**Определение периферийного устройства:** Устройства ЭВМ, предназначенные для восприятия и выдачи информации объектам внешнего мира, называются периферийными.

**Перечислите основные типы ПУ:** устройства ввода, устройства вывода, внешние запоминающие устройства.

**Основные принципы работы компьютера Фон-Неймана:**

*Архитектура фон Неймана* ([англ.](http://ru.wikipedia.org/wiki/%D0%90%D0%BD%D0%B3%D0%BB%D0%B8%D0%B9%D1%81%D0%BA%D0%B8%D0%B9_%D1%8F%D0%B7%D1%8B%D0%BA) Von Neumann architecture) — широко известный принцип совместного хранения [программ](http://ru.wikipedia.org/wiki/%D0%9A%D0%BE%D0%BC%D0%BF%D1%8C%D1%8E%D1%82%D0%B5%D1%80%D0%BD%D0%B0%D1%8F_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B0) и [данных](http://ru.wikipedia.org/wiki/%D0%94%D0%B0%D0%BD%D0%BD%D1%8B%D0%B5_%28%D0%B2%D1%8B%D1%87%D0%B8%D1%81%D0%BB%D0%B8%D1%82%D0%B5%D0%BB%D1%8C%D0%BD%D0%B0%D1%8F_%D1%82%D0%B5%D1%85%D0%BD%D0%B8%D0%BA%D0%B0%29) в [памяти](http://ru.wikipedia.org/wiki/%D0%9F%D0%B0%D0%BC%D1%8F%D1%82%D1%8C_%28%D0%BA%D0%BE%D0%BC%D0%BF%D1%8C%D1%8E%D1%82%D0%B5%D1%80%D0%BD%D0%B0%D1%8F%29) [компьютера](http://ru.wikipedia.org/wiki/%D0%9A%D0%BE%D0%BC%D0%BF%D1%8C%D1%8E%D1%82%D0%B5%D1%80).

1. Принцип программного управления.

Программа состоит из набора команд, которые выполняются процессором друг за другом в определенной последовательности.

2. Принцип однородности памяти.

Как программы (команды), так и данные хранятся в одной и той же памяти (и кодируются в одной и той же системе счисления - чаще всего [двоичной](http://ru.wikipedia.org/wiki/%D0%94%D0%B2%D0%BE%D0%B8%D1%87%D0%BD%D0%B0%D1%8F_%D1%81%D0%B8%D1%81%D1%82%D0%B5%D0%BC%D0%B0_%D1%81%D1%87%D0%B8%D1%81%D0%BB%D0%B5%D0%BD%D0%B8%D1%8F)). Над командами можно выполнять такие же действия, как и над данными.

3. Принцип адресуемости памяти.

Структурно основная память состоит из пронумерованных ячеек; процессору в произвольный момент времени доступна любая ячейка.

Компьютеры, построенные на этих принципах, относят к типу фоннеймановских.

**Обоснуйте важную роль канала ввода-вывода:** СУО - Система обменом информацией представляется как совокупность аппаратно-програмных средств, обеспечивающих выполнение операций обработки и ввода/вывода информации. Каналы ввода-вывода ([англ.](http://ru.wikipedia.org/wiki/%D0%90%D0%BD%D0%B3%D0%BB%D0%B8%D0%B9%D1%81%D0%BA%D0%B8%D0%B9_%D1%8F%D0%B7%D1%8B%D0%BA) IOC *- input-output channel*), и [интерфейсы](http://ru.wikipedia.org/wiki/%D0%98%D0%BD%D1%82%D0%B5%D1%80%D1%84%D0%B5%D0%B9%D1%81) обеспечивают взаимодействие центральных устройств [машины](http://ru.wikipedia.org/wiki/%D0%AD%D0%92%D0%9C) и периферийных устройств. КВВ и интерфейсы выполняют следующие функции

1) Позволяют иметь машины с переменным составом периферийных устройств.  
2) Обеспечивают параллельную работу периферийных устройств как между собой, так и по отношению к процессору.  
3) Обеспечивают автоматическое распознование и реакцию процессора на различные ситуации, возникающие в периферийных устройствах.

**Основное назначение периферийных устройств:** Периферийные устройства предназначены для внешней обработки данных, обеспечивающий их подготовку, ввод, хранение, управление, защиту, вывод и передачу на расстояние по каналам связи.

