL,[SectorDataBuffer+18h]
        mov     [MaxSector],AL
        mov     AL,[SectorDataBuffer+1Ah]
        dec     AL
        mov     [MaxHead],AL

; Очистить экран
        call    ClearScreen
; Вывести текстовые сообщения
        mov     SI,offset Txt1
        call    ShowColorString
        call    ShowColorString
        call    ShowColorString
        call    ShowColorString
; Установить зеленый цвет и черный фон
        mov     [TextColorAndBackground],LIGHTGREEN
; Отобразить параметры установленной дискеты
        mov     [ScreenString],2
        mov     [ScreenColumn],25
        mov     AL,[MaxHead]
        inc     AL
        call    ShowDecByte
        mov     [ScreenColumn],43
        mov     AL,[MaxSector]
        call    ShowDecByte
```

```
; Вывести описание управляющих клавиш
        mov     SI,offset Txt2
        call    ShowString
        call    ShowString
        call    ShowString

; ЦИКЛ ПО СЕКТОРАМ
        mov     AX,0B800h
        mov     ES,AX
@@ReadSector:
        mov     SI,offset Txt3
        call    ShowColorString
        mov     [ScreenString],5
        mov     [ScreenColumn],18
        mov     AL,[Cylinder]
        call    ShowDecByte
        mov     [ScreenColumn],31
        mov     AL,[Head]
        call    ShowDecByte
        mov     [ScreenColumn],44
        mov     AL,[Sector]
        call    ShowDecByte
; Прочитать сектор
        call    ReadFDDSector
; Отобразить на экран содержимое считанного сектора
; в ASCII-кодах
@@ShowSector:
        ; Установить начало окна отображения сектора
        mov     DI,7*160+8*2
        mov     SI,offset SectorDataBuffer
        ; Задать для символов светло-голубой
        ; цвет и синий фон
        mov     AH,LIGHTCYAN+BLUE*16
        mov     DX,8  ;счетчик строк
@@OutNextString:
        mov     CX,64 ;счетчик символов в строке
@@OutNextChar:
        lodsb
        stosw
        loop    @@OutNextChar
        add     DI,16*2
        dec     DX
        jnz     @@OutNextString

@@GetCommand:
        ; Ожидаем ввода следующей команды
        call    GetChar
        cmp     AL,0
        je      @@TestCommandByte
        call    Beep
        jmp short @@GetCommand

@@TestCommandByte:
        cmp     AH,B_Esc      ;команда "Выход"?
; Выполнить команду "Выход"
        je      @@End
@@TestDn:
        cmp     AH,B_DN
        jne     @@TestUp
; Выполнить команду "Стрелка вниз"
        mov     AL,[Sector]
        cmp     AL,[MaxSector]
        jae     @@IncHead
        ; Увеличить на 1 номер сектора
        inc     [Sector]
        jmp     @@ReadSector
@@IncHead:
        mov     AL,[Head]
        cmp     AL,[MaxHead]
        jae     @@IncCylinder
        ; Увеличить на 1 номер головки,
        ; установить сектор 1
        inc     [Head]
        mov     [Sector],1
        jmp     @@ReadSector
@@IncCylinder:
        mov     AL,[Cylinder]
        cmp     AL,[MaxCylinder]
        jae     @@CommandError
        ; Увеличить на 1 номер цилиндра, установить
```

```
        ; головку 0, сектор 1
        inc     [Cylinder]
        mov     [Head],0
        mov     [Sector],1
        jmp     @@ReadSector
@@TestUp:
        cmp     AH,B_UP
        jne     @@CommandError
; Выполнить команду "Стрелка вверх"
        cmp     [Sector],1
        jbe     @@DecHead
        ; Уменьшить на 1 номер сектора
        dec     [Sector]
        jmp     @@ReadSector
@@DecHead:
        mov     AL,[Head]
        cmp     AL,0
        je      @@DecCylinder
        ; Уменьшить на 1 номер головки, установить
        ; максимальный номер сектора
        dec     [Head]
        mov     AL,[MaxSector]
        mov     [Sector],AL
        jmp     @@ReadSector
@@DecCylinder:
        mov     AL,[Cylinder]
        cmp     AL,0
        je      @@CommandError
        ; Уменьшить на 1 номер цилиндра, установить
        ; максимальные номера головки и сектора
        dec     [Cylinder]
        mov     AL,[MaxHead]
        mov     [Head],AL
        mov     AL,[MaxSector]
        mov     [Sector],AL
        jmp     @@ReadSector
@@CommandError:
        call    Beep
        jmp     @@GetCommand

@@End:  ; Переустановить текстовый режим
        mov     ax,3
        int     10h
        ; Выход в DOS
        mov     AH,4Ch
        int     21h
ENDP ShowFDDSector

;*********************************************
;*       ПРОЧИТАТЬ СЕКТОР ГИБКОГО ДИСКА      *
;* Параметры передаются через глобальные     *
;* переменные:                               *
;* Cylinder – номер цилиндра (от 0 до 79);   *
;* Head – номер головки (0 или 1);           *
;* Sector – номер сектора (от 1 до 18).      *
;* Считанный сектор записывается в основной  *
;* сегмент данных по адресу SectorDataBuffer. *
;*********************************************
PROC ReadFDDSector NEAR
        pusha
        push    ES
        mov     AX,DS

        mov     ES,AX
        mov     CX,3    ;счетчик повторений
@@Repeat:
        mov     BX,offset SectorDataBuffer
        mov     AH,2    ;подфункция "Прочесть сектор"
        mov     AL,1    ;прочесть 1 сектор
        mov     CH,[Cylinder]
        mov     CL,[Sector]
        mov     DH,[Head]
        mov     DL,0    ;читать диск "A:"
        int     13h
        jnc     @@End
        ; Ошибка считывания, инициализировать систему
        mov     AH,0    ;подфункция "Инициализировать"
        mov     DL,0    ;диск "A:"
        int     13h
        loop    @@Repeat
        ; Фатальная ошибка
```

```
        jmp     @@Err
; Нормальное завершение процедуры
@@End:  pop     ES
        popa
        ret
; Аварийный выход – ошибка при чтении файла
@@Err:  ; Очистить экран
        call    ClearScreen
        ; Выдать сообщение об ошибке
        mov     SI,offset TErr
        call    ShowColorString
        call    ShowColorString
        call    GetChar
        ; Выход в DOS
        mov     AH,4Ch
        int     21h
ENDP ReadFDDSector
ENDS

; Подключить процедуры вывода данных на экран
include "lst_1_02.inc"
; Подключить процедуры перевода чисел в десятичный код
include "lst_2_05.inc"

