



Содержание

1	Стандартный параллельный интерфейс Centronics (LPT)	1
2	Параллельные периферийные адаптеры	3
3	Интерфейс RS-232 (последовательный порт - COM).....	12
4	Последовательные периферийные адаптеры	15
4.1	Fibre Channel.....	22
5	SCSI.....	25
5.1	Стандарты	25
6	Сетевые интерфейсы	40
6.1	Стандарты IEEE 802.3	42

1 Стандартный параллельный интерфейс Centronics (LPT)

Основным назначением интерфейса Centronics (аналог-ИРПП-М) является подключение к компьютеру принтеров различных типов. Поэтому распределение контактов разъема, назначение сигналов, программные средства управления интерфейсом ориентированы именно на это использование. В то же время с помощью данного интерфейса можно подключать к компьютеру и другие внешние устройства, имеющие разъем Centronics, а также специально разработанные УС. Основным достоинством использования Centronics для подключения УС по сравнению с ISA является значительно меньший риск вывести компьютер из строя. Главный недостаток этого подхода - значительно меньшая скорость обмена. Назначение 36 контактов разъема Centronics приведено в таблице 1.

Таблица 1. Назначение контактов разъемов Centronics



Вывод	Наименование	Направление	Описание
1	/STROBE	Out	Strobe (Строб)
2	D0	Out	Data Bit 0
3	D1	Out	Data Bit 1
4	D2	Out	Data Bit 2
5	D3	Out	Data Bit 3
6	D4	Out	Data Bit 4
7	D5	Out	Data Bit 5
8	D6	Out	Data Bit 6
9	D7	Out	Data Bit 7
10	/ACK	In	Acknowledge (Подтверждение)
11	BUSY	In	Busy (Занято)
12	PE	In	Paper End (Конец бумаги)
13	SEL	In	Select (Выбор)
14	/AUTOFD	Out	Autofeed (Перевод строки)
15	/ERROR	In	Error (Ошибка)
16	/INIT	Out	Initialize (Инициализация)
17	/SELIN	Out	Select In (Выбор)
18	GND	-	Signal Ground (Корпус)
19	GND	-	Signal Ground (Корпус)
20	GND	-	Signal Ground (Корпус)
21	GND	-	Signal Ground (Корпус)
22	GND	-	Signal Ground (Корпус)
23	GND	-	Signal Ground (Корпус)
24	GND	-	Signal Ground (Корпус)
25	GND	-	Signal Ground (Корпус)

Сигналы Centronics имеют следующее назначение (тип выходных каскадов для всех сигналов - ТТЛ):

D0...D7 - 8-разрядная шина данных для передачи из компьютера в принтер. Логика сигналов положительная.

-STROBE - сигнал стробирования данных. Данные действительно как по переднему, так и по заднему фронту

этого сигнала. Сигнал говорит приемнику (принтеру), что можно принимать данные.

-ACK - сигнал подтверждения принятия данных и готовности приемника (принтера) принять следующие данные. То есть здесь реализуется асинхронный обмен.

BUSY - сигнал занятости принтера обработкой полученных данных и неготовности принять следующие данные. Активен также при переходе принтера в состояние off-line или при ошибке, а также при отсутствии бумаги. Компьютер начинает новый цикл передачи только после снятия -ACK и после снятия BUSY.

-AUTO FD - сигнал автоматического перевода строки. Получив его, принтер переводит каретку на следующую строку. Остальные сигналы не являются, вообще говоря, обязательными.

PE - сигнал конца бумаги. Получив его, компьютер переходит в режим ожидания. Если в принтер вставить лист бумаги, то сигнал снимается.

SLCT - сигнал готовности приемника. С его помощью принтер говорит о том, что он выбран и готов к работе. У многих принтеров имеет постоянно высокий уровень.

-SLCT IN - сигнал принтеру о том, что он выбран и последует передача данных.

-ERROR - сигнал ошибки принтера. Активен при внутренней ошибке, переходе принтера в состояние off-line или при отсутствии бумаги. Как видим, здесь многие сигналы дублируют друг друга.

-INIT - сигнал инициализации (сброса) принтера. Его длительность не менее 2,5 мкс. Происходит очистка

буфера печати.

Временная диаграмма цикла передачи данных представлена на рисунке 1.

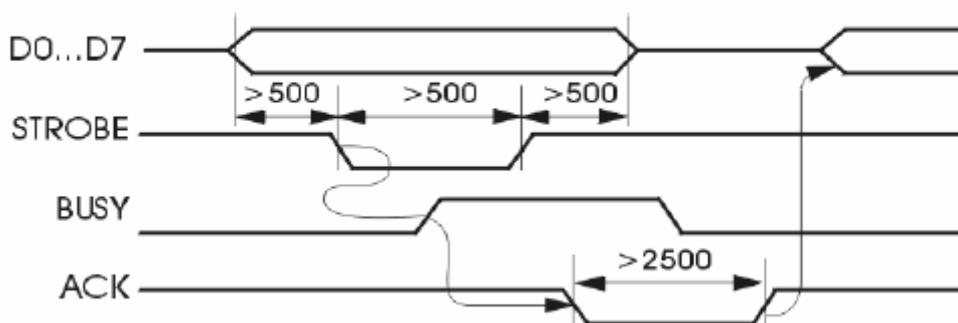


Рис.1. Временные диаграммы цикла передачи данных в Centronics (все временные интервалы в наносекундах).

Перед началом цикла передачи данных компьютер должен убедиться, что сняты сигналы BUSY и -ACK.

После

этого выставляются данные, формируется строб, снимается строб, и снимаются данные. Принтер должен успеть принять данные с выбранным темпом. При получении строба принтер формирует сигнал BUSY, а после окончания обработки данных выставляет сигнал -ACK, снимает BUSY и снимает -ACK. Затем может начинаться новый цикл.

Все сигналы интерфейса Centronics передаются в уровнях ТТЛ и рассчитаны на подключение одного стандартного входа ТТЛ. Максимальная длина соединительного кабеля по стандарту - 1,8 м.

Как видно из таблицы 1.10, в интерфейсе Centronics для подключения к компьютеру произвольных УС

мы

можем использовать 17 линий, назначение которых можно выбирать по своему усмотрению.

Формирование и прием сигналов интерфейса Centronics производится путем записи и чтения выделенных для

него портов ввода/вывода. В компьютере может использоваться три порта Centronics, обозначаемых LPT1 (базовый адрес 378h), LPT2 (базовый адрес 278h) и LPT3 (базовый адрес 3BCh). При этом LPT3 используется в том случае, когда контроллер принтера находится на плате графического адаптера Hercules или EGA. Прерывания портов принтеров (IRQ5 для LPT2 и IRQ7 для LPT1) используются очень редко.

Базовый адрес порта используется для передачи принтеру байта данных. Установленные на линиях данные

можно считать из этого же порта. Следующий адрес (базовый + 1) служит для чтения битов состояния принтера (бит 3 соответствует сигналу -EEROR, бит 4 - сигналу PE, бит 6 - сигналу -ACK, бит 7 - сигналу BUSY). Последний используемый адрес (базовый + 2) предназначен для записи битов управления принтером (бит 0 соответствует сигналу -STROBE, бит 1 – сигналу -AUTO FD, бит 2 - сигналу -INIT, бит 3 - сигналу -SLCT IN и наконец бит 4, равный единице, разрешает прерывание от принтера).

2 Параллельные периферийные адаптеры

ИРПР Угрюмов стр 306-314

§ 6.3. Параллельные периферийные адаптеры

Шинные формирователи и порты осуществляют лишь непосредственную или буферизованную во времени передачу данных между МП и шиной данных. Более сложные операции выполняются периферийными адаптерами. Программируемость адаптеров обеспечивает им широкую область применения вследствие изменяемости процедур обмена без изменений в схеме (с помощью команд программы), в том числе и во время работы микропроцессорной системы.

В схемах, обслуживающих обмен параллельными данными, как правило, используется базовая структура параллельного адаптера Intel 8255A, имеющего отечественный аналог К580ВВ55А. Эти БИС представляют собою однокристалльные устройства параллельного ввода/вывода и обеспечивают двунаправленный обмен с квити́рованием или без него при программном обмене или обмене по прерываниям. С их помощью ВУ, работающие с параллельными кодами, связываются с магистралью системы.

Параллельный периферийный адаптер (ППА, PPI) типа 55А (рис. 6.3) имеет три двунаправленных 8-разрядных порта PA, PB и PC, причем порт PC разделен на два четырехразрядных канала: старший PC_H и младший PC_L. Обмен информацией между каналами А, В, С и шиной данных МПС производится через буфер данных ВД в соответствии с сигналами управления.

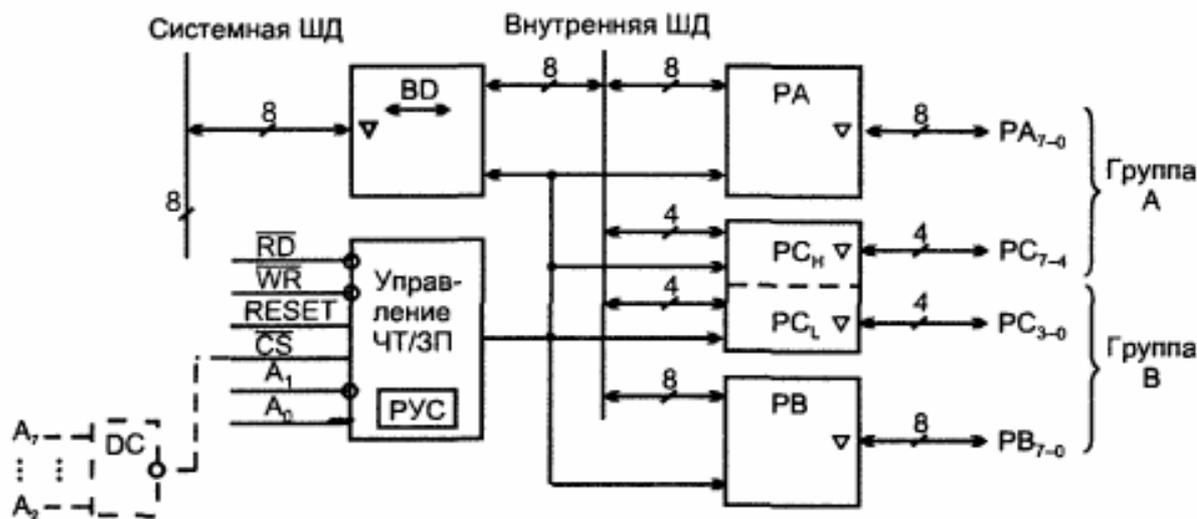


Рис. 6.3. Структура параллельного периферийного адаптера

Блок управления чтением/записью получает стробы чтения и записи \overline{RD} и \overline{WR} (это сигналы \overline{IOR} и \overline{IOW} стандартного интерфейса), сигнал сброса RESET, сигнал выбора адаптера \overline{CS} , получаемый декодированием старших



разрядов его адреса, и два младших разряда адреса A_1 и A_0 для адресации внутренних регистров. Адресуемых объектов 5: три порта (А, В и С), регистр управляющего слова РУС и команда установки/сброса битов порта С BSR (Bit Set/Reset). Адресация и направление передач информации определяются согласно табл. 6.2.

Таблица 6.2

A_1	A_0	\overline{RD}	\overline{WR}	\overline{CS}	Операция
0	0	0	1	0	Порт А → ШД
0	1	0	1	0	Порт В → ШД
1	0	0	1	0	Порт С → ШД
1	1	0	1	0	Запрещенная комбинация
0	0	1	0	0	ШД → Порт А
0	1	1	0	0	ШД → Порт В
1	0	1	0	0	ШД → Порт С
1	1	1	0	0	ШД → РУС при D7 = 1 ШД → BSR при D7 = 0
X	X	1	1	0	Шины отключены
X	X	X	X	1	Шины отключены

Как видно из таблицы, адрес $A_1A_0 = 11$ соответствует передаче управляющих слов РУС (УС1) или BSR (УС2), причем чтение по этому адресу запрещено, допускается только запись. Передача двух разных УС при одном и том же адресе возможна только потому, что признаком того или иного УС служит значение старшего бита слов D7. Таким образом, этот бит выполняет дополнительную адресацию управляющих слов.

Работа адаптера начинается после загрузки с ШД в РУС управляющего слова УС1, задающего портам адаптера один из трех возможных режимов и направленность порта (ввод или вывод).

Возможны *три режима работы портов*: 0, 1 и 2, причем порт А может работать в любом из трех режимов, порт В только в двух (0 и 1), а режим порта С зависит от режимов портов А и В.

Порт С имеет особенности, в отличие от портов А и В, которые оперируют со словами в целом, разряды порта С могут программироваться и использо-



ваться поодиночке. В частности, любой из восьми разрядов порта С может быть установлен или сброшен программным способом. Это нужно для передачи сигналов квитирования при обмене через порты А и В в режимах 1 и 2. При работе порта в режиме 1 для него требуются три линии под сигналы управления, в режиме 2 — пять.

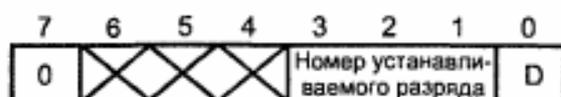
Режимы работы портов:

- режим 0 — однонаправленный ввод/вывод без квитирования, в этом режиме могут работать порты А и В, а также свободные (не занятые передачей служебных сигналов для портов А и В) линии порта С;
- режим 1 — однонаправленный ввод/вывод с квитированием;
- режим 2 — двунаправленный ввод/вывод с квитированием.

Квитирование, как известно, позволяет вести асинхронный обмен с учетом готовности абонента к передаче, т. е. иметь переменный темп обмена соответственно возможностям внешнего устройства.



а



б

Рис. 6.4. Форматы управляющих слов параллельного периферийного адаптера

Формат управляющего слова УС1 показан на рис 6.4, а. Разряд 7 содержит единицу, что является признаком управляющего слова УС1. Разряды 6–3 определяют режим и вид портов А и свободных от служебных сигналов линий порта С_n (старшей половины порта), а разряды 2–0 — то же для порта В и младшей половины порта С (С_l).

Режим порта А выбирается по условиям 00 — режим 0, 01 — режим 1, 1X — режим 2. Порт В имеет режим 0 или 1 при нулевом или единичном значении разряда 2 соответственно. Единичные значения разрядов 4, 3, 1 означают ввод, нулевые — вывод.



При записи нового УС1 все регистры портов сбрасываются. Управляющее слово УС2 задает значения 0 или 1 одному из разрядов порта С. Для приведения в определенное состояние нескольких выходов порта С нужно подать в адаптер соответствующее число слов УС2. В итоге словами УС2 на выходах порта РС задаются коды, определяющие режим работы ВУ и изменяемые программным способом.

Формат управляющего слова УС2 показан на рис. 6.4, б. Признаком этого слова служит нулевое значение разряда 7. Разряды 6...4 не используются. В разрядах 3...1 размещается двоичный код номера разряда, приводимого в то или иное состояние в порте С данным УС2. В нулевом разряде указывается состояние (0 или 1), которое следует придать данному разряду.

Режим 0

В режиме 0 осуществляется прямой однонаправленный ввод-вывод данных без сигналов их сопровождения. Каждый из 4-х портов может быть использован для ввода или вывода независимо от других, так что возможны 16 вариантов режима 0. При вводе поступающая из ВУ информация адаптером не фиксируется и должна присутствовать на его входе во время действия сигнала чтения. При выводе информация от МП фиксируется в буферном регистре порта по заднему фронту сигнала записи и сохраняется до нового цикла вывода или смены режима работы порта.

При вводе информация выдается на ШД при выполнении микропроцессором команды IN port, при выводе — при выполнении команды OUT port.

Такой вариант соответствует работе "с отдельной шиной", при которой внешним устройствам принадлежит отдельное адресное пространство. Не исключается и организация обращения к портам, как к ячейкам памяти (интерфейс "с общей шиной").

