

Министерство образования Республики Беларусь

Учреждение образования  
«Белорусский государственный университет  
информатики и радиоэлектроники»

Кафедра электронных вычислительных машин

М.А. Курмаз

## ЛАБОРАТОРНЫЙ ПРАКТИКУМ

по дисциплине «Периферийные устройства ЭВМ»  
с разделом "Интерфейсы ЭВМ"  
для студентов специальности I-40 02 01  
«Вычислительные машины, системы и сети»  
всех форм обучения

Минск 2007

## Содержание

Лабораторная работа №1. Локальная шина подключения периферийных устройств PCI. Конфигурирование устройств. ....	3
1.1. Краткие теоретические сведения .....	3
1.1.1. Основные сведения о шине PCI .....	3
1.1.2. Программный доступ к конфигурационному пространству .....	5
1.1.3. Конфигурационное пространство устройств PCI.....	5
1.2. Задание на лабораторную работу.....	7
1.3. Порядок выполнения работы .....	7
1.4. Содержание отчета .....	7
1.5. Контрольные вопросы.....	8
Лабораторная работа №2. Интерфейс подключения жестких дисков ATA. Команда идентификации устройства .....	9
2.1. Краткие теоретические сведения .....	9
2.1.1. Основные сведения об интерфейсе ATA/ATAPI .....	9
2.1.2. Адресация данных в интерфейсе ATA .....	9
2.1.3. Регистры устройства ATA .....	10
2.1.4. Протоколы обмена данными .....	14
2.2. Задание на лабораторную работу.....	16
2.3. Порядок выполнения работы .....	16
2.4. Содержание отчета .....	19
2.5. Контрольные вопросы.....	20
Лабораторная работа №3. Планшетные сканеры изображения. Изучение двух конкурирующих технологий изготовления светочувствительного датчика. ....	20
3.1. Краткие теоретические сведения .....	20
3.1.1. Сканер. Конструктивное исполнение планшетного сканера .....	20
3.1.2. Светочувствительный датчик типа CCD.....	21
3.1.3. Светочувствительный датчик типа CIS.....	22
3.1.4. О параметрах качества изображения .....	22
3.2. Задание на лабораторную работу.....	23
3.3. Порядок выполнения работы .....	23
3.4. Содержание отчета .....	23
3.5. Контрольные вопросы.....	23
Лабораторная работа №4. Интерфейс IEEE1394 (FireWire).....	25
4.1. Краткие теоретические сведения .....	25
4.1.1. Основные сведения об интерфейсе IEEE 1394.....	25
4.1.2. Пакет IOMeter – профессиональный инструмент для измерения эффективности подсистемы ввода-вывода .....	26
4.1.3. Принцип работы пакета IOMeter .....	26
4.2. Задание на лабораторную работу.....	30
4.3. Порядок выполнения работы .....	30
4.4. Содержание отчета .....	31
4.5. Контрольные вопросы.....	31

## Лабораторная работа №1. Локальная шина подключения периферийных устройств PCI. Конфигурирование устройств.

Цель работы: Изучить способы адресации устройств PCI. Отработать метод опроса, идентификации и конфигурирования устройств.

### 1.1. Краткие теоретические сведения

#### 1.1.1. Основные сведения о шине PCI

**PCI (Peripheral Component Interconnect)** – шина соединения периферийных компонентов, является основной шиной расширения современных компьютеров. Она разрабатывалась в расчете на платформу Intel Pentium, но нашла применение и на более современных платформах. Первая версия PCI 1.0 появилась в 1992 г. В настоящее время развитие шины PCI приостановлено ввиду появления периферийной шины нового поколения – PCI Express. Последней официальной версией, изданной курирующей организацией PCI SIG (Special Interest Group), является версия PCI 3.0. Это полностью переработанная спецификация шины, в которую внесены некоторые изменения, например, исключена поддержка устройств с напряжением питания 5 В.

Поначалу шина PCI вводилась как пристройка (mezzanine bus) к системам с основной шиной ISA, став позже центральной шиной: она соединяется с системной шиной процессора высокопроизводительным мостом (северным), входящим в состав чипсета системной платы. Остальные шины расширения ввода-вывода (ISA/EISA или MCA), а также локальная ISA-подобная шина X-BUS и интерфейс LPC, к которым подключаются микросхемы системной платы (ROM BIOS, контроллеры прерываний, клавиатуры, DMA, портов COM и LPT, НГМД и прочие «мелочи»), подключаются к шине PCI через «южный» мост. В современных системных платах с хабовой архитектурой шину PCI отодвинули на периферию, не ущемляя ее в мощности канала связи с процессором и памятью, но и не нагружая транзитным трафиком устройств других шин.

Шина является **синхронной** — фиксация всех сигналов выполняется по положительному перепаду (фронту) сигнала CLK. Номинальной **частотой синхронизации** считается 33 МГц; начиная с версии 2.1, допускается повышение частоты до 66 МГц при поддержке всех устройств на шине (режим PCI-66).

Номинальная **разрядность** шины данных — 32 бита, спецификация определяет возможность расширения разрядности до 64 бит. При частоте шины 33 МГц теоретическая **пропускная способность** достигает 132 Мбайт/с для 32-битной шины и 264 Мбайт/с для 64-битной; при частоте синхронизации 66 МГц – 264 и 528 соответственно. Однако эти пиковые значения достигаются лишь во время передачи пакета, а из-за протокольных накладных расходов реальная средняя суммарная (для всех задатчиков) пропускная способность шины оказывается ниже.

С устройствами PCI процессор может **взаимодействовать** командами обращения к памяти (MOV и т.д.) и портам ввода-вывода (IN/OUT), адресованным к областям, выделенным каждому такому устройству при конфигурировании. Устройства могут вырабатывать запросы маскируемых и немаскируемых прерываний. Понятия каналов DMA для шины PCI нет, но агент шины может сам выступать в роли задатчика, поддерживая высокопроизводительный обмен с памятью (и не только) и не занимая ресурсов центрального процессора. Таким образом, к примеру, может быть реализован обмен в режиме DMA с устройствами АТА, подключенными к контролеру PCI.

Спецификация PCI требует от устройств способности **перемещать** все занимаемые ресурсы в пределах доступного пространства адресации. Это позволяет обеспечивать бесконфликтное распределение ресурсов для многих устройств (функций). Для управления устройствами рекомендуется вместо портов ввода-вывода по возможности

использовать ячейки памяти. Одно и то же функциональное устройство может быть сконфигурировано по-разному, отображая свои регистры либо на пространство памяти, либо на пространство ввода-вывода. Драйвер может определить текущую настройку, прочитав содержимое регистра базового адреса устройства. Драйвер также может определить и номер запроса прерывания, который используется устройством.

Для шины PCI принята **иерархия** понятий адресации: шина, устройство, функция. Эти понятия фигурируют только при обращении к регистрам конфигурационного пространства. К этим регистрам обращаются на этапе конфигурирования — переучета обнаруженных устройств, выделения им непересекающихся ресурсов (областей памяти и пространства ввода-вывода) и назначения номеров аппаратных прерываний. При дальнейшей регулярной работе устройства будут отзываться на обращения по назначенным им адресам памяти и ввода-вывода, доведенным до сведения связанных с ними модулей ПО. Эти адреса принимаются с шины AD в начале каждой транзакции. Для доступа к конфигурационному пространству используются отдельные линии IDSEL.

**Устройством PCI** называется микросхема или карта расширения, подключенная к одной из шин PCI и использующая для идентификации выделенную ей линию IDSEL, принадлежащую этой шине. Устройство может быть многофункциональным, то есть состоять из множества (от 1 до 8) так называемых **функций**. Каждой функции отводится конфигурационное пространство в 256 байт. Многофункциональные устройства должны отзываться только на конфигурационные циклы с номерами функций, для которых имеется конфигурационное пространство. При этом функция с номером 0 должна быть обязательно, номера остальных функций назначаются разработчиком устройства произвольно (в диапазоне 1-7). Простые (однофункциональные) устройства, в зависимости от реализации, могут отзываться либо на любой номер функции, либо только на номер функции 0.

