

Министерство образования и науки Российской Федерации

Федеральное государственное бюджетное образовательное учреждение
высшего профессионального образования
«Владимирский государственный университет
имени Александра Григорьевича и Николая Григорьевича Столетовых»

В. И. БЫКОВ

СИСТЕМА ВВОДА-ВЫВОДА ЭВМ И ВС И ЕЕ ИНТЕРФЕЙСЫ

Учебное пособие



Владимир 2015

УДК 004.326.8(03)

ББК 32.973.2-044

Б95

Рецензенты:

Доктор технических наук, профессор
зав. кафедрой информационных систем и программной инженерии
Владимирского государственного университета
имени Александра Григорьевича и Николая Григорьевича Столетовых
И. Е. Жигалов

Доктор технических наук, профессор
зав. кафедрой информационных технологий
Владимирского филиала Российской академии народного хозяйства
и государственной службы при Президенте Российской Федерации
А. С. Шалумов

Печатается по решению редакционно-издательского совета ВлГУ

Быков, В. И.

Б95 Система ввода-вывода ЭВМ и ВС и ее интерфейсы : учеб.
пособие / В. И. Быков ; Владим. гос. ун-т им. А. Г. и Н. Г. Столе-
товых. – Владимир : Изд-во ВлГУ, 2015. – 230 с. – ISBN 978-5-
9984-0583-9.

Содержит описание системы ввода-вывода современных ЭВМ и их аппа-
ратных интерфейсов, прежде всего использующихся в ПК. Подробно рассмотре-
ны шины расширения ПК, универсальные внешние интерфейсы, интерфейсы
внешних запоминающих устройств, специализированные интерфейсы перифе-
рийных устройств, видеоинтерфейсы и беспроводные интерфейсы. При описа-
нии интерфейсов рассматриваются все три компонента: протоколы и алгорит-
мы, электрические параметры и конструктивное исполнение.

Предназначено для студентов направления 230100 «Информатика и вы-
числительная техника» при изучении дисциплины «ЭВМ и периферийные
устройства» и специальности 230101 «Вычислительные машины, комплексы,
системы и сети» дневной и заочной форм обучения при изучении дисциплины
«Интерфейсы периферийных устройств».

Рекомендовано для формирования профессиональных компетенций в со-
ответствии с ФГОС 3-го поколения.

Ил. 137. Табл. 14. Библиогр. : 8 назв.

УДК 004.326.8(03)

ББК 32.973.2-044

ISBN 978-5-9984-0583-9

© ВлГУ, 2015

ПРЕДИСЛОВИЕ

В современных ЭВМ используется большое количество периферийных устройств (ПУ) для ввода, хранения и вывода информации. Подключение ПУ к ЭВМ осуществляется через соответствующие интерфейсы.

Предлагаемое учебное пособие содержит описание системы ввода-вывода современных ЭВМ и аппаратных интерфейсов, используемых в ЭВМ и вычислительных системах (ВС). Подробно рассмотрены как традиционные интерфейсы (*RS-232C*, *IEEE-1284 Centronics*), реализованные в *SOM*- и *LPT*-портах, так и современные интерфейсы. Описаны шины расширения ПК, универсальные внешние интерфейсы, интерфейсы внешних запоминающих устройств, специализированные интерфейсы периферийных устройств, видеоинтерфейсы и беспроводные интерфейсы. При описании интерфейсов рассматриваются все три компонента: протоколы и алгоритмы, электрические параметры и сигналы, конструктивное исполнение (разъемы, кабели и платы).

Изучение интерфейсов является важной и необходимой составляющей при подготовке студентов по направлению 230100 «Информатика и вычислительная техника» и специальности 230101 «Вычислительные машины, комплексы, системы и сети» дневной и заочной форм обучения. Для направления 230100 «Информатика и вычислительная техника» в федеральном компоненте ГОС-3 предусмотрено изучение дисциплины «Электронно-вычислительные машины и периферийные устройства», включающей раздел «Интерфейсы периферийных устройств». Вопросы, связанные с ПУ и их интерфейсами, изучаются и в дисциплинах других специальностей, таких как «Вычислительные средства информационных систем», «Вычислительные машины, комплексы, системы и сети» и др. Поэтому предлагаемое учебное пособие будет полезно студентам многих специальностей.

ВВЕДЕНИЕ

В настоящее время круг задач, решаемых на ЭВМ, постоянно расширяется. В связи с этим расширяется и состав периферийных устройств (ПУ). Назначение ПУ – обмен информацией между ядром ЭВМ и внешним миром. Стоимость ПУ составляет 50 – 80 % от стоимости всей ЭВМ. В большинстве случаев ПУ являются слабым звеном в ЭВМ и ВС. Слабость периферийных устройств проявляется в меньшей надежности, в меньшем быстродействии (по сравнению с ядром ЭВМ уступают на несколько порядков), в меньшей степени своего совершенства и интеграции. Это связано с тем, что в ПУ используются все физические принципы, за исключением ядерной физики.

Классы задач, решаемых на ЭВМ, и роль периферийных устройств в этих задачах представлена в таблице. Все задачи, решаемые вычислительными машинами, с точки зрения использования ПУ, можно разделить на шесть классов.

Классификация периферийных устройств

Периферийные устройства могут быть классифицированы по многим признакам, основные из которых рассмотрены ниже.

I. По способу представления информации:

1. Графическая:

- статическая (графическая);
- динамическая (видеоинформация).

2. Текстовая.

3. Речевая.

4. Сенсорная.

5. Аналоговые сигналы.

6. Информация внешней памяти*.

7. Системы межмашинных связей (электрическая)*.

II. По направлению обмена и назначению:

1. Ввод.

2. Вывод.

3. Двусторонний обмен.

4. Устройства подготовки данных.

5. Декодирование информации.

* Представляют собой двоичные сигналы.

Классы задач, решаемых ЭВМ, и роль ПУ в этих задачах

№ п/п	Параметр	Классы задач					
		Обработка изображений	Игры	САПР	Управление объектом	Информационно-справочные, статистические	Научно-технические
1	Сложность алгоритма	↑	↑	↑	↓	↓	↑
2	Объем вычислений на одно входное сообщение	↑	>10	>100	<10	<10	Более 10 – 100 вычислений
3	Требуемая точность вычислений	↑	↓	↑	↓	Различная	↑
4	Объем вычислений и входной информации	↑	↑	↑	↑	↑	↓
5	Режимы	Индивидуальный	Реального времени	Индивидуальный пакетный	Реального времени	Индивидуальный	Индивидуальный, пакетный
6	Особенности входной и выходной информации	Графическая информация	Графическая информация	Графический характер информации	Требуется обработка аналоговой информации	Переменная длина квантов информации Большое число символьной информации	–

III. По типу носителей информации:

1. Бумажные.
2. Магнитные.
3. Оптические.
4. Магнитно-оптические.

К носителю информации предъявляются следующие требования:

- плотность записи;
- возможность многократного использования;
- надежность считывания;
- низкая стоимость;
- долговечность;
- быстрый доступ.

IV. По быстродействию:

1. Низкоскоростные (обеспечивают обработку информации со скоростью меньше 100 символов в секунду, например матричные принтеры).
2. Среднескоростные (до 1000 символов/с).
3. Высокоскоростные (до 1 М символов/с).
4. Сверхбыстродействующие (более 1 М символов/с).

V. По характеру цикла обмена:

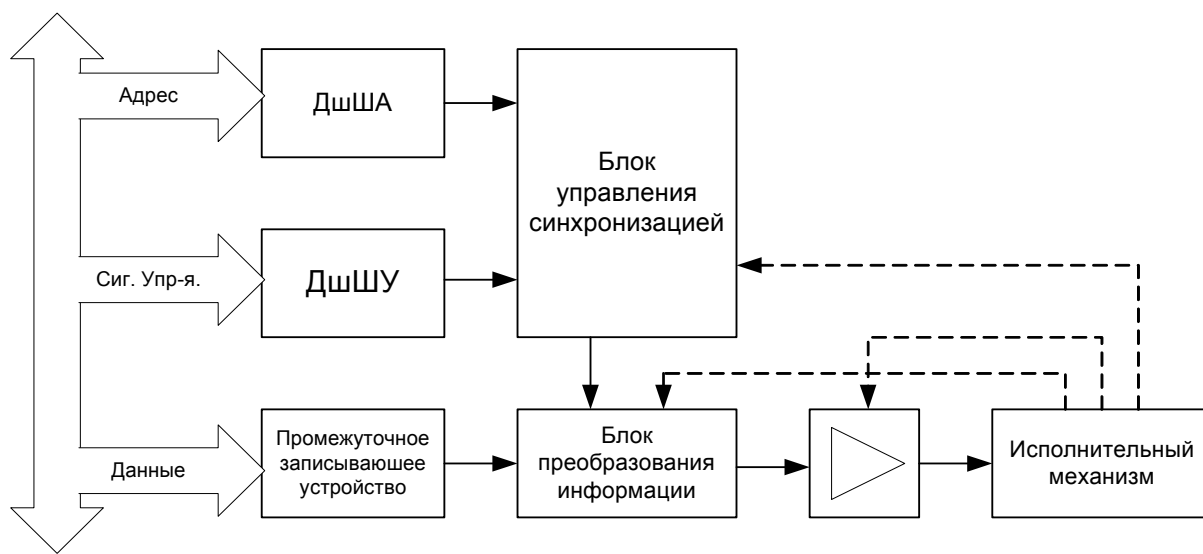
1. Синхронные ($T_{\text{ПУ}} = T_{\text{подготовки}} + T_{\text{обмена}}$).
2. Асинхронные ($T_{\text{ПУ}} = T_{\text{подготовки}} + T_{\text{обмена}} + T_{\text{ожидания}}$).

VI. По назначению:

1. Устройства отображения информации (УОИ) – дисплеи, индикаторы.
2. Устройства хранения информации (внешние запоминающие устройства – ВЗУ) – диски, ленты, карты, барабаны.
3. Устройства регистрации информации (УРИ) – принтеры, плоттеры.
4. Устройства ввода информации (УВИ):
 - ручные: клавиатура, мышь и т.п.;
 - автоматические: сканеры, фото- и видеокамеры и т.п.
5. Устройства виртуальной реальности (УВР) – шлемы, тренажерные комплексы и т.п.

Большое количество ПУ, высокие требования к скорости обмена требуют развитой системы ввода-вывода (СВВ) интерфейсов в современных ЭВМ. Поэтому роль дисциплины ЭВМ и ПУ очень важна в подготовке специалистов по направлению «Информатика и вычислительная техника» и специальности «Вычислительные машины, комплексы, системы и сети».

Обобщенная структурная схема ПУ представлена на рисунке.



Структурная схема ПУ (внешнего устройства)

Глава 1. СТРУКТУРА СИСТЕМЫ ВВОДА-ВЫВОДА ЭВМ

Система ввода-вывода (СВВ) предназначена для ввода-вывода информации. Её задача заключается в обеспечении повышения эффективности работы ЭВМ по обработке информации в целом.

Пути повышения эффективной производительности ЭВМ:

1. Увеличение номинального быстродействия устройств, участвующих в операциях.
2. Совмещение операций обработки и ввода-вывода.
3. Совмещение нескольких операций ввода-вывода (особенно важно при наличии разноскоростных устройств ввода-вывода).
4. Совмещение операций обработки (актуально, если ввод-вывод не является сдерживающим фактором).

1.1. Требования к системе ввода-вывода, ее функции и схемы

Исходя из вышеперечисленных путей повышения эффективной производительности ЭВМ, к системе ввода-вывода предъявляются следующие требования:

1. Обеспечение максимальной эффективной производительности. Это возможно сделать при автономных средствах ввода-вывода, таких как процессоры ввода-вывода, отдельные периферийные терминальные ЭВМ и др. Достоинство – разделение операций ввода-вывода и обработки по устройствам и совмещение по времени, недостаток – сложность аппаратуры, а следовательно, увеличение стоимости.

2. Обеспечение минимальной стоимости ЭВМ. Это достигается реализацией операций ввода-вывода программным путем. Достоинство – низкая стоимость аппаратуры, недостаток – повышение временных затрат.

3. Возможность реализации ЭВМ с переменной конфигурацией, в том числе и по составу периферийного оборудования. Это необходимо для настройки ЭВМ на конкретный класс задач. Для этого необходима стандартизация интерфейсов, что, в свою очередь, ведет к увеличению и аппаратных, и временных затрат.

4. Обеспечение возможности модификации отдельных ПУ без нарушения целостности системы.

5. Стандартизация и упрощение программирования операций ввода-вывода и независимость программирования от особенностей периферийных устройств.

6. Обеспечение автоматического распознавания и реакции ядра ЭВМ на многообразие ситуаций в периферийном устройстве.

Пути решения проблем при разработке устройств ввода-вывода (УВВ):

1. Модульность (агрегатность), т.е. каждое ПУ является самостоятельным устройством, таким, что при присоединении нового ПУ не происходит изменения текущей конфигурации.

2. Унификация форматов данных, которыми ПУ обмениваются с ядром ЭВМ (необходимы преобразования в самих ПУ).

3. Унифицированный интерфейс. Интерфейс – набор шин, соединений, электрических схем, сигналов, алгоритмов и протоколов обмена.

4. Унификация форматов и набора команд процессора для операций ввода-вывода.

Основные функции системы ввода-вывода:

1. Преобразование форматов или квантов информации, передаваемых ПУ, в форматы процессора или оперативной памяти (ОП).

2. Определение адреса ОП, где должны быть размещены и считаны сформированные кванты информации.

3. Формирование управляющих сигналов C_i для работы ПУ.

4. Получение и обработка осведомительных сигналов S_i от ПУ. Эти сигналы характеризуют состояние периферийного устройства.

5. Получение приказов от центральных устройств на выполнение ввода-вывода и формирование сообщений о состоянии СВВ.

6. Синхронизация процессов ПУ и центрального процессора (ЦП) и согласование скоростей их работы.

Общая структура СВВ показана на рис. 1.1.

Организационно (логически) СВВ состоит из каналов, а физически в СВВ включаются ПУ, интерфейсы, слоты, контроллеры, процессоры ввода-вывода.

Эта структура реализуется в двух видах схем:

1. Схема с каналами (процессорами) ввода-вывода (рис. 1.2).

Ранее использовалась лишь в *mainframe* и суперкомпьютерах. Это дорогая система, но она имеет хорошую производительность, так как процессор частично освобождается от выполнения команд ввода-вывода. В настоящее время применяется и в ПК на основе хэбовой (мостовой) структуры (рис.1.4).

2. Схема с общей шиной (рис. 1.3).

Эта схема в чистом виде сейчас применяется редко.

Достоинство – малая стоимость и простота реализации, недостаток – низкое быстродействие.

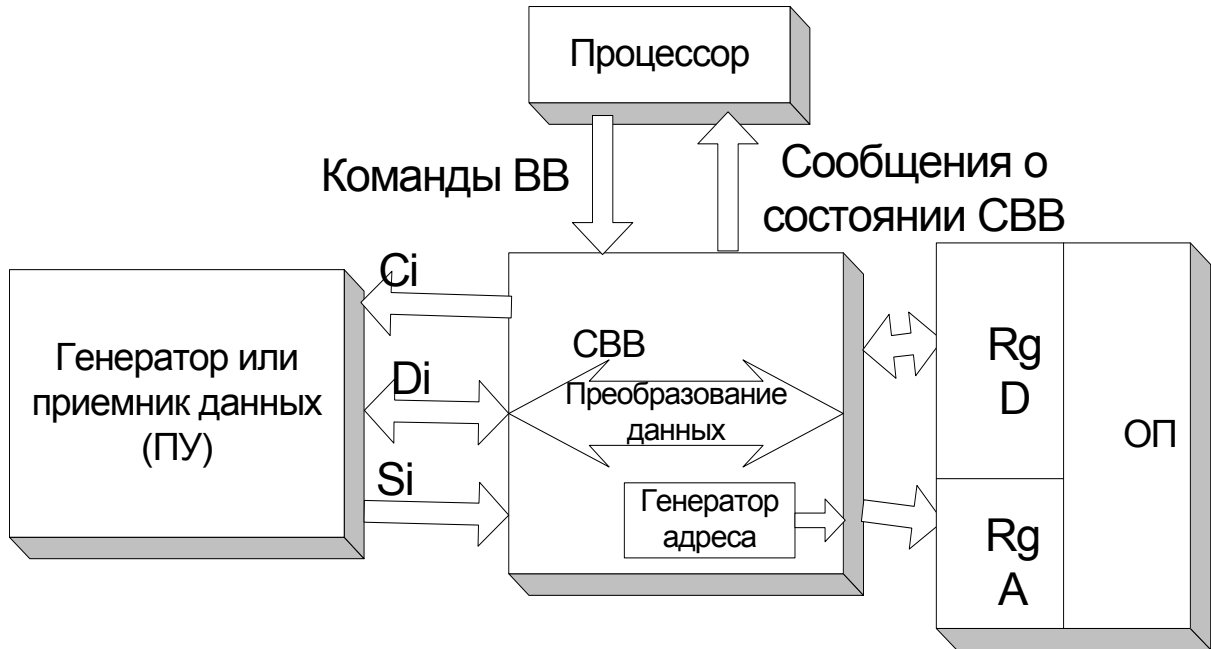


Рис. 1.1. Общая структура СВВ

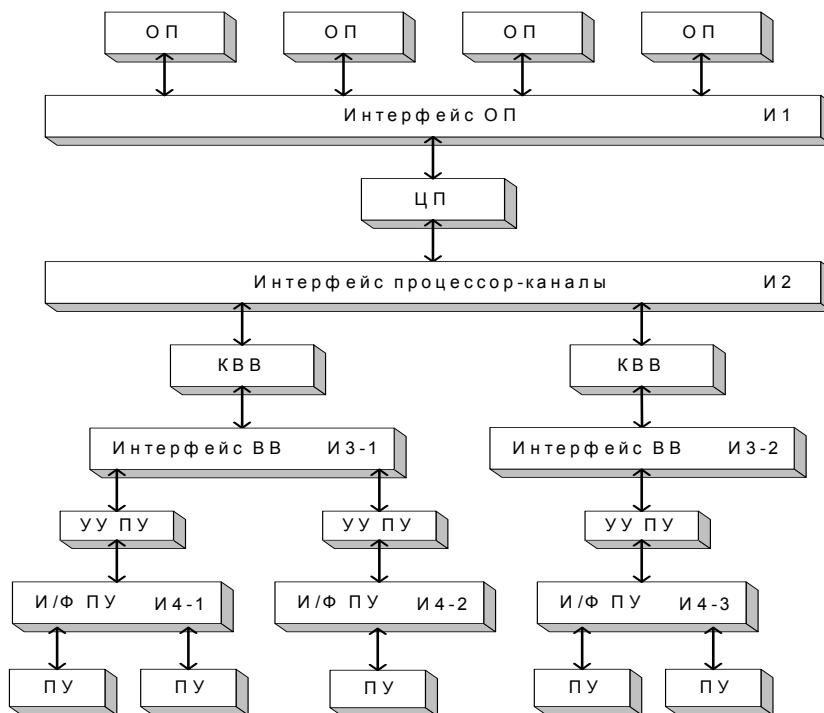


Рис. 1.2. Схема с каналами (процессорами) ввода-вывода

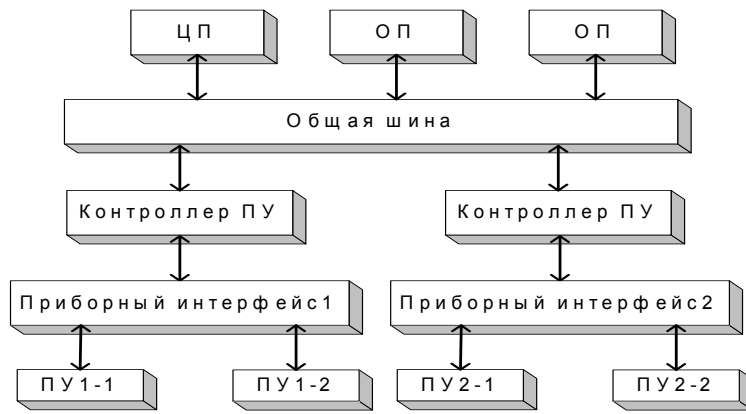


Рис. 1.3. Схема с общей шиной

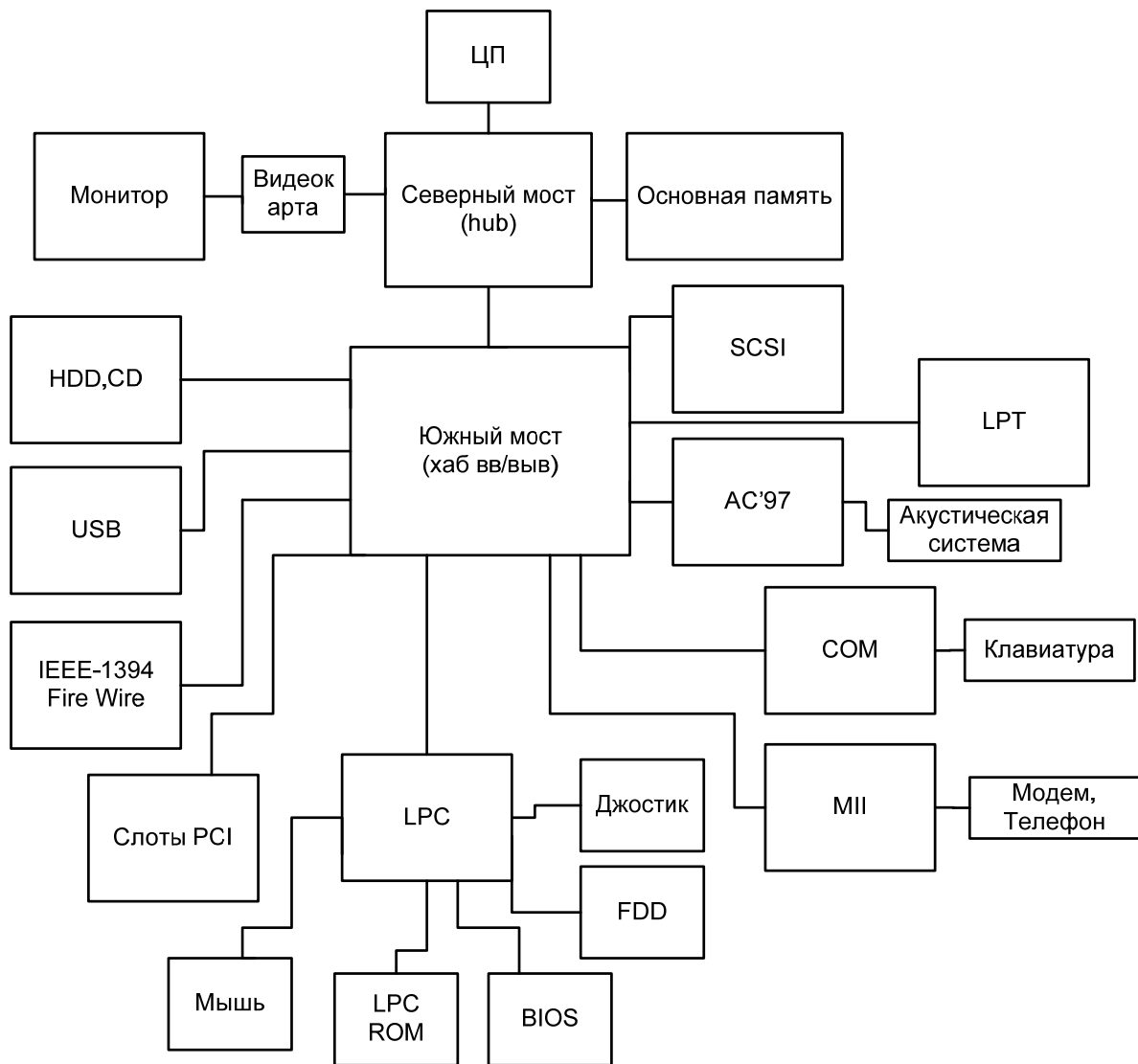


Рис. 1.4. Структура СВВ современных ПК

1.2. Интерфейсы

Интерфейс – совокупность правил и средств унифицированного обмена информацией между ЭВМ и ПУ.

Средства:

1. Аппаратные:

- шины, линии;
- электронные схемы;
- сигналы.

2. Конструктивные:

- разъемы;
- платы;
- кабели.

3. Программные:

- алгоритмические языки высокого уровня;
- протоколы;
- драйверы;
- прерывания;
- *BIOS*.

Система интерфейсов – группа интерфейсов, применяемая в одной вычислительной системе.

Шина – совокупность линий.

Интерфейс должен обеспечивать совместимость следующих видов:

- конструктивная (разъемы, кабели, форматы плат);
- электрическая (уровни сигналов, их временные параметры, микросхемы);
- информационная (протоколы, алгоритмы, драйверы).

Параметры интерфейсов:

- пропускная способность;
- направление обмена;
- тип интерфейса (последовательный, параллельный); для параллельного: ширина – количество бит, передаваемых параллельно;
- максимальная частота передачи сигнала;
- допустимое расстояние передачи;

- динамические параметры – длительность:
- а) передачи слова;
- б) передачи блока (с учетом процедур подготовки и разрушения канала);
- общее число линий интерфейса.

1.2.1. Классификация интерфейсов

Классы интерфейсов:

1. Организационно-системная классификация (по назначению (месту) в системе):

- 0 – интерфейс человек-машина;
- 1 – межмашинный (сетевой);
- 2 – системный интерфейс;
- 3 – интерфейс ОП;
- 4 – интерфейс процессор-каналы;
- 5 – интерфейс ввода-вывода (шины расширения);
- 6 – малый (приборный) интерфейс (интерфейс подключения ПУ).

2. Конструктивная классификация (по конструктивному назначению):

- межмашинный;
- межблочный;
- внутриблочный;
- межплатный;
- внутриплатный.

3. По типу связи:

- дуплексный – обмен в двух направлениях одновременно;
- полудуплексный – информация передается в двух направлениях раздельно во времени;
- симплексный – обмен в одном направлении.

4. По среде передачи информации:

- электрическая (провода, кабели);
- оптическая (оптоволокно, ИК-порт);
- электромагнитная (волноводы, радиоволны).

5. По типу кабеля:

- коаксиальный кабель;

- простая пара;
- витая пара;
- световод, оптоволокно;
- электромагнитная среда;
- волновод.

Скорость передачи по линии зависит от длины линии и её типа. При малой длине линии скорость определяется задержками в приемопередающей аппаратуре. При средней длине скорость пропорциональна длине.

6. Классификация по способу подключения:

- радиальный (рис. 1.5).

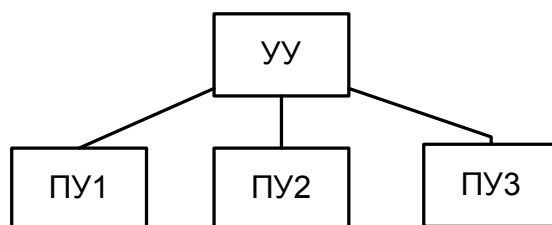


Рис. 1.5. Радиальный способ подключения

Достоинства – высокое быстродействие, возможность обслуживания нескольких ПУ, простота организации обмена. Недостаток – высокие сложность и стоимость.

- Магистральный (рис. 1.6).

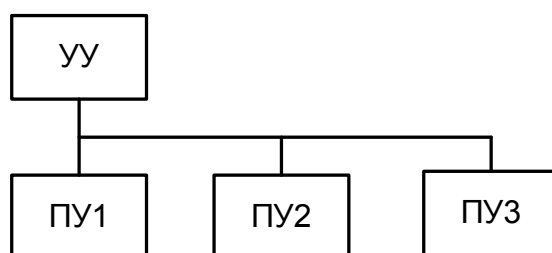


Рис. 1.6. Магистральный способ подключения

Используются коллективные средства соединения с разделением времени. Каждому ПУ присваивается свой адрес, который передается по специальным линиям связи для установления канала.

Достоинства – уменьшение количества линий в соответствии с типом интерфейса. Недостаток – невозможность обслуживания нескольких ПУ одновременно.

– Цепочечный интерфейс (рис. 1.7).



Рис. 1.7. Цепочечный способ подключения

При запросе адрес проходит через все периферийное устройство, что замедляет процесс адресации. Порядок запроса зависит от последовательности подключения.

Достоинства – малые аппаратные затраты, уменьшение времени опроса. Недостаток – увеличение временных затрат при передаче информации.

– Комбинированный интерфейс (рис. 1.8).

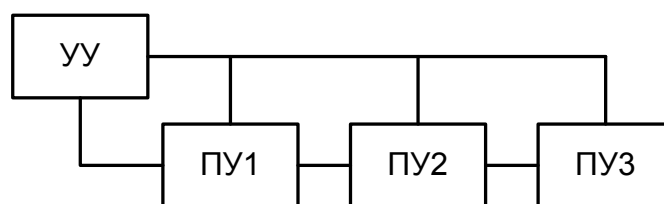


Рис. 1.8. Комбинированный способ подключения

Имеет магистральную структуру всех линий, кроме линии опроса. Достоинства – высокая скорость опроса и передачи данных.

– Древоподобный интерфейс (многоярусная звезда) (рис. 1.9).

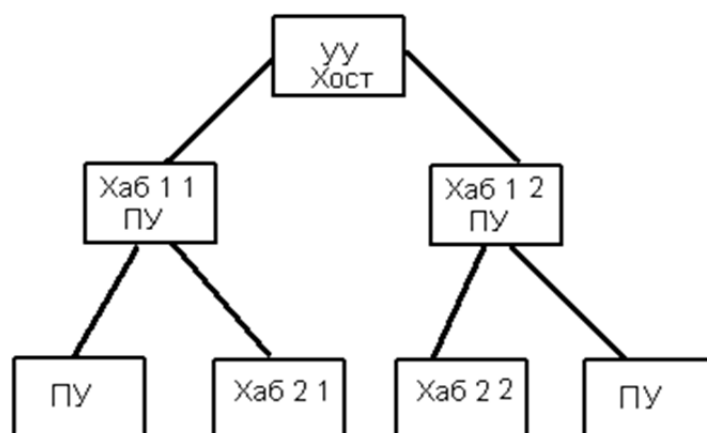


Рис. 1.9. Древоподобный способ подключения (многоярусная звезда)

1.2.2. Факторы, влияющие на сигнал в линии

При прохождении сигнала по линии он ослабляется и искажается. Причины этого:

- потери;
- отражение;
- помехи (наводки);
- перекрестные искажения;
- перекос.

Потери возникают из-за того, что любой провод имеет сопротивление, на котором падает часть напряжения сигнала. Кроме того, между сигнальным проводом и землей создается паразитная емкость, при заряде которой происходит увеличение фронта и задержки.

Отражение вызвано тем, что невозможно строгое согласование входного сопротивления приемника и выходного сопротивления передатчика сигнала. Поэтому часть сигнала отражается на входе приемника, что снижает полезный сигнал. Согласование особенно важно при использовании коаксиальных кабелей и волноводов.

Помехи на линии вызваны наводками на провод от электромагнитных полей, имеющих в окружающем пространстве. Электромагнитные поля создаются либо целенаправленно для нужд радиосвязи, радио- и телевидения и т.д., либо непреднамеренно – линиями электропередач, искрением щеток электромашин, проводов троллейбусов, разрядами молний и т.п.

Перекрестные искажения возникают в многожильных кабелях за счет взаимных наводок от одного провода на другой. Эти наводки связаны с тем, что вокруг любого провода, по которому течет ток, возникает электромагнитное поле, которое воспринимается соседними проводами.

Перекос может быть только в параллельных интерфейсах и связан с тем, что задержка сигнала в различных проводах одного кабеля различна. Поэтому биты, передаваемые параллельно, приходят на вход приемника не одновременно. Это может вызвать ложные срабатывания и требует строгой синхронизации на стороне приемника, чтобы их исключить.

Влияние всех рассмотренных факторов зависит от частоты сигнала и длительности импульсов. Чем выше частота сигнала, тем

больше величина электромагнитного поля и, следовательно, его влияние на соседние провода. Чем меньше длительность сигнала, тем больше в процентном отношении задержка, вызванная наличием паразитных сопротивлений и емкостей. В современных интерфейсах, работающих на частотах сотни мегагерц и гигагерц, все эти факторы влияют достаточно сильно. Для уменьшения помех и перекрестных искажений используют коаксиальные кабели и экранированные витые пары. Для уменьшения потерь и отражений применяют качественные медные провода и серебрение или золочение контактов. Для исключения перекоса используют последовательные интерфейсы.

1.2.3. Тенденции развития интерфейсов

В настоящее время в связи с расширением класса решаемых задач количество интерфейсов в вычислительных системах постоянно увеличивается. Это происходит как за счет появления новых интерфейсов, так и за счет того, что интерфейсы, ранее применяемые в дорогих больших ВС, находят все более широкое применение в недорогих бюджетных ПК. В связи с этим повышаются требования к интерфейсам. Основные требования:

- 1) техническое – пропускная способность (скорость, производительность);
- 2) эксплуатационные:
 - простота подключения;
 - компактность;
- 3) экономическое – стоимость.

Удовлетворение второго требования определяет одну из тенденций – применение беспроводных интерфейсов. Это приводит к внедрению существующих и появлению новых инфракрасных и радиоинтерфейсов.

Другой тенденцией современных интерфейсов, направленной на повышение производительности, является развитая многоуровневая система протоколов и требуемые для ее реализации сложные интеллектуальные контроллеры (хосты и хабы). Это может приводить к некоторому росту стоимости, но это не столь значительно, так как стоимость электронных компонентов (микросхем) постоянно снижается.

Третьей важной тенденцией, способствующей удовлетворению практически всех требований, можно назвать переход от параллельных к последовательным интерфейсам. Современные последовательные интерфейсы имеют следующие преимущества с точки зрения пользователя:

- возможность подключения большого количества ПУ;
- многообразие подключаемых периферийных устройств;
- более простая реализация возможности «горячего» подключения и динамической конфигурации в любом смысле;
- возможность питания устройств от интерфейса;
- возможность выделять гарантированные и изохронные каналы;
- лучшая помехоустойчивость;
- уменьшение количества проводов в кабеле и контактов в разъеме, т.е. экономия пространства (не бьющая по карману миниатюризация) и снижение сложности монтажа.

Последняя особенность означает удовлетворение требований компактности и стоимости. Эти преимущества обеспечиваются следующими конструктивными особенностями:

1. Развитая многоуровневая система протоколов.
2. Сложные интеллектуальные контроллеры (хосты и хабы), т. е. перенос все большей части практической реализации шины на кремний, что облегчает отладку, повышает гибкость и сокращает время разработки.
3. Переход от разделяемых шин с арбитражем и непредсказуемыми прерываниями, неудобными для надежных / критических систем, к более предсказуемым соединениям точка-точка.
4. Лучшая с точки зрения затрат и более гибкая с точки зрения топологии масштабируемость.
5. Перспектива органично использовать в будущем иные носители сигнала, например оптические.
6. Отсутствие перекрестных помех и лучшая защита от внешних помех за счет помехоустойчивого кодирования 8/10 или 128/130 и дифференциального способа передачи по линии.

Помехоустойчивое кодирование 8/10 состоит в том, что байт, имеющий 8 бит, кодируется десятью битами, т.е. с избытком. Из 1024 комбинаций 10-битного кода используются только 256 комбинаций.

Это позволяет проверить полученную информацию на недопустимые коды и обеспечить хорошую самосинхронизацию. Самосинхронизация необходима потому, что в последовательных интерфейсах синхроимпульсы не передаются и генератор синхронизации на приемной стороне подстраивается по перепадам полезного сигнала. Если байт не имеет таких перепадов (в байте все 0 или 1), генератор не получает сигналов для подстройки, что может привести к нарушению синхронности. Поэтому из 1024 комбинаций выбирают такие, в которых не может быть более трех подряд следующих 0 или 1.

Дифференциальный способ передачи заключается в том, что по одному проводу линии передается основной (фазный) сигнал $D+$, а по другому – противофазный $D-$. На приемной стороне противофазный сигнал инвертируется и складывается с основным (рис 1.10). Это позволяет в два раза увеличить амплитуду полезного сигнала и компенсировать помехи. При этом можно существенно уменьшить амплитуду передаваемого сигнала с единиц до десятых долей вольт.

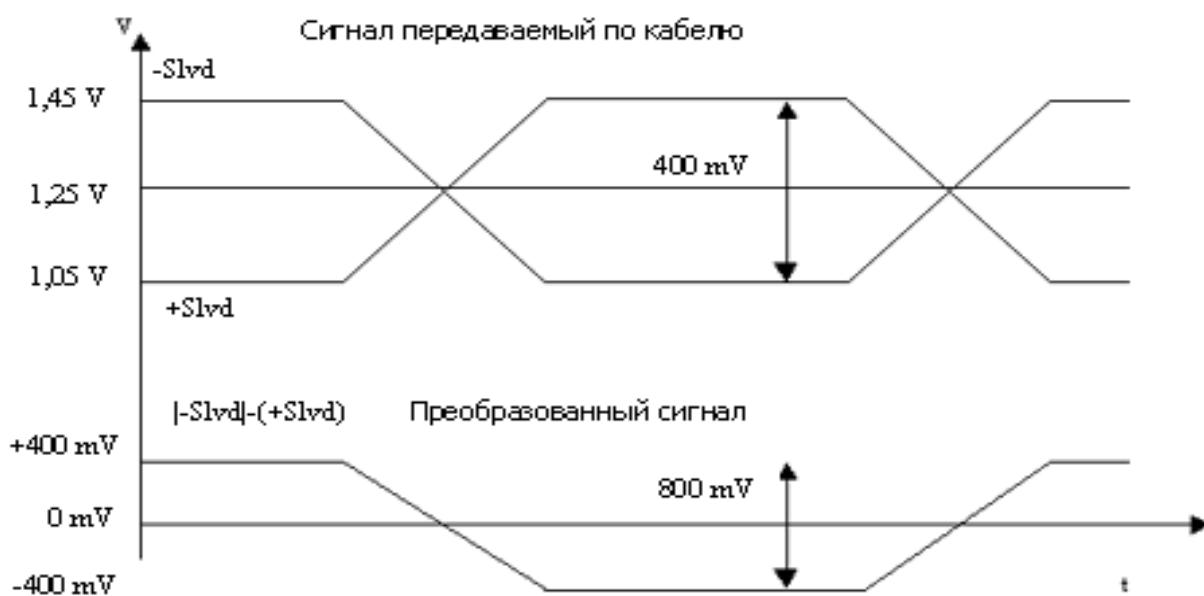


Рис. 1.10. Преобразование сигнала при дифференциальном способе передачи

Последняя особенность означает удовлетворение требований компактности и стоимости. В современных интерфейсах относительная составляющая стоимости кабелей постоянно возрастает в связи с необходимостью миниатюризации и защиты от внешних и перекрестных помех. Последовательные интерфейсы имеют более высокую

производительность по сравнению с параллельными, так как позволяют существенно (на порядок и более) повысить частоту обмена. Это связано с отсутствием перекрестных помех и необходимостью синхронизации между битами в байте или блоке данных. Простота подключения обеспечивается особенностями, перечисленными в пп. 3, 4, 5. В будущем же следует ожидать перехода на беспроводные шины, технологии подобные *UWB (UltraWideBand)*.

В настоящее время используется большое количество приборных интерфейсов. Основные параметры наиболее известных и употребляемых интерфейсов приведены в таблице.

Малые приборные интерфейсы и периферия,
подключаемая к ним

№ п/п	Название интерфейса	Тип	Скорость	Периферия
1	<i>RS232</i>	Последовательный	115 кбит/с	Мышь, клавиатура, модемы
2	<i>IEEE-1284 (Centronics, LPT-порт)</i>	Параллельный	50 кбит – 2 Мбит/с	Принтеры, сканеры, сменные накопители
3	<i>USB 1.0,2.0</i>	Последовательный	12 Мбит/с	Принтеры, сканеры, видеокамеры
4	<i>IEEE-1394 (FireWire)</i>	Последовательный	До 400 Мбит/с	Принтеры, сканеры, видеокамеры
5	<i>IDE(ATA, PATA)</i>	Параллельный	≥ 133 Мбайт/с	<i>HD, CD</i> , оптика
6	<i>SATA</i>	Последовательный	≥ 150 Мбайт/с	<i>HD, CD</i> , оптика
7	<i>SCSI</i>	Параллельный	≥ 320 Мбайт/с	<i>HD,CD</i> , оптика и др. ПУ
8	<i>Bluetooth</i>	Последовательный	–	Радиоинтерфейс
9	ИК интерфейс	То же	–	Радиоинтерфейс
10	Аудиопорт <i>AC97</i>	–//–	–	Специальный порт
11	Игровой порт	–//–	–	Специальный порт

1.3. Каналы ввода-вывода

Канал ввода-вывода (КВВ) – совокупность аппаратных и программных средств – предназначен для организации обмена, управления обменом и непосредственной передачи информации между ЦП и ПУ. Канал ввода-вывода является логическим элементом СВВ и может быть при необходимости создан, а затем «разрушен».

Основные характеристики канала ввода-вывода:

- номинальная пропускная способность (бит/с);
- нагрузочная способность (количество одновременно подключенных устройств к каналу).

Основные функции канала ввода-вывода:

1. Установление логической связи между ПУ и оперативной памятью:

- выявление приоритетного запроса от ПУ с помощью арбитра шины;
- определение маршрута для него;
- проверка работоспособности и готовности входящих в маршрут компонентов;
- если нет готовности, то выбрать альтернативный маршрут или сообщить об ошибке;
- передача всем компонентам управляющей информации, требуемой для начала обмена.

2. Непосредственно передача информации:

- а) определение текущего адреса ОП для записи или чтения;
- б) преобразование форматов данных, используемых ПУ и ОП;
- в) контроль передаваемых данных (по четности);
- г) определение особых условий в процессе выполнения операции, например, завершение передачи порции информации, необходимость завершения текущей информации.

3. Завершение обмена и «разрушение» канала:

- определение момента завершения обмена;
- определение причины завершения обмена;
- информирование ЦП об изменении состояния компонентов СВВ (КВВ);
- передача управляющей информации компонентам СВВ для приведения их в исходное состояние («разрушение»).

1.3.1. Способы обмена данными

Существуют следующие способы обмена данными:

1. Программно-управляемый обмен (асинхронный способ):

– операция чтения регистра состояния ПУ для анализа его готовности.

– ожидание готовности путем повторения предыдущего пункта.

– обмен байтом или словом данных.

Недостаток – загрузка процессора и низкая пропускная способность (155 кбит/с).

2. Программируемый ввод-вывод (*PIO – Programmed Input/Output*) построен на инструкциях блочной пересылки: *REPINS/OUTS* и осуществляет пересылку блока байтов, слов или двойных слов между ОП и портом ввода-вывода.

Для инструкции задаются:

– начальный адрес памяти;

– длина блока;

– адрес порта;

– направление пересылки.

Этот режим появился в 80286 процессоре. Скорость передачи определяется частотой процессора и шины. Этот метод применяется для обмена с дисками по интерфейсу *ATA (IDE)*. Он быстрее, чем по *DMA*. Для уменьшения скорости обмена до возможностей ПУ контроллер интерфейса аппаратно может вводить такты ожидания.

3. Обмен по каналу прямого доступа к памяти (ПДП, *DMA*).

Скорость равна 2...4 МБ/с. Почти не загружает процессор. Участвует контроллер *DMA*, подключаемый по схеме рис 1.11. Схема контроллера *DMA* приведена на рис. 1.12. В счетчик записывается количество байтов, которые необходимо передать.

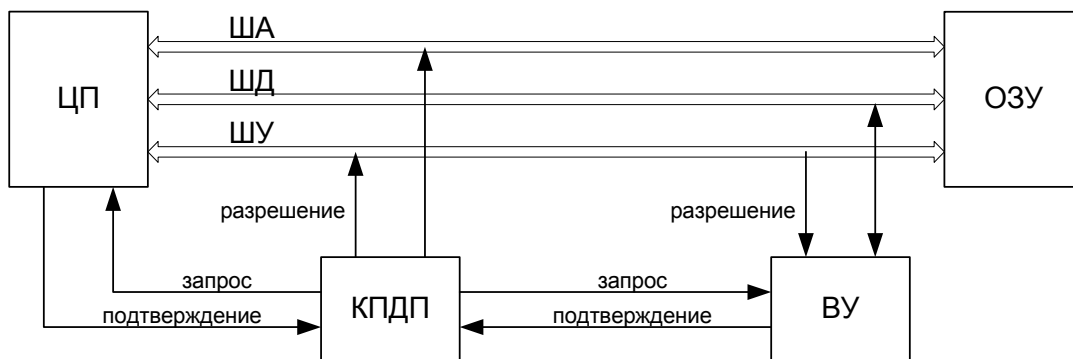


Рис. 1.11. Схема подключения контроллера *DMA*

4. Обмен в режиме прямого управления шиной (*UltraDMA*).

Такой обмен выполняется под управлением интеллектуального контроллера. ЦП загружается в наименьшей степени. Производительность выше, чем у *DMA*. Применяется в *ATA* устройствах.

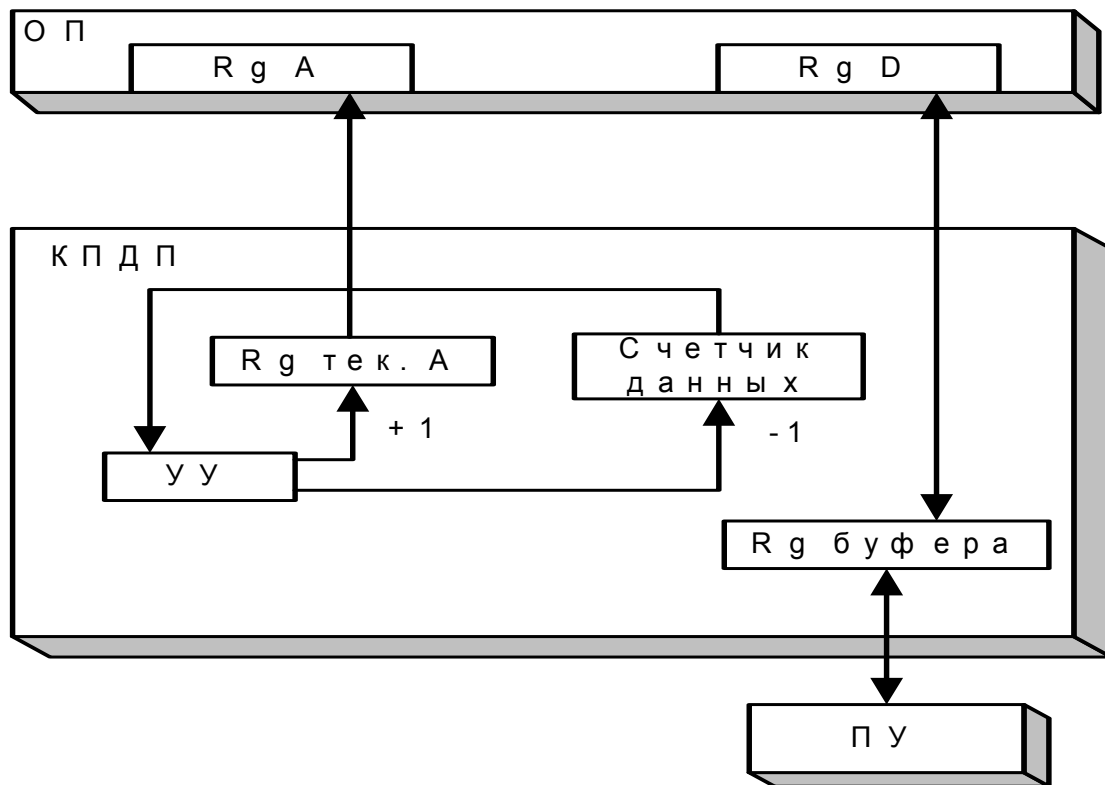


Рис. 1.12. Схема контроллера ДМА

Способы синхронизации обмена:

1. Обмен по опросу готовности. Недостаток – время тратится впустую при опросе занятого периферийного устройства.
2. Использование аппаратных прерываний (ПУ, которое хочет осуществить обмен, посылает аппаратное прерывание).
3. Поллинг (*Polling*) – опрос готовности ряда устройств по периодическому прерыванию.

Буферизация данных в ПУ

По характеру обмена ПУ можно разделить:

- на блочные устройства – обмен блоками (диски);
- поточные ПУ – идет поток информации, который поступает, а затем считывается. Это буфер типа *FIFO*;

– регистро-ориентированные ПУ – обмен производится через буферные регистры. Этот принцип применяется, например, в таких ПУ, как мышь и джойстик – в регистры заносится положение.

Возможна смесь этих видов обмена.

1.3.2. Система прерываний

Для синхронизации работы ЦП и ПУ используются прерывания и приостановки. Классификация прерываний и исключений приведена на рис. 1.13.

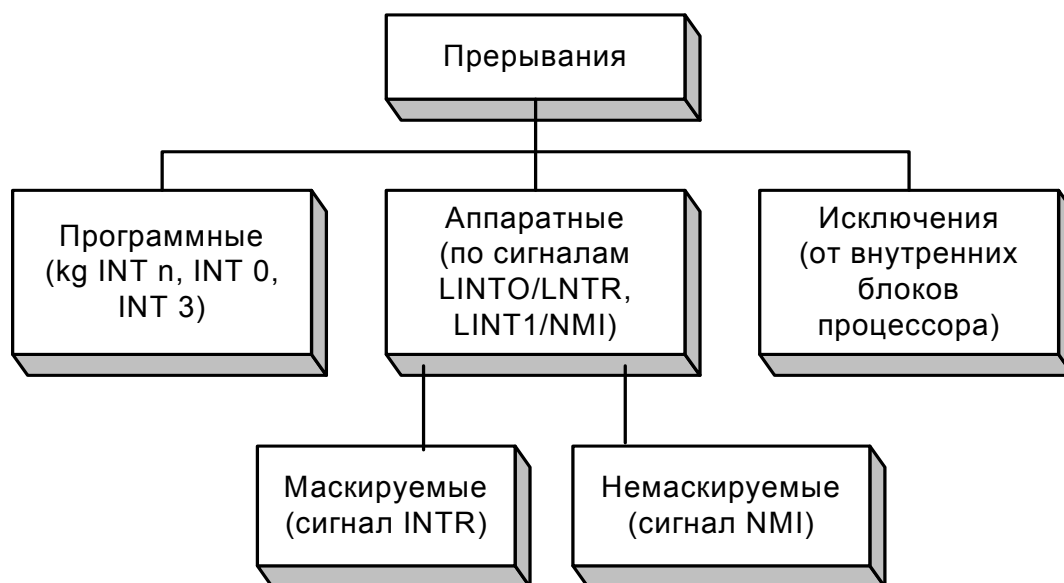


Рис. 1.13. Классификация прерываний и исключений

Виды исключений:

- ловушка (обнаруживается до выполнения программы, за ней следующая команда);
- ошибка (обнаруживается после выполнения программы, за ней предыдущая команда);
- отказ (серьезные нарушения в работе системы).

Основные функции системы прерываний:

- запоминание состояния прерываемой программы;
- переход от прерываемой программы к программе обработки прерываний;
- восстановление состояния прерванной программы.

Основные интервалы обработки прерывания (рис. 1.14):

- $t_p = t_2 - t_1$ – время реакции на прерывание;
- $t_{\Pi} = t_3 - t_2$ – время переключения (запоминания состояния прерванной программы);
- $t_{\text{оп}} = t_4 - t_3$ – время обработки прерывания;
- $t_{\text{восст}} = t_5 - t_4$ – время восстановления прерванной программы.

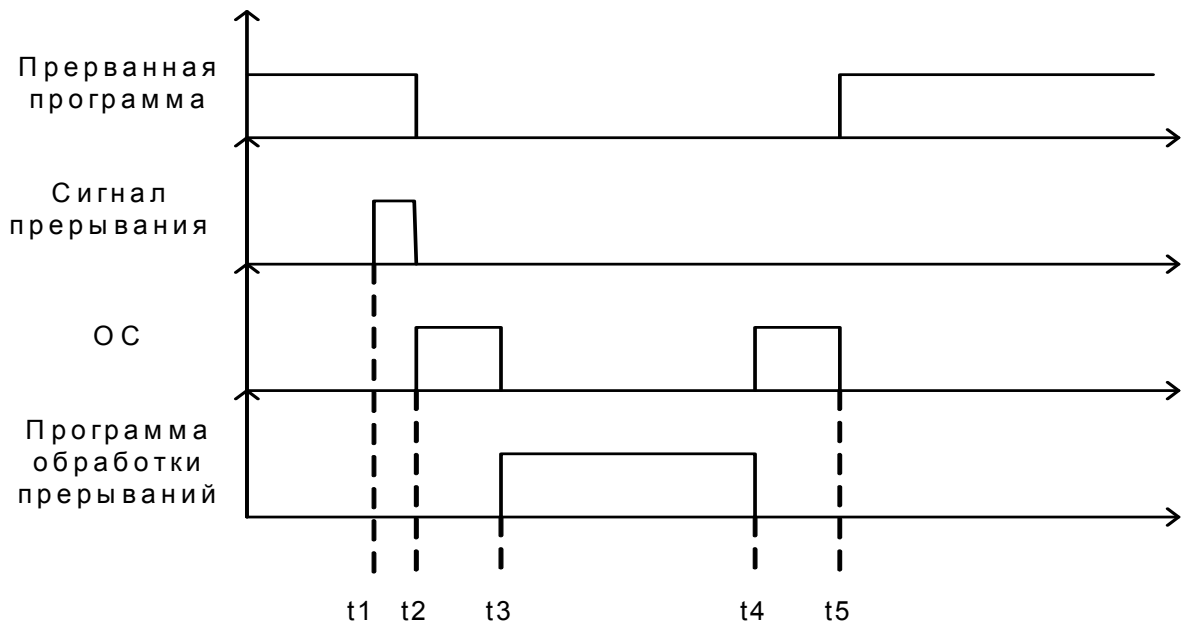


Рис. 1.14. Временная диаграмма обработки прерывания

Характеристики системы прерываний:

1. Общее число запросов прерывания (входов в СП).
2. Время реакции.
3. Затраты времени (накладное время).
4. Глубина прерывания – сколько прерываний может быть одновременно обслужено.
5. Число классов и уровней прерываний.
6. Насыщение СП (сколько прерываний более низкого приоритета могут ждать).
7. Допустимые моменты прерывания.
8. Приоритеты прерываний:
 - абсолютный (он сразу же обрабатывается);
 - относительный (он ждет конца обработки).

Способы опроса источников прерываний

Способы опроса источников прерываний (определение приоритета прерываний):

1. С опросом источников прерываний (программный подход).
Схема приведена на рис. 1.15.

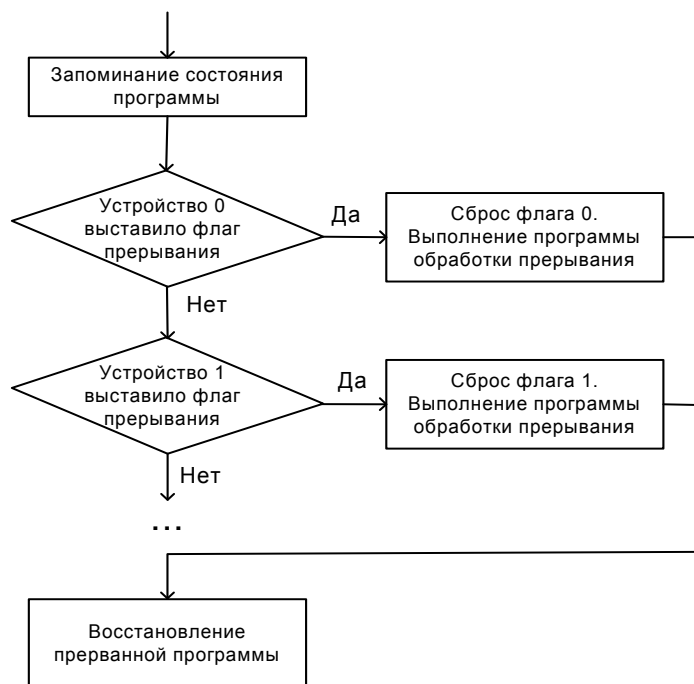


Рис. 1.15. Алгоритм программного подхода опроса источников прерываний

2. Циклического опроса источников прерывания (многотактный) – аппаратный подход (рис. 1.16).

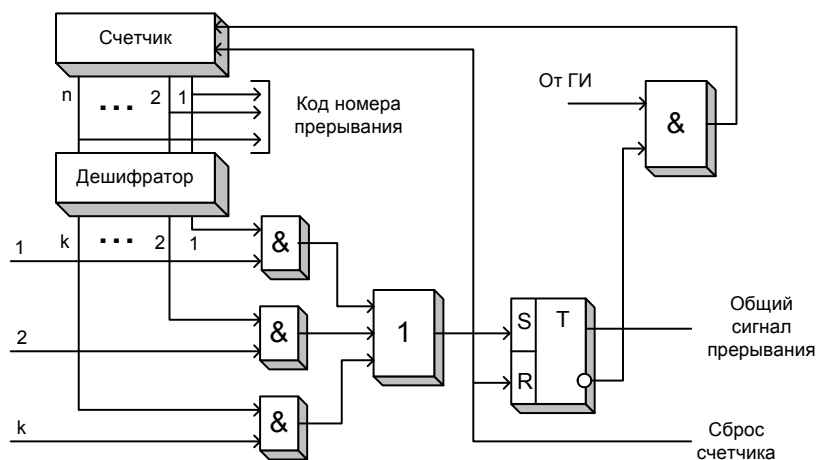


Рис. 1.16. Аппаратный подход опроса источников прерываний

Необходима большая скорость $2^n = k$.

Входы 1, 2, ..., k – сигналы запросов прерываний.

Когда придет сигнал обнаружения прерывания, счетчик перестает считать и выдает код номера прерывания.

3. Цепочечный одноктактный опрос (Дейзи-цепочка) (рис. 1.17).

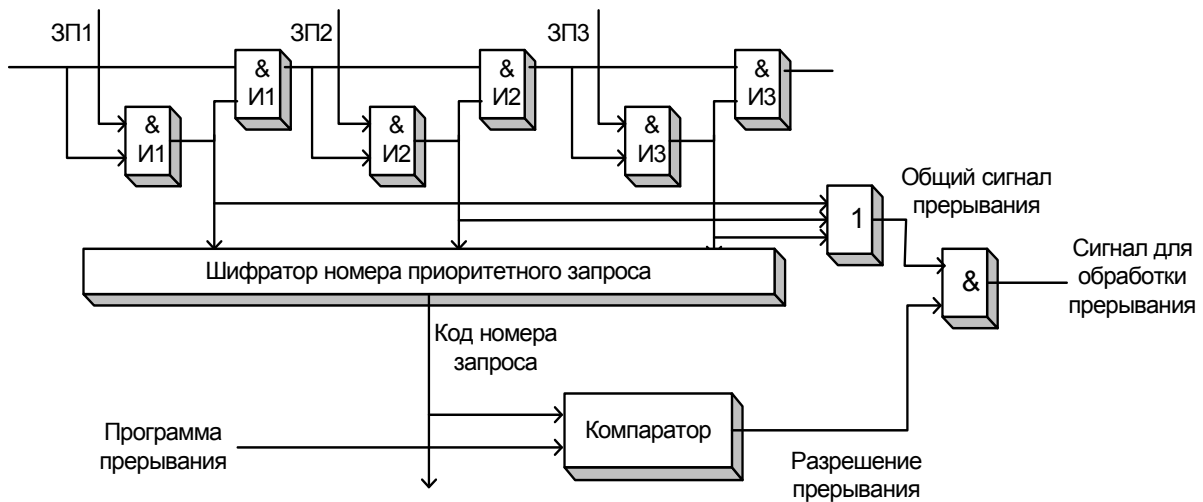


Рис. 1.17. Цепочечный одноктактный опрос

4. Векторное прерывание, применяемое в *IBM PC*.

Обработка прерываний в *IBM PC*

Порядок выполнения прерывания в *IBM PC*:

1. Запись данных из регистров *IP* и *CS* ЦП в стек.
2. Обращение к специальной таблице векторов обработки прерываний и получение адреса программы обработки прерываний.
3. Замещение данных в регистрах *CS* и *IP* новыми адресами.
4. Выполнение программы обработки прерываний (обработчика прерываний) до команды *IRET*.
5. Возврат из прерывания, для чего из стека извлекается адрес прерванной программы и помещается в регистры *CS* и *IP*.
6. Продолжение выполнения прерванной программы.

Аппаратные прерывания поступают через коммутатор и контроллеры прерываний. Коммутатор запросов является частью чипсета.

Контроллер *i8259A* обслуживает 8 прерываний. Для увеличения количества обслуживаемых прерываний до 15 соединяют каскадно 2 контроллера (рис. 1.18).

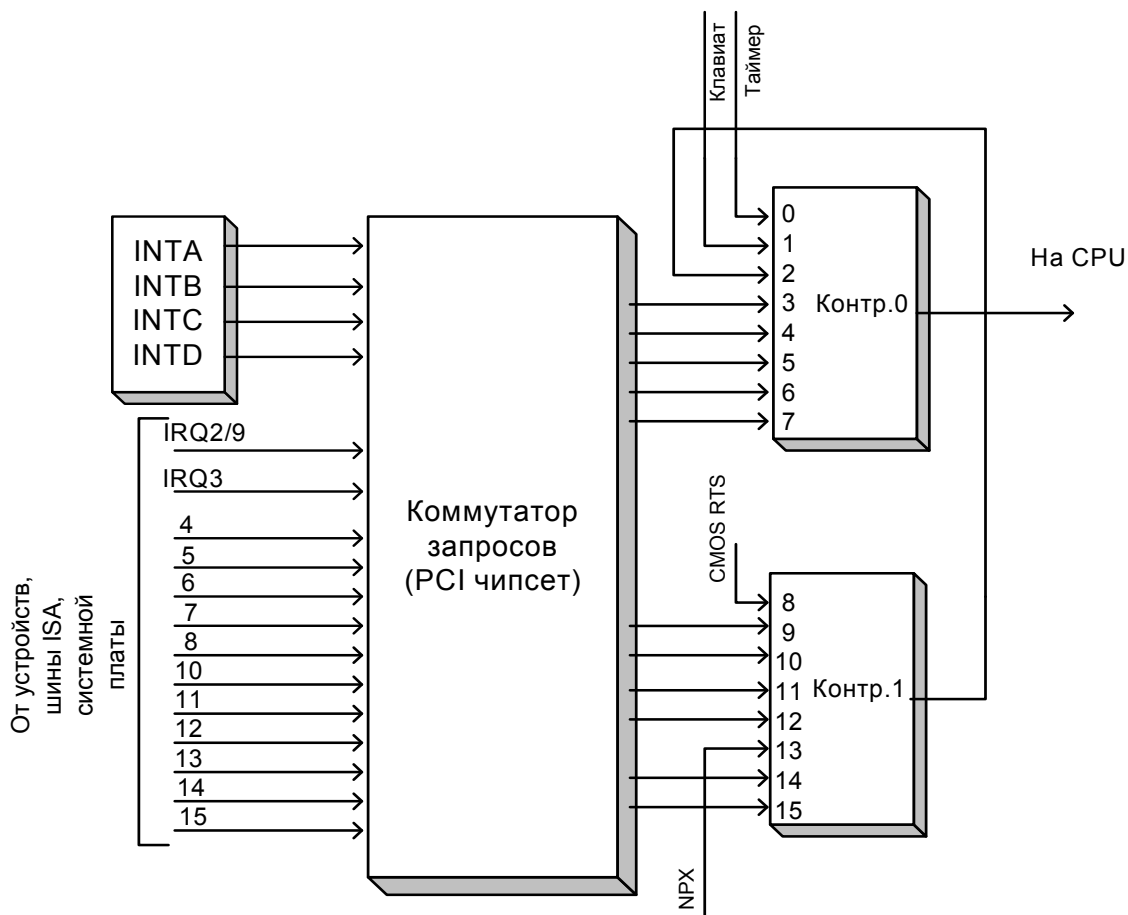


Рис. 1.18. Каскадное соединение контроллеров прерываний

1.4. Контроллеры

Назначением контроллеров является преобразование унифицированных информационных и управляющих сигналов, поступающих из канала в сигналы специальной формы, обеспечивающие работу периферийного устройства.

Контроллеры существенно различаются по степени интеллектуальности, т.е. по количеству функций, возлагаемых на контроллер.

Общая схема контроллера представлена на рис. 1.19.

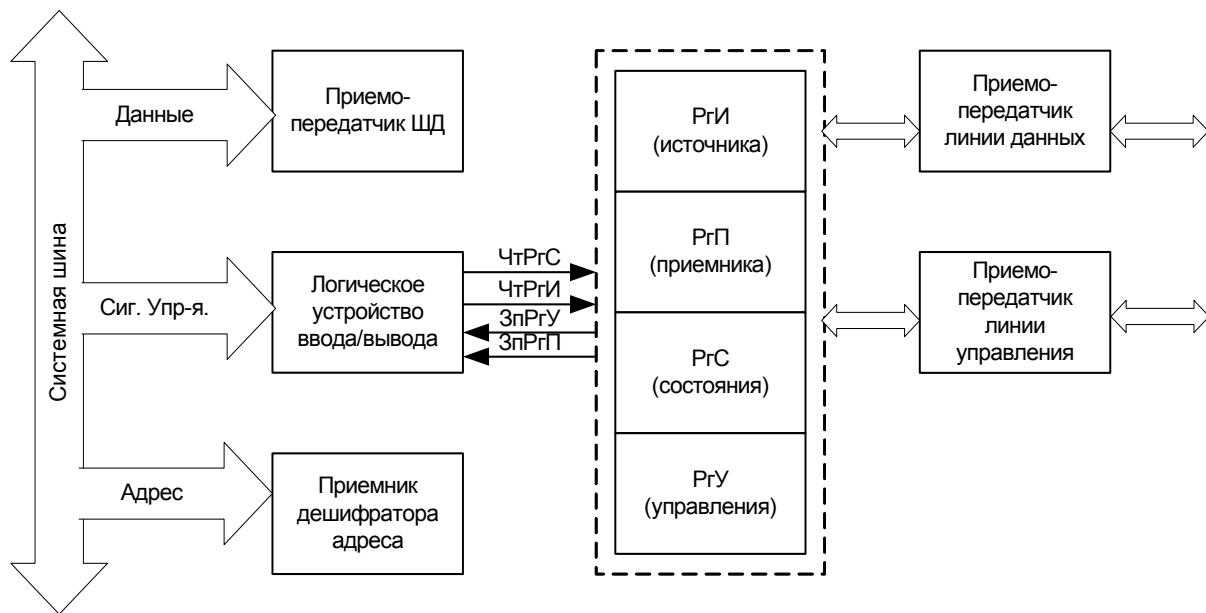


Рис. 1.19. Общая схема контроллера

Общие функции контроллера:

- дешифрация адреса периферийного устройства;
- дешифрация команд;
- выборка управляющих сигналов;
- контроль информации, поступающей с канала;
- выдача и формирование сигналов, сигнализирующих о состоянии периферийного устройства.

Адресация ПУ осуществляется с помощью задания адреса регистра контроллера. Адреса регистров: 0...256.

Факторы, определяющие структурную и функциональную организацию контроллера ПУ:

- формат данных и их особенности;
- режим работы ПУ;
- тип системного интерфейса.

Темы для закрепления знаний

1. Классы задач, решаемых на ЭВМ, и роль ПУ в этих задачах. Классификация периферийных устройств.
2. Проблемы при разработке СВВ, пути их решения.

3. Основное назначение СВВ и ее структура. Функции системы ввода-вывода.
4. Структурная схема системы ввода-вывода с общей шиной
5. Структурная схема СВВ с многоуровневой системой интерфейсов.
6. Основные функции канала ввода-вывода, его основные характеристики.
7. Классификация КВВ.
8. Общая структура ПДП. Схема включения КПДП. Схема контроллера ПДП.
9. Уровни интерфейса: классификация по назначению (месту) в системе.
10. Уровни интерфейса: классификация по конструктивному исполнению.
11. Параметры интерфейсов. Среда интерфейсов. Классификация интерфейсов по типу кабеля.
12. Структура вычислительной системы (ПК) мостовой (хабовой) архитектуры.
13. Способы и алгоритмы (протоколы) взаимодействия центрального устройства (ЦУ) и ПУ в ВС.
14. Назначение контроллеров. Общая схема и общие функции контроллера.
15. Структурная схема ПУ (внешнего устройства).
16. Адресация ПУ в системе ввода-вывода.
17. Тенденции развития интерфейсов.
18. Малые интерфейсы и периферия, подключаемая к ним.

Глава 2. ШИНЫ РАСШИРЕНИЯ ПК

2.1. Характеристики шин расширения

Шины расширения используются в ЭВМ для подключения контроллеров малых (приборных) интерфейсов и контроллеров ПУ. Они позволяют подключать и контроллеры нестандартных устройств и могут использоваться для соединения южного и северного мостов (табл. 2.1).

