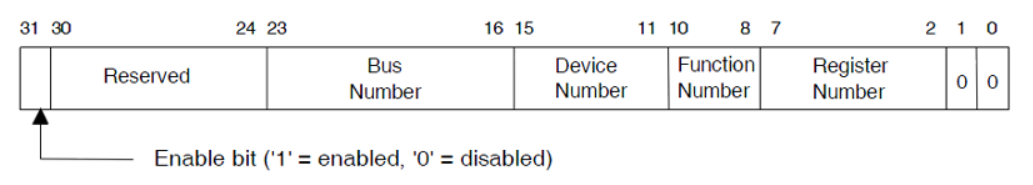
Лабораторная работа №2 – Конфигурационное пространство PCI

**Мост pci – для того, чтобы соединять основную шину pci с дополнительными**

**Конфигурационное пространство шины pci (просит всех рисовать это на доске), индекс регистра (register number на пикче ниже) в лабе - 0 (в этом регистре 32 бита, первые 16 это vendor id, старшие 16 – device id, мы их получаем сдвигами).**

[**https://wiki.osdev.org/PCI**](https://wiki.osdev.org/PCI) **тут можно подробнее почитать**

**Для того, чтобы получить содержимое регистра команды, сначала мы записываем адрес регистра команды в один порт (не помню название, но это не так важно), после этого читаем содержимое из другого порта, там как раз таки лежат нужные нам device id и vendor id.**

****

# Определение шины PCI. Отличия от шины PCI-Х, PCI-Express.

PCI (Peripheral Components Interconnect) – базовая системная магистраль (шина) компьютера архитектуры x86, предназначенная для подключения внутренних периферийных устройств и контроллеров внешних интерфейсов, таких как видеокарты, звуковые карты, сетевые адаптеры и другие периферийные устройства к материнской плате.

PCI (Peripheral Component Interconnect), PCI-X (PCI Extended), и PCI Express (PCIe) представляют различные поколения аппаратных шин для подключения устройств к материнской плате компьютера. Вот основные отличия между ними:

Напряжение частота разрядность пропускная способность

|  |  |  |  |
| --- | --- | --- | --- |
| **показатель** | **PCI** | **PCI-X** | **PCI-Express** |
| Пропускная способность, Мб/с | 133 МБ/с (PCI 2.3) 266 МБ/с (PCI 3.0)  533 PCI 66  133-533 | 1 ГБ/с – 4 ГБ/с | Каждая полоса по  250 МБ/с, при шине на 16 полос – до 4 ГБ/с pcie 1.0, pcie 5.0 – до 63 (4 почти 4 гб на полосе) |
| Физический интерфейс | 32 или 64-битная | 64-битный | x1, x2, x4, x8, x16 и x32, где xN – кол- во линий передачи |
| Обратная совместимость | не совместим с более новыми стандартами  PCI-X и PCIe без использования адаптеров | Частичная поддержка PCI, но с  ограничением пропускной способности | могут поддерживать устройства с меньшим количеством "полос" |
| Тактовая частота | 33-66 | от 66 МГц до 533 МГц | От 2.5 до 32 гт/c |
| Физическое размещение | классические разъемы, | классические разъемы,  подобные | более компактные разъемы,  обеспечивающие |

|  |  |  |  |
| --- | --- | --- | --- |
|  | подобные слотам для расширения | слотам для расширения | поддержку различных конфигурай |
| Распространение использования | Устарел и редко используется в современных компьютерах | Используется в некоторых серверных системах и старших рабочих станциях | сейчас является наиболее распространенным стандартом, используемым в большинстве современных компьютеров и серверов |
| Напряжение | 3.3 или 5 (зависит от стандарта) | 1.5 или 3.3 | 0.8 |
| Адресация | 32 или 64 бита  Не зависит от разрядности |  |  |
| топология | Многоуровневая шина | Шина? | звезда |