END
```

ПРИМЕЧАНИЕ

Приведенный пример универсален и его можно запускать на любом персональном компьютере - даже на древних машинах типа XT с пятидюймовыми дисководами.

Контрольные вопросы.

1. Какие дискеты и их форматы используются в PC?
2. Опишите стандартный формат дорожки дискеты.
3. Зачем нужны промежутки между заголовком и полем данных сектора, а также между секторами?
4. Для чего используется поле SYNC?
5. Что представляет собой аппаратный интерфейс накопителей на гибких магнитных дисках?
6. Опишите логику работы НГМД и его взаимодействие с контроллером.
7. Опишите в общих чертах контроллер НГМД (FDC) PC/AT.
8. Какие основные блоки можно выделить в структурной схеме БИС контроллера НГМД i8272.
9. Опишите регистровую архитектуру БИС контроллера НГМД i8272.
10. Каковы особенности подключения i8272 к НГМД?
11. Охарактеризуйте схему подключения контроллера i8272 к микропроцессорной системе.
12. Перечислите регистры контроллера НГМД PS/2 И PC/AT, и дайте их характеристику.
13. Охарактеризуйте регистры состояния S0 - S3.
14. Какие особенности имеет система команд контроллера НГМД?
15. Какие команды реализуют обмен информацией с накопителями на ГМД?
16. Каково назначение команд "Форматировать" и Определить"?
17. Какова особенность режимов работы с DMA и без DMA?
18. В чем заключаются особенности работы с регистрами данных?
19. Как выполняется начальный программный сброс?
20. Как осуществляется работа с дисками на низком уровне через BIOS?
21. Как вызываются функции прерывания INT 13h ?
22. Как можно получить таблицы параметров гибких и жестких дисков?

В данной главе использованы материалы из [1, 4, 9]. Более подробное описание БИС КНГМД i8272 (КР580ВТ72) можно найти в [9]. Особенности работы с дисками через BIOS и прерывание INT21h DOS достаточно подробно изложены в[4] и разделе 3. Там же приведены примеры программ, иллюстрирующих использование BIOS (INT 13h) и INT 21h DOS. Пример программы, иллюстрирующий работу с гибкими дисками на уровне команд контроллера НГМД, приведен в [4].

5 Память на жестких дисках

В отличие от накопителей на гибких дисках и их контроллеров, жестко стандартизованных и поэтому легко конфигурируемых, в PC применяется множество типов накопителей на жестких дисках, их интерфейсов и контроллеров, различающихся и способами конфигурирования.

Накопители на жестких магнитных дисках (НЖМД), они же *HDD* (Hard Disk Drive), появились с машинами PC/XT. Первые накопители имели интерфейс, являющийся, по сути, расширением интерфейса НГМД, и подключались к специальной плате контроллера с модулем дополнительной BIOS, распространяющим функции дискового сервиса на жесткий диск. Вся информация об установленных жестких дисках хранилась на плате контроллера. В машинах класса AT поддержку стандартного контроллера включили в системную BIOS, а параметры используемых жестких дисков стали хранить в памяти CMOS. Дальнейшее описание относится именно к дискам машин AT.

Традиционные версии BIOS поддерживают *до двух накопителей* на жестких дисках и хранят их параметры в ячейках памяти CMOS. *Расширенные версии BIOS* для современных двухканальных контроллеров ATA поддерживают *4 жестких диска* и хранят их параметры. Для каждого диска хранится номер его типа (1-46), соответствие номеров типов геометрическим параметрам (число цилиндров, головок и секторов) стандартно для большинства версий BIOS, хотя бывают и исключения. Тип 47 (USER TYPE) позволяет задать произвольные параметры, он используется и при *автоматическом определении параметров дисков IDE* опцией IDE Autodetection утилиты BIOS Setup. Для дисков с различными интерфейсами тип задается по-разному.

Для дисков с интерфейсом *ST-506/412* обычно задавали соответствующий номер типа с фиксированными параметрами, практически все существовавшие накопители укладывались в стандартную таблицу. При желании те же параметры можно было ввести и через тип 47.

Для дисков *ESDI* или *SCSI* в поле типа указывается *NONE,* и их контроллер, обязательно имеющий расширение BIOS, сам определяет параметры установленных накопителей.

Для дисков *IDE* (в опциях BIOS Setup обычно применяют это название вместо официального ATA) задаются логические параметры геометрии, которые транслируются в физические внутренним контроллером диска. Параметры трансляции могут изменяться программно, функция *IDE Autodetect* (BIOS Setup) определяет *логические* параметры по умолчанию. Один и тот же накопитель может иметь разную логическую (внешнюю) геометрию. По этой причине диск, отформатированный с фиксированным типом (1-46), по функции IDE Autodetect может получить иное сочетание параметров. С такой ситуацией часто сталкиваются при замене системной платы: на старой плате тип диска в CMOS посмотреть забудут, заменят системную плату, автоматически определят тип диска, а операционная система загружаться откажется (сообщение «Missing Operation System»). При этом, загрузив ОС с дискеты, может удастся просмотреть корневой каталог диска, но это не означает корректности доступа ко всем данным. В данной ситуации, если нет желания переформатировать диск (естественно, с потерей данных), следует найти и использовать номер ранее используемого фиксированного типа. Его можно и подобрать из сравнительно небольшого числа типов, у которых емкость близка к автоматически определенному значению. Критерием правильности подбора будет успех загрузки ОС.

Современные версии PnP-BIOS позволяют не указывать тип (параметры) диска IDE, а выбрать опцию *Auto* для автоматической установки типа во время POST по ответу на команду идентификации. Конечно, для старых дисков IDE, не поддерживающих эту команду, такая установка не сработает.

Для дисков IDE современные версии BIOS позволяют выбрать и режим адресации — CHS, ECHS (Large Disk) или LBA, различия и ограничения которых были рассмотрены выше.

Накопители на жестких дисках, подключаемые к таким внешним интерфейсам, как LPT-порт, шины USB и Fire Wire, функциями ROM BIOS пока не поддерживаются и конфигурируются уже на этапе загрузки операционной системы. Распространение последовательных шин как интерфейсов устройств внешней памяти, возможно, изменит эту ситуацию, тем более что эти шины «дружны» с системой PnP.

Для накопителей на жестких дисках используют интерфейсы ST-506/412, ESDI, ATA, SCSI. Накопители и контроллеры с интерфейсами ST-506/412 и ESDI практически сошли со сцены, поскольку эти устройства не выпускаются уже много лет, а ранее выпущенные уже выработали отпущенный им жизненный ресурс. По этой причине ограничимся лишь кратким их описанием, а подробнее рассмотрим устройства с современными интерфейсами ATA и SCSI. Возможно также подключение дисковых устройств и к параллельному порту, но через устройство, обеспечивающее один из вышеперечисленных интерфейсов. О дисках с интерфейсом USB говорить пока рано, а интерфейс Fire Wire является родственником SCSI-3.

5.1. Накопители с интерфейсами ST-506 (ST-412) и ESDI

Первые накопители на жестких дисках имели интерфейс, напоминающий интерфейс НГМД. Это и понятно, поскольку НЖМД отличается только большим числом рабочих поверхностей (головок записи-считывания), более высокой скоростью вращения, а, следовательно, и передачи данных, и несменяемостью носителя. Этот интерфейс по названию первых моделей контроллеров именуется *ST-*

506 (ST-412). Поскольку первые контроллеры работали по схеме кодирования MFM, этот интерфейс называют также и «интерфейсом дисков MFM». Такое название не совсем корректно, поскольку, по сути, этот же интерфейс использовался и для накопителей, допускающих использование схемы кодирования RLL. Некоторые модели накопителей выпускались в версиях MFM и RLL и различались только качеством носителя — у схемы RLL требования выше.

Интерфейс использует два плоских кабеля. *Магистральный* 34-проводной *кабель управления* позволяет подключать до двух накопителей, адрес накопителя определяется его положением на шлейфе. Все сигналы в данном кабеле имеют уровни ТТЛ, активный уровень — низкий. Система выборки напоминает применяемую в НГМД, но в кабеле перевернут фрагмент из 5 проводов 25-29. Устройства должны отзываться на сигнал DS0. К накопителю C: должен подходить прямой кабель, к D: — с перевернутым фрагментом. Каждый накопитель с контроллером соединяется и 20-проводным *кабелем данных,* по которым передаются аналоговые сигналы усилителей головок записи-чтения. На накопителях применяются разъемы с печатными ламелями, на контроллере — со штырьковыми контактами.

Передача аналоговых сигналов записи-чтения по длинным интерфейсным кабелям не позволяет достигать высокой скорости передачи данных и, следовательно, высокой плотности хранения информации. В режиме MFM диски имеют 17 секторов на трек, в режиме RLL — 26. Поскольку низкоуровневые форматы различных моделей контроллеров могут не совпадать, нет гарантии того, что данные накопителя, отформатированного на контроллере одной модели, окажутся доступными для контроллера другой модели (даже если оба контроллера работают по одной схеме кодирования — MFM или RLL). Замена контроллера в большинстве случаев требует низкоуровневого форматирования дисков, причем для производительности критичен фактор чередования секторов (интерливинг).

Интерфейс ESDI (Enhanced Small Device Interface — расширенный интерфейс малых устройств) появился как развитие ST-506. Здесь существенная часть узлов контроллера перенесена на дисковод для повышения производительности (XFER до 1 Мбайт/с) и плотности записи (до 32-80 SPT). Накопители ESDI хранят описатели своих геометрических параметров и список дефектных блоков на самом диске, отсюда следует и установка «None» вместо типа диска в параметрах CMOS-конфигурации. Непонятно, ради какой унификации (скорее — путаницы) интерфейс сделали конструктивно совпадающим с ST-506. Назначение сигналов этих интерфейсов различно, и, естественно, взаимной совместимости устройств с интерфейсами ESDI и ST-506 быть не может.

Конфигурирование устройств с интерфейсами ST-506/412 и ESDI сводится к заданию адреса и установке терминатора. Если на устройстве имеются джамперы, обозначенные как DS0/DS1, следует установить джампер DS0. Если они обозначены как DS1/DS2, следует установить джампер DS1. Иногда используют управляющий кабель без перевернутого фрагмента, тогда в случае установки двух накопителей на устройстве C: устанавливают DS0, а на устройстве D: — DS1. Управляющий кабель должен иметь терминатор на последнем устройстве шлейфа. Терминаторы обычно представляют собой плоскую резисторную сборку со штырьковыми контактами, которая по умолчанию устанавливается в гнезда всех накопителей. При установке двух накопителей на среднем устройстве шлейфа терминатор рекомендуется снимать.

Устройства с интерфейсами ST-506/412, ESDI имеют *внешний контроллер,* который обычно представляет собой плату, устанавливаемую в слот системной шины ввода/вывода. Контроллеры жесткого диска *HDC* (Hard Disk Controller) для компьютеров XT и AT различны как по регистровым моделям (табл. 5.1, 5.2), так и по занимаемым системным ресурсам.

HDC XT использует порты ввода/вывода 320h - 327h, аппаратное прерывание IRQ5 и канал DMA3. Контроллер для XT всегда имеет свой модуль расширения BIOS (начальный адрес C8000), поскольку в системной BIOS XT драйверы жесткого диска отсутствуют.

В компьютерах AT предусмотрена возможность установки двух контроллеров. HDC#1 занимает порты 1F0h - lF7h, *HDC#2* — 170h - 177h. Для первого контроллера предназначена линия IRQ14. Канал DMA для дисков в AT обычно не используется, поскольку его производительность уступает программному вводу-выводу PIO. На плате HDC AT расширение BIOS может и отсутствовать, если контроллер программно совместим со стандартным, драйвер которого имеется в системной BIOS AT.

В AT может работать и контроллер HDC XT, поскольку его драйвер из дополнительной BIOS перекроет стандартный драйвер AT. Проблемы могут возникнуть только при форматировании.

Дополнительный модуль BIOS HDC обычно содержит программу низкоуровневого форматирования, которую можно запустить, например, с помощью отладчика DEBUG командой G=C800:5.

Для контроллеров AT допустимы следующие *команды* (x задает параметры):

- 1xh — возврат к цилиндру 00.
- 2xh — чтение сектора.
- 3xh — запись сектора.
- 4xh — чтение с верификацией.
- 7xh — поиск цилиндра.

- 50h — форматирование трека.
- 90h — диагностика.

Таблица 5.1. Регистры контроллера HDC AT

Адрес	Назначение (R — чтение, W — запись)
1F0	RW: Регистр данных
1F1	W: Включение предкомпенсации записи
1F1	R: Регистр ошибок: Бит 7 — Bad Block — признак дефектного блока. Бит 6 — ECC Error — неисправимая ошибка данных. Бит 5 = 0. Бит 4 — Sec. ID not found — сектор не найден. Бит 3 = 0. Бит 2 — Command Aborted — команда отвергнута. Бит 1 — Track 0 Err. — ошибка поиска 0-трека. Бит 0 — Data Address Mark not found — адресный маркер данных не найден
1F2	RW: Счетчик секторов для операций обмена
1F3	RW: Номер начального сектора
1F4	RW: Младший байт номера цилиндра
1F5	RW: Старший байт номера цилиндра
1F6	RW: Номер диска и головки
1F7	W: Регистр команд
1F7	R: Регистр состояния: Бит 7: 1 = занято, выполняется команда. Бит 6: l = готов к чтению/записи/поиску. Бит 5: 1 = ошибка записи. Бит 4: 1 = поиск завершен. Бит 3: 1 = запрос данных. Бит 2: 1 = исправлена ошибка данных. Бит 1: 1 = датчик индексного отверстия. Бит 0: 1 = ошибка завершения предыдущей команды

Таблица 5.2. Регистры контроллера HDC XT

Адрес	Назначение (R — чтение, W — запись)
320h	RW: Регистр данных, команд и состояния. За байтом команды следует цепочка байт параметров
321h	W: Сброс контроллера (байтом 00h)
321h	R: Регистр состояния контроллера: Бит 5 — номер привода. Бит 1 — 1 = обнаружена ошибка
322h	W: Выбор контроллера, используется перед посылкой команды
323h	W: Регистр масок DMA и IRQ: Бит 1: 1 = разрешение прерываний (IRQ5). Бит 0: 1 = разрешение использования DMA3.

Контрольные вопросы.

1. Сколько жестких дисков поддерживает традиционная версия BIOS и расширения для дисков ATX (IDE)?
2. Как определяются и задаются параметры жестких дисков?
3. Какие жесткие диски не поддерживаются системной BIOS?
4. Дайте краткое описание аппаратной части интерфейса ST508/412.
5. Сколько секторов на дорожку имели диски MFM, RLL и ESDI?
6. В чем сходство и различие контроллеров дисков ST506/412 для PC XT и PC AY?
7. Охарактеризуйте регистры контроллеров HDC XT и HDC AT.

В этом разделе использованы материалы из [2, 3]. Более подробные сведения о конструктивных

особенностях современных жестких дисков, о их производительности, оптимизации, параметрах и принципах функционирования можно найти в [3, 6, 7].

5.2. Накопители с интерфейсом ATA (IDE)

Интерфейс ATA появился в результате воплощения идеи переноса контроллера диска непосредственно к накопителю. Так появился класс устройств *IDE* (Integrated Device Electronics) — *устройств со встроенным контроллером,* имеющих ряд преимуществ перед устройствами с отдельным контроллером:

- За счет минимального удаления контроллера от диска удается существенно повысить быстродействие, поскольку отпадает необходимость передавать высокочастотные сигналы записи и чтения по длинным интерфейсным проводам.
- Снимается проблема совместимости накопителей и контроллеров по физическим форматам записи. Обмен с устройствами IDE происходит информационными и управляющими байтами или словами, а не закодированными последовательностями импульсных сигналов.
- Появляется больший простор для внутренних усовершенствований устройств, направленных на повышение производительности, надежности, плотности хранения информации и другие цели. Эти усовершенствования отрабатываются встроенным контроллером и могут им не выноситься на уровень внешнего интерфейса.
- Упрощается схемотехника адаптера подключения устройств к шине компьютера. Сигналы интерфейса представляют собой сокращенный набор буферизованных сигналов системной шины, а обязательная встроенная буферная память устройства позволяет не привязывать скорость обмена по внешнему интерфейсу к скорости обмена данными с собственно носителем информации. Таким образом, устройства IDE можно подключать через соответствующие адаптеры как к высокопроизводительной системной шине, так и к медленному интерфейсу стандартного параллельного порта. Конечно, производительность обмена будет существенно различной, но принципиальная возможность сопряжения есть.

Для стандартной BIOS IDE-устройство эмулирует один контроллер типа ST-506 с подключенным накопителем. Увеличение количества накопителей до двух (что предусматривает стандартный контроллер ST-506) создает некоторые сложности, поскольку реально каждое устройство IDE входит в систему со своим контроллером. Однако некоторыми ухищрениями эти сложности преодолели, хотя и не без потерь.

5.2.1. Интерфейс ATA

Интерфейс ATA— AT Attachment for Disk Drives — разрабатывался в 1986 - 1990 гг. для подключения накопителей на жестких магнитных дисках к компьютерам IBM PC AT с шиной ISA. Стандарт, выработанный комитетом X3T10, определяет набор регистров и назначение сигналов 40-контактного интерфейсного разъема. Интерфейс появился в результате переноса контроллера жесткого диска ближе к накопителю, то есть создания устройств со встроенным контроллером — IDE (Integrated Drive Electronics). Стандартный для AT контроллер жесткого диска был перенесен на плату электроники накопителя с сохранением регистровой модели. При этом удлинилась связь с устройством со стороны системной шины, выводить которую непосредственно на длинный ленточный кабель было нецелесообразно. Это сказалось бы на скорости работы шины, надежности и цене. Из всех сигналов шины ISA выбрали минимальный набор сигналов, часть из которых буферизовали на небольшой плате, устанавливаемой в слот, а часть направили прямо на разъем ленточного кабеля нового интерфейса. Из сигналов системной шины потребовались следующие:

- шина данных — используется полностью;
- шина адреса — 3 младших бита поступают в интерфейс, старшие биты и сигнал AEN проходят через дешифратор, вырабатывающий сигналы выбора устройства CS0# и CS1#;
- шина управления — используются сигналы чтения и записи портов и аппаратного сброса, запрос прерывания, пара сигналов канала DMA, сигналы готовности и управления разрядностью передачи.

Поскольку стандартный контроллер AT позволял подключать до двух накопителей, эту возможность получил и интерфейс ATA. Однако теперь два накопителя стали означать и два контроллера. Их подключили к одной интерфейсной шине, а для программной совместимости бит выбора накопителя *(DEV)* в регистре номера головки и номера устройства (drive/head register) стали использовать для выбора устройства. Для взаимодействия пары устройств на шине ввели несколько дополнительных сигналов. Так появился интерфейс ATA для подключения устройств IDE к шине ISA. Позже их стали подключать и к локальным шинам, но набор сигналов интерфейса и протоколы обмена сохранились. Достаточно универсальный набор сигналов позволяет подключать любое устройство со встроенным контроллером, которому в пространстве портов ввода/вывода достаточно того же набора регистров, способное поддержать режим выбора устройства через вышеупомянутый бит. Принятая система команд и регистров, являющаяся частью спецификации ATA, ориентирована на блочный обмен

данными с устройствами прямого доступа. Для иных устройств существует спецификация ATAPI, основанная на тех же аппаратных средствах, но позволяющая обмениваться пакетами управляющей информации (PI — Package Interface). Пакетный интерфейс позволяет расширить границы применения шины ATA.

Адресация в ATA тоже имеет «дисковые корни»: для накопителей изначально указывали адрес цилиндра (cylinder), головки (head) и сектора (sector) — так называемая *трехмерная адресация CHS*. Сначала эта адресация точно соответствовала реальной геометрии — физически сектор действительно находился по указанному адресу. Позже по ряду причин диски ATA стали описывать внешней геометрией, отличающейся от реальной внутренней (реально, например, разные зоны треков имеют разное число секторов, причем часть секторов может резервироваться на случай замены дефектных). При этом одно и то же устройство может иметь различную внешнюю геометрию. Преобразование адресов в реальные выполняется встроенным контроллером устройства. Применительно к устройству ATA внешнюю геометрию (по которой адресуются к секторам устройства) часто называют логической, а реальную для накопителя — физической. Однако если рассматривать устройство ATA в окружении компьютера и BIOS, то *логической геометрией* принято называть ту, которая используется программами при вызове функций BIOS *INT13h*, а *физической геометрией* — ту, которая передается устройству в его командные регистры по шине ATA. С *реальной геометрией* (внутренней) устройства имеют дело лишь низкоуровневые утилиты тестирования и форматирования дисков. Эта терминология используется в данной книге повсеместно (чтобы не было путаницы), причем если упоминается CHS-адрес или геометрия без каких-либо прилагательных, подразумевается *внешняя* геометрия устройства ATA.

Позже пришли *к линейной адресации логических блоков LBA* (Logical Block Addressing), где адрес блока (сектора) определяется 28-битным числом.

В спецификации ATA фигурируют следующие компоненты:

- *Хост-адаптер* — средства сопряжения интерфейса ATA с системной шиной (набор буферных схем между шинами ISA и ATA). Хостом мы будем называть компьютер с хост-адаптером интерфейса ATA. *Хост-контроллер* — более развитый вариант хост-адаптера.
- Ленточный кабель (шлейф) с двумя или тремя 40-контактными IDC-разъемами. В стандартном кабеле одноименные контакты всех разъемов соединяются вместе.
- Ведущее устройство (Master) — ПУ, в спецификации ATA официально называемое Device-0 (устройство 0).
- Ведомое устройство (Slave) — ПУ, в спецификации официально называемое Device-1 (устройство 1).

Если к шине ATA подключено одно устройство, оно должно быть ведущим. Если подключены два устройства, одно должно быть ведущим, другое — ведомым. О своей роли (ведущее или ведомое) устройства «узнают» с помощью предварительно установленных конфигурационных джамперов. Если применяется «кабельная выборка» (см. ниже), роль устройства определяется его положением на специальном ленточном кабеле.

Оба устройства воспринимают команды от хост-адаптера одновременно. Однако исполнять команду будет лишь выбранное устройство. Если бит *DEV=0,* выбрано ведущее устройство, если *DEV=1* — ведомое. Выводить выходные сигналы на шину ATA имеет право только выбранное устройство. Такая система подразумевает, что начав операцию обмена с одним устройств, хост-адаптер не может переключиться на обслуживание другого до завершения начатой операции. Параллельно могут работать только устройства IDE, подключаемые к разным шинам (каналам) ATA. Спецификация ATA-4 определяет способ обхода этого ограничения.

Выполняемая операция и направление обмена данными между устройством и хост-адаптером определяется предварительно записанной командой. Непременным компонентом устройства является буферная память. Ее наличие позволяет выполнять обмен данными в темпе, предлагаемом хост-адаптером (в пределах возможности устройства), без оглядки на внутреннюю скорость передачи данных между носителем и буферной памятью ПУ.

Для устройств IDE существует несколько разновидностей интерфейса:

- ATA, он же AT-BUS — 16-битный интерфейс подключения к шине компьютера AT. Наиболее распространенный 40-проводный сигнальный и 4-проводный питающий интерфейс для подключения дисковых накопителей к компьютерам AT. Для миниатюрных (2,5" и менее) накопителей используют 44-проводный кабель, по которому передается и питание.
- PC Card ATA — 16-битный интерфейс с 68-контактным разъемом PC Card (PCMCIA) для подключения к блокнотным PC.
- XT IDE (8 бит), он же XT-BUS — 40-проводный интерфейс, похожий на ATA, но несовместимый с ним.
- MCA IDE (16 бит) — 72-проводный интерфейс, предназначенный специально для шины и накопителей PS/2.

- ATA-2 — расширенная спецификация ATA. Включает 2 канала, 4 устройства, PIO Mode 3, Multiword DMA Mode 1, Block mode, объем диска до 8 Гбайт, поддержка LBA и CHS.
- Fast ATA-2 разрешает использовать Multiword DMA Mode 2 (13,3 Мбайт/с), PIO Mode 4.
- ATA-3 — расширение ATA-2. Включает средства парольной защиты, улучшенного управления питанием, самотестирования с предупреждением приближения отказа — SMART (Self Monitoring Analysis and Report Technology).
- ATA/ATAPI-4 — расширение ATA-3, включающее режим Ultra DMA со скоростью обмена до 33 Мбайт/с и пакетный интерфейс ATAPI. Появляется поддержка очередей и возможность перекрытия команд.
- ATA/ATAPI-5 — ревизия ATA/ATAPI-4: удаляются устаревшие команды и биты, добавляются новые возможности защиты и управления энергопотреблением. Включает режим Ultra DMA со скоростью обмена до 66 Мбайт/с.
- ATA/ATAPI-6 — дополнения к ATA/ATAPI-5: потоковое расширение для чтения/записи аудио- и видеоданных, управление акустическим шумом, режим Ultra DMA со скоростью обмена до 100 Мбайт/с.
- E-IDE (Enhanced IDE) — расширенный интерфейс, введенный фирмой Western Digital. Реализуется в адаптерах для шин PCI и VLB. Позволяет подключать до 4 устройств (к двум каналам), включая CD-ROM и стримеры (ATAPI). Поддерживает PIO Mode 3, Multiword DMA Mode 1, объем диска до 8 Гбайт, LBA и CHS. С аппаратной точки зрения практически полностью соответствует спецификации ATA-2.

Устройства ATA IDE, E-IDE, ATA-2, Fast ATA-2, ATA-3, ATA/ATAPI-4, ATA/ ATAPI-5 и ATA/ATAPI-6 электрически совместимы. Степень логической совместимости достаточно высока (все базовые возможности ATA доступны). Однако для полного использования всех расширений необходимо соответствие спецификаций устройств, хост-адаптера и его ПО.

Разработкой спецификаций ATA/ATAPI занимается технический комитет T13 американского Национального Комитета по стандартизации в области информационных технологий (NCITS). Разработанные им спецификации оформляются в виде стандартов ANSI. В 1998 году был принят стандарт ATA/ATAPI-4 (ANSI NCITS 317-1998 AT Attachment - 4 with Packet Interface Extension), в 2000 году завершена работа над ATA/ATAPI-5 и ведется проработка ATA/ATAPI-6. Предполагается, что это будет последней версией параллельного интерфейса ATA, а в 2001-2002 году появится последовательный интерфейс *Serial ATA*. Про него пока что известно, что с программной точки зрения он будет совместим с прежними, а электрический интерфейс изменится в корне. Цель перехода на последовательный интерфейс — улучшение и удешевление кабелей и коннекторов, улучшение условий охлаждения (избавление от широкого шлейфа), обеспечение возможности разработки компактных устройств, облегчение конфигурирования устройств пользователем. Последовательный интерфейс ATA, как и его параллельный предшественник, предназначен для подключений устройств внутри компьютера. Для внешних устройств предназначаются шины SCSI, USB и Fire Wire.

5.2.1.1. Электрический интерфейс

Все сигналы ATA являются логическими со стандартными ТТЛ-уровнями:

- высокий уровень *выходного* сигнала не ниже 2,4 В (при втекающем токе до 400 мкА, сигнал DMARQ — до 500 мкА), низкий уровень не выше 0,5 В (при токе 4 мА, для линии DASP ради совместимости со старыми устройствами — 12 мА);
- высокий уровень *входного* сигнала не ниже 2,0 В, низкий уровень не выше 0,8 В.

Все информационные сигналы интерфейса передаются через 40-контактный разъем, у которого ключом является отсутствующий на вилке и закрытый на розетке контакт 20. Использование в качестве ключа выступа на корпусе розетки и прорези в бандаже вилки стандартом не приветствуется. Для соединения устройств спецификация требует применения плоского многожильного кабеля (Flat cable stranded 28 AWG) типа 3M 3365-40 с шагом проводов 1,27 мм или эквивалентного. Возможно применение экранированного кабеля (Flat cable stranded 28 AWG 3M 3517-40 shielded), но он практически не встречается. Длина кабеля не должна превышать 0,46 М (18"), допустимая емкость проводников не более 35 пФ. Терминаторы стандартом не предусматриваются (они имеются в каждом устройстве и хост-адаптере), но если кабель с тремя разъемами (розетками) используют для подключения одного устройства, то устройство и хост-адаптер рекомендуется подключать к противоположным концам кабеля. Состав информационных сигналов интерфейса ATA приведен в табл. 5.3. Вид кабеля приведен на рис. 5.1. В большинстве кабелей одноименные контакты всех разъемов соединяются своими проводами и все коннекторы равноправны. Встречаются (редко) ленточные кабели с кабельной выборкой, изображенные на рис. 5.2. В них провод 28 перерезан, так что контакт 28 (CSEL)для ведущего устройства заземлен через хост-адаптер, а для ведомого — не подключен. Кабель должен

соответствовать системе адресации, выбранной для обоих устройств.

Начиная с ATA/ATAPI-4 в шлейфах узаконили кабельную выборку и для подключения устройства-1 определили средний коннектор. В ней контакт 28 либо не соединен с проводом, либо просто отсутствует. Вполне понятно, что при использовании кабельной выборки хост-контроллер подключать к среднему коннектору нельзя (как и к правому на рис. 5.2). Если номер устройства назначается джампером, то для 40-проводного кабеля можно подключать устройства и хост-контроллер к любым коннекторам произвольно (но желательно избегать «висячих» концов).

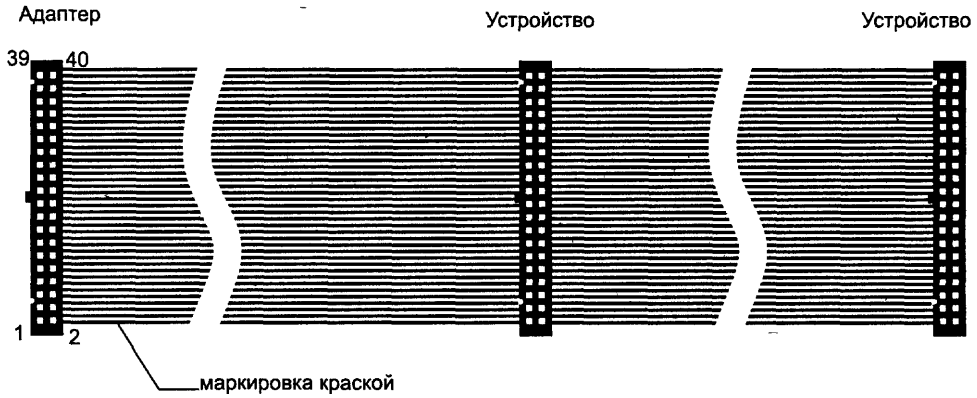

Рис. 5.1. Интерфейсный кабель ATA

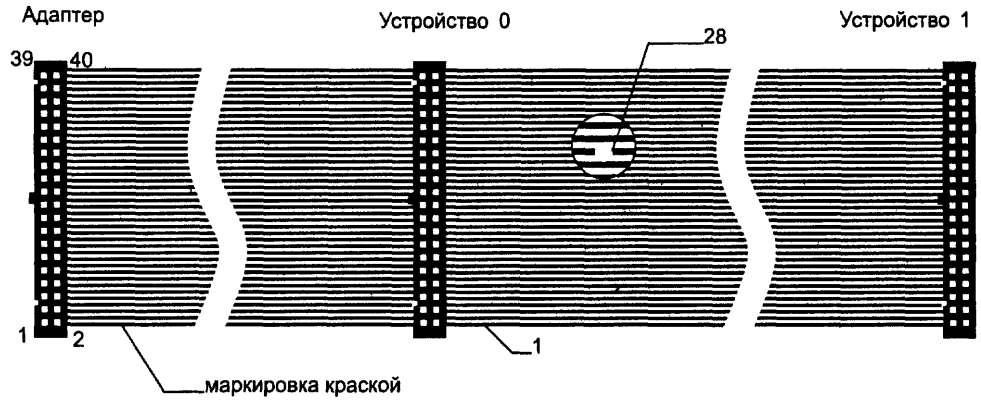

Рис. 5.2. Ленточный кабель интерфейса ATA с кабельной выборкой

Для устойчивой работы в режиме Ultra DMA рекомендуется применение *80-проводных кабелей,* обеспечивающих чередование сигнальных цепей и проводов схемной земли (GND). Такие кабели требуются для режимов Ultra DMA выше 2 (скорость выше 33 Мбайт/с). Эти кабели разделываются на специальные разъемы, имеющие 40-контактные гнезда с обычным назначением контактов, но ножевые контакты для врезки 80 проводов. В шлейфе для схемной земли используются либо все четные, либо все нечетные провода — это зависит от применяемых разъемов (на них должна быть маркировка «EVN GND» или «ODD GND» соответственно). Установка на одном шлейфе разнотипных разъемов недопустима (тогда все 80 проводов окажутся соединенными вместе). Спецификация предполагает ленточный кабель с шагом проводов 0,635 мм, импедансом 70-90 Ом и погонной емкостью 42-52 пФ/м. Допустимые размеры шлейфа те же (см. рис. 5.1). На 80-проводном кабеле в разъеме для подключения контроллера контакт 34 соединен с шиной GND и не соединен с проводом шлейфа, этим обеспечивается идентификация типа кабеля (CBLID). Провод шлейфа соединяет контакты 34 разъемов устройств, что обеспечивает прохождение сигнала PDIAG# от ведомого устройства к ведущему. С учетом возможности кабельной выборки, на 80-проводном шлейфе положение коннекторов уже однозначно:

- Коннектор *хост-контроллера* расположен на конце шлейфа, у него контакт 34 заземлен и не соединен с шлейфом. Корпус коннектора должен быть *синего* цвета.
- Коннектор *устройства 0* расположен на противоположном конце шлейфа, у него все контакты соединены с шлейфом. Корпус коннектора должен быть *черного* цвета.
- Коннектор *устройства 1* (необязательный) расположен в середине шлейфа, у него контакт 28 не соединен с шлейфом. Корпус коннектора должен быть серого цвета.

Если кабельная выборка не используется, то устройства 0 и 1 можно менять местами.

Устройства и адаптеры, рассчитанные на использование Ultra DMA, должны иметь в сигнальных цепях последовательные согласующие резисторы с сопротивлением для различных цепей 22, 33 или 82

Ом.

Спецификация ATA «узаконивает» как 40-контактный интерфейсный разъем, так и 4-контактный разъем питания (рис. 5.3), но для малогабаритных устройств питание может подаваться по 44-проводному интерфейсному кабелю.

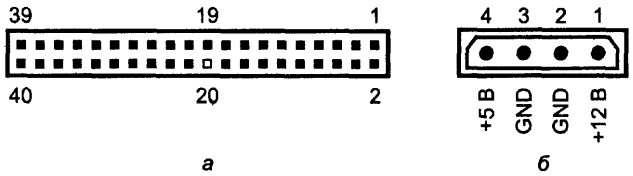

Рис. 5.3. Разъемы интерфейса ATA (вилки на устройствах) а — интерфейсный, *б*— питания

Таблица 5.3. Интерфейс ATA (IDE)

Сигнал	Тип[1]	Контакт	Контакт	Тип1	Сигнал
RESET#	I	1	2	-	GND
DD7	I/O TS	3	4	I/O TS	DD8
DD6	I/O TS	5	6	I/O TS	DD9
DD5	I/O TS	7	8	I/O TS	DD10
DD4	I/O TS	9	10	I/O TS	DD11
DD3	I/O TS	11	12	I/O TS	DD12
DD2	I/O TS	13	14	I/O TS	DD13
DD1	I/O TS	15	16	I/O TS	DD14
DD0	I/O TS	17	18	I/O TS	DD15
GND	-	19	20	-	Ключ (нет штырька)
DMARQ	O TS2	21	22	-	GND
DIOW# /STOP[3]	I	23	24	-	GND
DIOR# /HDMARDY# /HSTROBE#	I	25	26		GND
IORDY /DDMARDY# /DSTROBE[3]	O TS[2]	27	28	I/O	SPSYNC/CSEL[7]
DMACK#	I	29	30	-	GND
INTRQ	O TS[2]	31	32	O OK	IOCS16#[8]
DA4	I	33	34	I, O[4]	PDIAG#/CBLID[3]
DA0	I	35	36	I	DA2
CS0#	I	37	38	I	CS1#
DASP#	I/O OK[5]	39	40	-	GND
+5 В (Logic)	-	41[6]	42[6]	-	+5 В(Motor)
GND	-	43[6]	44[6]	-	Зарезервирован

[1] Тип сигнала для устройства: I — вход, O — выход, I/O — двунаправленный, TS — тристабильный, OK — открытый коллектор. Для хост-адаптера значения I и O имеют противоположный смысл.

[2] У старых устройств сигнал может иметь тип OK (при разнотипных сигналах на одной шине возможен конфликт).

[3] Сигналы, приведенные после символа /, используются только в режиме Ultra DMA (ATA-4).

[4] У ведущего устройства — вход, у ведомого — выход.

[5] У ведомого устройства — только выход.

[6] Контакты 41 - 44 используются только для миниатюрных дисков.

[7] Начиная с ATA-3 — только CSEL.

[8] Начиная с ATA-3 зарезервирован.

Для большинства устройств применяется 40-контактный разъем с шагом контактов 2,54 мм. Рядом с ним могут располагаться дополнительные контакты, служащие для конфигурирования устройства и технологических целей (диагностики и других операций по служебному последовательному интерфейсу). Спецификация ATA/ATAPI предусматривает два варианта, приведенные на рис. 5.4, *б* и *в*. Здесь пустым квадратиком обозначены позиции ключевых (пропущенных) выводов, контакты 1 - 40 используются для интерфейса (табл. 5.3), а контакты A - H — для установки джамперов (табл. 5.4) и технологических целей. Для миниатюрных устройств предназначен 50-контактный разъем с шагом

выводов 2 мм (рис. 5.4, а), у которого контакты 1-44 соответствуют табл. 5.3, контакты A-D используются для конфигурирования (табл. 5.4) и пара пропущенных является дополнительным ключом. Для 50-контактного кабеля принято назначение выводов накопителей IBM Thinkpad/Travelstar:

- Контакт «A» через резистор 10 кОм соединяется с шиной +5 В.
- Контакт «B» для устройства задает его роль: низкий уровень — устройство 0, высокий — устройство 1.
- Контакт «C» определяет режим устройства по включению питания: низкий уровень — Standby, высокий — Idle.
- Контакт «D» соединяется с контактом 28 (CSEL) и через резистор 10 кОм — с шиной +5 В.

Это назначение, позволяющее конфигурировать и адресацию, и режим потребления, применяется не на всех устройствах. На винчестерах фирмы Toshiba контакты A и B могут использоваться как выход и вход последовательного интерфейса (B подтянут к шине +5 В через резистор 47 кОм), C — GND, D — роль устройства (низкий уровень для устройства 1).

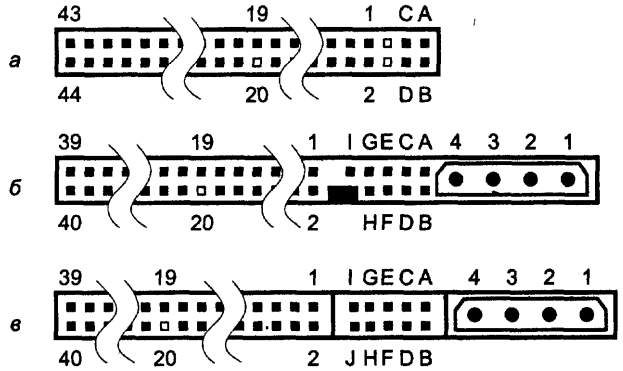

Рис. 5.4. Дополнительные контакты на разъемах интерфейса ATA: а — SFF8057, б — SFF8058, в — SFF8212 (50-контактный разъем)

Таблица 5.4. Использование дополнительных контактов

Выборка	SFF8057	SFF8058	SFF8212	Toshiba (1996)
Используемые контакты	E-H	A-F	A-D	A-D
Кабельная выборка	E-F	A-B	B-D	B-D
Master	G-H	E-F	-	-
Master при наличии Slave	G-H,E-F	E-F	-	-
Slave	-	C-D	A-B	C-D

Для малогабаритных внешних устройств существует довольно распространенный разъем HP 36, но в спецификацию ATA/ATAPI он не входит. Для устройств хранения на флэш-памяти используется коннектор, соответствующий спецификации Compact Flash Association.

В документации на устройства могут применяться несколько отличающиеся обозначения сигналов. Здесь приведены обозначения из стандарта ATA/ATAPI-4. Сигналы имеют следующее назначение:

RESET# (Device Reset) — сброс устройства (инвертированный сигнал сброса системной шины). Сигнал длительностью не менее 25 мкс вырабатывается после установления питающих напряжений. Сигнал вызывает сброс интерфейса в исходное состояние и устанавливает параметры по умолчанию.

DA[2:0] (Device Address) — три младших бита системной шины адреса, используемые для выбора регистров устройств.

DD[15:0] (Device Data) — двунаправленная 16-битная шина данных между адаптером и устройствами. При 8-битных обменах используются младшие биты D[7:0]. У устройства не должно быть «подтягивающего» резистора на линии DD7— на хост-контроллере эта линия через резистор 10 кОм соединена с шиной GND. Это позволяет хосту определить отсутствие устройства на шине сразу после аппаратного сброса: при чтении регистра состояния отсутствующего устройства бит *BSY* окажется сброшенным.

DIOR# (Device I/O Read) — строб чтения портов ввода/вывода. Данные фиксируются по положительному перепаду сигнала.

DIOW# (Device I/O Write) — строб записи портов ввода/вывода. Данные фиксируются по положительному перепаду сигнала.

IORDY (I/O channel Ready) — готовность устройства завершить цикл обмена. Низким уровнем сигнала во время цикла обмена устройство может ввести такты ожидания шины. Сигнал требуется при обмене в PIO Mode 3 и выше. На хост-контроллере эта линия через резистор 1 кОм должна

подтягиваться к шине питания.

IOCS16# — разрешение 16-битных операций. Обращение ко всем регистрам, кроме регистра данных, всегда 8-битное. Для PIO Mode 0,1, 2 при активном сигнале обращения 16-битные, при неактивном — 8-битные. Для PIO Mode 3,4 и DMA все обмены 16-битные, кроме дополнительных байтов (выходящих за границу 512-байтного сектора) «длинного» считывания и записи. Начиная с ATA/ATAPI-3 не используется.

DMARQ (DMA Request) — запрос обмена по каналу DMA (сигнал необязательный). При разрешении обмена сигнал (высокий уровень) вводится устройством по готовности к обмену. Введя сигнал DMARQ, устройство должно дождаться подтверждения от хост-адаптера сигналом DMACK#, после чего может снять запрос DMARQ. Для очередной передачи запрос должен быть введен снова. В режиме Multiword DMA запрос может удерживаться на время передачи всех данных. Выход должен быть тристабильным, во время работы с DMA он может быть в активном состоянии (0 или 1) только у выбранного устройства. В ATA-1 для этого сигнала мог использоваться как тристабильный, так и стандартный ТТЛ-выход. Работа на одной шине устройств с разнотипными выходами DMARQ может привести к конфликтам. На хост-контроллере линия должна соединяться с шиной GND через резистор 5,6 кОм.

DMACK# (DMA Acknowledge) — подтверждение DMA. Сигнал вырабатывается хост-адаптером как подтверждение цикла передачи. Передача слова данных управляется сигналами DIOR# или DIOW#. Во время обмена по каналу DMA сигналы IOCS16#, CS0# и CS1# не используются, обмен всегда производится 16-битными словами.

INTRQ (Interrupt Request) — запрос прерывания. Выход должен быть тристабильным, активный сигнал (логическую 1) вырабатывает только выбранное устройство, когда у него имеется необслуженный запрос прерывания и его вырабатывание не запрещено битом *nIEN* в регистре *Device Control.* Запрос сбрасывается по сигналу RESET#, установке бита *SRST* в регистре *Device Control,* записи в регистр команд или чтении регистра состояния. При обменах PIO запрос устанавливается в начале передачи каждого блока (сектора или группы секторов при многосекторных операциях). Исключения: по командам *Format Track, Write Sector(S), Write* Buffer и Write Long в начале передачи первого блока данных запрос прерывания не вырабатывается. При обменах DMA запрос прерывания вырабатывается только по завершении операции. На хост-контроллере эта линия через резистор 10 кОм должна подтягиваться к шине GND.

CS0# (Chip Select 0) — сигнал выбора блока командных регистров *(Command Block Registers).* Для первого канала он вырабатывается при наличии на системной шине адреса порта ввода/вывода в диапазоне 1F0h - 1F7h (сигнал также называют CS1FX#).

CS1# (Chip Select 1) — выбор блока управляющих регистров *(Control Block Registers).* Для первого канала он вырабатывается при наличии на системной шине адреса порта ввода/вывода в диапазоне 3F6h - 3F7h (часто этот сигнал называется CS3FX#).

PDIAG#'(Passed Diagnostics) — сигнал о прохождении диагностики. Ведущее устройство наблюдает за этим сигналом, который ведомое устройство должно выработать в ответ на сброс или команду диагностики. Если ведомое устройство обнаружено (по сигналу DASP#), ведущее устройство ожидает сигнал в течение 31с после сброса и 6 с после команды диагностики. Если за, это время сигнал не появился, ведущее устройство отмечает этот факт установкой бита 7 регистра ошибок. Если ведомое устройство не обнаружено, ведущее обнуляет регистр состояния ведомого устройства и сообщает свое состояние сразу после завершения собственной самодиагностики. Сигнал служит только для связи двух устройств и хост-адаптером не используется (в ATA-4 контакт задействован для сигнала CBLID#).

CBLID# (Cable assembly type identifier) — идентификация типа кабеля. В 80-проводной сборке контакт 34 на разъеме хост-адаптера соединяется с шиной GND, а контакты 34 разъемов устройств соединяются между собой, но связи с разъемом хост-адаптера не имеют. На устройстве эта линия через резистор 10 кОм должна подтягиваться к шине питания. После прохождения сброса (когда сигнал PDIAG# снимается) хост может определить наличие 80-проводного кабеля по низкому уровню сигнала.

DASP# (Device Active, Slave Present) — сигнал двойного назначения: индикатор активности устройства и присутствия ведомого устройства. Устройства имеют выход типа «открытый коллектор» с нагрузочным резистором 10 кОм к шине +5 В. После сброса по сигналу RESET# или при инициализации по включении питания оба устройства в течение 1 мс должны деактивировать этот сигнал, после чего не позже, чем через 400 мс, его вводит ведомое устройство для сообщения о своем присутствии. Ведущее устройство не активирует этот сигнал в течение 450 мс.

Сигнал деактивируется ведомым устройством после получения им команды или через 31с автоматически (смотря что произойдет раньше). После этого сигнал может быть введен любым устройством как индикатор активности. Адаптер использует этот сигнал для включения светодиодного индикатора доступа к диску.

SPSYNC/CSEL (Spindle Synchronization/Cable Select) — синхронизация шпинделя/ выборка кабелем. Сигнал двойного назначения, которое должно быть единым для обоих устройств. Сигнал SPSYNC позволяет синхронизировать шпиндели устройств (актуально для RAID-массивов); используется по

усмотрению производителя накопителя, начиная с ATA/ATAPI-3 из спецификации изъят. Сигнал CSEL позволяет устройствам определять свой адрес по положению на специальном кабеле с разрывом провода 28 между разъемами двух устройств (малораспространенная «кабельная выборка»). Эта линия на хост-адаптере заземлена, и ведущее устройство получает заземленную линию, а ведомое — неподключенную. На устройстве линия подтягивается к высокому уровню резистором 10 кОм. Состояние сигнала (если он управляется хост-адаптером) должно удерживаться по крайней мере 31с после сигнала RESET#.

При использовании режима Ultra DMA три линии получают новое назначение сигналов:

- STOP (Stop Ultra DMA burst) (23) — останов передачи пакета Ultra DMA.
- DDMARDY# (Device Ultra DMA ready) (25) — готовность устройства при приеме пакета Ultra DMA (управление потоком).
- DSTROBE (Host Ultra DMA data strobe) (27) — строб данных устройства при передаче пакета хосту. Данные передаются по обоим перепадам DSTROBE.
- HDMARDY# (Host Ultra DMA ready) (25) — готовность хоста при приеме им пакета Ultra DMA (управление потоком).
- HSTROBE (Host Ultra DMA data strobe) (27) — строб данных хоста при передаче пакета устройству. Данные передаются по обоим перепадам HSTROBE.

Для блокнотных ПК в стандарте имеется вариант интерфейса IDE на 68-контактном разъеме PCMCIA (PC Card), приведенный в табл. 5.5. Здесь имеется ряд специфичных сигналов:

SELATA# (Select 68-pin ATA) — сигнал, которым хост указывает режим использования разъема, PC Card (сигнал снят) или ATA (сигнал установлен, то есть низкий уровень). Этот сигнал хост должен установить до подачи питания на коннектор. В течение 19 мс после подачи питания устройство игнорирует все интерфейсные сигналы, кроме этого. Если этот сигнал активен, то устройство должно сконфигурироваться на режим ATA. Если сигнал неактивен, оно должно сконфигурироваться в режим PC Card или не отвечать на другие сигналы хоста.

CD1# и CD2# (Card Detect) заземляются в устройстве — по ним хост определяет присутствие устройства.

CS1# (Device chip select 1) — сигнал подается хостом на оба контакта (11 и 42), но устройство воспринимает только один из них.

Сигналы DMARQ, DMACK# и IORDY не обязательны для использования.

M/S# (Master/Slave) — инверсия сигнала CSEL. Хост выдает сигналы M/S# и CSEL до подачи питания, устройство воспринимает лишь один из них.

Для обеспечения «горячего» подключения разъем для цепи GND обеспечивает более раннее соединение при подключении и более позднее при отключении. В устройстве сигналы CS0#, CS1#, RESETS и SELATA# подтягиваются к пассивному состоянию.

Таблица 5.5. 68-контактный интерфейс ATA для PC Card (PCMCIA)

Конта	Сигнал	Контак	Сигнал
1	GND	35	GND
2	DD3	36	CD1#
3	DD4	37	DD11
4	DD5	38	DD12
5	DD6	39	DD13
6	DD7	40	DD14
7	CS0#	41	DD15
8		42	CS1#
9	SELATA#	43	
10		44	DIOR#
11	CS1#	45	DIOW#
12		46	
13		47	
14		48	
15		49	
16	INTRQ	50	
17	+5 B	51	+5 B
18		52	
19		53	
20		54	
21		55	M/S#

Таблица 5.5. (Продолжение)

Контакт	Сигнал	Контакт	Сигнал
22		56	CSEL
23		57	
24		58	RESET#
25		59	IORDY#
26		60	DMARQ
27	DA2	61	DMACK#
28	DA1	62	DASP#
29	DA0	63	PDIAG#
30	DD0	64	DD8
31	DD1	65	DD9
32	DD2	66	DD10
33		67	CD2#
34	GND	68	GND

Для *компьютеров класса XT* существует 8-битная версия интерфейса, называемая XT-IDE, реже — XT-Bus. Этот интерфейс (табл. 5.6), как и ATA, реализован на 40-проводном кабеле, и многие его сигналы совпадают с 16-битной шиной ATA. Интерфейс XT-IDE можно рассматривать как подмножество ATA, хотя прямой совместимости нет. Ряд устройств ATA имеют джампер выбора режима XT/AT (в накопителях Seagate на это указывает окончание «AX» в обозначении модели).

Таблица 5.6. Кабель интерфейса XT IDE

Сигнал	Тип[1]	Контакт	Контакт	Сигнал
RESET#	1	1	2	GND
DD7	1/O TS	3	4	GND
DD6	1/O TS	5	6	GND
DD5	1/O TS	7	8	GND
DD4	1/O TS	9	10	GND
DD3	1/O TS	11	12	GND
DD2	1/O TS	13	14	GND
DD1	1/O TS	15	16	GND
DD0	1/O TS	17	18	GND
GND	-	19	20	Ключ (нет штырька)
AEN	1	21	22	GND
DIOW#	1	23	24	GND
DIOR#	1	25	26	GND
DMACK#	1	27	28	GND
DMARQ	O OK	29	30	GND
INTRQ	O TS[2]	31	32	GND
DA1	1	33	34	GND
DA0	1	35	36	GND
CS0#	1	37	38	GND
DASP#	I/O OC[3]	39	40	GND

[1] Тип сигнала для устройства: I — вход, O — выход, I/O — двунаправленный, TS — тристабильный, OK — открытый коллектор.

[2] У старых устройств сигнал может иметь тип OK (при разнотипных сигналах на одной шине возможен конфликт).

[3] У ведомого устройства — только выход.

5.2.1.2. Регистры устройств

Каждое устройство ATA имеет стандартный набор регистров, адресуемых сигналами от хост-адаптера (CS0#, CS1#, DA2, DA1, DA0, DIOR# и DIOW#). Набор регистров (табл. 5.7) состоит из двух блоков, выбираемых сигналами CS0# и CS1#, из которых активным (низкий уровень, «0») может быть только один. В таблице приведены адреса регистров в пространстве ввода-вывода IBM PC-совместимого ПК для первого и второго каналов ATA. При обращении к регистрам сигнал DMACK# должен быть неактивным. *Блок командных регистров* служит для посылки команд устройству и чтения информации о его состоянии. *Блок управляющих регистров* используется для управления устройством и получения более подробной информации о его состоянии. На действительность содержимого регистров командного блока и альтернативного регистра состояния указывает нулевое значение бита *BSY* регистра состояния. Запись в регистры должна производиться лишь при BSY=0 и DRQ=0, кроме особо оговоренных случаев.

Если устройство поддерживает управление энергопотреблением, в «спящем» режиме содержимое этих регистров недействительно и запись игнорируется, кроме особо оговоренных случаев.

Таблица 5.7. Регистры контроллеров устройств ATA

Адрес для канала №		Сигналы адресации (0-низкий уровень, 1-высокий)					Назначение (R — чтение, W — запись)
1	2	CS0#	CS1#	DA2	DA1	DA0	
		1	1	X	X	X	Нет обращения (шина данных в третьем состоянии)
		0	0	X	X	X	Недопустимый адрес (шина данных в третьем состоянии)
3FX	37X	Control Block Registers – блок управляющих регистров					
		1	0	0	X	X	Не используется (шина данных в третьем состоянии)
		1	0	1	0	X	Не используется (шина данных в третьем состоянии)
3F6	376	1	0	1	1	0	R: *Alternate Status* (AS} — альтернативный регистр состояния
3F6	376	1	0	1	1	0	W: *Device Control* (DC) — регистр управления устройством
3F7	377	1	0	1	1	1	R: *Drive Address (DA)* — регистр адреса (не используется)[1]
1FX	17X	Command Block Registers – блок командных регистров					
1F0	170	0	1	0	0	0	R/W: *Data* (DR) — регистр данных
1F1	171	0	1	0	0	1	R: *Error* (ER) — регистр ошибок
1F1	171-	0	1	0	0	1	W: *Features* (FR) — регистр свойств
1F2	172	0	1	0	1	0	R/W: Sector *Count* (SC) — регистр счетчика секторов
1F3	173	0	1	0	1	1	R/W: *Sector Number* (SN) — регистр номера сектора/LBA [7:0][2]
1F4	174	0	1	1	0	0	R/W: *Cylinder Low(CL)* — регистр младшего байта номера цилиндра/LBA[15:8][2]
1F5	175	0	1	1	0	1	R/W: *Cylinder High* (CH) — регистр старшего байта номера цилиндра/LBA[23:16][2]
1F6	176	0	1	1	1	o	R/W: *Device/Head* (D/H) —регистр номера устройства и головки/LBA[27:24][2]
1F7	177	0	1	1	1	1	R: *Status* (SR) — регистр состояния
1F7	177	0	1	1	1	1	W: *Command* (CR) — регистр команд

[1] Рекомендуется, чтобы на сигнал чтения по этому адресу устройство не отвечало.

[2] Регистры сектора, цилиндра и головки в режиме LBA содержат указанные биты логического адреса.

Альтернативный регистр состояния *AS* (для первого канала адрес 3F6h, для второго — 376h) имеет те же биты, что и основной (см. ниже), но его чтение не приводит ни к каким изменениям состояния устройства.

Регистр управления устройством *DC* (3F6h, 376h) служит для программного сброса обоих устройств одновременно и управления разрешением прерывания выбранного устройства. Запись в этот регистр возможна в любой момент. Программный сброс через регистр *DC* должен отрабатываться и в состоянии *Sleep.*

Назначение битов регистра DC:

- Биты [7:3] зарезервированы.
- Бит 2 — SRST (Software Reset) — программный сброс, действует все время, пока бит не будет снят. Оба устройства на шине воспринимают программный сброс одновременно.
- Бит 1 — nIEN (Interrupt Enable) — инверсный бит разрешения прерывания. При нулевом значении бита выбранное устройство может вырабатывать сигнал *INTRQ* через тристабильный выход.
- Бит 0 = 0.

Регистр адреса устройства *DA* (3F7h, 377h) использовался только в первой версии ATA для совместимости со старыми контроллерами, чтением этого регистра можно было определить адресованный привод и головку. Регистр выпадает из блока (он совпадает с диагностическим регистром состояния контроллера НГМД), и рекомендуется, чтобы устройство ATA не отвечало на чтение этого регистра. Если устройство отвечает на чтение, то оно не должно управлять битом DD7 во избежание конфликта с контроллером НГМД, у которого по этой линии передается бит смены носителя. Из-за несоблюдения этого требования могут возникать проблемы, когда контроллер (адаптер) ATA находится

на разных платах с контроллером НГМД.

Назначение битов регистра DA:

- Бит 7 — (HiZ) — высокоимпедансный, при считывании не выдается на шину.
- Бит 6 — nWTG — инверсный признак записи (во время физического выполнения записи на носитель бит нулевой).
- Биты [5:2] — nHS [3:0] — номер головки (инверсные биты).
- Биты [1:0] — nDS [1:0] — выбор устройства (инверсные биты): 10 — выбрано устройство 0, 01 — выбрано устройство 1.

Регистр данных *DR* (1F0h, 170h) может использоваться как 8-битный или 16-битный в зависимости от типа данных, передаваемых в текущей команде. Обращение к этому регистру происходит в режиме обмена PIO (когда DMACK# неактивен), при выполнении передач протокола PO (PIO Out) хост производит запись в этот регистр, при PI (PIO In) — чтение. В режиме DMA обмен данными происходит через *порт данных,* при этом активны DMARQ и DMACK#, а CS0# и CS1# неактивны.

Регистр ошибок *ER* (lFlh, 171h) хранит состояние выполнения последней операции или диагностический код. После завершения операции на Наличие ошибки указывает бит *ERR* регистра состояния.

Назначение битов регистра ER:

- Бит 7 зарезервирован.
- Бит 6 — *UNC* (Uncorrectable Data Error) — неисправимая ошибка данных.
- Бит 5 — *MC* (Media Changed) — смена носителя. После смены носителя первая команда обращения отвергается и устанавливается данный бит. После сброса бита следующие команды будут выполняться нормальным образом.
- Бит 4 — *IDNF* (ID Not Found) — указывает на ненайденный идентификатор сектора.
- Бит 3 — *ABRT* (Aborted Command) — устанавливается, если команда отвергнута как недействительная или в случае возникновения иной ошибки.
- Бит 2 — *MCR* (Media Change Requested) — индикатор запроса смены носителя. После обнаружения запроса смены носителя команды *Door Lock* будут возвращать бит ошибки *ERR* и бит *MCR.* Бит *MCR* сбрасывается командами *Door Unlock, Media Eject или* аппаратным сбросом.
- Бит 1 — *TK0NF* (Track 0 Not Found) указывает на то, что по команде *Recalibrate* не удалось найти нулевой трек.
- Бит 0 — *AMNF* (Address Mark Not Found) — не найден адресный маркер данных в заголовке сектора.

После выполнения любого сброса или команды *Execute Device Diagnostic* регистр ошибок содержит диагностический код. Трактовка битов, за исключением бита 3 *(ABRT),* может меняться в зависимости от исполненной команды.

Регистр свойств *FR* (IFlh, 171h) используется в зависимости от команды. В команде *Set Features* через него задается код подкоманды. Некоторые старые устройства могут игнорировать запись в этот регистр. До принятия ATA-2 в этот регистр помещали значение рекомендуемого номера цилиндра для предкомпенсации записи.

Регистр счетчика секторов *SC* (lF2h, 172h) содержит число секторов, участвующих в обмене. Хост инициализирует этот регистр до подачи команды (нулевое значение соответствует 256 секторам). По успешному завершению операции доступа к данным регистр должен обнулиться. Если команда завершилась с ошибкой, в регистре будет число секторов, которые должны быть переданы для успешного завершения предыдущего запроса. Команды *Initialize Device Parameters или Write Same* могут переопределить значение этого регистра. В некоторых командах регистр используется для передачи иных параметров.

Регистры номера сектора *SN* (1F3h, 173h) и ***номера цилиндра*** — младшего *CL* (1F4h, 174h) и старшего байта *CH* (lF5h, 175h) — имеют двоякое назначение в зависимости от выбранной системы адресации (CHS или LBA). Они инициализируются хост-адаптером, а в случае возникновения ошибки при выполнении операции устройство поместит в них адрес, по которому встретилась ошибка.

Регистр номера устройства и головки *D/H* (lF6h, 176h) кроме хранения части адресной информации служит для выбора ведущего или ведомого устройства и метода адресации.

Назначение битов регистра D/H:

- Биты 7 и 5 вплоть до ATA-3 должны были быть единичными, в ATA/ATAPI-4 их объявили устаревшими.
- Бит 6 — *L* — единичным значением указывает на применение режима адресации LBA. При нулевом значении бита используется режим CHS.
- Бит 4 — *DEV* (Device) — выбор устройства. При *DEV=0* выбрано ведущее, при *DEV=1* — ведомое.

- Биты [3 : 0] имеют двоякое назначение в зависимости от выбранной системы адресации. В режиме CHS они содержат номер головки, в режиме LBA — старшие биты логического адреса.

Как и предыдущие (SN, CH и CL), этот адресный регистр инициализируется хост-адаптером, а в случае возникновения ошибки при операции устройство поместит в них адрес, по которому встретилась ошибка. До принятия ATA-2 считалось, что адресные регистры должны модифицироваться и после успешного выполнения операции, отражая текущее значение адреса в носителе.

Регистр состояния *SR* (lF7h, 177h) отражает текущее состояние устройства в процессе выполнения команд. Чтение регистра состояния разрешает дальнейшее изменение его битов и сбрасывает запрос аппаратного прерывания.

Назначение битов регистра SR:

- Бит 7 — *BSY* (Busy) указывает на занятость устройства, значение этого бита действительно всегда. При *BSY=1* устройство игнорирует попытки записи в командный блок регистров, а чтение этих регистров дает неопределенный результат. При *BSY=0* регистры командного блока доступны, в это время устройство не может устанавливать бит *DRQ,* изменять значение битов *ERR* и содержимое остальных командных регистров (могут меняться только значения битов *IDX, DRDY, DF, DSC* и *CORR).* Бит может устанавливаться на кратковременный интервал, так что хост может этого не заметить. Бит устанавливается:
  - при сбросе устройства;
  - по получении команды, если не устанавливается *DRQ;*
  - между передачами блоков данных в режиме PIO и после них, пока не обнулился *DRQ;*
  - во время передач данных в режиме DMA.
- Бит 6 — *DRDY* (Device Ready) указывает на готовность устройства к восприятию любых кодов команд. Если состояние бита изменилось, оно не может измениться обратно до чтения регистра состояния. При *DRDY=0* устройство воспринимает только команды *Execute Device Diagnostic* и *Initialize Device Parameters,* прекращая выполнение текущей команды и сообщая об этом флагом *ABRT* в регистре ошибок и *ERR* в регистре состояния. Другие команды приводят к непредсказуемым результатам. Устройства ATAPI сбрасывают бит по любому сбросу и команде *Execute Device Diagnostic.* Бит устанавливается устройством ATA, когда оно готово к выполнению всех команд. Устройство ATAPI устанавливает бит до завершения выполнения команд, за исключением *Device Reset* и *Execute Device Diagnostic.*
- Бит 5 — *DF* (Device Fault) — индикатор отказа устройства.
- Бит 4 — *DSC* (Device Seek Complete) — индикатор завершения поиска трека. В командах, допускающих перекрытие (см. п. 5.2.1.7), бит называется SERV (Service Required) — устройство требует обслуживания.
- Бит 3 — *DRQ* (Data Request) — индикатор готовности к обмену словом или байтом данных.
- Бит 2 — *CORR* (Corrected Data) — индикатор исправленной ошибки данных.
- Бит 1 — *IDX* (Index) — индекс, трактуется особо каждым производителем.
- Бит 0 — *ERR* (Error) — индикатор ошибки выполнения предыдущей операции. Дополнительная информация содержится в регистре ошибок. Если установлен бит *ERR,* до приема следующей команды, программного или аппаратного сброса устройство не изменит состояние этого бита, а также регистра ошибок, регистра количества секторов и регистров цилиндра, головки и номера сектора. Для команд *Packet* и *Service* бит называется *CHK* и служит признаком исключительной ситуаций.

В стандарте ATA/ATAPI-4 для некоторых команд биты 4 и 5 могут иметь иное назначение, а биты 1 и 2 объявлены устаревшими.

Назначение ***регистра команд*** *CR* (lF7h, 177h) очевидно из названия. Устройство начинает исполнять команду сразу, как только ее код записан в данный регистр. Команда *Device Reset* выполняется устройством ATAPI независимо от состояния битов *BSY* и *DRQ,* и даже в состоянии *Sleep.*

5.2.1.3. Адресация данных

В устройствах ATA/ATAPI минимальной адресуемой единицей данных является *логический блок,* как правило, размером 512 байт, который для традиционных дисковых устройств ATA является *сектором.* Применительно к секторам для устройств ATA существует традиционная *трехмерная адресация CHS:*

- Cylinder — номер цилиндра, который задается регистрами *CH* и *CL* (старший и младший байты) и может принимать значение 0-65535 (2^{16}-1). Устройство может иметь до 65 536 (2^{16}) цилиндров (Cyl).
- Head — номер головки, который задается битами 3:0 регистра *D/H* и может принимать значение 0-15 (2^4-1). Устройство может иметь до 16 (2^4) головок в цилиндре (HPC — Heads Per Cylinder).
- Sector — номер сектора, который задается регистром *SN* и может принимать значение 1-255 (2^8-1), нулевой номер сектора не используется. Устройство может иметь до 255 (2^8-1) секторов на каждом

треке (SPT — Sectors Per Track).

Здесь подразумевается ***внешняя геометрия*** — адреса, которые заносятся в командные регистры устройств. Реальная внутренняя геометрия устройств (реальное количество цилиндров, головок и секторов на треке, как правило, отличается от внешней, к которой обращаются через регистры (хотя бы потому, что у современных устройств число физических секторов меняется от трека к треку). В системе CHS устройство ATA позволяет адресовать до 267 386 880 (65 536×16×255) секторов (блоков), что при размере сектора в 512 байт дает 136 902 082 560 байт (около 137 Гбайт).

Геометрия по умолчанию (схема трансляции после включения питания) отражается словами 1, 3 и 6 блока параметров идентификации устройства (см. табл. 5.10). Эта схема может быть изменена по команде *Initialize Device Parameters,* и *текущая геометрия* отражается в словах 54, 55 и 56, а текущий объем (произведение C×H×S) — в словах 57-58. По команде *Initialize Device Parameters* хост «заказывает» желаемые значения SPT и HPC, число цилиндров устройство вычисляет само. Если устройство не способно принять предлагаемую конфигурацию, эта команда будет отвергнута и будет принята конфигурация по умолчанию (слова 1, 3 и 6), при этом слова 57-58 могут быть нулевыми.

Логическая адресация *LBA* (Logical Block Addressing) гораздо проще — здесь адрес блока (сектора) определяется одним 28-битным числом в диапазоне 0-268 435 455 (2^{28}-1), но, естественно, в пределах физических возможностей устройства. Число секторов, доступных в режиме LBA, отражается в словах 60-61.

Попытки обращения к сектору с номером LBA, превышающим максимально возможный, или же с координатами C, H, S, не вписывающимися в текущую геометрию, приведут к генерации ошибки *IDNF* или *ABRT* (обмена данными, естественно, не будет).

Возможность адресации LBA была заложена в самой первой спецификации ATA, и правила пересчета адресов CHS и LBA однозначны:

LBA = (C × HPC + H) × SPT + S - l, где C, H и S - «координаты» сектора, HPC - количество головок в цилиндре, SPT — количество секторов на треке. Эта же формула поясняет и действия при последовательном обращении: сначала нарастает номер сектора, затем номер головки, а потом номер цилиндра.

Для устройств ATA, поддерживающих и CHS, и LBA, режим адресации определяется для каждой команды битом *L* (бит 6) регистра *D/H;* режимы могут чередоваться произвольным образом. При изменении текущей геометрии логический адрес (LBA) любого реального сектора остается неизменным.

Геометрия может быть выбрана относительно произвольно, количество полных цилиндров будет определяться как целая часть от V/(SPT×HPC). При этом могут оставаться неполные цилиндры, доступ к которым не гарантирован для всех устройств. По этой причине, когда для диска меняют геометрию (опциями BIOS Setup), доступный объем может варьироваться. Геометрию стараются выбирать с таким расчетом, чтобы количество «висящих» секторов (orphan sectors) было минимальным. Оптимальной конфигурации, как правило, соответствуют значения HPC и SPT, взятые из слов 3 и 6 соответственно. Число «висящих» секторов не будет превышать SPT×HPC-1.

Стандарты ATA и ATA-2 допускают поддержку обоих режимов адресации, а на поддержку LBA указывает бит 9 в слове 49 паспорта диска.

В стандартах начиная с ATA-3 поддержка LBA для всех устройств обязательна, а устройства с объемом до 16 515 072 секторов включительно (около 8 Гбайт) должны поддерживать хотя бы одну схему CHS (определяемую словами 1, 3 и 6). В ATA-4 и далее это число уменьшено — 16 514 064. Для устройств большего объема поддержка CHS не обязательна. Если CHS поддерживается, то геометрия по умолчанию определяется словами 1,3 и 6, а текущая — словами 53-58. Если CHS не поддерживается, то все эти слова — нулевые.

В ATA-3 принимаются следующие правила определения геометрии по умолчанию. Для устройств объемом менее 1 032 192 секторов (528 Мбайт) число цилиндров (слово 1) не должно превышать 1024 (при этом не требуется трансляции в BIOS). Для устройств объемом до 16 515 072 секторов значения слова 1 принимаются в зависимости от числа головок (слова 3) в соответствии с табл. 5.8. Число секторов на треке (слово 6) устанавливается в пределах 1 - 63.

Таблица 5.8. Определение геометрии по умолчанию

Слово 3	Максимальное значение слова 1
1 - 4	65535
5 - 8	32767
10 - 16	16383

В приложении B к ATA/ATAP1-4 и приложении C к ATA/ATAPI-5,6 правила для геометрии по умолчанию несколько изменили: для устройств объемом до 16 514 063 секторов включительно число цилиндров (слово 1) может быть в диапазоне 1-65 535, в зависимости от слова 3 (см. табл. 5.8). Для устройств большего размера (там указано, что для устройств с более, чем 15 481 935 секторами) слово 1

должно быть 16 383 (3FFFh). Число головок (слово 3) может быть в пределах 1-16. При общем числе секторов не более 8 257 536 устанавливается значение 16 головок. Чтобы при трансляции методом сдвига не получилось 256 головок — рокового числа для ряда BIOS и ОС (барьер 4,2 Гбайт), при большем числе секторов может приниматься число головок 15. Число секторов на треке (слово 6) не должно превышать 63 (3Fh), для дисков с общим числом секторов более 1 032 193 принимается 63 сектора на трек.

Слова 54-56, определяющие текущую геометрию, обязательно должны использоваться устройствами с общим числом секторов более 1 032 192 (528 Мбайт). Они показывают объем диска, адресуемый в системе CHS, их произведение (оно же в словах 57-58) не должно превышать 16 514 064 (8,5 Гбайт). Это искусственное ограничение на сообщение параметров введено для обеспечения совместимости с традиционным сервисом BIOS, имеющим программный интерфейс CHS. Слова 60-61 содержат 32-битное значение реального размера диска, этот полный объем безоговорочно доступен в режиме адресации LBA.

Текущая геометрия может заказываться любой, в пределах габаритов 65 536×16×255. Команда *Set Max Address* может уменьшить число цилиндров и общее число доступных секторов в текущей конфигурации.

Современные диски объемом более 8,4 Гбайт сообщают CHS-геометрию по умолчанию 16 383×16×63=16 514 064 секторов (8,4 Гбайт), хотя из приложений к спецификации ATA следовала бы геометрия 16 383×15×63=15 481 935 секторов (7,9 Гбайт) — близкая к предельно возможной для традиционных сервисов BIOS *INT 13h*.

5.2.1.4. Система команд

Стандарт ATA задает систему команд, ориентированную на накопители на магнитных дисках. В табл. 5.9 графа «Протокол» определяет протокол передачи данных, требуемый для команды (см. п. 5.2.1.6): *PI* — ввод данных от устройства в режиме PIO, *PO* — вывод данных в устройство в режиме PIO, *P* — протокол передачи командного пакета (16 байт вывода в режиме PIO), *DM* — обмен данными по каналу DMA, *DMQ* — обмен DMA с очередями, *ND* — нет обмена данными, *DP* — протокол сброса, *VS* — специфично для устройства. Графа «Тип» определяет характеристику команды, раздельно для устройств ATA и ATAPI: О — обязательная, Д — дополнительная, *Н* — неприменима, С — специфическая, У—устаревшая, *И* — исключенная. Цифра после буквы означает номер версии спецификации ATA/ATAPI, в которой тип менялся. Графа «Код» содержит шестнадцатеричный код, загружаемый в регистр команды. В графе «Использование регистров» приняты следующие обозначения: у — регистр содержит параметры для команды, причем для регистра *D/H* используются оба параметра; D — в *D/H* используется только номер устройства, *d* — в *D/H* задействован только номер устройства, использование поля номера головки специфично для производителя, *D0* — команда адресуется к ведущему устройству, но выполняют ее оба устройства.

Таблица 5.9. Система команд ATA

Команда	Протокол	Тип		Код	Использование регистров				
		ATA	ATAPI		PR	SC	SN	CH,CL	D/H
Acknowledge Media Change — подтверждение смены носителя	VS	Д, У3, И5		DBh	-	-	-	-	D
Soot - *Post-Boot* — после загрузки носителя	VS	Д, У3, И5		DCh	-	-	-	-	D
Boot - Pre-Boot — до загрузки носителя	VS	Д, У3, И5		DDh	-	-	-	-	D
CFA Erase Sectors — стирание секторов	ND	Д4	Н	C0h	-	У	У	У	У
CFA Request Extended Error Code — чтение расширенного кода ошибки	ND	Д4	Н	03h	-	-	-	-	D
CFA Translate Sector — получение информации о состоянии сектора	PI	Д4	Н	87h	-	-	У	У	У
CFA Write Multiple Without Erase — множественная запись без операции стирания (в предварительно стертые секторы)	PO	Д4	Н	CDh	-	У	У	У	У

Таблица 5.9. Система команд ATA (продолжение).

Команда	Протокол	Тип		Код	Использование регистров				
		ATA	ATAP1		FR	SC	SN	CH,CL	D/H
CFA Write Sectors Without Erase — запись без операции стирания (в предварительно стертые секторы)	PO	Д4	Н	38h	-	У	У	У	У
Check Power Mode — проверка режима энергопотребления	ND	Д, О4	О4	E5h (98h)[1]	-	-	У	-	D
Door Lock — запретить смену носителя	VS	Д, У4	Д	DEh	-	-	-	-	D
Door Unlock — разрешить смену носителя	VS	Д, У4	Д	DFh	-	-	-	-	D
Device Reset — сброс устройства ATAP1	DR	Н	О4	08h	-	-	-	-	D
Download Microcode — загрузка микрокода внутреннего ПО	PO	Д	Н	92h	У	У	У	У	D
Execute Device Diagnostic — диагностика	ND	О	О	90h	-	-	-	-	D=0
Flush Cache — выгрузка кэша записи на носитель (время выполнения может достигать 30с)	ND	Д4, О5	Д4, О5	E7h	-	-	-	-	D
Format Track — форматирование трека	VS	С, У4	Н	50h	-	-	-	-	D[2]
Get Media Status — получение информации о состоянии носителя (наличие, смена, защита записи)	ND	Д4	Д4	DAh	-	-	-	-	D
Identify Device — идентификация устройства (не пакетного)	PI	О	Н	ECh	-	-	-	-	D
Identify Device DMA — идентификация устройства -с ответом в режиме DMA	DM	ДЗ, У4	Н	EEh	-	-	-	-	D
Identify Packet Device — идентификация устройства (пакетного)	PI	Н	О4	A1h	-	-	-	-	D
Idle — переход в состояние ожидания	ND	Д, О4	Д	E3h (97h)[1]	-	У	-	-	D
Idle Immediate — немедленный переход в состояние ожидания	ND	Д, О4	Д, О4	E1h (95h)[1]	-	-	-	-	D
Initialize Device Parameters — инициализация параметров устройства	ND	О	Н	91h	-	У	-	-	У
Media Eject — извлечь носитель	ND	Д2	Д2	EDh	-	-	-	-	D
Media Lock — заблокировать носитель	ND	Д4	Д4	DEh	-	-	-	-	D
Media Unlock — разблокировать носитель	ND	Д4	Д4	DFh	-	-	-	-	D
Nop — холостая команда	ND	Д	Д	00h	-	-	-	-	D
Packet—посылка командного пакета	P	Н	О4	A0h	У	У	У	У	D
Read Buffer — чтение буфера	PI	Д	Н	E4h	-	-	-	-	D
Read DMA (w/retry) — чтение по DMA с повторами	DM	О3	Н	C8h	-	У	У	У	У
Read DMA (w/o retry) — чтение по DMA без повторов	DM	О3, У5	Н	C9h	-	У	У	У	У

Таблица 5.9. Система команд ATA (продолжение).

Команда	Протокол	Тип		Код	Использование регистров				
		ATA	ATAPI		FR	SC	SN	CH,CL	D/H
Read DMA Queued — чтение no DMA с возможностью постановки команды в очередь и освобождения шины	DMQ	Д4	Н	C7h	У	У[2]	У	У	У
Read Long (w/retry) — «длинное» чтение с повторами	PI	Д, У4	Н	22h	-	У	У	У	У
Read Long (w/o retry) — «длинное» чтение без повторов	PI	Д, У4	Н	23h	-	У	У	У	У
Read Multiple — множественное чтение	PI	Д3, О4	Н	C4h	-	У	У	У	У
Read Native Max Address — чтение максимального адреса сектора (заводской установки)	ND	Д4	Н	F8h	-	-	-	-	D
Read Sectors) (w/retry) — чтение секторов с повторами	PI	О	О	20h	-	У	У	У	У
Read Sectors) (w/o retry) — чтение секторов без повторов	PI	О, У5	Н	21h	-	У	У	У	У
Read Verify Sectors) (w/retry) — верификационное чтение секторов с повторами	ND	О	Н	40h	-	У	У	У	У
Read Verify Sectors) (w/o retry) — верификационное чтение секторов без повторов	ND	О, У5	Н	41h	-	У	У	У	У
Recalibrate — рекалибровка (поиск нулевого трека)	ND	Д, У4	Д, У4	10h (1Xh)[1]	-	-	-	-	D
Security Disable Password — отмена защиты (с посылкой пароля)	PO	Д3	Д3	F6h	-	-	-	-	D
Security Erase Prepare — подготовка с защитному стиранию	ND	Д3	Д3	F3h	-	-	-	-	D
Security Erase Unit — защитное стирание (с посылкой пароля)	PO	Д3	Д3	F4h	-	-	-	-	D
Security Freeze Lock — блокировка команд защиты	ND	Д3	Д3	F5h	-	-	-	-	D
Security Set Password — установка пароля защиты	PO	Д3	Д3	Flh	-	-	-	-	D
Security Unlock — снятие защиты	PO	Д3	Д3	F2h	-	-	-	-	D
Seek—поиск	ND	О	Н	70h (7Xh)[1]	-	-	У	У	У
Service — передача данных и/или состояния команды, освободив-шей шину	P или DMQ	Н	Д4	A2h	-	У	У	У	D
Set Features — установка свойств	ND	Д, О3	Д, О3	EFh	У	-	-	-	D
Set Max — ограничение доступного размера с пародированием этой возможности	ND	Д5	Н	F9h	У	У	У	У	У
Set Max Address — ограничение доступного размера	ND	Д4	Н	F9h	-	У[2]	У	У	У
Set Multiple Mode — установка множественного режима	ND	О3	Н	C6h	-	У	-	-	D
Sleep — перевод в «спящий режим»	ND	Д, 04	Д, 04	E6h (99h)[1]	-	-	-	-	D

Таблица 5.9. Система команд ATA (продолжение).

Команда	Протокол	Тип		Код	Использование регистров				
		АТД	ДТАРІ		FR	SC	SN	CH.CL	D/H
SMART— группа команд мониторинга	ND.PI	Д3	Д3	B0h	У	У[2]	У[2]	У[2]	D
Standby — перевод в дежурный режим	ND	Д, О4	Д	Е2П (96h)[1]	-	У	-		D
Standby Immediate — немедленный перевод в дежурный режим	ND	Д, 04	Д, 04	E0h (94h)[1]					D
Write Buffer — запись в буфер	PO	Д	Н	E8h	-	-	-	-	D
Write DMA (w/retry) — запись по DMA с повторами	DM	О3	Н	CAh	-	У	У	У	У
Write DMA (w/o retry) — запись по DMA без повторов	DM	О3, У5	Н	CBп	-	У	У	У	У
Write DMA Queued — запись по DMA с возможностью постановки команды в очередь и освобождения шины	DMQ	Д4	Н	CCh	У	У	У	У	У
Write Long (w/retry) — «длинная» запись с повторами	PO	Д, У4	Н	32h	-	У	У	У	У
Write Long (w/o retry) «длинная» запись без повторов	PO	Д, У4	Н	33h	-	У	У	У	У
Write Multiple — множественная запись	PO	О3	Н	C5h	-	У	У	У	У
Write Same — запись с размножением	PO	Д, У2, И5	Н	E9h	У	У	У	У	У
Write Sectors) (with retry) — запись секторов с повторами	PO	О	Н	30h	-	У	У	У	У
Write Sector(S) (without retry) — запись секторов без повторов	PO	О, У5	Н	31h	-	У	У	У	У
Write Verify — запись с верификацией	PO	Д, У4	Н	3Ch	-	У	У	У	У

[1] Старый код команды.

[2] Использование регистра зависит от производителя.

Коды команд 9Ah,C0h - C3h, 8xh, F0h, F7h и FAh - FFh выделяются производителям оборудования для реализации специфических команд. Команды с кодами 10h, 21h-23h, 31h-33h, 3Ch, 41h, 50h, C9h, CBh и EEh в спецификации ATA/ATAPI-5 объявлены устаревшими (эти команды не рекомендуется использовать). В этой же спецификации коды 11h - lFh, 71h - 7Fh, 94h - 99h, DBh - DDh, E9h удалены из списков «живых», использование этих команд запрещается (в дальнейшем они могут быть определены для иных команд).

5.2.1.4.1. Основные команды

К этой группе относятся команды ATA для доступа к данным. Для устройств, существенно отличающихся от НЖМД, имеется альтернативный способ доставки команд и параметров — команда *Packet*. По этой команде, используя вывод в режиме PIO, хост-адаптер передает пакет с блоком дескриптора команды. Подробнее о пакетном интерфейсе см. в п. 5.2.1.8.

Минимальной адресуемой единицей команд чтения и записи является 512-байтный сектор.

Команды чтения секторов в режимах обмена PIO и DMA — *Read Sector(s)* и *Read DMA* — позволяют считывать последовательно расположенные секторы, количество которых задано в регистре SC, а адрес начального сектора — в регистрах *CH, CL, D/H* и *SN*. Команды чтения бывают с повторами или без них. В первом случае, если при чтении сектора обнаружена неисправимая ошибка, устройство автоматически делает несколько повторных попыток чтения. После исполнения команды в случае неисправимой ошибки блок командных регистров содержит адрес сектора с ошибкой. Если устройство использует ECC-код, то некоторые ошибки чтения оно исправляет без повторов, но указывает на этот факт в регистре ошибок.

Команда чтения в блочном режиме передачи *Read Multiple* отличается от обычного (с обменом PIO) тем, что запросы прерывания вырабатываются не на каждый сектор, а на блок секторов, размер которого задан командой *Set Multiple Mode*. Кроме того, в случае появления неисправимой ошибки чтения содержимое блока командных регистров не определено (не указывает на сбойный сектор). За счет

сокращения числа прерываний, которые должен обслужить процессор, блочный режим в многозадачной системе позволяет повысить производительность на 30%. Производительность обмена зависит от размера блока. Размер, оптимальный для устройства, может не быть оптимальным для ОС. Команда «длинного» чтения *Read Long* считывает сектор данных вместе с контрольными байтами. При ее вызове регистр SC должен указывать на запрос только одного сектора. Блок данных считывается 16-битными словами, а контрольные байты — 8-битными. Некоторые устройства ATA-1 неспособны быстро передавать байты ECC вслед за данными. Для их считывания используют 8-битный обмен в режиме PIO Mode 0.

Команда верификации Read Verify Sector(s) в отличие от обычного чтения не передает данные от устройства. В случае ошибки на адрес сбойного сектора указывает блок командных регистров. Запрос прерывания выполняется после исполнения команды.

Команды записи секторов Write DMA, Write Sector(s), Write Long, Write Multiple работают аналогично соответствующим им командам чтения и также имеют версии с повторами и без. Команда записи с верификацией *Write Verify* аналогична команде *Write Sectors),* но для каждого сектора после записи выполняется контрольное считывание. Для логической инициализации (очистки области) дисков имеется команда записи *Write Same,* которая позволяет содержимое 512 байт, принятых от хоста, записать в группу секторов. Если в регистр свойств занесен код 22h, область записываемых секторов определяется регистрами командного блока. Если в регистр свойств занесен код DDh, запись производится во все доступные секторы устройства.

Команды чтения и записи буфера Read Buffer и *Write Buffer* служат для обмена в режиме PIO с 512-байтной буферной памятью устройства (но не сектором носителя).

Команда форматирования трека Format Track по входным параметрам специфична для каждого устройства, и ее использование не рекомендуется. Многие устройства ее отвергают как недопустимую. Поскольку команда форматирования традиционно ориентирована на один трек, форматирование дисков ATA в режиме трансляции геометрии, при котором внешняя геометрия не совпадает с реальной, невозможно. В накопителях с зонным форматом записи форматирование трека будет «накрывать» переменное число реальных треков. Для низкоуровневого форматирования используются команды, специфичные для конкретных устройств, причем они могут быть недопустимыми в нормальном (не технологическом) режиме работы устройства.

Вспомогательное назначение имеют *команды поиска.* По команде поиска *Seek* устройство устанавливает головки на заданный цилиндр/трек и считывает идентификатор сектора.

Команда рекалибровки *Recalibrate* заставляет устройство найти нулевой цилиндр. Если устройству это не удается, устанавливается бит ошибки *Track 0 Not Found.* Эту команду обычно применяют при обработке ошибок: часто после такого «встряхивания» ошибка не повторяется. После успешного выполнения команды блок командных регистров содержит адрес первого сектора диска в формате, зависящем от режима адресации (LBA или CHS).

5.2.1.4.2. Инициализация, идентификация и конфигурирование устройств

Устройства ATA воспринимают три вида *сброса* в исходное состояние:

- Сброс по включению питания .(power on reset) — выполняется самотестирование, запускается двигатель, проверяется механика, устанавливаются параметры умолчания, интерфейс и регистры сбрасываются в исходное состояние.
- Аппаратный сброс (hardware reset) по сигналу RESET# — выполняется самотестирование, устанавливаются параметры умолчания, интерфейс сбрасывается в исходное состояние.
- Программный сброс (software reset) по установке бита *SRST* регистра DC — интерфейс сбрасывается в исходное состояние.

Для устройств ATAPI, кроме этих трех видов сброса, предназначена команда *Device Reset,* по которой интерфейс сбрасывается в исходное состояние. Устройства ATA эту команду отвергают.

После любого сброса или выполнения команды диагностики устройство в блоке командных регистров содержит *сигнатуру,* определяющую его тип:

- SC=0lh, SN=01h, CL=00h, CH=00h, DH-00h - устройства ATA;
- SC=01h, SN=0lh, CL=14h, CH=EBh, DH=00h или 10h - устройства ATAPI (значение DH=10h будет после выполнения команды *Device Reset* устройством 1).

В системе команд имеются средства идентификации и управления свойствами устройств, включая возможность обновления встроенного ПО.

Команда идентификации Identify Device позволяет считать из контроллера блок из 256 слов, характеризующих устройство ATA (паспорт диска). Устройства с пакетным интерфейсом эту команду должны отвергать, для них имеется аналогичная команда *Identify Packet Device* (см. п. 5.2.1.8). Блок параметров может храниться как в энергонезависимой памяти устройства, так и на самом носителе в месте, недоступном для обычных обращений. Если устройство не может сообщить все параметры (блок

хранится на носителе, а мотор остановлен), на это укажет бит 2 слова 0; при этом должны быть действительны хотя бы слова 0 и 2. Состав полей блока для непакетных устройств приведен в табл. 5.10. Графа «F/V» определяет свойства соответствующих полей: F — содержимое фиксировано, V — содержимое меняется в зависимости от состояния устройства или выполненных команд, X — специфично для данного устройства (может быть и F, и V), R — зарезервированные поля. Для устройств со сменными носителями значения полей F могут меняться при смене носителя.

Числовые данные в блоке параметров передаются в порядке младший-старший, для 32-разрядных чисел в первом слове идут младшие 16 бит, во втором — старшие. Строковые значения (только печатные символы ASCII) передаются слева направо: первая буква — в старшем байте первого слова, вторая — в младшем, третья — в старшем байте второго слова и т. д. Если строка короче отведенного поля, она справа расширяется пробелами (код 20h).

Таблица 5.10. Информация идентификации устройства ATA

Слово	F/V	Назначение
0		Общая конфигурационная информация Бит 15: 0 для всех устройств ATA (1 — для ATAPI) Бит 7: 1 —устройство со сменным носителем Бит 6: 1 —устройство (контроллер) с несменным носителем Бит 2 — сообщение параметров не завершено, см. слово 2 (5+) Бит 0 — зарезервирован(=0) Остальные биты свободны (2+); вATA-1 определяли свойства накопителя (метод кодирования, особенности форматирования, скорость передачи и др.)
1	F	Число цилиндров в конфигурации по умолчанию
2	R	Специфическая конфигурация (5+): 37C8h — устройство требует для раскрутки мотора подкоманду SET FEATURES, сообщенный блок параметров неполный 738Ch — устройство требует для раскрутки мотора подкоманду SET FEATURES, сообщенный блок параметров полный 8C73h — устройство не требует для раскрутки мотора подкоманды SET FEATURES, сообщенный блок параметров неполный C837h — устройство не требует для раскрутки мотора подкоманды SET FEATURES, сообщенный блок параметров полный
3	F	Число головок в конфигурации по умолчанию
4,5	X	Специфично, освобождено в 4+
6	F	Число секторов на треке в конфигурации по умолчанию
7-9	X	Специфично, освобождено в 4+
10-19	F	Серийный номер (20 символов ASCII), если слово 10 ненулевое, иначе — специфичная информация
20,21	X	Специфично, освобождено в 4+. В ATA-1 описывался тип буфера (слово 20: 0 — неизвестный, 1 — односекторный, 2 — двухпортовый многосекторный, 3 — двухпортовый многосекторный с кэшированием чтения) и размер (слово 21, размер в блоках по 512 байт)
21	X	Специфично
22	F	Число дополнительных байтов в командах «длинного» чтения и записи (в ATA-4 объявлено устаревшим)
23-26	F	Версия встроенного ПО (8 символов ASCII), если слово 23 ненулевое, иначе — специфичная информация
27-46	F	Номер модели (40 символов ASCII), если слово 27 ненулевое, иначе — специфичная информация
47	X F	Биты 15-8 специфичны (в 4+ — 80h) Биты 7-0 — возможности множественных операций *(Read/Write Multiple):* 00 — зарезервирован, 01h-FFh — максимальное количество секторов, передаваемых за сеанс
48	R	Зарезервировано

Таблица 5.10. Информация идентификации устройства ATA. (Продолжение)

Слово	F/V	Назначение
49		Возможности:
	R	Биты 15-14— зарезервированы для команды *Identify Packet Device*
	F	Бит 13 — значения таймера режима *Standby:* 1 — задается в соответствии со стандартом, 0 — управляются устройством
	R	Бит 12 зарезервирован для команды *Identify Packet Device*
	F	Бит 11: 1— IORDY поддерживается (для PIO Mode 3 и выше — обязательно), 0 — IORDY может поддерживаться
	F	Бит 10: 1— сигнал IORDY может быть запрещен командой Set Features
	F	Бит 9 =1 (в ATA-4 используется для Identify Packet Device, в ATA-2 — как указание на поддержку LBA и действительности слов 60-61)
	F	Бит 8 =1 (в ATA-4 используется для Identify Packet Device, в ATA-2 — как указание на поддержку DMA)
	X	Биты 7-0 специфичны (освобождены в 4+)
50	F	Возможности (ATA-4): Бит 15=0 Бит 14=1 Биты 13-1 —зарезервированы Бит 0=1 указывает на специфичное значение минимума для Standby Timer
51	F	Биты 15-8: длительность цикла обмена в режиме PIO. Если значение параметра не соответствует режимам 0,1 или 2, используется PIO Mode 0. Устарело в 5+
	X	Биты 7-0 специфичны (освобождены в 4+)
52	F	В ATA-2 биты 15-8: временной режим цикла одиночного обмена DMA (Single word DMA). Если значение параметра не соответствует режимам 0,1 или 2, используется Mode 0. Если поддерживаются слова 62 или 63, значение поля игнорируется (освобождены в 4+)
	X	Биты 7-0 специфичны (освобождены в 4+)
53	R	Биты 15-3 зарезервированы
	F	Бит 2: 1— поля в слове 88 действительны, 0 — нет
	F	Бит 1: 1 — поля в словах 64-70 действительны, 0 — нет. Любое устройство, поддерживающее PIO Mode 3 и старше или Multiword DMA Mode 1 и выше, должно использовать эти поля
	V	Бит 0: 1 — поля в словах 54-58 действительны, 0 — поля могут быть действительны (5+ — недействительны)
54	V	Текущее число цилиндров. Для устройств ATA-1, если команда Initialize Device Parameters не применялась, значение слова специфично
55	V	Текущее число головок. Для устройств ATA-1, если команда Initialize Device Parameters не применялась, значение слова специфично
56	V	Текущее число секторов на трек. Для устройств ATA-1, если команда Initialize Device Parameters не применялась, значение слова специфично
57-58	V	Текущая емкость в секторах для режима CHS (произведение слов 54,55,56)
59	R	Биты 15-9 зарезервированы
	V	Бит 8: 1 —установка для многосекторной передачи действительна
	V	Биты 7-0: текущая установка максимального числа секторов для многосекторных команд
60-61	F	Общее число секторов, адресуемых пользователем в режиме LBA
62	V	В ATA-2 — режим одиночного обмена DMA (3+ — устарело): Биты 15-8 — активный режим: бит 8=1 — Mode 0, бит 9=1 — Mode 1 и т. д. Единичное значение может иметь только один бит.
	F	Биты 7-0 — поддерживаемые режимы: бит 0=1 - Mode 0, бит 1=1— Mode 1 и т. д.
63	V	Режим множественного обмена Multiword DMA: Биты 15-8: активный режим: бит 8=1 — Mode 0, бит 9= 1 — Mode 1 и бит 10=1 — Mode 2. Единичное значение может иметь только один бит, но только если в слове 88 биты 15-8 нулевые (активным может быть только один режим DMA).
	F	Биты 7-0: поддерживаемые режимы: бит 0=1 — Mode 0, бит 1=1— Mode 1 и 0, бит 2=1 — Mode 2,1 и 0

Таблица 5.10. Информация идентификации устройства ATA. (Продолжение)

Слово	F/V	Назначение
64	R F	Биты 15-8 зарезервированы Биты 7-0 — поддерживаемые прогрессивные режимы программного обмена (Advanced PIO): бит 0=1 — PIO Mode 3, бит 1=1 —PIO Mode 4, биты 2-7 — зарезервированы
65	F	Минимальная длительность цикла передачи множественного DMA (в наносекундах)
66	F	Рекомендованная длительность цикла передачи множественного DMA (в наносекундах). При многосекторной передаче обеспечивает оптимальную скорость, при которой устройство не будет тормозить поток снятием запроса DMARQ
67	F	Минимальная длительность цикла передачи PIO без использования сигнала готовности (в наносекундах)
68	F	Минимальная длительность цикла передачи PIO с использованием сигнала готовности IORDY (в наносекундах)
69-70	R	Зарезервировано для поддержки перекрытия команд и очередей
71-74	R	Зарезервировано для команды *Identify Packet Device*
75	F	Биты 15-5 — зарезервированы Биты 4-0 — максимальная глубина очереди команд — 1 (0 — для устройств, не поддерживающих очереди)
76-79	R	Зарезервировано
80	F	Главный номер версии интерфейса (если не 0000h или FFFFh), единичное значение каждого бита означает поддержку соответствующей версии: Бит 0 — резерв, Бит 1 — ATA-1, Бит 2 — ATA-2, Бит 3 — ATA-3, Бит 4 — ATA/ATAPI-4 и т.д.
81	F	Дополнительный номер версии интерфейса (если не 0000h или FFFFh). Для опубликованных (ANSI) версий: 0004h — ATA 2, 000Ah — ATA-3, 0012h — ATA/ATAPI-4. Последние номера: 00015h —ATA/ATAPI-5 rev. 1, 00017h - ATA/ATAPI-4 rev. 17, 00018h —ATA/ATAPI-6 rev.0.
