

# ЭНЦИКЛОПЕДИЯ

Наиболее полное и подробное руководство

## Аппаратные интерфейсы ПК

Михаил Гук

традиционные порты  
и последовательные шины

шины расширения, интерфейсы  
устройства хранения и памяти

интерфейсы периферии,  
аудио- и видеоинтерфейсы



Михаил Гук

Наиболее полное и подробное руководство

# Аппаратные интерфейсы ПК

Москва • Санкт-Петербург • Нижний Новгород • Воронеж  
Ростов-на-Дону • Екатеринбург • Самара  
Киев • Харьков • Минск

2002

*Михаил Гун* Аппаратные  
интерфейсы ПК. Энциклопедия

Главный редактор  
Заведующий редакцией  
Литературный редактор  
Художник  
Корректоры  
Верстка

*Е. Строганова  
И. Корнеев  
А. Жданов  
Н. Биржаков  
Т. Брылёва, И. Хохлова  
Р. Гришианов*

ББК 32.973.2-044я20

УДК 681.327.8(03)

ГукМ.

Г93 Аппаратные интерфейсы ПК. Энциклопедия. — СПб.: Питер, 2002. — 528 с.: ил.

ISBN 5-94723-180-8

Книга посвящена аппаратным интерфейсам, используемым в современных персональных компьютерах и окружающих их устройствах. В ней подробно рассмотрены универсальные внешние интерфейсы, специализированные интерфейсы периферийных устройств, интерфейсы устройств хранения данных, электронной памяти, шины расширения, аудио и видеоинтерфейсы, беспроводные интерфейсы, коммуникационные интерфейсы, вспомогательные последовательные интерфейсы. Сведения по интерфейсам включают состав, описание сигналов и их расположение на разъемах, временные диаграммы, регистровые модели интерфейсных адаптеров, способы использования в самостоятельно разрабатываемых устройствах. Книга адресована широкому кругу специалистов, связанных с эксплуатацией ПК, а также разработчикам аппаратных средств компьютеризированной аппаратуры и их программной поддержки.

© ЗАО Издательский дом «Питер», 2002

Все права защищены. Никакая часть данной книги не может быть воспроизведена в какой бы то ни было форме без письменного разрешения владельцев авторских прав.

Информация, содержащаяся в данной книге, получена из источников, рассматриваемых издательством как надежные. Тем не менее, имея в виду возможные человеческие или технические ошибки, издательство не может гарантировать абсолютную точность и полноту приводимых сведений и не несет ответственность за возможные ошибки, связанные с использованием книги.

ISBN 5-94723-180-8

ООО «Питер Принт». 196105, Санкт-Петербург, ул. Благодатная, д. 67в.  
Лицензия ИД № 05784 от 07.09.01. Налоговая льгота -  
общероссийский классификатор продукции ОК 005-93, том 2; 953005 - литература учебная.  
Подписано в печать 05.07.02. Формат 70х100/16. Усл. п. л. 41,28. Тираж 5000 экз. Заказ № 714.  
Отпечатано с готовых диапозитивов в ФГУП «Печатный двор» им. А. М. Горького Министерства РФ  
по делам печати, телерадиовещания и средств массовых коммуникаций.  
197110, Санкт-Петербург, Чкаловский пр., 15.

# Краткое содержание

От автора .....	11
Введение .....	13
Глава 1. Параллельный интерфейс— LPT-порт .....	17
Глава 2. Последовательный интерфейс — COM-порт.....	48
Глава 3. Беспроводные интерфейсы .....	78
Глава 4. Последовательные шины USB и FireWire.....	87
Глава 5. Шина SCSI .....	114
Глава 6. Шины и карты расширения .....	145
Глава 7. Интерфейсы электронной памяти .....	235
Глава 8. Специализированные интерфейсы периферийных устройств .....	310
Глава 9. Интерфейсы устройств хранения .....	352
Глава 10. Интерфейсы компьютерных сетей.....	405
Глава 11. Вспомогательные последовательные интерфейсы и шины .....	420
Глава 12. Архитектурные компоненты IBM PC-совместимого компьютера .....	454
Глава 13. Интерфейсы питания, заземление и гальваническая развязка .....	505
Литература .....	516
Алфавитный указатель .....	517

# Содержание

От автора

Введение

Глава 11.1. Традиционный LPT-порт

1.2. Расширения параллельного порта

1.3. Стандарт IEEE 1284

1.3.1. 1.3.1. Полубайтный режим ввода — Nibble Mode

1.3.2. 1.3.2. Двухнаправленный байтный режим — Byte Mode

1.3.3. 1.3.3. Режим EPP

1.3.4. 1.3.4. Режим ECP

1.3.5. 1.3.5. Режимы и регистры ECP-порта

1.3.6. 1.3.6. Согласование режимов IEEE 1284

1.3.7. 1.3.7. Физический и электрический интерфейс

1.3.8. 1.3.8. Развитие стандарта IEEE 1284

1.4. 1.4.

Системная поддержка LPT-порта

1.5. 1.5.

Параллельный порт и PnP

1.6. 1.6. Применение LPT-порта

1.7. 1.7. Конфигурирование LPT-портов

1.8. 1.8. Неисправности и тестирование параллельных портов

Глава 2

2.1. 2.1. Интерфейс RS-232C

2.2. 2.2. Родственные интерфейсы и преобразователи уровней

2.3. 2.3. Асинхронный режим передачи

2.4. 2.4. Управление потоком данных

2.5. 2.5. Микросхемы асинхронных приемопередатчиков

2.6. 2.6.

Системная поддержка COM-портов

2.7. 1.7. Конфигурирование LPT-портов

2.8. 2.8. Использование COM-портов

2.9. 2.9. COM-порт и PnP

2.10. Неисправности и тестирование COM-портов

2.10.1. 2.10.1. Проверка конфигурирования

2.10.2. 2.10.2. Функциональное тестирование

2.10.3. 2.10.3. Питание от интерфейса, или Причины неработоспособности

МЫШИ

Глава 3

3.1. 3.1. Инфракрасный интерфейс IrDA

3.2. 3.2. Радиointерфейс Bluetooth

## Глава 4

### 4.1. Шина USB

- 4.1.1. 4.1.1. Организация шины USB
- 4.1.2. 4.1.2. Модель передачи данных
- 4.1.3. 4.1.3. Протокол
- 4.1.4. 4.1.4. Типы передач данных
- 4.1.5. 4.1.5. Синхронизация при изохронной передаче
- 4.1.6. 4.1.6. Хост
- 4.1.7. 4.1.7. Применение шины USB .....101
- 4.1.8. 4.1.8. Разработка собственных устройств USB

### 4.2. Шина IEEE 1394 - FireWire

- 4.2.1. 4.2.1. Физический уровень сети
- 4.2.2. 4.2.2. Протокол IEEE 1394
- 4.2.3. 4.2.3. Устройства и адаптеры 1394
- 4.2.4. 4.2.4. Использование 1394

## Глава 5

### 1.2. 5.1. Параллельные интерфейсы SCSI

- 5.1.1. 5.1.1. Кабели, разъемы, сигналы
- 5.1.2. 5.1.2. Терминаторы
- 5.1.3. 5.1.3. Протокол шины
- 5.1.4. 5.1.4. Конфигурирование устройств
- 5.1.5. 5.1.5. Подключение устройств к шине

### 5.2. 5.2. Интерфейс Fibre Channel

### 5.3. 5.3. Хост-адаптер SCSI

### 5.4. 5.4. Управление интерфейсом и выполнение команд

## Глава 6

### 6.1. Шины ISA, EISA и PC/104

- 6.1.1. 6.1.1. Обычная передача данных
- 6.1.2. 6.1.2. Прямой доступ к памяти — DMA
- 6.1.3. 6.1.3. Прямое управление шиной
- 6.1.4. 6.1.4. Регенерация памяти
- 6.1.5. 6.1.5. Прерывания
- 6.1.6. 6.1.6. Шина PC/104
- 6.1.7. 6.1.7. Конфигурирование интерфейсных карт ISA и EISA
- 6.1.8. 6.1.8. Спецификация Plug and Play для шины ISA

### 6.2. Шина PCI

- 6.2.1. 6.2.1. Адресация устройств PCI
- 6.2.2. 6.2.2. Протокол шины PCI
- 6.2.3. 6.2.3. Команды шины, адресация памяти и ввода-вывода<sup>2</sup>
- 6.2.4. 6.2.4. Таймеры, задержки и буферы
- 6.2.5. 6.2.5. Пропускная способность шины
- 6.2.6. 6.2.6. Прерывания
- 6.2.7. 6.2.7. Прямой доступ к памяти, эмуляция ISA DMA (PC/PCI)
- 6.2.8. 6.2.8. Электрический интерфейс, слоты и карты PCI
- 6.2.9. 6.2.9. Другие конструктивы с шиной PCI
- 6.2.10. Мосты PCI

- 6.2.11. Программный доступ к конфигурационному пространству и генерация специальных циклов
- 6.2.12. 6.2.12. Конфигурирование устройств
- 6.2.13. 6.2.13. Классы устройств PCI
- 6.2.14. 6.2.14. PCI BIOS
- 6.2.15. 6.2.15. Разработка собственных устройств PCI
  
- 6.3. 6.3. Магистральный интерфейс AGP
- 6.4. 6.4. Интерфейс LPC
- 6.5. 6.5. Шины и карты расширения блокнотных ПК
  - 6.5.1. Интерфейсы PCMCIA, PC Card и CardBus
  
- Глава 7
  - 7.1. 7.1. Динамическая память
    - 7.1.1. 7.1.1. Асинхронная память — FPM, EDO и BEDO DRAM
    - 7.1.2. 7.1.2. Синхронная память — SDRAM и DDR SDRAM
    - 7.1.3. 7.1.3. Память Rambus DRAM
    - 7.1.4. 7.1.4. Память с виртуальными каналами — VC DRAM
    - 7.1.5. 7.1.5. Модули динамической памяти
  - 7.2. 7.2. Статическая память
  - 7.3. 7.3. Энергонезависимая память
    - 7.3.1. 7.3.1. Постоянная и полупостоянная память — ROM, PROM, EPROM
    - 7.3.2. 7.3.2. EEPROM и флэш-память
    - 7.3.3. 7.3.3. Энергонезависимая память с последовательными интерфейсами
  
- Глава 8
  - 8.1. 8.1. Интерфейс клавиатуры
    - 8.1.1. 8.1.1. Интерфейс клавиатуры AT и PS/2
    - 8.1.2. 8.1.2. Контроллер интерфейса клавиатуры и мыши 8042/8242
    - 8.1.3. 8.1.3. Системная поддержка и программный интерфейс
  - 8.2. 8.2. Интерфейсы мыши
    - 8.2.1. 8.2.1. Последовательные мыши — MS Mouse и PC Mouse
    - 8.2.2. 8.2.2. Мышь PS/2
    - 8.2.3. 8.2.3. Мышь Bus Mouse
  - 8.3. 8.3. Интерфейсы принтеров и плоттеров
    - 8.3.1. 8.3.1. Параллельные интерфейсы — Centronics, IEEE 1284, IPNP
    - 8.3.2. 8.3.2. Последовательные интерфейсы
    - 8.3.3. 8.3.3. Системная поддержка принтера
  - 8.4. 8.4. Интерфейсы графических адаптеров
    - 8.4.1. 8.4.1. Дискретный интерфейс RGB TTL
    - 8.4.2. 8.4.2. Аналоговые интерфейсы RGB
    - 8.4.4. 8.4.4. Цифровые интерфейсы P&D, DVI и DFP
    - 8.4.5. 8.4.5. Внутренние цифровые интерфейсы
    - 8.4.6. 8.4.6. Видеоинтерфейсы
  - 8.5. 8.5. Интерфейсы аудиоустройств
    - 8.5.1. 8.5.1. Аналоговые интерфейсы
    - 8.5.2. 8.5.2. Цифровые интерфейсы
    - 8.5.3. 8.5.3. Интерфейс MIDI
    - 8.5.4. 8.5.4. Интерфейс дочерней карты
  - 8.6. 8.6. Интерфейс игровых устройств — Game-порт

**Глава 9****9.1. Интерфейс НГМД****9.1.1. 9.1.1. Аппаратный интерфейс****9.1.2. 9.1.2. Контроллер НГМД****9.2. Интерфейс АТА/АТАPI (IDE)****9.2.1. 9.2.1. Параллельный интерфейс АТА****9.2.2. 9.2.2. Последовательный интерфейс Serial АТА****9.3. Интерфейсы и конструктивы твердотельных носителей информации****9.3.1. 9.3.1. CompactFlash****9.3.2. 9.3.2. SmartMedia Card****9.3.3. 9.3.3. MultiMediaCard и Secure Digital****9.3.4. 9.3.4. Miniature Card****9.4. Системная поддержка устройств хранения****Глава 10****10.1.10.1. Стандарты и интерфейсы Ethernet****10.2.10.2. Сетевые адаптеры****10.3.10.3. Интерфейс коммутируемой телефонной линии и телефонного аппарата****Глава 11****11.1.1. Шина I2C****11.1.1.1.1.1. Шина I2C****11.1.2.1.1.2. Шина ACCESS.Bus****11.1.3.1.1.3. Шина SMBus****11.1.4.1.1.4. Сравнение шин I2C, ACCESS.bus и SMBus****11.2.11.2. Шина управления SMI интерфейса МП****11.3.11.3. Интерфейс SPI****11.4.11.4. Интерфейс JTAG****11.5.11.5. Программно-управляемая реализация последовательных интерфейсов****Глава 12****12.1.12.1. Пространство физической памяти****12.2.12.2. Пространство ввода-вывода****12.3.12.3. Аппаратные прерывания****12.3.1. Совместное использование прерываний****12.4. Прямой доступ к памяти — DMA****12.4.1. Контроллер прямого доступа 8237A****12.5. Процессоры x86****12.5.1. 12.5.1. Возможности адресации памяти процессорами различных поколений****12.5.2. 12.5.2. Проблемы страничной переадресации****12.5.3. 12.5.3. Инструкции ввода-вывода****12.5.4. 12.5.4. Прерывания****12.6.12.6. Аппаратные средства измерения времени****12.7.12.7. Способы запуска программ****12.8. Сервисы и прерывания BIOS****12.8.1. Int 09h, Int16 h — поддержка клавиатуры**



---

12.8.2. 12.8.2.....	int
10h — видеосервис	
12.8.3. 12.8.3.....	Int
13h — поддержка дисков	
12.8.4. 12.8.4. Int 14h — поддержка COM-портов	
12.8.5. 12.8.5. Int17 h — поддержка принтера	
12.8.6. 12.8.6. Int 1Ah и Int 15h—поддержка таймеров	
12.8.7. 12.8.7. PCI BIOS	
12.9. Расширения ROM BIOS	
12.9.1. 12.9.1. ROM BIOS карт ISA PnP	
12.9.2. 12.9.2. Expansion ROM карт PCI	
Глава 13	
13.1. 13.1. Общие вопросы электропитания и заземления	
13.2. 13.2. Гальваническая развязка	
13.3. 13.3. Интерфейсы блока питания PC	
Литература	
Алфавитный указатель	

## От автора

Книга «Интерфейсы ПК. Справочник», вышедшая следом за первым изданием энциклопедии «Аппаратные средства IBM PC» (1998 г.), была благосклонно принята читателями и даже выпущена «пиратами» на компакт-диске (правда, без указания автора и, естественно, без его уведомления). Работа над вторым изданием энциклопедии породила гору материала, который не помещался в книгу разумного размера, — так созрела идея новой книги об интерфейсах, которая сейчас перед вами. В нее вошли все «обрезки» «слишком большой энциклопедии» и ряд новых материалов. Эта книга адресована специалистам, которые уже знают общее устройство компьютера и имеют представление о взаимодействии его составляющих, но нуждаются в справочной информации для разработки собственной аппаратуры и программного обеспечения, тесно связанного с «железом». Название «Аппаратные интерфейсы ПК» определяет круг освещаемых вопросов — от ножек интерфейсных разъемов до программной модели интерфейсных адаптеров.

Книга начинается с глав, посвященных универсальным внешним интерфейсам, начиная с долгожителей — портов LPT и COM — и кончая современными шинами USB, Fire Wire, SCSI и беспроводными интерфейсами IrDA и Bluetooth. Далее идет «погружение в недра» системного блока ПК — интерфейсы шин расширения с особо детальным описанием шин PCI и ISA, самых интересных на сегодняшний день (шину ISA списывать рано, она еще послужит во встраиваемых компьютерах в обычном виде или в виде PC/104). В главе, посвященной интерфейсам электронной памяти, подробно рассматриваются модули динамической памяти всех современных типов, а также микросхемы статической и энергонезависимой памяти (флэш, EEPROM), с которыми часто приходится иметь дело. Далее в книге описываются специализированные интерфейсы периферийных устройств — клавиатуры, мыши, дисплея (как традиционного, так и плоских панелей), принтеров, аудио- и видеоинтерфейсы, игровой порт. Отдельная глава посвящена интерфейсам устройств хранения — НГМД, АТА (включая новый интерфейс Serial ATA), а также интерфейсам твердотельных устройств хранения (различным флэш-картам). Из интерфейсов компьютерных сетей основное внимание уделяется технологии Ethernet, практически вытеснившей все остальные из локальных сетей и ведущей наступление и в глобальных приложениях. Также рассматривается интерфейс обычной аналоговой телефонной линии, через которую большинство домашних пользователей подключается к Сети. В главе о вспомогательных последовательных интерфейсах рассматриваются все вариации на тему PC, а также интерфейсы SPI, MI и JTAG. Отдельная глава посвящена архитектурному окружению, в котором интерфейсы и их адаптеры (контроллеры) существуют в IBM PC-совместимом компьютере (пространство памяти, пространство ввода-вывода,

прерывания), и нюансам, связанным с различными режимами работы процессоров x86. В этой же главе описываются сервисы и модули расширения BIOS, а также способы загрузки ПО, позволяющие собирать специализированные бездисковые контроллеры на базе универсальных компонентов. Как обычно, книгу завершает тема правильного питания и электробезопасности.

Для удобства восприятия в книге принята система текстовых выделений. *Курсивом* выделены ключевые слова (например, первый раз встречающиеся определения), а также названия *состояний*, \* которых могут пребывать некоторые объекты. В названиях электрических сигналов, например CSO#, символ «#» указывает на инверсность (низкий уровень сигнала отвечает активному состоянию). Названия команд, регистров и битов имеют иной вид — например, INSW (команда процессора), DR (регистр данных), AX (регистр процессора). Подробные оглавление и предметный указатель помогут быстро найти необходимую информацию.

Я благодарен любознательным и внимательным читателям, присылающим свои замечания, вопросы и отзывы о моих книгах. Пользуясь случаем, еще раз обращаюсь к читателям — пишите письма! С вашей помощью исправляются многие ошибки, и я стараюсь поддерживать свои книги на личном сайте по адресу <http://www.neva.ru/mgook>, обновляя списки замеченных опечаток и публикуя статьи на смежные темы.

Как и все предыдущие, эта книга не смогла бы появиться без информационной поддержки коллектива RUSNet (<http://www.neva.ru>), обеспечивающего доступ к Сети в ЦНИИ РТК — «базовом лагере» автора. После выпуска второго издания «Энциклопедии» я снова погрузился в инженерную деятельность в НПО РТК. Это несколько притормаживает работу над книгами, но обогащает практический опыт, что выливается в дополнительные разъяснения актуальных вопросов.

Свои замечания и пожелания присылайте на адрес [mgook@stu.neva.ru](mailto:mgook@stu.neva.ru) (автор) или [comp@piter.com](mailto:comp@piter.com) (издательство «Питер», редакция компьютерной литературы). Информацию по всем книгам можно получить на сайте издательства «Питер» [www.piter.com](http://www.piter.com).

Михаил Гук

# Введение

Толковый словарь по вычислительным системам определяет понятие *интерфейс* (interface) как границу раздела двух систем, устройств или программ; элементы соединения и вспомогательные схемы управления, используемые для соединения устройств. Эта книга посвящена интерфейсам, позволяющим подключать к персональным (и не только) компьютерам разнообразные периферийные устройства (ПУ) и их контроллеры, а также соединять отдельные подсистемы компьютера. Рассмотрим вкратце основные свойства интерфейсов.

По способу передачи информации интерфейсы подразделяются на параллельные и последовательные. В параллельном интерфейсе все биты передаваемого слова (обычно байта) выставляются и передаются по соответствующим параллельно идущим проводам одновременно. В РС традиционно используется параллельный интерфейс Centronics, реализуемый LPT-портами, шины ATA, SCSI и все шины расширения. В последовательном интерфейсе биты передаются друг за другом, обычно по одной (возможно, и двухпроводной) линии. Эта линия может быть как однонаправленной (например, в RS-232C, реализуемой COM-портом, шине Fire Wire, SPI, JTAG), так и двунаправленной (USB, I<sup>2</sup>C).

При рассмотрении интерфейсов важным параметром является пропускная способность. Технический прогресс приводит к неуклонному росту объемов передаваемой информации. Если раньше матричные принтеры, печатающие в символьном режиме, могли обходиться и COM-портом с невысокой пропускной способностью, то современным лазерным принтерам при высоком разрешении не хватает производительности даже самых быстрых LPT-портов. То же касается и сканеров. А передача «живого» видео, даже с применением компрессии, требует ранее немислимой пропускной способности.

Вполне очевидно, что при одинаковом быстродействии приемопередающих цепей и пропускной способности соединительных линий по скорости передачи параллельный интерфейс должен превосходить последовательный. Однако повышение производительности за счет увеличения тактовой частоты передачи данных упирается в волновые свойства соединительных кабелей. В случае параллельного интерфейса начинают сказываться задержки сигналов при их прохождении по линиям кабеля и, что самое неприятное, задержки в разных линиях интерфейса могут быть различными вследствие неидентичности проводов и контактов разъемов. Для надежной передачи данных временные диаграммы обмена строятся с учетом возможного разброса времени прохождения сигналов, что является одним из факторов, сдерживающих рост пропускной способности параллельных интерфейсов. В последовательных интерфейсах, конечно же, есть свои проблемы повышения производительности, но поскольку в них используется меньшее число линий (в пределе — одна), повышение пропускной способности линий связи обходится дешевле.

Для повышения пропускной способности параллельных интерфейсов с середины 90-х годов стали применять двойную синхронизацию DDR (Dual Data Rate). Ее идея заключается в выравнивании частот переключения информационных сигнальных линий и линий стробирования (синхронизации). В «классическом» варианте данные информационных линий воспринимались только по одному перепаду (фронту или спаду) синхросигнала, что удваивает частоту переключения линии синхросигнала относительно линий данных. При двойной синхронизации данные воспринимаются и по фронту, и по спаду, так что частота смены состояний всех линий выравнивается, что при одних и тех же физических параметрах кабеля и интерфейсных схем позволяет удвоить пропускную способность. Волна этих модернизаций началась с интерфейса ATA (режимы UltraDMA) и прошла уже и по SCSI (UltraISO и выше), и по памяти (DDR SDRAM), и по системной шине процессоров (Pentium 4).

Немаловажен для интерфейса контроль достоверности передачи данных, который, увы, имеется далеко не везде. «Ветераном» контроля является шина SCSI с ее битом паритета (который незадачливые пользователи иногда норовят отключить, «чтоб не сбило»); контроль паритета применяется и в последовательных интерфейсах, и в шине PCI. Шина ISA в этом плане беззащитна, как и ее «потомок» — интерфейс ATA, в котором до UltraDMA контроля достоверности не было. В новых интерфейсах контролю достоверности уделяется серьезное внимание, поскольку они, как правило, рассчитываются на экстремальные условия работы (высокие частоты, большие расстояния и помехи). Контроль достоверности может производиться и на более высоких протокольных уровнях (контроль целостности пакетов и их полей), но на аппаратном уровне он работает, естественно, быстрее.

Для интерфейса, соединяющего (физически или логически) два устройства, различают три возможных режима обмена — дуплексный, полудуплексный и симплексный. Дуплексный режим позволяет по одному каналу связи одновременно передавать информацию в обоих направлениях. Он может быть асимметричным, если значения пропускной способности в направлениях «туда» и «обратно» существенно различаются, или симметричным. Полудуплексный режим позволяет передавать информацию «туда» и «обратно» поочередно, при этом интерфейс имеет средства переключения направления канала. Симплексный (односторонний) режим предусматривает только одно направление передачи информации (во встречном направлении передаются только вспомогательные сигналы интерфейса).

Другим немаловажным параметром интерфейса является допустимое удаление соединяемых устройств. Оно ограничивается как частными свойствами кабелей, так и помехозащищенностью интерфейсов. Часть помех возникает от соседних линий интерфейса — это перекрестные помехи, защитой от которых может быть применение витых пар проводов для каждой линии. Другая часть помех вызывается искажением уровней сигналов.

С появлением шин USB и Fire Wire в качестве характеристики интерфейса стала фигурировать и топология соединения. Для интерфейсов RS-232C и Centronics практически всегда применялась двухточечная топология PC — устройство (или PC — PC). Исключениями из этого правила являются различные устройства без-

опасности и защиты данных (Security devices), которые подключаются к COM- или LPT-портам, но имеют разъем для подключения внешнего устройства. Однако эти устройства для традиционной периферии прозрачны, поэтому можно считать, что они не нарушают общего правила. Аналогично обстоит дело и с адаптерами локальных сетей (например, Paraport) и внешних дисковых накопителей (Omega Zip), подключаемых к LPT-портам. Хотя разрабатываемые стандарты для параллельного порта (IEEE 1284.3) и предусматривают соединение устройств в цепочку (Daisy Chain) или через мультиплексоры, широкого распространения такие способы подключения пока не получили. К другому классу исключений относится построение моноканала на COM-портах, которое несколько лет назад применялись в «любительских» локальных сетях, но было вытеснено существенно более эффективной и подешевевшей технологией Ethernet. Интерфейсные шины USB и Fire Wire реализуют древовидную топологию, в которой внешние устройства могут быть как оконечными, так и промежуточными (разветвителями). Эта топология позволяет подключать множество устройств к одному порту USB или Fire Wire.

Важным свойством интерфейса, на которое часто не обращают внимание, является гальваническая развязка, а точнее — ее отсутствие. «Схемные земли» устройств, соединяемых интерфейсом с COM- или LPT-портом PC, оказываются связанными со схемной землей компьютера (а через интерфейсный кабель и между собой). Если между ними до подключения интерфейса была разность потенциалов, то по общему проводу интерфейса потечет уравнивающий ток, что плохо по целому ряду причин. Падение напряжения на общем проводе, вызванное протеканием этого тока, приводит к смещению уровней сигналов, а протекание переменного тока приводит к сложению полезного сигнала с переменной составляющей помехи. К этим помехам особенно чувствительны TTL-интерфейсы; в то же время в RS-232C смещение и помеху в пределах 2 В поглотит зона нечувствительности. В случае обрыва общего провода или плохого контакта, а гораздо чаще — при подключении и отключении интерфейсов без выключения питания устройств, разность потенциалов прикладывается к сигнальным цепям, а протекание уравнивающих токов через них часто приводит к пиротехническим эффектам. Откуда берется эта разность потенциалов, объяснить нетрудно (см. главу 13). Из рассматриваемых в книге интерфейсов гальваническую развязку устройств обеспечивают MIDI, «токовая петля», S/PDIF, шина Fire Wire, сетевые интерфейсы Ethernet.

Существенным свойством является возможность «горячего» подключения/отключения или замены устройств (Hot Swap), причем в двух аспектах. Во-первых, это безопасность переключений «на ходу» как для самих устройств и их интерфейсных схем, так и для целостности хранящихся и передаваемых данных и, наконец, для человека. Во-вторых, это возможность использования вновь подключенных устройств без перезагрузки системы, а также продолжения устойчивой работы системы при отключении устройств. Далеко не все внешние интерфейсы поддерживают «горячее подключение» в полном объеме, так, например, зачастую сканер с интерфейсом SCSI должен быть подключен к компьютеру и включен до загрузки ОС, иначе он не будет доступен системе. С новыми шинами USB и Fire Wire проблем «горячего подключения» не возникает. Для внутренних интерфейсов «горячее подключение» несвойственно. Это касается и шин расширения, и моду-

лей памяти, и даже большинства дисков ATA и SCSI. «Горячее подключение» поддерживается для шин расширения промышленных компьютеров, а также в специальных конструкциях массивов устройств хранения.

#### **ВНИМАНИЕ**-----

Карты расширения, модули памяти и процессоры можно устанавливать и извлекать только при выключенном питании компьютера. При этом выключения блока питания АТХ основной кнопкой недостаточно, поскольку при этом на системной плате остается напряжение 3,3 В. Эти блоки должны обесточиваться по входу (посредством извлечения питающего кабеля).

В ряде интерфейсов заложены возможности PnP (Plug and Play — включай и играй), которые предназначены для снятия с пользователей забот по конфигурированию подключаемых устройств. В современных интерфейсах эти возможности закладывались изначально (PCI, USB, Fire Wire, Bluetooth), и эти функции в большинстве случаев работают нормально. Однако для интерфейсов-ветеранов (например, ISA, SCSI) технология PnP является поздней искусственной надстройкой, работающей с переменным успехом (Plug and Pray — включай и молись). Часто побочные эффекты вызваны наследием «тяжелого прошлого» — соседством устройств PnP с традиционными (legacy) устройствами. На закате шины ISA ее система PnP в общем работала, но в SCSI от идей автоконфигурирования со временем отказались. При разработке собственных устройств встает вопрос выбора подходящего интерфейса подключения. Этот вопрос следует решать, исходя из принципа разумной достаточности, по возможности отдавая предпочтение внешним интерфейсам. Следует помнить, что разработка аппаратной части устройства (hardware) тесно связана и с программной поддержкой устройств — как модулями ПО, исполняемыми процессором компьютера (software), так и программами встроенного микроконтроллера (firmware), на базе которого, как правило, строятся современные устройства. Промышленностью выпускается множество моделей микроконтроллеров, имеющих популярные интерфейсы (USB, RS-232, PC и другие). Однако в ряде случаев приходится использовать и стандартизованные шины расширения ввода-вывода. Эти шины предоставляют более широкие возможности для взаимодействия процессора с аппаратурой, нескованные жесткими ограничениями внешних интерфейсов. Однако за универсальность и производительность внутренних шин расширения приходится расплачиваться более замысловатой реализацией интерфейсных схем и сложностями при обеспечении совместимости с другим установленным в компьютер оборудованием. Здесь ошибки могут приводить к потере работоспособности компьютера (хорошо если временной). Недаром серьезные производители компьютеров гарантируют работоспособность своих изделий только при установке сертифицированных (ими или независимыми лабораториями) карт расширения. При использовании внешних интерфейсов неприятности в случае ошибок чаще всего имеют отношение только к подключаемому устройству. На этом мы закончим краткий обзор интерфейсов и перейдем к их детальным описаниям.

## Глава 1

# Параллельный интерфейс — LPT-порт

Порт параллельного интерфейса был введен в PC для подключения принтера — отсюда и пошло его название *LPT-порт* (Line PrinTer — построчный принтер). Традиционный, он же стандартный, LPT-порт (так называемый *SPP-порт*) ориентирован на вывод данных, хотя с некоторыми ограничениями позволяет и вводить данные. Существуют различные модификации LPT-порта — двунаправленный, EPP, ECP и другие, расширяющие его функциональные возможности, повышающие производительность и снижающие нагрузку на процессор. Поначалу они являлись фирменными решениями отдельных производителей, позднее был принят стандарт IEEE 1284.

С внешней стороны порт имеет 8-битную шину данных, 5-битную шину сигналов состояния и 4-битную шину управляющих сигналов, выведенные на разъем-розетку DB-25S. В LPT-порте используются логические уровни ТТЛ, что ограничивает допустимую длину кабеля из-за невысокой помехозащищенности ТТЛ-интерфейса. Гальваническая развязка отсутствует — схемная земля подключаемого устройства соединяется со схемной землей компьютера. Из-за этого порт является уязвимым местом компьютера, страдающим при нарушении правил подключения и заземления устройств. Поскольку порт обычно располагается на системной плате, в случае его «выжигания» зачастую выходит из строя и его ближайшее окружение, вплоть до выгорания всей системной платы.

С программной стороны LPT-порт представляет собой набор регистров, расположенных в пространстве ввода-вывода. Регистры порта адресуются относительно базового адреса порта, стандартными значениями которого являются 3BCh, 378h и 278h. Порт может использовать линию запроса аппаратного прерывания, обычно IRQ7 или IRQ5. В расширенных режимах может использоваться и канал DMA.

Порт имеет поддержку на уровне BIOS — поиск установленных портов во время теста POST и сервисы печати Int 17h (см. п. 8.3.3) обеспечивают вывод символа (по опросу готовности, не используя аппаратных прерываний), инициализацию интерфейса и принтера, а также опрос состояния принтера.

Практически все современные системные платы (еще начиная с PCI-плат для процессоров 486) имеют встроенный адаптер LPT-порта. Существуют карты ISA с LPT-портом, где он чаще всего соседствует с парой COM-портов, а также с контроллерами



дисковых интерфейсов (FDC+IDE). LPT-порт обычно присутствует и на плате дисплейного адаптера MDA (монохромный текстовый) и HGC (монохромный графический «Геркулес»). Есть и карты PCI с дополнительными LPT-портами. К LPT-портам подключают принтеры, плоттеры, сканеры, коммуникационные устройства и устройства хранения данных, а также электронные ключи, программаторы и прочие устройства. Иногда параллельный интерфейс используют для связи между двумя компьютерами — получается сеть, «сделанная наколенке» (LapLink).

## 1.1. Традиционный LPT-порт

Традиционный, он же стандартный, LPT-порт называется *стандартным параллельным портом* (Standard Parallel Port, SPP), или *SPP-портом*, и является однонаправленным портом, через который программно реализуется протокол обмена *Centronics* (см. п. 8.3.1). Название и назначение сигналов разъема порта (табл. 1.1) соответствуют интерфейсу Centronics.

Таблица 1.1. Разъем стандартного LPT-порта

Контакт DB-25S	№ провода в кабеле	Назначение I/O <sup>1</sup>	Бит <sup>2</sup>	Сигнал
1	1	0/I	CR.OX	Strobe»
2	3	0(1)	DR.0	Data0
3	5	0(1)	DR.1	Data1
4	7	0(1)	DR.2	Data 2
5	9	0(1)	DR.3	Data 3
6	11	0(1)	DR.4	Data 4
7	13	0(0)	DR.5	Data 5
8	15	0(0)	DR.6	Data 6
9	17	0(0)	DR.7	Data 7
10	19	I <sup>3</sup>	SR.6	Ack#
11	21	I	SR.A	Busy
12	23	I	SR.5	PaperEnd(PE)
13	25	I	SR.4	Select
14	2	0/I	CR.1\	Auto LF# (AutoFeedW)
15	4	I	SR.3	Error»
16	6	0/I	CR.2	Init#
17	8	0/I	CR.3\	Select In#
18-25	10,12,14,16,18, 20, 22, 24, 26	—	—	—

<sup>1</sup> I/O задает направление передачи (вход-выход) сигнала порта. 0/I обозначает выходные линии, состояние которых считывается при чтении из портов вывода; б(1) — выходные линии, состояние которых может быть считано только при особых условиях (см. ниже).

<sup>2</sup> Символом «\» отмечены инвертированные сигналы (1 в регистре соответствует низкому уровню линии).

<sup>3</sup> Вход Ack# соединен резистором (10 кОм) с питанием +5 В.

Адаптер SPP-порта содержит три 8-битных регистра, расположенных по соседним адресам в пространстве ввода-вывода, начиная с базового адреса порта BASE (3BCh, 378h или 278h).

*Data Register (DR)* — *регистр данных*, адрес=BA5E. Данные, записанные в этот регистр, *выводятся* на выходные линии Data[7:0]. Данные, считанные из этого регистра, в зависимости от схемотехники адаптера соответствуют либо ранее записанным данным, либо сигналам на тех же линиях, что не всегда одно и то же. *Status Register (SR)* — *регистр состояния* (только чтение), адрес=BA5E+1. Регистр отображает *5-битный порт ввода* сигналов состояния принтера (биты SR.4-SR.7) и флаг прерывания. Бит SR. 7 инвертируется — низкому уровню сигнала соответствует единичное значение бита в регистре, и наоборот.

Ниже описано назначение бит регистра состояния (в скобках даны номера контактов разъема порта).

- ◆ SR.7 — Busy — инверсное отображение состояния линии Busy (11): при низком уровне на линии устанавливается единичное значения бита — разрешение на вывод очередного байта.
- ◆ SR. 6 — Ack (Acknowledge) — отображение состояния линии Ack# (10).
- ◆ SR. 5 — PE (Paper End) — отображение состояния линии Paper End (12). Единичное значение соответствует высокому уровню линии — сигналу о конце бумаги в принтере.
- ◆ SR. 4 — Select — отображение состояния линии Select (13). Единичное значение соответствует высокому уровню линии — сигналу о включении принтера.
- ◆ SR. 3 — Error — отображение состояния линии Error\* (15). Нулевое значение соответствует низкому уровню линии — сигналу о любой ошибке принтера.
- ◆ SR.2 — PIRQ — флаг прерывания по сигналу Ack# (только для порта PS/2). Бит обнуляется, если сигнал Ack# вызвал аппаратное прерывание. Единичное значение устанавливается по аппаратному сбросу и после чтения регистра состояния.
- ◆ SR[1:0] — зарезервированы.

*Control Register (CR)* — *регистр управления*, адрес=BA5E+2, допускает запись и чтение. Регистр связан с *4-битным портом вывода управляющих сигналов* (биты 0-3) для которых возможно и чтение; выходной буфер обычно имеет тип «открытый коллектор». Это позволяет корректно использовать линии данного регистра как входные при программировании их в высокий уровень. Биты 0,1,3 инвертируются.

Ниже описано назначение бит регистра управления.

- ◆ CR [7:6] — зарезервированы.
- ◆ CR.5 — Direction — бит управления направлением передачи (только для портов PS/2, см. ниже). Запись единицы переводит порт данных в режим ввода. При чтении состояние бита не определено.
- ◆ CR. 4 — AckINTEN (Ack Interrupt Enable) — единичное значение разрешает прерывание по спаду сигнала на линии Ack# — сигнал запроса следующего байта.

- ◆ CR.3 — Select In — единичное значение бита соответствует низкому уровню на выходе Select In# (17) — сигналу, разрешающему работу принтера по интерфейсу *Centronics*.
- ◆ CR.2 — In it — нулевое значение бита соответствует низкому уровню на выходе Init# (16) — сигнал аппаратного сброса принтера.
- ◆ CR. 1 — Auto LF — единичное значение бита соответствует низкому уровню на выходе Auto LF# (14) — сигналу на автоматический перевод строки (LF — Line Feed) по приему байта возврата каретки (CR). Иногда сигнал и бит называют AutoFD или AutoFDXT.
- ◆ CR.0 — Strobe — единичное значение бита соответствует низкому уровню на выходе Strobe\* (1) — сигналу стробирования выходных данных.

*Запрос аппаратного прерывания* (обычно IRQ7 или IRQ5) вырабатывается по отрицательному перепаду сигнала на выводе 10 разъема интерфейса (Ack#) при установке CR. 4=1. Во избежание ложных прерываний контакт 10 соединен резистором с шиной +5 В. Прерывание вырабатывается, когда принтер подтверждает прием предыдущего байта. Как уже было сказано, BIOS это прерывание не использует и не обслуживает.

Перечислим шаги процедуры вывода байта по интерфейсу Centronics с указанием требуемого количества шинных операций процессора.

1. Вывод байта в регистр данных (1 цикл IOWR#).
2. Ввод из регистра состояния и проверка готовности устройства (бит SR. 7 — сигнал Busy). Этот шаг зацикливается до получения готовности или до срабатывания программного тайм-аута (минимум 1 цикл IORD#).
3. По получению готовности выводом в регистр управления устанавливается строб данных, а следующим выводом строб снимается. Обычно, чтобы переключить только один бит (строб), регистр управления предварительно считывается, что к двум циклам IOWR# добавляет еще один цикл IORD#.

Видно, что для вывода одного байта требуется 4-5 операций ввода-вывода с регистрами порта (в лучшем случае, когда готовность обнаружена по первому чтению регистра состояния). Отсюда вытекает главный недостаток вывода через стандартный порт — невысокая скорость обмена при значительной загрузке процессора. Порт удается разогнать до скоростей 100-150 Кбайт/с при полной загрузке процессора, что недостаточно для печати на лазерном принтере. Другой недостаток функциональный — сложность использования в качестве порта ввода.

Стандартный порт асимметричен — при наличии 12 линий (и бит), нормально работающих на вывод, на ввод работает только 5 линий состояния. Если необходима симметричная двунаправленная связь, на всех стандартных портах работоспособен *режим полубайтного обмена* — *Nibble Mode*. В этом режиме, называемом также *Hewlett Packard Bi-tronics*, одновременно принимаются 4 бита данных, пятая линия используется для квитирования. Таким образом, каждый байт передается за два цикла, а каждый цикл требует по крайней мере 5 операций ввода-вывода.

Схемотехника выходных буферов данных LPT-портов отличается большим разнообразием. На многих старых моделях адаптеров SPP-порт данных можно использовать и для организации ввода. Если в порт данных записать байт с единицами во всех разрядах, а на выходные линии интерфейса через микросхемы с выходом типа «открытый коллектор» подать какой-либо код (или соединить ключами какие-то линии со схемной землей), то этот код может быть считан из того же регистра данных. Однако выходным цепям передатчика информации придется «бороться» с выходным током логической единицы выходных буферов адаптера. Схемотехника TTL такие решения не запрещает, но если внешнее устройство выполнено на микросхемах КМОП, их мощности может не хватить для «победы» в этом шинном конфликте. Однако современные адаптеры часто имеют в выходной цепи согласующий резистор с сопротивлением до 50 Ом. Выходной ток короткого замыкания выхода на землю обычно не превышает 30 мА. Простой расчет показывает, что даже в случае короткого замыкания контакта разъема на землю при выводе «единицы» на этом резисторе падает напряжение 1,5 В, что входной схемой приемника будет воспринято как «единица». Поэтому нельзя полагать, что такой способ ввода будет работать на всех компьютерах. На некоторых старых адаптерах портов выходной буфер отключается перемычкой на плате. Тогда порт превращается в обыкновенный порт ввода.

## 1.2. Расширения параллельного порта

Недостатки стандартного порта частично устраняли новые типы портов, появившиеся в компьютерах PS/2.

*Двухнаправленный порт 1 (Type 1 parallel port)* — интерфейс, введенный в PS/2. Такой порт кроме стандартного режима может работать в режиме ввода или двухнаправленном режиме. Протокол обмена формируется программно, а для указания направления передачи в регистр управления порта введен специальный бит CR. 5: 0 — буфер данных работает на вывод, 1 — на ввод. Не путайте этот порт, называемый также *enhanced bi-directional*, с EPP. Данный тип порта «прижился» и в обычных компьютерах, в CMOS Setup он может называться PS/2 или Bi-Di. *Порт с прямым доступом к памяти (Type 3 DMA parallel port)* применялся в PS/2 моделях 57,90,95. Был введен для повышения пропускной способности и разгрузки процессора при выводе на принтер. Программе, работающей с портом, требовалось только задать в памяти блок данных, подлежащих выводу, а затем вывод по протоколу Centronics производился без участия процессора.

Позже появились другие адаптеры LPT-портов\* реализующие протокол обмена Centronics аппаратно, — *Fast Centronics*. Некоторые из них использовали FIFO-буфер данных — *Parallel Port FIFO Mode*. Не будучи стандартизованными, такие порты разных производителей требовали наличия собственных специальных драйверов. Программы, использующие прямое управление регистрами стандартных портов, не могли задействовать их дополнительные возможности. Такие порты часто входили в состав мультикарт VLB. Существуют их варианты с шиной ISA, а также встроенные в системную плату.

## 1.3. Стандарт IEEE 1284

Стандарт на параллельный интерфейс *IEEE 1284*, принятый в 1994 году, описывает порты *SPP*, *EPP* и *ECP*. Стандарт определяет 5 режимов обмена данными, метод согласования режима, физический и электрический интерфейсы. Согласно IEEE 1284, возможны следующие режимы обмена данными через параллельный порт:

- ◆ *Режим совместимости (Compatibility Mode)* — однонаправленный (вывод) по протоколу Centronics. Этот режим соответствует SPP-порту.
- ◆ *Полубайтный режим (Nibble Mode)* — ввод байта в два цикла (по 4 бита), используя для приема линии состояния. Этот режим обмена подходит для любых адаптеров, поскольку задействует только возможности стандартного порта.
- ◆ *Байтный режим (Byte Mode)* — ввод байта целиком, используя для приема линии данных. Этот режим работает только на портах, допускающих чтение выходных данных (*Bi-Directional* или *PS/2 Type 1*, см. выше).
- ◆ *Режим EPP (EPP Mode)* — двунаправленный обмен данными (*EPP* означает Enhanced Parallel Port). Управляющие сигналы интерфейса генерируются аппаратно во время цикла обращения к порту. Эффективен при работе с устройствами внешней памяти и адаптерами локальных сетей.
- ◆ *Режим ECP (ECP Mode)* — двунаправленный обмен данными с возможностью аппаратного сжатия данных по методу *RLE* (Run Length Encoding) и использования FIFO-буферов и DMA (*ECP* означает Extended Capability Port). Управляющие сигналы интерфейса генерируются аппаратно. Эффективен для принтеров и сканеров (здесь может использоваться сжатие) и различных устройств блочного обмена.

Стандарт определяет способ, по которому ПО может определить режим, доступный и хосту (PC), и периферийному устройству (или присоединенному второму компьютеру). Режимы нестандартных портов, реализующих протокол обмена Centronics аппаратно (*Fast Centronics, Parallel Port FIFO Mode*), могут и не являться режимами IEEE 1284, несмотря на наличие в них черт EPP и ECP.

В компьютерах с LPT-портом на системной плате режим — SPP, EPP, ECP или их комбинация — задается в BIOS Setup. Режим совместимости полностью соответствует SPP-порту. Остальные режимы подробно рассмотрены ниже.

При описании режимов обмена фигурируют следующие понятия:

- ◆ *хост* — компьютер, обладающий параллельным портом;
- ◆ *ПУ* — периферийное устройство, подключаемое к этому порту;
- ◆ *Ptr* — в названиях сигналов обозначает передающее ПУ;
- ◆ *прямой канал* — канал вывода данных от хоста в ПУ;
- ◆ *обратный канал* — канал ввода данных в хост из ПУ.

### 1.3.1. Полубайтный режим ввода — Nibble Mode

Полубайтный режим предназначен для двунаправленного обмена и может работать на всех стандартных портах. Порты имеют 5 *линий ввода* состояния, используя которые ПУ может посылать в хост байт тетрадами (nibble — полубайт, 4 бита) за два приема. Сигнал Ask#, вызывающий прерывание, которое может использоваться в данном режиме, соответствует биту 6 регистра состояния, что усложняет программные манипуляции с битами при сборке байта. Сигналы порта приведены в табл. 1.2, временные диаграммы — на рис. 1.1.

Таблица 1.2. Сигналы LPT-порта в полубайтном режиме ввода

Контакт	Сигнал SPP	I/O	Бит	Описание
14	AutoFeed#	0	CR.1	HostBusy — сигнал квитирования. Низкий уровень означает готовность к приему тетрады, высокий подтверждает прием тетрады
17	SelectIn*	0	CR.3	Высокий уровень указывает на обмен в режиме IEEE 1284 (в режиме SPP уровень низкий)
10	Ack#	I	SR.6	PtrClk. Низкий уровень означает готовность тетрады, высокий — ответ на сигнал HostBusy
11	Busy PE	II	SR.7	Прием бита данных 3, затем бита 7 Прием
12	Select	1	SR.5	бита данных 2, затем бита 6 Прием бита
13	Error#	1	SR.4	данных 1, затем бита 5 Прием бита данных
15			SR.3	0, затем бита 4

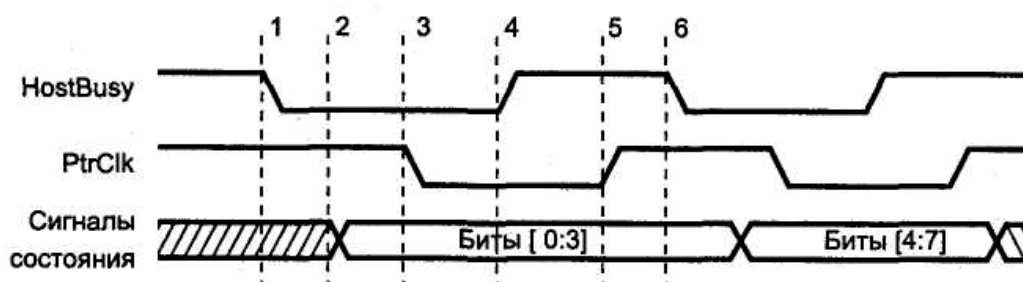


Рис. 1.1. Прием данных в полубайтном режиме Прием

байта данных в полубайтном режиме состоит из следующих фаз:

1. Хост сигнализирует о готовности приема данных установкой низкого уровня на линии HostBusy.
2. ПУ в ответ помещает тетраду на входные линии состояния.
3. ПУ сигнализирует о готовности тетрады установкой низкого уровня на линии PtrClk.
4. Хост устанавливает высокий уровень на линии HostBusy, указывая на занятость приемом и обработкой тетрады.

5. ПУ отвечает установкой высокого уровня на линии PtrClk.
6. Шаги 1-5 повторяются для второй тетрады.

Полубайтный режим сильно нагружает процессор, и поднять скорость обмена выше 50 Кбайт/с не удастся. Безусловное его преимущество в том, что он работает *на всех портах*. Его применяют в тех случаях, когда поток данных невелик (например, для связи с принтерами). Однако при связи с адаптерами локальных сетей, внешними дисковыми накопителями и CD-ROM прием больших объемов данных требует изрядного терпения со стороны пользователя.

### 1.3.2. Двухнаправленный байтный режим — Byte Mode

В этом режиме данные принимаются с использованием двухнаправленного порта, у которого выходной буфер данных может отключаться установкой бита CR.5=1. Как и предыдущие, режим является программно-управляемым — все сигналы квитирования анализируются и устанавливаются драйвером. Сигналы порта описаны в табл. 1.3, временные диаграммы — на рис. 1.2.

Таблица 1.3. Сигналы LPT-порта в байтном режиме ввода-вывода

Контакт	Сигнал SPP	Имя в байтном I/O режиме	Бит	Описание
1	Strobe*	HostClk	O	CR.0\ Импульс (низкого уровня) подтверждает прием байта в конце каждого цикла
14	AutoFeed*	HostBusy	O	CR.1\ Сигнал квитирования. Низкий уровень означает готовность хоста принять байт; высокий уровень устанавливается по приему байта
17	SelectIn*	1284Active	O	CR.3\ Высокий уровень указывает на обмен в режиме IEEE 1284 (в режиме SPP уровень низкий)
16	Init#	Init#	0	CR.2 Не используется; установлен высокий уровень
10	Ack#	PtrClk	1	SR.6 Устанавливается в низкий уровень для индикации действительности данных на линиях Data [0:7]. В низкий уровень устанавливается в ответ на сигнал HostBusy
11	Busy	PtrBusy	I	SR-Д Состояние занятости прямого канала
12	PE	AckDataReq <sup>1</sup>	I	SR.5 Устанавливается ПУ для указания на наличие обратного канала передачи
13	Select	Xflag <sup>1</sup>	I	SR.4 Флаг расширяемости
15	Error*	DataAvail* <sup>1</sup>	I	SR.3 Устанавливается ПУ для указания на наличие обратного канала передачи
2-9	Data [0:7]	Data [0:7]	I/O	DR[0:7] Двухнаправленный (прямой и обратный) канал данных

Сигналы действуют в последовательности согласования (см. ниже).

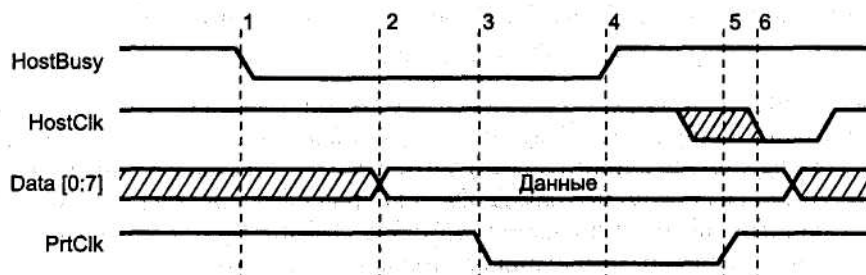


Рис. 1.2. Прием данных в байтном режиме

Фазы приема байта данных перечислены ниже.

1. Хост сигнализирует о готовности приема данных установкой низкого уровня на линии HostBusy.
2. ПУ в ответ помещает байт данных на линии Data [0:7].
3. ПУ сигнализирует о действительности байта установкой низкого уровня на линии PtrClk.
4. Хост устанавливает высокий уровень на линии HostBusy, указывая на занятость приемом и обработкой байта.
5. ПУ отвечает установкой высокого уровня на линии PtrClk,
6. Хост подтверждает прием байта импульсом HostClk.

Шаги 1-6 повторяются для каждого следующего байта. Квитирование осуществляется парой сигналов HostBusy и PtrClk; ПУ может и не использовать сигнал HostClk (это приглашение к выдаче следующего байта, напоминающее сигнал Ack# в интерфейсе Centronics). Побайтный режим позволяет поднять скорость обратного канала до скорости прямого канала в стандартном режиме. Однако он способен работать *только на двунаправленных портах*, которые раньше применялись в основном на малораспространенных машинах PS/2, но практически все современные порты можно сконфигурировать на двунаправленный режим (в настройках BIOS Setup - Bi-Di или PS/2).

### 1.3.3. Режим EPP

Протокол *EPP* (Enhanced Parallel Port — улучшенный параллельный порт) был разработан компаниями Intel, Xircom и Zenith Data Systems задолго до принятия стандарта IEEE 1284. Этот протокол предназначен для повышения производительности обмена по параллельному порту, впервые был реализован в чипсете Intel 386SL (микросхема 82360) и впоследствии принят множеством компаний как дополнительный протокол параллельного порта. Версии протокола, реализованные до принятия IEEE 1284, отличаются от нынешнего стандарта (см. ниже). Протокол EPP обеспечивает четыре типа циклов обмена:

- ◆ запись данных;
- ◆ чтение данных;



запись адреса;

чтение адреса.

Назначение циклов записи и чтения данных очевидно. Адресные циклы используются для передачи адресной, канальной и управляющей информации. Циклы обмена данными отличаются от адресных циклов применяемыми стробирующими сигналами. Назначение сигналов порта EPP и их связь с сигналами SPP объясняются в табл. 1.4.

Таблица 1.4. Сигналы LPT-порта в режиме ввода-вывода EPP

Контакт	Сигнал SPP	Имя в EPP	I/O	Описание
1	Strobe»	Write*	0	Низкий уровень — цикл записи, высокий — цикл чтения
14	AutoLF#	DataStb*	0	Строб данных. Низкий уровень устанавливается в циклах передачи данных
17	SelectIn*	AddrStb#	0	Строб адреса. Низкий уровень устанавливается в адресных циклах
16	Init#	Reset»	0	Сброс ПУ (низким уровнем)
10	Ack#	INTR#	1	Прерывание от ПУ
11	Busy	Wait#	1	Сигнал квитирования. Низкий уровень разрешает начало цикла (установку строга в низкий уровень), переход в высокий — разрешает завершение цикла (снятие строга)
2-9	Data [0:7]	AD[0:7]	I/o	Двухнаправленная шина адреса/данных
12	PaperEnd	AckDataReq <sup>1</sup>	1	Используется по усмотрению разработчика периферии
13	Select	Xflag'	1	Используется по усмотрению разработчика периферии
15	Error*	DataAvaiW <sup>1</sup>	1	Используется по усмотрению разработчика периферии

<sup>1</sup> Сигналы действуют в последовательности согласования (см. ниже).

EPP-порт имеет расширенный набор регистров (табл. 1.5), который занимает в пространстве ввода-вывода 5-8 смежных байт.

Таблица 1.5. Регистры EPP-порта

Имя регистра	Смещение	Режим	R/W	Описание
SPP Data Port	+0	SPP/EPP	W	Регистр данных SPP
SPP Status Port	+1	SPP/EPP	R	Регистр состояния SPP
SPP Control Port	+2	SPP/EPP	W	Регистр управления SPP
EPP Address Port	+3	EPP	R/W	Регистр адреса EPP. Чтение или запись в него генерирует связанный цикл чтения или записи адреса EPP
EPP Data Port	+4	EPP	R/W	Регистр данных EPP. Чтение (запись) генерирует связанный цикл чтения (записи) данных EPP

Имя регистра	Смещение	Режим	R/W	Описание
Not Defined	+5...+7	EPP	N/A	В некоторых контроллерах могут использоваться для 16-32-битных операций ввода-вывода

В отличие от программно-управляемых режимов, описанных выше, внешние сигналы EPP-порта для каждого цикла обмена формируются аппаратно по одной операции записи или чтения в регистр порта. На рис. 1.3 приведена диаграмма цикла записи данных, иллюстрирующая внешний цикл обмена, вложенный в цикл записи системной шины процессора (иногда эти циклы называют *связанными*). Адресный цикл записи отличается от цикла данных только стробом внешнего интерфейса.

Цикл записи данных состоит из следующих фаз.

Программа выполняет цикл вывода (IOWR\*) в порт 4 (EPP Data Port).

Адаптер устанавливает сигнал Write\* (низкий уровень), и данные помещаются на выходную шину LPT-порта.

При низком уровне Wait\* устанавливается строб данных.

Порт ждет подтверждения от ПУ (перевода Wait\* в высокий уровень).

Снимается строб данных — внешний EPP-цикл завершается.

Завершается процессорный цикл вывода.

ПУ устанавливает низкий уровень Wait\*, указывая на возможность начала следующего цикла.

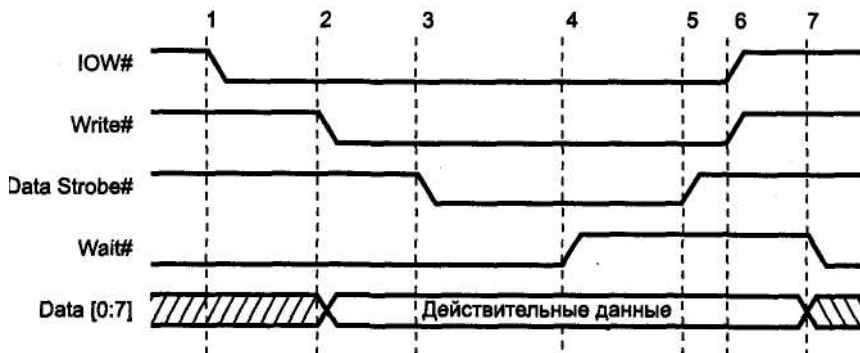


Рис. 1.3. Цикл записи данных EPP

Пример адресного цикла чтения приведен на рис. 1.4. Цикл чтения данных отличается только применением другого стробирующего сигнала.

Главной отличительной чертой EPP является выполнение внешней передачи во время одного процессорного цикла ввода-вывода. Это позволяет достигать высоких скоростей обмена (0,5-2 Мбайт/с). ПУ, подключенное к параллельному порту EPP, может работать со скоростью устройства, подключаемого через слот ISA

Протокол блокированного квитирования (*interlocked handshakes*) позволяет автоматически настраиваться на скорость обмена, доступную и хосту, и ПУ. ПУ может регулировать длительность всех фаз обмена с помощью всего лишь одного сигнала Wait\*. Протокол автоматически подстраивается под длину кабеля — вносимые задержки приведут только к удлинению цикла. Поскольку кабели, соответствующие стандарту IEEE 1284 (см. выше), имеют одинаковые волновые свойства для разных линий, нарушения передачи, связанного с «согласованиями» сигналов, происходить не должно. При подключении сетевых адаптеров или внешних дисков к EPP-порту можно наблюдать непривычное явление: снижение производительности по мере удлинения интерфейсного кабеля.

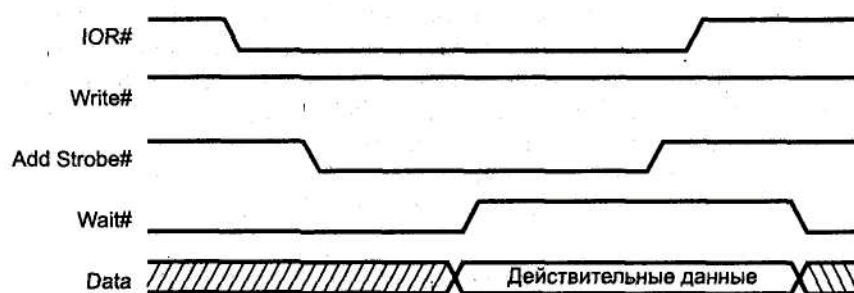


Рис. 1.4. Адресный цикл чтения EPP

Естественно, ПУ не должно «подвешивать» процессор на шинном цикле обмена. Это гарантирует механизм тайм-аутов PC, который принудительно завершает любой цикл обмена, длящийся более 15 мкс. В ряде реализаций EPP за тайм-аутом интерфейса следит сам адаптер — если ПУ не отвечает в течение определенного времени (5 мкс), цикл прекращается и в дополнительном (не стандартизованном) регистре состояния адаптера фиксируется ошибка.

Устройства с интерфейсом EPP, разработанные до принятия IEEE 1284, отличаются началом цикла: строб DataStb\* или AddrStb\* устанавливается независимо от состояния WAIT\*. Это означает, что ПУ не может задержать начало следующего цикла (хотя может растянуть его на требуемое время). Такая спецификация называется *EPP 1.7* (предложена Xigcom). Именно она применялась в контроллере 82360. Периферия, совместимая с IEEE 1284 EPP, будет нормально работать с контроллером EPP 1.7, но ПУ в стандарте EPP 1.7 может отказаться работать с контроллером EPP 1284.

С *программной* точки зрения контроллер EPP-порта выглядит просто (см. табл. 1.5). К трем регистрам стандартного порта, имеющим смещение 0, 1 и 2 относительно базового адреса порта, добавлены два регистра (EPP Address Port и EPP Data Port), чтение и запись в которые вызывает генерацию связанных внешних циклов.

Назначение регистров стандартного порта сохранено для совместимости EPP-порта с ПУ и ПО, рассчитанными на применение программно-управляемого обмена. Поскольку сигналы квитирования адаптером вырабатываются аппарат-

но, при записи в регистр управления CR биты 0, 1 и 3, соответствующие сигналам Strobe\*, AutoFeed\* и Selecting должны иметь нулевые значения. Программное вмешательство могло бы нарушить последовательность квитирования. Некоторые адаптеры имеют специальные средства защиты (*EPP Protect*), при включении которых программная модификация этих бит блокируется.

Использование регистра данных EPP позволяет осуществлять передачу блока данных с помощью одной инструкции REP INSB или REP OUTSB. Некоторые адаптеры допускают *16/32-битное обращение* к регистру данных EPP. При этом адаптер просто дешифрует адрес со смещением в диапазоне 4-7 как адрес регистра данных EPP, но процессору сообщает о разрядности 8 бит. Тогда 16- или 32-битное обращение по адресу регистра данных EPP приведет к автоматической генерации двух или четырех шинных циклов по нарастающим адресам, начиная со смещения 4. Эти циклы будут выполняться быстрее, чем то же количество одиночных циклов. Более «продвинутые» адаптеры для адреса регистра данных EPP сообщают разрядность 32 бит и для них до 4 байт может быть передано за один цикл обращения процессора. Таким образом обеспечивается производительность до 2 Мбайт/с, достаточная для адаптеров локальных сетей, внешних дисков, стриммеров и CD-ROM. Адресные циклы EPP всегда выполняются только в однобайтном режиме.

Важной чертой EPP является то, что обращение процессора к ПУ осуществляется в реальном времени — нет буферизации. Драйвер способен отслеживать состояние и подавать команды в точно известные моменты времени. Циклы чтения и записи могут чередоваться в произвольном порядке или идти блоками. Такой тип обмена удобен для *регістро-орієнтованих ПУ* или ПУ, работающих в *реальном времени*, например устройств сбора информации и управления. Этот режим пригоден и для устройств хранения данных, сетевых адаптеров, принтеров, сканеров и т. п.

К сожалению, режим EPP поддерживается не всеми портами — он отсутствует, к примеру, в ряде блокнотных ПК. Так что при разработке собственных устройств ради большей совместимости с компьютерами приходится ориентироваться на режим ECP.

### 1.3.4. Режим ECP

Протокол *ECP* (Extended Capability Port — порт с расширенными возможностями) был предложен Hewlett Packard и Microsoft для связи с ПУ типа принтеров или сканеров. Как и EPP, данный протокол обеспечивает высокопроизводительный двунаправленный обмен данными хоста с ПУ.

Протокол ECP в обоих направлениях обеспечивает два типа циклов:

- ◆ циклы записи и чтения данных;
- ◆ командные циклы записи и чтения.

Командные циклы подразделяются на два типа: передача канальных адресов и передача счетчика *RLC* (Run-Length Count).

В отличие от EPP вместе с протоколом ECP сразу появился стандарт на программную (регистровую) модель его адаптера, изложенный в документе «The IEEE 1284 Extended Capabilities Port Protocol and ISA Interface Standard» компании Microsoft. Этот документ определяет свойства протокола, не заданные стандартом IEEE 1284:

- ◆ компрессия данных хост-адаптером по методу RLE;
- ◆ буферизация FIFO для прямого и обратного каналов;
- ◆ применение DMA и программного ввода-вывода.

Компрессия в реальном времени по методу *RLE* (Run-Length Encoding) позволяет достичь коэффициента сжатия 64:1 при передаче растровых изображений, которые имеют длинные строки повторяющихся байт. Компрессию можно использовать, только если ее поддерживают и хост, и ПУ.

*Канальная адресация ECP* применяется для адресации множества логических устройств, входящих в одно физическое. Например, в комбинированном устройстве факс/принтер/модем, подключаемом только к одному параллельному порту, возможен одновременный прием факса и печать на принтере. В режиме SPP, если принтер установит сигнал занятости, канал будет занят данными, пока принтер их не примет. В режиме ECP программный драйвер просто адресует к другому логическому каналу того же порта.

Протокол ECP переопределяет сигналы SPP (табл. 1.6).

Таблица 1.6. Сигналы LPT-порта в режиме ввода-

вывода ECP

Контакт	Сигнал SPP	Имя в ECP	I/O	Описание
1	Strobe#	HostClk HostAck	O	Строб данных, используется в паре с PeriphAck для передачи в прямом направлении (вывод)
1	AutoLF		O	Указывает тип цикла (команда/данные) при передаче в прямом направлении. Используется как сигнал подтверждения в паре с PeriphClk для передачи в обратном направлении
17	SelectIn*	1284Active	O	Высокий уровень указывает на обмен в режиме IEEE 1284 (в режиме SPP уровень низкий)
16	Init#	ReverseRequest#	O	Запрос реверса. Низкий уровень сигнализирует о переключении канала на передачу в обратном направлении
10	Ack#	PeriphClk	I	Строб данных, используется в паре с HostAck для передачи в обратном направлении
11	Busy	PeriphAck	I	Используется как сигнал подтверждения в паре с HostClk для передачи в прямом направлении.
12	PaperEnd	AckReverse*	I	Индицирует тип команда/ данные при передаче в обратном направлении
13	Select	Xflag <sup>1</sup>	I	Подтверждение реверса. Переводится в низкий уровень в ответ на ReverseRequest* Флаг расширяемости

### 1.3. Стандарт IEEE 1284

Контакт	Сигнал SPP	Имя в ECP	I/O	Описание
15	Error*	PeriphRequest* <sup>1</sup>	I	Устанавливается ПУ для указания на доступность (наличие) обратного канала передачи <sup>1</sup>
2-9	Data [0:7]	Data [0:7]	I/O	Двухнаправленный канал данных

<sup>1</sup> Сигналы действуют в последовательности согласования (см. ниже).

Адаптер ECP тоже генерирует внешние протокольные сигналы квитирования аппаратно, но его работа существенно отличается от режима EPP.

На рис. 1.5, а приведена диаграмма двух циклов прямой передачи: за циклом данных следует командный цикл. *Тип цикла* задается уровнем на линии HostAck: в цикле данных — высокий, в командном цикле — низкий. В командном цикле байт может содержать *канальный адрес* или счетчик RLE. Отличительным признаком является бит 7 (старший): если он нулевой, то биты 0-6 содержат счетчик RLE (0-127), если единичный — то канальный адрес. На рис. 1.5, б показана пара циклов обратной передачи.

В отличие от диаграмм обмена EPP, на рис. 1.5 не приведены сигналы циклов системной шины процессора. В данном режиме обмен программы с ПУ разбивается на два относительно независимых процесса, которые связаны через FIFO-буфер. Обмен драйвера с FIFO-буфером может осуществляться как с использованием DMA, так и программного ввода-вывода. Обмен ПУ с буфером аппаратно выполняет адаптер ECP. Драйвер в режиме ECP не имеет информации о точном состоянии процесса обмена, но обычно важно только то, завершен он или нет.

*Прямая передача* данных на внешнем интерфейсе состоит из следующих шагов:

1. Хост помещает данные на шину канала и устанавливает признак цикла данных (высокий уровень) или команды (низкий уровень) на линии HostAck.
2. Хост устанавливает низкий уровень на линии HostClk, указывая на действие данных.
3. ПУ отвечает установкой высокого уровня на линии PeriphAck.
4. Хост устанавливает высокий уровень линии HostClk, и этот перепад может использоваться для фиксации данных в ПУ.
5. ПУ устанавливает низкий уровень на линии PeriphAck для указания на готовность к приему следующего байта.

Поскольку передача в ECP происходит через FIFO-буферы, которые могут присутствовать на обеих сторонах интерфейса, важно понимать, на каком этапе данные можно считать переданными. Данные считаются *переданными* на шаге 4, когда линия HostClk переходит в высокий уровень. В этот момент модифицируются счетчики переданных и принятых байт. В протоколе ECP есть условия, вызывающие прекращение обмена между шагами 3 и 4. Тогда эти данные не должны рассматриваться как переданные.

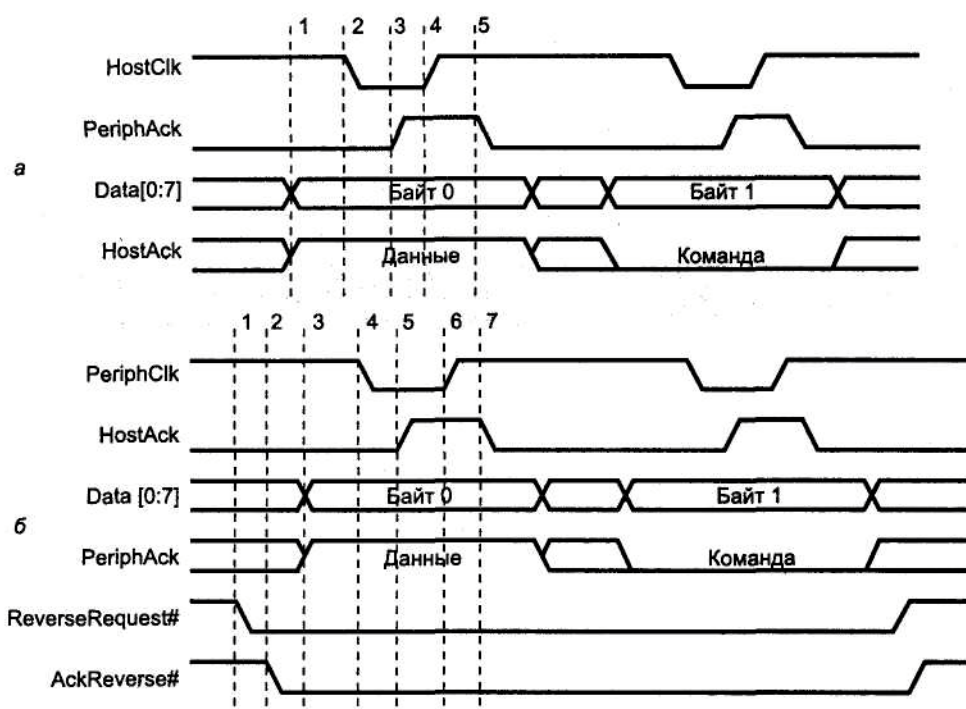


Рис. 1.5. Передача в режиме ECP: а — прямая, б — обратная

Из рис. 1.5 видно и другое отличие ECP от EPP. Протокол EPP позволяет драйверу чередовать циклы прямой и обратной передачи, не запрашивая подтверждения на смену направления. В ECP смена направления должна быть согласована: хост запрашивает реверс установкой `ReverseRequest*`, после чего он должен дождаться подтверждения сигналом `AckReverse*`. Поскольку предыдущий цикл мог выполняться по прямому доступу, драйвер должен дождаться завершения прямого доступа или прервать его, выгрузить буфер FIFO, определив точное значение счетчика переданных байт, и только после этого запрашивать реверс.

Обратная передача данных состоит из следующих шагов:

1. Хост запрашивает изменение направления канала, устанавливая низкий уровень на линии `ReverseRequest*`.
2. ПУ разрешает смену направления установкой низкого уровня на линии `AckReverse*`.
3. ПУ помещает данные на шину канала и устанавливает признак цикла данных (высокий уровень) или команды (низкий уровень) на линии `PeriphAck`.
4. ПУ устанавливает низкий уровень на линии `PeriphClk`, указывая на действительность данных.
5. Хост отвечает установкой высокого уровня на линии `HostAck`.

6. ПУ устанавливает высокий уровень линии PeriphClk; этот перепад может использоваться для фиксации данных хостом.
7. Хост устанавливает низкий уровень на линии HostAck для указания на готовность к приему следующего байта.

### 1.3.5. Режимы и регистры ECP-порта

Программный интерфейс и регистры ECP для адаптеров IEEE 1284 определяет спецификация Microsoft. Порт ECP может работать в различных режимах, приведенных в табл. 1.7, где код соответствует полю Mode регистра ECR (биты [7:5]).

Таблица 1.7. Режимы ECP-порта

Код	Режим
0	<i>SPP mode</i> , стандартный (традиционный) режим
1	<i>Bi-directional mode</i> , двунаправленный порт (тип 1 для PS/2)
10	<i>Fast Centronics</i> , однонаправленный с использованием FIFO и DMA
11	<i>ECP Parallel Port mode</i> , собственно режим ECP
100	<i>EPP Parallel Port mode</i> , режим EPP <sup>1</sup>
101	Зарезервировано
110	<i>Test mode</i> , тестирование работы FIFO и прерываний
111	<i>Configuration mode</i> , доступ к конфигурационным регистрам

<sup>1</sup> Этот режим не входит в спецификацию Microsoft, но трактуется как EPP многими адаптерами портов, если в CMOS Setup установлен режим ECP+EPP.

Регистровая модель адаптера ECP (табл. 1.8) использует свойства архитектуры стандартной шины и адаптеров ISA, где для дешифрации адресов портов ввода-вывода задействуются только 10 младших линий шины адреса. Поэтому, например, обращения по адресам Port, Port+400h, Port+800h... будут восприниматься как обращения к адресу Port, лежащему в диапазоне 0-3FFh. Современные PC и адаптеры декодируют большее количество адресных бит, поэтому обращения по адресам 0378h и 0778h будут адресованы двум различным регистрам. Помещение дополнительных регистров ECP «за спину» регистров стандартного порта (смещение 400-402h) преследует две цели. Во-первых, эти адреса никогда не использовались традиционными адаптерами и их драйверами, и их применение в ECP не приведет к сужению доступного адресного пространства ввода-вывода. Во-вторых, этим обеспечивается совместимость со старыми адаптерами на уровне режимов 000-001 и возможность определения факта присутствия ECP-адаптера посредством обращения к его расширенным регистрам.

Каждому режиму ECP соответствуют (и доступны) свои функциональные регистры. Переключение режимов осуществляется записью в регистр ECR. «Дежурными» режимами, включаемыми по умолчанию, являются 000 или 001. В любом из них работает полубайтный режим ввода. Из этих режимов всегда можно переключиться в любой другой, но из старших режимов (010-111) переключение возмож-



но только в 000 или 001. Для корректной работы интерфейса перед выходом из старших режимов необходимо дождаться завершения обмена по прямому доступу и очистки FIFO-буфера.

В режиме 000 (*SPP*) порт работает как однонаправленный программно-управляемый SPP-порт.

В режиме 001 (*Bi-Di PS/2*) порт работает как двунаправленный порт PS/2 типа 1. От режима 000 отличается возможностью реверса канала данных по биту CR. 5.

Режим 010 (*Fast Centronics*) предназначен только для высокопроизводительного вывода через FIFO-буфер с использованием DMA. Сигналы квитирования по протоколу Centronics вырабатываются аппаратно. Сигнал запроса прерывания вырабатывается по состоянию FIFO-буфера, но не по сигналу Ask# (запрос одиночного байта «не интересуется» драйвер быстрого блочного вывода).

Режим 011 является собственно режимом ECP, описанным выше, Поток данных и команд, передаваемых в ПУ, помещается в FIFO-буфер через регистры ECPDFIFO и ECPAFIFO соответственно. Из FIFO они выводятся с соответствующим признаком цикла (состояние линии HostAck). Принимаемый поток данных от ПУ извлекается из FIFO-буфера через регистр ECPDFIFO. Получение адреса в командном цикле от ПУ не предусматривается. Обмен с регистром ECPDFIFO может производиться и по каналу DMA.

Компрессия по методу RLE при передаче выполняется программно. Для передачи подряд более двух одинаковых байт данных в регистр EC PAFIFO записывается байт, у которого младшие 7 бит содержат счетчик RLC (значение RLC-127 соответствует 128 повторам), а старший бит нулевой. После этого в ECPDFIFO записывается сам байт. Принимая эту пару байт (командный байт и байт данных), ПУ осуществляет декомпрессию. При приеме потока от ПУ адаптер ECP декомпрессию осуществляет аппаратно и в FIFO-буфер помещает уже декомпрессированные данные. Отсюда очевидно, что вывод данных с одновременным использованием компрессии и DMA невозможен.

Режим 100 (*EPP*) — один из способов включения режима EPP (если таковой поддерживается адаптером и разрешен в CMOS Setup).

Режим 110 (*Test Mode*) предназначен для тестирования взаимодействия FIFO и прерываний. Данные могут передаваться в регистр TFIFO и из него с помощью DMA или программным способом. На внешний интерфейс обмен не воздействует. Адаптер отрабатывает операции вхолостую на максимальной скорости интерфейса (как будто сигналы квитирования приходят без задержек). Адаптер следит за состоянием буфера и по мере необходимости вырабатывает сигналы запроса прерывания. Таким образом программа может определить максимальную пропускную способность канала.

Режим 111 (*Configuration mode*) предназначен для доступа к конфигурационным регистрам. Выделение режима защищает адаптер и протокол от некорректных изменений конфигурации в процессе обмена.

Как уже упоминалось, каждому режиму ECP соответствуют свои функциональные регистры (табл. 1.8).

Таблица Регистры ECP  
1.8.

Смещение	Имя	R/W	Режимы ECP <sup>1</sup>	Название
000	DR	R/W	000-001	Data Register
000	ЕСРАFIF	R/W	011	ECP Address FIFO
001	SR	R/W	Все	Status Register
002	CR	R/W	Все	Control Register
400	SDFIFO	R/W	010	Parallel Port Data FIFO
400	ЕСPDFIF	R/W	011	ECP Data FIFO
400	TFIFO	R/W	110	Test FIFO
400	ЕСPCFGA	R	111	Configuration Register A
401	ЕСPCFGB	R/W	111	Configuration Register B
402	ECR	R/W	Все	Extended Control Register

<sup>1</sup> Регистры доступны только в данных режимах (указаны значения бит 7-5 регистра ECR).

*Регистр данных DR* используется для передачи данных только в программно-управляемых режимах (000 и 001).

*Регистр состояния SR* передает значение сигналов на соответствующих линиях (как в SPP).

*Регистр управления CR* имеет назначение бит, совпадающее с SPP. В режимах 010,011 запись в биты 0,1 (сигналы AutoLF\* и Strobe\*) игнорируется. *Регистр ЕСРАFIFO* служит для помещения информации командных циклов (канального адреса или счетчика RLE, в зависимости от бита?) в FIFO-буфер. Из буфера информация будет выдана в командном цикле вывода. *Регистр SDFIFO* используется для передачи данных в режиме 010. Данные, записанные в регистр (или посланные по каналу DMA), передаются через буфер FIFO по реализованному аппаратно протоколу Centronics. При этом должно быть задано прямое направление передачи (бит CR. 5-0).

*Регистр DFIFO* используется для обмена данными в режиме 011 (ECP). Данные, записанные в регистр или считанные из него (или переданные по каналу DMA), передаются через буфер FIFO по протоколу ECP.

*Регистр TFIFO* обеспечивает механизм тестирования FIFO-буфера в режиме 110.

*Регистр ЕСPCFGA* позволяет считывать информацию об адаптере (идентификационный код в битах [7:4]).

*Регистр ЕСPCFGB* позволят хранить любую информацию, необходимую драйверу. Запись в регистр не влияет на работу порта.

*Регистр ECR* — главный управляющий регистр ECP. Его биты имеют следующее назначение:

- ◆ ECR[7:5] - ECP MODE - задают режим ECP;
- ◆ ECR. 4 — ERRINTREN\* — (Error Interrupt Disable) запрещает прерывания по сигналу Error\* (при нулевом значении бита по отрицательному перепаду на этой линии вырабатывается запрос прерывания);

- ◆ ECR. 3 — DMAEN — (DMA Enable) разрешает обмен по каналу DMA;
- ◆ ECR. 2 — SERVICEINTR — (Service Interrupt) запрещает сервисные прерывания, которые вырабатываются по окончании цикла DMA (если он разрешен), по порогу заполнения/опустошения FIFO-буфера (если не используется DMA) и по ошибке переполнения буфера сверху или снизу;
- ◆ ECR.1 — FIFOFS — (FIFO Full Status) сигнализирует о заполнении буфера; при FIFOFS=LB буфере нет ни одного свободного байта;
- ◆ ECR.0 — FIFOES — (FIFO Empty Status) указывает на полное опустошение буфера; комбинация FIFOFS=FIFOES=1 означает ошибку работы с FIFO (переполнение сверху или снизу).

Когда порт находится в стандартном или двунаправленном режимах (000 или 001), первые три регистра полностью совпадают с регистрами стандартного порта. Так обеспечивается совместимость драйвера со старыми адаптерами и старых драйверов с новыми адаптерами.

По интерфейсу с программой ECP-порт напоминает EPP: после установки режима (записи кода в регистр ECR) обмен данными с устройством сводится к чтению или записи в соответствующие регистры. За состоянием FIFO-буфера наблюдают либо по регистру ECR, либо по обслуживанию сервисных прерываний от порта. Весь протокол квитирования генерируется адаптером аппаратно. Обмен данными с ECP-портом (кроме явного программного) возможен и по прямому доступу к памяти (каналу DMA), что эффективно при передаче больших блоков данных.

### 1.3.6. Согласование режимов IEEE 1284

ПУ в стандарте IEEE 1284 обычно не требуют от контроллера реализации всех предусмотренных этим стандартом режимов. Для определения режимов и методов управления конкретным устройством стандарт предусматривает *последовательность согласования (negotiation sequence)*. Последовательность построена так, что старые устройства, не поддерживающие IEEE 1284, на нее не ответят, и контроллер останется в стандартном режиме. Периферия IEEE 1284 может сообщить о своих возможностях, и контроллер установит режим, удовлетворяющий и хост, и ПУ.

Во время фазы согласования контроллер выставляет на линии данных *байт расширяемости (extensibility byte)*, запрашивая подтверждение на перевод интерфейса в требуемый режим или прием идентификатора ПУ (табл. 1.9). Идентификатор передается контроллеру в запрошенном режиме (любой режим обратного канала, кроме EPP). ПУ использует сигнал Xflag (Select в терминах SPP) для подтверждения запрошенного режима обратного канала, кроме полубайтного, который поддерживается всеми устройствами IEEE 1284. Бит Extensibility Link request послужит для определения дополнительных режимов в будущих расширениях стандарта.

Таблица 1.9. Биты в байте расширяемости

Бит	Описание	Допустимые комбинации бит [7:0]
7	Request Extensibility Link — зарезервирован	1000 0000
6	Запрос режима EPP	0100 0000
5	Запрос режима ECP с RLE	00110000
4	Запрос режима ECP без RLE	00010000
3	Зарезервировано	00001000
2	Запрос идентификатора устройства с ответом в режиме:	
	полубайтный	0000 0100
	байтный	0000 0101
	ECP без RLE	00010100
	ECP с RLE	00110100
1	Зарезервировано	0000 0010
0	Запрос полубайтного режима	0000 0001
	попе Запрос байтного режима	0000 0000

Последовательность согласования (рис. 1.6) состоит из следующих шагов.

1. Хост выводит байт расширяемости на линии данных.
2. Хост устанавливает высокий уровень сигнала SelectIn\* и низкий — AutoFeed\*, что означает начало последовательности согласования.
3. ПУ отвечает установкой низкого уровня сигнала Ack# и высокого — Error\*, PaperEnd и Select. Устройство, «не понимающее» стандарта 1284, ответа не даст, и дальнейшие шаги не выполняются.

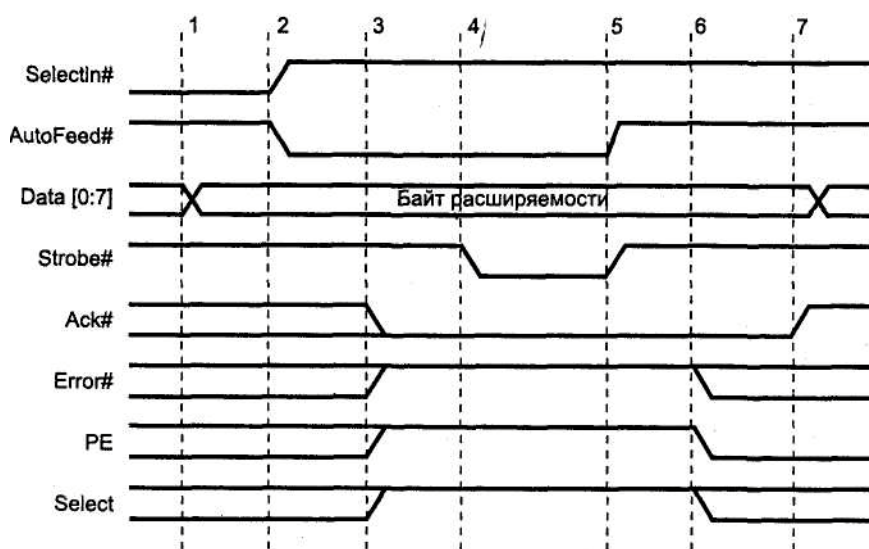


Рис. 1.6. Последовательность согласования режимов IEEE 1284

4. Хост устанавливает низкий уровень сигнала Strobe\* для записи байта расширяемости в ПУ.
5. Хост устанавливает высокий уровень сигналов Strobe\* и AutoLF\*.
6. ПУ отвечает установкой в низкий уровень сигналов PAperEnd и Error\*, если ПУ имеет обратный канал передачи данных. Если запрошенный режим поддерживается устройством, на линии Select устанавливается высокий уровень, если не поддерживается — низкий.
7. ПУ устанавливает высокий уровень на линии Ask# для указания на завершение последовательности согласования, после чего контроллер задает требуемый режим работы.

### 1.3.7. Физический и электрический интерфейсы

Стандарт IEEE 1284 определяет физические характеристики приемников и передатчиков сигналов, которые по уровням совместимы с ТТЛ. Спецификации стандартного порта не задавали типов выходных схем, предельных значений величин нагрузочных резисторов и емкости, вносимой цепями и проводниками. На относительно невысоких скоростях обмена разброс этих параметров не вызывал проблем совместимости. Однако расширенные (функционально и по скорости передачи) режимы требуют четких спецификаций. IEEE 1284 определяет два уровня интерфейсной совместимости. *Первый уровень (Level I)* определен для устройств медленных, но использующих смену направления передачи данных. *Второй уровень (Level II)* определен для устройств, работающих в расширенных режимах с высокими скоростями и длинными кабелями. К *передатчикам предъявляются* следующие требования.

- ◆ Уровни сигналов без нагрузки не должны выходить за пределы  $-0,5... +5,5$  В.
- ◆ Уровни сигналов при токе нагрузки 14 мА должны быть не ниже +2,4 В для высокого уровня ( $V_{OH}$ ) и не выше +0,4 В для низкого уровня ( $V_{OL}$ ) на постоянном токе.
- ◆ Выходной импеданс  $R_0$ , измеренный на разъеме, должен составлять  $50 \pm 5$  Ом на уровне  $V_{OH} - V_{OL}$ . Для обеспечения заданного импеданса используют последовательные резисторы в выходных цепях передатчика. Согласование импеданса передатчика и кабеля снижает уровень импульсных помех.
- ◆ Скорость нарастания (спада) импульса должна находиться в пределах 0,05-0,4 В/нс.

Ниже перечислены требования к *приемникам*.

- ◆ Допустимые пиковые значения сигналов  $-2,0...+7,0$  В.
- ◆ Пороги срабатывания должны быть не выше 2,0 В ( $V_{IH}$ ) для высокого уровня и не ниже 0,8 В ( $V_{IL}$ ) для низкого.
- ◆ Приемник должен иметь гистерезис в пределах 0,2-1,2 В (гистерезисом обладают специальные микросхемы — триггеры Шмитта).

- ♦ Входной ток микросхемы (втекающий и вытекающий) не должен превышать 20 мкА, входные линии соединяются с шиной питания +5 В резистором 1,2 кОм.
- ♦ Входная емкость не должна превышать 50 пФ.

Когда появилась спецификация ECP, компания Microsoft рекомендовала применение динамических терминаторов на каждую линию интерфейса. Однако в настоящее время следуют спецификации IEEE 1284, в которой динамические терминаторы не применяются. Рекомендованные схемы входных, выходных и двунаправленных цепей приведены на рис. 1.7.

Стандарт IEEE 1284 определяет три типа используемых *разъемов*. Типы А (DB-25) и В (Centronics-36) характерны для традиционных кабелей подключения принтера, тип С — новый малогабаритный 36-контактный разъем.

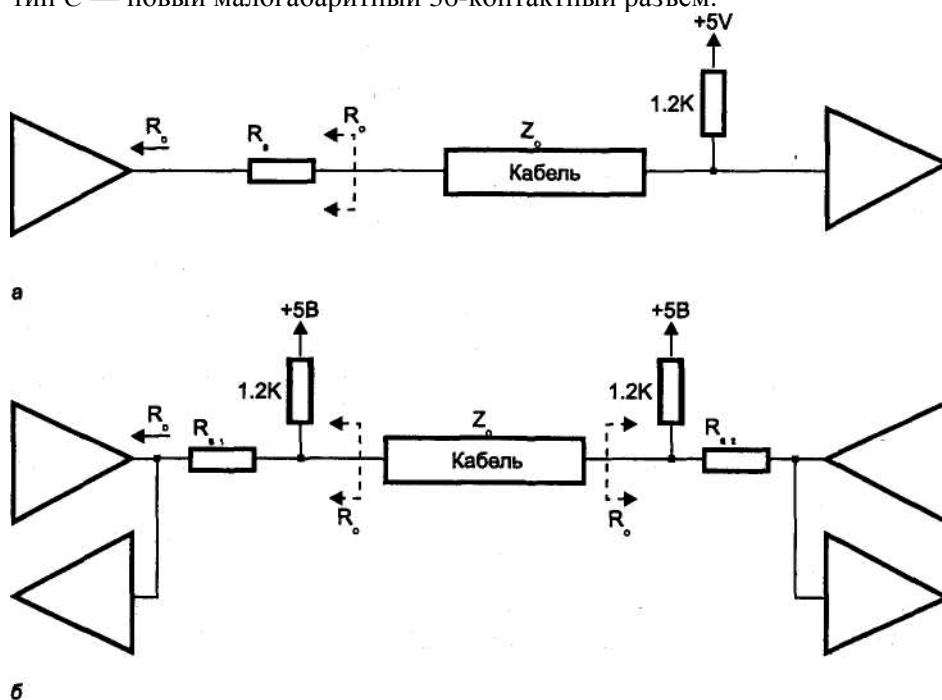


Рис. 1.7. Оконечные цепи линий интерфейса IEEE 1284: а — однонаправленные линии, б — двунаправленные

Традиционные интерфейсные кабели имеют от 18 до 25 проводов, в зависимости от числа проводников цепи GND. Эти проводники могут быть как перевитыми, так и нет. К экранированию кабеля жестких требований не предъявлялось. Такие кабели вряд ли будут надежно работать на скорости передачи 2 Мбайт/с и при длине более 2 м.

Стандарт IEEE 1284 регламентирует *свойства кабелей*.

- ♦ Все сигнальные линии должны быть перевитыми с отдельными обратными (общими) проводами.

- ◆ Каждая пара должна иметь импеданс  $62 \pm 6$  Ом в частотном диапазоне 4-16 МГц.
- ◆ Уровень перекрестных помех между парами не должен превышать 10 %.
- ◆ Кабель должен иметь экран (фольгу), покрывающий не менее 85 % внешней поверхности. На концах кабеля экран должен быть окольцован и соединен с контактом разъема.

Кабели, удовлетворяющие этим требованиям, маркируются надписью «IEEE Std 1284-1994 Compliant». Они могут иметь длину до 10 метров, обозначения типов приведены в табл. 1.10.

Таблица 1.10. Типы кабелей IEEE 1284

Тип	Расшифровка	Разъем 1	Разъем 2
AMA	Type A Male — Type A Male	A (вилка)	A(вилка)
AMAF	Type A Male — Type A Female	A(вилка)	A(розетка)
AB	Type A Male — Type B Plug — стандартный кабель к принтеру	A (вилка)	B
AC	Type A Male — Type C Plug — новый кабель к принтеру	A (вилка)	C
BC	Type B Plug — Type C Plug	B	C
CC	Type C Plug — Type C Plug	C	C

### 1.3.8. Развитие стандарта IEEE 1284

Ниже перечислены некоторые дополнения основного стандарта IEEE 1284.

- ◆ *IEEE P1284.1* «Standard for Information Technology for Transport Independent Printer/Scanner Interface (TIP/SI)». Этот стандарт разрабатывается для управления и обслуживания сканеров и принтеров на основе протокола NPAP (Network Printing Alliance Protocol).
- ◆ *IEEE P1284.2* «Standard for Test, Measurement and Conformance to IEEE Std. 1284» — стандарт для тестирования портов, кабелей и устройств на совместимость с IEEE 1284.
- ◆ *IEEE P1284.3* «Standard for Interface and Protocol Extensions to IEEE Std. 1284 Compliant Peripheral and Host Adapter Ports» — стандарт на драйверы и использование устройств прикладным программным обеспечением (ПО). Приняты спецификации BIOS для использования EPP драйверами DOS. Прорабатывается стандарт на разделяемое использование одного порта цепочкой устройств или группой устройств, подключаемых через мультиплексор.
- ◆ *IEEE P1284.4* «Standard for Data Delivery and Logical Channels for IEEE Std. 1284 Interfaces» направлен на реализацию пакетного протокола достоверной передачи данных через параллельный порт. Основой служит протокол MLC (Multiple Logical Channels) фирмы Hewlett-Packard, однако совместимость с ним в окончательной версии стандарта не гарантируется.

## 1.4. Системная поддержка LPT-порта

Системная поддержка LPT-порта включает поиск установленных портов и сервисы печати (Int 17h, см. п. 8.3.3). В процессе начального тестирования POST BIOS проверяет наличие параллельных портов по адресам 3BCh, 378h и 278h и помещает базовые адреса обнаруженных портов в ячейки BIOS Data Area 0:0408h, 040Ah, 040Ch, 040Eh. Эти ячейки хранят адреса портов LPT1-LPT4, нулевое значение адреса является признаком отсутствия порта с данным номером. В ячейки 0:0478, 0479, 047A, 047B заносятся константы, задающие тайм-аут для этих портов.

*Поиск портов* обычно ведется достаточно примитивно — по базовому адресу (в регистр данных предполагаемого порта) выводится тестовый байт (AAh или 55h), затем производится ввод по тому же адресу. Если считанный байт совпал с записанным, предполагается, что найден LPT-порт; его адрес помещается в ячейку BIOS Data Area. Базовые адреса портов могут быть впоследствии изменены программно. Адрес порта LPT4 система BIOS самостоятельно установить не может, поскольку в списке стандартных адресов поиска имеются только три вышеуказанных.

Обнаруженные порты *инициализируются* — записью в регистр управления формируется и снимается сигнал Init#, после чего записывается значение ОСЬ, соответствующее исходному состоянию сигналов интерфейса. В некоторых случаях сигнал Init# активен с момента аппаратного сброса до инициализации порта при загрузке ОС. Это можно заметить по поведению включенного принтера во время перезагрузки компьютера — у принтера надолго гаснет индикатор On-Line. Следствие этого явления — невозможность распечатки экранов (например, параметров BIOS Setup) по нажатию клавиши Print Screen до загрузки ОС.

## 1.5. Параллельный порт и PnP

Большинство современных периферийных устройств, подключаемых к LPT-порту, поддерживает стандарт 1284 и функции PnP. Для поддержки этих функций компьютером с аппаратной точки зрения достаточно иметь контроллер интерфейса, соответствующий стандарту 1284. Если подключаемое устройство поддерживает PnP, оно по протоколу согласования режимов 1284 способно «договориться» с портом, представляющим «интересы» компьютера, о возможных режимах обмена. Далее, для работы PnP подключенное устройство должно сообщить операционной системе все необходимые сведения о себе. Как минимум это идентификаторы производителя, модели и набор поддерживаемых команд. Более развернутая информация об устройстве может содержать идентификатор класса, подробное описание и идентификатор устройства, с которым обеспечивается совместимость. В соответствии с принятой информацией для поддержки данного устройства операционная система может предпринять действия по установке требуемого программного обеспечения.

Устройства с поддержкой PnP распознаются ОС на этапе ее загрузки, если, конечно же, они подключены к порту интерфейсным кабелем и у них включено пита-



ние. Если ОС Windows обнаруживает подключенное устройство PnP, отличающееся от того, что прописано в ее реестре для данного порта (или просто новое устройство), она пытается установить требуемые для устройства драйверы из дистрибутива ОС или из комплекта поставки нового устройства. Если Windows не желает замечать вновь подключенного устройства PnP, это может свидетельствовать о неисправности порта или кабеля. Система PnP не работает, если устройство подключается дешевым «не двунаправленным» кабелем, у которого отсутствует связь по линии SelectIn\* (контакт 17 порта LPT и контакт 36 разъема Centronics).

## 1.6. Применение LPT-порта

Обычно LPT-порт используют для подключения принтера (см. п. 8.3.1), однако этим его применение не исчерпывается.

Для связи двух компьютеров по параллельному интерфейсу применяются различные кабели в зависимости от режимов используемых портов. Самый простой и медленный — полубайтный режим, работающий на *всех* портах. Для этого режима в кабеле достаточно иметь 10 сигнальных и один общий провод. Распайка разъемов кабеля приведена в табл. 1.11. Связь двух PC данным кабелем поддерживается стандартным ПО типа Interlnk из MS-DOS или Norton Commander. Заметим, что здесь применяется свой протокол, отличный от описанного в п. 1.3.1.

Таблица 1.11. Кабель связи PC-PC (4-битный)

X1, разъем PC#1		X2, разъем PC#2	
Контакт	Контакт	Контакт	Бит
DR.0	2	15	SR.3
DR.1	3	13	SR.4
DR.2	4	12	SR.5
DR.3	5	10	SR.6
DR.4	6	11	SR.7
SR.6	10	5	DR.3
SR.7	11	6	DR.4
SR.5	12	4	DR.2
SR.4	13	3	DR.1
SR.3	15	2	DR.
GND	18-25	18-25	GND

Разъемы X1 и X2 - DB25-P (вилки).

Высокоскоростная связь двух компьютеров может выполняться и в режиме ECP (режим EPP неудобен, поскольку требует синхронизации шинных циклов ввода-вывода двух компьютеров).

В табл. 1.12 приведена распайка кабеля, в аналогичной таблице предыдущих книг автора (см. [2] и [5]) была ошибка (перепутаны контакты 13 и 15). Из всех сигналов

в кабеле не используется лишь PeriphRequest\* (контакт 15). В цепи линий данных рекомендуется вставить последовательные резисторы (0,5-1 кОм), препятствующие протеканию слишком больших токов, когда порты данных обоих компьютеров находятся в режиме вывода. Эта ситуация возникает, когда коммуникационное ПО компьютеров еще не запущено. Связь в режиме ECP поддерживается Windows Эх, в комплект поставки этих ОС входит драйвер PARALINK.VxD, но из-за внутренней ошибки он неработоспособен. «Заплатку» на этот драйвер, а также тестовую утилиту и необходимые описания можно найти в сети ([www.lpt.com](http://www.lpt.com), [www.lvr.com/parport.htm](http://www.lvr.com/parport.htm)).

**Таблица 1.12.** Кабель связи PC-PC в режиме ECP и байтном режиме

Разъем X1		Разъем X2	
Контакт	Имя в ECP	Имя в ECP	Контакт
1	HostClk	PeriphClk	10
14	HostAck	PeriphAck	11
17	1284Active	Xflag	13
16	ReverseRequest*	AckReverse*	12
10	PeriphClk	HostClk	1
11	PeriphAck	HostAck	14
12	AckReverse*	ReverseRequest*	16
13	Xflag	1284Active	17
2,3...9	Data [0:7]	Data [0:7]	2,3...9

Подключение *сканера* к LPT-порту эффективно, только если порт обеспечивает хотя бы двунаправленный режим (*Bi-Di*), поскольку основной поток — ввод. Лучше использовать порт ECP, если этот режим поддерживается сканером (или EPP, что маловероятно).

Подключение *внешних накопителей* (Omega Zip Drive, CD-ROM и др.), *адаптеров ЛВС* и других симметричных устройств ввода-вывода имеет свою специфику.

В режиме SPP наряду с замедлением работы устройства заметна принципиальная

асимметрия этого режима: *чтение данных* происходит в два раза медленнее, чем (весьма небыстрая) *запись*. Применение *двунаправленного* режима (*Bi-Di* или *PS/2Type 1*) устранил эту асимметрию — *скорости сравняются*. Только перейдя на EPP или ECP, можно получить *нормальную* скорость работы. В режиме EPP или ECP подключение к LPT-порту почти не уступает по скорости подключению через ISA- контроллер. Это справедливо и при подключении устройств со стандартным интерфейсом шин к LPT-портам через преобразователи интерфейсов (например, LPT- IDE, LPT- SCSI, LPT- PCMCIA). Заметим, что винчестер IDE, подключенный через адаптер к LPT-порту, для системы может быть представлен как устройство SCSI (это логичнее с программной точки зрения).

В табл. 1.13 описано назначение выводов разъема LPT-порта в различных режимах и их соответствие битам регистров стандартного порта.

.13. Назначение выводов разъема LPT-порта и бит регистров в режимах SPP, ECP и EPP					
Таблица 1					
Контакт	I/O	Бит <sup>1</sup>	SPP	ECP	EPP
1	0/I	CR.O\	Strobe#	HostClk	Write»
2	0/I	DR.O	DataO	Data 0	DataO
3	0/I	DR.1	Data1	Data1	Data1
4	0/I	DR. 2	Data 2	Data 2	Data 2
5	0/I	DR.3	DataS	DataS	DataS
6	0/I	DR.4	Data 4	Data 4	Data 4
7	0/I	DR.5	Data S	Data S	Data S
8	0/I	DR.6	Data 6	Data 6	Data 6
9	0/I	DR.7	Data 7	Data 7	Data 7
10	I	SR.6	Ack#	PeriphClk	INTR#
11	I	SR.A	Busy	PeriphAck	Wait#
12	I	SR.5	PaperEnd	AckReverse*	__ 2
13	I	SR.4	Select	Xflag	__ <sup>2</sup>
14	0/I	CR.1\	Auto LF#	HostAck	DataStb*
15	I	SR.3	Error»	PeriphRequest*	__ 2
16	0/I	CR.2	lnit#	ReverseRequest*	Reset»
17	0/I	CR.3\	Select ln#	1284Active	AddrStb#

<sup>1</sup> Символом «\» отмечены инвертированные сигналы (1 в регистре соответствует низкому уровню линии).

<sup>2</sup> Определяется пользователем.

## 1.7. Конфигурирование LPT-портов

Управление параллельным портом разделяется на два этапа — *предварительное конфигурирование* (Setup) аппаратных средств порта и *текущее* (оперативное) *переключение* режимов работы прикладным или системным ПО. Оперативное переключение возможно только в пределах режимов, разрешенных при конфигурировании. Этим обеспечивается возможность согласования аппаратуры с ПО и блокирования ложных переключений, вызванных некорректными действиями программы.

Конфигурирование LPT-порта зависит от его исполнения. Порт, расположенный на плате расширения (мультикарте), устанавливаемой в слот ISA или ISA+VLB, конфигурируется джамперами на самой плате. Порт на системной плате конфигурируется через BIOS Setup.

Ниже перечислены параметры, подлежащие конфигурированию.

- ♦ *Базовый адрес* — 3BCh, 378h или 278h. При инициализации BIOS проверяет наличие портов по адресам именно в этом порядке и, соответственно, присваивает обнаруженным портам логические имена LPT1, L PT2, L PT3. Адрес 3BСЬ имеет адаптер порта, расположенный на плате MDA или HGS. Большинство портов по умолчанию конфигурируется на адрес 378h и может переключаться на 278h.

- ◆ Используемая *линия запроса прерывания*: для L PT — IRQ7, для LPT2 — IRQ5. Традиционно прерывания от принтера не задействуются, и этот дефицитный ресурс можно сэкономить. Однако при использовании скоростных режимов ECP (или Fast Centronics) работа через прерывания может заметно повысить производительность и снизить загрузку процессора.
- ◆ Использование *канала DMA* для режимов ECP и Fast Centronics — разрешение и номер канала DMA.
- ◆ Режимы работы порта:
  - *SPP* — порт работает только в стандартном однонаправленном программно-управляемом режиме;
  - *PS/2*, он же *Bi-Directional* — отличается от SPP возможностью реверса канала (установкой CR. 5=1);
  - *Fast Centronics* — аппаратное формирование протокола Centronics с использованием FIFO-буфера и, возможно, DMA;
  - *EPP* — в зависимости от использования регистров порт работает в режиме SPP или EPP;
  - *ECP*— по умолчанию включается в режим SPP или PS/2, записью в ECR может переводиться в любой режим ECP, но перевод в EPP записью в ECR кода 100 не гарантируется;
  - *ECP+EPP* — то же, что и ECP, но запись в ECR кода режима 100 переводит порт в EPP.

Выбор режима EPP, ECP или Fast Centronics сам по себе не приводит к повышению быстродействия обмена с подключенными ПУ, а только дает возможность драйверу и ПУ установить оптимальный режим в пределах их «разумения». Большинство современных драйверов и приложений пытаются использовать эффективные режимы, поэтому «подрезать им крылья» установкой простых режимов без веских на то оснований не стоит.

*Принтеры и сканеры* могут пожелать режима ECP. Windows (3.x, Эх и NT) имеет системные драйверы для этого режима. В среде DOS печать через ECP поддерживается только специальным загружаемым драйвером.

*Сетевые адаптеры, внешние диски и CD-ROM*, подключаемые к параллельному порту, могут использовать режим EPP. Для этого режима специальный драйвер пока еще не применяется; поддержка EPP включается в драйвер самого подключаемого устройства.

## 1.8. Неисправности и тестирование параллельных портов

Тестирование параллельных портов разумно начинать с *проверки их наличия* в системе. Список адресов установленных портов появляется в таблице, выводимой BIOS на экран перед загрузкой ОС. Список можно посмотреть и с помощью тестовых программ или прямо в BIOS Data Area с помощью отладчика.

Если BIOS обнаруживает меньше портов, чем установлено физически, скорее всего, двум портам присвоен один и тот же адрес. При этом работоспособность ни одного из конфликтующих портов не гарантируется: они будут одновременно выводить сигналы, но при чтении регистра состояния конфликт на шине, скорее всего, приведет к искажению данных. Программное тестирование порта без диагностической заглушки (Loop Back) не покажет ошибок, поскольку при этом читаются данные выходных регистров, а они у всех конфликтующих (по отдельности исправных портов) совпадут. Именно такое тестирование производит BIOS при проверке на наличие портов. Разбираться с такой ситуацией следует, последовательно устанавливая порты и наблюдая за адресами, появляющимися в списке.

Если физически установлен только один порт, а BIOS его не обнаруживает, то либо порт отключен при конфигурировании, либо он вышел из строя (скорее всего из-за нарушений правил подключения). Если вам везет, неисправность устраняется «передергиванием» платы в слоте — там иногда возникают проблемы с контактами.

Наблюдаются и такие «чудеса» — при «теплой» перезагрузке DOS после Windows 95 порт не виден (и приложения не могут печатать из MS-DOS). Однако после повторной перезагрузки DOS порт оказывается на месте. С этим явлением легче смириться, чем бороться.

Тестирование портов с помощью диагностических программ позволяет проверить выходные регистры, а при использовании специальных заглушек — и входные линии. Поскольку количество выходных линий порта (12) и входных (5) различно, то полная проверка порта с помощью пассивной заглушки принципиально невозможна. Разные программы тестирования требуют применения разных заглушек (рис. 1.8).

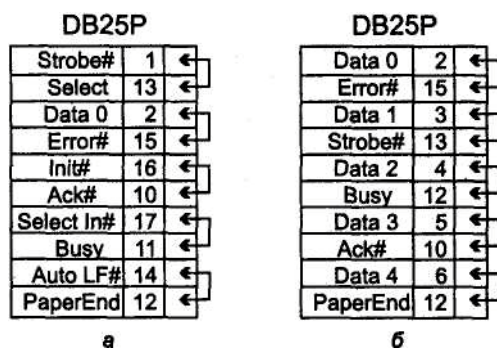


Рис. 1.8. Схема заглушки для тестирования LPT-порта: а — для Norton Diagnostics б -для Checklt,

Большинство неприятностей при работе с LPT-портами доставляют *разъемы и кабели*. Для проверки порта, кабеля и принтера можно воспользоваться специальными тестами из популярных диагностических программ (Checklt, PCCheck

и т. п.). Можно попытаться просто вывести на принтер какой-либо символьный файл.

- ◆ Если вывод файла с точки зрения DOS проходит (копирование файла на устройство с именем LPTn или PRN совершается быстро и успешно), а принтер (исправный) не напечатал ни одного символа — скорее всего, это обрыв (не контакт в разъеме) цепи Strobe\*.
- ◆ Если принтер находится в состоянии *On Line*, а появляется сообщение о его неготовности, причину следует искать в линии *Busy*.
- ◆ Если принтер, подключенный к порту, в стандартном режиме (SPP) печатает нормально, а при переходе в режим ECP начинаются сбои, следует проверить кабель — соответствует ли он требованиям IEEE 1284 (см. выше). Дешевые кабели с неперевитыми проводами нормально работают на скоростях 50-100 Кбайт/с, но при скорости 1-2 Мбайт/с, обеспечиваемой ECP, имеют полное право не работать, особенно при длине более 2 м.
- ◆ Если при установке драйвера PnP-принтера появилось сообщение о необходимости применения «двунаправленного кабеля», проверьте наличие связи контакта 17 разъема DB-25 с контактом 36 разъема Centronics. Хотя эта связь изначально предусматривалась, в ряде кабелей она отсутствует.
- ◆ Если принтер искажает информацию при печати, возможен обрыв (или замыкание) линий данных. В этом случае удобно воспользоваться файлом, содержащим последовательность кодов всех печатных символов. Если файл печатается с повтором некоторых символов или их групп, по периодичности повтора можно легко вычислить оборванный провод данных интерфейса. Этот же файл удобно использовать для проверки аппаратной русификации принтера.

*Аппаратные прерывания* от LPT-порта используются не всегда. Даже DOS-программа фоновой печати PRINT работает с портом по опросу состояния, а ее обслуживающий процесс запускается по прерыванию от таймера. Поэтому неисправности, связанные с цепью прерывания от порта, проявляются не часто. Однако по-настоящему многозадачные ОС (например, NetWare) стараются работать с портом по прерываниям. Протестировать линию прерывания можно, только подключив к порту ПУ или заглушку. Если к порту с неисправным каналом прерывания подключить адаптер локальной сети, то он, возможно, будет работать, но с очень низкой скоростью: на любой запрос ответ будет приходиться с задержкой в десятки секунд — принятый из адаптера пакет будет приниматься не по прерыванию (сразу по приходу), а по внешнему таймауту.

## Глава 2

# Последовательный интерфейс — СОМ-порт

Универсальный внешний последовательный интерфейс — *СОМ-порт* (Communications Port — коммуникационный порт) присутствует в РС начиная с первых моделей. Этот порт обеспечивает *асинхронный*<sup>1</sup> обмен по стандарту RS-232C. СОМ-порты реализуются на микросхемах универсальных асинхронных приемопередатчиков. (*UART*), совместимых с семейством *i8250/16450/16550*. Они занимают в пространстве ввода-вывода по 8 смежных 8-битных регистров и могут располагаться по стандартным *базовым адресам* 3F8h (СОМ1), 2F8h (СОМ2), 3E8h (СОМ3), 2E8h (СОМ4). Порты могут вырабатывать *аппаратные прерывания* IRQ4 (обычно используются для СОМ1 и СОМ3) и IRQ3 (для СОМ2 и СОМ4). С внешней стороны порты имеют линии последовательных данных передачи и приема, а также набор сигналов управления и состояния, соответствующий стандарту RS-232C. СОМ-порты имеют внешние разъемы-вгшсм (male — «папа») *DB25P* или *DB9P*, выведенные на заднюю панель компьютера (см. п. 2.1). Характерной особенностью интерфейса является применение «не ТТЛ» сигналов — все внешние сигналы порта двуполярные. Гальваническая развязка отсутствует — схемная земля подключаемого устройства соединяется со схемной землей компьютера. Скорость передачи данных может достигать 115 200 бит/с.

Компьютер может иметь до четырех последовательных портов СОМ 1-СОМ4 (для машин класса АТ типично наличие двух портов) с поддержкой на уровне BIOS. Сервис BIOS Int 14h обеспечивает инициализацию порта, ввод и вывод символа (не используя прерываний) и опрос состояния. Через Int 14h скорость передачи программируется в диапазоне 110-9600 бит/с (меньше, чем реальные возможности порта). Для повышения производительности широко используется взаимодействие программ с портом на уровне регистров, для чего требуется совместимость аппаратных средств СОМ-порта с программной моделью 18250/16450/16550.

Название порта указывает на его основное назначение — подключение коммуникационного оборудования (например, модема) для связи с другими компьютерами, сетями и периферийными устройствами. К порту могут непосредственно подключаться и периферийные устройств с последовательным интерфейсом: принтеры, плоттеры, терминалы и другие. СОМ-порт широко используется для подключения мыши, а также организации непосредственной связи двух компьютеров. К СОМ-порту подключают и электронные ключи.

---

<sup>1</sup> *Синхронный* обмен в РС поддерживают лишь специальные адаптеры, например SDLC или V.35.

Практически все современные системные платы (еще начиная с PCI-плат для процессоров 486) имеют встроенные адаптеры двух COM-портов. Один из портов может использоваться и для беспроводной инфракрасной связи с периферийными устройствами (IrDA). Существуют карты ISA с парой COM-портов, где они чаще всего соседствуют с LPT-портом, а также с контроллерами дисковых интерфейсов (FDC+IDE). Если возникает потребность в большом количестве последовательных интерфейсов, то в ПК можно установить специальные адаптеры-мультиплексоры. Это весьма дорогие карты, они выпускаются обычно на 4,8,12 и даже 16 портов. Такое большое число разъемов на заднюю стенку ПК вывести проблематично, и у мультиплексоров обычно имеется внешний блок с разъемами (и электроникой), соединяемый с адаптером кабелем с многоконтактными разъемами. BIOS мультиплексоры не поддерживает.

«Классический» COM-порт позволял осуществлять обмен данными только программно-управляемым способом, при этом для пересылки каждого байта процессору приходится выполнять несколько инструкций. Современные порты имеют FIFO-буферы данных и позволяют выполнять обмен по каналу DMA, существенно разгружая центральный процессор, что особенно важно на больших скоростях обмена.

## 2.1. Интерфейс RS-232C

Интерфейс RS-232C предназначен для подключения аппаратуры, передающей или принимающей данные (ООД — оконечное оборудование данных, или АПД — аппаратура передачи данных; DTE — *Data Terminal Equipment*), к оконечной аппаратуре каналов данных {АКД; DCE—*Data Communication Equipment*). В роли АПД может выступать компьютер, принтер, плоттер и другое периферийное оборудование. В роли АКД обычно выступает модем. Конечной целью подключения является соединение двух устройств АПД. Полная схема соединения приведена на рис. 2.1; интерфейс позволяет исключить канал удаленной связи вместе с парой устройств АКД, соединив устройства непосредственно с помощью нуль-модемного кабеля (рис. 2.2).



Рис. 2.1. Полная схема соединения по RS-232C

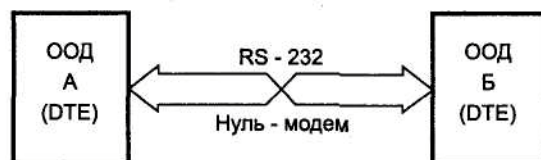


Рис. 2.2. Соединение по RS-232C нуль-модемным кабелем



Стандарт описывает управляющие сигналы интерфейса, пересылку данных, электрический интерфейс и типы разъемов. В стандарте предусмотрены асинхронный и синхронный режимы обмена, но СОМ-порты поддерживают только *асинхронный режим*. Функционально RS-232C эквивалентен стандарту МККТТ V.24/V.28 и стыку С2, но они имеют различные названия сигналов.

Стандарт RS-232C описывает несимметричные передатчики и приемники — сигнал передается относительно общего провода — схемной земли (симметричные дифференциальные сигналы используются в других интерфейсах — например, RS-422). Интерфейс *не обеспечивает гальванической развязки* устройств. Логической единице соответствует напряжение на *входе приемника* в диапазоне от -12 до -3 В. Для линий управляющих сигналов это состояние называется *ON* («включено»), для линий последовательных данных — *MARK*. Логическому нулю соответствует диапазон от +3 до +12 В. Для линий управляющих сигналов состояние называется *OFF* («выключено»), а для линий последовательных данных — *SPACE*. Диапазон от -3 до +3 В — зона нечувствительности, обуславливающая гистерезис приемника: состояние линии будет считаться измененным только после пересечения порога (рис. 2.3). Уровни сигналов на выходах передатчиков должны быть в диапазонах от -12 до -5 В и от +5 до +12 В для представления единицы и нуля соответственно. Разность потенциалов между схемными землями (SG) соединяемых устройств должна быть менее 2 В, при более высокой разности потенциалов возможно неверное восприятие сигналов.

Интерфейс предполагает наличие *защитного заземления* для соединяемых устройств, если они оба питаются от сети переменного тока и имеют сетевые фильтры.

**ВНИМАНИЕ**-----  
*Подключение и отключение интерфейсных кабелей* устройств с автономным питанием должно производиться *при отключенном питании*. Иначе разность невыровненных потенциалов устройств в момент коммутации может оказаться приложенной к выходным или входным (что опаснее) цепям интерфейса и вывести из строя микросхемы.

Стандарт RS-232C регламентирует *типы применяемых разъемов*.

На аппаратуре АПД (в том числе на СОМ-портах) принято устанавливать *вилки ПВ-25Р* или более компактный вариант — *DB-9Р*. Девятиштырьковые разъемы не имеют контактов для дополнительных сигналов, необходимых для синхронного режима (в большинстве 25-штырьковых разъемах эти контакты не используются).

На аппаратуре АКД (модемах) устанавливают *розетки DB-25S* или *DB-9S*.

Это правило предполагает, что разъемы АКД могут подключаться к разъемам АПД непосредственно или через переходные «прямые» кабели с розеткой и вилкой, у которых контакты соединены «один в один». Переходные кабели могут являться и переходниками с 9 на 25-штырьковые разъемы (рис. 2.4).

Если аппаратура АПД соединяется без модемов, то разъемы устройств (вилки) соединяются между собой *нуль-модемным кабелем* (*Zero-modem*, или *Z-modem*), имеющим на обоих концах розетки, контакты которых соединяются перекрестно по одной из схем, приведенных на рис. 2.5.

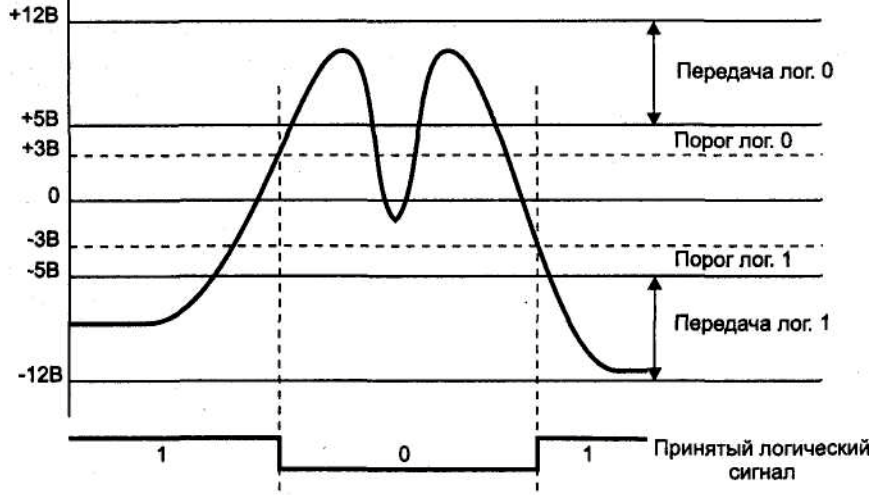


Рис. 2.3. Прием сигналов RS-232C

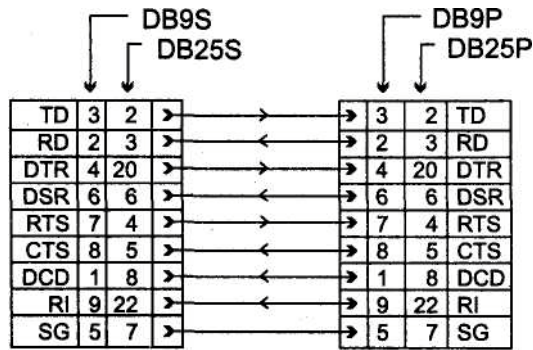
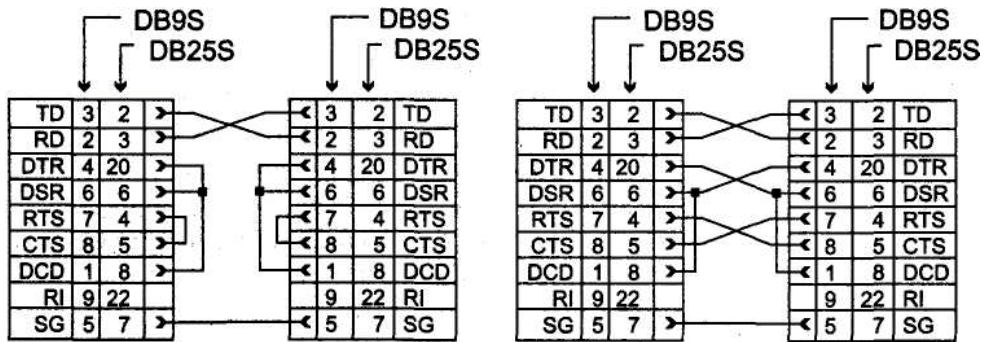


Рис. 2.4. Кабели подключения модемов



а

б

Рис. 2.5. Нуль-модемный кабель: а — минимальный, б — полный

Если на каком-либо устройстве АПД установлена розетка — это почти 100 % того, что к другому устройству оно должно подключаться прямым кабелем, аналогичным кабелю подключения модема. Розетка устанавливается обычно на тех устройствах, у которых удаленное подключение через модем не предусмотрено.

В табл. 2.1 приведено назначение контактов разъемов СОМ-портов (и любой другой аппаратуры передачи данных АПД). Контакты разъема DB-25S определены стандартом EIA/TIA-232-E, разъем DB-9S описан стандартом EIA/TIA-574. У модемов (АКД) название цепей и контактов такое же, но роли сигналов (вход-выход) меняются на противоположные.

Таблица 2.1. Разъемы и сигналы интерфейса RS-232C

Обозначение цепи			Контакт		№ провода кабеля				Направление
			разъема		выносного разъема РС				
СОМ-порт	RS-232	V.24 Стык 2	DB-25P	DB-9P	11	2 <sup>2</sup>	3 <sup>3</sup>	Л*	I/O
PG	AA	101	1	5	(10)	(10)	(10)	1	-
SG	AB	102	7	5	5	9	1	13	-
TD	BA	103	2	3	3	5	3	3	O
RD	BB	104	3	2	2	3	4	5	I
RTS	CA	105	4	7	7	4	8	7	O
CTS	CB	106	5	8	8	6	7	9	I
DSR	CC	107	6	6	6	2	9	11	I
DTR	CD	108/2	20	4	4	7	2	14	O
DCD	CF	109	8	1	1	1	5	15	I
RI	CE	125	22	9	9	8	6	18	I

<sup>1</sup> Ленточный кабель 8-битных мультикарт.

<sup>2</sup> Ленточный кабель 16-битных мультикарт и портов на системных платах.

<sup>3</sup> Вариант ленточного кабеля портов на системных платах.

<sup>4</sup> Широкий ленточный кабель к 25-контактному разъему.

Подмножество сигналов RS-232C, относящихся к асинхронному режиму, рассмотрим с точки зрения СОМ-порта РС. Для удобства будем пользоваться мнемоникой названий, принятой в описаниях СОМ-портов и большинства устройств (она отличается от безликих обозначений RS-232 и V.24). Напомним, что активному состоянию сигнала («включено») и логической единице передаваемых данных соответствует отрицательный потенциал (ниже -3 В) сигнала интерфейса, а состоянию «выключено» и логическому нулю — положительный (выше +3 В). Назначение сигналов интерфейса приведено в табл. 2.2. Нормальную последовательность управляющих сигналов для случая подключения модема к СОМ-порту иллюстрирует рис. 2.6.

Таблица 2.2. Назначение сигналов интерфейса RS-232C

Сигнал	Назначение
PG	<i>Protected Ground</i> — защитная земля, соединяется с корпусом устройства и экраном кабеля
SG	<i>Signal Ground</i> — сигнальная (схемная) земля, относительно которой действуют уровни сигналов
TD	<i>Transmit Data</i> — последовательные данные — выход передатчика
RD	<i>Receive Data</i> — последовательные данные — вход приемника
RTS	<i>Request To Send</i> — выход запроса передачи данных: состояние «включено» уведомляет модем о наличии у терминала данных для передачи. В полудуплексном режиме используется для управления направлением — состояние «включено» служит сигналом модему на переключение в режим передачи
CIS	<i>Clear To Send</i> — вход разрешения терминалу передавать данные. Состояние «включено» запрещает передачу данных. Сигнал используется для аппаратного управления потоками данных
DSR	<i>Data Set Ready</i> — вход сигнала готовности от аппаратуры передачи данных (модем в рабочем режиме подключен к каналу и закончил действия по согласованию с аппаратурой на противоположном конце канала)
DTR	<i>Data Terminal Ready</i> — выход сигнала готовности терминала к обмену данными. Состояние «включено» поддерживает коммутируемый канал в состоянии соединения
DCD	<i>Data Carrier Detected</i> — вход сигнала обнаружения несущей удаленного модема
RI	<i>Ring Indicator</i> — вход индикатора вызова (звонка). В коммутируемом канале этим сигналом модем сигнализирует о принятии вызова

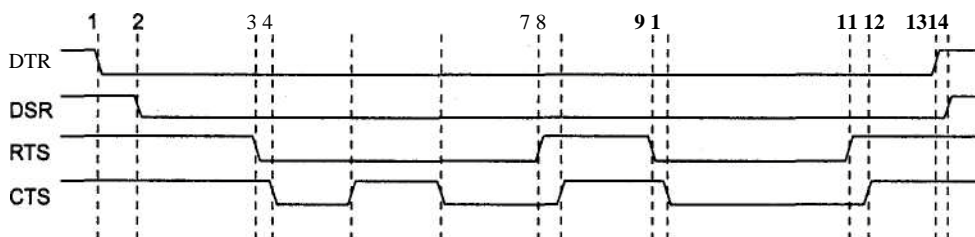


Рис. 2.6. Последовательность управляющих сигналов интерфейса

1. Установкой DTR компьютер указывает на желание использовать модем.
2. Установкой DSR модем сигнализирует о своей готовности и установлении соединения.
3. Сигналом RTS компьютер запрашивает разрешение на передачу и заявляет о своей готовности принимать данные от модема.
4. Сигналом CTS модем уведомляет о своей готовности к приему данных от компьютера и передаче их в линию.
5. Снятием CTS модем сигнализирует о невозможности дальнейшего приема (например, буфер заполнен) — компьютер должен приостановить передачу данных.
6. Сигналом CTS модем разрешает компьютеру продолжить передачу (в буфере появилось место).

7. Снятие RTS может означать как заполнение буфера компьютера (модем должен приостановить передачу данных в компьютер), так и отсутствие данных для передачи в модем. Обычно в этом случае модем прекращает пересылку данных в компьютер.
8. Модем подтверждает снятие RTS сбросом CTS.
9. Компьютер повторно устанавливает RTS для возобновления передачи.
10. Модем подтверждает готовность к этим действиям.
11. Компьютер указывает на завершение обмена.
12. Модем отвечает подтверждением.
13. Компьютер снимает DTR, что обычно является сигналом на разрыв соединения («повесить трубку»).
14. Модем сбросом DSR сигнализирует о разрыве соединения.

Из рассмотрения этой последовательности становятся понятными соединения DTR-DSR и RTS-CTS в нуль-модемных кабелях.

## 2.2. Родственные интерфейсы и преобразователи уровней

В последовательном интерфейсе далеко не всегда используют двуполярные сигналы RS-232C — это неудобно, хотя бы из-за необходимости использования дву-полярного питания приемопередатчиков. Сами микросхемы вышеописанных приемопередатчиков UART работают с сигналами логики ТТЛ или КМОП; такие же сигналы используются, например, и в сервисных портах винчестеров и других устройств. Многие устройства (в том числе карманные ПК и мобильные телефоны) имеют внешний последовательный интерфейс с уровнями низковольтной логики. Конечно, сигналы обычной логики не имеют столь высокой помехоустойчивости, как RS-232C, но не всегда это и требуется.

Для взаимного преобразования уровней интерфейса RS-232C и логики специально выпускаются буферные микросхемы приемников (с гистерезисом) и передатчиков двуполярного сигнала. При несоблюдении правил заземления и коммутации они обычно становятся первыми жертвами «пиротехнических» эффектов. Раньше их нередко устанавливали в «кроватьки», что облегчало их замену. Цоко-левка популярных микросхем формирователей сигналов RS-232C приведена на рис. 2.7. Часто буферные схемы входят прямо в состав интерфейсных БИС. Это удешевляет изделие, экономит место на плате, но в случае аварии оборачивается крупными финансовыми потерями. Вывести из строя интерфейсные микросхемы замыканием сигнальных цепей маловероятно: ток короткого замыкания передатчиков обычно не превышает 20 мА.

В специальных кабелях-адаптерах часто применяют преобразователи уровней фирмы Maxim и Surex; они удобны тем, что содержат и приемники, и передатчики. Из широкого ассортимента этих преобразователей легко подобрать подходящий по количеству приемников и передатчиков, а также по питанию (однополярному, двуполярному, низковольтному).

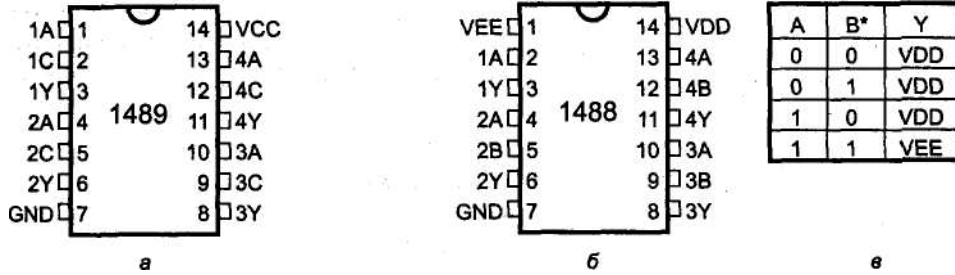


Рис. 2.7. Формирователи сигналов RS-232C: а — приемник 1489 (А — вход RS-232, С ^ управление гистерезисом (ТТЛ), Y — выход ТТЛ); б — передатчик 1488 (А, В — входы ТТЛ, Y — выход RS-232, VDD = +12 В, VEE = -12 В); в — таблица состояния выходов передатчика (\*1 В — логическая единица)

Когда требуется большая помехоустойчивость (дальность и скорость передачи), применяют иные электрические варианты последовательных интерфейсов: RS-422A (V.11, X.27), RS-423A (V.10, X.26), RS-485. На рис. 2.8 приведены схемы соединения приемников и передатчиков, а также показаны ограничения на длину линии (L) и максимальную скорость передачи данных (V). Несимметричные линии интерфейсов RS-232C и RS-423A имеют самую низкую защищенность от синфазной помехи, хотя дифференциальный вход приемника RS-423A позволяет в какой-то мере исправить ситуацию. Лучшие параметры имеют интерфейсы RS-422A и RS-485, работающие на симметричных линиях связи. В них для передачи каждого сигнала используются дифференциальные приемопередатчики с отдельной (витой) парой проводов для каждой сигнальной цепи.

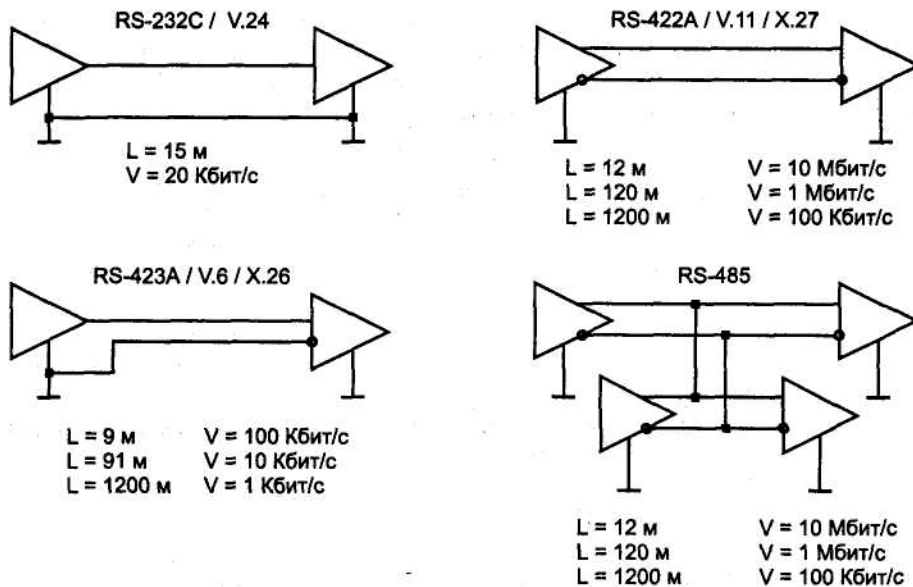


Рис. 2.8. Стандарты последовательных интерфейсов

Интерфейсы EIA-RS-422 (ITU-T V.11, X.27) и EIA-RS-485 (ISO 8482) используют симметричную передачу сигнала и допускают как двухточечную, так и шинную топологию соединений. В них информативной является разность потенциалов между проводниками А и В. Если на входе приемника  $U_A - U_B > 0,2$  В (А положительнее В) — состояние «выключено» (space),  $U_A - U_B < -0,2$  В (А отрицательнее В) — состояние «включено» (mark). Диапазон  $|U_A - U_B| < 0,2$  В является зоной нечувствительности (гистерезис), защищающей от воздействия помех. На выходах передатчика сигналы  $U_A$  и  $U_B$  обычно переключаются между уровнями 0 и +5 В (КМОП) или +1 и +4 В (ТТЛ), дифференциальное выходное напряжение должно лежать в диапазоне 1,5-5 В. Выходное сопротивление передатчиков 100 Ом. Интерфейсы электрически совместимы между собой, хотя и имеют некоторые различия в ограничениях. Принципиальное отличие передатчиков RS-485 — возможность переключения в третье состояние. Передатчики RS-422/485 совместимы с приемниками RS-423. Основные параметры интерфейсов приведены в табл. 2.3, топологию соединений иллюстрирует рис. 2.9.

Чтобы увеличить число узлов, можно повысить входное сопротивление приемников, но при этом снижается допустимая скорость или максимально возможная дальность передачи. Максимальная скорость передачи на коротких расстояниях (до 10 м) ограничивается быстродействием передатчиков (достижима частота 25 МГц). На средних расстояниях ограничение определяется емкостью кабеля (1200 бит/с - 25 нФ, 9600 бит/с - 30 нФ, 115 кбит/с - 250 пФ). Максимальная дальность (1200 м) ограничена сопротивлением петли постоянному току.

Таблица 2.3. Параметры интерфейсов RS-422 и RS-485

Параметр	RS-422	RS-485
Порог срабатывания, $ U_A - U_B $ , В	0,2	0,2
Допустимое напряжение синфазной помехи, В <sup>1</sup>	-6,8...+6,8	-6,8...+11,8
Допустимое напряжение на входах, В <sup>1</sup>	-7...+7	-7...+12
Входное сопротивление приемника, кОм	4	12
Минимальное сопротивление нагрузки передатчика, Ом	100 <sub>60</sub>	60
Максимальное число узлов	1 передатчик	32 (передатчиков,
	+10 приемников	приемников или их комбинаций)
Максимальная длина, м	1200 (100 кбит/с)	1200 (100 кбит/с)
Терминаторы, R=100 Ом	12(10Мбит/с)	12(10Мбит/с)
Ток короткого замыкания, мА	На дальнем конце от передатчика	На обоих концах
	<150 на шину GND	<250 на шину с потенциалом
		-7...+12 В или между проводами А и В

<sup>1</sup> Напряжение измеряется относительно «схемной земли» узла.

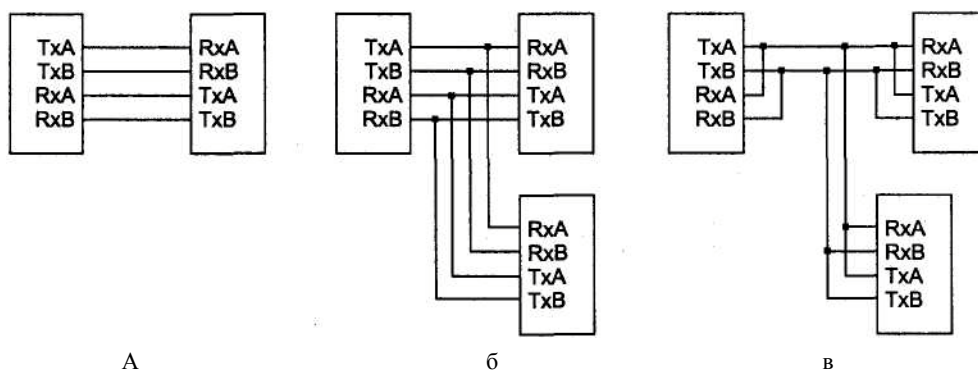


Рис. 2.9. Топология интерфейсов: а — RS-422, б — RS-485 четырехпроводной, в — RS-485 двухпроводной

Интерфейс RS-485 может быть в двух версиях: двухпроводной и четырехпроводной. Четырехпроводная версия (рис. 2.9, б) выделяет задающий узел (master), передатчик которого работает на приемники всех остальных. Передатчик задающего узла всегда активен — переход в третье состояние ему не нужен. Передатчики остальных ведомых (slave) узлов должны иметь тристабильные выходы, они объединяются на общей шине с приемником ведущего узла. В двухпроводной версии (рис. 2.9, в) все узлы равноправны.

В вырожденном случае — при двухточечном соединении — интерфейсы RS-485 и RS-422 эквивалентны, и третье состояние не используется.

Для определенности состояния покоя шины RS-485, когда нет активных передатчиков, на линию устанавливают активные терминаторы, «растягивающие» потенциалы проводов. В покое провод В должен иметь более положительный потенциал, чем А.

При многоточечном соединении необходимо организовать метод доступа к среде передачи. Чаще всего используют полинг (polling) — опрос готовности к передаче, выполняемый ведущим устройством, или передачу права доступа в соответствии с определенным (установленным) регламентом. Иногда используют и методы случайного доступа (аналогично Ethernet).

Дифференциальный вход интерфейсов защищает от действия помех, но при этом должно осуществляться соединение «схемных земель» устройств между собой и с шиной заземления. Для соединения устройств между собой используют третий провод интерфейса (можно и экран). Для того чтобы по третьему проводу не протекал большой ток, выравнивающий «земляные потенциалы», в его цепь включают резисторы (рис. 2.10).

Интерфейс RS-422 часто используется для подключения периферийных устройств (например, принтеров). Интерфейс RS-485 популярен в качестве шин устройств промышленной автоматики.

*Интерфейс «токовая петля»* для представления сигнала использует не напряжение, а ток в двухпроводной линии, соединяющей приемник и передатчик.



Логической единице (состоянию «включено») соответствует протекание тока 20 мА, а логическому нулю — отсутствие тока. Такое представление сигналов для вышеописанного формата асинхронной посылки позволяет обнаружить обрыв линии — приемник заметит отсутствие стоп-бита (обрыв линии действует как постоянный логический нуль).

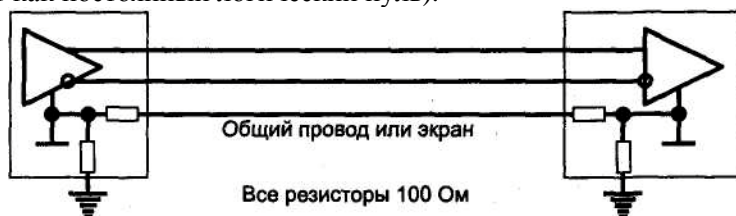


Рис. 2.10. Соединение «схемных земель» для интерфейсов RS-422 и RS-485

Токовая петля обычно предполагает *гальваническую развязку* входных цепей приемника от схемы устройства. При этом источником тока в петле является передатчик (этот вариант называют активным передатчиком). Возможно и питание от приемника (активный приемник), при этом выходной ключ передатчика может быть также гальванически развязан с остальной схемой передатчика. Существуют упрощенные варианты без гальванической развязки, но это уже вырожденный случай интерфейса. Заметим, что интерфейс MIDI (см. п. 8.5.3) с «классической» токовой петлей несовместим.

Токовая петля с гальванической развязкой позволяет передавать сигналы на расстояния до нескольких километров, но при невысоких скоростях (выше 19 200 бит/с не используют, а на километровых расстояниях допустима скорость до 9600 бит/с и ниже). Допустимое расстояние определяется сопротивлением пары проводов и уровнем помех. Поскольку интерфейс требует пары проводов для каждого сигнала, обычно используют только два сигнала последовательного интерфейса (4-проводная линия). В случае двунаправленного обмена применяются только сигналы передаваемых и принимаемых данных, а для управления потоком используется программный метод XON/XOFF. Если двунаправленный обмен не требуется, применяют одну линию данных, а для управления потоком обратная линия задействуется для сигнала CTS (аппаратный протокол) или встречной линии данных (программный протокол). При надлежащем ПО одной токовой петлей можно обеспечить двунаправленную полудуплексную связь двух устройств. При этом каждый приемник «слышит» как сигналы передатчика на противоположной стороне канала, так и сигналы своего передатчика. Они расцениваются коммуникационными пакетами просто как эхо-сигнал. Для безошибочного приема передатчики должны работать поочередно.

Токовая петля позволяет использовать выделенные физические линии без модемов, но на малых скоростях. Иногда по токовой петле подключают терминалы с интерфейсом RS-232C, если не хватает штатной длины интерфейса или требуется гальваническая развязка. Преобразовать сигналы RS-232C в токовую петлю несложно — на рис. 2.11 приведена простейшая схема преобразователя примени-

тельно к подключению терминала. Для получения двуполярного сигнала, требуемого для входных сигналов СОМ-порта, применяется питание от интерфейса. Схема может быть усложнена для защиты оптронов от перегрузки и улучшения формы потенциальных сигналов. Допустимая скорость определяется и быстродействием применяемых оптронов (скорость 9600 бит/с достигается практически на любых оптронах).

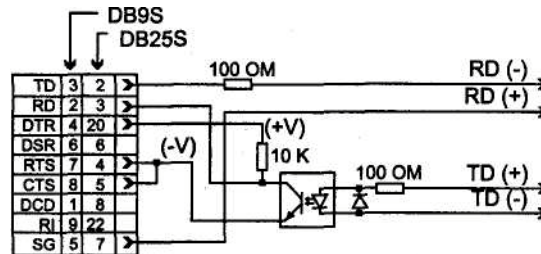


Рис. 2.11. Преобразование интерфейса RS-232C в «токовую петлю»

## 2.3. Асинхронный режим передачи

Асинхронный режим передачи является байт-ориентированным (символьно-ориентированным): минимальная пересылаемая единица информации — один байт (один символ). Формат посылки байта иллюстрирует рис. 2.12. Передача каждого байта начинается со *старт-бита*, сигнализирующего приемнику о начале посылки, за которым следуют *биты данных* и, возможно, *бит паритета* (четности). Завершает посылку *стоп-бит*, гарантирующий паузу между посылками. Старт-бит следующего байта посылается в любой момент после стоп-бита, то есть между передачами возможны паузы произвольной длительности. Старт-бит, имеющий всегда строго определенное значение (логический 0), обеспечивает простой механизм синхронизации приемника по сигналу от передатчика. Подразумевается, что приемник и передатчик работают на одной скорости обмена. Внутренний генератор синхронизации приемника использует счетчик-делитель опорной частоты, обнуляемый в момент приема начала старт-бита. Этот счетчик генерирует внутренние стробы, по которым приемник фиксирует последующие принимаемые биты. В идеале стробы располагаются в середине битовых интервалов, что позволяет принимать данные и при незначительном рассогласовании скоростей приемника и передатчика. Очевидно, что при передаче 8 бит данных, одного контрольного и одного стоп-бита предельно допустимое рассогласование скоростей, при котором данные будут распознаны верно, не может превышать 5 %. С учетом фазовых искажений и дискретности работы внутреннего счетчика синхронизации реально допустимо меньшее отклонение частот. Чем меньше коэффициент деления опорной частоты внутреннего генератора (чем выше частота передачи), тем больше погрешность привязки стробов к середине битового интервала, и требования к согласованности частот становятся более строгие. Чем выше частота передачи, тем больше влияние искажений фронтов на фазу принимаемого сигнала.

Взаимодействие этих факторов приводит к повышению требований к согласованности частот приемника и передатчика с ростом частоты обмена.



Рис. 2.12. Формат асинхронной передачи RS-232C

Формат асинхронной посылки позволяет выявлять возможные *ошибки передачи*.

- ◆ Если принят перепад, сигнализирующий о начале посылки, а по стробу стартового бита зафиксирован уровень логической единицы, стартовый бит считается ложным и приемник снова переходит в состояние ожидания. Об этой ошибке приемник может не сообщать.
- ◆ Если во время, отведенное под стоп-бит, обнаружен уровень логического нуля, фиксируется ошибка стоп-бита.
- ◆ Если применяется контроль четности, то после посылки бит данных передается *контрольный бит*. Этот бит дополняет количество единичных бит данных до четного или нечетного в зависимости от принятого соглашения. Прием байта с неверным значением контрольного бита приводит к фиксации ошибки.
- ◆ Контроль формата позволяет обнаруживать обрыв линии: как правило, при обрыве приемник «видит» логический ноль, который сначала трактуется как стартовый бит и нулевые биты данных, но потом срабатывает контроль стоп-бита.

Для асинхронного режима принят ряд *стандартных скоростей обмена*: 50, 75, 110, 150, 300, 600, 1200, 2400, 4800, 9600, 19200, 38400, 57600 и 115200 бит/с. Иногда вместо единицы измерения «бит/с» используют «бод» (baud), но при рассмотрении двоичных передаваемых сигналов это некорректно. В бодах принято измерять частоту изменения состояния линии, а при двоичном способе кодирования (широко применяемом в современных модемах) в канале связи скорости передачи бит (бит/с) и изменения сигнала (бод) могут отличаться в несколько раз.

Количество *бит данных* может составлять 5, 6, 7 или 8 (5- и 6-битные форматы распространены незначительно). Количество *стоп-бит* может быть 1, 1,5 или 2 («полтора бита» означает только длительность стопового интервала).

## 2.4. Управление потоком данных

Для управления потоком данных (Flow Control) могут использоваться два варианта протокола — аппаратный и программный. Иногда управление потоком путают с квитированием. *Квитирование* (handshaking) подразумевает посылку уведом-

ления о получении элемента, в то время как *управление потоком* предполагает посылку уведомления о возможности или невозможности последующего приема данных. Зачастую управление потоком основано на механизме квитирования.

*Аппаратный протокол управления потоком RTS/CTS* (hardware flow control) использует сигнал CTS, который позволяет остановить передачу данных, если приемник не готов к их приему (рис. 2.13). Передатчик «выпускает» очередной байт только при включенной линии CTS. Байт, который уже начал передаваться, задержать сигналом CTS невозможно (это гарантирует целостность посылки). Аппаратный протокол обеспечивает самую быструю реакцию передатчика на состояние приемника. Микросхемы асинхронных приемопередатчиков имеют не менее двух регистров в приемной части — сдвигающий, для приема очередной посылки, и хранящий, из которого считывается принятый байт. Это позволяет реализовать обмен по аппаратному протоколу без потери данных.

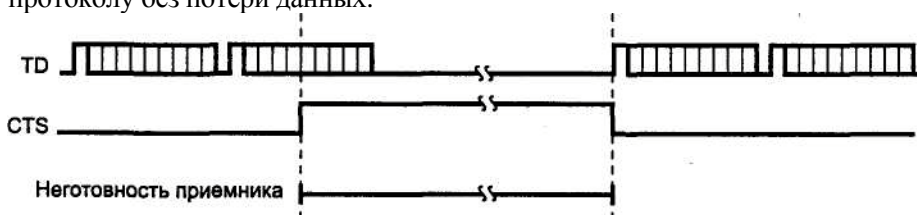


Рис. 2.13. Аппаратное управление потоком

Аппаратный протокол удобно использовать при подключении принтеров и плоттеров, если они его поддерживают. При непосредственном (без модемов) соединении двух компьютеров аппаратный протокол требует перекрестного соединения линий RTS — CTS.

При непосредственном соединении у передающего терминала должно быть обеспечено состояние «включено» на линии CTS (соединением собственных линий RTS — CTS), в противном случае передатчик будет «молчать».

Применяемые в IBM PC приемопередатчики 8250/16450/16550 сигнал CTS аппаратно не обрабатывают, а только показывают его состояние в регистре MSR (см. п. 2.5). Реализация протокола RTS/CTS возлагается на драйвер BIOS Int 14h, и называть его «аппаратным» не совсем корректно. Если же программа, пользующаяся COM-портом, взаимодействует с UART на уровне регистров (а не через BIOS), то обработкой сигнала CTS для поддержки данного протокола она занимается сама. Ряд коммуникационных программ позволяет игнорировать сигнал CTS (если не используется модем), и для них не требуется соединение входа CTS с выходом даже своего сигнала RTS. Однако существуют и иные приемопередатчики (например, 8251), в которых сигнал CTS обрабатывается аппаратно. Для них, а также для «честных» программ, использование сигнала CTS на разъемах (а то и на кабелях) обязательно. Преимущество протокола RTS/CTS во времени реакции (по сравнению с программным методом XON/XOFF) остается лишь для буферизированной (в режиме FIFO) передачи.

Программный протокол управления потоком *XON/XOFF* предполагает наличие двунаправленного канала передачи данных. Работает протокол следующим образом: если устройство, принимающее данные, обнаруживает причины, по которым оно не может их дальше принимать, оно по обратному последовательному каналу посылает байт-символ *XOFF* (13h). Противоположное устройство, приняв этот символ, приостанавливает передачу. Когда принимающее устройство снова становится готовым к приему данных, оно посылает символ *XON* (11h), приняв который противоположное устройство возобновляет передачу. Время реакции передатчика на изменение состояния приемника по сравнению с аппаратным протоколом увеличивается, по крайней мере, на время передачи символа (*XON* или *XOFF*) плюс время реакции программы передатчика на прием символа (рис. 2.14). Из этого следует, что данные без потерь могут приниматься только приемником, имеющим дополнительный буфер принимаемых данных и сигнализирующим о неготовности заблаговременно (имея в буфере свободное место).

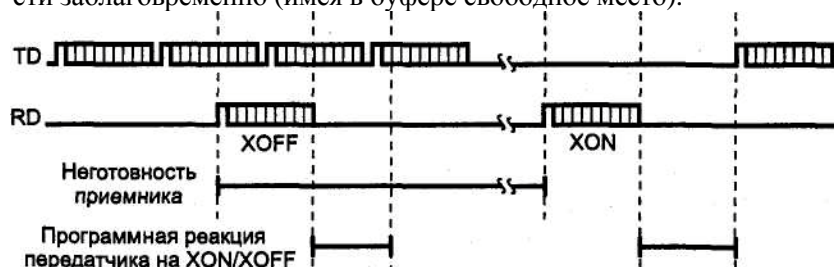


Рис. 2.14. Программное управление потоком *XON/XOFF*

Преимущество программного протокола заключается в отсутствии необходимости передачи управляющих сигналов интерфейса — минимальный кабель для двустороннего обмена может иметь только 3 провода (см. рис. 2.5, *a*). Недостатком, помимо обязательного наличия буфера и большего времени реакции (снижающего общую производительность канала из-за ожидания сигнала *XON*), является сложность реализации полнодуплексного режима обмена. В этом случае из потока принимаемых данных должны выделяться (и обрабатываться) символы управления потоком, что ограничивает набор передаваемых символов.

Кроме этих двух распространенных стандартных протоколов, поддерживаемых и ПУ, и ОС, существуют и другие.

## 2.5. Микросхемы асинхронных приемопередатчиков

В COM-портах преобразование параллельного кода в последовательный для передачи и обратное преобразование при приеме данных выполняют специализированные микросхемы UART (Universal Asynchronous Receiver-Transmitter — универсальный асинхронный приемопередатчик). Эти же микросхемы формиру-

ют и обрабатывают управляющие сигналы интерфейса. СОМ-порты IBM PC XT/ AT базируются на микросхемах, совместимых на уровне регистров с UART i8250 — 8250/16450/16550A. Это семейство представляет собой усовершенствование начальной модели, направленное на повышение быстродействия, снижение потребляемой мощности и загрузки процессора при интенсивном обмене. Отметим, что:

- ◆ 8250 имеет ошибки (появление ложных прерываний), учтенные в XT BIOS;
- ◆ 8250A — ошибки исправлены, но в результате потеряна совместимость с BIOS; эта микросхема работает в некоторых моделях AT, но непригодна для скорости 9600 бит/с;
- ◆ 8250B — исправлены ошибки 8250 и 8250A, восстановлена ошибка в прерываниях — возвращена совместимость с XT BIOS; работает в AT под DOS (кроме скорости 9600 бит/с).

Микросхемы 8250x имеют невысокое быстродействие по обращениям со стороны системной шины. Они не допускают обращения к своим регистрам в смежных шинных циклах процессора — для корректной работы с ними требуется введение программных задержек между обращениями CPU.

В компьютерах класса AT применяют микросхемы UART перечисленных ниже модификаций.

- ◆ 16450 — быстродействующая версия 8250 для AT. Ошибок 8250 и полной совместимости с XT BIOS не имеет.
- ◆ 16550 — развитие 16450. Может использовать канал DMA для обмена данными. Имеет FIFO-буфер, но некорректность его работы не позволяет им воспользоваться.
- ◆ 16550A — имеет работающие 16-байтные FIFO-буферы приема и передачи и возможность использования DMA. Именно этот тип UART должен применяться в AT при интенсивных обменах на скоростях 9600 бит/с и выше. Совместимость с этой микросхемой обеспечивает большинство микросхем контроллеров портов ввода-вывода, входящих в современные чипсеты.

Микросхемы UART 16550A с программной точки зрения представляют собой набор регистров, доступ к которым определяется адресом (смещением адреса регистра относительно базового адреса порта) и значением бита DLAB (бита 7 регистра LCR). В адресном пространстве микросхема занимает 8 смежных адресов. Список регистров UART 16550A и способы доступа к ним приведены в табл. 2.4. Микросхемы 8250 отличаются отсутствием регистра FCR и всех возможностей FIFO и DMA.

Доступ		Регистр		Название	Чтение/запись R/W
Смещение	DLAB	Имя			
0h	0 0	THR	Transmit Holding Register	WO	
0h	1	RBR	Receiver Buffer Register	RO	
0h		DLL	Divisor Latch LSB	R/W	

продолжение-^

Таблица 2.4 (продолжение)

Доступ	Регистр		Чтение/запись R/W	
	Смещение	Имя Название		
1h	1	DIM	Divisor Latch MSB	R/W
1h	0	IER	Interrupt Enable Register	R/W
2h	X	IIR	Interrupt Identification Register	RO
2h	X	FOR	FIFO Control Register	WO
3h	X	LCR	Line Control Register	R/W
4h	X	MCR	Modem Control Register	R/W
5h	X	LSR	Line Status Register	R/W
6h	X	MSR	Modem Status Register	R/W
7h	X	SCR	Scratch Pad Register	R/W

<sup>1</sup> Некоторые биты допускают только чтение. Запись в регистр может привести к сбою протокола.

**T N R** — *промежуточный регистр данных передатчика* (только для записи). Данные, записанные в регистр, будут пересланы в выходной сдвигающий регистр (когда он будет свободен), из которого поступят на выход при наличии разрешающего сигнала CTS. Бит 0 передается (и принимается) первым. При длине посылки менее 8 бит старшие биты игнорируются.

**RBR** — *буферный регистр принимаемых данных* (только для чтения). Данные, принятые входным сдвигающим регистром, помещаются в регистр RBR, откуда они могут быть считаны процессором. Если к моменту окончания приема очередного символа предыдущий не был считан из регистра, фиксируется ошибка переполнения. При длине посылки менее 8 бит старшие биты в регистре имеют нулевое значение.

**DLL** — *регистр младшего байта делителя частоты*.

**D L M** — *регистр старшего байта делителя частоты*. Делитель определяется по формуле  $D=115200/V$ , где  $V$  — скорость передачи, бит/с. Входная частота синхронизации 1,8432 МГц делится на заданный коэффициент, после чего получается 16-кратная частота передачи данных.

**IER** — регистр разрешения прерываний. Единичное значение бита разрешает прерывание от соответствующего источника.

Назначение бит регистра IER:

- ◆ биты [7:4]=0 — не используются;
- ◆ бит 3 — Mod\_IE — по изменению состояния модема (любой из линий CTS, DSR, RI, DCD);
- ◆ бит 2 — Rx\_L\_I E — по обрыву/ошибке линии;
- ◆ бит 1 — TxD\_I E — по завершении передачи;
- ◆ бит 0 — RxD\_I E — по приему символа (в режиме FIFO — прерывание по тайм-ауту).

**IIR** — *регистр идентификации прерываний и признака режима FIFO* (только для чтения). Для упрощения программного анализа UART выстраивает внутренние запросы прерывания по четырехуровневой системе приоритетов. Порядок

приоритетов (по убыванию): состояние линии, прием символа, освобождение регистра передатчика, состояние модема. При возникновении условий прерывания UART указывает на источник с высшим приоритетом до тех пор, пока он не будет сброшен соответствующей операцией. Только после этого будет выставлен запрос с указанием следующего источника. Ниже описано назначение бит регистра IIR.

- ◆ Биты [7:6] — признак режима FIFO:
  - 11-режим FIFO 16550A;
  - 10 - режим FIFO 16550;
  - 00 — обычный.
- ◆ Биты [5:4] — не используются.
- ◆ Бит 3 — прерывание по тайм-ауту приема в режиме FIFO (в буфере есть символы для считывания).
- ◆ Биты [2:1] — причина прерывания с наивысшим приоритетом (в обычном, не FIFO-режиме):
  - 11 — ошибка/обрыв линии, сброс выполняется чтением регистра состояния линии;
  - 10 — принят символ, сброс выполняется чтением данных;
  - 01 — передан символ (регистр THR пуст), сброс выполняется записью данных;
  - 00 — изменение состояния модема; сброс выполняется чтением регистра состояния модема.
- ◆ Бит 0 — признак необслуженного запроса прерывания (1 — нет запроса, 0 — есть запрос).

В режиме FIFO причину прерывания идентифицируют биты [3:1].

- ◆ 011 — ошибка/обрыв линии. Сброс выполняется чтением регистра состояния линии.
- ◆ 010 — принят символ. Сброс выполняется чтением регистра данных приемника.
- ◆ 110 — индикатор тайм-аута (за 4-кратный интервал времени символа не передано и не принято ни одного символа, хотя в буфере имеется, по крайней мере, один). Сброс выполняется чтением регистра данных приемника.
- ◆ 001 — регистр THR пуст. Сброс выполняется записью данных.
- ◆ 000 — изменение состояния модема (CIS, DSR, RI или DCD). Сброс выполняется чтением регистра MSR.

FCR — *регистр управления FIFO* (только для записи). Ниже описано назначение бит регистра FCR:

- ◆ Биты [7:6] — ITL (Interrupt Trigger Level) — уровень заполнения FIFO-буфера, при котором вырабатывается прерывание:
  - 00—1 байт (по умолчанию);
  - 01 — 4 байта;



- 10 -8 байт;
- 11-14 байт.
- ◆ Биты [5:4] зарезервированы.
- ◆ Бит 3 — разрешение операций DMA.
- ◆ Бит 2 - RESETTF (Reset Transmitter FIFO) - сброс счетчика FIFO-передатчика (записью единицы; сдвигающий регистр не сбрасывается).
- ◆ Бит 1 — RESETRF (Reset Receiver FIFO) — сброс счетчика FIFO-приемника (записью единицы; сдвигающий регистр не сбрасывается).
- ◆ Бит 0 — TRFIFOE (Transmit And Receive FIFO Enable) — разрешение (единицей) режима FIFO для передатчика и приемника. При смене режима FIFO-буферы автоматически очищаются.

LCR — *регистр управления линией* (настройки параметров канала). Ниже описано назначение бит регистра LCR.

- ◆ Бит 7 — DLAB (Divisor Latch Access Bit) — управление доступом к делителю частоты.
- ◆ Бит 6 — BRCON (Break Control) — формирование обрыва линии (посылка нулей) при BRCON=1.
- ◆ Бит 5 — STICPAR (Sticky Parity) — принудительное формирование бита паритета:
  - 0 — контрольный бит генерируется в соответствии с паритетом выводимого символа;
  - 1 — постоянное значение контрольного бита: при EVENPAR=1 — нулевое, при EVENPAR=0 — единичное.
- ◆ Бит 4 — EVENPAR (Even Parity Select) — выбор типа контроля: 0 — нечетность, 1 — четность.
- ◆ Бит 3 — PAREN (Parity Enable) — разрешение контрольного бита:
  - 1 — контрольный бит (паритет или постоянный) разрешен;
  - 0 — контрольный бит запрещен.
- ◆ Бит 2 — STOP B (Stop Bits) — количество стоп-бит:
  - 0—1 стоп-бит;
  - 1 — 2 стоп-бита (для 5-битного кода стоп-бит будет иметь длину 1,5 бит).
- ◆ Биты [1:0] — SERIALDB (Serial Data Bits) — количество бит данных:
  - 00-5 бит;
  - 01-6 бит;
  - 10-7 бит;
  - 11-8 бит.

MCR — *регистр управления модемом*. Ниже описано назначение бит регистра MCR.

- ◆ Биты [7:5]=0 — зарезервированы.
- ◆ Бит 4 — LME (Loopback Mode Enable) — разрешение режима диагностики:
  - 0 — нормальный режим;
  - 1 — режим диагностики (см. ниже).
- ◆ Бит 3 — IE (Interrupt Enable) — разрешение прерываний с помощью внешнего выхода OUT2; в режиме диагностики поступает на вход MSR. 7:
  - 0 — прерывания запрещены;
  - 1 — прерывания разрешены.
- ◆ Бит 2 — OUT1C (OUT1 Bit Control) — управление выходным сигналом 1 (не используется); в режиме диагностики поступает на вход MSR. 6.
- ◆ Бит 1 — RISC (Request To Send Control) — управление выходом RTS; в режиме диагностики поступает на вход MSR.4:
  - 0 — активен (-V);
  - 1 — пассивен (+V).
- ◆ Бит 0 — DTRC (Data Terminal Ready Control) — управление выходом DTR; в режиме диагностики поступает на вход MSR. 5:
  - 0 — активен (-V);
  - 1 — пассивен (+V).

LSR — *регистр состояния линии* (точнее, состояния приемопередатчика).

Ниже описано назначение бит регистра LSR.

