

# ТРАФИК АСИНХРОННЫХ ПАКЕТОВ ДЛЯ СИСТЕМ СБОРА ДАННЫХ

## Что ожидать от интерфейса IEEE-1394 (FireWire)?

При работе с компьютером неизбежно возникает необходимость переноса данных. Будь то печать текста на принтере, сканирование фотографий или запись CD-диска – все это передача данных. Параллельно с ростом вычислительной мощности требуется увеличение и пропускной способности каналов передачи. Если для обмена данными внутри вычислительного устройства все обстоит более или менее нормально, то при передаче их за пределы компьютера возникают технологические трудности.

Автор рассматривает особенности современных интерфейсов, включая USB. При обсуждении стандарта IEEE-1394 особое внимание уделено элементам, влияющим на понимание причин ограничения по скорости передачи данных. Анализ этой проблемы должен убедить разработчиков аппаратуры, осмелившихся работать со столь непростым стандартом, от многих ошибок.

По мере развития вычислительной техники для объединения внешних устройств с компьютером применялись интерфейсы разной сложности и производительности. Это и давно забытые каналы от IBM, и разнообразные расширители шин, например GPIB (IEEE-488), VME Bus или SCSI, и периферийные шины HP-IL, RS-485, IEEE-1284. Это и простейшие соединители RS-232C или Centronics. Проследим эволюцию интерфейсов по мере развития персональных компьютеров (ПК). Изначально в них применялись наиболее простые (значит – дешевые) интерфейсы RS-232 и Centronics. Затем для обслуживания высокопроизводительных устройств появились интерфейсы стандарта SCSI со своей долгой и непростой эволюцией. Но изделия с этим интерфейсом, обеспечивающим скорости передачи до 160 Мбайт/с, как правило, стоят дорого. Менее дорогое и сравнительно скоростное решение можно реализовать на двунаправленном интерфейсе Centronics. Новый стандарт IEEE-1248 сохранил разъем принтерного порта, но привнес дополнительные режимы.

SCSI, Centronics и IEEE-1248 – это интерфейсы, использующие параллельную шину. Поэтому с увеличением скорости существенно

растет и цена, что тормозит дальнейшее их применение. Чтобы сохранить низкую стоимость, все новые интерфейсы стали создавать на базе скоростных последовательных технологий. В современных спецификациях на ПК объявлено об обязательном использовании стандарта USB. Этот стандарт позволяет объединять устройства в сеть со скоростью тактирования передачи 12 Мбит/с. В качестве конкурента USB успешно выступает стандарт IEEE-1394 (FireWire) с заявленной скоростью обмена до 400 Мбит/с. За право быть лидером между этими стандартами идет непрерывная борьба: если для первого объявлено о переходе на скорость 480 Мбит/с, то для второго – вплоть до 3,2 Гбит/с.

Новые интерфейсы, существенно превосходя RS-232 (115 Кбит/с) и Centronics (500 Кбайт/с), должны с успехом применяться в скоростных системах сбора данных. Но все ли обстоит так благополучно? Действительно ли достижима заявленная скорость? Оказывается, нет. Мы проанализируем один из примеров ограничения скорости для систем сбора данных, построенных на интерфейсе стандарта IEEE-1394. Вскрывая природу ограничений, статья не затрагивает возможные пути разрешения проблемы. Это тема других публикаций. Но сначала – коротко об интерфейсах, применяемых сегодня в ПК.

### ИНТЕРФЕЙСЫ ПЕРЕДАЧИ ДАННЫХ В ПК

Параллельный интерфейс Centronics разрабатывался для передачи данных из компьютера на печатающее устройство. Обратная пересылка данных не предусматривалась, но была возможна через линии статуса и управления. Производители микросхем для Centronics сначала самостоятельно обеспечили возможность равноправной передачи данных в обоих направлениях, а затем эти изменения были закреплены документально, для ПК – в стандарте IEEE-1284. Кроме стандартного режима, в IEEE-1284 предусмотрена двунаправленная передача (режим EPP) и сетевая пересылка (режим ECP). Последний режим обеспечивает полноценное определение типа устройства в процессе идентификации Plug and Play (PnP). Хотя теоретическая

### Представляем автора статьи

**ПЕТРОВ Виктор Игоревич.** Инженер-электронщик и программист. Закончил Московский физико-технический институт. Работает в АО "Инструментальные системы". Проектированием сложных электронных изделий занимается 20 лет, семь из них – коммутацией и передачей речи и информации.