# Какие новые технологии и интерфейсы конкурируют с PCI в современных компьютерах?

1. NVMe (Non-Volatile Memory Express): NVMe - это протокол для обращения к накопителям с использованием интерфейса PCIe. Он предназначен специально для работы с современными SSD, и обеспечивает гораздо более высокие скорости передачи данных по сравнению с традиционными интерфейсами SATA.
2. USB 4: Этот интерфейс объединяет функции Thunderbolt и USB. Он обеспечивает высокую пропускную способность и поддерживает множество различных устройств, включая дисплеи, хранилища данных, звуковые карты и другие периферийные устройства.
3. Thunderbolt 4: Это новое поколение интерфейса Thunderbolt, который предоставляет высокую скорость передачи данных (до 40 Гбит/с) и поддерживает различные функции, включая подключение мониторов, зарядку устройств и передачу данных.
4. M.2 (NGFF): Это форм-фактор и интерфейс для подключения накопителей, который использует интерфейсы SATA, PCIe или USB. Он часто применяется для установки SSD и Wi-Fi модулей.
5. SATA Express: Этот интерфейс сочетает в себе преимущества PCIe и SATA, обеспечивая более высокую пропускную способность для подключения SSD и других устройств хранения данных.
6. U.2 (SFF-8639): Представляет собой стандарт для подключения накопителей с использованием интерфейса PCIe. U.2 поддерживает более высокие скорости передачи данных и предназначен для устройств хранения данных большой емкости.
7. Wi-Fi 6 (802.11ax): Новое поколение беспроводной сети, обеспечивающее более высокие скорости передачи данных и более эффективное управление сетью.
8. Ethernet - проводной протокол передачи данных в локальной сети интернет.

# Какие устройства могут быть подключены к шине PCI, и какие устройства являются типичными PCI-устройствами?

# Видеокарты, звуковые карты, сетевые карты, контроллеры (накопителей, usb), карты захвата, расширительные карты io.

1. Видеокарты (графические адаптеры): Видеокарты обрабатывают графическую информацию и отображают ее на мониторе. В начале своего развития PCI был основным интерфейсом для видеокарт, но впоследствии был заменен более высокопроизводительным PCIe. Сначала видеокарта связывалась через PCI с южным мостом, а затем через PCIe с северным и южным.
2. Звуковые карты: Звуковые карты обрабатывают аудиосигналы и обеспечивают звуковой вывод и ввод. PCI был популярным интерфейсом для звуковых карт до появления встроенного аудио на материнских платах.
3. Сетевые адаптеры (сетевые карты): Сетевые адаптеры позволяют компьютеру подключаться к сетям, в том числе к локальным сетям (LAN) и Интернету. PCI был широко использован для этого типа устройств.
4. Контроллеры накопителей (IDE, SATA, SCSI): Контроллеры накопителей управляют подключенными жесткими дисками и другими устройствами хранения данных.
5. Контроллеры USB и FireWire: Эти контроллеры позволяют подключать устройства через интерфейсы USB и FireWire.
6. Карты захвата видео и аудио: Эти устройства позволяют записывать и обрабатывать видео и аудио с различных источников.
7. Расширительные карты I/O (ввода/вывода): Эти карты предоставляют дополнительные порты и интерфейсы, такие как порты USB, COM, LPT и другие.
8. Kонтроллеры RAID: Они управляют массивами жестких дисков, обеспечивая более высокую производительность и надежность хранения данных.

# Какие виды операций можно выполнять с устройствами, подключенными к шине PCI?

# Чтение, запись, инициализация и настройка, обработка прерываний, настройка dma, организация данных, управление питанием

1. Чтение данных (Input): Программа может читать данные с устройства, чтобы получить информацию о его состоянии, параметрах или результаты выполненной работы.
2. Запись данных (Output): Программа может отправлять данные на устройство, чтобы управлять его работой или задавать определенные параметры.
3. Инициализация и настройка (Initialization and Configuration): Некоторые устройства требуют инициализации при запуске системы. Это может включать в себя настройку параметров, установку режимов работы и так далее.
4. Обработка прерываний (Interrupt Handling): Устройства могут генерировать прерывания для сообщения о событиях, которые требуют внимания ЦПУ. Это позволяет системе реагировать на различные события, такие как завершение операций или обнаружение ошибок.
5. Организация данных (Data Management): Устройства могут взаимодействовать с оперативной памятью для чтения и записи данных. Например, видеокарта может загружать текстуры в память видеокарты для последующего использования при рендеринге.
6. Управление питанием (Power Management): Некоторые устройства поддерживают режимы энергосбережения, которые позволяют им переходить в режимы низкого энергопотребления, когда они не используются.
7. Конфигурация DMA (Direct Memory Access): Устройства могут использовать DMA для прямой передачи данных между памятью и устройством без участия ЦПУ.
8. Контроль состояния устройства (Device Status Control): Это включает в себя проверку состояния устройства, определение его готовности к работе и т. д.
9. Контроль прерываний и запросов (Interrupt and Request Handling): Устройства могут использовать систему прерываний для связи с ЦПУ и запроса внимания.