- ◆ Бит 7 — FIFOE (FIFO Error Status) — ошибка принятых данных в режиме FIFO (буфер содержит хотя бы один символ, принятый с ошибкой формата, паритета или обрывом). В не FIFO-режиме всегда 0.
- ◆ Бит 6 — TEMPT (Transmitter Empty Status) — регистр передатчика пуст (нет данных для передачи ни в сдвиговом регистре, ни в буферных регистрах THR или FIFO).
- ◆ Бит 5 — THRE (Transmitter Holding Register Empty) — регистр передатчика готов принять байт для передачи. В режиме FIFO указывает на отсутствие символов в FIFO-буфере передачи. Может являться источником прерывания.
- ◆ Бит 4 — BD (Break Detected) — индикатор обрыва линии (вход приемника находится в состоянии 0 не менее чем время посылки символа).
- ◆ Бит 3 — FE (Framing Error) — ошибка кадра (неверный стоп-бит).
- ◆ Бит 2 — PE (Parity Error) — ошибка контрольного бита (паритета или фиксированного).

- ◆ Бит 1 — ОЕ (Overrun Error) — переполнение (потеря символа). Если прием очередного символа начинается до того, как предыдущий выгружен из сдвигающего регистра в буферный регистр или в регистр FIFO, прежний символ в сдвигающем регистре теряется.
- ◆ Бит 0 — DR (Receiver Data Ready) — принятые данные готовы (в DHR или FIFO-буфере). Сброс — чтением приемника.

Индикаторы ошибок — биты [4:1] — сбрасываются после чтения регистра LSR. В режиме FIFO признаки ошибок хранятся в FIFO-буфере вместе с каждым символом. В регистре они устанавливаются (и вызывают прерывание) в тот момент, когда символ, принятый с ошибкой, находится на вершине FIFO (первый в очереди на считывание). В случае обрыва линии в FIFO заносится только один «обрывной» символ, и UART ждет восстановления и последующего старт-бита.

MSR — *регистр состояния модема*. Ниже описано назначение бит регистра MSR:

- ◆ Бит 7 — DCD (Data Carrier Detect) — состояние линии DCD:
  - 0 — активна (-V);
  - 1 — пассивна (+V).
- ◆ Бит 6 — RI (Ring Indicator) — состояние линии RI:
  - 0 — активна (-V);
  - 1 — пассивна (+V).
- ◆ Бит 5 — DSR (Data Set Ready) — состояние линии DSR:
  - 0 — активна (-V);
  - 1 — пассивна (+V).
- ◆ Бит 4 — CIS (Clear To Send) — состояние линии CTS:
  - 0 — активна (-V);
  - 1 — пассивна (+V).
- ◆ Бит 3 — DDCD (Delta Data Carrier Detect) — изменение состояния DCD.
- ◆ Бит 2 — TE RI (Trailing Edge Of Ring Indicator) — спад огибающей RI (окончание звонка).
- ◆ Бит 1 — DDSR (Delta Data Set Ready) — изменение состояния DSR.
- ◆ Бит 0 — DCTS (Delta Clear To Send) — изменение состояния CTS.

Признаки изменения (биты [3:0]) сбрасываются по чтению регистра.

SCR — *рабочий регистр* (8 бит), на работу UART не влияет, предназначен для временного хранения данных (в 8250 отсутствует).

В *диагностическом режиме* (при LME=1) внутри UART организуется внутренняя «заглушка»:

- ◆ выход передатчика переводится в состояние логической единицы;
- ◆ вход приемника отключается;

- ♦ выход сдвигающего регистра передатчика логически соединяется со входом приемника;
- ♦ входы DSR, CIS, RI и DCD отключаются от входных линий и внутренне управляются битами DTRC, RISC, OUT1C, IE;
- ♦ выходы управления модемом переводятся в пассивное состояние (логический ноль).

Переданные данные в последовательном виде немедленно принимаются, что позволяет проверять внутренний канал данных порта (включая сдвигающие регистры) и обработку прерываний, а также определять скорость работы UART.

## 2.6. Системная поддержка COM-портов

COM-порты поддерживаются *сервисом* BIOS Int 14h, который обеспечивает следующие функции:

- ♦ *инициализация* (установка скорости обмена и формата посылок; запрет источников прерываний) — на сигналы DTR и RTS влияния не оказывает (после аппаратного сброса они пассивны);
- ♦ *вывод символа* — активируются сигналы DTR и RTS, и после освобождения регистра THR в него помещается выводимый символ;
- ♦ *ввод символа* — активируется только сигнал DTR (RTS переходит в пассивное состояние), и ожидается готовность принятых данных;
- ♦ *опрос состояния* модема и линии (чтение регистров MSR и LSR).

Аппаратные прерывания не используются, ожидание готовности ввода и вывода ограничивается по тайм-ауту. Готовность можно быстро проверить опросом состояния.

В процессе начального тестирования POST BIOS проверяет наличие последовательных портов (регистров UART 8250 или совместимых) по стандартным адресам и помещает базовые адреса обнаруженных портов в ячейки BIOS Data Area 0:0400, 0402, 0404, 0406. Эти ячейки хранят адреса портов с логическими именами COM1-COM4. Нулевое значение адреса является признаком отсутствия порта с данным номером. В ячейки 047C, 047D, 047E, 047F заносятся константы, задающие тайм-ауты для портов.

Обнаруженные порты *инициализируются* на скорость обмена 2400 бит/с, 7 бит данных с контролем на четность (even), 1 стоп-бит. Управляющие сигналы интерфейса DTR и RTS переводятся в исходное состояние («выключено» — положительное напряжение).

## 2.7. Конфигурирование COM-портов

Компьютер может иметь до четырех последовательных портов COM 1 - COM4; для машин класса AT типично наличие двух портов. Управление последовательным портом разделяется на два этапа — предварительное конфигурирование (Setup)

аппаратных средств порта и текущее (оперативное) переключение режимов работы прикладным или системным ПО. Конфигурирование COM-порта зависит от его исполнения. Порт на плате расширения конфигурируется джамперами на самой плате. Порт на системной плате конфигурируется через BIOS Setup.

Конфигурированию подлежат перечисленные ниже параметры:

- ◆ *Базовый адрес*, который для портов COM1-COM4 обычно имеет значение 3F8h, 2F8h, 3E8h и 2E8h. При инициализации BIOS проверяет наличие портов по адресам именно в этом порядке и присваивает обнаруженным портам логические имена COM1, COM2, COM3 и COM4. Для портов COM3 и COM4 возможны альтернативные адреса 3E0h, 338h и 2E0h, 238h соответственно. Для PS/2 стандартными для портов COM3-COM8 являются адреса 3220h, 3228h, 4220h, 4228h, 5220h и 5228h соответственно.
- ◆ *Используемая линия запроса прерывания*: для COM 1 и COM3 обычно используется IRQ4 или IRQ11, для COM2 и COM4 — IRQ3 или IRQ10. В принципе номер прерывания можно назначать в произвольных сочетаниях с базовым адресом (номером порта), но некоторые программы и драйверы (например, драйверы последовательной мыши) настроены на стандартные сочетания. Каждому порту, нуждающемуся в аппаратном прерывании, назначают отдельную линию, не совпадающую с линиями запроса прерываний других устройств. Прерывания необходимы для портов, к которым подключаются устройства ввода, UPS или модемы. При подключении принтера или плоттера прерываниями пользуются только многозадачные ОС (не всегда), и этот дефицитный ресурс PC можно экономить. Также прерывания обычно не задействуют при связи двух компьютеров нуль-модемным кабелем. Возможность разделяемого использования одной линии запроса несколькими портами (или ее разделения с другими устройствами) зависит от реализации аппаратного подключения и ПО. При использовании портов, установленных на шину ISA, разделяемые прерывания обычно не работают.
- ◆ *Канал DMA* (для микросхем UART 16450/16550, расположенных на системной плате) — разрешение использования и номер канала DMA. Режим DMA при работе с COM-портами используют редко.

## 2.8. Использование COM-портов

COM-порт широко применяется для подключения различных периферийных и коммуникационных устройств, связи с различным технологическим оборудованием, объектами управления и наблюдения, программаторами, внутрисхемными эмуляторами и прочими устройствами, используя протокол RS-232C. COM-порты чаще всего применяют для подключения *манипуляторов* (мышь, трекбол). В этом случае порт используется в режиме последовательного ввода. Мышь с последовательным интерфейсом — *Serial Mouse* — может подключаться к любому исправному порту. Для согласования разъемов порта и мыши возможно применение переходника DB-9S-DB-25P или DB-25S-DB-9P. Для мыши требу-

ется прерывание, для порта COM1 — IRQ4, для COM2 — IRQ3. То, что для использования мыши порт COM1 должен использовать прерывание IRQ4, является особенностью ее драйвера, но для пользователя важен сам факт ограничения. Каждое событие — перемещение мыши или нажатие-отпускание кнопки — кодируется двоичной посылкой по интерфейсу RS-232C. Применяется асинхронная передача; двуполярное питание обеспечивается от управляющих линий интерфейса.

Для подключения *внешних модемов* используется полный (9-проводный) кабель АПД-АКД, схема которого приведена на рис. 2.4. Этот же кабель используется для согласования разъемов (по количеству контактов); возможно применение переходников 9-25, предназначенных для мыши. Для работы коммуникационного ПО обычно требуются прерывания, но здесь есть свобода выбора номера (адреса) порта и линии прерывания. Если предполагается работа на скоростях 9600 бит/с и выше, то COM-порт должен быть реализован на микросхеме UART 16550A или совместимой. Возможности работы с использованием FIFO-буферов и обмена по каналам DMA зависят от коммуникационного ПО.

Для связи *двух компьютеров*, удаленных друг от друга на небольшое расстояние, используют и непосредственное соединение их COM-портов нуль-модемным кабелем (см. рис. 2.5). Программы типа Norton Commander или Interlnk MS-DOS позволяют обмениваться файлами со скоростью до 115,2 Кбит/с без применения аппаратных прерываний. Это же соединение может использоваться и сетевым пакетом Lantastic, предоставляющим более развитый сервис, и средствами ОС Windows.

COM-порт позволяет выполнять подключение *электронных ключей* (Security Devices), предназначенных для защиты от нелегального использования ПО. Эти устройства могут быть как «прозрачными» (обеспечивающими возможность подключения периферии через тот же порт), так и полностью занимающими порт.

COM-порт при наличии соответствующей программной поддержки позволяет превратить РС в *терминал*, эмулируя систему команд распространенных специализированных терминалов (VT-52, VT-100 и т. д.). Простейший терминал получается, если замкнуть друг на друга функции BIOS обслуживания COM-порта (Int 14h), телетайпного вывода (Int 10h) и клавиатурного ввода (Int 16h). Однако такой терминал будет работать лишь на малых скоростях обмена (если, конечно, его делать не на Pentium), поскольку функции BIOS хоть и универсальны, но не слишком быстры.

COM-порт может использоваться и как двунаправленный интерфейс, у которого имеется три программно-управляемых выходных линии (TD, DTP, RTS) и четыре программно-читаемых входных линии (CIS, DSR, DCD, RI) с двуполярными сигналами. Их можно использовать, например, для программной реализации синхронных последовательных интерфейсов (см. п. 11.5) и других целей. Во времена AT-286 была известна схема однобитного широтно-импульсного преобразователя, позволяющего записывать звуковой сигнал на диск РС, используя входную линию COM-порта. Воспроизведение этой записи через обычный динамик РС позволяло передать речь.

## 2.9. COM-порт и PnP

Современные ПУ, подключаемые к COM-порту, могут поддерживать спецификацию PnP. Основная задача ОС заключается в идентификации подключенного устройства, для чего разработан несложный протокол, реализуемый на любых COM-портах чисто программным способом. Этот протокол иллюстрирует рис. 2.15.

1. Порт инициализируется с состоянием линий DTR=ON, RTS=OFF, TXD=Mark —состояние покоя (Idle).
2. Некоторое время (0,2 с) ожидается появление сигнала DSR, которое указало бы на наличие устройства, подключенного к порту. В простейшем случае устройство имеет на разъеме переключку DTR-DSR, обеспечивающую указанный ответ. Если устройство обнаружено, выполняются манипуляции управляющими сигналами DTR и RTS для получения информации от устройства. Если ответ не получен, ОС, поддерживающая динамическое реконfigurирование, периодически опрашивает состояние порта для обнаружения новых устройств.
3. Порт программируется на режим 1200 бит/с, 7 бит данных, без паритета, 1 стоп-бит, и на 0,2 с снимается сигнал DTR. После этого устанавливается DTR=1, а еще через 0,2 с устанавливается и RTS=1.

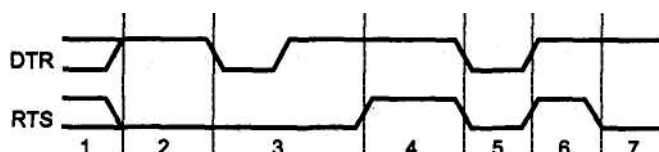


Рис.2.15. Запрос идентификатора устройства PnP

4. В течение 0,2 с ожидается приход первого символа от устройства. По приходу символа начинается прием идентификатора (см. ниже). Если за это время символ не пришел, выполняется вторая попытка опроса (см. п. 5), несколько отличающаяся от первой.
5. На 0,2 с снимаются оба сигнала (DTR=0 и RTS=0), после чего они оба устанавливаются (DTR=1 и RTS=1).
6. В течение 0,2 с ожидается приход первого символа от устройства, по приходу символа начинается прием идентификатора (см. ниже). Если за это время символ не пришел, то в зависимости от состояния сигнала DSR переходят к проверке отключения *Verify Disconnect* (при DSR=0) или в дежурное состояние *Connect Idle* (при DSR=1).
7. В дежурном состоянии *Connect Idle* устанавливается DTR=1, RTS=0, порт программируется на режим 300 бит/с, 7 бит данных, без паритета, 1 стоп-бит. Если в этом состоянии обнаружится DSR=0, ОС следует уведомить об отключении устройства.

Посимвольный прием идентификатора устройства имеет ограничения по таймауту в 0,2 с на символ, а также общее ограничение в 2,2 с, позволяющее принять

строку длиной до 256 символов. Строка идентификатора PnP должна иметь маркеры начала (28h или 08h) и конца (29h или 09h), между которыми располагается тело идентификатора в стандартизованном формате. Перед маркером начала может находиться до 16 символов, не относящихся к идентификатору PnP. Если за первые 0,2 с ожидания символа (шаг 4 или 6) маркер начала не пришел, или же сработал тайм-аут, а маркер конца не получен, или же какой-либо символ принят с ошибкой, происходит переход в состояние *Connect Idle*. Если получена корректная строка идентификатора, она передается ОС.

Для проверки отключения (*Verify Disconnect*) устанавливается DTR=1, RTS=0 и через 5 с проверяется состояние сигнала DSR. При DSR=1 происходит переход в состояние *Connect Idle* (см. п. 7), при DSR=0 происходит переход в состояние *Disconnect Idle*, в котором система может периодически опрашивать сигнал DSR для обнаружения подключения устройства.

Описанный механизм разрабатывался компанией Microsoft с учетом совместимости с устройствами, не относящимися к устройствам PnP, — он обеспечивает невозможность их вывода из строя и устойчивость системы к сообщениям, не являющимся PnP-идентификаторами. Например, обычная мышь Microsoft Mouse (не PnP) по включении питания от интерфейса ответит ASCII-символом «M» (трехкнопочная — строкой «M3»).

## 2.10. Неисправности и тестирование COM-портов

Неполадки с COM-портами случаются (выявляются) при установке новых портов или неудачном подключении внешних устройств.

### 2.10.1. Проверка конфигурирования

Тестирование последовательных портов (как и параллельных) начинают с проверки их опознавания системой. Список *адресов* установленных портов обычно появляется в таблице, выводимой BIOS перед загрузкой ОС. Список можно посмотреть с помощью тестовых программ или прямо в BIOS Data AREA с помощью отладчика.

Если BIOS обнаруживает меньше портов, чем установлено физически, значит, двум портам присвоен один адрес или установлен нестандартный адрес какого-либо порта. Проблемы могут возникать с адресами портов COM3 и COM4: не все версии BIOS будут искать порты по альтернативным адресам 3E0h, 338h, 2E0h и 238h; иногда не производится поиск по адресам 3E8h и 2E8h. Нумерация найденных портов, отображаемая в заставке, может вводить в заблуждение: если установлены два порта с адресами 3F8h и 3E8h, в заставке они могут называться COM1 и COM2, и по этим именам на них можно ссылаться. Однако те же порты в заставке могут называться COM1 и COM3 (поскольку 3E8h является штатным адресом для COM3), но попытка сослаться на порт COM3 будет неудачной, поскольку в дан-



ном случае адрес 3E8h будет находиться в ячейке 0:402h BIOS Data Area, соответствующей порту COM2, а в ячейке порта COM3 (0:404h) будет нуль — признак отсутствия такого порта. «Объяснить» системе, где какой порт, можно вручную с помощью любого отладчика, занеся правильные значения базовых адресов в ячейки BIOS Data Area (это придется делать каждый раз после перезагрузки ОС перед использованием «потерянного» порта). Существуют тестовые утилиты, позволяющие находить порты (например, Port Finder).

Если двум портам назначен один и тот же адрес, тестовая программа обнаружит ошибки порта только с помощью *внешней заглушки (ExternalLoopBack)*. Программное тестирование порта без заглушки не покажет ошибок, поскольку при этом включается диагностический режим (см. описание UART) и конфликтующие (по отдельности исправные) порты будут работать параллельно, обеспечивая совпадение считываемой информации. В «реальной жизни» нормальная работа конфликтующих портов невозможна. Разбираться с конфликтом адресов удобно, последовательно устанавливая порты и наблюдая за адресами, появляющимися в списке.

Если физически установлен только один порт и его не обнаруживает BIOS, причины те же, что с LPT-портом: либо он отключен при конфигурировании, либо вышел из строя. Неисправность может устраниться при вынимании/вставке платы адаптера в слот системной шины.

При работе с COM-портом задействуются соответствующие *аппаратные прерывания* — их используют при подключении модема, мыши и других устройств ввода. Неработоспособность этих устройств может быть вызвана некорректной настройкой запроса прерывания. Здесь возможны как конфликты с другими устройствами, так и несоответствие номера прерывания адресу порта.

## 2.10.2. Функциональное тестирование

В первом приближении COM-порт можно проверить диагностической программой (CheckIt) без использования заглушек. Этот режим тестирования проверяет микросхему UART (внутренний диагностический режим) и выработку прерываний, но не входные и выходные буферные микросхемы, которые являются более частыми источниками неприятностей. Если тест не проходит, причину следует искать или в конфликте адресов/прерываний, или в самой микросхеме UART.

Для более достоверного тестирования рекомендуется использовать *внешнюю заглушку*, подключаемую к разъему COM-порта (рис. 2.16). В отличие от LPT-порта у COM-порта количество входных сигналов превышает количество выходных, что позволяет выполнить полную проверку всех цепей. Заглушка соединяет выход приемника с входом передатчика. Обязательная для всех схем заглушек перемычка RTS-CTS позволяет работать передатчику — без нее символы не смогут передаваться. Выходной сигнал DTR обычно используют для проверки входных линий DSR, DCD, RI.

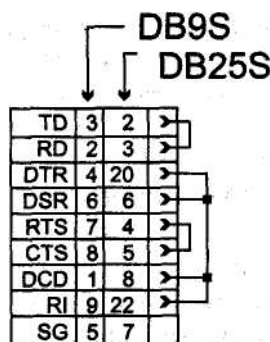


Рис. 2.16. Заглушка для проверки COM -портов (LoopBack для CheckIt и Norton Diagnostics)

Если тест с внешней заглушкой не проходит, причину следует искать во внешних буферах, их питании или в ленточных кабелях, служащих для подключения внешних разъемов. Здесь может помочь осциллограф или вольтметр. Последовательность проверки может быть следующей:

1. Проверить наличие двуполярного питания выходных схем передатчиков (этот шаг логически первый, но/поскольку он технически самый сложный, его можно отложить на крайний случай, когда появится желание заменить буферные микросхемы).
2. Проверить напряжение на выходах TD, RTS и DTR: после аппаратного сброса на выходе TD должен быть отрицательный потенциал около -12 В (по крайней мере, ниже -5 В), а на выходах RTS и DTR — такой же положительный. Если этих потенциалов нет, возможна ошибка подключения разъема к плате через ленточный кабель. Распространенные варианты:

- ленточный кабель не подключен;
- ленточный кабель подключен неправильно (разъем перевернут или вставлен со смещением);
- раскладка ленточного кабеля не соответствует разъему платы.

Первые два варианта проверяются внимательным осмотром, третий же может потребовать некоторых усилий. В табл. 2.1 приведены три варианта раскладки 10-проводного ленточного кабеля для разъема COM-порта, известных автору; для COM-портов на системных платах возможно существование и других. Теоретически ленточный кабель должен поставляться в соответствии с разъемом.

Если дело в ошибочной раскладке, то эти три выходных сигнала удастся обнаружить на других контактах разъемов (на входных контактах потенциал совсем небольшой). Если эти сигналы обнаружить не удалось, очевидно, вышли из строя буферные формирователи.

3. Соединив контакты линий RTS и CTS (или установив заглушку), следует попытаться вывести небольшой файл на COM-порт (например, командой COPY C: \AUTOEXEC. BAT COM1:). С исправным портом эта команда успешно выполнится за несколько секунд с сообщением об успешном копировании. При этом

потенциалы на выходах RTS и DTR должны измениться на отрицательные, а на выходе TD должна появиться пачка двуполярных импульсов с амплитудой более 5 В. Если потенциалы RTS и DTR не изменились, ошибка в буферных формирователях. Если на выходе RTS (и входе CTS) появился отрицательный потенциал, а команда COPY завершается с ошибкой, скорее всего, вышел из строя приемник линии CTS (или опять-таки ошибка в ленточном кабеле). Если команда COPY успешно проходит, а изменения на выходе TD не обнаруживаются (их можно увидеть стрелочным вольтметром, но оценить амплитуду импульсов не удастся), виноват буферный передатчик сигнала TD.

Замена микросхем приемников и передатчиков существенно облегчается, если они установлены в «кроватки». Перед заменой следует с помощью осциллографа или вольтметра удостовериться в неисправности конкретной микросхемы.

Если буферные элементы включены в состав интерфейсной БИС (что теперь весьма распространено), то такой порт ремонту не подлежит (по крайней мере, в обычных условиях). Неисправный COM-порт, установленный на системной плате, можно попытаться отключить в BIOS SETUP, но порт мог сгореть и вместе со схемой своего отключения — тогда он останется «живым мертвецом» в карте портов ввода-вывода и прерываний. Иногда он полностью выводит из строя системную плату.

Источниками ошибок могут являться *разъемы* и *кабели*. В разъемах встречаются плохие контакты, а кабели, кроме возможных обрывов, могут иметь плохие частотные характеристики. Частотные свойства кабелей обычно сказываются при большой длине (десятки метров) на высоких скоростях обмена (56 или 115 Кбит/с). При необходимости использования длинных кабелей на высоких скоростях сигнальные провода данных должны быть перевиты с отдельными проводами «схемной земли».

В ряде отечественных PC-совместимых (почти) компьютеров для последовательного интерфейса применялась микросхема KP580BB51 — аналог 18251. Однако эта микросхема является универсальным синхронно-асинхронным приемопередатчиком (УСАПП или USART — Universal Asynchronous Receiver-Transmitter). Совместимости с PC на уровне регистров COM-порта такие компьютеры не имеют. Хорошо, если у соответствующих компьютеров имеется «честный» драйвер BIOS Int 14h, а не заглушка, возвращающая состояние модема «всегда готов» и ничего не делающая.

### 2.10.3. Питание от интерфейса, или Причины неработоспособности мыши

При подключении к COM-порту устройств с небольшим энергопотреблением возникает соблазн получить питание от выходных линий интерфейса. Если линии управления DTR и RTS не используются по прямому назначению, их можно задействовать как питающие с напряжением около 12 В. Ток короткого замыкания на «схемную землю» ограничен буферной микросхемой передатчика на уров-

не 20 мА. При инициализации порта эти линии переходят в состояние «выключено», то есть вырабатывают *положительное* напряжение. Линия TD в покое находится в состоянии логической единицы, так что на выходе вырабатывается *отрицательное* напряжение. Потенциалами линий можно управлять через регистры COM-порта (выход TD вырабатывает положительное напряжение, если установить бит BRCON). Питание можно получать и с сигнальных линий через выпрямительные диоды с использованием накопительных конденсаторов. При этом, конечно, следует учитывать, сколько времени выходной сигнал находится в нужном состоянии (чтобы накопленной энергии хватало).

Двуполярным питанием от линий интерфейса (+V от DTR и RTS, -V от TD) пользуются все манипуляторы, подключаемые к COM-портам. Зная это, в случае неработоспособности мыши с данным портом следует проверить напряжения на соответствующих контактах разъема. Бывает, что с конкретным портом не работает только конкретная мышь (модель или экземпляр), хотя другие мыши с этим портом и эти же мыши с другими портами работают нормально. Здесь дело может быть в уровнях напряжений. Стандарт требует от порта выходного напряжения не менее 5 В (абсолютного значения), и если данный порт обеспечивает только этот минимум, некоторым мышам не хватит мощности для питания светодиодов (главных потребителей энергии).

Порт получает двуполярное питание через системную плату от блока питания компьютера. Отсутствие на выходе блока питания напряжения +12 В обычно обнаружится по неработоспособности дисков. Отсутствие напряжения -12 В «заметят» только устройства, подключенные к COM-портам. Блок питания теоретически контролирует наличие этих напряжений на своем выходе (сообщая о неполадках сигналом Power Good, вызывающим аппаратный сброс). Встречаются упрощенные схемы блоков питания, у которых контролируются не все напряжения. Кроме того, возможны плохие контакты в разъеме подключения питания к системной плате.

## Глава 3

# Беспроводные интерфейсы

Беспроводные (wireless) интерфейсы позволяют освободить устройства от связывающих их интерфейсных кабелей, что особенно привлекательно для малогабаритной периферии, по размеру и весу соизмеримой с кабелями. В беспроводных интерфейсах используются электромагнитные волны инфракрасного (IrDA) и радиочастотного (Bluetooth) диапазонов. Кроме этих интерфейсов периферийных устройств существуют и беспроводные способы подключения к локальным сетям (см. [3]).

## 3.1. Инфракрасный интерфейс IrDA

Применение излучателей и приемников инфракрасного (ИК) диапазона позволяет осуществлять беспроводную связь между парой устройств, удаленных на расстояние до нескольких метров. Инфракрасная связь — *IR (Infra Red) Connection* — безопасна для здоровья, не создает помех в радиочастотном диапазоне и обеспечивает конфиденциальность передачи. ИК-лучи не проходят через стены, поэтому зона приема ограничивается небольшим, легко контролируемым пространством. Инфракрасная технология привлекательна для связи портативных компьютеров со стационарными компьютерами или док-станциями. Инфракрасный интерфейс имеют некоторые модели принтеров, им оснащают многие современные малогабаритные устройства: карманные компьютеры (PDA), мобильные телефоны, цифровые фотокамеры и т. п.

Различают инфракрасные системы низкой (до 115,2 Кбит/с), средней (1,152 Мбит/с) и высокой (4 Мбит/с) скорости. Низкоскоростные системы служат для обмена короткими сообщениями, высокоскоростные — для обмена файлами между компьютерами, подключения к компьютерной сети, вывода на принтер, проекционный аппарат и т. п. Ожидаются более высокие скорости обмена, которые позволят передавать «живое видео». В 1993 году была создана ассоциация разработчиков систем инфракрасной передачи данных *IrDA* (Infrared Data Association), призванная обеспечить совместимость оборудования от различных производителей. В настоящее время действует стандарт *IrDA 1.1*, наряду с которым существуют и собственные системы фирм Hewlett Packard — *HP-SIR* (Hewlett Packard Slow Infra Red)

и Sharp — *ASK IR* (Amplitude Shifted Keyed IR). Эти интерфейсы обеспечивают следующие скорости передачи:

- ◆ IrDA SIR (Serial Infra Red), HP-SIR -9,6-115,2 Кбит/с;
- ◆ IrDA HDLC, известный и как IrDA MIR (Middle Infra Red) - 0,576 и 1,152 Мбит/с;
- ◆ IrDA FIR (Fast Infra Red) - 4 Мбит/с;
- ◆ ASK IR - 9,6-57,6 Кбит/с.

Излучателем для ИК-связи является светодиод, имеющий пик спектральной характеристики мощности 880 нм; светодиод дает конус эффективного излучения с углом около 30°. В качестве приемника используют PIN-диоды, эффективно принимающие ИК-лучи в конусе 15°. Спецификация IrDA определяет требования к мощности передатчика и чувствительности приемника, причем для приемника задается как минимальная, так и максимальная мощность ИК-лучей. Импульсы слишком малой мощности приемник не «увидит», а слишком большая мощность «ослепляет» приемник — принимаемые импульсы сольются в неразличимый сигнал. Кроме полезного сигнала на приемник воздействуют помехи: засветка солнечным освещением и лампами накаливания, дающая постоянную составляющую оптической мощности, и помехи от люминесцентных ламп, дающие переменную (но низкочастотную) составляющую. Эти помехи приходится фильтровать. Спецификация IrDA обеспечивает уровень битовых ошибок (Bit Error Ratio, BER) не более 10<sup>-9</sup> при дальности до 1 м и дневном свете (освещенность до 10 кюк). Поскольку передатчик почти неизбежно вызывает засветку своего же приемника, вводя его в насыщение, приходится задействовать полудуплексную связь с определенными временными зазорами при смене направления обмена. Для передачи сигналов используют двоичную модуляцию (есть свет — нет света) и различные схемы кодирования.

Спецификация IrDA определяет многоуровневую систему протоколов, которую рассмотрим снизу вверх.

Ниже перечислены варианты, возможные на физическом уровне IrDA.

- ◆ *IrDA SIR* — для скоростей 2,4-115,2 Кбит/с используется стандартный асинхронный режим передачи (как в COM-портах): старт-бит (нулевой), 8 бит данных и стоп-бит (единичный). Нулевое значение бита кодируется импульсом длительностью 3/16 битового интервала (1,63 мкс на скорости 115,2 Кбит/с), единичное — отсутствием импульсов (режим IrDA SIR-A). Таким образом, в паузе между посылками передатчик не светит, а каждая посылка начинается с импульса старт-бита. В спецификации 1.1 предусмотрен и иной режим — IrDA SIR-B, с фиксированной длительностью импульса 1,63 мкс для всех этих скоростей.
- ◆ *ASK IR* — для скоростей 9,6-57,6 Кбит/с также используется асинхронный режим, но кодирование иное: нулевой бит кодируется посылкой импульсов с частотой 500 кГц, единичный — отсутствием импульсов.

- ◆ *IrDA HDLC* — для скоростей 0,576 и 1,152 Мбит/с используется синхронный режим передачи и кодирование, аналогичное протоколу SIR, но с длительностью импульса 1/4-битового интервала. Формат кадра соответствует протоколу HDLC, начало и конец кадра отмечаются флагами 01111110, внутри кадра эта битовая последовательность исключается путем вставки битов (bit stuffing). Для контроля достоверности кадр содержит 16-битный CRC-код.
- ◆ *IrDA FIR (IrDA4PPM)* — для скорости 4 Мбит/с также применяется синхронный режим, но кодирование несколько сложнее. Здесь каждая пара смежных битов кодируется позиционно-импульсным кодом: 00 → 1000, 01 → 0100, 10 → 0010, 11 → 0001 (в четверках символов «1» означает посылку импульса в соответствующей четверти двухбитового интервала). Такой способ кодирования позволил вдвое снизить частоту включения светодиода по сравнению с предыдущим. Постоянство средней частоты принимаемых импульсов облегчает адаптацию к уровню внешней засветки. Для повышения достоверности применяется 32-битный CRC-код.

Над физическим уровнем расположен *протокол доступа IrLAP (IrDA Infrared Link Access Protocol)* — модификация протокола HDLC, отражающая нужды ИК-связи. Этот протокол инкапсулирует данные в кадры и предотвращает конфликты устройств: при наличии более двух устройств, «видящих» друг друга, одно из них назначается первичным, а остальные — вторичными. Связь всегда полудуплексная. IrLAP описывает процедуру установления, нумерации и закрытия соединений. Соединение устанавливается на скорости 9600 бит/с, после чего согласуется скорость обмена по максимуму из доступных обоим (9,6,19,2,38,4,57,6 или 115,2 Кбит/с) и устанавливаются логические каналы (каждый канал управляется одним ведущим устройством).

Над IrLAP располагается *протокол управления соединением IrLMP (IrDA Infrared Link Management Protocol)*. С его помощью устройство сообщает остальным о своем присутствии в зоне охвата (конфигурация устройств IrDA может изменяться динамически: для ее изменения достаточно поднести новое устройство или отнести его подальше). Протокол IrLMP позволяет обнаруживать сервисы, предоставляемые устройством, проверять потоки данных и выступать в роли мультиплексора для конфигураций с множеством доступных устройств. Приложения с помощью IrLMP могут узнать, присутствует ли требуемое им устройства в зоне охвата. Однако гарантированной доставки данных этот протокол не обеспечивает.

*Транспортный уровень* обеспечивается протоколом *Tiny TP (IrDA Transport Protocols)* — здесь обслуживаются виртуальные каналы между устройствами, обрабатываются ошибки (потерянные пакеты, ошибки данных и т. п.), производится упаковка данных в пакеты и сборка исходных данных из пакетов (протокол напоминает TCP). На транспортном уровне может работать и протокол IrTP.

*Протокол IrCOMM* позволяет через ИК-связь эмулировать обычное проводное подключение:

- ◆ 3-проводное по RS-232C (TXD, RXD и GND);
- ◆ 9-проводное по RS-232C (весь набор сигналов COM-порта);
- ◆ Centronics (эмуляция параллельного интерфейса).

*Протокол IrLAN* обеспечивает доступ к локальным сетям, позволяя передавать кадры сетей Ethernet и Token Ring. Для ИК-подключения к локальной сети требуется устройство-провайдер с интерфейсом IrDA, подключенное обычным (проводным) способом к локальной сети, и соответствующая программная поддержка в клиентском устройстве (которое должно войти в сеть).

*Протокол объектного обмена IrOBEX* (Object Exchange Protocol) — простой протокол, определяющий команды PUT и GET для обмена «полезными» двоичными данными между устройствами. Этот протокол располагается над протоколом Tiny TP. У протокола IrOBEX есть расширение для мобильных коммуникаций, которое определяет передачу информации, относящуюся к сетям GSM (записная книжка, календарь, управление вызовом, цифровая передача голоса и т. п.), между телефоном и компьютерами разных размеров (от настольного до PDA).

Этими протоколами не исчерпывается весь список протоколов, имеющих отношение к ИК-связи. Заметим, что для дистанционного управления бытовой техникой (телевизоры, видеомэгафоны и т. п.) используется тот же диапазон 880 нм, но иные частоты и методы физического кодирования.

Приемопередатчик IrDA может быть подключен к компьютеру различными способами; по отношению к системному блоку он может быть как внутренним (размещаемым на лицевой панели), так и внешним, размещаемым в произвольном месте. Размещать приемопередатчик следует с учетом угла «зрения» (30° у передатчика и 15° у приемника) и расстояния до требуемого устройства (до 1 м).

*Внутренние приемопередатчики* на скоростях до 115,2 Кбит/с (IrDA SIR, HP-SIR, ASK IR) подключаются через обычные микросхемы UART, совместимые с 16450/ 16550 через сравнительно несложные схемы модуляторов-демодуляторов. В ряде современных системных плат на использование инфракрасной связи (до 115,2 Кбит/с) может конфигурироваться порт COM2. Для этого в дополнение к UART чипсет содержит схемы модулятора и демодулятора, обеспечивающие один или несколько протоколов инфракрасной связи. Чтобы порт COM2 использовать для инфракрасной связи, в CMOS Setup требуется выбрать соответствующий режим (запрет инфракрасной связи означает обычное использование COM2). Существуют внутренние адаптеры и в виде карт расширения (для шин ISA, PCI, PC Card); для системы они выглядят как дополнительные COM-порты.

На средних и высоких скоростях обмена применяются специализированные микросхемы контроллеров IrDA, ориентированные на интенсивный программно-управляемый обмен или DMA, с возможностью прямого управления шиной. Здесь обычный приемопередатчик UART непригоден, поскольку он не поддерживает синхронный режим и высокую скорость. Контроллер IrDA FIR выполняется в виде карты расширения или интегрируется в системную плату; как правило, такой контроллер поддерживает и режимы SIR.

Приемопередатчик подключается к разъему *IR-Connector* системной платы напрямую (если он устанавливается на лицевую панель компьютера) или через промежуточный разъем (mini-DIN), расположенный на скобе-заглушке задней стенки корпуса. К сожалению, единой раскладки цепей на внутреннем коннекторе нет,



и для большей гибкости приемопередатчик (или промежуточный разъем) снабжают кабелем с отдельными контактами разъема. Собрать их в должном порядке предоставляют пользователю; варианты назначения контактов коннектора инфракрасного приемопередатчика приведены в табл. 3.1. Некоторые приемопередатчики, поддерживающие режимы FIR и SIR, имеют отдельные выходы приемников — IRRX (для SIR) и FIRRX (для FIR). Если контроллер поддерживает только один из режимов, один из контактов останется неподключенным.

Таблица 3.1. Коннектор инфракрасного приемопередатчика

Цепь	Назначение	Контакт/вариант			
		12	3	4	
IRRX(RX)	Вход с приемника	1	3	3	3
FIRRX	Вход с приемника FIR	5	-	-	4
IRTX(TX)	Выход на передатчик	3	5	1	1
GND	Общий	2,7	4	2	2
Vcc (+5B)	Питание	4,6	1	5	5
NC	Свободный	—	2	4	

*Внешние ИК-адаптеры* выпускают с интерфейсом RS-232C для подключения к COM-порту или же с шиной USB. Пропускной способности USB достаточно даже для FIR, COM-порт пригоден только для SIR. Внешний ИК-адаптер IrDA SIR для COM-порта не так прост, как казалось бы: для работы модулятора-демодулятора требуется сигнал синхронизации с частотой, равной 16-кратной частоте передачи данных (этот сигнал поступает на синхровход микросхемы UART COM-порта). Такого сигнала на выходе COM-порта нет и его приходится восстанавливать из асинхронного битового потока. Адаптер ASK IR в этом плане проще — передатчик должен передавать высокочастотные импульсы все время, пока выход TXD находится в высоком состоянии; приемник должен формировать огибающую принятых импульсов.

Для прикладного использования IrDA кроме физического подключения адаптера и трансивера требуется установка и настройка соответствующих драйверов. В ОС Windows 9x/ME/2000 контроллер IrDA попадает в группу Сетевое окружение. Сконфигурированное ПО позволяет устанавливать соединение с локальной сетью (для выхода в Интернет, использования сетевых ресурсов); передавать файлы между парой компьютеров; выводить данные на печать; синхронизировать данные PDA, мобильного телефона и настольного компьютера; загружать отснятые изображения из фотокамеры в компьютер и выполнять ряд других полезных действий, не заботясь ни о каком кабельном хозяйстве.

## 3.2. Радиоинтерфейс Bluetooth

Bluetooth (синий зуб) — это фактический стандарт на миниатюрные недорогие средства передачи информации с помощью радиосвязи между мобильными (и настольными) компьютерами, мобильными телефонами и любыми другими

портативными устройствами на небольшие расстояния. Разработкой спецификации занимается группа лидирующих фирм в областях телекоммуникаций, компьютеров и сетей — 3Com, Agere Systems, Ericsson, IBM, Intel, Microsoft, Motorola, Nokia, Toshiba. Эта группа, образовавшая Bluetooth Special Interest Group, и вывела данную технологию на рынок. Спецификация Bluetooth свободно доступна в Сети ([www.bluetooth.com](http://www.bluetooth.com)), правда, она весьма объемна (около 15 Мбайт PDF-файлов). Открытость спецификации должна способствовать ее быстрому распространению, что уже и наблюдается на практике. Здесь позволим себе сократить название технологии до «BT» (это не официальное сокращение). Само название представляет собой прозвище датского короля, объединившего Данию и Норвегию, — намек на всеобщую объединяющую роль технологии.

Каждое устройство BT имеет радиопередатчик и приемник, работающие в диапазоне частот 2,4 ГГц. Этот диапазон в большинстве стран отведен для промышленной, научной и медицинской аппаратуры и не требует лицензирования, что обеспечивает повсеместную применимость устройств. Для BT используются радиоканалы с дискретной (двоичной) частотной модуляцией, несущая частота каналов  $F=2402+k$  (МГц), где  $k=0 \dots 78$ . Для нескольких стран (например, Франции, где в этом диапазоне работают военные) возможен сокращенный вариант с  $F=2454+k$  ( $k=0 \dots 22$ ). Кодирование простое — логической единице соответствует положительная девиация частоты, нулю — отрицательная. Передатчики могут быть трех классов мощности, с максимальной мощностью 1, 2,5 и 100 мВт, при чем должна быть возможность понижения мощности с целью экономии энергии. Передача ведется с перескоком несущей частоты с одного радиоканала на другой, что помогает в борьбе с интерференцией и замираниями сигнала. Физический канал связи представляется определенной псевдослучайной последовательностью используемых радиоканалов (79 или 23 возможных частот). Группа устройств, разделяющих один канал (то есть «знающих» одну и ту же последовательность перескоков), образует так называемую *пикосеть* (piconet), в которую может входить от 2 до 8 устройств. В каждой пикосети имеется одно ведущее устройство и до 7 активных ведомых. Кроме того, в зоне охвата ведущего устройства в его же пикосети могут находиться «припаркованные» ведомые устройства: они тоже «знают» последовательность перескоков и синхронизируются (по перескокам) с ведущим устройством, но не могут обмениваться данными до тех пор, пока ведущее устройство не разрешит их активность. Каждое активное ведомое устройство пикосети имеет свой временный номер (1-7); когда ведомое устройство деактивируется (паркуется), оно отдает свой номер для использования другими. При последующей активизации оно уже может получить иной номер (потому-то он и временный). Пикосети могут перекрываться зонами охвата, образуя «разбросанную» сеть (scatternet). При этом в каждой пикосети ведущее устройство только одно, но ведомые устройства могут входить в несколько пикосетей, используя разделение времени (часть времени он работает в одной, часть — в другой пикосети). Более того, ведущее устройство одной пикосети может быть ведомым устройством другой пикосети. Эти пикосети никак не синхронизированы, каждая из них использует свой канал (последовательность перескоков).

Канал делится на тайм-слоты длительностью 625 мкс, слоты последовательно нумеруются с цикличностью  $2^{27}$ . Каждый тайм-слот соответствует одной частоте, несущей в последовательности перескоков (1600 перескоков в секунду). Последовательность частот определяется адресом ведущего устройства пикосети. Передачи ведутся пакетами, каждый пакет может занимать от 1 до 5 тайм-слотов. Если пакет длинный, то он весь передается на одной частоте несущей, но отсчет слотов по 625 мкс продолжается, и после длинного пакета следующая частота будет соответствовать очередному номеру слота (то есть несколько перескоков будут пропущены). Ведущее и ведомые устройства ведут передачу поочередно: в четных слотах передачу ведет ведущее устройство, а в нечетных — адресованное им ведомое устройство (если ему есть что «сказать»).

Между ведущим и ведомыми устройствами могут устанавливаться физические связи двух типов: синхронные и асинхронные.

*Синхронные связи* (они же изохронные) с установлением соединения, SCO link (Synchronous Connection-Oriented), используются для передачи изохронного трафика (например, оцифрованного звука). Эти связи типа «точка—точка» предварительно устанавливает ведущее устройство с выбранными ведомыми устройствами, и для каждой связи определяется период (в слотах), через который для нее резервируются слоты. Связи получаются симметричные двусторонние. Повторные передачи пакетов в случае ошибок приема не используются. Ведущее устройство может установить до трех связей SCO с одним или разными ведомыми устройствами. Ведомое устройство может иметь до трех связей с одним ведущим устройством или иметь по одной связи SCO с двумя различными ведущими устройствами. По сетевой классификации связи SCO относятся к *коммутации цепей*.

*Асинхронные связи* без установления соединения, ACL link (Asynchronous Connection-Less), реализуют *коммутацию пакетов* по схеме «точка—множество точек» между ведущим устройством и всеми ведомыми устройствами пикосети. Ведущее устройство может связываться с любым из ведомых устройств пикосети в слотах, не занятых под SCO, посылая ему пакет и потребовав ответа. Ведомое устройство имеет право на передачу, только получив обращенный к нему запрос ведущего устройства (безошибочно декодировав свой адрес). Для большинства типов пакетов предусматривается повторная передача в случае обнаружения ошибки приема. Ведущее устройство может посылать и безадресные ширококвещательные пакеты для всех ведомых устройств своей пикосети. С каждым из своих ведомых устройств ведущее устройство может установить лишь одну связь ACL.

Информация передается пакетами, в которых поле данных может иметь длину 0-2745 бит. Для связей ACL предусмотрено несколько типов пакетов с защитой CRC-кодом (в случае обнаружения ошибки предусматривается повторная передача) и 1 беззащитный (без повторных передач). Для связей SCO данные не защищаются CRC-кодом, и следовательно, повторные передачи по ошибке приема не предусмотрены.

Защита данных от искажения и контроль достоверности производится несколькими способами. Данные некоторых типов пакетов защищаются CRC-кодом, и приемник информации должен подтверждать прием правильного пакета или сообщить об ошибке приема. Для сокращения числа повторов применяется избыточное

кодирование FEC (Forward Error Correction code). В схеме FEC 1/3 каждый полезный бит передается трижды, что позволяет выбрать наиболее правдоподобный вариант мажорированием. Схема FEC 2/3 несколько сложнее, здесь используется код Хэмминга, что позволяет исправлять все однократные и обнаруживать все двукратные ошибки в каждом 10-битном блоке.

Каждый *голосовой канал* обеспечивает скорость по 64 Кбит/с в обоих направлениях. В канале может использоваться кодирование в формате PCM (импульсно-кодовая модуляция) или CVSD (Continuous Variable Slope Delta Modulation — вариант адаптивной дельта импульсно-кодовой модуляции). Кодирование PCM допускает компрессию по G.711; оно обеспечивает лишь сугубо «телефонное» качество сигнала (имеется в виду цифровая телефония, 8-битные выборки с частотой 8 Кбит/с). Кодер CVSD обеспечивает более высокое качество — он упаковывает входной PCM-сигнал с частотой выборок 64 Кбит/с, однако и при этом спектральная плотность сигнала в полосе частот 4-32 кГц должна быть незначительной. Для передачи высококачественного аудиосигнала голосовые (речевые) каналы BT непригодны, однако сжатый сигнал (например, поток MP3) вполне можно передавать по асинхронному каналу передачи данных.

*Асинхронный канал* может обеспечивать максимальную скорость 723,2 Кбит/с в асимметричной конфигурации (оставляя для обратного канала полосу 57,6 Кбит/с) или же 433,9 Кбит/с в каждую сторону в симметричной конфигурации.

Для обеспечения безопасности в BT применяется *аутентификация* и *шифрование данных* на уровне связи (link layer), которые, конечно же, могут дополняться и средствами верхних протокольных уровней.

Важной частью BT является *протокол обнаружения сервисов SDP* (Service Discovery Protocol), позволяющий устройству найти «интересного собеседника». В дальнейшем, установив с ним соединение, устройство сможет воспользоваться требуемыми сервисами (например, выводить документы на печать, подключиться к Сети и т. п.).

*Протокол RFCOMM* обеспечивает эмуляцию последовательного порта (9-проводного RS-232) через L2CAP. С его помощью традиционные кабельные соединения устройств (в том числе и нуль-модемные) могут быть легко заменены на радиосвязь, без каких-либо модификаций ПО верхних уровней. Протокол позволяет устанавливать и множественные связи (одного устройства с несколькими), и радиосвязь заменит громоздкие и дорогие мультиплексоры и кабели. Через протокол RFCOMM может работать протокол OBEX, используемый в инфракрасных беспроводных соединениях (в иерархии протоколов IrDA), Через RFCOMM может работать и протокол PPP, над которым стоят протоколы стека TCP/IP, — это открывает дорогу во все приложения для Интернета. Через RFCOMM работают и AT-команды, управляющие телефонными соединениями и сервисами передачи факсов (эти же команды используются в модемах для коммутируемых линий). Специальный бит-ориентированный *телефонный протокол TCS BIN* (Telephony Control protocol — Binary), определяющий сигнализацию вызова для связи устройств BT (речевой связи и обмена данными), тоже работает через L2CAP. В протоколе имеются и средства управления группами устройств TCS.

*Интерфейс хост-контроллера HCI (Host Controller Interface)* — это единообразный метод доступа к аппаратно-программным средствам нижних уровней ВТ. Он предоставляет набор команд для управления радиосвязью, получения информации о состоянии и собственно передачи данных. Через этот интерфейс происходит взаимодействие протокола L2CAP с аппаратурой ВТ. Физически аппаратура ВТ может подключаться к различным интерфейсам: шине расширения (например, PC Card), шине USB, COM-порту. Для каждого из этих подключений имеется соответствующий протокол транспортного уровня HCI — прослойка, обеспечивающая независимость HCI от способа подключения.

## Глава 4

# Последовательные шины USB и Fire Wire

Последовательные шины позволяют объединять множество устройств, используя всего 1-2 пары проводов. Функциональные возможности этих шин гораздо шире, чем у традиционных интерфейсов локальных сетей, — USB и FireWire способны передавать изохронный трафик аудио- и видеоданных. Последовательные шины по своей организации сильно отличаются от параллельных. В последовательных шинах нет отдельных линий для данных, адреса и управления — все протокольные функции приходится выполнять, пользуясь одной или двумя (в FireWire) парами сигнальных проводов. Это накладывает отпечаток на построение шинного протокола, который в последовательных шинах строится на основе пересылок *пакетов* — определенным образом организованных цепочек бит. Заметим, что в терминологии USB пакеты и кадры имеют несколько иную трактовку, нежели в сетях передачи данных. В параллельных шинах имеются возможности явной синхронизации интерфейсной части ведущих и ведомых устройств; исполнение каждого шага протокола обмена может быть подтверждено, и, при необходимости, некоторые фазы обмена могут продлеваться по «просьбе» не успевающего устройства. В последовательных шинах такой возможности нет — пакет пересылается целиком, а синхронизация возможна только по принимаемому потоку бит. Эти и другие особенности сближают последовательные шины с локальными сетями передачи данных.

Наибольшую популярность имеют шины USB и FireWire, хотя последняя пока что в PC-совместимых компьютерах используется не повсеместно. Последовательные шины FireWire и USB, имея общие черты, являются, тем не менее, существенно различными технологиями. Обе шины обеспечивают простое подключение большого числа ПУ (127 для USB и 63 для FireWire), допуская коммутации и включение/выключение устройств при работающей системе. По структуре топология обеих шин достаточно близка, но FireWire допускает большую свободу и пространственную протяженность. Хабы USB входят в состав многих устройств и для пользователя их присутствие зачастую незаметно. Обе шины имеют линии питания устройств, но допустимая мощность для FireWire значительно выше. Обе шины поддерживают технологию PnP (автоматическое конфигурирование при включении/выключении) и снимают проблему дефицита адресов, каналов DMA и прерываний. Различаются пропускная способность и управление шинами.

*Шина USB* ориентирована на периферийные устройства, подключаемые к PC. Изохронные передачи USB позволяют передавать цифровые аудиосигналы, а шина USB 2.0 способна нести и видеоданные. Все передачи управляются централизованно, и PC является необходимым управляющим узлом, находящимся в корне древовидной структуры шины. Адаптер USB пользователи современных ПК получают почти бесплатно, поскольку он входит в состав всех современных чипсетов системных плат. Правда, адаптеры USB 2.0 первое время будут выпускаться в виде карт PCI. Непосредственное соединение нескольких PC шиной USB не предусматривается, хотя выпускаются «активные кабели» для связи пары компьютеров и устройства-концентраторы.

*Шина FireWire* ориентирована на устройства бытовой электроники, которые с ее помощью могут быть объединены в единую домашнюю сеть. К этой сети может быть подключен компьютер, и даже не один. Принципиальным преимуществом шины 1394 является отсутствие необходимости в специальном контроллере шины (компьютере). Любое передающее устройство может получить полосу изохронного трафика и начинать передачу по сигналу автономного или дистанционного управления — приемники «услышат» эту информацию. При наличии контроллера соответствующее ПО может управлять работой устройств, реализуя, например, цифровую студию нелинейного видеомонтажа или снабжая требуемыми мультимедийными данными всех заинтересованных потребителей информации.

## 4.1. Шина USB

USB (Universal Serial Bus — универсальная последовательная шина) является промышленным стандартом расширения архитектуры PC, ориентированным на интеграцию с телефонией и устройствами бытовой электроники. Версия 1.0 была опубликована в начале 1996 года, большинство устройств поддерживает версию 1.1, которая вышла осенью 1998 года, — в ней были устранены обнаруженные проблемы первой редакции. Весной 2000 года опубликована спецификация USB 2.0, в которой предусмотрено 40-кратное повышение пропускной способности шины. Первоначально (в версиях 1.0 и 1.1) шина обеспечивала две скорости передачи информации: *полная скорость FS* (full speed) — 12 Мбит/с и *низкая скорость LS* (Low Speed) — 1,5 Мбит/с. В версии 2.0 определена еще и *высокая скорость HS* (High Speed) — 480 Мбит/с, которая позволяет существенно расширить круг устройств, подключаемых к шине. В одной и той же системе могут присутствовать и одновременно работать устройства со всеми тремя скоростями. Шина с использованием промежуточных хабов позволяет соединять устройства, удаленные от компьютера на расстояние до 25 м. Подробную и оперативную информацию по USB (на английском языке) можно найти по адресу <http://www.usb.org>.

### 4.1.1. Организация шины USB

USB обеспечивает обмен данными между хост-компьютером и множеством периферийных устройств (ПУ). Согласно спецификации USB, устройства (devices) могут являться хабами, функциями или их комбинацией. Устройство-дгйб (hub)

только обеспечивает дополнительные точки подключения устройств к шине. Устройство-функция (function) USB предоставляет системе дополнительные функциональные возможности, например подключение к ISDN, цифровой джойстик, акустические колонки с цифровым интерфейсом и т. п. *Комбинированное устройство* (compound device), содержащее несколько функций, представляется как хаб с подключенными к нему несколькими устройствами. Устройство USB должно иметь интерфейс USB, обеспечивающий полную поддержку протокола USB, выполнение стандартных операций (конфигурирование и сброс) и предоставление информации, описывающей устройство. Работой всей системы USB управляет *хост-контроллер* (host controller), являющийся программно-аппаратной подсистемой *хост-компьютера*. Шина позволяет подключать, конфигурировать, использовать и отключать устройства во время работы хоста и самих устройств. Шина USB является хост-центрической: единственным ведущим устройством, которое управляет обменом, является хост-компьютер, а все присоединенные к ней периферийные устройства — исключительно ведомые. Физическая топология шины USB — многоярусная звезда. Ее вершиной является хост-контроллер, объединенный с корневым хабом (root hub), как правило, двухпортовым. Хаб является устройством-разветвителем, он может являться и источником питания для подключенных к нему устройств. К каждому порту хаба может непосредственно подключаться периферийное устройство или промежуточный хаб; шина допускает до 5 уровней каскадирования хабов (не считая корневого). Поскольку комбинированные устройства внутри себя содержат хаб, их подключения к хабу 6-го яруса уже недопустимо. Каждый промежуточный хаб имеет несколько *нисходящих* (downstream) портов для подключения периферийных устройств (или нижележащих хабов) и один *восходящий* (upstream) порт для подключения к корневному хабу или нисходящему порту вышестоящего хаба. Логическая топология USB — просто звезда: для хост-контроллера хабы создают иллюзию непосредственного подключения каждого устройства. В отличие от шин расширения (ISA, PCI, PC Card), где программа взаимодействует с устройствами посредством обращений по физическим адресам ячеек памяти, портов ввода-вывода, прерываниям и каналам DMA, взаимодействие приложений с устройствами USB выполняется только через программный интерфейс. Этот интерфейс, обеспечивающий независимость обращений к устройствам, предоставляется системным ПО контроллера USB.

В отличие от громоздких дорогих шлейфов параллельных шин AT A и особенно шины SCSI с ее разнообразием разъемов и сложностью правил подключения, кабельное хозяйство USB простое и изящное. Кабель USB содержит одну экранированную витую пару с импедансом 90 Ом для сигнальных цепей и одну неэкранированную для подачи питания (+5 В), допустимая длина сегмента — до 5 м. Для низкой скорости может использоваться невитой неэкранированный кабель длиной до 3 м (он дешевле). Система кабелей и коннекторов USB не дает возможности ошибиться при подключении устройств (рис. 4.1, а и б). Для распознавания разъема USB на корпусе устройства ставится стандартное символическое обозначение (рис. 4.1, в). Гнезда типа «А» устанавливаются только на нисходящих портах хабов, вилки типа «А» — на шнурах периферийных устройств или восходя-



щих портов хабов. Гнезда и вилки типа «B» используются только для шнуров, отсоединяемых от периферийных устройств и восходящих портов хабов (от «мелких» устройств — мышей, клавиатур и т. п. кабели, как правило, не отсоединяются). Кроме стандартных разъемов, показанных на рисунке 4.1, применяются и миниатюрные варианты (рис. 4.2, *в, з, д*). Хабы и устройства обеспечивают возможность «горячего» подключения и отключения. Для этого разъемы обеспечивают более раннее соединение и позднее отсоединение питающих цепей по отношению к сигнальным, кроме того, предусмотрен протокол сигнализации подключения и отключения устройств. Назначение выводов разъемов USB приведено в табл. 4.1, нумерация контактов показана на рис. 4.2. Все кабели USB «прямые» — в них соединяются одноименные цепи разъемов.

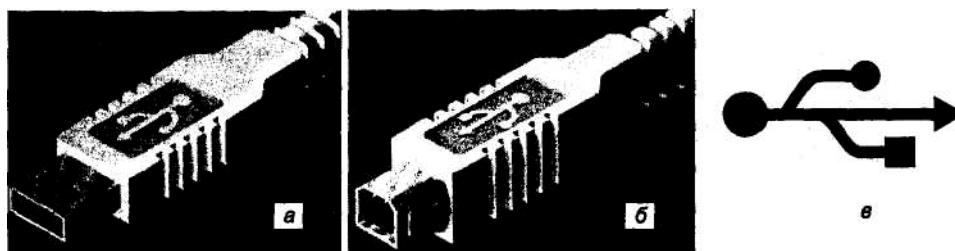


Рис. 4.1. Коннекторы USB: *а* — вилка типа «А», *б* — вилка типа «В», *в* — символическое обозначение

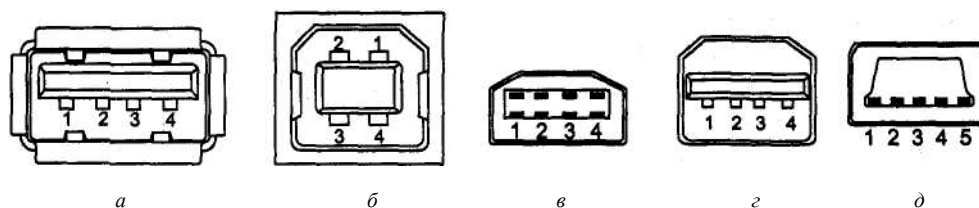


Рис. 4.2. Гнезда USB: *а* — типа «А», *б* — типа «В» стандартное, *в, г, д* — миниатюрные типа «В»

Таблица 4.1. Назначение выводов разъема USB

Контакт (рис. 4.2, <i>а–г</i> )	Контакт (рис. 4.2, <i>д</i> )	Цепь
1	1	VBus (+5 В)
2	2	D-
3	3	D+
4	5	GND

В шине используется дифференциальный способ передачи сигналов D+ и D- по двум проводам. Скорость устройства, подключенного к конкретному порту, определяется хабом по уровням сигналов на линиях D+ и D-, смещаемых нагрузочными резисторами приемопередатчиков: устройства с низкой скоростью «подтягивают» к высокому уровню линию D-, с полной — D+. Подключение устройства HS

определяется на этапе обмена конфигурационной информацией — физически на первое время устройство HS должно подключаться как FS. Передача по двум проводам в USB не ограничивается дифференциальными сигналами. Кроме дифференциального приемника, каждое устройство имеет линейные приемники сигналов D+ и D-, а передатчики этих линий управляются индивидуально. Это позволяет различать более двух состояний линии, используемых для организации аппаратного интерфейса.

Введение высокой скорости (480 Мбит/с — всего в 2 раза медленнее, чем Gigabit Ethernet) требует тщательного согласования приемопередатчиков и линии связи. На этой скорости может работать только кабель с экранированной витой парой для сигнальных линий. Для высокой скорости аппаратура USB должна иметь дополнительные специальные приемопередатчики. В отличие от формирователей потенциала для режимов FS и LS, передатчики HS являются источниками тока, ориентированными на наличие резисторов-терминаторов на обеих сигнальных линиях.

Скорость передачи данных (LS, FS или HS) выбирается разработчиком периферийного устройства в соответствии с потребностями этого устройства. Реализация низких скоростей для устройства обходится несколько дешевле (приемопередатчики проще, а кабель для LS может быть и неэкранированной невитой парой). Если в «старой» USB устройства можно было, не задумываясь, подключать в любой свободный порт любого хаба, то в USB 2.0 при наличии устройств и хабов разных версий появились возможности выбора между оптимальными, неоптимальными и неработоспособными конфигурациями.

Хабы USB 1.1 обязаны поддерживать скорости FS и LS, скорость подключенного к хабу устройства определяется автоматически по разности потенциалов сигнальных линий. Хабы USB 1.1 при передаче пакетов являются просто повторителями, обеспечивающими прозрачную связь периферийного устройства с контроллером. Передачи на низкой скорости довольно расточительно расходуют потенциальную пропускную способность шины: за то время, на которое они занимают шину, высокоскоростное устройство может передать данных в 8 раз больше. Но ради упрощения и удешевления всей системы на эти жертвы пошли, а за распределением полосы между разными устройствами следит планировщик транзакций хост-контроллера.

В спецификации 2.0 скорость 480 Мбит/с должна уживаться с прежними, но при таком соотношении скоростей обмена на FS и LS «съедят» возможную полосу пропускания шины без всякого «удовольствия» (для пользователя). Чтобы этого не происходило, хабы USB 2.0 приобретают черты коммутаторов пакетов. Если к порту такого хаба подключено высокоскоростное устройство (или аналогичный хаб), то хаб работает в режиме повторителя, и транзакция с устройством на HS занимает весь канал до хост-контроллера на все время своего выполнения. Если же к порту хаба USB 2.0 подключается устройство или хаб 1.1, то по части канала до контроллера пакет проходит на скорости HS, запоминается в буфере хаба, а к старому устройству или хабу идет уже на его «родной» скорости FS или LS. При этом функции контроллера и хаба 2.0 (включая и корневой) усложняются, поскольку транзакции на FS и LS расщепляются и между их частями вклиниваются

высокоскоростные передачи. От старых (1.1) устройств и хабов все эти тонкости скрываются, что и обеспечивает обратную совместимость. Вполне понятно, что устройство USB 2.0 сможет реализовать высокую скорость, только если по пути от него к хост-контроллеру (тоже 2.0) будут встречаться только хабы 2.0. Если это правило нарушить и между ним и контроллером 2.0 окажется старый хаб, то связь может быть установлена только в режиме FS. Если такая скорость устройство и клиентское ПО устроит (к примеру, для принтера и сканера это выльется только в большее время ожидания пользователя), то подключенное устройство работать будет, но появится сообщение о неоптимальной конфигурации соединений. По возможности ее (конфигурацию) следует исправить, благо переключения кабелей USB можно выполнять на ходу. Устройства и ПО, критичные к полосе пропускания шины, в неправильной конфигурации работать откажутся и категорично потребуют переключений. Если же хост-контроллер старый, то все преимущества USB 2.0 окажутся недоступными пользователю. В этом случае придется менять хост-контроллер (менять системную плату или приобретать PCI-карту контроллера). Контроллер и хабы USB 2.0 позволяют повысить суммарную пропускную способность шины и для старых устройств. Если устройства FS подключать к разным портам хабов USB 2.0 (включая и корневой), то для них суммарная пропускная способность шины USB возрастет по сравнению с 12 Мбит/с во столько раз, сколько используется портов высокоскоростных хабов.

Хаб является ключевым элементом системы PnP в архитектуре USB. Хаб выполняет множество функций:

- ◆ обеспечивает физическое подключение устройств, формируя и воспринимая сигналы в соответствии со спецификацией шины на каждом из своих портов;
- ◆ управляет подачей питающего напряжения на нисходящие порты, причем предусматривается установка ограничения на ток, потребляемый каждым портом;
- ◆ отслеживает состояние подключенных к нему устройств, уведомляя хост об изменениях;
- ◆ обнаруживает ошибки на шине, выполняет процедуры восстановления и изолирует неисправные сегменты шины;
- ◆ обеспечивает связь сегментов шины, работающих на разных скоростях.

Хаб следит за сигналами, генерируемыми устройствами. Неисправное устройство может не вовремя «замолчать» (потерять активность) или, наоборот, что-то «бормотать» (babble). Эти ситуации отслеживает ближайший к устройству хаб и запрещает восходящие передачи от такого устройства не позже, чем по границе (мик-ро)кадра. Благодаря бдительности хабов эти ситуации не позволяют неисправному устройству заблокировать всю шину.

Каждый из нисходящих (downstream) портов может быть разрешен или запрещен, а также сконфигурирован на высокую, полную или ограниченную скорость обмена. Хабы могут иметь *световые индикаторы* состояния нисходящих портов, управляемые автоматически (логикой хаба) или программно (хост-контроллером). Индикатор может представлять собой пару светодиодов — зеленый и желтый

(янтарный) или один светодиод с изменяющимся цветом. Состояние порта представляется следующим образом:

- ◆ не светится — порт не используется;
- ◆ зеленый — нормальная работа;
- ◆ желтый — ошибка;
- ◆ зеленый мигающий — программа требует внимания пользователя (Software attention);
- ◆ желтый мигающий — аппаратура требует внимания пользователя (Hardware attention).

Восходящий (upstream) порт хаба конфигурируется и внешне представляется как полноскоростной или высокоскоростной (только для USB 2.0). При подключении порт хаба USB 2.0 обеспечивает терминацию по схеме FS, в режим HS он переводится только по команде контроллера.

На рис. 4.3 приведен вариант соединения устройств и хабов, где высокоскоростным устройством USB 2.0 является только телекамера, передающая видеопоток без компрессии. Подключение принтера и сканера USB 1.1 к отдельным портам хаба 2.0, да еще и развязка их с аудиоустройствами, позволяет им использовать полосу шины по 12 Мбит/с каждому. Таким образом, из общей полосы 480 Мбит/с на «старые» устройства (USB 1.0) выделяется  $3 \times 12 = 36$  Мбит/с. Вообще-то можно говорить и о полосе в 48 Мбит/с, поскольку клавиатура и мышь подключены к отдельному порту хост-контроллера USB 2.0, но эти устройства «освоят» только малую толику из выделенных им 12 Мбит/с. Конечно, можно подключать клавиатуру и мышь к порту внешнего хаба, но с точки зрения повышения надежности системные устройства ввода лучше подключать наиболее коротким (по количеству кабелей, разъемов и промежуточных устройств) способом. Неудачной конфигурацией было бы подключение принтера (сканера) к хабу USB 1.1 — во время работы с аудиоустройствами (если они высокого качества) скорость печати (сканирования) будет падать. Неработоспособной конфигурацией явилось бы подключение телекамеры к порту хаба USB 1.1.

При планировании соединений следует учитывать способ питания устройств: устройства, питающиеся от шины, как правило, подключают к хабам, питающимся от сети. К хабам, питающимся от шины, подключают лишь маломощные устройства — так, к клавиатуре USB, содержащей внутри себя хаб, подключают мышь USB и другие устройства-указатели (трекбол, планшет).

*Управление энергопотреблением* является весьма развитой функцией USB. Для устройств, питающихся от шины, мощность ограничена. Любое устройство при подключении не должно потреблять от шины ток, превышающий 100 мА. Рабочий ток (не более 500 мА) заявляется в конфигурации. Если хаб не может обеспечить устройству заявленный ток, оно не конфигурируется и, следовательно, не может быть использовано.

Устройство USB должно поддерживать *режим приостановки* (suspended mode), в котором его потребляемый ток не превышает 500 мкА. Устройство должно автоматически приостанавливаться при прекращении активности шины.

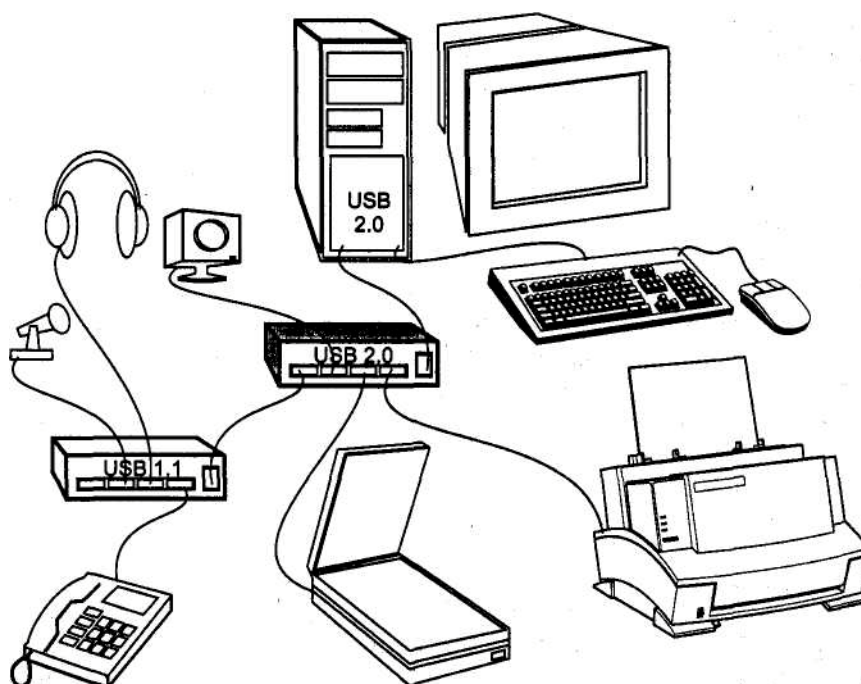


Рис. 4.3. Пример конфигурации соединений

Возможность *удаленного пробуждения* (remote wakeup) позволяет приостановленному устройству подать сигнал хост-компьютеру, который тоже может находиться в приостановленном состоянии. Возможность удаленного пробуждения описывается в конфигурации устройства. При конфигурировании эта функция может быть запрещена.

### 4.1.2. Модель передачи данных

Каждое устройство на шине USB (их может быть до 127) при подключении автоматически получает свой уникальный адрес. Логически устройство представляет собой набор независимых *конечных точек* (endpoint, EP), с которыми хост-контроллер (и клиентское ПО) обменивается информацией. Каждая конечная точка имеет свой номер и описывается следующими параметрами:

- ◆ требуемая частота доступа к шине и допустимые задержки обслуживания;
- ◆ требуемая полоса пропускания канала;
- ◆ требования к обработке ошибок;
- ◆ максимальные размеры передаваемых и принимаемых пакетов;
- ◆ тип передачи;
- ◆ направление передачи (для передач массивов и изохронного обмена).

Каждое устройство обязательно имеет конечную точку с номером 0, используемую для инициализации, общего управления и опроса состояния устройства. Эта точка всегда сконфигурирована при включении питания и подключении устройства к шине. Она поддерживает передачи типа «управление» (см. ниже). Кроме нулевой точки, устройства-функции могут иметь дополнительные точки, реализующие полезный обмен данными. Низкоскоростные устройства могут иметь до двух дополнительных точек, полноскоростные — до 15 точек ввода и 15 точек вывода (протокольное ограничение). Дополнительные точки (а именно они и предоставляют полезные для пользователя функции) не могут быть использованы до их конфигурирования (установления согласованного с ними канала).

*Каналом* (pipe) в USB называется модель передачи данных между хост-контроллером и конечной точкой устройства. Имеются два типа каналов: потоки и сообщения. *Поток* (stream) доставляет данные от одного конца канала к другому, он всегда однонаправленный. Один и тот же номер конечной точки может использоваться для двух поточных каналов — ввода и вывода. Поток может реализовывать следующие типы обмена: передача массивов, изхронный и прерывания. *Сообщение* (message) имеет формат, определенный спецификацией USB. Хост посылает запрос к конечной точке, после которого передается (принимается) пакет сообщения, за которым следует пакет с информацией состояния конечной точки. Последующее сообщение нормально не может быть послано до обработки предыдущего, но при обработке ошибок возможен сброс необслуженных сообщений. Двусторонний обмен сообщениями адресуется к одной и той же конечной точке.

С каналами связаны характеристики, соответствующие конечной точке (полоса пропускания, тип сервиса, размер буфера и т. п.). Каналы организуются при конфигурировании устройств USB. Для каждого включенного устройства существует канал сообщений (Control Pipe 0), по которому передается информация конфигурирования, управления и состояния.

### 4.1.3. Протокол

Все обмены (транзакции) с устройствами USB состоят из двух-трех пакетов. Каждая транзакция планируется и начинается по инициативе контроллера, который посылает *пакет-маркер* (token packet). Он описывает тип и направление передачи, адрес устройства USB и номер конечной точки. В каждой транзакции возможен обмен только между адресуемым устройством (его конечной точкой) и хостом. Адресуемое маркером устройство распознает свой адрес и готовится к обмену. Источник данных (определенный маркером) передает *пакет данных* (или уведомление об отсутствии данных, предназначенных для передачи). После успешного приема пакета приемник данных посылает *пакет квитирования* (handshake packet)? Последовательность пакетов в транзакциях иллюстрирует рис. 4.4. Хост-контроллер организует обмены с устройствами согласно своему плану распределения ресурсов. Контроллер циклически (с периодом  $1,0 \pm 0,0005$  мс) формирует *кадры* (frames), в которые укладываются все запланированные транзакции (рис. 4.5). Каждый кадр начинается с посылки маркера SOF (Start Of Frame),

который является синхронизирующим сигналом для всех устройств, включая хабы. В конце каждого кадра выделяется интервал времени EOF (End Of Frame), на время которого хабы запрещают передачу по направлению к контроллеру. В режиме HS пакеты SOF передаются в начале каждого *микрокадра* (период  $125 \pm 0,0625$  мкс). Хост планирует загрузку кадров так, чтобы в них всегда находилось место для транзакций управления и прерываний. Свободное время кадров может заполняться передачами массивов (bulk transfers). В каждом (микро)кадре может быть выполнено несколько транзакций, их допустимое число зависит от длины поля данных каждой из них.



Рис. 4.4. Последовательности пакетов: а — вывод, б — ввод

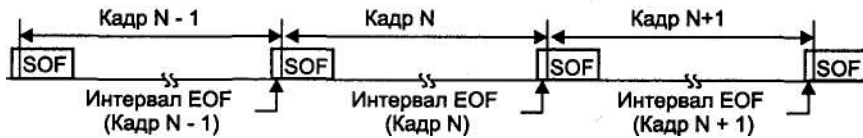


Рис. 4.5. Поток кадров USB

Для обнаружения ошибок передачи каждый пакет имеет контрольные поля CRC-кодов, позволяющие обнаруживать все одиночные и двойные битовые ошибки. Аппаратные средства обнаруживают ошибки передачи, а контроллер автоматически производит трехкратную попытку передачи. Если повторы безуспешны, сообщение об ошибке передается клиентскому ПО.

Все подробности организации транзакций от клиентского ПО изолируются контроллером USB и его системным программным обеспечением.

#### 4.1.4. Типы передач данных

Архитектура USB допускает четыре базовых типа передачи данных.

- ◆ *Управляющие посылки* (control transfers) используются для конфигурирования устройств во время их подключения и для управления устройствами в процессе работы. Протокол обеспечивает гарантированную доставку данных.
- ◆ *Передачи массивов данных* (bulk data transfers) — это передачи без каких-либо обязательств по задержке доставки и скорости передачи. Передачи массивов могут занимать всю полосу пропускания шины, свободную от передач других

типов. Приоритет этих передач самый низкий, они могут приостанавливаться при большой загрузке шины. Доставка гарантированная — при случайной ошибке выполняется повтор. Передачи массивов уместны для обмена данными с принтерами, сканерами, устройствами хранения и т. п.

- ◆ *Прерывания (interrupt)* — короткие передачи, которые имеют спонтанный характер и должны обслуживаться не медленнее, чем того требует устройство.

Предел времени обслуживания устанавливается в диапазоне 10-255 мс для низкой, 1-255 мс для полной скорости, на высокой скорости можно заказать и 125 мкс. При случайных ошибках обмена выполняется повтор. Прерывания используются, например, при вводе символов с клавиатуры или для передачи сообщения о перемещении мыши.

- ◆ *Изохронные передачи (isochronous transfers)* — непрерывные передачи в реальном времени, занимающие предварительно согласованную часть пропускной способности шины с гарантированным временем задержки доставки. Позволяют на полной скорости организовать канал с полосой 1,023 Мбайт/с (или два по 0,5 Мбайт/с), заняв 70 % доступной полосы (остаток можно заполнить и менее емкими каналами). На высокой скорости конечная точка может получить канал до 24 Мбайт/с (192 Мбит/с). В случае обнаружения ошибки изохронные данные не повторяются — недействительные пакеты игнорируются. Изохронные передачи нужны для потоковых устройств: видеокамер, цифровых аудиоустройств (колонки USB, микрофон), устройств воспроизведения и записи аудио- и видеоданных (CD и DVD). Видеопоток (без компрессии) шина USB способна передавать только на высокой скорости.

Полоса пропускания шины делится между всеми установленными каналами. Выделенная полоса закрепляется за каналом, и, если установление нового канала требует такой полосы, которая не вписывается в уже существующее распределение, запрос на выделение канала отвергается.

Архитектура USB предусматривает внутреннюю буферизацию всех устройств, причем чем большей полосы пропускания требует устройство, тем больше должен быть его буфер. Шина USB должна обеспечивать обмен с такой скоростью, чтобы задержка данных в устройстве, вызванная буферизацией, не превышала нескольких миллисекунд.

### 4.1.5. Синхронизация при изохронной передаче

Изохронная передача данных связана с синхронизацией устройств, объединяемых в единую систему. Возьмем пример использования USB, когда к компьютеру подключен микрофон USB (источник данных) и колонки USB (приемник данных), и эти аудиоустройства связаны между собой через программный микшер (клиентское ПО). Каждый из этих компонентов может иметь собственные «понятия» о времени и синхронизации: микрофон, к примеру, может иметь частоту выборки 8 кГц и разрядность данных 1 байт (поток 64 Кбит/с), стереоколонки — 44,1 кГц и разрядность 2x2 байта (176,4 Кбит/с), а микшер может работать на частоте выборки 32 кГц. Микшер в этой системе является связующим звеном, и его



источник синхронизации будем считать главным (master clock). Программный микшер обрабатывает данные пакетами, сеансы обработки выполняются регулярно с определенным периодом обслуживания (скажем, в 20 мс — частота 50 Гц). В микшере должны быть модули согласования частот выборки, которые объединяют несколько выборок в одну, если входная частота выше выходной, или «сочиняют» (интерполируют) новые промежуточные выборки, если выходная частота выше. В системе с USB приходится иметь дело со следующими частотами:

- ◆ частота выборки (sample rate) для источников (source) и приемников (sink) данных;
- ◆ частота шины USB — частота кадров (1 кГц) для полной скорости и микрокадров (8 кГц) для высокой (с этой частотой все устройства USB «видят» маркеры начала (микро)кадров SOF);
- ◆ частота обслуживания — частота, с которой клиентское ПО обращается к драйверам USB для передачи и приема изохронных данных.

В системе без общего источника синхронизации между парами синхросигналов возможны отклонения следующих типов:

- ◆ дрейф (drift) — отклонения формально одинаковых частот от номиналов (не бывает двух абсолютно одинаковых генераторов);
- ◆ дрожание (jitter) — колебание частот относительно номинала;
- ◆ фазовый сдвиг, если сигналы не связаны системой фазовой автоподстройки ФАПЧ (PLL).

В цифровой системе передачи данных эти отклонения выливаются в то, что у источника или приемника данных может образовываться излишек или недостаток данных, колеблющийся или прогрессирующий во времени. В USB по способу синхронизации конечных точек (источников или получателей данных) с системой различают асинхронный, синхронный и адаптивный классы устройств (точнее, конечных точек), каждому из которых соответствует свой тип канала USB.

*Асинхронные устройства* не имеют возможности согласования своей частоты выборок с метками SOF или иными частотами системы USB. Частота передачи данных у них фиксированная или программируемая. Число байт данных, принимаемых за каждый (микро)кадр USB, не является постоянным. Источник данных неявно сообщает свою скорость передачи данных числом выборок, генерируемых им за один (микро)кадр (клиентское ПО будет обрабатывать столько данных, сколько реально поступило). Приемник данных должен обеспечивать обратную связь для адаптивного драйвера клиентского ПО, чтобы согласовать темп выдачи потока (см. ниже). Примерами асинхронного устройства-источника может быть CD-плеер с синхронизацией от кварцевого генератора или приемник спутникового телевидения. Пример приемника — дешевые колонки, работающие от внутреннего источника синхронизации.

*Синхронные устройства* имеют внутренний генератор, синхронизируемый с метками SOF (системная частота 1 кГц); на высокой частоте передачи более точную

синхронизацию обеспечивает связь с микрокадрами. Источники и приемники за каждый (микро)кадр генерируют (потребляют) одинаковое количество байт данных, которое устанавливается на этапе программирования каналов. Примером синхронного источника может быть цифровой микрофон с частотой выборки, синтезируемой по SOF.

*Адаптивные устройства* имеют возможность подстройки своей внутренней частоты под требуемый поток данных (в определенных границах). Адаптивный источник позволяет менять скорость под управлением приемника, обеспечивающего обратную связь. Для адаптивного приемника информация о частоте задает входной поток данных. Он определяет мгновенное значение частоты по количеству данных, принятых за некоторый интервал усреднения. Примером адаптивного источника является CD-плеер со встроенным согласователем частоты SRC (sample rate converter) приемника — высококачественные колонки или наушники USB.

*Обратная связь* позволяет согласовать значения частот устройств с частотой шины. Асинхронный приемник должен явным образом сообщать хост-контроллеру желаемую частоту передачи данных относительно частоты (микро)кадров. Это позволит хост-контроллеру постоянно корректировать число передаваемых байт за каждый (микро)кадр, не допуская переполнения или опустошения буфера устройства-приемника. Адаптивный передатчик должен воспринимать информацию обратной связи, чтобы за каждый (микро)кадр генерировать ровно столько данных, сколько требуется хост-контроллеру. Для обратной связи в устройстве выделяется специальная конечная точка, через которую периодически передается информация о текущем значении желаемой относительной частоты.

В принципе контроллер USB может подстраивать частоту кадров, но, естественно, под частоту внутренней синхронизации только одного устройства. Подстройка осуществляется через механизм обратной связи, который позволяет изменять период кадра в пределах  $\pm 1$  битового интервала.

### 4.1.6. Хост

У каждой шины USB должен быть один (и только один!) хост — компьютер с контроллером USB. Хост делится на три основных уровня.

- ◆ *Интерфейс шины USB* обеспечивает физический интерфейс и протокол шины. Интерфейс шины реализуется хост-контроллером, имеющим встроенный корневой хаб, обеспечивающий точки физического подключения к шине (гнезда USB типа «А»). Хост-контроллер отвечает за генерацию (микро)кадров. На аппаратном уровне хост-контроллер обменивается информацией с основной памятью компьютера, используя прямое управление шиной (bus-mastering) с целью минимизации нагрузки на центральный процессор.
- ◆ *Система USB*, используя хост-контроллер(ы), транслирует клиентское «видение» обмена данными с устройствами в транзакции, выполняемые с реальными устройствами шины. Система отвечает и за распределение ресурсов USB —

полосы пропускания и мощности источников питания (для устройств, питающихся от шины). Система состоит из трех основных частей:

- *Драйвер хост-контроллера* — HCD (Host Controller Driver) — модуль, привязанный к конкретной модели контроллера, обеспечивающий абстрагирование драйвера USB и позволяющий в одну систему включать несколько разнотипных контроллеров.
- *Драйвер USB* — USB D (USB Driver) — обеспечивает основной интерфейс (USB D I) между клиентами и устройствами USB. Интерфейс HCD I (Host Controller Driver Interface) между USB D и HCD спецификацией USB не регламентируется. Он определяется разработчиками ОС и должен поддерживаться разработчиками хост-контроллеров, желающих иметь поддержку своих изделий конкретными ОС. Клиенты не могут пользоваться интерфейсом HCD I; для них предназначен интерфейс USB D I. USB D обеспечивает механизм обмена в виде пакетов IRP (I/O Request Packet — пакет запроса ввода-вывода), состоящих из запросов на транспортировку данных по заданному каналу. Кроме того, USB D отвечает за некоторое абстрактное представление устройства USB клиенту, которое позволяет выполнять конфигурирование и управление состоянием устройств (включая и стандартное управление через конечную точку «O»). Реализация интерфейса USB D I определяется операционной системой; в спецификации USB излагаются только общие идеи.
- *Программное обеспечение хоста* реализует функции, необходимые для функционирования системы USB в целом: обнаружение подключения и отключения устройств и выполнение соответствующих действий по этим событиям (загрузки требуемых драйверов), нумерацию устройств, распределение полосы пропускания и потребляемой мощности и т. п.
- ◆ *Клиенты USB* — программные элементы (приложения или системные компоненты), взаимодействующие с устройствами USB. Клиенты могут взаимодействовать с любыми устройствами (их конечными точками), подключенными к системе USB. Однако система USB изолирует клиентов от непосредственного обмена с какими-либо портами (в пространстве ввода-вывода) или ячейками памяти, представляющими интерфейсную часть контроллера USB.

В совокупности уровни хоста имеют следующие возможности:

- ◆ обнаружение подключения и отсоединения устройств USB;
- ◆ манипулирование потоками управления между устройствами и хостом;
- ◆ манипулирование потоками данных;
- ◆ сбор статистики активности и состояний устройств;
- ◆ управление электрическим интерфейсом между хост-контроллером и устройствами USB, включая управление электропитанием.

*Хост-контроллер* является аппаратным посредником между устройствами USB и хостом. Программная часть хоста в полном объеме реализуется операционной

#### 4.1. Шина USB

системой. До загрузки ОС может функционировать лишь усеченная часть ПО USB, поддерживающая только устройства, требующиеся для загрузки. Так, в BIOS современных системных плат имеется поддержка клавиатуры USB, реализующая функции сервиса Int 10h. При загрузке системы USB эта «дозагрузочная» поддержка игнорируется — система начинает работу с контроллером «с чистого листа», то есть со сброса и определения всех подключенных устройств. По окончании работы ОС передача состояния USB «дозагрузочной» поддержке не предусматривается, так что для нее это событие тоже может рассматриваться как первоначальное включение. В спецификации PC'2001 выдвигается требование к BIOS поддержки USB в такой мере, чтобы обеспечивалась загрузка ОС с устройств USB. USB поддерживает динамическое подключение и отключение устройств. Нумерация (перенумерация) устройств шины идет постоянно, отслеживая изменения физической топологии.

Все устройства подключаются через порты хабов. Хаб определяет подключение и отключение устройств к своим портам и сообщает состояние портов при запросе от контроллера. Хост разрешает работу порта и адресуется к устройству через канал управления, используя нулевой адрес — USB Default Address. При начальном подключении или после сброса все устройства адресуются именно так.

Хост определяет, является новое подключенное устройство хабом или функцией, и назначает ему уникальный адрес USB. Хост создает канал управления (control pipe) с этим устройством, используя назначенный адрес и нулевой номер точки назначения.

Если новое устройство является хабом, хост определяет подключенные к нему устройства, назначает им адреса и устанавливает каналы. Если новое устройство является функцией, уведомление о подключении передается диспетчером USB заинтересованному ПО.

Когда устройство отключается, хаб автоматически запрещает соответствующий порт и сообщает об отключении контроллеру, который удаляет сведения о данном устройстве из всех структур данных. Если отключается хаб, процесс удаления выполняется для всех подключенных к нему устройств. Если отключается функция, уведомление посылается заинтересованному ПО.

#### **4.1.7. Применение шины USB**

Благодаря своей универсальности и способности эффективно передавать разнородный трафик, шина USB применяется для подключения к PC самых разнообразных устройств. Она призвана заменить традиционные порты PC — COM и LPT, а также порты игрового адаптера и интерфейса MIDI. Спецификация USB 2.0 позволяет говорить и о подключении традиционных «клиентов» шин ATA и SCSI, а также захвате части ниши применения шины FireWire. Привлекательность USB придает возможность подключения/отключения устройств на ходу и возможность их использования практически сразу, без перезагрузки ОС. Удобна и возможность подключения большого количества (до 127) устройств к одной шине, правда, при наличии хабов. Хост-контроллер интегрирован в большинство современных сие-

темных плат. Выпускаются и карты расширения с контроллерами USB (обычно для шины PCI). Однако повсеместное применение USB сдерживается недостаточной активностью разработчиков ПО (производителей оборудования): просматривая перечни устройств, мы видим, что для всех указывается поддержка в Windows 98/SE/ME, а вот в графах Linux, MacOS, Unix и даже Windows 2000 часто стоят неприятные пометки N/A (Not Allowed — «не дозволено»). Для того чтобы система USB заработала, необходимо, чтобы были загружены драйверы хост-контроллера (или контроллеров, если их несколько). При подключении устройства к шине USB ОС Windows выдает сообщение «Обнаружено новое устройство» и, если устройство подключается впервые, предлагает загрузить для него драйверы. Многие модели устройств уже известны системе, и драйверы входят в дистрибутив ОС. Однако может потребоваться и драйвер изготовителя устройства, который должен входить в комплект поставки устройства, или его придется искать в Сети. К сожалению, не все драйверы работают корректно — «сырой» драйвер начальной версии, возможно, потребует замены более «правильным», чтобы устройство нормально опознавалось и хорошо работало. Но это общее горе пользователей любых устройств, а не только устройств для шины USB.

Перечислим основные области применения USB.

- ◆ *Устройства ввода* — клавиатуры, мыши, трекболы, планшетные указатели и т. п. Здесь USB предоставляет для различных устройств единый интерфейс. Целесообразность использования USB для клавиатуры неочевидна, хотя в паре с мышью USB (подключаемой к порту хаба, встроенного в клавиатуру) сокращается количество кабелей, тянущихся от системного блока на стол пользователя.
- ◆ *Принтеры*. USB 1.1 обеспечивает примерно ту же скорость, что и LPT-порт в режиме ECP, но при использовании USB не возникает проблем с длиной кабеля и подключением нескольких принтеров к одному компьютеру (правда, требуются хабы). USB 2.0 позволит ускорить печать в режиме высокого разрешения за счет сокращения времени на передачу больших массивов данных. Однако есть проблема со старым ПО, которое непосредственно работает с LPT-портом на уровне регистров, — на принтер USB оно печатать не сможет.
- ◆ *Сканеры*. Применение USB позволяет отказаться от контроллеров SCSI или от занятия LPT-порта. USB 2.0 при этом позволит еще и повысить скорость передачи данных.
- ◆ *Аудиоустройства* — колонки, микрофоны, головные телефоны (наушники). USB позволяет передавать потоки аудиоданных, достаточные для обеспечения самого высокого качества. Передача в цифровом виде от самого источника сигнала (микрофона со встроенным преобразователем и адаптером) до приемника и цифровая обработка в хост-компьютере позволяют избавиться от наводок, свойственных аналоговой передаче аудиосигналов. Использование этих аудио-компонентов позволяет в ряде случаев избавиться от звуковой карты компьютера — аудиокодек (АЦП и ЦАП) выводится за пределы компьютера, а все функции обработки сигналов (микшер, эквалайзер) реализуются центральным процессором чисто программно. Аудиоустройства могут и не иметь собственно колонок и микрофона, а ограничиться преобразователями и стандартными гнездами («Джеками») для подключения обычных аналоговых устройств.

- ◆ *Музыкальные синтезаторы и MIDI-контроллеры с интерфейсом USB.* Шина USB позволяет компьютеру обрабатывать потоки множества каналов MIDI (пропускная способность традиционного интерфейса MIDI уже гораздо ниже возможностей компьютера).
- ◆ *Видео- и фотокамеры.* USB 1.1 позволяет передавать статические изображения любого разрешения за приемлемое время, а также передавать поток видеоданных (живое видео) с достаточной частотой кадров (25-30 Кбит/с) только с невысоким разрешением или сжатием данных, от которого, естественно, страдает качество изображения. USB 2.0 позволяет передавать поток видеоданных высокого разрешения без сжатия (и потери качества). С интерфейсом USB выпускают как камеры, так и устройства захвата изображения с телевизионного сигнала и TV-тюнеры.
- ◆ *Коммуникации.* С интерфейсом USB выпускают разнообразные модемы, включая кабельные и xDSL, адаптеры высокоскоростной инфракрасной связи (IrDA FIR) — шина позволяет преодолеть предел скорости COM-порта (115,2 Кбит/с), не повышая загрузку центрального процессора. Выпускаются и сетевые адаптеры Ethernet, подключаемые к компьютеру по USB. Для соединения нескольких компьютеров в локальную сеть выпускаются специальные устройства, выполняющие коммутацию пакетов между компьютерами. Непосредственно (без дополнительных устройств) портами USB соединить между собой даже два компьютера нельзя — на одной шине может присутствовать лишь один хост-контроллер (см. выше). Специальное устройство для связи пары компьютеров выглядит как «таблетка», врезанная в кабель USB с двумя вилками типа «А» на концах. Объединение более двух компьютеров осложняется и топологическими ограничениями USB: длина одного сегмента кабеля не должна превышать 5 м, а использовать хабы для увеличения дальности неэффективно (каждый хаб дает всего 5 м дополнительного удаления).
- ◆ *Преобразователи интерфейсов* позволяют через порт USB, имеющийся теперь практически на всех компьютерах, подключать устройства с самыми разнообразными интерфейсами: Centronics и IEEE 1284 (LPT-порты), RS-232C (эмуляция UART 16550A — основы COM-портов) и другие последовательные интерфейсы (RS-422, RS-485, V.35...), эмуляторы портов клавиатуры и даже Game-порта, переходники на шину AT A, ISA, PC Card и любые другие, для которых достаточно производительности. Здесь USB становится палочкой-выручалочкой, когда встает проблема 2-го (3-го) LPT- или COM-порта в блокнотном ПК и в других ситуациях. При этом ПО преобразователя может обеспечить эмуляцию классического варианта «железа» стандартных портов IBM PC, но только под управлением ОС защищенного режима. Приложение MS-DOS может обращаться к устройствам по адресам ввода-вывода, памяти, прерываниями, каналами DMA, но только из сеанса MS-DOS, открытого в ОС с поддержкой USB (чаще это Windows). При загрузке «голой» MS-DOS «палочка-выручалочка» не работает. Преобразователи интерфейсов позволяют продлить жизнь устройствам с традиционными интерфейсами, изживаемыми из PC спецификациями PC'99 и PC'2001. Скорость передачи данных через конвертер USB — LPT может оказаться даже выше, чем у реального LPT-порта, работающего в режиме SPP.

- ◆ *Устройства хранения* — винчестеры, устройства чтения и записи CD и DVD, стриммеры — при использовании USB 1.1 получают скорость передачи, соизмеримую со скоростью их подключения к LPT, но более удобный интерфейс (как аппаратный, так и программный). При переходе на USB 2.0 скорость передачи данных становится соизмеримой с ATA и SCSI, а ограничений по количеству устройств достичь трудно. Особенно интересно использование USB для электронных устройств энергонезависимого хранения (на флэш-памяти) — такой накопитель может быть весьма компактным (размером с брелок для ключей) и емким (пока 16-256 Мбайт, в перспективах — гигабайт и более). Выпускаются устройства для мобильного подключения накопителей с интерфейсом ATA-ATAPI — по сути, это лишь преобразователи интерфейсов, помещенные в коробку-отсек формата 5" или 3,5", а иногда выполненные прямо в корпусе 36-контактного разъема ATA. Имеются и устройства чтения-записи карт SmartMedia Card и CompactFlash Card.
- ◆ *Игровые устройства* — джойстики всех видов (от «палочек» до автомобильных рулей), пульта с разнообразными датчиками (непрерывными и дискретными) и исполнительными механизмами (почему бы не сделать кресло автогонщика с вибраторами и качалками?) — подключаются унифицированным способом. При этом исключается ресурсопожирающий интерфейс старого игрового адаптера (упраздненного уже в спецификации PC'99).
- ◆ *Телефоны* — аналоговые и цифровые (ISDN). Подключение телефонного аппарата позволяет превратить компьютер в секретаря с функциями автодозвона, автоответчика, охраны и т. п.
- ◆ *Мониторы* — здесь шина USB используется для управления параметрами монитора. Монитор сообщает системе свой тип и возможности (параметры синхронизации) — это делалось и без USB по шине DDC. Однако USB-мониторы позволяют системе еще и управлять ими — регулировки яркости, контраста, цветовой температуры и т. п. могут теперь выполняться программно, а не только от кнопок лицевой панели монитора. В мониторы, как правило, встраивают хабы. Это удобно, поскольку настольную периферию не всегда удобно включать в «подстольный» системный блок.
- ◆ *Электронные ключи* — устройства с любым уровнем интеллектуальности защиты — могут быть выполнены в корпусе вилок USB. Они гораздо компактнее и мобильнее аналогичных устройств для COM- и LPT-портов.

Конечно же, перечисленными классами устройств сфера применения шины USB не ограничивается.

*Хабы USB* выпускаются как в виде отдельных устройств, так и встраиваются в периферийные устройства (клавиатуры, мониторы). Как правило, хабы питаются от сети переменного тока (они должны питать подключаемые устройства). Выпускают и хабы, устанавливаемые внутрь системного блока компьютера и питающиеся от его блока питания. Такие хабы дешевле внешних и не требуют дополнительной питающей розетки. Один из вариантов исполнения — установка хаба на скобку,

монтируемую в окно для дополнительных разъемов. Доступ к их разъемам со «спиной» системного блока не очень удобен для пользователей. Другой вариант — хаб, устанавливаемый в 3"-отсек. Его разъемы легкодоступны, индикаторы состояния портов хорошо видны, но не всегда удобны кабели, выходящие с передней панели системного блока. С другой стороны, для подключения электронных ключей (если их приходится часто менять) или миниатюрных накопителей этот вариант — самый удобный.

Недавно появились и новые вспомогательные устройства, увеличивающие дальность связи (distance extender). Это пара устройств, соединяемых между собой обычным кабелем «витая пара» (или оптоволокном), включаемая между периферийным устройством и хабом. «Удлинитель» со стороны периферии может иметь и хаб на несколько портов. К сожалению, увеличение дистанции упирается в ограничения на время задержки сигнала, свойственные протоколу шины USB, и достижимо лишь удаление до 100 м. Но даже и эта длина позволяет расширить сферу применения USB, например для удаленного видеонаблюдения.

### 4.1.8. Разработка собственных устройств USB

Несмотря на довольно сложный протокол обмена, интерфейсом USB можно снабдить и периферийные устройства собственной разработки. Для этого выпускается широкий ассортимент микросхем, со стороны USB различающихся скоростями обмена (LS, FS или HS), числом и возможностями конечных точек (тип передач, размер буфера). Функциональное назначение этих микросхем различно. С портом USB выпускаются микроконтроллеры на ядре MCS51, M68HC05, M68HC11 или RISC-архитектуры; они различаются объемом памяти (оперативной и энергонезависимой), производительностью, питанием, потреблением. Микроконтроллеры могут иметь встроенные устройства АЦП/ЦАП, дискретные линии ввода-вывода общего назначения, последовательные и параллельные порты различных типов. Их можно использовать для подключения устройств с любыми интерфейсами, сигнальных процессоров и т. п. Из этого ассортимента можно выбрать подходящую микросхему, на базе которой разрабатываемое устройство будет реализовано с минимальным числом дополнительных элементов. К микроконтроллерам прилагаются и средства разработки их встроенного ПО (firmware) — самой сложной части такого устройства. Есть микроконтроллеры с USB, способные работать без программирования энергонезависимой памяти; микроконтроллеры серии EzUSB фирмы Cypress Semiconductor каждый раз загружают свою программу в ОЗУ по шине USB из хост-компьютера в процессе подключения. Конечно, такая гибкость нужна не всегда, и до подключения к компьютеру устройство остается «мертвым».

Есть и периферийные микросхемы — порты USB, подключаемые к микроконтроллерам параллельной 8/16-битной шиной данных с обычным набором управляющих сигналов (CS#, RD#, WR#...), линией запроса прерывания и, возможно, сигналами канала DMA. Выпускаются и специализированные преобразователи интерфейсов USB в последовательный (RS-232, RS-422/485) и параллельный, не требующие



программирования (нужно лишь записать в EEPROM идентификатор устройства). Есть и микросхемы USB, сочетающие в себе и функции, и хабы. Все варианты не перечислить, тем более что все время появляются новые микросхемы. Информацию о них можно найти в Сети ([www.cypress.com](http://www.cypress.com), [www.devasys.com](http://www.devasys.com), [www.iged.com](http://www.iged.com), [www.microchip.com](http://www.microchip.com), [www.netchip.com](http://www.netchip.com), [www.motorola.com](http://www.motorola.com), [www.semiconductor.philips.com](http://www.semiconductor.philips.com), [www.natsemi.com](http://www.natsemi.com), [www.intel.com](http://www.intel.com), [www.ftdichip.com](http://www.ftdichip.com), [www.gigatechnology.com](http://www.gigatechnology.com)). Немаловажная часть разработки собственных устройств — программное обеспечение для хост-компьютера, которое доносит до пользователя всю пользу устройства. В ряде случаев удается воспользоваться готовыми драйверами (например, драйвером виртуального COM-порта для преобразователя интерфейса). В других случаях ПО приходится писать самостоятельно, и хорошо, когда изготовитель микросхем с USB заботится о предоставлении инструментальных средств разработки всех частей ПО.

## 4.2. Шина IEEE 1394 - FireWire

Стандарт для высокопроизводительной последовательной шины (High Performance Serial Bus), получивший официальное название IEEE 1394, был принят в 1995 году. Целью являлось создание шины, не уступающей параллельным шинам при существенном удешевлении и повышении удобства подключения (за счет перехода на последовательный интерфейс). Стандарт основан на шине FireWire, используемой Apple Computer в качестве дешевой альтернативы SCSI в компьютерах Macintosh и PowerMac. Название FireWire («огненный провод») теперь применяется и к реализациям IEEE 1394, оно сосуществует с кратким обозначением 1394. Другое название того же интерфейса — iLink, а иногда и Digital Link — используется фирмой Sony применительно к устройствам бытовой электроники. MultiMedia Connection — имя, используемое в логотипе 1394 High Performance Serial Bus Trade Association (1394TA).

Стандарт 1394 определяет три возможные частоты передачи сигналов по кабелям: 98,304, 196,608 и 393,216 Мбит/с, которые округляют до 100, 200 и 400 Мбит/с. Частоты в стандарте обозначаются как S100, S200 и S400 соответственно. В последней утвержденной ревизии стандарта, P1394-2000, новых скоростей (S800, S1600 и S3200) еще не появилось, и сейчас 1394 сосуществует с шиной USB, для которой в спецификации USB 2.0 уже определена скорость 480 Мбит/с.

Основные свойства шины FireWire перечислены ниже.

- ♦ *Многофункциональность.* Шина обеспечивает цифровую связь до 63 устройств без применения дополнительной аппаратуры (хабов). Устройства бытовой электроники — цифровые камкордеры (записывающие видеокамеры), камеры для видеоконференций, фотокамеры, приемники кабельного и спутникового телевидения, цифровые видеоплееры (CD и DVD), акустические системы, цифровые музыкальные инструменты, а также периферийные устройства компьютеров (принтеры, сканеры, устройства дисковой памяти) и сами компьютеры могут объединяться в единую сеть.

- ◆ *Высокая скорость обмена и изохронные передачи.* Шина позволяет даже на начальном уровне (S100) передавать одновременно два канала видео (30 кадров в секунду) широкоэмитательного качества и стерео-аудиосигнал с качеством CD.
- ◆ *Низкая цена компонентов и кабеля.*
- ◆ *Легкость установки и использования.* FireWire расширяет технологию PnP. Система допускает динамическое (горячее) подключение и отключение устройств. Устройства автоматически распознаются и конфигурируются при включении/отключении. Питание от шины (ток до 1,5 А) позволяет подключенным устройствам общаться с системой даже при отключении их питания. Управлять шиной и другими устройствами могут не только PC, но и другие «интеллектуальные» устройства бытовой электроники.

FireWire по инициативе VESA позиционируется как шина «домашней сети», объединяющей всю бытовую и компьютерную технику в единый комплекс. Эта сеть является одноранговой (peer-to-peer), чем существенно отличается от USB.

### 4.2.1. Физический уровень сети

Кабельная сеть 1394 собирается по простым правилам — все устройства соединяются друг с другом кабелями по любой топологии (древовидной, цепочечной, звездообразной). Каждое «полноразмерное» устройство (узел сети) обычно имеет три равноправных соединительных разъема. Некоторые малогабаритные устройства могут иметь только один разъем, что ограничивает возможные варианты их местоположения. Стандарт допускает и до 27 разъемов на одном устройстве, которое будет играть роль кабельного концентратора. Допускается множество вариантов подключения устройств, но со следующими ограничениями:

- ◆ между любой парой узлов может быть не более 16 кабельных сегментов;
- ◆ длина сегмента стандартного кабеля не должна превышать 4,5 м;
- ◆ суммарная длина кабеля не должна превышать 72 м (применение более качественного кабеля позволяет ослабить влияние этого ограничения);
- ◆ топология не должна иметь петель, хотя в последующих ревизиях предполагается автоматическое исключение петель в «патологических» конфигурациях.

Стандартный кабель 1394 содержит 6-проводов, заключенных в общий экран, и имеет однотипные 6-контактные разъемы на концах (рис. 4.6, а). Две витые пары используются для передачи сигналов (ТРА и ТРВ) отдельно для приемника и передатчика, два провода задействованы для питания устройств (8-40 В, до 1,5 А). В стандарте предусмотрена гальваническая развязка устройств, для чего используются трансформаторы (напряжение изоляции развязки до 500 В) или конденсаторы (в дешевых устройствах с напряжением развязки до 60 В относительно общего провода). Некоторые бытовые устройства имеют только один 4-контактный разъем меньшего размера (рис. 4.6, б), у которого реализованы только сигнальные цепи. Эти устройства подключаются к шине через специальный переходной кабель только как оконечные (хотя возможно применение специальных адаптеров-разветвителей). В кабелях FireWire сигнальные пары соединяются перекрестно (табл. 4.2), поскольку все порты равноправны.

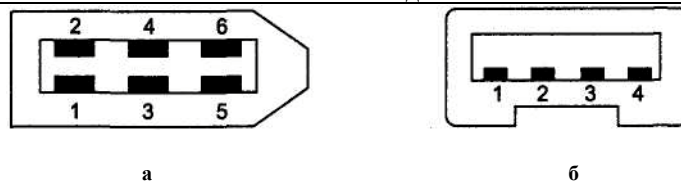


Рис. 4.6. Разъемы FireWire: а — 6-контактное гнездо, б — 4-контактное гнездо

Таблица 4.2. Соединительные кабели FireWire

Разъем А			Провод	Разъем Б		
4-конт.	6-конт.	Цепь		Цепь	6-конт.	4-конт.
-	1	Power	Белый	Power	1	
-	2	GND	Черный	GND	2	
1	3	УРЕ-	Красный	ТРА-	5	3
2	4	ТРЕ*	Зеленый	ТРА+	6	4
3	5	ТРА-	Оранжевый	ТРВ-	3	1
4	6	УРА*	Синий	ТРВ+	4	2
Экран	Экран	Экран	Экран	Экран	Экран	Экран

В грядущей версии, которая пока называется Р 1394Ь, предусматриваются и новые варианты среды передачи:

- ◆ кабель UTP категории 5 со стандартными коннекторами RJ-45 (используются две пары проводов), длина сегмента до 100 м — дешевый вариант для S100;
- ◆ пластиковое оптоволокно (два волокна POF для небольших расстояний и HPCF для больших дистанций) — дешевый вариант для S200;
- ◆ многомодовое оптоволокно (два волокна 50 мкм) — более дорогой вариант для будущих скоростей вплоть до S3200.

Каждое устройство, имеющее более одного разъема 1394, является повторителем. Сигнал, обнаруженный на входе приемника с любого разъема, ресинхронизируется по внутреннему тактовому генератору и выводится на передатчики всех остальных разъемов. Таким образом осуществляется доставка сигналов от каждого устройства ко всем остальным и предотвращается накопление «дрожания» (jitter) сигнала, ведущее к потере синхронизации.

Стандарт 1394 определяет две категории шин: кабельные шины и кросс-шины (Backplane). Под кросс-шинами подразумеваются обычно параллельные интерфейсы, объединяющие внутренние подсистемы устройства, подключенного к кабелю 1394. Сеть может состоять из множества шин, соединенных *мостами* — специальными устройствами, осуществляющими передачу пакетов между шинами, фильтрацию трафика, а для соединения разнородных шин еще и необходимые преобразования интерфейсов. Интерфейсная карта шины FireWire для PC представляет собой мост PCI — 1394. Мостами являются также соединения кабельной шины 1394 с кросс-шинами периферийных устройств. Мосты могут соединять и кабельные шины, что расширяет топологические возможности соединения устройств.

## 4.2.2. Протокол IEEE 1394

Протокол 1394 реализуется на трех уровнях (рис. 4.7).

- ◆ *Уровень транзакций* (Transaction Layer) преобразует пакеты в данные, предоставляемые приложениям, и наоборот. Он реализует протокол запросов-ответов, соответствующий стандарту ISO/IEC 13213:1994 (ANSI/IEEE 1212, редакции 1994 г.) архитектуры регистров управления и состояния CSR (Control and Status Register) для микрокомпьютерных шин (чтение, запись, блокировка). Это облегчает связь шины 1394 со стандартными параллельными шинами.
- ◆ *Уровень связи* (Link Layer) из данных физического уровня формирует пакеты и выполняет обратные преобразования. Он обеспечивает обмен узлами датаграммами с подтверждениями. Уровень отвечает за передачу пакетов и управление изохронными передачами.

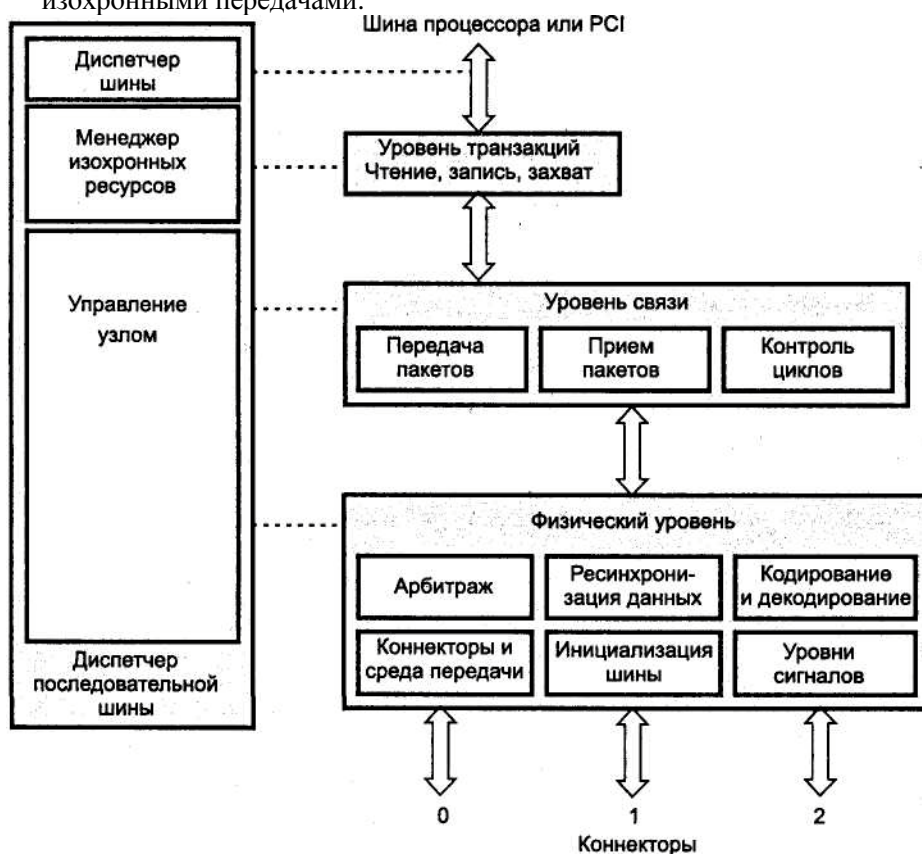


Рис. 4.7. Трехуровневая структура FireWire

- ◆ *Физический уровень* (Physical Layer) вырабатывает и принимает сигналы шины. Он обеспечивает инициализацию и арбитраж, предполагая, что в любой момент времени работает только один передатчик. Уровень передает потоки данных и уровни сигналов последовательной шины вышестоящему уровню. Между этими уровнями возможна гальваническая развязка, при которой микросхемы физического уровня питаются от шины. Гальваническая развязка необходима для предотвращения паразитных контуров общего провода, которые могут появиться через провода защитного заземления блоков питания.

Аппаратная часть FireWire обычно состоит из двух специализированных микросхем — трансиверов физического уровня PHY Transceiver и моста связи с шиной LINK Chip. Связь между ними возможна, например, по интерфейсу IBM-Apple LINK-PHY. Микросхемы уровня связи выполняют все функции своего уровня и часть функций уровня транзакций; остальная часть уровня транзакций выполняется программно.

Для передачи асинхронных сообщений используется 64-битная адресация регистров устройств 1394. В адресе выделяется 16 бит для адресации узлов сети: 6-битное поле идентификатора узла допускает до 63 устройств в каждой шине; 10-битное поле идентификатора шины допускает использование в системе до 1023 шин разного типа (включая внутренние), соединенных мостами. Протокол шины позволяет обращаться к памяти (регистрам) устройств в режиме DMA. В адресном пространстве каждого устройства имеются конфигурационные регистры, в которых содержится вся информация, необходимая для взаимодействия с ним других устройств. Данные передаются пакетами, в начале каждого пакета передаются биты состояния арбитража. Устройство может передавать данные только после успешного прохождения арбитража. Имеются два основных типа передач данных — изохронный, ради которого и строилась шина, и асинхронный. Изохронные передачи обеспечивают гарантированную полосу пропускания и время задержки, асинхронные передачи обеспечивают гарантированную доставку.

*Асинхронные сообщения* передаются между двумя устройствами. Инициатор посылает запрос требуемому устройству, на который оно сразу (через короткий интервал зазора, в котором шина находится в покое) отвечает подтверждением приема, положительным (ACK) или отрицательным (NACK), если обнаружена ошибка данных. Содержательный ответ на запрос (если требуется) будет передан обратно аналогичным способом (получатель должен послать подтверждение). Если подтверждение ACK не получено, передачи будут повторяться несколько раз до достижения успеха или фиксации ошибки.

*Изохронные передачи* ведутся широковещательно. В сети может быть организовано до 64 изохронных каналов, и каждый пакет изохронной передачи, кроме собственно данных, несет номер канала. Целостность данных контролируется CRC-кодом. Изохронные передачи всех каналов «слышат» все устройства шины, но из всех пакетов принимают только данные интересующих их каналов. Устройство-источник изохронных данных (камера, приемник, проигрыватель) на этапе конфигурирования получает номер и параметры выделенного ему канала.

#### 4.2. Шина IEEE 1394— FireWire<sub>III</sub>111

Шина поддерживает *динамическое реконфигурирование* — возможность «горячего» подключения и отключения устройств. Когда устройство включается в сеть, оно широковещательно передает короткий асинхронный пакет самоидентификации. Все уже подключенные устройства, приняв такой пакет, фиксируют появление новичка и выполняют процедуру сброса шины. По сбросу производится определение структуры шины, каждому узлу назначается физический адрес и производится арбитраж мастера циклов, диспетчера изохронных ресурсов и контроллера шины (см. ниже). Через секунду после сброса все ресурсы становятся доступными для последующего использования, и каждое устройство имеет полное представление обо всех подключенных устройствах и их возможностях. Отключение устройства от шины также обнаруживается всеми устройствами. Благодаря наличию линий питания интерфейсная часть устройства может оставаться подключенной к шине даже при отключении питания функциональной части устройства.

*Мастер циклов* — устройство, посылающее каждые 125 мкс короткие широковещательные пакеты начала циклов. В каждом таком пакете мастер циклов передает значение 32-битного счетчика времени, инкрементируемого с частотой 24,576 МГц, для каждого узла, поддерживающего изохронный обмен. В каждом цикле сначала передается по одному пакету каждого активного изохронного канала, затем на некоторое время зазора шина находится в состоянии покоя. После этого зазора начинается часть цикла, отводящаяся для передачи асинхронных пакетов. Каждое устройство, нуждающееся в асинхронной передаче, в этой части цикла может передать по одному пакету. Устройство, не имеющее пакета для передачи, шину и не занимает. После того как все нуждающиеся устройства передадут по одному пакету, в оставшееся время до конца цикла устройства могут передать и дополнительные пакеты.

*Диспетчер изохронных ресурсов* — устройство, ведающее распределением номеров каналов и полосы шины для изохронных передач. Диспетчер требуется, когда на шине появляется хоть одно устройство, способное к изохронной передаче. Диспетчер выбирается посредством арбитража из числа устройств, поддерживающих изохронный обмен. После сброса устройства, нуждающиеся в изохронной передаче, запрашивают требуемую полосу. Полоса измеряется в специальных единицах распределения, число которых в 125-микросекундном цикле составляет 6144. Единица занимает около 20 нс, что соответствует времени передачи одного квад-лета (quadlet, 32-битное слово) на частоте 1600 Мбит/с. Такой способ измерения полосы учитывает возможность совместной работы устройств с разными скоростями — в одном цикле соседние пакеты могут передаваться на разных скоростях. Как минимум 25 мкс цикла резервируется под асинхронный трафик, поэтому суммарная распределяемая полоса изохронного трафика составляет 4915 единиц. Для цифрового видео, например, требуется полоса 30 Мбит/с (25 Мбит/с на видеоданные и 3-4 Мбит/с на аудиоданные, синхронизацию и заголовки пакетов). В S100 устройства цифрового видео запрашивают около 1800 единиц, в S200 — около 900. Если требуемая полоса недоступна, диспетчер откажет устройству и не выделит ему номер канала. Устройство, не получившее канал, будет периодически повторять запрос. Когда изохронный обмен становится ненужным узлу, он должен освободить

свою полосу и номер канала, чтобы этими ресурсами смогли воспользоваться другие устройства. Обмен управляющей информацией устройств с диспетчером производится асинхронными сообщениями.

*Контроллер шины* (Bus Master) — необязательный элемент сети 1394, который осуществляет управление устройствами. Им может являться компьютер, редактирующее устройство цифровой записи или специальный интеллектуальный пульт управления. Контроллер шины, реализующий карты топологии и скоростей (TopologyMap и Speed\_Map), допускает использование нескольких частот в одной шине, в соответствии с возможностями конкретной пары устройств, участвующих в обмене. Иначе при подключении устройств, рассчитанных на разные скорости, все передачи будут происходить на скорости, доступной для всех активных устройств.

### 4.2.3. Устройства и адаптеры 1394

В PC-совместимом компьютере (в отличие от Macintosh) интерфейс 1394 пока не так распространен, как ставшая уже обязательной шина USB. Адаптеры FireWire чаще всего встречаются в виде карт расширения, но они уже встраиваются в некоторые модели системных плат. Адаптер 1394 для PC является мостом PCI — 1394, поскольку только шина PCI способна пропустить максимальный поток шины FireWire. Микросхемы для FireWire выпускает ряд фирм. Поначалу в основном использовались пары микросхем: LINK chip (микросхема уровня связи) и PHY chip (кристалл физического уровня). Это было связано со сложностью производства высокоскоростных микросхем физического уровня (на уровне связи S400 достигли быстро, а физический уровень на некоторое время «застрял» на S100 и S200). Модернизация такого адаптера сводилась лишь к последующей замене одного компонента. Сейчас применяют и однокристалльные решения. Например, микросхема VIA Fire II (VT6306) представляет собой трехпортовый адаптер S400 для шины PCI, поддерживающий и шину Card Bus (для мобильных компьютеров).

Интерфейс 1394 становится общепринятым для современной цифровой бытовой аудио-, видео- и фототехники, которые используют эту шину и без участия компьютера. Кроме цифровых устройств, имеющих встроенные адаптеры 1394, к шине FireWire возможно подключение и традиционных аналоговых и цифровых устройств (плееры, камеры, мониторы) через адаптеры-преобразователи интерфейсов и сигналов.

С интерфейсом 1394 выпускаются и устройства хранения данных — приводы CD и DVD, AV-диски (винчестеры, оптимизированные для записи и чтения мультимедийных данных). Выпускаются и преобразователи интерфейсов 1394-IDE, оформленные в виде корпусов для стандартных IDE-устройств форматов 5" или 3,5". В эти корпуса можно установить обычные винчестеры, приводы CD и DVD (включая и рекордеры), получая переносные устройства хранения данных. Для ОС и приложений устройства хранения выглядят как SCSI-устройства соответствующих классов. Это обеспечивается протоколом SBP-2 (Serial Bus Protocol), инкапсулирующим пакеты SCSI-3 в пакеты 1394.

#### 4.2.4. Использование 1394

Принципиальным преимуществом шины 1394 является отсутствие необходимости в контроллере. Любое передающее устройство может получить полосу изохронного трафика и начинать передачу по сигналу автономного или дистанционного управления — приемник «услышит» эту информацию. При наличии контроллера соответствующее ПО может управлять работой устройств, реализуя, например, цифровую студию нелинейного видеомонтажа или снабжая требуемыми мультимедийными данными всех заинтересованных потребителей информации. Для шины 1394 наиболее привлекательна возможность соединения устройств бытовой электроники (имеется в виду пока что не «наш», а «их» быт) в «домашнюю сеть», причем как с использованием PC, так и без. При этом стандартные однотипные кабели и разъемы 1394 заменяют множество разнородных соединений устройств бытовой электроники с PC. Разнотипные цифровые сигналы (сжатые видеосигналы, цифровые аудиосигналы, команды MIDI и управления устройствами, данные) мультиплексируются в одну шину, проходящую по всем помещениям. Используя одни и те же источники данных (приемники вещания, устройства хранения, видеокамеры и т. п.), можно одновременно в разных местах просматривать (прослушивать) разные программы с высоким качеством, обеспечиваемым цифровыми технологиями. Применение компьютера с адаптером 1394 и соответствующим ПО значительно расширяет возможности этой сети. Компьютер становится виртуальным коммутатором домашней аудио-видеостудии. Приложения для аудио- и видеоустройств используют логические «вилки» (plugs) и «розетки» (sockets), которые являются аналогами разъемов, применяемых в обычной аппаратуре. Вилки соответствуют выходам, розетки — входам соответствующих устройств. «Вставляя» эти «вилки» в «розетки» можно собрать требуемую систему. Конечно, для того чтобы она заработала, в устройствах должна быть реализована спецификация Digital Interface for Consumer Electronic Audio/Video Equipment — расширение стандарта IEEE-1394, предложенная DVC (Digital Video Consortium). Со временем она должна стать стандартом ISO/IEC.

Адаптер FireWire, например АНА-8940 фирмы Adaptec, может устанавливаться в любой PC (или Mac), имеющий свободный слот PCI. Для редактирования видео хватает мощности рядового современного ПК (минимальные требования — Pentium 133,32 Мбайт ОЗУ» 256 кбайт кэш, желательно быстрый SCSI-диск).

Поддержка 1394 имеется в ряде ОС, среди которых Windows 98, Windows 95 OSR 2.1 и более новые. Для редактирования аудио-видеофайлов (AVI) применимы, например, пакеты Adobe Premiere, Asymetrix Digital Video Producer, Ulead MediaStudio, MGI Video Wave. Кодек-конвертор цифровых видеоданных (DV), передаваемых по шине 1394, в AVI-файл поставляется фирмой Adaptec.

Одной из проблем цифровой передачи мультимедийной информации является защита авторских прав. Пользователь должен иметь возможность высококачественного воспроизведения принимаемых программ или приобретенных дисков, но их авторы (производители) должны иметь возможность защитить свои права, по своему усмотрению вводя ограничения на цифровое копирование. Для этих целей объединение «5С» (5 компаний: Sony, Matsushita, Intel, Hitachi и Toshiba) разрабатывает спецификацию шифрования данных.



# Глава 5

## Шина SCSI

Системный интерфейс малых компьютеров SCSI (Small Computer System Interface, произносится «скази») предназначен для соединения устройств различных классов: памяти прямого (жесткие диски) и последовательного доступа (стримеры), СО-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 И У выступает хост-адаптер. Каждое ЦУ может содержать до 8 независимо адресуемых логических устройств (ЛУ) со своими номерами LUN (Logical Unit Number), представляющих ПУ или их части.

Первая версия, шины, позже названная SCSI-1, была стандартизована ANSI в 1986 году (X3. 131-1986). Это была 8-битная параллельная шина с максимальной частотой переключений<sup>1</sup> 5 МТ/с, допускающая подключение до 8 устройств. Скорость передачи данных достигала 5 Мбайт/с, режим передачи данных — асинхронный. Впоследствии (1991 г) появилась спецификация SCSI-2 (X3.131-1994), расширяющая возможности шины. Частота переключений шины *Fast SCSI-2* достигает 10 МТ/с, а *Шина SCSI-2* — 20 МТ/с. Разрядность данных может быть увеличена до 16 бит — эта версия называется *Wide SCSI-2* (широкая), а 8-битную версию стали называть *Narrow* (узкая). 16-битная шина допускает включение 16 устройств. Стандарт SCSI-2 определяет и 32-битную версию интерфейса, которая не получила практического применения. Появился синхронный режим передачи данных, введена дифференциальная версия интерфейса. 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 разделение (совместное использование) пери-

---

<sup>1</sup> МТ/с - ^ миллионов передач в секунду (MT/s, Mega Transfer/sec). Называть это тактовой частотой некорректно, поскольку тактового сигнала в шине нет.

ферии несколькими компьютерами, подключенными к одной шине. Эти свойства обеспечивают SCSI неоспоримые преимущества перед ATA в качестве интерфейса для мощных систем хранения данных.

Спецификация SCSI-3 — дальнейшее развитие стандарта, направленное на увеличение количества подключаемых устройств, расширение системы команд и поддержку технологии PnP. 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. Спецификация SCSI-3 существует в виде широкого спектра документов, определяющих отдельные аспекты интерфейса на уровне физических соединений, транспортных протоколов и наборов команд. *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 (Fibre 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).

Ниже описана история развития параллельных интерфейсов в спецификации SCSI-3.

- ◆ В *стандарте SPI* (1995 г.) определен P-кабель и коннекторы для широкой шины на одном кабеле с 68-контактными разъемами, называемый «кабелем SCSI-3». SPI определяет скорость Fast SCSI (Fast Wide SCSI со скоростью 20 Мбайт/с). 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 переключений благодаря использованию интерфейса LVD. Интерфейс Fast-40 SCSI более известен как Ultra2 SCSI (Wide Ultra2 SCSI со скоростью передачи 80 Мбайт/с). Введен 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-кабель 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.) удвоена частота передачи, но уже за счет двойной синхронизации — интерфейс Fast-80DT (DT — Double Transition), известный как UltraS 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#/  
/

АСК#, 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 Auto-Matically — автоматическое конфигурирование устройств) и 32-битную шину.

- ♦ В стандарте SPI-4 (2001 г.) снова удваивается частота переключений и получается интерфейс Fast-160iDT, уже известный как Ultra320 SCSI (только широкая шина, 320 Мбайт/с).

## 5.1. Параллельные интерфейсы SCSI

Параллельный интерфейс SCSI существует в нескольких версиях, различающихся разрядностью шины, способами передачи сигналов и синхронизации. Физически «узкий» интерфейс 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 собственный 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.

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. В SCSI-1 применяются передатчики с открытым коллектором, приемники на биполярных транзисторах. Высокий уровень при пассивном состоянии передатчиков обеспечивают пассивные терминаторы (см. п. 5.1.2). В SCSI-2 стали применять и передатчики 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 выходная схема передатчика кратковременно форсирует переход линии к потенциалу пассивного состояния, после чего «отпускает» линию; при этом создается иллюзия возможности работы без терминаторов. В SCSI-3 стандарт SPI предписывает использование интерфейсных схем КМС-n(CMOS).

*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 скорость обмена. 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 не выдерживает.

*Низковольтный дифференциальный интерфейс LVD* позволяет работать на частотах 40, 80 и 160 МТ/с в устройствах Ultra2, UltraLGO 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.

Скорость передачи данных определяется *частотой переключений сигналов*, измеряемой в МТ/с, *разрядностью*, а в последних версиях и *способом синхронизации* (одиночная или двойная). Изначально разрядность шины SCSI составляла 8 бит (*Narrow*), а частота — до 5 МТ/с. Широкий (*wide*) вариант шины имеет разрядность 16 бит. Комбинации этих параметров обеспечивают широкий диапазон пропускной способности (табл. 5.1), 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 МТ/с, временные диаграммы для такого режима определены в SCSI-2. Краткое название *Fast-20* соответствует полному «Fast Wide SCSI» (16 бит, 10 МТ/с). Режим *Ultra SCSI* указывает на частоту передач 20 МТ/с, он определен для параллельного интерфейса в SCSI-3. Краткое название *Fast-40* соответствует полному «Wide Ultra SCSI» (16 бит, 20 МТ/с). Режим *Ultra2 SCSI* указывает на частоту передач 40 МТ/с, краткое название *Fast-80* соответствует полному «Wide Ultra2 SCSI» (16 бит, 40 МТ/с). 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 UltraS SCSI довольно широкое. Ultra160 SCSI означает скорость 160 Мбайт/с и существует 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 появился интерфейс Ultra320 SCSI со скоростью 320 Мбайт/с.

Таблица 5.1. Скорость передачи данных (Мбайт/с) по параллельной шине SCSI						
Разрядность шины, бит	Разновидность		Fast-20 (Ultra)	Fast-40 (Ultra2)	Ultra 160 (UltraS)	Ultra320
	Обычная	Fast				
8 (Narrow)	5	10	20	40	160	320
16 (Wide)	10	20	40	80		

Наиболее популярный интерфейс *Ultra2 SCSI* обеспечивает хорошее сочетание пропускной способности шины при ее большей длине, цены устройств и совместимости с традиционными устройствами SCSI.

### 5.1.1. Кабели, разъемы, сигналы

Для параллельного интерфейса характерен 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, причем только концевые; 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 часто 1,5 м. При расчете 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».

Таблица 5. 2\ Максимальная длина кабелей SCSI

Тип интерфейса	Обычный (5 МТ/с)	Fast (10 МТ/с)	Ultra (20 МТ/с)	Ultra 2 (40 МТ/с)	Ultra 160 (80 МТ/с)	UltraSZO (160 МТ/с)
Линейный (Single ended);	6м	3м	1,5 м (8устр.)	"_'"	.	—
> " ' .			3м '*"' (4устр.)			
Дифференциальный (HVD)	25 м	12м (16устр.)	6 м (16устр.)	. _ . . .		—
Дифференциальный низковольтный (LVD)	-	-	-	12м (16устр.)	12м (16устр.)	12м (16устр.)
				25 м (8устр.)	25 м (8устр.)	25 м (3устр.)

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 (о назначении контактов см; ниже). 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. Ниже описаны применяемые 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. 5.1, а). На устройствах устанавливаются *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. 5.1, б). Применяются для соединения внешних устройств. На корпусе устройства (и SCSI-адаптера) устанавливают *розетки* (CX-50F), на кабелях — *вилки* (CX-50M). Разъемы фиксируются проволоочными скобами, установленными на розетке, входящими в выемки на корпусе вилки. Часто называются «внешними SCSI-1» (SCSI-1 External).

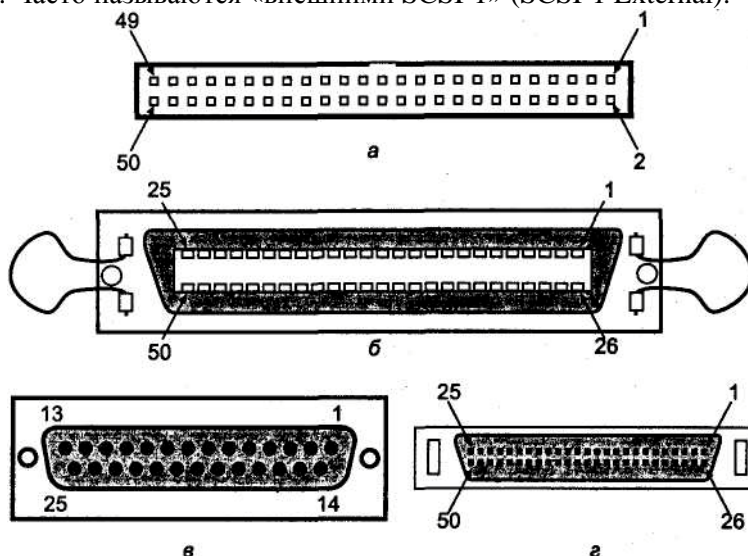


Рис. 5.1. Разъемы 8-битного устройства SCSI: а — IDC-50P, б — CX-50R в — DB-25P, г — HD-50F

- ◆ 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 выполняется с помощью винтов (рис. 5.1, в). Применяются на некоторых внешних устройствах (например, Zip).
- ◆ HD-50, они же MiniDSO (рис. 5.1, г), — разъемы со штырьковыми контактами в металлическом кожухе D-образной формы, с высокой плотностью контактов (High Density) — с шагом 0,05" (1,27 мм). На устройстве устанавливается *розетка* (HD-50F), на кабеле — *вилка* (HD-50M); фиксация выполняется с помощью защелок (клипсов). 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 клипсов или винтов, внутренние — только на трении. Часто называются разъемами SCSI-3, в настоящее время наиболее широко используются для «широкого» интерфейса. На рис. 5.2 показан внешний разъем, слева изображена клипса, справа — резьбовая буска.

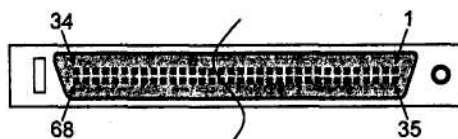


Рис. 5.2. Разъемы 16-битного устройства SCSI HD-68F

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. 5.3). На устройстве устанавливается вгря (MCX-80F), на шасси — розетка (MCX-80M). Кроме интерфейсных сигналов, на разъем выводятся шины питания, ^ 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 на плате адаптера).

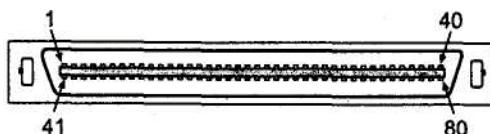


Рис. 5.3. Разъем устройства SCSI с «горячей» заменой SCA-80

Для версии Narrow SCSI использовались разъемы, изображенные на рис. 5.1, для Wide SCSI — на рис. 5.2. Для устройств с «горячей» заменой применяют миниатюрный D-образный разъем SCA-2, общий для питания и сигнальных цепей (рис. 5.3). 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. 5.1, а);
  - внешний А-кабель SCSI-1 имеет разъемы Centronics-50 (CX-50M, см. рис. 5.1, б);
  - внешний А-кабель SCSI-2 имеет разъемы MiniDSOM (HD-50M, см. рис. 5.1, в).
- ◆ *В-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 (см. рис. 5.1, з) — 8-битный, стандартный для Macintosh (назначение контактов см. ниже), 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. 5.1.5). 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. 5.3-5.7. 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 5.3. Разъемы A-кабеля SCSI

Контакт IDC-50 / CX-50	Цепь 3E/Diff	Контакт IDC-50 / CX-50	Цепь
1/1	GND/DBO+	2/26	DBO#
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/DBPO+	18/34	DBPO#
19/10	GND/GND	20/35	GND
21/11	GND/GND	22/36	GND

продолжение



Таблица 5.3 (продолжение)

Контакт IOC-50/ CX-50	УенbSE/Diff	Контакт IDC-50/CX-50	Цепь
23/12	Резерв	24/37	Резерв
25/13	Свободен	26/38	TERMPW
27/14	Резерв	28/39	R
29/15	GND	30/40	Резерв
31/16	GND/ATN+	32/41	GND
33/17	GND	34/42	ATN#
35/18	GND/BSY+	36/43	GND
37/19	GND/ACK+	38/44	BSY#
39/20	GND/RST+	40/45	ACK#
41/21	GND/MSG+	42/46	RST#
43/22	GND/SEL+	44/47	MSG#
45/23	GND/C/D+	46/48	SEL#
47/24	GND/REQ+	48/49	C/D#
49/25	GND/I/O+	50/50	REQ#
			I/O#

Таблица 5.4. Разъемы В-кабеля SCSI

Контакт внутр. / внешн.	УенbSE/Diff	Контакт внутр./внешн.	Цепь
<b>1/1</b>	GND	2/35	GND
<b>3/2</b>	GND/DB8+	4/36	DB8#
<b>5/3</b>	GND/DB9+	6/37	DB9#
<b>7/4</b>	GND/DB10+	8/38	DB10#
<b>9/5</b>	GND/DB11 +	10/39	DB11#
<b>11/6</b>	GND/DB12+	12/40	DB12#
<b>13/7</b>	GND/DB13+	14/41.	DB13#
<b>15/8</b>	GND/DB14+	16/42	DB14#
<b>17/9</b>	GND/DB15+	18/43	DB15#
<b>19/10</b>	GND/DBP1 +	20/44	DBP1#
<b>21/11</b>	GND/ACKB+	22/45	ACKB#
<b>23/12</b>	GND/GND	24/46	GND
<b>25/13</b>	GND/REQB+	26/47	REQB#
<b>27/14</b>	GND/DB16+	28/48	DB16#
<b>29/15</b>	GND/DB17+	30/49	DB17#
<b>31/16</b>	GND/DB18+	32/50	DB18#
<b>33/17</b>	TERMPWR	34/51	TERMPW
<b>35/18</b>	TERMPWR	36/52	R
<b>37/19</b>	GND/DB19+	38/53	TERMPW
<b>39/20</b>	GND/DB20+	40/54	R
<b>41/21</b>	GND/DB21 +	42/55	DB19#
<b>43/22</b>	GND/DB22+	44/56	DB20#
<b>45/23</b>	GND/DB23+	46/57	DB21#
			DB22#
			DB23#

## 5.1. Параллельные интерфейсы SCSI

Контакт внутр./внешн.	УенbSE/Diff	Контакт внутр./внешн.	Цепь
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 5.5. 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/DBO+	40	DBO#
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/DBPO+	48	DBPO#
15	GND	49	GND
16	DIFFSENS (GND)	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#
25	GND/RST+	59	RST#
26	GND/MSG+	60	MSG#
27	GND/SEL+	61	SEL#

продолжение^

Таблица 5 .5 (продолжение)

Контакт	UenbSE/Diff	Контакт	Цепь
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

Сигнал DIFFSENS определен только для интерфейса LVD.

Таблица 6. Разъем Mac SCSI (DB-25)

Контакт	Цепь	Контакт	Цепь
1	REQ#	14	GND
2	MSG#	15	C/D#
3	I/O*	16	GND
4	RST#	17	ATN#
5	ACK#	18	GND
6	BSY#	19	SEL#
7	GND	20	DBPO#
8	DBO#	21	DB1#
9	GND	22	DB2#
10	DB3#	23	• DB4#
11	DB5#	24	GND
12	DB6#	25	TERMPWR
13	DB7#		

Таблица 5.7. Разъем SCA-80

Контакт	Цепь	Контакт	4enbSE/Diff
01	12 Volt Charge	41	12V GND
02	12 Volt	42	12V GND
03	12 Volt	43	12V GND
04	12 Volt	44	Mated 1
05	Резерв	45	Резерв
06	Резерв	46	GND
07	DB11#	47	GND/DB11 +
08	DB10#	48	GND/DB10+
09	DB9#	49	GND/DB9+
10	DB8#	50	GND/DB8+
11	I/O*	51	GND/I/O+
12	REQ#	52	GND/REQ+

Контакт	Цепь	Контакт	UenbSE/Diff
13	C/D#	53	GND/C/D+
14	SEL#	54	GND/SEL+
15	MSG#	55	GND/MSG+
16	RST#	56	GND/RST+
17	ACK#	57	GND/ACK+
18	BSY#	58	GND/BSY+
19	ATN#	59	GND/ATN+
20	DBPO#	60	GND/DBPO+
21	DB7#	61	GND/DB7+
22	DB6#	62	GND/DB6+
23	DB5#	63	GND/DB5+
24	DB4#	64	GND/DB4+
25	DB3#	65	GND/DB3+
26	DB2#	66	GND/DB2+
27	DB1#	67	GND/DB1+
28	DBO#	68	GND/DBO+
29	DBP1#	69	GND/DBP1 +
30	DB15#	70	GND/DB15+
31	DB14#	71	GND/DB14+
32	DB13#	72	GND/DB13+
33	DB12#	73	GND/DB12+
34	5 Volt	74	Mated 2
35	5 Volt	75	5VGND
36	5 Volt Charge	76	5VGND
37	Резерв	77	Active LED
38	Auto Spin Up	78	Delayed Start
39	SCSI I D 0	79	SCSI I D 1
40	SCSI ID 2	80	SCSI I D 3

## 5.1.2. Терминаторы

Как было сказано выше, каждая физическая шина SCSI должна оканчиваться терминаторами, устанавливаемыми на обоих ее концах. Терминаторы могут быть как внутренними (установленными внутри контроллеров и периферийных устройств SCSI), так и внешними — маленькими блоками, устанавливаемыми на разъем кабеля или дополнительный разъем последнего устройства. Терминаторы шины SCSI должны выполнить две задачи:

- ◆ избавить линии шины от отражений сигналов с ее концов;
- ◆ обеспечить требуемый уровень сигнала пассивных линий.

Первая задача вытекает из того, что шлейф SCSI может иметь довольно большую протяженность, и в терминах теоретических основ электротехники каждая сиг-

нальная линия является «длинной линией». Чтобы сигналы не отражались от концов этой линии, оба конца должны быть нагружены согласованной нагрузкой. Согласованность означает совпадение волнового сопротивления линии с динамическим сопротивлением (импедансом) нагрузки. Волновое сопротивление линий кабельных шлейфов, применяемых в SCSI, обычно лежит в диапазоне 85-110 Ом. Если терминаторов не будет (или импеданс терминатора не совпадает с линией), «звон» отраженных сигналов будет приводить к помехам на шине. Вторая задача обусловлена спецификой интерфейса SCSI, где каждой сигнальной линией может управлять любое из нескольких устройств, подключенных к шине. Причем устройство, посылающее сигнал, формирует только активный уровень (низкий в недифференциальных версиях SCSI), а вернуть линию в пассивное состояние должны терминаторы. Если нет ни одного терминатора, то уровень на линиях, «отпущенных» устройством, за счет входных токов приемников сигнала тоже будет возвращаться в пассивное состояние, но гораздо медленнее. Если шина достаточно длинная и устройств много, то это возвращение будет слишком запоздалым и может вызвать сбой в протоколе. У высокоскоростных устройств применяется активное возвращение сигналов в пассивное состояние (active negation), от чего возникает иллюзия возможности работы шины без терминаторов. Однако устойчивая работа при большом количестве устройств (более двух на шине, включая контроллер) проблематична.

Из сказанного выше становится понятно, когда пренебрежение правилами установки терминаторов может «сойти с рук»: когда шина не очень длинная, устройств мало (скажем, контроллер и один винчестер), а скорость обмена невелика. Однако на одну шину SCSI чаще устанавливаются много устройств, при этом шина получается довольно длинной; устройства стараются использовать на высоких скоростях обмена, да еще и с высокими требованиями к надежности. В иных случаях было бы выгоднее применять устройства с иным интерфейсом — для устройств хранения данных это широко распространенный и дешевый интерфейс ATA. Рассмотрим, какие бывают терминаторы для наиболее популярных устройств SE и LVD. Многие устройства с интерфейсом LVD способны работать и с интерфейсом SE (но на малых скоростях); их интерфейс обозначается символами «LVD/SE». Режим работы эти устройства способны определять автоматически: если все устройства на шине (включая и терминаторы) «умеют» работать в режиме LVD, то этот режим и будет выбран (если ни на одном из устройств принудительно не установлен режим SE). Если же на шине хоть одно устройство способно только на режим SE, то в этот режим перейдут и все остальные (соответственно, снижая возможную скорость передачи данных). Заметим, что устройства HVD (Diff) в компанию к LVD/SE включать нельзя.

Для режимов SE и LVD различается способ передачи по сигнальным линиям и режим терминирования. Каждая сигнальная линия шины SCSI состоит из пары проводов: прямого и обратного. В режиме SE все обратные провода соединяются с «землей» (на каждом устройстве); терминирующие цепи подключаются только к прямым проводам. В режиме LVD по каждой паре проводов сигнал передается в дифференциальной парафазии форме; терминирующие цепи подключаются к обоим

проводам каждой пары. Варианты схем терминаторов для SE и LVD приведены на рис. 5.4, где изображены нагрузочные цепи для одной: сигнальной линии. Все терминаторы (не только «активные»!) нуждаются в питании; которое на них поступает по специальным линиям TERMPWR (+5В).

- ◆ *Пассивные терминаторы SE* (рис. 5.4, а) имеют импеданс 132 Ом, что плохо согласуется с ленточным кабелем шины. Эти терминаторы пригодны лишь для «обычного» интерфейса SCSI (скорость передачи до 5/10 Мбайт/с в «узком»/«широком» вариантах). Для Fast SCSI, Ultra SCSI и далее они непригодны.
- ◆ *Активные терминаторы SE* (рис. 5.4, б) имеют импеданс 110 Ом, что позволяет их использовать на более высоких скоростях в Fast SCSI; Их «активность» заключается лишь в наличии внутреннего источника опорного напряжения, (ИОН) +2,85 В, питающегося от тех же линий TermPWR. Микросхемы активных терминаторов имеют и электронные ключи, включенные последовательно в каждую линию. Ключи управляются общим сигналом, позволяющим включать-отключать терминатор.
- ◆ *Терминаторы FPTSE* (Forced Perfect Terminator) — улучшенный вариант активных терминаторов с диодными ограничителями выбросов, применяемые в высокоскоростных версиях SE-интерфейса.
- ◆ *Терминаторы для LVD* (рис. 5.4, в) имеют дифференциальной импеданс 105 Ом (линейный — 150 Ом). Здесь два источника опорных напряжений обеспечивают между прямым и обратным проводами смещение 112 мВ (в их пассивном состоянии).
- ◆ *Универсальные терминаторы LVD/SE* сочетают в себе активные SE-терминаторы, дифференциальные терминаторы LVD, схему определения режима и цепи коммутации каждого провода (прямого и обратного) шины SCSI на соответствующие терминирующие цепи.

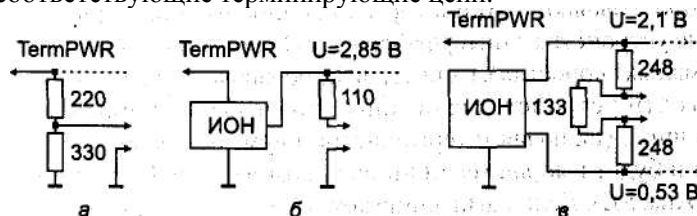


Рис. 5.4. Терминаторы SCSI: а — SE пассивные, б — SE активные, в — LVD'

Универсальные терминаторы LVD/SE, как и остальные устройства, в режиме работы шины по линии DIFSENSE; В старых устройствах SE контакт "азтлема", соответствующий этой линии, был заземлен. Устройства LVD используют этот контакт потенциал 1,3 В. Устройства HVD на этот контакт выводили потенциал выше 2,1 В. В терминаторе имеются компараторы, сравнивающие сигнал (той) линии с эталонами, и логика, переключающая режим терминатора (если обнаружится HVD, терминатор отключает все свои цепи). Специально для универсальных терминаторов...

ных терминаторов выпускаются микросхемы (например, DS2117M, DS2118M фирмы Dallas Semiconductor), выполняющие все функции автоматической терминации для 9 пар проводов. Для терминации 16-разрядной шины данных (Wide SCSI) и сигналов управления требуется 3 таких микросхемы. В микросхемах используются прецизионные резисторы с лазерной подгонкой, что обходится недешево. По исполнению терминаторы могут быть как *внутренними* (размещенными на печатной плате устройства), так и *внешними* (устанавливаемыми 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 крышечкой, под которой смонтирована их «начинка». Несмотря на внешнюю простоту, они имеют ощутимую цену — 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, интерфейс которых не является 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!

**5.1.3. Протокол шины**

Назначение сигналов параллельной шины раскрывает табл. 5.8. Все сигналы являются L-активными: активному состоянию и логической единице соответствует низкий потенциал, что в данной книге отмечено символом «#» после мнемоники цепи. Обратные (парафазные) цепи обозначаются знаком «+».

Таблица 5.8. Назначение сигналов шины SCSI

Сигнал	Назначение
BSY#	Busy — шина занята
SEL#	Select — выбор ЦУ инициатором (Select) или инициатора целевым устройством (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

Сигнал	Назначение
MSG#	Message — передача сообщения
DB[0:31]#	Data Bus — инверсная шина данных
DP[0:3]#	Data Parity — инверсные биты паритета, дополняют количество единичных битов байта до 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 DBO# (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 (в SCSI-1 синхронного режима не было). Шина может находиться в одной из перечисленных ниже *фаз*. Роли источников сигналов между ИУ и ЦУ описаны в табл. 5.9.



Таблица 5.9 Источники сигналов SCSI

Фаза шины	Сигнал		REQ#, C/D#, I/O#, MSG#	ACK#, ATN#	DBx#, DBPx#
	BSY#	SEL#			
Bus Free	-	-	-	-	-
Arbitration	AA <sup>1</sup>	WA <sup>2</sup>	-	-	SID <sup>3</sup>
Selection	I <sup>4</sup> , T <sup>5</sup>	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

<sup>1</sup> AA: источник сигнала — устройство, активное в арбитраже;

<sup>2</sup> WA: источник сигнала — устройство — победитель в арбитраже;

<sup>3</sup> SID: каждое устройство управляет только битом данных, соответствующим значению его SCSI ID;

\*I: источник сигнала — ИУ;

<sup>5</sup> T: источник сигнала — ЦУ.

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.

### 5.1. Параллельные интерфейсы SCSI

131

В фазах *Command*, *Data*, *Status* и *Message* по шине данных передается информация, фазы идентифицируются сигналами MSG#, C/D# и I/O\* (табл. 5.10), 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 5.10. Информационные фазы SCSI

Сигнал			Фаза	Направление
MSG#	C/D#	I/O#		
0	0	0	Data OUT	
0	0	1	Data IN	
0	1	0	Command	
0	1	1	Status	
1	0	0	Зарезервировано	
1	0	1	Зарезервировано	
1	1	0	Message OUT	
1	1	1	Message IN	

Временные диаграммы асинхронного обмена приведены на рис. 5.5. 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 ACK#. Асинхронный обмен поддерживается всеми устройствами для всех фаз передачи информации.

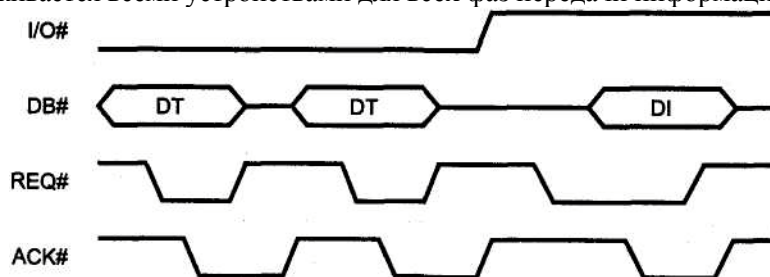


Рис. 5.5. Временные диаграммы асинхронного обмена (DI — данные от ИУ, DT — данные от ЦУ)

Фазы передачи данных *Data OUT* и *Data IN* по предварительной «договоренности» устройств могут выполняться и в синхронном режиме обмена, диаграммы которого приведены на рис. 5.6. 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. 5.6, а), в темпе, ограниченном установленными временными параметрами. ИУ фиксирует принимаемые данные по отрицательному перепаду сигнала REQ#,

но отвечать на них сигналом АСК# может с некоторым опозданием. Как только отставание числа принятых сигналов АСК# от числа посланных REQ# достигнет оговоренного предельного значения (в данном примере — 2), ЦУ приостановит обмен до прихода очередного подтверждения АСК#. Операция считается 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 АСК# (рис. 5.6, б).

### ВНИМАНИЕ

В спецификации SCSI-1 момент возобновления передачи после устранения отставания описан нечетко, в результате разработчики могли посчитать, что очередной запрос (и данные) может последовать лишь после окончания (положительного перепада) сигнала АСК#. Устройство, на это рассчитанное, может терять данные: для него последний сигнал REQ# (и данные) является неожиданным и выглядит как превышение согласованного смещения.

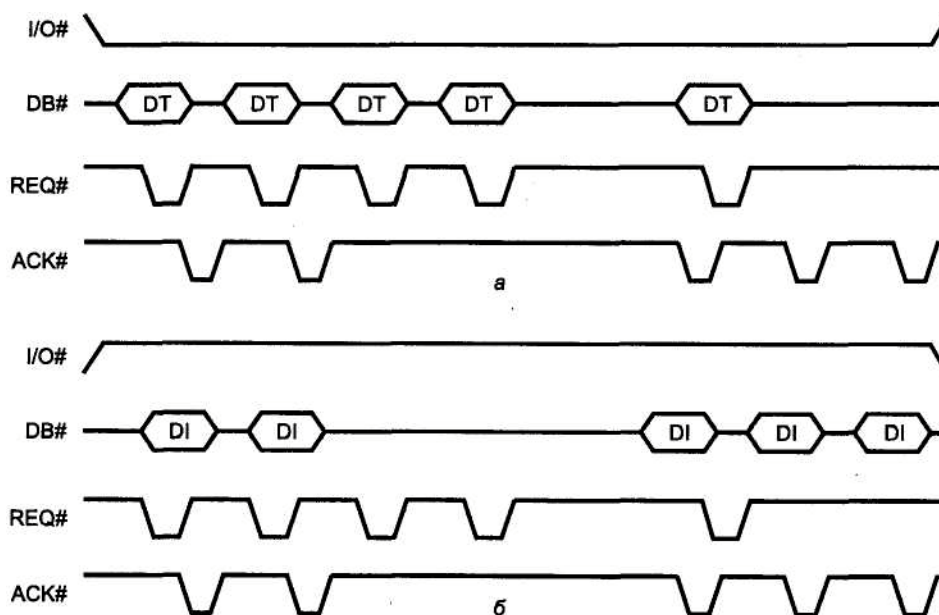


Рис. 5.6. Временные диаграммы синхронного обмена: а — передача; б — прием

Обмен при разрядности 16 бит происходит аналогично. Если в последней фазе данных используются не все байты, передатчик обязан снабдить корректным битом паритета и неиспользуемые байты.

При описании фаз передачи данных 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, распространя-

ющимся по кабелю с ограниченной скоростью. Если в широкой шине имеется пара кабелей (А и В, что на практике встречается редко), то в каждом из них используется своя пара управляющих сигналов (REQ#/ACK# и REQВ#/ACKВ#), поскольку эти кабели могут иметь разную длину.

В фазе *Command* ЦУ запрашивает от ИУ команду. В фазе *Status* ЦУ делает запрос на передачу И У 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 O UT* 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 III* 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 predetermined 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 Pointers* 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, определение реального количества переданных данных с помощью указателя дает ненадежные результаты.

### 5.1.4. Конфигурирование устройств

Все устройства на шине должны быть согласованно сконфигурированы. Для них требуется программно или с помощью джамперов установить перечисленные ниже 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. Ряд версий BIOS считает загрузочным только устройство с SCSI ID=0.
- ◆ *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 устанавливали в специальные гнезда (и извлекать их оттуда). При отсутствии внутренних терминаторов пользователь был вынужден 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)* 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, поддержива-

ющим 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 линии DIFFSENS.
- ◆ *Запрет синхронизации по обоим фронтам* (Disable U160). Позволяет принудительно перевести устройство UltraS SCSI в режим Ultra2.
- ◆ *Старт по команде* (Start on command), или *запрет автоматического запуска шпиндельного двигателя* (Disable Auto Spin up). При установке этого параметра 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).

### 5.1.5. 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 UltraGO и Ultra2 (или еще ниже), то шина будет работать в самом низком из этих режимов. Принудительно запретить режим UltraGO (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) идентификатор 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 работать нормально (см. выше). Поскольку все «узкие» устройства — это SE-устройства, 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). Рассмотрим различные конфигурации подключения устройств к контроллеру SCSI (рис. 5.7). 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.

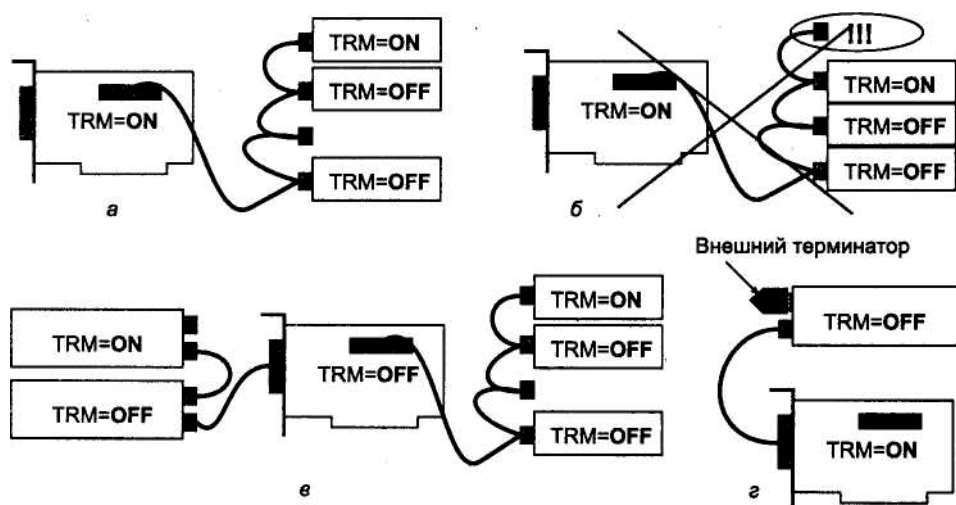


Рис. 5.7. Подключение устройств к карте контроллера SCSI: а, в, г — правильно; б — неправильно

Правила подключения достаточно просты:

- ◆ концы кабельного шлейфа не должны висеть в воздухе (как на рис. 5.7, б);
- ◆ на устройствах, подключенных к концам шлейфа, должны быть включены внутренние терминаторы (на рисунке обозначено как TRM=ON) или же установлен внешний (рис. 5.7, 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. 5.7, а) или только внешнее (рис. 5.7, г) подключение, терминатор на



контроллере *включают*. Когда используется и внешнее, и внутреннее подключение (рис. 5.7, *в*), терминатор на контроллере *отключают*.

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 включают/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. Если требуется только внутреннее или только внешнее подключение (рис. 5.8, *а* и *б*), то терминатор на контроллере *включают*. Если используется и внутреннее, и внешнее подключение (рис. 5.8, *в*), терминатор на контроллере *отключают*. Если используется универсальный кабель-шлейф с внутренними и внешним разъемами (как на рис. 5.8, *в*), но внешних устройств нет, то терминатор на контроллере должен быть *отключен*, а на внешнем разъеме должен быть установлен *внешний терминатор*.

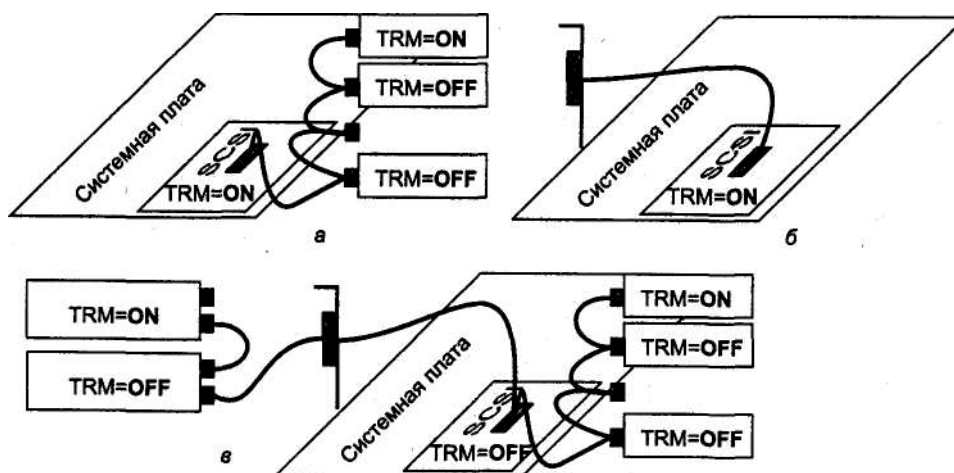


Рис. 5.8. Подключение устройств к интегрированному контроллеру SCSI

Кабели и терминаторы могут входить в комплект адаптеров SCSI или системных плат со встроенным контроллером SCSI, а могут приобретаться отдельно. То, что входит в стандартный комплект поставки, не всегда подходит для конкретного применения. Так, например, в комплекте с системной платой СТ-6BTS, имеющей контроллер Ultra-Wide SCSI, поставляется универсальный «широкий» (wide) шлейф (как показанный на рис. 5.8, *в*, но с меньшим количеством внутренних разъемов) и «узкий» внутренний. Для подключения только внутренних дисководов Wide SCSI такого комплекта недостаточно: нужен внешний терминатор! Мож-

но, 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 СТ-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. 5.9, *а* и *б*, приведены корректные способы смешанного подключения (устройства с терминаторами на концах шлейфов подразумеваются). На рис. 5.9, *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 (именно они и изображены на рис. 5.9, *в*) корректное смешанное подключение выполнить нельзя.

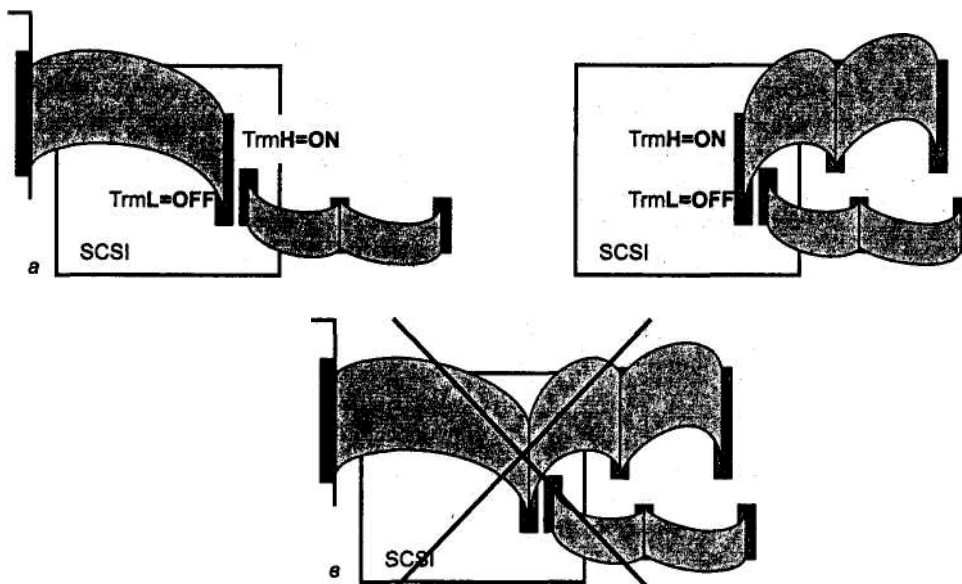


Рис. 5.9, Подключение узких и широких устройств SCSI: *а, б* — правильно; *в* — неправильно

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». 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, с не всегда заметными сбоями. 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 терминатора устройства работать вообще не будут. 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.

## 5.2. Интерфейс Fibre Channel

Кроме параллельного интерфейса, SCSI-3 может использовать и *последовательный* интерфейс *Fibre (Fiber) Channel*, или *FCAL (Fibre Channel Arbitrated Loop* — кольцо волоконного канала с арбитражем), который 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 относящихся к разным протоколам (например, SCSI и IP). Аппаратура Fibre 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 разделения (совместного 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 разделение ресурсов, обеспечивая и резервирование линий связи, но здесь пока имеются сложности на уровне операционных систем.

### 5.3. Хост-адаптер SCSI

Хост-адаптер является важнейшим узлом интерфейса, определяющим производительность системы SCSI. В его задачу входит передача данных между хостом (программой, исполняемой центральным процессором) и другими устройствами, подключенными к шине, по протоколам вышеописанных физических интерфейсов. Структуры передаваемых блоков данных и команды устройств стандартизованы, их описание приводится в литературе [4, 9]. Однако архитектуры и программные модели адаптеров не стандартизованы (в отличие, например, от адаптеров AT A). Существует широкий спектр адаптеров, к простейшим можно подключать только устройства, некритичные к производительности. Такие адаптеры могут входить, например, 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 расширения (ISA, EISA, MCA, PCI, VLB, PCMCIA, CardBus), шин USB и FireWire и для LPT-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 с 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 теста POST. Как и всякая карта расширения, хост-адаптер должен быть сконфигурирован 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 одностипных хост-адаптеров, для них используется ROM BIOS с одного адаптера, а 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, в том числе и хост-адаптеру, требуются специальные драйверы. Базовый драйвер дисковых устройств входит в BIOS хост-адаптера; он обычно эмулирует трехмерную адресацию дискового сервиса Int 13h. 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.

