

ЛАБОРАТОРНЫЙ ПРАКТИКУМ ПО ДИСЦИПЛИНЕ «ИНТЕРФЕЙСЫ И УСТРОЙСТВА ВЫЧИСЛИТЕЛЬНЫХ МАШИН»

1. Требования к лабораторным работам:

Язык программирования: C++

Операционная система: Windows

Не использовать высокоуровневые библиотеки для работы с устройства (как WMI в NET.)

ЭРУД по дисциплине: [ссылка](#)

2. Задания на лабораторные работы:

1. Энергопитание

Вывести информацию об энергопитании компьютера:

- в режиме реального времени показывать тип энергопитания;
- информация о типе батареи;
- уровень заряда батареи (%);
- текущий режим энергосбережения;
- реализовать переход в спящий режим и гибернацию из консоли.
- вывести время работы аккумулятора без подключения к зарядке;
- вывести оставшееся время работы аккумулятора после отключения зарядного устройства.

2. Конфигурационное пространство PCI

Вывести список всех устройств, подключенных к шине PCI, с их характеристиками (DeviceID и VendorID состоящие из 4-х символов) в виде таблицы.

Подключение к шине производить с помощью готовых библиотек нельзя!!! Подключение к шине реализовать с применением портов ввода-вывода.

Дополнительная информация для выполнения лабораторной работы представлена после всех заданий на лабораторные работы.

3. HDD / SSD

Вывести данные о диске (ах) по варианту, подключенных к компьютеру:

- модель;
- изготовитель;
- серийных номер;

- версия прошивки (firmware);
- сведения о памяти (свободно/занято/всего);
- тип интерфейса;
- список поддерживаемых режимов.

По вариантам:

- диск HDD;
- диск SSD.

Только в случае, отсутствия диска (HDD или SSD), указанного по варианту на сдаваемом компьютере, разрешается вывести информацию о имеющемся диске (ах). Отсутствие диска по варианту необходимо продемонстрировать в диспетчере устройств и сообщить преподавателю во время выдачи задания.

Вывод информации о подключенных USB-устройствах не делать.

Подключение к шине производить с помощью готовых библиотек нельзя!!! Подключение к шине реализовать через порты ввода-вывода.

[Дополнительная информация для выполнения лабораторной работы представлена после всех заданий на лабораторные работы.](#)

4. Веб-камера

Вывести информацию об установленной веб-камере.

Осуществить захват изображения (фото и видео) с последующим сохранением в файл.

По вариантам:

- предусмотреть скрытый вариант фотонаблюдения, когда на мониторе и на панели задач не отображается информация о Вашем работающем приложении;
- предусмотреть скрытый вариант видеонаблюдения, когда на мониторе и на панели задач не отображается информация о Вашем работающем приложении.

5. USB-устройства

Реализовать мониторинг USB-портов:

- отслеживать появление нового USB-устройства в системе;
- события безопасного и небезопасного извлечения устройства в консоли;
- отказ в безопасном извлечении.

Для всех извлекаемых USB-устройств предусмотреть возможность программного безопасного извлечения.

Приложение должно корректно работать с модифицированными USB устройствами – например, разбитыми на пару дисков или смонтированными как CD-ROM + Flash.

Список устройств подключение которых необходимо продемонстрировать:

- мышь;
- флэшка.

6. Работа с интерфейсом Bluetooth

Реализовать мониторинг устройств подключаемых через интерфейс Bluetooth. Подключиться к устройству и выполнить задание по одному из указанных ниже вариантов (музыкальный / голосовой)

Варианты передачи файла:

1. передать файл с ноутбука на умную колонку и реализовать механизм автоматического воспроизведения после передачи;
2. передать файл с ноутбука на ноутбук и воспроизвести;
3. передать файл с ноутбука на ноутбук и реализовать механизм автоматического воспроизведения после передачи и реализовать механизм автоматического воспроизведения после передачи (+1 балл за сложность);
4. передать файл с ноутбука на телефон и реализовать механизм автоматического воспроизведения после передачи и реализовать механизм автоматического воспроизведения после передачи (+2 балл за сложность)