E-mail: insys@orc.ru



скорость передачи данных в режиме ECP может достигать 2 Мбайт/с, многие производители компьютерных платформ сознательно ограничивают ее на уровне 500 Кбайт/с. Поскольку расстояние между компьютером и устройством ограничено двумя метрами, разъемные соединения неудобны и, что наиболее существенно, дальнейшее увеличение скорости не предусматривается, сегодня этот интерфейс в задачах передачи данных используют крайне редко.

Основным интерфейсом передачи данных на ПК был последовательный **RS-232**. Для многих он остается таким и сейчас. По мнению разработчиков программного обеспечения, его главное достоинство – простота. Эта простота сыграла и свою отрицательную роль. Многие разработчики после первых успехов передачи данных по RS-232 полагали, что проблемы уже решены. Однако они только начинались! Контроллер RS-232 обеспечивает лишь базовый (физический) уровень обслуживания связи, для которого почти не рассматриваются аспекты надежности и достоверности пересылки. А потери и искажения возникали в самый неподходящий момент. Видимо, причины неудач кроются не в особенностях RS-232, а в переоценке разработчиками своих возможностей. Они, несмотря на наличие детально проработанных стандартов и рекомендаций по организации пересылки данных по каналам связи, создавали собственные протоколы, зачастую существенно снижающие скорость. Некоторые разработчики по-прежнему хотят работать непосредственно с регистрами контроллеров, а не пользоваться стандартными элементами операционных систем (например, API и драйверами Windows). Сказанное, к сожалению, применимо и для других интерфейсов. Что касается RS-232, то на его основе можно строить даже сети. Однако его скорость – до 115 Кбит/с одновременно в обоих направлениях – падает с увеличением длины кабеля. Кроме того, не менее 20% времени занимает передача служебной информации.

В январе 1996 года была опубликована спецификация шины **USB** (Universal Serial Bus – универсальная последовательная шина) [1, 2], которой отводилась роль единого универсального интерфейса. С 1998 года она поддерживается операционными системами Windows и включена как обязательный элемент для компьютеров класса PC. В соответствии с ее концепцией, на ПК начала нового тысячелетия доступ ко всем стандартным внешним устройствам (клавиатуре, мыши, блоку настройки монитора, источникам бесперебойного питания и др.) будет возможен лишь через шину USB. Для каждого класса устройств операционной системе достаточно одного драйвера, не зависящего от реализации конкретного образца. Конечно, этой концепции могут навредить несогласованные действия разработчиков, но поглядим – увидим.

Шина USB позволяет адресовать до 127 устройств, хотя вряд ли такое количество понадобится. Устройства подключаются к центральному компьютеру, используя топологию дерева. В его корне всегда стоит активный контроллер компьютера. Далее через специальные точки ветвления (устройства со специальными функциями, HUB) “дерево” получает свои “ветки”, на концах которых находятся устройства. Такая топология исключает использование второго компьютера в сети, поэтому два компьютера нельзя объединить через интерфейс USB простым соединением кабелей.

Компьютер и устройство обмениваются пакетами в полудуплексном режиме. Передача тактируется частотой 12 МГц. Именно тактируется, поскольку скорость обмена данными ниже. Пакеты содержат заголовки, проверочные суммы, стартовые и стоповые последовательности. Между пакетами предусмотрены зазоры, обмен сопровождается специальными запросами и подтверждениями. Все ресурсы шины распределяет центральный компьютер в пределах кадра длительностью 1 мс. Кадры циклически повторяются.

В процессе обмена все устройства “слушают”, а центральный компьютер инициирует любую передачу в заданном направлении, монополюсно посылая запросы в нужное устройство. За запросами могут идти данные. Все пакеты (не только данных) защищены проверочной суммой и сопровождаются подтверждениями, по которым можно выявить ошибки и вовремя повторить передачу пакета. В стандарте оговорено четыре типа обмена данными. Групповые обмены (Bulk) обеспечивают надежную и скоростную передачу; транзакция прерывания (Interrupt) позволяет передать данные по мере их готовности; командные обмены (Control) предназначены для управления устройством; обмены в изохронных потоках (Isochronous) служат для передачи привязанных ко времени данных, без гарантии достоверности.

Ни в пакетах запроса, ни в пакетах данных размер данных не указывается. Он определяется из конфигурации устройства, которая считывается компьютером сразу после назначения устройству адреса. Поэтому работа шины USB невозможна без управляющего компьютера, и на него возлагается обязанность администратора шины, которая требует соответствующей производительности и аккуратности функционирования. Пока же аккуратность корневых драйверов шины USB, поставляемых с операционными системами Windows, недостаточна.