## 5.4. Управление интерфейсом и выполнение команд

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.

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* с 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 асин-

кронному 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, чем приведет к установке predetermined режимов по включению. Согласование режимов не должно инициироваться в каждом процессе, поскольку затраты времени на эту процедуру сводят на нет выигрыш в производительности.

*Система команд SCSI* включает общие команды, применимые для устройств всех классов, и специфические для каждого класса. 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, определяемая кодом операции (первым байтом блока), может составлять 6, 10 или 12 байт.

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. ЦУ переходит в фазу *Command* и принимает блок дескриптора команды *Read*. Интегрировав команду, ЦУ переходит в фазу *Data IN*, передает запрошенные данные, затем переводится в фазу *Status* и посылает состояние *Good*. Затем в фазе *Message IN* устройство посылает сообщение *Command Complete*, после чего освобождает шину (фаза *Bus Free*). Процесс завершен.

Рассмотрим тот же пример, но при условии *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*. ИУ вернет соответствующий набор указателей в активное состояние и продолжит выполнение процесса, как описано выше. Если ЦУ хочет отсоединиться, когда часть данных уже передана (например, головка диска 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 являются частью одного процесса. Команды не являются полностью независимыми — при 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 прочесть 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, занятым процессом от одного ИУ, принимать команды (начинать процесс) с другими ИУ.

*Маркированные очереди* (tagged queue) определены в SCSI-2 для ЛУ. Для каждой связи I\_T\_L (ИУ-ЦУ-ЛУ) существует своя очередь размером до 256 процессов. 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.

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 в цепочке команд, поскольку цепочка принадлежит одному процессу, а в очередь ставятся именно процессы. Здесь мы не рассматриваем 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.

## Глава 6

# Шины и карты расширения

Шины расширения (Expansion Bus) являются средствами подключения системного уровня: они позволяют адаптерам и контроллерам непосредственно использовать системные ресурсы РС — пространства памяти и ввода-вывода, прерывания, каналы прямого доступа к памяти. Устройства, подключенные к шинам расширения, могут и сами управлять этими шинами, получая доступ к остальным ресурсам компьютера (обычно к ячейкам памяти). Такое прямое управление (bus mastering) позволяет разгружать центральный процессор и добиваться высоких скоростей обмена данными. Шины расширения механически реализуются в виде слотов (щелевых разъемов) или штырьковых разъемов; для них характерна малая длина проводников, что позволяет достигать высоких частот работы. Эти шины могут и не выводиться на разъемы, но использоваться для подключения устройств в интегрированных системных платах.

В современных компьютерах основной шиной расширения является PCI; ее дополняет порт AGP. Шина ISA из настольных компьютеров уходит, но она сохраняет свои позиции в промышленных и встраиваемых компьютерах, как в традиционном слотовом варианте, так и в «бутербродном» варианте PC/104. В блокнотных компьютерах широко применяются слоты PCMCIA с шинами PC Card и Card Bus. Шина LPC является современным дешевым средством подключения нересурсоемких устройств на системной плате. Все эти шины подробно рассматриваются в данной главе. Информацию по отжившим шинам MCA, EISA, VLB можно найти в литературе [1, 2, 5].

Изготовителям карт расширения приходится точно следовать протоколам шины, включая жесткие частотные и нагрузочные параметры, а также временные диаграммы. Отклонения приводят к несовместимости с некоторыми системными платами. Если при подключении к внешним интерфейсам это ведет к неработоспособности только самого устройства, то некорректное подключение к системной шине может блокировать работу всего компьютера. Следует также учитывать ограниченность ресурсов РС. Самые дефицитные из них — линии запросов прерываний; проблема прерываний, известная по шине ISA, так и не была радикально решена с переходом на PCI. Другой дефицит — каналы прямого доступа шины ISA, используемые и для прямого управления шиной, — в шине PCI преодолен. Доступное адресное пространство памяти и портов ввода-вывода, в котором было тесновато абонентам шины ISA, в PCI существенно расширено. Проблемы распределения ресурсов на шинах решаются по-разному, но чаще всего применяется технология PnP.



В табл. 6.1 дана характеристика стандартных шин расширения PC.

**Таблица 6.1.** Характеристики шин расширения

Шина	Пропускная способность, DMA	Каналы Bus-	Bus-Master	ACFG <sup>2</sup> Мбайт/с <sup>1</sup>	Разрядность данных	Разрядность адреса	Частота, МГц
ISA-8	4	3			8	20	8
ISA-16	8	7	+		16	24	8
LPC	6,7	7	+		8/16/32	32	33
EISA	33,3	7	+	+	32	32	8,33
MCA-16	16		+	+	16	24	10
MCA-32	20		+	+	32	32	10
VLB	132				32/64	32	33-50(66)
PCI	132/264		+	+	32/64	32	33/66
PCI-X	532/1064		+	+	32/64	32/64	33/66
AGP	266/532/1064		+	+	32	32/64	66
1x/2x/4x							
PCMCIA	10/20	+		+	8/16	26	10
Card Bus	132		+	+	32	32	33

<sup>1</sup> Указана максимальная пропускная способность. Реальная примерно в 2 раза ниже за счет прерываний, регенерации и протокольных процедур.

<sup>2</sup> Поддержка автоматического конфигурирования. Для ISA PnP является позднейшей надстройкой, реализуемой адаптерами и ПО.

## 6.1. Шины ISA, EISA и PC/104

ISA Bus (Industry Standard Architecture) — шина расширения, применявшаяся с первых моделей PC и ставшая промышленным стандартом, В компьютере XT использовалась шина с разрядностью данных 8 бит и адреса — 20 бит. В компьютерах AT ее расширили до 16 бит данных и 24 бит адреса. Конструктивно, как показано на рис. 6.1, шина выполнена в виде двух шелевых разъемов с шагом выводов 2,54 мм (0,1 дюйма). В подмножестве ISA-8 используется только 62-контактный слот (ряды А, В), в ISA-16 применяется дополнительный 36-контактный слот (ряды С, D). Шина PC/104, разработанная для встраиваемых контроллеров на базе PC, отличается от обычной ISA только конструктивно. В шине EISA — дорогим 32-разрядном расширении ISA — используется «двухэтажный» слот, позволяющий устанавливать и обычные карты ISA.

Для шины ISA выпущено (и продолжает выпускаться) огромное количество разнообразных карт расширения. Ряд фирм выпускает карты-прототипы (Prototype Card), представляющие собой печатные платы полного или уменьшенного формата с крепежной скобой. На платах установлены обязательные интерфейсные цепи — буфер данных, дешифратор адреса и некоторые другие. Остальная часть платы свободна, и здесь разработчик может разместить макетный вариант своего устройства. Эти платы удобны для проверки нового изделия, а также для монтажа единичных экземпляров устройства, когда разработка и изготовление печатной платы нерентабельно.

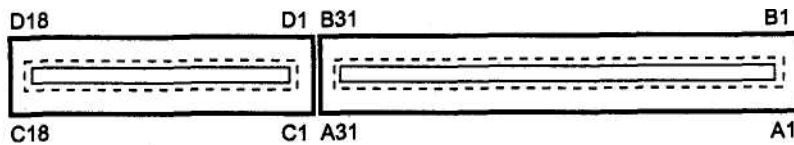


Рис. 6.1. Слот ISA

В каждый момент времени шиной может управлять только одно устройство-задатчик, обращающееся к ресурсам (портам или ячейкам памяти) *устройств-исполнителей*. Шина ISA обеспечивает возможность обращения к 8- или 16-битным регистрам устройств, отображенным на пространства ввода-вывода и памяти. *Диапазон адресов памяти* для устройств ограничен областью верхней памяти UM A (A0000-FFFFh). Для шины ISA-16 настройками CMOS Setup может быть разрешено пространство между 15-м и 16-м мегабайтом памяти (при этом компьютер не сможет использовать более 15 Мбайт ОЗУ). Для шины ISA *диапазон адресов ввода-вывода* сверху ограничен количеством задействованных для дешифрации бит адреса, нижняя областью адресов 0-FFh недоступна (зарезервирована под устройства системной платы). В PC была принята 10-битная адресация ввода-вывода, при которой линии адреса A[15:10] устройствами игнорировались. Таким образом, диапазон адресов устройств шины ISA ограничивается областью 100h—3FFh. Впоследствии стали применять 12-битную адресацию (диапазон 100h-FFFh). При этом приходится учитывать возможность присутствия на шине старых 10-битных адаптеров, которые «отзовутся» на адрес с подходящими ему битами A[9:0] во всей допустимой области 12-битного адреса четыре раза (у каждого 10-битного адреса будет еще по три 12-битных псевдонима). Полный 16-битный адрес используется только в шинах EISA и PCI.

Шина ISA-8 может предоставить до 6 линий запросов прерываний, ISA-16 — 11. Часть из них могут «отобрать» устройства системной платы или шина PCI.

Шина ISA-8 позволяет использовать до трех 8-битных каналов DMA. На 16-битной шине доступны еще три 16-битных и один 8-битный канал.

Все перечисленные ресурсы шины должны быть бесконфликтно распределены. Бесконфликтность подразумевает выполнение перечисленных ниже условий.

- ◆ Каждое устройство-исполнитель должно управлять шиной данных только при чтении по его адресам или по используемому им каналу DMA. Области адресов, по которым выполняется чтение регистров различных устройств, не должны пересекаться. Поскольку при записи шиной данных управляет лишь текущий задатчик, возможность конфликтов, приводящих к искажениям данных, исключена. «Подсматривать» операции записи, адресованные не данному устройству, не возбраняется.
- ◆ Назначенную линию IRQ<sub>x</sub> или DRQ<sub>x</sub> устройство должно держать на низком уровне в пассивном состоянии и переводить в высокий уровень для активации запроса. Неиспользуемыми линиями запросов устройство управлять не имеет права, они должны электрически отсоединяться или подключаться к буферу, находящемуся в третьем состоянии. Одной линией запроса может пользоваться только одно устройство. Такая нелепость (с точки зрения схемотехники TTL) была допущена в первых PC и из требований совместимости тиражируется до сих пор.

Задача распределения ресурсов для старых адаптеров решалась с помощью джамперов, затем появились программно конфигурируемые устройства, которые вытесняются автоматически конфигурируемыми платами PnP.

Назначение контактов слотов шин ISA и EISA приведено в табл. 6.2 и

**6.3. Таблица 6.2. Основной разъем шин ISA-8, ISA-16 и EISA**

Ряд В		РЯДА
GND	1	IOCH
Reset	2	SD7
+5 В	3	SD6
IRQ2/9 <sup>1</sup>	4	SD5
-5 В	5	SD4
DRQ2	6	SD3
-12 В	7	SD2
OWS# <sup>2</sup>	8	SD1
+12 В	9	SD0
GND	10	IOCH
SMEMW#	и	AEN
SMEMR#	12	SA19
IOWR#	13	SA18
IORD#	14	SA17
DACK3#	15	SA16
DRQ3	16	SA15
DACK1#	17	SA14
DRQ1	18	SA13
REFRESH»	19	SA12
BCLK	20	SA11
IRO7	21	SA10
IRQ6	22	SA9
IRQ5	23	SA8
IRQ4	24	SA7
IRQ3	25	SA6
DACK2#	26	SA5
TC	27	SA4
BALE	28	SA3
+5 В	29	SA2
Osc	30	SA1
GND	31	SA0

<sup>1</sup>B4: XT=IRQ2, AT=IRQ9.

<sup>2</sup>B8: XT-Card Selected.

**Таблица 6.3.** Дополнительный разъем шин ISA-16 и EISA

РядО	№	РядС
MEMCS16#	1	SBHE#
IOCS16#	2	LA23
IRQ10	3	LA22
IRQ 11	4	LA21
IRQ12	5	LA20
IRQ 15	6	LA19
IRQ14	7	LA18
DACKO#	8	LA17
DRQO	9	MEMR#
DACK5#	10	MEMW
DRQ5	11	SD8
DACK6#	12	SD9
DRQ6	13	SD10
DACK7#	14	SD11
DRQ7	15	SD12
+5B	16	SD13
MASTER*	17	SD14
GND	18	SD15

Сигналы шины ISA естественны для периферийных микросхем фирмы Intel (в стиле семейства 8080). Набор сигналов ISA-8 предельно прост. Программное обращение к ячейкам памяти и пространства ввода-вывода обеспечивают следующие сигналы.

- ◆ SD[7:0] — шина данных. Иное название сигналов — Data или D.
- ◆ SA[19:0] (Addr[19:0], A[19:0]) - шина адреса.
- ◆ AEN — разрешение адресации портов (запрещает ложную дешифрацию адреса в цикле DMA).
- ◆ IOW# (IOWC#, IOWR#) - запись в порт.
- ◆ IOR# (IORC#, IORD#) - чтение порта.
- ◆ SMEMW\* (SMEMWR#, SMWTC#) — запись в системную память (в диапазоне адресов 0-FFFFFFh).
- ◆ SMEMR\* (SMEMRD#, SMRDC#) — чтение системной памяти (в диапазоне адресов 0-FFFFFFh).

Ниже перечислены сигналы, относящиеся к сигналам запросов прерывания и каналам прямого доступа к памяти.

- ◆ IRQ2/9, IRQ[3:7] — запросы прерываний. Положительный перепад сигнала вызывает запрос аппаратного прерывания. Для идентификации источника высокий уровень должен сохраняться до подтверждения прерывания процессором, что затрудняет разделение (совместное использование) прерыва-

ний. Линия IRQ2/9 в шинах XT вызывает аппаратное прерывание с номером 2, а в AT — с номером 9.

- ◆ DRQ[1:3] — запросы 8-битных каналов DMA (положительным перепадом).
- ◆ DACK[1:3]# — подтверждение запросов 8-битных каналов DMA.
- ◆ TC — признак завершения счетчика циклов DMA.

Шина имеет и несколько служебных сигналов синхронизации, сброса и регенерации памяти, установленной на адаптерах.

- ◆ IOCHRDY (CHRDY, IOCHRDY) — готовность устройства, низкий уровень удлинит текущий цикл (не более 15 икс).
- ◆ BALE (ALE) — разрешение защелки адреса. После его спада в каждом цикле процессора линии SA[0:19] гарантированно содержат действительный адрес.
- ◆ REFRESH\* (REF#) — цикл регенерации памяти (в XT называется DACKO#).  
Сигнал появляется каждые 15 мкс, при этом шина адреса указывает на очередную регенерируемую строку памяти.
- ◆ ЮСНК# — контроль канала, низкий уровень вызывает NMI CPU (разрешение и индикация в системных портах 061h, 062h).
- ◆ RESET (RESDRV, RESETDRV) — сигнал аппаратного сброса (активный уровень — высокий).
- ◆ BCLK (CLK) — синхронизация шины с частотой около 8 МГц. ПУ могут не использовать этот сигнал, работая только по управляющим сигналам записи и чтения.
- ◆ OSC — несинхронизированная с шиной частота 14,431818 МГц (использовалась старыми дисплейными адаптерами).

Кроме логических сигналов шина имеет контакты для разводки питания +5, -5, +12 и -12 В.

*Дополнительный разъем*, расширяющий шину до 16-битной, содержит линии данных, адреса, запросов прерываний и каналов прямого доступа.

- ◆ SD[15:8] — шина данных.
- ◆ SBHE# — признак наличия данных на линиях SD[15:8].
- ◆ LA[23:17] — нефиксированные сигналы адреса, требующие защелкивания по спаду сигнала BALE. Такой способ подачи адреса позволяет сократить задержку. Кроме того, схемы дешифратора адреса памяти плат расширения начинают декодирование несколько раньше спада BALE.
- ◆ IRQ[10:12], IRQ[14:15] — дополнительные запросы прерываний.
- ◆ DRQ[5:7] — запросы 16-битных каналов DMA (положительным перепадом).
- ◆ DACK[5:7]# — подтверждение запросов 16-битных каналов DMA.
- ◆ DRQO и DACKO\* — запрос и подтверждение 8-битного канала DMA, освободившегося от регенерации памяти.

Перечисленные ниже сигналы связаны с переключением разрядности данных.

- ◆ MEMCS16#(M16#)— адресуемое устройство поддерживает 16-битные обращения к памяти.
- ◆ IOCS16\* (I/OCS16\*, IO16#) — адресуемое устройство поддерживает 16-битные обращения к портам.

К новым управляющим сигналам относятся следующие.

- ◆ MEMW# (MWTC#) — запись в память в любой области до 16 Мбайт.
- ◆ MEMR# (MRDC#) — чтение памяти в любой области до 16 Мбайт.
- ◆ OWS# (SRDY#, NOWS#, ENDXFR) — укорочение текущего цикла по инициативе адресованного устройства.
- ◆ MASTER\* (MASTER 16#) — запрос от устройства, использующего 16-битный канал DMA на управление шиной. При получении подтверждения DACK [5:7] Bus-Master может захватить шину.

В шине EISA на дополнительных контактах слотов (недоступных картам ISA) располагается расширение шин данных и адреса до 32 бит, а также набор сигналов, обеспечивающих передачу данных в синхронном режиме с возможностью пакетных циклов.

### 6.1.1. Обычная передача данных

Для передачи данных от исполнителя к задатчику предназначены *циклы чтения* ячейки памяти или порта ввода-вывода, для передачи данных от задатчика к исполнителю — *циклы записи* ячейки памяти или порта ввода-вывода. В каждом цикле текущий (на время данного цикла) задатчик формирует адрес обращения и управляющие сигналы, а в циклах записи еще и данные на шине. Адресуемое устройство-исполнитель в соответствии с полученными управляющими сигналами принимает (в цикле записи) или формирует (в цикле чтения) данные. Также оно может, при необходимости, управлять длительностью цикла и разрядностью передачи. Обобщенные временные диаграммы циклов чтения или записи памяти или ввода-вывода приведены на рис. 6.2. Здесь условный сигнал CMD\* изображает один из следующих сигналов:

- ◆ SMEMR#, MEMR# — в цикле чтения памяти;
- ◆ SMEMW#, MEMW# — в цикле записи памяти;
- ◆ IOR# — в цикле чтения порта ввода-вывода;
- ◆ IOW# — в цикле записи порта ввода-вывода.

В каждом из рассматриваемых циклов активными (с низким уровнем) могут быть только сигналы лишь из одной строки данного списка, и во время всего цикла сигнал AEN имеет низкий уровень. Цикл прямого доступа к памяти, в котором это правило не соблюдается, рассмотрен ниже, и в таком цикле сигнал AEN будет иметь высокий уровень. Сигналы SMEMR\* и SMEMW\* вырабатываются из сигналов MEMR# и MEMW# соответственно, когда адрес принадлежит диапазону 0-FFFFh. Поэтому сигналы SMEMR\* и SMEMW\* задержаны относительно MEMR# и MEMW\* на 5-10 нс.

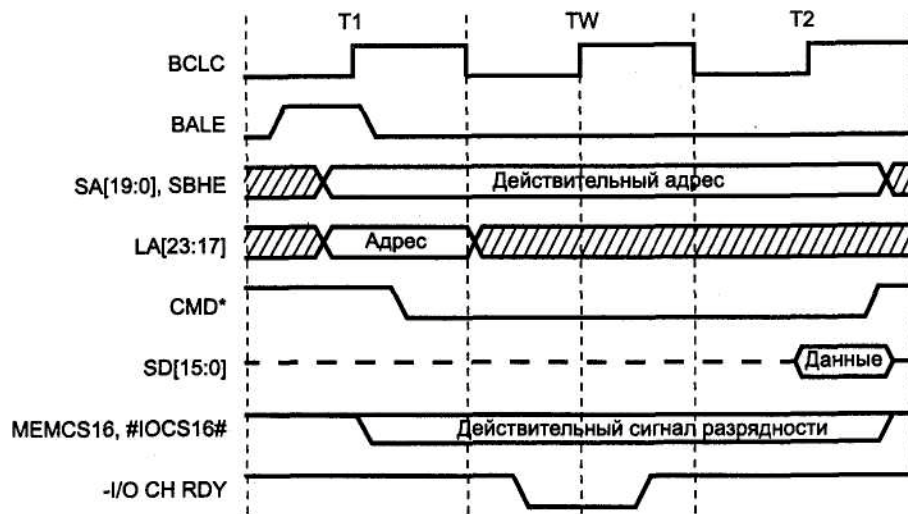


Рис. 6.2. Временные диаграммы циклов чтения или записи на шине ISA

В начале каждого цикла контроллер шины устанавливает адрес обращения: на линиях SA[19:0] и SBHE# действительный адрес сохраняется на время всего текущего цикла; на линиях LA[23:17] адрес действителен только в начале цикла, так что требуется его «зашелкивание». Каждое устройство имеет *дешифратор адреса* — комбинационную схему, срабатывающую только тогда, когда на шине присутствует адрес, относящийся к данному устройству. В фазе адресации устройства еще «не знают», к какому из пространств (памяти или ввода-вывода) относится выставленный адрес. Но дешифраторы адресов уже срабатывают, и, когда в следующей фазе шина управления сообщает тип операции, адресуемое устройство уже оказывается готовым к ее исполнению. Если устройство использует линии LA[23:17] (они нужны лишь для обращений к памяти выше границы FFFFh), то они на дешифратор адреса должны проходить через регистр-зашелку, «прозрачный» во время действия сигнала BALE и фиксирующий состояние выходов по его спаду. Это позволяет дешифратору, всегда вносящему некоторую задержку, начинать работу раньше, чем поступит управляющий сигнал чтения или записи. При обращении к портам ввода-вывода сигналы LA[32:17] не используются.

Если устройство имеет более одного регистра (ячейки), то для выбора конкретного регистра (ячейки) ему требуется несколько линий адреса. Как правило, старшие биты шины адреса поступают на вход дешифраторов адреса, формирующих сигналы выборки устройств, а младшие биты — на адресные входы самих устройств. Тогда каждое устройство в пространстве будет занимать наиболее компактную область смежных адресов размером в  $2^p$  байт, где  $p$  — номер младшей линии адреса, поступающей на дешифратор. Из них реально необходимы  $2^m$  адресов, где  $m$  — номер самой старшей линии адреса, участвующей в выборе регистра устройства. В идеале должно быть  $p=m+1$ : при большем значении  $p$  отведенное (по дешиф-

ратору) пространство адресов не будет использовано полностью и регистры устройства будут повторяться в отведенной области  $2^{n/m+1}$  раз, то есть у них появятся *адреса-псевдонимы* (alias). Адреса-псевдонимы будут отличаться от истинного адреса (минимального из всех псевдонимов) на  $K \times 2^{m+1}$ , где  $K$  — целое число. Меньшее значение  $n$  недопустимо, поскольку тогда не все регистры устройства будут доступны задатчику. В принципе можно использовать дешифратор адреса, срабатывающий только на какой-то части адресов из области  $2^{n+1}$  (не кратной степени двойки), если устройству требуется «неудобное» количество регистров. Однако на практике «фигурное выпиливание» областей из пространства адресов обычно не делают, так что часть адресов может пропадать бесполезно.

*Разрядность данных* в каждом цикле обращения определяется потребностями текущего задатчика и возможностями исполнителя. В IBM PC/XT и системная шина, и шина ISA были 8-разрядными, так что вопросов согласования разрядности не возникало. В IBM PC/AT286 (и 386-SX) системная шина уже 16-разрядная, и в современных ПК с 32- и 64-разрядными системными шинами контроллер шины ISA является ее 16-разрядным задатчиком. На системной плате имеется «косой буфер», он же перестановщик байтов, который при необходимости транслирует данные с младшего байта шины на старшую или обратно. Логика управления этим буфером использует сигналы SBHE#, SAO, IOCS16\* и MEMCS16\*. Поддержка 16-разрядных передач сообщается адресуемому исполнителем сигналами IOCS16\* и MEMCS16\* при срабатывании его дешифратора адреса. Сигнал IOCS16# влияет только на разрядность обращений к портам, MEMCS16\* — к памяти. Все операции обмена (транзакции) начинаются задатчиком единообразно, поскольку он еще не «знает» возможностей исполнителя. Развитие событий зависит от намерений задатчика и полученных сигналов разрешения 16-битных передач. В чисто 16-разрядных машинах начальный адрес однозначно соответствует передаваемому байту или младшему байту передаваемого слова<sup>1</sup>. В машинах с 32-разрядными процессорами начальный адрес, выставляемый на шине в начале транзакции, зависит от разрядности данных, запланированной задатчиком, и может зависеть от положения адресуемых данных относительно границы двойного слова (32 битного). 16-разрядные передачи выполняются за 1 цикл только при условии передачи по четному адресу (ACHO) и при ответе исполнителя сигналом IOCS16\* или MEMCS16\*, в иных случаях они разбиваются на два цикла. 32-разрядные передачи будут разбиваться на 2 (16+16), 3 (8+16+8) или 4 (8+8+8+8) цикла, в зависимости от возможностей исполнителя и четности адреса. Порядок, в котором передаются байты (во времени), неоднозначен (возможен как инкремент, так и декремент адреса), но в адресном пространстве они раскладываются по своим местам однозначно.

В табл. 6.4 приводятся состояния сигналов шины ISA для различных вариантов записи в порты ввода-вывода, проверенные экспериментальным путем. Вывод

<sup>1</sup> То есть адрес слова в L-N порядке, принятом для изделий Intel.



16-разрядных данных выполнялся командой OUT DX,AX (в DX — адрес порта, в AX — данные; AL содержит младший байт, AH — старший), вывод 8-разрядных командой OUT DX,AL. Несколько неожиданные (для автора) варианты 3 и 6 с тем адреса, возможно, будут иметь место не на всех системных платах, но их следует иметь в виду при проектировании устройств, претендующих на глобальную совместимость. Правда на практике 16-битных передач по нечетным адресам обычно избегают (даже чисто подсознательно), и побочные эффекты от такого порядка маловероятны.

**Таблица 6.4.** Состояние сигналов при 8- и 16-битных обращениях к устройству ISA

Сигнал (шина)	1 цикл	2 цикл
<i>1. Вывод 16-разрядных данных в 16-битное устройство по четному адресу</i>		
SBHE#	L	
SA	DX(A0=0)	
D[15:8]	AH	
D[7:0]	AL	-
IOCS16#	L	
<i>2. Вывод 16-разрядных данных в 16-битное устройство по нечетному адресу xxx1,xxx5,xxx9,xxxD</i>		
SBHE#	L	H
SA	DX(A0=1)	DX+1 (A0=0)
D[15:8]	AL	0
D[7:0]	AL	AH
IOCS16#	L	L
<i>3. Вывод 16-разрядных данных в 16-битное устройство по нечетному адресу xxx3,xxx7,xxxB,xxxF</i>		
SBHE#	H	L
SA	DX+1 (A0=0)	DX (A0= 1)
D[15:8]	0	AL
D[7:0]	AH	0
IOCS16*	L	L
<i>4. Вывод 16-разрядных данных в 8-битное устройство по четному адресу</i>		
SBHE#	L	L
SA[1:0]	DX(A0=0)	DX+1(A0=1)
D[15:8]	AH	AH
D[7:0]	AL	AH
IOCS16*	H	H
<i>5. Вывод 16-разрядных данных в 8-битное устройство по нечетному адресу xxx1,xxx5,xxx9,xxxD</i>		
SBHE#	L	H
SA[ 1:0]	DX (A0= 1)	DX+1 (A0=0)
D[15:8]	AL	0
D[7:0]	AL	AH
IOCS16#	H	H

Сигнал (шина)	1 цикл	2 цикл
<i>6. Вывод 16-разрядных данных в 8-битное устройство по нечетному адресу xxx3,xxx7, xxxB,xxxF</i>		
SBNE#	H	L
SA[1:0]	DX+1(AO=0)	DX(AO=1)
D[15:8]	0	AL
D[7:0]	AH	AL
IOCS16#	H	H
<i>7. Вывод 8-разрядных данных в 16-битное устройство по четному адресу</i>		
SBNE#	H	
SA[1:0]	DX(AO=0)	
D[15:8]	0	
D[7:0]	AL	
IOCS16*	L	
<i>8. Вывод 8-разрядных данных в 16-битное устройство по нечетному адресу</i>		
SBNE#	L	
SA[1:0]	DX(AO=1)	
D[15:8]	AL	
D[7:0]	0(AL?)	
IOCS16*	L	

Момент помещения действительных данных на линии SD[15:0] определяется управляющими сигналами чтения/записи, так что исполнителю не требуется синхронизация с тактовым сигналом шины. В циклах чтения адресованный исполнитель должен выдать данные на шину по началу (спаду) соответствующего сигнала чтения (IOR#, MEMR#, SMEMR#) и удерживать их до конца действия сигнала (пока не произойдет подъем сигнала). В циклах записи задатчик выставляет действительные данные несколько позже начала (спада) сигнала записи (IOW#, MEMW#, SMEMW#). Устройство-исполнитель должно фиксировать для себя эти данные в конце цикла по подъему сигнала записи. От устройства-исполнителя не предусматривается никаких подтверждений исполнения циклов; длительность цикла устанавливает задатчик, но исполнитель может потребовать удлинения или укорочения циклов. С помощью сигнала IOCHRDY исполнитель может удлинить цикл на произвольное число тактов, при этом задатчик будет вводить дополнительные такты ожидания (wait states). Обычно контроллер шины следит за длительностью цикла и по достижении критического времени принудительно его завершает (по тайм-ауту, возможно, и не сообщая об этом событии). Слишком длинные циклы тормозят работу компьютера, а превышение длительности 15 мкс может привести к сбою регенерации и потере данных в ОЗУ. С помощью сигнала OWS# исполнитель предлагает задатчику укоротить цикл, исключив такты ожидания. Реакция задатчика на одновременное использование сигналов IOCHRDY и OWS# непредсказуема, этой ситуации следует избегать.

#### **ВНИМАНИЕ**

Некорректное управление сигналом IOCHRDY (его «залипание» на низком уровне) тормозит работу компьютера.

Номинальная длительность цикла определяется чипсетом и может программироваться в BIOS Setup заданием числа тактов ожидания (wait states). При этом циклы обращения к памяти, как правило, короче циклов обращения к портам ввода-вывода. Для управления длительностью цикла используются также сигналы управления разрядностью передачи: если устройство поддерживает 16-битные передачи, предполагается, что оно может работать с меньшим количеством тактов ожидания. Этим объясняется, что в BIOS Setup длительности циклов ISA задаются отдельно как для памяти и ввода-вывода, так и для 8- и 16-битных операций. Кроме длительности цикла, устройства могут быть критичны к времени восстановления (recovery time) — длительности пассивного состояния управляющих сигналов чтения-записи между циклами. Этот параметр также может программироваться в BIOS Setup и тоже отдельно для 8- и 16-разрядных операций.

Карты расширения для подключения к шине данных, как правило, используют буферные микросхемы, отдельные для линий SD[7:0] и SD[15:8]. Здесь широко применяются микросхемы 74ALS245 (1533АП6) — 8-разрядные двунаправленные приемопередатчики. Буфер должен открываться сигналом OE# (Output Enable — разрешение выхода), когда на шине адреса присутствует адрес, относящийся к диапазону адресов подключаемого устройства. «Дежурным» является направление передачи «от шины — к устройству»; переключение в обратную сторону производится по сигналу IOR#, если устройство представляет порты ввода-вывода, или MEMRD\*, если устройство приписано к пространству памяти. Таким образом, буферы имеют право передавать данные на шину (управлять шиной данных) только во время действия сигнала чтения, относящегося к зоне адресов данного устройства. Карта расширения может являться комбинацией 8- и 16-битных устройств; например, некогда популярные мультикарты содержали 16-битный адаптер AT A и набор 8-битных контроллеров портов COM, LPT, GAME и контроллера НГМД. В таких картах логика управления буферами и сигналами IOCS16\* и MSC16\* управляется сигналами от дешифратора адреса. Если устройство по данному адресу является 8-разрядным (не формирует сигналы IOCS16\* или MSC16\*), то оно имеет право разрешать чтение только через буфер линий SD[7:0], а буфер старших линий SD[15:8] (если он имеется на карте) должен быть переведен в третье состояние. Если устройство по данному адресу является 16-разрядным, то оно формирует сигнал IOCS16\* или MSC16\*, а разрешением буферов управляют сигналы SBHE\* и SAO. В этом случае буфер линий SD[7:0] разрешается только при SAO=0, а буфер линий SD[15:8] разрешается только при SBHE#=L. Некорректное разрешение буферов может приводить к их конфликту с перестановщиком байтов системной платы и искажениям данных.

Восьмиразрядные устройства (например, микросхемы 8255, 8250, 8253 и т. п.) следует подключать только к линиям SD[7:0] и при обращении к ним не формировать сигналы IOCS16\* или MSC16\*. Никакие «косые» буферы (перестановщики байтов) на интерфейсных картах не нужны.

В одном из источников описывается эффект перестановки байтов при обращении к порту ввода-вывода: «Если прочитать слово из порта по четному адресу, значение одно, а если по нечетному — старшие 8 бит предыдущего значения становятся

младшими, а старшие нового = FFh». Первые подозрения падают на ошибку в логике управления буферами. На самом деле все объясняется гораздо проще. Пусть имеется устройство с двухбайтным регистром, младший байт которого имеет адрес RO (четный), старший — RO+1, а по адресу R+2 устройство (и никакие другие) не откликается. Пусть в данный момент в нем записано число AA55h, тогда чтением порта по команде IN AX, R0 получим в регистрах процессора AL=55h, AH=AAh. Теперь если попытаться его «прочитать по нечетному адресу», то есть командой IN AX, R0+1, то получим AL=AAh (содержимое RO+1, к которому мы на самом деле и адресовались!), а AH=FFh (результат чтения «пустоты»). Так что это не «эффект перестановки», а просто незнание общего правила «интеловской» адресации: адресом слова (двойного, учетверенного...) является адрес его младшего байта. Если в нашем устройстве применяется неполная дешифрация адреса (линия SA1 не используется ни для дешифрации адреса, ни для выбора регистра), то мы увидим полную перестановку байт — в AH=55h, результат чтения RO по адресу-псевдониму RO+2. Логика работы контроллера шины вместе со всеми буферами делает обращение к любой ячейке памяти или порту инвариантным к способу программной адресации — что закажешь, то и получишь, но требуется учитывать особенности периферийных устройств, у которых в адресации портов нередко встречаются псевдонимы. Адреса-псевдонимы встречаются и в пространстве памяти (например, копии образов BIOS под границей 1<sup>го</sup> и 16-го мегабайтами памяти в «классических» PC/AT).

## 6.1.2. Прямой доступ к памяти — DMA

*Прямой доступ к памяти* позволяет абоненту шины организовывать обмен данными между своим регистром и памятью под управлением контроллера DMA, минуя центральный процессор. До выполнения обмена канал DMA должен быть инициализирован — задан начальный адрес и размер пересылаемого блока памяти, направление и режим обмена. После инициализации канала обмен выполняется по инициативе ПУ.

Для интерфейса ПУ каждый канал DMA представляется парой сигналов: запрос обмена — DRQ<sub>x</sub> и подтверждение обмена — DACK<sub>x</sub>#, где x — номер используемого канала. На рис. 6.3 приведена диаграмма стандартного цикла передачи байта (для 8-битного канала) или слова (для 16-битного) от ПУ в память по каналу DMA. Цикл передачи блока байтов или слов в память будет выглядеть следующим образом.

1. По сигналу DRQ<sub>x</sub> контроллер DMA запрашивает управление шиной и дожидается его предоставления процессором (и другими контроллерами шины).
2. Контроллер выставляет адрес ячейки памяти и формирует в одном цикле шины сигналы IOR#, DACK<sub>x</sub>\* и MEMW#. Адрес на всех линиях, включая LA[23:17], действителен во время всего цикла (защелкивание не требуется), VALE=H в течение всего цикла. Сигнал DACK<sub>x</sub># указывает на то, что операция выполняется для канала x, а IOR# — на направление в канал (для пересылки из памяти в канал использовался бы сигнал IOW#). Чтобы по сигналу IOR# не было ложного чтения (по IOW# — ложной записи) порта, адрес которого совпадает с адресом памяти, присутствующим в цикле DMA, контроллер высоким уровнем сигнала

AEN запрещает портам дешифрацию адреса. Байт, считанный из ПУ, в том же цикле шины записывается в ячейку памяти.

3. Контроллер модифицирует счетчик адреса и повторяет шаги 1-2 для каждого следующего сигнала DRQx, пока не будет исчерпан счетчик циклов. В последнем цикле обмена контроллер формирует общий сигнал окончания TC (Terminate Count), который может быть использован устройством для формирования сигнала аппаратного прерывания.

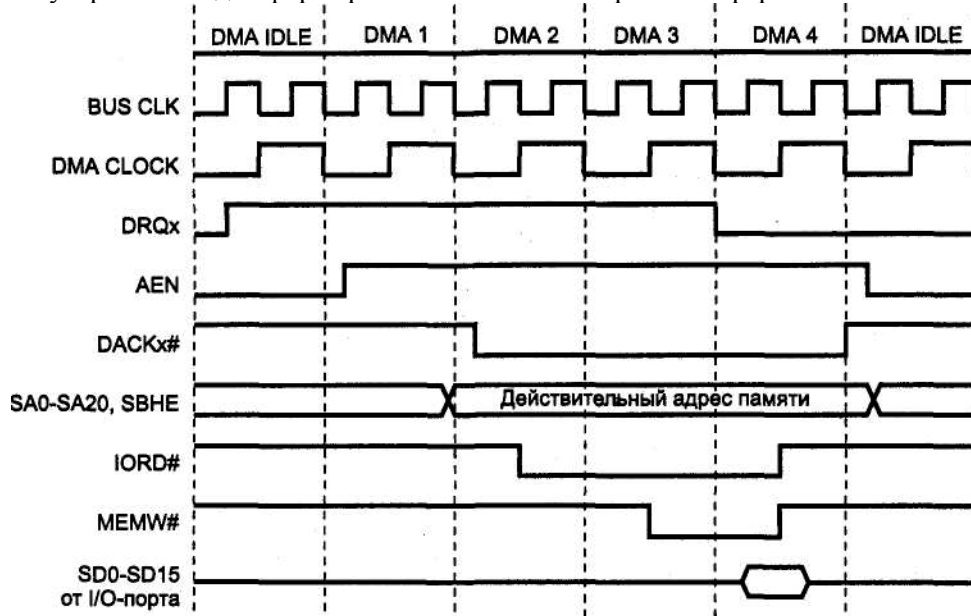


Рис. 6.3. Цикл обмена DMA

Диаграмма обратной пересылки (из памяти в ПУ) отличается только тем, что на месте сигнала IOR# будет сигнал MEMR#, а на месте MEMW\* — сигнал IOW#. Направление обмена и параметры режима задаются программированием контроллера DMA.

Цикл DMA, и так довольно длинный, может быть растянут устройством с помощью сигнала IOCHRDY (но сократить его сигналом OWS# невозможно). Запрашивать удлинение цикла может только адресованная память — подразумевается, что когда ПУ выставляет запрос DRQx, оно уже должно быть готово к обмену.

На 16-битной шине ISA доступно 7 каналов DMA: четыре 8-битных (номера 0-3) и три 16-битных (5-7), подключенные к первичному и вторичному контроллерам соответственно. Канал 4 используется для каскадирования (соединения контроллеров). На 8-битном слоте доступны только четыре 8-битных канала (в XT только 3 — канал 0 требовался для регенерации памяти).

Кроме приведенного на рисунке режима одиночной передачи возможны и иные, описанные в п. 12.4.

Напомним, что по 8-битным каналам DMA за один сеанс настройки контроллера можно передавать не более 64К *байт* данных, начинающихся с любого адреса (но не пересекая границ страниц, см. п. 12.3.2 и 12.5). По 16-битным каналам за сеанс можно передавать не более 64К *слов* данных, начинающихся с четного адреса, и границы страниц иные. Используя DMA в режимах, отличных от одиночного, длительность непрерывной передачи не должна превышать 15 мкс (для обеспечения регенерации памяти).

### 6.1.3. Прямое управление шиной

В случае *прямого управления шиной* (bus mastering) инициатором обмена становится контроллер какого-либо устройства или интерфейса, но не процессор и не канал DMA. Прямое управление позволяет контроллеру, не отвлекая центральный процессор, выполнять обмен данными с высокой производительностью и, возможно, по более сложным правилам и без ограничений, присущих каналам DMA (невозможность пересечения границы страницы). Контроллер, как правило, обеспечивает обмен данными между системным ОЗУ и своим периферийным устройством или интерфейсом. В принципе он может общаться с памятью или портами другого абонента шины, но для упрощения организации всех информационных связей центральным «перевалочным пунктом» все-таки является системное ОЗУ. Получить право на управление шиной ISA может только контроллер, подключенный к 16-битному слоту. Для арбитража запросов на управление шиной от нескольких источников — центрального процессора, контроллера DMA и контроллеров-абонентов шины используется второй контроллер DMA (каналы 5-7). Устройство, желающее получить управление, выставляет запрос DRQ<sub>x</sub> (x=5...7), по которому контроллер DMA запрашивает управление шиной у центрального процессора. Получив подтверждение от процессора, контроллер устанавливает сигнал AEN (для блокировки дешифрации адреса портов) и формирует сигнал DACK<sub>x</sub>. Получив этот сигнал, устройство устанавливает низкий уровень на линии MASTER\*, по которому контроллер шины DMA снимает сигнал AEN и освобождает линии командных сигналов. Таким образом, управление шиной переходит к данному устройству до тех пор, пока оно не снимет запрос DRQ<sub>x</sub>. В режиме прямого управления шиной ISA устройство становится полновластным и бесконтрольным хозяином как самой шины, так и системы в целом (через мост шины). Если шина захватывается более чем на 15 мкс, контроллер устройства должен заботиться о регенерации памяти (см. ниже). Устройства (и память), к которым обращается задатчик шины, могут потребовать введения тактов ожидания сигналом IOCHRDY, и это требование должно удовлетворяться.

### 6.1.4. Регенерация памяти

Динамическая память, применяемая в компьютере в качестве системного ОЗУ, а также, возможно, находящаяся на картах расширения, требует регенерации — периодического обновления (refresh) всех строк матрицы. На системной плате всегда имеется контроллер регенерации памяти, в задачу которого входит регуляр-

ный перебор строк памяти с формированием специального цикла регенерации. Каждые 15 мкс контроллер регенерации формирует цикл регенерации (сигнал REFRESH\*), диаграмма которого аналогична циклу DMA, но вместо чтения данных выполняется регенерация. В этом цикле считывания данных из микросхем памяти не производится, но стробы строк формируются (подробнее о методах регенерации см. в книге «Аппаратные средства IBM PC. Энциклопедия»). На шине ISA контроллер регенерации работает как нулевой канал DMA (в старых машинах он непосредственно и использовался). Каждые 15 мкс контроллер регенерации формирует сигнал REFRESH\*, во время действия которого устанавливается AEN=H и BALE=H, формируется сигнал MEMR#, а на линиях SA[7:0] устанавливается адрес текущей регенерируемой строки. Диаграмма цикла регенерации аналогична циклу DMA, но сигнал REFRESH\* указывает на то, что вместо чтения данных выполняется регенерация.

Если какое-либо устройство захватывает управление шиной более чем на 15 мкс, оно должно позаботиться о регенерации памяти. Для этого раз в 15 мкс оно должно вырабатывать сигнал REFRESH\*, «отпуская» все адресные и управляющие сигналы. При этом контроллер шины, расположенный на системной плате и отвечающий за регенерацию памяти, сформирует очередной адрес на линиях SA[7:0] и сигнал MEMR\*, а также установит AEN=0 и BALE=1 на все время цикла регенерации. Если устройство, надолго захватившее шину, сигнал регенерации не формирует, то запросы регенерации накапливаются в контроллере регенерации, и, как только управление будет возвращено, эти запросы будут немедленно удовлетворены. Однако число безнаказанно пропущенных запросов ограничено свойствами динамической памяти, и очередь запросов в контроллере регенерации может ограничиваться, например, четырьмя.

### 6.1.5. Прерывания

На шине ISA имеются линии запросов маскируемых и немаскируемых аппаратных прерываний. Линии запросов *маскируемых прерываний* IRQ2-IRQ7 поступают на входы первичного контроллера прерываний, IRQ9-IRQ15 — на входы вторичного. Контакт B4 (IRQ2/9) запрос IRQ2 вырабатывал только у машин PX/XT. На всех современных машинах (класса AT) он является запросом IRQ9, хотя на многих адаптерах (всех 8-битных) он обозначается как IRQ2. Приоритеты у запросов убывают по порядку IRQ9...IRQ15 и далее IRQ3...IRQ7. Запросы от конкретных линий могут быть замаскированы записью в регистры контроллера (см. п. 12.4), общий запрет/разрешение осуществляется манипулированием флагом разрешения прерываний (IF) процессора. Устройство может использовать одну или несколько линий запроса прерывания. На используемой линии запроса устройство в покое должно формировать низкий уровень сигнала, а при возникновении условия прерывания устанавливать на нем высокий уровень запроса. Неиспользуемые линии должны быть электрически отключены от шины или же их выходные формирователи должны переводиться в третье состояние. Переход из низкого в высокий уровень является сигналом для контроллера прерываний на формирование запроса прерывания к процессору. Устройство должно удерживать высокий уро-

вень запроса до тех пор, пока к нему не обратится программа-обработчик прерывания, что будет означать не только обнаружение, но и правильную идентификацию источника запроса прерывания. Если запрос снят преждевременно, идентификация будет некорректной. Детально механизм обслуживания прерываний рассмотрен в п. 12.4.

Способ подачи сигнала прерывания, принятый в ISA, — чувствительность к уровню, причем к высокому, — имеет меньшую помехозащищенность, чем срабатывание по отрицательному перепаду, и отрезает путь к нормальному разделению (совместному использованию) линий запросов.

Линия ЮСНК# позволяет вызывать *немаскируемое прерывание* (NMI), на которое процессор реагирует вне зависимости от каких-либо флагов. Это прерывание принято использовать для сообщения о серьезных ошибках, требующих реакции системы, но не для регулярной работы. Вызов NMI от данной линии разрешается установкой бита 3 (EIC) системного порта 061h, а признаком того, что прерывание NMI вызвано сигналом ЮСНК#, является единичное значение бита 6 (IOСНК) того же порта.

### 6.1.6. Шина PC/104

Шина PC/104, предназначенная для построения относительно несложных встраиваемых контроллеров, логически эквивалентна ISA. В ее названии 104 — число контактов коннектора, на который выводятся сигналы шины ISA. От ISA шина PC/104 отличается только типом коннектора и нагрузочными характеристиками линий: поскольку протяженность линий значительно сократилась, сигнальные цепи могут быть слаботочными. Формирователи сигналов MEMCS16#, IOCS16\*, MASTER\* и OWS# должны обеспечивать вытекающий ток до 20 мА, для остальных достаточно 4 мА.

Вид платы PC/104 приведен на рис. 6.4, назначение контактов дано в табл. 6.5 и 6.6. Плата компьютера с шиной PC/104 имеет *разъемы-розетки*. Плата расширения имеет *вилку* PC/104, которая вставляется в плату контроллера. Кроме вилки на плате расширения может присутствовать и розетка PC/104 (коннектор двусторонний), так что можно собрать «бутерброд» из нескольких плат (рис. 6.4, а). Для предотвращения неправильного соединения у розеток в позициях В10 и С19 нет ни контактов, ни отверстий; у вилок в этих позициях штырек отсутствует. Если плат более трех, то сверху «бутерброда» устанавливают терминатор. Для фиксации плат стандартизовано расположение крепежных отверстий. Платы скрепляются четырьмя несущими стоечками высотой 0,6" (или длинными винтами с втулками). Отметим особенности коннекторов^ ! — коннектор шины ISA-8, J2 — его расширение до ISA-16; эти коннекторы обычно имеют дюймовый шаг контактов (2,54 мм), но могут встречаться и метрические, с шагом 2,5 мм (они взаимно несовместимы!). Обратим внимание и на специфическую нумерацию рядов контактов В, А, С, D и номеров контактов (у J2 нумерация начинается с нуля, нулевые контакты не используются).



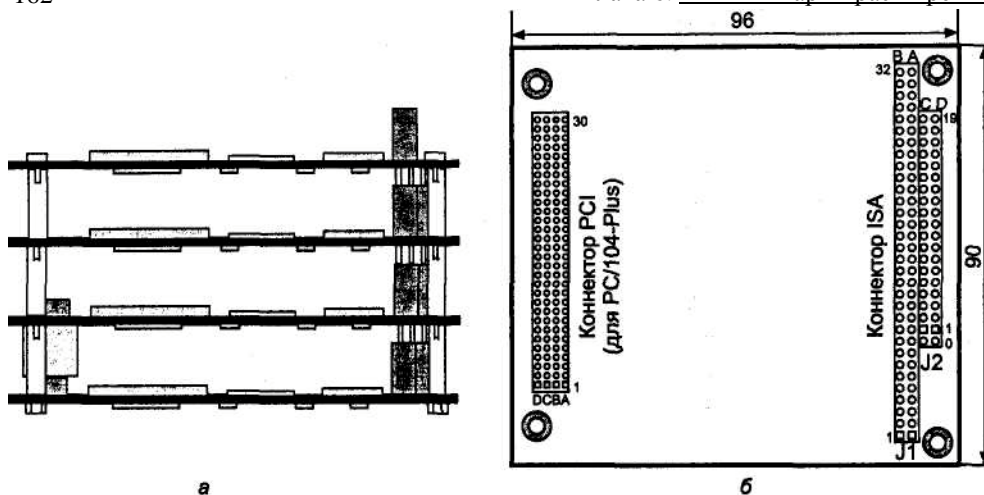


Рис. 6.4. Компьютер с шиной PC/104: а — стопка плат, б — расположение системных коннекторов

С широким использованием процессоров Pentium и следующих моделей в модуль ввели еще и шину PCI, так появился стандарт PC/104-Plus (именно такая карта показана на рис. 6.4). Трехрядный коннектор PCI имеет метрический шаг контактов 2 мм. Платы микрокомпьютеров с шинами PC/104 могут иметь и краевые разъемы ISA, PCI или комбинированный; с их помощью контроллеры подключаются к пассивной кросс-плате, в которую устанавливаются карты расширения (ISA, PCI) обычного (PC) конструктива.

Таблица 6.5. Разъем J1/P1 шины PC-104

РЯДА	№	Ряд В
ЮСНК#	1	GND
SD7	2	RESET
SD6	3	+5 В
SD5	4	IRQ2/9
SD4	5	-5 В
SD3	6	DRQ2
SD2	7	-12В
SD1	8	OVS#
SD0	9	+12В
IOCHDRY	10	Ключ
AEN	11	SMEMW#
SA19	12	SMEMR#
SA18	13	IOWR#
SA17	14	IORD#
SA16	15	DACK3#

Ряда	№	Ряд В
SA15	16	DRQ3
SAM	17	DACK1*
SA13	18	DRQ1
SA12	19	REFRESH*
SA11	20	BCLK
SA10	21	IRQ7
SA9	22	IRQ6
SA8	23	IRQ5
SA7	24	IRQ4
SA6	25	IRQ3
SA5	26	DACK2*
SA4	27	TC
SA3	28	BALE
SA2	29	+5 В
SA1	30	OSC
SA0	31	GND
GND	32	GND

Таблица 6.6. Разъем J2/P2 шины PC-104

РядС	№	РядО
GND	0	GND
SBHE	1	MEMCS16#
LA23	2	IOCS16*
LA22	3	IRQ10
LA21	4	IRQ11
LA20	5	IRQ12
LA19	6	IRQ15
LA18	7	IRQ14
LA17	8	DACKO*
MEMR#	9	DRQ0
MEMW#	10	DACK5*
SD8	11	DRQ5
SD9	12	DACK6*
SD10	13	DRQ6
SD11	14	DACK7*
SD12	15	DRQ7
SD13	16	+5 В
SD14	17	MASTER#
SD15	18	GND
Ключ	19	GND

### 6.1.7. Конфигурирование интерфейсных карт ISA и EISA

Как было указано выше, всем устройствам-абонентам шин ISA и EISA должны назначаться свои системные ресурсы — области адресов в пространствах памяти и ввода-вывода, линии запросов прерываний и каналы прямого доступа к памяти, причем устройства не должны конфликтовать по ресурсам. Иными словами, все устройства, подключенные к шине, требуется должным образом *сконфигурировать*. Под этим подразумевается бесконфликтная настройка их дешифраторов адресов и коммутация сигналов запросов прерываний и пар сигналов для работы с каналами DMA. Кроме того, выбранные аппаратные настройки должны быть сообщены программному обеспечению, непосредственно взаимодействующему с устройствами через порты, ячейки памяти, каналы DMA и контроллер прерываний.

Задача конфигурирования осложняется из-за отсутствия общего механизма автоматической передачи установленных параметров прикладному и системному ПО. Конфигурирование старых карт расширения выполняется переключением джамперов, затем установленные параметры заносятся в конфигурационные файлы.

Позже на картах ISA стали применять микросхемы энергонезависимой памяти (как правило, EEPROM), хранящей настройки. С такими картами поставляются утилиты настройки, позволяющие в диалоговом режиме задать требуемые параметры. Отсюда их названия: *программно конфигурируемые* (Software Configured), или безджамперные (jumperless). В общем, это позволило облегчить конфигурирование — для смены настроек не нужно вынимать карту и переставлять джамперы. Правда, пользователю все равно приходится вникать в распределение системных ресурсов. Однако работа конфигурационных утилит может осложняться (и блокироваться) соседними «недружественными» картами.

Идеальными условиями для программного конфигурирования, и тем более автоматического (без вмешательства пользователя), является изоляция карты от всех остальных (на время конфигурирования). Тогда ПО конфигурирования сможет вести с картой диалог, на который не повлияет присутствие других устройств. Для автоматического конфигурирования необходимо также обеспечить единый метод двустороннего обмена конфигурационной информацией между картой и конфигурационным ПО. Возможность изоляции карт при конфигурировании заложена в шины MCA, PCI и EISA, но в ISA такой возможности нет. В шине EISA можно выборочно управлять сигналом AEN (разрешающим дешифрацию адресов портов ввода-вывода) для каждого слота, причем эта возможность сохраняется и для карт ISA, установленных в слот EISA. В машинах с EISA имеется специальная энергонезависимая память конфигурирования слотов, с которой взаимодействует утилита конфигурирования EQU (EISA Configuration Utility); эта память и утилита могут использоваться и при установке карт ISA в машину EISA. Для шины ISA система автоматического конфигурирования — ISA PnP — была разработана лишь спустя десяток лет после начала массового выпуска компьютеров и карт расширения. Также были расширены функции BIOS — появилась спецификация PnP BIOS. Полная поддержка автоматического конфигурирования карт ISA требует

наличия PnP BIOS, карт и/или модулей ISA PnP на системной плате, а также ОС с поддержкой PnP или же специализированного ПО.

### 6.1.8. Спецификация Plug and Play для шины ISA

Аппаратно-программную спецификацию «Plug and Play ISA Specification» выпустили компании Intel и Microsoft в 1994 г. Она обеспечивает решение задач изоляции карт ISA, программного распределения системных ресурсов, конфигурирования и передачи параметров операционной системе и прикладному ПО. Вышеперечисленные задачи решаются для карт PnP, которые могут работать и в окружении так называемых *традиционных карт (Legacy Cards)*. Поскольку описание программной части этой спецификации достаточно объемно и выходит за рамки данной книги, рассмотрим принципы реализации PnP в основном с точки зрения аппаратных средств. Конфигурирование в системе PnP состоит из следующих шагов.

1. Производится изоляция одной карты от всех остальных.
2. Карте назначается номер CSN (Card Select Number — селективный номер карты), фигурально выражаясь, «присваивается ручка» (Assign a handle), за которую ее можно «ухватить» дальнейшим командам PnP.
3. С карты считываются данные о сконфигурированных и поддерживаемых ресурсах. Эти шаги повторяются для всех карт, после чего выполняются завершающие шаги.
4. Производится распределение (арбитраж) системных ресурсов, выделяемых каждой карте. Каждая карта конфигурируется согласно выбранному распределению ресурсов и активируется (переводится в рабочий режим).

Все шаги конфигурирования выполняет процедура POST (если BIOS имеет поддержку PnP) или операционная система при загрузке. PnP BIOS может ограничиться конфигурированием и активацией только устройств, участвующих в загрузке, оставляя конфигурирование и активацию дополнительных устройств операционной системе. BIOS без поддержки PnP может использовать необходимые для загрузки устройства, сконфигурированные с параметрами по умолчанию, а изоляцией карт, сбором информации и конфигурированием займется операционная система при загрузке. Вариантов много, но все они опираются на единые методы взаимодействия с картами ISA PnP. Конфигурирование выполняется в специальном состоянии плат, в которое их всех можно программно перевести с помощью специального ключа инициализации, защищающего конфигурационную информацию от случайного разрушения.

Для конфигурирования карт PnP необходимо всего три 8-битных *системных порта* (табл. 6.7), с которыми процессор может общаться, применяя инструкции ввода-вывода с однобайтной передачей данных. Карты PnP должны использовать 12-битное декодирование адреса ввода-вывода, а не 10-битное, как это принято в традиционных картах ISA.

Таблица 6.7. Системные порты ISA PnP

Имя порта	Адрес	Тип доступа
ADDRESS	0279П ( <i>Printer status port</i> ) 0A79h ( <i>Printer</i>	Только запись
WRITE_DATA	<i>status port</i> + OS00h) Перемещаемый в	Только запись
READ_DATA	диапазоне 0203h-03FFh	Только чтение

Порт ADDRESS используется для адресации регистров PnP — в него записывают индекс требуемого регистра (см. ниже пункт «Конфигурирование карт») перед обращением к портам WRITE\_DATA и READ\_DATA. Этот же порт используется и для записи последовательности кодов ключа инициализации. Выбор адреса для него обусловлен тем, что ни одна разумно сделанная карта расширения не будет пытаться задействовать для записи адрес регистра состояния стандартного LPT-порта.

Порты WRITE\_DATA и READ\_DATA используются для обмена данными с регистрами PnP. Адрес порта WRITE\_DATA традиционными картами с 10-битным декодированием будет восприниматься как тот же адрес, что и у предыдущего порта, так что конфликт опять-таки исключен. Перемещаемому адресу порта READ\_DATA программное обеспечение PnP во время исполнения протокола изоляции может легко найти бесконфликтное положение. Адрес этого порта сообщается всем картам записью в их управляющий регистр PnP.

Вышеперечисленные Три порта используются только для конфигурирования и управления картами PnP. Для взаимодействия прикладных программ с функциональными устройствами карты используются иные ресурсы, выделенные карте (порты, области памяти, прерывания и каналы DMA). По включению питания или аппаратному сбросу карты настраиваются на стандартную рабочую конфигурацию, принятую для них по умолчанию (она может храниться в энергонезависимой Памяти или задаваться джамперами). Это обязательно, по крайней мере для устройств, участвующих в загрузке, — ввод, вывод (экран) и собственно загрузочное устройство. Остальные устройства могут быть и логически отключенными от шины, до тех пор пока они не будут сконфигурированы программными средствами PnP. Карта PnP должна сообщать обо всех используемых ею ресурсах и, по возможности, предлагать альтернативные конфигурирования. Она обязана подчиняться конфигурационным командам PnP, включая команду деактивации (логического отключения от шины); при невозможности принять указанную конфигурацию карта (или ее логическое устройство) должна отключаться. Строгое выполнение данных требований всеми картами делает возможным работу системы PnP на неприспособленной для этого шине ISA. Непременным условием работоспособности является и уникальность идентификаторов карт (см. ниже).-

В плане PnP каждая карта может находиться в одном из четырех состояний.

- ◆ *Wait for key* (ожидание ключа) — состояние нормального функционирования (или отключения) логических устройств. В это состояние устройство входит при включении питания, по аппаратному сбросу и по завершению его конфигурирования системой PnP. До подачи ключа в этом состоянии конфигурация карты программными обращениями изменена быть не может.

- ◆ *Sleep* («спячка») — состояние, в котором карта ожидает пробуждающей команды *Wake* [CSN], переводящей ее либо в состояние изоляции, либо в состояние конфигурирования. При нулевом параметре CSN в команде все карты с неназначенным номером CSN переводятся в состояние изоляции. При ненулевом параметре CSN в команде карта с совпадающим номером CSN переводится в состояние конфигурирования.
- ◆ *Isolation*(изоляция) — карта отвечает только на чтение регистра *Serial Isolation* для реализации одноименного протокола (см. далее), с помощью которого ей назначается уникальный номер CSN. Остальные команды PnP ей недоступны.
- ◆ *Config* (конфигурирование) — состояние, в котором карта отвечает на все обращения к регистрам PnP, позволяя считывать и изменять ее конфигурацию. В этом состоянии может находиться лишь одна карта. По окончании конфигурирования карту переводят в режим ожидания ключа, защищая ее конфигурацию.

### Протокол изоляции

Для изоляции карт друг от друга имеется специальный режим работы, в который логика PnP переводится с помощью *ключа инициализации (Initiation key)*. Ключ представляет собой predetermined последовательность записей в порт ADDRESS 5 (279h). Аппаратная логика карты, проверяющая ключ, основана на сдвиговом регистре с обратными связями *LFSR (Linear Feedback Shift Register)*, схема которого приведена на рис. 6.5. Во время проверки ключа на вход C1 подается уровень логического нуля, а на вход C2 — стробы записи в порт ADDRESS. Логика, не показанная на рисунке, сравнивает код в сдвиговом регистре с текущей записью и при несовпадении сбрасывает регистр LFSR в исходное состояние (код 6Ah). В это же состояние регистр может быть переведен двумя последовательными записями нулей в порт ADDRESS. Сдвиг в регистре происходит при каждой записи в порт ADDRESS. Если ключ (последовательность из 32 записей требуемых байт) из исходного состояния LFSR будет приложен верно, то после последней записи логика карты перейдет в режим конфигурирования (это еще не состояние *config*) и подготовится к обработке протокола изоляции. Точная последовательность байт ключа в hex-формате выглядит следующим образом:

6A, B5, DA, ED, F6, 7D, BE, DF, 6F, 37, IB, 0D, 86, C3, 61, B0, 58, 2C, 16, 8B, 45, A2, D1, E8, 74, 3A, 9D, CE, E7, 73, 39

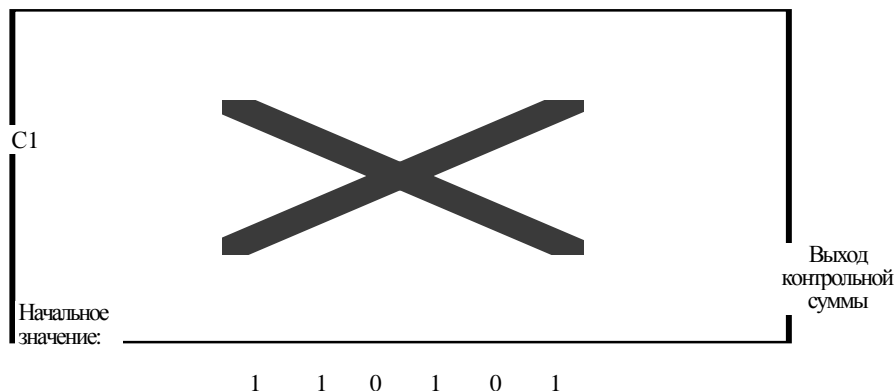


Рис. 6.5. Сдвиговый регистр LFSR карты PnP

Протокол изоляции основан на уникальном *последовательном идентификаторе* (*Serial Identifier*), хранящемся в памяти каждой карты PnP. Этот идентификатор представляет собой ненулевое 72-битное число, состоящее из двух 32-битных полей и 8-битного контрольного кода, вычисляемого с помощью того же регистра LFSR. Первое 32-битное поле представляет собой идентификатор производителя. Второе поле назначается производителем и уникально для каждого экземпляра всех выпускаемых им карт. Здесь может присутствовать серийный номер, а для адаптера Ethernet это может быть и частью MAC-адреса. Принцип построения последовательного идентификатора гарантирует, что в одной системе не могут встретиться две карты с совпадающими идентификаторами. Однако случалось, что незадачливые производители тиражировали (свои ли?) устройства, копируя всю «начинку», включая и серийные номера. Доступ к идентификатору осуществляется последовательно, начиная с бита 0 нулевого байта идентификатора производителя и заканчивая битом 7 контрольной суммы. Во время передачи идентификатора на вход C1 схемы LFSR поступают текущие биты идентификатора, а на вход C2 подаются стробы чтения регистра Serial Isolation (см. ниже). В тактах передачи контрольной суммы ее биты берутся с выхода сдвигового регистра.

Протокол изоляции может быть программно инициирован в любой момент времени посылкой корректного ключа инициализации, переводящего все карты в конфигурационный режим. В этом режиме каждая карта ожидает 72 пары операций чтения порта READ\_DATA. Ответ каждой карты на эти операции определяется значением очередного бита ее последовательного идентификатора.

Если текущий бит идентификатора карты имеет единичное значение, то ее буфер шины данных в первом чтении пары выводит на шину данных значение 55h. Если текущий бит нулевой, то буфер работает на *чтение* шины данных и логика карты анализирует ответ других карт — проверяет наличие комбинации «01» в битах D[1:0] (младшие биты числа 55h). В следующем цикле чтения пары карта с единичным битом выводит число AAh, а карта с нулевым текущим битом проверяет наличие комбинации «10».

Если карта, просматривающая вывод данных другими картами, обнаружила корректные коды в обоих циклах чтения пары, она в данной итерации изоляции исключается.

Если карта в текущей паре управляла шиной или карта читала шину, но не обнаружила корректных активных ответов других карт, она сдвигает идентификатор на один бит и готовится к приему следующей пары циклов чтения.

Эта последовательность выполняется для всех 72 бит идентификатора. В конце процесса останется лишь одна карта. Записью в управляющий регистр PnP (индекс 06) ей назначается селективный номер CSN, по которому она будет использоваться в дальнейших операциях. Карта с назначенным номером CSN в следующих итерациях протокола изоляции не участвует (на пары чтений не отвечает).

Во время протокола изоляции карты не имеют права удлинять шинные циклы с помощью сигнала IOCHRDY, поскольку это привело бы к неопределенности результатов наблюдения за «соседями». В других режимах этот сигнал может быть использован без особых ограничений.

Программа конфигурирования проверяет данные, возвращаемые во время всех пар циклов чтения, и побитно собирает прочитанный идентификатор. Если в паре приняты байты 55h и AAh, то соответствующий бит считается единичным, в других случаях он считается нулевым. При приеме идентификатора программа подсчитывает контрольную сумму и сравнивает ее с принятой. Несовпадение контрольной суммы или отсутствие среди принятых байт 55h и AAh указывает на то, что выбранный адрес порта READ\_DATA конфликтует с каким-либо устройством. Тогда программа пробует произвести итерацию, переместив адрес порта READ\_DATA в допустимом диапазоне адресов. Если при переборе нескольких возможных адресов не удастся считать корректный идентификатор, то принимается решение об отсутствии карт PnP в системе (вообще или с неназначенными номерами CSN). Программа должна обеспечивать задержку 1 мс после подачи ключа перед первой парой чтений и 250 мкс между парами чтений. Это дает карте время для доступа к информации, которая может храниться и в медленных устройствах энергонезависимой памяти.

Итак, по завершении протокола изоляции программное обеспечение имеет список идентификаторов обнаруженных карт и присвоенных им селективных номеров, сообщенных и самим картам. Далее общение программы с каждой картой идет по ее селективному номеру CSN, фигурирующему в командах PnP. Нулевой CSN присваивается картам по программному или аппаратному сбросу и используется как широковещательный адрес.

### Конфигурирование карт

Конфигурирование карт выполняется обращениями к регистрам PnP. Обращения к регистрам PnP представляют собой операции записи или чтения портов ввода-вывода по адресам WRITE\_DATA или READ\_DATA соответственно. При этом для указания конкретного регистра PnP используется *индекс* — номер этого регистра, предварительно записанный в регистр ADDRESS.

Каждая карта имеет стандартный набор регистров PnP, причем часть из них относится к карте в целом, а часть — к каждому логическому устройству, входящему в карту. Архитектура PnP поддерживает концепцию многофункциональности, согласно которой каждая карта может включать в себя несколько логических устройств. В любой момент времени в индексном пространстве регистров PnP отображаются стандартные регистры управления картой (см. ниже) и регистры только одного логического устройства (рис. 6.6). Выбор логического устройства, с которым производится общение, осуществляется записью в регистр Logical Device Number, входящий в группу управляющих регистров карты.

Все логические устройства карт PnP должны обеспечивать, по крайней мере, минимальную функциональность:

- ◆ регистры ресурсов при чтении должны отражать фактические текущие настройки;
- ◆ бит активации при чтении должен отражать реальное состояние активности устройства на шине ISA;



- ♦ если программа пытается «навязать» карте конфигурацию, не поддерживаемую устройством, это устройство не должно активироваться и, соответственно, при чтении его флаг активации должен быть сброшен.

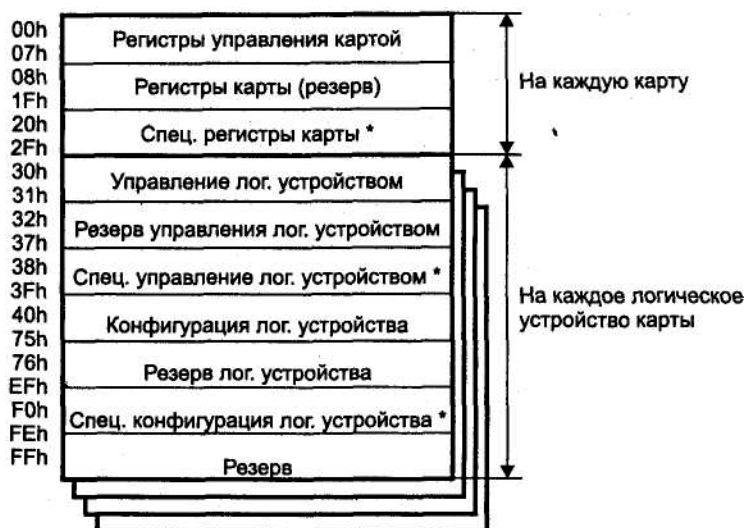


Рис. 6.6. Конфигурационные регистры PnP (\* — определяется разработчиком)

Для адресации к карте и ее логическим устройствам, а также для чтения конфигурационной информации используются стандартные регистры управления картой (табл. 6.8).

Таблица 6.8. Стандартные регистры управления картой PnP

Имя	Индекс и тип	Назначение
Set RD_DATA Port WO	00h,	Установка адреса порта для чтения. Биты [7:0] задают значение бит [9:2] адреса порта READ_DATA. Только для записи
Serial Isolation	01 h, RO	Чтение этого регистра в состоянии <i>Isolation</i> приводит к побитному анализу идентификатора
Config Control	02h,WO	Бит [2] — сброс CSN в 0. Бит [ 1 ] — возвращение в состояние <i>Wait for Key</i> . Бит [0] — сброс всех логических устройств и перевод конфигурационных регистров в состояние включения питания, но с сохранением номера CSN. Биты не запоминаются, так что необходимости в их программном сбросе нет
Wake[CSN]	03h,WO	Запись в этот регистр приведет карту, у которой номер CSN совпадает с записываемым байтом, к переходу из состояния <i>Sleep</i> в состояние <i>Config</i> (если данные не нулевые). Запись нулевого байта переводит все карты с неназначенным номером CSN в состояние <i>Isolation</i> . Указатель последовательно считываемых байт сбрасывается

Имя	Индекс и тип	Назначение
Resource Data	04h, RO	Чтение этого регистра возвращает очередной байт информации о ресурсах. Перед чтением должен опрашиваться регистр Status
Status	05h, RO	Регистр состояния. Единичное значение бита [0] указывает на возможность чтения очередного байта ресурсов
Card Select Number	06h, RW	Регистр хранения селективного номера CSN, обеспечивающего выбор конкретной карты командой Wake[CSN]. Обнуляется по включению питания и аппаратному сбросу
Logical Device Number	07h, RW	Выбор текущего логического устройства, к которому относятся все операции по обмену конфигурационной информацией, проверки диапазона адресов ввода-вывода и активации. Если карта имеет только одно устройство, регистр допускает только чтение и всегда имеет нулевое значение
Card Level Резерв	08h-1Fh 20h-2Fh	Зарезервированы на будущее Используются по усмотрению производителя
Card Level, Vendor Defined		

Конфигурирование карты начинается с команды WAKE [CSN] — записи байта CSN в регистр с индексом 3. Эта операция переводит карту с указанным номером CSN в состояние *Config* (конфигурирование), а остальные карты «засыпают» — переходят в состояние *Sleep*. Для конфигурируемой карты выполняются операции чтения ее конфигурационной информации (как карты в целом, так и логических устройств) и программирования используемых ресурсов. Программирование каждого логического устройства завершается установкой его бита активации, после чего логическое устройство активизируется на шине ISA (начнет реально использовать назначенные ресурсы). Программирование всей карты завершается переводом ее в состояние *Wait for key* (ожидание ключа). По окончании конфигурирования все карты PnP должны быть переведены в это состояние, и тогда их случайное реконфигурирование будет заблокировано 32-байтным ключом. Доступ к регистрам PnP через ключ возможен в любое время функционирования, однако запись в них должна производиться в полной уверенности о знании последствий. Возможно даже переназначение CSN «на ходу», но это требуется лишь в устройствах, допускающих «горячие» включения-выключения (что не приветствуется на шине ISA), док-станциях (*Docking Stations*) для подключения портативных компьютеров и системах управления энергопотреблением.

Стандартные регистры управления логическим устройством (табл. 6.9) используются для активации карт и проверки отсутствия конфликтов на шине ISA в выбранном диапазоне адресов ввода-вывода. Когда включен режим проверки конфликтов, на чтение по любому адресу установленного диапазона портов ввода-вывода логическое устройство отвечает байтом 55h или AAh в зависимости от состояния бита 0 регистра проверки. Естественно, что в рабочем режиме этот «автоответчик» должен быть отключен.

Таблица 6.9. Стандартные регистры управления логическим устройством PnP

Имя	Индекс и тип	Назначение
Activate	30h,RW	Регистр активации. Бит [0] единичным значением разрешает активность логического устройства на шине ISA. Биты [7:1] зарезервированы, при чтении должны возвращать нули. Перед активацией должна быть запрещена проверка диапазона адресов ввода-вывода
I/O Range Check	31 h, RW	Регистр проверки диапазона адресов ввода-вывода. Биты [7:2] зарезервированы, при чтении должны возвращать нули. Бит [1] — разрешение режима проверки. Бит [0] — управление диагностическим ответом: 0 — ответ AAh, 1 — ответ 55h
Logical Device Control Резерв	32h-37h	Зарезервированы на будущее
Logical Device Control Vendor Defined	38h-3Fh	Используются по усмотрению производителя

Оперативные данные конфигурирования доступны через регистры логических устройств. Каждое логическое устройство имеет собственные дескрипторы используемых системных ресурсов.

- ◆ Обычные 24-битные (4) или 32-битные (4) дескрипторы памяти. Для неиспользуемого дескриптора памяти его поля базового адреса и длины должны быть нулевыми. Одна карта не может одновременно задействовать обычные (24-битные) и 32-битные дескрипторы памяти.
- ◆ Дескрипторы областей портов ввода-вывода (8). Для неиспользуемого дескриптора портов ввода-вывода его поле базового адреса должно быть нулевым. Размер области адресов определяется в блоке данных, считанном из регистра Resource Data.
- ◆ Дескрипторы запросов прерываний (2). Неиспользуемый селектор запроса прерывания должен быть нулевым (поскольку нулевой номер запроса недопустим — занят системным таймером). Для линии IRQ2/9 шины ISA применяют номер 9.
- ◆ Дескрипторы каналов прямого доступа к памяти. Неиспользуемый дескриптор канала прямого доступа должен иметь значение 4 (этот канал недоступен, поскольку задействован для каскадирования контроллеров).

Назначение регистров дескрипторов и их положение в индексном пространстве PnP раскрывает табл. 6.10.

Таблица 6.10. Регистры дескрипторов системных ресурсов логических устройств PnP

Индекс и тип	Назначение
<b>40h-5Fh — обычные (24-битные) дескрипторы памяти</b>	
40h, RW	Дескриптор памяти 0: базовый адрес памяти, биты [23:16]
41h, RW	Дескриптор памяти 0: базовый адрес памяти, биты [15:8]

Индекс и тип	Назначение
42h, RW	<i>Дескриптор памяти 0</i> : управление. Бит [1] — режим обращения: 0 — 8 бит, 1 — 16 бит. Бит [0] (RO) — способ задания диапазона: 0 — следующее поле воспринимается как маска адреса, 1 — как старший адрес
43h, RW	<i>Дескриптор памяти 0</i> : маска или максимальный адрес области, биты [23:16]. Единичное значение бита в маске указывает, что соответствующий бит адреса участвует в дешифрации адреса области памяти (старшие биты); нулевое — бит адреса используется для адресации в пределах выделенной области
44h, RW	<i>Дескриптор памяти 0</i> : маска или максимальный адрес области, биты [15:8]
45h-47h	Заполнитель (зарезервировано)
48h-4Ch	<i>Дескриптор памяти Г</i> (аналогично предыдущему)
4Dh-4Fh	Заполнитель (зарезервировано)
50h-54h	<i>Дескриптор памяти 2</i> (аналогично предыдущему)
55h-57h	Заполнитель (зарезервировано)
58h-5Ch	<i>Дескриптор памяти 3</i> (аналогично предыдущему)
5Dh-5Fh	Заполнитель (зарезервировано)
<b>60h-6Fh — дескрипторы пространства ввода-вывода</b>	
60h, RW	<i>Дескриптор портов 0</i> : базовый адрес, биты [15:8]. Если логическое устройство использует только 10-битное декодирование адреса, биты [15:10] могут игнорироваться
61 h, RW	<i>Дескриптор портов 0</i> : базовый адрес, биты [7:0]
62h-63h, RW	<i>Дескриптор портов 1</i> (аналогично предыдущему)
64h-65h, RW	<i>Дескриптор портов 2</i> (аналогично предыдущему)
66h-67h, RW	<i>Дескриптор портов 3</i> (аналогично предыдущему)
68h-69h, RW	<i>Дескриптор портов 4</i> (аналогично предыдущему)
6Ah-6Bh, RW	<i>Дескриптор портов 5</i> (аналогично предыдущему)
6Ch-6Dh, RW	<i>Дескриптор портов 6</i> (аналогично предыдущему)
6Eh-6Fh, RW	<i>Дескриптор портов 7</i> (аналогично предыдущему)
<b>70h-73h — дескрипторы запросов прерываний</b>	
70h, RW	<i>Селектор запроса прерывания 0</i> . Биты [3:0] задают номер IRQ для <i>Interrupt 0</i>
71 h, RW	<i>Тип сигнала запроса прерывания 0</i> . Бит [1] — активный уровень: 1 — высокий, 0 — низкий. Бит [0] — тип: 1 — уровень, 0 — перепад. Если карта поддерживает только один тип сигнала, регистр может быть типа RO
72h, 73h RW	<i>Селектор и тип запроса прерывания 1</i> (аналогично предыдущему) <b>74h-75h — дескрипторы каналов прямого доступа</b>
74h, RW	<i>Селектор 0 канала DMA</i> . Биты [2:0] задают номер используемого канала (001 — DMA0....111 — DMA7)
75h, RW	<i>Селектор 1 канала DMA</i> (аналогично предыдущему)
<b>76h-A8h — 32-битные дескрипторы памяти</b>	
76h, RW	<i>32-битный дескриптор памяти 0</i> : базовый адрес памяти, биты [31:24]
77h, RW	<i>32-битный дескриптор памяти 0</i> : базовый адрес памяти, биты [23:16]
78h, RW	<i>32-битный дескриптор памяти 0</i> : базовый адрес памяти, биты [15:8]
79h, RW	<i>32-битный дескриптор памяти 0</i> : базовый адрес памяти, биты [7:0]

продолжение ^

Таблица 6.10 (продолжение)

Индекс и тип	Назначение
7Ah, RW	32-битный дескриптор памяти Отправление. Биты [7:3] — зарезервированы, при чтении должны возвращать нули. Биты [2:1] — управление доступом: 00 — 8-битная память, 01 — 16-битная память, 10 — зарезервировано, 11 — 32-битная память. Бит [0] (RO) — способ задания диапазона: 0 — следующее поле воспринимается как размер области, 1 — следующее поле воспринимается как старший адрес
7Bh, RW	32-битный дескриптор памяти 0: размер или старший адрес области, биты [31:24]
7Ch, RW	32-битный дескриптор памяти 0: размер или старший адрес области, биты [23:16]
7Dh, RW	32-битный дескриптор памяти 0: размер или старший адрес области, биты [15:8]
7Eh, RW	32-битный дескриптор памяти 0: размер или старший адрес области, биты [7:0]
7Fh	Заполнитель (зарезервировано)
80h-88h	32-битный дескриптор памяти 1
89h-8Fh	Заполнитель (зарезервировано)
90h-98h	32-битный дескриптор памяти 2
99h-9Fh	Заполнитель (зарезервировано)
A0h-A8h	32-битный дескриптор памяти 3

Дескрипторы требуемых ресурсов (данные о возможных конфигурациях логических устройств) могут быть считаны последовательно байт за байтом из регистра Resource Data и использованы для конфигурирования устройств, которое выполняется через регистры, перечисленные в табл. 6.10. Считываться будут данные из карты, находящейся в состоянии *config*. Если регистр считывается сразу после «победы» карты в протоколе изоляции, считывание начинается с дескриптора версии PnP. Если считывание начинается для карты после ее «пробуждения» командой Wake [CSN], сначала будут считаны 8 байт уникального идентификатора, затем байт контрольного кода, который будет недействительным, поскольку генерируется аппаратно регистром LFSR во время побитного считывания идентификатора. Только после этого начнется считывание дескрипторов ресурсов. Порядок считывания дескрипторов существенен — именно в этом порядке должны программироваться регистры дескрипторов ресурсов карты PnP. Последовательность считывания дескрипторов для каждого логического устройства завершается признаком завершения области дескрипторов.

Считав все дескрипторы всех устройств, программа, выполняющая конфигурирование PnP, получает исчерпывающую информацию об устройствах и их потребностях. После этого она пытается найти бесконфликтную конфигурацию для всех устройств, с учетом потребностей установленных устройств, не относящихся к устройствам PnP. В соответствии с принятым планом она конфигурирует все устройства, а те, которым не удастся выделить ресурсы, отключает. Данные о принятых настройках передаются «заинтересованному» ПО программным способом.

## 6.2. Шина PCI

PCI (Peripheral Component Interconnect) local bus — шина соединения периферийных компонентов является основной шиной расширения современных компьютеров. Она разрабатывалась в расчете на Pentium, но хорошо сочеталась и с процессорами 486. Сейчас PCI является четко стандартизированной высокопроизводительной и надежной шиной расширения. Первая версия PCI 1.0 появилась в 1992 г. В PCI 2.0 (1993 г.) введена спецификация коннекторов и карт расширения. В версии 2.1 (1995 г.) введена частота 66 МГц. В настоящее время действует спецификация PCI 2.2 (декабрь 1998 г.), которая уточняет и разъясняет некоторые положения предшествующей версии 2.1. Данное описание основано на тексте стандарта «PCI Local Bus Specification. Revision 2.2» от 18.12.1998, опубликованного организацией PCI SIG (Special Interest Group).

Поначалу шина PCI вводилась как пристройка (mezzanine bus) к системам с основной шиной ISA, став позже центральной шиной: она соединяется с системной шиной процессора высокопроизводительным мостом («северным»), входящим в состав чипсета системной платы. Остальные шины расширения ввода-вывода (ISA/EISA или MCA), а также локальная ISA-подобная шина X-BUS и интерфейс LPC, к которым подключаются микросхемы системной платы (ROM BIOS, контроллеры прерываний, клавиатуры, DMA, портов СОМ и LPT, НГМД и прочие «мелочи»), подключаются к шине PCI через «южный» мост. В современных системных платах с хэбовой архитектурой шину PCI Отодвинули на периферию, не ущемляя ее в мощности канала связи с процессором и памятью, но и не нагружая транзитным трафиком устройств других шин.

Шина является синхронной — фиксация всех сигналов выполняется по положительному перепаду (срронту) сигнала CLK. Номинальной частотой синхронизации считается 33 МГц, при необходимости частота может быть понижена (на машинах с процессором 486 использовали частоты 20-33 МГц). Во многих случаях частоту успешно разгоняют и до 41,5 МГц (половина типовой частоты системной шины 83 МГц). Начиная с версии 2.1 допускается повышение частоты до 66 МГц при согласии всех устройств на шине.

Номинальная разрядность шины данных — 32 бита, спецификация определяет и расширение разрядности до 64 бит. При частоте шины 33 МГц теоретическая пропускная способность достигает 132 Мбайт/с для 32-битной шины и 264 Мбайт/с для 64-битной; при частоте синхронизации 66 МГц — 264 и 528 соответственно. Однако эти пиковые значения достигаются лишь во время передачи пакета, а из-за протокольных накладных расходов реальная средняя суммарная (для всех задат-чиков) пропускная способность шины оказывается ниже.

С устройствами PCI процессор может взаимодействовать командами обращения к памяти и портам ввода-вывода, адресованным к областям, выделенным каждому такому устройству при конфигурировании. Устройства могут вырабатывать запросы маскируемых и немаскируемых прерываний. Понятия каналов DMA для шины PCI нет, но агент шины может сам выступать в роли зада'тчика, поддерживая высокопроизводительный обмен с памятью (и не только), не занимая ресурсов центрального процессора. Таким образом, к примеру, может быть реализован обмен в режиме DMA с устройствами АТ А, подключенными к контролеру PCI

IDE (см. п. 9.2.1). Спецификация PCI требует от устройств способности перемещать все занимаемые ресурсы в пределах доступного пространства адресации. Это позволяет обеспечивать бесконфликтное распределение ресурсов для многих устройств (функций). Для управления устройствами рекомендуется вместо портов ввода-вывода по возможности использовать ячейки памяти. Одно и то же функциональное устройство может быть сконфигурировано по-разному, отображая свои регистры либо на пространство памяти, либо на пространство ввода-вывода. Драйвер может определить текущую настройку, прочитав содержимое регистра базового адреса устройства, — признаком пространства ввода-вывода будет единичное значение бита 0 (см. п. 6.2.12). Драйвер также может определить и номер запроса прерывания, который используется устройством.

### 6.2.1. Адресация устройств PCI

Для шины PCI принята иерархия понятий адресации: шина, устройство, функция. Эти понятия фигурируют только при обращении к регистрам конфигурационного пространства (см. п. 6.2.12). К этим регистрам обращаются на этапе конфигурирования — переучета обнаруженных устройств, выделения им непересекающихся ресурсов (областей памяти и пространства ввода-вывода) и назначения номеров аппаратных прерываний. При дальнейшей регулярной работе устройства будут отзываться на обращения по назначенным им адресам памяти и ввода-вывода, доведенным до сведения связанных с ними модулей ПО. Эти адреса принимаются с шины AD в начале каждой транзакции. Для доступа к конфигурационному пространству используются отдельные линии IDSEL.

*Устройством PCI* называется микросхема или карта расширения, подключенная к одной из шин PCI и использующая для идентификации выделенную ей линию IDSEL, принадлежащую этой шине. Устройство может быть многофункциональным, то есть состоять из множества (от 1 до 8) так называемых *функций*. Каждой функции отводится конфигурационное пространство в 256 байт (см. п. 6.2.12). Многофункциональные устройства должны отзываться только на конфигурационные циклы с номерами функций, для которых имеется конфигурационное пространство. При этом функция с номером 0 должна быть обязательно, номера остальных функций назначаются разработчиком устройства произвольно (в диапазоне 1-7). Простые (однофункциональные) устройства, в зависимости от реализации, могут отзываться либо на любой номер функции, либо только на номер функции 0.

*Шина PCI* — набор сигнальных линий (см. п. 6.2.2), непосредственно соединяющих интерфейсные выводы группы устройств (слотов, микросхем на системной плате). В системе может присутствовать несколько шин PCI, соединенных *мостами PCI* (см. п. 6.2.10). Мосты электрически отделяют интерфейсные сигналы одной шины от другой, соединяя их логически; главный мост соединяет главную шину с ядром системы (процессором и памятью). Каждая шина имеет свой *номер шины* (PCI bus number). Шины нумеруются последовательно; главная шина имеет нулевой номер.

С точки зрения конфигурирования, минимальной адресуемой единицей этой иерархии является функция; ее полный адрес состоит из трех частей: номера шины, номера устройства и номера функции. Короткая форма идентификации вида PCIO:1:2 (например, в сообщениях ОС Unix) означает функцию 2 устройства 1, подключенного к главной (0) шине PCI.

В шине PCI принята географическая адресация — номер устройства определяется местом его подключения. *Номер устройства* (device number или dev) определяется той линией шины AD, к которой подключена линия сигнала IDSEL данного слота: KADU -dev0(МОСТ),AD12-dev1,...AD31 -dev20. В соседних слотах PCI, как правило, задействуются соседние номера устройств; их нумерация определяется разработчиком системной платы (или пассивной кросс-платы в промышленных компьютерах). Часто для слотов используются убывающие номера устройств, начиная с 20. Группы соседних слотов могут подключаться к разным шинам; на каждой шине PCI нумерация устройств независима (могут быть и устройства с совпадающими номерами dev, но разными номерами шин). Устройства PCI, интегрированные в системную плату, используют ту же систему адресации. Их номера «запаяны намертво», в то время как адреса карт расширения можно изменять перестановкой их в разные слоты. Одна карта PCI может содержать только одно устройство шины, к которой она подключается, поскольку ей в слоте выделяется только одна линия IDSEL. Если на карте размещают несколько устройств (например, 4-портовая карта Ethernet), то на ней приходится устанавливать мост — тоже устройство PCI, к которому и обращаются по линии IDSEL, выделенной данной карте. Этот мост организует на карте дополнительную шину PCI, к которой можно подключить множество устройств.

С точки зрения обращения к пространствам памяти и ввода-вывода, географический адрес (номер шины и устройства) безразличен (не принимая во внимание разницу в производительности, связанную с подключением устройств к разным шинам PCI). Однако номер устройства определяет номер линии прерывания, которой может пользоваться устройство. Подробнее об этом см. в п. 6.2.6, здесь же отметим, что на одной шине устройства с номерами, отличающимися друг от друга на 4, будут использовать одну и ту же линию прерывания. Возможность развести их по разным линиям прерывания может появиться лишь, если они находятся на разных шинах (это зависит от системной платы).

Разобраться с нумерацией устройств и полученных ими линий прерываний на конкретной плате можно просто: устанавливать одну карту PCI поочередно в каждый из слотов (отключая питание) и смотреть на сообщения об обнаруженных устройствах PCI, выводимых на дисплей в конце теста POST. В этих сообщениях будут фигурировать и устройства PCI, установленные непосредственно на системной плате (и не отключенные параметрами CMOS Setup).

Но чтобы не было иллюзий простоты и прозрачности, отметим, что «особо умные» операционные системы (Windows) не довольствуются полученными назначениями номеров прерываний и изменяют их по своему усмотрению (что никак не может отразиться на разделяемости линий).

### 6.2.2. Протокол шины PCI

В каждой транзакции (обмене по шине) участвуют два устройства — инициатор (initiator) обмена, он же *ведущее* (master) устройство, и целевое (target) устройство (ЦУ), оно же *ведомое* (slave). Шина PCI все транзакции трактует как пакетные: каждая транзакция начинается фазой адреса, за которой может следовать одна или несколько фаз данных. Состав и назначение интерфейсных сигналов шины приведены в табл. 6.11.



**Таблица 11** Сигналы шины PCI  
**6.**

Сигнал	Назначение
AD[31:0]	Address/Data — мультиплексированная шина адреса/данных. В начале транзакции передается адрес, в последующих тактах — данные
C/BE[3:0]#	Command/Byte Enable — команда/разрешение обращения к байтам. Команда, определяющая тип очередного цикла шины, задается четырехбитным кодом в фазе адреса
FRAME*	Кадр. Введением сигнала отмечается начало транзакции (фаза адреса), снятие сигнала указывает на то, что последующий цикл передачи данных является последним в транзакции
DEVSEL*	Device Select — устройство выбрано (ответ ЦУ на адресованную к нему транзакцию)
IRDY*	Initiator Ready — готовность ведущего устройства к обмену данными
TRDY*	Target Ready — готовность ЦУ к обмену данными
STOP*	Запрос ЦУ к ведущему устройству на остановку текущей транзакции
LOCK*	Сигнал захвата шины для обеспечения целостного выполнения операции. Используется мостом, которому для выполнения одной операции требуется выполнить несколько транзакций PCI
REQ#	Request — запрос от ведущего устройства на захват шины
GNT#	Grant — предоставление ведущему устройству управления шиной
PAR	Parity — общий бит паритета для линий AD[31:0] и C/BE[3:0]#
PERR*	Parity Error — сигнал об ошибке паритета (для всех циклов, кроме специальных). Вырабатывается любым устройством, обнаружившим ошибку
PME#	Power Management Event — сигнал о событиях, вызывающих изменение режима потребления (дополнительный сигнал, введенный в PCI 2.2)
CLKRUN*	Clock running — шина работает на номинальной частоте синхронизации. Снятие сигнала означает замедление или остановку синхронизации с целью снижения потребления (для мобильных применений)
PRSNT[1,2]#	Present — индикаторы присутствия платы, кодирующие запрос потребляемой мощности. На карте расширения одна или две линии индикаторов соединяются с шиной GND, что воспринимается системной платой
RST#	Reset — сброс всех регистров в начальное состояние
IDSEL	Initialization Device Select — выбор устройства в циклах конфигурационного считывания и записи
SERR#	System Error — системная ошибка. Ошибка паритета адреса данных в специальном цикле или иная катастрофическая ошибка, обнаруженная устройством. Активируется любым устройством PCI и вызывает NMI
REQ64*	Request 64 bit — запрос на 64-битный обмен. Сигнал вводится 64-битным инициатором, по времени он совпадает с сигналом FRAME*. Во время окончания сброса (сигналом RST*) сигнализирует 64-битному устройству о том, что оно подключено к 64-битной шине. Если 64-битное устройство не обнаружит этого сигнала, оно должно переконфигурироваться на 32-битный режим, отключив буферные схемы старших байтов
ACK64*	Подтверждение 64-битного обмена. Сигнал вводится 64-битным ЦУ, опознавшим свой адрес, одновременно с DEVSEL*. Отсутствие этого подтверждения заставит инициатор выполнять обмен с 32-битной разрядностью
INTA#, INTB*, INTD*	Interrupt A, B, C, D — линии запросов прерывания, чувствительность к уровню, INTA#, INTD* активный уровень — низкий, что допускает разделяемость (совместное использование) линий
CLK	Clock — тактовая частота шины. Должна лежать в пределах 20—33 МГц, в PCI2.1 — до 66 МГц
M66EN	66MHz Enable — разрешение частоты синхронизации до 66 МГц

Сигнал	Назначение
SDONE	Snoop Done — сигнал завершенности цикла слежения для текущей транзакции. Низкий уровень указывает на незавершенность цикла слежения за когерентностью памяти и кэша. Необязательный сигнал, используется только устройствами шины с кэшируемой памятью
SBO#	Snoop Backoff — попадание текущего обращения к памяти абонента шины в модифицированную строку кэша. Необязательный сигнал, используется только абонентами шины с кэшируемой памятью при алгоритме обратной записи
TCK	Test Clock — синхронизация тестового интерфейса JTAG
TDI	Test Data Input — входные данные тестового интерфейса JTAG
TOO	Test Data Output — выходные данные тестового интерфейса JTAG
TMS	Test Mode Select — выбор режима для тестового интерфейса JTAG
TRST	Test Logic Reset — сброс тестовой логики

В каждый момент времени шиной может управлять только одно ведущее устройство, получившее на это право от арбитра. Каждое ведущее устройство имеет пару сигналов — REQ# для запроса на управление шиной и GNT\* для подтверждения предоставления управления шиной. Устройство может начинать транзакцию (устанавливать сигнал FRAME\*) только при активном полученном сигнале GNT\*. Снятие сигнала GNT\* не позволяет устройству начать следующую транзакцию, а при определенных условиях (см. ниже) заставляет прекратить начатую транзакцию. Арбитражем запросов на использование шины занимается специальный узел, входящий в чипсет системной платы. Схема приоритетов (фиксированный, циклический, комбинированный) определяется программированием арбитра.

Для адреса и данных используются общие мультиплексированные линии AD. Четыре мультиплексированные линии C/BE[3:0] обеспечивают кодирование команд в фазе адреса и разрешения байт в фазе данных. В начале транзакции ведущее устройство активизирует сигнал FRAME\*, по шине AD передает целевой адрес, а по линиям C/BE# — информацию о типе транзакции (команде). Адресованное ЦУ отзывается сигналом DEVSEL\*. Ведущее устройство указывает на свою готовность к обмену данными сигналом IRDY#, эта готовность может быть выставлена и раньше получения DEVSEL\*. Когда к обмену данными будет готово и ЦУ, оно установит сигнал TRDY\*. Данные по шине AD передаются только при одновременном наличии сигналов IRDY# и TRDY\*. С помощью этих сигналов ведущее устройство и ЦУ согласуют свои скорости, вводя такты ожидания. На рис. 6.7 приведена временная диаграмма обмена, в которой и ведущее устройство, и ЦУ вводят такты ожидания. Если бы они оба ввели сигналы готовности в конце фазы адреса и не снимали их до конца обмена, то в каждом такте после фазы адреса передавались бы по 32 бита данных, что обеспечило бы выход на предельную производительность обмена.

Количество фаз данных в пакете явно не указывается, но перед последней фазой данных ведущее устройство при введенном сигнале IRDY\* снимает сигнал FRAME\*. В одиночных транзакциях сигнал FRAME\* активен лишь один такт. Если устройство не поддерживает пакетные транзакции в ведомом режиме, то оно должно потребовать прекращения пакетной транзакции во время первой фазы данных (введя сигнал STOP\* одновременно с TRDY\*). В ответ на это ведущее устройство завершит данную транзакцию и продолжит обмен последующей транзакцией с новым

значением адреса. После последней фазы данных ведущее устройство снимает сигнал  $IRDY\#$ , и шина переходит в состояние покоя (*PCI Idle*) — оба сигнала  $FRAME\#$  и  $IRDY\#$  находятся в пассивном состоянии. Инициатор может начать следующую транзакцию и без такта покоя, введя  $FRAME\#$  одновременно со снятием  $IRDY\#$ . Такие быстрые смежные транзакции (*Fast Back-to-Back*) могут быть обращены как к одному, так и к разным ЦУ. Первый тип поддерживается всеми устройствами PCI, выступающими в роли ЦУ. На поддержку второго типа (она необязательна) указывает бит 7 регистра состояния (см. п. 6.2.12). Инициатору разрешают (если он умеет) использовать быстрые смежные транзакции с разными устройствами (битом 9 регистра команд), только если все агенты шины допускают быстрые обращения.

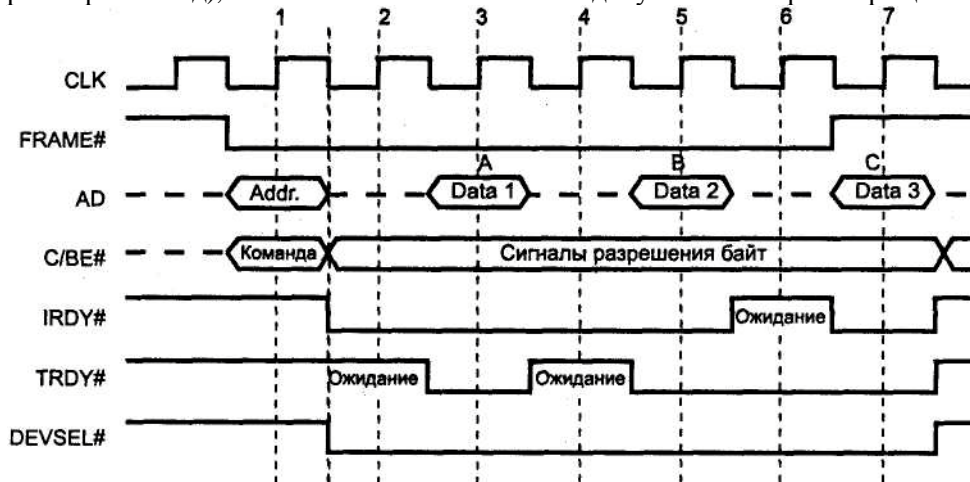


Рис. 6.7. Цикл обмена на шине PCI

Шина позволяет уменьшить мощность (ток), потребляемую устройствами, ценой снижения производительности, применяя *пошаговое переключение линий*  $AD[31:0]$  и  $PAR$  (address/data stepping). Здесь возможны два варианта.

- ◆ Плавный шаг (*continuous stepping*) — начало формирования сигналов слабо точными формирователями за несколько тактов до введения сигнала-квалификатора действительной информации ( $FRAME\#$  в фазе адреса,  $IRDY\#$  или  $TRDY\#$  в фазе данных). За эти несколько тактов сигналы «доползут» до требуемого значения при меньшем токе.
- ◆ Дискретный шаг (*diskrete stepping*) — нормальные формирователи срабатывают не все сразу, а группами (например, побайтно), в каждом такте по группе. При этом снижаются броски тока, поскольку одновременно переключается меньше формирователей.

Устройство само может и не пользоваться этими возможностями (см. бит 7 регистра команд), но должно «понимать» такие циклы. Задерживая сигнал  $FRAME\#$ , устройство рискует потерять право доступа к шине, если арбитр получит запрос от более приоритетного устройства.

Протокол квитирования обеспечивает надежность обмена — ведущее устройство всегда получает информацию об обработке транзакции ЦУ. Средством повышения надежности (достоверности) является применение контроля паритета: линии AD[31:0] и C/BE[3:0]# и в фазе адреса, и в фазе данных защищены битом паритета PAR (количество единичных бит этих линий, включая PAR, должно быть четным). Действительное значение PAR появляется на шине с задержкой в один такт относительно линий AD и C/BE#. При обнаружении ошибки ЦУ вырабатывается сигнал PERR\* (со сдвигом на такт после действительности бита паритета). В подсчете паритета при передаче данных учитываются все байты, включая и недействительные (отмеченные высоким уровнем сигнала C/BEх#). Состояние бит, даже и в недействительных байтах данных, во время фазы данных должно оставаться стабильным. Каждая транзакция на шине должна быть завершена планово или прекращена, при этом шина должна перейти в состояние покоя (сигналы FRAME\* и IRDY# пассивны). Завершение транзакции выполняется либо по инициативе ведущего устройства, либо по инициативе ПУ. Ведущее устройство может завершить транзакцию одним из следующих способов.

- ◆ Нормальное завершение (*Camletiori*) выполняется по окончании обмена данными.
- ◆ Завершение по тайм-ауту (*Time-out*) происходит, когда во время транзакции у ведущего устройства отбирают право на управление шиной (снятием сигнала GNT#) и истекает время, указанное в его таймере Latency Timer. Это может случиться, если адресованное ЦУ оказалось непредвиденно медленным или запланирована слишком длинная транзакция. Короткие транзакции (с одной - двумя фазами данных) даже в случае снятия сигнала GNT# и срабатывания таймера завершаются нормально. Транзакция отвергается (*Master-Abort*), когда в течение заданного времени ведущее устройство не получает ответа ЦУ (DEVSEL\*).

Транзакция может быть прекращена *по инициативе ЦУ*; для этого оно может ввести сигнал STOP\*. Возможны три типа прекращения.

- ◆ Повтор (*Retry*) — сигнал STOP\* вводится при пассивном сигнале TRDY\* до первой фазы данных. Эта ситуация возникает, когда ЦУ из-за внутренней занятости не успевает выдать первые данные в положенный срок (16 тактов). Повтор является указанием ведущему устройству на необходимость нового запуска той же транзакции.
- ◆ Отключение (*Disconnect*) — сигнал STOP\* вводится во время или после первой фазы данных. Если сигнал STOP\* введен при активном сигнале TRDY\* очередной фазы данных, то эти данные передаются и на том транзакция завершается. Если сигнал STOP\* введен при пассивном сигнале TRDY\*, то транзакция завершается без передачи данных очередной фазы. Отключение производится, когда ЦУ неспособно своевременно выдать или принять очередную порцию данных пакета.
- ◆ Отказ (*Target-Abort*) — сигнал STOP\* вводится одновременно со снятием сигнала DEVSEL\* (в предыдущих случаях во время появления сигнала STOP\* сигнал DEVSEL\* был активен). После этого данные уже не передаются. Отказ вводится, когда ЦУ обнаруживает фатальную ошибку или иные условия, по которым оно уже никак не сможет обслужить данный запрос.

Использование трех типов прекращения вовсе не обязательно для всех ЦУ, однако любое ведущее устройство должно быть готово к завершению транзакций по любой из этих причин.

### 6.2.3. Команды шины, адресация памяти и ввода-вывода

Работа шины контролируется несколькими таймерами, не позволяющими попусту расходовать такты шины и планировать распределение полосы пропускания.

Каждое ЦУ должно достаточно быстро отвечать на адресованную ему транзакцию. *Задержка первой фазы данных* (target initial latency), то есть задержка появления сигнала TRDY\* относительно FRAME\*, не должна превышать 16 тактов-шины. Если устройство по своей природе иногда может не успевать уложиться в этот интервал, оно должно формировать сигнал STOP\*, прекращая транзакцию. Это заставит ведущее устройство повторить транзакцию, и с большой вероятностью эта попытка окажется успешной. Если устройство медленное и часто не укладывается в 16 тактов, то оно должно откладывать транзакцию (Delayed Transaction). Кроме того, ЦУ имеет инкрементный механизм слежения за длительностью циклов (Incremental Latency Mechanism), который не позволяет интервалу между *соседними фазами данных* в пакете (target subsequent latency) превышать 8 тактов шины. Если ЦУ не успевает работать в таком темпе, оно обязано остановить транзакцию. Желательно, чтобы устройство сообщало о своем «неуспевании» как можно раньше, не выжидая предельных 16 или 8 тактов, — это экономит полосу пропускания шины.

Инициатор тоже не должен задерживать поток — допустимая *задержка от начала* FRAME\* до сигнала IRDY\* (master data latency) и между фазами данных не должна превышать 8 тактов. Если ЦУ время от времени отвергает операцию записи в память с запросом повтора (это, к примеру, может происходить при записи в видеопамять), то есть «предел терпения» для завершения операции. Таймер *максимального времени исполнения* (maximum complete time) имеет порог 10 мкс — 334 такта при 33 МГц или 668 тактов на 66 МГц, за которое инициатор должен иметь возможность «протолкнуть» хоть одну фазу данных. Таймер начинает отсчет с момента запроса повтора операции записи в память и сбрасывается при последующем завершении транзакции записи в память, отличном от запроса повтора. Устройства, не способные выдерживать ограничение на максимальное время исполнения записи в память, должны предоставлять драйверу возможность определять их состояние, в котором достаточно быстрая запись в память невозможна. Драйвер, естественно, должен учитывать это состояние и не «напрягать» шину и устройство бесплодными попытками записи.

Каждое ведущее устройство, способное сформировать пакет с более чем двумя фазами данных, должно иметь собственный программируемый *таймер задержки* (Latency Timer), регулирующий поведение ведущего устройства, когда у него отбирают право управления шиной. Таймер запускается по каждому сигналу FRAME\*, введенному этим ведущим устройством. Поведение ведущего устройства по достижении порога зависит от типа команды и состояния сигналов FRAME\* и GNT\* на момент срабатывания таймера.

- ◆ Если ведущее устройство снимает сигнал FRAME\* до срабатывания таймера, транзакция завершается нормально.
- ◆ Если сигнал GNT# снят и исполняемая команда не является записью памяти с инвалидацией, то инициатор обязан сократить транзакцию, сняв сигнал FRAME\*. При этом ему позволяется завершить текущую и выполнить еще одну фазу данных.
- ◆ Если сигнал GNT# снят и исполняется запись в память с инвалидацией, то инициатор должен завершить транзакцию по концу текущей (если передается последнее двойное слово строки) или следующей (если двойное слово — последнее) строки кэша.

*Задержка арбитража (arbitration latency)* определяется как число тактов от подачи инициатором запроса REQ# до получения права управления шиной GNT#. Эта задержка зависит от активности других инициаторов, быстродействия устройств (чем меньше они вводят тактов ожидания, тем лучше) и «проворности» собственно арбитра. В зависимости от исполняемой команды и состояния сигналов ведущее устройство должно либо сократить транзакцию, либо продолжать ее до запланированного завершения.

При конфигурировании ведущие устройства сообщают свои потребности, указывая максимально допустимую задержку предоставления доступа к шине (Max\_Lat) и минимальное время, на которое им должно предоставляться управление шиной (Min\_GNT). Эти потребности определяются присущим устройству темпом передачи данных и его организацией.

Для максимального использования возможностей шины устройства должны иметь буферы, чтобы накапливать в них данные для пакетных транзакций. Рекомендуется для устройств со скоростью передачи данных до 5 Мбайт/с иметь буфер, по крайней мере, на 4 двойных слова. Для более высоких скоростей рекомендуется буфер на 32 двойных слова. Для обмена с системной памятью наиболее эффективны транзакции, работающие с целыми строками кэша, что тоже учитывают при определении размера буфера. Однако увеличение размера буфера может вызвать трудности при обработке ошибок, а также вести к увеличению задержек доставки данных (пока устройство не заполнит определенный объем буфера, оно не начнет передачу этих данных по шине, и их потребители будут ожидать).

В спецификации приводится пример организации карты Fast Ethernet (скорость передачи — 10 Мбайт/с), у которой для каждого направления передачи имеется 64-байтный буфер, поделенный на две половины. Когда адаптер заполняет одну половину буфера приходящим кадром, он выводит в память накопленное содержимое другой половины, после чего они меняются местами. Каждая половина выводится в память за 8 фаз данных (около 0,25 мкс на частоте 33 МГц), что соответствует установке MIN\_GNT=1. При скорости прихода данных 10 Мбайт/с каждая половина заполняется за 3,2 мкс, что соответствует установке MAX\_LAT= 12 (здесь время задается в интервалах по 0,25 мкс).

#### 6.2.4. Таймеры, задержки и буферы

В каждой команде шины указывается адрес данных, передаваемых в первой фазе данных пакета. Адрес для каждой последующей фазы данных пакета увеличивается

ется на 4 (следующее двойное слово), но в командах обращения к памяти порядок может быть иным (см. ниже). Байты шины AD, несущие действительную информацию, выбираются сигналами C/BE[3:0]# в фазах данных. Внутри пакета эти сигналы могут менять состояние от фазы к фазе произвольным образом. Разрешенные байты могут быть разрозненными; возможны фазы данных, в которых не разрешено ни одного байта. В отличие от шины ISA, на PCI нет динамического изменения разрядности — все устройства должны подключаться к шине 32-разрядным способом. Если в устройстве PCI применяются функциональные схемы иной разрядности (к примеру, нужно подключить микросхему 8255, имеющую 8-битную шину данных и четыре регистра), то приходится принимать схемотехнические методы преобразования, отображающие все регистры на 32-разрядную шину AD.

*Адресация* памяти, портов и конфигурационных регистров различна.

- ◆ В циклах обращения к памяти адрес, выровненный по границе двойного слова, передается по линиям AD[31:2]; линии AD[1:0] задают порядок адресов в пакете:
  - 00 — линейное инкрементирование; адрес последующей фазы отличается от предыдущего на число байтов шины (4 для 32-битной и 8 для 64-битной шины).
  - 10 — Cacheline Wrap mode, сворачивание адресов с учетом длины строки кэш-памяти. В транзакции адрес для очередной фазы увеличивается до достижения границы строки кэша, после чего переходит на начало этой строки и увеличивается до адреса, предшествующего начальному. Если транзакция длиннее строки кэша, то она продолжится в следующей строке с того же смещения, что и началась. Так, при длине строки 16 байт и 32-битной шине транзакция, начавшаяся с адреса xxxxxxOSh, будет иметь последующие фазы данных, относящиеся к адресам xxxxxxOCh, xxxxxxOOh, xxxxxx04h; и далее к xxxxxxlSh, xxxxxxlCh, xxxxxxlOh, xxxxxx!4h. Длина строки кэша прописывается в конфигурационном пространстве устройства (см. п. 6.2.12). Если устройства не имеет регистра Cache Line Size, то оно должно прекратить транзакцию после первой фазы данных;
  - 01 и 11 — зарезервировано, может использоваться как указание на отключение (Disconnect) после первой фазы данных.
- ◆ В циклах обращения к портам ввода-вывода для адресации любого байта используются все линии AD[31:0]. При этом биты адреса AD[31:2] указывают на адрес двойного слова, к которому принадлежат передаваемые данные, а младшие биты адреса AD[1:0] должны соответствовать байтам, которые могут быть разрешены сигналами C/BE[3:0]#. При AD[1:0]=00 допустимо C/BE[3:0]# - xxxO или 1111, при AO[1:0]«01-C/BE[3:0]# = xx01 или 1111, при AO[1:0]=10-C/BE[3:0]# = x011 или 1111, при AD[1:0]=11 — C/BE[3:0]# = 0111 (передается лишь байт 3) или 1111 (ни один байт не разрешен). Эти циклы тоже могут быть пакетными, хотя на практике эта возможность используется редко.
- ◆ В циклах конфигурационной записи/считывания *устройство* (карта расширения) выбирается индивидуальным сигналом IDSEL; *функция* адресуется битами AD[10:8], а конфигурационные регистры (только двойные слова) адресуются битами AD[7:2], при этом AD[1:0]=00.

Команды шины PCI определяются значениями бит C/BE# в фазе адреса (табл. 6.12).

- ◆ *Команда подтверждения прерывания* предназначена для чтения вектора прерываний. По протоколу она выглядит как команда чтения, неявно адресованная к системному контроллеру прерываний. Здесь в фазе адреса по шине AD полезная информация не передается, но ее инициатор (главный мост) должен обеспечить стабильность сигналов и корректность паритета. В PC 8-битный вектор передается в байте 0 по готовности контроллера прерываний (по сигналу TRDY#). Подтверждение прерываний выполняется за один цикл (первый холостой цикл, который процессоры x86 делают в дань совместимости со стариной, мостом подавляется).
- ◆ *Специальный цикл* отличается от всех других тем, что является ширококестельным. Однако ни один агент на него не отвечает, а главный мост или иное устройство, вводящее этот цикл, всегда завершает его способом *Master Abort* (на него требуется 6 тактов шины). Специальный цикл предназначен для генерации ширококестельных сообщений — их могут читать любые «заинтересованные» агенты шины. Тип сообщения декодируется содержимым линий AD[15:0], на линиях AD[31:16] могут помещаться данные, передаваемые в сообщении. Фаза адреса в этом цикле для обычных устройств отсутствует, но мосты используют ее информацию для управления распространением сообщения. Сообщения с кодами 0000h, 0001h и 0002h требуются для указания на отключение (Shutdown), остановку (Halt) процессора или специфические функции процессора x86, связанные с кэшем и трассировкой. Коды 0003-FFFFh зарезервированы. Специальный цикл может генерироваться тем же аппаратно-программным механизмом, что и конфигурационные циклы (см. п. 6.2.11), но со специфическим значением адреса.
- ◆ *Команды чтения и записи ввода-вывода* служат для обращения к пространству портов. Линии AD содержат адрес байта, причем декодированию подлежат и биты ADO и AD1 (несмотря на то, что имеются сигналы VEx#). Порты PCI могут быть 16- или 32-битными. Для адресации портов на шине PCI доступны все 32 бита адреса, но процессоры x86 могут использовать только младшие 16 бит.
- ◆ *Команды обращения к памяти*, кроме обычного чтения и записи, включают чтение строк кэш-памяти, множественное чтение (нескольких строк), запись с инвалидацией.
- ◆ *Команды конфигурационного чтения и записи* адресуются к конфигурационному пространству устройств (см. п. 6.2.12). Обращение производится только двойными словами. Структура содержит идентификатор устройства и производителе для, состояние и команду, информацию о занимаемых ресурсах и ограничения на использование шины. Для генерации данных команд требуется специальный аппаратно-программный механизм (см. п. 6.2.11).
- ◆ *Чтение строк памяти* применяется, когда в транзакции планируется более двух 32-битных передач (обычно это чтение до конца строки кэша).
- ◆ *Множественное чтение памяти* используется для транзакций, пересекающих границы строк кэш-памяти.
- ◆ *Запись с инвалидацией* применяется к целым строкам кэша и позволяет оптимизировать циклы обратной записи «грязных» строк кэша.



- ♦ *Двухадресный цикл* позволяет по 32-битной шине обращаться к устройствам с 64-битной адресацией. В этом случае младшие 32 бита адреса передаются в цикле данного типа, а за ним следует обычный цикл, определяющий тип обмена и несущий старшие 32 бита адреса. Шина PCI допускает 64-битную адресацию портов ввода-вывода (для x86 это бесполезно, но PCI существует и на других платформах).

Таблица 6.12. Декодирование команд шины PCI

C/BE[3:0]	Тип команды
0000	Interrupt Acknowledge — подтверждение прерывания
0001	Special Cycle — специальный цикл
0010	I/O Read — чтение порта ввода-вывода
0011	I/O Write — запись в порт ввода-вывода
0100	Зарезервировано
0101	Зарезервировано
0110	Memory Read — чтение памяти
0111	Memory Write — запись в память
1000	Зарезервировано
1001	Зарезервировано
1010	Configuration Read — конфигурационное считывание
1011	Configuration Write — конфигурационная запись
1100	Multiple Memory Read — множественное чтение памяти
1101	Dual Address Cycle (DAC) — двухадресный цикл
1110	Memory-Read Line — чтение строки памяти
1111	Memory Write and Invalidate — запись с инвалидацией

## 6.2.5. Пропускная способность шины

Шина PCI является самой высокоскоростной шиной расширения современных ПК, однако и ее реальная пропускная способность, увы, не так уж и высока. Рассмотрим наиболее распространенный вариант: разрядность 32 бита, частота 33 МГц. Как указывалось выше, пиковая скорость передачи данных внутри пакетного цикла составляет 132 Мбайт/с, то есть за каждый такт шины передаются 4 байта данных ( $33 \times 4 = 132$ ). Однако пакетные циклы выполняются далеко не всегда. Процессор общается с устройствами PCI инструкциями обращения к памяти или вводу-выводу через главный мост, который шинные транзакции процессора транслирует в транзакции шины PCI. Поскольку у процессоров x86 основные регистры 32-разрядные, то одна инструкция порождает транзакцию с устройством PCI, в которой передается не более 4 байт данных, что соответствует одиночной передаче. Если же адрес передаваемого (двойного) слова не выровнен по соответствующей границе, то будут порождены два одиночных цикла или один пакетный с двумя фазами данных, но в любом случае это обращение будет выполняться дольше, чем при выровненном адресе.

Однако при *записи* массива данных в устройство PCI (передача с последовательно нарастающим адресом) мост может пытаться организовать пакетные циклы. У современных процессоров (начиная с Pentium) шина данных 64-битная и применяется буферизация записи, так что два последовательных 32-битных запроса записи объединятся в один 64-битный. Этот запрос, если он адресован к 32-битному устройству, мост попытается передать пакетом с двумя фазами данных. «Продвинутый» мост может пытаться собирать в пакет и последовательные запросы, что может породить пакет существенной длины. Пакетные циклы записи можно наблюдать, например, передавая массив данных из ОЗУ в устройство PCI строковой инструкцией MOVSD, используя префикс повтора REP. Тот же эффект даст и цикл последовательных операций LODSW, STOSW (и иных инструкциях обращения к памяти). Поскольку у современных процессоров ядро исполняет инструкции гораздо быстрее, чем шина способна вывести их результаты, между инструкциями, порождающими объединяемые записи, процессор может успеть выполнить еще несколько операций. Однако если пересылка данных организуется директивой языка высокого уровня, которая ради универсальности работает гораздо сложнее вышеприведенных ассемблерных примитивов, транзакции, скорее всего, будут уже одиночными (у буферов записи процессора не хватит «терпения» придержать один 32-битный запрос до появления следующего, или же произойдет принудительная выгрузка буферов записи процессора или моста по запросу чтения, см. п. 6.2.10).

Что касается чтения из устройства PCI, то здесь пакетный режим организовать сложнее. Буферизации чтения у процессора, естественно, нет (операцию чтения можно считать выполненной лишь по получению реальных данных), и даже строковые инструкции будут порождать одиночные циклы. Однако у современных процессоров имеются возможности генерации запросов чтения более 4 байт. Для этого можно использовать инструкции загрузки данных в регистры MMX (8 байт) или XMM (16 байт), а из них уже выгружать данные в ОЗУ (которое работает много быстрее устройств PCI).

Строковые инструкции ввода-вывода (INSW, OUTSW с префиксом повторения REP), используемые для программированного ввода-вывода блоков данных (PIO), порождают серии одиночных транзакций, поскольку все данные блока относятся к одному адресу PCI.

Посмотреть, каким образом происходит обращение к устройству, несложно при наличии осциллографа: в одиночных транзакциях сигнал FRAME\* активен всего 1 такт, в пакетных он длиннее. Число фаз данных в пакете соответствует числу тактов, во время которых активны оба сигнала IRDY\* и TRDY\*.

Стремиться к пакетизации транзакций записи стоит только в том случае, если устройство PCI поддерживает пакетные передачи в ведомом (target) режиме. Если это не так, то попытка пакетизации приведет даже к небольшой потере производительности, поскольку транзакция будет завершаться по инициативе ведомого устройства (сигналом STOP\*), а не инициатора обмена, на чем теряется один такт шины. Так, к примеру, можно наблюдать, как при записи массива в память PCI, выполняемой директивой языка высокого уровня, устройство среднего быстродействия (вводящее лишь 3 такта ожидания готовности) принимает данные каждые 7 тактов,

что при частоте 33 МГц и разрядности 32 бита дает скорость  $33 \times 4 / 7 = 18,8$  Мбайт/с. Здесь 4 такта занимает активная часть транзакции (от сигнала FRAME\* до снятия сигнала IRDY\*) и 3 такта паузы. То же устройство по инструкции MOVSD принимает данные каждые 8 тактов шины ( $33 \times 4 / 8 = 16,5$  Мбайт/с). Эти данные — результат наблюдения работы PCI-ядра, выполненного на основе микросхемы FPGA фирмы Altera, не поддерживающего пакетные транзакции в ведомом режиме. То же самое устройство при чтении памяти PCI работает существенно медленнее — инструкцией REP MOWSW с него удалось получать данные каждые 19-21 тактов шины (скорость  $33 \times 4 / 20 = 6,6$  Мбайт/с). Здесь сказывается и большая задержка устройства (оно выдает данные лишь в 8 такте после появления сигнала FRAME\*), и то, что процессор начинает следующую пересылку лишь дождавшись данных от предыдущей. Трюк с использованием регистра XMM здесь дает положительный эффект, несмотря на потерю такта (на прекращение транзакции непакетным устройством), поскольку каждый 64-битный запрос процессора выполняется парой смежных транзакций PCI, между которыми пауза всего в пару тактов. Для определения теоретического предела пропускной способности вернемся к рис. 6.7, чтобы определить минимальное время (число тактов) транзакций чтения и записи. В транзакции чтения после подачи команды и адреса инициатором (такт 1) меняется текущий «владелец» шины AD. На этот «разворот», или «пируэт» (turnaround), уходит такт 2, что обуславливается задержкой сигнала TRDY\* целевым устройством. Далее может следовать фаза данных (такт 3), если целевое устройство достаточно расторопно. После последней фазы данных требуется еще 1 такт на обратный «пируэт» шины AD (в нашем случае это такт 4). Таким образом, чтение одного слова (4 байта) занимает минимум 4 такта по 30 нс ( $33 \text{ МГц}$ ). Если эти транзакции следуют непосредственно друг за другом (если на такое способен инициатор и у него не отбирают право на управление шиной), то можно говорить о максимальной скорости чтения в 33 Мбайт/с при одиночных транзакциях. В транзакциях записи шиной AD все время управляет инициатор, так что здесь нет потери тактов на «пируэт». При расторопном целевом устройстве, не вносящем дополнительных тактов ожидания, скорость записи может достигать 66 Мбайт/с.

Скорость, соизмеримую с максимальной пиковой, можно получить только при пакетных передачах, когда имеют место дополнительные 3 такта при чтении и 1 при записи. Так, для чтения пакета с числом фаз данных 4 требуется 7 тактов ( $V = 16 / (7 \times 30)$  байт/нс - 76 Мбайт/с), а для записи - 5 ( $V = 16 / (5 \times 30)$  байт/нс = 106,6 Мбайт/с). При числе фаз данных в 16 скорость чтения может достигать 112 Мбайт/с, а записи — 125 Мбайт/с.

В этих выкладках не учитывались потери времени, связанные со сменой инициатора. Инициатор может начинать транзакцию по получении сигнала GNT#, только убедившись в том, что шина находится в покое (сигналы FRAME\* и IRDY\* пассивны); на фиксацию покоя уходит один такт. Как видно, захватывать для одного инициатора большую часть пропускной способности шины можно, увеличивая длину пакета. Однако при этом возрастет задержка получения управления шиной для других устройств, что не всегда допустимо. Отметим также, что далеко не все устройства способны отвечать на транзакции без тактов ожидания, так что реальные цифры будут скромнее.

Итак, для выхода на максимальную производительность обмена устройства PCI сами должны быть ведущими устройствами шины, причем способными генерировать пакетные циклы. Поддержку пакетного режима имеют далеко не все устройства PCI, а у имеющих, как правило, есть существенные ограничения на максимальную длину пакета. Радикально повысить пропускную способность позволяет переход на частоту 66 МГц и разрядность 64 бита, что обходится недешево. Для того, чтобы на шине могли нормально работать устройства, критичные к времени доставки данных (сетевые адаптеры, устройства, участвующие в записи и воспроизведении аудио-видеоданных и др.), не следует пытаться выжать из шины ее декларированную полосу пропускания полностью. Перегрузка шины может привести, например, к потере пакетов из-за несвоевременности доставки данных. Заметим, что адаптер Fast Ethernet (100 Мбит/с) в полудуплексном режиме занимает полосу около 13 Мбайт/с (10% декларируемой полосы обычной шины), а в полнодуплексном — уже 26 Мбайт/с. Адаптер Gigabit Ethernet даже в полудуплексном режиме вписывается в полосу шины уже с натяжкой (он «выживает» лишь за счет больших внутренних буферов), для него больше подходит 64 бит/66 МГц.

### 6.2.6. Прерывания

В PC-совместимых компьютерах прерывания от устройств PCI обслуживаются с помощью традиционной связки пары контроллеров 8259A, расположенных на системной плате (см. п. 12.4), к которым обращается команда «подтверждение прерывания». Прерывания на шине PCI свободны от одной из нелепостей системы прерываний ISA. Устройство PCI вводит сигнал прерывания *низким* уровнем (выходом с открытым коллектором или стоком) на выбранную линию INTA#, INTB#, INTC# или INTD\*. Этот сигнал должен удерживаться до тех пор, пока программный драйвер, вызванный по прерыванию, не сбросит запрос прерывания, обратившись по шине к данному устройству. Если после этого контроллер прерываний снова обнаруживает низкий уровень на линии запроса, это означает, что запрос на ту же линию ввело другое устройство, разделяющее данную линию с первым, и оно тоже требует обслуживания. Линии запросов от слотов PCI и PCI-устройств системной платы коммутируются на входы контроллеров прерываний относительно произвольно. Конфигурационное ПО может определить и указать занятые линии запросов и номер входа контроллера прерываний обращением к конфигурационному пространству устройства (см. п. 6.2.12). Программный драйвер, прочитав конфигурационные регистры, тоже может определить эти параметры для того, чтобы установить обработчик прерываний на нужный вектор и при обслуживании сбрасывать запрос с требуемой линии. К сожалению, в конфигурационных регистрах не нашлось стандартного места для бита, индицирующего введение запроса прерывания данным устройством, — тогда бы в прерываниях для PCI не было бы проблем с унификацией поддержки разделяемых прерываний.

Каждая функция устройства PCI может задействовать свою линию запроса прерывания, но должно быть готовым к ее разделению (совместному использованию) с другими устройствами. Если устройству требуется только одна линия запроса, то оно должно занимать линию INTA#, если две — INTA# и INTB#, и так далее. С учетом циклического сдвига линий запроса это правило позволяет установить в 4 сосед-

них слота 4 простых устройства, и каждое из них будет занимать отдельную линию запроса прерывания. Если какой-то карте требуется две линии, то для монопольного использования прерываний нужно оставить соседний слот свободным. PCI-устройства системной платы тоже задействуют прерывания с той же закономерностью (кроме контроллера IDE, который, к счастью, держится особняком).

Назначение прерываний устройствам (функциям) выполняет процедура POST, и этот процесс управляем лишь частично. Параметрами CMOS Setup (PCI/PNP Configuration) пользователь определяет номера запросов прерываний, доступных шине PCI. В зависимости от версии BIOS это может выглядеть по-разному: либо каждой линии INTA#...INTD# явно назначается свой номер, либо ряд номеров отдается «на откуп» устройствам PCI вместе с устройствами ISA PnP (в противоположность устройствам «Legacy ISA»). В итоге POST определяет соответствие линий INTA#...INTD# номерам запросов контроллера и соответствующим образом программирует коммутатор запросов. По воле пользователя может оказаться так, что не каждой линии запроса шины PCI достается отдельный вход контроллера прерываний. Тогда коммутатор организует объединение нескольких линий запросов PCI на один вход контроллера, то есть разделяемые прерывания. В самом худшем случае устройствам PCI не достанется ни одного входа контроллера прерываний. Заметим, что BIOS вряд ли отдаст шине PCI прерывания 14 и 15 (их забирает контроллер IDE, если он не отключен), а также 3 и 4 (COM-порты).

Драйвер (или иное ПО), работающий с устройством PCI, определяет вектор прерывания, доставшийся устройству (точнее, функции), чтением конфигурационного регистра Interrupt Line. В этом регистре указывается номер входа контроллера прерывания (255 — номер не назначен), и по нему определяется вектор (см. п. 12.4). Номер входа каждому устройству заносит тест POST. Для этого он считывает регистр Interrupt Pin каждой обнаруженной функции и по адресу устройства (!) определяет, какая из линий (PC1\_И...PC1\_4) используется. Заметим, что правила, по которым на системной плате определяется соответствие между Interrupt Pin и входными линиями коммутатора запросов в зависимости от номера устройства, строго не регламентированы (деление номера устройства на 4 — это всего лишь рекомендация), но их твердо знает версия BIOS данной системной платы. К этому моменту тест POST уже определил таблицу соответствия этих линий номерам входов; пользуясь этой таблицей, он записывает нужное значение в конфигурационный регистр Interrupt Line. Определить, есть ли еще претенденты на тот же номер прерывания, можно, лишь просмотрев конфигурационные регистры функций всех устройств, обнаруженных на шине (это не так уж сложно сделать, пользуясь функциями PCI BIOS). «Прелести» разделяемых прерываний обсуждаются в п. 12.4.1.

Спасением от бед «разделяемости» может быть перестановка карт в подходящий слот. Однако попадают «подарки разработчиков» интегрированных плат, у которых из нескольких слотов PCI неразделяемая линия прерывания есть только у одного (а то и нет вообще). Такие недуги без скальпеля и паяльника, как правило, не лечатся.

На шине PCI имеется и иной механизм оповещения об асинхронных событиях, основанный на передаче сообщений (PCI Message-Based Interrupts). Для сигнализации запроса прерывания устройство запрашивает управление шиной и, получив его, выполняет запись номера прерывания по заранее оговоренному адресу.

Этот механизм может использоваться на системных платах, имеющих «продвинутый» контроллер прерываний APIC. Запись номера запроса производится в соответствующий регистр APIC. Для системных плат на чипсете с хабом ICH2 82801 этот регистр находится по адресу памяти FEC00020h, а номер прерывания может быть в диапазоне 0-23L. Однако одновременно оба механизма работать не могут; если разрешена работа APIC, то логика контроллеров 8259 не используется, и наоборот.

### 6.2.7. Прямой доступ к памяти, эмуляция ISA DMA (PC/PCI)

Как было сказано выше, шина PCI не предоставляет возможности прямого доступа к памяти с использованием централизованного контроллера в стиле 8237A (как для шины ISA). Для разгрузки центрального процессора от рутинных перекачек данных предлагается прямое управление шиной со стороны устройств, называемых ведущими устройствами шины (PCI Bus Master). Степень интеллектуальности ведущего устройства может быть различной. В простейшем варианте ведущее устройство обеспечивает пересылку блоков данных между устройством и системной памятью (или памятью других устройств) по указанию от CPU. Здесь CPU командами обращения к определенным регистрам ведущего устройства задает начальный адрес, длину блока, направление пересылки и разрешает запуск передачи. После этого пересылка выполняется по готовности (или инициативе) устрой\* ства, без отвращения CPU. Таким образом выполняется прямой доступ к памяти (DMA). Более сложный контроллер DMA может организовывать сцепку буферов при чтении, разбросанную запись и т. п. — возможности, знакомые еще по «продвинутым» контроллерам DMA для ISA/EISA. Более интеллектуальное ведущее устройство, как правило, обладающее собственным микроконтроллером, не ограничивается такой простой работой по указке CPU — оно выполняет обмены уже по программе своего контроллера.

Для совместимости устройств PCI со старым PC-ориентированным ПО и упрощения устройств PCI фирма Intel разработала специальный *протокол PC/PCI DMA*, изменяющий назначение пары сигналов REQi# и GNTi# для заранее выбранного агента шины, являющегося «проводником» DMA. Этот агент имеет внешние (по отношению к шине PCI) пары сигналов DRQx# и DACKx# с логикой, аналогичной одноименным сигналам ISA (см. п. 6.1), а линии REQi# и GNTi# в процессе запроса управления шиной использует особым образом. Когда агент получает запрос DRQx (один или несколько), он по линии REQi# передает в последовательном коде номера активных линий запросов DRQx, с синхронизацией по линии CLK. В первом такте CLK передается старт-бит — низкий уровень REQi#, во втором — активность запроса DRQ0, затем DRQ1 и так далее до DRQ7, после чего сохраняется низкий уровень REQ#. На это сообщение арбитр ответит по линии GNTi# посылкой, также начинающейся со старт-бита, за которой последуют три бита кода номера канала, которому дается подтверждение DACK\* для передачи данных в этой транзакции. Агент должен сообщать арбитру обо всех изменениях линий запроса, в том числе и о снятии сигналов запроса. Механизм PC/PCI DMA может быть реализован только в чипсете системной платы.

## 6.2.8. Электрический интерфейс, слоты и карты PCI

Для работы на шине PCI используются микросхемы КМОП (CMOS), причем имеются две спецификации: с напряжениями питания интерфейсных схем 5 и 3,3 В. Для них применимы параметры сигналов на постоянном токе, приведенные в табл. 6.13. Однако мощность интерфейсных элементов (транзисторов для вентилях) выбрана меньшей, чем требовалось бы для переключения сигналов на высокой частоте (33 или 66 МГц). Здесь используется эффект отражения сигналов, формируемых микросхемами на проводниках шины, от несогласованных концов этих проводников, являющихся для таких высоких частот длинными линиями. На концах проводников шины нет терминаторов, поэтому от них проходящая волна сигнала отражается с тем же знаком и с той же амплитудой. Складываясь с прямым сигналом, обратная волна и обеспечивает нужный приемнику уровень сигнала. Таким образом, передатчик генерирует сигнал, который до прихода отраженного находится между уровнями переключения.

Линии управляющих сигналов FRAME\*, TRDY#, IRDY#, DEVSEL\*, STOP\*, SERR#, PERR#, LOCK#, INTA#, INTB#, INTC#, INTD#, REQ64\* и ACK64\* на системной плате подтягиваются к шине питания резисторами (типично 2,7 кОм для версии 5 В и 8,2 кОм для 3,3 В), чтобы не было ложных срабатываний при пассивности всех агентов шины.

Таблица 6.13. Параметры интерфейсных сигналов на постоянном токе

Параметр	5 В	3,3 В
Входное напряжение низкого уровня, В	$-0,5 < U_{\text{Н}} < 0,8$	$-0,5 < U_{\text{Н}} < 0,3 \times V_{\text{cc}}$
Входное напряжение высокого уровня, В	$2 < U_{\text{Н}} < V_{\text{cc}} + 0,5$	$V_{\text{cc}} / a S U_{\text{Н}} S V_{\text{cc}} + 0,5$
Выходное напряжение низкого уровня, В	$U_{\text{OL}} < 0,55$	$U_{\text{OL}} < 0,1 \times V_{\text{cc}}$
Выходное напряжение высокого уровня, В	$U_{\text{OH}} > 0,8$	$U_{\text{OH}} > 0,9 \times V_{\text{cc}}$
Напряжение питания Vcc, В	$4,75 < U_{\text{cc}} < 5,25$	$3,3 < U_{\text{cc}} < 3,6$

Электрическая спецификация рассчитана на два предельных варианта нагрузки одной шины: 2 устройства PCI на системной плате плюс 4 слота или 4 устройства и 2 слота. При этом подразумевается, что одно устройство на каждую линию шины PCI дает только единичную КМОП-нагрузку. В слоты могут устанавливаться карты, тоже дающие только единичную нагрузку. На длину проводников, а также топологию расположения элементов и проводников на картах расширения накладываются жесткие ограничения. Из-за этого изготовление самодельных карт PCI на логических микросхемах средней степени интеграции становится проблематичным.

Слоты PCI представляют собой щелевые разъемы, имеющие контакты с шагом 0,05 дюйма. Слоты расположены несколько дальше от задней панели, чем ISA/ EISA или MCA. Компоненты карт PCI расположены на левой поверхности плат. По этой причине крайний PCI-слот обычно совместно использует посадочное место адаптера (прорезь на задней стенке корпуса) с соседним ISA-слотом. Такой

слот называют *разделяемым* (shared slot), в него может устанавливаться либо карта ISA, либо PCI.

Карты PCI могут предназначаться для уровня интерфейсных сигналов 5 В и 3,3 В, а также быть универсальными. Слоты PCI имеют уровни сигналов, соответствующие питанию микросхем PCI-устройств системной платы: либо 5 В, либо 3,3 В. Во избежание ошибочного подключения слоты имеют ключи, определяющие номинал напряжения. Ключами являются пропущенные ряды контактов 12,13 и 50, 51. Для слота на 5 В ключ расположен на месте контактов 50, 51; для 3,3 В — 12, 13. На краевых разъемах карт PCI имеются ответные прорези на месте контактов 50,51 (5 В) и 12,13 (3,3 В); на универсальной карте имеется оба ключа. Ключи не позволяют установить карту в слот с неподходящим напряжением питания. Карты и слоты различаются лишь питанием буферных схем, которое поступает с линий +V I/O:

- ◆ на слоте «5 В» на линии +V I/O подается +5 В;
- ◆ на слоте «3,3 В» на линии +V I/O подается +(3,3-3,6) В;
- ◆ на карте «5 В» буферные микросхемы рассчитаны только на питание +5 В;
- ◆ на карте «3,3 В» буферные микросхемы рассчитаны только на питание +(3,3-3,6) В;
- ◆ на универсальной карте буферные микросхемы допускают оба варианта питания и будут нормально формировать и воспринимать сигналы по спецификациям 5 или 3,3 В, в зависимости от типа слота, в который установлена карта.

На слотах обоих типов присутствуют *питающие напряжения* +3,3, +5, +12и-12В на одноименных линиях. В PCI 2.2 определена дополнительная линия S.SVaux — «дежурное» питание +3,3 В для устройств, формирующих сигнал PME# при отключенном основном питании.

На системных платах чаще всего встречаются 5-вольтовые 32-битные слоты, заканчивающиеся контактами A62/B62; 64-битные слоты встречаются реже, они длиннее и заканчиваются контактами A94/B94. Конструкция разъемов и протокол позволяют устанавливать 64-битные карты и в 32-битные разъемы, и наоборот, но при этом, естественно, обмен будет в 32-битном режиме.

*Тактовая частота шины* определяется по возможностям чипсета и всех абонентов шины. Высокая частота 66 МГц может устанавливаться тактовым генератором только при высоком уровне на линии M66EN. Таким образом, установка любой карты, не поддерживающей 66 МГц (с заземленным контактом B49), приведет к понижению частоты, шины до 33 МГц. Серверные системные платы, на которых имеется несколько шин PCI, позволяют использовать на разных шинах разные частоты (66 и 33 МГц). Так, например, можно на 64-битных слотах использовать частоту 66 МГц, а на 32-битных — 33. Разгон нормальной частоты 33 МГц до 40-50 МГц аппаратно не контролируется, но может приводить к ошибкам работы карт расширения.

На рис. 6.8 изображена 32-битная карта максимального размера (Long Card), длина короткой платы (Short Card) — 175 мм, но многие карты имеют и меньшие размеры. Карта имеет обрамление (скобку), стандартное для конструктива ISA (раньше встречались карты и с обрамлением в стиле MCA IBM PS/2). Назначение выводов универсального разъема приведено в табл. 6.14.





Рис. 6.8. Карта расширения для шины PCI

Таблица 6.14. Разъемы шины PCI

Ряд В	№	РЯДА	Ряд В	№
-12 В	1	TRST#	GND/M66EN <sup>1</sup>	49
ТСК	2	+12В	GND/Ключ 5 В	50
GND	3	TMS	GND/Ключ 5 В	51
TDO	4	TDI	AD 8	52
+5 В	5	+5 В	AD 7	53
+5 В	6	INTA#	+3,38	54
INTB#	7	INTC#	AD S	55
INTD#	8	+5 В	AD3	56
PRSENT1*	9	Резерв	GND	57
Резерв	10	+VI/O	AD1	58
PRSENT2#	11	Резерв	+VI/O	59
GND/Ключ 3,3 В	12	GND/Ключ 3,3 В	ACK64#	60
GND/Ключ 3,3 В	13	GND/Ключ 3,3 В	+5 В	61
Резерв	14	3.3Vaux <sup>2</sup>	+5 В	62
GND	15	RST#	Конец 32-битного разъема	
CLK	16	+VI/O	Резерв	63
GND	17	GNT#	GND	64
REQ#	18	GND	C/BE6*	65
+VI/O	19	PME# <sup>2</sup>	C/BE4#	66
AD 31	20	AD 30	GND	67
AD 29	21	+3,3 В	AD63	68
GND	22	AD 28	AD61	69
AD 27	23	AD 26	+VI/O	70
AD 25	24	GND	AD59	71
+3,3 В	25	AD 24	AD57	72
C/BE3#	26	IDSEL	GND	73
AD 23	27	+3,3 В	AD55	74

Ряд В	№	РЯДА	Ряд В	№	РЯДА
GND	28	AD 22	AD 53	75	+VI/O
AD 21	29	AD 20	GND	76	AD 52
AD 19	30	GND	AD 51	77	AD 50 *
+3,3 В	31	AD 18	AD 49	78	GND
AD 17	32	AD 16	+VI/O	79	AD 48
C/BE2#	33	+3,3 В	AD 47	80	AD 46
GND	34	FRAME*	AD 45	81	GND
IRDY#	35	GND	GND	82	AD 44
+3,3 В	36	TRDY#	AD 43	83	AD 42
DEVSEL*	37	GND	AD 41	84	+VI/O
GND	38	STOP#	GND	85	AD 40
LOCK#	39	+3,3 В	AD 39	86	AD 38
PERR#	40	(SDONE#) <sup>3</sup>	AD 37	87	GND
+3,3 В	41	(SBOFF#) 3	+VI/O	88	AD 36
SERR#	42	GND	AD 35	89	AD 34
+3,3 В	43	PAR	AD 33	90	GND
C/BE1#	44	AD 15	GND	91	AD 32
AD 14	45	+3,3 В	Резерв	92	Резерв
GND	46	AD 13	Резерв	93	GND
AD 12	47	AD 11	GND	94	Резерв
AD 10	48	GND	Конец 64-битного разъема		

<sup>1</sup> Сигнал M66EN определен в PCI 2.1 только для слотов на 3,3 В.

<sup>2</sup> Сигнал введен в PCI 2.2 (прежде был резерв).

<sup>3</sup> Сигналы упразднены в PCI 2.2 (для совместимости на системной плате подтягиваются к высокому уровню резисторами 5 кОм).

На слотах PCI имеются контакты для тестирования адаптеров по интерфейсу JTAG (сигналы TCK, TDI, TDO, TMS и TRST#). На системной плате эти сигналы задействованы не всегда, но они могут и организовывать логическую цепочку тестируемых адаптеров, к которой можно подключить внешнее тестовое оборудование. Для непрерывности цепочки на карте, не использующей JTAG, должна быть связь TDI-TDO.

На некоторых старых системных платах позади одного из слотов PCI имеется разъем Media Bus, на который выводятся сигналы ISA. Он предназначен для размещения на графическом адаптере PCI звукового чипсета, предназначенного для шины ISA.

### 6.2.9. Иные конструктивы с шиной PCI

Шина PCI имеет и другие конструктивные исполнения; их спецификации доступны на сайте [www.pcisig.org](http://www.pcisig.org) (правда, только для членов данной организации либо за деньги).

*Low-Profile PCI* — низкопрофильный вариант карты PCI с обычным разъемом, но измененной крепежной скобкой. Эти карты можно устанавливать вертикально (без переходника riser card) даже в низкопрофильные корпуса (например, 19" формата высотой 2U). Для этих карт предусматривается напряжение питания интерфейсных схем только 3,3 В (но шина питания 5 В сохраняется).

*Small PCI (SPCI)* — спецификация PCI в миниатюрном исполнении, прежде называвшаяся *SFFPCI (Small Form-Factor)*. Эта спецификация, предназначенная, в основном, для портативных компьютеров, логически совпадает с обычной шиной PCI. Шина 32-битная, 64-битное расширение не предусматривается, и при частоте 33 МГц обеспечивается пропускная способность 132 Мбайт/с. Как и на всех шинах PCI/здесь поддерживается прямое управление (bus mastering). В дополнение к обычному набору сигналов появился новый CLKRUN, с помощью которого хост и устройства могут управлять частотой синхронизации в интересах энергосбережения. По размерам карта SPCI совпадает с PC Card и Card Bus, но специальные ключи предотвращают ошибки подключения. Для подключения карт SPCI на системной плате устанавливается двухрядный 108-контактный штырьковый разъем с шагом контактов 2 мм. Карта расширения может подключаться к нему непосредственно, но также может использоваться переходник с двусторонними ленточными контактами с шагом 0,8 мм. Шина SPCI является внутренней (карты расширения находятся под крышкой корпуса и устанавливаются изготовителем при выключенном питании) и поэтому не нацелена на замену Card BUS (шина для внешних подключений с возможностью горячей замены). Карты SPCI могут быть трех видов: с питанием 5 В, 3,3 В и универсальные 5/3,3 В. Благодаря уменьшению размеров (длины проводников) понижены требования к мощности сигналов. Карты SPCI позволяют использовать преимущества модульных решений (можно разгрузить системную плату), обеспечивая высокую производительность обмена (чего не обеспечивает Card Bus).

*Mini PCI Specification* — малогабаритный вариант карт PCI (2,75"х1,81"х0,22"). Логически и электрически соответствует PCI (32 бит), дополнительно используя сигнал CLKRUN для снижения энергопотребления и без сигналов JTAG. Имеет дополнительные сигналы для аудио- и видео-применений.

*PCI-X* — спецификация шины с высокой пропускной способностью, достигающей 1 Гбайт/с: тактовая частота 133 МГц и разрядность 64 бит. Разработана с учетом совместимости шины со старыми устройствами и новыми устройствами с обычной шиной PCI, для чего при наличии в системе хоть одного из «тихоходных» компонентов частота понижается до обычных 66 или 33 МГц. Кроме повышения тактовой частоты, имеются нововведения в протоколе шины (например, расщепленные транзакции), повышающие эффективность ее использования. Для устройств промышленного назначения в начале 1995 года был принят стандарт *Compact PCI*. Шина Compact PCI (cPCI) разрабатывалась на основе спецификации PCI 2.1. Этот стандарт принят организацией производителей промышленных компьютеров PCIMG (PCI Industrial Computer Manufacturers Group). Шина отличается большим количеством поддерживаемых слотов: 8 против 4. Появились новые 4 пары сигналов запросов и предоставления управления шиной.

Шина поддерживает 32-битный и 64-битный обмен (с индивидуальным разрешением байт). При частоте шины 33 МГц максимальная пропускная способность составляет 133 Мбайт/с для 32 бит и 266 Мбайт/с для 64 бит (в середине пакетного цикла). Возможна работа на частоте 66 МГц. Шина поддерживает спецификацию PnP. Кроме того, в шине возможно применение географической адресации, при этом адрес модуля (на который он отзывается при программном обращении) определяется его положением в каркасе. Для этого на коннекторе J1 имеются контакты GAO...GA4, коммутацией которых на «землю» для каждого слота задается его двоичный адрес. Географическая адресация позволяет переставлять однотипные модули, не заботясь о конфигурировании их адресов (хорошая альтернатива системе PnP — модуль «встанет» всегда в одни и те же адреса, которые без физического вмешательства больше не изменятся). Конструктивно платы Compact PCI представляют собой еврокарты высотой 3U (100 x 160 мм) с одним коннектором (J1) или 6U (233 x 160 мм) с двумя коннекторами (J1 и J2). Коннекторы — 7-ряд-ные штырьковые разъемы с шагом 2 мм между контактами, на кросс-плате — вилка, на модулях — розетки. Контакты коннекторов имеют разную длину: более длинные контакты цепей питания при установке модуля соединяются раньше, а при вынимании разъединяются позже, чем сигнальные. Такое решение позволяет производить «горячую» замену модулей. Собственно шина использует только один коннектор (J1), причем в 32-битном варианте не полностью — часть контактов может задействовать пользователь. 64-битная шина использует коннектор полностью. Одно посадочное место на кросс-плате резервируется под контроллер шины, на который возлагаются функции арбитража и синхронизации. На его коннекторе шиной используется большее число контактов, чем на остальных. У больших плат коннектор J2 предоставляется пользователю, а между коннекторами J1 и J2 может устанавливаться 95-контактный коннектор J3. Конструкция коннекторов позволяет для J2 применять специфические модификации (например, с разделяющим экраном и механическими ключами). В шине предусматривается наличие независимых источников питания +5 В, +3,3 В и ±12 В.

На базе шины Compact PCI фирмой National Instruments разработана *спецификация PXI* (PCI extensions for Instrumentation — расширение PCI для инструментальных систем) в тех же конструктивах. В шине PXI часть контактов, определенных в Compact PCI как свободные, предназначаются для дополнительных шин. Шина Trigger Bus (8 линий) звездообразно соединяет слот своего контроллера (первый после системного контроллера PCI) с остальными слотами. Шина позволяет осуществлять синхронизацию разных модулей, что зачастую требуется в измерительных системах. Для прецизионной синхронизации имеется сигнал опорной частоты 10 МГц PXI\_CLK, который звездообразно (с одинаковыми задержками распространения сигнала) разводится по слотам. В PXI определены локальные шины, предназначенные для связи соседних пар слотов. Каждая локальная шина имеет 13 линий, которые могут использоваться как для цифровых, так и аналоговых (до 48 В) сигналов. Локальные шины объединяют смежные слоты попарно (исключая слот системного контроллера), образуя цепочку. Кроме механических и электрических характеристик PXI определяет ПО модулей: основной ОС считается Windows NT/95, и модули должны поставляться с соответствующими драй-

верами. Это экономит время, необходимое для системной интеграции. Модули PXI совместимы с шиной Compact PCI, и модули Compact PCI — с шиной PXI. Однако все преимущества спецификации реализуются только при установке модулей PXI в шину PXI.

## 6.2.10. Мосты PCI

Для соединения шины PCI с другими шинами и между собой применяются специальные аппаратные средства—*мосты PCI (PCIBridge)*. *Главный мост (Host Bridge)* используется для подключения PCI к системной шине (системной памяти и процессору), *одноранговый мост (Peer-to-PeerBridge)* — для соединения двух шин PCI.

Соединения нескольких шин PCI характерно для серверов — таким образом увеличивают число подключаемых устройств. Мосты образуют иерархию шин, на вершине которой находится главная шина с нулевым номером. Главный мост чипсета системной платы может соединять центр (процессор и память) с несколькими равноранговыми шинами PCI, из которых условно главной будет шина с нулевым номером. Для подключения шин PCMCIA, CardBus, MCA, ISA/EISA, X-Bus и LPC используются специальные мосты, входящие в чипсеты системных плат или же являющиеся отдельными устройствами PCI (микросхемами). Мосты выполняют преобразование интерфейсов соединяемых ими шин, синхронизацию и буферизацию обменов данными. Мосты (включая и мосты PCI-PCI) допускают различие частот синхронизации на соединяемых ими шинах.

Каждый мост программируется — ему указываются диапазоны адресов в пространствах памяти и ввода-вывода, отведенные устройствам его шин. Если адрес ЦУ текущей транзакции на одной шине (стороне) моста относится к шине противоположной стороны, мост перенаправляет транзакцию на соответствующую шину и обеспечивает согласование протоколов шин. Таким образом, совокупность мостов PCI выполняет маршрутизацию (routing) обращений по связанным шинам. Считается, что устройство с конкретным адресом может присутствовать только на одной из шин, а на какой именно, «знают» запрограммированные мосты. Решать задачу маршрутизации призван также сигнал DEVSEL\*. Обращения, не востребованные абонентами PCI, могут быть перенаправлены, например, на шину ISA/EISA.

С мостами связаны понятия позитивного и субтрактивного декодирования адресов. Рядовые агенты PCI (устройства и мосты) отзываются только на обращения по адресам, принадлежащим областям, описанным в их конфигурационном пространстве (через базовые адреса и диапазоны памяти или ввода-вывода). Такой способ декодирования называется позитивным. Мост с *позитивным декодированием (positive decoding)* пропускает через себя только обращения, принадлежащие определенному списку, заданному в его конфигурационных регистрах. Мост с *суб-трактивным декодированием (subtractive decoding)* пропускает через себя обращения, не относящиеся к другим устройствам. Его области прозрачности формируются вычитанием (откуда и название) из общего пространства областей, описанных списком. Возможность субтрактивного декодирования имеется только у мостов определенного типа, и она является дополнением к позитивному декодированию.

Позитивное и субтрактивное декодирование относится только к обращениям, направленным в пространства памяти и ввода-вывода. Конфигурационные обращения маршрутизируются с помощью номера шины, передаваемого в циклах типа 1 (см. п. 6.2.11): каждый мост «знает» номера всех шин, его окружающих. На каждой шине PCI должно присутствовать центральное устройство, выполняющее следующие функции:

- ◆ централизованный арбитраж — прием сигналов запроса REQx# от ведущих устройств шины и предоставление им права на управление шиной сигналами GNTx#;
- ◆ «подтягивание» управляющих сигналов к высокому уровню;
- ◆ субтрактивное декодирование адресов;
- ◆ генерация конфигурационных и специальных циклов по командам процессора (с формированием индивидуальных сигналов IDSEL к адресуемому устройству PCI);
- ◆ формирование сигнала REQ64\* в момент окончания сброса в качестве признака 64-битной шины (если она таковая).

Эти функции, как правило, возлагаются на мост, соединяющий данную шину с более высокими уровнями иерархии шин PCI. Кроме выполнения этих функций, обслуживающих конкретную шину, мосты решают задачи оптимизации передачи данных между подсистемами компьютера, и от качества решения этих задач существенно зависит общая производительность системы.

Одной из особенностей применения шины PCI с ее мостовыми соединениями является возможность действительно одновременного выполнения более одного обмена данными по непересекающимся путям — Concurrent PCI Transferring или PCI Concurrency. Например, во время взаимодействия процессора с памятью ведущее устройство шины PCI может обмениваться данными с другим устройством PCI. Этот пример одновременности обмена скорее теоретический, поскольку ведущее устройство шины PCI, как правило, обменивается данными с системной памятью. Более интересный случай — обмен графического адаптера, подключенного к порту AGP (родственнику PCI, см. п. 6.3), с памятью одновременно с обменом процессора с устройством PCI или, наоборот, загрузка данных процессором в графический адаптер одновременно с обменом между ведущим устройством шины PCI и системной памятью. Одновременность требует довольно сложной логики централизованного арбитража запросов всех агентов системы и различных ухищрений в буферизации данных. Одновременность реализуется не всеми чипсетами (в описаниях она всегда специально подчеркивается) и может быть запрещена настройками CMOS Setup.

Для экономии полосы пропускания шины (уменьшения числа бесполезно занятых тактов) при обращениях к устройствам медленных шин (например, ISA) мост может выполнять *отложенные транзакции (delayed transaction)*. Работает этот механизм следующим образом. Инициатор, расположенный на «быстрой» шине (это может быть и главный мост), обращается к ЦУ на медленной шине, выставляя адрес и команду, а в командах записи еще и данные. Мост сохраняет эту информа-

цию в своих внутренних регистрах и «от имени ЦУ» вводит сигнал STOP\* (прекращение типа «повтор»), что вынуждает инициатор быстро освободить шину. Тем временем мост выполняет транзакцию с настоящим ЦУ и сохраняет у себя результат ее выполнения (для команд чтения это будут данные). Инициатор, получивший ответ «повтор», вынужден снова начать точно ту же транзакцию, и теперь на нее мост быстро ответит истинным результатом, полученным от ЦУ. В то время как мост выполняет длительную операцию обмена с медленным устройством, быстрая шина остается свободной для обслуживания других инициаторов. Если по каким-либо причинам ведущее устройство «забудет» выполнить повтор транзакции, мост должен аннулировать полученные результаты. Аннулирование выполняется по таймеру (discard timer), отсчитывающему время от начала отложенной транзакции до ее повтора. Отложенные транзакции применимы к командам обращения к портам, конфигурационному пространству, подтверждению прерывания и к обычным обращениям к памяти.

Выделение специальных циклов чтения строк, множественного чтения и записи с инвалидацией позволяет контроллеру памяти предпринять определенные меры для оптимизации данных передач. Однако эти инструкции применяют только для обращений к памяти в «чистом виде» (имеющей свойство Prefetchable). Когда инициатор обращается к источнику данных, отделенному мостом, командами чтения строки или множественного чтения, мост может считать из источника данных больше, чем инициатор заберет от него в данной транзакции. Однако в конце транзакции лишние данные в буфере моста проще всего аннулировать, поскольку до возможного последующего востребования в их реальном источнике они могут быть уже модифицированы. Более сложный мост может отслеживать и эти изменения, аннулируя лишь модифицированные данные. Обращения командами обычного чтения памяти разрешают мосту считать только точно затребованное количество данных. При этом возможности ускорения передач меньше, но не возникнет побочных эффектов от лишних чтений (чтение управляющих регистров, отображенных на память, может изменять их состояние).

Мосты, иницированные на одной стороне и обращающиеся к целевому устройству на другой стороне, могут выполнять *задержку передачи записи в память (posted write)*. При этом данные принимаются в буферы моста, и для инициатора транзакция завершится раньше, чем данные дойдут до реального получателя. Мост начнет транзакцию их доставки в удобное для другой стороны время. Порядок и количество байт, переданных получателем, обязательно должны совпадать с соответствующими данными инициатора обмена. Кроме того, сохраняется очередность операций записи и чтения. Обращение по чтению через мост вызывает принудительное освобождение буферов отложенной записи. Таким образом, инициатор имеет возможность принудительно вызвать доставку задержанных данных записи, выполняя операцию чтения «из-за моста». Записи в порты ввода-вывода откладывать имеет право только главный мост, но только для транзакций, иницированных центральным процессором. Операции чтения, естественно, выполняются сразу, поскольку без получения данных они бессмысленны.

Мосты могут преобразовывать транслируемые ими транзакции записи в память с целью оптимизации пропускной способности шины. Так, несколько последовательных транзакций записи могут объединяться в одну пакетную, в которой лишние записи могут блокироваться с помощью сигналов разрешения байтов. Например, последовательность одиночных записей двойных слов по адресам 0, 4, Ch может быть *скомбинирована* (*write combining*) в один пакет с начальным адресом 0, а во время третьей фазы данных (когда предполагается не требуемый адрес 8) все сигналы C/BE[3:0]# пассивны. Записи отдельных байтов в определенных случаях могут быть *объединены* (*byte merging*) в одну транзакцию. Так, например, последовательность записей байтов по адресам 3, 1, 0 и 2 может быть объединена в одну запись двойного слова, поскольку эти байты принадлежат одному адресуемому двойному слову. Комбинирование и объединение могут работать независимо (объединенные транзакции могут комбинироваться), однако эти преобразования не изменяют порядок следования физических записей в устройства. Наличие этих возможностей не обязательно — оно зависит от «ловкости» мостов. Цель преобразований — сократить число отдельных транзакций (каждая имеет по крайней мере одну «лишнюю» фазу адреса) и, по возможности, фаз данных. Устройства PCI должны нормально обрабатывать комбинирование записей — если устройство не допускает комбинирования, оно неправильно спроектировано. Если устройство не допускает объединения байтов, то оно в описании своей памяти должно иметь обнуленным бит Prefetchable.

С мостами PCI связано понятие *VGA Palette Snooping* — слежение за записью в палитры VGA, являющееся исключением из правила однозначной маршрутизации обращений к памяти и вводу-выводу. Графическая карта в компьютере с шиной PCI обычно устанавливается в эту шину или в порт AGP, что логически эквивалентно установке в шину PCI. На VGA-карте имеются регистры палитр (*Palette Registers*), традиционно приписанные к пространству ввода-вывода. Если графическая система содержит еще и карту смещения сигналов графического адаптера с сигналом «живого видео», перехватывая двоичную информацию о цвете текущего пиксела по шине VESA Feature Connector (снимаемую до регистра палитр), цветовая гамма будет определяться регистром палитр, размещенным на этой дополнительной карте. Возникает ситуация, когда операция записи в регистр палитр должна обрабатываться одновременно и в графическом адаптере (на шине PCI или AGP), и в карте видеорасширения, которая может размещаться даже на другой шине (в том числе и ISA). В CMOS Setup может присутствовать параметр PCI VGA Palette Snoot. При его включении запись в порты ввода-вывода по адресу регистра палитр будет вызывать транзакцию не только на той шине, на которой установлен графический адаптер, но и на других шинах. Чтение же по этим адресам будет выполняться только с самим графическим адаптером. Реализация может возлагаться на графическую карту PCI. Для этого она во время записи в регистр палитр фиксирует данные, но сигналы квитирования DEVSEL\* и TRDY\* не вырабатывает, в результате мост распространяет этот неопознанный запрос на шину ISA. В других реализациях мосту явно указывают на данное исключение, и он сам распространяет запись в регистры палитр на другие шины.



### 6.2.11. Программный доступ к конфигурационному пространству и генерация специальных циклов

Поскольку конфигурационное пространство PCI обособлено, в главный мост приходится вводить специальный механизм доступа к нему командами процессора, который «умеет» обращаться только к памяти или вводу-выводу. Этот же механизм используется и для генерации специальных циклов. Для PC-совместимых компьютеров предусмотрено два механизма, из которых в спецификации 2.2 оставлен только первый (Configuration Mechanism #1). Номер механизма, которым пользуется конкретная системная плата, можно узнать путем вызова PCI BIOS.

Конфигурационные циклы адресуются к конкретному устройству (микросхеме PCI), для которого должен быть сформирован сигнал выборки IDSEL (единичное значение). Номер функции и адрес регистра декодируется самим устройством. Поскольку сигнал IDSEL воспринимается устройством только в фазе адреса, для него используют позиционное кодирование в линиях старших битов шины AD (конфигурационное пространство всех устройств занимает лишь малую часть пространства с 32-битной адресацией). На этих линиях в фазе адреса конфигурационного обращения может быть лишь один единичный бит, остальные — нулевые. Таким образом, только одно устройство будет выбрано сигналом IDSEL.

*Для работы механизма № 1 в пространстве ввода-вывода зарезервированы 32-битные порты с адресами OCF8H и OCFCh, входящие в главный мост. Для обращения к конфигурационному пространству в порт CONFIG\_ADDRESS (RW, адрес CF8h) заносят 32-разрядный адрес, декодируемый в соответствии с рис. 6.9, а. После занесения адреса обращением к порту CONFIG\_DATA (RW, адрес CFCh) можно прочитать или записать содержимое требуемого конфигурационного регистра. В регистре CONFIG\_ADDRESS бит 31 является разрешением формирования конфигурационных и специальных циклов. В зависимости от номера шины, указанного в этом регистре, главный мост генерирует конфигурационные циклы одного из двух типов.*

Для обращения к устройству, находящемуся на нулевой шине (подключенной к главному мосту), используется *цикл типа 0* (биты 1:0=00). Главный мост декодирует поле номера устройства в позиционный код, помещаемый на линии AD[31:11]; номер адресуемой функции, адрес регистра и биты 1:0=00 передаются на шину прозрачно (рис. 6.9, б). Устройство 0 соответствует бит AD11, устройству 1 — AD12, устройству 20 — AD31. Поскольку нулевым устройством является главный мост, который и осуществляет декодирование, на шину единичное значение AD11 в цикле типа 0 не выводится. Устройства PCI, расположенные в микросхеме главного моста, могут использовать номера 21-31, для которых линий AD уже не хватает. На системной плате вход IDSEL каждого слота или микросхемы PCI-устройства соединяется со своей линией адреса. Как правило, слоты подключаются, начиная с линии AD31 (и «вниз»), микросхемы устройств PCI, расположенные на системной плате, подключаются, начиная с AD12 (и «вверх»), но может быть и иной порядок. Цикл типа 0 игнорируется всеми другими мостами, подключенными к нулевой шине.

