

**ФЕДЕРАЛЬНОЕ АГЕНТСТВО ВОЗДУШНОГО ТРАНСПОРТА**

**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ  
УЧРЕЖДЕНИЕ ВЫСШЕГО ПРОФЕССИОНАЛЬНОГО ОБРАЗОВАНИЯ**

**«МОСКОВСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ  
ГРАЖДАНСКОЙ АВИАЦИИ»**

---

*Кафедра вычислительных машин, комплексов, систем и сетей*

Гладышев Ю.С.

**МЕТОДИЧЕСКОЕ ПОСОБИЕ**

для выполнения лабораторной работы № 2

**Шина РСІ**

по курсу

**“ЭВМ и ПУ”**

*для студентов II курса специальности 22.01 дневного обучения*

Москва -2014

ББК 6Ф7.3

Г 52

Рецензент канд. техн. наук, доц. Л.А. Вайнейкис

Гладышев Ю.С.

Г 52 Шина PCI: Пособие по выполнению лабораторной работы №2. – М.: МГТУ ГА, 2014.-24 с.

Данное пособие издаётся в соответствии с рабочей программой дисциплины «ЭВМ и ПУ» по Учебному плану специальности 230101 для студентов 2 курса дневного обучения.

Рассмотрено и одобрено на заседаниях кафедры . .2014 г. и методического совета . .2014 г.

## Лабораторная работа № 2

### Типовая конфигурация ПК с шиной PCI.

#### 1. Цель работы:

1.1. Определить типовую конфигурацию ПК с шиной PCI. с помощью программы PCI Explorer.

1.2. Определить поля заголовка конфигурационного пространства для выбранных устройств.

1.3. Составить схему коммутации запросов прерываний устройств, подключенных к шине PCI.

#### 2. Теоретические сведения

##### 2.1 Архитектура компьютера с шиной PCI

PCI (*Peripheral Component Interconnect*) - шина **взаимодействия** периферийных компонентов. В современной архитектуре PC, является системной шиной процессора.

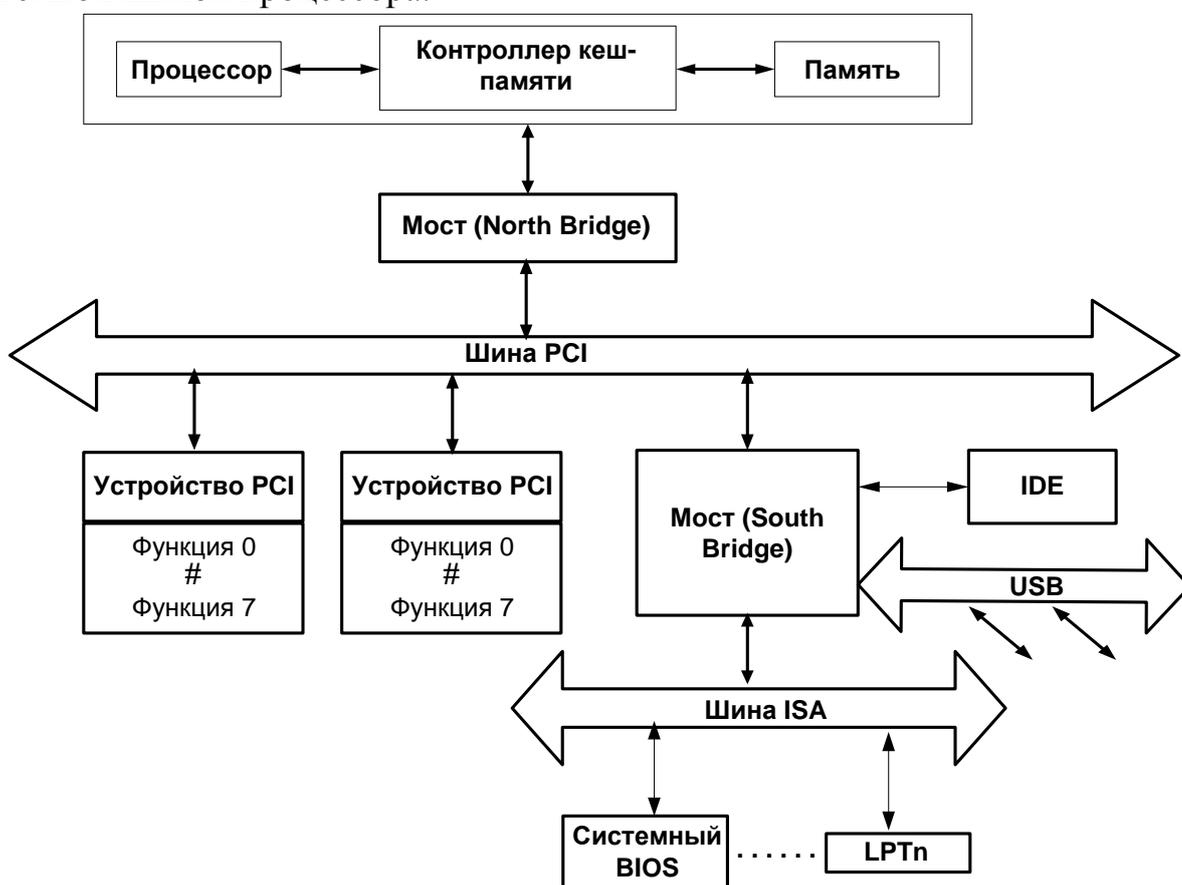


Рис. 1. Типовая конфигурация ПК с шиной PCI.

Шина обеспечивает синхронный 32-х или 64-х разрядный обмен данными. При этом для уменьшения числа контактов (и стоимости) используется мультиплексирование, то есть адрес и данные передаются по одним и тем же линиям.

Частота работы шины 33МГц\_(PCI v1.0) или 66МГц\_(PCI v2.1) позволяет обеспечить широкий диапазон пропускных способностей (с использованием пакетного режима):

132 МВ/сек при 32-bit/33МГц; 264 МВ/сек при 32-bit/66МГц;  
264 МВ/сек при 64-bit/33МГц; 528 МВ/сек при 64-bit/66МГц.

При этом для работы шины на частоте 66МГц необходимо, чтобы все периферийные устройства работали на этой частоте.

К основным характеристикам шины относятся:

- Полная поддержка *multiply bus master* (например, несколько контроллеров жестких дисков могут одновременно работать на шине).
- Поддержка межшинных мостов для осуществления режимов кэширования *write-back*, *write-through* и т. д.
- Автоматическое конфигурирование карт расширения при включении питания.
- Спецификация шины позволяет комбинировать до восьми функций на одной карте (например, видео + звук и т.д.).
- Шина позволяет устанавливать до 4 слотов расширения, однако возможно использование моста PCI-PCI для увеличения количества карт расширения.
- PCI-устройства оборудованы таймером, который используется для определения максимального промежутка времени, в течении которого устройство может занимать шину.

При разработке шины в ее архитектуру были заложены передовые технические решения, позволяющие повысить пропускную способность.

Шина поддерживает метод передачи данных, называемый "linear burst" (метод линейных пакетов). Этот метод предполагает, что пакет информации считывается (или записывается) "одним куском", то есть адрес автоматически увеличивается для следующего байта. Естественным образом при этом увеличивается скорость передачи собственно данных за счет уменьшения числа передаваемых адресов.

В более поздних версиях была достигнута частота работы шины 533MHz (PCI-X v2.0-2002г), при этом пиковая пропускная способность составила 4096 Мбайт/с, а частота работы шины (PCI Express v1.0-2002г.) составила 2.5 GHz.

Спецификация шины PCI определяет три типа ресурсов памяти: два обычных (диапазон памяти и диапазон ввода/вывода, как их называет компания Microsoft) и *configuration space* - "конфигурационное пространство".

## 2.2. Конфигурационное пространство шины PCI.

Конфигурационные регистры устройств PCI расположены в обособленном пространстве адресов (отдельном от пространства *адресов памяти* и *ввода-вывода*, таблица 1).

Таблица 1. Заголовок конфигурационного пространства одной функции устройства PCI.