Режим 1

В режиме 1 каждая из двух 12-разрядных групп (А и В) может быть запрограммирована на однонаправленный ввод или вывод с квитированием. При этом входные и выходные данные фиксируются адаптером. По линиям портов C_H и C_L передаются управляющие сигналы. Раздельная установка разрядов порта С позволяет ему играть роль схемы управления процедурами ввода-вывода, причем битам порта придается определенное функциональное назначение.

Режим 1 рассмотрим в полном объеме, т. к. он хорошо иллюстрирует принципы работы адаптера. При вводе используются следующие управляющие сигналы:

- \overline{STB} — строб загрузки данных в регистр (по заднему фронту);
- IBF (Input Buffer Full) — входной буфер полон, сигнал подтверждения загрузки данных;
- INT — запрос прерывания.

Временные диаграммы процесса ввода в режиме 1 показаны на рис. 6.5, а.

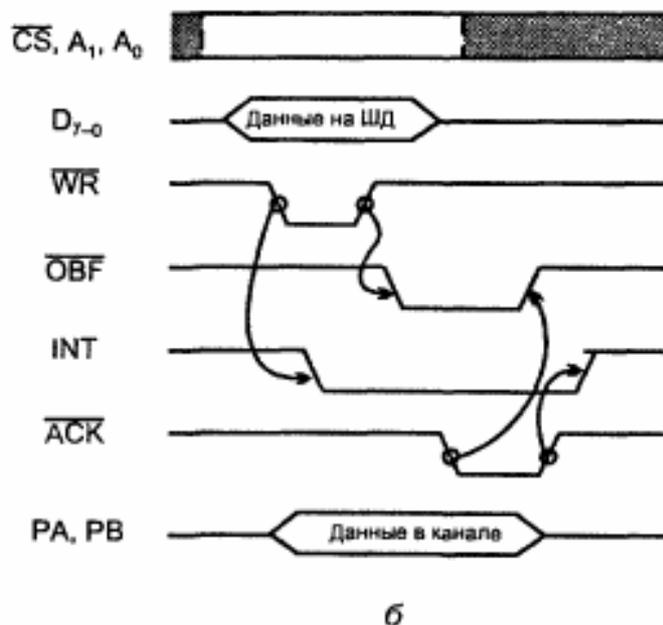
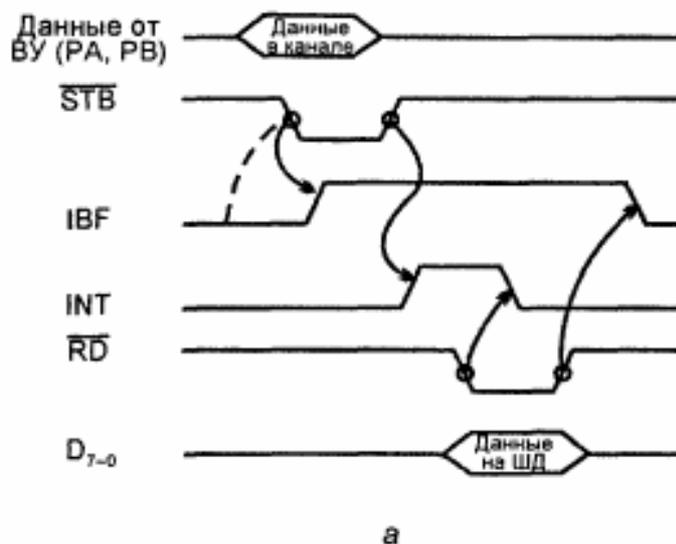


Рис. 6.5. Временные диаграммы процессов ввода (а) и вывода (б) в режиме 1 параллельного периферийного адаптера

Имея данные для ввода в порт, ВУ при условии $IBF = 0$ вырабатывает сигнал готовности информации \overline{STB} . Передний фронт этого сигнала устанавливает сигнал IBF , запрещающий внешнему устройству ввод следующего слова до освобождения адаптера (того порта, который имеется в виду). К моменту окончания \overline{STB} данные введены в буфер порта, и, если прерывания разрешены (внутренний триггер разрешения прерываний $INTE$ установлен командой программы), то адаптер формирует запрос прерывания INT для МП, переходящего к подпрограмме обслуживания, содержащей команду $IN port$. При этом на адаптер поступают сигналы адресации и \overline{RD} . Передний фронт \overline{RD} отмечает начало считывания слова микропроцессором



и снимает запрос на прерывание INT. Пока прерывания не разрешены, осуществляется хранение данных в адаптере. Задний фронт \overline{RD} отмечает завершение считывания слова микропроцессором и снимает сигнал IBF, допуская новую запись слова со стороны ВУ.

При выводе используются следующие управляющие сигналы:

- \overline{OBF} (Output Buffer Full) — выходной буфер полон, строб вывода новых данных;
- \overline{ACK} (Acknowledge) — подтверждение приема внешним устройством;
- INT — запрос прерывания.

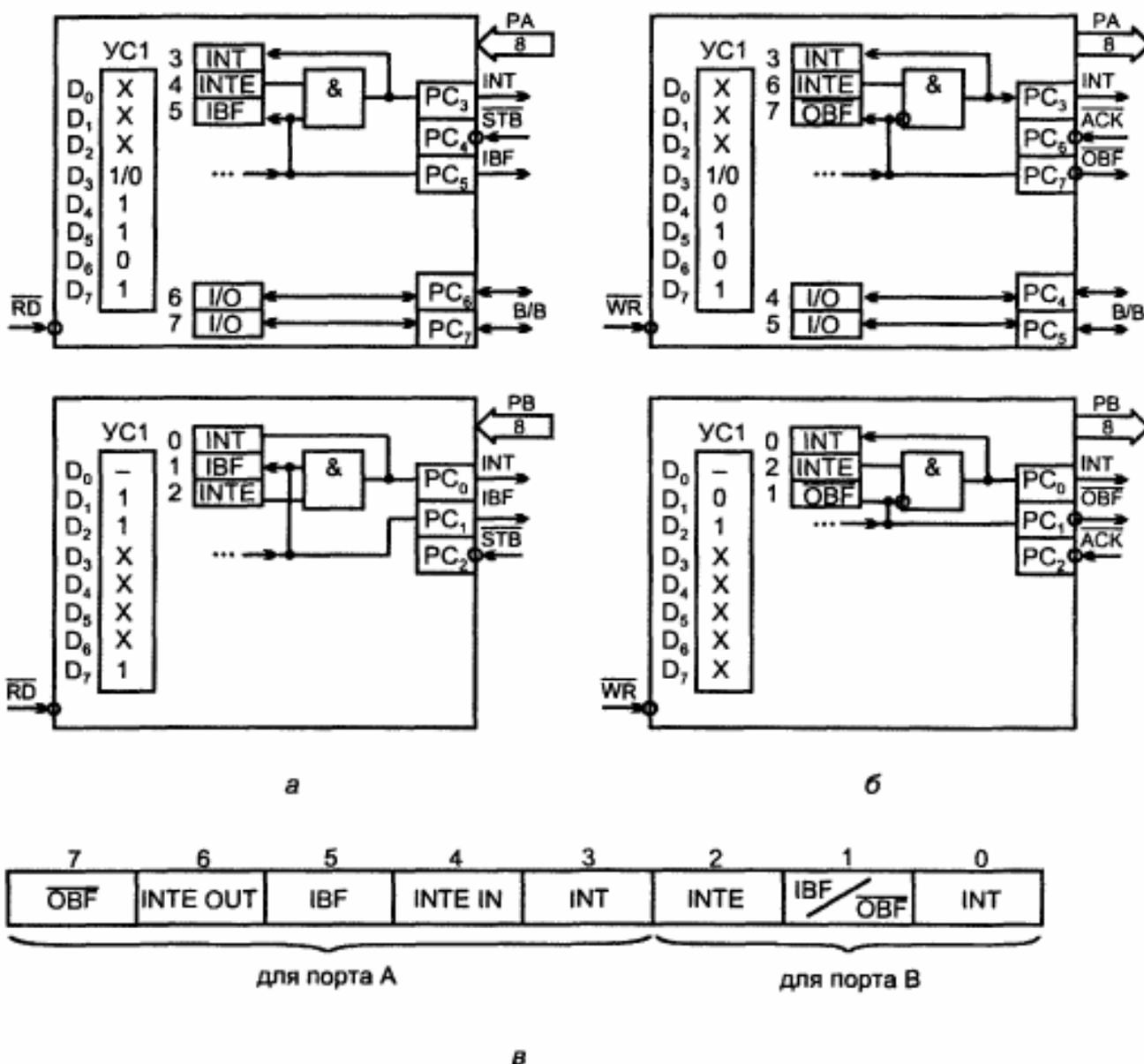


Рис. 6.6. Структура порта С, формирование слова состояния для режимов 1 параллельного периферийного адаптера (а, б) и формат слова состояния (в)



Временные диаграммы вывода в режиме 1 показаны на рис. 6.5, б. При выводе выполняется команда `OUT port`, и процессор устанавливает адрес порта и данные на ШД. При разрешенных прерываниях далее вырабатывается сигнал \overline{WR} , загружающий данные с ШД в буфер адаптера и сбрасывающий запрос прерывания `INT`. После окончания записи в адаптер формируется сигнал \overline{OBF} , указывающий на готовность данных для ВУ. Приняв данные, ВУ выдает сигнал подтверждения приема \overline{ACK} , снимающий \overline{OBF} , а по окончании сигнала \overline{ACK} восстанавливается запрос прерывания (если триггер `INTE` установлен), что вызывает обслуживание следующего цикла вывода.

Сигналы управления при обменах с квитированием передаются по отдельным линиям порта С, специально для них предназначенным. Распределение управляющих сигналов по этим линиям и формирование в адаптере слова состояния для ввода и вывода в режиме 1 показано на рис. 6.6, а, б.

К битам `INT` и `IBF` предусмотрен программный доступ через чтение порта С. В состав слова состояния входят также флажки `INTE`, управляемые командой сброса/установки битов порта С. Генерация запроса прерывания `INT` и установка связанного с ним одноименного флажка готовности в слове состояния возможны только при установленном флажке `INTE`. Это означает возможность маскирования запросов прерывания, позволяющего запрещать или разрешать работу ВУ.

Формат слова состояния показан на рис. 6.6, в. В слове состояния имеются независимые сигналы разрешения прерываний для ввода и вывода. Контроль текущего состояния портов в режимах 1 и 2 путем считывания порта С командой `IN port` позволяет анализировать процесс обмена, которым можно оперативно управлять.

Свободные линии порта С могут быть использованы для простого ввода/вывода.

Режим 2

Особенности функциональной схемы порта А допускают его применение для двунаправленной передачи между ШД и ВУ. При этом 5 линий порта С передают управляющие сигналы.

Двунаправленный асинхронный обмен через порт А выполняется как последовательность нескольких независимых этапов: записи с ШД в адаптер, ввода в адаптер из ВУ, чтения на ШД, вывода в ВУ, некоторые из которых могут совмещаться во времени. Используются сигналы управления: \overline{STB} , `IBF`, \overline{OBF} , \overline{ACK} , `INT`, т. е. те же, что и для режима 1.

Ввод в адаптер управляющих слов `UC1` и `UC2` производится программным способом с помощью последовательности команд непосредственной загрузки аккумулятора и вывода данных в адресованный порт. На языке ассемблера фрагмент программы имеет вид:

```
MVI A, b2
```

```
OUT port,
```



где загружаемый в аккумулятор байт b_2 представляет собою вводимое в адаптер слово $UC1$ или $UC2$, а $port$ — адрес регистров управления, шесть старших разрядов которого дают номер (адрес) адаптера, а два младших содержат единицы. Указанный фрагмент программы повторяется столько раз, сколько необходимо для задания адаптеру режима и функций, а выходам порта C нужных значений.

Подробное описание рассмотренного ППА имеется в работах [23], [37], [41], [45] и др.

Улучшенный вариант адаптера $VB55A$ отличается от предшественника $VB55$ работой с расширенным стробом записи, свойственным, в частности, и микропроцессору $K1821BM85A$.

Для связи с периферийными устройствами, удаленными от МПС (на расстояние не более 15 м), применяется интерфейс ИРПП (интерфейс радиальный параллельный), осуществляющий однонаправленные асинхронные передачи по 8- или 16-разрядной шине (в базовом варианте). Логические требования интерфейса ИРПП могут быть выполнены при использовании адаптера $VB55/55A$.

Пример применения ППА

На рис. 6.7 дан пример использования ППА в схеме подключения аналого-цифрового преобразователя (АЦП) и цифро-аналогового преобразователя (ЦАП) к МПС, выполняющей задачу цифрового управления некоторым аналоговым объектом.

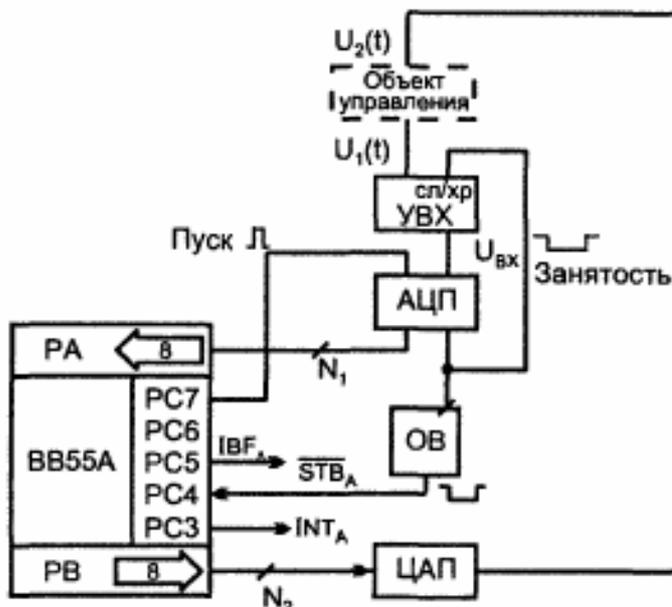


Рис. 6.7. Пример схемы использования параллельного периферийного адаптера для подключения АЦП и ЦАП к шинам микропроцессорной системы

Состояние объекта отображается сигналом напряжения постоянного тока $U_1(t)$, которое преобразуется в цифровой код N_1 и передается через адаптер процессору. Процессор согласно алгоритму управления объектом вырабатывает сигнал



3 Интерфейс RS-232 (последовательный порт - COM).

Последовательный интерфейс предназначен для передачи данных и использует одну сигнальную линию для передачи в одном направлении.

Универсальный асинхронный приёмопередатчик (УАПП, UART, Universal Asynchronous Receiver/Transmitter) — вид контроллера, устройства, которое переводит данные из последовательной в параллельную форму (и обратно). Обычно используется вместе со стандартами связи такими как [EIA RS-232](#), RS422, RS485, CL20, CL40, СТЫК-2, IRDA.

UART представляет собой отдельное устройство или является частью интегральной схемы, используется для передачи данных через последовательный порт компьютера или периферийного устройства. UART часто встраивают в [микроконтроллеры](#).

Ранее последовательный порт использовался для подключения терминала, позже для модема или мыши. Сейчас он используется для соединения с источниками бесперебойного питания, для связи с аппаратными средствами разработки встраиваемых вычислительных систем.

Информационные биты передаются последовательно друг за другом, отсюда и название интерфейса. Последовательная передача данных может осуществляться в синхронном и асинхронном режиме. При асинхронной передаче каждому байту данных предшествует старт-бит, за ним следуют биты данных, после них может передаваться бит паритета (четности) и, в завершение посылки, передается стоп-бит, гарантирующий определенную выдержку времени между соседними посылками (рис. 3.1). Старт-бит следующей посылки может передаваться в любой момент времени, начиная с момента окончания стоп-бита предыдущей посылки, таким образом, между передачами могут быть паузы произвольной (и неограниченной) длительности. Старт-бит позволяет организовать простую синхронизацию приемника по сигналу от передатчика. При этом подразумевается, что приемник и передатчик работают на одной и той же скорости обмена (т.е. имеют одну и ту же тактовую частоту), измеряемую количеством передаваемых бит в секунду. Внутренний генератор синхронизации приемника содержит счетчик-делитель частоты, обнуляемый в момент начала старт-бита от передатчика. Этот счетчик формирует внутренние стробы F_s , по которым приемник фиксирует принимаемые биты. В идеальном случае, эти внутренние стробы должны приходиться на середину битовых интервалов, однако, из-за разностей скоростей приемника и передатчика возникает накапливающееся рассогласование. Для безошибочного приема 8 бит данных, 1 бита паритета и стоп-бита предельное рассогласование не может превышать 5% на бит (в идеале, строб приходится на середину бита, при отклонении 50% мы примем бит с ошибкой, $50\% / 10 \text{ бит} = 5\%$). Очевидно, что чем больше скорость передачи, тем строже требования к синхронизации приемника и передатчика, кроме того, с ростом скорости обмена усиливается влияние фронтов импульсов.

