ТОМСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ СИСТЕМ УПРАВЛЕНИЯ И РАДИОЭЛЕКТРОНИКИ (ТУСУР)

С.Г. Михальченко

АППАРАТНОЕ И ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ ЭВМ

Раздел 2

Учебное пособие

ТОМСК — 2007

Федеральное агентство по образованию

ТОМСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ СИСТЕМ УПРАВЛЕНИЯ И РАДИОЭЛЕКТРОНИКИ (ТУСУР)

Кафедра промышленной электроники

С.Г. Михальченко

АППАРАТНОЕ И ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ ЭВМ

Раздел 2

Учебное пособие

2007

Михальченко С.Г.
Аппаратное и программное обеспечение ЭВМ: Учебное пособие. В 2-х разделах. — Томск: Томский государственный университет систем управления и радиоэлектроники, 2007. — Раздел 2. — 155 с.

ОГЛАВЛЕНИЕ

6 ДИСКОВАЯ ПОДСИСТЕМА КОМПЬЮТЕРА

Одной из наиболее важных подсистем компьютера является дисковая подсистема. Основным назначением этой подсистемы является хранение информации — программ и данных.

От эффективности и надежности работы дисковой подсистемы компьютера во многом зависит скорость работы программ и надежность хранения информации. Именно эти факторы, как правило, являются критичными в большинстве случаев использования компьютерной техники, будь то издательские системы или базы данных. Дисковая подсистема включает в себя накопители на жестких и гибких магнитных дисках, а также контроллер диска. Накопители на жестких и гибких магнитных дисках хранят информацию, а контроллер диска предназначен для подключения дисковых накопителей к компьютеру.

В литературе встречаются различные термины для определения накопителей на жестких и гибких магнитных дисках. Накопители на жестких магнитных дисках (***НЖМД***) называют жесткими дисками, винчестерами, а также используют аббревиатуру ***HDD*** (от названия *Hard Disk Drive*, что означает «дисковод для жесткого диска»). Накопители на гибких магнитных дисках (***НГМД***) называют дисководами для флоппи-дисков или используют аббревиатуру ***FDD*** (от английского *Floppy Disk Drive*, что означает «дисковод для флоппи-дисков»).

6.1 Контроллер диска

Контроллер диска — это специальное устройство, предназначенное для подключения жестких и гибких дисков к компьютеру. Контроллер выполняет всю работу по обмену данными между компьютером и дисками. Физически контроллер может быть выполнен в виде отдельной платы, вставляемой в слот расширения материнской платы компьютера, или может быть расположен непосредственно на материнской плате.

Обычно один контроллер диска можно использовать для подключения двух жестких и двух гибких дисков. Без использования дополнительных программных средств операционная сис-

тема может задействовать два накопителя на жестких дисках и два накопителя на гибких дисках.

Существует несколько типов контроллеров диска, отличающихся по способу подключения к дисководам, протоколу обмена данными между контроллером и накопителем на магнитных дисках, скоростью передачи данных и другими характеристиками.

Интерфейс ST506/412

Интерфейс ST506/412 использовался преимущественно в компьютерах IBM XT и IBM AT с небольшой емкостью диска — от 20 до 40 мегабайт. Характерным признаком этого интерфейса являлось подключение жесткого диска к контроллеру при помощи двух плоских кабелей (один кабель из 34 жил, второй — из 20 жил). К одному контроллеру могло быть подключено два жестких диска (в этом случае использовалось три кабеля — один широкий с двумя разъемами, и два узких). В настоящее время данный интерфейс устарел, вместо него используются IDE, ESDI или SCSI.

Интерфейс ESDI

Интерфейс ESDI обычно применялся в компьютерах IBM AT с тактовой частотой больше 16 мегагерц и с процессорами 80286 или 80386. Использовался для подключения жестких дисков емкостью более 100 МБ. К одному контроллеру можно было подключить два жестких диска — также двумя кабелями шириной 34 и 20 жил (дополнительный признак, по которому можно отличить контроллер ESDI от контроллера ST506/412 — наличие на плате контроллера микросхемы ПЗУ).

Низкоуровневое форматирование жесткого диска, подключенного к контроллеру ESDI, должно выполняться только с помощью программы, записанной в микросхеме ПЗУ контроллера.

Интерфейс SCSI

Интерфейс SCSI используется для подключения дисков большой емкости к высокопроизводительным компьютерам. Характерная особенность этого интерфейса — использование одного широкого кабеля (50 жил) для подключения всех дисковых накопителей.

Интерфейс SCSI используется не только для подключения жестких дисков. Данный интерфейс применяется для подключения к компьютеру таких устройств, как принтеры, сканеры, диски Бернулли, лазерные диски, перезаписываемые магнитооптические диски и т.д.

SCSI не накладывает никаких ограничений на связь между контроллером и периферийным устройством. Шину SCSI можно использовать для связи компьютера с несколькими периферийными устройствами (как внешними, так и внутренними). Более того, допускается совместное использование одного периферийного устройства несколькими компьютерами, подключенными к общей шине SCSI. Подключаемые к шине SCSI устройства могут играть роль ведущих (*Initiator*) или ведомых (*Target*), при этом одно и то же устройство может быть ведомым в одних случаях и ведущим — в других. Обмен между устройствами по магистрали SCSI происходит в соответствии с протоколом высокого уровня и адресация осуществляется на уровне логических блоков. Программы для работы со SCSI-устройствами не используют физические характеристики конкретного устройства (число головок, цилиндров и т.п.), а имеют дело с логическими блоками, что дает возможность работы фактически со всеми блочными устройствами.

Для подключения устройств SCSI возможны как синфазная, так и дифференциальная (с помощью «токовой петли») передача данных по кабелю; при синфазной передаче длина кабеля может достигать 6 м, при дифференциальной — 25 м. Для гарантированной передачи сигналов по магистрали SCSI линию требуется согласовывать с помощью терминаторов, устанавливаемых по обоим концам шины SCSI.

Спецификация SCSI предусматривает подключение к шине до восьми устройств, однако с учетом того, что каждое устройство может содержать 8 логических блоков, а каждый блок — 256 подблоков, возможности расширения являются фактически неограниченными. Каждое подключаемое к шине SCSI устройство имеет свой идентификатор, устанавливаемый с помощью перемычек или переключателей непосредственно в устройстве. Идентификаторы позволяют адресовать устройства и задают их приоритет (чем больше значение идентификатора, тем выше приоритет устройства).

На протяжении последних лет интерфейс SCSI был существенно расширен — появились спецификации ***Fast-SCSI*** и ***Wide-SCSI***, обеспечивающие более высокую скорость обмена данными с устройствами SCSI. В настоящее время интерфейс SCSI используется в основном в высокопроизводительных системах, предназначенных для коллективного использования (диски файловых серверов, сканеры и т.д.). В современных персональных компьютерах наиболее распространена дисковая подсистема IDE/ATA.

Интерфейс IDE/ATA

Спецификация IDE/ATA была предложена в качестве недорогой альтернативы интерфейсам ESDI и SCSI для персональных компьютеров семейств IBM PC XT/AT. Большинство функций контроллера реализовано непосредственно на плате дискового накопителя.

Функции контроллера перенесены на плату управления приводом диска и головок винчестера. Информация о геометрии диска (число головок, цилиндров и секторов) хранится в самом устройстве.

Как правило диски IDE имеют небольшую встроенную кэш-память (до 256 Кб) и позволяют работать с фактором чередования 1:1 (дорожка может быть прочитана целиком за один оборот диска).

Хост-адаптер для подключения дисков IDE зачастую устанавливается на системной плате. Подключение устройств к хост-адаптеру осуществляется с помощью 40-проводного плоского кабеля, к которому можно присоединить два винчестера. Для корректной адресации устройств один из винчестеров должен быть установлен в режим *Master* (ведущий), другой — в режим *Slave* (ведомый). Режим работы диска задается с помощью перемычек, расположенных, как правило, около сигнального разъема винчестера.

6.2 Типы накопителей на гибких дисках

Накопитель на гибких магнитных дисках — *НГМД* необходим для загрузки на компьютер программного обеспечения с гибких дисков — *дискет*, обмена данными с другими компьютерами, а также для создания на дискетах архивов данных и для ре-

зервного копирования ценной информации. Наиболее распространены на компьютерах типа IBM PC/XT/AT накопители на гибких магнитных дисках диаметром ***5,25*** и ***3,5*** дюйма.

Накопители на гибких магнитных дисках диаметром 5,25» бывают двух основных типов — двойной и высокой плотности. Первые позволяют записать на магнитный диск до 360 Кбайт, а вторые до 1,2 Мбайт информации. Магнитное покрытие дискет с двойной плотностью записи отличается от магнитного покрытия дискет с высокой плотностью.

Накопители на гибких магнитных дисках диаметром 3,5» также бывают двух основных типов — двойной и высокой плотности. Первые позволяют записать на магнитный диск 720 Кбайт, а вторые 1,44 Мбайт информации. Существуют также 3,5» дисководы сверхвысокой плотности, позволяющий хранить на дискете 2,88 Мбайт информации.

Дисковод с двойной плотностью записи позволяет разместить на дискете 40 дорожек, а дисковод с высокой плотностью записи — в два раза больше (80 дорожек). Это достигается за счет увеличения плотности записи на дискете, что возможно только на дискетах со специальным магнитным слоем.

6.3 Форматирование магнитных дисков

Жесткий диск состоит из нескольких магнитных дисков, вращающихся вокруг своей оси с большой скоростью. С обеих сторон каждого магнитного диска расположены магнитные головки. С помощью специального двигателя эти головки могут перемещаться вдоль радиуса диска. Контроллер диска может устанавливать их неподвижно на различном расстоянии от центра диска. Так как диски вращаются, то магнитные головки могут считывать и записывать информацию, расположенную на различных концентрических окружностях магнитных дисков, называемых *цилиндрами* или *треками*.

Запись на жестком (и гибком) магнитном диске производится отдельными блоками в отдельные сектора. Каждый сектор, кроме данных, содержит различную служебную информацию, необходимую для правильного функционирования контроллера дисковода. В частности эта служебная информация включает та-

кие данные, как номер дорожки, номер сектора и контрольную сумму данных, записанных в секторе.

Процедура форматирования как раз и включает в себя разметку диска на отдельные *дорожки* и *сектора*. Этот тип форматирования называется ***низкоуровневым форматированием***. После форматирования на низком уровне диск еще не готов к использованию операционной системой для записи файлов. Его надо специально подготовить к хранению файлов, для чего требуется выполнить операцию форматирования на высоком уровне. ***Форматирование на высоком уровне*** формирует на логическом диске *загрузочный сектор*, *таблицу распределения файлов* и *корневой каталог*.

После низкоуровневого форматирования (если оно необходимо) можно приступить к созданию на диске разделов и логических дисков. Для чего требуется выполнять следующие действия:

- создать первичный раздел;
- создать дополнительный (расширенный) раздел;
- создать на дополнительном разделе несколько логических дисков;
- выбрать активный (загрузочный) раздел.

Первичный раздел будет соответствовать первому логическому диску.

6.4 Интерфейс ATA

Если возникает потребность в выполнении низкоуровневых операций, то с гибкими дисками удобнее и безопаснее работать через функции BIOS (они более стандартны, чем контроллеры дисководов, разработанные разными фирмами), а с жесткими дисками, наоборот, гораздо проще взаимодействовать напрямую (взаимодействие по стандарту ATA унифицировано лучше, чем функции BIOS у различных изготовителей материнских плат).

Непосредственная работа с регистрами контроллера жесткого диска

Необходимость работать напрямую с регистрами контроллера диска возникает в следующих случаях:

- при переключении процессора в защищенный режим (прерывания DOS и BIOS становятся недоступными);
- при работе с дисками большого (свыше 8 Гбайт) объема или нестандартного формата;
- в измерительных системах и системах управления реального времени (для ускорения операций ввода/вывода);
- в системах с повышенными требованиями к обеспечению защиты информации (для предотвращения перехвата управления на уровне прерываний, например, компьютерным вирусом).

Стандарт ATA позволяет подключать к каждому каналу, то есть к одному кабелю, по два устройства. Современные IBM-подобные персональные компьютеры допускают использование до четырех каналов, следовательно, в общей сложности можно установить на компьютер до 8 дисководов с интерфейсом этого типа. Однако встроенный контроллер материнской платы поддерживает только два канала. Еще два дополнительных канала могут быть реализованы на платах расширения (например, для подключения устройства чтения компакт-дисков через звуковую карту), однако следует иметь в виду, что системный BIOS не будет с ними работать, в том числе — не будет выполнять процедуру поиска подключенных к ним устройств при включении питания. Для работы с дополнительными каналами нужно устанавливать специальные драйверы. Наличие на материнской плате двух каналов для подключения ATA — устройств фактически стало стандартом. Для этих каналов жестко закреплен диапазон адресов ввода/вывода и номера используемых прерываний (каналы 1 и 2 в табл. 6.1).

Таблица 6.1 — **Пространство ввода/вывода дисководов ATA**

Номер канала	Диапазон адресов		Номер IRQ сигнала прерывания
	Регистры команд	Регистры контроля	
1	1F0h—1F7h	3F6h—3F7h	14
2	170h—177h	376h—377h	15
3	1E8h—1EFh	3EEh—3EFh	11
4	168h—16Fh	36Eh—36Fh	10

С дополнительными каналами ситуация менее определенная — фирмы-изготовители договорились между собой о диапазоне адресов регистров, однако номера прерываний не закреплены жестко, они являются рекомендованными — изготовитель, пользователь или операционная система могут вместо них использовать любые другие свободные номера.

В табл. 6.2 даны адреса регистров для двух основных каналов (для дополнительных каналов порядок регистров аналогичный). Для адресации данных на диске с интерфейсом ATA можно использовать либо режим ***LBA***, либо режим ***CHS***, причем назначение регистров контроллера зависит от используемого режима адресации. Все регистры, за исключением регистра данных, 8-разрядные, а регистр данных — 16-разрядный.

Таблица 6.2 — **Функциональное назначение регистров контроллера жесткого диска**

<table>
<tr><td colspan="2">Адрес регистра</td><td colspan="2">Назначение регистра</td></tr>
<tr><td>Канал 1</td><td>Канал 2</td><td>В режиме чтения</td><td>В режиме записи</td></tr>
<tr><td>1FGh</td><td>170h</td><td colspan="2">регистр данных (DR)</td></tr>
<tr><td>1Flh</td><td>17 lh</td><td>регистр ошибок (ER)</td><td>регистр свойств (FR)</td></tr>
<tr><td>1F2h</td><td>172h</td><td>счетчик секторов (SC)</td><td></td></tr>
<tr><td rowspan="2">1F3h</td><td rowspan="2">173h</td><td colspan="2">CHS: регистр номера сектора (SN)</td></tr>
<tr><td colspan="2">LBA: разряды 0—7 адреса сектора</td></tr>
<tr><td rowspan="2">1F4h</td><td rowspan="2">174h</td><td colspan="2">CHS: регистр младшего байта номера цилиндра (CL)</td></tr>
<tr><td colspan="2">LBA: разряды 8—15 адреса сектора</td></tr>
<tr><td rowspan="2">1F5h</td><td rowspan="2">175h</td><td colspan="2">CHS: регистр старшего байта номера цилиндра (CH)</td></tr>
<tr><td colspan="2">LBA: разряды 16—23 адреса сектора</td></tr>
<tr><td rowspan="2">1F6h</td><td rowspan="2">176h</td><td colspan="2">CHS: регистр номера устройства и головки (DH)</td></tr>
<tr><td colspan="2">LBA: номер устр. и разряды 24—27 адреса сектора</td></tr>
<tr><td>1F7h</td><td>177h</td><td>регистр состояния (SR)</td><td>регистр команд (CR)</td></tr>
<tr><td>3F6h</td><td>376h</td><td>альтернативный регистр состояния (AC)</td><td>регистр управления устройством (DC)</td></tr>
<tr><td>3F7h</td><td>377h</td><td colspan="2">не используется</td></tr>
</table>

Рассмотрим формат регистров более подробно. Регистр данных (DR) используется при выполнении операции чтения или записи сектора в программном режиме ввода/вывода (PIO). Этот регистр недоступен, пока не начнется операция чтения или записи. Нельзя обращаться к регистру, когда происходит обмен ин-

формацией между диском и памятью в режиме прямого доступа (DMA).

Передача данных через регистр осуществляется 16-разрядными словами. Обратите внимание: это 16-разрядный регистр и его адресное пространство перекрывает следующий за ним регистр ошибок. Даже в том случае, если старший байт данных не используется (такая ситуация возможна в некоторых командах), чтение/запись все равно нужно выполнять целыми словами (обнуляя старший байт перед операцией записи и после операции чтения).

Специально для упрощения и ускорения работы с регистром данных контроллера жесткого диска в набор команд процессоров с архитектурой x86 были введены операции группового ввода/вывода INSW и OUTSW, хотя можно работать с данными и при помощи обычных команд ввода/вывода IN и OUT.

Регистр ошибок (ER)

Регистр ошибок (ER) доступен только по чтению. Он определяет состояние адаптера после выполнения операции. Содержимое этого регистра нужно проверять в двух случаях:

- после выполнения любой команды, если установлен бит ошибки ERR в регистре состояния;
- после выполнения команды «диагностика» или после выполнения внутренней диагностики адаптера по системному сбросу.

Коды регистра ошибок после выполнения команды «диагностика»:

- 01h — нет ошибки: устройство 0 исправно, устройство 1 — либо исправно, либо не присутствует в системе (не подключено);
- 00, 02h—7Fh — устройство 0 неисправно, устройство 1 — либо исправно, либо не подключено;
- 81h — устройство 0 исправно, устройство 1 неисправно;
- 80h, 82h—FFh — оба устройства (0 и 1) неисправны.

Во всех остальных случаях разряды регистра ошибок, формат которого показан на рис. 6.3, имеют следующее назначение:

- бит 0 (AMNF) — не найден адресный маркер сектора;
- бит 1 (TK0NF) — не найдена нулевая дорожка при выполнении команды «рекалибровка»;

- бит 2 (ABRT) — аварийное прекращение выполнения команды;
- бит 3 (MCR) — получен запрос на смену носителя информации;
- бит 4 (IDNF) — сектор с заданными координатами (цилиндр, головка, сектор) не найден;
- бит 5 (MC) — произведена смена носителя информации;
- бит 6 (UNC) — некорректируемая ошибка данных;
- бит 7 — зарезервирован.

Если при выполнении команды ошибок не было, то все разряды регистра ошибок содержат нули. Если при выполнении команды происходит какая-либо ошибка, то соответствующий разряд регистра устанавливается в 1.

X	UNC	MC	IDNF	MCR	ABRT	TKONF	AMNF
7	6	5	4	3	2	1	0

Рис. 6.3 — Формат регистра ошибок

Регистр свойств (FR)

Регистр свойств (FR) расположен по тому же адресу, что и регистр ошибок, но доступен только для записи. Формат данных регистра изменяется в зависимости от команды. При выполнении обычных операций ввода/вывода этот регистр не применяется.

Регистр счетчика секторов (SC)

В регистр счетчика секторов (SC) заносится количество секторов, которое должно быть считано или записано (при записи 0 в этот регистр происходит обработка 256 секторов). Значение этого регистра уменьшается на единицу после обработки каждого сектора. При выполнении мультисекторной операции сектора должны располагаться на диске последовательно, друг за другом (то есть область данных должна быть непрерывной). Этот регистр доступен не только для записи, но и для считывания — в случае возникновения ошибки при выполнении операции чтения или записи в этом регистре будет находиться число секторов, оставшихся необработанными.

Регистр номера сектора (SN)

В регистр номера сектора (SN) в режиме CHS загружается стартовый номер сектора при операциях чтения-записи. После обработки каждого сектора в этот регистр автоматически заносится номер следующего сектора, подлежащего обработке. Регистр доступен для чтения/записи. После выполнения команды он содержит номер последнего обработанного сектора. В режиме LBA регистр номера сектора содержит младший байт линейного адреса сектора (разряды 0—7).

Регистры номера цилиндра

Регистры младшего (CL) и старшего (CR) байтов номера цилиндра в режиме CHS определяют стартовый цилиндр для выполнения команды (в старых контроллерах дисков использовалось только 2 младших разряда регистра CR, то есть максимальный номер цилиндра был равен 1023). Регистры доступны для чтения/ записи. После выполнения команды они содержат текущий адрес цилиндра.

В режиме LBA регистр CL содержит разряды 8—15, а регистр CR — разряды 16—23 линейного адреса сектора.

В режиме CHS

1	0	1	DEV	HS3	HS2	HS1	HS0
7	6	5	4	3	2	1	0

В режиме LBA

1	1	1	DEV	LBA27	LBA26	LBA25	LBA24
7	6	5	4	3	2	1	0

Регистр номера устройства и головки

Регистр доступен для чтения и записи. В режиме CHS разряды регистра имеют следующие значения

- биты 0—3 (HS0—HS3) — номер головки;
- бит 4 (DEV) — выбор устройства (0 или 1);
- бит 5 — зарезервирован (должен быть установлен в 1);
- бит 6 (LBA) — признак режима LBA (должен быть сброшен в 0);
- бит 7 — зарезервирован (должен быть установлен в 1).

В режиме LBA назначение разрядов следующее:

- биты 0—3 (LBA24—LBA27) — разряды 24—27 линейного адреса сектора;
- бит 4 (DEV) — выбор устройства (0 или 1);
- бит 5 — зарезервирован (должен быть установлен в 1);
- бит 6 (LBA) — признак режима LBA (установлен в 1);
- бит 7 — зарезервирован (должен быть установлен в 1).

Регистр состояния

BSY	DRDY	DF	DSC	DRQ	CORR	IDX	ERR
7	6	5	4	3	2	1	0

Регистр состояния отображает состояние устройства и доступен только для чтения. Значения бит регистра состояния (возникновение определенного состояния индицируется установкой соответствующего бита в 1) перечислены ниже:

- бит 0 (ERR) — при выполнении команды произошла ошибка (этот бит сбрасывается при поступлении следующей команды или аппаратном сбросе устройства). В случае возникновения ошибки информацию о ней можно получить из регистра ошибок;
- бит 1 (IDX) — сигнал Index, каждым производителем трактуется по-своему (поэтому при работе с диском этот сигнал нужно просто игнорировать);
- бит 2 (CORR) — при считывании с диска имела место ошибка, но данные были успешно скорректированы;
- бит 3 (DRQ) — устройство готово к обмену данными с процессором;
- бит 4 (DSC) — головки чтения-записи завершили поиск заданного сектора;
- бит 5 (DF) — устройство неисправно;
- бит 6 (DRDY) — устройство готово к приему следующей команды;
- бит 7 (BSY) — устройство занято выполнением какой-то операции, ему нельзя передавать команды или данные, нельзя считывать содержимое регистров (во избежание получения ложных данных).

Регистр команд

Регистр команд (CR) используется для загрузки кода выполняемой команды. Запись кода команды должна производиться в последнюю очередь — только после того, как в остальные регистры занесены все необходимые для ее выполнения данные. Выполнение команды начинается сразу после записи кода в этот регистр.

Альтернативный регистр состояния (AC) по формату аналогичен регистру состояния (SR), но считывание данных из него не приводит к снятию запроса прерывания.

X	X	X	X	X	SRST	nIEN	0
7	6	5	4	3	2	1	0

Регистр управления устройством

Регистр управления устройством (DC) доступен только для записи. Значения битов этого регистра следующие (рис. 6.12):

- бит 0 — не используется (всегда должен быть сброшен в 0);
- бит 1 (nIEN) — запрет прерывания (0 — прерывание разрешено, 1 — запрещено);
- бит 2 (SRST) — программный сброс всех подключенных к данному каналу устройств (сброс происходит при установке этого бита в 1);
- биты 3—7 — зарезервированы (игнорируются).

Коды обязательных команд ATA

В соответствии со стандартом команды интерфейса ATA делятся на три основные группы:

- обязательные (Mandatory) команды;
- дополнительные (Optional) команды;
- специфические команды изготовителя (Vendor specific implementation).

Как и в случае с устройствами других типов, программисты, не связанные непосредственно с изготовителями оборудования, не имеют доступа к полной документации и вынуждены ограничиваться командами первой группы, полностью определенными в стандарте.

Интерфейс ATA66 — ATA133

Интерфейс ATA33 (50 Мбайт/с) не позволял реализовать возможности жестких дисков. На смену этому интерфейсу пришел сначала интерфейс ATA66 с максимальной пропускной способностью 66 Мбайт/с, а затем и интерфейс ATA100 с пропускной способностью 100 Мбайт/с. Несколько позднее был выпущен интерфейс ATA133 с максимальной пропускной способностью 133 Мбайт/с и 48-битной адресацией сектора (что позволяло адресовать диски с невероятно большим объемом — 144 Пбайт (петабайт).

Казалось бы, стандарт ***ATA133*** решил все проблемы и можно на этом успокоиться, но, увы, параллельная передача данных имеет свои ограничения. В этом смысле скорость интерфейса в 133 Мбайт/с является предельной, а значит, и у самого интерфейса отсутствует перспектива дальнейшего развития, то есть он не имеет потенциальной возможности для масштабирования.

6.5 Последовательный интерфейс Serial ATA

Serial ATA. В первой версии стандарта Serial ATA (SATA 1.0) предусмотрена максимальная пропускная способность 150 Мбайт/с, а об ограничениях на размеры дисков можно просто забыть на ближайшие лет десять. В следующих версиях ***SATA*** предусматривается удвоение скорости передачи, то есть сначала будет 300, а затем и 600 Мбайт/с.

Как уже отмечалось, стандарт SATA подразумевает последовательную передачу данных, а потому в кабелях передачи данных используются всего две симплексные дифференциальные пары. Одна из них работает на передачу, а другая — на прием. Всего же в кабеле SATA допускается (опционально) использование семи проводников, три из которых «земля». Максимальная длина кабеля при этом составляет 1 м.

По сравнению с традиционным параллельным интерфейсом интерфейс Serial ATA имеет бóльшую помехозащищенность и мало восприимчив к электромагнитным помехам благодаря использованию низкоуровневых дифференциальных сигналов.

На физическом уровне для передачи данных используется двухэтапное логическое кодирование ***8b/10b***. При логическом ко-

дировании 8b/10b каждые 8 бит исходной последовательности заменяются на 10 бит в соответствии с определенными правилами. В результате для 256 возможных комбинаций из 8 входных бит получаем 1024 возможные комбинации для 10 выходных бит. Но разрешенными из этих 1024 комбинаций являются только 256, а остальные — запрещенными. Как правило, такая избыточность используется для того, чтобы повысить помехоустойчивость кодирования (если при приеме обнаруживается запрещенная последовательность, то распознается ошибка передачи). Кроме того, незначительная избыточность улучшает спектральные характеристики сигнала, поскольку исключает возможность появления в цепочке передаваемых бит длинных последовательностей нулей и единиц. Также повышаются и самосинхронизирующие свойства кода.

При кодировании 8 исходных бит разбиваются на две подгруппы: из 5 бит и из 3 бит. На первом этапе подгруппа 5 бит подвергается кодированию ***5b/6b***, то есть каждые 5 бит заменяются на 6. На втором этапе оставшиеся 3 бита подвергаются кодированию ***3b/4b***.

Целесообразность использования двухэтапного кодирования вызвана тем, что при кодировании каждой группы (сначала 5 бит, а потом оставшихся 3 бит) формируется специальный бинарный контрольный сигнал ***rd*** (*Running Disparity*), который может быть либо отрицательным (rd-), либо положительным (rd+).

- При кодировании 5b/6b для 32 возможных 5-битных комбинаций на входе существует 46 6-битных возможных комбинаций на выходе (не 32 и не 64, а именно 46!). Получаются эти 46 возможных комбинаций следующим образом: каждой из 32 возможных 5-битных комбинаций на входе ставится в соответствие две 6-битные выходные последовательности: прямая и инверсная, за исключением тех 6-битных последовательностей, для которых количество «1» совпадает с количеством «0» — отсюда именно 46 возможных комбинаций на выходе.
- При кодировании 3b/4b для 8 возможных 3-битных комбинаций на входе существует 14 возможных 4-битных комбинаций на выходе, которые формируются так же, как и прежде. При кодировании 3b/4b, так же как и при 5b/6b, использование прямой или инверсной выходной последовательности определяется те-

кущим значением контрольного сигнала rd. Если сигнал rd положителен, то используется прямая последовательность, а если отрицателен — то инверсная. При этом текущее значение сигнала rd определяется по предыдущей переданной последовательности из 6 или 4 бит.

• Правило для формирования сигнала rd достаточно простое. Сигнал положителен, если количество единиц больше количества нулей в группе закодированных бит. Исключение составляют последовательности с равным количество нулей и единиц. Для последовательностей 000111 (подгруппа 6 бит) и 0011 (подгруппа 4 бит) сигнал считается положительным, а для последовательностей 111000 и 1100 — отрицательным. Во всех остальных случаях сигнал rd нейтрален и не меняет своего состояния.

К примеру, если входная последовательность 5 бит 00001 подвергается кодированию 5b/6b, то при положительном текущем сигнале rd+ последовательность 00001 будет заменена на последовательность 100010. Если текущее значение сигнала rd отрицательно (rd-), то будет сформирована инверсная последовательность 011101. Но в обоих случаях последующее состояние сигнала rd изменится на противоположное, так как при rd+ в последовательности 100010 нулей больше, чем единиц, и, следовательно, сигнал rd примет отрицательное значение, а при rd- в последовательности 011101 единиц больше, чем нулей, и сигнал rd станет положительным.

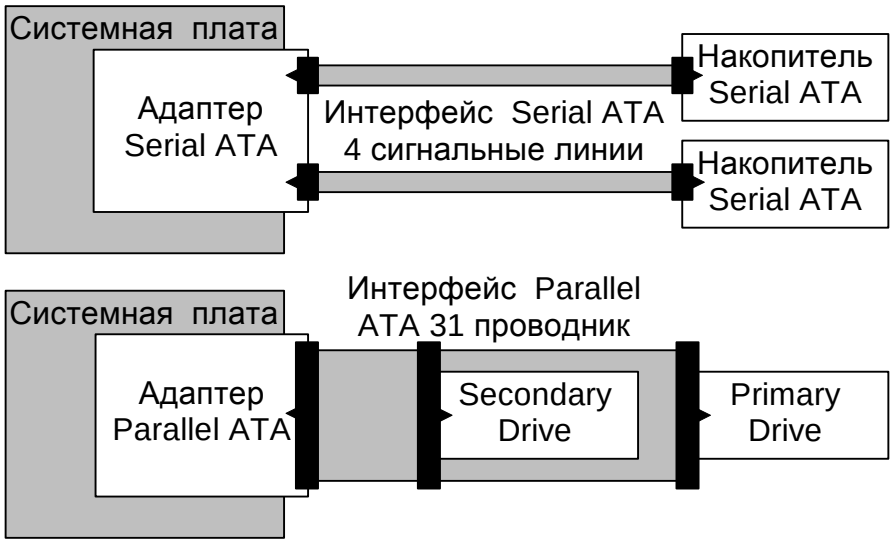


Рис. 6.4 — Параллельный и последовательный ATA-интерфейс

Аналогичному кодированию в зависимости от текущего значения сигнала rd подвергается и подгруппа из 3 бит (кодированию 3b/4b), причем с учетом того, что сперва происходит кодирование 5b/6b, а потом 3b/4b, текущее значение сигнала rd при кодировании 3b/4b определяется по результату кодирования 5b/6b.

Таким образом, в стандарте SATA предусмотрен довольно нетривиальный метод логического кодирования.

Кроме логического двухэтапного кодирования, при передаче данных используется метод циклического избыточного контроля CRC-32 (*Cyclic Redundancy Check*). На физическом уровне используется потенциальный код NRZ (*Non Return to Zero*).

Другой особенностью стандарта SATA является организация взаимодействия между контроллером и диском по принципу «точка-точка» (*peer-to-peer*): к одному контроллеру можно подключить только один жесткий диск, поэтому каждому устройству стандарта SATA предоставляется вся полоса пропускания целиком.

К тому же в стандарте SATA предусмотрена поддержка технологии «*hot swap*» (использование дисков с горячей заменой).

6.6 SATA II Phase 1

Стандарт SATA II является логическим продолжением стандарта SATA 1.0. Как правило, стандарт SATA II связывают с большей пропускной способностью интерфейса. Так, если для версии SATA 1.0 она составляет 150 Мбайт/с, то для SATA II — уже 300 Мбайт/с.

Впрочем, увеличенная в два раза полоса пропускания интерфейса в стандарте SATA II — это не единственное различие между SATA 1.0 и SATA II. На самом деле существует два этапа реализации стандарта SATA II. Первый этап реализации SATA II Phase 1 не подразумевает удвоения полосы пропускания. Речь идет лишь о расширении стандарта SATA 1.0 (SATA Extensions).

Основных нововведений в стандарте SATA II Phase 1 два: технология изменения очередности команд (Native Command Queuing) и технология умножения портов.

Технология Native Command Queuing, обеспечивающая значительный прирост производительности при операциях ввода-вывода, представляет собой особый алгоритм изменения очередности выполнения команд с целью увеличения пропускной способности. То есть при использовании технологии Native Command Queuing контроллер SATA-устройства анализирует очередь запросов от процессора и оптимизирует очередность их выполнения таким образом, чтобы максимизировать скорость передачи и минимизировать время поиска. Механизм работы технологии Native Command Queuing в общих чертах схож с организацией очередности выполнения команд в SCSI-устройствах. Однако если для SCSI-устройств контроллер поддерживает очередь глубиной в 256 команд, то для SATA-устройств обеспечивается оптимизация очереди глубиной в 32 команды.

Попытаемся на элементарном уровне объяснить, как за счет перераспределения очередности выполнения команд можно повысить производительность и сократить время доступа. Пусть, к примеру, имеется очередь из четырех команд A-B-C-D. Для выполнения этой очередности команд может потребоваться, скажем, 2,5 оборота диска. Если же выполнять те же команды в порядке B-D-A-C, то потребуется всего 1 оборот диска. Таким образом уменьшается время выполнения команд, а следовательно, увеличивается производительность выполнения операций ввода-вывода.

Другим нововведением в стандарте SATA II Phase 1 стала технология умножения портов. Напомним, что в стандарте SATA 1.0 взаимодействие между контроллером и диском организовано по принципу «точка-точка» (peer-to-peer), то есть к одному каналу контроллера можно подключить только один жесткий диск. Применение концентраторов (умножителей портов) позволяет подключить к одному каналу до 15 устройств вместо одного. Такое расширение возможностей подключения весьма существенно в задачах, где используется большое количество дисков; кроме того, оно увеличивает количество доступных пользователям портов.

6.7 SAS (Serial Attached SCSI)

На рынке серверов традиционным решением на протяжении многих лет считался параллельный интерфейс SCSI, в котором применяется низковольтная дифференциальная передача данных и который неплохо защищен от помех, а также отказоустойчив. За время своего существования этот интерфейс неоднократно совершенствовался, сохраняя при этом обратную совместимость. По сравнению с IDE-дисками для настольных ПК SCSI-диски всегда были более производительными, надежными и дорогими. В процессе эволюции SCSI-интерфейса появились стандарты Ultra160 SCSI и Ultra320 SCSI, которые обеспечивают пропускную способность 160 и 320 Мбайт/с соответственно, что вполне достаточно для современных систем хранения данных, RAID-массивов и серверных дисков.