82	F	Поддержка команд и свойств (если слова 82 и 83 не равны 0000h или FFFFh, 3+): Бит 15 — не используется Бит 14 — команды Nop Бит 13 — команды Read Buffer Бит 12 — команды Write Buffer Бит 11 — не используется Бит 10 — ограничение доступного пространства Бит 9 — команда Device Reset Бит 8 — вырабатывание прерывания во время выполнения команды Service Бит 7 — прерывание по освобождению шины Бит 6 — кэширование с упреждающим чтением Бит 5 — кэширование записи Бит 4 — команда Packet (для ATA должен быть 0) Бит 3 — управление энергопотреблением Бит 2 — смена носителя Бит 1 — команды Security Бит 0 — SMART
83	F	Поддержка команд и свойств (как и слово 82,3+): Бит 15= 0 Бит 14=1 Биты 13-9 зарезервированы Бит 8 — расширение SET MAX для защиты (5+) Бит 7 — резерв Бит 6 — специальная команда запуска двигателя после включения (5+) Бит 5 — Power-Up In Standby (5+) Бит 4 — уведомление о смене носителя БитЗ — расширенное управление энергопотреблением (APM) Бит 2 — команды CFA Бит 1 — команды Read/Write DMA Queued Бит 0 — команда Download Microcode

Таблица 5.10. Информация идентификации устройства ATA. (Продолжение)

Слово	F/V	Назначение
84	F	Поддержка команд и свойств (если слова 82, 83 и 84 не равны 0000h или FFFFh, 4+): Бит 15= 0 Бит 14=1 Биты 13-0 зарезервированы
85-87	F	*Разрешенные* команды и свойства; поля и правила аналогичны словам 82-84(4+)
88		Режим Ultra DMA (4+)
	R	Биты 15-13 зарезервированы
		Биты 8 -12: активный режим Ultra DMA: бит 8=1 — Mode 0, бит 9=1 — Mode 1 и т. д. Единичное значение может иметь только один бит, но только если в слове 63 биты 15-8 нулевые
	R	Биты 7-5 зарезервированы
	V	Биты 4-0: поддерживаемые режимы Ultra DMA: бит 0=1 — Mode 0, бит 1=1— Mode 1 и 0, и т. д.
89	F	Время защитного стирания (4+): 0 — не указано; значение 1-254, умноженное на 2, дает время в минутах, 255 — более 508 мин
90	F	Время расширенного защитного стирания (аналогично слову 89,4+)
91	V	Текущий уровень Advanced Power Management (4+)
92	V	Код номера главного пароля (Master Password Revision Code, 5+)
93	V	Результат аппаратного сброса (5+) Бит 15=0 Бит 14=1 Бит 13 — обнаруженный уровень CBLID#: 0 — ниже V_{IL}, 1 — выше V_{IH} Биты 12-8 — результаты сброса устройства 1 (в устройстве 0 обнулены) Бит 12 — резерв Бит 11=1 —устройство 1 ввело сигнал PDIAG#, 0 — нет Биты 10-9 — способ, которым устройство 1 узнало свой номер: 00 —резерв, 01 —по джамперу, 10—по CSEL, 11 —иным или неизвестным способом Бит 8 = 1 Биты 7-0 — результаты сброса устройства 0 (в устройстве 1 обнулены) Бит 7 — резерв Бит 6=1— устройство 0 ответило при выборе устройства 1,0— нет Бит 5=1— устройство 0 обнаружило сигнал DASP#, 0 — нет Бит 4=1— устройство 0 обнаружило сигнал PDIAG#, 0 — нет Бит 3=1— устройство 0 успешно прошло диагностику, 0 — нет Биты 2-1 — способ, которым устройство 0 узнало свой номер: 00 — резерв, 01 — по джамперу, 10 — по CSEL, 11 — иным или неизвестным способом Бит 0 = 1
94	V	Биты 15-8 — рекомендованное значение уровня управления акустическим шумом (6+) Биты 7-0 — текущее значение уровня управления акустическим шумом, FEh — максимальная производительность в ущерб тишине (6+)
95-97	V	Параметры качества сервиса для AV-команд
98-126	R	Зарезервировано
127	R	Биты 15-2 — зарезервированы
	F	Биты 1 -0 — уведомление о смене носителя (4+): 00 — не поддерживается, 01 — поддерживается, 1x — зарезервированы
128	V	Состояние защиты (3+): Биты 15-9 зарезервированы Бит 8 — уровень защиты: 0 — высокий, 1 — максимальный Биты 7-6 — зарезервированы Бит 5: 1— поддержка расширенного защитного стирания Бит 4: 1— счетчик попыток иссяк Бит 3: 1 — устройство заморожено Бит 2: 1 — устройство закрыто Бит 1: 1— защита установлена Бит 0: 1 — защита поддерживается
129-159	X	Специфично

Таблица 5.10. Информация идентификации устройства ATA. (Продолжение)

Слово	F/V	Назначение
160	R	Для устройств CFA, поддерживающих power mode *1:* Бит 15:1 — слово 160 поддерживается Бит 14—резерв Бит 13 — для выполнения одной или более команд устройства CFA требуется power mode 1 Бит 12 — CFA power mode 1 запрещен Биты 11 - 0 — максимальный ток, мА
161-175	R	Зарезервировано для CFA
160-254	R	Зарезервировано
255	F/V	Контроль целостности (5+): Биты 15-8 — контрольная сумма (сумма всех 512 байт по модулю 256 должна быть нулевой) Биты 7-0 — сигнатура A5h (признак использования контрольной суммы)

Команда установки параметров Initialize Device Parameters задает режим трансляции геометрии в системе CHS. Регистр SC указывает число секторов на трек, а в поле номера головки регистра *DH* указывается уменьшенное на единицу число головок. Если устройство не поддерживает запрошенный режим трансляции, оно выдаст ошибку «команда отвергнута» (старая спецификация ATA нечетко описывала эту ситуацию, и некоторые системы не обнаруживали отказ при данной команде). После запроса неподдерживаемого режима трансляции устройство обнуляет бит 0 слова 53 и блокирует доступ к носителю с ошибкой *ID Not Found* до запроса поддерживаемого режима трансляции. Устройство обязано поддерживать режим, описанный словами 1,3 и 6 блока его параметров. Некоторые устройства ATA-1 требовали, чтобы данная команда обязательно выполнялась до первого доступа к носителю.

Команда установки свойств Set Features (специфичная для устройства) подразумевает помещение кода подкоманды (табл. 5.11) в регистр свойств.

Таблица 5.11. Подкоманды задания свойств устройства

Код	История	Назначение
01h		Разрешение 8-битного обмена (в 5+ только для CFA)
02h		Разрешение кэширования записи *t*
03h		Установка режима передачи по значению регистра SC (табл. 5.12)
04h	У4	Разрешение автоматического переназначения дефектных блоков
05h		Разрешение расширенного управления энергопотреблением APM (уровень в регистре SC)
06h	5+	Разрешение включения в режиме Standby (действует и после любых сбросов)
07h	5+	Запуск двигателя при включении в режиме *Standby*
09h	5+	Резерв для смещения загрузочной области
0Ah	6+	Разрешение CFA Power Mode 1 (устройство может потреблять до 500 мА от источника 3,3 или 5 В)
31h		Запрет уведомления о смене носителя
33h	У4	Запрет повторов
40h	6+	Разрешение AV-команд
42h	6+	Разрешение управления акустическим шумом, в SC — уровень
44h	У4	Установка количества дополнительных байтов в командах Read Long/Write Long
54h	У4	Установка количества.сегментов кэша по значению регистра SC
55h		Запрет упреждающего чтения
5Dh		Разрешение прерывания по освобождению шины
5Eh		Разрешение прерывания при выполнении команды *Service*
66h		Запрет возврата к параметрам по умолчанию при включении питания
77h	У4	Запрет ECC
81h		Запрет 8-битного обмена (в 5+ только для CFA)
82h		Запрет кэширования записи
84h	У4	Запрет автоматического переназначения дефектных блоков
85h		Запрет расширенного управления энергопотреблением
86h	5+	Запрет включения в режиме *Standby*
88h	У4	Разрешение ECC
89h	6+	Резерв для смещения загрузочной области

Таблица 5.11. Подкоманды задания свойств устройства (Продолжение)

Код	История	Назначение
8Ah	6+	Запрет CFA Power Mode 1 (переход в Power Mode 0, при котором устройство может потреблять до 75 мА от источника 3,3В или до 100 мА от 5 В. В этом режиме команды чтения, записи и стирания могут и не выполняться)
95h		Разрешение уведомления о смене носителя
99h	У4	Разрешение повторов
9Ah	У4	Ограничение тока потребления значением в регистре SC, умноженным на 4(мА)
AAh		Разрешение упреждающего чтения
ABh	У4	Установка максимального количества кэшируемых блоков по регистру SC
BBh	У4	Установка 4-байтной длины дополнительного поля для команд Read Long/Write Long
C0h	6+	Запрет AV-команд
C2h	6+	Запрет управления акустическим шумом
CCh		Разрешение возврата к параметрам по умолчанию при включении питания
DDh		Запрет прерывания по освобождению шины
DEh		Запрет прерывания при выполнении команды Service
F0-FFh		Резерв для CFA

После включения питания или аппаратного сброса установленные свойства заменяются на принятые по умолчанию. Результат подкоманд можно и закрепить, подав подкоманду 66h (отмена — подкоманда с кодом CCh).

Таблица 5.12. Управление режимом обмена

Биты [7:0]	Режим обмена
00000 000	Режим PIO, принятый для устройства по умолчанию
00000 001	Режим PIO, принятый для устройства по умолчанию; запрет IORDY
00001 nnn	Режим PIO Mode x с сигналом IORDY
00010 nnn	Режим одиночного DMA Mode x[1]
00100 nnn	Режим множественного DMA Mode x
01000 nnn	Режим Ultra DMA Mode x
10000 nnn	Зарезервированы

[1] x определяется значением поля nnn.

Команда задания параметров блочного режима передачи Set Multiple Mode через регистр SC указывает число секторов, передаваемых с одним запросом прерывания. Значению SC=0 соответствует запрет блочного режима. После включения питания или аппаратного сброса блочный режим запрещен.

Команда диагностики Execute Device Diagnostic, адресуясь всегда к ведущему устройству, выполняется одновременно обоими устройствами. О ее результате ведомое устройство сообщает ведущему (сигналом PDIAG#). Состояние обоих устройств определяется по диагностическому коду (табл. 5.13), который считывается из битов [6:0] регистра ошибок ведущего устройства.

Фиктивная команда Nop, не изменяя содержимого регистров, позволяет считать информацию о состоянии устройства, выполнив всего одну 16-битную запись по адресу в регистре *D/H.* Команда всегда возвращает бит ошибки «команда отвергнута». В ATA-4 команда объявлена обязательной для устройств ATAPI и всех, поддерживающих перекрывающиеся команды. Она имеет подкоманды, указываемые в регистре свойств. Подкоманда с кодом 00 сбрасывает всю оставшуюся очередь, а с кодом 01 *(NOP Auto Poll)* на очередь не влияет.

Таблица 5.13. Диагностические коды.

Код	Ведущее устройство	Ведомое
01h	Нормально	Нормально или отсутствует
00h, 02h-7Fh	Неисправно	Нормально или отсутствует >
81h	Нормально	Неисправно
80h, 82h-FFh	Неисправно	Неисправно

Команда загрузки микрокода Download Microcode позволяет модифицировать firmware — встроенное ПО устройства. В зависимости от кода в регистре свойств загруженный микрокод будет действовать до выключения питания (FR=01h) или постоянно (FR=07h). Количество загруженных блоков задается регистром SN (старший байт) и SC (младший байт). Таким образом может быть загружено от 0

до 33 553 920 байт, что более чем достаточно для встроенного ПО. Загрузка некорректного микрокода может привести к выходу устройства из строя!

Устройства могут поддерживать и *нестандартные команды,* назначение которых определяется разработчиком устройств. Нестандартные команды позволяют оперировать служебной информацией дисков и управлять конфигурацией и свойствами устройств. Например, для винчестеров Quantum Fireball EX имеется *расширенная командам* (код F0h), которая обеспечивает чтение (но не запись!) списка дефектных блоков, а также чтение и запись конфигурационной информации (отличные от *Set Features).*

До выполнения подкоманды чтения списка дефектных блоков *Read Defect List* нужно определить длину списка с помощью команды *Read Defect List Length,* которая поместит в SC младший байт, а в *SN* старший байт длины. Перед подачей команды в командные регистры заносят определенные значения: SC=0 (код подкоманды), SN=CL=FFh и CH=3Fh («пароль»), D/H=Axh для устройства 0 D/H=Bxh для устройства 1, CR=F0h (код расширенной команды). После этой команды в блоке регистров окажутся значения, необходимые для чтения самого списка, и для чтения списка (предварительно прочитав его длину в регистрах SC и SN) остается только еще раз записать в командный регистр код расширенной команды F0h и принять поток данных обычным способом.

Список дефектов начинается с двух байтов заголовка (00h, 1Dh), за которыми следует двухбайтное поле длины списка (число записей, умноженное на 8, начиная со старшего байта), а за ними собственно записи дефектов (по 8 байт). Каждая запись в списке дефектов начинается с 3-байтного номера цилиндра (начиная со старшего байта), затем идет номер дефектной головки (1 байт) и 4 байта номера дефектного сектора. Если все байты номера сектора равны FFh, это означает дефектный цилиндр.

Подкоманда *READ CONFIGURATION* (код подкоманды 0lh) позволяет считать 512-байтный блок, описывающий текущую конфигурацию устройства: режимы кэширования, режимы коррекции ошибок, разрешение перемещения дефектных секторов при чтении и записи и т. п. Подкоманда *SET CONFIGURATION* (код подкоманды FFh) позволяет установить эти параметры, передав устройству такой же блок. Подкоманда *SET CONFIGURATION WITHOUT SAVING TO DISK* (код подкоманды FFh) делает то же, но без записи установок на диск (после сброса или выключения питания изменения будут отменены). Код подкоманды помещают в регистр *SC;* SN=CL=FFh и CH=3Fh («пароль»); D/H=Axh для устройства 0, D/H=Bxh для устройства 1; CR=F0h. В блоке параметров определены следующие биты и байты (остальные — нулевые):

- Байты 0-31 — конфигурационный ключ, строка «QUANTUM CONFIGURATION», дополненная справа пробелами. Если ключ не совпадет, команда будет отвергнута.
- Бит 32.1(первый бит тридцать второго байта) — *PE* (Prefetch Enable), разрешение предвыборки в кэш. По умолчанию 1.
- Бит 32.0 — *CE* (Cache Enable), разрешение кэширования чтения. По умолчанию 1.
- Бит 36.7 — *AWRE* (Automatic Write Reallocation Enabled), разрешение автоматического переназначения сбойного блока при записи. По умолчанию 1.
- Бит 36.6 — *ARR* (Automatic Read Reallocation), разрешение автоматического переназначения сбойного блока при чтении. Если RC=1, бит игнорируется. По умолчанию 1.
- Бит 36.4 —*RC* (Read Continuous), разрешение непрерывного чтения блока заказанной длины без задержек на процедуры восстановления. Позволяет повысить скорость в ущерб достоверности (ошибки чтения не сообщаются). По умолчанию 0.
- Бит 36.3 — *EEC* (Enable Early Correction), разрешение ранней коррекции («на лету», без повторного считывания), По умолчанию запрещено (0), то есть при ошибке чтения сначала выполняется повторное считывание, а потом коррекция. Если бит RC=1, EEC игнорируется.
- Бит 36.0 — *DCR* (Disable Correction), запрет коррекции, даже если ошибки исправимые.
- Байт 37 — число повторных считываний, выполняемых до попытки коррекции (по умолчанию 8).
- Байт 38 — *ECC CORRECTION SPAN,* максимальное число битов в группе, которые могут быть исправлены четырехкратным пакетным ECC (quadruple-burst ECC). По умолчанию 32 (20h).
- Бит 39.2 — *WCE* (Write Cache Enable), разрешение кэширования записи. По умолчанию 1.
- Бит 39.1 — *RUEE* (Reallocate Uncorrectable Error Enables), разрешение переназначения при неисправимых ошибках (при ARR=1). По умолчанию 1.

К сожалению, привести информацию по специфическим командам разных моделей накопителей нет возможности, они в общедоступной документации не описаны. Команды для данной модели взяты из документа «Quantum Fireball EX 3.2/ 5.1/6.4/10.2/12.7 GB AT. Product Manual».

5.2.1..4.3. Работа со сменными носителями

Для накопителей со сменными носителями требуются специальные меры, предотвращающие:

- потерю данных при смене носителя (страхуют от попытки записи по каталогу прежнего носителя);

- потерю данных при изъятии носителя, когда кэш записи еще не выгружен;
- несанкционированную смену носителя.

Для накопителей со сменными носителями в ATA-2 были предназначены команды загрузки и выгрузки, подтверждения смены носителя, блокировки и разблокировки устройства — *Boot - Post-Boot, Boot - Pre-Boot, Media Eject, Acknowledge Media Change, Door Lock, Door Unlock.* Их реализация специфична для каждой модели устройства.

В ATA-4 принят иной набор команд для сменных носителей. Здесь может работать один из двух механизмов (наборов свойств, features set) защиты от несанкционированной смены носителя, названия которых трудно поддаются легко различимому переводу:

- *Removable Media Status Notification* — уведомление о состоянии сменного носителя, применимо к устройствам и ATA, и ATAPI.
- *Removable Media* — сменный носитель, только для непакетных устройств.

В любой момент в устройстве может быть разрешен только один из этих механизмов.

Механизм уведомления является предпочтительным, он дает хосту возможность полного управления носителем. Нажатие кнопки на накопителе смены носителя не вызывает немедленной реакции устройства. Этот факт и состояние носителя может определяться по команде *Get Media Status.* Хост должен периодически выдавать эту команду и читать регистр *ER* (если хоть один бит установлен, то в регистре *SR* устанавливается бит *ERR).*

Назначение битов регистра *ER* после команды *Get Media Status:*

- Бит 1 — *NM* (No Media) — носитель отсутствует.
- Бит 2 — *ABRT* (Abort) — команда отвергнута (не поддерживается или не может быть завершена).
- Бит 3 — *MCR* (Media Change Request) — запрос смены носителя: устройство обнаружило нажатие на кнопку смены носителя. Бит устанавливается при нажатии, сбрасывается после выполнения этой команды.
- Бит 5 — *MC* (Media Changed) — носитель сменился. Бит устанавливается, когда устройство обнаружит вставленный новый носитель, сбрасывается после выполнения этой команды.
- Бит 6 — *WP* (Write Protected) — носитель защищен от записи.

Смену носителя для устройств ATA вызывает только команда *Media Eject* (для устройств ATAPI предназначена пакетная команда *Start/Stop Unit).*

Назначение битов регистра *ER* после команды *Media Eject:*

- Бит 1 — *NM* (No Media) — носитель отсутствует.
- Бит 2 — *ABRT* (Abort) — команда отвергнута (не поддерживается или не может быть завершена).

Включается механизм уведомления по команде *Set Features* подкомандой 95h, после выполнения которой в регистре *CL* оказывается номер версии механизма уведомления (пока только 00), а в регистр *CH* — дополнительная информация:

- Бит 0 — *PENA* (Prior Enabled) — механизм уже был включен на момент подачи команды.
- Бит 1 — *Lock* — устройство позволяет блокировать носитель.
- Бит 2 — *PEJ* (Physical Eject) — устройство имеет привод для смены носителя.

Механизм *выключается* по любому сбросу и команде диагностики устройства, а также по команде *Set Features* подкомандой 31h. При этом носитель остается незаблокированным, а биты состояния носителя и запроса его смены сбрасываются. Возможность и включение механизма уведомления определяется битом 4 слов 83 и 86 идентификатора устройства.

Альтернативный механизм смены носителя (Removable Media Feature Set) позволяет устройству непосредственно отрабатывать нажатие кнопки смены носителя (если носитель не заблокирован), менять носитель и определять его состояние. Смена носителя блокируется командой *Media Lock* и разрешается командой *Media Unlock,* эти же команды используются для определения состояния чтением регистра *ER* (если хоть один бит установлен, то в регистре *SR* устанавливается бит *ERR).* Если включен механизм уведомления, то эти команды не вызывают никаких действий (но и не отвергаются). Команда смены носителя *Media Eject* работает так же, как описано выше.

Назначение битов регистра *ER* после команд *Media Lock* и *Media Unlock:*

- Бит 1 — *NM* (No Media) — носитель отсутствует.

- Бит 2 — *ABRT* (Abort) — команда отвергнута (не поддерживается или не может быть завершена).
- Бит 3 — *MCR* (Media Change Request) — запрос смены носителя (только для команды *Media Lock):* устройство обнаружило нажатие на кнопку смены носителя. Бит устанавливается при нажатии, сбрасывается после разблокирования носителя.

По любому сбросу и команде диагностики устройства носитель разблокируется и бит запроса его смены сбрасывается. Возможность и включение механизма смены носителя определяется битом 2 слов 82 и 85 идентификатора устройства.

5.2.1.4.4. Команды для флэш-памяти

Для запоминающих устройств на флэш-памяти в ATA-4 ввели команды, начинающиеся с аббревиатуры CFA (Compact Flash Association — ассоциация производителей компактных флэш-карт). Специфика этих устройств заключается в записи: запись обеспечивается лишь в предварительно стертые ячейки (сектор), хотя есть устройства, автоматически осуществляющие стирание при записи. Операция записи выполняется существенно медленнее, чем чтения, скорость которого приближается к скорости динамической памяти. Операция стирания занимает еще больше времени. Поэтому ввели команды записи в предварительно стертые секторы — *CFA Write Without Erase* и *CFA Write Multiple Without Erase,* применение которых позволяет повысить производительность обмена с устройствами. Для стирания блоков предназначена команда *CFA Erase Sectors,* в которой указывается начальный адрес и количество стираемых секторов. Интерес представляет информация о состоянии сектора: стертый или нет и сколько раз выполнялась запись. Эту информацию можно получить по команде *CFA Translate Sector* — по ней в режиме PIO устройство выдаст блок данных (512 байт), описывающий сектор, адресованный командными регистрами. Структура блока следующая;

- 00h — Старший байт номера цилиндра;
- 01h — Младший байт номера цилиндра;
- 02h — Номер головки;
- 03h — Номер сектора;
- 04h — LBA биты 23:16;
- 05h — LBA биты 15:8;
- 06h — LBA биты 7:0;
- 07-12h — Резерв;
- 13h — Флаг состояния сектора: FF — стерт; 00 — не стерт;
- 14- 17h— Резерв;
- 18h — Счетчик циклов записей в сектор, биты 23:16;
- 19h — Счетчик циклов записей в сектор, биты 15:8;
- lAh — Счетчик циклов записей в сектор, биты 7:0;
- 1B - 1FFh — Резерв.

Поскольку организация флэш-памяти сильно отличается от традиционных устройств с подвижными носителями, сообщения об ошибках этих устройств не вписываются в стандартные определения битов регистра *ER.* В случае ошибки при выполнении команды в регистре *SR* устанавливается бит *ERR,* а причину ошибки можно уточнить командой *CFA Request Extended Error,* помещающей в регистр *ER* расширенный код ошибки.

Устройства CFA могут иметь 8-разрядную внутреннюю организацию, и для разрешения 8-битного обмена в режиме PIO служат соответствующие подкоманды *SET FEATURES.* Обмен в режиме DMA для данного класса устройств необязателен. Устройства, поддерживающие команды CFA, сообщают значение 848Ah в слове 0 идентификатора устройства или единицу во 2 бите слова 83.

5.2.1.4.5. Управление энергопотреблением и шумом

Средства управления энергопотреблением — Power Management — были заложены еще в первой спецификации ATA, в ATA/ATAPI-4 они уже считаются обязательными. Различают следующие состояния, перечисленные в порядке возрастания энергопотребления:

Sleep — «заснувшее» устройство потребляет минимум энергии, команды не воспринимаются, «разбудить» его может только сброс. Время «пробуждения» — не более 30 с.

Standby Mode (дежурный режим) — устройство способно принимать команду по интерфейсу, но для доступа к носителю может потребоваться столь же большое время. В это состояние устройство может перейти как по команде, так и по таймеру (Standby Timer), отсчитывающему время от последнего запроса, полученного в состоянии ожидания или активном состоянии. Время срабатывания таймера программируется; он может быть запрещен.

Idle Mode (состояние ожидания) — устройство способно сразу начать обслуживание обращения к

носителю, но не слишком быстро, так как некоторые узлы отключены.

Active Mode (активный режим) — устройство обслуживает все запросы за кратчайшее время.

Команда проверки режима энергопотребления *Check Power Mode* через регистр SC возвращает состояние: SC=00h — устройство находится в состоянии *Standby* или переходит в него, SC=80h — устройство в состоянии ожидания *(Idle),* SC=FFh — устройство активно (значение 80h определено начиная с ATA-3 — прежде и в состоянии *Idle* возвращалось значение FFh).

Команда *Idle* переводит устройство в режим ожидания из активного или дежурного режима, а также через регистр SC программирует таймер дежурного режима (табл. 5.14).

Команда *Idle Immediate* — непосредственный перевод в режим ожидания — таймер не программирует.

Таблица 5.14. Тайм-аут перехода в дежурный режим

Содержимое SC	Тайм-аут
0(00h)	Тайм-аут запрещен
x=1-240(01h-F0h)	(x×5)с
x=241-251(F1h-FBh)	((x-240)×30) мин
252 (FCh)	21 мин
253 (FDh)	Период, заданный производителем (в пределах 8-12 ч)
254 (FEh)	Зарезервировано
255 (FFh)	21 мин 15с

Команда *Sleep* является единственным способом перевода устройства в спящий режим. После получения команды устройство генерирует запрос прерывания. Хост должен считать регистр состояния, что сбросит прерывание и позволит устройству «заснуть». «Разбудить» устройство можно только аппаратным или программным сбросом. Состояние, в которое оно перейдет, определяется типом сброса и программируется. Поскольку не каждый хост «знает», что после команды нужно прочитать регистр состояния, устройство автоматически сбросит запрос прерывания и «заснет» через определенный период времени (не менее 2 с).

Команда *Standby* переводит устройство в дежурный режим потребления и через параметр в регистре SC программирует таймер. Команда *Standby Immediate* таймер не программирует и переводит устройство в этот режим немедленно.

В ATA/ATAPI-5+ ввели возможность перехода в состояние *Standby* по включению питания — *Power-Up In Standby,* включаемую подкомандой *Set Features.* При этом снижается пиковая нагрузка на блок питания в момент включения. Устройство может также поддерживать и специальную подкоманду *Set Features* для раскрутки шпинделя. Если эта подкоманда поддерживается, то ее необходимо выполнить до обращения к носителю; если не поддерживается — запуск произойдет по первой же команде, требующей обращения к носителю. На поддержку и разрешенность этих свойств указывают биты 5, 6 слов 83 и 86.

Устройства могут поддерживать *расширенное управление энергопотреблением* — APM (Advanced Power Management). При этом задается уровень *APM Level,* определяющий степень активности: 01h — минимальное потребление, FEh — максимальная производительность. Уровень выше 80h не позволяет устройству останавливать двигатель привода носителя (переходить в *Standby).* APM управляется подкомандами *Set Features:* подкомандой 05h APM разрешается, а через регистр SC задается уровень. Устройства с пакетным интерфейсом для управления энергопотреблением могут использовать также команды пакетного протокола.

В ATA/ATAPI-6 появилась новая возможность — управление уровнем акустического шума, издаваемого устройством при работе, *Automatic Acoustic Management.* Для этого вводятся специальные подкоманды *Set Features,* задающие уровень управления шумом (acoustic management level) в диапазоне от 0 (минимальный шум и производительность) до FFh (максимальная производительность без заботы о шуме). Управление необязательно должно быть плавным — может быть всего несколько различимых градаций, и один и тот же способ снижения шума может применяться для ряда соседних значений уровня.

5.2.1.4.6. Защита данных

Начиная с ATA-3 в стандарт введена *группа команд защиты,* ее поддержка определяется по команде идентификации словом 128. Если защита поддерживается, то устройство должно отрабатывать все команды группы Security. С точки зрения защиты, устройство может находиться в одном из трех состояний:

- *открыто* (unlocked) — устройство выполняет все свойственные ему команды. Устройство с установленной защитой можно открыть только командой *Security Unlock,* в которой передается блок данных, содержащий правильный пароль. Для осложнения подбора пароля (его длина составляет 32 байт) служит внутренний счетчик неудачных попыток открывания, по срабатывании которого

команды открывания будут отвергаться до выключения питания или аппаратного сброса;

- *закрыто* (locked) — устройство отвергает все команды, связанные с передачей данных и сменой носителя. Допустимы лишь команды общего управления, мониторинга состояния и управления энергопотреблением. Из команд защиты допустимы лишь команды стирания *(Security Erase)* и открывания *(Security Unlock).* В это состояние устройство с установленной защитой входит каждый раз по включению питания;
- *заморожено* (frozen) — устройство отвергает все команды управления защитой, но выполняет все остальные. В это состояние устройство переводится командой *Security Freeze Lock* или автоматически по срабатыванию счетчика попыток открывания устройства с неправильным паролем. Из этого состояния устройство может выйти только по аппаратному сбросу или при следующем включении питания. Срабатывание счетчика попыток отражается установкой бита 4 *(EXPIRE)* слова 128 блока параметров, бит сбросится по следующему включению питания или по аппаратному сбросу.

Устройство выпускается производителем с не установленной защитой, то есть по включению оно будет открыто. Система защиты поддерживает два пароля — *главный* (master password) и *пользовательский* (user password) — и два уровня защиты — высокий (high) и максимальный (maximum). При *высоком уровне* защиты устройство можно открывать любым из двух паролей. При *максимальном уровне* устройство открывается только пользовательским паролем, а по главному паролю доступна только команда стирания, при этом *вся информация с носителя будет стерта.*

Для *установки защиты и пароля* используется команда *Security Set Password,* которую выполняет только открытое устройство. При установке пароля пользователя устанавливается и защита, при установке главного пароля состояние и уровень защиты не меняется. В передаваемом блоке данных (512 байт = 256 слов) *нулевое слово* является управляющим:

- бит 0 определяет тип пароля (0 — *User,* 1 — *Master);*
- бит 8 при установке пользовательского пароля определяет степень защиты (0 — *High,* 1 — *Maximum).*

Слова 1-16 содержат пароль; слово 17 — код номера главного пароля (Master Password Revision Code), требуется при установке главного пароля; остальные слова игнорируются.

Главный пароль устанавливается изготовителем или продавцом устройства, впоследствии он может быть переопределен пользователем. Код номера главного пароля (master password revision code) сообщается словом 92 в блоке параметров устройства, он может быть в диапазоне 1 - FFFEh. Значения 0000 или FFFFh указывают на то, что код номера пароля не поддерживается устройством. Изготовитель выпускает устройство с номером пароля FFFEh.

Снять защиту можно командой *Security Disable Password* (предъявив один из двух паролей), которую выполняет только открытое устройство. После снятия защиты устройство не будет переходить в состояние *закрыто,* пароль пользователя перестает действовать, но главный пароль сохраняется (он будет снова активирован при последующем задании пароля пользователя).

Для предотвращения несанкционированных попыток смены пароля или уровня защиты служит команда *Security Freeze Lock,* после которой в устройстве «замораживается» состояние защиты.

Команда защитного стирания Security Erase UNIT подразумевает передачу устройству блока данных в 256 слов. Нулевое слово является управляющим, в нем определены только 2 бита:

- бит 0 определяет тип пароля (0 — *User,* 1 — *Master);*
- бит 1 определяет уровень защитного стирания (0 — обычное, 1 — расширенное).

Слова 1-16 содержат пароль, остальные игнорируются.

При *обычном стирании* устройство заполняет двоичными нулями всю область данных, видимую пользователем. При *расширенном стирании* заполняется весь носитель, включая и области, ранее переназначенные устройством (по угрозе неисправимых ошибок). Кроме того, вместо двоичных нулей используется образец данных, заданный производителем (им заполняются новые устройства). После стирания защита снимается, но главный пароль сохраняется. Чтобы застраховаться от случайного стирания, непосредственно перед этой командой должна быть выполнена команда *Security Erase Prepare,* иначе команда стирания будет отвергнута устройством.

Команды защиты в ATA-4 дополнены ограничением максимального адреса, доступного пользователю (сообщаемого в блоке параметров идентификации), командой *Set Max Address.* Узнать реальный максимальный адрес позволяет команда *Read Native Address.* В ATA/ATAP1-5+ команды Set Max расширены (введено их паролирование) за счет введения подкоманд, задаваемых через *FR.* Идея паролирования отличается от общей защиты: после включения питания можно ограничить максимальный адрес (но только один раз) или же ограничить доступ к этой команде. Все эти команды неприменимы для устройств со сменными носителями. Определены следующие подкоманды:

- 00 — *Set Max Address,* ограничение размера доступной области. В регистрах *CH, CL, D/H* и *SN* задается адрес последнего доступного сектора (в режиме *CHS* или *LBA),* в регистре SC бит 0 (VV — Value Volatile) определяет, сохранится ли установленное значение после выключения питания (1 — да, 0 — нет). Непосредственно перед данной командой должна быть успешно выполнена команда *Read Native Address.* Результат команды отражается в словах 1, 54, 57, 60 и 61 идентификатора устройства.
- 01 — Set *Max Set Password,* задание пароля для команд *Set Max.* Пароль содержится в словах 1-16 блока данных (512 байт), передаваемых в режиме PIO (слова 0 и 17-255 не используются). Пароль сохраняется и после последующего включения питания.
- 02 — *Set Max Lock,* включение парольной защиты для команд *Set Max* — перевод в состояние *Set_Mox_Locked,* в котором из команд *Set Max* исполняются только *Set Max Unlock* и *Set Max Freeze Lock.* Состояние сохраняется до выключения питания или успешного выполнения этих команд.
- 03 — Set *Max Unlock,* ввод пароля (см. выше), по которому устройство переводится в состояние *Set_Max_Unlocked,* в котором все команды Set *Max* доступны. Если пароль неправильный, то декрементируется счетчик попыток (по включению устанавливается 5 попыток), а по его обнулению устройство перестает воспринимать эти команды до следующего включения питания.
- 04 — Set *Max Freeze Lock,* «замораживание» команд Set *Max.* После нее все вышеприведенные команды будут отвергаться до повторного включения питания.

Любая подкоманда Set *Max,* введенная сразу после чтения настоящего максимального адреса, будет трактоваться устройством как Set *Max Address.*

5.2.1.4.7. Мониторинг состояния — SMART

Для предупреждения о возможном отказе устройства служит технология S.M.A.R.T. (Self-Monitoring, Analysis and Reporting Technology — технология самонаблюдения, анализа и сообщений). Предсказуемые отказы (predictable failure) появляются в результате выхода параметров за некоторый порог. Отслеживаемые параметры: время разгона до заданной скорости, время позиционирования, процент ошибок позиционирования, «высота полета» головок, производительность (зависящая от числа повторов), количество использованных резервных секторов и др. Мониторинг может осуществляться двояко: в рабочем режиме *(on-line)* он осуществляется одновременно с выполнением команд хоста (при возможном некотором замедлении), мониторинг *off-line* выполняется устройством в паузе между «полезными» командами, не снижая производительности. Если во время выполнения этой процедуры придет внешняя команда, то мониторинг прервется на время исполнения команды, но начало исполнения команды может задержаться на время до двух секунд. Значения атрибутов, за которыми ведется наблюдение, сохраняются в служебной области носителя.

Для непакетных устройств имеется команда *SMART* (пакетные используют для этих целей собственный протокол), подкоманды которой задаются через регистр свойств *FR.* Перед подачей команд в регистры *CL* и *CH* заносятся константы: CL=4Fh, CH=C2h. Из подкоманд *SMART* стандартизованы следующие:

- *SMART Read Data* (FR=D0h) — чтение блока данных SMART. Блок данных (512 байт) имеет стандартную структуру, в нем содержатся следующие сведения:
  - состояние мониторинга off-line: запускался ли, завершался ли успешно, прерывался ли и почему;
  - состояние самотестирования (чем кончился предыдущий тест, тестирование выполняется в данный момент);
  - время до завершения тестирования off-line;
  - возможности тестирования off-line, поддержка журнала регистрации ошибок;
  - рекомендуемое время, через которое имеет смысл запрашивать результат после запуска коротких и расширенных тестов;
  - информация, специфичная для производителя.
- *SMART Enable/Disable Attribute Autosave* (FR=D2h) — управление автосохранением атрибутов: SC=00h — запрет, SC=Flh — разрешение автосохранения. Подкоманда необязательная. Автосохранение в энергонезависимую память выполняется по наступлению какого-то события (зависит от разработчика).
- *SMART Save Attribute Values* (FR=D3h) — сохранение значений атрибутов в энергонезависимой памяти устройства.
- *SMART Execute Off-Line Immediate* (FR=D4h) — немедленное выполнение мониторинга off-line в режиме *off-line* или *captive.* В режиме off-line устройство, получив команду, сразу выполняет действия, обычные при завершении команд (снять флаг занятости и т. п.). В режиме captive команда выполняется как обычная, то есть устройство устанавливает флаг занятости на время ее выполнения;

по окончании в регистрах сообщается результат и снимается флаг занятости. Функции задаются в регистре *SN:*

- 0 — выполнение программы SMART off-line в режиме *off-line;*
- 1 — выполнение короткого самотестирования SMART в режиме *off-line;*
- 2 — выполнение расширенного самотестирования SMART в режиме *off-line;*
- 127 — прекращение самотестирования SMART;
- 128 — выполнение короткого самотестирования SMART в режиме *off-line;*
- 129 — выполнение короткого самотестирования SMART в режиме *captive;*
- 130 — выполнение расширенного самотестирования SMART в режиме *captive.*

Для подкоманд, определяющих режим *captive,* в случае ошибок после их выполнения регистры *CL* и *CH* позволяют уточнить ошибку: если CL=F4h, CH=2Ch, то не прошло самотестирование, если CL=4Fh, CH=C2h, то ошибка иная (неподдерживаемые функции, ошибка задания параметров и т. п.).

- *SMART Read Log* (FR=D5h) — чтение журнала ошибок (5+). В регистре SC указывается число передаваемых секторов, в регистре *SN* — адрес журнала. Журналов может быть несколько, имеется и каталог журналов. В спецификации задана структура каталога и некоторых журналов.
- *SMART Write Log* (FR=D6h) — запись в журнал ошибок или в каталог журналов (5+).
- *SMART Enable Operations* (FR=D8h) — разрешение команд и внутренних функций SMART.
- *SMART Disable Operations* (FR=D9h) — запрет команд и функций SMART, все команды SMART, кроме *Smart Enable Operations,* будут отвергаться. Состояние (разрешены или нет функции SMART) сохраняется и после последующих включений питания.
- *SMART Return Status* (FR=DAh) — опрос результатов мониторинга. Если какой-либо порог перейден, регистры CH и *CL* будут иметь значения C2h и 4Fh соответственно, если все в порядке — 2Ch и F4h.

5.2.1.4.8. Потоковое расширение команд ATA

Традиционно главным качеством устройств хранения данных является способность достоверного хранения данных, а производительность является вторичной, хотя ее всячески и стремятся повышать. Если при чтении (записи) блока данных происходит ошибка, то устройство автоматически выполняет серию повторных обращений к носителю, в результате чего чаще всего удается получить достоверный результат. Однако при этом возникает непрогнозируемая задержка доставки данных, которая крайне нежелательна для мультимедийных приложений, связанных с записью или чтением аудио- и видеоданных в реальном времени. В спецификацию ATA/ATAPI-6 предполагается ввести потоковое расширение системы команд (streaming AV commands). В нем дается определение *нормальных данных ND* (Normal Data), для которых критична достоверность, и *аудио/видеоданных AV* (Audio Visual Data), критичных к времени доставки. В командах чтения *Read AV* (код C9h) и записи *Write AV* (CBh) указывается допустимое время их выполнения. Если за это время устройству не удается выполнить команду без ошибок, оно должно выдать наилучший возможный результат и указать на область данных, содержащую неисправимые ошибки. Потоковое расширение разрешается/запрещается подкомандами *Set Features,* при этом устанавливаются параметры обмена (требуемая пропускная способность в длительном режиме, максимальная задержка), в которых требования хоста согласуются с возможностями устройства.

5.2.1.5. Протоколы взаимодействия хоста и устройства

Протокол взаимодействия хоста с устройством выглядит следующим образом:

1. Хост читает регистр состояния устройства, дожидаясь нулевого значения бита *BSY.* Если присутствуют два устройства, хост обращается к ним «наугад» — состояние будет сообщаться последним выбранным устройством.

2. Дождавшись освобождения устройства, хост записывает в регистр *D/H* байт, у которого бит *DEV* указывает на адресуемое устройство. Здесь кроется причина невозможности параллельной работы двух устройств на одной шине ATA: обратиться к устройству можно только после освобождения обоих устройств.

3. Хост читает основной или альтернативный регистр состояния адресованного устройства, дожидаясь признака готовности *(DRDY*=1).

4. Хост заносит требуемые параметры в блок командных регистров.

5. Хост записывает код команды в регистр команд.

6. Устройство устанавливает бит BSYи переходит к исполнению команды. Дальнейшие действия зависят от протокола передачи данных, заданного командой (см. графу «Протокол» в табл. 5.9).

Для команд, не требующих передачи данных (ND):

7. Завершив исполнение команды, устройство сбрасывает бит BSY и устанавливает запрос прерывания (если он не запрещен). К этому моменту в регистрах состояния и ошибок уже имеется

информация о результате исполнения. Выполнение завершается.

Единичное значение бита *BSY* может промелькнуть между шагами 6 и 7 так быстро, что хост его не зафиксирует, но для фиксации факта выполнения команды или ее части предназначен запрос прерывания.

Для команд, требующих чтения данных в режиме PIO (PI):

7. Подготовившись к передаче первого блока данных по шине ATA, устройство устанавливает бит *DRQ*. Если была ошибка, она фиксируется в регистрах состояния и ошибок. Далее устройство сбрасывает бит *BSY* и устанавливает запрос прерывания (если он не запрещен).

8. Зафиксировав обнуление бита *BSY* (или по прерыванию), хост считывает регистр состояния, что приводит к сбросу прерывания от устройства.

9. Если хост обнаружил единичное значение бита *DRQ,* он производит чтение первого блока данных в режиме *PIO* (адресуясь к регистру данных). Если обнаружена ошибка, считанные данные могут быть недостоверными.

После передачи блока данных возможно одно из следующих действий:

- Если на шаге 8 ошибка не обнаружена и требуется передача следующего блока, устройство устанавливает бит *BSY* и данная последовательность повторяется с шага 7.
- Если есть ошибка или передан последний блок данных, устройство сбрасывает бит *DRQ* и выполнение команды завершается.

Для операций записи данных после шага 6 для устройства начинается активная фаза записи на носитель, что отмечается установкой бита BSY.

Для команд, требующих записи данных в режиме PIO (PO и P):

7. Подготовившись к приему первого блока данных по шине ATA, устройство устанавливает бит *DRQ* (если нет ошибок) и сбрасывает бит *BSY.* Если была ошибка, она фиксируется.

8. Зафиксировав обнуление бита *BSY,* хост считывает регистр состояния.

9. Если хост обнаружил единичное значение бита *DRQ,* он производит запись первого блока данных в режиме PIO по адресу регистра данных.

10. После передачи блока данных возможно одно из следующих действий:

- Если обнаружена ошибка, устройство сбрасывает бит *DRQ,* устанавливает запрос прерывания и выполнение команды завершается. Переданные по шине данные остаются необработанными устройством (не записываются на носитель).
- Если ошибка не обнаружена, устройство устанавливает бит BSY и переходит к следующему шагу.

11. Устройство обрабатывает принятый блок данных, затем:

- Если нет ошибок и обработанный блок — последний, устройство сбрасывает бит BSY и устанавливает запрос прерывания, на чем выполнение команды успешно завершается.
- Если обнаружена ошибка, выполнение команды завершается таким же образом, но с установкой битов ошибок.
- Если нет ошибок и требуется передача следующего блока, выполняются следующие шаги:

12. По готовности приема следующего блока устройство устанавливает бит *DRQ,* сбрасывает бит *BSY* и устанавливает запрос прерывания.

13. По обнулению бита *BSY* (или по прерыванию) хост считывает регистр состояния.

14. Обнаружив бит *DRQ,* хост выполняет запись очередного блока в регистр данных, и последовательность повторяется с шага 11.

Команды с передачей данных в режиме DMA выполняются похожим образом, но:

- Вместо PIO используется прямой доступ к памяти. Хост должен проинициализировать канал DMA до записи кода в регистр команд, чтобы при появлении сигнала DMARQ начался обмен.
- Запрос прерывания даже в многосекторных передачах производится один раз — по выполнении команды.

5.2.1.6. Протоколы и режимы передачи данных

Программа общается с устройствами ATA через регистры, используя инструкций ввода/вывода *IN* и *OUT.* Для передачи данных с максимальной скоростью применяют программный доступ к регистру данных или DMA. Тип обмена задается командой. Программный доступ обязателен для всех устройств. Команды режима DMA устройствами могут не поддерживаться.

Программный доступ PIO (Programmed Input/Output) выполняется в виде следующих друг за другом

операций чтения или записи в пространстве ввода/вывода по адресу регистра данных. В отличие от программно-управляемого ввода/вывода, применяемого, например, для общения с LPT-портом, передача блока данных в режиме PIO производится без программного опроса какого-либо бита готовности для передачи каждого слова. Готовность устройства проверяется перед началом передачи блока, после чего хост производит серию операций в определенном темпе. Темп определяется выбранным режимом *PIO Mode 0 -4,* для каждого режима определены допустимые параметры временной диаграммы цикла обмена. Обмен PIO программно реализуется с помощью процессорных инструкций ввода/вывода строк *REP INS* или *REP OUTS с* занесенным в регистр *CX* количеством слов (или байтов) в передаваемом блоке. Эти инструкции обеспечивают максимально возможную скорость обмена для данного процессора и системной шины. «Обуздать» процессор в соответствии с выбранным режимом входит в задачу адаптера ATA, который использует для удлинения цикла сигнал готовности шины (для ISA — IOCHRDY). Традиционные режимы 0,1 и 2 имеют временные параметры, фиксируемые только хост-адаптером. Для прогрессивных режимов ATA-2 (PIO Mode 3 и старше) устройство может затормозить обмен, используя сигнал готовности IORDY. Программный обмен на все время передачи блока занимает и процессор, и системную шину.

Таблица 5.15. Параметры режимов передачи

Режим передачи	Минимальное время цикла, нс	Скорость передачи, Мбайт/с	Интерфейс
PIO mode 0	600	3,3	ATA
РЮ mode 1	383	5,2	ATA
PIO mode 2	240	8,3	ATA
PIO mode 3	180	11,1	E-IDE, ATA-2 (используется IORDY)
PIO mode 4	120	16,6	E-IDE, Fast ATA-2 (используется IORDY)
Singleword DMA Mode 0	960	2,08	ATA
Singleword DMA Mode 1	480	4,16	ATA
Singleword DMA Mode 2	240	8,33	ATA
Multiword DMA Mode 0	480	4,12	ATA
Multiword DMA Mode 1	150	13,3	ATA-2
Multiword DMA Mode 2	120	16,6	Fast ATA-2
Ultra DMA Mode 0	120[1]	16,6	ATA/ATAPI-4
Ultra DMA Mode 1	80[1]	25	ATA/ATAPI-4
Ultra DMA Mode 2	60[1]	33	ATA/ATAPI-4
Ultra DMA Mode 3	45[1]	44,4	ATA/ATAP1-5
Ultra DMA Mode 4	30[1]	66,6	ATA/ATAP1-5
Ultra DMA Mode 5	20[1]	100	ATA/ATAP1-6?

[1] В пакете данных режима Ultra DMA за каждый такт передаются два слова данных, один по фронту синхронизирующего сигнала, другой по спаду. Период следования синхросигналов равен удвоенному времени цикла.

Обмен по каналу DMA занимает исключительно шины ввода/вывода и памяти. Процессору требуется выполнить только процедуру инициализации канала, после чего он свободен до прерывания от устройства в конце передачи блока (этим могут воспользоваться многозадачные системы). Стандартные каналы DMA шины ISA для интерфейса ATA практически не используются из-за низкой пропускной способности. Высокопроизводительные адаптеры ATA могут иметь собственные более эффективные контроллеры. Режимы обмена по каналу DMA бывают одиночными и множественными. При *одиночном режиме — Single-word DMA* — устройство для передачи каждого слова вырабатывает сигнал запроса DMARQ и сбрасывает его по сигналу DMACK#, подтверждающему цикл обмена. При *множественном режиме — Multiword DMA* — на сигнал DMARQ хост отвечает потоком циклов, сопровождаемых сигналами DMACK#. Если устройство не справляется с потоком, оно может приостановить его снятием сигнала DMARQ, а по готовности установить его снова. Множественный режим позволяет развить более высокую скорость передачи.

В спецификации ATA/ATAPI-4 появился новый режим — *Ultra DMA,* позволяющий перешагнуть барьер в 16,6 Мбайт/с, свойственный традиционным режимам и используемому кабелю. При этом обеспечивается и контроль достоверности передачи данных по шине, чего не делалось ни в PIO, ни в стандартных режимах DMA (а зря!). Стандартом ATA-4 было определено 3 режима Ultra DMA (0,1 и 2), впоследствии ввели новые (табл. 5.15); выбор режима осуществляется командой *Set Features.* В режимах Ultra DMA сигналы DMARQ и DMACK# сохраняют свое назначение, а вот смысл сигналов DIOR#,

DIOW# и IORDY на время передачи пакета (Ultra DMA Burst) существенно меняется (см. п. 5.2.1.1). В пакете данные на шине сопровождаются стробом, генерируемым источником данных, причем для синхронизации используются оба перепада сигналов. Это позволяет повысить пропускную способность шины, не увеличивая частоту переключения сигналов сверх 8,33 $с^{-6}$ (этот предел для обычного кабеля достигается в режиме PIO Mode 4 и Multiword DMA Mode 2). Каждое переданное слово участвует в подсчете CRC-кода, который передается в конце пакета. Подсчет ведется и источником данных, и приемником. При несовпадении принятого и ожидаемого кода фиксируется ошибка передачи. Передача в пакете может приостановиться, если приемник снимет сигнал готовности (DDMARDY# или HDMARDY#). Передача пакета может прекращаться по инициативе устройства (снятием сигнала) или хоста (сигналом STOP). Противоположная сторона должна подтвердить окончание цикла сигналом STOP или DMARQ соответственно.

Тип режима обмена определяется возможностями хост-адаптера (и его драйвера), устройств и кабеля, и для каждого устройства он будет ограничен минимумом из максимальных возможностей всех этих компонент. Как правило, режимы устанавливаются системой автоматически, но пользователю дается возможность при необходимости «подрезать крылья» контроллеру опциями BIOS Setup. Для работы в режимах Ultra DMA Mode 3 и выше требуется 80-проводный кабель (см. п. 5.2.1.1), присутствие которого должно быть программно определено до включения этого режима обмена. Правда, в спецификации есть оговорка, что при двухточечном соединении (контроллер-устройство) для режимов 3 и 4 можно использовать 40-проводный кабель (без среднего разъема). Система не должна позволить пользователю применить высокоскоростные режимы на обычном кабеле, при этом желательно, чтобы она сообщала об обнаруженном несоответствии. Для определения типа кабеля есть несколько возможностей:

Определение через хост-контроллер, для чего он должен иметь приемник сигнала CBLID#. После включения питания или аппаратного сброса хост ожидает завершения протокола сброса, после чего подает команду идентификации *Identyfy(Packet) Device* устройству 1. Устройство 1 стандарта ATA-3 и выше обязано после сброса снять сигнал PDIAG#/CBLID# не позднее, чем по приходу первой команды. Если обнаружено старое устройство, то протокол идентификации кабеля хостом работать не будет (но со старым устройством на шлейфе включать высокоскоростной режим и не стоит). Современное устройство снимет сигнал, и хост-адаптер сможет определить наличие 80-проводного кабеля по низкому уровню CBLID#. Некорректно работающее устройство 1 может удержать сигнал в низком уровне, и в этом случае 40-проводный кабель будет ошибочно трактоваться как 80-проводный.

Определение через устройство не требует дополнительного приемника в контроллере — линия PDIAG#/CBLID# в хост-контроллере заземляется через конденсатор ёмкостью 0,047 мкф ±20%. Приемник сигнала идентификации кабеля располагается в устройстве. Для определения типа кабеля хост посылает команду *Identify (Packet) Device* устройству 1, чтобы оно освободило линию (сняло сигнал PDIAG#). Затем команда идентификации посылается устройству 0. Примерно через 30 мкс после получения команды устройство кратковременно обнуляет эту линию, затем отпускает и только через 20-30 мкс после этого считывает ее состояние и сообщает его в бите 13 слова 93. Линия подтягивается к +5 В резисторами 10 кОм, установленными в устройствах. Если используется 40-проводный кабель, то конденсатор в хост-адаптере не успеет перезарядиться и устройство сообщит нулевое значение бита. Если кабель 80-проводный, то конденсатор оказывается изолированным от линии PDIAG#/CBLID# в устройстве, и оно сообщит единичное значение бита. Если на хост-контроллере нет конденсатора, то даже 40-проводный кабель будет идентифицироваться как 80-проводный, что опасно при передаче данных. Если некорректно работающее устройство 1 своевременно не отпустит линию, то даже 80-проводный кабель будет казаться 40-проводным.

Комбинированный метод определения предполагает наличие на хост-контроллере и приемника сигнала CBLID#, и конденсатора (они друг другу не мешают). Решение о наличии 80-проводного кабеля принимается, только если оба метода укажут на его наличие. Возможная ошибка идентификации будет безопасной — при некорректном устройстве 1 на 80-проводном кабеле не будет включен высокоскоростной режим (наверное, это и к лучшему).

Правильный выбор режима обмена обеспечивает надежность и производительность. Все устройства поддерживают режим PIO Mode 0, в котором считывается блок параметров идентификации. В блоке имеются поля, описывающие режим обмена по умолчанию и более эффективные режимы обмена, поддерживаемые устройством. Командой *Set Features* можно изменить параметры режима. Иногда накопитель не обеспечивает надежной передачи данных в заявленном высокоскоростном режиме. Если данные начинают пропадать, первым делом следует понизить режим обмена.

BIOS определяет режим обмена с каждым устройством с учетом ограничений, заданных в Setup. Старые диски, не сообщающие своих параметров, могут не работать со старшими режимами PIO. На одном ленточном кабеле (канале ATA) могут присутствовать устройства с разным режимом обмена — спецификация это допускает. Однако реально могут возникать аппаратные или программные ограничения. Некоторые чипсеты не позволяют независимо программировать режим обмена для устройств канала. В таком случае при подключении двух разных устройств (например, PIO Mode 1 и 3) обмен с

обоими устройствами будет происходить со скоростью меньшего (PIO Mode 1). Поэтому не рекомендуется к одному каналу ATA (порту IDE) подключать быстрый винчестер и медленный CD-ROM. Иногда завязка режимов обмена двух устройств обусловлена ограниченным набором параметров конфигурации в BIOS. Быстрые режимы множественного обмена по DMA реализуются только драйверами ОС. «Глупый» драйвер может попытаться навязать медленный режим обоим устройствам канала, так что смешивать разные устройства не стоит и по этой причине.

Протокол обмена PIO хорош только для однозадачных ОС. Для многозадачных ОС больший интерес представляет обмен по DMA, если, конечно, поддерживаемый режим обеспечивает приемлемую скорость обмена (см. табл. 5.15).

5.2.1.7. Средства многозадачности (ATA-4)

Главным недостатком интерфейса ATA в многозадачных системах является то, что когда одно устройство на шине исполняет команду, другое использоваться не может. В этом ATA существенно уступает SCSI, где устройство на время длительной внутренней операции может освобождать шину и имеется эффективный механизм организации очередей процессов ввода/вывода. Спецификация ATA-4 определяет возможности параллельного выполнения команд обоими устройствами и создания очередей.

Перекрытие команд (overlapped feature) позволяет устройству, занятому длительной внутренней операцией, освободить шину. Для этого устройство должно сбросить биты *PRQ* и BSY регистра состояния. По окончаний выполнения операции устройство устанавливает бит *SERV* в своем регистре состояния. Если хост намерен использовать шину для обращения к другому устройству, он должен запретить прерывания от текущего выбранного устройства, установив бит *nIEN* в регистре управления. После того как хост вернется к обслуживанию данного устройства, он должен послать ему команду *NOP с* подкомандой 01 для получения информации о состоянии. При этом он может разрешить прерывание, которое произойдет по готовности устройства к продолжению. Обнаружив установленный бит *SERV,* хост посылает команду *Service,* которая вызовет продолжение исполнения команды, во время которой шина освобождалась. Принятый механизм продолжения менее эффективен, чем в SCSI, — он требует привлечения хоста для обнаружения готовности устройства. Готовность устройства определяется хостом путем полинга (периодического опроса состояния устройств), реализуемого хост-контроллером (аппаратно) или программой хост-компьютера. Перекрытие допускается только для команд *NOP* (с подкомандой 01), *Packet, Read DMA Queued, Service* и *Write DMA Queued.*

Устройства могут поддерживать *очереди команд,* но только для команд, допускающих перекрытие (эти свойства тесно связаны). Если при наличии команд в очереди устройство получает команду, не входящую в этот список, команда отвергается (с соответствующим битом в регистре ошибок) и очередь сбрасывается. Глубина очереди, поддерживаемой устройством, сообщается в блоке параметров идентификации словом 75, нулевому значению слова соответствует глубина очереди 1 — фактически, отсутствие поддержки очередей. Команды ставятся в очередь с уникальным идентификатором-тегом, который передается через регистр счетчика секторов SC. После исполнения команды *Service* значение тега для обслуживаемой команды считывается из того же регистра, что позволяет ее идентифицировать. Если устройство получает команду со значением тега, которое уже присутствует в очереди, и новая, и старая команды отвергаются (состояние не определено). При возникновении любой ошибки вся очередь сбрасывается. Следующая команда в очередь посылается лишь при освобожденной шине. Перед посылкой хост должен запретить прерывания, а разрешить их может только после посылки новой команды. Если для выполнения команды устройству не нужно освобождать шину, команда будет выполнена немедленно, а находящиеся в очереди команды, освободившие шину, будут выполнены позже. Очереди SCSI более эффективны как по набору команд (в очередь ставятся процессы, которые могут представлять собой цепочки команд), так и по гибкости управления. SCSI также обеспечивает независимость исполнения одних команд очереди от результатов выполнения других.

5.2.1.8. Пакетный интерфейс ATAPI

Для подключения к интерфейсу ATA накопителей CD-ROM и стримеров (а также других устройств) недостаточно набора регистров и системы команд ATA. Для них существует аппаратно-программный интерфейс ATAPI (ATA Package Interface — пакетный интерфейс ATA). Устройство ATAPI поддерживает минимальный набор команд ATA, который неограниченно расширяется 16-байтным командным пакетом, посылаемым хост-контроллером в регистр данных устройства по команде Packed Структура командного пакета пришла от SCSI, что обеспечивает схожесть драйверов для устройств со SCSI и ATAPI. Классификация устройств совпадает с принятой в SCSI (см. табл. 5.31), класс устройства сообщается им в начале блока параметров идентификации.

Интерфейс ATAPI может использоваться с неинтеллектуальными адаптерами ATA, поскольку для хост-адаптера поддержка ATAPI может выполняться чисто программно. Сложные контроллеры ATA, имеющие кэш-память и собственный процессор, не ориентированные на интерфейс ATAPI, могут не догадаться, что в регистр данных устройства кроме 512-байтаых блоков данных можно записывать 16-байтный блок с командным пакетом. Устройства ATAPI имеют следующие особенности:

- Команду *Identify Device* они должны отвергать, устанавливая в блоке командных регистров сигнатуру ATAPI, чтобы хост не пытался к ним обратиться как к ATA-устройствам. Для идентификации устройств ATAPI предназначена специальная команда *Identify Packet Device,* а блок параметров, сообщаемых устройством, трактуется иначе (табл. 5.16).
- Для программного сброса устройства ATAPI предназначена команда *Device Reset,* которую устройства ATA отвергают. Программный сброс через регистр управления не прекращает выполнение команды *Packet.*
- Специфические команды вместе с необходимыми параметрами передаются по команде *Packet,* код которой является недействительным для устройств ATA.

Таблица 5.16. Идентификация устройств ATAPI

Слово	F/V	Назначение .
0	F	Общая конфигурационная информация Биты [15:14] = 10 для всех устройств ATAPI (11 —зарезервировано) Бит 13 — зарезервирован Биты 12-8 — набор поддерживаемых команд (класс устройства) Бит 7:1 — устройство со сменным носителем. Бит 6-5: 00 — устройство установит *DRQ в* течение 3 мс после получения команды *Packet* 01 — устройство введет INTRQ по установке *DRQ* после получения команды *Packet* (устарело 5+) 10 — устройство установит DRQ в течение 50 мкс после получения ,. команды *Packet* 11 — зарезервировано Биты 4 -3 — зарезервированы Бит 2 — сообщение параметров не завершено, см. слово 2 (5+) Биты 1 -0 — длина командного пакета: 00 — 12 байт, 01—16 байт, 1х — Зарезервировано
1	R	Зарезервировано
2	R	Специфическая конфигурация (5+): 37C8h — устройство требует для раскрутки мотора подкоманду *Set Features,* сообщенный блок параметров неполный 738Ch — устройство требует для раскрутки мотора подкоманду *Set Features,* сообщенный блок параметров полный 8C73h — устройство не требует для раскрутки мотора подкоманды *Set Features,* сообщенный блок параметров неполный C837h — устройство не требует для раскрутки мотора подкоманды *Set Features,* сообщенный блок параметров полный
3-9	R	Зарезервировано
10-19	F	Серийный номер (20 символов ASCII), если слово 10 не нулевое; иначе — специфичная информация.
20-22	R	Зарезервировано
23-26	F	Версия встроенного ПО (8 символов ASCII), если слово 23 не нулевое; иначе — специфичная информация
27-46	F	Номер модели (40 символов ASCII), если слово 27 не нулевое; иначе — специфичная информация
47-48	R	Зарезервировано
49		Возможности:
	F	Бит 15:1 — поддержка чередующегося (interleaved) DMA
	F	Бит 14:1 — поддержка очередей команд
	F	Бит 13:1 — поддержка перекрывающихся команд
	F	Бит 12:1 —требуется программный сброс (устарело 4+)
	F	Бит 11: 1 — IORDY поддерживается (для PIO Mode 3 и выше — обязательно), 0 — IORDY не обязан поддерживаться
	F	Бит 10:1 — сигнал IORDY может быть запрещен командой *Set Features*
	F	Бит 9:1— поддержка LBA
	F	Бит 8:1— поддержка DMA
	X	Биты 7-0 специфичны
50	R	Зарезервировано

Таблица 5.16. Идентификация устройств ATAPI (Продолжение)

Слово	F/V	Назначение .
51	F	Биты 15-8: номер PIO Mode (устарело 5+)
	X	Биты 7-0 специфичны
52	R	Зарезервировано
53	R	Биты 15-3 зарезервированы
	F	Бит 2: 1 — поля в слове 88 действительны, 0 — нет.
	F	Бит 1: 1— поля в словах 64-70 действительны, 0 — нет. Устройство, поддерживающее PIO Mode 3 и старше или Multiword DMA Mode 1 и выше, должно использовать эти поля
	V	Бит 0:1— поля в словах 54-58 действительны, 0 — поля могут быть действительны
54-62	R	Зарезервировано
63	R	Режим множественного обмена Multiword DMA: Биты 15-11 зарезервированы Биты 10-8: активный режим: бит 8=1 — Mode0, бит 9=1 — Mode 1 и т. д. Единичное значение может иметь только один бит, но только если в слове 88 биты 15-8 нулевые (активным может быть только один режим DMA).
	R	Биты 7-3 зарезервированы
	V	Биты 2-0: поддерживаемые режимы: бит 0=1 — Mode 0, бит 1=1— Mode 1 и т. д.
64	R	Биты 15-8 зарезервированы
	F	Биты 7-0 — поддерживаемые прогрессивные режимы программного обмена (Advanced PIO): бит 0=1 — PIO Mode 3, бит 1=1 —PIO Mode 4, биты 2-7 — зарезервированы
65	F	Минимальная длительность цикла передачи множественного DMA (в наносекундах)
66	F	Рекомендованная длительность цикла передачи множественного DMA (в наносекундах). При многосекторной передаче обеспечивает оптимальную скорость, при которой устройство не тормозит поток снятием запроса DMARQ
67	F	Минимальная длительность цикла передачи PIO без использования сигнала готовности (в наносекундах)
68	F	Минимальная длительность цикла передачи PIO с использованием сигнала готовности (в наносекундах)
69-70	R	Зарезервированы для поддержки перекрытия команд и очередей
71	F	Типовое время от получения команды *Packet* до освобождения шины (в наносекундах)
72	F	Типовое время от получения команды *Service* до сброса *BSY* (в наносекундах)
73-74 ,	R	Зарезервировано
75	F	Глубина очереди команд: биты 15 - 5 — зарезервированы, биты 4 - 0 — максимальная глубина очереди-1
76-79	R	Зарезервировано
80	F	Главный номер версии интерфейса (если не 0000h или FFFFh): 1 —ATA-1, 2—ATA-2ит.д.
81	F	Дополнительный номер версии интерфейса (если не 0000h или FFFFh)
82	F	Поддержка команд и свойств (если слова 82 и 83 не равны 0000h или FFFFh): Бит 15 — не используется Бит 14 — команды Nop Бит 13 — команды Read Buffer Бит 12 — команды Write Buffer Бит 11 — не используется Бит 10 — ограничение доступного пространства Бит 9 — команда Device Reset Бит 8 — вырабатывание прерывания во время команды Service Бит 7 — прерывание по освобождению шины Бит 6 — кэширование с упреждающим чтением Бит 5 — кэширование записи Бит 4 = 1 — команда Packet Бит 3 — управление энергопотреблением Бит 2 — смена носителя Бит 1 — команды Security БИТ 0 — SMART

Таблица 5.16. Идентификация устройств ATAPI (Продолжение)

Слово	F/V	Назначение
83	F	Поддержка команд и свойств (как и слово 82): Бит 15 = 0 Бит 14=1 Биты 13-9 — зарезервированы Бит 8 — расширение *Set Max* для защиты (5+) Бит 7 — резерв Бит 6 —специальная команда запуска двигателя после включения (5+) Бит 5 —переход в режим Standby по включению питания (Power-Up In Standby, 5+) Бит 4 — уведомление о смене носителя Биты 3 - 1 —зарезервированы Бит 0 — команда *Download Microcode*
84	F	Поддержка команд и свойств (если слова 82, 83 и 84 не равны 0000h или FFFFh): Бит 15=0 Бит 14=1 Биты 13 - 0 —зарезервированы
85-87	F	Разрешенные команды и свойства — поля и правила аналогичны словам 82-84
88		Режим Ultra DMA:
	R	Биты 15-13 — зарезервированы Биты 10-8 — активный режим Ultra DMA: бит 8=1 — Mode 0, бит 9=1 — Mode 1 и т. д. Единичное значение может иметь только один бит, но только если в слове 63 биты 15-8 нулевые
	R	Биты 7 - 5 — зарезервированы
	V	Биты 4 - 0 — поддерживаемые режимы одиночного обмена DMA: бит 0=1 —Mode 0, бит 1=1 —Mode 1 и т.д.
89-92	R	Зарезервировано
93	V	Результат аппаратного сброса (5+) Бит 15 = 0 Бит 14 = 1 Бит 13 — обнаруженный уровень CBLID#: 0 — ниже V_{IL}., 1 — выше V_{IH} Биты 12 - 8 — результаты сброса устройства 1 (в устройстве 0 обнулены) Бит 12 — резерв Бит 11 = 1— устройство 1 ввело сигнал PDIAG#, 0 — нет Биты 10 - 9 — способ, которым устройство 1 узнало свой номер: 00—резерв, 01 — по джамперу, 10—по CSEL, 11 —иным или неизвестным способом Бит 8 = 1 Биты 7 - 0 — результаты сброса устройства 0 (в устройстве 1 обнулены) Бит 7—резерв Бит 6 = 1— устройство 0 ответило при выборе устройства 1,0— нет Бит 5 = 1— устройство 0 обнаружило сигнал DASP#, 0 — нет Бит 4 = 1 — устройство 0 обнаружило сигнал PDIAG#, 0 — нет Бит 3 = 1— устройство 0 успешно прошло диагностику, 0 — нет Биты 2 - 1 — способ, которым устройство 0 узнало свой номер: 00 — резерв, 01 — по джамперу, 10 — по CSEL, 11 — иным или неизвестным способом Бит 0 = 1
94 - 125	R	Зарезервировано
126	R	Поведение ATAPI при нулевом счетчике байтов
127	R	Биты 15-2 — зарезервированы
	F	Биты 1 - 0 — уведомление о смене носителя: 00 — не поддерживается, 01 — поддерживается, 1x—зарезервированы
128	V	Состояние защиты: Биты 15 - 9 — зарезервированы Бит 8 — уровень защиты: 0 — высокий, 1 — максимальный Биты 7-6 — зарезервированы Бит 5: 1 — поддержка расширенного защитного стирания Бит 4: 1 — счетчик попыток иссяк Бит 3: 1— команды защита блокированы Бит 2: 1 —устройство заблокировано Бит 1: 1— защита разрешена Бит 0:1 —защита поддерживается

Таблица 5.16. Идентификация устройств ATAPI (Продолжение)

Слово	F/V	Назначение
129-159 160-254 255	X R F/V	Специфично Зарезервировано Контроль целостности (5+): Биты 15 - 8 — контрольная сумма (сумма всех 512 байт по модулю 256 должна быть нулевой) Биты 7-0 — сигнатура A5h (признак использования контрольной суммы)

Для устройств ATAPI допустимы далеко не все команды (см. табл. 5.10). Обязательная команда *Read Sector(s)* устройствами ATAPI выполняется своеобразно: она отвергается, но в блоке командных регистров оставляет сигнатуру пакетного устройства.

Получив команду *Packet,* устройство устанавливает бит занятости *BSY* и готовится принять пакет. По готовности к приему оно устанавливает бит *DRQ* и сбрасывает BSY, что является сигналом хосту для передачи пакета в режиме PIO. Во время передачи последнего слова устройство сбрасывает *DRQ,* устанавливает *BSY* и начинает отработку пакетной команды.

При подаче команды *Packet* регистр свойств *FR* содержит признаки команды:

- Бит 0 — *DMA* — является указанием на использование DMA или Ultra DMA для обмена данными.
- Бит 1 — *OVL* — является признаком возможности перекрывающегося выполнения команд.

Регистры *CH* и *CL* содержат лимит счетчика байтов данных, передаваемых по каждому введению *DRQ* в режиме PIO. Если команда не предусматривает обмена данными, поле игнорируется. Если общее число требуемых байтов данных превышает лимит, значение лимита задается четным. Если оно равно или меньше лимита, лимит нечетен. Значение лимита FFFFh воспринимается устройством как FFFEh. Лимит счетчика, как и бит DMA, к передаче пакета отношения не имеет — они относятся к данным, передаваемым при исполнении команды, заключенной в пакет.

Регистр *D/H* используется только для выбора устройства. Для устройств, поддерживающих очереди команд, биты [7:3] регистра *SC* содержат тег (tag).

После подачи команды регистр SC кроме тега будет содержать следующие биты:

- Бит 2 — *REL* (Release) — признак освобождения шины (для перекрывающихся команд).
- Бит 1 — *I/O* (Input/Output) — указатель направления передачи данных (0 — передача к устройству, 1 — к хосту).
- Бит 0 — *C/D* (Command/Data) — признак передачи команды (1) или данных (0).

В регистре состояния *SR* некоторые биты получают новое назначение:

- Бит 5 — *DMRDY* (DMA Ready) — при DRQ=l используется как признак готовности к обмену в режиме (Ultra) DMA. При DRQ=0 является признаком отказа устройства *DF* (Device Fault).
- Бит 4 — *SERV* (Service) — признак готовности к обслуживанию команды, освободившей шину.
- Биты 1, 2 не используются.
- Бит 0 — *CHK* (Check) — признак ошибки.

Остальные биты сохранили обычное назначение.

После подачи команды Packet хост определяет состояние устройства, прочитав его регистры. Возможны следующие варианты:

- Ожидание команды *Packet:* в регистре SC биты *C/D=1, I/O=0*, *REL=0,* поле *Tag* содержит ранее установленное значение. В регистре состояния *BSY=0, DMRD=0, CHK=O, DRQ=1,* бит *SERV* может указывать на наличие команды, ожидающей обслуживания.
- Передача данных: в регистре SC биты *C/D=0*, *REL=0, I/O* указывает на направление передачи. Регистры *CH, CL* при использовании PIO содержат счетчик байтов данных. В регистре состояния *BSY=0,* DRQ=1, *CHK=0, DMRDY* указывает на использование DMA, бит *SERV* может указывать на наличие команды, ожидающей обслуживания.
- Освобождение шины: в регистре SC биты *C/D=0*, *1/O=0, REL=1,* поле Tag содержит тег. В регистре состояния *BSY=0*, *DMRDY=0, CHK=0, DRQ=0, бит SERV* может указывать на наличие команды, ожидающей обслуживания.
- Запрос обслуживания: в регистре SC биты *C/D=0, I/O=0*, *REL=1,* поле *Tag* содержит тег. В регистре состояния *BSY=0, DMRDY=0, CHK=0, DRQ=0,* бит *SERV=1.*
- Успешное завершение: в регистре SC биты *C/D=1, I/O=1*, *REL=0,* поле *Tag* содержит тег. В регистре состояния *BSY=0, DRDY=1, CHK=O , DRQ=0*, бит *SERV* может указывать на наличие команды, ожидающей обслуживания.
- Завершение с ошибкой (только после передачи последнего байта пакета): в регистре SC биты

C/D=1, *I/O=1*, *REL=0,* поле *Tag* содержит тег. В регистре состояния *BSY=0*, *DRDY=1, DRQ=0, DF=1* при отказе устройства, *CHK=1,* если регистр ошибок содержит код ошибки, бит *SERV* может указывать на наличие команды, ожидающей обслуживания. В регистре ошибок биты 0 и 1 трактуются в зависимости от команды, бит 2 — признак отвергнутой команды *(ABRT),* биты [7:4] могут содержать уточненное состояние.

Структуру командного пакета см. в п. 5.3.1.3 и 5.3.1.4. При любой длине блока дескрипторов, которая определяется кодом команды (нулевой байт пакета), передаваемый пакет имеет длину 16 байт, но используется только указанное количество байтов. Систему команд и структуру пакетов стандарт ATA/ATAPI-4 не описывает, но для каждого класса устройств существует стандартизованный набор команд с определенной структурой пакетов.

5.2.1.9. Адаптеры и контроллеры шины ATA

Простейший адаптер ATA содержит только буферы сигналов шины и дешифратор зоны адресов. Все регистры контроллера и схемы кодирования размещены в самом устройстве IDE. Шина ATA требует выделения системных ресурсов — двух областей портов ввода/вывода и линии прерывания; дополнительно может использоваться канал DMA. Интерфейсу ATA первого канала выделили ресурсы ранее использовавшиеся контроллером жестких дисков. Второму каналу назначили ресурсы альтернативного контроллера жестких дисков. Позже определили ресурсы еще для двух каналов (табл. 5.17). Традиционному контроллеру жестких дисков выделялся канал DMA3, но он является 8-битным, в то время как шина ATA требует 16-битного канала DMA. Производительности стандартных каналов DMA для шины ATA явно недостаточно.

Таблица 5.17. Системные ресурсы каналов ATA

Канал	CS0	CS1	IRQ
1	1F0h-1F7h	3F6h-3F7h	14
2	170h-177h	376h-377h	15 или 10
3	1E8h-1EFh	3EEh-3EFh	12 или 11
4	168h-16Fh	36Eh-36Fh	10 или 9

На системных платах с шиной PCI стандартом является установка двухканального адаптера, занимающего ресурсы каналов 1 и 2. В идеальном варианте двухканальные контроллеры имеют шины, полностью изолированные друг от друга буферными и логическими схемами. В самом дешевом варианте они используют общие буферы для линий данных и управляющих сигналов и отдельные только для некоторых сугубо индивидуальных сигналов. С точки зрения логики здесь все в порядке, но следует учитывать нагрузочную способность (влияние паразитных параметров): суммарная длина обоих ленточных кабелей не должна превышать 46 см, а суммарная емкость каждой линии со всеми устройствами не должна превышать 35 пФ. Иначе на высокоскоростных режимах обмена возможны неконтролируемые искажения передаваемых данных. Стандарт ATA/ATAPI-6 предписывает следующее:

- В каждом канале должны быть собственные формирователи управляющих сигналов DIOR#, DIOW# и приемник IORDY (рекомендуемый вариант); либо должны быть раздельные формирователи CS0# и CS1 # (второй вариант логичнее, но хуже в плане помех). Такая конфигурация позволяет использовать все режимы обмена, кроме Ultra DMA.
- Для поддержки Ultra DMA 0,1 и 2 (до 33 Мбайт/с) к этим требованиям добавляется наличие раздельных формирователей сигнала DMACK#.
- Для поддержки Ultra DMA 3 и выше (44-100 Мбайт/с) все линии должны иметь раздельные приемопередатчики. Общими могут быть только сигналы RESET#, INTRQ, DA(2:0), CS0#, CS1# и DASP#, но стандарт этого не рекомендует.

Поскольку скорость программного обмена задается хост-адаптером, интересно индивидуальное программирование PIO Mode для каждого канала/устройства. Ряд чипсетов этого не допускает и при инициализации назначает общий минимальный режим. В результате подключение «тихоходного» устройства замедляет обмен быстрого соседнего устройства.

Современные системные платы оснащаются высокопроизводительными контроллерами шины ATA, обеспечивающими *прямое управление шиной PCI* (Bus Mastering) при обмене с устройствами в режимах DMA и Ultra DMA. Прямое управление шиной повышает суммарную производительность компьютера в многозадачных и многопоточных операционных системах. Сами по себе режимы DMA не дают выигрыша в скорости обмена по шине ATA — только режимы UltraDMA Mode I и старше превосходят по скорости PIO Mode 4 (см. табл. 5.15). Однако обмен в режиме DMA значительно меньше загружает центральный процессор компьютера, и параллельно с дисковым обменом процессор может заниматься обработкой других потоков (задач). В однозадачных (и однопотоковых) системах во время дискового

обмена процессор все равно ничем другим не занимается, поэтому для них хорош и режим PIO. Для реального использования прямого управления операционной системой в ней должен быть установлен специальный драйвер Bus-Master, соответствующий используемому контроллеру ATA (как правило, чипсету системной платы). В этом режиме каждый контроллер имеет свою специфику, в то время как для режима PIO любой контроллер характеризуется лишь адресами портов (блоков командных и управляющих регистров) и номером прерывания. Операционная система MS DOS режимы DMA (и прямое управление) не использует. Для многозадачных ОС (Windows 9x/NT/2000, OS/2, Unix, Linux, NetWare...) драйверы могут входить в комплект поставки ОС или поставляться производителями системных плат (контроллеров ATA). И, наконец, режим DMA должны поддерживать подключаемые устройства. Практически все современные устройства поддерживают Ultra DMA (или Multiword DMA), но если в паре с таким устройством к одному контроллеру подключено старое устройство, не поддерживающее этот режим, то прогрессивные режимы могут оказаться недоступны (по вине чипсета или драйвера) и для нового устройства.

ВНИМАНИЕ

Режим UltraDMA привлекателен не только скоростью и разгрузкой процессора, но и контролем достоверности передач по шине ATA (правда, этот контроль корректно работает не со всеми драйверами).

Поскольку контроллеры ATA подключаются к 32-разрядной шине PCI, в них ввели возможность обращения к регистру данных ATA двойными словами. При этом за одну 32-битную операцию процессора и шины PCI по шине ATA последовательно передаются два 16-битных слова. Возможностью 32-разрядного доступа к регистру данных можно управлять через опцию BIOS Setup (IDE 32-bit Transfer — Enable/Disable).

Более сложные контроллеры (отдельные карты расширения) могут иметь собственную кэш-память и управляющий процессор. Они могут аппаратно поддерживать «зеркальные» диски и организовывать RAID-массивы ATA-дисков. Некоторые адаптеры позволяют соединять несколько физических дисков в один логический на уровне вызовов BIOS.

Адаптеры ATA одно время часто размещали на звуковых картах (для подключения CD-ROM). По умолчанию им назначают ресурсы каналов 3 или 4. К этим каналам можно подключать винчестеры, но будет ли их там искать BIOS во время POST — вопрос. Современные версии BIOS позволяют хранить конфигурационные параметры четырех жестких дисков, более старые версии — двух. Четыре канала ATA физически позволяют подключить до восьми накопителей, но работа с ними лимитирована программными ограничениями.

Существуют гибридные адаптеры для подключения ATA HDD к шинам XT и MCA или, например, к LPT-порту. В последнее время получили распространение переходные адаптеры, позволяющие подключать устройства ATA/ATAPI к шине USB. При использовании USB 2.0 простота подключения внешнего устройства будет сочетаться и с высокой скоростью передачи данных.

5.2.1.10. Категории устройств IDE

Интерфейс ATA позволяет подключать устройства различных категорий, отличающихся «уровнем интеллекта» встроенного контроллера. Первые дисковые накопители IDE относятся к категории неинтеллектуальных устройств — *Non-Intelligent IDE.* Они не выполняли трансляцию нумерации секторов — их параметры внешней геометрии совпадали с реальными. Команды идентификации устройства и установки параметров не выполнялись. Дефектные блоки, отмеченные в заводском списке, были видны пользователю. Низкоуровневое форматирование выполнялось непосредственно по команде, так что неудачное форматирование могло понизить производительность из-за нарушения оптимальных установок чередования и смещения.

Позже появились более интеллектуальные устройства — *Intelligent ATA IDE.* Они способны выполнять расширенные ATA-команды — идентификацию устройства и установку параметров. Поддерживается трансляция геометрии. Дефектные секторы скрыты от пользователя (до исчерпания резерва). Для ускорения обмена эти устройства поддерживают блочные режимы передачи *Read Multiple* и *Write Multiple,* а также высокоскоростные режимы обмена PIO и DMA. Низкоуровневое форматирование возможно только при установке внешней геометрии, совпадающей с реальной. Однако форматирование опять-таки может «снести» заводскую оптимизацию, хотя более поздние устройства либо игнорируют стандартную команду форматирования трека, либо просто заполняют все секторы трека нулями.

К следующей категории относятся современные устройства с зонным форматом записи — *Intelligent Zoned Recording IDE.* Поскольку они имеют различное количество секторов на разных треках (для повышения плотности хранения), трансляция геометрии является для них обязательной (спецификация ATA не предусматривает сообщения устройством способа разбиения на зоны и формата каждой зоны, так что обращаться к ним можно только по внешнему трехмерному (CHS) или линейному (LBA) адресу). Информация о зонном распределении хранится, как правило, в служебной области носителя. Она

используется микропрограммой контроллера устройства, и с помощью специальных программных средств до нее можно добраться и, например, отключить дефектную зону (уменьшив доступный объем диска). Низкоуровневое форматирование по стандартной команде как таковое не выполняется. Такие устройства либо отвергают эту команду, либо выполняют фиктивно (только позиционируя головки), либо просто заполняют все секторы трека нулями.

Устройства IDE отличаются также по интеллектуальности контроллера, которая выражается в наличии таких средств, как автоматический мониторинг внутренних параметров (SMART), температурная коррекция системы позиционирования, поддержка управления энергопотреблением, и различных усовершенствований, направленных на повышение производительности.

5.2.1.11. Конфигурирование устройств

Устройства ATA перед подключением к шине должны быть корректно сконфигурированы. Конфигурирование подразумевает выбор типа интерфейса и определение адреса устройства. Тип интерфейса— XT или AT — определяется моделью накопителя. В изделиях фирмы Seagate, например, тип обозначается последней буквой в шифре модели: A — ATA (16 бит,) X — для XT (8 бит), а сочетание AX означает возможность выбора AT/XT с помощью джампера.

Существует два способа задания адреса устройства — с помощью кабельной выборки или явным заданием адреса на каждом из устройств. Режим кабельной выборки включается перемычкой *CS* (Cable Select — кабельная выборка). В этом случае оба устройства на шине конфигурируются одинаково — в режим *CS,* а адрес устройства определяется его положением на специальном ленточном кабеле (рис. 5.2). Кабельная выборка будет работать, если она поддерживается и задана на всех устройствах канала, включая хост-адаптер, который обеспечивает заземление контакта 28. При этом способе задания адресов синхронизация шпинделей накопителей через тот же провод контакта 28 исключается (актуально в RAID-массивах). Кабельная выборка применяется редко. Ее условное преимущество — унификация конфигурирования устройств, а недостатком является привязка физического положения устройств к кабелю — ведущее устройство должно быть ближе к адаптеру, чем ведомое. Возможно подключение адаптера к среднему разъему, а устройств — к крайним, но это не всегда удобно.

Более распространен режим явной адресации, при котором используется обычный «прямой» кабель (см. рис. 5.1). В этом случае перемычка *CS* не устанавливается, а адрес устройства задается перемычками, состав которых варьируется. В принципе, достаточно лишь указать устройству его номер (0/1), но в устройствах, разработанных до стандарта ATA, ведущему устройству «подсказывали» о наличии ведомого (по интерфейсу ATA оно могло бы определить это само по сигналу DASP#). Итак, на устройствах IDE можно увидеть следующие комбинации джамперов:

- *M/S* (Master/Slave — ведущее/ведомое) — переключатель адреса. Если на шине присутствует одно устройство, оно должно быть сконфигурировано как ведущее. Если на шине два устройства — одно должно быть ведущим, другое ведомым. Иногда джампер обозначается как «C/D» (диск C:/диск D:), но для второго канала IDE такое название некорректно. Когда появились первые IDE-диски емкостью 1 Гбайт, для преодоления барьера в 504 Мбайт некоторые модели допускали конфигурирование в виде двух устройств (0 и 1) половинной емкости. В таком режиме на их ленточный кабель IDE второе физическое устройство подключать нельзя.
- *SP* (Slave Present), *DSP* (Drive Slave Present), *Master but Slave is not ATA-compatible, Master but Slave uses only PDIAG-signal* — устанавливается на ведущем устройстве для указания на присутствие ведомого. Если переключатель установлен, а ведомое устройство не подключено, POST даст сообщение об ошибке. Джампер применяется для дисков, не использующих сигнал DASP#.
- *Single Drive* — джампер, устанавливаемый на устройстве, если оно единственное на шине (встречается на дисках Western Digital). Устройство будет ведущим.
- ACT (Drive Active) — джампер, соединяющий линию DASP# с формирователем сигнала активности устройства. Устанавливается на устройстве 0, встречается редко.

HSP — джампер, заземляющий линию DASP# (положение, взаимоисключающее *ACT).* Устанавливается на устройстве 1 для сигнализации о его присутствии (встречается редко). Для полностью ATA-совместимых дисков перемычка ставится только на ведомое устройство, а его присутствие ведущее определит автоматически.

Разобраться с джамперами старых устройств трудно, если нет документации, однако обширная база данных по разным моделям встроена в Help-файл утилиты Disk Manager. У современных устройств лишние джамперы упразднили, а существующие комментируются на наклейке-шильдике. Если джамперы устанавливаются рядом с интерфейсным разъемом, вероятно, они расположены в соответствии со стандартом ATA (рис. 5.4).

ВНИМАНИЕ

Следует учитывать, что перестановка джамперов часто воспринимается устройством только по включении питания. Кроме того, установка на один ленточный кабель двух разнотипных не ATA-устройств часто невозможна.

5.2.2. Примеры программ, непосредственно работающих с контроллером жесткого диска ATA

Листинг 5.1 содержит набор процедур, предназначенных для непосредственной работы с регистрами контроллера жесткого диска:

- процедура ReadHDDSector считывает с заданный сектор с указанного диска, используя LBA и протокол ввода PIO;
- процедура SendCommandToHDD предназначена для посылки команды контроллеру диска (она является внутренней вспомогательной подпрограммой данного модуля и не должна вызываться из других модулей программы);
- процедура ReadHDD_ID считывает идентификационный блок данных заданного диска, также используя протокол PIO.

Указанные процедуры используют самые простые способы обмена данными с жестким диском: линейную адресацию LBA и программный ввод/вывод PIO; прерывания, DMA и мультисекторная передача данных не применяются. Для определения готовности устройства к началу обмена информацией применяется метод циклического опроса регистра состояния: вначале проверяется бит занятости BSY (если устройство занято, то остальные разряды могут содержать недостоверную информацию), затем — бит ошибки, а в последнюю очередь — признаки готовности к приему команды DRDY или передаче данных DRQ. Несмотря на использование самых примитивных режимов, процедуры из листинга 5.1 являются достаточно эффективными при работе в однозадачном режиме DOS благодаря наличию у современных жестких дисков встроенной кэш-памяти большого объема.

Обратите внимание на необходимость защиты от зависания по длительности выполнения операции, которая должна быть встроена в циклы опроса регистра состояния. В данном примере использовалось с указанной целью значение состояния системного таймера, которое периодически (18 раз в секунду) записывается операционной системой в область данных BIOS (в 32-разрядное слово по абсолютному адресу 46Ch). Длительность ожидания сигнала готовности устройства, вообще говоря, зависит от типа этого устройства — современные дисководы могут автоматически переходить в спящее состояние, для возврата из которого в рабочий режим может потребоваться несколько секунд (для разгона мотора дисковода). В данном случае задано предельное значение длительности ожидания равное 10 тикам системного таймера, что соответствует примерно 0,5 с.

Листинг 5.1. Процедуры для непосредственной работы с контроллером жестких дисков ATA