Существует несколько совместимых версий стандарта USB. Мы кратко рассмотрели USB 1.1. В дополнение к ней версия USB 2 позволяет тактировать передачу данных частотами 1,5; 12 и 480 МГц. В ней предусмотрена параллельная работа нескольких шин стандарта USB 1.1, что позволит компьютеру использовать одновременно большее число высокопроизводительных устройств без их переделки. Справится ли операционная система?

Контроллер шины USB поддерживается драйверами не во всех операционных системах. Для ПК класса PC применение USB возможно в первую очередь с ОС Windows 98/Me/2000, а в последнее время – и с ОС Linux. Шину USB удобно использовать в задачах передачи данных. Отметим, однако, что эти задачи трудно разрешимы без программистов соответствующей квалификации. В целом, USB – это весьма перспективный стандарт, особенно с появлением версии USB 2, которая теоретически обеспечит наибольшую скорость обмена без потерь из всех современных интерфейсов для ПК.

### НЕМНОГО ОБ ИНТЕРФЕЙСЕ IEEE-1394

Еще один высокоскоростной интерфейс – FireWire – был применен не так давно фирмой Apple в своих персональных компьютерах. Переименованный затем комитетом стандартизации Института инженеров электротехники и электроники в интерфейс IEEE-1394 [3], он получил широкую известность. В 1995 году была опубликована рассмотренная ниже первая версия стандарта IEEE Std 1394-1995. С 1999 года известно расширение IEEE-1394a [4], готовится к изданию редакция 1394b.

Тактовая частота шины стандарта IEEE-1394 может быть 98,304; 196,608 и 393,216 Мбит/с, что соответствует номиналам 100, 200 и 400 Мбит/с. Подключаемое к шине устройство может иметь любую максимальную скорость из этого набора, но обязано поддерживать и более низкие скорости.

Стандартный кабель длиной до 4,5 метров содержит две витые пары сигнальных проводов, по которым осуществляется обмен в полудуплексном режиме. Для простоты можно было бы считать, что одна пара проводов используется для передачи данных, а другая – для их тактирования, однако сигналы имеют более сложную природу. Дополнительно в кабеле может быть два провода для подачи

питающего напряжения (8–40 В, до 1,5 А; в ПК обычно 12 В). Конструкция кабеля такова, что порядок и направление подключения не имеют значения.

Каждое устройство является узлом сети (Node) и может содержать несколько равноправных разъемов (портов) для подсоединения кабеля. Между портами сигналы передаются через повторители, которые в нужный момент изменяют направление передачи или вообще блокируют сигнал. Такое соединение, хотя и вносит дополнительную задержку в распространение сигнала, гарантирует устойчивость работы, исключает петли и крайне полезно при арбитраже. Одна сеть может содержать до 63 узлов, но независимые сети могут объединяться (всего 1023 сети). Используя уникальный адрес сети и узла, устройства поочередно обмениваются пакетами данных. Для работы сети не нужен центральный компьютер – любое устройство может самостоятельно осуществлять обмен.

Первоначально все узлы сети равноправны, а ее топология не определена. Любое изменение в сети переводит шину в состояние Reset. Затем следует идентификация, которая начинается с процесса определения топологии (Tree-Identification) и заканчивается процессом самоидентификации (Self-Identification). В результате сеть приобретает топологию дерева, а узлы получают уникальные адреса. Если какой-нибудь узел не удовлетворен полученными параметрами сети, он может попытаться изменить их, повторив процесс инициализации. В течение одной секунды, пока длится инициализация и идентификация, передача данных невозможна. Поэтому нельзя подключить к сети новое устройство, не прервав передачу. Отметим, что с USB это возможно и без разрыва активных соединений.

Согласованием работы узлов занимаются специальные менеджеры. Так, за администрирование шины IEEE-1394 отвечает менеджер шины (Bus Manager), следящий за ее ресурсами и оптимальностью работы, в том числе и за синхронизацией изохронных циклов. Менеджер питания (Power Manager) учитывает потребителей питания с проводов шины. Менеджер изохронных пакетов (Isochronous Manager) регулирует трафик изохронных пересылок. Любому менеджеру требуется только при необходимости и может быть реализован на отдельном узле – как правило, на подключенном к сети компьютере. Число компьютеров в сети, как независимых, так и действующих согласованно, не ограничено.

В стандарте IEEE-1394 предусмотрена многоуровневая иерархия процедур. На физическом уровне (Physical Layer) выполняются действия, связанные с идентификацией, арбитражем и кодированием сигналов. На уровне соединения (Link Layer) контролируется корректность оформления пакетов и подтверждений. На уровне пересылок (Transaction Layer) обеспечивается правильный порядок пересылки пакетов. Средства же уровня управления (Management Layer) реализуют функциональность и обслуживают пакеты, предназначенные менеджерам. К счастью, производители контроллеров шины IEEE-1394 предусмотрели не только аппаратную реализацию ряда уровней (физического, соединения и частично пересылок), но и автоматизировали их взаимодействие, что существенно упрощает программирование (см. приложение “Железо для “огненной проволоки”, с.22).