Но… тенденция все сделать последовательным не миновала и SCSI-интерфейс. В результате появился новый стандарт Serial Attached SCSI (SAS), подразумевающий последовательную передачу данных для SCSI-дисков.

Первая спецификация Serial Attached SCSI еще пребывает в стадии разработки и утверждения (выход намечен на начало 2004 года), поэтому рановато говорить о том, что из нее получится и как она будет состязаться с Serial ATA. Хотя известно, что пропускная способность Serial Attached SCSI составит 300 и 600 Мбайт/с.

Впрочем, основное достоинство SAS-систем заключается не в пропускной способности, а в возможности подключения до 128 жестких дисков, полнодуплексном режиме работы, возможности использования кабеля длиной до 6 м (а следовательно, имеется возможность построения внешних дисковых систем с высокой плотностью монтажа).

Еще одна особенность стандарта Serial Attached SCSI — его совместимость с SATA-дисками на физическом уровне (кабели и разъемы) и на уровне команд, то есть возможность подключать SATA-диски к любому SAS-контроллеру. Правда, обратная ситуация уже невозможна, то есть к SATA-контроллеру нельзя подключить SAS-диск.

Таблица 6.3 — **В таблице дано сравнение стандартов SATA 1.0 и SAS**

Стандарт	SATA 1.0	SAS
Режим работы	Полудуплекс	Полный дуплекс
Пропускная способность канала, Мбайт/с	150	300
Длина кабеля, м	1	6
Количество устройств на канал	1	128
Совместимость	Только SATA	SAS, SATA

Для передачи и приема данных и команд используются две дифференциальные пары (так же как и в стандарте SATA 1.0), что позволяет организовать полнодуплексный режим работы (рис. 6.5). Как и в стандарте SATA 1.0, передача по дифференциальным парам происходит в противофазе, что позволяет избежать перекрестных наводок без скручивания проводов. Кроме того, используются низковольтные сигналы.

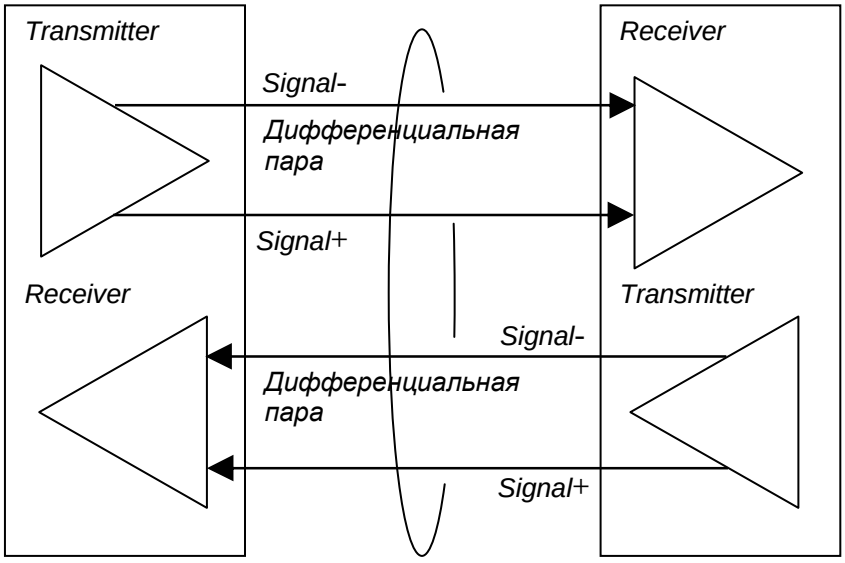


Рис. 6.5 — SATA 1.0. Полнодуплексный режим работы

Одна или несколько физических связей, каждая из которых является дифференциальной парой, образуют порт. Причем если в порте используется только одна физическая связь, то говорят об узком порте (Narrow port), а при объединении в одном порте нескольких связей получают широкий порт (Wide port) (рис. 6.6).

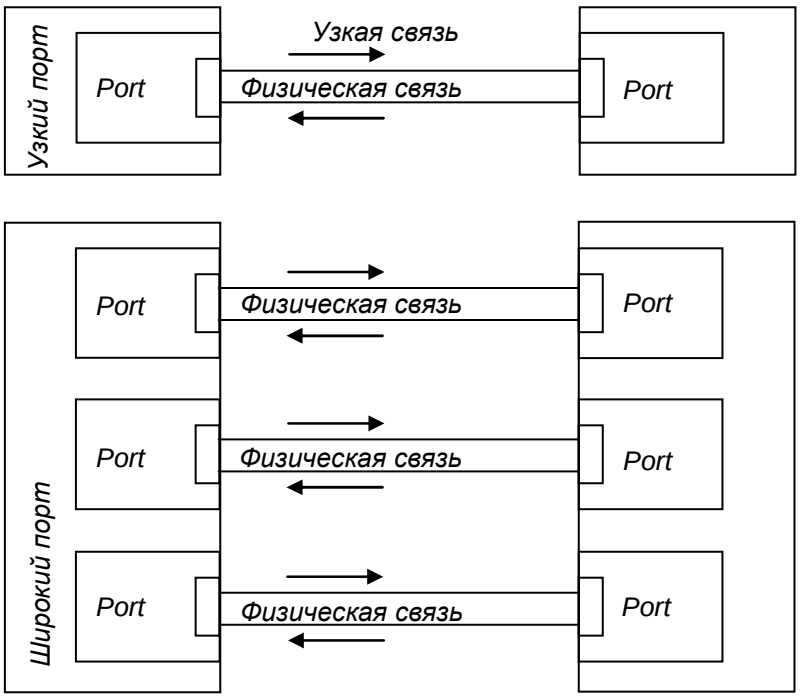


Рис. 6.6 — SATA 1.0. Широкий порт

Каждое физическое SAS-устройство должно иметь один или несколько узких или широких портов. Кроме конечных устройств (жестких дисков) и контроллеров в стандарте SAS используются расширители (Expander), которые также могут иметь как узкие, так и агрегированные широкие порты.

К расширителям подключаются как оконечные устройства, так и другие расширители. Причем используется два типа расширителей: оконечный (Edge Expander) и корневой (Fanout Expander). К оконечным расширителям подключаются только конечные SAS-устройства, например жесткие диски. Корневой расширитель предназначен для подключения к нему только оконечных расширителей и выполняет своеобразные функции маршрутизатора.

Использование корневого и оконечных расширителей позволяет создавать разветвленные топологии подключения SAS-устройств, причем всего можно подключать до 128 жестких дисков.

6.8 RAID-массивы

В современной компьютерной индустрии в качестве устройств хранения данных самое широкое распространение полу-

чили жесткие диски, поскольку, несмотря на все свои недостатки, они обладают наилучшими характеристиками для соответствующего типа устройств при доступной цене. Однако, несмотря на все усилия производителей, из-за конструктивных особенностей жестких дисков их производительность значительно отстает от производительности процессорной и других подсистем компьютера.

Невозможность значительного увеличения производительности жестких дисков заставляет искать другие пути повышения производительности системы хранения данных. Одним из таких путей является параллельная обработка данных. Если разделить блок данных на несколько частей и расположить на N дисках некоторого массива, обеспечив возможность одновременной обработки, то этот блок можно будет считать/записать в N раз быстрее (без учета времени формирования блока). К сожалению, при увеличении количества дисков в массиве его надежность уменьшается и возникает необходимость повышения отказоустойчивости.

Несмотря на то, что системы хранения данных, основанные на магнитных дисках, производятся уже более 40 лет, к разработке отказоустойчивых систем приступили совсем недавно — в 1987 году. Американские исследователи Паттерсон, Гибсон и Катц из Калифорнийского университета Беркли в своей статье «A Case for Redundant Arrays of Inexpensive Discs, RAID» («Избыточный массив недорогих дисков») описали, каким образом можно объединить несколько дешевых жестких дисков в одно логическое устройство так, чтобы в результате повышались емкость и быстродействие системы, а отказ отдельных дисков не приводил к отказу всей системы. Через некоторое время название технологии немного изменили, слово Inexpensive (недорогие) поменяли на Independent (независимые), что больше соответствовало действительности (все жесткие диски в то время были довольно дорогими устройствами) и сути технологии.

Итак, RAID — это избыточный массив независимых дисков (***Redundant Arrays of Independent Discs***), на который возлагается задача обеспечения отказоустойчивости и повышения производительности обработки данных. Повышение производительности обработки данных обеспечивается одновременной работой нескольких дисков, и в этом смысле чем больше дисков в массиве (до определенного предела), тем лучше. Одновременную работу

дисков в массиве можно организовать с использованием либо параллельного, либо независимого доступа. При параллельном доступе дисковое пространство разбивается на блоки (полоски) для записи данных.

Информация, подлежащая записи на диск, разбивается на такие же блоки. При записи отдельные блоки одновременно записываются на различные диски, что и приводит к увеличению производительности. Чтение также выполняется отдельными блоками одновременно с нескольких дисков, при этом производительность растет пропорционально количеству дисков в массиве.

Следует отметить, что модель с параллельным доступом реализуется лишь при условии, что размер запроса на обработку данных больше размера блока данных на диске. В противном случае параллельная обработка нескольких блоков становится просто невозможной.

Если размер записываемых данных меньше размера блока, то можно реализовать принципиально иную модель доступа — независимый доступ. Подобная модель может применяться и в том случае, когда размер записываемых данных больше размера одного блока. При независимом доступе все данные отдельного запроса записываются на отдельный диск, то есть ситуация идентична работе с одним диском. Преимущество модели с параллельным доступом заключается в том, что при одновременном поступлении нескольких запросов на запись (чтение) все они будут выполняться независимо, на отдельных дисках.

Отказоустойчивость массива достигается за счет избыточности информации, сохраняемой на жестких дисках, то есть часть емкости дискового пространства отводится для служебных целей, становясь недоступной для пользователя. Избыточная информация может либо размещаться на специально выделенном диске, либо распределяться между всеми дисками массива. Способов формирования избыточной информации довольно много. Простейший из них — полное дублирование (или ***зеркалирование***) — имеет 100-процентную избыточность. Для снижения избыточности (увеличения объема полезного дискового пространства) используются различные математические методы типа вычисления четности или применения кодов с коррекцией ошибок.

Сначала из-за высокой стоимости технология использовалась только в специализированных системах хранения данных и в дорогостоящих серверах масштаба предприятия. Высокая цена определялась как стоимостью контроллеров, так применением дорогих жестких дисков с SCSI-интерфейсом. По мере развития технологии и снижения стоимости всех ее компонентов RAID-массивы стали своеобразным стандартом де-факто даже для серверов начального уровня, а с появлением ***IDE RAID-контроллеров*** сфера применения RAID-массивов расширилась еще больше. RAID-массивы с использованием дешевых IDE-дисков появились на серверах начального уровня, высокопроизводительных рабочих станциях, а затем и на персональных компьютерах.

Сейчас большинство современных материнских плат для ПК снабжены возможностями построения RAID-массивов либо средствами южного моста чипсета, либо при помощи интегрированного на плате дополнительного RAID-контроллера.

В соответствии с различными типами доступа и способами формирования избыточной информации существуют и различные типы RAID-массивов, которые принято характеризовать уровнями RAID.

Уровни RAID

В настоящее время существует несколько RAID-уровней, которые можно считать стандартизованными, — это RAID 0, RAID 1, RAID 2, RAID 3, RAID 4, RAID 5 и RAID 6. Применяются также различные комбинации RAID-уровней, что позволяет объединить их достоинства. Обычно это комбинация какого-либо отказоустойчивого уровня и нулевого уровня, применяемого для повышения производительности (RAID 10, RAID 30, RAID 50).

Большинство уровней применяются в системах хранения данных и в серверах, но пока недоступны для пользователей персональных компьютеров. На персональных компьютерах сначала использовались RAID-массивы уровней RAID 0 и RAID 1, затем RAID 0+1, и, наконец, появилась возможность построения RAID-массива уровня RAID 5 (например, на интегрированном RAID-контроллере Silicon Image Sil3114). А недавно компания Intel внедрила технологию **Matrix RAID**, которая позволяет создать на двух жестких дисках одновременно RAID-массивы двух уров-

ней — RAID 0 и RAID 1, выделив для каждого из них часть дискового пространства. Такая возможность предусмотрена у многих RAID-контроллеров, используемых в серверах, но на персональных компьютерах она реализована впервые.

Напомним, что имеющаяся у многих RAID-контроллеров функция JBOD (Just a Bench Of Disks) не предназначена для создания массивов, а обеспечивает возможность подключения к RAID-контроллеру отдельных дисков.

Рассмотрим более подробно способы организации дискового массива в современных персональных компьютерах.

RAID 0

Представляет собой дисковый массив, в котором данные разбиваются на блоки (stripe) и каждый блок записывается (или же считывается) на отдельный диск. В результате обеспечивается возможность либо ускоренной обработки файлов большого размера, либо одновременного выполнения нескольких операций ввода-вывода небольших блоков информации (рис. 6.7).

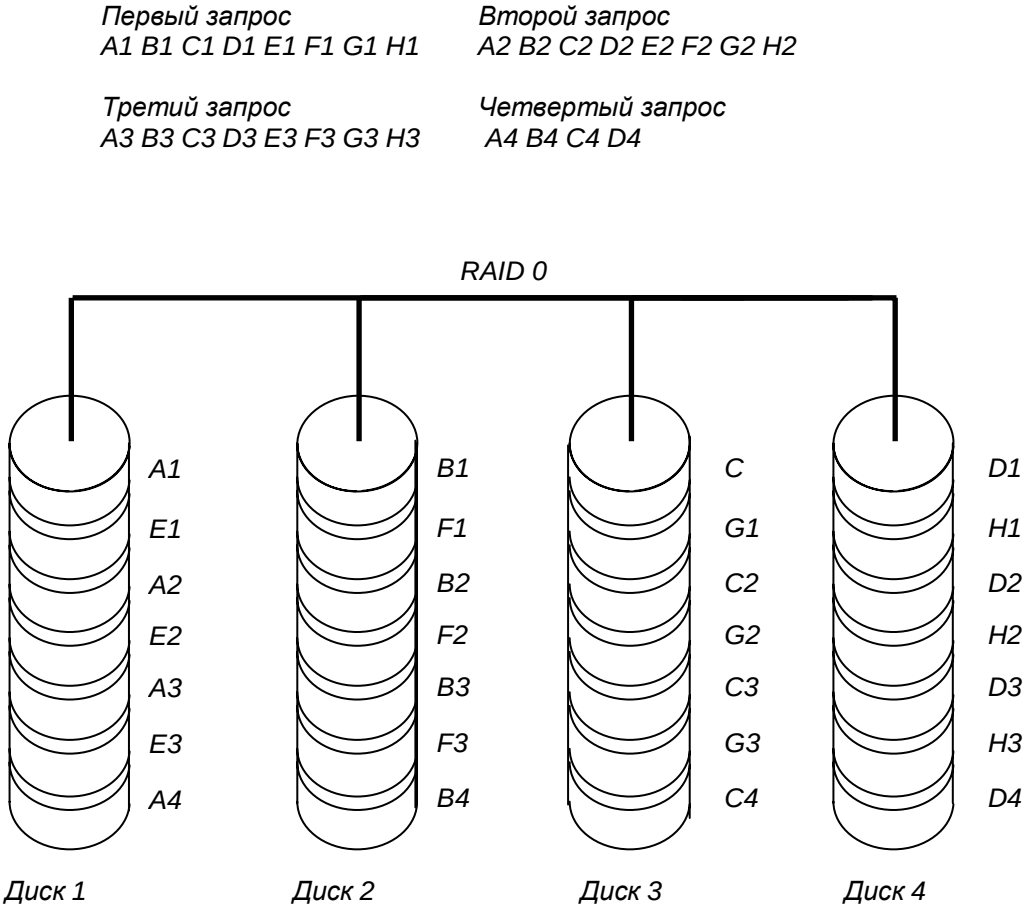


Рис. 6.7 — RAID 0

RAID 0 не является избыточным массивом, поэтому не обеспечивает надежности хранения данных — отказ одного диска влечет за собой потерю всех данных массива.

Тем не менее данный уровень находит широкое применение, что объясняется простотой реализации и максимальной эффективностью использования дискового пространства, поскольку не требуется места для хранения контрольных сумм, и, следовательно, низкой стоимостью на единицу объема.

RAID 0 применяется в тех случаях, когда необходимо обеспечить высокую производительность дисковой подсистемы, — например для размещения рабочих файлов при обработке изображений в Adobe Photoshop, обработке видео в Adobe Premiere, создании CD- или DVD-образов и при решении других подобных задач.

RAID 1

RAID 1 — это массив дисков со 100-процентной избыточностью, обладающий очень высоким уровнем надежности хранения данных. Запись выполняется сразу на два диска, при этом данные полностью дублируются (зеркалируются) и два диска содержат одинаковую информацию (рис. 6.8), что определяет основные недостатки массива — высокую стоимость хранения и невысокую скорость записи данных, равную скорости записи на одиночный диск. При выходе из строя одного из дисков его функции выполняет другой; восстановление массива выполняется простым копированием.

К достоинствам массива можно отнести простоту реализации и возможность увеличения скорости чтения информации, поскольку эта операция может выполняться одновременно с двух дисков (так могут работать не все RAID-контроллеры).

Такая схема хранения информации используется в основном в тех случаях, когда цена безопасности данных значительно превышает стоимость реализации системы хранения. В домашних условиях RAID-массив уровня 1 можно применить для хранения ценных фотографий, документов и финансовых записей.

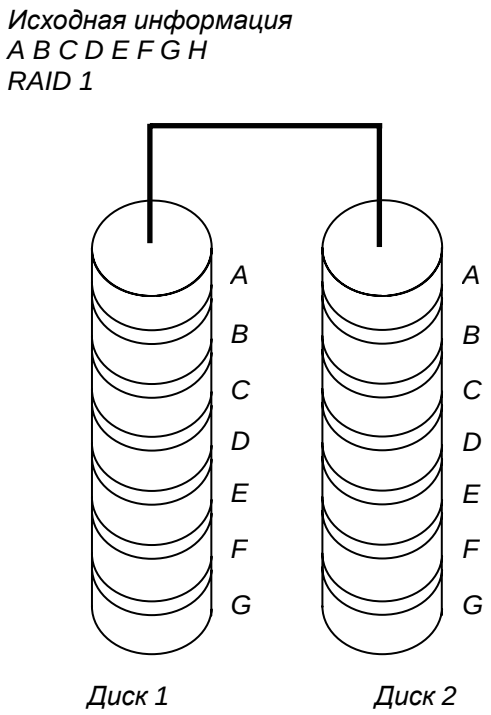


Рис. 6.8 — RAID 1

RAID 5

RAID 5 — это отказоустойчивый дисковый массив, данные в котором восстанавливаются при отказе одного из жестких дисков. Перед записью данные разбиваются на блоки и для блоков данных одного уровня рассчитывается контрольная сумма. Блоки данных и контрольные суммы циклически записываются на все диски массива, следовательно, отсутствует выделенный диск для хранения информации о контрольных суммах (рис. 6.9). При чтении четность блоков проверяется.

RAID 5 может быть построен на трех и более жестких дисках. С увеличением количества жестких дисков в массиве его избыточность уменьшается (1/2 при трех дисках, 1/3 — при четырех, 1/4 — при пяти и т.д.). Общая емкость дисковой подсистемы, доступной для записи, становится меньше ровно на один диск, при этом все диски массива должны быть одного размера. Например, если четыре диска имеют размер 100 Гбайт, то доступный для записи размер массива составляет 300 Гбайт, так как 100 Гбайт будет занято контрольной информацией. При создании и восстановлении массива в случае отказа жесткого диска выполняется большое количество вычислений, требующих наличия

специализированного процессора, поэтому применение RAID 5 до недавнего времени ограничивалось только серверами.

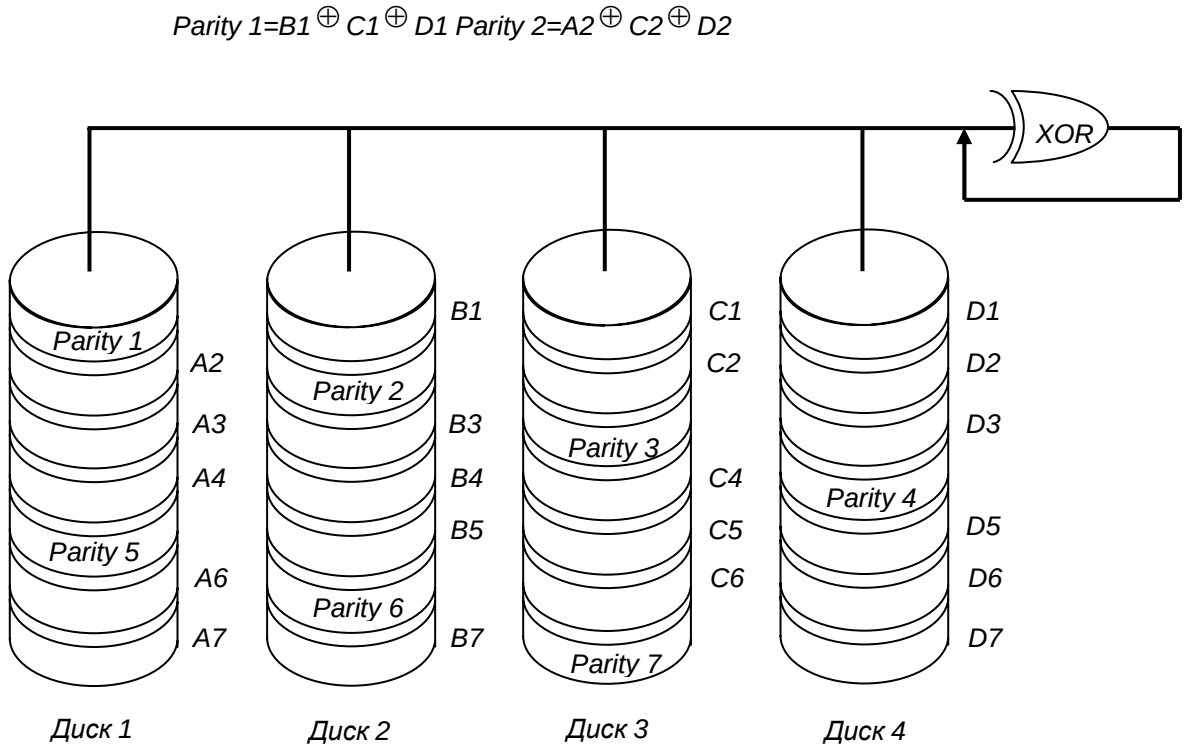


Рис. 6.9 — RAID 5

RAID 5 имеет архитектуру независимого доступа, что обеспечивает возможность одновременного выполнения нескольких операций считывания или записи. Прирост производительности обеспечивается как при одновременной обработке нескольких коротких запросов, так и при обработке больших блоков данных. При обработке единичных запросов чтения/записи данных малого объема производительность невысокая.

RAID 5 — наиболее универсальный уровень RAID-массива. Он успешно применяется на серверах различного назначения — сервер приложений, файловый сервер, почтовый сервер и др.

Если вы планируете использование дисковой подсистемы большой емкости, то стоит подумать о применении RAID 5, ведь накладные расходы 25 % (при четырех дисках) — не слишком высокая плата за гарантию сохранности данных.

Intel Matrix RAID

Как бы ни снижались цены на комплектующие, все равно применение RAID-массивов будет связано с дополнительными затратами, которые по плечу не каждому пользователю. Технология Intel Matrix RAID делает использование RAID-массивов более доступным. С ее помощью всего на двух жестких дисках можно построить два RAID-массива, образующих два логических диска, получив при этом и высокую производительность, свойственную RAID 0, и высокую надежность хранения данных, характерную для RAID 1 (рис. 6.10).

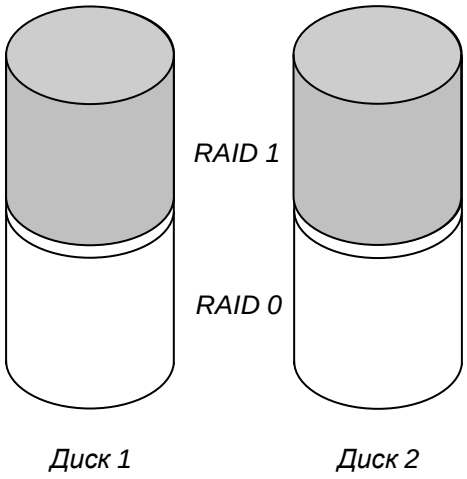


Рис. 6.10 — RAID массив

RAID-массивы занимают на дисках выделенное для них пространство и работают независимо. Соотношение объемов, занимаемых массивами на жестких дисках, можно выбрать в соответствии с потребностями пользователя. Данные, размещенные на разделе RAID 1, сохраняются при отказе одного жесткого диска.

Технология Intel Matrix RAID работает только на чипсетах Intel последних моделей. Для ее работы необходимо, чтобы на материнской плате был установлен южный мост 82801ER (ICH5R) или 82801FR (ICH6R) и выше. Технология работает только под управлением операционных систем Microsoft Windows 2000, Microsoft Windows XP Home Edition и Microsoft Windows XP Professional, а кроме того, обязательно нужно установить Intel Application Accelerator RAID Edition версии 4.0 или выше.

7 ВИДЕОСИСТЕМА ПК

7.1 Видеокарты

Видеокарта напоминает модель всего компьютера: у неё так же есть процессор, память, шины передачи данных и даже BIOS.

Ядром видеокарты является чип — ***GPU — Graphical Processor Unit*** (графический процессор) или ***VPU — Visual Processor Unit*** (визуальный процессор) он помещается под радиатором (на более старых образцах радиаторов не было). Вокруг расположены микросхемы памяти в корпусах PGA (старого образца) или новые — в BGA исполнении. Ещё на плате имеется ряд дополнительных микросхем (они будут рассмотрены отдельно). Естественно, на видеокарте есть разъёмы. Два из них присутствуют на карте обязательно — это AGP или PCI, и, собственно, вывод на монитор. Ещё могут быть (а могут и не быть) дополнительные разъёмы типа видеовыхода или разъема для второго (и даже третьего) монитора. Могут также присутствовать всяческие опции вроде ТВ-тюнера со своими антеннами или, контроллера шины IEEE 1394. BIOS на современных видеокартах также может перепрошиваться, это полезное свойство для повышения скорости и качества реализации отдельных функций.

7.2 Видеопроцессор

Видеопроцессорами во всём мире занимаются порядка десяти фирм.

NVIDIA. Эта фирма занимает значительный сегмент рынка как сверхмощных, так и бюджетных решений. Очень высокая скорость разработки видеокарт. Фирма представлена во всех сегментах рынка. Фирма не собирает карты сама, а предпочитает отдавать чипы сторонним сборщикам, которые и лепят из них готовые карты. Это позволяет фирме целиком сконцентрироваться на производстве могучих чипов под общим именем GeForce (Riva128, TNT и TNT2).

ATi — фирма, которая занимает оставшуюся долю на рынке. В отличие от NVIDIA, ATi имеет значительные договоры с поставщиками (сборщиками) компьютеров. Первоначально ATi

собирала карты самостоятельно и отвечала за качество перед OEM-партнёрами, однако, в последнее время, также выпускает только чипы. Текущий модельный ряд состоит из карт семейства Radeon.

Intel. Компания встраивает видеокарты в собственные чипсеты для своих же процессоров. Благодаря громадному объему выпуска чипсетов и процессоров, доля встроенных графических решений достаточно велика, особенно в офисных ПК.

Matrox. Основной рынок для фирмы — профессиональная графика и видеомонтаж. Отличается качественным видеовыходом. Крайняя бытовая модель — Matrox Parhelia.

SiS (Silicon Integrated Systems). Производит, в основном, чипсеты, но занимается и видеокартами. SiS Xabre.

VIA. Фирма изначально занималась чипсетами, скупила фирмы Cyrix, Centaur tech., S3 и т.п. и в наследство получила наработки в области видеоадаптеров. Выпускается семейство под названием Savage.

Imagination tech. Фирма занимается только разработками чипов, которые за неё изготавливает фирма **STMicroelectronics**, а собирает из них видеокарты Kyro в основном фирма Hercules.

7.2.1 Возможности видеопроцессора

GPU состоит из двух частей — одна занимается непосредственно построением картинки (конвейер растеризации), вторая же занимается подготовкой изображения перед закраской (конвейер трансформации и освещения, T&L).

Все современные видеокарты используют метод построения изображения, основанный на заполнении сцены треугольниками. Программа передаёт видеокарте две вещи — координаты вершин треугольника относительно камеры (точки, где находятся глаза наблюдателя в 3D-мире) и текстуру (рисунок), которую надо на него наложить. Видеокарта накладывает текстуру на треугольник, помещает треугольник на экране в надлежащее место, отображает видимую его часть. Множество треугольников образует полигоны (заведомо плоские многоугольники), из которых и строятся трёхмерные объекты.

Немного о терминологии: точка на экране называется пиксель (**pixel**) а на текстуре — тексель (**texel**, от texture pixel). В сущности, видеокарта сопоставляет пикселю тексель. Выполняет эту операцию конвейер растеризации — он определяет цвет пикселя, исходя из цвета текселя, а так же из условий видимости. Последнее означает, что тексель не будет отображаться, если он находится позади текселя уже нарисованного. Для определения, кто из них (текселей) ближе к камере, используется **Z-буфер** — массив, равный по размерам разрешению экрана, содержащий одно значение в каждой точке — глубину текущего текселя в каждом пикселе. Операция сравнения нового текселя со значением в Z-буфере называется Z-буферинг и занимает очень много времени.

Если на один тексель нужно наложить не одну, а несколько текстур, например, обычную текстуру и текстуру освещённости, то придется передать видеопроцессору один треугольник два раза. Технология под названием **TwinTexel** (видеокарты серии TNT) позволяет при обработке одного текселя обработать также второй, в тот же момент времени, ведь глубина и первого и второго текселя одинакова, и пиксель они займут один и тот же. Поэтому видеокарты TNT брали два текселя за раз, ATi Radeon берут уже по три, а Matrox Parhelia — четыре текселя.

Если пиксель получается больше текселя, применяют метод, называемый мип-мэппинг (**mip-mapping**). Он заключается в том, что текстура заранее просчитывается в разрешении, меньшем чем исходное, и используется на далёких треугольниках. Наоборот, если тексель оказывается больше пикселя, мы получим на экране квадраты. Чтобы этого не происходило, квадраты фильтруют, т.е. рассчитывают цвет пикселя в зависимости от положения между текселями. Фильтрация классифицируется по типу аппроксимирующей функции — **билинейная**, **трилинейная** и **анизотропная**. Под анизотропной понимается фильтрация, учитывающая направление наклона текстуры по отношению к камере.

Современные видеокарты способны за один такт наложить несколько текстур на несколько пикселей — это достигается за счёт нескольких конвейеров растеризации, работающих параллельно. Таким образом, скорость заполнения (количество текселей, обрабатываемых в секунду) видеокарт рассчитывается сле-

дующим образом: частота видеочипа, помноженная на число конвейеров. Соответственно для мультитекстурирования учитывается число пиксельных блоков. Так для GeForce4 с тактовой частотой 300 МГц имеем: 300МГц*4=1200 млн текселей для одной текстуры, 300*4*2/2=1200 млн текселей для двух текстур, 300*4*2/3=710 млн текселей для трёх. И так далее.

Но скорость заполнения ещё не всё. Производительность и возможности чипа определяются также конвейером трансформации и освещения.

7.2.2 Трансформация и освещение

Метод, получивший длинное название **Environment Mapping Bump Mapping**, или сокращённо **EMBM**, состоит из двух частей — Environment Mapping (EM, карты окружения) и Bump Mapping (BM, карты выпуклостей) соответственно. EM заключается в том, что на тексель, кроме стандартной текстуры, накладывается ещё одна, взятая откуда-то с другого треугольника. Таким методом формируются отражения. BM же представляет собой хитрый метод повышения детализации картинки. К стандартной текстуре прибавляется т.н. текстура глубины. В соответствии с тем, как освещён треугольник, выпуклости преобразуются в более светлые, а впадины в более тёмные участки. При этом треугольник остаётся плоским, а нам кажется, что он весь испещрен впадинками и холмиками. Проблема заключается в том, что для полноценной реализации метода необходимо накладывать три текстуры на такт (основа, отражения и освещения).

Развитием EMBM стала технология **T&L**, **Transforming & Lighting**, трансформация и освещение — впервые появилась в GeForce256. Освещение: блок T&L получает от программы вместе с данными о треугольниках, данные об источниках света. И далее сам рассчитывает освещение каждого треугольника, учитывая игру света и тени (тени он тоже сам рассчитывает). Под трансформацией понимается изменение (морфинг) вершин треугольников: блок T&L, получив координаты треугольников и координаты, куда эти треугольники поместить, выполнял перемещение сам. После введения в обиход блока T&L, чипы на картах стали гордо именоваться GPU.

Шейдеры — это мощнейшее средство для повышения реалистичности изображения — оно состоит в том, что программист может самостоятельно определить (на языке, похожем на ассемблер), как освещать треугольник. Шейдеры бывают двух видов — **вершинные** и **пиксельные**: вершинные определяют освещенность точки на треугольнике, в зависимости от положения её относительно вершин; пиксельные же позволяют управлять освещением каждого пикселя по отдельности. Вершинные (но не пиксельные) шейдеры могут эмулироваться программно. Пиксельные сами могут эмулировать бэмп-мэппинг. С появлением шейдеров GPU чаще стали называть VPU.

7.3 Видеопамять

Видеокарта потребляет следующую информацию:

- текстуры;
- координаты треугольников;
- координаты, интенсивность и цвета источников освещения;
- буфер кадра.

Буфер кадра подразумевает область ОЗУ (или собственной памяти видеокарты), куда GPU будет складывать результат своей деятельности, и откуда потом изображение пойдёт на монитор. Объём этой области пропорционален размеру кадра — разрешению экрана. Если система работает в разрешении 800*600 с 32-битным цветом, то нам потребуется 800*600*32=15360000 бит или примерно 1,8 Мбайт ОЗУ. Чем больше разрешение, тем больше буфер. Кроме того, буфер нужен двойного размера — один для отображения, а второй — для отрисовки.

7.3.1 Микросхема памяти

Основной характеристикой видеопамяти является её производительность — пропускная способность. Для более пристального её рассмотрения, разберём принцип работы памяти. Память для видеоадаптеров на сегодняшний день применяется только одного типа — **DRAM** (**Dynamic Random Access Memory**, динамическая оперативная память).

Но современные типы памяти, даже обладая гигантскими пропускными способностями, не способны удовлетворить растущие потребности GPU.

Hyper-Z. Технология — **Hierarchical Z** (иерархический Z) — заключается в том, что сцена, а точнее Z-буфер, разбиваются на небольшие участки, которые целиком умещаются в быстрой памяти самого GPU, и далее все операции проводятся в этом буфере с огромной скоростью. Видеочип сначала производит грубую оценку: рассматривать ли ему эту область или там один треугольник перекрывает всё остальное. Если требуется рассмотреть, то сцена просчитывается точнее, потом ещё точнее. Всё это позволяет существенно уменьшить время выполнения Z-буферинга. Ещё одна технология под названием **Fast Z Clear** (быстрая очистка Z) — раньше видеопроцессору приходилось после отрисовки кадра забивать область Z-буфера единицами. Новая технология позволяет делать это быстрее. И последний компонент — **Z-compression** (Z-сжатие). Позволяет сжимать данные в Z-буфере, что также уменьшает объём, который надо прокачать по шине данных.