```
; Максимальное время ожидания завершения операции
; (в "тиках")
MaxHDDWaitTime equ 10

DATASEG
; Стандартные базовые адреса каналов 1 — 4
StandardHDDBases DW 1F0h, 170h, 1E8h, 168h
; Номер канала
ChannelNumber DW ?
; Базовый адрес группы портов контроллера HDD
HDDBasePortAddr DW ?
; Номер диска
HDDNumber DB ?
; Параметры ATA-команды
ATAFeatures     DB ? ;особенности
ATASectorCount  DB ? ;количество обрабатываемых секторов
ATASectorNumber DB ? ;номер начального сектора
ATACylinder     DW ? ;номер начального цилиндра
ATAHead         DB ? ;номер начальной головки
ATAAddressMode  DB ? ;режим адресации (0 — CHS, 1 — LBA)
ATACommand      DB ? ;код команды, подлежащей выполнению
; Код ошибки (0 — нет ошибок, 1 — превышен допустимый
; интервал ожидания, 2 — неверный код режима адресации,
; 3 — неверный номер канала, 4 — неверный номер диска,
; 5 — неверный номер головки, 6 — ошибка при выполнении
; команды)
HDDErrorCode DB ?
; Момент начала очередной операции с диском
HDDTime DD ?
; Адрес считываемого сектора в режиме LBA
SectorAddress DD 0
; Область памяти для хранения прочитанного сектора
SectorDataBuffer DB 512 DUP (?)
ENDS

CODESEG
```

```
;************************************************
;*  ЧТЕНИЕ СЕКТОРА ЖЕСТКОГО ДИСКА В РЕЖИМЕ LBA   *
;* Входные параметры передаются через глобальные *
;* переменные:                                    *
;* ChannelNumber – номер канала (1 – 4);          *
;* HDDNumber – номер диска на канале (0 или 1);   *
;* SectorAddress – номер считываемого сектора.    *
;* Считывается основной сегмент данных            *
;* в массив SectorDataBuffer.                     *
;************************************************
PROC ReadHDDSector NEAR
        pushad
        push    ES
; Задать режим LBA
        mov     [ATAAddressMode],1
; Послать команду чтения сектора (с повторами)
        mov     [ATAFeatures],0
        mov     [ATASectorCount],1
        mov     EAX,[SectorAddress]
        mov     [dword ptr ATASectorNumber],EAX
        mov     [ATACommand],20h
        call    SendCommandToHDD
        cmp     [HDDErrorCode],0
        jne     @@End    ;закончить, сохранив код ошибки
; Ожидать готовность данных HDD
        mov     AX,0
        mov     ES,AX
        mov     DX,[HDDBasePortAddr]
        add     DX,7     ;адрес регистра состояния
@@WaitCompleet:
        ; Проверить время выполнения команды
        mov     EAX,[ES:046Ch]
        sub     EAX,[HDDTime]
        cmp     EAX,MaxHDDWaitTime
        ja      @@Error1 ;ошибка тайм-аута
        ; Проверить готовность
        in      AL,DX
        test    AL,80h   ;состояние сигнала BSY
        jnz     @@WaitCompleet
        test    AL,08h   ;состояние сигнала DRQ
        jz      @@WaitCompleet
; Принять сектор
        mov     AX,DGROUP
        mov     ES,AX
        mov     DI,offset SectorDataBuffer
        mov     DX,[HDDBasePortAddr] ;регистр данных
        mov     CX,256   ;число считываемых слов
        rep     insw     ;принять блок данных
; Сбросить признак ошибки
        mov     [HDDErrorCode],0
        jmp short @@End
; Записать номер ошибки
@@Error1:
        mov     [HDDErrorCode],1 ;ошибка тайм-аута
        jmp short @@End
@@End:  pop     ES
        popad
        ret
ENDP ReadHDDSector

;************************************************
;*          ПОСЛАТЬ КОМАНДУ ЗАДАННОМУ ДИСКУ     *
;* Входные параметры передаются через глобальные *
;* переменные:                                    *
;* ChannelNumber – номер канала (1 – 4);          *
;* HDDNumber – номер диска (0 или 1);             *
;* ATAFeatures – "особенности";                   *
;* ATASectorCount – количество секторов;          *
;* ATASectorNumber – номер начального сектора;    *
;* ATACylinder – номер начального цилиндра;       *
;* ATAHead – номер начальной головки;             *
;* ATAAddressMode – режим адресации (0-CHS, 1-LBA); *
;* ATACommand – код команды.                      *
;* После успешного выполнения функции:            *
;* в SectorDataBuffer – прочитанный сектор;       *
;* в HDDBasePortAddr – базовый адрес HDD;         *
;* в HDDTime – момент начала выполнения команды;  *
;* в HDDErrorCode – ноль.                         *
;* При возникновении ошибки в HDDErrorCode будет  *
;* возвращен код ошибки.                          *
;************************************************
```

```
PROC SendCommandToHDD near
        pushad
        push    ES
; Запомнить время начала операции с диском
        ; Загрузить в ES сегмент данных BIOS
        mov     AX,0
        mov     ES,AX
        ; Запомнить текущее время
        mov     EAX,[ES:046Ch]
        mov     [HDDTime],EAX
; Проверить значение кода режима
        cmp     [ATAAddressMode],1
        ja      @@Error2
; Проверить корректность номера канала
        mov     BX,[ChannelNumber]
        cmp     BX,1
        jb      @@Error3
        cmp     BX,4
        ja      @@Error3
; Установить базовый адрес
        dec     BX
        shl     BX,1
        mov     AX,[BX+StandardHDDBases]
        mov     [HDDBasePortAddr],AX
; Запретить прерывания от контроллера HDD
        mov     DX,[HDDBasePortAddr]
        add     DX,206h
        mov     AL,1010b
        out     DX,AL

; Ожидать "освобождения" HDD
        mov     DX,[HDDBasePortAddr]
        add     DX,7      ;адрес регистра состояния
@@WaitNotBSY:
        ; Проверить время выполнения команды
        mov     EAX,[ES:046Ch]
        sub     EAX,[HDDTime]
        cmp     EAX,MaxHDDWaitTime
        ja      @@Error1 ;ошибка тайм-аута
        ; Проверить состояние сигнала BSY
        in      AL,DX
        test    AL,80h    ;состояние сигнала BSY
        jnz     @@WaitNotBSY

; Ожидание готовности HDD к приему команды
        ; Выбрать нужный диск
        mov     DX,[HDDBasePortAddr]
        add     DX,6      ;адрес регистра головок
        mov     AL,[HDDNumber]
        cmp     AL,1      ;проверить номера диска
        ja      @@Error4
        shl     AL,4
        or      AL,10100000b
        out     DX,AL
        ; Ожидать, пока диск не будет готов
        inc     DX
@@WaitHDReady:
        ; Проверить время выполнения команды
        mov     EAX,[ES:046Ch]
        sub     EAX,[HDDTime]
        cmp     EAX,MaxHDDWaitTime
        ja      @@Error1 ;ошибка тайм-аута
        ; Проверить состояние BSY и DRDY
        in      AL,DX
        test    AL,80h    ;состояние сигнала BSY
        jnz     @@WaitHDReady
        test    AL,40h    ;состояние сигнала DRDY
        jz      @@WaitHDReady

; Загрузить команду в регистры контроллера
        mov     DX,[HDDBasePortAddr]
```

```
        inc     DX        ;регистр "особенностей"
        mov     AL,[ATAFeatures]
        out     DX,AL
        inc     DX        ;счетчик секторов
        mov     AL,[ATASectorCount]
        out     DX,AL
        inc     DX        ;регистр номера сектора
        mov     AL,[ATASectorNumber]
        out     DX,AL
        inc     DX        ;номер цилиндра (младший байт)
        mov     AX,[ATACylinder]
        out     DX,AL
        inc     DX        ;номер цилиндра (старший байт)
        mov     AL,AH
        out     DX,AL
        inc     DX        ;номер головки/номер диска
        mov     AL,[HDDNumber]
        shl     AL,4
        cmp     [ATAHead],0Fh  ;проверить номер головки
        ja      @@Error5
        or      AL,[ATAHead]
        or      AL,10100000b
        mov     AH,[ATAAddressMode]
        shl     AH,6
        or      AL,AH
        out     DX,AL
; Послать команду
        mov     AL,[ATACommand]
        inc     DX        ;регистр команд
        out     DX,AL
; Сбросить признак ошибки
        mov     [HDDErrorCode],0
        jmp short @@End

; Записать код ошибки
@@Error1:
        mov     [HDDErrorCode],1
        jmp short @@End
@@Error2:
        mov     [HDDErrorCode],2
        jmp short @@End
@@Error3:
        mov     [HDDErrorCode],3
        jmp short @@End
@@Error4:
        mov     [HDDErrorCode],4
        jmp short @@End
@@Error5:
        mov     [HDDErrorCode],5
        jmp short @@End
@@End:  pop     ES
        popad
        ret
ENDP SendCommandToHDD

;************************************************
;*     ЧТЕНИЕ ИДЕНТИФИКАТОРА ЖЕСТКОГО ДИСКА     *
;* Входные параметры передаются через глобальные *
;* переменные:                                   *
;* ChannelNumber – номер канала (1 – 4);         *
;* HDDNumber – номер диска на канале (0 или 1).  *
;* Идентификационный блок данных считывается     *
;* в массив SectorDataBuffer.                    *
;************************************************
PROC ReadHDD_ID near
        pushad
        push    ES
; Задать режим CHS
        mov     [ATAAddressMode],0
; Послать команду идентификации устройства
        mov     [ATAFeatures],0
        mov     [ATAHead],0
```