Байты конфигурационного пространства				
3	2	1	0	
Device ID		Vendor ID		00h
<b>Status Register</b>		<b>Command Register</b>		04h
Class Code		Revision ID		08h
BIST	Header Type	Lat Timer	CL Size	0Ch
<b>Base Address 0</b>				10h
<b>Base Address 1</b>				14h
<b>Base Address 2</b>				18h
<b>Base Address 3</b>				1Ch
<b>Base Address 4</b>				20h
<b>Base Address 5</b>				24h
Card Bus CIS Pointer				28h
Subsystem ID		Subsystem Vendor ID		2Ch
Expansion ROM Base Address				30h
Зарезервировано			Cap. Pointer	34h
Зарезервировано				38h
<b>Max_Lat</b>	<b>Min_Gnt</b>	<b>Int_Pin</b>	<b>Int_Line</b>	3Ch
Определяется пользователем				40h
				FFh

- *Device ID* - идентификационный номер устройства, назначаемый производителем.
- *Vendor ID* - идентификатор производителя устройства.
- *Revision ID* - версия продукта, назначенная производителем.
- *Header Type* - тип заголовка (биты [6:0]), определяющий формат ячеек в диапазоне 10-3Fh и несущий признак многофункционального устройства (если бит 7 установлен)
- *Class Code* - код класса, определяющий основную функцию устройства. Старший байт (адрес 0Bh) определяет базовый класс. Средний – подкласс, младший - программный интерфейс
- *BIST - Built-In Self Test* - встроенный тест

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

**Регистр команд** - Command (R/W) - служит для управления поведением устройства на шине PCI.

**Регистр состояния** - Status - служит для определения *состояния* и *свойств* устройства.

**Регистр Cache Line Size (CL Size) (RW)** - служит для задания размера строки кэша (0-128, допустимые значения  $2^n$ , иные трактуются как 0). По этому параметру инициатор определяет, какой командой чтения воспользоваться (обычное чтение, чтение строки или множественное чтение). Ведомое устройство использует этот параметр для поддержки пересечения границ строк при пакетных обращениях к памяти. По сбросу регистр обнуляется.

**Регистр Latency Timer (RW)** - задает значение таймера, ограничивающего длину транзакции при снятии сигнала GNT#. Значение указывается в виде числа тактов шины, часть битов может не допускать изменения (обычно младшие три бита неизменны, так что таймер программируется с дискретностью в 8 тактов).

**Регистр BIST (RW)** служит для управления встроенным самотестированием (Built-in Self Test).

**Регистр Card Bus CIS Pointer** (необязательный) содержит указатель на структуру описателя Card Bus для комбинированного устройства PCI + Card Bus.

**Регистр линии прерывания (Interrupt Line register)** является регистром «только для записи», который указывает приоритет и вектор (номер системного прерывания), которые **редиректор прерываний** должен присвоить данной функции. В персональных компьютерах Intel x86 значения 0x00-0x0F соответствует **IRQ0-IRQ15**, в системах с APIC может иметь и большее значение; 255 - неизвестный вход или не используется).

**Регистр вывода прерывания (Interrupt pin register)**. Является регистром «только для чтения», который идентифицирует **линию прерывания PCI (INTA#-INTD#)**, используемую данной функцией данного PCI устройства, причём: **0** - не используется, **1** - INTA#, **2** - INTB#, **3** - INTC#, **4** - INTD#, **5-FFh** - резерв.

**Регистр Min\_GNT (RO)** задает минимальное время, на которое ведущему устройству должно предоставляться управление шиной из расчета на частоту 33 МГц, в интервалах по 0,25 мкс.

**Регистр Max\_Lat (RO)** задает максимально допустимую задержку предоставления ведущему устройству доступа к шине, в интервалах по 0,25 мкс (0 - нет специальных требований).

**Регистры Subsystem ID (RO, задается производителем) и Subsystem Vendor ID (RO, производитель получает в PCI SIG)** хранят идентификаторы, позволяющие точно идентифицировать карты и устройства (в системе могут быть установлены несколько карт с совпадающими идентификаторами устройства и производителя Device ID/vendor ID). В поле 2Ch ставится идентификатор производителя карты PCI (может совпадать со значением в поле 0, если фирма выпускает и микросхемы, и карты).

**Регистр Capability Pointer (CAP\_PTR)** содержит указатель на цепочку блоков регистров свойств функции, представленных в конфигурационных регистрах. Каждый блок представляет собой набор регистров, начинающийся с границы двойного слова (в указателе биты [1:0] сброшены). Каждый блок

начинается с байта идентификатора типа свойства (CAP\_ID, определенный PCI SIG), за которым следует указатель на следующий блок (нулевой указатель является признаком конца списка блоков), после чего расположены байты описаний самих свойств. Через CAP\_PTR, например, отыскиваются регистры управления энергопотреблением (если есть), регистры AGP, некоторые регистры хост-контроллера USB 2.0 и ряд других.

*Регистры Base Address Registers (BAR)* описывают области памяти и портов ввода-вывода. Программными манипуляциями с регистрами можно определить размеры областей. Для областей памяти и портов описания различаются:

*Регистр Expansion ROM Base Address* управляет адресацией ПЗУ программной поддержки устройства. Размер ПЗУ определяется так же, как и в регистрах базовых адресов. Обращение к ПЗУ возможно лишь при разрешенном использовании памяти (бит 1 в регистре команд).

*Конфигурационные регистры обеспечивают возможность автоматической настройки всех устройств шины PCI.* К этим регистрам система обращается на этапе конфигурирования - *переучета* обнаруженных устройств, выделения им не перекрывающихся ресурсов (областей памяти и пространства ввода - вывода) и назначения номеров аппаратных прерываний.

При дальнейшей работе взаимодействие прикладного ПО с устройствами осуществляется преимущественно путем обращения по назначенным в процессе конфигурирования адресам памяти и ввода-вывода. Конфигурационные же регистры в регулярной работе используются для системных целей: настройки параметров, описывающих поведение устройства на шине, обработки ошибок, идентификации источника прерываний.

### **2.3. Протокол, команды и транзакции шин PC и PCI-XI.**

Обращения к регистрам и памяти устройств PCI выполняются командами шины PCI. Команды может подавать любой инициатор - как хост (главный мост) по командам центрального процессора, так и рядовое устройство PCI. Возможность распространения ряда команд зависит от взаимного расположения инициатора и целевого устройства на ветвях дерева шин PCI. Однако хост безусловно может подать любую команду любому устройству PCI. Только хост всегда имеет доступ к конфигурационным регистрам всех устройств (и мостов), поэтому он и должен заниматься конфигурированием. После конфигурирования любое устройство PCI может безусловно обратиться к системной памяти, то есть реализовать **прямой доступ к памяти (DMA)**.

*Обмен информацией по шине PCI и PCI-X организован в виде транзакций - логически завершенных операций обмена.* В типовой транзакции участвуют два устройства - инициатор обмена (**initiator**) - он же ведущее устройство (**master**) и целевое устройство (**target**) - оно же ведомое (**slave**). Правила взаимодействия этих устройств определяются

протоколом шины PCI Устройство может следить за транзакциями на шине и не являясь их участником (не вводя никаких сигналов); режиму слежения соответствует термин Snooping. Есть особый тип транзакции (**Special Cycle**) - широковещательный, в котором инициатор протоколно не взаимодействует ни с одним из устройств.

В каждой транзакции выполняется одна команда, как правило, чтение или запись данных по указанному адресу. Транзакция начинается с фазы адреса, в которой инициатор задает команду и целевой адрес. Далее могут следовать фазы данных, в которых одно устройство (источник данных) помещает данные на шину, а другое (приемник) их считывает. Транзакции, у которых присутствует множество фаз данных, называются пакетными. Есть и одиночные транзакции (с одной фазой данных). Транзакция может завершиться и без фаз данных, если целевое устройство (или инициатор) не готово к обмену. В шине PCI-X добавлена фаза атрибутов, в которой передается дополнительная информация о транзакции.