# Доступ к шине PCI и фазы транзакции.

Доступ к шине PCI осуществляется с помощью так называемых "транзакций". Транзакция - это базовая единица обмена данными между устройствами, подключенными к шине PCI.

Доступ к шине: 3 фазы адресная – адрес и команда, данных – от 1 до 3 при необходимости, конец – посылает устройству сигнал о завершении.

Существует несколько основных фаз транзакции в шине PCI:

1. F0 (Address Phase): В этой фазе инициатор (например, процессор или устройство) отправляет адрес и команду на шину. Адрес указывает на устройство или функцию, с которой будет происходить обмен данными.
2. F1 (Data Phase): Если транзакция предполагает передачу данных (например, чтение или запись), в этой фазе данные передаются по шине. Если это операция чтения, устройство, к которому обращаются, возвращает запрошенные данные.
3. F2 (Data Phase): В некоторых случаях, особенно при передаче данных в направлении от устройства к инициатору (например, когда устройство записывает данные в память), может потребоваться дополнительная фаза Data Phase.
4. F3 (Data Phase): Аналогично, в некоторых сценариях, например, при взаимодействии с 64-битными устройствами, может потребоваться дополнительная фаза Data Phase.
5. F4 (Disconnect Phase): После завершения передачи данных, происходит завершающая фаза, в которой устройство завершает транзакцию и освобождает шину.
6. F5 (End of Transaction): Эта фаза сигнализирует о завершении транзакции. То есть:

Транзакция на шине PCI состоит из нескольких фаз. Фактически фаза соответствует одному тактовому интервалу, за время которого по шине передается одна группа данных (32 или 64 бита). В транзакции обязательно имеется одна фаза адресации и одна или несколько фаз передачи данных.

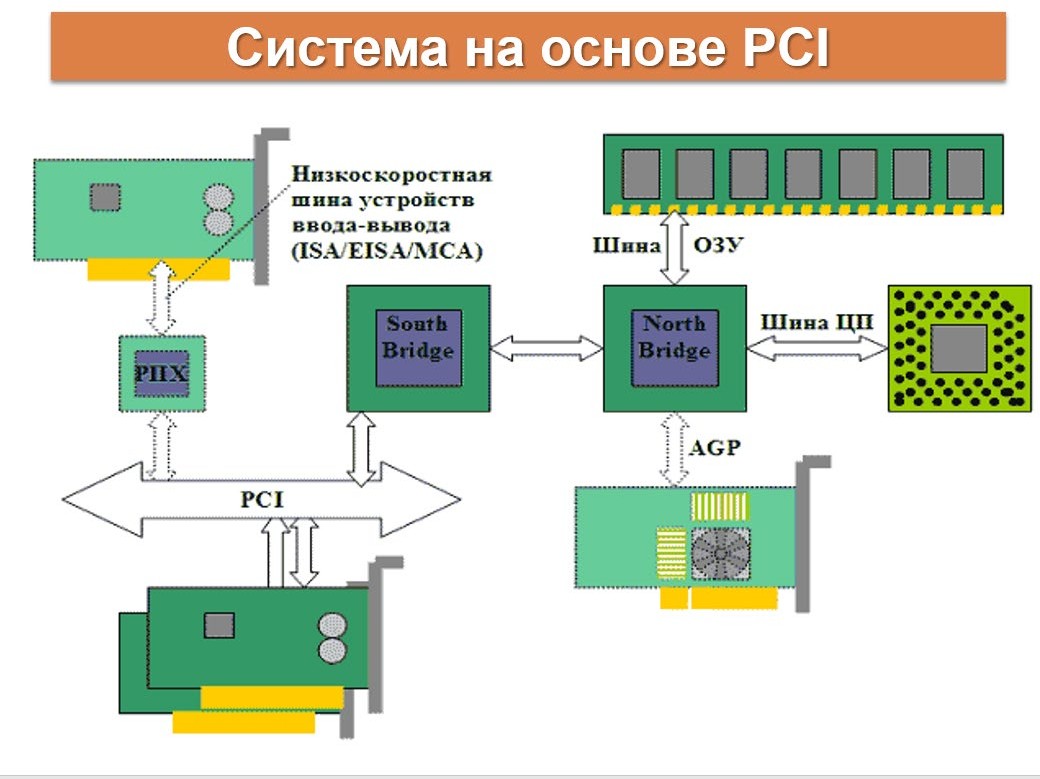
До начала транзакции устройство-инициатор подает запрос на доступ к шине. Для этого используется линия REQ#, подведенная к каждому устройству от арбитра шины.