Технология **Crossbar Memory Controller** (поперечный контроллер памяти), и заключается в том, что к памяти обращается не один контроллер шириной 128 бит (по ширине памяти), а сразу четыре шириной 32 бита. Поскольку видеопроцессору часто требуются данные из разных мест, такая организация памяти гораздо лучше традиционной. Также эта карта оснащена всем арсеналом средств по работе с Z-буфером (HyperZ), кроме того, присутствует сжатие и предварительный отброс невидимых пикселей.

Говоря об уменьшении нагрузки на память, нельзя не отметить одну очень сильную технологию — **тайловую**. Это целая философия построения видеопроцессоров: вся сцена разбивается на маленькие участки, которые обрабатываются видеопроцессором в локальной памяти. В результате значительно сокращается нагрузка на видеопамять. Проблема всех тайловых ускорителей в том, что большинство программ ожидает передачи всех треугольников и полигонов перед началом работы, в результате на большинстве программ тайловый чип простаивает.

7.4 Видеовход

Основной разъём для видеокарты — тот, которым она подключается к компьютеру. На современных системах это **AGP** (**Accelerated Graphic Port**, ускоренный графический порт). Преимущества этого разъёма перед классическим PCI огромно. Они проявляются:

- в высокой скорости передачи данных;
- низкой латентности;
- отсутствию арбитров шины;
- возможностью резервирования части оперативной памяти и прямого доступа к ней.

AGP рассчитан на подключение одного устройства — видеоадаптера. Ещё один разъём, который обязательно есть на карте — разъем для подключения монитора.

7.5 Передача видеосигнала

Рассмотрим, как формируется собственно изображение. В буфере кадра оно хранится в виде матрицы размером n*m, где n и m — размеры кадра (разрешение). Существует ряд стандартных значений для разрешений — 640*480, 800*600, 1024*768, 1152*864, 1280*1024, 1600*1200, 2048*1536 и некоторые другие. На каждую точку на экране отводится некоторое число бит для хранения информации о цвете. Стандартно 8, 16, 24, 32 бита. Число бит определяет число цветов — 2^n, где n — число бит. Однако обычный монитор на электронно-лучевой трубке — устройство аналоговое. И взять картинку напрямую из памяти он не может. Эту операцию выполняет **RAMDAC** (**Random Access Memory Digital Analog Converter**, цифро-аналоговый преобразователь оперативного запоминающего устройства или ОЗУЦАП по-русски). Это устройство получает из памяти (при этом память блокируется для видеопроцессора и AGP) буфер кадра и преобразует набор цифр в аналоговый сигнал, понятный для монитора. Основной параметр — частота, она определяется следующим образом: число точек по горизонтали умноженное на число точек по вертикали и умноженное на величину кадровой развёртки в герцах. Для хороших мониторов это: 2048*1536*85 = 267386880,

или 267 МГц. Современные видеокарты позволяют работать и на 2048*1536*120 = 377487360 или 380 МГц. Нормальному же монитору эти прелести не нужны, но чем больше максимальная частота RAMDAC, тем качественней картинка на меньших частотах.

Сигнал сформирован. На монитор он может попасть двумя способами: аналоговым или цифровым.

Разъём D-Sub представляет собой 15-контактную маленькую колодку. По кабелю по отдельным проводам передаются сигналы трёх цветов и синхронизации.На мониторе эти сигналы управляют соответственно электронными пушками трёх цветов и блоком развёртки. Недостаток такого подключения в том, что сигналы идут в одном кабеле и наводятся друг не друга. Второй вариант аналогового метода — т.н. **BNC разъём**. Он, в отличие от предыдущего, предполагает передачу трёх цветовых сигналов и одного сигнала синхронизации по четырём отдельным проводам. Такое подключение значительно сложнее, однако позволяет передать более чёткое изображение, поэтому применяется в основном в профессиональных моделях мониторов.

После появления жидкокристаллических мониторов, устройств по сути цифровых, отношение к стандартному способу подключения изменилось. Аналоговый интерфейс между памятью видеокарты и матрицей монитора выглядит плохо. Поэтому фирмы тут же стали предлагать свои решения по части цифровых интерфейсов. После долгих трений между интерфейсами PnD, DFP, PanelLink и некоторых ещё, производители объединились и назначили спецификацию нового интерфейса — **DVI** (**Digital Visual Interface**, цифровой визуальный интерфейс). **Разъём DVI** хорош также тем, что представляет из себя как бы «два в одном», т.е. позволяет передать как цифровой, так и классический аналоговый сигнал. Аналоговая часть, называемая **DVI-A**, выдаёт обычный сигнал, как D-Sub, что позволяет через переходник подключить нормальный монитор. Цифровая часть (**DVI-D**) передаёт по 24 контактам информацию о цвете в цифровом виде (по четыре провода на цвет + заземление и питание). Разъём, в котором присутствуют и DVI-A и DVI-D, называется **DVI-I**. Современные видеокарты выпускаются с установленными разъёмами DVI-I, поскольку их легко превратить в D-Sub.

На видеокарте может стоять не один, а два RAMDAC. Соответственно, изображение будет выводиться на два монитора.

Видеовыход. Изначально бытовые компьютеры подключались к телевизорам. Так было дешевле. Однако качества изображения явно не хватало, и компьютеры стали пользоваться собственными, специализированными телевизорами — видеомониторами. Однако в последнее время, в связи с тем, что на компьютере стало возможно проигрывание фильмов в очень высоком качестве, многие пользователи захотели смотреть их на экране телевизора. Желание было удовлетворено, и видеовыходы появились. С появлением второго RAMDAC стало возможным выводить фильм на телевизор, а всё остальное — на монитор. Кроме того, специально для видеовыходов могут ставиться дополнительные микросхемы.

7.6 Видеоформаты

PAL Телевизионный кадр стандарта PAL содержит 576 активных строк (всего их 625, но часть из них служебные), при этом каждая строка содержит 720 независимых отсчетов (а предельно достижимое разрешение ограничено 700 линиями). Таким образом, телевизионный кадр представляет собой матрицу **720x576**. Каждый кадр состоит из полей (field) — четных и нечетных строк (черезстрочная развертка), чтобы уменьшить мерцание, вызываемое быстрым потемнением фосфорного покрытия телеэкрана. Нечетные строки формируют нечетное поле, четные строки — четное поле. Видеосигнал может записываться с полной разверткой (25 кадров / 50 полей) и частичной разверткой (25 кадров / 25 полей).

MPEG Форматы, обеспечивающие компрессию с потерей качества и не имеющие настройки в процентах качества, а только по потоку данных в секунду. В 1988 году организована Moving Pictures Expert Group, состоящей из 25 экспертов. Теперь в нее входят 350 специалистов из 200 компаний и учреждений в 20 странах.

MPEG-1 появился в 1992 году, когда организация MPEG провела совместное совещание с Международной организацией по стандартизации (ISO) и Международной комиссией по элек-

тротехнике (IEC). Этот стандарт определяет три уровня кодирования аудио, причем третий уровень соответствует наилучшему качеству компрессии.

MPEG-2 В 1994 году организации — MPEG, ISO и IEC объявили о появлении нового стандарта компрессии мультимедиа — MPEG-2. Он предназначен главным образом для видео и содержит ряд новых функций, таких как поддержка видеосигналов с чередованием кадров. Аудиоформат, используемый в MPEG-2, отталкивается от MPEG-1. MPEG-2 сохраняет прямую и обратную совместимость с многоканальным аудио, так что он поддерживает и старую аудиотехнологию, такую как MPEG-1, и новые цифровые аудиоформаты, такие как Dolby 5.1. В MPEG-2 добавлена новая для стандарта MPEG функция: способность кодирования файлов с пониженными значениями частоты дискретизации: 16; 22,05 и 24 КГц. Это позволило повысить эффективность кодирования за счет более низкой разрядности данных. Иными словами, файлы получаются меньшего размера без заметной потери качества. Первично разрабатывался для цифровой передачи видео вещательного качества. Используется в DVD, цифровом TV и HDTV.

MPEG-3 Предназначался для телевидения высокой чёткости (HDTV), но позже стал частью стандарта MPEG-2 и отдельно теперь не упоминается. MP3. Полное название — MPEG Audio Layer 3. Предназначен только для сжатия аудио, используя компрессию с потерями (из исходного материала удаляется информация, слабо воспринимаемая человеческим ухом). Считается лучшим по соотношению качество / объём. На самом деле не является третей ступенью официального стандарта MPEG. Тройка означает третий уровень аудиокодирования — MPEG Layer III. Первоначально предполагалось, что MPEG-3 будет создаваться для HDTV, но так как в MPEG-2 нет встроенных инструментов для работы с HDTV, пока разрабатывался стандарт MPEG-3, многие средства для HDTV пришлось добавить в MPEG-2. А MP3, то есть MPEG Audio Layer III, является частью MPEG-2.

MPEG-4 Стандарт рассчитан на очень низкие потоки данных для применения в видеотелефонах, мультимедийной электронной почте, электронных информационных изданиях и т.п. MPEG-4, как MIDI, позволяет не просто воспроизводить, а синте-

зировать музыку. Но в отличие от MIDI звуки в программе MPEG-4 — не простые образцы (samples). Этот способ синтеза музыки получил название <метода Колмогорова>. Кроме того, MPEG-4 будет сочетать два языка программирования, используемых в цифровом аудио. Один из них, SAOL, применяется для обычного компьютерного аудио, а другой, SASIL, поддерживает MIDI. В своей простейшей форме MPEG-4 генерирует звук так же, как образуются файлы *.WAV, но файл MPEG-4 будет гораздо меньше.

Организация MPEG обсуждает уже и формат MPEG-7, который, похоже, снова полностью изменит подход к кодированию крупных аудио- и видеофайлов. Но MPEG-7 будет применяться не для создания файлов аудио и видео, а для их поиска. Организация надеется предложить пользователям Интернета быстрый и эффективный способ находить мультимедийные файлы.

В MPEG используется поточное сжатие видео, т. е. обрабатывается не каждый кадр по отдельности (как это происходит при сжатии видео с помощью алгоритмов Motion-JPEG), а анализируется динамика изменений видеофрагментов и устраняются избыточные данные, т. к. в большинстве фрагментов фон остается достаточно сжатие с создания **исходного (ключевого) кадра**, называемого **«I»** или **«Intra»** (И) кадр. И-кадры играют роль опорных при восстановлении остальных изображений и размещаются последовательно через каждые 10—15 кадров. Только некоторые фрагменты изображений, которые находятся между И-кадрами претерпевают изменения, и именно эта разница сохраняется при сжатии. Кроме И-кадров в MPEG-последовательности имеется еще два типа изображений:

Predicted (P) — предсказуемые (П) кадры, содержащие разность текущего изображения с предыдущим И или П с учетом смещений отдельных фрагментов.

Bi-directional Interpolated (B) — двунаправленные (Д) кадры, содержащие только отсылки к предыдущим или последующим изображениям типа И или П с учетом смещений отдельных фрагментов.

И-кадры составляют основу MPEG файла и через них осуществляется случайный доступ к какому-либо отрывку видео, но при этом у них довольно низкий коэффициент сжатия. П-кадры

кодируются относительно предыдущих кадров (И или П), и, обычно, используются как сравнительный образец для дальнейшей последовательности П-кадров. В этом случае достигается высокий коэффициент сжатия. Д-кадры обеспечивают наибольший коэффициент сжатия, но при этом для их привязки к видеопоследовательности необходимо использовать не только предыдущее, но и последующее изображение. Сами Д-кадры никогда не используются для сравнения. Изображения объединяются в группы (GOP — Group Of Pictures), представляющие собой минимальный повторяемый набор последовательных изображений. Типичной является группа вида:

```
(И0 Д1 Д2 П3 Д4 Д5 П6 Д7 Д8 П9 Д10 Д11)
   (И12 Д13 Д14 П15 Д16 Д17 П18...)
```

Отдельные изображения состоят из структурных единиц — макроблоков, соответствующих участку изображения размером 16×16 пикселов. Компьютер анализирует изображение и ищет идентичные или практически похожие макроблоки, сравнивая базовый и последующие кадры. В результате сохраняются только данные о различиях между кадрами, называемые вектором смещения (vector movement code).

Макроблоки, которые не претерпевают изменений игнорируются, и количество данных для реального сжатия и хранения существенно снижается. Для повышения устойчивости процесса восстановления изображений к возможным ошибкам передачи данных последовательные макроблоки объединяют в независимые друг от друга разделы (slices). В свою очередь каждый макроблок состоит из шести блоков, четыре из которых несут информацию о яркости Y, а по одному определяют цветовые U- и V-компоненты. Блоки являются базовыми структурными единицами, над которыми осуществляются основные операции кодирования, в том числе выполняется дискретное косинусное преобразование (DCT — discrete cosine transform). В результате при использовании MPEG можно добиться рабочего коэффициента сжатия более чем 200:1, хотя это приводит к некоторой потери качества.

Video CD (VCD) Стандарт записи видео в формате MPEG-1 на обычный Compact Disk (диаметр 120 мм, толщина 1.2 мм, одна

информационная сторона). Один диск обычно позволяет хранить до 74 минут видео, качество соизмеримо с VHS стандартом. Для воспроизведения достаточно односкоростного CD-ROM.

DVD (Digital Versatile Disk) Формат DVD-диска принят 8 декабря 1995 года. Первоначально аббревиатура DVD расшифровывалась, как Digital Video Disc (цифровой видеодиск), несколько позже появилась расшифровка аббревиатуры DVD, как Digital Versatile Disc (универсальный цифровой диск).

SIF — термин, описывающий компьютерное разрешение 352x288 (PAL), 352x240 (NTSC), соответствующее разрешению VHS.

CCIR-601 — стандарт, описывающий формат цифрового видео с разрешением 720x576 (PAL) и 720x480 (NTSC).

Field — Поле. Отдельное изображение в составе видеопотока. Каждый кадр состоит их четных и нечетных строк (черезстрочная развертка), чтобы уменьшить мерцание, вызываемое быстрым потемнением фосфорного покрытия телеэкрана. Нечетные строки формируют нечетное поле, четные строки — четное поле. Видеосигнал может записываться с полной разверткой (25 кадров / 50 полей) и частичной разверткой (25 кадров / 25 полей).

AVI — Audio Video Interleaved, оригинальная аббревиатура для Microsoft Video For Windows.

7.7 DirectX

DirectX представляет собой набор технологий и инструментов, которые позволяют создавать разработчику игры и мультимедиа приожения с неслыханным во времена MS-DOS качеством графики и звука. Кроме этого, DirectX служит для обработки клавиатуры, мыши, джойстика, а также для сетевого сообщения.

DirectX подразделяется на несколько частей, каждая из которых отвечает за что-то свое:

- **DirectDraw** — служит для ускорения отображения и обработки двумерной графики.
- **Direct3D** — для ускорения трехмерной графики.
- **DirectSound** — работает со звуком — микширование и 3D звук.

- **DirectInput** — для обработки клавиатуры, мыши, джойстика и так далее.
- **DirectPlay** — служит в основном для сетевой игры.
- **DirectAnimation** — для создания анимационных эффектов в WEB-стриницах.
- **DirectShow** — для применения мультимедиа в WEB.
- **DirectMusic** — новый раздел. Служит для применения музыки в играх.

DirectX разрабатывался специально, чтобы превратить платформу Windows как в основную для разработки игр. До этого разработчики использовали только MS-DOS и лишь совсем незначительная часть игр делалась для Windows 3.xx. Одной из более ранних попыток Microsoft был выпуск **WinG,** который позволял разработчикам не писать бесконечные поддержки для различных типов аудио-видеоадаптеров, однако появление DirectX полностью изменило дело в пользу Windows. Теперь, разработчики могли почти не отвлекаться на подержки различных карт, потому что если у карты была поддержка DirectX, то несовместимость больше не была проблемой.

DirectX — это интефейс довольно низкого уровня. С помощью своих API он предоставляет программисту прямой доступ к памяти адаптеров, где программист может создавать изображение, хранить графические образы, звуки и т.д. За счет непосредственной работы с памятью достигается ускорение, то есть теоретически частота, с которой программист сможет заставить прорисовываться экран будет зависеть только от частоты, поддерживаемой монитором.

7.8 Программирование видеосистемы

В современных вычислительных системах реализовано раздельное обращение программных продуктов к аппаратным устройствам компьютера, т. е. практически никакие программы, реализуемые на компьютере, непосредственно к устройствам не обращаются. С контроллерами взаимодействуют лишь программы драйверов соответствующих устройств. Под понятием ***драйвер*** (***driver***) обычно понимается программный модуль, который:

- непосредственно управляет некоторым внешним устройством, взаимодействуя с его контроллером с помощью команд ввода-вывода компьютера;
- обрабатывает прерывания от контроллера устройства;
- предоставляет прикладному программисту естественный интерфейс работы с устройством, экранируя от него низкоуровневые детали управления устройством и организации его данных;
- взаимодействует с ядром ОС с помощью строго оговоренного интерфейса, описывающего формат передаваемых данных, способы включения драйвера в состав ОС, способы вызова драйвера, набор общих процедур подсистемы ввода-вывода, которыми драйвер может пользоваться, и т.д.

Таким образом в современных вычислительных системах реализуется идея многослойного подхода к организации программного обеспечения. Контроллер представляет собой нижний слой управления устройством, выполняющий операции в терминах блоков и агрегатов устройства. Драйвер выполняет более сложные операции по организации взаимодействия контроллера, ядра ОС и прикладной программы. В результате, прикладная программа работает с данными, преобразованными в естественную для программиста форму (файлы, символы, таблицы данных).

По-существу, драйвер обеспечивает ввод/вывод с устройства отдельных единиц информации. При работе с экраном или клавиатурой в качестве этой информации могут рассматриваться выводимые на экран символы.

7.8.1 Работа с видеоадаптером

Современные видеоконтроллеры поддерживают разнообразные текстовые и графические режимы. Текстовые режимы различают по разрешению (по числу символов по горизонтали и вертикали) и цветовой палитре (монохромный или 16-цветный режим). Для графических режимов основным признаком является количество одновременно отображаемых цветов (или иначе, количество двоичных разрядов, необходимых для хранения одной точки изображения):

- монохромный (1-битное кодирование цвета точки);

- 4-цветный CGA (2-битное кодирование);
- 16-цветный EGA/VGA (4-битное кодирование);
- 256-цветный SVGA (8-битное кодирование);
- HiColor (16-битное кодирование);
- TrueColor (24-битное или 32-битное кодирование).

Прежде чем начинать работу в каком-либо режиме, необходимо перевести систему в этот режим.

Внимание! Не рекомендуется ставить эксперименты по прямому управлению режимами работы видеоконтроллера через его регистры: некорректная установка его параметров может вывести из строя монитор!

Для видеоконтроллеров разделение операций на программные и аппаратные особенно жестко: любые переключения режимов должны выполняться при помощи функций BIOS видеоконтроллера или фирменных драйверов, а выводить информацию можно напрямую в видеопамять. Устанавливать необходимый видеорежим рекомендуется один раз при запуске прикладной программы — на весь период ее выполнения. Переключать режимы в процессе работы крайне нежелательно. Существует два основных способа программной установки видеорежима: с помощью функций VGA BIOS и с помощью функций VESA BIOS.

Функции VGA BIOS. В настоящее время графические режимы VGA сильно устарели, но текстовые продолжают успешно применяться, поэтому интерес представляет только подгруппа текстовых функций из стандартного набора VGA BIOS:

- установка видеорежима;
- управление положением и размером курсора;
- переключение видеостраниц;
- управление шрифтами.

Для вызова функций VGA BIOS используется прерывание Int 10h. Набор функций очень большой, но устаревший. Рассмотрим лишь наиболее распространимые из них.

- Int 10h, функция 00h: установить видеорежим.
- Int 10h, функция 00h: установить видеорежим.
- Int 10h, функция 01h: установить размер курсора.
- Int 10h, функция 02h: установить позицию курсора.
- Int 10h, функция 03h: получить позицию и размер курсора.
- Int 10h, функция 05h: установить видеостраницу.

- Int 10h, функция 10h, подфункция 00h: установить один регистр палитры.
- Int 10h, функция 10h, подфункция 01h: установить цвет рамки экрана.
- Int 10h, функция 10h, подфункция 02h: установить все регистры палитры.
- Int 10h, функция 10h, подфункция 03h: переключить бит атрибута «мерцание/яркость».
- Int 10h, функция 10h, подфункция 07h: прочитать один регистр палитры.
- Int 10h, функция 10h, подфункция 08h: прочитать один регистр палитры.
- Int 10h, функция 10h, подфункция 09h: прочитать все регистры палитры.
- Int 10h, функция 10h, подфункция 10h: установить один регистр ЦАП.
- Int 10h, функция 10h, подфункция 12h: перезагрузить группу регистров ЦАП.
- Int 10h, функция 10h, подфункция 15h: прочитать один регистр ЦАП.
- Int 10h, функция 10h, подфункция 17h: прочитать группу регистров ЦАП.
- Int 10h, функция 11h, подфункция 00h: загрузить шрифт пользователя для текстового видеорежима.

Функции VESA BIOS. Фирмы-изготовители поставляют драйверы, в основном, только для Windows, поэтому для остальных ОС возникла необходимость в стандартизации операций по работе с видеоконтроллерами. Эту работу выполнила ассоциация VESA (Video Electronics Standard Association). Обращение к VESA BIOS осуществляется по прерыванию 10h с номером функции 4Fh. После выполнения этого вызова в регистре AX будет возвращен код результата (статус возврата).

- Int 10h, функция 4Fh, подфункция 00h: получить информацию о версии VESA BIOS.
- Int 10h, функция 4Fh, подфункция 01h: получить информацию о параметрах видеорежима.
- Int 10h, функция 4Fh, подфункция 02h: установить видеорежим с заданным номером.

- Int 10h, функция 4Fh, подфункция 03h: определить код текущего видеорежима.
- Int 10h, функция 4Fh, подфункция 04h: сохранить или восстановить состояние видеоконтроллера.
- Int 10h, функция 4Fh, подфункция 05h: управление окнами видеопамяти.
- Int 10h, функция 4Fh, подфункция 06h: получить или установить длину логической строки развертки.
- Int 10h, функция 4Fh, подфункция 07h: получить или установить координаты левого верхнего угла экрана.
- Int 10h, функция 4Fh, подфункция 08h: получить или изменить формат регистров палитры.
- Int 10h, функция 4Fh, подфункция 09h: сохранить или изменить содержимое регистров ЦАП.

Регистры видеоконтроллера. Необходимость непосредственной работы с регистрами видеоконтроллера реально возникает только после перехода в защищенный режим работы процессора *Intel x86,* когда недоступны функции BIOS. Все регистры видеоконтроллера являются 8-разрядными. В состав стандартного контроллера VGA-типа входит шесть групп управляющих регистров:

- внешние регистры;
- регистры контроллера электронно-лучевой трубки (ЭЛТ);
- регистры синхронизатора;
- регистры графического контроллера;
- регистры контроллера атрибутов;
- регистры ЦАП VGA.

В настоящее время изделия различных изготовителей совершенно несовместимы друг с другом на уровне регистров. Но с целью сохранения совместимости с устаревшим ПО, современные видеоадаптеры имитируют работу контроллера VGA в области некоторых (не всех) основных регистров.

Работа в текстовом режиме видеоконтроллера. Текстовый режим до сих пор широко применяется (например, при загрузке: 80×25, 16-цветный, код 03h), т.к. для многих задач он является вполне достаточным, простым, надежным и обладает невысокими требованиями к аппаратуре. Основные операции в текстовом режиме: вывод символов, управление курсором и загрузка шрифта.

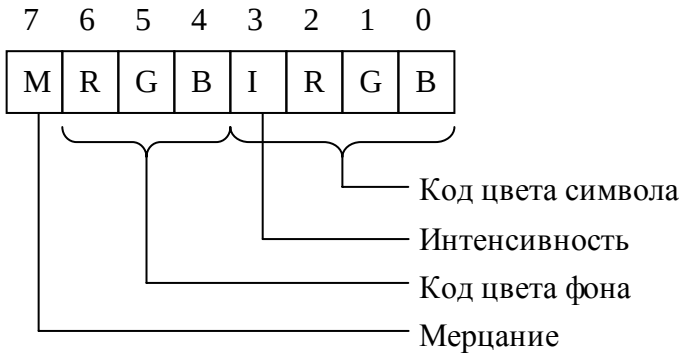


Рис. 7.1 — Формат байта цвета символа
и фона в текстовом режиме

Таблица 7.1 — **Цветовые коды для текстового режима**

Код	Цвет
0000	Черный
0001	Синий
0010	Зеленый
0011	Бирюзовый
0100	Красный
0101	Фиолетовый
0110	Коричневый
0111	Светло серый

Код	Цвет
1000	Серый
1001	Ярко синий
1010	Ярко зеленый
1011	Ярко бирюзовый
1100	Ярко красный
1101	Ярко фиолетовый
1110	Желтый
1111	Белый

Видеопамять в текстовом режиме организована следующим образом: на каждый символ приходится по два байта информации, причем первый байт хранит ASCII-код символа, а второй — цвет символа и цвет фона знакоместа этого символа. В этом режиме для видеопамяти выделено окно размером 32 Кбайт в первом мегабайте адресного пространства процессора, начальный линейный адрес окна `B8000h`.

Для вывода символа в видеопамять обычно используется дополнительный сегментный регистр данных ES, в который перед началом записи в видеопамять нужно занести число, равное абсолютному начальному адресу буфера, поделенному на 16.

Чтобы вывести символ в заданное знакоместо нужно помножить номер строки знакоместа на 160 (длину строки в байтах) и прибавить номер столбца знакоместа, после чего записать результат в индексный регистр. Далее, в соответствующий байт за-

носится ASCII-код символа с помощью косвенной адресации (относительно сегмента ES и избранного индексного регистра). При необходимости значение индексного регистра инкрементируется и в следующий байт записывается код цвета символа и фона.

Кроме вывода символов, в текстовом режиме часто применяется еще два вида операций — перемещение курсора и переключение видеостраниц. Управление курсором осуществляется при помощи регистров видеоконтроллера или функций BIOS.

Установить размер прямоугольника текстового курсора можно при помощи регистров начальной и конечной линии курсора или при помощи функции 01h прерывания Int 10h.

Перемещение курсора по тексту производится путем записи значения смещения курсора относительно начала видеопамяти в регистры старшего и младшего байт адреса курсора. Позиционировать курсор можно так же при помощи функции 02h прерывания Int 10h, но координаты курсора при вызове прерывания задаются в виде номера строки и колонки относительно начала видеостраницы.

Таблица 7.2 — **Знакоместо символов в текстовом 80×25 режиме**

	Столбец 0		Столбец 1		Столбец 2			Столбец 79	
Строка 0	Символ 0		Символ 1		Символ 2		…	Символ 79	
	код	цвет	код	цвет	код	цвет		код	цвет
Строка 1	Символ 80		Символ 81		Символ 82		…	Символ 159	
	код	цвет	код	цвет	код	цвет		код	цвет
Строка 2	Символ 160		Символ 161		Символ 162		…	Символ 239	
	код	цвет	код	цвет	код	цвет		код	цвет
							…		
Строка 24	Символ 1920		Символ 1921		Символ 1922		…	Символ 1999	
	код	цвет	код	цвет	код	цвет		код	цвет

Переключение видеостраниц осуществляется путем записи смещения левого верхнего угла видеостраницы относительно начала видеопамяти в регистры старшего и младшего байт начального адреса. Эту же операцию можно выполнить при помощи функции 05h прерывания Int 10h. Видеоконтроллер в текстовом режиме обеспечивает 8 видеостраниц, но используется обычно

только основная (нулевая) страница. Дополнительные страницы, как правило, используются для выдачи сообщений оператору, не разрушая текст на основной странице.

7.8.2 Работа в графическом режиме

Современные видеоконтроллеры отличаются от VGA-контроллеров тем, что обеспечивают работу с высокими разрешениями и позволяют использовать линейную адресацию видеопамяти.

В 256-цветных режимах каждой точке изображения на экране соответствует один байт видеопамяти, в который записывается код цвета точки. Этот код не используется непосредственно, а служит индексом в специальном массиве, содержащем 256 строк по 3 элемента — таблице цветов ЦАП. Каждый из трех элементов задает интенсивность одного из основных цветов (RGB) электронно-лучевой трубки (ЭЛТ). Значения интенсивностей, выбранные из строки, соответствующей хранящемуся в видеопамяти коду, передаются в ЦАП.

Видеопамять адресуется на экран слева направо и сверху вниз как показано в табл. 7.3.

Таблица 7.3 — **Отображение видеопамяти в 256-цветном режиме 640×480 точек**

	Столбец 0	Столбец 1	Столбец 2		Столбец 639
Строка 0	байт 0	байт 1	байт 2		байт 639
Строка 1	байт 640	байт 641	байт 642		байт 1279
Строка 2	байт 1280	байт 1281	байт 1282		байт 1919
Строка 479	байт 306560	байт 306561	байт 306562		байт 307199

В режимах группы ***DirectDraw*** (*HiColor* и *TrueColor*) информация поступает на ЦАП непосредственно из видеопамяти. Соответственно красная, зеленая и синяя (RGB) составляющие цвета точки представлены отдельными полями в выделенной для хранения точки области видеопамяти (от 2 до 4 байт на точку).

В режимах ***HiColor*** точка кодируется 16-разрядным словом, двумя вариантами: HiColor15 (формат 1:5:5:5) и HiColor16 (формат 5:6:5) как показано на рис. 7.2.

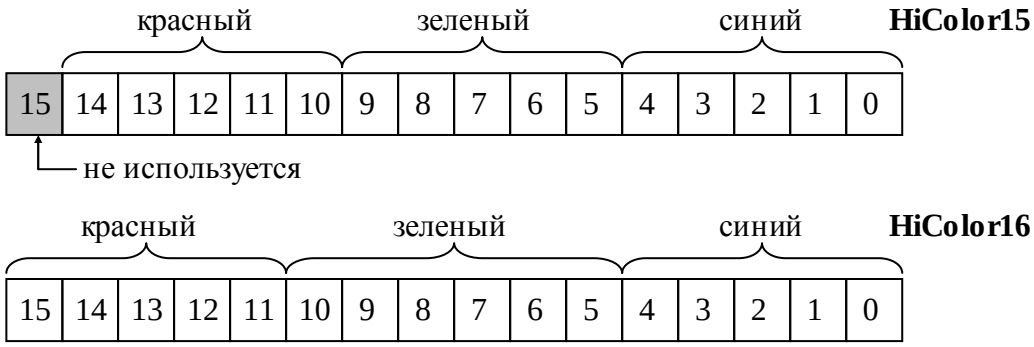


Рис. 7.2 — Форматы данных группы HiColor

В режимах TrueColor для хранения каждого компонента цвета точки выделено по одному байту видеопамяти. Существуют два варианта представления данных (см. рис. 7.3).

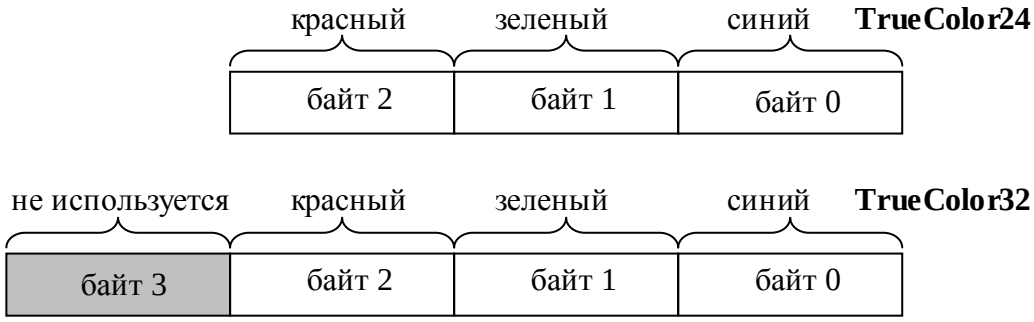


Рис. 7.3 — Форматы данных группы TrueColor

Дополнительный байт добавлен в режим TrueColor32 для выравнивания — на экран он не отображается. Наличие лишнего байта объясняется тем, что процессор может пересылать информацию лишь байтами, 16-разрядными или 32-разрядными словами, а так же тем, что передача данных словами приблизительно в три раза быстрее, чем побайтная.

7.8.3 Палитра VGA

Во времена монохромных дисплеев понятия «палитра» не существовало, и цветов было только два: черный (цвет пустого экрана) и зеленый, желтый или белый, как повезет. Имелись даже дисплеи с оранжевым свечением. Кодировался цвет очень просто —

одним разрядом: 0 — темная точка, 1 — светлая. Но скоро выяснилось, что этого явно недостаточно, и в монохромном режиме добавилось управление яркостью, т. е. второй разряд. И эти два разряда обеспечивали три или четыре уровня яркости (зачастую разряд яркости не влиял на черный цвет).

С созданием цветных дисплеев картина радикально изменилась, стало отображаться гораздо больше цветов. В первом массовом цветном видеоадаптере CGA для персональных компьютеров можно было использовать до 16 цветов в текстовом режиме и до четырех в графическом, что обусловлено ограничением объема видеопамяти (всего 16 Кбайт). При 4 тыс. знакомест текстового режима этого объема хватало с избытком, а вот при 64 тыс. точек графического (разрешение 320x200 точек) для кодирования цвета каждой точки нельзя было отвести больше двух разрядов.

Достаточно просто формировались 16 цветов CGA: по одному разряду на красную, зеленую и синюю составляющую и еще один разряд для повышенной яркости. Сразу возникает вопрос: если точка может быть только одного из четырех цветов, то какие цвета следует выбрать? Для этого в CGA предусмотрено два набора взаимно контрастных цветов: в один набор входят черный, голубой, малиновый и белый, в другой — черный, зеленый, желтый и красный. Значит, появились две фиксированные палитры. Иными словами, палитра позволяет одновременно использовать только несколько цветов из гораздо большего числа доступных. Таким образом, благодаря палитре снижаются до разумных пределов требования к объему видеопамяти. Она устанавливается для всего экрана, следовательно, нельзя сделать одну его часть бело-малиновой, а другую — красно-желтой.

В видеоадаптере EGA, сменившем CGA, яркость каждой цветовой составляющей можно было устанавливать независимо, т. е. на цветовую составляющую приходилось по два разряда (четыре градации яркости). Всего было доступно 64 цвета, но на экране одновременно появлялось не больше 16 в графическом режиме. Кроме того, увеличилось число палитр CGA до 6416 (в десятичной записи — 79228162514264337593543950336): каждый из 16 номеров цвета можно было выбирать из 64 доступных. Поэтому от номеров палитры отказались, а для ее определения стали использовать массив, где указывались составляющие трех ос-

новных цветов для каждого из номеров цвета. В текстовом режиме также появилась возможность изменить цвет, соответствующий определенному номеру.

Последний стандарт видеоадаптера VGA существенно расширил диапазон допустимых цветов. Теперь на любой номер цвета выделяется по три шестиразрядных регистра, по одному на цветовую составляющую. Каждый компонент мог выводиться в 64 градациях яркости, а общее число отображаемых цветов достигло 262 144. Конечно, глупо было бы ограничиться лишь 16 цветами, и потому применили режим с глубиной цвета 8 разрядов на точку, вследствие чего число цветов возросло до 256. Нетрудно подсчитать, что это потребовало разместить внутри видеоадаптера 768 регистров для хранения компонентов цвета.