*Команды PCI определяют направление и тип транзакций, а также адресное пространство, к которому они относятся.* Набор команд шины PCI включает следующие:

- **I/O Read, I/O Write** - команды чтения и записи ввода/вывода, служат для обращения к пространству портов;

- **Memory Read, Memory Write** - команды чтения и записи памяти, служат для выполнения коротких (как правило, не пакетных) транзакций. Их прямое назначение - обращение к отображенным на память устройствам ввода/вывода. Для «настоящей» памяти, допускающей предвыборку, предназначены команды чтения строк, множественного чтения и записи с инвалидацией;

- **Memory-Read Line** - чтение строки памяти, применяется, когда в транзакции планируется чтение до конца строки кэша. Выделение данного типа чтения позволяет повысить производительность обмена с памятью;

- **Multiple Memory Read** - множественное чтение памяти, используется для транзакций, затрагивающих более одной строки кэш-памяти. Использование данного типа транзакций позволяет контроллеру памяти выполнять упреждающую выборку строк, что дает дополнительный выигрыш производительности;

- **Memory Write and Invalidate (MWT)** - запись с инвалидацией, применяется к целым строкам кэша, причем все байты во всех фазах должны быть разрешены. Эта операция заставляет контроллер кэш-памяти очищать «грязные» строки кэша, соответствующие записываемой области, без их выгрузки в ОЗУ, что экономит время. Инициатор, вводящий эту команду, должен знать размер строки кэша в данной системе (для этого у него есть специальный регистр в конфигурационном пространстве);

- **Dual Address Cycle (DAC)** - двухадресный цикл, позволяет по 32-битной шине обращаться к устройствам с 64-битной адресацией. В этом случае младшие 32 бита адреса передаются одновременно с данной командой, а далее

следует обычный цикл, определяющий команду обмена и несущий старшие 32 бита адреса. Шина PCI допускает 64-битную адресацию как памяти, так и портов ввода/вывода (последнее для систем на x86 бесполезно, но PCI существует и на других платформах);

- **Configuration Read, Configuration Write** - команды конфигурационного чтения и записи, адресуются к конфигурационному пространству устройств. Обращение производится только выровненными двойными словами, биты AD[1:0] используются для идентификации типа цикла (см. ниже). Для генерации данных команд требуется специальный аппаратно-программный механизм;

- **Special Cycle - специальный цикл**, отличается от всех других тем, что является ширококестельным. Однако ни один агент на него не отвечает, а главный мост или иное устройство, вводящее этот цикл, всегда завершает его способом Master Abort (на него требуется 6 тактов шины). Специальный цикл предназначен для генерации ширококестельных сообщений, которые могут читать любые «за интересованные» агенты шины. Тип сообщения декодируется содержимым линий AD[15:0]; на линиях AD[31;16] могут помещаться данные, передаваемые в сообщении. Фаза адреса в этом цикле обычными устройствами игнорируется, но мосты используют ее информацию для управления распространением сообщения. Сообщения с кодами 0000h, 0001h и 0002h требуются для указания на отключение (сообщение **Shutdown**), остановку (сообщение **Halt**) процессора или специфические функции процессора x86, связанные с кэшем и трассировкой. Коды 0003h-FFFFh зарезервированы. Специальный цикл может генерироваться тем же аппаратно-программным механизмом, что и конфигурационные циклы, но со специфическим значением адреса;

- **Interrupt Acknowledge (INTA)** - команда подтверждения прерывания, предназначена для чтения вектора прерываний. По протоколу она выглядит как команда чтения, неявно адресованная к системному контроллеру прерываний. Здесь в фазе адреса по шине AD полезная информация не передается (BE[3:0]# задают размер вектора), но ее инициатор (главный мост) должен обеспечить стабильность сигналов и корректность бита четности, В x86-архитектуре 8-битный вектор передается в байте 0 по готовности контроллера прерываний (по сигналу **TRDY#**). Подтверждение прерываний выполняется за один цикл (первый холостой цикл, который процессоры x86 делают в дань совместимости со стариной, мостом подавляется).

Таблица 2. Декодирование команд шин PCI и PCI-X

<i>Код C/BE [3:0]</i>	<i>PCI Команда</i>	<i>PCI-X Команда</i>	<i>Длина</i>	<i>Возможность расщепления</i>
0000	Interrupt Acknowledge	<i>Interrupt Acknowledge</i> , подтверждение прерывания	DWORD	+

0001	Special Cycle'	<i>Special Cycle</i> , специальный широковещательный цикл	DWORD	-
0010	I/O Read	<i>I/O Read</i> , чтение ввода/вывода	DWORD	+
0011	I/O Write	<i>I/O Write</i> , запись ввода/вывода	DWORD	+
0100	Резерв	Резерв	-	-
0101	Резерв	<i>Device ID Message (DIM)</i> , посылка сообщения устройству (PCI-X 2.0)	Пакет	-
0110	<i>Memory Read</i>	<i>Memory Read DWORD</i> , одиночное чтение памяти	DWORD	+
0111	<i>Memory Write</i>	<i>Memory Write</i> , запись памяти	Пакет	-
1000	Резерв	<i>Alias to Memory Read Block</i> , псевдоним чтения блока памяти	Пакет	+
1001	Резерв	<i>Alias (o Memory Write Block</i> , псевдоним записи блока памяти	Пакет	-
1010	<i>Configuration Read</i>	<i>Configuration Read</i> , конфигурационное чтение	DWORD	+
1011	<i>Configuration Write</i>	<i>Configuration Write</i> , конфигурационная запись	DWORD	+
1100	<i>Memory Read Multiple</i>	<i>Split Completion</i> , завершение расщепленной транзакции	Пакет	-
1101	<i>Dual Address Cycle</i>	<i>Dual Address Cycle (DAC)</i> , цикл передачи расширенного адреса памяти	-	-
1110	<i>Memory Read Line</i>	<i>Memory Read Block</i> , чтение блока памяти	Пакет	+
1111	<i>Memory Write and Invalidate</i>	<i>Memory Write Block</i> , запись блока памяти	Пакет	-

В шине PCI байты шины AD, несущие реальную информацию, определяются сигналами C/BE[3:0]# в фазах данных (см. приложение 7.1). Разрешенные байты могут быть разрозненными; возможны фазы данных, в которых не разрешено ни одного байта.

Для каждого из трех пространств - *памяти, портов ввода/вывода и конфигурационных регистров* - адресация различна: в специальных циклах адрес игнорируется.

#### **2.4. Прерывания PCI: INTx#, PME#, MSI и SERR#**

Устройства PCI могут вырабатывать *запросы аппаратных прерываний*:

- *обычные маскируемые* - для сигнализации событий в устройстве; эти прерывания могут сигнализироваться как традиционным способом - по специальным сигнальным линиям, так и передачей сообщений (MSI);

- *немаскируемые* - для сигнализации о серьезных ошибках;
- *прерывания системного управления* (System Management Interrupt, SMI) - для сигнализации о событиях в системе управления энергопотреблением и некоторых системных целях (например, эмуляции работы стандартного контроллера клавиатуры с помощью устройств USB).

Наиболее эффективно возможности шины PCI используются при применении *активных устройств - мастеров шины (PCI Bus Master)*. Только эти устройства могут обеспечить скорость передачи данных, приближающуюся к декларированной пиковой пропускной способности. Максимальная производительность обменов по шине PCI достигается только в пакетных транзакциях значительной длины. Транзакции по инициативе программы, исполняемой на ЦП, проводимые главным мостом, как правило, являются одиночными (или очень короткими пакетными). По этой причине программно-управляемый обмен данными с устройствами PCI по производительности значительно уступает обмену, выполняемому устройством-мастером. Таким образом, применение активных устройств дает двойной эффект: разгружает центральный процессор и обеспечивает лучшее использование пропускной способности шины.

Вызов процедуры обслуживания прерываний в реальном и защищенном режимах процессора существенно различается:

- *в реальном режиме* таблица прерываний содержит 4-байтные *дальние указатели* (сегмент и смещение) на соответствующие процедуры, которые вызываются дальним вызовом (**Call Far** с предварительным сохранением флагов). Размер (256x4 байт) и положение таблицы (начинается с адреса 0) фиксированы;

- *в защищенном режиме* (и в его частном случае - режиме V86) таблица содержит 8-байтные *дескрипторы прерываний*, которые могут быть шлюзами прерываний (**Interrupt Gate**), ловушек (**Trap Gate**) или задач (**Task Gate**). Размер таблицы может быть уменьшен (максимальный - 256x8 байт), положение таблицы может меняться (определяется содержимым регистра **IDTR** процессора). Код обработчика прерываний должен быть не менее привилегированным, чем код прерываемой задачи (иначе сработает исключение защиты). По этой причине обработчики прерываний должны работать на уровне ядра ОС (на нулевом уровне привилегий). Смена уровня привилегии при вызове обработчика приводит к дополнительным затратам времени на переопределение стека. Прерывания, вызывающие переключение задач (через **Task Gate**), расходуют значительное время на переключение контекста - выгрузку регистров процессора в сегмент состояния старой задачи и их загрузку из сегмента состояния новой.

