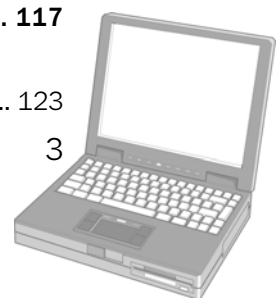
Оглавление



Введение

Персональные компьютеры применяются настолько широко, что, казалось бы, найти им новое применение в настоящее время не так и просто. Тем не менее, есть несколько сфер человеческой деятельности, где персональный компьютер только в последнее время стал завоевывать серьезные позиции. Одна из таких сфер – домашняя компьютерная электроника или, по-другому, использование ПК для создания собственных аппаратно-программных систем, способных выполнять самые разнообразные функции под управлением компьютера. Эта область включает не только создание различных робототехнических систем, но и устройств измерения, сигнализации и управления.

Эта книга посвящена практическим аспектам разработки систем компьютерной электроники, работающих под управлением операционных систем Windows на основе программно-аппаратных устройств, разработанных автором.

Литературы и документации по данной тематике мало, поскольку раскрытие этой темы сопряжено со значительными трудностями, связанными с тем, что охватывается очень широкий диапазон знаний – от элементов аналоговой и цифровой схемотехники до программирования USB и Bluetooth. Предлагаемая вашему вниманию книга призвана восполнить этот пробел.

Читатели без особого труда смогут адаптировать и усовершенствовать приведенный в книге программный код и схемотехнические решения при разработке собственных систем компьютерной электроники.

Книга рассчитана на широкий круг читателей – от начинающих до опытных пользователей.

Структура книги

Структура книги такова, что материал можно изучать выборочно, отдельными главами или последовательно, начиная с первой главы. Это позволяет различным категориям читателей изучать тот материал, который им более всего интересен.

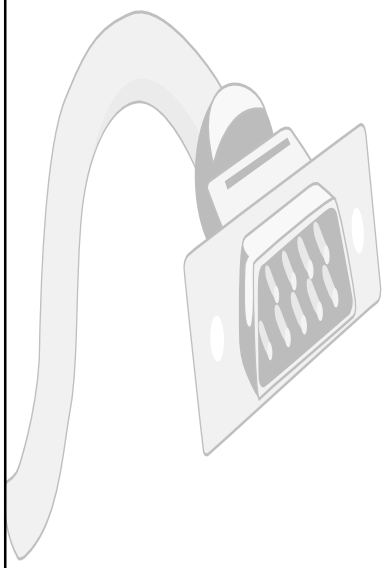
Книга состоит из 6 глав; краткий обзор каждой из них:

- глава 1 «Возможности персонального компьютера». В этой главе дается обзор основных вариантов применения персонального компьютера в системах домашней электроники;
- глава 2 «Архитектура ввода-вывода». Материал этой главы посвящен вопросам архитектуры подсистемы ввода-вывода персональных компьютеров. Рассматриваются

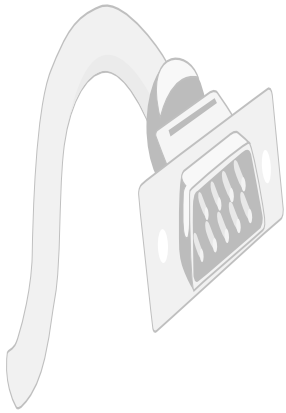
общие вопросы функционирования шинных интерфейсов, схемотехники и программирования устройств ввода-вывода пользователя;

- глава 3 «Параллельный порт в лабораторных разработках». В этой главе детально проанализированы принципы функционирования параллельного порта персонального компьютера и его программирование. Рассматриваются многочисленные аппаратно-программные проекты систем измерения и управления с управлением от параллельного порта ПК в операционных системах Windows;
- глава 4 «Последовательный порт персонального компьютера в любительских разработках». Эта глава содержит материал по аппаратной архитектуре, протоколам обмена и программированию последовательного порта персонального компьютера. Приводятся практические проекты аппаратно-программных систем с использованием последовательного порта;
- глава 5 «Звуковые карты и их применение». Материал главы посвящен вопросам разработки и программирования систем домашней электроники на основе звуковой карты. В главе проанализированы основы программирования генераторов частот и систем управления с использованием библиотеки DirectSound пакета DirectX;
- глава 6 «Интерфейсы USB и Bluetooth». В этой главе рассматривается широкий круг вопросов, связанных с применением устройств USB и Bluetooth, включая основы их функционирования и элементы программирования.

Автор благодарит коллектив издательства «ДМК» за помощь при подготовке книги к изданию. Особая признательность жене Юлии за поддержку и помощь при написании книги.