### ПЕРЕДАЧА ДАННЫХ ПО ШИНЕ IEEE-1394

В стандарте IEEE-1394 описаны пакеты асинхронной и изохронной передачи [5]. В дополнении к стандарту (P1394a) добавлены потоковые пакеты (Stream), которые в отличие от изохронных пакетов передаются вне цикла шины совместно с асинхронными пакетами [4]. Работа с шиной IEEE-1394 основывается на “честном” (FAIR) поведении узлов сети. Это означает, что все узлы должны быть ис-

правны, не перехватывать и не отвечать на чужие пакеты и занимать шину только в установленное для этого время. При нарушении данных правил сеть может потерять работоспособность.

Изохронный цикл длится 125 мкс и начинается со специального 20-байтного пакета – метки начала цикла. Устройство, желающее передать данные в изохронном пакете, резервирует канал и полосу пропускания (в сумме не более 80% времени цикла). Изохронные пакеты должны следовать один за другим в соответствии с номером канала, сразу после метки начала цикла. От устройства требуется, чтобы размер его пакетов не превысил зарезервированную полосу. Данные, содержащиеся в изохронных пакетах, могут временно быть принятыми несколькими устройствами, но они передаются, даже если никому не нужны. Объем передаваемых данных в одном изохронном пакете достигает 4096 байт (на скорости 400 Мбит/с). Заголовок с уникальным номером канала и проверочные суммы занимают 12 байт, поэтому эффективность изохронных пакетов весьма высока.

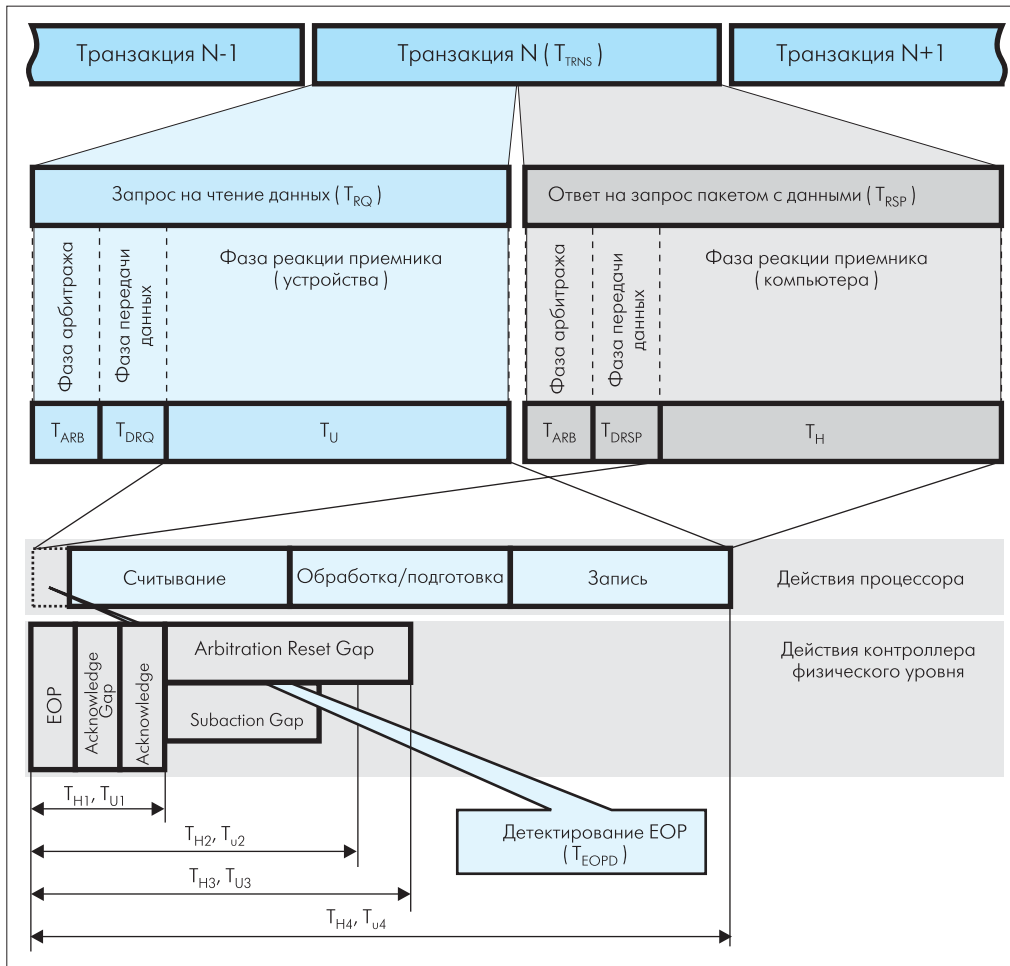
Изохронные пакеты удобны для передачи привязанных ко времени данных – таких, как аудио и видео, – для которых незначительные потери несущественны. Однако при обмене, где требуется высокая достоверность, лучше использовать другой тип пакетов – асинхронные.