Для асинхронного режима принят ряд стандартных скоростей обмена: 50, 75, 110, 150, 300, 600, 1200, 2400,

4800, 9600, 19200, 38400, 57600 и 115200 бит/сек.

Количество бит данных задается программно и может составлять 5, 6, 7 и 8 бит. Количество стоп-бит может быть 1, 1.5, 2 (по длине). Бит паритета может отсутствовать.

Таким образом, для установления связи при асинхронном режиме необходимо установить одинаковыми следующие параметры приемника и передатчика: скорость обмена, количество бит данных, наличие бита паритета, тип паритета (четность - нечетность), длину стоп-бита.

Асинхронный обмен в ПК реализуется с помощью COM порта с использованием протокола RS-232C.

Синхронный режим передачи данных предполагает постоянную активность канала связи. Посылка начинается с передачи синхробайта, за которым вплотную следует последовательность передаваемых бит. Если у передатчика нет данных для передачи, то он заполняет паузу непрерывной передачей старт-бит. Очевидно, что при синхронном обмене можно добиться больших скоростей обмена, т.к. не требуется передавать старт-бит, стоп-бит, бит паритета (в асинхронном на каждые 8 бит данных приходится не 3-4 служебных бита). При синхронном обмене необходима внешняя синхронизация, так как передается большой массив данных. При синхронизации только в начале передачи блока данных даже небольшая разность частот приведет к появлению быстро накапливающийся ошибке. Внешняя синхронизация может обеспечиваться как за счет использования отдельной линии для передачи сигнала синхронизации, так и за счет применения самосинхронизирующихся методов кодирования данных (манчестерский код, NRZ код), при использовании которых синхросигнал выделяется приемником из потока принимаемых данных.

Старт бит и стоп бит

Сигнальная линия может находиться в двух состояниях: включена и выключена. Линия в состоянии ожидания всегда включена. Когда устройство или компьютер хотят передать данные, они переводят линию в



состояние выключено - это установка Старт бита. Биты сразу после Старт бита являются битами данных. Стоп бит позволяет устройству или компьютеру произвести синхронизацию при возникновении сбоев.

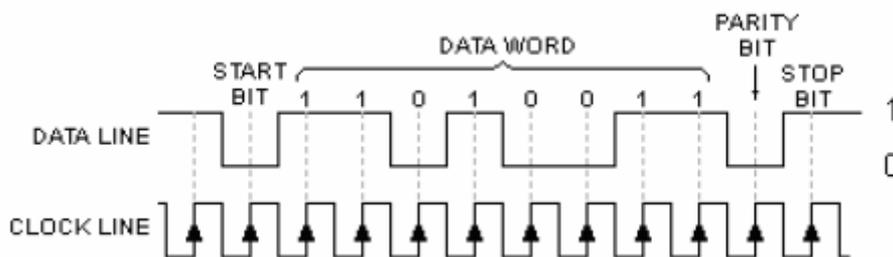
Например, помеха на линии скрыла Старт бит. Период между старт и стоп битами постоянен, согласно значению скорости обмена, числу бит данных и бита четности. Стоп бит всегда включен. Если приемник определяет выключенное состояние, когда должен присутствовать стоп бит, фиксируется появление ошибки. Стоп бит не просто один бит минимального интервала времени в конце каждой передачи данных. На компьютерах обычно он эквивалентен 1 или 2 битам, и это должно учитываться программой драйвера. Хотя, 1 стоп бит наиболее общий, выбор 2 бит в худшем случае немного замедлит передачу сообщения. (Есть возможность установки значения стоп бита равным 1.5. Это используется при передаче менее 7 битов данных. В этом случае не могут быть переданы символы ASCII, и поэтому значение 1.5 используется редко.)

Контроль четности

При передаче по последовательному каналу контроль четности может быть использован для обнаружения ошибок при передаче данных. При использовании контроля четности посылаются сообщения подсчитывающие число единиц в группе бит данных. В зависимости от результата устанавливается бит четности. Приемное устройство также подсчитывает число единиц и затем сверяет бит четности. Для обеспечения контроля четности компьютер и устройство должны одинаково производить подсчет бита четности. То есть, определиться устанавливать бит при четном (even) или нечетном (odd) числе единиц. При контроле на четность биты данных и бит четности всегда должны содержать четное число единиц. В противоположном случае соответствует для контроля на нечетность. Часто в драйверах доступны еще две опции на четность: Mark и Space. Эти опции не влияют на возможность контроля ошибок. Mark означает, что устройство всегда устанавливает бит четности в 1, а Space - всегда в 0.

Пример

В этом примере показана структура передаваемых данных со синхронизирующим тактовым сигналом. В этом примере используется 8 бит данных, бит четности и стоп бит. Такая структура также обозначается 8E1 (8 data, parity –even, 1-stop).



Примечание: Тактовый сигнал - для асинхронной передачи это внутренний сигнал

Управление потоком

Управление потоком представляет управлять передаваемыми данными. Иногда устройство не может обработать принимаемые данные от компьютера или другого устройства. Устройство использует управление потоком для прекращения передачи данных. Могут использоваться аппаратное или программное управление потоком. Аппаратный протокол управления потоком RTS/CTS. Он использует дополнительно два провода в кабеле, а не передачу специальных символов по линиям данных. Поэтому аппаратное управление потоком не замедляет обмен в отличие от протокола Хоп-Хoff. При необходимости послать данные компьютер устанавливает сигнал на линии RTS. Если приемник (модем) готов к приему данных, то он отвечает установкой сигнала на линии CTS, и компьютер начинает посылку данных. При неготовности устройства к приему сигнал CTS не устанавливается. Программный протокол управления потоком Хоп/Хoff использует два символа: Хоп и Хoff. Код ASCII символа Хоп - 17, а ASCII код Хoff - 19. Модем имеет маленький буфер, поэтому при его заполнении модем посылает символ Хoff компьютеру для прекращения посылки данных. При появлении возможности приема данных посылается символ Хоп и компьютер продолжит пересылку данных. Этот тип управления имеет преимущество в том, что не требует дополнительных линий, т.к. символы передаются по линиям TD/RD. Но на медленных соединениях это может привести к значительному замедлению соединения, т.к. каждый символ требует 10 битов.

Физические реализации последовательного интерфейса

Существует несколько родственных последовательных интерфейсов:

RS-232C, (COM порт ПК)

C2 (Российский (СССР) стандарт, читается «Стык два», почти равен RS232, но отсутствует часть служебных сигналов

ИРПС интерфейс радиальный последовательный (почти =C2)

RS-423A, RS-422A, RS-485A.

CL-20, CL-40 (токовая петля)

IRDA инфракрасный порт

Характеристики этих интерфейсов сведены в таблицу 3.1.

Таблица 3.1.



RS-232C	<p>дуплекс</p>	L=15 м, V=20 Кбит/с
RS-423A	<p>дуплекс</p>	L=9 м, V=100 Кбит/с L=91 м, V=10 Кбит/с L=1200 м, V=1 Кбит/с
RS-422A	<p>дуплекс</p>	L=12 м, V=10 Мбит/с L=120 м, V=1 Мбит/с L=1200 м, V=100 Кбит/с
RS-485A	<p>полудуплекс, до 32 параллельно соединенных приемопередатчиков</p>	L=12 м, V=10 Мбит/с L=120 м, V=1 Мбит/с L=1200 м, V=100 Кбит/с

Линии интерфейсов RS-232C и RS-423A несимметричны, имеют самую низкую защищенность от синфазной помехи. RS-423A имеет приемник с дифференциальным входом, что несколько повышает его помехозащищенность.

Лучшими параметрами обладает симметричный дуплексный интерфейс RS-422A и его полудуплексный магистральный аналог RS-485A. Приемник и передатчик RS-422A и RS-485A имеют дифференциальные входы и следовательно обладают высокой защищенностью от синфазных помех. Интерфейс RS-485A обладает передатчиком повышенной мощности с защитой от короткого замыкания линии, защитой от перегрева при длительной перегрузке и защитой от коллизий (одновременной работы нескольких параллельно включенных передатчиков). Одним из вариантов последовательного интерфейса является интерфейс типа "токовая петля". В этом интерфейсе сигналом является не уровень напряжения, а ток в двухпроводной линии. Обычно, за единицу принимают ток 20 мА, за ноль - отсутствие тока. В таком варианте интерфейса приемник может распознавать обрыв линии – при обрыве принимаются одни нули и обрыв распознается по отсутствию стоп-битов. Обычно, "токовая петля" предполагает наличие гальванической развязки приемника и передатчика, как правило, выполняемой при помощи оптронов.

Питание токовой петли может осуществляться от передатчика (вариант с активным передатчиком) или от приемника (активный приемник). Токовая петля с гальванической развязкой позволяет передавать данные на расстояние до нескольких километров, определяемое уровнем помех и сопротивлением пары проводов. Поскольку интерфейс требует пары проводов для каждого сигнала, то обычно применяют две пары - принимаемые данные и передаваемые данные, а управление потоком ведется по протоколу XON/XOFF. Одним из классических примеров интерфейса "токовая петля" является интерфейс MIDI, применяемый в звуковых картах. Все перечисленные выше варианты реализации интерфейса являются проводными, однако, существуют и беспроводные варианты, наибольшее распространение среди которых получил инфракрасный (ИК) интерфейс. Большинство ИК интерфейсов работают на расстоянии от одного до нескольких метров на низкой скорости (до 115,2 кбит/с), средней (до 1.152 Мбит/с) или высокой (до 4 Мбит/с) скоростью.

Уровни сигналов данных RS232C

Уровень	Передатчик	Приемник
Логический 0	От +5 В до +15 В	От +3 В до +25 В
Логический 1	от -5 В до -15 В	От -3 В до -25 В
Не определен	От -3 В до +3 В	

Уровни управляющих сигналов



Сигнал	На выходе устр-ва (Driver)	На входе устр-ва (Terminator)
"Off"	От -5 В до -15 В	от -3 В до -25 В
"On"	От 5 В до 15 В	от 3 В до 25 В

Перед соединением двух компьютеров через RS-232, каждый из которых питается от различных источников

рекомендуется выравнять напряжения между их сигнальными землями перед подключением.

Контакты разъемов

Контакт	Обозн.	Направление	Описание
1	SHIELD	---	Shield Ground - защитная земля, соединяется с корпусом устройства и экраном кабеля
2	TXD	-->	Transmit Data - Выход передатчика
3	RXD	<--	Receive Data - Вход приемника
4	RTS	-->	Request to Send - выход запроса передачи данных
5	CTS	<--	Clear to Send - вход разрешения терминалу передавать данные
6	DSR	<--	Data Set Ready - вход сигнала готовности от аппаратуры передачи данных
7	GND	---	System Ground - сигнальная (схемная) земля
8	CD	<--	Carrier Detect - вход сигнала обнаружения несущей удаленного модема
9-19	N/C	-	-
20	DTR	-->	Data Terminal Ready - выход сигнала готовности терминала к обмену данными
21	N/C	-	-
22	RI	<--	Ring Indicator - вход индикатора вызова (звонка)
23-25	N/C	-	-

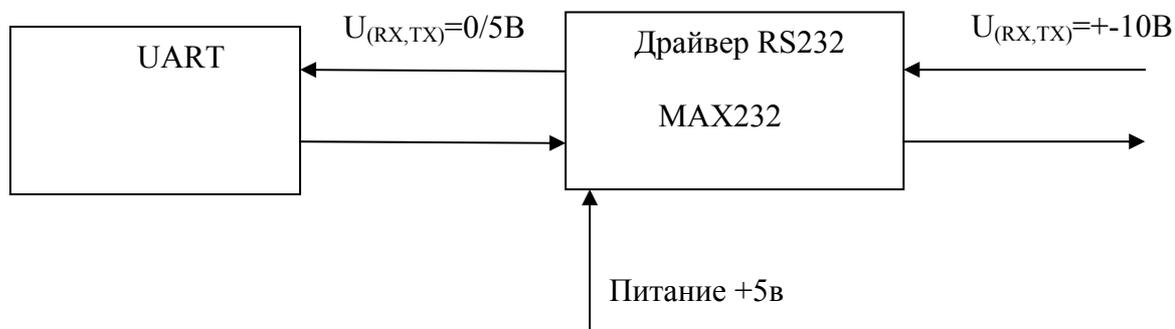
Контакт	Обозн.	Направление	Описание
1	CD	<--	Carrier Detect
2	RXD	<--	Receive Data
3	TXD	-->	Transmit Data
4	DTR	-->	Data Terminal Ready
5	GND	---	System Ground
6	DSR	<--	Data Set Ready
7	RTS	-->	Request to Send
8	CTS	<--	Clear to Send
9	RI	<--	Ring Indicator

4 Последовательные периферийные адаптеры

Вся гамма физических последовательных интерфейсов может быть реализована на одной и той же микросхеме UART (Universal Asynchronous Resiver/Transmitter)



Для реализации конкретного физического последовательного интерфейса (т.е. для преобразования логических сигналов Tx и Rx в сигналы и линии конкретного интерфейса) применяются микросхемы драйверов. Например для организации +12 вольтового интерфейса RS232 можно применить микросхему драйвера MAX232. В этой микросхеме имеется встроенный преобразователь напряжения питания +5В в +10В и минус 10В.



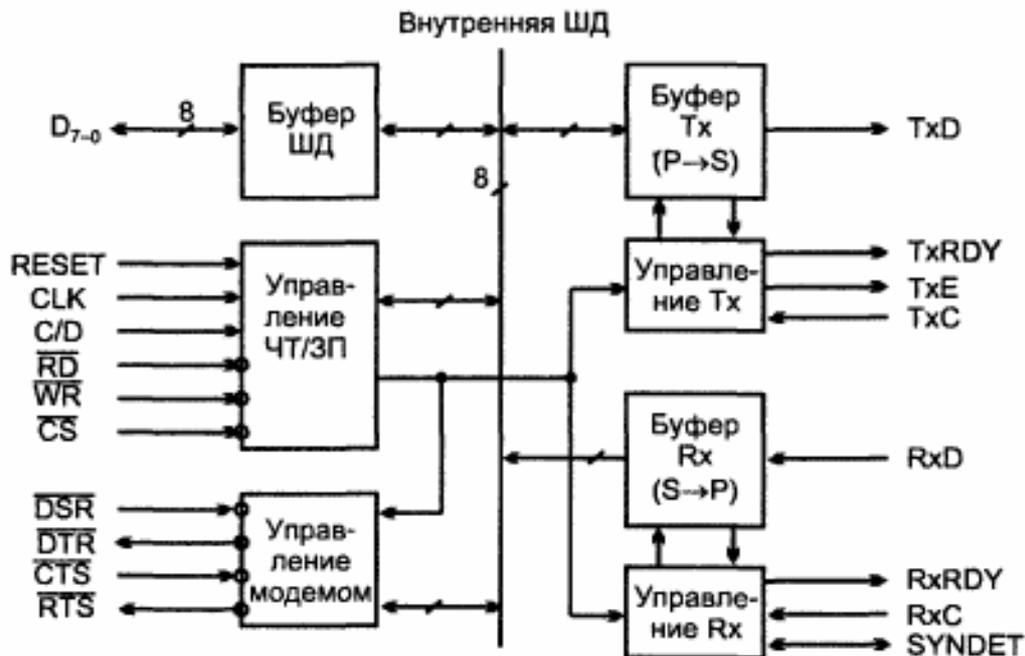


Рис. 6.10. Структура программируемого связного адаптера

Буфер ШД — двунаправленный, восьмиразрядный, с тремя состояниями. Он связывает адаптер с системной шиной данных и принимает данные по командам OUT port, выдает — по командам IN port. Через буфер передаются также управляющие и командные слова и слово состояния адаптера. В буфере имеются регистры данных (входной и выходной), команд и состояния.