Возможности персонального компьютера



1 Возможности персонального компьютера

Персональные компьютеры в настоящее время широко используются практически во всех сферах деятельности человека, которые известны всем и каждому: образование, экономика, научные исследования, коммуникации, индустрия развлечений. Тем не менее, есть и более специализированные сферы деятельности, в которых компьютер может найти и находит достаточно эффективное применение, например, системы сбора и обработки информации в науке и промышленности, робототехника и системы управления, домашняя электроника. Как раз использование персонального компьютера в различных электронных устройствах, а также создание небольших систем сбора и обработки информации, простых систем охранной сигнализации и многих других устройств обсуждается в этой книге.

Создание собственных аппаратно-программных проектов, в основе которых лежит использование домашнего персонального компьютера, требует определенных знаний, как архитектуры самого ПК, так и определенных знаний и навыков программирования. Кроме того, разработка таких систем требует определенного уровня знаний в схемотехнике, как в аналоговой, так и в цифровой. Тем не менее, рассмотренные в книге проекты под силу реализовать даже пользователям средней руки. Проектирование собственных устройств с использованием ПК базируется на применении возможностей аппаратных средств, входящих в состав компьютера:

- параллельного порта принтера;
- последовательного порта;
- звуковой карты;
- устройств USB и Bluetooth.

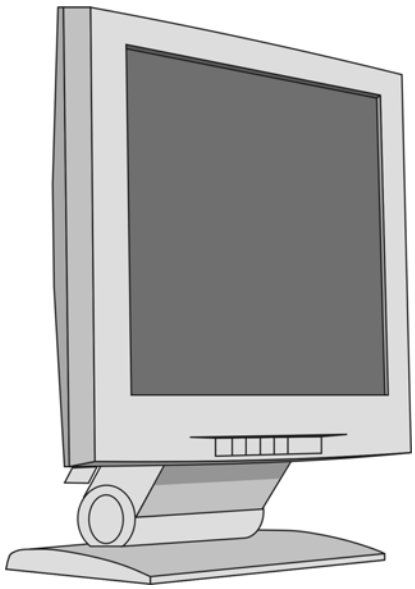
Все эти устройства, помимо их стандартного применения, позволяют создавать и проекты домашней электроники, которые могут, в принципе, не уступать даже лабораторным и промышленным системам. Более того, в мире выпускается очень много оборудования, применение которого базируется на использовании вышеперечисленных аппаратных средств. Достаточно вспомнить многочисленные версии «электронных осциллографов», базирующихся на применении звуковой карты и протокола USB, системы сбора данных на базе последовательного и параллельного портов, системы удаленного управления на базе технологии Bluetooth.

В последнее время ведущие разработчики программного обеспечения, в их числе и Microsoft, начали создание программного обеспечения, направленного на поддержку различных аппаратно-программных систем, которые может разработать любой пользователь, используя то или иное периферийное оборудование. Например, известный и быстро развивающийся программный продукт, такой, как Microsoft Robotics Studio, предлагает программный интер-

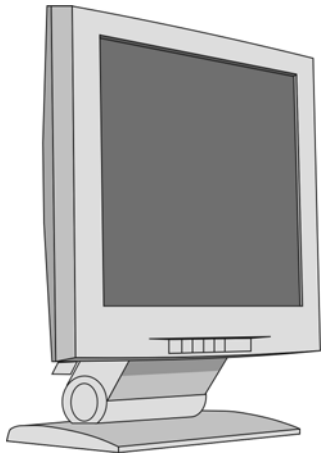
фейс, позволяющий создавать системы управления роботами и другими устройствами домашней электроники. Фирма Phidgets и целый ряд других фирм выпускают электронные модули на базе USB, которые работают с Robotics Studio, снабжены программным интерфейсом и легко интегрируются в ПК. С помощью таких устройств можно создавать системы управления и измерения.

Этот перечень можно продолжить. Интерес пользователей к разработкам собственных аппаратно-программных проектов с каждым днем возрастает, и с каждым днем на рынке появляется все больше и больше устройств, подобных тем, которые выпускает Phidgets. Более того, появление и быстрое развитие беспроводных технологий открывает новые горизонты для творчества. Кстати, в этой нише в последнее время появилось также много фирм, выпускающих аппаратно-программные модули для создания пользовательских электронных систем. К сожалению, такие системы являются относительно дорогими для отечественных пользователей, чтобы их можно было легко использовать в собственных электронных проектах. Предлагаемые в книге проекты не требуют больших финансовых затрат на комплектующие, а используемое в них программное обеспечение является бесплатным. Такие аппаратно-программные проекты позволяют решать довольно серьезные задачи.

Большинство читателей хорошо представляют себе архитектуру персонального компьютера, но, возможно, не знакомы с тем, как взаимодействуют устройства ввода-вывода (а к ним относятся практически все устройства, кроме памяти) с процессором. В следующей главе мы рассмотрим принципы функционирования устройств ввода-вывода, поскольку именно они будут применяться для разработки наших проектов.