3. ДОПОЛНИТЕЛЬНАЯ ИНФОРМАЦИЯ ДЛЯ ВЫПОЛНЕНИЯ ЛАБОРАТОРНЫХ РАБОТ №2 и №3

Лабораторная работа №2

В стандарт PCI заложены возможности автоматического распределения системных ресурсов (пространств памяти и ввода-вывода и линий запроса прерываний) между устройствами. Каждая функция каждого устройства имеет **конфигурационное пространство** (блок регистров) размером 256 байт, не приписанных ни к пространству памяти, ни к пространству ввода-вывода. Доступ к ним осуществляется по специальным циклам шины Configuration Read и Configuration Write, вырабатываемым с помощью одного из вышеописанных механизмов. В этом пространстве есть области, обязательные для всех устройств, и области специфические. Конкретное устройство может иметь регистры не во всех адресах, но должно поддерживать нормальное завершение для адресуемых к ним операций. При этом чтение несуществующих регистров должно возвращать нули, а запись выполняться как холостая операция.

После аппаратного сброса (или при включении питания) устройства PCI не отвечают на обращения к пространству памяти и ввода-вывода, они доступны только для операций конфигурационного считывания и записи. В этих операциях устройства выбирают по индивидуальным сигналам IDSEL и сообщают о потребностях в ресурсах и возможных вариантах конфигурирования. После распределения ресурсов, выполняемого программой конфигурирования (во время теста POST), в конфигурационные регистры устройства записываются параметры конфигурирования. В том числе в процессе конфигурирования выставляются биты, разрешающие устройствам отслеживать адреса, передаваемые по шине в обычных транзакциях ввода-вывода. И только после этого к устройствам становится возможным доступ по командам обращения к памяти и портам ввода-вывода. Для того чтобы всегда можно было найти работоспособную конфигурацию, все ресурсы, занимаемые картами, должны быть перемещаемыми в своих пространствах. Для многофункциональных карт каждая функция должна иметь собственное конфигурационное пространство.

Конфигурационное пространство устройства начинается со стандартного заголовка, в котором содержатся **идентификаторы производителя, устройства и его класса**, а также описание требуемых и занимаемых системных ресурсов. После заголовка могут располагаться регистры, специфичные для устройства; они могут занимать адреса конфигурационного пространства в пределах 40-FFh.

Формат заголовка приведен в табл. 1.2.

Таблица 1.2. Конфигурационное пространство устройства PCI, обязательные поля

смещение, байт	31	24	23	16	15	8	7	0
0x00	Device ID				Vendor ID			
0x04	Status				Command			
0x08	Class ID						Revision ID	
0x0C	BIST		Header type		Latency timer		Cache line size	
0x10-0x24	Base address registers							
0x28	Cardbus CIS pointer							
0x2C	Subsystem ID				Subsystem vendor ID			
0x30	Expansion ROM base address							
0x34	reserve						Capabilities ptr	
0x38	reserve							
0x3C	max lat		min gnt		Interrupt pin		Interrupt line	

Перечисленные ниже поля идентификации допускают только чтение:

- Device ID – идентификатор устройства, назначаемый производителем.
- Vendor ID – идентификатор производителя микросхемы PCI, назначенный PCI SIG. Идентификатор 0xFFFF является недопустимым; это значение должно возвращаться при чтении конфигурационного пространства несуществующего устройства.
- Revision ID – версия продукта, назначенная производителем. Используется как расширение поля Device ID.
- Header Type – тип заголовка (биты 6:0), определяющий формат ячеек в диапазоне 0x10-0x3F и несущий признак многофункционального устройства (если бит 7=1). На рисунке приведен формат заголовка типа 0, относящийся именно к устройствам PCI. Тип 01 относится к мостам PCI-PCI; тип 02 относится к мостам шины CardBus и т.д.
- Class Code – код класса, определяющий основную функцию устройства, а иногда и его программный интерфейс, если он стандартизирован. Старший байт (адрес 0x0B) определяет базовый класс, средний – подкласс, младний – программный интерфейс.
- Subsystem Vendor ID – название производителя устройства, которое содержит данный контроллер; если производитель контроллера и всего устройства совпадают, либо устройство входит в состав микросхем системной логики (чипсета), данное поле содержит 0.