Цикл при асинхронном обмене имеет произвольную, постоянно изменяющуюся длительность, которая зависит от числа желающих одновременно передать пакет. Устройства пытаются передать свои пакеты по очереди в соответствии с приоритетом (определяется топологией сети). Узел, уже передавший пакет, может осуществить повторную передачу, только если в течение некоторого интервала (интервал сброса арбитража, Arbitration Reset Gap) никто не захватил шину, т.е. только в следующем цикле (согласно “честному” протоколу). Минимально возможный зазор между пакетами в рамках одного цикла – Subaction Gap. Как правило, на любой пакет запроса должен прийти пакет ответа. Запросы бывают трех видов – чтения (Read), записи (Write) и комбинированная операция (Lock).

В заголовке асинхронного пакета передается адрес источника, адрес приемника, тип операции, размер пересылаемых данных и глобальный адрес в адресном пространстве устройства. В результате размер заголовка вместе с проверочной суммой может достигать 20–24 байт. В то же время максимальный объем данных на скорости 400 Мбит/с – 2048 байт (для скоростей 200 и 100 Мбит/с – 1024 и 512 бит, соответственно).

Недостаток таких пакетов – крупные заголовки. Если учесть, что весь цикл чтения состоит из запросов и ответов со своими заголовками, то на одну пересылку приходится 48 байт служебной информации. Даже при размере поля данных пакета в 32 байта эффективная скорость более чем в два раза отличается от максимальной. Эффективность обмена возрастает при увеличении размера пакета, но он ограничен как стандартом, так и параметрами приемника и передатчика. Кроме того, большие пакеты требуют больше времени на повторную пересылку, если были приняты с ошибками. Они увеличивают время ожидания для других узлов и могут значительно сместить метку начала изохронного цикла.

Для управления и контроля подходят только асинхронные пакеты. Они, в отличие от изохронных, обязательны для нормальной работы узла. Поэтому для передачи данных разумно использовать именно их. При этом естественно было бы ожидать скоростей передачи по крайней мере того же порядка, что и быстроедействие шины. Но на практике скорость обмена оказывается в десятки раз ниже. Чтобы понять причину такого торможения, рассмотрим процесс обмена асинхронными пакетами более детально.



**Рис. 1. Временная модель передачи данных последовательными асинхронными пакетами**

### МОДЕЛЬ АСИНХРОННОГО ОБМЕНА

Рассмотрим ситуацию, когда компьютер принимает непрерывный поток данных из устройства, а других узлов в сети нет. Данные передаются в асинхронных пакетах, другие пакеты отсутствуют. Считаем, что передача и прием происходят без ошибок. Определим эффективную скорость передачи данных.

Непрерывный обмен состоит из последовательности транзакций фиксированного размера, где конец текущей транзакции совпадает с началом следующей (рис. 1). Компьютер отправляет пакет запроса “чтение” в устройство, которое отвечает пакетом с данными. Запрос и ответ передаются последовательно в пределах одной транзакции.

Обмен обеспечивают два параллельно идущих процесса – физические действия на шине IEEE-1394 (контроллер) и работа по обслуживанию данных процессором (программа firmware). Передача пакета по шине включает:

- захват шины (фаза арбитража);
- собственно пересылку пакета (фаза передачи данных);
- отправку приемником подтверждения (Acknowledge)
- ожидание временного интервала между пакетами.

В это же время процессор принимает данные из контроллера, обрабатывает их (например, записывает на жесткий диск) и передает новые данные в тот же контроллер.

Фаза арбитража нужна, когда несколько узлов пытаются захватить шину. В нашем случае устройство одно, но время на арбитраж  $T_{ARB}$  все равно требуется. Длительность фазы передачи данных зависит от объема данных и темпа их тактирования. В запросе на чтение при-

сутствует только заголовок (16 байт) и проверочная сумма (4 байта). Ответ на запрос, кроме заголовка с проверочной суммой (24 байта), содержит данные со своей проверочной суммой из 4 байт. Число байт в поле данных пакета всегда кратно четырем.