Архитектура ввода-вывода

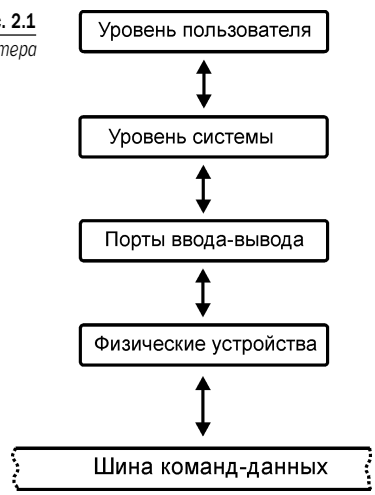


2 Архитектура ввода-вывода

В этой главе вкратце рассматриваются основные аппаратно-программные аспекты, касающиеся организации и функционирования подсистем ввода-вывода в компьютерных системах с различной конфигурацией системных шин, включая шину PCI платформы x86, которая наиболее часто применяется в персональных компьютерах. Такой анализ поможет глубже понять принципы функционирования различных периферийных устройств в различных системах и облегчит задачи программирования таких устройств. Под термином «подсистема ввода-вывода» обычно понимают как аппаратные устройства компьютера, так и программный интерфейс, позволяющий взаимодействовать с ними.

Операции ввода-вывода для большинства компьютерных систем отличаются от операций с памятью, как на уровне инструкций процессора, так и в плане схемотехнической реализации самих устройств. Схему ввода-вывода информации можно представить себе так, как показано на рис. 2.1.

Рис. 2.1
Иерархия подсистемы ввода-вывода компьютера



На самом нижнем уровне устройства взаимодействуют с аппаратурой компьютера через одну или несколько шин команд-данных, интерфейс которых для разных аппаратных платформ вообще отличается, хотя некоторые стандарты (например, PCI PCI Express и т. д.) используются разными платформами.

Взаимодействие устройств на уровне шин осуществляется с помощью электрических сигналов, временные характеристики которых соответствуют используемому стандарту, а аппаратный интерфейс физического устройства должен обеспечить генерацию всех необходимых сигналов обмена.

Обмен данными с устройством осуществляется посредством портов ввода-вывода и отображаемой памяти. В этом случае все операции с устройством выполняются на уровне инструкций процессора или микроконтроллера. Термин «вывод данных» означает операцию записи данных в устройство, а термин «ввод данных» означает чтение данных из устройства. Далее мы будем использовать эти термины как синонимы.

Схемотехническая реализация портов ввода-вывода зависит от временной диаграммы работы сигнальных линий процессора и реализации алгоритма обмена. Напомню, что центральным устройством любой компьютерной системы является процессор или микроконтроллер, который формирует сигналы обмена данными с устройствами в соответствии с определенными правилами.

Для стандартизации обмена данными процессора и устройств ввода-вывода используется целый ряд промышленных стандартов обмена данными (так называемых «шинных интерфейсов» или просто шин), наиболее известными из которых являются PCI и PCI Express. Шинный интерфейс предоставляет разработчикам устройств и пользователям единый стандарт обмена данными, что значительно облегчает интеграцию различных устройств в систему. Физически шина представляет собой набор электрических линий, сигналы на которых подчиняются определенным правилам или стандартам, например, стандарту PCI, откуда и название шины.

С другой стороны, в большинстве случаев набор сигналов процессора или контроллера не соответствует стандартам шины, поэтому между процессором и шиной помещают дополнительное устройство, которое обычно называют контроллером шины. Контроллер шины выполняет функции преобразования сигналов процессора в сигналы шинного интерфейса. Например, для систем с процессорами Intel контроллер шины преобразует сигналы процессора в стандарт PCI или PCI Express (рис. 2.2).

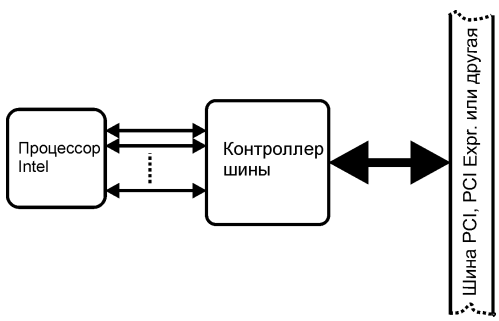


Рис. 2.2
Упрощенная структура шинной архитектуры

На этом рисунке контроллер шины для упрощения показан в виде одного модуля, хотя на самом деле он может включать несколько интегральных микросхем и быть довольно сложным. Шина PCI в данном примере является системной шиной, и к ней могут подключаться другие устройства, сигналы которых соответствуют стандарту данной шины. Например, к шине PCI может подключаться хост-контроллер шины USB, который, в свою очередь, будет формировать сигналы синхронизации на шине USB для подключения USB-устройств.