```
        mov     [ATACommand],0ECh
        call    SendCommandToHDD
        cmp     [HDDErrorCode],0 ;проверить код ошибки
        jne     @@End  ;закончить, сохранив код ошибки
; Ожидать готовность данных HDD
        mov     AX,0
        mov     ES,AX
        mov     DX,[HDDBasePortAddr]
        add     DX,7     ;адрес регистра состояния
@@WaitCompleet:
        ; Проверить время выполнения команды
        mov     EAX,[ES:046Ch]
        sub     EAX,[HDDTime]
        cmp     EAX,MaxHDDWaitTime
        ja      @@Error1 ;ошибка тайм-аута
        ; Проверить готовность
        in      AL,DX
        test    AL,80h   ;состояние сигнала BSY
        jnz     @@WaitCompleet
        test    AL,1     ;состояние сигнала ERR
        jnz     @@Error6
        test    AL,08h   ;состояние сигнала DRQ
        jz      @@WaitCompleet
; Принять блок данных от контроллера
        mov     AX,DGROUP
        mov     ES,AX
        mov     DI,offset SectorDataBuffer
        mov     DX,[HDDBasePortAddr] ;регистр данных
        mov     CX,256   ;число считываемых слов
        rep     insw     ;принять блок данных
        jmp short @@End
; Записать код ошибки
@@Error1:
        mov     [HDDErrorCode],1
        jmp short @@End
@@Error6:
        mov     [HDDErrorCode],6
@@End:  pop     ES
        popad
        ret
ENDP ReadHDD_ID
ENDS
```

5.2.3. Риск потери информации, связанный с выполнением операций форматирования и записи данных

Непосредственная работа с контроллером дисков чаще всего применяется для записи или считывания данных в режиме реального времени — например, при проведении экспериментов или обработке видеоизображений. Режим реального времени предполагает непрерывную передачу информации — система должна успевать обрабатывать каждый пакет данных до поступления следующего пакета (иначе один из пакетов будет потерян). При работе с дисковыми накопителями непрерывность передачи информации может нарушаться при периодической термокалибровке привода головок, при выполнении процедуры самотестирования, либо в результате фрагментации файлов. Для работы в реальном времени, вообще говоря, выпускаются специальные жесткие диски (например, мультимедийные цифровые видеомагнитофоны), но можно попробовать использовать и обычные диски с большим объемом буферной памяти, которая позволяет замаскировать термокалибровку и самотестирование. Альтернативным вариантом является создание буфера данных большого объема в оперативной памяти компьютера.

Поскольку непосредственная работа с контроллером резко увеличивает риск разрушения данных на диске при отладке программы в результате ошибок программиста, желательно максимально изолировать от остальных данных тот участок, с которым идет работа напрямую через регистры контроллера. Идеальным, но иногда недостижимым (по финансовым причинам) вариантом является выделение для работы в реальном времени отдельного жесткого диска; если невозможно выделить целый диск, то выделяется раздел диска, обычно — в конце адресного пространства (чтобы уменьшить риск стирания последующих разделов при ошибке адресации). В обоих случаях вместо стандартной разметки диска по методу FAT можно применить свою собственную, упрощенную организацию данных.

Непосредственная работа на уровне контроллера с обычными файлами произвольного типа и размера, произвольным образом размещенными на диске требует исключительной аккуратности. Программист в этом случае должен очень хорошо знать все особенности файловой системы, которую он использует; прежде чем начинать отладку процедур для записи данных на диск, он должен хорошо освоить адресацию данных в режиме считывания. Ошибки адресации опасны тем, что могут постепенно

разрушать информацию на диске, оставаясь при этом незамеченными в течение длительного времени. Если данные непременно должны быть записаны в виде файла в стандартный DOS-раздел, то, опять-таки, можно выделить для него неперемещаемую область (для чего файл нужно пометить как системный), желательно — фиксированного размера и в конце раздела, файл лучше при этом записывать в корневой каталог.

Выполнение операций низкоуровневого форматирования связано с очень серьезным риском. Не применяйте команды форматирования жестких дисков ни из набора прерываний DOS и BIOS, ни из набора ATA. Ни в коем случае не запускайте процедуру низкоуровневого форматирования из BIOS SETUP. Каждая фирма — изготовитель дисков выпускает для них свои собственные утилиты, следовательно, фактически отсутствует единый стандарт выполнения операций форматирования.

В нашей стране компьютеры обычно эксплуатируются до полного физического износа, то есть до тех пор, пока ремонт становится невозможным. Жесткий диск заканчивает свой жизненный путь двумя способами — либо сгорает аппаратура (мотор или встроенный контроллер), либо изнашивается поверхность диска и появляется много дефектных секторов. В последнем случае диск окончательно погибает, когда кто-нибудь запускает процедуру низкоуровневого форматирования из BIOS SETUP или универсальную форматирующую программу. Отформатированный такими способами диск часто не поддается восстановлению даже при помощи дисковых утилит фирмы-изготовителя. Жесткие диски в последнее время сильно подешевели, и если диск сильно изношен, не тратьте время на восстановление, а просто замените его. Кроме того, информация, содержащаяся на диске большого объема, стоит обычно гораздо больше самого диска — хранить данные на ненадежных дисках экономически неэффективно (если не верите, то попробуйте подсчитать трудозатраты на одну только запись информации на диск).

Все основные поставщики жестких дисков предоставляют бесплатно через Интернет программы для обслуживания своих изделий, в том числе — для низкоуровневого форматирования. Обычно эти программы размещаются на основном сервере или сервере технической поддержки:

- Fujitsu — http://www.fujitsu.com или http://www.fujitsu.com.tw;
- IBM — http://www.storage.ibm.com;
- Maxtor — http://www.maxtor.com;
- Quantum — http://www,quantum.com;
- Samsung — http://www.samsung.com;
- Seagate — http://www.seagate.com;
- Western Digital — http://www.wdc.com.

Контрольные вопросы

1. В чем преимущество перенесения контроллера диска на дисковод?
2. Что означают аббревиатуры: IDE, ATA и ATAPI?
3. Какие сигналы системной шины непосредственно используются на ленточном кабеле интерфейса ATA?
4. Как выбираются устройства на ленточном кабеле ATA?
5. Где производится преобразование внешних адресов во внутренние трехмерные адреса?
6. Какие компоненты фигурируют в спецификации ATA?
7. Как можно задавать ведущее (Master) и ведомое (Slave) устройства на ленточном кабеле ATA?
8. Может ли хост-адаптер ATA начав работу с одним устройством на плоском кабеле, переключиться на работу с другим устройством на этом же кабеле?
9. Может ли устройство IDE (ATA) не иметь внутреннюю буферную память?
10. Дайте краткую характеристику разновидностям интерфейса ATA (IDE).
11. Какие из этих интерфейсов электрически совместимы?
12. Что можно сказать о их логической совместимости?
13. Кто занимается разработкой спецификаций ATA, ATAPI и каково состояние по стандартизации на 2000 год?
14. Каковы электрические характеристики интерфейсного кабеля ATA?
15. На каких плоских кабелях можно подключать хост-адаптер и устройства к любым разъемам?
16. Какой кабель рекомендуется применять в режиме Ultra DMA и каковы его характеристики и разновидности?
17. Какие разъемы ATA используются и где?
18. Охарактеризуйте основные сигналы интерфейса ATA.
19. Какие линии интерфейса ATA получают новое назначение в режиме Ultra DMA?
20. Перечислите набор регистров контроллеров устройств ATA. Сколько блоков регистров используется?
21. Когда можно производить запись в эти регистры?
22. Охарактеризуйте регистры управления устройством (DC) и адреса устройства (DA).
23. Охарактеризуйте регистры данных(DR), ошибок (ER) и свойств (FR).
24. Охарактеризуйте регистры счетчика секторов (SC), номера сектора (SN), номера цилиндра (CL)

и номера устройства/головки (D/H).

25. Охарактеризуйте регистры состояния (SR) и команд (CR).
26. Что является адресуемой единицей в устройствах ATA?ATAPI?
27. Опишите компоненты трехмерного адреса сектора (внешняя геометрия). Какой объем памяти можно адресовать таким образом?
28. Что такое "Геометрия по умолчанию"?
29. Что такое "Логическая адресация" и как адрес LBA связаны с адресом CHS?
30. Где можно найти информацию о количестве секторов, доступных в режиме LBA?
31. Почему объем диска может быть разным в системах LBA и CHS?
32. Что такое "Весящие сектора" (orphan sectors) и каково минимальное их количество в режиме LBA?
33. Как используется CHS и LBA в различных стандартах ATA?
34. Как определяется геометрия по умолчанию в ATA-3 и ATAPI-4,5,6?
35. Что определяют слова 54-56 и 60-61 паспорта диска?
36. Какую геометрию сообщает диск объемом более 8.4 Гбайт?
37. Сколько команд используется в интерфейсе ATA?
38. Какие коды команд выделены производителям для реализации специальных команд?
39. Какие команды объявлены устаревшими в спецификации ATA/ATAPI-5 и какие удалены из списка?
40. Какие команды можно отнести к основным?
41. Для каких устройств используется команда Packet?
42. Охарактеризуйте команды чтения сектора.
43. Охарактеризуйте команды записи секторов.
44. Охарактеризуйте команды обмена с буфером диска, форматирования, поиска и рекалибровки.
45. Какие способы сброса в исходное состояние воспринимают устройства ATA и ATAPI?
46. Какую сигнатуру содержат устройства ATA и ATAPI в блоке командных регистров после выполнения сброса или команды диагностики?
47. Опишите команду идентификации (Identify Device).
48. Где может храниться паспорт диска?
49. В каком порядке передаются числовые и символьные данные паспорта и каков его размер в байтах?
50. Какая информация содержится в первых 20 словах паспорта?
51. Опишите команду установки параметров.
52. Охарактеризуйте команду установки свойств. Сколько подкоманд предусмотрено для этой команды?
53. Какие режимы обмена могут быть установлены подкомандой 03h?
54. Как выполняется команда диагностики и команда загрузки микрокода?
55. Что нужно учитывать при работе со сменными носителями?
56. Какой порядок работы со сменными носителями определен в ATA-4?
57. Какие команды ATA-2 предназначены для работы со сменными носителями?
58. Какие команды в ATA-4 предназначены для работы с флэш-памятью?
59. Как идентифицируются устройства, поддерживающие команды CFA?
60. Перечислите состояния энергосбережения в порядке возрастания энергопотребления?
61. Перечислите команды, используемые при управлении энергопотреблением?
62. Что понимается под расширенным управлением энергопотреблением APM?
63. В какой спецификации ATA/ATAPI предусмотрено управление акустическими шумами и какие команды для этого предусматриваются?
64. В каких состояниях могут находиться устройства ATA с точки зрения их защиты?
65. Как устанавливаются устройства ATA в эти состояния защиты?
66. Сколько паролей и условий защиты поддерживает система защиты?
67. Опишите команду установки защиты и пароля (Security Set Password).
68. Где записан код главного пароля и какое значение ему присваивает изготовитель?
69. Какие команды работают с системой защиты?
70. Какие дополнения в области защиты введены в ATA-4 и в ATA/ATAPI-5+?
71. Для чего служит технология SMART и как она может осуществляться в дисководах?
72. Где сохраняются значения атрибутов, за которыми ведется наблюдение?
73. Какие подкоманды команды SMART поддерживаются непакетными устройствами?
74. Для чего используется потоковое расширение команд и в какой спецификации оно поддерживается?
75. Опишите протокол взаимодействия хоста с устройством.
76. Как выполняется программный доступ PIO?
77. Как можно "притормозить" обмен в режиме PIO?

78. Какие режимы передачи PIO поддерживаются в ATA?
79. Какие режимы передачи DMA поддерживаются в ATA?
80. С какой спецификации ATA/ATAPI начал поддерживаться Ultra DMA?
81. Какие режимы Ultra DMA поддерживаются в ATA и как они выбираются?
82. Как осуществляется обмен в режиме Ultra DMA?
83. Чем определяется режим обмена и как он устанавливается?
84. Для реализации каких режимов обмена требуется 80-проводный кабель и для каких режимов применим 40-проводный кабель с двумя разъемами?
85. Какие возможности существуют для определения типа кабеля?
86. Как производится перекрытие команд в ATA-4 и какие команды допускают перекрытие?
87. Очереди каких команд допускаются и как определяется глубина очереди для конкретного устройства?
88. Для подключения каких устройств используется пакетный интерфейс ATAPI?
89. Каким способом расширяется набор команд ATA для ATAPI?
90. Какова основные особенности устройств ATAPI?
91. Что делает устройство, получив команду Packet?
92. Какие варианты состояния устройства может прочитать хост после выполнения устройством команды Packet?
93. Какова структура командного пакета?
94. Что представляет собой простейший адаптер ATA?
95. Какие системные ресурсы требует ATA и какие ресурсы закреплены за интерфейсами первого, второго, третьего и четвертого каналов?
96. Что содержат двухканальные адаптеры ATA и что предусмотрено им иметь спецификацией ATA/ATAPI-6?
97. Охарактеризуйте варианты адаптеров ATA.
98. Сколько каналов ATA ищут современные BIOS на стадии POST?
99. Какие диски IDE относятся к неинтеллектуальным?
100. Какие устройства относятся к Intelligent ATA IDE?
101. Охарактеризуйте устройства Intelligent Zoned Recording IDE?
102. Что понимается под конфигурированием устройств ATA?
103. Опишите два способа задания адреса дисков ATA.
104. Какие джамперы могут встречаться на устройствах IDE?
105. Охарактеризуйте программу, представленную листингом 5.1.
106. Какие предосторожности необходимо учитывать при работе с дисками через регистры?
107. Охарактеризуйте ограничения на низкоуровневое форматирование дисков ATA.
108. Где можно найти программы по обслуживанию дисков и их низкоуровневому форматированию?

В разделе использована информация из [2,3,4]. Дополнительные сведения о конструктивных особенностях дисковых накопителей ATA, по их классификации, сферах применения и пр. можно найти в [2,3,6,7].

5.3. Диски с интерфейсом SCSI

Шина SCSI, в отличие от вышеописанных интерфейсов, не является специализированным интерфейсом для дисковых устройств. Эта шина системного уровня. Наиболее распространенные современные SCSI-диски отвечают спецификациям SCSI-2, Fast SCSI-2, Fast-Wide SCSI-2 (16 бит), Ultra-Wide SCSI-2. Есть устройства и с оптоволоконным интерфейсом Fiber Channel, появившемся в спецификации SCSI-3.

Накопители с интерфейсом SCSI относятся к устройствам со встроенным контроллером. В настоящее время «выжил» и развивается только этот класс устройств, к которому относятся также и устройства с интерфейсом ATA. Отдельно взятые устройства с интерфейсами SCSI и ATA имеют близкие параметры, многие накопители с одним и тем же блоком HDA выпускаются в версиях SCSI и ATA. У таких устройств-близнецов будут практически совпадать такие параметры, как емкость, время доступа и скорость обмена данных. Однако в целом дисковая система с интерфейсом SCSI имеет ряд отличий, которые при определенных условиях будут обеспечивать их заметные преимущества (кроме цены) перед ATA-устройствами. Остановимся на них подробнее.

Для сопряжения SCSI-устройств с компьютером требуется довольно сложный *хост-адаптер,* который, в отличие от возможно и примитивного адаптера ATA, всегда должен обладать некоторым интеллектом. Хост-адаптер SCSI общается с устройствами своей шины, независимо от центрального процессора компьютера; он, как и сами устройства, является равноправным абонентом шины. Более того, устройства SCSI могут обмениваться данными между собой не выходя на системную шину компьютера, минуя его оперативную память. За эти не всегда используемые возможности приходится расплачиваться высокой ценой хост-адаптера, которая может быть соизмеримой со стоимостью накопителя. Есть и очень сложные и дорогие адаптеры, имеющие собственную кэш-память, исчисляемую мегабайтами, и мощный встроенный процессор. Такие адаптеры применяют в компьютерах-серверах, у которых требования к производительности дисковой системы высоки. Есть и совсем простые и дешевые хост-адаптеры, но они могут заметно ухудшить показатели производительности даже одиночного накопителя.

Интерфейс шины SCSI позволяет подключать к одному каналу хост-адаптера значительное число устройств: до 7 в «узком» (Narrow) варианте и до 15 в «широком» (Wide) варианте шины. Причем в отличие от «вымученного» подключения двух устройств к одной шине ATA, все устройства SCSI равноправны и могут выполнять операции независимо друг от друга и параллельно.

Конфигурирование дисковых устройств для BIOS специфично: утилитой BIOS Setup вместо типа диска указывается None. Если для каждого диска ATA в Setup необходимо описать параметры, а неописанные диски восприниматься не будут, то BIOS SCSI-адаптера подключит к системе все диски, имеющиеся в данный момент на его шине без предварительных описаний. В связи с этим хост-адаптеры SCSI всегда имеют расширения BIOS, необходимое для подключения дисков до загрузки операционной системы. Если в систему установлено несколько однотипных хост-адаптеров, дополнительный BIOS обычно включают только на одном из них, и он обеспечивает доступ к дискам всех адаптеров.

Для всех устройств SCSI принята *линейная адресация блоков,* 32-битный адрес 512-байтного блока позволяет иметь емкость устройств до 2 Тбайт. Правда, для совместимости с традиционными вызовами дискового сервиса SCSI-BIOSзаставляют придумывать фиктивную трехмерную геометрию. Но операционные системы, ориентированные на логическую адресацию, спрямляют этот зигзаг трансляций. Для устройств ATA-2 логическая адресация возможна, но не является обязательной.

Система команд SCSI ориентирована на многозадачное использование устройств. В SCSI-2 предусмотрена возможность создания очереди команд в самом устройстве. Получив свои цепочки команд, устройства выполняют их одновременно, независимо друг от друга. Более того, современные устройства способны внутренне изменять последовательность выполнения операций с целью оптимизации производительности. Получив, например, несколько запросов на чтение блоков, разбросанных по поверхностям носителя, «умный» встроенный контроллер соберет их в буферную память в том порядке, при котором минимальны перемещения головок. Раньше такая оптимизация если и выполнялась, то на уровне операционной системы (так называемый «лифтовый поиск» — Elevator Seeking в NetWare). Таким образом, в SCSI многозадачность проникает уже и внутрь устройств.

Производительность SCSI-дисков в системах с однозадачными ОС может уступать производительности аналогичных дисков с интерфейсом ATA за счет сложной процедуры инициирования обмена по SCSI-шине. Однако в системах с многозадачными ОС, особенно при наличии нескольких периферийных устройствах на шине, суммарная производительность системы со SCSI-2 будет существенно выше (при условии применения мощного хост-адаптера). Данные могут храниться в буферах устройств SCSI и передаваться по разделенной SCSI-шине по мере ее освобождения. При копировании данных может использоваться и обмен данными без участия CPU.

Обе шины — SCSI и ATA — имеют свои «ниши» преимущественного применения, однако устройства ATA используются только в IBM PC-совместимых компьютерах, в то время как SCSI используется на любых аппаратных платформах.

5.3.1. SCSI

Системный интерфейс малых компьютеров SCSI (Small Computer System Interface, произносится «скази») был стандартизован ANSI в 1986 году (X3.131-1986). Интерфейс предназначен для соединения устройств различных классов: памяти прямого (жесткие диски) и последовательного доступа (стриммеры), CD-ROM, оптических дисков однократной и многократной записи, устройств автоматической смены носителей информации, принтеров, сканеров, коммуникационных устройств и процессоров. *Устройством* SCSI(SCSI Device) называется как *хост-адаптер,* связывающий шину SCSI с какой-либо внутренней шиной компьютера, так и *контроллер целевого устройства* (target controller), с помощью которого устройство подключается к шине SCSI. С точки зрения шины все устройства могут быть равноправными и являться как *инициаторами обмена* (инициализирующими устройствами, ИУ), так и *целевыми устройствами* (ЦУ), однако чаще всего в роли ИУ выступает хост-адаптер. К одному контроллеру может подключаться несколько ПУ, по отношению к которым контроллер может быть как внутренним, так и внешним. Широкое распространение получили ПУ со встроенным контроллером SCSI (embeded SCSI controller), к которым относятся накопители на жестких магнитных дисках, CD-ROM, стриммеры. Каждое ЦУ может содержать до 8 независимо адресуемых логических устройств (ЛУ) со своими номерами LUN (Logical Unit Number), представляющих ПУ или их части.

По физической реализации интерфейс является 8-битной параллельной шиной с тактовой частотой 5 МГц. Шина допускает подключение до 8 устройств, скорость передачи данных в первоначальной версии достигала 5 Мбайт/с. Впоследствии (1991 г.) появилась спецификация — SCSI-2 (X3.131-1994), расширяющая возможности шины. Тактовая частота шины *Fast SCSI-2* достигает 10 МГц, а *Ultra SCSI-2* — 20 МГц. Разрядность данных может быть увеличена до 16 бит — эта версия называется *Wide SCSI-2* (широкий), а 8-битную версию стали называть *Narrow* (узкий). 16-битная шина допускает 16 устройств. Стандарт SCSI-2 определяет и 32-битную версию интерфейса, но стоимость интерфейса стала бы неоправданно высокой. Спецификация SCSI-2 определяет систему команд, которая включает набор базовых команд CCS (Common Command Set), обязательных для всех ПУ, и специфических команд для периферии различных классов. Стандарт полностью описывает протокол взаимодействия устройств, включая структуры передаваемой информации. Поддержка устройствами исполнения цепочек команд (до 256 команд) и независимость работы устройств друг от друга обуславливают высокую эффективность применения SCSI в многозадачных системах. Возможность присутствия на шине более одного контроллера (инициатора обмена) позволяет обеспечить разделяемое использование периферии несколькими компьютерами, подключенными к одной шине.

SCSI-3 — дальнейшее развитие стандарта, направленное на увеличение количества подключаемых устройств, расширение системы команд и поддержку Plug and Play. В качестве альтернативы параллельному интерфейсу SPI (SCSI-3 Parallel Interface) появляется возможность применения последовательного, в том числе волоконно-оптического интерфейса со скоростью 100 Мбайт/с. SCSI-3 существует в виде широкого спектра документов, определяющих отдельные аспекты интерфейса. Архитектурная модель SAM (SCSI-3 Architecture Model) изображена на рис. 5.5. Первичный набор общих команд SPC (SCSI-3 Primary Commands) для устройств различных классов дополняется набором команд соответствующего класса устройств:

- SBC (SCSI-3 Block Commands) — для устройств памяти прямого доступа;
- SSC (SCSI-3 Stream Commands) — для устройств памяти последовательного доступа;
- SGC (SCSI-3 Graphic Commands) — для принтеров и сканеров;
- SMC (SCSI-3 Medium Changer Commands) — для устройств смены носителей;
- SCC (SCSI-3 Controller Commands) — для хост-контроллеров.

Транспортный уровень может использовать различные протоколы с соответствующей поддержкой *физических соединений:*

- SPI (SCSI Parallel Interface) — параллельный интерфейс (разъемы, сигналы);
- SIP (SCSI-3 Interlocked Protocol) — протокол обмена традиционного интерфейса, физически реализуемый интерфейсом SPI;
- FCP (Fiber Channel Protocol) — протокол оптоволоконного канала с соответствующим физическим уровнем FC-PH со скоростью передачи данных 100 Мбайт/с;
- SBP (Serial Bus Protocol) — протокол последовательной шины, реализуемый интерфейсом 1394 (FireWire);
- GPP (Generic Packetized Protocol) — обобщенный пакетный протокол, реализуемый любым пакетным интерфейсом;
- SSP (Serial Storage Protocol) — последовательный протокол памяти, реализованный на архитектуре последовательной памяти SSA (Serial Storage Architecture).

К примеру, дисковый накопитель SCSI-3 с параллельным интерфейсом описывает набор стандартов SPI+SIP+SAM+SPC+SBC, а для того же устройства, но с последовательным интерфейсом вместо

SPC+SBC будет связка FCP+FC-PH.

Современные устройства с интерфейсом SCSI выпускаются в соответствии со стандартом SCSI-2 или SCSI-3. В отличие от стандарта SCSI-2, который явно подразумевал более высокую производительность шины и устройств, чем SCSI-1, заявка о поддержке устройством стандарта SCSI-3 непосредственно на повышение производительности не указывает. Эта заявка прямо означает лишь соответствие новому поколению документов. Однако в связи с общей тенденцией к росту производительности устройства SCSI-3 в большинстве случаев показывают более высокую производительность, чем SCSI-2. Стандарт SCSI-3 предполагает различные варианты протокольного и физического уровня интерфейса, включающие как параллельные, так и последовательные шины.

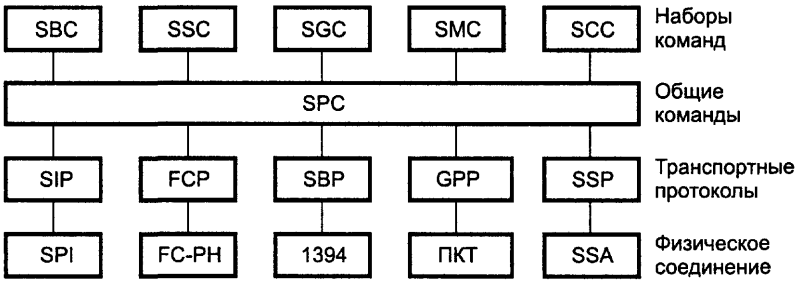

Рис. 5.5. Архитектурная модель SCSI-3

Для *параллельных шин* скорость передачи данных определяется *частотой передач,* измеряемой в миллионах передач в секунду — MT/s/ (Mega Transfer/sec), *и разрядностью.* Изначально разрядность шины SCSI составляла 8 бит *(Narrow),* «широкий» *(Wide)* вариант шины, появившийся в SCSI-2, имеет разрядность 16 бит (32-битные расширения не распространены). Комбинации частоты и разрядности обеспечивают широкий диапазон пропускной способности (табл. 5.18), достигающей уже 320 Мбайт/с. В обозначениях пропускной способности интерфейсов встречаются разночтения, здесь приводятся названия, используемые фирмой Western Digital в 2000 году. *Fast SCSI* означает частоту передач 10 MT/c, временные диаграммы для такого режима определены в SCSI-2. Краткое название *Fast-20* соответствует полному «Fast Wide SCSI» (16 бит, 10 MT/s). Режим *Ultra SCSI* указывает на частоту передач 20 MT/c, он определен для параллельного интерфейса в SCSI-3. Краткое название *Fast-40* соответствует полному «Wide Ultra SCSI» (16 бит, 20 MT/s). Режим *Ultra2 SCSI* указывает на частоту передач 40 MT/c, краткое название *Fast-80* соответствует полному «Wide Ultra2 SCSI» (16 бит, 40 MT/s). Этот режим, определенный в SCSI-3, в настоящее время является самым распространенным для новых устройств с параллельной шиной. Он реализован только в низковольтной дифференциальной версии интерфейса — LVD. В SCSI-3 понятие Ultra3 SCSI довольно широкое. Ultral60 SCSI означает скорость 160 Мбайт/с, применяется только в «широком» (16 бит) варианте. Здесь применяется синхронизация по обоим фронтам сигнала, а также контроль достоверности передач по CRC-кодам, что позволяет «выжать» из кабеля максимальную скорость передачи (как и в Ultra DMA интерфейса ATA). В 2001 году ожидается выход в свет интерфейса Ultra320 SCSI со скоростью 320 Мбайт/с. Эту спецификацию (SPI-4) разрабатывают фирмы Adaptec, LSI Logic и Seagate, они уже продемонстрировали образец контроллера и винчестера с этим интерфейсом.

Таблица 5.18. Скорость передачи данных по параллельной шине SCSI

Разрядность шины, бит	Разновидность				
	Обычный	Fast	Fast-20 (Ultra)	Fast-40 (Ultra2)	Fast-80 (Ultra3)
8 (Narrow) 16 (Wide) 32 (Wide)[1]	5 Мбайт/с 10 Мбайт/с 20 Мбайт/с[1]	10 Мбайт/с 20 Мбайт/с 40 Мбайт/с[1]	20 Мбайт/с 40 Мбайт/с 80 Мбайт/с[1]	40 Мбайт/с 80 Мбайт/с 160 Мбайт/с[1]	80 Мбайт/с 160 Мбайт/с

[1] Реализации не встречаются.

Наиболее популярный интерфейс *Ultra2 SCSI обеспечивает* прекрасное сочетание пропускной способности шины при ее большей длине, цены устройств и совместимости с традиционными устройствами SCSI. Здесь используется дифференциальная передача сигналов, но с низким уровнем напряжения. В настоящее время традиционный дифференциальный интерфейс получил название «высоковольтный» — HVD (High Voltage Differential), поскольку в SCSI-3 ему появилась низковольтная альтернатива — LVD (Low Voltage Differential). Низковольтный вариант позволяет достичь частоты передачи 40 MT/sec при длине шины до 25 м (до 8 устройств) или до 12 м (до 16 устройств). Новые устройства с интерфейсом LVD могут иметь возможность работы на шине вместе с устройствами с

линейным SE (Single Ended) интерфейсом — для этого их буферные схемы содержат автоматический определитель типа интерфейса. Однако эта совместимость относится только к LVD — традиционные устройства с HVD могут работать только с себе подобными.

Последовательный интерфейс *Fibre* (Fiber) *Channel,* или *FCAL* (Fibre Channel Arbitrated Loop — кольцо волоконного канала с арбитражем), занимает промежуточное положение между интерфейсами периферийных устройств (SCSI-3) и технологиями локальных сетей. Этот интерфейс может иметь как электрическую (коаксиальный кабель), так и оптоволоконную реализацию. В обоих случаях частота 1 ГГц обеспечивает скорость передачи данных 100 Мбайт/с. Медный кабель допускает длину шины до 30 м, оптический — до 10 км. Здесь используется иной протокольный и физический уровни интерфейса и имеется возможность подключения к шине до 126 устройств (а не 8 или 16, как для параллельного интерфейса). Для двухточечного соединения возможен полнодуплексный режим (200 Мбайт/с), что невозможно в обычных параллельных шинах. Недавно фирма Adaptec выпустила адаптер со скоростью 2 Гбит/с (и оптика, и медь), обратно совместимый с обычным (1 Гбит/с). В полнодуплексном режиме достигается суммарная пропускная способность 400 Мбайт/с. В кольцо может объединяться до 126 узлов, длина кольца может достигать 10 км. По организации кольцо напоминает FDDI — все узлы собираются в замкнутую цепочку и транслируют приходящие кадры дальше по кольцу. Синхронизация передатчика каждого узла автономна, а для компенсации расхождения частот синхронизации используются межкадровые слова-заполнители, часть из которых может периодически отбрасываться или вводиться дополнительно при трансляции. Для обеспечения надежной передачи применяется кодирование 8B/10B, для скорости 100 Мбайт/с с учетом накладных расходов на обрамление кадров требуется битовая скорость в линии 1,0625 Гбит/с. Архитектурная модель FCAL состоит из пяти уровней FC-0 … FC-4, нижний (FC-0) определяет среду передачи (оптоволокно или твинаксиальный кабель) и физический интерфейс. Верхний уровень (FC-4) определяет протоколы отображения, относящиеся как к интерфейсам периферийных устройств (SCSI и некоторые другие), так и к сетям (802.2 и IP). Информация по кольцу передается кадрами размером 36-2148 байт. Обмен данными между устройствами возможен как с установлением соединений, так и без них. Одновременно может быть открыто множество соединений, причем они могут относиться к разным протоколам (например, SCSI и IP). Аппаратура Fiber Channel включает интерфейсные адаптеры, концентраторы, коммутаторы и маршрутизаторы. *Интерфейсные адаптеры* представляют собой карты для компьютеров (для высокопроизводительных шин, например PCI). С интерфейсом FCAL выпускаются устройства хранения данных (дисковые и ленточные накопители, массивы накопителей). *Концентраторы* для FCAL в принципе необязательны, но они позволяют организовать кольцо на звездообразной топологии и обеспечить обход отказавших (отключенных) узлов — без них кольцо становится уязвимым при отказе линии или устройства. *Коммутаторы* (как, например, в технологии Ethernet) позволяют для группы подключенных устройств организовывать соединения «каждый с каждым», что эффективно, например, в случае разделяемого использования несколькими серверами нескольких устройств хранения. *Маршрутизаторы,* или *мосты,* позволяют соединять FCAL с другими средами передачи информации (например, с классическим интерфейсом SCSI или со средами локальных сетей). В настоящее время FCAL применяется для подключения устройств внешней памяти к серверам, когда требуется высокая производительность и значительное удаление устройств друг от друга. В принципе, FCAL позволяет организовывать разделяемое использование ресурсов, обеспечивая и резервирование линий связи, но здесь пока имеются сложности на уровне операционных систем.

В настоящее время наибольшее распространение имеют устройства SCSI-2 и SCSI-3 с параллельным интерфейсом, которые сохраняют совместимость с исходной версией, теперь называемой SCSI-1. Однако смешивать устройства SCSI-1 и SCSI-2 неэффективно, да и не всегда возможно из-за проблем, о которых речь пойдет ниже. Для краткости номер версии SCSI будем опускать, по умолчанию подразумевая Narrow SCSI-2/3. На ее примере разберем работу интерфейса, а особенности версии Wide отметим отдельно.

5.3.1.1. Параллельный интерфейс SCSI

Стандарт SCSI-1, разработанный в середине 80-х годов, определяет физический и протокольный уровень шины. Шина имеет разрядность 8 бит, линейные (SE — Single Ended) формирователи сигналов с открытым коллектором, приемники и передатчики на биполярных транзисторных схемах, обязательны пассивные терминаторы. Все передачи происходят только в асинхронном режиме. Разъемы с низкой плотностью контактов. Частота передачи до 5 MT/s, что обеспечивает скорость передачи до 5 Мбайт/с.

Стандарт SCSI-2 также определяет и физический, и протокольный уровень параллельного интерфейса, с разрядностью 8 или 16 бит. К асинхронному режиму добавлена возможность передачи данных в синхронном режиме, со скоростью до 10 MT/s (Fast 10), или 20 Мбайт/с в «широкой» версии. В линейном варианте (SE) применяются передатчики на биполярных транзисторах, по схеме с открытым коллектором или с активным снятием сигнала (active negation). Схема с открытым коллектором для снятия сигнала просто «отпускает» линию, и ее потенциал возвращается в исходное состояние только под действием терминаторов. При активном снятии схема форсирует переход линии к потенциалу

пассивного состояния, при этом создается даже иллюзия возможности работы без терминаторов. В дифференциальном варианте, который теперь называется высоковольтным дифференциальным HVD (High Voltage Differential), применяются передатчики EIA 485.

SCSI-3, представляющий собой набор документов, для параллельного интерфейса описывается документами SIP (SCSI Interlock Protocol) и SPI (SCSI Parallel Interface). *SIP* определяет набор команд параллельного интерфейса, он соответствует протоколу шины SCSI-2 и далее рассматривается наиболее подробно. Протоколы SIP соответствующих версий включены в стандарты SPI-2 и SPI-3.

В *стандарте SPI* (1995 г.) определен P-кабель и коннекторы для организации широкой (16 бит) шины на одном кабеле. Этот кабель с 68-контактными разъемами называют кабелем SCSI-3. SPI определяет скорость Fast SCSI (Fast Wide SCSI со скоростью 20 Мбайт/с). Интерфейсные схемы — КМОП (CMOS). Позже появилось дополнительная спецификация Fast-20, более известная как Ultra SCSI (Ultra Wide SCSI со скоростью 40 Мбайт/с).

В стандарте SPI-2 (1999 г.) снова удвоена частота передачи, Fast-40 SCSI более известен как Ultra2 SCSI (Wide Ultra2 SCSI со скоростью передачи 80 Мбайт/с)! В качестве интерфейсных схем используются низковольтные дифференциальные приемопередатчики LVD. Стандарт SPI-2 описывает подключение через один разъем SCA-2 (Single Connector Attachment) с возможностью «горячей замены» (hot swap) и 68-контактный разъем с «очень большой» плотностью контактов VHDCI (Very High Density Connector). В стандарт SPI-2 включен и A-кабедь SCSI-2, и P-кабель SPI. Это законченный документ, не ссылающийся на предыдущий и описывающий все параллельные интерфейсы SCSI вплоть до .Fast-40.

В *стандарте SPI-3* (2000 г.) опять удвоена частота передачи, но уже за счет двойной синхронизации — Fast-80DT, известный как Ultra3 SCSI или Ultral60. Для данного режима рассматривается только широкий (16 бит) вариант. Традиционный (высоковольтный) дифференциальный вариант, а также 32-битная шина с Q-кабелем упразднены. Рассматривается только LVD-интерфейс, синхронизация по фронтам и спадам сигналов REQ#/ACK# (DT — Double Transition), вводится CRC-контроль передач, пакетированные команды и сообщения (Packetized Commands and Messaging) и быстрый арбитраж (Quick Arbitration). Это тоже законченный документ, описывающий все параллельные интерфейсы SCSI вплоть до Fast-80DT и отменяющий HVD, SCAM (SCSI Configured AutoMatically-автоматическое конфигурирование устройств) и 32-битную шину.

В *SPI-4* (планируется на 2001 г.) снова удваивается частота и получается Fast-160DT, уже известный как Ultra320 SCSI (только широкая шина, 320 Мбайт/с).

*EPI (*Enhanced Parallel Interface) — технический отчет, описывающий построение SCSI-систем. Определяются возможные длины кабелей, величина нагрузки, совместное использование широких и узких устройств, переходные адаптеры, мосты, коммутаторы и другие вещи, не описанные стандартами.

Физически «узкий» SCSI представляет собой шину, состоящую из 18 сигнальных и нескольких питающих цепей. В «широком» варианте число сигнальных цепей увеличено. Для защиты от помех каждая сигнальная цепь имеет свой отдельный обратный провод. На применяемых двухрядных разъемах контакты сигнальных и обратных цепей располагаются друг против друга. Это позволяет применять в качестве кабелей как витые пары проводов, так и плоские ленточные кабели, где сигнальные и обратные провода чередуются.

По типу сигналов различают *линейные* (single ended) и *дифференциальные* (differential) версии SCSI. Их кабели и разъемы идентичны, но электрической совместимости устройств нет. Символические обозначения для разных версий приведены на рис. 5.6.

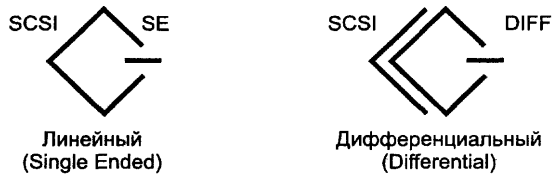

Рис. 5.6. Универсальные символические обозначения версий SCSI

В широко используемой *линейной* версии *SE* (Single Ended) каждый сигнал передается потенциалом с ТТЛ-уровнями относительно общего провода. Здесь общий (обратный) провод для каждого сигнала тоже должен быть отдельным, что снижает перекрестные помехи.

Дифференциальная версия *Diff* или *HVD* для каждой цепи задействует пару проводников, по которым передается парафазный сигнал. Здесь используются специальные дифференциальные приемопередатчики, применяемые и в интерфейсе RS-485, что позволяет значительно увеличить длину кабеля, сохраняя частоту обмена (табл. 5.19). Дифференциальный интерфейс применяется в дисковых системах серверов, но в обычных PC не распространен. Интерфейс HVD (но еще с названием Diff) появился в SCSI-2, а в стандарте SPI-3 (SCSI-3 1999 года) он уже упразднен, поскольку скорость Ultra2 и выше он уже не выдерживает.

Низковольтный дифференциальный интерфейс LVD позволяет работать на частотах 40,80 и 160

МТ/с в устройствах Ultra2, Ultral60 и Ultra320 SCSI при длине шины 25 м (8 устройств) или 12 м (16 устройств). Устройства LVD совместимы с устройствами SE благодаря возможности их автоматического переконфигурирования (Multimode LVD). Устройства LVD распознают напряжение на линии DIFFSENS и по низкому уровню напряжения на ней способны переключаться из режима LVD (дифференциальный) в SE (линейный). Контакт разъема, на который выводится эта цепь, в устройствах SE заземлен, что и обеспечивает автоматическое «понижение» режима всех устройств шины до SE, если имеется хотя бы одно устройство SE.

В качестве шины используется плоский или круглый гибкий кабель. *Плоский кабель* используется для соединения устройств, расположенных в одном корпусе. На нем может быть наколото несколько разъемов. При необходимости кабели могут сращиваться через специальные переходные разъемы. Кабели сращиваются только через концевые разъемы, Т-образные ответвления недопустимы (допускается длина отвода до 10 см, сюда входит длина проводника от ответвления до входа микросхемы приемопередатчика). *Круглый кабель,* состоящий из витых пар, используется для соединения вне корпусов устройств. ПУ внешнего исполнения обычно имеют два разъема, что позволяет соединить их в цепочку. Длина кабеля зависит от версии интерфейса и частоты (см. табл. 5.19). При подсчете суммарной длины кабеля следует учитывать возможность использования одного порта хост-адаптера одновременно для внешних и внутренних подключений и в случае такого подключения суммировать длины внутренних и внешних кабелей. У всех разъемов кабеля контакты одноименных цепей соединяются «один-в-один», назначение контактов см. в табл. 5.20 - 5.25.

В настоящее время ассортимент разъемов, применяемых в устройствах SCSI, довольно широк, что иногда заставляет использовать переходные адаптеры. Разъемы различаются как по числу, так и по форме и размеру контактов. Практически все разъемы двухрядные, и раскладка цепей рассчитана на чередование сигнальных и обратных проводов. Исключения составляют разъемы DB-25, у которых число «земляных» контактов меньше, чем сигнальных, и экзотические трехрядные DB-50. Применяются следующие типы разъемов:

- IDC-50 — разъемы для соединения внутренних устройств (аналогичны разъемам ATA, где применяются 40-контактные IDC-40). Разъемы имеют квадратные штырьковые контакты с шагом 0,1" (2,54 мм), пластмассовый корпус, без кожуха и дополнительных фиксаторов (рис. 5.7, *а).* На устройствах устанавливают *вилки* (IDC-50M), на ленточных кабелях — *розетки* (IDC-50F).
- CX-50 — разъемы типа Centronics, аналогичные применяемым в принтерах (но 50-контактные). Разъемы имеют пластинчатые контакты с шагом 0,085" (2,16 мм) и внешний металлический кожух (рис. 5.7, *б).* Применяются для соединения внешних устройств. На корпусе устройства (и SCSI-адаптера) *устанавливают розетки* (CX-50F), на кабелях — *вилки* (CX-50M). Разъемы фиксируются проволочными скобами, установленными на розетке, входящими в выемки на корпусе вилки. Часто называются «внешними SCSI-1» (SCSI-1 External).
- DB-25 — разъемы с круглыми штырьковыми контактами в металлическом кожухе D-образной формы (как на LPT-порте компьютера). На устройстве устанавливается *розетка* (DB-25F), на кабеле — *вилка* (DB-25M); фиксация с помощью винтов (рис. 5.7, б). Применяются на некоторых внешних устройствах (например, Zip).
- HD-50, они же MiniD50, — разъемы со штырьковыми контактами в металлическом кожухе D-образной формы, с высокой плотностью контактов (High Density) — с шагом 0,05" (1,27 мм), см. рис. 5.7, *г.* На устройстве устанавливается *розетка* (HD-50F), на кабеле — *вилка* (HD-50M); фиксация с помощью защелок-клипс. Часто называются «внешними SCSI-2» (SCSI-2 External).
- HD-68, они же MiniD68, — аналогичные разъемы, но с 68 контактами. На устройстве устанавливается *розетка* (HD-68F или MiniD68F), на кабеле — *вилка* (HD-68M или MiniD68M). Внешние разъемы фиксируются с помощью клипс или винтов, внутренние только на трении. Часто называются разъемами SCSI-3, в настоящее время наиболее широко используются для «широкого» интерфейса. На рис. 5.8 показан внешний разъем, слева изображена клипса, справа — резьбовая букса.
- VHDCI-68 — внешние разъемы с особо высокой плотностью (Very High Density Connector), контакты в стиле Centronics с шагом 0,8 мм. Применяются нечасто, иногда их ошибочно называют разъемами SCSI-4 или SCSI-5.
- MCX (Micro-centronics) — разъемы в стиле Centronics, но в миниатюрном варианте. Наиболее распространены разъемы MCX-68 и MCX-80, более известные как SCA.
- SCA (Single Connector Attachment) — разъем для подключения устройства одним разъемом. Предназначен для подключения дисков, устанавливаемых в шасси с возможностью «горячей» замены (или, по крайней мере, легкой, через лицевую панель). В настоящее время распространена спецификация SCA-2 на разъемах MCX-80 (рис. 5.9). На устройстве устанавливается *вилка* (MCX-80F), на шасси — *розетка* (MCX-80M). Кроме интерфейсных сигналов, на разъем выводятся шины питания, а также сигналы конфигурирования устройства (идентификатор, режимы и т. п.). На боковых направляющих имеются дополнительные контакты заземления. Конфигурационные

джамперы устанавливаются не на устройстве, а на шасси (или плате адаптера).

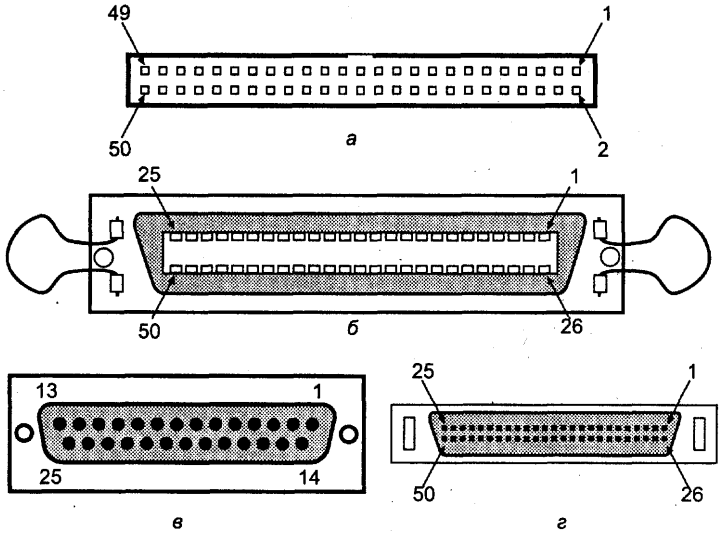

Рис. 5.7. Разъемы 8-битного устройства SCSI: а — IDC-50F; *б* — CX-50F; б — DB-25F; *г*— HD-50F

Для Narrow SCSI использовались разъемы, изображенные на рис. 5.7. Wide SCSI использует разъемы, изображенные на рис. 5.8. Для устройств с «горячей» заменой применяют миниатюрный D-образный разъем SCA-2, общий для питания и сигнальных цепей (рис. 5.9).

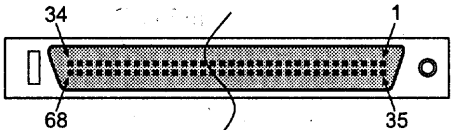

Рис. 5.8. Разъемы 16-битного устройства SCSI HD-68F

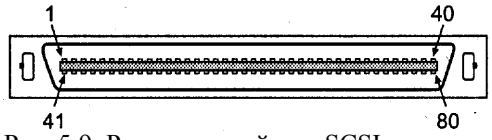

Рис. 5.9. Разъем устройства SCSI с «горячей» заменой SCA-80

Таблица 5.19. Максимальная длина кабелей SCSI

Тип интерфейса	Обычный (5 MT/sec)	Fast (10 MT/sec)	Ultra (20 MT/sec)	Ultra 2 (40 MT/sec)	Ultra 160 (80 MT/sec)	Ultra320 (160 MT/sec)
Линейный (Single ended)	6м	3м	1,5м (8 устр.) 3м (4 устр.)	-	-	-
Дифференциальный (HVD)	25м	12м (16 устр.) 25м (8 устр.)	6м (16 устр.) 25м (8 устр.)	-	-	-
Дифференциальный низковольтный (LVD)	-	-	-	12м (16 устр.) 25м (8 устр.)	12м (16 устр.) 25м (8 устр.)	12м (16 устр.) 25м (8 устр.)

Ассортимент кабелей SCSI довольно широк:

А-кабель: стандартный для 8-битного интерфейса, 25 пар проводов. Для внутренних устройств используется плоский ленточный кабель, для внешних — круглый кабель, состоящий из 25 витых пар в общем экране.

- Внутренний А-кабель SCSI-1 и SCSI-2 имеет разъемы с низкой плотностью контактов IDC-50 (розетки), см. рис. 5.7, а.
- Внешний А-кабель SCSI-1 имеет разъемы Centronics-50 (CX-50M), см. рис. 5.7, *б.*
- Внешний А-кабель SCSI-2 имеет разъемы MiniD50M (HD-50M), см. рис. 5.7, *в.*

B-кабель: малораспространенный 16/32-битный расширитель SCSI-2.

P-кабель: 8/16-битный кабель с 34 парами проводов, снабжен улучшенными миниатюрными экранированными разъемами. Применяется в интерфейсах SCSI-2/3, в 8-битном варианте контакты 1-5,31-39, 65-68 не используются.

- Внутренний P-кабель SCSI-3 имеет разъемы HD-68M без фиксаторов.
- Внешний P-кабель SCSI-3 имеет разъемы MiniD68M с фиксаторами.
- Внешний P-кабель SCSI SPI-2 имеет разъемы с особо высокой плотностью VHDCI-68M. Иногда такой кабель ошибочно называют кабелем SCSI-4 или SCSI-5.

Q-кабель: 68-проводное расширение до 32 бит, используется в паре с P-кабелем и имеет аналогичную конструкцию. Реально Q-кабель так и не использовался, в спецификации SCSI SPI-3 уже не рассматривается.

«Mac SCSI» — кабель с разъемами DB-25P (рис. 5.7, г) — 8-битный, стандартный для Macintosh (см. табл. 5.24), используется на некоторых внешних устройствах (Iomega ZIP Drive). Встречается иная раскладка цепей, если 25-контактный разъем установлен на хост-адаптере.

Существуют также различные варианты кабелей-переходников (с разнотипными разъемами) и адаптеров. Адаптеры представляют собой печатную плату или монолитную конструкцию с разнотипными разъемами. У адаптеров, соединяющих шины разной ширины, может присутствовать терминатор (должен быть отключаемым!) старшего байта (см. п. 5.23). Адаптер подключения SCA к обычной шине имеет стандартный разъем подключения питания, а также набор джамперов, задающих конфигурацию устройства.

Назначение контактов разъемов кабелей приведено в табл. 5.20 - 5.25. Неудобство вызывает система нумерации контактов, которая различна для внешних и внутренних разъемов. Однако физическая раскладка проводов на разъеме и в плоском кабеле одинакова, сигнальные линии (прямые) перемежаются обратными проводами (заземленными в SE-устройствах).

Таблица 5.20. Разъемы A-кабеля SCSI

Контакт IDC-50 / CX-50	Цепь SE/Diff	Контакт IDC-50/CX-50	Цепь
1/1	GND/DB0+	2/26	DB0#
3/2	GND/DB1+	4/27	DB1#
5/3	GND/DB2+	6/28	DB2#
7/4	GND/DB3+	8/29	DB3#
9/5	GND/DB4+	10/30	DB4#
11/6	GND/DB5+	12/31	DB5#
13/7	GND/DB6+	14/32	DB6#
15/8	GND/DB7+	16/33	DB7#
17/9	GND/DBP0+	18/34	DBP0#
19/10	GND/GND	20/35	GND
21/11	GND/GND	22/36	GND
23/12	Резерв	24/37	Резерв
25/13	Свободен	26/38	TERMPWR
27/14	Резерв	28/39	Резерв
29/15	GND	30/40	GND
31/16	GND/ATN+	32/41	ATN#
33/17	GND	34/42	GND
35/18	GND/BSY	36/43	BSY#
37/19	GND/ACK+	38/44	ACK#
39/20	GND/RST+	40/45	RST#
41/21	GND/MSG+	42/46	MSG#
43/22	GND/SEL+	44/47	SEL#
45/23	GND/C/D+	46/48	C/D#
47/24	GND/REQ+	48/49	REQ#
49/25	GND/I/O+	50/50	I/O#

Таблица 5.21. Разъемы B-кабеля SCSI

Контакт внутр. / внешн.	Цепь SE/Diff	Контакт внутр./ внешн.	Цепь
1/1	GND	2/35	GND
3/2	GND/DB8+	4/36	DB8#
5/3	GND/DB9+	6/37	DB9#
7/4	GND/DB10+	8/38	DB10#

Таблица 5.21. Разъемы B-кабеля SCSI (продолжение)

Контакт внутр. / внешн.	Цепь SE/Diff	Контакт внутр./ внешн.	Цепь
9/5	GND/DB11+	10/39	DB11#
11/6	GND/DB12+	12/40	DB12#
13/7	GND/DB13+	14/41	DB13#
15/8	GND/DB14+	16/42	DB14#
17/9	GND/DB15+	18/43	DB15#
19/10	GND/DBP1+	20/44	DBP1#
21/11	GND/ACKB+	22/45	ACKB#
23/12	GND/GND	24/46	GND
25/13	GND/REQB+	26/47	REQB#
27/14	GND/DB16+	28/48	DB16#
29/15	GND/DB17+	30/49	DB17#
31/16	GND/DB18+	32/50	DB18#
33/17	TERMPWR	34/51	TERMPWR
35/18	TERMPWR	36/52	TERMPWR
37/19	GND/DB19+	38/53	DB19#
39/20	GND/DB20+	40/54	DB20#
41/21	GND/DB21+	42/55	DB21#
43/22	GND/DB22+	44/56	DB22#
45/23	GND/DB23+	46/57	DB23#
47/24	GND/DBP2+	48/58	DBP2#
49/25	GND/DB24+	50/59	DB24#
51/26	GND/DB25+	52/60	DB25#
53/27	GND/DB26+	54/61	DB26#
55/28	GND/DB27+	56/62	DB27#
57/29	GND/DB28+	58/63	DB28#
59/30	GND/DB29+	60/64	DB29#
61/31	GND/DB30+	62/65	DB30#
53/32	GND/DB31+	64/66	DB31#
65/33	GND/DBP2+	66/67	DBP2#
67/34	GND/GND	68/68	GND

Таблица 5.22. Разъемы P-кабеля SCSI

Контакт	Цепь SE/Diff	Контакт	Цепь
1	GND/DB12+	35	DB12#
2	GND/DB13+	36	DB13#
3	GND/DB14+	37	DB14#
4	GND/DB15+	38	DB15#
5	GND/DBP1+	39	DBP1#
6	GND/DB0+	40	DB0#
7	GND/DB1+	41	DB1#
8	GND/DB2+	42	DB2#
9	GND/DB3+	43	DB3#
10	GND/DB4+	44	DB4#
11	GND/DB5+	45	DB5#
12	GND/DB6+	46	DB6#
13	GND/DB7+	47	DB7#
14	GND/DBP0+	48	DBP0#
16	GND	49	GND
16	DIFFSENS (GND)[1]	50	GND
17	TERMPWR	51	TERMPWR
18	TERMPWR	52	TERMPWR
19	Резерв	53	Резерв
20	GND	54	GND
21	GND/ATN+	55	ATN#
22	GND/GND	56	GND
23	GND / BSY+	57	BSY#
24	GND/ACK+	58	ACK#

Таблица 5.22. Разъемы P-кабеля SCSI (продолжение)

Контакт	Цепь SE/Diff	Контакт	Цепь
25	GND / RST+	59	RST#
26	GND/MSG+	60	MSG#
27	GND/SEL+	61	SEL#
28	GND/C#/D+	62	C#/D
29	GND/REQ+	63	REQ#
30	GND/I/O+	64	I/O#
31	GND/DB8+	65	DB8#
32	GND/DB9+	66	DB9#
33	GND/DB10+	67	DB10#
34	GND/DB11+	68	DB11#

[1] Сигнал DIFFSENS определен только для интерфейса LVD

Таблица 5.23. Разъемы Q-кабеля SCSI

Контакт	Цепь	Контакт	Цепь
1	GND	35	DB28#
2	GND	36	DB29#
3	GND	37	DB30#
4	GND	38	DB31#
5	GND	39	DBP3#
6	GND	40	DB16#
7	GND	41	DB17#
8	GND	42	DB18#
9	GND	43	DB19#
10	GND	44	DB20#
11	GND	45	DB21#
12	GND	46	DB22#
13	GND	47	DB23#
14	GND	48	DBP2#
15	GND	49	GND
16	GND	50	GND
17	TERMPWR	51	TERMPWRQ
18	TERMPWR	52	TERMPWRQ
19	Резерв	53	Резерв
20	GND	54	GND
21	GND	55	Терминирован
22	GND	56	GND
23	GND	57	Терминирован
24	GND	58	ACKQ#
25	GND	59	Терминирован
26	GND	60	Терминирован
27	GND	61	Терминирован
28	GND	62	Терминирован
29	GND	63	REQQ#
30	GND	64	Терминирован
31	GND	65	i DB24#
32	GND	66	DB25#
33	GND	67	DB26#
34	GND	68	DB27#

Таблица 5.24. Разъем Mac SCSI (DB-25)

Контакт	Цепь	Контакт	Цепь
1	REQ#	4	RST#
2	MSG#	5	ACK#
3	l/0#	6	BSY#
7	GND	17	ATN#
8	DB0#	18	GND
9	GND	19	SEL#

Таблица 5.24. Разъем Mac SCSI (DB-25) (продолжение)

Контакт	Цепь	Контакт	Цепь
10	DB3#	20	DBPO#
11	DB5#	21	DB1#
12	DB6#	22	DB2#
13	DB7#	23	DB4#
14	GND	24	GND
15	C/D#	25	TERMPWR
16	GND		

Таблица 5.25. Разъем SCA-80

Контакт	Цепь	Контакт	Цепь
01	12 Volt Charge	29	DBP1#
02	12 Volt	30	DB15#
03	12 Volt	31	DB14#
04	12 Volt	32	DB13#
05	Резерв	33	DB12#
06	Резерв	34	5 Volt
07	DB11#	35	5 Volt
08	DB10#	36	5 Volt Charge
09	DB9#	37	Резерв
10	DB8#	38	Auto Spin Up
11	I/O#	39	SCSI I D 0
12	REQ#	40	SCSI I D 2
13	C/D#	41	12VGND
14	SEL#	42	12VGND
15	MSG#	43	12VGND
16	RST#	44	Mated 1
17	ACK#	45	Резерв
18	BSY#	46	GND
19	ATN#	47	GND
20	DBP0#	48	GND
21	DB7#	49	GND
22	DB6#	50	GND
23	DB5#	51	GND
24	DB4#	52	GND
25	DB3#	53	GND
26	DB2#	54	GND
27	DB1#	55	GND
28	DB0#	56	GND
57	GND	69	GND
58	GND	70	GND
59	GND	71	GND
60	GND	72	GND
61	GND	73	GND
62	GND	74	Mated 2
63	GND	75	5VGND
64	GND	76	5VGND
65	GND	77	Active LED Out
66	GND	78	Delayed Start
67	GND	79	SCSI I D 1
68	GND	80	SCSI I D 3

Назначение сигналов раскрывает табл. 5.26. Все сигналы шины являются L-активными: активному состоянию и логической единице соответствует низкий потенциал, что отмечено символом «#» после мнемоники цепи. Обратные (парафазные) цепи обозначаются знаком «+».

На концах кабелей обязательно устанавливаются *терминаторы,* согласованные по импедансу с кабелем. Они предназначены для «подтягивания» уровня сигналов линий к высокому потенциалу. Терминаторы служат и для предотвращения отражения сигналов от концов кабеля. По исполнению терминаторы могут быть внутренние (размещенные на печатной плате устройства) и внешние

(устанавливаемые на разъемы кабеля или устройства). По электрическим свойствам различают пассивные, активные и FPT-терминаторы, подробнее о них см. в п. 5.3.1.5.3. Терминаторы требуют питания, для чего в интерфейсе имеются специальные линии TERMPWR, питание поступает от ИУ. В старых версиях SCSI терминаторы использовались для возврата уровня сигналов в пассивное состояние: устройства формировали только активный (низкий) уровень, а потом «отпускали» линии. Таким образом, без терминаторов старые устройства неработоспособны в принципе. У высокоскоростных устройств применяется активный возврат сигналов в пассивное состояние (active negation), этому процессу терминаторы не должны мешать. Из-за данного способа формирования сигналов возникает иллюзия возможности работы шины без терминаторов, но устойчивая работа при большом количестве устройств (более двух на шине, включая контроллер) проблематична.

Таблица 5.26. Назначение сигналов шины SCSI

Сигнал	Назначение
BSY#	Busy — шина занята
SEL#	Select — выбор ЦУ инициатором *{Select)* или инициатора ЦУ *(Reselect)*
C/D#	Control/Data — управление (низкий уровень/данные (высокий уровень)
I/O*	Input/Output — направление передачи относительно ИУ: вводу в ИУ соответствует низкий уровень. Используется для различия прямой *(Select)* и обратной *(Reselect)* выборки: фазе *Selection* соответствует низкий уровень
MSG#	Message — передача сообщения
DB[0:31]#	Data Bus — инверсная шина данных
DP[0:3]#	Data Parity — инверсные биты паритета, дополняют количество единичных битов байтадо нечетного, DPO# относится к DB[0:7],... DP3# — к DB[24:31]. В фазе арбитража не действуют
TERMPWR	Terminator Power — питание терминаторов
ATN#	Attention — внимание (намерение ИУ послать сообщение)
REQ#	Request — запрос от ЦУ на пересылку данных
ACK#	Acknowledge — подтверждение передачи (ответ на REQ#)
RST#	Reset — сброс
DIFFSENS	Признак дифференциального (LVD) интерфейса: ниже 0,7 В—линейный SE; 0,9-1,9 В—дифференциальный LVD; выше 2,4 В—дифференциальный HVD.

Каждое *устройство SCSI,* подключенное к шине, должно иметь свой уникальный *адрес,* назначаемый при конфигурировании. Для 8-битной шины диапазон значений адреса 0 - 7, для 16-битной — 0 - 15. Адрес задается предварительной установкой переключателей или джамперов. Для хост-адаптера возможно программное конфигурирование. Адресация устройств на шине в фазах выборки осуществляется через *идентификатор SCSI ID,* представляющий адрес в позиционном коде. Адрес определяет номер той линии шины данных, которая осуществляет выборку данного устройства. Устройство с нулевым адресом выбирается низким уровнем на линии DB0# (SCSI ID=00000001), с адресом 7 - на линии DB7# (SCSI ID=10000000). Для ИУ значение идентификатора определяет приоритет устройства при использовании шины, наибольший приоритет имеет устройство с большим значением адреса. Понятия адрес и идентификатор часто путают, но это всего лишь две различные формы представления одного и того же параметра.

В любой момент обмен информацией по шине может происходить только между парой устройств. Операцию начинает *инициатор обмена* ИУ (initiator), а *целевое устройство* ЦУ (target) ее исполняет. ИУ выбирает ЦУ по его идентификатору. Чаще всего роли устройств фиксированы: хост-адаптер является инициатором (ИУ), а периферийное устройство — целевым (ЦУ). Возможны комбинированные устройства, выступающие в роли и ИУ, и ЦУ. В ряде случаев роли устройств меняются: ЦУ может, пройдя фазу арбитража, выполнить обратную выборку *(Reselect)* ИУ для продолжения прерванной операции. При выполнении команды копирования *(Copy)* ИУ дает указание ведущему устройству копирования *(Copy Master)* на обмен данными, который может производиться и с другим ЦУ (для которого ведущее устройство копирования выступит в роли ИУ).

Информация по шине данных передается побайтно (пословно) асинхронно, используя механизм запросов (REQuest) и подтверждений (ACKnowledge). Каждый байт контролируется на нечетность (кроме фазы арбитража), но контроль может быть отключен. Интерфейс имеет возможность синхронной передачи данных, ускоряющей обмен.

5.3.1.1.1. Фазы шины

Шина может находиться в одной из десяти ниже перечисленных фаз. Роли источников сигналов между ИУ и ЦУ в этих фазах описаны в табл. 5.27.

Таблица 5.27. Источники сигналов SCSI

Фаза шины	Сигнал				
	BSY#	SEL#	REQ#, C/D#, I/O#, MSG#	ACK#, ATN#	DBx#, DBPx#
Bus Free	-	-	-	-	-
Arbitration	AA	WA	-	-	SID
Selection	I,T	I	-	I	I
Reselection	I,T	T	T	I	T
Command	T	-	T	I	I
Data IN	T	-	T	I	T
Data OUT	T	-	T	I	I
Status	T	-	T	I	T
Message IN	T	-	T	I	T
Message OUT	T	-	T	I	I

I: источник сигнала — ИУ;
T: источник сигнала — ЦУ;
AA: источник сигнала — устройство, активное в арбитраже;
WA: источник сигнала — устройство-победитель в арбитраже;
SID: каждое устройство управляет только битом данных, соответствующим значению его SCSI ID.

В *фазе Bus Free* шина находится в состоянии покоя —нет никаких процессов обмена; она готова к арбитражу. Признаком является пассивное состояние линий BSY# и SEL#.

В *фазе Arbitration* устройство может получить право на управление шиной. Дождавшись покоя шины (Bus Free), устройство вводит сигнал BSY# и свой идентификатор SCSI ID. Если идентификаторы выставили несколько устройств одновременно, то право на управление шиной получает устройство с наибольшим адресом, а остальные устройства отключаются до следующего освобождения шины. Устройство, выигравшее арбитраж, вводит сигнал SEL# и переходит в фазу *Selection* или *Reselection.*

В *фазе Selection* ИУ, выигравшее арбитраж, вводит на шину данных результат логической функции ИЛИ от пары идентификаторов — своего и ЦУ, —сопровождая его битом паритета. Установкой сигнала ATN# ИУ указывает, что следующей фазой будет Message OUT. ИУ снимает сигнал BSY#. Отсутствие сигнала I/O# отличает данную фазу от *Reselection.* Адресованное ЦУ отвечает сигналом BSY#, если паритет корректный и на шине данных присутствует только пара идентификаторов (его и ИУ). На некорректные значения данных устройства отвечать не должны. Если за заданное время ЦУ не ответило, срабатывает тайм-аут, ИУ освобождает шину или вводит сигнал сброса RST#.

Фаза Reselection аналогична предыдущей, но ее вводит ЦУ. Фаза появляется в том случае, когда ЦУ на время исполнения команды отключалось от шины. По завершении внутренней операции это устройство, выиграв арбитраж, будет вызывать ИУ, которое ранее породило исполнение операции. ЦУ снимает сигнал BSY#, активность сигнала I/O# отличает данную фазу от фазы *Selection.* Адресованное ИУ отвечает сигналом BSY#, условия ответа и тайм-аут аналогичны предыдущей фазе.

В *фазах Command, Data, Status* и *Message* по шине данных передается информация, фазы идентифицируются сигналами MSG#, C/D# и I/O# (табл. 5.28), которыми управляет ЦУ. ИУ может потребовать посылки сообщения *(фаза Message OUT)* введением сигнала ATN#, а ЦУ может освободить шину, сняв сигналы MSG#, C/D#, I/O# и BSY#.

Таблица 5.28. Информационные фазы SCSI

Сигнал			Фаза	Направление
MSG#	C/D#	I/O#		
0	0	0	*Data OUT*	I⇒T
0	0	1	*Data IN*	I⇐T
0	1	0	*Command*	I⇒T
0	1	1	*Status*	I⇐T
1	0	0	Зарезервировано	
1	0	1	Зарезервировано	
1	1	0	*Message OUT*	I⇒T
1	1	1	*Message IN*	I⇐T

Временные диаграммы асинхронного обмена приведены на рис. 5.10. Здесь передача каждого байта сопровождается взаимосвязанной парой сигналов REQ#/ACK#. ИУ фиксирует принимаемые данные, получив сигнал REQ# (по отрицательному перепаду). ЦУ считает принимаемые данные действительными по отрицательному перепаду сигнала АСК#. Асинхронный обмен поддерживается

всеми устройствами для всех фаз передачи информации.

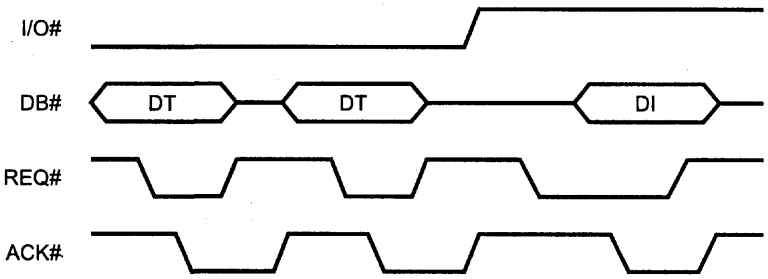

Рис. 5.10. Временные диаграммы асинхронного обмена (DI — данные от ИУ DT — данные от ЦУ)

Фазы передачи данных *Data OUT и Data IN* по предварительной «договоренности» устройств могут выполняться и в синхронном режиме обмена, диаграммы которого приведены на рис. 5.11. При согласовании синхронного режима определяются минимальные длительности и периоды управляющих импульсов ACK# и REQ#, а также допустимое отставание подтверждений от запросов (REQ/ACK offset agreement). ЦУ передает серию данных, сопровождаемых стробами REQ# (рис. 5.11, а), в темпе, ограниченном установленными временными параметрами. ИУ фиксирует принимаемые данные по отрицательному перепаду сигнала REQ#, но отвечать на них сигналом ACK# может с некоторым опозданием. Как только отставание числа принятых сигналов ACK# от числа посланных REQ# достигнет оговоренного предельного значения (в данном примере 2), ЦУ приостановит обмен до прихода очередного подтверждения ACK#. Операция будет считаться завершенной, когда число принятых подтверждений совпадет с числом посланных запросов. При приеме данных ЦУ механизм согласования остается тем же, но данные фиксируются по отрицательному перепаду сигнала ACK# (рис. 5.11, *б).*

ВНИМАНИЕ

В спецификации SCSI-1 момент возобновления передачи после устранения отставания описан нечетко, в результате чего разработчики могли считать, что очередной запрос (и данные) может последовать лишь после окончания (положительного перепада) сигнала ACK#. Устройство, на это рассчитанное, может терять данные: для него последний сигнал REQ# (и данные) является неожиданным и выглядит как превышение согласованного смещения.

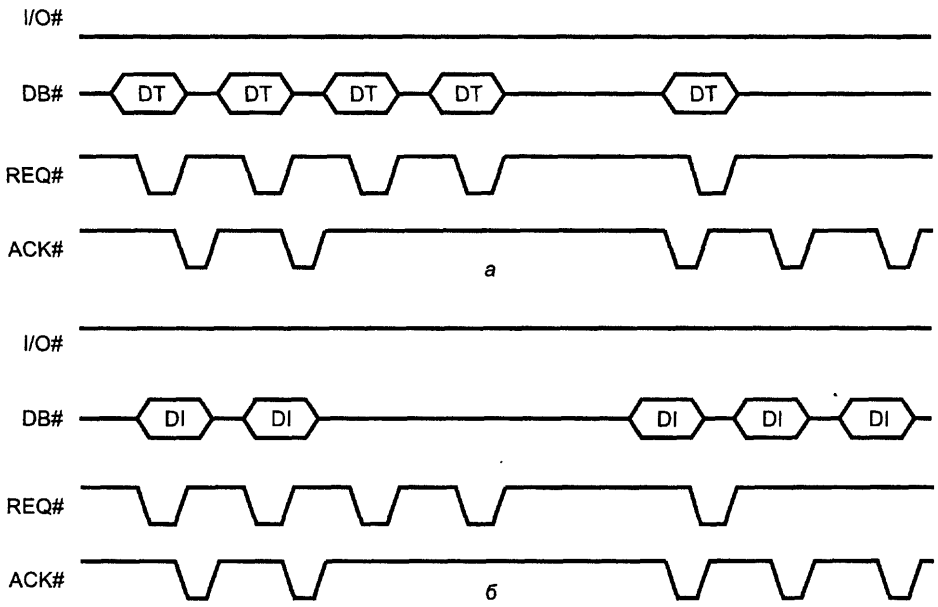

Рис. 5.11. Временные диаграммы синхронного обмена: а — передача; б — прием

Обмен при разрядности 16 и 32 бита происходит аналогично, но при использовании двух кабелей (B и Q) передача по ним управляется сигналами REQB#/ACKB# и REQQ#/ACKQ# соответственно. По обоим кабелям передачи выполняются в одинаковых режимах. Если в последней фазе данных используются не все байты, передатчик обязан снабдить их корректным битом паритета.

При описании фаз шины не говорилось о временных задержках. Они определяются спецификацией так, чтобы возможный «перекос» — неодновременный приход сигналов, вызванный задержкой как в электронных схемах, так и в разных проводах кабеля — не влиял на устойчивость протокола. В асинхронном режиме обмена на скорость передачи информации влияет и длина кабеля, поскольку

изменения состояний участников обмена привязываются к сигналам, распространяющимся по кабелю с ограниченной скоростью. Из-за необходимости учета задержек в случае применения пары кабелей в каждом из них используется своя пара REQ#/ACK#.

В фазе *Command* ЦУ запрашивает от ИУ команду. В фазе *Status* ЦУ делает запрос на передачу ИУ информации о своем состоянии. В фазах *Data IN* и *Data OUT* ЦУ делает запросы на передачу данных к ИУ и от него соответственно. Фазы *Message IN* и *Message OUT* служат*:* для передачи сообщений. Фазу *Message OUT* ЦУ вводит в ответ на условие *Attention,* порождаемое ИУ сигналом ATN#, когда оно нуждается в посылке сообщения ЦУ. Фазу *Message IN* ЦУ вводит при необходимости посылки сообщения ИУ.

Между фазами передачи информации сигналы BSY#, SEL#, REQ# и ACK# должны оставаться в неизменном состоянии, меняться могут только значения сигналов C/D#, I/O#, MSG# и шины данных.

Сигналы ATN# и RST# могут порождать условия *Attention* и *Reset* соответственно, причем асинхронно по отношению к фазам шины. Эти условия могут привести к изменению предопределенного порядка фаз. Сигнал ATN# вводится ИУ во время любой фазы, кроме арбитража и состояния покоя шины. Сигнал RST# вводится в любой момент любым устройством, и по условию Reset все устройства должны немедленно освободить шину. В зависимости от настройки, принятой для всех устройств конкретной системы, возможно выполнение одного из двух вариантов сброса. «Жесткий» сброс переводит устройства в состояние, принятое по включению питания, сбрасывая все текущие процессы, очереди и т. п. В случае «мягкого» сброса после освобождения шины устройства пытаются завершить начатые операции, сохраняя текущие значения настроек.

Каждый процесс ввода/вывода состоит из следующей последовательности фаз шины: из состояния *Bus Free* через фазу *Arbitration* переход к фазе *Selection* или *Reselection.* Далее следуют фазы передачи информации *(Command, Data, Status, Message).* Завершающей фазой является *Message IN,* в которой передается сообщение *Disconnect* или *Command Complete,* после чего шина переходит в состояние покоя *Bus Free.*

Архитектура SCSI обеспечивает для каждого процесса ввода/вывода сохранение набора из трех указателей (saved SCSI pointers), для команды, состояния и данных. ИУ имеет текущий набор указателей (только один), в который копируется сохраненный набор для текущего процесса. Текущие указатели указывают на очередной байт команды, состояния и данных, которые будут передаваться между памятью ИУ и ЦУ. Сохраненные указатели команды и состояния всегда указывают на начала блоков дескрипторов команд и состояния. Сохраненный указатель данных указывает на начало блока данных до тех пор, пока ЦУ не пришлет сообщение *Save Data Pointer.* По его приему будет сохранен текущий указатель данных. Когда ЦУ отключается от шины, информация о текущем процессе ввода/вывода содержится в сохраненном наборе указателей. При возобновлении процесса ЦУ сообщением *Restore Painters* может потребовать у ИУ скопировать сохраненный набор в текущий и продолжить выполнение команд данного процесса ввода/вывода.

ВНИМАНИЕ ———————————————————————————————————————

Поскольку указатель данных может быть модифицирован ЦУ до завершения ввода/вывода, использование указателя для определения реального количества переданных данных дает ненадежные результаты

5.3.1.1.2. Управление интерфейсом

Для управления интерфейсом служит система сообщений — *Message System,* которыми обмениваются ИУ и ЦУ. Обмен происходит в фазах *Message IN/OUT* (см. выше), в одной фазе может передаваться несколько сообщений. Одно сообщение не может расщепляться на несколько фаз. Форматы сообщений стандартизованы; каждое сообщение начинается с кода. Существуют однобайтные (коды 00h, 02h-lFh, 80h-FFh), двухбайтные (коды 20h-2Fh) и расширенные сообщения (код 0lh). В двухбайтном сообщении второй байт является аргументом сообщения. В расширенных сообщениях второй байт задает длину, а последующие байты несут код и аргументы сообщения. Коды сообщений приведены в табл. 5.29.

Таблица 5.29. Сообщения шины SCSI

Код	Направление	Сообщение	Назначение
00h	In	*Command Complete*	Процесс ввода/вывода завершен, информация о состоянии послана в сообщении Status
02h	In	*Save Data Pointer*	Сохранение указателя данных
03h	In	*Restore Pointers*	Восстановление указателей
04h	In	*Disconnect*	Текущие соединение разрывается, но для продолжения процесса потребуется повторное соединение (не вызывает сохранения текущих указателей)
04h	Out	*Disconnect*	Инструкция ЦУ на разрыв соединения

Таблица 5.29. Сообщения шины SCSI (продолжение)

Код	Направление	Сообщение	Назначение
05h	Out	*initiator Detected*	ИУ обнаружило ошибку на шине
06h	Out	*Abort*	Сброс всех процессов, связанных с данной парой I_T, и освобождение шины
07h	In/Out	*Message Reject*	Сообщение (или его параметры) недействительны для получателя
08h	Out	*No Operation*	ИУ нечего ответить на запрос сообщения
09h	Out	*Message Parity Error*	Последний байт сообщения принят с неверным паритетом
0Ah	In	*Linked Command Complete*	Цепочка команд завершена. ИУ может инициализировать указатели для новой цепочки
0Bh	In	*Linked Command Complete (With*	То же, но ИУ между цепочками команд вызывает прерывание в хост-системе
0Ch	Out	*Bus Device Reset*	Аппаратный сброс ЦУ и освобождение шины
0Dh	Out	*Abort Tag*	Сброс текущего процесса, связанного с данной парой I_T, и освобождение шины
0Eh	Out	*Clear Queue*	Сброс всех процессов и освобождение шины (эквивалентно серии сообщений Abort, принятых от всех ИУ)
0Fh	In/Out	*Initiate Recovery*	Сообщение для уведомления об асинхронных событиях (ЦУ временно становится ИУ)
10h	Out	*Release Recovery*	Завершение обработки асинхронных событий
11h	Out	*Terminate I/O Process*	Принудительное завершение текущего процесса без логического повреждения носителя у ЦУ
12h-1F 30h-7F		Зарезервированы для 1-байтных сообщений	
80h-FFh	Out	*Identify*	Установление связи типа I_T_L или I_T_R
80h-FFh	In	*Identify*	Восстановление связи типа I_T_L или I_T_R при повторном соединении. Вызывает восстановление сохраненных указателей
2 - байтовые сообщения			
20h	In	*Simple Queue Tag*	Процесс помещен в очередь, аргумент — Тег (00h-FFh)
20h	Out	*Simple Queue Tag*	Обращение к конкретному процессу маркированной очереди, аргумент — Тег (00h-FFh)
21h	Out	*Head Of Queue Tag*	Помещение процесса в начало маркированной очереди для данного LUN, аргумент— Тег (00h-FFh)
22h	Out	*Ordered Queue Tag*	Помещение процесса в конец маркированной очереди для данного LUN, аргумент — Тег (00h-FFh)
23h	In	*Ignore Wide Residue*	Последние байты последнего переданного слова недействительны (аргумент см. в табл. 5.30).
24h - 2Fh			Зарезервированы для 2-байтных сообщений
Расширенные сообщения. Первый байт - 01h, в поле "код" первым байтом указана длина, вторым - расширенный код.			
05h, 00h	In		*Modify Data Pointer* Запрос модификации указателя данных — суммирование текущего указателя с 4-байтным аргументом сообщения
03h, 01h	In/Out	*Synchronous Data Transfer Request*	Параметры синхронного режима обмена. 1 -и байт аргумента определяет период посылки, 2-й —допустимое смещение REQ/ACK
02h, 03h	In/Out		*Wide Data Transfer Request* Разрядность передач в фазах данных. Аргумент задает число байтов: 0 — 1 байт, 1 — 2 байта, 2—4 байта

В SCSI-2 для установления связи процесса с конкретным логическим устройством *I_T_L* (Initiator_Target_LUN) или с конкретной целевой программой *I_T_R* (Initiator_Target_TRN), а также для предоставления права разрыва соединения служат сообщения *Identify*. В байте их кодов биты [2:0], в зависимости от бита 5 *(LUNTAR),* задают номер *LUN (LUNTAR=0)* или *TRN (LUNTAR=1).* Каждый процесс может быть адресован только одному *LUN* или *TRN.* Если ЦУ во время выполнения процесса обнаружит сообщение с иным адресом связи, оно обязано освободить шину (ситуация неожиданного разрыва). Единичным значением бита 6 *(DiscPriv)* при передаче сообщения ИУ наделяет ЦУ правом

разрыва соединения.

Таблица 5.30. Игнорирование лишних данных

Значение 2-го байта	Игнорировать биты данных при разрядности обмена	
	32 бит	16 бит
01h	DB[31:24]	DB[15:8]
02h	DB[31:16]	Зарезервировано
03h	DB[31:8]	Зарезервировано
00h, 04h - FFh	Зарезервировано	Зарезервировано

ИУ может проинструктировать ЦУ на разрыв соединения, послав ему сообщение *Disconnect.* Получив сообщение, ЦУ посылает одноименное сообщение (предварительно ЦУ может потребовать сохранения указателя данных, послав сообщение *Save Data Pointer)* и освобождает шину. Если ЦУ не поддерживает эту возможность, оно отвечает сообщением *Message Reject.* Когда процесс, продолжающийся в устройстве, потребует передачи данных, через фазу арбитража ЦУ обратится к ИУ за продолжением обмена.

С помощью сообщений согласуются параметры синхронного режима и разрядность данных. Процесс согласования синхронного обмена называется *Synchronous Negotiation.* Устройство, запрашивающее синхронный обмен, посылает сообщение *Synchronous Data Transfer Request c* указанием допустимого периода цикла и отставания REQ/ACK. Если другой участник обмена поддерживает синхронный режим, он предложит свои параметры. Согласованными параметрами будут максимальный период и минимальное отставание (нулевое отставание эквивалентно асинхронному режиму). Выбранный режим будет относиться только к фазам передачи между данной парой устройств. Отвергнутое сообщение является требованием асинхронного режима. Поскольку старые хост-адаптеры не поддерживали согласование синхронного режима, на ЦУ запрос синхронного режима может быть заблокирован. О возможности работы в синхронном режиме хост может узнать, послав команды *Request Sense* и *Inquiry.*

Разрядность передач согласуется аналогично посредством сообщений *Wide Data Transfer Request.* Согласованные режимы будут действовать до сброса устройств по сообщению *Bus Device Reset* или «жесткого» сброса, что приведет к установке предопределенных режимов по включению. Согласование режимов не должно инициироваться в каждом процессе, поскольку затраты времени на эту процедуру сводят на нет выигрыш в производительности.

5.3.1.2. Типы периферийных устройств

Каждое *логическое устройство* (ЛУ), являющееся абонентом шины SCSI, может быть представителем одного или нескольких *однотипных периферийных устройств* (ПУ), перечень стандартизованных типов ПУ приведен в табл. 5.31.

Таблица 5.31. Типы ПУ SCSI

Код типа	Назначение
00h	Direct-access device — устройства прямого доступа (накопители на магнитных дисках)
01h	Sequential-access device — устройства последовательного доступа (накопители на магнитных лентах)
02h	Printer device — принтеры
03h	Processor device — процессоры (устройства обработки данных)
04h	Write-once device — устройства однократной записи (некоторые оптические диски)
05h	CD-ROM device — накопители CD-ROM
06h	Scanner device — сканеры
07h	Optical memory device — устройства оптической памяти
08h	Medium Changer device — устройства смены носителей (jukebox)
09h	Communications device — коммуникационные устройства
0Ah-0Bh	Устройства класса ASC ITS (Graphic Arts Pre-Press Devices — высококачественные устройства печати)
0Ch	Array controller device — контроллеры массивов накопителей
0Dh	Enclosure services device — сервисные устройства шкафов
0Eh	Reduced block command devices — блочные устройства с сокращенным набором команд
0Fh	Optical card reader/writer device — устройства считывания и записи оптических карт (штрих-коды)
10h-1Eh	Зарезервировано
1Fh	Неизвестный тип или устройство отсутствует

Сложное ПУ может представляться несколькими ЛУ SCSI. По характеру обмена данных устройства разделяются на 2 класса — блочные (block device) с типами 0, 4, 5, 7 и поточные (stream device) с типами 1, 2, 3, 9.

Устройства прямого доступа (0) позволяют сохранять блоки данных. Каждый блок хранится по уникальному логическому адресу LBA — Logical Block Address. Взаимное расположение логических блоков на носителе не регламентируется. Адрес первого логического блока — нулевой, последнего — (n - 1), где n — общее число блоков. В цепочках команд устройствами может поддерживаться относительная адресация, когда исполнительный адрес в команде определяется смещением относительно адреса, действовавшего в предыдущей команде.

Блоки данных хранятся на носителе вместе с дополнительной информацией, используемой контроллером для управления чтением и записью, а также для обеспечения надежности хранения данных (ECC или CRC-коды). Формат дополнительных данных не регламентируется, ЦУ скрывает эти данные от ИУ.

Для каждого блока может быть установлена своя длина, но чаще используют единую длину блока для всего носителя. Группа смежных блоков одинаковой длины называется *экстентом* (extent), экстенты определяются командой MODE SELECT, длину блока можно узнать по команде *MODE SENSE*. После изменения длины блока для активизации экстента обычно требуется форматирование.

Носитель может быть разделен на области, одна из которых используется для хранения блоков данных, другая резервируется для замены дефектных блоков, часть носителя может использоваться контроллером для обслуживания устройства. Дефектные блоки области данных могут быть переназначены на другую область носителя, что позволяет их скрыть.

Носитель может быть *фиксированным* и *сменяемым* (Removable). Сменяемый носитель в картридже (или чехле), если таковой предусматривается конструкцией, называют *томом* (volume). Для чтения/записи том должен быть смонтирован.

Устройство может быть зарезервировано ИУ, при этом доступ к нему других ИУ ограничивается. Ограничения распространяются на ЛУ или экстент.

Устройства, имеющие кэш данных, могут поддерживать политику обратной записи (WB, Write Back). При этом появляются интервалы времени, в течение которых внезапное отключение питания устройства приведет к потере данных, поскольку сообщение о завершении команды посылается после записи в кэш, а не на носитель. Сообщения об ошибках при WB поступают к ИУ с опозданием. Чтобы избежать этих неудобств, ИУ может запретить устройству использовать WB. Отдельные блоки в кэше можно фиксировать, не допуская их замещения при последующих операциях обмена.

Типичный пример устройств прямого доступа — накопитель на магнитных дисках (винчестер).

ВНИМАНИЕ ———————————————————————————————————

Винчестер с интерфейсом SCSI не имеет собственной трехмерной логической геометрии. В соответствии с системой команд SCSI доступ к его секторам выполняется только по линейному логическому адресу LBA. Трансляцию LBA в номер физического цилиндра, головки и сектора выполняет встроенный контроллер, скрывая от пользователя и зонный формат записи. Трехмерную логическую геометрию, по которой к диску можно обратиться через BIOS *INT 13h,* «придумывает» и эмулирует SCSI BIOS, «зашитая» в ПЗУ контроллера SCSI или загружаемая (см. п. 5.3.1.3).

Есть устройства прямого доступа на ленточном носителе — Floppy Tape. Именно для них эффективна команда SEEK. Устройства прямого доступа могут не иметь подвижных носителей, а основываться на памяти разной природы: SRAM, DRAM, FRAM, EEPROM, флэш-память.

Устройства последовательного доступа (1) имеют ряд особенностей, связанных с их принципом действия. Носитель представляет собой магнитную ленту с многодорожечным, серпантинным или наклонно-строчным типом записи. Носитель — всегда сменяемый, с некоторым конструктивным обрамлением (катушка, картридж), также называется томом. Том имеет начало носителя *BOM* (beginning-of-medium) и конец носителя *EOM* (end-of-medium). При записи ИУ должен заранее узнавать о приближении конца носителя, для чего определяется позиция раннего предупреждения *EW* (Early Warning) с соответствующим маркером. Это позволяет после записи блока данных из буфера поместить на носитель соответствующий концевой маркер. Том может иметь один или несколько *разделов* (partitions), нумеруемых с нуля. Разделы располагаются друг за другом, без перекрытия. Каждый раздел «x» имеет начало *BOPx* (Begining-Of- Partition x), конец *EOPx* и раннее предупреждение *EWx*. Между началом и концом раздела помещаются блоки данных и маркеры. Блоки данных, передаваемые ИУ, называются логическими. Логический блок может занимать один или несколько физических блоков данных на носителе, в последнем случае за блокирование-деблокирование отвечает ЦУ. Описатели записанных физических и/или логических блоков могут храниться на носителе (определяется форматом записи). Буфер устройства должен вмещать по крайней мере один логический блок.

Принтеры (2), подключаемые через интерфейс SCSI, не требуют особых команд для управления, поскольку функции реализуются через поток передаваемых данных. Однако двунаправленная связь по

шине позволяет ввести дополнительные команды, служащие для отслеживания состояния принтеров с буферной памятью, и обеспечить целостность заданий. Принтеры могут иметь встроенный контроллер SCSI или подключаться как ЛУ к внешнему контроллеру по интерфейсу Centronics или RS-232. Параметры подключения определяются командой *MODE SELECT*.

Процессорными устройствами (3) в терминологии SCSI являются источники и потребители пакетов информации, трактовка которой стандартом не определена. Примерами процессорных устройств являются компьютеры, обменивающиеся сообщениями односторонним или двусторонним образом. Процессорным устройством является и какое-либо сложное устройство отображения (графический дисплей), которое занято выводом потока сообщений. От коммуникационных устройств процессорные отличаются тем, что они являются источниками или потребителями информации, в то время как коммуникационные служат лишь посредниками.

Устройства оптической памяти (7) близки к устройствам прямого доступа со сменными носителями, но имеют ряд характерных особенностей. Большая емкость носителя вызывает необходимость применения команд с 12-байтным дескриптором. Устройства могут обеспечивать считывание, однократную или многократную запись. На носителе могут быть определены зоны, недоступные для записи. Блоки носителя имеют состояние «чистый» и «записанный», что отмечается соответствующим атрибутом. В устройствах многократной записи перед повторной записью необходимо стереть перезаписываемый блок. Стирание может выполняться специальной командой или автоматически по команде записи. В записи фаза стирания может отсутствовать. К этим устройством применимо понятие *обновления* (update) логического блока — запись новых данных по тому же адресу логического блока, но на другое место носителя. Прежние данные могут быть считаны специальной командой, указывающей кроме логического адреса блока и его поколение (generation). Емкость носителя в таком случае сообщается без учета множества поколений. Оптические носители по сравнению с магнитными имеют существенно более высокий уровень ошибок, так что приходится использовать более сложные алгоритмы восстановления информации.

Устройства однократной записи (4), обычно оптические, отличаются невозможностью перезаписи ранее записанного блока. Попытка повторной записи, в зависимости от реализации устройства, может привести к потере записанных данных. Каждый блок имеет состояние «записан» или «не записан», инициализация (форматирование) не применяется.

Накопители CD-ROM (5) предназначаются для работы с CD. Изначально диски содержали звукозапись и были рассчитаны не только на чтение блоков данных, но и на потоковый вывод на внешний аудиоинтерфейс. Запись не предусмотрена. Данные на диске адресуются по-разному. Физический сектор имеет 2352 байт, из которых обычно используется 2048, поле синхронизации 12 байт и поле тега адреса сектора 4 байт. Дополнительное поле 288 байт используется для исправления ошибок данных, но если ошибки допустимы, его тоже используют для хранения данных. Таким образом, физический сектор данных может иметь размер 2048 байт (CD-ROM Data Mode 1) или 2336 и даже 2340 (вместе с полем тега) байт (CD-ROM Data Mode 2). В зависимости от размера логического блока (1024, 512 или 256 байт) сектор CD-ROM может вмещать 2, 4 или 8 блоков.

Сканеры (6) передают ИУ данные, описывающие растровое изображение сканируемого объекта. Команды позволяют задавать окна сканирования, определяя в них режим и разрешение. Для некоторых функций требуется посылка данных в сканер (например, полутоновые маски). Для сканеров с автоподачей имеются команды позиционирования.

Устройства смены носителей (8) предназначены для автоматического манипулирования сменными носителями — дисками и картриджами с магнитными лентами. В SCSI они представляются ЛУ, отличающимися от первичных устройств хранения, которые они обслуживают. Модель устройства состоит из набора адресуемых элементов, каждый из которых может «держать» только одну единицу носителя:

- Medium transport element — элемент транспортировки носителя;
- Storage element — место хранения единицы носителя, когда он не находится ни в одном из элементов трех других типов;
- Import export element — элемент, с помощью которого устройство принимает носители извне или отдает их;
- Data transfer element — позиция первичного устройства, осуществляющего обмен данными с носителем.

Элементы адресуются 16-битными адресами; доступна информация об их состоянии. Каждый том носителя снабжается идентификатором-тегом, по которому определяется его нахождение в элементах.

Коммуникационные устройства (9) предназначены для обмена информацией с устройствами через внешнюю, по отношению к шине SCSI, среду передачи данных. Внешние протоколы стандартом SCSI не описываются — вся необходимая для них информация заключена в сообщениях, передаваемых и принимаемых ИУ по командам *SEND MESSAGE* и *GET MESSAGE*.

5.3.1.3. Адресация и система команд

Как указывалось выше, любое устройство SCSI на шине адресуется идентификатором *SCSI ID,* соответствующим заданному уникальному адресу. В ЦУ может быть определено до 8 ЛУ со своими номерами *LUN(* Logical Unit Number) в диапазоне 0-7. Понятие LUN неприменимо к ИУ (хост-адаптеру), но SCSI-устройство двойного назначения может иметь ЛУ.

Система команд и сообщений позволяет адресовать как ЦУ в целом, так и любое его ЛУ. В ЦУ может быть определено до 8 *целевых программ* TRN (Target Routine), которые не имеют непосредственной привязки к ЛУ. Целевые программы появились в SCSI-2, их адресация также производится через сообщения.

Система команд SCSI включает общие команды, применимые для устройств всех классов и специфические для каждого класса. Как общие, так и специфические наборы команд содержат обязательные (mandatory), дополнительные (optional) и фирменные (vendor specific) команды. Любое SCSI-устройство должно поддерживать обязательные команды общего набора и своего класса, чем обеспечивается высокий уровень совместимости. Команда передается ИУ в ЦУ через *блок дескриптора команды* (command descriptor block), посылаемый в фазе *Command.* Некоторые команды сопровождаются блоком параметров, следующим за блоком дескриптора в фазе *Data.* Форматы блоков стандартизованы, длина блока определяется кодом операции (operation code), который всегда является первым байтом блока и может составлять 6, 10 или 12 байт. Типовой блок содержит следующие поля (рис. 5.12):

- *OpCode* — код операции, 1 байт (граф 4 «Код» в табл. 5.33). Биты [7:5] определяют группу, а биты [4:0] код команды. В группе 0 блок дескриптора имеет длину 6 байт, в группах 1 и 2 — 10 байт, в группе 5—12 байт. Группы 3 и 4 зарезервированы, группы 6 и 7 отданы на усмотрение разработчиков.
- *LUN* — номер ЛУ (для совместимости со SCSI-1), всегда занимает биты [7:5] байта 1. Если для идентификации используются сообщения *Identify,* то это поле игнорируется (рекомендуется устанавливать *LUN=0).*
- *LBA* — адрес логического блока, 21 бит для 6-байтных блоков и 32 бит — для 10-и 12-байтных. В ряде команд поле не используется.
- *Length* — длина (количество блоков или байтов) передаваемых данных (transfer length), блока параметров (parameter list length) или блока, резервируемого ИУ под данные (allocation length). Поле имеет один из трех вариантов назначения, в зависимости от команды. При однобайтном задании длины 0 соответствует значению 256, в длинных формах 0 указывает на отсутствие передач. В ряде команд поле не используется.
- *Control* — байт управления. Биты [7:6] отданы на усмотрение разработчика, биты [5:2] — зарезервированы, бит 1 — *Flag,* бит 0 — *Link* (служит признаком объединения команд в цепочку). *Flag* определяет сообщение, передаваемое в случае успешного выполнения команд цепочки, при *Flag* = 1 сообщение *Linked Command Complete (With Flag)* будет вызывать прерывания между командами цепочки.

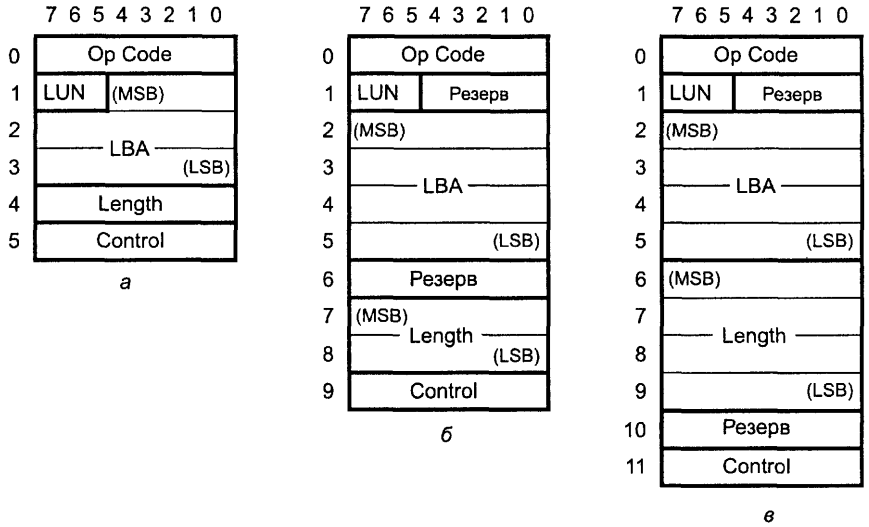

Рис. 5.12. Форматы блоков дескрипторов команд: а — 6 байт; б — 10 байт; в — 12 байт

Обратим внимание на порядок байтов: первыми передаются старшие байты (бит MSB — самый старший), за ними — младшие (бит LSB — самый младший). Зарезервированные поля для совместимости с будущими стандартами должны иметь нулевые значения.

Исполнение команды завершается в фазе *Status* передачей байта состояния *Status Byte.* Байт не

передается, если команда завершена по сообщению *Abort, Abort Tag, Bus Device Reset, Clear Queue,* по условию *Hard Reset* или в случае неожиданного разъединения. В байте состояния используются только биты [5:1], возможные состояния приведены в табл. 5.32 (остальные зарезервированы).

Таблица 5.32. Байты состояния

Биты	Состояние	Значение
76543210		
RR00000R	Good	Успешное завершение команды
RR00001R	*Check Condition*	Указание на асинхронное событие
RR00010R	*Condition Met*	Запрошенная операция выполнена (команды *Search Data* и *Pre-Fetch}*
RR00100R	*Busy*	Занято (невозможен прием команды)
RR01000R	*Intermediate*	Успешное выполнение команды в цепочке
RR01010R	*Intermediate Condition Met*	Удовлетворение запрошенной операции в цепочке команд
RR01100R	*Reservation Conflict*	Попытка обратиться к ЛУ, зарезервированному другим ИУ
RR10001R	*Command Terminated*	Завершение текущего процесса по сообщению *Terminate I/O Process* или по асинхронному событию
RR10100R	*Queue Full*	Очередь (маркированная) заполнена, процесс в очередь не поставлен

Набор команд для устройств классов 0-9 приведен в табл. 5.33. Любое ЦУ SCSI-2 обязано поддерживать четыре команды: *Inquiry, Request Sense, Send Diagnostic, Test Unit Ready:* Они используются для конфигурирования системы, тестирования устройств, а также сообщений об ошибках и исключительных ситуациях. Команда *Inquiry* позволяет получить информацию о ЛУ через стандартизованный блок данных длиной 96 байт: тип подключенного ПУ, возможность смены носителя, поддержка 32- или 16-битного расширения, синхронного обмена, относительной адресации, цепочек команд, очередей и уведомления об асинхронных событиях. Здесь же описывается уровень поддержки стандартов SCSI (ISO, ANSI, ECMA), идентификаторы производителя, устройства и т. п. Формат блока по усмотрению производителя может быть расширен.

В графе «Команда» в скобках указана длина блока дескрипторов, для случаев, когда существуют разные версии одноименных команд.

В графе «Применимость» указаны номера типов ПУ, соответствующие табл. 5.31 (символ * соответствует всем типам). Команды для каждого типа устройств могут быть обязательными — номер сопровождается символом m (Mandatory) — или необязательными — номер типа сопровождается символом "o" (Optional). Часть команд в устройствах разных классов реализуется по-разному, что отмечено обозначением *z.

Таблица 5.33. Команды SCSI

Команда	Код	Применимость	Назначение
Change Definition	40h	*o	Модификация определений операций для ЛУ
Compare	39h	*o	Побайтное сравнение данных двух ЛУ[1]
Copy	18h	*o	Копирование данных с одного ЛУ на другое[1]
Copy And Verify	3Ah	*o	Копирование данных с одного ЛУ на другое с верификацией[1]
Erase	19h	1m	Стирание (участка или до конца носителя)
Erase (10)	2Ch	7o	То же с 10-байтным блоком дескриптора
Erase (12)	ACh	7o	То же с 12-байтным блоком дескриптора
Exchange Medium	A6h	8o	Обмен носителями между двумя элементами устройства
Format	04h	2o	Выбор шрифтов и форм
Format Unit	04h	0m 7o	Форматирование устройства
Get Data Buffer Status	34h	6o	Опрос состояния буфера данных
Get Message	08h	9o	Прием пакета из коммуникационного устройства
Get Message (10)	28h	9o	То же с 10-.байтным блоком дескриптора
Get Message (12)	A8h	9o	То же с 12-байтным блоком дескриптора
Get Window	25h	6o	Получение информации о предварительно определенном окне

Таблица 5.33. Команды SCSI (продолжение)

Команда	Код	Применимость	Назначение
Initialize Element Status	07h	8o	Инициализация состояния элемента
Inquiry	12h	*m	Опрос типа устройства, уровня стандарта, идентификатора производителя, модели и т. п.
Load Unload	1Bh	1o	Загрузка/разгрузка носителя
Locate	2Bh	1o	Позиционирование на заданный логический блок
Lock-Unlock Cache	36h	0o 4o 5o 7o	Фиксация заданных логических блоков в кэше устройства и ее отмена
Log Select	4Ch	*o	Запись статистической информации, обработка которой поддерживается устройством, в ЦУ или ЛУ
Log Sense	4Dh	*o	Считывание статистической информации с ЦУ или ЛУ
Media Scan	38h	4o 7o	Сканирование — поиск непрерывной области чистых или записанных блоков
Mode Select (6)	15h	*z	Запись параметров носителя, ЦУ или ЛУ (с 6-байтным блоком дескриптора)
Mode Select (10)	55h	*z	То же с 10-байтным блоком дескриптора
Mode Sense (6)	1Ah	*z	Считывание параметров носителя, ЦУ или ЛУ (с 6-байтным блоком дескриптора)
Mode Sense (10)	5Ah	*z	То же с 10-байтным блоком дескриптора
Move Medium	A5h	8m	Передача носителя
Object Position	31h	6o	Позиционирование (загрузка/выгрузка) сканируемого объекта
Pause/Resume	4Bh	5o	Пауза/продолжение воспроизведения аудио
*Play Audio (*10)	45h	5o	Аудиовоспроизведение указанных логических блоков
Play Audio (12)	A5h	5o	То же с 12-байтным блоком дескриптора
Play Audio Ms f	47h	5o	Аудиовоспроизведение с адресацией MSF
Play Audio Track/Index	48h	5o	Аудиовоспроизведение с указанием треков и индексов
Play Track Relative (10)	49h	5o	Аудиовоспроизведение с адресацией относительно трека
Play Track Relative(12)	A9h	5o	То же с 12-байтным блоком дескриптора
Position To Element	2Bh	8o	Позиционирование транспортного элемента к указанному элементу
Pre-Fetch	34h	0o 4o 5o 7o	Считывание блоков данных в кэш (без передачи ИУ)
Prevent Allow Medium Removal	1Eh	0o 1o 4o 5o 7o 8o	Запрет/разрешение смены носителя в ЛУ
Print	0Ah	2m	Печать блока данных
Read (6)	08h	0m 1m 4o 5o 7o	Чтение данных (с 6-байтным блоком дескриптора)
Read 10)	28h	0m 4m 5m 6m 7m	То же с 10-байтным блоком дескриптора
Read (12)	A8h	4o 5o 7o	То же с 12-байтным блоком. дескриптора
Read Block Limits	05h	1m	Запрос ограничений на длину блока (минимальная и максимальная длины)
Read Buffer	3Ch	*o	Чтение буфера
Read Capacity	25h	0m 4m 7m	Определение емкости ЛУ
Read CD-Rom Capacity	25h	5m	Определение емкости CD-ROM (быстрое определение может давать погрешность)
Read Defect Data	37h	0o7o	Чтение списков дефектных блоков (P-list — исходный список от изготовителя, G-list — список, заполняемый при эксплуатации)
Read Defect Date (12)	B7h	7o	То же с 12-байтным блоком дескриптора
Read Element Status	B8h	8o	Чтение состояния элементов

Таблица 5.33. Команды SCSI (продолжение)

Команда	Код	Применимость	Назначение
Read Generation	29h	7o	Чтение максимально возможного поколения для указанного логического блока
Read Header	44h	5o	Чтение заголовка логического блока CD-ROM
Read Long	3Eh	0o 4o 5o 7o	«Длинное» чтение — данные блока и поля ECC
Read Position	34h	1o	Запрос позиции данных, находящихся в буфере (адрес начала и конца, количество блоков и байтов)
Read Reverse	0Fh	1o	Чтение блоков с текущей позиции в обратном направлении
Read Sub-Channel	42h	5o	Чтение данных субканала CD-ROM
Read TOC	43h	5o	Чтение таблицы содержимого CD-ROM
Read Updated Block	2Dh	7o	Чтение определенного поколения обновленного логического блока
Reassign Blocks	07h	0o 4o 7o	Переназначение дефектных блоков
Receive	08h	3o	Прием пакета
Receive Diagnostic Results	1Ch	*o	Получение результатов диагностики
Recover Buffered Data	14h	1o 2o	Восстановление данных, посланных в буфер, но не записанных (не напечатанных) из-за ошибки
Release	17h	0m 2m 4m 5m 6m 7m 8o	Освобождение зарезервированного ЛУ, экстента или элемента
Request Sense	03h	*m	Опрос уточненного состояния
Request Volume Element Address	B5h	8o	Передача результатов команды SEND VOLUME TAG
Reserve	16h	0m 1 m 2m 4m 5m 6m 7m 8o	Предотвращение использования ЛУ (его экстента или элемента) другим ИУ
Rewind	01h	1m	Перемотка носителя к началу раздела
Rezero Unit	01h	0o 4o 5o 7o 8o	Приведение ЛУ в определенное состояние
Scan	1Bh	6o	Сканирование данных в определенном окне
Search Data Equal	31h	0o 4o 5o 7o	Поиск данных, (не) совпадающих с эталоном[2]
Search Data Equal (12)	B1h	4o 5o 7o	То же с 12-байтным блоком дескриптора[2]
Search Data High	30h	0o 4o 5o 7o	Поиск данных, (не) больших эталона[2]
Search Data High (12)	B0h	4o 5o 7o	То же с 12-байтным блоком дескриптора[2]
Search Data Low	32h	0o 5o 7o	Поиск данных, (не) меньших эталона[2]
Search Data Low (12)	B2h	4o 5o 7o	То же с 12-байтным блоком дескриптора[2]
Seek{6)	0Bh	0o 4o 5o 7o	Позиционирование (с 6-байтным блоком дескриптора)[3]
Seek (10)	2Bh	0o 4o 5o 7o	Позиционирование — поиск логического адреса (с 10-байтным блоком дескриптора)[3]
Send	0Ah	3m	Посылка пакета
Send (10)	2Ah	6o	Посылка данных в устройство
Send Diagnostic	1Dh	*m	Запуск теста ЦУ. Ответом будет состояние Good, если тест прошел успешно, или Check Condition в случае ошибки
Send Message	0Ah	9m	Посылка пакета в коммуникационное устройство
Send Message (10)	2Ah	9o	То же с 10-байтным блоком дескриптора
Send Message (12)	AAh	9o	То же с 12-байтным блоком дескриптора
Send Volume Tag	B6h	8o	Посылка тега тома (шаблона) для поиска его в элементах или для создания нового тега
Set Limits	33h	0o 4o 5o 7o	Определение области логических адресов, над которыми могут выполняться операции цепочки команд
Set Limits (12)	B3h	4o 5o 7o	То же с 12-байтным блоком дескриптора
Set Window	24h	6m	Определение окна сканирования
Slew And Print	0Bh	2o	Прогон бумаги и печать
Space	11h	1m	Относительное позиционирование (вперед и назад) на заданное число блоков, файлов, маркеров и т. п.

Таблица 5.33. Команды SCSI (продолжение)

Команда	Код	Применимость	Назначение
Start Stop Unit	1Bh	0o 4o 5o 7o	Разрешение/запрет операций с носителем, извлечение носителя [4]
Stop Print	1Bh	2o	Останов печати с очисткой буфера или без нее
Synchronize Buffer	10h	2o	Синхронизация буфера — печать всего содержимого, при невозможности — сообщение об ошибке
Synchronize Cache	35h	0o 4o 5o 7o	Синхронизация кэша — запись несохраненных данных заданного диапазона адресов на носитель
Test Unit Ready	00h	*m	Опрос готовности ЛУ
Update Block	3Dh	7o	Обновление логического блока
Verify	2Fh	0o 5o 7o	Верификация — проверка возможности безошибочного считывания блоков данных с носителя
Verify	13h	1o	Тоже
Verify (10)	2Fh	4o 5o 7o	То же или проверка занятости блоков (с 10-байтным блоком дескриптора)
Verify (12)	AFh	4o 5o 7o	То же с 12-байтным блоком дескриптора
Write (6)	0Ah	0o 1m 4o 7o	Запись блоков данных, переданных ИУ (с 6-байтным блоком дескриптора)
Write (10)	2Ah	0o 4m 7m	Запись блоков данных, переданных ИУ (с 10-байтным блоком дескриптора)
Write(12)	AAh	4o 7o	То же с 10-байтным блоком дескриптора
Write And Verify	2Eh	0o 4o 7o	Запись блоков данных, переданных ИУ, с верификацией записи
Write And Verify (12)	AEh	7o4o	То же с 12-байтным блоком дескриптора
Wr/te *Buffer* микрокода	3Bh	*o	Запись в буфер (но не на носитель) или загрузка
Write Filemarks	10h	1m	Запись маркера файлов
Write Long	3Fh	0o 4o 7o	«Длинная» запись — запись блока данных и поля ECC
Write Same	41h	0o	Запись блока данных, переданных ИУ, в группу смежных блоков или до конца носителя[5]

[1] Команды копирования и сравнения данных оперируют парой ЛУ, которые могут принадлежать как одному ЦУ, так и разным, если эту возможность поддерживает ведущее устройство копирования — Copy Master. Копирование возможно между устройствами любых классов. Для устройств типов 8 и 9 эти команды не применяются.

[2] Данные ищутся сравнением указанного числа логических записей с эталоном. Логические записи определяются длиной, начальным логическим блоком и смещением внутри него. Можно потребовать попадания искомых данных в один логический блок.

[3] Команда эффективна для ленточных устройств прямого доступа.

[4] Команда *STOP UNIT для* устройств с кэшированием перед остановом автоматически выполняет синхронизацию кэша.

[5] Первые 4 байта в записываемых блоках могут быть заменены на физический или логический адрес блока.

5.3.1.4. Выполнение команд

Мы не будем рассматривать различные ситуации, приводящие к отклонениям от Нормальной последовательности событий интерфейса. К ним относятся некорректные соединения со стороны ИУ, выбор несуществующего ЛУ, неожиданные выборки ИУ, округление параметров, реакция на асинхронные события и т. п.

Рассмотрим процесс на шине SCSI на примере *одиночной команды* чтения *Read.* ИУ имеет активный набор указателей и несколько сохраненных наборов, по одному на каждый из допустимого числа одновременных конкурирующих процессов. ИУ восстанавливает указатели процесса в активный набор и, выиграв арбитраж, выбирает ЦУ. Как только ЦУ выбрано, оно берет на себя управление процессом. В фазе *Selection* ИУ вводит сигнал ATN#, сообщая о намерении послать сообщение *Identify* с указанием адресуемого ЛУ. Таким образом устанавливается связь *I_T_L с* данным процессом и его набором указателей. ЦУ переходит в фазу *Command* и принимает блок дескриптора команды *Read.* Интерпретировав команду, ЦУ переходит в фазу *Data IN,* передает данные, затем переводится в фазу *Status* и посылает состояние *Good.* Затем в фазе *Message* устройство посылает сообщение *Command Complete,* после чего освобождает шину (фаза *Bus Free).* Процесс завершен.

Рассмотрим тот же пример, но с использованием *отключения от шины (Disconnect)* в процессе

выполнения команды. Если устройство, получив команду *Read,* определит, что для получения затребованных данных необходимо много времени, оно освободит шину, послав сообщение *Disconnect.* Как только требуемые данные готовы в ЦУ, оно, выиграв арбитраж, выберет ИУ (в фазе *Reselect)* и в фазе *Message IN* пошлет ему сообщение *Identify.* По определенной этим сообщением связи *I_T_L* ИУ восстановит соответствующий набор указателей в активный и продолжит выполнение процесса, как описано выше. Если ЦУ хочет отсоединиться, когда часть данных уже передана (например, головка дошла до конца цилиндра и требуется время на позиционирование), оно посылает сообщение *Save Data Pointer,* а затем *Disconnect.* После повторного соединения передача данных возобновится с точки, определенной последним сохраненным значением указателя. Если произошла ошибка или исключение, ЦУ может повторить обмен данными, послав сообщение *Restore Pointers* или отсоединившись без сообщения *Save Data Pointers.*

Теперь рассмотрим процесс с *цепочкой связанных команд.* По успешному завершению каждой команды цепочки ЦУ автоматически переходит к исполнению следующей. Все команды цепочки адресуются к одной и той же связи *I_T_x* и являются частью одного процесса. Команды не являются полностью независимыми — при использовании бита относительной адресации последний блок, адресованный предыдущей командой, доступен для следующей. Так, например, можно исполнить команду *Search Data,* по которой на диске будет найден блок, содержащий информацию, совпадающую с эталоном поиска. Связав с ней команду чтения *Read,* можно прочитать этот блок или блок с указанным смещением относительно найденного. По выполнении связанных команд ЦУ посылает сообщения *Linked Command Complete* (возможно, с флагом), а ИУ обновляет набор сохраненных указателей, так что они указывают на очередную команду цепочки. Команды в цепочке выполняются как одиночные, но с возможностью относительной адресации.

Команды могут исполняться с *использованием очередей.* ЦУ могут поддерживать немаркированные и маркированные очереди. Поддержка *немаркированных очередей,* определенная еще в SCSI-1, позволяет любому ЛУ (LUN) или целевой программе (TRN), занятым процессом от одного ИУ, принимать команды (начинать процесс) с другими ИУ. При этом каждый процесс идентифицируется связью *I_T_x,* где x - LUN или TRN.

Маркированные очереди (tagged queue) определены в SCSI-2 для ЛУ (LUN, но не TRN). Для каждой связи *I_T_L* существует своя очередь размером до 256 процессов (немаркированные очереди можно считать вырожденным случаем маркированных с одноместными очередями). Каждый процесс, использующий маркированные очереди, идентифицируется связью *I_T_L_Q,* где Q— однобайтный *тег очереди* (queue tag). Теги процессам назначаются ИУ, их значения на порядок выполнения операций не влияют. Повторное использование тега возможно лишь по завершении процесса с этим тегом. Постановка в очередь выполняется через механизм сообщений, при этом очередной процесс можно поставить в очередь «по честному», а можно «пропихнуть» вне очереди: процесс, поставленный в очередь с сообщением *Head Of Queue Tag,* будет выполняться сразу после завершения текущего активного процесса. Процессы, поставленные в очередь с сообщением *Simple Queue Tag,* исполняются ЦУ в порядке, который оно сочтет оптимальным.

Процесс, поставленный в очередь с сообщением *Ordered Queue Tag,* будет исполняться последним. ИУ может удалить процесс из очереди, сославшись на него по тегу. Изменение порядка выполнения команд ЦУ не касается порядка команд в цепочке, поскольку цепочка принадлежит одному процессу, а в очередь ставятся именно процессы.

Устройства SCSI допускают *программирование* — Programmable Operating Definition. Программированием определений операций для ЛУ можно изменить такие параметры, как идентификация производителя, типа и модели устройства, уровень соответствия SCSI, номер спецификации, набор команд и т. д. Однако низкоуровневые параметры интерфейса (параметры временных диаграмм, определение паритета) сохраняются неизменными. Текущие определения могут быть считаны ИУ по командам *Inquiry, Mode Sense* и *Read Capacity.*

5.3.1.5. Конфигурирование устройств SCSI

Все устройства на шине должны быть согласованно сконфигурированы. Для них требуется программно или с помощью джамперов установить следующие основные параметры:

- *Идентификатор устройства SCSI ID* — адрес 0-7 (для Wide SCSI 0-15), уникальный для каждого устройства на шине. Обычно хост-адаптеру, который должен иметь высший приоритет, назначается адрес 7(15 для Wide SCSI, если все устройства 16-битные). Позиционный код, используемый для адресации, обеспечивает совместимость адресации 8- и 16-битных устройств на одной шине. Типовые заводские назначения идентификаторов устройств приведены в табл. 5.34, хотя они не являются обязательными. Ряд версий BIOS считает загрузочным только устройство с SCSI ID=0. В настоящее время прорабатывается спецификация PnP для устройств SCSI, позволяющая автоматизировать процесс назначения идентификаторов. Она обеспечивает сосуществование традиционных (Legacy SCSI) устройств, идентификаторы которых задаются джамперами, с автоматически конфигурируемыми PnP-устройствами.

- *Контроль паритета* — SCSI Parity. Если хотя бы одно устройство не поддерживает контроль паритета, он должен быть отключен для всех устройств на шине. Контроль паритета, особенно для дисковых устройств, является необходимым средством защиты от искажения данных при передаче по шине.
- *Включение терминаторов* — Termination. В современных устройствах применяются активные терминаторы, которые могут включаться одним джампером или программно-управляемым сигналом. Терминаторы должны включаться только на крайних устройствах в цепочке. Современные хост-адаптеры позволяют автоматически включать свой терминатор, если они являются крайними, и отключать, если используются внутренний и внешний разъем канала. Это позволяет подключать и отключать внешние устройства, не заботясь о переключении терминаторов. Ранее приходилось открывать корпус и переставлять джампер, а пассивные терминаторы устанавливать в специальные гнезда (и извлекать их оттуда). При отсутствии внутренних терминаторов пользователь вынужден был применять внешние, устанавливаемые на кабель.

ВНИМАНИЕ ———————————————————————————————————

Правильная установка терминаторов крайне существенна — отсутствие/избыток терминаторов может привести к неустойчивости или неработоспособности интерфейса.

- *Питание терминаторов* (Terminator power). Когда используются активные терминаторы (для современных устройств — всегда), питание терминаторов должно быть включено (джампером или программно) хотя бы на одном устройстве.
- *Согласование скорости синхронного обмена* (SCSI synchronous negotiation). Режим синхронного обмена, обеспечивающий высокую производительность, включается по взаимному согласию устройств. Если хотя бы одно устройство на шине его не поддерживает, рекомендуют запретить согласование на хост-адаптере. Если обмен будет инициирован целевым устройством, поддерживающим синхронный режим, нормальный хост-адаптер поддержит этот режим. Целевому устройству можно запретить запрос синхронного режима специальным джампером, который может называться «Enable TI-SDTR» (Target Initiated Synchronous Data Transfer Request Negotiation).
- *Разрешение отключения* (Enable disconnection). Позволяет устройствам отключаться от шины при неготовности данных во время длительных операций с носителем, что весьма эффективно в многозадачном режиме при нескольких ПУ на шине. В случае одного устройства отключение приводит только к дополнительным затратам времени на повторное соединение.
- *Согласование ширины шины данных* тоже выполняется по протоколу шины исходя из возможностей обоих участников обмена. Целевому устройству можно запретить запрос 16-битного режима специальным джампером, который может называться «Enable TI-WDTR» (Target Initiated Wide Data Transfer Request Negotiation).
- *Запрет 16-разрядного режима* (Disable wide). Позволяет подключить «широкое» устройство к «узкой» шине.
- *Принудительное переключение в линейный режим* (Force SE). Позволяет перевести устройство LVD в режим SE, независимо от состояния DIFFSENS.
- *Запрет синхронизации по обоим фронтам* (Disable U160). Позволяет принудительно перевести устройство Ultra3 SCSI в режим Ultra2.
- *Старт по команде* (Start on command) или запрет автоматического запуска шпиндельного двигателя (Disable Auto Spin up). При включении этой опции запуск двигателя устройства выполняется только по команде от хост-адаптера, что позволяет снизить пик нагрузки блока питания в момент включения. Хост будет запускать устройства последовательно.
- *Задержанный старт* (Delayed Start) в сочетании с джамперами выбора задержки позволяет автоматически запускать двигатель через указанный интервал после подачи питания (разным устройствам задают различные значения задержки).

Таблица 5.34. Заводская установка идентификаторов устройств

Устройство	SCSI ID
Хост-адаптер	7
Накопитель на магнитных дисках	6
Ленточный или R/W оптический накопитель	4
CD-ROM	3
Сканер, принтер	2
НЖМД, поддерживаемый BIOS хост-адаптера	0 или 1

5.3.1.5.1. Хост-адаптер SCSI

Хост-адаптер является важнейшим узлом интерфейса, определяющим производительность системы

SCSI. Существует широкий спектр адаптеров. К простейшим можно подключать только устройства, некритичные к производительности. Такие адаптеры входят в комплект поставки сканеров, а подключение к ним диска может оказаться невозможным. Высокопроизводительные адаптеры имеют собственный специализированный процессор, большой объем буферной памяти и используют высокоэффективные режимы прямого управления шиной для доступа к памяти компьютера. Адаптеры SCSI существуют для всех шин: ISA (8-16 бит), EISA, MCA, PCI, VLB, PCMCIA — и для параллельного порта. Ряд системных плат имеют встроенный SCSI-адаптер, подключенный к одной из локальных шин. При выборе интерфейса, к которому подключается хост-адаптер, учитывайте производительность — интерфейс не должен стать узким местом при обмене с высокопроизводительными устройствами SCSI. Наибольшую эффективность имеют хост-адаптеры для шины PCI. Конечно, за мощный адаптер для сервера приходится платить — его цена может превышать цену рядового настольного компьютера. Еще дороже хост-адаптеры со встроенными контроллерами RAID-массивов, которые содержат мощный RISC-процессор и большой объем локальной памяти.

Конфигурирование хост-адаптеров с точки зрения шины SCSI не отличается от конфигурирования других устройств. Для современных адаптеров вместо джам-перов используется программное конфигурирование. Утилита конфигурирования обычно входит в расширение BIOS, установленное на плате адаптера, и приглашение к ее вызову выводится на экран во время POST.

Как и всякая карта расширения, хост-адаптер должен быть сконфигурирован и с точки зрения шины расширения, к которой он подключается. Системные ресурсы для шинного SCSI-адаптера включают:

- область памяти для расширения ROM BIOS, необходимого для поддержки конфигурирования устройств и дисковых функций. Если в системе установлено несколько однотипных хост-адаптеров, для них используется ROM BIOS с одного адаптера. Разнотипные хост-адаптеры не всегда могут работать вместе;
- область разделяемой буферной памяти;
- область портов ввода-вывода (I/O port);
- IRQ — запрос прерывания;
- DMA — канал прямого доступ к памяти (для шин ISA/EISA), часто используемый для захвата управления шиной (bus mastering).

Всем устройствам SCSI требуются специальные драйверы. Базовый драйвер дисковых устройств входит в BIOS хост-адаптера. Расширения, например ASPI (Advanced SCSI Programming Interface), загружаются отдельно. От драйверов сильно зависит производительность устройств SCSI. «Умное» ПО способно эффективно загружать работой устройства, а иногда и «срезать углы» — выполнять копирование данных между устройствами без выхода на системную шину компьютера. Наиболее предпочтительны драйверы, работающие в режиме прямого управления шиной (bus mastering); их применение позволяет реализовать все преимущества SCSI в многозадачных системах.

5.3.1.5.2. Подключение устройств к шине

Подключение устройств к шине SCSI относительно несложно, но имеются нюансы при смешении разнотипных устройств на одной шине. Пропускная способность шины SCSI, «освоенная» компьютером, определяется, естественно, возможностями хост-контроллера. Шина SCSI обеспечивает хорошую совместимость устройств с параллельными интерфейсами разных поколений, «узких» и «широких», но зачастую одно старое устройство способно свести на нет мощь новых устройств, подключенных к шине. По типу интерфейса совместимыми являются только SE и LVD.

ВНИМАНИЕ ——————————————————————————————

Смешивать устройства LVD с HVD на одной шине нельзя!

Устройства LVD можно использовать на одной шине с SE, но при этом все устройства перейдут в режим SE и шина не сможет работать в режиме Ultra2, свойственном устройствам LVD. Интерфейс LVD, являясь дифференциальным, требует, чтобы каждый обратный провод (сигнал +) приходил на вход своего приемника; в версии SE все обратные провода на устройстве соединялись вместе и подключались к шине GND. Если на шине с устройствами LVD имеется хотя бы одно устройство SE, то линия DIFFSENS окажется заземленной и все устройства LVD перейдут в режим SE. При конфигурировании устройство-LVD может быть принудительно переведено в режим SE установкой джампера «Force SE».

Если на шине присутствуют устройства Ultral60 и Ultra2 (или еще ниже), то шина будет работать в самом низком из этих режимов. Принудительно запретить режим Ultral60 (понизить до Ultra2) можно джампером «Disable U160».

Подключение «узкого» устройства к «узкой» шине — самая простая задача, поскольку здесь обычно встречаются лишь два типа разъемов (не считая Mac SCSI) — внешний (типа Centronics) и внутренний. Устройства должны быть сконфигурированы (см. выше), каждому должен быть назначен уникальный (на шине) SCSI ID, формально — любой в диапазоне 0-7. Длина шины не должна превышать

допустимого предела, на обоих концах шины (и только там!) должны быть установлены и включены терминаторы. На линию TERMPWR должно подаваться питание (чаще от хост-адаптера), что можно проверить, замерив напряжение на соответствующих контактах разъема.

Подключение «широкого» устройства к «широкой» шине может оказаться несколько сложнее, поскольку здесь больше разнообразия в разъемах. Из-за этого может потребоваться применение переходных адаптеров от одного типа разъема к другому. Также могут возникнуть сложности с подключением терминаторов, особенно для устройств LVD, среди которых внутренние терминаторы встречаются редко, а внешние могут занимать отдельный разъем на кабеле. Идентификаторы устройств можно задавать в диапазоне 0-15. *Подключение «узкого» устройства к «широкой» шине* требует применения переходного адаптера с 68- на 50-контактный разъем. Старший байт в этом адаптере не должен терминироваться, если подключаемое устройство не крайнее на шине. Если же устройство крайнее, то в адаптере старший байт должен терминироваться и на самом устройстве должен быть установлен терминатор. Выбор положения устройства (крайнее или промежуточное) может определяться имеющимся адаптером. Идентификаторы устройств должны устанавливаться в диапазоне 0-7 для всех устройств, поскольку невидимость идентификаторов 8-15 узким устройством не позволит процедуре арбитража нормально работать (см. выше). Поскольку все «узкие» устройства всегда.SE, линия DIFFSENS окажется заземленной и все устройства LVD перейдут в режим SE. Существуют, однако, и адаптеры-мосты, при подключении SE-устройства позволяющие остальным находиться в режиме LVD. Определить режим можно, замерив напряжение на 16-м контакте 68-контактного разъема (46-м 80-контактного).

Подключение «широкого» устройства к «узкой» шине также потребует применения специального адаптера, и на «широком» устройстве следует установить джампер «Disable Wide». Дополнительно может потребоваться терминация старшего байта и относящихся к нему управляющих линий, чтобы обеспечить на них надежное пассивное состояние («висящие» входы восприимчивы к помехам). Некоторые версии встроенного микропрограммного обеспечения позволяют работать устройствам и без дополнительных терминаторов. Идентификаторы всех устройств должны быть в диапазоне 0-7 (по тем же соображениям, что и в предыдущем случае).

5.3.1.5.3. Терминаторы

В переводе с английского *терминатор* — это оконечная нагрузка. Терминаторы приходится устанавливать на концах интерфейсных линий связи в тех случаях, когда длина волны электромагнитного сигнала, передаваемого по линии, оказывается соизмеримой с длиной самой линии. Не вдаваясь в глубокие теоретические основы электротехники, напомним, что существует понятие «длинная линия», и для анализа процессов распространения в ней электрического сигнала приходится учитывать волновые явления. Если на конце А линии установлен источник сигнала, то импульсный сигнал от него распространяется вдоль линии в сторону В с определенной скоростью, составляющей десятки процентов от скорости света в вакууме. Если на конце В нагрузки нет (линия разомкнута), то, дойдя до него, импульс отразится (!) и начнет распространяться в обратную сторону. Отраженный импульс в точке В будет иметь ту же амплитуду и полярность, что и пришедший прямой сигнал. По пути вдоль линии отраженный сигнал будет суммироваться с прямым, распространяющимся от источника. В каждой точке линии мгновенное значение сигнала будет равно сумме мгновенных значений проходящих волн прямого и отраженного сигналов. Если конец, противоположный источнику, замкнут накоротко, то отражение тоже происходит, а отраженный импульс имеет ту же величину, но обратную полярность. Это вполне очевидно, поскольку сумма прямого и отраженного сигналов в точке В будет нулевой — ничего другого на коротком замыкании ожидать не приходится. Мы рассмотрели два крайних случая — обрыв и замыкание. Теперь заметим, что каждая длинная линия (однородная) характеризуется своим *волновым сопротивлением,* или *характеристическим импедансом*. Если нашу линию на конце В нагрузить резистором, сопротивление которого равно импедансу линии, то отражения от этого конца не будет, а весь сигнал поглотится нагрузкой. Это случай полного согласования нагрузки с линией. Если полного согласования нет, то будет отражение, величина и знак которых зависят от того, в какую сторону и насколько рассогласована нагрузка с линией. Предельные случаи — обрыв и замыкание — мы рассмотрели, все остальные варианты лежат между ними. Такова теория длинных линий, и ее никто пока не отменял.

Посмотрим, что из этого следует. Источник на конце А посылает последовательность импульсов, которая каким-то образом кодирует передаваемую информацию. Если время прохождения сигнала вдоль всей линии значительно меньше, чем длительность импульса, то отраженный сигнал большого вреда не нанесет — суммарный сигнал будет несколько меньше или больше переданного, но форма результирующего сигнала будет близка к передаваемой. Несогласованностью линий иногда даже сознательно пользуются: так, например, в шине PCI предусматривается удвоение сигнала из-за отражений от разомкнутых (в соответствии со спецификацией) концов сигнальных линий. Если же задержка распространения соизмерима с периодом следования импульсов, то в линии будет сигнальная «каша», которую «расхлебать» (выделить тот сигнал, который передавался) будет под силу далеко не каждому

приемнику, установленному на линии. Причем в разных точках линии смесь прямого и отраженного сигналов будет иметь разную форму. Чтобы избавиться от этих проблем, линии стараются согласовать с нагрузкой — свести отражения к нулю. Заметим, что отражения происходят от обоих несогласованных концов, а также ото всех неоднородностей линии.

Теперь вернемся к интерфейсной шине SCSI. Эта шина изначально разрабатывалась для связи устройств кабелем, длина которого может доходить до 25 метров (это не ATA с полуметровым шлейфом!). Частота передачи составляет 5,10,20,40 и 80 мегатранзакций в секунду для версий SCSI-1, Fast-, Ultra-, Ultra2- и Ultra3-SCSI-2/3 (грубо говоря, здесь можно оперировать и мегагерцами). В шине SCSI всегда требуется установка *терминаторов,* и в соответствии с определением этого слова они *должны устанавливаться на концах физических линий.* Терминаторы должны быть согласованы по импедансу с кабелем. Кроме функции согласования, описанной выше, они предназначены и для «подтягивания» уровня сигналов линий к высокому потенциалу. Этим обеспечивается пассивное состояние управляющих сигналов, когда шиной не управляет ни одно устройство. Для питания терминаторов в шине предусмотрены специальные линии TermPWR. По электрическим свойствам различают следующие типы терминаторов:

- *Пассивные* (SCSI-1) с импедансом 132 Ом, представляющие собой обычные резисторы. Каждая сигнальная линия соединяется резистором 330 Ом с шиной GND и 220 Ом с шиной TermPwr (+5 В). Эти терминаторы плохо согласуются с большинством кабелей, имеющих импеданс 85-110 Ом и непригодны для высокоскоростных режимов SCSI-2 (частоты 20 МГц и выше).
- *Активные* с импедансом 110 Ом — специальные терминаторы для обеспечения работы на частоте 20 МГц в SCSI-2. Они имеют источник опорного напряжения (стабилизатор), вырабатывающий нужный потенциал, и каждая линия соединяется резистором 110 Ом с выходом этого источника. Эти терминаторы рекомендуются для всех скоростей, а для частот 20 MT/s и выше — требуются.
- *FPT* (Forced Perfect Terminator) — улучшенный вариант активных терминаторов с ограничителями выбросов, применяемые в высокоскоростных версиях интерфейса.

Активный терминатор, применяемый для устройств LVD, тоже должен отслеживать линию DIFFSENS, и при напряжении ниже 0,7В переходить в режим SE, при напряжении 0,9-1,9В—в режим LVD. При напряжении на линии DIFFSENS выше 2,4 В терминатор должен «отпускать» шину, поскольку это соответствует режиму HVD.

По исполнению терминаторы могут быть как *внутренние* (размещенные на печатной плате устройства), так и *внешние* (устанавливаемые на разъемы кабеля или устройства). Внутренние терминаторы на каждом устройстве могут быть включены или выключены. В старых устройствах (SCSI-1) для включения терминаторов нужно было установить набор перемычек или вставить в специальную кроватку сборку резисторов. Активные терминаторы включаются-выключаются перестановкой одного джампера или даже бесконтактно — программно при конфигурировании устройства. Возможно даже автоматическое включение терминатора (если таковая возможность поддерживается устройством и разрешена при конфигурировании). Внешние терминаторы выглядят как разъемы с небольшой крышкой, под которой смонтирована их «начинка». Несмотря на внешнюю простоту, они имеют ощутимую цену — активный терминатор для Ultra-Wide SCSI стоит $10-15. Внешние терминаторы устанавливаются и снимаются только вручную.

Внутренние терминаторы, или, по крайней мере, панелька для их установки имеются практически во всех устройствах с не LVD-интерфейсом. В устройствах с LVD-интерфейсом терминаторы, как правило, отсутствуют в целях экономии: когда на шину устанавливается несколько устройств, терминатор используется лишь в последних. Однако при подключении одного устройства экономия на цене устройства незаметна, а вот расходы на приобретение терминатора вполне очевидны.

ВНИМАНИЕ ——————————————————————————————

Отсутствие терминаторов на устройствах с LVD не означает отказа от правил терминации!

Рассмотрим различные конфигурации подключения устройств к контроллеру SCSI. Контроллер может быть расположен на карте расширения, устанавливаемой в слот PCI или ISA, или же встроен в системную плату. Устройства, подключаемые к нему, могут быть как внутренними (разного рода дисковые и ленточные устройства), так и внешними (те же, а также сканеры и другие периферийные устройства). Терминаторы расставляются исходя из конкретных условий.

Правила подключения достаточно просты:.

- Концы кабельного шлейфа не должны висеть в воздухе (как на рис. 5.13, *б).*
- На устройствах, подключенных к концам шлейфа, должны быть включены внутренние терминаторы (на рисунке обозначено как TRM=ON) или же установлен внешний (рис. 5.13, г).
- На промежуточных устройствах терминаторы должны быть отключены (TRM=OFF).

Если контроллер SCSI смонтирован на дополнительной интерфейсной карте, то разъем, к которому подключаются внешние устройства, расположен достаточно близко к внутреннему, так что длинной линии между ними нет. В этом случае терминация внешнего разъема хлопот не доставляет: когда используется только внутреннее (рис. 5.13, *а)* или только внешнее (рис. 5.13, г) подключение, терминатор на контроллере **включают.** Когда используется и внешнее, и внутреннее подключение (рис. 5.13, в), терминатор на контроллере **отключают.**

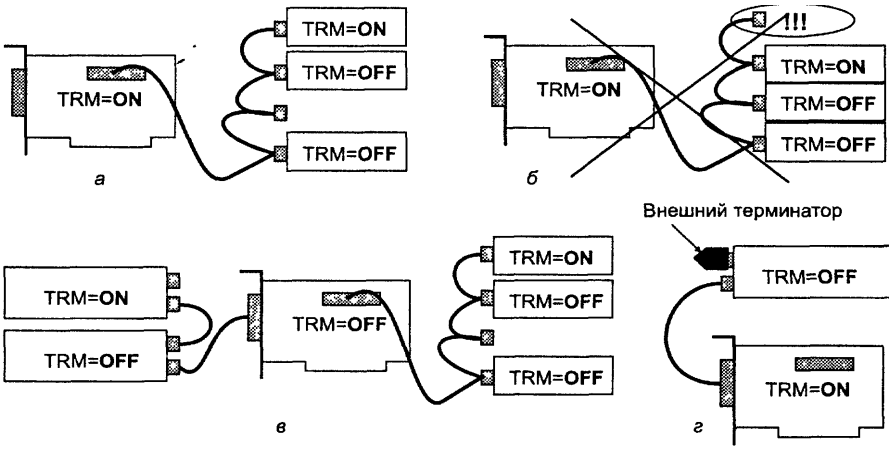

Рис. 5.13. Подключение устройств к карте контроллера SCSI: *а, в, г*— правильно; *б* — неправильно

Если используется внутреннее подключение, а внешние устройства подключаются не все время, то приходится переключать терминатор контроллера в соответствии с текущей конфигурацией. В старых контроллерах приходилось вскрывать системный блок и переставлять джамперы. В новых контроллерах вскрытия не требуется — терминаторы включаются/отключаются программно (утилитой SCSI Setup) или даже автоматически. Если по какой-либо причине переключать терминатор контроллера не хочется, можно его отключить и пользоваться внешним, устанавливая его на внешний разъем (снаружи корпуса компьютера), когда внешние устройства не подключены.

Когда контроллер SCSI установлен на системной плате, он имеет только один разъем, к которому подключается кабель-шлейф (рис. 5.14). Если требуется только внутреннее или только внешнее подключение (рис. 5.14, *а* и *б),* то терминатор на контроллере **включают.** Если используется и внутреннее, и внешнее подключение (рис. 5.14, б), терминатор на контроллере **отключают.** Если используется универсальный кабель-шлейф с внутренними и внешним разъемами (как на рис. 5.14, б), но внешних устройств нет, то терминатор на контроллере должен быть **отключен,** а на внешнем разъеме должен быть установлен **внешний терминатор.**

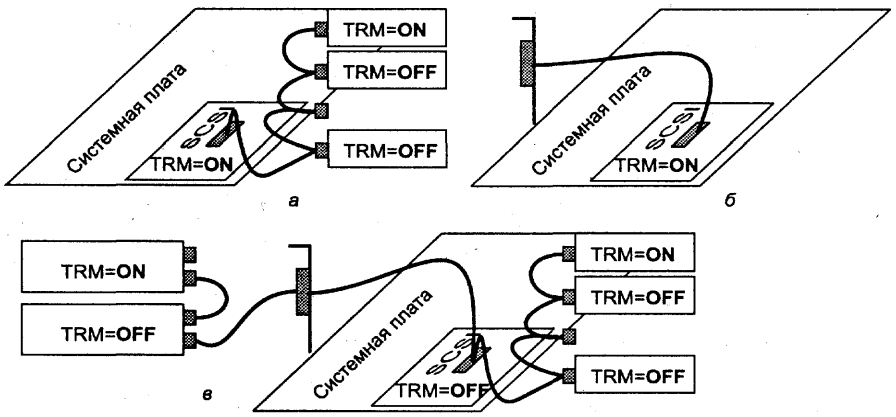

Рис. 5.14. Подключение устройств к интегрированному контроллеру SCSI

Кабели и терминаторы могут входить в комплект адаптеров SCSI или системных плат со встроенным контроллером SCSI, а могут приобретаться и отдельно. То, что входит в стандартный комплект поставки, не всегда подходит для конкретного применения. Так, например, в комплекте с системной платой CT-6BTS, имеющей контроллер Ultra-Wide SCSI, поставляется универсальный «широкий» (wide) шлейф (как на рис. 5.14, б, но с меньшим количеством внутренних разъемов) и «узкий» внутренний. Для подключения только внутренних дисководов Wide SCSI такого комплекта недостаточно: нужен внешний терминатор! Можно, конечно, отрезать часть шлейфа, идущую от разъема

для подключения контроллера к внешнему разъему, но жалко!

Контроллеры Wide SCSI обычно имеют и разъемы для подключения обычных («узких», или narrow) устройств. На той же системной плате CT-6BTS кроме 68-контактного разъема Wide SCSI имеется и 50-контактный для обычных устройств. «Узкий» (8-разрядный) интерфейс можно рассматривать как подмножество «широкого» (16-разрядного), у которого используется только младшая половина шины данных. В простых одноканальных контроллерах (как на этой плате) контакты «узкого» разъема запараллелены с частью контактов широкого. При этом можно использовать смесь широких и узких устройств, для чего терминаторы на контроллере разделены на две половины: терминаторы младшего байта (TrmL) и старшего байта (TrmH) могут управляться независимо. На рис. 5.15, а и *б,* приведены корректные способы смешанного подключения (устройства с терминаторами на концах шлейфов подразумеваются). На рис. 5.15, в приведена некорректная схема — здесь в младшем байте и в сигналах управления окажутся три терминатора (перегрузка передатчиков) или «висячий» конец (отражения!). Заметим, что штатными кабелями из комплекта поставки платы корректно смешанное подключение не выполнить (именно они и изображены на рис. 5.15, б).

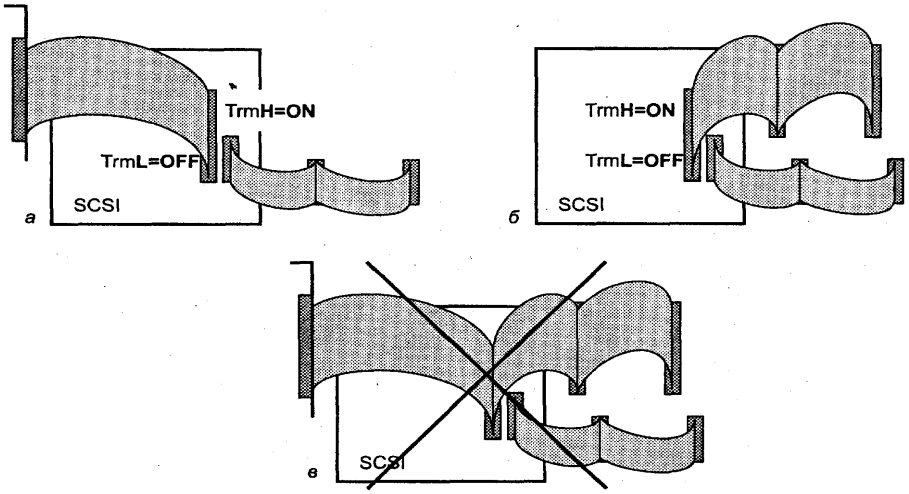

Рис. 5.15. Подключение узких и широких устройств SCSI: *а, б* — правильно; в — неправильно

В заключение заметим, что жизнь всегда многообразнее наших представлений о ней. Это касается и терминаторов. Если следовать вышеприведенным правилам (и не превышать разрешенную длину и число подключений), то при исправном оборудовании шина SCSI будет работать надежно, как ей и полагается. Если правила нарушать, то возможны варианты. Есть определенные модели контроллеров и устройств, для которых мелкие «шалости» с терминаторами сойдут с рук инсталляторам. Так, например, может быть прощен (или почти прощен) висящий конец без терминатора (если он не очень длинный). Устройства могут работать (ОС будет загружаться, диски читаться), но, возможно, со сбоями, которые не сразу заметны. Правда, если используется ОС Windows NT, то заглянув в журнал регистрации событий (Event Log), можно увидеть «букет» красных фонариков, связанных с устройствами SCSI. «Пышность» этого букета будет зависеть от тяжести нарушений и норова используемых устройств. Есть модели, придерживающиеся «строгих правил», и при нарушении терминации устройства работать не будут вообще. Как поступать в каждом конкретном случае, наверное, дело инсталлятора — на «лишний» терминатор или кабель другой конфигурации может просто не хватить денег. Но теория, увы, такова.

Контрольные вопросы

1. Является ли SCSI специализированным интерфейсом для дисковых устройств?
2. Охарактеризуйте отличия дисковых систем SCSI и ATA в части используемых хост-адаптеров, интерфейсов, способов конфигурирования, адресации, системы команд, производительности и т.д.
3. Когда был стандартизирован системный интерфейс малых компьютеров (SCSI)?
4. Для чего был предназначен этот интерфейс?
5. Какие устройства относятся к устройствам SCSI?
6. Может ли к одному контроллеру SCSI подключаться несколько устройств?
7. Дайте краткую характеристику первой (SCSI-1) и второй (SCSI-2) версиям стандарта.
8. Охарактеризуйте SCSI-3.
9. Какие группы команд предусмотрены в SCSI-3?
10. Дайте краткое описание архитектурной модели SCSI-3.
11. Охарактеризуйте скоростные параметры различных реализаций SCSI с параллельным интерфейсом.
12. Что понимается под аббревиатурами HVD и LVD?

13. Охарактеризуйте SCSI-3 с последовательным интерфейсом Fibre Channel (FCAL).
14. Какие версии SCSI наиболее распространены в настоящее время, в каком исполнении и с какой элементной базой?
15. В каких документах дано описание SCSI-3 с параллельным интерфейсом?
16. Опишите характерные особенности стандартов SPI, SPI-2, SPI-3 и SPI-4.
17. Дайте сравнительные характеристики линейной (SE), дифференциальной (LVD) и низковольтной дифференциальной (LVD) версиям линий интерфейса SCSI.
18. Какие разновидности кабелей используются в SCSI, каковы их характеристика и правила подключения?
19. Перечислите и охарактеризуйте типы разъемов, применяемых в SCSI.
20. Каково назначение терминаторов, и какие их разновидности применяются в кабельных системах SCSI?
21. Охарактеризуйте сигналы шины SCSI.
22. Какова адресация устройств на шине SCSI и их связь с приоритетами?
23. Какие взаимоотношения могут быть между ИУ (initiator) и ЦУ (target)?
24. Синхронный или асинхронный режим обмена поддерживает шина SCSI?
25. Охарактеризуйте фазы шины: Bus Free, Arbitration, Selection и Reselection.
26. Как и какими сигналами поддерживаются фазы Command, Data, Status и Message?
27. Охарактеризуйте фазы Data Out и Data In.
28. Чем отличается "мягкий" и "жесткий" сбросы?
29. Из каких фаз состоит каждый процесс ввода/вывода по шине SCSI?
30. Какой набор указателей для каждого процесса ввода/вывода сохраняет архитектура SCSI и для чего?
31. Какая система сообщений служит для управления интерфейсом SCSI и как она используется?
32. Кому может быть адресован процесс?
33. Как устанавливается синхронный режим и разрядность сообщений?
34. Перечислите стандартизированные типы устройств.
35. Какие типы устройств относятся к потоковым, а какие - к блочным?
36. Охарактеризуйте устройства прямого доступа (0).
37. Как можно обращаться к SCSI дискам через BIOS?
38. Что такое экстент (extent)?
39. Охарактеризуйте устройства последовательного доступа (1).
40. Зачем нужна точка раннего предупреждения (EW)?
41. Что такое BOM, EOM, BOPx и EOPx?
42. Охарактеризуйте принтеры (2) и процессорные устройства (3).
43. Охарактеризуйте устройства оптической памяти(7), устройства однократной записи (4) и накопители CD-ROM (5).
44. Охарактеризуйте сканеры (6), устройства смены носителя (8) и коммуникационные устройства (9).
45. Дайте общую характеристику системе команд SCSI.
46. Опишите поля блоков дескрипторов команд.
47. Какую длину имеют дескрипторы у нулевой группы команд, и какой - у пятой?
48. В какой фазе и чем завершается выполнение команды?
49. Охарактеризуйте возможные значения байта состояния.
50. Какие команды должно поддерживать любое ЦУ SCSI-2?
51. Ознакомьтесь со списком команд, приведенном в табл. 5.33.
52. Охарактеризуйте процессы на шине SCSI на примере выполнения одиночной команды Read.
53. Как изменятся процессы на шине, если в процессе выполнения команды Read используется отключение от шины (Disconnect)?
54. Как выполняются процессы с цепочкой связанных команд?
55. Охарактеризуйте маркированные и немаркированные очереди команд.
56. Какие параметры SCSI устройств необходимо изменять их программированием?
57. Какие параметры SCSI устройств необходимо устанавливать при их конфигурировании?
58. Какова заводская установка идентификаторов устройств SCSI?
59. Охарактеризуйте разновидности хост-адаптеров SCSI.
60. Как конфигурируется хост-адаптер на шине SCSI и на шине компьютера?
61. Охарактеризуйте драйверы хост-адаптеров.
62. Охарактеризуйте особенности подключения устройств к различным шинам SCSI.

В данном разделе использованы материалы из [3].

6.Особенности работы с устройствами, подключенными к шине PCI

Документацию по работе с шиной PCI распространяет организация PCI Special Interest Group (http://www.pcisig.com). Основные стандарты доступны по Сети только членам организации, а всем остальным они высылаются в виде книг или на компакт-дисках (услуга платная: стоимость диска с полным набором стандартов $50, стоимость книги с каким-либо отдельным стандартом — около $30). К сожалению, специфика российской действительности порождает определенные трудности не только с оплатой документации (обычное состояние всех бюджетных организаций — денег нет), но и с получением ее по почте (например, диск может быть вдребезги разбит штемпельным молотком). Сведения, изложенные ниже, собраны из многих источников.

6.1. Конфигурационное пространство устройства PCI

Для программистов интерес в первую очередь представляют функции PCI BIOS, поскольку они позволяют получить доступ к информации об адресном пространстве и пространстве ввода/вывода подключенных к шине PCI устройств. Описание этих функций дано в PCI BIOS Specification, несколько устаревший вариант которой (версию 2.0) [12] можно найти на некоторых компьютерных российских сайтах, а также в руководстве Phoenix BIOS User's Manual [13].

Интерфейс PCI BIOS обеспечивает аппаратно-независимый метод управления устройствами PCI (а также AGP) в любых возможных режимах работы архитектуры X86 (включая реальный, защищенный 16-разрядный, защищенный 32-разрядный режимы и режим с линейным адресным пространством). Основное назначение функций PCI BIOS — работа с конфигурационным пространством и генерация специальных циклов шины PCI.

Общий вид конфигурационного пространства устройства PCI показан на рис. 6.1. Конфигурационное пространство содержит три области:

- заголовок, не зависящего от типа устройства;
- область, определяемая устройством (значением Header Type);
- область, определяемая пользователем.

С точки зрения программиста, важными являются только первые две из перечисленных областей, а область пользователя фактически доступна только разработчикам устройства, поскольку детальное описание этой области есть только у них, и посторонним обычно не предоставляется. Впрочем, чтобы извлечь какую-то полезную информацию из полей, содержащих базовые адреса, также необходимо располагать документацией на устройство: каждый изготовитель трактует их по-своему. Для наглядности достаточно сравнить между собой распределение базовых адресов в видеоконтроллерах 3Dfx [14, 15] и Matrox [16, 17].

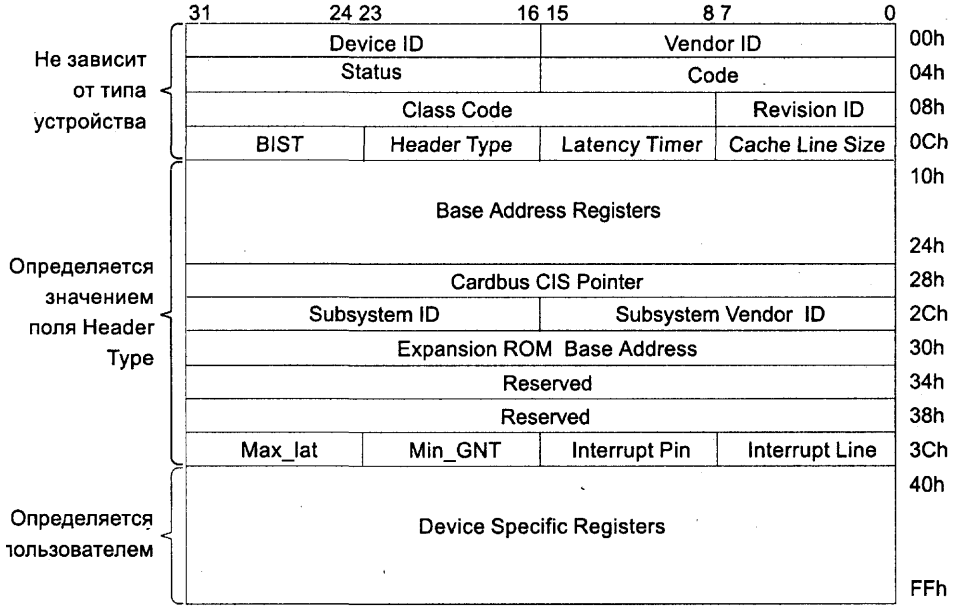

Рис. 6.1. Конфигурационное пространство устройства PCI

Если программисту требуется выполнять с устройством PCI только типовые операции, то ему совершенно не обязательно разрабатывать полноценный драйвер для устройства — для решения некоторых типовых задач достаточно той документации, которую фирмы-изготовители предоставляют через Интернет в режиме свободного доступа. На практике интерес представляют следующие поля кон-

фигурационного пространства:

- Vendor ID — код фирмы-изготовителя устройства;
- Device ID — код устройства;
- Class Code — код класса устройства;
- Base Address Registers — регистры базовых адресов;
- Interrupt Line — номер выделенного устройству прерывания IRQ.

Код изготовителя, код устройства и код класса применяются в процессе поиска заданного устройства. Если необходимо найти конкретное устройство, то поиск выполняется по кодам устройства и его изготовителя (см. вышеупомянутые описания контроллеров Matrox и 3Dfx); если требуется обнаружить все устройства определенного типа, то используется код класса. После того как искомое устройство обнаружено, при помощи регистров базовых адресов можно определить выделенные ему области в адресном пространстве памяти и пространстве ввода/вывода. Наибольший интерес, однако, представляет регистр Interrupt Line, позволяющий выяснить, какая линия IRQ была выделена устройству процедурой PnP BIOS в процессе начальной загрузки компьютера — в некоторых случаях это единственный документированный способ определения номера прерывания.

6.2.Функции PCI BIOS

Поскольку конфигурационное пространство не имеет привязки к какой-либо определенной области адресного пространства компьютера, доступ к нему связан с определенными трудностями. С целью упрощения работы с устройствами PCI в BIOS персональных компьютеров были внесены специальные дополнительные функции. Доступ к функциям PCI BIOS при 16-разрядном вызове выполняется через функцию Blh прерывания lAh. Для 32-разрядных вызовов используется 32-разрядная точка входа защищенного режима.

Функции PCI BIOS используют регистры процессора для передачи аргументов и получения результатов. При успешном выполнении функции флаг переноса CF сбрасывается в 0, в случае неудачи — устанавливается в 1.

Прерывание 1Ah, функция B101h: проверить присутствие PCI BIOS в системе

Для проверки присутствия PCI BIOS в системе по прерыванию 1Ah вызывается функция B101h.

Перед вызовом прерывания 1Ah требуется занести в регистр AX код 0Bl0lh. После выполнения функции в регистрах будут размещены следующие значения:

- в EDX — сигнатура «PCI» («P» — в DL, «C» — в DH и т. д., в старшем байте — пробел);
- в AH — признак присутствия (0 — BIOS присутствует, если в EDX правильная сигнатура, любое другое значение — PCI BIOS отсутствует);
- в AL — аппаратный механизм;
- в BH — номер версии интерфейса PCI (в двоично-десятичном коде);
- в BL — подномер версии интерфейса (в двоично-десятичном коде);
- в CL — номер последней шины PCI в системе (счет номеров начинается с нуля).

Флаг CF также будет содержать признак наличия PCI BIOS (0 — BIOS присутствует, 1 — отсутствует).

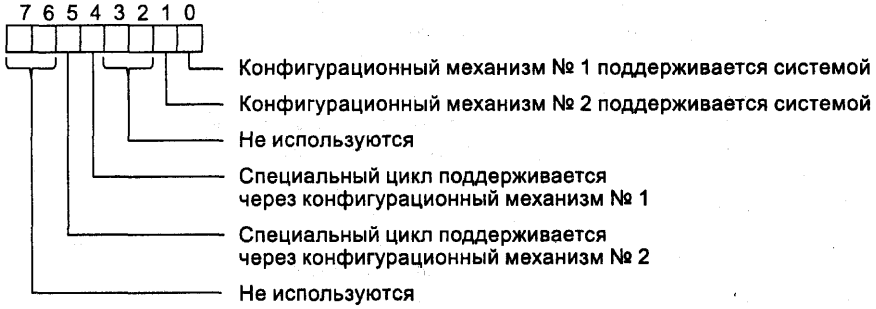

Рис. 6.2. Формат байта-описателя аппаратного механизма шины PCI

На рис. 6.2 показан формат байта-описателя аппаратного механизма шины: информация, возвращенная в регистре AL, показывает, какие механизмы функционирования шины PCI реализованы в данной аппаратуре. Спецификация PCI определяет два аппаратных механизма для доступа к конфигурационному пространству. Механизм 1 поддерживается, если установлен бит 0, механизм 2 — если установлен бит 1. Спецификация PCI определяет также механизмы генерации специальных циклов. Бит 4 установлен, если аппаратура может выполнять генерацию специальных циклов на основе механизма 1, бит 5 установлен, если аппаратура может выполнять генерацию специальных циклов на

основе механизма 2. Биты 2, 3, 6 и 7 зарезервированы и должны быть равны нулю.

Прерывание 1Ah, функция B102h: найти устройство PCI заданного типа

Для поиска устройства PCI заданного типа по прерыванию 1Ah вызывается функция B102h.

Перед вызовом прерывания требуется занести в регистры следующие значения:

- в AX — код 0B102h;
- в CX — идентификатор устройства (число от 0 до 65535);
- в DX — идентификатор изготовителя (от 0 до 65534);
- в SI — индекс (порядковый номер) устройства заданного типа (от 0 до N).

После выполнения функции в регистрах будут размещены следующие значения:

- в BH — номер шины, к которой подключено устройство (от 0 до 255);
- в BL — номер устройства в старших пяти битах и номер функции в трех младших;
- в AH — код возврата (может принимать значения SUCCESFUL, BAD_VENDORID и DEVICE_NOT_FOUND);
- в CF — статус возврата (0 — функция успешно выполнена, 1 — ошибка).

Если необходимо найти все устройства данного типа, то в SI заносится 0. После каждого выполнения функции значение SI надо увеличивать на 1, пока не будет получен код возврата DEVICE_NOT_FOUND (устройство не обнаружено).

Прерывание 1Ah, функция B103h: найти устройство PCI заданного класса

Для поиска устройства PCI заданного класса по прерыванию 1Ah вызывается функция B103h.

Перед вызовом прерывания требуется занести в регистры следующие значения:

- в AX — код 0B103h;
- в ECX — код класса (в младших трех байтах);
- в S I — индекс (порядковый номер) устройства заданного класса (от 0 до N).

После выполнения функции в регистрах будут размещены следующие значения:

- в BH — номер шины, к которой подключено устройство (от 0 до 255);
- в BL — номер устройства в старших пяти битах и номер функции в трех младших;
- в AH — код возврата (может принимать значения SUCCESFUL и DEVICE_NOT_ FOUND);
- в CF — статус возврата (0 — функция успешно выполнена, 1 — ошибка).

Если необходимо найти все устройства данного класса, то в SI заносится 0. После каждого выполнения функции значение SI надо увеличивать на 1, пока не будет получен код возврата DEVICE_NOT_FOUND (устройство не обнаружено).

Прерывание 1Ah, функция B106h: генерировать специальный цикл шины

Для генерации специального цикла шины по прерыванию 1Ah вызывается функция B106h.

Перед вызовом прерывания требуется занести в регистры следующие значения:

- в AX — код 0B106h;
- в BH — номер шины, к которой подключено устройство (от 0 до 255);
- в EDX — данные специального цикла.

После выполнения функции в регистрах будут размещены следующие значения:

- в AH — код возврата (может принимать значения SUCCESFUL и FUNC_NOT_ SUPPORTED);
- в CF — статус возврата (0 — функция успешно выполнена, 1 — ошибка).

Прерывание 1Ah, функция B108h: прочитать байт из конфигурационного пространства заданного устройства

Для считывания байта из конфигурационного пространства заданного устройства по прерыванию 1Ah вызывается функция B108h.

Перед вызовом прерывания требуется занести в регистры следующие значения:

- в AX — код 0B108h;
- в BH — номер шины, к которой подключено устройство (от 0 до 255);
- в BL — номер устройства в старших пяти битах и номер функции в трех младших;
- в DI — порядковый номер байта (от 0 до 255).

После выполнения функции в регистрах будут размещены следующие значения:

- в CL — считанный байт;
- в AH — код возврата (может принимать значения SUCCESFUL и BAD_REGISTER_ NUMBER);
- в CF — статус возврата (0 — функция успешно выполнена, 1 — ошибка).

Прерывание 1Ah, функция B109h: прочитать слово из конфигурационного пространства заданного устройства

Для считывания слова из конфигурационного пространства заданного устройства по прерыванию 1Ah вызывается функция B109h.

Перед вызовом прерывания требуется занести в регистры следующие значения:

- в AX — код 0B109h;
- в BH — номер шины, к которой подключено устройство (от 0 до 255);
- в BL — номер устройства в старших пяти битах и номер функции в трех младших;
- в DI — смещение слова в конфигурационном пространстве (0, 2, 4,..., 254).

После выполнения функции в регистрах будут размещены следующие значения:

- в CX — считанное слово;
- в AH — код возврата (может принимать значения SUCCESFUL и BAD_REGISTER_ NUMBER);
- в CF — статус возврата (0 — функция успешно выполнена, 1 — ошибка).

Прерывание 1Ah, функция B10Ah: прочитать двойное слово из конфигурационного пространства заданного устройства

Для считывания двойного слова из конфигурационного пространства заданного устройства по прерыванию 1Ah вызывается функция B10Ah.

Перед вызовом прерывания требуется занести в регистры следующие значения:

- в AX — код 0B10Ah;
- в BH — номер шины, к которой подключено устройство (от 0 до 255);
- в BL — номер устройства в старших пяти битах и номер функции в трех младших;
- в DI — смещение двойного слова в конфигурационном пространстве (0, 4, 8, ...,252).

После выполнения функции в регистрах будут размещены следующие значения:

- в ECX — считанное двойное слово;
- в AH — код возврата (может принимать значения SUCCESFUL и BAD_REGISTER_ NUMBER);
- в CF — статус возврата (0 — функция успешно выполнена, 1 — ошибка).

Прерывание 1Ah, функция B10Bh: записать байт в конфигурационное пространство заданного устройства

Для записи байта в конфигурационное пространство заданного устройства по прерыванию 1Ah вызывается функция B10Bh.

Перед вызовом прерывания требуется занести в регистры следующие значения:

- в AX — код 0B10Bh;
- в BH — номер шины, к которой подключено устройство (от 0 до 255);
- в B L — номер устройства в старших пяти битах и номер функции в трех младших;
- в DI — порядковый номер байта (от 0 до 255);
- в CL — записываемый байт.

После выполнения функции в регистрах будут размещены следующие значения:

- в AH — код возврата (может принимать значения SUCCESFUL и BAD_REGISTER_ NUMBER);
- в CF — статус возврата (0 — функция успешно выполнена, 1 — ошибка).

Прерывание 1Ah, функция B10Ch: записать слово в конфигурационное пространство заданного устройства

Для записи слова в конфигурационное пространство заданного устройства по прерыванию 1Ah вызывается функция B10Ch.

Перед вызовом прерывания требуется занести в регистры следующие значения:

- в AX — код 0B10Ch;
- в BH — номер шины, к которой подключено устройство (от 0 до 255);
- в BL — номер устройства в старших пяти битах и номер функции в трех младших;
- в DI — смещение слова в конфигурационном пространстве (0, 2, 4,..., 254).
- в CX — записываемое слово.

После выполнения функции в регистрах будут размещены следующие значения:

- в AH — код возврата (может принимать значения SUCCESFUL и BAD_REGISTER_ NUMBER);
- в CF — статус возврата (0 — функция успешно выполнена, 1 — ошибка).

Прерывание 1Ah, функция B10Dh: записать двойное слово в конфигурационное пространство заданного устройства

Для записи двойного слова в конфигурационное пространство заданного устройства по прерыванию 1Ah вызывается функция B10Dh.

Перед вызовом прерывания требуется занести в регистры следующие значения:

- в AX — код 0B10Dh;
- в BH — номер шины, к которой подключено устройство (от 0 до 255);
- в BL — номер устройства в старших пяти битах и номер функции в трех младших;
- в DI — смещение двойного слова в конфигурационном пространстве (0, 4, 8,.... 252).
- в ECX — записываемое двойное слово.

После выполнения функции в регистрах будут размещены следующие значения:

- в AH — код возврата (может принимать значения SUCCESFUL и BAD_REGISTER_ NUMBER, см. табл. 3.1);
- в CF — статус возврата (0 — функция успешно выполнена, 1 — ошибка).

Расшифровка значения кодов возврата приведена в табл. 6.1.

Таблица 6.1. Список кодов возврата

Код возврата	Значение в AH	Расшифровка
SUCCESFUL	00h	Операция успешно выполнена
FUNC_NOT_SUPPORTED	81h	Функция не поддерживается данным устройством
BAD_VENDOR_ID	83h	Неверный идентификатор изготовителя
DEVICE_NOT_FOUND	86h	Устройство не найдено
BAD_REGISTER_NUMBER	87h	Некорректное значение индекса

С точки зрения программиста особый интерес представляют подфункции 01h (проверка наличия PCI BIOS), 02h (найти устройство заданного типа), 03h (найти устройство заданного класса) и 08h-0Ah (прочитать данные из конфигурационного пространства устройства).

Проверять наличие PCI BIOS нужно при запуске программы: если его нет, то нет и возможности использовать относящиеся к нему функции. Проверку наличия в системе нужного устройства и определение его координат на шине PCI (номера шины, номера устройства и функции) также следует выполнять при запуске программы. После обнаружения устройства становится доступной для считывания вся информация из его конфигурационного пространства.

6.3. Поиск устройства PCI по коду класса

Поиск устройства определенного типа можно осуществлять по коду класса. Код класса состоит из трех байтов (рис. 6.3): старший байт задает базовый класс (Base Class), средний байт ~ подкласс (Sub-Class), младший байт — интерфейс (Interface). В табл. 6.2 приведен перечень базовых классов устройств PCI (из имеющегося в свободном доступе на сайте www.pcisig.com фрагмента документации [67]), а табл. 6.3 содержит полное описание кодов классов.

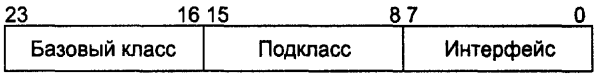

Рис. 6.3. Общая структура поля кода класса

Таблица 6.2. Коды базовых классов

Код	Тип устройства
00h	Устройство было изготовлено до введения стандарта на коды классов
0lh	Контроллер устройства массовой памяти
02h	Сетевой контроллер
03h	Видеоконтроллер
04h	Устройство мультимедиа
05h	Контроллер памяти
06h	Устройство типа «мост»
07h	Простой коммуникационный контроллер
08h	Основная системная периферия
09h	Устройство ввода
0Ah	Контроллер Docking station

0Bh	Процессор
0Ch	Контроллер последовательной шины
0Dh	Контроллер устройства беспроводной передачи данных
0Eh	Интеллектуальный контроллер ввода/вывода
0Fh	Сопутствующий коммуникационный контроллер
10h	Контроллер устройства цифрации/дешифрации
11h	Контроллер устройства сбора и обработки сигналов
12h-FEh	Зарезервированы
FFh	Устройство не подходит ни к одному из перечисленных классов

Таблица 6.3. Полное описание кодов классов устройств

Базовый класс	Подкласс	Интерфейс	Значение
00h	00h	00h	Все устройства, выпущенные до принятия стандарта, за исключением VGA-совместимых
	01h	00h	VGA-совместимые устройства
01h	00h	00h	SCSI-контроллер
	0lh	xxh(см.рис.6.4)	IDE-контроллер
	02h	00h	Контроллер дисковода гибких дисков
	03h	00h	IPI-контроллер
	04h	00h	RAID-контроллер
	80h	00h	Устройство массовой памяти другого типа
02h	00h	00h	Контроллер Ethernet
	0lh	00h	Контроллер Token Ring
	02h	00h	Контроллер FDDI
	03h	00h	Контроллер ATM
	04h	00h	Контроллер ISDN
	80h	00h	Сетевой контроллер другого типа
03h	00h	00h	VGA-совместимый контроллер
		0lh	8514-совместимый контроллер
	0lh	00h	Контроллер XGA
	02h	00h	3D-контроллер
	80h	00h	Другой видеоконтроллер
04h	00h	00h	Видеоустройство
	0lh	00h	Аудиоустройство
	02h	00h	Устройство для компьютерной телефонии
	80h	00h	Мультимедиа-устройство другого типа
05h	00h	00h	Контроллер оперативной памяти
	0lh	00h	Контроллер Flash-памяти
	80h	00h	Контроллер памяти другого типа
06h	00h	00h	Мост хоста
	0lh	00h	Мост ISA
	02h	00h	Мост EISA
	03h	00h	Мост MCA
	04h	00h	Мост PCI-to-PCI
		0lh	Мост PCI-to-PCI, поддерживающий вычитающее декодирование
	95h	00h	Мост PCMCIA
	06h	00h	Мост NuBus
	07h	00h	Мост CardBus
	08h	00h	Мост RACEway в режиме прозрачности
		0lh	Мост RACEway в режиме оконечного узла
	09h	40h	Полупрозрачный мост PCI-to-PCI, обращенный к хост-процессору «основной» стороной
		80h	Полупрозрачный мост PCI-to-PCI, обращенный к хост-процессору «вторичной» стороной

Таблица 6.3. Полное описание кодов классов устройств (продолжение)

Базовый класс	Подкласс	Интерфейс	Значение
	80h	00h	Мост другого типа
07h	00h	00h	Стандартный XT-совместимый контроллер последовательного порта
		0lh	16450-совместимый контроллер последовательного порта
		02h	16550-совместимый контроллер последовательного порта
		03h	16650-совместимый контроллер последовательного порта
		04h	16750-совместимый контроллер последовательного порта
		05h	16850-совместимый контроллер последовательного порта
		06h	16950-совместимый контроллер последовательного порта
	0lh	00h	Параллельный порт
		0lh	Двунаправленный параллельный порт
		02h	Параллельный порт типа ECP 1.X
		03h	Контроллер 1EEE1284
		FEh	Целевое устройство 1EEE1284 (не контроллер)
	02h	00h	Многопортовый последовательный контроллер
	03h	00h	Стандартный модем
		0lh	Hayes-совместимый модем, 16450-совместимый интерфейс
		02h	Hayes-совместимый модем, 16550-совместимый интерфейс
		03h	Hayes-совместимый модем, 16650-совместимый интерфейс
		04h	Hayes-совместимый модем, 16750-совместимый интерфейс
	80h	00h	Другое коммуникационное устройство
08h	00h	00h	Стандартный контроллер прерываний типа 8259
		0lh	Контроллер прерываний ISA
		02h	Контроллер прерываний EISA
		l0h	Контроллер прерываний ввода/вывода APIC
	0lh	00h	Стандартный контроллер ПДП типа 8237
		0lh	ISA-контроллер ПДП
		02h	EISA-контроллер ПДП
	02h	00h	Стандартный системный таймер типа 8254
		0lh	Системный таймер ISA
		02h	Системные таймеры EISA (два таймера)
	03h	00h	Стандартный контроллер часов реального времени
		0lh	Контроллер часов реального времени ISA
	04h	00h	Стандартный контроллер «горячего подключения» PCI
	80h	00h	Системная периферия другого типа
09h	00h	00h	Контроллер клавиатуры
	01h	00h	Контроллер дигитайзера
	02h	00h	Контроллер мыши
	03h	00h	Контроллер сканера
	04h	00h	Стандартный контроллер игрового порта
		01h	Контроллер игрового порта с программируемым интерфейсом
	80h	00h	Контроллер другого устройства ввода данных
0Ah	00h	00h	Стандартный контроллер Docking station
	80h	00h	Нестандартный контроллер Docking station
0Bh	00h	00h	Процессор типа 386
	0lh	00h	Процессор типа 486
	02h	00h	Процессор типа Pentium
	10h	00h	Процессор типа Alpha
	20h	00h	Процессор типа Power PC
	30h	00h	Процессор типа MIPS
	40h	00h	Сопроцессор
0Ch	00h	00h	IEEE1394 (FireWire)
		10h	1EEE1394, следующий за спецификацией 1394 OpenHCI
	0lh	00h	ACCESSbus
	02h	00h	SSA

Таблица 6.3. Полное описание кодов классов устройств (продолжение)

Базовый класс	Подкласс	Интерфейс	Значение
	03h	00h	Устройство USB, следующее за спецификацией Universal Host Controller
		10h	Устройство USB, следующее за спецификацией open Host Controller
		80h	Устройство USB без определенного программного интерфейса
		FEh	Устройство USB (не хост-контроллер)
	04h	00h	Fibre Channel
	05h	00h	SMBus
0Dh	00h	00h	IRDA-совместимый контроллер
	01h	00h	IR-контроллер потребителя
	10h	00h	RF-контроллер
	80h	00h	Другой тип контроллера беспроводной связи
0Eh	00h	xxh	Интеллектуальный контроллер ввода/вывода (120 (ISO)) с архитектурой, соответствующий спецификации 1.0
		00h	Буфер сообщений FIFO со смещением 040h
0Fh	0lh	00h	TV-контроллер
	02h	00h	Аудиоконтроллер
	03h	00h	Голосовой контроллер
	04h	00h	Контроллер данных
10h	00h	00h	Сетевой или компьютерный шифратор/дешифратор
	10h	00h	Игровой шифратор/дешифратор
	80h	00h	Шифратор/дешифратор другого типа
11h	00h	00h	DPIO-модуль
	80h	00h	Контроллер устройства сбора и обработки сигналов другого типа

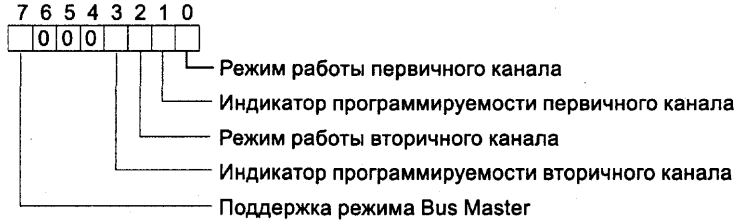

Рис. 6.4. Формат байта интерфейса кода класса IDE-контроллера

6.3.1. Пример программы поиска

В программе PCITest, приведенной в листинге 6.1, поиск VGA-совместимого видеоконтроллера осуществляется по коду класса, присвоенному устройствам такого типа (030000h). Обнаружив первое устройство с соответствующим кодом (в системе, вообще говоря, может быть несколько видеоконтроллеров), программа прекращает поиск и выводит на экран значения некоторых полей конфигурационного пространства. Для запуска данного примера пригоден любой компьютер с шиной PCI и PCI- или AGP-видеоконтроллером.

Листинг 6.1. Поиск видеоконтроллера, определение кода изготовителя, используемого адресного пространства и номера прерывания IRQ