Появление 240 дополнительных цветов наряду с 16 прежними вывело компьютерную графику и игровую индустрию на качественно новый уровень. Если раньше мало кто задумывался о целесообразности переопределения стандартной палитры, потому что «лишних» цветов, которые можно было бы заменить другими, не было, то теперь для этого открылись широкие возможности.

8 ПАРАЛЛЕЛЬНЫЕ ИНТЕРФЕЙСЫ

Параллельные интерфейсы характеризуются тем, что в них для передачи каждого бита в слове используются отдельные сигнальные линии, и байт (слово) передается за один цикл. Параллельные интерфейсы используют логические уровни ТТЛ (транзисторно-транзисторной логики), что ограничивает длину кабеля из-за невысокой помехозащищенности ТТЛ-интерфейса. Гальваническая развязка отсутствует. Передача данных может быть как однонаправленной, так и двунаправленной.

К параллельным интерфейсам, помимо шин, относят LPT-порт, используемый, как правило, для подключения принтеров, отсюда и название LPT-порт (***l**ine **p**rin**t**er* — построчный принтер).

8.1 Стандарт IEEE 1284-1994

Стандарт на параллельный интерфейс IEEE 1284, принятый в 1994 году, определяет 5 режимов обмена данными, метод согласования режима, физический (кабели, соединители) и электрический интерфейсы (драйверы, приемники, окончание линии, импеданс). Согласно IEEE 1284, возможны следующие режимы обмена данными через параллельный порт:

- ***Режим совместимости*** (*Compatibility Mode*) — однонаправленный (вывод) по протоколу *Centronics*. Этот режим соответствует стандартному порту *SPP* (*Standard Parallel Port*).
- ***Полубайтный режим*** (*Nibble Mode*) — ввод байта в два цикла (по 4 бита), используя для приема линии состояния. Этот режим обмена может использоваться на любых адаптерах.
- ***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*).
- ***Режим EPP*** (*Enhanced Parallel Port, EPP Mode*) — двунаправленный обмен данными. Управляющие сигналы интерфейса генерируются аппаратно во время цикла обращения к порту.
- ***Режим ECP*** (*Extended Capability Port, ECP Mode*) — двунаправленный обмен данными с возможностью аппаратного сжатия данных по методу *RLE* (*Run Length Encoding*) и использова-

ния *FIFO*-буферов и *DMA*. Управляющие сигналы интерфейса генерируются аппаратно.

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

Все параллельные порты в режимах Compatibility Mode, Nibble Mode и Byte Mode используют только ***программное управление передачей данных***. Драйвер должен устанавливать данные, проверять сигнал (*Busy#*), устанавливать соответствующие сигналы управления (*-Strobe#*) и затем переходить к следующему байту, что ограничивает эффективную скорость передачи данных на уровне от 50 до 100 Кбайт/с.

В режимах EPP и ECP для передачи данных используются ***аппаратные средства***. Например, в режиме EPP байт данных может быть передан периферии простой инструкцией *Out*. Контроллер ввода-вывода самостоятельно выполняет операции подтверждения связи и передачи данных на периферию.

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.

Адаптер параллельного интерфейса 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***. Порт имеет внешнюю 8-битную шину данных, 5-битную шину сигналов состояния и 4-битную шину управляющих сигналов.

BIOS поддерживает до четырех (иногда до трех) LPT-портов (LPT1—LPT4) своим сервисом — прерыванием ***INT 17h***, обеспечивающим через них связь с принтером по интерфейсу *Centronics*. Этим сервисом BIOS осуществляет вывод символа (по опросу готовности, не используя аппаратных прерываний), инициализацию интерфейса и принтера, а также опрос состояния принтера.

8.2 Интерфейс Centronics (Compatibility Mode)

Понятие ***Centronics*** (*Compatibility Mode, SPP — Standard Parallel Port*) относится как к набору сигналов и протоколу взаимодействия, так и к 36-контактному разъему на принтерах. Назначение сигналов приведено в табл. 1.2, а временные диаграммы обмена с принтером — на рис. 8.1.

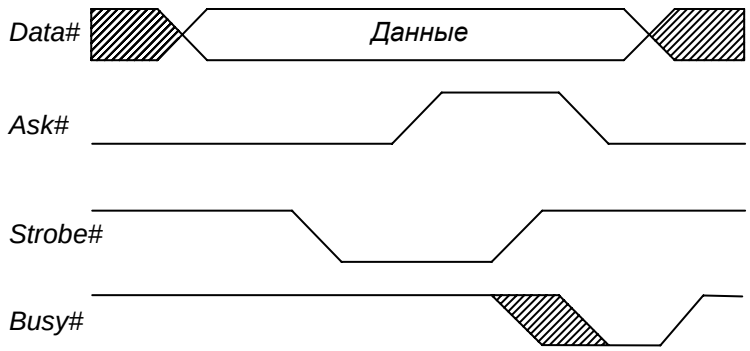


Рис. 8.1 — Передача данных по интерфейсу *Centronics*

Традиционный порт ***SPP*** (*Standard Parallel Port*) является однонаправленным портом, через который программно реализуется протокол обмена *Centronics*. Этот режим определяет протокол, используемый большинством PC, для передачи данных на принтер. В этом режиме данные помещаются на линии данных порта, состояние принтера не проверяется ни на какие ошибки и на занятость (сигнал *Busy#*), и затем программно формируется строб данных (*Strobe#*) для тактирования принтера.

Порт вырабатывает аппаратное прерывание по импульсу на входе *Ack#*. Сигналы порта выводятся на разъем *DB-25S* (розетка), установленный непосредственно на плате адаптера (или системной плате) или соединяемый с ней плоским шлейфом. Название и назначение сигналов разъема порта (табл. 8.1) соответствуют интерфейсу *Centronics*.

В табл. 8.1 обозначение *I/O* задает направление передачи (*вход/выход*) сигнала порта: *O/I* обозначает выходные линии, состояние которых считывается при чтении из портов вывода; *(I)* — выходные линии, состояние которых может быть считано только при особых условиях.

Вход Ack# соединен резистором (10 кОм) с питанием +5 В.

Таблица 8.1

Контакт DB-25S	***Провод шлейфа***	***I/O***	***Reg.Bit***	***Сигнал***
1	1	*O/I*	CR: 0 invert	*Strobe#*
2	3	*O(I)*	DR: 0	*Data#0*
3	5	*O(I)*	DR: 1	*Data#1*
4	7	*O(I)*	DR: 2	*Data#2*
5	9	*O(I)*	DR: 3	*Data#3*
6	11	*O(I)*	DR: 4	*Data#4*
7	13	*O(I)*	DR: 5	*Data#5*
8	15	*O(I)*	DR: 6	*Data#6*
9	17	*O(I)*	DR: 7	*Data#7*
10	19	*I*	SR: 6	*Ack#*
11	21	*I*	SR: 7 invert	*Busy#*
12	23	*I*	SR: 5	*PaperEnd#*
13	25	*I*	SR: 4	*Select#*
14	2	*O/I*	CR: 1 invert	*AutoLF#*
15	4	*I*	SR: 3	*Error#*
16	6	*O/I*	CR: 2	*Init#*
17	8	*O/I*	CR: 3 invert	*SelectIn#*
18—25	10, 12, 14, 16,	18, 20, 22, 24, 26	–	–

8.2.1 Фазовые переходы режима Compatibility Mode

1. Запись данных в регистр данных *Data Register (DR)*.

2. Программа читает регистр состояния *Status Register (SR)*, чтобы проверить принтер на занятость (*Busy#*).

3. Если принтер не занят, то производится запись в регистр управления *Control Register (CR)*, чтобы установить линию *Strobe#*.

4. Производится запись в регистр управления, чтобы сбросить линию *Strobe#*.

Как может быть замечено, для вывода одного байта данных требуется четыре I/O инструкции и много дополнительных инструкций. В результате полоса пропускания порта ограничена величиной примерно ***150 Кбайт/с***.

Этот устаревший режим был тем не менее включен в стандарт как способ обеспечить обратную совместимость с огромной

массой уже установленных принтеров и периферийных устройств. Для обеспечения обратной передачи и высокопроизводительной связи используются другие режимы.

8.2.2 Регистры режима Compatibility Mode

Стандартный порт имеет три 8-битных регистра, расположенных по соседним адресам в пространстве ввода/вывода, начиная с базового адреса порта (*Base#*).

Data Register (DR) — регистр данных, с адресом *Base#*. Данные, записанные в этот порт, выводятся на выходные линии интерфейса. 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. Если в порт 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. Таким образом, на многих старых моделях адаптеров можно реализовать порт ввода дискретных сигналов, однако выходным цепям передатчика информации придется *«бороться»* с выходным током логической единицы выходных буферов адаптера. Схемотехника ТТЛ 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 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»*. Так что такой способ ввода не будет работать на всех компьютерах.

На некоторых адаптерах портов выходной буфер отключается перемычкой на плате. Тогда порт превращается в обыкновенный порт ввода.

Status Register (SR) — регистр состояния; представляет собой 5-битный порт ввода сигналов состояния принтера (биты

SR:4 — SR:7), расположенный по адресу Base#+*1*. Бит SR:7 инвертируется — низкому уровню сигнала соответствует единичное значение бита в регистре, и наоборот.

Назначение бит регистра состояния (в скобках даны номера контактов разъема):

SR:7 — *Busy#* — инверсные отображения состояния линии *Busy#* (11): при низком уровне на линии устанавливается единичное значения бита — разрешение на вывод очередного байта.

SR:6 — *Ack#* (*Acknowledge*) — отображения состояния линии *Ack#* (10).

SR:5 — *PE#* (*Paper End*) — отображения состояния линии *PaperEnd#* (12). Единичное значение соответствует высокому уровню линии — сигналу о конце бумаги в принтере.

SR:4 — *Select#* — отображения состояния линии *Select#* (13). Единичное значение соответствует высокому уровню линии — сигналу о включении принтера.

SR:3 — *Error#* — отображения 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, SR:0 — зарезервированы.

Control Register (CR) — регистр управления, расположенный по адресу *Base#+2*. Как и регистр данных, этот 4-битный порт вывода допускает запись и чтение (биты 0—3), но его 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.

Назначение бит регистра управления:

CR:7, CR:6 — зарезервированы.

CR:5 — *Direction#* — бит управления направлением передачи (только для портов PS/2). 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 — *SelectIn#* — единичное значение бита соответствует низкому уровню на выходе *SelectIn#* (17) — сигналу, разрешающему работу принтера по интерфейсу *Centronics*.

CR:2 — *Init#* — нулевое значение бита соответствует низкому уровню на выходе *Init#* (16) — сигналу аппаратного сброса принтера.

CR:1 — *AutoLF#* — единичное значение бита соответствует низкому уровню на выходе *AutoLF#* (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 I*RQ7* или *IRQ5*) вырабатывается по отрицательному перепаду сигнала на выводе 10 разъема интерфейса (*Ack#*) при установке CR:4 в *«единицу»*. Во избежание ложных прерываний контакт 10 соединен резистором с шиной +5 В. Прерывание вырабатывается, когда принтер подтверждает прием предыдущего байта. BIOS это прерывание не использует и не обслуживает.

Процедура вывода байта по интерфейсу *Centronics* включает следующие шаги (в скобках приведено требуемое количество шинных операций процессора):

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

Из описания этой процедуры 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 работают 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 передаются 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.

8.3 Расширения параллельного порта

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

Двунаправленный порт 1 (***Type 1 parallel port***) — интерфейс, введенный в PS/2. Такой порт кроме стандартного режима может работать в режиме ввода или двунаправленном режиме. Протокол обмена формируется программно, а для указания направления передачи в регистр управления порта введен специальный бит CR:5 установленный на *«нуль»* — буфер данных работает на вывод, CR:5 при *«единице»* — на ввод. Не путайте этот порт, называемый также ***enhanced bi-directional***, с EPP. Данный тип порта прижился и в обычных компьютерах.

Порт с прямым доступом к памяти (***Type 3 DMA parallel port***) применялся в PS/2 моделей 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***.

Многие объединенные 1284 I/O контроллеры и по настоящий день осуществляют режим, который использует FIFO буфер, чтобы передать данные по протоколу режима *Compatibility Mode*. Этот режим называют «*Быстрый Centronics*» или «*Режим FIFO Параллельного Порта*». Когда установлен этот режим, данные, записываемые в FIFO порт, будут переданы принтеру с использованием аппаратных средств формирования стробов для подтверждения связи. Так как имеются очень небольшие времена ожидания между передачами и программное обеспечение не должно делать никакого стробирования или проверки рукопожатия, скорости передачи в некоторых системах достигают ***500 Кбайт/с***. Этот режим, однако, не входит в стандарт IEEE 1284.

Не будучи стандартизованными, такие порты разных производителей требовали использования собственных специальных драйверов. Программы, использующие прямое управление регистрами стандартных портов, не умели более эффективно их использовать. Такие порты часто входили в состав мультикарт VLB. Существуют их варианты с шиной ISA, в том числе встроенные.

8.4 Полубайтный режим ввода — Nibble Mode

Полубайтный режим ввода-вывода — ***Nibble Mode***, называемый так же ***режимом тетрады***, предназначен для двунаправленного обмена данными. 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 *Ack#,* 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. 8.2, временные диаграммы — на рис. 8.2.

Таблица 8.2

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

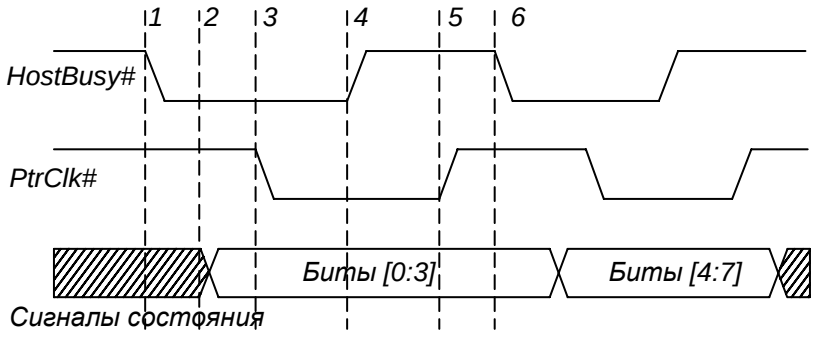


Рис. 8.2 — Прием данных в полубайтном режиме

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.

Режим *Nibble Mode* — наиболее общий способ получить данные из принтера или периферийного устройства. Этот режим обычно объединяется с режимом *Совместимости* или собственно прямым режимом канала, чтобы получить полный двунаправленный канал.

Полубайтный режим сильно нагружает процессор, и поднять скорость обмена выше ***50 Кбайт/с*** не удается. 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.

Режим *Nibble Mode*, подобно режиму *Compatibility Mode*, требует, чтобы программное обеспечение управляло протоколом, устанавливая и читая сигналы на линиях параллельного порта. Режим *Nibble Mode* требует наиболее интенсивной работы программного обеспечения для обратной передачи. По этой причине, имеется серьезное ограничение по скорости для этого типа передачи данных. Главное преимущество этого режима — способность функционировать на всех PC, которые имеют параллельный порт. Ограничения производительности, имеющие место в режиме *Тетрады*, не имеют большого значение для периферийных устройств с небольшими требованиями к пропускной способности канала в обратном направлении, типа принтеров, но могут оказаться недопустимыми для адаптеров локальной вычислительной сети, дисководов или CD-ROM.

8.5 Двунаправленный байтный режим — Byte Mode

В данном режиме данные принимаются с использованием двунаправленного порта, у которого выходной буфер данных может отключаться установкой бита *CR:5* в *«единицу»*. Как и предыдущие, режим является программно-управляемым, т. е. все сигналы квитирования анализируются и устанавливаются драйве-

ром. Сигналы порта описаны в табл. 8.3, временные диаграммы — на рис. 8.3.

Таблица 8.3

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

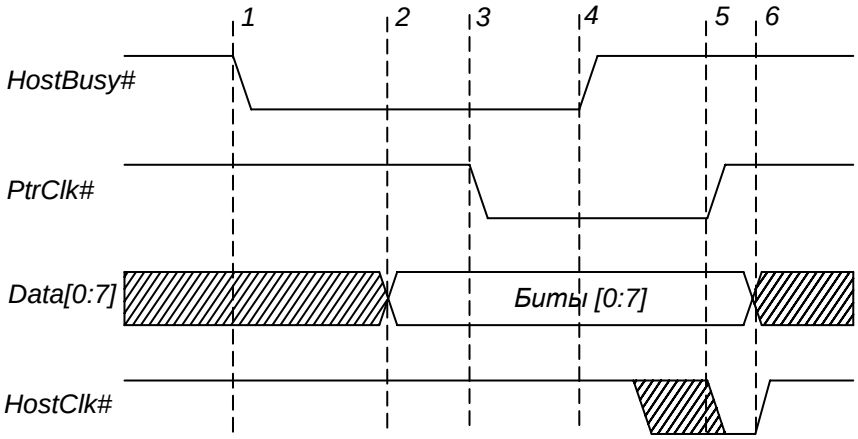


Рис. 8.3 — Прием данных в байтном режиме

Фазы приема байта данных:

1. Хост сигнализирует о готовности приема данных установкой низкого уровня на линии *HostBusy#*.

2. Периферийное устройство в ответ помещает байт данных на линии *Data[0:7]*.

3. Периферийное устройство 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#*.

7. Шаги 1—6 повторяются для каждого следующего байта.

Побайтный режим позволяет поднять скорость обратного канала до скорости прямого канала в стандартном режиме. Однако он может работать только на двунаправленных портах.

8.6 Режим EPP

Протокол ***EPP*** (*Enhanced Parallel Port* — улучшенный параллельный порт) был разработан компаниями Intel, Xircom и Zenith Data Systems задолго до принятия IEEE 1284. Он предназначен для повышения производительности обмена по параллельному порту. EPP 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.

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. 8.4.

Таблица 8.4

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

* Сигналы действуют в последовательности согласования.

Используемый в режиме EPP *Прозрачный протокол блокированного квитирования* (***interlocked handshakes***, *с рукопожатиями*) позволяет осуществить передачу данных на скорости самого медленного из двух интерфейсов: адаптера ведущего или периферийного устройства. Это свойство *адаптивной скорости* прозрачно и для ведущего, и для периферии. Протокол автоматически настраивается и под длину кабеля — вносимые задержки только приведут к удлинению цикла.

Все режимы передачи стандарта 1284 осуществлены с рукопожатиями.

Рукопожатие основано на том, что каждый переход сигнала управления подтвержден противоположной стороной интерфейса. В вышеупомянутой диаграмме *DataStrobe#* может быть установлен, потому что *Wait#* низок, *Wait#* сбрасывается в ответ на установление *DataStrobe#, DataStrobe#* сбрасывается в ответ на сбрасываемый *Wait#*, и наконец *Wait#* устанавливается в ответ на сбрасываемый *DataStrobe#.* Таким образом, периферия может управлять установкой времени, требуемого для действия. Это выполнено следующим способом: время установки является временем от установления *DataStrobe#* до сброса *Wait#,* периферия управляет этим временем. Преимущество рукопожатия также состоит в возможности формирования цикла передачи, независимого от длины кабеля. Режимы Тетрады, Байта, EPP и ECP используют *interlocked handshakes* (рукопожатие).

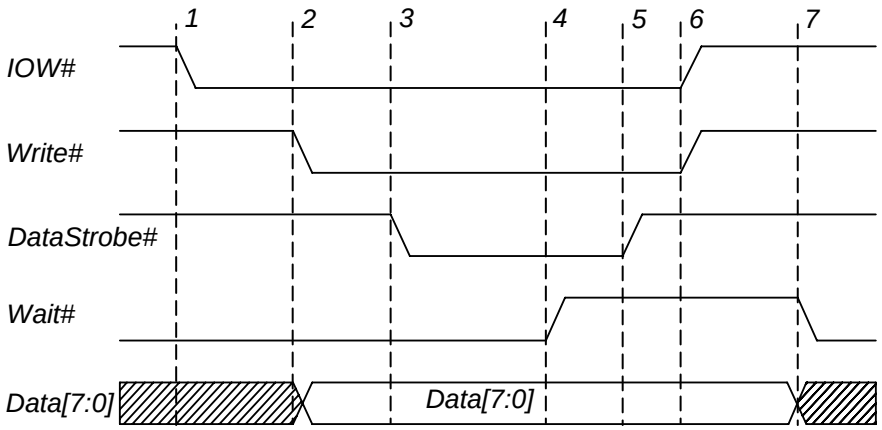


Рис. 8.4 — EPP цикла записи данных

Фазовые переходы цикла записи данных:

1. Программа выполняет I/O цикл записи (*IOWR#*) в порт 4 (*EPP Data Port*).

2. Адаптер устанавливает сигнал *Write#* (низкий уровень), и данные помещаются в выходную шину параллельного порта.

3. При низком уровне *Wait#* устанавливается строб данных (*DataStrobe*) в низкий уровень. С этого момента на Wait# установлен «нуль».

4. Порт ждет подтверждения от периферийного устройства (*Wait#* сброшен в высокий уровень).

5. Строб данных сброшен и внешний цикл EPP завершается.

6. Процессорный цикл ввода-вывода I/O окончен.

7. Периферийное устройство устанавливает на *Wait#* низкий уровень, чтобы указать, что может начинаться следующий цикл.

Одной из наиболее важных особенностей является то, что полная передача данных происходит в пределах одного I/O цикла ввода-вывода. Следовательно, используя EPP протокол для передачи данных, система может достигать скоростей передачи от *500 Кбайт/с* до *2 Мбайт/с*. Таким образом, периферийные устройства, подключенные к порту, могут работать с той же производительностью, что и вставная плата ISA. Способность получить этот уровень производительности от устройства, подключенного к параллельному порту — одна из главных особенностей EPP протокола.

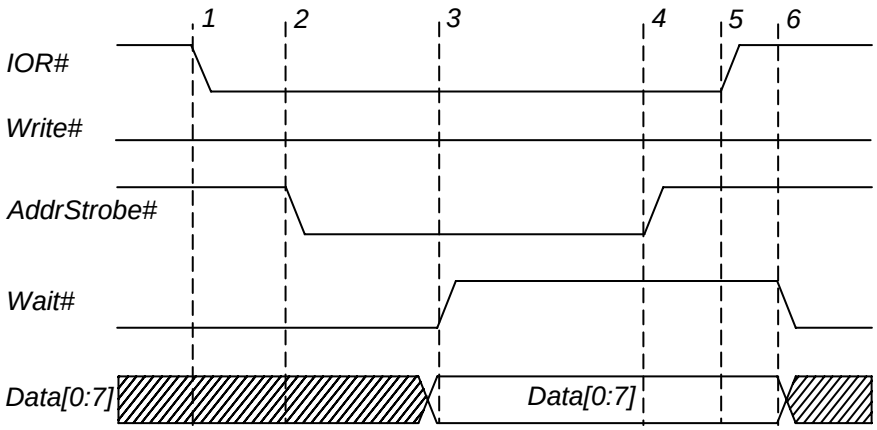


Рис. 8.5 — EPP цикл чтения адреса

Как уже отмечалось, ранние версии интерфейсов параллельного порта отклоняются от IEEE 1284 протокола. В начале цикла *DataStrobe#* или *AddrStrobe#* устанавливаются независимо от состояния сигнала *Wait#*. Это означает, что периферия не могла бы удерживать начало цикла при сброшенном *Wait#*. Этот протокол иногда упоминается как ***EPP 1.7***, в отношении предложения Xircom версии 1.7. Эта версия осуществлена Intel в оригинальном контроллере *I/O 82360*. IEEE 1284 EPP совместимая периферия будет работать должным образом с ведущим адаптером версии EPP 1.7, но периферия EPP 1.7 не может работать должным образом с 1284 совместимым ведущим.

8.6.1 Регистры интерфейса EPP

Самое простое представление EPP с точки зрения программного обеспечения — расширение определения регистров для стандартного параллельного порта. Как показано ранее, SPP состоит из трех регистров, смещенных от базового адреса порта: порт данных *Data Register (DR)*, порт статуса *Status Register (SR)* и порт управления *Control Register (CR)*. Обычно реализации EPP расширяют это определение, чтобы использовать порты, не определенные SPP.

Режим EPP имеет расширенный набор регистров (табл. 8.5), который занимает в пространстве ввода/вывода 5—8 смежных байт. При выполнении одиночной инструкции *Записи* по адресу *Base+4*, контроллер EPP произведет необходимые сигналы рукопожатия и стробы, чтобы передать данные, использующие *EPP DataWrite* цикл. Инструкции ввода-вывода по базовым адресам (порты от 0 до 2) будут выполняться точно так же, как принято для стандартного параллельного порта. Это гарантирует совместимость со стандартными периферийными устройствами параллельного порта и принтерами. Циклы *Адреса* генерируются тогда, когда чтение или запись производятся по адресу *Base+3*.

Таблица 8.5

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
Not Defined	+5...+7	EPP	N/A	В некоторых контроллерах могут использоваться для 16-, 32- битных операций ввода-вывода

Порты от 5 до 7 используются по-разному различными реализациями аппаратных средств. Они могут использоваться для реализации 16- или 32-разрядного программного интерфейса, или как регистры конфигурации, или не использоваться вообще. Например, карта *FarPoint Communications F/PortPlus* имеет только интерфейс данных с 8 битами, но к ней можно обращаться, используя 32-разрядный ввод-вывод для EPP операций с данными. ISA контроллер перехватит 32-разрядный ввод-вывод и фактически произведет 4 быстрых 8-разрядных цикла ввода-вывода. Первый цикл будет адресован порту ввода-вывода, используя *байт 0* (биты 0—7), второй цикл пойдет по адресу *порт+1* для *байта 1*, тогда *порт+2* для *байта 2* и, наконец, *порт+3* для *байта 3*. Эти дополнительные циклы производятся аппаратными средствами и прозрачны для программного обеспечения. Полное время для этих четырех циклов будет меньше, чем циклы для 4 независимых байтов. Например, в плате *F/PortPlus* (от *FarPoint Communications*) располагаются 4 порта ввода-вывода (со смещениями от 4 до 7) от внутреннего EPP регистра *Данных*. Это позволяет программному обеспечению использовать 32-разрядные операции

ввода-вывода для EPP передачи данных. Циклы Адреса все еще ограничиваются 8-разрядным вводом-выводом.

Способность передавать данные к или от персонального компьютера при помощи единственной инструкции позволяет параллельным портам в EPP режиме передавать данные на скоростях ISA шины. Быстрее, чем в программном цикле, блок данных может быть передан единственной *REP_IO* инструкцией. В зависимости от реализации порта ведущего адаптера и способности периферийного устройства, EPP порт может передавать данные со скоростями от *500 Кбайт/с* до почти *2 Мбайт/с*. Эта скорость передачи данных более чем достаточна для того, чтобы сетевые адаптеры, CD-ROM, стриммеры и другие периферийные устройства, работали при почти ISA уровне производительности.

Периферийное устройство, подключенное к параллельному порту EPP, может работать на уровне производительности устройства, подключаемого через слот ISA. Но при этом периферийное устройство может регулировать длительность всех фаз обмена с помощью всего лишь одного сигнала *WAIT#*. Протокол автоматически подстраивается и под длину кабеля — вносимые задержки только приведут к удлинению цикла. Поскольку применяемые кабели, соответствующие IEEE 1284, имеют вполне одинаковые волновые свойства для разных линий, нарушение передачи, связанных с состязаниями сигналов, происходить не должно. При подключении сетевых адаптеров или внешних дисков к параллельному порту можно наблюдать непривычное явление: снижение их производительности по мере удлинения интерфейсного кабеля.

Важной чертой EPP является то, что обращение процессора к периферийному устройству осуществляется в реальном времени — здесь нет никакой буферизации. Программный драйвер всегда способен наблюдать состояние и подавать команды в точно известные моменты времени. Циклы чтения и записи могут чередоваться в произвольном порядке или идти блоками. Такой тип обмена наиболее пригоден для регистро-ориентированной периферии или периферии, работающей в реальном времени — сетевых адаптеров, устройств сбора информации и управления, дисковых устройств и т.п.

EPP протокол и его текущие реализации обеспечивают тесную связь между периферийным драйвером и периферией. Это означает, что программа — драйвер всегда способна определять состояние связи с периферией и управлять им в любое время. Всегда можно смешивать операции чтения и записи и блочную передачу. Этот тип связи идеален для многих ориентируемых регистром или управляемых в реальном масштабе времени периферийных устройств типа сетевых адаптеров, систем сбора данных, портативных жёстких дисков и других устройств.

8.7 Режим ECP

Протокол ***ECP*** (*Extended Capability Port* — порт с расширенными возможностями был предложен фирмами Hewlett Packard и Microsoft как прогрессивный режим связи с периферией типа принтеров и сканеров. Как и EPP, данный протокол обеспечивает высокопроизводительный обмен данными хоста с периферийными устройствами. 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 счетчика ***RLC*** (*Run-Length Count*).

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

- компрессия данных хост адаптеров по методу RLE;
- буферизация FIFO для прямого и обратного канала;
- применение DMA и программного ввода/вывода.

Компрессия в реальном времени по методу ***RLE*** (*Run-Length Encoding*) позволяет достичь коэффициента сжатия до 64:1 при передаче растровых изображений, которые обычно имеют длинные строки имеющихся байт. Естественно, компрессию можно использовать, только если ее поддерживает и хост, и периферийное устройство.

Канальная адресация ECP применяется для адресации множества логических устройств, входящих в одно физическое. Например, в комбинированном устройстве факс/модем/принтер, подключаемом только к одному параллельному порту, возможен одновременный прием факса и печать на принтере. Каждая из этих отдельных функций может рассматриваться как отдельное логическое устройство внутри этого корпуса. В режиме SPP, если принтер установит сигнал занятости, канал будет занят ожидающими данными, пока принтер их не примет. В режиме ECP программный драйвер программный драйвер просто адресуется к другому логическому каналу того же порта.

Как и в других режимах 1284, протокол ECP переопределяет сигналы SPP (табл. 8.6).

Таблица 8.6

Сигнал SPP	***Имя в режиме ECP***	***I/O***	***Описание***
Strobe#	HostClk	O	Используется с PeriphAck, чтобы передать данные или адрес в прямом направлении
AutoFeed#	HostAck	O	Представляет состояние Команда/Данные в прямом направлении. Используется с PeriphClk для передачи данных в обратном направлении
SelectIn#	1284Active	O	Высокий уровень, когда компьютер находится в 1284 режиме передачи
Init#	ReverseRequest#	O	Выставляется низкий уровень для установления обратного направления передачи
Ack#	PeriphClk	I	Используется с HostAck для передачи данных в обратном направлении
Busy#	PeriphAck	I	Используется с HostClk, для передачи данных или адреса в прямом направлении. Представляет состояние Команда/Данные в обратном направлении

Окончание табл. 8.6

Сигнал SPP	***Имя в режиме ECP***	***I/O***	***Описание***
PE#	AckReverse#	I	Устанавливается низкий уровень, чтобы подтвердить nReverseRequest
Select#	Xflag	I	Флаг Расширяемости
Error#	PeriphRequest#	I	Низкий уровень устанавливается периферийным устройством, чтобы указать, что обратные данные доступны
Data[0:7]	Data[0:7]	Bi-Dir	Используются для обмена данными между периферийным устройством и компьютером

Прямая передача данных на внешнем интерфейсе состоит из следующих шагов (см. рис. 6.6):

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 действительность данных.

3. ПУ отвечает установкой высокого уровня на линии PeriphAck.

4. Хост устанавливает высокий уровень на линии HostClk, и этот перепад может использоваться для фиксации данных в ПУ.

5. ПУ устанавливает низкий уровень на линии PeriphAck для указания готовности к приему следующего байта.

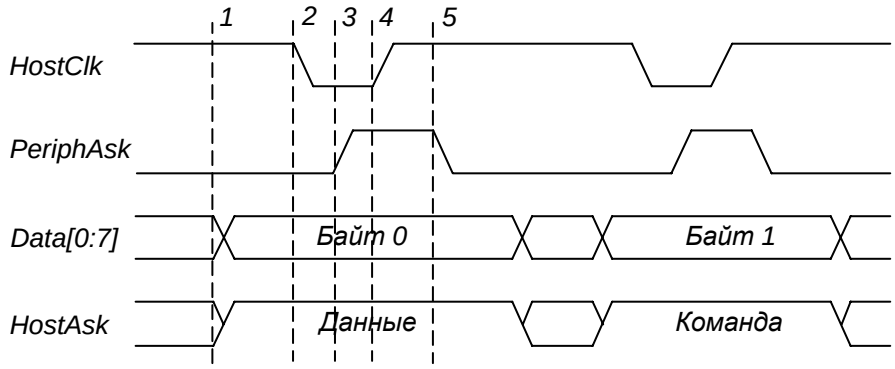


Рис. 8.6 — Прямая передача в режиме ECP

Обратная передача данных состоит из следующих шагов (см. рис. 8.7):

1. Хост запрашивает изменение направления канала, устанавливая низкий уровень на линии ReverseRequest#.

2. ПУ разрешает смену направлений 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 на линии PeriphClk, и этот перепад может использоваться для фиксации данных хостом.

7. Хост устанавливает низкий уровень на линии HostAck для указания готовности к приему следующего байта.

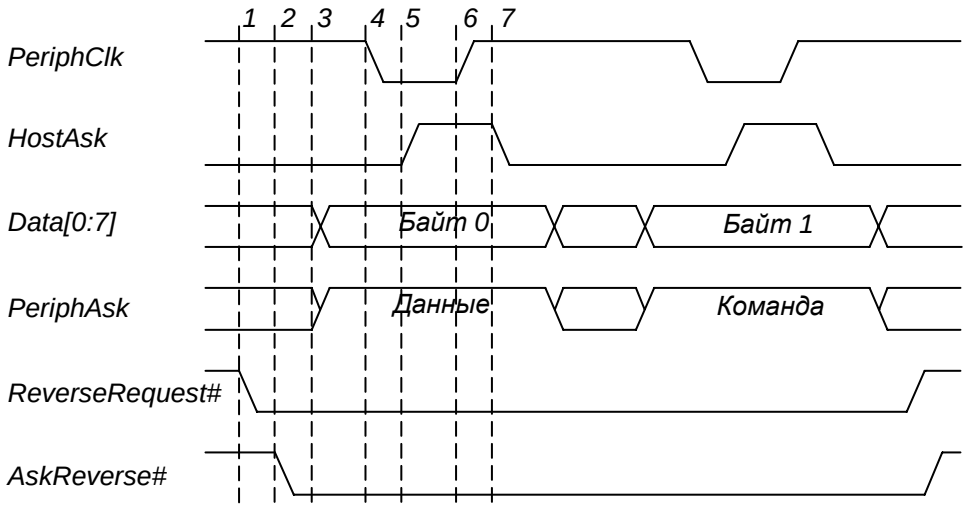


Рис. 8.7 — Обратная передача в режиме ECP

8.7.1 Программный и регистровый интерфейс ECP

Спецификация «The IEEE 1284 Extended Capabilities Port Protocol and ISA Interface Standard» определяет общий регистровый интерфейс для основанных на ISA 1284 адаптеров с ECP. Эта

спецификация также определяет множество режимов (см. табл. 8.7), в которых адаптер может работать.