Таким образом, в любой более-менее сложной компьютерной системе обмен данными между процессором и устройствами выполняется посредством одной или нескольких шин

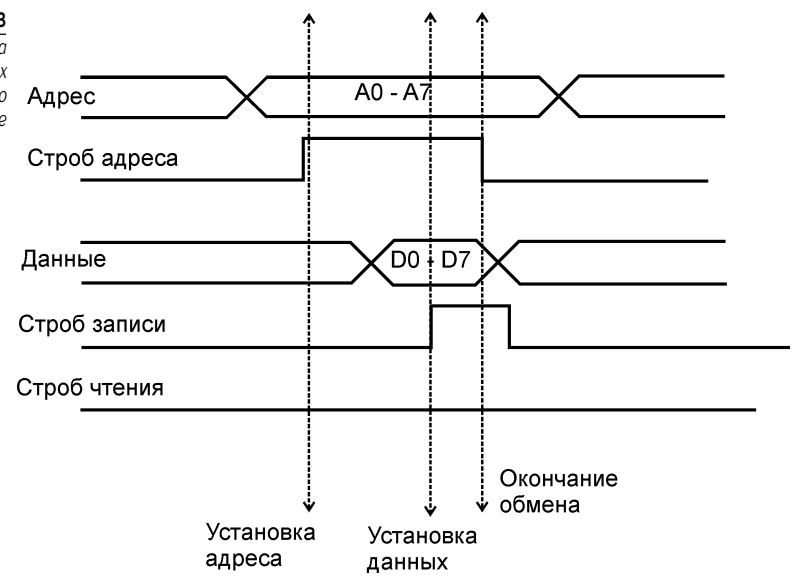
команд/данных. В дальнейшем для краткости мы будем использовать термин «системная шина». Шина PCI далеко не единственный тип системной шины, используемый в компьютерных системах. Существует целый ряд других, широко распространенных промышленных стандартов (CANBUS, I2C, SPI и т. д.), которые весьма широко используются в различных компьютерных системах.

Как осуществляется обмен данными по шине? В компьютерной системе для выполнения обмена данными с любым устройством необходимо выполнить стандартную последовательность шагов:

- установить на адресных линиях системной шины или шины команд/данных адрес устройства, к которому происходит обращение. При этом устройство, распознавшее адрес как собственный, должно выполнить необходимые переключения в схеме для подготовки обмена данными (дешифрация адреса, выбор/разрешение чтения/записи и т. д.);
- установить сигнал синхронизации адреса. Обычно это перепад напряжения на одной из управляющих линий системной шины, который предназначен для использования логикой устройства ввода-вывода;
- установить на линиях данных системной шины данные и сигнал синхронизации данных, сообщающий устройству, что данные действительны;
- закончить обмен данными, сняв адрес устройства с линий адреса и данные с линий данных.

Это весьма упрощенная схема функционирования большинства шин, но она дает представление о принципах их работы. Рассмотрим, как могли бы выполняться операции ввода-вывода данных при подключении устройства к гипотетической шине. Начнем с записи байта в устройство. Положим, что запись (вывод) байта осуществляется на шине с 8-ю линиями адресов и 8-ю линиями данных (рис. 2.3).

Рис. 2.3
Временная диаграмма записи данных в устройство на гипотетической шине



Наша гипотетическая шина имеет 8 линий адреса, 8 линий данных и две линии синхронизации записи и чтения данных.

Операцию обмена данными с устройством посредством сигналов шины называют «циклом шины». Обмен данными на нашей шине (см. рис. 2.3) начинается с установки адреса на линиях адреса A0–A7, для декодирования которого строб адреса устанавливается в высокий уровень. Цифровые схемы устройства на этом этапе должны обеспечить декодирование адреса. Если устройство распознает адрес на линиях адреса как «свой», то цифровая логика должна подготовить обмен данными (в данном случае, запись байта в устройство). При записи данных строб чтения на шине должен иметь низкий уровень.

Далее, по прошествии некоторого интервала времени, на линиях данных D0–D7 шины выставляется байт данных, который должен быть зафиксирован устройством после перехода строба записи в высокий уровень. Далее, процессор снимает адрес с линии адреса, снимает строб адреса и данных, заканчивая, таким образом, цикл записи данных в устройство.

Аппаратная реализация записи данных на стороне устройства не очень сложна. Функционально устройство ввода-вывода состоит из нескольких портов, которые схемотехнически представляют собой цифровые регистры с фиксацией данных (стробируемые регистры) для вывода данных и буферные усилители без стробирования для ввода данных в систему.