Таблица 2.1. Характеристики шин расширения и их интерфейсов

Шина	Производительность, МБ/с	Наличие DMA (кол-во)	Bus-Master	ACF G	Разрядность данных	Разрядность адреса	f , МГц	Применение
<i>ISA-8</i>	4	3	–	–	8	20	8	<i>PCXT</i>
<i>ISA-16</i>	8	7		–	16	24	8	<i>PCAT</i>
<i>LPC</i>	6,7	7	+	–	8/16/32	32	33	–
<i>EISA</i>	33,3	7	+	+	32	32	8,33	До i 386
<i>MCA-16</i>	16	–	+	+	16	24	10	–
<i>Mca-32</i>	20	–	+	+	32	32	10	–
<i>VLB</i>	132	–	+	–	32/64	32	33/66	–
<i>PCI</i>	132/264	–	+	+	32/64	32	33/66	–
<i>PCI-X</i>	532/ 1064	–	+	+	32/64	32	33/66	–
<i>AGP</i> (1x, 2x, 4x,8x)	266/512/1024 / 2132	–	+	+	32/64	32	66	–
<i>PCMCIA</i>	10/20	+	–	+	16	26	10	Ноутбуки
<i>CardBus</i>	132	–	+	+	32	32	33	То же
PCI-E 1,2,3	2,5; 5,0; 10,0 Gb/s	–	+	+	2	–	2,5; 5,0; 10,0 ГГц	

2.2. Шины и карты расширения мобильных ПК

В мобильных ПК применяют малогабаритные карты для шин расширения (табл. 2.2).

Таблица 2.2. Шины и карты расширения мобильных ПК

Параметры	<i>PCCard</i> 1995 г.	<i>smallPCCard</i>
Длина, мм	85,6	45,0
Ширина, мм	54,0	42,8
Высота, мм	3,3/5,0/10,5/16,0	3,3/5,0/10,5/16,0
Коннектор	Штырьковый	Штырьковый
Число контактов	68	68
Интерфейсы	Память, ввод-вывод и <i>CardBUS</i>	Память, ввод-вывод

Интерфейсы *PCM CIA*, *PC Card* и *CardBUS*

Интерфейсная память обеспечивает 8- и 16-битные с t цикла = 100 нс, то есть $V = 10 - 20$ МБ/с. Толщина разная, но меньшие встают в большие гнезда. Большинство адаптеров поддерживают «горячее подключение». $U_{пит} = 5, 3,3$ В. Введен механический ключ для недопущения установки карт 3,3 В в разъем 5 В. Определены контакты 43 ($VS1\#$) и 57($VS2\#$) для выбора. $U_{пит} = 5$ В – оба свободны. $U_{пит} = 3,3$ В – $VS1\#$ – заземлен, $VS2\#$ – свободен.

PC Card

Выпускаются различные устройства для подключения:

- память;
- устройства хранения;
- коммуникационные устройства;
- интерфейсные порты;
- игровые адаптеры;
- мультимедийные устройства и т.п.

Они дороже, чем для обычных систем.

Слоты *PCCard* подключаются к системной шине блокнотных ПК через мост с внутренней шиной *PCI*. Это мост *PCI – PCCard*. Могут быть и слоты *smallPCI (SPCI)*, но они не доступны без вскрытия корпуса ПК. Разъем – 68 контактов.

2.3. Интерфейс *LPC*

Интерфейс *LPC* (*LowPinCount* – малое число выводов) заменяет шину *ISA* или *X-BUS*.

К ней подключаются:

- контроллер НГМД;
- последовательный и параллельный порты;
- клавиатура;
- аудиокодек;
- *BIOS*.

Этот интерфейс обеспечивает те же циклы обращения, что и *ISA*:

- чтения-записи ОП, ввода-вывода;
- *DMA*;
- прямое управление шиной.

Устройства, подключенные к этой шине, могут вызвать прерывания для согласования скоростей работы. В отличие от *ISA* обеспечивает 32-битную адресацию памяти. Порты – адресация 16 бит.

Интерфейс синхронизирован с *PCI*, но устройства могут вызвать произвольное число тактов ожидания.

Этот интерфейс программно-прозрачный, т.е. он не требует драйверов (рис. 2.1).

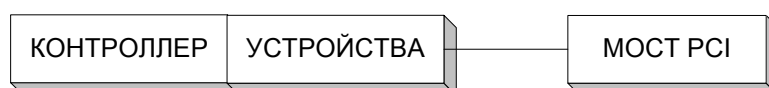


Рис. 2.1. Структура интерфейса *LPC*

Производительность как у *ISA*. При наличии буферов *FIFO* интерфейс наиболее выгодно использовать в режиме *DMA* – для порта *LPT*. Он занимает 47 % полосы интерфейса и измеряется $V = 2$ МБ/с; далее ИК-порт – 11,4 % полосы интерфейса при $V = 4$ МБ/с; остальные устройства (НГМД, *COM*-порт, аудиокодек) доводят общую полосу до 75 %.

Сигналы интерфейса *LPC*

Основные сигналы:

- *LAD*[3:0] – двунаправленная мультиплексированная ШД;

- *LFRAM#* – индикатор начала и конца цикла, управляемый хостом;
- *LRESET#* – сигнал сброса, тот же, что *RST#* на шине *PCI*;
- *LCLK* – синхронизация (33 МГц), тот же сигнал, что *CLK* на *PCI*.

Дополнительные сигналы:

- *LDRQ#* – кодированный запрос *DMA/BUSMaster* от ПУ;
- *SERIRO* – линия запросов прерываний (в последовательном коде), используется если нет стандартного *IRQ* в стиле *ISA*;
- *CLKRUN#* – сигнал, используемый для указания на остановку шины. Требуется для устройств, нуждающихся в *DMA* и *BUSMaster*;
- *PME* – сигнал о событиях системы, управление потреблением питания (как в *PCI*);
- *LPCPD#* – *PowerDown*- указание от хоста устройства на подготовку к выключению питания;
- *LSMI#* – запрос прерывания *SMI#* для повтора инструкции ввода-вывода.

Протокол *LPC* показан на рис. 2.2.

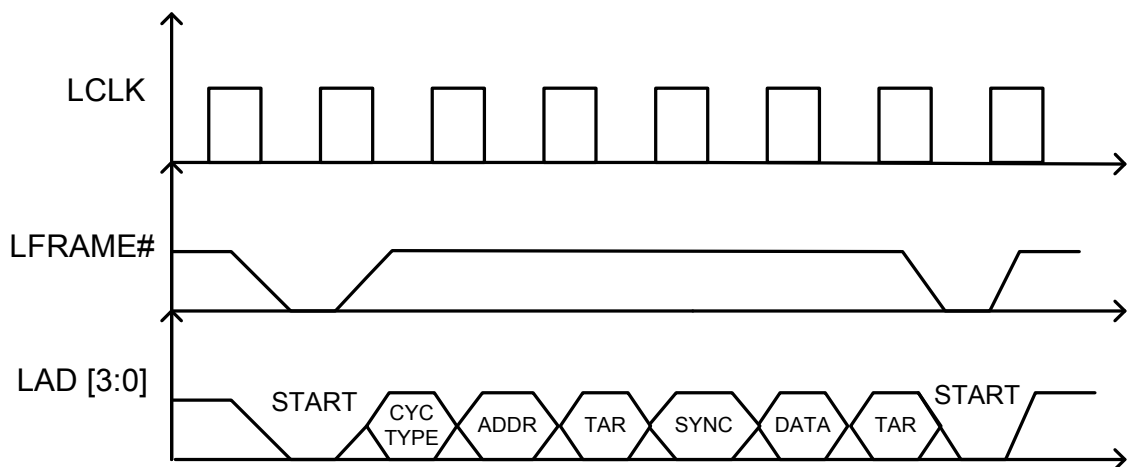


Рис. 2.2. Протокол *LPC*

Описание полей:

1. В поле ***START*** задается код действия:
 - 0000 – начало цикла обращения хоста к ПУ;
 - 0010 или 0011 – предоставление доступа ведущему устройству;

– 1111 – принудительное завершение цикла (*abort*).

2. **CYCTYPE** задает тип и направление передачи.

3. **TAR** (*TurnAround*) служит для смены владельца шины *LAD*, занимает 2 такта.

4. **ADDR** – для передачи адреса. В цикле ОП оно занимает 8 тактов, в цикле ВВ – 4 такта.

5. **DATA** – данные, на каждый байт – 2 такта. Начало с младшей тетрады (и с младших байтов, если их несколько).

6. **SYNC** служит для введения тактов ожидания устройством, к которому выполняется обращение. Оно обеспечивает контроль передачи, введение тактов ожидания и механизм таймаутов.

Начав цикл, хост считает поле синхронизации. Если в течение трех тактов адресованное устройство не отвечает, то хост считает, что его нет, и прекращает транзакцию. Если приходит код короткого ожидания, хост ждет его смены на готовность или ошибку. Но после 8 тактов ожидания он прервет транзакцию по таймауту.

Код длинного ожидания возлагает ответственность за отсутствие зависания на адресованное устройство.

При прямом управлении поле *SYNC* выставляется хостом и устройство должно ждать.

Последовательность полей при обращении хоста к памяти или портам показана на рис. 2.3.

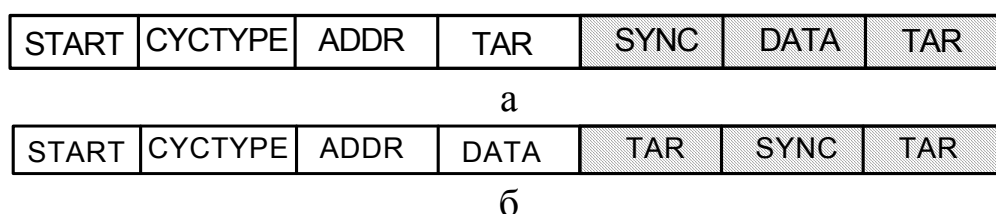


Рис. 2.3. Последовательность полей при обращении хоста к ОП и портам ввода-вывода: а – чтение; б – запись

Заштрихованные поля устанавливает устройство, к которому идет обращение.

Для чтения памяти требуется 21 такт (0,63 мкс), в том числе 5 тактов *SYNC* для доступа к памяти.

Для записи памяти требуется 17 тактов (0,51 мкс), в том числе 1 такт *SYNC*.

Обращения к портам требуют по 13 тактов за счет более короткой адресации и отсутствия тактов ожидания.

Передача данных по DMA. Передача данных по *DMA* выполняется под управлением хоста, но отличается от обращения к памяти и портам. Появляются поля:

- *SIZE* – определяет размер передачи (1 – 4 Б);
- *CANNEL* – для передачи хостом номера канала *DMA* (биты [2:0]) и признака конца цикла *TC* (бит 3).

Циклы чтения памяти в зависимости от длины (1, 2, 4 Б) без тактов ожидания занимают 11, 18 и 32 такта ($V = 3,03, 3,7, 4,17$ МБ/с соответственно).

Циклы записи занимают 11, 14 или 20 тактов ($V = 3,03, 4,76, 6,67$ МБ/с соответственно).

Прямое управление шиной. Ведущее устройство запрашивает, как и *DMA*, но указывая зарезервированный номер канала 4(100). Требуется 25, 27 или 31 такт (из них 6 – *SYNC*) для обращения к памяти ($V = 1,33, 2,47$ и $4,3$ МБ/с). При обращении к портам требуется 21, 23 или 27 тактов ($V = 1,59, 2,9$ и $4,94$ МБ/с).

Электрические сигналы интерфейса. *LAD*[3:0], *LFRAME*#, *LDRQ*#, *SERIRQ* соответствуют спецификации *PCI 2.1* для питания 3,3 В. Остальные сигналы в зависимости от системной платы будут с уровнем 5 или 3,3 В.

Конфигурирование устройств *LPC* не предусматривает протоколов *PCI* или *ISA*, так как все устройства известны системной *BIOS*.

Для обращения к устройствам *LPC* хост должен декодировать их адреса и направлять обращения к ним на контроллер *LPC*.

2.4. Шина *PCI*

Шина *PCI* (*Peripheral Component Interconnect*) разрабатывалась в расчете на *Pentium*-системы, но хорошо сочеталась с процессором 486. Является четко стандартизированной, высокопроизводительной и надежной шиной расширения. Шина является синхронной – фиксация всех сигналов выполняется по положительному перепаду (фронту) сигнала *CLK*. В каждой транзакции (обмене по шине) участвуют два устройства – инициатор обмена (*Initiator* или *MASTER*, иницирующее устройство) и целевое устройство (*Target* или *SLAVE*). Одним из

существенных преимуществ шины *PCI* является ее процессорная независимость – способность работать на различных платформах (например, *IBMPC* и *AppleMacintosh*).

Описание функционирования шины *PCI*. Шина *PCI* все транзакции трактует как пакетные: каждая транзакция начинается фазой адреса, за которой может следовать одна или несколько фаз данных (рис. 2.4).

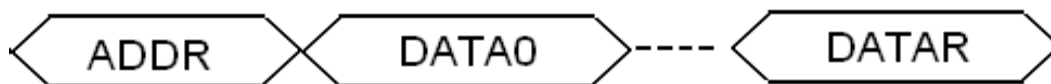


Рис. 2.4. Последовательность полей в интерфейсе *PCI*

Для адреса и данных используются общие мультиплексированные линии $AD[31:0]$. Четыре мультиплексированные линии $C/BE#[3:0]$ используются для кодирования команд в фазе адреса и разрешения байт в фазе данных. В начале транзакции инициирующее устройство активирует сигнал $FRAME\#$, по шине $AD[31:0]$ передается целевой адрес, а по линиям $C/BE#[3:0]$ – информация о типе транзакции (команде). Адресованное целевое устройство отвечает сигналом $DEVSEL\#$, после чего инициирующее устройство может указать на свою готовность к обмену данными сигналом $IRDY\#$. Когда к обмену данными будет готово и целевое устройство, оно установит сигнал $TRDY\#$. Данные по шине $AD[31:0]$ могут передаваться только при одновременном наличии сигналов $IRDY\#$ и $TRDY\#$. С помощью этих сигналов инициирующее и целевое устройства согласуют свои скорости, вводя такты ожидания. Количество фаз (циклов) данных в пакете заранее не определено, но перед последним циклом инициирующее устройство при введении сигнала $IRDY\#$ снимает сигнал $FRAME\#$.

Транзакция может быть прервана по инициативе целевого устройства. Для этого оно может ввести сигнал $STOP\#$. Возможны три типа прерывания:

1) Отключение (*Disconnect*) – сигнал $STOP\#$ вводится во время активности $TRDY\#$. В этом случае транзакция завершается после фазы данных.

2) Отключение с повтором (*Disconnect/Retry*) – сигнал $STOP\#$ вводится при пассивном состоянии $TRDY\#$, и последняя фаза данных

отсутствует. Является указателем инициирующему устройству на необходимость повтора транзакции.

3) Отказ (*Abort*) – сигнал *STOP#* вводится одновременно со снятием *DEVSEL#* (в предыдущих случаях во время появления сигнала *STOP#* сигнал *DEVSEL#* был активен). В этом случае последняя фаза данных тоже отсутствует, но повтор не запрашивается.

Средством повышения надежности (достоверности) является применение контроля паритета: линии *AD[31:0]* и *C/BE#[3:0]* в фазе адреса и в фазе данных защищены битом паритета *PAR* (количество единичных бит этих линий, включая *PAR*, должно быть четным). Действительное значение *PAR* появляется на шине с задержкой в один такт относительно линий *AD* и *C/BE*. При обнаружении ошибки целевым устройством со сдвигом на такт вырабатывается сигнал *PERR#*. В подсчете паритета при передаче данных учитываются все байты, включая и недействительные (отмеченные высоким уровнем сигнала *C/BE#*). Состояние бит и недействительных байт данных во время фазы данных должно оставаться стабильным.

Каждое целевое устройство имеет инкрементный механизм слежения за длительностью циклов (*IncrementalLatencyMechanism*), который не позволяет интервалу между соседними фазами данных в пакете превышать 8 тактов шины. Если целевое устройство не успевает работать в таком темпе, оно обязано остановить транзакцию.

В циклах обращения к портам ввода-вывода для адресации любого байта используются все линии *AD[31:0]*. В циклах конфигурационной записи/считывания устройство выбирается индивидуальным сигналом *IDSEL#*, конфигурационные регистры выбираются двойными словами по линиям *AD[7:2]*, при этом *AD[1:0] = 00*. Сигнал выборки *IDSEL#* воспринимается устройством только в фазе адреса, поэтому обычно в качестве него используют старшие биты адреса с позиционным кодированием адреса устройства.

Адресация устройств в *PCI*. Адрес содержит три составляющие.

1. Номер шины (до 4 шин *PCI*).
2. Номер устройства (до 21 устройства).
3. Номер функции (до 8 функций).

Прерывания *PCI*. Так же, как прерывания в шине *ISA*, прерывания в шине *PCI* обрабатываются с помощью двух контроллеров

i8259A. В *PCI*, в отличие от *ISA*, нет отдельного понятия *DMA*, есть другое – прямое управление шиной *UltraDMA*.

Электрический интерфейс *PCI*. Имеются три типа карт и портов *PCI* в зависимости от уровня питания буферных регистров:

1. $U_n = 3,3$ В – на этой карте ключи расположены в контактах 22 – 25 (слот-перемычка, карта-вырез).

2. $U_n = 1,5$ В – ключ в контактах 42 – 45.

3. Универсальная карта, которая может поддерживать любое из этих напряжений.

Слот не имеет перегородок, а карта имеет два выреза.

Системная плата узнает U питания карты по сигналу *TYPE_DET#*.

Универсальная карта узнает о $U_{пит}$ по уровню напряжения на контактах *Vddg*. На картах 3,3 В контакт *Vddg* свободен, а на картах 1,5 В и универсальных – заземлены.

Мосты *PCI*. Главный мост – для подключения к системной шине. Одноранговый мост – для соединения двух шин *PCI*. Возможно создание контроллеров для нестандартных устройств.

Интерфейс *PCI-X*. Интерфейс *PCI* получил развитие в виде *PCI-X* и *PCI-X 2.0*. Он имеет две модификации: *PCI-X 266* и *PCI-X 533*. Первая предусматривает частоту шины 266 МГц и скорость обмена данными до 2,1 ГБ/с, вторая – частоту шины 533 МГц и скорость обмена до 4,2 ГБ/с. Пользователи компьютеров, поддерживающих первую версию интерфейса *PCI-X*, смогут обновить его до версии 2.0.

2.5. Интерфейс *AGP*

Интерфейс *AGP* имеет несколько вариантов, различающихся производительностью: *Version 2 = 2x* (1998 г.), спецификация 8x (2004 г.). Этот порт – двухточечный. Он используется для подключения видеокарты.

Высокая производительность обеспечивается следующими особенностями:

1. Конвейеризация обращения к памяти (рис. 2.5).

2. Сдвоенная четырехкратная и восьмикратная скорость передачи данных:

1x – передача только по переднему фронту;

2x – по фронту и по спаду (*Side Band Control*) (*SB-STB*);
4x – это еще один сдвинутый сигнал по фазе (инверсный) – *SB-STB#*.

3. Демультимплексирование, или разделение шины адреса данных. Шина адреса в режиме демультимплексирования представляет 8 линий *SBA*, по которым за 3 такта синхронизации передается 4 байта адреса, длина запроса – 1 байт, команда – 1 байт.

Ранние версии *AGP* поддерживали не все возможности:

- может не работать конвейеризация;
- может не работать с текстурами.

Сигналы *AGP*. Все сигналы шины *PCI* и некоторые дополнительные сигналы.

Устройства подключения к *AGP* могут быть двойками:

- функции *AGP*;
- функции *PCI*.

Устройства должны выполнять роль ведущих *AGP* и могут выполнять роль ведущего *AGP* и ведомого *PCI*, а также быть ведущим устройством *PCI*. В режиме *AGP* устройство доступно только ОЗУ. В режиме *PCI* доступны память, ввод-вывод, конфигурационное пространство.

Транзакции. В режиме *PCI* транзакции инициализируются ускорителем. Обмен начинается с сигнала *FRAME#*. В этом режиме шина *AD* занята. Транзакции чтения занимают больше тактов, чем транзакции записи, так как в режиме чтения надо обеспечить подключение памяти (запись без ожидания подтверждения).

Конвейерные транзакции *AGP* (см. рис. 2.5) инициализируются ускорителем и адресованы к системе ОЗУ, эти транзакции ставятся в очередь на обслуживание и исполняются по мере возможности в зависимости от следующих причин:

- от уровня приоритета;
- от порядка поступления запроса;
- от готовности данных.

Транзакции, адресуемые к устройству *AGP*, обрабатываются им как ведомым устройством *PCI*, но в старых версиях (от 1 до 4) может быть быстрая запись в локальную память (*Fast Write*). В восьмой версии эта возможность не реализуется. *Fast Write* инициализируется процессором.

Состояния порта:

– *IDLE* – покой;
– *TABLE* – передача данных в виде конфигурированных транзакций;

– *AGP* – постановка в очередь команд *AGP*;

– *PCI* – выполнение транзакций в режиме *PCI*.

Из состояния покоя порт может вывести запрос транзакций *PCI* как со стороны графического ускорителя, так и со стороны системной платы и запроса *AGP* только со стороны ускорителя.

Состояние *PCI* – транзакция выполняется целиком от передачи адреса и транзакционной команды до передачи такта.

Состояние *AGP* – ведущее устройство передает команды и адрес для транзакций или по сигналу *PIPE#*, или через порт *SBA*. Команда ставится в очередь, задается новый адрес.

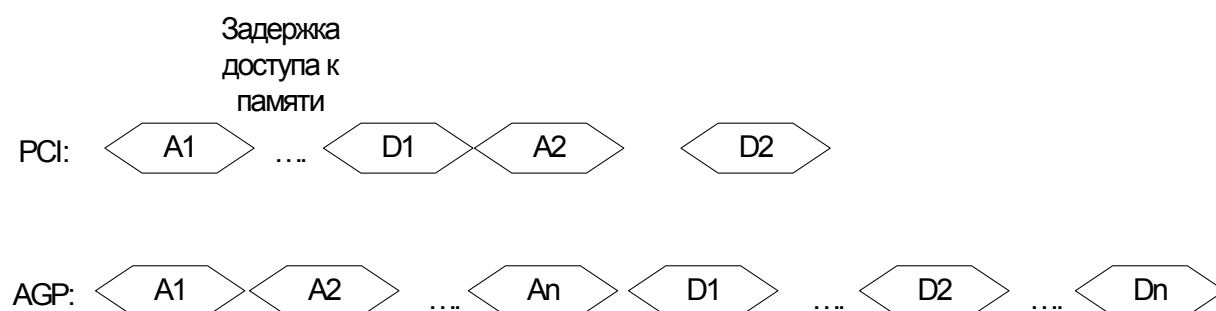


Рис. 2.5. Структура транзакций в разных состояниях порта

В очереди может быть 256 адресов и команд. Когда в порте есть необслуженные команды, прерывание происходит на границе данных.

Отличие транзакций *AGP* от *PCI*:

1. Фаза данных отделена от фазы адреса.
2. Используется собственный набор команд.
3. Транзакции адресуются только к системной памяти, используя то же пространство физических адресов.
4. Транзакции могут иметь длину, кратную 8 байтам. Длина транзакции явно указывается в запросе.
5. Конвейерные запросы не гарантируют когерентности ОП. Для операций, требующих когерентности, должен использоваться режим *PCI*.

Способы передачи команд AGP (каждый способ определяется текущей конфигурацией):

- запросы вводятся по шине адреса данных $AD[31:0]$ и шине $C/BE[3:0]$. С помощью сигнала $PIPE\#$ по каждому фронту CLK ведущее устройство передает очередное двойное слово запроса вместе с кодом команд. При подаче код команды кодируется C/BE . На шине адреса помещается $AD[31:3]$. При этом определяется 9 команд;

- внеполосная передача – команды передаются через внеполосные линии адреса $SBA[7:0]$. Внеполосность означает, что эти сигналы используются вне зависимости от занятости шины AD . По этой шине SBA передаются 4-битные послылки 4 типов. Каждая посылка передается по фронту и по спаду за 2 приема синхронизации.

SBA зависит от режима:

- 1x – каждая часть передается по фронту;
- 2x – для SBA используется отдельный строб. $SB-STB$ по его спаду передается старшая часть, а по следующему фронту – младшая часть $f(SB-STB)=f(CLK)$, но фаза различается (сдвиг по фазе);

- 4x – используется дополнительный инверсный строб $SB-STB\#$ (инверсный, срабатывает по нижнему уровню). Старшая – по $SB-STB$, а младшая – по $SB-STB\#$. Частота стробов в 2 раза выше CLK . За счет этого – кратное увеличение скорости.

Передача данных в AGP. Порядок данных соответствует порядку ранее пришедших команд от ускорителя (запросов). Имеются и независимые очереди для разных команд, чтение низкоприоритетных и высокоприоритетных команд, и запись низкоприоритетных и высокоприоритетных команд. Фазы использования команд разных очередей могут чередоваться (256 команд в очереди).

Приоритеты. Запросы AGP с высоким приоритетом выше, чем запросы ЦП и ведущих устройств шины PCI . Запросы AGP с низким приоритетом ниже, чем ЦП, но выше чем от остальных устройств.

Конфигурирование устройств в AGP. Как и у PCI выполняется через регистры конфигурационного пространства. Регистр состояния AGP сообщает свойства порта, используя поля от $[31:0]$, адрес этого регистра CAP_PTR+8 .

Электрический интерфейс. Напряжение $U_{пит} = 5; 3,3$ В. Карты могут быть либо на эти напряжения, либо универсальные. Идентификация как в PCI .

Существует спецификация *AGP_PRO*, которая описывает в 4 раза более мощный коннектор. Он имеет дополнительные контакты с обеих сторон обычного коннектора *AGP*: *GND* и $U_{п1} = 3,3$ В, $U_{п2} = 12$ В. Обычная карта может быть установлена в *AGP_PRO*, но не наоборот.

Отличия *AGP 8x*

1. Введен новый режим передачи по шинам *SBA* [7:0] и *AD*, обеспечивает пиковую производительность $V = 2,132$ ГБ/с;
2. Исключены команды длинного чтения и записи.
3. Исключают команды высокого приоритета, то есть приоритет одинаков для всех команд.
4. Исключена подача команд с помощью сигнала *PIPE*.
5. Меры по обеспечению когерентности кэша и данных.
6. Несколько изменены протоколы передаваемых данных.
7. Дополнительная возможность поддержки нескольких портов
8. Введение поддержки изохронных передач.
9. Возможность поддержки разных размеров страниц, описываемых в памяти, и обеспечение когерентности при обращении к отдельным страницам.

2.6. Интерфейс *PCI-Express*

Стандарт *PCI-Express (PCI-E)* впервые был представлен на форуме *Intel* в сентябре 2003 г. и по словам разработчиков должен был заменить *AGP*, ведь *PCI-E* имеет значительные преимущества перед *AGP*.

В наше время последовательные шины становятся более удобными не только с точки зрения конечного потребителя, но и с точки зрения банальной выгоды. Более подробно преимущества последовательных шин и интерфейсов рассмотрены в п.2.5.

Эти преимущества хорошо видны на примере новой стандартной системной шины *PCI-Express*.

***PCI-Express* – основные моменты**

Последовательная системная шина общего назначения. Имя – *PCI-Express*, на стадии проектирования была также известна как *3GIO* (ввод-вывод третьего поколения) или по кодовому имени рабочей группы и проекта «*Arahoe*», причем оба названия (*3GIO* и *PCI-Express*) являются зарегистрированными торговыми марками *PCISIG*.

Развитием стандарта *PCI-Express* занимается организация *PCI Special Interest Group*.

Дата рождения – 22 июля 2002 г. – опубликована базовая спецификация протокола и сигнального уровня, а также базовая спецификация на форм-фактор, энергопотребление карт и разъемы. Фактически эта шина представляет совокупность независимых самостоятельных последовательных каналов передачи данных. Сигнальный уровень 0,8 В. Каждый канал состоит из двух дифференциальных сигнальных пар (рис. 2.6), для чего необходимо только 4 контакта.



Рис. 2.6. Структура каналов передачи данных

Используется избыточное защищенное от помех кодирование – каждый байт при передаче представляется десятью битами. Пропускная способность 2,5 ГБ (250 МБ) в секунду для одного канала в каждом направлении одновременно (полный дуплекс), однако следует учесть, что эффективная скорость передачи данных за вычетом избыточного кодирования составляет 2 ГБ (200 МБ) ровно.

Стандартизированы 1, 2, 4, 8, 16 и 32-канальные варианты (до 6,4 эффективных гигабайт в секунду соответственно при передаче в одну сторону и вдвое больше при передаче в обоих направлениях). При передаче данных они передаются параллельно (но не синхронно) по всем доступным каналам. Вся контрольная информация передается по тем же линиям, что и данные, используется стек протоколов из нескольких уровней, включая маршрутизацию данных:

Стандарт предусматривает и альтернативные носители сигнала, такие как оптические волноводы, возможность распознавания и использования альтернативных (улучшенных) протоколов обмена, возможность динамического подключения и конфигурации устройств.

Преимущества *PCI-Express*

Сравнивая возможности господствовавшей многие годы параллельной шины *PCI* и архитектуру *PCI-Express*, можно выделить пять наиболее значимых преимуществ последней.

– *Высокая производительность* – повышение пропускной способности версии *x1* как минимум вдвое по сравнению с *PCI*, возможность линейного наращивания производительности путем линейного расширения шины. Помимо этого *PCI-Express* является реально дуплексной шиной.

– *Упрощение разводки периферии* – стандартизация там, где ранее использовались всевозможные варианты *PCI - AGP, PCI-X* и др.; снижение комплексных затрат на разработку и внедрение систем.

– *Уровневая архитектура* – основные затраты на развитие *PCI-Express* в дальнейшем ложатся лишь на разработку соответствующей обвязки, можно экономить на возможности работы с прежним программным обеспечением.

– *Современное поколение периферийных устройств* – *PCI-Express* позволяет реализовать новые возможности обмена данными и мультимедийным контентом за счет изохронной природы передачи (т.е. разнесения отдельных частей сигнала по времени).

– *Простота использования* – производить апгрейд и доработку систем устройствами *PCI-Express* значительно легче. Теперь появится возможность использовать *PCI-Express*-карты с «горячим» подключением.

PCI-Express* – ключевые отличия от *PCI

Подробнее остановимся на ключевых отличиях *PCI-Express* от *PCI*:

1. Как уже неоднократно упоминалось, новая шина последовательна, а не параллельна. Основные преимущества — снижение стоимости, миниатюризация, лучшее масштабирование, более выгодные электрические и частотные параметры (нет необходимости синхронизировать все сигнальные линии).

2. Спецификация разделена на целый стек протоколов, каждый уровень которого может быть усовершенствован, упрощен или заменен, не сказываясь на остальных. Например, может быть использован иной носитель сигнала или упразднена маршрутизация в случае выделенного канала только для одного устройства. Могут быть добавле-

ны дополнительные контрольные возможности. Развитие такой шины происходит гораздо менее болезненно – увеличение пропускной способности не требует изменять контрольный протокол и наоборот. Это позволит быстро и удобно разрабатывать адаптированные варианты специального назначения.

3. В изначальной спецификации заложены возможности горячей замены карт.

4. Заложены возможности создания виртуальных каналов, гарантирования пропускной полосы и времени отклика, сбора статистики *QoS* (*Quality of Service* – качество обслуживания).

5. Заложены возможности контроля целостности передаваемых данных (*CRC*).

6. Заложены возможности управления питанием.

Самый простой вариант перехода на *PCI-Express* для стандартных по архитектуре настольных систем показан на рис. 2.7.

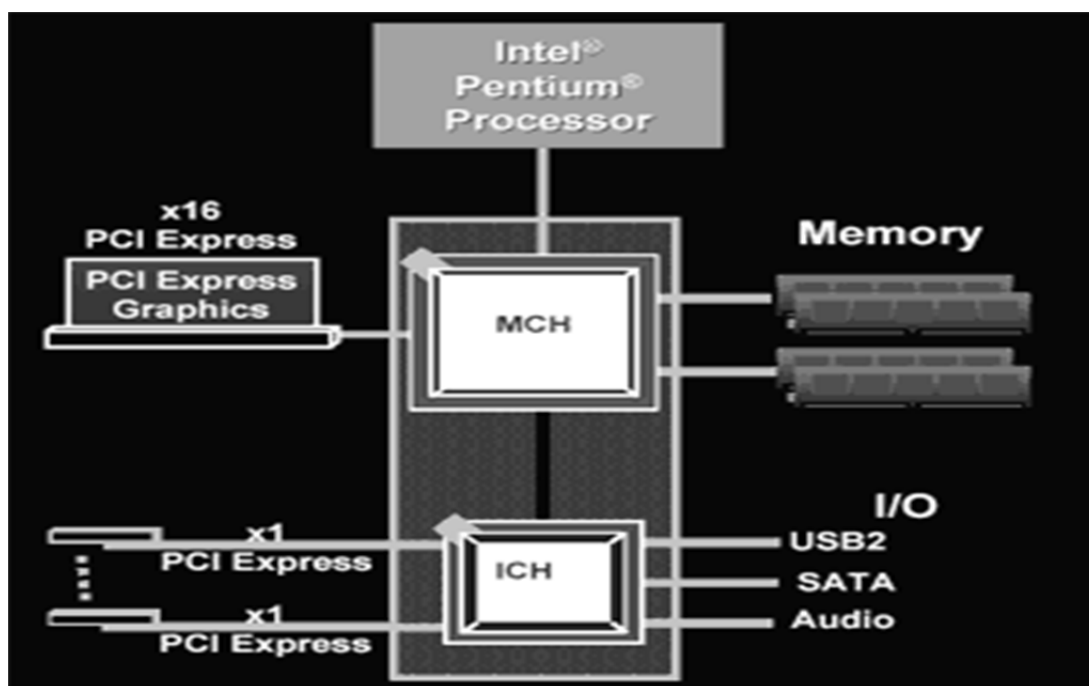


Рис. 2.7. Вариант перехода на *PCI-Express*

Однако в будущем логично ожидать появление некоего разветвителя *PCI-Express*. Тогда вполне оправданным станет и объединение северного и южного мостов. Приведем примеры возможных системных топологий (рис. 2.8 – 2.10).

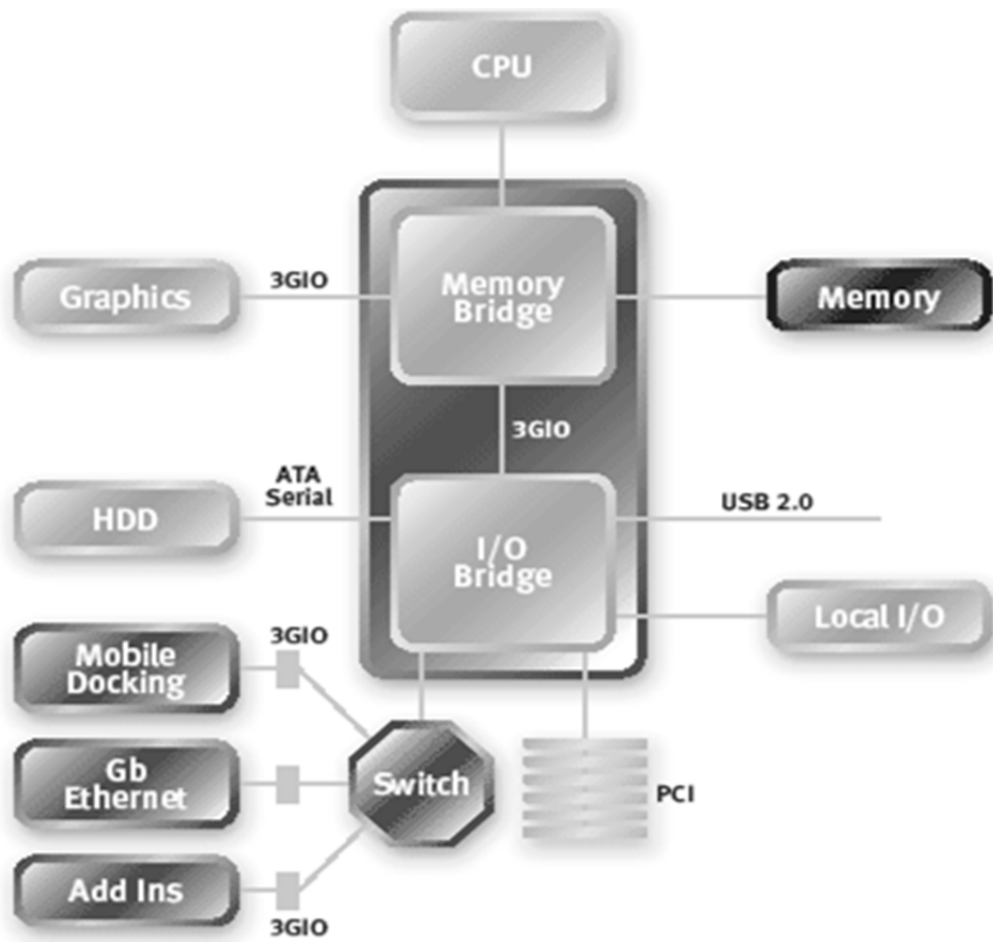


Рис. 2.8. Классическая архитектура ПК с двумя мостами

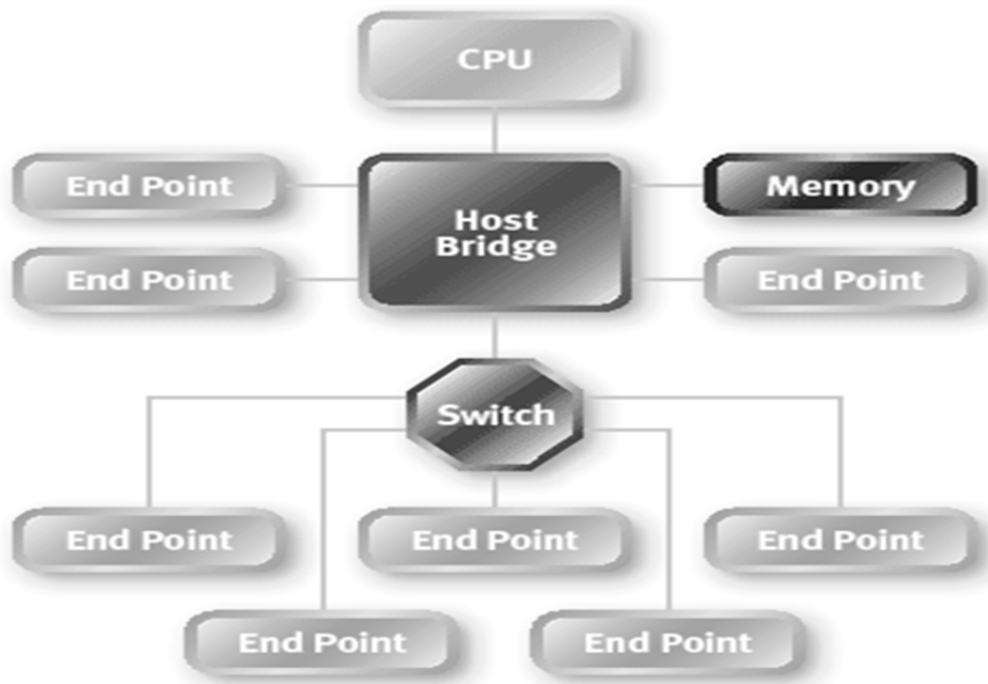


Рис. 2.9. Серверная архитектура с одним мостом

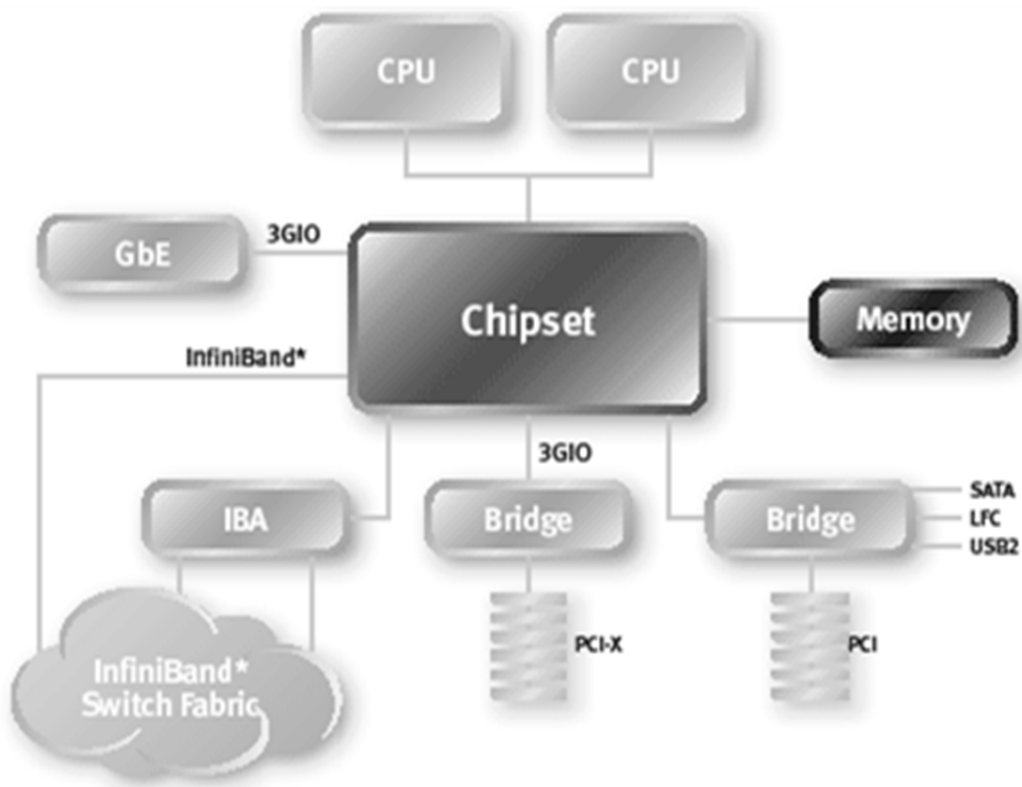


Рис. 2.10. Структура мощного сервера

Спецификации для интерфейса *PCI-E*

PCI-SIG утвердила окончательную версию спецификации протокола *PCI-E 2.0*. Утверждены также базовая спецификация и описание карт *PCI-Express* (бывшего *3GIO*).

Спецификация *PCI-Express* предусматривает скорость передачи данных 2,5 Гбит/с на канал в каждом направлении. Этот интерфейс предназначен для ПК. Он совместим с *PCI* по программному обеспечению, однако аппаратное обеспечение *PCI-Express* не будет поддерживать *PCI*.

Специалисты *Intel* считают, что благодаря новым решениям скорость передачи данных, обеспечиваемая этой архитектурой, может быть повышена до 10 Гбит/с на канал.

Развитием стандарта *PCI-Express* занимается организация *PCI Special Interest Group* (<http://www.pcisig.com/>).

В отличие от шины *PCI*, использовавшей для передачи данных общую шину, *PCI-Express* в общем случае является пакетной сетью с топологией типа звезда, устройства *PCI-Express* взаимодействуют между собой через среду, образованную коммутаторами, при этом

каждое устройство напрямую связано соединением типа точка-точка с коммутатором.

Шина *PCI Express* нацелена на использование только в качестве локальной шины. Так как программная модель *PCI-Express* во многом унаследована от *PCI*, то существующие системы и контроллеры могут быть доработаны для использования шины *PCI-Express* заменой только физического уровня без доработки программного обеспечения. Высокая пиковая производительность шины *PCI-Express* позволяет использовать её во многих подсистемах ЭВМ. В настоящее время она применяется прежде всего при подключении видеокарты вместо шины *AGP*.

Уровни протоколов *PCI-E*

- | | |
|---|---|
| 1. Уровень конфигурирования шины | прикладные программы ОС |
| 2. Прикладной уровень (совместимость <i>PCI</i> -драйверов) | драйверы <i>PCI</i> -устройств, данные |
| 3. Уровень транзакций | заголовок – данные |
| 4. Сетевой уровень (<i>link layer</i>) | номер – заголовок – данные – <i>CRC</i> |
| 5. Физический уровень | FR – номер – заголовок – данные – <i>CRC</i> – FR |

Формат пакетов шины *PCI-E*

- Символ начала кадра.
- Порядковый номер кадра.
- Заголовок.
- Данные.
- *CRC*-код.
- Символ конца кадра.

Для подключения устройства *PCI-Express* используется двунаправленное последовательное соединение типа точка-точка, называемое *lane*; это резко отличается от *PCI*, в которой все устройства подключаются к общей 32-разрядной параллельной однонаправленной шине. Соединение между двумя устройствами *PCI-Express* называется *link* и состоит из одного (называемого 1x) или нескольких (2x, 4x, 8x, 12x, 16x и 32x) двунаправленных последовательных соединений *lane*. Каждое устройство должно поддерживать соединение 1x.

На электрическом уровне каждое соединение использует низковольтную дифференциальную передачу сигнала (*LVDS*) (рис. 2.11). Приём и передача информации производятся каждым устройством *PCI-Express* по отдельной витой паре, таким образом, в простейшем случае устройство подключается к коммутатору *PCI-Express* всего лишь четырьмя проводниками.

Использование подобного подхода имеет следующие преимущества:

– карта *PCI-Express* помещается и корректно работает в любом слоте той же или большей пропускной способности (например, карта x1 будет работать в слотах x4 и x16);

– слот большего физического размера может использовать не все *lane* (например, к слоту 16x можно подвести линии передачи информации, соответствующие 1x или 8x, и всё это будет нормально функционировать; однако при этом необходимо подключить все линии «питание» и «земля», необходимые для слота 16x).

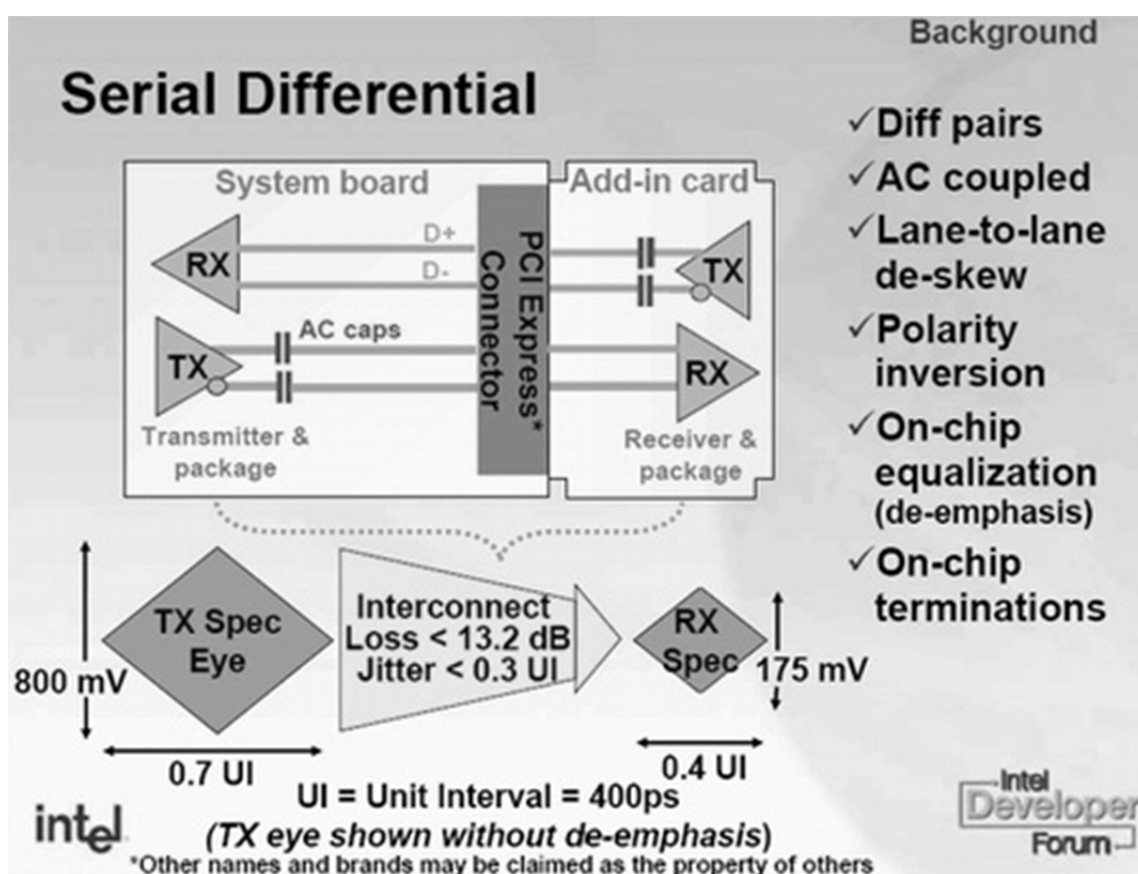


Рис. 2. 11. Физические уровни дифференциальной пары

В обоих случаях на шине *PCI-Express* будет использовать максимальное количество *lane*, доступных как для карты, так и для слота. Однако это не позволяет устройству работать в слоте, предназначенном для карт с меньшей пропускной способностью шины *PCI-Express* (например, карта x4 физически не поместится в слот x1, несмотря на то, что она могла бы работать в слоте 4x с использованием только одного *lane*).

PCI-Express пересылает всю управляющую информацию, включая прерывания, через те же линии, что используются для передачи данных. Последовательный протокол никогда не может быть заблокирован, таким образом задержки шины *PCI-Express* вполне сравнимы с таковыми для шины *PCI* (заметим, что шина *PCI* для передачи сигнала о запросе на прерывание использует отдельные физические линии *IRQ#A*, *IRQ#B*, *IRQ#C*, *IRQ#D*).

Во всех высокоскоростных последовательных протоколах (например, *GigabitEthernet*) информация о синхронизации должна быть встроена в передаваемый сигнал. На физическом уровне *PCI-Express* использует ставший общепринятым метод кодирования 8B/10B (8 бит данных заменяются на 10 бит, передаваемых по каналу, таким образом, 20 % передаваемого по каналу трафика являются избыточными), который позволяет поднять помехозащищенность.

Некоторые протоколы (например, *SONET/SDH*) используют другой метод помехозащищенного кодирования, который называется скремблинг (*scrambling*) для встраивания информации о синхронизации в поток данных. Спецификация *PCI-Express* также предусматривает алгоритм скремблинга, но скремблинг *PCI-Express* отличается от такового для *SONET*.

Пропускная способность шины *PCI-Express*

Пропускная способность соединения *lane* составляет 2,5 Гбит/с. Для расчета пропускной способности соединения *link* необходимо учесть то, что в каждом соединении передача дуплексная, а также учесть применение кодирования 8 бит/10 бит (8 бит в 10). Например, дуплексная пропускная способность соединения 1x составляет 0,5 ГБ/с. Пропускная способность, с учетом двунаправленной передачи для шин *PCI-Express* с разным количеством связей указана в табл. 2.3.

Таблица 2.3. Пропускная способность шин *PCI-Express*

Число линий <i>PCI-Express</i>	Пропускная способность в одном направлении	Суммарная пропускная способность
1	250 Мб/с	500 Мб/с
2	500 Мб/с	1 Гб/с
4	1 Гб/с	2 Гб/с
8	2 Гб/с	4 Гб/с
16	4 Гб/с	8 Гб/с
32	8 Гб/с	16 Гб/с

Материнская плата *EliteGroup 915P-Ac* имеет 2 разъема *PCI-Express 1x* и 1 разъем *PCI-Express 16x*, а также имеет 2 слота *PCI*. Как правило, в настольных системах (на чипсетах *Intel 915* и *925X*) присутствуют 1 слот *PCI-Express 16x*, предназначенный для установки видеокарты (заменяет разъем *AGP*) и до 4 слотов *PCI-Express 1x*. Серверные платы и платы, предназначенные для рабочих станций, кроме того, могут иметь слоты *PCI-Express 4x* и *8x*.

Архитектуру *PCI-Express* можно рассматривать послойно в сравнении с адресной моделью *PCI*. Конфигурация *PCI-Express* является стандартной для устройств, определенных *plug-and-play* спецификациями *PCI*: программный уровень генерирует запросы чтения/записи, уровень транзакций транспортирует эти запросы к периферийным устройствам с помощью разделенного пакетного протокола. Для поддержания высокой производительности шины соединительный (*link*) уровень добавляет пакетам очередность и *CRC*; базовый физический уровень состоит из двойного симплексного канала, осуществляющего функции приемной и передающей пары. Таким образом, исходная скорость 2,5 Гбит/с в каждом направлении позволяет говорить о создании дуплексного коммуникационного канала производительностью до 200 МБ/с, что в четыре раза превышает возможности классической шины *PCI*.

Рассматривая процессы, протекающие в шине на сигнальном уровне, нельзя не отметить уникальные плюсы *PCI-Express* – значительное снижение затухания в линиях передачи и повышенную чувствительность приемной части интерфейса. Из чего напрашивается вывод о менее критичных требованиях к импедансу входных цепей, а также возможность увеличения длины разводки проводников шины; в

нынешней версии стандарта *PCI-E* они лимитируются 12 дюймами для системных плат, 3,5 дюймами для контроллеров и 15 дюймами для межчиповых соединений. При этом не предъявляется никаких дополнительных требований к технологии разводки печатной платы: могут использоваться как обычные 4-слойные *PCB* толщиной 0,062 дюйма, так и варианты с шестью и более слоями.

Использование новых разъемов и других конструктивных возможностей, оговоренных спецификациями нового стандарта, позволяет говорить об увеличении энергопотребления конечных контроллеров до 75 Вт при токе до 5,5А (табл. 2.4).

Такие мощные контроллеры потребуют дополнительных мер по отводу тепла из корпуса. Разработчики уделили внимание проблеме масштабируемости производительности. Они отошли от принципа единого разъёма – в шине *PCI-Express* изначально предусмотрена возможность наращивания независимых линий передачи данных. Линия передачи *PCI-Express* x1 (одна линия) имеет весьма скромные показатели – эффективная пропускная способность до 200 МБ/с. Но за счет добавления стандартных секций в разъёме пропускная способность может быть легко увеличена до 6400 МБ/с – *PCI-Express* x32 (32 линии). Передаваемые данные поровну распределяются между линиями по принципу *n*-й байт на *n*-ю линию. При этом линии передачи данных в разъёме *PCI-Express* остаются независимыми, работают в асинхронном режиме. Ко всему достигается обратная совместимость: в многоканальные разъёмы *PCI-Express* можно вставлять платы расширения, рассчитанные на меньшее число каналов.

Таблица 2.4. Электрические параметры шины питания

Шина питания	Спецификация разъема x16
+3,3V Отклонение напряжения Ток	±9 % (max) 3,0 А (max)
+12V Отклонение напряжения Ток	±8 % (max) 5,5 А (max)
+3.3Vaux Отклонение напряжения Ток: активный режим спящий режим	±9 % (max) 375 мА (max) 19 А (max)

Разъёмы, платы и карты

Разъём *PCI-Express* делится ключом на две части. Первая часть (та, что ближе к задней стенке корпуса) одинакова для всех разъёмов и предназначена для питания карты. Сюда подводятся напряжения 3,3 и 12 В. Спецификацией предусматривается подводка мощности 60 Вт. По другую сторону от ключа расположены контакты секций линий передачи данных: от одной до тридцати двух. Соответственно количеству линий передачи меняется длина разъёма. Самые короткие разъёмы *PCI-Express* x1, длина *PCI-Express* x16 примерно равняется размеру обычного *PCI* слота.

Совершенно новые требования выдвигаются к механическим показателям *PCI-Express*. Для того чтобы периферийные платы не имели возможности вывалиться из слота при вибрации или транспортировке, разработаны повышенные требования к защелкам и крепежу разъёмов *PCI-Express*.

Несмотря на то что новый стандарт дает некую свободу конечным производителям при разработке крепежа, жестко оговоренными остаются следующие требования: энергопотребление – не более 75 Вт, вес – не более 350 г, высота – не более 115,15 мм.

Конечно, под такими монстрами прозрачно подразумеваются графические карты с интерфейсом *PCI-Express* 16x; во всех других случаях требования к крепежу и другим характеристикам контроллеров значительно скромнее.

Одними из первых устройств, которые стали массово выпускаться для шины *PCI-Express*, конечно же стали видеоадаптеры. Писали, что для видеоадаптера, да ещё с запасом на будущее, подойдет 164-контактный разъём *PCI-Express* x16. Эффективная пропускная способность *PCI-Express* x16 заметно выше таковой у *AGP* 8X:

На сегодняшний день видеокарта – не единственное устройство, нацелившееся на перспективную шину *PCI-Express*. В эту когорту следует добавить всех тех, кому уже не хватает возможностей морально устаревшей шины *PCI*. Это прежде всего сетевые контроллеры *Gigabit Ethernet*, *RAID*-контроллеры массивов жёстких дисков, карты для кодирования *HDTV*-потока в реальном времени. Шина *PCI-Express* нередко используется для установления связи между южным и северным мостами чипсета. Не приходится сомневаться, что коли-

чество устройств с поддержкой шины *PCI-Express* со временем будет только расти.

Особняком стоит реализация *PCI-Express* для мобильных устройств в виде стандарта *ExpressCard*. Эти модули используются в ноутбуках и миниатюрных настольных ПК, хотя уже известны случаи представления концепций серверных плат с разъемом *ExpressCard*. Основное преимущество применения таких модулей – подключение периферии практически без использования крепежного инструмента, а также инсталляции дополнительных драйверов. Технология *ExpressCard* заменит собой все устаревшие параллельные шины.

В настоящее время разработано два форм-фактора модулей *ExpressCard* – *ExpressCard/34* (ширина 34 мм) и *ExpressCard/54* (ширина 54). Оба модуля имеют высоту 5 мм как у стандарта *PC Card Type II*; длина модулей 75 мм, что на 10,6 мм меньше, чем у *PC Card*. При этом модули *ExpressCard/34* и *ExpressCard/54* обладают одинаковым интерфейсом. Каждый слот под модули *ExpressCard* может обслуживать шину *PCI-Express x1*.

Развитие интерфейса *PCI-Express*

Интерфейс *PCI-Express* является в настоящее время единственной современной и широко используемой шиной расширения (за исключением шины *LPC*, имеющей узкое применение). Поэтому интерфейс развивается с целью повышения пропускной способности. Сначала появилась спецификация протокола *PCI-E 2.0*. Пропускная способность этого варианта повышена вдвое, т.е. до 500 Мбит/с в одном направлении и до 1 Гбит/с в обоих направлениях на одну линию (line).

PCI-SIG утвердила окончательную версию спецификации протокола *PCI-E 3.0*. Пропускная способность этого варианта повышена вдвое по сравнению с *PCI-E 2.0*, т.е. до 1 и 2 Гбит/с соответственно.

Темы для закрепления знаний

1. Интерфейс *PCI* – общие характеристики, разновидности и параметры.
2. Интерфейс *PCI* – функции устройств *PCI* и особенности поведения графического адаптера.

3. Интерфейс *PCI* – особенности функционирования и способы повышения производительности.
4. Интерфейс *PCI* – особенности транзакций в различных режимах.
5. Интерфейсы *PCI-X* и *PCI-X 2.0*.
6. Интерфейс *AGP* – общие характеристики, разновидности и параметры.
7. Интерфейс *AGP* – функции устройств *AGP* и особенности поведения графического адаптера.
8. Интерфейс *AGP* – особенности функционирования и способы повышения производительности.
9. Интерфейс *AGP* – особенности транзакций в различных режимах.
10. Интерфейс *PCI-E* – общие характеристики, разновидности и параметры.
11. Интерфейс *PCI-E* – особенности функционирования и способы повышения производительности.
12. Интерфейс *PCI-E* – особенности транзакций в различных режимах.
13. Преимущества *PCI-Express*.
14. *PCI-Express* — ключевые отличия от *PCI* и схема включения в структуру ПК.
15. *PCI-Express* – описание протокола, его уровни и их функции.
16. *PCI-Express* – структура данных (информации).
17. *PCI-Express* – способы контроля и кодирования информации. Преимущества кодирования.
18. Электрическая схема соединения в *PCI-Express*.
19. Назначение *PCI-Express* и возможные подключаемые устройства.
20. *PCI-Express* для мобильных устройств.
21. *PCI-Express* – разъемы, платы и карты.

Глава 3. ТРАДИЦИОННЫЕ ИНТЕРФЕЙСЫ ПК

До появления ПК в вычислительной технике использовались различные специализированные интерфейсы, отличающиеся жесткой специализацией как по видам и типам ПУ, так и по системам ЭВМ. Эти интерфейсы не могли применяться в ПК в силу специализации, громоздкости и высокой стоимости. Поэтому в ПК первоначально использовали простой и дешевый интерфейс *RS-232C*, широко используемый в системах телеметрии в докомпьютерную эру. Этот интерфейс в силу простоты, дешевизны и больших расстояний (до сотен метров и более) до сих пор устанавливается в некоторые системные платы и используется для подключения к ПК различных бытовых приборов (счетчики электроэнергии, воды, газа и т.п.).

С появлением более быстродействующих ПУ, таких как принтеры и сканеры, был разработан параллельный интерфейс *IEEE-1284 (Centronics)*, который имел более высокую скорость по сравнению с *RS-232C*, но отличался высокой стоимостью, большим разъемом и толстым кабелем. Эти недостатки привели к тому, что этот интерфейс практически не применяется в современных компьютерах.

3.1. Интерфейс *RS-232C*

Общие характеристики

Интерфейс *RS-232C* предназначен для подключения аппаратуры, передающей или принимающей данные (ООД – оконечное оборудование данных, или АПД – аппаратура передачи данных; *DTE – Data Terminal Equipment*), к оконечной аппаратуре каналов данных (АКД; *DCE – Data Communication Equipment*). Обеспечивает синхронные и асинхронные режимы передачи данных.

Схемы соединения АПД и АКД

В роли АПД может выступать компьютер, принтер, плоттер и другое периферийное оборудование. В роли АКД обычно выступает модем. Конечной целью подключения является соединение двух устройств АПД.

Интерфейс позволяет исключить канал удаленной связи вместе с парой устройств АКД, соединив устройства непосредственно с помощью нуль-модемного кабеля (рис. 3.1).

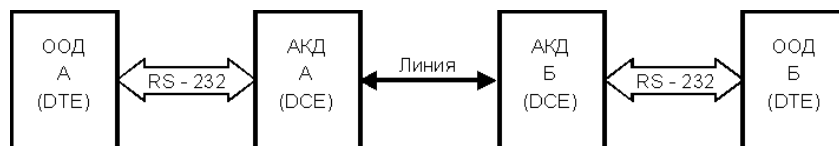


Рис. 3.1. Принцип соединения с помощью нуль-модемного кабеля

Асинхронный режим соединения представлен на рис. 3..

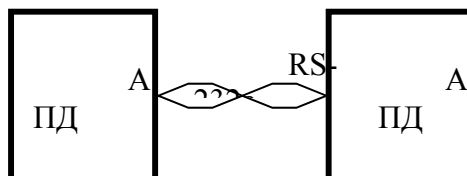


Рис. 3.2. Асинхронный режим соединения

3.1.1. Сигналы интерфейса RS-232C

Сигналы:

1. Передачи данных (*TXD*, *RXD*) – передача и прием данных.
2. Управляющие сигналы (*RTC* – запрос передачи, *CTC* – сброс передачи).
3. Сигналы синхронизации (*ТС*, *РС*) – используются в синхронном режиме. В асинхронном режиме используются 9 линий вместо 25.
4. Остальные сигналы:
 - PG* – защитная земля.
 - SG* – сигнальная земля.
 - TD* – последовательные данные – выход передатчика.
 - RD* – последовательные данные – вход приемника.
 - RTS* – выход запроса передачи данных.
 - CTS* – вход разрешения терминалу на передачу данных.
 - DSR* – вход сигнала готовности от АПД.
 - DTR* – выход сигнала готовности терминала к обмену данными.
 - DCD* – вход сигнала обнаружения несущей удаленного модема.
 - RI* – вход индикатора вызова.

3.1.2. Алгоритм управления

Временная диаграмма, отражающая алгоритм управления, отображена на рис. 3.3. Алгоритм управления содержит ряд действий:

1. Установка *DTR*. Компьютер указывает на желание использовать модем.
2. Установка *DSR*. Модем сигнализирует о своей готовности установить соединение.
3. Сигналом *RTS* компьютер запрашивает разрешение на передачу.
4. Установкой *STS* модем разрешает передачу по линии.
5. Снятием *STS* модем сигнализирует о невозможности дальнейшего приема. Компьютер должен приостановить передачу.
6. Сигналом *STS* модем разрешает продолжить передачу.

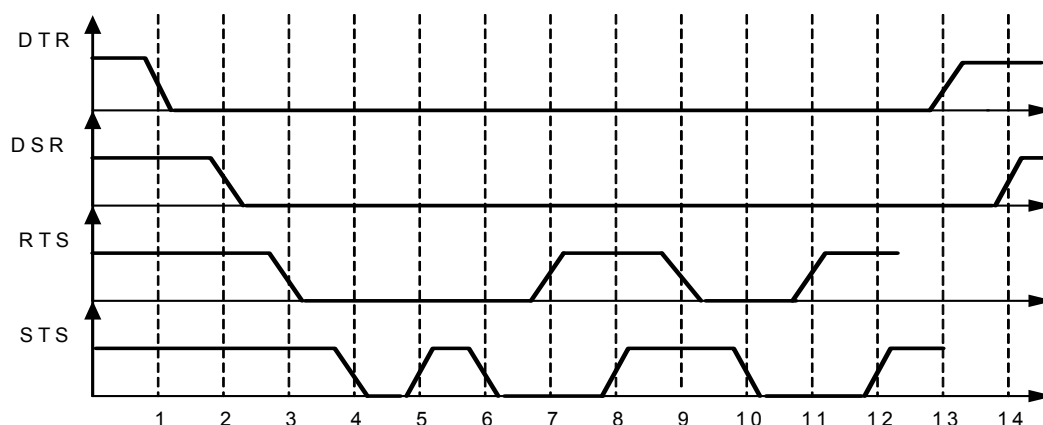


Рис. 3.3. Временная диаграмма управления

7. Снятие *RTS* может означать как заполнение буфера, так и отсутствие данных.
8. Сброс *STS* – модем подтверждает снятие *RTS*.
9. Компьютер повторно устанавливает *RTS* для возобновления передачи.
10. Модем подтверждает готовность к приему и передаче.
11. Компьютер указывает на завершение обмена.
12. Модем отвечает подтверждением (снятие *STS*).
13. Компьютер снимает *DTR*, что является сигналом разрыва соединения.
14. Модем «кладет трубку».

3.1.3. Типы разъемов

На аппаратуре АПД (в том числе на *COM*-портах) принято устанавливать вилки *DB-25P* или более компактный вариант – *DB-9P*.

9-штырьковые разъемы не имеют контактов для дополнительных сигналов, необходимых для синхронного режима (в большинстве 25-штырьковых разъемах эти контакты не используются).

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

3.1.4. Родственные интерфейсы и преобразователи уровня

Интерфейсы EIA-RS-422 (ITU-T V.11, X.27) и EIA-RS-485 (ISO 8482) используют симметричную передачу сигнала и допускают как двухточечную, так и шинную топологию соединений (рис.3.4).

Выходное сопротивление передатчиков 100 Ом.

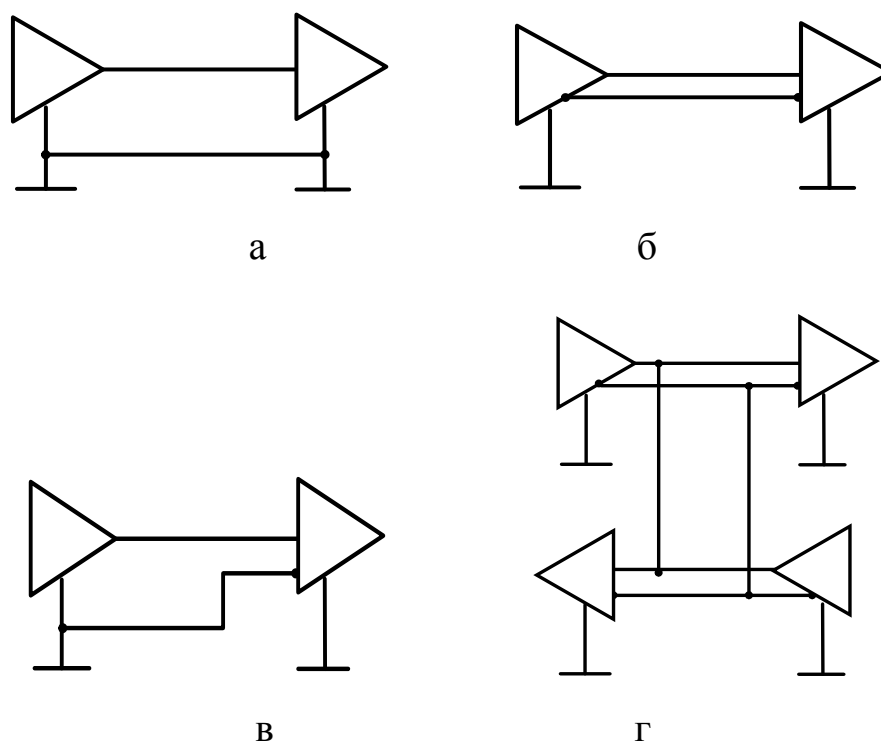


Рис. 3.4. Родственные интерфейсы и преобразователи уровня:
а – RS-232C; б – RS-422A; в – RS-423A; г – RS-485

Интерфейсы электрически совместимы между собой, хотя и имеют некоторые различия в ограничениях.