Номера векторов, используемых для аппаратных прерываний в операционных системах защищенного режима, отличаются от номеров, используемых в ОС реального режима, чтобы исключить их конфликты с векторами, используемыми для исключений процессора.

На *немаскируемое прерывание* (**NMI - Non-Maskable Interrupt**) процессор реагирует всегда (если обслуживание предыдущего NMI завершено); этому прерыванию соответствует фиксированный вектор 2. Немаскируемые прерывания в PC используются для сигнализации о фатальных аппаратных ошибках. Сигнал на линию **NMI** приходит от схем контроля памяти (четности или **ECC**), от линий контроля шины ISA (**IOCHK**) и шины PCI (**SERR#**). Сигнал **NMI** блокируется до входа процессора установкой в 1 бита 7 порта **070h**, отдельные источники разрешаются и идентифицируются битами порта **061h**:

- бит 2 R/W - **ERP** - разрешение контроля ОЗУ и сигнала **SERR#** шины PCI;
- бит 3 R/W - **EIC** - разрешение контроля шины ISA;
- бит 6 R - **IOCHK** - ошибка контроля на шине ISA (сигнал **IOCHK#**);
- бит 7 R - **PCK** - ошибка четности ОЗУ или сигнал **SERR#** на шине PCI.

Реакция процессора на *маскируемые прерывания* может быть задержана сбросом его внутреннего флага **IF** (инструкция **CLI** запрещает прерывания, **STI** - разрешает). Маскируемые прерывания используются для сигнализации о событиях в устройствах. По возникновении события, требующего реакции, адаптер (контроллер) устройства формирует *запрос прерывания*, который поступает на вход контроллера *прерываний*. Задача контроллера прерываний - довести до процессора запрос прерывания и сообщить вектор, по которому выбирается программная процедура обработки прерываний.

Процедура обработки прерывания от устройства должна выполнить действия по обслуживанию данного устройства, включая сброс его запроса для обеспечения возможности реакции на следующие события, и послать команды завершения в контроллер прерываний. Вызывая процедуру обработки, процессор автоматически сохраняет в стеке значение всех флагов и сбрасывает флаг **IF**, что запрещает маскируемые прерывания. При возврате из этой процедуры (по инструкции **IRET**) процессор восстанавливает сохраненные флаги, в том числе и установленный (до прерывания) **IF**, что снова разрешает прерывания. Если во время работы обработчика прерываний требуется реакция на иные прерывания (более приоритетные), то в обработчике должна присутствовать инструкция **STI**. Особенно это касается длинных обработчиков; здесь инструкция **STI** должна вводиться как можно раньше, сразу после критической (не допускающей прерываний) секции. Следующие прерывания того же или более низкого уровня приоритета контроллер прерываний будет обслуживать только после получения команды завершения прерывания *EOI (End Of Interrupt)*.

В IBM PC совместимых компьютерах применяются два основных типа контроллеров прерываний:

- *PIC (Peripheral Interrupt Controller)* - периферийный контроллер прерываний, программно совместимый с «историческим» контроллером 8259А, применявшимся еще в первых моделях IBM PC. Со времен IBM PC/AT применяется связка из пары каскадно-соединенных PIC, позволяющая обслуживать до 15 линий запросов прерываний;

- *APIC (Advanced Peripheral Interrupt Controller)* – усовершенствованный периферийный контроллер прерываний, введенный для поддержки мультипроцессорных систем в компьютеры на базе процессоров 4 -5 поколений (486 и Pentium) и используемый поныне для более поздних моделей процессоров. Кроме поддержки мультипроцессорных конфигураций современный APIC позволяет увеличивать число доступных линий прерываний и обрабатывать запросы прерываний от устройств PCI, посылаемые через механизм сообщений (MSI). Компьютер, оснащенный контроллером APIC, обязательно имеет возможность функционировать и в режиме, совместимом со стандартной связкой пары PIC. Этот режим включается по аппаратному сбросу (и включению питания), что позволяет использовать старые ОС и приложения MS DOS, «незнающие» - APIC и мультипроцессорования.

### **2.5. Коммутация запросов прерываний**

Традиционная схема формирования запросов прерываний с использованием пары PIC (*Peripheral Interrupt Controller*) изображена на рис. 2.

Для устройств PCI выделяется четыре проводных линии запросов (IRQX, IRQY, IRQZ, IRQW), соединяемых с контактами INTA#, INTB#, INTCS# и INTD# всех слотов PCI с циклическим смещением цепей (см. рис. 3). Соответствие линий INTx# и входов IRQ для устройства любой шины PCI приведено в табл. 3. Мосты PCI просто электрически соединяют одноименные линии INTx# своих первичных и вторичных шин.

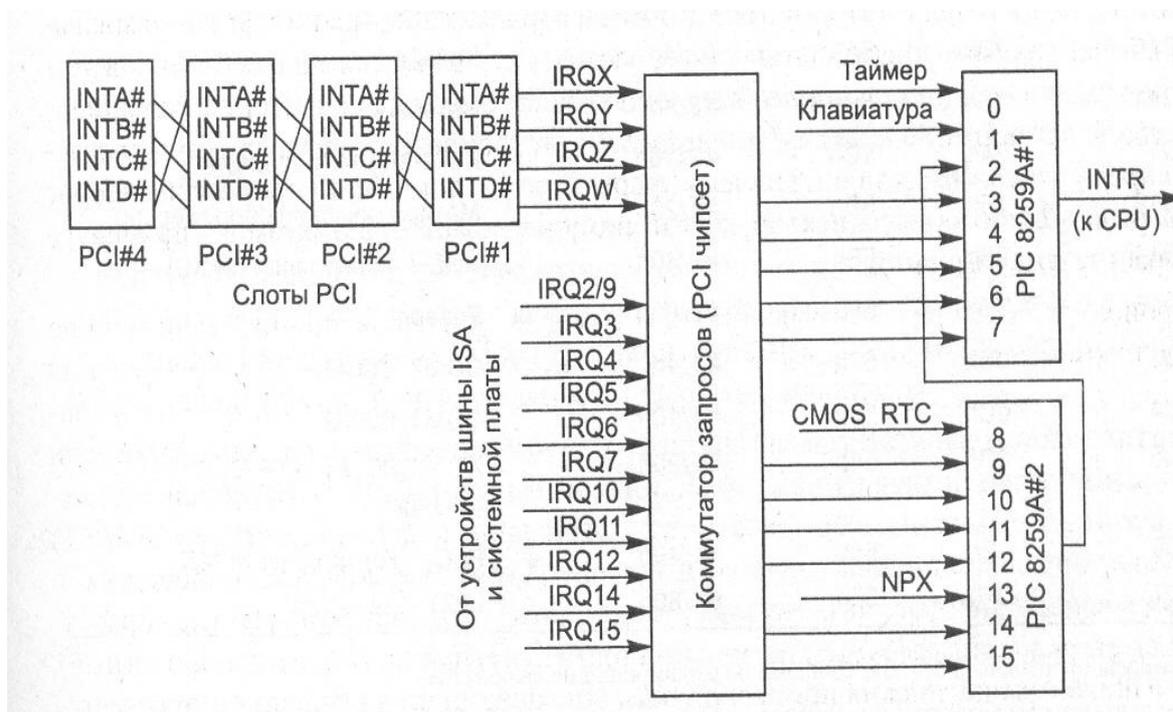


Рис. 2. Коммутация запросов прерываний

В системах с APIC, в которых число входов запросов увеличено до 24, дополнительные 8 входов могут использоваться периферийными устройствами, установленными на системной плате. На слотах PCI остаются доступными лишь четыре обычные линии запросов.