Арбитр анализирует пришедшие запросы и одному из устройств выдает разрешение – низкий уровень на линии GNT# (линии REQ# и GNT# разведены попарно).

Устройство, получившее разрешение, ожидает окончания текущей транзакции (снятие сигнала FRAME#), после чего начинает работу.

# Схема подключения ПУ к системной шине.

**ПУ -> ШИНА PCI -> МОСТ (южный) -> СИСТЕМНАЯ ШИНА**



# Классификация ПУ.

ПУ можно также классифицировать по другим признакам, например, по конструктивному исполнению:

Внешние – имеющие свой корпус и (зачастую) отдельный источник питания (если питание по интерфейсу не предусмотрено или его мощности недостаточно).

Внутренние – расположенные внутри корпуса системы и питающиеся от системного блока питания или интерфейса.

Встроенные – расположенные на системной (материнской) плате или являющиеся частью одной из микросхем на этой плате.

Общепринятой является классификация по основной функции ПУ. При этом в одном физическом корпусе могут объединяться несколько устройств различного класса (монитор со встроенными колонками и камерой, клавиатура с IP-телефоном, МФУ с принтером, сканером и факс-аппаратом и т.д.).

Каждому классу устройств присущ свой набор характеристик

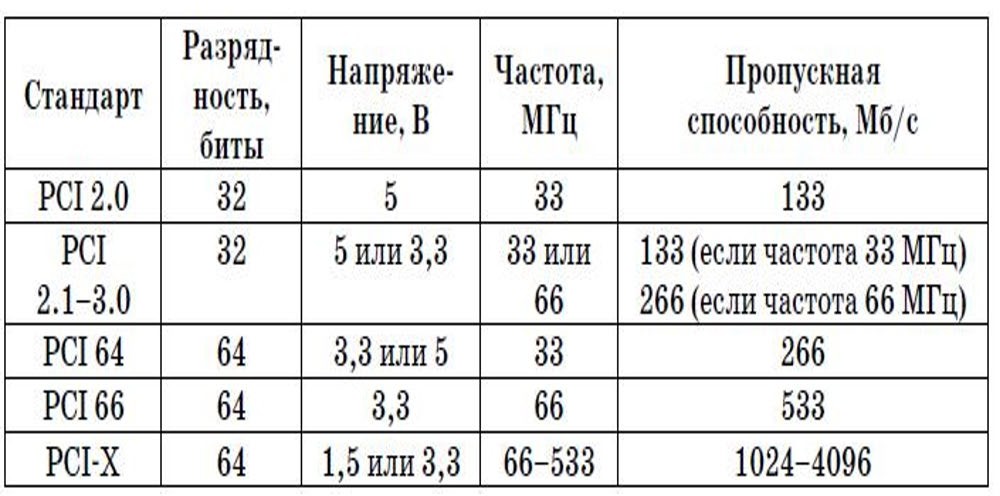
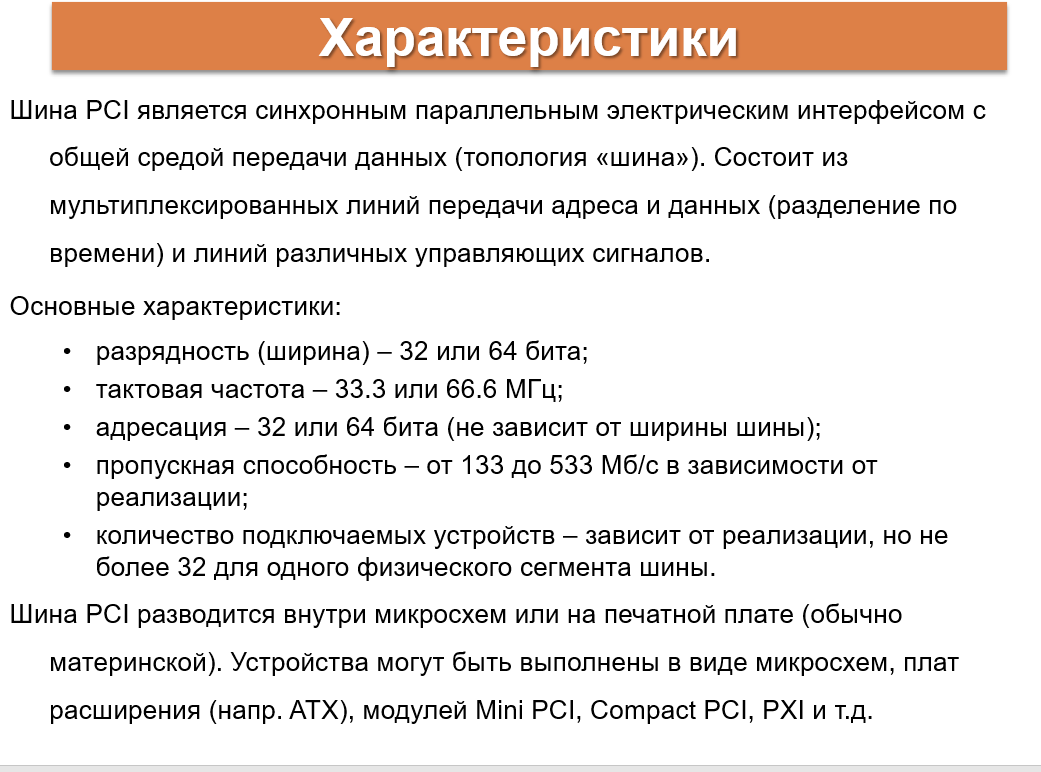
# Методы управления вводом/выводом.

Методы управления вводом/выводом (I/O) представляют различные подходы к организации обмена данными между компьютером и его периферийными устройствами. Вот некоторые из основных методов:

Методы управление – программное процессор управляет напрямую, прерывание – устройство отправляет сигнал прерывания, когда готово к обмену данных, программно-прерывательное, dma – прямой доступ к памяти без процессора, mmaped io, ipc.