Таблица 8.7 — **Режимы регистров ECP**

Режим	***Описание***
000	Режим SPP
001	Двунаправленный режим (байтовый режим)
010	Быстрый Centronics
011	Режим параллельного порта ECP
100	Режим параллельного порта EPP (замечание 1)
101	Зарезервирован
110	Тестовый режим
111	Режим конфигурации

Регистровая модель адаптера ECP (табл. 6.7) использует свойства архитектуры стандартной шины и адаптеров ISA, согласно которой для дешифрации адреса портов ввода/вывода использовались только 10 младших линий шины адреса. Старшие линии игнорируются, поэтому обращения по адресам, например, 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, например, 0x378h и 0x778h будет 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 использование для ECP не приведет к стеснению доступного адресного пространства ввода/вывода. Во-вторых, этим обеспечивается совместимость со старыми адаптерами на уровне режимов 000-001 и возможность определения присутствия ECP-адаптера попыткой 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 режимы 000 или 001. В любом из них работает полубайтный режим ввода (Nibble Mode). Из этих

режимов всегда можно переключиться в любой другой, но из старших режимов (010-111) переключение возможно только в 000 или 001. Для корректной работы интерфейса перед выходом из старших режимов необходимо дождаться завершения обмена по прямому доступу и опустошения FIFO-буфера.

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

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 возможностью использования DMA. Сигналы квитирования по протоколу Centronics вырабатываются аппаратно. Сигнал запроса прерывания вырабатывается по состоянию FIFO-буфера, но не по сигналу ACK# (запрос одиночного байта «не интересует» драйвер быстрого блочного вывода).

Режим 011 и является собственно режимом ECP, описанным выше. В этом режиме поток данных и команд, передаваемых в периферийное устройство, 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. При этом 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 ECPAFIFO 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. Из этого описания работы компрессии вполне очевидно, что вывод данных с одновременным использованием компрессии и DMA невозможен.

Режим 100 (EPP) является не совсем обязательным способом включения режима EPP.

Режим 110 (Test Mode) предназначен для тестирования взаимодействия механизмов FIFO и прерываний. В этом режиме данные могут передаваться в/из регистра TFIFO с помощью DMA или программным способом. При этом на внешний интерфейс этот обмен не воздействует. Адаптер отрабатывает операции вхолостую на максимальной скорости интерфейса (как будто сигналы квитирования приходят без задержек). При этом адаптер следит за состоянием буфера и по мере необходимости вырабатывает сигналы запроса прерывания. Так программа может определить максимальную пропускную способность канала.

Режим 111 (Configuration Mode) предназначен для доступа к конфигурационным регистрам. Его выделение защищает адаптер и протокол от некорректных изменений конфигурации в процессе обмена.

Таблица 8.8 — **Регистры ECP**

Смещение	***Имя***	***R/W***	***Режимы ECP****	***Назначение***
000	DR	R/W	000-001	Data Register
000	ECPAFIFO	R/W	01l	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	ECPDFIFO	R/W	011	ECP Data FIFO
400	TFIFO	R/W	110	Test FIFO
400	CNFGA	R	111	Configuration Register A
401	CNFGB	R/W	111	Configuration Register B
402	ECR	R/W	Все	Extended Control Register

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 AUTOFD# и STROBE#) игнорируется.

Регистр ECPAFIFO служит для помещения информации командных циклов (канального адреса или счетчика RLE, в зависимости от бита 7) в FIFO-буфер. Из буфера эта информация будет выдана в командном цикле вывода.

Регистр SDFIFO используется для передачи данных в режиме 010. Данные, записанные в этот регистр (или посланные по каналу DMA), передаются через буфер FIFO по аппаратно-реализованному протоколу Centronics. При этом должно быть задано прямое направление передачи (бит CR.5=0).

Регистр DFIFO используется для обмена данными в режиме ОН (ECP). Данные, записанные в этот 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 ECPCFGA позволяет считывать информацию об адаптере (идентификационный код в битах [7:4].

Регистр ECPCFGB предназначен для хранения информации, необходимой драйверу. Запись в этот регистр не влияет на работу порта.

Регистр ECR является главным управляющим регистром ECP. Его биты имеют следующее назначение:

ECR[7:5] ECP MODE — задают режим ECP.

ECR.4 ERRINTREN# — (Error Interrupt Disable) запрещает прерывания 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 или переопустошения буфера.

ECR.1 FIFOFS — (FIFO Full Status) сигнализирует о заполнении буфера (при FIFOFS=1 в 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 или переопустошение).

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

По интерфейсу с программой ECP-порт напоминает EPP: после установки режима (записью кода в регистр ECR) обмен данными с устройством сводится к операциям чтения или записи в соответствующие регистры. За состоянием (заполнением) FIFO-буфера наблюдают либо по опросу (чтением регистра 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.

Регистровая модель для ECP порта подобна модели для стандартного параллельного порта, но в ней используется преимущество одной особенности архитектуры шины ISA. В IBM совместимой архитектуре PC используются только первые 1024 адреса портов ввода-вывода. Это — базовое пространство ввода-вывода от 0x000h до 0x3ffh. Чтобы адресовать этот диапазон, необходимо только 10 битов адреса (AD0-9). Для уменьшения стоимости в старых PC использовались и декодировались только эти биты адреса на ISA шине и поэтому доступное пространство ввода-вывода ограничивалось этими 1024 регистрами. В более новых PC фактически используется и декодируется большее количество битов адреса, позволяя увеличить доступное пространство ввода-вывода. Это создает многократные «страницы» первого 1КБ портов ввода-вывода. Программный драйвер может получать доступ к этим страницам, добавляя 1024 (0x400h hex) к базовому адресу, по которому обращаются. Поэтому обращение к адресам 0x378h и 0x778h дает доступ к двум регистрам на двух отдельных страницах, но гарантирует отсутствие конфликтов с любым другим установленным ISA устройством. Преимущество со-

стоит в том, что, используя этот эффект «совмещения имен», новые платы могут «скрывать» регистры, расширяя таким образом доступное число регистров и поддерживая совместимость со старыми ISA платами, которые декодируют только 10 битов адреса.

Регистровая модель ECP использует это преимущество и определяет 6 регистров, которые фактически требуют только 3 порта ввода-вывода. В таблице 9 описаны эти регистры. Обратите внимание, что определение регистра может зависеть от текущего режима работы. Регистр ECR — наиболее важный аспект этой регистровой конфигурации. Это — регистр, который используется для задания текущего режима. Кроме того, этот регистр может использоваться программным обеспечением, чтобы определить, установлен ли ECP-совместимый порт в PC. Программное обеспечение для обнаружения может пытаться получить доступ к любым регистрам ECR, добавляя 0x402h к базовым адресам LPT портов, идентифицированных в таблицах портов LPT BIOS.

8.8 Согласование режимов IEEE 1284

В предыдущих разделах описаны все режимы IEEE 1284-совместимого интерфейса. Периферийному устройству не надо работать во всех режимах. Переговоры необходимы для ведущего устройства, чтобы определить возможности подключенной периферии и иметь метод установки интерфейса в один из режимов.

Для удовлетворения этой потребности была разработана концепция переговоров. Переговоры — это последовательность событий на параллельном интерфейсе, которая не влияет на старые устройства, но обеспечит идентификацию 1284 периферии. Они основаны на том, что старое устройство не будет отвечать на переговоры и поэтому ведущий останется в совместимом режиме, в то время как 1284 периферия ответит на последовательность, и тогда может быть установлен любой требуемый режим, поддерживаемый ведущим и периферией.

Во время переговоров ведущий помещает запрос на линиях данных и затем начинает последовательность переговоров. Запрос может быть предназначен для того, чтобы установить интерфейс в специфический режим или запросить идентификатор периферийного устройства. Идентификация устройства будет об-

суждена позже. На рис. 8.8 показана основная последовательность переговоров.

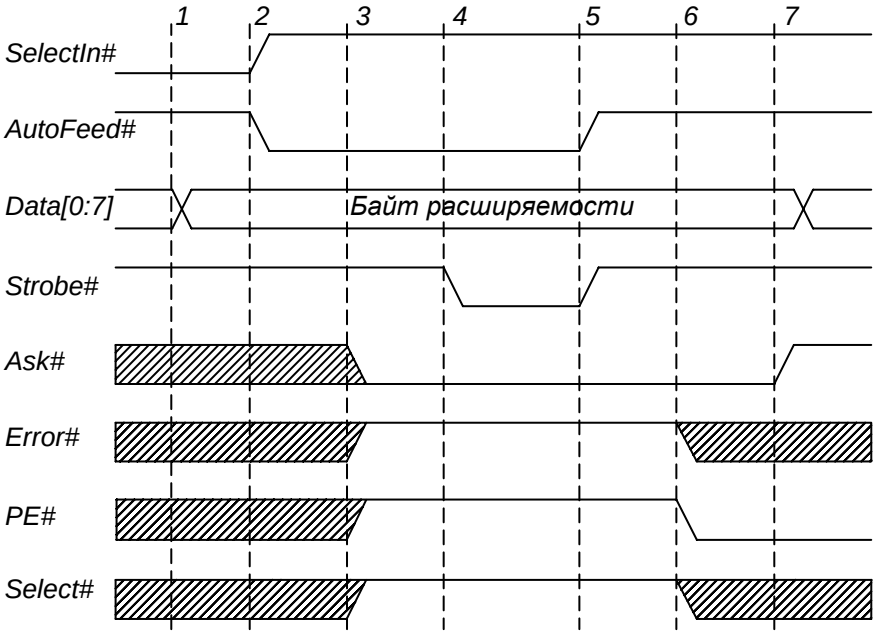


Рис. 8.8 — Основная последовательность переговоров

Байт Расширяемости используется в течение переговоров, чтобы попросить периферию войти в определенный режим передачи, или запросить у периферии идентификатор устройства. Идентификатор устройства может быть возвращен в любом обратном режиме канала, кроме EPP. В табл. 6.9 описан байт расширяемости и допустимые величины. XFlag используется периферией для подтверждения, что требуемый режим доступен. На XFlag будет всегда устанавливаться высокий уровень (№ 6 на рисунке 8.8) как положительное подтверждение для всех запросов, кроме обратной передачи в режиме Тетрады. Всем 1284-совместимым устройствам надо поддерживать режим Тетрады для обратной передачи. Бит запроса расширяемости связи используется, чтобы обеспечить механизм для будущего расширения и дополнения новых эксплуатационных режимов и особенностей.

Переговоры и идентификатор устройства — ключевые особенности будущей способности платформ PC определять конфигурацию системы и устанавливать подключенную к параллельному порту периферию согласно этому определение.

Таблица 8.9 — Значения битов байта расширяемости

Бит	***Описание***	***Допустимые величины бита (8765 4321)***
8	Запрос расширяемости связи	1000 0000
7	Запрос режима EPP	0100 0000
6	Запрос режима ECP с RLE	0011 0000
5	Запрос режима ECP без RLE	0001 0000
4	Зарезервирован	0000 1000
3	Запрос идентификатора устройства	
	Режим тетрады	0000 0100
	Режим байта	0000 0101
	Режим ECP без RLE	0001 0100
	Режим ECP с RLE	0011 0100
2	Зарезервирован	0000 0010
1	Режим байта	0000 0001
Нет	Режим тетрады	0000 0000

Фазовые переходы переговоров 1284

1. Ведущий помещает требуемый байт расширяемости на линии данных

2. Затем ведущий устанавливает на nSelectIn высокий уровень и на nAutoFeed низкий, сообщая о последовательности переговоров.

3. 1284 периферия ответит, установив на nAck низкий уровень, а на nError, PE и Select — высокий. Несовместимая с 1284 периферия не будет отвечать.

4. Ведущий устанавливает на nStrobe низкий уровень. Это используется для стробирования байта Расширяемости в периферии.

5. Тогда ведущий устанавливает на nStrobe и nAutoFeed высокий уровень, сообщая периферии, что она распознана как 1284 устройство.

6. Периферия отвечает, устанавливая на PE и nError низкий уровень, если периферия может передавать данные в обратном направлении, и на Select высокий уровень, если требуемый режим доступен, или на Select низкий уровень, если требуемый режим не доступен.

7. Теперь периферия устанавливает на nAck высокий уровень, сообщая, что последовательность переговоров закончена, и сигнальные линии находятся в состоянии, совместимом с запрошенным режимом.

8.9 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. Таким образом обеспечивается возможность согласования аппаратуры и программного обеспечения и блокирования ложных переключений, вызванных некорректными действиями программы.

Способ и возможности конфигурирования LPT-портов зависят от его исполнения и местоположения. Порт, расположенный на плате расширения (обычно на мультикарте), устанавливаемой в слот ISA или ISA+VLB, обычно конфигурируется джамперами на самой плате. Порт, расположенный на системной плате, обычно конфигурируется через BIOS Setup.

Конфигурированию подлежат следующие параметры:

- Базовый адрес, который может иметь значение 3BCh, 378h и 278h. При инициализации BIOS проверяет наличие портов по адресам именно в этом порядке и, соответственно, присваивает обнаруженным портам логические имена LPT1, LPT2, LPT3. Адрес 3BCh имеет адаптер порта, расположенный на плате MDA или HGC. Большинство портов по умолчанию конфигурируется на адрес 378h и может переключаться на 278h.
- Используемая линия запроса прерывания. Для LPT1 обычно используется IRQ7, для LPT2 — IRQ5. Во многих «настольных» применениях прерывания от принтера не используются, и этот дефицитный ресурс PC можно сэкономить. Однако при использовании скоростных режимов ECP (или Fast Centronics) работа по прерываниям может заметно повысить производительность и снизить загрузку процессора.

- Использование канала DMA для режимов ECP и Fast Centronics — разрешение и номер канала DMA.

Режим работы порта может быть задан в следующих вариантах:

- SPP — порт работает только в стандартном однонаправленном программно-управляемом режиме.
- PS/2, он же Bi-Directional — отличается от SPP возможностью реверса канала (с помощью установки 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 режима 100 не гарантируется.
- ECP+EPP — то же, что и ECP, но запись в ECR кода режима 100 переводит порт в режим EPP.

Выбор режима EPP, ECP или Fast Centronics в BIOS Setup или джамперами на плате сам по себе не приводит к повышению быстродействия обмена с подключенной периферией, а только дает возможность драйверу и периферийному устройству установить оптимальный режим в пределах их «умений». Однако большинство современных драйверов и приложений автоматически пытаются использовать эффективные режимы, так что «подрезать им крылья» установкой простых режимов без веских оснований не стоит.

Принтеры и сканеры могут пожелать использования режима ECP, причем Windows (3.1x, 95 и NT) имеет и 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, обычно могут использовать режим EPP. Для этого режима специальный драйвер пока еще не применяется, а возможность использования EPP обычно включена в драйвер самого подключаемого устройства.

8.10 Использование параллельных портов

Наиболее распространенным применением LPT-порта является, естественно, подключение принтера. Не вдаваясь в проблемы установки и использования программных драйверов, остановимся на аппаратных аспектах — режиме порта и кабеле подключения. Практически все принтеры могут работать с портом в режиме SPP, но применение расширенных режимов дает дополнительные преимущества:

- Двунаправленный режим (Bi-Di) не повышает производительность, но дает дополнительные возможности для сообщения состояния и параметров принтера. Скоростные режимы (Fast Centronics) существенно повышают производительность практически любого принтера (особенно лазерного), но могут потребовать более качественного кабеля (см. ниже). От принтера этот режим не требует каких-либо дополнительных «интеллектуальных» способностей.
- Режим ECP потенциально самый эффективный, и он имеет системную поддержку во всех вариантах Windows. Однако он реализует свои способности (включая аппаратную компрессию) не на всех принтерах. Из распространенных семейств ECP поддерживают принтеры HP DeskJet моделей бxx, LaserJet начиная с 4-го, современные модели фирмы Lexmark. Требует применения кабеля, по частотным свойствам соответствующего IEEE 1284.

Простейший вариант кабеля подключения принтера — 18-проводный кабель с неперевитыми проводами с успехом может использоваться для работы порта в режиме SPP. При длине кабеля более 2 м желательно, чтобы хотя бы линии Strobe# и Busy были перевиты с отдельными «общими» проводами. Однако для скоростных режимов он может оказаться непригодным, причем сбои могут происходить нерегулярно и лишь при определенных последовательностях передаваемых кодов. Иногда попадаются кабели Centronics, у которых отсутствует связь контакта 17 разъема PC с контактом 36 разъема принтера. Для обычных принтеров, работающих в режиме SPP, ее отсутствие малозаметно. Однако при попытке подключения таким кабелем принтера, работающего в стандарте 1284, появится сообщение о необходимости применения «двунаправленного кабеля». Без указанной цепи

принтер не сможет сообщить системе о поддержке расширенных режимов, на что рассчитывают его программные драйверы.

Неплохие электрические свойства имеют ленточные кабели, у которых сигнальные цепи чередуются с общими проводами (хотя бы для управляющих сигналов). Но их применение в качестве внешнего интерфейса не очень практично (они не имеют второго защитного слоя изоляции, поэтому весьма уязвимы) и не эстетично (круглые кабели смотрятся лучше).

Идеальным вариантом являются кабели, в которых все сигнальные линии перевиты с общими проводами и заключены в общий экран — то, что требует IEEE 1248. Такие кабели гарантированно работают на скоростях до ***2 Мбайт/с***, и допускается их длина до 10 метров.

В табл. 8.10 приводится распайка кабеля подключения принтера с разъемом XI типа A (DB-25P) со стороны PC и X2 типа B (Centronics-36) или типа C (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 A и B для этого недостаточно контактов (см. табл. 6.10, где в скобках указаны номера контактов разъема PC типа A, которым соответствуют обратные провода). В разъеме типа C обратный провод (GND) имеется для каждой сигнальной цепи, и сигнальным контактам с номерами 1-17 соответствуют контакты GND 19-35.

Таблица 8.10 — **Кабель подключения принтера**

X1, разъем PC типа A	***Сигнал***	***X2, разъем PRN типа B***	***X2, разъем PRN типа C***
1	-Strobe#	1	15
2	Data 0	2	6
3	Data l	3	7
4	Data 2	4	8
5	Data 3	5	9
6	Data 4	6	10
7	Data 5	7	11
8	Data б	8	12

Окончание 8.10

X1, разъем PC типа A	*Сигнал*	*X2, разъем PRN типа B*	*X2, разъем PRN типа C*
9	Data?	9	13
10	Ack#	10	3
11	Busy	11	1
12	PaperEnd	12	5
13	Select	13	2
14	AutoLFfl	14	17
15	Error#	32	4
16	Init#	31	14
17	SlctIn#	36	16
18	GND(l)	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(10 12 13)	28	20 21 23
25	GND (14 16 17)	30	32 34 35

Для связи двух компьютеров по параллельному интерфейсу применяются различные варианты кабелей, зависящие от режимов используемых портов. Самый простой способ (и самый медленный обмен) обеспечивает режим полубайтного обмена Nibble Mode, работающий на всех (исправных) портах. Для этого режима в кабеле достаточно иметь 10 сигнальных и один общий провод. Распайка разъемов кабеля связи, пригодного для данного режима, приведена в табл. 8.11. Связь двух PC данным кабелем поддерживается стандартным ПО типа MS-DOS Interlnk или Norton Commander.

Таблица 8.11 — **Кабель связи PC-PC (4-битный) Разъемы X1 и X2 — DB25-P (вилки)**

XI, разъем PC#1		*X2, разъем PCЯ2*	
Бит	***Контакт***	***Контакт***	***Бит***
DR.O	2	15	SR.3
DR.1	3	13	SR.4
DR.2	4	12	SR.5

Окончание табл. 8.11

XI, разъем PC#1		*X2, разъем PCЯ2*	
Бит	***Контакт***	***Контакт***	***Бит***
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.0
GND	18—25	18—25	GND

Специально для машин PS/2 с двунаправленным портом фирма IBM выпускала переходное устройство в комплекте с программной поддержкой Data Migration Facility. Переходник устанавливался на разъем LPT-порта PS/2, и к его разъему X2 типа Centronics присоединялся обычный принтерный кабель, подключенный к LPT-порту любого ПК. Таким образом, предлагалось решить проблему миграции файлов со старых компьютеров, оснащенных только 5″ дисководами, на тогда еще новомодные PS/2 с дисководами 3,5″. Распайки такого переходника приведены в табл. 8.12. Как видно из сравнения с предыдущей таблицей, данный переходник (весьма солидно исполненный) нельзя использовать при связи с помощью программ DOS Interlnk и Norton Commander. Если обе соединяемые машины имеют двунаправленные порты, переходник обеспечивает симметричную двунаправленную связь. По скорости обмена такое соединение превосходит вышеописанное полубайтное в 2 раза. Однако это соединение не соответствует двунаправленному режиму IEEE 1284 (Byte Mode).

Таблица 8.12 — **Переходник Data Migration для IBM PS/2. Разъемы X1 — DB25-P (вилка), X2 — Centronics-36 (розетка)**

X1		*X2*	
Контакт	***Бит***	***Бит***	***Контакт***
1	CR.O\	SR.6	10
2	DR.0	DR.0	2
3	DR.1	DR.1	3
4	DR.2	DR.2	4

Окончание табл. 8.12

X1		*X2*	
Контакт	***Бит***	***Бит***	***Контакт***
5	DR.3	DR.3	5
6	DR.4	DR.4	6
7	DR.5	DR.5	7
8	DR.6	DR.6	8
9	DR.7	DR.7	9
10	SR.6	CR.O\	1
12	SR.5	CR.3\	36
17	CR.3\	SR.5	12
18—25	GND	GND	19—30 33

Высокоскоростная связь двух компьютеров может выполняться и в режиме ECP (режим EPP для этих целей неудобен, поскольку он требует синхронизации шинных циклов ввода/вывода двух компьютеров). В табл. 8.13 приведена распайка кабеля для этого режима. В отличие от предыдущих таблиц, описывающих кабели для программно-управляемых режимов, в ней в качестве вспомогательной информации приведены имена сигналов, которые аппаратно генерируются адаптерами портов. Этот же кабель может использоваться и для связи в режиме Byte Mode (при наличии двунаправленных портов). Связь компьютеров с помощью такого кабеля поддерживается Windows 95.

Таблица 8.13 — **Кабель связи PC-PC в режиме ECP и Byte Mode**

Разъем X1		*Разъем X2*	
Контакт	***Имя в ECP***	***Имя в ECP***	***Контакт***
1	HostClk	PeriphClk	10
14	HostAck	PeriphAck	11
17	1284Active	+PeriphRequest#	15
16	ReverseRequest#	AckReverse#	12
10	PeriphCLk	HostClk	1
11	PeriphAck	HostAck	14
12	AckReverse#	ReverseRequest#	16
13	Xflag	–	–
15	PeriphRequest#	1284Active	17
2—9	Data[0:7]	Data[0:7]	2—9

Для удобства анализа различных соединений в табл. 8.14 сведено назначение выводов разъема LPT-порта в различных режимах и их соответствие битам регистров стандартного порта.

Таблица 8.14 — **Назначение выводов разъема LPT и бит регистров в режимах SPP, ECP и EPP**

Контакт	***I/O***	***Бит***	***SPP***	***ECP***	***EPP***
1	O/I	CR.O\	Strobe#	HostClk	Write#
2	O/I	DR.O	DataO	Data 0	Data 0
3	O/I	DR.l	Data 1	Data 1	Data 1
4	O/I	DR.2	Data 2	Data 2	Data 2
5	O/I	DR.3	Data 3	Data 3	Data 3
6	O/I	DR.4	Data 4	Data 4	Data 4
7	O/I	DR.5	Data 5	Data 5	Data 5
8	O/I	DR.6	Data 6	Data 6	Data 6
9	O/I	DR.7	Data 7	Data 7	Data 7
10	I	SR.6	Ack#	PeriphClk	INTR#
11	I	SR.7\	Busy	PeriphAck	Wait#
12	I	SR.5	PaperEnd	AckReverse#	user defined
13	I	SR.4	Select	Xflag	user defined
14	O/I	CR.1\	AutoLF#	HostAck	DataStb#
15	I	SR.3	Error#	PeriphRequest#	user defined
16	O/I	CR.2	Init#	ReverseRequest#	Reset#
17	O/I	CR.3\	SelectIn#	1284Active	AddrStb#

* Символом «\» отмечены инвертированные сигналы (1 в регистре соответствует низкому уровню линии).

Подключение сканера к LPT-порту эффективно, только если порт обеспечивает хотя бы двунаправленный режим (Bi-Di), поскольку в основном здесь используется ввод. Но лучше использовать порт ECP, если этот режим поддерживается сканером (или EPP, что маловероятно).

Подключение внешних накопителей (Iomega Zip Drive, CD-ROM), адаптеров ЛВС и других симметричных устройств ввода/вывода имеет общую специфику. Большинство таких устройств способно работать в любом из режимов порта (обычно исключая ECP), что обеспечивает их неограниченное применение на любых компьютерах. Но используя режим SPP, кроме весьма

небыстрой работы устройства, заметна принципиальная асимметрия этого режима: чтение данных (по сети или с дискового накопителя) будет в два раза медленнее, чем весьма небыстрая запись. Применение двунаправленного режима (Bi-Di или PS/2 Type 1) устранит эту асимметрию — скорости сравняются. Но только перейдя на EPP, можно получить удовлетворение от нормальной скорости работы, однако это почему-то не всегда подчеркивается в документации на устройства. В режиме EPP подключение к LPT-порту почти не уступает подключению аналогичного устройства через ISA-контроллер. Это же справедливо и при подключении устройств со стандартным интерфейсом шин к LPT-портам через преобразователи интерфейсов (например, LPT — IDE, LPT — SCSI, LPT — PCMCIA).

8.11 Физический и электрический интерфейсы

Стандарт IEEE 1284 определяет *физические характеристики приемников и передатчиков сигналов*. 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_O, измеренный на разъеме, должен составлять 50±5 Ом на уровне V_{OH}-V_{OL}. Для обеспечения заданно-

го 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.

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, фирма 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. 8.9.

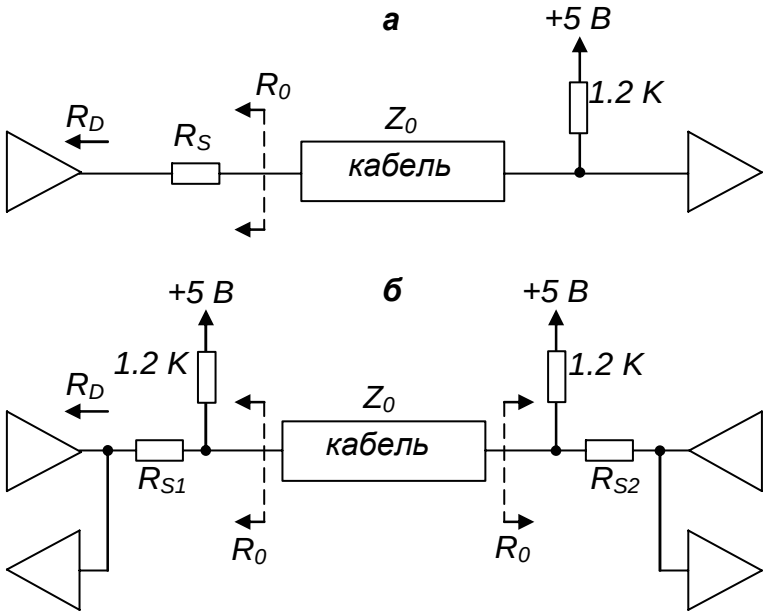


Рис. 8.9 — Оконечные цепи линий интерфейса IEEE 1284:
а — однонаправленных, *б* — двунаправленных

Стандарт IEEE 1284 определяет три типа используемых разъемов. Типы A (DB-25) и B (Centronics-36) используются в традиционных кабелях подключения принтера, тип C новый малогабаритный 36-контактный разъем.

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±б 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. 8.15.

Таблица 8.15

Тип	***Расшифровка***	***Разъем 1***	***Разъем 2***
AMAM	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

8.11.1 Соединители по IEEE1284

1284 стандарт идет вне описания новых режимов передачи данных и фактически определяет механический интерфейс и электрические свойства совместимого параллельного порта. Многие из проблем, связанных с подключаемыми к параллельному порту устройствами, являются результатом того, что не имелось никакого стандарта электрического интерфейса для параллельного порта. Розетка DB25 стала стандартом для PC или соединителя ведущего устройства, но имелись много различных реализаций драйверов, резисторов, конденсаторов, и т.д. для электрического интерфейса.

1284 комитет чувствовал, что в первую очередь нужно определить следующие цели:

1. Гарантировать электрическую и механическую совместимость среди всех 1284 совместимых устройств.

2. Гарантировать, что 1284 интерфейс работал бы с существующими периферияими параллельного порта и адаптерами.

3. Гарантировать передачу и целостность данных на самых высоким скоростях.

4. Увеличить расстояние до 10 м (30 футов).

Чтобы выполнить эти задачи, стандарт определяет требования к соединителям, электрическому интерфейсу и кабелю.

Соединители 1284.

Стандарт различает три типа соединителей для 1284 интерфейса:

- 1284 Тип A: 25-контактный DB25;
- 1284 Тип B: 36 контактный.085 centerline Champ connector with bale locks;
- 1284 Тип C: 36-контактный.050 centerline mini connector with clip latches.

Для новых проектов рекомендуется соединитель типа C. Он меньше, чем предыдущие соединители, имеет простой в использовании замок с зажимами для удержания кабеля и обеспечивает самый легкий монтаж кабеля с оптимальными электрическими свойствами. Кроме того, кабель, собранный с этим соединителем, предусматривает еще два сигнала. Эти сигналы — Peripheral Logic High и Host Logic High. Они могут использоваться, чтобы

определить, включено ли устройство на другом конце кабеля. Это даёт некоторую степень интеллектуального управления для 1284 интерфейсов.

9 ПОСЛЕДОВАТЕЛЬНЫЕ ИНТЕРФЕЙСЫ

Последовательный интерфейс предназначен для передачи данных и использует одну сигнальную линию для передачи в одном направлении. Информационные биты передаются последовательно друг за другом, отсюда и название.

Последовательная передача данных может осуществляться в синхронном и асинхронном режимах. При ***синхронном*** обмене передатчик и приемник синхронизируются по выполняемым операциям при помощи дополнительной линии синхронизации. Возможно также применение самосинхронизирующихся методов кодирования данных (NRZ код, Манчестерский код), при использовании которых синхросигнал выделяется приемником из потока принимаемых данных. В этом режиме данные передаются единым блоком без промежутков между отдельными байтами. Режим применяется при необходимости передать большое количество информации с большой скоростью.

Асинхронный режим используется при передаче отдельных байтов, причем промежуток времени между передачей соседних байтов может быть произвольным.

При асинхронном обмене информационные байты оформляются в слова, формат которых зависит от текущей настройки порта:

- Стартовый бит — сигнализирует приемнику о начале передачи слова.
- Информационная часть — содержит 5—8 бит передаваемого байта, (младший бит передается первым).
- Необязательный бит паритета (четности) — используется для проверки правильности данных на приемной стороне.
- Стоповые биты — сообщают приемнику о конце передачи, их количество программируется и может быть настроено на 1, 2 или полтора стоповых бита.

Очевидно, что при синхронном способе обмене можно добиться больших скоростей, т.к. не требуется предавать старт-бит, стоп-бит, бит паритета. В асинхронном режиме на каждые 8 бит данных приходится не 3—4 служебных бита. Асинхронный обмен в ПК реализуется обычно по ***протоколу RS-232C***. Стандарт EIA (Electronics Industries Association) введен в 1962 г.

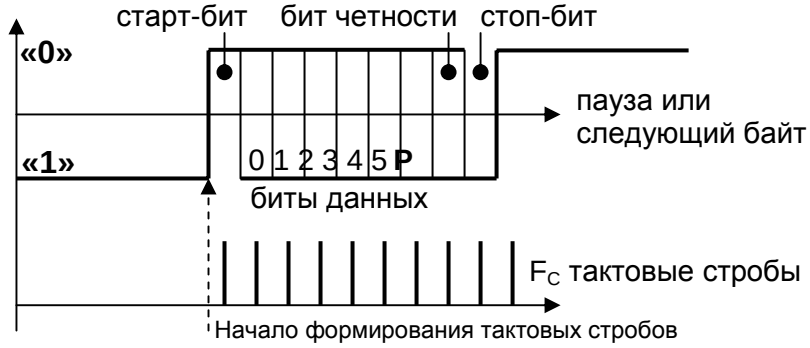


Рис. 9.1 — Последовательный интерфейс.
Асинхронный режим передачи

Старт-бит следующей посылки может передаваться в любой момент времени, начиная с момента окончания стоп-бита предыдущей посылки, таким образом, между передачами могут быть паузы произвольной (и неограниченной) длительности.

Старт-бит позволяет организовать простую синхронизацию приемника по сигналу от передатчика. При этом подразумевается, что приемник и передатчик работают на одной и той же скорости обмена (т.е. имеют одну и ту же тактовую частоту), измеряемой количеством передаваемых бит в секунду. Внутренний генератор синхронизации приемника содержит счетчик-делитель частоты, обнуляемый в момент начала старт-бита от передатчика. Этот счетчик формирует внутренние стробы Fc, по которым приемник фиксирует принимаемые биты. В идеальном случае, эти внутренние стробы должны приходиться на середину битовых интервалов, однако, из-за разностей скоростей приемника и передатчика возникает накапливающееся рассогласование. Для безошибочного приема 8 бит данных, 1 бита паритета и стоп-бита предельное рассогласование не может превышать 5 % на бит. Очевидно, что чем больше скорость передачи, тем строже требования к синхронизации приемника и передатчика, кроме того, с ростом скорости обмена усиливается влияние фронтов импульсов.

Для асинхронного режима принят ряд стандартных скоростей обмена: 50, 75, 110, 150, 300, 600, 1200, 2400, 4800, 9600, 19200, 38400, 57600 и 115200 бит/сек.

Таким образом, для установления связи при асинхронном режиме необходимо установить одинаковыми следующие пара-

метры приемника и передатчика: скорость обмена, количество бит данных, наличие бита паритета, тип паритета (четность-нечетность), длину стоп-бита.

9.1 Управление потоком данных

Сигналы линий самого популярного из последовательных интерфейсов — интерфейса RS-232C в стандартных разъемах DB-25S и DB-9S приведены в таблице 9.1.