**Шина PCI** – набор сигнальных линий, непосредственно соединяющих интерфейсные выводы группы устройств (слотов, микросхем на системной плате). В системе может присутствовать несколько шин PCI, соединенных мостами PCI. Мосты электрически отделяют интерфейсные сигналы одной шины от другой, соединяя их логически; главный мост соединяет главную шину с ядром системы (процессором и памятью). Каждая шина имеет свой **номер шины** (PCI bus number). Шины нумеруются последовательно; главная шина имеет нулевой номер.

С точки зрения конфигурирования, минимальной адресуемой единицей этой иерархии является **функция**; ее полный адрес состоит из трех частей: номера шины, номера устройства и номера функции. Короткая форма идентификации вида PCI 0:1:2 (например, в сообщениях ОС Unix) означает функцию 2 устройства 1, подключенного к главной (0) шине PCI.

В шине PCI принята **географическая адресация** — номер устройства определяется местом его подключения. Номер устройства (device number или dev) определяется той линией шины AD, к которой подключена линия сигнала IDSEL данного слота. Часто для слотов используются убывающие номера устройств, начиная с 20. Группы соседних слотов могут подключаться к разным шинам; на каждой шине PCI нумерация устройств независимая (могут быть и устройства с совпадающими номерами dev, но разными номерами шин). Устройства PCI, интегрированные в системную плату, используют ту же систему адресации. Их номера «запаяны намертво», в то время как адреса карт расширения можно изменять перестановкой их в разные слоты. Одна карта PCI может содержать только одно устройство шины, к которой она подключается, поскольку ей в слоте выделяется только одна линия IDSEL. Если на карте размещают несколько устройств (например, 4-портовая карта Ethernet), то на ней приходится устанавливать мост – тоже устройство PCI, к которому и обращаются по линии IDSEL, выделенной данной карте. Этот мост организует на карте дополнительную шину PCI, к которой можно подключить множество устройств.

### 1.1.2. Программный доступ к конфигурационному пространству

Поскольку конфигурационное пространство PCI обособлено, в главный мост приходится вводить **специальный механизм** доступа к нему командами процессора, который «умеет» обращаться только к памяти или вводу-выводу. Этот же механизм используется и для генерации специальных циклов. Для PC-совместимых компьютеров предусмотрено два механизма, из которых в спецификации PCI, начиная с версии 2.2, оставлен только первый (Configuration Mechanism #1). Номер механизма, которым пользуется конкретная системная плата, можно узнать путем вызова PCI BIOS.

**Конфигурационные циклы** адресуются к конкретному устройству (микросхеме PCI), для которого должен быть сформирован сигнал выборки IDSEL (единичное значение). Номер функции и адрес регистра декодируется самим устройством. Поскольку сигнал IDSEL воспринимается устройством только в фазе адреса, для него используют позиционное кодирование в линиях старших битов шины AD (конфигурационное пространство всех устройств занимает лишь малую часть пространства с 32-битной адресацией). На этих линиях в фазе адреса конфигурационного обращения может быть лишь один единичный бит, остальные — нулевые. Таким образом, только одно устройство будет выбрано сигналом IDSEL.

Для работы механизма № 1 в пространстве ввода-вывода зарезервированы 32-битные **порты** с адресами 0x0CF8 и 0x0CFC, входящие в состав главного моста. Для обращения к конфигурационному пространству в порт CONFIG\_ADDRESS (RW, адрес 0x0CF8) заносят 32-разрядный адрес, декодируемый в соответствии с табл. 1.1. После занесения адреса обращением к порту CONFIG\_DATA (RW, адрес 0x0CFC) можно прочитать или записать содержимое требуемого конфигурационного регистра.

31	30	24	23	16	15	11	10	8	7	2	1	0
1	0000000		номер шины		номер устройства		номер функции		адрес регистра		0	0

Таблица 1.1. Формат адреса для вызова конфигурационного цикла (механизм №1)

Если конфигурационный цикл не воспринимается ни одним из устройств, мосты могут эту ситуацию отрабатывать двояко: фиксировать отсутствие устройства (сработает Master Abort) или же выполнять операции вхолостую. Однако в любом случае чтение конфигурационного регистра несуществующего устройства (функции) должно возвращать значение 0x0FFFFFFF (это будет безопасной информацией, поскольку даст недопустимое значение идентификатора устройства). Если главный мост подключает несколько равноправных шин (peer buses), то одна из них назначается условно главной, так что вышеописанная логика сохраняется.

### 1.1.3. Конфигурационное пространство устройств PCI

В стандарт PCI заложены возможности автоматического конфигурирования системных ресурсов (пространств памяти и ввода-вывода и линий запроса прерываний). Автоматическое конфигурирование устройств (выбор адресов и прерываний) поддерживается средствами BIOS и ориентировано на технологию PnP.

Стандарт PCI определяет для каждого слота **конфигурационное пространство** размером до 256 регистров (8-битных), 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 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. 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.

смещение	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	

Таблица 1.2. Конфигурационное пространство устройства PCI, обязательные поля

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.
- Class Code — код класса, определяющий основную функцию устройства, а иногда и его программный интерфейс. Старший байт (адрес 0x0B) определяет базовый класс, средний — подкласс, младший — программный интерфейс (если он стандартизован).

Остальные поля заголовка являются регистрами устройств, допускающими как запись, так и чтение.

## 1.2. Задание на лабораторную работу

1. Изучить теоретические сведения о механизме конфигурирования устройств PCI и способам доступа к конфигурационному устройству.
2. Разработать программу формирования конфигурационного цикла для опроса и идентификации устройств PCI, которая будет считывать два первых поля конфигурационного пространства – коды Vendor ID (производитель) и Device ID (устройство).
3. Используя прилагаемый файл PCI\_DEVS.TXT, произвести расшифровку наименований производителей и устройств.
4. Результатом работы программы должна быть выводимая на экран или в файл таблица, содержащая следующую информацию:
  - 4.1. адрес устройства (номер шины, устройства и функции);
  - 4.2. 16-разрядный код производителя (в hex-формате);
  - 4.3. 16-разрядный код устройства (в hex-формате);
  - 4.4. производитель и название устройства.

## 1.3. Порядок выполнения работы

Программу рекомендуется разрабатывать на языке программирования C/C++ с применением вставок на языке ассемблера. Для обеспечения возможности доступа к портам ввода-вывода следует запускать программу только в среде операционной системы DOS.

Разбор файла PCI\_DEVS.TXT удобно проводить с помощью стандартных средств языка C/C++, в частности, функций типа fscanf/sscanf.

Файл PCI\_DEVS.TXT состоит из текстовых строк следующего формата:

char code	\t	Unsigned int ID	\t	char * string	\n
-----------	----	-----------------	----	---------------	----

где:

- code – признак кода Vendor ('V'), Device ('D'), Subsystem ('S') и т.д.;
- ID – код в hex-формате (0000-FFFF), зарезервированный за определенным производителем или устройством;
- string – строка, содержащая название устройства или производителя.

Соответствующая строка-шаблон для функции fscanf/sscanf будет выглядеть следующим образом:

```
"%c\t%04X\t%[^\\n]\\n"
```

Файл PCI\_DEVS.TXT организован по иерархическому принципу: после кода производителя идут коды устройств, им выпускаемых; после кода устройства могут идти уточняющие коды для ревизий, подсистем (плат или устройств, в состав которых входит данное устройство) и т.д.

## 1.4. Содержание отчета

- 1) Титульный лист.
- 2) Исходный код программы.
- 3) Результат работы программы.

### 1.5. Контрольные вопросы

- 1) Для чего предназначена шина PCI? Какова ее роль в архитектуре современного ПК?
- 2) Каковы основные характеристики шины PCI?
- 3) Что такое устройство PCI?
- 4) Каким образом выполняется доступ к устройству PCI со стороны центрального процессора?
- 5) Поддерживают ли устройства PCI обмен по механизму DMA?
- 6) Для чего устройства PCI должны поддерживать возможность перемещения ресурсов в пределах общего пространства памяти и ввода-вывода?
- 7) Отображается ли конфигурационное пространство устройств PCI на общее пространство памяти и ввода-вывода?
- 8) Какой программный механизм позволяет провести конфигурирование устройств?
- 9) Какие поля имеются в конфигурационном пространстве устройств PCI?
- 10) В каких случаях необходимо обращение к устройствам PCI с помощью конфигурационного цикла и географической адресации?