Для обращения к устройству, находящемуся на ненулевой шине, используется *цикл типа 1*. Здесь главный мост передает адресную часть регистра CONFIG\_ADDRESS на главную шину PCI, обнуляя старшие биты (31:24) и устанавливая в битах 1:0 признак типа «01» (рис. 6.9, а). Мост, опознавший номер подключаемой им шины, передает транзакцию на эту шину, декодируя поле номера устройства в позиционный код (как это делал главный мост для своей шины) и обнуляя биты AD[1:0].

<b>3</b>	30	24	23	16	15	1	10	8	7	2	10
<b>1</b>	Резерв (0)		Номер шины		Номер устройства		Номер функции		Адрес регистра		00

а

31					1	10	8	7	2	10
Позиционный бит (только один E)		код выборки сигнала IDSEL (бит может быть единичным)				Номер функции		Адрес регистра		00

б

31	24	23	16	15	1	10	8	7	2	10
0		Номер шины		Номер устройства		Номер функции		Адрес регистра		00

Рис. 6.9. Адресация конфигурационных циклов: а — формат регистра CONFIG\_ADDRESS, б — формат адреса на шине в цикле типа 0, в — формат адреса на шине в цикле типа 1

Если конфигурационный цикл не воспринимается ни одним из устройств, мосты могут эту ситуацию обрабатывать двояко: фиксировать отсутствие устройства (сработает *Master Abort*) или же выполнять операции вхолостую. Однако в любом случае чтение конфигурационного регистра несуществующего устройства (функции) должно возвращать значение FFFFFFFFh (это будет безопасной информацией, поскольку даст недопустимое значение идентификатора устройства). Если главный мост подключает несколько равноранговых шин (peer buses), то одна из них назначается условно главной, так что вышеописанная логика сохраняется. Чтобы не заботиться об изучении реальной топологии шин, для конфигурационных обращений программам удобно использовать вызовы PCI BIOS, которые имеются для всех режимов процессора (оптимизировать быстродействие здесь не приходится, поскольку эти обращения выполняются не так уж и часто).

*Специальный цикл* генерируется при записи в CONFIG\_DATA, когда в регистре CONFIG\_ADDRESS все биты [15:8] единичные, [7:0] нулевые, номер шины, на которой формируется цикл, задается битами [23:16]. В специальном цикле адресная информация не передается (он широкоэмиттерный), но путем задания номера шины можно управлять его распространением. Если хост генерирует специальный цикл с нулевым адресом шины, то на главную шину этот цикл выйдет как цикл типа 0 и всеми остальными мостами распространяться не будет. Если требуется полная широ-

ковещательность, то хост должен записать ненулевой адрес шины; тогда на главную шину специальный цикл выйдет с типом 1, и все остальные мосты пропустят его на свои шины. Специальный цикл 1, генерируемый ведущим устройством шины, действует только на шине этого устройства и не распространяется через мосты. Если требуется полная ширококовещательность, это ведущее устройство должно генерировать специальный цикл посредством записей в регистры `CONG I G_ADDRES 5` и `CONFIG_DATA`, задавая ненулевой номер шины.

*Для работы механизма №2в* пространстве ввода-вывода зарезервированы два 8-битных порта с адресами `OCF8H` и `OCFAh`, входящие в главный мост. Этот механизм использует отображение конфигурационного пространства устройств PCI на область `COOO-CFFF` пространства ввода-вывода. Поскольку этой области (4 К портов) недостаточно для отображения конфигурационного пространства всех устройств всех шин PCI, формирование адреса выполняется весьма замысловатым образом. В регистре `CSE` (Configuration Space Enable) с адресом `OCFSh` биты 7:4 являются ключом разрешения отображения: при нулевом ключе область `COOO-CFFFh` остается нормальной частью пространства ввода-вывода, а при ненулевом — отображается на конфигурационное пространство. Биты [3:1] несут номер функции, к пространству которой адресуются обращения. Единичное значение бита 1 вызывает формирование специального цикла (Special Cycle Enable, SCE). При обращении к конфигурационному пространству устройств нулевой шины чтение или запись двойного слова в порт по адресу `COOO-CFFCh` генерирует конфигурационный цикл, в котором из адреса порта биты [2:7] поступают на шину `AD[2:7]` как индекс регистра конфигурационного пространства, а биты [11:8] декодируются в позиционный код выбора устройства (линии `IDSEL`) на линиях `AD [31:16]`. Номер функции на линии `AD[10:8]` поступает из регистра `CSE`, линии `AD[1:0]` нулевые. Для обращения к устройствам ненулевой шины служит *регистр перенаправления* (Forward Register) с адресом `OCFAh`, в который помещают номер шины (по сбросу этот регистр обнуляется). Если номер шины ненулевой, то генерируется цикл типа 1 (см. рис. 6.9, в), в котором номер функции поступает из регистра `CSE`, младшие 4 бита номера устройства поступают с битов адреса (`AD15=0`), а номер шины — из регистра перенаправления (биты `AD[1:0]=01` и `AD[31:24]=0` формируются аппаратно).

Для генерации специального цикла по этому механизму в регистре `CSE` устанавливается ненулевой ключ, номер функции 111 и `SCE=1`, после чего выполняется запись по адресу порта `CFOOh`. В зависимости от содержимого регистра перенаправления будет сгенерирован специальный цикл типа 0 или типа 1 (см. выше).

## 6.2.12. Конфигурирование устройств

В стандарт заложены возможности автоматического конфигурирования системных ресурсов (пространств памяти и ввода-вывода и линий запроса прерываний). Автоматическое конфигурирование устройств (выбор адресов и прерываний) поддерживается средствами BIOS и ориентировано на технологию PnP. Стандарт PCI определяет для каждого слота конфигурационное пространство размером до 256 регистров (8-битных), не приписанных ни к пространству памяти, ни к

пространству ввода-вывода. Доступ к ним осуществляется по специальным циклам шины *Configuration Read* и *Configuration Write*, вырабатываемым с помощью одного из вышеописанных механизмов. В этом пространстве есть области, обязательные для всех устройств, и специфические. Конкретное устройство может иметь регистры не во всех адресах, но должно поддерживать нормальное завершение для адресуемых к ним операций. При этом чтение несуществующих регистров должно возвращать нули, а запись выполняться как холостая операция. После аппаратного сброса (или при включении питания) устройства PCI не отвечают на обращения к пространству памяти и ввода-вывода, они доступны только для операций конфигурационного считывания и записи. В этих операциях устройства выбираются по индивидуальным сигналам IDSEL и сообщают о потребностях в ресурсах и возможных вариантах конфигурирования. После распределения ресурсов, выполняемого программой конфигурирования (во время теста POST), в конфигурационные регистры устройства записываются параметры конфигурирования. Только после этого к устройствам становится возможным доступ по командам обращения к памяти и портам ввода-вывода. Для того чтобы всегда можно было найти работоспособную конфигурацию, все ресурсы, занимаемые картами, должны быть перемещаемыми в своих пространствах. Для многофункциональных карт каждая функция должна иметь собственное конфигурационное пространство.

Конфигурационное пространство устройства начинается со стандартного заголовка, в котором содержатся идентификаторы производителя, устройства и его класса, а также описание требуемых и занимаемых системных ресурсов. После заголовка могут располагаться регистры, специфичные для устройства; они могут занимать адреса конфигурационного пространства в пределах 40-FFh.

Формат заголовка приведен на рис. 6.10. Серым цветом здесь выделены поля, обязательные для всех устройств.

Перечисленные ниже *поля идентификации* допускают только чтение.

- ◆ *Device ID* — идентификатор устройства, назначаемый производителем.
- ◆ *Vendor ID* — идентификатор производителя микросхемы PCI, назначенный PCI SIG. Идентификатор FFFFh является недопустимым; это значение должно возвращаться при чтении конфигурационного пространства несуществующего устройства.
- ◆ *Revision ID* — версия продукта, назначенная производителем. Используется как расширение поля *Device ID*.
- ◆ *Header Type* — тип заголовка (биты 6:0), определяющий формат ячеек в диапазоне 10-3Fh и несущий признак многофункционального устройства (если бит 7=1). На рисунке приведен формат заголовка типа 0, относящийся именно к устройствам PCI. Тип 01 относится к мостам PCI-PCI; тип 02 относится к мостам для CardBus.
- ◆ *Class Code* — код класса, определяющий основную функцию устройства, а иногда и его программный интерфейс (см. п. 6.2.13). Старший байт (адрес 0Bh) определяет базовый класс, средний — подкласс, младший — программный интерфейс (если он стандартизован).

31	24	23	16	15	8	7	0	
Device ID				Vendor ID				00h
Status				Command				04h
ClassCode						Revision ID		08h
BIST		Header Type		Latency Timer		Cache Line Size		0Ch
Base Address Registers								10h 24h
CardBus CIS Pointer								28h
Subsystem ID				Subsystem Vendor ID				2Ch
Expansion ROM Base Address								30h
Резерв						Capabilities Pointer		34h
Резерв								38h
Max_Lat		Min_Gnt		Interrupt Pin		Interrupt Line		3Ch

Рис. 6.10. Формат заголовка конфигурационного пространства устройства PCI

Остальные поля заголовка являются *регистрами устройств*, допускающими как запись, так и чтение.

- ◆ **Command (RW)** — регистр команд, управляющий поведением устройства на шине PCI. Регистр допускает как запись, так и чтение. После аппаратного сброса все биты регистра (кроме специально оговоренных исключений) обнулены. Назначение бит регистра команд:
  - бит 0—10 **Space** — разрешение ответа на обращения к пространству вво да-вывода;
  - бит 1 — **Memory Space** — разрешение ответа на обращения к пространству памяти;
  - бит 2 — **Bus Master** — разрешение работы инициатором (в режиме управления шиной);
  - бит 3— **Special Cycles** — разрешение реакции на специальные циклы;
  - бит 4 — **Memory Write& and Invalidate enable**— разрешениеиспользовать команды «запись с инвалидацией» при работе инициатором (если бит обну лен, то вместо этих команд должна использоваться обычная запись в память);
  - бит 5 — **VGA palette snoop** — разрешение слежения за записью в регистр палитр;

- бит 6 — Parity Error Response — разрешение нормальной реакции (вырабатывать сигнал **PERR#**) на обнаруженную ошибку паритета (если бит обнулен, то устройство должно только фиксировать ошибку в регистре состояния, в то время как генерация бита паритета устройством выполняется всегда);
  - бит 7 — Stepping Control — возможность пошагового переключения (address/data stepping) линий (если устройство никогда этого не делает, бит регистра «запаян» в «0», если делает всегда — в «1», устройство с такой возможностью по сбросу устанавливает этот бит в «1»);
  - бит 8 — SERR# Enable — разрешение генерации сигнала ошибки SERR# (ошибка паритета адреса сообщается, когда этот бит и бит 6=1);
  - бит 9 — Fast Back-to-Back Enable (необязательный) — разрешение ведомому устройству использовать быстрые смежные обращения к разным устройствам (если бит обнулен, быстрые обращения допустимы лишь для транзакций с одним агентом);
  - биты 10-15 — резерв.
- ◆ Status — регистр состояния, допускающий кроме чтения еще и запись. Однако запись выполняется специфично — с ее помощью можно только обнулять биты, но не устанавливать. Биты, помеченные как RO, допускают только считывание. При записи в позиции обнуляемых бит устанавливаются единичные значения. Назначение бит регистра состояния:
- биты 0-3 — резерв;
  - бит 4 — Capable List (RO, необязательный) — указание на наличие указателя новых возможностей (смещение 34h в заголовке);
  - бит 5 — 66 MHz Capable (RO, необязательный) — поддержка частоты 66 МГц;
  - бит 6 — резерв;
  - бит 7 — Fast Back-to-Back Capable (RO, необязательный) — поддержка быстрых смежных транзакций (fast back-to-back) с разными устройствами;
  - бит 8 — Master Data Parity Error (только для устройств с прямым управлением) — устанавливается, когда устройство с установленным битом 6 в регистре команд, являясь инициатором, само ввело (при чтении) или обнаружило (при записи) сигнал **PERR#**;
  - биты 10:9 — DEVSEL Timing — скорость выборки: 00 — быстрая, 01 — средняя, 10 — низкая (определяет самую медленную реакцию DEVSEL\* на все команды, кроме Configuration Read и Configuration Write); бит 11 — Signaled Target Abort — устанавливается целевым устройством, когда оно отвергает транзакцию;
  - бит 12 — Received Target Abort — устанавливается инициатором, когда он обнаруживает отвергнутую транзакцию;
  - бит 13 — Received Master Abort — устанавливается ведущим устройством, когда оно отвергает транзакцию (кроме специального цикла);

- бит 14 — Si gnaled System Error — устанавливается устройством, подавшим сигнал SERR#;
- бит 15 — Detected Parity Error—устанавливается устройством, обнаружившим ошибку паритета.
- ◆ Cache Line Size (RW) — размер строки кэша (0-128, допустимые значения  $2^n$ , иные трактуются как 0). По этому параметру инициатор определяет, какой командой чтения воспользоваться (обычное чтение, чтение строки или множественное чтение). Ведомое устройство использует этот параметр для поддержки пересечения границ строк при пакетных обращениях к памяти. По сбросу регистр обнуляется.
- ◆ Latency Timer (RW) — значение таймера задержки (см. п. 6.2.4) в тактах шины. Часть битов может не допускать изменения (обычно младшие три бита неизменны, так что таймер программируется с дискретностью в 8 тактов).
- ◆ BIST (RW) — регистр управления встроенным самотестированием. Назначение бит регистра:
  - бит 7 — возможность BIST;
  - бит 6 — запуск теста: запись единицы иницирует тест, по окончании устройство сбрасывает бит (тест должен быть завершен не более чем за 2 с);
  - биты 5:4 — резерв (0);
  - биты 3:0 — код завершения теста: 0 — тест прошел успешно.
- ◆ Card Bus CIS Pointer (необязательный) — указатель на структуру описателя Card Bus для комбинированного устройства PCI+Card Bus.
- ◆ Interrupt Line( RW) — номер входа контроллера прерывания для используемой линии запроса (0-15 — IRQ0-IRQ15, 255 — неизвестный или не используется).
- ◆ Interrupt Pin (RO) — контакт, используемый для запроса прерывания: 0 — не используется, 1 - INTA#, 2 - INTB#, 3 - INTC#, 4 - INTD#, 5-FFh - резерв.
- ◆ Min\_GNT (RO) — минимальное время, на которое ведущему устройству должно предоставляться управление шиной из расчета на частоту 33 МГц, в интервалах по 0,25 мкс.
- ◆ Max\_Lat (RO) — максимально допустимая задержка предоставления ведущему устройству доступа к шине, в интервалах по 0,25 мкс (0 — нет специальных требований).
- ◆ Subsystem ID (задается производителем) и Subsystem Vendor ID (производитель получает в PCI SIG) — идентификаторы, позволяющие точно идентифицировать карты и устройства (в системе могут быть установлены несколько карт с совпадающими идентификаторами устройства и производителя Device ID и Vendor ID). В поле 2Ch ставится идентификатор производителя карты PCI (может совпадать со значением в поле 0, если фирма выпускает и микросхемы, и карты).

- ◆ Capability Pointer (CAP\_PTR)—указатель на список возможностей (свойств), описанных в конфигурационном пространстве (биты 1:0=00). Каждый элемент списка начинается с байта идентификатора типа элемента (CAP\_ID, определенного PCI SIG), за которым следует указатель на следующий элемент списка (нулевой указатель является признаком конца списка), после чего расположены байты описаний самих свойств. Все элементы выравниваются по границе двойного слова.
- ◆ Base Address Registers — базовые адреса областей памяти и портов ввода-вывода. Для областей памяти бит 0=0. Биты 2:1 — тип: 00 — располагаются в 32-битном адресном пространстве, 10 — в 64-битном (в этом случае регистр расширяется следующим за ним 4-байтным словом), 01 и 11 — резерв (01 в прежних версиях предназначались для памяти в пределах первого мегабайта). Бит 3 (Prefetchable) устанавливается, если чтение памяти не приводит к побочным эффектам, все байты считываются независимо от сигналов BE[3:0]# и отдельные записи мостом могут быть объединены (то есть это память в чистом виде); в иных случаях бит обнулен. Биты 31:4 — базовый адрес памяти. Для портов бит 0=1; бит 1 -0 (резерв); биты 31:2 — базовый адрес блока портов. Конфигурирующая программа может определить размеры требуемых областей. Для этого после аппаратного сброса она должна считать и сохранить значения базовых адресов (это будут адреса по умолчанию), записать в каждый регистр FFFFFFFFh и снова считать их значение. В полученных словах нужно обнулить биты декодирования типа (биты 3:0 для памяти и биты 1:0 для ввода-вывода), инвертировать и инкрементировать полученное 32-битное слово — результатом будет длина области (для портов биты 31:16 игнорировать). Метод подразумевает, что длина области выражается числом 2<sup>n</sup> и область выровнена естественным образом.
- ◆ Expansion ROM Base Address—базовый адрес ПЗУ программной поддержки карты. Бит 0 — разрешение использования ПЗУ; биты 1-10 — резерв; биты 11-31 — базовый адрес. Размер ПЗУ определяется так же, как и в регистрах базовых адресов (см. выше). Обращение к ПЗУ возможно лишь при разрешенном использовании памяти (бит 1 в регистре команд).

### 6.2.13. Классы устройств PCI

Важной частью спецификации PCI является классификация устройств и указание кода класса в его конфигурационном пространстве (3 байта Class Code). Старший байт определяет *базовый класс*, средний — *подкласс*, младший — *программный интерфейс* (если он стандартизован). Код класса позволяет идентифицировать наличие определенных устройств в системе, это может быть сделано с помощью PCI BIOS. Для стандартизованных устройств (например, 01:01:80 — контроллер IDE или 07:00:01 — последовательный порт 16450) «заинтересованная» программа может найти требуемое устройство и выбрать подходящий вариант драйвера. Классификатор определяет организация PCI SIG, он регулярно обновляется на сайте [www.pcisig.com](http://www.pcisig.com). Нулевые значения полей, как правило, дают самые неопределенные описания. Значение подкласса 80h относится к «иным устройствам».



## 6.2.14. PCI BIOS

Для облегчения взаимодействия с устройствами PCI имеются дополнительные функции BIOS, доступные как из реального, так и защищенного режима работы процессора. Эти функции, предназначенные для работы с конфигурационным пространством и генерации специальных циклов PCI, приходится поддерживать потому, что циклы конфигурационных обращений, как и специальный цикл, выполняются специфическим образом (см. выше). Остальные функции взаимодействия с устройствами через их пространства памяти и ввода-вывода, а также обработка прерываний, в поддержке со стороны BIOS не нуждаются, поскольку выполняются непосредственно командами процессора. Функция проверки наличия PCI BIOS позволяет определить доступные механизмы генерации этих особых циклов, и, зная их работу, программа в дальнейшем может и не пользоваться вызовами PCI BIOS.

Программы с помощью функций PCI BIOS могут искать интересующие их устройства по идентификаторам или кодам класса. Если стоит задача «переучета» установленных устройств, то она решается чтением конфигурационной информации по всем функциям всех устройств всех шин — это быстрее, чем перебирать все возможные сочетания идентификаторов или классов кодов. Для найденных устройств программы должны определять реальные настройки чтением регистров конфигурационного пространства, учитывая возможность перемещения ресурсов по всему пространству и даже между пространствами памяти и ввода-вывода.

Функции PCI BIOS для 16-битного интерфейса реального режима, V86 и 16-битного защищенного режима, вызываются через прерывание `Int 1Ah`. Номер функции задается при вызове в регистре `AX`. Возможна и программная имитация прерывания дальним вызовом по физическому адресу `000FFE6EH` (стандартная точка входа в обработчик `Int 1Ah`) с предварительным занесением в стек регистра флагов. Для 32-разрядных вызовов защищенного режима все эти же функции вызываются через точку входа, найденную через каталог 32-разрядных сервисов, при этом назначение входных и выходных регистров и флага `CF` сохраняется. До использования 32-разрядного интерфейса следует сначала найти его каталог и убедиться в наличии сервисов PCI. Вызовы требуют глубокого стека (до 1024 байт). Функции PCI BIOS:

- ◆ `AX = B10h` — проверка присутствия PCI BIOS;
- ◆ `AX = B102h` — поиск устройства по идентификатору;
- ◆ `AX = B103h` — поиск устройства по коду класса;
- ◆ `AX = B106h` — генерация специального цикла PCI;
- ◆ `AX = B109`, `B10A` и `B10Ah` — чтение байта, слова и двойного слова конфигурационного пространства устройства PCI;
- ◆ `AX = B10B`, `B10C`, `B10Dh` — запись байта, слова и двойного слова конфигурационного пространства устройства PCI.

## 6.2.15. Разработка собственных устройств PCI

При рассмотрении протокола PCI становится ясно, что разработка собственных PCI-устройств на логике малой и средней степени интеграции — занятие неблагодарное. Собственно протокол шины не так уж и сложен, но реализация требований к конфигурационным регистрам проблематична. Серийные устройства PCI, как правило, являются однокристальными — в одной микросхеме размещается и интерфейсная, и функциональная части устройства. Разработка таких микросхем весьма дорогостояща и имеет смысл лишь с перспективами массового выпуска. Для создания отладочных образцов и мелкосерийных изделий ряд фирм выпускают интерфейсные микросхемы PCI различного назначения. Со стороны PCI практически все эти микросхемы поддерживают одиночные целевые транзакции (*target transactions*), совершенные модели допускают и пакетные циклы. Более сложные микросхемы выполняют и функции ведущего устройства шины, организуя каналы DMA для обмена с системной памятью. Обмены по этим каналам могут инициироваться как программно со стороны хоста (*host initiated DMA*), так и с периферийной стороны микросхемы (*target initiated DMA*), в зависимости от возможностей микросхем. С периферийной стороны встречаются интерфейсы для подключения периферийных микросхем, микроконтроллеров и распространенных семейств микропроцессоров, универсальных и сигнальных. Довольно широкий выбор микросхем представлен на сайте [www.plxtech.com](http://www.plxtech.com), этой темой занимаются и иные фирмы.

Интересно решение построения интерфейса PCI на конфигурируемой логике FPGA (*Field Programmable Gate Array* — программируемый массив вентилей). Здесь PCI-ядро, а также функции целевого и ведущего устройств занимают 10-15 тысяч вентилей в зависимости от требуемых функций (см. [www.xilinx.com](http://www.xilinx.com), [www.altera.com](http://www.altera.com)). Микросхемы FPGA выпускаются на 20, 30 и 40 тысяч вентилей — оставшаяся часть может быть использована для реализации функциональной части устройства, буферов FIFO и т. п.

Быстро перевести разработки с шины ISA на PCI можно с помощью микросхем-мостов PCI-ISA (см., например, [www.iss-us.com](http://www.iss-us.com)).

## 6.3. Магистральный интерфейс AGP

В настоящее время самой быстрой универсальной шиной расширения является PCI, имеющая при тактовой частоте 66 МГц и разрядности 32 бит пиковую пропускную способность 264 Мбайт/с. Одним из главных потребителей пропускной способности шины является графический адаптер. По мере увеличения разрешения и глубины цвета требования к пропускной способности шины, связывающей дисплейный адаптер с памятью и центральным процессором компьютера, повышаются. Одно из решений состоит в уменьшении потока графических данных, передаваемых по шине. Для этого графические платы снабжают ускорителями и увеличивают объем видеопамати, которой пользуется ускоритель при выполнении построений. В результате поток данных в основном циркулирует внутри графиче-

ской карты, слабо нагружая внешнюю шину. Однако при трехмерных построениях ускорителю становится тесно в ограниченном объеме локальной памяти графического адаптера, и его поток данных снова «выплескивается» на внешнюю шину.

Фирма Intel на базе шины PCI 2.1 разработала стандарт подключения графических адаптеров — AGP (Accelerated Graphic Port — ускоренный графический порт). Первая версия стандарта вышла в 1996 году, в настоящее время действует версия 2.0 (1998 года), отличающаяся от первой в основном введением нового режима передачи 4x. В конце 2000 г. Intel опубликовала проект спецификации AGP8X, которая рассматривается как отдельная спецификация, а не просто развитие предыдущих. Ее особенности отметим в конце раздела.

Порт AGP представляет собой 32-разрядную шину с тактовой частотой 66 МГц, большая часть сигналов позаимствована из шины PCI. Однако в отличие от PCI, порт AGP представляет собой двухточечный интерфейс, соединяющий графический адаптер с памятью и системной шиной процессора напрямую логикой и каналами данных чипсета системной платы, не пересекаясь с «узким местом» — шиной PCI. Поначалу планировался переход на тактовую частоту 100 МГц, но и в спецификации AGP 2.0, и в предлагаемом варианте AGP8X фигурирует лишь одна частота — 66,6 МГц (нынешний предел и для шины PCI). «Ускоренность» порта обеспечивается следующими факторами:

- ◆ конвейеризацией обращений к памяти;
- ◆ удвоенной (2x) или учетверенной (4x) частотой передачи данных (относительно тактовой частоты порта);
- ◆ демultipлексированием шин адреса и данных.

Идею конвейеризации обращений к памяти иллюстрирует рис. 6.11, где сравниваются обращения к памяти PCI и AGP. В PCI во время реакции памяти на запрос шина простаивает (но не свободна). Конвейерный доступ AGP позволяет в это время передавать следующие запросы, а потом получить поток ответов.

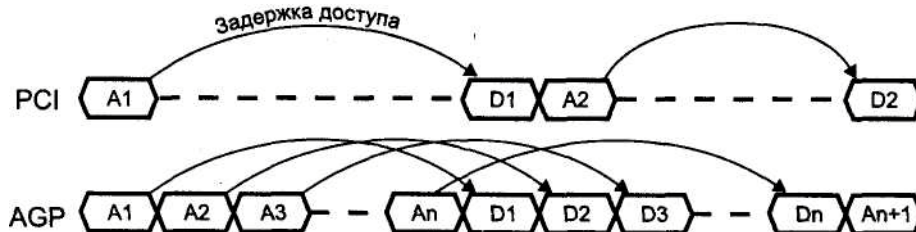


Рис. 6.11. Циклы обращения к памяти PCI и AGP

Удвоение и учетверение частоты передачи данных обеспечивает при частоте 66 МГц пропускную способность до 533 (2x) и 1066 Мбайт/с (4x), что для 32-битной шины несколько неожиданно. В этих режимах блоки данных передаются как по фронту, так и по спаду стробирующего сигнала (как в ATA Ultra DMA). Режимы 2x и 4x могут использоваться, лишь если их поддерживают и графический адаптер, и системная плата.

### 6.3. Магистральный интерфейс AGP

213

Демultipлексирование (разделение) шины адреса и данных сделано несколько необычным образом. С целью экономии числа интерфейсных линий шину адреса и команды в демultipлексированном режиме AGP представляют всего 8 линий SBA (SideBand Address), по которым команда, адрес и значение длины передачи передаются последовательно за несколько тактов. Поддержка демultipлексированной адресации не является обязательной для устройства AGP, поскольку имеется альтернативный способ подачи адреса по шине AD.

Порт AGP предназначен только для интеллектуального графического адаптера, имеющего SD-ускоритель (для краткости здесь этот адаптер будем

называть просто ускорителем). Системная логика порта AGP отличается сложным контроллером памяти, который выполняет глубокую буферизацию и высокопроизводительное обслуживание запросов AGP (от адаптера) и других своих клиентов — центрального процессора (одного или нескольких) и шины PCI. AGP может реализовать всю пропускную способность 64-битной системы памяти компьютера на процессорах Pentium и выше. При этом возможны конкурирующие обращения к памяти как со стороны процессора, так и со стороны мостов шин PCI. Фирма Intel ввела поддержку AGP в чипсеты для процессоров P6, однако нет «противопоказаний» для применения AGP и с Pentium.

Отметим, что многие преимущества AGP носят потенциальный характер и могут быть реализованы лишь при поддержке аппаратных средств графического адаптера и ПО. Графический адаптер с интерфейсом AGP может реально вести себя по-разному:

- ◆ не задействовать конвейеризацию, а использовать только быструю запись PCI (Fast Write);
- ◆ не работать с текстурами, расположенными в системной памяти, но использовать более быстрый обмен данными между памятью и локальным буфером;
- ◆ использовать оптимальное распределение текстур между локальной и системной памятью, избегая конфликтов с обращениями к памяти от процессора.

Порт AGP содержит практически полный набор сигналов шины PCI и дополнительные сигналы AGP. Устройство, подключаемое к порту AGP, может предназначаться как исключительно для операций AGP, так и быть комбинацией AGP+PCI. Устройство обязательно должно выполнять функции ведущего устройства AGP (иначе порт AGP для него теряет смысл) и функции ведомого устройства PCI со всеми его атрибутами (конфигурационными регистрами и т. п.); дополнительно оно может быть и ведущим устройством PCI. Для транзакций в режиме AGP ему доступно только системное ОЗУ. В режиме PCI обращения возможны как к пространству памяти, так и к пространству ввода-вывода и конфигурационному пространству.

*Транзакции в режиме PCI*, инициируемые ускорителем, начинаются с подачи сигнала FRAME\* и выполняются обычным для PCI способом (см. п. 6.2). Заметим, что при этом на все время транзакции шина AD занята, причем транзакции чтения памяти занимают шину на большее число тактов, чем транзакции записи, — после подачи адреса неизбежны такты ожидания на время доступа к памяти. Запись на

шине происходит быстрее — данные записи задатчик посылает сразу за адресом, а на время доступа к памяти они «оседают» в буфере контроллера памяти. Контроллер памяти позволяет завершить транзакцию и освободить шину до физической записи в память. Обращения со стороны процессора (или задатчиков шины PCI) обрабатываются адаптером так же, как обычным устройством PCI.

*Конвейерные транзакции AGP* (команды AGP) инициируются только ускорителем; логикой AGP они ставятся в очереди на обслуживание и исполняются в зависимости от приоритета, порядка поступления запросов и готовности данных. Эти транзакции могут быть адресованы ускорителем только к системному ОЗУ. Если устройству AGP требуется обратиться к локальной памяти каких-либо устройств PCI, то оно должно выполнять эти транзакции в режиме PCI. Транзакции, адресованные к устройству AGP, обрабатываются им как ведомым устройством PCI, однако имеется возможность *быстрой записи* в локальную память *FWXFast Write*), в которой данные передаются на скорости AGP (2x или 4x), и управление потоком их передач ближе к протоколу AGP, нежели PCI. Транзакции FW обычно инициируются процессором и предназначены для принудительного «заталкивания» данных в локальную память ускорителя.

Концепцию конвейера AGP иллюстрирует рис. 6.12. Порт AGP может находиться в одном из четырех состояний:

- ◆ *IDLE* — покой;
- ◆ *DATA* — передача данных конвейеризированных транзакций;
- ◆ *AGP* — постановка в очередь команды AGP;
- ◆ *PCI* — выполнение транзакции в режиме PCI.

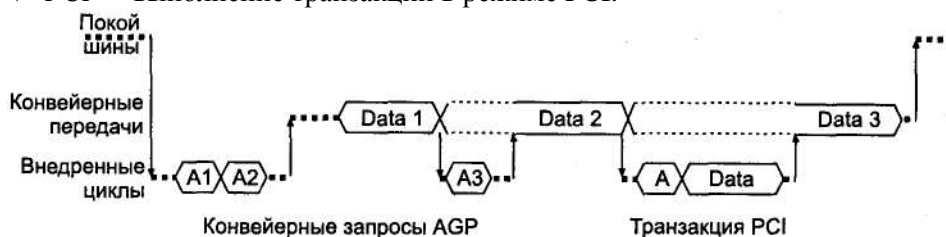


Рис. 6.12. Конвейер AGP

Из состояния покоя *IDLE* порт может вывести запрос транзакции PCI (как от ускорителя, так и с системной стороны) или запрос AGP (только от ускорителя). В состоянии *PCI* транзакция PCI выполняется целиком, от подачи адреса и команды до завершения передачи данных. В состоянии *AGP* ведущее устройство передает только команду и адрес для транзакции (по сигналу PIPE\* или через порт SBA), ставящейся в очередь; несколько запросов могут следовать сразу друг за другом. В состояние *DATA* порт переходит, когда у него в очереди имеется необслуженная команда, готовая к исполнению. В этом состоянии происходит передача данных для команд, стоящих в очереди. Это состояние может прерываться запросами *PCI* (для выполнения целой транзакции) и *AGP* (для постановки в очередь

новой команды), но прерывание возможно только на границах данных транзакций AGP. Когда порт AGP обслужит все команды, он снова переходит в состояние покоя. Все переходы происходят под управлением арбитра порта AGP, реагирующего на поступающие запросы (REQ# от ускорителя и внешние обращения от процессора или других устройств PCI) и ответы контроллера памяти. Транзакции AGP некоторыми моментами отличаются от транзакций PCI.

- ◆ Фаза данных отделена от фазы адреса, чем и обеспечивается конвейеризация.
- ◆ Используется собственный набор команд.
- ◆ Транзакции адресуются только к системной памяти, используя то же пространство физических адресов, что и PCI. Транзакции могут иметь длину, кратную 8 байтам, и начинаться только по 8-байтной границе. Транзакции чтения иного размера должны выполняться только в режиме PCI; транзакции записи могут использовать сигналы C/BE[3:0]# для маскирования лишних байтов.
- ◆ Длина транзакции явно указывается в запросе.
- ◆ Конвейерные запросы не гарантируют когерентность памяти и кэша. Для операций, требующих когерентности, должны использоваться транзакции PCI.

Возможны два способа подачи команд AGP (постановки запросов в очередь), из которых в текущей конфигурации выбирается один, причем изменение способа «на ходу» не допускается.

- ◆ Запросы вводятся по шине AD[31:0] и C/BE[3:0] с помощью сигнала PIPE#, по каждому фронту CLK ведущее устройство передает очередное двойное слово запроса вместе с кодом команды.
- ◆ Команды подаются через внеполосные (sideband) линии адреса SBA[7:0]. «Внеполосность» означает, что эти сигналы используются независимо от занятости шины AD. Синхронизация подачи запросов зависит от режима (1x, 2x или 4x).

При подаче команд по шине AD во время активности сигнала PIPE\* код команды AGP (CCCC) кодируется сигналами C/BE[3:0], при этом на шине AD помещается начальный адрес (на AD[31:3]) и длина п (на AD[2:0]) запрашиваемого блока данных. Определены следующие команды:

- ◆ 0000 (Read) — чтение из памяти (п+1) учетверенных слов (по 8 байт) данных, начиная с указанного адреса;
- ◆ 0001 (HP Read) — чтение с высоким приоритетом;
- ◆ 0100 (Write) — запись в память;
- ◆ 0101 (HP Write) — запись с высоким приоритетом;
- ◆ 1000 (Long Read) — «длинное» чтение (п+1)x4 учетверенных слов (до 256 байт данных);
- ◆ 1001 (HP Long Read) — «длинное» чтение с высоким приоритетом;
- ◆ 1010 (Flush) — очистка, выгрузка данных всех предыдущих команд записи по адресам назначения (на порте AGP выглядит как чтение, возвращающее произвольное учетверенное слово в качестве подтверждения исполнения; адрес и длина, указанные в запросе, значения не имеют);

- ◆ 1100 (Fence) — установка «ограждений», позволяющих низкоприоритетному потоку записей не пропускать чтения;
- ◆ 1101 (Dual Address Cycle, DAC) — двухадресный цикл для 64-битной адресации: в первом такте по AD передается младшая часть адреса и длина запроса, а во втором — старшая часть адреса (по AD) и код исполняемой команды (по C/BE[3:0]).

При внеполосной подаче команд по шине SBA[7:0] передаются 16-битные посылки четырех типов. Каждая посылка передается за два приема, по фронту и спаду синхросигнала. Тип посылки кодируется старшими битами:

- ◆ тип 1: 0AAA AAAA AAAA ALLL — поле длины (LLL) и младшие биты адреса (A[14:03]);
- ◆ тип 2: 10CC CCRA AAAA AAAA — код команды (CCCC) и средние биты адреса (A[23:15]);
- ◆ тип 3: 110R AAAA AAAA AAAA — старшие биты адреса (A[35:24]);
- ◆ тип 4: 1110 AAAA AAAA AAAA — дополнительные старшие биты адреса, если требуется 64-битная адресация.

Посылка из всех единиц является пустой командой (NOP); они посылаются в покое шины SBA. Биты «R» зарезервированы. Посылки типов 2,3 и 4 являются «липкими» (sticky) — значения, ими определяемые, сохраняются до введения новой посылки того же типа. Постановку команды в очередь инициирует посылка типа 1, задающая длину транзакции и ее младшие адреса, — код команды и остальная часть адреса должны быть определены ранее введенными посылками типов 2-4. Такой способ очень экономно использует такты шины для подачи команд при пересылках массивов. Синхронизация данных на SBA зависит от режима порта.

- ◆ В режиме 1x каждая часть передается по фронту CLK; начало посылки (старшая часть) определяется по получению байта, отличного от 1111111b, по последующему фронту передается младшая часть. Очередная команда может вводиться за каждую пару тактов CLK (когда код команды и старший адрес уже введены).
- ◆ В режиме 2x для SBA используется отдельный строб SB\_STB, по его спаду передается старшая часть, а по последующему фронту — младшая. Частота этого строба (но не фаза) совпадает с CLK, так что очередная команда может вводиться в каждом такте CLK.
- ◆ В режиме 4x используется еще и дополнительный (инверсный) строб SB\_STB#. Старшая часть фиксируется по спаду SB\_STB, а младшая — по последующему спаду SB\_STB#. Частота стробов в два раза выше, чем CLK, так что в каждом такте CLK может вводиться пара команд.

Конечно, полный цикл введения команд (с посылками всех четырех типов) с учетом посылки NOP оказывается больше — 10, 5 и 2,5 тактов частоты CLK для режимов 1x, 2x и 4x соответственно.

В ответ на полученные команды порт AGP выполняет *передачи данных*, причем фаза данных AGP явно не привязана к фазе команды/адреса. Фазы данных вводит порт AGP (системная логика), исходя из порядка ранее пришедших к нему команд от ускорителя.

*Передачи данных А GP* выполняются, когда шина находится в состоянии *DATA*. Как говорилось выше, фаза данных AGP явно не привязана к фазе команды/адреса. Фазы данных вводит порт AGP (системная логика), исходя из порядка ранее пришедших к нему команд от ускорителя. Ускоритель узнает о назначении шины AD в последующей транзакции по сигналам ST[2:0] (действительны только во время сигнала GNT#, коды 100-110 зарезервированы):

- ◆ 000 — ведущему устройству будут передаваться данные низкоприоритетного запроса чтения, ранее поставленного в очередь (или выполняется очистка);
- ◆ 001 — ведущему устройству будут передаваться данные высокоприоритетного запроса чтения;
- ◆ 010 — ведущее устройство должно будет предоставлять данные низкоприоритетного запроса записи;
- ◆ 0Н — ведущее устройство должно будет предоставлять данные высокоприоритетного запроса записи;
- ◆ 111 — ведущему устройству разрешается поставить в очередь команду AGP (сигналом PIPE#) или начать транзакцию PCI (сигналом FRAME\*).

Ускоритель узнает лишь тип и приоритет команды, результаты которой последуют в данной транзакции. Какую именно команду из очереди обрабатывает порт, ускоритель определяет сам, так как именно он ставил их в очередь (ему известен порядок). Никаких тегов транзакций (как, например, в системной шине процессоров P6) в интерфейсе AGP нет. Имеется только 4 независимых очереди для каждого типа команд (чтение низкоприоритетное, чтение высокоприоритетное, запись низкоприоритетная, запись высокоприоритетная). Фазы исполнения команд разных очередей могут чередоваться произвольным образом; порт имеет право исполнять их в порядке, оптимальном с точки зрения производительности. Реальный порядок исполнения команд (чтения и записи памяти) тоже может изменяться. Однако для каждой очереди порядок выполнения всегда совпадает с порядком подачи команд (об этом знают и ускоритель, и порт).

Запросы AGP с высоким приоритетом для арбитра системной логики являются более приоритетными, чем запросы от центрального процессора и ведущих устройств шины PCI. Запросы AGP с низким приоритетом для арбитра имеют приоритет ниже, чем от процессора, но выше, чем от остальных ведущих устройств. Хотя принятый протокол никак явно не ограничивает глубину очередей, спецификация AGP формально ее ограничивает до 256 запросов. На этапе конфигурирования устройства система PnP устанавливает реальное ограничение (в конфигурационном регистре ускорителя) в соответствии с его возможностями и возможностями системной платы. Программы, работающие с ускорителем (исполняемые и локальным, и центральным процессорами), не должны допускать превышения числа необслуженных команд в очереди (у них для этого имеется вся необходимая информация).



При передаче данных AGP управляющие сигналы, заимствованные от PCI, имеют почти такое же назначение, что и в PCI. Передача данных AGP в режиме 1x очень похожа на циклы PCI, но немного упрощена процедура квитирования (поскольку это выделенный порт и обмен выполняется только с быстрым контроллером системной памяти). В режимах 2x и 4x имеется специфика стробирования.

- ◆ В режиме 1x данные (4 байта на AD[31:0]) фиксируются получателем по положительному перепаду каждого такта CLK, что обеспечивает пиковую пропускную способность  $66,6 \times 4 = 266$  Мбайт/с.
- ◆ В режиме 2x используются стробы данных AD\_STBO и AD\_STB 1 для линий AD[0:15] и AD [16:31] соответственно. Стробы формируются источником данных, приемник фиксирует данные и по спаду, и по фронту строба. Частота стробов совпадает с частотой CLK, что и обеспечивает пиковую пропускную способность  $66,6 \times 2 \times 4 = 533$  Мбайт/с.
- ◆ В режиме 4x используются еще и дополнительные (инверсные) стробы AD\_STBO# и AD\_STB1#. Данные фиксируются по спадам и прямым, и инверсных стробов. Частота стробов в два раза выше, чем CLK, что и обеспечивает пиковую пропускную способность  $66,6 \times 2 \times 2 \times 4 = 1066$  Мбайт/с.

Порт AGP должен отслеживать состояние готовности буферов ускорителя к посылке или получению данных транзакций, поставленных в очередь. Сигналом RBF# (Read Buffer Full) ускоритель может информировать порт о неготовности к приему данных низкоприоритетных транзакций чтения (к приему высокоприоритетных он должен быть всегда готов). Сигналом WBF# (Write Buffer Full) он информирует о неспособности принять первую порцию данных быстрой записи (Fast Write, FW).

*Конфигурирование устройств* с интерфейсом AGP выполняется так же, как и обычных устройств PCI, — через обращения к регистрам конфигурационного пространства (см. п. 6.2.12). При этом AGP-устройства не требуют внешней линии IDSEL — у них внутренний сигнал разрешения доступа к конфигурационным регистрам соединен с линией AD16, так что обращение к конфигурационным регистрам AGP обеспечивается при AD16=1.

В процессе инициализации процедура POST только распределяет системные ресурсы, но операции AGP оставляет запрещенными. Работу AGP разрешает загруженная ОС, предварительно установив требуемые параметры AGP: режим обмена, поддержку быстрой записи, адресации свыше 4 Гбайт, способ подачи и допустимое число запросов. Для этого параметры устройств считываются из регистра состояния AGP, а согласованные параметры записываются в регистр команд AGP, расположенный в конфигурационном пространстве. Параметры настройки порта задаются через конфигурационные регистры чипсета системной платы (главного моста).

*Регистр состояния AGP* сообщает свойства порта: допустимое число запросов в очередях, поддержку внеполосной адресации, быстрой записи, адресации свыше 4 Гбайт, режимы 1x, 2x, 4x. В конфигурационном пространстве устройства AGP регистр, на который указывает CAP\_PTR, содержит CAP\_ID=02 (биты [7:0]) и номер версии спецификации AGP (биты [23:20] — старшая цифра, биты [19:16] — младшая).

*Регистр состояния AGP* (адрес CAP\_PTR+4) содержит следующие поля:

- ◆ биты [31:24] — RQ, допустимое суммарное число запросов, находящихся в очередях: 0 — 1 команда, 255 — 256 команд;
- ◆ биты [23:10] - резерв (0);
- ◆ бит 9 — SB A, поддержка внеполосной подачи команд;
- ◆ биты [8:6] — резерв (0);
- ◆ бит 5 — 4G, поддержка адресации памяти свыше 4 Гбайт;
- ◆ бит 4 — FW, поддержка быстрой записи;
- ◆ биты 3 — резерв (0);
- ◆ биты [2:0] — RATE, поддерживаемые режимы обмена по AD и SBA: бит 0 — 1x, бит 1 — 2x, бит 2 — 4x.

*Регистр команд AGP* служит для разрешения этих свойств. *Регистр команд AGP* (адрес CAP\_PTR+8) содержит следующие поля:

- ◆ биты [31:24] — RQ\_DEPTH, глубина очереди команд;
- ◆ биты [23:10] - резерв (0);
- ◆ бит 9 — 5BA\_ENABLE, установка внеполосной подачи команд;
- ◆ бит 8 — AGP\_ENABLE, разрешение операций AGP;
- ◆ биты [7:6] — резерв (0);
- ◆ бит 5 — 4G, разрешение адресации памяти свыше 4 Гбайт (двухадресных циклов и посылок 4-го типа по SBA);
- ◆ бит 4 — FW\_Enable, разрешение быстрой записи;
- ◆ биты 3 — резерв (0);
- ◆ биты [2:0] — DATA\_RATE, установка режима обмена: бит 0 — 1x, бит 1 — 2x, бит 2 — 4x (должен быть установлен лишь один бит).

Графический адаптер с интерфейсом AGP может быть встроен в системную плату, а может располагаться и на карте расширения, установленной в *слот AGP*. Внешне карты с портом AGP похожи на PCI (рис. 6.13), но у них используется разъем повышенной плотности с «двухэтажным» (как у EISA) расположением ламелей. Сам разъем находится дальше от задней кромки платы, чем разъем PCI.

Порт AGP может использовать два возможных номинала питания интерфейсных схем: 3,3 В и 1,5 В (уровни сигналов RST# и CLK всегда равны 3,3 В). Снижение напряжения питания буферных схем позволяет повысить достижимую частоту переключений. Для режимов 1x и 2x может использоваться любой из номиналов питания буферов, для режима 4x — только 1,5 В. Для работы в режимах 2x и 4x приемникам требуется опорное напряжение Vref. Его номинал для 3,3 В составляет 0,4xVddq, для 1,5 В — 0,5xVddq. Опорное напряжение для приемников генерируется на стороне передатчиков. На контакт A66 (Vrefgc) графическое устройство подает сигнал для порта, на контакт B66 (Vrefcg) порт (чипсет) подает напряжение для устройства AGP.

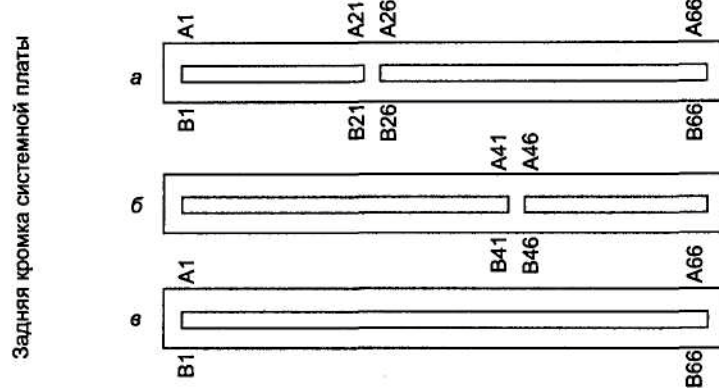


Рис. 6.13. Слоты AGP: а — 3,3 В, б — 1,5 В, в — универсальные

По уровню питания буферов карты и порты AGP могут быть трех типов: 3,3 В, 1,5 В и универсальные, причем имеются механические ключи, предотвращающие ошибочные подключения. Слот и карта 3,3 В имеют ключи на месте контактов 22-25 (перегородка в слоте, см. рис. 6.13, а, вырез на разъеме карты); слот и карта 1,5 В — на месте контактов 42-45. Универсальный слот не имеет перегородок, а универсальная карта имеет оба выреза. Универсальная системная плата узнает о номинале питания буферов установленной карты по сигналу TYPEDET\* — на картах 3,3 В контакт свободен, на картах 1,5 В и универсальных — заземлен. Универсальная карта узнает о номинале питания буферов по уровню напряжения на контактах Vddq (3,3 или 1,5 В). Таким образом и обеспечивается согласование.

Назначение контактов слота AGP приведено в табл. 6.15, в позициях ключей через дробь указано назначение для карт 3,3/1,5 В. На универсальном слоте присутствуют все эти цепи, на универсальных картах все цепи, назначенные ключам, отсутствуют. Из-за двух ключей на универсальной карте теряется пара контактов для подачи питания VCC3.3, и их остается только 4, что ограничивает потребляемый ток (допустимый ток для каждого контакта — 1 А). На универсальной карте также нет дополнительного питания 3,3Vaux, используемого для питания цепей формирования сигнала PME# в режиме «сна».

Таблица 6.15. Сигналы порта AGP

Ряд В	N	РЯДА	Ряд В	N	РЯДА
12V	1	OVRCNTff	Vddq	34	Vddq
TYPEDET»	2	5.0V	AD22	35	AD21
Резерв	3	5.0V	AD20	36	AD19
USB-	4	USB+	GND	37	GND
GND	5	GND	AD18	38	AD17
INTA*	6	INTB#	AD16	39	C/BE2*
RST#	7	CLK	Vddq	40	Vddq
GNT#	8	REQ#	FRAME*	41	IRDY#
VCC3.3	9	VCC3.3	Резерв/Ключ 1,5 В	42	3,3Vaux/IOi04 1,5 В

Ряд В	N	РЯДА	Ряд В	N	РЯДА
ST1	10	STO	GND/fc«041,5В	43	GND/Ключ 1,5 В
Резерв	11	ST2	Резерв/Ключ 1,5 В	44	Резерв/Ключ 1,5 В
PIPE»	12	RBF#	УСС3.3/Ключ1,5В	45	УСС3.3/Ключ1,5В
GND	13	GND	TRDY*	46	DEVSEL#
Резерв	14	Резерв	STOP*	47	Vddq3.3
SBA1	15	SBAO	PME#	48	PERR#
VCC3.3	16	VCC3.3	GND	49	GND
SBA3	17	SBA2	PAR	50	SERR#
SB_STB# <sup>1</sup>	18	SB_STB	AD15	51	C/BE1*
GND	19	GND	Vddq	52	Vddq
SBA5	20	SBA4	AD13	53	AD14
SBA7	21	SBA6	AD11	54	AD12
Ключ 3,3 В/резерв	22	Ключ 3,3 В/резерв	GND	55	GND
Ключ 3,3 В/GND	23	Ключ 3,3 В/GND	AD9	56	AD10
Ключ 3,3 В/резерв	24	Ключ 3,3 В/3,3Vaux	C/BE0#	57	AD8
Ключ 3,3 В/VCC3.3	25	Ключ 3,3 В/3,3Vaux	Vddq	58	Vddq
AD30	26	AD31	AD_STBO# <sup>1</sup>	59	AD_STBO
A028	27	AD29	AD6	60	AD7
VCC3.3	28	VCC3.3	GND	61	GND
AD26	29	AD27	AD4	62	ADS
AD24	30	AD25	AD2	63	AD3
GND	31	GND	Vddq	64	Vddq
AD_STB1# <sup>1</sup>	32	AD_STB1	ADO	65	AD1
C/BE3*	33	AD23	Vrefcg <sup>2</sup>	66	Vrefcg <sup>2</sup>

<sup>1</sup> Инверсные стробы отсутствуют на картах и слотах 3,3 В (там нет режима 4x).

<sup>2</sup> Опорное напряжение не требуется для слотов и карт 1x.

Кроме собственно AGP, в порте AGP заложены *сигналы шины USB*, которую предполагается заводить в монитор (линии USB+, USB- и сигнал OVRcnt\*, которым сообщается о перегрузке по току линии питания +5 В, выводимой в монитор).

Сигнал PME# относится к интерфейсу управления энергопотреблением (Power Management Interface). При наличии дополнительного питания 3,3Vaux этим сигналом карта может инициировать «пробуждение».

*Спецификация A GP Pro* описывает более мощный коннектор, позволяющий в 4 раза повысить мощность, подводимую к графической карте. При этом сохраняется односторонняя совместимость: карты AGP могут устанавливаться в слот AGP Pro, но не наоборот. Коннектор AGP Pro имеет дополнительные контакты с обеих сторон обычного коннектора AGP (рис. 6.14) для линий GND и питания 3,3 и 12 В, назначение этих контактов приведено в табл. 6.16. Для правильной установки обычной карты со стороны задней кромки системной платы дополнительная часть слота AGP Pro закрывается съемной пластмассовой заглушкой. Карта AGP Pro может также использовать 1-2 соседних слота PCI: чисто механически (как точки опо-

ры и место), как дополнительные коннекторы для подачи питания, как функциональные коннекторы PCI.

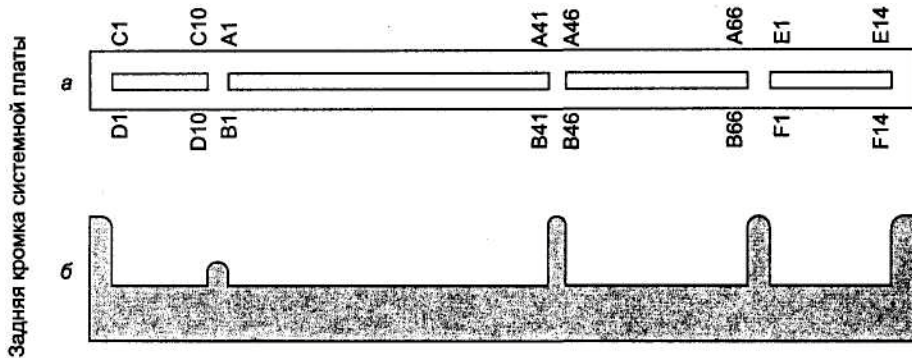


Рис. 6.14. Коннектор карты AGP Pro (показан ключ питания карты 1,5 В): а—вид сверху, б — профиль ключей

Таблица 6.16. Дополнительные контакты коннектора AGP Pro

	Контакт	РядС
VCC3.3	1	VCC3.3
VCC3.3	2	GND
VCC3.3	3	VCC3.3
VCC3.3	4	GND
VCC3.3	5	GND
VCC3.3	6	GND
VCC3.3	7	GND
VCC3.3	8	GND
PRSENT2*	9	Резерв
PRSENT1*	10	Резерв
РядЕ	Контакт	
Резерв	1	Резерв
Резерв	2	Резерв
GND	3	VCC12
GND	4	VCC12
GND	5	VCC12
GND	6	VCC12
GND	7	VCC12
GND	8	VCC12
GND	9	VCC12
GND	10	VCC12
GND	11	VCC12
GND	12	VCC12
GND	13	VCC12
GND	14	VCC12

В совокупности карта AGP Pro может потреблять до 110 Вт мощности, забирая ее по шинам питания 3,3 В (до 7,6 А) и 12 В (до 9,2 А) с основного разъема AGP, дополнительного разъема питания AGP Pro и одного-двух разъемов PCI. Карты AGP Pro большой мощности (High Power, 50-110 Вт) занимают 2 слота PCI, малой (Low Power, 25-50 Вт) — 1 слот. Соответственно скобка крепления к задней панели ПК у них имеет утроенную или удвоенную ширину. Кроме того, карты имеют крепеж к передней стенке ПК. На дополнительном разъеме цепь PRSNT1 # служит признаком наличия карты (контакт заземлен), а PRSNT2\* — признаком потребляемой мощности (до 50 Вт — контакт свободен, до 110 Вт — заземлен).

*В спецификации AGP8X* предполагаются следующие основные отличия:

- ◆ введен новый режим передачи по шинам AD и SBA — 8x, обеспечивающий пиковую производительность 2,132 Гбайт/с;
- ◆ исключены команды длинного чтения и записи;
- ◆ исключены команды высокого приоритета (и упразднены сами понятия низкого и высокого приоритета);
- ◆ исключена возможность подачи команд с помощью сигнала PIPE#;
- ◆ предпринимаются меры по обеспечению когерентности при обращениях к памяти, не лежащей в области GART;
- ◆ несколько изменены протоколы передачи данных, применяется динамическое инвертирование шины данных для минимизации переключений.

Дополнительно предполагается введение поддержки изохронных передач; возможность установки нескольких портов AGP; возможность поддержки разных размеров страниц, описанных в GART; обеспечение когерентности при обращениях к определенным страницам.

## 6.4. Интерфейс LPC

Интерфейс LPC (Low Pin Count — малое число выводов) предназначен для локального подключения устройств, ранее использовавших шину X-Bus или ISA: контроллеров НГМД, последовательных и параллельных портов, клавиатуры, аудиокодека, BIOS и т. п. Введение нового интерфейса обусловлено изживанием шины ISA с ее большим числом сигналов и неудобной асинхронностью. Интерфейс обеспечивает те же циклы обращения, что и ISA: чтение-запись памяти и ввода-вывода, DMA и прямое управление шиной (bus mastering). Устройства могут вырабатывать запросы прерываний. В отличие от ISA/X-Bus с их 24-битной шиной адреса, обеспечивающей адресацию лишь в пределах первых 16 Мбайт памяти, интерфейс LPC имеет 32-битную адресацию памяти, что обеспечивает доступ к 4 Гбайт памяти. 16-битная адресация портов обеспечивает доступ ко всему пространству 64 К портов. Интерфейс синхронизирован с шиной PCI, но устройства могут вводить произвольное число тактов ожидания. Интерфейс программно прозрачен — как и для ISA/X-Bus, не требует каких-либо драйверов. Контроллер интерфейса LPC является устройством-мостом PCI. По пропускной способности интерфейс

практически эквивалентен этим шинам. В спецификации LPC 1.0 приводится расчет пропускной способности интерфейса и устройств, его использующих. При наличии буферов FIFO интерфейс наиболее выгодно использовать в режиме DMA. В этом случае главным потребителем будет LPT-порт — при скорости передачи данных 2 Мбайт/с он займет 47 % полосы интерфейса. Следующим будет инфракрасный порт — 4 Мбит/с (11,4 %). Остальным устройствам (контроллер НГМД, СОМ-порт, аудиокодек) требуются еще меньшие доли, в результате они занимают до 75 % полосы при одновременной работе. Таким образом, перевод этих устройств с ISA/X-Bus на LPC не должен вызывать проблем производительности более острых, чем были на старых шинах. Интерфейс имеет всего 7 обязательных сигналов:

- ◆ LAD[3:0] — двунаправленная мультиплексированная шина данных;
- ◆ LFRAME\* — индикатор начала и конца цикла, управляемый хостом;
- ◆ LRESET# — сигнал сброса, тот же, что и RST# на шине PCI;
- ◆ LCLK — синхронизация (33 МГц), тот же сигнал, что и CLK на шине PCI;

Дополнительные сигналы интерфейса LPC:

- ◆ LDRQ\* — кодированный запрос DMA/Bus Master от периферии;
- ◆ SERIRQ — линия запросов прерывания (в последовательном коде), используется, если нет стандартных линий запросов IRQ в стиле ISA;
- ◆ CLKRUN\* — сигнал, используемый для указания на остановку шины (в мобильных системах), требуется только для устройств, нуждающихся в DMA/BusMaster в системах, способных останавливать шину PCI;
- ◆ PME# — событие системы управления потреблением (Power Management Event), может вводиться периферией, как и в PCI;
- ◆ LPCPD# — Power Down, указание от хоста устройствам на подготовку к выключению питания;
- ◆ L.SMI\* — запрос прерывания SMI# для повтора инструкции ввода-вывода.

Сигналы LFRAME\* и LAD[3:0] синхронизированы (являются действительными) по фронту LCLK. По шине LAD[3:0] в каждом такте цикла передаются поля элементов протокола. Обобщенная временная диаграмма цикла обмена по LPC приведена на рис. 6.15. Начало каждого цикла хост отмечает сигналом LFRAME\*, помещая на шину LAD[3:0] поле START. По сигналу LFRAME\* все ПУ должны прекратить управление шиной LAD[3:0], а по коду поля START они должны декодировать последующие события как цикл шины. В следующем такте хост снимет сигнал LFRAME\* и поместит на шину LAD[3:0] код типа цикла CYCTYPE. Сигнал LFRAME# может длиться и более одного такта, но признаком начала цикла (поля START) является последний такт перед снятием сигнала. С помощью сигнала LFRAME\* хост может принудительно прервать цикл (например, по ошибке тайм-аута), выставив соответствующий код.

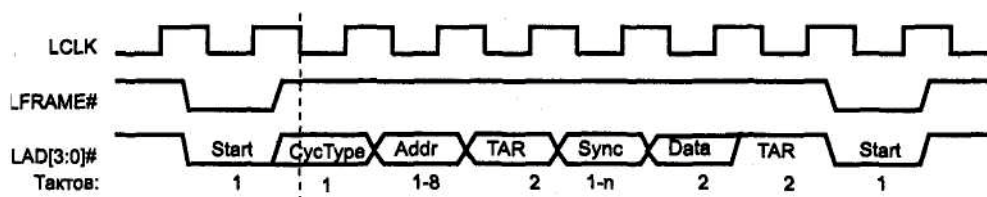


Рис. 6.15. Протокол LPC

В поле START возможны следующие коды:

- ◆ 0000 — начало цикла обращения хоста к устройству;
- ◆ 0010 — предоставление доступа ведущему устройству 0;
- ◆ 0011 — предоставление доступа ведущему устройству 1;
- ◆ 1111 — принудительное завершение цикла (abort).

Остальные коды зарезервированы.

Поле CYSTYPE задает тип и направление передачи. Бит 0 задает направление (0 — чтение, 1 — запись), биты [2:1] — тип обращения (00 — порт, 01 — память, 10 — DMA, 11 — резерв), бит 3 — резерв (0).

Поле TAR (Turn-Around) служит для смены «владельца» шины LAD[3:0], оно занимает 2 такта. В первом такте прежний владелец помещает код 1111, во втором переводит буферы в третье состояние.

Поле ADDR служит для передачи адреса. В цикле памяти оно занимает 8 тактов (32 бита), в цикле ввода-вывода — 4 такта. Передача адреса начинается со старших битов (чтобы раньше сработывал дешифратор адреса).

В поле DATA передаются данные. На каждый байт требуется 2 такта, передача начинается с младшей тетрады. Многобайтные передачи начинаются с младшего байта. Поле SYNC служит для введения тактов ожидания устройством, к которому выполняется обращение. Оно может содержать следующие коды (остальные зарезервированы):

- ◆ 0000 — готовность (без ошибок), для DMA означает снятие запроса для данного канала;
- ◆ 0101 — короткое ожидание (несколько тактов);
- ◆ 0110 — длинное ожидание;
- ◆ 1001 — готовность и присутствие запроса для канала DMA (для других типов обращения недопустимо);
- ◆ 1010 — ошибка: данные переданы, но возникли условия, при которых на шинах PCI или ISA выработался бы сигнал SERR\* или ЮСНК# (для DMA также означает снятие сигнала запроса).



Поле синхронизации обеспечивает контроль передачи, введение тактов ожидания и механизм тайм-аутов. Начав цикл, хост читает поле синхронизации. Если в течение трех тактов адресованное устройство не ответит, хост считает, что его нет на шине, и прекращает транзакцию. Если приходит код короткого ожидания, хост дожидается его смены на готовность или ошибку, но после 8 тактов ожидания он прервет транзакцию по тайм-ауту. Код длинного ожидания может приходиться сколь угодно долго, ответственность за отсутствие зависания ложится на адресованное устройство. При прямом управлении поле SYNC выставляется хостом, и устройство должно терпеливо ожидать готовности, без каких-либо тайм-аутов. В самом быстром варианте (без тактов ожидания) поле SYNC занимает один такт.

На рис. 6.16 приведена последовательность полей при обращениях хоста к памяти или портам (серым цветом помечены поля, вводимые устройством). Во всех этих обращениях передается по одному байту. Для чтения памяти, предполагая 5 тактов поля SYNC (время доступа к EPROM 120 нс), требуется 21 такт (0,63 мкс), что обеспечивает скорость чтения памяти 1,59 Мбайт/с. Если память конвейеризирована, то последующие обращения будут быстрее. Для записи в память поле SYNC займет 1 такт, а весь цикл — 17 тактов (0,51 мкс), что дает скорость записи 1,96 Мбайт/с. Обращения к портам за счет более короткой адресации и без тактов ожидания (1 такт SYNC) занимают по 13 тактов (0,39 мкс), что дает скорость 2,56 Мбайт/с для чтения и записи.

а	START	SYCTYPE	ADDR	TAR	lsyNc	DATA	TAR
б	START	SYCTYPE	ADDR	DATA	TAR	SYNQ	TAR

Рис. 6.16. Обращения к памяти и портам: а — чтение, б — запись

Для организации обмена по DMA и прямого управления хост должен иметь по одной входной линии LDRQ\* для каждого подключенного устройства, использующего эти функции. По этой линии устройство передает состояние запросов каналов DMA в последовательном коде, как показано на рис. 6.17. Посылка начинается со старт-бита, за которой следуют код номера канала и бит запроса АСТ: 1 (высокий уровень) — запрос активен, 0 — пассивен. Номер канала 4 (код 100) зарезервирован для прямого управления (он соответствует традиционно недоступному каналу DMA). Посылка передается при каждой смене состояния запросов. Нормально таким способом передается лишь запрос, а снятие его сигнализируется полем SYNC. Передача данных DMA (рис. 6.18) выполняется под управлением хоста, но несколько отличается от обычных обращений к портам и памяти. Здесь появляются новые поля:

- ◆ поле SIZE определяет размер передачи: 0000 — 1 байт, 0001 — 2 байта, 0011 — 4 байта, остальные значения зарезервированы;
- ◆ поле CHANNEL служит для передачи хостом номера канала DMA (биты [2;0]) и признака конца цикла (ТС, бит 3).

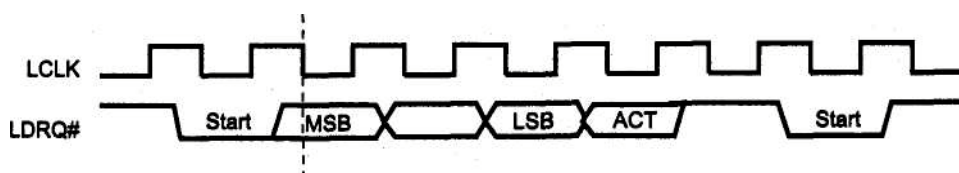


Рис. 6.17. Посылка запроса DMA или прямого управления

A	START	CYCTYPE	CHANNEL	SIZE	DATA	TAR	,JSYNC	TAR	
б	START	CYCTYPE	CHANNEL	SIZE	TAR	SYNC	DATA	TAR	

Рис. 6.18. Обращения DMA: а — чтение памяти, б — запись в память

Циклы чтения памяти, в зависимости от длины обращения (1, 2 или 4 байта), без тактов ожидания (они скрыты контроллером DMA) занимают 11,18 и 32 тактов (0,33, 0,54 или 0,96 икс), что обеспечивает производительность 3,03, 3,70 или 4,17 Мбайт/с соответственно. Циклы записи занимают 11,14 или 20 тактов (0,33, 0,42 или 0,60 мкс), что обеспечивает производительность 3,03, 4,76 или 6,67 Мбайт/с. На рисунке выделены рамкой последовательности полей, повторяющиеся при передаче 2 и 4 байтов.

*Прямое управление шиной* ведущее устройство запрашивает так же, как и прямой доступ, но указывая зарезервированный номер канала 4 (100). Предоставляя управление, хост в поле START задает номер мастера, который позже определит тип цикла (рис. 6.19). Прямое управление подразумевает обращение к ресурсам хоста (системной памяти, устройству PCI). Данные в 2- и 4-байтных передачах следуют друг за другом без зазора, но в циклах чтения памяти и портов всегда будут такты ожидания, поскольку потребуется время на арбитраж шины PCI или доступ к контроллеру памяти. Из расчета 6 тактов (меньше — вряд ли, возможно и больше) поля SYNC, циклы обращения к памяти (и чтения, и записи) потребуют 25,27 или 31 такт (0,75, 0,81 или 0,93 мкс), что обеспечивает производительность 1,33, 2,47 или 4,30 Мбайт/с. За счет более короткой адресации обращение к портам быстрее — 21, 23 или 27 тактов (0,63, 0,69 или 0,81 мкс), что обеспечивает производительность 1,59, 2,90 или 4,94 Мбайт/с.

a	START	TAR	CYCTYPE	ADDR	SIZE	TAR	SYNC	DATA	TAR
---	-------	-----	---------	------	------	-----	------	------	-----

б	START	TAR	CYCTYPE	ADDR	SIZE	DATA	TAR	SYNC	TAR
---	-------	-----	---------	------	------	------	-----	------	-----

Рис. 6.19. Обращения периферийного устройства (Bus Master): а — чтение памяти или порта, б — запись в память (порт)

Электрический интерфейс для сигналов LAD[3:0], LFRAME#, LDRQ# и SERIRQ соответствует спецификации PCI 2.1 для питания 3,3 В. Остальные сигналы в зависимости от системной платы могут быть с уровнями как 5 В, так и 3,3 В.

Конфигурирование устройств LPC не предусматривает использования протоколов PCI или ISA PnP, поскольку все устройства LPC априорно известны системной BIOS. Для обращения к устройствам LPC хост должен декодировать их адреса и направлять обращения по ним на контроллер LPC.

## 6.5. Шины и карты расширения блокнотных ПК

Портативные и блокнотные ПК поначалу строились безо всяких попыток унификации и обеспечения взаимозаменяемости компонентов, но со временем ситуация изменилась. Сейчас существуют несколько основных интерфейсов и конструктивов для устройств расширения, наиболее популярные приведены в табл. 6.17.

Таблица 6.17. Конструктивы и интерфейсы периферии портативных ПК

	PC Card	Small PC Card
Длина	85,6	45,0
Ширина	54,0	42,8
Высота	3,3/5,0/10,5	3,3/5,0/10,5
Коннектор	Штырьковый	Штырьковый
Число контактов	68	68
Интерфейсы	Память, ввод-вывод, CardBus	Память, ввод-вывод

Первый стандарт на карты расширения назывался PCMCIA. Впоследствии он был переименован в PC Card. Кроме слотов шин расширения блокнотные (и карманные) ПК могут иметь и слоты для подключения карт-носителей информации (см. п. 9.3).

Настольный ПК можно снабдить слотами PC Card с помощью специальной карты адаптера-моста, устанавливаемой в слот PCI или ISA. Сами слоты (1-2 штуки) оформляются в корпус 3" и выводятся на лицевую панель ПК; этот корпус соединяется с картой расширения ленточным кабелем-шлейфом.

### 6.5.1. Интерфейсы PCMCIA, PC Card и CardBus

В начале 90-х годов организация PCMCIA (Personal Computer Memory Card International Association — международная ассоциация производителей карт памяти для персональных компьютеров) начала работы по стандартизации шин расширения блокнотных компьютеров, в первую очередь предназначенных для расширения памяти. Первым появился стандарт PCMCIA Standard Release 1.0/JEIDA 4.0 (июнь 1990 г.), в котором был описан 68-контактный интерфейс и два типоразмера карт: Type I и Type II PC Card. Поначалу стандарт касался электрических и физических требований только для карт памяти. Был введен метаформат информационной структуры карты CIS (Card Information Structure), в которой описываются характеристики и возможности карты, — ключевой элемент взаимозаменяемости карт и обеспечения механизма PnP.

Следующая версия *PCMCIA 2.0* (1991 г.) для того же разъема определила интерфейс операций ввода-вывода, двойное питание для карт памяти, а также методики тестирования. В версии 2.01 была добавлена спецификация *PC CardATA*, новый типоразмер *Type III*, спецификация автоиндексируемой массовой памяти *AIMS* (Auto-Indexing Mass Storage) и начальный вариант сервисной спецификации (*Card Services Specification*). В версии 2.1 (1994 г.) расширили спецификации сервисов карт и сокетов (*Card and Socket Services Specification*) и развили структуру *CIS*.

Стандарт *PC Card* (1995 г.) явился продолжением предыдущих; в нем введены дополнительные требования для улучшения совместимости и новые возможности: питание 3,3 В, поддержка *DMA* и 32-битного режима прямого управления шиной *CardBus*. В дальнейшем в стандарт были введены и другие дополнительные возможности.

Все карты *PCMCIA* и *PC Card* имеют 68-контактный разъем, назначение контактов у которого варьируется в зависимости от типа интерфейса карты. Тип интерфейса «заказывается» картой при установке ее в слот, который, естественно, должен поддерживать требуемый интерфейс. *Интерфейс памяти* обеспечивает 8- и 16-битные обращения с минимальным временем цикла 100 нс, что дает максимальную производительность 10 и 20 Мбайт/с соответственно. *Интерфейс ввода-вывода* имеет минимальную длительность цикла 255 нс, что соответствует 3,92/7,84 Мбайт/с для 8-/16-битных обращений. *Интерфейс CardBus* поддерживает практически такой же протокол обмена, что и *PCI*, но с некоторыми упрощениями. Тактовая частота 33 МГц и разрядность 32 бита обеспечивают пиковую пропускную способность в пакетном цикле до 132 Мбайт/с, картам предоставляется возможность прямого управления шиной (*bus mastering*). Здесь используется та же система автоматического конфигурирования, что и в *PCI* (через регистры конфигурационного пространства). В интерфейс заложены дополнительные возможности для цифровой передачи аудиосигнала, причем как в традиционной форме ИКМ, так и в новой (забытой старой) форме ШИМ (*PWM*).

Для дисковых устройств *ATA* в формате *PC Card* имеется специальная спецификация интерфейса (см. п. 9.2.1).

Различают 4 типа *PC Card*: у них у всех размер в плане 54 x 85,5 мм, но разная толщина (меньшие адаптеры встают в большие гнезда):

- ◆ *PC Card Type I* — 3,3 мм — карты памяти;
- ◆ *PC Card Type II* — 5 мм — карты устройств ввода-вывода, модемы, адаптеры локальных сетей;
- ◆ *PC Card Type III* — 10,5 мм — дисковые устройства хранения;
- ◆ *PC Card Type IV* — 16 мм (упоминания об этом типе на сайте [www.pc-card.com](http://www.pc-card.com) найти не удалось).

Есть еще и маленькие карты *Small PC Card* размером 45 x 42,8 мм с тем же коннектором и теми же типами по толщине.

Для карт памяти (динамической, статической, постоянной и флэш-памяти) ассоциация *PCMCIA* поддерживает и стандарт *Miniature Card* (см. п. 9.3.4).

Назначение контактов разъемов для разных типов интерфейса приведено в табл. 6.18, назначение сигналов для интерфейсов карт памяти и ввода-вывода — в табл. 6.19. Для карт CardBus обозначение сигналов начинается с префикса «С», за которым следует имя сигнала, принятое для шины PCI (см. п. 6.2.2).

Таблица 6.18. Разъем PC Card

№	Тип интерфейса			№	Тип интерфейса		
	16 бит Mem	I/O+Mem	32 бита CardBus		16 бит Mem	I/O+Mem	32 бита CardBus
1	GND	GND	GND	35	GND	GND	GND
2	D3	D3	CADO	36	CD1#	CD1#	CCD1#
3	D4	D4	CAD1	37	D11	D11	CAD2
4	D5	D5	CAD3	38	D12	D12	CAD4
5	D6	D6	CAD5	39	D13	D13	CAD6
6	D7	D7	CAD7	40	D14	D14	Резерв
7	CE1#	CE1#	CCBEO*	41	D15	D15	CAD8
8	A10	A10	CAD9	42	CE2#	CE2#	CAD10
9	OE#	OE#	CAD11	43	VS1#	VS1#	CVS1
10	A11	A11	CAD12	44	Резерв	IORD#	CAD13
11	A9	A9	CAD14	45	Резерв	IOWR#	CAD15
12	A8	A8	CCBE1*	46	A17	A17	CAD 16
13	A13	A13	CPAR	47	A18	A18	Резерв
14	A14	A14	CPERR#	48	A19	A19	CBLOCK#
15	WE#	WE#	CGNT#	49	A20	A20	CSTOPff
16	READY	IREQ#	CINT#	50	A21	A21	CDEVSEL*
17	Vcc	Vcc	VCC	51	Vcc	Vcc	Vcc
18	Vpp1	Vpp1	Vpp1	52	VPP2	Vpp2	Vpp2
19	A16	A16	CCLK	53	A22	A22	CTRDY»
20	A15	A15	CIRDY*	54	A23	A23	CFRAME#
21	A12	A12	CCBE2#	55	A24	A24	CAD17
22	A7	A7	CAD 18	56	A25 .	A25	CAD 19
23	A6	A6	CAD20	57	VS2#	VS2#	CVS2
24	A5	A5	CAD21	58	RESET	RESET	CRST#
25	A4	A4	CAD22	59	WAIT#	WAIT#	CSERR#
26	A3	A3	CAD23	60	Резерв	INPACK*	CREQ#
27	A2	A2	CAD24	61	REG#	REG#	CCBE3*
28	A1	A1	CAD25	62	BVD2	SPKR#	CAUDIO
29	AO	AO	CAD26	63	BVD1	STSCHG*	CSTSCHG
30	DO	DO	CAD27	64	D8	D8	CAD28
31	D1	D1	CAD29	65	D9	D9	CAD30
32	D2	D2	Резерв	66	D10	D10	CAD31
33	WP	IOIS16»	CCLKRUN*	67	CD2#	CD2#	CCD2#
34	GND	GND	GND	68	GND	GND	GND

Таблица 6.19. Назначение сигналов карт памяти и ввода-вывода

Сигнал	I/O	Назначение
A[10:0]	I	Линии шины адреса
BVD1,BVD2	I/O	Battery Volt Detection — идентификаторы батарейного питания
STSCHGff	I/O	(Ю) Сигнализация хосту о смене состояния RDY/BSY* и Write Protect. Использование этого сигнала контролируется регистром управления и состояния карты Card Config and Status Register
SPKR#	0	(Ю) Дискретный аудиовыход (на динамик)
CD1#,CD2#	0	Card Detect — сигналы обнаружения (заземлены на карте), по которым хост определяет, что карта полностью вставлена в слот
CE1#,CE2#	I	(Ю, Mem) Card Enable — выбор (разрешение) карты и определение разрядности передачи. Сигнал CE2# всегда относится к нечетному байту, CE1 # — к четному или нечетному, в зависимости от АО и CE2#. С помощью этих сигналов 8-битный хост может обмениваться с 16-битными картами по линиям D[7:0]
D[15:0]	I/O	Шина данных (у 8-битных сигналы D[15:8] отсутствуют)
INPACK	0	(Ю) Input Acknowledge — подтверждение ввода, ответ карты на сигнал IORD* (по этому сигналу хост открывает свои буферы данных)
IORD#	1	Строб команды чтения портов
IOWR	1	Строб команды записи портов (данные должны фиксироваться по положительному перепаду)
#		Строб команды записи портов (данные должны фиксироваться по положительному перепаду)
OE#	1 1	Чтение данных из памяти, конфигурационных регистров и CIS
RDY/BSY	0 0	Готовность карты к обмену данными (при высоком уровне) Запрос прерывания (низким уровнем) Запрос прерывания (высоким уровнем)
# IREQ#	1	Выбор памяти атрибутов (Mem). Для карт Ю сигнал должен быть активен в циклах команд ввода-вывода. В режиме IDE пассивен (соединен с Vcc на стороне хоста)
INTRQ		Сброс (высоким уровнем)
REG#		
RESET	1 0	
VS1#,VS2#		Voltage Sense — сигналы определения номинала питания. Заземленный сигнал VS1 # означает способность чтения карты при питании 3,3 В
WAIT	0	Запрос (низким уровнем) на продление цикла обращения
»	1	Строб записи в память и конфигурационные регистры (в IDE не используется, соединяется хостом с Vcc)
WE#		Write Protect — защита от записи (для карт памяти), запись в память возможна при низком уровне
WP	0	
IOCS16#	0	Разрешение 16-битного обмена

Интерфейс карт памяти и ввода-вывода прост — он практически совпадает с интерфейсом статической асинхронной памяти. Карта выбирается сигналами CE#, действующими одновременно с установленным адресом. Чтение памяти и конфигурационных регистров выполняется по сигналу OE#, запись — по сигналу WE#. Признаком, разделяющим в этих обращениях основную память и конфигурационные регистры, принадлежащие области памяти атрибутов карты, является

сигнал REG#, действующий одновременно с CE# и адресом. Для обращения к портам ввода-вывода служат отдельные сигналы IORD\* и IOWR#; во время их действия должен быть активен и сигнал REG#. В процессе обращения к портам карта может выдать признак возможности 16-битных обращений сигналом IOISC16\* (как на шине ISA). Чтение порта устройство должно подтверждать сигналом INPACK\*, устанавливаемым и снимаемым картой по сигналу CE#. Благодаря этому сигналу хост может убедиться в том, что он читает не пустой слот.

Слоты PC Card могут предоставлять возможность прямого доступа к памяти (DMA). Реализация DMA — самый дешевый способ разгрузки процессора, но такая реализация имеется не на всех хостах, а только на простых, основанных на шине ISA. Для систем с шиной PCI более естественно прямое управление шиной CardBus, правда, для карт реализация прямого управления обходится не дешево.

Для мультимедийных карт имеется возможность переключения интерфейса в специальный режим *ZVPort* (Zoomed Video), в котором организуется отдельный двухточечный интерфейс передачи данных между картой и хост-системой. По смыслу интерфейс напоминает коннектор VFC графических карт — выделенная шина для передачи видеоданных, не связанная с остальными шинами (и не загружающая их), но имеет иной протокол. В режиме ZV Port адресные линии A[25:4], а также линии BVD2/SPKR\*, INPACK\* и IOIS16# получают иное назначение — по ним передаются видеоданные и 4 цифровых аудиоканала. Для обычного интерфейса остаются лишь 4 адресные линии, позволяющие адресоваться к 16 байтам общей памяти и атрибутов карты.

Интерфейс порта ZV соответствует временным диаграммам CCIR601, что позволяет декодеру NTSC в реальном времени доставлять видеоданные с карты в экранный буфер VGA. Видеоданные могут поступать на карту как с внешнего видеовхода, так и с декодера MPEG.

Карты имеют специальное выделенное пространство памяти атрибутов, в котором находятся конфигурационные и управляющие регистры карты, предназначенные для автоконфигурирования. Стандартом описан формат информационной структуры карты (Card Information Structure, CIS). Карты могут быть многофункциональными (например, комбинация модема и сетевого адаптера). В спецификации MFPC (Multiple Function PC Cards) для каждой функции предусматриваются отдельные конфигурационные регистры и определяются правила разделения (совместного использования) линии запроса прерывания.

Для устройств внешней памяти стандарт описывает форматы хранения данных, совместимые с FAT MS-DOS, а также ориентированные на флэш-память как основной носитель информации. Для непосредственного исполнения модулей ПО, хранящихся в ПЗУ карты, имеется спецификация XIP (eXecute In Place), описывающая программный интерфейс вызова этих модулей (вместо загрузки ПО в ОЗУ).

Стандарт описывает программный *интерфейс сервисов карт* (Card Services), обеспечивающий унификацию взаимодействия его *клиентов* (драйверов, прикладно-

го ПО и утилит) с устройствами. Имеется также и *интерфейс сервисов сокета* (Socket Services), с помощью которого выполняются операции, связанные с обнаружением фактов подключения-отключения карт, их идентификации, конфигурирования питания и аппаратного интерфейса.

В стандарте имеются описания специфических особенностей, свойственных двум организациям, ведущим стандарт PC Card.

- ◆ PCMCIA описывает автоиндексируемую массовую память (AIMS) для хранения больших массивов данных (изображений, мультимедийных данных) на блочно-ориентированных устройствах. Имеется также спецификация 15-контактного экранированного разъема для подключения модемов и адаптеров локальной сети (15-pin Shielded Modem I/O connector) и 7-контактного для подключения модемов (7-pin Modem I/O connector).
- ◆ JEDIA для карт памяти предлагает формат файлов Small Block Flash Format, упрощающий файловую систему. Формат SISRIF (Still Image, Sound and Related Information Format) предназначен для записи изображений и звука на карты памяти. Имеется и спецификация для карт динамической памяти.

Большинство адаптеров выпускается с поддержкой технологии PnP и предусматривает «горячее» подключение — интерфейсные карты могут вставляться и выниматься без выключения компьютера. Для этого контакты шин питания имеют большую длину, чем сигнальные, обеспечивая их упреждающее подключение и запаздывающее отключение. Два контакта обнаружения карты CD1 # и CD2# (Card Detect) короче остальных — их замыкание для хоста означает, что карта полностью вставлена в слот. Несмотря на возможность динамического конфигурирования, в некоторых случаях при изменении конфигурации требуется перезагрузка системы.

Первоначально карты и хост-системы использовали напряжение питания логики +5 В. Для перехода на низковольтное питание (3,3 В) был введен механический ключ, не допускающий установки карты на 3,3 В в слот, дающий только 5 В. Кроме того, были определены контакты 43 (VS1#) и 57 (VS2#) для выбора питающего напряжения. На картах с питанием 5 В они оба свободны; на картах 3,3 В контакт VS1# заземлен, а VS2# свободен. По этим линиям хост, допускающий оба варианта напряжения питания, определяет потребности установленной карты и подает соответствующее напряжение. Если хост не способен обеспечить требуемый номинал, он должен не подавать питание, а выдать сообщение об ошибке подключения. Карты обычно поддерживают управление энергопотреблением (APM), что особо актуально при автономном питании компьютера.

В стандарте PC Card выпускают самые разнообразные устройства — память, устройства хранения, коммуникационные средства, интерфейсные порты, игровые адаптеры, мультимедийные устройства и т. п., правда, все они существенно дороже своих крупногабаритных аналогов. Через слот PC Card портативные компьютеры могут подключаться к док-станциям, в которые может быть установлена обычная



периферия. Недостаточно строгое следование производителями стандарту иногда приводит к проблемам совместимости.

Слоты PC Card подключаются к системной шине блокнотного ПК через мост; для компьютеров с внутренней шиной PCI это будет мост PCI-PC Card. В блокнотных ПК могут быть и слоты *Small PCI (SPCI)*, см. п. 6.2.9), но они недоступны без вскрытия корпуса и не допускают «горячей» замены устройств.

## Глава 7

# Интерфейсы электронной памяти

Электронная память применяется практически во всех подсистемах РС, выступая в качестве оперативной памяти, кэш-памяти, постоянной памяти, полупостоянной памяти, буферной памяти, внешней памяти. В этой главе описаны интерфейсы микросхем и модулей динамической, статической и энергонезависимой памяти.

### 7.1. Динамическая память

Динамическая память — D&4M(Dynamic RAM) — получила свое название от принципа действия ее запоминающих ячеек, которые выполнены в виде конденсаторов, образованных элементами полупроводниковых микросхем. При отсутствии обращения к ячейке со временем за счет токов утечки конденсатор разряжается и информация теряется, поэтому такая память требует периодической подзарядки конденсаторов (обращения к каждой ячейке) — память может работать только в динамическом режиме. Этим она принципиально отличается от статической памяти, реализуемой на триггерных ячейках и хранящей информацию без обращений к ней сколь угодно долго (при включенном питании).

Запоминающие ячейки микросхем DRAM организованы в виде двумерной матрицы. Адреса строки и столбца передаются по мультиплексированной шине адреса MA (Multiplexed Address) и стробируются по спаду импульсов RAS# (Row Access Strobe) и CAS# (Column Access Strobe). Состав сигналов микросхем динамической памяти приведен в табл. 7.1.

Таблица 7.1. Сигналы микросхем динамической памяти

Сигнал	Назначение
RAS#	Row Access Strobe — строб выборки адреса строки. По спаду сигнала начинается любой цикл обращения; низкий уровень сохраняется на все время цикла. Перед началом следующего цикла сигнал должен находиться в неактивном состоянии (высокий уровень) не менее, чем время предварительного заряда RAS ( $T_{RP}$ — RAS precharge time)

*продолжение*<sup>^</sup>

Таблица 7.1 (продолжение)

Сигнал	Назначение
CAS#	Column Access Strobe — строб выборки адреса столбца. По спаду сигнала начинается цикл записи или чтения; минимальная длительность ( $T_{cd}$ ) определяется спецификацией быстродействия памяти. Минимальная длительность неактивного состояния между циклами (высокий уровень) должна быть не менее, чем время предварительного заряда CAS ( $T_{cp}$ — CAS precharge time)
MAi	Multiplexed Address — мультиплексированные линии адреса. Во время спада сигнала RAS# на этих линиях присутствует адрес строки, во время спада CAS# — адрес столбца. Адрес должен устанавливаться до спада соответствующего stroba и удерживаться после него еще некоторое время. Микросхемы с объемом 4 М ячеек могут быть с симметричной организацией — 11 бит адреса строк и 11 бит адреса колонок или асимметричными — 12х10 бит соответственно
WE#	Write Enable — разрешение записи. Данные записываются в выбранную ячейку либо по спаду CAS# при низком уровне WE# (Early Write — ранняя запись, обычный вариант), либо по спаду WE# при низком уровне CAS# (Delayed Write — задержанная запись). Переход WE# в низкий уровень и обратно при высоком уровне CAS# записи не вызывает, а только переводит выходной буфер EDO DRAM в высокоимпедансное состояние
OE#	Output Enable — разрешение открытия выходного буфера при операции чтения. Высокий уровень сигнала в любой момент переводит выходной буфер в высокоимпедансное состояние
DB-In	Data Bit Input — входные данные (только для микросхем с однобитной организацией) DB-Out
Data Bit Output	— выходные данные (только для микросхем с однобитной организацией). Выходные буферы стандартных микросхем открыты только при сочетании низкого уровня сигналов RAS#, CAS#, OE# и высокого уровня WE#; при невыполнении любого из этих условий буферы переходят в высокоимпедансное состояние. У микросхем EDO выходные буферы открыты и после подъема CAS#. Логика управления предусматривает возможность непосредственного объединения выходов нескольких микросхем
DQx	Data Bit — объединенные внутри микросхемы входные и выходные сигналы данных (объединение экономит количество выводов для микросхем с многобитной организацией)
N.C.	No Connection — свободный вывод

*Выбранной* микросхемой памяти является та, на которую во время активности (низкого уровня) сигнала RAS# приходит сигнал CAS# (тоже низким уровнем). Тип обращения определяется сигналами WE# и CAS#. Временная диаграмма «классических» циклов записи и чтения приведена на рис. 7.1. Как из нее видно, при чтении данные на выходе относительно начала цикла (сигнала RAS#) появятся не раньше, чем через интервал  $TR_{AC}$ , который и является *временем доступа*. Микросхемы DRAM имеют множество временных параметров, из которых выделим несколько важнейших, с которыми иногда приходится сталкиваться при настройке параметров циклов в CMOS Setup.

- ♦ **Время доступа  $T^A_C$  (RAS Access Time)** — задержка появления действительных данных на выходе относительно спада импульса RAS (см. рисунок). Этот основной параметр спецификации памяти, измеряемый в единицах или десятках наносекунд, обычно является последним элементом обозначения микросхем и модулей (xxx-7 и xxx-70 означают время доступа 70 нс). Для современных микросхем характерно время доступа 40-100 нс.

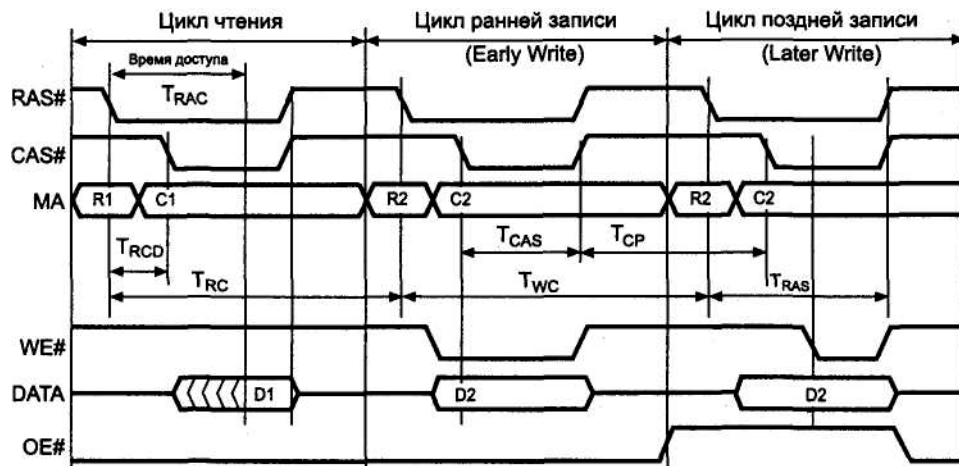


Рис. 7.1. Временные диаграммы чтения и записи динамической памяти

- ◆ Время цикла (cycle time) — минимальный период между началами соседних циклов обращения ( $T_{wc}$  для записи и  $T_{RC}$  для чтения). Для современных микросхем лежит в пределах 75-125 нс.
- ◆ Время цикла (период следования импульсов CAS#) в страничном режиме  $T_{PC}$  (Page CAS Time - см. п. 7.1.1).
- ◆ Длительность сигналов RAS# и CAS# —  $T_{RAS}$  и  $T_{CAS}$  — минимальная длительность активной части (низкого уровня) стробирующих сигналов (см. рисунок).
- ◆ Время предварительного заряда RAS и CAS  $T_{RP}$ , и  $T_{CP}$  (RAS и CAS Precharge Time) — минимальное время нахождения соответствующих сигналов в высоком состоянии.
- ◆ Время задержки между импульсами RAS# и CAS#  $T_{RCD}$  (RAS to CAS Delay).
- ◆ Задержка данных относительно импульса CAS# ( $T_{CAS}$ ).

Все эти параметры и определяют предел производительности памяти. В табл. 7.2 приведены типовые значения временных параметров, отвечающих конкретной спецификации быстродействия. На них можно ориентироваться при задании циклов обращений к памяти в CMOS Setup, но при этом необходимо учитывать, что микросхемы различных производителей могут несколько отличаться друг от друга по отдельным параметрам.

Таблица 7.2. Ключевые параметры временной диаграммы DRAM

Спецификация быстродействия	$T_{RC}$ , нс	$T_{MC}$ , нс	$T_{RC2}$ , нс	нс	$T_{CP}$ , нс
-5	75	40	15	6	6
-6	100	50	20	8	8
-7	104	60	25	10	10
-7	110	70	30	12	12

Отметим, что все, даже самые «модные» типы памяти — SDRAM, DDR SDRAM и Rambus DRAM — имеют запоминающее ядро, которое обслуживается описанным выше способом.

Поскольку обращения (запись или чтение) к различным ячейкам памяти обычно происходят в случайном порядке, то для поддержания сохранности данных применяется *регенерация* (Memory Refresh — обновление памяти) — регулярный циклический перебор ее ячеек (обращение к ним) с холостыми циклами. Циклы регенерации могут организовываться разными способами, классическим является цикл без импульса CAS#, сокращенно именуемый *ROR* (RAS Only Refresh — регенерация только импульсом RAS#). Другой вариант — цикл *CBR* (CAS Before RAS), поддерживаемый практически всеми современными микросхемами памяти. В этом цикле регенерации спад импульса RAS# осуществляется при низком уровне сигнала CAS# (в обычном цикле обращения такой ситуации не возникает). Адрес регенерируемой строки для цикла ROR генерирует контроллер памяти, для CBR этот адрес берется из внутреннего счетчика каждой микросхемы памяти. Цикл *скрытой регенерации* (hidden refresh) является разновидностью цикла CBR. Микросхемы синхронной динамической памяти выполняют циклы CBR по команде *Auto Refresh*. А по команде *Self Refresh* или *Sleep Mode* они выполняют автономную регенерацию в энергосберегающем режиме.

### 7.1.1. Асинхронная память — FPM, EDO и BEDO DRAM

Временная диаграмма, приведенная на рис. 7.1, может быть модифицирована для случая последовательного обращения к ячейкам, принадлежащим к одной строке матрицы. В этом случае адрес строки выставляется на шине только один раз и сигнал RAS# удерживается на низком уровне на время всех последующих циклов обращений, которые могут быть как циклами записи, так и чтения. Такой режим обращения называется режимом быстрого страничного обмена FPM (Fast Page Mode), или просто режимом страничного обмена (Page Mode), его временная диаграмма приведена на рис. 7.2. Понятие «страница» на самом деле относится к строке (row), а состояние с низким уровнем сигнала RAS# называется «открытой страницей». Преимущество данного режима заключается в экономии времени за счет исключения фазы выдачи адреса строки из циклов, следующих за первым, что позволяет повысить производительность памяти. Режим FPM поддерживает и самая обычная асинхронная память, называемая стандартной (Std).

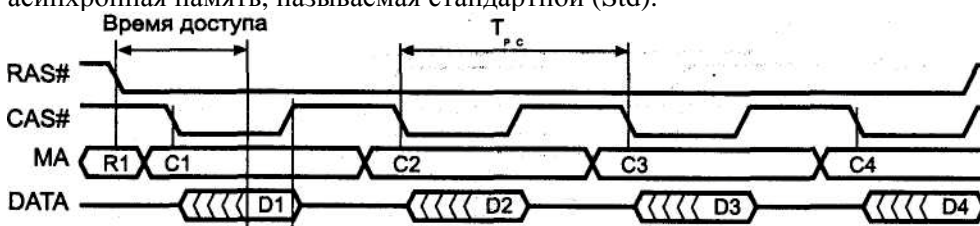


Рис. 7.2. Страничный режим считывания стандартной памяти DRAM (FPM)

Память EDO DRAM (Extended или Enhanced Data Out) содержит регистр-защелку (data latch) выходных данных, что обеспечивает некоторую конвейеризацию работы для повышения производительности при чтении. Регистр «прозрачен» при низком уровне сигнала CAS#, а по его подъему фиксирует текущее значение выходных данных до следующего его спада. Перевести выходные буферы в высоко-импедансное состояние можно либо подъемом сигнала OE# (Output Enable), либо одновременным подъемом сигналов CAS# и RAS#, либо импульсом WE#, который при высоком уровне CAS# не вызывает записи (в PC управление по входу OE# практически не используют).

Временная диаграмма работы с EDO-памятью в режиме страничного обмена приведена на рис. 7.3; этот режим иногда называют гиперстраничным режимом обмена HPM (Hyper Page mode). Его отличие от стандартного заключается в подъеме импульса CAS# до появления действительных данных на выходе микросхемы. Считывание выходных данных может производиться внешними схемами вплоть до спада следующего импульса CAS#, что позволяет экономить время за счет сокращения длительности импульса CAS#. Время цикла внутри страницы уменьшается, повышая производительность в страничном режиме на 40 %.

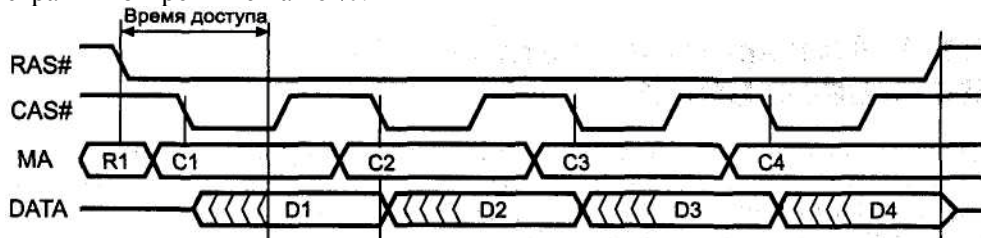


Рис. 7.3. Страничный режим считывания EDO DRAM (HPM)

Установка EDO DRAM вместо стандартной памяти в неприспособленные для этого системы может вызвать конфликты выходных буферов устройств, разделяющих с памятью общую шину данных. Скорее всего, этот конфликт возникнет с соседним банком памяти при чередовании банков. Для отключения выходных буферов EDO-памяти внутри страничного цикла обычно используют сигнал WE#, не вызывающий записи во время неактивной фазы CAS# (рис. 7.4, кривая а). По окончании цикла буферы отключаются лишь по снятию сигнала RAS# (рис. 7.4, кривая б).

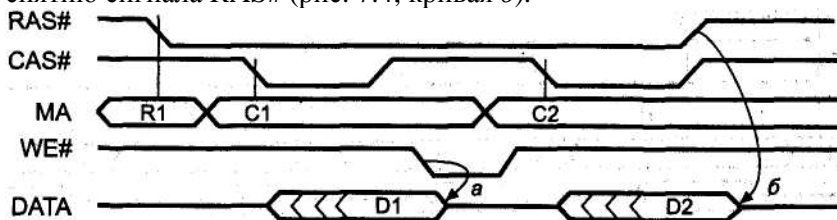


Рис. 7.4. Управление выходным буфером EDO DRAM

Из принципиального различия в работе выходных буферов следует, что в одном банке не стоит смешивать EDO и стандартные модули. EDO-модули поддерживаются не всеми чипсетами и системными платами (в большей мере это относится к системным платам для процессоров 486).

В памяти BEDO DRAM (Burst EDO) кроме регистра-защелки выходных данных, стробируемого теперь по фронту импульса CAS#, содержится еще и внутренний счетчик адреса колонок для пакетного цикла. Это позволяет выставлять адрес колонки только в начале пакетного цикла (рис. 7.5), а во 2-й, 3-й и 4-й передачах импульсы CAS# только запрашивают очередные данные. В результате удлинения конвейера выходные данные как бы отстают на один такт сигнала CAS#, зато следующие данные появляются без тактов ожидания процессора, чем обеспечивается лучший цикл чтения. Задержка появления первых данных пакетного цикла окупается повышенной частотой приема последующих. BEDO-память применяется в модулях SIMM-72 и DIMM, но поддерживается далеко не всеми чипсетами.

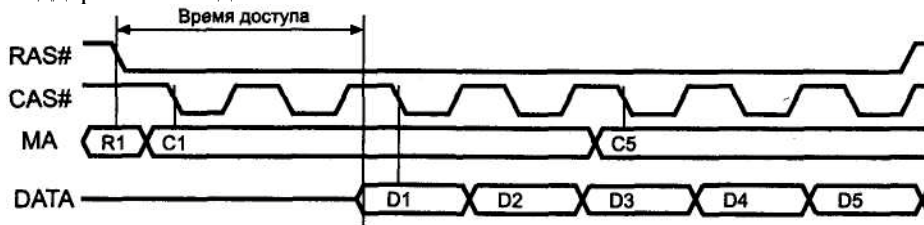


Рис. 7.5. Страничный режим считывания BEDO DRAM

Вышеперечисленные типы памяти являются *асинхронными* по отношению к тактированию системной шины компьютера. Это означает, что все процессы инициируются только импульсами RAS# и CAS#, а завершаются через какой-то определенный (для данных микросхем) интервал. На время этих процессоров шина памяти оказывается занятой, причем в основном ожиданием данных.

## 7.1.2. Синхронная память — SDRAM и DDR SDRAM

Микросхемы синхронной динамической памяти SDRAM (Synchronous DRAM) представляет собой конвейеризированные устройства. По составу сигналов интерфейс SDRAM близок к обычной динамической памяти: кроме входов синхронизации здесь есть мультиплексированная шина адреса, линии RAS#, CAS#, WE# (разрешение записи) и CS# (выбор микросхемы) и линии данных (табл. 7.3). Все сигналы стробируются по положительному перепаду синхроимпульсов, комбинация управляющих сигналов в каждом такте кодирует определенную команду. С помощью этих команд организуется та же последовательность внутренних сигналов RAS и CAS, которая рассматривалась и для памяти FPM.

Таблица 7.3. Назначение сигналов в микросхемах SDRAM

Сигнал	I/O	Назначение
CLK		Clock Input—синхронизация, действует по положительному перепаду
CKE		Clock Enable — разрешение синхронизации (высоким уровнем). Низкий уровень переводит микросхему в режим <i>Power Down</i> , <i>Suspend</i> или <i>Self Refresh</i>
CS#		Chip Select — разрешение декодирования команд (низким уровнем). При высоком уровне новые команды не декодируются, но выполнение начатых продолжается
RAS#, CAS#, WE#		Row Address Strobe, Column Address Strobe, Write Enable — сигналы, определяющие операцию (код команды)
BSD, BS1 или BAD, BA1		Bank Selects или Bank Address — выбор банка, к которому адресуется команда
A[0:12]		Address — мультиплексированная шина адреса. В циклах Bank Activate определяют адрес строки. В циклах Read/Write линии A[0:9] и A11 задают адрес столбца. Линия A10 в циклах Read/Write включает режим автопредзаряда (при A10=1), в цикле Precharge A10=1 задает предзаряд всех банков (независимо от BSO, BS1)
DQx	I/O	Data Input/Output — двунаправленные линии данных
DQM		Data Mask — маскирование данных. В цикле чтения высокий уровень переводит шину данных в высокоимпедансное состояние (действует через 2 такта). В цикле записи высокий уровень запрещает запись текущих данных, низкий — разрешает (действует без задержки)
V <sub>ss</sub> , V <sub>OD</sub>		Общий провод и питание ядра
VSSQ <sub>1</sub> , V <sub>DDQ</sub>		Общий провод и питание выходных буферов. Изолированы от питания ядра для снижения помех

Данные для первой передачи *пакета записи* устанавливаются вместе с командой WR. Данные для остальных передач пакета передаются в следующих тактах. Первые данные *пакета чтения* появляются на шине через определенное количество тактов после команды. Это число, называемое *CAS Latency (CL)*, определяется временем доступа T<sub>CAC</sub> <sup>n</sup> тактовой частотой. Остальные данные пакета выдаются в последующих тактах. Временные диаграммы работы SDRAM приведены на рис. 7.6. Здесь показана команда записи WR, за которой следует команда чтения RD из той же страницы, предварительно открытой командой ACT. Далее страница закрывается командой PRE. Длина пакета 2, CL - 3.

Регенерация (цикл CBR с внутренним счетчиком адреса регенерируемой строки) выполняется по команде REF, которую можно вводить только при состоянии покоя (*idle*) всех банков.

Микросхемы SDRAM оптимизированы для пакетной передачи. У них при инициализации программируется длина пакета (*burst length=1, 2, 4, 8* элементов), порядок адресов в пакете (*wrap mode: interleave/linear* — чередующийся/линейный) и операционный режим. Пакетный режим может включаться как для всех операций (*normal*), так и только для чтения (*Multiple Burst with Single Write*). Этот выбор позволяет оптимизировать память для работы либо с WB, либо с WT-кэшем.



Обратим внимание, что внутренний счетчик адреса работает по модулю, равному запрограммированной длине пакетного цикла (например, при burst length=4 он не позволяет перейти границу обычного четырехэлементного пакетного цикла).

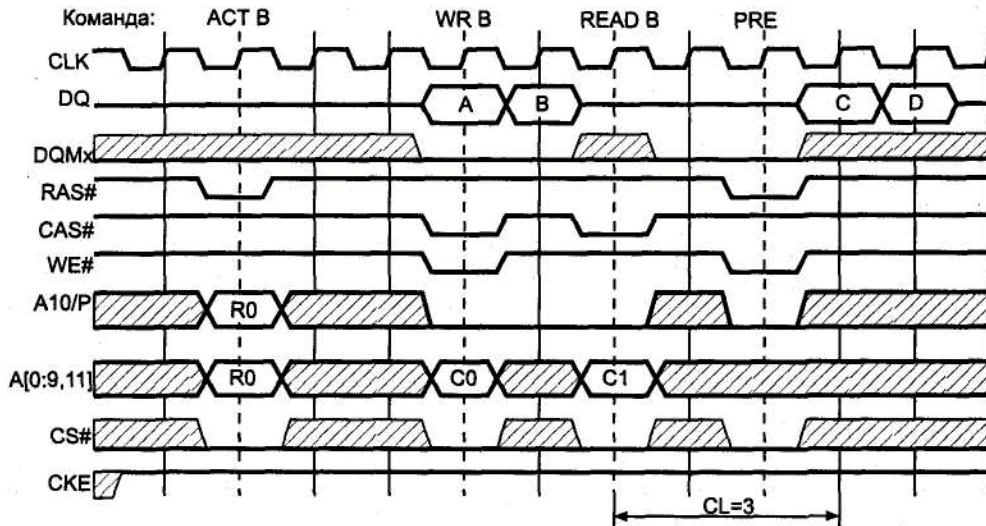


Рис. 7.6. Временные диаграммы пакетных циклов SDRAM: А и В — данные для записи по адресу RO/CO и RO/CO+1, С и D — данные, считанные по адресу RO/C1 и RO/C1 +1

Пакетные циклы могут прерываться (принудительно завершаться) последующими командами. При этом оставшиеся адреса отбрасываются, и прерывающий пакет будет иметь полную длину (если его самого не прервут).

В команде Write имеется возможность блокирования записи данных любого элемента пакета — для этого достаточно в его такте установить высокий уровень сигнала DQM. Этот же сигнал используется и для перевода в высокоимпедансное состояние буферов данных при операции чтения.

Микросхемы SDRAM имеют средства энергосбережения, для управления ими используется вход разрешения синхронизации CKE.

В режиме саморегенерации (Self Refresh) микросхемы периодически выполняют циклы регенерации по внутреннему таймеру и не реагируют на внешние сигналы, поэтому внешняя синхронизация может быть остановлена.

Режимы пониженного потребления (Power Down Mode) устанавливаются при переводе CKE в низкий уровень командой NOP или INHBT. В этих режимах микросхема не воспринимает команд. Поскольку в данных режимах регенерация не выполняется, длительность пребывания в них ограничена периодом регенерации.

Если во время выполнения команды чтения или записи установить CKE=L, то микросхема перейдет в режим Clock Suspend Mode, в котором «замораживается» внутренняя синхронизация (нет продвижения данных) и не воспринимаются новые команды.

Для памяти SDRAM ключевыми параметрами являются:

- ◆ допустимая тактовая частота;
- ◆ CL (Cas Latency) — число скрытых тактов (2 или 3);
- ◆  $T_{RCD}$  — задержка RAS-CAS, выраженная в тактах (2 или 3);
- ◆  $T_{RP}$  — время предварительного заряда RAS;
- ◆  $T_{RC}$  — минимальное время цикла обращений к строкам одного банка;
- ◆  $T_{AC}$  — время задержки появления данных на выходе относительно фронта синхросигнала.

По тактовой частоте для SDRAM, применяемой в качестве ОЗУ PC-совместимых компьютеров, имеется три градации: PC66 (поначалу ее так не называли, поскольку другой и не было), PC100 и PC133 для максимальных частот 66,6, 100 и 133 МГц соответственно. Их ключевые параметры приведены в табл. 7.4. В обозначении быстродействия микросхем SDRAM обычно фигурирует  $T_{AC}$ ; период частоты синхронизации, естественно, не может быть меньше этой задержки. Микросхемы со спецификацией -10 могут устойчиво работать в модулях лишь на частоте 66 МГц. Микросхемы -8 могут работать на частоте 100 МГц, но, в зависимости от модификации, с разной латентностью. Так, например, для памяти Micron микросхемы с маркировкой -8A...-8C могут работать на частоте 100 МГц с CL = 3, а -8D или -8E — с CL = 2. Естественно, память может работать и на частотах ниже максимальной. Для микросхем SDRAM, применяемых, например, в графических адаптерах, существуют и иные спецификации быстродействия.

Таблица 7.4. Ключевые параметры временной диаграммы SDRAM

Спецификация	CL	$T_{RCD}$	$T_{RP}$	$T_{RC}$	Примечание
PC66	3	2	3	8	Медленный вариант
	2	2	2	7	Самый быстрый вариант
PC100	3	3	3	8	Медленный вариант
	3	2	2	7	Средний вариант
	2	2	2	7	Самый быстрый вариант
PC 133	3	3	3	9	Медленный вариант
	3	2	2	8	Средний вариант
	2	3	2	8	Средний вариант
	2	2	2	8	Самый быстрый вариант

Синхронный интерфейс позволяет довольно эффективно использовать шину и обеспечить на частоте 100 МГц пиковую производительность 100 Мбит/с на 1 вывод шины данных. SDRAM используют в составе модулей DIMM с 8-байт-ной разрядностью, что дает производительность 800 Мбайт/с. При частоте шины 133 МГц пиковая производительность уже достигла 1064 Мбайт/с. Однако эта теоретическая производительность не учитывает накладные расходы на регенерацию и подразумевает, что требуемые страницы уже открыты. Из-за указанных выше ограничений на реальном произвольном потоке запросов производительность, конечно же, будет ниже. Потенциальные возможности почти одновременно-

го обслуживания множества запросов, предоставляемые микросхемами SDRAM, будут реализованы лишь при достаточно «умном» контроллере памяти. От его предусмотрительности эффективность памяти зависит, пожалуй, больше, чем у простых модулей FPM и EDO DRAM.

*Память DDR SDRAM* представляет собой дальнейшее развитие SDRAM. Как и следует из названия (Dual Data Rate — удвоенная скорость данных), у микросхем DDR SDRAM данные внутри пакета передаются с удвоенной скоростью — они переключаются по обоим фронтам синхроимпульсов (рис. 7.7). На частоте 100 МГц DDR SDRAM имеет пиковую производительность 200 Мбит/с на вывод, что в составе 8-байтных модулей DIMM дает производительность 1600 Мбайт/с. На высоких тактовых частотах (100 МГц) двойная синхронизация предъявляет очень высокие требования к точности временных диаграмм. Для повышения точности синхронизации предпринят ряд мер.

- ◆ Сигнал синхронизации микросхемы подается в дифференциальной форме по двум линиям CLK и CLK# (Differential clock inputs). Это позволяет снизить влияние смещения уровней на точность определения момента синхронизации — дифференциальный приемник срабатывает в момент равенства уровней напряжения.
- ◆ Для синхронизации данных в интерфейс введен новый двунаправленный стробирующий сигнал DQS. Стробы генерируются источником данных: при операциях чтения DQS генерируется микросхемой памяти, при записи — контроллером памяти (чипсетом). При чтении фронты и спады этого сигнала точно центруются в моменты смены данных, приемник должен стробировать данные с небольшой задержкой относительно переключений DQS. При записи фронты и спады центруются точно посередине окна действительности данных и масок DQM.
- ◆ Для синхронизации DQS с системной тактовой частотой (CLK) микросхемы имеют встроенные схемы DLL (Delay Locked Loop) для автоподстройки задержки сигнала DQS относительно CLK. Эта схема работает наподобие фазовой автоподстройки и способна выполнять синхронизацию (обеспечивать совпадение фронтов DQS и CLK) лишь в некотором ограниченном диапазоне частот синхронизации.

Есть микросхемы DDR SDRAM с возможностью отключения схем DLL; для этого они имеют дополнительный расширенный регистр режима. Отключение DLL необходимо при снижении тактовой частоты (в целях энергосбережения). При отключенной схеме DLL стробы DQS не привязаны к синхросигналу CLK, и у разных микросхем, работающих в одной системе, они будут иметь разные частоты.

В отличие от обычных микросхем SDRAM, у которых данные для записи передаются одновременно с командой, в DDR SDRAM данные для записи (и маски DQM) подаются с задержкой на один такт (write latency). Значение CAS Latency может быть и дробным (CL = 2, 2,5, 3).

В перспективе ожидается появление микросхемы DDR-II SDRAM, в которой обмен будет на четырехкратной частоте синхронизации.

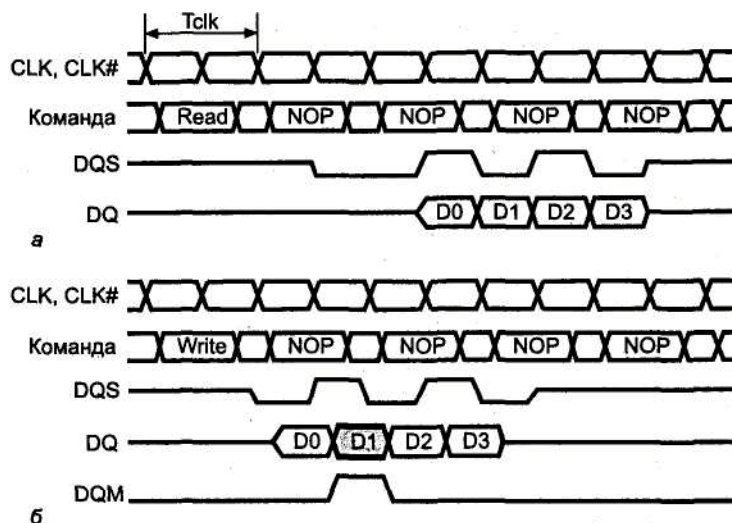


Рис. 7.7. Временные диаграммы пакетных циклов DDR SDRAM: а — чтение,  $CL = 2$ , длина пакета 4; б — запись, длина пакета 4, данные D1 не записываются

Перед «штатным» использованием микросхем SDRAM их требуется инициализировать. После подачи питания и установления синхросигнала должен быть выполнен предварительный заряд всех банков, после чего запрограммирован регистр режима. Параметр CL (CAS Latency) выбирают, исходя из спецификации микросхем и тактовой частоты так, чтобы задержка, обусловленная CL, была бы минимальной, но не меньше  $T_{CAS}$ . В DDR SDRAM возможны и дробные значения CL, так что настройка может быть более тонкой. В DDR SDRAM из-за необходимости настройки DLL программирование сложнее.

По причине существенного отличия интерфейса от традиционной асинхронной памяти микросхемы SDRAM не могут быть установлены в модули SIMM; они применяются в DIMM или устанавливаются прямо на системную (или графическую) плату. Интерфейс DDR SDRAM сильно отличается и от обычных микросхем SDRAM. Возможность использования этих типов памяти определяется чипсетом системной платы. Память SDRAM в конце 90-х годов стала самой распространенной, поддержка DDR SDRAM появилась лишь сравнительно недавно.

### 7.1.3. Память Rambus DRAM

Память RDRAM (Rambus DRAM) имеет синхронный интерфейс, существенным образом отличающийся от вышеописанного. Запоминающее ядро этой памяти построено на все тех же КМОП-ячейках динамической памяти, но пути повышения производительности интерфейса совершенно иные. Подсистема памяти (ОЗУ) RDRAM состоит из контроллера памяти, канала и собственно микросхем памяти. По сравнению с DDR SDRAM, при той же производительности RDRAM имеет более компактный интерфейс и большую масштабируемость. Разрядность ОЗУ RDRAM (16 байт) не зависит от числа установленных микросхем, а число банков,

доступных контроллеру, и объем памяти суммируются по всем микросхемам канала. При этом в канале могут присутствовать микросхемы разной емкости в любых сочетаниях.

*Запоминающее ядро* микросхем имеет многобанковую организацию — 64-мегабит-ные микросхемы имеют 8 банков, 256-мегабитные — 32 банка. Каждый банк имеет собственные усилители считывания, благодаря чему в микросхеме может быть активировано несколько банков. Для сокращения числа усилителей применяют и их разделение (совместное использование) парой смежных байт, что накладывает ограничения на их совместную активацию (до активации банка его смежник должен быть заряжен). Разрядность ядра 16 байт — 128 или 144 (с контрольными разрядами) бит. Ядро работает на 1/8 частоты канала, взаимодействие с ядром осуществляется по внутренним сигналам RAS и CAS. В современных RDRAM применяются ячейки памяти с временем доступа 40-53 нс.

*Канал RDRAM (Rambus Channel)* представляет собой последовательно-параллельную шину. Такая организация позволяет ограничить количество линий интерфейса, что, в свою очередь, позволяет упорядочить разводку проводников ради повышения частоты передачи сигналов. Небольшое количество сигналов дает возможность выровнять задержки распространения сигналов по разным линиям и применить сверхбыстродействующие интерфейсные схемы. Тактовая частота канала — до 400 МГц, стробирование информации осуществляется по обоим фронтам синхросигнала. Таким образом, пропускная способность одной линии составляет 800 Мбит/с. Канал состоит из 30 основных линий с интерфейсом *RSL* (Rambus System Logic) и 4 вспомогательных линий КМОП, используемых для инициализации микросхем. Структура канала изображена на рис. 7.8. На канале может быть установлено до 32 микросхем, все микросхемы соединяются параллельно. Для того чтобы контроллер мог адресоваться к определенной микросхеме, каждой из них назначается уникальный адрес DEVID. Нумерация микросхем (Device Enumeration) осуществляется в процессе инициализации, который выполняется с использованием вспомогательного последовательного КМОП-интерфейса. Этот интерфейс имеет линии синхронизации SCK, команд CMD данных SIO.

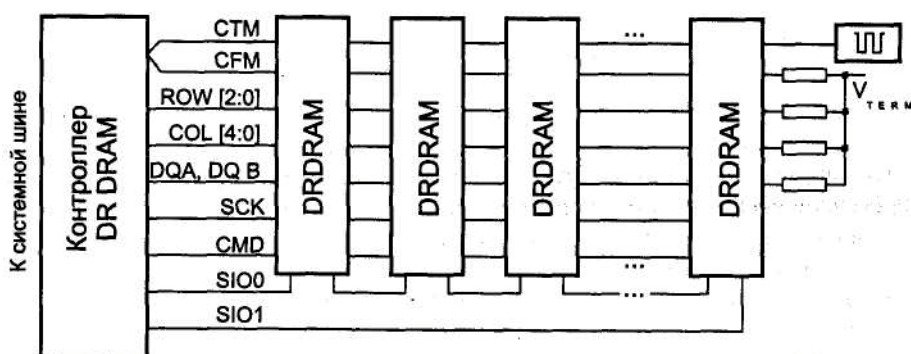


Рис. 7.8. Память Direct RDRAM

Синхросигнал вводится в канал с «дальнего конца» и распространяется в сторону контроллера по линии СТМ (Clock To Master). По этому сигналу микросхемы памяти стробируют данные, посылаемые к контроллеру (при чтении). Дойдя до контроллера, синхросигнал выходит на линию CFM (Clock From Master) и идет по каналу до терминатора, установленного на конце. По этой линии синхронизируется информация, посылаемая от контроллера к микросхемам памяти. Микросхемы привязывают данные чтения к синхросигналу с помощью встроенных схем DLL (Delay Locked Loop) для автоподстройки задержки сигнала DQS относительно CLK. *Физический уровень интерфейса* учитывает волновой характер процессов распространения сигналов в канале. Интерфейс RSL имеет малый размах сигнала: уровню логический 0 соответствует потенциал терминаторов  $V_{\text{TERM}} = 1,8 \text{ В}$ , лог. 1 — потенциал 1,0 В. Передатчики имеют выход типа «открытый сток N-МОП» и активны при передаче логической единицы. Передатчики управляют значением генерируемого тока с тем, чтобы обеспечить требуемый уровень сигнала (падение напряжения на сопротивлении терминатора). Уровень переключения приемника  $V_{\text{REF}} = 1,4 \text{ В}$  задается делителем напряжения  $V_{\text{TERM}}$ . Сигнал синхронизации передается в дифференциальной форме по линиям СТМ, СТМN к контроллеру и по линиям CFM, CFMN от него. Дифференциальная форма снижает погрешность стробирования, вызванную смещением уровней сигналов.

Канал разделен на *три независимые шины*: 3-битная шина строк ROW[2:0], 5-битная шина колонок СОЦ[4:0] и двухбайтная (2x9 бит) шина данных DQA[8:0] и DQB[8:0]. Дополнительный бит байта данных (имеется не у всех микросхем RDRAM) может использоваться для контроля достоверности. По каждой шине информация передается *пакетами*, занимающими 4 такта (8 интервалов) синхронизации (10 нс). Пакет содержит 8 элементов; пакет строк имеет емкость 24 бит, колонок — 40 бит и данных — 16 байт по 8 или 9 бит.

Высокая производительность шины управления (строк и колонок) позволяет отказаться от пакетных (в терминологии BEDO и SDRAM) передач и упростить протокол шины. Память может одновременно обслуживать до четырех транзакций на полной скорости передачи данных.

*Транзакции чтения* приведены на рис. 7.9, по виду они аналогичны транзакциям SDRAM с тем лишь отличием, что за время одного такта (SDRAM) передается *пакет*. Пакет ROW для второй транзакции пропущен, поскольку страницу оставили открытой. Транзакция чтения со стороны контроллера представляет собой петлю: он посылает пакеты ROWA и COLC, которые за некоторое время достигают целевой микросхемы и ею обрабатываются за время  $T_{\text{CAS}}$ . Далее микросхема отвечает пакетом данных, которому для достижения контроллера также требуется некоторое время. Пакетам для путешествий к дальним микросхемам и от них требуется больше времени, чем для путешествий к ближним, и эта разница оказывается большей, чем длительность периода синхронизации. Для того чтобы контроллер получал ответ на транзакцию чтения от любой микросхемы через одно и то же число тактов, у микросхем памяти устанавливаются разную задержку данных относительно пакетов COLC. Группы соседних микросхем, у которых программируется одинаковая задержка, называют *доменами синхронизации*. В канале может быть несколько доменов синхронизации.

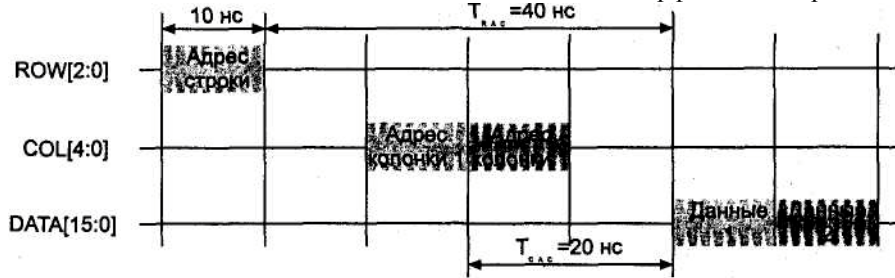


Рис. 7.9. Транзакции чтения RDRAM

Транзакции записи (рис. 7.10) являются однонаправленными и для них проблем синхронизации не возникает. В отличие от стандартных микросхем DRAM и SDRAM, где данные для записи передаются одновременно с адресом колонки, в RDRAM данные задерживают относительно пакета COLC на  $T_{CWD}$  (несколько тактов). Эта задержка соответствует задержке между пакетами COLC и данными при чтении (на стороне контроллера). Задержка записи позволяет сократить вынужденные простои шины данных при переключении с записи на чтение (в SDRAM они равны CAS Latency и длятся 2-3 такта по 10 нс). Контроллер может посылать данные для записи уже в такте, следующем за последними данными предыдущей транзакции чтения. Однако если за записью следует чтение, то на шине данных будет вынужденная пауза в 1-5 тактов в зависимости от длины канала. За это время последние данные записи дойдут от контроллера до самой дальней микросхемы памяти.

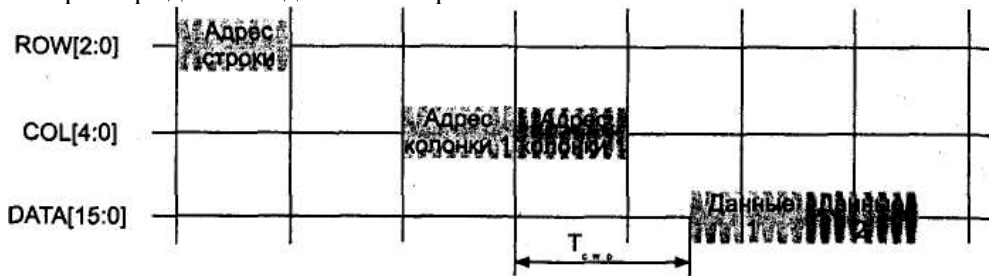


Рис. 7.10. Транзакции записи RDRAM

В микросхемах RDRAM применяется механизм *отложенной*, или *буферизированной*, записи. Данные для записи (принятый пакет D) сначала помещаются в буфер, из которого несколько позже они выгружаются в усилители считывания-записи (sens amp) по явной команде выгрузки (retire) или автоматически. Буфер записи хранит сами данные, а также номер банка и адрес столбца (но не строки). Буферизация записи позволяет контроллеру посылать команду записи на  $T_{RTR}$  раньше, чем этого требует параметр  $T_{RCD}$ , что повышает коэффициент использования шины.

Конвейерное выполнение операций RDRAM обеспечивается многобанковой организацией с отдельными усилителями считывания. Пакеты команд по линиям ROW и COL могут идти сплошным потоком, при этом на шине может присутство-

вать до четырех транзакций. При произвольных обращениях увеличению производительности способствует большое количество банков, практически недостижимое в памяти SDRAM.

*Регенерация* осуществляется по команде, адресуемой к определенному банку одной или всех микросхем; за период регенерации эта команда должна выдаваться для всех банков.

Память RDRAM отличается высоким энергопотреблением. *Средства управления энергопотреблением* отключают питание неиспользуемых узлов, однако за 100-кратное снижение мощности в самом экономичном режиме приходится расплачиваться 250-кратным увеличением времени доступа. Микросхемы RDRAM требуют периодической (раз в 100 мс) подстройки выходного тока и термокалибровки; для этих целей имеются специальные команды. Во время подстройки тока микросхемы способны сообщать о своем перегреве.

*Вспомогательная шина* с сигналами SCK, CMD и SIO служит для обмена данными с управляющими регистрами и вывода микросхем из состояний пониженного потребления (*PDN* и *NAP*). Информация по этой шине тоже передается пакетами.

*Управляющие регистры* хранят информацию об адресе микросхемы, управляют работой микросхемы в различных режимах, содержат счетчики регенерации для банков и строк, параметры настройки временных циклов. В них же можно прочитать информацию о конкретной микросхеме — организация, версия протокола и т. п. В составе управляющих есть и тестовые регистры.

*Инициализация* памяти включает определение наличия микросхем на шине, назначение им идентификаторов и программирование их параметров. После сброса микросхемы не имеют собственных адресов, а линии SIO0 и SIO1 у них соединены. В таком состоянии контроллер по шине CMD посылает широковещательную команду на разъединение линий, после чего для него по линии SIO оказывается доступной только ближайшая микросхема канала. Ей назначается адрес (SDEVID) и дается команда на соединение линий, в результате к контроллеру подключается вторая микросхема. Она будет пронумерована очередной командой, заставляющей все доступные нenumерованные микросхемы (то есть именно ее) принять указанный номер. Далее замыкаются ее линии SIO, и этот процесс продолжается до самой дальней микросхемы.

После завершения этого «переучета» включается нормальная синхронизация, и дается время для установления режима схем DLL. После двукратной активации и предварительного заряда каждого банка каждой микросхемы память готова к определению доменов синхронизации и назначению каждой микросхеме соответствующих параметров задержек. Также им должны быть присвоены идентификаторы в канале (DEVID), которые могут и не совпадать со значениями SDEVID (идентификатором на последовательной шине).

Обязательным «фирменным» компонентом ОЗУ на RDRAM является *контроллер памяти*. В его задачу входит обслуживание микросхем памяти, установленных в канале, по запросам, поступающим со стороны интерфейса системной шины компьютера. Часть контроллера, обращенная к каналу, инвариантна к архитектуре-



ре компьютера. Именно она «знает» протокол RDRAM и является продуктом фирмы Rambus. Контроллер RDRAM встраивается в чипсеты для процессоров P6 (например, 1820, 1840), Pentium 4 (1850 с 32-разрядным каналом, то есть уже под пары модулей RIMM) и других архитектурных линий.

В соответствии со спецификацией RDRAM в одном канале может быть до трех слотов под RIMM, и их интерфейсные линии соединяются змейкой. В слоты могут устанавливаться RIMM различной емкости (сейчас они выпускаются на 64, 96, 128 и 256 Мбайт). Однако пока что фирме Intel не удалось достичь устойчивой работы канала с тремя модулями и пришлось ограничиться двумя. Теперь в памяти появился новый элемент-пустышка *Continuity module*. Это как бы модуль RIMM, но без микросхем памяти, и нужен он для того, чтобы замыкать цепь канала Rambus. Такая «заглушка» должна устанавливаться во все слоты канала, не занятые под модули RIMM. Если используются не все слоты, то память выгоднее ставить ближе к контроллеру — она будет работать быстрее (см. выше).

### 7.1.4. Память с виртуальными каналами — VC DRAM

Идея архитектуры памяти с виртуальными каналами (VirtualChannel Memory Architecture, не путать с виртуальной памятью!) заключается в помещении между массивом запоминающих ячеек и внешним интерфейсом микросхемы памяти набора *канальных буферов*. При этом операции обмена данными разделяются на два процесса: «фасадный» обмен данными с каналами и «тыловой» обмен между каналами и массивом запоминающих ячеек. Оба процесса выполняются по командам со стороны внешнего интерфейса почти независимо друг от друга. Архитектура виртуальных каналов приложима к памяти любого типа, включая ПЗУ и флэш-память, но наиболее интересна она в приложении к динамической памяти — VC DRAM.

Устройство VC DRAM рассмотрим на примере микросхем емкостью 128 Мбит, на которых строятся выпускаемые модули DIMM VC DRAM. По интерфейсу (составу и уровням сигналов) микросхемы и модули VC DRAM аналогичны обычным микросхемам SDRAM, но отличаются системой команд. Микросхемы имеют такую же внешнюю организацию по 4,8 или 16 бит данных, но совершенно иную внутреннюю архитектуру. Они имеют две матрицы (два банка) запоминающих ячеек размером 8 Кх8 К, то есть каждая строка имеет объем 8 Кбит и состоит из четырех *сегментов* размером по 2 Кбит. Между матрицами и внешним интерфейсом имеется 16 *канальных буферов*, каждый объемом 2 Кбит. За одно обращение к матрице выполняется параллельная передача 2 Кбит данных между одним из буферов и сегментом выбранной строки. Это «тыловой» обмен реализуют команды PRF (Prefetch — чтение массива в буфер) и RST (Restore — сохранение буфера в массиве), в которых микросхеме указывается номер банка, номер сегмента и номер канала. Предварительно командой ACT должна быть активирована требуемая строка матрицы (при подаче этой команды задается банк и адрес строки). Деактивация строк (предварительный заряд) может быть автоматической, сразу после выполнения обращений к массиву (для этого имеются специальные команды предвыборки и сохранения — PRFA и RSTA) или же по специальным командам, деактивирующим выбранный банк или оба банка сразу.

«Фасадный» обмен с канальными буферами выполняется по командам чтения и записи (READ и WRITE), в которых указывается номер канала и часть адреса, соответствующая адресу колонки в обычной микросхеме DRAM или SDRAM. Этот обмен выполняется в пакетном режиме, длина пакета программируется (1,2,4,8 или 16 передач), но пакет может быть укорочен подачей следующей команды обращения к каналу. Первые данные при чтении канала появляются с задержкой (Read Latency) в 2 такта относительно команды чтения, следующие идут в каждом такте. В некоторых моделях микросхем имеется поддержка комбинированной команды PFR (перед которой тоже должна быть команда ACT) — предвыборка с автопредзарядом и чтение буфера. После подачи этой команды первые данные появляются на 4-м такте — не раньше и не позже, чем при последовательной подаче команд PRF (A) и READ.

Регенерация VC DRAM выполняется так же, как и в SDRAM, — либо периодической подачей команд REF (авторегенерация по внутреннему счетчику адреса регенерируемых строк), либо в энергосберегающем режиме саморегенерации, в который микросхемы переходят по команде SELF.

Как видно из этого описания, работа VC DRAM очень похожа на работу SDRAM, но операции обмена данными разделены на две сравнительно независимые фазы. Активация-деактивация банков выглядит так же, но при чтении VC DRAM данные появляются даже позже, чем в SDRAM: у SDRAM эта задержка, CL (CAS Latency), составляет 2-3 такта, а у VC DRAM — 4 такта. Тем не менее применение VC DRAM дает прирост производительности памяти почти по всем тестам. Этот выигрыш получается за счет поддержки многозадачности в самих микросхемах и в контроллере памяти. Для работы с VC DRAM контроллер памяти должен «знать» ее систему команд, не имеющую прямой совместимости с командами SDRAM. Поддержка VC DRAM имеется далеко не во всех чипсетах — ее вводят, например, VIA и SiS, но фирма Intel эту память игнорирует. Механически и электрически модули VC DRAM совместимы с обычными модулями DRAM. Во время начального тестирования (POST) модули VC DRAM могут быть опознаны по информации, хранящейся в микросхеме EEPROM последовательной идентификации модуля, либо по поведению после инициализации.

Память VC DRAM по сравнению с другими типами динамической памяти обеспечивает меньшее среднее время задержки данных в многозадачных системах. Однако по пиковой скорости передачи она не имеет преимуществ перед SDRAM и проигрывает RDRAM и DDR SDRAM.

### 7.1.5. Модули динамической памяти

Динамическая память чаще всего применяется в виде модулей с разрядностью 1, 2,4 или 8 байт, которые могут устанавливаться пользователем без каких-либо приспособлений. Модули стандартизованы, поэтому обеспечивается взаимная совместимость.

- ◆ *SIPPn SIMM-30* — самые первые модули с однобайтной организацией, применялись вплоть до 486-х процессоров.
- ◆ *SIMM-72-pin* — 4-байтные модули, применявшиеся на системных платах для 486 и Pentium.

- ◆ *DIMM-168* — 8-байтные модули для Pentium и выше. Существует два поколения, существенно различных по интерфейсу. Модули DIMM 168-pin Buffered (1-го поколения), как и слоты для них, встречаются редко и с широко распространенными модулями DIMM 2-го поколения несовместимы даже механически (по ключам). Наиболее популярно второе поколение с микросхемами SDRAM. Различают модификации в зависимости от наличия буферов или регистров на управляющих сигналах: 'Unbuffered, Buffered и Registered.
- ◆ *DIMM-184* — 8-байтные модули DDR SDRAM для системных плат 6-7 поколений процессоров.
- ◆ *RIMM* — 2-байтные модули RDRAM для системных плат 6-7 поколений процессоров.
- ◆ *SO DIMM* (72 и 144-pin) и *50 RIMM* — малогабаритные варианты модулей (для блокнотных ПК).
- ◆ AИММ (AGP Inline Memory Module), они же GPA Card (Graphics Performance Accelerator) — 66-контактные 32- или 16-битные модули SDRAM, предназначенные для расширения памяти графических адаптеров, встроенных в системную плату.

Не пересчитывая контакты, отличить «короткие» SIMM от «длинных» и DIMM-модулей легко по их размеру: длина модуля SIMM-30 pin примерно 89 мм, SIMM-72 — 108 мм. Модули DIMM-168 и DIMM-184 имеют одинаковую длину около 134 мм (5,25"), но у 168-контактных модулей два ключа, а у 184-контактных — один (за счет чего больше контактов); кроме того, у DIMM-184 по две прорези по бокам, а не по одной. Модули RИММ имеют ту же длину, но легко отличимы по меньшему числу контактов — середина краевого разъема свободна от ламелей. У модулей RИММ микросхемы памяти закрыты пластиной радиатора. Кроме того, их левый ключ гораздо ближе к центру, чем у DIMM.

Модули памяти применяются и в принтерах (лазерных) — DIMM-168, 100-Pin DIMM, AИММ, SO DIMM-144, но иногда для них требуются и специальные модули (по конструктиву или параметрам).

Современные модули памяти имеют шину данных разрядностью 1,4 или 8 байт. Кроме основных информационных бит, модули могут иметь дополнительные контрольные биты с различной организацией.

- ◆ Модули без контрольных бит (*non Parity*) имеют разрядность 8,32 или 64 бита и допускают независимое побайтное обращение с помощью отдельных для каждого байта линий CAS#.
- ◆ Модули с контролем паритета (*Parity*) имеют разрядность 9, 36 или 72 бита и также допускают независимое побайтное обращение, контрольные биты по обращению приписаны к соответствующим байтам.
- ◆ Модули с генератором паритета (*Fake Parity, Parity Generator, Logical Parity*) так же допускают независимое побайтное обращение, логические генераторы паритета по чтению приписаны к соответствующим байтам. Действительного контроля памяти они не обеспечивают.

- ◆ Модули с контролем по схеме *ECC* имеют разрядность 36,40,72 или 80 бит. Обычно они допускают побайтное обращение к информационным битам, но контрольные биты у них привязаны к одному или нескольким сигналам *CAS#*, поскольку *ECC* подразумевает обращение сразу к целому слову.
- ◆ *ECC-Optimized* — модули, оптимизированные под режим *ECC*. От обычных модулей *ECC* они отличаются тем, что могут не обеспечивать побайтное обращение к информационным битам.
- ◆ *ECC-on-Simm* (*EOS*) — модули со встроенной схемой исправления ошибок. Каждый байт модуля имеет встроенные средства контроля и исправления ошибок, работающие прозрачно. Для системы модули функционируют как обычные паритетные — в случае обнаружения неисправимой ошибки они генерируют ошибочный бит паритета. Эти модули обеспечивают отказоустойчивость по памяти (*Kill Protected Memory*) для системных плат, поддерживающих только контроль паритета. По «благородству» поведения (делают больше, чем «говорят») они являются прямой противоположностью модулям с генератором паритета.

Набор сигналов модуля *SIMM* в основном совпадает с сигналами одиночных микросхем динамической памяти. Основные характеристики распространенных модулей приведены в табл. 7.5, более подробное описание — в следующих разделах.

Таблица 7.5. Основные характеристики модулей памяти

Модуль	Разрядность <sup>1</sup> , бит	Объем, Мбайт	Тип	Питание, В	Спецификация
<i>SIMM</i> -30. <i>SIPP</i>	8(9)	0,25-4	<i>FPM</i> , <i>EDO</i>	5	60,70, 80 не
<i>SIMM</i> -72	32 (36)	1-32	<i>FPM</i> , <i>EDO</i> , <i>BEDO</i>	5	50,60,70 не
<i>DIMM</i> -168-I	64(72,80)	8-256	<i>FPM</i> , <i>EDO</i>	5	50, 60,70 не
<i>DIMM</i> -168-II	64 (72, 80)	8-512	<i>FPM</i> , <i>EDO</i>	5,3,3	50,60,70 не
<i>DIMM</i> -168-III	64 (72, 80)	8-1024	<i>SDRAM</i>	3,3	<i>PC66</i> , <i>PC100</i> , <i>PC133</i>
<i>DIMM</i> -184	64 (72, 80)	128,256...	<i>DDR SDRAM</i>	2,5	<i>PC1600</i> , <i>PC2100</i>
<i>MMM</i>	32	4	<i>SDRAM</i>	3,3	166 МГц
100-Pin <i>DIMM</i>	32	4-128	<i>SDRAM</i>	3,3	100,125 МГц
100-Pin <i>DIMM</i>	32	4-32	<i>FPM</i> , <i>EDO</i>	3,3	50, 60 не
<i>SODIMM</i> -72	32(36)	4-32	<i>FPM</i> , <i>EDO</i>	3,3	50, 60 не
<i>SODIMM</i> -144	64(72)	32,64	<i>FPM</i> , <i>EDO</i>	3,3	50, 60 не
<i>SODIMM</i> -144	64(72)	32-256	<i>SDRAM</i>	3,3	66,100,125,133 МГц
<i>RIMM</i>	16(18)	64,96, 128,256	<i>RDRAM</i>	2,5	<i>PC600</i> , <i>PC700</i> , <i>PC800</i>

<sup>1</sup>В скобках указана разрядность с учетом битов паритета или *ECC*.

Спецификация быстродействия у разных типов памяти отражает различные параметры и выбирается исходя из технических и маркетинговых соображений. Для асинхронной памяти указывают время доступа (в наносекундах). Для памяти *SDRAM* указывается тактовая частота, на которой она работает с достойным зна-

чением латентности (на более высокой частоте она, возможно, и будет работать, но с большим значением CL). Обозначения PC66, PC 100 и PC 133 здесь тоже указывают на частоту (отсутствие обозначения соответствует 66 МГц — поначалу иных спецификаций не было), а также на соответствие спецификациям Intel. Для DDR SDRAM числа в спецификации отражают пиковую пропускную способность (Мбайт/с): PC1600 (8 байт, 2 x 100 МГц), PC2100 (8 байт, 2 x 133 МГц). Для RDRAM числа в названии (600, 700 и 800) обозначают округленную частоту (2 x 300, 2 x 356 и 2 x 400 МГц) схода двухбайтных данных с конвейера RDRAM. Таким образом, их пиковая производительность составляет 1200, 1424 и 1600 Мбайт/с.

*Маркировка модулей SDRAM*, согласно спецификациям Intel, имеет вид PCX-abc-defY, где X — частота, МГц; a - CL (Cas Latency, в тактах), b - Trcd (задержка RAS-CAS), c = Tgr (время предзаряда RAS), d - Tас (время доступа), e — ревизия последовательной идентификации (SPD), f — резервный символ, Y — символ архитектурных особенностей (R — признак наличия регистров; отсутствие символа означает отсутствие регистров и буферов). Временные характеристики задаются в десятках не, но Tас может задаваться и в наносекундах. Номер ревизии SPD может содержать как последнюю цифру, так и обе. Так, модуль PC100-322-620 работает на частоте 100 МГц при CL - 3 и Tас = 60 не, SPD ревизии 1.2. Но он может обозначаться и как PC 100-322-60120. Модуль PC100-322-620R имеет те же параметры, но еще снабжен и регистрами.

Существуют *адаптеры*, преобразующие форматы модулей SIMM (*SIMMVerter*, *SIMMSaver*). Они позволяют, например, сложить из четырех SIMM-30 один SIMM-72 или из двух односторонних SIMM-72 сложить один двусторонний. Трудно назвать такие конструктивные решения элегантными и надежными (появляется слишком много механических соединений и контактов), но их применение может быть оправданно при дефиците гнезд на плате. Или, например, при наличии четырех 4-мегабайтных модулей SIMM-30 можно сделать 16-мегабайтный SIMM-72. Следует помнить о повышенной нагрузке на шины, вносимой такими «супермодулями» с непомерным количеством микросхем и проводников.

### Идентификация модулей

Для автоматической идентификации наличия и типа установленного модуля применяются различные методы, основанные на считывании конфигурационной информации с модуля (параллельная или последовательная идентификация) или «исследовании» свойств модуля во время начального тестирования по включении питания.

Метод *параллельной идентификации* начал применяться с модулями SIPP и SIMM-30 фирмы IBM. В интерфейс этих модулей были введены два дополнительных вывода, и по заземленным (на модуле) сигналам системная плата могла распознать наличие и объем установленной памяти. В SIMM-72 для идентификации предназначались 4 вывода (для ECC-модулей — 5), которые должны были нести информацию об объеме, быстродействии и типе применяемой памяти. Этот метод не выдержал натиска новых типов памяти, поскольку описать их важнейшие

параметры четырьмя битами невозможно. В SO DIMM-72 используют 7 бит, в DIMM-168 первого поколения — 10, что тоже не решает проблем.

Новые модули памяти — DIMM-168 второго поколения, SO DIMM-144, DIMM-184 используют последовательную идентификацию (Serial Presence Detection). На модуль устанавливается микросхема специальной энергонезависимой памяти с последовательным доступом по двухпроводному интерфейсу I<sup>2</sup>C, хранящая исчерпывающую конфигурационную информацию. Формат конфигурационных данных стандартизован JEDEC, из доступных 256 байт под параметры пока определены только первые 32 и еще 32 зарезервированы, 64 байта отданы под информацию производителя (табл. 7.6). Основные параметры описываются в явном виде, например, временные — в наносекундах, количество бит адреса задается числами. Интерфейс I<sup>2</sup>C позволяет легко объединять его сигналы со всех модулей, что существенно проще, чем коммутация 4-10 линий параллельной идентификации. На разъеме модулей DIMM-168 выведены 3 бита адреса SA[0:2], что позволяет разводкой этих выводов адресовать до восьми модулей с объединенными линиями синхронизации и данных. При необходимости расширения следующие восемь модулей потребуют от контроллера (чипсета) еще только одной двунаправленной или выходной линии. Адрес в SO DIMM-144 фиксирован, так что двухпроводный интерфейс позволяет опрашивать только один модуль, а каждый следующий модуль потребует по одной дополнительной линии.

Байты 128-255 конфигурационной памяти свободны. Эту область в принципе можно занимать для пометки компьютера (точнее, модуля памяти) с целью привязки программного обеспечения к конкретному экземпляру PC. Однако при неосторожном использовании модулей с микросхемами без защиты от модификации случайная запись в ячейки 0-127 может привести к недоступности модуля памяти. «Оживить» его можно будет только записью корректных данных.

**Таблица 7.6.** Назначение байт последовательной идентификации

Байт	Назначение
<i>Стандартизованная информация о микросхеме</i>	
0	Число записанных байт конфигурационной памяти
1	Разрядность адреса микросхемы Serial PD (определяет объем конфигурационной памяти: 1—2 байта, 2 — 4 байта, ODh — 8 Кбайт)
2	Тип памяти: 00 — резерв, 01 — Std FPM, 02 — EDO, 03 — Pipelined Nibble (BEDO), 04 — SDRAM
3	Количество бит адреса строк в банке 1 (биты 0-3) и банке 2 (биты 4-7) по модулю 16 (0 — не определено, 1 — 1 или 16, 2 — 2 или 17 и т. д.) Если банки одинаковые, то биты 4-7 нулевые
4	Количество бит адреса столбцов (аналогично предыдущему)
5	Количество банков (рядов микросхем)
6-7	Разрядность данных с учетом контрольных бит (если менее 255, байт 7 — 0)
8	Уровень напряжения интерфейса: 0 — 7TL/5B, 01 — LVTTTL (не допускает 5 В), 02 — HSTL 1.5, 03 — SSTL 3.3, 04 — SSTL 2.5

*продолжение*<sup>^</sup>

Таблица 7.6 (продолжение)

Байт	Назначение
9	Для DRAM — RAS Access time (в наносекундах). Для SDRAM — минимальное время цикла (Tclk) для максимального значения CL (десятые доли не в BCD-коде)
10	Для DRAM — CAS Access time (в наносекундах). Для SDRAM — время доступа относительно тактового импульса (Tacc) аналогично предыдущему
11	Схема контроля: 00 — Non-Parity, 01 — Parity, 02 — ECC
12	Частота (тип) регенерации: 00 — Normal (распределенный цикл 156 мкс), 01 — Reduced 0.25x (39 мкс), 02 — Reduced 0.5x (78 мкс), 03 — Extended 2x (313 мкс), 04 — Extended 4x (625 мкс), 05 — Extended 8x (125 мкс). Бит7 является признаком саморегенерации (биты 6:0 кодируют те же периоды)
13	Разрядность микросхем основной памяти, бит. Бит 7 равен 1, если имеется второй банк с удвоенной разрядностью микросхем. Если банк один или оба банка одинаковы, бит 7 равен 0
14	Разрядность микросхем контрольных разрядов, бит (аналогично)
15-30	Детальное описание временных и организационных параметров SDRAM
31	Объемы банков (рядов микросхем): бит0 — 4 Мбайт, бит 1—8 Мбайт, бит7 — 512 Мбайт, единичное значение устанавливается в одном или нескольких (двух) битах 32-35
	Время предварительной установки и удержания входных сигналов 36-61 Резерв
62	Ревизия SPD (две BCD-цифры)
63	Контрольная сумма байт 0-62 по модулю 256
<i>Информация изготовителя</i>	
64-71	Идентификатор производителя по JEDEC
72	Код страны производителя
73-90	Код изделия (ASCII)
91-92	Код модификации
93-94	Дата изготовления (wwyy — неделя, год)
95-98	Серийный номер
99-127	Специальные данные изготовителя
126	Спецификация частоты (для Intel) DIMM SDRAM. Частота 66 МГц задается кодом 66п, более высокие значения — числом МГц (100 = 64h)
127	Детализация для SDRAM 100 МГц (для Intel)

### Модули SIMM-30, SIPP, SIMM-72

Модули SIMM (Single In-Line Memory Module) и SIPP (Single In-Line Pin Package) представляют собой небольшие печатные платы с односторонним краевым разъемом. Kontakтами модулей SIMM являются позолоченные (или покрытые специальным сплавом) площадки, расположенные на обеих поверхностях вдоль одной из сторон. Слово Single (одионый) в названии подразумевает, что пары площадок на обеих сторонах эквивалентны (электрически соединяются между собой). У малораспространенных модулей SIPP контакты штырьковые (pin — иголка); эти контакты при необходимости можно припаять к площадкам модулей SIMM (такие контакты

когда-то даже продавались в комплекте с модулями SIMM). Модули SIPP оказались непрактичными — их контакты не выдерживают транспортировки и многократной установки.

На модулях смонтированы микросхемы памяти в корпусах SOJ или TSOP, их адресные входы объединены. Количество и тип микросхем определяются требуемой разрядностью и объемом хранимых данных. Архитектура модулей обеспечивает возможность побайтного обращения, что существенно для записи (byte-write); выбор байт производится отдельным входом CAS# для каждого байта. Распространенные модули имеют напряжение питания 5 В, их параметры приведены в табл. 7.7.

Таблица 7.7. Организация модулей SIMM

Емкость, Мбайт	С паритетом		Без паритета	
	30-pin	72-pin	30-pin	72-pin
256 Кбайт	256Кx9	-	256 Кx 8	-
1	1Мx9	256 К x 36	1Мx8	256 К x 32
2	-	512Кx36	-	512Кx32
4	4Мx9	1 Мx36	4Мx8	1Мx32
8	-	2Мx36	-	2Мx32
16	-	4Мx36	-	4Мx32
32	-	8Мx36	-	8Мx32
64	-	16Мx36	-	16Мx32

По логической организации различают *односторонние* и *двусторонние* модули.

У «односторонних» модулей микросхемы смонтированы на одной (передней) поверхности, у «двусторонних» *двойной комплект* — *два банка* — микросхем смонтирован на обеих сторонах платы. Эти названия не совсем точны, но имеют прочные позиции и иностранное происхождение (single side и double side). Часто встречаются модули, у которых на второй стороне смонтировано несколько микросхем, дополняющих набор первой стороны до требуемой разрядности (чаще там размещаются контрольные биты). Такие модули являются логически односторонними. У «истинно двусторонних» на обеих сторонах обычно симметрично расположены одинаковые комплекты микросхем.

«Короткие», или *SIMM 30-pin*, модули SIMM (старый тип) имеют 30 печатных выводов (рис. 7.11) и однобайтную организацию. Разводка выводов у модулей фирмы IBM (для компьютеров IBM PS/2) отличается от общепринятых стандартных. Различия делают несовместимыми модули с объемом более 1 Мбайт: модули IBM могут быть двусторонними (2 Мбайт), стандартные — только односторонними. Малораспространенные модули SIPP имеют 30 штырьковых выводов и совпадают по разводке со стандартными модулями SIMM 30-pin (SIMM-30). Применение однобайтных модулей особенно в 32-битных системных платах сильно сковывает свободу выбора объема памяти. Назначение выводов SIMM-30 и SIPP приведено в табл. 7.8.



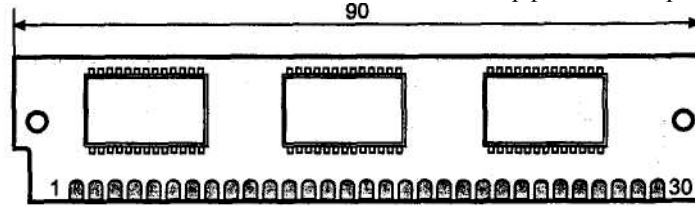


Рис. 7.11. Модули SIMM-30

Таблица 7.8. Назначение выводов модулей SIPP и SIMM 30-pin

Контакт	STD	IBM	Контакт	STD <sup>1</sup>	IBM <sup>2</sup>
1	+5 В	+5 В	16	DQ4	DQ4
2	CAS#	CAS#	17	MA8	MAS
3	DQ0	DQ0	18	MA9	MA9
4	MA0	MA0	19	MAЮ	RAS1#
5	MA1	MA1	20	DQ5	DQ5
6	DQ1	DQ1	21	WE#	WE#
7	MA2	MA2	22	GND	GND
8	MA3	MA3	23	DQ6	DQ6
9	GND	GND	24	N.C.	PD(GND)
10	DQ2	DQ2	25	DQ7	DQ7
11	MA4	MA4	26	PB-Out	PD(1M=GND)
12	MA5	MA5	27	RAS#	RASO#
13	DQ3	DQ3	28	CAS-Parity#	N.C.
14	MA6	MA6	29	PB-In	PB-(In/Out)
15	MA7	MA7	30	+5В	+5В

<sup>1</sup> STD-стандартный SIMM (SIPP).

<sup>2</sup> IBM - SIMM фирмы IBM.

«Длинные», или *SIMM 72-pin* (SIMM-72), модули SIMM имеют 72 печатных вывода (рис. 7.12, табл. 7.9) и 4-байтную организацию с возможностью независимого побайтного обращения по сигналам CASx#. По сигналам выборки строк биты данных делятся на два слова, DQ[0:15] выбираются сигналом RASO# для первого банка и RAS1 # для второго, DQ[ 16:31] выбираются соответственно сигналом RAS2\* и RAS3\*. В односторонних модулях (1,4,16,64 Мбайт — 1 банк) используется только одна пара сигналов выборки RASO\* и RAS2#, в двусторонних (2,8, 32 Мбайт — 2 банка) — две пары сигналов RAS#. Заметим, что использование всеми модулями обеих пар линий RAS# поддерживается не всеми системными платами. Контрольные биты модулей с паритетом по выборке приписываются к соответствующим<sup>1</sup> байтам, в ECC-модулях возможны различные варианты. Модули без паритета имеют разрядность 32 бит, с паритетом — 36 бит, модули ECC — 36 или 40 бит. Модули ECC-36 и ECC-40 (ECC-optimised) не допускают побайтного обращения и существенно отличаются от 32-битных и паритетных модулей.

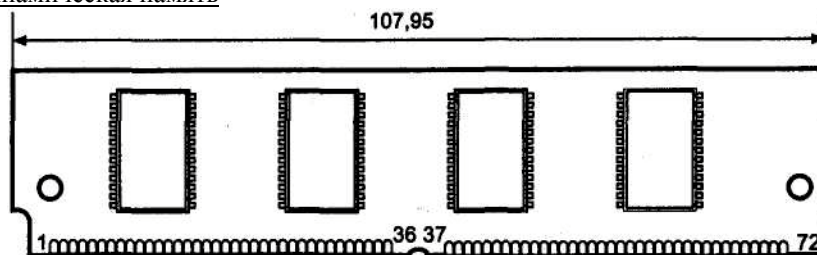


Рис. 7.12. Модули SIMM-72

Таблица 7.9. Назначение выводов модулей SIMM 72-pin

Контакт	Назначение для модулей x32, Parity/ECC <sup>1</sup>	Контакт	Назначение для модулей x32, Parity/ECC <sup>1</sup>
1	GND	37	PQ1/DQ19
2	DQ0/DQ0	38	PQ3/DQ20
3	DQ16/DQ1	39	GND
4	DQ1/DQ2	40	CAS0#
5	DQ17/DQ3	41 <sup>2</sup>	CAS2#/MA10
6	DQ2/DQ4	42 <sup>2</sup>	CAS3#/MA11
7	DQ18/DQ5	43	CAS1»
8	DQ3/DQ6	44	RAS0#
9	DQ19/DQ7	45	RAS1#
10	+5B	46 <sup>2</sup>	(OE1#)/DQ21
11 <sup>2</sup>	(CAS-Parity#)/PD5	47	WE#
12	MA0	48 <sup>2</sup>	Reserved/ECC
13	MA1	49	DQ8/DQ22
14	MA2	50	DQ24/DQ23
15	MA3	51	DQ9/DQ24
16	MA4	52	DQ25/DQ25
17	MA5	53	DQ10/DQ26
18	MA6	54	DQ26/DQ27
19*	MA10/OE#	55	DQ11/DQ28
20	DQ4/DQ8	56	DQ27/DQ29
21	DQ20/DQ9	57	DQ12/DQ30
2,2	DQ5/DQ10	58	DQ28/DQ31
23	DQ21/DQ11	59	+5B
24	DQ6/DQ12	60	DQ29/DQ32
25	DQ22/DQ13	61	DQ13/DQ33
26	DQ7/DQ14	62	DQ30/DQ34
27	DQ23/DQ15	63	DQ14/DQ35
28	MA7	64 <sup>2</sup>	DQ31/DQ36
29 <sup>2</sup>	MA11(OE0#)/DQ16	65 <sup>2</sup>	DQ15/DQ37

продолжение^

Таблица 7.9 (продолжение)

Контакт	Назначение для модулей x32, Parity/ECC <sup>1</sup>	Контакт	Назначение для модулей x32, Parity/ECC <sup>1</sup>
30	+5B	66 <sup>2</sup>	(OE2#)/DQ38
31	MA8	67	PD1
32	MA9	68	PD2
33 <sup>2</sup>	RAS3#/NC	69	PD3
34 <sup>2</sup>	RAS2#/NC	70	PD4
35	PQ2/DQ17	71 <sup>†</sup>	(OE3#)/DQ39
36	PQ0/DQ18	72	GND

<sup>1</sup> Модули ECC различных производителей могут отличаться по назначению выводов. Некоторые модули по выводам совпадают с паритетными, но могут различаться по связям контрольных бит с сигналами RASx# и CASx#.

<sup>2</sup> Могут существенно отличаться по назначению у модулей ECC. Сигналы DQ[36:39] имеются только в модулях ECC-40. В скобках приведены назначения выводов модулей фирмы IBM.

Сигналы модулей SIMM (табл. 7.10) в основном совпадают с сигналами микросхем динамической памяти. Для идентификации модулей предназначены сигналы PD[1:5]. По заземленным (на модуле) сигналам системная плата может распознать быстродействие (тип) и объем установленной памяти. Стандарт JEDEC для SIMM-72 определяет следующее назначение выводов (0 — заземлен, 1 — свободен):

- ◆ сигналы PD[1: 2] (контакты 67,68) — объем памяти модуля, Мбайт: 00=4,11=8,01=16, 10=32;
- ◆ сигналы PD[3:4] (контакты 69,70) - время доступа, нс: 00=100, 10=80, 01=70,11=60;
- ◆ сигнал PD5 может являться признаком ECC-модуля (заземленный контакт).

Таблица 7.10. Сигналы модулей SIMM

Сигнал	Назначение
MAi	Multiplexed Address — мультиплексированные линии адреса. Во время спада сигнала RAS# на этих линиях присутствует адрес строки, во время спада CAS# — адрес столбца. Модули SIMM объемом 16 Мбайт могут быть с симметричной (square — квадратной) организацией — 11 бит адреса строк и 11 бит адреса колонок или асимметричной — 12x10 бит соответственно
DQx	Data Bit — биты данных (объединенные входы и выходы) PQx
Parity Bit	— бит паритета x-го байта
PB-In, PB-Out	Parity Bit Input, Output — вход и выход микросхемы бита паритета (для SIPP PB-Out и SIMM-30). Для хранения паритета в этих модулях всегда используются микросхемы с однобитной организацией, у которых вход и выход разделен. Обычно эти контакты на модуле соединены
WE#	Write Enable — разрешение записи. При низком уровне сигнала во время спада CAS# выполняется запись в ячейку. Переход WE# в низкий уровень и обратно при высоком уровне CAS# переводит выходной буфер EDO DRAM в высокоимпедансное состояние
RASx#	Стробы выборки строк. Сигналы RAS0* и RAS1 # используются соответственно для бит [0:15] и [16:31] первого банка, RAS1#nRAS3#— для бит [0:15] и [16:31] второго банка

Таблица 7.10. Сигналы модулей SIMM

Сигнал	Назначение
CASx#	Стробы выборки столбцов, отдельные для каждого байта: CAS0# — DQ[0:7], PQ0; CAS1# — DQ[8:15], PQ1; CAS2# — DQ[16:23], PQ2; CAS3# — DQ[24:31], PQ3. В ECC-модулях возможно обращение только ко всему модулю по сигналам CAS0* и CAS1 #
CAS-Parity*	Строб выборки столбцов для контрольных разрядов (редко используемый вариант)
OEx#	Output Enable — разрешение открытия выходного буфера. Эти выводы на системной плате обычно соединяются с логическим нулем, а для управления буфером используются сигналы RAS#, CAS# и WE#. На некоторых модулях SIMM могут отсутствовать
PD[1:5]	Presence Detect — индикаторы присутствия (обычно не используются) N.C.
No Connection	— свободный вывод

### Модули DIMM-168 и DIMM-184

Модуль памяти *DIMM-168* (Dual-In-line-Memory Module) имеет 168 независимых печатных выводов, расположенных с обеих сторон (контакты 1-84 — с фронтальной стороны, 85-168 — с тыльной). Разрядность шины данных — 8 байт, организация рассчитана на применение в компьютерах с четырех- и восьмибайтной шиной данных. Конструкция и интерфейс модулей соответствует стандарту JEDEC 21-C. Модули устанавливаются на плату вертикально в специальные разъемы (слоты) с ключевыми перегородками, задающими допустимое питающее напряжение и тип (поколение) применимых модулей. Модули выпускаются для напряжения питания 3,3 и 5 В. Вид модулей и сочетания ключей представлены на рис. 7.13. Толщина модулей с микросхемами в корпусах SOJ не превышает 9 мм, в корпусах TSOP - 4 мм.

По внутренней архитектуре модули близки к SIMM-72, но имеют удвоенную разрядность и, соответственно, удвоенное количество линий CAS#. Также удвоено число сигналов разрешения записи и разрешения выходных буферов, что позволяет организовывать модули в виде двух 4-байтных банков с возможностью их чередования (Bank Interleaving). Модули могут иметь разрядность 64, 72 или 80 бит, дополнительные разряды 72-битных модулей организуются либо по схеме контроля паритета (приписываясь к соответствующим байтам), либо по схеме ECC; 80-битные — только по схеме ECC.