Таблица 9.1

Сигнал	***DB25S***	***DB9S***	***Назначение***
PG	1	–	(*Frame Ground*) Защитная земля. Соединяется с корпусом устройства и экраном кабеля
SG	7	5	(*Signal Ground*) — Сигнальная земля (относительно нее действуют линии сигналов)
TxD	2	3	(*Transmitted Data*) — Передаваемые данные (по этой линии DTE отправляет к DCE данные). Выход передатчика
RxD	3	2	(*Received Data*) — Принимаемые данные (по этой линии DTE получает от DCE данные). Вход приемника.
RTS	4	7	(*Request To Send*) — Запрос для передачи (DTE указывает, что оно готово начать передачу). Состояние ON уведомляет модем о том, что у терминала есть данные для передачи. В полудуплексе — переключение модема в режим передачи
CTS	5	8	(*Clear To Send*) — Сброс для передачи (DCE сообщает, что оно готово принимать данные от DTE). Вход разрешения передачи данных терминалу. Состояние OFF аппаратно запрещает передачу данных. Применяется для аппаратного управления потоками данных
DTR	20	4	(*Data Terminal Ready*) — Готовность выходных данных (при помощи этого сигнала DTE указывает DCE что он готов к проведению сеанса связи и что DCE должно выполнить необходимую коммутацию)
DSR	6	6	(*Data Set Ready*) — Готовность данных (этим сигналом DCE сообщает своему DTE, что оно

Окончание табл. 9.1

Сигнал	***DB25S***	***DB9S***	***Назначение***
			может осуществить связь с ним). Вход сигнала готовности от аппаратуры передачи данных
DCD	8	1	(*Data Carrier Detect*) — Детектор обнаружения несущей (при помощи этого сигнала DCE сообщает своему DTE, что связь с DCE на другом конце линии установлена)
RI	22	9	(*Ring Indicator*) — Индикатор вызова (DCE сообщает DTE что на линии связи находится устройство, желающее осуществить обмен). Вход индикатора вызова (звонка)

В общем случае на ***симплексном асинхронном канале*** процедура передачи в одну сторону от DTE к DCE будет выглядеть так:

1) приемник и передатчик выставляют активные сигналы готовности на выходах (DTR для передатчика и DSR для приемника);

2) оба проверяют наличие готовности партнера на своих входах (передатчик проверяет DSR, а приемник — DTR);

3) передатчик, при готовности приемника и имеющихся для передачи данных, выставляет запрос передачи данных (RTS);

4) приемник, при готовности передатчика, по принятому сигналу (RTS) выставляет разрешение передачи данных (CTS);

5) по принятому передатчиком сигналу разрешения передачи данные передаются;

6) при необходимости прекратить передачу данных, приемник в произвольное время снимает разрешение передачи данных (CTS), при этом передатчик либо обрывает передачу очередного байта (если на момент снятия разрешения передано менее половины бит) — в этом случае байт считается непереданным, либо продолжает передачу до конца байта, если передано более половины бит — в этом случае байт считается успешно переданным;

7) при окончании данных передатчик снимает сигнал запроса передачи (RTS), с этого момента любые данные, принятые приемником, считаются недостоверными. В некоторых случаях передатчик может снять запрос передачи данных, не дожидаясь конца собственной передачи.

На ***полудуплексном канале*** принято, что запрос от DTE на передачу есть всегда, а сигнал RTS включает передачу в обратном направлении — от DCE к DTE, при этом также считается, что DTE всегда готово принять данные от DCE. (В полном стандарте, вообще-то, есть линии и для обратного квитирования). В остальном процедура аналогичная. Для двух DTE (нуль-модемное соединение) правило квитирования сигналов следующее:

1. Выход готовности одного идет на вход готовности другого (DTR → DSR). Этот же сигнал нужно пустить на вход DCD второго устройства. Для сокращения числа проводников можно эти сигналы между аппаратами не гонять, и собственный DTR замкнуть на собственные же DSR_DCD (для каждого устройства). При этом роль сигнала готовности будет неявно выполнять сигнал разрешения на передачу — CTS.

2. Выход передатчика одного устройства идет на вход приемника другого (TxD → RxD).

3. Сигнал RTS одного аппарата идет на CTS другого. Здесь RTS используется как бы не по назначению — не как запрос передачи от себя к партнеру, а как выход разрешения передачи от партнера к себе. Но, поскольку этот сигнал формируется в АПД не аппаратно, а программно, проблем не возникает — коммуникационная программа использует его как надо.

Для управления потоком данных в полнодуплексной линии связи могут использоваться два вида протоколов — аппаратный и программный.

- ***Аппаратный протокол управления RTS/CTS***. Использует сигнал CTS, который позволяет остановить передачу данных, если приемник не готов к работе. Передача данных по этому протоколу показана на рис. 9.2. Байт, передаваемый на момент прихода CTS, будет передан, однако с момента окончания его передачи передатчик переходит к ожиданию готовности приемника (т.е. снятия CTS).

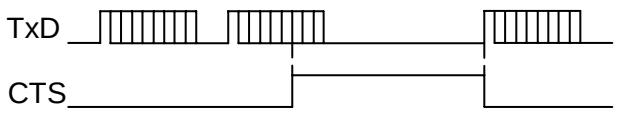


Рис. 9.2

Протокол RTS/CTS обеспечивает самую быструю реакцию передатчика на состояние приемника, позволяет организовать обмен, не прибегая к буферизации. Часто используется в принтерах и для соединения компьютеров. В случае с принтером линия CTS ПК должна соединяться с линией RTS принтера, при соединении двух ПК необходимо перекрестное соединение CTS-RTS. Если аппаратный протокол обмена не используется, то на линию CTS ПК необходимо подать сигнал «включено», что обычно достигается соединением CTS ПК с его же RTS перемычкой на разъеме. Аппаратный обмен невозможен через минимальный нуль-модемный кабель.

- ***Аппаратный протокол DTR/DSR***. Аналогичен протоколу RTS/CTS, но использует другую пару сигналов (в данном случае «готовность приемника»).
- ***Программный протокол XON/XOFF***. Предполагает наличие двунаправленного канала обмена. Временные диаграммы обмена показаны на рис. 9.3. Предполагает наличие у приемника буфера, так как время реакции передатчика tp может оказаться достаточно большим. Когда буфер приемника заполняется до определенного уровня (обычно 80—90 %), он передает передатчику команду XOFF (байт с кодом 13h). Приняв эту команду, передатчик прекращает передачу и переходит в состояние ожидания до прихода команды XON (байт с кодом 11h), по которому передатчик возобновляет передачу.

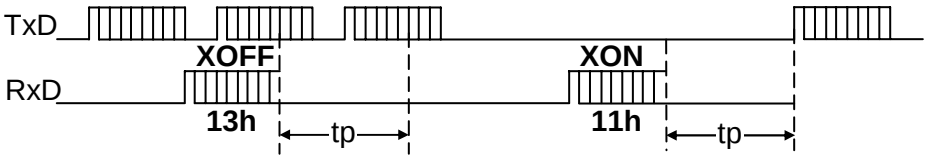


Рис. 9.3

- ***Программный протокол ACK***. При обмене по этому протоколу для получения очередного байта приемник посылает передатчику команду ACK (байт с кодом 6h). В ответ передатчик посылает приемнику один байт (или пакет байт определенного размера).

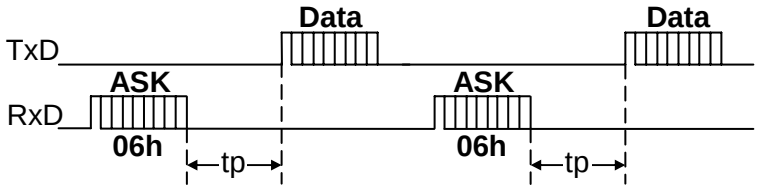


Рис. 9.4

9.2 Физические реализации последовательного интерфейса

Последовательный ***асинхронный адаптер*** представляет собой отдельный модуль или расположен на материнской плате. Каждый адаптер может содержать несколько *COM-портов*, через которые к компьютеру подключаются внешние устройства (каждый COM-порт имеет свой номер от 1 до 4).

Каждому порту соответствует несколько регистров для организации взаимодействия с ним и линия IRQ для реализации механизма работы по аппаратным прерываниям (иногда несколько портов используют одну линию запроса прерывания).

В каждый момент времени в обмене информацией (как в приеме, так и в передаче) участвуют два устройства:

- Терминальное устройство (*Data Terminal Equipment* — ***DTE***) — конечное устройство, выдающее или принимающее информацию — ***адаптер последовательного интерфейса*** компьютера или терминала. Это устройство осуществляет преобразование параллельного кода в последовательный и наоборот.
- Устройство связи (*Data Communication Equipment* — ***DCE***) — промежуточное устройство, преобразующее сигнал в форму, удобную для транспортировки (производит модуляцию-демодуляцию сигнала — ***модем***) или периферийное оборудование (принтеры, графопостроители и т.п.). В целом линия связи выглядит так:

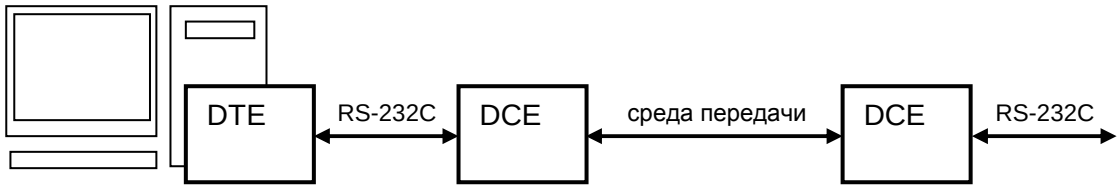
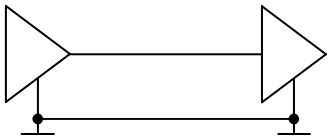
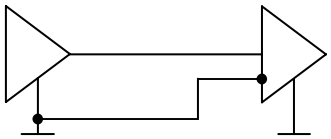
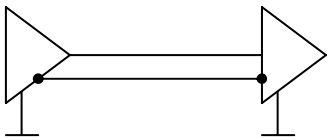
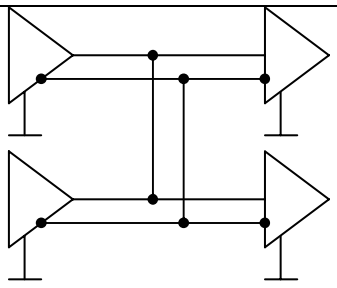


Рис. 9.5 — Последовательный интерфейс. Линия связи

При малом расстоянии между DTE можно организовать связь между ними напрямую (без DCE). Для осуществления такой связи применяется специальный кабель, называемый нуль-модем.

Существует несколько родственных последовательных интерфейсов: RS-232C, RS-423A, RS-422A, RS-485A. Характеристики этих интерфейсов сведены в таблицу 9.2.

Таблица 9.2

RS-232C		Дуплекс, L=15 м, V=20 Кбит/с
RS-423A		Дуплекс, L=9 м, V=100 Кбит/с L=91 м, V=10 Кбит/с L=1200 м, V=1 Кбит/с
RS-422A		Дуплекс, L=12 м, V=10 Мбит/с L=120 м, V=1 Мбит/с L=1200 м, V=100 Кбит/с
RS-485A		Полудуплекс, до 32 параллельно соединенных приемо-передатчиков, L=12 м, V=10 Мбит/с L=120 м, V=1 Мбит/с L=1200 м, V=100 Кбит/с

Линии интерфейсов RS-232C и RS-423A несимметричны, имеют самую низкую защищенность от синфазной помехи. RS-423A имеет приемник с дифференциальным входом, что несколько повышает его помехозащищенность. Лучшими параметрами обладает симметричный дуплексный интерфейс RS-422A и его полудуплексный магистральный аналог RS-485A. Приемник и передатчик RS-422A и RS-485A имеют дифференциальные входы

и, следовательно, обладают высокой защищенностью от синфазных помех. Интерфейс RS-485A обладает передатчиком повышенной мощности с защитой от короткого замыкания линии, защитой от перегрева при длительной перегрузке и защитой от коллизий (одновременной работы нескольких параллельно включенных передатчиков).

Одним из вариантов последовательного интерфейса является интерфейс типа «токовая петля». В этом интерфейсе сигналом является не уровень напряжения, а ток в двухпроводной линии. Обычно, за единицу принимают ток 20 мА, за ноль — отсутствие тока. В таком варианте интерфейса приемник может распознавать обрыв линии — при обрыве принимаются одни нули и обрыв распознается по отсутствию стоп-битов. Обычно, «токовая петля» предполагает наличие гальванической развязки приемника и передатчика, как правило, выполняемой при помощи оптронов. Питание токовой петли может осуществляться от передатчика (вариант с активным передатчиком) или от приемника (активный приемник). Токовая петля с гальванической развязкой позволяет передавать данные на расстояние до нескольких километров, определяемое уровнем помех и сопротивлением пары проводов. Поскольку интерфейс требует пары проводов для каждого сигнала, то обычно применяют две пары — принимаемые данные и передаваемые данные, а управление потоком ведется по протоколу XON/XOFF. Одним из классических примеров интерфейса «токовая петля» является интерфейс MIDI, применяемый в звуковых картах.

Все перечисленные выше варианты реализации интерфейса являются проводными, однако, существуют и беспроводные варианты, наибольшее распространение среди которых получил инфракрасный (ИК) интерфейс. Большинство ИК интерфейсов работают на расстоянии от одного до нескольких метров на низкой скорости (до 115,2 кбит/с), средней (до 1.152 Мбит/с) или высокой (до 4 Мбит/с) скоростью.

9.3 Физический и электрический интерфейс RS-232C

Интерфейс RS-232C использует несимметричные приемники и передатчики, сигнал передается относительно общего про-

вода (схемной земли). Интерфейс RS-232C не обеспечивает гальванической развязки устройств. Логической единице на входе приемника соответствует уровень напряжения –3...–12 В. Для линий управляющих сигналов это состояние называют «ON», а для линий последовательных данных — «MARK». Логическому «0» соответствует напряжение +3...+12 В (называемое «OFF» или «SPACE», соответственно). Между уровнями +3...–3 В существует зона нечувствительности, обуславливающая гистерезис приемника. Состояние на выходе приемника изменяется только при пересечении напряжением порога +3 или –3 В. На рис. 9.6 показан искаженный помехой сигнал, который принимается без ошибки за счет гистерезиса приемника. Уровни сигналов на выходах передатчика должны лежать в диапазоне +5...+12 В или –5...–12 В.

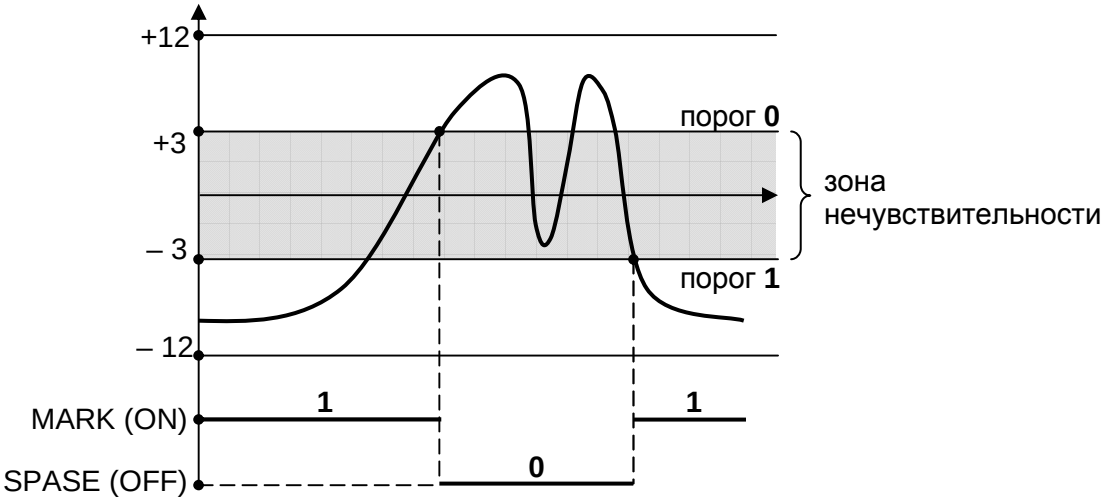


Рис. 9.6 — Принимаемый сигнал

Интерфейс предполагает наличие защитного заземления обоих устройств. Присоединение и отключение устройств с автономным питанием должно производиться при отключенном питании, иначе разность невыровненных потенциалов устройств в момент коммутации может превысить допустимые пределы и вывести из строя микросхемы порта.

При передаче данных на большие расстояния без использования специальной аппаратуры из-за помех, наводимых электромагнитными полями, возможно возникновение ошибок. Специ-

фикации RS-232-C не ограничивают максимальную длину кабеля, но ограничивают максимальное значение его емкости 2500 пФ. Емкость интерфейсных кабелей различна, однако общепринятой длиной считается длина 50 футов (**15,24 м**) (до 20 000 бод). Чем выше скорость передачи, тем больше искажения сигнала, вызванные емкостными характеристиками кабеля. Выпускаются специальные интерфейсные кабели прямой связи RS-232C низкой емкости, которые удовлетворительно работают со скоростью 9600 бод на расстоянии до 500 футов (150 м). Однако на практике это расстояние может быть значительно больше. Оно непосредственно зависит от скорости передачи данных. Согласно McNamara (Technical Aspects of Data Communications, Digital Press, 1982) определены следующие значения:

Таблица 9.3

Скорость передачи в бодах	Максимальная длина (экранированный кабель), м	Максимальная длина (неэкранированный кабель), м
110	1524,0	914,4
300	1524,0	914,4
1200	914,4	914,4
2400	304,8	152,4
4800	304,8	76,2
9600	76,2	76,2

Число подключаемых приемников и передатчиков, подключаемых к одной линии — 1/1, (в отличие от стандартов RS422 — 1/10 или RS485 — 32/32).

Кабели для соединения устройств. Обычно со стороны DTE (например, в ПК) ставят разъем типа «вилка», со стороны DCE (например, в модеме) — разъем типа «розетка». Тип применяемых разъемов — DB-25S или DB-9S. Для подключения модема (или другого DCE) используется такая распайка кабеля, при которой соединяются одноименные контакты, такой кабель называется *прямым*.

Рис. 9.7 — Разъемы DB-25S или DB-9S

Если аппаратура DTE соединяется без модема, то используются так называемые *нуль-модемные кабели*, имеющие на обоих концах розетки. Контакты таких кабелей соединяются перекрестно, причем некоторые линии могут не соединяться друг с другом (так называемый минимальный кабель, недостающие линии эмулируются перемычками на разъемах).

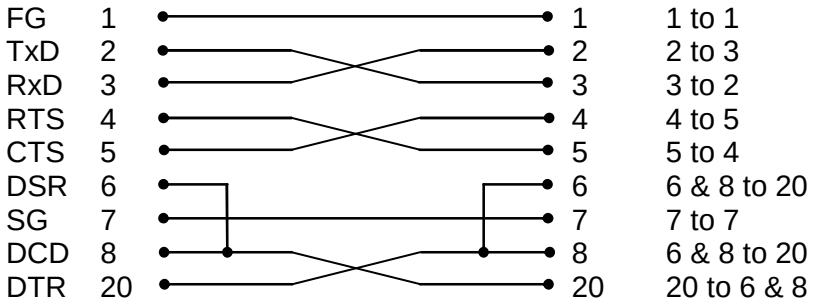


Рис. 9.8 — Разводка нуль-модемного кабеля
для разъемов DB-25S

9.4 Работа с COM портом на низком уровне

Микросхемы асинхронных приемопередатчиков. Преобразование параллельного кода в последовательный при передаче (и последовательного в параллельный при приме), контроль ошибок, формирование запросов прерывания и ряд других сервисных функций осуществляется специализированной микросхемой, называемой UART (Universal Asynchronous Receiver — Transmitter). COM порты IBP PC базируются на микросхемах, совместимых на уровне регистров с UART i8250 — 8250A/ 16450/ 16450A. Каждая из перечисленных микросхем совместима с предыдущей, но не наоборот. Это следует учитывать при составлении программы

управления — если программа ориентирована на i8250, то она будет работать со всеми последующими модификациями UART.

С 16550A совместимо подавляющее большинство микросхем — контроллеров COM парта современных ПК. Особенностью 16550 и 16550A является то, что они имеют дополнительные, по сравнению с 8250, регистры. Многие биты, считающиеся в 8250 резервом, в 16550 задействованы для управления его новыми функциями. Однако все регистры 8250 совпадают с соответствующими регистрами 16550, что обеспечивает совместимость.

Базовый адрес COM порта определяется путем чтения переменных BIOS, адреса которых приведены в табл. 9.4. Стандартным адресом порта COM является 3F8h, COM2 — 2F8h. Перед работой с портом рекомендуется определить его адрес путем чтения переменной BIOS, а не брать стандартное значение.

Таблица 9.4 — Адреса переменных BIOS для COM портов

Имя порта	***Адрес в BIOS***	***Тип переменной***	***Описание***
COM1	0040:0000h	Word	Базовый адрес порта COM1. Если переменная равна 0, то порт не найден
COM2	0040:0002h	Word	
COM3	0040:0004h	Word	
COM4	0040:0006h	Word	

Порты могут вырабатывать аппаратные прерывания **IRQ4** (обычно используются для COM1 и COM3) и **IRQ3** (для COM2 и COM4). Возможно разделяемое использование одной линии запроса несколькими портами (или ее разделения с другими устройствами), что зависит от реализации аппаратного подключения и программного обеспечения.

Структура регистров УАПП 16550A. У контроллера последовательного интерфейса имеется 8 регистров, занимающих смежные адреса в пространстве устройств ввода-вывода (табл. 9.6). Следует обратить особое внимание на то, что регистров реально больше: при чтении регистра 0 байт читается из регистра приемника, при записи байт запишется в регистр передатчика.

Кроме того, регистры со смещением +0 и +1 мультиплексируются битом 7 (**DLAB**) регистра управления (+3). Когда этот бит равен 0 (нормальное рабочее состояние), регистр 0 является регистром приемопередатчика, а регистр 1 задает маску прерываний. Когда этот бит равен 1, то регистры 0 и 1 применяются для чтения/записи старшего и младшего байтов делителя, определяющего скорость обмена. Значение делителя определяется по формуле:

$$K = \frac{115\,200}{V},$$

где V — скорость передачи [бит/с]. Входная частота синхронизации 1.8432 МГц делится адаптером на K и получается 16-ти кратная частота передачи.

Таблица 9.5

Делитель	Скорость передачи в бодах	Делитель	Скорость передачи в бодах
1040	110	24	4800
768	150	12	9600
384	300	6	19200
192	600	3	38400
96	1200	2	57600
48	2400	1	115200

Назначение регистров UART и их адреса относительно базового приведены в таблице 9.6.

Таблица 9.6 — **Регистры UART 16550A**

Смещение	***DLAB***	***R/W***	***Совместимость с i8250***	***Значение***
+0	1	R/W	+–	Регистр скорости обмена (LO)
	0	WO	++	Регистр передатчика
	0	RO	++	Регистр приемника
+1	1	R/W	+–	Регистр скорости обмена (HI)
	0	R/W	+–	Регистр разрешения прерываний
+2		RO	+–	Регистр идентификации прерываний
		WO	–	Регистр управления FIFO

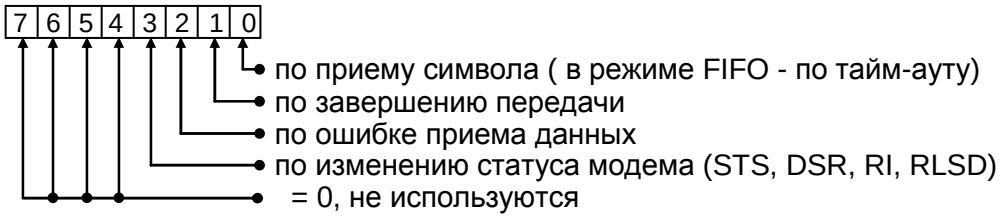
Окончание табл. 9.6

Смещение	*DLAB*	*R/W*	*Совместимость с i8250*	*Значение*
+3		R/W	+	Регистр управления линией (настройка параметров канала)
+4		R/W	+	Регистр управления модемом
+5		R/W	+	Регистр состояния линии
+6		R/W	+	Регистр состояния модема
+7		R/W	–	Временный регистр для хранения данных

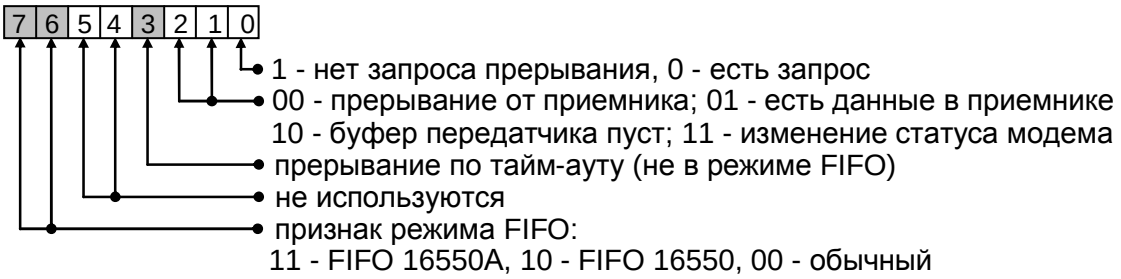
12.4.1 Регистр передаваемых данных (+0, byte, DLAB=0). Для передачи в регистр со смещением +0 COM-порта необходимо записать байт передаваемых данных. После приема данных от внешнего устройства они могут быть прочитаны из этого регистра. В зависимости от состояния бита **DLAB** управляющего слова, выводимого в *регистр управления линией* (со смещением +3), назначение нулевого регистра порта может измениться. Если этот бит равен 0, то порт используется для записи/чтения передаваемых данных.

12.4.2 Регистр скорости обмена (+0 +1, word, DLAB=1). Объединяет два регистра 0 и 1 для хранения старшего (HI) и младшего (LO) байтов делителя частоты тактового генератора, определяющего скорость обмена. Старший байт делителя записывается в первый регистр, а младший — в нулевой. Необходимо, чтобы значение седьмого бита DLAB регистра управления линией (+3) было равно 1.

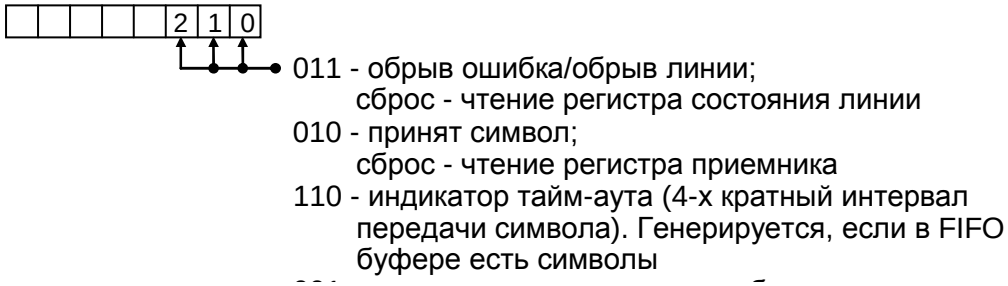
12.4.3 Регистр разрешения прерываний (+1, byte, DLAB=0). Бит DLAB регистра управления линией должен равняться 0. Если работа с портом идет без использования прерываний, то их следует запретить путем записи 0 в этот регистр. Единичное значение каждого бита соответствует разрешению соответствующего прерывания.



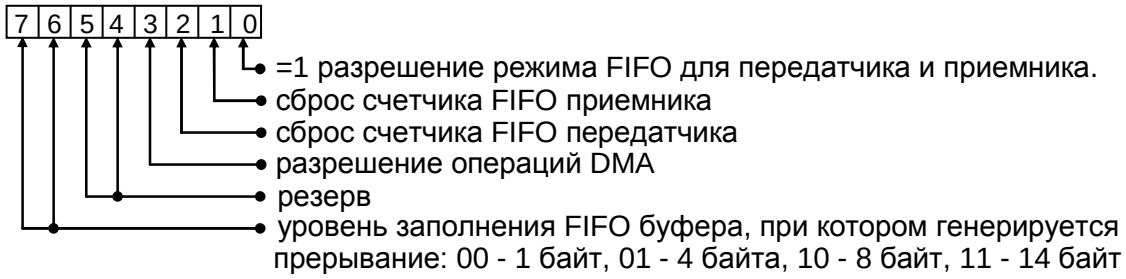
12.4.4 Регистр идентификации прерываний (+2, byte). Ввиду того, что контроллер обслуживается одним прерыванием, а возникать оно может, в общем случае, в 4-х различных ситуациях, процедуре обработки прерывания необходимо выяснить, по какой причине произошло прерывание. Причину прерывания программа может определить по его содержимому этого регистра. Биты 7,6 и 3 не используются в контроллере 8250.



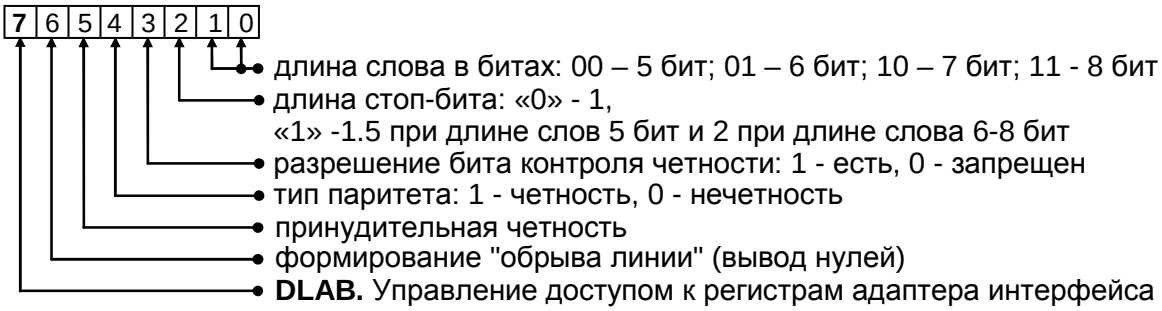
В режиме использования FIFO-буфера биты данного регистра 2-0 имеют иной смысл:



12.4.5 Регистр управления FIFO (+2, byte, только для записи, только 16550+) Для очистки буферов FIFO необходимо запретить и затем снова разрешить режим FIFO. При смене режима FIFO буфер очищается.

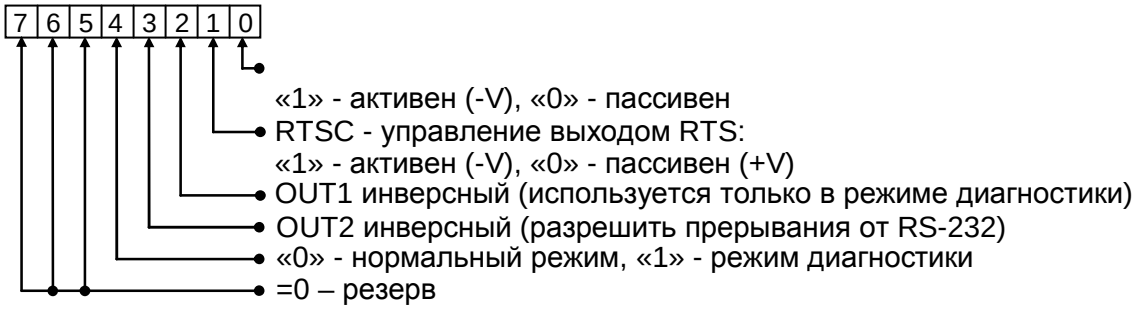


12.4.6 Регистр управления линией (+3, byte). Для задания основных параметров адаптера применяется регистр управления линией. Бит 7 регистра управления линией называется DLAB и предназначен для определения назначения регистров 0 и 1. Он устанавливается в 1 только на время программирования делителя управления скоростью передачи.

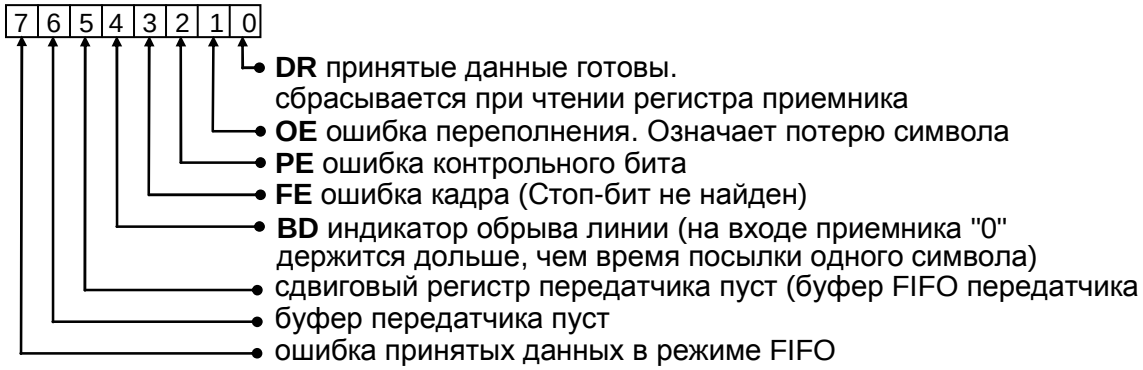


Бит 5 определяет принудительную четность (если «0», то бит паритета генерируется в соответствии с паритетом символа, «1» — всегда равен инверсному значению бита 4 этого регистра). Бит 4 управляет типом паритета, бит 3 задает разрешение контроля паритета. Бит 2 задает длину стоп — бита. Биты 1-0 определяют длину слова. Наиболее часто применяются слова длиной 7 бит и 8 бит.

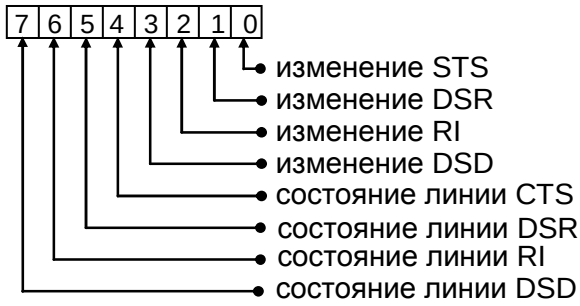
12.4.7 Регистр управления модемом (+4, byte) позволяет управлять модемом. В режиме диагностики выход передатчика переводится в состояние «1», вход приемника отключается, выход сдвигающего регистра передатчика логически соединяется с входом приемника. Входы DSR, CTS, RI, DCD отключаются от входных линий и соединяются с DTRC, RTSC, OUT1, OUT2 соответственно. Выходы управления модемом переводятся в пассивное состояние. Переданный в таком режиме байт должен немедленно приниматься, что позволяет проверять внутренние каналы данных порта и сдвиговые регистры.



12.4.8 Регистр состояния линии (+5, byte) позволяет определить состояние приемопередатчика. Бит 7 определяет ошибку принятых данных в режиме FIFO (буфер FIFO содержит хотя бы один символ, принятый с ошибкой). Бит 6: буфер передатчика пуст (символов нет ни в регистре передатчика, ни в сдвиговом регистре, ни в буфер FIFO). Бит 5: сдвиговый регистр передатчика пуст (буфер FIFO передатчика пуст). Может генерировать прерывание.



12.4.9 Регистр состояния модема (+6, byte) позволяет определить состояние модема. Биты 3-0 вызывают прерывание «Изменение состояние модема». Единица в любом из этих битов свидетельствует о том, что состояние линии изменилось по сравнению с состоянием на момент последнего чтения этого регистра.



