

Интерфейсы периферийных устройств

Курс лекций в форме презентации

Автор и разработчик: преподаватель СФТИ НИЯУ МИФИ

Чернышев Олег Юрьевич

Часть 4

Снежинск

2012

Стандартные интерфейсы

ВВОДА-ВЫВОДА

Для понимания необходимости и смысла интерфейса между компьютером и устройствами ввода-вывода, прежде всего, важно представлять, что собой представляет компьютерная система (любая), как соединяются ее компоненты, и что требуется сделать для подключения к ней периферийных устройств ввода-вывода. Задача заключается в том, что для взаимодействия устройств ввода-вывода, необходимо обеспечить их связь с системной шиной, по которой передаются (пересылаются) от процессора ко всем устройствам, независимо где они расположены территориально, управляющие сигналы (команды), данные и информация. Эта в целом информация двунаправлена, то есть пересылается во взаимно противоположных направлениях.

Системная шина (иногда ее не совсем точно называют шиной процессора) территориально расположена вне процессора, но она управляется теми же сигналами, что и схемы процессора, то есть **имеет вполне конкретные ограничения электрической природы**. В этом случае говорят, что эта шина **аппаратно** зависима от процессора. К ней могут подключаться только устройства, работающие с сигналами, параметры которых совпадают с параметрами сигналов процессора и одинаков «темп работы» - скорость взаимодействия с процессором. Универсальной эту шину сделать невозможно. Поэтому непосредственно к этой шине может быть подключено ограниченное число устройств, в частности, таких, как оперативная память. Следовательно, нужна шина, которая не имела бы столь жестких требований и ограничений, и позволяла бы подключать к ней совершенно разнотипные устройства ввода-вывода. Выход был найден в разработке, так называемых, **шин расширения**, для которых можно использовать стандартную схему сигналов. Был разработан целый ряд соответствующих стандартов. Одни стандарты являются официальными формальными, другие стали «стандартами де-факто», то есть общепризнанными, содержащими приемлемые для большинства соглашения.

Здесь будут рассматриваться стандартные интерфейсы персональных компьютеров. Принципиального значения это не имеет, так как решение вопросов согласования периферийных устройств с любым типом компьютеров хотя и имеет свою специфику, но в основном использует общие положения и правила разработки интерфейсов этого назначения. Наоборот, названные интерфейсы разработаны для компьютеров самого широкого применения и распространения. Это свидетельствует о том, что они достаточно хорошо отработаны и требуют меньших затрат в производстве. Это означает также, что проработаны общие для всех подобных интерфейсов вопросы, связанные с их функциональным назначением. Например, IBM разработала для своего персонального компьютера шину ISA (Industry Standard Architecture), которая вместе с компьютером стали настолько популярны, что производители устройств ввода-вывода стали снабжать свои устройства ISA-совместимыми интерфейсами, и ISA стал популярным де-факто.

На рисунке 4.1 показано, как шины различных типов интерфейса ввода-вывода используются в типичной компьютерной системе.

Из названного рисунка можно заключить, что стандарт PCI определяет шину расширения на системной (материнской) плате. Шины стандарта SCSI и USB предназначены для подключения дополнительных устройств как внутри, так и вне корпуса компьютера. SCSI представляет собой высокоскоростную параллельную шину, предназначенную для подключения таких устройств как диски и дисплеи. Шина USB поддерживает последовательную передачу данных. Она используется для подключения самого разнообразного оборудования,

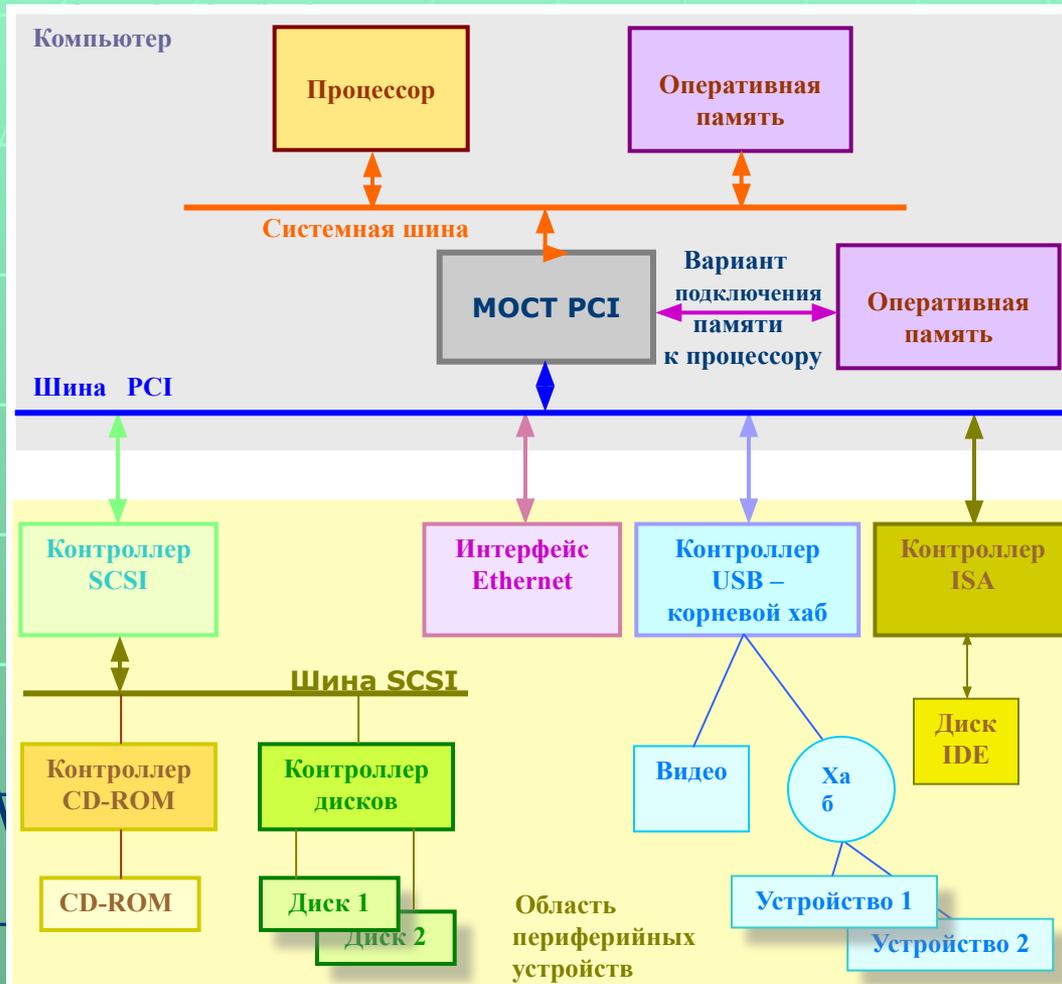


Рис. 4.1. Пример компьютерной системы, в которой используются различные стандартные интерфейсы

вых устройств, а также для Интернет-интерфейсов. На рисунке показана интерфейсная шина («Интерфейс ISA»), устаревшего стандарта, позволяющая подключение к компьютеру популярны в свое время, диски IDE (Integrated Drive Electronics). Данная шина используется и для подключения к локальной сети Ethernet – это широко распространенная архитектура локальных сетей, обеспечивающая высокоскоростное соединение компьютеров в здании, или в нескольких близко расположенных зданиях. В общем, в компьютере может использоваться сразу несколько различных типов шин.

Шина PCI

Шина PCI – это разновидность системной шины, появившейся в ответ на потребность в стандартизации используемых периферийных устройств. Она поддерживает функции, типич-

ные для шины процессора, но в стандартизованном формате, независимо от типа процессора.

Подключенные к шине PCI периферийные устройства представляются процессору непосредственно соединенными с его собственной шиной. В таких случаях говорят, что шина PCI для процессора является прозрачной. Подключенным к шине периферийным устройствам назначаются адреса из адресного пространства памяти процессора.

Шина PCI унаследовала многие принципы шинных стандартов, применяющихся преимущественно в компьютерах IBM PC. Последним прототипом шины PCI являлась 16-разрядная шина ISA. В 1980-е годы были разработаны и другие шины со сходными возможностями, среди которых наиболее известными считались Microchannel и NuBus, использовавшиеся соответственно в компьютерах IBM PC и Macintosh.

Создавалась шина PCI как недорогое устройство, по-настоящему независимое от процессора. Потребность в ее разработки обуславливалась необходимостью поддержки работы высокоскоростных дисковых и графических устройств, а также специфическими нуждами многопроцессорных систем. Шина до сих пор популярна как промышленный стандарт.

Важная особенность шины заключается в том, что она явилась пионером нового механизма подключения устройств ввода-вывода, получившего название **Plug-and-play** (подключи и играй). Теперь достаточно вставить интерфейсную карту (плату) в разъем на шине, а все остальное будет сделано программным обеспечением компьютера.

Пересылка данных

В современных компьютерах при выполнении операции пересылки данных поддерживается режим перемещения целого блока информации. Шина PCI рассчитана на поддержку именно такого режима. Операция чтения или записи одного слова интерпретируется ею как чтение или запись пакета длиной в одно слово.

Шина поддерживает три независимых адресных пространства: памяти, ввода-вывода и конфигурации. Напомним, что под понятием конфигурирования понимается процесс подключения или отключения устройств. Адресное пространство памяти необходимо для взаимодействия с ней различных устройств, а адресное пространство ввода-вывода используется как самостоятельное некоторыми типами компьютеров. Для ввода-вывода может использоваться и адресное пространство памяти – такой режим называют **отображением в память**, даже тогда, когда процессор поддерживает отдельное пространство ввода-вывода. **Конфигурационное адресное пространство** предназначено для поддержки технологии **Plug-and-play**. Указанием на то, какое адресное пространство будет использовано процессором в предстоящем процессе пересылки данных является специальная 4-разрядная команда, сопровождающая адрес.

Согласно терминологии PCI, хозяин шины называется **инициатором**. Им может быть либо процессор, либо контроллер ПДП. Устройство, к которому адресуется хозяин шины и отвечающее на команды «чтение» или «запись», называется **целевым**.