Таблица 3. Коммутация запросов прерываний для устройств PCI

Контакт слота	Вход коммутатора запроса для устройства с номером:			
	0,4,8,----- 28	1,5,9, ----- 29	2,6,10--- 30	3,7,11----31
INTA#	IRQW	IRQX	IRQY	IRQZ
INTB#	IRQX	IRQY	IRQZ	IRQW
INTC#	IRQY	IRQZ	IRQW	IRQX
INTD#	IRQZ	IRQW	IRQX	IRQY

Устройство PCI вводит сигнал прерывания *низким* уровнем (выходом с открытым коллектором или стоком) на выбранную линию INTx#. Этот сигнал должен удерживаться до тех пор, пока программный драйвер, вызванный по прерыванию, не сбросит запрос прерывания, обратившись по шине к данному устройству. Если после этого контроллер прерываний снова обнаруживает низкий уровень на линии запроса, это означает, что запрос на ту же линию ввело другое устройство, разделяющее данную линию с первым, и оно тоже требует обслуживания.

На входы контроллеров прерываний поступают запросы от системных устройств (клавиатура, системный таймер, CMOS-таймер, сопроцессор), периферийных контроллеров системной платы и карт расширения. Традиционно все линии запросов, не занятые перечисленными устройствами, присутствуют на всех слотах шины ISA/

EISA. Эти линии обозначаются как IRQx и имеют общепринятое назначение (табл. 4).

Часть этих линий отдается в распоряжение шины PCI. В табл. 4 отражены и приоритеты прерываний - (запросы расположены в порядке их убывания). Номера векторов, соответствующих линиям запросов контроллеров, система приоритетов и некоторые другие параметры задаются программно при инициализации контроллеров. Эти основные настройки остаются традиционными для обеспечения совместимости с программным обеспечением, но различаются для ОС реального и защищенного режимов. Так, например, в ОС Windows базовые векторы для ведущего и ведомого контроллеров - 50h и 58h соответственно.

Таблица 4. Аппаратные прерывания в порядке убывания приоритета

<i>Имя (номер<sup>1</sup>)</i>	<i>Вектор<sup>2</sup></i>	<i>Вектор<sup>3</sup></i>	<i>Контрол- лер/маска</i>	<i>Описание</i>
NMI	02h			Контроль канала, четность памяти (в XT - сопроцессор)
IRQ0	08h	50h	#1/1 h	Таймер (канал 0 8253/8254)
IRQ1	09h	51h	#1/2h	Клавиатура
IRQ2	0Ah	52h	#1/4h	XT - резерв, AT - недоступно (подключается каскад IRQ8-IRQ15)
IRQ8	70h	58h	#2/1 h	CMOS RTC - часы реального времени
<b>IRQ9</b>	<b>71h</b>	<b>59h</b>	<b>#2/2h</b>	<b>Резерв</b>
<b>IRQ10</b>	<b>72h</b>	<b>5Ah</b>	<b>#2/4h</b>	<b>Резерв</b>
<b>IRQ11</b>	<b>73h</b>	<b>5Bh</b>	<b>#2/8h</b>	<b>Резерв</b>
<b>IRQ12</b>	<b>74h</b>	<b>5Ch</b>	<b>#2/10h</b>	<b>PS/2-Mouse (резерв)</b>
IRQ13	75h	5Dh	#2/20h	Математический сопроцессор
IRQ14	76h	5Eh	#2/40h	HDC - контроллер НЖМД
IRQ15	77h	5Fh	#2/80h	Резерв
IRQ3	0Bh	52h	#1/4h	COM2, COM4
IRQ4	0Ch	53h	#1/10h	COM1, COM3
IRQ5	0Dh	54h	#1/20h	XT - HDC, AT - LPT2, Sound (резерв)
IRQ6	0Eh	55h	#1/40h	FDC - контроллер НГМД
IRQ7	0Fh	56h	#1/80h	LPT1 - принтер
<sup>1</sup> Запросы прерываний 0,1,8 и 13 на шины расширения не выводятся.				
<sup>2</sup> Указаны номера векторов при работе в реальном режиме процессора.				
<sup>3</sup> Указаны номера векторов при работе в ОС Windows.				

Устройство, подключаемое к PCI и представляющее одну функцию, должно использовать только линию INTA#.

Многофункциональные устройства могут использовать комбинации из четырёх линий, начиная с INTA#. Единственное ограничение состоит в том, что каждая из восьми функций (возможных в одном устройстве) может использовать только одну линию прерывания. Соответственно, устройство с внутренними восемью функциями может задействовать имеющиеся линии INTA#, INTB#, INTC#, INTD# следующим образом:

1. Все восемь функций подключены к INTA#.
2. Семь подключены к INTA#, одна к INTB#.
3. Две подключены к INTA#, две к INTB#, две к INTC# и две к INTD#.
4. Четыре подключены к INTA#, четыре к INTB# и т.п.

Спецификация PCI относительно безразлична к приоритетам прерываний. Приоритет в данном случае зависит от внешнего контроллера, который переадресует запрос на прерывание PCI устройства в соответствующую линию системных прерываний.

Например, на персональных компьютерах **редиректор** (коммутатор запросов) может преобразовать запрос функциональной единицы PCI по линиям INTA#-INTD# в запрос по одной из линий IRQ0-IRQ15.

Чтобы можно было это сделать, любая функция внутри PCI устройства, которая генерирует прерывание, должна задействовать два конфигурационных регистра (в своём конфигурационном пространстве):

1. **Регистр вывода прерывания** (Interrupt pin register). Является регистром «только для чтения», который идентифицирует линию прерывания PCI (INTA#-INTD#), используемую данной функцией данного PCI устройства.

2. **Регистр линии прерывания** (Interrupt line register). Является регистром «только для записи», который указывает приоритет и вектор (номер системного прерывания), которые редиректор прерываний должен присвоить данной функции. В персональных компьютерах Intel x86 значения 0x00-0x0F соответствует **IRQ0-IRQ15**.

Такая схема является достаточно гибкой, поскольку не навязывает никаких ограничений, обусловленных спецификой механизма прерываний в системе, конструктору устройств или системы, что делает возможным использование этой архитектуры в процессорных средах, отличных от Intel x86.

(Драйвер (или иное ПО), работающий с устройством PCI, определяет номер входа контроллера прерывания, доставшийся устройству (точнее, функции), чтением конфигурационного регистра **Interrupt Line**. По этому номеру определяется вектор (см. табл. 4), значение 255 означает, что номер не назначен. Номер входа каждому устройству заносит тест POST. Для этого он считывает регистр **Interrupt Pin** каждой обнаруженной функции и по номеру устройства определяет, какая из линий INTA#...INTD# (на входе коммутатора запросов) используется. Заметим, что правила, по которым на системной плате определяется соответствие между **Interrupt Pin** и входными линиями коммутатора запросов в зависимости от номера устройства, строго не регламентированы (деление номера устройства на 4 -

это всего лишь рекомендация), но их твердо знает версия BIOS данной системной платы. К этому моменту тест POST уже определил таблицу соответствия этих линий номерам входов; пользуясь этой таблицей, он записывает нужное значение в конфигурационный регистр **Interrupt Line**. Определить, есть ли еще претенденты на тот же номер прерывания, можно, лишь просмотрев конфигурационные регистры функций всех устройств, обнаруженных на шине (это не так уж сложно сделать, пользуясь функциями PCI BIOS).)

### 3. Порядок подготовки к лабораторной работе

- 3.1. Ознакомиться с программой PCI Explorer.
- 3.2. Изучить поля заголовка конфигурационного пространства одной функции устройства PCI.
- 3.3. Изучить схему формирования запросов прерываний с использованием пары PIC (Peripheral Interrupt Controller).

### 4. Порядок выполнения лабораторной работы

- 4.1. Ознакомиться с теоретическими сведениями, приведенными в методическом указании
- 4.2. Установить и запустить программу PCI Explorer. Определить типовую конфигурацию ПК с шиной PCI и составить соответствующую схему подключенных к шине PCI устройств.
- 4.3. Запустить программу PCI\_32. Записать значения полей *Int\_Pin* и *Int\_Line*, а также *Max\_Lat* и *Min\_Gnt* из конфигурационного пространства каждого устройства. Определить конфигурационное пространство для IDE-контроллера, USB-контроллера устройств, подключенных к шине. (Для того, чтобы запустить программу, откройте *Пуск->Выполнить->cmd* и *открыть pci32.exe*).
- 4.4. Составить схему коммутации запросов прерываний устройств, подключенных к шине PCI.