1. Программное Управление (Programmed I/O): В этом методе ЦПУ напрямую управляет операциями ввода/вывода. Она посылает команды устройству и ожидает завершения операции.
2. Прерывания (Interrupt-Driven I/O): Устройство генерирует сигнал прерывания, когда оно готово к обмену данными. Это уведомляет ЦПУ, что данные готовы к чтению или записи.
3. Программно-прерывательное управление (Polling with Interrupts): Комбинирует программное управление и прерывания. Операции ввода/вывода проверяются периодически в цикле программы, но если устройство готово к операции, оно может сгенерировать прерывание для уведомления ЦПУ.
4. Доступ к памяти напрямую (Direct Memory Access - DMA): Устройство получает прямой доступ к оперативной памяти без участия ЦПУ. Оно может читать и записывать данные непосредственно в память, что ускоряет операции ввода/вывода.
5. Многоканальный контроллер (Multi-Channel Controllers): Используется для организации параллельных операций ввода/вывода, что может повысить эффективность обмена данными.
6. Обмены между процессами (Interprocess Communication - IPC): В многозадачных и многопроцессорных системах, процессы могут взаимодействовать друг с другом через различные методы, такие как сокеты, каналы и разделяемая память.
7. Memory-Mapped I/O: В этом методе устройства отображают свои регистры и буферы в адресное пространство памяти компьютера. ЦПУ может читать и записывать данные в эти адреса, что упрощает взаимодействие с устройствами.
8. I/O-порты (I/O Ports): В некоторых архитектурах (например, x86), существуют специальные команды для чтения и записи данных в определенные порты, предназначенные для обмена с периферийными устройствами.