Передача данных всегда завершается специальным состоянием на шине – маркером конца пакета EOP. Контроллер детектирует EOP и выставляет процессору запрос на прерывание. Время обработки процессором этого прерывания  $T_{EOPD}$  может составить несколько микросекунд, но в нашем случае мы считаем его пренебрежимо малым. С этого момента данные можно считать принятыми и процессор начинает их обработку. Но действия на шине еще не завершены. Контроллер приемника выжидает зазор Acknowledge Gap и отправляет передатчику 8-битное подтверждение Acknowledge.

Сразу же после передачи подтверждения контроллеры всех устройств начинают отсчет времени. По истечении зазора Subaction Gap устройство может от-

ветить пакетом с данными, если их уже подготовил процессор. Если время подготовки данных превысило  $T_{SUBACTION}$ , передача пакета может начаться не ранее чем через интервал Arbitration Reset Gap (см. рис. 1). Длительности  $T_{SUBACTION}$  и  $T_{ARBRESET}$  зависят только от параметра GapCounter ( $N_{GC}$ ), одинакового для всех узлов сети. Этот параметр в интервале от 0 до 63 устанавливает Bus Manager в зависимости от длины и задержек сети. В стандарте они задаются как

$$T_{SUBACTION} = (27 + 16N_{GC})T_{S1}; T_{ARBRESET} = (51 + 32N_{GC})T_{S1},$$

где  $T_{S1}$  – время передачи одного бита данных на скорости 100 Мбит/с.

На рис. 1 изображен случай, когда время реакции процессора  $T_{U4}$  ( $T_{H4}$ ) превышает время установления готовности контроллера  $T_{U3}$  ( $T_{H3}$ ). В общем случае, стандарт IEEE-1394 определяет время реакции приемника запросов как

$$T_U = \begin{cases} \max(T_{U3}, T_{U4}), & \text{если } T_{U4} > T_{U3} \\ T_{U3}, & \text{если } T_{U3} = T_{U4} \\ T_{U1}, & \text{если } T_{U1} = T_{U4} \end{cases} \quad (1)$$

Время реакции приемника ответов

$$T_H = \max(T_{H3}, T_{H4}). \quad (2)$$

В этих формулах  $T_{U1} = T_{H1} = T_{EOP} + T_{ACK}$ ;

$T_{U2} = T_{H2} = T_{EOP} + T_{ACK} + T_{SUBACTION}$ ;

$T_{U3} = T_{H3} = T_{EOP} + T_{ACK} + T_{ARBRESET}$  (см. рис. 1).

Значения параметров для расчета скорости приведены в табл. 1. Очевидно, что фактическая скорость передачи данных  $F = 32N_{QD} / T_{TRNS}$  (бит/с), где  $T_{TRNS}$  – время, требуемое на выполне-

Таблица 1. Параметры шины IEEE-1394

Обозначение	Значение	Параметр
$N_{GC}$	От 0 до 63	Текущая величина GapCounter
$T_{SPD}$	$T_{S1} = 10,2$ нс $T_{S2} = 5,1$ нс $T_{S4} = 2,5$ нс	Время передачи одного бита данных на скоростях 100, 200 и 400 Мбит/с
$N_{QD}$	Не более 512	Число передаваемых слов за одну транзакцию (в одном слове 4 байта)
$T_{ARB}$	Не более 180 нс	Паспортное время арбитража
$T_{EOP}$	Не более 260 нс	Время генерации конца пакета
$T_{ACK}$	Не более 170 нс	Время подтверждения приема пакета

ние одной транзакции. Это время складывается из времени выполнения запроса ( $T_{RQ}$ ) и времени ответа ( $T_{RSP}$ ):  $T_{TRNS} = T_{RQ} + T_{RSP}$ .

В свою очередь,  $T_{RQ} = T_{ARB} + T_U + T_{DRQ}$  и  $T_{RSP} = T_{ARB} + T_H + T_{DRSP}$ . Раскрывая времена передачи данных  $T_{DRQ}$  и  $T_{DRSP}$  через число передаваемых слов и скорость передачи, получим

$$T_{RQ} = T_{ARB} + T_U + 32 \cdot 5 \cdot T_{SPD}$$

$T_{RSP} = T_{ARB} + T_H + 32 \cdot T_{SPD} \cdot (6 + N_{QD})$ . Времена отклика устройства  $T_U$  и компьютера  $T_H$  задают формулы (1) и (2).

Времена реакции процессоров устройства и компьютера  $T_{Hд}$  и  $T_{Hц}$  зависят от типа процессоров и алгоритмов. Их значения были получены экспериментальным путем.