Остальные поля заголовка являются регистрами устройств, допускающими как запись, так и чтение. В данной методичке они не рассматриваются.

Лабораторная работа №3

Интерфейс ATA (AT Attachment for Disk Drives) разрабатывался в 1986-1990 гг. для подключения накопителей на жестких магнитных дисках к компьютерам IBM PC AT с шиной ISA. Стандарт, выработанный комитетом ANSI X3T10, определяет набор регистров устройств и назначение сигналов 40-контактного интерфейсного разъема. Интерфейс появился в результате переноса стандартного (для PC/AT) контроллера жесткого диска ближе к накопителю, то есть создания устройств со встроенным контроллером – IDE (Integrated Drive Electronics). Название устройства со временем стало неофициальным названием интерфейса, и сегодня аббревиатура IDE часто используется как синоним ATA.

Стандартный контроллер AT позволял подключать до двух накопителей. В интерфейсе ATA было обеспечено параллельное подключение двух контроллеров, то есть двух устройств хранения данных. Обращение к ним возможно по очереди, после выполнения адресации одного из устройств.

В спецификации ATA фигурируют следующие компоненты:

- Хост-адаптер — средства сопряжения интерфейса ATA с шиной компьютера. Хост – это вычислительное ядро системы, которое выполняет заданный код. Хост-контроллер – это более развитый вариант хост-адаптера.
- Ведущее устройство (Master) – устройство, в спецификации ATA официально называемое Device0 (устройство 0). Это жесткий диск с интегрированным контроллером, подключенный к хост-контроллеру или адаптеру.
- Ведомое устройство (Slave) – устройство, в спецификации официально называемое Device1 (устройство 1).

Хост-адаптер и устройства объединяются кабелем-шлейфом, соединяющим параллельно одноименные контакты интерфейсных разъемов. Регистры обоих контроллеров оказываются расположенными в одних и тех же областях пространства ввода-вывода. Для **выбора устройства**, исполняющего текущую команду, используется бит выбора накопителя (DEV) в регистре номера устройства и головки (drive/head register, DH). Если бит DEV=0, выбрано ведущее устройство, если DEV=1 – ведомое. Запись в этот регистр воспринимается сразу обоими устройствами, на обращения к остальным регистрам реагирует только выбранное.

Достаточно универсальный набор сигналов позволяет подключать любое устройство со встроенным контроллером, которому в пространстве портов ввода-вывода достаточно того же набора регистров. Принятая система команд и регистров, являющаяся частью спецификации ATA, ориентирована на блочный обмен данными с устройствами прямого доступа. Для иных устройств существует спецификация ATAPI, основанная на тех же аппаратных средствах, но позволяющая обмениваться пакетами управляющей информации (ATA Package Interface, ATAPI). Структура и наполнение пакетов ATAPI позаимствованы из универсального интерфейса SCSI. Пакетный интерфейс позволяет расширить границы применения шины ATA, распространив его на оптические накопители, ленточные накопители, чейнджеры, карточки памяти и т.п.

2.1.2. Адресация данных в интерфейсе ATA

Адресация в ATA имеет «дисковые корни»: для накопителей изначально указывали адрес цилиндра (cylinder), головки (head) и сектора (sector) – элементы так называемой **трехмерной геометрии** CHS. Сначала эта адресация точно соответствовала реальной геометрии – физически сектор действительно находился по указанному адресу. Позже по ряду причин диски ATA стали описывать внешней геометрией, отличающейся от реальной внутренней. Преобразование адресов в реальные выполняется встроенным контроллером устройства. В системе CHS устройство ATA позволяет адресовать до 267 386 880 (65536x16x255) секторов (блоков), что при размере сектора в 512 байт дает 136 902 082 560 байт (около 137 Гб).

Позже пришли к **линейной адресации логических блоков** LBA (Logical Block Addressing), где адрес блока (сектора) определяется 28-битным числом, что позволяет адресовать до 268 435 455 (228 Гб) блоков – немного больше, чем в CHS. Для устройств ATA, поддерживающих и CHS, и LBA, режим адресации определяется для каждой команды битом L (бит 6) регистра DH; режимы могут чередоваться произвольным образом. В настоящее время устройства ATA используют **расширенную 48-битную адресацию** LBA, в которой адрес запрашиваемого сектора формируется из двух компонентов линейного адреса – базового 28-битного и расширенного 20-битного.