## Лабораторная работа №2. Интерфейс подключения жестких дисков АТА. Команда идентификации устройства

Цель работы: Изучить программный интерфейс доступа к жестким дискам. Отработать механизм идентификации устройств с помощью команды IDENTIFY DEVICE.

### 2.1. Краткие теоретические сведения

#### 2.1.1. Основные сведения об интерфейсе АТА/АТАPI

Интерфейс АТА (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). Стандартный контроллер АТ позволял подключать до двух накопителей, что в интерфейсе АТА означает параллельное подключение контроллеров двух устройств. В спецификации АТА фигурируют следующие компоненты.

- Хост-адаптер — средства сопряжения интерфейса АТА с шиной компьютера. Хостом мы будем называть компьютер с хост-адаптером интерфейса АТА. Хост-контроллер – это более развитый вариант хост-адаптера.
- Ведущее устройство (Master) – устройство, в спецификации АТА официально называемое Device0 (устройство 0).
- 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). 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, способное поддерживать режим выбора устройства через вышеупомянутый бит. Принятая система команд и регистров, являющаяся частью спецификации АТА, ориентирована на блочный обмен данными с устройствами прямого доступа. Для иных устройств существует спецификация АТАPI, основанная на тех же аппаратных средствах, но позволяющая обмениваться пакетами управляющей информации (Package Interface, PI). Структура и наполнение пакетов позаимствованы из универсального интерфейса SCSI. Пакетный интерфейс позволяет расширить границы применения шины АТА.

#### 2.1.2. Адресация данных в интерфейсе АТА

Адресация в АТА 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 АТА стали описывать внешней геометрией, отличающейся от реальной внутренней. Например, разные зоны треков имеют разное число секторов, причем часть секторов может резервироваться на случай замены дефектных. При этом одно и то же устройство может иметь различную внешнюю геометрию. Преобразование адресов в реальные выполняется встроенным контроллером устройства. В системе CHS устройство АТА позволяет адресовать до 267 386 880 (65536x16x255) секторов (блоков), что при размере сектора в 512 байт дает 136 902 082 560 байт (около 137 Гбайт).

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) блоков – немного больше, чем в CHS. Для устройств АТА, поддерживающих и CHS, и LBA, режим адресации определяется для каждой команды битом L (бит 6) регистра D/H; режимы могут чередоваться произвольным образом. В настоящее время устройства АТА используют **расширенную 48-битную адресацию** LBA, в которой адрес запрашиваемого сектора формируется из двух компонентов – базового 28-битного и расширенного 20-битного.

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

Если к шине АТА подключено одно устройство, оно должно быть ведущим. 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», роль устройства определяется его положением на специальном ленточном кабеле. Оба устройства воспринимают команды от хост-адаптера одновременно. Однако исполнять команду будет лишь выбранное устройство. Выводить выходные сигналы на шину АТА имеет право только выбранное устройство. Такая система подразумевает, что, начав операцию обмена с одним из устройств, хост-адаптер не может переключиться на обслуживание другого до завершения начатой операции. Параллельно могут работать только устройства IDE, подключаемые к разным шинам (каналам) АТА. Спецификация АТА-4 определяет способ обхода этого ограничения, но эту возможность используют редко.

### 2.1.3. Регистры устройства АТА

Каждое устройство АТА имеет стандартный набор регистров, адресуемых сигналами от хост-адаптера (CS0#, CS1#, DA2, DA1, OAO, 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 АТА.

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

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]	
0x1F7	0x176	DH – номер устройства и головки или LBA [27:24]	
0x1F7	0x177	SR – регистр состояния	CR – регистр команд

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

**Блок командных регистров** (Табл. 2.2) служит для посылки 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. 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. Трактровка 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 двоякое 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 обнулится 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 PacketnService 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 DN 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).
- 4) Хост заносит требуемые параметры в блок командных регистров.
- 5) Хост записывает код команды в регистр команд.
- 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:

- 7) 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).
- 8) Зафиксировав обнуление бита BSY (или по прерыванию), хост считывает регистр состояния, что приводит к сбросу прерывания от устройства.
- 9) 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:
  - если на шаге 8 ошибка не обнаружена и требуется передача следующего блока, устройство устанавливает бит BSY и данная последовательность повторяется с шага 7;
  - 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:

- 7) 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.
- 8) Зафиксировав обнуление бита BSY, хост считывает регистр состояния.

- 9) Если хост обнаружил единичное значение бита DRQ, он производит запись первого блока данных в режиме PIO по адресу в регистре данных.
- 10) 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.
- 11) 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.
- 12) По готовности приема следующего блока устройство устанавливает бит DRQ, сбрасывает бит BSY и устанавливает запрос прерывания.
- 13) По обнулении бита BSY (или по прерыванию) хост считывает регистр состояния.
- 14) Обнаружив бит DRQ, хост выполняет запись очередного блока в регистр данных, и последовательность повторяется с шага 11.

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

- 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.
- Запрос прерывания даже в многосекторных передачах производится один раз — по выполнении команды.

## 2.2. Задание на лабораторную работу

- 1) Разработать программу, использующую механизм PIO для доступа к жесткому диску, подключенному с помощью интерфейса ATA.
- 2) В качестве тестовой команды, не разрушающей информацию на жестком диске в случае неправильного задания параметров, использовать команду IDENTIFY DEVICE.
- 3) Результатом работы программы должна быть таблица с данными обо всех жестких дисках, подключенных по интерфейсу ATA:
  - a. Модель жесткого диска.
  - b. Серийный номер.
  - c. Версия прошивки (firmware).
  - d. Список поддерживаемых режимов PIO, DMA и UDMA.
  - e. Версия спецификации стандарта ATA/ATAPI, которой устройство соответствует.
  - f. Объем жесткого диска в байтах.

## 2.3. Порядок выполнения работы

Программу рекомендуется разрабатывать на языке программирования C/C++ с применением вставок на языке ассемблера. Для обеспечения возможности доступа к



портам ввода-вывода следует запускать программу только в среде операционной системы DOS.

Команда IDENTIFY DEVICE для устройств ATA (не ATAPI) имеет следующий код: 0x0EC. Эта команда относится к категории команд чтения, после ее вызова устройство пересылает хосту блок конфигурационных данных («паспорт диска») объемом 512 байт (один сектор), или 256 слов. Формат блока приведен в таблице 2.3.

Номер слова	Описание
0	Общая информация о конфигурации устройства: бит 0 - зарезервирован; биты 1-5 - определяются изготовителем; бит 6 – 1 - устройство со сменным носителем, 0 – нет; бит 7 – 1 - устройство с несменным носителем, 1 – нет; биты 8-14 - определяются изготовителем; бит 15 - тип интерфейса (0 – ATA, 1 – ATAPI)
1	Количество цилиндров для адресации CHS
2	Зарезервировано
3	Количество головок для адресации CHS
4	Определяется изготовителем
5	Определяется изготовителем
6	Количество секторов на дорожке для адресации CHS
7-9	Определяются изготовителем
10-19	Серийный номер (20 ASCII-символов)
20	Определяется изготовителем
21	Определяется изготовителем
22	Число байтов (определяемых изготовителем) в командах «длинного» чтения и записи
23-26	Версия прошивки (8 ASCII-символов)
27-46	Наименование модели устройства (40 ASCII-символов)
47	Биты 0-7 - значение 0 зарезервировано, значения 0x01-0xFF соответствуют максимальному количеству секторов, которое можно передавать в командах группового чтения и записи; Биты 8-15 - определяются изготовителем.
48	Зарезервировано
49	Возможности устройства Биты 0-7 - определяются изготовителем; Биты 8-9 - не используются; Бит 10 - 1 - сигнал IORDY может быть запрещен; Бит 11 - поддержка IORDY (0 - может отсутствовать, 1 - присутствует); Бит 12 – зарезервирован; Бит 13 - значения таймера Standby задаются изготовителем устройства (0) или текущим стандартом интерфейса ATA (1); Биты 14-15 – зарезервированы.
50	Зарезервировано
51	Биты 0-7 - определяются изготовителем Биты 8-15 - длительность цикла передачи данных в режиме PIO
52	Биты 0-7 - определяются изготовителем Биты 8-15 - зарезервированы
53	Бит 0 - значения в словах 54-58: 0 – не действительны, 1 – действительны Бит 0 - значения в словах 64-70: 0 - не действительны; 1 - действительны Биты 2-15 -зарезервированы