### 5. Содержание отчёта

- 5.1. Наименование, цель работы.
- 5.2. Составить схему конфигурации вашего ПК с шиной PCI.
- 5.3. Привести пример конфигурационного пространства для IDE-контроллера НЖД, USB-контроллера клавиатуры и мыши.
- 5.4. Составить схему коммутации запросов прерываний устройств, подключенных к шине PCI).
- 5.5. Выводы по лабораторной работе.

### 6. Контрольные вопросы

- 6.1. Сколько устройств можно подключить к шине PCI?
- 6.2. Может ли любое устройство PCI стать инициатором транзакции?
- 6.3. Могут ли несколько устройств (функций) разделять одну и ту же линию запроса прерывания?
- 6.4. Поясните назначения регистров *Int\_Pin* и *Int\_Line* в конфигурационном пространстве.

- 6.5. Поясните физический смысл параметров *Max\_Lat* и *Min\_Grant*.
- 6.6. Сколько диапазонов адресов может затребовать каждая функция в адресном пространстве памяти или в адресном пространстве ввода-вывода PCI?
- 6.7. Поясните назначение регистра конфигурационного пространства *Cache Line Size* (CL Size).

## 7. Приложения

### 7.1. Сигналы шины PCI

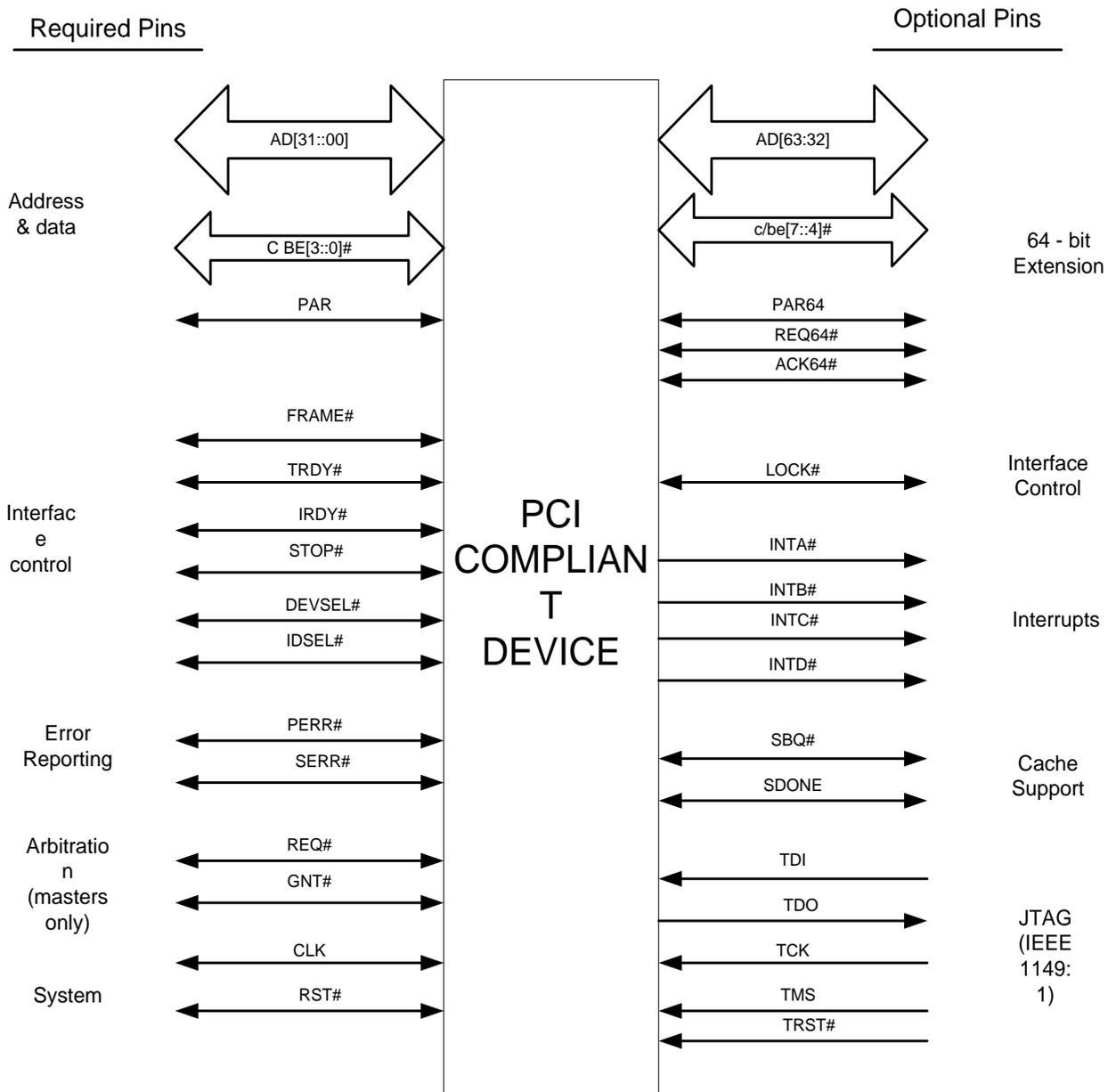


Рис. 3. Сигналы шины PCI.

## 7.2. Детальная информация об устройстве вывода графического изображения.

The screenshot shows the PCI Explorer interface with the configuration space of a graphics card selected. The tree view on the left shows the hierarchy of PCI devices, with the selected device being a function of a multi-function device (0.1.0). The configuration space table below provides detailed information about the device's registers and BARs.

Offset	Size	Name	Value	Decoded Information
00h	WORD	Vendor ID	10DEh	Nvidia Corp
02h	WORD	Device ID	0640h	Aries 16000P
04h	WORD	Command Register	0007h	Bus Master, Memory Space, I/O Space
06h	WORD	Status Register	0010h	Fast Device Select Timing, Capabilities List Implemented
08h	BYTE	Revision ID	A1h	
09h	BYTE	Programming Interface	00h	
0Ah	BYTE	Subclass Code	00h	VGA-Compatible Controller
0Bh	BYTE	Class Code	03h	Display Controller
0Ch	BYTE	Cache Line Size	08h	32 bytes
0Dh	BYTE	Latency Timer	00h	
0Eh	BYTE	Header Type	00h	
0Fh	BYTE	BIST	00h	
10h	DWORD	BAR 0	FD000000h	32-bit Memory (Base = FD000000h, Size = 00000000h)
14h	DWORD	BAR 1	C000000Ch	64-bit Prefetchable Memory (Base = 00000000h:C000000Ch, Size = 00000000h)
18h	DWORD	BAR 2	00000000h	
1Ch	DWORD	BAR 3	FA000004h	64-bit Memory (Base = 00000000h:FA000000h, Size = 00000000h)
20h	DWORD	BAR 4	00000000h	
24h	DWORD	BAR 5	00009C01h	I/O Range (Base = 00009C00h, Size = 00000000h)
28h	DWORD	Cardbus CIS Pointer	00000000h	
2Ch	WORD	Subsystem Vendor ID	10DEh	Nvidia Corp
2Eh	WORD	Subsystem Device ID	0551h	
30h	DWORD	Expansion RDM BAR	00000000h	
34h	BYTE	Capabilities Pointer	60h	
35h	BYTE	Reserved	00h	
36h	BYTE	Reserved	00h	
37h	BYTE	Reserved	00h	
38h	DWORD	Reserved	00000000h	
3Ch	BYTE	Interrupt Line	10h	IRQ16
3Dh	BYTE	Interrupt Pin	01h	INTA#
3Eh	BYTE	Min Grant	00h	
3Fh	BYTE	Max Latency	00h	

Рис. 4. Детальная информация об устройстве вывода графического изображения

### 7.3. Просмотр адресов прерываний

The screenshot shows the PCI Explorer interface. The top part is a tree view of PCI devices. The bottom part is a table showing the Configuration Space for BAR 0, with columns for Address and hex values from +00h to +0fh, plus an ASCII column.