Устройства ATAPI используют принятую в SCSI 32-битную логическую адресацию, позволяющую адресовать до 2 Тбайт (при 512-байтном блоке).

Если к шине ATA подключено одно устройство, оно должно быть ведущим (Master). Если подключены два устройства, одно должно быть ведущим, другое – ведомым. О своей роли (ведущее или ведомое) устройства «узнают» с помощью предварительно установленных конфигурационных джамперов. Если применяется «кабельная выборка» (Cable select), роль устройства определяется его положением на ленточном кабеле. Оба устройства воспринимают команды от хост-адаптера одновременно. Однако исполнять команду будет лишь выбранное устройство.

2.1.3. Регистры устройства ATA

Каждое устройство ATA имеет стандартный набор регистров, адресуемых сигналами от хост-адаптера (CS0#, CS1#, DA00-DA02, DIOR# и DIOW#). Набор регистров состоит из двух блоков, выбираемых сигналами CS0# и CS1#, из которых активным (низкий уровень, «0») может быть только один. В таблице приведены адреса регистров в пространстве ввода-вывода IBM PC-совместимого ПК для первого и второго каналов ATA.

Таблица 2.1. Управляющие регистры, адресуемые по сигналам CS0# = 1, CS1# = 0

Адрес порта (канал 1)	Адрес порта (канал 2)	Назначение при чтении	Назначение при записи
0x3F6	0x376	AS – альтернативный статус	DC – управление устройством
0x3F7	0x377	DA – адрес устройства	

Таблица 2.2. Командные регистры, адресуемые по сигналам CS0#=0, CS1#=1

Адрес порта (канал 1)	Адрес порта (канал 2)	Назначение при чтении	Назначение при записи
0x1F0	0x170	DR – регистр данных	
0x1F1	0x171	ER – регистр ошибок	FR – регистр свойств
0x1F2	0x172	SC – счетчик секторов	
0x1F3	0x173	SN – номер сектора или LBA [7:0]	
0x1F4	0x174	CL – младший байт номера цилиндра или LBA [15:8]	
0x1F5	0x175	CH – старший байт номера цилиндра или LBA [23:16]	
0x1F6	0x176	DH – номер устройства и головки или LBA [27:24]	
0x1F7	0x177	SR – регистр состояния	CR – регистр команд

Блок командных регистров (табл. 2.2) служит для отправки команд устройству и чтения информации о его состоянии. **Блок управляющих регистров** (табл. 2.1) используется для управления устройством и получения более подробной информации о его состоянии. На действительность содержимого регистров командного блока и альтернативного регистра состояния указывает нулевое значение бита BSY регистра состояния (см. ниже). Запись в регистры должна производиться лишь при BSY=0 и DRQ=0, кроме особо оговоренных случаев. Если устройство поддерживает управление энергопотреблением, в «спящем» режиме содержимое этих регистров недействительно и запись игнорируется, кроме особо оговоренных случаев.

Альтернативный регистр состояния AS (для первого канала адрес 3F6h, для второго — 376h) имеет те же биты, что и основной (см. ниже), но его чтение не приводит ни к каким изменениям состояния устройства.

Регистр управления устройством DC служит для программного сброса обоих устройств одновременно и управления разрешением прерывания выбранного устройства. Запись в этот регистр возможна в любой момент. Программный сброс через регистр DC должен обрабатываться из состояния Sleep. Назначение битов регистра DC:

- биты [7:3] зарезервированы;
- бит 2 – SRST (Software Reset) – программный сброс, действует все время, пока бит не будет снят (оба устройства на шине воспринимают программный сброс одновременно);
- бит 1 – nIEN (Interrupt Enable) – инверсный бит разрешения прерывания (при нулевом значении бита выбранное устройство может вырабатывать сигнал INTRQ через тристабильный выход);
- бит 0 – 0.