**Приведите еще примеры периферийных устройств в каждом из трех классов.**

[Устройства ввода](http://ru.wikipedia.org/wiki/%D0%A3%D1%81%D1%82%D1%80%D0%BE%D0%B9%D1%81%D1%82%D0%B2%D0%B0_%D0%B2%D0%B2%D0%BE%D0%B4%D0%B0)

[Клавиатура](http://ru.wikipedia.org/wiki/%D0%9A%D0%BB%D0%B0%D0%B2%D0%B8%D0%B0%D1%82%D1%83%D1%80%D0%B0)

[Мышь](http://ru.wikipedia.org/wiki/%D0%9A%D0%BE%D0%BC%D0%BF%D1%8C%D1%8E%D1%82%D0%B5%D1%80%D0%BD%D0%B0%D1%8F_%D0%BC%D1%8B%D1%88%D1%8C), [трекбол](http://ru.wikipedia.org/wiki/%D0%A2%D1%80%D0%B5%D0%BA%D0%B1%D0%BE%D0%BB) или [тачпад](http://ru.wikipedia.org/wiki/%D0%A2%D0%B0%D1%87%D0%BF%D0%B0%D0%B4)

[Джойстик](http://ru.wikipedia.org/wiki/%D0%94%D0%B6%D0%BE%D0%B9%D1%81%D1%82%D0%B8%D0%BA)

[Сканер](http://ru.wikipedia.org/wiki/%D0%A1%D0%BA%D0%B0%D0%BD%D0%B5%D1%80)

[Устройства вывода](http://ru.wikipedia.org/w/index.php?title=%D0%A3%D1%81%D1%82%D1%80%D0%BE%D0%B9%D1%81%D1%82%D0%B2%D0%B0_%D0%B2%D1%8B%D0%B2%D0%BE%D0%B4%D0%B0&action=edit&redlink=1)

[Монитор (дисплей)](http://ru.wikipedia.org/wiki/%D0%9C%D0%BE%D0%BD%D0%B8%D1%82%D0%BE%D1%80_%28%D0%B4%D0%B8%D1%81%D0%BF%D0%BB%D0%B5%D0%B9%29)

Колонки/наушники

Печатающие устройства

[Принтер](http://ru.wikipedia.org/wiki/%D0%9F%D1%80%D0%B8%D0%BD%D1%82%D0%B5%D1%80)

[Плоттер](http://ru.wikipedia.org/wiki/%D0%9F%D0%BB%D0%BE%D1%82%D1%82%D0%B5%D1%80) (графопостроитель)

[Модем](http://ru.wikipedia.org/wiki/%D0%9C%D0%BE%D0%B4%D0%B5%D0%BC) — для связи по телефонной линии

**Какая возможна реализация КВВ?**

Реализация КВВ может быть как программной, так и аппаратной.

При программной реализации КВВ процессор должен иметь возможность переключаться на выполнение функций КВВ по сигналу о готовности от ПУ.

При аппаратной реализации КВВ процессор должен иметь возможность получать информацию от канала о ходе выполнения операций ввода-вывода и об их завершении.

**Приведите примерное соотношение количеств мультиплексного и селекторного каналов у ЭВС средней производительности.**

У ЭВС средней производительности имеется, как правило, один мультиплексный и от двух до шести селекторных каналов в зависимости от модели. К каждому каналу может быть подключено до 256 ПУ (рисунок 1.5.3.). Если МКВВ может работать как в мультиплексном, так и в монопольном режиме, то СКВВ всегда работает в монопольном режиме.

**Назовите те ПУ, которые Вам знакомы и отнесите их к одной из трех категорий.**

*Н:* Мышь/клавиатура

К низкоскоростным ПУ относятся оперативные устройства ручного ввода (клавиатура, мышь, трекбол, световое перо) и средства отображения (дисплеи, объемные голографические средства отображения, световые табло), а также устройства речевого ввода-вывода.

*С:* Модем/Принтер

К среднескоростным ПУ относятся считыватели графиков (планшеты, сканеры) и регистрирующие устройства (принтеры, графопостроители). Сюда же мы отнесем модемы - устройства, модулирующие и демодулирующие сигналы, передаваемые с помощью средств связи в случае использования каналов связи для коммутации ЭВС в вычислительных сетях, хотя по сути они занимают промежуточное значение между низкоскоростными и среднескоростными устройствами ввода-вывода.