Блок управления чтением/записью принимает сигналы от системной шины данных и генерирует сигналы управления работой всех блоков адаптера.

Выводы и сигналы ПСА

Выводы и сигналы ПСА имеют следующее назначение:

- RESET — установка адаптера в исходное состояние, после него адаптер находится в бездействии до записи нового набора управляющих слов для определения задаваемых ему функций. В состояние бездействия адаптер вводится также программой по команде сброса;
- CLK — вход тактовой частоты для внутреннего тактирования процессора. Внешние входы и выходы адаптера не привязаны к тактам сигнала CLK, но частота этого сигнала должна быть выше битовой частоты передачи данных не менее чем в 30 раз;
- \overline{RD} , \overline{WR} и \overline{CS} — сигналы, смысл которых уже известен (стробы чтения и записи и сигнал выбора микросхемы);
- C/D (Control/Data) — указывает на тип передаваемой информации, при единичном значении этого сигнала вводятся управляющие слова или выводится слово состояния адаптера, при нулевом — передаются данные. Вме-



сте с сигналами \overline{RD} и \overline{WR} определяет характер передачи. Обычно на этот вход подключается младший разряд адреса A_0 . Направления передачи и характер информации задаются для адаптера таблицей (табл. 6.3).

Таблица 6.3

C/D	\overline{RD}	\overline{WR}	\overline{CS}	Операция
0	0	1	0	ШД ← данные адаптера
0	1	0	0	Данные адаптера ← ШД
1	0	1	0	ШД ← слово состояния
1	1	0	0	Управляющее слово ← ШД
X	1	1	0	Отключено
X	X	X	1	Отключено

Адаптер имеет набор управляющих входных и выходных сигналов для управления модемом. Модем указан здесь как наиболее типичное устройство, работающее во взаимодействии с ПСА, хотя, в сущности, это сигналы общего назначения, которые могут быть использованы и для управления другими устройствами. Для управления модемом (терминалом) имеются две пары сигналов квитирования;

- \overline{DSR} (Data Set Ready) — запрос готовности передатчика терминала, сигнал связан с одноразрядным портом и может быть проверен процессором чтением слова состояния. Низкий уровень этого сигнала говорит о том, что модем (терминал) имеет информацию для передачи процессору;
- \overline{DTR} (Data Terminal Ready) — этот сигнал является реакцией на запрос \overline{DSR} . Активизируется соответствующим битом командного слова, если процессором разрешен обмен с модемом. Связан с разрешением модему посылки данных на вход приемника адаптера;
- \overline{RTS} (Request to Send) — сигнал связан с одноразрядным выходным портом. Является запросом от адаптера готовности приемника терминала принять данные. Задается программированием соответствующего бита в командном слове, когда процессором разрешен обмен с модемом;
- \overline{CTS} (Clear to Send) — сигнал готовности приемника терминала принять данные. Низкий уровень этого сигнала разрешает адаптеру передачу последовательных данных, если установлен бит $TxEN$ в командном слове. При снятии $TxEN$ или \overline{CTS} во время работы передатчика он будет передавать все данные, записанные до запрещения передачи, прежде чем остановится.



Передатчик ПСА

Буфер передатчика адаптера (буфер Tx) принимает параллельные данные от буфера ШД, преобразует их в поток последовательных битов, вводит в этот поток служебные символы или биты и выдает составленный необходимым образом поток битов на вывод TxD по отрицательным фронтам импульсов частоты TxС. Передача начинается после ее разрешения и при условии $\overline{CTS} = 0$. Вывод TxD принимает высокий уровень напряжения после сброса, запретов по условиям TxEN или \overline{CTS} либо при условии "передатчик пуст", связанном с сигналом TxЕ (TxEmpty).

Схема управления передатчиком (управление Tx) вырабатывает следующие внутренние и внешние сигналы для процессов передачи последовательных данных:

- TxRDY — этот выходной сигнал указывает процессору на готовность передатчика адаптера принять символ данных. Сигнал может проверяться чтением слова состояния или использоваться как запрос прерывания (он может маскироваться битом TxEN командного слова). Автоматически сбрасывается передним фронтом stroba записи \overline{WR} , когда символ данных загружается из процессора;
- TxЕ — сигнал устанавливается, когда адаптер не имеет символа для передачи (входной буфер в блоке "буфер ШД" пуст, и после выхода символа из сдвигающего регистра передатчика этот регистр будет нечем загрузить). Сбрасывается после получения символа от процессора, если передача разрешена, и остается высоким, если передача запрещена соответствующим битом командного слова. Сигнал может быть использован для индикации конца режима передачи и оповещения процессора о моменте переключения линии передачи на другое направление в полудуплексном режиме работы. В синхронном режиме высокий уровень сигнала показывает, что символ не был загружен и в поток данных следует вводить синхросимволы. Пока передаются синхросимволы, высокий уровень сигнала сохраняется;
- TxС и RxС — сигналы синхронизации передатчика и приемника, задающие скорость следования последовательных битов. При синхронных передачах базовая скорость равна частоте TxС (RxС), при асинхронных она является частью частоты TxС (RxС) (это 1, или 1/16 или 1/64 от TxС или RxС). Очень часто частоты TxС и RxС идентичны. Их синхронности с сигналом CLK не требуется.

Приемник ПСА

Буфер приемника принимает последовательные данные, преобразует их в параллельные, проверяет биты или символы, специфичные для посылок данного типа и посылает принятый символ в процессор. Вывод RxD служит входом последовательных данных.



Блок управления приемником Rx обеспечивает управление всеми действиями, связанными с приемом информации. Схемы этого блока предотвращают восприятие неиспользуемой линии данных как L-активной в режиме паузы. Для начала приема требуется появление высокого уровня (марки) на входе RxD после сброса системы. Если это выполняется, то разрешается поиск отрицательного фронта входного сигнала (старт-бита). Истинность старт-бита устанавливается проверкой уровня сигнала в его середине. Ошибки работы адаптера устанавливают соответствующие биты в слове состояния (четности, формата или переполнения, если новая информация замещает старую раньше, чем она была использована).

RxRDY — выходной сигнал, показывающий, что адаптер имеет символ, готовый к выводу в процессор. Может проверяться чтением слова состояния или использоваться как запрос прерывания для процессора. Если команда разрешения приема RxEN отсутствует, то сигнал RxRDY находится в состоянии сброса. Отсутствие чтения принятого символа из выходного регистра адаптера до появления следующего ведет к загрузке нового символа и потере старого. Устанавливается ошибка переполнения.

SYNDET (SYNC Detect/Break Detect) этот вывод в синхронном режиме используется как SYNDET и может быть входом или выходом в зависимости от программирования адаптера. При внутренней синхронизации является выходом и устанавливается как признак выявления синхросимвола в режиме приема. Если запрограммированы два синхросимвола, SYNDET установится в середине последнего бита второго синхросимвола. Сигнал автоматически сбрасывается после операции чтения состояния. Когда используется как входной (режим внешней синхронизации), его появление заставляет адаптер начать прием данных. В асинхронном режиме вывод используется для сигнала Break Detect, который устанавливается при низком уровне на интервалах стоп-битов в двух последовательных посылках. Сигнал может быть выявлен чтением слова состояния. Сбрасывается при сбросе адаптера или возвращении входного сигнала к нормальному состоянию (появлению единиц на интервалах стоп-битов)

Программирование адаптера

Адаптер программируется загрузкой в него по команде OUT port *начального и текущего управляющих слов*. Начальное управляющее слово (инструкция режима MI — Mode Instruction) вводится после сброса и задает режим работы адаптера, формат передаваемых символов, скорость передачи/приема, характеристику контроля, тип синхронизации. Формат MI показан на рис. 6.11. Как видно, трактовка разрядов D₇ и D₆ слова MI зависит от режима.

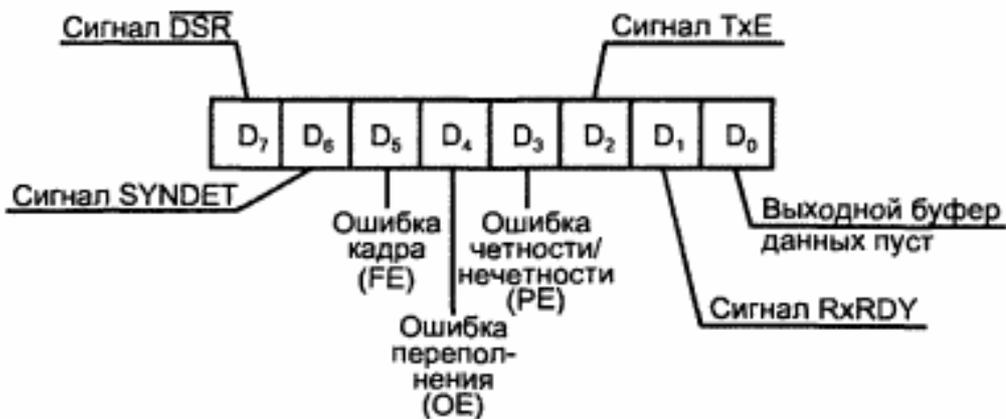
При синхронном обмене и внутренней синхронизации после инструкции режима MI в адаптер вводятся один или два синхросимвола, для хранения которых в схеме управления приемником имеются два специальных регистра



Рис. 6.11. Формат управляющего слова режима программируемого связного адаптера



а



б

Рис. 6.12. Форматы командного слова (а) и слова состояния адаптера (б)



Fibre Channel Protocol (FCP) — [транспортный протокол](#) (как [TCP](#) в [IP](#)-сетях), который, как правило, доставляет команды [SCSI](#) по сетям Fibre Channel. Может использоваться как несущий и для других протоколов — например, [ATM](#), [IP](#), [HIPPI](#) и других.

Поддерживается как оптическая, так и электрическая среда ([витая пара](#), [коаксиальный](#) или [твинаксиальный](#) кабели, а также многомодовое или одномодовое [волокно](#)), со скоростью передачи [данных](#) от 133 мегабит/с до 10 гигабит/с на расстояния до 50 [километров](#)

История Fibre Channel началась в 1985 году, а в 1994 году был утверждён ANSI как стандарт, упрощавший интерфейс [HIPPI](#), для которого применялся массивный 50-парный кабель с громоздкими коннекторами. Первоначально интерфейс Fibre Channel должен был повысить дальность и упростить подключение линий передачи, а не увеличить скорость.

Версии Fibre Channel

Название	Пропускная способность (Gbps)	Производительность (MBps)
1GFC	1.0625	100
2GFC	2.125	200
4GFC	4.25	400
8GFC	8.5	800
10GFC Последовательный	10.51875	1000
20GFC	10.52	2000
10GFC Параллельный	12.75	



Рисунок ?? -Коннекторы **Fibre channel** - LC слева и SC справа

Таблица ?? Физические типы портов

Тип среды	Скорость (Mbyte/s)	Передачик	Модификация	Расстояние
<u>Одномодовое ВОЛОКНО</u>	400	1300 нм Длинноволновой лазер	400-SM-LL-I	2 м - 2 км
	100	1550 нм Длинноволновой лазер	100-SM-LL-V	2 м - >50 км
		1300 нм Длинноволновой лазер	100-SM-LL-I	2 м - 2 км
	200	1550 нм Длинноволновой лазер	200-SM-LL-V	2 м - >50 км
		1300 нм Длинноволновой лазер	200-SM-LL-L	2 м - 10 км
		1300 нм Длинноволновой лазер	200-SM-LL-I	2 м - 2 км
<u>Многомодово</u>	100	850 nm Коротковолновой	100-M5-SM-I	0.5 м - 150



<u>е волокно (50μм)</u>				М
	200	лазер	200-M5-SN-I	М 0.5 м - 300
	100		100-M6-SN-I	М 0.5 м - 300
			100-M6-SL-I	2 м - 175 м

5 SCSI

SCSI ([англ. Small Computer System Interface](#), произносится *эсцэсси* или *эссиэсэй*, в разговорной речи часто называют *скази*) — [интерфейс](#), разработанный для объединения на одной шине различных по своему назначению устройств, таких как [жёсткие диски](#), накопители на магнитооптических дисках, приводы [CD](#), [DVD](#), [стримеры](#), [сканеры](#), [принтеры](#) и т. д. Раньше имел неофициальное название Shugart Computer Systems Interface в честь создателя [Алана Ф. Шугарта](#)

Теоретически возможен выпуск устройства любого типа на шине SCSI.

После стандартизации в [1986](#) году SCSI начал широко применяться в компьютерах [Apple Macintosh](#) [источник не указан 22 дня], [Sun Microsystems](#). В компьютерах, совместимых с [IBM PC](#), SCSI не пользуется такой популярностью в связи со своей сложностью и сравнительно высокой стоимостью и применяется преимущественно в серверах.

SCSI широко применяется на серверах, высокопроизводительных рабочих станциях; [RAID](#)-массивы на серверах часто строятся на жёстких дисках со SCSI-интерфейсом (однако, в серверах нижнего ценового диапазона всё чаще применяются RAID-массивы на основе [SATA](#)). В настоящее время устройства на шине [SAS](#) постепенно вытесняют устаревшую шину SCSI.

Система команд SCSI на уровне программного обеспечения употребляется в единых стеках поддержки устройств хранения данных в ряде операционных систем, таких, как [Microsoft Windows](#).

Существует реализация системы команд SCSI поверх оборудования (контроллеров и кабелей) IDE/ATA/SATA, называемая [ATAPI](#) - ATA Packet Interface. Все используемые в компьютерной технике подключаемые по IDE/ATA/SATA приводы CD/DVD/Blu-Ray используют эту технологию.

5.1 Стандарты

Существует три стандарта SCSI:

SE ([англ. single-ended](#)) - ассиметричный SCSI, для передачи каждого сигнала используется отдельный проводник.

LVD ([англ. low-voltage-differential](#)) — интерфейс дифференциальной шины низкого напряжения, сигналы положительной и отрицательной полярности идут по разным физическим



проводам - витой паре. На один сигнал приходится по одной витой паре проводников. Используемое напряжение при передаче сигналов $\pm 1,8$ В.

HVD ([англ. high-voltage-differential](#)) — интерфейс дифференциальной шины высокого напряжения, отличается от LVD повышенным напряжением и специальными приемопередатчиками.

Первый стандарт SCSI имеет 50-контактный неэкранированный разъем для внутрисистемных соединений и аналогичный экранированный разъем типа Centronics (Alternative 2) для внешних подключений. Передача сигналов осуществляется 50 контактным кабелем типа - A-50 на 8 разрядной (битной) шине.

В стандарте SCSI-2 для 8 битной шины предусматривался кабель типа А, который как и в SCSI-1 поддерживал 50-контактными разъемами типа D с уменьшенным шагом выводов (Alternative 1). Разъемы типа Centronics (Alternative 2) в SCSI-2 построены на 8 и 16 битной шине. Передача информации осуществляется по 68-контактным кабелям типа - А-68 и P-68(Wide). Для 32 битной версии шины был предусмотрен тип кабеля В, который должен был параллельно подключаться одновременно с кабелем А в одно устройство. Однако кабель В не получил широкого признания и из стандарта SCSI-3 исключен.

В стандарте SCSI-3 кабеля А-68 и P-68 поддерживались экранированными, либо неэкранированными разъемами типа D. Кабеля в SCSI-3 снабжены фиксаторами-защелками, а не проволочными кольцами, как разъемы Centronics. Начиная с этой версии SCSI в массивах накопителей используется 80-контактный разъем, называемый Alternative 4. Накопители с таким разъемом поддерживают "горячее" подключение устройств, т.е. устройства SCSI можно подключать и отключать при включенном питании.