Принципиальное отличие передатчиков *RS-485* – возможность переключения в третье состояние. Передатчики *RS-422/485* совместимы с приемниками *RS-423*.

Максимальная скорость передачи на коротких расстояниях (до 10 м) ограничивается быстродействием передатчиков (достижима частота 25 МГц).

На средних расстояниях ограничение определяется емкостью кабеля (1200 бит/с – 25 нФ, 9600 бит/с – 30 нФ, 115 кбит/с – 250 пФ).

Максимальная дальность (1200 м) ограничена сопротивлением петли постоянному току.

Интерфейс *RS-485* может быть в двух версиях: двухпроводной и четырехпроводной. Четырехпроводная версия выделяет задающий узел (*master*), передатчик которого работает на приемники всех остальных. Передатчик задающего узла всегда активен. Передатчики *slave* узлов должны иметь трехстабильные выходы, они объединяются на общей шине с приемником ведущего узла. В двухпроводной версии все узлы равноправны.

При многоточечном соединении необходимо организовать метод доступа к среде.

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

Токовая петля для представления сигнала использует не напряжение, а ток в двухпроводной линии, соединяющей приемник и передатчик.

Логической единице (состоянию «включено») соответствует протекание тока 20 мА. Логическому нулю – отсутствие тока. Такое представление сигналов асинхронной посылки позволяет обнаружить обрыв линии.

Интерфейс *RS-232C* может быть преобразован в «токовую петлю». Интерфейс *MIDI* для аудиоаппаратуры позволяет объединить до 16 устройств; имеет 3 провода.

Разъем типа *DIN-5*:

4-й контакт – *IN*;

5-й контакт – *OUT*;

2-й контакт – экран (соединяется с общим проводом только на стороне передатчика).

Скорости передачи и соответствующие им расстояния следующие:

RS-232C: $L = 15$ м, $V = 20$ Кбит/с

RS-422A (витая пара):

$L = 12$ м, $V = 10$ Мбит/с

$L = 120$ м, $V = 1$ Мбит/с

$L = 1200$ м, $V = 100$ Кбит/с

RS-423A:

$L = 9$ м, $V = 100$ кбит/с

$L = 91$ м, $V = 10$ кбит/с

$L = 1200$ м, $V = 1$ кбит/с

RS-485(дуплексная связь):

$L = 12$ м, $V = 100$ Мбит/с

$L = 120$ м, $V = 1$ Мбит/с

$L = 1200$ м, $V = 10$ кбит/с

Схема соединения земель для интерфейсов *RS-422A* и *RS-485* приведена на рис. 3.5.

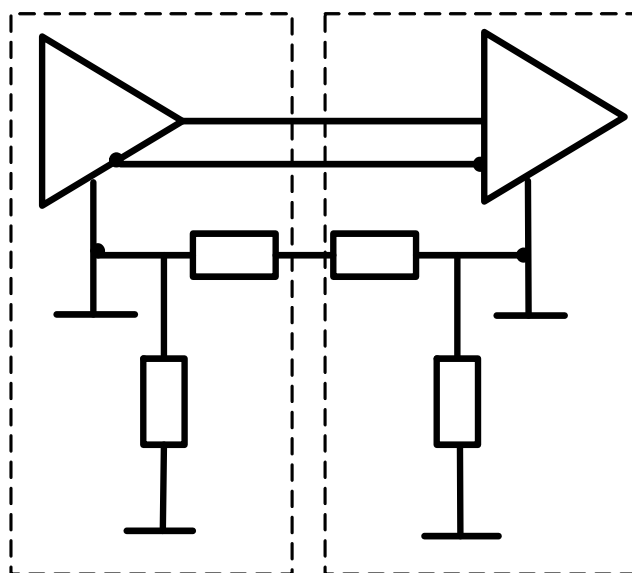


Рис. 3.5. Схема соединения земель для интерфейсов *RS-422A* и *RS-485*

3.1.5. Электрический интерфейс *RS-232C*

Имеется несимметричный передатчик/приемник.

Приемник:

логический "0": $-12 \dots -3$ В;

логическая "1": $+3 \dots +12$ В;

$-3 \dots +3$ В – зона нечувствительности.

Передатчик:

логический "0": -12 ... -5 В;

логическая "1": +5 ... +12 В;

-5 ... +5 В – зона нечувствительности.

Разность потенциалов между землями должна быть меньше 2 В. Так же должно быть защитное заземление при питании от сети переменного тока.

3.1.6. Асинхронный режим передачи

Асинхронный режим передачи является байт-ориентированным (символьно-ориентированным): минимальная пересылаемая единица информации – один байт (один символ). Передача каждого байта начинается со старт-бита, сигнализирующего приемнику о начале посылки, за которым следуют биты данных и, возможно, бит четности (*Parity*) (рис. 3.6). Завершает посылку стоп-бит, гарантирующий паузу между посылками. Старт-бит следующего байта посылается в любой момент после стоп-бита, т. е. между передачами возможны паузы произвольной длительности.

Пример: А=1000001

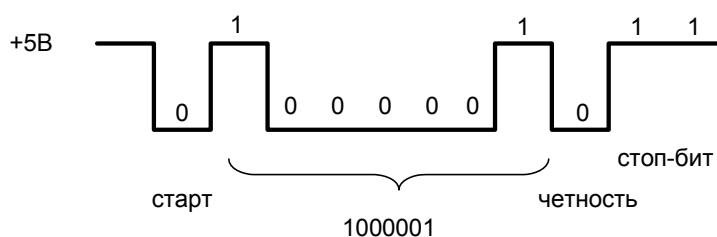


Рис. 3.6. Структура передаваемой информации в асинхронном режиме

Кодирование битов информации передается в инверсном коде (рис. 3.7).

Старт-бит, имеющий всегда строго определенное значение (логический 0), обеспечивает простой механизм синхронизации приемника по сигналу от передатчика. Подразумевается, что приемник и передатчик работают на одной скорости обмена. Внутренний генератор синхронизации приемника использует счетчик-делитель опорной частоты, обнуляемый в момент приема начала старт-бита. Этот счетчик генерирует внутренние стробы, по которым приемник фиксирует последующие принимаемые биты.



Рис. 3.7. Формат посылки байта

Формат асинхронной посылки позволяет выявлять возможные ошибки передачи.

Если принят перепад, сигнализирующий о начале посылки, а по стробу старт-бита зафиксирован уровень логической единицы, старт-бит считается ложным и приемник снова переходит в состояние ожидания. Об этой ошибке приемник может не сообщать.

Если во время, отведенное под стоп-бит, обнаружен уровень логического нуля, фиксируется ошибка стоп-бита.

Если применяется контроль четности, то после посылки бит данных передается контрольный бит. Этот бит дополняет количество единичных бит данных до четного или нечетного в зависимости от принятого соглашения. Прием байта с неверным значением контрольного бита приводит к фиксации ошибки.

Контроль формата позволяет обнаруживать обрыв линии: как правило, при обрыве приемник «видит» логический ноль, который сначала трактуется как старт-бит и нулевые биты данных, но потом срабатывает контроль стоп-бита.

3.1.7. Аппаратный протокол управления потоком *RTS/CTS*

Использует сигнал *CTS*, который позволяет остановить передачу данных, если приемник не готов к их приему. Передатчик «выпускает» очередной байт только при включенной линии *CTS*. Байт, который уже начал передаваться, задержать сигналом *CTS* невозможно (это гарантирует целостность посылки) (рис. 3.8). Аппаратный протокол обеспечивает самую быструю реакцию передатчика на состояние приемника. Микросхемы асинхронных приемопередатчиков имеют не

менее двух регистров в приемной части – сдвигающий, для приема очередной посылки, и хранящий, из которого считывается принятый байт. Это позволяет реализовать обмен по аппаратному протоколу без потери данных. Аппаратный протокол удобно использовать при подключении принтеров и плоттеров, если они его поддерживают. При непосредственном (без модемов) соединении двух компьютеров аппаратный протокол требует перекрестного соединения линий *RTS* – *CTS*.

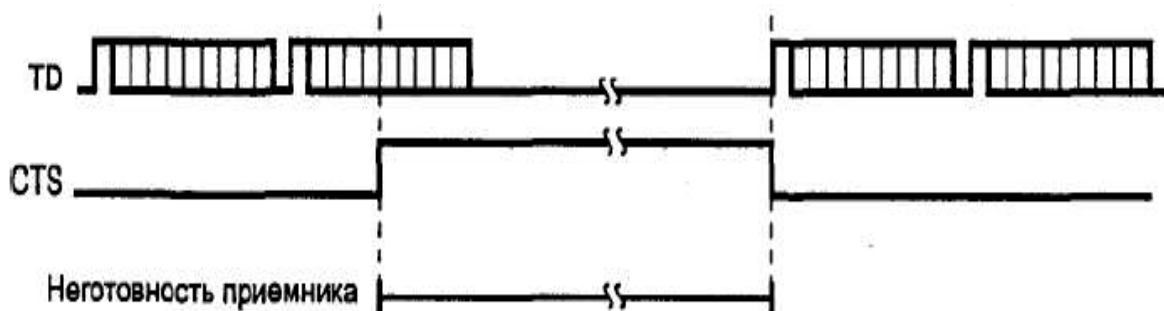


Рис. 3.8. Диаграмма аппаратного управления потоком

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

3.1.8. Программный протокол управления потоком *XON/XOFF*

Предполагает наличие двунаправленного канала передачи данных. Работает протокол следующим образом: если устройство, принимающее данные, обнаруживает причины, по которым оно не может их дальше принимать, оно по обратному последовательному каналу посылает байт-символ *XOFF(13h)*. Противоположное устройство, приняв этот символ, приостанавливает передачу. Когда принимающее устройство снова становится готовым к приему данных, оно посылает символ *XON(11h)*, приняв который противоположное устройство возобновляет передачу.

Время реакции передатчика на изменение состояния приемника по сравнению с аппаратным протоколом увеличивается по крайней мере на время передачи символа (*XON* или *XOFF*) плюс время реакции программы передатчика на прием символа (рис. 3.9). Из этого

следует, что данные без потерь могут приниматься только приемником, имеющим дополнительный буфер принимаемых данных и сигнализирующим о неготовности, заблаговременно (имея в буфере свободное место).

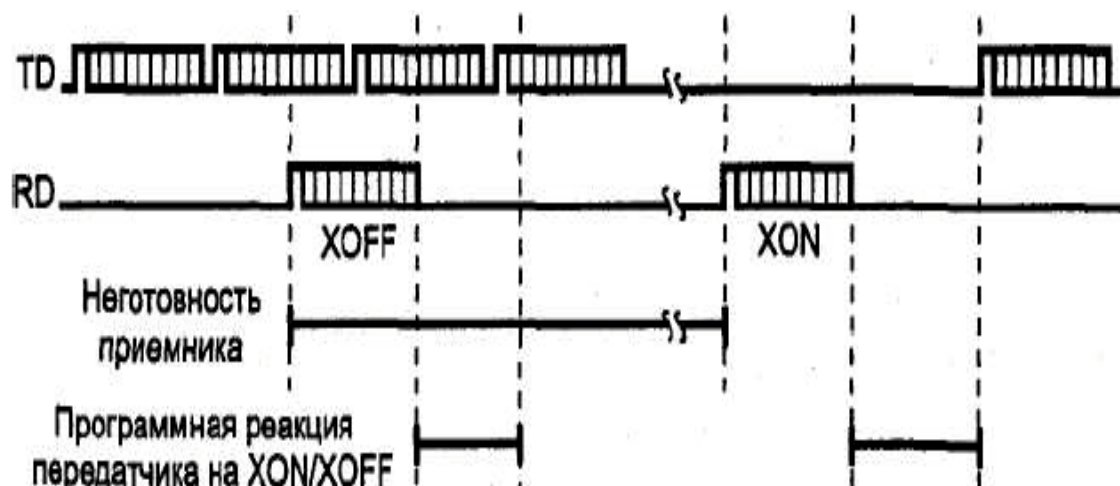


Рис. 3.9. Диаграмма программного управления потоком

Преимущество программного протокола заключается в отсутствии необходимости передачи управляющих сигналов интерфейса – минимальный кабель для двустороннего обмена может иметь только 3 провода.

Недостатком помимо обязательного наличия буфера и большего времени реакции, снижающего общую производительность канала из-за ожидания сигнала *XON*, является сложность реализации полнодуплексного режима обмена. В этом случае из потока принимаемых данных должны выделяться (и обрабатываться) символы управления потоком, что ограничивает набор передаваемых символов.

3.1.9. Микросхемы асинхронных приемопередатчиков

Микросхема асинхронного приемопередатчика (*UART*) *Intel*8250 (рис. 3.10):

8250 – имела ошибки;

8250А – потеряна совместимость с другими микросхемами;

8250В – исправлены ошибки и возвращена совместимость.

Эти микросхемы имеют небольшое быстродействие. В связи с этим выпущена новая серия: *NC* 16450, *NC* 16550, *NC* 16550А, кото-

рая используется до сих пор. В них повышено быстродействие, но они не имеют полной совместимости.

Микросхема *NC 16550A* – это набор из 12 регистров, доступ к которым определяется адресом смещения относительно базового адреса порта.

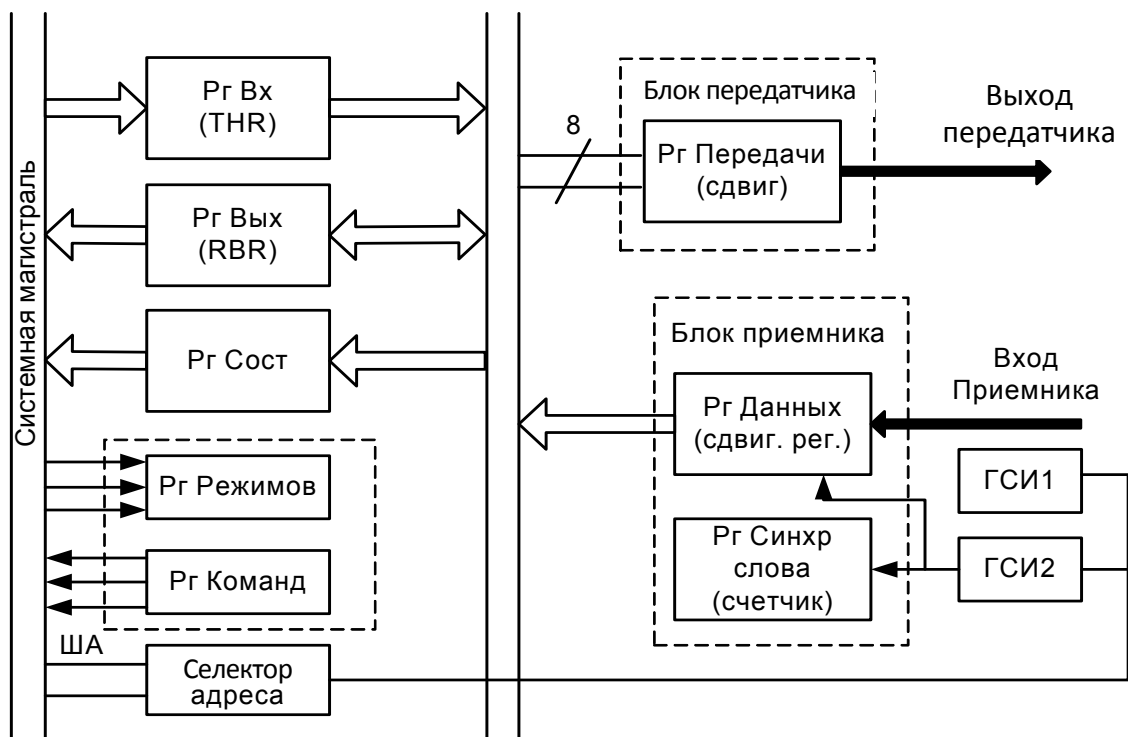


Рис. 3.10. Упрощенная схема *UART*

Регистры:

1. *THR* – промежуточный регистр данных передатчика – хранит данные, которые будут переданы.
2. *RBR* – буферный регистр передаваемых данных для чтения.
3. *DLL* – регистр младшего байта делителя частоты.
4. *DLM* – регистр старшего байта делителя частоты.
Делитель частоты определяется по формуле $D = 115200/V$, где V – скорость передачи. $F_{\text{вх синхр}} = 1,8432$ МГц.
5. *IER* – регистр разрешения прерываний.
6. *MR* – регистр идентификации прерываний и признак режима *FIFO* (для чтения). Четырехуровневая система приоритетов.
7. *FCR* – регистр управления *FIFO* (для записи).
8. *LCR* – регистр управления линией.
9. *MCR* – регистр управления модемом.

10. *LSR* – регистр состояния линии.
11. *MSR* – регистр состояния модема.
12. *SCR* – рабочий регистр.

3.1.10. Аппаратная реализация

Аппаратная реализация интерфейса *RS-232* включает в себя последовательный адаптер и собственно механический интерфейс (разъемное соединение). Структурная схема типичного варианта адаптера последовательного порта *RS-232* представлена на рис. 3.11.

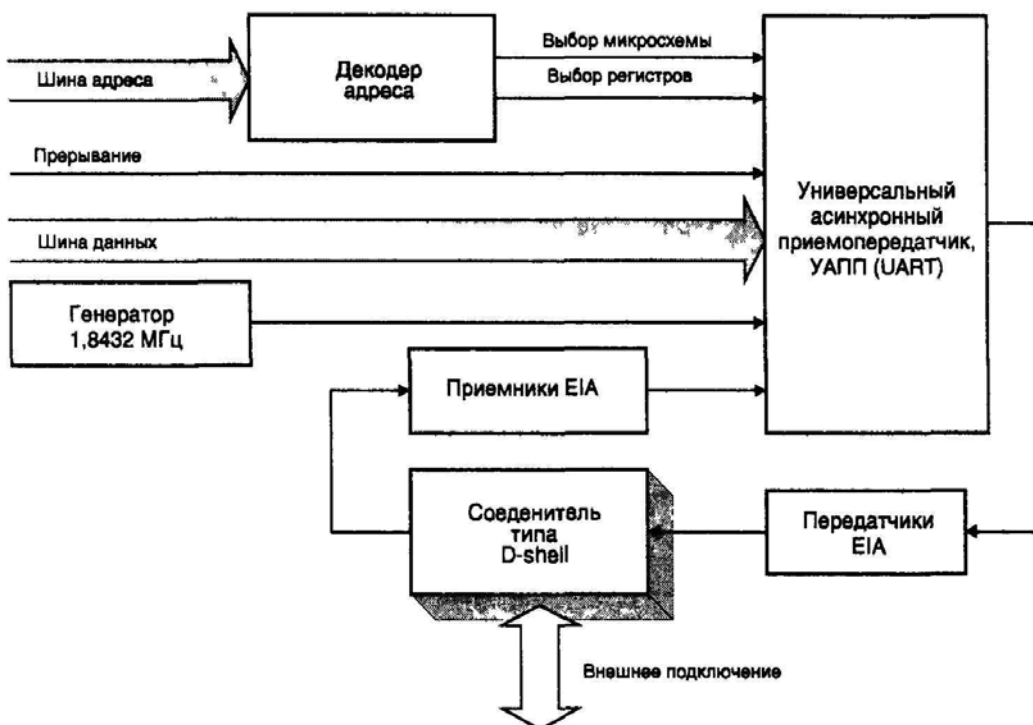


Рис. 3.11. Структурная схема адаптера *RS-232C*

3.2. COM-порт

Универсальный внешний последовательный интерфейс *COM-port* (*CommunicationsPort* – коммуникационный порт) присутствует в РС начиная с первых моделей. Этот порт обеспечивает асинхронный обмен по стандарту *RS-232C*. *COM*-порты реализуются на микросхемах универсальных асинхронных приемопередатчиков (*UART*), совместимых с семейством *i8250/16450/16550*. Они занимают в пространстве ввода-вывода по 8 смежных 8-битных регистров и могут

располагаться по стандартным базовым адресам $3F8h$ (*COM1*), $2F8h$ (*COM2*), $3E8h$ (*COM3*), $2E8h$ (*COM4*). Порты могут вырабатывать аппаратные прерывания *IRQ4* (обычно используются для *COM1* и *COM3*) и *IRQ3* (для *COM2* и *COM4*). С внешней стороны порты имеют линии последовательных данных передачи и приема, а также набор сигналов управления и состояния, соответствующий стандарту *RS-232C*. *COM*-порты имеют внешние разъемы (*male – nana*) *DB25P* или *DB9P*, выведенные на заднюю панель компьютера. Характерной особенностью интерфейса является применение не *TTL* сигналов – все внешние сигналы порта двуполярные. Гальваническая развязка отсутствует – схемная земля подключаемого устройства соединяется со схемной землей компьютера. Скорость передачи данных может достигать 115 200 бит/с.

Системная поддержка *COM*-портов заключается в том, что этот порт поддерживается прерыванием *BIOS-14h*. Применяется взаимодействие с портом на уровне регистров.

Функции, которые поддерживает прерывание *14h*:

1. Инициализация, при которой устанавливается скорость обмена, формат посылок, запрет/разрешение источников прерывания.
2. Вывод символов активизируется сигналами *DTR* и *RTS*. После освобождения регистра *THR* в него помещается выводимый символ.
3. Ввод символа: активизируется только сигнал *DTR*, а *RTS* – пассивен и ожидает готовности приема данных.
4. Опрос состояния модема и линии – читаются регистры *MSR* и *LSR*.

В начале процесса тестирования *BIOS* проверяет наличие *COM*-портов и помещает их базовые адреса в ячейки *BIOSDATAAREA*.

COM1 – 0:0400, *COM2* – 0:0402, *COM3* – 0:0404, *COM4* – 0:0406. (0 – адрес сегмента, 0400 – адрес внутри сегмента).

В ячейки 0:047C, 0:047D, 0:047E, 0:047F заносятся константы *TimeOut* для портов (1 байт)

Порт инициализируется для скорости 2400 бит/с. 7 бит – данные с контроллера о четности, 1 строб-бит – управляющие сигналы.

3.2.1. Управление портом

Управление портом разделяется на два этапа:

1. Предварительное конфигурирование.
2. Изменение текущего состояния в процессе работы (переключение режимов работы прикладного или системного ПО).

Конфигурируются следующие параметры:
– базовый адрес (*3F8h*, *2F8h*, *3E8h*, *2E8h*);
– используемые линии запросов на прерывания (либо та, либо другая):

COM1, *COM3* - *IRQ4*, *IRQ11*

COM2, *COM4* - *IRQ3*, *IRQ10*;

– определяется канал *DMA* (разрешение использования и номер канала). Использование *DMA* неэффективно, так как скорость порта мала.

3.2.2. Использование COM-портов

Название порта указывает на его основное назначение – подключение коммуникационного оборудования для связи с другими компьютерами, сетями и периферийными устройствами.

COM-порт используется:

- для подключения манипуляторов (мышь, клавиатура);
- внешних модемов;
- для связи двух компьютеров;
- электронных ключей;
- персональных компьютеров к *MainFrame*;
- как двунаправленный интерфейс (3 программно-управляемых выходных линии *TD*, *DTR*, *RTS* и 4 программно-читаемых выходных линии *CTS*, *DSR*, *DCD*, *RI*).

3.2.3. COM-порт и PnP

Современные устройства могут поддерживать этот режим. Инициализация подключаемого устройства реализуется с помощью несложного протокола. Основная задача ОС заключается в идентификации подключенного устройства, для чего разработан несложный протокол, реализуемый на любых COM-портах чисто программным способом. Этот протокол иллюстрирует рис. 3.12.

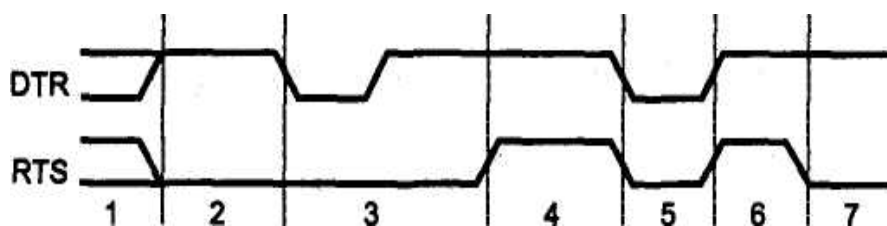


Рис. 3.12. Запрос идентификатора устройства PnP

1. Порт инициализируется с состоянием линий $DTR = ON$, $RTS = OFF$, $TXD = Mark$ – состояние покоя (*Idle*).

2. Некоторое время (0,2 с) ожидается появление сигнала DSR , которое указало бы на наличие устройства, подключенного к порту. В простейшем случае устройство имеет на разъеме переключку $DTR-DSR$, обеспечивающую указанный ответ. Если устройство обнаружено, выполняются манипуляции управляющими сигналами DTR и RTS для получения информации от устройства. Если ответ не получен, ОС, поддерживающая динамическое реконfigurирование, периодически опрашивает состояние порта для обнаружения новых устройств.

3. Порт программируется на режим 1200 бит/с, 7 бит данных, без паритета, 1 стоп-бит, и на 0,2 с снимается сигнал DTR . После этого устанавливается $DTR = 1$, а еще через 0,2 с устанавливается и $RTS=1$.

4. В течение 0,2 с ожидается приход первого символа от устройства. По приходу символа начинается прием идентификатора (см. ниже). Если за это время символ не пришел, выполняется вторая попытка опроса (см. п. 5), несколько отличающаяся от первой.

5. На 0,2 с снимаются оба сигнала ($DTR=0$ и $RTS = 0$), после чего они оба устанавливаются ($DTR = 1$ и $RTS = 1$).

6. В течение 0,2 с ожидается приход первого символа от устройства, по приходу символа начинается прием идентификатора (см. ниже). Если за это время символ не пришел, то в зависимости от состояния сигнала DSR переходят к проверке отключения *VerifyDisconnect* (при $DSR = 0$) или в дежурное состояние *ConnectIdle* (при $DSR = 1$).

7. В дежурном состоянии *ConnectIdle* устанавливается $DTR = 1$, $RTS = 0$, порт программируется на режим 300 бит/с, 7 бит данных, без паритета, 1 стоп-бит. Если в этом состоянии обнаружится $DSR = 0$, ОС следует уведомить об отключении устройства.

Посимвольный прием идентификатора устройства имеет ограничения по тайм-ауту в 0,2 с на символ, а также общее ограничение в 2,2 с, позволяющее принять строку длиной до 256 символов. Строка идентификатора *PnP* должна иметь маркеры начала ($28h$ или $08h$) и конца ($29h$ или $09h$), между которыми располагается тело идентификатора в стандартизованном формате. Перед маркером начала может находиться до 16 символов, не относящихся к идентификатору *PnP*. Если за первые 0,2 с ожидания символа (шаг 4 или 6) маркер начала не пришел или же сработал тайм-аут, а маркер конца не получен или же

какой-либо символ принят с ошибкой, происходит переход в состояние *ConnectIdle*. Если получена корректная строка идентификатора, она передается ОС.

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

3.3. Интерфейс *Centronics IEEE-1284* и *LPT*-порт

Порт параллельного интерфейса изначально был предназначен для принтера. Но в результате развития приобрел многовариантность и многорежимность.

Базовыми адресами порта могут быть три значения:

- $3BCh$
- $378h$
- $278h$

Запросы прерываний:

- $IRQ7$
- $IRQ5$

Шины порта:

- ШД – 8 бит
- ШУ – 4 бита
- ШСост – 5 бит

Прерывание *BIOS:INT17h*.

Порт имеет три восьмибитных регистра данных:

- $RGD (BASE)$
- $SRG (BASE+1)$
- $CRG (BASE+2)$

3.3.1. Схемы соединения

Определяют два уровня совместимости:

1. Для медленных устройств (обеспечивают смену направлений данных) (рис. 3.13, *a*).

2. Для быстрых широкополосных устройств (с длинными кабелями) (рис. 3.13, б).

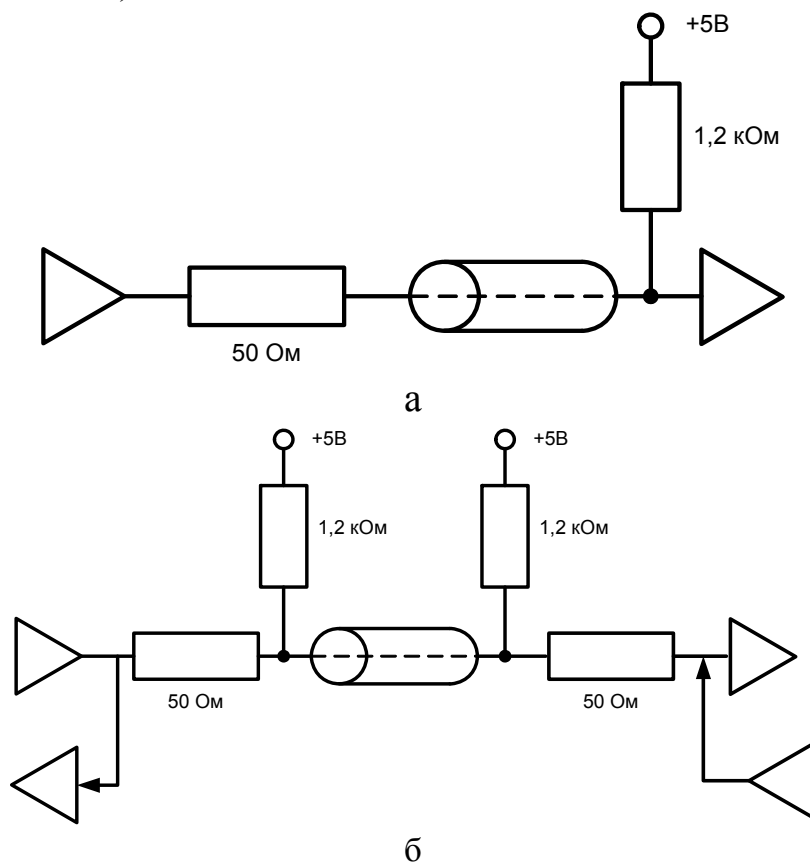


Рис. 3.13. Схемы соединения:
а – односторонняя передача; б – двунаправленная передача

3.3.2. Электрические параметры

Параметры передатчика:

1. Уровни сигналов

– без нагрузки U_c б/н = 0,5 – 5,5 В;

– под нагрузкой ($I_n = 14$ мА);

U_c п/н $\geq 2,4$ В – "1" (уровень ОН);

U_c п/н $\leq 0,4$ В – "0" (уровень ОЛ).

2. Выходной импеданс $R_{\text{вых}} = 50 \pm 5$ Ом.

3. Скорость нарастания импульса $\tau = 0,05 - 0,4$ В/нс (10 – 20 % от $\tau_{\text{имп}}$).

Параметры приемника:

1. Допустимое пиковое значение сигнала $U_{\text{пик}} = -2 \dots +7$ В.

2. Пороги срабатывания $U_{п"1"} = 2 ВU_{п"0"} = 0,8 В$.
3. Входной ток $I_{вх} \leq 20 \text{ мкА}$ (при больших токах не гарантируется нормальная работа интерфейса).
4. Входная емкость $C_{вх} \leq 50 \text{ пФ}$.

3.3.3. Конструктивы (разъемы)

Представлены на рис. 3.14.

Виды разъемов:

- штырьковый, 25-контактный – *IEEE-1284A*;
- 36-контактный – *IEEE-1284-B*;
- разъем высокой плотности *IEEE-1284-C*;
- такой же, как *IEEE-1284-B*, только меньших размеров.

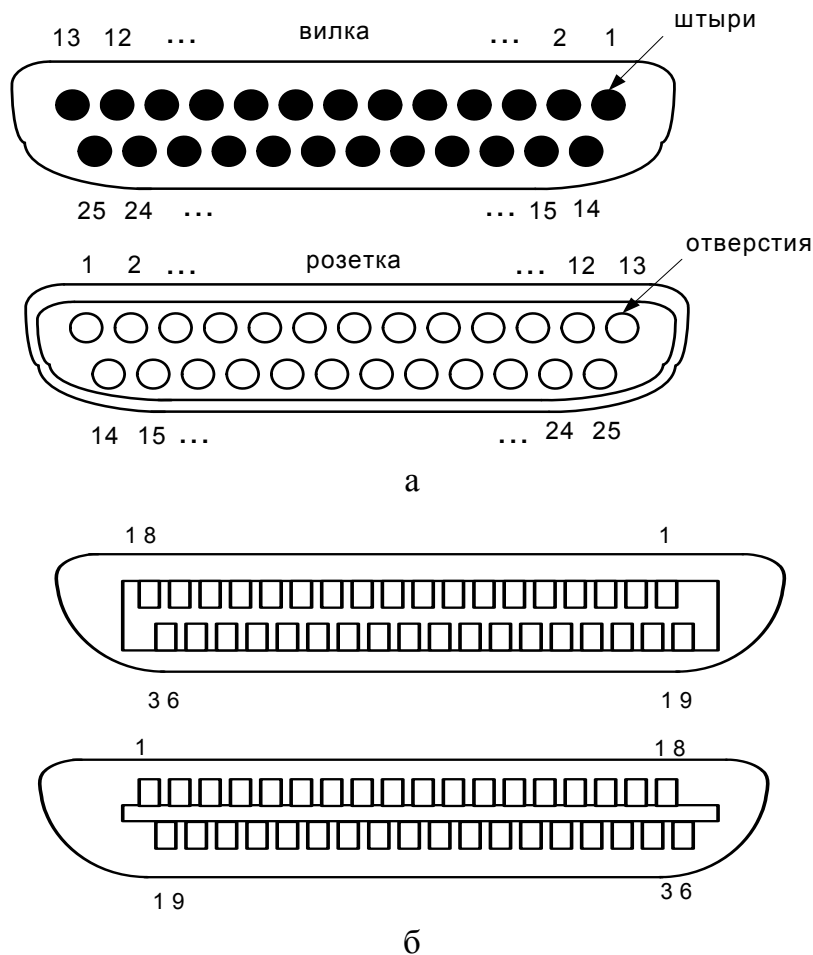


Рис. 3.14. Разъемы LPT-порта:
а – *IEEE-1284A*; б – *IEEE-1284-B*

3.3.4. Режимы работы IEEE-1284

1. Режим совместимости – однонаправленный вывод по протоколу *Centronics* (соответствует стандарту порта *SPP*).

2. Полубайтный режим – ввод байта в два цикла по 4 бита (рис. 3.15); использует для приема линии состояния. Используется в любых адаптерах порта.

3. Байтный режим – ввод байта целиком (рис. 3.16). Используется в портах, допускающих чтение выходных данных.

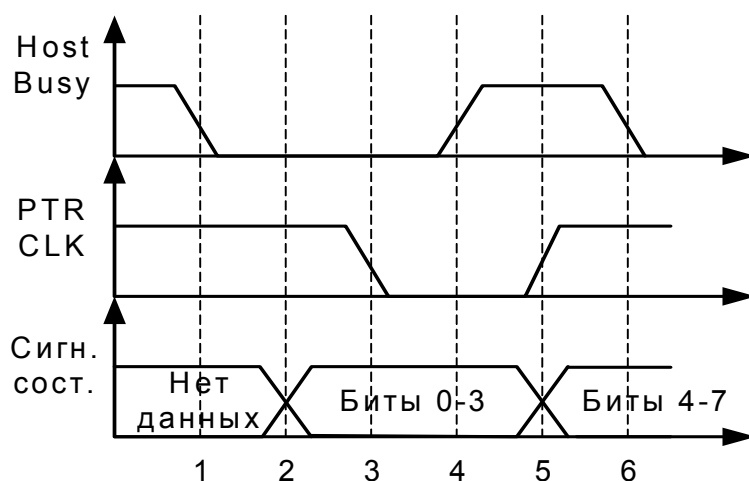


Рис. 3.15. Временная диаграмма полубайтного режима

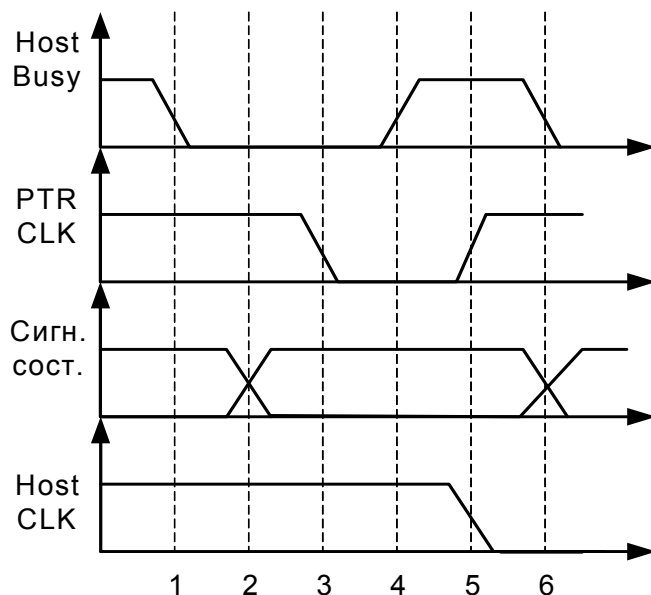


Рис. 3.16. Временная диаграмма двунаправленного байтного режима

4. Режим *EPP* – используется в портах, имеющих такое же название. Двухнаправленный вывод данных. Управление шиной генерируется аппаратно во время цикла обращения к порту. Имеет повышенную скорость обмена. Используется для обмена с внешними накопителями и адаптерами локальных сетей.

5. Режим *ECP* – двухнаправленный обмен данными с возможностью сжатия данных по методу *RLE*. Самый быстродействующий. Использует *FIFO*-буферы и *DMA*. Режимы задаются в *BIOS Setup*.

Полубайтный режим:

1. Сигнал *HostBusy* установкой низкого уровня определяет готовность к приему данных.

2. ПУ на линиях состояния помещает тетраду данных – биты (0÷3).

3. ПУ сигнализирует о готовности данных сигналом *PtrClk*.

4. *Host* устанавливает высокий уровень, указывая на занятость приемом и обработкой 4-битной тетрады.

5. ПУ снимает сигнал *PtrClk* готовности.

6. Повтор шагов 1 – 5 для второй тетрады – биты 4 – 7:

«+» – возможность работы на любых портах;

«-» – малая скорость (50 кб/с).

Байтный режим:

1. Установка низкого уровня.

2. Установка данных.

3. Установка *PTRCLK*.

4. Изменение *HostBusy*.

5. Установка режима *HostCLK*.

Скорость обмена = 150 кБ/с, но работает только на двухнаправленных портах.

Режим EPP:

Циклы обмена:

– 1, 2 – запись/чтение данных (рис. 3.17);

– 3, 4 – запись/чтение адреса.

Циклы обмена данными и адресами различаются стробирующими сигналами. Имеется расширенный набор из 8 регистров. Этот порт реализует сигналы аппаратно.

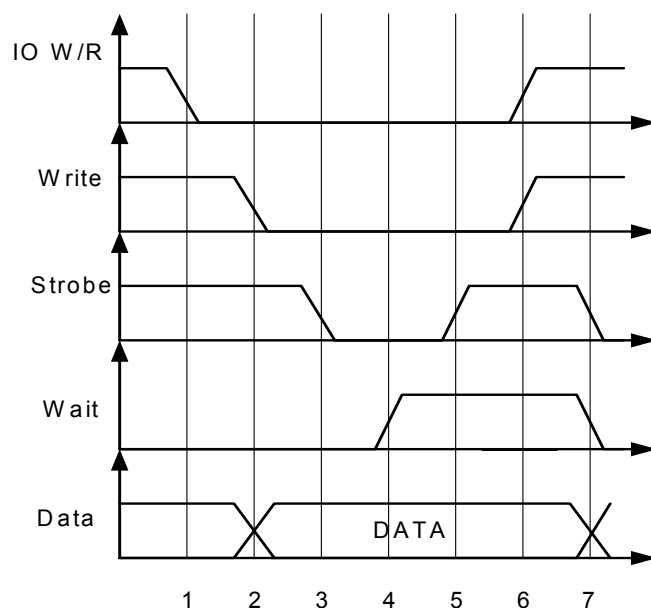


Рис. 3.17. Циклы обмена записи/чтения данных

Цикл записи данных:

1. Программа выполняет цикл ввода в порт 4.
2. Адаптер устанавливает сигнал *Write* и данные.
3. При низком уровне сигнала устанавливается строб данных.
4. Порт ждет подтверждения от ПУ (перевода *Wait* в высокий уровень).
5. Снимается строб данных.
6. Завершается процессорный цикл данных.
7. Возможность следующего цикла – низкий уровень *Wait*.

При записи адреса данные устанавливаются на 4-м такте. Стробирующий сигнал проходит по другой линии. Скорость существенно выше – от 0,5 до 2,5 МБ/с.

Режим *ЕСР*:

Этот режим характеризуется высокими скоростями. Порт более стандартизован. Канальная адресация.

Два типа циклов:

1. Циклы записи/чтения данных.

2. Командные циклы:
 - передача канальных адресов;
 - передача счетчиков *RLC*.

3.3.5. Конфигурирование портов

Различают следующие виды конфигурирования портов:

- предварительное конфигурирование аппаратных средств (*SETUP*);

- текущее переключение режимов работы порта осуществляется прикладным ПО.

Способы конфигурирования аппаратных средств:

1. Базовый адрес может принимать три значения:

- 3BCh (*LPT1*)

- 378h (*LPT2*)

- 278h (*LPT3*)

2. Линии запросов на прерывания:

LPT1 – *IRQ7*

LPT2 – *IRQ5*

LPT3 – *IRQ3*

3. Использование канала *DMA*. Здесь конфигурируются наличие разрешения и номер канала.

4. Параллельные режимы работы порта:

- *SPP* – стандартный режим [000];

- *PS/2* – возможность реверса канала [001];

- *FastCentronics* – аппаратное формирование протокола *Centronics* с использованием *FIFO*-буфера и *DMA* [010];

- *EPP* – режим работает либо по обычному режиму, либо по *IPP* в зависимости от используемого регистра [100];

- *ECP* – универсальный. Может переводиться в любой режим.

По умолчанию [011];

- *ECP+EPP* – совместный.

3.3.6. Системная поддержка порта и функции BIOS

При начальном тестировании *BIOS* проверяет наличие порта и помещает адреса в счетчик *BIOS* по заданным адресам. Адреса обнаружения *LPT*-портов: 0408h, 040Ah, 040Ch, 040Eh. Если порт не обна-

ружен, то по этому адресу записывается "0". В адреса 0478h, 0479h, 047Ah, 047Bh заносятся соответствующие *TimeOut* для этих портов.

Программное прерывание 17h. Есть возможность написания собственных драйверов.

В зависимости от содержимого *AH*:

– 00h – вывод символа из регистра *AL* по протоколу *Centronics* в принтер;

– 01h – инициализация интерфейса и принтера;

– 02h – опрос состояния принтера.

Поддержка *PNP* для параллельных портов. На аппаратном уровне необходим соответствующий контроллер. Если ПУ поддерживает *PNP*, то оно может сообщить о себе идентификатор производителя, идентификатор модели и набор поддерживаемых команд. ОС предпримет действия по установке соответствующего ПО. Если ОС обнаружит устройство, отличное от того, что записано в реестре для данного порта, то она попытается установить соответствующий драйвер. Если ОС не замечает устройство, то неисправен либо кабель, либо само устройство.

3.4. *Game*-порт

С самых первых моделей *IBM PC* был введен и фактически стандартизован интерфейс игрового адаптера *Game-порт*, к которому можно подключить до двух джойстиков (руль, педали или иные устройства).

Суммарно на порте доступно 4 координатных датчика (*X1*, *X2*, *Y1* и *Y2*), изменяющих сопротивление, и 4 дискретных входа для кнопок управления.

3.4.1. Достоинства и недостатки

Достоинства:

– надёжность, связанная с конструктивом разъёма и защитой по питанию в большинстве материнских плат;

– поддержка в большинстве существующих ОС (*UNIX*, *DOS*, *AmigaOS*, *Microsoft Windows*(до win 7) и *Windows NT*, ОС на ядре *Linux* и др.).

Недостатки:

- низкая пропускная способность порта;
- ограниченные возможности (отсутствие «*AutoFire*» и т. д.);
- большая нагрузка ЦП (только *PC*-архитектура, на *Amiga* – специальный чип).

3.4.2. Способы подключения

Современные игровые устройства имеют свой интеллект (микрочип) и подключаются к компьютеру цифровым интерфейсом по шине *USB* или через *COM*-порт. Их функциональные возможности богаче, они позволяют устанавливать и двустороннюю связь с игроком (вводить механические воздействия).

3.4.3. Адаптер Game-порта

Адаптер *Game*-порта (рис. 3.18) отличается следующим:

- имеет в пространстве ввода-вывода один регистр с адресом $201h$;
- преобразование выполняется не быстро (до 1,12 мс);
- порт имеет разъем-розетку *DB-15S*;
- через разъем «*Game*» вместе с джойстиком могут подключаться и внешние *MIDI*-устройства.

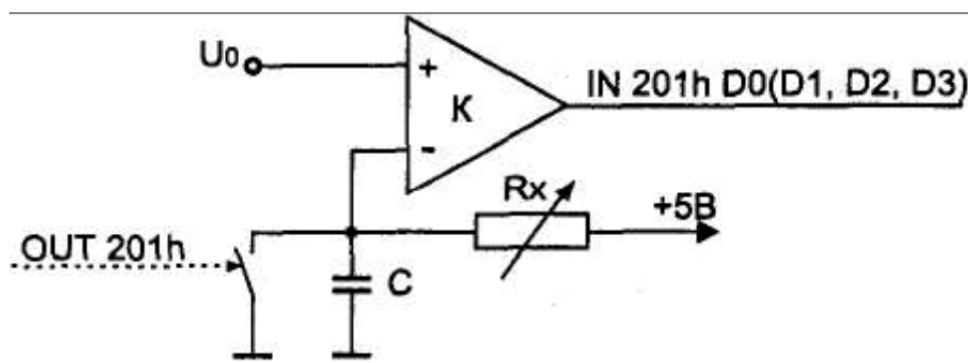


Рис. 3.18. Адаптер *Game*-порта

3.4.4. Подключение датчиков к игровому адаптеру

Подключение датчиков к игровому адаптеру показано на рис. 3.19. Системная поддержка джойстика: сервис *BIOS Int 15h* при *AH = 84h*. При вызове в *DX* задается код подфункции:

– $DX = 0$ – опрос кнопок, возвращает в $AL[7:4]$ состояние кнопок (соответствует битам порта 201h);

– $DX = 1$ – чтение координат X, Y джойстика А (в регистры AX, BX) и В (в CX, DX). При ошибочном задании кода в DX устанавливается $CF = 1$. Стандартный джойстик поддерживается и ОС *Windows*.

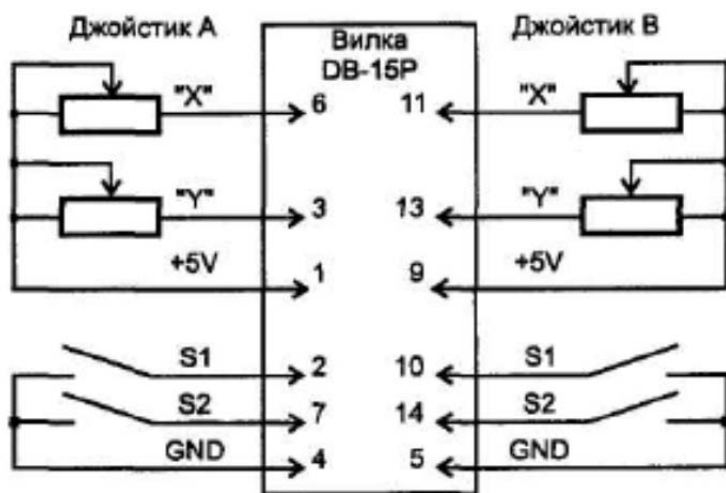


Рис. 3.19. Подключение датчиков к игровому адаптеру

Темы для закрепления знаний

1. Интерфейс *RS-232C*. Общие характеристики. Схемы соединения АПД и АКД.
2. Интерфейс *RS-232C*. Типы разъемов. Сигналы.
3. Интерфейс *RS-232C*. Алгоритм управления.
4. Родственные интерфейсы и преобразователи уровня. Схемы соединения и параметры.
5. Интерфейс *RS-232C*. Асинхронный режим. Формат передачи и временная диаграмма. . Выявление ошибок.
6. Интерфейс *RS-232C*. Управление потоком данных. Аппаратный протокол *RTS/CTS*.
7. Интерфейс *RS-232C*. Управление потоком данных. Программный протокол *XON/XOFF*.
8. Интерфейс *RS-232C*. МС асинхронных приемопередатчиков (*UART*). Регистры *UART*. Марки и особенности МС.
9. Интерфейс *RS-232C*. МС асинхронных приемопередатчиков (*UART*).Схема *UART*.

10. Системная поддержка и конфигурирование *COM*-порта. Использование *COM*- порта. *COM*-порт и *PnP*.

11. *Centronics*, *IEEE1284*, ИРПР (*LPT*-порт), общие характеристики. Схемы соединения.

12. *Centronics*, *IEEE1284*, ИРПР (*LPT*-порт) – электрические параметры. Конструктивы (разъемы и кабели).

13. *Centronics*, *IEEE1284*, ИРПР (*LPT*-порт) – логическая организация обмена и временные диаграммы при байтном режиме.

14. *Centronics*, *IEEE1284*, ИРПР (*LPT*-порт) – логическая организация обмена и временные диаграммы при режиме *EPPO* – особенности режима *ЕСР*.

15. *Centronics*, *IEEE1284*, ИРПР (*LPT*-порт) – конфигурирование, системная поддержка.

16. *Centronics*, *IEEE1284*, ИРПР (*LPT*-порт) – использование параллельных интерфейсов.

17. *Game*-порт.

Глава 4. ПОСЛЕДОВАТЕЛЬНЫЕ ИНТЕРФЕЙСЫ

Как было отмечено в гл. 1, в настоящее время прослеживается четкая тенденция перехода к последовательным интерфейсам. Примеры тому – интерфейсы *PCI-E*, *SATA* и т.д. Естественно, что и вновь разрабатываемые интерфейсы являются последовательными. Наиболее старым из них считается *IEEE-1394 (Fire Ware)*. В свое время он был самым скоростным, но в настоящее время по производительности он уступает и *USB 3.0*, и тем более *Thunderbolt*.

4.1. Интерфейс *USB*

UniversalSerialBus – универсальная последовательная шина. Является промышленным стандартом. Предназначена для подключения многих устройств.

4.1.1. Особенности и преимущества *USB*

1. Легко реализуемое расширение периферии *PC*.
2. Дешевое решение, поддерживающее высокие скорости передачи (до 12 Мбит/с).

3. Полная поддержка в реальном времени передачи аудио- и (сжатых) видеоданных.

4. Гибкость протокола смешанной передачи изохронных данных и асинхронных сообщений.

5. Интеграция с выпускаемыми устройствами.

6. Доступность *PC* всех конфигураций и размеров.

7. Обеспечение стандартного интерфейса хорошего качества.

8. Создание новых классов устройств, расширяющий *PC*.

9. Широкий спектр подключаемых ПУ.

Особенности с точки зрения пользователя:

– простота кабельного соединения;

– скрытие подробностей электрического подключения от конечного пользователя;

– самоидентифицирующиеся ПУ, автоматическая связь устройств с драйверами и конфигурирование;

– возможность динамического подключения и конфигурирования ПУ.

4.1.2. Скорости

LS (LowSpeed) – низкая – 1,5 Мб/с (*USB 1.0*);

FS (Full Speed) – полная – 12 Мб/с (*USB 1.0*);

HS (High Speed) – высокая 480 Мб/с (*USB 2.0*);

SS (Super Speed) – сверхвысокая – 5 Гбит/с (*USB 3.0*);

SS+ (Super Speed+) – сверхвысокая плюс 10 Гбит/с (*USB 3.1*).

USB 1.0

Версия 1.0 была опубликована в январе 1996 г. Архитектура *USB* определяется следующими критериями:

– легко реализуемое расширение периферии *PC*;

– дешевое решение, поддерживающее скорость передачи до 12 Мбит/с;

– полная поддержка в реальном времени передачи аудио- и (сжатых) видеоданных;

– гибкость протокола смешанной передачи изохронных данных и асинхронных сообщений;

– интеграция с выпускаемыми устройствами;

– доступность в *PC* всех конфигураций и размеров;

- обеспечение стандартного интерфейса, способного быстро завоевать рынок;
- создание новых классов устройств, расширяющих *PC*;
- с точки зрения конечного пользователя привлекательны следующие черты *USB*:
 - простота кабельной системы и подключений;
 - скрытие подробностей электрического подключения от конечного пользователя;
 - самоидентифицирующиеся ПУ, автоматическая связь устройств с драйверами и конфигурирование;
 - возможность динамического подключения и конфигурирования ПУ;
 - высокая скорость обмена – 12 Мбит/с;
 - максимальная длина кабеля для высокой скорости обмена – 3 м;
 - низкая скорость обмена – 1,5 Мбит/с;
 - максимальная длина кабеля для низкой скорости обмена – 5 м;
 - максимальное количество подключённых устройств (включая размножители) – 127;
 - возможно подключение устройств с различными скоростями обмена;
 - напряжение питания для периферийных устройств – 5 В;
 - максимальный ток потребления на одно устройство – 500 мА.

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* и др.

Разработку второй спецификации *USB* начали компании *Intel*, *Microsoft*, *Compaq* и *NEC*, а впоследствии к ним присоединились *Hewlett-Packard*, *Lucent Technologies* и др.

USB 2.0

Впервые о выходе *USB 2.0* заговорили в середине 1999 г., но окончательная редакция шины была представлена только весной 2000 г. Основное отличие новой версии от предыдущей – 40-кратное (!) увеличение максимальной пропускной способности до 480 Мбит/с (ранее

планировалось лишь 20-кратное увеличение). Кроме того, она полностью совместима с *USB 1.1* и так же физически реализована. Значит, всякое старое устройство *USB* будет работать с новыми разветвителями *USB*. Благодаря тому что не внесено изменений в строение проводов и разъемов, для новых продуктов подойдут и старые кабели. Сохранилась прежней и модель использования устройств *USB*.

USB 2.0 отличается от *USB 1.1* только большей скоростью и небольшими изменениями в протоколе передачи данных для режима *Hi-speed* (480 Мбит/с). Существуют три скорости работы устройств *USB 2.0* :

- *Low-speed* 10 – 1500 Кбит/с (используется для интерактивных устройств: клавиатуры, мыши, джойстика);
- *Full-speed* 0,5 – 12 Мбит/с (аудио-видеоустройства);
- *Hi-speed* 25 – 480 Мбит/с (видеоустройства, устройства хранения информации).

На самом деле, хотя в теории скорость *USB 2.0* может достигать 480 Мбит/с, устройства типа жёстких дисков и вообще любых носителей информации в реальности никогда не достигают такой скорости обмена по шине, хотя и могут развивать её. Это можно объяснить достаточно большими задержками шины *USB* между запросом на передачу данных и собственно началом передачи и другими накладными расходами.

USB OTG

USB OTG (аббр. от *On-The-Go*) – дальнейшее расширение спецификации *USB 2.0*, предназначенное для лёгкого соединения периферийных *USB*-устройств друг с другом без необходимости подключения к ПК. Например, цифровой фотоаппарат можно подключать к фотопринтеру напрямую, если они оба поддерживают стандарт *USB OTG*. Этот стандарт возник из-за резко возросшей в последнее время необходимости надёжного соединения различных *USB*-устройств без использования ПК. В данной спецификации устройства обходятся без персонального компьютера, т.е. выступают как одноранговые передатчики (на самом деле только создаётся такое ощущение). В действительности же устройства определяют, кто из них будет мастер-устройством, а кто подчиненным.

USB 3.0

В спецификации *USB 3.0* разъёмы и кабели обновлённого стандарта физически и функционально совместимы с *USB 2.0*. Кабель *USB 2.0* содержит в себе четыре линии – пару для приёма / передачи данных, плюс и ноль питания. В дополнение к ним *USB 3.0* добавляет еще четыре линии связи (две витых пары), в результате чего кабель стал гораздо толще. Новые контакты в разъемах *USB 3.0* расположены отдельно от старых на другом контактном ряду. Теперь можно будет с лёгкостью определить принадлежность кабеля к той или иной версии стандарта, просто взглянув на его разъём. Спецификация *USB 3.0* повышает максимальную скорость передачи информации до 4,8 Гбит/с, что на порядок больше 480 Мбит/с, которые может обеспечить *USB 2.0*. Спецификация *USB 3.0* повышает эту скорость еще в 2 раза – до 10 Гбит/с, что сравнимо с частотой самого современного интерфейса *Thunderbolt*.

Таким образом, скорость передачи возрастает с 60 Мбайт/с до 600 Мбайт/с и позволяет перекинуть 1 Тб не за 8 – 10 ч, а за 40 мин – 1 ч. Версия 3.0 обладает не только более высокой скоростью передачи информации, но и увеличенной силой тока с 500 мА до 900 мА. Версия 3.1 вообще уникальна по возможностям питания, так как позволяет не только увеличить ток до 2 А, но и использовать другие напряжения 12 В и 20 В при токе до 5 А. Отныне пользователь может не только подпитывать от одного хаба большее количество устройств, но и сами устройства во многих случаях смогут избавиться от отдельных блоков питания.

Компания *Asus* выпустила материнскую плату *P6X58 Premium*, у которой есть два *USB 3.0*-порта. А компания *Gigabyte* выпустила первую материнскую плату с поддержкой *USB 3.0* и *SATA 6Gb/s* для процессоров *AMD* – *Gigabyte GA-790FXTA-UD5*. Порты *USB 3.0* на материнской плате синего цвета.

4.1.3. Структура USB и организация шины

Шина *USB* имеет топологию типа многоярусная звезда (рис. 4.1). Допускается 5 уровней хабов (больше нельзя). Все подключенные устройства могут обслуживаться хостом одновременно. Каждый ка-

большой сегмент соединяет две точки (хаб с хабом или хаб с функцией). Примеры функций: указатели, сканеры, принтеры и т.п.

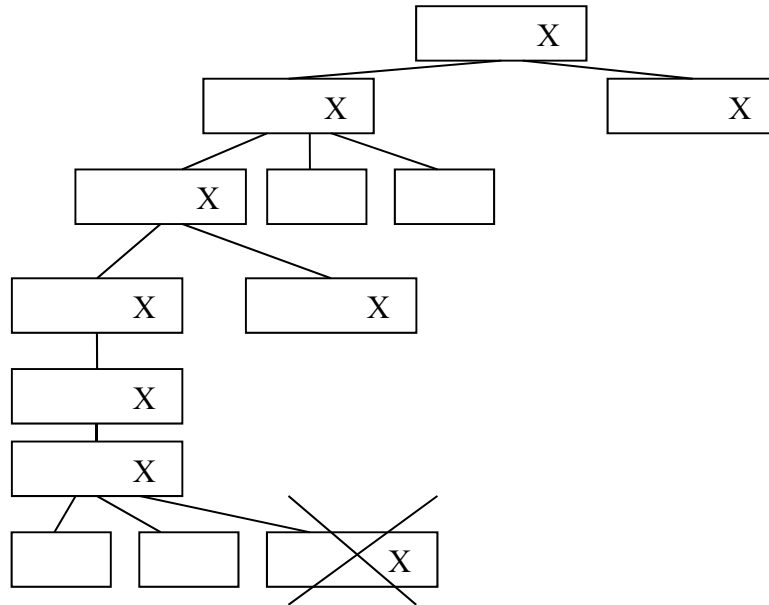


Рис. 4.1. Структура *USB* и организация шины

4.1.4. Протоколы и уровни взаимодействия между хостом и ПУ

Протокол включает три уровня взаимодействия (рис. 4.2):

1. Верхний – уровень функций. В хосте для уровня функций существует программная часть *ClientSW*. Он взаимодействует с функцией *Function* (виртуальное взаимодействие на уровне данных).

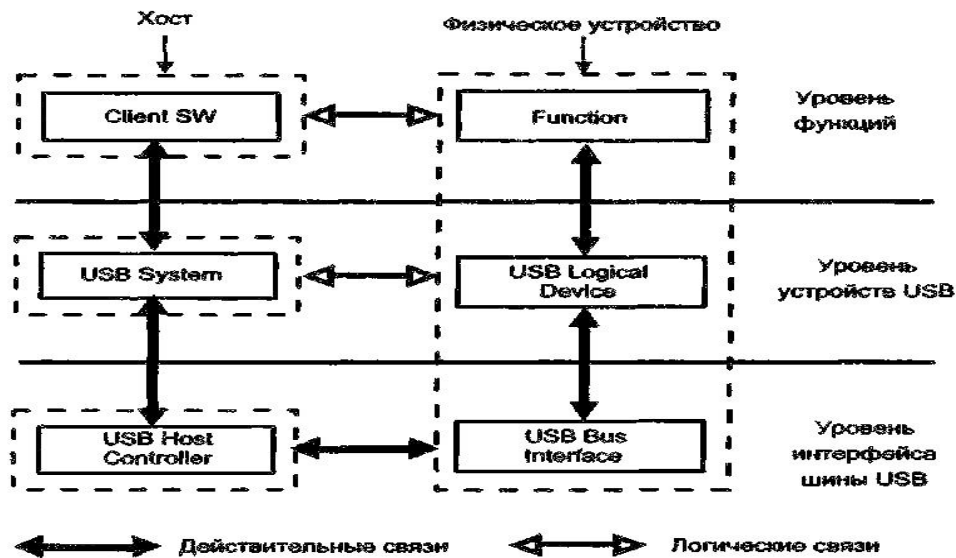


Рис. 4.2. Уровни взаимодействия между хостом и ПУ

2. Уровень устройств *USB*.

USB-system взаимодействует виртуально с *USBLogicalDevice*.

ClientSW – ПО, соответствующее конкретному устройству, исполняемое на хосте. Может являться составной частью ОС или отдельным ПО (драйвер) устройства.

USBSystemSW – системная поддержка *USB*, независимая от конкретных устройств (драйвер шины).

USBHostController – аппаратно-программные средства для подключения устройств *USB* к хост-компьютеру.

3. Уровень интерфейса шины.

Физическое устройство *USB* – устройство на шине, выполняющее функции интерфейса конечного пользователя.

4.1.5. Физический интерфейс

Физический интерфейс включает в себя четыре провода:

2 провода – сигнальные, 1 провод – земля, 1 провод – напряжение питания. Контакты:

1-й контакт – напряжение U_{bus} ($U_{пит} = \pm 5 \text{ В}$);

2-й контакт – $D+$;

3-й контакт – $D-$;

4-й контакт – земля (-5 В).

Соединение

Два вида соединения:

– витая пара (максимальная длина 5 м) – экранированная;

– витая пара (максимальная длина 3 м) – неэкранированная.

Различие в качестве и дальности проводов. Витая пара используется для подключения полноскоростных устройств (полная и высокая скорость). Оно представлено на рис. 4.3.

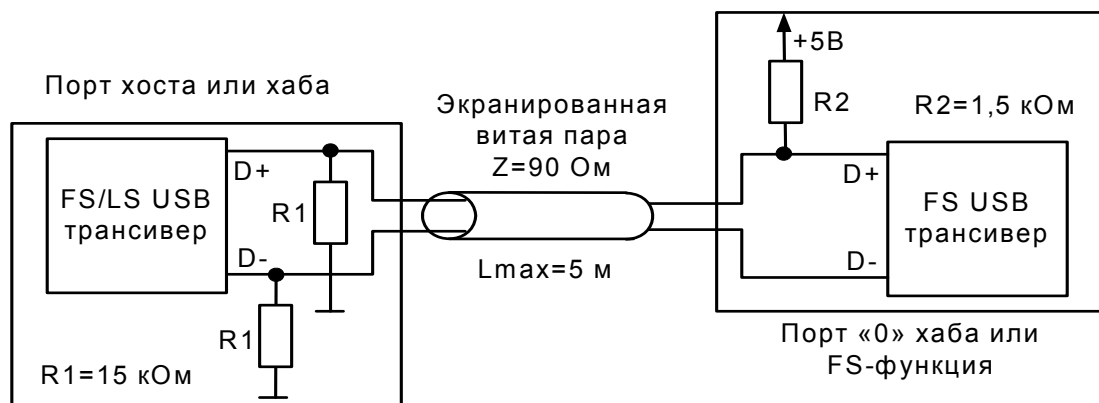


Рис. 4.3. Витая пара

Неэкранированная витая пара может использоваться только для подключения *LS*-функции. При этом длина может быть не более 3 м (рис. 4.4).

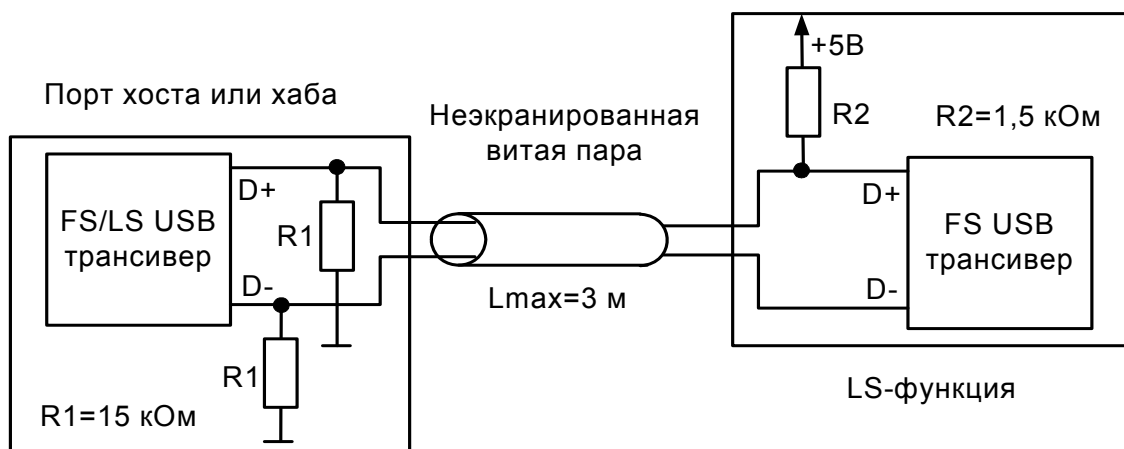


Рис. 4.4. Неэкранированная витая пара

Кабели двух типов:

Тип *A* – для подключения к хабам устройств, не отсоединяемых на своем конце (мышь, клавиатура).

Тип *B* – для устройств, у которых кабель может отсоединяться (принтер, сканер).

Разъемы типов *A* и *B*, представленные на рис. 4.5, различаются механически, 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.

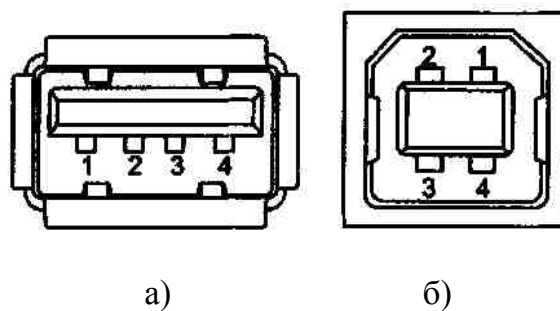


Рис. 4.5. Типы разъемов: а – разъем типа *A*; б – разъем типа *B*

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. Максимальный потребляемый ток ограничен в сумме не более 500 мА.

На физическом уровне *USB* для формирования сигналов *D+* и *D-* используются различные приемопередатчики: универсальный дифференциальный – для скоростей *LS/FS*, дифференциальный источник тока – для скорости *HS*, линейные приемники и детекторы сигналов. Уровни сигналов на приемнике должны быть следующими:

- *Diff0*: $(D+) - (D-) > 200$ мВ при $D+ > 2$ В;
- *Diff1* $(D-) - (D+) > 200$ мВ при $D- > 2$ В;
- *SE0* (*single-ended zero*) $D+ < 0,8$ В, $D- < 0,8$ В.

При скоростях *LS/FS* для формирования сигнала используется управляемый дифференциальный передатчик, на скорости *HS* дифференциальный токовый передатчик. Он формирует ток 17,78 мА, который протекает через нагрузку 22,5 Ом, что обеспечивает сигнал около ± 400 мВ. На высокой скорости *HS* к форме сигнала предъявляются очень высокие требования.

Передатчик должен иметь возможность перехода в высокоимпедансное состояние.

При передаче информации используется кодирование *NRZI* (без возврата к нулю): при передаче нулевого бита в начале битового интервала состояние сигнала меняется на противоположное; при передаче единичного – не меняется. Для исключения потери синхронизации на монотонном сигнале применяется вставка бит: после каждых 6 подряд следующих единиц вставляется нуль.

Физический интерфейс версий *USB* 3.0 и 3.1 существенно отличается. Это связано с тем, что появились две дополнительные витые пары, позволяющие обеспечить дуплексный режим. Эти линии используют другой способ кодирования – 8/10. Их наличие существенно изменило разъемы и кабели. Кабель *USB* 3.0 значительно толще прежнего (рис. 4.6). Разъемы имеют дополнительные контакты, что привело к увеличению размеров разъема типа В (рис. 4.7) и микро-В (рис. 4.8).