*В:* Монитор/ HDTV

К высокоскоростным ПУ относятся накопители на магнитных носителях и лазерные носители информации.

Цикл работы ПУ – это время, необходимое для формирования кванта информации в ПУ и передачи его в ОЗУ или обратно. Независимо от типа ПУ, этот процесс можно представить в виде суммы двух интервалов:

,

где Тпод - длительность цикла подготовки и преобразования информации в ПУ, Тпер- длительность цикла пересылки подготовленного кванта информации между ПУ и ОЗУ. Заведомо Тпод>>Тпер , и именно эта особенность лежит в основе идеи параллельного выполнения операций обработки и ввода-вывода.

**Попробуйте сделать вывод о том, какие существуют способы повышения производительности информационной системы, как учесть разницу в быстродействии ПУ и ЦП.**

Повысить производительность в таких случаях возможна за счёт организации || выполнения операции обработки и ввода-вывода. При введении средств автономного управления ПУ, непосредственное участие центральных устройств в обмене инфо в ЦУ может ограничиваться тактом передачи инфо ПУ, который значительно короче такта подготовки инфо. Кроме того, при операции обработки, лишь часть времени выполнение … в АЛУ уходит в обращение к АЗУ, поэтому введение спец средств управления и доступа к памяти со стороны ПУ позволяет существенно повысить производительность. Однако, при этом д.б. реализован другой асинхронный принцип управления, обеспечивающий независимость работы ПУ, АЗУ и АЛУ.

**Какие способы согласования скорости работы ЦП и пропускной способности канала между ПУ и ЦП (канала ввода-вывода - КВВ) существуют.**

**Назовите возможные решения для увеличения коэффициента перекрытия.**

Для увеличение Кп необходимо :

1. Управление при подготовки квантов информации должно осуществляться автономно , независимо от ЦП.
2. В ЭВМ д.б. предусмотрены средства и связи для передачи квантов информации между ПУ и АЗУ, минуя АЛУ, т.е. средство прямого доступа к памяти.
3. В ЭВМ д.б. предусмотрены средства для || выполнения процессов обработки в ЦП и подготовки кванта информации в ЦУ.
4. В течение всего процесса ввода-вывода инфо ЦП д.б. загружен операциями обработки, чтобы не возникли *простой* из-за нехватки исходных данных или команд.

**Какой недостаток имеет такой метод обмена данными?**

Обмен массивами имеет недостаток, заключающийся в необходимости дополнительного оборудования, обеспечивающего собственно обмен и управление им. Аппаратура передачи данных, представляющая собой совокупность каналов обмена и устройств сопряжения с ПУ, на время обмена выделяет каждому ПУ подканал. В нем осуществляется хранение управляющей информации (адреса размещения данных обмена, набора управляющих признаков, определяющих режимы обмена) об обмене с одним из ПУ, подлежащих обслуживанию. Подканалы могут быть разделенными и неразделенными. Разделенный подканал используется для связи с несколькими ПУ с разделением во времени (мультиплексный канал). Неразделенный подканал осуществляет обмен с одним определенным ПУ.

**Познакомьтесь с примером организации опережающего ввод (Spooling) в большой вычислительной машине IBM Mainframe, которая повышает производительность выполнения программы при чтении на вводе и записи на выводе.**

Рассмотрим пример организации опережающего ввода (Spooling) в большой вычислительной машине IBM Mainframe (рисунок . Операционная система управляет работой системы ввода-вывода следующим образом:

Используется память устройства прямого доступа DASD(*direct access storage devices*) как буферная память, чтобы снизить задержку при передаче данных между периферийным оборудованием и выполняемой программой.

Чтение и запись входных и выходных потоков осуществляется на промежуточных устройствах для их дальнейшей обработки или для вывода на печать.

Выполнение такой операции, как печать, в то время, когда компьютер выполняет другие виды работ.

Существует два вида буферизации (**spooling): ввода и вывода.**  Оба они повышают производительность выполнения программы при чтении на вводе и записи на выводе.