**ЭКСПЕРИМЕНТ**

Величины  $T_{Hд}$  и  $T_{Hц}$  определялись в ходе эксперимента на оборудовании фирмы "Инструментальные системы" ([www.insys.ru](http://www.insys.ru)). Задающим узлом выступал компьютер на базе материнской платы ASUS-P3B-1394 производства фирмы Asustec со встроенным контроллером IEEE-1394 TSB12LV22 (Texas Instruments) и с процессором Pentium III – 400 МГц (тактовая частота шины процессора 100 МГц, PCI – 33 МГц). Источником данных служило одноплатное устройство сбора ADP6x01 ("Инструментальные системы") с процессором TMS320c6701 (цикл 6,7 нс) и с контроллером IEEE-1394 TSB12LV31, обеспечивающим обмен на скоростях до 200 Мбит/с. Полученные значения  $T_{Hд}$  и  $T_{Hц}$  приведены в табл.2.

Таблица 2. Время реакции компьютера TH и устройства TU в зависимости от размеров поля данных в пакете

Параметр	Поле данных в пакете, слов (x4 байт)									
	1	2	4	8	16	32	64	128	256	512*
$T_{Hд}$ , нс	12925	13350	14325	16150	19850	27350	41315	71095	130657	249779
$T_{Hц}$ , нс	30350	30750	32375	33500	37725	44400	58553	87559	145572	261598

\* Измерение выполнено в обход ограничения стандарта IEEE-1394 на максимальный объем данных для скорости 200 Мбит/с.

При выборе оптимального режима передачи данных можно варьировать скорость передачи данных (параметр  $T_{SPD}$ ), размер поля данных в пакете ( $N_{QD}$ ), величину GapCounter ( $N_{GC}$ ) и время отклика процессоров ( $T_{Hд}$  и  $T_{Hц}$ ). Все остальные параметры – константы.

Анализируя результаты, мы наблюдаем практически неразличимую зависимость скорости передачи данных от величины GapCounter, а значит и от протяженности шины и количества узлов. Чтобы обнаружить эту зависимость, скорость ПК должна быть существенно выше, чем у существующих сегодня. Такая зависимость для гипотетического компьютера приведена на рис.2. Влияние скорости шины заметнее, но не определяющее. Из рис.3 следует, что, увеличивая размер пакета, можно существенно увеличить скорость передачи данных.

**Но самый существенный ограничитель скорости – быстроедействие компьютера и устройства** (рис.4). Это означает, что при выборе аппаратуры и способов передачи в системах сбора

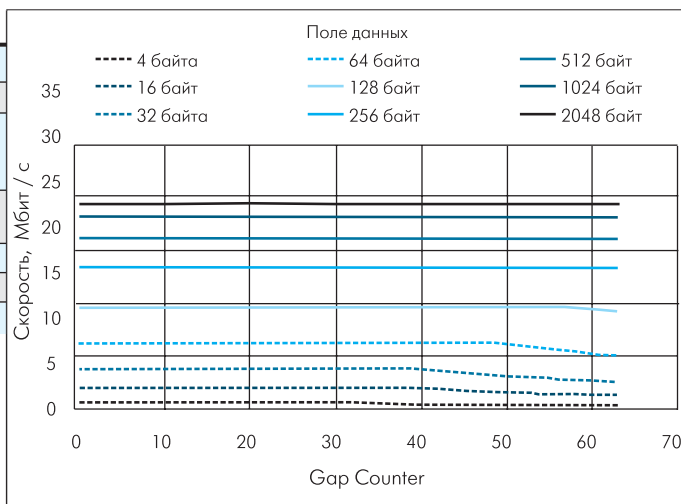


Рис.2. Зависимость скорости передачи данных от величины GapCounter для гипотетического компьютера с очень высокой производительностью

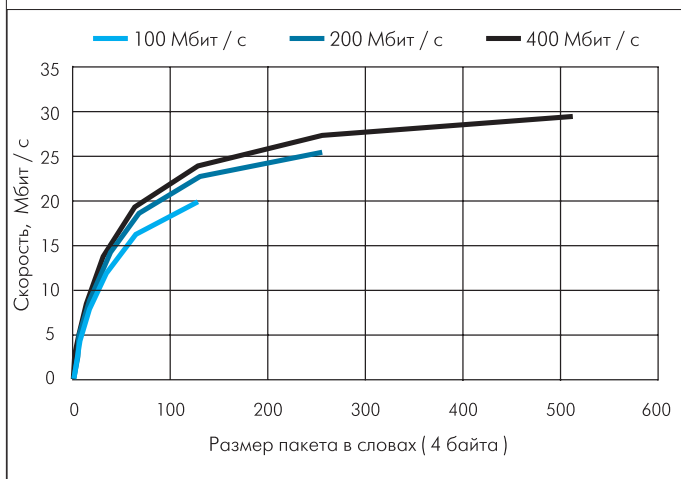


Рис.3. Зависимость скорости передачи данных от размера поля данных в асинхронном пакете