```
IDEAL
P386
LOCALS
MODEL MEDIUM

; Подключить файл мнемонических обозначений цветов
include "lst_1_05.inc"

SEGMENT sseg para stack 'STACK'
DB 400h DUP(?)
ENDS
```

```
DATASEG
; Параметры устройства PCI
VendorID        DW ? ;идентификатор изготовителя
DeviceID        DW ? ;идентификатор устройства
ClassCode       DD ? ;тип устройства
BaseAddress0    DD ? ;базовый адрес 0
BaseAddress1    DD ? ;базовый адрес 1
BaseAddress2    DD ? ;базовый адрес 2
BaseAddress3    DD ? ;базовый адрес 3
BaseAddress4    DD ? ;базовый адрес 4
BaseAddress5    DD ? ;базовый адрес 5
InterruptLine   DB ? ;номер используемого прерывания IRQ
; Координаты устройства PCI
BusNumber       DB ? ;номер шины
DeviceNumber    DB ? ;номер устройства и номер функции
; Счетчик операций нажатия/отпускания клавиш
PressCounter DW ?
; Текстовые сообщения
PCI DB 0,25,"Тестирование функций PCI BIOS",0

        mov     AX,0B103h
        mov     ECX,030000h
        mov     SI,0
        int     1Ah
        jc      @@DeviceNotFound
        mov     [BusNumber],BH
        mov     [DeviceNumber],BL

; Получить идентификатор изготовителя
        mov     AX,0B109h ;читать слово
        mov     DI,0      ;смещение слова
        int     1Ah
        jc      @@BadRegisterNumber
        mov     [VendorID],CX
; Получить идентификатор устройства
        mov     AX,0B109h ;читать слово
        mov     DI,2      ;смещение слова
        int     1Ah
        jc      @@BadRegisterNumber
        mov     [DeviceID],CX
; Получить тип устройства (самопроверка)
        mov     AX,0B10Ah ;читать двойное слово
        mov     DI,8      ;смещение слова
        int     1Ah
        jc      @@BadRegisterNumber
        shr     ECX,8
        mov     [ClassCode],ECX
; Получить базовый адрес 0
        mov     AX,0B10Ah ;читать двойное слово
        mov     DI,10h    ;смещение слова
        int     1Ah
        jc      @@BadRegisterNumber
        mov     [BaseAddress0],ECX
; Получить базовый адрес 1
        mov     AX,0B10Ah ;читать двойное слово
        mov     DI,14h    ;смещение слова
        int     1Ah
        jc      @@BadRegisterNumber
        mov     [BaseAddress1],ECX
; Получить базовый адрес 2
        mov     AX,0B10Ah ;читать двойное слово
        mov     DI,18h    ;смещение слова
        int     1Ah
        jc      @@BadRegisterNumber
        mov     [BaseAddress2],ECX
; Получить базовый адрес 3
        mov     AX,0B10Ah ;читать двойное слово
        mov     DI,1Ch    ;смещение слова
        int     1Ah
        jc      @@BadRegisterNumber
        mov     [BaseAddress3],ECX
; Получить базовый адрес 4
        mov     AX,0B10Ah ;читать двойное слово
        mov     DI,20h    ;смещение слова
        int     1Ah
        jc      @@BadRegisterNumber
        mov     [BaseAddress4],ECX
; Получить базовый адрес 5
        mov     AX,0B10Ah ;читать двойное слово
        mov     DI,24h    ;смещение слова
        int     1Ah
        jc      @@BadRegisterNumber
```

```
        mov     [BaseAddress5],ECX

; Получить номер используемого устройством
; прерывания IRQ
        mov     AX,0B108h ;читать байт
        mov     DI,3Ch    ;смещение байта
        int     1Ah
        jc      @@BadRegisterNumber
        mov     [InterruptLine],CL

; Вывести полученные данные на экран
; в шестнадцатеричном коде
        ; Отобразить основные параметры устройства
        mov     [ScreenString],6
        mov     [ScreenColumn],12
        mov     AL,[BusNumber]
        call    ShowByteHexCode

        inc     [ScreenString]
        mov     [ScreenColumn],18
        mov     AL,[DeviceNumber]
        shr     AL,3
        call    ShowByteHexCode

        inc     [ScreenString]
        mov     [ScreenColumn],15
        mov     AL,[DeviceNumber]
        and     AL,111b
        call    ShowByteHexCode

        inc     [ScreenString]
        mov     [ScreenColumn],28
        mov     AX,[VendorID]
        call    ShowHexWord

        inc     [ScreenString]
        mov     [ScreenColumn],26
        mov     AX,[DeviceID]
        call    ShowHexWord

        inc     [ScreenString]
        mov     [ScreenColumn],16
        mov     EAX,[ClassCode]
        call    ShowHexDWord

        ; Отобразить базовые адреса
        inc     [ScreenString]
        mov     [ScreenColumn],17
        mov     EAX,[BaseAddress0]
        call    ShowHexDWord

        inc     [ScreenString]
        mov     [ScreenColumn],17
        mov     EAX,[BaseAddress1]
        call    ShowHexDWord

        inc     [ScreenString]
        mov     [ScreenColumn],17
        mov     EAX,[BaseAddress2]
        call    ShowHexDWord

        inc     [ScreenString]
        mov     [ScreenColumn],17
        mov     EAX,[BaseAddress3]
        call    ShowHexDWord

        inc     [ScreenString]
        mov     [ScreenColumn],17
        mov     EAX,[BaseAddress4]
        call    ShowHexDWord

        inc     [ScreenString]
        mov     [ScreenColumn],17
        mov     EAX,[BaseAddress5]
        call    ShowHexDWord

        ; Вывести номер используемого прерывания IRQ
        inc     [ScreenString]
        mov     [ScreenColumn],32
        mov     AL,[InterruptLine]
        cmp     AL,0FFh       ;прерывание используется?
        je      @@IRQNotUsed
        call    ShowByteHexCode
        jmp short @@End
```

```
@@IRQNotUsed:
        mov     SI,offset NoIRQ
        call    ShowString
        jmp short @@End

; Обработка ошибок
@@BadRegisterNumber:
        ; Неверный номер регистра
        mov     SI,offset BadRegister
        jmp     @@ErrorMessage
@@DeviceNotFound:
        ; Не найден видеоконтроллер
        mov     SI,offset NotSVGA
        jmp     @@ErrorMessage
@@PCIBIOSNotFound:
        ; Не поддерживается PCI BIOS
        mov     SI,offset NotPCI
@@ErrorMessage:
        ; Выдать сообщение о типе ошибки
        call    ClearScreen
        call    ShowColorString
        mov     SI,offset Exit
        call    ShowColorString

; Окончание работы
@@End:  ; Ожидать нажатия любой клавиши
        call    GetChar
; Переустановить текстовый режим и очистить экран
        mov     AX,3
        int     10h
; Выход в DOS
        mov     AH,4Ch
        int     21h
ENDP PCITest
ENDS

; Подключить процедуры вывода данных на экран
include "lst_1_02.inc"

END
```

6.4. Особенности реализации режима DMA на системных платах с шиной PCI

Режим прямого доступа к памяти (Direct Memory Access, сокращенно DMA) обеспечивает, возможность обмена данными между периферийными устройствами и оперативной памятью компьютера без непосредственного участия процессора.

В первую очередь он необходим в том случае, если используемые периферийные устройства имеют небольшие встроенные буферы памяти (такие устройства нуждаются в немедленном обслуживании, и не могут ждать, пока процессор освободится от выполнения задач с более высоким приоритетом). Кроме того, в некоторых случаях режим DMA обеспечивает более высокую скорость передачи данных, чем режим PIO.

К сожалению, у современных персональных компьютеров прямой доступ к памяти реализован довольно заумным образом. Проблема DMA имеет исторические корни: при разработке архитектуры компьютеров с дешевыми 8-разрядными процессорами режим прямого доступа пытались вообще исключить из системы с целью снижения ее себестоимости. При переходе на микропроцессоры с 16-разрядной, а затем 32-разрядной архитектурой возникла потребность в режиме DMA, и инженерам пришлось использовать разнообразные ухищрения, чтобы не потерять совместимости со старым программным обеспечением.

Работа с жестким диском на персональных компьютерах, совместимых с IBM AT, изначально осуществлялась через процессор, то есть только в режиме PIO, так как канал DMA шины ISA вообще функционирует с недостаточно высокой скоростью, а для работы с диском был зарезервирован 8-разрядный канал. После того, как контроллер дисков был интегрирован в системную плату и подсоединен к шине PCI, появилась возможность использования канала DMA этой шины при работе с дисками. PCI DMA обеспечивает некоторый выигрыш по скорости по сравнению с PIO не только за счет собственно прямого доступа к памяти, но и за счет большей разрядности передаваемого слова (в режиме PIO — 16 разрядов, в режиме PCI DMA — 32 разряда).

Контроллер PCI IDE может работать в одном из двух режимов — в режиме совместимости или в режиме чистой PCI. В режиме совместимости размещение регистров и распределение линий прерывания полностью совпадает с тем, которое принято для шины ISA (см. табл. 5.7).

В режиме PCI имеется возможность произвольным образом задавать адреса групп регистров каналов IDE и номера выделенных им прерываний. При этом местоположение регистров хранится в конфигурационном пространстве PCI в 32-разрядных регистрах со следующими смещениями:

- смещение 10h — адрес блока регистров команды первичного канала;
- смещение 14h — адрес блока регистров контроля первичного канала;
- смещение 18h — адрес блока регистров команды вторичного канала;
- смещение 1Ch — адрес блока регистров контроля вторичного канала.

При работе в режиме чистой PCI обращаться к регистрам контроллера IDE можно только по адресам, заданным в регистрах конфигурационного пространства.

Работать с контроллером PCI IDE можно при помощи функций PCI BIOS. Чтобы получить доступ к конфигурационному пространству контроллера PCI IDE, нужно определить его координаты на шине PCI, а сделать это можно либо по коду изделия и коду изготовителя, либо по коду класса контроллера. Первый способ пригоден только в том случае, если программное обеспечение ориентировано на какой-то чипсет определенного изготовителя. Второй способ более универсален, но требует перебора нескольких возможных вариантов значения младшего байта (байта интерфейса) кода класса (оба старших байта кода содержат значения 0lh).

Разряды байта интерфейса кода класса (см. рис. 6.4) контроллера PCI IDE имеют следующее назначение.

- Бит 0 — режим работы первичного канала (0— режим совместимости, 1 — режим чистой PCI). Содержимое данного разряда имеет значение только в том случае, если значение бита 1 равно нулю.
- Бит 1 — признак поддержки первичным каналом обоих режимов работы (0 — поддерживается только режим, определяемый битом 0, 1 — поддерживаются оба режима).
- Бит 2 — режим работы вторичного канала (0 — режим совместимости, 1 — режим чистой PCI). Содержимое данного разряда имеет значение только в том случае, если значение бита 3 равно нулю.
- Бит 3 — признак поддержки вторичным каналом обоих режимов работы (0 — поддерживается только режим, определяемый битом 2, 1 — поддерживаются оба режима).
- Биты 4-6 — не используются (должны быть установлены в 0).
- Бит 7 — признак поддержки режима Bus Master (0 — режим не поддерживается, 1 — режим поддерживается).

Для того чтобы можно было работать с дисками в режиме DMA, контроллер PCI должен иметь возможность захватывать шину, то есть должен поддерживать режим Bus Master IDE. Так как контроллеры современных материнских плат поддерживают по обоим каналам одинаковые режимы, то значение байта интерфейса может быть равно 80h, 85h или 8Ah. Соответственно, поиск контроллера нужно проводить по трем возможным кодам класса: 010180h, 010185h и 01018Ah.

С регистрами конфигурационного пространства PCI программист обычно не работает — программирование контроллера PCI IDE и моста PCI-to-ISA возлагается на операционную систему. На современных системных платах настройку временных параметров циклов PIO и DMA для установленных в системе дисков осуществляет при включении компьютера процедура начальной загрузки BIOS.

Таблица 6.4. Блок регистров ввода/вывода PCI IDE

Смещение от базового адреса	Размер	Мнемони-ческое обозначен	Назначение
00h	BYTE	BMICP	Командный регистр первичного канала IDE
0lh	BYTE	-	Зарезервирован
02h	BYTE	BMISP	Регистр состояния первичного канала IDE
03h	BYTE	-	Зарезервирован
04h	DWORD	BMIDTPP	Указатель на таблицу дескрипторов первичного канала IDE
08h	BYTE	BMICS	Командный регистр вторичного канала IDE
09h	BYTE	-	Зарезервирован
0Ah	BYTE	BMISS	Регистр состояния вторичного канала IDE
0Bh	BYTE	-	Зарезервирован
0Ch	DWORD	BMIDTPS	Указатель на таблицу дескрипторов вторичного канала IDE

Однако кроме регистров шины PCI у контроллера имеется также набор обычных регистров ввода/вывода, через которые осуществляется настройка контроллера PCI DMA на область оперативной памяти, с которой диск должен осуществлять обмен данными. Описание блока регистров ввода/вывода

PCI IDE приведено в табл. 6.4 (все регистры доступны как для записи, так и для считывания информации). Адрес блока регистров хранится в регистре базового адреса, размещенном в конфигурационном пространстве PCI IDE со смещением 20h (разряды 4 - 15 этого регистра хранят разряды 4 - 15 базового адреса блока, а разряды 0 - 3 базового адреса всегда равны 0; базовый адрес может принимать значения от 300h до FFF0h). Чтобы определить базовый адрес блока регистров, нужно прочитать значение (16 - разрядное слово) из конфигурационного регистра со смещением 20h и выделить адрес командой AND со значением FFF0h.

- Формат ***командного регистра*** канала показан на рис. 6.5. Разряды регистра имеют следующее назначение:
- бит 0 — управление работой Bus Master (пуск/останов: 1 — начать передачу в режиме DMA, 0 — остановить передачу);
- биты 1 и 2 — зарезервированы (должны быть установлены в 0);
- бит 3 — направление передачи данных (0 — чтение данных, 1 — запись данных);
- биты 4 - 7 — зарезервированы (должны быть установлены в 0).

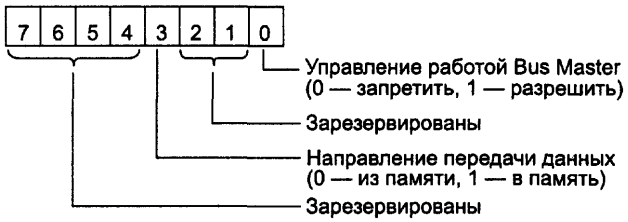

Рис. 6.5. Формат командного регистра канала PCI IDE

Формат ***регистра состояния канала*** приведен на рис 6.6. Назначение разрядов регистра:

- бит 0 — признак активности Bus Master IDE (этот разряд принимает значение 1, если установлен бит 0 командного регистра);
- бит 1 — признак ошибки передачи данных (устанавливается в 1 при возникновении ошибки; сбрасывается программно путем записи в этот разряд единицы);
- бит 2 — признак прерывания (устанавливается в 1 при поступлении прерывания от устройства IDE; сбрасывается программно путем записи в этот разряд единицы);
- биты 3 и 4 — зарезервированы (должны быть установлены в 0);
- бит 5 — индикатор способности диска 0 работать в режиме DMA (данный разряд программно устанавливается в 1 драйвером диска или BIOS при выполнении самотестирования в процессе начальной загрузки системы, если Master-диск поддерживает режим DMA и параметры канала оптимизированы на максимальную скорость передачи);
- бит 6 — индикатор способности диска 1 работать в режиме DMA (данный разряд программно устанавливается в 1 драйвером диска или BIOS при выполнении самотестирования в процессе начальной загрузки системы, если Slave-диск поддерживает режим DMA и параметры канала оптимизированы на максимальную скорость передачи);
- бит 7 — признак симплексного режима (0 — первичный и вторичный каналы независимы друг от друга и могут работать одновременно; 1 — в каждый момент времени допускается работа только одного из каналов).

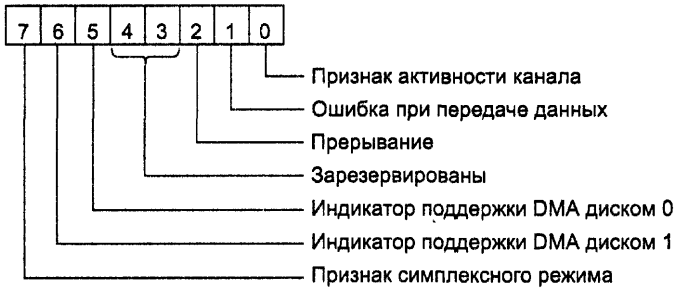

Рис. 6.6. Формат регистра состояния канала PCI IDE

Регистр указателя на таблицу дескрипторов канала должен содержать линейный (абсолютный) 32-разрядный адрес таблицы. Значение адреса выравнивается на двойное слово, то есть два младших разряда регистра обязательно содержат нули.

Чтобы контроллер мог осуществлять передачу данных, необходимо указать ему область оперативной памяти, с которой должна выполняться эта операция. Область памяти задается при помощи

таблицы дескрипторов физических областей памяти PRDT (Physical Region Descriptor Table). Таблица состоит из одного или нескольких 8-байтных дескрипторов PRD (Physical Region Descriptor), описывающих используемые при передаче данных участки памяти. Адрес таблицы должен быть выровнен на двойное слово (младшие два бита адреса всегда нулевые); размер таблицы не должен превышать 64 Кбайт, то есть таблица может содержать до 8192 элементов.

Формат дескриптора PRD приведен в табл. 6.5. Физический адрес участка памяти выравнивается на слово, то есть должен быть четным 32-разрядным числом. Счетчик байт должен быть четным 16-разрядным числом. В слове признака конца таблицы используется только старший разряд (остальные разряды зарезервированы и должны быть установлены в 0), который устанавливается в 1 у последнего элемента таблицы, а у остальных элементов равен 0. Суммарный размер всех описанных в таблице PRDT областей не должен быть меньше объема данных, передача которых задана командой, посланной диску. Более детально формат дескриптора изображен на рис. 6.7.

Таблица 6.5. Формат дескриптора PRD (элемента таблицы PRDT)

Смещение	Размер	Назначение
00h	DWORD	Физический адрес участка памяти
04h	WORD	Размер участка памяти (счетчик байтов)
06h	WORD	Признак конца таблицы

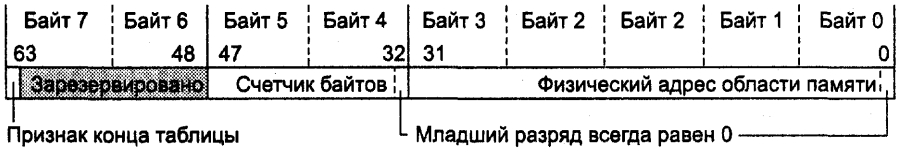

Рис. 6.7. Формат дескриптора PRD

Физические адреса участков памяти могут быть заданы произвольным образом, то есть описываемые дескрипторами участки могут быть не смежными и размещаться в произвольном порядке (рис. 6.8).

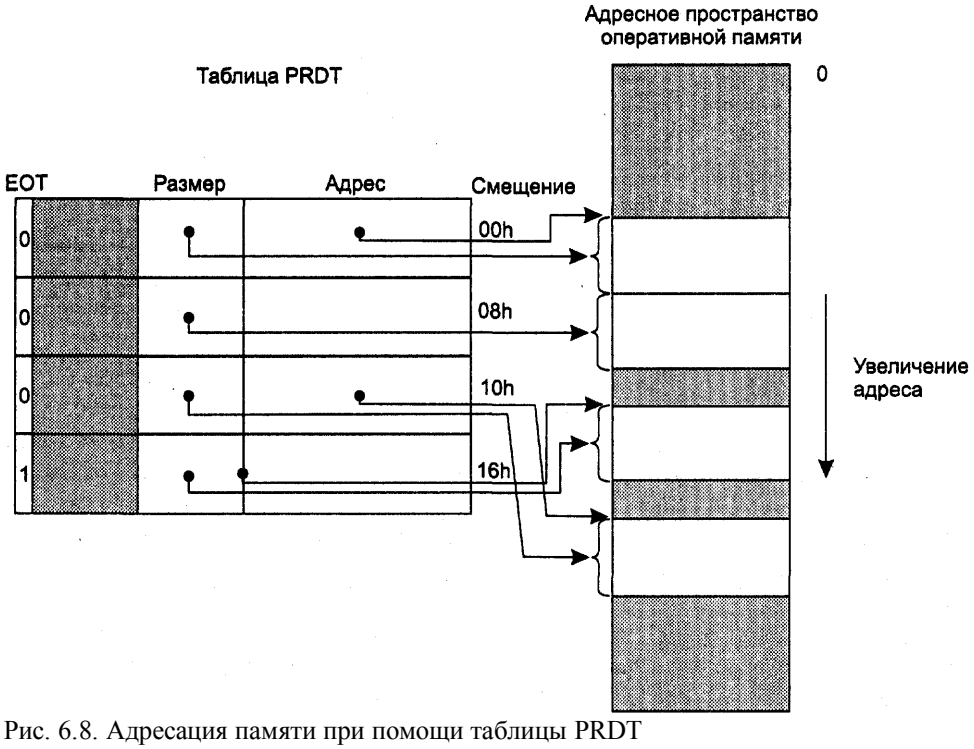

Рис. 6.8. Адресация памяти при помощи таблицы PRDT

В поле счетчика байт может находиться любое четное число от 0 до FFFEh, причем 0 соответствует значению 10000h, то есть можно задавать размер участка памяти от 2 байт до 64 Кбайт. На практике значение счетчика байтов выбирается равным размеру 16-разрядного сегмента (64 Кбайт), размеру страницы свопинга (4 Кбайт) или размеру одной из дисковых структур (сектора или кластера жесткого диска). Размер всех используемых участков, как правило, одинаковый.

При выполнении операции передачи данных между памятью и диском в режиме DMA действия рекомендуется выполнять в следующем порядке:

1. Подготовить таблицу PRD в оперативной памяти.
2. Загрузить адрес таблицы PRD в регистр указателя на таблицу дескрипторов соответствующего канала (канала, к которому подключен диск).
3. Установить значение бита направления передачи данных в командном регистре используемого канала в соответствии с выполняемой операцией.
4. Стереть бит прерывания и бит ошибки в регистре состояния используемого канала.
5. Послать диску команду, инициирующую требуемую операцию считывания или записи данных в режиме DMA.
6. Включить режим Bus Master путем установки в 1 бита управления режимом работы в командном регистре соответствующего канала.
7. Ожидать завершения передачи данных между диском и памятью (после завершения передачи диск выдает сигнал прерывания, что приводит к установке в 1 бита признака прерывания в регистре состояния).
8. Отключить режим Bus Master, сбросив в 0 бит управления режимом в командном регистре.
9. Проверить значения в регистрах состояния контроллера и диска, чтобы удостовериться в успешном завершении операции.

6.4.1. Пример программы

В листинге 6.2 приведен пример программы IDE_DMA_Test, осуществляющей считывание в режиме DMA загрузочного сектора Master-диска канала 1 (программа рассчитана на стандартный способ подключения жесткого диска к персональному компьютеру). Кроме процедур ввода/вывода общего назначения, описанных в данной главе и главе 1 «Работа с клавиатурой», программа использует также следующие подпрограммы:

- подпрограмма SearchBusMasterIDEContr осуществляет поиск контроллера PCI IDE по коду класса, а затем определяет адрес блока регистров ввода/вывода PCI DMA;
- подпрограмма HDD_Presence_Test проверяет, подключен ли диск 0 к каналу 1;
- подпрограмма ReadBootSector подготавливает канал к работе в режиме DMA, посылает диску команду считывания сектора в режиме DMA, а затем отображает сектор на экран в ASCII-коде;
- подпрограмма FatalError осуществляет вывод на экран сообщения об ошибке, ожидает нажатия любой клавиши, а затем выполняет аварийное завершение работы программы.

Листинг 6.2. Считывание загрузочного сектора Master-диска канала 1 в режиме DMA

```
IDEAL
P386
LOCALS
MODEL MEDIUM
; Подключить файл мнемонических обозначений цветов
include "lst_1_05.inc"

DATASEG
; Текстовые сообщения
Txt0 DB LIGHTCYAN,0,17
     DB "ТЕСТИРОВАНИЕ РЕЖИМА DMA MASTER-ДИСКА КАНАЛА 1",0
     DB YELLOW,5,8,"Содержимое загрузочного сектора "
     DB "Master-диска канала 1:",0
AnyK DB YELLOW,24,29,"Нажмите любую клавишу",0
; Сообщения об отсутствии компонентов системы,
; необходимых для запуска примера
NoPCI  DB LIGHTRED,12,18
       DB "Система не поддерживает интерфейс PCI BIOS",0
NoIDEC DB LIGHTRED,12,26
       DB "Контроллер PCI IDE не найден",0
BadReg DB LIGHTRED,12,28,"Неверный номер регистра",0
NoHDD  DB LIGHTRED,12,21
       DB "Master-диск на канале 1 не обнаружен.",0
NoLBA  DB LIGHTRED,12,24
       DB "Диск не поддерживает режим LBA.",0
; Сообщения об ошибках при выполнении
; операции считывания
DErr1  DB LIGHTRED,12,8,"Тайм-аут: превышена допустимая "
       DB "длительность выполнения операции.",0
DErr2  DB LIGHTRED,12,25
       DB "Неверный код режима адресации.",0
DErr3  DB LIGHTRED,12,27,"Неправильный номер канала.",0
DErr4  DB LIGHTRED,12,27,"Неправильный номер диска.",0
DErr5  DB LIGHTRED,12,26,"Неправильный номер головки.",0
DErr6  DB LIGHTRED,12,17,"Ошибка при выполнении "
```

```
       DB "команды жестким диском.",0
; Адрес блока регистров контроллера PCI IDE
IDEContrRegsBaseAddr DW ?
ENDS

; Буфер для работы с диском в режиме DMA
SEGMENT DMA_BUF para public 'DATA'
        DB 8000h DUP(?)
ENDS

; Буфер для таблицы PRD
SEGMENT PRD_Table para public 'DATA'
        DB 1024 DUP(?)
ENDS

SEGMENT sseg para stack 'STACK'
DB 400h DUP(?)
ENDS

CODESEG
;*****************************
;* Основной модуль программы *
;*****************************
PROC IDE_DMA_Test
        mov     AX,DGROUP
        mov     DS,AX
; Установить текстовый режим и очистить экран
        mov     AX,3
        int     10h
; Скрыть курсор – убрать за нижнюю границу экрана
        mov     [ScreenString],25
        mov     [ScreenColumn],0
        call    SetCursorPosition
; Вывести заголовок на экран
        mov     SI,offset Txt0
        call    ShowColorString
        call    ShowColorString
; Задать номер канала и номер диска
        mov     [ChannelNumber],1
        mov     [HDDNumber],0
; Проверить наличие и работоспособность диска
        call    HDD_Presence_Test
; Найти контроллер PCI IDE
        call    SearchBusMasterIDEContr
; Создаем таблицу PRD, состоящую из одной записи
        mov     AX,PRD_Table
        mov     ES,AX
        ; Вычислить и записать адрес области данных
        xor     EAX,EAX
        mov     AX,DMA_BUF
        shl     EAX,4
        mov     [ES:0],EAX
        ; Записать количество передаваемых байтов данных
        mov     [word ptr ES:4],512
        ; Установить признак последней строки таблицы
        mov     [word ptr ES:6],8000h
; Загрузить физический адрес таблицы PRD в
; соответствующий регистр контроллера PCI IDE
        ; Вычислить абсолютный адрес PRDT (умножить
        ; номер сегмента PRD_Table на 16)
        xor     EAX,EAX
        mov     AX,PRD_Table
        shl     EAX,4
        ; Вычислить адрес регистра контроллера
        mov     DX,[IDEContrRegsBaseAddr]
        add     DX,04h
        ; Записать адрес PRD в регистр
        out     DX,EAX
; Прочитать загрузочный сектор диска в режиме DMA
        call    ReadBootSector

; Окончание работы
@@End:  ; Ожидать нажатия любой клавиши
; Переустановить текстовый режим и очистить экран
        mov     AX,3
        int     10h
; Выход в DOS
        mov     AH,4Ch
        int     21h
ENDP IDE_DMA_Test
```

```
;*******************************************************
;*   НАЙТИ КОНТРОЛЛЕР PCI IDE И ОТОБРАЗИТЬ НА ЭКРАН    *
;* СОДЕРЖИМОЕ РЕГИСТРОВ КОНФИГУРАЦИОННОГО ПРОСТРАНСТВА *
;*******************************************************
PROC SearchBusMasterIDEContr near
        pushad
; Проверить наличие PCI BIOS
        mov     AX,0B101h
        int     1Ah
        jc      @@PCIBIOSNotFound
        cmp     EDX,20494350h
        jne     @@PCIBIOSNotFound

; Найти контроллер Bus Master IDE по коду класса
        mov     AX,0B103h
        mov     ECX,010180h  ;первый вариант кода
        mov     SI,0
        int     1Ah
        jnc     @@ReadPCIRegisters ;устройство найдено
        mov     AX,0B103h
        mov     ECX,010185h  ;второй вариант кода
        mov     SI,0
        int     1Ah
        jnc     @@ReadPCIRegisters ;устройство найдено
        mov     AX,0B103h
        mov     ECX,01018Ah  ;третий вариант кода
        mov     SI,0
        int     1Ah
        ; Устройство найдено?
        jc      @@DeviceNotFound  ;устройство не найдено

; Устройство обнаружено, его координаты на шине PCI
; находятся в регистре BX
@@ReadPCIRegisters:
; Получить базовый адрес блока регистров контроллера IDE
        mov     AX,0B10Ah ;читать двойное слово
        mov     DI,20h    ;смещение слова
        int     1Ah
        jc      @@BadRegisterNumber
        and     CX,0FFF0h ;сбросить младшие 4 разряда
        mov     [IDEContrRegsBaseAddr],CX
        popad
        ret
; Обработка ошибок
@@BadRegisterNumber:
        ; Неверный номер регистра
        mov     SI,offset BadReg
        call    FatalError
@@DeviceNotFound:
        ; Не найден контроллер IDE
        mov     SI,offset NoIDEC
        call    FatalError
@@PCIBIOSNotFound:
        ; Не поддерживается PCI BIOS
        mov     SI,offset NoPCI
        call    FatalError
ENDP SearchBusMasterIDEContr

;*******************************************************
;* ПРОВЕРИТЬ ПРИСУТСТВИЕ ДИСКА И НАЛИЧИЕ ПОДДЕРЖКИ LBA *
;*******************************************************
PROC HDD_Presence_Test near
        pusha
; Опросить Master-диск канала 1
        call    ReadHDD_ID
        ; Диск подключен?
        cmp     [HDDErrorCode],0
        jne     @@DiskNotFound
        ; Режим LBA поддерживается?
        cmp     [dword ptr SectorDataBuffer+60*2],0
        je      @@LBANotSupported
        popa
```

```
        ret
; Обработка ошибок
@@DiskNotFound:
        ; Диск не найден
        mov     SI,offset NoHDD
        call    FatalError
@@LBANotSupported:
        ; Не поддерживается LBA
        mov     SI,offset NoLBA
        call    FatalError
ENDP HDD_Presence_Test
;*****************************************
;* ПРОЧИТАТЬ ЗАГРУЗОЧНЫЙ СЕКТОР В РЕЖИМЕ DMA *
;*****************************************
PROC ReadBootSector near
        pushad
; Сбросить разряды ошибки и прерывания в регистре
; состояния канала 1
        mov     DX,[IDEContrRegsBaseAddr]
        add     DX,02h
        mov     AL,110b
        out     DX,AL
; Настроить контроллер DMA на запись данных в память
; по каналу 1
        ; Загрузить адрес управляющего регистра канала 1
        mov     DX,[IDEContrRegsBaseAddr]
        ; Сбросить управляющий регистр
        mov     AL,0
        out     DX,AL
        ; Установить бит направления передачи данных
        mov     AL,1000b
        out     DX,AL
; Послать команду чтения сектора 0 (загрузочного)
        ; Подготовить параметры команды
        mov     [ATAAddressMode],1 ;режим LBA
        mov     [dword ptr SectorAddress],0    ;сектор 0
        mov     [dword ptr ATASectorNumber],0
        mov     [ATAFeatures],0
        mov     [ATASectorCount],1 ;прочесть один сектор
        mov     EAX,[SectorAddress]
        ; Послать команду диску
        mov     [ATACommand],0C8h  ;чтение в режиме DMA
        call    SendCommandToHDD
        ; Проверить код ошибки
        cmp     [HDDErrorCode],0   ;имелась ошибка?
        jne     @@Error            ;обработка ошибки

; Активировать канал 1 контроллера DMA
        mov     DX,[IDEContrRegsBaseAddr]
        mov     AL,1001b
        out     DX,AL

; Ожидать готовность данных HDD
        mov     AX,0
        mov     ES,AX
        mov     DX,[HDDBasePortAddr]
        add     DX,7     ;адрес регистра состояния
@@WaitCompleet:
        ; Проверить время выполнения команды
        mov     EAX,[ES:046Ch]
        sub     EAX,[HDDTime]
        cmp     EAX,MaxHDDWaitTime
        ja      @@Err1   ;ошибка тайм-аута
        ; Проверить готовность
        in      AL,DX
        test    AL,88h   ;состояние сигнала BSY и DRQ
        jnz     @@WaitCompleet
; Ожидать конца цикла DMA
        mov     DX,[IDEContrRegsBaseAddr]
        add     DX,02h
@@WaitDMAOperationEnd:
        in      AL,DX
        test    AL,1
        jnz     @@WaitDMAOperationEnd

; Сбросить управляющий регистр канала 1 контроллера DMA
        mov     DX,[IDEContrRegsBaseAddr]
        mov     AL,0
```

```
        out     DX,AL

; Отобразить на экран содержимое прочитанного сектора
; в ASCII-кодах
        push    DS
        mov     AX,DMA_BUF
        mov     DS,AX
        mov     AX,0B800h
        mov     ES,AX
        mov     DI,7*160+8*2
        xor     SI,SI
        ; Задать светло-голубой цвет и синий фон
        mov     AH,LIGHTCYAN+BLUE*16
; Цикл по строкам
        mov     DX,8  ;счетчик строк
@@OutNextString:
; Цикл по символам строки
        mov     CX,64 ;счетчик символов в строке
@@OutNextChar:
        lodsb
        stosw
        loop    @@OutNextChar
        add     DI,16*2
        dec     DX
        jnz     @@OutNextString
        pop     DS
; Ожидать нажатия любой клавиши и выйти из процедуры
        mov     SI,offset AnyK
        call    ShowColorString
        call    GetChar
        popad
        ret
; Обработка ошибок
@@Error:
        cmp     [HDDErrorCode],1
        je      @@Err1
        cmp     [HDDErrorCode],2
        je      @@Err2
        cmp     [HDDErrorCode],3
        je      @@Err3
        cmp     [HDDErrorCode],4
        je      @@Err4
        cmp     [HDDErrorCode],5
        je      @@Err5
        cmp     [HDDErrorCode],6
        je      @@Err6
@@Err1: mov     SI,offset DErr1
        call    FatalError
@@Err2: mov     SI,offset DErr2
        call    FatalError
@@Err3: mov     SI,offset DErr3
        call    FatalError
@@Err4: mov     SI,offset DErr4
        call    FatalError
@@Err5: mov     SI,offset DErr5
        call    FatalError
@@Err6: mov     SI,offset DErr6
        call    FatalError
ENDP ReadBootSector
;*****************************************************
;*                  ФАТАЛЬНАЯ ОШИБКА                 *
;*       ВЫДАТЬ СООБЩЕНИЕ И ВЫЙТИ ИЗ ПРОГРАММЫ       *
;* Параметры:                                        *
;* DS:SI – указатель на "цветную" строку сообщения   *
;          об ошибке (в формате для ShowColorString).*
;*****************************************************
PROC FatalError near
; Вывести сообщение об ошибке
        call    ClearScreen
        call    ShowColorString
        mov     SI,offset AnyK
        call    ShowColorString
        call    GetChar
; Аварийный выход
        mov     AH,4Ch
        int     21h
ENDP FatalError
ENDS
```

```
; Подключить процедуры вывода данных на экран
include "lst_1_02.inc"
; Подключить процедуру непосредственного считывания
; сектора с жесткого диска
include "lst_6_07.inc"