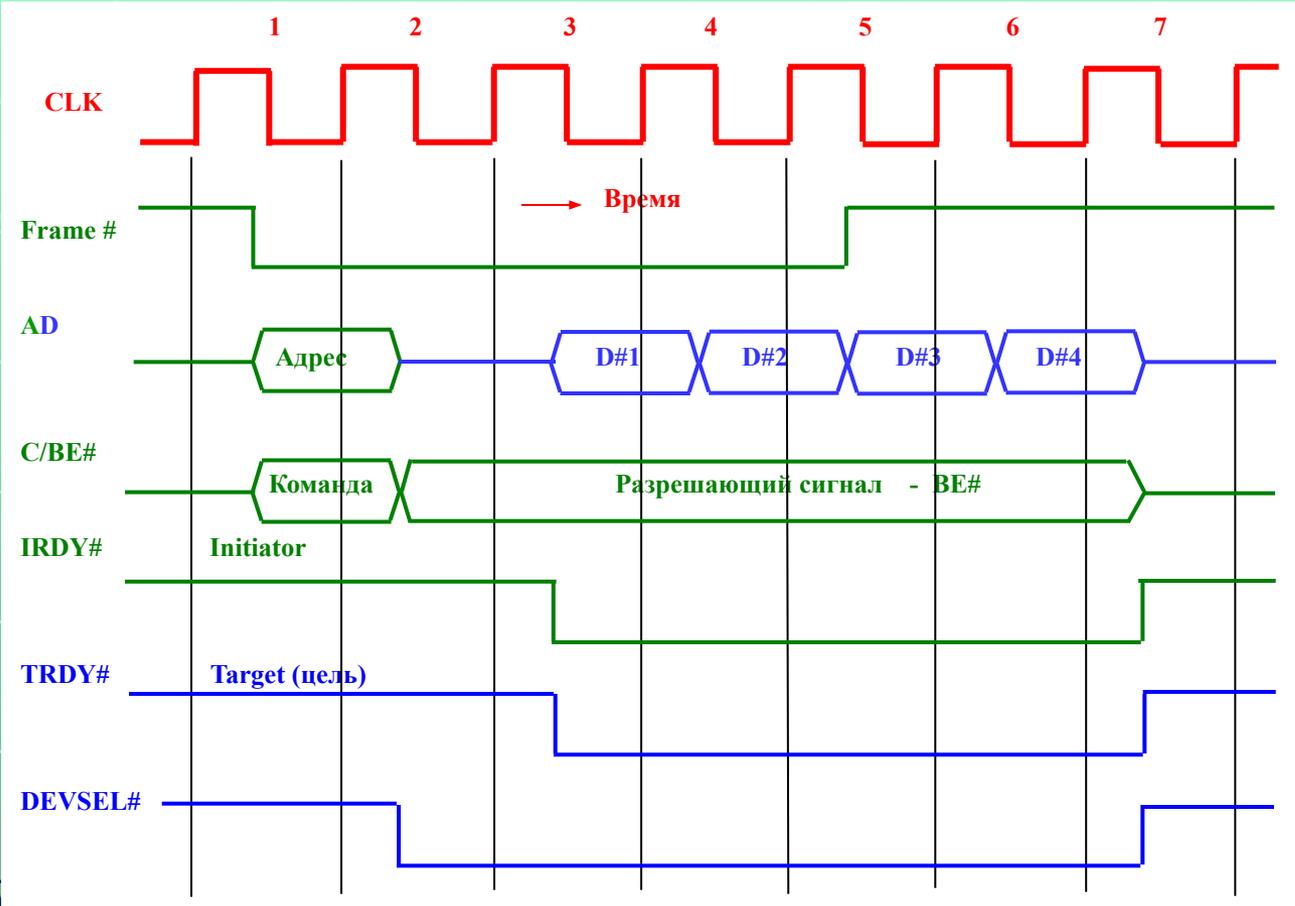


Рис. 4.2. Операция чтения на шине PCI

Чтобы понять работу шины, давайте рассмотрим типичную операцию пересылки данных. Процесс пересылки сопровождается работой определенных сигналов. На следующем слайде приведена таблица 4.1, описывающая команды и функции, реализуемые на шине, и сигналами. Сигналы, в которых присутствует разряд (#) являются низковольтными. Давайте проанализируем операцию в результате которой инициатор (хозяин шины) считывает из памяти 32-разрядное слово. Следовательно, память в нашем примере является целевым устройством. Полная операция

пересылки данных по шине, включающая пересылку адреса и пакета данных, называется **транзакцией**. Пересылка отдельного слова в ходе транзакции, называется **фазой**. Последовательность событий на шине показана на рисунке 4.2. Тактовый сигнал используется для координации различных фаз транзакции. Все изменения сигналов инициируются передним фронтом тактового сигнала. На рисунке показаны временные задержки сигналов относительно передних фронтов тактовых импульсов. События на диаграмме показаны в следующей последовательности.

Таблица 4.1. Сигналы пересылки данных по шине PCI

Имя		Описание
CLK	Clock	Тактовый сигнал с частотой 33 или 66 МГц
Frame#	- кадр	Активизируется инициатором с целью определения длительности транзакции
AD	Address-Data	Представляет 32 линии для пересылки адресов и данных (количество линий при необходимости может быть увеличено до 64)
C/BE#	Command/ Byte Enable	Представляет 4 линии для команды считывания (Command) и массив, указывающий, какие байты подлежат считыванию
IRDY#	Initializer-ready	Сигнал готовности инициатора
TRDY#	Target-ready	Сигнал готовности целевого устройства
DEVSEL#	Device Select	Ответ устройства, указывающий, что оно распознало свой адрес и готово к операции пересылки данных
IDSEL#	Initialization Device Select	Сигнал выбора инициализируемого устройства

На такте 1 хозяин шины помещает сигнал Frame#, сообщая целевому устройству о начале транзакции. Одновременно он помещает адрес на линии AD и команду на линии C/BE#. В нашем случае команда указывает, что запрошена операция чтения и что используется адресное пространство памяти.

Такт 2 предназначен для переключения линий шины AD. Инициатор удаляет с них адрес и выключает свои выходные цепи. Целевое устройство, при этом, опознав, что инициатор обращается именно к нему, включает свои входные цепи на линиях шины AD, сигналом DEVSEL# информирует инициатор о том, что оно будет с ним взаимодействовать, и в начале такта 3 помещает на линии запрошенные данные D#1, D#2, D#3 и т.д. Сигнал DEVSEL# поддерживается в течение всего времени пересылки данных (до конца транзакции).

Четыре линии C/BE#, используемые на такте 1 для передачи команды (ввод или вывод), далее используют для указания количества одновременно пересылаемых байтов данных. Если пересылаются 4 байта данных, то на каждой линии устанавливается 1.

Через некоторое время после того, как на линиях шины появится адрес, и он будет распознан целевым устройством, инициатор активизирует сигнал **IRDY#** - в нашем примере это происходит в начале такта 3. Таким образом инициатор сообщает целевому устройству о готовности к получению данных.

Для определенности давайте инициатором будем считать контроллер ПДП диска, а целевым устройством – оперативную память.

Если целевое устройство (память) уже готово к отправке данных, оно устанавливает сигнал **TRDY#** и помещает данные на линиях данных шины. В нашем случае это происходит в начале такта 3. В конце этого же такта инициатор загружает данные в свой входной буфер. На тактах 4-6 целевое устройство отсылает инициатору еще три слова данных.

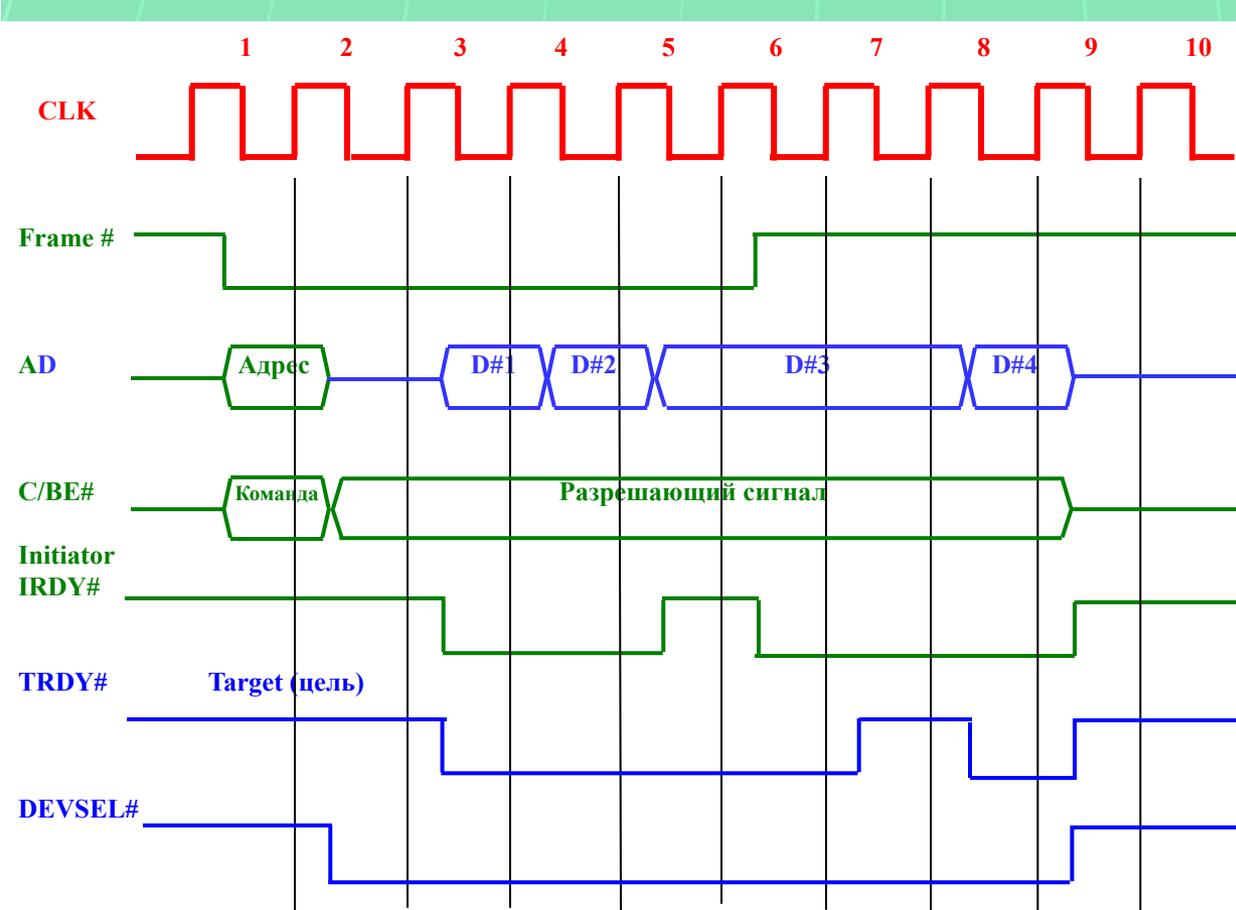


Рис. 4.3. Роль сигналов **IRDY#** и **TRDY#** в операции чтения

Длительность сигнала **FRAME#**, и в прочем, определяет длину пакета данных. Во время начала передачи предпоследнего слова инициатор снимает с линии этот сигнал, отмечая окончание пакета. В нашем примере это происходит на такте 5, во время которого пересылается третье (предпоследнее) слово данных. После отправки четвертого слова инициатор снимает с линии этот сигнал, и в начале такта 7 снимает с линии сигнал **DEVSEL#**.

На рисунке 4.3 приведен более подробный пример, иллюстрирующий расширенные возможности.