Для выполнения операции буферизации пишется специальная программа (скрипт) на языке **JCL** (job control language), в котором вы задаете команду определения буферного хранилища данных (Data Defenition): **// DD \***. Эта команда означает создание файла, содержащего все записи, заключенные между этой командой (// DD \* statement) и командой окончания файла /\* . Созданный файл хранится в специальной области, называемой «**JES spool area**» - это большой файл на специально выделенном для целей буферизации диске.

После того, как программа будет выполнена и потребуется чтение данных, подсистема **JES** (Job Entry Subsystem) “вытащит” записи из памяти и предоставит их программе (со скоростью работы диска).

Чтобы выполнить операцию буферизации для вывода данных, также пишется специальный скрипт на языке **JCL** (job control language), в котором задается опция SYSOUT для команды DD. Опция SYSOUT определяет пустой файл в буфере, выделенный для логических записей длиной в 132 символа в том или ином формате печати (EBCDIC/ASCII/UNICODE).

Нужен рисунок!

ПУ начинает работу по команде запуска от ЦП, после чего работает автономно, подготавливая блок информации в течение времени . ЦП при этом продолжает выполнение текущей программы. Подготовив информацию к передаче, ПУ посылает сигнал запроса в ЦП и ЦП приостанавливает выполнение текущей программы для операции обмена информацией. После окончания передачи информации (времени ) ПУ посылает в ЦП сигнал, подтверждающий окончание операции. Таким образом, из диаграммы видно, что параллельная работа ПУ и ЦП шла в тактах подготовки информации. Следовательно, на показатель эффективности будет влиять уже только параметр . Таким образом, цикл работы ПУ,где - коэффициент перекрытия или совпадения во времени операций обработки и ввода-вывода к цикле работы ПУ. При отсутствии перекрытия , при полном перекрытии .

Для увеличения возможны следующие действия:

управление ПУ при подготовке информации должно осуществляться автономно, независимо от ЦП;

должны быть предусмотрены средства связи для прямой передачи между ПУ и памятью, минуя ЦП, так называемые средства ***прямого доступа к памяти;***

должны быть средства синхронизации параллельного выполнения процессов обработки в ЦП и подготовки информации в ПУ;

в течение всего процесса ввода-вывода ЦП должен быть загружен операциями обработки, чтобы не была возможна такая причина простоев, как нехватка исходных данных или команд, то есть, должна быть организована опережающая подготовка данных (spooling).Обработка в ЦП не должна задерживаться из-за отсутствия в основной памяти информации, подлежащей обработке, или занятости необходимых областей результатами предыдущей обработки.При однопрограммном режиме работы, то есть, когда программы в ЦП выполняются строго последовательно и переход к следующей осуществляется только после завершения предыдущей, вероятность простоя ЦП из-за отсутствия необходимых для продолжения вычисления данных может быть значительно уменьшена путем организации ***опережающего ввода***.При опережающем вводе вся информация, подлежащая вводу, разбивается на несколько порций. Вначале производится ввод первой порции, содержащей программу и данные, необходимые для начала операции обработки. Вторую порцию информации вводят при параллельно работающем ЦП. Если ввод второй порции информации завершается до окончания обработки первой порции, то ЦП простаиватъ не будет. Для такой организации вычислительного процесса необходимо, чтобы программа предвосхитила потребность в данных и начала их ввод заранее. При однопрограммном режиме работы ввод данных может быть организован параллельно с обработкой, начиная с момента их получения.Значительно большие возможности при загрузке ЦП предоставляет ***мультипрограммный режим***, который предполагает наличие в ОЗУ нескольких независимых программ, принятых на обслуживание. Программы или участки одной программы можно считать независимыми, если каждая из них может быть выполнена без использования результатов обработки других программ. Все программы (или их запросы) находятся в очередях к соответствующим устройствам (УВв, УВыв, ЦП, ОЗУ) и переходят по мере выполнения соответствующих операций из одной очереди в другую. Большинство функций по согласованию скоростей, обеспечению синхронизации автономно работающих устройств, формированию адресов, преобразованию форматов и так далее выполняется каналом ввода- вывода (КВВ), который в последнее время чаще называют интерфейсом.