9.5 Программирование COM порта

Примеры работы с последовательным портом на уровне регистров. Работа с последовательным портом на низком уровне аналогична работе с параллельным портом — вначале определяем базовый адрес порта, затем работаем с его регистрами. Главная отличительная особенность состоит в необходимости уста-

новки параметров COM порта перед началом работы. Приведем пример программы получения адреса COM-порта:

```
mov   ax,    40h
mov   es,       ax         ; в es - сегмент = 0040h
mov   dx,       es:[00]    ; dx = базовый адрес порта COM1
mov   COM1_adr, dx    ; запомнили адрес порта COM1 в переменной
```

Пример программы настройки параметров порта (скорость 4800 кбод, 1 стоп-бит, четность, 8 бит/слово):

```
mov   dx,       COM1_adr  ; DX - базовый адрес COM1 (+0)
add   dx,       3       ; DX - адрес регистра управления (+3)
mov   al,       80h       ; Бит DLAB=1 - настройка делителя
out   dx,       al        ; Рабочая скорость - 4800 Кбод
dec   dx
dec   dx    ; DX - адрес старшего байта делителя скорости (+1)
mov   al,       0
out   dx,       al
dec   dx   ; DX - адресу младшего байта делителя скорости (+0)
mov   al,       18h       ; Установка младшего байта делителя
out   dx,       al
add   dx,       3     ; DX - адрес регистра управления (+3)
mov   al,    00011011b ; DLAB=0, четность, 1 стоп-бит, 8 бит
out   dx,       al
dec   dx
dec   dx      ; DX - адрес регистра разрешения прерываний (+1)
mov   al,       00h       ; Прерывания запрещены
out   dx,       al
```

Пример чтения байта с ожиданием без использования прерываний:

```
mov   dx,    COM1_adr  ; DX - базовый адрес COM1 (+0)
add   dx,    5         ; DX - адрес регистра состояния (+5)
wait:
in    al,    dx        ; чтение байта состояния
test  al,    01h       ; Бит 1= "1" (принят байт)
jz    wait             ; Нет - ждем
sub   dx,    5         ; DX - адрес регистра приемника (+0)
in    al,    dx        ; чтение принятого байта
```

Пример передачи байта из AH (без контроля готовности приемника):

```
mov   dx,    COM1_adr  ; DX - базовый адрес COM1 (+0)
add   dx,    5         ; DX - адрес регистра состояния (+5)
wait:
in    al,    dx        ; чтение байта состояния
test  al,    40h       ; Бит 6=1 (готов к передаче байта)
```

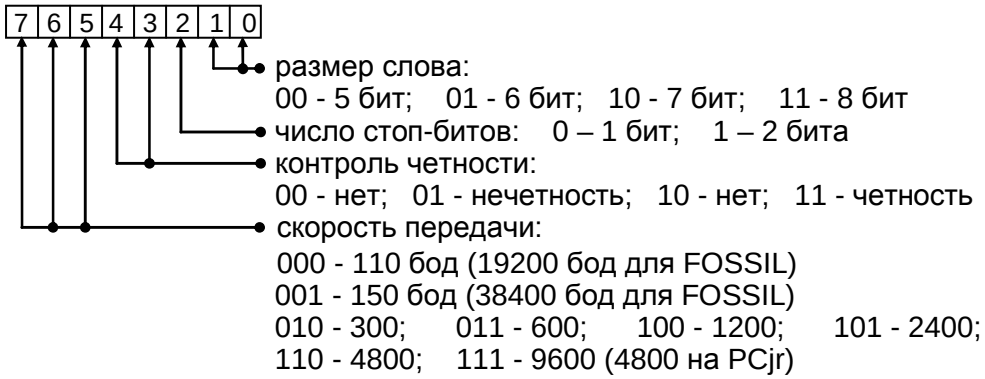
```
jz    wait                  ; Нет - ждем
sub   dx,    5          ; DX - адрес регистра передатчика (+0)
mov   al,    ah
out   dx,    al         ; Передача
```

Работа с COM портом на уровне прерываний. BIOS представляет программе пользователя набор стандартных функций прерывания INT 14h для работы с последовательным портом. Данное прерывание может быть перехвачено или замещено программой — драйвером. Основные функции этого прерывания и их входные — выходные параметры приведены в табл. 9.7.

Таблица 9.7

Функция	***Регистры при вызове***	***Регистры при возврате***
Инициализация порта	AH = 00h AL = параметры инициализации порта DX = номер порта (0-3)	AH = состояние линии AL = состояние модема
Передать символ	AH = 01h AL = символ DX = номер порта (0-3)	AH(бит 7) = 1 если успешное выполнение AH(бит 7) = 0 в случае ошибки. AH(биты 6-0) — состояние порта
Принять символ	AH = 02h DX = номер порта (0-3)	AH = состояние линии AL = принятый символ, если AH(бит 7) = 1
Получить состояние порта	AH = 03h DX = номер порта (0-3)	AH = состояние линии AL = состояние модема

Параметры инициализации порта:



Байты состояния линии и модема, возвращаемые обработчиком INT 14h, полностью соответствуют регистру состояния линии и регистру состояния модема адаптера порта:

Таблица 9.8

AH = статус линии		***AL = статус модема***	
bit 7:	timeout	bit 7:	received line detect signal
bit 6:	trans shift reg empty	bit 6:	ring indicator
bit 5:	trans holding reg empty	bit 5:	data set ready
bit 4:	break detect	bit 4:	clear to send
bit 3:	framing error	bit 3:	delta recv line signal detect
bit 2:	parity error	bit 2:	trailing edge ring detector
bit 1:	overrun error	bit 1:	delta data set ready
bit 0:	data ready status	bit 0:	delta clear to send

Пример — передать один символ на внешнее устройство:

```
mov   ax,    00h
mov   ah,    01h       ; функция 01 – передать символ
mov   al,    simbol    ; передаваемый символ <symbol>
mov   dx,    00h       ; номер COM-порта (COM1)
int   14h              ; вызов прерывания
mov   err,   ah    ; получить состояние порта в <err>
```

10 УНИВЕРСАЛЬНЫЙ ПОСЛЕДОВАТЕЛЬНЫЙ ИНТЕРФЕЙС

10.1 Введение в USB

Интерфейс ***USB*** (Universal Serial Bus — Универсальный Последовательный Интерфейс) предназначен для подключения периферийных устройств к персональному компьютеру. Позволяет производить обмен информацией с периферийными устройствами на трех скоростях (спецификация *USB 2.0*):

- Низкая скорость (*Low Speed* — LS) — 1,5 Мбит/с.
- Полная скорость (*Full Speed* — FS) — 12 Мбит/с.
- Высокая скорость (*High Speed* — HS) — 480 Мбит/с.

Для подключения периферийных устройств используется 4-жильный кабель: питание +5 В, сигнальные провода *D*+ и *D*-, общий провод.

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 о захвате части ниши применения шины FireWire.

Привлекательность USB придает возможность подключения и 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 USB входит в чипсеты практически всех системных плат PC различных размеров — и блокнотных, и настольных, и серверов.

Для того чтобы система USB заработала, необходимо, чтобы были загружены драйверы хост-контроллера (или контроллеров, если их несколько). При подключении устройства к USB шине ОС 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 Internet.

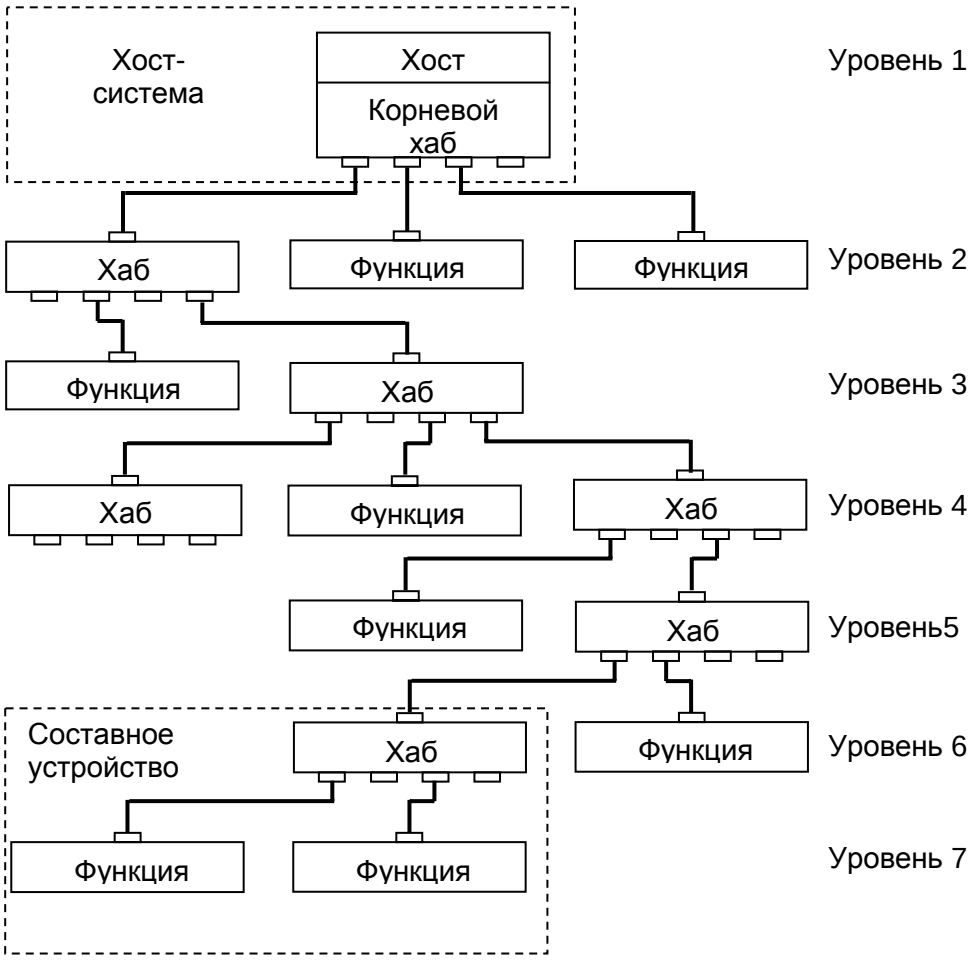


Рис. 10.1

Интерфейс USB соединяет между собой ***хост*** (*host*) и устройства. Хост входит в состав чипсета персонального компьютера и управляет работой всего интерфейса. Для того чтобы к одному порту USB можно было подключать более одного устройства, применяются ***хабы*** (*hub* — устройство, обеспечивающее подключение к интерфейсу других устройств). ***Корневой хаб*** (*root hub*) находится внутри компьютера и подключен непосредственно к хосту.

Функция — это логически законченное устройство, выполняющее какую-либо специфическую функцию.

Топология интерфейса USB представляет собой набор из 7 уровней: на первом уровне находится хост и корневой хаб, а на

последнем — только функции. Устройство, в состав которого входит хаб и одна или несколько функций, называется ***составным*** (*compaund device*).

Порт хаба или функции, подключаемый к хабу более высокого уровня, называется восходящим портом (*upstream port*), а порт хаба, подключаемый к хабу более низкого уровня или к функции называется нисходящим портом (*downstream port*).

Хаб может распознать подключение устройств к портам или отключение от них и управлять подачей питания на их сегменты. Каждый из портов может быть разрешен или запрещен и сконфигурирован на полную или ограниченную скорость обмена. Хаб обеспечивает изоляцию сегментов с низкой скоростью от высокоскоростных.

Хабы могут управлять подачей питания на нисходящие порты, предусматривается установка ограничения на ток, потребляемый каждым портом.

Все передачи данных по интерфейсу инициируются хостом. Данные передаются в виде пакетов. В интерфейсе USB используется несколько разновидностей пакетов:

- ***пакет-признак*** (*token paket*) описывает тип и направление передачи данных, адрес устройства и порядковый номер конечной точки (КТ — адресуемая часть USB-устройства); пакет-признаки бывают нескольких типов: *IN*, *OUT*, *SOF*, *SETUP*;
- ***пакет с данными*** (*data packet*) содержит передаваемые данные;
- ***пакет согласования*** (*handshake packet*) предназначен для сообщения о результатах пересылки данных; пакеты согасования бывают нескольких типов: *ACK*, *NAK*, *STALL*.

Таким образом, каждая транзакция состоит из трех фаз: фаза передачи пакета-признака, фаза передачи данных и фаза согласования.

В интерфейсе USB используются несколько типов пересылок информации.

- ***Управляющая пересылка*** (*control transfer*) используется для конфигурации устройства, а также для других специфических для конкретного устройства целей.
- ***Потоковая пересылка*** (*bulk transfer*) используется для передачи относительно большого объема информации.

- ***Пересылка с прерыванием*** (*iterrupt transfer*) используется для передачи относительно небольшого объема информации, для которого важна своевременная его пересылка. Имеет ограниченную длительность и повышенный приоритет относительно других типов пересылок.
- ***Изохронная пересылка*** (*isochronous transfer*) также называется потоковой пересылкой реального времени. Информация, передаваемая в такой пересылке, требует реального масштаба времени при ее создании, пересылке и приеме.

Потоковые пересылки характеризуются гарантированной безошибочной передачей данных между хостом и функцией посредством обнаружения ошибок при передаче и повторного запроса информации.

Когда хост становится готовым принимать данные от функции, он в фазе передачи пакета-признака посылает функции *IN*-пакет. В ответ на это функция в фазе передачи данных передает хосту пакет с данными или, если она не может сделать этого, передает *NAK*- или *STALL*-пакет. *NAK*-пакет сообщает о временной неготовности функции передавать данные, а *STALL*-пакет сообщает о необходимости вмешательства хоста. Если хост успешно получил данные, то он в фазе согласования посылает функции *ACK*-пакет. В противном случае транзакция завершается. Когда хост становится готовым передавать данные, он посылает функции *OUT*-пакет, сопровождаемый пакетом с данными. Если функция успешно получила данные, он отсылает хосту *ACK*-пакет, в противном случае отсылается *NAK*- или *STALL*-пакет.

Управляющие пересылки содержат не менее двух стадий: *Setup-стадия* и *статусная стадия*. Между ними может также располагаться *стадия передачи данных*. *Setup-стадия* используется для выполнения *SETUP-транзакции*, в процессе которой пересылается информация в управляющую КТ функции. *SETUP-транзакция* содержит *SETUP*-пакет, пакет с данным и пакет согласования. Если пакет с данными получен функцией успешно, то она отсылает хосту *ACK*-пакет. В противном случае транзакция завершается. В *стадии передачи данных* управляющие пересылки содержат одну или несколько *IN*- или *OUT*-транзакций, принцип передачи которых такой же, как и в потоковых пересылках. Все транзакции в стадии передачи данных должны произво-

диться в одном направлении. В *статусной стадии* производится последняя транзакция, которая использует те же принципы, что и в потоковых пересылках. Направление этой транзакции противоположно тому, которое использовалось в стадии передачи данных. Статусная стадия служит для сообщения о результате выполнения SETUP-стадии и стадии передачи данных. Статусная информация всегда передается от функции к хосту. При *управляющей записи* (*Control Write Transfer*) статусная информация передается в фазе передачи данных статусной стадии транзакции. При *управляющем чтении* (*Control Read Transfer*) статусная информация возвращается в фазе согласовании статусной стадии транзакции, после того как хост отправит пакет данных нулевой длины в предыдущей фазе передачи данных.

Пересылки с прерыванием могут содержать *IN*- или *OUT*-пересылки. При получении *IN*-пакета функция может вернуть пакет с данными, *NAK*-пакет или *STALL*-пакет. Если у функции нет информации, для которой требуется прерывание, то в фазе передачи данных функция возвращает *NAK*-пакет. Если работа КТ с прерыванием приостановлена, то функция возвращает *STALL*-пакет. При необходимости прерывания функция возвращает необходимую информацию в фазе передачи данных. Если хост успешно получил данные, то он посылает *ACK*-пакет. В противном случае согласующий пакет хостом не посылается.

Изохронные транзакции содержат *фазу передачи признака* и *фазу передачи данных*, но не имеют *фазы согласования*. Хост отсылает *IN*- или *OUT*-признак, после чего в фазе передачи данных КТ (для *IN*-признака) или хост (для *OUT*-признака) пересылает данные. Изохронные транзакции не поддерживают фазу согласования и повторные посылки данных в случае возникновения ошибок.

Система USB разделяется на три уровня с определенными правилами взаимодействия. Устройство USB содержит интерфейсную часть, часть устройства и функциональную часть. Хост тоже делится на три части — интерфейсную, системную и ПО устройства. Каждая часть отвечает только за определенный круг задач, логическое и реальное взаимодействие между ними иллюстрирует рис. 10.2.

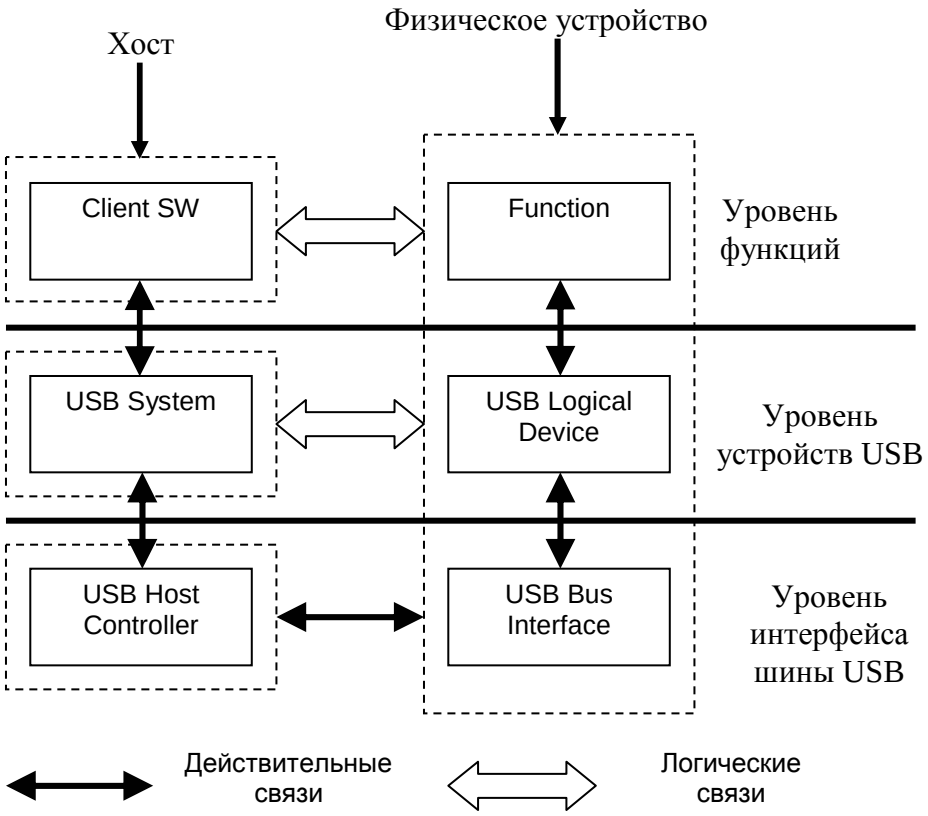


Рис. 10.2 — Взаимодействие компонентов USB

В рассматриваемую структуру входят следующие элементы:

- Физическое устройство USB — устройство на шине, выполняющее функции, интересующие конечного пользователя.
- Client SW — ПО, соответствующее конкретному устройству, исполняемое на хост-компьютере. Может являться составной частью ОС или специальным продуктом.
- USB System SW — системная поддержка USB, независимая от конкретных устройств и клиентского ПО.
- USB Host Controller — аппаратные и программные средства для подключения устройств USB к хост-компьютеру.

10.2 Физический и электрический интерфейс

Стандарт USB определяет электрические и механические спецификации шины. Информационные сигналы и питающее напряжение 5 В передаются по четырехпроводному кабелю. Используется дифференциальный способ передачи сигналов D+ и D- по двум проводам. Уровни сигналов передатчиков в статиче-

ском режиме должны быть ниже 0,3 В (низкий уровень) или выше 2,8 В (высокий уровень). Приемники выдерживают входное напряжение в пределах — 0,5...+3,8 В. Передатчики должны уметь переходить в высокоимпедансное состояние для двунаправленной полудуплексной передачи по одной паре проводов.

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. Состояния Diff0 и Diff1 определяются по разности потенциалов на линиях D+ и D- более 200 мВ при условии, что на одной из них потенциал выше порога срабатывания VSE. Состояние, при котором на обоих входах D+ и D- присутствует низкий уровень, называется линейным нулем (SEO — Single-Ended Zero). Интерфейс определяет следующие состояния:

- Data J State и Data К State — состояния передаваемого бита (или просто J и К), определяются через состояния Diff0 и Diff1.
- Idle State — пауза на шине.
- Resume State — сигнал «пробуждения» для вывода устройства из «спящего» режима.
- Start of Packet (SOP) — начало пакета (переход из Idle State в К).
- End of Packet (EOP) — конец пакета.
- Disconnect — устройство отключено от порта.
- Connect — устройство подключено к порту.
- Reset — сброс устройства.

Состояния определяются сочетаниями дифференциальных и линейных сигналов; для полной и низкой скоростей состояния DiffO и Diff1 имеют противоположное назначение. В декодировании состояний Disconnect, Connect и Reset учитывается время нахождения линий (более 2,5 мс) в определенных состояниях.

Шина имеет два режима передачи. Полная скорость передачи сигналов USB составляет 12 Мбит/с, низкая — 1,5 Мбит/с. Для полной скорости используется экранированная витая пара с импедансом 90 Ом и длиной сегмента до 5 м, для низкой — невитой неэкранированныгй кабель до 3 м. Низкоскоростные кабели и

устройства дешевле высокоскоростных. Одна и та же система может одновременно использовать оба режима; переключение для устройств осуществляется прозрачно.

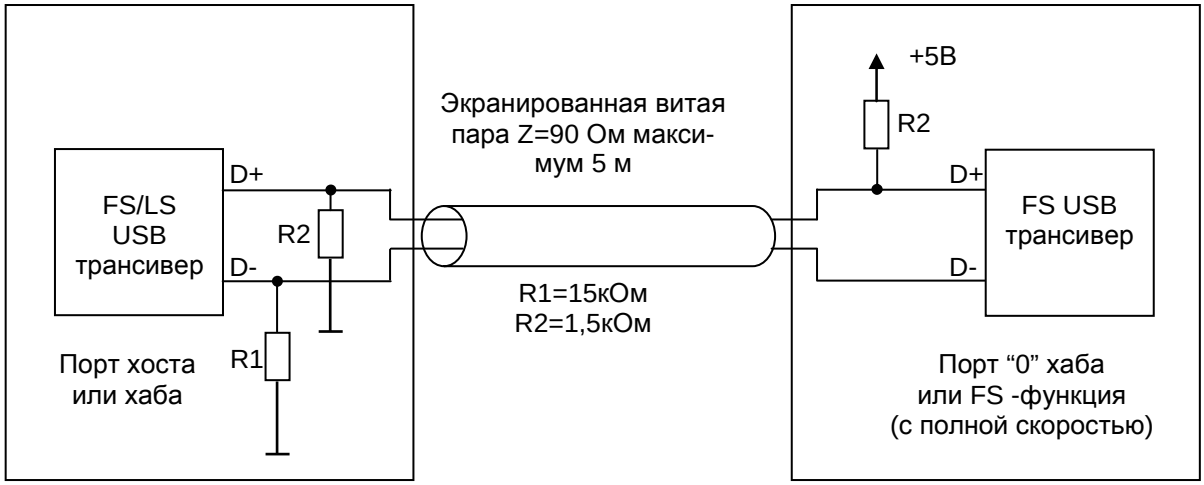


Рис. 10.3 — Подключение полноскоростного устройства

Низкая скорость предназначена для работы с небольшим количеством ПУ, не требующих высокой скорости. Скорость, используемая устройством, подключенным к конкретному порту, определяется хабом по уровням сигналов на линиях D+ и D-, смещаемых нагрузочными резисторами R2 приемопередатчиков

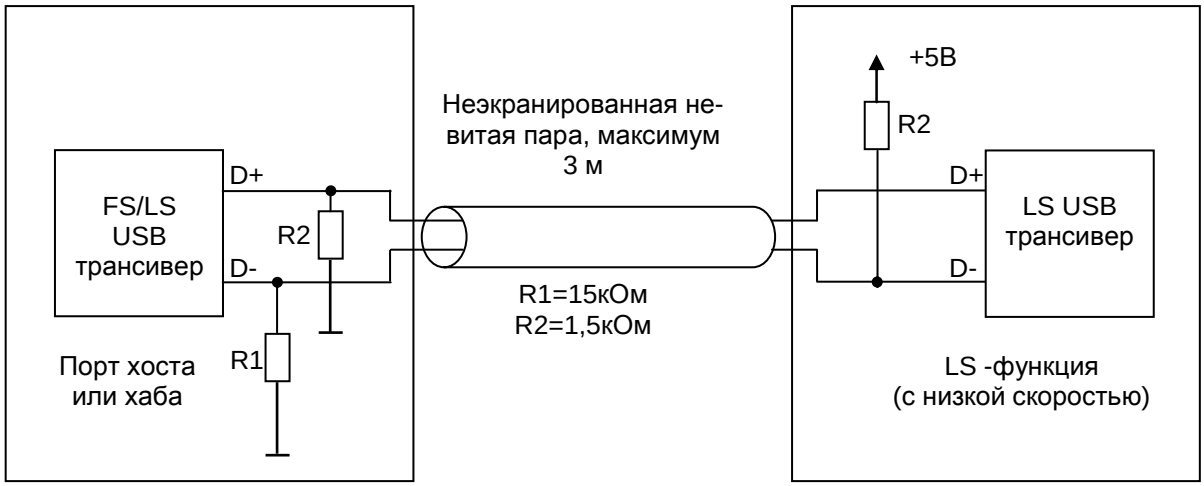


Рис. 10.4 — Подключение низкоскоростного устройства

Сигналы синхронизации кодируются вместе с данными по методу NRZI (Non Return to Zero Invert), его работу иллюстрирует рис. 10.5.

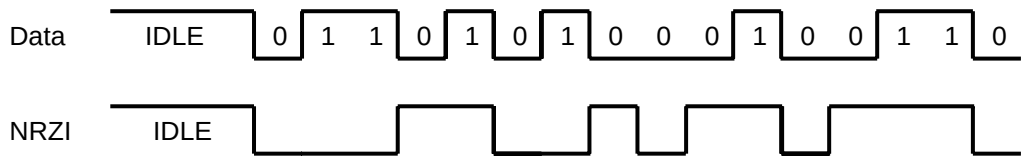


Рис. 10.5 — Подключение низкоскоростного устройства

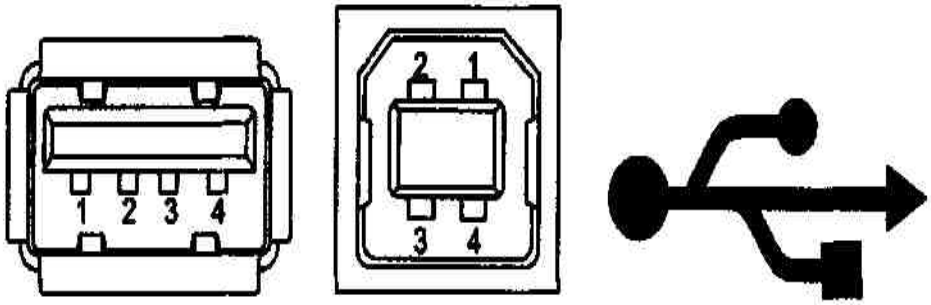
Каждому пакету предшествует поле синхронизации SYNC, позволяющее приемнику настроиться на частоту передатчика. Кабель также имеет линии VBus и GND для передачи питающего напряжения 5 В к устройствам. Сечение проводников выбирается в соответствии с длиной сегмента для обеспечения гарантированного уровня сигнала и питающего напряжения.

USB разъемы имеют следующую нумерацию и цвет контактов:

Таблица 10.1

Номер контакта	Назначение	Цвет провода
1	V BUS	Красный
2	D–	Белый
3	D+	Зеленый
4	GND	Черный
Оплетка	Экран	Оплетка

Разъемы типа «А» применяются для подключения к хабам (Upstream Connector). Вилки устанавливаются на кабелях, не отсоединяемых от устройств (например, клавиатура, мышь и т. п.). Гнезда устанавливаются на нисходящих портах (Downstream Port) хабов. Разъемы типа «В» (Downstream Connector) устанавливаются на устройствах, от которых соединительный кабель может отсоединяться (принтеры и сканеры). Ответная часть (вилка) устанавливается на соединительном кабеле, противоположный конец которого имеет вилку типа «А».



а *б* *в*

Рис. 10.6 — Гнезда USB:
а — типа «А»; *б* — типа «В», *в* — символическое обозначение

Разъемы типов «А» и «В» различаются механически (рис. 7.5), что исключает недопустимые петлевые соединения портов хабов. Четырехконтактные разъемы имеют ключи, исключающие неправильное присоединение. Конструкция разъемов обеспечивает позднее соединение и раннее отсоединение сигнальных цепей по сравнению с питающими. Для распознавания разъема USB на корпусе устройства ставится стандартное символическое обозначение.

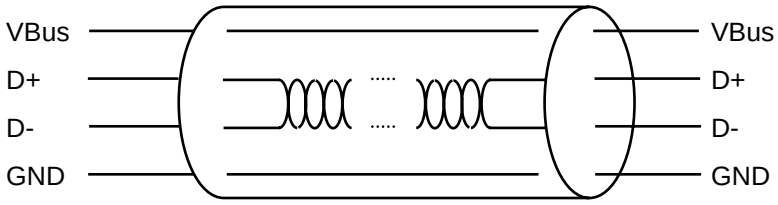


Рис. 10.7 — Сигналы USB передаются
по 4-х проводному кабелю

Здесь GND — цепь «корпуса» для питания периферийных устройств, Vbus — +5V также для цепей питания. Шина D+ предназначена для передачи данных по шине, а шина D- для приема данных. Кабель для поддержки полной скорости шины (full-speed) выполняется как витая пара, защищается экраном и может также использоваться для работы в режиме минимальной скорости (low-speed). Кабель для работы только на минимальной скорости (например, для подключения мыши) может быть любым и неэкранированным.

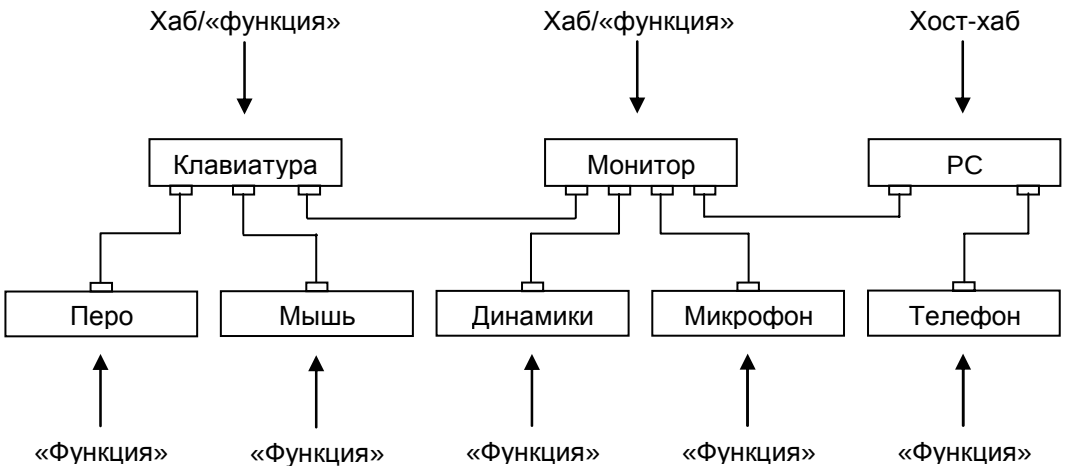


Рис. 10.8 — Пример подключения устройств USB

Питание устройств USB возможно от кабеля (Bus-Powered Devices) или от собственного блока питания (Self-Powered Devices). Хост обеспечивает питанием непосредственно подключенные к нему ПУ. Каждый хаб, в свою очередь, обеспечивает питание устройств, подключенных к его нисходящим портам. При некоторых ограничениях топологии допускается применение хабов, питающихся от шины. На рис. 10.8 приведен пример схемы соединения устройств USB. Здесь клавиатура, перо и мышь могут питаться от шины.

10.3 Модель передачи данных

Каждое устройство USB представляет собой набор независимых конечных точек (Endpoint), с которыми хост-контроллер обменивается информацией. Конечные точки описываются следующими параметрами:

- требуемой частотой доступа к шине и допустимыми задержками обслуживания;
- требуемой полосой пропускания канала;
- номером точки;
- требованиями к обработке ошибок;
- максимальными размерами передаваемых и принимаемых пакетов;
- типом обмена;

- направлением обмена (для сплошного и изохронного обменов).

Каждое устройство обязательно имеет конечную точку с номером 0, используемую для инициализации, общего управления и опроса его состояния. Эта точка всегда сконфигурирована при включении питания и подключении устройства к шине. Оно поддерживает передачи типа «управление» (см. далее).

Кроме нулевой точки, устройства-функции могут иметь дополнительные точки, реализующие полезный обмен данными. Низкоскоростные устройства могут иметь до двух дополнительных точек, полноскоростные — до 16 точек ввода и 16 точек вывода (протокольное ограничение). Точки 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 (Endpoint) устройства. Имеются два типа каналов: потоки (Stream) и сообщения (Message). Поток 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: сплошной, изохронный и прерывания. Доставка всегда идет в порядке «первым вошел — первым вышел» (FIFO); с точки зрения USB, данные потока неструктурированы. Сообщения имеют 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 отработке ошибок возможен сброс необслуженных сообщений. Двухсторонний обмен сообщениями адресуется к одной и той же конечной точке. Для доставки сообщений используется только обмен типа «управление».

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 со-

общений (Control Pipe 0), по которому передается информация конфигурирования, управления и состояния.

10.4 Типы передачи данных

USB поддерживает как однонаправленные, так и двунаправленные режимы связи. Передача данных производится между ПО хоста и конечной точкой устройства. Устройство может иметь несколько конечных точек, связь с каждой из них (канал) устанавливается независимо.

Архитектура USB допускает четыре базовых типа передачи данных:

- ***Управляющие посылки (Control Transfers),*** используемые для конфигурирования во время подключения и в процессе работы для управления устройствами. Протокол обеспечивает гарантированную доставку данных. Длина поля данных управляющей посылки не превышает 64 байт на полной скорости и 8 байт на низкой.
- ***Сплошные передачи (Bulk Data Transfers)*** сравнительно больших пакетов без жестких требований ко времени доставки. Передачи занимают всю свободную полосу пропускания шины. Пакеты имеют поле данных размером 8, 16, 32 или 64 байт. Приоритет этих передач самый низкий, они могут приостанавливаться при большой загрузке шины. Допускаются только на полной скорости передачи.
- ***Прерывания (Interrupt)*** — короткие (до 64 байт на полной скорости, до 8 байт на низкой) передачи типа вводимых символов или координат. Прерывания имеют спонтанный характер и должны обслуживаться не медленнее, чем того требует устройство. Предел времени обслуживания устанавливается в диапазоне 1—255 мс для полной скорости и 10—255 мс — для низкой.
- ***Изохронные передачи (Isochronous Transfers)*** — непрерывные передачи в реальном времени, занимающие предварительно согласованную часть пропускной способности шины и имеющие заданную задержку доставки. В случае обнаружения ошибки изохронные данные передаются без повтора — недействительные пакеты игнорируются. Пример — цифровая передача голоса. Пропускная способность определяется требованиями к

качеству передачи, а задержка доставки может быть критичной, например, при реализации телеконференций.

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. USB должна обеспечивать обмен с такой скоростью, чтобы задержка данных в устройстве, вызванная буферизацией, не превышала нескольких миллисекунд.

Изохронные передачи классифицируются по способу синхронизации конечных точек — источников или получателей данных — с системой: различают асинхронный, синхронный и адаптивный классы устройств, каждому из которых соответствует свой тип канала USB.

10.5 Протокол

Все ***обмены (транзакции)*** по USB состоят из 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 подтверждения (Handshake Packet).***

Планирование транзакций обеспечивает управление поточными каналами. На аппаратном уровне использование отказа от транзакции (NAck) при недопустимой интенсивности передачи предохраняет буферы от переполнения сверху и снизу. Маркеры отвергнутых транзакций повторно передаются в свободное для

шины время. Управление потоками позволяет гибко планировать обслуживание одновременных разнородных потоков данных.