Спомощью сигналов IRDY# и TRDY# инициатор и целевое устройство могут делать паузы в процессе проведения транзакции. В нашем примере целевое устройство отсылает инициатору третье слово на такте 5. Предположим, что в этот момент инициатор не готов к его получению. Тогда он снимает сигнал IRDY#, а целевое устройство сохраняет третье слово на линиях AD до тех пор, пока сигнал IRDY# не появится снова. На такте 6 инициатор активизирует сигнал IRDY# и в конце этого такта мог бы загрузить данные в свой буфер, но теперь целевое устройство по какой-то причине оказалось не готово к немедленной отправке четвертого слова данных инициатору. В начале такта 7 оно снимает сигнал TRDY#, но в начале такта 8 снова включает его и одновременно помещает четвертое слово на линии AD. Поскольку на третьем слове данных сигнал FRAME# снимается, то после пересылки четвертого слова данных транзакция завершается.

Конфигурирование устройств

Конфигурирование заключается в физическом (аппаратное) и программном подключении (или отключении) устройства к компьютеру или компьютерной системе. Для обеспечения начала взаимодействия соединяемых систем необходимо кроме физического, еще и логическое подключение. При этом требуется некоторая информация о подключаемом устройстве. Компьютеру необходимо знать, какое устройство подключается, сколько адресов ему требуется выделить (для адреса буфера, адреса регистра состояния и т.д.), по каким линиям оно может передавать сигналы запроса прерывания, с какой скоростью может работать и др. Такая информация помещается в специальную конфигурационную память устройства ROM-память небольшого объема. Но как компьютеру взять эту информацию от устройства. На рисунке 4.4 схематически показан способ подключения интерфейсной платы устройства ввода-вывода к шине PCI компьютера. На этой шине старшие 21 линия (AD11 – AD 31) используются для адресования к подключаемым устройствам, то есть к шине можно подключить не более 21 устройства. Каждое устройство имеет соответственно контакт с одним из названных проводников шины, по другому сказать имеет один из названных аппаратных адресов, находящийся в состоянии 1, остальные линии для этого устройства установлены в 0. Через этот контакт на устройство подается сигнал IDSEL# (Initialization Device Select – выбор инициализируемого устройства). Выбор устройства выполняется с помощью этого сигнала, а не с помощью адреса, поданного на AD-входы устройства. Младшие адресные линии, от AD10 до AD00, используются для задания типа операции и для доступа к содержимому конфигурационной ROM устройства.

После установки интерфейсной платы устройства ввода-вывода в разъем шины и включения питания компьютера, конфигурационное программное обеспечение сканирует 21 область конфигурационного адресного пространства,

После установки интерфейсной платы устройства ввода-вывода в разъем шины и включения питания компьютера, конфигурационное программное обеспечение сканирует 21 область конфигурационного адресного пространства, чтобы узнать, какие устройства подключены к компьютеру. При этом считывается содержимое ROM-памяти устройства, на основании которого определяется тип устройства, его опции и характеристики, после анализа которых устройству назначается адрес (или адреса). Назначенный устройству адрес записывается в один из его регистров (или в несколько регистров). Кроме того, конфигурационное программное обеспечение устанавливает

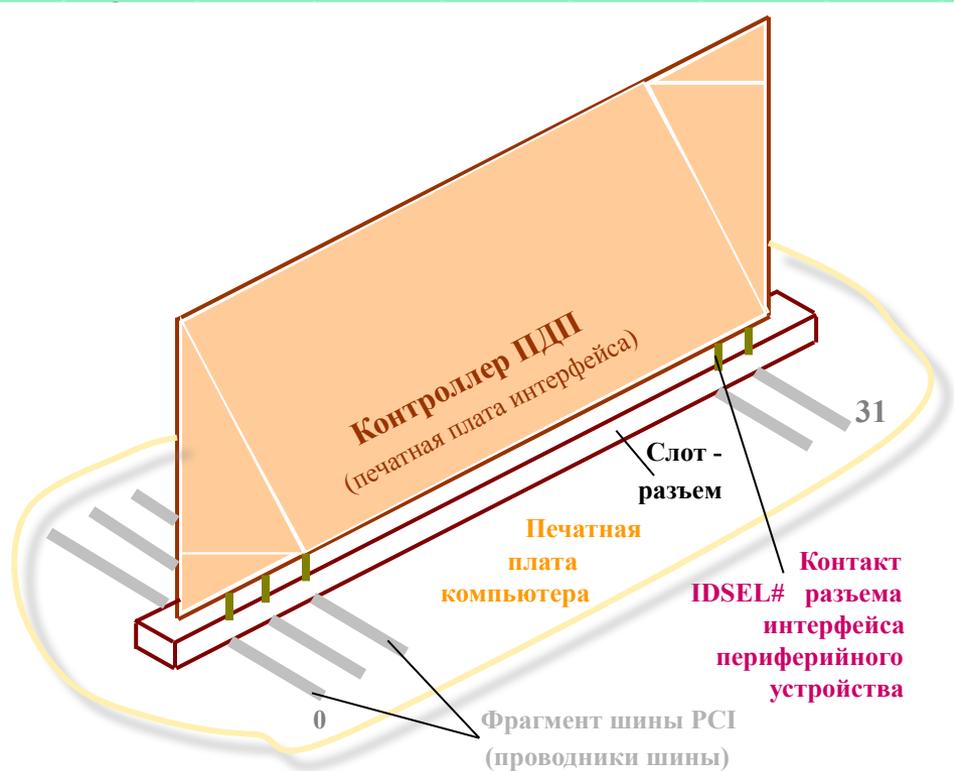


Рис. 4.4. Схема аппаратного подключения интерфейсной платы периферийного устройства ввода-вывода к интерфейсной PCI шине

параметры, как приоритет прерываний устройства. PCI включает четыре линии запроса прерывания. Программное обеспечение компьютера информирует устройство ввода-вывода о том, какие из этих линий оно может использовать для запроса прерывания.

Если устройству необходима инициализация, то установка программных переменных (адресов, счетчиков, переключателей, индикаторов) в нуль или другие специальные значения, предназначенный для этой цели код записывается в специальную ROM его интерфейса. Это не та ROM, которая используется в процессе конфигурирования. Программное обеспечение PCI считывает данный код и выполняет его для инициализации устройства.

Описанная процедура освобождает пользователя от необходимости участия в конфигурационном процессе. Теперь ему необходимо лишь вставить интерфейсную плату в разъем компьютера и включить питание. Все остальное де-

лает программное обеспечение, по окончании работы которого устройство готово к использованию.

В заключение можно отметить, что шина PCI может работать с питающими напряжениями 5 или 3,3 вольта.

Интерфейс SCSI

Интерфейс SCSI был разработан в конце 1970-х гг. организацией Shugart Associates. Первоначальное название SASI (Shugart Associates System Interface), после стандартизации в 1986 году он выпускается уже под именем SCSI (Small Computer Systems Interface - читается «скази», а переводится как интерфейс малых компьютерных систем, проще персональных ЭВМ) и становится одним из промышленных стандартов для подключения периферийных устройств – винчестеров, стримеров (быстрая перезапись данных с диска на магнитную ленту), сменных жестких и магнитооптических дисков, сканеров, CD-ROM и CD-R, DVD-ROM и т.п. К шине SCSI можно подключить до восьми устройств, включая основной контроллер SCSI (или хост-адаптер) (рис.4.5). Контроллер SCSI является по сути самостоятельным процессором и имеет свою собственную BIOS (которая иногда может размещаться в BIOS системной платы). Он выполняет все операции по обслуживанию и управлению шиной SCSI,

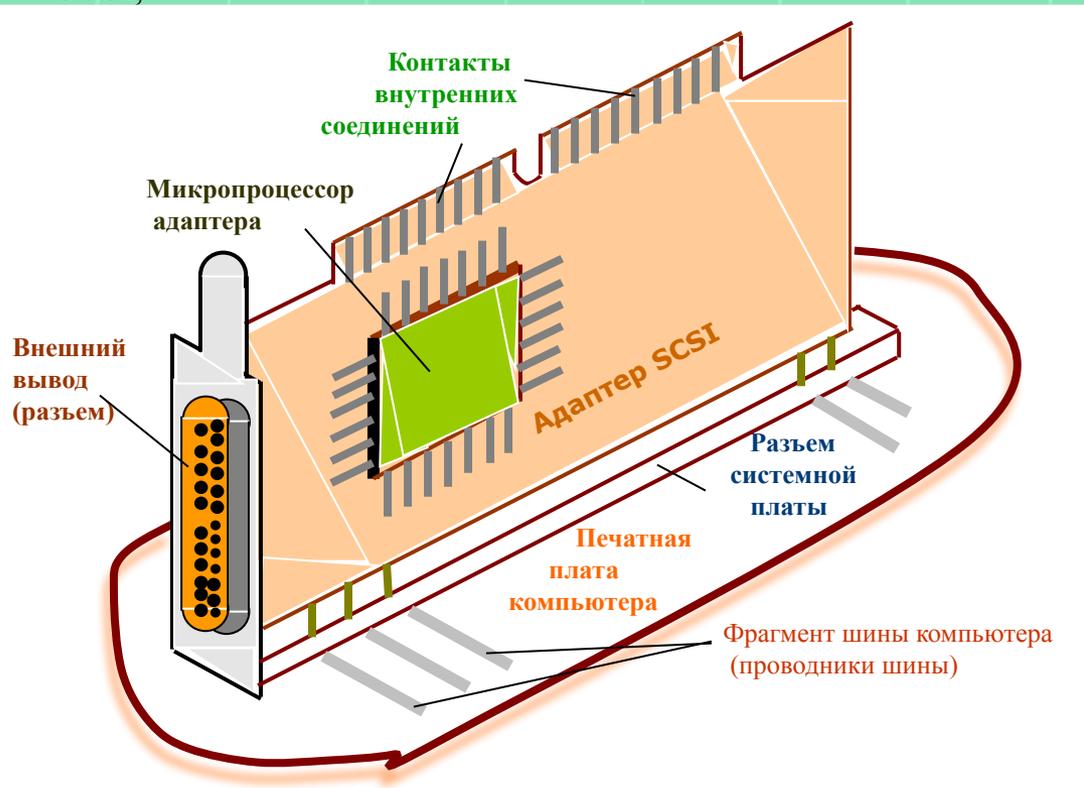


Рис. 4.5. Схема аппаратного подключения периферийного устройства к интерфейсной SCSI шине с помощью адаптера

обходя от этого центральный процессор.

о шины SCSI в компьютерной системы
итически показано на рисунке 4.1. Шина
етствует стандарту, определенному под
ром x3.131, согласно которому, такие
ойства как магнитные диски, должны сое-
ться с компьютером при помощи 50-про-
ого кабеля длиной до 25 м, по которому
ые могут передаваться со скоростью до
айт/с.

тандарт SCSI претерпел много изменений,
ультате которых скорость передачи дан-
постепенно увеличивалась, почти удваива-
ь каждые два года. Шина может иметь
ий данных, сопровождаемых линией кон-
четности и девять управляющих линий –
м случае она называется) и передает дан-
то одному байту за раз.

Широкая шина SCSI (wide SCSI) имеет 16 линий данных и пересылает информацию по 16 бит за раз. Уровень напряжения сигналов принятый в более позднее время составляет 3,3 В и называется Low Voltage Differential (LVD).