# Характеристики шины PCI.



1. **Основные принципы программирования доступа к периферийным устройствам**

# ГПТШКА:

Перед началом программирования необходимо изучить документацию на периферийное устройство, чтобы понимать его характеристики, регистры, команды и прочее.

**Инициализация и Конфигурация:**

**Знание Спецификаций Устройства:**

* 1. Перед началом работы с устройством, оно часто требует инициализации. Это включает в себя установку параметров, настройку режимов работы и так далее.

**Управление Режимами Работы:**

* 1. Многие устройства имеют различные режимы работы. Например, видеокарта может иметь режимы разного разрешения и частоты обновления. Программист должен знать, как переключаться между ними.

**Обработка Ошибок и Исключений:**

* 1. Всегда предусматривайте обработку

ошибок, так как периферийные устройства могут генерировать ошибки в процессе работы.

* 1. Это включает в себя правильное

**Эффективное Использование Ресурсов:**

распределение памяти, минимизацию обращений к устройству, а также эффективное использование каналов ввода/вывода.

* 1. Во многих операционных

**Использование Системных Вызовов и API:**

системах существуют специальные API и системные вызовы для взаимодействия с устройствами. Используйте их, чтобы упростить программирование и обеспечить совместимость с разными устройствами.

* 1. Если несколько частей программы работают с одним устройством одновременно, уделяйте внимание синхронизации доступа, чтобы избежать конфликтов.

**Документирование Кода:**

**Правильная Синхронизация:**

* 1. Хорошая документация к коду помогает другим разработчикам (и вам самим в будущем) понимать, как работает программа взаимодействия с устройством.

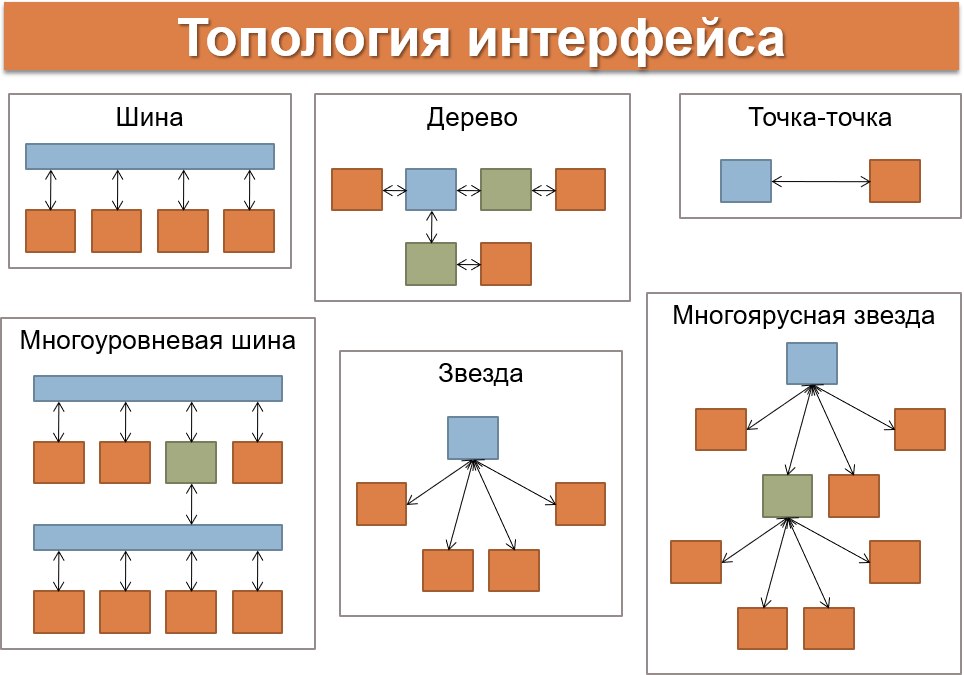
**Тестирование и Отладка:**

* 1. Важно тестировать программу на различных условиях и обработать крайние случаи. Отладка также является важной частью процесса разработки.
  2. Всегда убедитесь, что ваши

**Соблюдение Принципов Безопасности:**

программы работают в пределах разрешений и прав, и не предоставляют нежелательный доступ к системным ресурсам.