Основные реализации SCSI (в хронологическом порядке):

Обзор интерфейсов SCSI

Наименование	Разрядность шины	Частота шины	Пропускная способность	Максимальная длина кабеля	Максимальное количество устройств
SCSI	8 бит	5 МГц	5 МБит/сек	6 м	8
Fast SCSI	8 бит	10 МГц	10 МБит/сек	1,5-3 м	8
Wide SCSI	16 бит	10 МГц	20 МБит/сек	1,5-3 м	16
Ultra SCSI	8 бит	20 МГц	20 МБит/сек	1,5-3 м	5-8
Ultra Wide SCSI	16 бит	20 МГц	40 МБит/сек	1,5-3 м	5-8
Ultra2 SCSI	8 бит	40 МГц	40 МБит/сек	12 м	8
Ultra2 Wide SCSI	16 бит	40 МГц	80 МБит/сек	12 м	16
Ultra3	16 бит	40 МГц	160 МБит/сек	12 м	16



SCSI		МГц DDR	МБит/сек		
Ultra-320 SCSI	16 бит	80 МГц DDR	320 МБит/сек	12 м	16

5.1.1 SCSI-1

Стандартизован [ANSI](#) в 1986 г.

Использовалась восьмибитная шина, с пропускной способностью в 3,5 МБайт/сек в асинхронном режиме и 5 МБайт/сек в синхронном режиме. Максимальная длина кабеля — до 6 метров.

5.1.2 SCSI-2

Этот стандарт был предложен в [1989](#) году и существовал в двух вариантах — Fast SCSI и Wide SCSI.

Fast SCSI характеризуется удвоенной пропускной способностью (до 10 МБайт/сек).

Wide SCSI в дополнение к этому имеет удвоенную разрядность шины (16 бит), что позволяет достичь скорости передачи до 20 МБ/сек.

При этом максимальная длина кабеля ограничивалась тремя метрами.

Также в этом стандарте была предусмотрена 32-х битная версия Wide SCSI, которая позволяла использовать два шестнадцатибитных кабеля на одной шине, но эта версия не получила распространения!

5.1.3 SCSI-3

Также известен под названием Ultra SCSI.

Предложен в [1992](#) году.

Пропускная способность шины составила 20 МБайт/сек для восьмибитной шины и 40 МБайт/сек — для шестнадцатибитной. Максимальная длина кабеля так и осталась равной трём метрам.

Устройства, отвечающие этому стандарту, известны своей чувствительностью к качеству элементов системы (кабель, терминаторы).

5.1.4 Ultra-2 SCSI

Предложен в [1997](#) году.

Использует [LVDS](#). Максимальная длина кабеля — 12 метров, пропускная способность — до 80 МБайт/сек..

5.1.5 Ultra-3 SCSI

Также известен под названием Ultra-160 SCSI.



Предложен в конце [1999](#) года.

Имеет удвоенную пропускную способность (по сравнению с Ultra-2 SCSI), которая составила 160 МБайт/сек. Увеличения пропускной способности удалось достичь за счёт одновременного использования фронтов и срезов импульсов.

В этот стандарт было добавлено использование [CRC](#) (Cyclic Redundancy Check), предупреждение ошибок.

5.1.6 Ultra-320 SCSI

Развитие стандарта Ultra-3 с удвоенной скоростью передачи данных (до 320 МБайт/сек).

5.1.7] Ultra-640 SCSI

Также известен под названием Fast Ultra-320.

Предложен в начале 2003 года.

Удвоенная пропускная способность (640 МБайт/сек). В связи с резким сокращением максимальной длины кабеля неудобен для использования с более чем двумя устройствами, поэтому не получил широкого распространения.

5.1.8 Протокол команд SCSI

В терминологии SCSI взаимодействие идёт между [инициатором](#) и целевым устройством. Инициатор посылает команду целевому устройству, которое затем отправляет ответ инициатору.

Команды SCSI посылаются в виде блоков описания команды ([англ.](#) *Command Descriptor Block, CDB*). Длина каждого блока может составлять 6, 10, 12 или 16 байт. В последних версиях SCSI блок может иметь переменную длину. Блок состоит из однобайтового кода команды и параметров команды.

После получения команды целевое устройство возвращает значение *00h* в случае успешного получения, *02h* в случае ошибки или *08h* в случае, если устройство занято. В случае, если устройство вернуло ошибку, инициатор обычно посылает команду запроса состояния. Устройство возвращает Key Code Qualifier (KCQ).

Все команды SCSI делятся на четыре категории: N (non-data), W (запись данных от инициатора целевым устройством), R (чтение данных) и В (двусторонний обмен данными). Всего существует порядка 60 различных команд SCSI, из которых наиболее часто используются:

- Test unit ready — проверка готовности устройства, в т.ч. наличия диска в дисковом.
- Inquiry — запрос основных характеристик устройства.
- Send diagnostic — указание устройству провести самодиагностику и вернуть результат.
- Request sense — возвращает код ошибки предыдущей команды.
- Read capacity — возвращает ёмкость устройства.
- Format Unit
- Read (4 варианта) — чтение.
- Write (4 варианта) — запись.
- Write and verify — запись и проверка.
- Mode select — установка параметров устройства.
- Mode sense — возвращает текущие параметры устройства.



Каждое устройство на SCSI-шине имеет как минимум один номер логического устройства (LUN — [англ. Logical Unit Number](#)). В некоторых более сложных случаях одно физическое устройство может представляться набором LUN.

Семейство стандартов SCSI включает в себя ряд стандартов уровня аппаратуры, стандарты SAM и SPC, описывающие главнейшие команды и структуры типа развернутой информации об ошибке, и специфичных для класса устройств стандартов.

Одним из последних является [MMC](#) - Multimedia Command Set, полностью описывающий систему команд приводов CD/DVD/Blu-Ray, в том числе их разновидностей с возможностью записи. Некоторые приводы, например, производства [Asus](#) и [Pioneer](#), используют конкурирующий стандарт Mt. Fuji, отличающийся от MMC в некоторых нюансах.

5.1.9 Терминирование

Параллельные шины SCSI всегда должны [терминироваться](#) с обеих сторон для обеспечения нормального функционирования. Подавляющее большинство контроллеров и многие устройства имеют возможность *автотерминирования* — использования встроенного терминатора.

5.1.10 Дальнейшее развитие

- Для передачи команд протокола SCSI по IP-сетям используется сетевой протокол [iSCSI](#), утверждённый [IETF](#) как стандартный в [2003](#) году.
- Для замены параллельной шины предложена технология подключения устройств по последовательной шине [Serial Attached SCSI \(SAS\)](#).

3 Интерфейс USB

Введение

Шина USB (Universal Serial Bus - универсальная последовательная шина) появилась по компьютерным меркам довольно давно - версия первого утвержденного варианта стандарта появилась 15 января 1996 года. Разработка стандарта была инициирована весьма авторитетными фирмами - Intel, DEC, IBM, NEC, Northern Telecom и Compaq.

Основная цель стандарта, поставленная перед его разработчиками - создать реальную возможность пользователям работать в режиме Plug&Play с периферийными устройствами. Это означает, что должно быть предусмотрено подключение устройства к работающему компьютеру, автоматическое распознавание его немедленно после подключения и последующей установки соответствующих драйверов. Кроме этого, желательно питание маломощных устройств подавать с самой шины. Скорость шины должна быть достаточной для подавляющего большинства периферийных устройств. Попутно решается историческая проблема нехватки ресурсов на внутренних шинах IBM PC совместимого компьютера - контроллер USB занимает только одно прерывание независимо от количества подключенных к шине устройств.

Технические характеристики

Возможности USB (версия 1.1) следуют из ее технических характеристик:

- Две скорости обмена:
- Высокая скорость (full-speed signaling bit rate) - 12 Mb/s (максимальная длина кабеля - 5 м)
- Низкая скорость (low-speed signaling bit rate) - 1.5 Mb/s (максимальная длина кабеля - 3 м)
- Максимальное количество подключенных устройств (включая размножители) - 127
- Возможно подключение устройств с различными скоростями обмена
- Отсутствие необходимости в установке пользователем дополнительных элементов, таких как терминаторы для SCSI

- Возможность питания периферийных устройств от шины - 5 V

- Максимальный ток потребления на одно устройство - 500 mA. Питание непосредственно от USB

возможно только для устройств с малым потреблением, таких как клавиатуры, мыши, джойстики и т. п.

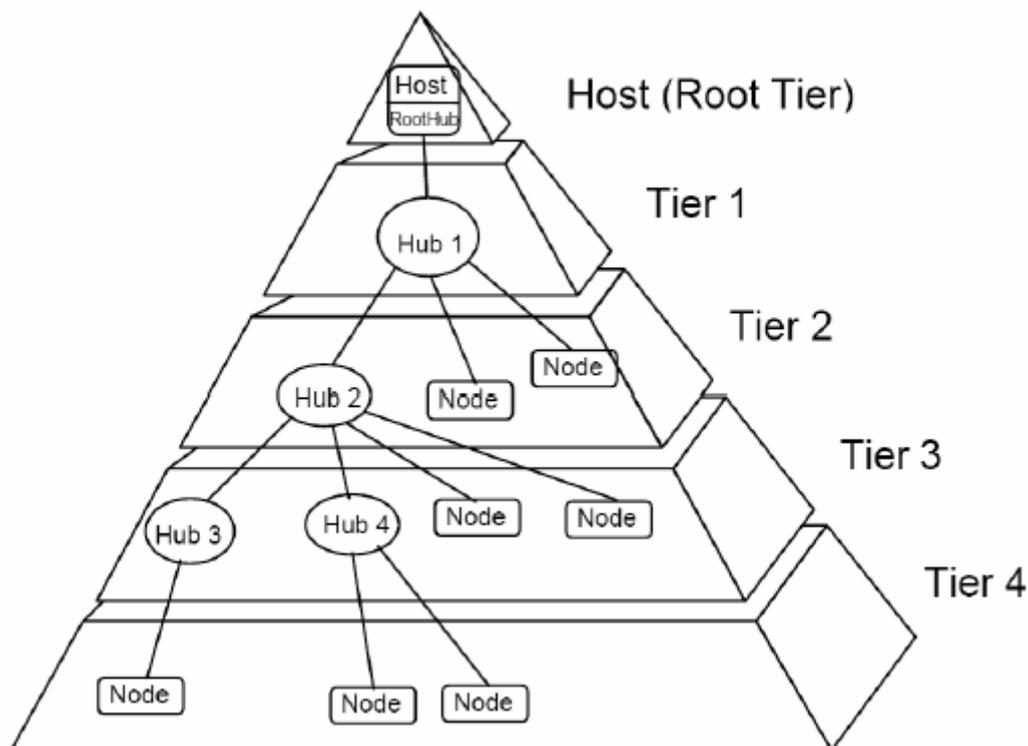
Особенно удобен этот интерфейс для подключения часто подключаемых/отключаемых приборов, таких как цифровые фотокамеры. Конструкция разъемов для USB рассчитана на многократное сочленение/расчленение. Возможность использования только двух скоростей обмена данными ограничивает



применяемость шины, но существенно уменьшает количество линий интерфейса и упрощает аппаратную реализацию. На шине USB допускаются два типа устройств:

- ведущее или хост (host)
- ведомое (slave, device)

При этом хост на шине всегда один. Топология шины — звезда с 6-уровневой древовидной структурой. Для увеличения количества подключаемых устройств используются разветвители (hub).



Модель передачи данных

Логическое устройство USB представляет собой набор независимых *конечных точек* (Endpoint, EP), с которыми хост-контроллер (и клиентское ПО) обменивается информацией. То есть каждое устройство может организовать до 15 независимых потоков передачи данных. Состав конечных точек зависит от устройства, но нулевая EP всегда используется для служебных целей, например, конфигурирование устройства. Каждому логическому устройству USB. Каждая конечная точка логического устройства идентифицируется своим номером (0-15) и направлением передачи (*IN* - передача к хосту, *OUT* - от хоста).

Архитектура USB допускает четыре базовых типа передач данных между хостом и периферийными устройствами:

Изохронные передачи (isochronous transfers) - потоковые передачи в реальном времени, занимающие предварительно согласованную часть пропускной способности шины с гарантированным временем задержки доставки. На полной скорости (FS) можно организовать один канал с полосой до 1,023 Мбайт/с (или два по 0,5 Мбайт/

с), заняв 70 % доступной полосы (остаток можно занять и менее емкими каналами). На высокой скорости (HS) можно получить канал до 24 Мбайт/с (192 Мбит/с). Надежность доставки не гарантируется - в случае обнаружения ошибки

изохронные данные не повторяются, недействительные пакеты игнорируются. Шина USB позволяет с помощью

изохронных передач организовывать *синхронные соединения* между устройствами и прикладными программами.

Изохронные передачи нужны для потоковых устройств: видеокамер, цифровых аудиоустройств (колонки USB, микрофон), устройств воспроизведения и записи аудио- и видеоданных (CD и DVD). Видеопоток (без компрессии) шина USB способна передавать только на высокой скорости.

• *Прерывания* (interrupts) - передачи спонтанных сообщений, которые должны выполняться с задержкой не большей, чем требует устройство. Предел времени обслуживания устанавливается в диапазоне 10-255 мс для низкой и 1-255 мс для полной скорости. На высокой скорости можно заказать и 125 мкс. Доставка гарантирована, при случайных ошибках обмена выполняется повтор (правда, при этом время обслуживания увеличивается). Прерывания используются, например, при вводе символов с клавиатуры или передаче сообщений о перемещениях мыши.



Прерываниями можно передавать данные и к устройству (как только устройство сигнализирует о потребности в данных, хост своевременно их передает). Размер сообщения может составлять 0-8 байт для низкой скорости, 0-64 байт

- для полной и 0-1024 байт - для высокой скорости передачи.

- **Передачи массивов данных** (bulk data transfers) - это передачи без каких-либо обязательств по своевременности доставки и по скорости. Передачи массивов могут занимать всю полосу пропускания шины, свободную от передач других типов. Приоритет этих передач самый низкий, они могут приостанавливаться при большой загрузке шины. Доставка гарантированная - при случайной ошибке выполняется повтор. Передачи массивов уместны для обмена данными с принтерами, сканерами, устройствами хранения и т. п.