Для различных версий шины используются разные разъемы: 50-, 68- и 80-контактные. Максимальная скорость современных периферийных устройств варьируется от 5 до 60 Мбайт/с, а последняя версия шины SCSI поддерживает скорость до 320 Мбайт/с.

Стандарт определяет два способа передачи сигналов: однополярный или ассиметричный (Single ended), и дифференциальный (Differential). В первом случае имеется один провод с нулевым потенциалом («земля»), относительно которого передаются сигналы по линии данных с уровнями сигналов, соответствующими ТТЛ-логике. При дифференциальной передаче сигнала для каждой линии данных выделено два провода, и сигнал на этой линии получается вычитанием потенциалов на их выходах. При этом достигается лучшая помехозащищенность, что позволяет увеличить длину соединяющего устройства кабеля.

Интерфейс SCSI с необходимостью требует наличия «терминаторов» - согласующих (или поглощающих) сопротивлений, которые поглощают сигналы на концах кабелей, тем самым препятствуя появлению эха сигнала.

Максимальная скорость передачи по конкретной шине SCSI зависит от длины кабеля и количества подключенных к нему устройств. Чем короче кабель и чем меньше устройств к нему подключено, тем выше скорость передачи. Для достижения максимальной скорости обычно используется кабель длиной не более 1,6 м при передаче однополярных сигналов (SE) и не более 12 м при передаче дифференциального сигнала.

В отличие от устройств, подключенных к системной шине, являющейся для процессора прозрачной, устройства, подключаемые к шине SCSI, не являются частью адресного пространства процессора. Это адресное пространство шины SCSI. Как ранее было отмечено шина SCSI соединяется с шиной процессора (системной шиной) через SCSI-контроллер, как показано на рисунке 4.1. Для пересылки пакетов данных между оперативной памятью компьютера и устройствами, работающими через шину SCSI, контроллер SCSI применяет технологию прямого доступа к памяти.

Контроллер шины SCSI может быть **инициатором** или **целевым устройством**. Инициатор обладает способностью выбирать конкретное целевое устройство и направлять ему команды, определяющие выполняемую операцию. В организации взаимодействия с диском, очевидно, что инициатором должен выступать контроллер SCSI, а целевым устройством – диск. Инициатор устанавливает **логическое соединение** с выбранным целевым устройством. Соединение может временно прерываться, тогда шина может быть использована для передачи информации другим устройствам. Эта способность чередовать запросы пересылки данных является одной из важнейших особенностей шины SCSI, определяющих ее высокую производительность.

Давайте на конкретном примере взаимодействия процессора с дисковым устройством рассмотрим функционирование шины SCSI. Предположим, что для процессора необходимо прочитать данные с диска в виде нескольких блоков, расположенных непоследовательно в различных секторах диска. Процессор направляет SCSI-контроллеру команду, в ответ на которую выполняются следующие действия.

1. Контроллер SCSI становится **инициатором**, и как инициатор, запрашивает управление шиной SCSI.
2. Выиграв арбитраж, он выбирает целевой контроллер диска и **передает ему управление шиной**.
3. Целевой контроллер начинает операцию вывода, а инициатор дополнительно направляет ему команду, определяющую операцию чтения.
4. Целевой контроллер (контроллер диска), который вначале должен выполнить операцию поиска данных на диске, отправляет инициатору сообщение, указывающее, что он временно разрывает соединение, после чего освобождает шину.
5. Целевой контроллер начинает управление поиском данных на диске, для чего направляет диску команду переместить считывающую головку диска к первому, содержащему запрошенные данные сектору. Затем он считывает данные и сохраняет их в своем буфере данных. Когда контроллер готов начать пересылку данных инициатору, он запрашивает управление шиной SCSI. Выиграв арбитраж, он снова выбирает инициаторский контроллер (SCSI-контроллер), возобновляя тем самым временно разорванное соединение.
6. Целевой контроллер (со стороны диска) пересылает инициатору содержимое буфера данных параллельно по 8 или 16 бит, в зависимости от ширины шины и еще раз разрывает соединение.
7. Вновь целевой контроллер направляет диску команду выполнить следующую операцию поиска данных, на другом секторе диска, которые по прежней технологии пересылает инициатору. Когда пересылка полностью завершается, логическое соединение между двумя контроллерами разрывается.
8. Получив все необходимые данные, SCSI-контроллер сохраняет их (пересылает) в оперативной памяти с использованием технологии прямого доступа к памяти (ПДП).
9. Контроллер SCSI направляет процессору сигнал запроса прерывания, для того чтобы проинформировать его о завершении операции пересылки данных с диска.

Стандартом SCSI определяется множество управляющих сообщений, которыми могут обмениваться взаимодействующие контроллеры для управления различными типами устройств ввода-вывода. Кроме того, им определяются сообщения для обработки различных ошибок и нестандартных ситуаций, которые могут возникнуть в ходе работы устройств и пересылки данных.

Сигналы шины SCSI

В нижеследующей таблице 4.2 приведены сигналы узкой шины SCSI. Для этой шины принято активным (то есть его логическим значением является 1) считать сигнал, имеющий низковольтный уровень (#). Адресные линии и линии данных шины совмещены.

Таблица 4.2. Сигналы шины SCSI

Категория сигнала	Имя	Описание сигнала
Данные	от DB(0)# до DB(7)#	Линии данных: - идентификация устройства на фазе арбитража, выбора и повторного выбора;
	от DB0# до DB7#	- пересылка бита информации по каждой линии на фазе пересылки
Фаза	DB(P)#	Бит четности для шины данных
	BSY# I, T	Действует когда шина не свободна
	SEL# I, T	Действует во время выбора и повторного выбора
Тип информации	C/D# T	Управляющая информация: команда, состояние
	MSG# T	или сообщение
Квитирование	REQ# T	Устанавливается целевым устройством для запроса цикла пересылки данных
	ACK# I	Активизируется инициатором после завершения им операции пересылки данных (ответ на REQ#)
Направление пересылки	I/O# T	Активизируется для операции ввода (с точки зрения инициатора)
Прочее	ATN# I	Внимание! Устанавливается инициатором при необходимости послать сообщение целевому устройству
	RST# I, T	Вызывает отключение от шины всех контроллеров и их переход в исходное состояние

В качестве адресов используются номера – цифры от 0 до 7, присваиваемые подключенным к шине устройствам (контроллерам) и одновременно линиям данных шины. Для адресации инициатор активизирует соответствующую линию данных, устанавливая на ней 1. После установления соединения между двумя контроллерами адресация больше не требуется и линии данных можно использовать для пересылки данных.

Как и в шине PCI, в шине SCSI предполагается возможность обмена информацией между любой парой устройств. Конечно, чаще всего обмен производится между хост-адаптером и периферийными устройствами. Копирование данных между устройствами может производиться без выхода на системную шину компьютера. Здесь большие возможности имеют интеллектуальные хост-адаптеры со встроенной кэш-памятью. В каждом обмене по шине принимает участие его инициатор (Initiator) и целевое устройство (Target).

Арбитраж

Арбитраж – это одна из важных и часто выполняемых частей процесса пересылки данных по шине SCSI. О занятости шины свидетельствует высоковольтное (высокоимпедансное) состояние сигнала BSY# на управляющей ее линии. В это время запрос на владение шиной может поступить от любого подключенного к ней контроллера. Может поступить одновременно несколько обращений, потому и необходим арбитраж. Для того, чтобы запросить управление шиной, контроллер активизирует сигнал BSY# и идентифицирует себя, активизируя соответствующую

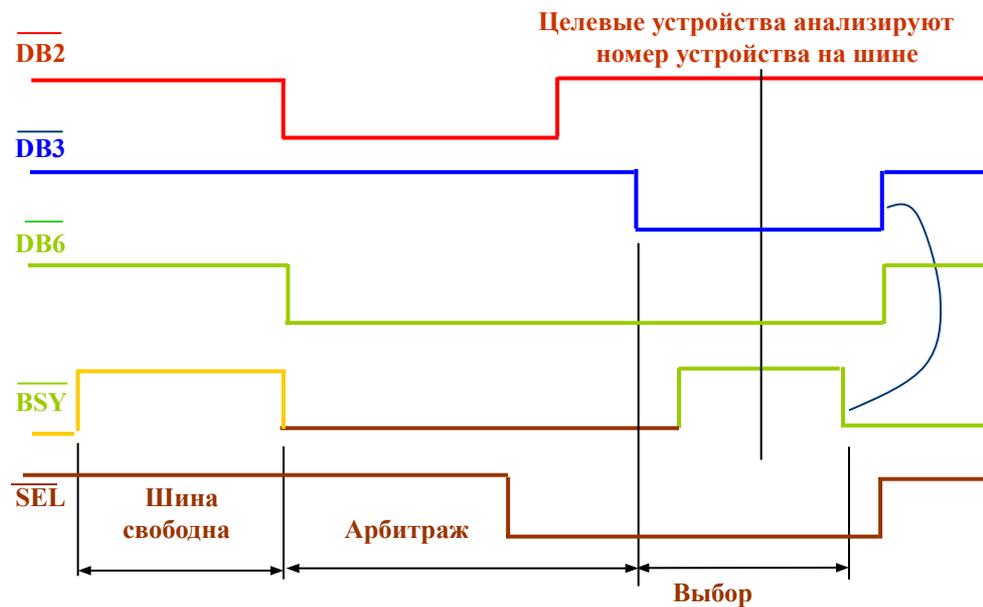


Рис.4.6. Арбитраж и выбор на шине SCSI

линию данных. Шинной SCSI используется такая схема распределенного арбитража, когда контроллеры 2 и 6 запрашивают управление шиной одновременно. Эта схема представлена на рисунке 4.6.

Каждому подключенному к шине контроллеру в соответствии с его номером назначается приоритет. Наивысший приоритет имеет контроллер 7. Когда активизируется сигнал BSY#, каждый контроллер, запросивший управление шиной, анализирует линии данных и определяет, запросил ли управление шиной контроллер с более высоким приоритетом.

Если такового не оказалось, контроллер 7 считает, что он имеет самый высокий приоритет среди претендентов, а значит, он и выиграл арбитраж. Остальные контроллеры отсоединяются от шины и ждут снятия сигнала BSY# (освобождения шины).

В схеме, на рисунке 4.6, предполагается, что контроллер 6 является инициатором, желающим установить соединение с контроллером 5. Выиграв арбитраж у устройства 2, контроллер 6 переходит к фазе выбора и идентифицирует целевой контроллер 5.

Выбор

Выиграв арбитраж, контроллер 6 сохраняет активными сигналы BSY# и DB6 (свой адрес), для того, чтобы сообщить, что он хочет выбрать для соединения контроллер 5. Он активизирует поочередно линии SEL# и DB5. как только активизируется сигнал SEL#, все остальные контроллеры, участвующие в арбитраже (подобно контроллеру 2 на рисунке 4.6), снимают с линии данных свои адреса (если они этого еще не сделали). Поместив на шину адрес целевого контроллера, инициатор освобождает линию BSY#.