Устройство вывода с одним портом вывода применительно к временной диаграмме шины из рис. 2.3 можно представить приблизительно такой функциональной схемой (рис. 2.4):

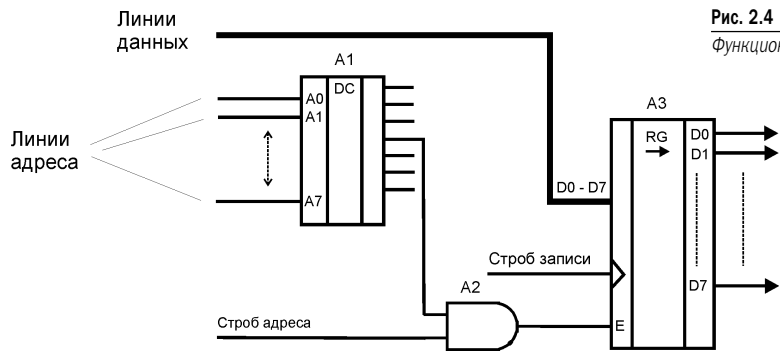


Рис. 2.4
Функциональная схема вывода данных

Здесь показана весьма упрощенная функциональная схема порта вывода данных. Если на шину адреса выставлен адрес порта вывода, то при высоком уровне строба адреса сигнал с выхода элемента «И» на микросхеме A2 разрешает работу регистра защелки A3. Если на шину данных будут выставлены данные (здесь показана 8-разрядная шина данных), то при перепаде строба записи из низкого в высокий уровень данные будут записаны в регистр A3. По этой схеме предполагается, что системная шина имеет отдельные линии адреса и данных, а адрес удерживается действительным во время всего цикла записи данных. Может возникнуть вопрос: нельзя ли вместо регистра-защелки использовать обычный буфер?

Такой вариант также возможен, но если записанные данные далее должны каким-то образом обрабатываться в устройстве, то следует их зафиксировать во избежание потери, что и выполняет регистр-защелка A3.

Чтение (ввод) данных с устройства может быть реализовано в простейшем цикле чтения нашей шины по схожей временной диаграмме, только вместо строба записи должен использоваться сигнал синхронизации чтения (строб чтения данных) (рис. 2.5).

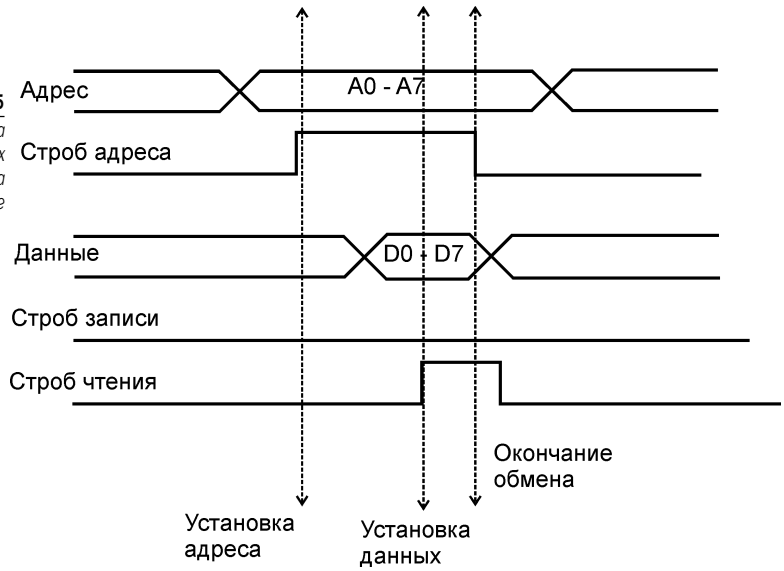


Рис. 2.5
Временная диаграмма чтения данных из устройства на гипотетической шине

В отличие от записи данных, сигнал строба записи теперь будет иметь низкий уровень, а чтение байта данных осуществляться по высокому уровню строба чтения.

Схемотехническая реализация устройства ввода данных не намного сложнее, чем при выводе данных. Функциональная схема порта ввода данных показана на рис. 2.6.

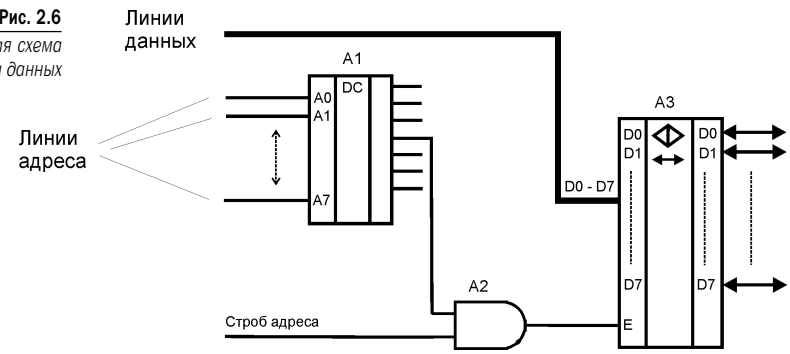


Рис. 2.6
Функциональная схема ввода данных

Порт ввода данных схемотехнически реализуется, как правило, проще, поскольку данные могут быть сразу прочитаны, без фиксации в регистре-защелке.