Регистр адреса устройства DA использовался только в первой версии ATA для совместимости со старыми контроллерами, чтением этого регистра можно было определить адресованный привод и головку. Регистр выпадает из блока (он совпадает с диагностическим регистром состояния контроллера НГМД) и рекомендуется, чтобы устройство ATA не отвечало на чтение этого регистра. Если устройство отвечает на чтение, то оно не должно управлять битом DD7 во избежание конфликта с контроллером НГМД, у которого по этой линии передается бит смены носителя. Из-за несоблюдения этого требования могут возникать проблемы, когда контроллер (адаптер) ATA и контроллер НГМД находятся на разных платах. Назначение битов регистра DA:

- бит 7 — (HiZ) — высокоимпедансный, при считывании не выдается на шину;
- бит 6 — nWTG — инверсный признак записи (во время физического выполнения записи на носитель бит нулевой);

- биты [5:2] — nHS [3:0] — номер головки (инверсные биты);
- биты [1:0] — nDS [1:0] — выбор устройства (инверсные биты): 10 — выбрано устройство 0, 01 — выбрано устройство 1.

Регистр данных DR может использоваться как 8-битный или 16-битный в зависимости от типа данных, передаваемых в текущей команде. Обращение к этому регистру происходит в режиме обмена PIO (когда сигнал DMACK# неактивен); при выполнении передач протокола PO (PIO Out) хост производит запись в этот регистр, при PI (PIO In) — чтение. В режиме DMA обмен данными происходит через порт данных, при этом активны сигналы DMARQ и DMACK#, а сигналы CS0# и CS1# неактивны.

Регистр ошибок ER хранит состояние выполнения последней операции или диагностический код. После завершения операции на наличие ошибки указывает бит ERR регистра состояния SR.

Назначение битов регистра ER:

- бит 7 — зарезервирован;
- бит 6 — UNC (Uncorrectable Data Error) — неисправимая ошибка данных;
- бит 5 — MC (Media Changed) — смена носителя (после смены носителя первая команда обращения отвергается и устанавливается данный бит, после сброса бита следующие команды будут выполняться нормальным образом);
- бит 4 — IDNF (ID Not Found) — указывает на проблему поиска сектора с заданным идентификатором;
- бит 3 — MCR (Media Change Requested) — индикатор запроса смены носителя (после обнаружения запроса смены носителя команды Door Lock будут возвращать бит ошибки ERR и бит MCR, бит MCR сбрасывается командами Door Unlock, Media Eject или сигналом аппаратного сброса);
- бит 2 — ABRT (Aborted Command) — устанавливается, если команда отвергнута как недействительная или в случае возникновения иной ошибки;
- бит 1 — TK0NF (Track 0 Not Found) — указывает на то, что по команде Recalibrate не удалось найти нулевой трек;
- бит 0 — AMNF (Address Mark Not Found) — не найден адресный маркер данных в заголовке сектора.

После выполнения любого сброса или команды Execute Device Diagnostic регистр ошибок содержит диагностический код. Трактовка битов, за исключением бита 2 (ABRT), может меняться в зависимости от исполненной команды.

Регистр свойств FR используется в зависимости от команды. В команде Set Features через него задается код подкоманды. Некоторые старые устройства могут игнорировать запись в этот регистр. До принятия спецификации ATA-2 в этот регистр помещали значение рекомендуемого номера цилиндра для предкомпенсации записи.

Регистр счетчика секторов SC содержит число секторов, участвующих в обмене. Хост инициализирует этот регистр до подачи команды (нулевое значение соответствует 256 секторам). По успешному завершению операции доступа к данным регистр должен обнулиться. Если команда завершилась с ошибкой, в регистре будет число секторов, которые должны быть переданы для успешного завершения предыдущего запроса. Команды Initialize Device Parameters или Write Same могут переопределить значение этого регистра. В некоторых командах регистр используется для передачи иных параметров.

Регистры номера сектора SN и номера цилиндра — младшего CL и старшего байта CH — имеют двойное назначение в зависимости от выбранной системы адресации (CHS или LBA). Они инициализируются хост-адаптером, а в случае возникновения ошибки при выполнении операции устройство поместит в них адрес, по которому встретилась ошибка.

Регистр номера устройства и головки DH, кроме хранения части адресной информации, служит для выбора ведущего или ведомого устройства и метода адресации.