Выбранный целевой контроллер отвечает активизацией линии BSY#. Тем самым он сообщает инициатору, что запрошенное соединение установлено и что можно удалять с шины адрес целевого контроллера. На этом процесс выбора завершается, а целевой контроллер (контроллер 5) сохраняет линию BSY# активной. С этого момента шиной управляет целевой контроллер, что необходимо для фазы пересылки информации.

Передача данных

Определение – «передача данных» в текущем контексте означает пересылку информации между инициатором и устройством ввода-вывода или между двумя контроллерами. Информация может представлять собой команду, направляемую инициатором целевому устройству, данные о состоянии целевого устройства, направляемые инициатору или данные, пересылаемые между устройствами ввода-вывода. Для управления пересылкой используется механизм **квитирования**, то есть сигналы REQ# (Request – запрос) и ACK# (Acknowledge – подтвердить (получение)) соответствующие сигналам Master-ready и Slave-ready, о которых говорилось при описании пересылок по системной шине. При выполнении операции ввода (пересылки данных от целевого контроллера к инициатору) целевой контроллер активизирует еще и сигнал I/O# на все время этой операции. Кроме того он может активизировать сигнал C/D#, указывающий, что пересылаются не данные, а команда или информация о состоянии целевого контроллера.

В высокоскоростных шинах применяется тактирование сигналов и данных двумя фронтами – передним и задним, в результате чего удваивается скорость пересылки.

В конце фазы пересылки целевой контроллер снимает сигнал BSY# и тем самым освобождает шину для использования ее другими устройствами. Если данные не переданы в полном объеме, целевой контроллер может вновь запросить доступ к шине SCSI, применяя при этом операцию повторного выбора описанную ниже.

Повторный выбор

Когда логическое соединение временно разорвано и целевой контроллер готов его восстановить, он сначала должен снова получить управление шиной. Для этого он начинает цикл арбитража и, выиграв его, выбирает иницирующий контроллер – точно также, как в свое время иницирующий контроллер выбрал его. Теперь сигнал занятия шины BSY# активизирует выбранный инициатор, сообщающий о возобновлении соединения с целевым контроллером. Далее инициатор должен передать управление шиной целевому контроллеру. Этот процесс заключается в том, что инициатор удерживает на шине сигнал BSY#, чтобы целевой контроллер в свою очередь также активизировал этот сигнал BSY#. Затем инициатор снимает свой сигнал BSY# с шины. В результате соединение возобновлено и целевой контроллер управляет шиной. Можно начинать пересылку очередной «порции» данных.

Описанная сигнальная схема шины определяет механизм установки логического соединения между контроллерами и обмена сообщениями. В любой момент соединение может быть временно прервано и снова восстановлено. Поскольку только целевой контроллер «знает», когда будут готовы данные, а следовательно, когда необходимо временно разорвать и снова восстановить соединение, ему предоставлено право управлять этим процессом.

Шина USB

Современные компьютерные системы включают множество периферийных устройств, таких как клавиатуры, микрофоны, цифровые видеокамеры, динамики, дисплеи и т.д. И почти все они имеют проводное или беспроводное соединение с Интернетом. Для этих систем важно иметь простой и недорогой механизм подключения к компьютеру. Одной из последних разработок в этой области стала универсальная последовательная шина, названная USB (Universal Serial Bus – универсальная серийная шина). Шиной ее можно назвать вероятно потому, что эта, в общем, деревообразная структура позволяет или обеспечивает подключение достаточно многочисленных периферийных устройств.

Изначально шина поддерживает два скоростных режима функционирования: низкоскоростной – 1,5 Мбит/с и полноскоростной – 12 Мбит/с.

В версии USB 2.0 появился высокоскоростной режим -480 Мбит/с.

Разработчики USB ставили перед собой следующие задачи:

- создать простую, дешевую и удобную систему соединения, позволяющую разрешить ограничения, связанные с ограниченным числом портов ввода-вывода в компьютере;
- учесть широкий диапазон параметров пересылаемых данных, присущих различным устройствам ввода-вывода, в том числе модемам;
- облегчить для пользователей процесс подключения периферийных устройств за счет применения режима plug-and-play.

Ограничения по количеству портов ввода-вывода в компьютерах основаны на сугубо практических соображениях. Для подключения нового периферийного устройства пользователь должен был вмешаться в конструктивное оформление компьютера, не имея необходимой для этого квалификации. Кроме того, в процессе подключения и настройки оборудования пользователь должен был ответить на значительное количество вопросов со стороны операционной системы компьютера, не обладая никакими знаниями об этом процессе.

Другая трудность процесса подключения периферийных устройств обусловлена их многообразием и разнообразием.. Разнообразные скорости обмена данными и временные характеристики этого процесса. Например, клавиатура генерирует один символ каждый раз, когда нажимается какая-либо клавиша, а это может произойти в любой момент. Поскольку событие нажатия клавиши ограничено скоростью работы оператора и не синхронизировано ни с каким другим событием в компьютерной системе, генерируемые клавиатурой данные называются **асинхронными**.

Существует множество других устройств, которые генерируют данные такого типа. К их числу относятся мыши и игровые манипуляторы.

Имеются и другого типа источники данных. Многие современные компьютеры снабжены микрофонами – либо встроенными, либо подключенными в качестве внешних устройств. Воспринимаемый микрофоном звук преобразуется в аналоговый электрический сигнал, который перед обработкой компьютером должен быть преобразован в цифровую форму. Для этого аналоговый сигнал **дискретизируется** с помощью аналого-цифрового преобразователя (АЦП), путем измерений через определенные дискретные промежутки времени характеристик звука (например, амплитуды) и преобразования их в соответствующие **n-разрядные** значения. Аналоговый звуковой сигнал, замеряемый в конкретный момент времени, называется **отчетом**. Количество битов, **n**, выбирается исходя из требуемой точности представления звука. Позже, когда эти данные передаются на динамик, обратный цифро-аналоговый преобразователь (ЦАП) восстанавливает аналоговый сигнал из цифрового формата.

В процессе дискретизации генерируется непрерывный поток числовых значений, поступающих через равные промежутки времени. Этот процесс синхронизируется с помощью тактового сигнала. Процесс, последовательные события которого происходят через равные промежутки времени, называется **изохронным**.

Для того, чтобы при оцифровке звука сохранялись те его составляющие, которые меняются с очень большой скоростью, частота дискретизации должна быть очень высокой. Если она составляет s значений в секунду, максимальная частота звука, формируемого в ходе оцифровки, будет $s/2$. Например, человеческая речь адекватно записывается при частоте дискретизации 8 кГц, что позволяет сохранять звуковые частоты до 4 кГц. Для более высоких звуков, которые должны записываться музыкальными системами, требуется большая частота дискретизации. Стандартная частота дискретизации цифрового звука составляет 44,1 кГц. Каждый отсчет передается посредством 4 байт данных, что позволяет представить достаточно широкий диапазон уровней звука (динамический диапазон), необходимый для высококачественного воспроизведения звука. Скорость потока данных составляет при этом 1,4 Мбит/с.

Для процесса оцифровки голоса или музыки важно сохранить точное соответствие частоты дискретизации записи и воспроизведения звука. Значительное расхождение тактовой частоты при записи и воспроизведении звука совершенно недопустимо. Поэтому механизм пересылки данных между компьютером и музыкальной системой должен обеспечить строго одинаковые промежутки времени между отсчетами. В противном случае потребуются сложная схема буферизации и повторного тактирования. С другой стороны, вполне допустимы отдельные ошибки и пропущенные отсчеты. Они либо вообще не замечаются слушателем, либо приводят к возникновению щелчков при воспроизведении записи.

К качеству графических и видеофайлов предъявляются похожие требования, но для их передачи нужна значительно более широкая полоса пропускания канала. Термин «полоса пропускания» означает пропускную способность коммуникационного канала, измеряемую в битах или байтах в секунду. Для воспроизведения видео с качеством коммерческого телевидения составляющие его отдельные изображения должны иметь объем около 16 Кбайт и передаваться со скоростью 30 изображений в секунду, для чего необходима полоса пропускания 44 Мбит/с. Для более качественного видео, такого как HDTV (High Definition TV), требуются более высокие характеристики.

Запоминающие устройства большого объема, в частности жесткие диски и CD-ROM, предъявляют несколько иные требования к коммуникационным интерфейсам. Эти устройства являются частью иерархии памяти компьютера. Их соединение с компьютером должно обладать пропускной способностью порядка 50 Мбит/с. Особенности функционирования диска таковы, что при его работе происходят задержки порядка миллисекунды. Поэтому небольшие дополнительные задержки при пересылке данных между диском и компьютером не имеют значения, равно как и незначительные колебания тактовой частоты.

Технология plug-and-play

Как нам уже известно, технология plug-and-play предполагает, что новое устройство можно подключить в любое время прямо к работающей системе. Система сама должна обнаружить новое устройство, идентифицировать его и соответствующее ему программное обеспечение (драйвер, а также другие, необходимые для его работы средства), назначить ему адреса и установить логические соединения для взаимодействия с другими устройствами. Одной из главных задач разработчиков шины USB была реализация принципа plug-and-play.

Архитектура шины USB

Разнообразие устройств вычислительной техники, а также информационных форм данных потребовали разработку новых технологий для их пересылки. Широкие шины обеспечивают высокие скорости передачи, но, в то же время вносят проблемы, связанные с увеличением фазовых сдвигов сигнала с расстоянием. Для шины USB выбран последовательный формат пересылки данных, обеспечивающий ее меньшую стоимость и наибольшую гибкость. Тактирующий сигнал и данные кодируются вместе и передаются как единый сигнал. В результате нет никаких ограничений в отношении тактовой частоты или расстояний, связанных со сдвигом данных, благодаря чему становится возможной высокая пропускная способность соединений с высокой тактовой частотой.

Для того, чтобы к шине USB можно было одновременно подключать большое количество устройств, удаляемых и подсоединяемых в любое время, эта шина имеет древовидную структуру (рис.4.7). В узлах дерева располагаются устройства, называемые **хабами** и действующие как промежуточные управляющие компоненты между хостом и устройствами ввода-вывода. **Корневой хаб** соединяет все дерево с хост-компьютером. Листьями дерева являются устройства ввода-вывода (клавиатура, динамики, соединение с Интернетом, цифровой телевизор и т.п.), в терминологии USB называемые **функциями**.

Показанная на рисунке древовидная структура позволяет соединять множество устройств с помощью последовательных соединений «точка-точка». Каждый хаб имеет ряд портов, к которым можно подключать любые устройства, в том числе и другие хабы. В нормальном режиме хаб копирует полученное входное сообщение в свои выходные порты. В результате посланное компьютером сообщение передается всем устройствам ввода-вывода, но отвечает на него только адресуемое устройство. В этом отношении USB функционирует подобно шине в ее общем понимании. Однако, в отличие от такой шины, сообщение от устройства ввода-вывода пересылается только вверх, в направлении корневого узла, и другие устройства его не получают. Таким образом, USB дает возможность хосту взаимодействовать с устройствами ввода-вывода, но не позволяет им взаимодействовать друг с другом.