54	Текущее число цилиндров (CHS)
55	Текущее число головок (CHS)
56	Текущее число секторов на дорожке (CHS)
57-58	Текущая емкость в секторах (CHS)
59	Биты 0-7 - текущее количество секторов, которое может быть передано за одно прерывание в групповых операциях чтения/записи; Бит 8- параметры групповых операций действительны; Биты 9-15 –зарезервированы;
60-61	Текущая емкость в секторах (LBA)
62	Зарезервировано
63	Поддерживаемые режимы Multiword DMA (режим поддерживается, если соответствующий бит установлен в 1): бит 0 - поддержка режима 0; бит 1 - поддержка режима 1; бит 2 - поддержка режима 2; бит 3 - поддержка режима 3; бит 4 - поддержка режима 4; бит 5 - поддержка режима 5; бит 6 - поддержка режима 6; бит 7 - поддержка режима 7. Активные режимы Multiword DMA: бит 8 - активность режима 0; бит 9 - активность режима 1; бит 10 - активность режима 2; бит 11 - активность режима 3; бит 12 - активность режима 4; бит 13 - активность режима 5; бит 14 - активность режима 6; бит 15 - активность режима 7.
64	Поддерживаемые улучшенные режимы PIO бит 0- поддержка режима 3; бит 1 -поддержка режима 4; Биты 8-15 – зарезервированы.
65	Минимальное время цикла передачи слова Multiword DMA в наносекундах
66	Рекомендованное изготовителем время цикла передачи Multiword DMA в наносекундах
67	Минимальное время цикла передачи PIO (без проверки сигнала готовности) в наносекундах
68	Минимальное время цикла передачи PIO (с проверкой сигнала готовности) в наносекундах
69-79	Зарезервировано для будущих команд, реализующих очереди и перекрытие

80	Поддерживаемые основные версии АТА (версия поддерживается, если соответствующий бит установлен в 1 ; если слово содержит значения 0x0000 или 0xFFFF, то его значение не действительно): бит 0 - зарезервирован; бит 1 - поддерживается АТА-1; бит 2 - поддерживается АТА-2; бит 3 - поддерживается АТА-3; бит 4 - поддерживается АТА-4; бит 5 - поддерживается АТА-5; бит 6 - поддерживается АТА-6; бит 7 - поддерживается АТА-7; бит 8 - зарезервирован для АТА-8; бит 9 - зарезервирован для АТА-9; бит 10 - зарезервирован для АТА-10; бит 11 - зарезервирован для АТА-11; бит 12 - зарезервирован для АТА-12; бит 13 - зарезервирован для АТА-13; бит 14 - зарезервирован для АТА-14; бит 15 – зарезервирован.
81	Номер реализации версии АТА (если в поле записано значение 0x0000 или 0xFFFF, то поле не действительно)
82	Поддерживаемый набор команд (набор поддерживается, если соответствующий бит установлен в 1; если в словах 82 и 83 записаны значения 0x0000 или 0xFFFF, то поле не действительно): бит 0 - набор команд Smart; бит 1 - набор команд шифрования и ограничения доступа; бит 2 - набор команд для устройств со сменными носителями; бит 3 - набор команд управления энергопотреблением; биты 4-15 - зарезервированы
83	Признак достоверности слова 82: биты 0-13 - зарезервированы; бит 14 - должен быть установлен в 1; бит 15 - должен быть сброшен в 0.
84-127	Зарезервировано
128	Статус секретности: бит 0 - поддержка секретности (0 - нет, 1 - есть); бит 1 - использование секретности (0 - запрещено, 1 - разрешено); бит 2 - блокировка секретности (0 - нет, 1 - есть); бит 3 - приостановка секретности (0 - нет, 1 - есть); бит 4 - счетчик секретности (0 - нет, 1 - есть); биты 5-7 - зарезервированы; бит 8 - уровень секретности (0 - низкий, 1 - высокий); биты 9-15 - зарезервированы
129-159	Определяется изготовителем

## 2.4. Содержание отчета

- 4) Титульный лист.
- 5) Исходный код программы.
- 6) Результат работы программы.

## 2.5. Контрольные вопросы

- 11) Сколько устройств можно подключить по интерфейсу ATA?
- 12) Возможна ли работа с двумя (тремя и т.д.) устройствами ATA одновременно, то есть доступ к одному устройству, когда второе занято выполнением команды? Почему?
- 13) Какие способы адресации применяются при работе с жесткими дисками?
- 14) В чем специфика работы с устройствами ATAPI?
- 15) Каков протокол доступа к устройству ATA?
- 16) Какие регистры необходимо проверять в процессе работы с устройством ATA?
- 17) Какие существуют режимы обмена данными?
- 18) Что принимает и возвращает команда IDENTIFY DEVICE?

## Лабораторная работа №3. Планшетные сканеры изображения. Изучение двух конкурирующих технологий изготовления светочувствительного датчика.

Цель работы: Ознакомиться с принципом действия и основными характеристиками планшетных сканеров. Проверить на практике влияние типа светочувствительного датчика на основные характеристики сканеров.

### 3.1. Краткие теоретические сведения

#### 3.1.1. Сканер. Конструктивное исполнение планшетного сканера

**Сканером** принято называть устройство ввода графической информации, выполняющее построчное или постраничное преобразование света, отраженного или пропущенного через плоский графический оригинал, в цифровую форму. Существует множество вариантов конструктивного исполнения сканеров. Наибольшим распространением в области непрофессионального применения получили **планшетные сканеры** – благодаря сравнительной простоте конструкции, компактности, хорошему соотношению «цена/качество результата».

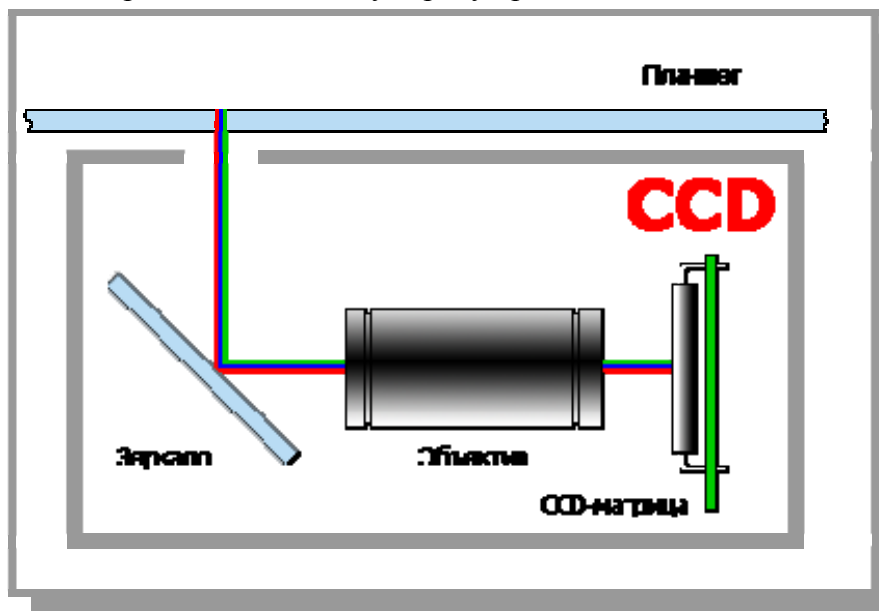
Планшетный сканер представляет собой плоскую коробку с прозрачным окном, по размерам соответствующим размеру оригинала (обычно А4 или А3). Обычно планшетный сканер работает с непрозрачными оригиналами и измеряет интенсивность света, отраженного от поверхности. Однако существуют модели со встроенными или отдельно подключаемыми слайд-модулями, позволяющими производить сканирование оригинала «на просвет»; ярко-белая крышка сканера также дает возможность сканировать прозрачные оригиналы, хотя и с худшим качеством.

Основной элемент конструкции планшетного сканера – **подвижная каретка**, перемещаемая вдоль оригинала с помощью привода шагового двигателя. Каретка содержит **светочувствительный датчик изображения**, выполняющий регистрацию уровня светового излучения, отраженного от оригинала или пропущенного им. Данные с датчика поступают на АЦП и дальше с помощью контроллера кодируются и передаются по интерфейсу на хост-систему, к которой подключен сканер.