Назначение битов регистра DH:

- биты 7 и 5 вплоть до ATA-3 должны были быть единичными, в ATA/ATAPI-4 их объявили устаревшими;
- бит 6 —единичным значением указывает на применение режима адресации LBA, при нулевом значении бита используется режим CHS;
- бит 4 — DEV (Device) — выбор устройства, при DEV=0 выбрано ведущее, при DEV=1 — ведомое;
- биты [3:0] имеют двойное назначение в зависимости от выбранной системы адресации, в режиме CHS они содержат номер головки, в режиме LBA — старшие биты логического адреса.

Как и предыдущие, адресный регистр DH инициализируется хост-адаптером, а в случае возникновения ошибки при операции устройство поместит в них адрес, по которому встретилась ошибка. До принятия спецификации ATA-2 считалось, что адресные регистры должны модифицироваться и после успешного выполнения операции, отражая текущее значение адреса в носителе.

Регистр состояния SR отражает текущее состояние устройства в процессе выполнения команд: занятость, готовность, наличие ошибок и др. Чтение регистра состояния разрешает дальнейшее изменение его битов и сбрасывает запрос аппаратного прерывания. Назначение битов регистра SR:

- бит 7 — BSY (Busy) — указывает на занятость устройства, значение этого бита действительно всегда. При BSY=1 устройство игнорирует попытки записи в командный блок регистров, а чтение этих регистров дает неопределенный результат. При BSY=0 регистры командного блока доступны; в это время устройство не может устанавливать бит DRQ, изменять значение битов ERR и содержимое остальных командных регистров (могут меняться только значения битов IDX, DRDY, DF, DSC и CORR). Бит может устанавливаться на кратковременный интервал, так что хост может этого не заметить. Бит устанавливается:
 - при сбросе устройства;
 - по получении команды, если не устанавливается DRQ;
 - между передачами блоков данных в режиме PIO и после них, пока не обнулился DRQ;
 - во время передач данных в режиме DMA.
- бит 6 — DRDY (Device Ready) — указывает на готовность устройства к восприятию любых кодов команд. Если состояние бита изменилось, оно не может вернуться обратно до чтения регистра состояния. При DRDY=0 устройство воспринимает только команды Execute Device Diagnostic и Initialize Device Parameters, прекращая выполнение текущей команды и сообщая об этом флагом ABRT в регистре ошибок и флагом ERR в регистре состояния. Другие команды приводят к непредсказуемым результатам. Устройства ATAPI сбрасывают бит по любому сбросу и команде Execute Device Diagnostic. Бит устанавливается устройством ATA, когда оно готово к выполнению всех команд. Устройство ATAPI устанавливает бит до завершения выполнения команд, за исключением команд Device Reset и Execute Device Diagnostic.
- бит 5 — DF (Device Fault) — индикатор отказа устройства.
- бит 4 — DSC (Device Seek Complete) — индикатор завершения поиска трека. В командах, допускающих перекрытие, бит называется SERV (Service Required) — устройство требует обслуживания.
- бит 3 — DRQ (Data Request) — индикатор готовности к обмену словом или байтом данных.
- бит 2 — CORR (Corrected Data) — индикатор исправленной ошибки данных.
- бит 1 — IDX (Index) — индекс, трактуется особо каждым производителем.
- бит 0 — ERR (Error) — индикатор ошибки выполнения предыдущей операции.

Дополнительная информация содержится в регистре ошибок. Если установлен бит ERR, до приема следующей команды, программного или аппаратного сброса устройство не изменит состояние этого бита, а также регистра ошибок, регистра количества секторов и регистров цилиндра, головки и номера сектора. Для команд PacketService бит называется СНК и служит признаком исключительной ситуации.

В стандарте ATA/ATAPI-4 для некоторых команд биты 4 и 5 могут иметь иное назначение, а биты 1 и 2 объявлены устаревшими.

Назначение регистра команд CR очевидно из названия. Устройство начинает исполнять команду сразу, как только ее код записан в данный регистр. Команда Device Reset выполняется устройством ATA/ATAPI независимо от состояния битов BSY и DRQ, и даже в состоянии Sleep.