Как древовидная структура соединений USB отвечает задачам этой шины. Дерево позволяет подключать к компьютеру большое количество устройств, используя всего один или несколько портов (корневых хабов). В то же время каждое устройство подключается к компьютеру или хабу с помощью последовательного соединения «точка-точка».

Основой функционирования шины лежит принцип опроса устройств. **Устройство может отослать сообщение только в ответ на запрос хоста.** Поэтому передаваемые хосту сообщения не конфликтуют и не пересекаются друг с другом, и никакие два устройства не могут отослать сообщения одновременно. Здесь работает еще один принцип, связанный со скоростными возможностями шины.

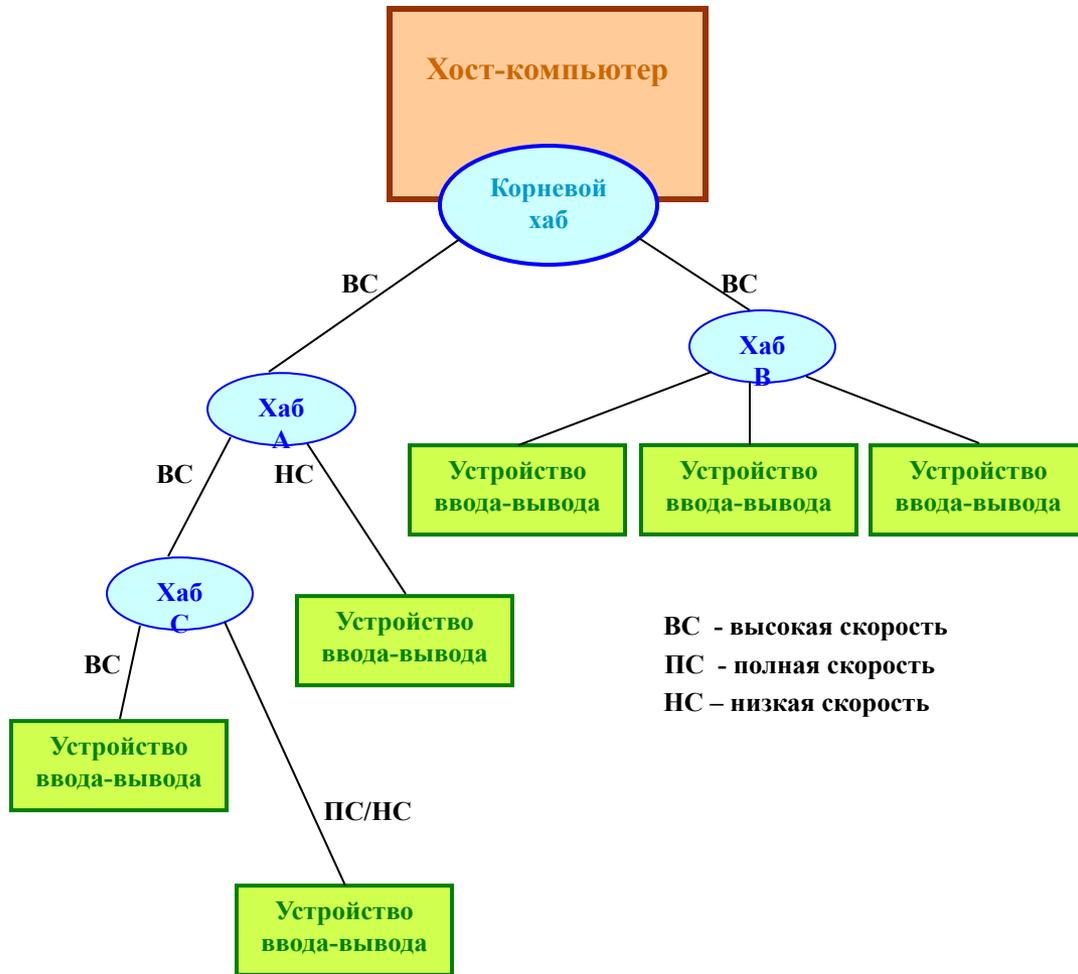


Рис.4.7. Структура дерева USB

А подключен к корневому хабу через высокоскоростное соединение (см. рис. 4.7). Он использует одно высокоскоростное соединение с хабом С и одно низкоскоростное устройство ввода-вывода. Протокол шины USB требует, чтобы сообщения, по высокоскоростным каналам шины передавались только на высокой скорости, даже если они адресованы низкоскоростным устройствам. Поэтому сообщение, предназначенное для низкоскоростного устройства ввода-вывода пересылается от корня хаба к хабу А с высокой скоростью, а хаба А к конкретному низкоскоростному устройству ввода-вывода — на низкой скорости. Последний этап пересылки займет много времени, в течение которого разрешается пропустить высокоскоростной трафик к другим устройствам. Например, хаб А в это время может общаться с подключенными к нему устройствами ввода-вывода.

Фактически, при этом, имеет место разделение шины между высокоскоростными и низкоскоростными участками. В таких случаях, сообщения, следующие по высокоскоростным участкам шины к медленно работающим устройствам ввода-вывода предваряются и завершаются специальными командами (для хаба А) о переходе в режим разделения трафика, а затем выхода из этого режима.

Стандартом USB определяются особенности аппаратной реализации соединений USB, а также структура программного обеспечения хоста и требования к самому обеспечению. Последнее предназначено для поддержки двунаправленных коммуникационных соединений между прикладным программным обеспечением и устройствами ввода-вывода. Эти соединения называют **каналами (pipe)**. Любые данные, входящие в канал с одного конца, обязательно достигнут другого. Уже упоминалось, что программы, обеспечивающие пересылку данных от устройства к устройству, называются **драйвером**. Каждый драйвер разрабатывается с учетом характеристик того устройства, для которого он предназначен. Поэтому канал USB определяется как соединение устройства ввода-вывода с его драйвером. Соединение устанавливается после подключения устройства и назначения ему программным обеспечением USB уникального адреса. При наличии соединения данные могут пересылаться по мере надобности.

Адресация

Корневой хаб USB соединяется с системной шиной компьютера, через которую процессор обращается к хабу как к единственному устройству шины. Заметим, что корневой хаб может быть встроен в компьютер или быть автономным устройством. Программное обеспечение хост-компьютера взаимодействует с отдельными устройствами, подключенными к шине USB, направляя им информацию в форме пакетов, которую корневой хаб перенаправляет по цепочке соединений USB конкретному устройству.

Каждому устройству на шине USB, будь то хаб или устройство ввода-вывода, назначается 7-битовый адрес. Следовательно, общее число устройств, подключенных к шине может быть равно 125. Адрес локален для дерева USB и никак не соотносится с адресами, используемыми на шине компьютера (своя адресная система). Адреса назначаются произвольным образом. Когда устройство активизируется или подключается к хабу, оно имеет адрес 0. Аппаратное обеспечение хаба обнаруживает новое устройство, о чем делает соответствующую пометку в своей информации о состоянии. Периодически хост опрашивает все хабы, собирая сведения об их состоянии, и узнает о добавленных или отключенных устройствах. Когда хост узнает о подключении нового устройства, он с помощью специальной последовательности команд направляет в порт хаба сигнал сброса, считывает из памяти устройства информацию о его типе, назначении и другие характеристики, и направляет ему конфигурационную информацию и присваивает ему уникальный USB-адрес. После этого начинается обычное функционирование устройства, имеющего свой уникальный адрес.

Описанная процедура подключения является основой реализации принципа plug-and-play, осуществляемого полностью программным обеспечением хоста.

При отключении устройства от шины выполняются аналогичные процедуры механизма. Соответствующий хаб сообщает об этом факте программному обеспечению USB, которое обновляет свои таблицы. Если отключенное устройство само является хабом, программное обеспечение логически отсоединяет все подключенные к нему устройства. Программное обеспечение USB должно постоянно иметь полную картину топологии шины и подключенных к нему устройств.

Устройства ввода-вывода и хабы имеют в своем составе многочисленные регистры состояний, буферные, управления, адресные, счетчики и т.д., называемые в компьютерной терминологии **конечными точками**. Они идентифицируются (адресуются) 4-разрядными числами. Точнее сказать, 4-разрядное число адресует пару конечных точек: одну для ввода данных, другую - для вывода (регистр имеет один адрес для адресации либо ввода, либо вывода). Таким образом, у устройства может быть до 16 входных-выходных пар конечных точек. Каждый канал USB, будучи двунаправленным, соединяется с одной такой парой. Один канал, под номером 0, соединенный с конечными точками, существует всегда и создается сразу после включения или перезапуска устройства. Это управляющий канал, который используется программным обеспечением USB в процессе инициализации устройства. В ходе этого процесса создаются другие каналы, количество которых зависит от потребностей и сложности устройства. Четырехразрядный номер конечной точки является частью адресной информации, которую хост отправляет устройствам ввода-вывода.

Протоколы USB. Пересылка информации

Информация, пересылаемая через соединения USB, организуется в пакеты, каждый из которых включает один или несколько байтов данных. Используется 16 типов пакетов, выполняющих различные управляющие функции.

Пересылаемая по шине USB информация делится на две категории: управляющая информация и данные. Управляющие пакеты используются для адресации устройств при инициировании пересылки данных, а также для подтверждения факта получения правильных данных и сообщений об ошибках. Пакеты данных содержат входные и выходные данные, которыми хост обменивается с устройством, и некоторую другую информацию.



Рис.4.8. Формат пакета маркера (токен-пакета) USB

Каждый пакет состоит из одного или нескольких полей, содержащих разные типы информации. Первое поле любого пакета называется идентификатором и обозначается как (Packet Identifier).

Поле PID идентифицирует один из 16 возможных типов пакетов. В этом поле 4 бита информации, которые передаются дважды. Первые четыре бита представляют их реальные значения, а следующие четыре бита – инверсные (по компьютерной терминологии – дополненные – имеется в виду до единицы), как показано на рисунке 4.8. Это позволяет устройству-получателю проверить достоверность полученного байта PID. Некоторые



Рис.4.9. Структура пакета идентификатора

пакеты, и в частности ACK (Acknowledge –подтверждение),

состоят только из одного байта PID. Пакеты, используемые для

управления операциями пересылки данных, называются пакетами

маркера (см.рис.4.8). Пакет маркера начинается с поля PID, в котором содержится одно из двух значений, идентифицирующих тип пакета, - IN или OUT. Пакеты типа IN предназначены для управления операциями ввода, а пакеты типа OUT – операциями вывода. За полем PID следует 7-разрядный адрес устройства и 4-разрядный номер конечной точки внутри этого устройства. Последние пять разрядов в пакете предназначены для его же проверки на наличие ошибок. Для этой цели предназначен метод, называемый циклическим контролем избыточности (Cyclic Redundancy Check, CRC). Биты CRC вычисляются на основе содержимого полей адреса и конечной точки. С помощью обратного вычисления устройство-получатель может определить наличие ошибки в пакете.