Тип светочувствительного датчика определяет основные характеристики сканера – конструкцию каретки, энергопотребление и схему питания, максимально достижимые параметры качества сканирования, стоимость.

### 3.1.2. Светочувствительный датчик типа CCD

В планшетном сканере, использующем сканирующий элемент CCD (ChargeCoupled Device), оригинал располагается на предметном стекле, вдоль которого передвигается каретка с источником света (лампой) и сканирующей головкой. Оптическая система сканера CCD состоит из объективов и зеркал или призм, проецирующих световой поток от сканируемого оригинала на сканирующий элемент – линейку ПЗС, благодаря которой можно сканировать только одну строку оригинала.



В этом отличие линейки от матрицы, на которую экспонируется не одна, а одновременно несколько строк. К слову, сканирующие элементы на ПЗС-матрицах содержат многие профессиональные сканеры, а также цифровые фотоаппараты и видеокамеры, которые также могут быть квалифицированы как специфические сканеры.

На сканирующем элементе осуществляется разделение информации о цветах. Для этой цели в сканер вмонтированы одновременно три параллельных линейки из равного числа отдельных светочувствительных оптико-электронных преобразователей, которые принимают информацию о содержании отдельных составляющих цветов – красном, зеленом и голубом (RGB). На линейке ПЗС уровень освещенности преобразуется в аналоговые электрические сигналы.

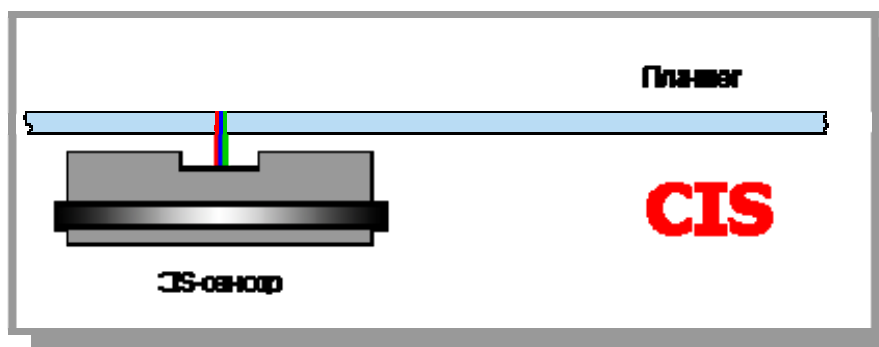
В процессе сканирования оригинал освещается источником излучения – сбалансированными по цвету флуоресцентными лампами, после чего отраженный (преломленный) свет с помощью специальной оптической системы направляется на линейку светочувствительных элементов, которые преобразуют интенсивность принимаемого света в соответствующее значение напряжения. В оптической системе простейшего планшетного сканера CCD содержится три и более зеркал. Таким образом, CCD-сканер достаточно громоздок, имеет высокое энергопотребление, а также довольно критичен к механическим воздействиям.

Датчики ПЗС представляют собой твердотельный электронный компонент, состоящий из множества крошечных светочувствительных элементов, которые формируют электрический заряд, пропорциональный интенсивности падающего на них света. В основу работы ПЗС положена зависимость р-n-перехода обыкновенного полупроводникового диода от степени освещенности. В профессиональных широкоформатных сканерах ПЗС-камеры выполнены по технологии All-Digital: на выходе они уже имеют цифровой сигнал, что гарантирует минимум шума и расширенный динамический диапазон.

Если сканируется прозрачный оригинал, то используется так называемый слайд-адаптер, для которого включается вторая лампа, расположенная в крышке сканера. Вторая лампа перемещается параллельно считывающей головке сканера.

### 3.1.3. Светочувствительный датчик типа CIS

Сканер, использующий технологию **контактного датчика изображения** – CIS (Contact Image Sensor), не содержит сложной оптики. CIS – исторически более поздняя технология планшетных сканеров. Приемный элемент CIS состоит из линейки датчиков (нескольких одинаковых сканирующих матриц), непосредственно воспринимающих световой поток от оригинала.



В процессе перемещения каретки сканера CIS под лежащим на стекле оригиналом освещается подлежащая обработке область. Для этой цели на каретке смонтирован источник света – узел из трех групп светодиодов, генерирующих свет в трех диапазонах волн – красном, зеленом и голубом. Суммарное излучение, представляющее собой луч белого света, падает на оригинал, отражается и возвращается на сканирующую головку. Отраженный свет несет информацию о яркости и спектральном составе области отражения и экспонируется на оптическую систему сканера.

**Оптическая система** планшетного CIS-сканера располагается в непосредственной близости от стекла и представляет собой линейку приемных сенсоров – сканирующих элементов на ПЗС или фототранзисторах. Длина светочувствительной линейки такого преобразователя CIS соответствует ширине планшета сканера, поэтому дополнительные элементы, фокусирующие или перенаправляющие световой поток (зеркала, призмы или линзы) в сканере подобной конструкции отсутствуют.

Преимущества этой технологии – это прежде всего простота конструкции и малая по сравнению с CCD-сканером толщина корпуса. Сканер с полупроводниковым осветителем на основе светодиодного банка отличается низкой потребляемой мощностью и малой чувствительностью к механическим воздействиям. Вместе с тем, сканеры подобного типа имеют ограниченную область применения: они не способны работать со слайд-адаптерами, датчик типа CIS обеспечивает более высокий уровень шума на изображении, имеет низкую разрешающую способность и очень малую глубину резкости.

### 3.1.4. О параметрах качества изображения

**Глубина резкости** – расстояние от светочувствительных элементов до оригинала, обеспечивающее резкость изображения. Эта величина может колебаться от нескольких миллиметров до нескольких сантиметров, и, естественно, чем она выше, тем лучше. Глубина резкости CCD-сканеров в несколько раз больше, нежели у CIS-сканеров: при сканировании, например, объемных объектов или не очень ровных планшетов с использованием CIS-технологии изображение получится нерезким и размытым. Именно поэтому в CIS-сканерах оригинал должен быть достаточно плотно прижат к самим датчикам, что не позволяет сканировать ветхие и потрепанные материалы без риска их повреждения.

**Глубина цвета** отражает такую потребительскую характеристику сканера, как цветопередача. Как правило, цветовое разрешение сканеров указывается в битах, что на деле означает лишь количество интервалов, на которые будет разбит весь диапазон воспринимаемых цветов. Фактически же глубина цвета зависит от качества аналого-цифрового преобразователя и матрицы. Слабая фокусировка, обусловленная отсутствием оптики, и небольшие зазоры между соседними матрицами в CIS-сканерах не мешают сканированию текста и схематичных монохромных изображений, но вот для работы с полноцветной графикой больше подойдет сканер, построенный на основе традиционной CCD-технологии.

### 3.2. Задание на лабораторную работу

- 1) Получить две модели сканеров для проведения работы.
- 2) Определить по внешним признакам, какого типа светочувствительный элемент используется в каждом из сканеров.
- 3) Подключить по очереди каждый из сканеров и провести измерение скорости сканирования в каждом из доступных разрешений и в двух режимах – цветном (Color) и монохромном (Grayscale).
- 4) Провести сканирование объемного объекта (книги на развороте, мышки или другого устройства).
- 5) Построить таблицу или график, на котором будет отмечена зависимость времени, затрачиваемого на выполнение сканирования, от выбранного разрешения и режима.

### 3.3. Порядок выполнения работы

Сканер производства компании HP не требует отдельной установки драйверов, он опознается и конфигурируется в Windows XP. Для сканера производства Mustek требуется установить драйвер.

Сканирование можно выполнять с помощью встроенного апплета «Панели управления» Windows, которая обращается к интерфейсу WIA для выполнения сканирования.

Рекомендуется выбирать формат результирующего файла .BMP или .TIFF, чтобы исключить влияние операции сжатия изображения на общее время выполнения сканирования.

Необходимо учесть, что сканер с датчиком типа CCD требует времени на полный прогрев лампы CCL. Перед началом работы следует сделать несколько тестовых прогонов сканирования.

Замер времени следует проводить подручными средствами.

### 3.4. Содержание отчета

- 1) Титульный лист.
- 2) Таблица с результатами измерения скорости сканирования.
- 3) Два образца сканирования объемного объекта (в низком разрешении).