В схеме, показанной на рис. 2.6, для ввода данных с порта ввода используется тристабильный буфер А3. Опять-таки, для мультиплексированных шин данных типа PCI функциональная схема порта ввода будет сложнее. Комбинируя регистры ввода-вывода в одном устройстве, можно создать простое устройство ввода-вывода с двумя портами ввода-вывода. Приблизительно такая реализация шины ввода-вывода применяется в интерфейсе параллельного порта персонального компьютера, хотя количество портов ввода-вывода там больше (два порта вывода и один ввода).

Интерфейсы современных шин, естественно, намного сложнее, чем наша гипотетическая шина, что обусловлено влиянием нескольких факторов. Одним из них является размерность адресного пространства. В современных компьютерах используется как минимум 32-разрядная шина адреса, что потребовало бы как минимум 32 отдельных линий шины. Если учесть, что в компьютерных системах используется 8 бит данных, то только для линий адрес/данных потребуется как минимум 40 линий. Схемотехнически это выполнить очень сложно, особенно учитывая специфику работы с цепями сигналов высокой частоты. Для реализации эффективного обмена данными в этом случае в большинстве современных компьютерных шин используется механизм мультиплексирования, когда одни и те же линии данных используются в качестве линий адреса и данных.

Вторым важным моментом является то, что разные устройства имеют разное быстродействие, и система должна знать, готово ли устройство принять/передать данные, завершилась операция или нет и т. д. Реализовать обратную связь с устройством помогает механизм квитирования («рукопожатия»), когда начало любой операции на шине зависит от ответа устройства.

Эти механизмы реализованы в большинстве шин обмена данными, например, в той же PCI. Проанализируем, как происходит, например, запись данных в устройство на шине PCI (рис. 2.7).

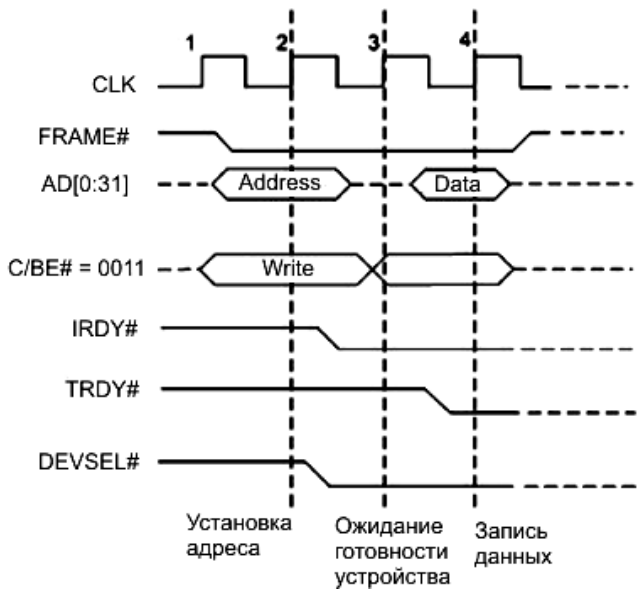


Рис. 2.7
Временная диаграмма работы цикла записи на шине PCI

Здесь я не буду анализировать назначение сигнальных линий PCI-шины – их назначение описано в многочисленной литературе и спецификациях на стандарт. По ходу я объясню смысл сигналов, присутствующих на шине. Самое первое, что следует отметить при анализе работы шины PCI – синхронизацию любых операций по фронту импульсов тактовой частоты CLK. Информация на шине будет действительна при перепаде сигнала CLK из высокого уровня в низкий, причем это касается команд, адресов, данных и сигналов состояния.

Новый цикл шины начинается с установки сигнала FRAME# в низкий уровень, который свидетельствует о выполнении операции на шине. Высокий уровень на этой линии запрещает обмен данными. После установки сигнала FRAME# в низкий уровень на линии адреса

AD[0:31] выставляется адрес устройства, которому необходимо передать данные. Одновременно на линии команд C/BE# выставляется код команды, которая будет выполняться (в данном случае, код равен 0011, что соответствует записи).

На линию DEVSEL# выставляется сигнал выбора устройства (активный уровень низкий). Наконец, контроллер шины выставляет на линию сигнал IRDY# готовности данных для записи устройства. Пока целевое устройство (target) не выставит сигнал готовности к приему данных на линию TRDY#, никакой записи данных в устройство не произойдет, и логика шины будет ожидать, пока сигнал на линии TRDY# не станет активным (низкий уровень).

Из временной диаграммы видно, что при 3-м тактовом импульсе CLK запись данных все еще невозможна, поскольку нет готовности устройства к приему данных (об этом свидетельствует высокий уровень на линии TRDY#). Наконец, устройство сигнализирует о готовности принять данные, которые и записываются в устройство по фронту 4-го синхроимпульса CLK. Цикл записи заканчивается при установке сигнала FRAME# в высокий уровень.