Устойчивость к ошибкам обеспечивают следующие свойства USB:

- Высокое качество сигналов, достигаемое благодаря дифференциальным приемникам/передатчикам и экранированным кабелям.
- Защита полей управления и данных CRC-кодами.
- Обнаружение подключения и отключения устройств и конфигурирование ресурсов на системном уровне.
- Самовосстановление протокола с тайм-аутом при потере пакетов.
- Управление потоком для обеспечения изохронности и управления аппаратными буферами.
- Независимость функций от неудачных обменов с другими функциями.

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.

10.6 Форматы пакетов

Байты передаются по шине последовательно, начиная с младшего бита. Все посылки организованы в пакеты. Каждый пакет начинается с поля синхронизации Sync, которое представляется последовательностью состояний KJKJKJKK (кодированную по NRZI), следующую после состояния Idle. Последние два бита (КК) являются маркером начала пакета SOP, используемым для идентификации первого бита ***идентификатора пакета PID***. Идентификатор пакета является 4-битным полем PID[3:0], идентифицирующим тип пакета (табл. 10.2), за которым в качестве контрольных следуют те же 4 бита, но инвертированные.

Таблица 10.2

Тип PID	Имя PID	PID[3:0]	Содержимое и назначение
Token	OUT	0001	Адрес функции и номер конечной точки — маркер транзакции функции
Token	IN	1001	Адрес функции и номер конечной точки — маркер транзакции хоста
Token	SOF	0101	Маркер начала кадра
Token	SETUP	1101	Адрес функции и номер конечной точки — маркер транзакции с управляющей точкой
Data	Data_O Data_l	0011 1011	Пакеты данных с четными нечетным PID чередуются для точной идентификации подтверждений
Handshake	Ack	0010	Подтверждение безошибочного приема пакета
Handshake	NAK	1010	Приемник не сумел принять или передатчик не сумел передать данные. В транзакциях прерываний является признаком отсутствия не обслуженных прерываний
Handshake	STALL	1110	Конечная точка требует вмешательства хоста
Special	PRE	1100	Преамбула передачи на низкой скорости

В пакетах-маркерах IN, SETUP и OUT следующими являются адресные поля: 7-битный адрес функции и 4-битный адрес конечной точки. Они позволяют адресовать до 127 функций USB (нулевой адрес используется для конфигурирования) и по 16 конечных точек в каждой функции.

В пакете SOF имеется 11-битное поле номера кадра (Frame Number Field), последовательно (циклически) увеличиваемое для очередного кадра.

Поле данных может иметь размер от 0 до 1023 целых байт. Размер поля зависит от типа передачи и согласуется при установлении канала.

Поле СКС-кола присутствует во всех маркерах и пакетах данных, оно защищает все поля пакета, исключая PID. CRC для

маркеров (5 бит) и данных (11 бит) подсчитываются по разным формулам.

Каждая транзакция инициируется хост-контроллером посылкой маркера и завершается пакетом квитирования. Последовательность пакетов в транзакциях иллюстрирует рис. 10.9.

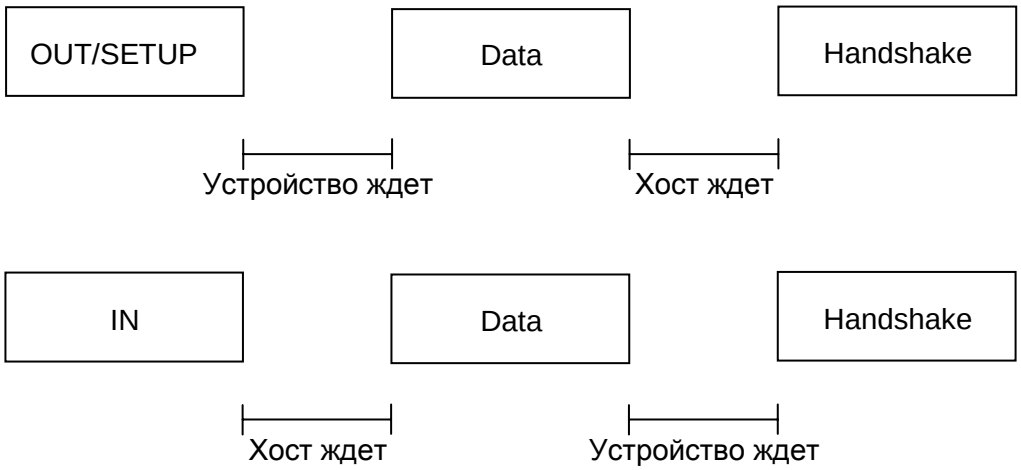


Рис. 10.9 — Последовательности пакетов

Хост-контроллер организует обмены с устройствами согласно своему плану распределения ресурсов. Контроллер циклически (с периодом 1 мс) формирует ***кадры (Frames),*** в которые укладываются все запланированные транзакции. 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. Каждый кадр имеет свой номер. Хост-контроллер оперирует 32-битным счетчиком, но в маркере SOF передает только младшие 11 бит. Номер кадра увеличивается (циклически) во время EOF. Хост планирует загрузку кадров так, чтобы в них всегда находилось место для транзакций управления и прерывания. Свободное время кадров может заполняться сплошными передачами (Bulk Transfers).

Для изохронной передачи важна синхронизация устройств и контроллера. Есть три варианта:

- синхронизация внутреннего генератора устройства с маркерами SOF;

- подстройка частоты кадров под частоту устройства;
- согласование скорости передачи (приема) устройства с частотой кадров.

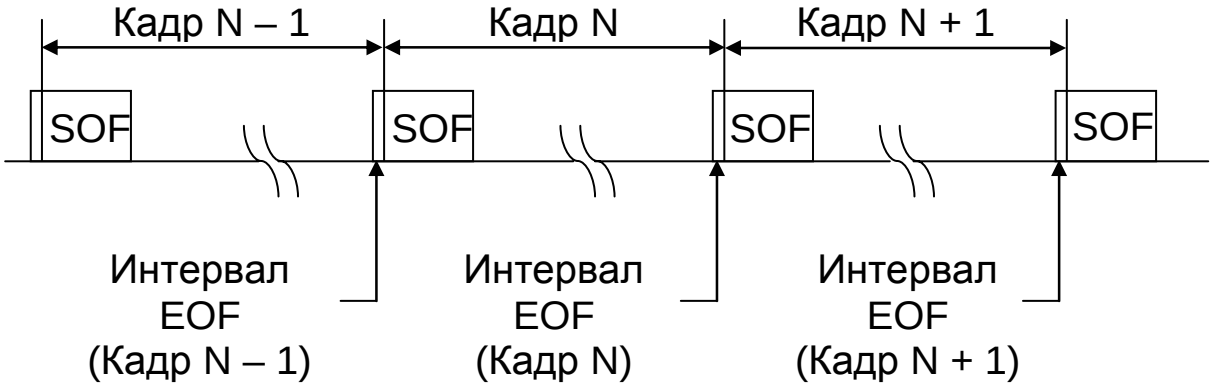


Рис. 10.10 — Поток кадров USB

Подстройка частоты кадров контроллера возможна, естественно, под частоту внутренней синхронизации только одного устройства. Подстройка осуществляется через механизм обратной связи, который позволяет изменять период кадра в пределах ±1 битового интервала.

10.7 Системное конфигурирование

USB поддерживает динамическое подключение и отключение устройств. Нумерация устройств шины является постоянным процессом, отслеживающим изменения физической топологии.

Все устройства подключаются через порты хабов. Хабы определяют подключение и отключение устройств к своим портам и сообщают 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. Нумерация устройств, подключенных к шине (Bus Enumeration), осуществляется динамически по мере их подключения (или включения их питания) без какого-либо вмешательства пользователя или клиентского ПО. Процедура нумерации выполняется следующим образом:

1. Хаб, к которому подключилось устройство, информирует хост о смене состояния своего порта ответом на опрос состояния. С этого момента устройство переходит в состояние Attached (подключено), а порт, к которому оно подключилось, в состояние Disabled.

2. Хост уточняет состояние порта.

3. Узнав порт, к которому подключилось новое устройство, хост дает команду сброса и разрешения порта.

4. Хаб формирует сигнал Reset для данного порта (10 мс) и переводит его в состояние Enabled. Подключенное устройство может потреблять от шины ток питания до 100 мА. Устройство переходит в состояние Powered (питание подано), все его регистры переводятся в исходное состояние, и оно отзывается на обращение по нулевому адресу.

5. Пока устройство не получит уникальный адрес, оно доступно по дежурному каналу, по которому хост-контроллер определяет максимально допустимый размер поля данных пакета.

6. Хост сообщает устройству его уникальный адрес, и оно переводится в состояние Addressed (адресовано).

7. Хост считывает конфигурацию устройства, включая заявленный потребляемый ток от шины. Считывание может затянуться на несколько кадров.

8. Исходя из полученной информации, хост конфигурирует все имеющиеся конечные точки данного устройства, которое переводится в состояние Configured (сконфигурировано). Теперь хаб позволяет устройству потреблять от шины полный ток, заявленный в конфигурации. Устройство готово.

Когда устройство отключается от шины, хаб уведомляет об этом хост и работа порта запрещается, а хост обновляет свою текущую топологическую информацию.

10.8 Устройства USB — функции и хабы

Возможности шины USB позволяют использовать ее для подключения разнообразных устройств. Не касаясь «полезных» свойств ПУ, остановимся на их интерфейсной части, связанной с шиной USB. Все устройства должны поддерживать набор общих операций, перечисленных ниже. Динамическое подключение и отключение. Эти события отслеживаются хабом, который сообщает о них хост-контроллеру и выполняет сброс подключенного устройства. Устройство после сигнала сброса должно отзываться на нулевой адрес, при этом оно не сконфигурировано и не приостановлено. После назначения адреса, за которое отвечает хост-контроллер, устройство должно отзываться только на свой уникальный адрес.

Конфигурирование устройств, выполняемое хостом, является необходимым для их использования. Для конфигурирования обычно используется информация, считанная из самого устройства. Устройство может иметь множество интерфейсов, каждому из которых соответствует собственная конечная точка, представляющая хосту функцию устройства. Интерфейс в конфигурации может иметь альтернативные наборы характеристик; смена наборов поддерживается протоколом. Для поддержки адаптивных драйверов дескрипторы устройств и интерфейсов имеют поля класса, подкласса и протокола.

Передача данных возможна посредством одного из четырех типов передач (см. выше). Для конечных точек, допускающих разные типы передач, после конфигурирования доступен только один из них.

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, и если хаб не сможет обеспечить устройству заявленный ток, оно не конфигурируется и, следовательно, не может быть использовано.

Устройство USB должно поддерживать приостановку (Suspended Mode), в котором его потребляемый ток не превышает 500 мкА. Устройство должно автоматически приостанавливаться при прекращении активности шины.

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.

Хаб в USB выполняет коммутацию сигналов и выдачу питающего напряжения, а также отслеживает состояние подключенных к нему устройств, уведомляя хост об изменениях. Хаб состоит из двух частей — контроллера (Hub Controller) и повторителя (Hub Repeater). Повторитель представляет собой управляемый ключ, соединяющий выходной порт со входным. Он имеет средства поддержки сброса и приостановки передачи сигналов. Контроллер содержит регистры для взаимодействия с хостом. Доступ к регистрам осуществляется по специфическим командам обращения к хабу. Команды позволяют конфигурировать хаб, управлять нисходящими портами и наблюдать их состояние.

Нисходящие (Downstream) порты хабов могут находиться в следующих состояниях:

- ***Powered (питание отключено)*** — на порт не подается питание (возможно только для хабов, коммутирующих питание). Выходные буферы переводятся в высокоимпедансное состояние, входные сигналы игнорируются.
- ***Disconnected*** (отсоединен) — порт не передает сигналы ни в одном направлении, но способен обнаружить подключение устройства (по отсутствию состояния SEO в течение 2,5 мкс). Тогда порт переходит в состояние Disabled, а по уровням входных

сигналов {DiffO или Diff1 в состоянии Idle) он определяет скорость подключенного устройства.

- ***Disabled (запрещен)*** — порт передает только сигнал сброса (по команде от контроллера), сигналы от порта (кроме обнаружения отключения) не воспринимаются. По обнаружении отключения (2,5 мкс состояния SEO) порт переходит в состояние Disconnect, а если отключение обнаружено «спящим» хабом, контроллеру будет послан сигнал Resume.
- ***Enabled (разрешен)*** — порт передает сигналы в обоих направлениях. По команде контроллера или по обнаружении ошибки кадра порт переходит в состояние Disabled, а по обнаружении отключения — в состояние Disconnect.
- ***Suspended (приостановлен)*** — порт передает сигнал перевода в состояние останова («спящий» режим). Если хаб находится в активном состоянии, сигналы через порт не пропускаются ни в одном направлении. Однако «спящий» хаб воспринимает сигналы смены состояния незапрещенных портов, подавая «пробуждающие» сигналы от активизировавшегося устройства даже через цепочку «спящих» хабов. Состояние каждого порта идентифицируется контроллером хаба с помощью отдельных регистров. Имеется общий регистр, биты которого отражают факт изменения состояния каждого порта (фиксируемый во время EOF). Это позволяет хост-контроллеру быстро узнать состояние хаба, а в случае обнаружения изменений специальными транзакциями уточнить состояние.

10.9 Хост-контроллер

У каждой шины USB должен быть один (и только один!) хост — компьютер с контроллером USB. Хост должен выполнять следующие функции:

- обнаруживать подключения и отсоединения устройств USB, по этим событиям загружать и выгружать соответствующие им драйверы и уведомлять об этих событиях «заинтересованное» ПО;
- манипулировать потоками управления и данных между устройствами и использующими их модулями ПО;
- собирать статистику активности и состояний устройств;

- управлять электрическим интерфейсом между хост-контроллером и устройствами USB, включая управление электропитанием.

Аппаратным посредником между устройствами USB и хостом является хост-контроллер, от которого требуется следующее:

- Обработка состояний (state handling) — как компонент хоста, он должен управлять своим состоянием и сообщать его хосту.
- Последовательные преобразования данных из байт, естественных для хоста, в поток бит шины и обратно.
- Генерация (микро) кадров.
- Отработка запросов на передачу данных от хоста и к хосту.
- Отработка протокола USB.
- Обработка ошибок передачи по шине.
- Обеспечение удаленного пробуждения хоста по сигналу от устройств USB.
- Реализация корневого хаба — предоставление как минимум одного порта со всеми свойствами хаба USB.
- Реализация интерфейса с хост-системой — обеспечение высокоскоростного обмена данными с основной памятью хост-компьютера.

10.10 Работа с USB устройствами

Поскольку USB — шина последовательная, то в каждый момент времени передается только один поток данных. Данные передаются пакетами. Каждый пакет имеет заголовок отвечающий за транспортировку и маршрутизацию данных. В отличие от Ethernet, USB полоса ***(bandwidth)*** (общая пропускная способность) шины делится между устройствами не по принципу «кто первый занял» (random access). В USB ресурсы распределяются централизовано — концентратором (hub) шины. У шины может быть только один корневой концентратор ***(root hub),*** управляющий работой всей шины (он распределяет ресурсы шины между устройствами). Поэтому нельзя соединить два компьютер напрямую проводом (наподобие нуль-модема) через USB — получится конфигурация с двумя корневыми концентраторами. К корневому концентратору подключаются устройства — всего до 127. Уст-

ройство может быть в свою очередь концентратором, контролирующим нижележащий сегмент шины. Таким образом, USB шина может выглядеть как многоранговая звезда (дерево).

10.10.1 Программная модель

Рассматривая программную модель USB для Windows прежде всего стоит отметить следующие особенности:

1. Каждое USB устройство должно обслуживаться собственным драйвером. В отличие от, скажем, устройств, подключаемых к LPT, для которых наличие собственного драйвера в общем необязательно. На шину USB нельзя просто передать сырой поток данных.

2. Все USB контроллеры соответствуют спецификации ACPI, т.е. поддерживают функции PnP и управления питанием. Поэтому работать с устройством можно только если оно подключено к шине.

3. Драйвера в Windows образуют так называемый ***стек***, по которому и передаются данные вверх и вниз, не следует пытаться работать напрямую с портами контроллера.

В состав каждого чипсета входит несколько независимых USB контроллеров, каждый из которых имеет по два порта. Каждый контроллер имеет в диапазоне ввода/вывода набор портов, через которые ими можно управлять.

Производитель контроллера обеспечивает программный интерфейс посредством драйвера. Драйвер корневого концентратора (***usbhub.sys***) создает логическое устройство позволяющие для вышележащих драйверов абстрагироваться от подробностей работы драйвера USB контроллера. Для организации интерфейса с клиентом (драйвером) существует драйвер шины (***usbd.sys***). Его интерфейс документирован в DDK. Это набор управляющих кодов для диспетчера (просто процедура, обрабатывающая запросы) запросов типа RP_MJ_INTERNAL_DEVICE_CONTROL. С помощью этих запросов можно получить разнообразную информацию о состоянии шины, количестве портов, их статусе и пр. Все эти коды имеют символические имена вида: IOCTL_INTERNAL_USB_XXXX.

Особое внимание следует обратить на запрос IOCTL_INTERNAL_USB_SUBMIT_URB. Управление устройством, запись и чтение данных осуществляется именно через этот запрос. Каждое USB-устройство обязано иметь свой драйвер, который имеет диспетчер для IRP_MJ_WRITE (который вызывается диспетчером ввода-вывода для записи). У некоторых классов устройств существую стандартные драйвера: для HID устройств (устройства интерфейса — мыши, клавиатуры и.т.п.), USB audio — устройства воспроизведения звука (ЦАП располагается прямо в локальной аудиоустановке, а по шине передается цифровой поток). Для устройств, принадлежащих к этим классам, не нужно драйверов.

Поговорим немного о модели ввода/вывода. Для разработчиков пользовательских программ это, возможно, абсолютно незнакомая тема. С аппаратной точки зрения вывод/вывод бывает программный — использование инструкций ***in, out, mov*** а также (наиболее интересный случай!) использование строковых операций с префиксом повтора (REP MOVSB) или с использованием контроллера прямого доступа к памяти (DMA). Однако, в данном случае речь идет о другом. Аппаратным вводом/выводом занимается как раз драйвер USB контроллера. А нас интересует, куда попадают принятые данные (и куда помещаются данные, предназначенные для передачи)? Существует два подхода: «буферизованный» и «прямой» ввод/вывод.

При буферизованном выводе происходит следующее:

1. Пользовательское приложение вызывает функцию (например, WriteFile), передав указатель на буфер содержащий данные.

2. ОС вызывает диспетчер IRP_MJ_WRITE драйвера и передает туда (через структуру IRP) указатель на буфер данных

3. Драйвер копирует данные в свой внутренний буфер. После этого, возможно, сообщает, что данные переданы или откладывает это сообщение до момента актуальной передачи.

4. Актуальная передача осуществляется когда-то позже.

При буферизованном вводе все аналогично. Главное достоинство этого метода — принятые данные не могут «пропасть» (если только внутренний буфер не переполниться). Буферизованный ввод/вывод осуществляет, например, драйвер последова-

тельного порта. Для медленных устройств ввода/вывода это рекомендованный способ.

Для быстрых устройств, особенно передающих данные большими пакетами, использование программных буферов имеет два основных недостатка — большие накладные расходы на копирование данных в (из) промежуточный буфер, нерационально используется системная память. Поэтому в данном случае используется прямой ввод/вывод — данные принимаются непосредственно в буфер (или передаются из этого буфера), зарезервированный пользовательской программой. Производительность такого метода заметно выше. Однако возникает некоторые сложности с приемом данных — у драйвера всегда должен быть «под рукой» буфер. И это должен обеспечить клиент (пользовательская программа или вышележащий драйвер).

10.10.2 Устройства и каналы шины USB

Каждое устройство на шине USB имеет несколько, так называемых, конечных точек — endpoints. Эти пронумерованные точки являются концами логических каналов данных между хостом и устройством. Таким образом, между хостом и устройством реализуется мнококанальная передача данных. Каналы симплексны.

Идентификационный номер устройства дается ему хостом при подключении устройства к шине, номера же конечных точек задаются при изготовлении устройства.

Любое устройство должно поддерживать endpoint 0, так как это средство конфигурирования устройства по умолчанию в USB (Default Control Pipe) после включения или получения сигнала сброса по шине. Все остальные EP и каналы появляются после конфигурирования устройства хостом. DCP может так же использоваться специфичным клиентским ПО хоста. В этом случае системное ПО хоста USB является посредником между клиентским ПО и DCP устройства.

Конечные точки имеют собственные характеристики, их необходимо знать клиентскому ПО для определения типа соединения:

- частота передачи;

- требования по пропускной способности канала;
- номер EP;
- требования по обработке ошибок;
- максимальный размер пакета, который может отправлять/принимать эта EP;
- тип передачи (режим);
- направление передачи.

Виртуальные каналы между конечными точками устройства и ПО хоста гипотетически могут реализовывать два типа передачи:

- ***Поток (stream)*** — не имеет определенной структуры в системе USB.
- ***Сообщение (message)*** — имеет определенную в USB структуру.

10.10.3 Признаки и идентификаторы пакетов на шине USB

Передача по шине USB состоит из пакетов данных, идентифицированных специальными кодами, называемыми Packet ID (идентификаторами пакетов). PID показывает, какого типа пакет был передан. Нижеследующий рисунок иллюстрирует передачу по шине USB от хоста к устройству.

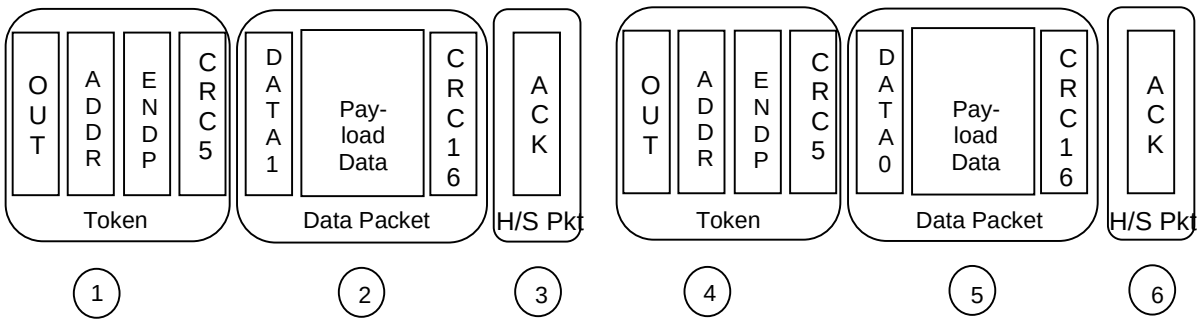


Рис. 10.11

Пакет 1 — признак передачи, индицируемый OUT PID. Признак отображает, что данные от хоста будут переданы по шине. *Пакет 2* содержит данные, о чем говорит PID DATA1. *Пакет 3* (handshake) направляется устройством с использованием ACK PID для подтверждения хосту, что устройство получило

дынные без ошибок. Вторая передача начинается с еще одного признака OUT (*Пакет* 4), далее следуют данные (*Пакет* 5), на этот раз предваряемые PID DATA0. В конце передачи устройство опять подтверждает успешную передачу, передавая ACK PID (*Пакет 6*).

Наличие DATA1 и DATA0 обусловлено коррекцией ошибок. Как описано выше, ACK — это сигнал хосту, что периферия получила дынные без ошибки (поля CRC в пакетах используются для определения ошибок). Но что произойдет, если пакет повредится во время передачи? Для определения этого, каждая сторона устанавливает ***data toggle bit***, переключаемый между передачами пакетов данных. Состояние этого флажка сравнивается с PID, приходящим с данными, с каждым DATA0 или DATA1. При передаче данных хост или устройство посылают перемежающиеся DATA0-DATA1 идентификаторы пакетов. Путем сравнения идентификатора DATA с состоянием внутреннего переключаемого флажка, хост или устройство могут определить поврежденный handshake packet.

Признак SETUP предназначен только для передач типа CONTROL. Они предваряют 8 байт данных, из которых периферия определяет запросы хоста.

Признак SOF приходит каждую миллисекунду, обозначая фрэйм USB.

Существуют три PID согласования: ACK, NAK и STALL:

- ACK означает, что данные были приняты без ошибок.
- NAK означает — занято, передайте еще раз. Это не означает ошибки, так как ошибка — это отсутствие ответа.
- STALL — означает, что произошло нечто непредвиденное (возможно как результат потери связи или несогласования программного обеспечения разных уровней). Устройство посылает STALL, показывая, что не поняло запрос.

PID PRE предшествует передаче на малой скорости (1.5 Мбит). Мы его игнорируем.

10.10.4 Передача данных по шине USB

Для передачи данных устройству, хост передает признак OUT, за которым следуют данные. Если устройство имеет воз-

можность принять данные и принимает их без ошибки, оно возвращает ACK хосту. Если оно занято, оно отсылает NAK.

При передаче данных хосту периферийное устройство ждет прихода от хоста запроса IN. Если хост не посылает эти запросы, данные навсегда останутся в буфере устройства.

Хост в системе USB обеспечивает распределение времени, передавая SOF (начало фрейма) каждую миллисекунду. Пакет SOF включает в себя инкрементирующийся 11-разрядный счетчик фреймов.

Напомним, что в системе USB возможны четыре типа передачи. Они соответствуют требованиям различных типов данных, передаваемых по шине:

- Control.
- Bulk.
- Interrupt.
- Isochronous.

Далее мы затронем форматы передачи данных для всех этих режимов и поясним это рисунками.

Режим Bulk

Схематическое изображение процесса обмена пакетами в этом режиме показано на рисунке:

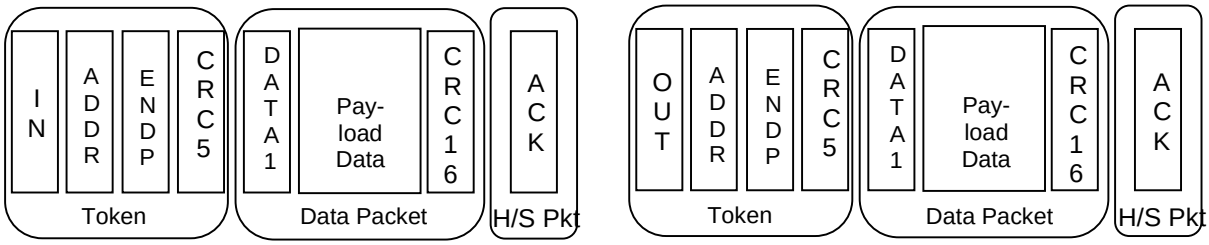


Рис. 10.12

Передача типа Bulk — «прорывная», пакетами по 8, 16, 32, 64 байта. Доставка информации гарантирована по причине автоматической перепосылки поврежденных данных. Хост позволяет передавать пакеты bulk, когда шина освобождается. Этот тип передачи применяется для принтера, сканера или модема. Данные, передаваемые таким способом имеют встроенный контроль передачи, обеспечиваемый пакетами согласования.

Режим Interrupt

Схематическое изображение процесса обмена пакетами в этом режиме показано на рисунке:

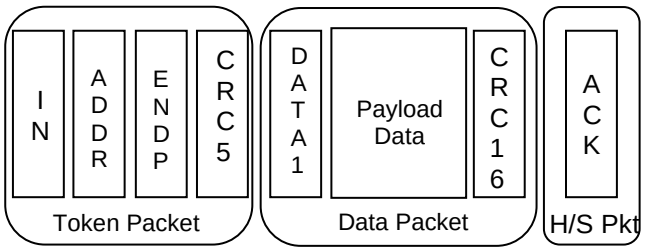


Рис. 10.13

Этот тип передачи похож на bulk, но передача происходит только для IN каналов. Блок данных может иметь длину от 1 до 64 байт. Каналы interrupt типа имеют ассоциированный временной интервал, через эти интервалы хост посылает признак IN.

Режим Isochronous

Схематическое изображение процесса обмена пакетами в этом режиме показано на рисунке:

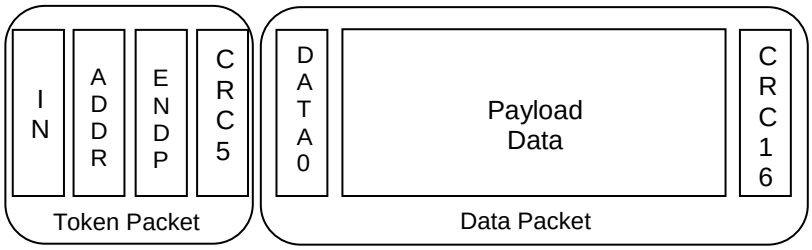


Рис. 10.14

Режим предназначен для передачи потоковой, критичной ко времени информации, такой как аудио или видео информация, целесообразно применять этот режим передачи. Для такой информации время доставки. основное требование.

В каждом фрейме USB определенная жесткая часть времени отводится для передачи типа isochronous. Заглядывая вперед скажем, что такая передача не имеет признаков согласования (ACK/NAK/STALL) и не имеет перепосылок. Определение ошибок происходит только на уровне согласования контрольной

суммы CRC16. Передача isochronous не использует механизм переключения флажков DATA, передается всегда PID DATA0.

Режим Control

Схематическое изображение процесса обмена пакетами в этом режиме показано на рис.10.15.

Этот тип передачи используется для конфигурирования и посылки команд устройству. Так как эти задачи жизненно важны для функционирования системы, при передаче типа control обеспечивается наилучшая защита от ошибок. Хост резервирует часть каждого фрэйма для передачи control.

Передачи control состоят из двух или трех стадий. Стадия SETUP содержит 8 байт управляющих данных. Опциональная стадия DATA содержит дополнительные данные, если необходимо. Стадия STATUS позволяет устройству индицировать благополучное завершение процесса конфигурирования.

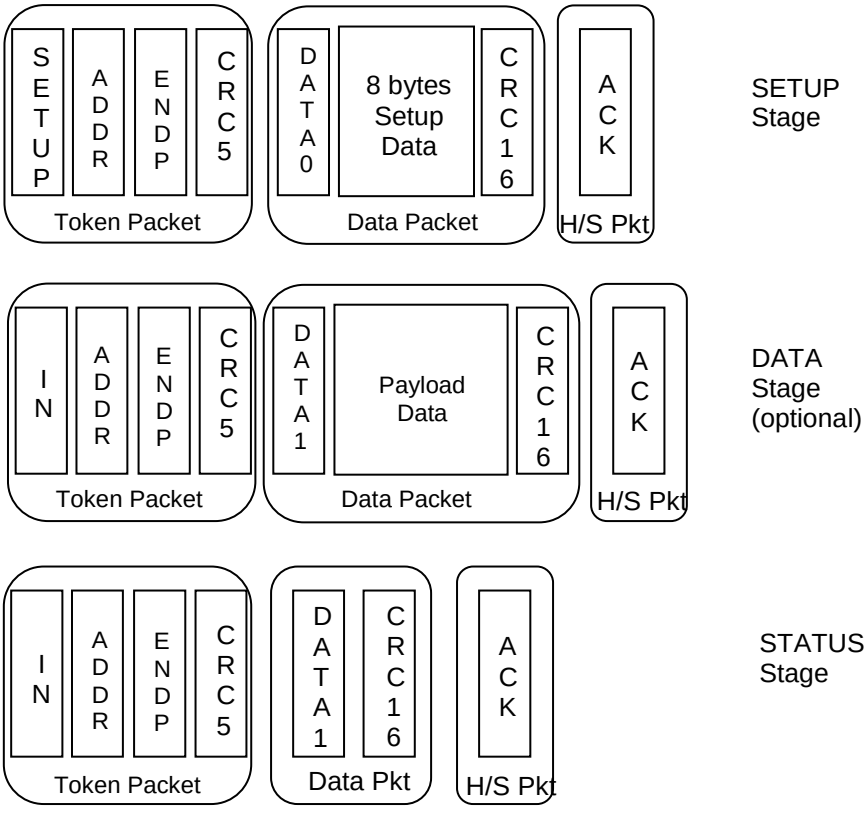


Рис. 10.15

10.10.5 Включение в систему и нумерация устройств на шине USB

При включении нового устройства в шину хост производит последовательность.перепеси. устройств:

- Хост посылает запрос «Get_Descriptor/Device» на адрес 0 (впервые подключенные устройства отвечают по адресу 0).
- Устройство обязательно отвечает на запрос, посылая байты идентификатора обратно хосту, говорящие, что за устройство подключено.
- Хост посылает устройству запрос «Set_Address», дающий уникальный адрес устройству, позволяющий отличить это устройство от других, подключенных к шине.
- Хост посылает еще запросы «Get_Descriptor», запрашивая остальную информацию об устройстве. Из этих ответов хост узнает, как много каналов устройство имеет, его требования по питанию, необходимую пропускную способность канала и какой драйвер загрузить.
- Этот процесс «переписи» называется Enumeration (нумерация).

USB была разработана группой из семи компаний, которые видели необходимость во взаимодействии для обеспечения дальнейшего роста и развития расцветающей индустрии интегрированных компьютеров и телефонии. Эти семь компаний, продвигающие USB, следующие: Compaq, Digital Equipment Corp, IBM PC Co., Intel, Microsoft, NEC и Northern Telecom.

ЛИТЕРАТУРА

1. ОС ТУСУР 6.1-97. Работы студенческие учебные и выпускные квалификационные. Общие требования к правилам оформления.

2. Бондарь А.В. Эксплуатация и развитие компьютерных сетей и систем. Часть 1: Программно-аппаратная реализация персонального компьютера: Учебное пособие. — Томск: ТУСУР, 1999.

3. Брябин В.М. Программное обеспечение персональных ЭВМ. — М.: Наука. Гл. ред. физ.-мат. лит., 1988. — 272 с.

4. Фролов А.В., Фролов Г.В. Аппаратное обеспечение IBM PC. В 2-х ч. — М.: ДИАЛОГ-МИФИ, 1992. — 208 с.

5. Воробьев Н.И. Проектирование электронных устройств: Учебное пособие для вузов по спец. «Автоматика и управление в технических системах». — М.: Высш. шк., 1989. — 223 с.

6. Левкин Г.Н., Левкина В.Е. Введение в схемотехнику IBM PC/AT. — М.: Изд-во МПИ, 1991. — 96 с.

7. Сопряжение датчиков и устройств ввода данных с компьютерами IBM PC: Пер. с англ. / Под ред. У. Томпкинса, Дж. Уэстера. — М.: Мир, 1992. — 592 с.

8. Шарапов А.В. Микропроцессорные устройства и системы: Методические указания к выполнению курсового проекта. — Томск: ТУСУР, 1998. — 38 с.

9. Мячев А.А. Системы ввода-вывода ЭВМ. — М.: Энергоатомиздат, 1983. — 168 с.

10. Гибсон Г.Ю. Аппаратные и программные средства микро-ЭВМ. — М.: Финансы и статистика, 1983. — 304 с.

11. Хвощ С.Т. и др. Организация последовательных мультиплексных каналов систем автоматизированного управления. — Л.: Машиностроение, 1988. — 120 с.

12. Якимов О.П. Газоразрядные матричные индикатроные панели. — М.: Сов. радио, 1980. — 72 с.

13. Однокристальные микроЭВМ: Справочник / Под ред. А.В. Боборыкина, А.А. Сергеева и др. — М.: МИКАП, 1994. — 400 с.