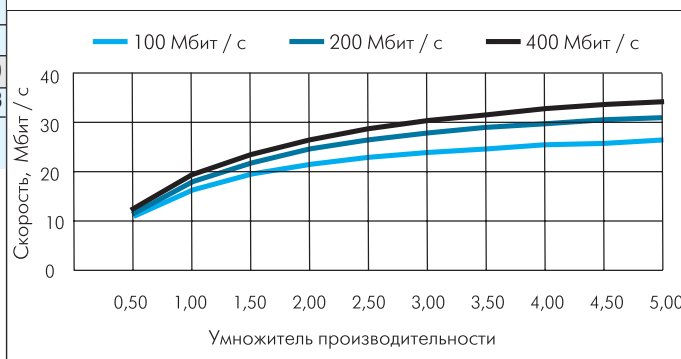


Рис.4. Зависимость скорости передачи данных от производительности компьютера. Объем данных в асинхронном пакете – 64 слова (256 байт). Умножитель производительности показывает, во сколько раз уменьшается время реакции компьютера  $T_{Hд}$

данных (да и в других системах) нельзя опираться только на быстроедействие выбранной шины. На общую производительность могут влиять и другие части системы, от которых мы, по привычке, не ожидаем "подвоха". В рассмотренном примере таким ограничителем стал компьютер.



## ЗАКЛЮЧЕНИЕ

Рассматривая различные компьютерные интерфейсы передачи данных, мы вправе были ожидать большей производительности от быстрых интерфейсов. До определенной скорости тактирования, пока влияние производительности компьютера и устройств на передачу было невысоко, эта закономерность прослеживалась. Но уже для шин IEEE-1394 стало заметным снижение темпа наращивания скорости передачи данных.

**Таблица 3. Штатные интерфейсы современных ПК**

Интерфейс	Максимальная скорость, Кбайт/с	Примечания
RS-232	14,400	Без учета накладных расходов на протокол
ECP	От 500 до 2000	Как правило, не выше 500 Кбайт/с
USB V1.1	1000	Независимо от типа передачи
<b>IEEE-1394</b>		
асинхронный режим	От 800 до 4000	В зависимости от производительности компьютера, устройства и выбранного размера пакета
изохронный режим	До 32768 в одном канале До 40000 в двух каналах	В зависимости от возможностей устройства

Из сравнения интерфейсов современных ПК (табл.3) видно, что наилучший результат по скорости перекачки данных показывает все-таки интерфейс IEEE-1394. Однако для асинхронных пересылок этот результат не стабилен и зависит от ряда факторов. Выигрыш в скорости с увеличением размера пакета возможен, только когда вероятность возникающих на линии ошибок мала. В противном случае скорость будет потеряна при повторных пересылках. Кроме то-

го, не все контроллеры, предназначенные для использования в устройствах, содержат необходимые для больших пакетов ресурсы. Переход к изохронным пакетам оправдан лишь для форматных данных, в которых частичное их искажение не приводит к серьезным последствиям. В противном случае требуются особые алгоритмы и одновременное использование изохронных и асинхронных пересылок. При этом можно добиться передачи данных на скоростях до 40 Мбайт/с.

В то же время в задачах передачи данных конкуренцию стандарту IEEE-1394 составляет интерфейс USB, обеспечивающий стабильную скорость 1 Мбайт/с. В отличие от IEEE-1394, USB есть практически во всех современных ПК и, если не требуется скорость передачи выше 1 Мбайт/с, выгодно применять его. Но сравнение этих двух интерфейсов – тема отдельной статьи.

## ЛИТЕРАТУРА

1. **Don Anderson.** Universal Serial Bus System Architecture. – MindShare, Inc. – Addison-Wesley.
2. **Jan Axelson.** USB Complete. Everything You Need to Develop Custom USB Peripherals. – Lakeview Research – Madison – WI 53704.
3. IEEE Standard for a High Performance Serial Bus. – IEEE Std 1394–1995.
4. P1394a Draft Standard for a High Performance Serial Bus (Supplement), 1999.
5. **Don Anderson.** FireWire® System Architecture, Second Edition. IEEE 1394a. – MindShare, Inc. – Addison-Wesley.