END
```

ПРИМЕЧАНИЕ. Для успешного запуска программы необходимо, чтобы жесткий диск, способный работать в режиме UDMA, был подключен к каналу 1 в качестве Master-диска. Кроме того, контроллер PCI системной платы должен поддерживать режим Bus Master, а система BIOS должна обеспечивать начальную установку временных параметров цикла DMA в процессе начальной загрузки системы.

Контрольные вопросы.

1. Кто распространяет и как достать документацию по шине PCI?
2. Где можно найти информацию по PCI BIOS?
3. Что обеспечивает интерфейс PCI BIOS и каково основное назначение его функций?
4. Дайте общую характеристику конфигурационного пространства устройства PCI.
5. Какие поля конфигурационного пространства составляют практический интерфейс и где они используются?
6. Как осуществляется доступ к функциям PCI BIOS?
7. Сколько функций используется прерыванием 1Ah для обслуживания PCI и каково их назначение?
8. Охарактеризуйте функцию B101h. Какую информацию она возвращает?
9. Охарактеризуйте функции B102h, B103h и B106h.
10. Охарактеризуйте функции B108h, B109h и B10Ah.
11. Охарактеризуйте функции B10Bh, B10Ch и B10Dh.
12. Какие значения могут принимать коды возврата?
13. Перечислите коды базовых классов устройств PCI.
14. Какова общая структура поля кода класса устройства?
15. Ознакомьтесь с таблицей 6.3 полного описания кодов классов устройств PCI.
16. Как задается байт интерфейса кода класса IDE - контроллера?
17. Охарактеризуйте листинг 6.1 программы поиска PCI-устройста.
18. В каких режимах может работать контроллер PCI IDE?
19. Где хранится информация о местонахождении регистров контроллеров IDE в режиме PCI?
20. Как можно получить доступ к конфигурационному пространству контроллера PCI IDE?
21. Когда контроллер PCI IDE может работать в режиме DMA?
22. Какие регистры входят в состав блока регистров ввода/вывода контроллера PCI IDE и как определить их адреса?
23. Охарактеризуйте форматы командного регистра и регистра состояния канала.
24. Что представляет собой таблица дескрипторов физических областей памяти PRDT и каков формат дескриптора PRD?
25. Как могут располагаться адреса участков памяти, используемые для обмена в режиме PCI DMA?
26. В каком порядке рекомендуется выполнять операции передачи данных между памятью и диском в режиме DMA?
27. Охарактеризуйте программу, приведенную в листинге 6.2.

В данном разделе использованы материалы из [4].

7. Устройства массовой памяти на сменных носителях

К устройствам массовой памяти на сменных носителях можно отнести устройства, имеющие емкость, значительно превышающую емкость обычных дискет. Они предназначены для систем архивации данных и переноса больших объемов информации. В зависимости от назначения такие устройства выполняются внутренними и внешними, стационарными и портативными. Большинство этих устройств имеют интерфейс SCSI или ATA, портативные устройства часто имеют интерфейс подключения к параллельному порту, что облегчает подключение к любому компьютеру, но приводит к проигрышу в скорости передачи информации и повышенной загрузке процессора при обмене данными. Более современным и удобным является интерфейс USB, особенно в версии 2.0, который имеет пропускную способность не хуже ATA и SCSI. Устройства на сменных носителях, как дисковые, так и ленточные, могут и не иметь поддержки на уровне ROM BIOS, при этом доступ к ним становится возможным только после загрузки специальных драйверов, соответствующих используемой операционной системе.

Из всех рассматриваемых типов сменных носителей (не считая обычных дискет, которые массовой памятью не назовешь) наибольшей популярностью пользуются CD и магнитооптические диски. Для работы с этими носителями кроме одиночных накопителей выпускаются и автоматы, сочетающие в себе хранилище на несколько десятков (а то и сотен) дисков с одним или несколькими накопителями. Под программным управлением специальный механизм может извлечь из набора требуемый диск и установить его на свободный накопитель и, конечно, выполнить обратную операцию. Эти устройства могут быть как настольного, так и напольного исполнения. Используемый интерфейс — SCSI. Эффективное использование таких автоматов возможно лишь при наличии сложного специального программного обеспечения, которое, естественно, должно быть рассчитано на применяемую операционную систему.

7.1. Магнитные диски

Накопители на магнитных дисках могут иметь различные уровни сменяемости. Обычный накопитель на жестком диске (винчестер) устанавливается на компьютер надолго, и для его подключения требуется открыть корпус системного блока и отключить питание. Существуют и накопители, допускающие «горячую замену» (hot swap), для которой не требуется отключение питания и имеется специальный конструктив, позволяющий устанавливать и снимать винчестер со стороны лицевой панели, не разбирая компьютер. Однако такая замена производится в основном в сервисных целях. *Съемные винчестеры* — устройства с аналогичными свойствами, но предназначенные для переноса или обеспечения безопасного (в смысле конфиденциальности) хранения данных. Выпускаются недорогие переходники Mobile Rack, позволяющие использовать в качестве съемного обычный ATA-диск. При этом 3,5" накопитель устанавливается во внешний 5" отсек. Однако следует помнить, что обычные накопители не рассчитаны на вибрацию и удары, опасность которых при частой переноске устройств повышается. Винчестер же, тщательно закутанный в мягкую упаковку, занимает много места. Поэтому больший интерес представляют накопители именно со съемными носителями.

Диски Бернулли (Bernoulli removable media drive). Привод размером с 5" дисковод использует 3,5" гибкие диски в жесткой кассете. Объем 35-150 Мбайт. При вращении диска со скоростью 3600 об/мин возникает эффект Бернулли, подтягивающий диск к плоскости с головками. По скоростным параметрам близки к винчестеру. Кассета устойчива к внешним воздействиям. Интерфейс: внутренние — IDE, SCSI; внешние — SCSI, LPT-порт. Производитель: Iomega. Встречаются редко.

Устройства с *кассетными жесткими дисками* (removable media drives), выпускаемые фирмой SyQuest, используют картриджи, в которых размещены и диски (1-2 пластины), и головки. Они имеют все атрибуты современных винчестеров — встроенную сервоинформацию, зонную запись, стандартный размер сектора 512 байт, небольшое время поиска. Устройства серии EZ используют картриджи PDC (Power Disk Cartridge), выпускаемые фирмами SyQuest, Nomai, Kao Infosystems, Maxell, Polaroid и Xyratex с рядом емкости 135, 230, 270 и 540 Мбайт. В картридже расположена одна пластина формата 3,5". Устройство *EZFlyer* на 230 Мбайт выпускается во внешнем (LPT) и внутреннем (EIDE, в отсек 3,5") исполнениях. Раньше выпускались и внешние с интерфейсом SCSI. Время доступа 13,5 мс, скорость вращения 3600 об/мин, скорость обмена до 2,4 Мбайт/с (IDE). Совместимо с картриджами на 135 Мбайт, используемыми в ранее выпускавшихся устройствах *EZ135* с похожими свойствами.

Устройство *SparQ* использует специальный картридж 1 Гбайт (тоже с одной пластиной 3,5"). Выпускается во внешнем (LPT) и внутреннем (EIDE, в отсек 3,5") исполнении. В устройстве *SyJet* используется картридж емкостью 1,5 Гбайт с двумя пластинами 3,5". Во внешнем исполнении имеет интерфейсы LPT или SCSI, во внутреннем — EIDE, устанавливается в отсек 3,5". Устройства SparQ и SyJet никак не совместимы.

По параметрам производительности кассетные жесткие диски большой емкости вполне сопоставимы с винчестерами: среднее время доступа 12 мс, время перехода на соседний трек 1,5 мс, скорость вращения 5400 об/мин, длительная скорость обмена 3,7-6,9 Мбайт/с (для IDE и SCSI). Однако

высокая цена картриджей ограничивает сферу их применения — они эффективны, когда выдвигаются жесткие требования к производительности (особенно при записи).

Накопители на гибких магнитных дисках в классическом исполнении остановились на емкости носителя в 2,88 Мбайт, но, по крайней мере на нашем рынке, ни устройств, ни дискет не наблюдается, и реальный «потолок» — 1,44 Мбайт. Для повышения емкости требуется радикальное увеличение плотности записи, в первую очередь за счет увеличения числа треков. Однако при этом требуется точное позиционирование, что на обычном шаговом двигателе (разомкнутая система) недостижимо. Решением проблемы стало совмещение технологий: магнитная запись/считывание с оптическим позиционированием, или так называемые *гибкие магнитооптические диски* (floptical drives) — 3,5" НГМД сверхвысокой плотности. Первые модели имели емкость около 20 Мбайт: 755 дорожек, 27 секторов по 512 байт. Скорость вращения 720 об/мин. Интерфейс — SCSI, ATA или специальный адаптер, позволяющий использовать их как дисковод A:. Накопитель совместим и с обычными дискетами 720 Кбайт и 1,44 Мбайт (2,88 Мбайт только по чтению). По удельной стоимости хранения информации при относительно небольшой емкости дискеты эти устройства не вызывали особого интереса.

Современные устройства *LS-120* (Laser Servo 120 Мб) имеют емкость дискеты 120 Мб — по 1736 информационных треков на каждой стороне с зонным форматом записи. Устройство имеет интерфейс ATAPI, логическая геометрия — 960 цилиндров × 8 головок × 32 сектора на трек. Физически на каждом треке размещается 51-92 сектора размером 512 байт. Лазерное позиционирование использует 900 сервотреков. Накопитель существенно дешевле конкурирующих с ним магнитооптических устройств, о которых речь пойдет ниже, но удельная стоимость носителя гораздо выше. Скорость обмена достигает 90-100 Кбайт/с в среде DOS и 200-300 Кбайт/с в среде Windows. Современные версии BIOS имеют поддержку этих устройств и позволяют в последовательность загрузочных устройств включить и LS-120. Устройство позволяет отказаться от использования стандартного дисковода (совместимо с дискетами 1,44 Мбайт по чтению и записи). Конечно, специальное ПО (например, программы тиражирования дисков), работающее с регистрами контроллера НГМД, дискету в LS-120 не увидит. Как и обычные дискеты, носители для LS-120 чувствительны к сильным магнитным полям.

Весьма популярные накопители *Zip* фирмы Iomega имеют емкость носителя 100 и 250 Мбайт, причем устройства на 250 Мбайт совместимы с носителями на 100 Мбайт (но не наоборот). Носитель— гибкий диск, помещенный в пластмассовый картридж 3" формата. Накопитель несколько дороже LS-120, зато носители дешевле; правда, совместимости с обычными дискетами нет. Интерфейс — IDE, SCSI, USB или LPT. Для устройств SCSI и USB предлагаются переходники на PC Card, есть и варианты подключения к Fire Wire (скорость до 2 Мбайт/с). Среднее время доступа — 29 мс. Для устройств на 100 Мбайт характерна скорость обмена до 1,4 Мбайт/с, для 250 Мбайт — до 2,4 Мбайт/с. Эта скорость достигается лишь на быстрых интерфейсах (SCSI и ATA), для LPT и USB скорость ограничивается на уровне 0,8-1 Мбайт/с. Реальная средняя скорость в среде DOS порядка 95 Кбайт/с и 750-1000 Кбайт/с в среде Windows. Время короткого форматирования носителя (без верификации) — 10 с, длинного (с верификацией) — 10 мин. Накопители выдерживают 10 000 установок носителей, картридж — до 2000. Накопители Zip широко используются как устройства архивации данных, а также для переноса данных. Благодаря невысокой цене они популярны как у корпоративных, так и у «домашних» пользователей.

Скорость вращения диска — 2941 об/мин. В исходном положении головки выведены из картриджа, слегка разведены и лежат на очищающей площадке; В рабочем состоянии головки вводятся в картридж с торца и зависают над поверхностью вращающегося диска, не касаясь ее. Носители чувствительны к термодеформации — диск, записанный в горячем состоянии, будет плохо читаться в холодном. Сами устройства Zip особенно не греются, но при внутреннем исполнении могут нагреваться от других устройств компьютера (например, от винчестера).

С накопителями Zip связана история о так называемых «щелчках смерти» — щелканье дисководов при чтении некоторых «зараженных» дискет. Щелчки возникают, когда накопитель обнаруживает неисправимую ошибку чтения — при этом он выводит головки из картриджа на парковочную площадку, очищая их, и вводит повторно. Само по себе это щелканье безобидно, но является признаком плохой записи (повреждения сервометок, низкоуровневого формата, служебных областей). Носитель, вызывающий щелканье одного накопителя, скорее всего, заставит щелкать и другие — отсюда миф о «заразности» щелчков. Крайне редко, но все-таки встречаются и действительно «смертоносные» носители, у которых поврежден (порван, помят) наружный край диска. Если такой носитель вставить в дисковод, при вводе головок он может их буквально срубить. После этого дисковод может долго пытаться «искать» данные на носителе (но уже без головок!), щелкая при повторных введениях-выведениях остатков головок. Перенос такого носителя на другой дисковод наверняка погубит и его, поскольку 3000 об/мин — скорость нешуточная.

ВНИМАНИЕ ———————————————————————————————————

Если дисковод при считывании начинает щелкать, следует внимательно осмотреть край носителя, отведя защитную шторку и проворачивая диск вручную на полный оборот. Диск с малейшими обнаруженными дефектами (зазубринами, замятиями и т. п;) должен быть сразу изъят из обращения.

Накопители *Jaz* той же фирмы Iomega являются развитием Zip, но уже на твердом носителе: емкость 1 Гбайт у первых моделей в дальнейшем увеличена до 2 Гбайт. Интерфейс — IDE, SCSI, возможны переходники на USB или LPT. Пиковая скорость достигает 20 Мбайт/с, средняя — 7,5 Мбайт/с, время доступа около 16 мс. Накопители Jaz распространены не так широко, как Zip (сказывается цена, особенно на носители), они хорошо подходят для архивации больших объемов данных.

Фирма Iomega выпускает и карманный вариант — *PocketZip Mobile Drive* с емкостью диска 40 Мбайт. Кроме внешнего интерфейса, он снабжен устройством считывания флэш-карт CompactFlash+ и SmartMedia емкостью до 16 Мбайт, применяемых во многих цифровых фотокамерах. Такие же диски используются и в цифровых аудиоплейерах *HipZip* той же фирмы. Устройства ZipCD к обычным картриджам Zip/Jaz отношения не имеют — это просто устройства CD-RW (см. п. 7.3.2).

7.2. Магнитооптические диски

В устройствах с магнитооптическими дисками, *МОД* (MOD — Magneto-Optical Drives), в процессе записи и чтения магнитного носителя используется лазер. МОД сочетают практически неограниченное число перезаписей, свойственное магнитным носителям, с чрезвычайно надежным хранением записанной информации. Устройство привода МОД укладывается в традиционную схему — имеется шпиндель, который вращает диск, и головка, перемещаемая над поверхностью диска системой позиционирования. Расстояние от головки до поверхности диска около 1 мм, что исключает риск касания поверхности диска (отсюда очень высокая надежность). На головке расположен лазер и оптическая система, фокусирующая луч на магнитном слое носителя. На диске поверх магнитного слоя имеется слой субстрата, защищающего магнитный от механических воздействий. Луч на поверхности диска (субстрата) дает пятно размером около 1 мм, так что пылинки и микро царапины не оказывают существенного влияния на происходящие процессы. На магнитном слое, находящемся под прозрачным слоем субстрата (толщина субстрата около 1,2 мм), за счет фокусировки пятно уменьшается уже до микронных размеров, что и определяет достижимые (в плане уменьшения) размеры хранящих «ячеек». Расстояние между треками у дисков 128 Мбайт — 1,6 мкм, у 540/650 Мбайт — 1,1 мкм. Один бит на треке у дисков 128 Мбайт занимает 0,52 мкм, у 540/650 Мбайт — 0,31 мкм. Принципы записи и чтения МОД весьма своеобразны.

Запись осуществляется термомагнитным способом: магнитное поле головки способно перемагнитить только микроскопическую зону носителя, разогреваемую лазерным лучом до температуры выше точки Кюри (порядка 200°C). Зона, вышедшая из-под луча, «замораживает» полученное состояние намагниченности. Традиционно в магнитооптике используют двухпроходную запись. Для того чтобы записать информацию в секторе, после позиционирования головки за первый оборот сектор стирают. Для этого головка создает постоянное магнитное поле, а лазер включается на полную мощность, когда под ним проходит требуемый сектор (секторы). В результате все «засвеченные» области данных переводятся в состояние с одним и тем же направлением намагниченности. На следующем обороте выполняется собственно запись — направление магнитного поля головки меняется на противоположное и формируются мощные импульсы лазера над теми точками, состояние которых нужно изменить, чтобы закодировать требуемую информацию. В результате именно эти ячейки будут перемагничены. Для большей достоверности на третьем обороте выполняется верификация — считывание записанной информации. В принципе возможна и однопроходная запись, если модулировать магнитное поле (переключать его направление) над каждой битовой областью. Однако в данной технологии (на большом расстоянии головки от диска, которое уменьшать не хотят принципиально) из-за явления индукции магнитная модуляция при больших скоростях записи требует непомерных затрат энергии. Модуляция магнитного поля *MFM* (Magnetic Field Modulation) применяется при записи мини-дисков, используемых в аудиотехнике (там не требуется высокая скорость записи). Однопроходная запись для компьютерного применения основана на технологии LIMDOW, о которой будет сказано ниже.

Считывание информации с магнитного слоя носителя выполняется тоже с помощью лазера (при малой мощности излучения) и основано на эффекте Керра — изменения поляризации света под действием магнитного поля. Отраженный луч проходит через поляризационную оптику, в результате на фотоприемник приходит луч, интенсивность которого модулирована (по амплитуде) в соответствии с записью на магнитном слое. Разрешающая способность оптики и фотоприемника определяет достижимую плотность хранения информации — записать можно и более мелкие ячейки, но считать их не удастся. Повысить разрешение при считывании позволяет технология MSR (см. ниже).

Магнитооптические диски организованы так же, как и магнитные — у них имеются дорожки, разбитые на секторы, только нумерация дорожек начинается от центра диска. Размер сектора может быть стандартным (512 байт данных) или увеличенным (2048 байт данных). Больший размер сектора снижает долю накладных расходов на служебную информацию. Количество секторов на треке переменно — здесь тоже применяется зонная запись. Кроме пользовательской области, на диске в каждой зоне имеются

служебные и резервные области, позволяющие переназначать дефектные секторы прозрачно для пользователя. Встроенный контроллер накопителя МОД выполняет трансляцию физической геометрии (с зонным форматом) в логическую и имеет внутренние средства для переназначения дефектных блоков.

Магнитооптические диски бывают двух размеров — 5,25" (двусторонние) емкостью 650 Мбайт, 1,3, 2,6, 4,6 Гбайт и 3,5" (односторонние) емкостью 128, 230, 540, 640 Мбайт и 1,3 Гбайт.

Магнитооптические устройства ввиду стандартизации носителя являются распространенным средством не только архивации, но и обмена данными. Носители стандартизованы ISO/IEC (для 3,5" дисков стандарт 15041:1997), что обеспечивает совместимость дисков и устройств разных производителей.

Существует ряд градаций объемов МОД, обозначаемых через кратность объему начальной модели диска (табл. 7.1). Диски различаются плотностью размещения треков; методами модуляции — PWM (Pulse Width Modulation — широтно-импульсная модуляция) для 1x и 2x, PPM (Pulse Position Modulation — позиционно-импульсная модуляция) для 4x; методами кодирования — RLL 2.7 для 1x и 2x, RLL 1.7 для 4x и 5x; а также организацией зонного формата записи. Устройства и диски имеют обратную совместимость: устройства большей емкости могут работать и с дисками меньшей емкости. Устройства 4x (5x) обычно могут полноценно работать с дисками 2x и читать диски 1x.

Таблица 7.1. Емкость магнитооптических дисков

Плотность	Диск 130 мм (5")	Диск 90 мм (3,5")
1x	560/650 Мбайт	128 Мбайт
2x	1,2/1,3 Гбайт	230 Мбайт
4x	2,3/2,6 Гбайт	—
5x	—	540/640 Мбайт
10x	4,6Гбайт	1,3 Гбайт

Для одних и тех же устройств могут указываться два значения емкости, например 540/ 640 Мбайт. Меньшее значение соответствует форматированию со стандартным (512 байт) размером сектора, большее - с увеличенным (в нашем случае до 2048 байт).

Приводы МОД выпускаются во внешнем и внутреннем исполнениях. Поскольку при записи происходит значительное тепловыделение, в ряде случаев предпочтительно внешнее исполнение устройств. Устройства имеют интерфейс SCSI, IDE или Fiber Channel, обеспечивающий скорость передачи, достойную МОД. Для подключения к LPT и USB обычно применяются переходные адаптеры SCSI или IDE.

Благодаря малому размеру диска наиболее популярны устройства 3,5" формата, для которых уже достигнута емкость 1,3 Гбайт. Магнитооптические накопители и диски производятся множеством фирм, лидером в производстве устройств 3,5" является Fujitsu. Современные устройства имеют скорость вращения 3000-4500 об/мин, что обеспечивает малую задержку доступа (7 мс) и среднее время поиска 20-30 мс. Скорость передачи данных даже у одного устройства зависит от емкости диска — диски большей емкости работают быстрее, поскольку у них больше секторов на треке. Скорость передачи на начальных цилиндрах, которыми в МОД являются внутренние, заметно ниже, чем на внешних — следствие использования зонной записи (ZBR). Для магнитооптики с традиционной технологией характерно значительное различие скоростей записи и считывания — запись происходит существенно медленнее из-за необходимости двух-трех проходов над одним и тем же треком. По скоростным характеристикам считывания МОД находятся между накопителями на гибких (типа ZIP) и жестких (Jaz) магнитных дисках. Время доступа (позиционирования) упирается в массу головки (инерционность).

Более прогрессивная технология записи *LIMDOW* (Light Intensity Modulation Direct OverWrite — непосредственная перезапись с модуляцией интенсивности луча) позволяет исключить проход стирания, чем повышается скорость записи в 2 раза (в 1,5 раза, если выполняется и верификация). Диски для такой технологии записи имеют более сложную структуру — они имеют несколько магнитных слоев с различающимися свойствами (с разной температурой Кюри), и в них используется магнитное взаимодействие элементов соседних слоев. Верхний слой (на который падает луч) является информационным, состояние намагниченности нижнего слоя задается при инициализации диска (на этапе изготовления) и остается неизменным. Головка при записи также создает постоянное магнитное поле, но мощность лазера модулируется. Поле от головки старается намагнитить информационный слой в одном направлении («0»), а поле нижнего слоя — в противоположном. Для записи «0» вырабатывается сильный импульс лазера, для «1» — более слабый. В процессе остывания результатом взаимодействия слоев будет запомненное направление намагниченности информационного слоя, отвечающее мощности импульса. Для того чтобы эти процессы были не слишком критичны к точности температуры нагрева, диски имеют довольно сложную многослойную структуру, что, естественно, заметно сказывается на их стоимости. Ускоренную запись позволяют выполнять только накопители, поддерживающие эту технологию, и только для специальных дисков, которые имеют пометку «OW». Диски OW, по сравнению с

обычными, обладают еще и большим гарантированным сроком хранения. Технология LIMDOW появилась только с дисками 540/640 Мбайт (реализована на приводах DynaMO, GigaMO фирмы Fujitsu).

Форматирование МОД выполняется специальной утилитой, входящей в комплект драйверов устройства (MO Formatter). Низкоуровневое форматирование МОД (LLF) выполняется в процессе производства, в дальнейшем пользователь может выполнить его с помощью утилиты. Поводом для LLF может быть потеря производительности диска (большое число ошибок). При низкоуровневом форматировании задается и размер сектора. В зависимости от объема диска и производительности привода LLF может занимать от нескольких минут до получаса. Форматирование верхнего уровня выполняется под конкретную файловую систему. В целях экономии времени имеет смысл приобретать предварительно форматированные диски (preformatted MO Disk), для которых выполнены обе стадии форматирования.

Форматирование магнитооптических дисков на верхнем уровне может выполняться в *стиле дискет* (super floppy) или в *стиле винчестера* (PC/AT HDD). В первом случае диск представляется в виде очень большой дискеты, в его нулевом логическом блоке содержится загрузчик и дескриптор носителя (без таблицы разделов). Дискета, безусловно, подразумевает сменяемость, но в этом режиме МОД может быть использован только с помощью драйвера, загружаемого уже операционной системой. Этот формат поддерживается и Windows, и OS/2. При форматировании в стиле винчестера диск начинается с таблицы разделов, и для системы выглядит как жесткий диск, который, в случае использования интерфейса SCSI, может обслуживаться и BIOS хост-адаптера без всяких загружаемых драйверов. Это позволяет даже загружать ОС с МОД, но потом не всякая BIOS «догадается», что у этого «жесткого диска» возможна смена носителя. Загрузка с МОД ATAPI возможна далеко не во всех версиях BIOS. Некоторые версии Windows 9x не позволяют использовать разделы сменного диска, кроме первого, — для них приходится выбирать формат super floppy. Некорректная отработка смены носителя может привести к потере данных, когда после смены диска ОС не обновит в ОЗУ дескриптор диска и таблицы размещения файлов. Заметим, что смена носителей для системы Macintosh отрабатывается иначе, чем в PC, поэтому накопители МОД с интерфейсом SCSI имеют переключатель «Mac-PC», который должен быть корректно установлен.

ВНИМАНИЕ ———————————————————————————————————

При использовании размера сектора 2048 байт при загрузке с МОД могут возникнуть проблемы — в отличие от специального загружаемого драйвера, BIOS может «не догадаться» учесть нестандартный размер сектора. С таким размером сектора не работает DriveSpace, а также FDISK.

Магнитооптические диски устойчивы к сильным внешним магнитным полям, поскольку при нормальной температуре коэрцитивная сила носителя велика; они не боятся перепадов температуры, солнечного света, вибрации. Гарантированное время жизни диска около 40 лет.

У магнитооптических дисков, в принципе, есть еще возможности повышения плотности хранения информации. Технология *магнитного суперразрешения MSR* (Magnetic Super Resolution) позволяет увеличить плотность хранения данных, по сравнению с чисто оптическими при том же размере пятна луча. При записи температура нагрева распределена по пятну неравномерно — в центре она значительно выше. Поэтому пороговая температура достигается не на всей площади пятна, а только в меньшей центральной области, и только в этой области происходит перемагничивание. Поскольку поверхность носителя движется относительно головки, нагретая область представляет собой эллипс, вытянутый вдоль дорожки. Процесс считывания отличается от классического — его технология напоминает LIMDOW. Диск MSR имеет несколько магнитных слоев (в упрощенной схеме можно рассматривать три). Для оптического считывания виден только верхний слой (reading out layer), а информация записана па нижний (recording layer). Между ними расположен промежуточный уровень (intermediate layer), который при определенной температуре способен транслировать состояние нижнего слоя в верхний, видимый. Считывание производится довольно мощным лучом, воздействующим на магнитные свойства верхних слоев, но не «задевая» (по температуре) нижнего хранящего, в присутствии постоянного магнитного поля. Сверхразрешение достигается благодаря эффекту двойного маскирования: в центре пятна луча, где температура высокая, промежуточный слой теряет магнитные свойства и не передает состояние слоя хранения на выходной слой, который намагничивается внешним полем. На краю пятна работает низкотемпературная маска — здесь промежуточный слой намагничивается внешним полем перпендикулярно слою хранения, и слой считывания опять-таки оказывается намагниченным внешним полем головки. И только в узкой области умеренной температуры состояние слоя хранения транслируется на видимый слой, благодаря чему сигнал принимается от области, гораздо меньшей, чем пятно луча лазера, с хорошим отношением сигнал/шум.

Дальнейшим развитием технологии MSR являются MAMMOS (Magnetic AMplified Magneto Optical System) и DWDD (Domain Wall Displacement Detection), находящиеся еще в стадии лабораторных исследований.

7.3. Оптические диски — CD, DVD, PD

В оптических дисках для хранения информации используется изменение оптических свойств (в основном, степени отражения) поверхности носителя. В процессе считывания при освещении трека лазерным лучом возникает модуляция интенсивности отраженного луча, воспринимаемого фотоприемником. В модулированном луче закодирована двоичная информация, размещенная на треке. На этом принципе основаны диски CD, их «потомки» DVD и особняком стоящие диски PD.

В компьютер оптический диск пришел из техники цифровой аудиозаписи. *Аудио компакт-диски,* называемые Audio-CD (Compact Disk), были разработаны фирмами Sony и Philips в 1982 году. Диаметр компакт-диска 120 мм, толщина 1,2 мм. Как и грампластинки, диски имеют одну спиральную дорожку, но начинающуюся с внутренней стороны диска. Эта спираль имеет 22 188 витков (поперечная плотность около 600 витков на 1 мм) и длину более 5 километров. Область с диаметром 46-50 мм является *вводной* (lead-in), область 116-117 мм — *выводная* (lead-out). Область между этими зонами называется *программной* (program area). Дорожка представляет собой цепочку «ямок» (pits) в прозрачной основе диска, за которой расположен светоотражающий слой. Поперечный шаг витков спирали — 1,6 мкм, ширина дорожки (ямок) — 0,5 мкм, глубина ямок — 0,125 мкм. Края ямок соответствуют двоичным «1» канальной информации, кодирующей полезную информацию, записанную на диске. Участок без изменения глубины (как ямка, так и «равнина») соответствует двоичным «0», число нулевых битов определяется длиной этого участка. Длина ямок лежит в пределах 0,83-3,56 мкм. Для считывания применяют инфракрасный лазер с длиной волны 780 нм (в воздухе). Глубина ямок выбрана равной 1/4 длины волны луча лазера в прозрачном материале основы диска. Благодаря этому луч, отраженный от дна ямки, возвращается в приемник в противофазе с лучом, отраженным от поверхности (он в обе стороны проходит лишние 1/2 волны), — они взаимно уничтожаются, что повышает контрастность восприятия ямок. Для выравнивания продольной плотности записи диск вращается с переменной угловой скоростью, а привод обеспечивает постоянство линейной скорости носителя, проходящего под головкой. Этим обусловлено большое время доступа, поскольку время уходит и на разгон (торможение) диска при значительно быстром перемещении головки. Скорость считывания аудиоданных, требуемая для звуковоспроизведения в реальном масштабе времени, соответствует информационной скорости 150 Кб/с. Диск способен хранить информацию 74 минут звучания стереосигнала с частотой дискретизации 44,1 кГц и 16-разрядными выборками (Ходит легенда о том, что время звучания было определено исходя из возможности помещения на один диск записи 9-й симфонии Бетховена, длящейся именно столько времени). На диске используется только одна поверхность. В таком виде появились и первые компакт-диски для хранения данных, для считывания которых применяются приводы CD-ROM (Compact Disk Read Only Memory). Информационная емкость стандартного диска CD-ROM — 650 Мбайт. Впоследствии были разработаны диски, записываемые пользователем, CD-R (CD Recordable), и устройства записи (рекордеры) CD-Writer, CD-Recorder (естественно, способные и считывать информацию). Эти диски и устройства позволяли лишь *однократную запись,* их еще называли *CD-WORM* (Write Once, Read Many — одноразовая запись, многократное чтение) или *CD-WO* (Write Once). Устройства с возможностью *многократной записи* на оптический диск первоначально назвали *CD-E* (Erasable — стираемые). Однако с маркетинговой точки зрения такое название показалось непривлекательным (от того, что данные могут быть стерты, восторг мало кто испытывает), и его заменили на *CD -R*W (Rewritable — перезаписываемые), хотя физическая суть та же. Особую привлекательность CD-R и CD-RW придает совместимость этих дисков с обычными приводами CD-ROM и даже аудиоплейерами. Существуют также и оптические диски PD (Phase change Disk), допускающие многократную перезапись, но никак не совместимые с приводами CD-ROM. Правда, имеются комбайны CD/PD, работающие с разными типами дисков. Более новые и емкие диски DVD (Digital Video Disk — цифровой видеодиск, или Digital Versatile Disk — универсальный цифровой диск) совместимости с CD не имеют. Но и здесь появились комбайны, позволяющие работать и с DVD, и с CD.

В начале 90-х годов в Японии были распространены миниатюрные диски *CD Single* диаметром 80 мм, емкостью около 200 Мбайт. Для этих дисков на лотках приводов CD имеется небольшое углубление, позволяющее их использовать наравне с «большими». На маленьких дисках выпускались продукты CD-XA (смесь видео, аудио и данных, см. ниже). Их не следует путать с мини-дисками (Sony Mini Disk), являющимися перезаписываемыми магнитооптическими дисками.

7.3.1. Носители информации CD, CD-R, CD-RW

Оптические диски CD, CD-R и CD-RW имеют прозрачную поликарбонатную (пластиковую) основу, над которой расположен слой, хранящий информацию, защищенный сверху лаком. На верхней поверхности этого «пирога» может быть нанесена этикетка. Хранящий слой расположен ближе всего к верхней стороне; механические повреждения (царапины, вмятины) с верхней стороны чаще приводят к неисправимым ошибкам чтения. Царапины, как и пылинки, с нижней стороны, через которую светит лазер, не так страшны — через них проходит луч с еще довольно большим диаметром пятна (порядка 1 мм). Луч фокусируется в точку микронных размеров уже на самом хранящем слое, так что мелкие дефекты на внешней поверхности не оказывают существенного влияния на оптические процессы.

Устройство хранящего слоя может быть различно:

- штампованные (печатные) диски CD имеют рельефную верхнюю сторону прозрачной основы, покрытую светоотражающим напылением. Ямки (pits) и ровные участки трека (lands) дают разную интенсивность отраженного луча, которая регистрируется фотоприемником. Штампованные диски изготовляются на специальном заводском оборудовании. Исходная информация для штампа берется с записанного мастер-диска, с которого за несколько технологических этапов получают пресс-формы.
- однократно записываемые диски CD-R имеют слой органического красителя, нанесенный на основу, сверху которого нанесено светоотражающее напыление (золото или сплав серебряного цвета). При записи выжигаются фрагменты красителя, в результате отраженный луч также будет промодулирован по интенсивности.
- перезаписываемые диски CD-RW, .они же CD-E, под отражающим слоем имеют регистрирующий слой, который может менять свое состояние между поликристаллическим и аморфным. Прозрачность слоя зависит от его состояния При перезаписи состояние отдельных участков изменяется: в зависимости от степени нагрева участка лучом записывающего лазера при остывании фиксируется то или иное его состояние. В отличие от печатных дисков и CD-R, отражающих около 70% мощности падающего луча, диски CD-RW обладают существенно меньшей отражающей способностью.

Помещение информации на диски может производиться как печатью с матрицы (наподобие тиражирования грампластинок), так и непосредственной записью на носитель в устройстве CD-R или CD-RW, о котором речь пойдет ниже. Диски, изготовленные матричным способом, имеют себестоимость, измеряемую единицами центов. Индивидуально записанные диски обходятся дороже — порядка $1-2. Независимо от способа записи оптические диски могут быть прочитаны на любом устройстве считывания, поддерживающем данный формат записи (конечно, при использовании качественного носителя и записывающей аппаратуры). Однако между штампованными и записанными дисками все же имеется некоторая разница. Перезаписываемые диски CD-RW по сравнению с CD и CD-R при считывании дают меньшую амплитуду сигнала. По этой причине приводы без автоматической регулировки чувствительности приемника (старые модели, до 1998 г., включая ряд моделей 8×) не могут считывать диски CD-RW. На способность привода читать CD-RW указывает логотип «Multi Read».

Для упрощения записывающей аппаратуры *на болванке* (target) — чистом диске для записи — по всей поверхности при изготовлении наносится спиральная дорожка разметки (pregroove). Разметка отпечатана на верхнем слое поликарбонатного субстрата, по ней при записи наводится головка. Эта дорожка содержит коды ATIP (Actual Time In Pregroove) — разметки диска по времени, по которой при записи и диск разбивается на кадры. На этой же разметке имеется и информация о требуемой мощности лазера и возможной скорости записи. Скорость записи зависит как от диска, так и от привода. При попытке записи на диск со скоростью большей, чем гарантированная, четкость изменения оптических свойств участков ухудшается и диск может оказаться нечитаемым.

Болванки для записи имеют маркировку типа:

- CD-R или Compact Disc Recordable -- диски с однократной записью, подходят для устройств CD-R и CD-RW;
- CD-RW или Compact Disc Rewritable — диски с многократной записью, подходят только для устройств CD-RW.

На диске также указывается возможная скорость записи. Если диск пригоден для записи на скорости 2× и выше, то на нем должна быть отметка «Multi Speed» и указана максимальная скорость, например, «4× compatible». Отсутствие таких отметок указывает на возможность записи лишь на однократной скорости. Высокоскоростные болванки, естественно, дороже низкоскоростных, но это расплата за возможную экономию времени при записи. Полностью диск на скорости 1× записывается 74 минуты (плюс еще несколько минут может потребоваться на запись служебной информации). Скорости 2×, 4×, 6×... позволяют сократить это время примерно в 2, 4, 6... раз.

Стандартный размер болванки — 74 минуты или 650 Мбайт («двоичных»), иногда в рекламных целях указывают 680 Мбайт, но уже «десятичных». Оба этих числа соответствуют 333 000 секторам по 2048 байт пользовательских данных — 681 984 000 байт. Иногда пишут даже 780 Мбайт, это те же 333 000 секторов, но уже «сырых» данных по 2336 байт без ECC. Бывают болванки и большего размера, вмещающие до 80 минут аудио (около 700 Мбайт данных), но они существенно дороже и их используют редко.

Диски бывают разных цветов, в зависимости от цвета отражающего и регистрирующего слоев:

- Серебряный цвет имеют печатные диски (прозрачная подложка, алюминиевый отражающий слой). Алюминий на диске хоть и медленно, но все-таки окисляется и меняет свои отражающие свойства, поэтому время жизни печатных дисков оценивают в 10-15 лет.
- В голубых и зеленых болванках (если смотреть снизу) CD-R в регистрирующем слое используется цианин (cyanine) — материал голубого цвета (cyan, откуда и название). Зеленый цвет

болванок дает золотой отражающий слой, голубой остается при отражающем слое из серебра или сплавов алюминия. Эти болванки имеют среднюю стойкость к перепадам температуры и солнечному свету. Предполагаемое время жизни диска при нормальных условиях — 75 лет.

- Золотистые болванки CD-R имеют регистрирующий слой из фталоцианина (phtalocyanine), они более стойкие к внешним воздействиям. Предполагаемое время жизни диска — 200 лет.
- Серо-коричневый цвет имеют болванки CD-RW (цвет регистрирующего слоя).

Верхняя поверхность болванки может быть защищена довольно прочным покрытием, на котором можно даже печатать этикетки специальным принтером для дисков. Это отражается пометкой «Scratch resistant printable surface» на упаковке диска. От руки диск можно подписывать только мягким фломастером с войлочным пером и чернилами на водной основе, но ни в коем случае не шариковой ручкой и не карандашом.

Диски рекомендуется хранить в специальных пластмассовых футлярах, где они фиксируются за отверстие. Для того чтобы аккуратно вынуть диск, нужно одним пальцем нажать на фиксатор, а двумя другими взять диск за торцевые поверхности. При этом диск не будет сильно изгибаться (при нажатии фиксатор отпускает диск) и пачкаться руками. Особенно чувствительны к изгибу диски CD-RW, поскольку их регистрирующий слой находится в аморфном (полужидком) состоянии. Хранить диски, которыми пользуются неоднократно, в конвертах не рекомендуется — при вынимании и укладке поверхности диска будут неизбежно царапаться. В такой упаковке часто приходят дистрибутивы ПО, а также диски, вложенные в книги.

Все диски следует беречь от деформаций, царапин, нагревания, действия солнечных лучей.

Болванки обычно продаются в футлярах, коробками по 10,50,100 и 500 штук. Для крупных потребителей есть и «шпиндельный» вариант поставки — 100 или более дисков продаются стопкой, без индивидуальных футляров. Это удешевляет поставки, поскольку сам диск в несколько раз легче и тоньше футляра; болванки везут издалека, а футляры можно использовать и местного производства.

Материал регистрирующего слоя CD-R подвержен старению: запись критична к оптическим свойствам материала, которые со временем изменяются. Поэтому время жизни болванок до записи ограничено несколькими годами (изготовители обещают 5-10 лет). Этот материал также чувствителен к ультрафиолетовым лучам и солнечному свету.

7.3.2. Устройство приводов CD-ROM, CD-R и CD-RW

По устройству считыватели и рекордеры компакт-дисков напоминают обычные дисководы. Однако кроме приводов шпинделя и головки они имеют еще и механизм загрузки диска и более сложную головку. Головка состоит из лазерного излучателя, фотоприемника и наклонного зеркала, зафиксированных на подвижной каретке головки. На каретке на качающейся подвеске закреплена пластмассовая фокусирующая линза. С линзой связана малогабаритная катушка индуктивности, помещенная в поле постоянного магнита, установленного на каретке. Катушка и магнит образуют магнитоэлектрический привод подвески линзы, обеспечивающий ее перемещение в направлении, перпендикулярном плоскости диска. Привод подвески линзы под управлением контроллера, встроенного в накопитель, обеспечивает точную фокусировку лучей оптической системы на светоотражающем слое диска, отслеживая биения поверхности диска при вращении. Понятно, что из-за инерционности системы фокусировки есть предел скорости вращения, на котором система уже не успевает отслеживать колебания. При юстировке оптической системы линзу выставляют параллельно плоскости диска с помощью регулировочных винтов на каретке. Механика привода достаточно проста, но следует помнить о нежности пластмассовой линзы: неаккуратная чистка может оставить на ней микроскопические царапины, и считывание станет неустойчивым. Существуют накопители с самоочищающимися линзами, что называется self-cleaning lenses. Для изоляции от окружающей среды (drive sealing) в накопителях могут применяться двойные пылезащитные дверки.

Способ загрузки диска в накопитель зависит от конструкции. Больше всего *распространены устройства с выдвигающимся лотком,* или подносом, так называемые накопители Tray-Type (tray — поднос). Диск кладется на лоток рабочей стороной вниз, после чего специальный привод затягивает поднос внутрь устройства и укладывает диск на шпиндель. Движением лотка можно управлять программно или с помощью кнопки на лицевой панели устройства. Лоток втягивается даже просто от легкого подталкивания (при включенном питании). Устройства с лоточной подачей самые дешевые. Их недостаток заключается в том, что диск приходится брать руками, при этом его можно испачкать, уронить или поцарапать.

Существуют так называемые *накопители Caddy-Type,* у которых CD укладывается в специальную защитную кассету (caddy), аналогичную защитному чехлу 3" дискеты. Кассета просто вставляется в окно накопителя, и специального привода для загрузки диска не требуется. Такое решение предохраняет диски от случайного повреждения и позволяет их загружать в любом положении накопителя. При частой смене дисков желательно иметь несколько кассет, а в идеале в таких кассетах следует диски и хранить, но это несколько накладно.

Существуют также накопители, не очень корректно названные *CD-Changer,* в которые можно одновременно установить несколько CD (до четырех). В этих устройствах механизм загрузки иной: диск (без чехла) вставляется в щель, и специальный привод затягивает его внутрь корпуса. После этого устройству можно «скормить» следующий диск. При извлечении диск выдвигается из щели примерно наполовину, после чего его вынимают рукой. Существуют и варианты с кассетой-магазином на несколько дисков, которая «заряжается» и вставляется в устройство. Считывание в таком накопителе возможно, конечно, только с одного (активного) диска, а его смена занимает 1-5 с и производится вручную от кнопок пульта или под управлением программы. Однако неудачная программная поддержка (по крайней мере, первыми версиями драйверов для Windows 95) может отбить охоту использовать этот сервис, особенно когда прослушиваешь аудиодиск — ведет себя это устройство слишком суетливо.

Накопители CD-ROM имеют внешние размеры, соответствующие 5" формату устройств половинной высоты. Считывать диски CD-ROM может в любом положении, но загрузка диска с лотка не в горизонтальном положении крайне затруднительна и небезопасна как для диска, так и для накопителя. Есть лотки и со специальными фиксаторами, поддерживающими диск в вертикальном положении привода. Накопители Caddy-Type и CD-Changer менее прихотливы — они могут работать и принимать диск в любом положении.

Первые накопители имели *собственные интерфейсы,* по фирмам-производителям называемые интерфейсами *Sony, Panasonic* и *Mitsumi.* Все эти интерфейсы по смыслу напоминают 8-битный вариант шины ATA, но о совместимости с ней нет и речи. Применение накопителей с этими интерфейсами может сильно затрудняться отсутствием соответствующих драйверов устройств. Современные накопители выпускаются в основном с интерфейсами SCSI и ATA (ATAPI), возможно подключение к шине USB, PC Card и к LPT-порту. Последний вариант может применяться даже в рекордерах, но он работает только в режиме порта EPP и только с довольно мощным процессором. Для блокнотных ПК имеются малогабаритные накопители, размещаемые в их корпусах.

Вслед за первыми моделями, имеющими скорость считывания 150 Кбайт/с, вскоре появились устройства с удвоенной и учетверенной скоростью считывания — Double-Speed, Quadro-Speed (QuadSpin). Для устройств с большей скоростью слов уже не нашлось, и теперь кратность скорости обозначают числами: 2×, 4×, 6×, 8×, 10×, 12×, 16×, 20×, 24×... 52×... Вплоть до 8-12× устройство действительно обеспечивало скорость n×150 Кбайт/с по всему объему диска. Однако при большей кратности для считывания внутренних витков спирали потребовалась бы слишком высокая частота вращения (чтобы добиться требуемой линейной скорости). При большой частоте вращения усиливаются биения диска из-за его несбалансированности (даже из-за рисунка этикетки), и система слежения уже не справляется с обеспечением фокусировки на прыгающем треке. Накопители с большой кратностью обеспечивают указанную скорость лишь на внешних витках спирали, а постоянную линейную скорость обеспечивают не по всему радиусу — ближе к центру переходят на постоянную угловую скорость. Есть накопители, которые держат постоянную угловую скорость на всем диске, а требуемую скорость (например, для аудиодиска) поддерживают за счет буферной памяти. Из-за возможных повторных считываний неустойчиво воспроизводимых данных реальная скорость может оказаться ниже номинальной даже на внешних витках. Чем выше кратность скорости накопителя, тем обычно меньше и время доступа. Естественно, что высокоскоростные накопители часто более критичны к носителям (плохо читают «пиратские» диски из-за возможно низкого качества носителя и тиражирующей аппаратуры).

Практически все устройства CD-ROM позволяют воспроизводить и аудиодиски, для чего они имеют встроенные ЦАПы (цифроаналоговые преобразователи) и *аналоговый интерфейс с* линейным выходом стереосигнала уровня 0,2 В. На лицевую панель обычно выносят гнездо для подключения наушников и регулятор громкости. Для проигрывания аудиодисков накопители часто снабжают кнопкой, по которой можно начать воспроизведение без помощи программных средств. Если при запуске аудиодиска индикатор на лицевой панели CD-ROM светится, а звука нет, причину следует искать в несовпадении разводки аналогового интерфейсного кабеля с разъемом звуковой карты. В принципе, возможно и считывание аудиодисков в цифровом виде по интерфейсу передачи данных с целью дальнейшей цифровой обработки или сохранения на другом носителе, но это позволяют осуществить не все накопители. Существуют и накопители, считывающие аудиодиски с выходом на цифровой интерфейс S/P DIF, применяемый в цифровой аудиоаппаратуре.

CD-рекордеры по конструкции не сильно отличаются от устройств чтения, но их лазер в режиме записи имеет гораздо большую мощность. Практически все современные рекордеры способны и записывать, и перезаписывать диски, в зависимости от типа болванок. Способности рекордеров (как, впрочем, и считывателей) сильно зависят от версии встроенного микропрограммного обеспечения (firmware), и ряд проблем, возникающих при эксплуатации, решается обновлением встроенного ПО. Для этого используются специальные утилиты или перепрограммируется ПЗУ привода.

Стоит отметить, что время наработки на отказ у устройств CR-R/RW значительно меньше, чем у CD-ROM. Кроме того, из-за более сложной и тяжелой головки время доступа CD-R/RW в режиме считывания больше, чем у обычных CD-ROM, да и скорость считывания ниже. По этим причинам

использовать CD-R/RW для интенсивного чтения дисков нецелесообразно.

Применительно к устройствам чтения и записи компакт-дисков принята следующая терминология:

- *CD-ROM* — устройство считывания компакт-дисков. Обозначение 2×, 4× ...52×... указывает скорость считывания (максимальную). Приводы, действительно выдерживающие указанную кратность скорости по всему объему, называют TrueX.
- *CD-R, CD-Recorder* — устройство для записи дисков CD-R, способное также считывать печатные и записываемые диски. Обозначение вида 2×/4× указывает скорости записи/считывания.
- *CD-RW, CD-ReWriter* — устройство для записи дисков CD-R и CD-RW, способное также считывать печатные, записываемые и перезаписываемые диски. Обозначение вида 4×/2×/32× указывает скорости записи (на CD-R)/перезаписи (на CD-RW)/считывания. Самые быстрые (2000 г.) устройства обеспечивают скорости 16×/10×/40×, более распространены 12×/10×/32×.
- *Multisession CD-ROM* позволяют считывать данные, записанные за несколько сеансов, или сессий.
- *XA-Ready CD-ROM* не имеют собственного ADPCM-декодера, и аудиоданные формата XA могут быть считаны только с помощью звуковой карты.
- *MultiRead CD-ROM* — приводы, способные считывать диски различных форматов, включая диски CD-RW, а также диски, записанные в пакетном режиме. Старые приводы (включая и некоторые модели 8×) не имеют этой способности.
- *Multimedia CD-ROM* — привод, удовлетворяющий спецификации MPC (Multi-Media PC). Должен иметь внешний аудиовыход для проигрывания CD-DA, скорость не ниже 1× (MPC1), 2× (MPC2) или 4× (MPC3); начиная с MPC2 должен читать и многосеансовые диски, включая режим секторов Mode 2. Спецификация MPC1 (Level l) была принята в 1993 г. (процессор — 386SX), MPC3 — в 1995 г. (процессор Pentium-75; MPEG-1 декодер 352×240/288,30/25 кадров/с). Более высоких уровней MPC не вводили.

7.3.3. Организация дисков CD

Как уже говорилось, вместо концентрических треков обычных дисков оптический диск имеет одну спиральную дорожку, начинающуюся с внутренней части рабочей поверхности диска. Специфика записи — требование к непрерывности записанной области в сочетании с невозможностью перезаписи (кроме дисков CD-RW) — не позволяет рассматривать оптический диск как носитель с прямым (произвольным) доступом к данным, что обычно ассоциируется с понятием «диск». Минимальной адресуемой единицей хранимой информации является *блок* «полезной» длиной 2352 байт, который также называют *сектором.* Система адресации данных произошла от аудиодисков, и координаты каждого блока описываются в форме абсолютного времени (A-Time) Min:Sec:Frac — минуты (0-73), секунды (0-59), номер блока (фракция) в секунде (0-74). Отсчет абсолютного времени начинается с начала программной области (сразу после вводной области). Блок аудиодиска целиком используется для данных. Блок диска CD-ROM может содержать 2048 байт данных, защищенных от ошибок ECC- и EDC-кодом, или 2324 (или 2336) байт информации с меньшей защищенностью, остальные байты используются для служебной информации. Последовательность блоков одного назначения (формата) объединяется в *трек* (в аудиодисках треку соответствует одна музыкальная запись). Минимальный размер трека — 300 секторов (4 секунды), максимальный — весь диск. На диске может быть до 99 треков с номерами 01-99. В начале диска расположена *вводная зона* (lead-in), занимающая до 4500 секторов (1 минута, около 9 Мбайт), в конце — *выводная зона* (lead-out) 6750 секторов (1,5 минуты, 13,5 Мбайт). У многосеансовых дисков (см. ниже) после выводной зоны одной сессии следует вводная зона следующей сессии.

Во вводной зоне располагается *таблица содержания диска* TOC (Table Of Content), в которой описаны координаты (абсолютное время) начала каждого трека, а также начала выводной зоны.

Существует ряд стандартов, описывающих компакт-диски. Часть из них коротко называется по цвету обложки исходного документа, другие имеют более официальные названия. Физические форматы описываются следующими документами:

- *Red Book* — «Красная книга» — описывает формат обычных аудиодисков, называемых CD-DA (Digital Audio — цифровой звук).
- *Yellow Book* (ISO 10149) — «Желтая книга» — описывает физический формат CD для хранения данных — CD-ROM.
- *Green Book* — «Зеленая книга» — описывает физический формат дисков CD-I (Interactive). На этих дисках хранится видеозапись со звуковым сопровождением для воспроизведения на видеоплейере со стандартным телевизионным видеомонитором.
- *Orange Book* — «Оранжевая книга» — описывает физический формат записываемых дисков CD-R(Recordable — с возможностью записи). Ее часть 1 (Part I) посвящена магнитооптическим компакт-дискам CD-MO (Magneto-Optical); часть 2 (Part II) — однократно записываемым CD-WO (Write-Once), включая «гибридную» спецификацию для PhotoCD; часть 3 (Part III) описывает перезаписываемые диски CD-RW (Rewritable).

- • *White Book* — «Белая книга» — описывает формат видеодисков VideoCD, основанных на структуре сектора CD-ROM XA. Для сжатия данных используется MPEG.

7.3.3.1. CD-DA

Стандартом на аудиодиски CD-DA (Compact Disk Digital Audio — компакт-диск для цифрового звука) является IEC 908 — «Красная книга» (1982 г.). Поскольку оптическое считывание по своей природе имеет высокий уровень ошибок распознавания, в аудиодисках была принята довольно сложная система кодирования потока цифровых выборок. Данные на треке записаны в виде последовательности *канальных кадров* (frame). Каждый байт информации кадра (8 бит) кодируется 14-битным канальным символом, кодирование называется EFM (Eight to Fourteen Modulation — кодирование «8-14»), что позволяет улучшить распознаваемость символов. Канальные символы отделяются друг от друга трехбитными зазорами, формируемыми по определенным правилам: в генерируемой битовой последовательности число следующих друг за другом нулей лежит в диапазоне 2—10; последовательность битов 10000000000100000000000010 может встретиться только в поле синхронизации; мощность низкочастотной составляющей спектра оптического сигнала, снимаемого при считывании, минимизируется. Состав кадра (в канальных символах и без указания зазоров) следующий:

- синхронизация (последовательность 24 канальных битов);
- символ субкода;
- 12 символов данных;
- 4 символа контрольного кода;
- 12 символов данных;
- 4 символа контрольного кода.

Таким образом, в кадре, состоящем из 588 канальных битов, имеются лишь 192 информационных бита (24 байта), 64 бита контрольного кода (8 байт) и 1 байт субкода. Последовательность из 98 кадров и образует блок (2352 байт данных), адресуемый вышеописанным способом. Блок содержит 588 выборок сигнала (2 канала по 16 бит), что при скорости считывания 75 блоков/с соответствует частоте дискретизации 44,1 кГц.

Благодаря избыточности канального кодирования и применению контрольных кодов удается обнаружить подавляющее большинство ошибок и многие из них скорректировать. Для обнаружения и исправления ошибок применяют чередующийся код Рида-Соломона CIRC (Cross-Interleaved Reed-Solomon Code), что позволяет получить уровень битовых ошибок порядка 10^{-9}. Ошибочные (но не восстановленные) выборки игнорируются, в этих местах аудиосигнал интерполируется. Для того чтобы уменьшить субъективные потери качества звука (щелчки) из-за ошибок считывания канальных символов, выборки перемешивают и «размазывают» по блоку. Благодаря этому приему локальные дефекты дорожки (например, от радиальных царапин) не приводят к потере большого числа смежных выборок (вспомним о микронных размерах ямок) и легко сглаживаются интерполятором.

Символом *субкода* в каждом кадре кодируется байт — 8 бит субканалов P, Q, R, S, T, U, V, W. При считывании блока собираются 98-битные слова каждого субканала, которые используются для несения служебной информации. Нетрудно видеть, что пропускная способность каждого подканала равна 1/192 от пропускной способности основного канала данных. Реально из восьми субканалов в аудиодисках используются лишь P и Q, остальные были заложены на развитие стандарта.

Субканал используется для маркировки начала/конца аудиотрека: перед окончанием каждого трека (и вводной зоны) в канале записано 2-3 секунды единиц, переход в «0» отмечает начало звучания нового трека. В зависимости от длительности паузы между треками (она может быть любой — аудиозаписи могут идти и встык друг за другом) область единиц может начинаться как во время звучания предыдущего трека, так и после него — информативен переход из «1» в «0». На выводной зоне диска (lead-out) последовательности всех нулей и единиц чередуются с частотой 2 Гц. *Субканал Q* хранит информацию о положении блоков и дополнительную информацию о содержимом треков. Его структура следующая:

- 2 бита синхронизации.
- 4 бита управления:
  - бит 3 — число каналов: 2 по 16 бит — обычные стереодиски, или 4 по 8 бит — квадрафонические диски (реально не существуют);
  - бит 4 свободен;
  - бит 5 — разрешение цифрового копирования. Обычно плейеры при единичном значении этого бита запрещают работу аудиодекодера (для спасения усилителя, колонок и ушей слушателя), считая бит признаком цифровых, а не аудиоданных;
  - бит 6 — признак частотных предыскажений (pre-emphasis). Для повышения отношения сигнал/шум при записи уровень верхних частот поднимают на 20 дБ; при проигрывании такой записи верхние частоты должны быть понижены, на что и указывает данный бит.
- 4 бита — режим использования подканала (определены 3: 0001 — q-Mode 1, 0010 - q-Mode 2 и

0011 - q-Mode 3.

- 72 бит — данные подканала (9 байт), см. ниже.
- 16 бит — CRC-код подканала.

Данные субканала в основном трактуются как байты с двумя упакованными BCD-цифрами (00-99), за исключением специальных признаков, в которых используются шестнадцатиричные числа (например, A0).

Во вводной зоне диска (lead-in) Q-субканал в режиме q-Mode 1 содержит таблицу содержимого диска *TOC,* в которой указываются координаты начала каждого трека и начала выводной зоны (lead-out). В каждом блоке содержится одна запись таблицы, причем каждая запись трижды повторяется в трех смежных блоках. Запись содержит указатель номера трека и позицию (P-MIN, P-SEC и P-FRAC). После троекратных повторений записей с указателями на все треки пользовательской области в таблице также троекратно повторяются записи со специальными значениями указателя: A0, A1 и A2. Вся таблица повторяется многократно по всей вводной зоне и может закончиться на любой записи.

Структура записи TOC (9 байт данных подканала):

- 1 байт TNO=0 — номер текущего трека.
- 1 байт указателя (pointer): 00-99 — номер трека, описываемого элементом TOC; A0, A1 и A2 — специальные признаки.
- 3 байта — время текущего блока (min:sec:frac) от начала вводной зоны.
- 1 байт нулей.
- 3 байта — позиция (P-MIN, P-SEC и P-FRAC). Для записей с указателем 01-99 задает абсолютное время первой секции указанного трека, в которой Index=01. Для записей с указателем A2 указывает абсолютное время начала выводной зоны. Для записей с указателем A0 и A1 в байте P-MIN указывается номер начального и последнего трека пользовательской области соответственно.

В зоне треков 01-99 и выводной зоне Q-субканал в режиме q-Mode 1 указывает текущее положение и имеет следующую структуру:

- 1 байт TNO — номер текущего трека (01 -99) или признак выводной зоны (AA).
- 1 байт — индекс (01 -99) внутри трека или пауза между треками (00). Индексы могут быть расставлены в произвольных местах трека, с их помощью можно указать «логическую» позицию внутри музыкального произведения. Индексы должны начинаться с 01 и последовательно нарастать. В выводной зоне в поле индекса записано 01.
- 3 байта — относительное время внутри трека или время до конца паузы (MIN, SEC и FRAC). Относительное время устанавливается в 00:00:00 в первой секции, у которой Index=01. В начале паузы время устанавливается в соответствии с ее длительностью и декрементируется с каждой секцией до нуля (до появления Index 01).
- 1 байт нулей.
- 3 байта — абсолютное время на диске (A-min:A-sec:A-frac). Устанавливается в ноль в начале первого трека пользовательской области, инкрементируется с каждой секцией до конца выводной зоны.

Q-субканал в режиме q-Mode 2 содержит каталоговый номер диска — 23 BCD-цифры (то же, что и в штрих-коде) и 1 байт номера блока в секунде (абсолютное время берется из смежного блока с Q-каналом в Mode 1). Каталоговый номер должен быть единым на всем диске. Если каталоговый номер не используется, то либо номер должен указываться нулевым, либо q-Mode 2 не должно быть на диске.

Q-субканал в режиме q-Mode 3 содержит 60-битное поле с кодом ISRC (International Standard Recording Code — стандартное международное кодирование записей) для каждого трека и 1 байт номера блока в секунде. В ISRC входят коды страны и владельца, год и серийный номер записи. Эта информация указывается далеко не всегда.

Способ кодирования и алгоритмы обнаружения и исправления ошибок для аудиодисков были выбраны исходя из соображений относительной простоты реализации. Потеря отдельных выборок для аудиосигнала не фатальна — хороший интерполятор сделает это незаметным для большинства слушателей.

При проигрывании CD-DA в компьютере на приводе CD-ROM происходит декодирование сигнала и преобразование его в аналоговую форму. В идеале на ЦАП должны приходить корректные (или хотя бы интерполированные) цифровые коды выборок точно с частотой 44,1 кГц. Реально из-за различных дестабилизирующих факторов происходит дрожание (jitter) — колебания фазы считанных цифровых выборок, которое должно быть минимизировано за счет буферизации и генерации стабильной опорной частоты. В процессе проигрывания CD-DA участвует корректор ошибок, интерполятор, ЦАП и фильтры привода CD-ROM, и на звуковую карту компьютера поступает аналоговый стереосигнал. Его качество определяется исключительно приводом; в дешевых приводах могут использоваться упрощенные схемы (например, ЦАП с разрядностью менее 16 бит, значительное дрожание, плохие фильтры, большой

уровень шумов). Возможность цифрового считывания аудиодисков в приводах CD-ROM — DAE — предоставляют далеко не все модели приводов, а также не всякое ПО (драйверы). Скорость этого считывания может быть ниже, чем считывание обычных данных (DAE со скоростью 20× считается выдающимся результатом).

Несмотря на то, что на аудиодисках информация хранится в цифровом виде, даже цифровое копирование аудиозаписей не обеспечивает идентичности копии и оригинала. Это связано с хотя и не очень высоким (10^{-9}), но ощутимым уровнем битовых ошибок. Если «сырые» аудиоданные оригинала скопировать на другой диск, то случайные ошибки считывания запишутся на копию, а при последующем считывании к ним могут добавиться и новые. Копирование через аналоговый выход тем более не может быть точным, поскольку здесь будут присутствовать коррекция и интерполяция, погрешность преобразования ЦАП привода и АЦП звуковой карты, дрожание фаз выборок, биения частот дискретизации привода и АЦП (генераторы асинхронны), шумы в аналоговом тракте и тому подобные неприятности.

7.3.3.2. CD-ROM

Компакт-диски, предназначенные для считывания данных — CD-ROM — были описаны в документе Sony и Philips, известном под названием «Желтая книга». На ее основе был выпущен стандарт ISO/IEC 10149 (1995 г.), и с небольшими изменениями в 1996 г. вышел стандарт ECMA-130 «Data interchange on read-only 120 mm optical data disks (CD-ROM)». Этот стандарт описывает физический формат компакт-диска, предназначенного только для считывания данных — CD-ROM. Диск CD-ROM содержит треки с цифровыми данными, а также может содержать аудиотреки (в обычном формате CD-DA).

Для хранения цифровых данных недопустима даже одна битовая ошибка на файл, поэтому для CD-ROM приняли несколько иной формат сектора, нежели в CD-DA. Для преемственности технологии и совместимости дисков формат канального кадра (588 канальных битов) сохранили. Сектор, содержащий 2352 байта, начинается с заголовка:

- байты 0-11 — поле синхронизации (1 байт 00h, 10 байт FFh, 1 байт 00h);
- байты 12, 13 и 14 — адрес сектора (Min, Sec, Frac);
- байт 15 — номер режима сектора (Sector Mode).

Способ задания адреса сектора зависит от его положения, адрес повторяет информацию Q-субканала, полученную с декодера EFC («14 - 8») на момент поступления поля синхронизации на дескремблирование (см. ниже). Минуты (поле Min) в заголовке соответствуют минутам Q-субканала, сложенным с числом A0h. Во вводной зоне адрес задает положение сектора относительно ее начала, в зоне данных и выводной зоне он соответствует абсолютному времени (A-Time) с точностью ±1 с.

Использование оставшейся части сектора зависит от режима, определенного в заголовке:

CD-ROM Mode 00: байты 16-2351 нулевые (нет данных).

CD-ROM Mode 01 (максимально защищенные данные):

- байты 16-2063 — данные (user data, 2048 байт);
- байты 2064-2067 — контрольный код EDC (4 байта), 32-битный код CRC для байт 0-2063;
- байты 2068-2075 - нули (8 байт);
- байты 2076-2351 — контрольный код ECC, (276 байт):
  - байты 2076-2247 — P-паритет (172 байта), подсчитанный для байтов 12-2075;
  - байты 2248-2351 — Q-паритет (104 байта), подсчитанный для байтов 12-2247.

CD-ROM Mode 02 (минимально защищенные данные):

- байты 16-2351 — данные (user data, 2336 байт).

Во всех режимах (даже Mode 0, где нет данных) блок защищен кодированием CIRC (на уровне кадров, как и в аудиодисках). В режиме Mode 1 защита тройная: CIRC, EDC (Error Detection Code — код для обнаружения ошибок, 4 байта) и ECC (Error Correction Code — код для исправления ошибок, байты P- и Q-паритета, которые иногда путают с битами одноименных субканалов). В режиме Mode 2 забота о дополнительном повышении достоверности возлагается на внешнее программное обеспечение (по отношению к устройству CD-ROM).

При записи последовательность байтов 12-2351 проходит через скремблирование (перемешивание битов по определенному алгоритму, обратный процесс называется дескремблированием), байты синхронизации скремблером не затрагиваются. 2352 байта, выходящие со скремблера, отображаются в последовательность 24-байтных кадров (F -Frames), в которых соседние байты (четные и нечетные) попарно меняются местами. Эти кадры проходят через CIRC-кодирование и превращаются в 32-байтные кадры (F -Frames), в которых появляются по 8 дополнительных байтов паритета. К каждому из этих кадров добавляется контрольный байт, и получаются 33-байтные кадры (F -Frames). Байты этих кадров

преобразуются в 14-битные канальные символы, из которых формируется канальный кадр. Канальный кадр аналогичен CD-DA (см. выше), в нем имеется синхронизирующий заголовок, а символы разделены еще 3-битными промежутками. Наконец, эта последовательность кодируется «ямками» — на месте каждой битовой «1» делается переход от равнины к ямке или обратно. При считывании диска все эти преобразования выполняются в обратном порядке.

Кроме основного канала, по которому проходят блоки указанной структуры, в CD-ROM остаются и субканалы. Как и в CD-DA, биты данных подканалов собираются из символов субкода 98 смежных кадров, образующих *секцию*. *Субканал P* используется так же, как в CD-DA — отмечает конец трека. Последняя секция с единицами в P-субканале является первой секцией, несущей пользовательские данные. Во вводной зоне P-субканал несет нули. *Субканал Q* почти такой же, как в CD-DA, но отличается трактовкой управляющих битов: комбинация 0100 запрещает цифровое копирование данных, 0110 — разрешает; остальные комбинации используются только в CD-DA. Поле управляющих битов может меняться только во время паузы длительностью не менее 2 с, кроме бита разрешения копирования, который может меняться между секциями. Для цифровых треков возможны режимы подканала Q-Mode 1 и 2, Q-Mode 3 применим только к аудиотрекам. Информация режима Q-Mode 1 встречается 9 раз в последовательности из 10 секций, информация в Q-Mode 2 (если присутствует) встречается не менее одного раза на 100 секций. Для CD-ROM установлено требование к нумерации треков и индексов: они должны начинаться с 01 и нарастать последовательно, без пропуска номеров.

Секторы и секции асинхронны друг относительно друга — их начала могут не совпадать. Начало сектора определяется по полю синхронизации, которое не трансформируется при скремблировании.

Информационная область диска должна содержать вводную зону (lead-in area), зону пользовательских данных (user data area) и выводную зону (lead-out area). Вводная и выводная зона содержат по одному информационному треку (аудио или цифровому), в пользовательской зоне треков может быть от 1 до 99. Треки с цифровыми данными должны быть структурированы в виде последовательности секторов, аудиотреки должны соответствовать IEC 908.

Треки в информационной области могут содержать области, необходимые как связующие:

- Пауза (pause) — часть трека, в которой передается только информация субканалов, а пользовательских данных нет.
- Предзазор (pre-gap) — начало цифрового трека, не содержащее пользовательских данных и кодируемое как пауза. Первый интервал зазора — не менее 75 секций (1 секунда), в котором управляющее поле Q-подканала несет режим сектора предыдущего трека. Второй интервал зазора — не менее 150 секций (2 секунды), в котором управляющее поле Q-подканала несет режим сектора данного трека; здесь уже должна быть структурированность в секторы.
- Постзазор (post-gap) — конец цифрового трека, не содержащий пользовательских данных и кодируемый как пауза. Не менее 150 секций (2 секунды), в котором управляющее поле Q-подканала и байт режима сектора несут ту же информацию, что и в части данного трека, несущей данные. Структурированность в секторы сохраняется.

Вводная зона, если она является цифровым треком, структурируется в секторы и завершается постзазором. Выводная зона, если она является цифровым треком, структурируется в секторы, но не содержит предзазора. Для пользовательской зоны определены следующие правила:

- Если первый трек цифровой, он должен начинаться с паузы не менее 2 секунд, кодируемой как вторая часть предзазора.
- Цифровой трек (не первый) должен начинаться с предзазора, если ему предшествует аудиотрек или трек с иным режимом сектора.
- Цифровой трек должен завершаться постзазором, если за ним следует аудио-трек или выводная зона.

7.3.3.3. CD-R и CD-RW

Структура записываемого (CD-R) и перезаписываемого (CD-RW) диска несколько отличается от CD-DA и CD-ROM. До штатного начала вводной зоны здесь имеются зоны PCA и PMA, известные только рекордерам (и их программному обеспечению). Диск начинается с зоны калибровки мощности лазера PCA (Power Calibration Area), используемой для предварительной настройки рекордера перед записью. Далее следует область для временного хранения информации о текущем состоянии записи PMA (Program Memory Area). В эту область заносятся координаты начала и конца каждого уже записанного трека. Информация в PMA записывается перед тем, как диск извлекают из рекордера.

Вводная зона резервируется для хранения TOC, запись в нее производится только после записи всех треков в программную область. При закрытии диска (disk finalizing) сразу за программной областью записывается выводная зона, а затем вводная зона с таблицей TOC. Если диск записывается не за один прием, то TOC формируется по информации из PMA. Запись программной области может производиться как целиком (DAO — диск за один прием), так и по частям: трек за прием (TAO) или инкрементно

(пакетная запись). В любом случае, закрыт диск или нет, информация в программной области при изъятии диска из рекордера должна быть записана одной непрерывной областью. Запись в пакетном режиме не гарантирует совместимости со всеми считывателями, ориентированными только на стандарт CD-ROM («Желтая книга»).

Лазер не может включиться мгновенно в режим записи, и секторы, записываемые во время переходного процесса, могут оказаться нечитаемыми. Поэтому, если запись производится по частям, в местах стыковки приходится оставлять неиспользуемые промежутки.

При *потрековой записи* все секторы пользовательских данных трека записываются единым непрерывным потоком.

При *пакетной (инкрементной) записи* секторам с пользовательскими данными должен предшествовать один связующий сектор (link) и четыре вводных сектора (run-in). После пользовательских секторов должны следовать два выводных сектора (run-out). На связующем секторе лазер включается, а вводные и выводные секторы обеспечивают синхронизацию. Эта последовательность называется *пакетом,* и длиной пакета считается число пользовательских секторов. Длина пакета, в зависимости от типа трека, может быть как фиксированной (в пределах трека), так и переменной. Пакеты должны записываться последовательно друг за другом, разрывы между пакетами не допускаются. Пакетная запись уменьшает пространство диска, доступное для пользовательских данных, из-за накладных расходов (7 «лишних» секторов на каждый пакет). Накладные расходы тем меньше, чем больше длина пакета. Тип сектора в пакете кодируется битами 7, 6, 5 байта режима сектора (15-й байт сектора):

- 000 — сектор данных
- 001 — 4-й сектор Run-in
- 010 — 3-й сектор Run-in
- 011 — 2-й сектор Run-in
- 100 — 1-й сектор Run-in
- 101 — связующий сектор
- 110 — 2-й сектор Run-out
- 111 — 1-й сектор Run-out.

При любом методе записи каждый трек должен начинаться с *предзазора* согласно «Желтой книге», а если по ней предзазор не предусматривается, то все равно записывается зазор длительностью 2 с (2-я часть). Во время записи второй части предзазора записывается сектор-дескриптор трека.

При инкрементной записи, если первому пользовательскому сектору на треке предшествует связующий и вводные секторы, предзазор должен кончаться на последнем вводном секторе (run-in), так что начало трека будет соответствовать адресу первого пользовательского сектора.

Сектор-дескриптор трека несет информацию о размере второй части предзазора, номере трека, методе записи трека (TAO, пакеты фиксированной длины или пакеты переменной длины), длине пакета (если она фиксирована). Дополнительно в нем может быть описание предыдущего трека.

В перезаписываемых дисках имеется возможность перезаписи отдельных треков и даже пакетов секторов. Однако требование непрерывности последовательности уже записанных секторов сохраняется. Используя пакетную запись, перезаписываемые диски можно организовать как устройства прямого доступа, но для соблюдения условия непрерывности диск должен быть отформатирован. Форматирование CD-RW подразумевает запись треков с пакетами фиксированной длины.

Для *идентификации секторов* на диске вместо «трехмерного» (MIN, SEC и FRAC) адреса, фигурирующего в заголовке секторов и Q-субканале, удобно использовать *логический номер сектора LSN* (Logical Sector Number). Он может определяться по одному из двух методов:

- 1-й метод — для всех секторов всех пользовательских треков LSN считать по формуле LSN=(((MIN×60) + SEC)×75 + FRAC) - 150, где MIN, SEC и FRAC - адрес из заголовка сектора. Вычитание 150 исключает из нумерации секторы зазора, начинающего первый трек.
- 2-й метод — по вышеприведенной формуле считается логический номер только для первого сектора каждого трека, и далее последовательными числами нумеруются только все секторы с пользовательскими данными (связующие, вводные и выводные секторы пропускаются). Этот метод используется лишь для пакетной записи с фиксированным размером пакета. Для потрековой записи он не нужен (лишних секторов нет), а для переменной длины пакета он неприменим, поскольку места пропускаемых секторов непредсказуемы.

Нетрудно видеть, что нумерация логических секторов по второму методу между концом пакетно-записанного трека и началом следующего трека не будет непрерывной. По первому методу будут разрывы нумерации секторов с данными на границе каждого пакета.

Особой разновидностью организации записываемых дисков являются *многосеансовые диски* (multisession CD), описанные ниже.

7.3.3.4. CD-I

Физический формат «интерактивных» дисков CD-I (Compact Disk Interactive) описан в «Зеленой книге». На этих дисках может храниться любая смесь данных (графика, текст), аудио и видео, то есть они являются мультимедийным вариантом CD-ROM. От обычного формата CD-ROM сектор CD-I отличается наличием 8-байтного подзаголовка CD-I после байта режима и отсутствием пропуска 8 байтов после кода EDC. Формат сектора CD-I для *Mode 01:*

- байты 0-15 — стандартный заголовок CD-ROM (см. выше);
- байты 16-23 — подзаголовок CD-I (subheader, 8 байт);
- байты 24-2071 — данные (user data, 2048 байт);
- байты 2072-2075 — контрольный код EDC (4 байта);
- байты 2076-2351 — контрольный код ECC (276 байт: 172 байта P-паритета + 104 байта Q-паритета).

Диски предназначаются для проигрывания на «продвинутых» телевизионных системах, а не обязательно на компьютере. Аудиоданные могут быть записаны в разных форматах:

- CD-DA — 1 стереоканал с полосой 20 кГц, кодирование PCM, поток 171,1 Кбайт/с;
- «A» — 2 стерео/4 моноканала с полосой 17,5 кГц, ADPCM, поток 85,1 Кбайт/с на стереоканал;
- «B» — 4 стерео/8 моноканалов с полосой 17,5 кГц, ADPCM, поток 42,5 Кбайт/с на стереоканал;
- «C» — 16 стерео/8 моноканалов с полосой 8,5 кГц, ADPCM, поток 21,3 Кбайт/с на стереоканал.

Видеоданные могут храниться в форматах, совместимых с NTSC (360×240, 720×240 и 720×480) и PAL/SECAM (384×280, 768×280 и 768×560). Могут применяться следующие схемы кодирования цвета:

- DYUV (Delta Y-U/V): сокращает объем стандартного кадра до 85 Кбайт (NTSC) или 105 Кбайт (PAL);
- RGB 5:5:5:16 бит, по 5 бит на цвет плюс бит прозрачности, кадр 170 Кбайт (NTSC) или 210 Кбайт (PAL);
- CLUT (Color Look up Table): используются палитры, 4, 7 или 8 бит/пиксел. Кадр 85 Кбайт (NTSC) или 105 Кбайт (PAL).

На диске может быть информация в формате MPEG — CD-I вмещает около 60 минут видео (с аудио). С дисков CD-I началось использование MPEG в бытовой электронике.

7.3.3.5. CD-ROM XA

CD-ROM XA (eXtended Architecture — расширенная архитектура) — стандарт, являющийся мостом между «Желтой книгой» (CD-ROM) и дисками CD-I (иное название — CD-Bridge или CD-I Bridge). Архитектура допускает два режима физической записи диска: Mode 1 и Mode 2. Формат секторов Mode-1 соответствует «Желтой книге» (CD-ROM) — 2048 байт данных, защищенных ECC и EDC. Режим Mode 2 формально укладывается в CD-ROM Mode 2, используя все поле данных сектора.

Mode-2 допускает на одном диске записи двух форм: *форма 1* (Form-1) с секторами по 2048 байт данных, защищенных кодами EDC и ECC, позволяющими обнаруживать и исправлять ошибки (совпадает с CD-I Mode 1), и *форма 2* (Form-2) с секторами по 2324 байта данных, защищенных EDC, но без защиты ECC, так что здесь имеется только обнаружение ошибок. Во всех режимах и формах, естественно, сохраняются нижние ступени кодирования и защиты — EFC и CIRC. Форматы секторов следующие:

CD-ROM XA Mode 1 совпадает с CD-ROM Mode 1 (см. выше).

CD-ROM XA Mode 2 Form 1:

- байты 0-15 — стандартный заголовок CD-ROM (см. выше)
- байты 16-23 — подзаголовок CD-I (subheader, 8 байт)
- байты 24-2071 — данные (user data, 2048 байт).
- байты 2072-2075 — контрольный код EDC (4 байта)
- байты 2076-2351 — контрольный код ECC (276 байт: 172 байта P-паритета + 104 байта Q-паритета)

CD-ROM XA Mode 2 Form 2:

- байты 0-15 — стандартный заголовок CD-ROM (см. выше)
- байты 16-23 — подзаголовок CD-I (subheader, 8 байт)
- байты 24-2071 — данные (user data, 2324 байта)
- байты 2348-2351 — контрольный код EDC (4 байта)

Физический формат XA совместим с файловыми системами ISO 9660 и High Sierra, о которых речь пойдет ниже. Формат допускает чередование (interleaving) сегментов данных (форма 1) и потоков аудио- и видеоинформации (форма 2). Для аудио- и видеоинформации используются те же методы представления и сжатия, что и в CD-I. Формат XA используется для Video CD и Photo CD.

В пределах одной сессии можно использовать лишь один из режимов, но в Mode 2 возможно произвольное чередование треков с разными формами.

Диски могут быть записаны в разных физических форматах. В традиционном режиме CD-ROM Mode 1 диск имеет емкость около 650 Мбайт, в режиме Mode 2 Form 2 — 780 Мбайт. Меньшая емкость в режиме Mode 1 обусловлена большой избыточностью кодирования хранимых данных, необходимой для коррекции неизбежных ошибок (интерполяция, позволяющая безболезненно пропускать некорректно считанные выборки аудиосигнала, при хранении данных, естественно, недопустима). В Mode 2 CD-ROM XA на диск записываются «сырые» данные без кодов ECC, причем здесь могут чередоваться, например, области аудиоинформации и данных. Забота об обеспечении достоверности считываемых данных, записанных в форме 2, ложится на внешнее программное обеспечение.

7.3.3.6. Многосеансовые диски

Специальным вариантом записываемого диска является *многосеансовый диск,* в котором видимое при обычном считывании содержимое записываемого диска может меняться пользователем несколько раз. Структуру многосеансового диска иллюстрирует рис. 7.1, на котором серым цветом отмечены записанные области (две закрытые сессии и третья — незакрытая).

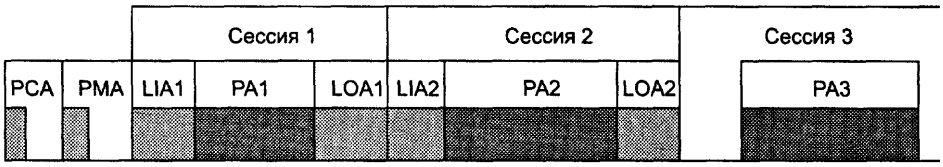

Рис. 7.1. Структура многосеансового диска: PCA — область калибровки, PMA — область сохранения координат, LIA — вводная зона, PA — область данных сессии. LOA — выводная зона

Сессией (session) называют набор треков (от 1 до 99), которому предшествует вводная зона, содержащая TOC с указателями начала каждого из этих треков. За последним треком имеется и выводная зона (lead out), начало которой также задано в TOC. Сразу за выводной зоной может быть записана вводная зона следующей сессии. Размер выводной зоны для второй и последующих сессий уменьшен до 2250 секторов (0,5 минуты, около 4 Мбайт).

Каждая сессия (структура, записанная за один сеанс) выглядит как обычный CD-ROM, но есть нюансы в записях вводной зоны. Сессия называется *закрытой,* когда ее программная область обрамлена вводной и выводной зонами. Однако в ее TOC указатель на выводную зону (A2) может указывать либо на начало выводной зоны, либо на ее конец, то есть на начало вводной зоны следующей сессии. Когда указатель описывает начало выводной зоны, диск становится *закрытым* — следующую сессию к нему уже не добавить. Когда он указывает на конец вводной зоны, на диск возможна запись последующей сессии (если хватает ресурсов: места на диске, места в PMA и номеров треков).

Первый трек первой сессии должен иметь номер 01, следующие треки — последовательно нарастающие номера. Номер первого трека очередной сессии должен быть следующим за номером последнего трека предыдущей сессии. Максимальный номер трека — 99.

Многосеансовые, или мультисессионные (multi session CD), диски содержат более одной сессии, и все сессии физически доступны для чтения. Очередная записываемая сессия может быть полностью независимой (ее TOC содержит ссылки только на ее собственные треки), а может быть и связанной *с* предыдущими сессиями (linked session). Связь может быть как на уровне треков (абсолютные координаты «старых» треков, все или частично, включают в TOC новой сессии), так и на уровне файлов (для CD-ROM). Связь на уровне файлов требует внесения ссылок на файлы прежних сессий в каталогах и таблице путей, являющихся логической частью файловой системы CD-ROM.

Возможности использования информации конкретных сессий зависят от устройства считывания и его ПО. Аудиоплейеры, как правило, считывают только первую сессию — остальными они просто не интересуются. Приводы CD-ROM и их ПО могут, в принципе, читать любую сессию. Если они будут читать последнюю сессию, появляется возможность «перезаписать» диск CD-R, на самом деле только дописывая следующую сессию. На этих свойствах построены диски *CD Plus,* они же *CD Extra,* у которых первая сессия предназначена для аудиоплейеров, а вторая (и последующие) — для приводов CD-ROM. Приводы CD-ROM и аудиоплейеры способны считывать только закрытые сессии, незакрытые сессии доступны только рекордерам. Диск, в принципе, закрывать необязательно, но может встретиться привод совместно с ПО, не желающий читать незакрытый диск.

Многосеансовая запись впервые появилась в PhotoCD, а затем и в CD-ROM XA. Сейчас на нее

распространяется стандарт, описанный в «Оранжевой книге», согласно которой многосеансовая запись может производиться в физическом формате Mode l (CD-ROM) или Mode 2 (CD-ROM XA). Все сессии одного диска должны записываться в одном из этих режимов.

Поддержка многосеансовых дисков появилась уже в ряде моделей приводов 4×, ее имеют практически все накопители 8× и более высокоскоростные. По умолчанию привод, поддерживающий многосеансовый режим считывания, должен обращаться к последней сессии. В таблице путей, записанной в этой сессии, могут содержаться и ссылки на файлы из предыдущих сессий. Таким образом, в зависимости от наличия этих ссылок через таблицу путей последней сессии оказываются доступными не только ее данные, но и любые файлы предыдущих сессий. При этом оказывается возможным и «обновление» прежних файлов, которое сводится к записи новых их версий и включению в таблицу путей ссылок только на эти версии. «Удаление» файлов сводится к тому, что ссылка на них не включается в таблицу путей последней сессии. Обычные CD-плейеры и накопители CD-ROM, не поддерживающие многосеансовый режим, читают данные TOC (и таблицу путей) только первой сессии. Некоторые программные драйверы, обнаружив носитель с физическим форматом Mode I CD-ROM (а не Mode 2 CD-ROM XA), ошибочно рассматривают его как односеансовый и обращаются к первой сессии. В результате остальные сессии оказываются недоступными, но эта проблема решается просто заменой драйвера на более корректную версию.

Какая из сессий доступна по умолчанию — первая или последняя — зависит от программного обеспечения. В MS-DOS по умолчанию доступна первая сессия, в Windows 95 — последняя. Более «ловкое» ПО позволяет выбрать номер доступной сессии.

7.3.4. Файловые системы

Для аудиодисков не требовалось создавать какую-либо специальную логическую структуру — достаточно того, что каждой аудиозаписи соответствует собственный трек. Для хранения данных требуется организация файловой системы, которая не может просто повторять дисковую файловую систему, например, MS DOS. Такая система при большом времени доступа с большим количеством каталогов и файлов (а емкость 650 Мб к этому располагает) работала бы крайне медленно. В настоящее время для CD, используемых в PC, распространены почти эквивалентные файловые системы HSF и ISO 9660, которые иногда отождествляют. Однако в структурах записей этих систем есть небольшие различия, которые должны отрабатываться программным обеспечением (но даже MSCDEX их разделяет).

HSF(High Sierra Format), он же *HSG* (High Sierra Group) — фактический стандарт на доступ к данным из среды DOS, UNIX и других ОС. Стандарт назван по географическому названию места его разработки.

ISO 9660 — первый стандарт (1988 г.) для хранения данных на CD-ROM. Данные тома CD-ROM начинаются с логического сектора 16 первого трека сессии (для первой сессии адрес в заголовке сектора будет 00:02:16), в котором находится набор дескрипторов тома. Набор должен начинаться с первичного дескриптора тома PVD (Primary Volume Descriptor). За ним может следовать произвольное число дополнительных дескрипторов SVD (Supplementary Volume Descriptor), и завершается эта последовательность дескрипторов терминатором (volume descriptor set terminator). Таким образом, один том (набор файлов) может быть описан несколькими дескрипторами. Область логических секторов 0-15 объявлена системной (ее использование стандартом не предусматривается). В секторе 17 размещается загрузочная запись. Структуры записей включают две формы представления многобайтных чисел — одну для платформ Intel (порядок байтов младший-старший), другую для платформ Motorola (порядок ~ старший-младший). Структуру тома описывает таблица содержимого (TOC или VTOC), которая хранится в его логических секторах (на треке тома). Эту таблицу иногда путают с TOC компакт-диска, записанной в Q-субканале во вводной зоне и указывающей на начала треков и выводной зоны, но в данном контексте эта таблица физического уровня нас не интересует.

Один диск (или сессия) может иметь и несколько томов, но эта возможность поддерживается не всяким ПО и практически не используется. В TOC описаны все файлы, присутствующие на диске, — имя, дата создания, атрибуты, положение всех экстентов файла (экстент — фрагмент файла, записанный в непрерывной цепочке блоков).

Файлы на диске располагаются в каталогах, образующих древовидную структуру, и каждый каталог содержит список входящих в него файлов, их атрибуты и указатели на секторы, в которых располагается начало файлов или их экстентов. Для ускорения поиска файлов на диске кроме каталогов имеется дополнительная *таблица путей* (path table), содержащая список путей (в символьном формате) ко всем подкаталогам диска и адреса их начальных секторов. Записи каталогов и таблицы путей имеют переменную длину, определяемую длиной имени. Чтобы таблицы были компактными (тогда они смогут целиком размещаться в ОЗУ), выгоднее использовать наиболее короткие имена каталогов. С учетом поиска через таблицу путей выгодно большое количество файлов распределять по большему количеству каталогов, при этом время на поиск в каталоге (перебор) сократится.

Стандарт имеет три вложенных уровня совместимости. Самые жесткие ограничения имеет *первый*

уровень (Level 1): файловая система в стиле MS DOS, имена файлов по схеме «8.3», глубина вложенности каталогов до 8. В именах файлов и каталогов могут использоваться только заглавные буквы латинского алфавита, цифры и символ подчеркивания. После имени файла через символ «.» может указываться номер версии файла, который многими ОС игнорируется. Фрагментация файлов не допускается — файл может располагаться только в непрерывной цепочке секторов. Первый уровень совместимости обеспечивает гарантированную переносимость носителя между любыми системами, поддерживающими данный формат. *Второй уровень* (Level 2) позволяет использовать длинные имена файлов с более богатым набором допустимых символов, допустима глубина вложения каталогов до 32. Этот формат воспринимается не всеми операционными системами (в частности, его не поддерживает MS DOS). *Третий уровень* (Level 3) допускает даже фрагментацию и чередование файлов. Это означает, что файл может размещаться во множестве экстентов и экстенты разных файлов могут чередоваться. Эта возможность используется при инкрементной записи дисков в формате UDF (см. ниже).

Rock Ridge — расширение стандарта ISO 9660 для ОС UNIX, обеспечивающее работу с длинными именами файлов и символьными связями.

Формат *HFS(* Hierarchical File System — иерархическая файловая система) используется вместо ISO 9660 в дисках для Macintosh, на PC эти диски воспринимают ОС Linux и OS/2. Формат допускает до 31 символа в имени файла, вложенные каталоги и некоторые специальные возможности, свойственные системе Mac.

Joliet — расширение файловой системы ISO 9660 от Microsoft для Windows 9x/NT, снимающее ограничения на имена файлов. Позволяет использовать любые символы в именах, преодолеть ограничение на длину имени «8.3», увеличить возможную глубину вложенности каталогов и снять ограничения на имена каталогов. Допускает длину имен файлов и каталогов до *64* символов (длинные имена Windows могут иметь длину до 255 символов). Символы имен кодируются двумя байтами (Unicode), что обеспечивает возможность использования различных языков. Расширение строится на возможности организации множества дескрипторов одного и того же тома, заложенной в ISO 9660. Файлы на диске размещаются так же, как и в ISO 9660, и первичный дескриптор тома так же указывает на структуры формата ISO, с таблицей путей, доступной MS DOS. Но добавляется еще и дополнительный дескриптор тома (SVD), который указывает на альтернативную таблицу путей, содержащую расширенные имена Windows. Обе таблицы путей описывают один и тот же набор файлов.

Romeo — расширение файловой системы для поддержки длинных имен (до 128 символов) Windows 95 и NT. Если имена не длиннее 31 символа, диск может читаться и на Mac. К *Joliet* отношения не имеет.

Поскольку диск CD-RW допускает перезапись, структура файловой системы ISO 9660, не ориентированная на возможность модификации уже записанных данных, для него не лучшее решение.

Файловая система UDF (Universal Data Format) с пакетами переменной длины построена иначе. Здесь файлы хранятся рядом со своими описаниями, допустима длина имен до 127 символов. Каждый пакет представляет отдельный файл (или его экстент), в начале пакета имеется описание файла (имя, дата, атрибуты, длина файла и данного экстента). Никаких общих таблиц размещения файлов и экстентов для UDF не требуется — последовательное чтение пакетов позволяет собрать все файлы диска. Конечно, для быстрого поиска нужного файла в памяти компьютера строится виртуальная таблица размещения файлов. Диск (сессия) с пакетами переменной длины может быть закрытым и с дескрипторами тома файловой системы ISO 9660 (Level 3, поскольку требуется поддержка фрагментации), тогда он будет читаться традиционными средствами. Иначе для его чтения требуется специальный драйвер UDF. Организация диска в виде пакетов переменной длины очень эффективно использует дисковое пространство, поскольку здесь мало «отходов» (на каждом файле теряется в среднем половина сектора размером 2 Кбайт). Обычная файловая система FAT 16 на таком же объеме диска имеет размер кластера 16 Кбайт, так что средние потери — 8 Кбайт на файл. Диск, «забитый под завязку» пакетами переменной длины, может и не иметь возможности закрытия под ISO 9660. Такой диск (с незакрытой сессией) будет читаем только на рекордере. Формат с переменной длиной довольно устойчив к авариям при записи. Диск будет испорчен лишь в том случае, если пропадет питание во время записи пакета. В любой другой момент прерывание процесса записи не приводит к катастрофе — файлы пишутся по принципу «все свое ношу с собой», для доступа к ним не нужны никакие таблицы.

Для перезаписываемых дисков появляется еще одна возможность организации — в виде пакетов фиксированной длины. Чистый диск форматируется пакетами фиксированной длины (обычно 2 сектора — 4 Кбайт) и выглядит совсем как большая дискета. Правда, при этом велики накладные расходы на организацию формата — около 18%. Обычная (650 Мбайт) чистая болванка может быть отформатирована на 494 Мбайт полезного пространства. При этом диск в обычном смысле не закрывается, и с ним может общаться (даже по чтению) только устройство-рекордер. Обращение к такому диску будет довольно медленным. Для формата с фиксированной длиной требуется хранение таблицы размещения файлов прямо на диске. Обновления файлов потребуют внесения изменения в таблицу. Каждый раз изменения записывать нельзя, поскольку число перезаписей CD-RW все-таки ограничено (порядка 1000 раз). Поэтому в процессе работы таблица размещения файлов хранится в памяти компьютера, и только когда носитель извлекают, предварительно она записывается на CD-RW. Из этого следует уязвимость

данной системы: авария во время записи до изъятия носителя приведет к несоответствию между таблицей, записанной на носителе, и реальным расположением файлов (то есть к потере доступа к данным). Диск с пакетами фиксированной длины не может быть закрыт как ISO 9660 и поэтому стандартными средствами не читается. С ним можно работать лишь на приводе MultiRead или рекордере CD-RW через специальный драйвер (например, UDF Reader).

Международные стандарты ISO/IEC 13346 и ISO/IEC 13490 и европейские ECMA (European Computer Manufacturers Association) — ECMA-167 и ECMA-168 (1994 г.) призваны заменить ISO-9660 (ECMA-119). Здесь снимаются некоторые ограничения, свойственные прежним стандартам.

ECMA-168 (Volume and File Structure ofRead-Only and Write-Once Compact Disk Media for Information Interchange) описывает структуру файлов и томов на дисках CD-ROM и CD-RW. Спецификация предусматривает любой метод записи — сплошной (сессия или диск за один заход), потрековый и пакетный. Диск ECMA-168 более удобен для хранения данных, чем ISO 9660. Стандарт вобрал возможности Rock Ridge: длинные имена, глубокую вложенность каталогов, а также расширенный набор символов в именах. Хотя прямой совместимости с ISO 9660 нет, возможно создание «компромиссных» дисков со структурами записей в обоих форматах.

DirectCD — приложение Windows, выпущенное фирмой Adaptec для облегчения работы с CD-рекордером. При запущенном приложении DirectCD с диском в устройстве CD-RW можно работать так же, как с обычным диском: создавать, копировать, изменять, удалять, переименовывать файлы (только скорость операций записи ниже, чем у винчестера). Диск форматируется пакетами фиксированной длины. Приложение позволяет работать и с CD-R, но, естественно, с дополнительными ограничениями.

7.3.5. Разновидности компакт-дисков

В настоящее время встречается множество разновидностей компакт-дисков, большинство из которых при определенных условиях может использоваться в ПК:

- *PC CD-ROM* — диски с файлами данных для IBM PC-совместимых компьютеров. Файловая система, как правило, ISO 9660, для длинных имен Windows используется расширение Joliet.
- *MAC CD-ROM* — диски с файлами данных для компьютеров Macintosh. Файловая система HFS.
- *CD-R и CD-RW*— болванки для записи и перезаписи на CD-рекордере; логическое содержание определяется записанной программой.
- *CD-DA* (Digital Audio) — традиционный стандарт аудиозаписи (до 99 треков, до 74 минут, до 99 точек входа на каждый трек), подробнее см. в п. 7.3.3.1. Аудио-диски CD-DA имеют логотип «Compact Disc Digital Audio».
- *CD-Text* — формат, предложенный Philips для записи текстовой информации на аудиодиски (например, автор, название произведения и т. п.). Текст записывается в субканалах R ...W. Плейеры, не поддерживающие текст, просто воспроизводят аудиотреки. Специальные плейеры имеют алфавитно-цифровой дисплей (2 строки по 20 знаков или 21 строку по 40 знакомест), предусматривается выбор информации для просмотра и произведений для прослушивания с помощью меню. Приводы CD-ROM из данного формата воспринимают только аудиоинформацию (не поддерживают декодирование дополнительных субканалов).
- *CD-Graphics* (CD-G, CD+G) — формат для записи графики (текста) в субканалах R ...W. Графика записывается с разрешением 288×24, 2 цвета (линейная графика); 288×192, 16 цветов (телевизионная графика, CD-G) или 288×192, 256 цветов (расширенная телевизионная графика, CD-EG). Применяется в проигрывателях «Караоке».
- *PhotoCD* — хранение изображений в формате Kodak. Использует физический стандарт CD-ROM XA и спецификации многосеансовых записываемых (CD-R) дисков. Изображения хранятся в форматах, соответствующих применению:
  - *Photo CD Master* — до 100 кадров с разрешением 2048×3072, для бытовой 35-мм фотопленки;
  - *Pro Photo CD Master* — 25-100 кадров с разрешением 4096×6144, для профессиональной работы;
  - *Photo CD Portfolio* — до 700 кадров с разрешением 512×768 или 1024×1536, для интерактивных презентаций;
  - *Photo CD Catalogue* — до 6000 кадров с разрешением 512×768, для хранения каталогов;
  - *Print Photo CD* — до 100 кадров с разрешением 2048×3072, для полиграфической печати.

- *Video CD* — высококачественная цифровая видеозапись с MPEG-сжатием, видеоданные чередуются с аудиоданными. Спецификация описана в «Белой книге». Диск может содержать до 74 минут видеопотока, сжатого по MPEG-1, с разрешением 352×240, 30 к/с (NTSC) или 352×280, 25 к/с (PAL/SECAM), со стереозвуковым сопровождением. Неподвижные изображения могут иметь разрешение до 720×480/576 (могут использоваться для меню). На каждом треке может быть расставлено до 98 точек входа, на всем диске — до 500. Диски Интерактивны: с помощью меню можно выбирать точки входа (начала воспроизведения), формировать последовательность исполнения

(playlist), быстро «перематывать» вперед и назад, смотреть и скрывать субтитры. Диск содержит не менее двух треков. На первом треке записана прикладная программа проигрывания, списки точек входа, статические изображения и некоторые вспомогательные данные. На втором и далее треках размещаются видеоданные. Дополнительно могут присутствовать и аудиотреки CD-DA. Использует физический стандарт CD-ROM XA. Диски предназначены для Video CD-плейеров, но могут воспроизводиться и приводом CD-ROM на ПК с MPEG-декодером. Видеодиски Video *CD* имеют логотип «Compact Disc Digital Video» и (или) надпись «VideoCD». Заметим, что диски CD-Video — это совсем другие диски, на которых записана аналоговая видеоинформация, и к устройствам чтения-записи, применяемым в ПК, эти диски не имеют никакого отношения.

- *Super Video CD* (SVCD) — видеодиски с более высоким качеством изображения (напоминают DVD, но имеют меньшую емкость). Разрешение 480×480/576, сжатие по MPEG-2 (средняя скорость потока 2,6 Мбит/с), 2 стерео- или 4 монофонических канала, 37 минут видео при максимальной скорости потока, может воспроизводиться на приводе со скоростью не ниже 2×.
- *CD Plus,* они же *CD Extra* — мультимедийные диски, содержащие две сессии — первую аудио (до 98 треков, воспроизводима стандартным аудиоплейером) и вторую в формате CD-ROM XA. Называется также Enhanced Music CD.
- *CD-I* — интерактивные диски с видео, аудио и другими данными, предназначенные для воспроизведения на специальном плейере. Файловая система отличается от ISO 9660. Подробнее см. п. 6.3.3. Диски CD-I имеют логотип «Compact Disc Interactive».
- *CD-Bridge* или *CD-I Bridge* — диски, соответствующие «Белой книге»: формат сектора CD-ROM XA Mode 2. Видеопрограммы могут проигрываться на плейере CD-I и CD-ROM XA. Файловая система ISO 9660. Подробнее см. п. 6.3.3.
- *Mixed-Mode Disc* — диск со смесью данных (трек 1) и аудиотреков (треки 2-99). На старых аудиоплейерах первый трек может ошибочно трактоваться как аудио и воспроизводиться как страшный шум, способный повредить акустические системы и уши. Из разных вариантов (CD-Extra или CD-Plus, Hidden Track и др.) более известны Enhanced CD. Чтение этих дисков в приводах CD-ROM поддерживается в Windows 9x.

7.3.6. Запись на оптические диски

Запись на оптический диск имеет свою специфику, связанную как с организацией диска (одна спиральная дорожка), так и с особенностями управления лазером. В отличие от магнитных и магнитооптических дисков, обеспечивающих произвольный доступ к любому сектору как по чтению, так и по записи, информация должна записываться непрерывным потоком в цепочку секторов. Поначалу рекордеры (записывающие устройства) могли записывать за одно включение записывающего лазера не менее, чем целый трек оптического диска. Следующим заходом можно было дописать последующий трек (треки), таким образом, по записи CD-R являются устройствами с последовательным доступом. С перезаписываемыми дисками CD-RW появился пакетный режим записи, который позволяет снять это ограничение, правда, ценой некоторых потерь пространства на диске и увеличения времени записи.

В течение всего времени записи, когда работает прожигающий лазер, на рекордер в требуемом темпе должна поступать записываемая информация. Опустошение буфера устройства (underrun) не допускается — в режиме записи устройство не может ждать. Прерывание процесса записи (приостановка потока данных), как правило, губит болванку. Для устройств и дисков CD-R возможны следующие режимы записи:

- Весь диск сразу (DAO — Disk At Once), когда лазер включается на время записи всего диска от начала до конца. В этом режиме записывается вся информация на диск, включая вводную и выводную зоны, и последующая запись на эту болванку уже невозможна (даже если остается место). Для записи в режиме DAO требуются чистые болванки. Диски, записанные в режиме DAO, будут читаться на любых приводах и могут быть использованы как мастер-диски для производства печатных (штампованых) CD. Режим DAO реализован не во всех рекордерах, он может не поддерживаться записывающим ПО (пакетом и драйверами).
- Сессия сразу (SAO — Session At Once), когда за одно включение лазера записываются все треки, вводная и выводная зоны одной сессии. Режим малораспространенный, используется для дисков CD-Extra.
- Потрековая запись (TAO — Track At Once), при которой лазер включается на время записи одного трека. В начале каждого трека записывается предзазор (pre-gap) длительностью 2 секунды (150 секторов). Этот режим применяется как для односеансовой, так и для многосеансовой записи. Режим пригоден для дисков любого назначения (аудио, CD-ROM и т. п.). Нормально записанные диски будут читаться на любых приводах. В этом режиме сначала на диск пишутся информационные треки, а вводная зона остается свободной. Координаты начала треков, а также координаты начала свободной области, следующей за последним уже записанным треком, временно сохраняются в служебной области болванки (PMA). Вводная и выводная зоны записываются позже — при закрытии сессии. До

закрытия сессии (в ISO 9660 включающей запись логической TOC и таблицы путей) записанные данные для обычных приводов CD-ROM остаются недоступными.

- Пакетная запись (packet writing), позволяющая за одно включение лазера записывать произвольное количество блоков — пакет. Длина пакета не превышает объема буфера рекордера, благодаря чему опустошение буфера при записи не грозит порчей диска. Лазер включается на запись, только если в буфере уже имеется полный пакет. Между пакетами записывается всего 7 промежуточных блоков (2 run-out, 4 run-in, 1 link). Пакеты могут быть фиксированной или переменной длины (см. п. 6.3.3). Пакетную запись ввели на CD-RW, благодаря ей появилась возможность прямого доступа по записи к отдельным блокам диска (при пакетах фиксированной длины). Пакетная запись поддерживается не всеми рекордерами. Диски, записанные в пакетном режиме, читаются не всеми приводами CD-ROM (у них возникают проблемы с чтением промежуточных блоков). Для аудиодисков пакетная запись непригодна. Для чтения пакетно-записанного диска требуется драйвер файловой системы UDF. Сессия (диск) с пакетами переменной длины может быть закрыта и в формате ISO 9660 Level 3, тогда она будет читаться и с помощью редиректора (типа MSCDEX), поддерживающего Level 3 (старые MSCDEX поддерживают только Level 1, без чередования и фрагментации файлов).

Стирание диска предполагается только для CD-RW — при стирании вся стираемая область переходит в одно состояние (фазу). Стирание может быть полным (full erase) или быстрым (quick erase). При полном стирании выполняется «зачистка» всего диска, включая и информацию TOC во вводной зоне. Быстрое стирание очищает лишь отдельные области диска. Диск со стертой TOC будет выглядеть пустым, но при этом он может содержать информацию (до которой очень трудно добраться). Быстрое стирание, затрагивающее лишь структуры данных томов, используют на дисках с пакетной записью (например, в DirectCD). Диск может быть настолько испорчен, что рекордер не сможет выполнить и стирание. В этом случае может помочь стирание солнечным светом или ультрафиолетовыми лучами (в устройстве для стирания ультрафиолетовых ПЗУ).

Для дисков CD-R штатного стирания (с целью последующего использования диска) не предусмотрено; однако это не означает, что записанный диск CD-R невозможно стереть (то есть испортить), используя стандартный рекордер (может потребоваться лишь нестандартная программа).

За одну *сессию* на диске должна быть сформирована стандартная структура, включающая как треки с данными (программная область), так и служебные зоны lead-in и lead-out. Как было сказано выше, в Q-субканале вводной зоны содержится таблица содержимого диска TOC, в которой описаны абсолютные координаты начала всех треков, а также выводной зоны.

При записи очередной сессии многосеансового диска данные предыдущих сессий можно (выборочно) включать в оглавления (на уровне треков — в TOC вводной зоны или на уровне файлов в дескрипторах тома). Тогда для считывателя многосеансовый диск будет выглядеть как единое целое, а запись очередной сессии может изменить его видимое содержание. Напомним, что последняя сессия станет доступной для чтения только после ее закрытия (finalize). Закрывать диск в принципе необязательно. После закрытия диска записать на него новые сессии уже невозможно, и, следовательно, невозможным станет «изменение» его файлов. Первая сессия «съедает» 20 Мбайт, каждая новая сессия приводит к потере 13,5 Мбайт емкости диска (накладные расходы на вводную и выводную зоны), так что записывать множество мелких сессий невыгодно.

Закрытием сессии называется процесс записи вводной зоны со сформированной таблицей TOC, а также выводной зоны. До закрытия сессии стандартные устройства чтения не располагают информацией о координатах начала треков — эта информация временно сохраняется рекордером в специально отведенной зоне PMA, не входящей в стандартную область, доступную для записи (650 Мбайт). Про эту область «знает» только записывающее ПО, и читают ее только рекордеры. Незакрытая сессия недоступна никаким устройствам чтения в «штатном» режиме. Рекордер может дописывать в незакрытую сессию треки до тех пор, пока на диске есть доступное место, пока не будет достигнуто предельное число треков в сессии (99) и пока есть место в PMA для временного хранения координат начала трека. После закрытия к сессии уже не могут быть добавлены треки, но может быть открыта новая сессия, если не закрыт диск.

Закрытием диска (финализацией) называют запись вводной и выводной области, причем в TOC указывается начало выводной дорожки (а не начало возможной вводной для последующей сессии). После закрытия диска к нему уже не могут быть добавлены сессии (и треки). Закрытие диска в принципе необязательно, достаточно закрытия сессии.

Когда закрывается сессия или диск с файловой системой ISO 9660, кроме «физической» TOC, описывающей положения треков, на диск (в программную область) записывается и «логическая» TOC тома, в которой описывается положение всех записанных файлов (при желании включая и файлы предыдущих сессий). При этом в области уже записанных файлов никаких изменений не производится. После такого закрытия диск становится читаемым стандартным приводом CD-ROM (для CD-RW требуется MultiRead CD-ROM) со стандартными драйверами (MSCDEX для DOS, встроенные средства Windows 9x/NT). До этого закрытия логическая TOC существует лишь в памяти (на жестком диске) пишущего компьютера, а физическая TOC — в PMA на записываемом носителе. Если диск вынуть из

рекордера до закрытия (или в случае аварии), логическая TOC не попадет на диск. Данные на диске остаются, но доступа к ним нет.

Форматирование диска UDF с пакетами фиксированной длины действительно форматирует (записывает) весь диск и к тому же закрывает его (физически, записывая вводную и выводную зоны и делая необходимые ссылки в TOC Q-субканала вводной зоны). При этом на диск можно записывать (удалять, переименовывать...) файлы. Диск кроме рекордера будет читаем в MultiRead CD-ROM с драйверами UDF; но традиционным читающим средствам он недоступен.

«Форматирование» диска UDF с пакетами переменной длины на самом деле только «наводит» каталог на свободную область сессии. На диск можно записывать (только дописывать) файлы, до тех пор пока не закрыта сессия или диск. После закрытия сессии можно будет открыть новую и продолжать запись, после закрытия диска — уже нет. На считывающих приводах данные будут читаться лишь для закрытой сессии (диска), открытую сессию может читать только рекордер (он пользуется PMA). Сессия может быть закрыта в формате ISO, тогда диск будет читаем всеми традиционными средствами.

Хотя средства записи и позволяют смешивать форматы Mode 1 и Mode 2, при считывании таких дисков наверняка возникнут проблемы. Если есть необходимость сочетать треки с «сырыми» (без ECC) и защищенными от ошибок данными, то должен использоваться режим Mode 2, в котором возможны две соответствующие формы.

Чистые болванки на самом деле не совсем пустые. На их поликарбонатном субстрате отпечатана спиральная дорожка, содержащая временную разметку. Эту разметку «понимает» только рекордер. На этой же дорожке отпечатана информация о носителе, которая может не полностью соответствовать действительности (штампом могут пользоваться разные производители):

- Manufacturer — производитель матрицы (штампа), но не обязательно диска.
- Writable/Rewritable — тип болванки (CD-R или CD-RW).
- Dye type — тип краски (для CD-R), информация для настройки записывающего лазера. Однако краска может быть иной (см. выше), а для настройки все равно используется область PCA.
- Spiral length in blocks — длина спирали (количество блоков, доступных для записи). Соответствует действительности, так как определяется только штампом.
- Rated speed — допустимая скорость записи. Если не указана, то для CD-R допустима скорость 1×, для CD-RW — 2×. Превышение скорости чревато порчей диска.
- Audio — болванка может использоваться и на автономном рекордере аудиодисков (более высокое качество).

Практически все модели современных рекордеров CD позволяют работать с болванками обоих типов — и CD-R, и CD-RW. Выбор типа носителей делают с учетом назначения записи (передача информации, архивация данных с необходимостью сохранения предыстории и без, другие задачи). Перезаписываемые болванки дороже, но они дают право на «бесплатную» ошибку.

По способу подготовки данных для записи различают *запись с образа CD* и *запись «налету»*. Более надежен способ с предварительным созданием образа CD (CD Image или Virtual CD — виртуальный CD, не путать с Video CD). При этом вся информация для записываемого диска должна быть предварительно сформирована в виде файла-образа на каком-либо носителе (винчестере). При записи образ считывается и передается на рекордер с требуемой скоростью, не допуская опустошения буфера. Для хранения образа требуется наличие свободного дискового пространства (до 650 Мбайт при записи целого диска). В качестве носителя образа могут работать отнюдь не все магнитные диски: если во время записи винчестер вдруг займется непрерываемой внутренней термокалибровкой, болванка будет загублена. Запись «на лету» (on-a-fly) не требует резервирования большого объема внешней памяти для хранения образа — файлы считываются с мест своего обычного хранения, но также должна быть гарантирована скорость и непрерывность считывания.

Чтобы застраховаться от опустошения буфера рекордера, компьютер, предназначенный для записи, должным образом конфигурируют:

- Компьютер должен быть достаточно мощным (быстродействующий процессор, большой объем ОЗУ, быстрые диски).
- Предпочтительный интерфейс рекордера и дисков — SCSI-2 (SCSI-3), имеющий более высокую производительность, чем ATA при одновременной работе с несколькими устройствами (в данном случае работают по крайней мере два устройства — винчестер и рекордер).
- Если используется интерфейс ATA, то винчестер с образом и рекордер следует устанавливать на разных каналах ATA.
- Предпочтительный режим работы драйверов — прямое управление шиной (bus mastering).
- Для хранения образа желательно иметь отдельный раздел жесткого диска (а то и отдельный винчестер), который следует регулярно дефрагментировать (не во время записи!).
- На время записи компьютер не должен играть роль сервера сети (его диски и принтеры не должны быть разделяемы), поскольку неожиданный приход запроса внешнего пользователя может

загрузить компьютер так, что поток данных на рекордер приостановится. Если имеется модем, для его ПО должна быть запрещена реакция на звонки.

- На компьютере на время записи должен быть запрещен автоматический запуск приложений по расписанию (ScanDisk, Defrag, антивирусные программы) — их внезапный запуск тоже может слишком загрузить компьютер.
- На компьютере на время записи должен быть отключен «хранитель экрана» (screensaver).
- На компьютере должно быть отключено автоматическое распознавание диска CD-ROM, иначе во время записи система неожиданно «увидит» новый диск и попытается его «проиграть», что почти наверняка прервет процесс записи.
- Средства управления энергопотреблением рекомендуется отключить, чтобы компьютер случайно не «заснул» в процессе записи («заснуть» может и привод CD-ROM, с которого делают копию).
- На время записи не следует запускать лишних приложений, особенно ресурсоемких. Не стоит также прослушивать аудиодиск (если имеется и привод CD-ROM) — хотя при правильной настройке этот процесс потребляет мало ресурсов, в случае ошибки чтения (или иных нештатных ситуаций) система может быть на некоторое время заблокирована.
- Предпочтительны модели рекордеров с большим объемом буфера. Чем больше объем буфера для записи имеет устройство CD-R, тем оно менее чувствительно к неравномерности входного потока данных.
- Скорость записи должна выбираться исходя из качества болванок и производительности компьютера. Чем выше скорость записи, тем выше требования к скорости подачи входного потока данных. Запись на пониженной скорости может оказаться более качественной (будет меньше проблем считывания на разных приводах).

Для защиты от порчи болванок в современных рекордерах применяют технологию BURN-Proof (Buffer UnderRuN Proof — проверка буфера на опустошение) — очередная порция записи не начинается, если в буфере недостаточно информации. При большом размере буфера (2-4 Мбайт) эта технология работает довольно надежно.

Приложения, (пере)записывающие CD, часто имеют функцию тестирования, при которой имитируется весь процесс записи диска, но без включения лазера. Таким образом удается проверить, все ли компоненты будущего диска находятся на своих местах и доступны (особенно важно при записи «на лету», а не с образа) и достаточна ли скорость подачи данных на рекордер. Однако прохождение теста не гарантирует успех последующей реальной записи, если вмешается один из внезапных факторов, перечисленных выше. Поскольку в тестовом режиме TOC не записывается, то не сработает автоматическое распознавание диска; а если оно не было отключено, то при реальной записи оно и сработает, скорее всего, некстати.

7.3.7. Системная поддержка CD-ROM

Поскольку CD-ROM по организации данных (файловой системе) существенно отличается от традиционных дисков (гибких и жестких), для обеспечения «прозрачного» доступа приложений к файлам на CD-ROM требуются специальные программные средства. Несмотря на возможность загрузки ОС с CD-ROM, реализованную в современных версиях BIOS, полной поддержки доступа к CD-ROM (такой, как к обычным дискам) BIOS не предоставляет. Приложениям доступ к CD-ROM обеспечивают только сервисы операционной системы, встроенные в ОС или загружаемые. Для MS DOS имеется утилита MSCDEX.EXE (Microsoft DOS extensions for CD-ROM), загружающая резидентный драйвер эмуляции логического диска DOS с файловой системой FAT16 при чтении CD-ROM с томами формата ISO 9660 или HSF. Кроме обычных дисковых операций драйвер MSCDEX обеспечивает и некоторые дополнительные функции, в том числе и воспроизведение аудиотреков с указанной позиции. Перед загрузкой утилиты MSCDEX должен быть успешно загружен драйвер установленного привода CD-ROM. Драйверы большинства приводов с интерфейсом ATAPI (IDE) между собой взаимосовместимы; для приводов SCSI драйвер привязывается к модели контроллера SCSI. Каждому обнаруженному приводу драйвер назначает сигнатуру (например, MSCD000). Загрузка MSCDEX связывает обнаруженные приводы с буквой логического имени диска, по умолчанию начиная с первой свободной после логических дисков винчестеров и до последней доступной. Если свободных букв не хватает, в списке доступных дисков CD-ROM не появится. Помочь в этом случае может строка LASTDRIVE=x в файле CONFIG.SYS. Возможно и принудительное назначение буквы ключом /L:x. Успешно подключенный CD-ROM будет выглядеть как логический диск DOS со всеми его атрибутами (см. п.2), и согласно правилам FAT16 размер кластера у него будет 16 Кбайт. К поддержке CD-ROM в среде MS DOS относятся строки файла CONFIG.SYS:

DEV1CE=<имя_драйвера>.SYS [/D:MSCD000] (ключ /D: задает сигнатуру)

LASTDRIVE=x (необязательная строка, x - последняя буква, доступная для дисков, в том числе и CD-ROM: по умолчанию - «E»)

В файле AUTOEXEC.BAT должна присутствовать строка:

[<путь>]MSCDEX /D:MSCD000 [/L:x] (ключ /D: задает сигнатуру, совпадающую с заданной при загрузке драйвера в CONFIG.SYS)

Для ОС Windows 9x и ряда других, изначально спроектированных с учетом поддержки CD-ROM, расширение типа MSCDEX не требуется, но, естественно, сам привод должен быть поддерживаем системным драйвером. Для других ОС, еще «не знающих» CD-ROM, существуют расширения, аналогичные MSCDEX.

В Windows 9x/NT доступ к параметрам и свойствам приводов CD-ROM осуществляется через Панель управления. Здесь можно задавать режимы обмена, логическое имя (букву) диска, а также автоматическое распознавание диска AIN (Auto Insert Notification — автоматическое уведомление системы об установке носителя). На автоматическом распознавании базируется Autoplay — автоматический запуск проигрывателя при загрузке аудиодиска, а также Autorun — автозапуск приложения (программы), указанного в файле AUTORUN.INF. Когда эти функции разрешены, пользователю остается только вставить диск — все остальное делается автоматически, хочет он того или нет.

В файле AUTORUN.INF должны быть следующие строки:

[Autorun] (заголовок)
open=runme.exe (имя запускаемого файла)
icon=someicon.ico (имя файла-значка данного диска, необязательная строка)

В файле могут быть описаны и довольно сложные конструкции, позволяющие организовывать меню запуска и прочие вещи, рассказ о которых выходит за рамки данной книги.

7.3.8. Загружаемые CD-ROM

Для обеспечения возможности загрузки ОС с CD-ROM фирмы Phoenix и IBM выпустили спецификацию «El Torito Bootable CD-ROM Format Specification», версия 1.0 была опубликована в январе 1995 г. Цель спецификации — обеспечить возможность загрузки ОС и приложений с CD-ROM средствами BIOS (на «голой» машине). При этом имеются следующие возможности:

- Загрузка ОС по выбору из загрузочного каталога (Boot Catalog), находящегося на CD-ROM.
- Предоставление выбора конфигурирования CD-ROM в виде жесткого диска или дискеты.
- Переименование существующих приводов (если необходимо).
- Использование существующей технологии BIOS (доступ в режиме LBA) для обращения к кодам и данным.
- Совместимость с приложениями DOS и Windows, использующими функции *INT13h.*

Спецификация расширяет традиционный набор функций BIOS, она ориентирована на приводы с интерфейсом ATAPI и SCSI. В пункт выбора последовательности загрузочных устройств утилиты CMOS Setup должно быть введено новое устройство — CD-ROM и должен предлагаться выбор последовательности опроса устройств, например:

A:, CD-ROM, C:
CD-ROM, A:, C:
CD-ROM, C:, A:
C:, A:, CD-ROM
CD-ROM only

В опциях всегда должна быть возможность запрета загрузки с CD-ROM, поскольку формат начального сектора CD может быть произвольным (в зависимости от назначения диска), а загружаться можно только со специального CD-ROM. Загружаемый CD-ROM должен иметь *загрузочную запись, загрузочный каталог* и *файл-образ* дискеты 1.2, 1.44 или 2.88 Мбайт или жесткого диска. На одном CD-ROM может быть несколько образов; все они должны быть описаны в загрузочном каталоге.

При загрузке CD-ROM может стать диском A: или C:, в зависимости от типа образа.

При загрузке образа дискеты CD-ROM представляется в виде диска A:, при этом существующий первый дисковод получает имя B:. Если в системе два дисковода, то второй становится недоступным. При загрузке с образа жесткого диска CD-ROM замещает диск C: (номер устройства 80h), номера жестких дисков сдвигаются, они все остаются доступными. Для доступа приложений и ОС к данным, расположенным в области образа диска, могут использоваться вызовы *INT 13h* без необходимости загрузки каких-либо драйверов. Образ диска выглядит как обычный диск, но защищенный от записи. Для доступа к остальному содержимому CD-ROM должны загружаться драйверы, необходимые операционной системе — эти драйверы должны быть размещены в образе диска.

Функции начального загрузчика (bootstrap loader) *INT 19h,* выполняемого в конце POST, расширяются. Если ему предписана попытка загрузки с CD-ROM, то он читает загрузочную запись тома (boot record volume descriptor, сектор 17, см. табл. 7.2) и, если она присутствует, берет из нее ссылку на загрузочный каталог (booting catalog) CD-ROM, в котором должны быть ссылки на загрузочный образ. Возможны два варианта реализации загрузчика в BIOS: с единственным образом (single-image *INT 19h)* или со множеством образов (multiple-image *INT 19h).* «Единообразный» загрузчик обращается к образу, описанному в элементе каталога, предназначенном для загрузки по умолчанию (initial/default entry). «Многообразный» загрузчик позволяет выбрать требуемый образ из списка возможных, автоматически или запрашивая ответ пользователя.

Собственно диск CD-ROM может иметь разные варианты структуры: обычный (незагрузочный), загрузочный с одним образом, загрузочный с множеством образов, а также незагрузочный, но с образами, доступными через *INT 13h.* После загрузки ОС доступ ко всем образам, указанным в загрузочном каталоге, возможен через *INT 13h* как к обычным дискам, и если их файловая система поддерживается загруженной ОС, доступ к файлам и каталогом тоже будет обычным. Доступ к областям CD-ROM, не входящим в образы, на логическом уровне осуществляется через специальный драйвер. Доступ ко всем секторам CD-ROM возможен через функции 41-48h *INT 13h* в режиме LBA, размер блока 2048 байт.

Спецификация совместима с дисками ISO-9660. Согласно спецификаций El Torito, в секторе 17 (11h) от начала последней сессии должна находиться загрузочная запись (boot record, табл. 7.2), содержащая абсолютный указатель на загрузочный каталог (номер логического сектора от начала сессии). Загрузочный каталог (boot catalog) представляет собой набор 32-байтных описателей, упакованных по 64 в сектор CD-ROM. Размер каталога не ограничен, просмотром этого каталога при загрузке можно выбрать требуемый образ. Образы виртуализируются в приводы (диски), доступные по *INT 13h, с* номерами 00 или 80h для эмуляции загрузочных дисков, и с номерами, следующими за последним обнаруженным жестким диском, для эмуляции незагрузочных дисков. Образ может и не эмулировать диск, а просто являться программным кодом, загружаемым начальным загрузчиком в память (после чего ему передается управление).

В загрузочном каталоге определено 5 типов описателей (табл. 7.3):

- Validation entry — первый элемент каталога, удостоверяющий его наличие на CD-ROM. Если этот элемент не найден, то остальные и не рассматриваются.
- Initial/default entry — ссылка на загрузочный образ, в котором программы и драйверы могут пользоваться только функциями BIOS *INT 13h* (0-19h и, возможно, 40-48h). Этот образ используется «однообразным» загрузчиком BIOS.
- Section header — заголовок секции, предшествующий группе образов, с которых может быть загружен компьютер. В заголовке имеется идентификатор платформы (x86, Mac, Power PC), и если BIOS распознает свой идентификатор, она предлагает выбор образов из данной секции для загрузки (вместо образа по умолчанию). Этот выбор, к примеру, может использоваться для многоязычных продуктов.
- Section entry — элемент секции, который может быть выбран для загрузки. Выбор может выполняться даже автоматически, тип критерия выбора указывается в элементе. Критерии могут быть различными (их определяет разработчик BIOS). Для выбора по языку поле критерия должно начинаться с трехбайтного кода языка. Если размер поля критерия недостаточен, то возможно использование расширения элемента секции.
- Section entry extension — расширение элемента секции, определяющее дополнение критерия выбора. Расширение должно следовать непосредственно за расширяемым элементом. Возможна организация цепочки расширений, признаком наличия продолжения является единица в бите 5 первого байта, у последнего расширения бит 5=0.

Таблица 7.2. Формат загрузочной записи тома (boot record volume descriptor)

Смещение	Тип	Назначение
0	байт.	Идентификатор загрузочной записи (00h)
1	байт	Идентификатор платформы: 0 — 80x86,1 — Power PC, 2 — Mac
1-5	байты	Идентификатор ISO-9660 («CD001»)
6	байт	Версия дескриптора (1)
7-26h	байты	Идентификатор загрузчика (строка «EL TORITO SPECIFICATION», дополненная справа нулями)
27-46h	байт	Не используются(0)
47-4Ah	двойное слово	Абсолютный указатель на первый сектор загрузочного каталога
4B-7FFh	байт	Не используются(0)

Таблица 7.3. Структура элементов загрузочного каталога

Смещение	Тип	Назначение
Индикатор наличия каталога (validation entry)		
0	байт	Идентификатор заголовка, 01 h
1	байт	Идентификатор платформы: 0 — 80x86,1 — Power PC, 2 — Mac
2 - 3	слово	Резерв(0)
4 - 1Bh	символы	Идентификатор производителя CD-ROM (строка)
1C - 1Dh	слово	Контрольная сумма (сумма всех слов в элементе должна быть нулевой)
1Eh	байт	Ключевой байт (55П)
1Fh	байт	Ключевой байт (ААП)
Описатель образа по умолчанию (initial/default entry)		
0	байт	Индикатор загружаемости: 88h — загружаемый, 0 — нет
1	байт	Биты 0-3— тип загрузочного носителя: 0 — без эмуляции; 1 — дискета 1,2 Мбайт; 2 — дискета 1,44 Мбайт; 3 — дискета 2,88 Мбайт; 4 — жесткий диск
2-3	слово	Сегмент памяти, в который загружается образ. Если 0, то используется стандартный адрес сегмента — 7C0h.
4	байт	Код файловой системы образа (копия байта со смещением 4 из элемента таблицы разделов), см. табл. 4.2
5	байт	Не используется(0)
6-7	слово	Число виртуальных секторов (по 512 байт), которые должны быть загружены в указанный сегмент при начальной загрузке
8-0Bh	двойное слово	Адрес загрузочного блока (Load RBA) — относительный адрес начала образа виртуального диска (номер логического сектора CD-ROM)
0C-1Fh	байт	Не используется(0)
Заголовок секции (section header)		
0	байт	Индикатор заголовка: 90h — не последний заголовок секции, 91 h — последний заголовок секции
1	байт	Идентификатор платформы: 0 — 80x86,1 — Power PC, 2 — Mac
2-3	слово	Число элементов в секции
4-1Fh	символы	Идентификатор секции (строка). Загрузчик (BIOS INT 19h*)* просматривает элементы только знакомых ему секций
Элемент секции (section entry)		
0	байт	Индикатор загружаемости: 88h — загружаемый, 0 — нет
1	байт	Биты 0-3— тип загрузочного носителя: 0 — без эмуляции; 1 — дискета 1,2 Мбайт; 2 — дискета 1,44 Мбайт; 3 — дискета 2,88 Мбайт; 4 — жесткий диск
2-3	слово	Сегмент памяти, в который загружается образ. Если 0, то используется стандартный адрес сегмента — 7C0h.
4	байт	Код файловой системы образа (копия байта со смещением 4 из элемента таблицы разделов)
5	байт	Не используется(0)
6-7	слово	Число виртуальных секторов (по 512 байт), которые должны быть загружены в указанный сегмент при начальной загрузке
8-0Bh	двойное слово	Адрес загрузочного блока (Load RBA) — относительный адрес начала виртуального диска (номер логического сектора CD-ROM)
0C-1Fh	байт	Тип критерия выбора: 0—нет критерия; 1 — информация о языке и версии (IBM); 2-FFh — резерв
0D-1Fh	байты	Критерий выбора
Расширение элемента секции (section entry extension)		
0	байт	Индикатор расширения (44h)

1	байт	Биты 0-4 — не используются, Бит 5:1 — следует продолжение расширения, 0 — расширение последнее; Биты 6-7 — не используются
2-1Fh	байты	Критерий выбора

Загрузочный образ представляет собой посекторную копию эмулируемого диска, секторы копируются друг за другом в естественном порядке (по нарастанию номера LBA от 0 до конца диска). Эмулятор, встроенный в BIOS INT 13h, для традиционных вызовов (функции 0-19h) осуществляет пересчет виртуальных секторов (размером 512 байт, как у всех дисков) в 2048-байтные секторы CD-ROM. BIOS *INT 13h* отвечает за представление трехмерной (CHS) геометрии образа, по которой к нему могут обращаться программы, использующие этот интерфейс. При эмуляции дискет геометрия определяется типом: 1,2 Мбайт — 80×2×15, 1,44 Мбайт — 80×2×18, 2,88 Мбайт — 80×2×36. При эмуляции жесткого диска BIOS определяет геометрию из таблицы разделов образа, которая должна содержать корректное трехмерное описание начала и конца раздела. Спецификация требует, чтобы образ жесткого диска содержал только один раздел, описанный обязательно в первом элементе таблицы разделов. Когда работает эмуляция диска, его геометрию можно узнать по функции *INT 13h*(8); таблицей параметров для жесткого диска, на которую указывает вектор *INT 41h,* пользоваться нельзя. Функция *INT 13h(A8h)* формирует расширенную таблицу параметров диска, в которой эмулируемый диск описывается как ATAPI (в байте 10 бит 6=1). Это позволяет программам при необходимости распознать факт эмуляции.

При обращении к CD-ROM в процессе загрузки проверяется индикатор загружаемости выбранного образа (в загрузочном каталоге); если он отличен от 88h, то производится обращение к следующему устройству в загрузочной последовательности, заданной в CMOS Setup. Но при этом включается эмуляция дисков, и ОС, загруженная с другого устройства, получает доступ к содержимому образов через *INT 13h* без всяких драйверов.

К сервисам BIOS *INT 13h* добавляется несколько новых функций:

Функция 4Ah — начать эмуляцию диска (initiate disk emulation) *AL=0, DS:SI* указывают на пакет спецификации эмулируемого диска (specification packet, табл. 7.4). При возврате CF=0 означает, что диск с указанным номером является эмуляцией указанного образа и номера устройств изменились; CF=1 означает, что эмуляция не начата (код ошибки в AX*).*

Функция 4Bh — завершить эмуляцию диска (terminate disk emulation) и сообщить состояние. Завершение эмуляции приводит к сдвигу номеров устройств. При *AL=0* — сообщить состояние и завершить эмуляцию, *AL=1* — только сообщить состояние. В DL — номер устройства, для которого завершается эмуляция (DL=7Fh — для всех устройств). DS:SI указывают на пустой пакет спецификации. При возврате CF=0 означает завершение эмуляции; CF=1 означает, что эмуляции не было. В *AX* — код ошибки. *DS:SI* указывают на заполненный пакет спецификации.

Функция 4Ch — начать эмуляцию диска и выполнить загрузку (initiate disk emulation & boot) *AL=0, DS:SI* указывают на пакет спецификации (табл. 7.4). При возврате CF=1 означает невозможность загрузки (код ошибки в *AX); CF=0* возвратиться не может, поскольку в случае успеха (перезагрузки) возврат не предусматривается.

Функция 4Dh — прочитать секторы загрузочного каталога (return boot catalog):

AL=0, DL — номер устройства, DS:SI — указатель на командный пакет (табл. 7.5). При возврате CF=0 означает нормальное выполнение; CF=1, если устройство не является эмуляцией или командный пакет не вписывается в границы. В *AX* — код ошибки.

Функции 41-48h позволяют обращаться к любым логическим секторам CD-ROM (в режиме LBA с размером сектора 2048 байт), когда для данного привода включена эмуляция.

Таблица 7.4. Структура пакета спецификации эмулируемого диска

Смещение	Тип	Назначение
0	байт	Размер пакета (13h)
1	байт	Биты 0-3— тип загрузочного носителя: 0 — без эмуляции; 1 — дискета 1,2 Мбайт; 2 — дискета 1,44 Мбайт; 3 — дискета 2,88 Мбайт; 4 — жесткий диск. Биты 4-5 — резерв (0) Бит 6 — образ на приводе ATAPI, байты 8 и 9 относятся к IDE Бит 7 — образ на приводе SCSI, байты 8 и 9 относятся к SCSI
2	байт	Номер устройства; 0 —для эмуляции дискеты, 80h —для эмуляции загрузочного жесткого диска, 81-FFh —для эмуляции незагрузочного жесткого диска

3	байт	Индекс контроллера, указывает номер контроллера выбранного устройства
4	двойное слово	Логический адрес сектора CD-ROM, с которого начинается образ эмулируемого диска
8	слово	Спецификация устройства. Для SCSI байт 8 содержит номер логического (LUN) и физического (SCSI ID) устройства, байт 9 — номер шины. Для ATAPI бит 0 байта 8 определяет номер устройства: 0 — устройство 0 (ведущее), 1 — устройство 1 (ведомое)
A	слово	Сегмент буфера пользователя (если не нуль), используемого для кэширования чтения CD. Размер буфера 3 Кбайт
C	слово	Сегмент памяти, в который загружается образ. Если 0, то используется стандартный адрес сегмента — 7C0h
E-F	слово	Число виртуальных секторов (по 512 байт), которые должны быть загружены в указанный сегмент при начальной загрузке
10h	байт	Младшие биты числа цилиндров, совпадает со значением CH после возврата функции *INT 13h(8)*
11h	байт	Число секторов и старшие биты числа цилиндров, совпадает со значением CL после возврата функции *INT 13h(8)*
12h	байт	Число головок, совпадает со значением DH после возврата функции *INT 13h(8}*

Таблица 7.5. Структура командного пакета

Смещение	Тип	Назначение
0	байт	Размер пакета (8)
1	байт	Число секторов для чтения из загрузочного каталога
2	Двойное слово	Указатель на буфер для чтения
6	слово	Начальный номер сектора (при первом вызове функции 4D должен быть нулевым)

7.3.9. Оптические диски с прямым доступом

Кроме CD-ROM, CD-R и CD-RW существуют не столь широко распространенные оптические диски с прямым доступом.

PD/CD — комбинированный накопитель, записывающий информацию на специальный носитель (и с него же считывающий) по методу изменения фазы вещества (Phase change Disk). Носитель представляет собой многослойный диск в защитном картридже, у которого в одном из слоев может изменяться фаза состояния (как и в CD-RW, но в PD этот принцип применили раньше). Считывание основано на изменении степени отражения участков с разной фазой состояния. В отличие от CD с одним спиральным треком, PD имеет концентрические треки (как у магнитных дисков) и, следовательно, произвольный метод доступа. Шпиндель накопителя поддерживает постоянную угловую скорость для каждой зоны треков (а не постоянную линейную, как в CD). Это позволяет снизить время доступа при поиске в пределах зоны, поскольку не тратится время на разгон или торможение диска. Емкость PD, как и у CD, составляет 650 Мб, но PD не может быть считан на накопителе CD-ROM. В то же время устройство PD/CD (например, модель PD 650) считывает и обычные CD — тип установленного носителя определяется автоматически. Большим преимуществом PD перед CD-R является возможность многократных циклов стирания-записи и, естественно, считывания, при прямом доступе к данным. Недостаток — несовместимость PD и CD.

WORM (Write Once, Read Many times) — устройства с однократной записью и многократным считыванием специфического носителя. Устойчивый к внешним воздействиям 5" картридж емкостью 650 Мбайт - 1,3 Гбайт записывают по технологии, похожей на CD-WORM. Стоимость устройств высокая, стандартов нет.

Термин *WARM* (Write And Read Many times) подразумевает многократную запись и считывание, но стандартизованных оптических устройств данного типа пока нет.

Перечисленные устройства не выдержали конкуренции с CD и DVD.

7.3.10. Диски DVD

Название *DVD* поначалу расшифровывалось как Digital Video Disk — диск для цифровой видеозаписи, сейчас же подразумевается иное — Digital Versatile Disk — универсальный цифровой диск. DVD представляют собой развитие принципов CD, направленное на повышение плотности хранения и скорости передачи информации. Эти диски имеют те же внешние размеры, что и CD (диаметр 120 мм и толщину 1,2 мм), однако представляют собой «бутерброд» из двух пластин. Для повышения емкости ширина трека и продольный размер битовой ячейки уменьшены примерно вдвое, снижены издержки

избыточности кодов коррекции ошибок. Кроме того, могут использоваться две стороны диска, а на каждой стороне информация может храниться в двух слоях, таким образом, один Диск может иметь уже четыре рабочих плоскости. Для считывания DVD требуется лазер с длиной волны 635/650 нм (для CD используется длина волны 780 им). Изменена система канального кодирования (применяются 16-битные коды) и система избыточного кодирования.

Каждая пластина DVD может быть как однослойной (по конструкции аналогична диску CD), так и двухслойной. В двухслойной пластине «ямки» расположены в двух плоскостях, нижний слой сверху покрыт полупрозрачной/полуотражающей пленкой, а верхний — отражающей (см. рис. 7.2). Какой из слоев считывается, определяется фокусировкой луча (сигналы другого слоя из-за расфокусировки размываются и на их фоне требуемый слой различим).

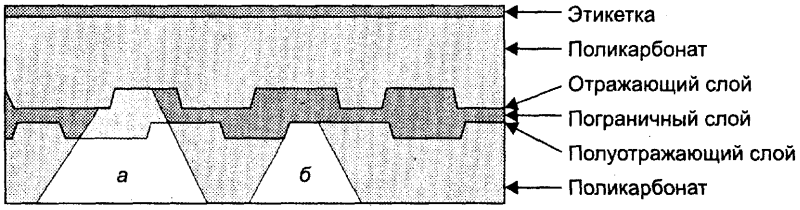

Рис. 7.2. Устройство двухслойного диска DVD: а — луч, сфокусированный на верхнем слое; б — на нижнем

Базовой скоростью DVD является скорость, достаточная для считывания видеодисков — около 11 Мбайт/с, что эквивалентно 9× CD-ROM. Однако эта скорость передачи обеспечивается при скорости вращения DVD примерно в 3 раза медленнее, чем CD-ROM. Этим объясняется тот факт, что высокоскоростные приводы DVD-ROM, читающие и CD, обеспечивают скорость считывания CD меньшую, чем DVD, — сказывается ограничение скорости вращения, при которой биения еще приемлемы. Высокие скорости DVD-ROM (2×, 4×...) требуются лишь для ускорения обмена данными при работе компьютерных приложений — для видео достаточно и 1×. В 2000 г. достигли скорости 16×, но такая высокая скорость пока остается невостребованной. Время доступа у приводов DVD такое же большое, как и у приводов CD (около 200 мс).

Приводы DVD позволяют считывать информацию и с обычных CD. Уже появились комбайны, способные читать и писать диски CD и читать диски DVD, например привод Toshiba SD-R1002 сочетает в себе возможности CD-R 4×, CD-RW 4×, CD-ROM 24× и DVD-ROM 4,8×.

Диски DVD выпускаются в разных сочетаниях количества сторон и слоев (SS — Single Sided, односторонние; DS — Dual Sided, двусторонние; SL — Single Layer, однослойные; DL — Dual Layer, двухслойные).

- DVD-5 — диски емкостью 4,7 Гбайт (SS/SL), информация записана в одном слое одной пластины. Вторая пластина может использоваться для этикетки.
- DVD-10 — диски емкостью 9,4 Гбайт (DS/SL), информация записана в одном слое обеих пластин. Для считывания обеих сторон во многих моделях приводов приходится переворачивать диск. Маркировка дисков проблематична, поскольку обе стороны используются для считывания.
- DVD-9 — диски объемом 8,54 Гбайт (SS/DL), используются два слоя одной стороны. Из-за двухслойности размер ячейки увеличили примерно на 10%, за счет чего емкость каждого слоя несколько снизилась.
- DVD-18 — диски объемом 17,08 Гбайт (DS/DL), используются два слоя и две стороны. Производство таких дисков очень сложно, а потому и дорого.
- DVD-R — записываемые диски, однослойные, односторонние (4,7 Гбайт) и двусторонние (9,4 Гбайт).
- DVD-RAM — перезаписываемые (Rewritable) диски, однослойные, односторонние (SS/SL, 2,6 Гбайт) и двусторонние (DS/SL, 5,2 Гбайт).
- DVD-RW — перезаписываемые (ReRecordable) диски, однослойные, односторонние (SS/SL, 4,7 Гбайт).

Диски *DVD-Video* несут видеоинформацию с разрешением-720x480/576, сжатие по MPEG-2 (средняя скорость потока 3,5 Мбит/с), аудиосигнал по схеме 5.1 (Dolby Digital Sound), 133 минуты.

Диски *DVD-Audio* несут высококачественный многоканальный (объемный) аудиосигнал с новыми возможностями, недоступными в CD-DA. Полный диск (4,7 Гбайт) вмещает 2 часа объемного или 4 часа стереофонического воспроизведения.

Диски *DVD-ROM, DVD-RAM, DVD-RW* и *DVD+RW* используются для хранения и переноса больших объемов информации, они должны заменить CD в тех случаях, когда объема 650 Мбайт уже недостаточно (многотомные игры, энциклопедии, базы данных, архивы).

Первые записываемые диски DVD-R и рекордеры появились в 1997 г., но были очень дороги. Они используют ту же технологию, что и CD-R, но более тонкую геометрию и иной тип краски. Диски DVD-R полностью совместимы с DVD-ROM, DVD-Video и DVD-Audio, так что они могут считываться на любых устройствах, подходящих по назначению (проигрывателях или приводах для компьютеров). Диск DVD-R содержит 5 областей:

- PCA (Power Calibration Area) — область калибровки мощности;
- RMA (Recording Management Area) — область управления записями;
- lead-in area — вводная область;
- data recordable area — область данных;
- •lead-out area — выводная область.

Диски DVD-R допускают инкрементную запись, аналогично многосеансовым CD-R. Определены два типа инкрементной записи:

- Type 1 — допускает чтение дисков DVD-R в системах с ISO 9660, использует файловую систему UDF Bridge.
- Type 2 — для непосредственного манипулирования файлами (прямого доступа), использует UDF без возможностей ISO 9660.

Для обоих типов каждая секция данных пишется в обрамленную область (bordered area) после области border in, за которой следует область border out и border in, предшествующая следующей секции данных. Каждая обрамленная область начинается с файловой системы UDF и завершается виртуальной таблицей размещения файлов VAT (Virtual Allocation Table).

В первой версии DVD-R (1997 г.) емкость диска была 3,95 Гбайт, во второй версии (1999 г.) емкость увеличена до 4,7 Гбайт.

Диски и рекордеры по спецификации DVD-форума *DVD-RAM* (перезаписываемые) появились позже DVD-R, но по более низким ценам. В них используется изменение фазы (аморфная-поликристаллическая), как и в CD-RW. В первой версии DVD-RAM емкость диска была 2,6 Гбайт на одну сторону, во второй версии (1999 г.) емкость увеличена до 4,7 Гбайт. Вместо постоянной линейной скорости здесь используется разделение диска на зоны, и в каждой зоне постоянна угловая скорость (CAV — Constant Angular Velocity), так что средняя линейная скорость по всему диску постоянна. Диски DVD-RAM заключены в картриджи, и для работы с ними требуются специальные приводы (Caddy-Type).

Сейчас DVD-форумом рассматриваются и иные форматы перезаписываемых дисков - DVD-RW и DVD+RW (это разные вещи!).

Формат *DVD+RW* предлагается фирмами Hewlett-Packard, Philips и Sony при поддержке Verbatim, Ricoh и Yamaha (ветеранов CD-RW). Диски совместимы с DVD-ROM и не требуют картриджа. Доступны диски объемом 3 Гбайт и прорабатывается формат на 4,7 Гбайт. Для хранения данных применяется постоянная угловая скорость, для видео — постоянная линейная.

Формат *DVD-RW* предлагается фирмой Pioneer и принят DVD-форумом для рассмотрения в качестве очередного стандарта. У дисков DVD-RW отражающая способность выше, чем у DVD-RAM и DVD+RW, так что они могут считываться и на существующих приводах DVD-ROM. Объем диска 4,7 Гбайт, картридж не требуется.

Есть еще и формат перезаписываемых дисков *MMVF* (Multimedia Video File) от фирмы NEC, предлагающей диски 5,2 Гбайт, удобные для видеоданных.

Для защиты от копирования и неавторизованного использования высококачественной видео- и аудиопродукции используется скремблирование содержимого (CSS — Content Scrambling System). Видео- и аудиофайлы шифруются по ключам, уникальным для каждого диска и каждого набора произведений, записанных на диске. Ключ в зашифрованном виде хранится на диске и считывается декодером воспроизводящего устройства. Алгоритмы шифрования держатся в секрете компаниями, занимающимися производством видеодисков и проигрывающей аппаратуры. Файлы данных не шифруются. На DVD-диске может стоять защита файлов от перезаписи через аналоговый выход композитного видеосигнала Macro-vision APS (Analogue Protection System) — в сигнал вводятся специальные искажения. На аналоговые выходы RGB и YUV эта защита пока не распространяется. Для установки флага защиты требуется лицензирование у Macrovision. Разработкой новых методов защиты от пиратства занимается рабочая группа CPTWG (Copy Protection Working Group) DVD-форума.

Диски DVD-Video имеют код региона, который должен совпадать с кодом региона, установленным в проигрывателе. Весь мир поделили на 6 зон (Россия относится к зоне 5), и каждый диск предназначен для продажи (проигрывания) лишь в своем регионе. В приводе DVD, устанавливаемом в компьютер, номер региона можно задавать программно. Возможность смены кода региона определяется уровнем контроля региональной защиты RPC (Region Protection Control), реализованном в микропрограмме привода. Привод RPC-1 позволяет менять код региона неограниченное число раз, RPC-2 — не более 5 раз. «Народные умельцы», естественно, уже нашли способ «лечения этого недуга» заменой «прошивки»,

однако найти версию микропрограммы (firmware) с RPC-1 можно не для каждой модели DVD.

В DVD используется файловая система микро-UDF — подмножество UDF (Universal Disk Format). Файловая система не зависит от платформы, обеспечивает эффективный файловый обмен, ориентирована на диски CD-ROM и CD-R, основана на стандарте ISO 13346. Имеется расширение UDF для поддержки перезаписываемых дисков. Комбинация UDF и ISO 9660, известная как UDF Bridge, позволяет обращаться к данным дисков как из ОС, не поддерживающих UDF (например, Windows 95), так и поддерживающих UDF (Windows 98). Диски DVD-видео и аудио используют только файлы в системе UDF, размер файла не должен превышать 1 Гбайт. Как для компьютерных, так и для телевизионных приложений диски DVD должны иметь единую файловую систему. Видео- и аудио файлы на дисках DVD должны находиться в каталогах VIDEO_TS и AUDIO_TS соответственно, расположенных в корневом каталоге диска.

Для воспроизведения DVD-видео в компьютере должен быть аппаратный или программный декодер MPEG-2. Аппаратный декодер может работать даже на маломощном процессоре типа Pentium 133, как правило, аппаратный декодер имеет и выход аналогового видеосигнала для подключения телевизора. Для программного декодирования требуется Pentium II-266 с соответствующим графическим адаптером.

Поддержка DVD в Windows 98 включает:

- Чтение секторов данных и поддержка системы команд DVD-ROM.
- Поддержка файловой системы UDF.
- Поддержка потоковых данных типа MPEG-2 для видео- и Dolby Digital для аудиоданных.
- Интерфейс DirectShow (ActiveMovie), заменяющий MCI (Media Control Interface) для улучшения совместимости с новым стандартным интерфейсом проигрывания VOB-файлов (Video Object).
- DirectDraw, поддерживающий передачу декодированного видеопотока с декодера MPEG-2 на графическую карту через выделенные шины.
- Поддержка защиты копирования и кодов регионов для аппаратных и программных декодеров.

Контрольные вопросы.

1. Какие устройства относятся к устройствам массовой памяти на сменных носителях?
2. Какие магнитные диски относятся к категории сменных устройств массовой памяти?
3. Охарактеризуйте диски Бернулли.
4. Охарактеризуйте "флоптические" гибкие магнитные диски.
5. Охарактеризуйте накопители Zip и Jaz.
6. Охарактеризуйте магнитооптические диски.
7. Как производится чтение и запись на магнитооптические диски?
8. Какова организация магнитооптических дисков?
9. Какова емкость и размеры МОД?
10. Какие стандарты поддерживают МОД?
11. Какая плотность записи и какие методы кодирования и модуляции применяются в мод?
12. Какие размеры секторов обычно используются в МОД?
13. Какова скорость вращения МОД, время доступа, скорость записи и считывания?
14. Охарактеризуйте технологию записи LIMDOW.
15. Охарактеризуйте особенности форматирования МОД.
16. Какие проблемы могут возникнуть при использовании в МОД секторов, размером в 2048 байт?
17. Боятся ли МОД относительно сильных внешних магнитных полей?
18. Охарактеризуйте технологию MSR.
19. Охарактеризуйте CD-R, CD-RW, CD-WO.
20. Охарактеризуйте Audio-CD.
21. Охарактеризуйте CD-E, CD-RW, PD.
22. Охарактеризуйте разновидности носителей CD.
23. Охарактеризуйте диски, выпускаемые для записи (болванки) и их маркировку.
24. Отчего зависит цвет оптических дисков и каков срок их службы?
25. Дайте краткую характеристику устройству приводов CD-ROM, CD-R, CD-RW.
26. Какая терминология принята применительно к устройствам чтения и записи компакт-дисков?
27. Как организована адресация и размещение данных на CD?
28. Какими стандартами описываются физические форматы компакт-дисков?
29. Опишите основные положения стандарта на аудиодиски CD-DA.
30. Опишите основные положения стандарта на CD-ROM.
31. Охарактеризуйте отличия структуры CD-R, CD-RW от CD-DA и CD-ROM.
32. Охарактеризуйте физический формат дисков CD-I.
33. Охарактеризуйте стандарт CD-ROM XA.
34. Охарактеризуйте многосеансовые диски.
35. Как расшифровываются аббревиатуры EFM, CIRC, ECC, EDC и что они означают?

36. Охарактеризуйте файловые системы HSG(HSF) и ISO 9660.
37. Охарактеризуйте файловые системы Rock Ridge, HFS, Joliet, Romeo.
38. Охарактеризуйте файловую систему UDF.
39. Охарактеризуйте новый стандарт файловой системы ECMA-168 и приложение Windows Direct CD.
40. Охарактеризуйте разновидности компакт-дисков, используемых в PC.
41. Какие режимы записи возможны для устройств и дисков CD-R?
42. Опишите особенности стирания перезаписываемого диска, открытие и закрытие сессии записи и закрытие диска.
43. Как формируется диск UDF?
44. Что содержат "чистые болванки" записываемых и перезаписываемых дисков?
45. Как должен конфигурироваться компьютер, чтобы застраховаться от опустошения буфера рекордера?
46. Охарактеризуйте системную поддержку CD-ROM.
47. Дайте общую характеристику спецификации для загружаемых CD-ROM.
48. Какие описатели определены в загрузочном каталоге загружаемого CD-ROM?
49. Что представляет собой загрузочный образ?
50. Охарактеризуйте дополнительные функции прерывания INT 13 BIOS, используемые для работы с CD?
51. Охарактеризуйте структуру пакета спецификации эмулируемого диска и командного пакета.
52. Охарактеризуйте диски с прямым доступом PD/CD, WORM и WARM.
53. Дайте общую характеристику DVD.
54. Охарактеризуйте разновидности DVD.
55. Охарактеризуйте DVD-R, DVD-RAM и другие диски DVD с перезаписью.
56. Как осуществляется защита от копирования и неавторизованного использования DVD?
57. Какая файловая система используется в DVD?
58. Что необходимо иметь в компьютере для воспроизведения DVD-видео?

В данном разделе использованы материалы из [3]. Более подробное описание принципов работы запоминающих устройств на оптических дисках можно найти в [6, 8, 11, 12].

8. Установка, обслуживание и тестирование дисков

Эта заключительная глава посвящена практике использования дисковых устройств. Здесь описываются типовые проблемы, с которыми сталкиваются пользователи, дается их объяснение на основе материала предыдущих глав и практические советы. Завершает главу краткое описание способов преодоления физических ограничений дисков — построение RAID-массивов.

8.1. Интерфейс подключения устройств

При всем богатстве выбора интерфейсов (ATA, SCSI, USB, 1394, Fiber Channel, LPT, PC Card), которые встречаются у устройств внешней памяти (как дисковых, так и ленточных), в большинстве случаев выбирать приходится между двумя основными — ATA (IDE) и SCSI.

Интерфейс ATA является самым массовым, у него есть ряд достоинств, но есть и недостатки, в ряде случаев делающие его применение нежелательным.

Плюсы ATA:

- Низкая цена устройств и кабелей, отсутствие потребности в терминаторах.
- Легкость конфигурирования устройства (у современных устройств достаточно указать его положение одним джампером).
- Относительно высокая скорость передачи данных по шине — у современных устройств и адаптеров в режиме UltraDMA типовая скорость 33 или 66 Мбайт/с, у «продвинутых» — до 100 Мбайт/с и выше. В традиционных режимах — до 16,6 Мбайт/с.
- Высокая эффективность в простых конфигурациях (при отсутствии потребности в одновременной работе с двумя устройствами на одной шине).

Минусы ATA:

- Малое количество подключаемых устройств. К одной шине могут подключаться не более двух устройств, типовое количество шин — 2, максимальное — 4 (предел — 8 устройств в компьютере).
- Используется только для подключения внутренних устройств (правда, в их число могут входить и съемные винчестеры).
- Ассортимент устройств уже, чем для SCSI, правда, все устройства широкого применения (винчестеры, МОД, Zip, считыватели CD и DVD, CD-рекордеры) существуют и с интерфейсом ATA/ATAPI.
- Низкая эффективность использования шины при работе многозадачных систем с несколькими устройствами. На время исполнения команды (до готовности к передачи данных) одно устройство блокирует шину, не позволяя даже дать задание другому. Средства многозадачности, описанные в ATA-4, гораздо слабее, чем в SCSI, и реально практически не используются.
- Незащищенность от ошибок передачи по шине. Контроль передач выполняется только в режимах Ultra DMA, его поддержка применяется далеко не всегда (из-за несовершенства драйверов и незнания пользователей).

У *интерфейса SCSI* самый главный недостаток — более высокая цена как на устройства, так и на адаптеры и аксессуары (кабели и терминаторы).

Плюсы SCSI'.

- Большое число устройств, которые можно подключить: до 7 устройств к одной узкой (или гибридной) шине контроллера и до 15 к широкой. Контроллер может быть 2-, 3-канальным, контроллеров может быть установлено несколько.
- Используется для подключения и внутренних, и внешних устройств.
- Высокая скорость передачи данных по шине — у распространенных современных устройств и контроллеров Wide Ultra2 SCSI — 80 Мбайт/с, у «продвинутых» Ultra3 — 160 Мбайт/с, в ближайшее время ожидается 320 Мбайт/с.
- Высокая эффективность использования шины в многозадачных системах:на время выполнения команды (пока данные к передаче не готовы) устройство не занимает шину, освобождая ее для обмена контроллера с другими устройствами.
- Высокая эффективность работы устройств в многозадачных системах: поддержка очередей и цепочек команд.
- Широкий ассортимент подключаемых устройств, выпускаемых как для IBM PC-совместимых компьютеров, так и для других платформ.

Минусы SCSI:

- Высокая цена устройств и контроллеров, усугубляемая ценой кабелей и терминаторов.
- Сложность конфигурирования (установка большого числа конфигурационных джамперов, правильная установка терминаторов).

Шина USB очень удобна для подключения внешних (переносных) устройств, но ее пропускная способность не превышает 1,5 Мбайт/с и разделяется между всеми устройствами шины. В ближайшее время ожидается переход на шину USB-2.0, которая при небольшом удорожании обещает пиковую скорость до 60 Мбайт/с (480Мбит/с).

Порт LPT тоже применяется для подключения внешних устройств, но он не так удобен (громоздкие кабели и разъемы) и не эффективен (при той же скорости, что и USB, сильно загружает процессор).

Шины PC Card (16-битный интерфейс) и *Card Bus* (32 бита) применяются для подключения внешних устройств к блокнотным ПК.

Шина IEEE 1394 (Fire Wire) в устройствах памяти применяется сдержанно, контроллеры этого интерфейса в PC пока что мало распространены. Скорость передачи по нынешним меркам средняя (400 Мбит/с).

Fiber Channel (FCAL) применяют для подключения очень больших устройств внешней памяти к мощным серверам. Этот интерфейс позволяет значительно разносить устройства памяти и компьютеры (возможно разделяемое использование устройств). Эти свойства ценны для особо ответственных применений, когда требуется обеспечить живучесть системы даже при частичных разрушениях (при стихийных бедствиях, катастрофах и т. п. «радостях современного бытия»).

8.2. Установка новых устройств

Теоретически установка нового устройства в компьютер производится довольно просто:

1. На устройстве устанавливаются конфигурационные джамперы согласно планируемому месту подключения. У *устройств ATA/ATAPI* (см. п. 5.2.1.10) устанавливается только джампер адресации в одно из положений: M (Master, ведущее, оно же устройство 0), S (Slave, ведомое, оно же устройство 1) или, что гораздо реже, CS (Cable Select, кабельная выборка).

ВНИМАНИЕ

На одном канале ATA не должно быть двух ведущих или двух ведомых устройств. В старых системах (с неинтеллектуальным контроллером ATA) для работы ведомого наличие ведущего в канале было обязательным, новые контроллеры при отсутствии ведущего сами исполняют его интерфейсные обязанности.

У *устройств SCSI* устанавливается идентификатор SCSI ID и некоторые опции (терминаторы, контроль паритета и др., см. п. 5.3.1.5).

ВНИМАНИЕ

На одном канале SCSI не должно быть устройств с совпадающими идентификаторами. Каждый канал должен иметь включенные терминаторы на обоих концах шлейфа. Тип интерфейса устройства — линейный (SE), дифференциальный (Diff) или низковольтный дифференциальный (LVD) —должен соответствовать интерфейсу контроллера. Сочетание дифференциального (Diff) с линейным (SE) или низковольтным дифференциальным (LVD) недопустимо.

Когда в системе имеются диски и ATA, и SCSI, порядок устройств определяется опциями CMOS Setup и SCSI Setup, а не только джамперами, установленными на дисках.

2. Устройство устанавливается в **выключенный** компьютер, к нему правильно (без переворотов, смещений и заламываний контактов) подключаются интерфейсный и питающий кабели.

3. Компьютер включается и выполняется встроенная утилита CMOS Setup. В пункте *Standard CMOS Setup* устанавливаются требуемые типы дисков: 0-47 для старых дисков небольшого объема, 47 (*User Type)* для дисков ATA любой геометрии, *Auto* (автоматическое распознавание) или *None* (нет диска ATA).

Для жестких дисков ATA удобно выбрать опцию *Auto;* тогда если BIOS с устанавливаемым диском работает без проблем (см. п. 8.3), то все установки и замены устройств не будут требовать входа в CMOS Setup. Определение параметров *User Type,* как правило, производится через пункт *IDE Autodetection* (автоматическое определение подключенных устройств и выбор способа трансляции геометрии). В некоторых случаях (например, при проблемах BIOS с большими дисками) параметры диска вводят вручную: задают число цилиндров, головок, секторов на треке, способ трансляции *(LBA* или *CHS).* Выбор режима LBA наверняка означает лишь способ трансляции геометрии (см. п. 3.1.2), при этом обращения к жесткому диску могут выполняться как в режиме LBA, так и в CHS, на усмотрение BIOS. Параметры парковки *(Landing Zone)* и предкомпенсации записи *(Write Precomp)* ни на что теперь не влияют, их можно не указывать.

Для дисков SCSI и *устройств ATAPI* (CD-ROM, CD-RW, CD-R, DVD-ROM...) в опциях CMOS Setup параметры не указывают; если эти диски единственные (дисков ATA нет), то в опциях ставится *None* или *Auto.* Если используются и диски ATA, то параметры в CMOS Setup относятся именно к ним. Параметры жестких дисков SCSI сообщаются контроллером SCSI (его BIOS). Когда POST передает управление

модулю SCSI BIOS для инициализации, на экран может выводиться приглашение ко входу в утилиту SCSI Setup по определенному сочетанию клавиш *(Ctrl+A, Ctrl+B, F2, F10).* Эта утилита может предоставить возможность выбора параметров («геометрию») подключенных дисков, а также выполнить их низкоуровневое форматирование.

В пункте *BIOS Features Setup* выбирают требуемую последовательность опроса загрузочных устройств, в которой могут присутствовать следующие:

- A: — стандартный НГМД;
- C:, D:, E:, F — диски ATA по порядку их подключения (ведущее устройство первого канала, ведомое устройство первого канала, ведущее устройство второго канала, ведомое устройство второго канала). Диск, с которого произошла загрузка, получает имя C:. Если по порядку подключения он не первый, то нумерация остальных дисков ATA сдвигается.
- SCSI — диск SCSI с минимальным значением идентификатора. При загрузке с него он получает имя C:, нумерация дисков ATA, если они есть, сдвигается. Для ряда версий BIOS загрузка возможна только с устройства с ID=0.
- CD-ROM (устройство ATAPI или SCSI);
- LS-120 (устройство ATAPI).

По умолчанию обычно установлено «A:, C:» — при наличии дискеты в дисководе загрузка будет осуществляться с нее, при отсутствии — с диска C. Для регулярной работы лучше устанавливать «C:, *A:»* или «C Only», тогда при случайно забытой дискете компьютер не будет пытаться с нее загрузиться (меньше шансов «поймать» загрузочный вирус). Если нужно грузиться с устройства SCSI при наличии дисков ATA, то ставится, например, «SCSI, A:».

4. Производится загрузка ОС с выбранного диска. Непосредственно перед загрузкой на экран могут выводиться списки обнаруженных устройств, их параметры, режим обмена. Согласно последним указаниям фирмы Microsoft, эта информация для пользователей считается излишней, так что пользователь может не увидеть ничего, кроме заставки загружаемой ОС.

Если вновь установленный диск не является первым (и загрузочным) и он сконфигурирован (разбит на разделы) и отформатирован, то остается лишь убедиться, что все его разделы «видит» загруженная ОС. После этого остается лишь «обживать» свободное пространство, при необходимости подправив пути к файлам и каталогам ранее установленных приложений. Если диск не сконфигурирован или не отформатирован, следует выполнить эти операции. Если диск должен стать загрузочным (первым), то на него следует еще и установить операционную систему. Если новый диск должен заменить старый, то после установки ОС на него требуется скопировать данные со старого диска.

С новыми дисками могут поставляться утилиты, облегчающие процесс установки и «переезда». Они выполняют конфигурирование и форматирование, а также копирование файлов со старого диска и даже перенос операционной системы. Попутно они могут установить «заплатки» на BIOS, если системная BIOS имеет проблемы с большими дисками. Однако эти утилиты и «заплатки» рассчитаны не на все случаи жизни и ОС и не всегда экономят время. Вспоминая старую поговорку о том, что «два переезда равны одному пожару», следует быть внимательным при проведении всяких «дисковых революций» — чтобы не было «мучительно больно» от потери «добра, нажитого непосильным трудом».

К сожалению, на практике встречаются проблемы разных уровней сложности. Самые простые и часто встречающиеся — неправильное физическое подключение (перепутаны джамперы, разъемы, кабели).

ВНИМАНИЕ

Неправильное подключение разъемов ATA (с переворотом или смещением) может блокировать системную плату — после включения она не будет подавать признаков жизни. Правильным подключением ситуация нормализуется.

Ошибочно собранный питающий разъем может вывести из строя подключаемое устройство (когда контакты цепей+5 В и+12 В перепутаны).

После внесения каких-либо изменений в дисковую подсистему (добавления/ удаления физических дисков, разделов, логических дисков) возможны изменения букв, связанных с логическими дисками. При этом некоторые разделы (логические диски) могут оказаться невидимыми. Это может произойти по ряду причин:

- По умолчанию DOS резервирует под диски всего пять логических имен (A:, B:, C:, D: и E:). Это связано с экономией памяти (каждый диск «съедает» свою порцию). Поскольку жесткие диски начинаются с C:, то при суммарном числе логических дисков (разделов на жестких дисках и CD-ROM) более трех последние будут невидимы. Для преодоления этого ограничения в файле CONFIG.SYS указывают параметр LASTDRIVE=x, где x — последняя доступная буква.
- Для CD-ROM в Windows 9x могут резервироваться специально выбранные буквы, причем в панели управления можно задать начальную и конечную буквы. Этот диапазон не должен перекрывать

буквы, предназначенные для логических дисков винчестера.

- Раздел жесткого диска может быть перекрыт сетевым диском, назначенным на его букву.
- Windows 98 позволяет скрывать логические диски, но эта возможность устанавливается по желанию пользователя (например, утилитой Tweak UI).
- Сжатый логический диск может оказаться невидимым, после того как его несущий диск переместился на иную букву (DriveSpace их теряет). Чтобы их снова увидеть, требуется запустить «Мастер компрессии» или вручную подправить файл DBLSPACE.INI.

Пожалуй, самые сложные проблемы возникают с установкой больших (по меркам BIOS и загружаемой ОС) дисков. Им посвящен следующий параграф.

Относительно безобидны проблемы производительности — когда обмен данными с диском оказывается медленнее, чем рекламируется. Здесь следует помнить, что в системе контроллер-кабель-диск максимально возможные режимы обмена определяются самым слабым звеном. Высокие скорости обмена требуют использования качественных шлейфов, и без нарушений правил топологии.

ВНИМАНИЕ ———————————————————————————————

Интерфейс ATA в режимах PIO и DMA не предусматривает контроля достоверности передачи по самой шине. Возможные неконтролируемые ошибки в этом тракте могут быть причиной нарушения целостности данных, включая серьезные повреждения файловой системы. Контроль достоверности на шине ATA выполняется только в режиме UltraDMA. Шина SCSI имеет контрольный бит паритета, но его использованием часто пренебрегают, запрещая контроль при конфигурировании устройств (а зря!).

Для дисков ATA в CMOS Setup можно задавать некоторые параметры, влияющие на производительность:

- IDE PIO (Auto, 0-4) — режим передачи IDE (PIO Mode). Ограничивает максимальный режим, предлагаемый контроллером (устройство ограничит его своими возможностями). Автоматическое согласование режимов работает не со всеми устройствами, поэтому иногда приходится его явно ограничивать. Может задаваться отдельно для каждого канала или устройства;
- IDE DMA Transfer Mode (Disabled, Type B , Standard) — режим DMA при передаче данных по интерфейсу ATA;
- IDE Ultra DMA (Enable/Disable) — разрешение использования режимов Ultra DMA;
- IDE Multiple Sector Mode или IDE Block Mode (Enable/Disable или число) — разрешение мультисекторного режима передачи (указание максимального количества секторов);
- IDE 32-bit Transfer (Enable/Disable) — разрешение 32-битного обращения к регистру данных IDE. Ускоряет обмен с дисками, но может быть источником проблем при некорректных драйверах.

Эти параметры определяют возможности, которыми может пользоваться контроллер и BIOS при обращениях к данным дискам. Реальное использование этих возможностей зависит от применяемой ОС. В выводимой табличке BIOS может показать, например, режим обмена UDMA 4 (со скоростью 66 Мб/с). Однако это лишь потенциальные возможности — MS DOS режим DMA вообще не использует, а другие ОС могут потребовать установки специальных драйверов (и, естественно, 80-проводного шлейфа). Без дополнительных усилий пользователя обмен с диском в режиме Ultra DMA поддерживается в Windows 98. Стандартные драйверы для Windows 95 и NT, OS/2 Warp 4.0 используют режимы PIO; Windows 95 OSR2 использует DMA. В этих ОС для использования Ultra DMA требуется обновление драйверов.

8.3. Проблемы использования больших дисков

Преодоление барьера 528 Мбайт (см. п. 3.1.2), вызванного несогласованностью формата вызова BIOS *INT 13h* и формата регистров дисков ATA, было только началом «бега с препятствиями», начавшегося в середине 90-х годов. Проблемы больших дисков порождаются интерфейсными ограничениями устройств и вызовов дискового сервиса BIOS, недочетами реализации алгоритмов трансляции в BIOS, форматами хранения информации о дисках, способами определения параметров диска и методами работы с дисками, используемыми ОС, а также принятой системой конфигурирования дисков.

Когда пользователь запускает автоматическое определение типа диска ATA (пункт *IDE Autodetection* в BIOS Setup), BIOS для каждого диска запрашивает параметры (командой *Identify Device)* и предлагает пользователю один или несколько вариантов геометрии, различающихся способами трансляции (LBA, CHS, ECHS). Среди этих вариантов один считается (по мнению BIOS) предпочтительным; он будет принят автоматически, если пользователь установит тип Auto. Выбранная геометрия — число цилиндров, головок и секторов на трек — заносится в CMOS; эти параметры можно установить и вручную. Далее при инициализации диска BIOS пытается выполнить ATA-команду *Initialize Device Parameters,* заказывая указанное число головок и секторов на треке. Если эта попытка увенчалась успехом, диск с заданными параметрами воспринимается как существующий и он становится

полноправным участником дальнейших событий — загрузки ОС и определения логических дисков. Если диск отказывается принять данные параметры, то выдается сообщение «Hard Disk fail» и данный диск уже не воспринимается системой. Возможен вариант, когда BIOS для автоматического определения геометрии использует данные из таблицы разделов, если таковая уже имеется на диске (по границе любого раздела можно определить число секторов на треке и число головок). С одной стороны, это обеспечивает переносимость дисков с машины на машину — на новом месте BIOS будет работать по той же геометрии, что была принята до него. Но если таблица разделов повреждена, BIOS будет пытаться навязать диску некорректные параметры, которые могут быть и неприемлемы для диска. Результатом будет сообщение «Hard Disk Fail» и недоступность диска. Такой диск можно «оживить», задав параметры вручную и переписав (или просто обнулив) таблицу разделов.

При установке большого диска могут проявиться какие-нибудь неожиданные ограничения, как правило, связанные с BIOS компьютера. Самый безобидный вариант — BIOS неправильно определит размер диска (он окажется неожиданно малым). При этом с другого диска (дискеты, CD-ROM) загрузка будет выполняться нормально и можно будет предпринять программные меры преодоления очередного барьера. Правильные параметры можно попытаться установить вручную (установить тип *User и* ввести значения CHS), но BIOS может и не позволить установить нужное число цилиндров. Хуже, когда с появлением нового диска компьютер «зависает» на определении параметров диска во время POST (если в CMOS Setup стоит тип *Auto),* при входе в CMOS Setup или во время начальной загрузки. В этом случае следует в BIOS Setup задать (вручную) «безопасные» параметры CHS 1024×16×63 (504 или 528 Мбайт, смотря как считать) или даже задать *Type 1* (615×4×17), что позволит выполнить загрузку ОС (с другого устройства). Далее следует либо обновить BIOS, либо прибегнуть к применению фирменных спасательных средств, которые предоставляются изготовителями дисков. Если предполагается обновление BIOS, то для выполнения этой процедуры можно отключить новый диск логически (поставив для него *None)* или даже физически, отключив от него интерфейсный кабель.

Самый радикальный способ преодоления барьеров — обновление BIOS. Если это сделать не удается, то используют дополнительные конфигурационные джамперы, изменяющие логическую организацию диска, а также загружаемые резидентные «заплатки» для BIOS. С помощью дополнительных джамперов на дисках ATA возможны следующие изменения:

- Ограничение объема диска на уровне ближайшего актуального барьера (как правило, за счет уменьшения сообщаемого числа цилиндров). При этом доступным оказывается не весь объем диска.
- Изменение сочетания параметров сообщаемой геометрии по умолчанию (слова 1, 3 и 6 паспорта диска). Тогда BIOS воспользуется более благоприятным (для данной реализации) алгоритмом трансляции и обойдет барьер. Так, например, ряд моделей дисков имеют два варианта геометрии — с 16 и 15 головками.
- Сообщение числа цилиндров в слове 1 паспорта диска менее 4092 (для преодоления барьера 2,1 Гбайт), но при этом в словах 60-61 указывается настоящая полная емкость. Позволяет определить реальную емкость накопителя расширенному сервису *INT 13h* и ряду ОС.
- Конфигурирование одного диска в виде двух устройств половинной емкости, но при этом канал ATA он занимает полностью, как устройство 0 и устройство 1 одновременно. Этот вариант встречался на некоторых дисках объемом около 1 Гбайт (так обходили первый барьер).

Наличие и назначение дополнительных джамперов зависит от конкретной модели устройства, а их восприятие — от применяемой ОС, а также версии BIOS (системной или загружаемой «заплатки», см. п. 8.3.2). Чаще всего современные диски имеют один дополнительный джампер AC (Alternate Capacity), который заставляет диски объемом 2,1-33,8 Гбайт сообщать геометрию по умолчанию 4092×16×63 (2,1 Гбайт), а диски большего размера — сообщать объем 33,8 Гбайт в словах 60-61 (трехмерная геометрия для них 16 383×15×63 — 7,9 Гбайт). Если приходится использовать джампер для ограничения емкости в угоду BIOS, то для использования полного объема диска в ОС, пользующихся информацией BIOS (DOS, Windows, OS/2), следует использовать драйвер DDO (см. п. 8.3.2). В инструкциях к большим дискам даются рекомендации по их конфигурированию. Так, например, диски Fujitsu MPF3102/3153/3204AT и MPE3xxx AH имеют джампер 1-2, ограничивающий число цилиндров до 4092 (при этом доступная емкость ограничивается на уровне 2,1 Гбайт). Для получения полной емкости в системе с барьером 2,1 Гбайт следует не устанавливать джампер, а задать число 4092 цилиндра в CMOS Setup вручную, после чего запустить утилиту ONTRACK Disk Manager (DiskGo!).

8.3.1. Ограничения BIOS и ATA

Для начала заметим, что диски ATA в IBM PC имеют шесть (!) различных описаний геометрии, иногда противоречащих друг другу. «Букет» этих параметров сообщает, например, утилита GLCHK (от Western Digital); пример ее отчета приводится на рис. 8.1.