- **Управляющие передачи** (control transfers) используются для конфигурирования устройств во время их подключения и для управления устройствами в процессе работы. Протокол обеспечивает гарантированную доставку данных и подтверждение устройством успешности выполнения управляющей команды. Управляющая передача позволяет подать устройству команду (запрос, возможно, с дополнительными данными) и получить на него ответ (подтверждение или отказ от выполнения запроса и, возможно, данные). Только управляющие передачи на USB обеспечивают *синхронизацию запросов и ответов*; в остальных типах передач явной синхронизации потока ввода с потоком вывода нет. Каждая единица клиентского ПО (обычно представляемая драйвером) связывается с одним интерфейсом своего устройства (функции) монополюно и независимо

Программная организация обмена

Ниже перечислены компоненты *программной части USB*:

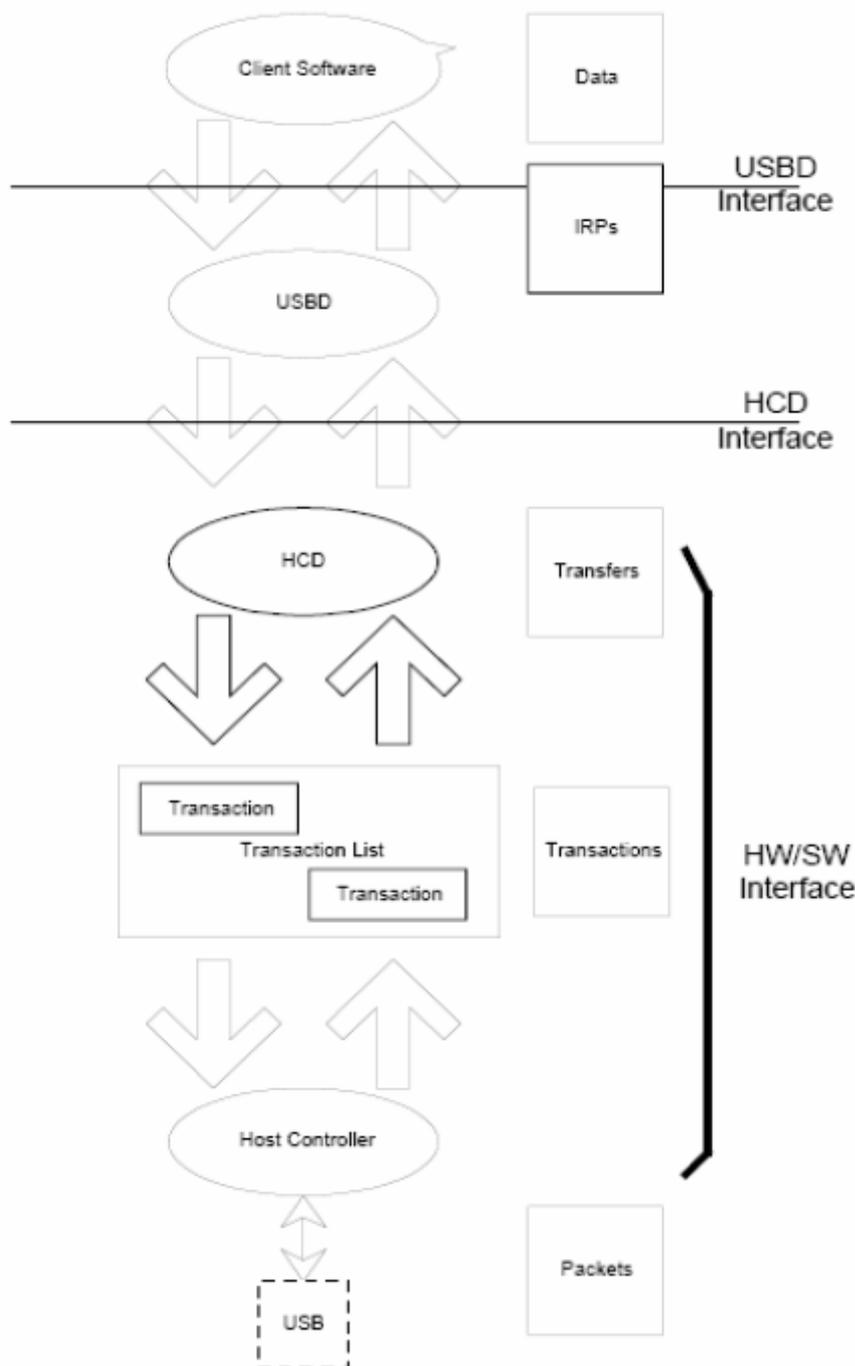
- **Клиентское ПО** (Client Software, CSw) - это драйверы устройств USB, обеспечивающие доступ к устройствам со стороны прикладного ПО. Драйверы взаимодействуют с устройствами только через программный интерфейс с общим драйвером USB (USB D). Непосредственного обращения к каким-либо регистрам аппаратных средств драйверы устройств USB не выполняют.

- **Драйвер USB** (USB Driver, USB D) «заведует» всеми устройствами USB системы, их нумерацией, конфигурированием, предоставлением служб, распределением пропускной способности шины, мощности питания и т. п.

- **Драйвер хост-контроллера** (Host Controller Driver, HCD) преобразует запросы ввода-вывода в структуры данных, размещенные в коммуникационной области оперативной памяти, и обращается к регистрам хост-контроллера. Хост-контроллер выполняет физические транзакции, используя эти структуры данных.

Для передачи или приема данных клиентское ПО посылает каналу пакет запроса ввода-вывода (Input/Output Request, IRP) и ждет уведомления о завершении его отработки. Формат IRP определяется реализацией драйвера USB D в конкретной ОС. В IRP имеются только сведения о запросе (местоположение буфера передаваемых данных в оперативной памяти и длина передачи); от свойств конкретного текущего подключения (скорость, допустимый размер

пакета) драйвер устройства абстрагируется. Обработкой запроса в виде транзакций на шине USB занимается драйвер USB D; он при необходимости длинные запросы разбивает на части (пакеты), пригодные для передачи за одну транзакцию. Транзакция на шине USB - это последовательность обмена пакетами между хостом и ПУ, в ходе которой может быть передан или принят один пакет данных (возможны транзакции, в которых данные не передаются). Обработка запроса считается завершенной в случае успешного выполнения всех связанных с ним транзакций. «Временные трудности», встречающиеся при их выполнении (неготовность к обмену данными), до сведения клиентского драйвера не доводятся - ему остается только ждать завершения обменов (или выхода по тайм-ауту). Однако устройство может сигнализировать о серьезных ошибках (ответом STALL), что приводит к аварийному завершению запроса; о последнем уведомляется клиентский драйвер. В этом случае отбрасываются и все последующие запросы к данному каналу. Возобновление работы с данным каналом возможно лишь после явного уведомления об обработке ошибочной ситуации, которое драйвер устройства делает с помощью специального запроса (тоже вызова USB D).



Длинные запросы разбиваются на транзакции так, чтобы размер пакета был максимальным. Последний пакет с остатком может оказаться короче максимального размера. Хост может считать короткий пакет либо разделителем, указывающим на конец блока данных, либо признаком ошибки, по которому канал останавливается. При передаче массивов использование укороченных пакетов в качестве разделителей наиболее естественно. Например, в одном из вариантов протоколов для устройств хранения данных укороченные пакеты известной длины применяются в качестве управляющих.

Каналы с каждой конечной точкой может быть связан один коммуникационный канал. Коммуникационные каналы USB разделяются на 2 типа:

- **Потоковый канал (streaming pipe)** доставляет данные от одного конца канала к другому, он всегда **однонаправленный**. Передачи данных в **разных** потоковых каналах друг с другом **не синхронизированы**. Это означает, что запросы клиентских драйверов для **разных каналов**, поставленные в определенном порядке относительно друг друга, могут выполняться другим порядком. Если во время исполнения какого-либо запроса происходит серьезная ошибка (об этом устройство сообщает ответом **STALL**), поток останавливается. Поток может реализовывать передачи массивов, изохронные передачи и прерывания.

- **Канал сообщений (message pipe)** является **двунаправленным**. Передачи сообщений во встречных направлениях **синхронизированы** друг с другом и строго **упорядочены**. На каждое сообщение противоположная сторона обязана ответить подтверждением его приема и обработки. Форматы сообщений определяются



спецификацией USB: имеется набор стандартных сообщений (запросов и ответов) и зарезервированных идентификаторов сообщений, формат которых определяется разработчиком устройства или интерфейса. С каналами связаны характеристики, соответствующие конечной точке (полоса пропускания, тип сервиса, размер пакета и т. п.). Каналы организуются при конфигурировании устройств USB. Полоса пропускания шины делится между всеми установленными каналами. Каналы различаются и по назначению:

- **Основной канал сообщений** (default pipe, он же control pipe 0), владельцем которого является USBД, используется для доступа к конфигурационной информации всех устройств. Этот канал устанавливается с нулевой конечной точкой (Endpoint Zero, EP0), которая у всех устройств всегда поддерживает только управляющие передачи.

- **Клиентские каналы** (client pipes) - каналы, владельцами которых являются драйверы устройств. По этим каналам могут передаваться как потоки, так и сообщения; они поддерживают любые типы передач USB (изохронные передачи, прерывания, массивы, управляющие передачи).

Интерфейс устройства, с которым работает клиентский драйвер, представляет собой **связку клиентских каналов** (pipe's bundle). Для этих каналов драйверы устройств являются единственными источниками и потребителями передаваемых данных.

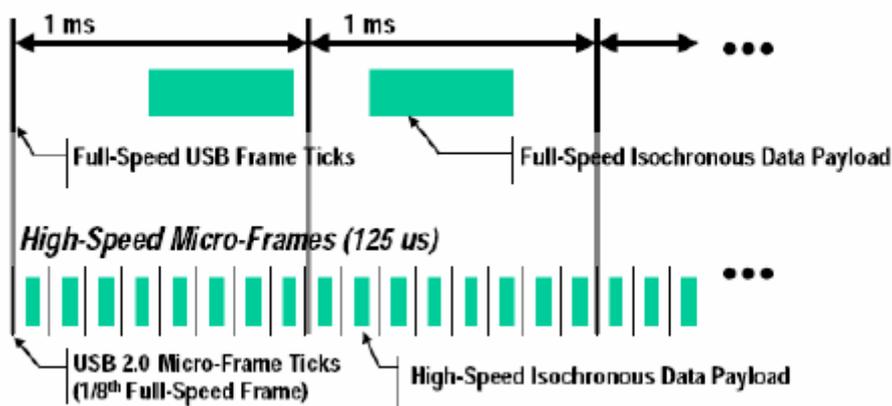
Владельцем основных каналов сообщений всех устройств является драйвер USB (USBД); по этим каналам передается информация конфигурирования, управления и состояния. Основным каналом сообщений может пользоваться и клиентский драйвер для текущего управления и чтения состояния устройства, но опосредованно - через USBД.

Организация обмена по шине

Хост организует обмены с устройствами согласно своему плану распределения ресурсов. Для этого хост-контроллер циклически с периодом 1 мс формирует **кадры** (frames), в которые укладываются все запланированные транзакции! (рис. 17.4). Каждый кадр начинается с посылки пакета-маркера **SOF** (Start Of Frame), который является синхронизирующим сигналом для изохронных устройств, а также для хабов. Кадры нумеруются последовательно, в маркере **SOF** передаются 11 младших битов номера кадра. В режиме HS каждый кадр делится на 8 **микрокадров**, и пакеты **SOF** передаются в начале каждого микрокадра (с периодом 125 мкс). При этом во всех восьми микрокадрах **SOF** несет один и тот же номер кадра; новое значение номера кадра передается в нулевом микрокадре. В каждом кадре (микрокадре) может быть выполнено несколько транзакций, их допустимое число зависит от скорости, длины поля данных каждой из них, а также от задержек, вносимых кабелями, хабами и устройствами. Все транзакции кадров должны быть завершены до начала интервала времени **EOF** (End of Frame). Период (частота) генерации кадров (микрокадров) может немного варьироваться с помощью специального регистра хост-контроллера, что позволяет подстраивать частоту для изохронных передач.

Кадрирование используется и для обеспечения живучести шины. В конце каждого кадра (микрокадра) выделяется интервал времени **EOF**, на время которого хабы запрещают передачу по направлению к контроллеру. Если хаб обнаружит, что с какого-то порта в это время ведется передача данных (к хосту), этот порт отключается, изолируя «болтливое» устройство, о чем информируется USBД.

Full / Low-Speed Frame Size (1 ms)



Хост планирует загрузку кадров так, чтобы помимо запланированных изохронных транзакций и прерываний в них всегда находилось место для транзакций управления. Свободное время кадров может заполняться передачами массивов. К одной шине USB можно подключать устройства, работающие на существенно различающихся скоростях передачи. Чтобы обеспечить рациональное распределение времени кадров (микрокадров), для каждой из скоростей приняты соответствующие ограничения на максимальную длину поля данных пакета. Из этих ограничений вытекают ограничения достижимой пропускной способности устройств и шины в целом:

- На низкой скорости (LS), составляющей 1,5 Мбит/с, в пакете может быть не более 8 байт данных, при этом двухстадийная транзакция управления (без фазы данных) занимает 30 % кадра, а транзакция прерывания - 14 %. Максимальная скорость обмена с одной точкой не превышает 8 Кбайт/с.

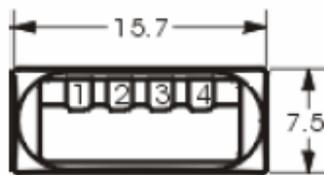


- На полной скорости (FS), составляющей 12 Мбит/с, поле данных для изохронных обменов содержит до 1023 байт (транзакция занимает 69 % кадра), максимальная скорость конечной точки составляет 1023 Кбайт/с. Для остальных типов допустима длина до 64 байт (5 % кадра). При этом, если с конечной точкой в каждом кадре выполняется одна транзакция, достигается скорость 64 Кбайт/с. В один кадр может уместиться до 19 таких транзакций, что соответствует суммарной пропускной способности шины 1,216 Мбайт/с. В принципе такая пропускная способность может достаться и одному устройству (его конечной точке), если другие устройства неактивны.

- На высокой скорости (HS), составляющей 480 Мбит/с, поле данных содержит до 3 x 1024 байт для прерываний и изохронных обменов (14% микрокадра), максимальная скорость конечной точки составляет 24,576 Мбайт/с. Для передач массивов и управляющих передач допустим размер до 512 байт (7-8 % микрокадра), скорость при одной транзакции в микрокадре составляет 4,096 Мбайт/с. На практике одной точке для передач массивов может доставаться пропускная способность около 24 Мбайт/с (при расторопном хост-контроллере и неактивности других устройств). Суммарная пропускная способность шины может достигать 53,2 Мбайт/с. Приведенные значения максимальной суммарной пропускной способности шины достигаются лишь в идеале. Реально они ниже, поскольку транзакции могут выполняться дольше из-за необходимости передачи дополнительных вставленных битов (в худшем случае они удлиняют пакеты в 7/6 раз), а также задержек в кабелях и хабах.

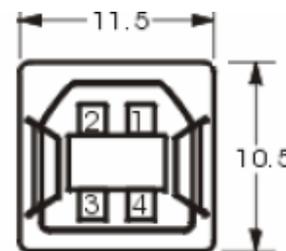
Кабели и разъемы

Сигналы USB передаются по 4-х проводному кабелю. Разъемы, используемые для подключения периферийных устройств, показаны на рисунке.



Series "A"

Предназначены для подключения кабеля к ведущему устройству, т.е. к компьютеру или хабу



Series "B"

Предназначены для подключения кабелей периферийному устройству

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

4 Интерфейс IEEE 1394 FireWire

История IEEE 1394, теперь известного также как FireWire и как i-Link, началась ещё в 1986 году, когда члены

Microcomputer Standards Committee (Комитет по Стандартам Микрокомпьютеров) захотели объединить существовавшие в то время различные варианты последовательной шины (Serial Bus). Новый проект был призван объединить существовавшие на то время разработки: IEEE 1014 VME, IEEE 1296 Multibus II, и IEEE 896 FutureBus+R. Задачей разработчиков стало создание универсального I/O (Input/Output) внешнего интерфейса, пригодного как для работы с мультимедиа, так и для работы с накопителями данных (Mass Storage Device), не говоря уже о более простых вещах - вроде принтеров, сканеров, и тому подобного.

Результатом труда разработчиков стал окончательно утверждённый 12 декабря 1995 года 10 мегабайтный документ под названием 1394-1995.pdf, который описывал IEEE 1394. В названии стандарта нет никакого тайного смысла - просто это был 1394 по счёту стандарт, выпущенный комитетом. Интерфейс, который описывался в этом документе был воистину революционным. Он обеспечивал просто невероятные по тем временам скорости и удобство. Ведущую роль в разработке стандарта сыграла, была Apple, которая дала ему имя FireWire, поэтому нет ничего удивительного в том, что она сразу же сделала ставку на использование этого стандарта в своих компьютерах (как обычно, Apple пошёл своим путём, и, пока пользователи PC заглядывали в рот Intel с недавно появившемся USB, сделала ставку на FireWire. Хотя и USB не был забыт. Настоящей лебединой песней для IEEE 1394 стало появление любительских DV камер. Ещё при их разработке стало ясно, что, кроме IEEE 1394 в качестве внешнего интерфейса для них ничего не подходит. Поэтому, Digital



VCR Conference (DVC) приняла решение использовать IEEE 1394 как стандартный интерфейс для цифровых камер. Первой ласточкой стала Sony с DCRVX1000 и DCR-VX700 цифровыми камерами, которые впервые имели IEEE 1394 выход. Но, вскоре за Sony подтянулись и другие производители. И сегодня IEEE 1394 практически монополизировал этот быстро развивающийся рынок. Сегодня любая, произведённая сегодня DV камера в обязательном порядке оснащается IEEE 1394 интерфейсом.