Address	+00h	+01h	+02h	+03h	+04h	+05h	+06h	+07h	+08h	+09h	+0ah	+0bh	+0ch	+0dh	+0eh	+0fh	ASCII
00000000h	01h	00h	FDh	1Fh	00h	00h	00h	00h	01h	10h	FDh	1Fh	00h	00h	00h	00h	.....
00000010h	01h	20h	FDh	1Fh	00h	00h	00h	00h	01h	30h	FDh	1Fh	00h	00h	00h	00h	.....0.....
00000020h	01h	40h	FDh	1Fh	00h	00h	00h	00h	01h	50h	FDh	1Fh	00h	00h	00h	00h	.....@.....P.....
00000030h	01h	60h	FDh	1Fh	00h	00h	00h	00h	01h	70h	FDh	1Fh	00h	00h	00h	00h	.....p.....
00000040h	01h	80h	FDh	1Fh	00h	00h	00h	00h	01h	90h	FDh	1Fh	00h	00h	00h	00h	.....
00000050h	01h	A0h	FDh	1Fh	00h	00h	00h	00h	01h	B0h	FDh	1Fh	00h	00h	00h	00h	.....
00000060h	01h	C0h	FDh	1Fh	00h	00h	00h	00h	01h	D0h	FDh	1Fh	00h	00h	00h	00h	.....
00000070h	01h	E0h	FDh	1Fh	00h	00h	00h	00h	01h	F0h	FDh	1Fh	00h	00h	00h	00h	.....
00000080h	01h	D0h	FCh	1Fh	00h	00h	00h	00h	01h	80h	FCh	1Fh	00h	00h	00h	00h	.....
00000090h	01h	00h	EBh	1Fh	00h	00h	00h	00h	01h	10h	EBh	1Fh	00h	00h	00h	00h	.....
000000A0h	01h	20h	EBh	1Fh	00h	00h	00h	00h	01h	30h	EBh	1Fh	00h	00h	00h	00h	.....0.....
000000B0h	01h	40h	EBh	1Fh	00h	00h	00h	00h	01h	50h	EBh	1Fh	00h	00h	00h	00h	.....@.....P.....
000000C0h	01h	60h	EBh	1Fh	00h	00h	00h	00h	01h	70h	EBh	1Fh	00h	00h	00h	00h	.....p.....
000000D0h	01h	80h	EBh	1Fh	00h	00h	00h	00h	01h	90h	EBh	1Fh	00h	00h	00h	00h	.....
000000E0h	01h	A0h	EBh	1Fh	00h	00h	00h	00h	01h	B0h	EBh	1Fh	00h	00h	00h	00h	.....
000000F0h	01h	C0h	EBh	1Fh	00h	00h	00h	00h	01h	D0h	EBh	1Fh	00h	00h	00h	00h	.....
00000100h	01h	E0h	EBh	1Fh	00h	00h	00h	00h	01h	F0h	EBh	1Fh	00h	00h	00h	00h	.....
00000110h	01h	00h	ECh	1Fh	00h	00h	00h	00h	01h	10h	ECh	1Fh	00h	00h	00h	00h	.....
00000120h	01h	20h	ECh	1Fh	00h	00h	00h	00h	01h	30h	ECh	1Fh	00h	00h	00h	00h	.....0.....
00000130h	01h	40h	ECh	1Fh	00h	00h	00h	00h	01h	50h	ECh	1Fh	00h	00h	00h	00h	.....@.....P.....
00000140h	01h	60h	ECh	1Fh	00h	00h	00h	00h	01h	70h	ECh	1Fh	00h	00h	00h	00h	.....p.....
00000150h	01h	80h	ECh	1Fh	00h	00h	00h	00h	01h	90h	ECh	1Fh	00h	00h	00h	00h	.....
00000160h	01h	A0h	ECh	1Fh	00h	00h	00h	00h	01h	B0h	ECh	1Fh	00h	00h	00h	00h	.....
00000170h	01h	C0h	ECh	1Fh	00h	00h	00h	00h	01h	D0h	ECh	1Fh	00h	00h	00h	00h	.....
00000180h	01h	E0h	ECh	1Fh	00h	00h	00h	00h	01h	F0h	ECh	1Fh	00h	00h	00h	00h	.....
00000190h	01h	00h	EDh	1Fh	00h	00h	00h	00h	01h	10h	EDh	1Fh	00h	00h	00h	00h	.....
000001A0h	01h	20h	EDh	1Fh	00h	00h	00h	00h	01h	30h	EDh	1Fh	00h	00h	00h	00h	.....0.....
000001B0h	01h	40h	EDh	1Fh	00h	00h	00h	00h	01h	50h	EDh	1Fh	00h	00h	00h	00h	.....@.....P.....
000001C0h	01h	60h	EDh	1Fh	00h	00h	00h	00h	01h	70h	EDh	1Fh	00h	00h	00h	00h	.....p.....
000001D0h	01h	80h	EDh	1Fh	00h	00h	00h	00h	01h	90h	EDh	1Fh	00h	00h	00h	00h	.....
000001E0h	01h	A0h	EDh	1Fh	00h	00h	00h	00h	01h	B0h	EDh	1Fh	00h	00h	00h	00h	.....
000001F0h	01h	C0h	EDh	1Fh	00h	00h	00h	00h	01h	D0h	EDh	1Fh	00h	00h	00h	00h	.....

Рис. 5. Просмотр адресов прерываний

### 7.4. Детальная информация об IDE контроллере, подключенном к шине PCI

The screenshot shows the PCI Explorer interface. The top pane displays a tree view of PCI devices. The selected device is 0.31.2 Function (8086/2820) - Intel Corporation /, which is highlighted in blue. Below the tree view is a configuration space table for BAR 0.

Offset	Size	Name	Value	Decoded Information
00h	WORD	Vendor ID	8086h	Intel Corporation
02h	WORD	Device ID	2820h	
04h	WORD	Command Register	0007h	Bus Master, Memory Space, I/O Space
06h	WORD	Status Register	02B0h	Fast Back-to-Back Capable, 66 MHz Capable, Capabilities List Implemented
08h	BYTE	Revision ID	02h	
09h	BYTE	Programming Interface	8Fh	
0Ah	BYTE	Subclass Code	01h	IDE Controller
0Bh	BYTE	Class Code	01h	Mass Storage Controller
0Ch	BYTE	Cache Line Size	00h	
0Dh	BYTE	Latency Timer	00h	
0Eh	BYTE	Header Type	00h	
0Fh	BYTE	BIST	00h	
10h	DWORD	BAR 0	0000EC01h	I/O Range (Base = 0000EC00h, Size = 00000000h)
14h	DWORD	BAR 1	0000E881h	I/O Range (Base = 0000E880h, Size = 00000000h)
18h	DWORD	BAR 2	0000E801h	I/O Range (Base = 0000E800h, Size = 00000000h)
1Ch	DWORD	BAR 3	0000E481h	I/O Range (Base = 0000E480h, Size = 00000000h)
20h	DWORD	BAR 4	0000E401h	I/O Range (Base = 0000E400h, Size = 00000000h)
24h	DWORD	BAR 5	0000E081h	I/O Range (Base = 0000E080h, Size = 00000000h)
28h	DWORD	Cardbus CIS Pointer	00000000h	
2Ch	WORD	Subsystem Vendor ID	1043h	ASUSTeK Computer Inc
2Eh	WORD	Subsystem Device ID	81ECh	
30h	DWORD	Expansion ROM BAR	00000000h	
34h	BYTE	Capabilities Pointer	70h	
35h	BYTE	Reserved	00h	
36h	BYTE	Reserved	00h	
37h	BYTE	Reserved	00h	
38h	DWORD	Reserved	00000000h	
3Ch	BYTE	Interrupt Line	13h	IRQ19
3Dh	BYTE	Interrupt Pin	02h	INTB#
3Eh	BYTE	Min Grant	00h	
3Fh	BYTE	Max Latency	00h	