Как видно из описания цикла работы шины PCI, она, по существу, является асинхронной двунаправленной шиной обмена данными, скорость обмена на которой может варьироваться в зависимости от характеристик производительности устройства ввода-вывода. Аппаратный интерфейс устройства ввода-вывода для такой шины будет, естественно, несколько сложнее, чем тот, который мы рассмотрели для гипотетической шины обмена данными ранее в этой главе.

Что же касается программной части, то доступ к портам ввода-вывода осуществляется посредством операций чтения-записи с использованием стандартных инструкций процессора (для процессоров, совместимых с Intel, это команды in/out и их модификации). Если же устройство работает с регистрами, отображаемыми на память, то доступ к ним осуществляется с использованием инструкций mov. Данные, с которыми оперирует порт, могут иметь размер 1 байт, 2 байта (слово) и 4 байта (двойное слово).

Проанализируем, как выполнить чтение регистра состояния параллельного порта. Для этой цели можно использовать инструкцию in, например:

```
mov DX, 0x379
in AL, DX
```

Первая команда помещает в регистр DX адрес порта, а вторая читает данные размером в 1 байт в регистр AL процессора.

Из всего пространства адресов ввода-вывода (0–0xFFFF) часть адресов резервируется системой и не должна использоваться в программах пользователей — это адреса с 0xF8 по 0xFF.

Все устройства персонального компьютера взаимодействуют с процессором посредством шины PCI или PCI Express. При этом взаимодействие различных интерфейсов осуществляется с помощью специальных контроллеров, преобразующих сигналы на специализированных шинах в сигналы шин PCI или PCI Express. Например, для взаимодействия устройства USB с системой необходимо преобразовать сигналы на шине USB в сигналы шины PCI, для чего в компьютерной системе имеется хост-контроллер USB (как правило, он интегрируется в материнскую плату, хотя может быть выполнен и как отдельное устройство).

Второй пример. Для преобразования сигналов интерфейса RS-232, используемого при обмене данных через последовательный порт, в сигналы PCI на материнской плате имеется контроллер UART (асинхронного приемопередатчика). Специализированные контроллеры используются и для любых других интерфейсов(параллельного порта, Bluetooth, Wi-Fi и т. д.).

В более ранних операционных системах Windows 98/Me программы пользователя могли напрямую обращаться к портам ввода-вывода посредством инструкций процессора in и out, о которых я упоминал ранее. Для операционных систем Windows 2000/XP/2003, в которых код пользовательской программы не имеет доступа к адресному пространству системы (к которому принадлежат и порты ввода-вывода), выполнение операций с устройствами ввода-вывода возможно только посредством драйверов устройств, а прямой доступ к портам в некоторых случаях невозможен даже из драйвера.

При работе процессора в защищенном режиме защита доступа к портам ввода-вывода обеспечивается двумя механизмами:

- установкой уровня привилегий ввода-вывода в поле IOPL (I/O Provilege level) (биты 12-13 в регистре флагов процессора EFLAGS);
- картой, содержащей биты доступа в сегменте состояния задачи TSS (Task State Segment).

Поле IOPL в регистре флагов позволяет управлять доступом к пространству адресов посредством ограничения использования определенных инструкций процессора. Этот механизм защиты позволяет устанавливать необходимый для операционной системы уровень привилегий для выполнения операций ввода-вывода. В общем случае, доступ к адресному пространству ввода-вывода ограничен уровнями 0 и 1. В этом случае операции ввода-вывода могут выполнять драйверы ядра и драйверы устройств, в то время как приложения с меньшим уровнем привилегий операции ввода-вывода выполнить не смогут. Для менее привилегированных программ доступны механизмы ввода-вывода посредством вызовов функций интерфейса WinAPI.

Следующие инструкции будут выполнены только в том случае, если уровень привилегий выполняющейся программы (CPL, Current Privilege Level) меньше или равен установленному в IOPL: in, ins, out, outs, cli, sti.

Любая попытка выполнить эти инструкции в программе, уровень привилегий которой недостаточен, вызывает исключение общей защиты (General-Protection Exceptin). Поскольку каждая выполняющаяся задача получает собственную копию регистра флагов, то для нее устанавливается свой IOPL.

Рассмотрим инструкции процессора, используемые при выполнении операций ввода-вывода. Всех их можно условно разделить на две группы: инструкции, выполняющие пересылку одиночных данных (байта, слова или двойного слова), и инструкции, выполняющие пересылку строк байтов, строк слов и строк двойных слов.

Инструкции in (ввод данных с порта) и out (вывод данных в порт) передают данные между портом и регистрами EAX (32-разрядная операция), AX (16-разрядная операция) или AL (операция с одним байтом). Адрес порта может быть задан непосредственно в команде или посредством регистра DX.