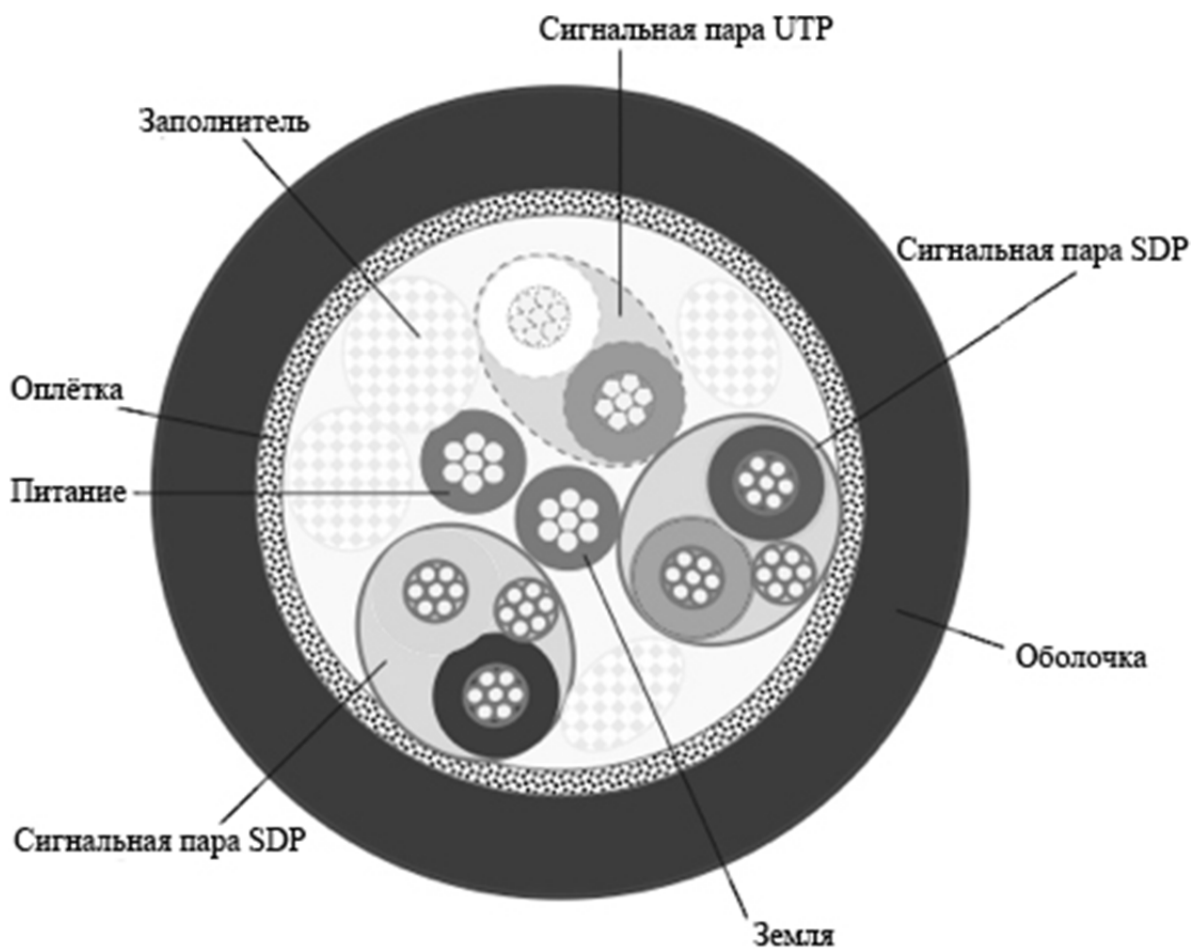


Рис. 4.6. Схема кабеля *USB 3.0*

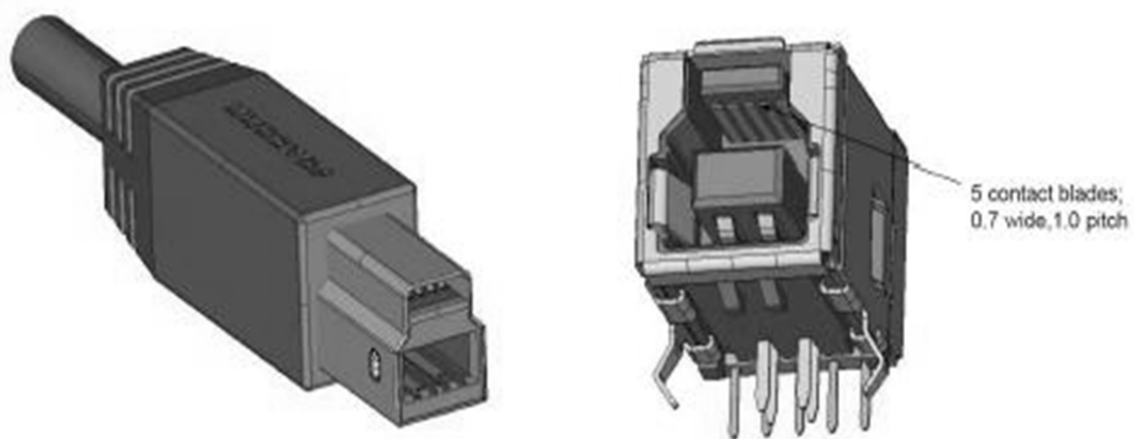


Рис. 4.7. Розетка и вилка *USB 3.0*, тип В

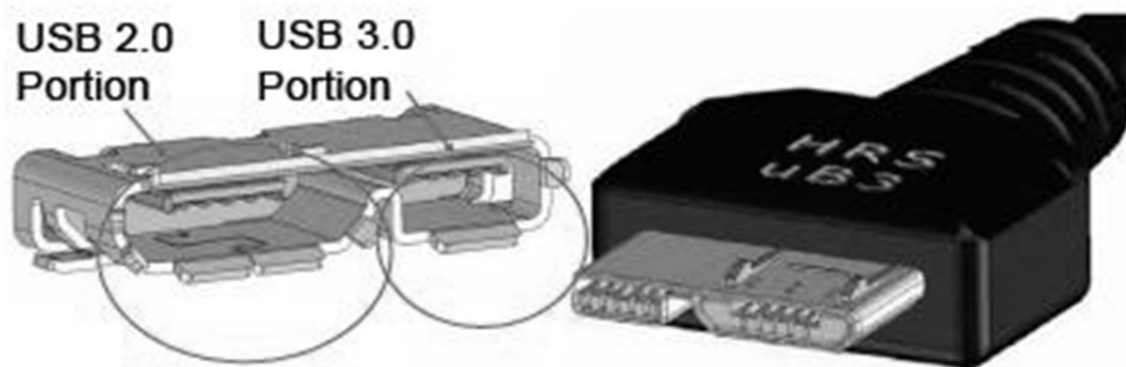


Рис. 4.8. Розетка и вилка *USB 3.0*, тип *micro-B*

4.1.6. Модель передачи данных

Устройство *USB* – набор конечных точек. Конечная точка имеет свой номер и описывается следующими параметрами:

1. Номер точки.
2. Требуемая частота доступа к шине или допустимая задержка.
3. Требуемая полоса пропускания канала.
4. Требования к обработке ошибок.
5. Максимальные размеры передаваемых и принимаемых пакетов.
6. Тип обмена (тип передачи).
7. Направление обмена при передаче массивов и изохронных передачах.

Конечная точка с номером 0 используется для инициализации общего управления и опроса состояния устройства. Она конфигурируется при включении питания. На каждом устройстве разрешены до 15 конечных точек (для низкоскоростных – до 2).

Канал (*pipe*) – модель передачи данных между конечной точкой и хост-контроллером.

Могут быть каналы двух видов:

- потоки;
- сообщения, массивы.

Поток может реализовать следующие виды обмена:

- сплошной;
- изохронный;
- прерываний.

Сообщение имеет формат, определенный *USB*. Хост посылает запрос к конечной точке, после которого передает пакет сообщения и информацию о состоянии конечной точки (квитанция о приеме).

4.1.7. Типы передачи данных

В *USB* используются четыре базовых типа передачи данных:

– управляющие посылки – для конфигурирования во время подключения или для управления устройствами в процессе работы. Они обеспечивают гарантированную доставку данных. Длина поля данных для *FS*: $L \leq 4$ Б, для *LS*: $L \leq 8$ Б;

– сплошные передачи, или передачи массива данных – это передачи больших пакетов без жесткого требования по времени доставки. Поле данных имеет длину $L = 8, 16, 32, 64$ Б. Сплошные передачи занимают свободный остаток пропускной способности канала, следовательно, имеет самый низкий приоритет;

– прерывания должны обслуживаться быстро по мере поступления. Длина поля данных *FS*: $L < 64$ Б, *LS*: $L < 8$ Б, *HS*: $0 - 1024$ Б. Установлены пределы времени обслуживания: *LS*: $t = 10 - 25$ мс, *FS*: $t = 1 - 255$ мс, *HS*: $t \geq 125$ мс;

– изохронные передачи – непрерывные передачи в реальном времени. Занимают предварительно согласованную часть полосы шины и имеют заданную задержку доставки. Скорость передачи для полной скорости *FS* = 1,023 МБ/с, т. е. передача займет 70 % канала. Скорость передачи для высокой скорости *HS* = 24 МБ/с (та же занятость шины). Изохронные передачи используются для видео- и аудиоинформации в реальном времени.

Полная пропускная способность шины делится между всеми каналами. Если новый канал требует ширину полосы, которой нет в распоряжении, то этот канал не будет открыт. Чем больше пропускная способность, тем больше буфер. Все устройства должны иметь буфер.

4.1.8. Протокол обмена

Весь обмен происходит кадрами. Кадр состоит из транзакций. Транзакция состоит из пакетов. Пакеты бывают трех типов:

– пакет-маркер;

- пакет данных;
- пакет-подтверждение (квитанция); при изохронных передачах этот пакет отсутствует.

Параметры кадра: длительность равна 1 мс, частота 1кГц. На высокой и сверхвысокой скоростях каждый кадр дополнительно делится на микрокадры длительностью 125 нс и частотой 8 кГц.

Каждая транзакция планируется контроллером и идентифицируется по следующим параметрам:

1. Наличие пакета-маркера. Он описывает тип и направление передачи, адрес устройства и номер конечной точки.
2. Источник данных, определяемый маркером. Передает пакет данных или уведомление об отсутствии данных для передачи.
3. После успешного приема пакета приемник присылает подтверждение, причем может быть получен отказ от приема данных, если буфер переполнен.

Обнаружение ошибок

Для обнаружения ошибок передачи каждый пакет имеет контрольные поля CRC-кодов (позволяет точно обнаружить двойные ошибки). При обнаружении ошибки контроллер производит 3-кратную попытку передачи.

Устойчивость к ошибкам обеспечивают следующие свойства *USB*:

1. Высокое качество сигнала, достигаемое за счет дифференциации сигнала приемопередатчика и экранированных кабелей.
2. Защита полей управления и данных с помощью CRC-кодов.
3. Обнаружение подключения-отключения устройств и конфигурирование ресурсов на системном уровне.
4. Самовосстановление протокола с *TimeOut* при утере данных.
5. Управление потоком для обеспечения изохронности и управления аппаратными буферами.
6. Независимость одних функций от неудачных обменов других функций.

Форматы пакетов

Биты в байтах передаются последовательно, начиная с младшего. Используются состояния *J* и *K*. Уровни сигналов зависят от скорости: при *LS J* соответствует уровень *Diff0*, *K* – *Diff1* при *FS* и *HS* – наоборот.

Пакет начинается с поля синхронизации *SYNC*: *KJKJKJKK*. Последние 2 бита *KK*-маркер начала пакета (*SOP*). Далее идет идентификатор пакета – 4-битовое поле (*PID*) [3:0]. Далее этот идентификатор *PID* в инвертированном виде [3:0].

Итого: *KJKJKJKK* | *PID* | инверсный (*PID*).

При скоростях *LS* и *FS* используются пакеты 10 типов (4 класса):

Четыре типа объединяют в класс ***Token***:

- *OUT* – 0001 – адрес функции и номер конечной точки – маркер транзакции функции;
- *IN* – 1001 – адрес функции и номер конечной точки – маркер транзакции хоста;
- *SOF* – 0101 – маркер начала кадра;
- *SETUP* – 1101 – адрес функции и номер конечной точки – маркер транзакций с управляющей точкой.

Класс ***Data*** – пакеты данных с четным и нечетным *PID* чередуются для точной идентификации подтверждений:

- *Data OUT* – 0011;
- *Data IN* – 1011.

Класс ***HandShake*** – подтверждения:

- *ACK* – 0010 – подтверждение безошибочного приема пакета;
- *NAK* – 1010 – приемник не смог принять или передать данные;
- *STALL* – 1110 – конечная точка требует вмешательства хоста.

Класс ***Special***:

- *PRE* – 1100 – преамбула передачи на низкой скорости.

В *USB 2.0* прежде всего на высокой скорости *HS* дополнительно используются шесть типов пакетов:

- *PING* – 0100 – пробный маркер управления потоком;
- *DANA2* – 0111 и *MDATA* – 1111 – дополнительные типы пакетов данных, используемые в транзакциях с широкополосными изохронными точками;
- *NYET* – 0110 – подтверждение безошибочного приема, но указание на отсутствие места для следующего пакета максимального размера;
- *ERR* – 1100 – сигнал об ошибке в расщепленной транзакции;
- *SPLIT* – 1000 – маркер расщепленной транзакции.

Количество и расположение полей зависит от типа пакета и для наиболее часто используемых пакетов приведено на рис. 4.9. Каждый пакет всегда начинается с поля *SYNC* и заканчивается переводом канала в состояние *IDLE*, когда передача данных не происходит.

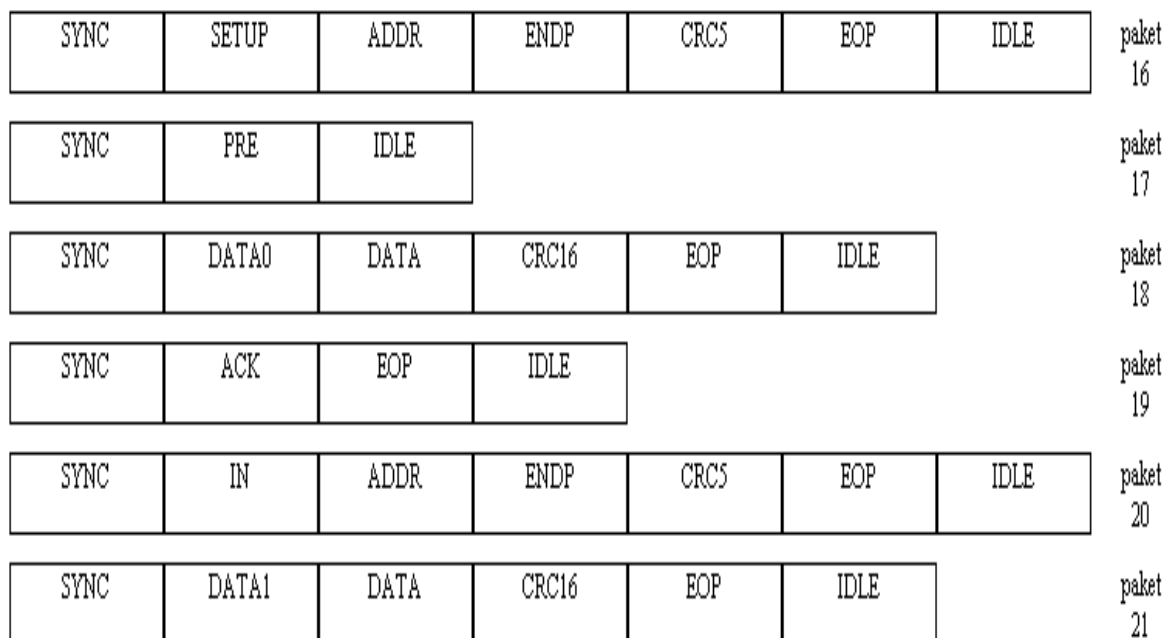


Рис. 4.9. Примеры следования полей в пакетах транзакций

Поля пакетов имеют следующее значение:

sync – все пакеты начинаются с поля синхронизации *SYNC*. Оно является кодированной последовательностью, которая генерирует максимальную плотность границ передач. Оно используется входной схемой для синхронизации входных данных с локальными часами и определено как последовательность из 8 бит *SYNC* → 00000001;

setup – идентификатор пакета *PID* следует непосредственно за *SYNC* полем каждого пакета *USB*. *PIDS* разделены на 4 кодовые группы: маркер, данные, квитиование и специальные. Тип *PID SETUP* относится к маркерным пакетам, он предваряет адрес и номер конечной точки функции для установки ее в точку управления хост-контроллером. *SETUP* состоит из 4 бит, *SETUP* → 1101;

addr – поле адреса функции (*ADDR*) определяет адрес функции, которая является источником или приемником пакета данных в зависимости от значения маркера *PID*. Поле *ADDR* определено для марке-

ров *IN*, *SETUP* и *OUT*. Каждое значение *ADDR* определяет единственную функцию. Поле *ADDR* состоит из 7 бит и может определять до 128 функций;

endp – маркер конца пакета. Состоит из 4 бит и вставляется только в конце маркерных пакетов перед данными *CRC*;

crc5 – служит для проверки целостности данных пришедших пакетов с использованием циклического *CRC*-кода 5-го порядка. Для маркерных пакетов предусмотрено 5-битное поле *CRC*. Оно покрывает поля *ADDR* и *ENDP* маркеров *IN*, *SETUP* и *OUT* или поле отметки времени маркера *SOF*. Полином генерируется по формуле $G(x)=X^5+X^2+1$. Двоичная битовая маска, которая отражает этот полином: 00101. Если все биты маркера получены без ошибки, пятибитный остаток в приемнике будет 01100;

crc16 – для проверки целостности данных пришедших пакетов. Для пакетов с данными применяется 16-битный *CRC*, его полином генерируется по формуле $G(x) = X^{16}+X^{15}+X^2+1$. Двоичная битовая маска, которая отображает этот полином: 1000 0000 0000 0101. Если все данные и биты *CRC* получены без ошибки, 16-битный остаток будет: 1000 0000 0000 1101;

eop – все пакеты передаваемой информации имеют четкие разграничители начала и конца пакета. Начало пакета *SYNC*, *EOP* – конец пакет;

idle – это состояние, когда в шине не происходит передача данных. На данных диаграммах принято, что после передачи каждого пакета дополнительные данные для передачи отсутствуют и шина переводится в состояние *IDLE*. В данном состоянии концентраторы (хабы) и функции не переходят в подвешенное состояние (*suspend mode*) и требуемая мощность выдается в полном объеме;

pre – это идентификатор типа пакета (*PID*), относящегося к кодовой группе – специальных *PIDS*. *PRE* – это выдаваемая хостом команда, которая разблокирует трафик вниз по иерархии шины к низкоскоростным устройствам. *PRE* состоит из 4 бит: 1100;

data0 – это идентификатор пакета (*PID*), относящегося к кодовой группе пакетов данных. Данным идентификатором пакета обозначаются четные пакеты данных. Данный *PID* состоит из 4 бит: 0011;

data1 – это идентификатор пакета (*PID*), относящегося к кодовой группе пакетов данных. Данным идентификатором пакета обозначаются нечетные пакеты данных. Данный *PID* состоит из 4 бит: 1011;

data – это непосредственно передаваемые данные. Это поле может быть длиной от 0 до 1023 байт и должно содержать целое число байт. Размер и форма представления данных должна быть в соответствии с типом текущей передачи;

ack – это тип идентификатора пакета (*PID*), относящегося к группе пакетов квитирования. Данный *PID* информирует о том, что приемник принял пакет данных без ошибок. Данный *PID* состоит из 4 бит: 0010;

in – это тип идентификатора пакета (*PID*), относящегося к группе маркерных пакетов. Данным пакетом предваряются адрес и номер конечной точки функции. Данный *PID* состоит из 4 бит: 1001.

4.1.9. Синхронизация при изохронной передаче

Для синхронизации в *USB* используются следующие частоты, имеющиеся в системе:

1. Частота выборки – для источника данных и приемника.
2. Частота шины – частота кадров: 1кГц – для *FS* и микрокадров: 8 кГц – для *HS*.
3. Частота обслуживания – частота, с которой клиентское ПО обращается к драйверам *USB* для передачи и приема изохронных данных.

Типы отклонений, возможные между парами синхросигналов при отсутствии общего источника синхронизации:

1. Дрейф – отклонение формально одинаковых частот от номинала.
2. Дрожание – колебание частот относительно номинала.
3. Фазовый сдвиг – если сигналы не связаны ФАПЧ.

Для подстройки частот используют три подхода:

1. Синхронизация внутреннего генератора устройства с маркерами *SOF*.
2. Подстройка частоты кадров под частоту устройства.
3. Согласование скорости передачи (приема) устройства с частотой кадров.

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

4.1.10. Системное конфигурирование

Хост разрешает работу порта и адресует к устройству через канал управления, используя нулевой адрес *USBDefaultAddress*. 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 (*ControlPipe*) 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 – функция, то уведомление о подключении передается диспетчером *USB* заинтересованному устройству.

Когда устройство отключается, хаб автоматически запрещает соответствующий порт и сообщает об отключении контроллеру, который удаляет сведения о данном устройстве из всех структур данных.

Нумерация устройств выполняется динамически по мере их подключения 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 *Resrt* 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 со-

стояние *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* (skonфигурировано). Теперь хаб позволяет устройству потреблять от шины полный ток, заявленный в конфигурации. Устройство готово.

Все устройства должны поддерживать следующий набор операций:

- динамическое подключение/отключение (это событие отслеживается хабом);
- конфигурирование устройств, выполняемое хостом;
- передача данных посредством одного из 4 типов;
- управление энергопотреблением (при этом устройство должно поддерживать приостановку, при которой ток снижается до 500 мА или меньше);
- возможность удаленного пробуждения – позволяет приостановленному устройству подать сигнал хосту (компьютеру), который тоже может находиться в приостановленном состоянии.

Функции хаба:

1. Обеспечивает физическое подключение устройств.
2. Управляет подачей питающего напряжения на нисходящие порты.
3. Отслеживает состояние подключенных устройств, уведомляя хост об изменениях.
4. Обнаруживает ошибки на шине, выполняет процедуры восстановления и изолирует неисправные сегменты шины.

5. Обеспечивает связь сегментов шины, работающих на разных скоростях.

Состояния нисходящих портов хаба:

1. Питание отключено. На порт не подается питание (возможно только для хабов, коммутирующих питание). Выходные буферы переводятся в высокоимпедансное состояние, входные сигналы игнорируются.

2. Отсоединен. Порт не передает сигналы ни в одном направлении, но способен обнаружить подключение устройства (по отсутствию сигнала *SEO* в течение 2,5 мкс). Тогда порт переходит в состояние «Запрещен». А по уровням входных сигналов *Diff0* и *Diff1* он определяет скорость подключенного устройства.

3. 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 «Отсоединен», а если отключение обнаружено «спящим» хабом, контроллеру будет послан сигнал *Resume*.

4. Разрешен. Порт передает сигналы в обоих направлениях. По команде контроллера или по обнаружении ошибки кадра порт переходит в состояние "Отсоединен".

5. 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 активизированного устройства даже через цепочку «спящих» хабов.

4.1.11. Концентратор (хаб)

Концентратор можно разделить на две части – повторитель (*hubrepeater*) и контроллер (*hub controller*) (рис. 4.10). Он содержит следующие основные блоки (рис. 4.11):

1. Регистр адреса данных относится к управляющей части концентратора. Он содержит адрес *RAM*, по которому произойдет обмен данными.

2. Динамическая память концентратора – здесь хранятся переменные данные, необходимые для работы управляющей части концентратора, а также отложенные пакеты транзакций.

3. ПЗУ. В нем размещена микропрограмма функционирования концентратора. Данная микропрограмма непосредственно зависит от характеристик каждого конкретного концентратора.

4. Регистр адреса следующей микрокоманды. Задает адрес микрокоманды, которую в следующий такт выполнит концентратор.

5. Счетчик микрокоманд. Непосредственно задает значение в регистре микрокоманд.

6. Модуль *USB* части концентратора. Он состоит из *upstream*, *downstream* портов и повторителя и управляется контроллером концентратора. Его задача: в соответствии с командами управляющей части обеспечить прохождение трафика вверх и вниз по иерархии шины и информирование вышестоящего уровня о состояниях *downstream* портов и самого концентратора.

7. Блоки для микросхем расширения. В соответствии с вставленными микросхемами можно влиять на характеристики концентратора.

8. Непосредственно *USB*-разъемы для подключения нижних уровней устройств *USB* шины.

9. Указатель вершины стека, расположенного в *RAM*. Используется управляющей программой концентратора.

10. Указатель на данные. Используется управляющей программой концентратора. Может указывать как на данные микропрограммы, так и на данные отложенных пакетов транзакций.

11. АЛУ. Одна из частей контроллера непосредственно участвующая в вычислениях.

12. Регистр выбора операции АЛУ. Задает операцию АЛУ по команде микропрограммы.

13. Генератор и местные часы. Используется для тактирования работы концентратора, синхронизации с получаемыми данными и для планирования транзакций для *downstream* портов.

14. Регистр *B*. Что-то вроде промежуточного регистра для данных.

15. Аккумулятор. Что-то вроде промежуточного регистра для данных.

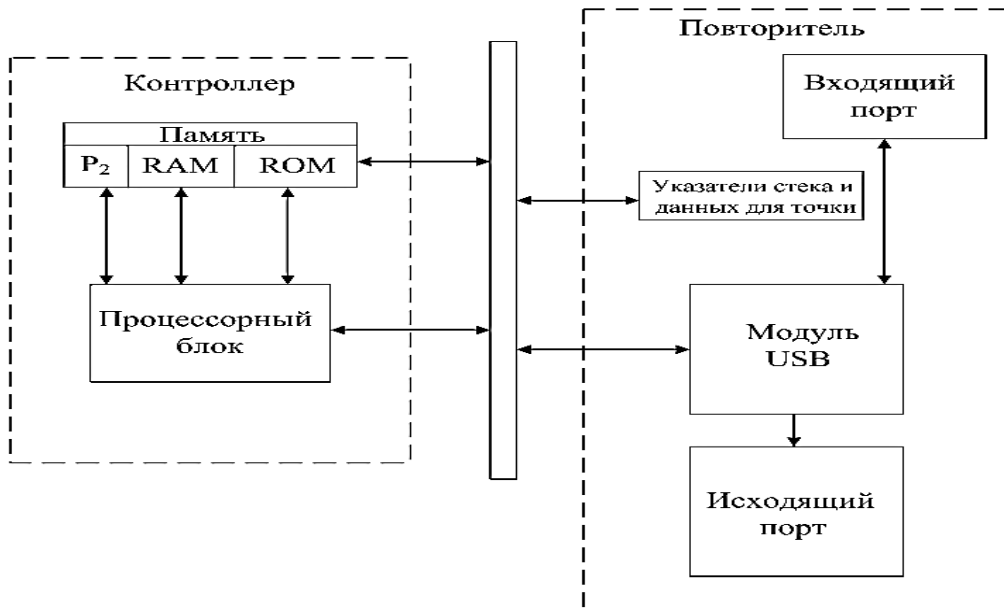


Рис. 4.10. Упрощенная структурная схема концентратора

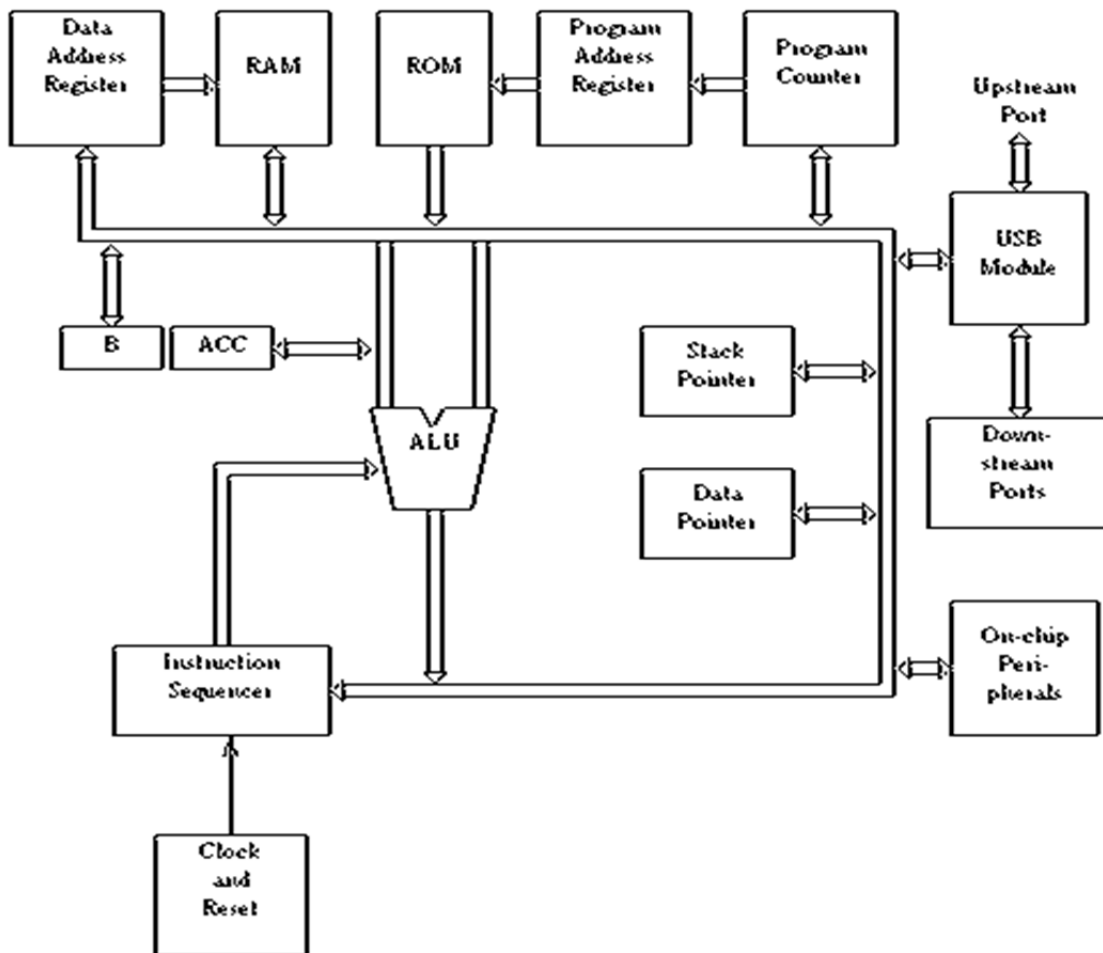


Рис. 4.11. Структурная схема концентратора

4.1.12. Хост-контроллер USB

Хост-контроллер обеспечивает взаимодействие хоста и устройств *USB*. Имеются 3 спецификации хост-контроллеров, в каждой из которых используется свой комплект драйверов хост-части.

Хост-контроллеры *UHC (Universal Host Controller)* и *OHC (Open Host Controller)* разработаны для *USB 1.x*. Расширенный хост-контроллер *EHC (Enhanced Host Controller)* разработан для поддержки высокой скорости шины *USB 2.0* и выше. Все эти варианты выполняют следующие функции:

- обнаружение подключения и отсоединения устройств *USB*;
- манипулирование потоком управления между устройствами и хостом;
- управление потоками данных;
- сбор статистики;
- управление электропитанием подключенных ПУ.

Для выполнения этих функций в хост-контроллере имеются регистры и память. В памяти хранятся структуры данных, описывающие очереди передач, список кадров и т.п. Каждая передача описывается дескриптором. Более детальные сведения об этих структурах можно найти в [5].

Уровни хоста:

1. Интерфейс шины *USB* обеспечивает физический интерфейс и протокол шины. Интерфейс шины реализуется хост-контроллером. Он отвечает за генерацию микрокадров и обменивается с ПО, используя прямое управление шиной.

2. Система *USB*, используя хост-контроллер, транслирует клиентское «видение» обмена данными с устройствами в транзакции. Она отвечает за распределение ресурсов полосы пропускания и мощности напряжения питания. Система состоит из трех основных частей:

- драйвер хост-контроллера (*HCD*) – модуль, привязанный к конкретной модели контроллера;
- драйвер *USB (USBD)* – обеспечивает основной интерфейс между клиентами и устройствами *USB*. Интерфейс между *HCD* и *USBD* не регламентируется и должен определяться разработчиком ОС;

– ПО хоста реализует функции, необходимые для работы *USB* в целом: обнаружение отключения и подключения устройства, загрузка соответствующих драйверов, нумерация устройств, распределение полосы и питания.

3. Клиенты *USB*–программные элементы, взаимодействующие с устройствами *USB*.

Урошенная структурная схема хост-контроллера приведена на рис. 4.12. Он содержит:

– порты для обмена данными и командами с центральным процессором посредством двух очередей соответственно для команд и данных, организованных по принципу *FIFO*;

– ПЗУ – постоянная память, содержит микропрограмму, по которой функционирует хост-контроллер в соответствии с командами процессора и драйвером *USB*;

– динамическую память, используемую как со стороны процессора, так и со стороны хост-контроллера. В ней размещены очереди *FIFO* данных и команд, переменные микропрограммы хоста и пакеты планируемых транзакций;

– интерфейс шины. Он преобразует поток данных от управляющей части хост-контроллера и к ней в соответствии с форматами данных *USB* и данных в том виде, в котором они представлены в хост-контроллере;

– процессорный блок хост-контроллера состоит из блоков выборка инструкций, АЛУ, блока регистров и интерфейса данных с памятью *RAM*. Все эти блоки вместе составляют процессор хост-контроллера. Он осуществляет функционирование остальных частей хоста и всей системы *USB* в целом;

– обработчик прерываний. В соответствии с поступающими прерываниями от нижнего уровня устройств иерархии *USB* обрабатывает их и вносит изменения в работу управляющей части контроллера;

– таймер для синхронизации и планирования обмена данными между системной шиной и хост-контроллером;

– непосредственно модуль *USB*. Здесь расположены передатчик и порты хост-контроллера. Здесь же задается мощность, подаваемая на шину.

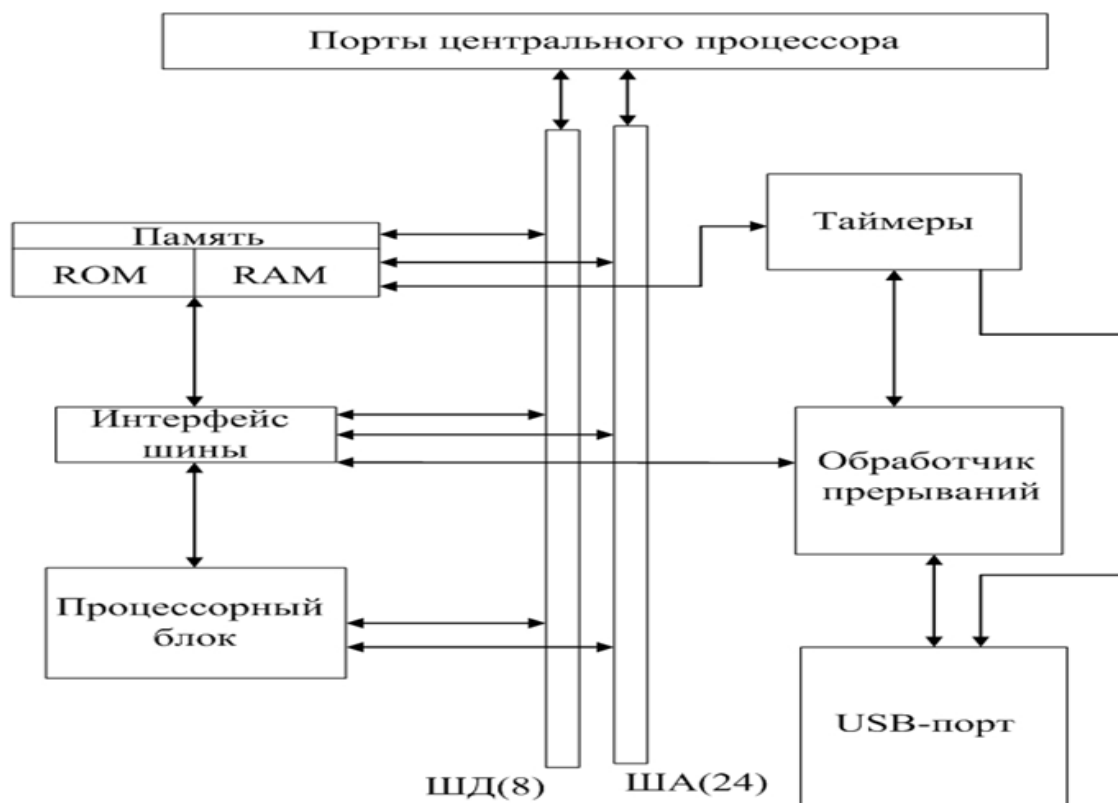


Рис. 4.12. Упрощенная структурная схема хост-контроллера *USB*

Применение шины

Шина *USB* может использоваться для подключения практически любых периферийных устройств от самых медленных (клавиатура, мышь) до самых быстрых (дисплей), а именно:

- устройства ввода клавиатуры, мыши;
- принтеры;
- сканеры;
- аудиоустройства;
- музыкальные синтезаторы и *MIDI*-контроллеры;
- видео- и фотокамеры;
- коммуникации;
- устройства хранения;
- игровые устройства;
- телефоны;
- мониторы;
- электронные ключи;
- преобразование интерфейсов;

– собственные устройства *USB*, в том числе на основе микроконтроллеров.

ПО по управлению шиной

Системное ПО контроллера управляет взаимодействием между устройствами и их ПО, функционирующим на хост-компьютере, для согласования:

- нумерации и конфигурации устройств;
- изохронных передач данных;
- асинхронных передач данных;
- управления энергопотреблением;
- информации об управлении устройствами и шиной.

4.2. Интерфейс *IEEE-1394 (FireWare)*

Другие названия: *iLink*, *HPSB-1995* и др.

Скорости: *S100*, *S200*, *S400*, 800, 1600 и 3200 Мб/с.

Основные характеристики шины можно свести к следующим показателям:

- скорость передачи данных до 3200 Мб/с;
- 16-разрядный адрес позволяет адресовать до 64К узлов на шине;
- предельная теоретическая длина шины 224 м;
- «горячее» подключение/отключение без потери данных;
- автоматическое конфигурирование, аналогичное *Plug&Play*;
- произвольная топология шины – по аналогии с локальными сетями может использоваться как «звезда» так и общая шина (только в виде цепочки в отличие от сети на коаксиальном кабеле);
- никакие терминаторы не требуются;
- возможность обмена с гарантированной пропускной способностью, что крайне необходимо для передачи видеоизображений.

4.2.1. Этапы развития и варианты

IEEE-1394a – 2000 г. – введено время ожидания 1/3 с на сброс шины, пока не закончится переходный процесс установки надежного

подсоединения или отсоединения устройства; добавлена информация о скорости в каждый пакет, если скорость отличается от предыдущей;
IEEE-1394b – 2002 г. – 800 и 1600 Мб/с, 3,2 Гбит/с.
IEEE-1394c – 2006 г. – оптоволокно.

4.2.2. Основные свойства

1. Многофункциональность, связь до 63 устройств без дополнительных хабов.

Могут быть устройства следующих 2 групп:

– устройства бытовой электроники (видео- и фотокамеры, камеры для видеоконференций, приемники кабельного и спутникового ТВ, цифровые видеоплееры *CD* и *DVD* форматов, акустические системы, цифровые музыкальные инструменты);

– периферийные устройства компьютера (принтеры, сканеры, диски, ПК между собой).

2. Высокая скорость обмена и возможность изохронных передач (можно передавать 2 канала видео по 30 кадров в секунду и стерео, аудио).

3. Низкая цена компонентов и кабеля.

4. Легкость установки и использования (*PnP*, горячее подключение, питание от шины с током до 1,5А).

4.2.3. Уровни описания

Стандартный интерфейс имеет три уровня описания (рис. 4.13):

1. Уровень транзакций (или взаимодействия) преобразует пакеты в данные и наоборот.

2. Уровень связи из данных физического уровня формирует пакеты и реализует обмен узлов дейтаграммами, отвечает за передачу пакетами и изохронные передачи.

3. Физический уровень вырабатывает и принимает сигналы шины. Обеспечивает инициализацию и арбитраж шины (в любой момент времени только один передатчик), уровни передачи данных и сигналы вышестоящим уровням. Микросхемы физического уровня могут питаться от шины, при этом обеспечивая гальваническую разрезку.

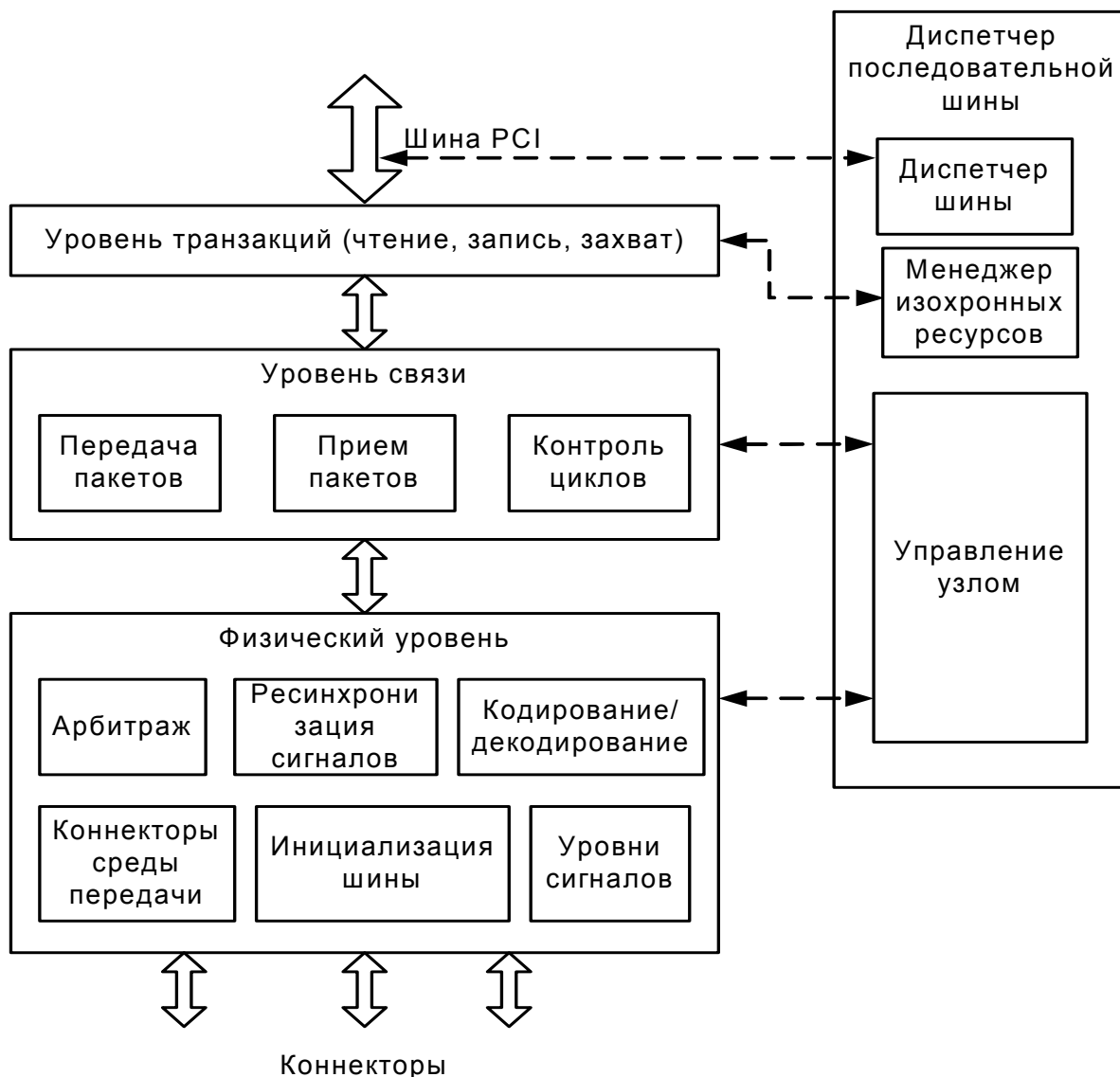


Рис. 4.13. Уровни описания интерфейса *IEEE-1394*

Топология шины

1. Между любой парой узлов может быть не более 16 кабельных сегментов (связей). Длина сегмента стандартного кабеля меньше 4,5 м или равна ей. Суммарная длина кабеля меньше 72 м.
2. Топология не должна иметь петель.
3. На одном устройстве может быть до 27 разъемов.

4.2.4. Кабели и разъемы

Стандартный кабель – это 6 проводов в общем экране, из которых:

- 2 витые пары для передачи сигналов *TPA*, *TPB*;
- 2 провода питания (U от 8 до 40 В, $I \leq 1,5$ А).

Разъемы могут быть 4- и 6-контактные.

В стандарте предусматривается гальваническая развязка, если с помощью трансформаторов, то до 500 А, если с помощью конденсаторов, то до 60 В.

В новой серии стандартов *IEEE-1394В* предусматриваются другие кабели: *UTP*, пластиковое оптоволокно *S200*, многомодовое волокно и др. Для соединения нескольких устройств между собой используются концентраторы и повторители (мосты).

Адресация происходит не по адресу устройства, а согласно страничной модели памяти. Адрес пакета имеет 64 бита (10 – адрес сети; 6 – адрес устройства; 48 – адрес устройства). *Протокол*: временной интервал разбивается на фреймы (кадры) по 125 мкс. В фрейме размещаются пакеты с данными.

4.2.5. Протоколы передачи

Два типа передач:

- изохронный;
- асинхронный.

Особенности фреймов: вначале данные изохронных каналов (до 64 каналов). Остальная часть отводится для других данных в асинхронном режиме. Пакет, который передается в синхронном режиме, состоит из адреса приемника, адреса передатчика и данных.

Изохронная передача передается широковещательно, и каждый пакет несет номер канала. Целостность изохронных передач контролируется *CRC*-кодом.

Изохронный канал – канал, обеспечивающий некоторую постоянную скорость передачи. Такие каналы нужны для передачи такой информации, которая воспринимается человеком (аудио-, видеоинформации).

Мастер цикла – это устройство, посылающее каждые 125 мкс короткие широковещательные пакеты начала цикла. В этом пакете передаются счетчики времени (32 разряда), инкрементируемые с частотой $f = 24,576$ МГц для любого узла изохронного обмена.

Диспетчер изохронных ресурсов

Выделяемая полоса измеряется в специальных единицах – интервалах-квадлетах длительностью $t = 20$ нс (1 квадлет равен 32-битовому слову, $f = 1600$ Мб/с). Всего 6144 ед. Для изохронных передач 4915 ед. (100 мкс), для асинхронных – 1229 ед. (25 мкс).

Для цифрового видео требуется полоса 30 Мб/с, из которых 25 Мб/с – видеоизображение и 3 – 4 Мб/с – аудиоданные.

В стандарте $S100$ – 1800 ед., $S200$ – 900 ед., $S400$ – 450 ед.

4.2.6. Устройства и адаптеры интерфейса IEEE-1394

Адаптер связывает мост с шиной *PCI*. Первоначально адаптер реализовался как совокупность двух микросхем: *LINK-chip* – микросхема уровня связи и *PHY-chip* – физический уровень.

В настоящее время адаптер выполняется в одной микросхеме или встраивается в южный мост.

Шина *IEEE 1394* может использоваться:

- с компьютерами;
- аудио- и видеомультимедийными устройствами;
- принтерами и сканерами;
- жесткими дисками, массивами *RAID*;
- цифровыми видеокамерами и видеомагнитофонами.

4.3. Интерфейс *Thunderbolt*

Интерфейс *Thunderbolt* был представлен фирмой *Intel* 24 февраля 2011 г. Новый интерфейс обеспечивает скорость передачи информации 10 Гбит/с. Он имеет обозначение, приведенное на рис. 4.14.



Рис. 4.14. Интерфейс *Thunderbolt* – удар молнии

Технически *Thunderbolt* представляет собой объединение шин *PCI Express* и *Display Port* в одном кабеле (рис. 4.15). Для подключе-

ния используются существующие кабели и коннекторы типоразмера *Mini DisplayPort*, а шина *Thunderbolt* способна запитывать устройства аналогично шине *USB*.

Первыми устройствами, оснащёнными новым интерфейсом, стали ноутбуки *Apple MacBook Pro 2011 г.*

Интерфейс *Thunderbolt* разработан компанией *Intel* совместно с *Apple*. На этапе разработки носил кодовое название *Light Peak*. На русский язык «*Thunderbolt*» переводится как «Удар молнии».

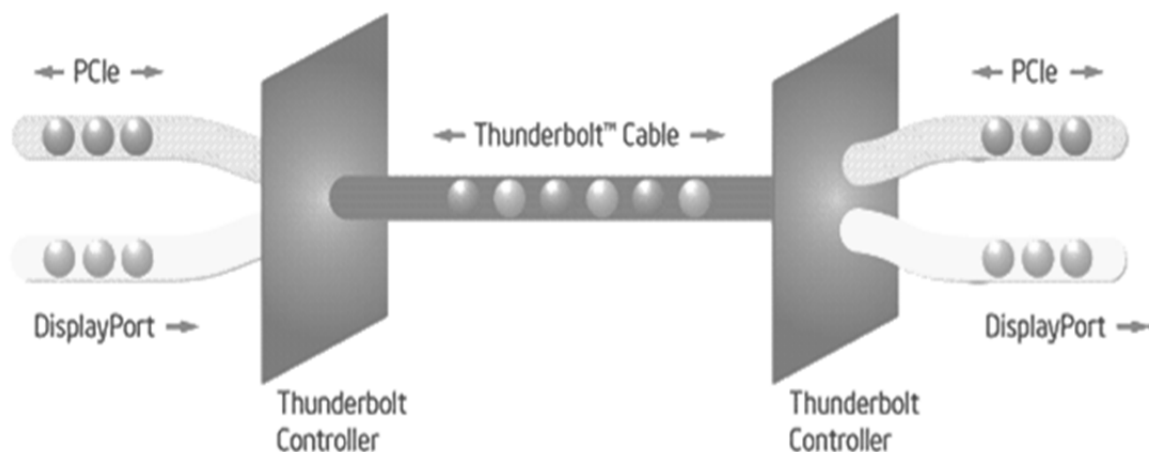


Рис. 4.15. Диаграмма работы контроллера *Thunderbolt*

Как заявляет *Intel*, *Thunderbolt* является самой высокоскоростной среди всех существующих на сегодняшний день технологий. Интерфейс поддерживает передачу данных на скорости 10 Гб/с. Так, к примеру, 30 Гб фильм с *HD*-видео по *Thunderbolt* может быть передан за 30 с. *Thunderbolt* в 2 раза быстрее *USB 3.0* – самого быстрого на сегодня интерфейса передачи информации.

4.3.1. Особенности технологии *Intel Thunderbolt*

Технология *Thunderbolt* разработана компанией *Intel* (кодовое название проекта – *Light Peak*) и представлена на рынок при технической поддержке *Apple*. *Thunderbolt* – это новая, высокоскоростная, двухпротокольная технология ввода-вывода, созданная с целью обеспечения максимальной производительности, гибкости и лёгкости в использовании.

Эта высокоскоростная технология передачи данных обеспечивает следующие возможности:

- два канала 10 Gbps (Гбит/с) на каждый порт;
- двунаправленный интерфейс;
- два-в-одном (Интерфейсы *PCI Express* и *DisplayPort* через один кабель);
- совместимость с существующими устройствами *DisplayPort*;
- последовательно-приоритетная цепочка устройств (*Daisy-chained devices*);
- медные или оптические кабели;
- низкие задержки и высокая точность синхронизации времени;
- использует собственные драйверы протоколов;
- питание устройств через шину по кабелям.

Технология *Thunderbolt* обрабатывает потоки данных в обоих направлениях одновременно, таким образом пользователи получают полную скорость в обоих направлениях по одному кабелю. Полная пропускная способность в 10 Гб/с при использовании двух независимых каналов доступна не только первому, но и последующим устройствам.

Контроллеры *Thunderbolt* от *Intel* соединяют ПК и другие устройства (рис. 4.16), принимая и передавая пакеты трафика обоих протоколов – и *PCIe*, и *DisplayPort*. *Intel* выпускает различные варианты контроллеров, различающиеся количеством каналов (1 – 4), выходов *DisplayPort* (0 – 2), линий *PCIe* (2 – 4) и мощностью цепи питания (0,7 – 4 Вт). В контроллере *Thunderbolt* установлены мультиплексор и демultipлексор, которые отвечают за передачу данных разных протоколов в едином потоке.

Все устройства *Thunderbolt* используют стандартный коннектор, позволяя подключать их друг за другом по цепочке при помощи совместимых кабелей.

К одному порту *Thunderbolt* возможно подключить до 6 устройств, одно из которых – монитор. Питание, доступное по шине *Thunderbolt*, ограничено 10 Вт.

Кабель интерфейса *Thunderbolt* представлен на рис. 4.17. Он имеет 4 витые дифференциальные пары для передачи информационных сигналов, 4 провода питания, внешние экран и оболочку и внутреннюю нейлоновую жилу для обеспечения механической прочности.

В настоящее время интерфейс *Thunderbolt* применяется в основном в компьютерах фирмы *Apple*: *MacBook Air*, *Mac mini* и др.

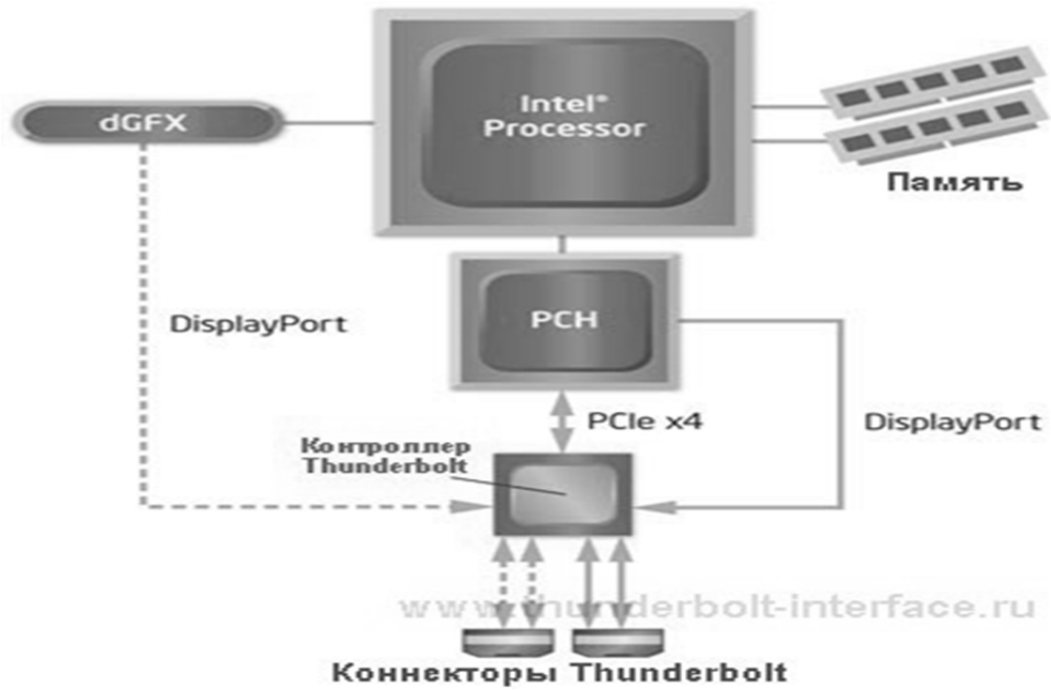


Рис. 4.16. Схема подключения контроллера *Thunderbolt*

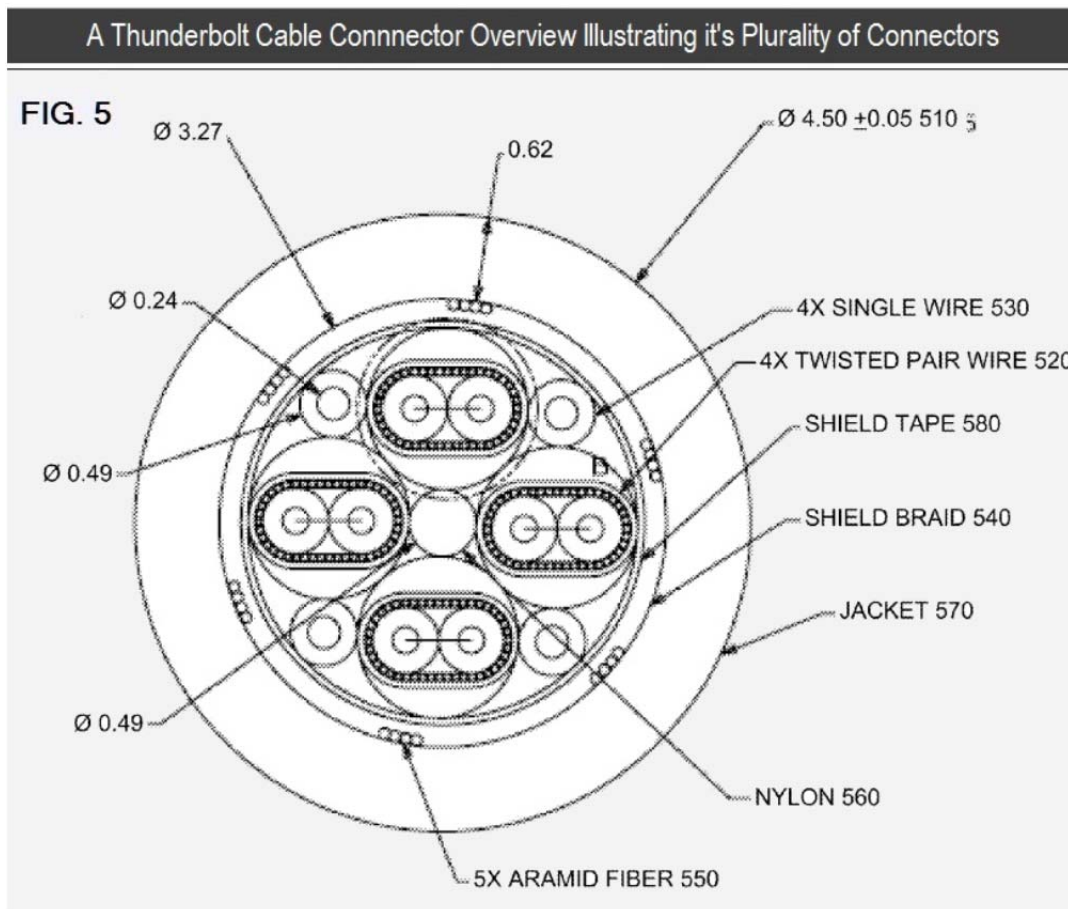


Рис. 4.17. Кабель интерфейса *Thunderbolt*

Несмотря на ограниченность применения, *Intel* представила следующее поколение – *Thunderbolt 2*, имеющий пропускную способность 20 Гбит/с в двухстороннем канале. *Intel* предполагает интегрировать его в будущие фирменные чипсеты. 21 апреля 2014 г. появилась информация о новом поколении интерфейса – *Thunderbolt 3*. Его скорость увеличена еще в 2 раза – до 40 Гбит/с, при этом снижено энергопотребление и уменьшен размер коннектора, а мощность питания увеличена до 100 Вт. Планируется представить контроллеры этого интерфейса не раньше выхода процессоров *Srylake* от *Intel*, который намечен на 2015 – 2016 гг.

4.4. Сравнение характеристик последовательных интерфейсов

Рассмотренные последовательные интерфейсы широкого применения имеют высокие скорости обмена, хотя и различаются по этому показателю (рис. 4.18). Наибольшей производительностью отличаются самый современный интерфейс *Thunderbolt* и вариант *USB 3.1*. Они позволяют подключать любые ПУ, в том числе дисплей и даже не один. Недостаток *Thunderbol* – его высокая стоимость и малая поддержка со стороны производителей аппаратуры. Лишь фирма *Apple* применяет его в своих продуктах.

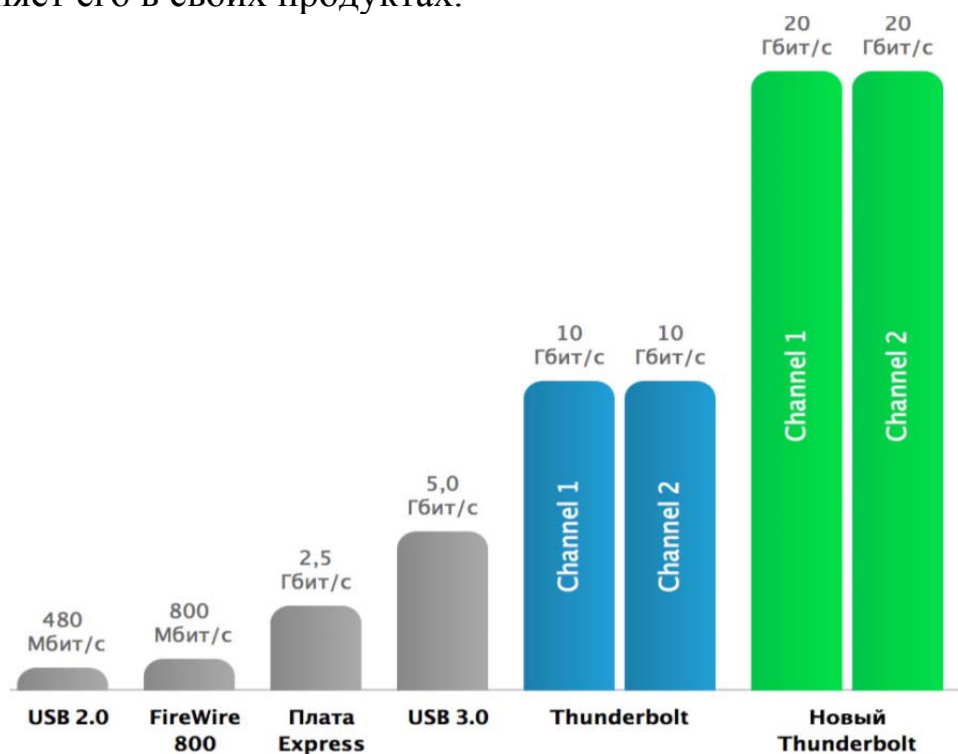


Рис. 4.18. Сравнение производительности последовательных интерфейсов

Наиболее распространенным считается интерфейс *USB*, имеющий многочисленные варианты как по производительности, так и по конструктивному исполнению. Эти особенности и широкая поддержка со стороны производителей и как следствие низкая стоимость делают его действительно универсальным. Он применяется в абсолютном большинстве компьютеров и во многих других электронных устройствах от сотовых телефонов до телевизоров.

Интерфейс *IEEE-1394* появился раньше других, но никогда широко не применялся для ПУ компьютеров. Лишь фирма *Apple* применяла его в своих продуктах для этих целей. В настоящее время и она отказалась от него, так как по производительности он отстает от *USB* и *Thunderbolt*. Основной областью применения этого интерфейса являются мультимедиа технологии. Он позволяет подключать цепочкой различные устройства без применения хост-компьютера и обеспечивать различные напряжения питания до 48 В с током до 1,5 А. Достоинством можно считать и большие расстояния (до 72 м), что позволяет обеспечивать работу сцен больших размеров.

Темы для закрепления знаний

1. *USB* – общие характеристики, особенности и преимущества. Применение шины.
2. *USB* – организация и топология шины. Структурные схемы подключения устройств.
3. *USB* – логическая организация обмена и временные диаграммы. Протокол обмена.
4. *USB* – уровни взаимодействия.
5. *USB* – виды и электрические параметры сигналов. Разъемы и кабели.
6. *USB* – типы и модель передачи данных.
7. *USB* – форматы пакетов.
8. *USB* – задачи синхронизации при изохронном обмене (частоты и типы отклонений).
9. *USB* – системное конфигурирование. Операции, поддерживаемые устройствами.

10. *USB* – функции и состояние хаба.
11. *USB* – задачи хоста и ПО по управлению шиной. Уровни хоста.
12. *IEEE 1394 (Fire Ware)*. Основные параметры, характеристики и свойства. Назначение.
13. *IEEE 1394 (Fire Ware)*. Трехуровневая структура.
14. *IEEE 1394 (Fire Ware)*. Физический уровень.
15. *IEEE 1394 (Fire Ware)*. Адресация. Виды передач и их структура (временные соотношения).
16. *IEEE 1394 (Fire Ware)*. Изохронные передачи.
17. *IEEE 1394 (Fire Ware)*. Конфигурирование. Устройства и адаптеры.
18. Интерфейс *Thunderbolt*.
19. Сравнение характеристик последовательных интерфейсов.

Глава 5. ИНТЕРФЕЙСЫ ВНЕШНИХ ЗАПОМИНАЮЩИХ УСТРОЙСТВ (ВЗУ)

5.1. Интерфейсы жестких дисков *IDE(ATA)*

История интерфейсов ВЗУ

Интерфейс *ST-506/412*. 1981г. – фирма *Seagate*. Имел ограниченную емкость $Q = 233$ МБ. Была поддержка *PnP*.

Интерфейс *ESDI*. 1983 – 84 гг. Модернизация – в 1991 г. Сократилось количество ошибок, увеличилась скорость, появилась возможность считывания карты дефектов.

Интерфейс *IDE (ATA)* – 1989 г. Высокое быстродействие, возможность подключения к различным системным шинам, дешева.

Интерфейс *IDE (ATA)*

Названия:

- *IDE – Intervel Drive Electronics*.
- *ATA – AT Attachment*.
- *PATA – Parallel ATA*.

IDE предназначен для подключения не более 2 дисков, но сейчас есть контроллеры 2 по 2. Контроллеры дисков размещаются на самом диске, а контроллер интерфейсов встраивается в чипсет.

Применяется модифицированная фазовая модуляция (*MFM*) и *RLL*-кодирование – кодирование большим количеством бит с меньшим количеством переключений.

История *IDE* началась с интерфейса *ST-506/412* в 1981г., фирма *Seagate*, и насчитывает более 10 модификаций.

1. *ATA (ATA BUS)* – шина 16 бит, 40-проводной сигнальный кабель + 4 провода (силовой кабель).
2. *PC Card ATA* – шина 16 бит, 68 проводов.
3. *XTADE* – 16 бит, 40 проводов.
4. *MCA IDE* – 16 бит, 72 провода – для интерфейса систем PC/2.
5. *ATA-2* – расширение 2 канала, 4 устройства, различные режимы.
6. *E-IDE* – аналог *ATA-2*.
7. *Fast ATA-2* – дополнительные возможности.
8. *ATA-3* – средства защиты, управление питанием.
9. *ATA/ATAPI-4* – расширение *ATA-3*. Режим *Ultra DMA*, повышенная скорость до 33 МБ/с за счет пакетного режима.
10. *ATA/ATAPI-5* – скорость 66 МБ/с.
11. *ATA/ATAPI-6* – потоковое расширение для аудио- и видеоданных, $U = 100$ МБ/с.
12. *Serial ATA (SATA)*.

5.1.1. Компоненты спецификации ATA

1. Хост-адаптер – средства сопряжения интерфейса *ATA* с шиной компьютера (*PCI* и *ISA*). Хост-контроллер – более развитый вариант хост-адаптера.

2. Ведущее (*master*) устройство – ПУ-0 (*Device 0*).

3. Ведомое (*slave*) устройство – ПУ-1 (*Device 1*).

Адресация дискового пространства:

– трехмерная (*CHS*). Такая адресация позволила адресовать до 137 ГБ (6553 бцилиндров, 16 головок, 255 секторов, 512 байт). Получим 137 ГБ;

– линейная адресация логических блоков (*LBA*). Адрес блока – 28-битное число.

В регистре *D/H* шестой бит *L* определяет вид адресации.

5.1.2. Физические параметры

Сигналы уровня ТТЛ:

$U_{\text{ВЫХ}}"1" \geq 2,4 \text{ В}$, $U_{\text{ВЫХ}}"0" \leq 0,5 \text{ В}$.

$U_{\text{ВХ}}"1" \geq 2,0 \text{ В}$, $U_{\text{ВХ}}"0" \leq 0,8 \text{ В}$.

Разъем.

40 контактов – *SFF* 8057.

4 контакта – min+12В.

2,3 контакта – +5В.

Для *UltraDMA* – 80-контактный разъем.

Для малогабаритных устройств – объединенный 50-контактный разъем *SFF*8212.

5.1.3. Регистры устройств в АТА

1. Блок командных регистров (*CBR*)

DR – Рг данных;

ER – Рг ошибки;

FR – Рг свойств;

SR – Рг счетчика секторов;

SN – Рг номер сектора;

CL – Рг младшего байта номер цилиндра;

CH – Рг старшего байта номер цилиндра;

D/H – Рг номер устройства и головки;

SR – Рг состояния;

CR – Рг команды.

2. Блок управляющих регистров

– *AS* – альтернативный Рг состояния;

– *DC* – Рг управления устройством;

– *DA* – Рг адреса.

5.1.4. Режимы передачи

Интерфейс *ATA* имеет множество режимов передачи, различающихся скоростями (см. таблицу).