Рис. 6. Детальная информация об IDE контроллере, на шине PCI

## 7.5. Результат выполнения программы PCI32

```

C:\WINDOWS\system32\cmd.exe
Address 4 is an I/O Port : 0000D800h
System IRQ 19, INT# B

Bus 0 <PCI Express>, Device Number 29, Device Function 2
Vendor 8086h Intel Corporation
Device 2832h 82801H <ICH8 Family> USB UHCI #3
Command 0005h <I/O Access, BusMaster>
Status 0280h <Supports Back-To-Back Trans., Medium Timing>
Revision 02h, Header Type 00h, Bus Latency Timer 00h
Self test 00h <Self test not supported>
PCI Class Serial, type USB <UHCI>
Subsystem ID 81EC1043h Unknown
Subsystem Vendor 1043h ASUSTeK Computer Inc
Address 4 is an I/O Port : 0000D880h
System IRQ 18, INT# C

Bus 0 <PCI Express>, Device Number 29, Device Function 7
Vendor 8086h Intel Corporation
Device 2836h 82801H <ICH8 Family> USB2 EHCI #1
Command 0006h <Memory Access, BusMaster>
Status 0290h <Has Capabilities List, Supports Back-To-Back Trans., Medium Timin
g>
Revision 02h, Header Type 00h, Bus Latency Timer 00h
Self test 00h <Self test not supported>
PCI Class Serial, type USB 2.0 <EHCI>
Subsystem ID 81EC1043h Unknown
Subsystem Vendor 1043h ASUSTeK Computer Inc
Address 0 is a Memory Address (anywhere in 0-4Gb) : FEBFF800h
System IRQ 23, INT# A
New Capabilities List Present:
  Power Management Capability, Version 1.1
    Does not support low power State D1 or D2
    Supports PME# signalling from mode(s) D0, D3hot, D3cold
    PME# signalling is currently disabled
    Current Power State : D0 <Device operational, no power saving>
    3.3v AUX Current required : 375mA
  USB 2.0 EHCI Debug Port Capability

Bus 0 <PCI Express>, Device Number 30, Device Function 0
Vendor 8086h Intel Corporation
Device 244Eh 82801HB Hub Interface to PCI Bridge <ICH8 B0 step>
Command 0107h <I/O Access, Memory Access, BusMaster, System Errors>
Status 0010h <Has Capabilities List, Fast Timing>
Revision F2h, Header Type 01h, Bus Latency Timer 00h
Self test 00h <Self test not supported>
PCI Class Bridge, type PCI to PCI <Subtractive Decode>
PCI Bridge Information:
  Primary Bus Number 0, Secondary Bus Number 5, Subordinate Bus Number 5
  Secondary Bus Command 0002h (<)
  Secondary Bus Status 2280h <Supports Back-To-Back Trans., Received Master Abo
rt, Medium Timing>
  Secondary Bus Latency 20h
  I/O Port Range Passed to Secondary Bus : None
  Memory Range Passed to Secondary Bus : None
  Prefetchable Memory Range Passed to Secondary Bus : None
New Capabilities List Present:
  Subsystem ID & Subsystem Vendor ID Capability
    SSVID : 1043h
    SSID : 81ECh

Bus 0 <PCI Express>, Device Number 31, Device Function 0
Vendor 8086h Intel Corporation
Device 2810h 82801HB/HR <ICH8/R> LPC Interface Controller
Command 0007h <I/O Access, Memory Access, BusMaster>
Status 0210h <Has Capabilities List, Medium Timing>
Revision 02h, Header Type 80h, Bus Latency Timer 00h
Self test 00h <Self test not supported>
PCI Class Bridge, type PCI to ISA
Subsystem ID 81EC1043h Unknown
Subsystem Vendor 1043h ASUSTeK Computer Inc
New Capabilities List Present:
  Vendor-Dependant Capability

Bus 0 <PCI Express>, Device Number 31, Device Function 2
Vendor 8086h Intel Corporation
Device 2820h 82801H <ICH8 Family> 4 port SATA Controller
Command 0007h <I/O Access, Memory Access, BusMaster>
Status 02B0h <Has Capabilities List, Supports 66MHz, Supports Back-To-Back Tran

```

Рис. 7. Результат выполнения программы PCI32

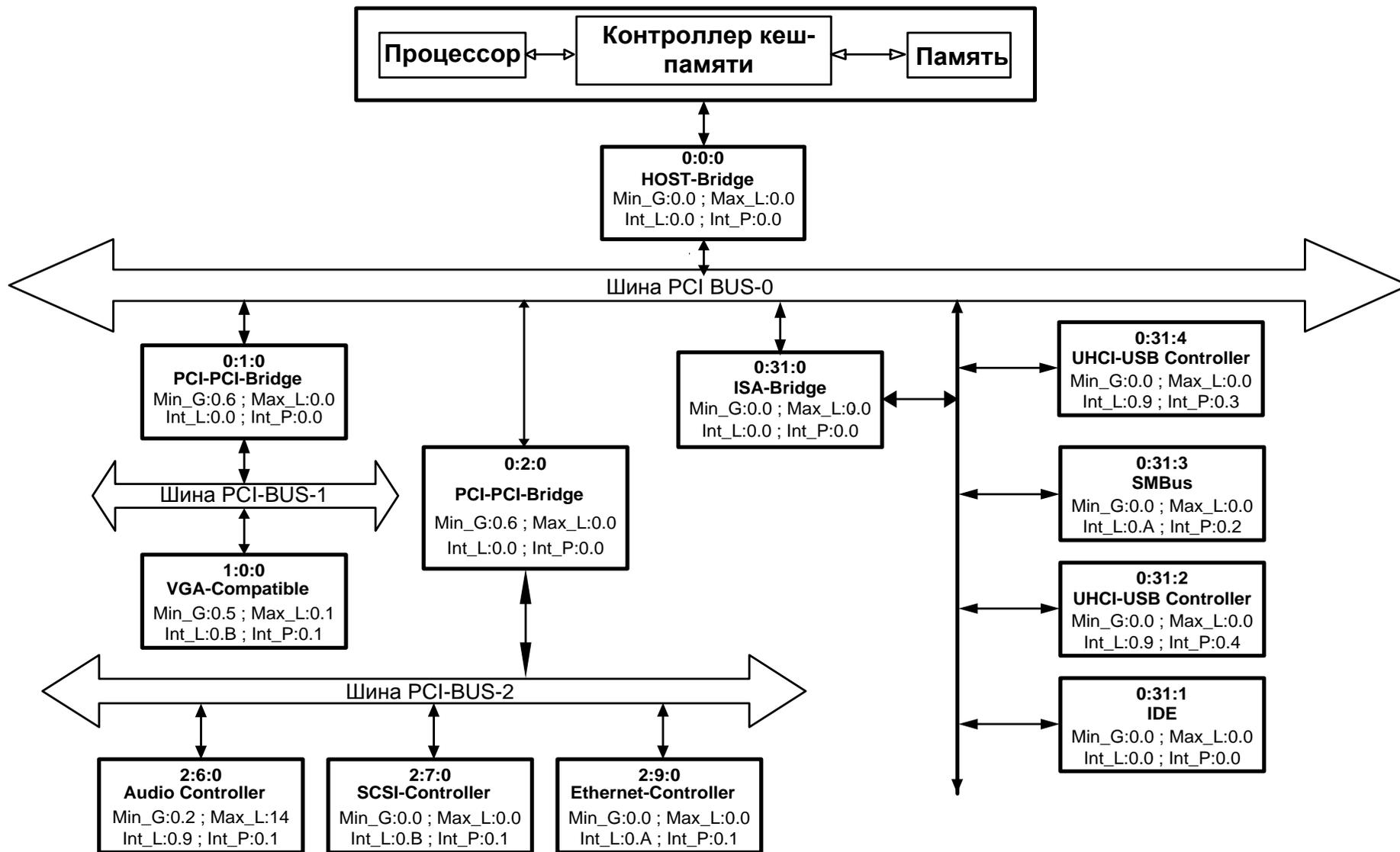


Рис. 8. Пример конфигурации ПК с шиной PCI

## **8. Литература**

- 8.1. Скотт Мюллер. Модернизация и ремонт ПК. 19-е издание.  
Издательство: Вильямс, 2011.
- 8.2. Э.Таненбаум, Т. Остин. Архитектура компьютера. 6-е издание.  
Издательство Питер, 2013г.
- 8.3. [Ru.wikipedia.org/wiki/PCI](http://Ru.wikipedia.org/wiki/PCI)
- 8.4. [Thg.ru/mainboard/pci\\_express\\_3.0](http://Thg.ru/mainboard/pci_express_3.0)