Модули *DIMM* первого поколения (по IBM) были ориентированы на асинхронную память (FPM, EDO и BEDO); по архитектуре они напоминают SIMM-72. В модулях применяется параллельная идентификация — параметры быстродействия и объема передаются через 8 буферизованных выводов идентификации (Presence Detect pins). Модули первого поколения не получили широкого распространения, поскольку не принесли принципиальных новшеств в подсистему памяти.

*Модули второго поколения* отличаются тем, что позволяют использовать микросхемы как асинхронной (FPM и EDO), так и синхронной динамической памяти

(SDRAM). Внешне они похожи на модули первого поколения, но обличаются ключом, не допускающим ошибочную установку. Унифицированное назначение выводов позволяет в одни и те же слоты устанавливать как модули DRAM, так и SDRAM. Нумерация бит данных единая для всех типов организации — контрольные биты CBx имеют отдельную нумерацию, их наличие зависит от организации (паритет, ECC-72, ECC-80).

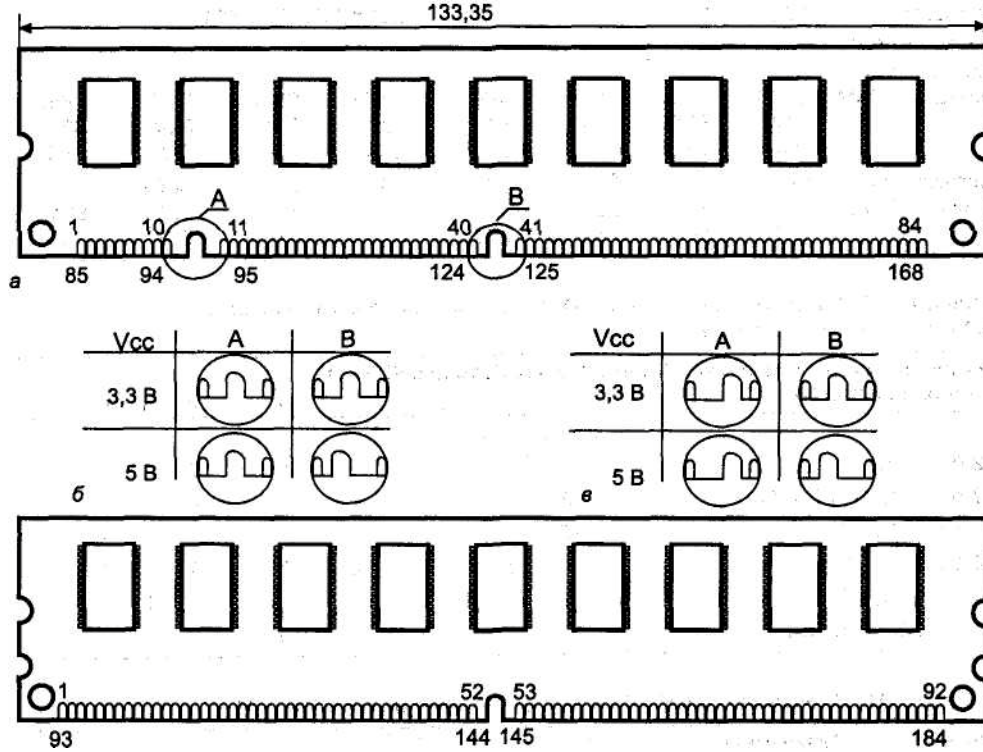


Рис. 7.13. Модули DIMM: а — вид модуля DIMM-168, б — ключи для модулей первого поколения, в — ключи для модулей второго поколения, г — вид модуля DIMM-184

Модули с любой организацией используют побайтное распределение информационных бит по сигналам CASx# (табл. 7.11), распределение контрольных бит представлено в табл. 7.12. Младший бит адреса приходит по одной линии на все микросхемы модуля. Сигналы управления модулей SDRAM значительно отличаются от модулей DRAM. Исполняемая операция SDRAM определяется сигналами\* RAS#, CAS# и WE#, синхронизируемыми по фронту соответствующих сигналов CKx. Назначение сигналов модулей приведено в табл. 7.13, назначение выводов модулей DRAM — в табл. 7.14, SDRAM — в табл. 7.15.

Таблица 7.11. Организация информационных и управляющих сигналов для модулей DIMM-168 второго поколения

Линии CAS# (DQMB для SDRAM)	0	1	2	3	4	5	6	7
Биты данных	0–7	8–15	16–23	24–31	32–39	40–47	48–55	56–63
Сигналы 0 для банка DRAM	OE0#, WE0#, RAS0#				OE2#, WE2#, RAS2#			
Сигналы для банка 1 DRAM	OE0#, WE0#, RAS1#				OE2#, WE2#, RAS3#			
Сигналы для банка 0 SDRAM	CKE0 S0# CK0	CKE0 S0# CK1	CKE0 S2# CK2	CKE0 S2# CK3	CKE0 S0# CK0	CKE0 S0# CK1	CKE0 S2# CK2	CKE0 S2# CK3
Сигналы для банка 1 SDRAM	CKE1 S1# CK0	CKE1 S1# CK1	CKE1 S3# CK2	CKE1 S3# CK3	CKE1 S1# CK0	CKE1 S1# CK1	CKE1 S3# CK2	CKE1 S3# CK3

Таблица 7.12. Связь контрольных бит с управляющими сигналами для модулей DIMM-168 второго поколения

Организация (разрядность микросхем DRAM)	Линии CAS# (DQMB для SDRAM)							
	0	1	2	3	4	5	6	7
	Контрольные биты							
72-бит Parity	0	1	2	3	4	5	6	7
72-бит ECC, (x4 x16/x4)	–	0–3	–	–	–	4–7	–	–
72-бит ECC, (x8)	–	0–7	–	–	–	–	–	–
72-бит ECC, (x18)	0	1	2	3	4	5	6	7
80-бит ECC, (x4)	–	0–3	8–11	–	–	4–7	12–15	–
80-бит ECC, (x8, x16)	–	0–7	–	–	–	8–15	–	–

Таблица 7.13. Сигналы модулей DIMM-168 второго поколения и DIMM-184

Сигнал	Назначение
<i>Общие сигналы для FPM, EDO, BEDO и SDRAM</i>	
RAS[0:3]#, RAS#	Row Address Strobe — стробы выборки строк
CAS[0:7]#, CAS#	Column Address Strobe — стробы выборки столбцов
WE0#, WE2#	Read/write Input — сигналы разрешения записи, отдельные для банков
OE0#, OE2#	Output Enable — сигналы разрешения выходных буферов, отдельные для банков
A[0:13]	Address Inputs — мультиплексированная шина адреса
DQ[0:63]	Data Input/Output — биты данных
CB[0:15]	Check Bit Data Input/Output — контрольные биты, отсутствуют в 64-битных модулях. В 72-битных модулях отсутствуют CB[8:15]
SCL	Serial Presence Detect Clock — синхронизация интерфейса идентификации I <sup>2</sup> C

продолжение\*?

Таблица 7.13 (продолжение)

Сигнал	Назначение
SDA	Serial Presence Detect Data — данные интерфейса идентификации I <sup>2</sup> C
SA[0:2]	Serial Presence Detect Address — адрес модуля в интерфейсе I <sup>2</sup> C, задается коммутацией выводов гнезд для модулей на уровне логических «0» и «1»
WP	Write Protect — защита записи в EEPROM
VCC	Power — питание (+5 или +3,3 В)
VSS	Ground — общий провод
NC	No Connect — неиспользуемый (свободный) контакт
DU	Don't Use — запрещенный к использованию контакт
<i>Специфические сигналы SDRAM</i>	
DQMB0– DQMB7	Data Mask Byte — маски байт (синхронизируются по фронту СК). Высокий уровень во время операции чтения переводит выходные буферы соответствующего байта в высокоимпедансное состояние с задержкой на 2 такта, операция записи блокируется без задержки
S0#, S1#, S2#, S3#	Select — разрешение (низким уровнем) декодирования команд микросхемами SDRAM соответствующих банков. При высоком уровне новые команды игнорируются, но выполнение предыдущей не прерывается
CK[0:3]	Clock Inputs — тактовые импульсы системной шины, положительный перепад синхронизируют все входные сигналы (кроме СKE)
SKE0, SKE1	Clock Enables — разрешение синхронизации (высокий уровень) для банков микросхем. Низкий уровень переводит в режим пониженного потребления или саморегенерации
A[0:9], A[11:13] A10/AP	Address Inputs, Address Input10/Autoprecharge — в цикле команды активации банка A[0:13] определяют адрес строки (по подъему СК). В цикле команды чтения или записи A[0:8] определяют адрес столбца, AP используется для указания (высоким уровнем) на операцию автопредзаряда (autoprecharge) банка A (BA0=0) или B (BA1=1) по окончании текущего пакетного цикла. В цикле команды предзаряда при высоком уровне AP предзаряд осуществляется в обоих банках, при низком — только в банке, определяемом линией BA0
BA0, BA1	SDRAM Bank Address — выбор внутреннего банка микросхемы SDRAM (использует линии, назначенные на адреса A11, A12 модулей DRAM)
REGE	Register Enable — разрешение синхронной работы регистров управляющих и адресных сигналов. При высоком уровне регистр защелкивает сигналы по фронту СК, а микросхемы памяти зафиксируют эти значения в следующем такте. При низком уровне регистр работает в режиме буфера (допустимо лишь для 66 МГц)
<i>Дополнительные сигналы модулей DDR SDRAM</i>	
DQS[0:17]	Двунаправленные стробы данных, формируемые источником
CK#	Инверсный вход синхронизации (пара к СК)
V <sub>REF</sub>	Вход опорного напряжения интерфейса SSTL_2
RESET#	Вход асинхронного сброса регистра
V <sub>DDQ</sub>	Питание выходных буферов микросхем
V <sub>DD</sub>	Питание ядра микросхем
V <sub>DDSPD</sub>	Питание микросхемы последовательной идентификации
V <sub>DDID</sub>	Вход V <sub>DD</sub> identification flag

В модулях SDRAM вместо отдельных сигналов RAS[0:3]# для выбора банков (рядов микросхем) используются сигналы S0#, S1#, S2# и S3#; вместо CAS[0:7]# для выбора байтов — сигналы DQMB0–DQMB7; сигналы WE2#, OEO# и OE2# не используются.

Таблица 7. 14. Назначение выводов DIMM-168 DRAM второго поколения

Контакт	Цепь	Контакт	Цепь	Контакт	Цепь	Контакт	Цепь
1	VSS	85	VSS	43	VSS	127	VSS
2	DQ0	86	DQ32	44	OE2#	128	DU
3	DQ1	87	DQ33	45	RAS2#	129	RAS3#
4	DQ2~	88	DQ34	46	CAS2#	130	CAS6#
5	DQ3	89	DQ35	47	CAS3#	131	CAS7#
6	VCC	90	VCC	48	WE2#	132	DU
7	DQ4	91	DQ36	49	VCC	133	VCC
8	DQ5	92	DQ37	50	CB10	134	CB14
9	DQ6	93	DQ38	51	CB11	135	CB15
10	DQ7	94	DQ39	52	CB2	136	CB6
11	DQ8	95	DQ40	53	CB3	137	CB7
12	VSS	96	VSS	54	VSS	138	VSS
13	DQ9	97	DQ41	55	DQ16	139	DQ48
14	DQ10	98	DQ42	56	DQ17	140	DQ49
15	DQ11	99	DQ43	57	DQ18	141	DQ50
16	DQ12	100	DQ44	58	DQ19	142	DQ51
17	DQ13	101	DQ45	59	VCC	143	VCC
18	VCC	102	VCC	60	DQ20	144	DQ52
19	DQ14	103	DQ46	61	NC	145	NC
20	DQ15	104	DQ47	62	DU	146	DU
21	CBO	105	CB4	63	NC	147	NC
22	CB1	106	CBS	64	VSS	148	VSS
23	VSS	107	VSS	65	DQ21	149	DQ53
24	CB8	108	CB12	66	DQ22	150	DQ54
25	CB9	109	CB13	67	DQ23	151	DQ55
26	VCC	110	VCC	68	VSS	152	VSS
27	WEO#	111	DU	69	DQ24	153	DQ56
28	CASO#	112	CAS4#	70	DQ25	154	DQ57
29	CAS1#	113	CAS5#	71	DQ26	155	DQ58
30	RASO#	114	RAS1#	72	DQ27	156	DQ59
31	OEO#	115	DU	73	VCC	157	VCC
32	VSS	116	VSS	74	DQ28	158	DQ60
33	AO	117	A1	75	DQ29	159	DQ61
34	A2	118	A3	76	DQ30	160	DQ62
35	A4	119	A5	77	DQ31	161	DQ63
36	A6	120	A7	<b>78</b>	VSS	162	VSS
37	A8	121	A9	79	NC	163	NC
38	A10	122	A11	80	NC	164	NC
39	A12	123	A13	81	NC	165	SAO
40	VCC	124	VCC	82	SDA	166	SA1
41	VCC	125	DU	83	SCL	167	SA2
42	DU	126	DU	84	VCC	168	VCC



Таблица 7.15. Назначение выводов DIMM-168 SDRAM

Контакт	Цепь	Контакт	Цепь	Контакт	Цепь	Контакт	Цепь
1	VSS	85	VSS	43	VSS	127	VSS
2	DQ0	86	DQ32	44	DU <sup>2</sup>	128	CKE0
3	DQ1	87	DQ33	45	S2#	129	S3#
4	DQ2	88	DQ34	46	DQMB2	130	DQMB6
5	DQ3	89	DQ35	47	DQMB3	131	DQMB7
6	VCC	90	VCC	48	DU <sup>2</sup>	132	A13
7	DQ4	91	DQ36	49	VCC	133	VCC
8	DQ5	92	DQ37	50	CB10	134	CB14
9	DQ6	93	DQ38	51	CB11	135	CB15
10	DQ7	94	DQ39	52	CB2	136	CB6
11	DQ8	95	DQ40	53	CB3	137	CB7
12	VSS	96	VSS	54	VSS	138	VSS
13	DQ9	97	DQ41	55	DQ16	139	DQ48
14	DQ10	98	DQ42	56	DQ17	140	DQ49
15	DQ11	99	DQ43	57	DQ18	141	DQ50
16	DQ12	100	DQ44	58	DQ19	142	DQ51
17	DQ13	101	DQ45	59	VCC	143	VCC
18	VCC	102	VCC	60	DQ20	144	DQ52
19	DQ14	103	DQ46	61	NC	145	NC
20	DQ15	104	DQ47	62	Vref	146	Vref
21	CBO	105	CB4	63	CKE1	147	REGE
22	CB1	106	CBS	64	VSS	148	VSS
23	VSS	107	VSS	65	DQ21	149	DQ53
24	CB8	108	CB12	66	DQ22	150	DQ54
25	CB9	109	CB13	67	DQ23	151	DQ55
26	VCC	110	VCC	68	VSS	152	VSS
27	we#	111	CAS#	69	DQ24	153	DQ56
28	DQMBO	112	DQMB4	70	DQ25	154	DQ57
29	DQMB1	113	DQMB5	71	DQ26	155	DQ58
30	S0#	114	S1#	72	DQ27	156	DQ59
31	DU <sup>2</sup>	115	RAS#	73	VCC	157	VCC
32	VSS	116	VSS	74	DQ28	158	DQ60
33	AO	117	A1	75	DQ29	159	DQ61
34	A2	118	A3	76	DQ30	160	DQ62
35	A4	119	AS	77	DQ31	161	DQ63
36	A6	120	A7	78	VSS	162	VSS
37	AS	121	A9	79	CK2	163	CK3
38	AЮ(AP)	122	BAO	80	NC <sup>1</sup>	164	NC
39	BA1	123	A11	81	WP	165	SAO
40	VCC	124	VCC	82	SDA	166	SA1
41	VCC	125	CK1	83	SCL	167	SA2
42	CKO	126	A12	84	VCC	168	VCC

<sup>1</sup> NC — не подключен<sup>2</sup> DU-н *it* использовать!

## Т.1. Динамическая память

В модулях, начиная со второго поколения, применена *последовательная идентификация параметров* на двухпроводном интерфейсе (PC) для чтения атрибутов (идентификации) из специальной конфигурационной памяти (обычно EEPROM 24C02), установленной на модулях.

*168-pin Unbuffered DIMM* — модули, у которых все цепи не буферизованы (одноименные адресные и управляющие сигналы микросхем соединены параллельно и заводятся прямо с контактов модуля). Эти модули сильнее нагружают шину памяти, но позволяют добиться максимального быстродействия. Они предназначены для системных плат с небольшим (1-4) количеством слотов DIMM или имеющих шину памяти, буферизованную на плате. Модули выполняются на микросхемах DRAM или SDRAM. Высота модулей не превышает 51 мм. Объем 8-512 Мбайт.

*168-pin Registered DIMM* — модули синхронной памяти (SDRAM), у которых адресные и управляющие сигналы буферизованы регистрами, синхронизируемыми тактовыми импульсами системной шины. По виду этот тип DIMM легко отличим — кроме микросхем памяти и EEPROM на них установлено несколько микросхем регистров-защелок. За счет регистров эти модули меньше нагружают шину памяти, что позволяет набирать больший объем памяти. Применение регистров повышает точность синхронизации и, следовательно, — тактовую частоту. Однако регистр вносит дополнительный такт задержки. Кроме того, на модулях может быть установлена микросхема ФАПЧ (PLL), формирующая тактовые сигналы для микросхем памяти и регистров-защелок. Это делается для разгрузки линий синхронизации, причем в отличие от обычной буферизации сигнала, вводящей задержку между входом и выходом, схема PLL обеспечивает синфазность выходных сигналов (их на выходе PLL несколько, каждый для своей группы микросхем) с опорным сигналом (линия СКО). Модули на 64 Мбайт могут быть и без схем PLL — в них линии СК[0:3] разводятся прямо на свои группы микросхем памяти. Регистры могут быть переведены в режим асинхронных буферов (только на 66 МГц), для чего на вход REGE нужно подать низкий уровень. Для модулей на 66 МГц возможна замена регистров асинхронными буферами.

*Модули DIMM-184* предназначены для микросхем DDR SDRAM. По габаритам они аналогичны модулям DIMM-168, но у них имеются дополнительные вырезы по бокам (см. рис. 7.13, г) и отсутствует левый ключ. Разрядность — 64 или 72 бит (ECC), имеются варианты с регистрами в адресных и управляющих цепях (Registered DDR SDRAM) и без них. Напряжение питания — 2,5 В. Идентификация последовательная. Состав сигналов в основном повторяет набор для DIMM SDRAM, назначение выводов приведено табл. 7.16. Модули отличаются большим количеством стробирующей сигналов DQSx — по линии на каждые 4 бита данных (DQS8 и DQS17 используются для стробирования контрольных битов). Вход тактовой частоты только один, но дифференциальный — раздачу сигналов по микросхемам памяти и регистрам осуществляет микросхема DLL.

Таблица 7 .16. Назначение выводов DIMM-184 DDR SDRAM

Контакт	Цепь	Контакт	Цепь	Контакт	Цепь	Контакт	Цепь
1	VREF	47	DQS8	93	VSS	139	VSS
2	DQO	48	AO	94	DQ4	140	DQS17
3	VSS	49	CB2	95	DQ5	141	A10
4	DQ1	50	VSS	96	VDDQ	142	CB6
5	DQSO	51	CB3	97	DQS9	143	VDDQ
6	DQ2	52	BA1	98	DQ6	144	CB7
7	VDD	53	DQ32	99	DQ7	145	VSS
8	DQ3	54	VDDQ	100	VSS	146	DQ36
9	NC	55	DQ33	101	NC	147	DQ37
10	RESET*	56	DQS4	102	NC	148	VDD
11	VSS	57	DQ34	103	A13	149	DQS13
12	DQ8	58	VSS	104	VDDQ	150	DQ38
13	DQ9	59	BAO	105	DQ12	151	DQ39
14	DQS1	60	DQ35	106	DQ13	152	VSS
15	VDDQ	61	DQ40	107	DQS10	153	DQ44
16	DU	62	VDDQ	108	VDD	154	RAS#
17	DU	63	WE#	109	DQ14	155	DQ45
18	VSS	64	DQ41	110	DQ15	156	VDDQ
19	DQ10	65	CAS#	111	CKE1	157	S0#
20	DQ11	66	VSS	112	VDDQ	158	S1#
21	CKEO	67	DQS5	113	BA2	159	DQS14
22	VDDQ	68	DQ42	114	DQ20	160	VSS
23	DQ16	69	DQ43	115	A12	161	DQ46
24	DQ17	70	VDD	116	VSS	162	DQ47
25	DQS2	71	DU	117	DQ21	163	DU
26	VSS	72	DQ48	118	A11	164	VDDQ
27	A9	73	DQ49	119	DOS11	165	DQ52
28	DQ18	74	VSS	120	VDD	166	DQ53
29	A7	75	DU	121	DQ22	167	FETEN
30	VDDQ	76	DU	122	A8	168	VDD
31	DQ19	77	VDDQ	123	DQ23	169	DQS15
32	A5	78	DQS6	124	VSS	170	DQ54
33	DQ24	79	DQSO	125	A6	171	DQ55
34	VSS	80	DQ51	126	DQ28	172	VDDQ
35	DQ25	81	VSS	127	DQ29	173	NC
36	DQS3	82	VDDID	128	VDDQ	174	DQ60
37	A4	83	DQ56	129	DQS12	175	DQ61
38	VDD	84	DQ57	130	A3	176	VSS
39	DQ26	85	VDD	131	DQSO	177	DOS16
40	DQ27	86	DQS7	132	VSS	178	DQ62
41	A2	87	DQ58	133	DQ31	179	DQ63
42	VSS	88	DQ59	134	CB4	180	VDDQ
43	A1	89	VSS	135	CBS	181	SAO
44	CBO	90	WP	136	VDDQ	182	SA1
45	CB1	91	SDA	137	CKO	183	SA2
46	VDD	92	SCL	138	CKO#	184	VDDSPD

## Модули RIMM

Модули *RIMM* (Rambus Interface Memory Module), по форме похожие на обычные модули памяти (рис. 7.14), специально предназначены для памяти RDRAM. У них 30-проводная шина проходит вдоль модуля слева направо, и на эту шину без ответвлений напаиваются микросхемы RDRAM в корпусах BGA. Сигналы интерфейса модуля (табл. 7.17) соответствуют сигналам канала Rambus, но в их названии имеется еще приставка L (Left) и R (Right) для левого и правого вывода шины соответственно. Модуль RIMM содержит до 16 микросхем RDRAM, которые всеми выводами (кроме двух) соединяются параллельно. Микросхемы памяти закрыты пластиной радиатора. В отличие от SIMM и DIMM, у которых объем памяти кратен степени числа 2, модули RIMM могут иметь более равномерный ряд объемов — в канал RDRAM память можно добавлять хоть по одной микросхеме.

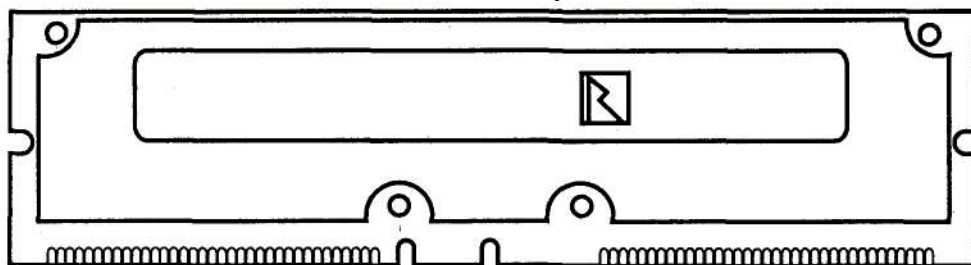


Рис. 7.14. Модули RIMM

Таблица 7.17. Назначение выводов RIMM

Контакт	Цепь	Тип	Назначение
116, 32	SIO0, SIO1	I/O CMOS	Serial I/O — последовательные данные обмена с управляющими регистрами
34, 35, 42, 51, 53, 118, 119, 126, 135, 137	VDD		Питание +2,5 В
1, 3, 5, 7, 9, 11, 13, 15, 17, 19, 21, 23, 25, 27, 29, 31, 33, 37, 38, 39, 41, 45, 48, 49, 52, 54, 56, 58, 60, 62, 64, 66, 68, 70, 72, 74, 76, 78, 80, 82, 84, 85, 87, 89, 91, 93, 95, 97, 99, 101, 103, 105, 107, 109, 111, 113, 115, 117, 121, 123, 125, 129, 132, 133, 138, 140, 142, 144, 146, 148, 150, 152, 154, 156, 158, 160, 162, 164, 166, 168	GND		Общий
2, 86, 4, 88, 6, 90, 8, 92, 10	LDQA8...LDQA0	I/O RSL	Шина данных A
94	LCFM	I RSL	Синхронизация (+) от ведущего устройства (для приема данных)

продолжение\*?

Таблица 7.17 (продолжение)

Контакт	Цепь	Тип	Назначение
96	LCFMN	I RSL	Синхронизация (-) от ведущего устройства (для приема данных) Пороговый уровень сигналов RSL (1,8 В)
40,124 12	VREF LCTMN	IRSL	Синхронизация (-) к ведущему устройству (для передачи данных)
14	LCTM	IRSL	Синхронизация (+) к ведущему устройству (для передачи данных)
98,16,100	LROW2...LROW ⌞	IRSL	Шина строк (для управляющей и адресной информации)
18,102, 20,104,22	LCOL4... LCOLO	I RSL	Шина столбцов (для управляющей и адресной информации)
114,30,112,28,110, 26,108,24,106	LOQB8...LOQBO	I/ORSL	Шина данных В
120	LCMD	ICMOS	Последовательные команды (для обмена с управляющими регистрами). Используется и для управления энергопотреблением
36	LSCK	I CMOS	Синхронизация последовательных команд и данных (для обмена с управляющими регистрами) Шина данных А
83,167, 81,165,79, 159	RDQA8...RDQAO	I/O RSL	
157	RCFM	IRSL	Синхронизация (+) от ведущего устройства (для приема данных)
73	RCFMN	IRSL	Синхронизация (-) от ведущего устройства (для приема данных)
71	RCTMN	IRSL	Синхронизация (-) к ведущему устройству (для передачи данных)
155,69,153	RCTM	IRSL	Синхронизация (+) к ведущему устройству (для передачи данных)
67,151,65,148,63	RROW2...RROW ⌞	IRSL	Шина строк (для управляющей и адресной информации)
139,55,141,57,143, 59,145,61,147	RCOL4...RCOLO	I RSL	Шина столбцов (для управляющей и адресной информации)
134	RDQB8...RDQBO	I/ORSL	Шина данных В
134	RCMD	ICMOS	Последовательные команды (для обмена с управляющими регистрами). Используется и для управления энергопотреблением
50	RSCK	I CMOS	Синхронизация последовательных команд и данных (для обмена с управляющими регистрами)
46	RSCK	I CMOS	Синхронизация последовательной идентификации

271

## 7.1. Динамическая память

Контакт	Цепь	Тип	Назначение
47	SDA	I/O CMOS	Данные последовательной идентификации
131,130	SA1.SAO	ICMOS	Адрес последовательной идентификации
43,44,127,128	VT		Питание терминаторов (1,4 В)

## Модули SO DIMM-72 pin

72 pin SO DIMM (Small-Outline-Dual-Inline-Memory Module) — малогабаритный (длина 2,35" — 60 мм) модуль с двусторонним 72-контактным разъемом, нечетные контакты расположены с фронтальной стороны, четные — с тыльной (рис. 7.15, табл. 7.18 и 7.19). Модули

комплекуются микросхемами DRAM в корпусах TSOP, емкость 2-32 Мбайт, разрядность данных — 32 или 36 бит (с контролем паритета). 36-битные модули отличаются только наличием дополнительных бит PQx. Память организована в виде двух двухбайтных слов с возможностью побайтного обращения и предназначена для двух- и четырехбайтных применений. Информация об объеме, организации, адресации, быстродействии и регенерации передается через семь линий параллельной идентификации:

- ◆ PD7 — регенерация: 1=стандартная, 0=расширенная или саморегенерация;
- ◆ PD6, PD5 - время доступа: 00=50 нс, 10=70 нс, 11=60 нс;
- ◆ PD[4:1] — организация.

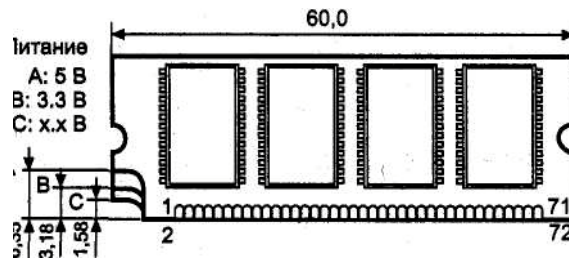


Рис. 7.15. Модули SO DIMM-72 pin

Таблица 7.18. Организация информационных и управляющих сигналов модулей SO DIMM-72

Линии CAS#	CAS0#	CAS1#	CAS2#	CAS3#
Биты данных и паритета	DQ[0:7], PQ8	DO[9:15], PO1	DO[18:25], PO2	DO[27:34], PO35
Выбор банка 0	RAS 0#:	; RAS2#		
Выбор банка 1	RAS1*	RAS3#		

Таблица 7. 19 Назначение выводов SO DIMM-72 pin

Контакт	Цепь	Контакт	Цепь
1	VSS	2	DQ0
3	DQ1	4	DQ2
5	DQ3	6	DQ4
7	DQ5	8	DQ6
9	DQ7	10	VCC
11	PD1	12	A0
13	A1	14	A2
15	A3	16	A4
17	A5	18	A6
19	A10	20	PQ8
21	DQ9	22	DQ10
23	DQ11	24	DQ12
25	DQ13	26	DQ14
27	DQ15	28	A7
29	A11	30	VCC
31	A8	32	A9
33	RAS3*	34	RAS2#
35	DQ16	36 <sup>1</sup>	PQ17
37	DQ18	38	DQ19
39	VSS	40	CAS0#
41	CAS2#	42	CAS3#
43	CAS1#	44	RAS0#
45	RAS1#	46	A12
47	WE#	48	A13
49	DQ20	50	DQ21
51	DQ22	52	DQ23
53	DQ24	54	DQ25
55'	PQ26	56	DQ27
57	DQ28	58	DQ29
59	DQ31	60	DQ30
61	VCC	62	DQ32
63	DQ33	64	DQ34
65'	PQ35	66	PD2
<sup>4</sup> 67	PD3	68	PD4
69	PD5	70	PD6
71	PD7	72	VSS

<sup>1</sup> У 32-битных модулей контакт свободен

### Модули SO DIMM-144 pin

Модуль *144pin SO DIMM* — малогабаритный модуль (длина 2,35" — 60 мм) с двусторонним 144-контактным разъемом (рис. 7.16, табл. 7.20), емкость 8-64 Мбайт, разрядность данных — 64 или 72 бит ECC. Модули обеспечивают побайтное обращение по сигналам CAS[0:7]#, сигнал RAS0\* выбирает банк 0, сигнал RAS1 # — банк 1 (при его наличии). Напряжение питания — 5 или 3,3 В, механический ключ напряжения питания расположен между контактами 59-60 и 61-62. Нечетные контакты находят-

ся с фронтальной стороны, четные — с тыльной. Идентификация последовательная. Модули могут содержать микросхемы как DRAM, так и SDRAM, объем 8-256 Мбайт.

Таблица 7.20. Назначение выводов модулей SO DIMM-144 pin

Контакт	Цепь <sup>1</sup>	Контакт	Цепь <sup>1</sup>	Контакт	Цепь <sup>1</sup>	Контакт	Цепь <sup>1</sup>
1	VSS	2	VSS	71	RAS1#	72	NC
3	DQ0	4	DQ32	73	OE	74	NC
5	DQ1	6	DQ33	75	VSS	76	VSS
7	DQ2	8	DQ34	77	CB2	78	CB6
9	DQ3	10	DQ35	79	CB3	80	CB7
11	VCC	12	VCC	81	VCC	82	VCC
13	DQ4	14	DQ36	83	DQ16	84	DQ48
15	DQ5	16	DQ37	85	DQ17	86	DQ49
17	DQ6	18	DQ38	87	DQ18	88	DQ50
19	DQ7	20	DQ39	89	DQ19	90	DQ51
21	VSS	22	VSS	91	VSS	92	VSS
23	CAS0#/ DQMBO	24	CAS4#/ DQMB4	93	DQ20	94	DQ52
25	CAS1#/ DQMB1	26	CAS5#/ DQMB5	95	DQ21	96	DQ53
27	VCC	28	VCC	97	DQ22	98	DQ54
29	AO	30	A3	99	DQ23	100	DQ55
31	A1	32	A4	101	VCC	102	VCC
33	A2	34	A5	103	A6	104	A7
35	VSS	36	VSS	105	A8	106	A11
37	DQ8	38	DQ40	107	VSS	108	VSS
39	DQ9	40	DQ41	109	A9	110	A12
41	DQ10	42	DQ42	111	A10	112	A13
43	DQ11	44	DQ43	113	VCC	114	VCC
45	VCC	46	VCC	115	CAS2#/ DQMB1	116	CAS6#/ DQMB6
47	DQ12	48	DQ44	117	CAS3#/ DQMB3	118	CAS7#/ DQMB7
49	DQ13	50	DQ45	119	VSS	120	VSS
51	DQ14	52	DQ46	121	DQ24	122	DQ56
53	DQ15	54	DQ47	123	DQ25	124	DQ57
55	VSS	56	VSS	125	DQ26	126	DQ58
57	CBO	58	CB4	127	DQ27	128	DQ59
59	CB1	60	CBS	129	VCC	130	VCC
				131	DQ28	132	DQ60
				133	DQ29	134	DQ61
				135	DQ30	136	DQ62
61	DU/CLKO	62	DU/CKEO	137	DQ31	138	DQ63
63	VCC	64	VCC	139	VSS	140	VSS
65	DU/RAS*	66	DU/CAS#	141	SDA	142	SCL
67	WE#	68	NC/CKE1	143	VCC	144	VCC
69	RASO#/SO#	70	NC/A12				

<sup>1</sup> DRAM/ SDRAM



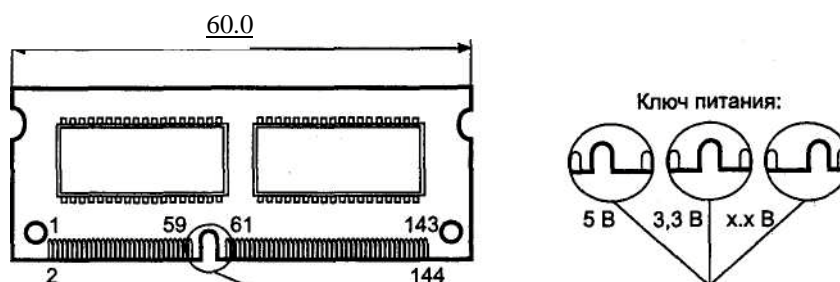


Рис. 7.16. Модули SO DIMM-144 pin

### Модули DRAM cards-88 pin

Модули 88 pin DRAM cards — миниатюрные модули (3,37"х2,13"х0,13" — 85,5х х54х3,3 мм) в пластиковом корпусе размером с карту PCMCIA (PC Card). Имеют 88-контактный разъем (не PCMCIA!), разрядность 18, 32 или 36 бит, емкость 2-36 Мбайт. Комплекуются микросхемами DRAM в корпусах TSOP. Информация о быстродействии и объеме передается по восьми выводам. Внутренняя архитектура близка к SIMM-72. Напряжение питания — 5 или 3,3 В. Применяются в малогабаритных компьютерах, легко устанавливаются и снимаются.

## 7.2. Статическая память

Статическая память — *SRAM* (Static Random Access Memory), как и следует из ее названия, способна хранить информацию в статическом режиме — то есть сколь угодно долго при отсутствии обращений (но при наличии питающего напряжения). Ячейки статической памяти реализуются на триггерах — элементах с двумя устойчивыми состояниями. По сравнению с динамической памятью эти ячейки более сложные и занимают больше места на кристалле, однако они проще в управлении и не требуют регенерации. Быстродействие и энергопотребление статической памяти определяется технологией изготовления и схемотехникой запоминающих ячеек.

*Асинхронная статическая память (Asynchronous SRAM, Async SRAM)*, она же обычная, или стандартная, подразумевается под термином *SRAM* по умолчанию, когда тип памяти не указан.

Микросхемы этого типа имеют простейший асинхронный интерфейс, включающий шину адреса, шину данных и сигналы управления *CS#*, *OE#* и *WE#*. Микросхема выбирается низким уровнем сигнала *CS#* (Chip select), низкий уровень сигнала *OE#* (Output Enable) открывает выходные буферы для считывания данных, *WE#* (Write Enable) низким уровнем разрешает запись. Временные диаграммы циклов обращения приведены на рис. 7.17. При операции записи управление выходными буферами может производиться как сигналом *OE#* (цикл 1), так и сигналом *WE#* (цикл 2). Для удобства объединения микросхем внутренний сигнал *CS#* может собираться по схеме «И» из нескольких внешних, например *CSO#*, *CSt*

и CS2# — в таком случае микросхема будет выбрана при сочетании логических сигналов 0,1,0 на соответствующих входах.

*Время доступа* — задержка появления действительных данных на выходе относительно момента установления адреса — у стандартных микросхем SRAM составляет 12,15 или 20 наносекунд, что позволяет процессору выполнять пакетный цикл чтения 2-1-1-1 (то есть без тактов ожидания) на частоте системной шины до 33 МГц. На более высоких частотах цикл будет не лучше 3-2-2-2.

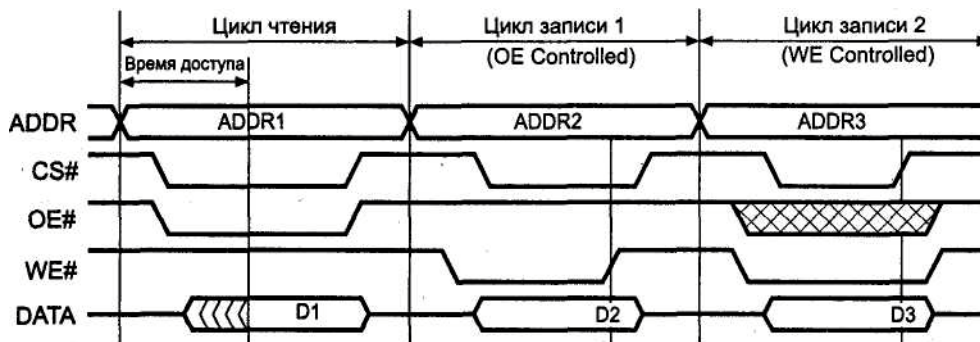


Рис. 7.17. Временные диаграммы чтения и записи асинхронной статической памяти

*Синхронная пакетная статическая память, Sync Burst SRAM*, оптимизирована под выполнение пакетных (burst) операций обмена, свойственных кэш-памяти. В ее структуру введен внутренний двухбитный счетчик адреса (не позволяющий перейти границу четырехэлементного пакетного цикла). В дополнение к сигналам, характерным для асинхронной памяти (адрес, данные, CS#, OE# и WE#), синхронная память использует сигнал CLC (Clock) для синхронизации с системной шиной и сигналы управления пакетным циклом ADSP#, CADS\* и ADV#. Сигналы CADS\* (Cache Address Strobe) и ADSP# (Address Status of Processor), которыми процессор или кэш-контроллер отмечает фазу адреса очередного цикла, являются стробами записи начального адреса цикла во внутренний регистр адреса. Любой из этих сигналов инициирует цикл обращения, одиночный (single) или пакетный (burst), а сигнал ADV# (ADVance) используется для перехода к следующему адресу пакетного цикла. Все сигналы, кроме сигнала управления выходными буферами OE#, синхронизируются по положительному перепаду сигнала CLK. Это означает, что значение входных сигналов должно установиться до перепада и удерживаться после него еще некоторое время. Выходные данные при считывании будут также действительны во время этого перепада. Микросхемы синхронной статической памяти, как и SDRAM, обычно имеют сигнал, выбирающий режим счета адреса: чередование { для процессоров Intel) или последовательный счет (для Power PC).

*Конвейерно-пакетная статическая память, PB SRAM (Pipelined Burst SRAM)*, — усовершенствование синхронной памяти (слово «синхронная» из ее названия для краткости изъяли, но оно обязательно подразумевается). Конвейером является дополнительный внутренний регистр данных, который, требуя дополнительного такта в первой пересылке цикла, позволяет остальные данные получать без тактов

ожидания даже на частотах выше 75 МГц. Задержка данных относительно синхронизирующего перепада у современных микросхем PB SRAM составляет 4,5-8 не! Но, как и в случае Sync Burst SRAM, этот параметр не является временем доступа в чистом виде (не следует забывать о двух-трех тактах в первой передаче), а отражает появление действительных данных относительно очередного перепада сигнала синхронизации. Интерфейс PB SRAM аналогичен интерфейсу Sync Burst SRAM.

### 7.3. Энергонезависимая память

Обобщенное понятие энергонезависимой памяти (*NV Storage*) означает любое устройство, хранящее записанные данные даже при отсутствии питающего напряжения (в отличие от статической и динамической полупроводниковой памяти). В данном разделе рассматриваются только электронные устройства энергонезависимой памяти, хотя к энергонезависимой памяти относятся и устройства с подвижным магнитным или оптическим носителем. Существует множество типов энергонезависимой памяти: ROM, PROM, EPROM, EEPROM, Flash Memory, FRAM, различающихся по своим потребительским свойствам, обусловленным способом построения запоминающих ячеек, и сферам применения. Запись информации в энергонезависимую память, называемая *программированием*, обычно существенно сложнее и требует больших затрат времени и энергии, чем считывание. Программирование ячейки (или блока) — это целая процедура, в которую может входить подача специальных команд записи и верификации. Основным режимом работы такой памяти является считывание данных, а некоторые типы после программирования допускают только считывание, что и обуславливает их общее название ROM (Read Only Memory — память только для чтения) или ПЗУ (постоянное запоминающее устройство).

Запоминающие ячейки энергонезависимой памяти по своей природе обычно асимметричны и, как правило, позволяют записывать только нули в нужные биты предварительно стертых (чистых) ячеек, содержащие единицы. Для некоторых типов памяти чистым считается нулевое состояние ячеек. Однократно программируемые микросхемы позволяют изменять только исходное (после изготовления) состояние ячеек. Для стирания (если оно возможно) требуются значительные затраты энергии (мощности и времени), и процедура стирания обычно существенно дольше записи. Стирание ячеек выполняется либо для всей микросхемы, либо для определенного блока, либо для одной ячейки (байта). Стирание приводит все биты стираемой области в одно состояние (обычно во все единицы, реже — во все нули).

Процедура программирования многих старых типов памяти требует относительно высокого напряжения программирования (12-26 В), а для однократно программируемых (прожигаемых) микросхем и специального (не TTL) интерфейса управления. После программирования требуется верификация — сравнение записанной информации с оригиналом, причем некачественное управление программированием (или брак микросхемы) может приводить к «зарастанию» записанной ячейки, что потребует повторного (возможно, и неудачного) ее программирова-

ния. Возможен и обратный вариант, когда «пробиваются» соседние ячейки, что требует повторного стирания (тоже, возможно, неудачного). Стирание и программирование микросхем может выполняться либо в специальном устройстве — программаторе, либо в самом целевом устройстве, если у него предусмотрены соответствующие средства. Микросхемы различают по способу программирования.

- ◆ Микросхемы, программируемые при изготовлении, — масочные ПЗУ, содержимое которых определяется рисунком технологического шаблона. Такие микросхемы используют лишь при выпуске большой партии устройств с одной и той же прошивкой.
- ◆ Микросхемы, программируемые однократно после изготовления перед установкой в целевое устройство, — ППЗУ (программируемые ПЗУ) или *PROM* (Programmable ROM). Программирование осуществляется прожиганием определенных хранящих элементов на специальных устройствах-программаторах.
- ◆ Микросхемы, стираемые и программируемые многократно, — РПЗУ (репрограммируемые ПЗУ) или EPROM (Erasable PROM — стираемые ПЗУ). Для стирания и программирования требуется специальное оборудование. Микросхемы программируются в программаторе. Иногда возможно программирование микросхем прямо в целевом устройстве, подключая внешний программатор, — так называемый метод ОЯР (Op-Boag! Programming). Наиболее распространены микросхемы УФПЗУ, стираемые ультрафиолетовым облучением, — их обычно называют просто *EPROM* или *UV-EPROM* (Ultra-Violet EPROM). В этом классе имеются и электрически стираемые ПЗУ (ЭСПЗУ) или *EEPROM* (Electrical Erasable PROM).
- ◆ Микросхемы, перепрограммируемые многократно в целевом устройстве, используя программу его процессора, — так называемый метод *ISP* или *ISW* (In-System Programming или In-System Write). К этому классу относятся чисто электрически перепрограммируемые микросхемы *NVRAM* и *FRAM*, но наибольшее распространение получила флэш-память и современные модели *EEPROM*.

*NVRAM* (Non-Volatile Random Access Memory) — энергонезависимая память с произвольным доступом. Это название подразумевает возможность произвольной смены информации не только во всей ее области или блоке, но и в отдельной ячейке, причем не процедурой, а обычным шинным циклом. К этому классу относятся микросхемы *FRAM* и, с некоторой натяжкой, *EEPROM*. У последних время выполнения внутренней операции записи обычно довольно большое, и после интерфейсной операции записи ячейки памяти недоступны ни для каких операций в течение нескольких мс (а то и десятков мс). Флэш-память к этому классу относить нельзя, поскольку изменение информации, недаром называемое программированием, в этой памяти осуществляется специальной программной процедурой.

*Ферроэлектрическая память FRAM* (Ferroelectric RAM) — энергонезависимая память с истинно произвольным доступом, запись и чтение ее осуществляются как в обычных микросхемах статической памяти. При ее изготовлении используется железо — ее можно считать эхом старинной памяти больших машин на магнит-

ных сердечниках. Ячейки FRAM по структуре напоминают DRAM, но информация хранится не в виде заряда конденсатора (который нужно поддерживать регенерацией), а в виде направления поляризации кристаллов. Запись производится непосредственно, предварительного стирания не требуется. Как и флэш-память, она используется в самых портативных системах класса PDA (personal digital assistants — персональный цифровой ассистент). Над этими устройствами активно работает фирма Hitachi совместно с фирмой Ramtron ([www.ramtron.com](http://www.ramtron.com)) и фирма Matsushita совместно с фирмой Symetrix. В настоящее время выпускаются микросхемы емкостью 4-256 Кбит (технология 0,35 мкм) с параллельным интерфейсом (как SRAM) и временем доступа 70-120 нс, а также с последовательным интерфейсом PC. Кроме массивов памяти FRAM используется и в специальных энергонезависимых регистрах — есть, например, микросхемы FM573 и FM574, которые при включенном питании ведут себя аналогично стандартным 8-битным регистрам '573 и '574, но при выключении питания помнят свое состояние. Микросхемы FRAM имеют интерфейс КМОП, питание 5 В, но имеются изделия и на 2,7 В. В отличие от флэш-памяти, у которой число циклов перезаписи принципиально ограничено (хотя и очень велико), ячейки FRAM практически не деградируют в процессе записи — гарантируется до  $10^{10}$  циклов перезаписи. Провозглашается замена на FRAM даже динамической памяти, однако в PC память FRAM автору пока встречать не доводилось.

### 7.3.1. Постоянная и полупостоянная память — ROM, PROM, EPROM

*Масочные постоянные запоминающие устройства* — ПЗУ или ROM — имеют самое высокое быстродействие (время доступа 30-70 нс). Эти микросхемы в PC широкого применения не получили ввиду сложности модификации содержимого (только путем изготовления новых микросхем); они иногда применялись в качестве знакогенераторов в некоторых моделях графических адаптеров CGA, MDA, HGC.

*Однократно программируемые постоянные запоминающие устройства* — ППЗУ или PROM — имеют аналогичные параметры и благодаря возможности программирования изготовителем оборудования (а не микросхем) находят более широкое применение для хранения кодов BIOS и в графических адаптерах. Программирование этих микросхем осуществляется только с помощью специальных программаторов, в целевых устройствах они устанавливаются в «кроватки» или запаиваются. Как и масочные, эти микросхемы практически нечувствительны к электромагнитным полям (в том числе и к рентгеновскому облучению), и несанкционированное изменение их содержимого в устройстве исключено (конечно, не считая отказа).

*Репрограммируемые постоянные запоминающие устройства* — РПЗУ TМNEPROM -т до недавних пор были самыми распространенными носителями BIOS как на системных платах, так и в адаптерах, а также использовались в качестве знакогенераторов. Наиболее популярные микросхемы имеют восьмибитную организацию

и обозначение вида 27xx-tt или 27Сxx-tt для микросхем CMOS. Здесь xx определяет емкость в килобитах: 2708 — 1 К x 8 — родоначальник семейства, 2716/32/64/128/ 256/512 имеют емкость 2/4/8/16/32/64 Кбайт соответственно, 27010 и 27020 -128 и 256 Кбайт. Время доступа tt лежит в диапазоне 50—250 нс. Шестнадцатитбитные микросхемы (например, 27001 или 27002 емкостью 64 К или 128 К 16-битных слов) в РС применяются редко.

Микросхемы EPROM тоже программируются на программаторах, но относительно простой интерфейс записи позволяет их программировать и в устройстве (но не в штатном его режиме работы, а при подключении внешнего программатора). *Стирание* микросхем осуществляется ультрафиолетовым облучением в течение нескольких минут. Специально для стирания микросхемы имеют стеклянные окошки. После программирования эти окошки заклеивают, предотвращая стирание под действием солнечного или люминесцентного облучения. Время стирания зависит от расстояния до источника облучения, его мощности и объема микросхемы (более емкие микросхемы стираются быстрее). Вместо штатных стирающих устройств можно пользоваться и обычной медицинской ультрафиолетовой лампой с расстояния порядка 10 см. Для микросхем 2764 ориентировочное время стирания составляет 5 минут. Стирание переводит все биты в единичное состояние. «Недостертые» микросхемы при программировании могут давать ошибки, передержка при стирании снижает количество возможных циклов перепрограммирования (в пределе — до нуля).

Некоторые микросхемы, похожие по виду и обозначению на стираемые ультрафиолетом, не имеют окна (они упакованы в дешевый пластмассовый корпус). Эти микросхемы либо стираются рентгеновским облучением (что не совсем удобно), либо допускают лишь однократно программирование, которое может выполняться и по заказу фирмой-производителем микросхем. Их интерфейс полностью совпадает с интерфейсом обычных микросхем EPROM 27xx.

С программированием ПЗУ приходится сталкиваться при русификации графических адаптеров (CGA, MDA, HGC) и принтеров с незагружаемыми знакогенераторами, а также при замене (или восстановлении) системной микросхемы BIOS или микросхемы Boot ROM — микросхемы удаленной загрузки для адаптера локальной сети. Распространенные *программаторы EPROM* имеют интерфейс подключения к COM- или LPT-порту РС или подключаются через собственную карту расширения (обычно с шиной ISA). Время программирования зависит от типа и объема микросхемы и применяемого алгоритма программирования. Классический алгоритм с 50-миллисекундными импульсами записи каждой ячейки для современных микросхем практически не используется. Более быстрые «интеллектуальные» алгоритмы позволяют записывать 8 килобайт (2764) менее чем за минуту. Вся процедура программирования может затягиваться при использовании медленного интерфейса связи программатора с РС (например, COM-порт на скорости 2400 бод) за счет длительной процедуры копирования данных в буфер программатора.

Интерфейс микросхем постоянной памяти в режиме *чтения* совпадает с интерфейсом статической памяти. Для *программирования* (записи) требуется приложение ко входу  $V_{pp}$  напряжения программирования, которое для различных типов EPROM лежит в диапазоне 12-26 В (обычно указывается на корпусе микросхемы). Комбинации управляющих сигналов, формирующие импульсы записи для EPROM разной емкости, различны. При напряжении на входе  $V_{pp}$  5 В и ниже модификация памяти (запись) невозможна ни при каких комбинациях управляющих сигналов, и микросхемы работают строго в режиме ROM. Этот режим и используется для микросхем BIOS, так что никакой вирус им не страшен.

В РС чаще всего применяют микросхемы EPROM в корпусах DIP и PLCC (табл. 7.21), расположение выводов популярных микросхем приведено на рис. 7.18 и 7.19.

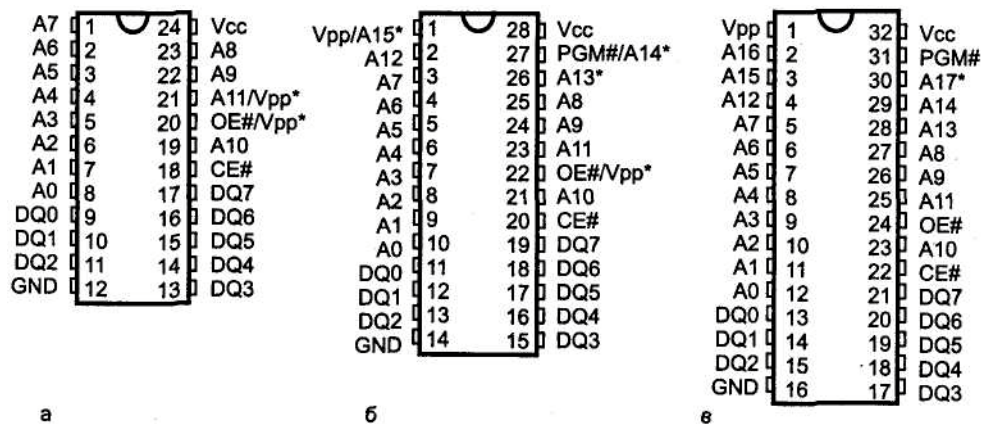


Рис. 7.18. Расположение выводов микросхем EPROM в корпусах DIP: а — DIP-24, б — DIP-28, в — DIP-32

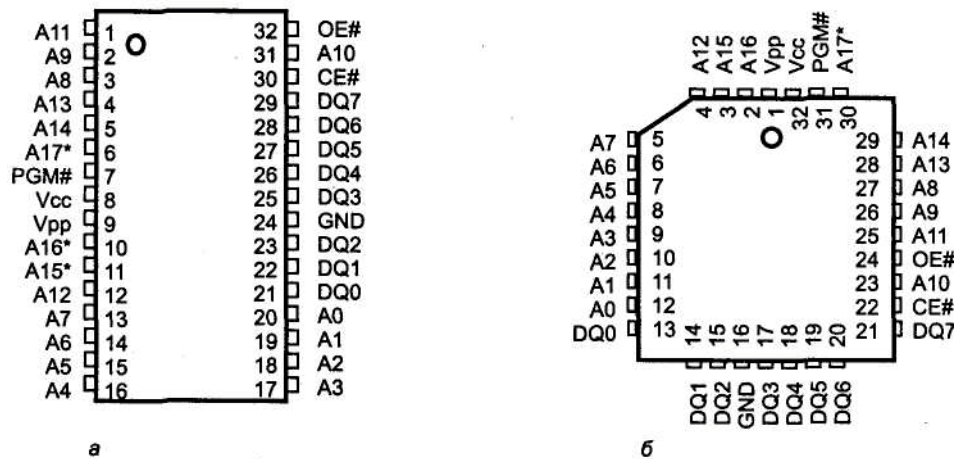


Рис. 7.19. Расположение выводов микросхем EPROM в корпусах TSOP и PLCC: а — TSOP-32, б — PLCC-32

Таблица 7.21. Популярные микросхемы EPROM

Микросхема и организация	Корпус	Рисунок	Примечание
2716 — 2Кх8	DIP-24	7.18,a	20 = OE#; 21 = V <sub>pp</sub>
2732 — 4Кх8	DIP-24	7.18,a	20 = OE#/V <sub>pp</sub> , 21=A11
2764 — 8Кх8	DIP-28	7.18,6	1 = V <sub>pp</sub> , 22 = OE#; 26 = NC, 27 = PGM#
27128 — 16Кх8	DIP-28	7.18,6	1 = V <sub>pp</sub> , 22 = OE#; 26 = A13, 27 = PGM#
27256 — 32 Кх 8	DIP-28	7.18,6	1 = V <sub>pp</sub> , 22 = OE#; 26 = A13, 27 = A14
27512 — 64Кх8	DIP-28	7.18,6	1 = A15, 22 = OE#/V <sub>pp</sub> , 26 = A13, 27 = A14
27010 — 128 Кх 8	DIP-32	7.18,8	30 = NC
27010 — 128 Кх 8	TSOP-32	7.19,a	6 = NC
27010 — 128 Кх8	PLCC-32	7.19,6	30 = NC
27020 — 256 Кх 8	DIP-32	7.18,b	-
27020 — 256 Кх 8	TSOP-32	7.19,a	-
27020 — 256 Кх 8	PLCC-32	7.19,6	-

Назначение выводов микросхем EPROM приведено в табл. 7.22.

Таблица 7.22. Назначение выводов микросхем EPROM

Сигнал	Назначение
CE#	Chip Enable — разрешение доступа. Низкий уровень разрешает обращение к микросхеме, высокий уровень переводит микросхему в режим пониженного потребления
OE#	Output Enable — разрешение выходных буферов. Низкий уровень при низком уровне CE# разрешает чтение данных из микросхемы. У некоторых типов микросхем на этот же вывод в режиме программирования подается напряжение V <sub>pp</sub>
DQx	Data Input/Output — двунаправленные линии шины данных. Время доступа при чтении отсчитывается от установки действительного адреса или сигнала CE# (в зависимости от того, что происходит позднее)
Ax	Address — входные линии шины адреса. Линия A9 допускает подачу высокого (12В) напряжения для чтения кода производителя (AO = 0) и устройства (AO = 1), при этом на остальные адресные линии подается логический ноль
PGM#	Programm — импульс программирования (некоторые микросхемы не имеют этого сигнала, их программирование осуществляется по сигналу CE# при высоком уровне V <sub>pp</sub> )
	V <sub>pp</sub> Программирующее напряжение питания (для некоторых типов — импульс) V <sub>cc</sub> Питание (+5 В)

Отметим *основные свойства EPROM*.

- ◆ Стирание информации происходит сразу для всей микросхемы под воздействием облучения и занимает несколько минут. Стертые ячейки имеют единичные значения всех бит.
- ◆ Запись может производиться в любую часть микросхемы побайтно, в пределах байта можно маскировать запись отдельных бит, устанавливая им единичные значения данных.



- ♦ Защита от записи осуществляется подачей низкого (5 В) напряжения на вход  $V_{pp}$  в рабочем режиме (только чтение).
- ♦ Защита от стирания производится заклеивкой окна.

### 7.3.2. EEPROM и флэш-память

Электрически стираемая (и перезаписываемая) память *EEPROM*, или *E<sup>2</sup>PROM* (Electrical Erasable PROM), отличается простотой выполнения записи. В простейшем (для пользователя) случае программирование сводится к записи байта по требуемому адресу, после чего некоторое время микросхема не способна выполнять операции чтения/записи и по другим адресам, вплоть до окончания выполнения внутренней операции программирования (со встроенным стиранием). Микросхемы могут поддерживать и режим страничной записи (Page Write), в котором они принимают поток байт записи смежных ячеек в страничный буфер на нормальной скорости интерфейса, после чего вся страница записывается в энергонезависимую память. Страничная запись экономит время (запись страницы выполняется за то же время, что и одной ячейки), но размер страничного буфера, как правило, небольшой (4-32 байт для микросхем небольшого объема и до 128-256 байт — большого). Более сложный интерфейс записи использует систему команд, в которую могут входить команды разрешения/запрета стирания и записи, стирание (отдельной ячейки или всей памяти), запись. Микросхема может иметь и специальные внутренние регистры, например регистр состояния, определяющий готовность микросхемы к обмену данными и возможные режимы защиты от модификации ячеек. Некоторые старые микросхемы для стирания требуют подачи сравнительно высокого (12 В) напряжения на определенные выводы. По процедуре программирования некоторые микросхемы EEPROM схожи с флэш-памятью. В настоящее время EEPROM применяются наряду с флэш-памятью, причем они могут соседствовать даже в одной микросхеме (например, микроконтроллере). Это объясняется очень большим гарантированным числом циклов перезаписи ( $10^6$  и более) EEPROM, но меньшим достижимым объемом. Также EEPROM обычно имеет и большее гарантированное время сохранности информации (до 100 лет). Флэш-память при большем объеме и более производительных способах записи и стирания допускает меньшее число циклов перезаписи, и время сохранения информации у нее меньше (может быть и всего 10 лет). Микросхемы EEPROM выпускаются с различными интерфейсами, последовательными (Serial EEPROM) с интерфейсами PC, SPI и иными и параллельными (Parallel EEPROM) с интерфейсами статической памяти (и EPROM).

*Флэш-память* по определению относится к классу EEPROM (электрическое стирание), но использует особую технологию построения запоминающих ячеек. Стирание во флэш-памяти производится сразу для целой области ячеек (блоками или полностью всей микросхемы). Это позволило существенно повысить производительность в режиме записи (программирования). Флэш-память обладает сочетанием высокой плотности упаковки (ее ячейки на 30 % меньше ячеек DRAM), энергонезависимого хранения, электрического стирания и записи, низкого потребле-

ния, высокой надежности и невысокой стоимости. Первые микросхемы флэш-памяти были предложены фирмой Intel в 1988 году и с тех пор претерпели существенные изменения по архитектуре, интерфейсу и напряжению питания. Каждая ячейка флэш-памяти состоит всего из одного униполярного (полевого) транзистора. Ячейки организованы в матрицу; разрядность данных внешнего интерфейса — 8 или 16 бит (ряд микросхем имеет переключаемую разрядность). Чистые (стертые) ячейки содержат единицу во всех битах; при записи (программировании) нужные биты обнуляются. Возможно последующее программирование и уже записанных ячеек, но при этом можно только обнулять единичные биты, но не наоборот. В единичное состояние ячейки переводятся только при стирании. Стирание выполняется для всей матрицы ячеек; стирание одиночной ячейки невозможно. Чтение флэш-памяти ничем не отличается от чтения любой другой памяти — подается адрес ячейки, и через некоторое время доступа (десятки-сотни нс) на выходе появляются данные. Запись выглядит несколько сложнее — для программирования каждого байта (слова) приходится выполнять процедуру, состоящую из операций записи и считывания, адресованных к микросхеме флэш-памяти. Однако при этом шинные циклы обращения к микросхеме являются нормальными для процессора, а не растянутыми, как для EPROM и EEPROM. Таким образом, в устройстве с флэш-памятью легко реализуется возможность перепрограммирования без извлечения микросхем из устройства. Большинство микросхем флэш-памяти имеют интерфейс, аналогичный асинхронной статической памяти (SRAM), а при чтении он упрощается до интерфейса ROM/PROM/EPROM. Существуют версии с интерфейсом динамической памяти, асинхронным и синхронным, а также и со специальными интерфейсами, в том числе и  $1^2C$ . Первые микросхемы работали только при напряжении питания 5 В, а для программирования и стирания требовали дополнительное питание  $V_{pp} = +12$  В. Затем появились микросхемы всего с одним напряжением питания +5 В. Дальнейшее развитие технологии позволило снизить напряжение питания до 2,7-3,3 В и 1,65-2,2 В, а  $V_{pp}$  — до 5, 3,3, 2,7 и даже 1,65 В. В производстве микросхем используются технологические процессы с разрешением 0,3, 0,22, 0,18 мкм (чем мельче ячейки, тем они экономичнее). Микросхемы первых выпусков (1990 г.) имели гарантированное число циклов стирания-программирования 10 000, современные — 100 000.

Флэш-память имеет время доступа при чтении 35-200 нс. Стирание информации (поблочное или во всей микросхеме) у микросхем середины 90-х годов занимает 1-2 секунды, программирование (запись) байта — порядка 10 мкс. У современных микросхем время стирания и записи заметно сократилось. Процедура записи от поколения к поколению упрощается (см. ниже). От ошибочного стирания (записи) применяются различные методы программной и аппаратной защиты. Программной защитой является ключевая последовательность команд, нарушение которой не позволяет начать операции стирания и записи. Аппаратная защита не позволяет выполнять стирание и запись, если на определенные входы не поданы требуемые уровни напряжения. Аппаратная защита может защищать как весь массив целиком, так и отдельные блоки.

По организации массива в плане стирания групп ячеек различают следующие архитектуры:

- ◆ *Bulk Erase* (BE) — все ячейки памяти образуют единый массив; запись возможна в произвольную ячейку; стирание возможно только для всего объема сразу;
- ◆ *Boot Block* (BB) — массив разделен на несколько блоков разного размера, стираемых независимо, причем один из блоков имеет дополнительные средства защиты от стирания и записи;
- ◆ *Flash File* — массив разделен на несколько равноправных независимо стираемых блоков обычно одинакового размера, что позволяет их называть микросхемами с симметричной архитектурой (Symmetrical Architecture, SA).

Организация BE применялась только в микросхемах первого поколения, ее недостатки вполне очевидны (получается просто аналог EEPROM с более удобным способом стирания и интерфейсом программирования). Все современные микросхемы секторизованы (разбиты на отдельно стираемые блоки), так что остается лишь деление на симметричную и несимметричную архитектуру.

В симметричной архитектуре (SA), как правило, используется разбиение на блоки по 64 Кбайт; один из крайних блоков (с самым большим или самым маленьким адресом) может иметь дополнительные средства защиты.

В асимметричной архитектуре один из 64-килобайтных блоков разбивается на 8 блоков по 8 Кбайт. Один из блоков имеет дополнительные аппаратные средства защиты от модификации и предназначается для хранения жизненно важных данных, не изменяемых при запланированных модификациях остальных областей. Эти микросхемы специально предназначены для хранения системного программного обеспечения (BIOS), а привилегированный блок (*Boot Block*) хранит минимальный загрузчик, позволяющий загрузить (например, с дискеты) и выполнить утилиту программирования основного блока флэш-памяти. В обозначении этих микросхем присутствует суффикс *T* (Top) или *B* (Bottom), определяющий положение *Boot*-блока либо в старших, либо в младших адресах соответственно. Первые предназначены для процессоров, стартующих со старших адресов (в том числе, x86, Pentium), вторые — для стартующих с нулевого адреса, хотя возможны и противоположные варианты, когда некоторые биты шины адреса перед подачей на микросхему памяти инвертируются. Старые микросхемы В В малого объема имели немного другое распределение, например микросхема *28F001BX-T* (*28F001BN-T*), часто применяемая для флэш-BIOS в PC, содержит:

- ◆ основной блок (Main Block) объемом 112 Кбайт (00000h-1BFFFh);
- ◆ два блока параметров (Parameter Block) объемом по 4 Кбайт (1C000h-1CFFFh и 1D000h-1DFFFh);
- ◆ загрузочный блок (Boot Block) объемом 8 Кбайт (1E000h-1FFFFh), стирание и программирование которого возможны лишь при особых условиях.

Основной блок и блоки параметров по защите равноправны; выделение небольших блоков параметров позволяет в них хранить часто сменяемую информацию, например ESCD технологии PnP.

Выпускают и комбинированные микросхемы, например MT28C3214P2FL представляет собой комбинацию флэш-памяти 2 М x 16 и SRAM 256 К x 16. По организации матрицы ячеек различают архитектуры NOR и NAND. В традиционной организации NOR транзисторы на одном проводе объединяются своими стоками параллельно, как бы образуя логический элемент ИЛИ-НЕ (NOR — Not OR). Эта организация обеспечивает высокое быстродействие произвольного считывания, что позволяет исполнять программы прямо из флэш-памяти (не копируя в ОЗУ) без потери производительности. В организации NAND несколько транзисторов разных ячеек соединяются последовательно, образуя логический элемент И-НЕ (NAND — Not AND), что дает высокую скорость последовательных обращений.

В первых микросхемах флэш-памяти каждая ячейка (всего один транзистор) предназначалась для хранения одного бита информации (1 — стерта, 0 — «прошита»). Позже появилась технология хранения двух битов в одной ячейке — благодаря совершенствованию технологии удалось надежно различать 4 состояния ячейки, что и требуется для хранения двух битов. Два бита в ячейке хранит память Intel StrataFlash, емкость одной такой микросхемы уже достигла 128 Мбит (16 Мбайт).

Флэш-память постоянно развивается как в плане повышения емкости и снижения потребления, так и в плане расширения возможностей и повышения производительности. Так, например, в ряде микросхем AMD имеется возможность чтения одновременно с записью других блоков (чтение во время стирания стало возможным еще со второго поколения флэш-памяти).

Некоторые микросхемы обеспечивают быстрый обмен в *страничном режиме (Page Mode)*. Страницей являются 4 или 8 смежных ячеек; первое чтение в странице выполняется со временем доступа 70 нс. Если микросхема остается выбранной, то другие ячейки этой страницы (отличающиеся значением младших битов адреса) можно считывать циклами длительностью по 20 нс. Доступ к одиночным ячейкам не отличается от обычного. Микросхемы с *пакетным режимом (Burst Mode)* вдобавок к одиночному и страничному режимам (асинхронным) могут работать и в синхронном режиме. Для этого они имеют вход синхронизации CLK. Адрес начала пакета передается вместе с сигналом ADV# (фиксируются по положительному перепаду CLK). Первые данные на выходе появятся через 3 такта, после чего в каждом следующем такте будут выдаваться очередные данные.

*Синхронная флэш-память* имеет интерфейс (и даже упаковку в корпуса), совпадающий с SDRAM. В настоящее время выпускаются микросхемы с частотой 66 МГц (например, MT28S4M16LC — 1 М x 16 x 4 банка), ожидаются микросхемы и на 133 МГц. Такая память удобна для встраиваемых компьютеров для хранения ПО, исполняемого прямо на месте (без копирования в ОЗУ).

Микросхемы флэш-памяти с симметричной архитектурой выпускаются и с *интерфейсом DRAM* (динамической памяти) — с мультиплексированной шиной памяти, стробируемой сигналами RAS# и CAS#. Они предназначены для применения в модулях SIMM или DIMM, устанавливаемых в гнезда для обычной динамической памяти. Таким образом реализуются, например, модули PostScript для лазерных принтеров и любые резидентные программные модули. Эти модули, естественно, не будут определяться системой как основная память — на попытку обычной записи и считывания, предпринимаемую в тесте POST при определении установленной памяти, они ответят весьма своеобразно. Также они не будут восприниматься и как модули дополнительной системы BIOS, поскольку займут неподходящие для этого физические адреса. Использовать эти модули смогут только с помощью специального драйвера, который «объяснит» чипсету, какому диапазону адресов пространства памяти соответствуют сигналы выборки банков флэш-памяти. Поскольку интерфейс модулей SIMM и DIMM не предполагает сигналов защиты записи, системного сброса и дополнительного питания +12 В, все вопросы, связанные с программированием и защитой, решаются дополнительными элементами, устанавливаемыми на модулях. При использовании 16-битных микросхем такие модули непосредственно не обеспечивают независимую побайтную запись, но она может обеспечиваться программно, маскированием (записью OFFh) немодифицируемых байт.

Для хранения BIOS появились микросхемы *флэш-памяти с интерфейсом LPC*, называемые хабами (firmware hub).

Для некоторых сфер применения требуются специальные меры по блокированию изменения информации пользователем. Так, Intel в некоторые микросхемы вводит однократно записываемые регистры OTP (One-Time-Programmable). Один 64-битный регистр содержит уникальный заводской номер, другой может программироваться пользователем (изготовителем устройства) только однажды.

Фирма Intel выпускает микросхемы «Wireless Flash Memory» — за интригующим названием скрывается, конечно же, «нормальный» электрический интерфейс с проводами (wireless — без проводов). Однако они ориентированы на применение в средствах беспроводной связи (сотовые телефоны с доступом к Интернету): питание 1,85 В, наличие регистров OTP для защиты от мошенничества и т. п.

### **Корпуса, интерфейс и обозначение микросхем флэш-памяти**

Микросхемы флэш-памяти упаковывают в корпуса со стандартизованным назначением выводов. Первые микросхемы выпускались в корпусах DIP, что обеспечивало легкость замены микросхем (E)EPROM на флэш-память. Далее в целях миниатюризации перешли к корпусам PLCC, TSOP и TSOP-II. Применение корпусов FBGA (Fine Pitch Ball Grid Array) — матрицы 6x8 шариковых выводов с шагом 0,8 мм — позволяет уменьшить размер корпуса до минимума, требуемого для упаковки кристалла. Для микросхем, используемых в картах SmartMedia, применяют и оригинальную упаковку KGD (Known Good Die).

На рис. 7.20-7.22 приведено расположение выводов распространенных микросхем флэш-памяти (основной вариант цоколевки). Многие микросхемы имеют два варианта цоколевки для корпусов поверхностного монтажа — основной и зеркальный (реверсный). Это позволяет существенно упростить разводку печатных проводников (серпантином) при объединении большого количества микросхем в массивы флэш-памяти.

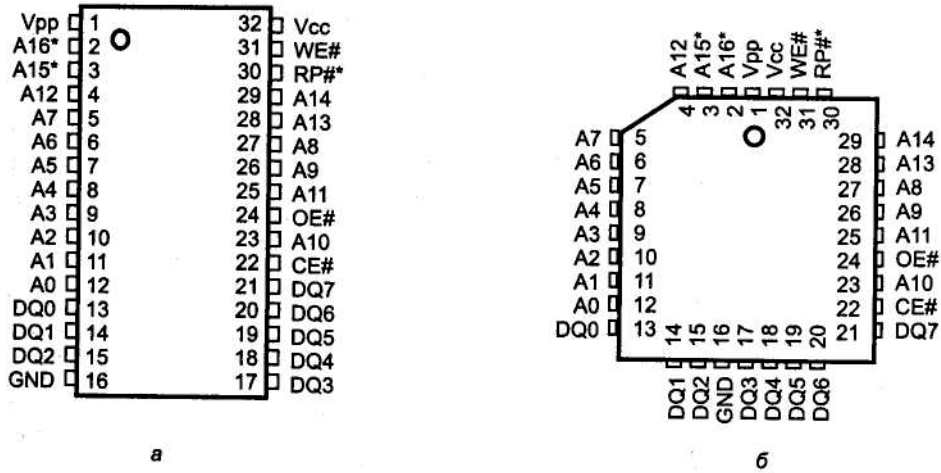


Рис. 7.20. Расположение выводов микросхем флэш-памяти с 8-битной организацией в корпусах DIP и PLCC: а — DIP-32, б — PLCC-32

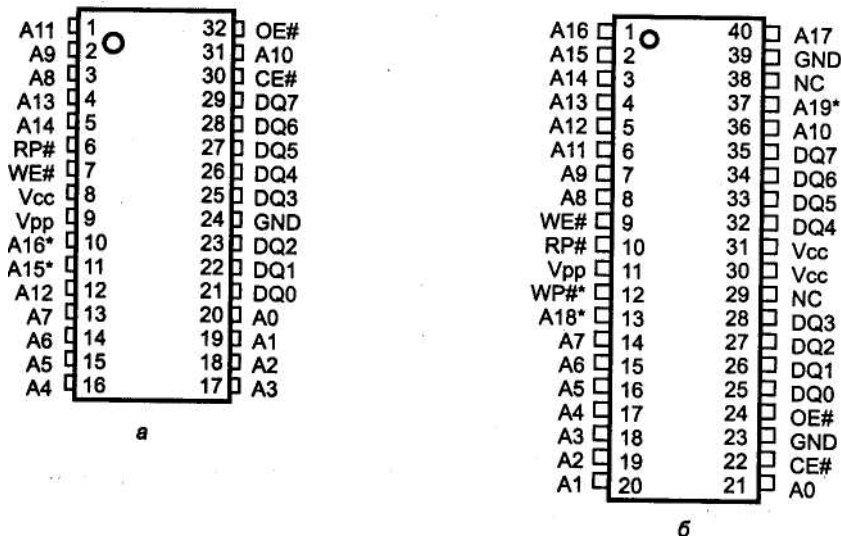


Рис. 7.21. Расположение выводов микросхем флэш-памяти с 8-битной организацией в корпусах TSOP: а — TSOP-32, б — TSOP-40

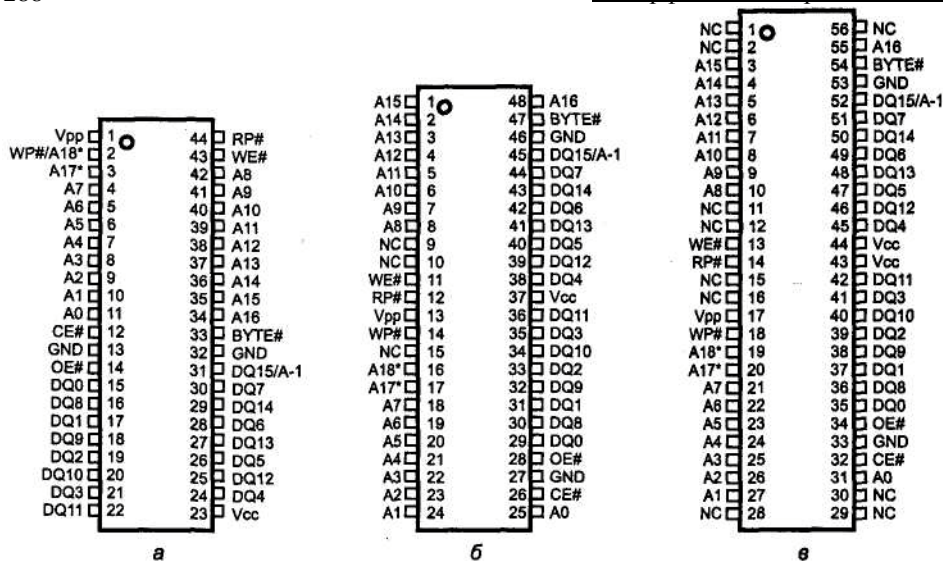


Рис. 7.22. Расположение выводов микросхем флэш-памяти с 8/16-битной организацией в корпусах TSOP-44: а — TSOP-44, б — TSOP-48, в — TSOP-56

Назначение сигналов микросхем флэш-памяти приведено в табл. 7.23; микросхемы разных изготовителей и моделей могут иметь не все из приведенных управляющих сигналов.

Таблица 7.23. Назначение сигналов микросхем флэш-памяти

Сигнал	Назначение
CE#	Chip Enable — разрешение доступа. Низкий уровень разрешает обращение к микросхеме, высокий уровень переводит микросхему в режим пониженного потребления. Доступ к микросхеме, имеющей два входа (CE1 # и CE2#), возможен при низком уровне на обоих входах
OE#	Output Enable — разрешение выходных буферов. Низкий уровень при низком уровне сигнала CE# разрешает чтение данных из микросхемы. Подача высокого (12В) напряжения во время подачи команды стирания или программирования позволяет модифицировать и Boot-блок (этот метод используется редко, поскольку требует не-ТТЛ сигнала)
WE#	Write Enable — разрешение записи. Низкий уровень при низком уровне сигнала CE# разрешает запись и переводит выходные буферы в высокоимпедансное состояние независимо от сигнала OE#. Временные диаграммы шинного цикла записи аналогичны обычной статической памяти, что позволяет подключать флэш-память непосредственно к системной шине процессора. Допустимы оба способа управления — как с помощью сигнала WE# на фоне низкого уровня CE#, так и наоборот. Минимальная длительность импульса записи совпадает со временем доступа
DQx	Data Input/Output — двунаправленные линии шины данных. Время доступа при чтении отсчитывается от установки действительного адреса или сигнала CE# (в зависимости от того, что происходит позднее). Фиксация данных при записи происходит по положительному перепаду WE# или CE# в зависимости от того, что происходит раньше

Сигнал	Назначение
BYTE*	Управляющий сигнал для выбора режима обращения к микросхемам с 8/16-битной организацией. Они имеют два 8-битных банка, и их ячейки памяти адресуются 16-битными словами. Низкий уровень сигнала BYTE* задает восьмибитный режим обмена по линиям DQ[0:7], при этом линия DQ15/A-1 становится самой младшей линией адреса, переключающей банки, а линии DQ[8:14] переходят в высокоимпедансное состояние
Ax	Address — входные линии шины адреса. Линия A9 допускает подачу высокого (12В) напряжения (как и EPROM) для чтения кода производителя (A0=0) и устройства (A0=1), при этом на остальные адресные линии подается логический «0»
RP# (PWD#)	Reset/Power Down, раньше обозначался PWD# (PowerDown). Низкий уровень сбрасывает регистр команд и переводит микросхемы в режим «глубокого сна» (Deep Powerdown) с минимальным (доли микроампера) потреблением питания. Перевод сигнала в высокий логический уровень «пробуждает» микросхему (примерно за 0,3-0,8 мкс), после чего она переходит в режим чтения данных. Подача высокого (12В) напряжения разрешает программирование даже защищенного Boot-блока
WP#	Write Protect — защита записи. При низком уровне WP* модификация Boot-блока или других блоков с установленным битом защиты возможна только при наличии высокого (12В) напряжения на входе RP*. При высоком уровне защита блоков игнорируется
RY/BY#	Ready/Busy* — сигнал готовности (высокий уровень) микросхемы к очередной операции программирования или стирания. Низкий уровень указывает на занятость управляющего автомата (WSM) выполнением операции стирания или программирования. Выход обычно не управляется сигналами OE# и CE*. В микросхемах 28F016SA и выше выход имеет тип «открытый коллектор», запираемый по команде, и программируемое назначение
3/5*	Сигнал, переключающий режим работы микросхемы в зависимости от питающего напряжения, введен для оптимизации быстродействия или потребления

Интерфейс микросхем флэш-памяти хорошо сочетается со стандартными сигналами, используемыми в микропроцессорных системах. *Внутренние циклы* стирания, записи и верификации выполняются автономно от *шинных циклов* внешнего интерфейса, что является существенным преимуществом перед микросхемами EPROM и EEPROM. В режиме чтения они полностью совместимы с EPROM, совпадая с ними и по расположению основных выводов.

*Обозначение микросхем* и для изделий лидеров в области разработки и производства флэш-памяти — фирм Intel и AMD — несколько отличаются. Остальные производители для своих изделий, по свойствам аналогичных, в основном придерживаются системы обозначений лидеров.

*Обозначение микросхем флэш-памяти Intel* начинается с признака 28F, за которым следует трехзначный код объема (табл. 7.24), а за ними — два символа технологии и архитектуры:

- ◆ B5, BC, BX, BR - Boot Block с питанием 5 В;
- ◆ C3 — Boot Block с питанием 3 В;
- ◆ F3 — Boot Block с питанием 3 В, повышенное быстродействие;
- ◆ J3 и J5 — StrataFlash (SA) с питанием 3 и 5 В соответственно;
- ◆ S3 и S5 — Flash File (SA) с питанием 3 и 5 В соответственно.



Для флэш-памяти AMD первая часть обозначения определяет тип и характеристики микросхем:

- ◆ Am29BDS — 1,8 В, считывание одновременно с записью, пакетный режим чтения;
- ◆ Am29DS — 1,8 В, считывание одновременно с записью;
- ◆ Am29SL- 1,8 В;
- ◆ Am29LV-3В;
- ◆ Am29DL — 3 В, считывание одновременно с записью;
- ◆ Am29BL — 3 В, пакетный режим чтения;
- ◆ Am29PL — 3 В, страничный режим чтения;
- ◆ AmSOLV - 3 В, UltraNAND;
- ◆ Am29F-5В.

Далее следует трехзначный код объема, за ним символ технологии изготовления (В, С или D), за которым следует символ архитектуры:

- ◆ Т — boot sector, верхний;
- ◆ В — boot sector, нижний;
- ◆ Н — симметричная, защищен со старшим адресом;
- ◆ L — симметричная, защищен с младшим адресом;
- ◆ U (нет символа) — симметричная;
- ◆ J40 — число 100 %-годных блоков (только для UltraNAND).

Оставшаяся часть определяет параметры питания, быстродействие, тип корпуса, температурный диапазон и некоторые особенности.

Таблица 7.24. Популярные микросхемы флэш-памяти

Обозначение	Организация <sup>1</sup>
256	32Кх8BE
512	64Кх8BE
010	128Кх8BE
020	256 Кх 8 BE
001	128Кх8BB
002	256Кх8BB
004	512Кх8BB, SA
008	1Мх8BB, SA
016	2 М х 8 BB, SA
200	256Кх8/128Кх16BB
400	512Кх8/256Кх16BB
800	1024Кх8/512Кх16BB
160	2Мх8/1 Мх16SA, BB
320	4Мх8/2Мх16SA
640	8Мх8/4Мх16SA

<sup>1</sup> BE — Bulk Erase (стираемые целиком), BB — Boot Block (несимметричные блоки), SA — Symmetric Architecture (симметричные блоки). Через косую черту указана организация для микросхем с переключаемой разрядностью данных.

Флэш-память с интерфейсом PCMCIA (PC Card) оптимизирована для построения внешней памяти миниатюрных РС. Модуль флэш-памяти в формате PC Card имеет интерфейс дисков IDE (ATA) как на уровне электрических сигналов, так и по системе команд. Кроме собственно микросхем накопителя этот модуль обычно содержит управляющую микросхему программируемой логики. Флэш-память в стандарте PC Card логически является устройством *внешней* памяти. Ее не следует путать с похожей по виду памятью в формате Credit Card, которая является оперативной и вставляется в специальный (не PCMCIA) слот компьютера. Внешнюю память, в отличие от оперативной, в принципе можно вставлять и вынимать без перезагрузки ОС.

### Организация и программирование флэш-памяти Intel

По организации и программированию можно выделить три поколения флэш-памяти Intel.

Микросхемы *первого поколения* (28F256, 28F512, 28F010, 28F020) представляют собой единый массив памяти, стираемый целиком (*bulk erase*). Для выполнения стирания и записи микросхемы имеют внутренний регистр команд и управляющий автомат WSM (Write State Machine). Стирание и программирование флэш-памяти возможны только при подаче на вход  $V_{PP}$  напряжения 12 В по командам, записываемым во внутренний регистр в шинном цикле записи по сигналу WE#.

Выполнение команд инициируется записью кодов команд во внутренний регистр, для чего процессор должен выполнить команду записи в память по адресу, принадлежащему области программируемой микросхемы флэш-памяти. На микросхему при этом должны прийти сигналы CE# (выбор) и WE# (запись). Последующие обращения к этой области как по записи (W), так и по чтению (R) должны соответствовать исполняемой команде (табл. 7.25). В шинном цикле записи адрес (если он требуется для данной команды) фиксируется по спаду сигнала WE#, фиксация данных выполнения команды происходит по фронту WE#. Большинство команд подается безадресно (по любому адресу, принадлежащему данной микросхеме); команда верификации стирания и второй цикл команды программирования подаются по адресу требуемой ячейки. Результаты стирания и программирования считываются по адресу конкретной интересующей ячейки.

Ниже описано назначение команд.

- ◆ Read Memory — команда чтения данных, переводящая микросхему в режим чтения, совместимый по интерфейсу с EPROM.
- ◆ Read ID — команда чтения идентификаторов. В последующих шинных циклах чтения по адресу 0 считывается M\_Id (Manufacturer Identifier — идентификатор производителя, 89h), по адресу 1 — D\_Id (Device Identifier — идентификатор устройства, для микросхем 8F256, 28F512, 28F010, 28F020 это B9h, B8h, B4h и BDh соответственно). Из этого режима микросхема выходит по записи любой другой действительной команды. Идентификаторы можно читать и путем подачи высокого напряжения на A9 (как и для EPROM).

Таблица 7.25. Команды микросхем флэш-памяти Intel первого поколения

Команда	Число циклов шины	Первый цикл шины <sup>1</sup>			Второй(третий)цикл шины <sup>1</sup>		
		R/W	Адрес	Данные	R/W	Адрес	Данные
Read Memory	1	W	X	ООп	-	-	-
Read ID	3	W	X	90h	R	0(1)	M <sub>id</sub> (D <sub>id</sub> )
Set-up Erase/Erase	2	W	X	20h	W	X	20h
Erase Verify	2	W	EA	АОп	R	X	EVD
Set-up Program/Program	2	W	X	40h	W	PA	PD
Program Verify	2	W	X	COh	R	X	PVD
Reset	2	W	X	FFh	W	X	FFh

<sup>1</sup> Здесь X обозначает несущественный адрес, M<sub>Id</sub> и D<sub>Id</sub> — идентификаторы производителя и устройства, EA — адрес ячейки, в которой контролируется стирание, EVD — данные, считанные при верификации стирания (должны быть FFh), PA и PD — адрес и данные программируемой ячейки, PVD — данные, считанные при верификации программирования.

- ◆ Set-up Erase/Erase — подготовка и собственно стирание. Внутренний цикл стирания начинается по подъему сигнала WE# во втором шинном цикле и завершается по последующему шинному циклу записи или по внутреннему таймеру (Stop Timer). Последующей командой обычно является верификация стирания. Два шинных цикла записи, необходимые для выполнения команды, снижают вероятность случайного стирания и позволяют отказаться от выполнения стирания посылкой команды Reset. Наличие внутреннего таймера позволяет не заботиться о точной выдержке времени для стирания, необходимо только выдержать минимальный интервал (около 10 мс). Перед стиранием все биты микросхемы должны быть предварительно запрограммированы в нули.
- ◆ Erase Verify — верификация стирания. Отличается от обычного считывания тем, что проверяемая ячейка ставится в более жесткие условия считывания для повышения достоверности контроля стирания. Между шинными циклами команды верификации должна быть пауза не менее 6 мкс. *Алгоритм быстрого стирания* (Quick-Erase) предусматривает предварительное обнуление всех ячеек (командами программирования) и выполнение команды стирания, сопровождаемой верификацией. Команды верификации последовательно выполняются для каждой ячейки микросхемы. Если результат считывания отличается от FFh, производится повторное стирание (длительностью 10 мс), и последующая верификация может начинаться с первой ранее не стертой ячейки. Если количество повторов стирания превышает 3 000, фиксируется ошибка стирания и микросхема признается негодной. Алгоритм позволяет выполнить полное стирание микросхемы менее чем за секунду.
- ◆ Set-up Program/Program — подготовка и собственно программирование. Команда выполняется аналогично стиранию, но во втором шинном цикле передается адрес и данные программируемой ячейки, а последующая выдержка должна составлять не менее 10 мкс.

- ◆ Program Verify— верификация программирования (аналогично верификации стирания), обычно следующая после команды программирования. Между шинными циклами команды верификации должна быть пауза не менее 6 икс. *Алгоритм быстрого программирования* (Quick-Pulse Programming) предусматривает формирование внутреннего цикла программирования длительностью 10 мкс с последующей верификацией. В случае несовпадения результата выполняется повторное программирование (до 25 раз для каждой ячейки), а если и это не помогает — фиксируется отказ микросхемы.
- ◆ Reset — команда сброса, прерывающая команду программирования или стирания. Эта команда не меняет содержимое памяти; после нее требуется подача другой действительной команды.

По включении питания внутренний регистр команд обнуляется, что соответствует команде чтения, и микросхема работает как обычная микросхема PROM или EPROM. Это позволяет устанавливать микросхемы флэш-памяти вместо EPROM аналогичной емкости. При подаче на вход  $V_{PP}$  низкого напряжения (0-6,5 В) стирание и программирование невозможны, и микросхема ведет себя как обычная EPROM.

Микросхемы *второго поколения* секторизованы — ячейки группируются в блоки, допускающие независимое стирание (асимметричное разбиение — *Boot Block* и симметричное — *Flash File*). Длительная операция стирания одного блока может прерываться для считывания данных других блоков, что значительно повышает гибкость и производительность устройства. Микросхемы имеют более сложный внутренний управляющий автомат и в них введен регистр состояния, что позволяет разгрузить внешний процессор и программу от забот по отслеживанию длительности операций программирования и стирания, а также упростить эти процедуры.

В отличие от микросхем первого поколения, в шинном цикле записи адрес и данные фиксируются по положительному перепаду  $WE\#$ . Низкий уровень дополнительного управляющего сигнала  $RP\#$  (в первых версиях обозначался как  $PWD\#$ ) предназначен для перевода микросхемы в режим с минимальным потреблением. В этом режиме модификация содержимого памяти невозможна. Соединение этого вывода в нормальном режиме (когда не требуется перезапись *Boot*-блока) с системным сигналом  $RESET^*$  предохраняет микросхему от выполнения ложных команд, которые могут появиться в процессе подачи питания. Внутренние операции стирания и программирования выполняются после отправки соответствующих кодов во внутренний *регистр команд*. Команды приведены в табл. 7.26. Как и в первом поколении, этот регистр для большинства команд безадресный, но команды программирования и стирания посылаются по требуемому адресу ячейки (блока). Обработка операций внутренним управляющим автоматом отображается соответствующими битами *регистра состояния* SR (Status Register), по значению которых внешняя программа может получить информацию о результате выполнения и возможности отправки следующих команд. Чтение

регистра SR выполняется по специальной команде; есть и команда его очистки. Назначение бит регистра состояния описано ниже.

◆ SR. 7 — WSMS (Write State Machine Status) — состояние управляющего автомата:

- 0 — Busy (занят операцией стирания или программирования);
- 1 — Ready (свободен).

◆ SR.6—ESS (Erase Suspend Status) — состояние операции стирания:

- 0 — стирание завершено или выполняется;
- 1 — стирание приостановлено.

◆ SR. 5 — ES (Erase Status) — результат стирания блока:

- 0 — блок стерт успешно;
- 1 — ошибка стирания.

◆ SR. 4 — PS (Program Status) — результат программирования байта:

- 0 — байт записан успешно;
- 1 — ошибка записи.

◆ SR.3 — VPPS ( $V_{PP}$  Status) — состояние  $V_{PP}$  во время программирования или стирания:

- 0 — напряжение было в норме;
- 1 — зафиксировано понижение напряжения, и операция прервана.

◆ S R [ 2 :0 ] — зарезервированы.

Таблица 7.26. Команды микросхем флэш-памяти Intel второго поколения

Команда	Число циклов шины	Первый цикл шины <sup>1</sup>			Второй (третий) цикл шины <sup>1</sup>		
		R/W	Адрес	Данные	R/W	Адрес	Данные
Read Array/Reset	1	W	X	FFh	-	-	-
Read ID (DJd)	3	W	X	90h	R	0(1)	MJd
Read Status Register	2	W	X	70h	R	X	SRD
Clear Status Register	1	W	X	50h	-	-	-
Erase Setup/Erase Confirm	2	W	BA	20h	W	BA	DOh
Erase Suspend/Erase Resume	2	W	X	BOh	W	X	DOh
Program Setup/Program	2	W	PA	40h	w	PA	PD
Alternate Program Setup/Program <sup>2</sup>	2	W	PA	10h	w	PA	PD

<sup>1</sup> Здесь X обозначает несущественный адрес, M\_Id и D\_Id — идентификаторы производителя и устройства, SRD — данные, считанные из регистра состояния, PA и PD — адрес и данные программируемой ячейки, BA — адрес блока.

<sup>2</sup> Альтернативный код команды программирования; доступен для микросхем емкостью 2,4 и 8 Мбит.

Ниже описано назначение команд.

- ◆ Read Array/Reset — чтение массива памяти (перевод в режим, совместимый с EPROM) и прерывание операций стирания и программирования.
- ◆ Read ID — чтение идентификаторов производителя и устройства.
- ◆ Read Status Register — чтение регистра состояния.
- ◆ Clear Status Register — сброс регистра состояния.
- ◆ Erase Setup/Erase Conf i rm — подготовка и стирание блока. В отличие от микросхем первого поколения, все внутренние операции, необходимые для стирания (обнуление ячеек блока, стирание и верификация), выполняются автоматически. При получении команды в регистре состояния устанавливается признак занятости ( $SR. 7=0$ ) и любая шинная операция чтения микросхемы будет передавать данные этого регистра. Внешняя программа, периодически опрашивая регистр состояния, дожидается окончания выполнения стирания (когда  $SR. 7=1$ ). Результат стирания определяется по значению бит 3, 4, 5 (их нулевое значение соответствует успешному выполнению операции).
- ◆ Erase Suspend/Erase Resume — приостановка/продолжение стирания. Операцию стирания блока (как самую длительную) можно приостановить для чтения данных из других блоков. После выполнения команды Erase Suspend (код B0h) необходимо дождаться признака приостановки стирания ( $SR. 6=1$ ), после чего, подав команду Read Array, можно считывать данные другого блока. По окончании считывания подается команда Erase Resume (код D0h), которая продолжает процесс стирания и снова переводит микросхему в режим чтения регистра состояния.
- ◆ Program Setup/Program — подготовка и программирование ячейки. Эта команда выполняется аналогично подготовке и выполнению стирания, но не может быть приостановлена. Команда выполняет сразу и программирование, и верификацию.

Команды стирания блока и программирования можно подавать, только когда управляющий автомат свободен (бит  $SR. 7=1$ ). Во время этих операций микросхема следит за уровнем напряжения  $V_{PP}$ , и, если оно понижается до порога  $V_{P_{PLK}}$ , этот факт регистрируется в регистре состояния и операция прерывается. Также операция прерывается при понижении напряжения питания  $V_{CC}$  до 2,5 В.

При считывании *регистра состояния* его мгновенное значение фиксируется по спаду сигнала  $CE\#$  или  $OE\#$  (самого позднего из них в шинном цикле считывания).

Программирование и стирание Boot-блока отличаются от операций с другими блоками тем, что для них требуется подача высокого потенциала  $V_{HH}$  (не TTL, а +12 В) на вход  $PWD\#$  перед выдачей команды стирания или программирования и удержание его до успешного завершения операции. Альтернативный способ — подача такого же потенциала, но на вход  $OE\#$  на время пар шинных циклов записи команд стирания или программирования. Попытка программирования Boot-блока без выполнения этих условий не удастся, а в регистре состояния одновременно

устанавливаются единичные значения бит ES и PS, что индицирует попытку модификации защищенного блока.

Микросхемы *Flash-file* организованы в виде набора одинаковых блоков, равноправных (симметричных) по защите (Symmetrical Architecture, SA). Защита от модификации для 28F008SA может осуществляться только для всей микросхемы подачей низкого напряжения на вход  $V_{pp}$ . По интерфейсу и командам микросхемы совпадают с микросхемами Boot Block (исключая специфику Boot-блока).

Архитектура микросхем 28F016SA существенно изменена, что значительно повышает производительность программирования (до 28,6 Мбайт/с в пакетном режиме) и обеспечивает поблочную защиту. Микросхема имеет два буфера данных для записи по 256 Кбайт. Флэш-память организована как 32 блока по 64 Кбайт, допускающих однобайтное или двухбайтное обращение. С каждым блоком связан собственный 8-битный *регистр состояния блока* BSRx (Block Status Register). Адреса регистров смещены относительно начального адреса блока на 2 или 1 для режимов обращения x8 или x16 соответственно. Назначение бит BSR описано ниже.

- ◆ BSR. 7 — BS (Block Status) — состояние блока:
  - 1 — готов;
  - 0 — занят.
- ◆ BSR. 6 — BLS (Block-Lock Status) — состояние защиты блока:
  - 1 — программирование и стирание запрещены;
  - 0 — блок не защищен.
- ◆ BSR.5 — BOS (Block Operation Status) — состояние операции с блоком:
  - 1 — операция завершена безуспешно;
  - 0 — операция успешно завершена или выполняется.
- ◆ BSR. 4 — BOAS (Block Operation Abort Status) — состояние отмены операции с блоком:
  - 1 — операция отменена;
  - 0 — операция не отменялась.
- ◆ BSR. 3 — QS (Queue Status) — состояние очереди:
  - 1 — очередь заполнена;
  - 0 — очередь доступна.
- ◆ BSR. 2 -VPPS( $V_{pp}$  Status)- состояние  $V_{pp}$ :
  - 1 — обнаружен низкий уровень, операция прервана;
  - 0 —  $V_{pp}$  в норме.
- ◆ BSR[1:0] — зарезервированы.

Глобальный регистр состояния GSR (Global Status Register) несет информацию о состоянии микросхемы в целом. К GSR можно обращаться по адресу, смещенному относительно начального адреса любого блока на 4 или 2 для режимов обращения x8 или x16 соответственно. Назначение бит GSR описано ниже.

- ◆ GSR.7—W5MS (Write State Machine Status) — состояние автомата записи (и завершенности внутренних операций):
  - 1 — занят;
  - 0 — свободен.
- ◆ GSR.6 — 055 (Operation Suspend Status) — состояние приостановки операции:
  - 1 — операция приостановлена;
  - 0 — операция выполняется или завершена.
- ◆ G5R. 5 — DOS (Device Operation Status) — состояние операции (копирует бит регистра состояния текущего блока):
  - 1 — операция завершена неудачно;
  - 0 — операция успешно завершена или выполняется.
- ◆ G5R. 4 — DSS (Device Sleep Status) — состояние ожидания:
  - 1 — ожидание (Sleep);
  - 0 — нормальный режим.
- ◆ GSR.3 — QS (Queue Status) — состояние очереди:
  - 1 — очередь заполнена;
  - 0 — очередь доступна.
- ◆ GSR. 2 — PBAS (Page Buffer Available Status) — состояние буферов записи:
  - 1 — есть свободный буфер;
  - 0 — нет свободного буфера.
- ◆ GSR. 1 — PBS (Page Buffer Status) — состояние выбранного буфера записи:
  - 1 — выбранный буфер свободен;
  - 0 — буфер занят операцией с WSM.
- ◆ GSR.0— PBSS (Page Buffer Select Status) — номер выбранного буфера:
  - 1 — выбран буфер 1;
  - 0 — выбран буфер 0.

Для сохранения программной совместимости имеется безадресный регистр CSR (Compatible Status Register), полностью совпадающий с регистром состояния 28F008SA и микросхем с архитектурой Boot Block. Все команды этих микросхем доступны. Введены новые команды (табл. 7.27), обеспечивающие расширение функций. Ниже перечислены дополнительные возможности микросхем.



- ◆ Буферированное страничное программирование. Кроме обычного побайтного или двухбайтного программирования возможно быстрое заполнение буфера шинными циклами записи. Далее переписывание его содержимого (всего или фрагмента) во флэш-память выполняется одной командой. Содержимое буфера может быть считано после подачи соответствующей команды.
- ◆ Двухбайтное программирование при 8-битном использовании.
- ◆ Поддержка очереди команд позволяет при наличии свободного места в очереди подавать последующие команды стирания или программирования, не дожидаясь освобождения автомата *WSM*. Признак готовности *WSM* установится только после выполнения всех команд очереди.
- ◆ Автоматическая запись из буфера во флэш-массив во время стирания другого блока. Программная защита позволяет для любого блока установить бит защиты в специальную энергонезависимую область. Запись и стирание защищенного блока может осуществляться только после снятия общей защиты записи по сигналу *WP#*. Сброс бита защиты блока осуществляется только при его успешном стирании или перезаписи.
- ◆ Стирание всех незащищенных блоков может выполняться одной командой.
- ◆ Программирование использования сигнала *RY/BY\**. Возможно разрешение отображения бита готовности глобального регистра состояния, подачи импульсного сигнала по завершении программирования или стирания (на выбор), а также запрет его формирования.
- ◆ Перевод микросхемы в режим ожидания (*Sleep*) с пониженным потреблением. В этом режиме возможно считывание состояния и получение команд.

Таблица 7.27. Дополнительные команды микросхем 28F016SA

Команда	Первый цикл шины			Второй цикл шины			Третий цикл шины		
	R/W	Адрес <sup>1</sup>	Данные <sup>2</sup>	R/W	Адрес <sup>1</sup>	Данные <sup>2</sup>	R/W	Адрес <sup>1</sup>	Данные <sup>2</sup>
Read Extended Status Register (чтение дополнительных регистров)	W	X	xx71h	R	RA	GSRD BSRD	-	-	-
Page Buffer Swap (смена буфера)	W	X	xx72h	-	-	-	-	-	-
Read Page Buffer (чтение буфера)	W	X	xx75h	R	PBA	PBD	-	-	-
Single Load to Page Buffer (одиночная запись в буфер)	W	X	xx74h	W	PBA	PBD	-	-	-

### 7.3. Энергонезависимая память

Команда	Первый цикл шины			Второй цикл шины			Третий цикл шины		
	R/W	Адрес <sup>1</sup>	Данные <sup>2</sup>	R/W	Адрес <sup>1</sup>	Данные <sup>2</sup>	R/W	Адрес <sup>1</sup>	Данные <sup>2</sup>
Sequential Load to Page Buffer x8/x16 (последовательная запись в буфер)	W	X	xxE0h	W	X	BCL/WCL <sup>3</sup>	W	X	BCH/WCH <sup>3</sup>
Page Buffer Write to Flash x8/x16 (перепись буфера в массив)	W	X	xx0Ch	W	A0	BC(LH)/WCL <sup>3</sup>	W	X	BC(HL)/WCH <sup>3</sup>
Two-Byte Write x8 (двухбайтное программирование в режиме x8)	W	X	xxFBh	W	A0	WD(L,H) <sup>3</sup>	W	WA	WD(H,L) <sup>3</sup>
Lock Block/Confirm (защита блока)	W	X	xx77h	W	BA	xxD0h	-	-	-
Upload Status Bits/Confirm (выгрузка состояния блока в GSR)	W	X	xx97h	W	X	xxD0h	-	-	-
Upload Device Information (выгрузка информации об устройстве)	W	X	xx99h	W	X	xxD0h	-	-	-
Erase All Unlocked Blocks/Confirm (стирание всех незащищенных блоков)	W	X	xxA7h	W	X	xxD0h	-	-	-
RY/BY# Enable to Level-Mode (разрешение отображения GSR.7 потенциалом)	W	X	xx96h	W	X	xx01h	-	-	-
RY/BY# Pulse-On-Write (импульс по окончании программирования)	W	X	xx96h	W	X	xx02h	-	-	-

продолжение

Таблица 7.27 (продолжение)

Команда	Первый цикл шины			Второй цикл шины			Третий цикл шины		
	R/W	Адрес <sup>1</sup>	Данные <sup>2</sup>	R/W	Адрес <sup>1</sup>	Данные <sup>2</sup>	R/W	Адрес <sup>1</sup>	Данные <sup>2</sup>
RY/BY# Pulse- Op-Erase (импульс по окончании стирания)	W	X	xx96h	W	X	xx03h	-	-	-
RY/BY# Disable (запрет сигнала RY/BY#)	W	X	xx96h	W	X	xx04h	-	-	-
Sleep (режим ожидания)	W	X	xxF0h	-	-	-	-	-	-
Abort (отмена операции)	W	X	xx80h	-	-	-	-	-	-

<sup>1</sup> BA - Block Address — адрес блок, PBA - Page Buffer Address — адрес внутри буфера, RA - Extended Register Address — адрес дополнительного регистра (BSRx или GSR), WA - Write Address — адрес во флэш-массиве. AO указывает на порядок следования байт в режиме x8 (при низком уровне BYTE\*):  
O - сначала младший, затем старший; 1 - наоборот.

<sup>2</sup> AD - Array Data — данные из массива, PBD - Page Buffer Data — данные буфера, WD (L,H) - Write Data (Low, High) — данные для записи в массив, BSRD - BSR Data — информация регистра состояния блока, GSRD - GSR Data — информация глобального регистра состояния.

<sup>3</sup> WC (L,H) - Word Count (Low, High) — счетчик слов. WCL-0 соответствует записи одного слова. Для буфера 256 байт WCH-0. BC (L,H) - Byte Count (Low, High) - счетчик байт. WCL-0 соответствует записи одного байта. Для буфера 256 байт WCH-0.

Микросхема 28F032SA представляет собой два параллельно соединенных кристалла 28F016SA в одном корпусе. Входы CE# одного из них соединены с выводами CEO# и CE1 #, второго — с CEO# и CE2#.

*Третье поколение* — современные микросхемы, выполненные по технологии *SmartVoltage*, допускают стирание и программирование при напряжении  $V_{pp}$  как 12 В, так и 5 В. В последнем случае эти операции занимают больше времени. Кроме того, операции чтения возможны при пониженном (3,3 и даже 2,7 В) напряжении питания  $V_{cc}$ , при этом снижается потребление, но увеличивается время доступа. Для управления защитой данных введен логический сигнал WP# (Write Protect). При его высоком уровне программирование и стирание защищенных блоков выполняются так же, как и остальных. При низком уровне WP# модификация защищенных блоков возможна только при наличии высокого (12 В) напряжения на входе RP#. Для полной защиты от стирания и программирования на вход  $V_{pp}$  должен подаваться низкий логический уровень (или 0 В), а не 5 В, как у микросхем с программированием напряжением 12 В.

Настройка (оптимизация потребления и быстродействия) происходит по уровню напряжения на выводе  $V_{cc}$  по включении питания, переход на другое значение должен производиться через выключение питания.

## Флэш-память фирмы AMD

Фирмой AMD выпускается несколько семейств микросхем флэш-памяти. Первые из них были близки по характеристикам к флэш-памяти Intel первого поколения

(Bulk Erase, стирание и программирование 12 В): это *Am28F256/512/010/020*. В отличие от аналогичных микросхем Intel, *Am28F256/512* не имели стоп-тайме-ра, что требовало точной выдержки при программировании и стирании. Следующим этапом были микросхемы *Am28F256A/512A/010A/020A* со встроенным алгоритмом программирования, отличающимся от алгоритма микросхем Intel второго поколения как последовательностью команд, так и способом определения момента окончания операций. Для защиты от случайного выполнения команды состоят из 3-6 шинных циклов, причем для них существенен и адрес (табл. 7.28). Состояние выполнения операций стирания или программирования определяется по результату данных, полученных в шинном цикле чтения по адресу ячейки, участвующей в операции (а не регистра состояния, как у Intel). Для определения окончания операций может использоваться метод *Data# Polling* или *Toggle Bit*. Метод *Data# Polling* основан на анализе бита D7 считанных данных. В начале выполнения внутреннего цикла он устанавливается инверсным по отношению к тому, что должно быть записано в ячейку. По успешном окончании операции он принимает желаемое значение (при стирании — 1). Метод *Toggle Bit* основан на анализе бита Об, который при каждом шинном цикле считывания во время выполнения операции меняет свое значение на противоположное. По окончании операции он остановится в каком-либо состоянии, при этом об успешности можно судить по биту 7. Единичное значение бита D5 — *Exceeded Timing Limits* — указывает на превышение допустимого времени выполнения операции.

Микросхемы семейства *Am29Fxxx* выполняют все операции при одном питающем напряжении 5 В и имеют секторизованную структуру (Sector Erase), симметричную (аналогично Flash File) или несимметричную (Boot Block), с верхним (Т) и нижним (В) положением Boot-блока. С помощью программатора каждый сектор может быть защищен от модификации в целевой системе (в отличие от Intel способ установки и снятия защиты фирмой AMD широко не раскрывается). По расположению выводов и интерфейсу микросхемы соответствуют стандарту JEDEC для флэш-памяти с одним питающим напряжением. Микросхемы позволяют выполнять одновременное стирание группы секторов. Все эти микросхемы, кроме *Am29F010*, имеют возможность приостановки стирания сектора (Erase Suspend) для выполнения чтения других секторов, а *Am29F080/016* позволяют еще и программировать байты во время приостановки стирания.

Таблица 7.28. Команды флэш-памяти *Am29F010*

Команда	Reset/Read	Autoselect	Byte Program	Chip Erase	Sector Erase
Количество циклов	3	3	4	6	6
1-й цикл Addr	5555h	5555h	5555h	5555h	5555h
Data	AAh	AAh	AAh	AAh	AAh
2-й цикл Addr	2AAAh	2AAAh	2AAAh	2AAAh	2AAAh
Data	55h	55h	55h	55h	55h

продолжение &

Таблица 7.28 (продолжение)

	Команда	Reset/Read	Autoselect	Byte Program		
3-й цикл	Addr	5555h	5555h	5555h	-	-
	Data	FOh	90h	AOh	-	-
4-й цикл	Addr	-	XX00h/XX0	PA <sup>1</sup>	-	-
	Data	-	01h/20h	PD <sup>2</sup>	-	-
5-й цикл	Addr	-	-	-	-	-
	Data	-	-	-	-	-
6-й цикл	Addr	-	-	-	-	-
	Data	-	-	-	-	-

<sup>1</sup> PA " адрес программируемой ячейки.

<sup>2</sup> PD - данные для записи в программируемую ячейку.

<sup>3</sup> SA - адрес стираемого сектора (значимы биты A16, A15 и A14).

В командах значение бит A15, A16 существенно только при задании адреса ячейки. Ниже описано назначение команд.

- ◆ **Reset/Read** — сброс и перевод в режим считывания массива. Производится автоматически по включении питания и при получении некорректной команды (или адреса) в цепочке.
- ◆ **Autoselect** — чтение кодов идентификации производителя (A=0), устройства (A=1) или состояния защиты сектора (биты A16-A14 задают адрес сектора, A2-1, остальные биты адреса — нулевые). Результат считывания состояния защищенного сектора — 01h, незащищенного — 00h. Идентификаторы и состояние защиты могут быть считаны и путем подачи высокого напряжения на вход A9 в шинном цикле считывания.
- ◆ **Byte Program** — программирование байта. После четвертого цикла шины начинается внутреннее выполнение программирования, при этом чтение по адресу программируемой ячейки выводит биты состояния.
- ◆ **Chip Erase** — стирание всех незащищенных секторов. На время выполнения стирания чтение по любому адресу (кроме адресов, принадлежащих защищенным секторам) выводит биты состояния.
- ◆ **Sector Erase** — стирание сектора или группы секторов. Стирание начинается через 80 нкс после окончания последнего шинного цикла цепочки. До этого момента можно посылать цепочки команд стирания других секторов, выполнение начнется через 80 мкс после окончания последней цепочки. Если среди указанных секторов имеется защищенный, его стирание не выполняется. На время выполнения стирания чтение по адресу любого из стираемых секторов (кроме защищенных) выводит биты состояния. Бит DQ3 — Sector Erase Timer — указывает на начало выполнения стирания сектора (очередную последовательность команд стирания сектора можно начинать, пока бит 3=0).

Следующим этапом стала секторированная флэш-память *Am29LVxxx* с одним питающим напряжением (3,0 В) для всех операций. У этих микросхем защита

любого сектора также устанавливается с помощью программатора стандартной микросхемы EPROM.а также возможно временное снятие защиты в целевой системе. Кроме программной индикации окончания операции (биты 5-7, считанные по адресу ячейки), имеется и аппаратная (сигнал RY/BY\*). Также имеется сигнал аппаратного сброса, переводящий в режим чтения.

Вышеперечисленные микросхемы имеют традиционную архитектуру NOR. От них значительно отличается микросхема Am30LV0064D — 64 Мбит (8 Мх8) с архитектурой UltraNAND, обеспечивающей быстрый последовательный доступ к данным выбранной страницы. Каждая страница имеет 512 байт данных и 16 дополнительных байт, используемых, например, для хранения ECC-кода. Для выбора страницы при чтении (загрузки во внутренний 528-байтный регистр) требуется около 7 мкс, после чего данные считываются последовательно со скоростью до 20 Мбайт/с (50 не/байт). Таким образом, среднее время на чтение одного байта составляет всего 65 не. Для записи данные (страница полностью или частично) загружаются в регистр с той же скоростью, после чего запись их в массив хранящих ячеек требует всего 200 мкс. Таким образом, среднее время на запись одного байта составляет всего 430 не — в 20 раз быстрее обычной (NOR) флэш-памяти (скорость записи 2,3 Мбайт/с). Стирание выполняется блоками по 8 Кбайт за 2 мс (в обычной — 600 мс). Микросхем питается от 3 В. Планируется достижение объема микросхемы до 1 Гбит. Надежность хранения — 10 лет,  $10^4$  циклов безошибочного программирования, более  $10^6$  циклов программирования с коррекцией ошибок. Применение — «твердые диски», цифровые камеры, диктофоны и т. п.

### Флэш-память других фирм

Микросхемы флэш-памяти выпускаются многими фирмами. Они различаются по организации, интерфейсу, напряжению питания и программирования, методам защиты и другим параметрам. Лидеры в области разработки и производства флэш-памяти — фирмы AMD, Fujitsu Corporation, Intel Corporation и Sharp Corporation летом 1996 года приняли спецификацию *CFI* (Common Flash Memory Interface Specification), обеспечивающую совместимость разрабатываемого программного обеспечения с существующими и разрабатываемыми моделями флэш-памяти. Эта спецификация определяет механизм получения информации о производителе, организации, параметрах и возможных алгоритмах программирования микросхем флэш-памяти. Для этого микросхемы флэш-памяти должны поддерживать команду опроса QUERY. В первом шинном цикле в микросхему по адресу 55h посылается код 98h (микросхема может и игнорировать адрес, «отзываясь» только на код данных 98h). Во втором шинном цикле, адресованном к этой микросхеме, выполняется чтение данных идентификационной структуры (на шину адреса микросхемы подается адрес интересующего байта структуры). Этими парами циклов структура может быть считана в произвольном порядке. Идентификационная структура содержит:

- ◆ ключ-признак наличия структуры — цепочка символов «QRY»;
- ◆ идентификатор (2 байта) первичного набора команд и интерфейса программирования;

- ◆ указатель на таблицу параметров для программирования (и саму таблицу);
- ◆ идентификатор, указатель и таблицу параметров альтернативного набора команд и интерфейса (если имеется);
- ◆ минимальные и максимальные значения напряжений питания (основного и программирующего);
- ◆ значение тайм-аутов для операций стирания (блока и всей микросхемы) и записи (байта, слова, буфера);
- ◆ объем памяти;
- ◆ максимальное число байтов для многобайтной записи;
- ◆ описания независимо стираемых блоков.

Для перевода в режим чтения массива микросхемы должны воспринимать команду чтения массива Read Array — запись кода FFh (FOh) по любому адресу в микросхеме.

Для большинства изделий справедливы тенденции, описанные при рассмотрении микросхем Intel и AMD, а именно — повышение объема, снижение напряжений питания и потребляемой мощности, повышение производительности и упрощение внешнего интерфейса для операций стирания и программирования. По интерфейсу программирования микросхемы, у которых в начальной части обозначения стоит число «28», как правило, близки к флэш-памяти Intel, а с числом «29» — к флэш-памяти AMD.

Микросхемы с буферизованным программированием или страничной записью (Fast Page Write) могут не иметь в своей системе команд отдельной операции стирания сектора. Внутренняя операция стирания (и предварительного обнуления сектора) выполняется при страничном программировании.

Для защиты от случайного выполнения ключевые последовательности команд содержат от 2 до 6 шинных циклов, причем у них может быть важен и адрес (как в микросхемах AMD). Методы защиты секторов имеют различную как программную, так и аппаратную реализацию. Для временного снятия защиты используют различные способы, одним из которых является ключевая последовательность семи шинных циклов чтения.

*Микросхемы флэш-памяти Micron* совместимы с Intel и обозначаются аналогично, но начинаются с признака MT28F. Среди них есть и особенные, например: MT28F321P2FG - 2 М x 16 Page Flash Memory, MT28F322D18FH - 2 М x 16 Burst Flash Memory.

Фирма *Silicon Storage Technology* выпускает разнообразные микросхемы флэш-памяти с одним напряжением питания для всех операций. Их свойства можно определить по обозначению вида SST xx YY zzz — ttt, где xx — семейство:

- ◆ 28 — побайтное программирование, посекторное стирание;
- ◆ 29 — страничное программирование с прозрачным стиранием (команда стирания сектора отсутствует, внутренняя операция выполняется автоматически перед записью страницы в массив).

Элемент YY задает функциональный тип и напряжение питания:

- ◆ *EE*— EEPROM-совместимые, выполнение одной инструкции,  $V_{cc} = 5 \text{ В}$ ;
- ◆ *LE* - то же, что и *EE*,  $V_{cc} \sim 3 \text{ В}$ ;
- ◆ *VE* - то же, что и *EE*,  $V_{cc} = 2,7 \text{ В}$ ;
- ◆ *SF*— операции Super Flash Command Register,  $V_{cc} = 5 \text{ В}$ ;
- ◆ *IF*-то же, что и *SF*,  $V_{cc} = 3 \text{ В}$ ;
- ◆ *W<sup>7</sup>*-то же, что и *SF*,  $V_{cc} = 2,7 \text{ В}$ ;
- ◆ *DM* — Disk Media (для флэш-дисков, требует внешнего контроллера),  $V_{cc} = 5 \text{ В}$ ;
- ◆ *LM* — то же, что и *DM*,  $V_{cc} \sim 3 \text{ В}$ ;
- ◆ *Ш*-тоже, что и *DM*,  $V_{cc} = 2,7 \text{ В}$ ;
- ◆ *PC* — PCMCIA (интерфейс и протоколы),  $V_{cc} = 5 \text{ В}$ .

Элемент zzz задает объем микросхемы:

- ◆ 572 - 512 Кбит (64 К x 8);
- ◆ 070 - 1 Мбит (128 К x 8);
- ◆ 040 - 4 Мбит (512 К x 8);
- ◆ 050 - 8 Мбит (1 М x 8);
- ◆ 016 - 16 Мбит (2 М x 8);
- ◆ 032 - 32 Мбит (4 М x 8).

Элемент ttt задает время доступа при чтении.

Микросхемы *SST29EEQ10*, *29LE010* и *29VE010*, часто применяемые в качестве носителя флэш-BIOS, организованы как 1024 страницы по 128 байт с программной и аппаратной защитой. Каждая страница может быть защищена независимо от других. Временные диаграммы стирания и программирования, а также необходимое напряжение программирования генерируются внутри микросхемы. Окончание операции определяется по алгоритму *Toggle Bit* или *Data# Polling*.

Аналогичные параметры имеют микросхемы *29EE011*, *29LE011*, *29VE011* фирмы Winbond.

### 7.3.3. Энергонезависимая память с последовательными интерфейсами

Для микросхем энергонезависимой памяти малого объема, от которых не требуется высокой производительности обмена данными, часто применяют последовательные интерфейсы. Это позволяет упаковывать микросхемы памяти любого объема в корпуса, имеющие минимальное число выводов (рис. 7.23, 7.24, табл. 7.29). С таким интерфейсом выпускаются микросхемы EEPROM, FRAM и флэш-памяти. Микросхемы EEPROM и флэш-памяти выполняют внутренние операции записи автономно; о завершении выполнения операции можно судить



по результатам опроса ее состояния. Более сложные микросхемы имеют блочную организацию и средства управления доступом к каждому блоку с помощью программируемых регистров состояния и внешнего вывода управления записью (программированием). Микросхемы FRAM выполняют все операции на скорости интерфейса (на то они и RAM). Существуют модификации микросхем, позволяющие блокировать запись данных пользователем в определенную область (или всю микросхему, что превращает ее в ROM). Вывод управления защитой у разных типов микросхем функционирует и называется по-разному: WP# — Write Protect, WC — Write Control, PP — Programm Protect. Для выбора микросхемы используются либо входы задания внутреннего адреса A[0:2], либо сигнал выборки CS#, с помощью которого контроллер может обратиться к одному из требуемых устройств. Для упрощения внешних схем могут использоваться и несколько сигналов выборки S[0:2], один из которых (S1) иногда инвертирован.

Таблица 7.29. Популярные микросхемы памяти с последовательным интерфейсом

Микросхема	Организация	Рисунок	Примечание
24C01.24C01	16x8, 128x8	7.23, а	1 <sup>2</sup> C. Выводы 1, 2, 3, 7 = NC
24C02.24C164	256x8, 2Kx8	7.23, а	
24F016	2Kx8	7.23, б	
24F128	16Kx8	7.23, в	
X76F041	512x8	7.23, г	
FM24C04.FM24C16, FM24C64, FM24C256	512x8, 2Kx8, 8Kx8, 32Kx8	7.24 а	SPI
FM25040.FM25160, FM25256	512x8, 2Kx8, 64Kx8	7.24 б	

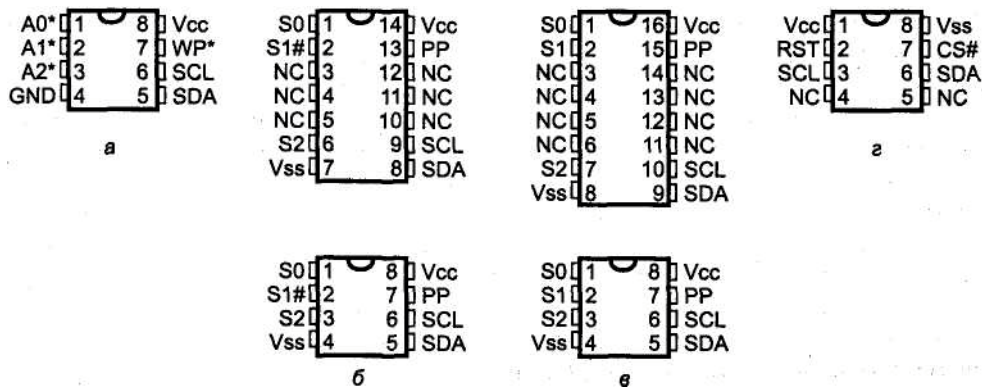


Рис. 7.23. Назначение выводов микросхем EEPROM с интерфейсом I<sup>2</sup>C: а — 24Cxx, б — 24F016, в — 24F128, г — X76F041

Микросхемы EEPROM 24C02 с интерфейсом I<sup>2</sup>C объемом 256 байт применяются для последовательной идентификации модулей DIMM-168 второго поколения, DIMM-184 и SODIMM-144.

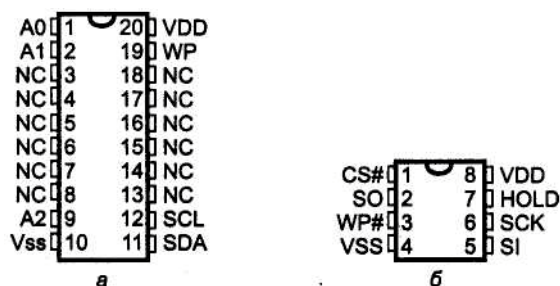


Рис. 7.24. Назначение выводов микросхем FRAM: а — FM24C256, б — FM25640

Кроме обычных устройств энергонезависимой памяти с интерфейсом I<sup>2</sup>C выпускают и специализированные *устройства защиты* (Security Devices). Например, микросхема X76F041, представляющая собой 4 блока памяти по 128 байт, имеет 64-битный регистр пароля, доступный только по записи. Обращение к микросхеме возможно только при предъявлении правильного пароля (который считать невозможно в принципе). Программируемый управляющий регистр (тоже энергонезависимый) позволяет для каждого блока установить свой режим доступа (полный доступ, только чтение, возможность только обнуления бит при записи, доступ только по предъявлении пароля конфигурации). Кроме того, есть возможность включения режима саморазрушения после превышения количества попыток доступа с неверным паролем. Такие устройства могут применяться в аппаратных ключах, защищающих программные средства от несанкционированного исполнения и пиратского копирования.

Последовательные интерфейсы памяти имеются на специальных устройствах-программаторах и копировщиках; их имеют многие модели микроконтроллеров, а там, где нет специальных аппаратных средств, они могут быть легко реализованы программно, занимая несколько бит портов общего назначения. В IBM PC может присутствовать внутренний интерфейс PC для считывания последовательных идентификаторов модулей памяти (в виде шины SMBus, см. п. 11.1.3). Последовательный интерфейс памяти используется во многих микросхемах программируемой логики (например, в популярных устройствах FPGA фирмы Altera), применяемых в различных устройствах, включая карты расширения PC. Эти микросхемы при инициализации (по включению питания, сбросу) сами «засасывают» описание своей конфигурации (или программное обеспечение) с рядом стоящей микросхемы энергонезависимой памяти. Внешние последовательные интерфейсы, в которых нуждаются разработчики и изготовители электронной аппаратуры с последовательной памятью, могут быть реализованы программно, с использованием несложных адаптеров для LPT-порта.

Для микросхем памяти часто применяют интерфейсы PC, SPI (Microwire) и иные «безымянные», называемые просто 2-, 3- и 4-проводными (считаются только сигнальные линии, общий провод подразумевается). Некоторые из этих интерфейсов являются собственными изобретениями разработчиков микросхем памяти (необходимые сведения обычно имеются в информационных листках), здесь же остановимся на наиболее популярных.

## Память с интерфейсом I<sup>2</sup>C

Широко распространенный *двухпроводный интерфейс PC* (см. п. 11.1.1) обеспечивает невысокую скорость передачи (до 100 или 400 Мбит/с).

Диаграмма обмена данными с памятью по интерфейсу I<sup>2</sup>C приведена на рис. 7.25. Выполнив условие *Start*, ведущее устройство передает байт, содержащий адрес устройства, и признак операции RW и ожидает подтверждения. При операции записи следующей посылкой от ведущего устройства будет 8-битный адрес записываемой ячейки, а за ней — байт данных (для микросхем объемом памяти более 256 байт адрес ячейки посылается двумя байтами). Получив подтверждения, ведущее устройство завершает цикл условием *Stop*, а адресованное устройство может начать свой внутренний цикл записи, во время которого не реагирует на сигналы интерфейса. Ведущее устройство может проверять готовность устройства к следующей операции посылкой команды записи (байт адреса устройства) и анализом бита подтверждения, формируя затем условие *Stop*. Если устройство откликнулось битом подтверждения, значит, оно завершило внутренний цикл и готово к следующей операции.

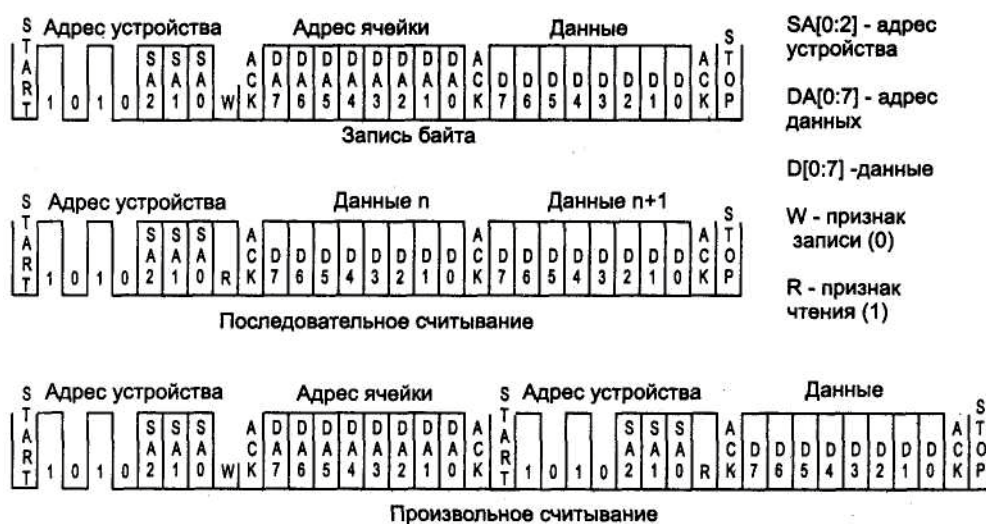


Рис. 7.25. Диаграмма обмена с памятью по интерфейсу I<sup>2</sup>C

Операция считывания инициируется так же, как и запись, но с признаком RW=1. Возможно чтение по заданному адресу, по текущему адресу или последовательное. Текущий адрес хранится во внутреннем счетчике микросхемы памяти; он содержит увеличенный на единицу адрес ячейки, участвовавшей в последней операции. Получив команду чтения, устройство дает бит подтверждения и посылает байт данных, соответствующий текущему адресу. Контроллер может ответить подтверждением, тогда устройство пошлет следующий байт (последовательное чтение). Если на принятый байт данных контроллер ответит условием *Stop*, операция чтения завершается. Начальный адрес для считывания контроллер задает фиктив-

ной операцией записи, в которой передается адрес устройства и адрес ячейки, а после подтверждения приема адреса ячейки формируется повторный старт (S г) и передается адрес устройства, но уже с указанием на операцию чтения. Так реализуется считывание произвольной ячейки (или последовательности ячеек). Отметим, что спецификация на интерфейс РС не оговаривает правил модификации адреса данных при последовательных обращениях — их определяет разработчик устройства в соответствии с его функциями. Для памяти вышеописанный автоинкремент адреса упрощает последовательные обращения. Заметим, что последовательная память малого объема может использовать и упрощенный вариант интерфейса, правда, и не называемого РС. Так, например, память Atmel AT24C01 объемом 128 байт используют упрощенную систему адресации: адрес ячейки передается вместо 7-битного адреса устройства РС, а в остальной операции записи и чтения выполняются по тем же правилам. При этом шина вырождается в двухточечный интерфейс, но в ряде случаев этого и достаточно. Специальные адреса здесь не требуются, так что все 128 значений 7-битного адреса доступны для адресации памяти. Микросхемы той же фирмы объемом 256 байт используют вышеприведенный механизм адресации устройств РС с возможностью объединения до 8 устройств. А в микросхемах объемом 512, 1024 и 2048 байт 1, 2 или все 3 бита номера устройства в адресе отбираются под адрес 256-байт-ной страницы памяти. Память большого объема использует 7-битную адресацию устройств, обычную для РС, а адрес ячейки задается последующими двумя байтами (сначала старшим, затем младшим).

### **Память с интерфейсом SPI**

Трехпроводный интерфейс SPI (см. п. 11.3) обеспечивает скорость передачи до 5 Мбит/с. Обмен с микросхемой, выбранной сигналом CS#, начинается с подачи по линии SI 8-битного кода команды. В команде записи за кодом команды следует адрес ячейки, за которым следуют байты данных. Команда чтения начинается так же, но после приема адреса в следующем же такте по линии SO начинается вывод данных. Есть и специальные безадресные команды, например, разрешение/запрет записи и стирания. Микросхемы памяти с SPI имеют регистр состояния, доступ к которому обеспечивается специальными командами. В этом регистре отражается состояние управляющего автомата записи (занят/свободен), в нем же могут присутствовать и биты управления защитой записи (в регистр состояния возможна и запись). Состав регистров и набор команд зависит от модели устройства.

## Глава 8

# Специализированные интерфейсы периферийных устройств

В этой главе описываются интерфейсы периферийных устройств, от самых необходимых (клавиатура, дисплей, мышь, принтер) до мультимедийных (видео- и аудиоустройства) и развлекательных (джойстик и его «родственники»).

### 8.1. Интерфейс клавиатуры

Традиционная клавиатура PC представляет собой унифицированное устройство ввода со стандартным разъемом и последовательным интерфейсом связи с системной платой. В настоящее время используются так называемые *расширенные* (enhanced) *клавиатуры* AT или PS/2, имеющие более 100 клавиш. Они вытеснили первые 84-клавишные клавиатуры AT и тем более клавиатуры XT. По электрическому интерфейсу клавиатуры XT и AT совпадают, за исключением того, что двунаправленный интерфейс позволяет клавиатуре AT принимать команды от системной платы. Однако по логическому интерфейсу они несовместимы (клавиатура AT иногда имеет переключатель режима XT/AT). Клавиатура PS/2 отличается от AT только исполнением разъема, при необходимости можно использовать переходник.

Клавиатуры имеют внутренний микроконтроллер, способный определить факты нажатия и отпускания клавиш, при этом можно нажимать очередную клавишу, даже удерживая несколько ранее нажатых. При нажатии клавиши клавиатура передает идентифицирующий ее *скан-код*. При удержании клавиши в нажатом положении через некоторое время клавиатура начинает автоповтор передачи скан-кода нажатия этой клавиши. Задержка автоповтора (typematic delay) и скорость автоповтора (typematic rate) для клавиатур AT программируются. Расширенная клавиатура позволяет выбирать 1 из 3 наборов скан-кодов.

С распространением шины USB появились клавиатуры и с этим интерфейсом; они имеют и встроенный хаб, например для подключения мыши USB. Клавиатура USB питается от шины. Для клавиатуры USB требуется специальная поддержка со стороны BIOS; она имеется в современных системных платах.

### 8.1.1. Интерфейс клавиатуры AT и PS/2

Для подключения клавиатуры предназначен последовательный синхронный двунаправленный интерфейс, состоящий из двух обязательных сигналов KB-Data и KB-Clock. Обе линии на системной плате подтягиваются резисторами к шине +5 В. На обеих сторонах интерфейса выходные сигналы низкого уровня формируются выходами элементов с открытым коллектором (стоком), а состояние линий может быть прочитано через входные линии контроллеров. Вид разъемов (со стороны задней панели) и назначение контактов приведены на рис. 8.1. Конструктивно возможны два варианта разъема — обычная 5-контактная розетка DIN (клавиатура AT) или малогабаритная розетка mini-DIN (PS/2). На этот же разъем через плавкий предохранитель поступает напряжение питания клавиатуры +5 В.

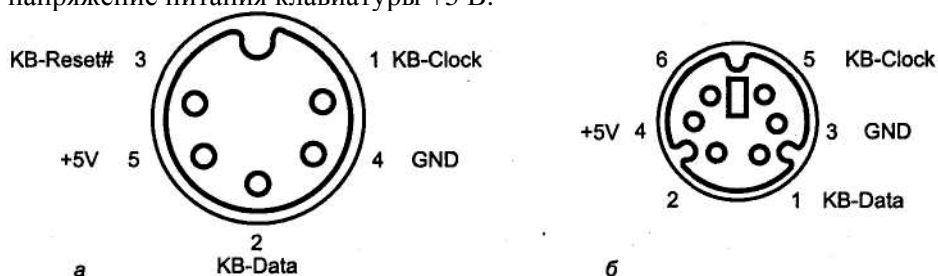


Рис. 8.1. Разъемы (вид со стороны контактов) подключения клавиатур: а — AT и б — PS/2

#### ВНИМАНИЕ

Питание от разъема клавиатуры часто используется такими устройствами, как внешние накопители или адаптеры локальных сетей, подключаемыми к параллельному порту. Плавкий предохранитель, установленный на системной плате, может не выдержать броска тока, потребляемого этими устройствами. При этом откажется работать и клавиатура — ее индикаторы даже и не мигнут при включении.

Процессор общается с клавиатурой через *контроллер интерфейса клавиатуры* — микроконтроллер 8042 или программно-совместимый с ним, установленный на системной плате. Для обмена информацией в основном используется порт 60h, из которого принимаются скан-коды. О необходимости чтения скан-кода контроллер сигнализирует процессору через аппаратное прерывание IRQ1, сигнал которого вырабатывается по каждому событию клавиатуры (нажатию и отпусканью клавиши). Задание параметров автоповтора, выбор таблиц скан-кодов, управление светодиодными индикаторами, а также управление режимом сканирования матрицы клавиш и запуск диагностического теста осуществляется командами, посылаемыми в этот же порт. Контроллер транслирует команды в посылки, направляемые к клавиатуре.

Работу двунаправленного интерфейса иллюстрирует рис. 8.2, а, где серым цветом помечены сигналы, формируемые контроллером, а черным — клавиатурой. В исходном состоянии обе линии «отпущены» выходными формирователями в состояние с высоким уровнем. Клавиатура может начать передачу данных в произволь-

ный момент, когда интерфейс находится в покое. Клавиатура формирует стартовый бит (низкий уровень) на линии KB-Data и первый импульс KB-Clock, что является сигналом контроллеру о необходимости начала приема. После подъема KB-Clock она выводит 0-й бит данных на линию KB-Data, а затем и следующий импульс KB-Clock. Контроллер должен «зашелкивать» принятый бит данных по спаду KB-Clock. Так передаются все 8 бит данных и бит паритета, дополняющий число единичных бит до нечетного. После синхроимпульса бита паритета *контроллер клавиатуры* должен сформировать импульс KB-Clock, подтверждающий прием байта (Ack). Если весь байт с битом паритета не будет получен контроллером за 2 мс, контроллер прекращает прием данного байта и фиксирует ошибку тайм-аута.

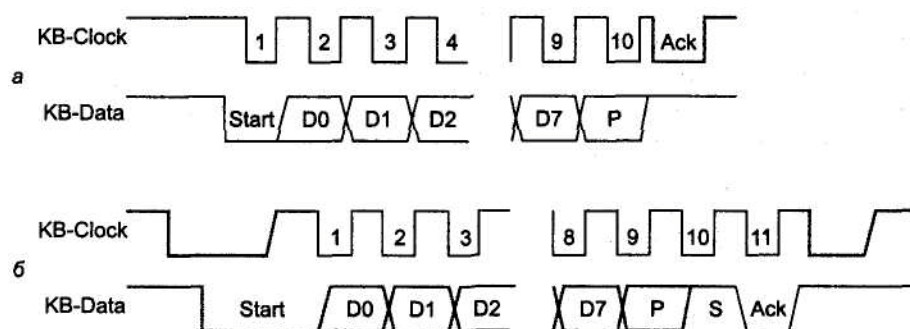


Рис. 8.2. Временные диаграммы интерфейса клавиатуры: а — прием посылки от клавиатуры, б — передача команды в клавиатуру

Обратная передача — вывод команды контроллера в клавиатуру — происходит несколько сложнее (рис. 8.2, б). Из состояния покоя *контроллер* устанавливает низкий уровень KB-Clock на 250 мкс и формирует старт-бит (низкий уровень) — это сигнал клавиатуре на прием команды. На него *клавиатура* должна ответить серией из 11 импульсов KB-Clock. По спаду очередного синхроимпульса контроллер выставляет очередной бит данных, а клавиатура его «зашелкивает» по фронту формируемого ею же синхроимпульса. После бита паритета (9-й импульс) и единичного стоп-бита (10-й) на 11-м импульсе *клавиатура* формирует нулевой бит подтверждения (Ack). После этого *контроллер* формирует импульс KB-Clock (60 мкс), который является запросом на прием ответа клавиатуры. Контроллер ожидает окончания ответа на этот запрос не более 20 мс и, если ответ не придет за это время, сформирует ошибку тайм-аута. Ошибка будет также в случае, если клавиатура не введет первые данные, включая стоп-бит, за 2 мс с момента появления синхроимпульса бита 0.

На системной плате PC/XT контроллера 8042 не было, а интерфейс клавиатуры (однаправленный) был реализован аппаратной логикой — регистром сдвига, параллельный выход которого подключается к входам порта A системного интерфейса 18255. По приему байта от клавиатуры вырабатывается аппаратное прерывание IRQ1, обработчик которого может прочитать принятый байт из порта 60h. С помощью бит 7 и 6 порта 61h возможны программная блокировка и сброс клавиатуры соответственно. Сброс клавиатуры XT осуществляется обнулением линии KB-Clock.

## 8.1.2. Контроллер интерфейса клавиатуры и мыши 8042/8242

*Программируемый микроконтроллер клавиатуры 18042*, КВС (keyboard controller), является посредником между клавиатурой, подключенной к нему по вышеописанному последовательному интерфейсу, и центральным процессором, с которым он связан через параллельный интерфейс. В микроконтроллере постоянно исполняется внутренняя микропрограмма, реагирующая на сигналы интерфейса клавиатуры и команды, поступающие от процессора. Эта микропрограмма (КВС BIOS) хранится во внутреннем масочном ПЗУ контроллера; внешне она недоступна, и контроллер можно рассматривать как устройство с заданными свойствами. Поскольку логика работы контроллера реализована программой, его реакция на команды процессора и сигналы интерфейса относительно медленная — время отклика измеряется десятками микросекунд. Помимо управления клавиатурой через программно-управляемые и программно-читаемые линии внешних портов контроллера формируются сигналы управления вентилем *Gate A20*, аппаратного системного сброса, а также считываются сигналы от конфигурационных джамперов системной платы. Контроллер 18242, кроме интерфейса клавиатуры, поддерживает аналогичный интерфейс дополнительного устройства — PS/2-Mouse. При инициализации (по аппаратному сбросу) контроллер установится в режим PS/2 или AT, в зависимости от состояния определенного вывода микросхемы. В режиме AT контроллер не выполняет функции интерфейса мыши и игнорирует все команды, относящиеся к мыши. Режим выбирается соответствующей настройкой BIOS Setup.

Связь контроллера с центральным процессором обеспечивается через 8-битную шину данных. Контроллер выбирается сигналом CS# от дешифратора адреса, срабатывающего по адресам 0060h и 0064h; внутренние регистры контроллера выбираются линией SA2 системной шины адреса. Чтение и запись выполняется по сигналам IORD\* и IOWR#, генерируемых при выполнении процессором инструкций IN и OUT. Таким образом, контроллер располагается в пространстве ввода-вывода CPU по адресам 60h (регистр данных) и 64h (регистр состояния и команд), назначение регистров приведено в табл. 8.1. Из *регистра данных* считываются данные, принимаемые по интерфейсам от клавиатуры и мыши, а также данные, возвращаемые контроллером в ответ на адресованные ему команды. *Запись в регистр данных* используется для подачи команд и данных, адресованных к клавиатуре и мыши, а также данных для команд, адресованных контроллеру. В *регистр команд* записываются команды, адресованные контроллеру. Режим работы контроллера (разрешение работы интерфейсов клавиатуры и мыши и прерываний от них, трансляция скан-кодов и другие параметры) задается *командным байтом*, посылаемым в контроллер по специальной команде. Перед любой записью в контроллер необходимо убедиться в его готовности. Признаком готовности/занятости контроллера является значение бита 1 регистра состояния (порт 064h).



Таблица 8.1. Назначение регистров контроллера клавиатуры

Порт, R/W	Назначение
060 RW	Порт данных 8042
064 R	Регистр состояния 8042 (R/O): бит 7 — ошибка четности при последнем обмене с клавиатурой; бит 6 — тайм-аут приемника/общий тайм-аут <sup>1</sup> ; бит 5 — тайм-аут передатчика/выходной буфер интерфейса мыши полон (Mouse,DBF) <sup>1</sup> ; бит 4: 0 — клавиатура на замке; бит 3:1 — последняя запись была командой, 0 — данными; бит 2: системный флаг, устанавливается в 0 по включении питания, в 1 — программно (что означает состояние завершения системного сброса Reset OK); бит 1:1 — входной буфер интерфейса клавиатуры полон, 0 — готовность к приему команды/данных; бит 0:1 — выходной буфер интерфейса клавиатуры полон (DBF)
064 W	Регистр команд 8042

<sup>1</sup> Второе назначение бита относится к контроллеру i8242B, имеющему дополнительный интерфейс для подключения PS/2-Mouse.

Контроллер имеет два внешних порта, с помощью которых и реализуются последовательные интерфейсы, а также управление вентилем GateA20, сигналом сброса процессора и чтением сигналов от джамперов системной платы и ключа блокировки клавиатуры. Эти порты не имеют непосредственного отображения в пространстве адресов ввода-вывода PC; доступ к ним осуществляется через команды контроллера. Кроме этих портов контроллер имеет два специальных входа T0 и T1, которые могут считываться его микропрограммой и использоваться в качестве источников его внутренних прерываний (это не прерывания CPU). Каждая из двунаправленных интерфейсных линий KB-Data, KB-Clock, MS-Data и MS-Clock реализуется битом порта вывода и битом ввода. В режиме PS/2 для чтения линий KB-Clock и MS-Clock используются входы T0 и T1, в режиме AT вход T1 используется для линии KB-Data.

Порт ввода P1 доступен по команде C0h; в режиме PS/2 имеет следующее назначение бит:

- ◆ бит 7 — ключ блокировки клавиатуры (*Keylock*): 0 — клавиатура заблокирована;
- ◆ бит 6 — переключатель установки видеорежима: 0 — цветной (Color), 1 — монохромный (Mono);
- ◆ бит 5 — системная переключатель: 0 — замкнута;
- ◆ бит 4 — переключатель задания объема ОЗУ: 0 — 256 Кбайт, 1 — 512 Кбайт и более;
- ◆ биты 3, 2 — не используются;
- ◆ бит 1 — линия MS-Data;
- ◆ бит 0 — линия KB-Data.

В современных ПК используются только биты 0,1 и 7; в режиме AT биты 0 и 1 не используются.

Порт вывода (P2), доступный для записи и чтения по командам D1h и D0h соответственно, имеет следующее назначение бит:

- ◆ бит 7 — линия KB-Data;
- ◆ бит 6 — линия KB-Clk;

- ◆ бит 5 — запрос прерывания от дополнительного интерфейса (IRQ12), в режиме АТ не используется;
- ◆ бит 4 — запрос прерывания от клавиатуры (IRQ1);
- ◆ бит 3 — линия MS-Clk, в режиме АТ не используется;
- ◆ бит 2 — линия MS-Data, в режиме АТ не используется;
- ◆ бит 1 — вентиль линии адреса A20 (*Gate A20*, см. п. 12.3): 0 — A20 обнулен, 1 — управляется выводом процессора;
- ◆ бит 0 — альтернативный сброс процессора (сигнала INIT, без формирования общего сигнала сброса).

Приняв посылку от клавиатуры, контроллер выполняет внутреннюю трансляцию скан-кода (если это не запрещено командным байтом) и устанавливает в регистре состояния OBF=1, что приводит к генерации запроса прерывания IRQ1 (если это не запрещено командным байтом). В ответ на это хост должен считать данные (транслированный скан-код, префиксы и т. п.) из порта данных (60h). Трансляция обеспечивает программную совместимость клавиатур XT и АТ (см. ниже) по чтению скан-кодов из порта 60h. При получении посылки от мыши контроллер не выполняет никаких преобразований и устанавливает в регистре состояния Mouse\_OBF-1, что приводит к генерации запроса прерывания IRQ12 (если это не запрещено командным байтом). По этому сигналу данные от мыши должны быть считаны хостом из того же порта 60h. То же самое происходит и при программной записи байта в выходной буфер клавиатуры (код D2h) или мыши (код D3h), с установкой соответствующих битов состояния и генерацией запросов IRQ1 или IRQ 12.

После получения команды, по которой контроллер должен возвращать данные, он устанавливает в регистре состояния OBF=1, что приводит к генерации запроса прерывания IRQ1 (если это не запрещено командным байтом). После этого данные должны быть считаны из порта данных (по адресу 60h). Если команда возвращает несколько байтов данных, прерывание генерируется для каждого байта.

Взаимодействие с контроллером осуществляется подачей *команд* — записью в регистр команд (по адресу 064h). Перед посылкой команды необходимо убедиться в готовности контроллера к ее приему — бит 1 порта 64h должен иметь нулевое значение. Подробно команды контроллера описаны в литературе [2, 8]. Контроллеры клавиатуры различаются версиями встроенного ПО, поэтому замена контроллера другим, с отличающейся версией ПО (КВС BIOS), может оказаться проблематичной — микросхема BIOS системной платы должна знать особенности контроллера клавиатуры.

### 8.1.3. Системная поддержка и программный интерфейс

Клавиатура имеет системную поддержку на уровне BIOS — обработку фактов нажатия и отпускания клавиш и обеспечение сервисов ввода символов с клавиатуры, а также управления ее параметрами (задержка и частота автоповтора) и индикаторами. Коды, принятые от клавиатуры ее контроллером, считываются и обра-

батываются обработчиком аппаратного прерывания IRQ1 (вектор 09h). Результат обработки помещается в *клавиатурный буфер*, из которого по программному прерыванию Int 16h этот результат для дальнейшей обработки может быть извлечен значительно позже. Нажатие «системной» комбинации Ctrl+Alt+Del, клавиши PrintScreen (SysRq) и некоторых других к записи в клавиатурный буфер не приводит, а вызывает специальные процедуры. Прикладной программе, для которой требуется нестандартное использование клавиатуры (например, в качестве музыкальной), придется самой заниматься обработкой аппаратного прерывания IRQ1, перехватывая вектор Int 09h. Перехват этого вектора требуется и для вызова каких-либо функций резидентных программ по «горячим» клавишам.

При начальном тестировании процедура POST инициализирует клавиатуру (и ее контроллер) и запускает диагностический тест. Во время этого теста клавиатура мигнет всеми индикаторами, после чего может остаться включенным только индикатор NumLock (зависит от установки в BIOS Setup). В случае обнаружения ошибки клавиатуры на консоль выводится сообщение с возможным указанием скан-кода залипшей клавиши и обычно предлагается нажать клавишу F1 для продолжения. То же самое произойдет, если тест не обнаружит клавиатуру (например, из-за вывалившегося разъема или перегоревшего предохранителя), но в этом случае нажатия клавиши F1 будет уже недостаточно. Ошибку диагностики даст и подключение к компьютеру АТ клавиатуры от XT, обратное «скрещивание» тоже неработоспособно. Чтобы начальная загрузка не останавливалась по ошибке (отсутствию) клавиатуры (POST будет дожидаться получения кода клавиши F1), тестирование клавиатуры может быть отменено настройкой CMOS Setup.

Для обслуживания клавиатуры используются ячейки ОЗУ из области данных BIOS (BIOS Data Area):

- ◆ 0:0417,0:418 — флаги клавиатуры;
- ◆ 0:0419 — аккумулятор кода Alt-набора;
- ◆ 0:041 A — указатель головы буфера (Buffer Head), 2байта (модифицируется при помещении символа в буфер);
- ◆ 0:041C — указатель хвоста буфера (Buffer Tail), 2 байта (модифицируется при извлечении символа из буфера);
- ◆ 0:041E-0:042D — область кольцевого буфера (16 слов).

## 8.2. Интерфейсы мыши

Мышь является устройством, предназначенным для ввода координат и подачи команд. Интерфейс мыши применим для любого физического воплощения устройства (мышь, трекбол). По интерфейсу с компьютером различают три основных вида мышей: Bus Mouse, Serial Mouse и PS/2-Mouse. Появились мыши с интерфейсом USB, но они пока не получили широкого распространения (как и клавиатура USB, к порту которой удобно подключить мышь USB).

С интерфейсами Serial Mouse и PS/2-Mouse иногда возникают недоразумения. Хотя оба они последовательные, но имеют существенные принципиальные различия в уровнях сигналов, способе синхронизации, частоте и формате посылок.

- ◆ Интерфейс PS/2 использует однополярный сигнал с уровнями ТТЛ, питание мыши — однополярное с напряжением +5 В относительно шины GND. Интерфейс RS-232C, применяемый в Serial Mouse, использует двухполярный сигнал (см. п. 2.1) с уровнями срабатывания +3 В и -3 В, и для него требуется двухполярное (относительно шины GND) питание мыши.
- ◆ Интерфейс PS/2 использует две отдельные сигнальные линии, одну для передачи данных, другую — для сигналов синхронизации. Serial Mouse использует асинхронный способ передачи данных всего по одной линии.

Даже не рассматривая частоты и форматы посылок, становится ясно, что прямой совместимости между этими интерфейсами быть не может. Тем не менее выпускаются и продаются переходники (пассивные!), позволяющие выбирать способ подключения мыши. Эти переходники предназначены только для универсальных мышей, у которых встроенный контроллер по напряжению питания способен распознать, к какому интерфейсу его подключили, и установить соответствующий тип своего выходного интерфейса. Универсальные мыши не особо распространены, поэтому часто приходится слышать о неудачных попытках применения таких переходников к обычной мыши Serial Mouse или PS/2-Mouse.

Дополнительную путаницу вносят мыши для компьютера Macintosh, которые имеют разъем, с виду напоминающий разъем PS/2. Однако при ближайшем рассмотрении и неудачной попытке включения его в PC становится ясно, что разъемы эти разные, да и интерфейс совершенно иной.

### 8.2.1. Последовательные мыши — MS Mouse и PC Mouse

*Serial Mouse* — мышь с последовательным интерфейсом, подключаемая через 9- или 25-контактный разъем COM-порта (табл. 8.2). Эта мышь имеет встроенный микроконтроллер, который обрабатывает сигналы от координатных датчиков и кнопок. Каждое событие — перемещение мыши или нажатие-отпускание кнопки кодируется двоичной посылкой по интерфейсу RS-232C. Для передачи информации применяется асинхронная передача, а двухполярное питание, требуемое по протоколу RS-232, обеспечивается от управляющих линий интерфейса. Недостатком Serial Mouse является то, что она занимает COM-порт и требует монополярного владения его штатной линии прерывания (IRQ4 для COM1 и IRQ3 для COM2). Конечно, то, что для использования мыши порту COM1 требуется именно прерывание IRQ4, является недостатком не самой мыши, а ее программного драйвера, но для пользователя, не увлекающегося написанием «мышинных» драйверов, важен только факт этого ограничения. Две основные разновидности — *MS Mouse* (Microsoft Mouse) и *PC Mouse* (Mouse Systems Mouse) — требуют раз-

ных драйверов, многие мыши имеют переключатель MS/PC. Эти два типа «мышей» при одинаковой скорости 1200 бит/с, одном стоп-бите и отсутствии контроля паритета используют различные форматы посылок.

- ◆ *MS Mouse*: 1 бит данных, трехбайтный пакет (в «классическом» варианте), положительным значениям соответствует перемещение по координате X вправо, а по координате Y вниз. Для трехкнопочных мышей добавляется четвертый байт, передаваемый только при изменении состояния средней кнопки. Для 3D-мыши четвертый байт имеет иное назначение.
- ◆ *PC Mouse*: 8 бит данных, пятибайтный пакет, положительным значениям соответствует перемещение по координате X вправо, а по координате Y вверх.

Из рассмотрения данных форматов становятся понятными беспорядочные перемещения указателя мыши на экране при несоответствии драйвера типу мыши. Несовместимость может проявляться и более неприятным образом: к примеру, ОС Windows 95 при загрузке (и установке) вообще не воспринимает мышь, работающую в режиме PC Mouse (ей «не нравится» идентификатор, сообщаемый мышью при инициализации). При загруженной ОС переключение режима приводит «только» к непредсказуемым прыжкам указателя мыши и ложным срабатываниям кнопок.

Таблица 8.2. Разъемы Serial Mouse

Сигнал	Контакт DB9	Контакт DB25	Цепь COM-порта
Data	2	3	RxD
GND	5	7	GND
+V(питание)	7,(4)	4, (20)	RTS, (DTR)
-V (питание)	3	2	TxD

*Системная поддержка последовательной мыши* осуществляется только на уровне ОС (сервисы вызываются через Int 33п), драйвер мыши — загружаемый или встроенный в ОС. BIOS мышь не поддерживает, даже если и пользуется ею для навигации в BIOS Setup. Еще раз подчеркнем, что для работы мыши обязательно требуется линия аппаратного прерывания — IRQ4 или IRQ3 для последовательных мышей на портах COM1 или COM2 соответственно.

## 8.2.2. Мышь PS/2

*PS/2-Mouse* — мышь, появившаяся с компьютерами PS/2. Ее интерфейс и разъем 6-pin mini-DIN аналогичен клавиатурному (см. рис. 8.1) и, как правило, реализуется тем же контроллером клавиатуры 8242 (см. п. 8.1.2). Адаптер и разъем PS/ 2-Mouse устанавливаются на многих современных системных платах (рис. 8.3). Контроллер мыши PS/2 может быть также на карте расширения (ISA) и занимать дополнительные адреса в пространстве ввода-вывода. С мышью PS/2 связь двусторонняя: процессор может посылать контроллеру 8242 специальные команды,

но, в отличие от интерфейса клавиатуры, перед записью в порт 60h каждого «мышинного» байта (и команды, и ее параметра) в порт 64h должен записываться код D4h.

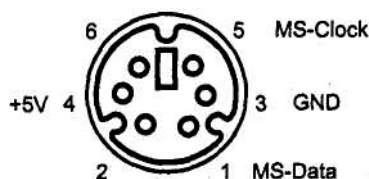


Рис. 8.3. Разъем PS/2-Mouse

Мышь может работать в одном из двух режимов. В *поточном режиме* (stream mode) мышь посылает данные по любому изменению состояния; в *режиме опроса* (remote mode) мышь передает данные только по запросу процессора. Есть еще *диагностический режим* (wpar mode), в котором мышь возвращает эхом данные, посылаемые ей контроллером. По приему пакета от мыши контроллер устанавливает флаг Mouse\_OBF и вырабатывает прерывание IRQ12, если оно не запрещено командным байтом 8242.

Устройства-указатели с интерфейсом PS/2 (мышь PS/2) имеют *поддержку BIOS*, обеспечивающую настройку параметров мыши (посылку вышеперечисленных команд). Собственно драйвер мыши (обработчик прерывания по вектору 74h от запроса IRQ12), обрабатывающий ее информационные послышки, входит лишь в состав ОС или загружается отдельно. Поддержка мыши вызывается через BIOS Int 15h с кодами функций C200-C209h.

### 8.2.3. Мышь Bus Mouse

*Bus Mouse* (шинная мышь) — вариант, применявшийся в первых мышах. Здесь мышь содержит только датчики и кнопки, а обработка их сигналов производится на специализированной плате адаптера (обычно ISA). Кабель 9-проводный, разъем специальный (рис. 8.4), хотя на первый взгляд и напоминающий разъем PS/2-Mouse. Главный недостаток такой системы заключается в том, что адаптер занимает слот системной шины, адреса ввода-вывода и линию запроса прерывания. Иногда встречались мультипортовые карты ISA (COM-, LPT- и GAME-пор-ты), на которых установлен и адаптер Bus Mouse. Поскольку компания Microsoft одна из первых выпустила такую мышь, снабдив ее своим логотипом, с понятием Bus Mouse иногда отождествляют и MS-Mouse, хотя последние могут иметь любой из трех видов интерфейсов.

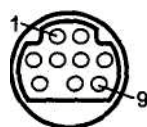


Рис. 8.4. Разъем Bus Mouse

### 8.3. Интерфейсы принтеров и плоттеров

Современные принтеры, печатающие графические изображения (в том числе и текст в графическом режиме) с высоким разрешением, требуют высокоскоростной передачи данных по внешнему интерфейсу. Большинство принтеров имеют традиционный параллельный интерфейс Centronics или более производительный IEEE 1284, что позволяет достигать скоростей передачи 0,15-2 Мбайт/с, в зависимости от производительности компьютера и выбранного режима передачи (см. ниже). Эти же интерфейсы используются и в плоттерах.

Для подключения принтера с параллельным интерфейсом используется LPT-порт в различных модификациях, от традиционного SPP-порта до теперь уже стандартного и эффективного IEEE 1284 (см. п. 1.3).

Поначалу все принтеры с параллельным интерфейсом обязательно поддерживали протокол Centronics, а более «продвинутые» вдобавок могли работать и в режиме ECP, поддерживая согласование режимов по IEEE 1284. Их инсталляционные программы старались установить драйверы «продвинутых» режимов, если того позволяла ОС и возможности LPT-порта. Теперь ситуация изменилась, и появились принтеры с параллельным интерфейсом, не поддерживающие Centronics. При инсталляции они требуют подключения по «двунаправленному интерфейсу» IEEE 1284 (обычно режим ECP), и через LPT-порт в режиме SPP они работать отказываются. С такими принтерами в среде MS DOS без специальных драйверов работать невозможно.

В некоторых принтерах используется последовательный интерфейс RS-232C, RS-422 или «токовая петля», но здесь теоретический предел скорости около 11 Кбайт/с (115 Кбит/с), а практически она едва достигает 1 Кбайт/с (9600 бит/с). Эти принтеры можно подключать к COM-порту непосредственно или через адаптер — преобразователь уровня сигналов.

В последнее время стали чаще применять шину USB, однако этот переход не так уж безоблачен: шина USB поддерживается не всеми ОС. Старые приложения (ведь не всегда есть необходимость и возможность перехода на новые), работающие с принтером через функции BIOS Int 17h или непосредственно с регистрами LPT-порта, например для приглашения к подаче бланков, не могут работать с принтером USB даже в среде ОС, поддерживающих USB в полном объеме. Что касается скорости передачи данных, то у USB 1.0 со скоростью 12 Мбит/с скорость передачи данных отнюдь не достигает 1,5 Мбайт/с (12:8) хотя бы из-за накладных расходов шины. В USB 2.0, которая сейчас выходит на рынок, пиковая скорость может достигать 50 Кбайт/с (скорость в шине — 480 Мбит/с), что для принтера пока что более чем достаточно. Однако для этого и принтер, и компьютер должны поддерживать USB 2.0, и между ними не должно быть старых (USB 1.0) хабов.

Принтеры могут иметь интерфейс SCSI (редкий вариант), а также подключаться не к компьютеру, а к локальной сети по интерфейсу Ethernet (10 или 100 Мбит/с). Такое подключение удобно для принтеров коллективного пользования, и при грамотно построенной сети оно не доставляет забот пользователям. Шина Fire Wire для принтеров применяется пока очень сдержанно.

### 8.3.1. Параллельные интерфейсы — Centronics, IEEE 1284, IPNP

Параллельный интерфейс *Centronics* ориентирован на передачу потока байт данных к принтеру и прием сигналов состояния принтера. Этот интерфейс поддерживается всеми LPT-портами компьютеров. Его отечественным аналогом является интерфейс *IPNP-M*. Понятие «Centronics» относится как к набору сигналов и протоколу взаимодействия, так и к 36-контактному разъему на принтерах. Назначение сигналов интерфейса приведено в табл. 8.3, а временные диаграммы обмена с принтером показаны на рис. 8.5.

Таблица 8.3. Сигналы интерфейса Centronics

Сигнал	I/O <sup>1</sup>	Контакт	Назначение
Strobe#	I	1	Строб данных. Данные фиксируются по низкому уровню сигнала
Data [0:7]	I	2–9	Линии данных. Data 0 (контакт 2) — младший бит
Ack#	O	10	Acknowledge — импульс подтверждения приема байта (запрос на прием следующего). Может использоваться для формирования запроса прерывания
Busy	O	11	Занято. Прием данных возможен только при низком уровне сигнала
PaperEnd	O	12	Высокий уровень сигнализирует о конце бумаги
Select	O	13	Сигнализирует о включении принтера (обычно в принтере соединяется резистором с цепью + 5 В)
Auto LF#	I	14	Автоматический перевод строки. При низком уровне принтер, получив символ CR (Carriage Return — возврат каретки), автоматически выполняет и функцию LF (Line Feed — перевод строки)
Error#	O	32	Ошибка: конец бумаги, состояние OFF-Line или внутренняя ошибка принтера
Init#	I	31	Инициализация: сброс в режим параметров умолчания, возврат к началу строки и страницы
Select In#	I	36	Выбор принтера (низким уровнем). При высоком уровне принтер не воспринимает остальные сигналы интерфейса
GND	–	19–30, 33	Общий провод интерфейса

<sup>1</sup> I/O задает направление (вход или выход) применительно к принтеру.