Режим	<i>V</i> пер., МБ/с	Тип интерфейса
PIO mode 0 3 4	3,3 16,6	ATA ATA-2 FAST ATA-2
Single word DMA mode 0 1 2	2,08 8,33	ATA ATA ATA
Multi word mode 0 1 2	4,16 16,66	ATA-2 ATA-2 FAST AT-A2
Ultra DMA mode 0 1 2 3 4 5	16,6 100	ATA/ATAPI-4 ATA/ATAPI-6

5.1.5. Протокол передачи

1. Хост читает регистр состояния устройства, дожидаясь нулевого бита (*BSY*-бит в регистре состояния). Если два устройства подсоединились, то к обоим обращается. Состояние будет передаваться последнему выбранному устройству.

2. Дождавшись освобождения устройства, хост записывает в регистр *D/H* байт, у которого есть бит *DEV*, и он указывает на адресуемое устройство. Если *DEV* = 0, то это ведущее устройство, если *DEV* = 1, то это ведомое.

3. Хост читает основной или альтернативный регистр состояния адресуемого устройства. *DRDY* = 1 – признак готовности.

4. Хост заносит требуемые параметры обмена в блок командных регистров.

5. Хост записывает код выполненной команды в регистр команд.

6. Устройство устанавливает бит $BSY = 1$ и переходит к выполнению команды.

Интерфейс *ATA* имеет множество режимов передачи, различающихся скоростями.

5.1.6. Адаптеры и контроллеры шины IDE

– Простейшие адаптеры, подключаемые к *PCI* и *ISA*.

– Двухканальные контроллеры *PCI-IDE*.

Конфигурирование устройств

1. Выбор типа интерфейса: ХТ или АТ.

2. Определение адреса устройства:

– с помощью кабельной выборки;

– явное задание адреса на каждом устройстве джамперами.

5.2. Интерфейс *SerialATA*

Интерфейс *SerialATA* – последовательный, протокол которого совместим с интерфейсом *ATA IDE*. *SATA-1*; предложен в 2000 г., *SATA-2* – с большей скоростью и средствами поддержки сетевых дисков.

Работа над *SerialATA* была начата еще 8 лет назад. Версия 0.9 спецификации была представлена на *Intel Developer Forum* в 2000 г., и окончательно доработанная 1.0 – в конце того же года. *Serial ATA* – высокоскоростной последовательный интерфейс, предназначенный для устройств хранения информации.

В июле 2008 г. *SATA-IO* представила спецификации *SATA 3.0* с пропускной способностью 6 Гбит/с. Полный стандарт был выпущен в мае 2009 г. Эффективная пропускная способность составила 600 Мб/с, а частота функционирования 6,0 ГГц. Совместимость сохранилась как в методе передачи данных, так и в разъёмах и проводах; улучшено управление питанием.

Основной сферой применения, где требовалась такая пропускная способность, стали *SSD* (твёрдотельные) накопители. Выигрыш для них был в более высокой скорости передачи данных из кэш (*DRAM-cache*) памяти диска.

eSATA (External SATA) — интерфейс подключения внешних устройств, поддерживающий режим «горячей замены» (*Hot-swap*). Был создан несколько позже *SATA* (в середине 2004 г.). Это портативная разновидность интерфейса *SATA*, скорость передачи которого выше, чем у *USB 2.0* и *IEEE 1394*.

Основные изменения в сравнении с *SATA*:

- разъемы экранированы и более стойкие для многократного подключения;
- изменена компенсация потерь сигналов, что позволило увеличить максимальную длину кабеля до 2 м.

Требует подключения 2 разъемов: один питания, второй интерфейсный.

Особенности *SATA*:

1. Скорость передачи от 150 до 600 Мб/с.
2. Поддержка «горячей» замены.
3. Два режима энергосбережения.
4. Перекрытие инструкцией.
5. Очередь тегированных команд.
6. Шлейф с 7 проводами шириной 6 мм и длиной 1 м.
7. Количество дисков – до 128.
8. Мощность потребления – 250 мВт.

Часто среди обоснований перехода на новый стандарт называют ограниченную скорость передачи параллельного интерфейса в 133 Мбайт/с, но это ограничение конкретной его версии, а не его вида вообще (а у *Serial ATA* не намного и больше – 150 Мбайт/с).

Преимущества *SerialATA*:

- каждое устройство получает монопольный канал связи с контроллером, что позволяет повысить скорость;
- исключаются ненужные взаимодействия ведомого и ведущего устройств на уровне протокола, т.е. улучшается совместимость;
- реализуется эффективная одновременная работа контроллера с несколькими устройствами с поддержкой очередей;
- упрощается конфигурирование устройств;
- упрощаются кабели и разъемы;
- обеспечивается «горячая» замена;
- улучшаются условия охлаждения устройств.

По внутренней организационной структуре – четырехуровневой модели (рис. 5.1) – он очень похож на другие стандарты, например *USB* или *SCSI-3*, которые, в свою очередь, всегда напоминали модель *OSI* сетей передачи данных.

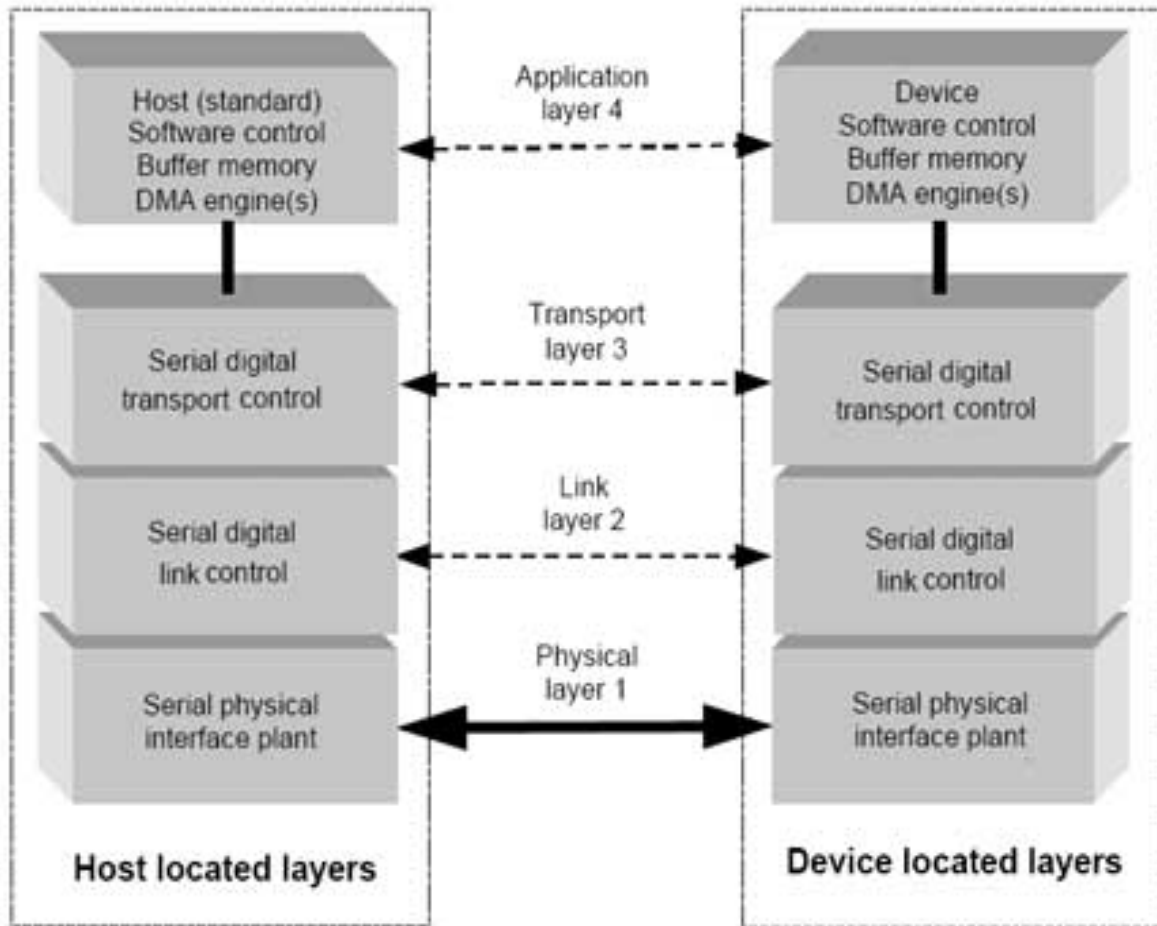


Рис. 5.1. Уровни взаимодействия устройства *SATA*

Использование многоуровневой модели дает возможность для гибкого дальнейшего совершенствования стандарта, доработкой только необходимых уровней и не правя весь стандарт сразу.

Для компьютера, в котором установлен *Serial ATA*, принципиально ничего не меняется, взаимодействие операционной системы через драйвер контроллера в целях обеспечения совместимости с предыдущими версиями *ATA* оставлено прежним.

Интерфейс между контроллером нового стандарта и драйвером (левая часть блока «контроллер») остался прежним: *Serial ATA* полностью эмулирует *Parallel ATA*, использует для обмена те же ре-

гистры и команды. Изменилось лишь взаимодействие между ним и непосредственно дисками. Сразу же обращает на себя внимание то, что для подключения тех же двух устройств контроллер уже имеет два порта (рис. 5.2) и каждый накопитель подключается отдельным кабелем.

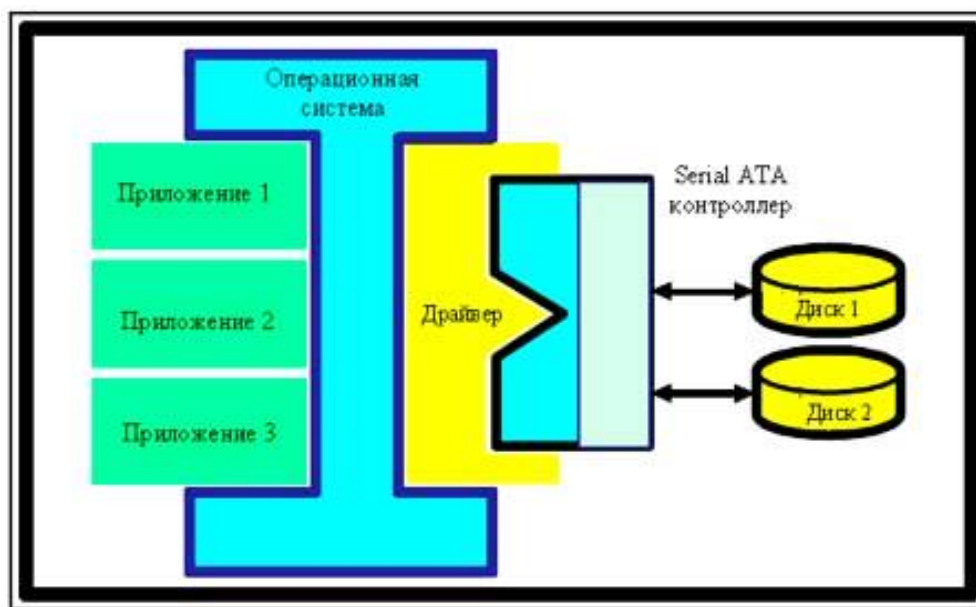


Рис. 5.2. Принцип подключения накопителей

В *Serial ATA* нет понятия ведомого и ведущего устройства (*Master/Slave*). Наконец-то пришла пора прекратить тягать перемычки из своих винчестеров. Все подключенные накопители являются независимыми устройствами, и их работа выглядит с точки зрения старого стандарта как *Master with no Slave Present* (но не нужно бояться вероятных проблем со старым ПО в этом месте: из соображений совместимости предусмотрена возможность эмуляции работы устройств в режиме *Master+Slave*: в зависимости от бита, используемого для определения устройства назначения на шине, обращения просто распределяются между портами *SATA*, хотя в этом режиме оказываются недоступными многие из возможностей новой шины).

Физический уровень занимается передачей битов по физическим каналам связи. Здесь определяются основные характеристики среды, используемой для передачи данных, и характеристики электрических сигналов.

Сигналы. Мы уже вскользь касались этого вопроса. При современных технологиях использование 5-вольтовых сигналов стало очень затруднительно, и, кроме того, с ростом скорости работы возникают дополнительные сложности при переключении из одного состояния в другое. С такой проблемой уже однажды столкнулась *SCSI*, и сейчас в *Serial ATA* был использован тот же подход. Уровень сигналов снижен и составляет 250 мВ.

Способ передачи. Кроме того, вместо использовавшейся ранее в *ATA* однополярной передачи, обладающей низкой помехоустойчивостью, применена двухполярная (или еще ее называют дифференциальной так же, как и *SCSI*). Преимущество ее в гораздо большей помехозащищенности. При дифференциальной передаче по двум проводам передается один и тот же сигнал, но разной полярности. Шумы, наводимые в проводах, симметричны, и, сложив оба полученных разнополярных сигнала, можно получить шум, а вычав его из полученного сигнала, – непосредственно чистый переданный сигнал. Собственно использование дифференциальной передачи и дало возможность снизить уровни используемого сигнала. Для кодирования передаваемой информации используется потенциальный код без возвращения к нулю (*Non Return to Zero, NRZ*). Он является одним из самых простых в реализации, благодаря двум резко различающимся потенциалам обладает хорошей распознаваемостью ошибок, но не обладает необходимым свойством самосинхронизации. Но с этим недостатком в *SATA* успешно борются гениально простым методом, о котором скажем ниже.

Физическая среда. Используется последовательная шина, состоящая из двух пар проводов (одной на передачу и одной на прием) и нескольких нулевых, всего семь, представленных на рис. 5.3. Провод, которым соединяются устройства последовательной *ATA*, становится таким образом тонким и круглым, гибким и удобным в использовании, не препятствует воздухообмену. Задача согласования (терминирования) решается теперь намного дешевле, а длина кабелей может достигать 1 м. *SATA* может быть не только интерфейсом внутренних устройств хранения, но и внешних.

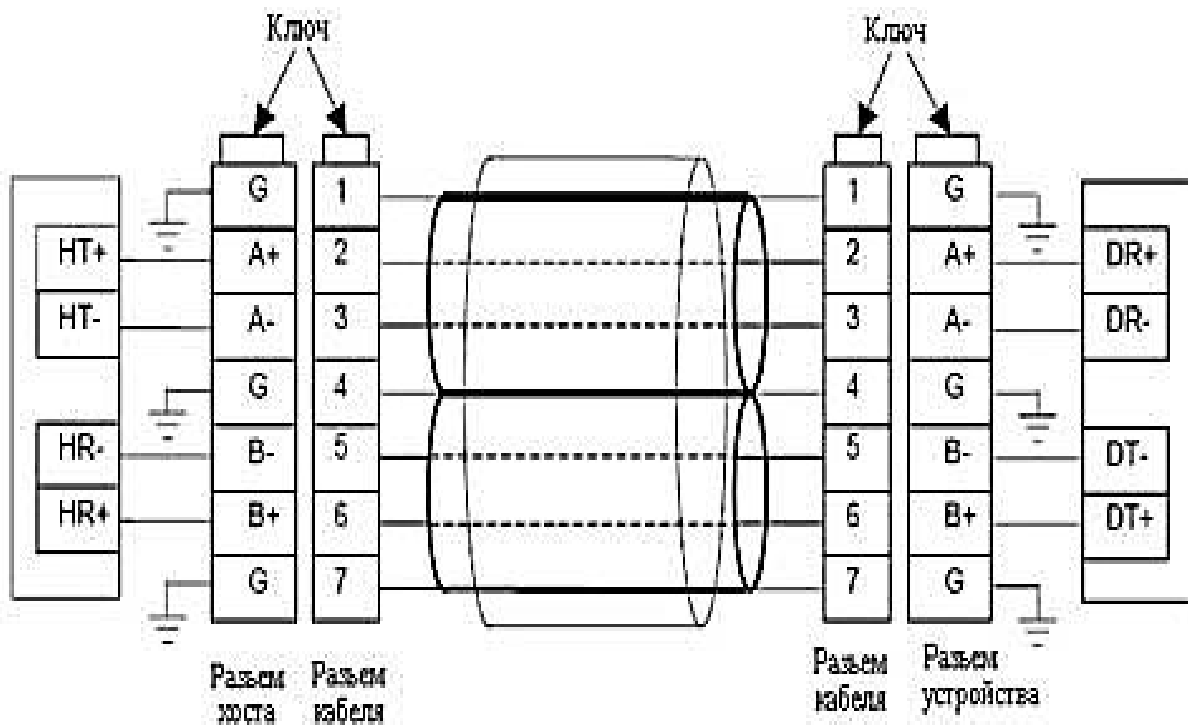


Рис. 5.3. Структура шины

Задача согласования (терминирования) решается теперь намного дешевле. А длина кабелей может достигать 1 метра. *SATA* может быть не только интерфейсом внутренних устройств хранения, но и внешних.

Разъемы. Так как проводов мало, то разъемы соответственно получаются очень компактными и удобными в использовании. На рис. 5.4 для сравнения представлены коннекторы традиционного параллельного и нового последовательного интерфейса (слева).

Разъем питания по размерам теперь превосходит разъем *data*-кабеля. Конструктивно они оба выполнены с защитой. Шире разъем питания потому, что контакты в нем больше и надежнее. Их не 4, как в старом разъеме, добавилась возможность использовать питание 3,3 В.

Спецификация *SATA* жестко не ограничивает размещение основных разъемов и выполнение дополнительных и предлагает несколько вариантов (рис. 5.5). Один – родной, а второй, как всегда, для поддержки предыдущих версий.

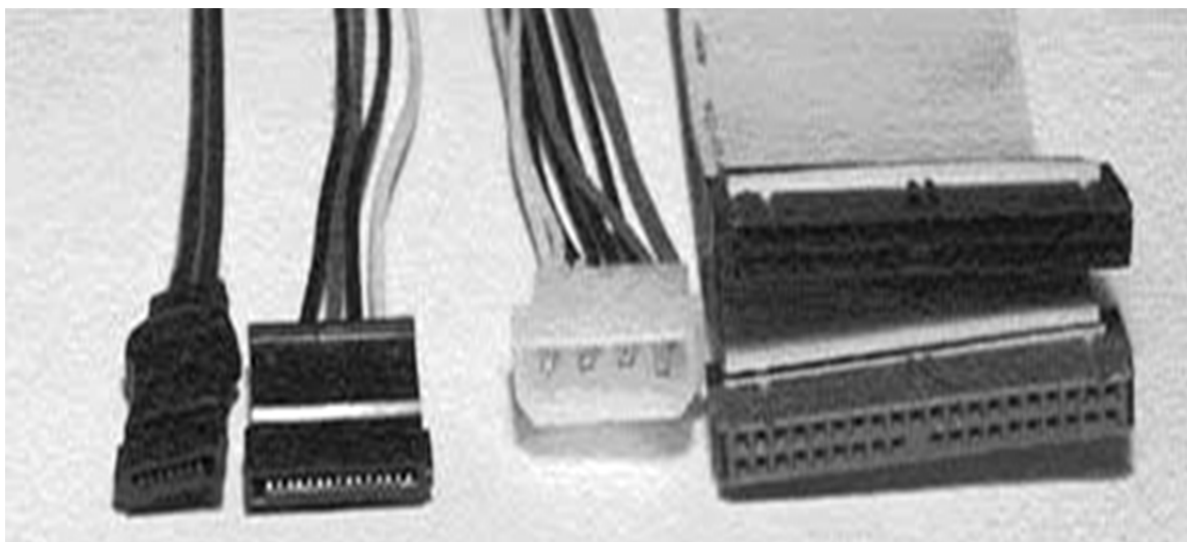


Рис. 5.4. Разъемы *SATA* (слева) и *PATA* (справа)

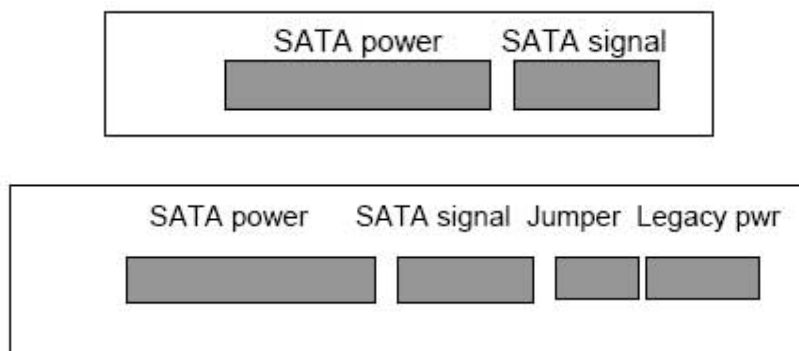


Рис. 5.5. Варианты разъемов *SATA*

Физический уровень осуществляет над поступившим кадром необходимые преобразования – конвертирует в последовательность, кодирует и выдает в линию и то же самое в обратном порядке, когда получает данные с физической шины, т.е. от другого устройства. Взаимодействие физического уровня с вышележащим происходит по параллельной шине шириной 10, 20, 40 бит или другой на усмотрение разработчика.

Канальный уровень выполняет функции арбитража и результата выполнения операций передачи данных, скремблирование, а также реализует механизмы обнаружения и коррекции ошибок.

Арбитраж и контроль передачи. Сам по себе физический уровень может только передавать данные, поступившие на его вход. Для предотвращения конфликтов, когда одновременно и устройство, и хост-контроллер хотят передать данные, на канальном уровне преду-

смотрен специальный механизм контроля, получивший название арбитража. Кроме того, в обязанности канального уровня входит слежение за тем, были ли данные успешно переданы, и сообщение об этом вышестоящему транспортному уровню.

Обнаружение и коррекция ошибок. В отличие от стандарта *ATA*, который различными механизмами обнаружения и коррекции ошибок обрастал по мере роста быстродействия, в *Serial ATA* несколько механизмов заложены изначально. Во-первых, хорошей распознаваемостью обладает используемый на физическом уровне *NRZ*. Но это не главное, ряд ошибок может успешно его миновать. Как метод применяется избыточное кодирование 8 бит/10 бит. Суть его проста: 8 бит исходных данных дополняются 2 дополнительными битами. Итого получается 10 бит, т.е. 1024 возможных битовых комбинаций может быть в получившемся коде, в то время как в исходном только 256.

Из результирующего кода отбирают 256 комбинаций, которые будут соответствовать 256 комбинациям исходного кода, а остальные считают запрещенными. Это позволяет распознавать искажение данных, если принята запрещенная последовательность, то при передаче произошла ошибка. Кроме того, в *Serial ATA* используется *CRC*-код. Кстати, то что жесткие диски с *SATA* имеют максимальную скорость обмена по интерфейсу в 150 Мбайт/с при том, что для *SATA* заявлена скорость передачи на физическом уровне в 1,5 Гбит/с, объясняется использованием избыточного 8 бит /10 бит кодирования, снижающего полезную пропускную способность интерфейса до 1,2 Гбит/с.

Скремблирование. Код *NRZ*, используемый на физическом уровне, не обладает свойством самосинхронизации, так как при последовательности нулей или единиц сигнал в линии просто превращается в постоянный сигнал определенного уровня. Скремблирование помогает бороться с этим явлением, перемешивая данные, подлежащие передаче, определенным образом так, чтобы вероятность появления единиц и нулей на выходе была приблизительно одинаковой.

Работает канальный уровень так: получает информационный кадр от транспортного, выполняя логическое кодирование и вычисление *CRC*, и спускает вниз к физическому уровню. При получении данных от физического уровня порядок действия обратный.

Задачей **транспортного уровня** является обеспечение вышележащим протоколам передачи с той степенью надежности, которая им

требуется. Он упаковывает поступившие от прикладного уровня *ATA* команды в кадры и передает их следующему или распаковывает поступившие снизу данные и передает на прикладной уровень. Обмен между контроллером и устройством на этом уровне осуществляется следующими информационными структурами:

- передачи регистров;
- установки битов;
- *DMA Activate*;
- *DMA Setup*;
- *PIO Setup*;
- данных – *FIS Data*;
- *BIST Activate*.

Задачей **прикладного уровня** считается организация взаимодействия между драйвером контроллера и всего программного обеспечения, стоящего за ним, и самим контроллером через блок регистров. Новый блок *Рг SCR (SATA Status and Control Rg)* имеет 16 32-разрядных *Рг SCR0 – SCR15* (используются 3, остальные – резерв):

- SCR0 – Sstatus* - *Рг* текущего состояния хост-адаптера;
- SCR1 – SError* - *Рг* диагностической информации;
- SCR2 – Scontrol* – *Рг* управления интерфейсом.

Адаптеры и контроллеры *SATA*:

- микросхема *Intel 31244*;
- контроллер *SATA* с интерфейсом *AHCI (Advanced Host Controller Interface)* – поддерживает 1 – 32 порта *SATA*.

5.3. Интерфейс *SCSI*

SCSI (Small Computer System Interface) разрабатывался как универсальный параллельный интерфейс для подключения различных устройств. В процессе развития он претерпел многочисленные изменения и имеет множество вариантов. У этого параллельного интерфейса высокая скорость протокола. Все устройства на шине могут быть инициализаторами или целевыми устройствами.

Достоинства:

- возможность подключения семи устройств к одному компьютеру;
- возможность подключения широкого спектра устройств;
- возможность подключения как внутренних, так и внешних устройств;

- большая длина кабеля – 3 – 6 м, а в некоторых случаях до 25 м;
 - могут использоваться кэширование и технологии *RAID*-массивов;
 - скорости при 8-битной шине 5 МГц = 5Мб/с (это минимум).
- Недостаток – высокая стоимость.

5.3.1. Варианты стандартов SCSI

- *SPI* – 1995 г. *FastSCSI* – 20 МБ/с, *WideSCSI* – 40 МБ/с (узкий).
- *SPI 2* – 1999 г. Удвоенная частота передачи. $U = 80$ МБ/с. Допускается «горячая» замена. 63-разъемный кабель.
- *SPI 3* – 2000 г. Двойная синхронизация (по переднему и заднему фронтам). *Fast80DT* или *Ultra 3 SCSI* – широкая шина – 16 бит.
- *SPI 4* – 2001 г. Удвоение частоты, *Ultra 320 SCSI*. $U = 320$ МБ/с. Только широкая шина – 16 бит.

5.3.2. Архитектурная модель SCSI (*SPI 3*)

Модель имеет следующие уровни (рис. 5.6):

- общие команды;
- наборы команд;
- транспортные протоколы;
- физические соединения.

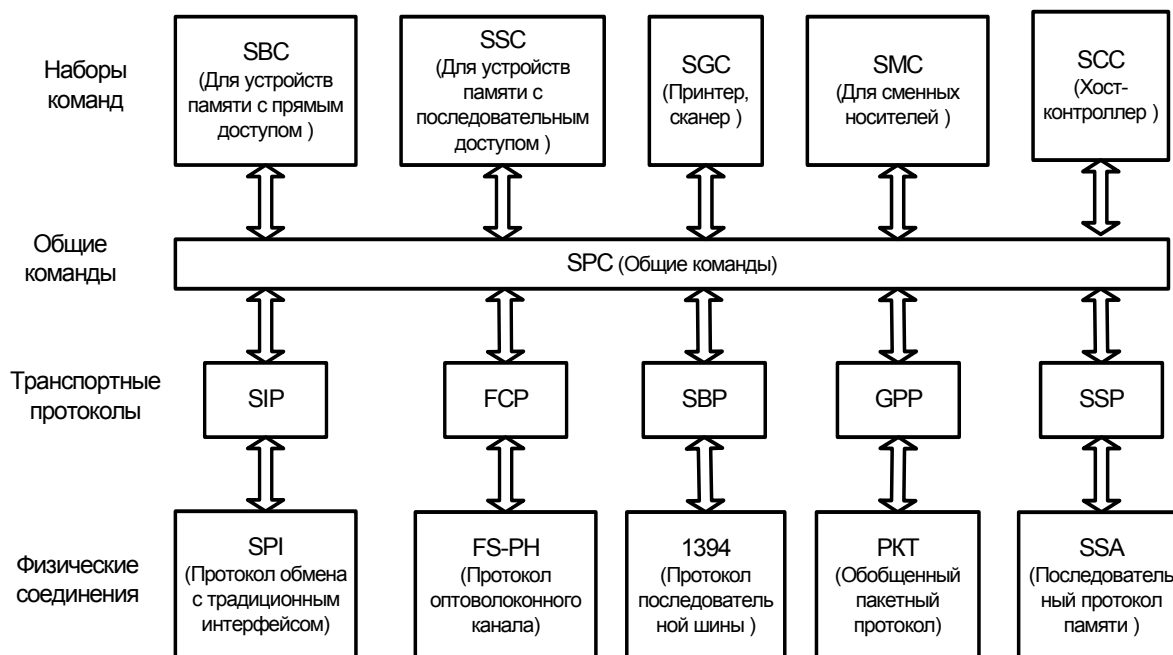


Рис. 5.6. Уровни SCSI

5.3.3. Физический интерфейс

Сигналы

Линейные – каждый сигнал с *TTL*-уровнем. У каждой пары свой отдельный провод. Длина:

- 6 м (обычный);
- 3 м (*Fast*, средняя скорость) – 4 устройства;
- 1,5 м (*Ultra*, высокая скорость) – 8 устройств.

Передатчики могут быть с открытым коллектором с активным снятием сигнала.

Дифференциальные – для каждой линии пара проводов, по которой передается парафазный сигнал, т.е. тоже *TTL*-уровень +*RS 485*. При этом длина кабеля может быть увеличена. Используется в основном в серверах.

Низковольтный дифференциальный сигнал *LVD* 40, 80, 60, 120 МГц. Длина 25 м (8 устройств), 12,5 м (16 устройств).

Последовательный вариант *SAS* может использовать витую пару или оптоволоконный кабель.

Обозначения интерфейсов в зависимости от уровней сигналов представлены на рис. 5.7.

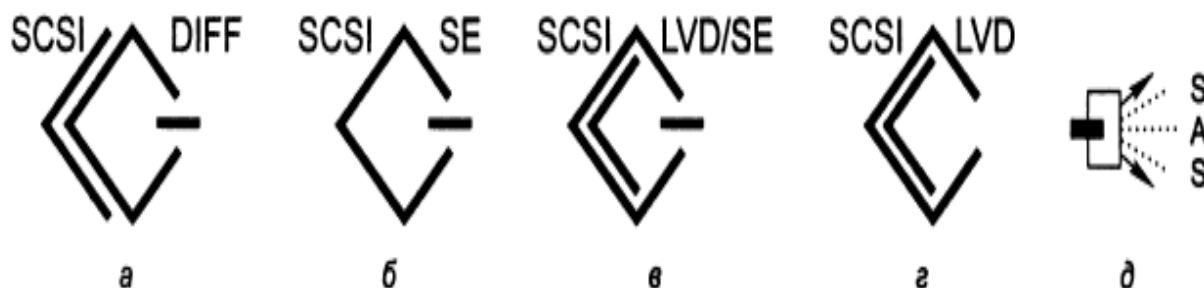


Рис. 5.7. Виды электрических сигналов:
а – HVD; б – SE; в – *LVD/SE*; г – *LVD*; д – *SAS*

На рис. 5.8 и 5.9 представлены диаграмма напряжений для сигналов *SE* и *LVD* и преобразование сигнала *LVD* при обработке на приемной стороне.

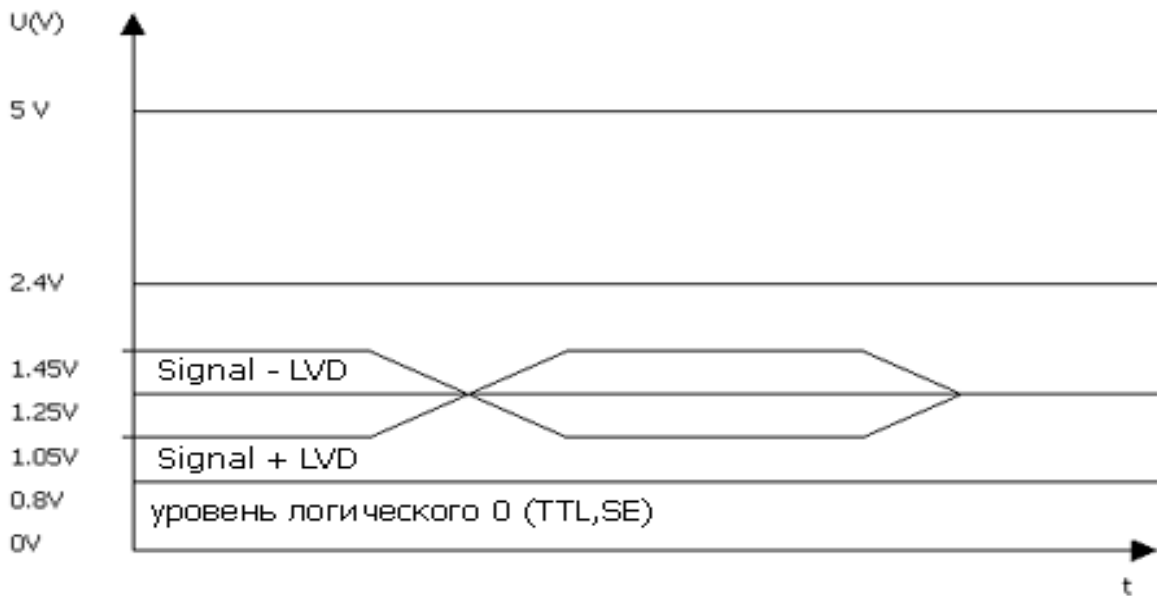


Рис. 5.8. Диаграмма напряжений для сигналов *SE* и *LVD*

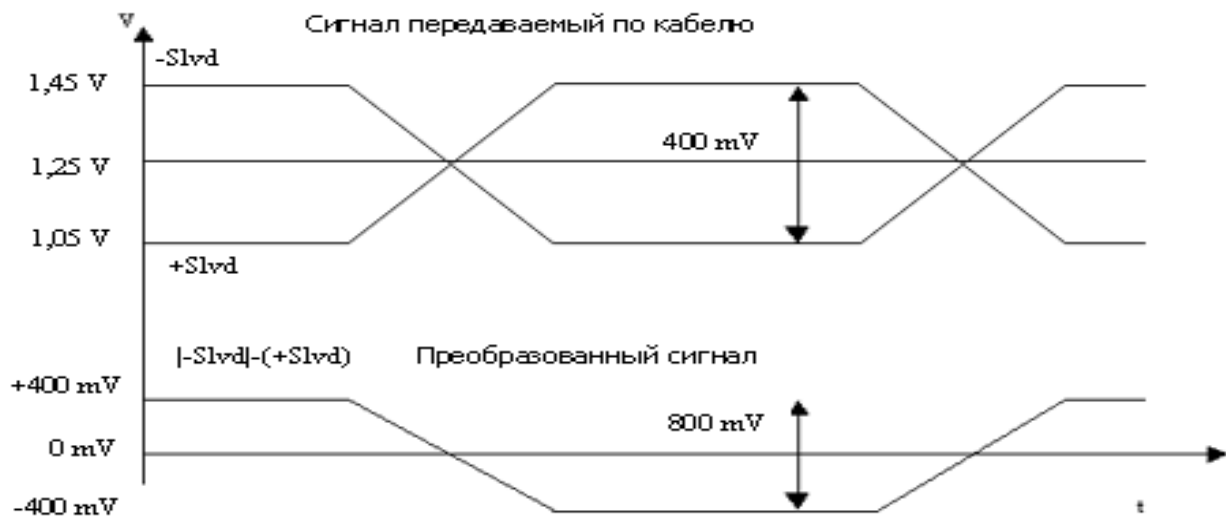


Рис. 5.9. Преобразование сигнала *LVD*

Кабели:

Плоский (для устройств в одном корпусе);

Круглый (для подключения внешних устройств);

A-кабель (для 8-битного интерфейса – 25 пар);

B-кабель (малораспространен, для 16-, 32-битного интерфейса);

P-кабель (для 8, 16-битного интерфейса – 34 пары);

Q-кабель (расширенный *P*-кабель для 32-битного интерфейса);

MacSCSI – 8 бит специально для *Macintosh*.

Разъемы

Narrow (узкий) – 8 бит:

1. *IDC-50* (для внутреннего соединения без корпуса);
2. *CX-50* (типа *Centronics*);
3. *LB-25* (круглые штырьки, как в LPT-порте);
4. *HD-50 MinuD50* (с высокой плотностью контакта).

Для 16-битной *SCSI*:

1. *HD-68 MinuD68* (наиболее часто используется для *SCSI-3*, внутренний монтаж);
2. *VHDCI 68* – внешний разъем с высокой плотностью в стиле *Centronics*.

Для «горячей» замены:

1. *MCX-68, 80* – в стиле *Centronics*, в миниатюрном исполнении;
2. *SCA-80* или *SCA-2*, или *MCX-80*.

Адресация устройств

Производится через идентификатор *SCSIID*, представляющий адрес в позиционном виде (позиции бита). Любое устройство должно иметь уникальный адрес.

Диапазон значений адресов:

- для 8-битной шины: 0 – 7;
- для 16-битной шины: 0 – 15.

На устройстве адрес задается предварительной перестановкой джамперов. Адрес определяется номером шины данных, с которой осуществляется выборка устройства при низком уровне.

Временные диаграммы асинхронного обмена представлены на рис. 5.10.

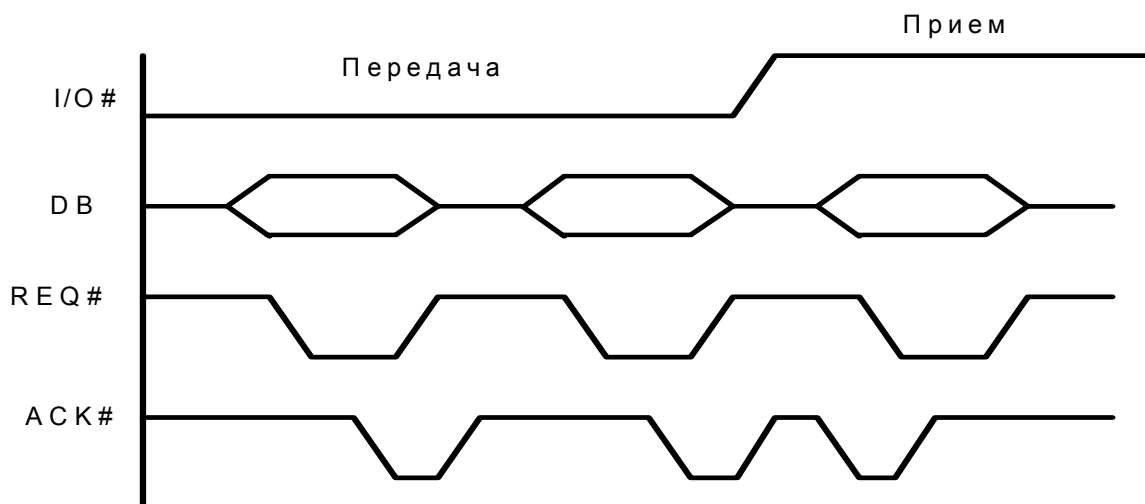


Рис. 5.10. Временные диаграммы асинхронного режима

5.3.4. Контроллеры SCSI

Хост-адаптеры выпускаются для шины *ISA*, *EISA*, *PCI*, *VLB*, *PCCMCIA*, *PCI-E* и для параллельного порта. Схема включения контроллера *SCSI* приведена на рис. 5.11, а внутренняя структура контроллера *DP5380* – на рис. 5.12.

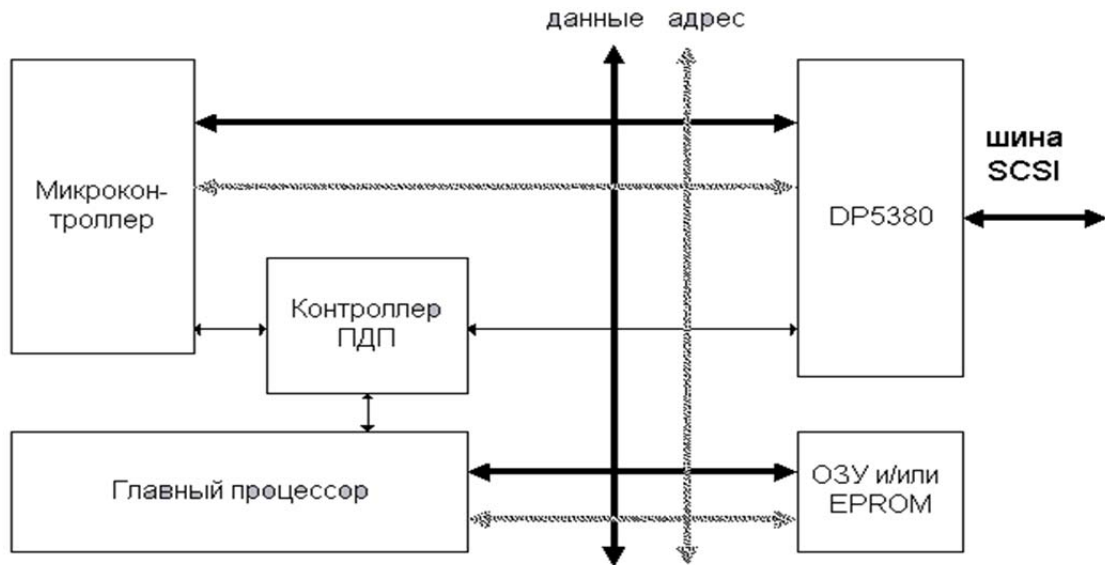


Рис. 5.11. Схема включения контроллера *SCSI*

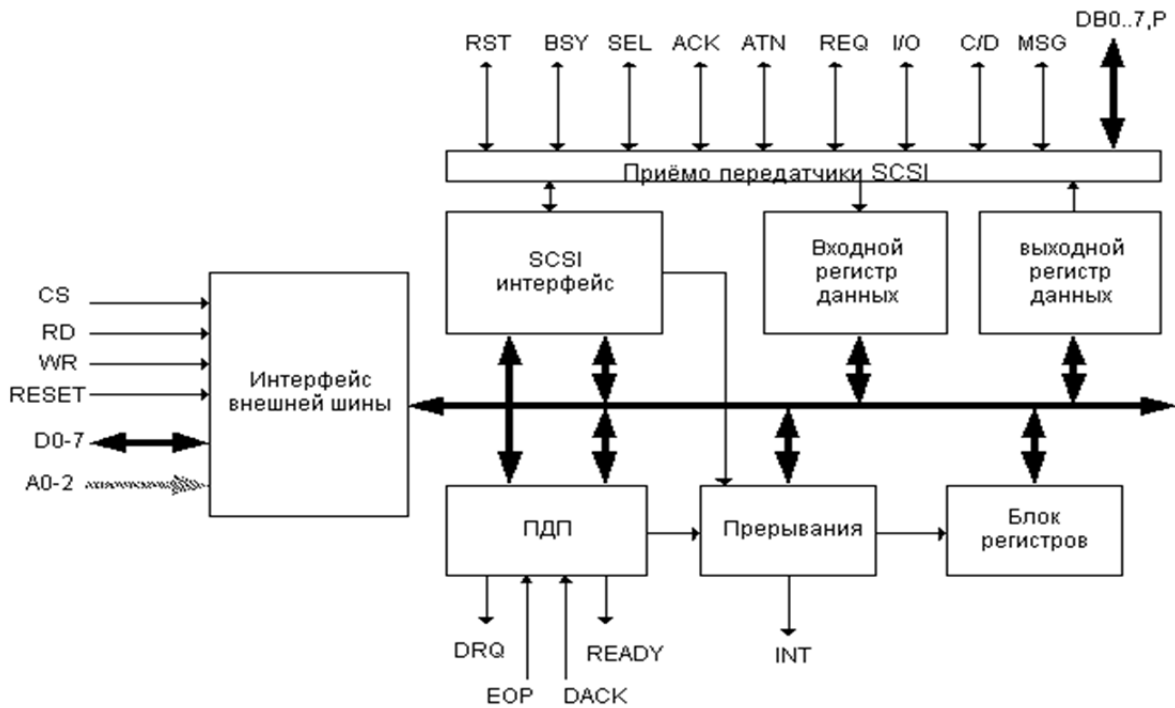


Рис. 5.12. Блок-схема контроллера *DP5380*

Интерфейс *SCSI* является универсальным и позволяет подключать различные виды периферийного оборудования:

- все виды дисков (CD, DVD, магнитные, оптические);
- сканеры;
- подключаемые жесткие диски;
- принтеры;
- последовательный порт.

Однако основным назначением интерфейса *SCSI* является формирование дисковых массивов в серверах. До появления интерфейса *SATA* для дисковых массивов мог использоваться только *SCSI*.

Подключение ПУ осуществляется с использованием терминаторов. Терминаторы могут быть в положении "ON" – включен и "OFF" – отключен. Для промежуточных устройств терминаторы отключаются, а на конечных – включаются, что показано на рис. 5.13.

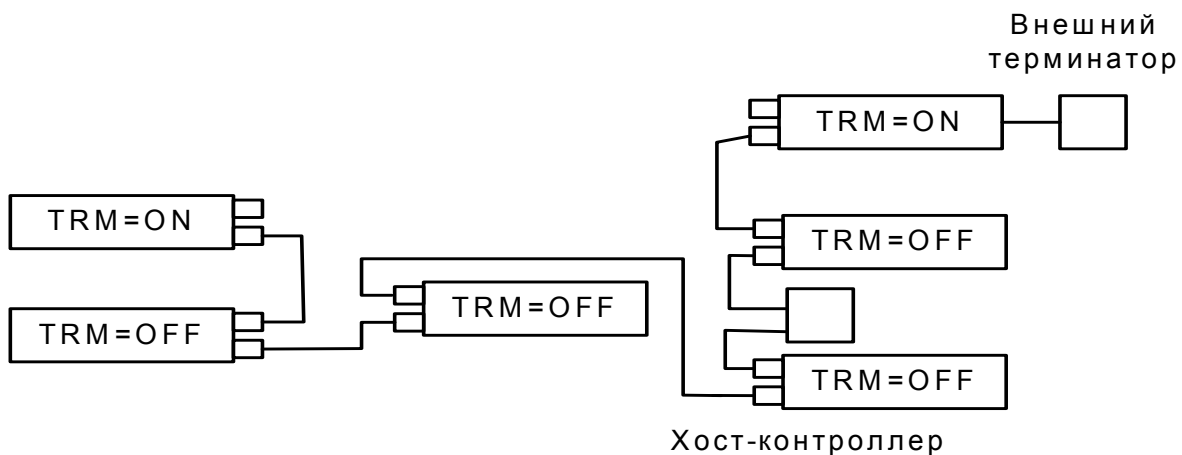


Рис. 5.13. Подключение терминаторов

BIOS

Различные варианты подключения:

- микросхема со *SCSI-BIOS* есть на системном контроллере;
- образ *SCSI-BIOS* прошит во *Flash* материнской платы;
- *SCSI-BIOS* нет вообще.

Дополнительные элементы:

- светодиоды;
- разъем для модулей памяти;
- контроллер гибких дисков;
- *IDE*-контроллер;

– могут быть аудио- и видеоплаты.

Конфигурирование *SCSI*

Устанавливаются параметры:

- идентификатор устройства (адрес);
- контроль паритета (по четности);
- включение активных терминаторов (для крайних устройств);
- питание терминаторов;
- согласование скоростей синхробомена;
- старт по команде или задержанный старт;
- разрешение отключения (автоматическое отключение при неготовности данных).

5.3.5. Устройства *SCSI* с последовательным интерфейсом *SAS*

В настоящее время наиболее часто используется вариант интерфейса с последовательным соединением. Последовательный интерфейс для подключения устройств *SCSI* (*Serial Attached SCSI, SAS*) разработан на основе физического интерфейса *Serial ATA*. Однако устройства *SAS* используют свой транспортный протокол (*SSP*), который отвечает общей идеологии *SCSI*. Для подключения устройств *SAS* к компьютеру служит хост-адаптер (*HBA*) с интерфейсом *SAS*, к которому можно подключать как устройства *SAS*, так и устройства *SATA* – их физические интерфейсы совместимы. Возможные варианты подключения устройств приведены на рис. 5.14.

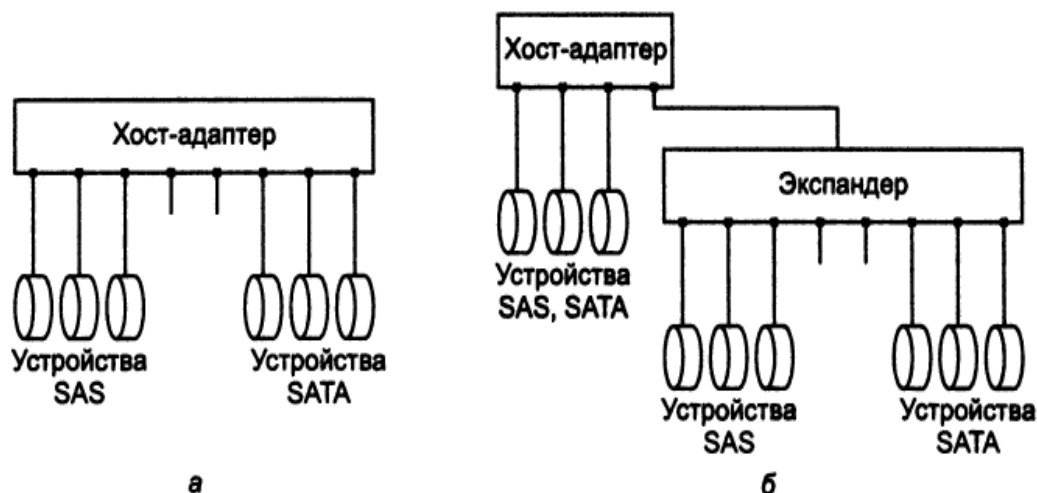


Рис. 5.14. Подключение устройств *SAS* и *SATA* к хост-адаптеру:
а – прямое; б – через экспандер

Архитектурная модель SAS

Архитектурная модель SAS состоит из набора уровней:

- Физический уровень (*physical layer*) определяет коннекторы, кабели и электрические параметры приемопередатчиков.
- *Phy*-уровень (*Phy layer*) обеспечивает последовательную передачу данных (кодирование 8 / 10 В) и специальную «внеполосную» (*OOB*) сигнализацию для служебных целей.
- Канальный уровень (*link layer*) формирует и распознает примитивы, адресные кадры и соединения. На канальном уровне решаются задачи идентификации подключенных устройств, выполнения их сброса, управления соединениями. Для каждого протокола (*SSP*, *STP* и *SMP*) канальный уровень определяет свои правила обмена кадрами и примитивами.
- Уровень порта (*port layer*) является прослойкой между канальным и транспортным уровнями, обеспечивающей установление и разрыв соединений в портах.
- Транспортный уровень (*transport layer*) формирует структуры кадров и транспортные сервисы для протоколов *SSP*, *STP* и *SMP*.
- Прикладной уровень (*application layer*) для протокола *SSP* определяет процедуры выполнения команд *SCSI* согласно архитектурной модели *SAM*. Для протокола *SMP* прикладной уровень определяет функции, необходимые для идентификации устройств, выяснения топологии домена и управления экспандерами. Для протокола *STP* прикладной уровень SAS не вводит каких-либо особенностей по сравнению с *SATA*.

Физический уровень SAS

Физический уровень SAS (*physical layer*) определяет коннекторы и кабели, а также электрические спецификации приемопередатчиков. Эти спецификации совместимы с *SATA*, но различаются в некоторых деталях. В *SATA* заданы характеристики передатчиков и кабелей, которые косвенно задают характеристики приемников; в SAS требования к приемникам заданы явно. В *SATA* допускается связь между устройствами как по постоянному, так и по переменному току; в приемнике SAS должны присутствовать разделительные конденсаторы (в передатчике – не обязательно), так что связь имеется только по переменному току. В *SATA* возможно применение синхронизации с расширением спектра (*SSC*), в SAS расширение спектра не применяется.

Для SAS используются кабели с волновым сопротивлением 100 Ом, в качестве трансиверов используются те же компоненты, что и для современных последовательных интерфейсов *Fibre Channel*, *Gigabit Ethernet*, *XAUI*, *InfiniBand*, *1394b*, *PCI Express*.

За основу разъема для внутреннего исполнения (рис. 5.15, а) взят коннектор *Serial ATA*, в котором в сигнальной секции добавлены контакты для вторичного физического интерфейса (*secondary physical link*). В питающей секции резервный (в *SATA*) контакт *P11* использован для индикации готовности устройства. Подключаемые кабели могут содержать один или два физических интерфейса. Определен и разъем кросс-шины для непосредственного подключения устройств SAS с двумя физическими интерфейсами (устройства *SATA* с этим разъемом несовместимы из-за *L*-образных ключей).

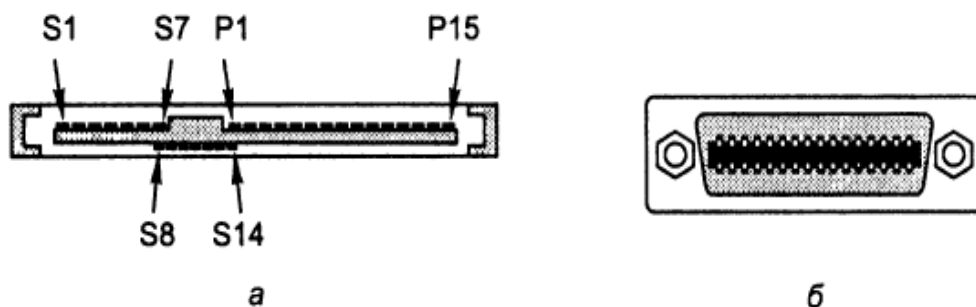


Рис. 5.15. Разъемы устройств SAS: а – внутренний; б – внешний

Для *SAS-1.1* фирма HP предлагает использовать внутренний 4-контактный коннектор, на котором питание и индикация не предусмотрены, но введены контакты для дополнительных интерфейсов. Этот тип разъемов определен в стандарте *SFF-8484*.

Для внешних соединений в SAS приняты 4-канальные кабели и 32-контактные разъемы (рис. 5.15, б), у которых сигналы сгруппированы по направлениям (группа входов и группа выходов); питание и индикация не предусмотрены.

Протокол SSP

Протокол *SSP* обеспечивает выполнение заданий устройствами *SCSI*. Для работы протокола *SSP* требуется установление соединения *SSP*. Запросить соединение *SSP* может как инициатор, так и целевое устройство. После установления соединения *SSP* трансиверы инициатора и целевого устройства обмениваются между собой кадрами *SSP*.

После обмена кадрами по взаимному согласию партнеров соединение закрывается. Канальный уровень для *SSP* обеспечивает гарантированную доставку кадров: на прием каждого кадра трансивер посылает примитив *ACK* (положительное подтверждение) или *NAK* (ошибка *CRC*-контроля). Кадры неверной длины игнорируются.

Канальный уровень обеспечивает управление потоком на основе кредитов: для того чтобы передать кадр, передатчик должен иметь ненулевой кредит. Кредит ему дает партнер по соединению посылкой примитива *RRDY*. При установлении соединения кредиты обнуляются, получение примитива *RRDY* увеличивает кредит на единицу, посылка кадра – уменьшает на единицу. Примитивом *CREDIT BLOCKED* устройство предупреждает об отсутствии кредита.

Кадр *SSP* состоит из примитива *SOF*, двойных слов данных и примитива *EOF*. Обмен выполняется в полнодуплексном режиме: кадры могут передаваться в обоих направлениях. Примитивы-подтверждения и примитивы кредитов могут внедряться в поток двойных слов данных, передаваемых во встречных направлениях.

В протоколе *SSP* используются блокированные и неблокированные передачи кадров. В случае блокированной передачи кадров (*inter-locked frames*) очередной кадр посылается только после получения подтверждения (*ACK* или *NAK*) в приеме предыдущего. Блокированные передачи используются для кадров *COMMAND*, *TASK*, *XFER_RDY*, *RESPONSE*; темп этих передач сдерживается временем оборота между источником и приемником.

Для кадров данных (*DATA*) используется неблокированная передача (*noninter-locked frames*): очередной кадр может быть послан без ожидания подтверждения предыдущих (но подтверждения передаются, хотя приходят позже). Здесь темп потока не зависит от времени прохождения данных между источником и приемником. Поточковые передачи данных аналогичны синхронной передаче в параллельных версиях *SCSI*.

В протоколе *SSP* используется формат кадра, заимствованный из *Fibre Channel*. Кадр начинается с 24-байтного заголовка, за которым может следовать от 0 до 1024 байтов информационного блока. При необходимости кадр выравнивается с помощью байтов-заполнителей (0 – 2); завершается кадр 4-байтным полем *CRC*. Для кадров определено 5 типов: *TASK*, *COMMAND*, *RESPONSE*, *DATA* и *XFER_RDY*.

Кадр *TASK* требуется для управления заданиями (в *Fibre Channel* для этих целей используется кадр *COMMAND*).

Для команд *SCSI*, не требующих передачи данных, инициатор посылает кадр *COMMAND*, по исполнению команды устройство посылает кадр *RESPONSE*.

Для выполнения команд записи (передача данных от инициатора) целевое устройство после получения кадра *COMMAND* посылает кадр *XFER_RDY* (готовность к приему), на который инициатор отвечает одним или несколькими кадрами *DATA*. Посылка *XFER_RDY* и ответная посылка кадров *DATA* могут производиться несколько раз, пока не будут переданы все данные для этой команды (или не возникнет особое условие). По исполнении команды целевое устройство посылает кадр *RESPONSE*.

Для выполнения команд чтения целевое устройство после получения кадра *COMMAND* посылает один или несколько кадров *DATA*, пока не будут переданы все данные для этой команды (или не возникнет особое условие). По исполнении команды целевое устройство посылает кадр *RESPONSE*.

Для двунаправленных команд целевое устройство организует доставку данных по своему усмотрению, запрашивая данные записи кадрами *XFER_RDY* и посылая данные чтения по своей инициативе.

Протокол *SMP*

Протокол *SMP* служит для управления экспандерами и определения топологии домена.

Для протокола *SMP* канальный уровень работает проще, чем *SSP*. Соединение *SMP* может запросить только инициатор. В каждом соединении передаются только два кадра – запрос от инициатора (*SMP_REQUEST*) и ответ на него от адресованного устройства (*SMP_RESPONSE*), после чего обменом примитивами *CLOSE* соединение закрывается.

Внутри соединения *SMP* используются только примитивы *SOF* и *EOF*, обрамляющие кадры. Подтверждать прием кадров (*ACK* или *NAK*) не нужно. Соединения *SMP* работают в полудуплексном режиме. Управление потоком не требуется: открытие соединения является неявным кредитом для посылки единственного кадра инициатором и целевым устройством.

Инициаторами *SMP* могут быть как конечное устройство *SAS*, так и самоконфигурируемые экспандеры. Кадр *SMP* состоит из 4-байтного

заголовка, поля данных (0 – 1024 байт, дополненных до целого числа двойных слов) и 4-байтного поля *CRC*. В заголовке кадра указывается код функции *SMP*, которую запрашивает инициатор протокола *SMP*. С помощью функций *SMP* определяется топология домена, конфигурируются экспандеры и осуществляется управление их портами.

Протокол *STP*

Протокол *STP* обеспечивает туннель между инициатором *SATA* (представленным хост-контроллером *SAS*) и мостом *STP/SATA*, к которому подключено устройство *SATA*. В *SATA* хост и устройство взаимодействуют непосредственно без установления соединений. Протокол *STP* обеспечивает устройству *SATA*, подключенному к мосту *STP/SATA*, и иллюзию непосредственного соединения с хост-контроллером *SATA*, правда, с дополнительными задержками. Эту иллюзию создает *STP*-соединение между портом инициатора протокола *STP* (хост-адаптера *SAS*) и портом моста *STP/SATA* в экспандере, к которому подключено устройство *SATA*. Если устройство *SATA* подключено непосредственно к хост-адаптеру *SAS*, то протокол *STP* не используется (хост-адаптер с данным устройством работает по обычному протоколу *SATA*).

Канальный уровень *STP* построен с учетом протокола *SATA*, который несколько проще протокола устройств *SAS*. Устройства *SATA* никогда не передают кадры во встречных направлениях одновременно. В связи с этим в соединении *STP* используется полудуплексный режим: в одном направлении передаются двойные слова данных, во встречном – примитивы *R_IP*. Кадры *SATA* по конструкции аналогичны кадрам *SAS* (4-байтный заголовок, поле данных и *CRC*-код), но поле данных (*SATA FIS*) может иметь длину до 8192 байт.

Обмен кадрами и примитивами при установлении соединения *STP* аналогичен установлению соединения *SSP*; этот обмен выполняется между хостом и мостом и не выходит на порт подключения устройства *SATA*. Мост подтвердит установление соединения, только если порт устройства *SATA* находится в покое. Закрытие соединения *STP* также происходит невидимо для устройства *SATA*, и оно также возможно только в состоянии покоя. Внутри открытого соединения кадры и примитивы передаются по сквозному пути между трансивером инициатора и трансивером устройства *SATA*. Передача кадров *SATA* в соединении *STP* аналогична протоколу *SATA*.

Устройства *SATA*, в отличие от *SAS* (и всех устройств *SCSI*), рассчитаны на работу лишь с одним хостом. Целевое устройство протокола *STP* (мост *STP/SATA*) может работать и с несколькими инициаторами. Чтобы защитить устройство *SATA* от работы с несколькими инициаторами, используется механизм привязки (*affiliation*). Мост устанавливает соединение с первым инициатором, который это соединение запрашивает, и привязывает этот мост (фактически устройство *SATA*) к данному инициатору. Данная привязка удерживается, пока не произойдет одно из следующих событий:

- выключается питание моста;
- соединение закрывается специальным примитивом *CLOSE (CLEAR AFFILIATION)*, посланным данным инициатором;
- управляющими функциями протокола *SMP (PHY CONTROL)* выполняется сброс трансивера, к которому подключено устройство *SAS*;
- физический интерфейс устройства *SATA* начинает последовательность сброса.

Инициатор, не привязанный к данному устройству, на запрос соединения *STP* будет получать отказ.

В *SATA* определены примитивы запроса режимов пониженного энергопотребления интерфейса *SATA_PMREQ_P* (режим *Partial*) и *SATAJPMREQJS* (режим *Slumber*). Эти запросы в *STP* не поддерживаются.

Устройства, порты и соединения SAS

Подключение устройства *SAS* обеспечивает трансивер *Phy* – приемник и передатчик последовательного интерфейса. Приемник и передатчик работают на одинаковых скоростях (*G1* – 1,5 Гбит/с, *G2* – 3 Гбит/с), прием и передача идут независимо друг от друга (полный дуплекс). Каждый трансивер в устройстве имеет собственный идентификатор (номер).

Устройства *SAS* содержат порты. Порт – это группа трансиверов (от 1 до 128) с одинаковыми *SAS*-адресами, подключенных к другой группе трансиверов с одинаковыми (но другими) *SAS*-адресами. «Узкий» (*narrow*) порт содержит один трансивер, «широкий» – два и более трансиверов. Порты определяются (конфигурируются) на этапе инициализации устройства.

SAS-адрес – это глобально уникальный 64-битный идентификатор, который присваивается каждому *SAS*-порту и каждому *SAS*-экспандеру.

В SAS определены два класса устройств: конечные устройства (*end device*) и устройства-экспандеры (*expander device*).

Конечные устройства – это устройства SAS, являющиеся инициаторами или/ и целевыми устройствами SCSI (или SATA).

Примером конечного устройства считается SAS-диск, у которого два трансивера всегда имеют отдельные SAS-адреса; устройство двухпортовое (для диска «широкий» порт не требуется).

Другой пример – хост-адаптер, содержащий 8 трансиверов, в котором возможны разные варианты конфигурации:

- один SAS-адрес на все трансиверы – это адаптер с одним, потенциально «очень широким» портом;

- два SAS-адреса, каждый на группу из четырех трансиверов – двухпортовый адаптер (удобен для подключения четырехканальными кабелями);

- восемь SAS-адресов – восьмипортовый адаптер, к которому можно подключить до 8 устройств SAS и SATA.

Устройства-экспандеры служат для объединения конечных устройств в сложных конфигурациях. Экспандер имеет собственный SAS-адрес для управления его функциями по протоколу SMP. Экспандер может содержать и внутренние устройства SAS (со своими SAS-адресами), подключаемые к его портам с виртуальными трансиверами. Эти устройства могут, например, использоваться для управления блоком (питание, климат, защита и т. п.). Каждый трансивер экспандера имеет собственный идентификатор (номер), уникальный в пределах экспандера. Порты экспандера могут служить для подключения инициаторов и целевых устройств SAS, а также других экспандеров. По этим портам будут передаваться кадры любых протоколов (SSP, STP, SMP). Экспандер может (не обязательно) содержать мосты STP/SATA (один или несколько), позволяющие к портам экспандера (узким) подключать устройства SATA. По этим портам будут передаваться только кадры SATA.

Для выполнения задания SCSI (передачи команды, данных или сообщения состояния) необходимо установить SSP-соединение (*SSP-connection*) – временную связь между трансиверами инициатора и целевого устройства. Соединения устанавливаются также и для обмена с устройствами SATA (*STP-соединение*), и для управления экспандерами (*SMP-соединения*). Установление соединения – характерная

черта SAS, обусловленная идеологией SCSI (в SATA понятие соединения отсутствует).

Для установления соединения устройство посылает запрос, указав SAS-адрес желаемого партнера. Экспандеры, выполняя маршрутизацию запроса, прокладывают путь (*pathway*) – определяют набор физических связей (*links*) между инициатором и целевым устройством соединения. Этот путь сохраняется до закрытия (или разрыва) соединения, и по нему передаются кадры и примитивы как по выделенному двухточечному соединению со скоростью, доступной всем участкам пути.

Соединения адресуются к портам, но устанавливаются между трансиверами. Широкий порт (с N трансиверами) может устанавливать соединения одновременно с N различными портами. Два широких порта могут устанавливать друг с другом до N соединений. SAS-диски всегда представлены парой «узких» портов, «широкие» порты имеют только хост-адаптеры и RAID-контроллеры.

Топология домена и маршрутизация

Домен SAS в простейшем случае состоит из непосредственно соединенных друг с другом портов инициатора и целевого устройства SSP. Если используются экспандеры, то в домене появляются еще и порты инициаторов и целевых устройств SMP, необходимые для конфигурирования домена (экспандеров). Если используются и устройства SATA, то соответственно появляются порты-инициаторы и целевые устройства STP. Возможны сложные конфигурации с несколькими доменами (рис. 5.16).

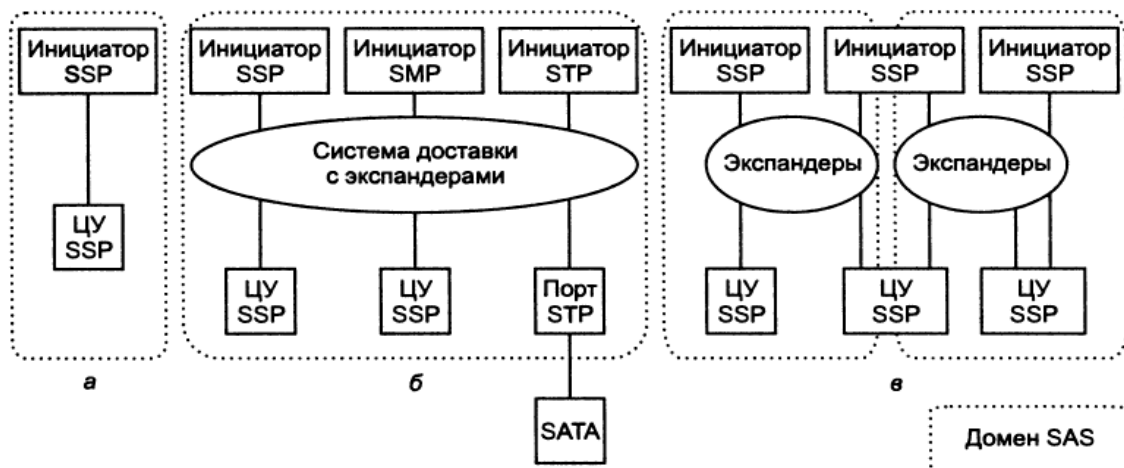


Рис. 5.16. Логическая структура доменов SAS: а – простейшая двухточечная; б – с экспандерами; в – с разделяемыми устройствами (инициатором и ЦУ)

Экспандеры по назначению разделяются на два типа:

- из *E*-экспандеров (*edge expander*) собираются наборы *E*-экспандеров (*edge expander device set*), обеспечивающие подключение конечных устройств;

- *F*-экспандеры (*fanout expander*) служат для объединения множества наборов *E*-экспандеров и конечных устройств.

Набор *E*-экспандеров представляет собой древовидную структуру экспандеров. Набор может вырождаться и в один экспандер. Каждый набор может содержать до 128 *SAS*-адресов (устройств-экспандеров и портов конечных устройств).

В одном домене *SAS* может присутствовать не более одного *F*-экспандера, с помощью которого объединяются до 128 наборов. Если *F*-экспандеров нет, то в домен могут объединяться не более двух наборов. Конечные устройства могут подключаться и к *F*-экспандерам, и к *E*-экспандерам (рис. 5.17). Между любыми устройствами (и конечными, и экспандерами) возможны соединения «широкими» интерфейсами, петлевые соединения запрещены, множественные пути отсутствуют.