Например,

```
in  AL, 0x378
mov DX, 0x379
out DX, AL
```

Строковые инструкции ins и outs пересылают данные между памятью и портом ввода-вывода. При этом адрес порта заносится в регистр DX, а адрес области памяти определяется одной из пар регистров DS:ESI (при вводе) или ES:EDI (при выводе данных).

НЕОБЫЧНОЕ ИСПОЛЬЗОВАНИЕ ПК

При использовании префиксов повторения (например, rep) инструкции ins и outs выполняют поблочную передачу данных. В этом случае регистр ESI или EDI инкрементируется или декрементируется (в зависимости от значения флага DF), чтобы указывать на следующий байт, слово или двойное слово.

Следующие, более высокие уровни взаимодействия с устройствами используются операционными системами (системный уровень) и пользовательскими приложениями. Во всех современных операционных системах все устройства ввода-вывода, а также все файлы и каталоги для пользовательских процессов представлены как объекты файловой системы, а это означает, что доступ к объектам и операции с ними выполняются по некоторым унифицированным правилам, независимо от природы объекта (устройство, файл, именованный канал, сокет и т. д.). Такая модель позволяет пользовательским и системным процессам использовать для работы с объектами файловой системы набор стандартных функций интерфейса API. Замечу, что эта модель взаимодействия изначально была использована в операционных системах UNIX и хорошо себя зарекомендовала.

Программный код, выполняющийся в таких системах, может функционировать в одном из двух режимов: пользовательском или в режиме ядра. Большая часть программных компонентов операционной системы (системные службы, драйверы ядра и устройств и т. д.) функционирует в режиме ядра. Программный код режима ядра имеет практически неограниченный доступ к аппаратно-программным ресурсам системы, включая устройства ввода-вывода.

В рассмотренном только что примере чтение порта ввода с адресом 0x379 возможно только из программы, выполняющейся в режиме ядра. Например, программный код пользовательской программы в Windows работает только в режиме пользователя и взаимодействует с операционной системой посредством интерфейса прикладного программирования WinAPI, который реализован в виде подсистемы Win32.

В следующих главах на практических примерах мы увидим, как работать с такими устройствами ввода-вывода как параллельный и последовательный порты, звуковая карта, устройства USB и Bluetooth.

Параллельный порт в лабораторных разработках

3 Параллельный порт в лабораторных разработках

Параллельный порт персонального компьютера является одним из ранних аппаратно-программных интерфейсов, изначально разработанных для обмена данными с печатающими устройствами (принтерами). В настоящее время параллельный порт, помимо своего прямого назначения, довольно часто используется в разнообразных проектах для управления несложными аппаратными устройствами, а также для обработки и сбора данных.

Параллельный порт позволяет принимать 9 бит данных от устройства или отправлять 12 бит данных устройству. Аппаратно интерфейс реализован посредством 4-х линий управления, 5-ти линий состояния и 8-ми линий данных, которые подключены к 25-контактному разъему типа DB-25, расположенному на задней стенке компьютера.

Самым первым стандартом параллельного порта стал интерфейс, известный под названием Centronics. Он описывает сигналы, протокол обмена и расположение внешних контактов на разъемах в компьютере и принтере. С точки зрения схемотехники, интерфейс Centronics реализован как группа из трех регистров ввода-вывода: данных, управления и состояния, которые доступны программисту.

В таблице 3.1 приведено описание сигналов интерфейса, нумерация выводов на разъеме DB-25 в компьютере и направление сигналов по отношению к параллельному порту компьютера.

Указанная спецификация сигналов, соответствующая интерфейсу Centronics, была стандартизована под названием SPP (Standard Parallel Port). В дальнейшем спецификации режимов обмена данными получили дальнейшее развитие в протоколах EPP (Enhanced Parallel Port) и ECP (Extended Capabilities Port), которые были разработаны и приняты под эгидой стандарта IEEE 1284, обеспечивая обратную совместимость со стандартным режимом SPP.

Проанализируем упрощенную временную диаграмму обмена для параллельного порта, работающего в стандартном режиме SPP или, проще говоря, Centronics. Перед анализом работы интерфейса следует отметить, что при обмене данными между устройствами одно из них является инициатором (источником) обмена, а другое — приемником.

Для стандартного интерфейса SPP источником обмена является обычно компьютер, или, по–другому, хост, а приемником — принтер или иное устройство. Временная диаграмма обмена по протоколу SPP показана на рис. 3.1.

Рассмотрим пример передачи байта данных от источника к приемнику. Передача выполняется в следующей последовательности:

- вначале проверяется состояние сигнала BUSY: если он равен 0, то источник приступает к передаче байта данных, выставляя на линиях D0–D7 биты данных, после чего

Конец ознакомительного фрагмента.
Приобрести книгу можно
в интернет-магазине
«Электронный универс»
e-Univers.ru