```
Model: ST38421A
      Serial. 6BD0CG3H
    Firmware: 8.01
    INT 13h #: 80h                    Sectors  Total
                  Cylinders  Heads/Track   Sectors           Capacity
                  ---------  ----- \----   -------           --------
      Drive Default:   16368     16    63   16498944=Max LBA   8447 MB
      Drive Current:   16368     16    63   16498944=Max CHS   8447 MB
   INT 41h Physical:   16360     16    63   16490880=C*H*S     8443 MB
    INT 41h Logical:    2046    128    63   16498944=C*H*S     8447 MB
    Regular INT 13h:    1023    128    63    8249472=C*H*S     4223 MB
   Extended INT 13h:    2046    128    63   16498944           8447 MB
```

Рис. 8.1. Пример отчета о параметрах жесткого диска

Расшифруем английские названия параметров:

- *Drive Default* — заводские параметры диска по умолчанию — C, H, S и общее число доступных секторов. Эти параметры сообщаются в словах 1, 3,.6 и 60-61 паспорта диска по ATA-команде *Identify Device.*
- *Drive Current* — текущие параметры диска, сообщаемые в словах 54, 55, 56 паспорта диска. Устанавливаются по ATA-командам *Initialize Device Parameters* и *Set Max Address.* Команду *Initialize Device Parameters* выдает BIOS при инициализации жесткого диска во время POST, указывая желаемое число головок и секторов на треке в соответствии с принятой схемой трансляции.
- *INT41h/46 Physical* — физические параметры диска (C, H, S), определенные в HDPT (см. табл. 3.3). Для первого диска на HDPT указывает вектор *INT 41 h,* для второго — *INT 46h;* адрес HDPT любого диска возвращается в *ES:BX* по INT 73h(8). Таблицы HDPT формируются BIOS по параметрам, хранящимся в энергонезависимой памяти, или автоматически формируются в соответствии с обнаруженными устройствами. По этим параметрам BIOS обращается к устройствам ATA при выполнении стандартных вызовов в системе CHS.
- *INT41h/46 Logical* — логические параметры диска (C, H, S), определенные в HDPT. Формально ограничены значениями 65535×256×63. Эти параметры фигурируют в программном интерфейсе при стандартных вызовах *INT 13h.*
- • *Regular INT 13h* — логические параметры диска (максимальные значения C, H, S), возвращаемые в регистрах при вызове *INT 13h*(8). Ограничены значениями 1023, 255, 63. Как видно, они могут отличаться от логических параметров, указанных в HDPT, на которую ссылаются результаты этого же вызова.
- • *Extended INT 13h* — физические параметры диска (C, H, S и общее число секторов), возвращаемые в буфере по вызову *INT 13h*(48h), см. табл. 3.10. Трехмерные параметры могут быть недействительными (при числе секторов более 15 482 880), на это укажет нулевое значение бита 1 в слове 2. Трехмерные параметры задаются 32-битными числами, общее число секторов — 64-битным числом, так что ограничение пока необозримо.

Есть еще и седьмое описание — реальная геометрия устройства — число поверхностей, цилиндров и секторов на трек (переменное для разных зон), но, к счастью, никакое системное ПО и утилиты его не видят и не используют.

Теперь перечислим барьеры емкости для дисков ATA.

BIOS *INT 13h* (без расширений) имеет теоретический предел **8,5 Гбайт** (1024×256×63×512=8455716864), так что операционные системы, работающие с дисками через старый сервис BIOS *INT 13h* (например, MS DOS), уже не могут использовать современные диски в полном объеме. Расширенный сервис оперирует 64-битным линейным адресом сектора, так что о барьере 2^{64}×512 байт в ближайшие десятилетия можно не вспоминать.

Спецификация ATA имеет предел адресации **137 Гбайт:** в режиме CHS 65 536 цилиндров, 16 головок, 255 секторов на трек дают максимальное количество 267 386 880 секторов, при размере 512 байт — 136 902 082 560 байт (136,9 Гбайт). В режиме LBA используется 28-битный адрес сектора, что дает 2^{28}×512=137 438 953 472 байт (137,4 Гбайт). В 2001-2002 годах этот барьер в промышленности, судя по темпам роста прошлых лет, будет пройден. Однако, согласно спецификациям ATA/ATAPI (см. п. 5.2.1.4.2), в своем паспорте диск сообщает трехмерную (CHS) геометрию по умолчанию только в пределах 16 383×15×63=15 481 935 секторов **— 7,9 Гбайт**, даже если его реальная (и адресуемая) емкость больше.

Барьер 528 Мбайт порожден несовпадением ограничений BIOS и ATA по отдельным координатам CHS. Проблема возникла в 1993 году, в 1994-м была принята идея трансляции геометрии в BIOS (см. п. 3.1.2).

Кроме этих основополагающих препятствий есть еще ряд частных.

Барьер 2,1 Гбайт (1996 год) проявился в тех машинах, где в BIOS экономили ячейки энергонезависимой памяти, предназначенной для хранения параметров диска. Там в байте с логическим номером головки два старшие бита использовали для номера цилиндра, чем ограничили разрядность номера головки в 6 бит. Однако для того, чтобы число физических цилиндров, начиная с 4096, можно было втиснуть в 10 бит логического интерфейса, его требуется разделить уже на 8 и на столько же умножить число головок. Однако 4-битное число головок ATA в пределах шести отведенных битов можно умножить лишь на 4. Отсюда и выходит предел транслируемого (в режиме ECHS) объема диска 4095×16×63×512=2 113 413 120 байт. При загрузке с большего диска происходят зависания либо объем диска усекается (остается число цилиндров по модулю 4096). Так, диск 2,5 Гбайт воспринимался как 429 Мбайт. Диски большего размера часто выпускаются со специальным джампером, переключающим их в режим 4092×16×63. Радикально проблема решается обновлением BIOS.

Барьер 3,2 Гбайт был порожден ошибкой трансляции в ряде реализации Phoenix BIOS версий 4.03 и 4.04, исправленной в 4.05 и далее.

Барьер 4,2 Гбайт (1997 год) возник по вине DOS, «не понимающей» числа головок 256 (очевидно, где-то для его представления используется один байт, и 256 превращается в 0). При использовании трансляции ECHS (со сдвигом битов) число цилиндров делится на степень двойки, так чтобы получилось не более 1023, а число головок умножается на то же число. В результате для дисков, сообщающих геометрию с 16 головками и 63 секторами, пределом и будет 4,2 Гбайт (8192×16×63 транслируется в 1024×128×63). Больший объем потребует увеличения числа цилиндров, и следующим шагом окажется роковое число головок 256. Выход из положения есть — вместо геометрии с 16 головками, предлагаемой по IDE Autodetecting в BIOS Setup, вручную установить геометрию с 15 головками. При этом число цилиндров относительно сообщенного нужно увеличить в 16/15= 1,06667 раз и округлить до меньшего целого. Более новые версии BIOS (Phoenix) выполняют этот пересчет для устройств с 16 головками и числом цилиндров более 8192, после чего выполняют обычную трансляцию с помощью сдвигов. Трансляция через LBA не имеет данного барьера.

Барьер 7,9 Гбайт в режиме ECHS возникает, когда предыдущий (4,2 Гбайт) обходят, уменьшая число головок до 15 (умножением на 16 оно превратится в 240). Тогда предел логической геометрии оказывается 1024×240×63×512=7 927 234 560 байт. В режиме LBA этого барьера нет.

Барьер 8,4 Гбайт (1998 год) обусловлен отказом от использования конфигураций с 256 логическими головками (в угоду DOS/Windows). Достигается только при трансляции LBA: 1024×255×63×512=8 422 686 720 байт — немногим меньше теоретического предела традиционного *INT 13h.*

Для дисков объемом более 8,4 Гбайт понятие логической геометрии уже не представляет интереса, поскольку в рамки трехмерного интерфейса BIOS они не вписываются. На всякий случай приняли, что они сообщают геометрию 16 383×16×63, но работают с ними в полном объеме уже в режиме линейной адресации.

Однако на этом барьеры не кончились.

Барьер 33,8 Гбайт (1999 год) появился оттого, что стандартное число головок 16 и 63 сектора на трек дает число цилиндров более 65 535, и теперь оно не помещается в отведенное для этих целей 16-разрядное слово, в частности и в таблице параметров жесткого диска (HDPT) в BIOS. У Award BIOS 4.51 встреча с таким диском может вызвать зависание POST. На дисках большего размера джампер альтернативной конфигурации уже не используют (считается, что компьютер уже готов к таким объемам). Такие диски сообщают геометрию по умолчанию как 65 536×16×63, а работа ОС со всем объемом диска уже не требует использования устаревших описаний дисков в BIOS.

8.3.2. Преодоление ограничений BIOS

Если системная BIOS компьютера имеет один из вышеперечисленных барьеров, то следует, по возможности, ее обновить. Новые версии BIOS можно найти в Сети (на сайте производителя системной платы или в других местах). Замена BIOS в современных системных платах с флэш-памятью не составляет особого труда (правда, может потребовать временного отключения защиты записи и стирания, согласно инструкции к системной плате). Если же заменить BIOS не удается, то существует ряд идеологически сходных утилит, поставляемых производителями с новыми дисками. Их основой является Disk Manager фирмы ONTRACK Data International Inc. Название Ontrack Disk Manager ассоциируется как со способом установки, так и размещением программных средств. Одно время утилита Disk Manager поставлялась прямо на начальных треках (on track) новых дисков. Утилиты могут запускаться и со специальной загрузочной дискеты. Самая главная часть ПО — динамический оверлейный драйвер DDO (Dynamic Drive Overlay) — располагается не в файле, а на фиксированном месте нулевого трека диска. Драйвер DDO, называемый также XBIOS (расширение BIOS), решает три основные задачи:

- автоматически идентифицирует подключенные устройства ATA. В старых версиях BIOS иногда эта функция просто отсутствовала, в более новых могла «ломаться» на вышеперечисленных барьерах.

При использовании джампера альтернативной емкости диска, «обманывающего» системную BIOS, драйвер определяет реальную емкость диска по словам 60-61 его паспорта;

- позволяет поддерживать дополнительный контроллер ATA (в старых BIOS предусматривался только один);
- корректно работает с дисками, используя трансляцию геометрии для объемов свыше 528 Мбайт и обходя последующие барьеры.

Драйвер XBIOS (или его аналог EZ-BIOS) замещает программный код сервиса *INT 13h* системной BIOS, и он должен загружаться в память до загрузки ОС. Для этого на место стандартного MBR (цилиндр 0, головка 0, сектор 1) помещают загрузчик этого драйвера, MBR (модифицированный) — в сектор 2, а код обработчика *INT 13h* — в секторах 3-17. При загрузке с жесткого диска система загружает в память первый сектор диска и передает управление на его начало, благодаря чему будет исполнена программа загрузки драйвера (DDO) с начальных секторов диска. После загрузки драйвера управление передается главному загрузчику, который определяет активный раздел и загружает его, уже пользуясь новым кодом драйвера *INT 13h.* Таким образом, «заплатка» на BIOS будет загружаться всякий раз в начале загрузки с жесткого диска. Драйвер занимает 5 Кбайт стандартной памяти (conventional memory), совместим со стандартными менеджерами памяти DOS/ Windows и может быть перемещен в верхнюю память.

Утилита Disk Manager в процессе установки позволяет настраивать режимы, в которых драйвер будет обращаться к устройствам ATA: режимы PIO или DMA, многосекторные передачи, кэширование чтения и записи, 32-разрядный доступ к регистру данных контроллера и тому подобные опции. Эти настройки отражаются в «теле» драйвера, записываемом в начальные секторы диска.

Если в системе установлено более одного диска, то драйвер помещается, естественно, только на первый диск (с которого выполняется загрузка), а обслуживать он может и все остальные. Если этот диск потребуется изъять, тот, который станет первым, должен быть преобразован: на него должен быть установлен Disk Manager.

При обычной загрузке с дискеты загрузчик «заплатки» исполняться не будет, поэтому доступ к жесткому диску будет некорректным, что может привести к потере данных. Для возможности загрузки с дискеты предусмотрено следующее: перед загрузкой ОС нужно вынуть дискету, что приведет к загрузке первого сектора жесткого диска и XBIOS. Когда в системе установлен Disk Manager, до загрузки ОС (после инициализации XBIOS) на экране появляется голубой аншлаг с заставкой ONTRACK, а для загрузки с дискеты следует нажимать «пробел» до и после установки загрузочной дискеты. На загрузочной дискете (ее готовит утилита Disk Manager) кроме ОС располагается файл драйвера, на него имеется ссылка из файла AUTOEXEC.BAT.

Драйвер EZ-Drive — вариация фирмы Micro House на тему Disk Manager (поставляется с дисками Western Digital) — при инициализации предлагает нажать клавишу *Ctrl* — тогда появится сообщение о состоянии и предложение нажать клавишу «A» для загрузки с дискеты. После ее нажатия следует вставить загрузочную дискету и нажать любую клавишу. Для предотвращения обращений к жесткому диску в обход EZ-BIOS имеется опция защиты «floppy boot protection», автоматически включающаяся при инсталляции EZ-Drive. Однако эта опция работает только для DOS/Windows 9x и не поддерживается для Windows NT 3.51/4.0 и OS/2 Warp3.0.

Драйверы данного типа совместимы с ОС DOS 5.0 и выше, Windows 9x, Windows NT 3.51 и 4.0 (не используя дополнительных установок джамперов), Windows 3. lx, OS/2 Warp 3.0 и 4.0. **Не совместимы** с ОС Macintosh, Novell NetWare, Unix, Linux. Для Windows NT не должен устанавливаться дополнительный джампер, ограничивающий число цилиндров для дисков более 4,2 Гбайт, а в CMOS Setup диск описывается вручную как 1023×16×63 (или как Type 9). Драйверы совместимы с компрессорами Stacker, DriveSpace, DoubleSpace.

Если в системе уже используются диски, сконфигурированные с помощью Disk Manager, утилита установки EZ-Drive позволяет преобразовать их под новый EZ-Drive (вместе Disk Manager и EZ-Drive использоваться не могут). Утилита ONTRACK Disk Manager позволяет выбирать формат (местоположение) таблицы разделов между стандартным и собственным. Собственный формат (Ontrack Proprietary Format) предотвращает доступ к таблице разделов, когда оверлейный драйвер не загружен (это полезно для защиты от потери данных). Однако этот формат «понимают» только ОС MS-DOS, PC-DOS, Novell DOS, Windows 3.x, OS/2 V3.0 (Warp Full Pack), OS/2 V2.0-3.0 (требуется обновление драйверов), Windows NT 3.51/4.0, Windows 95. Другие ОС обращаются к таблице разделов в обход BIOS и будут искать таблицу разделов в обычном месте.

Поскольку программы данного типа загружаются и исполняются до загрузки операционной системы, они воспринимаются ею как часть системной BIOS. На них можно возложить и дополнительные функции — например, пародирование загрузки ОС, а также антивирусную защиту (мониторинг попыток записи в загрузочные секторы и MBR). Драйвер может перехватывать обращение к MBR (вызов BIOS) и ограничивать доступ к нему по своему усмотрению. Он может вместо сектора 0,0,1 подставлять какой-нибудь другой сектор, но, конечно же, если драйвер загружен в память.

Драйверы диска, загружаемые с него самого, конечно, не являются лучшим решением проблем больших дисков. Это хорошо как подручное средство для временного решения проблемы. Применение драйверов данного типа доставляет дополнительные сложности и в условиях вирусных атак — «лечение» загрузочных вирусов может разрушить и сам драйвер диска. Кроме того, некоторые антивирусные программы могут ошибочно принять драйвер за загрузочный вирус и «вылечить» его (насмерть!). Диагностические программы могут быть несовместимы с этими драйверами.

Если удается обновить BIOS (перепрограммировав флэш-память), то следует деинсталлировать EZ-Drive с помощью штатной утилиты, которая аккуратно вернет MBR и таблицу разделов на штатное место, без потери данных на диске. Если данные не жалко, то можно «снести» EZ-Drive, запустив утилиту FDISK с ключом /MBR. После этого следует запустить FDISK повторно, сконфигурировать и отформатировать диск.

Если требуется деинсталлировать EZ-Drive, когда использовались и преобразованные разделы Ontrack Disk Manager, то предварительно следует удалить 63-секторное смещение этих разделов. Для удаления смещения в утилите имеется опция «Remove 63-sector offset», но для OS/2 Warp и Windows NT удалять смещение нельзя.

8.3.3. «Трехмерная геометрия» дисков SCSI

Для дисков SCSI (самих устройств) понятие трехмерной геометрии попросту отсутствует — при операциях обмена с ними пользуются линейным адресом (LBA), причем его длина составляет 32 бита. Таким образом, при 512-байтном секторе можно адресовать до 2 Тбайт данных (2^{32} секторов). Однако программный интерфейс стандартного дискового сервиса BIOS оперирует с трехмерной геометрией, и ее приходится эмулировать. Расширенный сервис, использующий ту же линейную адресацию (но уже 64-битную), никаких преобразований по пути к диску не требует.

Для дисков SCSI обработку *INT 13h* перехватывает BIOS контроллера SCSI, следовательно, геометрию должна «придумать» именно BIOS контроллера. Сам диск сообщает только свой объем — количество (M) и размер секторов (обычно 512 байт), который требуется «нарезать» на треки (назначить число секторов на трек S) и цилиндры (назначить число головок H и определить число цилиндров C= M/(H×S)). Привлекательно сочетание H=64, S=32 — тогда получается «круглое» значение объема цилиндра — 1 Мбайт (двоичный, то есть 2^{20} байт). Однако оно не годится для больших дисков, когда полученное число цилиндров превысит 1023. Здесь проявляются различия взглядов разработчиков SCSI BIOS на сочетания (H, S):

- Всегда использовать (64, 32).
- Пробовать (64, 32), а при получении C > 1024 использовать (255, 63).
- Выбирать сочетание установкой опций контроллера (SCSI Setup), например из (64,32) и (255,63) или (16,63), (64,32) и (64,63).
- Пробовать H=64, S=32, а при получении C≥1024 использовать H=128, S=32; если и при этом получается C≥1024, переходить на H=255, S=63.
- Выбирать первое из сочетаний (64,32), (64,63), (128,63) и (255,63), при котором C<1024; в крайнем случае усекать до C=1023.
- Читать таблицу разделов диска (если она есть) и из определения какого-либо существующего раздела брать значения последней головки (Hmax) и сектора (Smax), тогда H=Hmax+l, S=Smax. Если таблицы нет или значения из нее неприемлемы, то выбирать сочетание одним из вышеприведенных способов.
- Читать таблицу разделов диска (см. выше), а если C или S получается слишком большим, принимать S=17 и H=2^n так, чтобы C<1024. Это странный способ, поскольку при M> 128×1024×17 (1,1 Гбайт) число головок получится более 256 и будет усекаться по модулю 256.

Идея использования информации из таблицы разделов хороша тем, что повышает шанс переносимости диска с одного контроллера на другой. Для большей строгости стоит посмотреть на описание раздела с самым большим начальным цилиндром и проверить, совпадает ли его трехмерное описание с линейным. Если это так и число цилиндров, вычисленное по общему объему (M), не превысит 1023, то следует принимать данное описание; иначе исходить только из общего объема и действовать согласно логике SCSI BIOS. Эти меры предосторожности страхуют от использования ошибочной информации из таблицы разделов.

Какой из вариантов представления геометрии будет использован в каждом конкретном случае, зависит от модели контроллера SCSI, версии его BIOS, установок конфигурирования контроллера и, наконец, от самого диска (есть ли на нем таблица разделов). Из этого следует возможная непереносимость дисков с информацией с машины на машину или при смене контроллера. Если новый контроллер не пользуется информацией из таблицы разделов и его «воззрения» на геометрию не совпадут с использованной во время предыдущего конфигурирования и форматирования дисков, то диск в DOS/Windows перестанет загружаться. Такому диску придется заново (с чистого листа!) создавать

таблицу разделов, естественно, с потерей прежних данных. Чтобы не ломать голову над изучением проблем совместимости, часто для дисков SCSI при установке выполняют низкоуровневое форматирование, которое «сносит» и MBR с таблицей разделов, и всю информацию с диска. После этого геометрия определится согласно логике SCSI BIOS (см. выше). Целесообразность низкоуровневого форматирования для «зачистки» таблицы разделов сомнительна — достаточно обнулить нулевой логический блок любым средством редактирования дисков. В системе команд SCSI низкоуровневое форматирование является стандартной и обязательной командой (не то что в ATA). Распространенные рекомендации по применению этой процедуры, похоже, относятся к области «древней истории» дисков. Для современных дисков низкоуровневое форматирование может оказаться даже вредным, поскольку оно может испортить заводские параметры оптимизации, правда, «умный» диск не даст себя испортить. Низкоуровневое форматирование дисков обычно можно запустить из меню конфигурирования SCSI-контроллера, в которое можно попасть по специальному приглашению после POST. Поскольку поддержка дисков SCSI находится в модуле BIOS контроллера, соответствующими опциями или джамперами использование этого модуля ROM BIOS должно быть разрешено. Если в системе стоит несколько однотипных контроллеров, то достаточно разрешить ROM BIOS на одном из них. Если контроллер расположен на системной плате, SCSI BIOS входит в область системной BIOS.

Из вышеизложенного следует, что если диск SCSI кажется меньшим, чем указано изготовителем, то проблема гнездится в SCSI BIOS. Возможно, ее удастся решить, изменив настройки в Setup контроллера SCSI.

8.3.4. Ограничения операционных систем

Ограничения операционных систем определяются тем, как они работают с диском. Если ОС использует для работы с диском традиционные функции BIOS INT 13h, то в лучшем случае она может работать с диском до 8,4 Гбайт, в худшем — с каким-либо барьером BIOS (см. выше). Если ОС работает через расширенные функции, то барьер пока необозрим (но таких ОС пока немного: Windows 95B (OSR2), Windows 98, Windows 2000). Если ОС обращается к дискам только через BIOS, то и объем диска она спрашивает через INT 13h(8) или INT 13h(48h), со всеми вытекающими последствиями.

Таблица 8.1. Ограничения на размеры разделов и дисков для различных операционных систем

Операционная система	файловая система	Предельный размер раздела		Предел объема диска	Требуется ли поддержка BIOS	Возможность использования EZ Drive
		загрузочного	другого			
DOS 4.0-6.22	FAT 16	2,1 Гбайт	2,1 Гбайт	8,4 Гбайт	Да	Да
Windows 95A	FAT 16	2,1 Гбайт	2,1 Гбайт	8,4 Гбайт	Да	Да
Windows 95B (OSR2)	FAT 16 [1]	2,1 Гбайт	2,1 Гбайт	2,0 Тбайт	Да	Да
Windows 95B (OSR2)	FAT 32 [1]	2,0 Тбайт	2,0 Тбайт	2,0 Тбайт	Да	Да
Windows 98	FAT 32[1]	2,0 Тбайт	2,0 Тбайт	2,0 Тбайт	Да	Да
Windows NT 3.51	VFAT16	4,0 Гбайт	4,0 Гбайт	8,4 Гбайт	Да	Да
Windows NT 3.51	NTFS	4,0 Гбайт	8,4 Гбайт	8,4 Гбайт	Нет	Да
Windows NT 4.0	VFAT16	4,0 Гбайт	4,0 Гбайт	8,4 Гбайт	Да	Да
Windows NT 4.0	NTFS[2]	4,0??7, 8 Гбайт	2,0 Тбайт	2,0 Тбайт	Нет	Да
Windows 2000	NTFS	Неогр.	Неогр.	Неогр.	Нет	
OS/2 Warp	HPFS	4,3 Гбайт	4,3 Гбайт	6,4 Гбайт[5]	Да	Да
Unix	FFS .	2,0 Тбайт	2,0 Тбайт	2,0 Тбайт	Дет	Не требуется
NetWare 3.12	NetWare[3]		8,4 Гбайт	8,4 Гбайт		Нет
NetWare 4.1	NetWare[4]		8,4 Гбайт	8,4 Гбайт		Нет
NetWare 5	NetWare[4]		2,0 Тбайт	2,0 Тбайт		Нет
Red Hat Linux5,2	ext2fs	8,4 Гбайт	2,0 Тбайт	2,0 Тбайт		Нет

[1] Для дисков более 8,4 Гбайт требуется поддержка расширений *INT 13h.*

[2] Для дисков более 8,4 Гбайт требуется установка Service Pack 3 и 4 (новый драйвер atapi.sys).

[3] Для дисков более 8,4 Гбайт следует использовать драйверы NWPA с модулями IDEATA.HAM от 25.08.98 и IDEHD.CDM от 05.08.98 или более поздними.

[4] Для дисков более 8,4 Гбайт требуется установка драйверов IDEATA.HAM от 25.08.98 и IDEHD.CDM от 05.08.98 или более поздних.

[5] Для дисков более 8,4 Гбайт требуется установка «заплаток».

Если ОС определяет размер диска в обход BIOS, то интересен вопрос, как она это делает. Для дисков SCSI объем (количество секторов на всем устройстве) сообщается однозначно по соответствующей команде SCSI. Для дисков ATA ситуация сложнее: согласно спецификациям ATA/ATAPI (см. п. 5.2.1.4.2), при объеме более 7,9 Гбайт в словах 1, 3 и 6 паспорта сообщается трехмерная геометрия 16 383×15×63=15 481 935 секторов (около 7,9 Гбайт). Истинный объем можно узнать только из слов 60-61. Операционные системы Windows NT 3.51 и 4.0, Nowell NetWare 4.11 (стандартная поставка) читают слова 1, 3 и 6 паспорта диска (трехмерную геометрию по умолчанию) и перемножают эти величины. На слова 60-61 (общее число секторов) они не обращают внимания, по старинке полагая, что увидят там примерно то же самое. Таким образом, они увидят объем диска не более 7,9 Гбайт. Для того чтобы эти ОС пользовались словами 60-61 и видели полный объем дисков ATA, требуется обновление драйверов. Для Windows NT 4.0 должен быть установлен Service Pack 4. Для NetWare 4.11 и NetWare 5.0 требуются драйверы IDEATA.HAM от 25.08.98 и IDEHD.CDM от 05.08.98 или более поздние. Для NetWare 3.12, где используются драйверы IDE.DSK или ISADISK.DSK, формально решение проблемы не предлагается, но можно установить драйверы NWPA (комплект модулей в файле NWPA_UP.EXE), которые работают с теми же драйверами IDEATA.HAM и IDEHD.CDM.

В табл. 8.1 приводятся ограничения, актуальные для популярных операционных систем. Для систем, работающих в обход BIOS, ограничение на загрузочный раздел (размер и положение) может быть жестче, чем на другие ее разделы. Это связано с тем, что загрузчику ОС на начальном этапе все-таки приходится пользоваться средствами BIOS (пока не будут загружены собственные драйверы). Под «другим разделом» подразумевается раздел, отведенный под логический диск (расширенный раздел DOS, который показывает утилита FDISK, может вмещать произвольное число таких разделов).

8.4. Конфигурирование и форматирование дисков

Для использования в качестве хранилища данных диск должен быть отформатирован. Различают два уровня форматирования дисков:

- низкоуровневое форматирование (Low Level Formatting, LLF) — формирование треков (запись заголовков и заполнение области данных физических секторов) на всех рабочих поверхностях дисков;
- логическое форматирование (высокоуровневое) — для DOS/Windows 9x это создание загрузочного сектора, FAT, корневого каталога и пометка в FAT дефектных (не прошедших верификацию) кластеров.

Для жестких дисков между этими уровнями вклинивается *конфигурирование диска* — разбивка на разделы, выбор активного раздела (только на первом физическом диске) и создание логических дисков в расширенном разделе. Низкоуровневое форматирование жестких дисков (кроме SCSI) пользователями в обычных условиях не выполняется.

ВНИМАНИЕ ——————————————————————————————————

Выполняя конфигурирование и форматирование, а также используя утилиты восстановления, выполняющие низкоуровневое форматирование, рекомендуется отключать все остальные винчестеры, дабы избежать возможных ошибок пользователя.

Логическое, или просто *форматирование* выполняет утилита FORMAT, которой обязательно нужно указать имя логического диска (A:, B:, C:...). Напомним, что дискета вся является логическим диском; на винчестере логическими дисками являются разделы, помеченные кодами, «понимаемыми» данной операционной системой. В DOS/Windows 9x на жестком диске можно форматировать только первичный раздел или те подразделы расширенного раздела, для которых определены логические диски (разделы и логические диски определяются утилитой FDISK). Кроме утилиты FORMAT, форматирование выполняют и некоторые другие утилиты, а в Windows 9x форматирование диска можно запустить из меню, выскакивающему, например, при щелчке правой кнопкой мыши по значку диска.

При форматировании выполняется инициализация загрузочного сектора — в него помещается программа-загрузчик и инициализируются поля, определяющие структуру диска (см. табл. 2.3). Таблица FAT и область корневого каталога обнуляются, после чего диск выглядит пустым. «Быстрое» форматирование на этом и останавливается, но по полной программе следует еще и проверить читаемость всех секторов. Для этого выполняется верификация (холостое чтение с проверкой на наличие ошибок), и кластеры с ошибочными секторами помечаются в FAT как дефектные. Хуже, когда дефектные секторы попадают в область FAT или загрузочного сектора. Область FAT в принципе можно переместить вглубь диска, увеличив число зарезервированных секторов (между загрузочным сектором и первой копией FAT), правда, утилиты такую возможность не предлагают, а вручную это слишком кропотливый труд. Положение загрузочного сектора (раздела) на жестком диске в принципе можно изменить через описатель раздела. Но на дискете положение этого сектора фиксировано, как и положение MBR на жестком диске, так что диск с дефектом первого сектора нулевой головки нулевого цилиндра для использования непригоден.

После форматирования в области данных остаются данные старых файлов (и каталогов). Для дискет полное форматирование совмещается с низкоуровневым форматированием. После такого форматирования старые данные затираются заполнителем. Для жестких дисков ATA низкоуровневое форматирование в обычных условиях не выполняется. Для дисков SCSI утилита низкоуровневого форматирования обычно «зашита» в BIOS контроллера SCSI.

Конфигурирование жестких дисков — формирование таблиц разделов — выполняется утилитой FDISK. Она позволяет прочитать существующую таблицу разделов, добавить или удалить разделы, создавать и удалять логические диски в расширенном разделе. Один из разделов (но не расширенный) может быть помечен как активный, правда, утилита позволяет устанавливать флаг активности только у первого физического диска. Если при помощи FDISK необходимо сделать активным раздел другого диска, его придется сделать первым (аппаратным переключением или конфигурированием через BIOS Setup).

ВНИМАНИЕ ———————————————————————————————————

Любой жесткий диск должен иметь таблицу разделов! Таблицы разделов являются важным элементом дисковой памяти, и на случай восстановления после сбоев (или поражения вирусами) полезно иметь их копии на дискете. Эти копии можно создать, а при необходимости и восстановить на диск, например, утилитой DISKEDIT.

Форматирование и конфигурирование затрагивает не все доступные секторы диска. За сектором с таблицей разделов (главной и расширенными) всегда остается свободное место до конца логического трека (поскольку любой раздел должен начинаться с начала трека). Это место (около 32 Кбайт у современных дисков с 63 секторами на треке) может использоваться втайне от операционной системы:

- в него помещают загружаемые программные «заплатки» на BIOS (см. п. 8.3.2);
- это место «гнездовья» загрузочных вирусов. Эти вирусы замещают программный код главного загрузчика своим, который до загрузки ОС загружает в память свои вредоносные процедуры;
- иногда его используют для привязки ПО к конкретной машине (на самом деле, к винчестеру).

Хотя для «зачистки» диска достаточно обнулить сектор с MBR (после этого его можно конфигурировать и форматировать с «чистого листа»), искушенные пользователи обнуляют весь нулевой трек нулевого цилиндра (если там нет ничего полезного). При этом сами разделы никоим образом не затрагиваются, и к ним можно получить доступ, восстановив MBR и таблицу разделов.

Обратим внимание на то, как в таблице разделов задаются границы разделов: здесь указываются как трехмерные (CHS) координаты начала и конца раздела, так и линейный адрес начала и длина. При этом условились, что раздел должен заканчиваться по концу цилиндра (максимальный номер головки и сектора), а начинаться по началу трека (номер сектора — 1).

Утилита FDISK при создании разделов должна определять и записывать их границы в обеих системах. Однако описание CHS своим трехбайтным форматом принципиально ограничивает делимый объем диска на уровне чуть меньше 8,5 Гбайт, у больших дисков разделы будут перекрываться из-за малой емкости описателей (24 бит). Линейная форма описания оперирует четырехбайтными числами, следовательно, его емкость ограничена на уровне около 2 Тбайт (2^{32} секторов). Старые (от MS DOS 6.x) утилиты FDISK позволяют разделять диски объемом до 8,5 Гбайт, но объемы существующих больших разделов показывают правильно (пользуются линейным описателем длины). Новые утилиты этот барьер преодолевают, при этом вместо запредельных значений CHS-координат подставляются «заглушки» 1023, 255, 63, но не всегда. Утилита FDISK из комплекта Windows 98 и Windows 95B (OSR2) при разрешенном использовании поддержки больших дисков ставит границы разделов в формате CHS некорректно. Номера начального и конечного цилиндров будут представлены по модулю 1024. Так, например, для диска 20,4 Гбайт (логическая геометрия 2491×255×63), разделенного пополам, начало второго раздела будет показано на 222-м цилиндре (вместо 1246-го), а конец на 442-м (вместо 2490-го). Корректными будут только линейные описатели.

Для дисков до 8,5 Гбайт имеется явная избыточность описания. Казалось бы, можно пользоваться любой формой описания, но все не так просто. Возникает вопрос: какое ПО каким описанием пользуется и как оно его трактует. К примеру, новые версии утилиты DISKEDIT пользуются исключительно линейной адресацией секторов, и, казалось бы, это хорошо. Однако если посмотреть такой утилитой на диск, сконфигурированный старой (от MS DOS 6.22) утилитой FDISK, то в режиме логического просмотра физического диска не удастся забраться дальше раздела с первым логическим диском. Если первый логический диск из расширенного раздела по размеру совпадает с первичным (так часто получается, если использован предельный размер для FAT16 в 2 Гбайт), то утилита будет «топтаться на месте» (для нее описание как бы зацикливается само на себя). Если размеры разные, то переход по начальному линейному адресу следующей EPR выведет отнюдь не на нужный сектор. Старая версия DISKEDIT, использующая трехмерные описатели, перемещается по всем разделам этого диска без проблем.

Границы разделов задаются утилитой FDISK, и трехмерное описание будет занесено в таблицу

разделов в той геометрии диска, которая сообщается функцией BIOS INT13h(8*)*. Именно тут кроется причина возможной непереносимости дисков размером более 528 Мбайт с одной машины на другую, если у этих машин применяются различные схемы трансляции геометрии в BIOS. Главный загрузчик, расположенный на диске, предполагает, что границы разделов, заданные в MBR, соответствуют геометрии (логической), которой пользуется BIOS *INT 13h*. Если оказывается, что это не так, загружаемый раздел не будет найден и загрузка с жесткого диска окажется невозможной. Аналогично, загрузчик логического диска, а потом и ОС «верят» в геометрию диска, описанную в его загрузочном секторе. Границы разделов задаются утилитой FDISK, параметры логических дисков задаются утилитой FORMAT. Эти утилиты «спросят» геометрию диска у BIOS первой машины, но из-за возможных различий трансляции на другой машине для того же диска BIOS будет использовать иную геометрию.

Поясним сказанное на примере с числами. Пусть диск конфигурировали и форматировали на машине, где он был логически описан как H=64, S=63, и, согласно MBR, первый раздел начнется с трехмерного адреса C=0, H=1, S=1 (на границе трека) и линейного 63. Теперь переставим его в машину, где для него же H=128, S=32, и попытаемся с него загрузиться. Главный загрузчик, согласно таблице разделов, читает сектор 0,1,1, но при новой схеме трансляции это оказывается физический блок диска с линейным номером 32. Естественно, сигнатуру загрузочного сектора (55AAh в самом конце), как и самого загрузочного сектора, реально расположенного в 63 блоке, он не находит и останавливает систему с сообщением «Missing Operating System». Если бы число секторов в обеих схемах совпадало, а различалось число головок, то загрузчик первого разделся нашелся бы. Однако раздел, начинающийся с ненулевого цилиндра, тоже бы потерялся. Проблемы появились бы и у загрузчика первого раздела: он прочитает параметры своего диска (см. табл. 4.3), отсчитает количество секторов до корневого каталога (с учетом положения первого сектора), поделит их на секторы и цилиндры согласно этому описанию и сформирует запрос (C,H,S, например 0,8,61) на чтение корневого каталога. Согласно первой схеме, это будет физический блок с логическим адресом 0×64×63+8×63+61-1=564. Но запрос будет отработан по второй схеме как на блок 0×128×32+8×32+61-1= 316, в котором загрузчик не найдет вожделенных названий загрузочных файлов IO.SYS и MSDOS.SYS, так что загрузка ОС опять сорвется, но теперь уже с сообщением «Non-System Disk or Disk Error...».

Этой проблемы не будет, если пользоваться только линейной адресацией LBA — она инвариантна. Однако груз совместимости — традиционного повсеместного использования CHS — не позволяет одним махом решить проблему. В Windows 95 появились новые типы разделов (коды 0Eh, 0Fh, а начиная с OSR2 еще и 0Ch), предназначенные для работы исключительно в режиме LBA. Для их использования поддержка LBA должна быть полной: и со стороны диска, и со стороны BIOS (должны поддерживаться функции расширенного сервиса с номерами 4xh). Трехмерные границы в таблице разделов задаются для них из расчета 63 секторов на треке и 255 головок, эти же параметры фигурируют и в логических дисках. Однако эти типы разделов для DOS (а 0Ch и для первой редакции Windows 95) являются неизвестными и, следовательно, недоступными. Эти типы назначаются утилитой FDISK из комплекта Windows 9x. В первой версии Windows 95 обращение к разделам с типами 0E и 0F из режима эмуляции MS DOS сопровождалось ошибкой, связанной с изменениями в структурах описателей дисков при переключениях между 32-битными драйверами Windows и 16-битными драйверами DOS 7. При записи файла на логический диск в разделе 0E из режима эмуляции MS DOS возможно появление «фантомных» дисков, выглядящих зеркальными копиями действительных дисков. Это чревато потерей данных на всем диске при записи данных в режиме MS-DOS. Проблема кроется в старом драйвере DISKTSD.VXD (каталог WINDOWS\SYSTEM\IOSUBSYS). Microsoft выпустила «заплатку» DSKTSUPD.EXE, автоматически обновляющую этот файл (новая версия от 22.03.96). Избежать этой проблемы можно просто — перед установкой Windows 95 сконфигурировать диск утилитой FDISK из комплекта MS DOS 6.22, а если использовать утилиту из Windows 95, запускать ее с ключом, запрещающим использование LBA (FDISK /X). Однако если с этой проблемой столкнулись, когда на диске имеется важная (и пока уцелевшая) информация, можно рискнуть изменить тип раздела (0Eh на 06h и 0Fh на 05h) с помощью какого-либо средства редактирования дисков. При этом следует позаботиться и о корректности задания границ разделов в трехмерном виде (и соответствии геометрии той, что используется логическими дисками). Конечно, перед этим «хирургическим» вмешательством важные данные должны быть сохранены на другом носителе. В более поздних выпусках Windows 9x эта проблема уже решена.

Перед выполнением конфигурирования диска пользователь должен спланировать желаемое разбиение дисков, учитывая ограничения файловых систем применяемых ОС. Так, например, если требуется доступ к разделам (логическим дискам) из MS DOS, то размер раздела не должен превышать 2,1 Гбайт, а файловая система должна быть FAT 16. Кроме того, такой раздел не должен выходить за пределы первых 8 Гбайт диска, доступных через стандартную BIOS.

Конфигурирование диска закладывает фундамент дисковой подсистемы компьютера, и после «постройки и заселения здания» (форматирования дисков и установки ПО) внесение изменений становится проблематичным. Windows 9x (как и DOS) не позволяет изменять размеры логических дисков (разделов) на ходу, то есть без потери данных. ОС с более сложными файловыми системами, например Novell NetWare и Windows NT, позволяют изменять (увеличивать) размеры логических дисков (томов)

без потери данных.

Все изменения размеров логических дисков выполняются утилитой FDISK, при этом **данные всех разделов, участвующих в изменениях, теряются.** По этой причине разбиение диска должно планироваться заранее, иначе «революции» потребуют сохранения данных на каком-либо дополнительном носителе. Правда, если изменения касаются последних логических дисков (расширенных разделов), а на первичном разделе и предшествующих логических дисках достаточно места, то временно сохранить данные можно и на дисках, не участвующих в- изменениях. Старая утилита FDISK позволяет удалять логические диски только с конца; освободившееся место можно использовать под логические диски иного размера. Утилита FDISK из Windows 98 позволяет удалять логические диски с любого места.

Несмотря на обязательность конфигурирования жестких дисков, Windows 98 ошибочно позволяет форматировать жесткий диск и без таблицы разделов (как сменяемый), правда, при этом он будет виден в системе только при установленном флажке «Removable». Однако использование жесткого диска в таком режиме чревато возможной потерей данных.

При выборе между FAT 16 и FAT32 (для разделов не более 2 Гбайт) следует учитывать различия в размерах кластеров (их влияние на производительность и эффективность использования диска, см. п. 2.1). Windows 98 (как и 95B) позволяет на ходу преобразовать файловую систему раздела из FAT 16 в FAT32. При этом велика вероятность, что существующие данные сохранятся. Однако перед выполнением этой процедуры следует **сохранить важные данные на другом носителе.** В ряде случаев преобразование может и не пройти успешно, например, если на диске имеются сбойные секторы или же просто внезапно пропадет питание.

Используя большие разделы и FAT32, нужно быть готовым к некоторым сюрпризам. Программы, не рассчитанные специально на FAT32, могут неправильно определять и показывать объем свободного дискового пространства (особенно если оно более 2 Гбайт). Утилиты установки новых дисков, копирующие «все файлы диска» со старого диска (FAT16) на новый (FAT32), могут «забыть» пересчитать размер кластера. После такого «копирования» жесткий диск с FAT32 должен быть переформатирован.

8.5. УТИЛИТЫ обслуживания дисков

Диски время от времени требуют некоторого обслуживания. После первоначальной установки периодически следует выполнять дефрагментацию диска, чтобы производительность компьютера не деградировала со временем, а также проверять диски на логические ошибки. В случае нештатных ситуаций (авария питания, вирусная атака...) приходится ликвидировать их последствия различной степени тяжести. Все эти действия выполняются с помощью утилит различного назначения и происхождения. Утилит существует великое множество, здесь перечислим лишь основные и наиболее известные утилиты.

8.5.1. Системные утилиты

Стандартные утилиты форматирования и конфигурирования входят в комплект поставки операционных систем.

FDISK — утилита конфигурирования жестких дисков. Интерфейс в виде меню позволяет:

- отобразить текущее разбиение диска на разделы и логические диски;
- создавать разделы и логические диски требуемого объема;
- удалять разделы и логические диски;
- назначать активный раздел (только для первого физического диска);
- выбирать номер физического диска, с которым выполняются манипуляции.

Утилита работает с информацией таблиц разделов (основной и расширенных). При запуске с ключом /MBR записывает в сектор 1 головки 0 цилиндра 0 код стандартного главного загрузчика (не меняя таблицу разделов). С помощью этого ключа можно «убить» (не всегда!) загрузочный вирус.

ВНИМАНИЕ ———————————————————————————————

При работе с драйверами типа EZ-Drive и им подобными использование ключа /MBR затрет их загрузчик, в результате чего после перезагрузки обращение к дискам может стать некорректным и привести к потере данных. Однако если драйвер загружен (на момент запуска FDISK), то он может не позволить переписать сектор 0,0,1.

Утилита FDISK из комплекта Windows 98 и Windows 95B (OSR2) при запуске спрашивает разрешение на использование поддержки больших дисков. Если эту поддержку разрешить, то любому создаваемому разделу размером более 512 Мбайт будет присвоен код файловой системы FAT-32.

FORMAT — утилита форматирования жестких и гибких дисков. Запускается из командной строки с обязательным указанием имени диска, например, FORMAT A:. При запуске можно указывать ключи, задающие параметры форматирования дискет и дополнительные опции: /Q— быстрое форматирование

(без верификации), /U — безусловное форматирование (без возможности отката), /S — помещение на диск системных файлов (диск становится загрузочным).

Остальные ключи можно узнать, запустив утилиту с ключом /?. Утилита из комплекта Windows 95B/98 имеет ключ /Z, позволяющий задать число секторов в кластере (для FAT32).

Утилиты FORMAT и FDISK используют информацию и сервисы BIOS *INT 13h*. Утилиты из комплекта MS DOS б.х и Windows 95A могут работать с дисками до 7,9 Гбайт, а также создавать и форматировать разделы FAT 16 размером до 2 Гбайт. Утилиты из Windows 95B и 98 работают с большими дисками, позволяя создавать разделы с FAT32 или FAT16. Версия утилит должна соответствовать версии операционной системы, из среды которой они запускаются.

ScanDisk — утилита проверки и исправления ошибок на диске, входит в состав MS DOS б.х, Windows 9x. Позволяет обнаруживать и исправлять ошибки в FAT, потерянные кластеры, пересекающиеся цепочки (кластеры, «принадлежащие» более чем одному файлу), ошибки в структуре каталогов, логические ошибки на сжатых дисках, а также сканировать поверхность дисков, выявлять и помечать сбойные кластеры и, по возможности, перемещать данные из сбойных секторов в нормальные. В Windows 9x утилита автоматически запускается при инсталляции (и переустановке) ОС, а также после аварийного завершения (прекращения) работы.

В Windows 9x при размере разделов более 16 Гбайт Scandisk может «вылетать» с ошибкой «Out of Memory Error». Для Windows 98 на сайте Microsoft имеется обновление, решающее эту проблему.

Когда ложные сообщения об ошибках, которые выдает Scandisk, не позволяют завершить установку Windows 9x, можно запустить SETUP с ключом /IS — запуск Scandisk выполняться не будет.

Defrag — утилита дефрагментации файлов, входит в состав MS DOS б.х, Windows 9x. Упорядочивает расположение файлов, стараясь разместить каждый файл в одной цепочке кластеров, что значительно повышает производительность дисковой подсистемы. Новый дефрагментатор для Windows 98, в дополнение к обычным действиям, старается еще и разместить файлы, используемые каждым ранее запущенным приложением, близко друг к другу. Естественно, это выполняется, лишь когда в системе накопится статистика связи файлов с приложениями (в LOG-файле). Дефрагментацию дисков желательно производить регулярно, при интенсивной записи файлов даже ежедневно. Дефрагментация больших дисков, особенно с FAT32 (при слишком большом числе кластеров), может занимать значительное время (до нескольких часов в особо запущенных случаях). Если она выполняется регулярно, то каждый сеанс будет занимать умеренное время.

DrvSpace — утилита обслуживания сжатых дисков, входит в состав MS DOS 6.x (может встречаться под именем DblSpace), Windows 9x. Позволяет сжимать диски, форматировать и «разжимать» сжатые диски, управлять монтированием томов сжатых дисков.

Вышеперечисленные утилиты работают с жесткими дисками только на уровне чтения/записи секторов, не забираясь в область низкоуровневого форматирования. При обнаружении сбойных секторов максимум, что они могут, — это пометить в FAT сбойные кластеры. Когда число сбойных кластеров становится значительным, имеет смысл изъять из обращения дефектные секторы, а то и треки и даже целые поверхности. Это можно сделать только добираясь до низкоуровневого форматирования. Для дисков SCSI имеется специальная команда, форматирующая весь диск. Интеллектуальный встроенный контроллер современного диска SCSI скроет все физически дефектные блоки, и диск вновь станет непрерывным массивом работоспособных секторов (но с полной потерей данных). Для дисков ATA низкоуровневое форматирование не стандартизовано, и каждый производитель использует свои собственные алгоритмы выполнения этой процедуры, причем они могут различаться от семейства к семейству. Если для низкоуровневого форматирования не требуется специального оборудования, то изготовители дисков поставляют к своим продуктам специальные утилиты. Они позволяют проверять поверхности диска и скрывать дефектные области за счет резервных секторов или уменьшения доступного объема диска. Эти утилиты можно найти на сайтах производителей дисков (но не всех).

ВНИМАНИЕ ———————————————————————————————————

«Низкоуровневые» утилиты пригодны лишь для работы с дисками, модели которых перечислены в аннотации к конкретной версии утилиты.

8.5.2. УТИЛИТЫ от производителей дисков

Производители дисков для облегчения их установки в систему поставляют специальные утилиты, в большинстве своем включающие ONTRACK Disk Manager-. Хотя все производители используют одну и ту же основу фирмы ONTRACK, лицензирование не позволяет запустить утилиту, полученную от конкретного производителя, если в системе не обнаруживается ни одного его диска. Если в системе стоят несколько дисков, то ряд функций разрешен только для «родных» дисков поставщика утилиты.

Утилита **Disk Manager** (DM. EXE) построена в виде меню, она руководит действиями пользователя и предлагает пояснения из встроенного руководства. В утилите имеется база данных по большому числу моделей накопителей разных фирм, выпущенных до появления очередной версии утилиты, в которой приводятся сведения о конфигурационных джамперах. Утилита позволяет конфигурировать (разбивать на разделы) и форматировать жесткие диски, не используя стандартных утилит FDISK и FORMAT.

Дополнительно утилита позволяет копировать системные файлы со старого диска, избавляя от необходимости реинсталляции ОС при установке нового диска., Можно и полностью скопировать содержимое старого диска на новый. Для смягчения последствий инсталляции дополнительных дисков имеется специальный пункт для коррекции ссылок на привод CD-ROM (должен работать для Windows).

Если BIOS не позволяет использовать полный объем диска, утилита в загрузочный сектор помещает программный код дискового сервиса, снимающий ограничения в 33,8 Гбайт (версия 9.53), 8,4 Гбайт, 2,1 Гбайт и 528 Мбайт (см. п. 8.3.1).

С помощью пунктов меню можно настроить параметры общения загружаемого драйвера с дисками ATA. После первоначальной установки их можно сменить — для этого следует запустить утилиту, сменить настройки и обновить код драйвера (для этого имеется специальный пункт меню Update Dynamic Data Overlay). Обновление кода может потребоваться и для установки новой версии драйвера (снимающей новые обнаруженные проблемы). Если используется драйвер DDO, то можно использовать не стандартный формат таблицы разделов, а собственный (Ontrack Proprietary Format). Однако если принять этот формат, его придется установить на всех подключенных дисках (но не все ОС с ним могут работать, см. п. 8.3.4). В утилите есть функция преобразования формата, но пользоваться ею следует осмотрительно.

ВНИМАНИЕ ———————————————————————————————

Начатое преобразование невозможно остановить (оно может занимать десятки минут для каждого диска). Если оно прервется из-за сбоя питания или по инициативе пользователя, данные будут потеряны.

Для пользователей Windows 3x и 9x в комплект входит утилита CD Update (CDUPDATE.EXE), которая позволяет подправить все ссылки на CD-ROM в файлах *.INI, *.STF, *.PIF, *.BAT, "WAR, *.LNK, *.ARG. ".CFG (за исключением AUTOEXEC.BAT и NET.CFG) и даже в реестре Windows. Коррекция ссылок может потребоваться, когда изменяется общее число логических дисков на жестких дисках после внесения каких-либо изменений в дисковую подсистему. Утилита ищет файлы данных типов во всех каталогах указанных дисков и вносит требуемые изменения (которые можно, к счастью, и отменить — все или выборочно).

Утилита позволяет выполнять низкоуровневое форматирование (результат не гарантируется), а также заполнить все секторы диска нулями, что можно использовать для уничтожения всех данных в целях соблюдения конфиденциальности.

В комплект Disk Manager входят и дополнительные файлы с драйверами для OS/2, Windows NT и Windows 3.x (32-разрядный доступ к диску); для DOS и Windows 9x никаких дополнительных драйверов не требуется.

Утилита **DiskGol** фирмы ONTRACK пришла на смену DM.EXE. По сути, это тот же Disk Manager (см. выше), «завернутый» в графическую оболочку, функционирующую в среде Windows. Утилита руководит действиями пользователя, желающего установить новый диск или изменить конфигурацию установленных. Она в графическом виде и с анимацией показывает, куда и как следует установить диск, как установить джамперы у конкретной модели (в базе имеются сведения о дисках IBM, Fujitsu, Quantum, Conner, Seagate, Samsung, Maxtor, Western Digital) при установке в требуемой позиции. Выдав руководство к действию, она завершает работу Windows и при последующей загрузке проверяет, выполнены ли ее указания по подключениям и/или переключениям. Далее утилита предлагает сконфигурировать диски и отформатировать разделы (в FAT16 или FAT32), а также, при необходимости, скопировать файлы с одного диска на другой. Утилиту можно загрузить с сайта производителя дисков — получить EXE-файл объемом 1,6 Мбайт, генерирующий стартовую дискету, с которой утилита устанавливается на компьютер с ОС Windows. На этой дискете остается и обычный вариант Disk Manager, который можно запускать из среды DOS.

На сайтах производителей можно найти и специализированные утилиты, иногда даже с низкоуровневым форматированием.

Western Digital

Для дисков *Western Digital* из утилит, доступных на сайте фирмы (www.wdc.com), отметим следующие:

EZ-Drive 9.06W и 9.09W — утилита, поставляемая с новыми дисками Western Digital. Утилита похожа на Disk Manager: позволяет конфигурировать (разбивать на разделы) и форматировать жесткие диски, не используя стандартных утилит FDISK и FORMAT. Если BIOS не позволяет использовать полный объем диска, утилита помещает в загрузочный сектор программный код дискового сервиса, снимающий ограничения в 8,4 и 2,1 Гбайт, а также 528 Мбайт (см. п. 8.3.2). Дополнительно утилита позволяет копировать системные файлы со старого диска, избавляя от необходимости реинсталляции ОС при установке нового диска. Можно и полностью скопировать содержимое старого диска на новый. Если уже используется Ontrack Disk Manager, утилита позволяет преобразовать его код в EZ-Drive, поддерживающий существующие диски (вместе они использоваться не могут).

Data Lifeguard Tools — набор утилит для обслуживания жестких дисков, включает:

- EZ-Install (DLGEZ.EXE) — инсталляция нового диска (включая установку загружаемого драйвера), заменяет утилиты EZ-Drive и Disk Manager.
- Diagnostics (DLGDIAG.EXE) — диагностика и низкоуровневое форматирование. Позволяет восстанавливать безошибочное состояние поверхностей дисков (только для дисков WD). Отдельные дефектные секторы скрываются без потери данных, исключение дефектных треков приводит к потере данных.
- BIOS Check (DLGCHK.EXE) — проверка поддержки дисков со стороны BIOS, отображение шести описаний дисков ATA (см. п. 9.3.1).
- Ultra ATA/66 Management (DLGUDMA.EXE) — управление опциями Ultra ATA (для дисков WD, поддерживающих эти режимы).

WD DataBack — утилита «спасения» файлов, потерянных или удаленных в результате аппаратных сбоев, действий вирусов, ошибок ПО и пользователей, повреждения поверхности диска. Работает в два этапа — на первом считывает с диска всю хоть как-то читаемую информацию и в упакованном формате сохраняет ее на другом носителе. На втором этапе занимается собственно логическим восстановлением файлов. Занимает две дискеты (для каждого этапа — своя дискета). Восстанавливает жесткие диски ATA, SCSI, сменные SCSI и подключаемые к LPT-портам. Лицензируется (регистрируется) на конкретный экземпляр жесткого диска, может быть перенесена вместе с этим диском на другой компьютер.

Maxtor

Фирма Maxtor для установки своих дисков предлагает пакет **MAXBLAST,** по сути, сходный с EZ-Drive, но производства StorageSoft. Утилита MAXBLAST.EXE создает загрузочную дискету со следующими утилитами, выполняющимися последовательно:

- SPLASH.EXE — загрузка драйвера EZ-Drive в оперативную память;
- EZMAX.EXE — определение подключенных дисков (типы, параметры дисков и их поддержка BIOS);
- DG.EXE — утилита установки диска, по функциям аналогичная DM.EXE (запускается предыдущей утилитой);
- MAXDIAG.EXE — утилита диагностики дисков ATA (только для дисков Maxtor), для некоторых семейств позволяет выполнять низкоуровневое форматирование. Запускается вручную.

Для отдельных семейств дисков на сайте можно найти утилиты низкоуровневого форматирования LXTLLF.EXE и MXTLLF.EXE.

Seagate

На сайтё предлагается «Мастер установки диска» DiskWizard — Disk Manager в графической оболочке под Windows. Для тестирования дисков предлагается SeaTools Disc Diagnostic, «понимающая» логическую структуру дисков с FAT (и FAT32); на физическом уровне может выполнять тесты чтения-записи секторов. Имеются также утилиты конфигурирования Ultra ATA, определения параметров диска, расшифровки записей таблиц разделов и некоторые другие.

IBM, Fujitsu, Quantum,

Эти фирмы для установки своих дисков предлагают утилиты **DiskGo!** (от ONTRACK). Хотя сама по себе утилита DiskGo! универсальна, лицензионные ограничения могут не позволить использовать утилиту, полученную от одной фирмы, с дисками другого производителя. IBM предлагает утилиту тестирования своих (и только!) дисков ATA и SCSI на пригодность (DFT — Disk Fitness Test), позволяющую также стирать информацию (обнулять все секторы) и выполнять низкоуроровневое форматирование дисков SCSI (но не ATA).

8.5.3. УТИЛИТЫ «третьих фирм»

В данном случае третьими фирмами являются те, которые не производят ни собственно дисков, ни операционных систем.

Утилита ***PartitionMagic*** фирмы PowerQuest предназначена для комфортного (для любителей мышино-графического интерфейса) создания, удаления, преобразования и объединения разделов жестких дисков. Утилита «понимает» структуры разделов с файловыми системами FAT12, FAT16, FAT32, NTFS, HPFS и Linux ext2 и работает с дисками любых размеров (в версии 5.0 упоминаются диски размером 20 Гбайт и более). В отличие от стандартной пары FDISK-FORMAT утилита позволяет «двигать» разделы даже без потери данных, а также преобразовывать разделы между перечисленными файловыми системами. Утилита позволяет и скрывать разделы, что полезно, например, для тестирования нестабильного и потенциально опасного ПО. В комплект входит BootMagic — средство выбора

загружаемого раздела, обеспечивающее «мирное сосуществование» нескольких ОС на одном диске. «Предметом и результатом труда» для утилиты являются стандартные структуры таблиц разделов (правда, иногда в них оказываются нестандартные параметры), никаких резидентных драйверов утилита не устанавливает. Утилита позволяет копировать разделы внутри диска, между дисками и даже копировать диски, содержащие несколько разделов. При копировании на больший диск выполняется масштабирование — на новом диске будет тот же состав разделов, но их размеры пропорционально увеличатся. Цель данного копирования — снять необходимость переустановки ОС и приложений при обновлении диска. Реальные результаты работы утилиты не всегда соответствуют заманчивым обещаниям.

Очень популярны утилиты от фирмы **Symantec** — ***Norton Utility,*** позволяющие обслуживать диски с разной степенью углубления в тайны их организации.

NDD (Norton Disk Doctor) — утилита, по назначению и возможностям напоминающая утилиту ScanDisk (см. выше).

SPEEDDISK — утилита дефрагментации, аналогичная Defrag.

DISKED IT — мощное средство просмотра и редактирования структуры дисков, позволяющее вносить изменения в любой байт любого сектора диска. Утилита предназначена для квалифицированных пользователей, знающих логическую структуру диска. Утилита позволяет просматривать и редактировать все области диска — таблицы разделов (главную и расширенные), загрузочные секторы разделов, таблицы размещения файлов, элементы каталогов и области данных. Просмотр содержимого может производиться как в логической форме в виде таблиц, описывающих просматриваемый сектор в соответствии с его назначением, так и в виде дампа с представлением каждого байта в шестнадцатеричном виде и ASCII. Для любого выделенного байта дампа легко прослеживается связь с элементом в логическом представлении, и наоборот. Утилита позволяет сохранять в виде файла любую область диска (например, FAT) и загружать эти файлы в требуемые секторы диска. Возможен выбор используемой копии FAT, копирование секторов, заполнение областей и ряд других возможностей.

8.6. Основные причины отказов дисков

Современные диски сами по себе являются весьма надежными устройствами с развитыми средствами контроля достоверности хранения информации. В то же время различные аварии в подсистеме внешней памяти — явление, увы, нередкое. Чтобы уяснить причины аварий и предупредить возможные потери данных — а это самое ценное в компьютере! — напомним основные моменты.

Диск связан с программой, исполняемой процессором компьютера, через память, контроллер интерфейса и собственно интерфейс. Даже самая правильная программа (включая ОС, драйверы, приложения и утилиты) на самом надежном диске может создавать логические ошибки, приводящие к потере данных, если в этой цепочке есть ненадежно работающее звено. Процессор, если он не разогнан, как правило, работает правильно. Источником ошибок может быть *оперативная память* совместно с кэшем и схемами управления памятью. К сожалению, эти ошибки зачастую не ловятся тестовыми программами, но проявляются в виде неожиданных «вылетов» или «зависаний» программ. «Лечатся» эти неисправности проверкой правильности конфигурирования памяти, соответствия модулей требованиям системной платы, корректности настроек временных диаграмм. Возможна «психологическая несовместимость» конкретных моделей модулей памяти с конкретными системными платами — скорее всего, здесь дело в геометрии проводников, влияющих на уровень помех. «Разогнанная» память — потенциальный источник ошибок. Достоверность хранения данных оперативной памятью в массовых ПК не применяется (память с паритетом отжила свой век). Для серьезных серверов, где данные особенно дороги, применяют память с ECC-контролем солидных производителей (например, Kingston). Кроме оперативной памяти, следует обращать внимание и на кэш-память. У современных процессоров кэш, как правило, имеет ECC-контроль, который зачастую отключают через BIOS Setup, а зря! Причиной сбоев памяти может стать и некачественный блок питания, особенно если у него отсутствует сетевой фильтр (бывает и так в погоне за дешевизной).

Следующее после памяти слабое звено этой цепи — интерфейс устройства. В шине ATA, вплоть до начала использования режимов UltraDMA, не было никакого контроля достоверности передачи информации по самой шине. В принципе, шина достаточно надежна (иначе работать с ней было бы невозможно), но только при соблюдении ограничений на длину и нагрузку кабелей. Во многих двухканальных контроллерах шины данных обоих каналов электрически соединены друг с другом (это дешевле!), но если на каждом канале использовать кабель максимально разрешенной длины, то в сумме получается перебор (по емкости проводников). Для скоростных режимов UltraDMA предназначен 80-проводный шлейф, но систему иногда можно заставить работать и на обычном шлейфе. К счастью, здесь работает CRC-контроль (правда, не все драйверы корректно отрабатывают его ошибки).

В шине SCSI применяется контроль паритета, хотя его иногда отключают — «чтоб лучше работало!». В самых «скорострельных» вариантах SCSI стали применять уже более сложные методы контроля достоверности передачи. Проблемы на шине чаще всего вызываются неправильным использованием терминаторов (см. п. 5.3.1.5.3).

Для повышения надежности хранения иногда используют зеркальное отражение (дублирование) дисков (см. п. 8.7). Однако от логических ошибок, порожденных неисправной памятью или программными ошибками (сбоями), эти меры не спасают.

Самая распространенная причина появления логических ошибок дисков — внезапное отключение питания, нажатие кнопки *Reset* или неуместное применение «комбинации трех пальцев» *(Ctrl+Alt+Del)* во время работы ОС и приложений.

Это, как правило, приводит к образованию потерянных или пересекающихся кластеров и, следовательно, повреждению пользовательских данных. Утилиты типа ScanDisk лечат эти недуги довольно быстро и эффективно (но данные могут быть потеряны).

ВНИМАНИЕ ———————————————————————————————————

Питание ПК можно выключать лишь при «погашенном» (shutdown) состоянии ОС — после появления разрешающей надписи (и когда после этого затихнет винчестер). Еще лучше, когда ОС сама выключает питание компьютера (для настольных ПК это возможно только в конструктиве ATX).

Распространенные ошибки пользователей — завершение работы приложения без сохранения нужных данных, ошибочные удаления и переименования файлов и прочие небрежности.

Самые неприятные последствия, как правило, вызывает действие особо «злобных» вирусов. Они могут разрушать данные на всех уровнях: на уровне файлов, системных областей логических дисков, таблицы разделов. Предотвратить это можно только соблюдением правил «гигиены» и профилактики с помощью свежих антивирусных средств. Обновление этих средств (как и сами вирусы) можно получить по Сети, которая становится доступной все большему числу пользователей.

Однако мелкие или крупные аварии дисковой подсистемы все же могут случаться и случаются.

ВНИМАНИЕ ———————————————————————————————————

Чтобы потом не было мучительно больно. Регулярно сохраняйте важные данные на носителе, отчуждаемом от компьютера.

Это могут быть дискеты, оптические диски CD-R или CD-RW, магнитооптические диски, Zip, Jaz, кассеты стримеров и другие, подходящие по объему, быстродействию и стоимости. При этом должно быть как минимум два комплекта носителей, чтобы в случае несчастья во время записи текущей копии вдали от компьютера был бы носитель с предыдущей более или менее свежей версией данных. В руководствах к программам резервного копирования иногда приводятся описания различных дисциплин смены носителей, обеспечивающих их круговорот в соответствии с требованиями к надежности и режиму эксплуатации носителей.

8.7. Повышение надежности и производительности дисков (RAID-массивы)

У накопителя на жестких дисках есть пределы физических возможностей, которых в ряде случаев бывает недостаточно. Здесь рассмотрим, как можно получить длительную скорость передачи данных выше, чем внутренняя скорость передачи устройства, а также повысить надежность дисковой системы. Эти задачи решаются с помощью параллельного использования нескольких дисков — дисковых массивов RAID (Redundat Array of Inexpensive Disks — избыточный массив недорогих дисков). Идея заключается в подключении группы обычных (недорогих), как правило, однотипных дисков к RAID-контроллеру — устройству, которое для хост-компьютера этот массив представляет как один диск с улучшенными свойствами. Улучшения касаются повышения надежности и (или) скорости обмена данными. В зависимости от алгоритма представления диска различают следующие типы (уровни) RAID:

- RAID 0 — дисковый массив без отказоустойчивости, простейшее средство повышения производительности. Данные разбиваются на блоки, которые равномерно распределяются по всем дискам массива. При параллельных обращениях хоста к различным областям «виртуального» диска, попадающим на разные диски, обращения реально распараллеливаются, что повышает производительность как по чтению, так и по записи. Отказ любого диска приводит к отказу всего массива. Пространство всех дисков используется полностью (избыточности нет).
- RAID I — зеркальное отражение (mirroring). Два (или более) диска дублируют друг друга. Запись информации выполняется одновременно на все диски, чтение — с любого свободного, в результате чего производительность чтения повышается. Отказ одного диска приводит только к снижению скорости чтения. Отказавший диск может быть заменен, и для ввода его в действие требуется просто копирование данных с оставшегося диска. Синхронизация нового диска (установление идентичности содержимого) требует довольно большого времени, но может выполняться на фоне обращений к «живому». Эффективность использования пространства дисков низкая (при двух дисках эффективный объем составляет лишь половину их суммарного, при большем числе — и того меньше).
- RAID 2 — избыточный массив, в котором биты данных «размазываются» по нескольким дискам, и еще несколько дисков несут проверочные коды Хэмминга (CRC), позволяющие исправить битовые ошибки. Отказоустойчивость ниже, чем у RAID 1, производительность невысокая. Избыточность

меньше, чем в RAID 1.

- RAID 3 — избыточный массив, отличающийся от RAID 2 тем, что вместо кодов Хэмминга (несколько дисков) используется лишь бит паритета (1 диск). Поскольку отказ (ошибка чтения) каждого диска определяется его встроенным контроллером, RAID-контроллер «видит» ошибочный диск и его бит вычисляет через бит паритета. Производительность, отказоустойчивость и эффективность использования пространства довольно высокие.
- RAID 4 — избыточный массив, напоминающий RAID 3, но расслоение данных производится на уровне секторов (а не байтов и битов). Реализация сложнее, но эффективнее обслуживание коротких запросов.
- RAID 5 — распределение и чередование данных и паритета по дискам, но для паритета не выделяется специальный диск (биты паритета распределяются по всем дискам). Обеспечивает высокую скорость записи, но скорость чтения ниже, чем в RAID 4.
- RAID 6 — более сложная схема, устойчивая даже к двойным отказам (но ценой низкой производительности).

Возможны и более сложные массивы, в которых используются двухступенчатые комбинации данных уровней:

- RAID 10 — массив RAID 0, собранный из пар зеркальных дисков (RAID 1). Обеспечивает высокую скорость и надежность, но ценой большой избыточности.
- RAID 30 — массив RAID 0, собранный из блоков RAID 3.
- RAID 50 — массив RAID 0, собранный из блоков RAID 5. Обеспечивает отказоустойчивость и высокую производительность.

Массивы RAID могут быть реализованы для хост-компьютера как аппаратно, так и программно. Аппаратный RAID-контроллер представляет собой интеллектуальное устройство со своим мощным микропроцессором, имеющее интерфейсы для подключения дисков, а также интерфейс подключения к хост-компьютеру. В качестве интерфейса дисков чаще всего используют SCSI, поскольку требуется подключение большого числа устройств, и высокая эффективность использования шины SCSI здесь вне конкуренции. Есть и RAID-контроллеры с интерфейсом ATA, но это не типично. В качестве интерфейса с хост-компьютером для внешних RAID-контроллеров, представляющих собой отдельное устройство со своим блоком питания и отсеками для установки дисков, может использоваться SCSI или FCAL. Для внутренних RAID-контроллеров характерно использование шины PCI (раньше использовали EISA и даже ISA, встречалась и VLB).

Функции RAID могут быть реализованы и программно, средствами ОС компьютера (например, эту возможность имеет Windows NT). При этом в качестве интерфейсов дисков используются интерфейсы существующих контроллеров (предпочтительно SCSI, но используют и ATA). Программный RAID-массив вполне справляется с задачей повышения надежности, но для решения задачи повышения скорости требуется большая вычислительная мощность процессора.

Контрольные вопросы.

1. Охарактеризуйте достоинства и недостатки интерфейса ATA.
2. Охарактеризуйте достоинства и недостатки интерфейса SCSI.
3. Дайте краткие характеристики шинам USB, PC Card, IEEE1394 (Fire Wire), порту LPT и интерфейсу Fiber Channel (FCAL) с точки зрения их использования для подключения дисковых накопителей.
4. Как выполняется конфигурирование дисков ATA (IDE) и SCSI?
5. Какие параметры дисков устанавливаются в пунктах Standard CMOS Setup и BIOS Features Setup?
6. По каким причинам после установки/удаления физических и логических дисков могут стать невидимыми некоторые разделы дисков и как устраняются эти недостатки?
7. Какие параметры, влияющие на производительность, устанавливаются в CMOS Setup для дисков ATA?
8. Чем порождаются проблемы больших дисков?
9. Что является самым радикальным средством для преодоления барьеров максимальной емкости дисков ATA?
10. Что можно изменять с помощью дополнительных джамперов дисков ATA?
11. Сколько различных описаний геометрии может встречаться в IBM PC для дисков ATA?
12. Перечислите барьеры емкости для дисков ATA.
13. Где располагаются драйвер DDO (XBIOS), какие задачи и как он решает?
14. Что позволяет делать утилита Disk Manager?
15. Как осуществлять загрузку ОС с дискеты при использовании дисков с Disk Manager?

16. С какими ОС совместимы драйверы типа EZ-Drive и Disk Manager?
17. Охарактеризуйте недостатки драйверов дисков, загружаемых с них самих.
18. Как организуется "трехмерная геометрия" дисков SCSI?
19. Какова логика работы SCSI BIOS при определении параметров H и S?
20. Охарактеризуйте особенности низкоуровневого форматирования SCSI дисков.
21. Охарактеризуйте ограничения ОС при работе с большими дисками.
22. Охарактеризуйте процедуру логического форматирования диска утилитой FDISK.
23. Охарактеризуйте разметку диска утилитой FDISK.
24. Какие участки дисков могут быть использованы в тайне от ОС?
25. В какой геометрии задаются границы разделов утилитой FDISK?
26. С чем связана возможная непереносимость дисков, емкостью более 528 Мбайт, с одной машины на другую?
27. Охарактеризуйте системные утилиты работы с дисками: FDISK, FORMAT, ScanDisk, Defrag и DrvSpace.
28. Охарактеризуйте утилиты Disk Manager и DiskGol.
29. Охарактеризуйте утилиты фирмы Western Digital.
30. Перечислите основные причины отказов дисков.
31. Дайте краткую характеристику различным типам RAID-массивов.

В данном разделе использованы материалы из [3].

Рекомендуемая литература

1. Вегнер В.А. и др. Аппаратура персональных компьютеров и ее программирование. IBM PC/AT и PS/2. М: Радио и связь, 1995. 224 с.: ил. (Библиотека системного программиста).
2. Гук М. Аппаратные средства PC. Энциклопедия - СПб: Питер Ком, 1998. -816 с.: ил.
3. Гук М. Дисковая подсистема ПК. - СПб.: Питер, 2001. - 336 с.: ил.
4. Кулаков В. Программирование на аппаратном уровне. Специальный справочник. - СПб: Питер, 2001. - 496 с.: ил.
5. Павлов В.А. Учебные материалы по курсу «Периферийные устройства ЭВМ»: Часть 1. СарФТИ, Саров, 2001. - 232с.: ил.
6. Павлов В.А. Учебные материалы по курсу «Периферийные устройства ЭВМ»: Часть 2. СарФТИ, Саров, 2001. - 271с.: ил.
7. Чепурной В. Устройства хранения информации. – СПб.: BHV- Санкт-Петербург, 1998. – 208 с., ил.
8. Николин В.А. Компакт-диски и CD устройства. – СПб.: Издательство Лань, 1997. – 112 с., ил.
9. Микропроцессорный комплект К1810: Структура, программирование, применение: Справочная книга/ Ю.М. Казаринов и др. - М.: Высш. школа, 1990. - 269 с.: ил.
10. Накадзима Х., Огава Х. Цифровые грампластинки: Пер. с япон. - М.: Радио и связь, 1988. - 168 с.; ил.
11. Михайлов В.И., Князев Г.И., Макарычев П.П. Запоминающие устройства на оптических дисках. - М.: Радио и связь, 1991. - 224 с.: ил.
12. PCI BIOS Specification . Revision 2.0. - PCI Special Interest Grope. 1993.
13. Phoenix BIOS 4.0 User's Manual. - Phoenix Technologies Ltd., 2000.
14. Avenger Super High Performance Graphics Engine for 3D Game Acceleration. — 3Dfx Interactive Inc., 1999.
15. Banshee High Performance Graphics Engine for 3D Game Acceleration. — 3Dfx Interactive Inc., 1999.
16. Matrox MGA-2064W Specification. - Matrox Graphics Inc., 1996.
17. Matrox MGA-G400 Specification. - Matrox Graphics Inc., 1999.

Учебное издание

ПАВЛОВ Виктор Александрович

ПОДСИСТЕМА ДИСКОВОЙ ПАМЯТИ ПК

Учебное пособие для вузов