### 3.5. Контрольные вопросы

- 1) Какой тип сканеров является наиболее распространенным и почему?
- 2) Каковы преимущества сканеров типа CCD?
- 3) Каковы преимущества сканеров типа CIS?
- 4) Какой тип сканеров обеспечивает наибольшую глубину резкости?
- 5) Какой тип сканеров пригоден для сканирования прозрачных оригиналов?
- 6) Как по внешнему виду можно отличить сканеры различных типов?
- 7) Как объяснить характерную разницу в скорости сканирования в цветном и монохромном режиме у сканера типа CIS?





## Лабораторная работа №4. Интерфейс IEEE1394 (FireWire).

Цель работы: Изучить характеристики внешнего последовательного интерфейса IEEE1394. Измерить пропускную способность интерфейса при соединении двух ПК.

### 4.1. Краткие теоретические сведения

#### 4.1.1. Основные сведения об интерфейсе IEEE 1394

Стандарт для высокопроизводительной последовательной шины (High Performance Serial Bus), получивший официальное название **IEEE 1394**, был принят в 1995 году. Целью являлось создание шины, не уступающей параллельным шинам при существенном удешевлении и повышении удобства подключения за счет перехода на последовательный интерфейс. Стандарт основан на шине FireWire, используемой Apple Computer в качестве дешевой альтернативы SCSI в компьютерах Macintosh и PowerMac. Название FireWire («огненный провод») теперь применяется и к реализациям IEEE 1394, оно сосуществует параллельно с кратким обозначением "1394". Другое название того же интерфейса — iLink, а иногда и Digital Link – используется фирмой Sony применительно к устройствам бытовой электроники. MultiMedia Connection – название, используемое в логотипе 1394 High Performance Serial Bus Trade Association (1394TA).

Стандарт 1394 определяет три возможные частоты передачи сигналов по кабелям: 98.304, 196.608 и 393.216 Мбит/с, которые округляют до 100, 200 и 400 Мбит/с соответственно. Частоты в стандарте обозначаются как S100, S200 и S400 соответственно. В новых версиях спецификации IEEE 1394 введены понятия скоростей S800, S1600 и S3200, в коммерческих продуктах реализована только скорость S800.

**Основные свойства** шины FireWire:

- Многофункциональность. Шина обеспечивает цифровую связь до 63 устройств без применения дополнительной аппаратуры (хабов). Устройства бытовой электроники – цифровые камкордеры (записывающие видеокамеры), камеры для видеоконференций, фотокамеры, приемники кабельного и спутникового телевидения, цифровые видеоплееры (CD и DVD), акустические системы, цифровые музыкальные инструменты, а также периферийные устройства компьютеров (принтеры, сканеры, устройства дисковой памяти) и сами компьютеры могут объединяться в единую сеть.
- Высокая скорость обмена и изохронные передачи. Шина позволяет даже на начальном уровне (S100) передавать одновременно два канала видео (30 кадров в секунду) широкоэмитательного качества и стереосигнал с качеством CD.
- Низкая цена компонентов и кабеля.
- Легкость установки и использования. FireWire расширяет технологию PnP. Система допускает динамическое (горячее) подключение и отключение устройств. Устройства автоматически распознаются и конфигурируются при включении/отключении. Питание от шины (ток до 1,5 А) позволяет подключенным устройствам общаться с системой даже при отключении их питания. Управлять шиной и другими устройствами могут не только PC, но и другие «интеллектуальные» устройства бытовой электроники.

FireWire по инициативе VESA позиционируется как шина «домашней сети», объединяющей всю бытовую и компьютерную технику в единый комплекс. Эта сеть является одноранговой (peer-to-peer), чем существенно отличается от USB.

#### 4.1.2. Пакет IOMeter – профессиональный инструмент для измерения эффективности подсистемы ввода-вывода

Пакет **IOMeter** – это профессиональный промышленный инструмент для тестирования подсистем ввода-вывода. Разработкой этого пакета в свое время занимался специальный отдел корпорации Intel. Через некоторое время проект был остановлен и передан в руки свободных разработчиков. Сегодня IOMeter является открытым проектом, официальный сайт которого находится по адресу [www.iometer.org](http://www.iometer.org).

IOMeter позволяет выполнить две **задачи** – сгенерировать поток операций ввода-вывода с заданными характеристиками для любого из доступных сетевых и дисковых интерфейсов и проанализировать реакцию системы на созданную нагрузку. С помощью IOMeter можно успешно эмулировать реальную нагрузку в конкретной вычислительной среде, используя собранные статистические данные, а также создавать синтетическую нагрузку для проверки пиковых характеристик интерфейсов. Анализатор производительности позволяет измерять общую производительность подсистемы ввода-вывода (количество операций в секунду), а также отдельные характеристики – пропускную способность, латентность, частоту возникновения ошибок, загрузку центрального процессора, скорость выполнения соединения и транзакции.

Благодаря описанным выше возможностям IOMeter широко применяется в следующих областях:

- тестирование и анализ эффективности работы дисковых и сетевых контроллеров;
- измерения пропускной способности и латентности шин и интерфейсов;
- тестирование сетевых соединений и каналов;
- низкоуровневое тестирование жестких дисков и дисковых массивов;
- низкоуровневое тестирование сетевого оборудования;
- измерение пропускной способности интерфейсов внешних дисковых устройств.

#### 4.1.3. Принцип работы пакета IOMeter

**Архитектура.** Пакет IOMeter состоит из двух отдельных приложений – *iometer.exe* и *dynato.exe*. Первое приложение представляет собой управляющий сервер и интерфейсную надстройку, в задачи которой входит подготовка шаблонов нагрузки, передача управляющих сигналов модулю Dynamo, сбор и анализ результатов работы. Модуль IOMeter является Windows-приложением, для каждой тестовой конфигурации он запускается в единственном экземпляре.

Модуль Dynamo представляет собой клиентское консольное приложение, которое занимается выполнением низкоуровневых операций ввода-вывода и измерением производительности системы при их проведении. Фактически это генератор нагрузки и анализатор возникающих при этом задержек. Модуль Dynamo является многопоточным кросс-платформенным приложением, поддерживает одновременную работу нескольких своих копий для эмуляции сложных вычислительных сред, способен соединяться с управляющим сервером посредством сетевого интерфейса.

Интерфейс IOMeter может собирать данные одновременно от нескольких копий Dynamo, причем эти копии могут как генерировать, так и принимать данные, а также отсылать ответы на запросы от другой копии. Если одна из копий Dynamo работает на удаленной машине, что возможно благодаря разделению интерфейсной и функциональной частей, то интерфейс IOMeter может анализировать пропускную способность сетевого соединения между локальной и удаленной машиной.

При тестировании дисков (не только винчестеров – любых дисковых устройств, доступных системе) IOMeter может работать как через файловую систему, используя системные вызовы, так и напрямую (в случае жестких дисков с интерфейсом ATA). В

первом случае на диске при первом запуске теста создается файл `iobw.tst`, который занимает часть свободного пространства. Если диск не содержит файловой системы (не разбит на разделы и не отформатирован), IOMeter будет использовать для тестов все доступное пространство диска. Для повышения точности рекомендуется использовать последний способ.

**Терминология.** *Нагрузка* (Workload) – поток данных, генерируемый модулем Dynamo. Нагрузка описывается *спецификацией* (Access Specification), в которой содержатся данные о том, с какой вероятностью модуль Dynamo генерирует запрос на чтение или запись, какой объем данных при этом будет запрошен и как будет сгенерирован адрес первого сектора данных (эмуляция последовательного и произвольного доступа). IOMeter последовательно выполняет каждую спецификацию в течение заданного времени, обнуляя счетчики при переходе к следующей спецификации.