Рис. 8.5. Передача данных по протоколу Centronics

Передача данных начинается с проверки готовности принтера — состояния линии Busy. Строб данных может быть коротким — доли микросекунды, и порт заканчи-



вадет его формирование, не обращая внимания на сигнал *Busy*. Во время строба данные должны быть действительными. Подтверждением приема байта (символа) является сигнал *Ask#*, который вырабатывается после приема строба через неопределенное время (за это время принтер может выполнять какую-либо длительную операцию, например прогон бумаги). Импульс *Ask#* является запросом принтера на прием следующего байта, его задействуют для формирования сигнала прерывания от порта принтера. Если прерывания не используются, то сигнал *Ask#* игнорируется и весь обмен управляется парой сигналов *Strobe\** и *Busy*. Свое состояние принтер может сообщить порту по линиям *Select*, *Error\**, *PaperEnd* — по ним можно определить, включен ли принтер, исправен ли он и есть ли бумага. Формированием импульса на линии *Init#* принтер можно проинициализировать (при этом он очистит и весь свой буфер данных). Режимом автоматического перевода строки, как правило, не пользуются, и сигнал *AutoLF\** имеет высокий уровень. Сигнал *SelectIn\** позволяет логически отключать принтер от интерфейса.

Через параллельный порт (LPT) протокол Centronics может быть реализован чисто программно, используя стандартный режим порта (*SPP*), достигая скорости передачи до 150 Кбайт/с при полной загрузке процессора. Благодаря «продвинутым» режимам порта протокол может быть реализован и аппаратно (*Fast Centronics*), при этом скорость до 2 Мбайт/с достигается при меньшей загрузке процессора.

Большинство современных принтеров с параллельным интерфейсом поддерживают и стандарт IEEE 1284, в котором оптимальным режимом передачи является ECP (см. п. 1.3.4).

Для подключения принтера требуется кабель Centronics, пригодный для любых режимов параллельного интерфейса. Простейший вариант кабеля — 18-проводный с неперевитыми проводами — может использоваться для работы в режиме *SPP*. При длине более 2 м желательно, чтобы хотя бы линии *Strobe\** и *Busy* были перевиты с отдельными общими проводами. Для скоростных режимов (*Fast Centronics*, ECP) такой кабель может оказаться непригодным — возможны нерегулярные ошибки передачи, возникающие лишь при определенных последовательностях передаваемых кодов. Встречаются кабели Centronics, у которых отсутствует связь контакта 17 разъема PC с контактом 36 разъема принтера. При попытке подключения таким кабелем принтера, работающего в стандарте 1284, появится сообщение о необходимости применения «двунаправленного кабеля». Принтер не может сообщить системе о поддержке расширенных режимов, на что рассчитывает драйвер принтера. Другое проявление отсутствующей связи — «зависание» принтера по окончании печати задания из Windows. Эту связь можно организовать подпайкой дополнительного провода или же просто заменить кабель.

Неплохие электрические свойства имеют ленточные кабели, у которых сигнальные цепи (управляющих сигналов) чередуются с общими проводами. Но их применение в качестве внешнего интерфейса непрактично (нет второго защитного слоя изоляции, высокая уязвимость) и неэстетично (круглые кабели смотрятся лучше). Идеальным вариантом являются кабели, в которых все сигнальные линии перебиты с общими проводами и заключены в общий экран — то, что требует IEEE 1248. Такие кабели гарантированно работают на скоростях до 2 Мбайт/с при длине до 10 м.

В табл. 8.4 приводится распылка *кабеля подключения принтера* с разъемом X1 типа А (DB25-P) со стороны РС и X2 типа В (*Centronics-III* или типа С (миниатюрный) со стороны принтера. Использование общих проводов (GND) зависит от качества кабеля (см. выше). В простейшем случае (18-проводный кабель) все сигналы GND объединяются в один провод. Качественные кабели требуют отдельного обратного провода для каждой сигнальной линии, однако в разъемах типа А и В для этого недостаточно контактов (в табл. 8.4 в скобках указаны номера контактов разъема РС типа А, которым соответствуют обратные провода). В разьеме типа С обратный провод (GND) имеется для каждой сигнальной цепи; сигнальным контактам 1-17 этого разъема соответствуют контакты GND 19-35.

**Таблица 8.4.** Кабель подключения принтера

X1, разъем РС типа А	Сигнал	X2, разъем PRN типа В	X2, разъем PRN типа С
1	Strobe»	1	15
2	DataO	2	6
3	Data1	3	7
4	Data2	4	8
5	DataS	5	9
6	Data4	6	10
7	DataS	7	11
8	Data6	8	12
9	Data7	9	13
10	Ack#	10	3
11	Busy	11	1
12	PaperEnd	12	5
13	Select	13	2
14	Auto LF#	14	17
15	Error#	32	4
16	Init#	31	14
17	Select In#	36	16
18	GND(1)	19	33
19	GND (2 3)	20 21	24 25
20	GND (4 5)	22 23	26 27
21	GND (6 7)	24 25	28 29
22	GND (8 9)	26 27	30 31
23	GND(11 15)	29	19 22
24	GND(101213)	28	20 21 23
25	GND(141617)	30	32 34 35

Ряд отечественных (и стран бывшего СЭВ) принтеров имеет интерфейс *IPNP* (IFSP в документации на принтеры ROBOTRON). Он является близким родственником интерфейса Centronics, а отличия перечислены ниже.

- ◆ Линии данных инвертированы.
- ◆ Протокол квитирования несколько иной.

- ◆ Ко всем входным линиям (на принтере) подключены пары согласующих резисторов: 220 Ом к питанию +5 В и 330 Ом к общему проводу. Это позволяет использовать длинные кабели, но перегружает большинство интерфейсных адаптеров PC.
- ◆ Сигналы ошибки и конца бумаги отсутствуют.

Интерфейс IPNP может быть программно реализован через обычный LPT-порт, но для устранения перегрузки выходных линий согласующие резисторы из принтера желательно удалить. Порт, перегруженный по выходу, может преподносить всякого рода сюрпризы (естественно, неприятные и трудно диагностируемые).

### 8.3.2. Последовательные интерфейсы

Из последовательных интерфейсов в принтерах чаще всего используется RS-232C для подключения к COM-порту. Встречаются принтеры с последовательными интерфейсами «токовая петля» или RS-422, которые подключаются к COM-порту через специальные переходники. Принтеры работают всегда по асинхронному протоколу передачи и, как правило, позволяют настраивать конфигурацию последовательного интерфейса. Задается частота передачи, формат посылки (число информационных, старт- и стоп-битов, контроль паритета) и протокол управления потоком: программный XON/XOFF или аппаратный RTS/CTS. Подключение принтеров и плоттеров к COM-порту требует применения кабеля, соответствующего выбранному протоколу, схемы кабелей приведены на рис. 8.6 и 8.7. Аппаратный протокол предпочтительнее — стандартный драйвер COM-порта пользуется именно им. Естественно, параметры интерфейса принтера должны соответствовать параметрам, заданным для задействованного COM-порта. Порт конфигурируется, например, DOS-командой MODE. Заметим, что при печати средствами DOS (командами COPY или PRINT) прерывания от порта не используются.

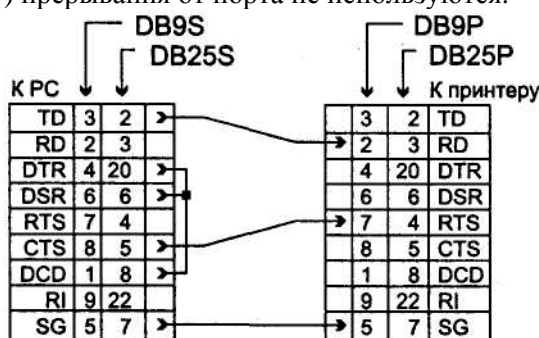


Рис. 8.6. Кабель подключения принтера с протоколом RTS-CTS

Если принтер имеет интерфейс «токовая петля», то для него потребуется преобразователь сигналов, простейшая схема которого приведена на рис. 8.8. Здесь принтер подключается по токовой петле к COM-порту с аппаратным управлени-

ем потоком. Для получения двуполярного сигнала, требуемого для входных сигналов СОМ-порта, применяется питание от интерфейса.

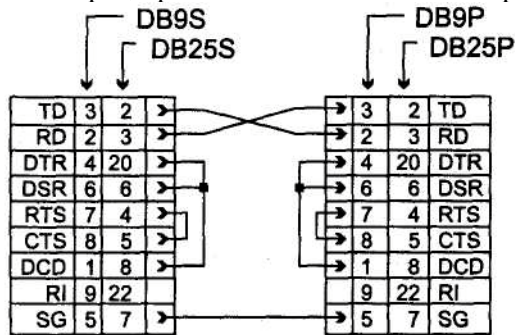


Рис. 8.7. Кабель подключения принтера по протоколу XON/XOFF

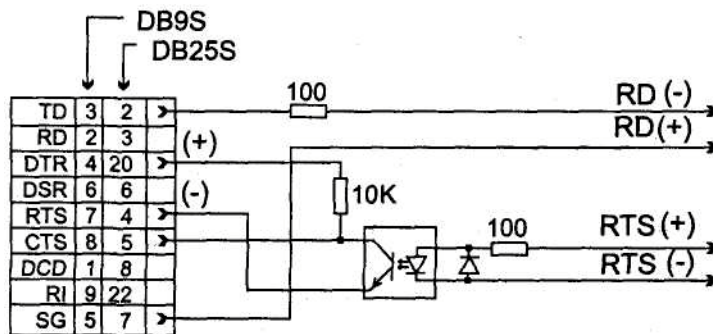


Рис. 8.8. Подключение принтера с интерфейсом «токовая петля 20 мА» к СОМ-порту

### 8.3.3. Системная поддержка принтера

*Вывод на принтер* через порт LPT в стандартном режиме (SPP) по интерфейсу Centronics имеет поддержку на уровне BIOS. Поддержка всех других режимов работы порта (Fast Centronics, ECP) осуществляется только дополнительными драйверами или средствами ОС. Сервисы BIOS Int 17h обеспечивают инициализацию, вывод байта данных и опрос состояния принтера. Перехват прерывания Int 17h является удобным способом внедрения собственных драйверов принтера. Потребность в них может возникать при подключении к порту принтера с интерфейсом *IPNP* или необходимости перекодировки символов.

*Печать содержимого экрана* (Print Screen) поддерживается прерыванием BIOS Int 0 5 h. Обработчик этого прерывания посимвольно выводит содержимое видеопамати (в текстовом режиме) на порт LPT1. Обработчик пользуется ячейкой 0050:0000 для отражения своего текущего состояния: 00 — неактивен, 01 — выполняется печать, FF — во время последнего вызова произошла ошибка ввода-вывода. Прерыв-

вание Int 05h вызывается обработчиком аппаратного прерывания от клавиатуры (Int 09h), когда обнаруживается нажатие клавиши PrintScreen (PrtSc).

## 8.4. Интерфейсы графических адаптеров

Для подключения дисплея (монитора) к графическому адаптеру компьютера используются специализированные/интерфейсы, по которым передается информация о мгновенном значении яркости базисных цветов (RGB) и сигналы строчной и кадровой синхронизации. Способ передачи уже прошел первый виток спирали развития: от дискретного интерфейса первых адаптеров (MDA, CGA, EGA) через аналоговый интерфейс VGA снова возвращаются к цифровому способу (DVI, P&D, DFP). Интерфейсы мониторов в большинстве своем стандартизованы организацией VESA ([www.vesa.org](http://www.vesa.org)).

Видеоинтерфейсы используются для вывода информации на обычные телеприемники и телевизионные мониторы, а также ввода видеоданных в компьютер. Видеоданные в цифровом виде могут передаваться и приниматься по шине Fire Wire, а также по USB версии 2.0.

Многие графические адаптеры имеют внутренний разъем VFC или VAFC — это параллельная шина для обмена пиксельной информацией с дополнительными картами видеообработки.

К системе (процессору и памяти) графический адаптер подключается через какую-либо шину расширения (AGP, PCI, ISA), их интерфейсы описаны в главе 6.

### 8.4.1. Дискретный интерфейс RGB TTL

Дискретный интерфейс с уровнями TTL — *RGB TTL* применялся в мониторах для графических адаптеров MDA, HGC (Hercules), CGA и EGA. Для этого интерфейса требуется разъем DB-9 (розетка на адаптере), назначение контактов приведено в табл. 8.5. В монохромных мониторах используются лишь два сигнала — Video (включить/выключить луч) и Intens (повышенная яркость). В цветных мониторах класса *CD (ColorDisplay)* для адаптеров CGA используется по одному сигналу для включения каждого луча и общий сигнал повышенной яркости. Таким образом можно было задать 16 цветов. В улучшенном цветном дисплее *ECD (Enhanced Color Display)* для адаптера EGA требуются два сигнала на каждый базисный цвет: RED, GREEN, BLUE и Red, Green, Blue — соответственно старшие и младшие биты базисных цветов. Таким образом можно задавать 64 цвета.

Строчная и кадровая синхронизация монитора осуществляется сигналами H.Sync и V.Sync. Монохромные адаптеры MDA и HGC, работающие с высоким разрешением (720x350 пикселей), используют высокую частоту развертки. Адаптер CGA работает с низкими частотами (параметры синхронизации близки к телевизионным). Адаптеры и мониторы EGA могут работать с любыми из этих частот. Для облегчения переключения режимов генератора развертки монитора задействуют сигнал V.Sync: полярность импульсов определяет диапазон частот развертки текущего видеорежима.

Таблица 8.5. Дискретный интерфейс монитора (RGB TTL)

Контакт	Монитор		
	Mono	Color	Enhanced Color/Mono
1	GND	GND	GND
2	GND	GND	Red
3	-	RED	RED
4	-	GREEN	GREEN
5	-	BLUE	BLUE
6	Intens.	Intensiv.	Green/Intens.
7	Video	Резерв	Blue/Video
8	+H.Sync.	+H.Sync.	+H.Sync.
9	-V.Sync.	-V.Sync.	-(+)V.Sync.

## 8.4.2. Аналоговые интерфейсы RGB

Интерфейс *RGB Analog* с аналоговой передачей сигналов яркости базисных цветов позволяет передавать формально неограниченное число оттенков. Сигналы базисных цветов в современных адаптерах формируются 8-разрядными ЦАП, что позволяет выводить 16,7 миллионов цветов (*True Color*). Для уменьшения перекрестных помех эти сигналы передаются по витым парам, с собственными обратными линиями (Return). Для согласования с кабелем в мониторе каждая сигнальная пара нагружается резистором. Черному цвету соответствует нулевой потенциал на линиях всех цветов, полной яркости каждого цвета соответствует уровень +0,7 В (не все графические адаптеры обеспечивают полную амплитуду сигнала). Сигналы управления, состояния и синхронизации передаются сигналами ТТЛ. Временные диаграммы интерфейса RGB (они применимы и к интерфейсу RGB TTL) иллюстрирует рис. 8.9. Сигналы R, G, B, здесь показаны условно — изображены интервалы, во время которых сигналы отображаются засветкой точек экрана (видимая часть изображения — в областях пересечения отображения по кадру и по строке, в остальное время луч принудительно гасится). На рисунке показаны основные временные параметры сигналов. Стандарт VESA DMT (Discrete Monitor Timing, 1994-1998 гг.) задает дискретный ряд вариантов параметров для различных режимов разрешения. Несколько более поздний стандарт VESA GTF (Generalized Timing Formula Standard) задает формулы для определения всех параметров синхронизации, исходными данными для расчета являются следующие:

- ◆ формат экрана в пикселях (например, 800x600);
- ◆ необходимость дополнительного видимого обрамления (overscan borders);
- ◆ тип развертки — построчная (non-interlaced) или черезстрочная (interlaced);
- ◆ одна из заданных частот: кадров, строк или пикселей.

Поскольку стандартов много, один и тот же набор этих параметров разными графическими картами и их драйверами может использовать несколько отличающи-

еся временные параметры сигнала. Эти вариации приходится компенсировать настройками монитора (размер и смещение по вертикали и горизонтали). Примеры параметров синхронизации для построчной развертки приведены в табл. 8.6. Обратим внимание, что для строчной развертки параметры синхронизации задаются в микросекундах, а для кадровой — в числе строк за это время.

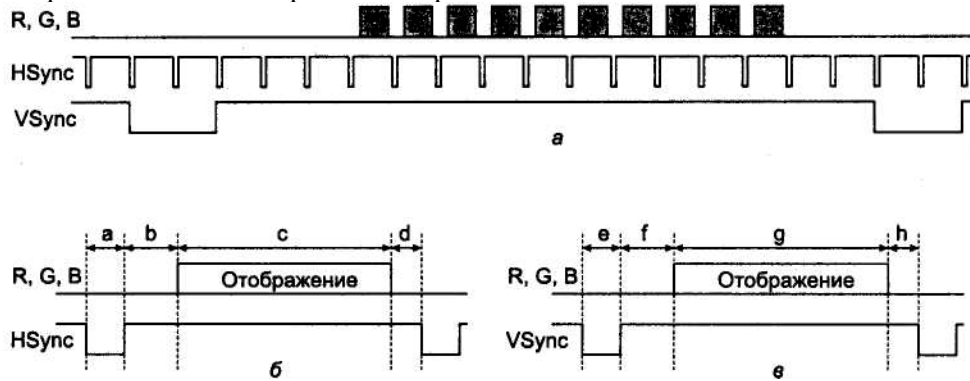


Рис. 8.9. Временные диаграммы интерфейса RGB: а — общая картина, б — строчная развертка, в — кадровая развертка

Таблица 8.6. Параметры синхронизации

Режим VESA	Частота кадров, Гц Fv	Частота строк, кГц Fh	Частота пикселей, МГц Fp	Строчная развертка, мкс				Кадровая развертка, строк			
				A	b	c	d	e	f	g	h
640×480 VGA (60Hz)	59.9	31.5	25.2	3.8	1.9	25.4	0.6	2	33	480	10
640×480 VGA (72Hz)	72.8	37.9	31.5	1.3	3.8	20.3	1.0	3	28	480	9
640×480 VGA (85Hz)	85.0	43.3	35.9	1.6	2.2	17.8	1.6	3	25	480	1
800×600 SVGA (60Hz)	60.3	37.9	40.0	3.2	2.2	20.0	1.0	4	23	600	1
800×600 SVGA (85Hz)	85.1	53.7	56.3	1.1	2.7	14.2	0.6	3	27	600	1
1024×768 XGA (60Hz)	60.0	48.4	64.8	2.1	2.5	15.8	0.4	6	29	768	3
1024×768 XGA (85Hz)	85.0	68.7	94.8	1.0	2.2	10.8	0.5	3	36	768	1
1152×864 (85Hz)	85.0	77.5									
1280×1024 SXGA (60Hz)	60.0	64.0	107.6	1.0	2.3	11.9	0.4	3	38	1024	1

Режим VESA	Частота кадров, Гц Fv	Частота строк, кГц Fh	Частота пикселей, МГц Fp	Строчная развертка, мкс Abcdefgh	Кадровая развертка, строк
1280x1024 SXGA(75Hz)	75.0	79.9	135		
1280x1024 SXGA(85Hz)	85.0	91.1			
1600x1024 (60Hz)	60.0	63.6			
1600x1024 (85Hz)	85.0	91.4			
1600x1200 (85Hz)	85.0	106.3			
1920x1200 (60Hz)	60.0	74.5			
1920x1200 (85Hz)	85.0	107.1			
2048x1536 (75Hz)	75.0	120.2			
2304x1440 (80Hz)	80.0	120.6			

Впервые аналоговый интерфейс был применен на адаптере PGA фирмы IBM, где для него использовался 9-контактный разъем DB-9S (табл. 8.7). В дальнейшем, начиная с адаптеров VGA, стали применять малогабаритный 15-контактный разъем с таким же внешним размером (табл. 8.8). По назначению сигналов эти интерфейсы в основном совпадают, существуют даже переходные кабели с 15- на 9-контактные разъемы (табл. 8.9). В адаптере PGA используется *совмещенная синхронизация (Composite Sync)* сигналом (H+V)Sync.; этот режим поддерживают и многие современные мониторы.

Таблица 8.7. Аналоговый интерфейс монитора PGA (разъем DB-9S)

Контакт	Сигнал
1	Red
2	Green
3	Blue
4	(H+V)Sync
5	Mode Control
6	Red Return
7	Green Return
8	Blue Return
9	GND



Контакт DB-15	Видеоадаптер	Монитор	
	MCGA/VGA/SVGA/XGA	Mono	Color
1	Red	–	Red
2	Green	Video	Green
3	Blue	–	Blue
4	ID2	–	–
5	GND/DDC Return <sup>1</sup>	SelfTest/DDC Return	SelfTest/DDC Return
6	Red Return	Key	Red Return
7	Green Return	Video Return	Green Return
8	Blue Return	–	Blue Return
9	Ключ (нет контакта) <sup>1</sup>	– <sup>1</sup>	– <sup>1</sup>
10	GND (Sync Return)	GND (Sync Return)	GND (Sync Return)
11	ID0	–	GND
12	ID1/SDA <sup>1</sup>	– /SDA <sup>1</sup>	GND/SDA <sup>1</sup>
13	H.Sync/(H+V)Sync <sup>2</sup>	H.Sync/(H+V)Sync <sup>2</sup>	H.Sync/(H+V)Sync <sup>2</sup>
14	V.Sync	V.Sync	V.Sync
15	ID3/SCL <sup>1</sup>	ID3/SCL <sup>1</sup>	ID3/SCL <sup>1</sup>

<sup>1</sup> Сигналы DDC Return, SDA и SCL задействуются только при поддержке DDC. При этом контакт 9 может использоваться для питания логики DDC (+5 В).

<sup>2</sup> Сигнал (H+V)Sync используется при совмещенной синхронизации (Composite

Sync).

**Таблица 8.9.** Переходник 9-15 аналогового интерфейса монитора

Контакт DB9	Сигнал	Контактов 15
1	Red	1
2	Green	2
3	Blue	3
4	H.Sync	13
5	V.Sync	14
6	Red Return	6
7	Green Return	7
8	Blue Return	8
9	GND	10,11

Несмотря (в прямом смысле) на наличие ключа — D-образного кожуха, 15-контактные разъемы ухитряются вставляться в перевернутом положении, при этом один из контактов среднего ряда подгибается, а потом и ломается (штырьки этих разъемов тоньше и слабее, чем у 9-контактных). Естественно, монитор, подключенный таким образом, работать не будет.

В компьютерах Macintosh монитор, совместимый по параметрам с VGA, имеет разъем DB-15P (такой же, как и у Game-порта PC). Назначение его выводов приведено в табл. 8.10.

Таблица 8.10. Разъем VGA Macintosh

Контакт	Сигнал
1	Red Return
2	Red
3	Comp.Sync
4	IDO
5	Green
6	Green Return
7	ID1
8	Не используется
9	Blue
10	ID2
11	Sync. GND
12	V.Sync
13	Blue Return
14	H.SyncGND
15	H.Sync

Кроме передачи изображения, по интерфейсу передают информацию, необходимую для автоматизации согласования параметров и режимов монитора и компьютера. «Интересы» компьютера представляет дисплейный адаптер, к которому и подключается монитор. С его помощью обеспечиваются идентификация монитора, необходимая для поддержки PnP, и управление энергопотреблением монитора.

Для простейшей *идентификации* в интерфейс ввели четыре логических сигнала ID0-ID3, по которым адаптер мог определить тип подключенного монитора IBM. Со стороны монитора эти линии либо подключались к шине GND, либо оставались неподключенными. Однако из этой системы идентификации использовали лишь сигнал ID1, по которому определяют факт подключения монохромного монитора. Монохромный монитор может быть опознан адаптером и иначе — по отсутствию нагрузки на линиях Red и Blue.

Параллельную идентификацию мониторов заменила последовательная по каналу цифрового интерфейса *VESA DDC* (Display Data Channel). Этот канал построен на интерфейсах PC (DDC2B) или *ACCESS.Bus* (DDC2AB), которые требуют всего два ТТЛ-сигнала SCL и SDA. Интерфейс *DDC1* является однонаправленным — монитор посылает адаптеру блок своих параметров по линии SDA (контакт 12), которые синхронизируются сигналом V.Sync (контакт 14). На время приема блока параметров адаптер может повысить частоту V.Sync до 25 кГц (генератор кадровой развертки по такой высокой частоте синхронизироваться не будет). Интерфейс *DDC2* является двунаправленным; для синхронизации используется выделенный сигнал SCL (контакт 15). Интерфейс *DDC2AB* отличается тем, что допускает подключение ПУ, не требующих высокой скорости обмена, к компьютеру по последовательной шине *ACCESS.Bus* (см. п. 11.1.2).

Блок параметров расширенной идентификации дисплея *EDID* (Extended Display Identification) имеет одну и ту же структуру для любой реализации DDC (табл. 8.11).

Таблица 8.11. Блок расширенной идентификации EDID

Смещение, байт	Длина, байт	Назначение
0	8	Заголовок (индикатор начала потока EDID)
8	10	Идентификатор изделия (назначается производителем)
18	2	Версия EDID
20	15	Основные параметры и возможности дисплея
35	19	Установленные параметры синхронизации
54	72	Дескрипторы параметров синхронизации (байты 4-18)
126	1	Флаг расширения
127 г	1	Контрольная сумма

Для управления энергопотреблением монитора в соответствии со стандартом *VESA DPMS* (Display Power Management Signaling) используются сигналы кадровой и строчной синхронизации *V.Sync* и *H.Sync* (табл. 8.12).

Таблица 8.12. Управление энергопотреблением монитора (VESA DPMS)

Режим	H.Sync	V.Sync
On	Активен	Активен
Standby	Неактивен	Активен
Suspend	Активен	Неактивен
Off	Неактивен	Неактивен

Разъемы, применяемые в современных адаптерах и мониторах *SVGA*, не предназначены для передачи высокочастотных сигналов. Пределом для них является примерно 150 МГц, что для высокого разрешения и высокой частоты регенерации недостаточно. Поэтому на больших профессиональных мониторах с высокими разрешениями и частотами синхронизации и на соответствующих адаптерах имеются *BNC-разъемы* для соединения с помощью коаксиальных кабелей. Мониторы с коаксиальными входами могут быть подключены к адаптерам с разъемом *DB-15*, для чего выпускаются специальные переходные кабели. У этих кабелей может быть 3-5 75-омных коаксиальных разъемов *BNC*:

- ◆ 3 разъема — сигналы базисных цветов, смешанная синхронизация (composite sync) передается в канале зеленого цвета;
- ◆ 4 разъема — смешанная синхронизация передается по отдельному кабелю;
- ◆ 5 разъемов — вертикальная и горизонтальная синхронизация передается по раздельным кабелям.

С помощью коаксиальных кабелей возможно удаление монитора от компьютера на расстояние до 10-15 м при хорошем изображении.

Для расширения частотного диапазона (и учитывая тенденцию к использованию последовательных шин USB и Fire Wire) для подключения ПУ к системному блоку компьютера VESA в 1995 г. предложила новый тип разъема *EVC* (Enhanced Video Connector). В 1998 г. была принята новая редакция, и разъем переименован в P&D-A (Plug&Display-Analog) с небольшими изменениями, касающимися резервных контактов и цепей питания зарядного устройства. Кроме обычного аналогового интерфейса RGB и канала DDC2, разъем P&D-A (EVC) имеет контакты для видеовхода, входные и выходные стереоаудиосигналы, шины USB и Fire Wire, а также линии питания постоянного тока для зарядки аккумуляторов портативных ПК. Разъем имеет две секции: высокочастотную для присоединения четырех коаксиальных кабелей и низкочастотную на 30 контактов (рис. 8.10, табл. 8.13). Контакты высокочастотной секции, хотя и не являются коаксиальными, позволяют передавать сигналы с частотами до 2 ГГц. Контакт экранов является крестообразной перегородкой. При использовании 75-омных коаксиальных кабелей на частоте 500 МГц гарантируется уровень отражений и перекрестных помех не выше 2 %. Высокочастотная секция — контакты C1-C4 и C5 (экран) — требуется для передачи цветных сигналов R, G, B и синхросигнала пикселей PX Clock. Синхросигнал пикселей «интересен» матричным дисплеям (с их цифровой природой), его использование позволяет уменьшить погрешности передачи видеоинформации. Частота этого сигнала равна либо частоте сканирования пикселей, либо ее половине (на высокой частоте нужна двойная синхронизация, по фронту и спаду, что уравнивает требования к полосе пропускания для линий цветных данных и линии синхронизации пикселей).

Таблица 8.13. Разъем P&amp;D-A (EVC)

Контакт	Цепь	Контакт	Цепь	Контакт	Цепь
1	Audio Output, Right	11	Charging power input, +	21	Audio input, left
2	Audio Output, Left	12	Charging power input, -	22	Audio input, right
3	Audio Output, Return	13	Video input, Y или composite in	23	Audio input, return
4	Sync Return	14	Video input, return	24	Stereo sync (TTL)
5	Horizontal Sync (TTL)	15	Video input, C in	25	DDC return
6	Vertical Sync (TTL)	16	USB Data +	26	DDC Data (SDA)
7	Резерв	17	USB Data -	27	DDC Clock (SCL)
8	Резерв	18	USB/1394 common mode shield	28	+5 В
9	1394TPA-	19	1394VG	29	1394TPB+
10	1394TPA+	20	1394VP	30	1394TPB-
C1	R (аналог.)			C3	PX Clock
C2	G (аналог.)	C5	GND (для R, G, B)	C4	B (аналог.)

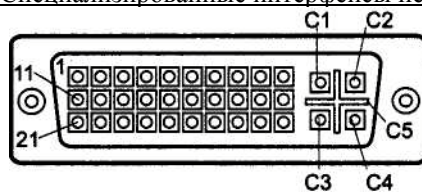


Рис. 8.10. Разъем Б/С и P&amp;D (розетка)

Разъем поделен на компактные зоны для каждой группы сигналов, правда, шины USB и 1394 используют общий контакт для экрана. Назначение контактов видеовхода (S-Video или композитный, PAL или NTSC) может программироваться по каналу DDC2.

Стандарт определяет три уровня реализации: базовый, мультимедийный и полный. Базовый включает только видеосигналы и DDC, в мультимедийном должны быть аудиосигналы. При использовании коннектора в полном объеме монитор превращается в коммутационный центр, который соединяется с компьютером одним кабелем, а все остальные ПУ (включая клавиатуру, мышь, принтер) подключаются к монитору. Разъем может использоваться для подключения портативного ПК к док-станции. EVC собирает сигналы от разных подсистем — графической, видео, аудио, последовательных шин и питания. Этот общий разъем, устанавливаемый на корпусе системного блока, может соединяться с разными платами внутренними кабелями через промежуточные разъемы. Этот разъем не следует путать с похожим по виду и названию разъемом P&D-A/D, описанным в следующем пункте. Разъемы EVC на компьютерах встречаются нечасто, и это объясняется не только их довольно высокой ценой. Устанавливать EVC на графическую карту неудобно (она «обрастет» лишними интерфейсными шлейфами), а интегрированные системные платы редко имеют графические адаптеры с выдающимися параметрами, для которых он нужен.

#### 8.4.4. Цифровые интерфейсы P&D, DVI и DFP

Повсеместный переход на цифровые технологии коснулся и видеомониторов. Традиционный аналоговый канал передачи видеосигналов стал узким местом видеосистемы. По пути от ЦАП к входам видеоусилителей монитора сигнал проходит через пару разъемов и кабель. Несогласованность элементов, вызывающая отражения сигналов («звон») и неравномерности частотных характеристик, приводит к искажению формы сигналов цветов, что становится особо заметным на режимах с высоким разрешением и высокой частотой регенерации. Повысить качество изображения можно, перенеся устройства ЦАП в монитор, прямо на плату видеоусилителей, и подав на них цифровые сигналы базисных цветов. Плоские дисплеи (матрицы TFT) строятся на основе цифровых технологий, и им приходится входные аналоговые сигналы преобразовывать обратно в цифровую форму. Все эти причины привели к необходимости разработки цифрового интерфейса для передачи информации в монитор. От этого интерфейса требуется огромная пропускная способность: к примеру, при частоте пикселей 150 МГц и кодирова-

нии каждого пиксела 24-битным числом (True Color) требуется пропускная способность 3,6 Гбит/с (450 Мбайт/с).

Для подключения плоских дисплеев был разработан специальный интерфейс Panel-Link, в 1996 г. его спецификация (FPDI-2) была утверждена VESA. Схема интерфейса приведена на рис. 8.11. Цифровой интерфейс имеет 3 канала передачи данных (Data[0:3]) и канал синхронизации Clock. В каналах используется дифференциальная передача сигналов с минимизацией переходов — так называемый протокол T.M.D.S. (Transition Minimized Differential Signaling). Каждый канал данных образован кодером, расположенным на видеокарте, линией связи и декодером, расположенным в дисплее. На вход кодера каждого канала поступают 8 бит кода яркости базисного цвета текущего пиксела. Кроме того, на вход кодера канала 0 поступают сигналы строчной и кадровой синхронизации, а на остальные каналы — дополнительные управляющие сигналы CTL[0:3], по паре на каждый канал. Кодеры преобразуют данные в последовательный код, для минимизации переключений 8 входных бит кодируются 10-битным символом, передаваемым по каналу последовательно. В зависимости от входного сигнала разрешения данных DE кодеры передают либо данные цветовых каналов, либо синхросигналы и управляющие биты. На приемной стороне сигналы декодируются и восстанавливаются в том же виде, в котором они поступали на входы кодеров. Частота пикселей может достигать 165 МГц, интерфейс обеспечивает максимальное разрешение 1280x1024 (24 бита на пиксел).

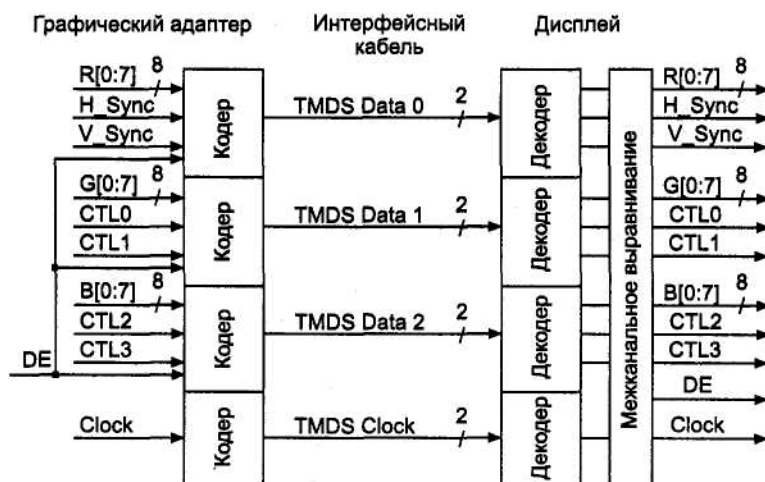


Рис. 8.11. Схема цифрового интерфейса

Физические линии реализованы экранированными витыми парами. Передатчики являются дифференциальными коммутируемыми источниками тока (12 мА), входы дифференциальных приемников подтянуты нагрузочными резисторами 50 Ом к уровню питания +3,3 В, амплитуда сигнала 500 мВ. Выбранный метод кодирования пригоден и для передачи по оптоволоконному кабелю (сигнал не

имеет постоянной составляющей), но пока спецификация определяет только электрический интерфейс.

Вышеописанный протокол используется в интерфейсах P&D, DVI и DFP, из которых наибольшее распространение получили DVI (как самый мощный и универсальный) и DFP (как самый дешевый специализированный). Разъемы этих интерфейсов можно встретить на многих графических адаптерах с двумя выходами. Почти не прижившийся дорогой P&D можно рассматривать как комбинацию усеченного EVC с усеченным DVI. Благодаря использованию стандартизованных сигналов (T.M.D.S.) при несовпадении разъема монитора и графической карты возможно применение пассивных переходников-адаптеров.

В интерфейсе VP&D (VESA Plug-and-Display, 1997 г.), он же P&D, используется такой же разъем, как в EVC (см. рис. 8.10). Здесь нет цепей аналоговых аудиосигналов и видеовхода, а контакты, требовавшиеся для них, теперь назначены на цифровые каналы передачи сигналов. Интерфейс существует в двух вариантах: комбинированном и чисто цифровом. На комбинированный разъем P&D-A/D (табл. 8.14) выведены и аналоговые сигналы (RGB и синхронизация), что обеспечивает возможность подключения как цифрового, так и традиционного аналогового монитора. В чисто цифровом варианте P&D контактов аналоговых сигналов нет; монитор с аналоговым входом (с разъемом EVC или P&D-A) с ним работать не может (конструкция разъема и не позволит его подключить). Точно так же не удастся подключить и монитор с чисто цифровым входом P&D к выходу P&D-A (EVC).

Таблица 8.14. Разъем P&D-A/D

Контакт	Цепь	Контакт	Цепь	Контакт	Цепь
1	Data 2+	11	Data1 +	21	Data 0-
2	Data 2-	12	Data1-	22	Data 0+
3	Экран2	13	Экран 1	23	Экран 0
4	Sync Rtn	14	Clock+	24	Stereo Sync TTL
5	H.SyncTTL	15	Clock-	25	DDC Return
6	V.SyncTTL	16	USB Data+	26	DDC Data
7	Экран Clock	17	USB Data-	27	DDC Clock
8	CHRG+	18	13943КраН/CHRG-	28	+5V
9	1394TPA-	19	1394VG	29	1394TPB+.CLOCK+
10	1394TPA+	20	1394VP	30	1394TPB-.CLOCK-
C1	R (аналог.)			C3	PX Clock
C2	G (аналог.)	C5	GND (для R, G, B)	C4	B (аналог.)

Интерфейс плоских дисплеев DFP (Digital Flat Panel, 1999 г., [www.dfp-group.org](http://www.dfp-group.org)) использует дешевый разъем типа MDR (mini-D ribbon) с ленточными контактами (рис. 8.12), на который выведены лишь 3 пары сигналов для цифровых каналов данных, пара для цифрового канала синхронизации, питание (+5В), канал DDC2 (табл. 8.15) и сигнал обнаружения «горячего» подключения (HPD). Ча-

стога пикселей может достигать 85 МГц (для плоских панелей не требуется слишком высокая частота развертки). Интерфейс пригоден (пока?) для режимов вплоть до 1280x1024 (24 бита на пиксел).

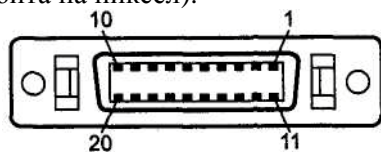


Рис. 8.12. Разъем плоского дисплея DFP

Таблица 8.15. Разъем DFP

Контакт	Цепь	Контакт	Цепь
1	TX1 +	11	TX2+
2	TX1-	12	TX2-
3	SHLD1	13	SHLD2
4	SHLDC	14	SHLDO
5	TXC+	15	TXO+
6	TXC-	16	TXO-
7	GND	17	NC
8	+5V	18	HPD
9	NC	19	DDC_DA
10	NC	20	DDC CLK

*Интерфейс D<sup>v</sup>* (Digital Visual Interface) разработан группой DDWG (Digital Display Working Group — рабочая группа по цифровым дисплеям, [www.ddwg.org](http://www.ddwg.org)) в 1999 г. и предназначен для подключения дисплеев любого типа (ЭЛТ и матричных) к компьютеру, причем возможны два варианта коннекторов и интерфейса: чисто цифровой и цифровой с традиционными аналоговыми сигналами. Во втором случае к разъему DVI через пассивный переходник может быть подключен монитор с обычным аналоговым VGA-интерфейсом.

*Минимальный вариант цифрового интерфейса* содержит канал синхронизации и три канала данных (Data0-2). В таком варианте интерфейс почти ничем не отличается от аналогового — меняется только местоположение ЦАП и применяется цифровой способ доставки данных. При этом гамма-коррекция возлагается на дисплей. Однако интерфейс предусматривает способ повышения пропускной способности за счет более эффективного использования времени. Дело в том, что традиционные ЭЛТ-мониторы имеют довольно значительное время обратного хода луча по строке и кадру, в течение которого пиксели на экран, естественно, не выводятся, — в это время интерфейс простаивает. Для матричных дисплеев этих пауз не требуется, поэтому тот же объем информации о пикселях может передаваться за большее время — практически за весь период кадра. Следовательно,



можно либо снижать тактовую частоту передачи пикселей (не меняя разрешения и частоты развертки), либо с той же (предельно достижимой) частотой передачи увеличить разрешение или (и) частоту развертки. Спецификация DVI предполагает, что возможность передачи данных в течение всего периода кадра может появиться и у цифровых дисплеев, построенных на обычных ЭЛТ, за счет внутренней буферизации. При наличии буферизации экрана в дисплее можно пойти и дальше — вместо непрерывной регенерации экрана, которой озабочены традиционные видеоадаптеры, передавать данные только при изменениях изображения, но это пока лишь возможные перспективы. В полном варианте добавляются еще 3 цифровых канала (Data3-5), информационная нагрузка должна распределяться поровну между парами каналов. Таким образом, четные пиксели будут передаваться по каналам 0 (R), 1 (G) и 2 (B), а нечетные — соответственно по 3,4 и 5, и интерфейс позволит передавать пиксели с частотой до 330 МГц (165х2). Предусматривается и иное использование дополнительных каналов: когда 8 бит на кодирование базисного цвета покажется недостаточным<sup>^</sup>), каналы 3,4 и 5 могут дополнить (как младшие биты) данные каналов 0, 1 и 2 (старшие).

Кроме сигналов T.M.D.S. в интерфейс DVI входят сигналы интерфейса VESA DDC2: DDC Data и DDC Clock, а также линия питания +5 В, по которой от видеокарты питаются цепи DDC, позволяя обмениваться конфигурационной информацией даже с выключенным монитором. Конфигурационная информация позволяет системе определить возможности монитора и должным образом сконфигурировать имеющиеся каналы данных, согласуя возможности и видеокарты, и дисплея. Имеется также сигнал HPD (Hot Plug Detect), с помощью которого система может следить за подключением/отключением дисплея. «Горячее» подключение обеспечивается также и механическими особенностями разъемов, поддерживающих требуемую последовательность соединения/рассоединения разных групп контактов. Таким образом, дисплеи с DVI обеспечивают все необходимые функции для реализации принципов PnP. Интерфейс и дисплеи с DVI должны обеспечивать стандартные (VESA) графические режимы, начиная от 640х480/60 Гц (частота пикселей 22,175 МГц). Его предел - 2048х1536 пикселей (частота 330 МГц). Интерфейс поддерживает сигнализацию управления энергопотреблением (DPMS).

Вид коннекторов DVI приведен на рис. 8.13, расположение сигнальных контактов дано в табл. 8.16.

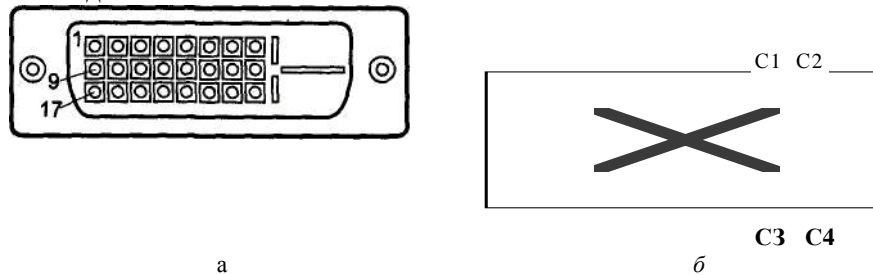


Рис. 8.13. Коннекторы DVI (розетки): а — только цифровой, б — цифровой с аналоговым

Таблица 8.16. Коннектор DVI

Контакт	Цепь	Контакт	Цепь	Контакт	Цепь
1	Data2-	9	Data1-	17	DataO-
2	Data2+	10	Data1 +	18	DataO+
3	Экран 2/4	11	Экран 1/3	19	Экран 0/5
4	Data4-	12	DataS-	20	DataS-
5	Data4+	13	Data3+	21	Data5+
6	DDC Clock	14	+5 В	22	Экран Clock
7	DDC Data	15	GND (для +5 В, HSync и VSync)	23	Clock*
8	VSync(ТТЛ)	16	HPD	24	Clock-
C1	R (аналог.)			C3	в (аналог.)
C2	G (аналог.)	C5	GND (для R, G, B)	C4	HSync (ТТЛ)

### 8.4.5. Внутренние цифровые интерфейсы

Для расширения возможностей дисплейного адаптера, главным образом, в сторону обработки видеоизображений, многие графические адаптеры имеют внутренний интерфейс для передачи пиксельной информации синхронно с регенерацией экрана. Этот интерфейс используется для связи графического адаптера с видеооверлейными платами (видеобластерами), декодерами MPEG. Разъем графического адаптера связывается с таким же разъемом видеоплаты плоским кабелем-шлейфом.

На адаптерах VGA присутствовал краевой 26-контактный разъем *VGA Auxiliary Video Connector* с шагом ламелей 0,4". Впоследствии был стандартизован *VESA Feature Connector* (VFC) (табл. 8.17), у которого назначение сигналов практически сохранилось, но используется двухрядный штырьковый разъем. Этот разъем графического адаптера VGA и SVGA позволяет получать поток байт данных сканируемых пикселей при работе адаптера в режиме до 640x480 пикселейx256 цветов. Нормально интерфейс работает на вывод и синхронизируется от генератора графического адаптера. Однако, установив низкий уровень сигнала Data Enable, видеоплата может заставить графическую карту принимать пиксели; сигнал Sync Enable переключает графический адаптер на прием сигналов строчной и кадровой синхронизации; сигнал PCLK Enable переключает графический адаптер на работу от внешнего сигнала синхронизации пикселей.

Для режимов до 1024x768 с глубиной цвета High Color и True Color предназначен разъем VAFC — *VESA Advanced Feature Connector* (табл. 8.18) — двухрядный, с шагом 0,05" и расстоянием между рядами 0,4". Он имеет разрядность 16/32 бит и при максимальной частоте точек 37,5 МГц обеспечивает скорость потока данных 150 Мбайт/с. 16-битная версия VAFC использует первые 56 контактов, а 32-битная — все 80-контактов разъема. Допустимая длина шлейфа — 7". В этом интерфейсе сигналы GRDY и VRDY означают готовность (способность генерировать данные пикселей) графического адаптера и видеосистемы соответственно, а направлением передачи данных управляет сигнал EVID#.

Таблица 8.17. Разъем VFC

Сигнал	Контакт	Контакт	Сигнал
GND	2	1	DataO
GND	4	3	Data1
GND	6	5	Data 2
Data enable	8	7	DataS
Sync, enable	10	9	Data 4
PCLK enable	12	11	Data S
(Vcc)	14	13	Data 6
GND	16	15	i Data 7
GND	18	17	PCLK
GND	20	19	BLANK
GND	22	21	HSYNC
(Vcc)	24	23	VSYNC
(GND)	26	25	GND

Кроме этих стандартов существует и специальная внутренняя 32-битная шина для обмена данными между мультимедийными устройствами — *VESA Media Channel* (VM Channel). Эта шина (канал), в отличие от вышерассмотренных двухточечных интерфейсов, ориентирована на широковещательную передачу данных между несколькими абонентами.

Таблица 8.18. Разъем VAFC

Контакт	Сигнал	Назначение	Контакт	Сигнал	Назначение
1	RSRVO	Резерв	41	GND	Ground
2	RSRV1	Резерв	42	GND	Ground
3	GENCLK	Genclock input	43	GND	Ground
4	OFFSETO	Pixel offset 2	44	GND	Ground
5	OFFSET1	Pixel offset 1	45	GND	Ground
6	FSTAT	FIFO buffer status	46	GND	Ground
7	VRDY	Video ready	47	GND	Ground
8	GRDY	Graphics ready	48	GND	Ground
9	BLANK*	Blanking	49	GND	Ground
10	VSYNC	Vertical sync	50	GND	Ground
11	HSYNC	Horizontal sync	51	GND	Ground
12	EGEN#	Enable genclock	52	GND	Ground
13	VCLK	Graphics data clock	53	GND	Ground
14	RSRV2	Резерв	54	GND	Ground
15	DCLK(PCLK)	Video data (Pixel) clock	55	GND	Ground
16	EVIDEO*	Video data direction control	56	GND	Ground
17	PO	Video data 0	57	P1	Video data 1
18	GND	Ground	58	P2	Video data 2

Контакт	Сигнал	Назначение	Контакт	Сигнал	Назначение
19	P3	Video data 3	59	GND	Ground
20	P4	Video data 4	60	P5	Video data 5
21	GND	Ground	61	P6	Video data 6
22	P7	Video data 7	62	GND	Ground
23	P8	Video data 8	63	P9	Video data 9
24	GND	Ground	64	P10	Video data 10
25	P11	Video data 11	65	GND	Ground
26	P12	Video data 12	66	P13	Video data 13
27	GND	Ground	67	P14	Video data 14
28	P15	Video data 15	68	GND	Ground
29	P16	Video data 16	69	P17	Video data 17
30	GND	Ground	70	P18	Video data 18
31	P19	Video data 19	71	GND	Ground
32	P20	Video data 20	72	P21	Video data 21
33	GND	Ground	73	P22	Video data 22
34	P23	Video data 23	74	GND	Ground
35	P24	Video data 24	75	P25	Video data 25
36	GND	Ground	76	P26	Video data 26
37	P27	Video data 27	77	GND	Ground
38	P28	Video data 28	78	P29	Video data 29
39	GND	Ground	79	P30	Video data 30
40	P31	Video data 31	80	GND	Ground

### 8.4.6. Видеоинтерфейсы

В традиционной технике цветного телевизионного вещания видеосигнал непосредственно несет информацию о мгновенном значении яркости (в нем присутствуют и синхроимпульсы отрицательной полярности), а цветовая информация передается в модулированном виде на дополнительных частотах. Таким образом обеспечивается совместимость черно-белого приемника, игнорирующего цветовую информацию, с цветным передающим каналом. Однако способ кодирования цветовой информации и частоты разверток в системах PAL, SECAM и NTSC различны. В видеотехнике используют различные низкочастотные интерфейсы (радиочастотный тракт здесь не рассматривается).

В интерфейсе *Composite Video* полный стандартный видеосигнал с размахом около 1,5 В передается по коаксиальному кабелю (75 Ом). Для соединения используются коаксиальные разъемы RCA («колокольчики»). Данный интерфейс характерен для бытовых видеомагнитофонов, аналоговых телекамер, телевизоров. В ПК этот интерфейс используется как дополнительный выходной интерфейс графической карты и как входной интерфейс в устройствах захвата видеосигнала.

Интерфейс *S-Video* (Separate Video) использует отдельные сигнальные линии: Y для канала яркости и синхронизации (luminance+sync, обычный черно-белый

видеосигнал) и С для сигнала цветности. По линии С передается поднесущая частота, модулированная цветоразностными сигналами (burst signal). Сигнал Y имеет размах 1 В, сигнал С в стандарте NTSC имеет размах 0,286 В, в PAL/SECAM — 0,3 В. Обе линии должны нагружаться терминатором 75 Ом. Стандартный 4-контактный разъем S-Video типа mini-DIN (рис. 8.14, а) используется как интерфейс высококачественных видеосистем, его синонимами являются названия S- VHS и Y/C. Этот интерфейс в ПК тоже может использоваться в качестве входного и дополнительного выходного; он обеспечивает более высокое качество передачи видеоизображений. Иногда задействуют и 7-контактные разъемы mini-DIN, у них внешние 4 контакта имеют то же назначение, а 3 внутренних контакта используются для разных целей (там может быть и композитный сигнал). Выход S-Video легко преобразовать в сигнал для композитного входа (рис. 8.14, б); эта схема не обеспечивает должного согласования импедансов, но обеспечивает приемлемое качество изображения. Обратное преобразование этой схемой выполняется гораздо хуже, поскольку на яркостный сигнал будет воздействовать помеха в виде сигнала цветности.

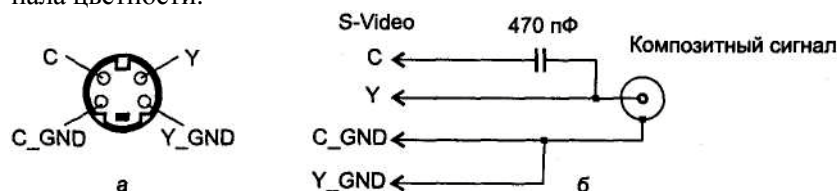


Рис. 8.14. Интерфейс S-Video: а — разъем, б — преобразование в композитный сигнал

Наивысшее качество передачи обеспечивает *профессиональный* (студийный) интерфейс YUV (professional video), использующий три сигнальные линии: здесь цветоразностные сигналы U и V передаются в немодулированном виде.

## 8.5. Интерфейсы аудиоустройств

Звуковая карта имеет набор разъемов для подключения внешних аудиосигналов, аналоговых и цифровых, а также интерфейс MIDI для связи с электронными музыкальными инструментами. Аудиоданные в цифровом виде могут передаваться также и по универсальным шинам USB и Fire Wire (см. п. 4.2).

### 8.5.1. Аналоговые интерфейсы

Аналоговые интерфейсы позволяют подключать стандартную бытовую аппаратуру, микрофон, аналоговый выход CD-ROM. На большинстве карт массового потребления для аналоговых сигналов применяют малогабаритные разъемы — «мини-джеки» (jack) диаметром 3,5 мм, моно и стерео. Эти разъемы универсальны (используются на бытовой аппаратуре), но имеют весьма низкое качество контактов — они являются источником шумов (шорохов и тресков), а также иногда

просто теряют контакт. Их полноразмерные 6-миллиметровые «родственники», характерные для профессиональной аппаратуры, имеют весьма высокое качество, но из-за крупных габаритов на звуковых картах не используются. На некоторых высококачественных картах сигналы линейного входа и выхода выводятся на пары разъемов RCA, которые обеспечивают очень хороший контакт, особенно в позолоченном варианте. В просторечии такие разъемы, часто используемые на бытовых видеомагнитофонах, называют «колокольчиками» или «тюльпанами».

Раскладка цепей на мини-джеках унифицирована: левый канал — на центральном контакте, экран (земля) на внешнем цилиндре, правый канал — на промежуточном цилиндре. Если стереоджек включить в моногнездо и наоборот, сигнал пойдет только по левому каналу. Все соединения в стереосистемах осуществляются «прямыми» кабелями (контакты разъемов соединяются «один в один»). Для подключения центрального и низкочастотного каналов в 6-колоночной системе единого подхода нет — может потребоваться перекрестный кабель. Неправильное подключение будет заметно по «писку» низкочастотной колонки (сабвуфера) и «бубнению» центральной колонки.

Подключение к звуковой карте устройств через внешние разъемы проблем обычно не вызывает — они унифицированы, и достаточно знать назначение разъемов, маркированных на задней панели.

- ◆ *Line In* — линейный вход от магнитофона, тюнера, проигрывателя, синтезатора и т. п. Чувствительность порядка 0,1-0,3 В.
- ◆ *Line Out* — линейный выход сигнала на внешний усилитель или магнитофон, уровень сигнала порядка 0,1-0,3 В.
- ◆ *Speaker Out* — выход на акустические системы или головные телефоны. Подключать к нему внешний усилитель мощности нецелесообразно, поскольку здесь искажения больше, чем на линейном выходе.
- ◆ *Mic In* — микрофонный вход, чувствительность 3-10 мВ. Этот вход обычно монофонический, но иногда используется трехконтактное гнездо (как в стерео), у которого дополнительный контакт (на месте правого канала) выделен для подачи питания на электретный микрофон.

Подключение внутренних устройств к аналоговым входам может доставить больше забот. Для этого используются четырехштырьковые разъемы, различающиеся как шагом между выводами, так и их назначением. Для подключения CD-ROM часто ставят рядом два, а то и три разъема с параллельно соединенными сигнальными контактами, но и это может не помочь, если кабель имеет другое расположение сигналов. Спасти может перестановка контактов на разъеме кабеля, для чего иголкой нажимают на фиксирующий выступ контакта. После этого контакт можно вытянуть в сторону кабеля и переставить в другое гнездо. Вид и варианты расположения сигнальных контактов аудиовходов приведены на рис. 8.15. Для полноты картины добавим, что разъем может иметь ключ с противоположной стороны (по ошибке сборщика кабеля или по внутреннему стандарту его производителя). Задача подключения все-таки не безнадежна, поскольку требует правильной расстановки только двух сигнальных контактов, а контакты общего провода выделя-

ются тем, что на плате соединяются с шиной, а на кабеле — с экраном. Положение левого и правого каналов аудио-CD в большинстве случаев не так уж и важно.

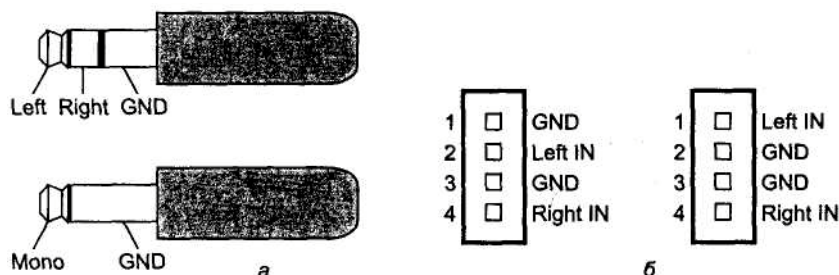


Рис. 8.15. Разъемы подключения аудиосигналов

## 8.5.2. Цифровые интерфейсы

*S/PDIF* (Sony/Philips Digital Interface Format) — цифровой последовательный интерфейс (и форматы данных) для передачи аудиосигналов между блоками бытовой цифровой аудиоаппаратуры (DAT, CD-ROM и т. п.). Этот интерфейс является упрощенным вариантом студийного интерфейса AES/EBU (Audio Engineers Society/European Broadcast Union). Интерфейс AES/EBU использует симметричный двухпроводный экранированный кабель с импедансом 110 Ом, разъемы XLR, уровень сигнала — 3-10 В, длина кабеля — до 12 м.

Интерфейс *S/PDIF* использует коаксиальный кабель 15 Ом, разъемы RCA или BNC, уровень сигнала — 0,5-1 В, длина кабеля — до 2 м. В звуковых картах внутренние разъемы *S/PDIF* проще — это просто пара штырьков (как у джамперов) на плате с соответствующей ответной частью на кабеле. Такие же упрощенные разъемы применяются и на новых приводах CD-ROM, имеющих выход *S/PDIF*. «Штатная» схема передатчика *S/PDIF* содержит разделительный импульсный трансформатор (1:1), благодаря которому соединяемые устройства гальванически развязываются. Встречаются и упрощенные варианты, без разделительного трансформатора. При стыковке устройств с нестандартными интерфейсами возможны проблемы, связанные с несоответствием уровней сигналов. При этом сигнал может быть неустойчивым (звук будет прерываться) или не приниматься совсем. Эти проблемы могут быть решены подручными средствами — установкой дополнительных формирователей сигнала.

Кроме электрической версии существует и оптическая версии интерфейса *S/PDIF* — Toslink, стандарт EIAJ CP-1201 — с инфракрасными излучателями (660 нм). Применение оптики позволяет обеспечить полную гальваническую развязку устройств, что необходимо для снижения уровня наводок. Для пластикового волокна (POF) длина кабеля не более 1,5 м, для стеклянного волокна — 3 м. В Сети предлагается ряд схем преобразования интерфейсов, одна из которых приведена на рис. 8.16. Здесь первый инвертор посредством обратной связи выведен на линейный участок передаточной характеристики, благодаря чему малый входной сигнал вызы-

вает его переключение. В схеме предлагается микросхема НСТ74U04 (6 инверторов); вместо светодиода можно использовать и фирменный трансивер Toslink, его следует подключать без балластного резистора (220 Ом) прямо к выходу инвертора (резистор находится в трансивере).

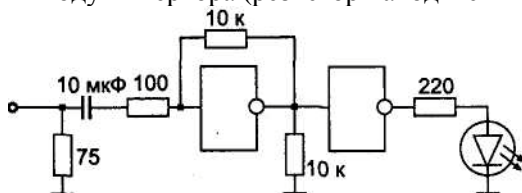


Рис. 8.16. Схема преобразователя электрического интерфейса S/PDIF в оптический (Toslink)

По интерфейсу S/PDIF информация передается в последовательном коде покад-рово, с обеспечением синхронизации и контролем достоверности передачи (кодами Рида-Соломона). В кадре имеется признак формата данных — PCM или не PCM, что позволяет по данному интерфейсу передавать и упакованные цифровые данные (например, MPEG для AC-3). Имеется также бит защиты от копирования, признак предискажений и некоторые другие служебные данные. В режиме PCM выборки каждого канала могут иметь разрядность 16, 20 или 24 бит, частота выборок определяет частоту цифрового сигнала. Приемник S/PDIF сам определяет частоту выборок по принимаемому сигналу, наиболее употребимые частоты - 32, 44,1 и 48 кГц.

Кроме этих интерфейсов в студийной аппаратуре применяют интерфейсы ADAT и TDIF, которые имеются только на дорогих профессиональных звуковых картах.

Для обмена данными с приводами DVD применяется цифровой последовательный интерфейс I2S.

### 8.5.3. Интерфейс MIDI

Цифровой интерфейс музыкальных инструментов *MIDI* (*Musical Instrument Digital Interface*) является последовательным асинхронным интерфейсом с частотой передачи 31,25 Кбит/с. Этот интерфейс, разработанный в 1983 году, стал фактическим стандартом для сопряжения компьютеров, синтезаторов, записывающих и воспроизводящих устройств, микшеров, устройств специальных эффектов и другой электромusicalной техники. В настоящее время интерфейс MIDI имеют и дорогие синтезаторы, и дешевые музыкальные клавиатуры, которые могут использоваться в качестве устройств ввода компьютера. По интерфейсу MIDI устройства обмениваются между собой сообщениями, кратко описанными в книге [1]. На одном интерфейсе может быть организовано до 16 логических каналов, каждый из которых может управлять своим инструментом.

В *физическом интерфейсе* применяется *токовая петля 5 мА* (возможно до 10 мА) с гальванической (оптронной) развязкой входной цепи. Логическому нулю соответствует наличие тока, логической единице (и покою) — отсутствие тока (в «классической» токовой петле телекоммуникаций все наоборот).



Интерфейс определяет три типа портов: *MIDI-In*, *MIDI-Out* и *MIDI-Thru*.

*Входной порт MIDI-In* представляет собой вход интерфейса «токовая петля», гальванически развязанного от приемника оптроном с быстродействием не хуже 2 мкс. Устройство отслеживает информационный поток на этом входе и реагирует на адресованные ему команды и данные.

*Выходной порт MIDI-Out* представляет собой выход источника тока, гальванически связанного со схемой устройства. Ограничительные резисторы предохраняют выходные цепи от повреждения при замыкании на землю или источник 5 В. На выход подается информационный поток от данного устройства. При специальной настройке устройства в этом потоке может содержаться и транслированный входной поток, но это нетипично.

*Транзитный порт MIDI-Thru* служит только для ретрансляции входного потока, по электрическим свойствам он аналогичен выходному. Его наличие не является обязательным для всех устройств.

В качестве разъемов применяются 5-контактные разъемы DIN, распространенные в бытовой звуковой аппаратуре, схема соединительного кабеля приведена на рис. 8.17.

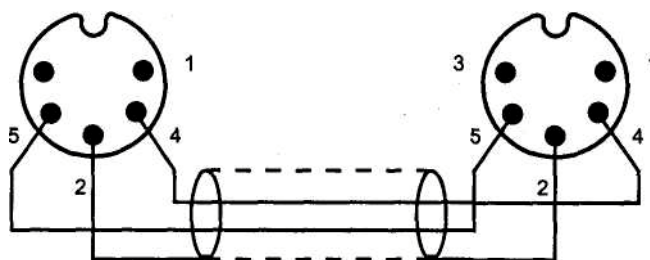


Рис. 8.17. Соединительные кабели MIDI

Внешний порт MIDI (с сигналами ТТЛ) обычно выводится на неиспользуемые контакты (12 и 15) разъема игрового адаптера (DB-15S). При этом для подключения стандартных устройств MIDI требуется *переходной адаптер*, реализующий интерфейс «токовая петля» (на разъеме карты интерфейс ТТЛ). Переходной адаптер обычно встраивается в специальный кабель, вариант схемы которого приведен на рис. 8.18. Некоторые модели PC имеют встроенные адаптеры и стандартные 5-штырьковые разъемы MIDI.

Программно порт MIDI обычно совместим с UART MPU-401. *MPU-401* фирмы Roland — первая карта расширения для PC с интерфейсом MIDI, получившая широкое распространение. MPU расшифровывается как MIDI Processing Unit — устройство обработки сообщений MIDI. Этот контроллер кроме асинхронного последовательного порта (UART), реализующего физический интерфейс MIDI, имел развитые аппаратные средства для использования PC в качестве секвенсора. Контроллер MPU-401 поддерживал простой режим работы — *UART mode*, в котором использовался только двунаправленный асинхронный порт; в современных звуковых картах совместимость с MPU-401 поддерживается только в этом режиме.

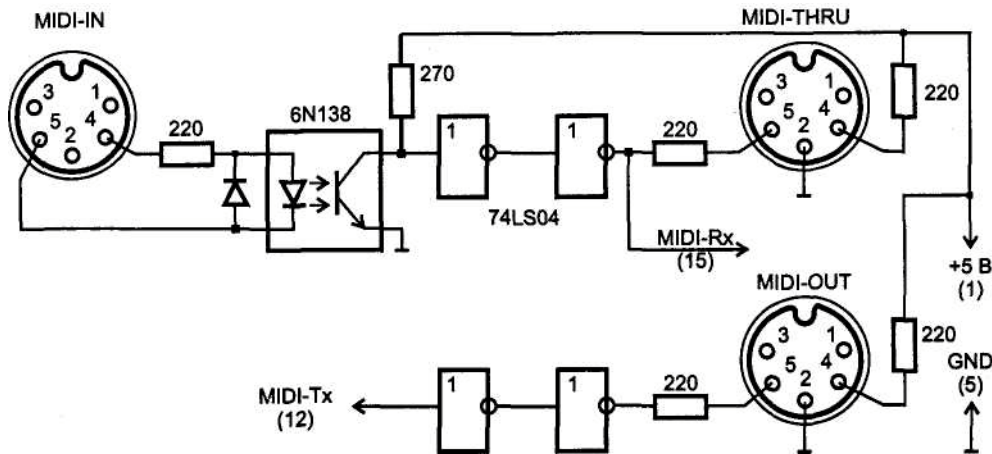


Рис. 8.18. Вариант схемы кабеля-адаптера MIDI

В пространстве ввода-вывода MPU-401 занимает два смежных адреса MPU (обычно 330B) и MPU+1.

- ◆ Порт DATA (адрес MPU+0) — запись и считывание байт, передаваемых и принимаемых по интерфейсу MIDI. В интеллектуальном режиме через этот же порт считываются и вспомогательные данные от MPU (не относящиеся к потоку MIDI).
- ◆ Порт STATUS/COMMAND (адрес MPU+1) — чтение состояния/запись команд (запись — только для интеллектуального режима). В байте состояния определены следующие биты:
  - бит 7 — D5R (Data Set Ready) — готовность (DSR=0) принятых данных для чтения (бит устанавливается в единицу, когда все принятые байты считаны из регистра данных);
  - бит 6 — DRR (Data Read Ready) — готовность (DRR=0) UART к записи в регистр данных или команд (условие готовности к записи не возникнет, если приемник имеет непрочитанный байт данных).

По включении питания «настоящая» карта MPU-401 устанавливается в интеллектуальный режим, из которого в режим UART ее можно перевести командой с кодом 3Fh. Программный сброс MPU-401 (опять-таки в интеллектуальный режим) осуществляется командой RESET (код FFh), на эту команду MPU ответит подтверждением ACK (FEh). Байт подтверждения извлекается из регистра данных, до его прихода следующую команду MPU не воспримет. На команду с кодом 3Fh MPU подтверждением не отвечает (некоторые эмуляторы отвечают и на эту команду).

Ввод данных может осуществляться по программному опросу бита DSR или по прерываниям. Аппаратные прерывания от MPU в режиме UART вырабатываются по приему байта. Обработчик прерывания должен считать все поступившие байты, проверив перед выходом, что DSR=1 (иначе возможны потери принятых байт).

*Вывод данных* разрешается битом DRR, прерывания по готовности вывода не вырабатываются.

*Совместимость с MPU-401*, имеющаяся у большинства современных звуковых карт с интерфейсом MIDI, означает наличие приемопередатчика, программно совместимого с MPU-401 в режиме UART; функции интеллектуального режима обычно не поддерживаются.

На некоторых системных платах применяются БИС контроллеров интерфейсов, в которых режим UART, используемый для COM-порта, конфигурированием через BIOS SETUP может быть переведен в режим MIDI-порта.

Для подключения к компьютеру большого числа устройств MIDI можно использовать шину USB. Для этого, например, фирма Roland выпускает 64-канальный процессорный блок S-MPU64, который кроме шины USB имеет 4 входных и 4 выходных порта MIDI. Программное обеспечение допускает объединение до 4 блоков на одной шине USB, что увеличивает число каналов до 256.

## 8.5.4. Интерфейс дочерней карты

Ряд моделей звуковых карт имеют внутренний интерфейсный разъем подключения дочерней карты с MIDI-синтезатором (Daughterboard Connector). На разъем (табл. 8.19) с основной карты выводится сигнал MIDI-порта (ТТЛ, как и на разъем джойстика) и сигнал аппаратного сброса синтезатора, а с дочерней карты принимается стереофонический аналоговый сигнал, который поступает в микшер основной карты. В шинах питания аналоговая земля (AG) отделена от цифровой (DG). Дополнительно может использоваться и вход MIDI (тоже ТТЛ). Разъем может обозначаться и как WT (Wavetable) Connector, Waveblaster Connector.

Таблица 8.19. Назначение контактов разъема подключения дочерней карты

Контакт	Цепь	Контакт	Цепь
1	DG	2	
3	DG	4	MIDI <sub>1</sub> Dut*
5	DG	6	+5 B
7	DG	8	MIDI_In# (необязательный)
9	DG	10	+5 B
11	DG	12	
13		14	+5 B
15	AG	16	
17	AG	18	+12 B
19	AG	20	Audio (R)
21	AG	22	- 12 B
23	AG	24	Audio (L)
25	AG	26	Reset*

Подключение дочерней карты эквивалентно подключению внешнего синтезатора к MIDI-выходу звуковой карты. Если на звуковой карте отсутствует разъем под-

ключения дочерней карты, то дочернюю карту можно подключить и к внешнему разъему джойстика/MIDI и аналоговым входам звуковой карты. Конечно, на дочернюю карту нужно подать питание, а также сигнал аппаратного сброса.

## 8.6. Интерфейс игровых устройств — Game-порт

Игровые устройства — джойстик, руль и педали автомобиля или иные — вырабатывают некоторые аналоговые и дискретные сигналы, которые можно ввести в компьютер. С самых первых моделей IBM PC был введен и фактически стандартизован интерфейс игрового адаптера — *Game port*, к которому можно подключить до двух джойстиков или иных устройств. Суммарно на порте доступно 4 координатных датчика (X1, X2, Y1 и Y2), изменяющих сопротивление, и 4 дискретных входа для кнопок управления. Назначение координатных датчиков зависит от игры и конструкции манипулятора. Для авиасимуляторов X1 может соответствовать перемещению рукоятки вверх-вниз, Y1 — влево-вправо, X2 — нажатие левой и правой педалей, Y2 — рукоятка сектора газа. Для автомобильных рулей X1 — руль, Y1 — газ, X2 — тормоз (газ и тормоз могут быть совмещены в координате Y1). Кроме игровых целей порт может применяться и для подключения «серьезных» датчиков.

Современные игровые устройства имеют свой интеллект (микроконтроллер) и подключаются к компьютеру цифровым интерфейсом — по шине USB или через COM-порт. Их функциональные возможности богаче, они позволяют устанавливать и двустороннюю связь с игроком (вводить механические воздействия).

*Адаптер Game-порта* имеет в пространстве ввода-вывода один регистр с адресом 201h, биты которого при чтении отображают состояние кнопок и компараторов аналоговых сигналов. Ввод дискретных сигналов от кнопок пояснений не требует. Упрощенная схема одного канала аналогового ввода приведена на рис. 8.19. В начале преобразования конденсатор разряжается через ключ, после чего начинается его заряд, скорость которого определяется величиной сопротивления датчика (чем больше сопротивление, тем медленнее заряд). Напряжение на конденсаторе контролируется компаратором, который срабатывает по достижении определенного уровня. Выходы компараторов всех четырех каналов преобразования, как и дискретные входы, собираются в регистр (см. ниже), который может быть программно считан. Преобразование выполняется чисто программно и начинается по выводу любого байта в регистр адаптера (201h), при этом биты 0-3 устанавливаются в единицу. Далее программа циклически выполняет чтение регистра адаптера и измеряет время до возврата в нулевое состояние бит 0-3, соответствующих четырем аналоговым каналам. Если аналоговый вход замкнут на шину GND или цепь измеряемого сопротивления разорвана, соответствующий бит не обнулится. Поэтому в программе преобразования должен быть предусмотрен тайм-аут. Для измеряемых сопротивлений в диапазоне 0-100 кОм время определяется по формуле

$$T(\text{мкс})=24,2+1 \times K(\text{кОм}).$$

Точность и линейность преобразования невысока, преобразование выполняется не быстро (до 1,12 мс) и сильно загружает процессор. Однако в отличие от «настоящих» аналого-цифровых преобразователей, этот достается даром — игровой адаптер входит в состав практически всех комбинированных плат последовательных и параллельных портов и звуковых карт.

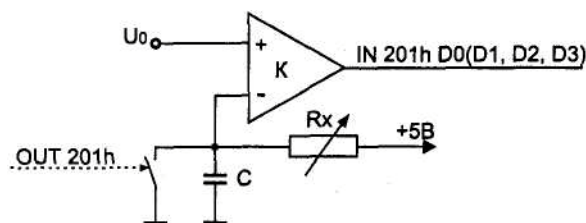


Рис. 8.19. Канал аналогового ввода

Порт имеет разъем-розетку DB-15S. Назначение выводов и соответствие сигналов битам регистра приведено в табл. 8.20. Резисторы подключаются к шине питания +5 В, кнопки — к шине GND (рис. 8.20). Замыканию кнопок соответствуют нули в битах 5-7. Аналоговые каналы можно использовать для дискретного ввода, если их входы подключить к кнопкам, замыкающим их на шину GND, и к резисторам, «подтягивающим» их к уровню +5 В. Два джойстика (А и В) подключаются через Y-образный переходник-разветвитель. На звуковых картах через разъем «Game» вместе с джойстиками могут подключаться и внешние MIDI-устройства, используя специальный кабель-адаптер, обеспечивающий гальваническую развязку входного сигнала и ограничение выходного тока (см. рис. 8.18). Для интерфейса MIDI используются контакты 12 и 15, ранее предназначавшиеся для шин GND и +5V. Такое назначение делает безопасным подключение адаптера MIDI к «чистому» игровому порту и обычного джойстика к игровому порту с сигналами MIDI.

Таблица 8.20. Интерфейс игрового адаптера и MIDI

Бит	Назначение	Контакт
7	Джойстик В кнопка #2	14
6	Джойстик В кнопка #1	10
5	Джойстик А кнопка #2	7
4	Джойстик А кнопка #1	2
3	Джойстик В Y-координата (Y2)	13
2	Джойстик В X-координата (X2)	11
1	Джойстик А Y-координата (Y1)	6
0	Джойстик А X-координата (X1)	3
-	GND	4,5,(12)
-	+5 В	1,8,9,(15)
-	MIDI In (Rx) — вход (на звуковой карте)	15
-	MIDI Out (Tx) — выход (на звуковой карте)	12

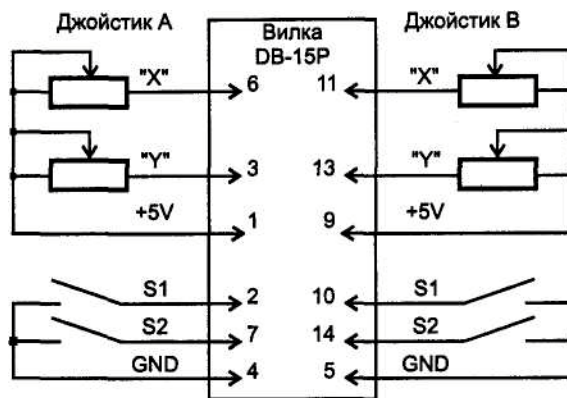


Рис. 8.20. Подключение датчиков к игровому адаптеру

Системную поддержку джойстика обеспечивает сервис BIOS Int 15 h при AH=84h. При вызове в DX задается код подфункции:

- ◆ DX=0 — опрос кнопок, возвращает в AL[7:4] состояние кнопок (соответствует битам порта 201h);
- ◆ DX=1 — чтение координат X, Y джойстика А (в регистры AX, BX) и В (в CX, DX). При ошибочном задании кода в DX устанавливается CF=1. Стандартный джойстик поддерживается и ОС Windows.

## Глава 9

# Интерфейсы устройств хранения

Устройства хранения данных — накопители на магнитных дисках (гибких и жестких, фиксированных и сменяемых), оптические диски CD и DVD, ленточные и твердотельные — подключаются к компьютеру самыми разнообразными способами. Первые устройства хранения в ПК — накопители на гибких магнитных дисках (НГМД) — подключались интерфейсным кабелем-шлейфом к контроллеру, отделенному от самих устройств. Этот специализированный интерфейс сохранился до сих пор, им подключают дисководы, требующие скорости передачи всего 500 Кбит/с (около 60 Кбайт/с). Скорость 1000 Кбит/с, требуемая для так и не распространившихся дисководов на 2,88 Мбайт, осталась невостребованной. К интерфейсу дисководов подключали и старые стриммеры (очень тихходные). Аналогичный интерфейс поначалу использовался и для подключения винчестеров (так называемых MFM и RLL), по нему передавались «сырые» данные записи чтения с головок диска, правда, усиленные. Позже накопители слегка «интеллектуализировали», и появился (ненадолго) интерфейс дисков ESDI, обеспечивающий скорость передачи данных аж 1 Мбайт/с. Однако вскоре контроллер полностью переселился на само устройство, да еще и с собственной буферной памятью (сначала на один сектор, а потом объем стал стремительно расти), и из внешнего интерфейса устройства хранения ушла вся специфика, связанная с магнитной записью и воспроизведением данных. Так появились устройства с интерфейсом AT A, начавшим свою историю в 1988 г. и ставшим в наши дни самым распространенным. Для устройств, логически отличающихся от жестких дисков — оптических, магнитооптических, ленточных и любых других, — в 1996 г. была принята спецификация AT API. Это пакетное расширение интерфейса, которое позволяет передавать по шине ATA устройству блоки командной информации, структура которых была позаимствована из SCSI. Потолок скорости ATA — 100 Мбайт/с (Ultra DMA Mode 5). Интерфейс ATA имеет уже вполне видимую границу по адресации около 137 Гбайт, в AT API используется принятая в SCSI 32-битная адресация, позволяющая адресовать до 2 Тбайт (при 512-байт-ном блоке). Дальнейшее развитие интерфейса — Serial ATA; здесь повышается скорость обмена с устройством, решается проблема одновременной работы с несколькими устройствами, значительно расширяются возможности адресации.

Для устройств хранения используют и универсальные интерфейсы, первым конкурентом ATA является шина SCSI, как в параллельном варианте так и в последовательном (FCAL). Примерно при тех же скоростях обмена SCSI позволяет эффек-

тивно работать с несколькими устройствами на одной шине, которая во время выполнения команды с длительной фазой ожидания данных свободна для обмена с другими устройствами. В отличие от сугубо внутренней шины AT A, SCSI позволяет подключать и внешние устройства. Для внешних устройств хранения с успехом применяются и шины USB, реже Fire Wire, а также подключение к LPT-порту.

## 9.1. Интерфейс НГМД

Накопители на гибких магнитных дисках (НГМД) подключаются к контроллеру через специальный стандартный интерфейс. Основные функции по управлению НГМД, а также кодирование-декодирование данных выполняет контроллер, расположенный на большинстве современных системных плат. Раньше контроллер часто выносили на специальную карту расширения («в компании» с интерфейсом НЖМД). На плате электроники, установленной на корпусе НГМД, расположены только схемы приводов двигателей, усилители-формирователи сигналов записи и считывания и формирователи сигналов от датчиков.

### 9.1.1. Аппаратный интерфейс

Все сигналы интерфейса НГМД являются логическими с уровнями ТТЛ, активный уровень — низкий. Формирователи выходных сигналов накопителя имеют выход типа «открытый коллектор». Интерфейс подразумевает наличие терминаторов — нагрузочных резисторов — для каждой сигнальной линии устройства. Теоретически их предполагалось включать только на последнем дисковом в шлейфе, практически же их никогда и не отключают. Современные трехдюймовые накопители используют «распределенный терминатор» — резисторы с относительно высоким сопротивлением (1-1,5 кОм), постоянно соединяющие входные линии интерфейса с шиной +5 В. Низкие частоты интерфейсных сигналов позволяют не задумываться о точности согласования импеданса шлейфа и сопротивления терминатора. Однако если на шлейфе стоят только старые 5"-накопители со снятыми терминаторами, они могут отказаться надежно работать (выходные линии с открытым коллектором останутся без нагрузки).

Логически интерфейс довольно прост. Для того чтобы заставить накопитель работать, его нужно выбрать сигналом Drive Sel и запустить мотор шпинделя сигналом Motor On. Для выборки накопитель имеет четыре сигнала DSO...DS3, но отзывается только на один из них, определенный установкой джамперов. 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.

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 (MFM) виде (но усиленные и сформированные в TTL-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, его иное название — Low Current или FDHDIN. Для переключения головок на «вертикальную запись» (для дискет 2,88 Мбайт) служит сигнал FDEDIN. Оба эти сигнала вырабатываются контроллером, но для самого дисководов они дублируются сигналами от датчиков типа дискеты (сигнал FDEDIN необязателен, дисковод сам переключится по сигналу от датчика). Некоторые модели дисководов позволяют изменить описанный способ работы датчиков типа дискеты, принятый для PC-совместимых ПК, — они могут быть отключены или выполнять информирование контроллера. Однако практически все кбн-троллеры сами управляют линиями интерфейса, соответствующими сигналам от этих датчиков. В этом управлении учитывается тип дисководов, описанный в CMOS Setup, и заказанный формат дискеты. Сигнал Reduce Write (низкий уровень) формируется контроллером при любом обращении к дисководу, описанному в CMOS как HD (High Density — высокая плотность, емкость 1,2 или 1,44 Мбайт), для работы с дискетами DD или QD (360 или 720 Кбайт). В некоторых контроллерах этот сигнал формировался, только когда контроллер настроен на скорость 300 Кбит/с (дискета 360/720 Кбайт в дисководе на 1,2 Мбайт). Такой контроллер может надежно форматировать и записывать дискеты 720 Кбайт в приводе на 1,44 Мбайт только при наличии правильно сконфигурированного датчика HD, иначе он все 3,5" дискеты будет записывать с высоким током записи, недопустимым для дискет QD.

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 (Extra High Density, малораспространенные дисководы на 2,88 Мбайт), причем способ использования этого контакта может определяться джамперами дисководов. 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, — для PC непригодно.

Заметим, что в интерфейсе нет никаких сигналов, прямо информирующих контроллер о готовности — наличии установленной дискеты. Контроллер может определить готовность, лишь выбрав накопитель и запустив мотор. Тогда отсутствие импульсов Index будет означать неготовность — нет дискеты или она не зафиксирована на шпинделе, или же не подключен дисковод (интерфейс или питание). Наличие дисководов контроллер может определить с помощью команды рекалибровки (см. ниже) — при ее выполнении дисковод должен подать сигнал Track00. 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 устройств 5" и двухрядными штырьковыми контактами у устройств 3,5". Используемый в PC кабель-шлейф имеет перевернутый фрагмент из 7 проводов с номерами 10-16 (рис. 9.1). Этот поворот позволяет подключать к контроллеру одним шлейфом до двух НГМД, причем адрес накопителя.

определяется его положением на шлейфе: для привода А: фрагмент перевернут, для В: — нет. Универсальный шлейф с пятью разъемами, изображенный на рисунке, позволяет подключать пару любых дисководов, которые должны располагаться в разных зонах шлейфа. Некоторые разъемы могут и отсутствовать, что скрывает свободу конфигурирования дисководов. В табл. 9.1 описан интерфейсный кабель с сигналами, приходящими на разные накопители. Направление сигналов (I/O — ввод-вывод) указано относительно контроллера.

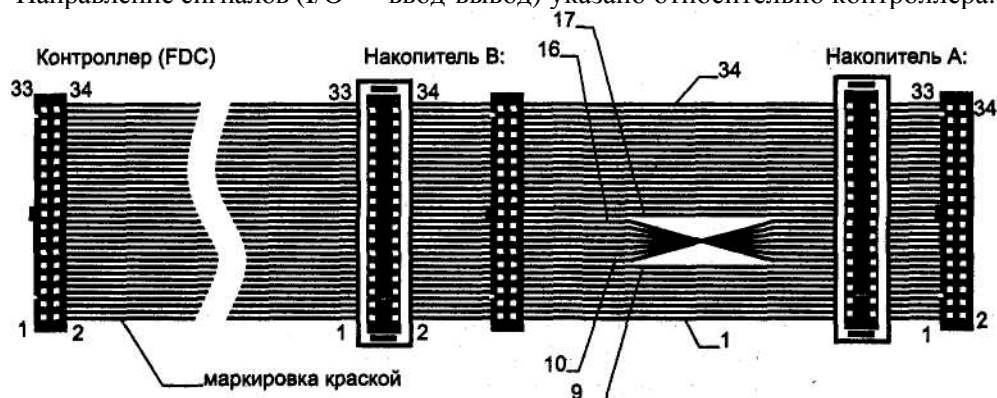


Рис. 9.1. Кабель интерфейса НГМД

Таблица 9.1. Кабель интерфейса НГМД

Контроллер		Дисковод В:		Дисковод А:		
Контакт <sup>1</sup>	Сигнал	I/O	Контакт <sup>1</sup>	Сигнал	Контакт <sup>1</sup>	Сигнал
2	FDHDI (Reduce)	O	2	Low Current	2	Low Current
4	Резерв	-	4	Резерв	4	Резерв
6	FDEDEIN	-	6	FDEDIN	6	FDEDIN (DS3)
8	Index	I	8	Index	8	Index
10	Motor On A	O	10	DSO	16	Motor <sup>2</sup>
12	Drive Sel 1	O	12	DS11	14	DS2
14	Drive Sel 0	O	14	DS2	12	DS11
16	Motor On B	O	16	Motor <sup>2</sup>	10	DSO
18	Direction	O	18	Direction	18	Direction
20	Step	O	20	Step	20	Step
22	Write Data	O	22	WData	22	Wdata
24	Write Gate	O	24	WGate	24	Wgate
26	Track 00	I	26	TB 00	26	TROO
28	Write Protect	I	28	WProt	28	WProt
30	Read Data	I	30	RData	30	Rdata
32	Side 1	O	32	Sidel	32	Side 1
34 <sup>3</sup>	Disk Changed	I	34 <sup>3</sup>	DC	34 <sup>3</sup>	DC

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

<sup>2</sup> Пара сигналов, обеспечивающая выборку FDD (Motor On A и Drive Sel 0 для дисковода А: и Motor On

В и Drive Sel 1 для дисковода В:). <sup>3</sup>

Контакт 34 в XT не используется.

Контроллер НГМД и интерфейсный кабель, принятый в РС, позволяют адресоваться к одному из двух накопителей и включать мотор сигналами Drive Sel 0 и Motor On A для накопителя А: и Drive Sel 1 и Motor On B для накопителя В:. При этом на обоих накопителях джамперы устанавливаются так, что они отзываются на сигнал DS 1 (контакт 12 разъема). Обычно джамперы на дисководе обозначаются DSO / DS1 / DS2 / DS3, и следует установить джампер DS1. Если джамперы обозначаются как DS1 / DS2 / DS3 / DS4, что встречается нечасто, то следует установить DS2. Принятая система выборки позволяет все дисководы конфигурировать однотипно, а адрес задавать положением на шлейфе. В некоторых специфических клонах РС применяют иную систему выборки накопителей и «прямой» кабель-шлейф. При этом используется выборка устройства сигналом DS0, ко переключение выборки на эту линию некоторыми накопителями не поддерживается, в результате замена накопителей в этих «фирменных» машинах может стать хлопотным делом, особенно при отсутствии технической документации.

## 9.1.2. Контроллер НГМД

Программное взаимодействие с НГМД производится через *контроллер накопителей на гибких дисках FDC* (Floppy Drive Controller), как правило, при участии контроллера DMA и прерываний. Программирование контроллера НГМД для выполнения операций с дискетами — довольно хлопотное занятие (см. [4,8,9]). Все функции, необходимые для использования НГМД в качестве устройств хранения данных, реализованы сервисами BIOS Int 13h (см. 12.8.1) и ОС. Игнорирование сервисов BIOS и даже ОС оправданно в основном лишь для нетривиальных задач типа работы с ключевыми дискетами и т. п.

Все контроллеры, применяемые в IBM РС, обеспечивают совместимость с «исторической» микросхемой контроллера NEC PD765, которая является аналогом i8272. Контроллер *FDC A T* поддерживает два накопителя; он позволяет работать со скоростями передачи данных 250 Кбит/с (для работы с дискетами SD, DD и QD в старых дисководах 5"), 500 Кбит/с для плотности HD (1,2 и 1,44 Мбайт), а также 300 Кбит/с для работы с дискетами SD, DD и QD в приводах HD 5". Современные контроллеры, поддерживающие накопители ED (2,88 Мбайт), должны обеспечивать и скорость 1000 Кбит/с. Режимы контроллеров, соответствующие скоростям 250/300,500 и 1000 Кбит/с, обозначаются соответственно 1М, 2М и 4М. В режиме 1М на одном треке умещается 9 секторов (по 512 байт); для этого при частоте вращения 300 об/мин требуется скорость 250 Кбит/с, а при 360 об/мин -300 Кбит/с. В режиме 2М при частоте вращения 300 об/мин (у дисководов HD 3,5") на одном треке умещается 18 секторов, а при частоте вращения 360 об/мин (у дисководов HD 5") — 15 секторов. Режим 4М, похоже, так и останется невостребованным. Выбор скорости задает частоту синхронизации при записи и форматировании, а также настраивает цепи Ф АПЧ сепаратора данных контроллера. Сепаратор данных способен устойчиво выделять данные и синхросигнал из сигнала, принятого с головки, лишь при небольших (порядка  $\pm 10\%$ ) отклонениях скорости от номинала, поэтому скорость должна быть правильно выбрана для всех операций обмена данными с диском.

В карте ресурсов АТ имеется место под два контроллера НГМД:

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

Контроллеры вырабатывают запрос *аппаратного прерывания* IRQ6 (BIOS I n t 0E h) по окончании выполнения внутренних операций. Для обмена данными может использоваться канал DMA2. Назначение регистров контроллера приведено в табл. 9.2.

Адрес регистра состояния FDC 3F7 (377) совпадает с адресом регистра HDC (давно неиспользуемого), в связи с чем раздельная работа FDC и HDC, расположенных на разных платах, проблематична. В нормальном (не диагностическом) режиме из этого регистра интерес представляет только бит 7 — признак смены носителя.

Таблица 9.2. Регистры контроллера НГМД Адрес

Назначение (R — чтение, W — запись)

3F2 (372) RW: DOR (Digital Output Register) — регистр управления: биты [4:7] — включение моторов А, В, С, D: 1 — включен; бит 3:1 — разрешение использования DMA2 и IRQ6; бит 2:0 — сброс, 1 — разрешение контроллера; биты [1:0] — номер выбранного привода 0-3. В АТ биты 6,7 и 1 не используются
3F3 (373) RW: TDR (Tape Drive Register) — регистр стриммера: биты [1:0] — номер устройства, которому соответствует стриммер. По сбросу устанавливается 00, но это означает отсутствие стриммера (вместо дисководов А: стриммер не поставить)
3F4 (374) R: MSR (Main State Register) — главный регистр состояния: бит 7 (DQM) — запрос: 1 — готов к приему/передаче байта; бит 6 (DIO) — направление данных: 1 — FDC -> CPU; бит 5 (NON DMA) — использование DMA: 1 — DMA не используется; БИТ4(CMDBSY): 1 — контроллер занят выполнением команды; биты [0:3] — привод А:, В:, С:, D: занят (в АТ используются только биты 0,1)
3F4 (374) <sup>1</sup> W: DSR (Datarate Select Register) — регистр выбора скорости: бит 7:1 — сброс контроллера (обнуляется автоматически); бит 6:1 — отключение питания контроллера; бит 5:1 — разрешение внешней ФАПЧ (должен быть 0); биты [4:2] — выбор времени предкомпенсации записи (000 — предкомпенсация по умолчанию); биты [1:0] — скорость обмена (00 — 500Кбит/с, 01 — 300 Кбит/с, 10 — 250 Кбит/с, 11 — 1 Мбит/с)
3F5 (375) RW: DR (Data Register) — регистр команд/данных
3F7 (377) W: CCR (Configuration Control Register) — регистр параметров контроллера (АТ): биты [7:2] — не используются; биты [1:0] — скорость обмена (00 — 500 Кбит/с, 01 — 300 Кбит/с, 10 — 250 Кбит/с, 11 — 1 Мбит/с)
3F7 (377) R: DIR (Digital Input Register) — регистр состояния (только в АТ): бит 7:1 =смена носителя (чтение инвертированной линии DC); биты [6:0] — не используются, при чтении не выводятся на шину данных

<sup>1</sup> Регистр DSR обеспечивает совместимость с контроллером 18272; в современных контроллерах НГМД на АТ-совместимых компьютерах вместо него используют CCR, поскольку требуется только задание скорости.

Контроллер НГМД может быть использован и для работы со стриммерами; специально для этого в контроллере имеется регистр TDR (Tape Drive Register), пара младших битов которого задает номер устройства, которому соответствует стриммер (для него иначе настраиваются цепи ФАПЧ сепаратора данных).

Все операции с дискетамы выполняются по командам, посылаемым хостом в регистр DR (3F5) согласно состоянию битов регистра MSR (3F4). Запись байта команды или данных в регистр DR разрешается лишь при текущем значении MSR=10xxxxxxb, чтение — при MSR=1xxxxxxb. Для записи/чтения этого регистра приходится использовать отдельные подпрограммы, не только ожидающиеся разрешающих значений MSR, но и имеющие аварийный выход по тайм-ауту. Хост также выполняет запись в регистр DOR (3F2) для запуска/останова дисководов, а также в CCR (3F7) или DSR (3F4) для выбора скорости передачи данных. В операциях обмена данными с дискетамы обычно участвует контроллер DMA, и он должен быть своевременно проинициализирован. Сигнал TC (завершение цикла DMA) используется как признак завершения фазы данных. В общем виде процедура обмена данными состоит из следующих шагов.

1. Запуск мотора и выбор дисководов (записью в регистр DOR).
2. Установка скорости (записью в регистр CCR).
3. Выполнение команды рекалибровки.
4. Ожидание раскрутки двигателя (если мотор работает еще менее 0,5 с).
5. Позиционирование головки на требуемый цилиндр.
6. Инициализация контроллера DMA.
7. Посылка команды чтения/записи.
8. Ожидание прерывания от контроллера. Прерывание произойдет, когда завершится фаза исполнения, во время которой контроллер обычно обменивается данными с хостом. Если за определенное время прерывание не получено, фиксируется неудачная попытка обращения с ошибкой тайм-аута.
9. По прерыванию от контроллера считываются байты результата, и если ошибок нет, на этом обмен успешно завершается. Если есть ошибки, то снова переходят на шаг инициализации DMA и далее повторяют команду чтения/записи. Если за несколько (3) раз успех не достигается, то выполняется рекалибровка, затем инициализация DMA и повторные попытки чтения/записи. Если успех не достигается и после нескольких рекалибровок, обмен прекращается аварийно.

Помимо описанного механизма имеется механизм автоматического выключения мотора НГМД, если к нему долго нет обращения. Для этого в фиксированном месте ОЗУ (BIOS Data Area, ячейка 0:0440) имеется счетчик времени работы мотора НГМД и адрес дисководов (в позиционном коде), мотор которого включен (0:043F). При каждом обращении к диске (когда в регистре DOR устанавливается бит включения мотора) в счетчик времени заносится константа, соответствующая выдержке на отключение (по умолчанию — 2 с). Обработчик аппаратного прерывания от таймера IRQ0 (BIOS Int 08h) с интервалом около 55 мс декрементирует счетчик времени и по его обнулению отключает мотор (записью в регистр DOR). Таким образом, если в течение заданного интервала нет обращений, мотор автоматически выключается. Конечно, это работает, лишь когда BIOS обрабатывает аппаратные прерывания от таймера, так что не отключающийся мотор дисководов может быть косвенным признаком «зависания» компьютера.

Исполнение каждой команды имеет три фазы.

1. *Фаза команды.* Контроллер устанавливает биты DQM=1 и DIO=0, что является приглашением к вводу команды. В регистр DR посылается байт команды, после которого посылаются байты параметров в строго предписанном порядке. На прием каждого байта контроллер отзывается обнулением DQM на время обработки. После получения последнего требуемого байта DQM остается обнуленным и контроллер переходит в фазу исполнения. В качестве параметров фигурируют номер цилиндра C, номер головки H, номер сектора R, код размера N или длина поля данных сектора DTL, номер последнего сектора на треке EOT, число секторов SC, длительность зазоров GPL и некоторые другие данные.
2. *Фаза исполнения* требует передачи данных от хоста к контроллеру или обратно, передача может происходить как в режиме DMA, так и чисто программно (PIO). В режиме DMA обмен выполняется по сигналам DRQ и DACK# используемого канала. Если DMA не используется, то запросом на передачу является бит DQM и сигнал прерывания. По окончании фазы вырабатывается сигнал прерывания (и его признак в регистре ST0), и контроллер переходит в фазу результата.
3. В *фазе результата* DQM=1 и OIO=1, хост должен считать байты результата из DR, после чего биты установятся в значение DQM=1 и OIO=0, что соответствует переходу в фазу приема команды.

С момента получения команды и до окончания фазы результата в регистре MSR бит CMO BSY=1. Контроллер всегда можно принудительно сбросить (перевести в начало фазы команды) записью в регистр DOR или DSR. Если контроллер получит неподдерживаемую команду, он сразу выставит DQM=1 и OIO=1, что является указанием на необходимость чтения DR. Хост должен прочитать из DR код состояния, в котором будет значение 80h (недопустимая команда).

В современных контроллерах НГМД имеется буфер FIFO глубиной 16 байт, работающий при передаче данных (но не команд и параметров) через регистр DR; в «историческом» контроллере 18272 буфера FIFO не было. Логика общения с регистром DR остается той же. Буфер должен быть своевременно обслужен, чтобы не происходило его переполнения или опустошения. Логика буфера устанавливает запрос данных так, чтобы не отвлекать хост «по пустякам». При чтении из DR запрос устанавливается, когда в буфере имеется (*16-яорог*) байтов или же считан последний байт сектора. При записи в OR запрос первоначально устанавливается и удерживается до заполнения буфера, впоследствии он устанавливается снова, когда в буфере остается *порог* байтов. Переход в фазу результата происходит, когда завершается обмен хоста с DR и буфер опустошается. Общение с регистром DR при записи команд и параметров должно выполняться согласно значениям старших битов регистра MSR (буфер FIFO при этом не используется).

## 9.2. Интерфейс ATA/ATAPI (IDE)

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 ХЗТ10, определяет на-

бор регистров устройств и назначение сигналов 40-контактного интерфейсного разъема. Интерфейс появился в результате переноса стандартного (для PC/AT) контроллера жесткого диска ближе к накопителю, то есть создания устройств со встроенным контроллером — IDE (Integrated Drive Electronics). Стандартный контроллер AT позволял подключать до двух накопителей, что в интерфейсе ATA означает параллельное подключение контроллеров двух устройств. В спецификации ATA фигурируют следующие компоненты.

- ◆ *Хост-адаптер* — средства сопряжения интерфейса ATA с шиной компьютера.  
Хостом мы будем называть компьютер с хост-адаптером интерфейса ATA.  
*Хост-контроллер* — более развитый вариант хост-адаптера.
- ◆ *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).

Хост-адаптер и устройства объединяются кабелем-шлейфом, соединяющим параллельно одноименные контакты интерфейсных разъемов. Регистры обоих контроллеров оказываются расположенными в одних и тех же областях пространства ввода-вывода. Для выбора устройства, исполняющего текущую команду, используется бит выбора накопителя (DEV) в регистре номера устройства и головки (drive/head register). Если бит DEV4, выбрано ведущее устройство, если DEV=1 — ведомое. Запись в этот регистр воспринимается сразу обоими устройствами, на обращения к остальным регистрам реагирует только выбранное. Достаточно универсальный набор сигналов позволяет подключать любое устройство со встроенным контроллером, которому в пространстве портов ввода-вывода достаточно того же набора регистров, способное поддерживать 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 AT API, основанная на тех же аппаратных средствах, но позволяющая обмениваться пакетами управляющей информации (Package Interface, PI). Структура и наполнение пакетов позаимствованы из универсального интерфейса SCSI. Пакетный интерфейс позволяет расширить границы применения шины ATA.

Адресация в ATA 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 (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. В системе CHS устройство ATA позволяет адресовать до 267 386 880 (65 536x16x255) секторов (блоков), что при размере сектора в 512 байт дает 136 902 082 560 байт (около 137 Гбайт). Позже пришли к *линейной*

адресации логических блоков LBA (Logical Block Addressing), где адрес блока (сектора) определяется 28-битным числом, что позволяет адресовать до 268 435 455 ( $2^{28}$ ) блоков (немного больше, чем в CHS). Для устройств AT A, поддерживающих и CHS, и LBA, режим адресации определяется для каждой команды битом L (бит 6) регистра D/H; режимы могут чередоваться произвольным образом. Устройства ATAPI используют принятую в SCSI 32-битную логическую адресацию, позволяющую адресовать до 2 Тбайт (при 512-байтном блоке).

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. Выводить выходные сигналы на шину ATA имеет право только выбранное устройство. Такая система подразумевает, что, начав операцию обмена с одним из 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, но эту возможность используют редко.

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.
- ◆ Serial ATA — последовательный интерфейс.
- ◆ 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 стриммеры (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<sup>+</sup> 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. Спецификация ATA/ATAPI-6 объявлена последней версией параллельного интерфейса ATA, за которой следует последовательный интерфейс *Serial ATA*.

### 9.2.1. Параллельный интерфейс ATA

Параллельный интерфейс ATA представляет собой шину, в которой все сигналы соответствуют стандартной логике TTL:

- ◆ высокий уровень *выходного* сигнала не ниже 2,4 В (при 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 применяется плоский многожильный кабель-шлейф, 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 сигнала-

лов интерфейса ATA приведен в табл. 9.3, вид кабеля — на рис. 9.2. В большинстве кабелей одноименные контакты всех разъемов соединяются своими проводами и все коннекторы равноправны. Встречаются (редко) ленточные кабели с кабельной выборкой, изображенные на рис. 9.3. 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.

Таблица 9.3. Интерфейс ATA (IDE)

Сигнал	Тип <sup>1</sup>	Контакт	Контакт	Тип <sup>1</sup>	Сигнал
RESE#	1	1	2	-	GND
DD7	I/OTS	3	4	I/OTS	DD8
DD6	I/OTS	5	6	I/OTS	DD9
DD5	I/OTS	7	8	I/O TS	DD10
DD4	I/OTS	9	10	I/OTS	DD11
DD3	I/OTS	11	12	I/O TS	DD12
DD2	I/OTS	13	14	I/OTS	DD13
DD1	I/OTS	15	16	I/OTS	DD14
DD0	I/OTS	17	18	I/OTS	DD15
GND	-	19	20	-	Ключ (нет штырька)
DMARQ	OTS <sup>2</sup>	21	22	-	GND
DIOWt/STOP <sup>3</sup>	1	23	24	-	GND
DIOR#/HDMARDY#/HSTROBE <sup>3</sup>	1	25	26	-	GND
IORDY/DDMARDY#/DSTROBE <sup>3</sup>	OTS <sup>2</sup>	27	28	I/O	SPSYNC/CSEL <sup>7</sup>
DMACK#	1	29	30	-	GND
INTRQ	OTS <sup>2</sup>	31	32	OOK	IOCS16# <sup>8</sup>
DA1	1	33	34	I,O>	PDIAG#/CBLID <sup>3</sup>
DAO	1	35	36	1	DA2
CSO#	1	37	38	1	CS1#
DASP#	I/O OK <sup>5</sup>	39	40	-	GND
+5 В (Logic)	-	41 <sup>e</sup>	42 <sup>6</sup>	-	+5 В (Motor)
GND	-	43 <sup>6</sup>	44 <sup>6</sup>	-	Зарезервирован

<sup>1</sup> 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.

<sup>2</sup> У старых устройств сигнал может иметь тип ОК (при разнотипных сигналах на одной шине возможен конфликт).

<sup>3</sup> Сигналы, приведенные после символа /, используются только в режиме Ultra DMA (AT A-4).

<sup>4</sup> У ведущего устройства — вход, у ведомого — выход.

<sup>5</sup> У ведомого устройства — только выход.

<sup>6</sup> Контакты 41-44 используются только для миниатюрных дисков.

<sup>7</sup> Начиная с ATA-3 - только CSEL.

<sup>8</sup> Начиная с AT A-3 зарезервирован.

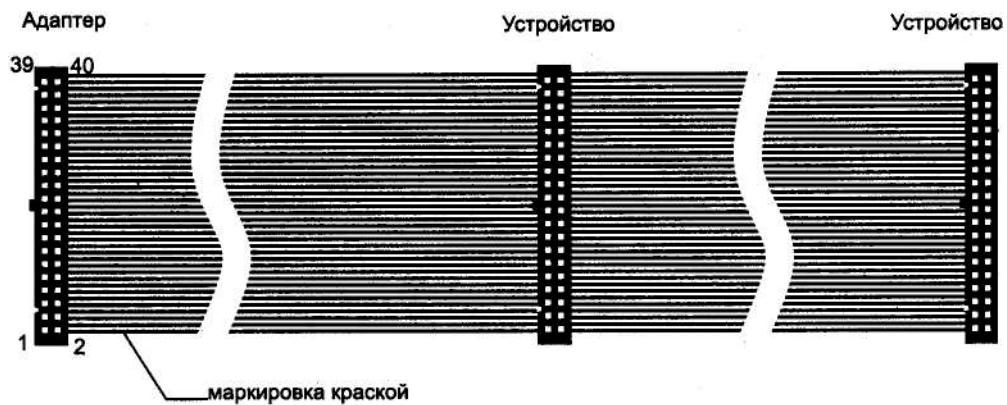


Рис. 9.2. Интерфейсный кабель АТА

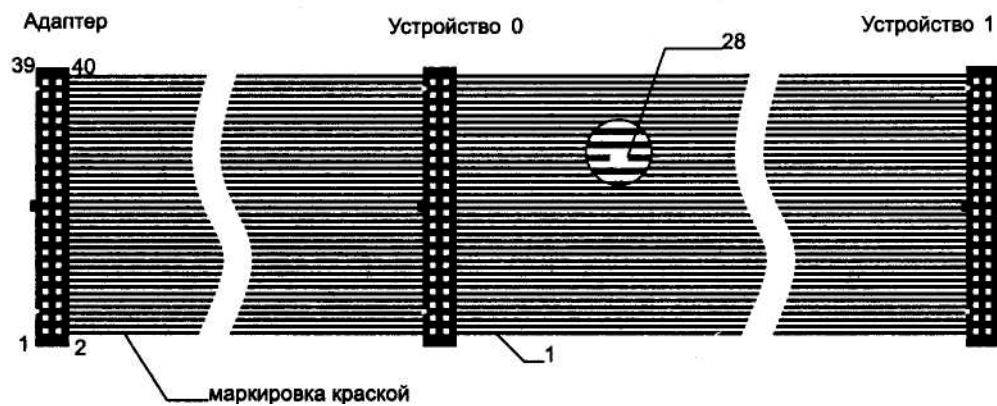


Рис. 9.3. Ленточный кабель интерфейса АТА с кабельной выборкой

Начиная с АТА/АТАРІ-4 в шлейфах узаконили кабельную выборку и для подключения устройства 1 определили средний коннектор. В нем 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. 9.3). 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).

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 UltraDMA выше 2 (скорость выше 33 Мбайт/с). Эти кабели разделяются 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). На 80-проводном кабеле в разьеме 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 со шлейфом. Корпус коннектора должен быть *синего* цвета.
- ◆ Коннектор *устройства 0* расположен на противоположном конце шлейфа, у него все контакты соединены со шлейфом. Корпус коннектора должен быть *черного* цвета.
- ◆ Коннектор *устройства 1* (необязательный) расположен в середине шлейфа, у него контакт 28 не соединен со шлейфом. Корпус коннектора должен быть *серого* цвета.

Если кабельная выборка не используется, то устройства 0 и 1 можно менять местами.

Спецификация ATA «узаконивает» как 40-контактный интерфейсный разъем, так и 4-контактный разъем питания (рис. 9.4), но для малогабаритных устройств питание может подаваться по 44-проводному интерфейсному кабелю.

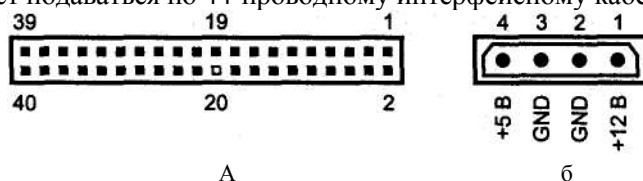


Рис. 9.4. Разъемы интерфейса ATA (вилки на устройствах): а — интерфейсный, б — питания

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 AT A/ AT API предусматривает два варианта, приведенные на рис. 9.5, а и б. Здесь пустым квадратиком обозначены позиции ключевых (пропущенных) выводов, контакты 1-40 используются для интерфейса (см. табл. 9.3), а контакты А-Н — для установки джамперов (табл. 9.4) и технологических целей. Для миниатюрных устройств предназначен 50-контактный разъем с шагом выводов 2 мм (рис. 9.5, в), у которого контакты 1-44 соответствуют табл. 9.3, контакты А-Д — табл. 9.4 (используются для конфигурирования, а пара пропущенных контактов является дополнительным ключом). Для 50-контактного кабеля принято назначение выводов накопителей IBM Thinkpad/Travelstar:

- ◆ контакт А через резистор 10 кОм соединяется с шиной +5 В;
- ◆ контакт В для устройства задает его роль: низкий уровень — устройство 0, высокий — устройство 1;

- ◆ контакт C определяет режим устройства по включении 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).

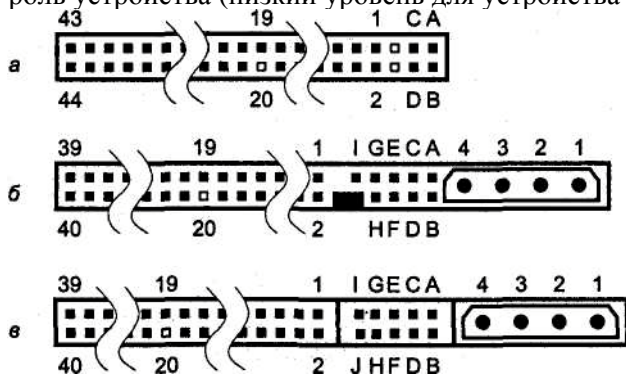


Рис. 9.5. Дополнительные контакты на разъемах интерфейса ATA: а — SFF8057, б — SFF8058, в — SFF8212 (50-контактный разъем)

Таблица 9.4. 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 CompactFlash Association.

В документации на устройства могут быть указаны несколько отличающиеся обозначения сигналов. Здесь приведены обозначения из стандарта ATA/ATAPI-4.

- ◆ 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 факт отсутствия 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 (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 TTL-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 единицу) вырабатывает только вы-

бранное устройство, когда у него имеется необслуженный запрос прерывания и его выработка не запрещено битом  $\pi$  IE N 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 !F0h-!F7h (сигнал так же 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 6с 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 сиг-

нал в течение 450 мс. Сигнал деактивируется ведомым устройством после получения им команды или через 31 с 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 или 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 этот сигнал 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 четыре линии получают новое назначение сигналов.

- ◆ STOP (Stop Ultra DMA burst) — останов передачи пакета Ultra DMA.
- ◆ DDMARDY\* (Device Ultra DMA ready) — готовность устройства при приеме пакета Ultra DMA (управление потоком).
- ◆ DSTROBE (Host Ultra DMA data strobe) — строб данных устройства при передаче пакета хосту. Данные передаются по обоим перепадам DSTROBE.
- ◆ HDMARDY\* (Host Ultra DMA ready) — готовность хоста при приеме им пакета Ultra DMA (управление потоком).
- ◆ HSTROBE (Host Ultra DMA data strobe) — 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. 9.5. Здесь имеется ряд специфичных сигналов.

- ◆ SELATA\* (Select 68-pin ATA) — сигнал, которым хост идентифицирует 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 на режим PC Card или не отвечать на другие сигналы хоста.
- ◆ CD1 # и CD2# (Card Detect) заземляются в устройстве — по этим сигналам хост определяет присутствие устройства.
- ◆ CS1# (Device chip select 1) — выбор устройства, подается хостом на оба контакта (Ни 42), но устройство воспринимает только один из них.



- ◆ 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 CSO#, CS1#, RESET\* и SELATA\* подтягиваются к пассивному состоянию.

Таблица 9.5.68-контактный интерфейс ATA для PC Card (PCMCIA)

Контакт	Сигнал	Контакт	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	CSO#	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#
22		56	CSEL
23		57	
24		58	RESET#
25		59	IORDY*
26		60	DMARQ
27	DA2	61	DMACK
28	DA1	62	DASP#
29	DAO	63	PDIAG*
30	DDO	64	DD8
31	DD1	65	DD9
32	DD2	66	DD10
33		67	CD2#
34	GND	68	GND

Для компьютеров класса ЛТ существует 8-битная версия интерфейса, называемая XT-IDE, реже — XT-Bus. Этот интерфейс [2,5], как и AT A, реализован на 40-проводном кабеле, и многие его сигналы совпадают с 16-битной шиной AT A. Интерфейс XT-IDE можно рассматривать как подмножество AT A, хотя прямой совместимости нет. Ряд устройств ATA имеют джампер выбора режима XT/AT (в накопителях Seagate на это указывает окончание «AX» в обозначении модели).

### Регистры устройств

Каждое устройство ATA имеет стандартный набор регистров, адресуемых сигналами от хост-адаптера (CSO#, CS1#, DA2, DA1, OAO, DIOR\* и DIOW#). Набор регистров (табл. 9.6) состоит из двух блоков, выбираемых сигналами CSO# 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 9.6. Регистры контроллеров устройств ATA

Адрес		Сигналы адресации канала № (0 — низкий уровень, 1 — высокий)					Назначение (R — чтение, W — запись)
1	2	CSO#	CS1#	DA2	DA1	DA10	
		1	1	X	X	X	Нет обращения (шина данных в третьем состоянии)
		0	0	X	x	X	Недопустимый адрес (шина данных в третьем состоянии)
3Fх	37X	<i>Control Block Registers — блок управляющих регистров</i>					
		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) <sup>1</sup>

продолжение^

Таблица 9.6. (продолжение)

Адрес	Сигналы адресации канала №					Назначение (R — чтение, W — запись) (0 — низкий уровень, 1 — высокий)
1	2	CS0#	CS1#	DA2	DA1	OAO
1FX 17X	<i>Command Block Registers</i> — блок командных регистров					
1F 170	0	1	0	0	0	R/W: Data (DR) — регистр данных
1F 171	0	1	0	0	1	R: Error (ER) — регистр ошибок
1F 171	0	1	0	0	1	W: Features (FR) — регистр свойств
1F 172	0	1	0	1	0	R/W: Sector Count (SC) — регистр счетчика секторов
1F 173	0	1	0	1	1	R/W: Sector Number (SN) — регистр номера сектора/LBA[7:0] <sup>2</sup>
1F 174	0	1	1	0	0	R/W: Cylinder Low (CL) — регистр младшего байта номера цилиндра/DBA[15:8] <sup>2</sup>
1F 175	0	1	1	0	1	R/W: Cylinder High (CH) — регистр старшего байта номера цилиндра/1-BA[23:1 б] <sup>2</sup>
1F 176	0	1	1	1	0	R/W: Device/Head (D/H) — регистр номера устройства и головки/1_BA[27:24] <sup>2</sup>
1F 177	0	1	1	1	1	R: Status (SR) — регистр состояния
1F 177	0	1	1	1	1	W: Command (CR) — регистр команд

<sup>1</sup> Рекомендуется, чтобы на сигнал чтения по этому адресу устройство не отвечало.

<sup>2</sup> 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 обрабатываться ив 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 — n IE N (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 онтроллера 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 и контроллер НГМД находятся на разных платах. 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 (IFOh, 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 сигнал 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 сигналы DMARQ и DMACK\*, а сигналы CS0# и CS1# неактивны.

*Регистр ошибок ER (IFlh, 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 — IDN F (ID Not Found) — указывает на ненайденный идентификатор сектора;
- ◆ бит 3 — 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 сигналом аппаратного сброса);
- ◆ бит 2 — ABRT (Aborted Command) — устанавливается, если команда отвергнута как недействительная или в случае возникновения иной ошибки;
- ◆ бит 1 — TK0NF (Track ONot 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 2 (ABRT), может меняться в зависимости от исполненной команды.

*Регистр свойств FR (1F1h, 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 спецификации ATA-2 в этот регистр помещали значение рекомендуемого номера цилиндра для предкомпенсации записи.

*Регистр счетчика секторов SC (1F2h, 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 S N (1 F3h, 173h) и номера цилиндра — младшего C L (1 F4h, 174h) и старшего байта CH (1F5h, 175h)* — имеют двойное 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 (1FGh, 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 AT A/ AT API -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 двойное назначение в зависимости от выбранной системы адресации, в режиме CHS они содержат номер головки, в режиме LBA — старшие биты логического адреса.

Как и предыдущие (5M, CHS1\_), адресный регистр D/H инициализируется хост-адаптером, а в случае возникновения ошибки при операции устройство поместит в них адрес, по которому встретилась ошибка. До принятия спецификации ATA-2 считалось, что адресные регистры должны модифицироваться и после успешного выполнения операции, отражая текущее значение адреса в носителе. *Регистр состояния S R (1 F7h, 177h)* отражает текущее состояние устройства в процессе выполнения команд: занятость, готовность, наличие ошибок и др. Чтение регистра состояния разрешает дальнейшее изменение его битов и сбрасывает запрос аппаратного прерывания.

Назначение битов регистра SR описано ниже.

- ◆ 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 E RR 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 обнулится 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 вернуться 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 флагом ERR в регистре состояния. Другие команды приводят к непредсказуемым результатам. Устройства AT API сбрасывают бит по любому сбросу и команде Execute Device Diagnostic. Бит устанавливается устройством AT A, когда оно готово к выполнению всех команд. Устройство AT API устанавливает бит до завершения выполнения команд, за исключением команд 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, 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 — I DX (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 PacketnService бит называется СНК и служит признаком исключительной ситуации.

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 (1F7h, 177h) очевидно из названия. Устройство начинает исполнять команду сразу, как только ее код записан в данный регистр. Команда Devi с Reset выполняется устройством AT API независимо от состояния битов BSY и DRQ, и даже в состоянии *Sleep*.

Полный список команд ATA приведен в литературе [4,5,9].

### Протоколы и режимы передачи данных

Программа общается с устройствами ATA через регистры, используя инструкции ввода-вывода IN и OUT. Для передачи данных с максимальной скоростью применяют программный доступ *PIO* к регистру данных инструкциями INSW/OUTSW или по каналу DMA. Тип обмена (*PIO* или *DMA*) определяется командой обращения. Программный доступ *PIO* обязателен для всех устройств, команды режима *DMA* устройствами могут не поддерживаться. Параметры различных режимов обмена приведены в табл. 9.7.

Таблица 9.7. Параметры режимов передачи

Режим передачи	Минимальное время цикла, нс	Скорость передачи, Мбайт/с	Интерфейс
PIO mode 0	600	3,3	ATA
PIO 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	FastATA-2
Ultra DMA Mode 0	120'	16,6	ATA/ATAPI-4
Ultra DMA Mode 1	80'	25	ATA/ATAPI-4
Ultra DMA Mode 2	60 <sup>1</sup>	33	ATA/ATAPI-4
Ultra DMA Mode 3	45 <sup>1</sup>	44,4	ATA/ATAPI-5
Ultra DMA Mode 4	30 <sup>1</sup>	66,6	ATA/ATAPI-5
Ultra DMA Mode 5	20 <sup>1</sup>	100	ATA/ATAPI-6

<sup>1</sup> 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.

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. Готовность устройства проверяется перед началом

передачи блока, после чего хост производит серию операций в определенном темпе, который определяется выбранным режимом *PPIO Mode 0-4*. Для каждого режима определены допустимые параметры временной диаграммы цикла обмена. Обмен PPIO 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 AT A, 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 (PPIO Mode 3 и выше) устройство может затормозить обмен, используя сигнал готовности IORDY. Программный обмен на все время передачи блока занимает и процессор, и системную шину.

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 не используются из-за низкой пропускной способности. Высокопроизводительные адаптеры ATA имеют собственные более эффективные контроллеры. Режимы обмена по каналу DMA бывают одиночными и множественными. При *одиночном режиме* — *Singleword 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 PPIO, ни в стандартных режимах DMA (а зря!). Стандартом ATA-4 было определено 3 режима Ultra DMA (0,1 и 2), впоследствии ввели новые; выбор режима осуществляется командой Set Features. В режимах Ultra DMA сигналы DMARQ и DMACK\* сохраняют свое назначение, а вот смысл сигналов DIOR#, DIQW# и IORDY на время передачи пакета (Ultra DMA Burst) существенно меняется. В пакете данные на шине сопровождаются стробом, генерируемым источником данных, причем для синхронизации используются оба перепада сигналов. Это позволяет повысить пропускную способность шины до 33 Мбайт/с, не увеличивая частоту переключения сигналов сверх  $8,33 \text{ с}^{-1}$  (этот предел для обычного кабеля достигается в режиме PPIO Mode 4 и Multiword DMA Mode 2). Каждое переданное слово участвует в подсчете CRC-кода, который передается хост-контроллером в конце пакета. Подсчет ведется и источником данных, и приемником.



При несовпадении принятого устройством кода с ожидаемым кодом фиксируется ошибка передачи, о которой устройство сообщает в конце исполнения команды. 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.

Способ сообщения об ошибке передачи зависит от типа выполнявшейся команды. Для команд READ DMA, WRITE DMA, READ DMA QUEUED или WRITE DMA QUEUED в регистре ошибок ER устанавливается бит 7 (ICRC) и бит 2 (ABRT). Для пакетной команды REQUEST SENSE (уточнить состояние) в случае ошибки в регистре состояния SR устанавливается бит 0 (CHK) и сообщается ключ состояния (Sense key) 0Bh (команда отвергнута). Для всех других пакетных команд в случае ошибки устанавливается бит CHK и сообщается состояние 04h (аппаратная ошибка), а в последующих командах REQUEST SENSE сообщается значение I5C/A5CO 08h/03h (ошибка CRC при связи с логическим устройством). Получив сообщение об ошибке, хост должен повторить команду. Если ошибки появляются постоянно, хост должен понизить скорость обмена (вплоть до выхода из режима Ultra DMA).

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 компонентов. Как правило, режимы устанавливаются системой автоматически, но пользователю дается возможность при необходимости «подрезать крылья» контроллеру настройками BIOS Setup. Для работы в режимах Ultra DMA Mode 3 и выше требуется 80-проводный кабель, 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 [4].

- ◆ *Определение типа кабеля через хост-контроллер*, для чего хост-контроллер должен иметь приемник сигнала CBLID\*. После включения питания или аппаратного сброса хост ожидает завершения протокола сброса и затем подает команду идентификации Identyf у (Packet) Devi се 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 может удерживать низкий уровень сигнала, и в этом случае 40-проводный кабель будет ошибочно трактоваться как 80-проводный.

- ◆ *Определение типа кабеля через устройство* не требует дополнительного приемника в контроллере — линия PDIAG#/CBLID# в хост-контроллере заземляется через конденсатор емкостью 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 ведомое устройство своевременно не отпустит линию, то даже 80-проводный кабель будет казаться 40-проводным.
- ◆ *Комбинированный метод определения типа кабеля* 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 это подтвердят оба метода. 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 с новыми режимами PIO. На одном ленточном кабеле (канале АТА) 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 АТА (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 протоколу DMA, если, конечно, поддерживаемый режим обеспечивает приемлемую скорость обмена. С точки зрения драйвера есть различия лишь между типами режимов (PIO, DMA и UltraDMA); аппаратные нюансы режимов внутри типа влияют только на скорость передачи. Режим UltraDMA значительно отличается от обычного DMA необходимостью обработки возможных ошибок передачи по шине; в случае постоянных ошибок драйвер должен понизить режим UltraDMA (вплоть до перехода на традиционные режимы).

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 BSY и переходит к исполнению команды.

Дальнейшие действия зависят от протокола передачи данных, заданного командой.

*Для команд, не требующих передачи данных*, следующий шаг (шаг 7) — последний. 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. 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*, процедура продолжается следующим образом.

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, процедура после шага 6 продолжается следующим образом.*

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 в регистре 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, а переданные по шине данные остаются устройством необработанными (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, процедура продолжается.
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 с некоторыми исключениями.

- ◆ Вместо PIO используется прямой доступ к памяти. Хост должен инициализировать 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.

### 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 потребоваться 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. 9.8). 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 9.8. 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 отдельные 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 отдельных формирователей сигнала DMACK#.
- ◆ Для поддержки Ultra DMA 3 и выше (44-100 Мбайт/с) все линии должны иметь отдельные 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 режимов 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 AT A, 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 1 и выше превосходят по скорости режим PIO Mode 4 (см. табл. 9.7). 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 однопоточных) системах во время дискового обмена процессор все равно ничем другим не занимается, поэтому для них хорош и режим PIO Mode. Для реального использования режима прямого управления в операционной системе должен быть установлен специальный драйвер Bus-Master, соответствующий используемому контроллеру ATA (как правило, чипсету системной платы). Стандартный контроллер PCI IDE описан ниже. 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 недоступными (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 параметр IDE 32-bit Transfer (Enable/Disable) BIOS Setup.

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 канала 3 или 4. К этим каналам можно подключать винчестеры, но будет ли их там искать BIOS во время теста 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 AT A/ AT API к шине USB. При использовании USB 2.0 простота подключения внешнего устройства будет сочетаться и с высокой скоростью передачи данных.

### Контроллер PCI IDE

Для шины PCI существует стандартный вариант интерфейса контроллера IDE (название «ATA» здесь не очень уместно, поскольку оно ориентировано на ISA-подобную шину). Спецификация «PCI IDE Controller Specification» появилась еще в 1994 году, она описывала реализацию контроллера, совместимого с интерфейсом ATA. По сравнению с обычным контроллером, у данного контроллера в блоке управляющих регистров имеется лишь один адрес (устаревший регистр адреса недоступен). Чуть позже была опубликована спецификация «Programming Interface for Bus Master IDE Controller», описывающая работу с устройствами в режиме DMA с прямым управлением шиной. Двухканальный контроллер является одной функцией PCI; четырехканальный контроллер будет уже многофункциональным устройством PCI. Для контроллера определены два режима распределения ресурсов:

- ◆ *режим совместимости* (compatibility), в котором каналам выделяются традиционные области адресов ввода-вывода и линии прерываний;
- ◆ *естественный режим PCI* (native-PCI), в котором базовые адреса блоков регистров и линии прерывания задаются в регистрах конфигурационного пространства и могут быть произвольно перемещаемы в любую область.

Распределение ресурсов для контроллера приведено в табл. 9.9. В режиме совместимости контроллер может работать только на первичной шине PCI, поскольку мост PCI-PCI не будет транслировать обращения по стандартным адресам на другую шину. В режиме PCI с перемещаемыми ресурсами контроллер может находиться на любой шине. Конкретный контроллер может поддерживать изменение режима или один из режимов. Если контроллер по умолчанию (или всегда) работает в режиме совместимости, то для ПО он «прозрачен» — работа с ним не отличается от традиционного контроллера ATA. Текущий режим и возможность

его смены отражается в байте программного интерфейса, являющегося последним элементом идентификатора класса устройства. Байт интерфейса имеет следующее назначение битов:

- ◆ бит 7 (bus mastering) — возможность работы в режиме прямого управления;
- ◆ биты 6:4 — 000 (признак стандартного интерфейса, соответствующего PCI IDE Controller Specification);
- ◆ бит 3 — индикатор программируемое™ режима для второго канала (возможности изменения бита 2);
- ◆ бит 2 — режим второго канала: 0 — совместим со стандартным контроллером АТА ІДЕ, 1 - PCI ІДЕ;
- ◆ бит 1 — индикатор программируемости режима для первого канала (возможности изменения бита 0);
- ◆ бит 0 — режим первого канала: 0 — совместим с АТА ІДЕ, 1 — PCI ІДЕ.

Таблица 9.9. Ресурсы контроллера PCI ІДЕ

Ресурс	Режим совместимости: Естественный режим PCI:			
	ресурсы для каналов		базовый адрес (смещение в конфигурационном пространстве) для канала	
	12	12	12	12
Блок командных регистров	1F0h-1F7h	170h-177h	10h	14h
Блок управляющих регистров	3F6h	376h	18h	1Ch
Прерывание	14	15		

Типовой контроллер имеет класс 01:01:80h и реализован в чипсетах большинства современных системных плат. Контроллер выглядит как расширение стандартного АТА-контроллера, обеспечивающего доступ к регистрам устройств АТ А/ АТ АРІ по известным адресам. Расширение касается прямого управления шиной PCI (bus mastering), благодаря которому можно реализовать обмен данными с устройствами в режимах DMA. Контроллер позволяет использовать все доступные ему режимы обмена РЮ (3,3-16,6 Мбайт/с), режимы DMA в стиле контроллера 8237 А (2-16,6 Мбайт/с) и режимы UltraDMA (16,6-100 Мбайт/с). Все настройки временных параметров выполняются через конфигурационные регистры PCI, их состав может быть специфичным. Однако с этими регистрами должна иметь дело только процедура POST, устанавливая для каждого обнаруженного устройства АТ А/ АТ АРІ оптимальные режимы РЮ и DMA/UltraDMA. В эти настройки может вмешаться пользователь, установив какие-либо ограничивающие параметры в BIOS Setup. В итоге после конфигурирования во время POST при обращении программ к устройствам остается выбор лишь между РЮ и DMA (если устройство поддерживает DMA). Для обмена в режиме РЮ никаких специальных действий не требуется, программа просто выполняет чтение или запись в регистр данных командами REP IN5/OUTS. Для обмена в режиме DMA требуется «зарядить» и запустить контроллер прямого управления, о чем и пойдет речь ниже.



Двухканальный контроллер имеет 16-байтный блок регистров, расположенный в пространстве портов ввода-вывода. *Базовый адрес* блока хранится в конфигурационном пространстве устройства по смещению 20h (берутся младшие 16 бит двойного слова). Расположение регистров в блоке показано в табл. 9.10, имена регистров имеют префикс ВМІ (Bus Master IDE) и окончание Р (Primary) для первого канала, S (Secondary) — для второго; в описании будем использовать окончание х (любой канал).

**Таблица 9.10.** Блок регистров контроллера PCI IDE

Смещение для канала	Длина, байт	Регистр для первого/второго каналов
J	2	
0	8	1
1	9	1
2	0Ah	1
3	0Bh	1
4	0Cи	4

*Регистр команд* ВМІСх, доступный по чтению и записи, используется для запуска контроллера и задания направления передачи. Запуск осуществляется при переходе бита 0 из нуля в единицу, останов для текущего сеанса необратим. Останов контроллера (сброс бита) обычно выполняют по окончании выполнения команды АТА/АТАРІ (по прерыванию). Преждевременный останов приведет к ошибке выполнения команды с соответствующим сообщением. Направление должно быть задано до запуска контроллера, изменять его «на ходу» нельзя.

В *регистре состояния* ВМІSх биты 1 и 2 устанавливаются аппаратно, а сбрасываются при записи байта с единичным значением в соответствующем бите. После сброса бита запроса прерывания он снова установится только по следующему фронту сигнала прерывания от устройства. Биты 5 и 6 устанавливаются программно, обычно когда POST определяет возможности подключенных устройств и программирует режимы контроллера. Регистр состояния должен быть прочитан после завершения выполнения команды для определения успешности операций на шине PCI. В регистр ВМІ DPTx заносят *адрес таблицы дескрипторов областей памяти*, с которыми производится обмен данными. Контроллер способен при чтении памяти собрать поток данных из произвольного числа областей (gathered read), а при записи «разбросать» поток по этим областям (scatter write). Такая возможность ветре-

чалась еще в EISA-системах, она позволяет преодолевать барьеры на границах страниц, свойственные стандартным контроллерам DMA и страничной переадресации процессоров x86. Каждый *дескриптор* занимает 8 байтов:

- ◆ байты 0-3 (двойное слово) — физический адрес начала области (четный);
- ◆ байты 4-5 (слово) — счетчик байтов (четный, 0000 соответствует 65 536);
- ◆ байты 6-7 (слово) — признак конца таблицы (бит 15), биты 0-14 не используются (0).

Каждая область может быть расположена в произвольном месте памяти (кроме отображаемой на шину ISA) и иметь произвольный размер, но не должна пересекать границы страниц размером 64 Кбайт. Таблица дескрипторов должна быть выровнена по границе двойного слова и не должна пересекать границы страниц, имеющих размер 64 Кбайт. Число дескрипторов в таблице произвольно, последний должен содержать признак конца таблицы. Обмен начинается с области, описанной первым дескриптором; за ней идет область следующего дескриптора и так далее до последнего. Контроллер остановится по исчерпанию счетчика в последнем дескрипторе или по инициативе устройства, если в обмене должно участвовать меньшее число данных. Если устройству данных не хватит, оно укажет на ошибку при завершении команды.

После запуска контроллера второй и нулевой биты регистра состояния отражают следующие ситуации:

- ◆ 0,1 — происходит обмен с памятью, прерывания по концу операции еще нет;
- ◆ 1,0 — обмен завершен, переданный объем соответствует объему описанных буферов (нормальное завершение);
- ◆ 1,1 — устройство завершило обмен, но по команде АТА передано меньше данных, чем описано в таблицах (допустимое завершение);
- ◆ 0,0 — ошибка на шине (бит 1), или в таблицах описано меньше данных, чем в команде АТА.

### Конфигурирование устройств

Устройства АТА перед подключением к шине должны быть корректно сконфигурированы. Конфигурирование подразумевает выбор типа интерфейса и определение адреса устройства. Тип интерфейса — ХТ или АТ — определяется моделью накопителя. В изделиях фирмы Seagate, например, тип обозначается последней буквой в шифре модели: А — АТА (16 бит,) Х — для ХТ (8 бит), а сочетание АХ означает возможность выбора АТ/ХТ с помощью джампера.

Существует два способа задания адреса устройства — кабельной выборкой или явным заданием адреса на каждом из устройств. Режим кабельной выборки включается переключком 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 (см. рис. 9.3). 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. 9.2). В этом случае переключатель CS не устанавливается, а адрес устройства задается переключками, 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#). Ниже перечислены комбинации джамперов, которые можно увидеть на устройствах IDE.

- ◆ 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 или 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 правильно сконфигурированные устройства определяются автоматически. Современные контроллеры ATA позволяют подключать даже единственное устройство как ведомое — интерфейсные функции ведущего берет на себя контроллер.

Разобраться с джамперами старых устройств трудно, если нет документации, однако обширная база данных по разным моделям встроена в справочный 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 (см. рис. 9.5).

#### ВНИМАНИЕ-----

Следует учитывать, что перестановка джамперов часто воспринимается устройством только по включении питания. Кроме того, установка на один ленточный кабель двух разнотипных устройств, если они не являются ATA-устройствами, часто невозможна.

### 9.2.2. Последовательный интерфейс Serial ATA

Параллельный интерфейс ATA исчерпал свои ресурсы пропускной способности, достигшей 100 Мбайт/с в режиме UltraDM A Mode 5. Для дальнейшего повышения пропускной способности интерфейса (но, конечно же, не самих устройств хранения, которые имеют гораздо меньшие внутренние скорости обмена с носителем) было принято решение о переходе на последовательный интерфейс. Цель перехода — улучшение и удешевление кабелей и коннекторов, улучшение условий охлаждения устройств внутри системного блока (избавление от широкого шлейфа), обеспечение возможности разработки компактных устройств, облегчение конфигурирования устройств пользователем. Попутно расширяется адресация блоков (достижение предельной емкости адресации ATA в 137 Гбайт не за горами). Спецификация Serial ATA версии 1.0 опубликована в 2001 г. и доступна в Сети по адресу [www.serialata.org](http://www.serialata.org). Сейчас уже ведутся работы над новой спецификацией Serial ATA II с большей пропускной способностью и специальными средствами для поддержки сетевых устройств хранения. Приведенная ниже информация относится к версии 1.0.

Интерфейс Serial ATA является хост-центрическим, в нем определяется только взаимодействие хоста с каждым из подключенных устройств, а взаимодействие между ведущим и ведомым устройствами, свойственное традиционному интерфейсу ATA, исключается. Программно хост видит множество устройств, подключенных к контроллеру, как набор каналов ATA, у каждого из которых имеется единственное ведущее устройство. Имеется возможность эмуляции пар устройств (ведущее — ведомое) на одном канале, если такая необходимость возникнет. Программное взаимодействие с устройствами Serial ATA практически совпадает с прежним, набор команд соответствует ATA/ATAPI-5. В то же время аппаратная реализация хост-адаптера Serial ATA сильно отличается от примитивного (в исходном варианте) интерфейса ATA. В параллельном интерфейсе ATA хост-адаптер был простым средством, обеспечивающим программное обращение к регистрам, расположенным в самих подключенных устройствах. В Serial ATA ситуация иная: хост-адаптер имеет блоки так называемых «теневого» регистров (Shadow Registers), совпадающих по назначению с обычными регистрами устройств ATA. Каждому подключенному устройству соответствует свой набор регистров. Обращения к этим теневым регистрам вызывают процессы взаимодействия хост-адаптера с подключенными устройствами и исполнение команд.

В стандарте рассматривается многоуровневая модель взаимодействия хоста и устройства, где прикладным уровнем является обмен командами, информацией о состоянии и хранимыми данными. На физическом уровне для передачи информации между контроллером и устройством используются две пары проводов. Данные передаются кадрами, транспортный уровень формирует и проверяет корректность информационных структур кадров (Frame Information Structure, FIS). Для облегчения высокоскоростной передачи на канальном уровне данные кодируются по схеме 8B/10B (8 бит данных кодируются 10-битным символом) и скремблируются, после чего по физической линии передаются по простейшему методу NRZ (уровень сигнала соответствует передаваемому биту). Между канальным и прикладным уровнем имеется транспортный уровень, отвечающий за доставку кадров. На каждом уровне имеются свои средства контроля достоверности и целостности.

В первом поколении Serial ATA данные по кабелю передаются со скоростью 1500 Мбит/с, что с учетом кодирования 8B/10B обеспечивает скорость 150 Мбайт/с (без учета накладных расходов протоколов верхних уровней). В дальнейшем планируется повышать скорость передачи, и в интерфейсе заложена возможность согласования скоростей обмена по каждому интерфейсу в соответствии с возможностями хоста и устройства, а также качеством связи. Хост-адаптер имеет средства управления соединениями, программно эти средства доступны через специальные регистры Serial ATA.

В стандарте предусматривается управление энергорежимом интерфейсов. Каждый интерфейс кроме активного состояния может находиться в состояниях *PARTIAL* и *SLUMBER* с пониженным энергопотреблением, для выхода из которых требуется заметное время (10 мс).

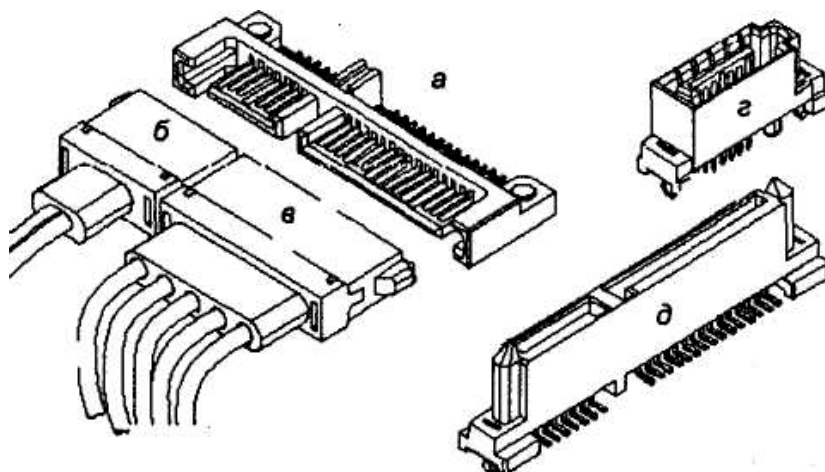
Команды, требующие передачи данных, могут исполняться в различных режимах обмена. Обращение в режиме PIO и традиционный способ обмена по DMA (legacy DMA) выполняется аналогично привычному интерфейсу AT A. Однако внутренний протокол обмена между хост-адаптером и устройствами позволяет передавать между ними разноплановую информацию (структуры FIS определены не только для команд, состояния и собственно хранимых данных). В приложении D к спецификации описывается весьма своеобразный способ обмена по DMA, который предполагается основным (First-party DMA) для устройств Serial ATA. В традиционном *контроллере DMA адаптера AT A* для каждого канала имеется буфер, в который перед выполнением операции обмена загружают дескрипторы блоков памяти, участвующей в обмене (см. п. 9.2.1). Теперь же предполагается, что адресная информация, относящаяся к оперативной памяти хост компьютера, будет доводиться до *устройства хранения*, подключенного к адаптеру Serial ATA. Эта информация из устройства хранения при исполнении команд обмена выгружается в контроллер DMA хост-адаптера и используется им для формирования адреса текущей передачи. Мотивы и полезность этого нововведения не совсем понятны; расплатой за некоторое упрощение хост-адаптера (особенно многоканального) является усложнение протокола и расширение функций, выполняемых устройством хранения. Все-таки более привычно традиционное разделение функций, при котором задача устройств внешней памяти — хранить данные, «не интересуясь» тем, в каком месте оперативной памяти компьютера они должны находиться при операциях обмена.

### Физический интерфейс Serial ATA

Последовательный интерфейс ATA, как и его параллельный предшественник, предназначен для подключений устройств внутри компьютера. Длина кабелей не превышает 1 м, при этом все соединения радиальные, каждое устройство подключается к хост-адаптеру своим кабелем. В стандарте предусматривается и непосредственное подключение устройств к разъемам кросс-платы с возможностью «горячей» замены. Стандарт определяет новый однорядный двухсегментный разъем с механическими ключами, препятствующими ошибочному подключению. Сигнальный сегмент имеет 7 контактов (S1-S7), питающий — 15 (P1-P15); все контакты расположены в один ряд с шагом 1,27 мм. Назначение контактов приведено в табл. 9.11. Малые размеры разъема (полная длина — около 36 мм) и малое количество цепей облегчают компоновку системных плат и карт расширения. Питающий сегмент может отсутствовать (устройство может получать питание и от обычного 4-контактного разъема ATA). Вид разъемов приведен на рис. 9.6. Для обеспечения «горячего» подключения контакты разъемов имеют разную длину, в первую очередь соединяются контакты «земли» P4 и P12, затем остальные «земли» и контакты предзаряда конденсаторов в цепях питания P3, P7 и P13 (для уменьшения броска потребляемого тока), после чего соединяются основные питающие контакты и сигнальные цепи.

**Таблица 9.11.** Разъем Serial ATA

Контакт	Цепь	Назначение
51	GND	Экран
52	A+	Дифференциальная пара сигналов A
53	A-	Дифференциальная пара сигналов A
54	GND	Экран
55	B-	Дифференциальная пара сигналов B
56	B+	Дифференциальная пара сигналов B
57	GND	Экран
<i>Ключи и свободное пространство</i>		
P1	V33	Питание 3,3 В
P2	V33	Питание 3,3 В
P3	V33	Питание 3,3 В, предзаряд
P4	GND	Общий
P5	GND	Общий
P6	GND	Общий
P7	V5	Питание 5 В, предзаряд
P8	V5	Питание 5 В
P9	V5	Питание 5 В
P10	GND	Общий
P11	Резерв	
P12	GND	Общий
P13	V12	Питание 12В, предзаряд
P14	V12	Питание 12 В
P15	V12	Питание 12 В



**Рис. 9.6.** Разъемы Serial ATA: а — полный разъем на устройстве, б — сигнальный сегмент кабельного разъема, в — питающий сегмент кабельного разъема, г — сигнальный сегмент разъема хост-адаптера, д — разъем хоста для непосредственного подключения устройства

## Регистры Serial ATA

Каждое устройство, подключенное к адаптеру Serial ATA, представляется тремя блоками регистров, два из которых соответствуют традиционным регистрам ATA (см. п. 9.2.1) и называются «теньевыми», третий блок — новый. Привязка адресов блоков к адресному пространству хоста стандартом не регламентируется; для PCI-контроллера блоки задаются регистрами конфигурационного пространства и «теньевые» регистры могут располагаться по стандартным адресам ATA.

В блоке управляющих регистров, как и в ATA, используется лишь один (AS для чтения, DC для записи). В блоке командных регистров разрядность регистров SC, SN, CL и CH расширена до 16 бит, назначение младших байтов сохранилось. В режиме LBA старшие байты регистров SN, CL и CH несут биты логического адреса [24:31], [32:39] и [40:47] соответственно. В регистре D/H бит DEV игнорируется (при эмуляции пар устройств на одном канале бит DEV используется для выбора устройства). Из спецификации не совсем ясно, используются ли младшие биты D/H для задания бит LBA[27:24], поскольку эти же биты фигурируют в старшем байте SN.

Новый блок регистров SCR (Serial ATA Status and Control registers) состоит из 16 смежных 32-разрядных регистров SCR0-SCR15, из которых пока определены лишь 3 (остальные зарезервированы).

*Регистр SStatus* (SCR0) — регистр текущего состояния интерфейса хост адаптера (только чтение).

◆ Биты [3:0] — поле DET, подключение устройств:

- 0000 — устройство не обнаружено, физической связи нет;
- 0001 — устройство обнаружено, но физическая связь не установлена;
- 0010 — устройство обнаружено, физическая связь установлена;
- 0100 — устройство отключено (запретом интерфейса или запуском внутреннего теста).

- ◆ Биты [7:4] — SPD, скорость:
  - 0000 — нет согласованной скорости (устройство не подключено или связь не установлена);
  - 0001 — согласована скорость 1-го поколения.
- ◆ Биты [11:8] — поле IPM, состояние энергопотребления интерфейса:
  - 0000 — устройство не обнаружено, физической связи нет;
  - 0001 — интерфейс в активном состоянии;
  - 0010 — интерфейс в состоянии *PARTIAL*;
  - 0110 — интерфейс в состоянии *SLUMBER*.

Остальные биты и значения полей зарезервированы.

*Register SError (SCR1)* — регистр диагностической информации, относящейся к интерфейсу. В регистре представлены ошибки, накапливающиеся с момента последней очистки регистра. Регистр очищается операцией сброса, отдельные биты можно сбрасывать операцией записи в регистр (единицы в позициях сбрасываемых бит).

- ◆ Биты [15:0] — поле *ERR*, ошибки, обрабатываемые обычным программным обеспечением:
  - бит 9 — С — устойчивая неисправимая ошибка связи или нарушение целостности данных (может возникать при неправильном подключении, отказе или отключении устройства);
  - бит 11 — Е — внутренняя ошибка, обнаруженная хост-адаптером (если ошибка повторяется после сброса, она может быть признаком несовместимости адаптера и устройства);
  - бит 0 — I — исправленное нарушение целостности данных (не требует действий со стороны ПО, но может учитываться, например, для принятия решения о снижении скорости);
  - бит 1 — М — исправленная ошибка связи (может возникать при временном отключении устройства, потере синхронизации, не требует действий со стороны ПО);
  - бит 10 — Р — протокольная ошибка, требует сброса интерфейса и повтора операции (в случае устойчивости может быть признаком несовместимости устройства и адаптера);
  - бит 8 — Т — не исправленное нарушение целостности данных, требует повтора операции.
- ◆ Биты [31:16] — поле *DIAG*, используемое диагностическим ПО:
  - бит 19 — В — ошибка декодирования 10В/8В;
  - бит 21 — С — ошибка CRC на канальном уровне;
  - бит 20 — D — ошибка паритета блоков данных (Disparity);
  - бит 26 — F — неопознанный тип FIS (ошибка, обнаруженная на транспортном уровне при корректном CRC-коде);
  - бит 17 — I — внутренняя ошибка физического уровня интерфейса;



- бит 16 — N — сигнал готовности физического уровня менял состояние;
- бит 22 — H — ошибка подтверждения кадров (может возникать как следствие ошибок B, C, D);
- бит 23 — S — ошибка последовательности состояний канального уровня;
- бит 24 — T — ошибка на транспортном уровне;
- бит 18 — W — обнаружен пробуждающий сигнал *Comm Wake*.

Остальные биты зарезервированы.

*Regisrtr SControl (SCR2)* — регистр управления интерфейсом (запись и чтение).

- ◆ Биты [3:0] — поле DET, управление обнаружением и инициализацией устройств:
  - 0000 — нет действий;
  - 0001 — инициализация и установление соединения (эквивалентно аппаратному сбросу);
  - 0100 — запрет интерфейса и перевод физического уровня в режим *Offline*.
- ◆ Биты [7:4] — поле SPD, задает ограничение на скорость при согласовании соединений:
  - 0000 — нет ограничений;
  - 0001 — скорость не выше 1-го поколения.
- ◆ Биты [11:8] — поле IPM, управление энергорежимом интерфейса:
  - 0000 — нет ограничений;
  - 0001 — запрет перехода в состояние *PARTIAL*;
  - 0010 — запрет перехода в состояние *SLUMBER*;
  - 0011 — запрет перехода в состояние *PARTIAL* и *SLUMBER*.

Остальные биты и значения полей зарезервированы.

### 9.3. Интерфейсы и конструктивы твердотельных носителей информации

Твердотельные устройства хранения используются в миниатюрных компьютерах, а также компьютеризированных цифровых устройствах бытовой электроники — фотокамерах, плеерах, приемниках глобальной системы позиционирования (GPS), музыкальных инструментах и прочих. В большинстве своем эти устройства основаны на микросхемах флэш-памяти, в наиболее современных используется память со структурой NAND (см. п. 7.3.2). Этому типу флэш-памяти присуще быстрое чтение, запись и стирание небольших блоков (256 или 512 байт), что удобно для записи файлов. Правда, для этой памяти характерны довольно медленное чтение произвольного байта и отсутствие возможности побайтной записи, но приложениям этих карт такая возможность и не требуется, поскольку они ориентированы на блочный обмен. Устройства на флэш-памяти являются энергонезависимыми (в режиме хранения не требуют питания), экономичными в плане потребления,

особенно при чтении, достаточно производительными, но, увы, недешевыми. Запись на эти носители специфична: быстрее всего она выполняется в чистый (стертый) блок (сектор диска), а перезапись требует относительно длительного стирания. Кроме того, флэш-память имеет хоть и большое (порядка  $10^5$ ), но ограниченное число циклов стирания-записи — как ни странно, у носителей с подвижным носителем с этим ограничением не сталкиваются. Устройства хранения обычно представляют комбинацию собственно микросхем памяти и микроконтроллера, обеспечивающего внешние интерфейсные функции. Этим они отличаются от карт памяти с линейным доступом, например Miniature Card, на которых располагаются только микросхемы памяти (встроенный контроллер не требуется).

Есть и другие твердотельные хранители — например, ферроэлектрическая память (FRAM), но пока что массового применения они не имеют. Менее чем за десятилетие устройства хранения на флэш-памяти прошли большой путь от электронных «дисков» («винчестеров») размером в 3" до современных, размером с почтовую марку. Интерфейсы твердотельных устройств хранения — карт внешней памяти — тесно связаны с их конструктивами; основные характеристики наиболее распространенных карт приведены в табл. 9.12.

**Таблица 9.12.** Основные характеристики карт внешней памяти

	CompactFlash	SmartMedia Card	MultiMedia Card	Secure Digital	Miniature Card
Длина	36,0	45,0	32,0	32,0	33,0
Ширина	43,0	37,0	24,0	24,0	38,0
Высота	3,3/5,0	0,76	1,4	2,1	3,5
Коннектор	Штырьковый	Печатный	Печатный	Печатный	Эластомер
Число контактов	50	22	7	9	60

Вышеперечисленные карты можно подключать и к обычным компьютерам. Для этого существуют различные адаптеры: для слотов PC Card (к блокнотным ПК), для шины USB (для самых разных ПК) и для других внешних интерфейсов. С помощью этих адаптеров компьютер «видит» подключенную карту как обычный сменный носитель информации (диск). Конечно, компьютеры общего назначения в такой внешней памяти не нуждаются — своя и больше, и дешевле. Главная цель подключения карты к ПК — быстрая передача прикладных данных бытового устройства (фотографий, музыки и т. п.) или перенос (хранения) информации (вместо дискет).

В качестве интерфейса твердотельных устройств хранения может использоваться и самый обычный вариант AT A. Существуют устройства DOM (Disk On Module) — небольшие модули, которые вставляются в обычный 40-контактный разъем АТА, имеющийся на любой современной системной плате. Правда, этот «винчестер» не блещет выдающимися параметрами: объем — 4-256 Мбайт, средняя скорость передачи данных — 1,6 Мбайт/с, удельная стоимость хранения — около \$1,5 за мегабайт. Внутри модуля находится флэш-память структуры NAND с контроллером, эмулирующим систему команд АТА. Основное применение этих модулей — хранение ПО встраиваемых компьютеров, но можно ими пользоваться и как сменными устройствами хранения.

### 9.3.1. CompactFlash

Карты CompactFlash (рис. 9.7), поддерживаемые ассоциацией CFA (Compact Flash association), широко используются в различных электронных приборах: цифровых фотокамерах, фотопринтерах, MP3-плеерах, цифровых диктофонах, персональных коммуникаторах и, конечно же, компьютерах — настольных, карманных, автомобильных. Карты имеют размер 42,8 x 36,4 x 3,3 мм (4 мм с учетом выступа) и 50-контактный разъем (розетка на карте, двухрядный штырьковый разъем с шагом 1,27 мм на слоте). Назначение контактов приведено в табл. 9.13. Через переходник с 50 на 68-контактный разъем карты могут устанавливаться в слот PC Card Type II или III, имеющийся практически во всех блокнотных ПК. Объем памяти выпускаемых в настоящее время карт — от 4 Мбайт до 1 Гбайт, напряжение питания — 5 или 3,3 В. Карты могут работать в одном из трех режимов: карт памяти (Mem), карт ввода-вывода PC Card (I/O), «чистого» режима IDE (ATA). В первых двух режимах карты работают с теми же интерфейсными сигналами, что и PC Card. В режиме IDE электрический интерфейс и система команд полностью совместимы со спецификацией ATA (см. п. 9.2.1), правда, обмен данными возможен только в режиме PIO. Режим IDE выбирается заземлением на стороне хоста сигнала ATA\_SEL#. При этом из шины адреса используются только A[2:0] (остальные заземлены хостом); шина данных при обращениях к регистрам ATA имеет разрядность 8 бит, а при передаче данных — 16. Сигналы CS0# и CS1# используются для выбора блока командных и управляющих регистров соответственно. Сигналы PDIAG#, DASP#, CSEL#, RESET\* и IORDY соответствуют спецификации ATA. Сигналом CSEL# выбирается роль карты: при заземленном контакте — устройство 0 (master), при разомкнутом — устройство 1 (slave); можно воспользоваться и «кабельной выборкой». Сигналы REG# и WE# должны подключаться к шине питания (V<sub>cc</sub>). Сигналы CD1# и CD2# являются индикаторами установки карты (их контакты замыкаются последними, на карте они заземлены). В табл. 9.14 описан чисто пассивный переходник, позволяющий подключить карту Compact Flash к обычному порту ATA (IDE), имеющемуся на любой современной системной плате.

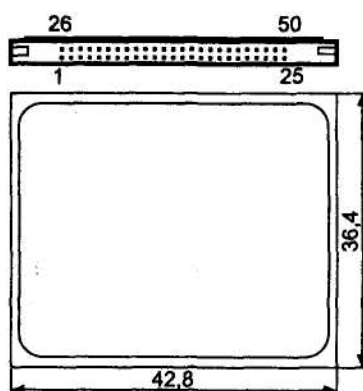


Рис. 9.7. Карты CompactFlash

Таблица 9.13. Назначение контактов карт CompactFlash

№	Mem	I/O	IDE	№	Mem	I/O	IDE
1	GND	GND	GND	26	CD1#	CD1#	CD1#
2	DOS	DOS	DOS	27	D11	D11	D11
3	D04	D04	D04	28	D12	D12	D12
4	DOS	DOS	DOS	29	D13	D13	D13
5	D06	D06	D06	30	D14	D14	D14
6	D07	D07	D07	31	D15	D15	D15
7	CE1#	CE1#	CSO#	32	CE2#	CE2#	CS1#
8	A10	A10	A10	33	VS1#	VS1#	VS1#
9	OE#	OE#	ATA_SEL#	34	IORD#	IORD#	IORD#
10	A09	A09	A09	35	IOWR#	IOWR#	IOWR#
11	A08	A08	A08	36	WE#	WE#	WE#
12	A07	A07	A07	37	RDY/BSY	IREQ	INTRQ
13	VCC	VCC	VCC	38	VCC	VCC	VCC
14	A06	A06	A06	39	CSEL#	CSEL#	CSEL#
15	A05	A05	A05	40	VS2#	VS2#	VS2#
16	A04	A04	A04	41	RESET	RESET	RESET#
17	A03	A03	A03	42	WAIT*	WAIT#	IORDY
18	A02	A02	A02	43	INPACK*	INPACK*	INPACK#
19	A01	A01	A01	44	REG#	REG#	REG#
20	AGO	AOO	AOO	45	BVD2	SPKR#	DASP#
21	DOO	DOO	DOO	46	BVD1	STSCHG#	PDIAGtf
22	D01	D01	D01	47	D08	DOS	DOS
23	D02	D02	D02	48	D09	D09	D09
24	WP	IOIS16*	IOCS16*	49	D10	D10	D10
25	CD2#	CD2#	CD2#	50	GND	GND	GND

Таблица 9.14. Переходник CompactFlash Card — ATA

ATA, конт. №	Цепь	CFC, конт. №	CFC, конт. №	Цепь	ATA, конт. №
2	GND	1	26	CD1#	-
11	DOS	2	27	D11	10
9	D04	3	28	D12	12
7	DOS	4	29	D13	14
5	D06	5	30	D14	16
3	D07	6	31	D15	18
37	CSO#	7	32	CS1#	38
GND <sup>1</sup>	A10	8	33	VS1#	-
GND <sup>1</sup>	ATA_SEL#	9	34	IORD#	25
GND <sup>1</sup>	A09	10	35	IOWR#	23
GND <sup>1</sup>	A08	11	36	WE#	+5V <sup>2</sup>
GND <sup>1</sup>	A07	12	37	INTRQ	31
+5V <sup>2</sup>	VCC	13	38	VCC	+5V <sup>2</sup>
GND <sup>1</sup>	A06	14	39	CSEL#	28 <sup>3</sup>

продолжение &amp;

Таблица 9.14 (продолжение)

АТА, конт. №	Цепь	СFC, конт. №	СFC, конт. №	Цепь	АТА, конт. №
GND <sup>1</sup>	A05	15	40	VS2#	-
GND <sup>1</sup>	A04	16	41	RESET*	1
GND <sup>1</sup>	A03	17	42	IORDY	27
36	A02	18	43	INPACK#	-
33	A01	19	44	REG#	+5V <sup>2</sup>
35	A00	20	45	DASP#	39
17	DOO	21	46	PDIAG#	34
15	D01	22	47	D08	4
13	D02	23	48	D09	6
32	IOCS16#	24	49	D10	8
-	CD2#	25	50	GND	GND'

<sup>1</sup> С шиной GND должны быть соединены контакты 2,19,22,24,26,30 и 40 разъема АТА, а также контакт разъема питания.

<sup>2</sup> Питание +5V на переходник подается от дополнительного разъема.

<sup>3</sup> Возможно подключение не к разъему АТА, а к шине GND через джампер: замкнут — ведущее устройство, разомкнут — ведомое.

### 9.3.2. SmartMedia Card

Карты *SmartMedia Card*, поддерживаемые ассоциацией PCMCIA, предназначены примерно для того же круга приложений, что и CompactFlash. Они совсем тонкие, имеют менее «нежный» печатный разъем с малым числом контактов (всего 22) и не боятся не только повышенной влажности, но и воды. Карты основаны на микросхемах флэш-памяти с организацией запоминающих ячеек NAND. Средняя скорость передачи данных — 2 Мбайт/с, пиковая — до 10. Вид карт приведен на рис. 9.8, назначение контактов — в табл. 9.15. Карты на 5 В выпускаются объемом 16и32 Мбит (2 и 4 Мбайт). Карты на 3,3 В выпускаются объемом 16, 32 или 64 Мбит (2, 4, 8 Мбайт); у них контакт 17 соединен с V<sub>cc</sub>. Для карт SmartMedia выпускают простые переходные адаптеры на слот PC Card Type II. Появились даже устройства FlashPath™ для считывания этих карт в обычном дисководе(!) 1,44 Мбайт.

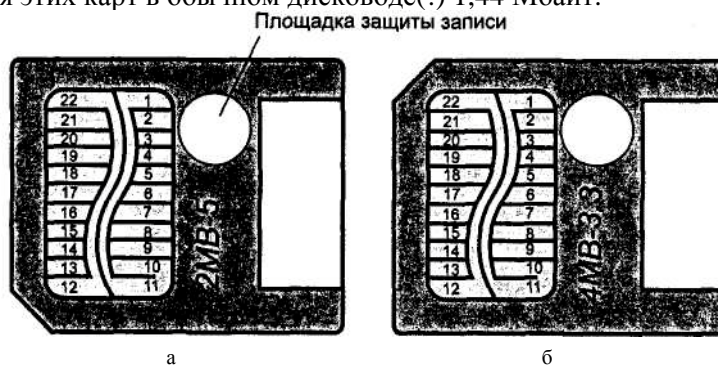


Рис. 9.8. Карты SmartMedia: а — питание 5 В, б — питание 3,3 В

Таблица 9.15. Назначение контактов SmartMedia Card

№	Цепь	№	Цепь
22	VCC(5 В)	1	Vss
21	CE#	2	CLE
20	RE#	3	ALE
19	R/B	4	WE#
18	GND	5	WP#
17	Power Detect	6	I/O1
16	I/O8	7	I/O2
15	I/O?	8	I/O3
14	I/O6	9	I/O4
13	I/O5	10	Vss
12	Vcc	11	vss

### 9.3.3. MultiMediaCard и Secure Digital

Карты *MultiMediaCard* (MMC) и *Secure Digital* (SD) еще компактнее — их размер в плане всего 32x24 мм при толщине 1,4 мм у MMC и 2,1 мм у SD. Эти карты имеют краевой печатный разъем, 7-контактный у MMC и 9-контактный у SD. У карт MMC на разъем выводятся питание, управляющий сигнал, сигнал синхронизации и двухразрядная шина данных. У карт SD за счет пары дополнительных контактов шина данных расширена до 4 бит, что позволяет повысить скорость обмена. Конструктивно карты SD сделаны так, что устройства, работающие с ними, физически могут работать и с картами MMC, но не наоборот (в тонкий слот MMC более толстую карту SD и не вставить). Логическая совместимость должна обеспечиваться программным обеспечением хоста (устройства, в которое вставляют карту). Карты MMC и SD предназначены примерно для того же спектра устройств, что и CompactFlash; к ним стоит добавить и электронные книги (eBook). Однако назначение этих карт различно: MMC предназначены для широкого распространения данных — музыки, игр, электронных книг, — и они являются довольно дешевыми носителями информации. Карты SD предназначены для безопасного (в смысле конфиденциальности) распространения информации, и они гораздо дороже. Карты SD разработаны альянсом трех компаний — Toshiba, Matsushita (более известная по торговой марке Panasonic) и SanDisk, которые организовали ассоциацию SDA (Secure Digital Association), выпускающую спецификации, принятые уже сотнями фирм во всем мире. Членство в ассоциации платное и дорогостоящее; по вполне понятным причинам технические детали SD широко не освещаются (иначе грош цена провозглашаемой безопасности).