Свою лепту в развитие IEEE 1394 внесла и Texas Instruments, организовавшая массовое производство действительно дешёвых микросхем для реализации IEEE 1394 интерфейса, что сыграло огромную роль в бурном росте количества IEEE 1394 контролёров в персональных компьютерах.

Несмотря на такой успех нового стандарта (он оказался востребованным ещё до выхода окончательной спецификации), разработчики не стояли на месте. Уже в 2000 году вышла 1394a-2000 версия протокола, сразу же с энтузиазмом воспринятая производителями. А сегодня разрабатывается P1394b. Как уже говорилось, разработчики опирались на выпущенные ранее стандарты, и в IEEE 1394 вошло всё лучшее, что существовало на тот момент. Из главных особенностей IEEE 1394 можно отметить:

- Последовательная шина вместо параллельного интерфейса позволила использовать кабеля малого диаметра и разъёмы малого размера.
- Поддержка горячего подключения и отключения всего чего угодно.
- Питание внешних устройств через IEEE 1394 кабель.
- Высокая скорость
- Возможность строить сети из различных устройств и самой различной конфигурации.
- Простота конфигурации и широта возможностей. Через IEEE 1394 может работать самое различное оборудование, причём пользователю не придётся мучаться вопросом, как это всё правильно подключить.

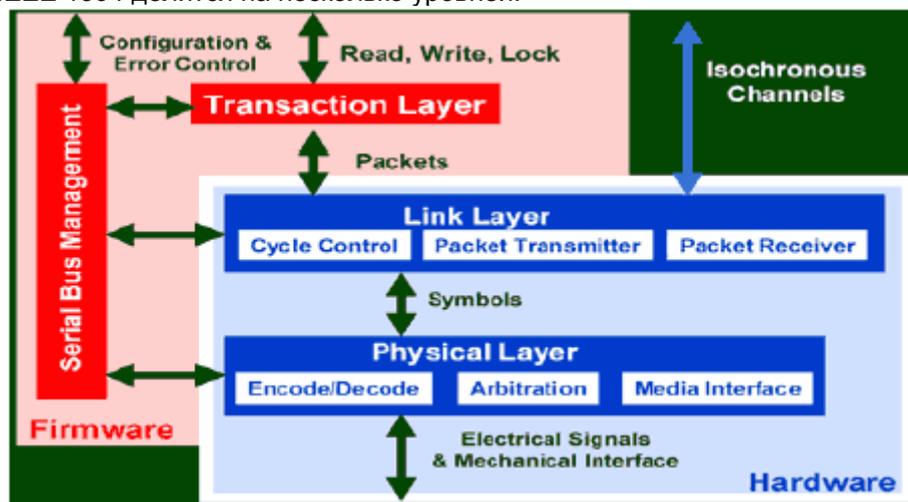
- Поддержка асинхронной и синхронной передачи данных.

На последнем пункте необходимо остановиться поподробнее.

Асинхронная передача. Asynchronous, от греческого Aсyn - другой и Chronous - время. Это означает, что данные обязательно будут доставлены в целости и сохранности, пусть и не всегда в срок. Получение каждого пакета проверяется и подтверждается, если пакет не дошёл, передача будет повторена заново.

Синхронная передача. Isochronous, от греческого Iso - тот же, такой же и Chronous - время. Это означает, что скорость и непрерывность потока важнее, чем сохранность данных. Если пакет пришёл с ошибкой, или не пришёл вообще, это даже не проверяется, не говоря уже о том, чтобы переслать пакет заново. Этот тип передачи отлично подходит для мультимедийных приложений, где потеря какой-либо части информации менее критична, чем большая задержка.

IEEE 1394 делится на несколько уровней.



Внизу находится физический уровень (Physical Layer). Аппаратная составляющая, которая отвечает за перевод сигналов, полученных по кабелям в понятную компьютеру форму (и наоборот - за перевод данных в электрические сигналы, идущие по кабелям). Эта же часть отвечает за управление физическим каналом, т.е. определяет, должно устройство занимать канал прямо сейчас, или должно подождать. Кроме того, этот же уровень обеспечивает интерфейс для кабелей и разъёмов и отвечает за следующие процессы:

Интерфейс среды (Media Interface) - отвечает за состояние сигнала, передаваемого по кабелям.

Арбитраж (Arbitration) - различные IEEE 1394 устройства, включенные в сеть разбираются между собой, кто и в каком порядке может действовать. **Кодирование/Декодирование (Encode/Decode)** - перевод данных в электрические сигналы, которые могут передаваться по кабелям и обратно. Уровнем выше расположен уровень канала (Link Layer). Сюда доставляются уже готовые пакеты данных.

Именно этот уровень отвечает за пересылку данных вверх и вниз, тут происходят следующие процессы:

Приёмник пакетов (Packet Receiver) - организует и отвечает за приём пакетов данных.

Передачик пакетов (Packet Transmitter) - организует и отвечает за передачу пакетов данных.



Контроль циклов (Cycle Control) - пакеты передаются не поодиночке, а циклами. Здесь и осуществляется контроль над этими циклами.

Эти два уровня реализованы в "железе", т.е. выполняются аппаратно. Они полностью отвечают за формирование сигнала из данных, формирование данных из сигнала, и приём/передачу в нужное время и в нужное место. Поэтому, только этих двух уровней и хватает при синхронной передаче, когда никакого контроля над тем что передаётся и получается не требуется. При асинхронной передаче это не так, и там в действие вступает:

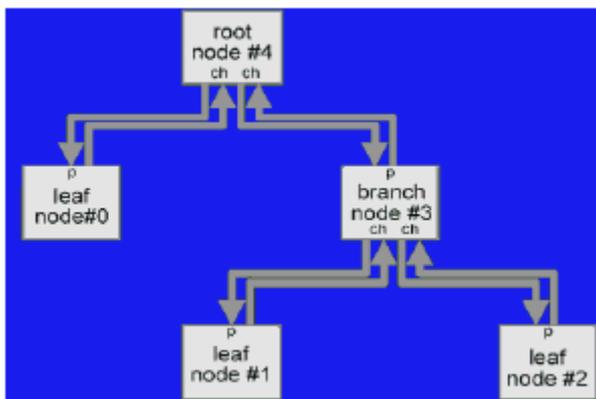
Сетевой уровень (Transaction Layer). На этом уровне происходит проверка полученных данных. Если всё нормально (ни один пакет не потерялся или не повредился), данные отправляются потребителю. Если обнаружена ошибка - возвращаемся на физический уровень и повторяем всё сначала, пока данные не будут получены без ошибок. Все уровни (в том числе и первые два) контролируются firmware, и этот процесс называется менеджмент последовательной шины (Serial Bus management).

Такие процессы происходят в каждом IEEE 1394 устройстве, и два любых устройства образуют между собой соединение типа точка-точка (point-to-point). Но, кроме этого, IEEE 1394 позволяет объединять множество таких устройств и соединений в одну логическую сеть. Для этого физический уровень (physical layer) позволяет иметь больше одного физического интерфейса на одном устройстве.

Рассмотрим подробнее, как разные устройства в одной логической сети разбираются, кто, когда, и что должен делать. Инициализация сети происходит в несколько этапов:

Сброс (reset) - происходит каждый раз, когда требуется. Причиной для сброса может стать, например, физическое изменение конфигурации сети (подключили новое устройство или отключили старое). Со сброса шины и начинается процесс инициализации сети. Конфигурация, сформировавшаяся при этом, остаётся действительной и неизменной до следующего сброса шины.

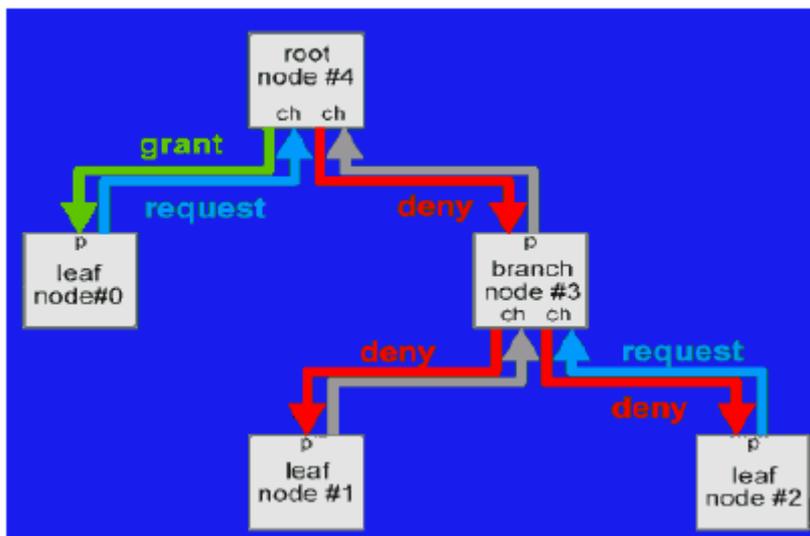
Идентификация дерева (Tree identification) - подключенные устройства выясняют, какие из них родительские, а какие дочерние, и формируют логическое дерево. Определяется корневое устройство для всего дерева. Первое, что определяет устройство после включения, это сколько подключенных портов оно имеет. Один (leaf) или несколько (branch). Затем определяется родительские (parent) и дочерние (child) устройства (какое к какому подключено). На основе этих данных строится дерево и определяется корневое устройство.



Самоидентификация (Self identification) - каждое из устройств получает свой собственный ID узла внутри дерева, и выясняет на каких скоростях могут работать его непосредственные соседи. Топология полностью определена. Для адресации используются принципы, описанные в IEEE 1212. Это означает 64 битную прямую адресацию (48 бит на узел, остальные 16 используются для идентификации шины), что позволяет организовать иерархическую адресацию для 63 узлов на 1023 шинах. Единственное ограничение - между двумя устройствами, которые хотят общаться между собой, должно быть не более 16 "хопов" (сегментов). Инициализация сети завершена, в действие вступает нормальный арбитраж - рабочий режим работы сети. Устройства обмениваются данными, а корневое устройство следит за тем, чтобы они друг другу не мешали.

Происходит это так:

Устройство, которое хочет начать передачу, вначале посылает запрос своему родительскому устройству. Родительское устройство, получив запрос, запрещает передачу всем остальным дочерним (в один момент обрабатывается только один запрос) и, в свою очередь, передаёт запрос дальше, своему родительскому устройству, где всё повторяется. В итоге запрос доходит до корневого устройства, которое, в свою очередь, разрешает передачу тому устройству, чей запрос пришёл первым. Всем остальным передача запрещается. Таким образом, если два устройства одновременно пошлют запрос на передачу данных, то ответ будет зависеть от того, чей запрос первым достигнет корневого устройства. Оно выигрывает арбитраж и получает право начать передачу. Проигравшее устройство. Не получив разрешения на передачу, вынуждено ждать, пока выигравшее не освободит шину.



Всё это происходит на физическом уровне (physical layer). После того, как разрешение на передачу данных получено и требуется начать передачу данных, в дело вступает уровень канала (link layer). Как уже говорилось, именно он формирует пакеты и определяет - когда и сколько пакетов должно отсылаться. Передача данных начинается с запроса готовности к приему устройства, для которого предназначены данные, и, получив подтверждение готовности, начинает передачу. Данные идут пакетами, между которыми есть промежутки (gap). Типичный пакет данных 256 байт, или 2048 бит, из которых 160 бит приходится на заголовок. Таким образом, общая эффективность (сколько в пакете действительно данных, а не служебной информации) весьма высока и чем больше пакет, тем выше эффективность). В заголовок входит информация об отправителе, получателе и CRC. После пакета идёт небольшой промежуток, длиной меньше 0.75 msec (acknowledge gap), после чего получатель должен выслать 8-ми битовый блок данных, подтверждающий, что пакет получен в целостности и сохранности (ack packet). Потом следует более длинный промежуток, длиной больше 1 msec, разделяющий пакеты (subaction gap). И так далее - пакет, acknowledge gap, подтверждающий байт (ack), subaction gap.

Для того, чтобы одно устройство, начав передавать данные, не заняло весь канал, не оставив соседям никаких шансов начать передачу, пока оно не закончит, введено понятие fairness interval. В течении одного fairness interval каждое устройство в шине получает одну возможность передать свои данные. После того как разрешение получено (арбитраж выигран), и порция данных передана, устройство должно ждать конца fairness interval и начала следующего цикла, прежде чем оно вновь получит возможность передать следующую порцию данных. Заканчивается fairness interval так называемым reset gap, который длиннее subaction gap, и вызывает сброс всей шины.

Для синхронной передачи используется несколько другая методика. Данные передаются "выстрелами", длина каждого 125 msec. Таких выстрелов производится столько, сколько позволяет канал. Даже на одинарной (98.304 Mbit/sec) скорости за один такой цикл передаётся до 1000 байт. Чем выше скорость, тем больше данных успевают пройти. При этом, при синхронной передаче абсолютно не важно, получило принимающее устройство данные или нет. Пакеты просто идут один за другим, разделённые subaction gap, никаких ack packet никто не ждёт. Для того, чтобы принимающее устройство смогло разобраться, где синхронные, а где асинхронные данные, subaction gap при синхронной передаче короче. Это позволяет комбинировать в одном сеансе синхронные данные с асинхронными. Однако, в синхронном режиме одному устройству никогда не позволяют захватить весь доступный канал. На синхронные

данные может приходиться не более 85% доступного канала, причём одно устройство не может занять больше 65%.

IEEE 1394 позволяет передавать данные на скорости 98.304 Mbit/sec. Кроме этого, возможна передача в 2-х (196.608 Mbit/sec) и 4-х (393.216 Mbit/sec) режимах.

Первоначально появились чипы, которые способны работать только на 100 Мбитах (хотя спецификация позволяла и больше), но 200 и 400-мегабитные чипы не заставили себя долго ждать. Несмотря на такой кажущийся беспорядок, пользователи не должны испытывать ни малейших неудобств (это было одно из обязательных условий, которое ставилось перед разработчиками). Поэтому IEEE 1394 позволяет в одной сети использовать самые разные устройства одновременно. Причём, пользователю не придётся беспокоиться о том, что он может неправильно их подключить. Подключать можно что угодно, и в каких угодно сочетаниях, железки сами разберутся, кто с кем и на какой скорости может "разговаривать".

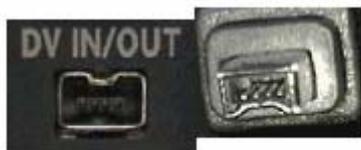
Для работы на таких высоких скоростях потребовались соответствующие кабели. Кабель для IEEE 1394 весьма сложная система, и спаять его самостоятельно (что возможно для USB) вряд ли возможно. Данные передаются по двум витым парам, каждая из которых отдельно экранирована. Для пущей надёжности, дополнительно экранируется и весь кабель. Кроме двух сигнальных пар, в кабеле предусмотрены две питающие жилы, которые могут обеспечить любое внешнее устройство током силой до 1.5 А и напряжением до 40 V. В разрезе кабель выглядит так:



Выбору разъёма, к которому должны подключаться IEEE 1394 устройства, было уделено самое пристальное внимание, ведь от разъёма в немалой степени зависит то, насколько удобно будет пользоваться новым интерфейсом. Разъём должен быть небольшим, но в то же время прочным, должен обеспечивать надёжное соединение, но в то же время легко соединяться-отсоединяться даже вслепую. Всем требованиям удовлетворил разъём, используемый в Nintendo GameBoy.