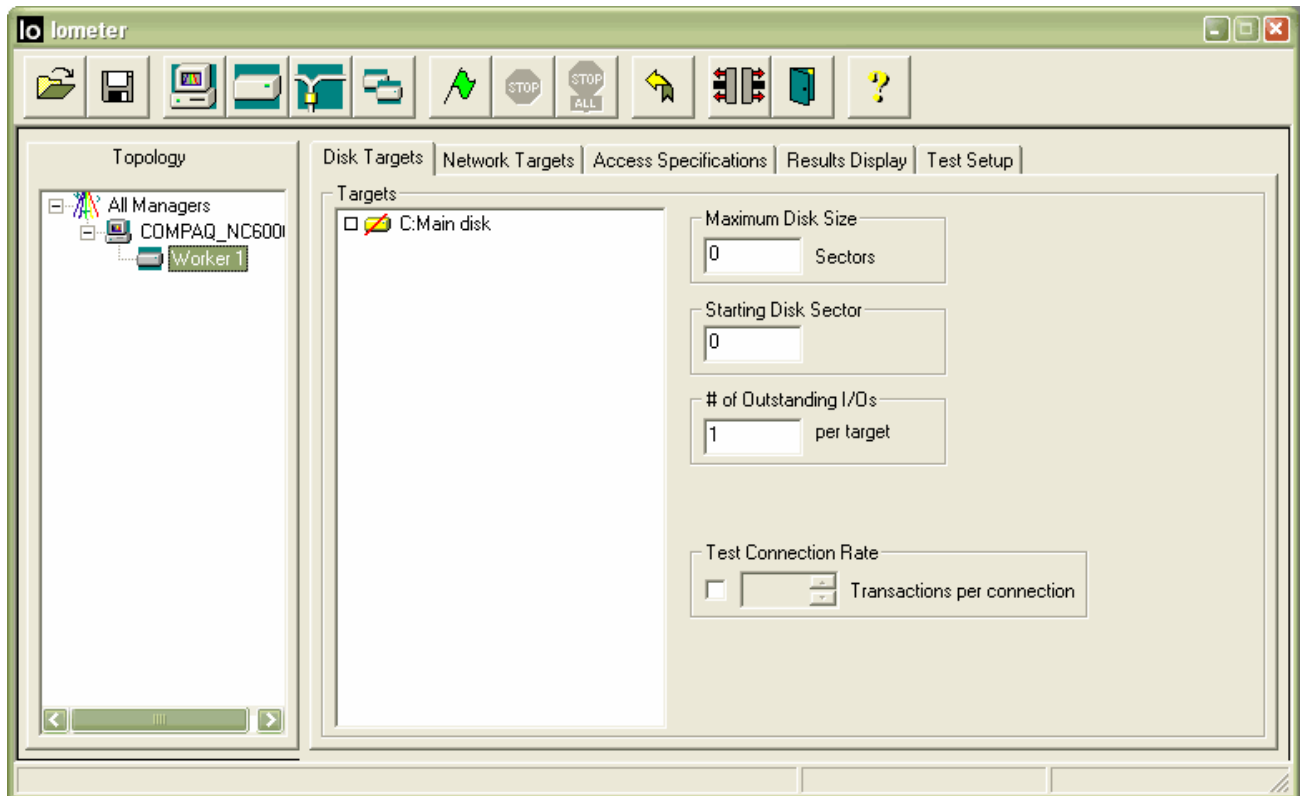
IOMeter может генерировать нагрузку в несколько *независимых потоков* ввода-вывода (Outstanding I/O) для эмуляции одновременного доступа к дисковой подсистеме ввода-вывода нескольких приложений. Количество потоков может меняться (нарастать) с заданным интервалом и скоростью. Можно также задать максимальное количество секторов на диске (или в файле `iobw.tst`) и номер начального сектора. Для сетевых интерфейсов независимые потоки и сектора не задаются.

Машина с запущенным на ней модулем Dynamo называется *менеджером* (Manager). IOMeter автоматически обнаруживает доступные менеджеры в сети и устанавливает с ними соединение. Каждый менеджер позволяет задать один или несколько *рабочих потоков* (Worker). Рабочий поток содержит сведения о дисках или сетевых соединениях, для которых генерируется нагрузка, а также шаблоны, описывающие параметры нагрузки. В общем случае все рабочие потоки выполняются одновременно.

*Целевое устройство* (Target) используется рабочими потоками при генерации нагрузки. Оно может быть как диском (Disk Target), так и сетевым контроллером с заданным IP-адресом (Network Target). Каждому рабочему потоку назначается одно или несколько целевых устройств. В общем случае рабочий поток генерирует запросы ко всем выбранным для него целевым устройствам.

IOMeter может не только задействовать все доступные рабочие потоки для каждой спецификации нагрузки, но и циклически увеличивать объем нагрузки во время выполнения спецификации. В частности, можно наращивать арифметически или геометрически количество независимых потоков ввода-вывода, количество рабочих потоков (от 1 до всех созданных), количество целевых устройств (от 1 до всех назначенных, причем различными способами – у всех рабочих потоков одновременно или по очереди). Это позволяет оценить эффективность работы подсистемы ввода-вывода при возрастании количества работающих с ней приложений, устройств и компьютеров, а также протестировать эффективность ее масштабирования при добавлении новых каналов и устройств.

**Интерфейс.** Экранный интерфейс модуля `iometer.exe` состоит из трех областей – панели инструментов (вверху), панели управления менеджерами и рабочими потоками (слева) и панели настроек тестирования.



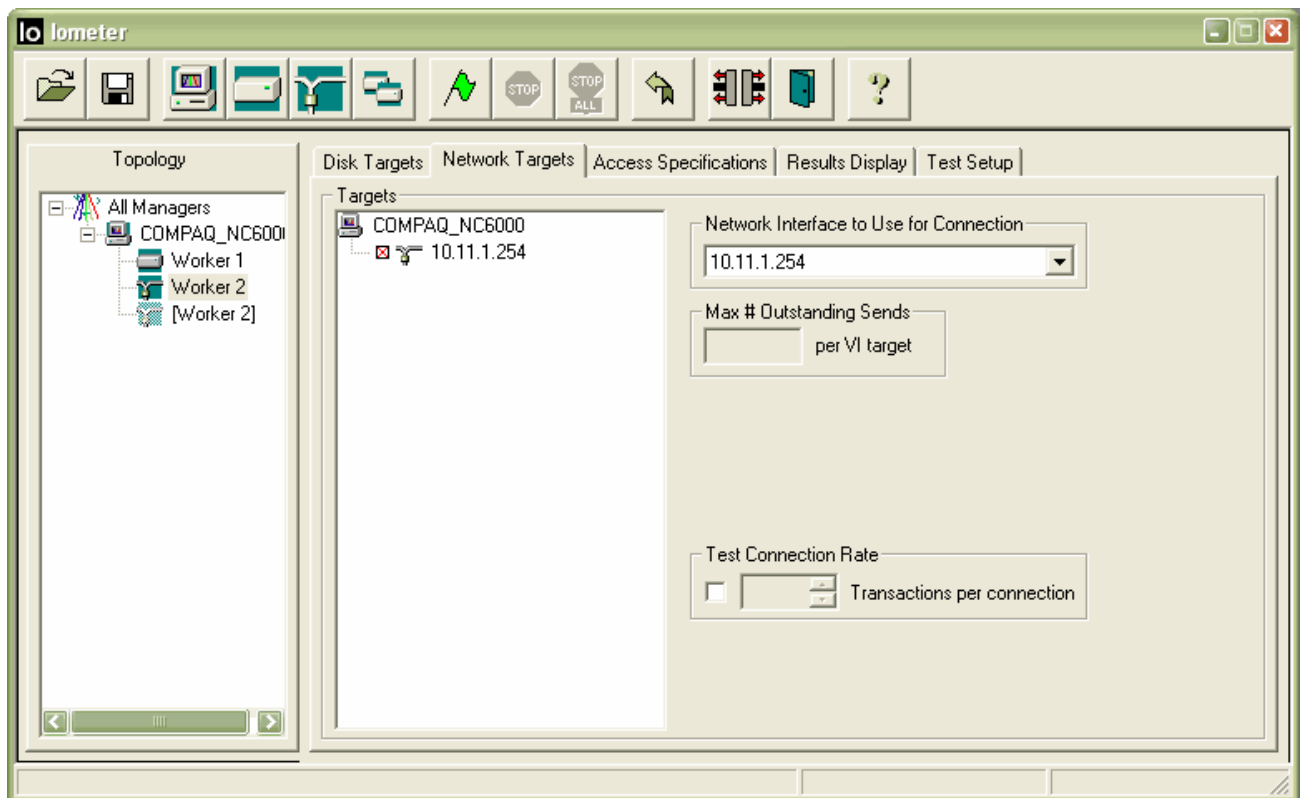
*Панель инструментов* содержит следующие кнопки: открыть файл настроек, сохранить настройки в файл, добавить новый менеджер (запустить копию dynamo.exe), добавить дисковый рабочий поток, добавить сетевой рабочий поток, запустить тест, остановить тест, удалить рабочий поток и некоторые другие.