Рис.4.10. Формат пакета данных USB

т формат, показанный на рисунке 4.10. За полем иден-

тификатора пакета следуют не более 8192 бита данных, за-

тем 16 контрольных битов. Для идентификации пакетов

используются три разных значения PID, так что пакеты дан-

ных могут иметь номера 0, 1 и 2. Обратите внимание, что пакеты

данных не содержат ни адреса устройства, ни номера конечной точки. Такая информация включается в пакет IN или OUT, инициировавший пересылку данных (в маркер).

С помощью рисунка 4.11 рассмотрим пример операции вывода информации от хост-компьютера. Хост-компьютер отправляет корневому хэбу маркера типа OUT, а за ним пакет данных, содержащий выходную информацию. Поле PID пакета данных идентифицирует его как пакет данных с номером 0. Адресуемый хэб убеждается, что пересылка выполнена без ошибок, проверив контрольные биты пакета, а затем отправляет хосту пакет-подтверждение (ACK). Пакет маркера и пакет данных хэб отправляет далее вниз по дереву. Эти пакеты получают все устройства ввода-вывода, но только одно из них распознает свой адрес в пакете маркера и принимает данные, содержащиеся в следующем пакете, убедившись, что принятая информация не содержит ошибок, и устройство отправляет хосту пакет ACK.

Пакет данных, успешно пересланный на полной или низкой скорости, имеет соответственно номер 0 или 1. Это упрощает процесс восстановления данных после обнаружения ошибок, связанных с пересылкой. Если в результате ошибки, допущенной при пересылке, теряется пакет маркера, пакет данных или пакет-подтверждение, отправитель должен повторно отослать весь набор пакетов. Проверив номер пакета данных в поле PID, получатель может выявить и проигнорировать повторяющиеся пакеты. Высокоскоростные пакеты данных последовательно нумеруются как 0, 1, 2, 0 и т.д.

Операции ввода выполняются аналогичным образом. Хост отправляет пакет маркера IN, содержащий адрес устройства. Этот пакет используется для опроса и дает указание распознавшему свой адрес устройству отправить имеющиеся у него входные данные. Устройство отвечает отправкой пакета данных, а также пакета ACK. Если ус-

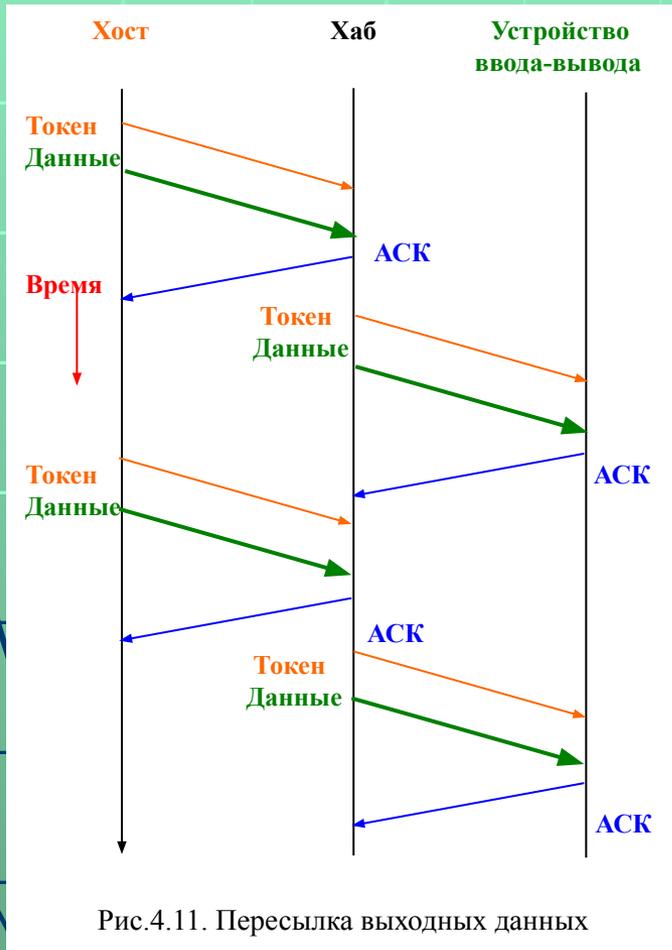


Рис.4.11. Пересылка выходных данных

стройство не содержит готовых данных, оно сообщает об этом, отправляя NAK.

Когда шина содержит низко-, полноскоростные и высокоскоростные пакеты связи, то она может работать в разделенном режиме, чтобы не загромождать сообщения, направляемые по высокоскоростным линиям. В некоторых случаях пакеты IN и OUT, предназначенные для полно- и низкоскоростных устройств, предваряются специальными управляющими пакетами, инициирующими переход в режим разделенного трафика. Следует отметить, что шина USB имеет возможность работать с большим числом протоколов пересылки данных.

Изохронный трафик по шине USB

Одной из ключевых задач USB является поддержка пересылки изохронных данных, например оцифрованного голоса. Устройствам, передающим и получающим изохронные данные, для управления процессом дискретизации и восстановления сигнала необходима тактовая информация. С этой целью передача данных по шине USB осуществляется по кадровому протоколу.

В случае полно- и низкоскоростных данных длительность кадра составляет 1ms. Ровно через каждую миллисекунду корневой хаб генерирует пакет SOF (Start Of Frame – начало кадра), отмечающий начало кадра.

Момент прибытия пакета SOF устройство воспринимает как тактовый сигнал. Для упрощения процесса функционирования устройств, которым требуются более длительные такты, в пакет включается 11-разрядный номер кадра (рисунок 4.12). Вслед за отправкой пакетов SOF хост выполняет пересылку данных изохронным



Рис.7.12. Формат пакета SOF (Start Of Frame) USB

... означает, что каждое из подключенных к шине изохронных устройств сможет каждую миллисекунду получать и отправлять данные.

Основным требованием к изохронному трафику является строгое тактирование. При этом вполне допустимы отдельные случайные ошибки. Это означает, что нет нужды ни в повторной отправке потерянных пакетов, ни в подтверждении факта получения каждого пакета. На рисунке 4.13 показаны два первых фрагмента данных, следующих за пакетом SOF, а именно управляющий пакет с адресом устройства 3 и пакет данных этого же устройства. Данные могут быть входными или выходными, в зависимости от того, относится управляющий пакет к типу IN или OUT. Подтверждающий пакет в этом случае не отправляется. Следующая группа пакетов относится к устройству 7.

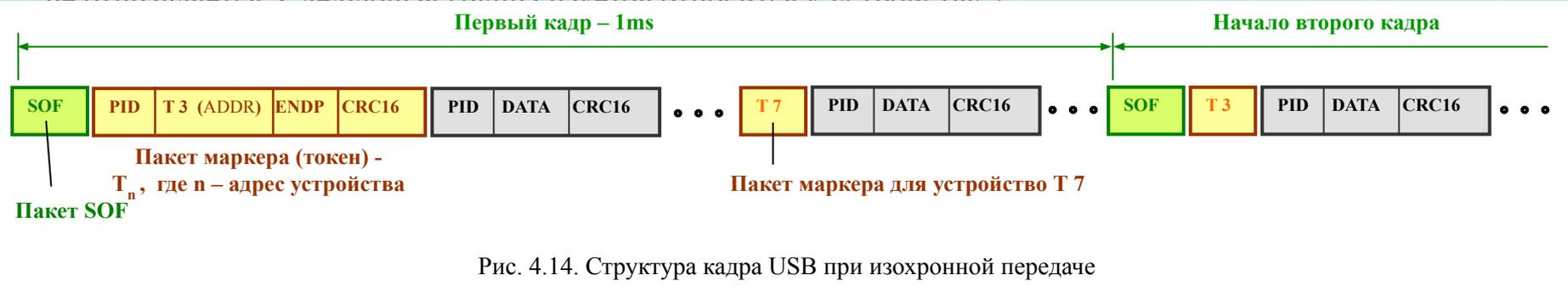


Рис. 4.14. Структура кадра USB при изохронной передаче

Пакет данных для устройства 3, к примеру, может содержать 8 кбит данных. В каждом кадре отправляется один такой пакет, в результате чего создается изохронный канал с пропускной способностью 64 кбит/с. Такой канал может использоваться для голосовой связи. Для пересылки 8 кбит данных, необходим 3-байтовый пакет маркера, за которым следует 1027-байтовый пакет данных (включающий поля PUD и CRC), - итого 1133 байта. При скорости 12 Мбит/с вся пересылка займет около 13 мкс. Очевидно, что в течение оставшегося времени кадра можно обслужить еще несколько таких устройств. После обслуживания всех подключенных к шине изохронных устройств оставшееся время кадра используется для обслуживания асинхронных устройств и обмена

Изохронные данные могут передаваться только по полно- или высокоскоростным линиям шины. В случае высокоскоростных соединений в течение каждого кадра длительность которого составляет 1 мс, пакет SOF повторяется 8 раз, через равные промежутки времени, в результате чего создаются микрокадры длительностью 125 мкс.

Кабели, используемые для USB-соединений, состоят из четырех проводов. Два из них - + 5 В и «земля» предназначены для подачи напряжения питания. Поэтому хабы и устройства ввода-вывода могут иметь собственные источники питания, а могут запитываться непосредственно от шины. Два других провода предназначены для пересылки данных. При высокоскоростной соединении устройств применяется дифференциальная пересылка данных, а в случае низкоскоростных устройств – передача нулей и единиц производится путем установки на одном из двух сигнальных проводов напряжения + 5 В.

Стандартный внешний коммуникационный интерфейс RS-232-C

Обычно персональный компьютер оборудован хотя бы одним последовательным асинхронным адаптером (расположен либо на системной плате, либо в виде сменной печатной платы), по-другому называемым последовательным портом RS-232-C. В операционной системе каждому порту RS-232-C присваивается логическое имя COM1: - COM4: . Интерфейс предназначен для подключения к компьютеру стандартных внешних (периферийных) устройств типа сканера, принтера, мыши и др., а также для связи компьютеров между собой. Данные могут передаваться как в одну, так и в другую сторону – дуплексный режим. Формат данных показан на рисунке 4.15.

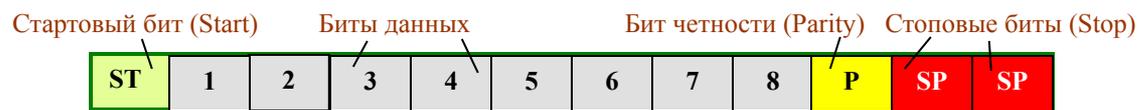


Рис.4.15. Формат кадра при асинхронной передаче

Основу последовательного порта составляет микросхема UART (Universal Asynchronous Receiver-Transmitter – универсальный асинхронный приемопередатчик – Intel 16450/16550/16550A). UART содержит буферные, сдвиговые и другие регистры приемника и передатчика данных. При передаче байта информации он вначале записывается в буферный регистр передатчика, затем в сдвиговый регистр, откуда выдвигается по битам для последовательной передачи по линии связи. Обратный процесс происходит при приеме данных. Разъем для подключения последовательного порта может содержать 25 или 9 выводов, обозначаемых соответственно D25 или D9. Структура разъемов представлена в таблице 4.3.