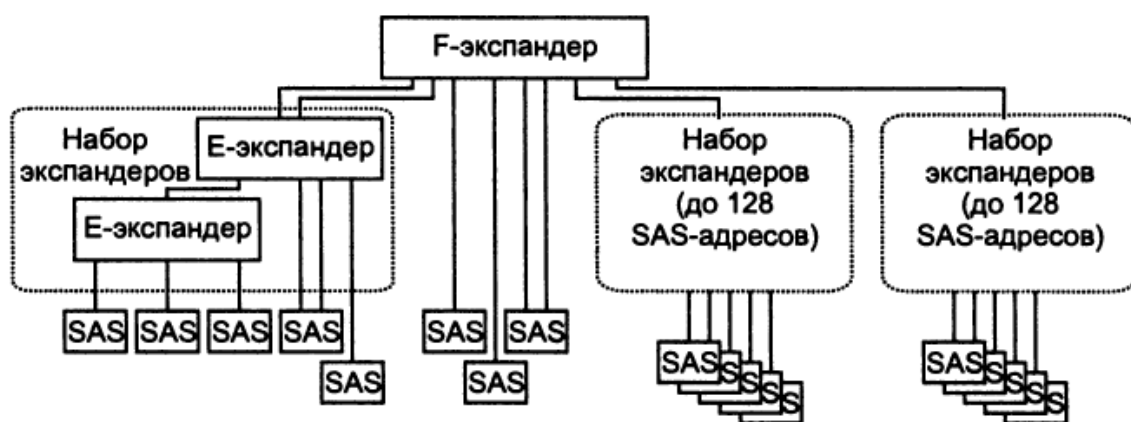


Рис. 5.17. Домен *SAS*

Маршрутизацию в домене *SAS* выполняют экспандеры. Выполняется она при установлении соединений (*SSP*, *STP* или *SMP*), внутри соединения обмен между инициатором и целевым устройством происходит по проложенному маршруту. Для маршрутизации экспандеры должны быть предварительно сконфигурированы: заполнены их таблицы маршрутизации. Этим занимается устройство-инициатор

протокола *SMP*. Экспандер может быть самоконфигурируемым – у него имеется внутренний инициатор *SMP*, который заполнит таблицы (свои и других экспандеров).

Приняв запрос соединения от какого-то трансивера (кадр *OPEN*), экспандер по *SAS*-адресу назначения, содержащемуся в этом кадре, ищет свой трансивер, в который нужно послать данный запрос.

Для каждого трансивера экспандера известны *SAS*-адрес, тип и допустимая скорость подключенного устройства. Эта информация получается на этапе идентификации, завершающем инициализацию физического интерфейса. Если к трансиверу подключено конечное устройство, то к нему выполняется прямая маршрутизация: в этот трансивер посылаются запросы с *SAS*-адресом, совпадающим с адресом подключенного устройства. Если к трансиверу подключен другой экспандер, то сначала выполняется табличная маршрутизация. В экспандере имеются таблицы маршрутизации, в которых указываются *SAS*-адреса, доступные для каждого его трансивера. Если для пришедшего запроса в таблицах не найден трансивер, в который его следует послать, то используется субтрактивная маршрутизация. В *E*-экспандере может присутствовать субтрактивный порт (набор из одного или нескольких трансиверов, к которым подключены трансиверы с одинаковыми *SAS*-адресами). В этот порт посылаются запросы, для которых не удалось найти трансивер при прямой или табличной маршрутизации. Субтрактивный порт может быть подключен только к порту другого экспандера, и этот порт должен задействовать только табличную маршрутизацию. *F*-экспандеры не используют субтрактивную маршрутизацию. Они могут служить средством объединения множества наборов *E*-экспандеров и обычно поддерживают большие размеры таблиц для табличной маршрутизации.

В *E*-экспандерах различают нисходящие (*downstream*) трансиверы, использующие прямую и табличную маршрутизации, и восходящие (*upstream*) трансиверы, использующие субтрактивную маршрутизацию. На границе набора *E*-экспандеров может находиться корневой экспандер, восходящий (субтрактивный) порт которого подключается к *F*-экспандеру или к другому набору *E*-экспандеров.

5.4. Технология маршрутизации команд

Технология маршрутизации команд (*Native Command Queuing – NCQ*) была предложена и реализована в интерфейсе *SCSI*, а позднее в виде дополнения к стандарту *Serial ATA 1.0* и является неотъемлемой частью нового стандарта *Serial ATA 2*.

Native Command Queuing – это расширенный протокол команд *Serial ATA*, который позволяет нескольким различным приложениям обращаться с запросами ввода-вывода в одно и то же время. Накопители с поддержкой *NCQ* используют специальный буфер, внутри которого команды, стоящие в очереди, могут динамически перестраиваться в соответствии с положением считывающей головки и текущей загрузкой. Кроме того, *NCQ* описывает механизм, который позволяет сортировать команды, в то время как дисковый накопитель работает с данными другой команды. Таким образом, теоретически применение *NCQ* позволяет повысить общую производительность системы, когда приложения активно используют технологию асинхронных запросов ввода-вывода.

При использовании механизма *NCQ* контроллер анализирует запросы и оптимизирует очередность их выполнения таким образом, чтобы увеличить скорость передачи данных и минимизировать время поиска. Кроме того, контроллеры *Native Serial ATA* способны самостоятельно организовывать очереди запросов и выполнять их без вмешательства процессора. Механизм работы *Serial ATA NCQ* в общих чертах можно сравнить с организацией очереди команд интерфейса *SCSI*, но в отличие от 256-уровневой архитектуры в случае со *SCSI* поддерживает 32-уровневую глубину маршрутизации команд.

Темы для закрепления знаний

1. Виды интерфейсов дисковых накопителей. Интерфейс *IDE*. Разновидности.
2. Интерфейс *IDE (ATA)*. Виды адресации. Виды адаптеров. Конфигурирование.
3. Интерфейс *IDE (ATA)*. Физические параметры.

4. Интерфейс *IDE (ATA)*. Виды протоколов и режимы передачи данных.
5. Интерфейс *IDE (ATA)*. Протокол – основные шаги.
6. Интерфейс *SATA*.
7. *SCSI*. Общие параметры, характеристики и свойства. Назначение. Преимущества. Основные параметры спецификации *SCSI-3*.
8. *SCSI-3*. Архитектурная модель. Адресация. Временные диаграммы обмена.
9. *SCSI*. Физический уровень. Версии по видам сигналов. Разъемы и кабели.
10. *SCSI*. Контроллер – интерфейсы для его подключения и дополнительные элементы в нем. *SCSI*-устройства и их подключение.
11. Схема включения контроллера *SCSI*.
12. Блок-схема контроллера *DP5380*.
13. Подключение терминаторов *SCSI*.
14. *SCSI*. Конфигурирование. Варианты *BIOS*.
15. Технология маршрутизации команд.
16. Протоколы *SCSI*.

Глава 6. БЕСПРОВОДНЫЕ ИНТЕРФЕЙСЫ

Беспроводные интерфейсы в настоящее время завоевывают все большую популярность, так как позволяют избавиться от обилия проводов на столах и в офисе. Основной носитель сигнала в беспроводных интерфейсах – радиоволны. Инфракрасное излучение не получило широкого распространения, так как имеет малый угол охвата и требует жесткой направленности. Поэтому оно широко применяется лишь в чисто симплексных (однонаправленных) интерфейсах для управления устройствами (проекторами, телевизорами и т.п.).

Параметры основных радиоинтерфейсов приведены в табл. 6.1. В ней представлены как интерфейсы периферийных устройств ЭВМ, так и сетевые интерфейсы, которые далее детально не рассматриваются.

Таблица 6.1. Основные характеристики технологий беспроводной передачи данных

№ п/п	Технология	Максимальная возможная скорость соединения, Мбит/с	Дальность соединения, м	Рабочий диапазон частот, МГц	Сфера использования
1	<i>WiMAX</i>	70	50000	2000 – 66000	Доступ в интернет, «последняя миля»
2	<i>Wi-Fi 802.11n</i>	240	250	2400, 5000	<i>LAN</i> , доступ в интернет через хост-слоты
3	<i>CAT-1Q (DECT)</i>	3,8	500	1880 – 1900	Телефония и дополнительные сервисы
4	<i>Wireless USB</i>	480	10	3100 – 10600	Сопряжение периферийных устройств
5	<i>Bluetooth 2.0</i>	3 – 24	100	2400	Сопряжение периферийных устройств
6	<i>WiGig</i>	6000	10	10000 – 60000	Сопряжение периферийных устройств
7	<i>UltraWide Band (UWB)</i>	1000	~50	1000	То же

6.1. Bluetooth

Интерфейс появился в 1998 г. Его поддерживает группа компаний *SIG-2000*. Интерфейс регламентируется стандартом *IEEE 802.11* – локальные беспроводные сети.

Существует конкурирующий интерфейс *HomeRF*, но у него пропускная способность ниже.

Интерфейс *Bluetooth* использует для связи открытый диапазон *ISM* (промышленный, научно-технический, медицинский диапазон) с частотами $f = 2,4 - 2,4835$ ГГц, для работы в котором не требуется разрешений. В некоторых странах (Франция, Япония и др.) этот диапазон уменьшен с 80 до 23 МГц.

Скорость передачи зависит от версии интерфейса и объема передаваемого файла. Для *Bluetooth* 1.2 она составляет:

- 720 Кб/с в асимметричном режиме;
- 420 Кб/с в полудуплексном режиме.

Для *Bluetooth* 3.0 и выше она составляет:

- для малых файлов – 3 Мб/с;
- для больших файлов – 20 Мб/с;

Мощность передатчика не должна превышать $P \leq 10$ мВт. При этом дальность связи составляет от 10 до 100 м.

Частотные скачки

Весь доступный диапазон делится на поддиапазоны по $\Delta f = 1$ МГц. Количество поддиапазонов зависит от страны и может быть от 23 до 79. Каждый канал представляет собой псевдослучайную последовательность (ПСП) скачков частоты по 79 – 23 поддиапазоном. По времени весь канал делится на сегменты (тайм-слоты) длительностью 625 мкс. Скачки происходят синхронно со скоростью 1600 скачков в секунду. В одной пикосети может быть до 7 подчиненных устройств и 1 основное. Подчиненные устройства могут входить в несколько сетей. Через 227 мс происходит переключение (227 – длина ПСП (всего 10 вариантов)). Параметры ПСП устанавливает основное (ведущее) устройство и сообщает всем подчиненным (ведомым) при установлении связи. Каждая пикосеть имеет свои параметры ПСП, что позволяет работать двум пикосетям в одном помещении, не мешая друг другу, так как вероятность совпадения поддиапазонов в каждый момент времени достаточно мала.

Спецификации

• *Bluetooth* 1.0

Устройства версий 1.0 (1998) и 1.0B имели плохую совместимость между продуктами различных производителей. В 1.0 и 1.0B была обязательной передача адреса устройства (*BD_ADDR*) на этапе установления связи, что делало невозможной реализацию анонимности соединения на протокольном уровне и было основным недостатком данной спецификации.

• *Bluetooth* 1.1

В *Bluetooth* 1.1 было исправлено множество ошибок, найденных в 1.0B, добавлена поддержка для нешифрованных каналов, индикация уровня мощности принимаемого сигнала (*RSSI*).

- *Bluetooth 1.2*

В версии 1.2 была добавлена технология адаптивной перестройки рабочей частоты (*A FH*), что улучшило сопротивляемость к электромагнитной интерференции (помехам) путём использования различных частот в последовательности перестройки. Также увеличилась скорость передачи и добавилась технология *eSCO*, которая улучшила качество передачи голоса путём повторения повреждённых пакетов. В *HCI* добавилась поддержка трёхпроводного интерфейса *UART*.

Главные улучшения включают следующее:

- быстрое подключение и обнаружение;
- адаптивную перестройку частоты с расширенным спектром (*A FH*), которая повышает стойкость к радиопомехам;
- более высокие по сравнению с версией 1.1 скорости передачи данных, практически до 721 кбит/с;
- расширенные синхронные подключения (*eSCO*), которые улучшают качество передачи голоса в аудиопотоке, позволяя повторную передачу повреждённых пакетов, и при необходимости могут увеличить задержку аудиоинформации, чтобы оказать лучшую поддержку для параллельной передачи данных;
- в *HostControllerInterface (HCI)* добавлена поддержка трёхпроводного интерфейса *UART*;
- утверждён как стандарт *IEEE Standard 802.15.1-2005*;
- введены режимы управления потоком данных (*FlowControl*) и повторной передачи (*RetransmissionModes*) для *L2CAP*.

- *Bluetooth 2.0 + EDR*

Bluetooth версии 2.0 был выпущен 10 ноября 2004 г. Имеет обратную совместимость с предыдущими версиями 1.x. Основным нововведением стала поддержка *EDR (EnhancedDataRate)* для ускорения передачи данных. Номинальная скорость *EDR* около 3 Мбит/с, однако на практике это позволило повысить скорость передачи данных только до 2,1 Мбит/с. Дополнительная производительность достигается с помощью различных радиотехнологий для передачи данных.

Стандартная (или базовая) скорость передачи данных использует Гауссово кодирование со сдвигом частот (*GFSK*), модуляцию радиосигнала при скорости передачи в 1 Мбит/с. *EDR* использует сочетание *GFSK* и *PSK*-модуляции с двумя вариантами: *2/4 DQPSK* и

8 *DPSK*. Они имеют большие скорости передачи данных по воздуху 2 и 3 Мбит/с соответственно.

Bluetooth SIG издала спецификацию «Технология *Bluetooth 2.0 + EDR*», которая подразумевает, что *EDR* является дополнительной функцией. Кроме *EDR*, есть и другие незначительные усовершенствования по сравнению со спецификацией 2.0, и продукты могут соответствовать «Технологии *Bluetooth 2.0*», не поддерживая более высокую скорость передачи данных. По крайней мере, одно коммерческое устройство *HTC TyTN Pocket PC* использует «*Bluetooth 2.0 без EDR*» в своих технических спецификациях.

Согласно 2.0 + *EDR* спецификации *EDR* обеспечивает следующие преимущества:

- увеличение скорости передачи в 3 раза (2.1 Мбит/с) в некоторых случаях;
- уменьшение сложности нескольких одновременных подключений из-за дополнительной полосы пропускания;
- более низкое потребление энергии благодаря уменьшению нагрузки.

- *Bluetooth 2.1*

Эта спецификация принята в 2007 г. В ней добавлена технология расширенного запроса характеристик устройства (для дополнительной фильтрации списка при сопряжении), энергосберегающая технология *SniffSubrating*, которая позволяет увеличить продолжительность работы устройства от одного заряда аккумулятора в 3 – 10 раз. Кроме того, обновлённая спецификация существенно упрощает и ускоряет установление связи между двумя устройствами, позволяет производить обновление ключа шифрования без разрыва соединения, а также делает указанные соединения более защищёнными благодаря использованию технологии *NearFieldCommunication*.

- *Bluetooth 2.1 + EDR*

В августе 2008 г. *Bluetooth SIG* представил версию 2.1+*EDR*. Новая редакция *Bluetooth* снижает потребление энергии в 5 раз, повышает уровень защиты данных и облегчает распознавание и соединение *Bluetooth*-устройств благодаря уменьшению количества шагов, за которые оно выполняется.

- *Bluetooth 3.0 + HS*

3.0 + *HS* спецификация была принята *Bluetooth SIG* 21 апреля 2009 г. Она поддерживает теоретическую скорость передачи данных

до 24 Мбит/с. Её особенностью является добавление *AMP* (асимметричная мультипроцессорная обработка) (альтернативно *MAC/PHY*), дополненное в стандарте 802.11 как высокоскоростное сообщение. Две технологии были предусмотрены для *AMP*: 802.11 и *UWB*, но *UWB* отсутствует в спецификации.

Модули с поддержкой новой спецификации соединяют в себе две радиосистемы: первая обеспечивает передачу данных в 3 Мб/с (стандартная для *Bluetooth 2.0*) и имеет низкое энергопотребление; вторая совместима со стандартом 802.11 и обеспечивает возможность передачи данных со скоростью до 24 Мбит/с (сравнима со скоростью сетей *Wi-Fi*). Выбор радиосистемы для передачи данных зависит от размера передаваемого файла. Небольшие файлы передаются по медленному каналу, а большие – по высокоскоростному. *Bluetooth 3.0* использует более общий стандарт 802.11 (без суффикса), т. е. не совместим с такими спецификациями *Wi-Fi*, как 802.11b/g или 802.11n.

- *Bluetooth 4.0*

В декабре 2009 г. *Bluetooth SIG* анонсировала стандарт *Bluetooth 4.0*. Технология, прежде всего, предназначена для миниатюрных электронных датчиков (использующихся в спортивной обуви, тренажерах, миниатюрных сенсорах, размещаемых на теле пациентов и т. д.).

В *Bluetooth 4.0* достигается низкое энергопотребление за счёт использования специального алгоритма работы. Передатчик включается только на время отправки данных, что обеспечивает возможность работы от одной батарейки типа *CR2032* в течение нескольких лет. Стандарт определяет скорость передачи данных в 1 Мбит/с при размере пакета данных 8 – 27 байт. В новой версии два *Bluetooth*-устройства смогут устанавливать соединение менее чем за 5 мс и поддерживать его на расстоянии до 100 м. Для этого используется усовершенствованная коррекция ошибок, а обеспечивает необходимый уровень безопасности 128-битное *AES*-шифрование.

Сенсоры температуры, давления, влажности, скорости передвижения и т. д. на базе этого стандарта могут передавать информацию на различные устройства контроля: мобильные телефоны, КПК, ПК и т. п.

6.1.1. Принцип работы и соединения *Bluetooth*-устройств

Первоначально все *Bluetooth*-устройства находятся в режиме *Standby*, в котором они каждые 1,28 с прослушивают 32 фиксирован-

ные частоты. Устройство, оказавшись в незнакомом окружении, начинает посылать запросы на этих частотах.

При приеме запроса *Bluetooth*-устройство реагирует на него в соответствии с режимом, в котором оно находится. В режиме отклика (*discoverablemode*) устройство всегда отвечает на все полученные запросы. Режим ограниченного отклика (*limiteddiscoverablemode*) подразумевает, что устройство может отвечать на запросы только ограниченное время или при других определенных условиях. Третий режим отказа в отклике (*non-discoverablemode*) запрещает устройству отвечать на запрос.

Кроме того, при подключении устройство может быть подключаемым (*connectablemode*) или недоступным (*non-connectablemode*). Во втором случае устройство не позволяет настроить ряд параметров соединения и обмениваться данными.

После взаимного обнаружения устройства определяют используемый диапазон частот, размер страниц, количество и порядок смены частот и другие физические параметры соединения.

Каждое *Bluetooth*-устройство имеет глобальный уникальный адрес (аналог *MAC*-адресов у сетевых плат), но на уровне пользователя обычно используется имя устройства, которое может быть любым и не быть уникальным. Имя может быть длиной до 248 байт и использовать кодовую страницу в соответствии с *Unicode UTF-8*.

Bluetooth-устройства могут устанавливать соединения не только типа точка-точка, но, что и является основным, многоточечные соединения. При этом они объединяются в пикосети. Процедура соединения инициируется одним, причем любым из устройств, которое и становится главным (*master*) в пикосети. При этом в одной пикосети может быть не более семи ведомых (*slave*) устройств.

6.1.2. Этапы автоматической установки соединения

1. Поиск других устройств – для этого посылается запрос. Ведущее устройство «договаривается» о параметрах связи. Ответ на этот запрос зависит от режима, в котором находится устройство. Режимов может быть три:

– *DiscoverableMode* – устройство отвечает на все полученные запросы;

- *LimitedDiscoverableMode* – устройство может отвечать на запросы только ограниченное время;
- *NonDiscoverableMode* – устройство не отвечает на запросы – пассивный режим.

2. Если процесс обнаружения прошел нормально, то новое устройство получает набор адресов. При этом любое устройство должно иметь свой глобальный адрес, но при работе в сети используется не этот адрес, а специальная кодовая страница, и название устройства принимает вид – 48 символов, а всего может быть 248 байт.

3. Автоматическое подключение *Bluetooth* к службам других устройств *ServiceDiscovery*. Для поиска используется специальный протокол *SDP*.

6.1.3. Стек протокола

На рис. 6.1 показано, что стек протокола состоит из уровня радио в основании, которое формирует физический интерфейс соединения. Уровень протокола связи *baseband* и *LinkManagerProtocol(LMP)* устанавливают и контролируют соединения между устройствами *Bluetooth*. Эти три уровня являются основными и реализованы в оборудовании и программном обеспечении.

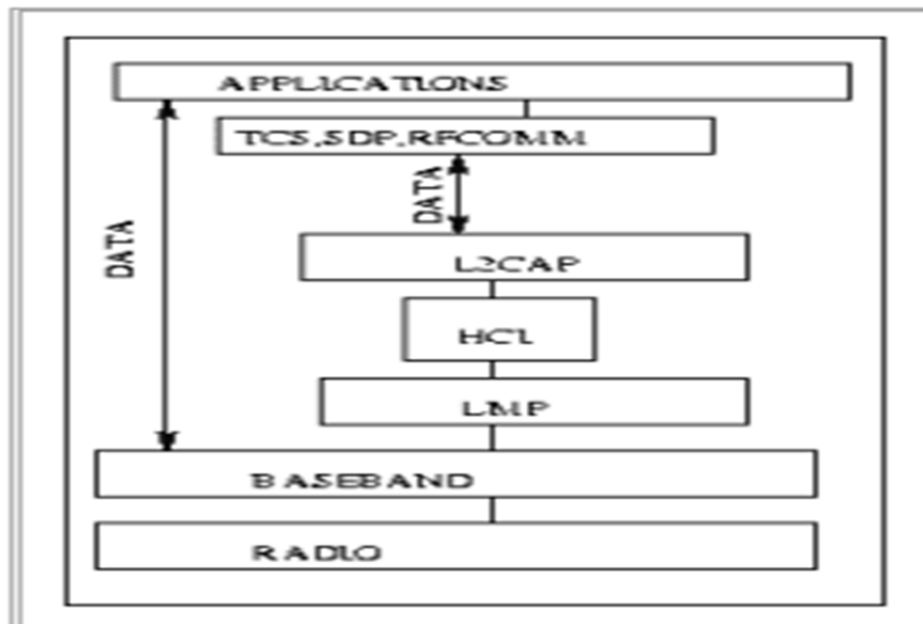


Рис. 6.1. Стек протокола *Bluetooth*

Уровень *HostController* требуется, чтобы соединить нижние уровни с помощью интерфейса *Bluetooth* с верхним протоколом *L2CAP (Logical Link Control Adaptation Protocol)*. Ведущий контроллер требуется только тогда, когда *L2CAP* постоянно находится в программном обеспечении в хосте. Если *L2CAP* есть также на модуле *Bluetooth*, этот уровень не требуется, так как *L2CAP* может непосредственно связываться с *LMP* и *baseband*. Приложения постоянно находятся выше *L2CAP*. Следующие подразделы дают краткое описание каждого уровня.

6.1.4. Слои протоколов *BlueTooth*

Слои протоколов *BT* описаны в табл. 6.2.

Таблица 6.2. Слои протоколов *BlueTooth*

№ п/п	Протокольный слой	Протоколы в стеке
1	Корневые протоколы (<i>CoreProtocol</i>)	<i>BaseBand, LMP, L2CAP, SDP</i>
2	Протокол с заменой кабеля (кабельное соединение)	<i>RFComm</i>
3	Протокол управления телефонией	<i>TCS binary, AT-команда</i>
4	Воспринятые протоколы (<i>AdaptedProtocol</i>)	<i>PPP, UDP/TCP/IP, OBEX, WAP, VCARD, IrMC, WAE</i>

Корневые протоколы

1. Протокол базовой полосы и уровень управления *LCL* обеспечивают физическую радиочастотную связь между устройствами. Он использует процедуру опроса и пейджинга для синхронизации и таймеров. Реализует два способа доступа: синхронный (*SCO*); асинхронный (*ACL*).

2. Протокол диспетчера подключения *LMP* ответственен за установление подключения между устройствами, он решает и контролирует вопросы безопасности, имеет более высокий приоритет, чем другие протоколы.

3. Протокол управления логическим подключением и адаптацией *L2CAP* адаптирует протоколы верхнего уровня к базовой полосе, яв-

ляется базовым протоколом передачи данных, работает только с асинхронными соединениями.

Основные особенности этого протокола:

- возможность определить, к какому протоколу или службе относится этот пакет и кто его ждет;
- обеспечивает разбивку большого пакета на несколько более мелких и последующую сборку;
- позволяет отслеживать свободные ресурсы и загружать их при необходимости;
- поддерживает адресацию нескольким клиентам, широкодоступный доступ.

4. Протокол обнаружения услуг – *SDP*.

5. Протокол, заменяющий кабель, *RFComm*. Он позволяет преобразовывать данные в аналог интерфейса *RS-232C*.

6. Контроль телефонии *TCS binary* – двоичный протокол, ориентирован на битовую передачу информации. Обеспечивает контроль синхронизации вызова.

7. Протокол управления телефонией – *AT*-команды.

8. *Voice* или *BlueToothAudi* – протокол передачи аудиоданных. При передаче аудиоинформации звуковой поток квантуется по уровню 16 битами и обеспечивает полосу 24 кГц. Он не позволяет передавать видеоинформацию.

6.1.5. Модели использования

В интерфейсе ВТ используются четыре специальных профиля доступа:

- типовой профиль доступа *CAP*;
- последовательный профиль доступа *SPP*;
- профиль обнаружения прикладных услуг *SDAP*;
- профиль общего назначения для обмена объектами *GOEP*.

6.1.6. Асинхронный режим

Для кодирования пакетной информации используется частотная манипуляция. Каждый пакет информации передается в течение одного временного интервала (рис. 6.2), однако в особых случаях количество таких интервалов может быть увеличено до пяти (рис. 6.3).

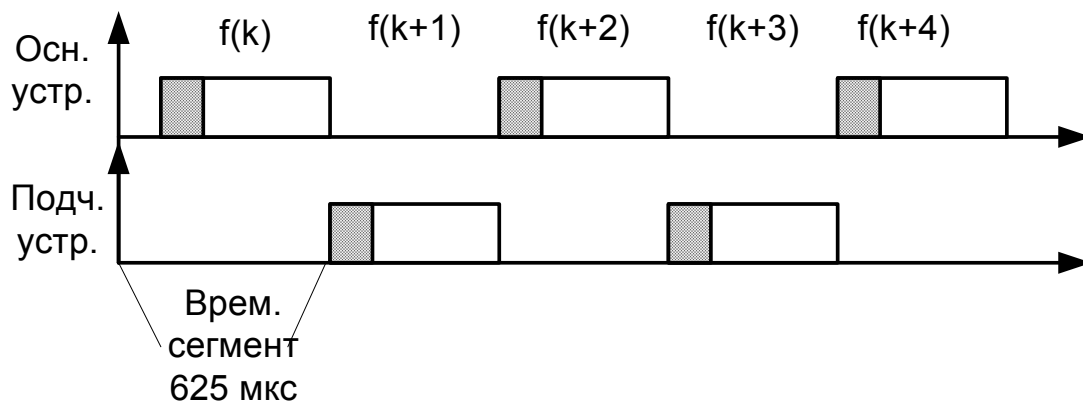


Рис. 6.2. Временная диаграмма асинхронного режима

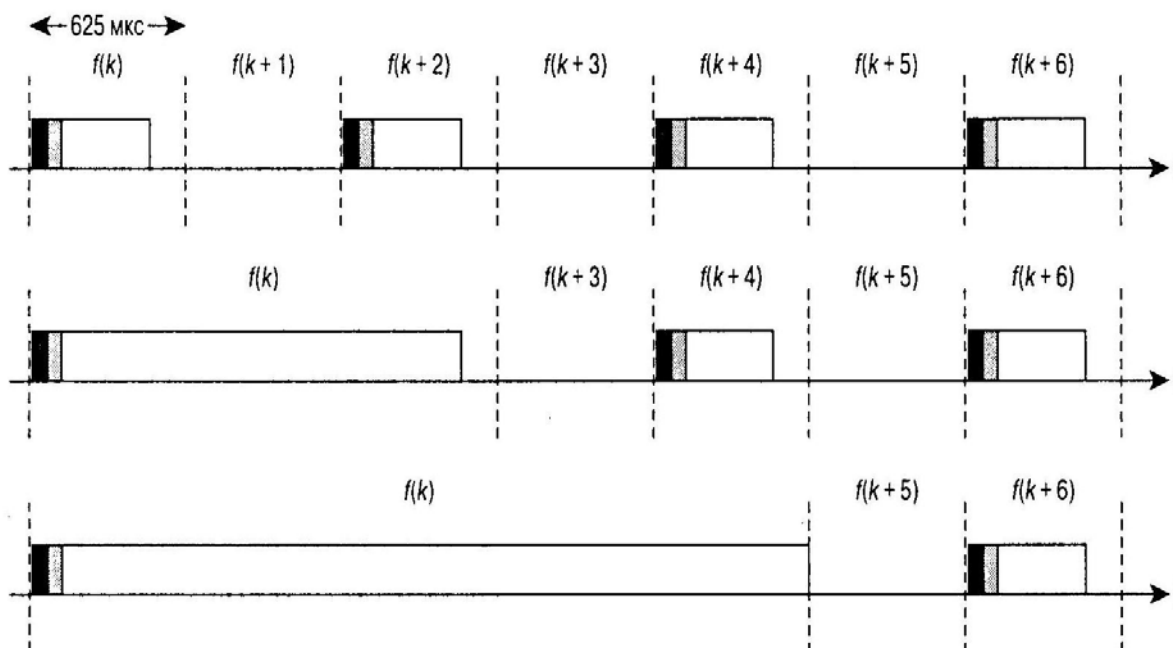


Рис. 6.3. Пример многослововых пакетов

Асинхронные пакеты передаются в то время, которое не зарезервировано для синхронных.

6.1.7. Пакеты и их структура

Структура пакета представлена на рис. 6.4 и содержит три поля:

1. Код доступа – 72 бита; он состоит из подполей:

- начальная часть – 4 бита;
- слово синхронизации – 64 бита;
- завершитель – 4 бита.

2. Заголовок – 54 бита, который состоит из шести подполей и может отсутствовать.

3. Полезная нагрузка (данные) – 0 – 2745 бит, он состоит из подполей:

- заголовок полезной нагрузки – 8 или 16 бит;
- тело полезной нагрузки;
- CRC-код – 16 бит.

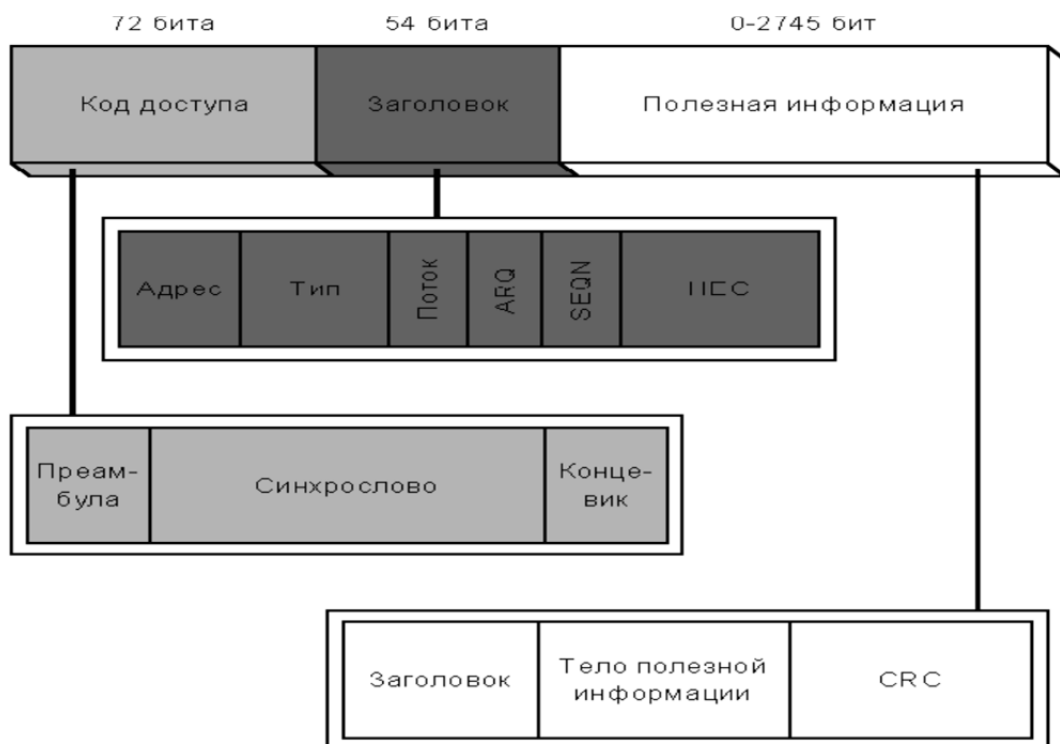


Рис. 6.4. Структура пакета *Bluetooth*

6.1.8. Структура устройства

Общая структура устройства представлена на рис. 6.5.

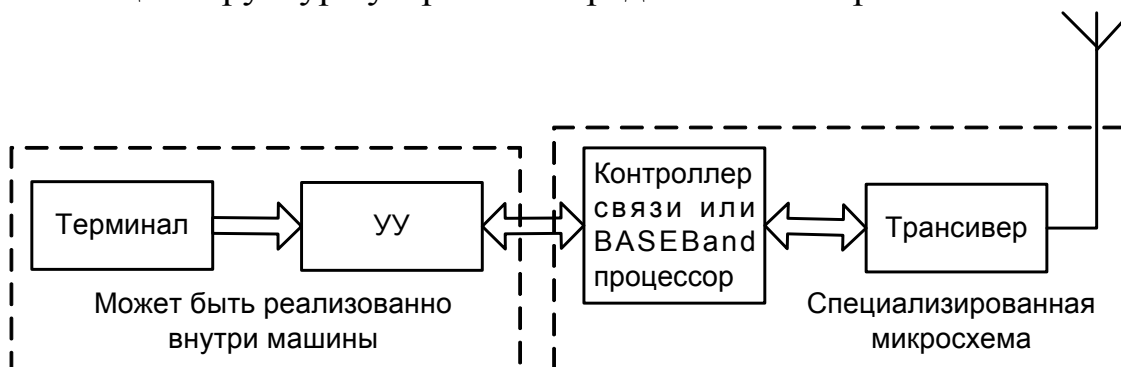


Рис. 6.5. Общая структура устройства

фицируются соответствующие записи в настольном компьютере и наоборот.

2. Автоматическая резервная синхронизация. Представьте себе, что босс изменил ваш план мероприятий, в то время как вы находитесь в поездке. Офисный ПК посылает изменения на ваш сотовый телефон, который автоматически соединяется с ноутбуком и передает модифицированное расписание.

3. Соединение между собой всех периферийных устройств. Беспроводное соединение настольного ПК или ноутбука с принтером, сканером и локальной вычислительной сетью. Особенно усиливает ощущение «свободы» беспроводное соединение «мышки» и клавиатуры с персональным компьютером.

4. Составление *E-mail*-сообщения с использованием ноутбука в салоне летящего самолета. Уже в аэропорту, стоя в очереди к таможеннику, вы включаете мобильный телефон, и все сообщения немедленно отсылаются.

5. Насладитесь комфортом автомобильного комплекта *handsfree*. Сотовый телефон остается в кармане, а *handsfree* соединяется с ним посредством беспроводной связи. Отдавая речевые команды, вы набираете номер и говорите по телефону.

6. Передача фотографий и видеоклипов из любой точки. Цифровая камера соединяется с мобильным телефоном (без проводов), а вы добавляете комментарии посредством клавиатуры на телефоне или ноутбуке. Фото и текст отправляются адресату.

7. Соединение всех участников для мгновенного обмена информацией. Находясь на встрече или конференции, можно быстро обсудить интересующий вопрос со всеми ее участниками. Кроме того, теперь вы можете дистанционно выполнять функции управления, например, включить проектор.

8. Использование ноутбука для доступа в Интернет независимо от вида подключения – через мобильный телефон, модем или по локальной сети.

9. Использование портативного ПК в качестве спикерфона. Соединив беспроводный головной телефон (*headset*) с ноутбуком, можно использовать его в офисе, автомобиле или дома.

6.2. Технология Ultra Wideband (UWB) и интерфейс Wireless USB

В основе этого интерфейса лежит концепция платформы сверхширокополосной (*Ultra Wideband, UWB*) беспроводной технологии. Классическое определение звучит так: *UWB* – это беспроводная технология, предназначенная для передачи данных на короткие – до 10 метров – расстояния, с высокой пропускной способностью (до 480 Мбит/с) и низкой потребляемой мощностью. *UWB* – это решение для беспроводной передачи высококачественного мультимедийного контента, например видео, между устройствами бытовой электроники и периферийными устройствами ПК. Одно из основных преимуществ технологии *UWB* заключается в том, что она не создает помех для других беспроводных технологий, используемых в настоящее время, – таких как *Wi-Fi*, *WiMAX* и сотовой связи.

Под современной *UWB*-технологией подразумевается применение модуляции мультиплексированием по ортогональным несущим частотам (*OFDM, Orthogonal Frequency Division Multiplexing*). Это требует использования очень широких частотных диапазонов. Схематически принцип *UWB* подразумевает генерацию передатчиком миллиардов импульсов в очень широком – порядка нескольких гигагерц – частотном спектре, перекрывающем многие уже используемые диапазоны. Приемная часть преобразовывает импульсы в данные путем отслеживания схожих последовательностей импульсов.

Однако специфический принцип модуляции и отсутствие несущей частоты приводят к своеобразному широкополосному «размазыванию» сигнала по всему спектру – что-то вроде широкополосного «белого шума», не превышающего по уровню обычные фоновые помехи. Он обладает в то же время высокой защищенностью информации из-за импульсного характера передачи.

В случае использования *OFDM* в сочетании с несколькими частотными диапазонами мы получаем технологию *MultiBand OFDM*, имеющую значительные преимущества перед вариантами со сравнительно узкой полосой, например *IEEE802.11a*, что выражается в легкой адаптации стандарта к требованиям комитетов электросвязи любых государств, возможности отличного масштабирования в будущем и обратной совместимости обновленных версий.

В общем случае под *UWB* подразумевается любая радиочастотная технология, занимающая спектр с полосой более 20 % несущей частоты

передатчика, или работающая в диапазоне более 500 МГц. Комиссия управления перспективных военных научно-исследовательских работ Министерства обороны США (*DARPA*) к сверхширокополосным относительным системам и сигналам, обладающим коэффициентом N в пределах от 0,25 до 1:

$$N = (f_b - f_n) / (f_b + f_n),$$

где f_b и f_n – верхняя и нижняя частоты диапазона соответственно.

Пакеты данных, передача которых осуществляется с использованием технологии *UWB* и применением модуляции *OFDM*, формируются по принципу транзакций *USB 2.0*, а переносятся с помощью уже известного протокола множественного доступа с разделением каналов по времени, т. е., *TDMA (Time Division Multiple Access)*.

Долгое время разработку технологий стандарта *IEEE 802.15.3a*, т. е. *UWB*, вели практически параллельно сразу две отраслевые группы – *WiMedia Alliance* и *Multi-band OFDM Alliance (MBOA)*. Множество компаний вступили в оба альянса и долгое время вели работу над обоими проектами. Кончилась вся эта история заявлением о слиянии групп *WiMedia Alliance* и *Multi-band OFDM Alliance (MBOA)* ради продолжения совместной работы над чистой версией технологии.

Частотный план *UWB* по отношению к спектрам излучения современных радиоустройств показан на рис. 6.7. На практике использование частотного диапазона 3,3 – 10,4 ГГц для *UWB* в настоящее время «легализовано» только в США: согласно ограничениям *FCC Part 15* спектральная плотность излучения не должна превышать -41 дБ·м/МГц.

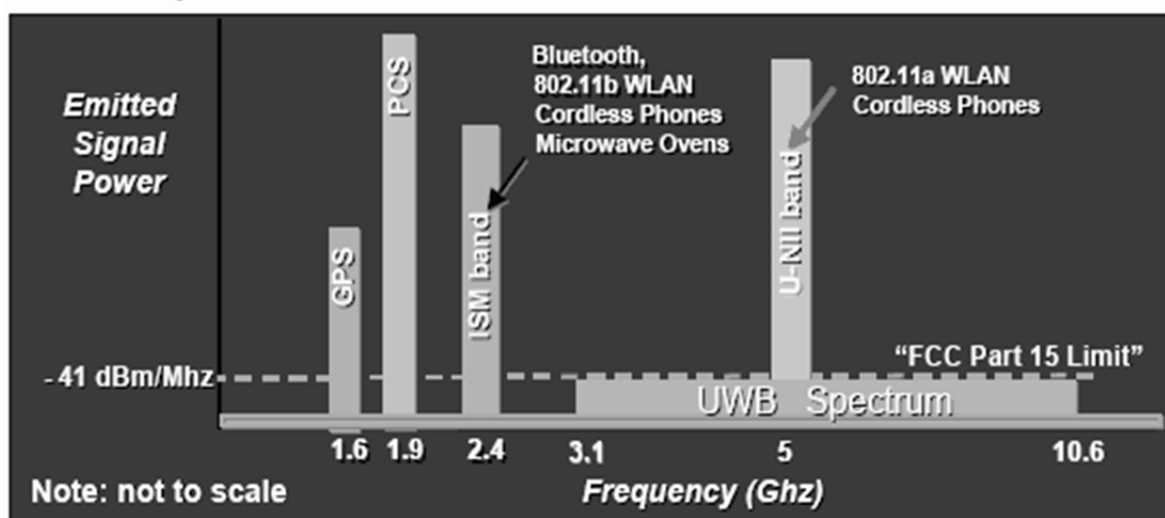


Рис. 6.7. Распределение спектра излучения ключевых беспроводных технологий

В случае со стандартом транспортного уровня *MultiBand OFDM*, на котором базируется *Wireless USB*, спектральный участок шириной 7,5 ГГц разделен на пять каналов и несколько отдельных 528 МГц поддиапазонов в каждом канале. На рис. 6.8 указаны средние частоты для каждого диапазона.

Таким образом, в результате получается 14 поддиапазонов шириной 528 МГц каждый, сгруппированных в 5 частотных участков, при этом следует особо подчеркнуть, что каждый из 14 поддиапазонов применительно к стандарту *Wireless USB* обладает возможностью поддержки обмена данными со скоростью до 480 Мбит/с.

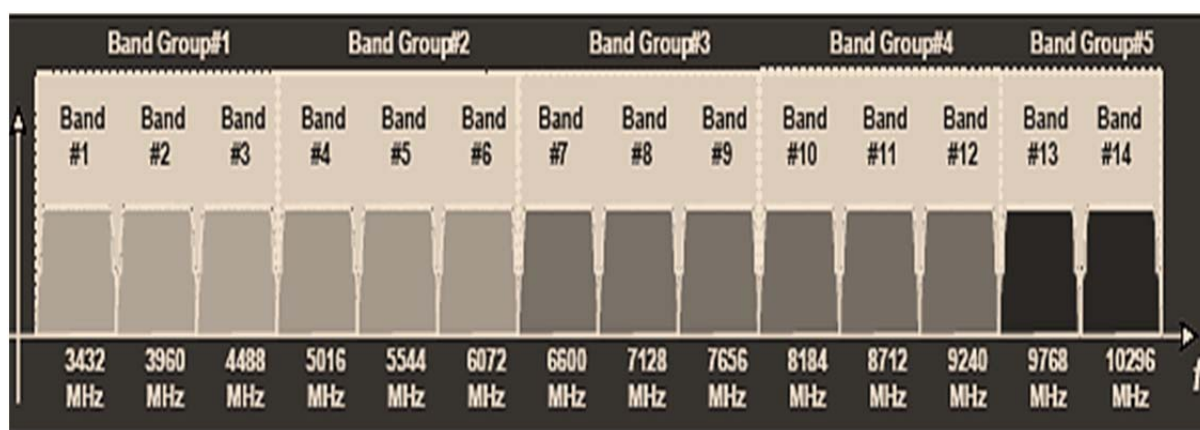


Рис. 6.8. Распределение частотного спектра *UWB*

6.2.1. Интерфейс *Wireless USB*

Радиоинтерфейсов много, но нет ни одного, который был бы «заточен» специально для обмена мощными потоками мультимедийного контента на близкие расстояния, т. е. что-то вроде проводных *USB* и *FireWire*.

Иными словами, что-то вроде *Wireless USB* рано или поздно должно было возникнуть на основе технологии, использующей сверхширокополосную модуляцию (*UWB, UltraWideBand*) с низкой спектральной плотностью сигнала. В настоящее время продвижением *Wireless USB* занимается индустриальная группа *Wireless USB Promoter Group*. Место *Wireless USB* в современном и будущем «беспроводном мире» показано на рис. 6.9.

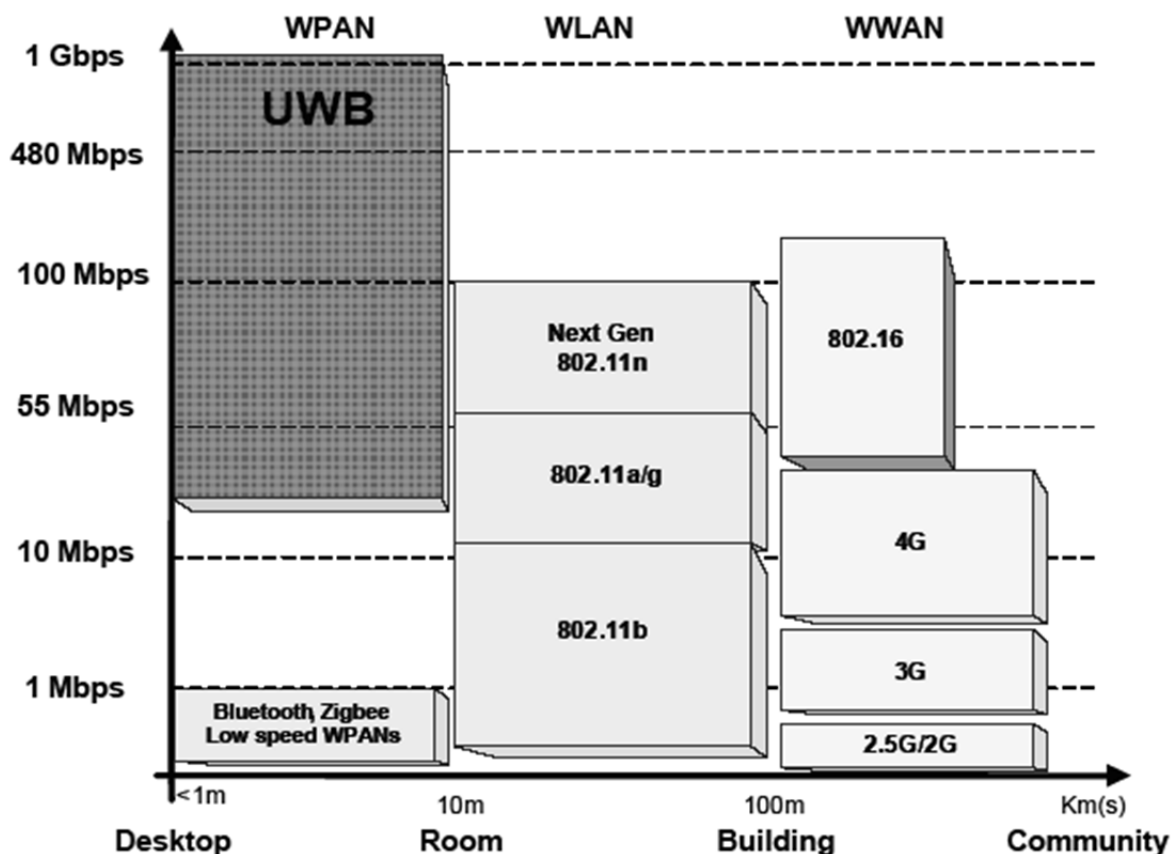


Рис. 6.9. Место *Wireless USB* в современном и будущем «беспроводном мире»

Стандарт *Wireless USB* подразумевает использование двух основных «слоев» для обмена данными – транспортного и физического уровня. Транспортный уровень как раз и базируется на сверхширокополосной (*UWB*) технологии, физический представляет собой уровень формирования среды передачи данных, где помимо *WUSB* с легкостью могут фигурировать *W1394 (Wireless FireWire)*, *Bluetooth* и прочие, к настоящему времени еще не изобретенные и не сформулированные протоколы. Просто *Wireless USB* стал одной из первых *UWB*-технологий, доведенных до состояния коммерческого стандарта.

6.2.2. Топология *Wireless USB*

По аналогии с проводным *USB*, устройства *Wireless USB* обладают собственным адресом, получаемым при подключении или перечислении. Каждое устройство *Wireless USB* поддерживает один или несколько каналов для связи с хостом.

Каждое устройство *Wireless USB* может работать как *MAC Layer* устройство. Стандарт *Wireless USB* описывает три категории устройств, представляющие разные степени реализации механизма *MAC Layer*, от «автосигнального» (*self beaconing*) типа до полного отсутствия его поддержки.

Базовыми элементами инфраструктуры *WUSB* являются концентратор и радиальные линии. В такой топологии (рис. 6.10) хост-контроллер инициирует любой обмен данными между подключенными к нему устройствами, выделяя временные интервалы и полосу пропускания каждому подключенному устройству. Подобная группа называется кластером. Описанные соединения относятся к типу «точка-точка» и осуществляются между *WUSB*-хостом и *WUSB*-устройством.

WUSB-хост с логически подключенными к нему *WUSB*-устройствами (максимально до 127) образует неформальный *WUSB*-кластер. Описываемая топология будет поддерживать модель двойного применения, при которой устройство может в ограниченном объеме выполнять функции хоста. Такая модель позволяет мобильным устройствам пользоваться сервисами, которые обеспечивает центральный хост (например, принтерами и устройствами отображения). Кроме того, эта модель позволяет устройству получить доступ к данным, расположенным за пределами кластера, к которому в текущий момент подключено это устройство. Для этого устройство должно создать второй кластер, выступая в качестве хоста с ограниченными возможностями.

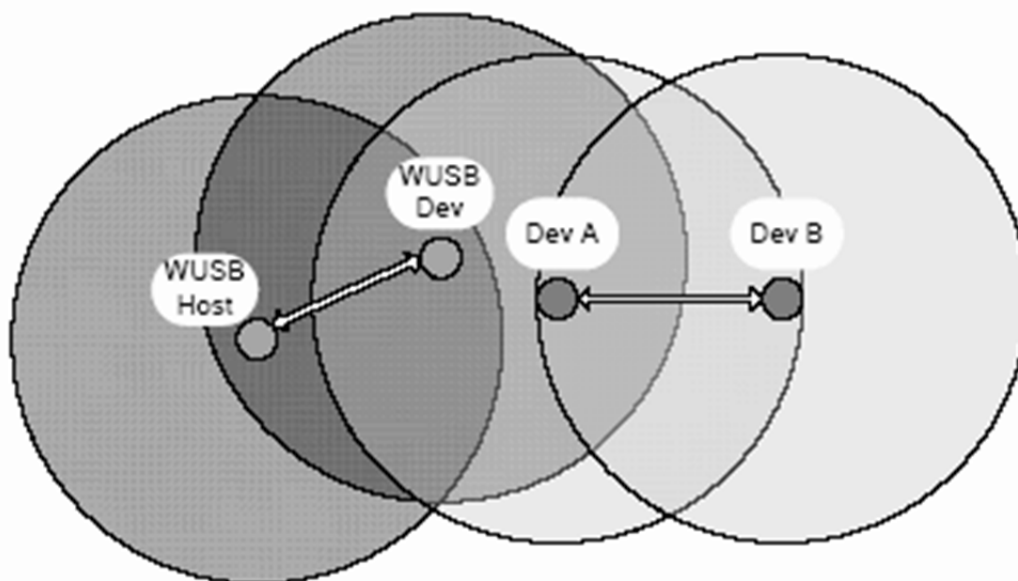


Рис. 6.10. Топология *Wireless USB*

6.2.3. Частотный спектр WUSB. Физический уровень MBOA PHY

Из пяти каналов 7,5 ГГц диапазона *MultiBand OFDM* согласно требованиям *MBOA* поддержка самого первого канала *Channel 1*, включающего в себя три первых частотных участка, является обязательным требованием для всех *UWB* устройств. Использование диапазонов в каналах со второго по пятый является необязательным.

Смысл именно такого деления на каналы и поддиапазоны не случаен и является подходящим сразу по нескольким причинам. Первая из них заключается в том, что подобная частотная организация обеспечивает поддержку одновременно до четырех частотно-временных режимов модуляции на канал или в сумме до 20 каналов для нынешнего деления *MB-OFDM*. На практике предполагается, что *Wireless USB* устройства первого поколения будут работать только в первом частотном участке (*Band#1 - Band#3*).

Наконец, разработчиками стандарта учтена даже дифференциация назначения разных диапазонов для работы с разными приложениями. Так, использование «низкочастотных» диапазонов *MB-OFDM* может оказаться предпочтительным в случае обмена данными на большие расстояния, и, наоборот, высокочастотные диапазоны будут отданы для коротких расстояний.

Использование идеи канальной диверсификации позволит добиться надежной передачи данных и высокой помехоустойчивости плюс использовать и небольшую мощность передатчика целенаправленно в более узком участке.

Базовым элементом – «квантом» – для *UWB*-обмена является *OFDM-Symbol* – *OFDM*-посылка данных со стандартной длиной 312,5 нс. (рис. 6.11).

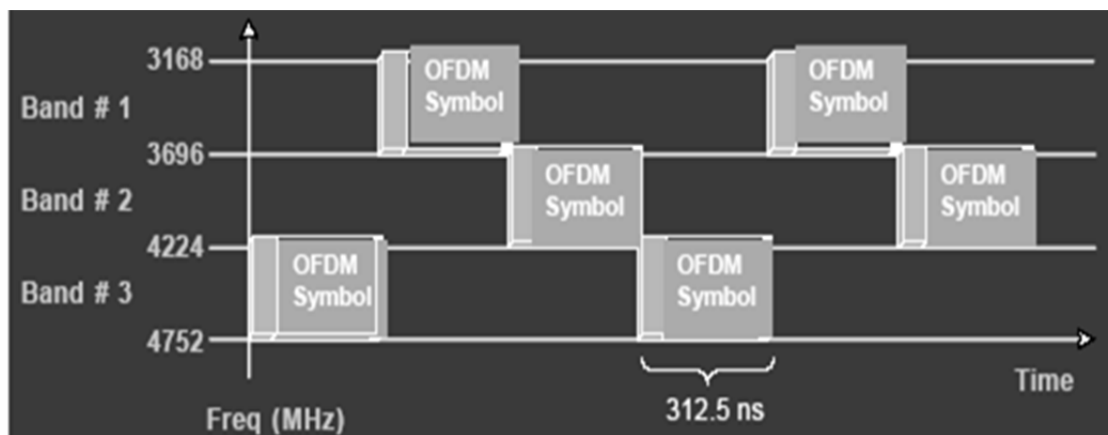


Рис. 6.11. Распределение *Symbol-OFDM* по каналам

Каждая *OFDM*-посылка «шириной» 4 МГц вмещает в себя 100 «тональных» посылок данных. Формат посылок остается неизменным для любой скорости обмена данными, изменяется только «тональность» модуляции в посылках и между ними, что как раз и отражает различные битрейты потока данных и различные уровни устойчивости передачи.

Шесть последовательных посылок формируют в итоге базовый пакет продолжительностью 1,875 мкс (рис. 6.12), который, в свою очередь, впоследствии преобразовывается в конкретные биты данных.

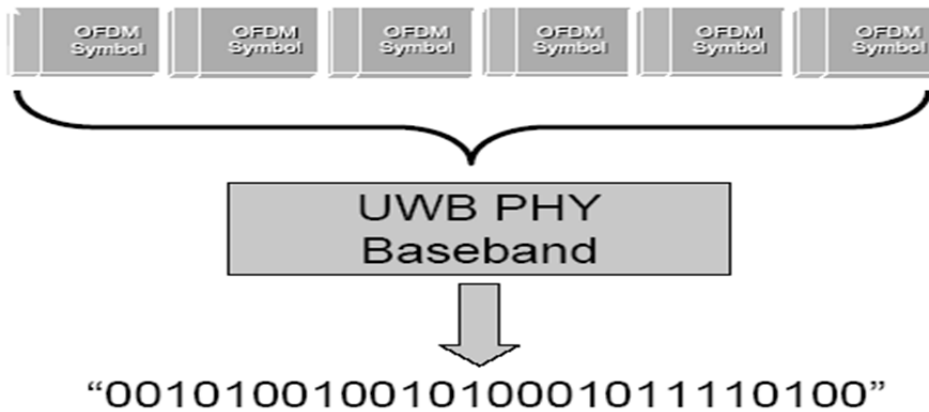


Рис. 6.12. Структура *OFDM-Symbol*

Частотно-временные кодеки (*TFC, Time Frequency Codecs*) позволяют формировать и разносить пакеты как по нескольким диапазонам канала, так и передавать их в едином диапазоне (рис. 6.13).

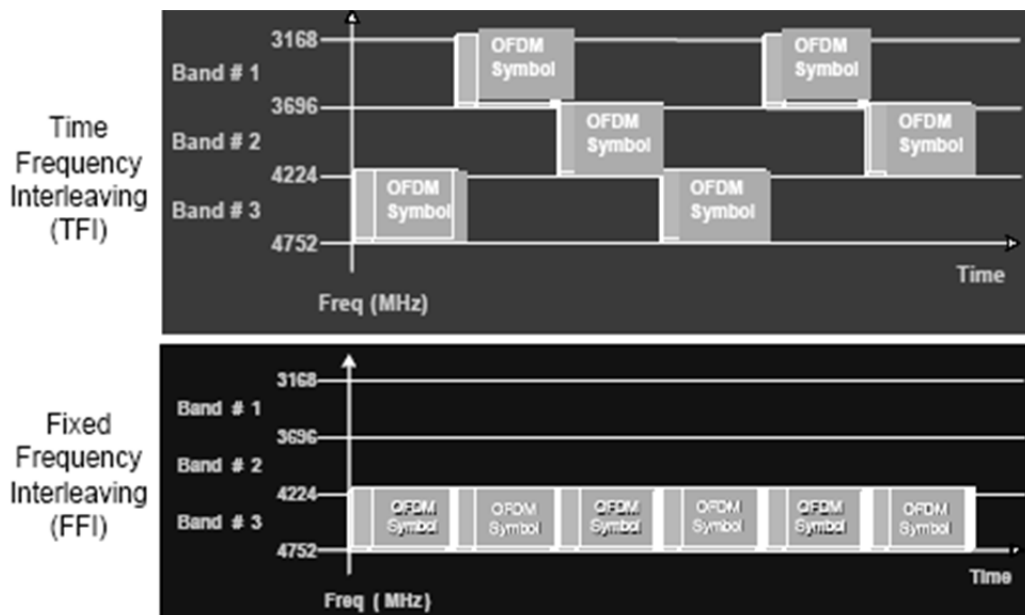


Рис. 6.13. Распределение *OFDM-Symbol* по нескольким диапазонам канала (вверху) и в едином диапазоне (внизу)

Каждый *UWB*-пакет формируется из определенного количества компонентов, но конечная его длина зависит от скорости обмена данными.

6.2.4. Канал *Wireless USB*: формирование транзакций

Каждый канал *Wireless USB Channel* инкапсулирован в канал *MBOA* с использованием *PHY* (кадры, фреймы) и *MAC* (заголовки, шифрование и пр.) компонентов *MBOA*. Использование *MBOA PHY/MAC* совместимых компонентов подразумевает использование сигнальной и фреймовой архитектуры *MBOA PHY* в сочетании с системой заголовков, защитной инкапсуляцией *MAC etc.*

Канал *Wireless USB* представляет собой продолжительную последовательность связанных управляющих пакетов данных, излучаемых хостом в оговоренных стандартом временных рамках. Временные рамки канала *WUSB* оговорены в спецификациях каналов *MBOA* и формируются управляющими командами *MMC – Micro-schedule Management Commands*.

Как уже было отмечено выше, *PHY* в конечном итоге позволяет создавать 64 КБ пакеты данных, так называемые суперфреймы, которые формируются с помощью хорошо известного протокола множественного доступа с разделением каналов по времени – *TDMA (Time Division Multiple Access)*, весьма схожего с используемым в проводном *USB 2.0* (рис. 6.14).

Обмен данными инициализирует хост-контроллер. В полной аналогии с проводным *USB* каждый трансфер (транзакция) состоит из трех логических пакетов: маркера (*token*), данных (*data*) и пакета, определяющего параметры передачи (*handshake*). Для повышения эффективности физического уровня обмена данными хост комбинирует несколько маркеров в единый пакет, в котором определяются режим приема данных (*OUT*), отправки данных (*IN*) или пакет *handshake*.

Передача данных с использованием протокола *Wireless USB* основана на тех же типах трансферов, что и проводной *USB*. Однако в связи с повышенной возможностью возникновения ошибок протокол *Wireless USB* определяет несколько иной тип изохронной передачи данных, включающий пакеты *handshake* для подтверждения доставки данных и некоторое буферирование, позволяющее повысить надежность работы изохронного канала.

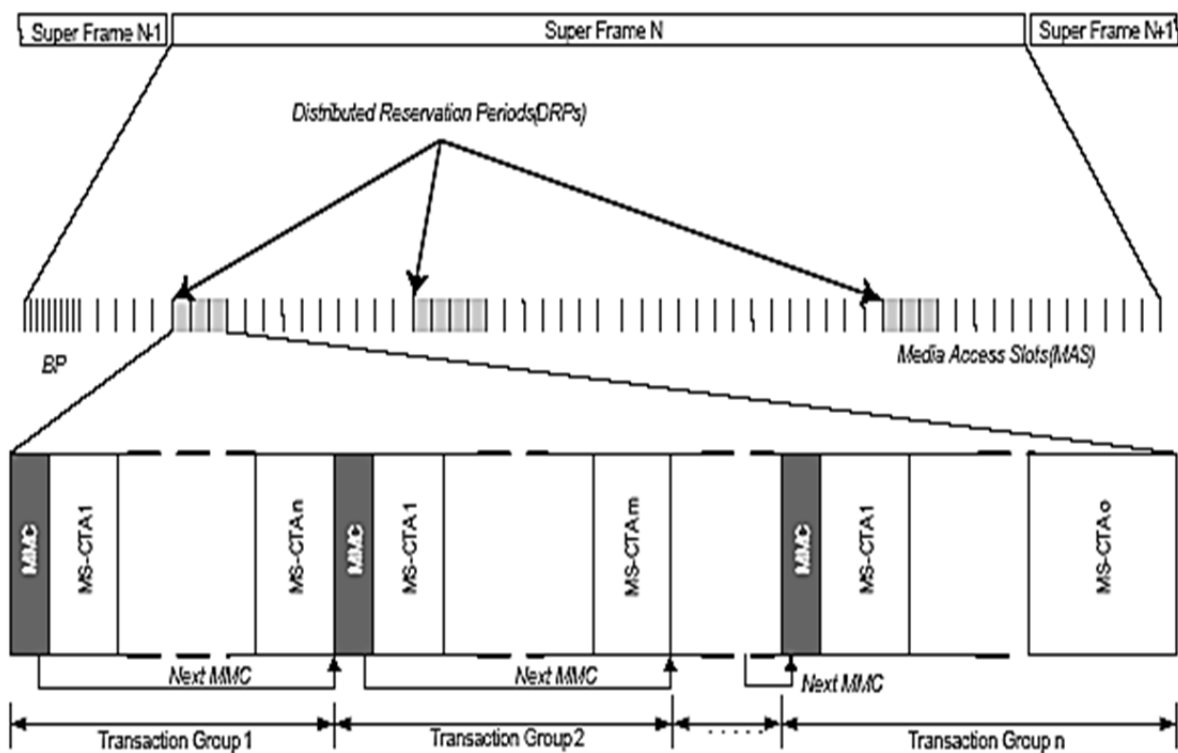


Рис. 6.14. Протокол на базе TDMA

Базовая структура, используемая для реализации протокола *Wireless USB*, представляет собой последовательность команд MMC с набором заключенных в ней информационных элементов, плюс фазу идентификации до появления следующей команды MMC (рис. 6.15).

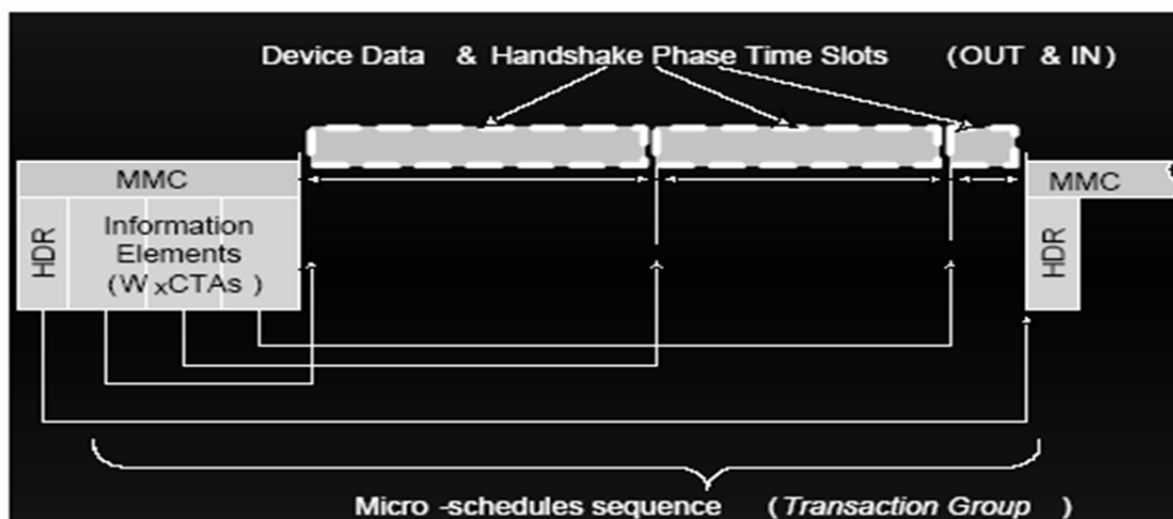


Рис. 6.15. Группа транзакций WUSB

Аналогия между классическим *USB* и *WUSB* показана на рис. 6.16.

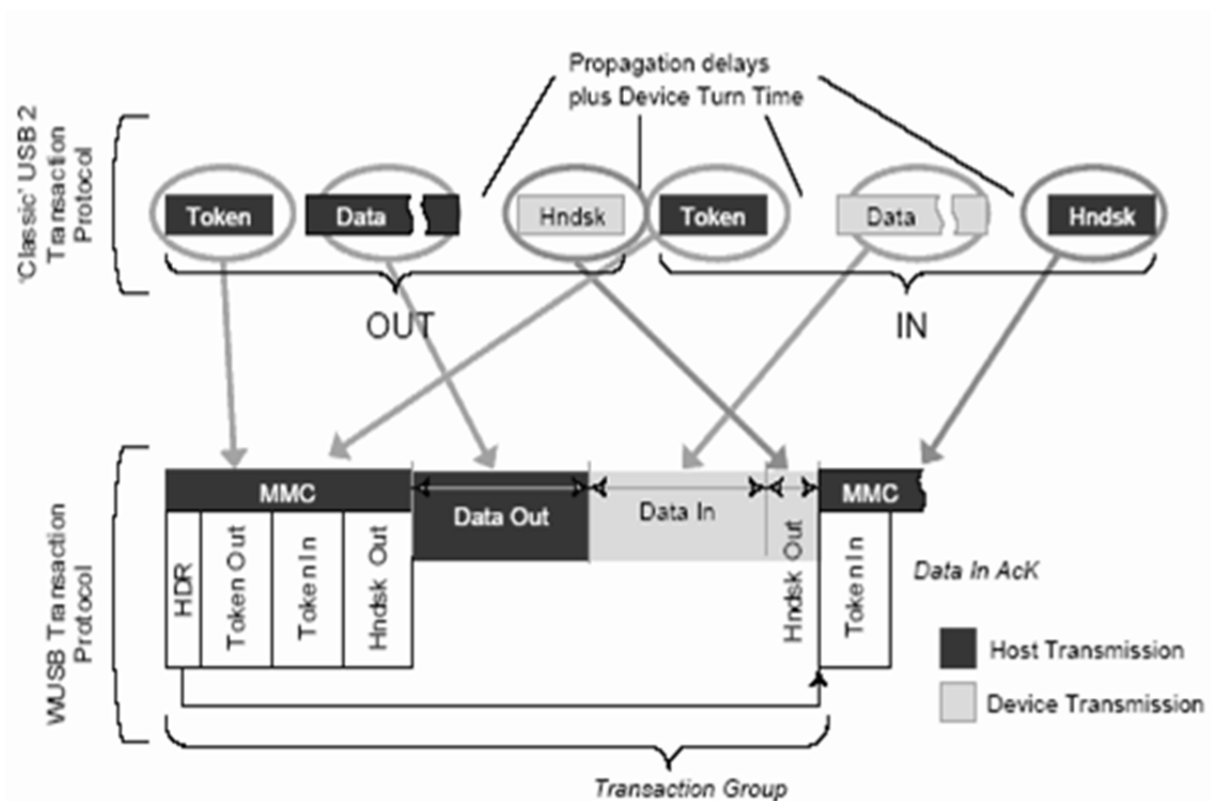


Рис. 6.16. Аналогия между классическим *USB* и *WUSB*

Памятуя о задаче сохранения максимальной обратной совместимости с традиционным проводным интерфейсом *USB 2.0*, разработчики постарались импортировать организацию транзакций по его образу и подобию, включая сигнальные события (соединение, разъединение, временное прекращение обмена, возобновление и т.д.), особенности построения протокола организации транзакций и тому подобное, что в целом удалось на достаточно высоком уровне. Из уникальных идентификаторов *WUSB* стоит отметить индекс *MSSI* (*Micro-Scheduled Stream Index*), означающий использование именно *Wireless USB*, а также регистра *V_OUI*, индицирующего возможность работы устройства в качестве хоста.