*Панель управления менеджерами* (панель топологии теста) содержит древовидную схему подключенных компьютеров (менеджеров dynamo) и запущенных на них рабочих потоков. Основное назначение панели – переключение между активными рабочими потоками. Большинство настроек тестирования относятся к тому потоку, который выделен на данной панели.

*Панель настроек тестирования* содержит 5 закладок: выбор дисков, выбор сетевых интерфейсов, настройка шаблонов нагрузки, просмотр результатов и настройка основных параметров теста.

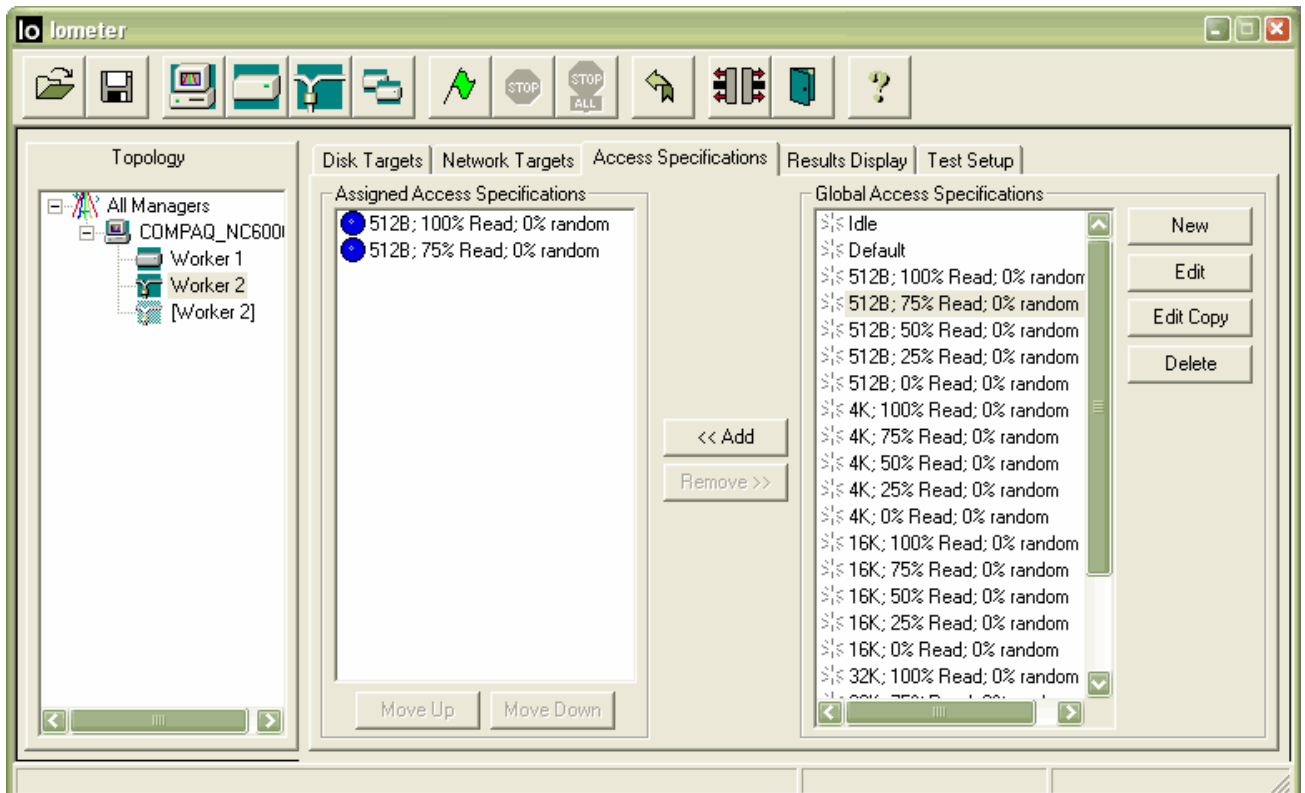
Рассмотрим порядок выполнения настроек для проведения теста.

- 1) Для начала нужно сконфигурировать рабочие потоки – для каждого из тестируемых компьютеров добавить требуемые потоки (сетевые или дисковые), а также удалить ненужные (один дисковый поток порождается автоматически).
- 2) Далее следует настроить рабочие потоки. Для этого нужно выделить требуемый поток, переключиться на закладку Disk Targets или Network Targets и выбрать целевой диск или сетевой интерфейс (обозначается IP-адресом).



При подключении сетевого интерфейса на исходном компьютере к интерфейсу целевого компьютера на той стороне автоматически порождается ответный сетевой поток.

3) Настроить шаблоны нагрузки и перевести их в режим выполнения.



Список справа содержит исходные шаблоны нагрузки, список слева – шаблоны, готовые к выполнению или выполняемые в данный момент.

## 4.2. Задание на лабораторную работу

1. Изучить теоретические сведения об интерфейсе IEEE 1394.
2. Соединить два компьютера с помощью кабеля IEEE 1394, проверить работоспособность сетевого канала.
3. Настроить IOMeter на тестирование сетевого канала.
4. Выполнить тестирование сетевого канала с помощью следующих шаблонов:
  - 4.1. размер запроса:
    - 4.1.1. 512 байт
    - 4.1.2. 2 Кб
    - 4.1.3. 8 Кб
    - 4.1.4. 32 Кб
    - 4.1.5. 128 Кб
    - 4.1.6. 512 Кб
    - 4.1.7. 2 Мб
  - 4.2. направление передачи данных – чтение (во всех случаях);
  - 4.3. длительность выполнения каждого теста – 3 минуты.

## 4.3. Порядок выполнения работы

Для того чтобы IOMeter имел возможность корректно измерять параметры сетевого соединения, необходимо построить специальную конфигурацию – обеспечить обмен данными через сетевой канал между двумя копиями модуля Dynamo. Последовательность действий следующая:

- 1) На локальной и удаленной машинах устанавливаются копии IOMeter.
- 2) На удаленной машине из командной строки запускается модуль Dynamo со следующими параметрами:  
`dynamo.exe /i ip1 /m ip2`,  
 где ip1 – IP-адрес локальной машины, на которой будут производиться измерения,  
 ip2 – IP-адрес данной (удаленной) машины.  
 Вместо IP-адресов можно использовать сетевые имена (NetBIOS) машин.
- 3) Необходимо проследить, чтобы модуль Dynamo выдал сообщение вида “Sending login request...” и указал название текущей машины и IP-адрес вызываемой машины, на которой будут проводиться измерения. Если сообщение не было выдано, либо после него выданы сообщения об ошибках, процедуру следует повторить.
- 4) На локальной машине запускается IOMeter (iometer.exe). Программа должна запускаться без сообщений об ошибках, в окне “Topology” должны появиться два компьютера – текущий и удаленный.
- 5) Развернув в окне “Topology” текущий компьютер, следует нажать кнопку панели “New Network Worker” (новый сетевой рабочий поток).
- 6) Выбрав созданный рабочий поток, следует обратить внимание на закладку “Network Targets” (сетевые устройства назначения). Возле IP-адреса удаленной машины (но не текущей!) следует установить крестик. Одновременно в окне “Topology” на удаленной машине появится новый сетевой рабочий поток, у которого в закладке “Network Targets” автоматически выбирается IP-адрес текущей машины.

Теперь необходимая конфигурация создана, две копии модулей Dynamo обнаружили друг друга и установили связь. Данные от локальной машины будут поступать на удаленную (операция записи), а данные от удаленной – на локальную (операция чтения). Получая

данные от обеих копий Dynamo, IOMeter может анализировать пропускную способность сетевого соединения в обоих направлениях.

#### 4.4. Содержание отчета

- 1) Титульный лист.
- 2) Результаты измерений в табличной или графической форме.

#### 4.5. Контрольные вопросы

- 1) Каково назначение интерфейса IEEE 1394?
- 2) В чем его основное отличие от USB? В чем эти интерфейсы похожи?
- 3) Благодаря чему возможно создание сетевого соединения с использованием IEEE 1394 в качестве физической среды передачи данных?
- 4) Какую пропускную способность канала можно получить практически?