Карты SD (рис. 9.9) имеют размер 32x24x2,1 мм; справа на рисунке виден переключатель защиты записи (WP), предохраняющий от случайного стирания. На печатной плате карты SD смонтирована флэш-память структуры NAND, SD-контроллер и вспомогательные компоненты. 9-контактный разъем карт SD (табл. 9.16) по выводам 1-7 совпадает с картами MMC. Карты SD допускают «горячее» подключение/отключение. Конструкция коннектора рассчитана на 10 000 циклов вставки-изъятия. Карты выдерживают до 200 000-300 000 циклов записи в каждый блок флэш-памя-

ти и падение на пол с высоты 3 м. Они устойчивы и к жаре, и к морозу. Скорость передачи данных у первых карт SD — 2 Мбайт/с, объем — 8-512 Мбайт; в 2002 г. планируется подъем скорость до 10 Мбайт/с и объем до 1 Гбайт, а в 2004 — до 20 Мбайт/с и 4 Гбайт. Начальная удельная цена хранения высока — \$3 за 1 Мбайт, но постепенно она должна снизиться до уровня \$1 за 1 Мбайт (что тоже немало!) и ниже.

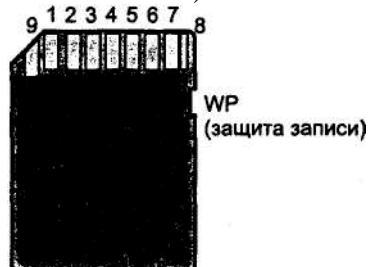


Рис. 9.9. Карта памяти SD (вид сзади)

Для карт SD разработаны (и разрабатываются) спецификации форматов данных на трех уровнях:

- ◆ физический уровень описывает перезаписываемые карты (SD-Rewritable) и постоянную память (SD-Read Only);
- ◆ уровень файловой системы использует стандарт ISO 9293;
- ◆ прикладной уровень учитывает специфику атрибутов различных типов содержимого — музыки (SD-Audio), изображений (SD-Picture), речи (SD-Voice), видео (SD-Video) и других.

Таблица 9.16. Назначение контактов карт MMC и SD

Контакт	Цепь	Контакт	Цепь
1	Command	6	Data
2	Vss	7	Data1
3	Vdd	8	Data 2
4	Clock	9	Data3
5	Vss		

На всех уровнях действуют средства безопасности (Security), являющиеся основным «коньком» SD. В SD используются технологии обеспечения безопасности CPRM (Content Protection for Recordable Media — защита содержимого записываемых носителей) — стандарта шифрования и сертификации/аутентификации, разработанного и лицензируемого фирмами IBM, Intel, Matsushita (Panasonic) и Toshiba. Карты SD-Audio отвечают требованиям SDMI<sup>1</sup> (Secure Digital Music Initiative) к портативным устройствам.

<sup>1</sup>SDMI — коалиция производителей аудиоаппаратуры RIAA (Recording Industry Association of America) и компаний музыкальной индустрии (включающей ведущие звукозаписывающие фирмы UMG, BMS, EMI, WMG, SME), направленная на защиту авторских прав на музыкальные произведения.

Карта SD имеет три области хранения с разными возможностями доступа: область хранения ключей шифрования и аутентификации, область секретных данных и область данных общего назначения. Секретные данные хранятся и передаются в зашифрованном виде, их кодирование-декодирование выполняется хостом (устройством, в которое устанавливается карта). Для того чтобы установить канал обмена секретными данными, требуется взаимная аутентификация хоста и карты: хост должен «признать» карту, а карта — хост. Таким образом, обмен данными с защищенной областью карты возможен только на «фирменных» устройствах (до тех пор пока алгоритмы и ключи шифрования не попадут в руки хакеров).

Карты SD продаются в форматированном виде; при необходимости они могут быть переформатированы на специальном устройстве или хосте SD, имеющем функции форматирования. Нештатное форматирование (например, на компьютере) может привести карту в состояние, негодное для дальнейшего применения, — защита может сработать как дверь, захлопнувшаяся на замок, единственный ключ от которого остался внутри.

Совместимость SD и MMC весьма ограничена. Хосты SD-карт совместимы с картами MMC на физическом уровне; в картах MMC и SD используется одна и та же структура таблицы размещения файлов (FAT), чем обеспечивается совместимость и на уровне записи/чтения файлов. Однако на прикладном уровне программные спецификации «начинки» этих карт могут и различаться. Так, например, в SD MP3-плееры нельзя вставлять карты MMC — они используют различные форматы данных. Данные общего назначения (не секретные) могут быть перенесены с MMC на SD, но перенос секретных данных хосты не допустят (если хост — не компьютер со взломанным ПО). Хосты MMC-карт с SD-картами несовместимы даже чисто физически (эти карты в слот для MMC не влезут по толщине).

Кроме карт SD-памяти, планируется выпуск и *карт ввода-вывода* (SD I/O card). В первую очередь рассматривается интерфейс Bluetooth, позволяющий быстро и без проводов синхронизировать данные устройств, имеющих слот SD, друг с другом и с устройствами, уже имеющими этот интерфейс.

### 9.3.4. Miniature Card

Карты Miniature Card предназначены для использования в недорогих устройствах бытовой электроники в качестве сменной флэш-памяти, а также расширения динамической памяти (рис. 9.10). В интерфейсе карт используется линейный доступ к произвольной ячейке памяти с адресуемым объемом до 64 Мбайт. Карты имеют немultipлексированную 16-разрядную шину данных и могут работать в пакетном режиме передачи данных. Также имеется возможность работы с 8-разрядным хостом (старший и младший байты шины запараллеливаются, линия BS8# заземляется). Интерфейс хоста для этих карт конфигурируется на работу в режиме DRAM или флэш-памяти; карты снабжаются микросхемой энергонезависимой памяти идентификации с интерфейсом PC. Карты имеют прорези для правильного позиционирования и коннектор, обеспечивающий «горячее» (даже для DRAM!) подключение. При установке передний край карты вводят в слот, при этом подключаются контакты «первой очереди»: GND, VCC и CINS\* (контакт, за-



мыкающийся на «землю»). Затем карта опускается вниз, прижимаются контакты основного эластомерного разъема и замыкается контакт обнаружения CD#. Карты могут работать с питанием 5 и 3,3 В; для идентификации питания служат сигналы VS1 #, VS2#. Карта Miniature Card через переходный адаптер может устанавливаться в слот PC Card типа 2.

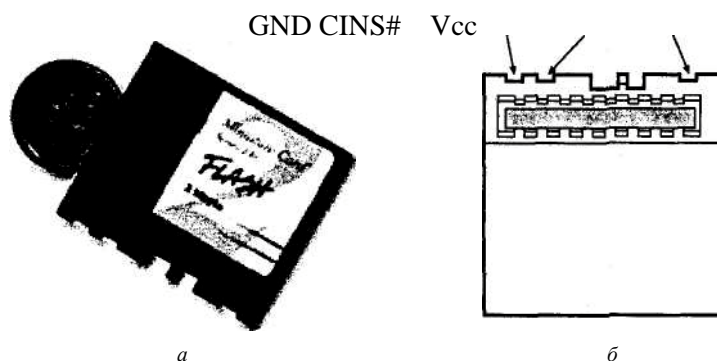


Рис. 9.10. Карта Miniature Card: а — общий вид, б — вид снизу

При работе по интерфейсу DRAM используются линии адреса  $A[12:0]$  и управляющие сигналы  $RAS\#$ ,  $CASH^*$  и  $CASL\#$  (стробы для старшего и младшего байтов), а также  $WE\#$ . Карты имеют отдельную линию питания  $V_{ee}$  для регенерации при отключенном основном питании.

При работе по интерфейсу флэш-памяти используются линии адреса  $A[24:0]$  и управляющие сигналы  $OE\#$  (чтение),  $WE\#$  (запись),  $SEN\#$  и  $CE1\_ \#$  (выборка для старшего и младшего байтов).

## 9.4. Системная поддержка устройств хранения

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 13 h 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. [1,4,9]). 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. Для работы с большими дисками введены новые функции BIOSInt 13hc 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 устройства (A:, B:, C:,..., Z:). Один *физический диск* (жесткий) может содержать несколько логических дисков. На физическом диске могут присутствовать системные области (MBR и резервированные секторы), доступ к которым операционная система не предоставляет.

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

- ◆ Возможно обращение к контроллеру дисков на уровне регистров путем задания адреса блока и количества 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.
- ◆ Сервисы BIOS Int 13h также позволяют получить 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 недоступными. Для осмысленного доступа к данным необходимо знать структуры FAT и каталогов, а 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, а также 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.

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 обеспечения 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 Int 13h.

Спецификация расширяет традиционный набор функций BIOS, она ориентирована на приводы с интерфейсом ATAPI и SCSI.

## Глава 10

# Интерфейсы компьютерных сетей

Компьютерные сети разделяются на локальные (в пределах здания или группы соседних зданий) и глобальные. В настоящее время в локальных сетях установилось практически полное господство технологии Ethernet с пропускной способностью 10, 100 и 1000 Мбит/с, не за горами и скорость 10 Гбит/с. Из множества технологий и интерфейсов глобальных подключений здесь рассмотрим лишь интерфейс обычной (аналоговой) коммутируемой телефонной сети. Именно этот интерфейс широко используется для подключения модемов. Других технологий — локальных ARCNet, Token Ring, 1 OOVG- Any LAN и беспроводных, а также глобальных касаться не будем. Как и Ethernet, они достаточно подробно описаны в литературе [3].

### 10.1. Стандарты и интерфейсы Ethernet

Технология Ethernet позволяет использовать различные среды передачи, для каждой из которой имеется стандартное название вида  $XBaseY$ , где  $X$  — скорость передачи, Мбит/с (10, 100, 1000...);  $Base$  — ключевое слово (обозначает немодулированную передачу);  $Y$  — условное обозначение среды передачи и дальности связи. Все современные версии Ethernet используют кабель «витая пара» или оптоволоконный и звездообразную топологию. *Центральным устройством* звезды может быть повторитель (он же хаб, hub) или коммутатор (switch). Возможно и двухточечное соединение двух узлов. Для старых версий на коаксиальном кабеле была характерна шинная топология, главным недостатком которой является низкая надежность всей сети. Есть и экзотический вариант пассивной оптической шины 10BaseFP. Иногда в сетях применяются *конверторы среды передачи* (media converter), преобразующие типы интерфейсов. Чаще всего используются конверторы «витой пары» в оптику, также применяют и конверторы одномодового оптоволокна в многомодовое.

Для Ethernet со скоростью 10 Мбит/с существуют следующие стандарты.

- ♦ *10Base5* — сеть на толстом коаксиальном кабеле RG-8 (50 Ом) с шинной топологией, максимальная длина кабельного сегмента — 500 м. Для подключения сетевой адаптер должен иметь интерфейс AUI, подключаемый кабелем-спуском (4 экранированные витые пары) к трансиверу, установленному на кабеле. В настоящее время для новых сетей не применяется (дорого, громоздко, неэффективно и бесперспективно).

- ◆ *10Base2* — сеть на тонком коаксиальном кабеле RG-58 (50 Ом) с шинной топологией, максимальная длина кабельного сегмента — 185 м. Для подключения сетевой адаптер должен иметь интерфейсный разъем BNC (или AUI с трансивером). Это самый дешевый (по оборудованию) вариант сети; перспектив на развитие нет.
- ◆ *WBaseT* — сеть на витой паре категории 3 и выше (2 пары проводов), длина луча — до 100 м (на кабеле категории 5 можно достигать дальности и 200 м, но это не рекомендуется). Для подключения сетевой адаптер должен иметь интерфейсный разъем RJ-45 (или АШ с трансивером). Это эффективный вариант сети начального уровня, позволяет расширять полосу пропускания заменой концентраторов-повторителей на коммутаторы. При кабельной проводке категории 5 и выше позволяет переходить на скорости 100 и даже 1000 Мбит/с (с заменой карт и концентраторов).
- ◆ *WBaseF* и *FOIRL* — сеть на оптоволоконном кабеле (пара волокон). Для подключения адаптер должен иметь интерфейс AUI, на который устанавливается оптический трансивер. Используются дешевые многомодовые трансиверы (длина волны — 850 нм) с дальностью до 1 км. Для дальних дистанций (десятки км на одномодовом волокне) используются одномодовые трансиверы (1310 нм), которые могут работать и с многомодовым волокном (до 2 км).

Для сетей *Fast Ethernet* со скоростью 100 Мбит/с существуют следующие стандарты.

- ◆ *100BaseTX* — сеть на витой паре категории 5 и выше (2 пары проводов), длина луча — до 100 м. Сетевой адаптер подключается через разъем RJ-45. Это популярный и оптимальный (цена/производительность) вариант подключения узлов к сети. При качественной кабельной проводке позволяет переходить на скорость 1000 Мбит/с (с заменой карт и концентраторов).
- ◆ *100BaseT4* — сеть на витой паре категории 3 и выше (4 пары проводов), длина луча — до 100 м. Разъем RJ-45, вариант малораспространенный.
- ◆ *100BaseFX* — сеть на оптоволоконном кабеле (пара волокон). Используются одномодовые трансиверы (1310 нм), которые могут работать и с многомодовым волокном (до 2 км). Дальность в полнодуплексном режиме — десятки км.
- ◆ *100BaseSX* — сеть на оптоволоконном кабеле с дешевыми многомодовыми трансиверами (850 нм), дальность — до 300 м. Совместима с *10BaseF*, поддерживается автосогласование режима и скорости (10/100).

Для сетей *Gigabit Ethernet* со скоростью 1000 Мбит/с существуют следующие стандарты.

- ◆ *1000BaseCX* — соединение активного оборудования коротким (до 25 м) кабелем STP или двухосевым кабелем.
- ◆ *1000BaseT* — соединение витой парой категории 5 и выше (4 пары) на расстояние до 100 м. Разъемы RJ-45.
- ◆ *1000BaseSX* — соединение по паре многомодовых волокон, дальность — 200-500 м (в зависимости от параметров волокна).
- ◆ *1000BaseLX* — соединение по паре одномодовых волокон, дальность — до 50 км (в зависимости от параметров трансиверов).

Выше были приведены ограничения на длину каждого физического соединения в сети, однако для работоспособности (надежной работы протокола разрешения коллизий) должны выполняться и дополнительные условия, подробно описанные в литературе [3]. Задача уменьшения диаметра домена коллизий решается применением коммутаторов, а преодоление коллизионных ограничений на длину каждого соединения обеспечивается переходом на полнодуплексный режим связи (в котором коллизий нет как таковых). Для 10-Мбитных сетей *Ethernet* должны соблюдаться перечисленные ниже условия.

- ◆ Для коаксиала — правило «5-4-3»: не более 5 сегментов могут соединять не более 4 повторителей, станции (адаптеры) можно подключать не более чем в 3 сегментах.
- ◆ Для витой пары (и оптики) — между любой парой узлов может быть не более 4 повторителей (хабов).
- ◆ Для любых сетей: диаметр домена коллизий — самое большое расстояние («электрическая» длина кабелей между парой узлов) — не должен превышать 5 км.
- ◆ Число узлов в домене коллизий — не более 1024 (реально их не должно быть более 30-50).

Для сетей *Fast Ethernet* ограничения жестче.

- ◆ Диаметр домена коллизий — не более 205 м.
- ◆ Число повторителей в домене коллизий — не более двух класса II, не более одного класса I.

В *Gigabit Ethernet* применяются только коммутаторы, так что действуют только ограничения на длину соединений.

Для оптических соединений применяемые разъемы разнообразны: ST, SC, MT-RJ и другие. Коаксиальные разъемы для «толстого» и «тонкого» кабелей различны (серии «N» и BNC соответственно). Отметим, что каждый сегмент коаксиала должен оканчиваться терминаторами 50 Ом и быть заземлен в одной точке. «Схемная земля» компьютера не имеет гальванической связи с экраном коаксиального разъема, так что следует избегать случайного касания BNC-разъемов с металлическими частями, связанными с шасси компьютера. Сети на коаксиале требуют правильного заземления, нарушение правил чревато выгоранием адаптеров.

Для витой пары применяются разъемы RJ-45 (рис. 10.1), назначение контактов разъема сетевого адаптера (порт MDI) приведено в табл. 10.1. Порты концентраторов ЮBaseT, 100BaseTX и 100BaseT4 имеют тип MDIX, у них сигналы TX и RX поменяны местами. Для подключения конечных узлов к портам активного оборудования (соединение портов MDI-MDIX, рис. 10.2, а) используется «прямой» кабель (рис. 10.3, а), для непосредственного соединения адаптеров (MDI-MDI, рис. 10.2, б) или соединения двух коммуникационных устройств (MDIX-MDIX) применяют «перекрестный» кабель (рис. 10.3, б). В коммуникационных устройствах, как правило, один из портов снабжают переключателем MDI-MDIX или дополнительным разъемом.

Таблица 10.1. Разъем RJ-45 адаптера Ethernet

Контакт	10BaseT/100BaseTX	100BaseT4	1000BaseTX
1	Tx+	Tx_D1+	BD1+
2	Tx-	Tx_D1-	BD1-
3	RX+	Rx_D2+	BD2+
4	Не подключен	BI_D3+	BI_D3+
5	Не подключен	BI_D3-	BI_D3-
6	Rx-	Rx_D2-	BI_D2-
7	Не подключен	BI_D4+	BI_D4+
8	Не подключен	BI_D4-	BI_D4-

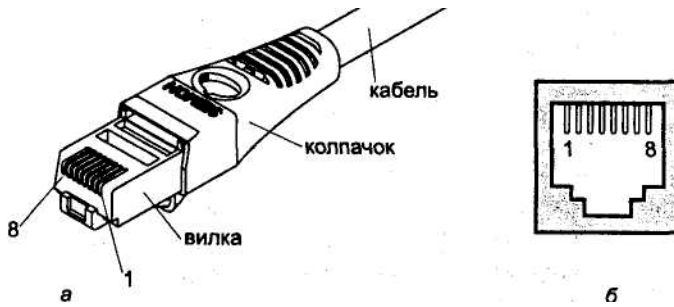


Рис. 10.1. Разъем RJ-45: а — вилка, б — розетка

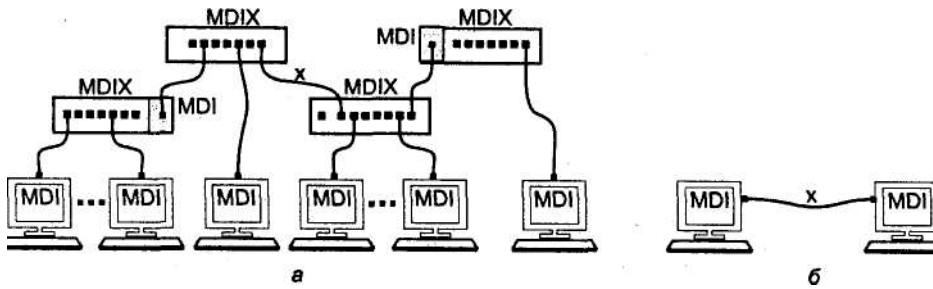


Рис. 10.2. Сеть 10BaseT/100BaseTX: а — звезда, б — двухточечное соединение

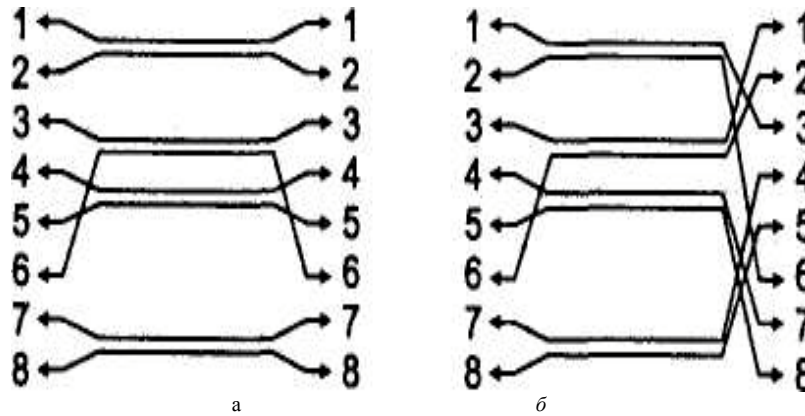


Рис. 10.3. Интерфейсные кабели Ethernet: а — «прямой», б — «перекрестный»

В локальных сетях обычно используется кабельная проводка, состоящая из стационарных кабелей, оканчивающихся розетками, и коммутационных шнуров. Стационарную проводку выполняют так, что она обеспечивает «прямое» соединение контактов своих интерфейсных разъемов. Коммутационные шнуры могут быть как «прямыми», так и «перекрестными». Заметим, что связи контактов 4, 5, 7 и 8 требуются только в 100BaseT4 и 1000BaseTX, но для 10BaseT и 100BaseTX они не мешают, так что кабельное хозяйство едино.

В Gigabit Ethernet 1000BaseTX применяются только «прямые» кабели. Универсальные порты совместимы с Fast Ethernet (поддерживают автосогласование). Если два порта Gigabit Ethernet соединить «перекрестным» кабелем, они свяжутся в режиме 100BaseTX.

Для приведенных выше реализаций Ethernet на витой паре предусмотрен *протокол согласования режимов* (autonegotiation), который выполняется каждый раз при установлении соединения после физического подключения и (или) инициализации портов. Протокол основан на обмене служебными импульсами (они отличны от кадров передаваемой информации). Этот протокол позволяет соединяемым портам выбрать самый эффективный из режимов, доступных обоим портам. Приоритеты режимов в порядке убывания: 1000BaseT, 100BaseTX полнодуплексный, 100BaseT4, 100BaseTX полудуплексный, 10BaseT полнодуплексный, 10BaseT полудуплексный. Протокол автоматического согласования может быть отключен (или не реализован), в этом случае режим работы задается принудительно при конфигурировании порта. Возможность переключения режимов отражается в названиях портов (например, Fast Ethernet 10/100), поддержка режима 100BaseT4 встречается нечасто.

Для оптических вариантов тоже появился протокол согласования, но его возможности ограничиваются из-за вероятного несовпадения длин волн, используемых в разных вариантах. Правда, здесь автосогласование не так уж и необходимо, поскольку оптических соединений гораздо меньше, их тщательно планируют и не так уж часто реконфигурируют.

В стандарте Ethernet (10 Мбит/с) определен интерфейс AUI (Attachment Unit Interface — интерфейс устройства подключения), с помощью которого к адаптеру можно подключать трансивер (приемопередатчик) для любой среды передачи. В трансивере располагаются оконечные цепи передатчика, приемника и детектор коллизий. Назначение контактов интерфейса AUI приведено в табл. 10.2, здесь используется разъем DB-15 (розетка на адаптере, вилка на трансивере).

**Таблица 10.2.** Разъем AUI -интерфейса

Ethernet Контакт	Сигнал
1	Collision
2	Collision +
3	Transmit +
4	Receive (экран)

*продолжение ^*



Таблица 10 .2 (продолжение)

Контакт	Сигнал
5	Receive +
6	DC Power GND
7	Не подключен
8	Не подключен
9	Collision -
10	Transmit -
11	Transmit (экран)
12	Receive -
13	DC Power (+12В)
14	DC Power (экран)
15	Не подключен

В стандарте на Fast Ethernet фигурирует интерфейс МП (Media Independent Interface — интерфейс, независимый от среды передачи). В МП данные для приемника и передатчика передаются в некодированном виде по 4-битным параллельным шинам (с частотой синхронизации 2,5 и 25 МГц для скоростей 10 и 100 Мбит соответственно) или в последовательном коде (для 10 Мбит/с). В интерфейсе имеются сигналы синхронизации и управления приемником и передатчиком, состояния линии (наличие несущей, коллизия), а также последовательный интерфейс управления SMI (см. п. 11.2), по которому можно общаться с управляющими регистрами трансивера. Определен и физический разъем для подключения сменных модулей (40-контактный двухрядный), но в ПК он практически не встречается.

## 10.2. Сетевые адаптеры

Интерфейсы локальных сетей в ПК обеспечивают *сетевые адаптеры*, или *сетевые интерфейсные карты* (Network Interface Card, NIC). Адаптеры имеют передающую и принимающую части, которые в случае поддержки полного дуплекса должны быть независимы друг от друга. Задача передающей части: по получении со стороны центрального процессора (ЦП) блока данных и адреса назначения для передачи получить доступ к среде передачи, сформировать и передать кадр (добавить преамбулу, CRC-код), делая повторные попытки в случае обнаружения коллизий. Адаптер должен сообщить процессору об успехе или невозможности передачи. Приемная часть, просматривая заголовки всех кадров, проходящих в линии, «выуживает» из этого потока кадры, адресованные данному узлу уникальным, ширококвотельным или групповым способом. Адаптер можно программно настроить и на «неразборчивый» режим (promiscuous mode), в котором он будет принимать все кадры без разбора. Кадры принимаются в буфер и проверяются на отсутствие ошибок (длина кадра, корректность CRC). О приеме корректных кадров уведомляется центральный процессор и организуется передача кадра из локального буфера адаптера в системную память компьютера. Ошибочные кадры, как правило,

игнорируются, хотя адаптер может собирать статистику их появления. На практике попадаются и адаптеры, не обнаруживающие ошибок в поврежденных кадрах. Диагностика сети с таким адаптером не проста.

Сетевые адаптеры для PC выпускаются для шин ISA, EISA, MCA, VLB, PCI, PC Card. Существуют адаптеры, подключаемые к стандартному LPT-порту PC; их преимущество — отсутствие потребностей в системных ресурсах (порты, прерывания и т. п.) и легкость подключения (без вскрытия компьютеров), недостаток — при обмене они значительно загружают процессор и не обеспечивают высокой скорости передачи («потолок» — 10 Мбит/с). Есть адаптеры и для шины USB. Сетевые адаптеры интегрируются и в некоторые модели системных плат.

Эффективная скорость обмена данными по сети очень сильно зависит от архитектуры сетевых адаптеров и, при прочих равных условиях, от скорости передачи данных между локальной памятью адаптера и системной памятью компьютера, а также от возможности параллельного выполнения нескольких операций. В качестве «средств доставки» используются каналы прямого доступа к памяти (DMA), программный ввод-вывод (PIO), прямое управление шиной. Стандартные 8-битные каналы прямого доступа шины ISA способны развивать скорость до 2 Мбайт/с, 16-битные — до 4 Мбайт/с. Кадр максимальной длины (1514 байт) они передают примерно за 1,3 или 2,6 мс соответственно. По сравнению с 12 мс, требуемыми для передачи кадра в среде Ethernet, это время относительно невелико. Однако для Fast Ethernet, где тот же кадр в среде передается за 1,2 мс, такая транспортировка оказывается слишком медленной. Более высокую скорость обмена с буфером адаптера обеспечивает режим программного ввода-вывода (PIO), но он полностью загружает центральный процессор на время передачи. Более эффективны интеллектуальные адаптеры с прямым управлением шиной (bus mastering) ISA/ EISA, сочетающие относительно высокую скорость (до 8 Мбайт/с ISA 16 бит и до 33 Мбайт/с EISA). Однако для скорости 100 Мбит/с производительности шины ISA уже недостаточно. На сегодняшний день широко применяются адаптеры шины PCI, где для 32-разрядного интерфейса при частоте 33 МГц пропускная способность достигает 132 Мбайт/с. Но для технологии Gigabit Ethernet и этого только-только хватает, правда, у PCI есть резервы: переход на частоту 66 МГц и разрядность 64 бит, что позволяют далеко не все системные платы. Особенно эффективны активные адаптеры, имеющие собственный процессор, для шины PCI. Они выполняют передачи на полной скорости PCI, практически не загружая центральный процессор. Это свойство особенно важно для серверов. Параллельное выполнение операций подразумевает поддержку полного дуплекса — полную независимость принимающей и передающей частей, а также возможность одновременного выполнения приема кадра в буфер, передачи другого кадра и обмена данными между буферной памятью адаптера и системной памятью компьютера. На производительность адаптера для ISA/EISA влияет и объем буферной памяти: при ограниченной (по сравнению со скоростью в линии) пропускной способности шины применяют буферную память объемом до 64 Кбайт, которую делят между передатчиком и приемником либо поровну, либо с преимуществом для передатчика. Для шины PCI при эффективных средствах доставки (интеллектуальное прямое управление шиной) для скорости 100 Мбит/с большой буфер не нужен —

достаточно по 2 Кбайт на приемник и передатчик. Однако адаптеры Gigabit Ethernet опять снабжают буфером значительного размера (256 Кбайт).

Адаптеры можно разделить на две группы — адаптеры для рабочих станций и адаптеры для серверов. Деление условно — адаптеры для рабочих станций могут иметь черты, относящиеся к серверным. Использовать простые карты в серверах не стоит — они могут стать узким местом сети и «пожирателями» ресурсов ЦП.

*Адаптеры для рабочих станций* проще и дешевле — для них не требуется (пока?) скорости выше 100 Мбит/с, полный дуплекс встречается редко, к использованию процессорного времени особо жестких требований не предъявляют. Долгие годы широко применяются адаптеры, программно совместимые с картами NE2000 — 16-битными неинтеллектуальными картами для шины ISA разработки Novell-Eagle. Совместимость с этой моделью имеют и ряд карт для шины PCI. Наиболее удобны и популярны двухскоростные карты 10/100 Мбит/с — для их подключения в современных сетях легко найти оптимальное место. Карты обычно имеют колодку для установки Boot ROM, современные модели часто обеспечивают возможность «пробуждения» по сети (remote wake up), поддерживают интерфейс DMI и ACPI. Для этого они имеют специальный дополнительный 3-проводный интерфейс — кабель с коннектором, подключаемый к системной плате. По этому кабелю системная плата с питанием в стандарте ATX подает дежурное напряжение (линия +5VSB), даже когда основное питание на системную плату и все устройства не подается. От этой линии питается «дежурная» принимающая схема, которая настроена на прием кадра специфического формата (Magic Packet) по сетевому интерфейсу. По приему этого кадра сетевой адаптер через кабель подает пробуждающий сигнал PME на системную плату, которая дает сигнал на включение блока питания; компьютер включается и загружается ОС с поддержкой DMI. Теперь администратор может выполнить все запланированные действия, а по окончании ОС на компьютере, завершая свою работу, выключает питание.

*Адаптеры для серверов* должны иметь высокопроизводительную шину — сейчас используют PCI32/64 бит 33/66 МГц, раньше в серверах часто применяли шину EISA или MCA. Для серверных карт критична загрузка ЦП при обмене данными, поэтому эти карты наделяют интеллектом для прямого управления шиной и параллельной работы узлов адаптера. Полнодуплексные адаптеры должны поддерживать управление потоком по 802.3x. Ряд совершенных моделей поддерживают приоритизацию трафика по 802.1p, фильтрацию многоадресного трафика, поддержку ВЛС с маркированными кадрами (tagged VLAN), Fast IP, аппаратный подсчет контрольных сумм IP-пакетов. Поддержка ВЛС позволяет серверу, подключенному одной линией к коммутатору, быть членом нескольких ВЛС, определенных на всей локальной сети. Для повышения надежности серверные карты могут поддерживать резервирование линий (Resilient Link) — резервный адаптер и линия связи заменяют основной канал в случае его отказа. При этом резервному адаптеру присваивается MAC-адрес основного, чтобы сеть «не заметила» подмены. Резервирование линий должно поддерживаться программными драйверами, чтобы замена происходила прозрачно и для серверных приложений. «Самоизлечающиеся» драйверы (Self-Healing Drivers) в случае обнаружения проблем функ-

ционирования («зависании») могут автоматически выполнить сброс и повторную инициализацию адаптера. Удаленная загрузка и пробуждение по сети серверам, как правило, не требуется. Адаптеры (совместно с драйверами) могут поддерживать SNMP и RMON. Для серверов выпускаются и многопортовые (как правило, на 4 порта) адаптеры, конфигурируемые как для отдельного независимого использования, так и для резервирования друг друга. Такие карты позволяют экономить слоты PCI (для шины EISA проблема экономии слотов не была острой). Типовая скорость для серверных карт на сегодняшний день — 100 Мбит/с, производительность Gigabit Ethernet может быть востребована лишь очень мощными серверами.

Адаптер может иметь один или несколько интерфейсных разъемов:

- ◆ BNC — коаксиальный разъем для подключения к сегменту сети 10Base2;
- ◆ AUI — розетка DB-15 для подключения внешних адаптеров (трансиверов) 10BaseS, 10Base2, 10BaseT, 10BaseF, FOIRL;
- ◆ RJ-45 — 8-контактное гнездо для подключения кабелем «витая пара» к концентратору (хабу или коммутатору) 10BaseT, 100BaseTX и/или 100BaseT4;
- ◆ SC (пара), иногда ST — оптические разъемы для подключения к концентраторам 100BaseFX, 100BaseSX, 100BaseLX.

Для 10-мегабитных адаптеров характерны сочетания BNC+AUI или RJ-45+AUI, наиболее универсальные «Combo» имеют полный 10-мегабитный набор BNC/AUI/ RJ 45. Первые модели карт на 10 и 100 Мбит/с имели пару разъемов RJ-45 — каждый для своей скорости. При наличии нескольких разных разъемов (например, BNC и RJ-45) одновременно они не используются — адаптер не может работать в качестве повторителя. Большинство современных адаптеров имеют один разъем RJ-45 и поддерживают два стандарта — 10BaseT и 100BaseTX. Многопортовые серверные карты имеют несколько независимых адаптеров, каждый со своим интерфейсом.

Интерфейсные карты потребляют *системные ресурсы* компьютера.

- ◆ *Пространство ввода-вывода* — как правило, 4-32 смежных адреса из области, адресуемой 10-битным (для шины ISA) или 16-битным (EISA, PCI) адресом.  
Используется для обращения к регистрам адаптера при инициализации, текущем управлении, опросе состояния и передаче данных.
- ◆ *Запрос прерывания* — одна линия (IRQ3,5,7,9,10,11,12 или 15), возбуждаемая по приему кадра, адресованного данному узлу, а также по окончании передачи кадра (успешной или безуспешной из-за коллизий). Без прерываний сетевые карты работать не могут, при некорректном назначении обращения к сети «зависают».
- ◆ *Канал прямого доступа к памяти (DMA)* используется в некоторых картах ISA/EISA; для прямого управления (bus mastering) шины ISA пригодны только 16-битные каналы 5-7.
- ◆ *Разделяемая память (adapter RAM)* адаптера — буфер для передаваемых и принимаемых кадров — для карт ISA обычно приписывается к области верхней памяти (UMA), лежащей в диапазоне A0000h-FFFFFh. Карты PCI могут располагаться в любом месте адресного пространства, не занятого оперативной памятью компьютера. Разделяемую память используют не все модели карт.

- ◆ *Постоянная память* (adapter ROM) — область адресов для модулей расширенного ROM BIOS, 4/8/16/32 Кбайт в диапазоне C0000-DFFFFh. Используется для установки ПЗУ удаленной загрузки (Boot ROM) и антивирусной защиты.

*Под конфигурированием адаптера* подразумевается настройка на использование системных ресурсов PC и выбор среды передачи. Конфигурирование, в зависимости от модели карты, может осуществляться разными способами.

- ◆ С помощью переключателей (джамперов), установленных на карте. Используются на адаптерах первых поколений шины ISA. Для выбора каждого ресурса, а также среды передачи имеется свой блок джамперов.
- ◆ С помощью энергонезависимой памяти конфигурации (NVRAM, EEPROM), установленной на карте с шиной ISA. Эти карты не имеют джамперов (jumperless), но конфигурируются вручную. Для конфигурирования требуется специальная утилита, специфическая для конкретной модели (семейства) карт.
- ◆ С помощью энергонезависимой памяти конфигурации, установленной на карте с шиной EISA или MCA, и системной памяти конфигурирования устройств (ESCD для EISA). Конфигурирование ресурсов осуществляется пользователем с помощью системной утилиты ECU (EISA Configuration Utility) для шины EISA.
- ◆ Автоматическое — PnP для шин ISA и PCI. Распределение ресурсов осуществляется на этапе загрузки ОС.

Выбор среды и скорости передачи может быть ручным (программным) или автоматическим. В ряде случаев имеет смысл делать явные назначения, чтобы избежать сбоев PnP из-за излишней автоматизации. Эти сбои, как правило, порождаются недостаточной согласованностью адаптеров и их драйверов. При этом драйвер не может правильно распознать установленный режим и воспользоваться его преимуществами. Автоматическая настройка вносит дополнительные задержки в процесс инициализации (при загрузке) и не со всяким сетевым оборудованием работает корректно. Для некоторых моделей карт с интерфейсом 10Base2 (BNC-разъем) предлагается расширенный режим, увеличивающий дальность связи до 305 м против штатных 185. При необходимости длинных сегментов этим режимом можно воспользоваться, но при условии, что он имеется и включен во всех картах данного сегмента. В утилитах конфигурирования могут предлагаться и дополнительные настройки — оптимизация для клиента или сервера, поддержка модема и некоторые другие. Их установка должна соответствовать конкретному применению.

### 10.3. Интерфейс коммутируемой телефонной линии и телефонного аппарата

*Коммутируемые линии* (dial-up lines) являются наиболее распространенным внешним интерфейсом телефонной сети, к которому компьютер подключается с помощью модема. Эти линии являются аналоговыми, по одной двухпроводной линии передаются сигналы в обоих направлениях (прием и передача). Коммутируемые линии обеспечивают прохождение сигнала (речевого или данных через модем)

в полосе частот шириной 3,1 кГц (300-3400 Гц), а также поддерживают сигнализацию, принятую для аналоговой телефонии. Коммутируемые линии позволяют устанавливать временные соединения с любыми абонентами телефонной сети в пределах возможностей, предоставляемых АТС, к которой произведено подключение. В качестве соединителей при подключении абонента в нашей стране широко используются 4-контактные разъемы, изображенные на рис. 10.4, а. Для некоторых специфических случаев применения эти разъемы могут иметь размыкатели и дополнительный конденсатор. Как правило, используется только одна пара контактов. Сейчас их вытесняют соединители международного стандарта RJ-11 — 6-позиционный модульный коннектор с четырьмя контактами, из которых задействуется только средняя пара (рис. 10.4, б).

#### ВНИМАНИЕ

Телефонные линии находятся под напряжением оборудования АТС. Все подключаемые к ним устройства должны иметь полную гальваническую развязку от схемной земли компьютера и иных устройств. Для развязки используют трансформаторы и оптроны. Изоляция должна выдерживать напряжение 1-1,5 кВ. Цепи, подключаемые к линии, должны иметь защиту от перенапряжений (особенно актуальную при подключении к воздушным телефонным линиям).



Рис. 10.4. Розетки для телефонных линий: а — стандарт бывшего СЭВ, б — международный стандарт

Упрощенная схема стандартной нагрузки линии — телефонного аппарата — приведена на рис. 10.5. В *состоянии покоя* (idle) аппаратура АТС посылает в линию напряжение постоянного тока 60 В через ограничительные резисторы с суммарным сопротивлением 700-1500 Ом и следит за током в линии. При *опущенной трубке* (on-hook) КТ переключается и подключает через конденсатор вызывное устройство (Зв), при этом телефон не нагружает линию по постоянному току. Для *вызова абонента* АТС посылает серию импульсов амплитудой около 120 В с частотой 25 Гц. Эти импульсы через конденсатор проходят в обмотку звонка и вызывают колебания молоточка. При *снятой трубке* (off-hook) к линии подключается разговорный узел (это положение показано на рисунке), его сопротивление постоянному току — около 150-600 Ом. *Разговорный узел* содержит микрофон, телефон и схему подавления прослушивания сигнала собственного микрофона. Для предотвращения травм органов слуха предусмотрен ограничитель напряжения звукового сигнала. Подключение разговорного узла приводит к протеканию постоянного тока в линии, что позволяет станции фиксировать факт снятия трубки. При снятии трубки станция посылает непрерывный тональный сигнал ответа (425 Гц) и готовится принять сигналы набора номера. В это время (а также во время разго-

вора после установления соединения) на нагрузке телефона падение напряжения составляет около 5-15 В постоянного тока, и на фоне этого уровня переменная составляющая (звуковой сигнал разговора) имеет амплитуду порядка десятков сотен милливольт. *Отбой* (вешание трубки, on-hook) сигнализируется разрывом цепи для постоянного тока. На данной схеме показаны цепи *импульсного набора номера (pulse dialing)*. При наборе номера (трубка снята) разговорный узел отключается контактом КН-1. Во время «взвода» диска контакт КН-2 замыкается, во время обратного хода он п раз замыкается (по одному разрыву на единицу набираемой цифры, 0—10 разрывов). После окончания набора цифры контакты КН-1 снова подключают разговорный узел. Длительность (60 мс) и частота ( $10 \pm 1$  имп/с) разрывов стандартизованы и рассчитаны на время срабатывания шаговых искателей «древних» станций. Импульсный набор ради совместимости со старыми телефонами поддерживается всеми АТС. О результате коммутации станция сигнализирует короткими гудками «занято» (busy) — тональный сигнал 425 Гц, длительность посылки и паузы 0,35 с или длинными гудками (1с посылка, 3 с пауза), синхронными не всегда с сигналами, вызывающими абонента. Временная диаграмма сигналов на внешних контактах телефонного аппарата в разных стадиях показана на рис. 10.6.

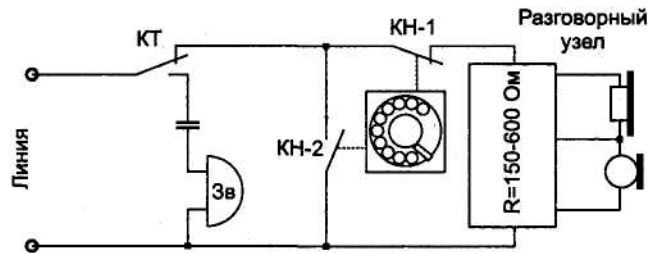


Рис. 10.5. Упрощенная схема телефонного аппарата (трубка снята)

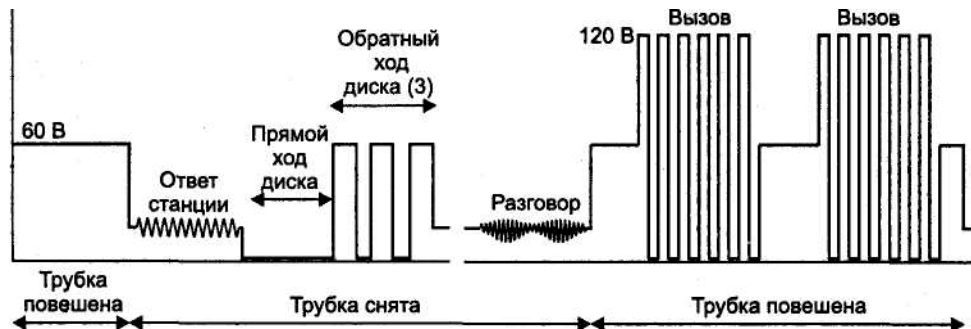


Рис. 10.6. Временная диаграмма работы телефонного аппарата с импульсным набором

При *тональном наборе (tone dialing)* каждая цифра кодируется парой из восьми тональных частот звукового диапазона, передаваемых телефоном в линию. Допустимы 16 комбинаций, которые позволяют кодировать 10 цифр и дополнитель-

ные символы (кнопки #, \*) для получения дополнительных услуг от АТС. Тональный набор выполняется гораздо быстрее (один символ передается за 40 мс) и надежнее, но поддерживается еще не всеми АТС. Возможность тонального набора имеют большинство моделей «кнопочных» телефонов, но для совместимости со старыми АТС они позволяют переключаться на импульсный набор. Тональный набор позволяет вводить и дополнительные цифры для выбора одного из абонентов мини-АТС, подключенной к обычной телефонной линии. Дополнительные цифры набираются вызывающим абонентом после установления соединения с мини-АТС по обычному номеру, после специального приглашения автоответчика мини-АТС.

Для *блокированных (спаренных) телефонов* одна двухпроводная линия от АТС обслуживает двух абонентов с разными номерами поочередно. Для разделения линий используется пассивное *устройство-блокиратор* — два однополупериодных выпрямителя сигнала с линии от АТС. Здесь АТС задействует положительную полярность сигнала для работы с абонентом 1 и отрицательную — для работы с абонентом 2. Когда приходит внешний вызов, его сигнал станция посылает в полярности, соответствующей вызываемому абоненту. Для опроса состояния (поднята/опущена трубка) АТС чередует полярность опрашивающего напряжения с низкой частотой (этот рокот можно услышать, если в момент плавного снятия трубки контакты включения трубки сработают раньше, чем замыкающие контакты нагрузочного резистора). Как только обнаружена снятая трубка у одного из абонентов, станция останавливает чередование опроса на «его» полярности и работает вышеописанная сигнализация. Для спаривания телефонов применяют и частотное разделение: для одного из абонентов сигнал передается на высокой частоте (с модуляцией). Блокирование (спаривание) телефонов позволяет экономить физические линии, но имеет ряд очевидных неудобств: абоненты могут пользоваться телефонами лишь поочередно, «захватывая» линию на время до «отбоя». Для внешних звонков к абоненту линия представляется занятой как во время его разговора, так и во время разговора соседа. Сблокированные линии «не любят» некоторые модели модемов.

Для *автоматического определения номера (АОН)* вызывающего абонента имеется специальная система сигнализации, поддерживаемая большинством отечественных АТС. Ее работу иллюстрирует диаграмма, показанная на рис. 10.7. Когда телефон с АОН получает вызывной сигнал, он до подключения разговорного узла включает цепь, на которой падает напряжение около 24 В, и через 250-275 мс в линию посылает тональный сигнал «запрос». Этот сигнал должен иметь частоту 500 Гц и длительность 100 мс. В ответ на этот сигнал АТС передает несколько раз повторяющийся пакет из десяти двухчастотных посылок, в котором сообщается категория и 7-значный номер вызываемого абонента. Каждая посылка продолжается 40 мс и содержит две из шести возможных тональных частот (700, 900, 1100, 1300, 1500 и 1700 Гц). Таким образом кодируются цифры 0-9 и служебные символы «начало» и «повтор». Символ «начало» обрамляет пакет, символ «повтор» используется, если последующая цифра повторяет предыдущую (чтобы облегчить выделение посылок). После приема ответа АОН может имитировать (для вызы-



вающего абонента) длинные гудки, в это время вызываемый абонент может решить, снимать трубку или нет, включить автоответчик и т. п. На время разговора (или работы автоответчика) включается РУ с обычным сопротивлением, и напряжение падает до 5-15 В. Отбой выполняется обычным образом. Для зарубежных АТС аналогом услуги АОН (которая первоначально не предназначалась для широкого круга пользователей) является услуга идентификации вызывающего абонента (Caller Id), предоставляемая только цифровыми станциями. Она работает по совсем иному протоколу.



Рис. 10.7. Последовательность сигналов при автоматическом определении номера

Современные кнопочные телефоны, а также модемы и факс-модемы обрабатывают телефонную сигнализацию по вышеописанной схеме. Коммутирующими элементами у них могут быть электронные ключи или малогабаритные реле (их щелчки на модемах хорошо слышны при работе), вместо электромагнитного звонка применяется электронный генератор сигнала, традиционный угольный микрофон может заменяться электретным с усилителем. Схемотехнически они могут заметно отличаться от вышеприведенной схемы, но всяко должны обеспечивать сопротивление постоянному току при повешенной трубке (и в момент прерывания при наборе номера) не менее 250 кОм, во время набора номера (в фазе замыкания) — не более 50 Ом.

Параметры сигналов зарубежных АТС и выпускаемых для них телефонов несколько отличаются от отечественных: номинальное напряжение — 40 В, вызывные импульсы — до 90 В, уровень звукового сигнала при разговоре ниже. Из-за этих различий, например, абонент отечественного телефона хуже слышит абонента импортного телефона, а абонент импортного телефона слышит первого лучше. Из-за слишком большой амплитуды вызывного напряжения (на некоторых АТС она может достигать 200 В) аппараты могут выходить из строя. Защита линии от перенапряжения с помощью стабилизаторов или варисторов в этом случае не позволит дозвониться до «защищенного» абонента: при большой амплитуде импульсов ток потечет через ограничитель и станция зафиксирует это как снятие трубки. Далее подается обычное напряжение (ниже ограничения), ток прекращается и станция фиксирует отбой. Выходом в такой ситуации может быть применение схемы «стабилизатора вызывного напряжения» (рис. 10.8), с помощью которой автор сумел подключить модем с защитой от перенапряжений к линии от «ретивой»

АТС. Любопытным оказался факт, что ограничитель напряжения был спрятан в кабеле, прилагавшемся модему (с виду — обычный шнур с вилками RJ-11).  
 Схема не претендует на оптимальность решения, но работает.

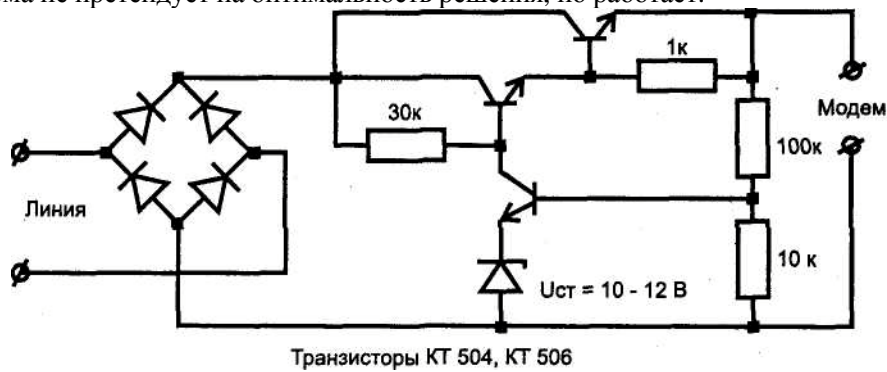


Рис. 10.8. Стабилизатор вызывного напряжения

На линиях отечественных АТС рекомендуется использовать оборудование, специально адаптированное для нашей страны (и сертифицированное Министерством связи).

## Глава 11

# Вспомогательные последовательные интерфейсы и шины

В этой главе рассматриваются интерфейсы и шины, предназначенные, в основном, для «внутреннего использования». В этом качестве PC используется для идентификации модулей DIMM, информация о которых хранится в маленьких микросхемах энергонезависимой памяти. В ряде современных системных плат присутствует шина SMBus, основанная на том же интерфейсе PC. Эта шина используется для считывания идентификационной информации модулей памяти, по ней же осуществляется доступ к памяти идентификаторов и средствам термоконтроля процессоров Xeon. Она же входит и в состав сигналов слота CNR (слот подключения расширений аудиокодека и телекоммуникаций) для конфигурирования аудио- и коммуникационного оборудования. По интерфейсу PC, входящему в интерфейс VESA DDC1/2B, современные мониторы обмениваются конфигурационной и управляющей информацией с графическим адаптером (а через него и с центральным процессором). Канал DDC входит в обычный 15-контактный аналоговый интерфейс VGA, его развитие EVC и цифровые интерфейсы P&D, DVI и DFP. А при поддержке графическим адаптером и монитором интерфейса DDC1/2AB пользователь теоретически получает возможность внешнего подключения дополнительных устройств по шине ACCESS.Bus через разъем, расположенный на мониторе (практически таких мониторов встречать не доводилось). Этими устройствами могут быть устройства позиционирования (планшеты, мыши), считыватели карт, штрих-кодов и т. п. Последовательная шина ACCESS.Bus, основанная на PC, разработана фирмой DEC для взаимодействия компьютера с его аксессуарами — например, монитором (канал VESA DDC), интеллектуальными источниками питания (Smart Battery) и т. п. С помощью интерфейса PC можно загружать программы (firmware) в энергонезависимую память (флэш) ряда популярных микроконтроллеров. Интерфейс PC обеспечивает скорость передачи данных до 100 и даже до 400 Кбит/с, при этом он гораздо проще и дешевле интерфейса RS-232C с его «потолком» 115 Кбит/с и позволяет легко подключать несколько устройств, с поддержкой «горячего» подключения/отключения и технологии PnP. Недавно в спецификацию PC была введена высокая скорость передачи, до 3,4 Мбит/с, но на такой скорости могут работать лишь новые микросхемы со специальной аппаратной поддержкой интерфейса.

Интерфейс SMI в явном виде в ПК встречается нечасто, он «родом» из коммуникационной аппаратуры Fast Ethernet, где широко используется для управления модулями физического уровня (в том числе и сменными модулями концентраторов). Интерфейсы SPI и JTAG встречаются в аппаратуре, основанной на микроконтроллерах и конфигурируемой логике, — наиболее часто они используются для загрузки конфигурационной информации (и кодов программ). Для тех же целей применяются и другие трехпроводные и четырехпроводные интерфейсы, но здесь мы ограничимся лишь упоминанием об их существовании.

## 11.1. Последовательные шины на базе I<sup>2</sup>C

Интерфейс последовательной шины PC, введенной фирмой Philips как простое и дешевое средство сопряжения микросхем бытовой электроники, стал фактическим промышленным стандартом для устройств различного назначения. Он очень удобен для обмена небольшими объемами данных, например, для конфигурации различных устройств. Спецификация шины PC определяет протокол двусторонней передачи данных по двум сигнальным линиям. Приложения этого протокола могут быть самыми разнообразными, информационная «начинка» зависит от конкретных применений. На основе интерфейса PC построены шины ACCESS.bus и SMBus, рассмотренные ниже (о работе интерфейса PC с микросхемами памяти см. п. 7.3.3).

### 11.1.1. Шина I<sup>2</sup>C

Шина *Inter IC Bus* (шина соединения микросхем), или, кратко, PC, — синхронная последовательная шина, обеспечивающая двустороннюю передачу данных между подключенными устройствами. Шина ориентирована на 8-битные передачи. Передача данных может быть как одноадресной, к выбранному устройству, так и широкоадресной. Уровни сигналов — стандартные, совместимые с широко распространенной логикой TTL, КМОП, N-МОП, как с традиционным питанием +5 В, так и с низковольтным (3,3 В и ниже). Микросхемы с интерфейсом PC, как правило, имеют аппаратную поддержку протокольных функций. Протокол позволяет взаимодействовать на одной шине устройствам с различным быстродействием интерфейса. Требования к временным параметрам сигналов весьма свободные, так что на компьютерах и микроконтроллерах, не имеющих аппаратной поддержки шины PC, ее протокол может быть реализован даже чисто программно.

Шина PC используется уже давно, ее официальная *версия 1.0* вышла в 1992 г. По сравнению с предшествующими (черновыми) версиями, здесь отсутствует (как запутанная и неиспользуемая) возможность программного задания адреса ведомого устройства. Также отсутствует низкоскоростной режим (Low speed), являющийся частным случаем *стандартного режима* — *Standard Mode (5)* — со скоростью 0-100 Кбит/с. В версии 1.0 появились определение *быстрого режима* — *Fast Mode (F)* — со скоростью 0-400 Кбит/с и связанные с ним изменения требований к форме сигнала и фильтрации помех. Также здесь веден режим 10-битной адресации устройств. *Версия 2.0* вышла в 1998 г., когда интерфейс PC стал фактиче-

ски промышленным стандартом, используемым в большом числе различных ИС. Здесь появился новый *высокоскоростной режим* — *High speed (Hs)*, — в котором скорость передачи может достигать 3,4 Мбит/с. Препятствия *FH* Алогически работают одинаково, и для них используют обобщенное обозначение *F/S*. В этой версии пересмотрены требования к уровням и форме сигналов с учетом высоких скоростей и возможности подключения низковольтных устройств с питанием 2 В и ниже. В *версии 2.1* (2000 г.) уточнены некоторые моменты, касающиеся временных диаграмм в режиме *Hs*. Приведенная здесь информация основана на спецификации шины PC версии 2.1, доступной на сайте [www.philips.com](http://www.philips.com). Параметры интерфейсных сигналов приводятся в п. 11.1.4, где они сопоставляются с требованиями SMBus и ACCESS.Bus.

Интерфейс PC использует две сигнальные линии: данных SDA (Serial Data) и синхронизации SCL (Serial Clock). В обменах участвуют два устройства — *ведущее (master)* и *ведомое (slave)*. Ведущее и ведомое устройства могут выступать в роли и *передатчика*, и *приемника* данных. Протокол допускает наличие на шине нескольких ведущих устройств и имеет простой механизм арбитража (разрешения коллизий).

Протокол обмена для обычных устройств *F/S* иллюстрирует рис. 11.1. Обе сигнальные линии имеют нагрузочные резисторы, «подтягивающие» их уровень к напряжению питания. На устройстве к каждой линии подключен приемник и передатчик типа «открытый коллектор» («открытый сток»), у ведомого устройства передатчик на линии SCL не обязателен. Все одноименные передатчики соединяются по схеме «Монтажное И»: уровень в линии будет высоким, если все передатчики пассивны, и низким, если хоть у одного передатчика выходной транзистор открыт. В покое (*Idle*, исходное состояние шины) все передатчики пассивны. Синхронизацию задает ведущее устройство, но ведомое, если оно не имеет достаточного быстродействия, может замедлять обмен данными.

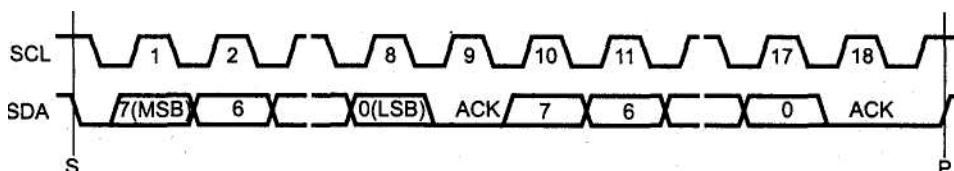


Рис. 11.1. Протокол передачи данных I<sup>2</sup>C

Начало любой передачи — условие *Start* — инициируется ведущим устройством, убедившись в том, что шина свободна (высокий уровень сигналов SCL и SDA). Условие *Start* (на диаграммах обозначается как S) — перевод сигнала SDA из высокого в низкий при высоком уровне SCL. Завершается операция переводом сигнала SDA из низкого уровня в высокий при высоком уровне SCL — условие *Stop* (обозначается как P), также вводящееся ведущим устройством. При передаче данных состояние линии SDA может изменяться только при низком уровне SCL, биты данных считаются действительными во время высокого уровня SCL. Ведущее устройство может начать очередную передачу вслед за текущей, не вводя условие

*Stop*, — это называется *repeated Start* (повторный старт, обозначаемый S г). В протоколе условия S и Sg почти равнозначны. Каждая посылка данных состоит из 8 бит данных, формируемых передатчиком (старший бит — MSB — передается первым), после чего передатчик на один такт освобождает линию данных для получения подтверждения. Приемник во время девятого такта формирует бит подтверждения Aс k, по которому передатчик убеждается, что его «услышали». После передачи бита подтверждения ведомое устройство может задержать следующую посылку, удерживая линию SCL на низком уровне. Ведомое устройство в режимах F/S может замедлить передачу по шине и на уровне приема каждого бита, удерживая SCL на низком уровне после его спада, сформированного передатчиком. Поэтому ведущее устройство должно генерировать сигнал SCL не «вслепую», а анализируя состояние линии SCL: сняв этот сигнал, новый импульс (открытие ключа передатчика) оно имеет право вводить, лишь убедившись, что сигнал SCL вернулся в пассивное состояние (высокий уровень). В противном случае синхронизация будет потеряна. Сигнал SCL может быть растянут и другим устройством, пытающимся захватить шину в это же время. Тактовый сигнал SCL не обязательно будет равномерным: время его нахождения на низком уровне будет определяться максимальным временем, в котором его захочет удержать самое медленное из устройств, участвующих в данном обмене (даже и конфликтующих); время нахождения на высоком уровне будет определяться самым быстрым из конфликтующих ведущих устройств.

*Коллизия* (конфликт) на шине может возникнуть, когда два (или более) устройства, убедившись в покое шины, одновременно (или почти одновременно) инициируют обмен данными. Все они управляют линиями SCL и SDA и наблюдают за ними. Если устройство, передающее единицу (высокий уровень), в данном такте на линии SDA видит ноль (низкий уровень), оно должно признать свой проигрыш в конфликте и освободить линии SCL и SDA (при этом ему позволительно управлять линией SCL до конца передачи текущего байта). Выигравшее устройство даже и не заметит проигравших конкурентов и продолжит работу. Арбитраж может закончиться в любом месте посылки, формируемой ведущим устройством. Искажения информации, передаваемой выигравшим устройством, не происходит (приятное отличие от коллизий в сетях Ethernet). Если ведущее устройство, проигравшее в конфликте, имеет и функции ведомого устройства, по признанию проигрыша оно должно перейти в режим ведомого, поскольку конфликт мог быть вызван и попыткой обращения к нему победившего ведущего устройства.

*Бит подтверждения АС К*, вводящийся в конце каждого байта устройством-приемником, выполняет несколько функций. Когда *передатчиком* является *ведущее устройство*, приемник (ведомый) должен вводить нулевой бит АС К, свидетельствующий о нормальном получении очередного байта. Единичный бит АС К (нет подтверждения) в ответ на посылку адреса свидетельствует об отсутствии адресованного ведомого устройства на шине или его занятости внутренними процессами. Отсутствие подтверждения байта данных свидетельствует о занятости устройства. Не получив бита подтверждения, ведущее устройство должно сформировать условие *Stop*, чтобы освободить шину. Когда *ведущее устройство* является *приемником*, оно должно формировать нулевой бит АС К после каждого принятого байта, кроме последнего. Единичный бит АС К в этом случае является указанием ведомому

устройству на окончание передачи — оно теперь должно освободить линии SDA и SCL, чтобы ведущее устройство смогло сформировать условие P или S г.

На вышеописанной физической основе строится протокол обмена данными по PC. Каждое *ведомое устройство* имеет свой *адрес*, уникальный на шине. В начале любой передачи ведущее устройство после условия S или S г посылает адрес ведомого устройства или специальный адрес (табл. 11.1). Ведомое устройство, опознавшее свой адрес после условия *Start*, становится *выбранным*; оно обязано ответить подтверждением на адрес и последующие сигналы со стороны ведущего устройства, до получения условия P или S г. В первоначальном варианте интерфейса разрядность адреса устройства составляла 7 бит, впоследствии был введен и режим 10-битной адресации, совместимый с 7-битной. На одной шине могут присутствовать устройства и с 7-битной, и 10-битной адресацией.

При *7-битной адресации* в первом байте после S (Sr) ведущее устройство передает 7 бит адреса (A[6:0] в битах [7:1]) и признак операции RW (в бите 0 RW=1 — чтение, RW=0 — запись). Адреса ведомых устройств не должны попадать в области, указанные в таблице. Диапазоны адресов устройств различных типов централизованно выдаются изготовителям микросхем фирмой Philips. Для микросхем памяти, например, 7-битный адрес содержит две части: старшие 4 бита A[6:3] несут информацию о типе устройства (EEPROM — 1010), а младшие 3 бита A[0:2] определяют номер устройства данного типа на шине. Микросхемы с интерфейсом PC имеют три адресных входа, коммутацией которых на логические уровни 1 и 0 задается номер устройства, на который оно «отзовется», а тип устройства «защит» в нем самом его изготовителем.

Когда *ведущее устройство* является *передатчиком* данных, оно в первом байте передает адрес ведомого устройства, при этом RW=0. Выбранное ведомое устройство отвечает подтверждением (ACK=0), после чего ведущее устройство посылает один или несколько байт данных, на каждый из которых ведомое устройство должно отвечать подтверждением.

Когда *ведущее устройство* является *приемником* данных, оно в первом байте передает адрес ведомого устройства с RW=1. Выбранное ведомое устройство также отвечает подтверждением (ACK=1), после чего происходит смена направления передачи и данные уже передает ведомое устройство. Ведущее устройство подтверждает каждый принятый байт, кроме последнего.

Эти передачи могут завершаться условием P, вводимым ведущим устройством, после которого шину может захватить любое ведущее устройство. Возможны и комбинированные передачи, когда ведущее устройство после окончания очередного обмена не отдает шину, а формирует повторный старт (Sr), после чего обращается к тому же или иному устройству.

Отметим, что спецификация PC не оговаривает правил модификации внутреннего (для микросхемы) адреса данных при последовательных обращениях — их определяет разработчик устройства в соответствии с его функциями. Для памяти естественен автоинкремент адреса, упрощающий последовательные обращения. Для регистроориентированных устройств автоинкремент обычно не нужен.

Таблица 11.1. Специальные адреса I<sup>2</sup>C

Биты [7:1]	Бит 0 (RW)	Назначение
0000000	0	General call address — адрес общего вызова
0000 000	1	Start — начало активного обмена
0000 001	X	Адрес устройства шины CBUS (для совместимости)
0000 010	X	Адрес для устройств иных шин
0000 011	X	Зарезервировано
0000 1XX	X	Код ведущего устройства режима <i>Hs</i>
1111 1XX	X	Зарезервировано
1111 0XX	X	Признак 10-битной адресации

Специальные коды, приведенные в табл. 11.1, трактуются следующим образом.

- ♦ Общий вызов является широкоэмитательным, на него должны отвечать лишь устройства, поддерживающие соответствующие функции (см. ниже).
- ♦ Байт Start предназначен для облегчения программной реализации протокола PC (для функций ведомых устройств, не имеющих полной аппаратной реализации протокола). На байт *S t a r t* не должно отвечать ни одно устройство. Формируемый сигнал SDA (рис. 11.2) может быть использован как запрос аппаратного прерывания, по которому процессор «вплотную» займется обработкой сигналов PC. До получения этого байта процессор (микроконтроллер) может не отвлекаться на слежение за сигналами интерфейса.
- ♦ На адреса шины CBUS (трехпроводный «родственник» шины PC) и иных шин устройства PC отвечать не должны.
- ♦ При использовании 10-битной адресации биты [2:1] содержат старшую часть адреса, форматы 10-адресных посылок рассмотрены ниже.

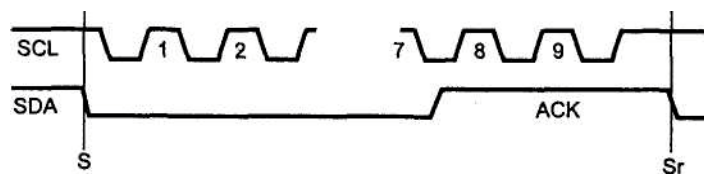


Рис. 11.2. Временная диаграмма байта Start

Посылка *общего вызова* используется для одной из двух целей, определяемых вторым байтом, младший бит которого называется *V*. При *V=0* устройства, принявшие общий вызов, должны считать программируемую часть своего аппаратного адреса, выполнив (второй байт 00000110Б) или не выполнив (00000Ю0Б) внутреннюю процедуру сброса. Другие значения байта устройства должны игнорировать. Каким образом программируется адрес, зависит от устройства (указывается в его описании). При *V=1* общий вызов используется для *широкоэмитательной передачи данных*. В этом случае ведущее устройство в старших 7 битах второго байта сообщает свой адрес (тот же, на который оно отзывается в роли ведомого), а далее



посылает требуемое количество байт данных, которые оно желает донести до неизвестного ему приемника. Приемник (как правило, это интеллектуальное устройство) должен отвечать подтверждением на каждый принятый байт начиная с первого (общий вызов), затем на адрес ведущего устройства и последующие байты данных.

Широковещательная передача может использоваться, например, аппаратным контроллером клавиатуры, не знающим, на какой адрес посылать свои асинхронно возникающие сообщения. Вместо этого возможна и иная схема: по включении (и сбросу) это устройство становится ведомым приемником, которому ведущее устройство (системный контроллер) сообщит адрес потребителя информации для дальнейших «узконаправленных» передач, в которых это устройство будет уже ведущим.

Введение *10-битной адресации* стало решением проблемы дефицита адресов: при 7-битной адресации с учетом зарезервированных комбинаций остается только 112 адресов, при 10-битной доступны еще 1024 адреса. *Передача* данных *ведущим устройством* с 10-битной адресацией выглядит просто: в первом байте после S (S г) биты 2:1 несут старшие биты адреса, бит 0 — признак RW=0; второй байт несет младшие 8 бит адреса, а последующие байты — передаваемые данные. Приемник дает подтверждения обычным способом. *Прием* данных *ведущим устройством* несколько сложнее, поскольку признак RW является явным указателем на смену направления передачи и младшую часть адреса ведущее устройство передать уже не может. Прием по 10-битному адресу ведущее устройство начинает как фиктивную передачу: в первом байте посылает признак 10-битной записи и 2 бита адреса, во втором байте — остаток адреса. Далее ведущее устройство выполняет условие Sg и посылает признак 10-битного чтения (RW=1) с двумя старшими битами того же адреса. Ведомый передатчик, получив те же 2 старшие бита адреса, что и до Sg, отвечает подтверждением и начинает передавать данные ведущему устройству. Этот процесс продолжается и завершается так же, как и с 7-битной адресацией. Возможны и комбинированные обмены: реальная передача ведущего устройства 10-битному устройству, за которой после Sg следует чтение того же устройства. Также возможно комбинирование (через Sg) 7- и 10-битных обращений к разным устройствам. Широковещание с 10-битной адресацией выглядит так же, как и с 7-битной, но адрес ведущего устройства передается уже двумя байтами (младшие 8 бит адреса передаются на месте первого байта данных). *Высокоскоростной режим (Hs)* позволяет обмениваться данными со скоростью до 3,4 Мбит/с, причем обеспечивается обратная совместимость устройств Hs с быстрыми и стандартными (-F/5). Для обеспечения возможности обмена на столь высокой скорости выходные и входные буферы микросхем должны переключаться в специальный режим работы, отличающийся параметрами формируемых и принимаемых импульсов. Сигналы высокоскоростных устройств обозначаются как SDAH и SCLH; в смешанных системах для работы в Hs они должны отделяться от линий SDA и SCL обычных устройств специальными мостами (поведение устройств F/S на таких частотах непредсказуемо). В режиме Hs уже нет речи о разрешении конфликтов — арбитраж выполняется на скоростях F/S; также здесь нет

возможности синхронизации по каждому биту (замедления передачи ведомым устройством), а ведущим устройством устанавливаются жесткие соотношения длительности низкого и высокого уровней сигнала SCLH (2:1). Ведомое устройство может притормаживать обмен только после выдачи бит подтверждений. Для перехода в режим *Hs* ведущее устройство в первом байте (после 5) использует зарезервированное значение 0000 1xxx, в котором xxx несет код ведущего устройства. Во время передачи этого байта (на скорости *F/S*) выполняется арбитраж — если обмен одновременно пытаются начать несколько устройств, продолжать его может только выигравшее ведущее устройство. Код ведущего устройства в режиме *Hs* назначается при конфигурировании, а все ведущие устройства на шине в режиме *Hs* должны иметь различные коды (код 000 зарезервирован), чем и обеспечивается завершение арбитража за время передачи первого байта. Ведущее устройство может переключиться в режим *Hs*, только если оно выиграло арбитраж и получило единичный бит подтверждения. В этом случае он перестраивает свои выходные и входные буферы на параметры *Hs* и формирует повторный старт (S г). Далее обмен логически выполняется точно так же, как и в режиме *F/S*, но уже на высокой скорости. Режим *Hs* может распространяться на несколько последующих передач, разделенных условиями S г, и завершится работа в режиме *Hs* по условию P, по которому буферные схемы снова вернутся к параметрам *F/S*.

### 11.1.2. Шина ACCESS.Bus

Последовательная шина ACCESS.Bus (Accessory Bus) разрабатывалась фирмой DEC как унифицированный недорогой интерфейс взаимодействия компьютера с внешними устройствами — клавиатурой, координатными устройствами, текстовыми устройствами (принтеры, считыватели штрих-кодов), мониторами (в плане обмена управляющей и конфигурационной информацией по каналу VESA DDC). История ACCESS.Bus начинается с 1991 г.; несколько позже в шину ввели дополнительную спецификацию для взаимодействия с внутренними устройствами, например, интеллектуальными источниками питания (Smart Battery) и т. п. К внутренним относятся устройства системного управления SM (System Management), и в спецификации имеются точки соприкосновения с шиной SMBus, основанной на том же интерфейсе PC. Формально шина позволяет обмениваться сообщениями устройствам числом до 125 (предел принятой системы адресации). Над аппаратным протоколом PC в шине ACCESS.Bus имеется базовый программный протокол, с которым взаимодействуют протоколы конкретных подключенных устройств. Протоколы обеспечивают подключение/отключение устройств без перезагрузки ОС и автоматическое динамическое назначение адресов. Последняя доступная версия ACCESS.bus Specifications Version 3.0 опубликована ACCESS.bus Industry Group в 1995 г., дальнейшее описание сделано на ее основе.

На аппаратном уровне шина логически полностью соответствует шине PC со стандартной скоростью (до 100 Кбит/с) и 7-битной адресацией ведомых устройств. Здесь работают те же механизмы синхронизации и арбитража. Однако из всех возможных способов передачи и приема данных в ACCESS.bus основным является *передача данных ведущим устройством* и их *прием ведомым устройством* — это

самый простой способ, при котором в каждой транзакции отсутствует смена направления передачи. Из этого следует, что для двустороннего обмена информацией все устройства должны поддерживать функции ведущего устройства (передатчика) и ведомого устройства (приемника). Для совместимости с SMBus разрешена возможность чтения данных ведущим устройством и комбинированные передачи через условие 5г.

По электрическим сигналам имеются две спецификации, для внешних и внутренних устройств соответственно.

*Спецификация для внешних устройств* (Off-board ACCESS.bus), являющаяся основной для этой шины, определяет использование 4-контактного экранированного модульного разъема (MOLEX SEMCONN или AMP SDL), назначение контактов которого приведено в табл. 11.2. Хост-компьютер должен обеспечивать питание 5 В с током 50-1000 мА. Каждое устройство (и кабель), характеризуется потребляемым током  $I$  (мА) и вносимой емкостью сигнальных проводов  $C$  (пФ). Предельное число подключаемых устройств ограничивается суммарной вносимой емкостью (не более 1000 пФ) и током потребления. До ограничения по адресации (125 устройств) дело практически не доходит. Максимальная суммарная длина кабеля (без повторителей) не должна превышать 10 м. По сравнению с PC в шине ток нагрузки линий SDA и SCL увеличен до 6 мА (выходной ток низкого уровня). Для улучшения формы импульсов и защиты от статического электричества устройства рекомендуется подключаться к линиям SDA и SCL через последовательные резисторы 51 Ом. Входы микросхем рекомендуется защищать диодами, соединенными с шинами GND и +5 В.

Таблица 11.2. Назначение контактов внешнего разъема ACCESS.bus

Контакт	Назначение	Цвет провода
1	GND	Черный
2	SDA	Зеленый
3	+5 В (питание устройств)	Красный
4	SCL	Белый

Ассоциация VESA для вывода внешней шины ACCESS.Bus на корпус мониторов предлагает иной, 5-контактный разъем ACCESS.Bus; назначение его контактов приведено в табл. 11.3.

Таблица 11.3. Разъем ACCESS.Bus (VESA)

Контакт	Назначение
1	GND
2	Ключ
3	SDA
4	+5 В (питание устройств)
5	SCL

*Спецификация для внутренних устройств* (On-board ACCESS.bus) рассчитана на меньшие токи нагрузки (350 мкА); здесь допускаются последовательные резисторы большего сопротивления. Вместо ограничения на вносимую емкость задаются требования к фронтам и спадам сигналов. Эта спецификация была введена только в 1995 г., она нацелена на совместимость с устройствами SMBus, и в нее введены дополнительные ограничения на максимальные длительности различных фаз, соответствующие SMBus.

*Базовый протокол шины ACCESS.bus 3.0* состоит из двух наборов: *протокол PA* для устройств с программируемым адресом (Programmable Address) и *протокол FA* для устройств с фиксированным адресом. В устройстве (как внешнем, так и внутреннем) может быть реализован любой из них или оба. Предыдущая версия спецификации описывала только внешние устройства PA; внутренние устройства с FA называются SM-устройствами (System Management). Протокол FA практически соответствует шине SMBus, описанной ниже, без «архитектурных излишеств» в виде PEC и динамического назначения адресов. Устройства SM могут общаться с хостом по протоколу Write Word (см. ниже). Базовый протокол PA основан на передаче *однонаправленных сообщений* (см. ниже). Шина ACCESS.bus является хост-центрической: сообщения передаются от устройства к хосту и от хоста к устройству; исключением является попытка сброса устройства-двойника (см. ниже). После включения питания устройства должны отвечать только на «дежурный» адрес 0110 111; в процессе конфигурирования каждому устройству назначается личный адрес. В рабочем состоянии шина позволяет обнаруживать подключение новых устройств и их конфигурировать без перезагрузки («горячее подключение»).

*Сообщения* передаются в виде пакетов, формат пакета приведен на рис. 11.3. Адрес назначения DestAddr воспринимается получателем аппаратно (это адрес ведомого устройства PC). Адрес источника SrcAddr позволяет получателю идентифицировать источник данных (и определить, куда посылать ответ). Флаг протокола P позволяет различать назначение тела пакета: P=0 — «полезные» данные устройства (Device Data Stream); P=1 — управление/состояние (control/status). Поле Length определяет длину тела пакета (в байтах); само тело (Body) размещается в последующих байтах. Контрольный байт Checksum является результатом выполнения функции XOR (Исключающее ИЛИ) над всеми предшествующими байтами пакета начиная с адреса приемника. Признаком целостности пакета является нулевой результат функции XOR от всех байтов пакета включая контрольный. Подлежат обработке только пакеты с корректным контрольным байтом. Минимальная длина всего пакета — 4, максимальная — формально 131 (127 байт тело и 4 байта обрамления). Однако максимальную длину пакета ограничивает и время, разрешенное устройству для передачи пакета.

Каждому устройству назначается свой адрес, на который оно должно отзываться битами подтверждения при приеме сообщения. Адрес выражают однобайтным числом, причем всегда четным, поскольку в PC 7-битный адрес дополняется младшим битом RW, нулевым в ACCESS.bus. Адрес 50h всегда назначается хост-компьютеру, адрес 10h зарезервирован для хоста SM-устройств. Адрес 6Eh является

«дежурным» адресом, на который отзываются лишь устройства с неназначенным личным адресом. Для личных адресов устройств остаются диапазоны 02-4Eh; 52-GCh; 70-FEh — 125 адресов с некоторыми исключениями, зарезервированными для фиксированных адресов SM-устройств и мониторов.

Бит/Байт	7	6	1	0
1	DestAddr — адрес приемника			0
2	SrcAddr — адрес источника			0
3	P	length		
4	Body — тело пакета			
	...			
Length + 4	Checksum — контрольный байт			

Рис. 11.3. Формат пакета сообщения ACCESS.Bus

Для ACCESS.bus определено 9 протокольных сообщений (у них флаг P=1), обязательных для реализации интерфейсных функций шины (автоконфигурирования). «Полезными» прикладными сообщениями могут обмениваться только сконфигурированные устройства и только после явного разрешения этого обмена.

Ниже перечислены сообщения от хоста к устройствам.

- ◆ Reset — сброс устройства и перевод его в режим ответа на «дежурный» адрес. Тело состоит из однобайтного кода FOh. Это же сообщение может послать и устройство, обнаружившее на шине помеху в виде устройства-двойника с тем же адресом. Послав это сообщение по своему же собственному адресу, устройство заставит двойника перейти на «дежурный» адрес.
- ◆ Identification Request — запрос идентификационной строки. Тело состоит из однобайтного кода Flh.
- ◆ Assign Address — назначение устройству, имеющему совпадающую идентификационную строку, нового адреса. Тело (длина 30) начинается с кода F2h, за которым следует 28-байтный идентификатор устройства, а за ним — байт нового адреса.
- ◆ Capabilities Request — запрос фрагмента информации о возможностях устройства. В теле за кодом F3h следует 16-битный параметр — смещение требуемых данных относительно начала структуры данных возможностей. Для упрощения логики устройств параметр ограничивается значениями, обеспечивающими чтение первого фрагмента (с нулевым смещением), следующего и переспрос последнего переданного.
- ◆ Enable Application Report — разрешение передачи прикладных данных. За кодом F5h следует байт кода операции: 00h — запрет, 01
- ◆ Presence Check — проверка наличия устройства по данному адресу. За кодом F7h следует нулевой байт (зарезервирован на будущее).

Далее перечислены *сообщения от устройств к хосту*.

- ◆ **Attention** — запрос на конфигурирование (устройство включилось и завершило автоинициализацию). Тело состоит из однобайтного кода E0h.
- ◆ **Identification Reply** — ответ на запрос идентификационной строки. Тело (длина 29) содержит код E1h, за которым следует 28-байтная строка идентификации.
- ◆ **Capabilities Reply** — ответ на запрос фрагмента описания возможностей. Тело (длина 3-35) начинается с кода E3h, за которым следует 16-битное смещение (см. запрос) и собственно данные (0-32 байт). Хост собирает фрагменты, используя смещение.

Также в спецификации определены дополнительные протокольные сообщения, используемые для управления потреблением, распределением ресурсов и иных целей (у этих сообщений также флаг P-1).

- ◆ **Resource Request** — запрос ресурса (от устройства к хосту). За кодом E5h следует байт-описатель ресурса и необходимые данные. Команда позволяет запросить адрес в личное пользование и освободить его; запросить сообщение о текущем времени; запросить хост о сохранении блока данных, а также о возврате его обратно; запросить хост о сохранении питания на шине (для окончания внутренних операций); запросить дополнительную полосу шины.
- ◆ **Resource Grant** — выделение ресурса, ответ хоста на запрос. За кодом F4h следует описатель ресурса и необходимые данные.
- ◆ **Application Hardware Signal** — запрос устройства на генерацию высокопримитивного аппаратного сигнала хост-компьютеру. За кодом A0h следует байт со следующим кодом сигнала:
  - 1 — Reset — попытка аппаратного сброса компьютера;
  - 2 — Halt — вызов отладчика;
  - 3 — Attention — генерация сигнала внимания (аппаратное прерывание).
- ◆ **Application Test** — команда от хоста на выполнение устройством прикладного теста (код B1h).
- ◆ **Application Test Reply** — сообщение устройством о результатах выполнения теста. За кодом A1h следует код результата (0 — успешное выполнение, иначе — ошибка) и 0-30 байт дополнительных данных.
- ◆ **Application Status Message** — сообщение устройством об изменении своего состояния (в прикладном плане). За кодом A2h следует нулевой байт, за ним байт состояния и 2 байта специфических данных. Байт состояния:
  - 00 — готово;
  - 01 — не готово;
  - 02 — изменились свойства;
  - 03 — потеряно внутреннее состояние;
  - 04 — потеряны прикладные данные (может, и от переполнения).

- ◆ Device Power Management Command — команда управления потреблением устройства. За кодом F6 следует байт кода операции:
  - 00 — режим *Run*;
  - 01 — режим *Standby*;
  - 02 — режим *Suspend*;
  - 03 — режим *Shutdown*;
  - 04 — совет отключить питание;
  - 05 — рестарт;
  - 06 — сообщить режим потребления.

Остальные коды протокольных сообщений задаются разработчиком в соответствии со спецификой устройств. Напомним, что прикладные данные передаются с флагом P=0.

*Строка идентификации устройства ACCESS.bus* длиной 28 байт состоит из ряда символьных полей — байта ревизии протокола (protocol revision), 7-байтного поля ревизии модуля (module revision), 8-байтных имен производителя (vendor name) и модуля (module name), за которым следует 32-битный уникальный номер устройства (device number). Этот номер может быть либо фиксированным (уникальность обеспечивает производитель, что не дешево), либо случайным числом, генерируемым по включению (на весь сеанс работы). Системное ПО, распознавая устройство для подключения драйверов, не должно руководствоваться этой строкой — возможности устройства (Capabilities) описываются (и сообщаются) в специальной структуре данных. Эта структура зависит от типа устройства.

На уникальности идентификатора и основан механизм *автоконфигурирования*: на запрос идентификатора по «дежурному» адресу отвечают все устройства, еще не имеющие личных адресов. Однако в ходе арбитража до конца сообщения доходит только одно из этих устройств, после чего хост ему назначает личный адрес. В следующем общем опросе идентификаторов «победит» уже другое устройство и так далее, пока всем устройствам не будут назначены личные адреса (об этом хост узнает по отсутствию ответа на общий опрос). Устройство-«новичок» на шине заявит о своем появлении сообщением *Attention*, в ответ на которое хост выполнит вышеописанную процедуру идентификации и назначения адреса.

Спецификация ACCESS.bus определяет структуру программного обеспечения на хост-компьютере. Центральным элементом ПО является *менеджер шины* — ACCESS.bus Manager — программный драйвер, управляющий всеми операциями с устройствами, подключенными к шине. Этот драйвер, с одной стороны, связывается с аппаратными средствами хост-контроллера через драйвер минипорта MPD; с другой стороны, к нему обращаются драйверы устройств. Прикладное ПО обращается либо к драйверам нужных устройств, либо к менеджеру шины (но никак не напрямую к хост-контроллеру). Менеджер шины инициализирует шину и управляет ею, определяя вновь подключенные и отключенные устройства. Он связывает драйверы устройств (или прикладное ПО) с самими устройствами, проверяет входящие сообщения и работает как двунаправленный коммутатор данных,

переформатирующий и буферизирующий входящие и исходящие сообщения. *Драйвер мини-порта* MPD (Mini Port Driver) служит для изоляции менеджера шины от аппаратных особенностей хост-контроллера. *Драйверы устройств* являются двусторонними интерфейсами между прикладными программами и специфическими устройствами. В спецификации ACCESS.bus описываются программные интерфейсы драйверов (Device Driver, Mini Port Driver), а также протоколы для клавиатур, указателей (Locator), мониторов, батарей и текстовых устройств.

### 11.1.3. Шина SMBus

Шина SMBus (System Management Bus — шина системного управления) — двухпроводной интерфейс для обмена данными между микросхемами различных системных компонентов компьютера, а также связи их с самим компьютером. Основное назначение интерфейса — управление подсистемой питания и сопутствующими подсистемами. Первоначально шина разрабатывалась для интеллектуальных батарей и зарядных устройств, и первые версии спецификации SMBus шли под заголовком «Smart Battery System Specifications» (спецификации системы «умных» батарей). Система с «умными», или «ловкими», батареями (Smart Battery System) состоит из собственно батарей (аккумуляторов) и зарядных устройств, способных обмениваться управляющей информацией между собой и с хост-системой, которую она питает. Этот обмен позволяет батареям сообщать свои параметры (текущие значения, ожидаемые величины), подключаться в режим работы (питания хост-системы) или зарядки. Часть управляющих функций выполняется при участии хоста, а часть — автономно. Первая версия спецификации SMBus вышла в 1995 г., версия 1.1 — в 1998 г. Версия 2.0 вышла в 2000 г., она уже не имеет «батарейного» заголовка. Данное описание основано на документе «System Management Bus (SMBus) Specification Version 2.0», выпущенном форумом разработчиков систем с интеллектуальными батареями (SBS, [www.sbs-forum.org](http://www.sbs-forum.org)), в который входит большое число производителей источников питания, а также фирма Intel. Спецификация покрывает три нижних уровня модели взаимодействия открытых систем (OSI), от физического до сетевого.

Шина SMBus основана на интерфейсе I<sup>2</sup>C, и к ней применимо все сказанное в п. 11.1.1. В шину введена возможность динамического реконfigurирования (подключения и отключения абонентов шины в процессе работы) и автоматического назначения адресов устройств. По сравнению с PC в шине несколько изменены граничные требования к уровням сигналов и временным диаграммам (см. п. 11.1.4), но в основном они совпадают. Шина позволяет реализовать все режимы обменов, разрешенные для PC, но имеет нюансы в правилах генерации подтверждений. Для шины SMBus в BIOS имеется стандартизованная поддержка. Особенностью шины SMBus, связанной с ее ролью в управлении системой питания, является способность автономной работы — соединяемые ею устройства могут обмениваться информацией, даже когда питание на центральном процессоре (и других подсистемах) отсутствует. Конечно, о взаимодействии с устройствами шины через BIOS в таком состоянии нет и речи, поскольку BIOS еще «спит».



На физическом уровне (1-м уровне OSI) спецификация определяет электрические и временные параметры сигналов. По уровням сигналов (и нагрузочной способности) имеются две различные спецификации. *Маломощная* (low power) спецификация соответствует «родному» применению SMBus в устройствах с батарейным питанием; здесь характерны малые токи. *Мощная* (high power) спецификация предназначена для работы абонентов SMBus в окружении источников значительных помех (например, на плате PCI). Маломощные и мощные устройства на одной шине вместе работать в общем случае не смогут, но это и не требуется. При необходимости совместного использования устройств разных классов организуются разные сегменты шины, соединенные мостом.

В спецификациях временных параметров приняты меры по ограничению времени отклика и предотвращению «зависания» шины. Частота тактового сигнала ограничена и снизу (10 кГц), и сверху (100 кГц); введены ограничения на максимальную длительность нахождения синхросигнала в высоком и низком состоянии и максимальную суммарную «растяжку» битовых интервалов на каждый байт. Предусматривается механизм тайм-аутов, по которому устройства, обнаружившие «зависание» шины, должны немедленно прекратить обмен и повторно инициализироваться. В спецификации PC эти моменты не рассматривались.

На шине SMBus могут использоваться *дополнительные аппаратные сигналы*.

- ◆ Сигнал SMBSUS\* указывает на приостанов шины и устройств. Этот сигнал вырабатывается устройством управления питанием; во время его активности (низкий уровень) сигналы SDA и SCL штатным образом (как в PC) не воспринимаются (они могут использоваться для уточнения режима приостанова).
- ◆ Сигнал SMBALERT# служит для оповещения ведущего устройства о необходимости обмена с ведомым устройством, не имеющим возможности выступить в роли ведущего устройства. Этот сигнал собирается по схеме «Монтажное И» от всех устройств. Получив его, хост должен дать команду чтения байта по адресу 0001 100, на которое просигналившее устройство должно ответить байтом со своим адресом (возможен вариант и с дополнительным байтом PEC).

На уровне связи (2-м уровне OSI) определяются те же правила передачи данных, что и в PC: условия S, P, Sr; биты подтверждения; 7-битная адресация и признак RW в первом байте, следующем за условием 5 (5r). Как и в PC, ведущее устройство может выполнять и запись, и чтение данных ведомого устройства; используются и комбинированные транзакции (через условие Sr). В генерации подтверждений на SMBus есть особенности. На собственный адрес устройство всегда должно отвечать битом подтверждения AC K, даже если оно занято внутренними операциями. Это правило обеспечивает работу механизма определения присутствия данного устройства на шине. Ведомое устройство может вводить бит NACK в ответ на любой неадресный байт, если оно занято или запрашиваемые командой данные недоступны. В этом случае у него есть и альтернатива поведения — задержать синхросигнал на низком уровне (в разрешенных пределах). Ведомое устройство должно вводить бит NACK в ответ на недопустимые коды команд или данных. Ответ NACK вынуждает ведущее устройство прекратить транзакцию (ввести P). Ведущее устройство, будучи приемником, ответом NACK информирует передатчик о приеме последнего ожидаемого байта.

*Сетевой уровень* (3-й уровне OSI) определяет «лицо» шины SMBus и заслуживает более детального рассмотрения.

### Сетевой уровень SMBus

В шине SMBus введено понятия «хоста» (host) — абонента шины, выполняющего координирующие и конфигурирующие функции. Хост является ведущим устройством шины, при этом должен выполнять ряд функций ведомого устройства и обрабатывать сообщения уведомления.

Каждое ведомое устройство имеет свой уникальный адрес; в диапазоне 7-битных значений адреса выделяются специальные значения (табл. 11.4), которых несколько больше, чем в PC. 10-битная адресация в текущей версии не рассматривается. Адреса устройств разделяются по типам. Для устройств однозначно понятного назначения SMBus WG выделяет специальные адреса (Purpose-assigned addresses). Например, батареи Smart Battery имеют адрес 0001 011, их зарядные устройства— 0001 001. Устройства с этими адресами обязаны соответствовать требованиям SMBus, предъявляемым к устройствам данного класса. Ряд систем с SMBus определяют и используют эти устройства, основываясь на их адресе. Другие системы могут и не доверять одному только адресу, а определять типы присутствующих устройств иным образом. Для устройств разнообразного назначения, а также устройств, не полностью отвечающих спецификациям SMBus для своего класса, производители назначают иные адреса, с которыми можно ознакомиться на сайте [www.smbus.org](http://www.smbus.org). Адреса устройств-прототипов задействуются исключительно в экспериментально-отладочных целях и в коммерческих изделиях использоваться не должны. В спецификации SMBus 2.0 появилась возможность автоматического динамического назначения адресов устройств, которая будет рассмотрена ниже.

**Таблица 11.4.** Специальные адреса SMBus Биты

[7:1]	Бит 0 (RW)	Назначение
0000 000	0	General call address — адрес общего вызова
0000 000	1	Start — начало активного обмена
0000 001	X	Адрес устройства шины CBUS (для совместимости)
0000 010	X	Адрес для устройств иных шин
0000 011	X	Зарезервировано
0000 1XX	X	Зарезервировано
0101000	X	Хост шины ACCESS.bus
0110111	X	«Дежурный» адрес ACCESS.bus
1111 OXX	X	Признак 10-битной адресации
1111 1XX	X	Зарезервировано
0001 000	X	Хост шины SMBus
0001 100	X	Адрес ответа на сигнальные сообщения SMBus
1100001	X	«Дежурный» адрес SMBus
1001 OXX	X	Адрес устройств-прототипов

Типичное устройство SMBus поддерживает определенный набор команд, с помощью которых выполняется обмен данными. Команда кодируется одним байтом, передаваемым в транзакции вслед за адресным байтом. Команды могут использовать один из 11 протоколов, определенных в SMBus.

В версии 1.1 спецификации SMBus введена возможность контроля ошибок пакета PEC (Packet error checking). Механизм PEC основан на добавлении в конец каждого передаваемого пакета байта CRC-кода, вычисляемого по всем предыдущим байтам пакета, начиная с адресного. Почти все протоколы могут иметь два варианта — без PEC и с PEC; на одной шине могут присутствовать устройства и с поддержкой PEC, и без. На байт PEC приемник отвечает подтверждением, но трактовка ответа неоднозначна. Если передатчик в ответ на PEC получил ответ NACK, это означает, что приемник не подтвердил корректный прием пакета. Однако ответ ACK не является подтверждением достоверности приема: приемник может «не понимать» PEC и отвечать на него как на обычный байт данных; приемник может и не выполнять контроль в реальном времени приема потока данных. Более «достоверный контроль достоверности» могут обеспечить лишь протоколы высших уровней. Так, например, для контроля достоверности записи в устройство можно использовать последующее чтение тех же данных с PEC, и по анализу всего принятого пакета ведущее устройство сделает вывод об успешности или ошибке операции записи.

*Шинные протоколы SMBus* основаны на транзакциях PC с 7-битной адресацией.

- ◆ *Quick Command*, короткая команда, — посылка адресного байта; действие команды определяется битом RW адресного байта.
- ◆ *Send Byte*, посылка байта, — передача ведущим устройством вслед за адресным байтом (RW=0) одного байта данных. В варианте с PEC передаются два байта, последний — PEC.
- ◆ *Receive Byte*, прием байта, — прием ведущим устройством вслед за адресным байтом (RW=1) одного байта данных. В варианте с PEC принимаются два байта, последний — PEC.
- ◆ *Write Byte*, *Write Word*, запись байта/слова, — передача ведущим устройством вслед за адресным байтом (RW=0) одного байта команды, за которым следует 1 или 2 байта (младший, а затем старший) данных. В варианте с PEC в конец добавляется контрольный байт.
- ◆ *Read Byte*, *Read Word*, чтение байта/слова, — комбинированные транзакции:  
сначала посылается адресный байт (RW=0), за которым передается код команды. Далее, через условие 5 посылается адресный байт с тем же адресом устройства, но RW=1, после которого принимается 1 или 2 байта данных. В варианте с PEC в конце ожидается прием дополнительного (контрольного) байта.
- ◆ *Block Write*, запись блока, — передача ведущим устройством вслед за адресным байтом (RW=0) одного байта команды, за которым следует байт-указатель длины (количество последующих байт) и собственно байты данных. В варианте с PEC в конец добавляется контрольный байт. Указатель длины не учитывает байт PEC; он не может быть нулевым; одной блочной командой можно пересылать до 32 байт данных.

- ◆ *Block Read*, чтения блока, — комбинированная транзакция: сначала посылается адресный байт ( $RW=0$ ), за которым передается код команды. Далее, через условие  $S$  посылается адресный байт с тем же адресом устройства, но  $RW=1$ , после которого принимается байт-указатель длины, а за ним и собственно байты данных. В варианте с PEC в конце ожидается прием дополнительного (контрольного) байта. Указатель длины — аналогично блочной записи.
- ◆ *Process Call*, вызов процесса, — комбинация команды *Write Word* с приемом слова (двух байт) от устройства с тем же адресом. Команда называется вызовом процесса, поскольку ожидает данных, зависящих от посланного кода команды и слова данных. В варианте с PEC контрольный байт ожидается только в самом конце, вслед за последним байтом принятых данных.
- ◆ *Block Write-Block Read Process Call* — комбинация записи блока с последующим чтением блока по тому же адресу устройства. В варианте с PEC контрольный байт ожидается только в самом конце, вслед за последним байтом принятых данных.

В случае, когда ведущим устройством шины собирается выступать рядовое устройство (не хост), оно должно использовать протокол уведомления хоста (SMBus host notify protocol): устройство на адрес хоста с  $RW=0$  (он известен) посылает байт с собственным адресом, за которым следует слово (два байта) собственно уведомления. Хост обязан понимать эти уведомления; дополнительно может использоваться и необязательный сигнал внимания SMBALERT\* (см. ниже).

### Автоматическое назначение адресов

Динамическое реконфигурирование шины SMBus — возможность «горячего» подключения/отключения основано на базовых принципах протокола PC. Автоматическое назначение адресов, появившееся в версии 2.0, использует еще и контрольные байты пакетов (PEC). Задача динамического реконфигурирования включает распознавание фактов подключения/отключения устройств и обеспечение бесконфликтного распределения их адресов. Подключение новых устройств может распознаваться двумя способами. Устройство, которое может работать ведущим устройством, при подключении (после своей инициализации по включению питания) может послать хосту уведомление, содержащее его адрес. Другой вариант обнаружения — периодический опрос устройств ведущим устройством, ведающим «переучетом» всех устройств на шине.

Для динамического бесконфликтного назначения личных адресов устройств используется протокол ARP (Address Resolution Protocol). Назначение адресов основано на стандартном механизме арбитража (разрешения конфликтов) шины SMBus (и PC). Назначенный адрес запоминается устройством на все время, пока подано питание. Возможны и устройства PSA (Persistent Slave Address), «вспоминающие» ранее назначенный адрес после повторного включения питания. После назначения адреса обмен с устройством выполняется точно так же, как и с устройством с фиксированным адресом. Назначение адресов может выполнять любое ведущее устройство шины SMBus.

Для динамического назначения адреса требуется изоляция устройств — возможность диалога ведущего устройства-нумератора с каждым устройством без помех со стороны других устройств (типичная задача настройки системы PnP). Изоляция основана на уникальном идентификаторе устройства UDID (Unique Device Identifier) — 128-битной структуре, содержащей описание возможностей, версию, идентификаторы производителя, устройства, подсистемы и специфическую информацию. Идентификатор начинается с байта возможностей (Device Capabilities), в котором два старших бита характеризуют способности динамической адресации, а младший бит — поддержку PEC. Чтение идентификатора выполняется ведущим устройством ARP по протоколу блочного чтения по «дежурному» адресу SMBus. На это чтение отзываются все устройства с еще не назначенными адресами, и на арбитраже этой операции работает изоляция устройств. Первый считанный байт (указатель длины) у всех устройств одинаков, по нему конфликтов нет. Далее устройства передают идентификаторы, и в арбитраже будут иметь приоритет те устройства, у которых нулевое значение бит данных встретится раньше. С учетом этого принята следующая кодировка классов устройств в старших битах первого байта идентификатора:

- ◆ 00 — устройства с фиксированными адресами, идентифицируются в первую очередь;
- ◆ 01 — динамические устойчивые (persistent) адреса устройств PSA;
- ◆ 10 — динамические изменчивые (volatile) адреса;
- ◆ 11 — устройства со случайными номерами, идентифицируются в последнюю очередь.

Младший бит этого байта указывает на поддержку PEC и всех возможностей, ознанных на идентификаторе. При нулевом значении этого бита о поддержке PEC ничего определенного сказать нельзя.

Последний элемент UDID — 32-битный идентификатор устройства, играющий важную роль в распознавании похожих устройств. Это может быть либо число, назначаемое изготовителем (отвечающим за его неповторимость), либо случайное число, генерируемое (и запоминаемое) устройством каждый раз при включении или выполнении команды сброса.

Устройство, поддерживающее ARP, должно иметь специальные *флаги*:

- ◆ AR (Address Resolved) — данному устройству процедурой ARP назначен адрес;
- ◆ AV (Address Valid) — устройство имеет действительный личный адрес, на который оно отзывается в обычных транзакциях (при AV=0 устройство должно игнорировать все адреса, кроме «дежурного»).

При AV=AR=0 устройство не имеет личного адреса и должно участвовать в процессе ARP (отвечать на общий опрос идентификатора). При AV=1 и AR=0 устройство имеет личный адрес, но должно участвовать в ARP. При AV=AR=1 устройство имеет личный адрес, но не должно отвечать на общий запрос идентификатора. При этом оно должно обрабатывать адресованную ему команду назначения адреса (и впоследствии пользоваться новым назначенным адресом). Комбинация AV=0 и AR=1 недопустима.

Для протокола ARP введены специальные команды.

- ◆ Get UDID (general) — общий запрос идентификатора — команда чтения блока данных с адресом 1100 001 и кодом команды 3. На нее устройства, поддерживающие ARP, отвечают посылкой блока с длиной 17 байт, сопровождаемого PEC. Первые 16 байт блока — UDID, 17-й байт — адрес устройства (с единицей в младшем бите). Если у устройства флаг AV=0, оно вместо адреса передает код «1111 111». Команда не влияет на флаги AV и AR.
- ◆ Assign address — назначить адрес — команда записи блока данных с адресом 1100 001 и кодом команды 4 и длиной 17 байт, сопровождаемого PEC. Первые 16 байт блока — UDID, 17-й байт — назначаемый адрес устройства (младший бит игнорируется). По этой команде устройство, обнаружившее полное совпадение UDID со своим собственным, устанавливает флаги AV=AR=1.
- ◆ Get UDID (directed) — направленный запрос идентификатора—команда чтения блока данных с адресом 1100 001, в поле кода команды находится адрес интересующего устройства с единицей в младшем бите. На нее отвечает только устройство, опознавшее свой адрес в поле команды, и отвечает посылкой блока с длиной 17 байт, сопровождаемого PEC. Первые 16 байт блока — UDID, 17-й байт — адрес устройства (с единицей в младшем бите). Команда не влияет на флаги AV и AR.
- ◆ Reset device (general) — общий сброс устройства — посылка по адресу 1100 001 байта команды с кодом 2, сопровождаемого байтом PEC. По этой команде все устройства инициализируются и обнуляют флаги AR и AV (PSA-устройства флага AV не изменяют).
- ◆ Reset device ARP (directed) — направленный сброс устройства — посылка по адресу 1100 001 байта с адресом целевого устройства (с нулем в младшем бите), сопровождаемого байтом PEC. По этой команде выбранное устройство инициализируется и обнуляет флаги AR и AV (PSA-устройства флага AV не изменяют).
- ◆ Notify ARP master — уведомление ведущего устройства ARP — посылка на адрес 0001 000 байта с «дежурным» адресом 1100 0010, за которым следуют два байта нулей. Устройство может посылать это уведомление о необходимости выполнения ARP при включении питания, а также при обнаружении коллизии в процессе выполнении чтения данных по индивидуальному адресу устройства.

На каждый байт команд ARP-устройства, поддерживающие этот протокол, отвечают подтверждением ACK. Отсутствие подтверждений трактуется ведущим ARP-устройством как отсутствие ARP-устройств на шине.

«Переучет» и назначение адресов вкратце выглядит следующим образом: ведущее ARP-устройство выполняет команду общего запроса идентификатора и по ней получает UDID и, возможно, адрес первого «победителя» в арбитраже. Далее этому победителю назначается личный адрес — тот же или по усмотрению ведущего ARP-устройства, после чего он уже не участвует в арбитраже по общему запросу. Сведения об устройстве и его адрес ведущим ARP-устройством заносятся

в таблицу устройств. После этого общий запрос и назначение адреса повторяются снова и снова, пока все ARP-устройства не будут удовлетворены. Признаком «общего удовлетворения» для ведущего ARP-устройства будет отсутствие подтверждений приема команд общего запроса.

### Поддержка SMBus в BIOS и ACPI

Шина SMBus, в отличие от ACCESS.bus, сразу получила спецификацию ее поддержки на уровне BIOS. Позже появились спецификации, позволяющие интегрировать устройства SMBus в систему ACPI.

В 1995 г. была опубликована спецификация интерфейса SMBus BIOS — System Management Bus BIOS Interface Specification, основные идеи которой вкратце изложены ниже. Интерфейс позволяет верхним уровням ПО абстрагироваться от аппаратной реализации хост-контроллера. Поддержка BIOS обеспечивается для трех типов режимов работы процессора: реального (и V86), защищенного 16-разрядного и защищенного 32-разрядного. Вызов функций может выполняться либо через BIOS Int 15h (в реальном режиме и в V86), либо через точку входа, полученную при подключении в соответствующем режиме. Для подключения (и отключения) также используется сервис BIOS Int 15h; после подключения доступ через Int 15 h блокируется (до отключения). В защищенном режиме вызов интерфейсных функций возможен только через точку входа, полученную при подключении. Поддержка точки входа для реального режима необязательна.

Спецификация SMBus BIOS обеспечивает хост-центрическое обращение к абонентам шины: по инициативе вызывающей программы хост-контроллер посылает устройству команду, которая может предполагать и немедленный ответ устройства. Однако устройства могут посылать хосту сообщения по собственной инициативе, при этом они обязаны использовать протокол Write Word. Хост способен помещать принятые сообщения в небольшую очередь, из которой они могут программно извлекаться путем вызова функции 7 (программа должна периодически выполнять этот вызов для проверки наличия сообщений в очереди). В очереди каждое сообщение представлено байтом адреса источника и парой байт тела сообщения.

Шина SMBus тесно связана с оборудованием, управляющим питанием и участвующим в генерации запросов системного прерывания SMI (System Management Interrupt — особое аппаратное прерывание процессора) и их обработки. Из-за этого в BIOS введен специальный механизм, позволяющий обнаружить обработку SMI во время выполнения транзакций и в иное время. Это необходимо, поскольку обработчик SMI, работающий в режиме SMM, совершенно невидим прикладной программе, а результаты его работы могут существенно влиять на работу программы, вызывающей BIOS SMBus.

*Функции общих обращений к SMBus:*

- ◆ SMBus Installation Check (01h) — проверка наличия функций;
- ◆ SMBus Real Mode Connect (02h) — подключение в реальном режиме;
- ◆ SMBus 16-Bit Connect (03h) — подключение в 16-битном защищенном режиме;

- ◆ SMBus 32-Bit Connect (04h) — подключение в 32-битном защищенном режиме;
- ◆ SMBus Disconnect (05h) — отключение от сервиса;
- ◆ SMBus Device Address (06h) — получение списка адресов устройств SMBus;
- ◆ SMBus Critical Messages (07h) — чтение сообщений устройств, переданных хосту.

Для взаимодействия с конкретными устройствами SMBus предназначен набор функций, позволяющих генерировать запросы протокольных команд SMBus и получать результаты их выполнения. Функции запросов и получения ответов разделены, что позволяет на время выполнения (и передачи) довольно длительных команд не занимать время центрального процессора. Большинство протокольных команд вводится за один вызов BIOS; исключение составляет команда записи блока, данные для которой передаются за один или более последующих запросов продолжения. Результаты большинства команд также принимаются за один вызов; результат чтения блока получается за несколько вызовов продолжения.

*Вызовы протокольных команд устройств SMBus:*

- ◆ SMBus Request (10h) — запрос команды устройству;
- ◆ SMBus Request Continuation (11h) — продолжение запроса для записи блока;
- ◆ SMBus Request Abort (12h) — отказ от выполнения ранее посланного запроса;
- ◆ SMBus Request Data and Status (13h) — запрос данных и состояния.

Шина SMBus является одним из главных коммуникационных средств в ACPI. Интерфейс SMBus для ACPI определен в спецификации *SMBus CMI*— документе System Management Bus (SMBus) Control Method Interface Specification Version 1.0, опубликованном в конце 1999 г. Этот интерфейс позволяет легко использовать все возможности SMBus независимо от происхождения или особенностей реализации оборудования. Ниже перечислены основные цели спецификации.

- ◆ Обеспечить эффективный и надежный интерфейс ACPI для аппаратных средств хост-контроллера независимо от его реализации (со встроенным микроконтроллером или без него).
- ◆ Обеспечить системную синхронизацию доступа ко всем ресурсам SMBus.
- ◆ Гарантировать соответствие интерфейса версиям 1.0 и 1.1 спецификации SMBus и возможности его расширения для поддержки новых свойств в будущих версиях.

Для успешного функционирования системы требуется поддержка соответствующими драйверами операционной системы. Для управления питанием форум разработчиков «интеллектуальных» батарей — Smart Battery System Implementers Forum (SBS-IF) — разработал спецификацию драйверов для ОС Windows Эх/2000, с которой можно ознакомиться на сайте <http://www.sbs-forum.org/smbus/>.

Рассмотрение программного интерфейса CMI выходит за рамки данной книги. Отметим лишь, что коды протоколов SMBus, используемые в CMI, отличаются от кодов, используемых в SMBus BIOS. Те же протоколы, но с байтом PEC, кодируются с единицей в старшем бите (значение увеличено на 80h).



### 11.1.4. Сравнение шин I<sup>2</sup>C, ACCESS.bus и SMBus

Как видно из вышеприведенных описаний, все эти три шины являются «близкими родственниками»; однако они имеют ряд электрических, конструктивных и протокольных отличий.

*Электрический интерфейс* шин (табл. 11.5) достаточно близок, и при обычном (5 В) питании схем проблем совместимости не возникает ( $V_{dd}$  — напряжение питания).

Таблица 11.5. Электрические параметры шин I<sup>2</sup>C, ACCESS.bus и SMBus

Параметр	I <sup>2</sup> C	ACCESS.bus	ACCESS.bus	SMBus	SMBus
		внешняя	внутренняя	мощная	маломощная
Входной уровень лог. 0, не более	1,5В или 0,3V <sub>m</sub>	0,3V <sub>dd</sub>	0,3V <sub>M</sub>	0,8 В	0,8 В
Входной уровень лог. 1, не менее	3,0 В или 0,7V <sub>dd</sub>	0,7V <sub>dd</sub>	0,7V <sub>M</sub>	2,1В	2,1В
Выходной уровень лог. 0, не более	0,4 В или 0,2V <sub>dd</sub>	0,6 В	0,6 В	0,4 В	0,4 В
Выходной ток лог. 0, мА	3	6	0,35	4	0,35
Максимальная емкость линии шины, пФ	400	1000	1	400	400
Частота, КГц	0-100,0-400,0-3400	0-100	0-100	10-100	10-100

<sup>1</sup> Вместо емкости задаются допустимые длительности фронтов и спадов.

*Конструктивно* выделяется шина ACCESS.Bus (внешняя), в которой определены разъемы и кабели, а также питание (5 В) устройств от шины; в других — нет.

*Протокольные различия* более существенны, из них отметим лишь следующие.

- ◆ В шине ACCESS.Bus определены лишь передачи данных ведущим устройством, в остальных шинах ведущее устройство может и принимать данные.
- ◆ В шинах ACCESS.Bus и SMBus ведущее устройство не имеет права повторного захвата шины ранее, чем через 50 мкс после начала передачи. Сигнал SCL в этой шине может удерживаться на низком уровне не дольше 2 мс. В PC таких ограничений нет.
- ◆ В шине SMBus ведущее устройство ожидает от ведомого устройства немедленного ответа, в шине ACCESS.Bus ответ приходит независимо, но ожидается не позже чем через 40 мс. В шине SMBus и ведущее устройство, и ведомое устройство могут притормаживать обмен на уровне приема битов, что позволяет им вписываться в свои временные диаграммы (успевать «просыпаться», обрабатывать прерывания).

## 11.2. Шина управления SMI интерфейса МП

В спецификации IEEE 802.3u сетевой технологии Ethernet от 1995 г. появился интерфейс МП (Media Independent Interface — интерфейс, независимый от среды передачи), позволяющий отделять специфические устройства физического уровня (PHY) от вышестоящих уровней. Кроме интерфейса передачи «полезных» данных сети в МП входит последовательный интерфейс SMI (Serial Management Interface), обеспечивающий доступ к внутренним

регистрам РНУ. Эти регистры позволяют отслеживать состояние РНУ и управлять им; в частности, и управлять согласованием режимов сетевых портов. В спецификации описаны регистры, обязательные для устройств, а также отведено место для специфических регистров, используемых по воле разработчика. Интерфейс SMI представляет собой последовательную синхронную трехпроводную шину (сигналы MDC, МОЮ и шина GND), обеспечивающую доступ по записи и чтению к 16-разрядным регистрам устройств. Протокол шины обеспечивает 5-битную адресацию подключаемых устройств, что позволяет непосредственно объединять до 32 устройств на одной шине. Каждое устройство может иметь до 32 регистров. Шина является хост-центрической — ею управляет один контроллер, инициирующий транзакции, а все подключаемые устройства являются ведомыми. Контроллер генерирует импульсы синхронизации по однонаправленной линии MDC, а по двунаправленной линии МОЮ формирует биты команд и данных (при записи) и принимает данные чтения. Устройство, опознавшее свой адрес, должно выполнить адресуемую ему команду. Уровни сигналов — обычные для логики TTL и КМОП; для линии МОЮ обычно используются формирователи с открытым коллектором (стоком) и резисторы, «подтягивающие» пассивную линию к уровню питания (+5 или +3,3 В). Частота импульсов MDC, не обязательно постоянная, не должна превышать 2,5 МГц, что позволяет реализовать функции контроллера чисто программно. Временные диаграммы сигналов для транзакций чтения и записи приведены на рис. 11.4. Отметим, что устройство должно фиксировать биты по положительному перепаду MDC, а контроллер фиксирует биты по отрицательному перепаду. Адреса и данные передаются начиная со старших бит.

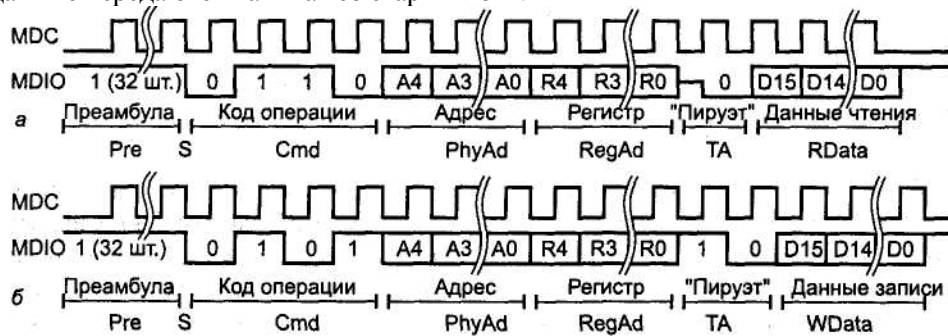


Рис. 11.4. Транзакции на шине SMI: а — запись, б — чтение

В общем случае каждой транзакции должна предшествовать преамбула Preamble — не менее 32 тактов нахождения МОЮ на высоком уровне. Собственно транзакция начинается с условия *Start* — вывода контроллером битовой последовательности «01», за которым следует двухбитная команда Cmd: «10» — чтение или «01» — запись. Далее контроллер передает 5-битный адрес устройства (PhyAd), за которым следует 5-битный номер регистра (RegAd). После этого следует двухбитный «пируэт» (turnaround) TA для перехода к передаче данных. При операции записи «пируэт» сводится к передаче контроллером последовательности «10», а затем он передает 16 бит данных (WData) для записи в регистр. При чтении с начала «пируэта» контроллер отпускает линию МОЮ, а на время второго бита отозвавшееся адресованное устройство устанавливает на этой линии нуль, что используется для определения факта ответа. В следующие 16 тактов устройство передает считанные данные (RData) контроллеру.

Преамбула нужна только для приведения интерфейсной части устройств в исходное состояние (после подачи питания); если питание устройством не терялось, то последующие транзакции могут выполняться и без преамбулы. Имеется также нюанс в «пируэте»: для некоторых устройств второй его бит оказывается лишним — он сбивает синхронизацию данных на 1 такт.

### 11.3. Интерфейс SPI

*Интерфейс SPI* (Serial Peripheral Interface), он же Microwire, — трехпроводный синхронный с отдельными линиями входных и выходных данных — применяется для связи микроконтроллеров с периферийными микросхемами и микросхемами памяти. С его помощью программируются многие микросхемы программируемой логики (например, FPGA фирмы Altera), применяемые в различных узлах ПК и периферии. По сравнению с I<sup>2</sup>C интерфейс SPI обеспечивает более высокую скорость передачи данных. Частота синхронизации может достигать 5 МГц (зависит от подключаемых устройств); при этом за каждый такт синхронизации одновременно может и передаваться, и приниматься очередной бит данных. В основном варианте использования предполагается, что интерфейс соединяет одно ведущее устройство с одним или несколькими ведомыми устройствами. Интерфейс может применяться и для обмена между несколькими микроконтроллерами, допуская наличие на шине нескольких ведущих устройств, но протокол предоставления доступа не стандартизован. Ниже рассматривается работа интерфейса с одним ведущим устройством. В интерфейсе используются 3 обязательных сигнала:

- ◆ SCK (Serial Clock) — синхросигнал, которым ведущее устройство стробирует каждый бит данных;
- ◆ MOSI (Master Output Slave Input) — выходные данные ведущего устройства и входные данные ведомого устройства;
- ◆ MISO (Master Input Slave Output) — входные данные ведущего устройства и выходные данные ведомого устройства.

Кроме того, может использоваться сигнал выбора ведомого устройства SS# (Slave Select, или CS# — Chip Select): ведомое устройство должно реагировать на сигналы интерфейса и генерировать выходные данные на линии MISO только при низком уровне этого сигнала; при высоком уровне выход MISO должен переводиться в высокоимпедансное состояние. С помощью сигналов SS#, отдельно формируемых ведущим устройством для каждого из ведомых устройств, ведущее устройство может выбирать партнером в транзакции одно из ведомых. При этом получается гибридная топология соединений: по сигналам SCK, MOSI и MISO — топология шинная, по SS# — звездообразная (центр — ведущее устройство).

Возможен и иной способ общения ведущего устройства с несколькими ведомыми устройствами с чисто шинной топологией. Здесь сигнал SS# для выбора устройств не используется, но ведомые устройства все время «слушают» шину, ожидая специальной адресной посылки в начале транзакции. Опознав свой адрес, устройство обрабатывает дальнейшие протокольные посылки, и теперь ему разрешается выводить данные на линию MISO до конца текущей транзакции. Однако этот вариант требует довольно сложной логики поведения ведомого устройства.

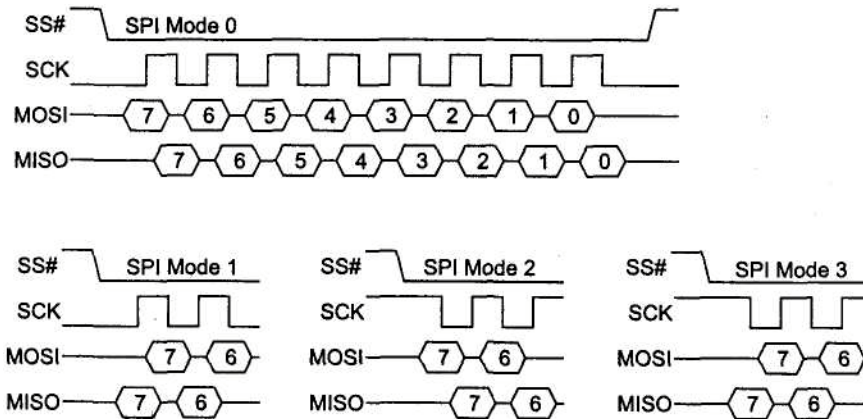


Рис. 11.5. Форматы передач по интерфейсу SPI

Данные передаются старшим битом вперед. Устройства различных производителей могут использовать разные режимы обмена (SPI Mode 0... SPI Mode 3), различающиеся фазой и полярностью синхросигналов. Рис. 11.5 иллюстрирует передачу и прием байта в различных режимах. Заметим, что в любом режиме входные и выходные данные стробируются по разным перепадам синхроимпульсов. Кроме того, есть варианты в использовании сигнала SS#. В первом варианте (он и представлен на рисунке) для ведомого устройства признаком начала транзакции является спад сигнала SS#, а завершения — его подъем. Во втором варианте начало транзакции определяется по первому фронту (спаду) SCK, а сигнал SS# может иметь низкий уровень постоянно. Второй вариант иногда предпочитают для простейшего двухточечного интерфейса с единственным ведущим устройством и единствен-

ным ведомым устройством, когда нет опасения потери синхронизации. Микроконтроллеры с аппаратным портом SPI, как правило, имеют возможность программного выбора режима. Форматы транзакций зависят от устройств, но общая идея такова: ведущее устройство по линии MOSI посылает код операции и адресную информацию для устройства; затем следуют данные. В операции записи они передаются ведущим устройством по той же линии, в операции чтения устройство их посылает по линии MISO. На время всей транзакции ведущее устройство сохраняет активное состояние линии SS#, число требуемых синхроимпульсов зависит от формата команды.

## 11.4. Интерфейс JTAG

Интерфейс JTAG (Joint Test Action Group) предназначен для тестирования сложных логических схем, установленных в целевое устройство. Этот интерфейс описан в стандарте IEEE 1149.1 Boundary Scan Architecture и является последовательным синхронным, но в связи со спецификой назначения значительно отличается от вышеописанных интерфейсов, ориентированных на передачу данных. Интерфейсом JTAG управляет одно устройство-контроллер (чаще всего это ПК с соответствующим интерфейсным адаптером), к которому может быть подключено несколько тестируемых устройств. Ниже перечислены сигналы интерфейса JTAG.

- ◆ TCK (Test Clock) — сигнал синхронизации последовательных данных; генерируется контроллером. Частота синхронизации может достигать 16 МГц.
- ◆ IMS (Test Mode Select) — сигнал выбора тестового режима. Генерируется контроллером.
- ◆ TDI (Test Data Input) — входные данные, принимаемые устройством в последовательном двоичном коде (младшим битом вперед).
- ◆ TOO (Test Data Output) — выходные данные, передаваемые устройством в последовательном двоичном коде.
- ◆ TRST (Test Logic Reset) — необязательный сигнал сброса логики интерфейсного порта, генерируемый контроллером.

Эти сигналы (все однонаправленные), имеющие обычные логические уровни, образуют тестовый порт TAP (Test Access Port), через который тестируемое устройство подключается к тестирующему оборудованию (контроллеру). В задачу тестирующего оборудования входит формирование тестовых сигналов по программе тестирования, определенной разработчиком тестируемого устройства, и сравнение полученных результатов с эталонами. Один и тот же контроллер и порт могут использоваться для тестирования любого числа устройств, поддерживающих JTAG. Для этого устройства своими портами TAP соединяются в цепочку (рис. 11.6). Стандартизованный логический формат позволяет контроллеру независимо общаться с каждым из устройств цепочки (для этого, конечно, они должны иметь исправные ячейки JTAG).

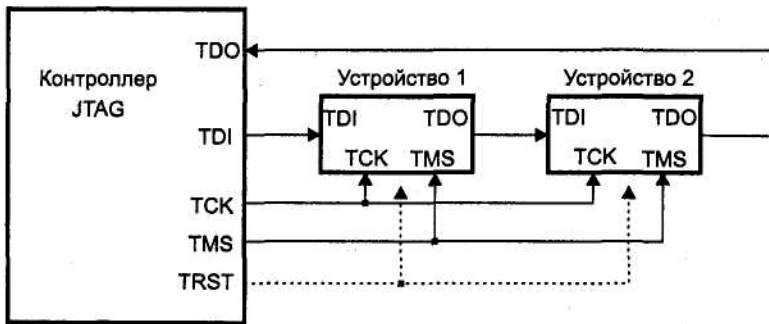


Рис. 11.6. Цепочка устройств с интерфейсом JTAG

Идею тестирования любой цифровой схемы иллюстрирует рис. 11.7, на котором показана условная цифровая система, имеющая входные, выходные (возможно, с третьим состоянием) и двунаправленные сигналы. Ячейки тестирования *B/S* врезаются между реальными внешними выводами устройства и собственно логическим устройством, то есть располагаются на логической границе (*boundary*) устройства. TAP-контроллер способен сканировать ячейки — управлять ими и считывать с них информацию. Отсюда и пошло название *Boundary Scan*, которое можно перевести как «периферийное сканирование». При включенном тестовом режиме TAP-контроллер может логически отсоединить сигналы от внешних выводов, задавать входные воздействия и считывать результаты — собственно, это все, что необходимо для тестирования последовательных схем (автоматов с памятью). Прелесть JTAG заключается в том, что независимо от сложности устройства оно тестируется с помощью всего лишь четырех сигналов — все сложности прячутся в достаточно простые ячейки, «окутывающие» его сигнальные выводы.

Тестовая логика, встраиваемая в устройство, поддерживающее JTAG, состоит из следующих элементов:

- ◆ тестовый порт TAP (четыре интерфейсных сигнала);
- ◆ TAP-контроллер, управляющий тестовыми регистрами;
- ◆ регистр инструкций IR (*Instruction Register*), который принимает последовательный код со входа TDI (код инструкции используется для выбора выполняемой тестовой операции или регистра тестовых данных, к которым производится обращение);
- ◆ регистры тестовых данных, из которых любое устройство должно иметь три обязательных регистра: B PR (*Bypass Register*), DID (*Device Identification Register*) и BSR (*Boundary Scan Register*).

Регистры инструкций и данных представляют собой независимые сдвиговые регистры, соединенные параллельно. На их входы (старшие биты) приходит сигнал TDI, с выходов (младшие биты) снимается сигнал TDO. По каждому положительному перепаду данные продвигаются на один бит.

Регистр B P R имеет длину в один бит. Он используется как кратчайший обходной путь для последовательных данных, когда остальные регистры не участвуют в обмене.

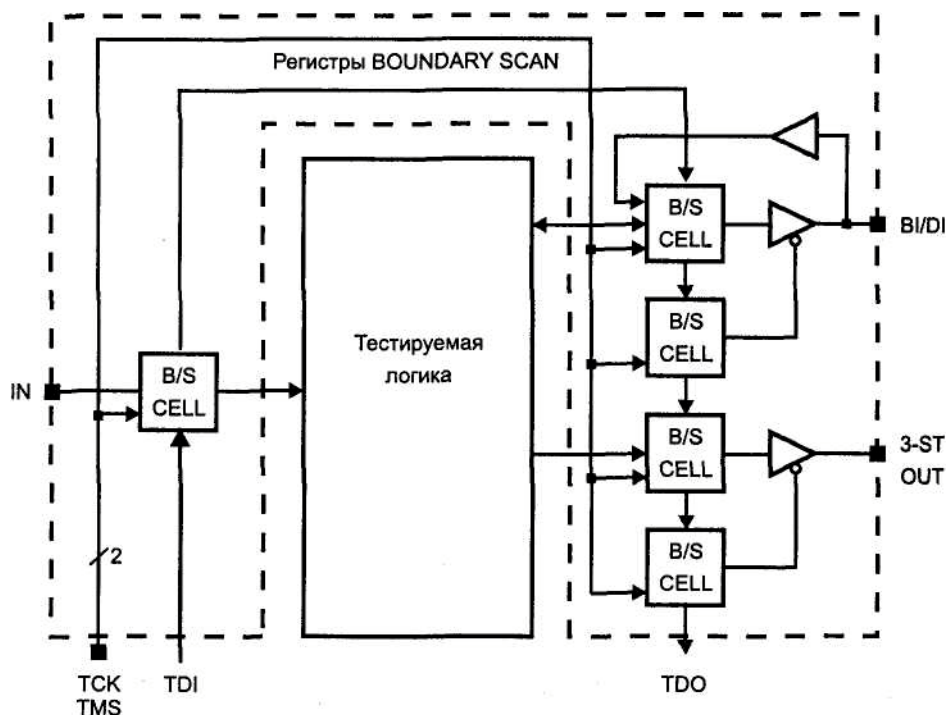


Рис. 11.7. Включение ячеек сканирования

Регистр BSR представляет собой длинный сдвигающий регистр, каждым битом которого являются пограничные ячейки, установленные на всех входных и выходных сигналах процессора. Для двунаправленных сигналов (или их групп), кроме собственно информационных ячеек регистра, соответствующих внешним сигналам, имеются и управляющие ячейки, задающие режим работы информационных ячеек. К примеру, у процессоров P6 длина BSR составляет 159 бит.

Регистр DID длиной 32 бита содержит идентификатор производителя, код устройства и номер версии, по которым TAP-контроллер может распознать, с каким устройством он имеет дело.

Регистр IR служит для хранения исполняемой тестовой инструкции. Его длина зависит от тестируемого устройства. Для всех устройств обязательными являются инструкции BYPASS, IDCODE, SAMPLE и EXTEST.

Инструкция BYPASS (все биты кода — единичные) предназначена для подключения однобитного обходного регистра, обеспечивая скорейшее прохождение данных через устройство; при этом оно никак не реагирует на проходящий поток. Вход TDI обычно «подтягивают» резистором к высокому уровню, при этом разрыв цепочки JTAG приведет к подключению обходных регистров во всех устройствах после точки обрыва. Это исключает возможные непредсказуемые действия устройств в случае обрыва.

Инструкция идентификации IDCODE (младшие биты кода — 10) подключает к интерфейсу регистр DID, позволяя считать его содержимое (поступающие входные данные не могут изменить его значение).

Инструкция SAMPLE/PRELOAD (младшие биты кода — 01) имеет два назначения. Когда TAP-контроллер находится в состоянии *Capture-DR* (см. ниже), эта инструкция позволяет выполнить моментальный «снимок» состояния всех внешних сигналов без воздействия на работу устройства. Значение сигналов фиксируется по положительному перепаду TCK. В состоянии *Update-DR* по этой инструкции данные загружаются в выходные ячейки тестового порта (но еще не на выходы устройства), откуда впоследствии они будут выводиться (подаваться на выходы процессора) по инструкции EXTEST. Данные загружаются по спаду сигнала TCK.

Инструкция EXTEST (младшие биты кода — 00) предназначена для проверки внешних цепей (по отношению к тестируемому устройству). При этом на выходные выходы подаются сигналы, предварительно записанные в регистр BSR, а состояние входных сигналов фиксируется в этих регистрах. Двухнаправленные сигналы предварительно конфигурируются соответствующими им управляющими битами ячеек BSR.

Стандарт 1149.1 предусматривает и инструкцию тестирования внутренней логики устройства INTEST, но ее поддерживают не все устройства.

*Контроллер тестового порта* (TAP-controller) представляет собой синхронный конечный автомат, изменяющий состояние по фронту сигнала TCK и по включению питания. Сменой состояний управляет сигнал TMS (Test Mode State), воспринимаемый по положительному перепаду TCK. Граф состояний и переходов управляющего автомата представлен на рис. 11.8. Около стрелок переходов указаны значения сигнала TMS во время фронта TCK.

В исходное состояние *Test-Logic-Reset* контроллер автоматически переходит по включении питания и из любого другого состояния может быть переведен высоким уровнем TMS, удерживаемым не менее пяти тактов TCK. Для перевода в состояние *Test-Logic-Reset* иногда используют и дополнительный сигнал TRST. В этом состоянии тестовая логика запрещена, и устройство работает в нормальном режиме.

Состояние *Run-Test/Idle* является промежуточным между выполнением тестовых операций. В этом состоянии регистры не изменяют своего значения.

В состоянии *Capture-DR* во время выполнения инструкций EXTEST и SAMPLE/PRELOAD сканирующий регистр фиксирует только данные на входных линиях.

В состоянии *Shift-DR* данные с TDI продвигаются через подключенный сдвиговый регистр на выход TDO.

В состоянии *Pause-DR* контроллер временно запрещает продвижение данных через сдвиговый регистр.

В состоянии *Update-DR* по спаду TCK сигналы из сдвигового регистра фиксируются на выходах тестовых ячеек.

В состоянии *Capture-IR* контроллер загружает в сдвиговый регистр инструкций код «безобидной» инструкции SAMPLE.



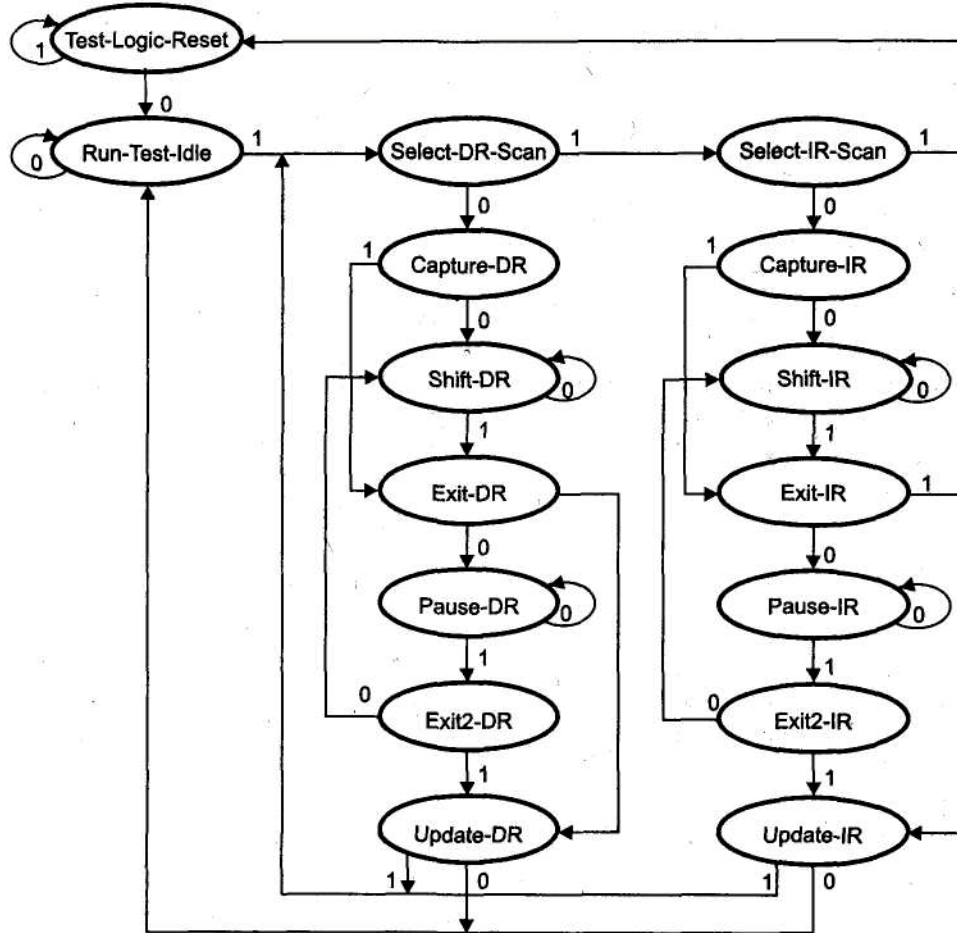


Рис. 11.8. Граф состояний и переходов контроллера TAP

В состоянии *Shift-IR* в цепь между TDI и TDO включается сдвиговый регистр инструкций, но еще выполняется предыдущая инструкция.

В состоянии *Pause-IR* контроллер временно запрещает продвижение данных через сдвиговый регистр инструкций.

В состоянии *Update-IR* по спаду TCK фиксируется новая исполняемая инструкция, и в цепь TDI-TDO включается соответствующий ей регистр.

Кроме этих основных состояний контроллера, определяющих действия тестового оборудования, имеются и временные промежуточные состояния, необходимые для реализации переходов автомата. К ним относятся *Select-DR-Scan*, *Exit1-DR*, *Exit2-DR*, *Select-DR-Scan*, *Exit1-IR* и *Exit2-IR*.

Для интерфейса JTAG существует специальный язык описания устройств *BSDL* (Boundary Scan Description Language). Состав и порядок следования информации-

онных и управляющих ячеек в сдвиговом регистре данных специфичен для каждого устройства (для чего и существует идентификационный регистр) и сообщается его разработчиками.

Интерфейс JTAG используется не только для тестирования, но и для программирования различных устройств, в том числе и энергонезависимой памяти микроконтроллеров. Контакты для сигналов JTAG имеются на шине PCI, однако в их использовании единообразия не наблюдается (либо остаются неподключенными, либо соединяются для организации цепочки). Интерфейс JTAG имеется в современных процессорах; здесь он позволяет не только тестировать сам процессор (это не представляет особого прикладного интереса), но и организовать зондовый режим отладки (probe mode, см. [6, 7]). Зондовый режим является мощным средством отладки системного программного обеспечения; обычный процессор, связанный с тестовым контроллером интерфейсом JTAG, превращается во внутрисхемный эмулятор — мечту разработчика системного ПО.

## 11.5. Программно-управляемая реализация последовательных интерфейсов

Рассмотренные выше интерфейсы — PC, SMBus, SMI, SPI и JTAG — имеют общее свойство: они управляются и синхронизируются контроллером и не требуют фиксированной частоты синхронизации. Это позволяет для многих применений программно реализовать их на любом компьютере или микроконтроллере. На рис. 11.9 приведена схема простейшего адаптера интерфейсов I<sup>2</sup>C, SMBus или SMI для LPT-порта. Здесь сигнал SCL (или MDC в SMI) формируется непосредственно от выходной линии Strobe; для его переключения достаточно последовательно записывать в бит 0 регистра CR (порт с адресом LPT\_BASE+2) нули (высокий уровень сигнала) и единицы (низкий). Двухнаправленный сигнал SDA (МОЮ в SMI) реализуется чуть сложнее: для передачи нуля в бит 1 регистра CR (LPT\_BASE+2) записывается единица, для передачи единицы — нуль. На время чтения в этот бит должен быть записан нуль (чтобы на выходе был высокий уровень), данные (инверсные) считываются из бита 7 регистра SR (LPT\_BASE+1). В адаптере должен использоваться диод с малым прямым падением напряжения. Лучше всего для этого подходят германиевые меза-диоды Д310 или Д311; кремниевые диоды, даже с барьерами Шоттки, здесь работают хуже (возможен слишком высокий формируемый уровень нуля).

Программная реализация протоколов в среде DOS и Windows 9x не вызывает особых затруднений» поскольку здесь программист может обращаться к регистрам порта непосредственно. В более защищенных ОС (UNIX, Linux, Windows NT/ 2000) для обращения к регистрам LPT-порта требуются вызовы API.

При использовании данных адаптеров следует внимательно относиться к проблемам защиты от помех. Непосредственно с адаптера сигналы можно передавать

лишь на небольшие расстояния (десятки сантиметров). Для передачи на большие расстояния (если ПК не приблизить к объекту) подойдут преобразователи уровней сигналов ТТЛ в дифференциальные сигналы интерфейса RS-422 и обратно. При этом адаптер разделяется на 2 блока, соединенные кабелем, — первый блок устанавливается на LPT-порт, другой — около подключаемого объекта.

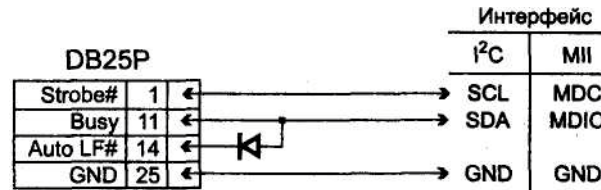


Рис. 11.9. Адаптер трехпроводных интерфейсов для LPT-порта

Несколько сложнее получаются адаптеры этих интерфейсов для COM-портов. Здесь можно программно генерировать сигналы DTR и RTS и считывать состояния сигналов CIS, DSR, DCD и RI. Усложнение заключается в необходимости использования преобразователей уровней сигналов RS-232C в ТТЛ (для этого выпускается ряд микросхем, например фирмами Maxim, Сурех), для которых требуется подача питания.

Для интерфейсов SPI и JTAG тоже можно использовать LPT-порт, здесь даже не требуется организация двунаправленных линий. Сигналы SCK и MOSI (SPI), TCK, TMS, T00 и TRST (JTAG) можно подключить к любым выходным линиям порта, MISO (SPI), TDI (JTAG) — к любой входной линии. Простейший адаптер содержит лишь последовательные резисторы 100-150 Ом, включенные в сигнальные цепи для уменьшения «звона» и влияния кабеля на схему. Однако такая схема может надежно работать лишь при небольшой длине кабеля (20-30 см), что не всегда удобно. Более надежна и удобна схема с буферами, способными переходить в высокоимпедансное состояние (например, 74НС244 или 1556АП5). Плата с буфером может соединяться с БРТ-портом довольно длинным кабелем, а от нее к устройству идет короткий кабель. Программно-управляемый перевод буфера выходных сигналов в третье состояние позволяет логически отключать адаптер от программируемой схемы, что особенно удобно в процессе отладки программируемых устройств. Именно так устроен популярный адаптер «ByteBlaster», применяемый для программирования конфигурируемой логики фирмы Altera и других. Адаптер вместе с ПО может обеспечивать протокол SPI (быть ведущим устройством в варианте с двухточечной топологией), JTAG и собственный протокол программирования устройств «Serial Passive». Схемы различных адаптеров и ПО для них можно легко найти в Сети.

Программная реализация последовательных протоколов ограничивает скорость передачи данных на уровне 50-150 Кбит/с при работе LPT-порта в стандартном режиме. В режиме ЕРР или ЕСР можно достичь скорости и 1-2 Мбит/с, но

#### 11.5. Программно-управляемая реализация последовательных интерфейсов 453

при этом адаптер несколько усложняется (поскольку в этих интерфейсах требуется одновременно принимать и передавать данные). Решить проблему производительности, а заодно и расширить функциональные возможности позволяет использование специализированных интерфейсных адаптеров для шин PCI или ISA, выпускаемых рядом фирм. Есть и внешние устройства с интерфейсами USB или Ethernet. Правда, цена этих адаптеров и устройств существенно отличается от цены простого адаптера, который можно изготовить и самостоятельно.

## Глава 12

# Архитектурные компоненты IBM PC-совместимого компьютера

Аппаратные интерфейсы, описанные в книге, в IBM-PC-совместимом компьютере «живут» в специфическом архитектурном окружении. Эту специфику приходится учитывать при проектировании аппаратной части устройств, чтобы обеспечить с ними эффективное программное взаимодействие. В этой главе вкратце рассматриваются особенности процессоров x86 и связанные с этими особенностями распределение памяти, организация ввода-вывода и прерываний. Здесь же рассматривается традиционный контроллер DMA, системные средства измерения времени, а также способы внедрения собственных расширений BIOS и нетрадиционной (бездисковой) загрузки ПО в специализированные компьютеры на базе IBM PC.

### 12.1. Пространство физической памяти

Основную часть физического адресного пространства PC занимает *оперативная память* (ОЗУ), начинающаяся с нулевого адреса. В нее вклинивается область адресов A0000h-FFFFh — *Upper Memory Area* (UMA), 384 Кбайт — *верхняя* память, зарезервированная со времен IBM PC для системных нужд. В UMA размещаются области буферной памяти адаптеров шины (E)ISA (например, видеопамять) и постоянная память (BIOS с расширениями). ОЗУ продолжается и за область UMA. Под самой верхней границей физического адресного пространства имеется образ памяти системной ROM BIOS.

Для доступности сервисов BIOS в реальном режиме все ПК имеют образ ROM BIOS в адресах E0000h-FFFFh или F0000h-OFFFFh. Кроме того, образ BIOS должен находиться и под самой верхней границей адресного пространства, поскольку все процессоры x86 по аппаратному сбросу стартуют с адреса начала последнего параграфа памяти (FFFF0h - 8086/88, FFFFF0h - 80286 и 386SX, FFFFFFF0h - 386DX и выше с 32-разрядной шиной адреса, FFFFFFFF0h - P6 и выше с 36-разрядной шиной адреса).

Для компьютеров класса AT-286 и 386SX с 24-битной шиной адреса верхняя граница оперативной памяти — FDFFFFh (максимальный размер 15,9 Мбайт). Область

FEOOOOh-FFFFFFh содержит образ ROM BIOS, обращение к этой области эквивалентно обращению к ROM BIOS по адресам OE0000h-OFFFFFFh. Для ПК на процессорах 386DX и выше с 32-битной шиной адреса теоретический предел объема ОЗУ — почти 4 Гбайт, верхний образ BIOS находится в адресах FFFEOOOOh-FFFFFFFh. Для ПК на процессорах P6+ с 36-битной шиной адреса предел объема ОЗУ — почти 64 Гбайт и верхний образ BIOS находится в адресах FFFFEOOOOh-FFFFFFFh.

Области физических адресов, не занятые ОЗУ и ROM BIOS, могут быть использованы устройствами шин (E)ISA и PCI (AGP).

Для памяти адаптеров, устанавливаемых в шину ISA, безусловно доступна часть области адресов UMA A0000h-EFFFFh или A0000h-DFFFFh (до начала ROM BIOS). В этой области располагаются и модули расширений BIOS (см. п. 12.7.1). Карты ISA могут иметь память и в области FE000h-FFFFFFh, но она программно доступна лишь в защищенном (и большом^реальном) режимах процессора. Для отображения этой области памяти на шину ISA (а не ОЗУ) в CMOS Setup предусмотрен параметр Memory Hole At 15-16M, но его включение не позволит использовать оперативную память свыше 15 Мбайт.

Поскольку шина ISA имеет 24-разрядную шину адреса, ведущие устройства этих

шин (ISA Bus Master) способны обращаться к памяти (ОЗУ и память адаптеров)

в пределах первых 16 Мбайт (000000-FFFFFFh). То же ограничение касается и стандартного контроллера DMA, которым могут пользоваться устройства шины

ISA (и иные устройства системной платы).

л

Шина PCI имеет 32-разрядную шину адреса, так что ее ведущим устройствам доступна вся физическая память<sup>1</sup>. Для устройств PCI могут выделяться любые области адресов, свободные от ОЗУ, ROM BIOS и устройств ISA. Области адресов памяти, используемые каждым устройством PCI, описаны в заголовках их конфигурационных пространств. Эти данные требуются при распределении ресурсов и настройке мостов PCI в процессе инициализации шины.

## 12.2. Пространство ввода-вывода

Пространство ввода-вывода в IBM PC, как и в процессорах x86, отделено от пространства памяти. У всех этих процессоров, в том числе и 32-разрядных, в пространстве ввода-вывода используется 16-разрядная адресация (диапазон адресов 0-FFFFh). Для дешифрации адресов портов в оригинальном PC из 16 бит использовались только младшие 10 (A0-A9), что обеспечивает обращением портам в диапазоне адресов 0-3Fh. Старшие биты адреса, хотя и поступают на шину, устройствами игнорируются. В результате обращения по адресам, к примеру 378h, 778h, B78h и F78h, будут восприниматься устройствами одинаково. Это упрощение, нацеленное на снижение стоимости как системной платы, так и схем плат адаптеров, для шины ISA никто не отменял. Традиционные адаптеры для шины ISA, называемые *Legacy Card* («наследие тяжелого прошлого»), для старших бит адреса не имеют

<sup>1</sup> Для пока еще экзотических вариантов системных плат с памятью более 4 Гбайт и 36-разрядной физической адресацией последних моделей процессоров PCI может использовать двухадресные циклы обращения с 64-битным адресом.

<sup>1</sup> даже печатных ламелей на своем краевом разъеме. Впоследствии перешли к 12-битной адресации устройств шины ISA, но ее приходится применять с оглядкой на возможное присутствие устройств с 10-битной адресацией. В адаптерах для шин MCA и PCI и во всех современных системных платах используются все 16 бит адреса. Карта распределения адресов ввода-вывода стандартных устройств PC приведена в табл. 12.1. Эта карта подразумевает 10-битную дешифрацию адреса. Естественно, что в конкретном компьютере реально присутствуют не все перечисленные устройства, но в то же время там могут оказаться другие, не попавшие в таблицу.

**Таблица 12.1.** Стандартная карта портов ввода-вывода АТ и

PS/2	PC/XT	Назначение
000-00F	000-00F	Контроллер DMA #1 8237
010-01F		PS/2 — расширение DMA#1
020-021	020-021	Контроллер прерываний #1 — 8259A
040-05F	040-043	Таймер (PC/XT: 8253, AT: 8254)
060	060	Диагностический регистр POST (только запись)
	060-063	Системный интерфейс 8255
060,064		Контроллер клавиатуры AT 8042
061		Источники NMI и управление звуком
070-07F		Память CMOS и маска NMI
080		Диагностический регистр-
080-08F	080-083	Регистры страниц DMA
090-097		PS/2 микроканал, арбитр
	0A0	Маска NMI
0A0-0A7		Контроллер прерываний #2 — 8259A
0C0-0C7		Контроллер DMA #2 8237A-5
0E0-0FF		Сопроцессор 80287
100-1EF		PS/2 управление микроканалом
170-177		Контроллер НЖМД #2 (IDE#2)
1F0-1F7		Контроллер НЖМД #1 (IDE#1)
200-207	200-20F	Игровой адаптер
	210-217	Блок расширений
238-23F		COM4
278-27F	278-27F	Параллельный порт LPT2 (LPT3 при наличии MDA)
	2A2-2A3	4асbiMSM48321RS
2C0-2DF	2C0-2DF	EGA #2
2E0-2E7		COM4
2E8-2EF		COM4
2F8-2FF	2F8-2FF	COM2
300-31F		Плата прототипа
	320-32F	Жесткий диск XT
338-33F		COM3 <
370-377		Контроллер НГМД #2
376-377		Порты комайд IDE#2
378-37F	378-37F	Параллельный порт LPT1 (LPT2 при наличии MDA)

АТ и PS/2	PC/XT	Назначение
380-38F	380-38F	Синхронный адаптер SDLC/BSC #2
3АО-3AF	3АО-3A9	Синхронный адаптер BSC#1
3В0-3ВВ	3В0-3ВВ	Монохромный адаптер (MDA)
3В4-3С9		PS/2 видеосистема
3ВС--3ВF	3ВС-3ВF	Параллельный порт LPT1 платы MDA
3СО--3СF	3СО-3СF	ЕGA#1
3СО-3Df	3СО-3Df	VGA
3D0-3Df	3D0-3Df	CGA/EGA
3Е0-3Е7		COM3
3Е8-3ЕF		COM3
3F0-3F7	3F0-3F7	Контроллер НГМД #1
3F6-3F7		Порты команд IDE*1
3F8-3Ff	3F8-3Ff	COM1

Каждой шине назначается своя область адресов ввода, поэтому дешифратор адресов, расположенный на системной плате, при чтении открывает соответствующие буферы данных, так что реально считываться будут данные только с одной шины. При записи в порты данные (и сигнал записи) могут распространяться по всем шинам компьютера. В стандартном распределении адреса 0h-OFFh отведены для устройств системной платы. При наличии (и разрешении работы) периферийных устройств на системной плате чтение по этим адресам не распространяется на шины расширения. Для современных плат со встроенной периферией и несколькими шинами (ISA, PCI) распределением адресов управляет BIOS через регистры конфигурирования чипсета.

## 12.3. Аппаратные прерывания

Аппаратные прерывания обеспечивают реакцию процессора на события, происходящие асинхронно по отношению к исполняемому программному коду. Прерывания в процессорах x86 подробно рассмотрены в литературе [6,7]. Здесь напомним, что аппаратные прерывания делятся на маскируемые и немаскируемые. На *немаскируемое прерывание* (NMI) процессор реагирует всегда (если обслуживание предыдущего NMI завершено); этому прерыванию соответствует фиксированный вектор 2. Немаскируемые прерывания в PC используются для сигнализации о фатальных аппаратных ошибках. Сигнал на линию NMI приходит от схем контроля паритета памяти, от линий контроля шины ISA (IOCHK) или шины PCI (SERR#). Сигнал NMI блокируется до входа процессора установкой в 1 бита 7 порта 070h, отдельные источники разрешаются и идентифицируются битами порта 061h:

- ◆ бит 2 R/W — ERP — разрешение контроля ОЗУ и сигнала SERR# шины PCI;
- ◆ бит 3 R/W — EIC — разрешение контроля шины ISA;
- ◆ бит 6 R — IOCHK — ошибка контроля на шине ISA (сигнал IOCHK#);
- ◆ бит 7 R — PCK — ошибка четности ОЗУ или сигнал SERR\* на шине PCI.



Реакция процессора на *маскируемые прерывания* может быть задержана сбросом его внутреннего флага IF (инструкции CLI — запретить прерывания, STI — разрешить). По возникновении события, требующего реакции, адаптер (контроллер) устройства формирует запрос прерывания, который поступает на вход контроллера прерываний. Контроллер прерываний формирует общий запрос маскируемого прерывания для процессора, а когда процессор подтверждает этот запрос, контроллер сообщает процессору вектор прерывания, по которому выбирается программная процедура обработки прерываний. Процедура должна выполнить действия по обслуживанию данного устройства, включая сброс его запроса для обеспечения возможности реакции на следующие события и посылку команды завершения в контроллер прерываний. Вызывая процедуру обработки, процессор автоматически сохраняет в стеке значение всех флагов и сбрасывает флаг IF, что запрещает маскируемые прерывания. При возврате из процедуры (по инструкции I RET) процессор восстанавливает сохраненные флаги, в том числе и установленный IF, что снова разрешает прерывания. Если во время работы обработчика прерываний требуется реакция на иные прерывания (более приоритетные), то в обработчике должна присутствовать инструкция STI. Особенно это касается длинных обработчиков; здесь инструкция STI должна вводиться как можно раньше, сразу после критической (не допускающей прерываний) секции. Следующие прерывания того же или более низкого уровня приоритета контроллер прерываний будет обслуживать только после получения команды завершения прерывания EOI (End Of Interrupt).

Маскируемые прерывания используются для сигнализации о событиях в устройствах. Обработка сигналов запросов прерывания выполняется контроллером прерываний, программно совместимым с 8259A. Контроллер прерываний позволяет маскировать отдельные входы запросов и организовывать систему приоритетов запросов от различных входов. В машинах класса AT применяется каскадное соединение двух контроллеров. Ведущий контроллер 8259A#1 обслуживает запросы 0, 1, 3-7; его выход подключается к входу запроса прерываний процессора. К его входу 2 подключен ведомый контроллер 8259A#2, который обслуживает запросы 8-15. При этом поддерживается вложенность приоритетов — запросы 8-15 со своим рядом убывающих приоритетов вклиниваются между запросами 1 и 3 ведущего контроллера, приоритеты запросов которого также убывают с ростом номера. В XT каскадирование не применялось, и один контроллер 8259A обслуживал все 8 линий запросов.

Контроллер прерываний 8259A подробно описан в литературе [1, 7]; здесь приведем лишь необходимые сведения, в большинстве случаев достаточные для работы. Контроллеры расположены по адресам 20-21h (8259A#1) и A0-A1h (8259A#2), обращаться к ним следует как к однобайтным портам ввода-вывода. После инициализации (процедурой POST и при загрузке ОС) все неиспользуемые входы контроллеров замаскированы (на запросы прерываний реагировать не будут), а их векторы прерываний указывают на «заглушку» — процедуру с единственной инструкцией IRET. Первым делом программа должна загрузить в память свой обработчик и подставить указатель на его начало в соответствующее место таблицы прерываний. Далее следует размаскировать вход, для чего выполняется чтение регистра маски (адрес 21h для 8259A#1, A1h для 8259A#2), обнуление соответ-

ствующего бита (см. табл. 12.2) и запись в регистр нового значения маски. При работе с контроллером прерываний от программы требуется лишь управление маской своего запроса (при инициализации программы нужно обнулить маску требуемого запроса) и корректное завершение обработки прерываний. Каждая процедура обработки аппаратного прерывания должна завершаться командой EOI (End Of Interruption), посылаемой контроллеру:

- ◆ для 1-го контроллера — посылка байта 20h по адресу 20h;
- ◆ для 2-го контроллера — посылка байта 20h по адресу A0h (EOI для ведомого контроллера), затем посылка байта 20h по адресу 20h (EOI для ведущего контроллера).

Некорректно завершенная процедура не позволит повторно использовать данный или другие запросы прерываний. Если обработчик прерывания удаляется из памяти, предварительно должен быть замаскирован соответствующий ему вход контроллера. Все изменения в таблице прерываний должны выполняться при замаскированных прерываниях, чтобы избежать попытки использования вектора в процессе его модификации (это приведет к «вылету» программы).

На входы контроллеров прерываний поступают запросы от системных устройств (клавиатура, системный таймер, CMOS-таймер, сопроцессор), периферийных контроллеров системной платы и от карт расширения. Традиционно все линии запросов, не занятые перечисленными устройствами, присутствуют на всех слотах шины ISA/EISA. Эти линии обозначаются как IRQ<sub>x</sub> и имеют общепринятое назначение (табл. 12.2). Часть этих линий отдается в распоряжение шины PCI. В таблице отражены и приоритеты прерываний — запросы расположены в порядке их убывания. Номера векторов, соответствующих линиям запросов контроллеров, система приоритетов и некоторые другие параметры задаются программно при инициализации контроллеров. Эти основные настройки остаются традиционными для обеспечения совместимости с программным обеспечением.

**Таблица 12.2.** Аппаратные прерывания (в порядке убывания приоритета)

Имя (номер <sup>1</sup> )	Вектор	Контроллер/маска	Описание
NMI	02h		Контроль канала, паритет (в XT — сопроцессор)
IRQ0	08h	#1/1h	Таймер (канал £8253/8254)
IRQ1	09h	#1/2h	Клавиатура
IRQ2	0Ah	#1/4h	XT — резерв, AT — недоступно (подключается каскад IRQ8-IRQ15)
IRQ8	70h	#2/1 h	CMOS RTC — часы реального времени
IRQ9	71h	#2/2h	Резерв
IRQ10	72h	#2/4h	Резерв
IRQ11	73h	#2/8h	Резерв
IRQ12	74h	#2/10h	PS/2-Mouse (резерв)
IRQ 13	75h	#2/20h	Математический сопроцессор

*продолжение*^

Таблица 12.2 (продолжение)

Имя (номер <sup>1</sup> )	Вектор	Контроллер/маска	Описание
IRQ14	76h	#2/40h	НОС — контроллер НЖМД
IRQ15	77h	#2/80h	Резерв
IRQ3	0Bh	#1/4h	COM2, COM4
IRQ4	0Ch	#1/10h	COM1, COM3
IRQ5	0Dh	#1/20h	XT — НОС, AT — LPT2, Sound (резерв)
IRQ6	0Eh	#1/40h	FDC — контроллер НГМД
IRQ7	0Fh	#1/80h	LPT1 — принтер

<sup>1</sup> Запросы прерываний 0, 1, 8 и 13 на шины расширения не выводятся.

Для запросов прерывания с шины PCI используются 4 линии запросов прерывания, которые обозначают как INTR A, B, C, D. Эти линии работают по низкому уровню, что дает возможность их разделения (совместного использования). Линии циклически сдвигаются в слотах и независимо коммутируются на доступные линии IRQx с помощью конфигурационных регистров чипсета. Линии IRQx, используемые шиной PCI, становятся недоступными для шины ISA. «Дележку» линий между шинами, а также управление чувствительностью отдельных линий обеспечивают параметры CMOS Setup, а также система PnP. В параметрах ISA или Legacy подразумевают использование линий IRQx традиционными адаптерами шины ISA (статическое распределение), а PCI/PnP — использованием адаптерами шины PCI или адаптерами PnP для шины ISA (динамическое распределение). Общая схема формирования запросов прерываний изображена на рис. 12.1.

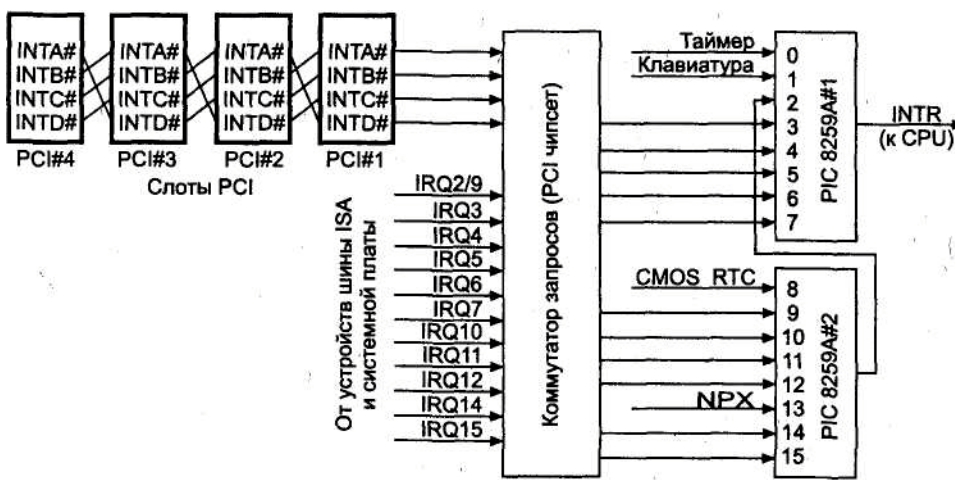


Рис. 12.1. Коммутация запросов прерываний

Каждому устройству, для поддержки работы которого требуются прерывания, должен быть назначен свой номер прерывания. Назначения номеров прерываний выполняются с двух сторон: во-первых, адаптер, нуждающийся в прерываниях,

должен быть сконфигурирован на использование конкретной линии шины (джамперами или программно). Во-вторых, программное обеспечение, поддерживающее данный адаптер, должно быть проинформировано о номере используемого вектора. В процессе назначения прерываний может участвовать система PnP для шин ISA и PCI, для распределения линий запросов между шинами служат специальные параметры CMOS Setup.

Контроллер прерываний позволяет программировать свои входы на чувствительность к уровню или перепаду сигнала.

- ♦ *Чувствительность к уровню* (level sensitive) означает, что контроллер прерываний вырабатывает запрос прерывания процессора по факту обнаружения определенного уровня (на ISA — высокого) на входе DRQx. Если к моменту завершения обработки этого запроса (после записи команды EOI в регистр контроллера прерываний) контроллер снова обнаруживает активный уровень на том же входе DRQx, то он снова сформирует запрос на прерывание процессора.
- ♦ *Чувствительность к перепаду* (edge sensitive) означает, что контроллер прерываний вырабатывает запрос прерывания процессора только по факту обнаружения перепада (на ISA — положительного) на входе DRQx. Повторно запрос по этому входу возможен только по следующему такому же перепаду, то есть сигнал предварительно должен вернуться в исходное состояние.

В любом случае сигнал запроса аппаратного прерывания IRQx должен удерживаться генерирующей его схемой, по крайней мере, до цикла подтверждения прерывания процессором. В противном случае источник прерывания корректно идентифицирован не будет, и контроллер сообщит ложный вектор прерывания (spurious interrupt), соответствующий его входу с максимальным номером (IRQ7 для первого контроллера и IRQ 15 для второго). Обычно адаптеры строят так, что сигнал запроса сбрасывается при обращении программы обслуживания прерывания к соответствующим регистрам адаптера.

В шине ISA прерывание вырабатывается по *положительному перепаду сигнала* на линии запроса. Это плохо по двум причинам: такой способ подачи сигнала, во-первых, имеет меньшую помехозащищенность, чем срабатывание по отрицательному перепаду, во-вторых, отсекает путь к нормальному разделению линий запросов (см. ниже), для которого полностью пригоден способ подачи сигнала по низкому уровню. Поскольку традиционный контроллер позволяет задавать чувствительность — *уровень* (Level) или *перепад* (Edge) — только для всех входов одновременно, в общем случае разделяемые прерывания на шине ISA вместе с корректной работой системных устройств использоваться не могут.

На современных системных платах функции контроллеров прерываний возлагаются на чипсет, который может иметь и более гибкие возможности управления, чем пара контроллеров 8259A. В операционном режиме всегда сохраняется программная совместимость с 8259A. Процедура инициализации контроллеров может и отличаться от традиционной, но ею занимается тест POST, который «знает» особенности системной платы. В симметричных мультипроцессорных системах аппаратные прерывания работают сложнее, поскольку их могут обслуживать различные процессоры. Для реализаций системы прерываний процессоры Pentium и выше имеют встроенный контроллер прерываний APIC (Advanced Programmable

Interruption Controller). Внутренние контроллеры процессоров связаны между собой по шине APIC, к которой подключена и «ответная часть» чипсета, преобразующая запросы аппаратных прерываний в сигналы протокола APIC. В операционном режиме такая связка также совместима с 8259 А.

### 12.3.1. Совместное использование прерываний

Линии запросов прерываний в компьютере, насыщенном дополнительными адаптерами, являются самым дефицитным ресурсом, поэтому возникает желание использовать эти линии совместно, то есть применять разделяемые прерывания между несколькими устройствами (shared interrupts). Обработчики прерываний (программы) от разных устройств, разделяющих одну линию запроса (и следовательно, общий вектор прерывания), должны быть выстроены в цепочку. В процессе обработки прерывания очередной обработчик в цепочке чтением известного ему регистра своего устройства должен определить, не это ли устройство вызвало прерывание. Если это, то обработчик должен выполнить необходимые действия и сбросить сигнал запроса прерывания от своего устройства, после чего передать управление следующему обработчику в цепочке; в противном случае он просто передает управление следующему обработчику.

Разделяемые прерывания для разнотипных устройств в общем случае работоспособными считать нельзя. Во-первых, у каждого устройства факт прерывания программно обнаруживается по-своему, и этот способ знает только драйвер этого устройства. Так что программно для совместного использования прерываний их обработчики должны уметь выстраиваться в цепочки, что на практике выполняется не всегда корректно. Во-вторых, возможны потери прерываний от устройств, требующих быстрой реакции. Это может происходить, если обработчик такого устройства окажется в конце цепочки, а предшествующие ему обработчики окажутся «нерасторопными» (не самым быстрым способом обнаружат, что прерывание — не их). Поведение системы в такой ситуации может меняться в зависимости от порядка загрузки драйверов. Для нескольких однотипных устройств (например, сетевых адаптеров на одном и том же кристалле), пользующихся одним драйвером, разделяемые прерывания работают вполне успешно.