В целом подобная организация обмена данными позволяет передавать несколько пакетов данных за одну транзакцию, таким образом увеличивая эффективность работы интерфейса (рис. 6.17).

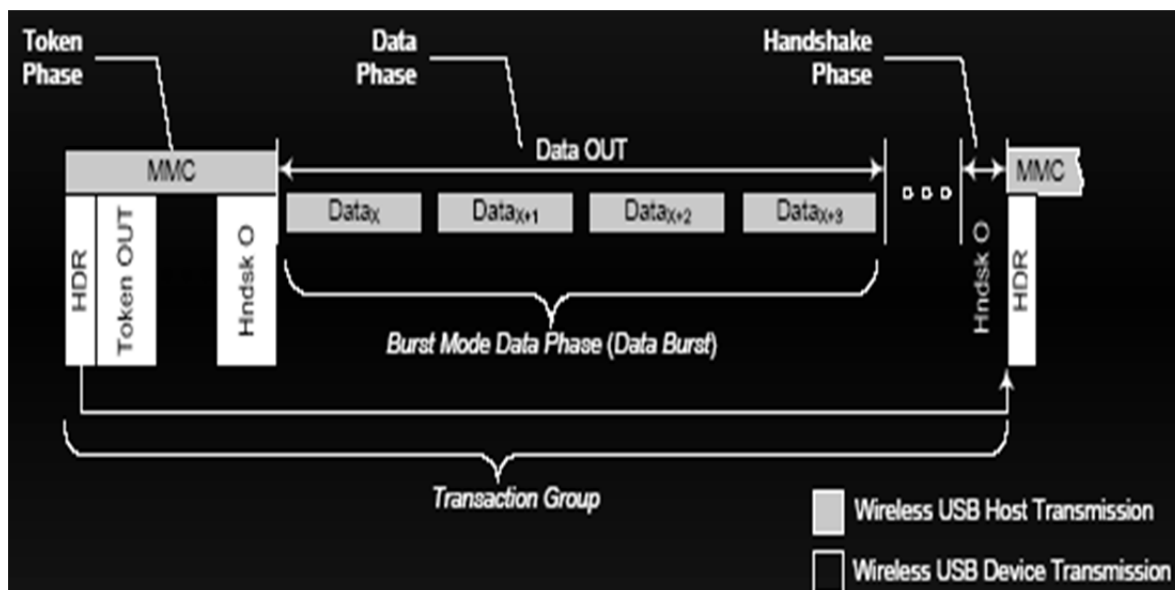


Рис. 6.17. Организация передачи пакетов данных протокола *Wireless USB*

6.2.5. Управление энергопотреблением

Каждое *Wireless USB* устройство, равно как его драйверы, будет обладать собственной системой управления энергопотреблением, без перекладывания этой проблемы «на голову» хост-контроллера.

Всего предполагается использовать три схемы экономии энергии:

1. Сохранение энергии в режиме нормального обмена данными: прекращение излучения в промежутках между *MMC*-посылками и везде, где это имеет смысл в текущий момент.

2. Спящий режим: увеличение промежутков опроса устройства на предмет наличия канала; хост-контроллер перестает генерировать трафик.

3. Разъединение (с ведома хост-контроллера).

При этом доступно два состояния канала: когда работоспособность *WUSB* канала поддерживается и экономия энергии происходит в рабочем режиме, или когда *WUSB* канал разорван, работа системы временно приостановлена (*S3* и глубже), хост оповещает устройство о временном приостановлении обмена.

Для первого поколения устройств *Wireless USB* пиковое потребление мощности *PHY* ограничено уровнем 130 – 160 мВт, далее ожидается ужесточение этого показателя.

6.2.6. Производительность *Wireless USB*

Основной плюс беспроводного интерфейса *Wireless USB* – мгновенная эффективная масштабируемость трафика. В зависимости от расстояния между хостом и устройством скорость обмена данными может изменяться в пределах от 53,3 до 480 Мбит/с. Масштабирование происходит примерно таким образом:

- 106,7 Мбит/с на расстоянии до 10 м в реальном многозадачном окружении;
- 200 Мбит/с на расстоянии не менее 4 м в реальном многозадачном окружении;
- До 480 Мбит/с на расстоянии не менее 2 м в реальном многозадачном окружении.

6.2.7. Безопасность *Wireless USB*

По словам разработчиков стандарта, технология *Wireless USB* в перспективе будет обладать очень надежной защитой трафика от несанкционированного доступа на уровне проводного стандарта *USB 2.0* (рис. 6.18). На практике в первом поколении *Wireless USB* использовано шифрование *AES-128* с применением *CBC-MAC (CCM)*.

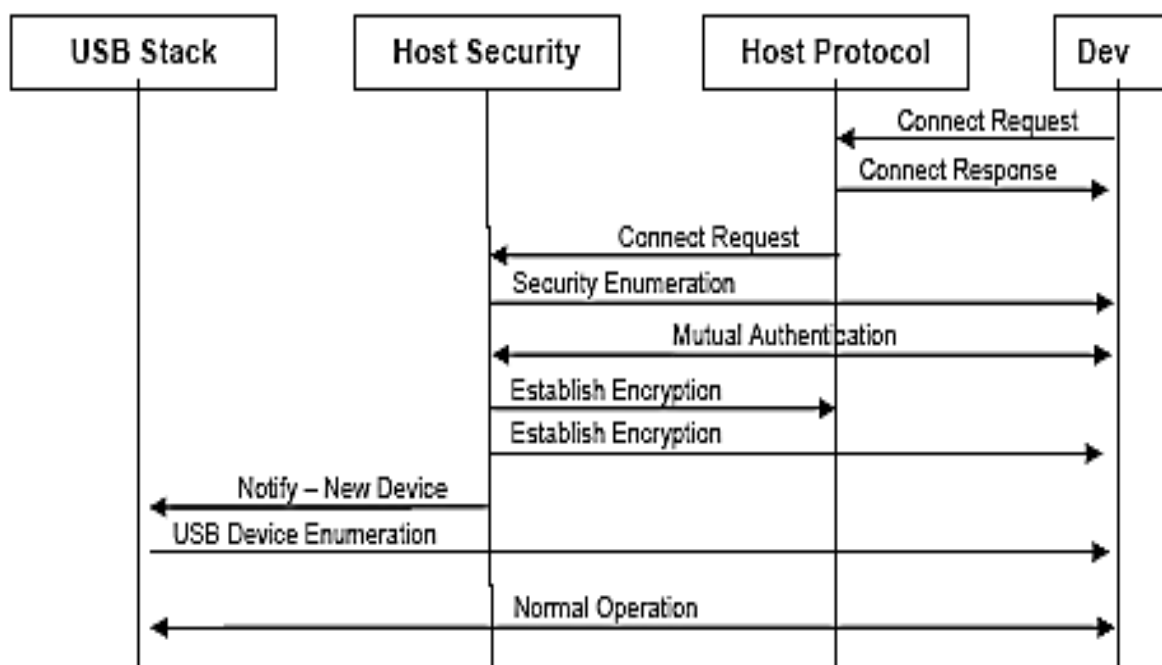


Рис. 6.18. Структура защиты информации в *Wireless USB*

Wireless USB также поддерживает шифрование с открытыми ключами, но только для аутентификации. Устройства могут инициализировать первичное соединение с использованием открытых ключей. Шифрование с применением открытых ключей может использовать тот же уровень шифрования и даже более защищенный – *RSA* с 3072-битным ключом и хэшем *SHA-256*. Более высокие уровни шифрования пока будут использоваться на уровне приложений.

Стоит отметить, что архитектура шифрования при смешанных проводных *USB/Wireless USB* соединениях также предполагает шифрование трафика, проходящего через проводные соединения. Это позволяет избежать путаницы и ошибок при сортировке трафика на проводной/беспроводной.

6.2.8. Программная реализация *Wireless USB*

Компания *Microsoft*, стоявшая у истоков создания спецификаций *Wireless USB*, как никто другой заинтересована в реализации поддержки стандарта на уровне платформы.

Программная хост-архитектура реализации поддержки *UWB* включает в себя:

- поддержку решений под шины *PCI* и *PCI Express*, подразумевающие варианты в виде интерфейсных слотовых карт и автоматически вытекающая из этого поддержка версий под *CardBus* и *ExpressCard*;

- поддержку *WUSB* решений со стандартными интерфейсными разъемами «проводного *USB*» (*USB Dongles*).

Программный арбитраж и управление *UWB*-трафиком осуществляют взаимодействие между устройствами, поддерживают связь с другими хостами на предмет оптимизации скорости обмена данными, управляют *PCI*-ресурсами и контролируют топологию.

Нововведения, связанные с поддержкой *WirelessUSB* нынешними и будущими операционными системами *Windows*, заключаются в дополнении новыми компонентами. Новые компоненты помечены более светлым цветом на рис 6.19.

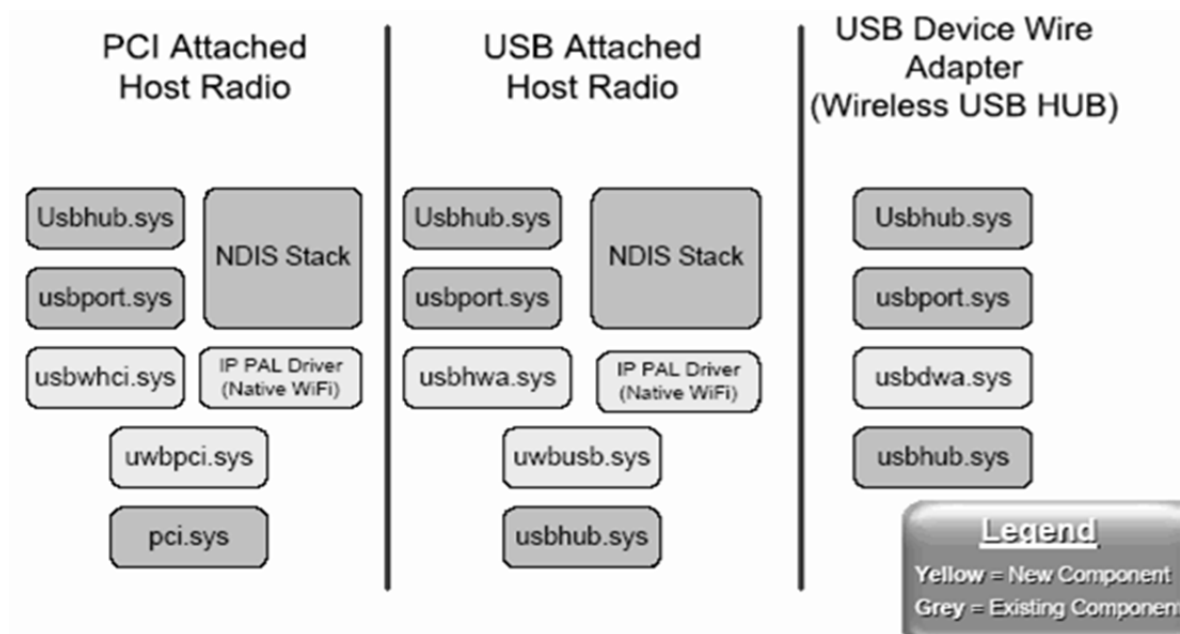


Рис. 6.19. Структура программного обеспечения *Wireless USB* в ОС *Windows*

Таким образом достигается абсолютная прозрачность работы операционной системы с *WUSB* устройствами: в конечном итоге ей совершенно безразлично, используется ли *EHCI (USB 2.0)* или *WHCI (Wireless USB)* контроллер, на практике *Wireless USB* соединение воспринимается операционной системой как обычное проводное *USB* соединение.

6.3. Инфракрасный порт

InfraredDataAssociation – IrDA, ИК-порт. Инфракрасный порт описывается группой стандартов, включающей протоколы физического и логического уровня передачи данных с использованием инфракрасного диапазона световых волн в качестве носителя. Он является разновидностью атмосферной оптической линии связи ближнего радиуса действия.

Эта технология была особо популярна в конце 1990-х – начале 2000-х годов; в настоящее время практически вытеснена более современными способами связи, такими как *WiFi* и *Bluetooth*. Вопреки распространенному мнению, основной причиной отказа от *IrDA* была вовсе не низкая скорость передачи данных, а ограниченная дальность действия и требования прямой видимости пары приемник – передатчик. Скоростные возможности, напротив, до сих пор в несколько раз

превышают, например, возможности последней на сегодняшний момент версии протокола Bluetooth (спецификация 2.0).

Спецификации включают в себя протоколы *IrPHY*(*SIR*, *MIR*, *FIR*, *VFIR*, *UFIR*), *IrLAP*, *IrLMP*, *IrCOMM*, *Tiny TP*, *IrOBEX*, *IrLAN*, *IrSimple* и *IrFM* (находится в разработке).

6.3.1. Особенности интерфейса

Аппаратная реализация, как правило, представляет собой пару из передатчика, в виде светодиода, и приемника, в виде фотодиода расположенных на каждой из сторон линии связи. Наличие и передатчика, и приемника на каждой из сторон является необходимым для использования протоколов гарантированной доставки данных.

В ряде случаев, например при использовании в пультах дистанционного управления бытовой техникой, одна из сторон может быть оснащена только передатчиком, а другая только приемником.

Иногда устройства оснащают несколькими приемниками, что позволяет одновременно поддерживать связь с несколькими устройствами. Использование при этом одного передатчика возможно благодаря тому, что протоколы логического уровня для обеспечения гарантированной доставки данных требуют лишь незначительного обратного трафика.

Наличие нескольких передатчиков встречается гораздо реже.

Большинство оптических сенсоров, используемых в фото- и видеокамерах, имеет диапазон чувствительности гораздо шире видимой части спектра. Благодаря этому работающий инфракрасный передатчик можно увидеть на экране или фотоснимке в виде яркого пятна.

Диоды

В качестве источника и приемника ИК-излучения используются диоды. Излучающий светодиод (рис. 6.20, а) обеспечивает угол излучения 30° . Пин-диод (фотодиод) (рис. 6.20, б) принимает излучение в секторе 15° .

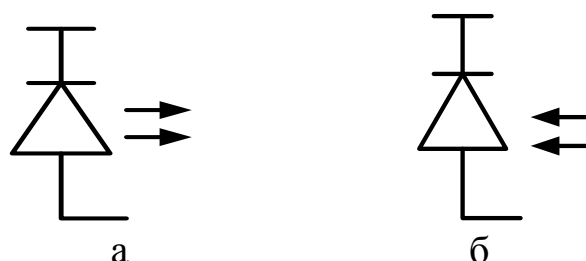


Рис. 6.20. Диоды

6.3.2. Параметры интерфейса

Длина надежной связи не более 10 м. Тип связи – полудуплексный. Этот последовательный порт поддерживает три скорости:

- низкую $U = 115,2$ Кб/с;
- среднюю $U = 115,2$ Кб/с;
- высокую $U = 4$ Мб/с.

Стандарты:

1. *Sharp – IR*. Режим поддерживает двунаправленную связь, используя *DASK*. Скорость – до 38,4 Кб/с.
2. *IrDA – 1.0 SIR* – двунаправленная связь. Позволяет передавать данные со скоростью 115,2 Кб/с (низкая скорость). Стандарт фирмы *HP*.
3. *IrDA – 1.1 MIR* – средняя скорость, *FIR* – высокая скорость.
4. Стандарт для бытовой аппаратуры.

6.3.3. Возможности

В повседневной жизни мы постоянно сталкиваемся с ИК-портами каждый день.

Дистанционный пульт управления передает команды на телевизор или видеомаягнитофон с помощью *IrDA* (рис. 6.21). Сейчас ИК-портами все ещё оснащаются большинство мобильных телефонов, ноутбуков и карманных компьютеров. ИК-портами оснащаются некоторые принтеры и цифровые фотоаппараты. Большинство настольных ПК, напротив, не имеет инфракрасного порта в стандартной системной конфигурации, и для них необходим ИК-адаптер, который подключается к компьютеру через *USB*, *COM*-порт или в специальный разъем на материнской плате.

Через ИК-порт с помощью протокола высокого уровня *IrOBEX* можно, например, передать цифровую визитную карточку, мелодию, картинку или файл на другой мобильник или компьютер, на котором также имеется ИК-порт. Этот же протокол позволяет организовывать синхронизацию данных.

Типовая блок-схема организации *IrDA*-канала показана на рис. 6.21, а архитектура порта – на рис. 6.22.

Протокол *IrCOMM* позволяет использовать мобильный телефон как беспроводной модем.

Протокол *IrLAN* позволяет подключить и связать устройства в локальную сеть наподобие *Ethernet*. Ввиду того что пульты дистан-

ционного управления используют этот же протокол, КПК со встроенным ИК-портом можно использовать как пульт для управления. Для этого, как правило, необходимо установить соответствующее ПО.

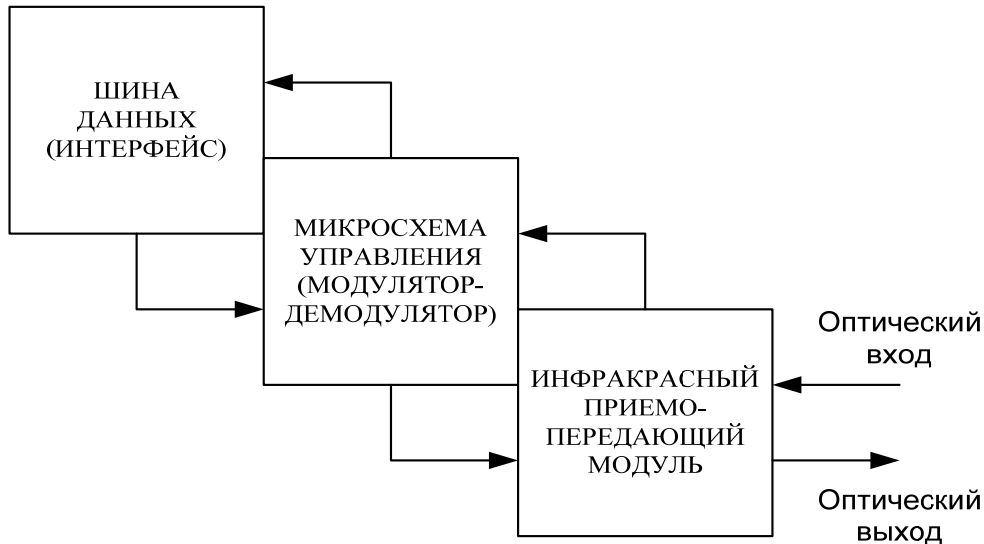


Рис. 6.21. Типовая блок-схема организации IrDA-канала

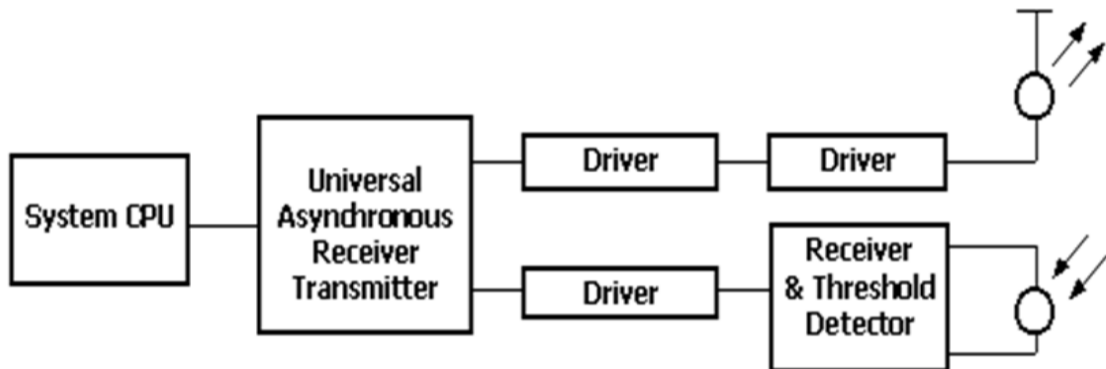


Рис. 6.22. Архитектура порта IrDA

Рис 6.23. Формат данных IrDA

6.3.4. Протоколы

1. Физический уровень. Его временная диаграмма представлена на рис. 6.23. Тактовая частота задается 24 МГц (тактовая частота генератора = 1МГц). Время импульса 1,63 мкс = 3/16 бита.

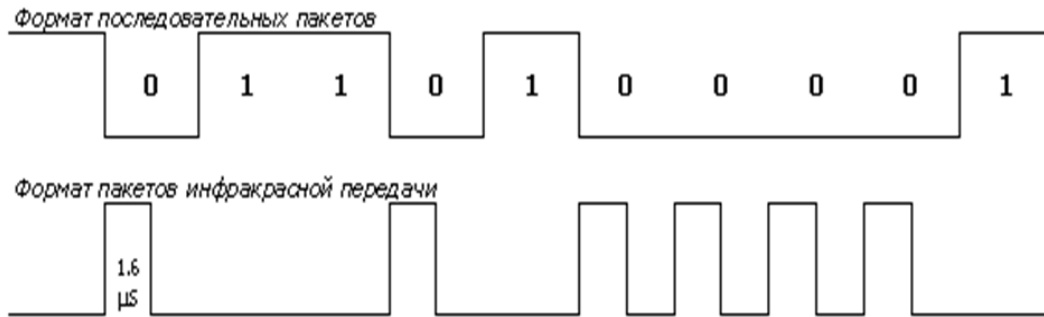


Рис. 6.23. Формат данных IrDA

В режимах *MIR* и *FIR* есть понятие кадра (рис. 6.24). Начало и конец отмечаются флагами. Для контроля целостности информации используются CRC-коды: *MIR* – 16-битный, *FIR* – 32-битный (позиционно-импульсный код для любой пары бит). Структура заголовков пакетов зависит от длины кадра и представлена на рис. 6.25). В простейшем случае для контроля вводится бит четности (рис. 6.26).

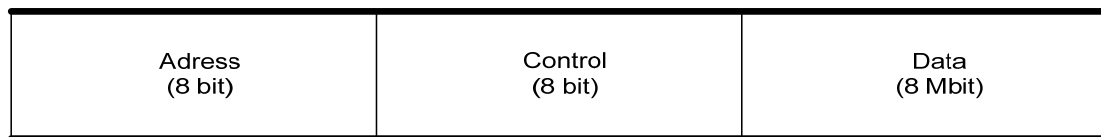
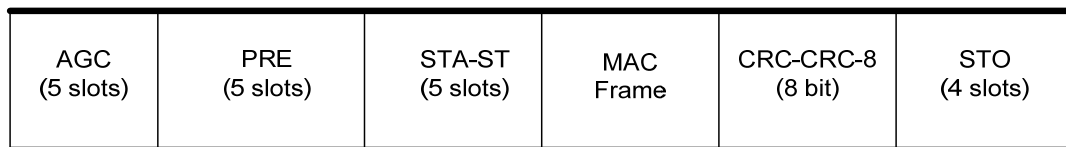


Рис. 6.24. Структура фрейма (кадра)

Short Frame Structure (72 б)



Long Frame Structure (776 б)

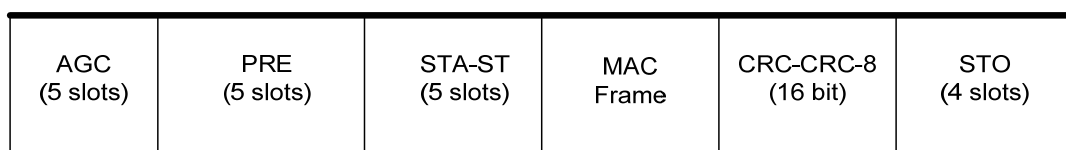


Рис. 6.25. Виды пакетов

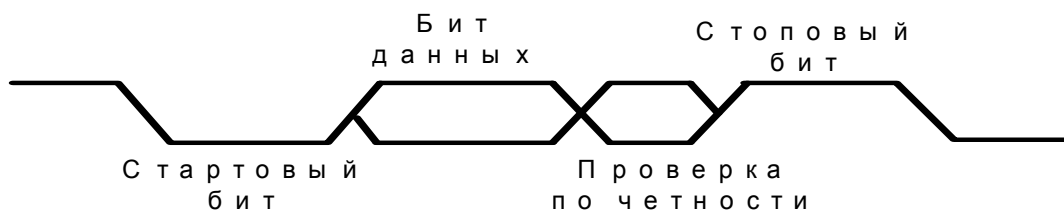


Рис. 6.26. Временная диаграмма физического уровня

2. *IrLAP* – ИК-модификация протокола *HDLC* для ИК-связи. Протокол описывает процедуру установления, нумерации и закрытия соединения.

3. Протокол установления соединения *IrLMP*. В соответствии с ним устройство сообщает о своем присутствии. Он позволяет обнаруживать сервисы, предоставляемые другими устройствами, проверять потоки данных и выступать в роли мультиплексора.

4. Транспортный протокол *TinyTP* обслуживает внутренние каналы, производит установку данных в пакеты и сборку данных из пакетов.

5. *IrCOMM* позволяет через ИК-связь эмулировать:

– обычное 3-проводное подключение (аналог *COM*-порта) с использованием сигналов *TXD*, *RXD*, *GND* по протоколу программного управления связью;

– 9-проводное подключение *RS-232C*.

6. *IrLAN* обеспечивает доступ к локальным сетям *Ethernet*, *Token-Ring*. Требуется устройствам с интерфейсом *IrDA*, подключенным к сети.

7. Протокол объектного обмена (высокий уровень). *IrOBEX* определяет команды ввода-вывода и имеет расширение для мобильной связи.

6.3.5. Контроллеры и приемопередатчики ИК- порта

Для ИК-связи используются специальные контроллеры и приемопередатчики. Приемопередатчики могут быть нескольких видов.

Внутренние приемопередатчики для низкой скорости подключаются с помощью схемы *UART* (рис. 6.27).

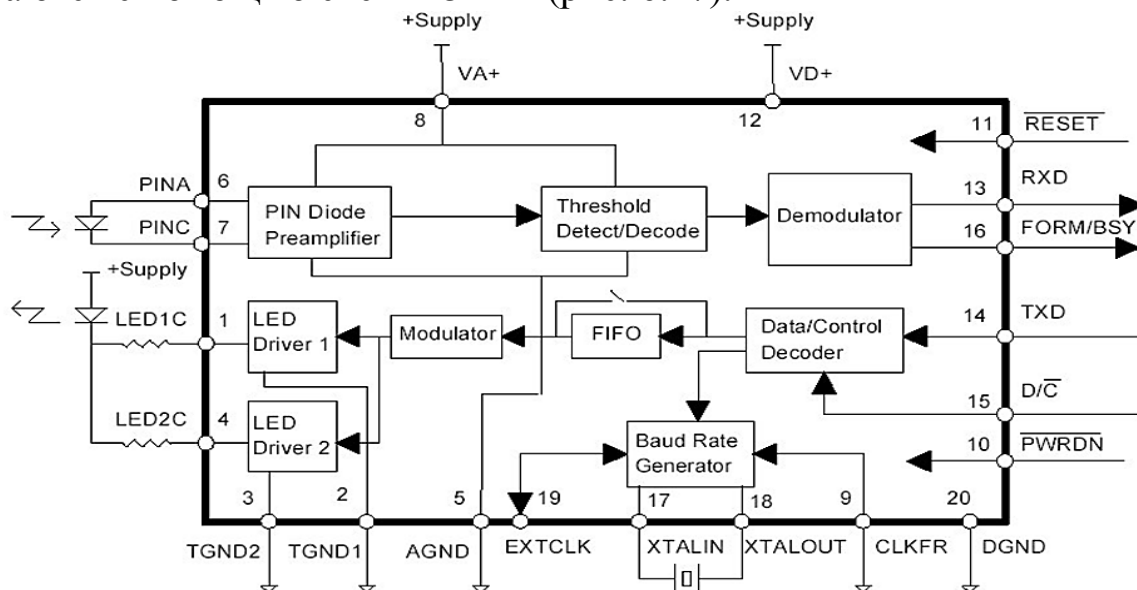


Рис. 6.27. Функциональная схема ИК-приемопередатчика

Для средней и высокой скорости используется специализированная схема контроллеров, которая обеспечивает прямое обращение к шине (*PC87109*). Структурная и функциональная схемы контроллеров приведены на рис. 6.28 и 6.29. Контроллер выполнен в виде карты расширения или интегрируется в системную плату. Приемопередатчик подключается напрямую или через специальный разъем.

Внешние ИК-адаптеры выполняются с интерфейсом *RS-232C* или с шиной *USB*.

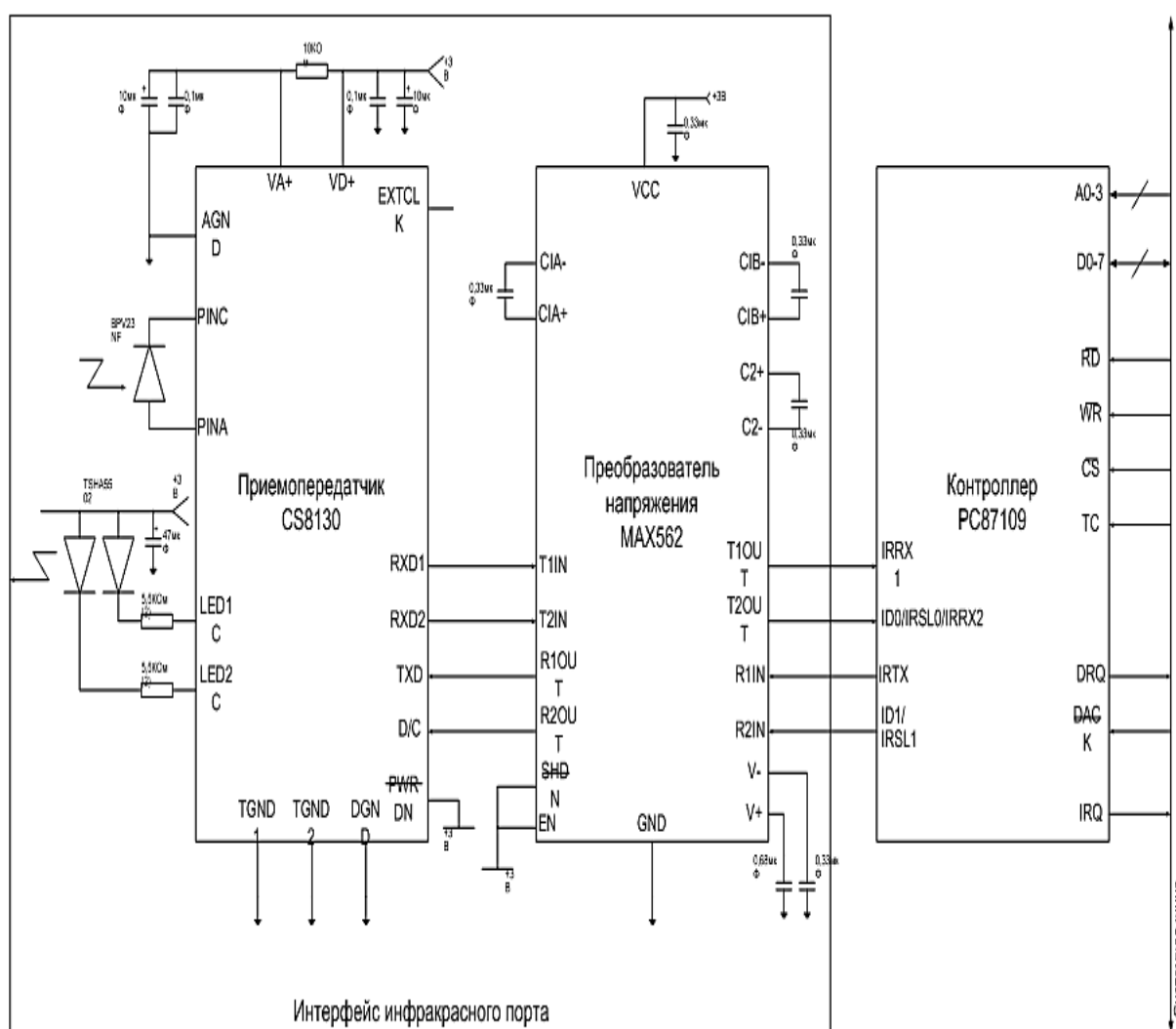


Рис. 6.28. Функциональная схема контроллера ИК-порта

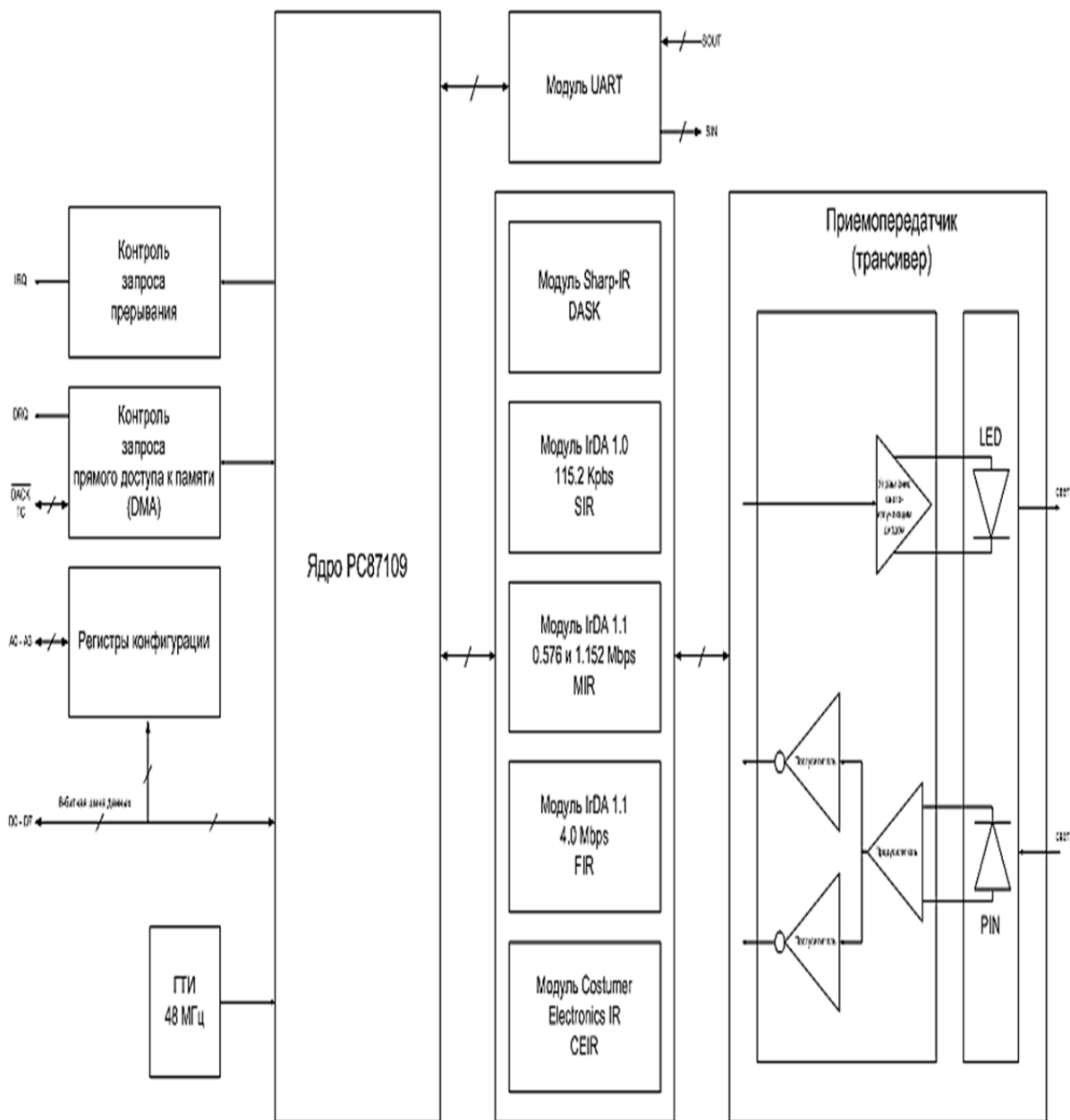


Рис. 6.29. Структурная схема контроллера ИК-порта

Системная поддержка

Системная поддержка ИК-портов позволяет обеспечить:

- связь между компьютерами;
- вывод на печать;
- соединение с ЛВС;
- синхронизацию данных;
- подключение сотовых телефонов;
- подключение цифровых фотоаппаратов и видеокамер.

6.3.6. Преимущества технологии беспроводной передачи в инфракрасном диапазоне

Использование ИК диапазона (или неиспользование радиодиапазона) обеспечивает следующие преимущества:

– загруженность и засоренность радиоэфира приводит к тому, что в крупных городах получить частотную полосу становится весьма проблематичным, а вседоступность «открытых» диапазонов не может гарантировать качества канала в коммерческих и служебных системах связи, несмотря на использование технологий передачи со скачком частоты и сложным цифровым кодированием;

– высокая конфиденциальность связи. Передача осуществляется узким лучом при полном отсутствии боковых излучений;

– отсутствие необходимости в разрешениях на использование радиочастотного спектра часто служит определяющим фактором при выборе оборудования передачи;

– главное преимущество – отсутствие принципиальных сложностей в ИК технологии с пределом скорости передачи. Если в радиочастотных системах для занятия разумной ширины полосы передачи приходится применять изощренное кодирование, которое к тому же снижает другие характеристики системы (к примеру, отношение сигнал/шум в приемнике), то все эти сложности не имеют никакого отношения к инфракрасным системам. Скоростные характеристики канала передачи в ИК системах в основном определяются техническими характеристиками модулирующих усилителей и частотными свойствами фотодиодов! Но технология, как известно, развивается весьма бурными темпами. Уже сейчас, когда самой старой коммерческой беспроводной ИК системе вряд ли будет 12 лет, скорости достигли отметки 2,5 Гбит/с, а при мультиплексировании по длине волны – до 10 Гбит/с.

Темы для закрепления знаний

1. *Bluetooth*. Основные параметры, характеристики и свойства. Назначение.

2. *Bluetooth*. Способ связи. Элементная база.

3. *Bluetooth*. Этапы установления соединения.

4. *Bluetooth*. Режимы состояния устройства.

5. *Bluetooth*. Слои протокольного стека. Корневые протоколы.
6. *Bluetooth*. Модели использования и профили «общего назначения».
7. *Bluetooth*. Внутренняя организация и виды соединений и связей. Временные диаграммы.
8. Технология *UltraWide Band (UWB)*.
9. Интерфейс *Wireless USB*.
10. Инфракрасный порт. Стандарты и системы. Устройство и параметры.
11. Инфракрасный порт. Физический уровень и его варианты. Формат потока последовательных данных.

Глава 7. ИНТЕРФЕЙСЫ КЛАВИАТУРЫ И МЫШИ

7.1. Интерфейс клавиатуры

Клавиатура предназначена для ввода информации и представляет собой унифицированное устройство. Она имеет стандартный разъем и последовательный интерфейс связи с системной платой. В первых поколениях *PC* использовались 84-клавишные клавиатуры *AT*. В настоящее время используются так называемые расширенные (*enhanced*) клавиатуры *AT* или *PS/2*, имеющие более 100 клавиш. Электрический интерфейс клавиатур *XT* и *AT* совпадает, но двунаправленный интерфейс позволяет клавиатуре *AT* принимать команды от системной платы. Логический интерфейс различается и делает их несовместимыми. Клавиатура *PS/2* отличается от *AT* только исполнением разъема, при необходимости можно использовать переходник.

Структурная схема клавиатуры включает внутренний микроконтроллер, способный определить факты нажатия и отпускания клавиш. При нажатии клавиши клавиатура передает идентифицирующий ее скан-код нажатия. При удержании клавиши в нажатом положении через некоторое время клавиатура начинает автоповтор передачи скан-кода нажатия этой клавиши. При отпускании клавиши клавиатура передает идентифицирующий ее скан-код отпускания. Задержка автоповтора (*typematic delay*) и скорость автоповтора (*typematic rate*)

для клавиатур *AT* программируются. Расширенная клавиатура позволяет выбирать 1 из 3 наборов скан-кодов.

В настоящее время для подключения клавиатуры используется интерфейс *USB*, при этом клавиатура может иметь встроенный хаб, например для подключения мыши *USB*. Клавиатура *USB* питается от шины. Для клавиатуры *USB* требуется специальная поддержка со стороны *BIOS*; она имеется в современных системных платах.

7.1.1. Интерфейс клавиатуры *AT* и *PS/2*

Интерфейсы *AT* и *PS/2* являются последовательными синхронными двунаправленными интерфейсами. Они имеют две обязательные сигнальные линии *KB-Data* и *KB-Clock*.

Обе линии на системной плате подтягиваются резисторами к шине +5 В. Для формирования выходных сигналов низкого уровня на обеих сторонах используются элементы с открытым коллектором (стоком), а состояние линий может быть прочитано через входные линии контроллеров. На системной плате линии подтягиваются резисторами к шине +5 В.

Разъемы клавиатуры могут быть двух видов – обычная 5-контактная розетка *DIN* (клавиатура *T*) или малогабаритная розетка *mini-DIN* (*PS/2*) (рис. 7.1). На этот же разъем через плавкий предохранитель поступает напряжение питания клавиатуры +5 В. На разьеме *mini-DIN* могут присутствовать и сигналы мыши *PS/2* (для встроенных трекболов).

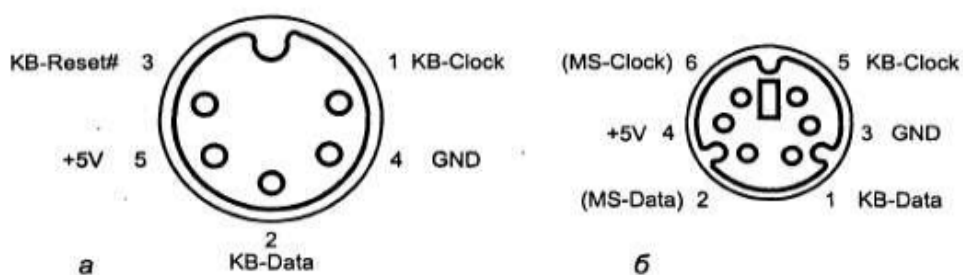


Рис. 7.1. Разъемы (вид со стороны контактов) подключения клавиатур:
а – *AT* и б – *PS/2*

На системной плате для работы с клавиатурой используется контроллер интерфейса клавиатуры – микроконтроллер 8042 или программно-совместимый с ним. Для обмена информацией в основ-

ном используется порт $60h$, из которого принимаются скан-коды. О необходимости чтения скан-кода контроллер сигнализирует процессору через аппаратное прерывание $IRQ1$, сигнал которого вырабатывается по каждому событию клавиатуры – нажатию и отпусканию клавиши.

Управление параметрами клавиатуры (автоповтора, выбор таблиц скан-кодов и т.п.) осуществляется командами, посылаемыми в порт $60h$. Контроллер транслирует команды в посылки, направляемые к клавиатуре.

Диаграмма сигналов двунаправленного интерфейса клавиатуры приведена на рис. 7.2, а. В исходном состоянии обе линии «отпущены» выходными формирователями в состояние с высоким уровнем. Клавиатура может начать передачу данных в произвольный момент, когда интерфейс находится в покое. Клавиатура формирует стартовый бит (низкий уровень) на линии *KB-Data* и первый импульс *KB-Clock*, что является сигналом контроллеру о необходимости начала приема. После подъема *KB-Clock* она выводит 0-й бит данных на линию *KB-Data*, а затем и следующий импульс *KB-Clock*. Контроллер должен «защелкивать» принятый бит данных по спаду *KB-Clock*. Так передаются все 8 бит данных и бит паритета, дополняющий число единичных бит до нечетного. После синхроимпульса бита паритета контроллер клавиатуры должен сформировать импульс *KB-Clock*, подтверждающий прием байта (*Ack*). Если весь байт с битом паритета не будет получен контроллером за 2 мс, контроллер прекращает прием данного байта и фиксирует ошибку тайм-аута.

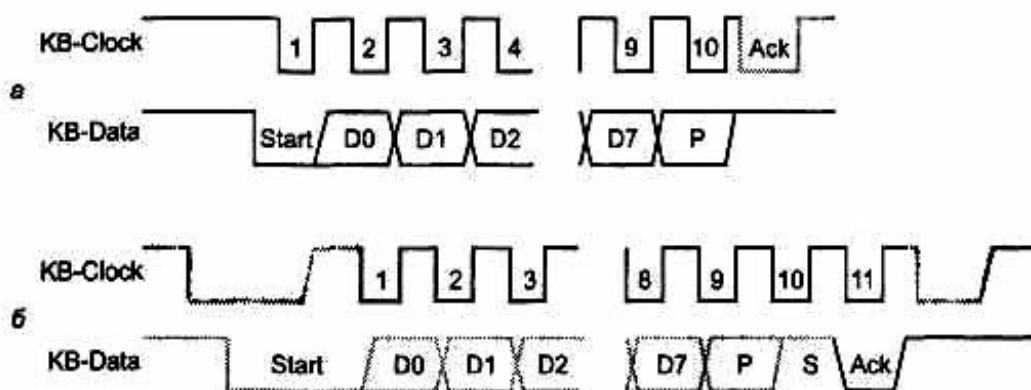


Рис. 7.2. Временные диаграммы интерфейса клавиатуры:
а – прием посылки от клавиатуры; б – передача команды в клавиатуру

Вывод команды контроллера в клавиатуру происходит по диаграмме рис. 7.2, б. Из состояния покоя контроллер устанавливает низкий уровень *KB-Clock* на 250 мкс и формирует старт-бит (низкий уровень) – это сигнал клавиатуре на прием команды. На него клавиатура должна ответить серией из 11 импульсов *KB-Clock*. По спаду очередного синхроимпульса контроллер выставляет очередной бит данных, а клавиатура его «защелкивает» по фронту формируемого ею же синхроимпульса. После бита паритета (9-й импульс) и единичного стоп-бита (10-й) на 11-м импульсе клавиатура формирует нулевой бит подтверждения (*Ack*). После этого контроллер формирует импульс *KB-Clock* (60 мкс), который является запросом на прием ответа клавиатуры. Контроллер ожидает окончания ответа на этот запрос не более 20 мс и, если ответ не придет за это время, формирует ошибку тайм-аута. Ошибка будет также в случае, если клавиатура не введет первый синхроимпульс за 15 мс от начала запроса или контроллер не примет данные, включая стоп-бит, за 2 мс с момента появления синхроимпульса бита 0.

7.1.2. Системная поддержка и программный интерфейс

Системная поддержка клавиатуры заключается в обработке фактов нажатия и отпускания клавиш и обеспечении сервисов ввода символов с клавиатуры, а также управления ее параметрами (задержка и частота автоповтора) и индикаторами. Она реализована на уровне *BIOS*.

Обработка скэн-кодов, принятых от клавиатуры ее контроллером, производится обработчиком аппаратного прерывания $09h$ (линия *IRQ1*). Результат обработки помещается в клавиатурный буфер, из которого по программному прерыванию *Int 16h* этот результат для дальнейшей обработки может быть извлечен значительно позже. Нажатие некоторых комбинаций клавиш не приводит к записи в клавиатурный буфер, а вызывает специальные процедуры. Прикладной программе, для которой требуется нестандартное использование клавиатуры (например, в качестве музыкальной), придется самой заниматься обработкой аппаратного прерывания *IRQ1*, перехватывая вектор *Int 09h*. Перехват этого вектора требуется и для вызова каких-либо функций резидентных программ по «горячим» клавишам.

Клавиатура диагностируется процедурой *POST*, которая анализирует клавиатуру. Во время этого теста клавиатура мигнет всеми индикаторами, после чего может остаться включенным только индикатор *NumLock* (зависит от установки в *BIOS Setup*). В случае обнаружения ошибки клавиатуры на консоль выводится сообщение с возможным указанием скан-кода залипшей клавиши. То же самое произойдет, если тест не обнаружит клавиатуру, например, из-за вывалившегося разъема или перегоревшего предохранителя. Чтобы начальная загрузка не останавливалась по ошибке клавиатуры, тестирование клавиатуры может быть отменено настройкой *CMOS Setup*.

7.2. Интерфейс мыши

Мышь и подобные ей устройства (трекбол и др.) служат для ввода координат указателя и подачи команд. Имеются 4 основных вида мышей, отличающихся своим интерфейсом: *Bus Mouse*, *Serial Mouse* и *PS/2-Mouse* и *USB-Mouse*.

Интерфейсы *Serial Mouse* и *PS/2-Mouse* последовательные, но имеют существенные принципиальные различия в уровнях сигналов, способе синхронизации, частоте и формате посылок. Интерфейс *PS/2* использует однополярный сигнал с уровнями ТТЛ, питание мыши – однополярное с напряжением +5 В относительно шины *GND*. Интерфейс *RS-232C*, применяемый в *Serial Mouse*, использует двухполярный сигнал с уровнями срабатывания +3 и –3 В, и для него требуется двухполярное (относительно шины *GND*) питание мыши. Интерфейс *PS/2* использует две отдельные сигнальные линии, одну для передачи данных, другую – для сигналов синхронизации. *Serial Mouse* использует асинхронный способ передачи данных всего по одной линии.

Для совместимости этих интерфейсов выпускаются и продаются переходники (пассивные), позволяющие выбирать способ подключения мыши. Эти переходники предназначены только для универсальных мышей, у которых встроенный контроллер по напряжению питания способен распознать, к какому интерфейсу его подключили, и установить соответствующий тип своего выходного интерфейса.

Мыши для компьютера *Macintosh* хотя и имеют разъем, с виду напоминающий разъем *PS/2*, но разъемы эти разные, да и интерфейс совершенно иной.

7.2.1. Последовательные мыши *MS Mouse* и *PC Mouse*

Serial Mouse – мышь, которая использует последовательный интерфейс *RS-232C* и подключается через 9- или 25-контактный разъем *COM*-порта. Эта мышь имеет встроенный микроконтроллер, который обрабатывает сигналы от координатных датчиков и кнопок. Для передачи информации применяется асинхронная передача, а двупольное питание, требуемое по протоколу *RS-232*, обеспечивается от управляющих линий интерфейса. Недостатком *Serial Mouse* является то, что она занимает *COM*-порт и требует монопольного владения его штатной линией прерывания (*IRQ4* для *COM1* и *IRQ3* для *COM2*).

Имеются две основные разновидности – *MS Mouse (Microsoft Mouse)* и *PC Mouse (Mouse Systems Mouse)*, которые требуют разных драйверов и используют различные форматы посылок.

MS Mouse имеет 7 бит данных, трехбайтный пакет (в «классическом» варианте), положительным значениям соответствует перемещение по координате *X* вправо, а по координате *Y* вниз. Для трехкнопочных мышей добавляется четвертый байт, передаваемый только при изменении состояния средней кнопки. Для *3D*-мыши четвертый байт имеет иное назначение. *PC Mouse* имеет 8 бит данных, пятибайтный пакет, положительным значениям соответствует перемещение по координате *X* вправо, а по координате *Y* вверх.

Распределение сигналов по контактам разъемов приведено в таблице.

Разъемы *Serial Mouse*

Сигнал	Контакт <i>DB9</i>	Контакт <i>DB25</i>	Цепь <i>COM</i> -порта
<i>Data</i>	2	3	<i>RxD</i>
<i>GND</i>	5	7	<i>GND</i>
+ <i>V</i> (питание)	7,(4)	2,(20)	<i>RTS,(DTR)</i>
- <i>V</i> (питание)	3	2	<i>TxD</i>

На уровне *BIOS* мыши *Serial Mouse* не поддерживаются. Системная поддержка последовательной мыши осуществляется только на уровне ОС (сервисы вызываются через *Int 33h*), драйвер мыши – загружаемый или встроенный в ОС. Для работы мыши обязательно требуется линия аппаратного прерывания *IRQ4* или *IRQ3* для последовательных мышей на портах *COM1* или *COM2* соответственно.

7.2.2. Мышь PS/2

С появлением компьютеров *PS/2* пришла и мышь с одноименным названием *PS/2-Mouse*. Ее интерфейс и разъем *6-pin mini-DIN* аналогичны клавиатурному и, как правило, реализуются тем же контроллером клавиатуры 8242.

Адаптер и разъем *PS/2-Mouse* устанавливаются на многих современных системных платах. Контроллер мыши *PS/2* может быть расположен как на системной плате, так и на карте расширения. Он занимает дополнительные адреса в пространстве ввода-вывода. Разъем *PS/2-Mouse* показан на рис. 7.3. С мышью *PS/2* связь двусторонняя: процессор может посылать контроллеру 8242 специальные команды, но в отличие от интерфейса клавиатуры перед записью в порт $60h$ каждого «мышинного» байта (и команды, и ее параметра) в порт $64h$ должен записываться код $D4h$.

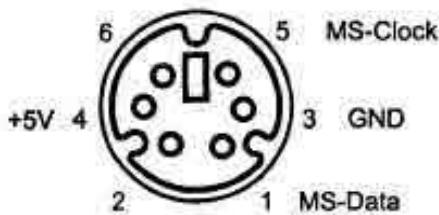


Рис. 7.3. Разъем *PS/2-Mouse*

Мышь может работать в одном из двух режимов. В потоковом режиме (*stream mode*) мышь посылает данные по любому изменению состояния; в режиме опроса (*remote mode*) мышь передает данные только по запросу процессора. Есть еще диагностический режим (*wrap mode*), в котором мышь возвращает эхом данные, посылаемые ей контроллером. По приему пакета от мыши контроллер устанавливает флаг *Mouse_OBF* и вырабатывает прерывание *IRQ12*, если оно не запрещено командным байтом 8242.

На уровне *BIOS* мыши с интерфейсом *PS/2* поддерживаются через прерывание *Int 15h* с кодами функций $C200 - C209h$. *BIOS* обеспечивает настройку параметров мыши (посылку вышеперечисленных команд). Собственно драйвер мыши (обработчик прерывания по вектору $74h$ от запроса *IRQ12*), обрабатывающий ее информационные посылки, входит лишь в состав ОС или загружается отдельно.

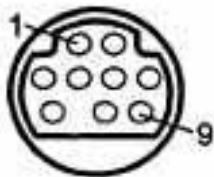


Рис. 7.4. Разъем *Bus Mouse*

7.2.3. Мышь Bus Mouse

Вариант *Bus Mouse* (шинная мышь) применялся в первых мышах. Здесь мышь содержит только датчики и кнопки, а их сигналы обрабатываются на специализированной плате адап-

тера (обычно *ISA*, в настоящее время неприменяемой). Кабель 9-проводный, разъем специальный (рис. 7.4), хотя на первый взгляд и напоминающий разъем *PS/2-Mouse*. Главный недостаток такой системы заключается в том, что адаптер занимает слот системной шины, адреса ввода-вывода и линию запроса прерывания.

Темы для закрепления знаний

1. Интерфейс клавиатур *AT* и *PS/2*.
2. Временные диаграммы интерфейса клавиатуры.
3. Системная поддержка и программный интерфейс клавиатуры.
4. Интерфейс *MS Mouse*.
5. Интерфейс *PC Mouse*.
6. Интерфейс мыши *PS/2*.
7. Интерфейс мыши *Bus Mouse*.

Глава 8. ИНТЕРФЕЙСЫ АУДИОУСТРОЙСТВ

Звуковая система ПК – комплекс программно-аппаратных средств, выполняющих следующие функции:

- запись звуковых сигналов, поступающих от внешних источников, например микрофона или магнитофона, путем преобразования входных аналоговых звуковых сигналов в цифровые и последующего сохранения на жестком диске;
- воспроизведение записанных звуковых данных с помощью внешней акустической системы или головных телефонов (наушников);
- воспроизведение звуковых компакт-дисков;
- микширование (смешивание) при записи или воспроизведении сигналов от нескольких источников;
- одновременная запись и воспроизведение звуковых сигналов (режим *FullDuplex*);
- обработка звуковых сигналов: редактирование, объединение или разделение фрагментов сигнала, фильтрация, изменение его уровня;

- обработка звукового сигнала в соответствии с алгоритмами объемного (трехмерного – *3D-Sound*) звучания;
- генерирование с помощью синтезатора звучания музыкальных инструментов, а также человеческой речи и других звуков;
- управление работой внешних электронных музыкальных инструментов через специальный интерфейс *MIDI*.

8.1. Устройство аудиосистемы ПК

Звуковая система ПК конструктивно представляет собой звуковые карты, устанавливаемые в слот материнской платы либо интегрированные на материнскую плату или карту расширения другой подсистемы ПК. Отдельные функциональные модули звуковой системы могут выполняться в виде дочерних плат, устанавливаемых в соответствующие разъемы звуковой карты.

Классическая звуковая система, показанная на рис. 8.1, содержит:

- модуль записи и воспроизведения звука;
- модуль синтезатора;
- модуль интерфейсов;
- модуль микшера;
- акустическую систему.

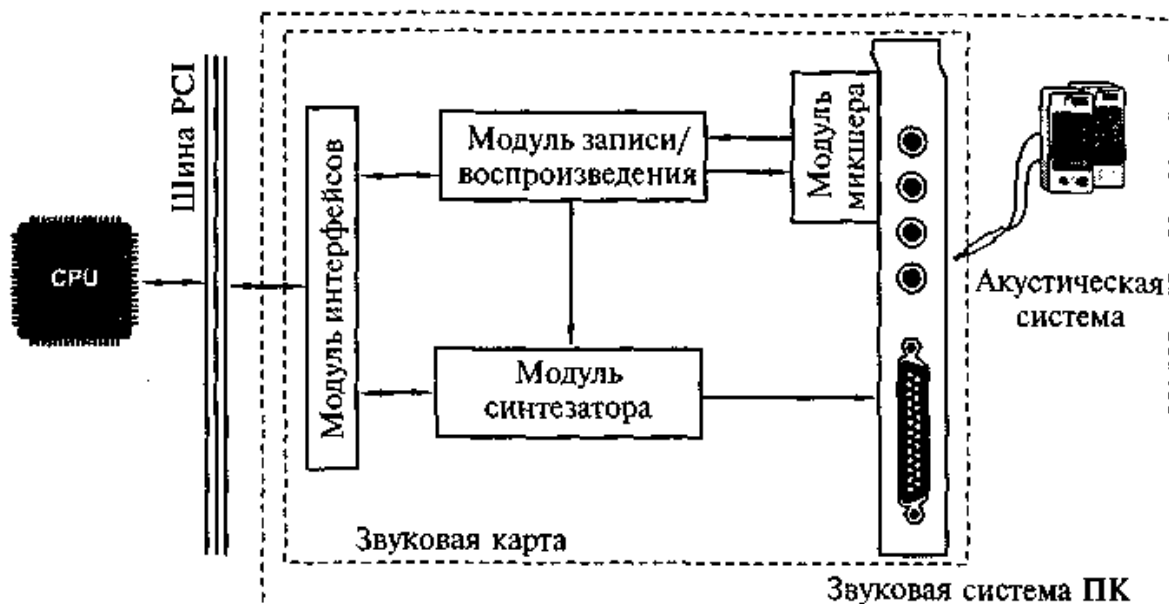


Рис. 8.1. Структура звуковой системы ПК

Первые четыре модуля, как правило, устанавливаются на звуковой карте. Причем существуют звуковые карты без модуля синтезатора или модуля записи/воспроизведения цифрового звука. Каждый из модулей может быть выполнен либо в виде отдельной микросхемы, либо входить в состав многофункциональной микросхемы. Таким образом, чипсет звуковой системы может содержать как несколько, так и одну микросхему.

Звуковая карта имеет набор разъемов для подключения внешних аудиосигналов, аналоговых и цифровых, а также интерфейс *MIDI* для связи с электронными музыкальными инструментами. Аудиоданные в цифровом виде могут передаваться также и по универсальным шинам *USB* и *FireWire*.

8.2. Аналоговые интерфейсы аудиоустройств

Для подключения стандартной бытовой аппаратуры (микрофон, аналоговый выход *CD-ROM* и т.д.) на большинстве карт массового потребления применяют аналоговые интерфейсы. Для этих интерфейсов применяют различные разъемы – «мини-джеки» (*jack*) или *RCA* диаметром 3,5 мм, моно и стерео. Мини-джеки имеют диаметр 3,5 мм и могут быть моно и стерео. Качество контактов у них довольно низкое и поэтому они являются источником шумов (шорохов и тресков), а также иногда просто теряют контакт. Их полноразмерные 6-миллиметровые «родственники», характерные для профессиональной аппаратуры, имеют весьма высокое качество, но из-за крупных габаритов на звуковых картах не используются. На некоторых высококачественных картах сигналы линейного входа и выхода выводятся на пары разъемов *RCA* («колокольчики» или «тюльпаны»), которые обеспечивают очень хороший контакт, особенно в позолоченном варианте.

Раскладка цепей на мини-джеках унифицирована: левый канал – на центральном контакте, экран (земля) на внешнем цилиндре, правый канал – на промежуточном цилиндре. Если стереоджек включить в моногнездо и наоборот, сигнал пойдет только по левому каналу. Для подключения центрального и низкочастотного каналов в 6-колоночной системе единого подхода нет – может потребоваться перекрестный кабель.

Для подключения к звуковой карте устройств на задней панели имеются разъемы:

– *Line In* – линейный вход от магнитофона, тюнера, проигрывателя, синтезатора и т. п. Чувствительность порядка 0,1 – 0,3 В.

– *Speaker Out* – выход на акустические системы или головные телефоны. Подключать к нему внешний усилитель мощности нецелесообразно, поскольку здесь искажения больше, чем на линейном выходе.

– *Mic In* – микрофонный вход, чувствительность 3 – 10 мВ. Этот вход обычно монофонический, но иногда используется трехконтактное гнездо (как в стерео), у которого дополнительный контакт (на месте правого канала) выделен для подачи питания на электретный микрофон.

Подключение внутренних устройств осуществляется через четырехштырьковые разъемы, различающиеся как шагом между выводами, так и их назначением. Для подключения *CD-ROM* часто ставят рядом два, а то и три разъема с параллельно соединенными сигнальными контактами. Расположение сигнальных контактов аудиовходов приведено на рис. 8.2.

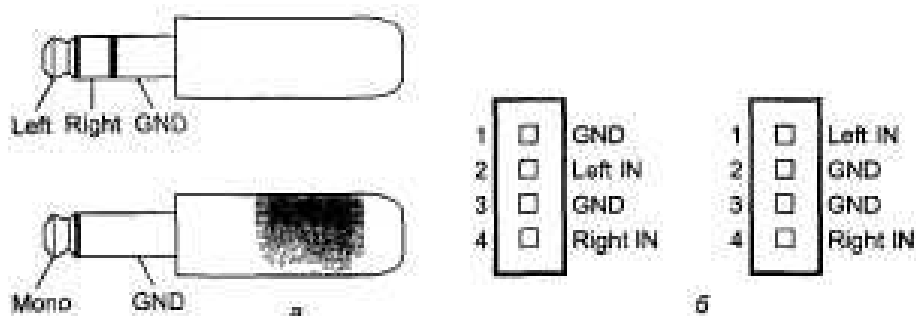


Рис. 8.2. Разъемы подключения аудиосигналов

8.3. Цифровые интерфейсы аудиоустройств

В аудиосистеме применяются различные последовательные цифровые интерфейсы. Наиболее распространен *S/PDIF* (*Sony/Philips Digital Interface Format*), который используется для передачи аудиосигналов между блоками бытовой цифровой аудиоаппаратуры (*DAT*, *CD-ROM* и т. п.). Этот интерфейс является упрощенным вариантом студийного интерфейса *AES/EBU* (*Audio Engineers Society/European*

Broadcast Union). В интерфейсе *AES/EBU* используется симметричный двухпроводный экранированный кабель с импедансом 110 Ом, разъемы *XLR*, он имеет уровни сигнала 3 – 10 В, длину кабеля – до 12 м.

Интерфейс *S/PDIF* имеет сигнальные уровни 0,5 – 1 В и использует коаксиальный кабель 75 Ом. В нем применяются различные разъемы: *RCA* или *BNC*, коаксиальный кабель 75 Ом длиной до 2 м. Внутренние разъемы *S/PDIF* на звуковых картах – это просто пара штырьков (как у джамперов) на плате с соответствующей ответной частью на кабеле. В схеме передатчика *S/PDIF* имеется разделительный импульсный трансформатор (1:1) для обеспечения гальванической развязки устройств. В некоторых вариантах разделительный трансформатор может отсутствовать. При стыковке устройств с нестандартными интерфейсами возможны проблемы, связанные с несоответствием уровней сигналов. Решение этих проблем заключается в установке дополнительных формирователей сигнала.

Существует также оптическая версия интерфейса *S/PDIF* – *Toslink*, стандарт *EIAJ CP-1201* с инфракрасными излучателями (660 нм). Применение оптики позволяет обеспечить полную гальваническую развязку устройств, что необходимо для снижения уровня наводок. Длина кабеля зависит от вида оптоволокну: для пластикового волокна (*POF*) не более 1,5 м, для стеклянного волокна – 3 м. Для преобразования электрического сигнала в оптический предлагаются различные схемы, одна из которых приведена на рис. 8.3. Здесь первый инвертор посредством обратной связи выведен на линейный участок передаточной характеристики, благодаря чему малый входной сигнал вызывает его переключение. В схеме могут быть использованы различные микросхемы инверторов, вместо светодиода можно использовать и фирменный трансивер *Toslink* [1].

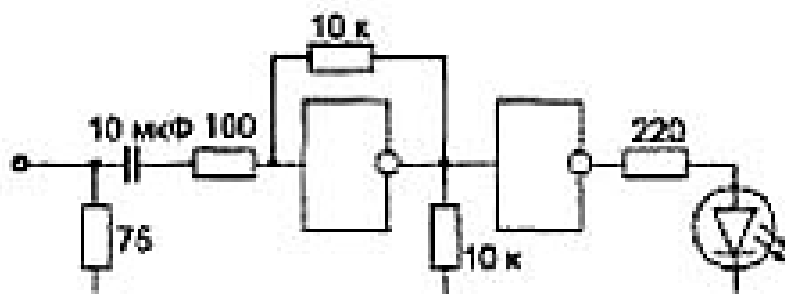


Рис. 8.3. Схема преобразователя электрического интерфейса *S/PDIF* в оптический (*Toslink*)

В последовательном интерфейсе *S/PDIF* информация передается с использованием кадров, с обеспечением синхронизации и контролем достоверности передачи (кодами Рида – Соломона). В кадре имеются средства контроля достоверности передачи (кодами Рида – Соломона), признак формата данных – РСМ или не РСМ. Это позволяет по данному интерфейсу передавать и упакованные цифровые данные (например, *MPEG* для *AC-3*). Имеется также бит защиты от копирования, признак предискажений и некоторые другие служебные данные. Разрядность выборок в режиме РСМ может быть 16, 20 или 24 бит, частота дискретизации в соответствии с теоремой Котельникова может быть 32 (редко) или 44,1 кГц (в большинстве случаев) и выше – для профессиональных звуковых карт. Эти параметры влияют на частоту цифрового сигнала. В приемнике *S/PDIF* частота выборок определяется по принимаемому сигналу.

Кроме этих интерфейсов в студийной аппаратуре применяют интерфейсы *ADAT* и *TDIF*, которые имеются только на дорогих профессиональных звуковых картах. В них используется высокая частота дискретизации – до сотен кГц.

Для обмена данными с приводами *DVD* применяется цифровой последовательный интерфейс *I2C*.

8.4. Интерфейс *MIDI*

Цифровой интерфейс *MIDI* (*Musical Instrument Digital Interface*) разработан в 1983 г. для сопряжения компьютеров, музыкальных инструментов и другой электромusical техники. Он использует последовательную асинхронную передачу с частотой 31,25 Кбит/с. В настоящее время интерфейс *MIDI* имеют и дорогие синтезаторы, и дешевые музыкальные клавиатуры, которые могут использоваться в качестве устройств ввода компьютера. Сообщения, которыми устройства обмениваются между собой по интерфейсу *MIDI*, описаны в книге [2]. На одном интерфейсе может быть организовано до 16 логических каналов, каждый из которых может управлять своим инструментом.

Физический уровень интерфейса использует токовую петлю с током 5 мА (возможно до 10 мА) с гальванической (оптронной) развяз-

кой входной цепи. Логическому нулю соответствует наличие тока, логической единице (и покою) – отсутствие тока.

Интерфейс определяет три типа портов: входной *MIDI-In*, выходной *MIDI-Out* и транзитный *MIDI-Thru*.