Таблица 4.3. Структура разъемов интерфейса RS-232-C

D25 Номер контакта	D9 Номер контакта	Содержание информации		Вход или выход
1	-	FG	(Frame ground – Защитное (силовое) заземление)	-
2	3	TD	(Transmitted data – Передаваемые данные)	Выход
3	2	RD	(Received data – Принимаемые данные)	Вход
4	7	RTS	(Request to send – Запрос для передачи)	Выход
5	8	CTS	(Clear to send – Сброс для передачи)	Вход
6	6	DSR	(Data set ready – Готовность данных)	Вход
7	5	SG	(Signal ground – Сигнальное заземление)	-
8	1	DCD	(Data carrier detect – Обнаружение передачи данных)	Вход
20	4	DTR	(Data terminal ready – Данные готовы к передаче)	Выход
22	9	RI	(Ring indicator – Индикатор вызова)	Вход

Таблица 4.4. Сигналы стандарта EIA RS-232-C (рекомендация V24 ССИТТ)

EIA	ССИТТ	Номер контакта ¹	Описание
AA	101	1	Защитное заземление (Protective ground)
AB	102	7	Специальное заземление (Signal ground)
BA	103	2	Передаваемые данные (Transmitted Data)
BB	104	3	Принимаемые данные (Received Data)
CA	105	4	Запрос для передачи Request to send)
CB	106	5	Сброс для передачи (Clear to send)
CC	107	6	Готовность данных (готовность модема) (Data Set Ready)
CD	108.2	20	Готовность выходных данных (Data Terminal Ready)
CE	125	22	Индикатор вызова (Ring Indicator)
CF	109	8	Детектор принимаемого с линии сигнала (Received line signal detector)
CG	110	21	Детектор качества сигнала (Signal quality detector)
CH	111	23 ⁴	Выбор скорости передачи данных (Data signal rate selector) от DTE ² (компьютер) к DCE ³ (модем)
CI	112	23 ⁴	Выбор скорости передачи данных (Data signal rate selector) от DTE ³ (модем) к DCE ² (компьютер)
DA	113	24	Тактирование сигнального элемента передатчика (DTE ²)
DB	114	15	Тактирование сигнального элемента передатчика (DCE ³)
DD	115	17	Тактирование сигнального элемента приемника (DCE ³)
SBA	118	14	Вторичные передаваемые данные
SBB	119	16	Вторичные принимаемые данные

Таблица 4.4. Сигналы стандарта EIA RS-232-C (рекомендация V24 CCITT)

EIA	CCITT	Номер контакта ¹	Описание
SCA	101	19	Вторичный запрос для передачи
SCB	102	13	Вторичный сброс для передачи
SCF	103	12	Вторичный детектор принимаемого с линии сигнала

¹ Контакты 9 и 10 используются для тестирования, а контакты 11, 18 и 25 - запасные

² Терминальное оборудование

³ Коммуникационное оборудование

⁴ Имя сигнала на этом контакте зависит от направления сигнала.

Коммуникационный интерфейс RS-232-C определен стандартом EIA (Electronics Industries Association – Ассоциация электронной промышленности, известный также как рекомендация CCITT (Comite Consultatif Internationale de Telegraphique et Telephonique – Международный консультативный комитет по телеграфии и телефонии, ММЕЕ) под номером V24. С его помощью устанавливается связь между компьютерами и терминальным оборудованием.

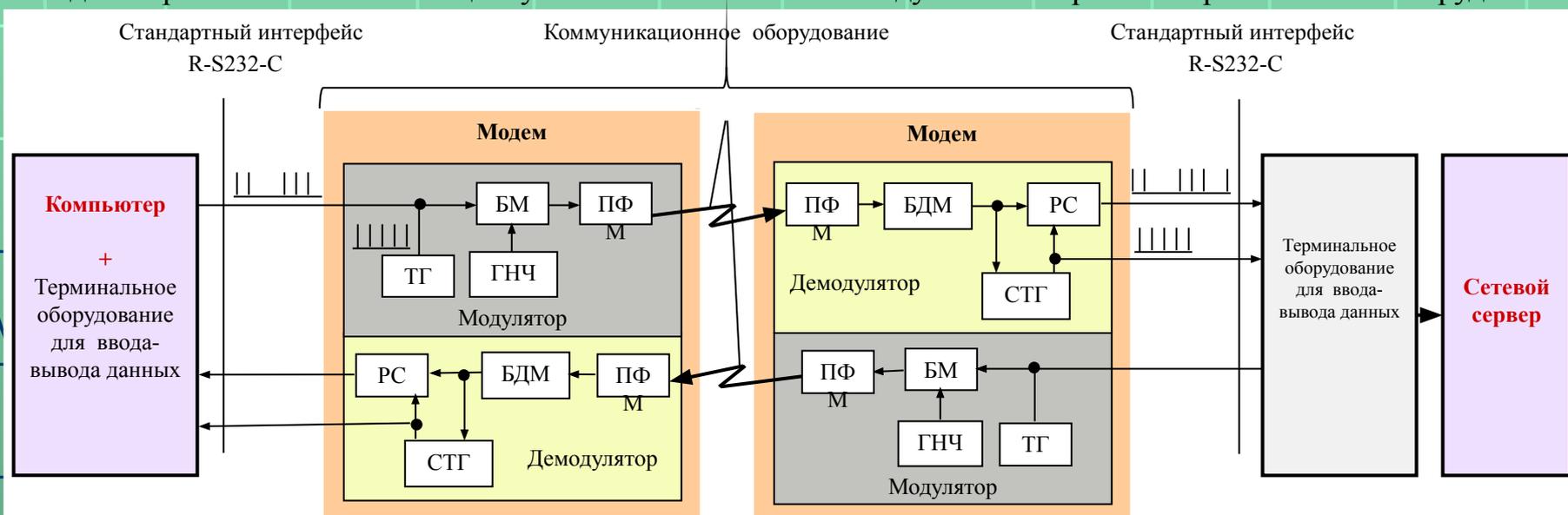


Рис. 4.16. Пример структуры канала с модуляцией, а также основные элементы модулятора и демодулятора

Схема на рисунке 4.16 иллюстрирует пример использования интерфейса RS-232 в канале связи взаимодействующих компьютера и сетевого сервера с через посредство модемов соединенных коммутируемой городской телефонной сетью. Модемы формируют сигналы о начале и завершении сеанса связи, передают тоновые сигналы набора телефонного номера и обнаруживают входящий сигнал вызова. Но, главное, с их помощью реализуется возможность передачи цифровых сигналов в полосе пропускания телефонной сети с использованием частотной модуляции. Для этого выбираются синусоидальные частоты. В выбранном нами примере: 1275 и 1075 Гц – один канал; 2225 и 2025 Гц – второй канал, представляющие логические значения 0 и 1 соответственно.

Ниже, на рисунке 4.17 приведена последовательность шагов и логических сигналов, генерируемых модемами в процессе установления соединения, передачи данных и разъединении.

1. Когда сетевой сервер готов к приему звонка, он устанавливает сигнал готовности выходных данных (CD) в 1
2. Модем В выполняет мониторинг телефонной линии, и обнаружив сигнал входящего звонка, сообщает об этом серверу путем установки сигнала индикатора вызова CE в 1. Если в этот момент CD=1, модем автоматически передает сигнал о снятии трубки. Затем он устанавливает сигнал готовности модема CC в 1
3. Сервер указывает модему В на необходимость начать передачу значения частоты, используемой для представления единицы (2225 Гц), для чего он устанавливает сигнал запроса для передачи CA в 1. Передав указанную частоту, модем В устанавливает сигнал сброса для передачи CB в 1. Определив эту частоту, модем А устанавливает сигнал детектора принимаемого с линии сигнала CF в 1.
4. Компьютер устанавливает CA в 1. Модем А передает сигнал с частотой 1275 Гц и устанавливает CB и CC в 1. Когда модем В определяет, что частота равна 1275 Гц, он устанавливает CF в 1.
5. Теперь между сервером и компьютером установлена дуплексная связь, которая может использоваться для передачи данных в любом направлении. Для этих целей применяются контакты интерфейса ВА (передаваемые данные) и ВВ (получаемые данные); все остальные сигналы интерфейса остаются неизменными.
6. Когда пользователь заканчивает связь, сервер устанавливает сигналы запроса о готовности выходных данных и возможности их передачи, CA и CD, в 0, и в ответ модем В сбрасывает сигнал 2225 Гц и отсоединяется от линии. Сигналы CB, CF и CC устанавливаются модемом В в 0. Когда модем А обнаруживает отсутствие сигнала на линии, он устанавливает сигнал детектора принимаемого с линии сигнала CF в 0.
7. Модем А сбрасывает сигнал 1275 Гц, устанавливает CB и CC в 0 и передает сигнал о завершении связи.
8. Сервер устанавливает сигнал готовности выходных данных CD в 1 и ждет поступления нового звонка.

Применяемая модемами процедура установки соединения предполагает обмен сообщениями, с помощью которых две стороны договариваются о таких параметрах, как схема кодирования, скорость передачи, размер блоков

Применяемая модемами процедура установки соединения предполагает обмен сообщениями, с помощью которых две стороны договариваются о таких параметрах, как схема кодирования, скорость передачи, размер блоков данных и т.д. Интерфейс RS-232 позволяет установить последовательное соединение между любыми двумя цифровыми устройствами.

Шаг	Компьютер	Интерфейс компьютера Сигналы	Модем А	Модем В	Интерфейс сервера. Сигналы	Сервер
1	Набираемые цифры телефонного номера интерфейса сервера			Включение автоматического ответа	CD ← (выходные данные готовы)	1
2				1 → Снимает трубку	CE (идет вызов сервера)	
				1 →	CC (модем готов)	
3	Связь установлена 1 (запрос для переа-чи значения частоты)	CF ←	1 (включение детектора на прием значения частоты 2225 Гц)	← 2225 Гц (передача закончена, сброс сигнала)	CA ←	1 (передача пойдет на частоте 2225 Гц)
4		→ CA CB ← CC ←	1275 Гц (0) 1 (передача законч.)	1 →	CB (детектор принимает с линии сигнал)	
5	Входные данные Выходные данные		ВВ ← Данные ВА 1275(0)/1075(1) Гц	← 2225 (0)/2025(1) Гц Данные →	ВА (данные передаются) ВВ (данные принимаются)	Выходные данные Входные Данные
6	(0) Конец связи	CF ← (отключение детектора модема А)	0	Снимает 2225 Гц и отключается 0 → 0 → 0 →	CA ← CD ← CF CC CB	0 0 Данные Переданы
7		CA CB ← CC ←	Сброс 1275 Гц 0 0			
8					CD ←	1

Интерпретация отдельных сигналов, в частности таких как SA и CD, зависит от функциональных возможностей устройств. Когда эти сигналы не нужны, они просто игнорируются обоими устройствами. В большинстве случаев используется не более 9 сигналов, из приведенных в таблице 4.4.