Чтобы прерывания, одновременно возникающие от нескольких устройств, не терялись, контроллер прерываний должен быть *чувствительным к уровню*, а не к перепаду на входе запроса. В соответствии со схмотехникой логики TTL и КМОП активным уровнем должен быть низкий; выходной формирователь сигнала запросов у адаптеров — с открытым коллектором (TTL) или открытым стоком (КМОП); вход запроса у контроллера должен быть «подтянут» к высокому уровню резистором. Тогда непосредственное соединение этих выходов со входом контроллера («Монтажное И») даст требуемый результат в аппаратном плане, а в программном плане необходимо корректно выстроить обработчики в цепочку.

Поясним, почему надежное разделение прерываний при чувствительности к перепаду на линии запроса невозможно. Если устройство 1 выработает сигнал запроса после того, как его выработает (но еще не снимет) устройство 2, то контроллер обработает только один запрос. Цепочка программных обработчиков окажется ненадежной: если обработчик устройства 1 в этой цепочке будет проверять свое

устройство до возникновения прерывания, то прерывание будет потеряно. Поскольку прерывания по своей природе обычно асинхронны, работа этих устройств совместно с поддерживающими программами будет загадочно нестабильной.

Как уже говорилось, в шине ISA прерывание вырабатывается по положительному перепаду сигнала на линии запроса. Стандартный контроллер 8259A позволяет задавать чувствительность — уровень или перепад — только для всех входов одновременно, поэтому разделяемые прерывания на шине ISA неработоспособны. Тем не менее некоторые чипсеты, реализующие контроллеры прерываний, допускают индивидуальное управление чувствительностью каждого входа. Тогда при соответствующих возможностях CMOS Setup, адаптеров и их ПО разделяемые прерывания технически реализуемы.

Для шины PCI, казалось бы, проблема разделения прерываний решена — здесь активным уровнем запроса является низкий, так что, запрограммировав входы контроллера на чувствительность к уровню, создается аппаратная база совместного использования. Однако на практике разделяемые прерывания работают не всегда, и иногда приходится подбирать положение карт расширения в слотах PCI, при которых устройства не конфликтуют друг с другом по прерываниям. Виной конфликтов могут быть как сами карты расширения, так и их драйверы, неспособные выстраиваться в корректную цепочку.

Если карта PCI использует одну линию запроса прерываний, то этой линией по умолчанию является INTR A. Если все четыре карты PCI используют по одной линии запроса, то, как это видно на рис. 12.1, каждая линия занимает монополю. Однако если сложная карта нуждается в большем числе линий запроса, то ей придется разделять линии с соседними картами. На современных системных платах часто устанавливают более четырех слотов PCI, при этом, естественно, «угроза» совместного использования линий запросов «нависает» и над картами с одной линией запроса. Порт AGP в плане прерываний следует рассматривать наравне со слотами PCI.

Проявления конфликтов и ошибок назначения прерываний могут быть разнообразными. Сетевая карта при ошибке в прерываниях не сможет принимать кадры из сети (при этом она может их успешно посылать). У устройств хранения доступ к данным будет поразительно медленным (иногда можно минутами ожидать, например, появления информации о файлах и каталогах) или вообще невозможным. Звуковые карты будут молчать или «заикаться», на видеопроекторе изображение будет дергаться и так далее. Конфликты могут приводить и к внезапным перезагрузкам компьютера, например по приходу кадра из сети или сигналу от модема.

## 12.4. Прямой доступ к памяти — DMA

*Прямой доступ к памяти* (Direct Memory Access, DMA) позволяет выполнять пересылку данных между регистрами устройств и памятью, минуя центральный процессор. Для устройств, использующих DMA, различают два типа доступа.

- ◆ Пассивный доступ, он же Slave DMA, — устройство пользуется общим контроллером DMA, расположенным на системной плате.

- ◆ Активный доступ, он же Bus Master DMA, — устройство само является ведущим на своей шине и способно генерировать обращение к памяти (как правило, системной). Реализация активного DMA зависит от типа шины расширения, к которой подключается устройство (см. главу 6). Примером устройств с активным DMA являются контроллеры АТА, расположенные на современных системных платах.

Процессор при обмене по DMA занят только инициализацией контроллера, которая сводится к записи в его регистры нескольких байт, задающих начальный адрес и размер пересылаемого блока памяти, направление и режим обмена. В самом обмене данных занят только контроллер DMA, память, к которой он обращается, и связующие их шины. Во время операций DMA процессор может продолжать работу, если выбранный режим обмена не занимает всей пропускной способности шин, используемых процессором в данный момент (шины памяти, шины PCI, через которые подключается ISA в современных компьютерах). Контроллер DMA можно считать простейшим сопроцессором ввода-вывода, разгружающим центральный процессор от рутинных операций обмена.

Обмен по DMA не всегда дает выигрыш в скорости обмена, в ряде случаев быстрее работает программированный ввод-вывод (PIO). Однако PIO занимает процессор полностью, а во время DMA процессор может заниматься полезной работой. Поскольку для инициализации контроллера DMA требуется выполнение ряда инструкций ввода-вывода, передача коротких блоков по каналу DMA нецелесообразна. *Пассивный DMA* реализуется стандартизованным контроллером, который первоначально был ориентирован на шину ISA. Для интерфейса ПУ каждый канал DMA представляется парой сигналов: запрос обмена — DRQx и подтверждение обмена — DACKx\*. В PC/AT доступны 7 каналов DMA — четыре 8-битных (номера 0-3) и три 16-битных (5-7), — подключенные к первичному и вторичному контроллерам соответственно. Канал 4 используется для каскадирования (соединения контроллеров). В PC/XT были только три 8-битных канала, канал 0 использовался для регенерации памяти. Контроллеры DMA программно совместимы с системами i8237, применяемыми в первых моделях PC/XT и AT. Стандартные каналы и адреса регистров приведены в табл. 12.3.

Таблица 12.3. Стандартные каналы прямого доступа к памяти

Номер канала DMA*	0 <sup>1</sup>	1	2	3	4 <sup>2</sup>	5	6	7
Стандартное назначение	XT	MRFR -	FDD	HDD	Отсутствуют			
Разрядность, байт	дт	-	FDD	-	Каскад -			
Макс. размер блока		1 •			2 с четного адреса			
Граница блока		64Кбайта			128 Кбайт, четный			
Регистр страниц		Кратна 1000h			Кратна 2000h			
Адреса регистров:		8бит A16-A23			7бит A17-A23			
— страниц		087 083	081	082	08F	08B	089	087
— адреса		000 002	004	006	0C0	0C4	0C8	0CE
— счетчика		001 003	005	007	0C2	0C6	0CA	0CE

<sup>1</sup> Канал 0 в XT использовался для регенерации памяти (MRFR). <sup>2</sup> Канал 4 доступен только в PS/2 MCA.

16-битные каналы DMA 5-7 могут быть использованы интеллектуальными устройствами для *прямого управления шиной ISA* (bus mastering), при этом контроллер DMA фактически лишь играет роль арбитра шины.

Устройства, использующие стандартные каналы DMA, могут располагаться лишь в слотах ISA/EISA или на системной плате (контроллер НГМД, LPT-порт в режиме *ECP wjm Fast Centonics*, аудиокодек). Если эти устройства системной платы используют каналы DMA, то данные каналы становятся недоступными для абонентов шины ISA.

На время переходного периода, связанного с «изживанием» шины ISA, потребовалась возможность эмуляции каналов DMA для устройств шины PCI. Существует два механизма эмуляции каналов DMA: PC/PCI и DDMA. Механизм PC/PCI (см. п. 6.2.7) был разработан фирмой Intel для обеспечения возможности использования слотов ISA блокнотными ПК, подключаемыми к док-станции по шине PCI. Альтернативное решение — механизм DDMA (Distributed DMA — распределенный DMA) позволяет «расчлнить» стандартный контроллер и отдельные его каналы эмулировать средствами карт PCI. Оба этих механизма реализуемы только как часть моста между первичной шиной PCI и шиной ISA, поэтому их поддержка может обеспечиваться (или не обеспечиваться) только на системной плате и разрешаться в CMOS Setup.

### 12.4.1. Контроллер прямого доступа 8237A

Микросхема 8237A, применявшаяся в PC вплоть до первых моделей AT, представляет собой четырехканальный контроллер прямого доступа к памяти, допускающий каскадирование. Вторичный контроллер (8237#2) каскадно соединен с первичным, при этом теряется возможность использования одного канала вторичного контроллера. Контроллер 8237A имеет 16-разрядные регистры адреса и счетчики, что обеспечивает возможность программирования передачи блока данных размером до 64 Кбайт или слов. Назначение регистров контроллеров DMA, применительно к их адресам в пространстве ввода-вывода компьютера, приведено в табл. 12.4. Контроллер допускает довольно гибкое конфигурирование. Корректное оперативное управление отдельными каналами не затрагивает общих настроек. Общее конфигурирование контроллеров (запись в регистры 008 и ODO) выполняет BIOS при инициализации во время теста POST; в XT тогда же программируется и канал 0, применяемый для регенерации памяти. Для использования каналов устройствами шины ISA запись в регистры 008 и ODO не рекомендуется. Обмен с регистрами контроллера выполняется только однобайтными операциями ввода-вывода. Для загрузки 16-битных значений задействуется триггер младшего/старшего байта. По сбросу контроллера или записи любого байта по адресу 00Ch (0D8h для второго контроллера) этот триггер сбрасывается, и контроллер готовится к приему младшего байта. После приема этого байта триггер меняет состояние, и контроллер воспринимает старший байт, после которого триггер опять переключается.



Таблица 12.4. Регистры контроллера DMA 8237A 8237#1

8237#1	8237#2	R/W, Назначение регистров
000, 002, 004, 006	0C0, 0C4, 0C8, 0CC	W — регистры начального адреса для каналов 0–3 (8237#1) и 4–7 (8237#2) R — регистры текущего адреса тех же каналов
001, 003, 005, 007	0C2, 0C6, 0CA, 0CE	W — начальное значение счетчика передач для каналов 0–3 (8237#1) и 4–7 (8237#2) R — текущее значение счетчика передач тех же каналов
008	0D0	W, <i>Command Register</i> — конфигурактор контроллера. Бит 7: 1 — активный уровень (DACK# — высокий, 0 — низкий); бит 6: 1 — активный уровень (DRQ — низкий, 0 — высокий); бит 5: 1 — режим расширенной записи (должен быть 0); бит 4: 0 — фиксированный приоритет, 1 — циклический; бит 3: 1 — укороченный цикл обмена (должен быть 0); бит 2: 1 — запрет работы контроллера; бит 1: 1 — фиксация адреса 0 канала (должен быть 0); бит 0: 1 — передача память-память (в PC не используется, должен быть 0)
008	0D0	R, <i>Status Register</i> — состояние каналов. Биты 4–7: запросы каналов 0–3; биты 0–3: завершение цикла каналов 0–3
009	0D2	W, <i>Request Register</i> — регистр программных запросов. Биты 7–3 не используются; бит 2: 1 — установка, 0 — сброс бита запроса; биты 1–0: выбор канала (00 — 0; 01 — 1; 10 — 2; 11 — 3)
00A	0D4	W, <i>Single Mask Bit Register</i> — управление масками. Биты 7–3 не используются; бит 2: 1 — установка, 0 — сброс бита маски; биты 1–0: выбор канала (00 — 0; 01 — 1; 10 — 2; 11 — 3)
00B	0D6	W, <i>Mode Register</i> — режимы работы каналов. Биты 7–6: режим передачи (00 — по запросу, 01 — одиночный, 10 — блочный, 11 — каскадирование); бит 5: 0 — инкремент, 1 — декремент адреса; бит 4: 1 — разрешение автоматической реинициализации
00B	0D6	Биты 3–2: тип передачи (00 — холостой, проверка канала, 01 — запись в память, 10 — чтение памяти, 11 — недопустимо); биты 1–0: выбор канала (00 — 0; 01 — 1; 10 — 2; 11 — 3)
00C	0D8	W, <i>Clear Byte Pointer Flip/Flop</i> — сброс триггера младшего/старшего байта
00D	0DA	W, <i>Master Clear</i> — общий сброс 8237 (вывод любого байта в регистр вызывает сброс)
00E	0DC	W, <i>Clear Mask Register</i> — общий сброс масок всех каналов (вывод любого байта в регистр вызывает сброс)
00F	0DE	W, <i>All Mask Register Bits</i> — регистр масок всех каналов. Биты 0–3: маски каналов 0–3 (0 — канал разрешен, 1 — замаскирован); биты 4–7 не используются

Программирование контроллера для каждого канала определяет начальный адрес, направление его модификации (инкремент/декремент), количество пересылаемых байт (слов), режим работы канала. *Регистры адреса* контроллеров 8237 A — 16-разрядные, и для расширения разрядности адреса для каждого канала имеются специальные *регистры страниц* (DMA page register), внешние по отношению к контроллерам 8237A. В отличие от адресных регистров контроллера 8237A регистры страниц при выполнении циклов DMA не модифицируются — в них по команде процессора до начала обмена по каналу загружается требуемое значение. В PC/AT регистры страниц хранят биты A[23:16] для 8-битных каналов и A[23:17] — для 16-битных. В PC/XT регистры страниц хранили только 4 бита A[19:16]. Контроллер 16-битных каналов подключен к шине адреса со смещением на 1 бит, так что линией АО он не

управляет. При передаче по 16-битным каналам всегда  $A0=0$  (передачи слов по четным адресам). *Счетчики циклов* каналов — 16-разрядные, что позволяет передавать блоки до 64 Кбайт (для 8-битных каналов) или слов (для 16-битных каналов). При инициализации в счетчик загружается число, на единицу меньшее требуемого числа циклов, так что FFFF соответствует 65 534 передачам ( $2^{16}$ ). В последнем цикле передачи (когда счетчик отсчитает требуемое количество циклов) контроллер вырабатывает сигнал завершения ТС, общий для всех каналов. Этот сигнал устройство может использовать для выработки сигнала прерывания, а программа может определить, какой канал «отстрелялся», прочитав регистр состояния каналов в контроллере DMA. В зависимости от выбранного режима по окончании счета контроллер либо остановит работу данного канала, либо выполнит автоматическую реинициализацию (автозагрузку) — восстановит прежде записанные значения регистра адреса и счетчика циклов и будет снова готов к пересылке такого же блока данных.

#### ВНИМАНИЕ

При достижении регистром-счетчиком адреса значения FFFFh следующее его значение будет 0000h, а внешний регистр адреса страницы, естественно, останется неизменным. Таким образом, если блок начинается не с границы 64 Кбайт страницы памяти, возможно его «сворачивание» в кольцо. Но если для процессоров 80x86 в реальном режиме сегменты, также «сворачиваемые» в кольца, могут начинаться с адреса любого 16-байтного параграфа (границы кратны 10h), то при прямом доступе эти границы кратны 10000h. Этот эффект обязательно необходимо учитывать при программировании прямого доступа — блок, пересекающий данную границу, должен пересылаться за два сеанса циклов DMA, между которыми канал (включая и регистр страниц) должен быть реинициализирован. Эффект «сворачивания» сегментов в 16-битных каналах аналогичен, только для каналов. 5-7 эти «кольца» имеют размер 64 Кбайт слов и границы, кратные 20000h.

Разрядность передаваемых данных по каналу DMA должна соответствовать типу канала — 16-битный канал всегда пересылает данные словами, и расщепление их на одиночные байты невозможно.

Каждый канал может работать в одном из трех логических режимов.

- ◆ Режим одиночной передачи (single transfer mode) — получив подтверждение DACKx#, устройство сразу снимает сигнал запроса DRQx, а контроллер DMA организует один цикл передачи. Счетчик адреса в контроллере модифицируется, счетчик циклов декрементируется.
- ◆ Режим блочной передачи (block transfer mode) — получив подтверждение DACKx#, устройство сразу снимает сигнал запроса DRQx, а контроллер DMA организует последовательность циклов передачи до обнуления счетчика циклов. Если разрешена автозагрузка канала, то для пересылки следующего блока требуется повторная подача DRQx. На время передачи всего блока контроллер монополюбно захватывает шину, при этом не выполняется регенерация памяти.
- ◆ Режим передачи по запросу (demand transfer mode) — получив подтверждение DACKx#, устройство не снимает сигнал запроса DRQx до тех пор, пока у него есть потребность в передаче. При наличии этого сигнала контроллер DMA организует последовательность циклов передачи вплоть до обнуления счетчика циклов. Если сигнал запроса снят до обнуления счетчика, контроллер DMA отдает управление шине, а при последующем появлении этого запроса продолжит обмен с того места, на котором остановился.

**ВНИМАНИЕ** -----

Используя DMA в режимах, отличных от одиночного, следует соблюдать осторожность, чтобы длительность непрерывной передачи не превышала 15 мкс.

---

Стандартный контроллер DMA на шине ISA с частотой 8 МГц работает на половинной частоте и требует для одиночной передачи не менее пяти своих тактов. Длительность одиночного цикла составляет 1,125 мкс. В блочных передачах пропускная способность DMA достигает 1 Мбайт/с для 8-битных каналов и 2 Мбайт/с для 16-битных (время цикла составляет 1 мкс). На современных компьютерах контроллер DMA реализуется чипсетом системной платы; при сохранении программной совместимости с 8237A он может работать на шине гораздо быстрее. Количество тактов шины на один цикл может программироваться опциями BIOS Setup.

## 12.5. Процессоры x86

Все программы в IBM PC-совместимом компьютере исполняются центральным процессором, принадлежащим к семейству x86. Любое устройство для процессора представляет собой лишь набор регистров (ячеек), отображенных в пространство памяти и (или) ввода-вывода, и необязательно источник аппаратных прерываний. Современные процессоры x86, работающие в защищенном режиме, имеют довольно сложные механизмы виртуализации памяти, ввода-вывода и прерываний, из-за которых приходится различать физические и логические пространства (адреса памяти и ввода-вывода) и события (операции ввода-вывода, прерывания). *Физический адрес* ячейки памяти или порта ввода-вывода — это адрес, формируемый для обращения к данной ячейке на физических шинах компьютера (системной шине процессора, шине PCI, ISA). *Логический адрес* — это тот адрес, который формируется исполняемой программой (по замыслу программиста) для доступа к требуемой ячейке. *Физическая операция* ввода-вывода или обращения к памяти — это процесс (шинный цикл), во время которого генерируются электрические сигналы, обеспечивающие доступ к данной ячейке (порту). *Логическая операция* — это исполнение программной инструкции (команды) обращения к интересующей ячейке. Логическая операция не всегда порождает ожидаемую физическую операцию: при определенных условиях она может блокироваться средствами защиты процессора, вызывая даже принудительное завершение программы, или же эмулироваться, создавая иллюзию физического исполнения.

Безопасность в защищенном режиме базируется на 4-уровневой системе привилегий. В большинстве современных ОС ради упрощения и экономии процессорного времени используются только два крайних уровня — нулевой (supervisor), с неограниченными возможностями, и третий (user), с самыми жесткими ограничениями. Смена уровней привилегий при исполнении программы занимает много тактов процессора, но это вынужденная плата за реализацию защиты, без которой устойчивую ОС не построить. Более подробно механизмы защиты и виртуализации памяти, ввода-вывода и прерываний в процессорах x86 описаны в литературе [6, 7], здесь же изложены лишь некоторые прикладные аспекты их работы.

## 12.5.1. Возможности адресации памяти процессорами различных поколений

Сложность обращения к памяти в PC обусловлена свойствами процессоров x86 разных поколений и требованием обратной совместимости новых процессоров и компьютеров со старым ПО.

*Процессорам 8086/88* было доступно адресное пространство 1 Мбайт с диапазоном адресов 0-FFFFh, причем физический 20-битный адрес вычислялся с помощью двух 16-битных компонентов по формуле  $Addr = Seg \times 16 + Offset$ , где Seg — содержимое сегментного регистра (CS, DS, SS или ES), а Offset — исполнительный адрес, формируемый из одного или нескольких слагаемых в соответствии с выбранным режимом адресации. Эта сегментная модель адресации позволяет программам оперировать с непрерывными блоками памяти (сегментами) размером не более 64 Кбайт. Для манипуляций с памятью большего размера требовалось переключение сегментов с помощью специальных инструкций процессора, что усложняло программирование. Заметим, что при  $Seg = FFFFh$  и  $Offset = FFFFh$  данная формула дает адрес 1OFFEFh, но ввиду 20-битного ограничения на шину адреса эта комбинация в физической памяти указывает на OFFEFh. Таким образом, адресное пространство как бы сворачивается в кольцо с небольшим «нахлестом».

*В процессоре 80286* шина физического адреса была расширена до 24 бит, и введен новый режим работы — *защищенный* (Protected Mode), в котором программа может обращаться к 16-мегабайтному пространству физической памяти через логическое пространство виртуальной памяти. Здесь виртуальная память строилась на основе той же сегментной модели памяти с 16-разрядными регистрами. Физический адрес формировался суммированием 16-разрядного исполнительного адреса (смещения внутри сегмента) с 24-разрядным базовым адресом сегмента.

Кроме защищенного режима, в процессоре 80286 имеется и реальный режим, в котором процессор ведет себя почти так же, как и 8086 (но более быстрый). Здесь физический адрес вычисляется так же, как и в 8086/88, но из-за ошибки разработчиков та самая единица в бите A20, которая отбрасывалась в процессорах 8086/88, теперь попадает на шину адреса, и в результате максимально доступный физический адрес в реальном режиме достиг 1 OFFEFh. Для обеспечения полной совместимости с процессором 8086/88 в схему PC ввели вентиль линии A20 шины адреса — *GateA20*, который либо пропускает сигнал от процессора, либо принудительно обнуляет линию A20 системной шины адреса. Этот вентиль должен быть открыт при работе в защищенном режиме, а также когда в реальном режиме нужны дополнительные (64 К-16) байт памяти. Вентиль управляется через контроллер клавиатуры (см. п. 8.1.2) или иным специфическим способом.

*В 32-разрядных процессорах*, начиная с 80386, сохранена та же идея обращения к памяти с участием сегментных регистров (16-разрядных), но регистры процессора, участвующие в формировании адреса, позволяют адресовать уже  $2^{32} = 4$  Гбайт памяти в каждом сегменте. *Базовый адрес сегмента* берется из специальных структур данных — *дескрипторов сегментов*. Кроме базового адреса в дескрипторе описывается его лимит (длина), назначение (код или данные), возможность записи

и чтения, а также уровень привилегий программы, позволяющий обращаться к данному сегменту. Deskрипторы предварительно программно формируются в памяти, где их наборы хранятся в виде таблиц deskрипторов. Процессор имеет средства защиты памяти, контролирующие использование сегментов. Программа может обращаться лишь к тем сегментам памяти, описание которых имеется в доступных deskрипторах. Виртуальное адресное пространство, доступное программе, имеет объем до  $(16 \text{ К} - 2)$  сегментов (число возможных deskрипторов), каждый из которых может иметь размер до 4 Гбайт. Deskриптор выбирается с помощью *селектора*, загружаемого в сегментный регистр (CS, DS, SS, E5, F5 или GS). Однако это виртуальное адресное пространство отображается блоком сегментации в логическое адресное пространство с опять-таки 32-разрядным линейным адресом, то есть объемом 4 Гбайт. По замыслу разработчиков процессора, это отображение с подкачкой требуемых сегментов с диска и выгрузкой неиспользуемых должно выполняться диспетчером виртуальной памяти операционной системы. Практически такая виртуализация применялась на процессорах 80286 (с 16-разрядными регистрами), поскольку иных механизмов не существовало.

Для виртуализации памяти (и защиты) в 32-разрядных процессорах применяется иной механизм, основанный на блоке страничной переадресации — принципиальной новинке 32-разрядных процессоров x86. В его задачу входит отображение 32-разрядного линейного адреса (продукта блока сегментации) на 32- или 36-разрядный физический адрес, формируемый на системной шине процессора при его обращениях к памяти. В отличие от блока сегментации, оперирующего блоками разного размера (сегментами), блок страничной переадресации оперирует страницами одинакового размера. Переадресация выполняется на основе *таблиц страниц*, где для каждой страницы логической памяти имеется свой описатель. В этом описателе имеется признак присутствия страницы в физической памяти, и для присутствующих страниц указывается базовый адрес физического отображения. Кроме того, имеются биты, управляющие доступом к странице по чтению и записи с различных уровней привилегий, возможностью ее кэширования, и некоторые служебные биты. При обращении программы к отсутствующей странице процессор вырабатывает исключение, обработчик которого занимается *подкачкой* нужной страницы из внешней памяти (с диска) в ОЗУ. Этот обработчик и реализует *виртуальную память с подкачкой страниц по запросу* (Demand-Paged Virtual Memory), которая в настоящее время обычно и подразумевается под виртуальной памятью. При недостатке свободного места в физической памяти обработчик выполняет и замещение страниц, по его мнению, наименее нужных, выгружая их на диск. Создав несколько наборов описателей страниц, можно получить несколько виртуальных адресных пространств, каждое из которых имеет размер до 4 Гбайт, причем страницы разных пространств могут быть полностью изолированы друг от друга, а могут и частично пересекаться. В многозадачной ОС каждая задача (виртуальная машина) имеет собственное (как ей представляется) адресное пространство.

Первоначально блок страничной переадресации работал со страницами размером 4 Кбайт. В дополнение к этому базовому механизму в процессор Pentium ввели возможность работы и со страницами размером 4 Мбайт (режим *PSE*). В ряде

процессоров P6 разрядность физического адреса увеличена до 36 бит, и все процессоры P6 имеют возможность включения режима переадресации *PAE*, позволяющего отображать страницы размером 4 Кбайт и 2 Мбайт с расширением физического адреса. С процессорами Pentium III появился режим преобразования *PSE-36*, в котором блок оперирует 4-Мбайтными страницами в 36-битном физическом пространстве и сохраняется возможность работы со стандартными 4-Кбайтными страницами базового режима. Это позволяет довольно эффективно управляться с современными объемами физической памяти компьютера.

*В стандартном реальном режиме 32-разрядные процессоры* работают с памятью так же, как и 80286, с возможностью адресации в диапазоне 0-10FFEFh, причем вентиль Gate A20 ввели уже в сам процессор. Физический адрес вычисляется с участием сегментных регистров, размер непрерывного сегмента — 64 Кбайт. По умолчанию в реальном режиме адреса формируются с использованием только младших 16 бит 32-разрядных регистров, правда, для каждой инструкции можно с помощью префиксов изменить разрядность адресных компонентов на 32 бита. Однако и при этом невозможно пересечь границу 64-Кбайтного сегмента — сработает исключение защиты.

В стандартном реальном режиме блок страничной переадресации не работает, и физический адрес совпадает с линейным. С помощью временного переключения в защищенный режим можно настроить таблицы страниц, разрешить преобразование и далее в реальном режиме задействовать страничное преобразование. Этот трюк используется менеджерами памяти типа EMM386 для работы со свободными блоками UMA.

Есть и еще один режим, неофициальный, но тоже работающий на всех 32-разрядных процессорах x86, — «нереальный» (unreal), он же «большой реальный» (big real). Он позволяет процессору в реальном режиме обращаться к данным, расположенным в любом месте 4-Гбайтного пространства линейных (и физических) адресов. Этот режим базируется на логике блока сегментации, которая при вычислении линейного адреса во время обращений к памяти пользуется скрытыми программно-недоступными регистрами дескрипторов сегментов. Из этих регистров берется базовый адрес, из них же берется и лимит, который используется схемой защиты. В этих регистрах кэшируются дескрипторы сегментов, загружаемые из памяти во время исполнения инструкций, переопределяющих значения сегментных регистров (CS, DS, SS, ES, FS и GS) в защищенном режиме. По аппаратному сбросу в эти скрытые регистры заносятся «неинтересные» параметры стандартного реального режима, с лимитом 64 Кбайт. В реальном режиме при переопределении сегментных регистров значение базового адреса берется как 16-кратное значение, загружаемое в соответствующий сегментный регистр, а лимит устанавливается в 64 Кбайт. Тем не менее, если в защищенном режиме в сегментный регистр загрузить селектор дескриптора, в котором описан сегмент размером 4 Гбайт с нулевым базовым адресом и возможностью полного доступа на любом уровне привилегий, переключиться в реальный режим и не трогать этот сегментный регистр, то далее процессор будет иметь доступ ко всему этому сегменту в данной модификации реального режима. Однако такая «благодать» распространяется

только лишь на доступ к *данным* через сегментные регистры FS и GS, которые используются в инструкциях обращений к памяти, снабженных префиксами замены сегмента. Эти сегментные регистры появились только с 32-разрядными процессорами, и никакие традиционные сервисы BIOS (и DOS) их не затрагивают. Остальные сегментные регистры настолько часто используются, что «время жизни» описания большого сегмента в их кэширующих регистрах будет слишком коротким. Программный код, увы, выполняется только из сегмента, которым командует CS, поэтому для него остается лишь первый мегабайт с 64-Кбайтными сегментами. Так что большие программные модули приходится подгружать в эту область по мере надобности, но это можно выполнять довольно быстро пересылками данных из любого места «большого сегмента». Большой реальный режим широко используется менеджерами памяти, а также игровыми DOS-программами, всецело захватывающими ресурсы компьютера.

Итак, самые широкие возможности адресации имеются в защищенном 32-разрядном режиме, наиболее естественном для современных процессоров. В этом режиме может использоваться как плоская, так и сегментная модели памяти. Под *плоской* (flat) понимается модель, в которой все сегментные регистры указывают на один и тот же сегмент памяти (как правило, начинающийся с нулевого адреса), и его лимит может достигать 4 Гбайт, что позволяет адресовать этот немалый (даже по нынешним меркам) объем памяти без манипуляций сегментными регистрами. Однако при этом теряются все возможности виртуализации памяти на основе сегментов, а также отсутствует сегментная защита. В *сегментной модели* памяти сегментные регистры кода, стека и данных настраиваются на разные, возможно и не пересекающиеся сегменты. Здесь имеются все возможности сегментной защиты и сегментной виртуализации памяти. Поскольку современным приложениям пока достаточно 4 Гбайт памяти (надолго ли?), сегментную модель ради упрощения диспетчера памяти стараются не использовать. Защита памяти имеется и на уровне страниц, правда, не такая развитая и надежная, как сегментная.

## 12.5.2. Проблемы страничной переадресации

В реальном режиме (при отключенной страничной переадресации) логический адрес, формируемый прикладной программой, совпадает с физическим адресом, фигурирующим на шинах расширения. Тут все просто, правда, в стандартном (а не большом) реальном режиме доступен только первый мегабайт адресов (только ко устройства в области UMA).

В защищенном режиме в принципе доступно все физическое адресное пространство, но появляются проблемы, связанные с отображением логических адресов на физические. Отображением (поддержкой таблиц переадресации) ведает ОС, приложения могут только узнать карту отображений (получить список физических адресов страниц для какой-то области своей виртуальной памяти). Какие-то области могут в данный момент и не присутствовать в ОЗУ (они могут быть выгружены на диск). У драйверов устройств возможностей больше — они могут за просить блок памяти с последовательными физическими страницами и потребовать фиксации определенных страниц (запретить их выгрузку из ОЗУ).

При организации прямого доступа к памяти, как по стандартным каналам DMA, так и используя ведущие устройства шин ISA и PCI, возникает проблема пересечения границ страниц. Если приложение хочет выполнить обмен по DMA с областью доступной ей памяти непосредственно, то оно должно запросить у ОС физический адрес, которому соответствует логический адрес предполагаемого буфера обмена. Именно этот физический адрес должен задаваться устройству, выполняющему DMA, при инициализации сеанса обмена (указании начального адреса, длины блока и запуске канала). В каждом сеансе обмена не должна пересекаться граница страницы, которой оперирует блок страничной переадресации, поскольку следующая логическая страница может иметь физическое отображение в произвольном (относительно предыдущей страницы) месте. Чаще всего ОС оперирует страницами по 4 Кбайт, при этом пересылка больших блоков данных ведется «короткими перебежками», между которыми процессор должен выполнять повторную инициализацию DMA. Эта проблема решается усложнением контроллеров DMA — применением «разбросанной записи» в память (scatter write) и «собирающего чтения» памяти (gather read). Контроллеру DMA задается список описателей блоков (начальный адрес и длина). Отработав очередной блок памяти, контроллер переходит к следующему, и так до конца списка. Такие возможности имеет, например, стандартный контроллер PCI IDE (см. п. 9.2.1). Стандартный контроллер DMA имеет и другую «страничную проблему», связанную с реализацией регистров страниц (см. п. 12.4).

Проблема пересечения границ может решаться и иначе, без усложнения контроллера DMA. Для этого в памяти резервируется буфер значительного размера, отображенный на непрерывную область физической памяти, и обмен данными физическое устройство выполняет только с этим буфером. Однако такой буфер рядовое приложение создать не может; он может быть организован лишь драйвером устройства. Приложения могут только получать указатели на этот буфер и обмениваться с ним данными. Таким образом, по пути от приложения к устройству появляется дополнительная «перевалочная база» (буфер драйвера) и дополнительная пересылка данных, что приводит к дополнительным затратам времени.

### 12.5.3. Инструкции ввода-вывода

Для обращения программы к пространству ввода-вывода предназначены всего четыре инструкции процессора: IN (ввод из порта в регистр процессора), OUT (вывод в порт из регистра процессора), INS (ввод из порта в элемент строки памяти) и OUTS (вывод элемента из строки памяти в порт). Последние две инструкции, появившиеся с процессором 80286, могут использоваться с префиксом повтора REP, что обеспечивает быструю пересылку блоков данных между портом и памятью. Обмен данными с портами, при котором применяют строковые инструкции ввода-вывода, получил название *PIO* (Programmed Input/Output — программированный ввод-вывод). Скорость такого обмена превышает скорость стандартного канала прямого доступа (DMA), правда, DMA в отличие от PIO почти не занимает процессорного времени.



Разрядность слова, передаваемого за одну инструкцию ввода-вывода, может составлять 8, 16 или 32 бита. В зависимости от «выровненности» адреса по границе слова и разрядности данных используемой шины это слово может передаваться за один или несколько циклов шины с указанием соответствующего нарастающего адреса в каждом цикле обращения к памяти. Инструкции ввода-вывода порождают шинные циклы обмена, в которых вырабатываются сигналы чтения порта/записи в порт. На шине ISA это сигналы **IORD\*** и **IOWR\*** соответственно; они и отличают пространство ввода-вывода от пространства памяти, где соответствующие операции чтения и записи вырабатывают сигналы **MEMRD\*** и **MEMWR\***. На шине PCI разделение памяти и пространства ввода-вывода происходит иначе — здесь тип операции кодируется четырехбайтной командой, в зависимости от типа инструкции, выполняемой процессором.

Во избежание недоразумений и для экономии шинных циклов рекомендуется выравнивать адреса 16-битных портов по границе слова, а 32-битных — по границе двойного слова. Обращения по выровненным адресам выполняется за один цикл системной шины. Обращение по невыровненным адресам выполняется за несколько циклов, причем однозначная последовательность адресов обращений (которая зависит от модели процессора) не гарантируется. Так, например, одна инструкция вывода слова по нечетному адресу приведет к генерации двух смежных шинных циклов записи. При программировании обращений следует учитывать специфику устройств ввода-вывода. Если, например, устройство допускает только 16-разрядные обращения, то старший байт его регистров будет доступен лишь при вводе-выводе слова по четному адресу.

Некоторую сумятицу в стройную систему адресации вводят регистры АТ А. Здесь регистр 1FO (1 канал) является 16-битным регистром данных, но в то же время есть и совершенно независимый от него 8-битный регистр 1F1. В Serial ATA эта тема развита — здесь имеются еще четыре 16-битных регистра с адресами (относительно базового адреса блока командных регистров) 2, 3, 4 и 5, которые раньше были 8-битными.

В реальном режиме процессора программе доступно все пространство адресов ввода-вывода. В защищенном режиме 32-разрядных процессоров (частным случаем которого является и виртуальный режим V86) имеется возможность программного ограничения доступного пространства ввода-вывода, определяя его максимальный размер (начиная с нулевого адреса и в пределах 64 К), а внутри разрешенной области доступ может быть разрешен или запрещен для каждого конкретного адреса. Размер области и карта разрешенных портов (IO Permission Bitmap) задается операционной системой в дескрипторе сегмента состояния задачи (TSS). При обращении по неразрешенному адресу вырабатывается исключение процессора, а поведение его обработчика определяется операционной системой. Возможно снятие задачи-нарушителя (знаменитое сообщение «приложение... выполнило недопустимую операцию и будет закрыто»). Возможен и другой вариант, когда по обращению к порту монитор операционной системы выполняет некоторые действия, создавая для программы иллюзию реальной операции ввода-вывода. Таким образом виртуальная машина по операциям ввода-вывода может общаться с виртуаль-

ными устройствами. Заметим, что ОС Windows 9x не особо заботится о виртуализации и защите ввода-вывода; здесь, например, из DOS-окна можно обращаться к любым портам, даже к портам устройств, занятых операционной системой.

## 12.5.4. Прерывания

В процессорах x86 используются аппаратные прерывания, программные прерывания и исключения. *Аппаратные прерывания* были описаны выше; кроме того, к ним относится и специфичное (и неиспользуемое прикладными программами) прерывание SMI для входа в режим системного управления (SMM). *Программные прерывания* по сути прерываниями и не являются — это лишь короткая форма дальнего вызова ограниченного количества процедур, выполняемая инструкцией `Int N` ( $N=0-255$ ). Программные прерывания, в частности, используются для вызовов сервисов BIOS и DOS. *Исключения* генерируются процессором и сопроцессором, когда при исполнении инструкций возникают особые условия (например, деление на ноль или срабатывание защиты). Исключения занимают векторы прерываний 0-31, которые частично пересекаются с векторами аппаратных прерываний ведущего контроллера и NMI, а также с векторами сервисов BIOS. В процессорах 8086/88 исключения назывались внутренними прерываниями, их было совсем мало. По мере «взросления» процессоров добавлялись новые исключения; исключениями особо богаты современные процессоры при работе в защищенном режиме. На исключениях строится защита и виртуальная память в многозадачных ОС защищенного режима.

В реальном режиме прерывания работают довольно просто, и их обработчики могут находиться в любом месте физически адресуемой памяти (ОЗУ или ПЗУ). В таблице прерываний, начинающейся с нулевого адреса, каждый вектор прерываний представляется дальним указателем на процедуру обработки (16-байтное смещение и сегмент). Внедрение собственных обработчиков прерываний представляет собой несложную задачу, если прерывание используется монополично одним устройством и соответствующим ему единственным модулем ПО. В реальном режиме любая программа может управлять флагом разрешения аппаратных прерываний; некорректное управление флагом может приводить к различным неприятностям — от сбоя системного времени до «зависания» компьютера.

В защищенном режиме прерывания работают гораздо сложнее. Таблица прерываний здесь содержит 8-байтные дескрипторы прерываний. Их обработчики должны быть подключены к ядру ОС, постоянно присутствующему в физической памяти. Иначе возможна ситуация, когда, например, аппаратное прерывание вызовет обработчик, выгруженный в данный момент на диск менеджером виртуальной памяти. Обработка такого прерывания будет чрезвычайно долгой (потребуется подкачка страницы). Позволять любой программе управлять флагом разрешения прерываний для многозадачных ОС нельзя из соображений общей устойчивости системы. Сам процесс обработки прерываний (и исключений) в защищенном режиме существенно отличается от прерываний реального режима, и в современных процессорах и ОС имеются средства виртуализации прерываний, о чем подроб-

нее можно прочитать в [6,7]. Заметим, что в ОС Windows 9x каждое окно MS-DOS представляет собой отдельную виртуальную машину с собственной таблицей прерываний «реального» вида, и работа с прерываниями в ней практически не отличается от работы в «чистой» MS-DOS. Для установки обработчиков прерываний, требуемых программам защищенного режима, используются вызовы специальных сервисов ОС, и обработчик оформляется особым образом в соответствии с соглашениями этой ОС (не так, как для MS-DOS).

## 12.6. Аппаратные средства измерения времени

В IBM PC/AT имеются аппаратные средства для измерения времени. *Трехканаль-ный счетчик-таймер*, программно совместимый с 18254 (в XT — 8253), выполняет следующие функции:

- ◆ канал 0: — генерация аппаратных прерываний (IRQ0) каждые 54,936 мс (частота 18,206 Гц), вызывающих инкремент *системного таймера* (счетчика в ячейке 40:006E BIOS Data Area);
- ◆ канал 1 — генерация запросов на регенерацию памяти;
- ◆ канал 2 — генерация звуковых сигналов или измерение времени.

Внутренние счетчики микросхемы имеют разрядность 16 бит, но общение с ними возможно только 8-битными операциями. При этом можно задавать значение только младшего байта счетчика (LSB), только старшего (MSB) или обоих (LSB/MSB), причем сначала передается младший, а потом старший байт. Программирование микросхемы осуществляется записью байт в управляющий регистр по отдельности для каждого канала. Назначение регистров счетчиков-таймеров приведено в табл. 12.5. Входная частота для всех каналов 1,19318 МГц. Штатно все каналы работают в режиме генерации импульсов. Счет для каналов 0 и 1 разрешен постоянно. В канале 2 используется управляющий вход GATE, разрешающий счет, который управляется битом 0 (T2G, R/W) системного порта AT (061h). Выходной сигнал канала 2 может быть программно считан (T20, бит 5 того же порта). При использовании канала 2 для измерения времени необходимо отключить формирование звука (обнулив бит SPK, R/W, бит 1 порта 061h).

Таблица 12.5. Регистры счетчиков-таймеров Порт,

R/W	Назначение
040 RW	Счетчик 0 — системные часы. Режим 011, LSB/MSB, Binary, константа счетчика равна 0 (соответствует коэффициенту деления 65 536)
041 RW	Счетчик 1 — регенерация памяти. Режим 010, LSB, Binary, константа счетчика равна 12h (18)
042 RW	Счетчик 2 — генератор звука, измерение времени. Вход GATE от бита 0 порта B 8255 (061). Режим 011, LSB/MSB, Binary, значение счетчика определяет высоту тона

**Порт, R/W Назначение**

043 W	Управляющий регистр. Биты 7,6 — выбор счетчика 0,1,2. Биты 5,4 — режим, обращения: 00 — защелка текущего значения; 01 — LSB — только младший байт; 10 — MSB — только старший байт; 11 — LSB/MSB — сначала младший, затем старший байты. Биты 3-1 — режим счетчика: 000 — прерывание по счетчику; 001 — ждущий мультивибратор (одновибратор, у 8254 несколько отличается от 8253); x10 — генератор коротких импульсов заданной частоты; x11 — генератор меандра; 100 — счетчик событий с разрешением; 101 — счетчиксобытий с перезапуском. Бит 0 — 0=Bin (двоичный счет), 1 =BCD — (двоично-десятичный счет)
-------	---

Часы-колендарь CMOS RTC являются частью комбинированной микросхемы MC146818 с батарейным питанием, используемой в IBM PC/AT для хранения ряда аппаратных настроек. Часы синхронизируются от собственного генератора (32,768 кГц), они содержат:

- ◆ часы-календарь (год, месяц, число, час, минута, секунда);
- ◆ будильник, подающий сигнал в назначенный час, минуту и секунду;
- ◆ генератор меандра, позволяющий формировать запросы прерываний с заданной частотой (как правило, 1024 Гц).

CMOS RTC является источником аппаратных прерываний (IRQ8). Прерывания могут возникать от будильника, генератора меандра и после смены времени в часах. Отдельные источники прерывания идентифицируются чтением ячейки OSB и разрешаются записью в ячейку OBh.

Доступ к ячейкам CMOS RTC осуществляется через порты ввода-вывода 070h (индекс ячейки) и 071h (данные). Заметим, что бит 7 порта 70h используется и для блокировки NMI (см. п. 12.4), так что диапазон адресов памяти CMOS ограничен пределами 0-7Fh. Поскольку эта память имеет быстродействие порядка единиц микросекунд, между командами записи адреса и чтения-записи данных необходима программная задержка. Во время изменения состояния часов данные, считываемые из ячеек 0-9, могут оказаться некорректными. Признаком этой ситуации является единичное значение бита 7 ячейки OAh. Для определения момента окончания смены состояния часов можно пользоваться и разрешением соответствующего источника прерывания. Назначение ячеек CMOS RTC, относящихся к таймерной части, приведено в табл. 12.6 (полное определение ячеек см. в [1]).

Таблица 12.6. Назначение ячеек таймерной части CMOS RTC

Индекс	Назначение
00h-09h, 32h (37 в PS/2)	Ячейки RTC в BCD-формате: 00 — секунды; 01 — секунды будильника; 02 — минуты; 03 — минуты будильника; 04 — часы; 05 — часы будильника; 06 — день недели; 07 — день месяца; 08 — месяц; 09 — год (2 младшие цифры); 32h — век-1 (2 старшие цифры года); 37h — век-1 (2 старшие цифры года) в PS/2

продолжение А

Таблица 12.6 (продолжение)

Индекс	Назначение
0Ah	RTC Status Register A (регистр статуса A): бит 7 — обновление времени (0 — готов к чтению); биты [6:4] — делитель частоты (для кварца на 32,768 кГц — 010); биты [3:0] — 0110 — выходная частота меандра 1024 Гц
0Bh	RTC Status Register B (регистр статуса B): бит 7 — остановка часов (0 — нормальный ход); бит 6 — разрешение периодических прерываний (0 — запрещено); бит 5 — разрешение прерывания от будильника (0 — запрещено); бит 4 — разрешение прерывания по окончании смены времени (0 — запрещено); бит 3 (см. также регистр 0Ah) — разрешение выходного меандра (0 — запрещено); бит 2 — формат BCD/BIN (0 — BCD); бит 1 — 12/24-часовой режим (1 — 24-часовой); бит 0 — зимнее/летнее время (0 — переключение запрещено)
0Ch	RTC Status Register C (регистр статуса C): чтение флагов идентификаторов прерывания: бит 7 — IRQF (общий запрос прерывания); бит 6 — PF (периодические прерывания); бит 5 — AF (прерывание от будильника); бит 4 — UF (прерывание по окончании смены времени); биты [3:0] — зарезервированы
0Dh	RTC Status Register D (регистр статуса D): бит 7 — питание (1 — норма, 0 — разрядбатарен); биты [6:0] — зарезервированы

Аппаратные таймеры имеют поддержку функциями BIOS (подробнее см. [1,8,9]). Сервисы BIOS Int 1 Ah позволяют считывать и модифицировать значения системного таймера (ячейки 40:006Eh в BIOS Data Area), а также даты, времени и будильника CMOS RTC.

Функции BIOS Int 15 h позволяют с помощью CMOS RTC вводить задержку или запускать таймер установки флага (через заданное время установить бит 7 указанной ячейки памяти). Время задается в микросекундах, но минимальная выдержка зависит от производительности ПК (достижимы единицы миллисекунд), максимальная выдержка — около 70 часов.

Начиная с процессоров Pentium, появилась возможность измерения времени с точностью до такта ядра процессора. Для этого процессоры имеют внутренний 64-битный счетчик TSC (Time Stamp Counter), обнуляющийся по аппаратному сбросу (сигналом RESET\*). Разрядность позволяет считать без переполнения в течение нескольких столетий. Для доступа к счетчику имеется специальная инструкция RDTSC, правда, установкой флага TSD в управляющем регистре CR4 (процессора) ОС может сделать ее привилегированной (доступной только на нулевом уровне привилегий). В этом случае приложении, исполняемое на уровне 3, может аварийно завершаться по отказу исполнения инструкции. ОС может и позволить обращение к этому регистру, но «подсовывая» программе удобное ей значение времени. Заметим, что из-за внутреннего умножения частоты в процессоре результат чтения счетчика может отставать от реального времени на число, достигающее коэффициента умножения частоты. Правда, такая точность никому и не нужна (она потеряется в измеряющих программах).

## 12.7. Способы запуска программ

Традиционным способом запуска программ является загрузка их с какого-либо устройства хранения (диска) в оперативную память и исполнение в ОЗУ. До того как пользователь получает возможность такого запуска, требуется множество предварительных действий. По включении питания (и аппаратному сбросу) процессор начинает исполнять процедуру POST (начальный запуск и самотестирование) из ROM BIOS, причем большая часть кода исполняется прямо в ПЗУ. POST инициализирует стандартное оборудование ПК (о котором «знает» ROM BIOS системной платы), а также обнаруживает модули расширения ROM BIOS и запускает их процедуры инициализации. Далее POST определяет загрузочное устройство (обычно диск), ищет на нем загрузочную запись (сектор), загружает этот сектор в фиксированную область ОЗУ и передает управление на его начальный адрес. С этого момента, как правило, начинается процесс загрузки операционной системы с того же носителя: сначала базовой системы, а затем и всех необходимых дополнительных компонентов в виде драйверов и автоматически загружаемых программ. Только после этого пользователь может запускать требуемые программы с любого доступного устройства хранения (в том числе и через сеть). Программы могут загружаться и автоматически, без участия пользователя, по предварительно составленному и сохраненному сценарию (файлы config.sys, autoexec.bat и т. п. средства).

Для ряда специальных применений ПК приходится нарушать эти традиции. Для сравнительно простых систем можно отказаться от использования операционных систем. Программу функционирования компьютера можно «защитить» в ПЗУ, оформив в виде модуля расширения BIOS, и эта программа получит управление от POST. Можно и не связываться с ПЗУ, а загружать программу с устройства хранения простым загрузчиком, первая ступень которого должна совпадать со стандартным начальным загрузчиком. Однако не стоит отказываться от операционной системы без веских причин, поскольку она обеспечивает не только удобное операционное окружение, но и средства разработки и отладки программ. Операционную систему и необходимые программы можно загружать не только с привычных дисков (гибких, жестких, оптических), но и с компактных твердотельных носителей (см. п. 9.3). Эти носители могут подключаться к обычным интерфейсам устройств хранения.

Интересный вариант «твердотельного диска» — DiskOnChip — для микрокомпьютеров и микроконтроллеров, не имеющих стандартных интерфейсов устройств хранения, предлагает фирма M-Systems. Это микросхема, имеющая интерфейс 8/16-битной статической памяти, легко подключаемый к шине ISA (или локальной шине). Модель Millenium Plus объемом 32 Мбайт содержит массив флэш-памяти архитектуры NAND, модуль статической памяти SRAM (1 Кбайт), интерфейсные схемы, логику защиты записи и чтения и схемы обнаружения и исправления ошибок. Микросхема отображается на 8-Кбайтную страницу пространства памяти компьютера в области CS000-EFFFFh. По сигналу аппаратного сброса начальный блок из флэш-памяти выгружается в SRAM; если обнаруживается ошибка, то

берется следующий (резервный) блок. Этот блок содержит процедуру инициализации «диска», которая обнаруживается тестом POST как модуль расширения BIOS. Процедура загружает из флэш-массива в системное ОЗУ драйвер своего «электронного диска» (блочного устройства), которое становится первым или последним логическим жестким диском (по выбору при конфигурировании). Далее к этому «диску» можно обращаться обычным способом (через Int 13h), с него же может и загружаться ОС. Интерфейс допускает каскадирование — объединение в единый диск до 4 микросхем, увеличивая его объем до 128 Мбайт, при этом все микросхемы отображаются через общее окно памяти (используют общий сигнал выборки). Встроенное ПО обеспечивает полную эмуляцию диска с прозрачным исправлением ошибок и переназначением дефектных секторов. Микросхема поддерживает длительную скорость записи 750 Кбайт/с, считывания — 2,4 Мбайт/с. Пиковая скорость считывания/записи достигает 20 Мбайт/с. В устройстве имеется уникальный идентификационный номер, область для однократного программирования (OTP), возможность защиты от записи отдельных зон и возможность ограничения доступа по паролю (нечитаемому).

## 12.8. Сервисы и прерывания BIOS

Системная BIOS предоставляет ряд сервисов низкого уровня, в основном предназначенных для обслуживания ввода-вывода и имеющих отношения к стандартным аппаратным интерфейсам. Традиционные сервисы BIOS обычно вызываются в реальном режиме или V86 посредством инструкций программных прерываний (Int xx). Большинство сервисов может быть вызвано и через фактически стандартизованные точки входа (адреса в области ROM BIOS) дальними вызовами процедур (CALL far) с предварительным помещением в стек регистра флагов (сервисы построены как обработчики прерываний). Все традиционные сервисы BIOS работают в 16-разрядном режиме процессора, и ими можно пользоваться в реальном режиме, V86 и малопривлекательном 16-разрядном защищенном режиме.

Для процессоров 386+ оптимальным по эффективности является 32-разрядный защищенный режим. Для того чтобы из этого режима можно было пользоваться сервисами BIOS (правда, не всеми) без промежуточных переключений, по инициативе фирмы Phoenix ввели 32-разрядные вызовы BIOS32. Адрес точки входа BIOS32 заранее не известен, но известен способ его нахождения: в диапазоне адресов памяти 0E0000-0FFFFh на границе параграфов (младшие 4 бита адреса нулевые) ищется строка-сигнатура " \_ 32 \_ " (число 325F5F33H) заголовка, за которой следует физический адрес точки входа. Сами сервисы вызываются дальними вызовами точки входа в сервис. Номер, параметры вызываемых функций и результаты передаются на регистрах процессора.

Прерывания, обслуживаемые системной BIOS, перечислены ниже. Кроме них несколько векторов используются как указатели на различные структуры данных.

*Внутренние прерывания:*

- ◆ Int 00h — деление на 0;
- ◆ Int 01h — пошаговый режим;
- ◆ Int 03h — точка останова;
- ◆ Int 04h — переполнение;
- ◆ Int 06h — недопустимая команда 286+;
- ◆ Int 07h — вызов отсутствующего NPU.

*Аппаратные прерывания:*

- ◆ Int 02h — немаскируемое прерывание;
- ◆ Int 08h - таймер 8253/8254;
- ◆ Int 09h — клавиатура;
- ◆ Int 0Ah - IRQ2/9;
- ◆ Int 0Bh- IRQ3;
- ◆ Int 0Ch-IRQ4;
- ◆ Int 0Dh — IRQ5;
- ◆ Int 0Eh — IRQ6 — контроллер гибких дисков;
- ◆ Int 0Fh-IRQ7;
- ◆ Int 70h - CMOS-таймер;
- ◆ Int 71h — IRQ9 (перенаправлено на Int 0Ah);
- ◆ Int 72h- IRQ10;
- ◆ Int 73h — IRQ11;
- ◆ Int 74h — IRQ 12 (контроллер мыши PS/2);
- ◆ Int 75h — IRQ 13 — исключение сопроцессора;
- ◆ Int 76h — IRQ14— контроллер жестких дисков;
- ◆ Int 77h-IRQ15.

**ПРИМЕЧАНИЕ**

Прерывания Int 70h-77h имеют место только в AT.

---

*Функции ROM BIOS (16-битные сервисы):*

- ◆ Int 05h (FOOO:FF54h) - печать экрана;
- ◆ Int 10h — видеосервис;
- ◆ Int 11h — чтение списка оборудования (слово из BDA 0040:001 0h), возвращает в AX:
  - биты 15:14 — число обнаруженных LPT-портов: 00 — 0, ... 11 — 3;
  - бит 13 — резерв;
  - бит 12 — обнаружен игровой адаптер;



- биты 11:9 — число обнаруженных СОМ-портов: 000 — 0, ... 111 — 7;
- бит 8 — наличие контроллера DMA;
- биты 7:6 — число обнаруженных НГМД: 00 — 1, ... 11 — 4;
- биты 5:4 — активный видеорежим: 00 — резерв, 10 — 80-колоночный цветной, 01 — 40-колоночный цветной, 11 — монохромный;
- биты 3:2 — размер ОЗУ на системной плате (теперь обычно 00);
- бит 1 — присутствие математического сопроцессора;
- бит 0 — присутствие дисководов;
- ◆ Int 12h — размер непрерывной памяти;
- ◆ Int 13h — дисковый сервис (блочный ввод-вывод);
- ◆ Int 14h — обслуживание СОМ-портов;
- ◆ Int 15h — АТ-функции (системный сервис, функции определяются значениями ем АН/АХ):
  - 00-03h — управление и обмен данными с кассетным магнитофоном (были когда-то и такие «стриммеры»!) на старых РС;
  - 4fh — перехват клавиатуры;
  - 53xxh — сервисы управления потреблением АРМ (Advanced Power Management);
  - 8300h — запуск таймера, устанавливающего флаг в заданной ячейке;
  - 830 1h — сброс того же таймера;
  - 84h — джойстик (см. п. 8.6);
  - 86h — программируемая задержка;
  - 87h — перемещение блока расширенной памяти;
  - 88h — получение размера расширенной памяти;
  - 89h — переключение в режим V86;
  - С0h — получение системной конфигурации, при успешном выполнении (CF=0, АН=0) ES:ВХ указывает на таблицу данных конфигурации;
  - 80-82h, 85h, 90h, 91h — функции многозадачных ОС (BIOS устанавливает заглушки);
- ◆ Int 16h — клавиатурный ввод-вывод;
- ◆ Int 17h — обслуживание LPT-портов;
- ◆ Int 18h — процедура восстановления при неудаче начальной загрузки (прежде - ROM-Basic);
- ◆ Int 19h — начальная загрузка (вызов процедуры Bootstrap);
- ◆ Int 1Ah — системное время, дата, будильник и 16-битные вызовы сервисов PCI;
- ◆ Int 1Bh — обработчик нажатия клавиш Ctrl+Break;

- ◆ Int 1Ch — User Timer Interrupt, процедура, вызываемая обработчиком Interrupt 08h каждые 55 мс; BIOS устанавливает простую заглушку (IRET), но программы могут перехватывать это прерывание; на время отработки этой процедуры *все аппаратные прерывания запрещены* (кроме NMI).
- ◆ Int 33h — поддержка мыши;
- ◆ Int 4Ah — обработчик будильника пользователя, установленного функцией BIOS Int 1Ah(6); прерывание вызывается асинхронно, так что при возврате из процедуры все регистры и флаги должны быть в том же состоянии, что и при входе; BIOS ставит заглушку (IRET);
- ◆ Int 67h - EMS-функции.

*Указатели на таблицы:*

- ◆ Int 1Dh — видеопараметры;
- ◆ Int 1Eh — параметры дискет;
- ◆ Int 1Fh — знакогенератор CGA;
- ◆ Int 41h — параметры HDD 0;
- ◆ Int 46h — параметры HDD 1;
- ◆ Int 43h — знакогенератор EGA.

### 12.8.1. Int 09h, Int16h — поддержка клавиатуры

Поддержка клавиатуры заключается в обработке прерываний от устройства ввода и предоставлении сервисов ввода прикладным программам.

*Прерывания*, вызванные приходом кодов *нажатия и отпускания клавиш*, обрабатывает BIOS Int 9h. Каждый принятый скан-код (или цепочка) обрабатывается с учетом состояния *клавиатурных флагов*. Результат обработки (как правило, ASCII-символ в младшем байте и скан-код в старшем) помещается в *клавиатурный буфер*, расположенный в ОЗУ. По приему каждого символа указатель головы буфера увеличивается. Буфер организован в виде кольца, после достижения конца области буфера указатель головы установится на начало области. В случае переполнения буфера (указатель головы «догнал» указатель хвоста) очередное слово не записывается, и подается звуковой сигнал. Размер позволяет хранить описание шестнадцати фактов нажатий клавиш. Нажатие клавиш Ctrl, Shift, Alt и некоторых комбинаций в буфере не отмечается, но приводит к модификации бит ячеек флагов клавиатуры. Нажатие «системной» комбинации Ctrl+Alt+Del, клавиши PrintScreen (SysRq) и некоторых других к записи в клавиатурный буфер не приводит, а вызывает специальные процедуры.

Для обслуживания клавиатуры используются ячейки ОЗУ из области данных BIOS (BIOS Data Area):

- ◆ 0:0417,0:418 — флаги клавиатуры;
- ◆ 0:0419 — аккумулятор кода Alt-набора;

- ◆ 0:041A — указатель головы буфера (Buffer Head), 2 байта (модифицируется при помещении символа в буфер);
- ◆ 0:041C — указатель хвоста буфера (Buffer Tail), 2 байта (модифицируется при извлечении символа из буфера);
- ◆ 0:041E-0:042D — область кольцевого буфера (16 слов).

Обработчик аппаратного прерывания до обработки принятого скан-кода вызывает прерывание BIOS Int1Sh с AH=4Fh, а в AL находится принятый скан-код. Стандартный обработчик **Int 15h(4Fh)** просто выполняет возврат с CF=0, но его можно заменить специальным обработчиком, который будет при необходимости подменять принятые скан-коды на какие-либо иные (оставляя их в AL), что должно отмечаться установкой CF=1. В старых версиях BIOS такой возможности перехвата не было, ее наличие можно определить вызовом Int 15h(C0h).

Для клавиатуры *USB* или иного устройства ввода, заменяющего клавиатуру в качестве консоли, прерывание Int 9h должно вызываться программно при обработке каждого клавиатурного события. Обработчик этого прерывания должен выполнять те же действия: скан-код пропускать через Int 15h(4Fh) и помещать в клавиатурный буфер, а также модифицировать флаги клавиатуры.

*Интерфейс прикладного уровня* для клавиатуры представляет BIOS Int 16h. Его основное назначение — извлечение слов из клавиатурного буфера. Функция задается в регистре AH при вызове, результат помещается в регистр AX.

- ◆ AH = 00h — чтение (с ожиданием готовности) и выборка слова из буфера (меняется указатель хвоста). Индикаторы клавиатуры обновляются в соответствии с состоянием флагов. Если буфер пуст, то на AT выполняется прерывание Int 15h (подфункция 90), что может использоваться ОС, например, для переключения задач. Чтобы программа не «зависала» на ожидании символа, предварительно стоит проверить готовность функцией 01h. Символы расширенной клавиатуры фильтруются — преобразуются в их аналоги 83-клавишной клавиатуры.
- ◆ AH = 01h — проверка готовности, чтение без выборки (указатели не изменяются). Признак наличия символа в буфере — установленный флаг ZF.
- ◆ AH = 02h — чтение состояния флагов (в AL — байт 0:417h, см. выше).
- ◆ AH = 03h — установка задержки и частоты автоповтора: BL — код задержки (00=250, 01=500, 02=750, 03=1000 мс), BH - код частоты (см. п. 9.2.1).
- ◆ AH = 05h — запись слова из регистра CX в буфер (меняется указатель головы). Признак успешной записи — AL=Q, если в буфере нет места, то AL=1.
- ◆ AH = 10h и AH=\*11h — функции, аналогичные 00h и 01h, но предназначены специально для 101/102-клавишных клавиатур — в них не выполняется фильтрация символов расширенной клавиатуры. Для ряда клавиш, отсутствующих в клавиатуре AT-84, эти функции дадут результаты, отличающиеся от вызовов 00h и 01h.

♦ AH=12h . — чтение расширенного состояния флагов (в AX — слово KbdShif tFlagsIORec), в котором младший байт совпадает с тем, что дает функция 02h (слово из 0:417h), а старший байт похож на слово из 0:418h. Назначение бит AX:

- бит 0 — клавиша Shift (правая) нажата;
- бит 1 — клавиша Shift (левая) нажата;
- бит 2 — клавиша Ctrl (любая) нажата;
- бит 3 — клавиша Alt (любая) нажата;
- бит 4 — включен индикатор Scroll Lock;
- бит 5 — включен индикатор Num Lock;
- бит 6 — включен индикатор Caps Lock;
- бит 7 — включен режим Insert;
- бит 8 — клавиша Ctrl (левая) нажата;
- бит 9 — клавиша Alt (левая) нажата;
- бит 10 — клавиша Ctrl (правая) нажата;
- бит 11 — клавиша Alt (правая) нажата;
- бит 12 — клавиша Scroll Lock нажата;
- бит 13 — клавиша Num Lock нажата;
- бит 14 — клавиша Caps Lock нажата;
- бит 15 — клавиша SysReq нажата.

*Функции чтения буфера* (00 и 10h) в регистре AL возвращают ASCII-код символа, в AH — *скан-код*. Символы, полученные нестандартным способом (в русском регистре или Alt-набором), сопровождаются нулевым скан-кодом. Alt-набор позволяет ввести в буфер любой символ — для этого его код в десятичной системе набирается на цифровой клавиатуре при нажатой клавише Alt, результат заносится в буфер при отпускании клавиши Alt.

При AL=0 регистр AH содержит *расширенный ASCII-код* (Extended ASCII Keystroke). Дополнительные клавиши 101/102 клавиатур при использовании функций 10h-12h генерируют код E0h в младшем байте и скан-код, соответствующий аналогичным управляющим клавишам 83/84-клавишных клавиатур.

*Функция записи* (05h), несколько неожиданная для клавиатуры, позволяет легко имитировать работу оператора для различных демонстрационных программ. Если прикладная программа не перехватывает обслуживание клавиатуры на уровне аппаратного прерывания (In t 9h), то резидентная программа может ей «подбрасывать» слова в буфер, которые будут восприниматься как нажатие клавиш.

ASCII-коды буфера, соответствующие нажатию клавиш, приведены в [1, 7]. При русификации (или другой локализации) клавиатуры отслеживание переключения регистров (языков) ложится на обработчик аппаратного прерывания клавиатуры.

## 12.8.2. int 10h — видеосервис

Int 10h — видеосервис — предназначен для работы с графическим адаптером. Его первичной задачей является управление видеорежимом (BIOS Video Mode), определяющим формат экрана. BIOS адаптера должна выполнять программирование всех стандартных и специфических управляющих регистров для установки (смены) требуемого видеорежима и выбранных параметров развертки — кроме нее о способах этих переключений остальное ПО может и не знать. В пределах возможностей установленного видеорежима видеосервис предоставляет возможности отображения информации на различных уровнях. Простейший для программиста телетайпный режим позволяет посылать поток символов, которые будут построчно отображаться на экране с обработкой символов возврата каретки, перевода строки, обеспечивая «прокрутку» изображения при заполнении экрана. Есть функции и для полноэкранный работы с текстом, при которой доступны и атрибуты символа. В графическом режиме имеется возможность чтения и записи пиксела с указанными координатами. Однако видеосервисом Int 10h программисты пользуются далеко не всегда, поскольку работает он довольно медленно. Подробно рассматривать функции видеосервиса не будем (этому посвящены отдельные книги), отметим особо лишь *функцию телетайпного вывода* Int 10h(0Eh). При вызове AH=0Eh, в AL — код выводимого символа, в BL — цвет (только для графического режима). Символ выводится в текущую позицию курсора, и курсор сдвигается на следующую, переходя на новую строку после конца предыдущей и прокручивая экран при его заполнении. Специальные символы вызывают возврат на начало строки (CR, код 0Dh), перевод строки (LF, 0Ah) и короткий гудок (BEL, 07h). Этой функцией часто пользуются для вывода сообщений программами, работающими на нижнем уровне (например, модули инициализации ПЗУ расширений BIOS, загрузчики и другие, не имеющие еще доступа к сервисам операционных систем). Программа вывода получается простейшей, работает на всех адаптерах и во всех режимах, но довольно медленно.

## 12.8.3. Int 13h — поддержка дисков

Функции дискового сервиса вызываются программным прерыванием Int 13h. Традиционно дисковый сервис подразделяет физические диски на *дискеты* (diskette) и *фиксированные диски* (fixed disk). Набор функций (табл. 7.8) для этих классов устройств несколько различается как по составу, так и по реализации. Классы различаются по диапазонам номеров физических устройств: для дискет отводятся номера 0-7Fh (реально только 0-3), а для фиксированных дисков — 80h-FFh.

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, модуль BIOS этого контроллера, инициализирующийся во время теста POST, вставал на место Int 13h, а указатель на исходный обработчик дискового сервиса (драйвер НГМД из системной BIOS) сохранялся на месте Int 40h. Хотя поддержка жестких дисков давно уже включена в системную BIOS, ради совместимости

возможность использования прерывания Int 40h 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 — Int 0Eh (линия IRQ 6) и от контроллера жестких дисков — Int 76h (линия IRQ 14). При наличии двухканального порта АТА второй канал обычно задействует линию IRQ 15 (вектор 77h). В XT контроллер жестких дисков занимал линию IRQ 5 (вектор 0Dh). Дополнительные контроллеры дисков могут использовать и другие прерывания. Аппаратные прерывания вырабатываются контроллерами по завершении (нормальному и аварийному) внутренних операций. На эти прерывания BIOS не реагирует, а при инициализации их векторы направляются на программную заглущку (инструкцию IRET).

Стандартные драйверы дисковых функций BIOS (включая и расширенный сервис) имеют однозадачное происхождение. Во время выполнения функции значительное процессорное время может затрачиваться на ожидание завершения операции устройством. Драйверы многозадачного 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.

### 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. Сервис вызывается программным прерыванием Int 13h, при вызове принимаются следующие соглашения:

- ◆ номер функции задается в регистре AH и не должен превышать 3Fh;
- ◆ логический номер диска задается в регистре DL (бит 7 = 0 — признак обращения к НГМД);
- ◆ номер цилиндра (0-1023) задается в регистре CH (младшие 8 бит) и C[7:6] (старшие 2 бита);
- ◆ номер головки (0-255) задается в регистре DH;
- ◆ номер начального сектора (1-63) задается в регистре CL [5:0];
- ◆ 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 ESI: BX;
- ◆ результат выполнения операции определяется по флагу переноса: CF = 0 — успешное выполнение операции, CF = 1 — обнаружены ошибки (код состояния возвращается в регистре AH, код завершения последней операции с дискетами хранится по адресу 40:41h, с жесткими дисками — 40:74h);
- ◆ таблица параметров диска для дискет (DPT) задана указателем в памяти по адресу 0:78h, для жестких дисков (HDPT) — 0:104h или 0:118h.

Список функций традиционного сервиса приведен в табл. 12.7, подробнее они описаны в [4, 9]. Устройства могут не поддерживать некоторые функции, о чем драйверы должны «честно сообщить» кодом возврата 0lh.

Таблица 12.7. Функции традиционного дискового сервиса

Номер функции АН	Назначение параметров	Использование регистров указателей и таблиц				
		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	+	+	+	+	+
08h	Get Drive Parameters — получение параметров диска	+ <sup>3</sup>	+ <sup>3</sup>	+ <sup>3</sup>	-	+ <sup>3</sup>
09h <sup>1</sup>	Initialize Drive Parameters — инициализация таблиц параметров диска	+	-	-	-	+
0Ah <sup>1</sup>	Read Long — «длинное» чтение (сектор и поле ECC)	+	+	+	+	+
0Bh <sup>1</sup>	Write Long — «длинная» запись (сектор и поле ECC)	+	+	+	+	+
0Ch <sup>1</sup>	Seek — поиск цилиндра	+	+	-	-	-
0Dh <sup>1</sup>	Alternative Disk Reset — альтернативный сброс (не затрагивая контроллера дискет)	+	-	-	-	-
10h <sup>1</sup>	Test Drive Ready — проверка готовности	+	-	-	-	-
11h <sup>1</sup>	Recalibrate — рекалибровка (позиционирование на нулевой цилиндр)	+	-	-	-	-
14h <sup>1</sup>	Controller Internal Diagnostics — диагностика контроллера жестких дисков	-	-	-	-	-
15h	Read DASD Type — получение типа диска:	+	-	-	-	-
	<ul style="list-style-type: none"> <li>◆ АН=0 — нет диска;</li> <li>◆ АН=1 — дискета, без датчика смены диска;</li> <li>◆ АН=2 — дискета, с датчиком смены диска;</li> <li>◆ АН=3 — жесткий диск;</li> <li>◆ иные значения — код ошибки.</li> </ul>					
	СХ:DX содержат число 512-байтных секторов на диске					

## 12.8. Сервисы и прерывания BIOS

Номер функции АН	Назначение параметров	Использование регистров указателей и таблиц				
		DL	DH, CL, CH	AL	ES:BX	DPT/HDPT
16h <sup>2</sup>	Diskette Change Line Status — проверка статуса смены дискеты: <ul style="list-style-type: none"> <li>◆ CF=0: АН=0 — смены носителя не было;</li> <li>◆ CF=1: АН=1 — недопустимый номер диска;</li> <li>◆ АН=6 — была смена диска или определение смены не поддерживается;</li> <li>◆ АН=80h — дисковод не готов или не установлен;</li> <li>◆ иные значения — код ошибки</li> </ul>	+	-	-	-	-
17h <sup>2</sup>	Set Diskette Type for Format — установка типа дискеты для форматирования (перед форматированием)	+		+ <sup>3</sup>		
18h <sup>2</sup>	Set Media Type for Format — установка типа носителя (для форматирования)	+	+ <sup>3</sup>	-	-	-
20h <sup>2</sup>	Get Media Type — получение типа установленного носителя	+	-	-	-	-
24h <sup>1</sup>	Set Multiple Mode — установка параметров режима многосекторного обращения (в AL — число секторов за операцию)	+	-	+	-	-
25h <sup>1</sup>	Identify Drive ATA — идентификация накопителя (только для ATA-дисков)	+	-	-	+	-

<sup>1</sup> Только для фиксированных дисков.

<sup>2</sup> Только для дискет и других сменных носителей.

<sup>3</sup> Назначение отличается от обычного.

Формально традиционный сервис позволяет работать с дисками, имеющими до  $1024 \times 256 \times 63 = 16\,515\,072$  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. ниже). Однако при работе с устройствами АТА имеется еще и барьер в 528 Мбайт. Дело в том, что контроллер жесткого диска АТА, на который ориентированы драйверы Int 13h, 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:

$$(2^{10} = 1024 \text{ цилиндра}) \times (2^4 = 16 \text{ головок}) \times (2^6 - 1 = 63 \text{ сектора}) \times 512 \text{ байт} = 528\,482\,304 \text{ байт.}$$

Для преодоления 528-мегабайтного барьера дисков АТА, не трогая программного интерфейса, в BIOS ввели расширение традиционного дискового сервиса. Интер-



фейс АТА в трехмерной геометрии позволяет реализовать довольно большой (но уже не запредельный) объем диска:

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

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 АТА-дисков. В функции, которая сообщает параметры диска (функция 8), производится обратная трансляция, так что *на стороне вызова программного интерфейса Int 13h присутствуют только логические параметры*. Естественно, логический объем диска не может превышать физического:  $(C \times H \times S)_{\text{лог}} \leq (C \times H \times 8)_{\text{физ}}$

Подробнее о преодолении барьеров и способах трансляции (LBA, Large Disk, ECH5) см. в [1,4,9]

### Расширенный сервис BIOS

Чтобы получить возможность работы через BIOS с дисками объема более 8,4 Гбайт, 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 логическому адресу 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 IEEE1394), 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 установки (FDISK, FORMAT), поскольку в регулярной работе применяются собственные 32-разрядные драйверы. Расширения BIOS Int 13h 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;

<sup>1</sup> Зная устройство современных винчестеров, здесь и далее не будем добираться до истинно физических параметров — реального номера цилиндра, головки и сектора.

- ◆ поддержка расширенных дисков (enhanced disk drive (EDD) support subset) — функции 41h и 48h.

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 в сравнении с передачей параметров традиционного сервиса через регистры процессора довольно просторный.

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. Функция *установка аппаратной конфигурации* (4Eh) позволяет управлять режимом передачи (PIO, DMA), а также предварительной выборкой (поиском).

Для эмуляции дисков на загружаемых CD-ROM к сервисам BIOS `Int 13h` добавляется несколько новых функций:

- ◆ *начать/завершить эмуляцию диска* (4Ah/4Bh), *начать эмуляцию диска и вы полнить загрузку* (4Ch);
- ◆ *прочитать секторы загрузочного каталога* (4Dh); 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.

Подробнее расширенный сервис рассмотрен в [4].

#### 12.8.4. `Int 14h` — поддержка COM-портов

COM-порты поддерживаются *сервисом* BIOS `Int 14h`, который обеспечивает описанные ниже функции.

- ◆ `00h` — *инициализация* (установка скорости обмена и формата посылок, заданных регистром AL; запрет источников прерываний). На сигналы DTR и RTS влияния не оказывает (после аппаратного сброса они пассивны).
- ◆ `01h` — *вывод символа* из регистра AL (без аппаратных прерываний). Активируются сигналы DTR и RTS, и после освобождения регистра THR в него помещается выводимый символ. Если за заданное время регистр не освобождается, фиксируется ошибка тайм-аута и функция завершается.

- ◆ 02h — *ввод символа* (без аппаратных прерываний). Активируется только сигнал DTR (RTS переходит в пассивное состояние), и ожидается готовность принятых данных, принятый символ помещается в регистр AL. Если за заданное время данные не получены, функция завершается с ошибкой тайм-аута.
- ◆ 03h — *опрос состояния* модема и линии (чтение регистров MSR и LSR). Эту гарантированно быструю функцию обычно вызывают перед функциями ввода-вывода во избежание риска ожидания тайм-аута.

При вызове `Int 14h` номер функции задается в регистре AH, номер порта (0-3) — в регистре DX (0 — COM1, 1 — COM2...). При возврате из функций 0, 1 и 3 регистр AH содержит *байт состояния линии* (регистр LSR), AL — *байт состояния модема* (MSR). При возврате из функции 2 нулевое значение бита 7 регистра AH указывает на наличие принятого символа в регистре AL; ненулевое значение бита 7 — на ошибку приема, которую можно уточнить функцией 3. *Байт состояния линии* (регистр AH) имеет следующий формат:

- ◆ бит 7 — ошибка тайм-аута (после вызова функции 2 — признак любой ошибки);
- ◆ бит 6 — регистр сдвига передатчика пуст (пауза передачи);
- ◆ бит 5 — промежуточный регистр передатчика пуст (готов принять символ для передачи);
- ◆ бит 4 — обнаружен обрыв линии;
- ◆ бит 3 — ошибка кадра (отсутствие стоп-бита);
- ◆ бит 2 — ошибка паритета принятого символа;
- ◆ бит 1 — переполнение (потеря символа);
- ◆ бит 0 — регистр данных содержит принятый символ.

*Байт состояния модема* (регистр AL при возврате из функций 0, 1, 3) имеет следующий формат:

- ◆ бит 7 — состояние линии DCD;
- ◆ бит 6 — состояние линии RI;
- ◆ бит 5 — состояние линии DSR;
- ◆ бит 4 — состояние линии CTS;
- ◆ бит 3 — изменение состояния DCD;
- ◆ бит 2 — изменение огибающей RI;
- ◆ бит 1 — изменение состояния DSR;
- ◆ бит 0 — изменение состояния CTS.

При *инициализации* порта биты регистра AL имеют следующее назначение:

- ◆ биты [7:5] — скорость обмена:
  - 000=110; 100=1200; 001=150; 101=2400;
  - 010=300; 110=4800; 011=600; 111=9600 бит/с;

- ◆ биты [4:3] — контроль паритета:
  - 01 — число единиц нечетное;
  - 11 — четное;
  - 0 и 10 — без контроля;
- ◆ бит 2 — количество стоп-бит: 0 — 1 бит, 1 — 2 бита (на скорости ПО бит/с — 1,5 стоп-бит);
- ◆ биты [1:0] — длина посылки: 00 — 5 бит, 01 — 6 бит, 10 — 7 бит, 11 — 8 бит.

В процессе начального тестирования POST BIOS проверяет наличие последовательных портов (регистров UART 8250 или совместимых) по стандартным адресам и помещает базовые адреса обнаруженных портов в ячейки BIOS Data Area 0:0400, 0402, 0404, 0406. Эти ячейки хранят адреса портов с логическими именами COM1-COM4. Нулевое значение адреса является признаком отсутствия порта с данным номером. В ячейки 0:047C, 047D, 047E, 047F заносятся константы, задающие тайм-аут для портов.

Обнаруженные порты *инициализируются* на скорость обмена 2400 бит/с, 7 бит данных с контролем на четность (even), 1 стоп-бит. Управляющие сигналы интерфейса DTR и RTS переводятся в исходное состояние («выключено» — положительное напряжение).

### 12.8.5. Int17 h — поддержка принтера

Сервисы *BIOS* Int 17 h: обеспечивают инициализацию, вывод байта данных и опрос состояния принтера, подключенного к LPT-порту. При вызове функция задается в регистре AH, номер LPT-порта — в регистре DX.

- ◆ AH = 00h — *вывод байта* из регистра AL по протоколу *Centronics* (без аппаратных прерываний). Данные помещаются в выходной регистр, и, дождавшись готовности принтера (снятия сигнала Busy), формируется строб.
- ◆ AH = 01h — *инициализация* интерфейса и принтера (установка исходных уровней управляющих сигналов, формирование импульса Init#, запрет аппаратных прерываний и переключение на вывод двунаправленного интерфейса).
- ◆ AH = 02h — *опрос состояния* принтера (чтение регистра состояния порта).

При возврате регистр AH содержит *байт состояния*, который собирается из бит регистра состояния SR [7:3] и программно формируемого флага тайм-аута. Биты 6 и 3 относительно байта, считанного из регистра состояния, инвертированы. Назначение бит байта состояния:

- ◆ бит 7 — не занято (сигнал Busy); нулевое значение означает, что принтер занят (буфер полон или состояние Off-Line, или ошибка);
- ◆ бит 6 — подтверждение (сигнал Ack#); единичное значение означает, что принтер подключен;

- ◆ бит 5 — конец бумаги (сигнал PaperEnd);
- ◆ бит 4 — принтер готов (сигнал Select); нулевое значение означает, что принтер в состоянии Off-Line;
- ◆ бит 3 — ошибка принтера (сигнал Error\*); единичное значение соответствует ошибке;
- ◆ биты 2:1=00 (не используются);
- ◆ бит 0 — флаг тайм-аута, устанавливается при неудачной попытке вывода символа, если сигнал Busy не снимается в течение времени, определенного для данного порта в ячейках тайм-аута (в BIOS Data Area); в этом случае согласно протоколу *Centronics* строб данных не вырабатывается.

Перехват прерывания Int 17h является удобным способом внедрения собственных драйверов принтера. Потребность в них может возникать при подключении к порту принтера с интерфейсом *IPNP* или необходимости перекодировки символов. Если разрабатываемый драйвер предназначен не только для перекодировки, но и изменения протокола (через Int 17h можно организовать вывод через LPT-порт по протоколу *IPNP* и даже через COM-порт), следует внимательно отнестись к битам возвращаемого байта состояния. При их неправильном формировании попытки вывода на печать могут приводить к ошибочным сообщениям.

### 12.8.6. Int 1Ah и Int 15h—поддержка таймеров

*Сервисы BIOS* Int 1Ah позволяют считывать и модифицировать значения системного таймера, даты и времени, а также установки будильника часов реального времени CMOS RTC. Перечисленные ниже номера функций указываются при вызове в регистре AH.

1. AH=0 — чтение системного таймера (двойного слова по адресу 40:006Eh в BIOS Data Area, инкрементируемого по прерываниям от канала 0 счетчика-таймера 8253/8254 примерно раз в 55 мс. Таймер обнуляется при выполнении теста POST после аппаратного сброса). Возвращает значение таймера, в CX — старшую часть, в DX — младшую. AL\_=0, если за последние 24 часа не было переполнения таймера. В современных версиях сброс AL возвращает счетчик переполнений таймера, хранящийся в ячейке 40:0070h (в старых версиях это был флаг).
2. AH=1 — установка системного таймера (CX — старшая часть, в DX — младшая) и сброс флага (счетчика) переполнения таймера в ячейке 40:0070h. В случае ошибки устанавливается флаг CF=1.
3. AH=2 — чтение времени из RTC. Возвращает в упакованном BCD-формате час (в регистре CH), минуту (CL), секунду (DH) и признак коррекции летнего/зимнего времени (DL=1 — коррекция используется, DL=0 — нет). Признаком успешной операции является флаг CF=0. Во избежание ошибок некоторых BIOS при вызове флаг CF должен быть сброшен.
4. AH=3 — установка времени в RTC, назначение регистров и признак результата аналогичен функции 2.

5. AH=4 — чтение даты из RTC. Возвращает в упакованном BCD-формате век(в регистре CH), две старшие цифры года (CL), месяц (DH) и день (DL). Признаком успешной операции является флаг CF=0. Во избежание ошибок некоторых BIOS при вызове флаг CF должен быть сброшен.
6. AH=5 — установка даты в RTC, назначение регистров и признак результата аналогичен функции 4.
7. AH=6 — установка времени срабатывания будильника RTC. Возвращает в упакованном BCD-формате час (в регистре CH), минуту (CL) и секунду (DH). Если будильник уже установлен, переустановка не производится и возвращается флаг CF=1. При срабатывании будильник вызывает прерывание Int 4Ah.
8. AH=7 — отмена установки будильника.

Функции BIOS Int 15h позволяют программировать *таймер CMOS RTC* — вводить задержку или запускать таймер установки флага, указывая время в микросекундах (CX — старшее слово, DX — младшее). Нулевое значение интервала не вызывает никаких действий. Достижимое разрешение в зависимости от производительности ПК может достигать единиц миллисекунд, максимальная выдержка — около 70 часов. Перечисленные ниже номера функций указываются при вызове в регистре AH или AX.

9. AH=86h — задержка на заданное время. Управление будет возвращено вызвавшему процессу только через указанный интервал. По окончании задержки будет установлен бит 7 в ячейке BDA 0040:00A0. Таймер может оказаться занятым, тогда вызов сразу возвратит флаг CF=1 (при успехе CF=0, а в AL окажется маска, записанная в 8259A#2).
  10. AX=8300h — запуск таймера, устанавливающего флаг после указанной задержки бит 7 в ячейке, заданной регистрами ES: BX. При успешном запуске CF=0; если таймер занят (он один) — CF=1 и AL=0. Управление возвращается процессу сразу, а флаг будет установлен через заданное время. Перед завершением программа, запускавшая таймер, должна его сбросить функцией 830 1h (во-первых, чтобы освободить; во-вторых, чтобы снять «адскую машинку», которая неожиданно сама может изменить значение ячейки памяти, вполне возможно уже задействованную другим, ничего не «подозревающим» процессом).
- И. AX=8301h — сброс того же таймера.

### 12.8.7. PCI BIOS

Функции PCI BIOS используются только для поиска и конфигурирования устройств PCI — процедур, требующих доступа к их конфигурационному пространству (см. п. 8.2). Регулярная работа с этими устройствами выполняется через обращения к регистрам устройств по адресам, полученным при конфигурировании, и обработке известных номеров прерываний от этих устройств. Для *16-битного интерфейса реального режима, V86* и *16-битного реального режима*, функции PCI BIOS вызываются через прерывание Int 1Ah; номер функции задается при вызове в регистре AX. Возможна и программная имитация прерывания дальним вызо-

вом по физическому адресу 000FFEBEH (стандартная точка входа в обработчик Int 1Ah) с предварительным занесением в стек регистра флагов.

Признаком нормального выполнения является CF=0 и AH=0; при CF=1 AH содержит *код ошибки*:

- ◆ 81h — неподдерживаемая функция;
- ◆ 83h — неправильный идентификатор производителя;
- ◆ 86h — устройство не найдено;
- ◆ 87h — неправильный номер регистра PCI.

Вызовы требуют глубокого стека (до 1024 байт). Для 32-разрядных вызовов защищенного режима все эти же функции вызываются через точку входа, найденную через каталог 32-разрядных сервисов (см. выше), при этом назначение входных и выходных регистров и флага CF сохраняется. До использования 32-разрядного интерфейса следует сначала найти его каталог и убедиться в наличии сервисов PCI. Функции PCI BIOS перечислены ниже:

- ◆ *AH = B101H* — проверка присутствия PCI BIOS. При наличии **PCI BIOS** возвращает CF=0, AH=0 и EDX=20494350h (строка символов "PCI "); проверяться должны все три признака. При этом в AL находится описатель аппаратного механизма доступа к конфигурационному пространству и генерации специальных циклов PCI:
  - Бит 0 — поддержка механизма №1 для доступа к конфигурационному пространству;
  - Бит 1 — поддержка механизма №2 для доступа к конфигурационному пространству;
  - Биты 2:3=00 (резерв);
  - Бит 4 — поддержка генерации специального цикла по механизму №1;
  - Бит 5 — поддержка генерации специального цикла по механизму №2;
  - Биты 6:7=00 (резерв).

В регистрах BH и BL возвращается старший и младший номер версии (BCD-цифры), в CL — максимальный номер шины PCI, присутствующий в системе (число шин +1, поскольку они нумеруются с нуля последовательно). В регистре EDI может возвращаться линейный адрес точки входа 32-разрядных сервисов BIOS. Этот адрес возвращается не всеми версиями BIOS (некоторые не изменяют EDI); для проверки можно при вызове обнулять EDI и проверять на ноль возвращенное значение.

- ◆ *AH = B102H* — поиск устройства по идентификатору. При вызове в CX указывается идентификатор устройства, в DX — идентификатор производителя, в SI — индекс (порядковый номер) устройства. При успешном возврате в BH — номер шины, в BL[7:3] — номер устройства, BL[2:0] — номер функции. Для нахождения всех устройств с указанными идентификаторами вызовы выполняют, последовательно инкрементируя SI от 0 до получения кода возврата 86h.

- ◆  $AX = B103h$  — поиск устройства по коду класса. При вызове в  $ECX[23:16]$  указывается код класса, в  $ECX[15:8]$  — подкласса, в  $ECX[7:0]$  — интерфейс, в  $SI$  — индекс устройства (аналогично предыдущему). При успешном возврате в  $BH$  — номер шины, в  $BL[7:3]$  — номер устройства,  $BL[2:0]$  — номер функции.
- ◆  $AX = B106h$  — генерация специального цикла *PCI*. При вызове в  $BL$  указывается номер шины, в  $EDX$  — данные специального цикла.
- ◆  $AX = B108h$  — чтение байта конфигурационного пространства устройства *PCI*. При вызове в  $BH$  — номер шины, в  $BL[7:3]$  — номер устройства,  $BL[2:0]$  — номер функции, в  $DI$  — номер регистра (O-FFh). При успешном возврате в  $CL$  — считанный байт.
- ◆  $AX = B109h$  — чтение слова конфигурационного пространства устройства *PCI*. При вызове в  $BH$  — номер шины, в  $BL[7:3]$  — номер устройства,  $BL[2:0]$  — номер функции, в  $DI$  — номер регистра (O-FFh, четный). При успешном возврате в  $CX$  — считанное слово.
- ◆  $AX = B10Ah$  — чтение двойного слова конфигурационного пространства устройства *PCI*. При вызове в  $BH$  — номер шины, в  $BL[7:3]$  — номер устройства,  $BL[2:0]$  — номер функции, в  $DI$  — номер регистра (O-FFh, кратный 4). При успешном возврате в  $ECX$  — считанное двойное слово.
- ◆  $AX = B10Bh$  — запись байта конфигурационного пространства устройства *PCI*. При вызове в  $BH$  — номер шины, в  $BL[7:3]$  — номер устройства,  $BL[2:0]$  — номер функции, в  $DI$  — номер регистра (O-FFh), в  $CL$  — записываемый байт.
- ◆  $AX = B10Ch$  — запись слова конфигурационного пространства устройства *PCI*. При вызове в  $BH$  — номер шины, в  $BL[7:3]$  — номер устройства,  $BL[2:0]$  — номер функции, в  $DI$  — номер регистра (O-FFh, четный), в  $CX$  — записываемое слово.
- ◆  $AX = B10Ah$  — запись двойного слова конфигурационного пространства устройства *PCI*. При вызове в  $BH$  — номер шины, в  $BL[7:3]$  — номер устройства,  $BL[2:0]$  — номер функции, в  $DI$  — номер регистра (O-FFh, кратный 4), в  $ECX$  — записываемое двойное слово.

## 12.9. Расширения ROM BIOS

В микросхеме ROM BIOS, установленной на системной плате, поддерживаются только стандартные (по назначению и реализации) устройства. При необходимости дополнительные устройства, устанавливаемые в слоты шин расширения (ISA, PCI, PCMCIA), могут иметь микросхемы ПЗУ своей программной поддержки — Additional ROM BIOS (дополнительные модули ROM BIOS), они же Expansion ROM. Эта необходимость возникает, когда программная поддержка устройств требуется до загрузки ОС и прикладного ПО. В таком модуле может содержаться и вся программа функционирования специализированного бездискового контроллера на базе PC. Расширения ROM BIOS используют графические адаптеры EGA/VGA/SVGA, некоторые контроллеры жестких дисков, контроллеры SCSI, сетевые адаптеры с удаленной загрузкой и другие периферийные устройства. Для модулей расширения устройств с шиной ISA в пространстве памяти зарезервирована область CS000h-



F4000h. POST сканирует эту область с шагом 2 Кбайт в поисках дополнительных модулей BIOS на завершающем этапе выполнения (после загрузки векторов прерываний указателями на собственные обработчики). Дополнительный модуль BIOS графического адаптера (EGA, VGA, SVGA...) имеет фиксированный адрес C0000 и инициализируется раньше (на шаге инициализации видеоадаптера). Устройства с шиной PCI в своем конфигурационном пространстве содержат лишь признак использования модуля расширения, а его приписку к адресам памяти назначает POST, Дополнительный модуль ROM BIOS должен иметь заголовок, выровненный по границе 2-килобайтной страницы памяти, формат заголовка ПЗУ иллюстрирует табл. 12.8.

**Таблица 12.8.** Заголовок модуля дополнительного ПЗУ

Смещение	Длина	Назначение
0	2	Сигнатура (признак начала модуля): байт 0=55h, байт 1=AAh
2	1	Длина, указанная в блоках по 512 байт
3	3	Точка входа процедуры инициализации, заканчивающейся дальним возвратом Ret Far (вызывается инструкцией Far Call во время POST). Обычно здесь располагается трехбайтная инструкция JMP, указывающая на начало процедуры Резерв
6-17h		Резерв
18h	2	Указатель на структуру данных PCI (только для карт PCI), см. ниже п. 12.9.1
1Ah	2	Указатель на структуру расширенного заголовка карт ISA PnP, см. ниже п. 12.9.2

В традиционном заголовке присутствовали только первые 3 поля, указатели на структуры PCI и ISA PnP ввели позже. Корректным считается модуль, начинающийся с признака AA55H (значения слова с учетом порядка байтов) и нулевой суммой (по модулю 256) всех байтов в объявленной области (реальная длина модуля может превышать объявленную, но байт контрольной суммы, естественно, должен входить в объявленную область).

В случае обнаружения корректного модуля POST дальним вызовом (Call Far) вызывает процедуру инициализации модуля, начинающуюся с 3-го адреса заголовка модуля. Ответственность за ее корректность полностью ложится на разработчика. Процедура может переопределять векторы прерываний, обслуживаемых BIOS. Переопределив на себя Bootstrap (Int 19h), можно получить управление при загрузке, что и используется, например, для удаленной загрузки компьютеров через локальную сеть (Remote Boot Reset). Если стандартное продолжение процедуры загрузки не требуется, а дополнительный модуль представляет собой, например, управляющую программу для какого-либо оборудования, вместо процедуры инициализации в ПЗУ может находиться и основная программа, не возвращающая управление системной последовательности POST.

Процедура инициализации и программная поддержка устройства в ПЗУ должны быть написаны таким образом, чтобы им были безразличны абсолютные адреса,

по которым они размещаются в пространстве памяти. На картах расширения, как правило, имеются аппаратные средства изменения базового адреса, а иногда и размера ПЗУ (джамперы или программно-управляемые переключатели). Это позволяет бесконфликтно разместить модули ПЗУ нескольких установленных карт. По сравнению с традиционным способом использования ПЗУ, когда оно, будучи разрешенным, постоянно присутствует в области памяти, имеется более рациональный способ подключения расширений ROM BIOS, основанный на модели *DDIM* (*Device Driver Initialization Model* — модель инициализации драйвера устройств). POST определяет наличие ПЗУ по найденному заголовку и копирует его содержимое (по объявленной длине) в свободное пространство верхней памяти (ОЗУ), оставляя разрешенной запись в эту область. Далее в этой копии (в ОЗУ) вызывается процедура инициализации (по адресу 3). Эта процедура, написанная в соответствии с моделью *DDIM*, должна определить, каким образом ее запустили: традиционным (в ПЗУ) или в соответствии с *DDIM* (в ОЗУ). Определить это она может просто — попыткой модификации области ее «тела», которая в ПЗУ, естественно, не приведет к изменению содержимого памяти. Обнаружив режим *DDIM* (память модифицируема), процедура выполняет все необходимые действия по инициализации. Далее она определяет, какую часть копии (начиная от начала заголовка) требуется оставить в памяти на время загрузки и регулярной работы системы, отсекая не нужное в дальнейшем тело процедуры инициализации. Попутно она может в оставляемом модуле установить какие-либо параметры, требуемые для работы драйвера устройства. Наконец, она модифицирует поле длины в заголовке и контрольную сумму так, чтобы фрагмент модуля оставался корректным, и дальнейшим возвратом отдает управление тесту POST. Теперь POST снова анализирует заголовок (но уже копии в ОЗУ) и запрещает запись в область ОЗУ (страницу с размером, кратным 4 К), которую «попросила» оставить процедура инициализации. Далее POST, по возможности, запрещает работу (отображение в область UMA) исходного модуля ПЗУ и продолжает свой путь к вызову процедуры начальной загрузки. Основное преимущество данного метода — возможность более рационального использования памяти в UMA (обычные ПЗУ «висят» в ней в полном объеме, невзирая на реальные потребности). Второе преимущество — возможность сохранения параметров, вычисляемых процедурой инициализации (она может задействовать параметры, полученные в процессе выполнения теста POST), в «замораживаемой» области памяти. Кроме того, как правило, медленные микросхемы ПЗУ на все время исполнения подменяются быстрым системным ОЗУ (не используя напрямую механизма теневой памяти). Процедура инициализации ПЗУ карт ISA с моделью *DDIM* должна проверять окружение, в котором она работает (см. выше); безусловная работа *DDIM* гарантируется только для карт PGI.

Для более эффективной работы *DDIM* желательно использовать не только стандартную, но и расширенную память (за пределами первого мегабайта), в то время как POST работает в реальном режиме процессора. Решить эту проблему помогает режим «Big Real Mode», который поддерживают все 32-разрядные процессоры (см. п. 12.3.1). Специально для предоставления доступа ко всей памяти процедурами инициализации фирмы Phoenix и Intel разработали спецификацию *PMM*

(POST Memory Manager Specification), версия 1.01 была опубликована в конце 1997 г. Эта спецификация определяет несколько дополнительных сервисов BIOS, позволяющих выделять, находить и освобождать блоки в любой, в том числе и расширенной памяти. Клиенты этого сервиса запрашивают блок памяти требуемого размера, а BIOS возвращает физический 32-разрядный адрес начала выделяемого блока (если она способна его выделить). Клиент помечает свой блок 32-битным индексом (handle), по которому его в дальнейшем можно найти функцией поиска. Анонимный блок (индекс FFFFFFFFh) поиску не поддается. Этими сервисами можно пользоваться только до начала процедуры начальной загрузки (Int 19h), работу с вентилем *Gate A20* они берут на себя. Перед начальной загрузкой BIOS освобождает и обнуляет все блоки расширенной памяти, занятые с помощью этих сервисов. Сервисами PMM могут пользоваться процедуры инициализации карт расширения, а также процедуры BSV, описанные в заголовке ПЗУ карт PnP (см. ниже). Процедуры, на которые указывает BEV, ими пользоваться не могут, поскольку вызываются после входа Bint 19h. Наличие сервисов PMM определяется по контрольной структуре, начинающейся со строки-сигнатуры \$PMM и расположенной на границе параграфа в области E0000-FFFF0h. Программный интерфейс можно найти в вышеуказанном документе, который доступен на сайте <http://www.phoenix.com/techs>.

### 12.9.1. ROM BIOS карт ISA PnP

Для поддержки технологии PnP и расширения возможностей управления начальной загрузкой в спецификации PnP BIOS ввели дополнительный указатель на *структуру расширенного заголовка* (Expansion Header Structure). Расширенный заголовок имеет формат, приведенный в табл. 12.9. В одном ПЗУ может находиться несколько расширенных заголовков (это требуется для многофункциональных карт расширения), связанных в цепочку. В каждом расширенном заголовке может указываться смещение следующего заголовка относительно начала стандартного заголовка. Наличие и действительность расширенного заголовка проверяется по его сигнатуре и контрольной сумме. Сумма всех байтов расширенного заголовка, включая байт контрольной суммы, должна быть нулевой.

Таблица 12.9. Расширенный заголовок ROM BIOS карт ISA PnP

Смещение	Длина	Назначение
0h	4 байта	Сигнатура, строка \$PnP (символы ASCII)
04h	байт	Версия структуры (01 h)
05h	байт	Длина (в параграфах по 16 байт)
06h	слово	Смещение следующего заголовка (0000h, если нет больше)
08h	байт	Резерв(0)
09h	байт	Контрольная сумма
0Ah	двойное слово	Идентификатор устройства PnP
0Eh	слово	Указатель на строку идентификатора производителя (0, если нет)
10h	слово	Указатель на строку названия продукта (0, если нет)
12h	3байт	Код типа устройства

Смещение	Длина	Назначение
15h	байт	Индикаторы устройства
16h	СЛОВ	Вектор подключения BСV (Boot Connection Vector) — 0, если нет
18h	СЛОВ	Вектор отключения (Disconnect Vector) — 0, если нет
1Ah	СЛОВ	Точка входа для загрузки BEV (Bootstrap Entry Point) — 0, если нет
1Ch	СЛОВ	Резерв (0)
1Eh	СЛОВ	Вектор получения информации о статических ресурсах (Static Resource Information Vector) — Real/Protected mode (0 если нет)

Расширенный заголовок позволяет определить *идентификатор устройства PnP*, его *название* и *код производителя*. *Код типа* состоит из байта общего типа, байта подтипа и байта идентификатора программного интерфейса, по которым система может узнать знакомые устройства.

*Байт индикаторов* устройства имеет следующее назначение битов:

- ◆ бит 7 — ПЗУ поддерживает модель инициализации устройства (Device Driver Initialization Model);
- ◆ бит 6 — ПЗУ может затеняться оперативной памятью;
- ◆ бит 5 — ПЗУ может каптироваться по чтению;
- ◆ бит 4 — ПЗУ требуется лишь для загрузки с данного устройства;
- ◆ бит 3 — резерв (0);
- ◆ бит 2 — является устройством загрузки (IPL-устройство);
- ◆ бит 1 — устройство ввода (может заменять клавиатуру);
- ◆ бит 0 — устройство вывода (дисплей).

*Вектор подключения BСV* (Boot Connection Vector, он же Interrupt Connection Vector) указывает смещение для процедуры, дальний вызов которой приведет к перехвату векторов прерываний первичных устройств ввода, вывода или загрузки (сервисов Int 9h, Int 10h или Int 13h соответственно), в зависимости от параметров, переданных в регистрах процессора. При вызове этой процедуры в регистре AX единицы в битах 0,1 и 2 запрашивают перехват сервисов Int 9h, Int 10h или Int 13h соответственно (остальные биты нулевые), ES: DI указывает на контрольную структуру PnP BIOS (System BIOS PnP Installation Check Structure), BX содержит селективный номер (CSN) карты ISA PnP (для других карт FFFFh); DX содержит адрес порта чтения ISA PnP (для других карт FFFFh).

*Вектор отключения* указывает на процедуру, восстанавливающую старое значение векторов при безуспешной попытке загрузки с данного устройства.

*Точка входа для загрузки BEV* требуется, если устройство может использоваться в качестве загрузочного, но не обеспечивает блочных функций сервиса Int 13h. Тогда системная микросхема BIOS может дальним вызовом вызвать эту процедуру вместо выполнения сервиса Int 19h. Таким образом, например, может выполняться удаленная загрузка (Remote Programm Loading, RPL) по сети.

*Вектор получения информации о статических ресурсах* задает смещение процедуры, вызов которой выгрузит в память дескрипторы занимаемых ресурсов (в форматах, аналогичных структурам для ISA PnP). Адрес буфера размером не менее 1024 байт задается регистрами ES: DI при вызове данной процедуры.

*Процедура инициализации*, которая начинается со смещения 3 в заголовке ПЗУ, для карт PnP должна подчиняться определенным требованиям (до этой спецификации специальных требований не было), принимать параметры и сообщать код возврата.

- ◆ При вызове процедуры E S: DI указывает на контрольную структуру PnP BIOS, BX содержит CSN карты ISA PnP (для других карт FFFFh); DX — адрес порта чтения ISA PnP (для других карт FFFFh).
- ◆ При выполнении процедура может переопределять любые векторы прерываний и изменять данные в BDA и E BDA, но перед возвратом она обязана восстановить прежние указатели для Int 9h, Int 10h, Int 13h и все прежние значения связанных с ними переменных в BDA и EBDA.
- ◆ При возврате в AX возвращаются признаки проинициализированного устройства:
  - бит 8 — устройство начальной загрузки, поддерживающее блочные функции INT 13h;
  - бит 7 — устройство вывода, поддерживающее символьный вывод («телетайпный» режим) INT 10h;
  - бит 6 — устройство ввода, поддерживающее символьный ввод INT 9h;
  - биты 5:4 — состояние подключения загрузочного устройства: 00 — не подключено, 01 — неизвестно, 10 — подключено (для устройства RPL соединение установлено), 11 — резерв;
  - биты 3:2 — состояние подключения устройства вывода (аналогично предыдущему);
  - биты 1:0 — состояние подключения устройства ввода (аналогично предыдущему).

Как видно из данного описания, расширенный заголовок и правила поведения процедуры инициализации позволяют системной BIOS более гибко пользоваться функциями дополнительных модулей BIOS — традиционные дополнительные модули со стороны системной BIOS были практически неуправляемыми. Заголовок данного вида может использоваться и картами ISA без поддержки PnP. Если эти карты будут передавать информацию о своих статических ресурсах, они окажут большую услугу для распределения ресурсов системой BIOS с поддержкой PnP. Таким образом, приспособить карту для работы в среде PnP можно всего лишь модификацией содержимого ее ПЗУ расширения BIOS.

Карта ISA PnP может быть установлена в разные системы, имеющие BIOS как с поддержкой PnP, так и без. Процедура инициализации должна выполняться адекватно обнаруженной среде: без PnP BIOS она должна работать традиционным способом, при необходимости загрузки перехватывая INT 19h, а в среде PnP BIOS она должна вести себя скромнее, лишь предоставляя системной BIOS требуемые интерфейсы и точки входа.

## 12.9.2. Expansion ROM карт PCI

Для содержимого ПЗУ расширения BIOS, установленных на картах PCI, принят стандарт, несколько отличающийся от традиционных дополнительных модулей ROM BIOS. Заголовок ПЗУ соответствует традиционному, но дополнительно имеет указатель на *структуру данных PCI* (табл. 12.10). Идентификаторы производителя и устройства, а также код класса совпадают с описанными в конфигурационном пространстве устройства PCI. Поскольку шина PCI используется не только в PC, в ПЗУ карты может храниться несколько модулей. Каждый модуль начинается со структуры данных, сам модуль следует сразу за структурой. За ним начинается структура для следующего модуля (если у предыдущего не установлен признак последнего модуля) и так далее. Тип платформы (процессора) указывается в заголовке модуля, и при инициализации BIOS активизируется только нужный. Такой механизм позволяет, например, один и тот же графический адаптер устанавливать и в IBM PC, и в Power PC.

Таблица 12.10. Структура данных PCI

Смещение	Длина, байт	Назначение
0	4	Сигнатура, строка символов "PCIR"
4	2	Идентификатор производителя
6	2	Идентификатор устройства
8	2	Резерв <sup>1</sup>
Ah	2	Длина структуры (байт), начиная с сигнатуры
Ch	1	Версия структуры (0 для данной версии)
Dh	3	Код класса
10h	2	Длина образа
12h	2	Версия кода/данных
14h	1	Тип кода: 0 — x86 для PC-AT, 2 — HP PA-RISC
15h	1	Индикатор: 1 — последний образ, 0 — не последний
16h	2	Резерв

<sup>1</sup> До спецификации PCI 2.2 здесь помещался указатель на строку Vital Product Data (важные сведения о продукте).

Применительно к дополнительному ПЗУ карты PCI имеется три параметра, относящихся к размерам. Размер ПЗУ определяется чтением конфигурационного пространства. Размер, указанный во 2-м байте заголовка, указывает на длину модуля на этапе инициализации. Этот модуль POST загружает в ОЗУ перед тем, как вызвать процедуру инициализации (точка входа со смещением 3). Контрольная сумма, расположенная обычно в конце модуля, обеспечивает нулевую сумму всех байт. Длина образа, указанная в структуре данных PCI (слово со смещением 10h), описывает размер области, которая должна оставаться в памяти в режиме нормального функционирования (она может быть меньше, поскольку код процедуры инициализации уже не требуется). Эта область также защищается контрольной суммой. Структура данных PCI должна оставаться в памяти все время.

Работа с модулями ПЗУ для карт PCI выполняется в соответствии с моделью DDIM(см. выше). POST определяет наличие ПЗУ по полю Expansion ROM Base Address в конфигурационном пространстве и назначает ему адрес в свободном пространстве памяти. После этого программированием регистра команд разрешается считывание ПЗУ, и в нем ищется сигнатура заголовка AA55H. Когда сигнатура найдена, POST ищет подходящий образ (по типу кода и совпадающий по идентификаторам с обнаруженными устройствами PCI) и загружает его в ОЗУ (в область C0000-DFFFFh), оставляя разрешенной запись в эту область. Далее чтение ПЗУ запрещается (записью в поле Expansion ROM Base Address), и вызывается процедура инициализации (по адресу 3). При вызове процедуры POST сообщает номер шины (в регистре AH), номер устройства (AL[7:3]) и номер функции (AL[2:0]), благодаря чему процедура узнает точные координаты аппаратных средств. После этого определяется размер области, которую следует оставить в памяти (по байту 2, который может быть модифицирован процедурой инициализации), и для этой области запрещается запись. Если процедура инициализации «урезает» занимаемую память, она должна позаботиться о достоверности контрольной суммы области, описанной байтом 2. Если память освобождается полностью (процедура обнуляет байт 2), то контрольная сумма, естественно, не нужна. Расширение для VGA (определяется по коду класса) обрабатывается особым образом — загружается по адресу C0000h. Процедура инициализации может определить наличие PnP BIOS в системе, проверив значение контрольной структуры PnP по адресу, указанному в E5: DI, и исполняться в зависимости от обнаруженного системного окружения.

## Глава 13

# Интерфейсы питания, заземление и гальваническая развязка

По традиции «аппаратную» книгу заключает глава о «здоровом питании», очень необходимом для выживания аппаратных интерфейсов. Все внутренние устройства ПК, включая и интерфейсные адаптеры, получают напряжение от блока питания системного блока и связаны общей «схемной землей» — шиной GND. Часть внешних устройств получают питание от того же блока питания, пользуясь штатными и нештатными способами. Штатное питание выводится на интерфейсные разъемы клавиатуры и мыши PS/2 (+5 В), USB (+5 В) и Fire Wire (8-40 В). «Нештатным» способом питание можно получить от сигнальных линий LPT-порта (менее 5 В) и COM-порта (двуполярное, около 12 В), но лишь с небольшим током нагрузки и некоторыми аппаратными и программными ухищрениями. Питание от сигнальных цепей интерфейса используется мышью, электронными ключами защиты и иными устройствами с малым потреблением. Остальные внешние устройства питаются от собственных источников питания со своей «схемной землей» и цепями заземления, и при их стыковке с системным блоком (и между собой) могут возникать проблемы заземления, о которых речь пойдет ниже. Проблемы заземления радикально решаются применением гальванической развязки сигнальных цепей от «схемной земли», но эту развязку имеют далеко не все интерфейсы.

### 13.1. Общие вопросы электропитания и заземления

Рассмотрим правила подключения к питающей сети с точки зрения безопасности как человека, так и компьютера.

Практически каждый блок питания компьютера или периферийного устройства имеет сетевой фильтр (рис. 13.1). Конденсаторы этого фильтра предназначены для шунтирования высокочастотных помех питающей сети на землю через провод защитного заземления и соответствующие трехполосные вилку и розетку. «Земляной» провод соединяют с контуром заземления, но допустимо его соединять и с «нулем» силовой сети (разница ощущается только в особо тяжелых условиях эксплу-



атации). При занулении необходимо быть уверенным в том, что «ноль» не станет фазой, если кто-нибудь вдруг перевернет вилку питания. Если же «земляной» провод устройства никуда не подключать, на корпусе устройства появится напряжение порядка 100 В переменного тока (рис. 13.2): конденсаторы фильтра работают как емкостной делитель напряжения, и поскольку их емкость одинакова, 220 В делится пополам.

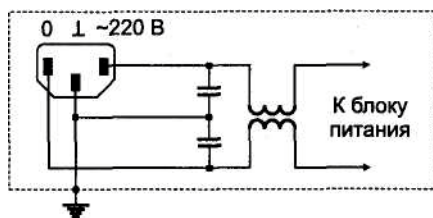


Рис. 13.1. Входные цепи блока питания

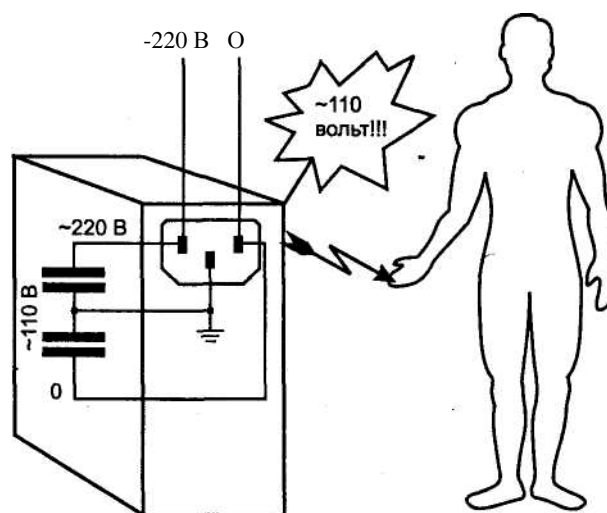


Рис. 13.2. Образование потенциала на корпусе компьютера

Конечно, мощность этого «источника» ограничена — ток короткого замыкания  $I_{КЗ}$  на землю составляет от единиц до десятков миллиампер, причем чем мощнее блок питания, тем больше емкость конденсаторов фильтра:

Здесь  $U_{пит}=220$  В,  $f=50$  Гц — частота питающей сети,  $C$  — емкость конденсатора фильтра. При емкости конденсатора  $C=0,01$  мкФ этот ток будет около 0,7 мА. Заметим, что здесь мы учитываем лишь частоту питающей сети. Для высокочастотных (импульсных) помех, приходящих как по сети, так и от входного преобразователя блока питания, те же конденсаторы представляют собой гораздо меньшее сопротивление, и ток короткого замыкания может возрастать во много раз.

Такое напряжение и ток опасны для человека. Попасть под напряжение можно, прикоснувшись одновременно к неокрашенным металлическим частям корпуса компьютера и, например, к батарее отопления. Это напряжение является одним из источников разности потенциалов между устройствами, от которой страдают интерфейсные схемы.

Посмотрим, что происходит при соединении двух устройств (компьютера и принтера) интерфейсным кабелем. Общий провод интерфейсов последовательных и параллельных портов связан со «схемной землей» и корпусом устройства. Если соединяемые устройства надежно заземлены (занулены) через отдельный провод на общий контур (рис. 13.3), проблемы разности потенциалов не возникает.

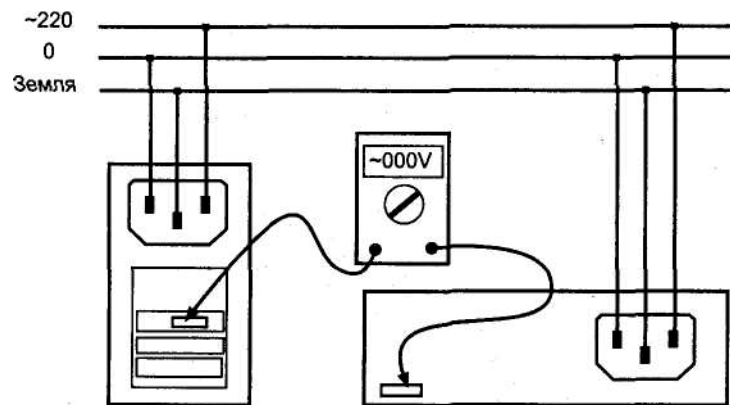


Рис. 13.3. Правильное подключение ПУ

Если же в качестве заземляющего провода использовать нулевой провод питания при разводке питающей сети с трехполюсными розетками двухпроводным кабелем, на нем будет набегать разность потенциалов, вызванная падением напряжения от протекающего силового тока  $I_{NUL}$  (рис. 13.4). Если в эти же розетки включать устройства с большим энергопотреблением, разность потенциалов (и импульсные помехи при включении-выключении) будет ощутимой. При этом эквивалентный источник напряжения при относительно невысокой э.д.с.  $E_{NUL}$  (несколько вольт) будет иметь очень низкое выходное сопротивление, равное сопротивлению участка нулевого провода (доли ом).

Уравнивающий ток через общий провод интерфейса  $I_{INT}$  можно оценить по следующей формуле:

$$I_{INT} = E_{NUL} / (R_{NUL} + R_{INT})$$

Здесь  $E_{NUL} = I_{NUL} * R_{NUL}$ ;  $I_{NUL} = P / 220$ ,  $R_{NUL}$  — сопротивление нулевого провода и соединительных контактов розеток,  $R_{INT}$  — сопротивление общего провода интерфейса,  $P$  — мощность, потребляемая устройствами, расположенными на рисунке справа ( $P = P_2 + P_3$ ). Поскольку обычно сопротивление интерфейсного кабеля больше питающего, через общий провод интерфейса потечет ток, существенно меньший, чем силовой. Но при нарушении контакта в нулевом проводе питания

через интерфейсный провод может протекать и весь ток, потребляемый устройством. Он может достигать нескольких ампер, что повлечет выход устройств из строя. Невыровненные потенциалы корпусов устройств являются также источником помех в интерфейсах.

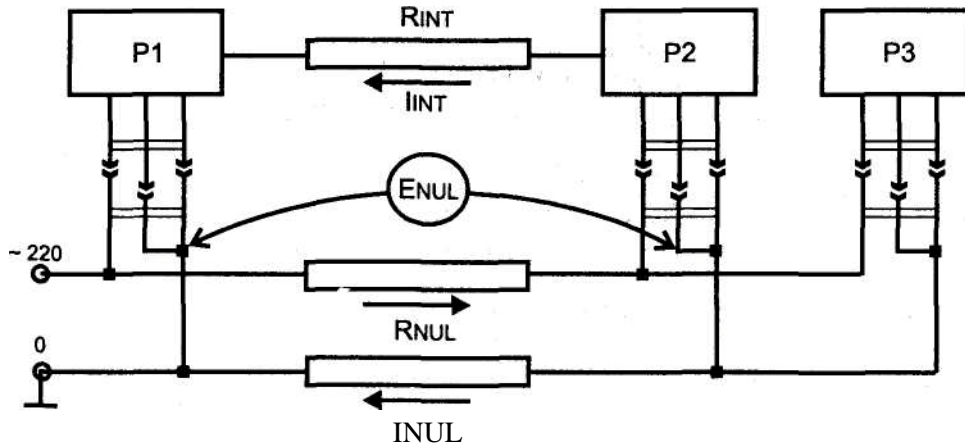


Рис. 13.4. Появление разности потенциалов при двухпроводном кабеле питания

Если оба соединяемых устройства не заземлены, в случае их питания от одной фазы сети разность потенциалов между ними будет небольшой (вызванной разбросом емкостей конденсаторов в разных фильтрах). Уравнивающий ток через общий провод интерфейса будет мал, и разность потенциалов между схемными землями устройств будет тоже мала. Но не следует забывать о безопасности человека. Если незаземленные устройства подключены к разным фазам, разность потенциалов между их несоединенными корпусами будет порядка 190 В, при этом уравнивающий ток через интерфейс может достигать десятка миллиампер. Когда все соединения/разъединения выполняются при отключенном питании, для интерфейсных схем такая ситуация почти безопасна. Но при коммутациях в условиях включенного питания возможны неприятности: если контакты общего провода интерфейса соединяются позже (разъединяются раньше) сигнальных, разность потенциалов между схемными землями прикладывается к сигнальным цепям и они выгорают. Самый тяжелый случай — соединение заземленного устройства с незаземленным (рис. 13.5), особенно когда у последнего мощный блок питания.

Для устройств, блоки питания которых имеют шнуры с двухполюсной вилкой, эти проблемы тоже актуальны. Такие блоки питания зачастую имеют сетевой фильтр, но с конденсаторами малой емкости (ток короткого замыкания достаточно мал). Весьма коварны сетевые шнуры компьютеров с двухполюсной вилкой, которыми подключаются блоки питания с трехполюсным разъемом. Пользователи, подключающие свои компьютеры в бытовые розетки, могут столкнуться с проблемами из-за отсутствия заземления.

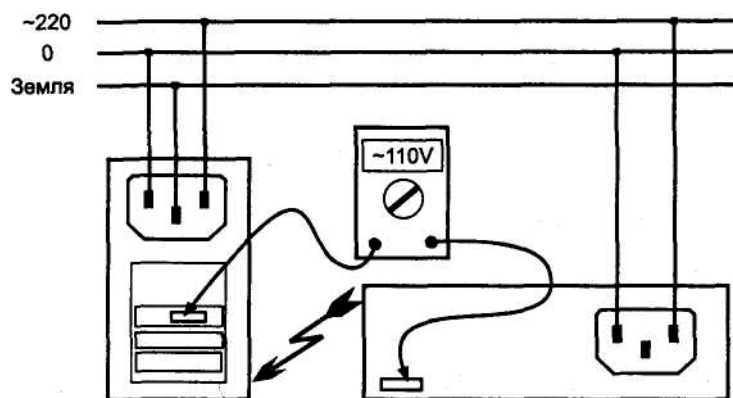


Рис. 13.5. Подключение незаземленного устройства

Локально проблемы заземления решает применение сетевых фильтров типа «Pilot» и им подобных. Питание от одного фильтра всех устройств, соединяемых интерфейсами, решает проблему разности потенциалов. Еще лучше, когда этот фильтр включен в трехполюсную розетку с заземлением (занулением). Однако заземляющие контакты (обжимающие «усики») многих розеток могут иметь плохой контакт вследствие своей слабой упругости или заусениц в пластмассовом кожухе. Кроме того, эти контакты не любят частого вынимания и вставки вилок, так что обесточивание оборудования по окончании работы лучше выполнять выключателем питания фильтра (предварительно выключив устройства).

#### ВНИМАНИЕ-----

Настоятельно рекомендуется отключать питание при подключении и отключении интерфейсных кабелей. Небольшая разность потенциалов, которая практически исчезнет при соединении устройств общими проводами интерфейсов, может пробить входные (и выходные) цепи сигнальных линий, если в момент присоединения разъема контакты общего провода соединятся позже сигнальных. От такой последовательности обычные разъемы не страхуют.

Правила заземления в документации по импортной аппаратуре приводятся не всегда, поскольку подразумевается, что трехполюсная вилка *всегда* должна включаться в соответствующую (трехполюсную) розетку с заземлением, а не в двухполюсную с рассверленными отверстиями. В нашей стране распространены так называемые трехполюсные евროрозетки (рис. 13.6, *я*). Заземление выполняется с помощью контактов-усиков, центральный заземляющий штырь используется нечасто. На рисунке показано правильное положение контактов нуля, фазы и заземления на розетке. При подключении к нему стандартного шнура питания на гнезде, обращенном к блоку питания, раскладка цепей будет соответствовать рис. 13.6, *б*.

К помехам, вызванным разностью потенциалов схемных земель (корпусов) устройств, наиболее чувствительны параллельные порты. У последовательных портов зона нечувствительности шире (пороги  $\pm 3$  В). К этим помехам практически нечувствительны интерфейсы с гальванической развязкой сигнальных цепей от схемной земли.



Рис. 13.6. Положение «нуля», «фазы» и «земли»: а — на питающей розетке, б — на выходном гнезде шнура питания

Проблемы разводки электропитания и заземления стоят особенно остро в локальных сетях, поскольку здесь, как правило, имеется большое количество устройств (компьютеров и коммуникационного оборудования), соединенных между собой интерфейсными кабелями и значительно разнесенных в пространстве (локальная сеть может охватывать и многоэтажное здание). При заземленных корпусах устройств, сильно *разнесенных территориально*, между их корпусами будет разность потенциалов, обусловленная падением напряжения на заземляющих проводах. Эта разность будет особенно ощутимой, если разводка питания и заземления выполнена двухпроводным кабелем (см. рис. 13.4). В сетях на коаксиальном кабеле приходится обеспечивать надежное заземление кабельного сегмента (причем только в одной точке!); нарушение правил заземления коаксиала и соединяемых компьютеров может приводить к сбоям и выгоранию сетевых адаптеров. Коаксиальный кабель и разъемы не должны иметь контактов с металлическими частями корпусов аппаратуры. В сетях на неэкранированной витой паре (UTP) требуется лишь правильно (с заземлением) запитать все компьютеры и коммуникационное оборудование. Использование экранированной витой пары (STP) вносит дополнительные проблемы с соединением и заземлением экранов. Подробнее о решении проблем питания и заземления в сетях см. в [3].

## 13.2. Гальваническая развязка

Гальваническая развязка сигнальных цепей — это отсутствие связи по постоянному току между ними, «схемной землей» и другими питающими шинами. При наличии гальванической развязки потенциал сигнальных цепей относительно «схемной земли» может быть весьма значительным, но не должен превышать напряжения изоляции, допустимого для данного интерфейса. Гальваническая развязка может обеспечиваться разными способами, применимость их зависит от требований к напряжению изоляции и особенностей интерфейсных сигналов.

*Оптическая развязка* позволяет передавать дискретные сигналы в широком диапазоне частот, от постоянного тока до предела, обусловленного быстродействием приемника. Оптическая развязка может выполняться на оптронах — комбинациях излучателя (светодиода) и приемника (фотодиода, фототранзистора с усили-

телем-формирователем) в одной микросхеме. Напряжение изоляции может достигать 1-1,5 кВ, максимальная частота — от десятков кГц до десятков МГц. Оп-тронная развязка применяется, например, в интерфейсах «токовая петля», MIDI. Еще лучшую развязку (по напряжению) обеспечивают интерфейсы с оптоволоконной связью, где между излучателем и приемником располагается оптический кабель с коннекторами. Такая связь применяется в линиях Fiber Channel, оптических версиях Ethernet (и других сетевых технологиях), а также цифровой аудио-технике (S/PDIF). Полоса частот может достигать единиц и десятков Гигагерц, но это требует дорогостоящих излучателей и приемников. В оптических интерфейсах используется стеклянное и пластиковое волокно. Стеклянное волокно позволяет обеспечивать большую дальность связи, но все компоненты довольно дороги, а оконцовка волокна разъемами — довольно сложная процедура, которая может упрощаться за счет применения дорогих компонентов. Если дальность связи ограничивается единицами-десятками метров, то применяют гораздо более дешевое пластиковое волокно.

*Трансформаторная развязка* не позволяет передавать сигналы постоянного тока, но она гораздо дешевле оптической, и достижение высоких частот здесь не имеет столь существенных проблем. Напряжение изоляции разделительных трансформаторов, применяемых в интерфейсных схемах, составляет 0,5-2,5 кВ. Трансформаторная развязка применяется в локальных сетях (все адаптеры электрических версий Ethernet имеют импульсные трансформаторы во входных и выходных цепях), в Fibre Channel, модемах для телефонных и выделенных линий, цифровой аудиотехнике (S/PDIF).

*Конденсаторная развязка* — самый дешевый, но и неэффективный способ развязки, практически не защищающий от помех в интерфейсах. Такая развязка может применяться в дешевых устройствах Fire Wire.

Гальваническая развязка применяется также в источниках питания, где она необходима для обеспечения безопасности работы с устройствами. Гальваническая развязка между входом и выходом имеется у всех источников питания, в которых используются трансформаторы. У источников с трансформаторным входом на первичную обмотку трансформатора подается входное напряжение переменного тока (110-240 В, 50-60 Гц), а ко вторичной обмотке подключается выпрямитель (и стабилизатор напряжения, если имеется). У источников с бестрансформаторным входом основная часть схемы (выпрямитель, преобразователь-стабилизатор) не развязана с входом; отсутствие трансформатора позволяет им работать и от сети постоянного тока.

### 13.3. Интерфейсы блока питания PC

Блок питания PC обеспечивает напряжениями постоянного тока системный блок со всеми его сложными и часто «привередливыми» устройствами. С самых первых моделей PC здесь применяется двухтактная схема преобразователя с бестрансформаторным входом; без революционных изменений эта схема дошла и до наших дней. Преобразователь является регулирующим элементом стабилизатора напря-

жения основного источника +5 В. Остальные напряжения могут быть стабилизированы дополнительными выходными стабилизаторами, но чаще их оставляют нестабилизированными. При этом чем больше нагрузка блока по основной (стабилизированной) цепи, тем выше напряжения на остальных шинах. Блоки питания PC не критичны к частоте сети (50 или 60 Гц) и могут работать даже от сети постоянного тока. Относительно старые блоки питания имеют переключатели диапазона входного напряжения. Переключение диапазона входного напряжения легко осуществляется переключателем, который преобразует мостовую схему выпрямителя в схему выпрямителя с удвоением для питания от сети 110-127 В. При включении блока, предназначенного для работы при напряжении ПО В, в сеть 220 В часто выходят из строя ключевые транзисторы или диоды. Современные блоки, у которых указано свойство Autoswitching Power Supply, имеют компоненты с большим запасом по допустимому напряжению и не требуют переключения номинала входного питающего напряжения — они работают в диапазоне 110-230 В.

Блок питания PC обычно имеет стандартный конструктив и набор жгутов с разъемами питания системной платы и периферийных устройств. Выключатель питания в старых конструктивах располагался на боковой или задней стенке блока питания. Позже его вынесли с блока питания на лицевую панель корпуса и стали присоединять к блоку кабелем со съёмными контактами. К этому кабелю, проходящему через весь системный блок, следует относиться с вниманием, поскольку он является источником опасности и помех. В конструктиве ATX главный выключатель питания вернулся на блок питания, а с передней панели блоком питания управляют с помощью кнопки.

*Мощность блока питания* зависит от назначения корпуса системного блока и лежит в диапазоне от 100-150 Вт для обычных компьютеров и до 350-500 Вт для мощных серверов.

Традиционный (не ATX) блок питания вырабатывает основное стабилизированное напряжение +5 В при токе до 10-50 А; +12 В при токе 3,5-15 А для питания двигателей устройств и интерфейсных цепей; -12 В при токе 0,3-1 А для питания интерфейсных цепей; -5 В при токе 0,3-0,5 А (обычно не используется, присутствует только для соблюдения стандарта ISA Bus).

Кроме питающих напряжений блок вырабатывает сигнал P.G. (Power Good) — питание в норме. Этот сигнал с уровнем в 3-6 В вырабатывается через 0,1-0,5 с после включения питания при нормальных выходных напряжениях блока. При отсутствии этого сигнала на системной плате непрерывно вырабатывается сигнал аппаратного сброса процессора, появление сигнала «выпускает» систему в нормальную работу. Этот сигнал должен сброситься раньше, чем пропадет напряжение +5 В при отключении блока. Отсутствие должной задержки сигнала при включении и запаздывание при выключении может приводить к потере информации в CMOS и ошибкам при загрузке по включении питания. Нажатие кнопки Reset по действию почти эквивалентно замыканию сигнала P.G. на «схемную землю».

Выходные цепи блоков питания выводятся гибкими жгутами проводов со стандартным набором разъемов (рис. 13.7). Разъемы для питания накопителей имеют ключи, исключающие возможность неправильного соединения. Однако иногда встречаются блоки с ошибочно собранными разъемами, в результате на шину питания +5 В попадает +12 В, чего устройства, как правило, не выдерживают. Такая ошибка в практике автора привела, например, к выходу из строя подряд двух 3" дисководов — ошибку в питании стали искать лишь после обнаружения неработоспособности второго дисковода. Традиционные разъемы питания системной платы PS-8, PS-9 всегда устанавливаются рядом так, чтобы четыре черных провода GND шли подряд. Их ключи весьма условны, а ошибка подключения чревата выгоранием системной платы. Цвета проводов в жгутах стандартизованы:

- ◆ GND — черный;
- ◆ -12V — коричневый;
- ◆ +5V — красный;
- ◆ -5V — голубой;
- ◆ +12V — желтый;
- ◆ P.G. — белый (питание в норме).

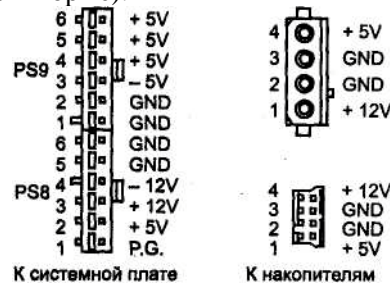


Рис. 13.7. Выходные разъемы традиционного блока питания

Блок питания в стандарте ATX значительно отличается от традиционных блоков питания как по габаритным размерам, так и по электрическому интерфейсу. Блок имеет дополнительный источник напряжением +3,3 В для питания процессора и «дежурный» (Standby) маломощный источник с выходной цепью +5VSB. Дежурный источник с допустимым током нагрузки 10 мА (ATX 2.01) включается при подаче сетевого напряжения. Он предназначен для питания цепей управления энергопотреблением и устройств, активных и в спящем режиме (например, факс-модема, способного при поступлении входящего звонка «разбудить» машину). В дальнейшем предполагается увеличить мощность данного источника до допустимого тока 720 мА, что позволит «будить» компьютер даже по приему пакета от дежурного адаптера локальной сети. В интерфейс блока питания введен управляющий сигнал PS-ON, включающий основные источники +5, +3,3, +12, -12 и -5 В (рис. 13.8). Напряжение от этих источников поступает на выход блока только при



удержании сигнала PS-ON на низком логическом уровне. При высоком уровне или свободном состоянии цепи выходные напряжения этих источников поддерживаются около нулевого уровня. О нормальном напряжении питания сигнализирует сигнал PW-OK (Power O'Key), по действию аналогичный сигналу P.G. традиционных блоков. Интерфейс управления питанием позволяет выполнять программное отключение питания.



Рис. 13.8. Временная диаграмма интерфейса управления питанием АТХ

Все питающие и сигнальные провода к системной плате подключаются одним основным разъемом с надежным ключом (рис. 13.9, а). На разъемах подключения накопителей, естественно, сохранилось традиционное назначение контактов. Расширенная спецификация для блока питания АТХ предусматривает передачу информации от датчиков вентилятора на системную плату, что обеспечивает контроль скорости вращения вентилятора и температуры воздуха. Для этих целей предназначен дополнительный (необязательный) жгут с разъемом, изображенный на рис. 13.9, б. Сигнал FanM представляет собой выход типа «открытый коллектор» от тахометрического датчика вентилятора блока питания, вырабатывающего два импульса на каждый оборот ротора. Сигнал FanC предназначен для управления скоростью вентилятора подачей напряжения в диапазоне 0...+12 В при токе до 20 мА. Если уровень напряжения выше +10,5 В, вентилятор будет работать на максимальной скорости. Уровень ниже +1 В означает запрос от системной платы на остановку вентилятора. Промежуточные значения уровня позволяют плавно регулировать скорость. Внутри блока питания сигнал FanC подтягивается к уровню +12 В, так что, если дополнительный разъем оставить неподключенным, вентилятор будет всегда работать на максимальной скорости. На дополнительном разъеме также имеются контакты 1394V (+) и 1394R (-) изолированного от схемной земли источника напряжения 8-48 В для питания устройств шины IEEE-1394 (FireWire). Цепь +3.3V Sense служит для подачи сигнала обратной связи стабилизатору напряжения +3,3 В.

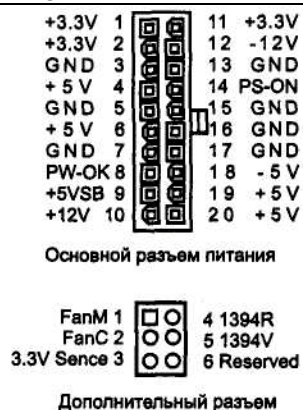


Рис. 13.9. Разъемы блока питания ATX для подключения к системной плате

Цепи блоков питания ATX имеют стандартизованную цветовую маркировку:

- ◆ **COM** - черный (соответствует цепи GND традиционных блоков);
- ◆ +5V — красный;
- ◆ +12V—желтый;
- ◆ -5V - белый;
- ◆ -12V— синий;
- ◆ +3.3V — оранжевый;
- ◆ +3.3V Sense — коричневый (может подходить к контакту 11);
- ◆ +5VSB — малиновый;
- ◆ PS-ON — зеленый;
- ◆ **PW-ON**-серый.

Дополнительный разъем:

- ◆ +3.3V Sense — белый с коричневыми полосками;
- ◆ FanC — белый с синими полосками;
- ◆ FanM — белый;
- ◆ **1394V** — белый с красными полосками;
- ◆ **1394R** —белый с черными полосками.

## Литература

1. *Тук М.* Аппаратные средства IBM PC: Энциклопедия. 2-е изд. — СПб.: Питер Ком, 2001.
2. *Тук М.* Аппаратные средства IBM PC: Энциклопедия. — СПб.: Питер Ком, 1998.
3. *Тук М.* Аппаратные средства локальных сетей: Энциклопедия. — СПб.: Питер, 2000.
4. *Тук М.* Дисковая подсистема ПК. — СПб.: Питер, 2001.
5. *Тук М.* Интерфейсы ПК: Справочник. — СПб.: Питер Ком, 1999.
6. *Тук М., Юров В.* Процессоры Pentium 4, Athlon и другие. — СПб.: Питер, 2001.
7. *Тук М., Юров В.* Процессоры Pentium III, Athlon и другие. — СПб.: Питер, 2000.
8. *Кулаков В.* Программирование на аппаратном уровне: Специальный справочник. - СПб.: Питер, 2001.
9. *Кулаков В.* Программирование дисковых подсистем. — СПб.: Питер, 2002.

## Алфавитный указатель

- 1000BaseCX, 406
- 1000BaseLX, 406
- 1000BaseSX, 406
- 1000BaseT, 406
- 100BaseFX, 406
- 100BaseSX, 406
- 100BaseT4, 406
- 100BaseTX, 406
- 10Base2, 406
- 10BaseS, 405
- 10BaseF, 406
- 10BaseT, 406
- 168-pin Registered DIMM, 267
- 168-pin Unbuffered DIMM, 267
  
- A20, 313,315
- ACCESS.Bus, 331,427
- Additional ROM BIOS, 497
- AGP, 211 режим 1x, 218 2x, 218 4x, 218 сигналы, 212 слот, 219 транзакции, 213
- AGP Pro, 221 AGP8X, 223
- APIC, 461 ASK IR, 79
- ATA, 359
  - Bus-Master, 383 PIO Mode, 377 Ultra DMA, 377 адаптер, 382 джамперы, 388 кабельная выборка, 387 каналы, 383 конфигурирование устройств, 387 протокол, 380 регистры устройства, 371 режим передачи, 376 DMA, 377 PIO, 376 сигналы, 363,366
- ATA (*продолжение*)
  - устройства Master и Slave, 360
  - , хост-адаптер, 360 , электрический интерфейс, 362
  - ATA-2, 361
  - ATA-3, 361
  - ATA/ATAPI-4, 361,362
  - ATAPI, 360
  - ATX, разъем питания, 513
  - AUI, 409
  
- В
- BEDO 240
- Bi-tron'ics, 20
- BIOS
  - 32-разрядные вызовы, 480, /
  - COM-порт, 69,491,493
  - Int0Sh, 325
  - Int 08h, 358
  - Int09h, 316,325
  - Int0Eh, 357
  - Int 13h, 403
  - Int 14h, 48
  - Int 15h, 478
  - Int16h, 316
  - Int17h, 17,320,325
  - Int 19h, 498
  - Int 1 Ah, 210,478
  - Int 33h, 318
  - PCI, 210
  - SMBus, 440
  - видеорежимы, 486
  - интерфейс клавиатуры, 484
  - клавиатура, 316
  - мышь, 318
  - принтер, 325
  - расширение, 497
  - сервисы, 481
  - стандартные драйверы дисков, 487 BIOS Int 15h, 495
  - BIOS Int 16h, 484,493 BIOS Int 1Ah, 494 BIOS Int 9h, 483
  - BIOS32, 480 Bluetooth, 82 BMC-разъемы, 332

- Bootable CD-ROM,  
 404 Boundary Scan,  
 447 bus mastering,  
 411,465 Bus-Master,  
 360,383 Byte Mode,  
 24
- Caller Id, 418  
 CardBus, 229  
 CAS Latency, 241  
 CD-ROM  
   загружаемый, 404  
   системная поддержка,  
 404 Centronics, 321 CHS,  
 адресация, 360 CMOS  
   RTC, 477  
   память конфигурации и часы,  
 477  
   программирование таймера,  
 478,495  
   таймер, 477  
 COM-порт,  
 70  
   UART 8250/16450/16550, 63  
   использование, 70  
   конфигурирование, 70  
   неисправности и  
   тестирование, 73  
   питание от интерфейса, 76  
   поддержка BIOS, 69,493  
   разъемы, 52  
 Compact PCI, 196  
 Composite Video,  
 341 сPCI 196 CR,  
 321
- OCE, 49  
 DDC1, 331  
 DDC2, 331  
 DDC2AB, 331  
 DDC2B, 331  
 DDIM, 499  
 DDMA, 465  
 DDR SDRAM, 267  
 DFP, 336  
 dial-up, 414  
 DIMM, 261  
 DiskOnChip, 479  
 DMA, 21,70,157,463  
   16-битные каналы, 465  
   8-битные каналы, 464  
   LPT-порт, 21  
   PC/PCI, 465  
   каналы, 464  
   пересечение границ,  
 473 DPMS, 332
- DRAM, 235  
   BEDO,  
   240 EDO,  
   239  
 DTE, 49  
 DVI, 337
- E-IDE, 362 ECP,  
 22,29,322 EDD,  
 490 EDO, 239  
 EEPROM, 164  
 EISA, 146,151,164  
 El Tonto, 404 EOI,  
 459 EPP, 22,25  
 Ethernet, 405 AUI,  
 409 7<sup>^</sup>ж 410  
   адаптер, 412 AUI,  
 413 BNC, 413  
   Combo, 413 TP,  
 413  
   интерфейсы, 413  
   конфигурирование, 414  
   ресурсы, 413 серверный,  
 412 интерфейсная карта,  
 410 коммутатор, 405  
   перекрестный кабель, 407  
   повторитель, 405 прямой  
   кабель, 407 сетевой  
   адаптер, 410  
   согласование режимов,  
 409 EVC, 333 expansion  
   bus, 145 Expansion ROM,  
 497,503 Extended ASCII  
   Keystroke, 485
- fast ATA-2, 361  
 Fast Centronics, 21  
 Fast Ethernet, 406  
 Fast IP, 412  
 Fast SCSI, 117  
 Fast Wide SCSI, 117  
 FCAL 140  
 FDC, 356  
 Feature connector, 340  
 Fibre Channel, 140  
   адаптер, 141  
   коммутатор, 141  
   концентратор, 141  
   маршрутизатор,  
   141 мост, 141

- FireWire, 106  
адаптер, 112  
асинхронные сообщения, 110  
динамическое  
реконфигурирование, 111  
диспетчер изохронных ресурсов,  
111  
изохронные передачи, 110  
использование, 113  
кабельная сеть, 107  
контроллер шины, 112  
мастер циклов, 111  
протокол, 109  
разъем, 107  
сравнение с USB, 87  
устройства,  
112 Flow  
Control, 60  
FOIRL, 406  
FRAM, 305
- Game-порт, 349  
GateA20, 315,469  
Gigabit Ethernet, 406
- Н
- Handshaking, 60  
НРЧТП7SIIIMAf  
, »U
- I
- PC, 331,421  
i8042, 313  
IDE  
ATA, AT-BUS, 361  
XT, 371  
разновидности интерфейса,  
361 IEEE 1149.1, 446 IEEE 1284,  
22,322  
ECP, 29  
EPP, 25  
LPT-порт, 22  
PnP, 41  
двунаправленный обмен, 24  
кабели и разъемы, 40  
полубайтный обмен,  
23 IEEE 1394, 106,109  
IN, 376  
Infra Red Connection, 78  
Int05h, 325 Int08h, 358  
Int09h, 316,325 IntOЕh,  
357,487 Int 10h, 71  
Int13h, 142,403  
Int 13h (продолжение)  
параметры вызова, 487  
функции 00-24h, 487 41-  
49h, 491 4A-4D, 491 Int  
u^ 4g59,71,491 Int 15h-  
478 Int 16h, 71,316  
Int17h, 17,320,325 Int  
19h, 498 Int 1Ah, 210,478  
Int33h, 318 Int 40h, 486  
Int 76h, 487 Int 9h, 484 Ir  
TP, 80 IR-Connector, 81  
IrCOMM, 80 IrDA, 78  
FIR, 79  
HDLC, 79  
MIR, 79  
SIR, 79  
приемопередатчики,  
81 IrDA4PPM, 80  
IrLAN, 81 IrLAP, 80  
IrLMP, 80 IrOBEX, 81  
IRQ 15, 487 IRQ 6, 487  
IRQ11, 70 IRQ4, 70  
IRQ5> 20 IRQ7, 20 ISA,  
146  
bus mastering, 159  
Bus-Master, 360  
legacy card, 165  
PnP, 165  
каналы DMA, 147  
карта  
безджамперная, 164  
прототип, 146  
конфигурирование, 164  
порты PnP, 165  
прерывания, 147,160  
пространство памяти, 147  
прямое управление  
шиной 159  
прямой доступ к памяти,  
157  
распределение ресурсов,  
147  
регенерация памяти, 159  
сигналы, 149  
слот, 148  
шина, 146  
ISO\_8482,  
56

JTAG, 446  
 jumperless,  
 164,414

LapLink, 18 LBA,  
 адресация, 361  
 Legacy Card, 455 LF,  
 321 Loop Back, 46  
 Low-Profile PCI, 196  
 LPC, 223 LPT-порт,  
 17  
 Fast Centronics, 21  
 IEEE 1284, 22  
 двунаправленный, 21  
 использование, 42  
 кабель подключения  
 принтера, 323  
 конфигурирование, 44  
 расширения, 21  
 с прямым доступом к  
 памяти, 21  
 связь компьютеров, 42  
 стандартный  
 определение,  
 18 прерывания,  
 20 регистры, 19  
 тестирование, 45  
 функции BIOS, 41

#### М

Magic Packet, 412  
 Master, 360  
 MDI, 407  
 MDIX, 407  
 MI-PC, 232  
 Microwire, 444  
 MIDI, 345,350  
 Mil, 410  
 Miniature Card, 229,402  
 , 400  
 mouse  
 Bus, 319  
 MS Mouse, 317  
 PC Mouse, 317  
 PS/2, 318  
 Serial, 317 MPU-  
 401, 346  
 MultiMediaCard, 400  
**N**  
 Nibble Mode, 20,23  
 NIC, 410 NVRAM,  
 277

OFF-Line, 321  
 OUT, 376  
 Output Enable,  
 156

P&D, 336 P&D-  
 A, 333 P&D-  
 A/D, 336 PC  
 Card, 229  
 сигналы, 229  
 шина, 228  
 PC/104, 161  
 PC/PCI, 465 PC/PCI  
 DMA, 191 PCI, 175  
 BIOS, 210  
 concurrency, 199  
 posted write, 200  
 автоконфигурирование, 204  
 адресация, 183  
 арбитраж, 179  
 ведомое устройство, 177  
 завершение транзакции, 181  
 классы устройств, 209  
 команды, 185  
 конфигурационный  
 механизм, 202  
 мост, 198  
 номер  
 устройства, 177  
 шины, 176  
 отложенные транзакции, 199  
 позитивное декодирование,  
 198  
 прерывания, 189  
 сигналы, 177  
 слоты, 192,193  
 субтрактивное  
 декодирование, 198  
 таймеры, 182  
 устройство, 176  
 частота шины, 193  
 шина, 175  
 электрический интерфейс,  
 192  
 эмуляция DMA, 191  
 PCI Bridge, 198 PCI  
 IDE, 384 PCI-X, 196  
 PCMCIA, 228 Peer-to-  
 Peer Bridge, 198 PIO,  
 376,473 PIO Mode, 377  
 PnP BIOS, 500 PS/2, 21  
 pulse dialing, 416 PXI,  
 197

## Алфавитный указатель

RCA, разъемы,  
341 RDRAM,  
245,269  
    инициализация, 249  
    канал, 246  
    пакеты, 247  
refresh, 238 RGB  
Analog, 327 RJ-11,  
415 RAM, 22,30  
ROM BIOS, 454,500  
RS-232C, 49,55,389  
    аппаратное управление  
        потоком RTS/CTS, 61  
    микросхемы UART, 63  
    программное управление  
        потоком XON/XOFF,  
62 RS-422, 55,56,57 RS-423  
55 RS-485, '55,56,57

S-VHS, 342  
S-Video,  
341 S/PDIF,  
344 SBA,  
213 SCAM,  
116 SCSI  
SPI, 115  
    адресация устройств,  
    129 версии, 116  
    дифференциальная версия,  
    116 инициатор обмена, 129  
    кабели, 118,120  
    конфигурирование, 134,142  
    линейная версия, 116  
    низковольтный  
    дифференциальный  
    интерфейс LVD, 117  
    подключение устройств,  
    135 разъем  
        CX-50, 119  
        DB-25, 119  
        HD-50, 119  
        HD-68, 119  
        IDC-50, 118  
        MiniDSO, 119  
        MiniD68, 119  
        SCA, 120  
        VHDCI-68, 120  
    разъемы, 118,121  
    система команд, 143  
    терминаторы, 125  
    узкий интерфейс, 116

SCSI (*продолжение*)  
    управление, 142  
    фазы 129  
    хост-адаптер, 141  
    целевое устройство, 129  
    шина, 114,134,142  
SCSI.2, ш SCSI-3, 115  
SD 4(ю)  
SDRAM, 240,245,262  
Secure Digital, 400  
Self-Healing Driver,  
412 Serial ATA, 389  
Setup, 44 SIMM, 257  
SIMM-30, 257 SIPP,  
256 Slave, 360  
SmartMedia Card, 399  
SMBus, 433 SMI Mil,  
443 SPCI, 196 SPI,  
115,444 SPP, 18  
SRAM, 274

Tagged VLAN, 412  
TinyTP, 80 tone  
dialing, 416  
Toslink, 345 type 1,  
21

## **И**

UART, 63  
    8250/16450/16550, COM-  
    порт, 63 MPU-401, 346 RS-  
    232C, 63  
Ultra DMA, 361,362,377  
Ultra SCSI, 117  
Ultral60SCSI, 117  
Ultra2SCSI, 117  
Ultra320 SCSI, 117  
unreal, 471  
USB, 88  
    автоконфигурирование,  
    101 адаптивные  
    устройства, 99  
    асинхронные  
    устройства, 98  
    взаимодействие, 89  
    высокая скорость (HS),  
    88 изохронные  
    передачи, 97 интерфейс,  
    90 кадры, 95



## USB

*(продолжение*

) канал, 95  
 конечные точки, 94  
 микрокадр, 96 низкая  
 скорость (LS), 88  
 передачи массивов  
 данных, 96 подключение  
 устройств, 91 полная  
 скорость (FS), 88 поток,  
 95  
 преобразователи  
 интерфейсов, 103  
 прерывания, 97 применение  
 101 протокол, 95  
 синхронизация, 97  
 синхронные устройства, 98  
 сообщения, 95 сравнение с  
 FireWire, 87 удаленное  
 пробуждение, 94 управление  
 энергопотреблением, 93  
 управляющие посылки, 96  
 устройства, 88 функции, 89  
 хаб, 92 хост, 89,99 хост-  
 контроллер, 100 шина, 88

V.11, 56  
 VESA, 332  
 VESA DDC, 331  
 VESA Feature connector, 340  
 VGA Palette Snooping, 201  
 VP&D, 336  
 VT-100, 71  
 VT-42 71

## w

Wide Ultra SCSI, **117**  
 Wide Ultra2 SCSI,  
**117**

X27 56  
 XON/XOFF 62

## Y

Y/C, **342**

## z

ZV Port, **232**

автоконфигурирование  
 PCI, 204  
 USg IQ1  
 устройств, 204  
 адаптер  
 ATA, 382  
 bus master, 411  
 Ethernet, 410  
 Fibre Channel, 141  
 ЛВС,  
 подключение, 43  
 адресация  
 CHS, 360  
 LBA, 361  
 ввода-вывода, 223  
 канальная, 30  
 памяти,  
 216,223 адресный  
 пакет, 491 АКД,  
 49  
 аналоговые  
 аудиоинтерфейсы, 342 АОН,  
 417 АПД, 49  
 аппарат телефонный,  
 415 асинхронный  
 режим, 50

байтный ввод, 24  
 барьер  
 528 Мбайт, 489  
 блок питания  
 ATX, 513  
 PC, 511  
 сетевой фильтр, 505  
 традиционный, 512  
 заблокированный телефон,  
 417

## В

ввод  
 байтный, 24  
 двунаправленный, 24  
 полубайтный,  
 23 ввод-вывод  
 РЮ, 473  
 дешифрация 10/12 бит, 455  
 инструкции, 473  
 карта  
 разрешенных обращений, 474  
 распределения портов, 456  
 вентиль линии А20, 315 видео  
 графические режимы,  
 486 видеоинтерфейсы,  
 341 видеосигнал, 341  
 виртуальная память, 470  
 ВЛС, 412 выборка  
 кабельная, 387

гальваническая развязка, 58,510  
главный мост, 198  
горячее подключение, 89

**д**

двунаправленный LPT-порт, 21  
двунаправленный ввод, 24  
двунаправленный обмен, 24  
джойстик, 349  
дискеты  
дисковый сервис, 486  
диски ATA  
логические параметры, 490  
физические параметры, 490  
дисковый сервис BIOS, 486  
расширенный, 490  
традиционный, 487  
дисплейный адаптер BNC-разъемы, 332  
VESA Feature Connector, 339  
VGA Auxiliary Video Connector, 339  
идентификация монитора, 331  
интерфейс DDC, 331  
RGBTTL, 326  
RGB аналоговый, 327  
композитный видеосигнал, 341  
управление энергопотреблением монитора, 332  
дифференциальный интерфейс SCSI, 116  
драйвер самоизлечивающийся, 412  
дуплексный обмен, 14

заземление, 506  
зануление, 507  
защищенный режим, 469  
звуковая карта, 348

**и**

идентификация источника запроса прерывания, 161  
монитора, 331  
питания, 402  
интерфейс ACCESS.bus, 442  
AUI, 409  
DDC, дисплейный адаптер, 331  
DFP, 336  
DVI, 337

**интерфейс**

*(продолжение)*

Ethernet, 405  
PC, 421,442  
IrDA, 78  
JTAG, 446  
LPC, 223  
microwire, 444  
МП, 410  
P&D, 336  
RGBTTL, дисплейный адаптер, 326  
RGB аналоговый, 327  
RS-232C, 55  
RS-422, 55  
RS-423, 55  
RS-485, 55  
S-VHS, 342  
S-video, 341  
S/PDIF, 344  
SCSI, 116  
Serial ATA, 389  
SMBus, 442  
SMI, 443  
SPI, 444  
USB, 90  
Y/C, 342  
аудио, 342  
аудиоцифровой, 344  
беспроводный, 78  
видео, 341  
гальваническая развязка, 15  
IPNP, 323  
клавиатуры, 311  
монитора, 326  
мыши, 316  
НГМД, 353,354  
параллельный, 13  
Centronics, 321  
LPT-порт, 17  
последовательный, 13  
MIDI, 345  
RS-232C, 49  
асинхронный, 59  
инфракрасный, 78  
принтеров, 324  
токовая петля, 57  
управление потоком, 61  
принтеров и плоттеров, 320  
телефония, 415  
токовая петля, 57  
топология, 14  
устройства хранения, 394  
интерфейсная карта, 410  
интерфейсные кабели, 39  
инфракрасная связь, 78  
IPNP, 323

**К**

кабели, 39  
 Fire Wire, 106  
 IDE ATA, 363  
 MIDI, 345  
 SCSI, 120  
 STP, 406  
 USB, 88,89,333  
 UTP, 108  
 и разъемы, IEEE 1284, 40  
 интерфейса НГМД, 355  
 интерфейсные, 39  
 монитора, 326  
 нуль-модемные, 70  
 питания, 505  
 принтера, 323  
 кабельная выборка, 387  
 каналы ATA, 383  
 карта  
 AGP, 212  
 CompactFlash, 396  
 ISA, 44,81  
 MMC, 400  
 PC Card 81228  
 PCI, 18,81  
 PCMCIf 228,399  
 SD, 400  
 SmartMedia Card, 399  
 прототип, 146  
 квитиование, 60  
 клавиатура, 311  
 Alt-набор, 485  
 ASCII-коды, 485  
 Int09h 316  
 Int16h, 316,483,484,493  
 Int9h, 483,484,493  
 буфер 483  
 интерфейс, 311  
 программный, 484  
 область в BDA, 483  
 прерывания, 483  
 разъемы XT, AT, PS/2, 311  
 расширенный ASCII-код, 485  
 скан-код, 310  
 коммутатор Fibre Channel, 141  
 коммутируемая линия, 414  
 компрессия, 30,34  
 контроллер  
 DMA 8237, 465  
 FDCAT, 356  
 PCI IDE, 384  
 НГМД  
 программирование, 356  
 регистры, 357  
 прерываний 8259A, 456

конфигурирование, 44,89  
 ATA, 387  
 COM-порта, 70  
 jumperless, 414  
 LPT-порта, 44  
 SCSI, 134,142  
 адаптеров Ethernet, 414  
 ISA, 164  
 концентратор Fibre Channel, 141

**Л**

линейный интерфейс SCSI, 116  
 линия телефонная коммутируемая, 414

**М**

манипулятор, 70  
 маршрутизатор Fibre Channel, 141  
 маскируемые прерывания, 458  
 монитор, 341  
 мост  
 Fibre Channel, 141  
 PCI, 198  
 главный, 198  
 одноранговый, 198  
 мышь, 70

**Н**

накопители  
 внешние, 43  
 дисковые, 24  
 подключение, 43  
 НГМД  
 интерфейс, 353,354  
 контроллер, 356  
 программирование, 356  
 регистры контроллера, 357  
 «режим» 356  
 скорость передачи данных, 356  
 смена носителя, 354  
 немаскируемое прерывание, 457

**Обмен**

дуплексный, 14  
 полудуплексный, 14  
 симплексный, 14  
 обработка аппаратного прерывания, 459  
 одноранговый мост, 198  
 ООД, 49

**П**

память  
 DDR SDRAM, 244  
 DRAM, 235

## память

*(продолжение)*

NVRAM, 277  
 RDRAM, 245  
 SDRAM, 240  
   инициализация, 245  
   команды, 240  
   латентность, 241  
 SIPP, 256  
 адаптеры модулей,  
 254 верхняя, 454  
 виртуальная, 470  
 время доступа, 236  
 динамическая, 235  
   BEDO, 240  
   EDO, 239  
   FPM, 238  
   асинхронная, 238  
   временные параметры, 236  
   модули, 251  
   синхронная, 240  
 маркировка модулей,  
 254 модули  
   DIMM, 261  
   DIMM-184, 267  
   DRAM cards, 274  
   ECC, 253  
   RIMM, 269  
   SIMM-30, 257  
   SIMM-72, 258  
   SIPP 257  
   SO DIMM-144, 272  
   SODIMM-72, 271  
   без паритета, 252  
   двусторонние, 257  
   идентификация, 254,255  
   односторонние, 257  
   организация контроля, 252  
   с паритетом, 252  
   с фиктивным паритетом, 252  
   со встроенной коррекцией,  
 253 регенерация, 238  
   CBR, 238  
   ROR, 238  
   автономная, 238  
   без импульса CAS, 238  
   скрытая, 238  
 с интерфейсом  
   I2C, 308  
   SPI, 309  
 с последовательными  
 интерфейсами, 305 средства  
 энергосбережения, 242,249  
 статическая, 274  
   асинхронная, 274  
   синхронная пакетная,  
 275 физическая, 454

память *(продолжение)*

энергонезависимая,  
 276 EEPROM,  
 277,282 EPROM,  
 277,280 FRAM, 277  
 PROM, 277 UV-  
 EPROM, 277  
 масочная, 278  
   однократно программируемая,  
 278 программаторы, 279  
   программирование, 276  
   репрограммируемая, 278  
   стирание EPROM, 279 флэш,  
 282  
 параллельный интерфейс,  
 13 Centronics, 321 SCSI 116  
 подключение  
   адаптеров LBC, 43 накопителей, 43  
   сканера, 43 полубайтный ввод, 23  
   полубайтный обмен, 23  
   полудуплексный обмен, 14  
   последовательный интерфейс,  
 13,49,59,78 поток данных, 60  
   прерывания ISA, 461 LPT-порта, 20  
   NMI, 457 PCI, 460  
   аппаратные, 20,457,459,481  
   клавиатуры, 483  
   конфигурирование, 460  
   конфликты, 463 ложные, 461  
   маскируемые, 458  
   мультипроцессорных систем,  
 461 разделяемые, 462 таблица  
   назначений, 459  
   чувствительность, 461  
   процессор x86, 468 прямое  
   управление шиной, 411,465  
   прямой доступ к памяти,  
 463,465  
  
 развязка  
   гальваническая, 15,510  
   конденсаторная, 511  
   оптическая, 510  
   трансформаторная,  
 511 разъемы  
   ATA, 363  
   COM-порта, 52

разъемы (*продолжение*)

DFP, 337  
 DVI, 337  
 EVC, 333  
 FireWire, 107  
 Game-порта, 349  
 LPT-порта, 42,43  
 MIDI, 346  
 P&D, 336  
 P&D-A, 333,336  
 RCA, 341  
 RS-232C, 52  
 SCSI, 121  
 USB, 89  
 VAFC, 326  
 VFC, 326  
 аудио, 342  
 блока питания ATX, 514  
 видео, 334  
 клавиатуры XT, AT,  
 PS/2, 311  
 НГМД, 354  
 питания  
 накопителей, 513  
 системной платы, 513  
 расширения LPT-порта,  
 21 режим  
 DMAATA, 377  
 большой реальный, 471  
 передачи  
 АТА, 376  
 РЮ, 376  
 стандартный реальный,  
 471

## сброс

счетчика, 494  
 таймера, 495  
 флага, 494  
 связь через LPT-порт,  
 42 сектор, 360 сетевой  
 фильтр, 506 сжатие, 34  
 симплексный  
 обмен, 14  
 системный таймер,  
 476 сканер, 43  
 скорость передачи данных,  
 55 слот  
 AGP, 219  
 AGP Pro, 221  
 ISA, 27,44,137  
 PC Card, 228,229  
 PCI, 113,137  
 PCMCIA, 145  
 Small PCI, 234 средства  
 измерения времени, 476

## таймер

CMOS RTC, 477  
 системная поддержка, 478,494  
 системный, 476  
 трехканальший, 476  
 телефония  
 автоматическое определение  
 номера, 417  
 блокиратор, 417  
 импульсный набор, 416  
 интерфейс, 415  
 параметры сигналов, 418  
 тональный набор,  
 416 телефонная  
 линия, 414  
 телефонный аппарат,  
 415 терминаторы  
 SCSI, 125  
 тестирование, 45  
 COM-порта, 73  
 LPT-порта, 45 токовая  
 петля, 57,58 топология  
 соединения, 14  
 трансляция  
 геометрии, 490 трекбол, 70  
 трехканальный счетчик-таймер,  
 476

## управление SCSI,

142 потоком,  
 369  
 аппаратное RTS/CTS, 61  
 варианты, 60 посылка  
 уведомления, 61 программное  
 XON/XOFF, 62  
 энергопотреблением, 93,332  
 устройства защиты, 307

## фиксированный диск,

486 флэш-память, 282  
 boot block, 284,293  
 bulk erase, 284,291  
 data polling, 301  
 flash file, 284,296  
 NAND, 285  
 NOR, 285  
 PCMCIA, 291  
 smart voltage, 300  
 toggle bit, 301  
 архитектура  
 асимметричная, 284  
 симметричная, 284  
 корпуса, обозначение, 286

## Алфавитный указатель

флэш-память

(продолжение)

назначение

выводов, 287

сигналов, 288

с интерфейсом

DRAM, 286

LPC, 286

с синхронным

интерфейсом, 285

сектарированная, 293

фирмы

AMD, 301

Intel, 283

хост-адаптер, 141

ATA, 360

SCSI, 141

### ш

шина

ACCESS.Bus, 427

CardBus, 229

Compact PCI, 196

EISA, 146,148,151

IEEE 1394, 106

ISA, 146

каналы DMA,

147 прерывания,

147

шина (продолжение)

пространство памяти, 147

распределение ресурсов,  
147

сигналы, 149

слот, 148 Mini

PCI, 196 Miniature

Card, 229 PC

Card, 228,229

PC/104, 161 PCI,

175

автоконфигурирование, 204

команды, 185

мост, 198

слот, 193

PCI-X, 196

PCMCIA, 228

PXI, 197

SCSI, 114,134,142

Small PCI, 196

SMBus, 433 USB,

88 расширения,

145,146

электропитание

заземление,

506

оборудования локальных

сетей, 510 сетевой фильтр,

506