# Топологии интерфейса



* 1. **Режимы кэширования шины PCI.**

Шина PCI (Peripheral Component Interconnect) поддерживает несколько режимов кэширования данных для оптимизации работы с периферийными устройствами. Вот основные режимы кэширования шины PCI:

1. Write-Through (Прямая запись): В этом режиме все записи в память выполняются непосредственно в основную системную память. Кэш используется только для чтения данных. Этот режим обеспечивает наивысшую надежность и соответствие данным в кэше и основной памяти.
2. Write-Back (Отсроченная запись): Записи сначала попадают в кэш, а затем с определенной задержкой записываются в основную память. Это уменьшает количество обращений к памяти, что может повысить производительность.
3. Write-Combined (Совмещенная запись): В этом режиме записи сначала кэшируются, а затем передаются в память в пачках (чаще всего в размере 32 или 64 байта). Это подходит для сценариев, где происходит множество последовательных записей.
4. Non-Cacheable (Некэшируемый): В этом режиме операции чтения и записи не кэшируются, что обеспечивает точное взаимодействие с периферийными устройствами и памятью.
5. Prefetchable (Предварительное извлечение): Этот режим предполагает, что данные будут использованы в ближайшем будущем, поэтому они предварительно извлекаются в кэш для более быстрого доступа.

Выбор конкретного режима зависит от требований к производительности и надежности системы, а также от характеристик конкретного периферийного устройства. Например, для видеокарты, которая работает с большими объемами данных, может быть выбран режим Write-Back для повышения производительности. Однако для устройства хранения данных, такого как жесткий диск, может быть предпочтителен режим Write-Through для обеспечения надежности и согласованности данных.

# Определение понятия интерфейса.

Средства (аппаратные и программные), используемые для соединения двух компонентов или систем, называются интерфейсом.

В компьютерной архитектуре - это точка взаимодействия между различными компонентами системы.

# Определение транзакции.

Атомарная операция обмена данными между двумя устройствами PCI называется транзакцией. В рамках транзакции определены два объекта – инициатор обмена (Initiator) и целевое устройство (Target). В рамках одной физической шины в конкретный момент может происходить только одна транзакция. Если физических шин несколько, то транзакции на них могут выполняться одновременно (Peer Concurrency), если пути прохождения данных не пересекаются.

# Какие функции выполняют PCI-мосты, и почему они важны в архитектуре компьютера?

1. Преобразование протоколов: PCI-мосты предоставляют перевод между различными протоколами шин. Например, они могут связывать устройства с интерфейсами PCI с шиной AGP или PCI-Express.
2. Разделение трафика: Они могут разделять трафик, обеспечивая, что данные поступают в нужное устройство, и предотвращая конфликты.
3. Обеспечение совместимости: PCI-мосты позволяют устройствам с разными интерфейсами взаимодействовать в одной системе. Например, они позволяют подключать новые PCI-Express устройства к системам, основанным на старых PCI или AGP.
4. Управление прерываниями: PCI-мосты также могут обрабатывать прерывания, что существенно снижает нагрузку на процессор, особенно в системах с множеством подключенных устройств.
5. Поддержка многопроцессорных систем: В многопроцессорных системах мосты имеют дополнительные функции для управления и синхронизации доступа к ресурсам.
6. Важность PCI-мостов в архитектуре компьютера заключается в том, что они позволяют эффективно интегрировать различные компоненты и устройства в единую систему. Без них было бы крайне сложно подключать и взаимодействовать с устройствами, использующими разные протоколы и интерфейсы. Это делает PCI-мосты важными компонентами для обеспечения совместимости и расширяемости компьютерных систем.