2.1.4. Протоколы обмена данными

Программа общается с устройствами ATA через регистры, используя инструкции ввода-вывода IN и OUT. Для передачи данных с максимальной скоростью применяют программный доступ PIO к регистру данных инструкциями INSW/OUTSW или по каналу DMA. Тип обмена (PIO или DMA) определяется командой обращения. Программный доступ PIO обязателен для всех устройств, команды режима DMA устройствами могут не поддерживаться.

Программный доступ PIO (Programmed Input/Output) выполняется в виде следующих друг за другом операций чтения или записи в пространстве ввода-вывода по адресу регистра данных. Готовность устройства проверяется перед началом передачи блока, после чего хост производит серию операций в определенном темпе, который определяется выбранным режимом PIO Mode 0-4. Для каждого режима определены допустимые параметры временной диаграммы цикла обмена. Обмен PIO программно реализуется с помощью процессорных инструкций ввода-вывода строк REP INS или REP OUTS с занесенным в регистр CX количеством слов (или байтов) в передаваемом блоке. Эти инструкции обеспечивают максимально возможную скорость обмена для данного процессора и системной шины. «Обуздать» процессор в соответствии с выбранным режимом входит в задачу адаптера ATA, который использует для удлинения цикла сигнал готовности шины. Традиционные режимы 0, 1 и 2 имеют временные параметры, фиксируемые только хост-адаптером. Для прогрессивных режимов ATA-2 (PIO Mode 3 и выше) устройство может затормозить обмен, используя сигнал готовности IORDY. Программный обмен на все время передачи блока занимает и процессор, и системную шину.

Обмен по каналу DMA занимает исключительно шины ввода-вывода и памяти. Процессору требуется выполнить только процедуру инициализации канала, после чего он свободен до прерывания от устройства в конце передачи блока (этим могут воспользоваться многозадачные системы). Стандартные каналы DMA шины ISA для интерфейса ATA не используются из-за низкой пропускной способности. Высокопроизводительные адаптеры ATA имеют собственные более эффективные контроллеры. Режимы обмена по каналу DMA бывают одиночными и множественными. При одиночном режиме — Singleword DMA — устройство для передачи каждого слова вырабатывает сигнал запроса DMARQ и сбрасывает его по сигналу DMACK#, подтверждающему цикл обмена. При множественном режиме — Multiword DMA — на сигнал DMARQ хост отвечает потоком циклов, сопровождаемых сигналами DMACK#. Если устройство не справляется с потоком, оно может приостановить его снятием сигнала DMARQ, а по готовности установить его снова. Множественный режим позволяет развить более высокую скорость передачи.

В спецификации ATA/ATAPI-4 появился новый режим — Ultra DMA, позволяющий перешагнуть барьер в 16,6 Мбайт/с, свойственный традиционным режимам и

используемому кабелю. При этом обеспечивается и контроль достоверности передачи данных по шине, чего не делалось ни в PIO, ни в стандартных режимах DMA. Стандартом ATA-4 было определено три режима Ultra DMA (0, 1 и 2), впоследствии ввели новые; выбор режима осуществляется командой Set Features. Каждое переданное слово участвует в подсчете CRC-кода, который передается хост-контроллером в конце пакета. Подсчет ведется и источником данных, и приемником. При несовпадении принятого устройством кода с ожидаемым кодом фиксируется ошибка передачи, о которой устройство сообщает в конце исполнения команды. Передача в пакете может приостановиться, если приемник снимет сигнал готовности (DDMARDY# или HDMARDY#). Передача пакета может прекращаться по инициативе устройства (снятием сигнала) или хоста (сигналом STOP). Противоположная сторона должна подтвердить окончание цикла сигналом STOP или DMARQ соответственно.

Протокол взаимодействия хоста с устройством выглядит следующим образом.

- 1) Хост читает регистр состояния устройства, дожидаясь нулевого значения бита BSY. Если присутствуют два устройства, хост обращается к ним «наугад» — состояние будет сообщаться последним выбранным устройством.
- 2) Дождавшись освобождения устройства, хост записывает в регистр DH байт, у которого бит DEV указывает на адресуемое устройство. Здесь кроется причина невозможности параллельной работы двух устройств на одной шине ATA: обратиться к устройству можно только после освобождения обоих устройств.
- 3) Хост читает основной или альтернативный регистр состояния адресованного устройства, дожидаясь признака готовности (DRDY=1), который свидетельствует о том, что адресуемое устройство присутствует и способно воспринимать команды.
- 4) Хост заносит требуемые параметры в блок командных регистров (адрес сектора, код подкоманды, другие требуемые параметры, кроме кода команды).
- 5) Хост записывает код команды в регистр команд (CR).
- 6) Устройство устанавливает бит BSY и переходит к исполнению команды.