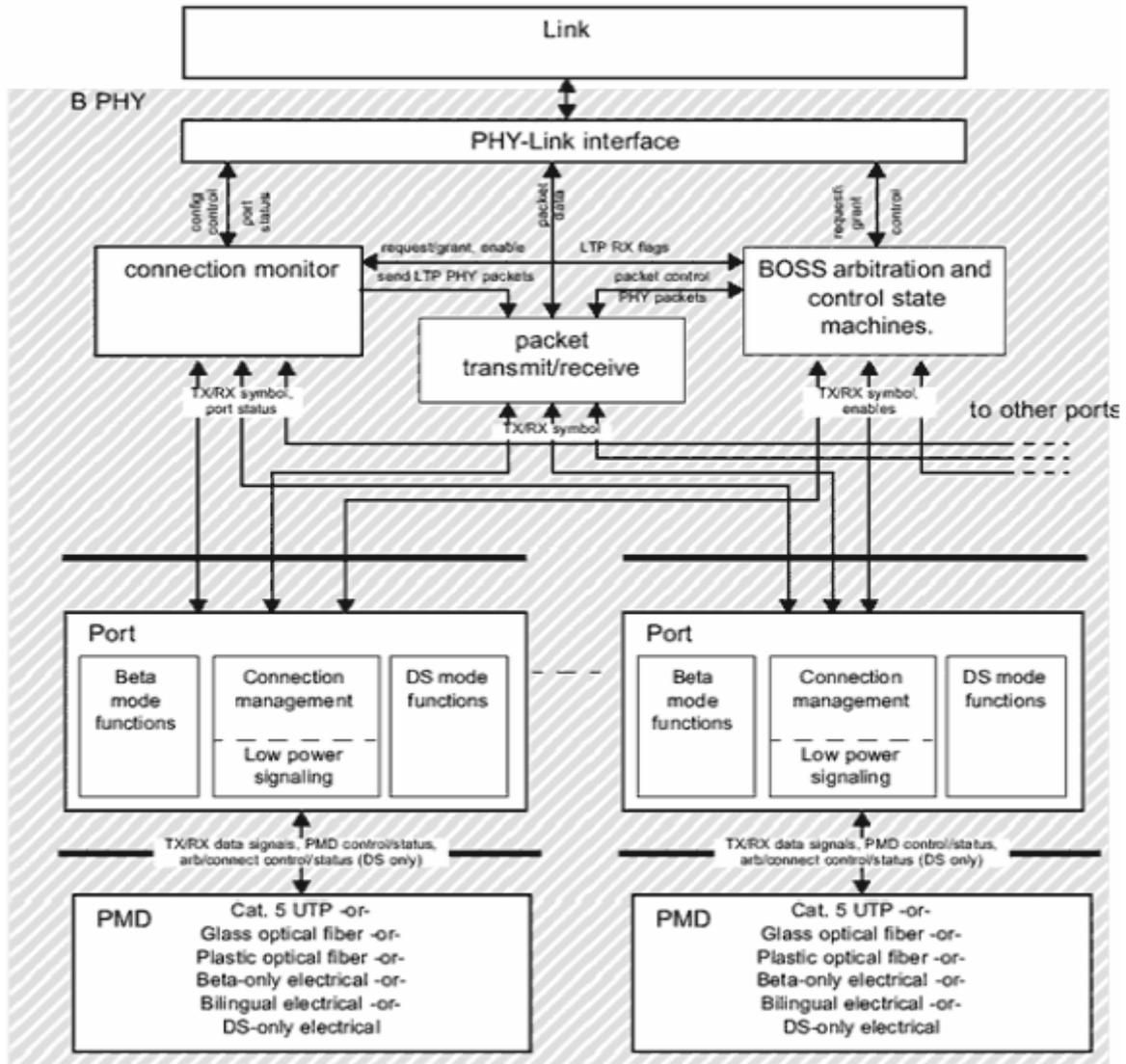
Как видно из фотографии, все контакты выведены в середину разъёма, а снаружи защищены толстым ободком из твёрдой пластмассы. Надёжность этой схемы доказана основательной его проверкой на устройствах GameBoy. Кроме того, имеется четырехконтактная версия разъёма без проводов питания.



Перспектива

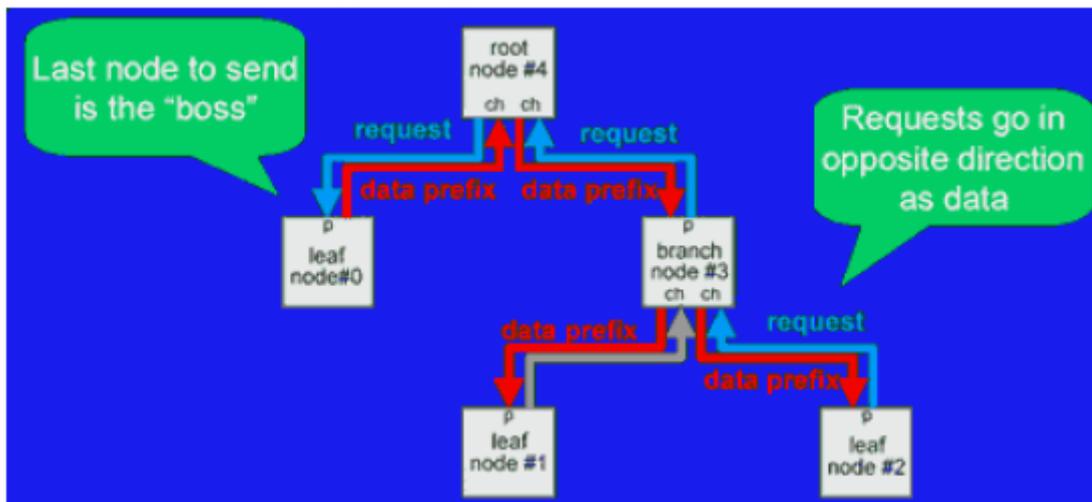
В настоящее время идут работы над 1394b версией стандарта, которая должна стать большим шагом вперёд в его развитии. Изменения коснутся всего, в том числе физической составляющей (кабелей и разъёмов).

Главное, что обещают в 1394b, это появление новых скоростей, в 800 и 1600 мегабит в секунду (возможно, и 3200 мбит/сек), и увеличение максимальной длины кабеля до 50, 70 и даже до 100 метров. Изменились разъёмы, которые могут использоваться, и кабели. В 1394b могут применяться даже простые UTP кабели 5 категории, но только на скоростях до 100 мбит/сек. Для достижения максимальных скоростей на максимальных расстояниях предусмотрено использование оптики (fiber optics), пластмассовой - для длины до 50 метров, и стеклянной - для длин до 100 метров. Архитектура 1394b устройства выглядит так:



метод арбитража. В отличие от 1394 и 1394a, где главным арбитром во всех спорах из за шины было корневое устройство в дереве, в 1394b такие права может получить то устройство, которому они наиболее нужны. Новый метод арбитража называется BOSS (Bus Owner/Supervisor/Selector). Конечно же, наряду с этим, поддерживаются и методы арбитража из 1394 и 1394a. Рассмотрим новый метод подробнее.

Bus Owner/Supervisor/Selector (BOSS) Этот метод арбитража может работать только в 1394b среде. То есть, если в сети есть хоть одно старое устройство, для арбитража будет применяться старый метод. Принципиальная схема BOSS метода выглядит так:



Устройство, которому требуется переслать данные, постоянно шлёт запросы. В результате, когда предыдущий BOSS заканчивает передачу, то ждущее и посылающее в данный момент запросы устройство



оказывается последним, пославшим запрос (все остальные либо уже закончили либо ещё не начали передачу), и получает права BOSS. Что означает, что пока передаёт данные - оно контролирует шину. Как только передача закончена, то устройство остаётся BOSS до тех пор, пока кто-то ещё не пошлёт запрос на передачу. Как только запрос послан - пославшее его устройство немедленно становится BOSS. Эта схема может работать только благодаря full-duplex природе 1394b, ибо, в противном случае, постоянно гуляющие по сети запросы помешали бы передаваемым данным.

6 Сетевые интерфейсы

Сетевая модель OSI (*базовая эталонная модель взаимодействия открытых систем*, [англ. Open Systems Interconnection Basic Reference Model](#)) — абстрактная [сетевая модель](#) для коммуникаций и разработки [сетевых протоколов](#). Представляет уровневый подход к [сети](#). Каждый уровень обслуживает свою часть процесса взаимодействия. Благодаря такой структуре совместная работа [сетевого оборудования](#) и [программного обеспечения](#) становится гораздо проще и прозрачнее.

Физический уровень — самый нижний уровень [сетевой модели OSI](#), предназначен непосредственно для передачи потока данных. Осуществляет передачу электрических или оптических сигналов в кабель или в радиоэфир и, соответственно, их приём и преобразование в биты данных в соответствии с [методами кодирования цифровых сигналов](#). Другими словами, осуществляет интерфейс между сетевым носителем и сетевым устройством.

На этом уровне работают [концентраторы](#) (хабы), [повторители](#) (ретрансляторы) сигнала и медиаконверторы.

Функции физического уровня реализуются на всех устройствах, подключенных к сети. Со стороны компьютера функции физического уровня выполняются сетевым адаптером или последовательным портом. К физическому уровню относятся физические, электрические и механические интерфейсы между двумя системами. Физический уровень определяет такие свойства среды сети передачи данных как оптоволокно, витая пара, коаксиальный кабель, спутниковый канал передачи данных и т. п. Стандартными типами сетевых интерфейсов, относящимися к физическому уровню, являются: [V.35](#), [RS-232C](#), [RJ-11](#), [RJ-45](#), разъемы [AUI](#) и [BNC](#).

В настоящее время основным используемым семейством протоколов является [TCP/IP](#), разработка которого не была связана с моделью OSI.



Стандартные сетевые интерфейсы

- [IEEE 802.2](#) — [LLC](#) (Logical Link Control) — управление логическими соединениями
- [IEEE 802.3](#) — технология [Ethernet](#)
- [IEEE 802.4](#) — маркерная шина (token bus)
- [IEEE 802.5](#) — [маркерное кольцо](#) (англ. *token ring*)
- [IEEE 802.6](#) — Metropolitan Area Network, MAN - сети мегаполисов.
- [IEEE 802.7](#) — Broadband Technical Advisory Group - техническая консультационная группа по широкополосной передаче.
- [IEEE 802.8](#) — Fiber Optic Technical Advisory Group - техническая консультационная группа по волоконно-оптическим сетям.
- [IEEE 802.9](#) — Integrated Voice and data Networks - интегрированные сети передачи голоса и данных.
- [IEEE 802.10](#) — Network Security - сетевая безопасность.
- [IEEE 802.11](#) — высокоскоростные беспроводные локальные сети WiFi
- [IEEE 802.12](#) — Demand Priority Access LAN, 100VG-AnyLAN - локальные сети с методом доступа по требованию с приоритетами.
- [IEEE 802.15](#) — [Беспроводные персональные сети \(WPAN\)](#), [Bluetooth](#)
- [IEEE 802.16](#) — беспроводная городская сеть, [WiMAX](#)
- [IEEE 802-2001](#) — Стандарт для локальных и региональных вычислительных сетей. Обзор и архитектура.
- [IEEE 1149](#) — Стандарт периферийного сканирования микросхем ([Boundary Scan](#)) - тестирование, программирование и локализация неисправностей печатных плат.
- [IEEE 1284](#) — параллельный интерфейс



- [IEEE 1394](#) — FireWire(i-Link)— последовательная высокоскоростная шина, предназначенная для обмена цифровой информацией между компьютером и другими электронными устройствами.
- [IEEE 1059](#) — Guide for Software Verification and Validation Plans — Руководство по планированию верификации и подтверждения достоверности программного обеспечения.

IEEE 802.3 - стандарты [IEEE](#), касающиеся функционирования сетей. Семейство этих протоколов так же называется [Ethernet](#)

6.1 Стандарты IEEE 802.3

Номер стандарта	Дата принятия	Описание
Experimental Ethernet	1972	2.94 Мбит/с (367 кБайт/с) через коаксиальную шину (кабель)
Ethernet II (DIX v2.0)	1982	10 Мбит/с (1.25 МБайт/с) через тонкий коаксиальный кабель (англ. <i>thinnet</i>)— у фреймов появляется поле типа (данных).
IEEE 802.3	1983	10BASE5 10 Мбит/с (1.25 МБайт/с) через толстый коаксиальный кабель (аналогичен Ethernet II, за исключением замены поля типа на поле «размер» и добавлением LLC заголовка IEEE 802.2 , следующего за заголовком IEEE 802.3.
802.3a	1985	10BASE2 10 Мбит/с (1.25 МБайт/с) через тонкий коаксиальный кабель
802.3b	1985	10BROAD36
802.3c	1985	10 Мбит/с (1.25 МБайт/с) спецификации повторителя
802.3d	1987	FOIRL (англ. <i>Fiber-Optic Inter-Repeater Link</i> , оптоволоконные линии между повторителями)
802.3e	1987	1BASE5 или StarLAN
802.3i	1990	10BASE-T 10 Мбит/с (1.25 МБайт/с) через витую пару (3-ей категории)
802.3j	1993	10BASE-F 10 Мбит/с (1.25 МБайт/с) через оптоволоконно
802.3u	1995	100 Мбит/с (12.5 МБайт/с), автосогласование скорости (совместимость с IEEE 802.3i)
802.3x	1997	поддержка полнодуплексной связи; совместимость с DIX
802.3y	1998	100BASE-T2 100 Мбит/с (12.5 МБайт/с) через низкокачественную витую пару
802.3z	1998	1000BASE-X GigabitEthernet через оптоволоконный кабель; 1 Гбит/с (125 МБайт/с)
802.3-1998	1998	Версия, включающая в себя все предыдущие стандарты с исправленными ошибками.
802.3ab	1999	1000BASE-T GigabitEthernet по витой паре; 1 Гбит/с (125 МБайт/с)
802.3ac	1998	Увеличение максимального размера фрейма до 1522 байт (для поддержки информации о VLAN стандарта IEEE 802.1Q и приоритета стандарта IEEE 802.1p)



802.3ad	2000	Агрегация каналов
802.3-2002	2002	Версия, включающая в себя все предыдущие стандарты с исправленными ошибками
802.3ae	2003	10 Гбит/с (1,250 МБайт/с) Ethernet через оптоволокно; 10GBASE-SR , 10GBASE-LR , 10GBASE-ER , 10GBASE-SW , 10GBASE-LW , 10GBASE-EW
802.3af	2003	PoE — электропитание через Ethernet (Power over Ethernet)
802.3ah	2004	Ethernet in the First Mile ("Первая миля")
802.3ak	2004	10GBASE-CX4 10 Gbit/s (1,250 МБайт/с) Ethernet over twin-axial cable
802.3-2005	2005	Ревизия основного стандарта, включающая четыре предшествующих изменения.
802.3an	2006	10GBASE-T 10 Gbit/s (1,250 МБайт/с) Ethernet по неэкранированной витой паре(UTP)
802.3ap	2007	Backplane Ethernet (1 and 10 Gbit/s (125 and 1,250 МБайт/с) для объединительной платы)
802.3aq	2006	10GBASE-LRM 10 Gbit/s (1,250 МБайт/с) Ethernet по многомодовому оптическому волокну
802.3ar	Hold	On Управление перегрузкой
802.3as	2006	Расширение формата кадров
802.3at	2008	exp. Ethernet Питание через Ethernet повышение мощности питания по Ethernet
802.3au	2006	exp. Требования изоляции для Питание через Ethernet (802.3-2005/Cor 1)
802.3av	2009	exp. 10 Gbit/s EPON
802.3aw	2007	exp. Fixed an equation in the publication of 10GBASE-T (released as 802.3-2005/Cor 2)
802.3ax	2008	exp. Move Link aggregation out of 802.3 to IEEE 802.1
802.3ay	2008	exp. Maintenance to base standard
802.3ba	2009	exp. 10m Cu cable assembly (4x25 Gbit or 10x10 Gbit lanes) and 100 m of MMF and 100 Gbit/s up to 10 m or Cu cable assembly, 100 m of MMF or 40 km of SMF respectively Выборка самых распространённых интерфейсов Эзернет 802.3

Стандарт	Дата принятия	Описание
802.3		10BASE5 , («Толстый Ethernet»)
802.3a		10BASE2 («Тонкий Ethernet»)
802.3i		10BASE-T (витая пара)
802.3j		10BASE-F (оптическое волокно)
802.3u	1995	Fast Ethernet



802.3z	1997	Gigabit Ethernet
802.3ab	1999	Gigabit Ethernet по медному кабелю 5-й категории
802.3an		10GBaseT — 10 Gigabit Ethernet по медному кабелю 6-й (7-й) категории
802.3ad		Агрегация каналов
802.3x		Управление потоком в полнодуплексном режиме