Входной – *MIDI-In* представляет собой вход интерфейса «токовая петля», гальванически развязанного от приемника оптроном с быстродействием не хуже 2 мкс. Устройство отслеживает информационный поток на этом входе и реагирует на адресованные ему команды и данные.

Выходной – *MIDI-Out* представляет собой выход источника тока, гальванически связанного со схемой устройства. Ограничительные резисторы предохраняют выходные цепи от повреждения при замыкании на землю или источник 5 В. На выход подается информационный поток от данного устройства. При специальной настройке устройства в этом потоке может содержаться и транслированный входной поток, но это нетипично.

Транзитный – *MIDI-Thru* служит только для ретрансляции входного потока, по электрическим свойствам он аналогичен выходному. Его наличие не является обязательным для всех устройств.

Разъемы интерфейса *MIDI* типа *DIN* имеют 5 контактов, из которых используются только 3 (рис. 8.4).

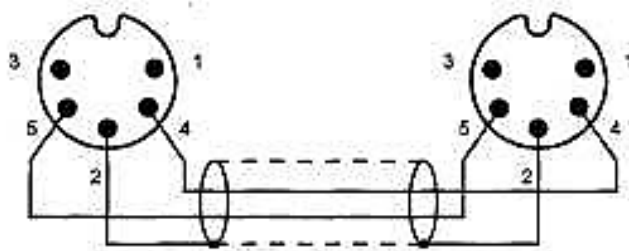


Рис. 8.4. Соединительные кабели *MIDI*

На компьютере порт *MIDI* (с сигналами ТТЛ) обычно выводится на неиспользуемые контакты (12 и 15) разъема игрового адаптера (*DB-15S*). Для преобразования его сигнала с уровнями ТТЛ в «токовую петлю» используется переходной адаптер, обычно встраиваемый в специальный кабель, вариант схемы которого из [1] приведен на рис. 8.5. Некоторые модели *PC* имеют встроенные адаптеры и стандартные 5-штырьковые разъемы *MIDI*.

Программная совместимость порта *MIDI* с *UART MPU-401* фирмы *Roland* кратко рассматривается в [1]. Контроллер *MPU-401* поддерживал простой режим работы *UARTmode*, в котором использовался только двунаправленный асинхронный порт; в современных звуковых картах совместимость с *MPU-401* поддерживается только в этом режиме.

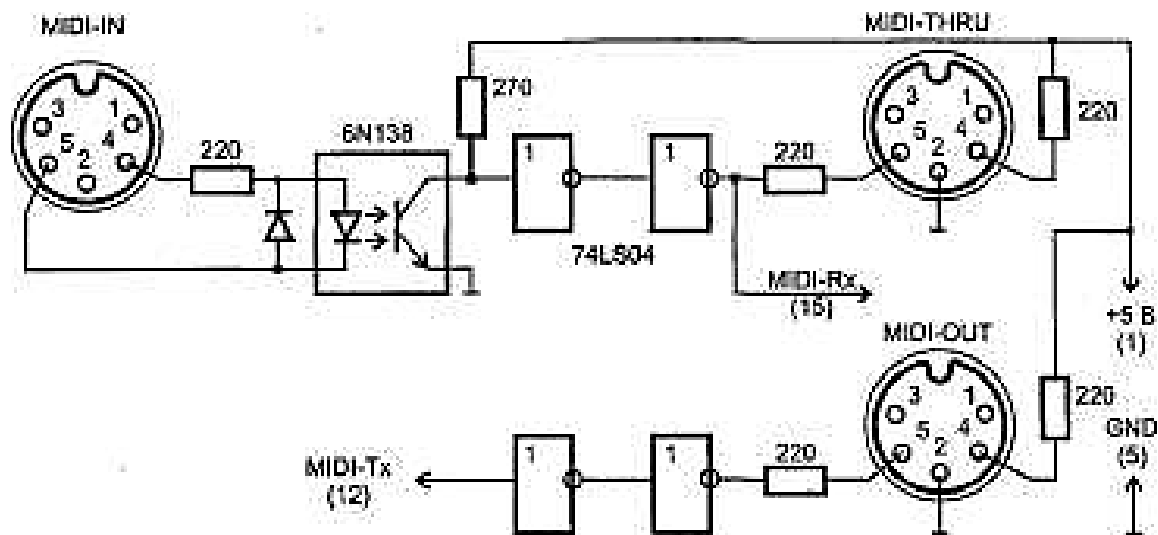


Рис. 8.5. Вариант схемы кабеля-адаптера *MIDI*

8.5. Интерфейс дочерней карты

Ряд моделей звуковых карт имеют внутренний интерфейсный разъем подключения дочерней карты с *MIDI*-синтезатором. Для ее подключения некоторые модели звуковых карт имеют внутренний интерфейсный разъем (*Daughterboard Connector*), описанный в [1]. На разъем с основной карты выводится сигнал *MIDI*-порта (ТТЛ, как и на разъем джойстика) и сигнал аппаратного сброса синтезатора, а с дочерней карты принимается стереофонический аналоговый сигнал, который поступает в микшер основной карты. В шинах питания аналоговая земля (*AG*) отделена от цифровой (*DG*) (см. таблицу). Дополнительно может использоваться и вход *MIDI* (см. таблицу). Разъем может обозначаться и как *WT (Wavetable) Connector*, *Waveblaster Connector*.

Подключение дочерней карты эквивалентно подключению внешнего синтезатора к *MIDI*-выходу звуковой карты. Если на звуковой карте отсутствует разъем подключения дочерней карты, то до-

черную карту можно подключить и к внешнему разъему джойстика *MIDI* и аналоговым входам звуковой карты. Конечно, на дочернюю карту нужно подать питание, а также сигнал аппаратного сброса.

Назначение контактов разъема подключения дочерней карты

Контакт	Цепь	Контакт	Цепь
1	<i>DG</i>	2	–
3	<i>DG</i>	4	<i>MIDI_Out#</i>
5	<i>DG</i>	6	+5В
7	<i>DG</i>	8	<i>MIDI_In#</i> (необязательный)
9	<i>DG</i>	10	+5В
11	<i>DG</i>	12	
13		14	+5В
15	<i>AG</i>	16	
17	<i>AG</i>	18	+12В
19	<i>AG</i>	20	<i>Audio(R)</i>
21	<i>AG</i>	22	–12В
23	<i>AG</i>	24	<i>Audio(L)</i>
25	<i>AG</i>	26	<i>Reset#</i>

Темы для закрепления знаний

1. Аналоговые интерфейсы аудиоустройств.
2. Цифровые интерфейсы аудиоустройств.
3. Интерфейс *MIDI*.
4. Интерфейс дочерней карты.

Глава 9. ИНТЕРФЕЙСЫ ДИСПЛЕЕВ

Первые графические интерфейсы были дискретные – *MDA*, *CGA*, *EGA*. Интерфейсы второго поколения были аналоговыми – *VGA*, *SVGA*.

Интерфейсы третьего поколения, предназначенные для жидкокристаллических (*LCD*) дисплеев, цифровые – *DVI*, *P&D*, *DFP*.

Дискретные: интерфейсы имели уровни сигналов *TTL*, 9-контактный разъем: 6 линий информационных контактов, 1 – земля, 2 – синхронизация.

Разное количество цветов:

- *MDA* – черно-белый;
- *CGA* – 16 цветов;
- *EGA* – 64 цвета.

9.1. Аналоговые интерфейсы

Два стандарта – *VGA* и *SVGA* – формируют цветное изображение из трех основных цветов – *RGB* (красный, зеленый, синий) с различным количеством оттенков от 256 до 16 млн.



Рис. 9.1. Разъем *VGA*

Разрешение экрана определяется в зависимости от стандарта. В стандарте *VESA DTM* (1994-1998 гг.) задается дискретный ряд значений разрешения; в *VESA GTF* – разрешение определяется по формулам. Исходные данные: формат экрана в пикселях, необходимые об-рамления, тип развертки – через строчную засветку, построчную, одна из заданных частот (частота кадров, строк или пикселей).

Эти стандарты ранее использовали 9-контактный разъем *DB-9S*, сейчас – 15-контактный *VGA-15* (рис. 9.1). Для определения типа монитора и его параметров были введены дополнительные логические сигналы идентификации *ID0...ID3*. Позднее их заменили двумя линиями *SCL*, *SDA* с последовательной передачей.

Разъем *DB-15* хорошо работает лишь на частотах до 150 МГц. Для повышения частоты перешли к другому разъему *BNC*, где передача осуществляется не по витой паре, а по коаксиальным кабелям. Количество кабелей 3 – 5. Волновое сопротивление $p = 75$ Ом, длина $l = 10 - 15$ м.

Универсальный разъем *EVC* (затем разъем стал называться *P&D-A*) имеет 4 коаксиальных кабеля, частоту до 2 ГГц каждый; $p = 75$ Ом. Низкочастотный 30-контактный разъем имеет 3 уровня реализации: полый, видео и *DDS*, мультимедиа и аудиоинтерфейс. Имеется вариант *EVS* для портативных ПК.

9.2. Цифровые интерфейсы

Интерфейс *Panel-Link* определяет протокол и структуру, используемые во всех цифровых интерфейсах, рассматриваемых ниже. В 1996 г. его спецификация (*FPDI-2*) была утверждена *VESA*.

Цифровой интерфейс имеет 3 канала передачи данных (*Data[0:3]*) и канал синхронизации *Clock*. Существует расширенный вариант, имеющий 6 каналов передачи данных.

Схема интерфейса приведена на рис. 9.2. В каналах используется дифференциальная передача сигналов с минимизацией переходов – протокол *TMDS*. (*Transition Minimized Differential Signaling*).



Рис. 9.2. Интерфейс *Panel-Link*

Каждый канал данных образован кодером, расположенным на видеокарте, линией связи и декодером, расположенным в дисплее. На вход кодера каждого канала поступают 8 бит кода яркости базисного цвета текущего пиксела. На вход кодера канала 0 поступают сигналы строчной и кадровой синхронизации, а на остальные каналы – дополнительные управляющие сигналы *CTL[0:3]* по паре на каждый канал.

Кодеры преобразуют данные в последовательный код для минимизации переключений, при этом 8 входных бит кодируются 10-битным символом, передаваемым по каналу последовательно. В зависимости от входного сигнала разрешения данных *DE* кодеры пере-

дают либо данные цветовых каналов, либо синхросигналы и управляющие биты. На приемной стороне сигналы декодируются и восстанавливаются в том же виде, в котором они поступали на входы кодеров.

Частота пикселей может достигать 165 МГц, интерфейс обеспечивает максимальное разрешение 1280x1024 (24 бита на пиксел).

Интерфейс *DVI* (*Digital Visual Interface*) – стандарт на интерфейс и соответствующий разъем, предназначенный для передачи видеоизображения на цифровые устройства отображения, такие как жидкокристаллические мониторы и проекторы. Разработан группой *DDWG* (*Digital Display Working Group* – рабочая группа по цифровым дисплеям) в 1999 г. и предназначен для подключения дисплеев любого типа (ЭЛТ и матричных) к компьютеру.

Виды *DVI*:

- *DVI-A* – только аналоговая передача;
- *DVI-I* – аналоговая и цифровая передача;
- *DVI-D* – только цифровая передача.

Интерфейс *DVI* использует несколько вариантов разъема (рис. 9.3); наиболее часто используются два из них:

1. Чисто цифровой – содержит канал синхронизации и три канала данных (*Data 0-2*) и предусматривает способ повышения пропускной способности за счет обратных переходов.

2. Цифровой с традиционными аналоговыми сигналами.

Дополнительно в интерфейс *DVI* входят сигналы интерфейса *VESA DDC2*:

- *DDC Data*;
- *DDC Clock*;
- линия питания +5 В;
- сигнал *HPD* (*Hot Plug Detect*) – отслеживание подключения/отключения дисплея.

«Горячее» подключение обеспечивается также и механическими особенностями разъемов, поддерживающих требуемую последовательность соединения / разъединения разных групп контактов.

Интерфейс и дисплеи с *DVI* должны обеспечивать стандартные (*VESA*) графические режимы, начиная от 640x480/60 Гц (частота пикселей 22,175 МГц). Его предел – 2048x1536 пикселей (частота 330 МГц).

Интерфейс поддерживает сигнализацию управления энергопотреблением (*DPMS*). Вариант с аналоговыми сигналами *VESA DDC2* позволяет подключить аналоговые мониторы. Этот интерфейс является самым универсальным, мощным и наиболее распространенным среди цифровых интерфейсов.

Интерфейс *P&D* обладает пропускной способностью 3,6 Гбит/с (450 Мбайт/с) при частоте пикселей 150 МГц и представлении цвета 24-битным числом (*True Color*). Разрешение 1280x1024 (24 бита на пиксел).

Физические линии реализованы экранированными витыми парами. Передатчики являются дифференциальными коммутируемыми источниками тока (12 мА), входы дифференциальных приемников подтянуты нагрузочными резисторами 50 Ом к уровню питания +3,3 В, амплитуда сигнала 500 мВ. Разъем у *P&D* такой же, как и у *EVC*. На контакты выведены цифровые сигналы. Его можно рассматривать как комбинацию усеченных интерфейсов *DVI* и *EVC*.

P&D дорогой и используется редко.

Интерфейс *DFP* – интерфейс плоских дисплеев *DFP* (*Digital Flat Panel*, 1999 г.) использует дешевый разъем (рис. 9.4) типа *MDR* (*mini-D ribbon*) с ленточными контактами.

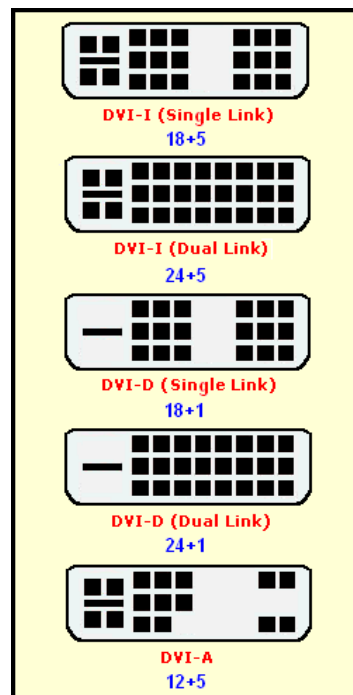


Рис. 9.3. Виды разъемов интерфейса *DVI*

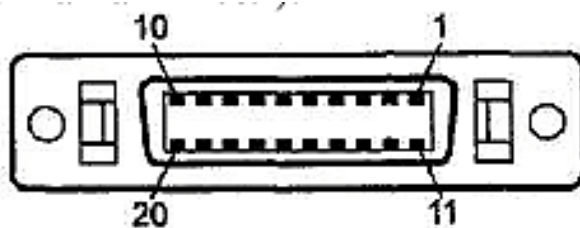


Рис. 9.4. Разъем интерфейса *DFP*

Три пары сигналов для цифровых каналов данных:

- пара для цифрового канала синхронизации;
- питание (+5 В);

- канал *DDC2*;
- сигнал обнаружения «горячего» подключения (*HPD*).

Частота пикселей может достигать 85 МГц (для плоских панелей не требуется слишком высокая частота развертки). Интерфейс для режимов до 1280x1024 (24 бита на пиксел). Достоинство интерфейса – низкая стоимость, недостаток – невозможность подключения мониторов с аналоговым входом.

Интерфейс *HDMI* – интерфейс для мультимедиа высокой чёткости, позволяющий передавать цифровые видеоданные высокого разрешения и многоканальные цифровые аудиосигналы с защитой от копирования.

Разъём *HDMI* (рис. 9.5) обеспечивает цифровое *DVI*-соединение нескольких устройств с помощью соответствующих кабелей. Основное различие между *HDMI* и *DVI* состоит в том, что разъём *HDMI* меньше по размеру, а также поддерживает передачу многоканальных цифровых аудиосигналов. Пропускная способность *HDMI* – 4,9 – 10,2 Гбит/с, длина кабеля – до 10 м, но может быть увеличена до 20 – 35 м при применении внешних или внутренних усилителей-повторителей.



Рис. 9.5. Разъём интерфейса *HDMI*

Кабель *HDMI* содержит внешнюю оболочку, экранирующую оплётку из проволок с дополнительной медной неизолированной жилой для пайки, экран из алюминиевой фольги, полипропиленовую оболочку, экранированные витые пары пятой категории с волновым сопротивлением 100 Ом для сигнала синхронизации и сигналов данных. Экран каждой витой пары имеет внешнюю изоляцию и проволоку для пайки (дренажный проводник). Кроме этого имеется неэкранированная витая пара для сигналов *SDA*, *SCL* и отдельно идущие проводники для питания и управляющих сигналов.

В интерфейсах *HDMI* и *DVI* используется технология высокоскоростной передачи цифровых потоков *TMDS*, рассмотренная выше. Используются три канала, передающие потоки аудио/видео и дополнительных данных, с пропускной способностью до 3,4 Гбит/с на канал. Кроме того, в *HDMI* может быть встроена поддержка передачи команд

и управляющих сигналов с помощью *CEC* (*Consumer Electronics Control*). Информация о параметрах подключенных устройств передается с использованием протокола *Access Bus* (сигналы *SCL* и *SDA*) и *DDC* (*Display Data Channel*).

К настоящему времени разработаны многочисленные версии интерфейса *HDMI* от 1.0, появившейся в 2002 г., до 2.0, предложенной в 2013 г. Версия 2.0 предполагает увеличение скорости передачи до 18 Гбит/с, что позволит поддерживать разрешение *Full HD 3D* со скоростью 120 кадров в секунду и разрешение 4 K со скоростью 60 кадров в секунду (*FPS*).

Интерфейс *DisplayPort*

Интерфейс *DisplayPort* предполагается к использованию в качестве наиболее современного интерфейса соединения аудио- и видеоаппаратуры, в первую очередь для соединения компьютера с дисплеем или компьютера и систем домашнего кинотеатра. Как и *HDMI*, *DisplayPort* использует технологию высокоскоростной передачи цифровых потоков *TMDS*, но отличается технологией защиты от копирования. Максимальная скорость передачи данных версии *DisplayPort* 1.2 составляет 21,6 Гбит/с на расстоянии до 3 м, что больше, чем *HDMI*. Разъем интерфейса показан на рис. 9.6.

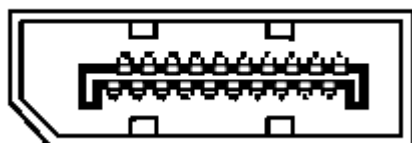


Рис. 9.6. Разъем интерфейса *DisplayPort*

Дополнительным преимуществом *DisplayPort* по сравнению с *HDMI* является то, что его использование не предполагает никаких выплат.

К настоящему времени разъем *Mini DisplayPort* заменен на аналогичный по внешнему виду разъем *Thunderbolt* (рассмотренный в гл. 4), который является обратно совместимым с ним, однако имеет значительно бóльшие возможности для использования.

Темы для закрепления знаний

1. Аналоговые интерфейсы дисплеев.
2. Интерфейс *Panel-Link* и протокол *TDMS*.
3. Интерфейс *DVI*.

4. Интерфейс *P&D*.
5. Интерфейс *DFP*.
6. Интерфейс *HDMI*.
7. Интерфейс *DisplayPort*.

Глава 10. ВНУТРИПЛАТНЫЕ ИНТЕРФЕЙСЫ

10.1. Шина I^2C

Шина *Inter IC Bus*, или кратко I^2C , предназначена для обмена информацией о параметрах микросхем при их соединении между собой. Шина обеспечивает двустороннюю передачу 8-битных данных между подключенными устройствами с использованием синхронного последовательного обмена. Передача данных может быть как одноадресной к выбранному устройству, так и широковещательной.

Шина поддерживает следующие режимы:

- стандартный *Standard Mode (S)* со скоростью 0 – 100 Кбит/с;
- быстрый *Fast Mode (F)* со скоростью 0 – 400 Кбит/с. Он появился в версии 1.0 и предъявляет повышенные требования к форме сигнала и фильтрации помех. Также здесь введен режим 10-битной адресации устройств.
- высокоскоростной *High speed (Hs)*, — в котором скорость передачи может достигать 3,4 Мбит/с. Он описан в версии 2.0, которая вышла в 1998 г., когда интерфейс I^2C стал фактически промышленным стандартом, использующимся в большом числе различных ИС.

10.1.1. Физический уровень I^2C

На физическом уровне I^2C использует синхронный последовательный обмен и поэтому имеет две сигнальные линии: данных *SDA (Serial Data)* и синхронизации *SCL (Serial Clock)*. Уровни сигналов – стандартные, совместимые с широко распространенной логикой ТТЛ, КМОП, *N*-МОП как с традиционным питанием +5 В, так и с низковольтным (3,3 В и ниже) [1].

Обе сигнальные линии имеют нагрузочные резисторы, «подтягивающие» их уровень к напряжению питания. На устройстве к каждой линии подключен приемник и передатчик типа «открытый кол-

лектор» («открытый сток»), у ведомого устройства передатчик на линии *SCL* не обязателен. Все одноименные передатчики соединяются по схеме «Монтажное И»: уровень в линии будет высоким, если все передатчики пассивны, и низким, если хоть у одного передатчика выходной транзистор открыт. В покое (*Idle*, исходное состояние шины) все передатчики пассивны.

Микросхемы с интерфейсом I^2C , как правило, имеют аппаратную поддержку протокольных функций. Протокол позволяет взаимодействовать на одной шине устройствам с различным быстродействием интерфейса. Требования к временным параметрам сигналов весьма свободные, так что на компьютерах и микроконтроллерах, не имеющих аппаратной поддержки шины I^2C , ее протокол может быть реализован даже чисто программно.

10.1.2. Протокол передачи данных I^2C

В интерфейсе I^2C участвующие в обменах устройства делятся на ведущее (*master*) и ведомое (*slave*). Ведущее и ведомое устройства могут выступать в роли и *передатчика*, и *приемника* данных. Протокол допускает наличие на шине нескольких ведущих устройств и поэтому использует механизм арбитража.

Обмен осуществляется по сигналам синхроимпульсов, которые выдает ведущее устройство, но ведомое, если оно не имеет достаточного быстродействия, может замедлять обмен данными. Диаграмма обмена для обычных устройств *F/S* представлена на рис. 10.1.

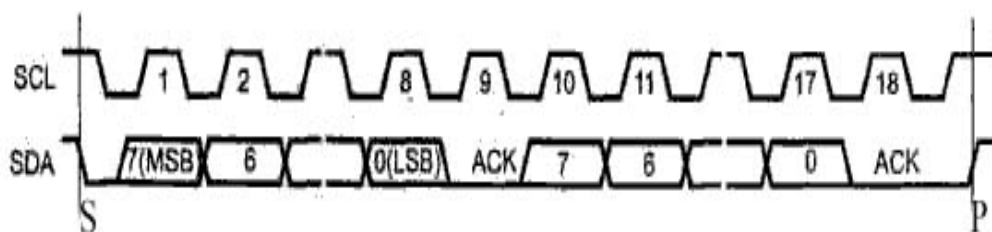


Рис. 10.1. Диаграмма передачи данных I^2C

В протоколе можно выделить несколько состояний:

– состояние **Start** (обозначается как *S*) : определяет начало любой передачи, инициируется ведущим устройством (высокий уровень сигналов *SCL* и *SDA*);

– состояние *Stop* (обозначается как P): завершает операцию переводом сигнала *SDA* из низкого уровня в высокий при высоком уровне *SCL*;

– состояние *repeated Start* (повторный старт, обозначаемый *Sr*): он предваряет начало очередной передачи вслед за текущей без ввода условия *Stop*;

– бит подтверждения *Ack* (9-й такт) о получении очередного байта.

Каждая посылка данных состоит из 8 бит, формируемых передатчиком (старший бит *MSB* передается первым).

10.1.2. Адресация

На вышеописанной физической основе строится протокол обмена данными по *I²C*. В связи с тем, что на шине *I²C* может быть несколько ведомых устройств, каждое из них должно иметь свой уникальный адрес. В начале любой передачи ведущее устройство после состояния *S* или *Sr* посылает адрес ведомого устройства или специальный адрес. Ведомое устройство с соответствующим адресом становится выбранным и обязано ответить подтверждением на адрес и последующие сигналы со стороны ведущего устройства до получения условия P или *Sr*. Разрядность адреса устройства первоначально составляла 7 бит, впоследствии был введен и режим 10-битной адресации, совместимый с 7-битной. На одной шине могут присутствовать устройства с 7-битной и 10-битной адресацией.

В варианте 7-битной адресации адрес от ведущего устройства передается в первом байте после *S* (*Sr*) в битах [7:1] (*A*[6:0]). В нулевом бите передается признак операции *RW* (*RW* = 1 – чтение, *RW* = 0 – запись). Не все адреса могут быть использованы для ведомых устройств, некоторые из них зарезервированы для специальных целей (см. таблицу в [1]). Диапазоны адресов устройств различных типов централизованно выдаются изготовителям микросхем фирмой *Philips*. Для микросхем памяти, например, 7-битный адрес содержит две части: старшие 4 бита *A*[6:3] несут информацию о типе устройства (*EEPROM* – 1010), а младшие 3 бита *A*[0:2] определяют номер устройства данного типа на шине.

Тип устройства «зашит» в его микросхеме изготовителем. Для номера устройства в микросхеме с интерфейсом *I²C* имеются три ад-

ресных входа, коммутацией которых на логические уровни 1 и 0 задается номер.

Когда ведущее устройство является передатчиком данных, оно в первом байте передает адрес ведомого устройства, при этом $RW = 0$. Выбранное ведомое устройство отзывается подтверждением ($ACK = 0$), после чего ведущее устройство посылает один или несколько байт данных, на каждый из которых ведомое устройство должно отвечать подтверждением.

Когда ведущее устройство является приемником данных, оно в первом байте передает адрес ведомого устройства с $RW=1$. Выбранное ведомое устройство также отзывается подтверждением ($ACK=0$), после чего происходит смена направления передачи, и данные уже передает ведомое устройство. Ведущее устройство подтверждает каждый принятый байт, кроме последнего. Эти передачи могут завершаться условием P , вводимым ведущим устройством, после которого шину может захватить любое ведущее устройство. Возможны комбинированные передачи, когда ведущее устройство после окончания очередного обмена не отдает шину, а формирует повторный старт (Sr), после чего обращается к тому же или иному устройству.

Для решения проблемы адресов была введена 10-битная адресация. При 10-битной адресации адрес передается в двух байтах: в первом байте после S (Sr) биты 2:1 несут старшие биты адреса, бит 0 – признак $RW = 0$; второй байт несет младшие 8 бит адреса, а последующие байты – передаваемые данные. Приемник дает подтверждения обычным способом.

Прием данных ведущим устройством несколько сложнее, поскольку признак RW является явным указателем на смену направления передачи и младшую часть адреса ведущее устройство передать уже не может. Прием по 10-битному адресу ведущее устройство начинает как фиктивную передачу: в первом байте посылает признак 10-битной записи и 2 бита адреса, во втором байте – остаток адреса. Далее ведущее устройство выполняет условие Sr и посылает признак 10-битного чтения ($RW = 1$) с двумя старшими битами того же адреса. Ведомый передатчик, получив те же 2 старшие бита адреса, что и до Sr , отвечает подтверждением и начинает передавать данные ведущему устройству. Этот процесс продолжается и завершается так же, как и с 7-битной адресацией.

10.2. Шина ACCESS.Bus

Шина *ACCESS.Bus* (*Accessory Bus*) была разработана фирмой *DEC* в 1991 г. для подключения к компьютеру сравнительно медленных внешних устройств – клавиатуры, координатных устройств и т.п. Несколько позже в шину ввели дополнительную спецификацию для взаимодействия с внутренними устройствами, например, интеллектуальными источниками питания (*Smart Battery*) и т.п. К внутренним относятся устройства системного управления *SM* (*System Management*). Последняя доступная версия *ACCESS.bus Specifications Version 3.0* опубликована *ACCESS.bus Industry Group* в 1995 г.

Шина *ACCESS.Bus* является последовательной и использует протокол интерфейса I^2C и дополнительный базовый программный протокол, с которым взаимодействуют протоколы конкретных подключенных устройств. Эти протоколы обеспечивают подключение/отключение устройств без перезагрузки ОС и автоматическое динамическое назначение адресов. Количество адресуемых устройств ограничено 125. Основным способом передачи и приема данных в *ACCESS.bus* является передача данных ведущим устройством и их прием ведомым устройством — это самый простой способ, при котором в каждой транзакции отсутствует смена направления передачи. Из этого следует, что для двустороннего обмена информацией все устройства должны поддерживать функции ведущего устройства (передатчика) и ведомого устройства (приемника).

10.2.1. Физический уровень

Физический уровень шины *ACCESS.Bus* практически полностью соответствует шине I^2C со стандартной скоростью (до 100 Кбит/с) и 7-битной адресацией ведомых устройств [1]. Здесь работают те же механизмы синхронизации и арбитража.

По электрическим сигналам имеются две спецификации для внешних и внутренних устройств соответственно.

В интерфейсе *ACCESS.bus* предусмотрено:

- использование 4-контактного экранированного модульного разъема (*MOLEX SEMCONN* или *AMP SDL*) (табл. 10.1);
- питание от интерфейса 5 В с током 50 – 1000 мА;

– максимальная суммарная длина кабеля (без повторителей) не более 10 м.

Для улучшения формы импульсов и защиты от статического электричества устройства рекомендуется подключать к линиям *SDA* и *SCL* через последовательные резисторы 51 Ом. Входы микросхем рекомендуется защищать диодами, соединенными с шинами *GND* и +5 В.

Ассоциация *VESA* для вывода внешней шины *ACCESS.Bus* на корпус мониторов предлагает иной 5-контактный разъем *ACCESS.Bus* (табл. 10.2).

Таблица 10.1. Назначение контактов внешнего разъема *ACCESS.bus*

Контакт	Назначение	Цвет провода
1	<i>GND</i>	Черный
2	<i>SDA</i>	Зеленый
3	+5В (питание устройств)	Красный
4	<i>SCL</i>	Белый

Таблица 10.2. Разъем *ACCESS.Bus (VESA)*

Контакт	Назначение
1	<i>GND</i>
2	Ключ
3	<i>SDA</i>
4	+5В (питание устройств)
5	<i>SCL</i>

10.2.2. Протокол передачи данных шины *ACCESS.Bus*

Шина *ACCESS.bus* является хост-центрической. В ней сообщения передаются от устройства к хосту и от хоста к устройству. Протокольный уровень шины *ACCESS.bus* 3.0 состоит из двух протоколов. Для устройств с программируемым адресом используется протокол *PA* (*Programmable Address*), для устройств с фиксированным адресом – протокол *FA*. В устройстве (как внешнем, так и внутреннем) может быть реализован любой из них или оба. Внутренние устройства с *FA*

называются *SM*-устройствами (*System Management*). Протокол *FA* практически соответствует шине *SMBus*, описанной ниже.

Базовый протокол *PA* основан на передаче однонаправленных сообщений. Конфигурирование устройств начинается с назначения личного адреса. После включения питания устройства должны отвечать только на «дежурный» адрес 0110111. В рабочем состоянии шина позволяет обнаруживать подключение новых устройств и их конфигурировать без перезагрузки («горячее подключение»).

Каждому устройству назначается свой адрес, на который оно должно отзываться битами подтверждения при приеме сообщения. Адрес выражается однобайтным числом, причем всегда четным, поскольку в *IC* 7-битный адрес дополняется младшим битом *RW*, нулевым в *ACCESS.bus*. Адрес *50h* всегда назначается хост-компьютеру, адрес *10h* зарезервирован для хоста *SM*-устройств. Адрес *6Eh* «дежурный», на который отзываются лишь устройства с неназначенным личным адресом. Для личных адресов устройств остаются диапазоны *02-4Eh*; *52-6Ch*; *70-FEh* – 125 адресов с некоторыми исключениями, зарезервированными для фиксированных адресов *SM*-устройств и мониторов.

Сообщения передаются в виде пакетов, формат пакета приведен на рис. 10.2 [1]. Адрес назначения *DestAddr* воспринимается получателем аппаратно (это адрес ведомого устройства *PC*). Адрес источника *SrcAddr* позволяет получателю идентифицировать источник данных (и определить, куда посылать ответ). Флаг протокола *P* позволяет различать назначение тела пакета: *P* = 0 – «полезные» данные устройства (*Device Data Stream*); *P* = 1 – управление/состояние (*control / status*). Поле *Length* определяет длину тела пакета (в байтах); само тело (*Body*) размещается в последующих байтах. Контрольный байт *Checksum* является результатом выполнения функции *XOR* (исключающее ИЛИ) над всеми предшествующими байтами пакета, начиная с адреса приемника. Признаком целостности пакета является нулевой результат функции *XOR* от всех байтов пакета, включая контрольный. Подлежат обработке только пакеты с корректным контрольным байтом. Минимальная длина всего пакета – 4 байта, максимальная – 131 байт (127 байт тело и 4 байта обрамления). Однако максимальную длину пакета ограничивает и время, разрешенное устройству для передачи пакета.

Бит/Байт	7	6	1	0
1	DestAddr — адрес приемника			0
2	SrcAddr — адрес источника			0
3	P	length		
4	Body — тело пакета			
	...			
Length + 4	Checksum — контрольный байт			

Рис. 10.2. Формат пакета сообщения *ACCESS.Bus*

Для *ACCESS.bus* определено 9 протокольных сообщений с флагом $P = 1$, обязательных для реализации интерфейсных функций шины (автоконфигурирования). «Полезными» прикладными сообщениями могут обмениваться только сконфигурированные устройства и только после явного разрешения этого обмена.

10.3. Шина *SMBus*

Шина *SMBus* (*System Management Bus* – шина системного управления) – двухпроводной интерфейс для обмена данными микросхем между собой и самим компьютером. Первая версия спецификации *SMBus* вышла в 1995 г., версия 1.1 – в 1998 г., версия 2.0 – в 2000 г. Первоначально основным назначением интерфейса было управление подсистемой питания и сопутствующими подсистемами.

Прежде всего, это предназначалось для интеллектуальных батарей и зарядных устройств, и первые версии спецификации *SMBus* шли под заголовком «*Smart Battery System Specifications*» (спецификации системы «умных» батарей). Эта система состоит из собственно батарей (аккумуляторов) и зарядных устройств, способных обмениваться управляющей информацией между собой и с хост-системой, которую она питает. Этот обмен позволяет батареям сообщать свои параметры (текущие значения, ожидаемые величины), подключаться в режим работы (питания хост-системы) или зарядки. Часть управляющих функций выполняется при участии хоста, а часть – автономно.

Спецификация 2.0 имеет трехуровневую модель взаимодействия и включает нижние уровни модели открытых систем (*OSI*) от физического до сетевого [1].

Шина *SMBus* основана на двухпроводном интерфейсе I^2C . По сравнению с I^2C в шине несколько изменены граничные требования к уровням сигналов и временным диаграммам, но в основном они совпадают. Для шины *SMBus* в *BIOS* имеется стандартизованная поддержка.

Особенностью шины *SMBus*, связанной с ее ролью в управлении системой питания, является способность автономной работы – соединяемые ею устройства могут обмениваться информацией, даже когда питание на центральном процессоре (и других подсистемах) отсутствует. Это связано с ее ролью в управлении системой питания. Такие обмены проходят без участия *BIOS*.

10.3.1. Уровни *SMBus*

1. Физический уровень (1-й уровень *OSI*) определяет две спецификации электрических и временных параметров сигналов.

Маломощная (*low power*) спецификация соответствует «родному» применению *SMBus* в устройствах с батарейным питанием; здесь характерны малые токи.

Мощная (*high power*) спецификация предназначена для работы абонентов *SMBus* в окружении источников значительных помех (например, на плате *PCI*). Маломощные и мощные устройства на одной шине вместе работать в общем случае не могут. При необходимости совместного использования устройств разных классов организуются разные сегменты шины, соединенные мостом.

В спецификациях временных параметров приняты меры по ограничению времени отклика и предотвращению «зависания» шины. Частота тактового сигнала ограничена и снизу (10 кГц), и сверху (100 кГц); введены ограничения на максимальную длительность нахождения синхросигнала в высоком и низком состояниях и максимальную суммарную «растяжку» битовых интервалов на каждый байт. Предусматривается механизм тайм-аутов, по которому устройства, обнаружившие «зависание» шины, должны немедленно прекратить обмен.

2. Уровень связи (2-й уровень *OSI*) определяет правила передачи данных. Эти правила аналогичны I^2C : условия *S*, *P*, *Sr*; биты подтверждения; 7-битная адресация и признак *RW* в первом байте, следующим за условием *S* (*Sr*). Ведущее устройство может выполнять и запись, и чтение данных ведомого устройства; используются и комбинированные транзакции (через условие *Sr*).

Некоторые особенности по сравнению с I^2C есть при генерации подтверждений. На собственный адрес устройство всегда должно отвечать битом подтверждения *ACK*, даже если оно занято внутренними операциями. Это правило обеспечивает работу механизма определения присутствия данного устройства на шине. Ведомое устройство может вводить бит *NACK* в ответ на любой неадресный байт, если оно занято или запрашиваемые командой данные недоступны. В этом случае у него есть и альтернатива поведения – задержать синхросигнал на низком уровне (в разрешенных пределах). Ведомое устройство должно вводить бит *NACK* в ответ на недопустимые коды команд или данных. Ответ *NACK* вынуждает ведущее устройство прекратить транзакцию (ввести *P*). Ведущее устройство, будучи приемником, ответом *NACK* информирует передатчик о приеме последнего ожидаемого байта.

3. Сетевой уровень *SMBus* вводит понятие «хоста» (*host*) – абонента шины, выполняющего координирующие и конфигурирующие функции. Хост является ведущим устройством шины. Однако он должен выполнять и ряд функций ведомого устройства и обрабатывать сообщения уведомления.

Адресация устройств аналогична 7-битному варианту I^2C , хотя диапазон специальных значений несколько больше, чем в I^2C . Адреса устройств разделяются по типам. Для устройств однозначно понятного назначения *SMBus WG* выделяет специальные адреса (*Purpose-assigned addresses*). Например, батареи *Smart Battery* имеют адрес 0001 011, их зарядные устройства – 0001 001. Устройства с этими адресами обязаны соответствовать требованиям *SMBus*, предъявляемым к устройствам данного класса. Ряд систем с *SMBus* определяют и используют эти устройства, основываясь на их адресе. Другие системы могут и не доверять одному только адресу, а определять типы присутствующих устройств иным образом. В спецификации *SMBus 2.0* по-

явилась возможность автоматического динамического назначения адресов устройств.

Шина *SMBus* для управления устройствами использует определенный набор команд, с помощью которых выполняется обмен данными. Команда кодируется одним байтом, передаваемым в транзакции вслед за адресным байтом.

В версии 1.1 спецификации *SMBus* введена возможность контроля ошибок пакета *PEC* (*Packet error checking*). Механизм *PEC* основан на добавлении в конец каждого передаваемого пакета байта *CRC*-кода, вычисляемого по всем предыдущим байтам пакета, начиная с адресного. Почти все протоколы могут иметь два варианта — без *PEC* и с *PEC*; на одной шине могут присутствовать устройства и с поддержкой *PEC*, и без нее.

10.3.2. Автоматическое назначение адресов

Как и в *I²C*, в шине *SMBus* предусмотрено динамическое реконфигурирование, т.е. возможность «горячего» подключения-отключения. Автоматическое назначение адресов, появившееся в версии 2.0, использует еще и контрольные байты пакетов (*PEC*).

Задача динамического реконфигурирования включает распознавание фактов подключения / отключения устройств и обеспечение бесконфликтного распределения их адресов.

Используются два способа распознавания подключения новых устройств:

1. Уведомительный – устройство, которое может работать ведущим устройством, при подключении (после своей инициализации по включению питания) может послать хосту уведомление, содержащее его адрес.

2. Опросный – включающий периодический опрос устройств ведущим устройством, ведающим «переучетом» всех устройств на шине.

Для динамического бесконфликтного назначения личных адресов устройств используется протокол *ARP* (*Address Resolution Protocol*). Назначенный адрес запоминается устройством на все время, пока подано питание. Возможны и устройства *PSA* (*Persistent Slave Address*), «вспоминающие» ранее назначенный адрес после по-

вторного включения питания. После назначения адреса обмен с устройством выполняется точно так же, как и с устройством с фиксированным адресом.

Любое ведущее устройство шины *SMBus* может выполнять назначение адресов. Для динамического назначения адреса требуется диалог ведущего устройства-нумератора с каждым устройством без помех со стороны других устройств (типичная задача настройки системы *PnP*). При этом используется уникальный идентификатор устройства *UDID* (*Unique Device Identifier*) – 128-битная структура, содержащая описание возможностей, версию, идентификаторы производителя, устройства, подсистемы и специфическую информацию. Идентификатор начинается с байта возможностей (*Device Capabilities*), в котором два старших бита характеризуют способности динамической адресации, а младший бит – поддержку *PEC*. Эти идентификаторы читаются ведущим устройством *ARP* по протоколу блочного чтения по «дежурному» адресу *SMBus*. Все устройства с еще не назначенными адресами отзываются на это чтение. Выполняется арбитраж этой операции по значению битов второго байта. Первый считанный байт (указатель длины) у всех устройств одинаков, по нему конфликтов нет. Приоритет имеют те устройства, у которых нулевое значение бит данных второго байта встретится раньше. На этом основана изоляция устройств.

10.4. Сравнение шин *I²C*, *ACCESS.Bus* и *SMBus*

Все три шины основаны на протоколе *I²C* и потому являются «близкими родственниками»; однако они имеют ряд электрических, конструктивных и протокольных отличий [1] (табл. 10.3).

Электрический интерфейс шин достаточно близок, и при обычном (5 В) питании схем проблем совместимости не возникает. Конструктивно выделяется шина *ACCESS.Bus* (внешняя), в которой определены разъемы и кабели, а также питание (5 В) устройств от шины, чего в других нет.

Протокольные различия более существенны:

– в шине *ACCESS.Bus* определены лишь передачи данных ведущим устройством, в остальных шинах ведущее устройство может и принимать данные;

Таблица 10.3. Электрические параметры шин I^2C , $ACCESS.bus$ и $SMBus$

Параметр	I^2C	$ACCESS.bus$ внешняя	$ACCESS.bus$ внутренняя	$SMBus$ мощная	$SMBus$ маломощная
Входной уровень лог. 0, не более	1,5 В или $0,3V_{dd}$	$0,3V_{dd}$	$0,3V_{dd}$	$0,8V$	$0,8V$
Входной уровень лог. 0, не менее	3,0 В или $0,7V_{dd}$	$0,7V_{dd}$	$0,7V_{dd}$	$2,1V$	$2,1V$
Входной уровень лог. 0, не более	0,4 В или $0,2V_{dd}$ при $V_{dd} < 2$ В	$0,6V$	$0,6V$	$0,4V$	$0,4V$
Выходной ток лог. 0, мА	3	6	0,35	4	0,35
Максимальная емкость линии шины, пФ	400	1000	–	400	400
Частота, кГц	0 – 100, 0 – 400, 0 – 3400	1 – 100	1 – 100	10 – 100	10 – 100

– в шинах $ACCESS.Bus$ и $SMBus$ ведущее устройство не имеет права повторного захвата шины ранее чем через 50 мкс после начала передачи. Сигнал SCL этой шине может удерживаться на низком уровне не дольше 2 мс. В I^2C таких ограничений нет;

– в шине $SMBus$ ведущее устройство ожидает от ведомого немедленного ответа, в шине $ACCESS.Bus$ ответ приходит независимо, но ожидается не позже чем через 40 мс. В шине $SMBus$ и ведущее, и ведомое устройства могут притормаживать обмен на уровне приема битов, что позволяет им вписываться в свои временные диаграммы.

10.5. Последовательный интерфейс SMI

Последовательный интерфейс SMI (*Serial Management Interface*) определен в спецификации $IEEE 802.3u$ сетевой технологии *Ethernet* как часть интерфейса MII (*Media Independent Interface* – интерфейс, не зависящий от среды передачи). Интерфейс SMI позволяет контролировать устройства физического уровня (PHY) и управлять ими, обес-

печивая доступ к внутренним регистрам устройства. Физический уровень определяет полудуплексный синхронный обмен и 3 линии шины: линию синхронизации *MDC*, информационную линию (ввода-вывода) *MDIO* и провод земли *GND*. Шиной управляет один контроллер-хост, который инициирует транзакции, а все подключенные устройства являются ведомыми. Импульсы синхронизации генерируются контроллером. По линии *MDIO* передаются команды от контроллера и данные от него (при записи) или от устройства (при чтении). Адресуемое устройство выполняет заданную команду. Сигналы линий имеют уровень ТТЛ-логики. Частота синхросигнала не должна превышать 2,5 МГц, но может быть переменной, что позволяет формировать ее программно. В протоколе реализована 5-битная адресация, что позволяет адресовать 32 устройства, каждое из которых может иметь до 32 регистров. Протокол позволяет осуществить запись и чтение 16-разрядных регистров устройства.

В соответствии с протоколом после подачи питания формируется преамбула *Pre* – не менее 32 тактов *MDIO* высокого уровня. Транзакция (рис. 10.3) [1] начинается с сигнала *Start* биты 01, далее следуют 2 бита команды *Cmd*-10 – чтение, 01 – запись. Затем передаются адрес устройства *PhyAd* (5 бит) и номер регистра *RegAd* (5 бит). После этого следует 2-битный сигнал *TA* (*turnaround*), определяющий «хозяина» шины *MDIO* при передаче информации. При записи контроллер формирует последовательность бит 10 и 16 бит данных. При чтении контроллер опускает линию *MDIO* и на втором бите адресуемое устройство устанавливает 0, сигнализируя факт ответа. Затем устройство передает 16 бит данных.

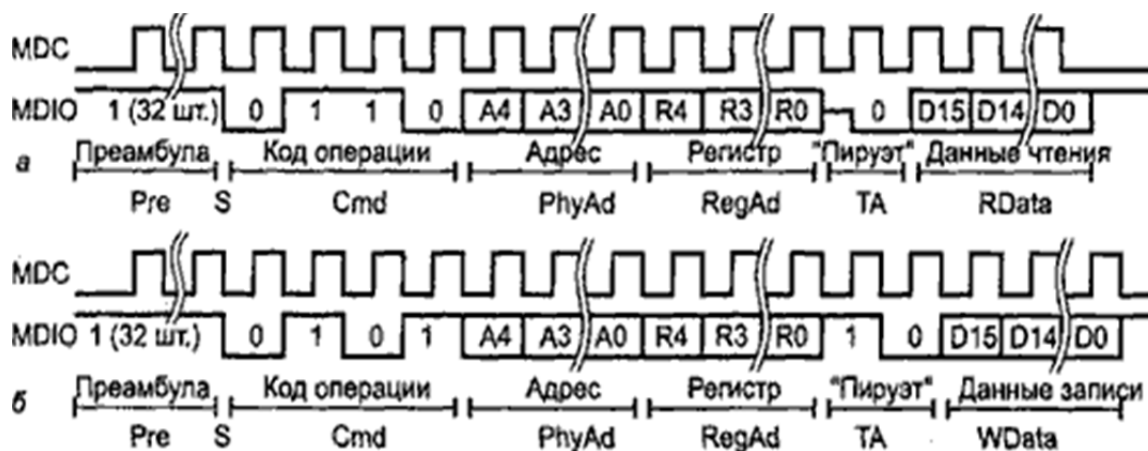


Рис. 10.3. Транзакции на шине *SMI*: а – запись; б – чтение

10.6. Интерфейс SPI

Интерфейс *SPI* (*Serial Peripheral Interface*), или *Microwire*, предназначен для связи микроконтроллеров с различными микросхемами, в том числе для программирования микросхем программируемой логики. Интерфейс синхронный и имеет 3 обязательных сигнала и 1 дополнительный. Обязательными сигналами являются *SCL* (*Serial Clock*) – синхросигнал, *MOSI* (*Master Output Slave Input*) – данные выходные для ведущего и входные для ведомого устройства, *MISO* (*Master Input Slave Output*) – данные входные для ведущего и выходные для ведомого устройства (рис. 10.4) [1]. Эти сигналы соединяются по шинной топологии (рис. 10.5).

Дополнительный сигнал *SS#* (*Slave Select* или *CS# - Chip Select*) используется для выбора ведомого устройства и имеет звездообразную топологию. Ведомое устройство передает выходные данные по линии *MISO* при низком уровне сигнала *SS#*. Если сигнал *SS#* не используется, ведомое устройство откликается на свой адрес, передаваемый в начале транзакции после кода операции. Биты данных в транзакции передаются начиная со старшего. Режимы обмена (*Mode0...Mode3*) и форматы транзакций зависят от устройств.

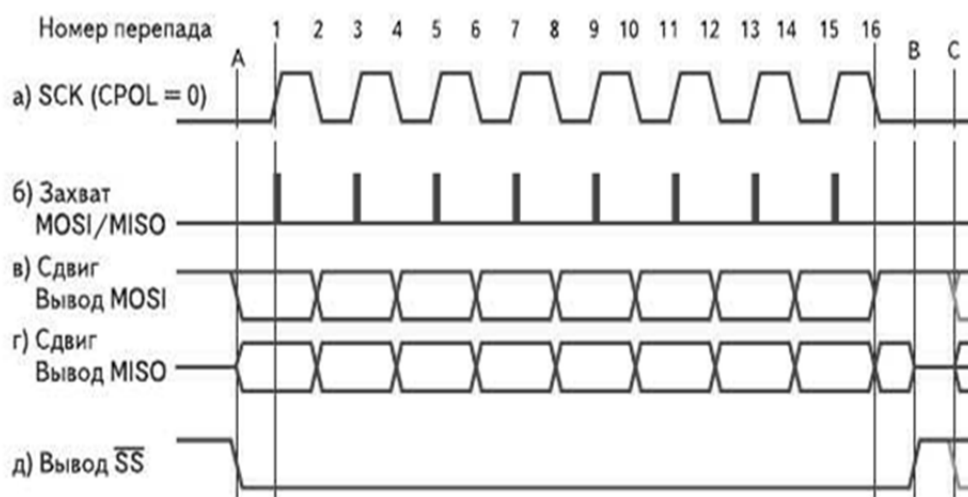


Рис. 10.4. Диаграмма SPI, режим 0

Структура связей и линии интерфейса *SPI* показаны на рис. 10.5. Могут использоваться различные схемы подключения устройств, такие как радиальная (рис. 10.6) и кольцевая (рис. 10.7).

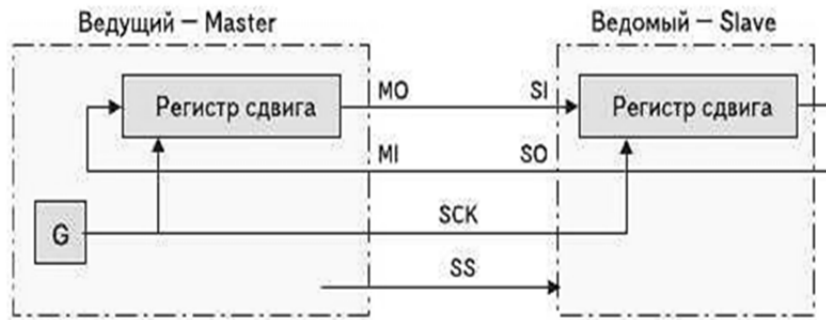


Рис. 10.5. Структура связей и линии интерфейса SPI

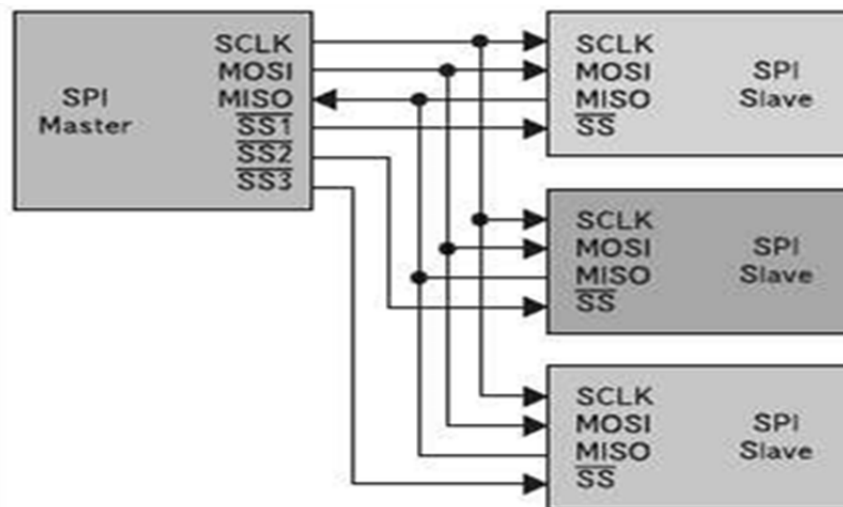


Рис. 10.6. Радиальная структура связи с несколькими ведомыми через SPI

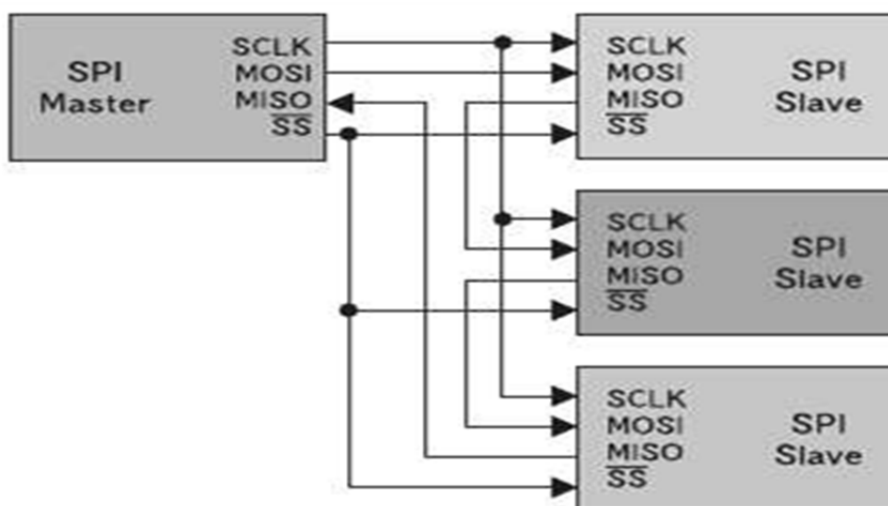


Рис. 10.7. Кольцевая топология связи через SPI

10.7. Интерфейс *JTAG*

Интерфейс *JTAG* (*Joint Test Action Group*) описан в стандарте *IEEE 1149.1 Boundary Scan Architecture* и предназначен для тестирования логических схем. Несколько тестируемых устройств может быть подключено к одному устройству – контроллеру. В роли контроллера чаще всего выступает ПК с соответствующим адаптером и ПО.

Интерфейс является последовательным и имеет 5 однонаправленных сигналов уровня ТТЛ, которые образуют тестовый порт *TAP* (*Nest Access Port*), к которому присоединяются тестируемые устройства (одно или несколько). Несколько устройств соединяются в цепочку своими портами *TAP* (рис. 10.8). Назначение сигналов интерфейса:

- *TCK* (*Test Clock*) – сигнал синхронизации, генерируемый контроллером с частотой до 16 МГц;
- *TMS* (*Test Mode Select*) – сигнал выбора тестового режима, генерируемый контроллером;
- *TDI* (*Test Data Input*) – входные данные для устройства (младшим битом вперед);
- *TDO* (*Test Data Output*) – выходные данные от устройства;
- *TRST* (*Test Logic Reset*) – необязательный сигнал сброса логики интерфейсного порта, генерируемый контроллером.

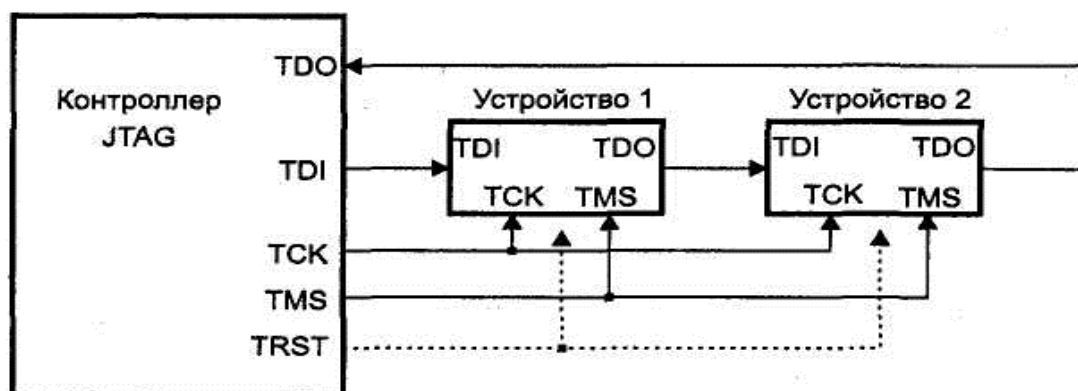


Рис. 10.8. Цепочка устройств с интерфейсом *JTAG*

Устройство, поддерживающее *JTAG*, должно иметь встраиваемую тестовую логику, включающую следующие элементы:

- 4 сигнала тестового порта *TAP*;
- *TAP*-контроллер, который управляет тестовыми регистрами;
- регистр инструкций *IR* (*Instuction Register*), принимающий последовательный код со входа *TDI*;

- регистры тестовых данных, три из которых обязательны для любого устройства: *BPR (Bupus Register)*, *DID (Device Identification Register)*, *BSP (Boundary Scan Register)*.

Более детально интерфейс *JTAG* рассмотрен в [1].

В заключение рассмотрения вспомогательных и внутрисхемных интерфейсов необходимо отметить, что все они могут быть реализованы программно на компьютере или микроконтроллере. Сигналы передаются через какой-либо порт компьютера *LPT* или *COM* с использованием адаптеров. В программе необходимо реализовать обращение к регистрам порта непосредственно или через *API*. При использовании адаптеров необходимо обратить внимание на защиту от помех.

Темы для закрепления знаний

1. Шина *I²C*.
2. Шина *ACCESS.Bus*.
3. Шина *SMBus*.
4. Сравнение шин *I²C*, *ACCESS.Bus* и *SMBus*.
5. Последовательный интерфейс *SMI*.
6. Интерфейс *SPI*.
7. Интерфейс *JTAG*.

ТЕМЫ ДЛЯ РЕФЕРАТОВ И ДОКЛАДОВ

1. Общая структура ПДП. Схема включения КПДП. Жесткие магнитные диски.
2. Схема контроллера ПДП. Гибкие магнитные диски.
3. Структура вычислительной системы (ПК) мостовой (хабовой) архитектуры.
4. Интерфейс *LPC*. Устройства ручного ввода информации (УРВИ).
5. Интерфейс *Centronics*, *IEEE1284*, ИРПР (*LPT*-порт). Матричные принтеры.
6. Интерфейс *AGP*. ЖК-мониторы.
7. Интерфейс *RS_232C*. *COM*-порт. Мыши.
8. Интерфейс *USB*. Твердотельные внешние ЗУ. (*FLESH*-карты и т.п.).
9. Интерфейс *IEEE 1394 (Fire Ware)*. Цифровые фото- и видеокамеры.

10. Инфракрасный порт. Устройства ручного ввода информации.
11. Интерфейс *Bluetooth*. Сканеры.
12. Интерфейс *SCSI*. Системы резервного копирования (СРК).
Магнитные ленточные накопители.
13. Интерфейс *IDE (PATA)*. Оптические диски *CD* и *DVD*, *BlueRay*.
14. Интерфейс *SATA*. Магнитооптические накопители.
15. Аудиосистема и ее интерфейсы.
16. Видеосистема. Структурная схема (с указанием интерфейсов) и схема видеоадаптера. Аналоговые и цифровые интерфейсы графического адаптера.
17. Интерфейс *FC*. Твердотельные внешние ЗУ. (*FLESH*-карты и т.п.).
18. ЖК-мониторы. Цифровой интерфейс монитора.
19. ЭЛТ-мониторы. Аналоговый интерфейс монитора.
20. Устройства автоматического ввода информации (сканеры, цифровые фотокамеры). Инфракрасный порт.
21. Матричные принтеры. Интерфейс *RS-232C*. *COM*-порт.
22. Лазерные принтеры. Интерфейс *USB*.
23. Струйные принтеры. Интерфейс *Centronics*, *IEEE1284*, ИРПР (*LPT*-порт).
24. Принтеры *3D*. Интерфейс *SCSI*.
25. Способы представления (кодирования) цифровой информации при записи на магнитный и другие виды носителей. Интерфейс *SATA*.
26. Жесткие магнитные диски. Интерфейс *IDE (PATA)*.
27. Гибкие магнитные диски. Интерфейс *LPC*.
28. Оптические диски *CD*, *DVD* и *BlueRay*. Интерфейс *USB*.
29. Магнитооптические накопители. Интерфейс. *IEEE 1394 (Fire Ware)*.
30. Системы резервного копирования (СРК). Магнитные ленточные накопители.

Содержание реферата (отчета)

1. Назначение устройства (интерфейса).
2. Параметры и характеристики и их типовые значения.
3. Фирмы-производители.
4. Принцип работы.
5. Представление информации.

6. Функциональные и (или) структурные схемы устройства и его узлов.
7. Электрические параметры.
8. Конструктивные особенности.
9. Протоколы и временные диаграммы.
10. Литература и источники.

ЗАКЛЮЧЕНИЕ

В предлагаемом учебном пособии рассмотрены основные разделы, предусмотренные ГОС и программой дисциплины «ЭВМ и периферийные устройства», читаемой во второй части студентам направления 230100 «Информатика и вычислительная техника». Учебный материал знакомит читателя с системами ввода-вывода ЭВМ и конкретными интерфейсами, используемыми в них.

Книга дает представление о принципах построения и функционирования каналов ввода-вывода, их параметрах, способах организации обмена. Рассмотрена организация системы прерываний в ЭВМ вообще и в ПК в частности. Кратко описана видеосистема ЭВМ как наиболее ресурсоемкая и быстроразвивающаяся.

В настоящее время появляется большое количество новых интерфейсов, поэтому при рассмотрении интерфейсов как традиционных, так и новых, современных, основное внимание уделяется принципам построения и функционирования. Меньшее внимание уделено справочным материалам, которые можно найти в литературе и интернете. Рассматриваются тенденции развития систем ввода-вывода ЭВМ и интерфейсов.

Дисциплина «ЭВМ и периферийные устройства» является одной из основных в подготовке специалиста и должна дать конкретный материал, необходимый в практической деятельности. Поэтому в учебное пособие включены некоторые необходимые справочные сведения. Для студентов заочной формы обучения приведены темы рефератов, необходимые для самостоятельной работы.

Интерфейсы периферийных устройств изучаются не только студентами направления 230100 «Информатика и вычислительная техника», но и студентами других специальностей и направлений в рамках родственных дисциплин, такой как «Вычислительные средства информационных систем» и т.п. Предлагаемое учебное пособие может быть полезно и этим студентам.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Гук, М. Аппаратные интерфейсы ПК : энциклопедия / М. Гук. – СПб. : Питер, 2002. – 528 с. – ISBN 5-94723-180-8.
2. Он же. Аппаратные средства IBM PC : энциклопедия / М. Гук. – 3-е изд., стер. – СПб. : Питер, 2006. – 928 с. – ISBN 5-318-00047-9.
3. Он же. Шины PCI, USB и FireWire : энциклопедия / М. Гук. – СПб. : Питер, 2005. – 528 с. – ISBN 5-94723-180-8.
4. Он же. Интерфейсы устройств хранения: ATA, SCSI и другие : энциклопедия / М. Гук. – СПб. : Питер, 2007. – 447 с. – ISBN 5-469-01531-9.
5. Мюллер, С. Модернизация и ремонт ПК / С. Мюллер. – 12-е изд., стер. – СПб. : Вильямс, 2001. – 1184 с. – ISBN 5-8459-0167-7.
6. Танненбаум, Э. Архитектура компьютера / Э. Танненбаум. – 4-е изд., стер. – СПб. : Питер, 2003. – 704 с.
7. Кулаков, В. Программирование дисковых подсистем / В. Кулаков. – СПб. : Питер, 2002. – 768 с. – ISBN 5-318-00623-X.
8. Организация ЭВМ / К. Хамахер [и др.]. – 5-е изд., стер. – СПб. : Питер ; Киев : BHV, 2003. – 848 с. – ISBN 5-8046-0162-8.

ОГЛАВЛЕНИЕ

ПРЕДИСЛОВИЕ.....	3
ВВЕДЕНИЕ.....	4
ГЛАВА 1. СТРУКТУРА СИСТЕМЫ ВВОДА-ВЫВОДА ЭВМ	8
1.1. Требования к системе ввода-вывода, ее функции и схемы.....	8
1.2. Интерфейсы.....	12
1.3. Каналы ввода-вывода	21
1.4. Контроллеры.....	28
Темы для закрепления знаний	29
ГЛАВА 2. ШИНЫ РАСШИРЕНИЯ ПК	31
2.1. Характеристики шин расширения	31
2.2. Шины и карты расширения мобильных ПК.....	32
2.3. Интерфейс <i>LPC</i>	33
2.4. Шина <i>PCI</i>	36
2.5. Интерфейс <i>AGP</i>	39
2.6. Интерфейс <i>PCI-Express</i>	43
Темы для закрепления знаний	55
ГЛАВА 3. ТРАДИЦИОННЫЕ ИНТЕРФЕЙСЫ ПК	57
3.1. Интерфейс <i>RS-232C</i>	57
3.2. <i>COM</i> -порт	68
3.3. Интерфейс <i>Centronics IEEE-1284</i> и <i>LPT</i> -порт	72
3.4. <i>Game</i> -порт.....	79
Темы для закрепления знаний	81
ГЛАВА 4. ПОСЛЕДОВАТЕЛЬНЫЕ ИНТЕРФЕЙСЫ.....	82
4.1. Интерфейс <i>USB</i>	82
4.2. Интерфейс <i>IEEE-1394 (FireWare)</i>	107
4.3. Интерфейс <i>Thunderbolt</i>	111
4.4. Сравнение характеристик последовательных интерфейсов	115
Темы для закрепления знаний	116
ГЛАВА 5. ИНТЕРФЕЙСЫ ВНЕШНИХ ЗАПОМИНАЮЩИХ УСТРОЙСТВ (ВЗУ)	117
5.1. Интерфейсы жестких дисков <i>IDE(ATA)</i>	117
5.2. Интерфейс <i>SerialATA</i>	121
5.3. Интерфейс <i>SCSI</i>	129

5.4. Технология маршрутизации команд	147
Темы для закрепления знаний	147
ГЛАВА 6. БЕСПРОВОДНЫЕ ИНТЕРФЕЙСЫ	148
6.1. <i>Bluetooth</i>	149
6.2. Технология <i>Ultra Wideband (UWB)</i> и интерфейс <i>Wireless USB</i>	162
6.3. Инфракрасный порт.....	175
Темы для закрепления знаний	183
ГЛАВА 7. ИНТЕРФЕЙСЫ КЛАВИАТУРЫ И МЫШИ	184
7.1. Интерфейс клавиатуры.....	184
7.2. Интерфейс мыши	188
Темы для закрепления знаний	191
ГЛАВА 8. ИНТЕРФЕЙСЫ АУДИОУСТРОЙСТВ	191
8.1. Устройство аудиосистемы ПК.....	192
8.2. Аналоговые интерфейсы аудиоустройств	193
8.3. Цифровые интерфейсы аудиоустройств	194
8.4. Интерфейс <i>MIDI</i>	196
8.5. Интерфейс дочерней карты.....	198
Темы для закрепления знаний	199
ГЛАВА 9. ИНТЕРФЕЙСЫ ДИСПЛЕЕВ	199
9.1. Аналоговые интерфейсы.....	200
9.2. Цифровые интерфейсы	201
Темы для закрепления знаний	205
ГЛАВА 10. ВНУТРИПЛАТНЫЕ ИНТЕРФЕЙСЫ	206
10.1. Шина <i>I²C</i>	206
10.2. Шина <i>ACCESS.Bus</i>	210
10.3. Шина <i>SMBus</i>	213
10.4. Сравнение шин <i>I²C</i> , <i>ACCESS.Bus</i> и <i>SMBus</i>	217
10.5. Последовательный интерфейс <i>SMI</i>	218
10.6. Интерфейс <i>SPI</i>	220
10.7. Интерфейс <i>JTAG</i>	222
Темы для закрепления знаний	223
ТЕМЫ ДЛЯ РЕФЕРАТОВ И ДОКЛАДОВ	223
ЗАКЛЮЧЕНИЕ	225
БИБЛИОГРАФИЧЕСКИЙ СПИСОК	226

Учебное издание

БЫКОВ Валерий Ильич

СИСТЕМА ВВОДА-ВЫВОДА ЭВМ И ВС
И ЕЕ ИНТЕРФЕЙСЫ

Учебное пособие

Редактор А. П. Володина
Технический редактор Н. В. Тупицына
Корректор Е. П. Викулова
Компьютерная верстка Е. Кузьмина

Подписано в печать 18.09.15.
Формат 60x84/16. Усл. печ. л. 13,25. Тираж 52 экз.

Заказ

Издательство

Владимирского государственного университета
имени Александра Григорьевича и Николая Григорьевича Столетовых.
600000, Владимир, ул. Горького, 87