Дальнейшие действия зависят от протокола передачи данных, заданного командой.

Для команд, не требующих передачи данных, следующий шаг (шаг 7) — последний. Завершив исполнение команды, устройство сбрасывает бит BSY и устанавливает запрос прерывания (если он не запрещен). К этому моменту в регистрах состояния и ошибок уже имеется информация о результате исполнения. Единичное значение бита BSY может «промелькнуть» между шагами 6 и 7 так быстро, что хост его не зафиксирует, но для фиксации факта выполнения команды или ее части предназначен запрос прерывания.

Для команд, требующих **чтения данных** в режиме PIO, процедура продолжается следующим образом:

- 1) Подготовившись к передаче первого блока данных по шине ATA, устройство устанавливает бит DRQ. Если была ошибка, она фиксируется в регистрах состояния и ошибок. Далее устройство сбрасывает бит BSY и устанавливает запрос прерывания (если он не запрещен).
- 2) Зафиксировав обнуление бита BSY (или по прерыванию), хост считывает регистр состояния, что приводит к сбросу прерывания от устройства.
- 3) Если хост обнаружил единичное значение бита DRQ, он производит чтение первого блока данных в режиме PIO (адресуясь к регистру данных). Если обнаружена ошибка, считанные данные могут быть недостоверными. После передачи блока данных возможно одно из следующих действий:
 - a. если на шаге 8 ошибка не обнаружена и требуется передача следующего блока, устройство устанавливает бит BSY и данная последовательность повторяется с шага 7;
 - b. если есть ошибка или передан последний блок данных, устройство сбрасывает бит DRQ и выполнение команды завершается.

Для операций записи данных после шага 6 для устройства начинается активная фаза записи на носитель, что отмечается установкой бита BSY.

Для команд, требующих **записи данных** в режиме PIO, процедура после шага 6 продолжается следующим образом:

- 1) Подготовившись к приему первого блока данных по шине ATA, устройство устанавливает бит DRQ (если нет ошибок) и сбрасывает бит BSY. Если была ошибка, она фиксируется.
- 2) Зафиксировав обнуление бита BSY, хост считывает регистр состояния.
- 3) Если хост обнаружил единичное значение бита DRQ, он производит запись первого блока данных в режиме PIO по адресу в регистре данных.
- 4) После передачи блока данных возможно одно из следующих действий:
 - если обнаружена ошибка, устройство сбрасывает бит DRQ, устанавливает запрос прерывания и выполнение команды завершается, а переданные по шине данные остаются устройством необработанными (не записываются на носитель);
 - если ошибка не обнаружена, устройство устанавливает бит BSY и переходит к следующему шагу.
- 5) Устройство обрабатывает принятый блок данных, затем:
 - если нет ошибок и обработанный блок — последний, устройство сбрасывает бит BSY и устанавливает запрос прерывания, на чем выполнение команды успешно завершается;
 - если обнаружена ошибка, выполнение команды завершается таким же образом, но с установкой битов ошибок;
 - если нет ошибок, и требуется передача следующего блока, процедура продолжается.
- 6) По готовности приема следующего блока устройство устанавливает бит DRQ, сбрасывает бит BSY и устанавливает запрос прерывания.
- 7) По обнулении бита BSY (или по прерыванию) хост считывает регистр состояния.
- 8) Обнаружив бит DRQ, хост выполняет запись очередного блока в регистр данных, и последовательность повторяется с шага 3.

Команды с передачей данных в режиме DMA выполняются похожим образом, но с некоторыми исключениями.

1. Вместо PIO используется прямой доступ к памяти. Хост должен инициализировать канал DMA до записи кода в регистр команд, чтобы при появлении сигнала DMARQ начался обмен.
2. Запрос прерывания даже в многосекторных передачах производится один раз — по выполнении команды.