При обмене между центральным и периферийными устройствами данные объединяют в массивы, называемые также блоками, кадрами или пакетами. Обмен инициируется командой ЭВМ, а затем протекает автономно, без участия ЦП, причем обмен ведется с заранее выделенной областью памяти. На рисунке показана схема механизма передачи *N* слов единым массивом. В памяти выделена область хранения *N* слов; при этом предполагается, что слова по одному заносятся в память или считываются из нее. Имеются три адреса, которые нужно помнить: начальный, текущий и конечный Они хранятся в трех специальных регистрах, называемых регистрами начального адреса слов РгНАСл, конечного адреса слов РгКАСл и текущего адреса слов РгТАСл. Вначале текущий адрес слова совпадает с начальным адресом. Затем, по мере того как очередное слово заносится в память (или считывается из нее), текущий адрес слова увеличивается на единицу, так что всякое очередное слово будет заноситься в следующий адрес (или же считываться из него). Перед каждым изменением текущего адреса оно сравнивается с конечным адресом слова. Если текущий адрес совпадает с конечным, то обмен массивами прекращается и вырабатывается сигнал прерывания, по которому ЭВМ переходит к исполнению соответствующей программы. Для выработки текущих адресов слов РгТАСл должен выполнять функции счетчика.

**Какие типы каналов используются при этом?**

Подканалы могут быть разделенными и неразделенными. Разделенный подканал используется для связи с несколькими ПУ с разделением во времени (мультиплексный канал). Неразделенный подканал осуществляет обмен с одним определенным ПУ.

Каналы, образуемые совокупностью подканалов, осуществляют прием и расшифровку запросов на обмен, хранение запросов на обмен до их завершения, введение управляющих слов в подканалах, логическое подключение ПУ к конкретным подканалам. Каналы обмена включают наборы линий связи, по которым осуществляется передача информации и сигналов управления.

Каналы ввода-вывода позволяют вычислительной системе выполнять одновременно ввод, вывод и обработку информации, освобождая ЦП от непосредственного управления вводом-выводом. Конструктивно каналы могут быть либо независимыми устройствами (Switch), имеющими в своем составе все необходимые логические устройства и память, либо частью процессора и использовать для выполнения своих функций оборудование ЦП. В обоих случаях **функции канала** заключаются в следующем:

в выборке заданного ПУ,

управлении передачей информации между ПУ и ОЗУ,

распределении данных при обмене,

промежуточном хранении информации ввода-вывода,

преобразовании форматов вводимых и выводимых данных,

контроле правильности выполнения ввода-вывода

Поскольку обмен между ОЗУ и ПУ осуществляется массивами - страницами или зонами, то емкость БЗУ выбирается равной информационной емкости массива. Выигрыш во времени при такой организации обмена:

,где - время передачи массива из ОЗУ в БЗУ, - время передачи массива из БЗУ в ПУ.

При наличии лишь одного блока (секции) ОЗУ это время для выполнения вычислений процессором не используется. Если же использовать несколько секций ОЗУ и соответствующим образом организовать потоки информации, то передачу массивов можно проводить между БЗУ и теми секциями ОЗУ, которые не. участвуют в вычислениях в данный промежуток времени (рисунок 4). Обмен массивами освобождает процессор от операции обмена, оставляя за ним только ввод заказов на обмен и передачу управления соответствующей программе после окончания обмена.

**Почему использование нескольких секций ОЗУ повышает производительность обмена?**

Улучшение производительности производится за счёт увеличения секций ОЗУ, т.к. при этом производится повышение коэффициента размерности .

Отличительная черта любой вычислительной системы - это наличие канала ввода-вывода (КВВ), то есть, аппаратуры согласования работы ЦП и ПУ. КВВ – это функциональный элемент, служащий для организации связей и управления обменом между ПУ и внутренней памятью. КВВ представляет собой совокупность аппаратных и программных средств, предназначенных для организации, управления обменом и непосредственной передачи данных между ОЗУ и ПУ. Он образует маршрут (или физический канал) передачи данных между ОЗУ и ПУ и осуществляет управление обменом. Физическая реализация подключения ПУ к КВВ выполняется средствами ***интерфейса***.

Основные функции КВВ можно разделить на 3 группы:

установление логической связи между ПУ и ОЗУ (***образование канала***);

непосредственная ***передача данных*** между ПУ и ОЗУ;

завершение обмена (***разрушение канала).***

По характеру обслуживания различают:

Мультиплексный КВВ (МКВВ) и

Селекторный КВВ (СКВВ)

**Какое количество ПУ может обслуживать один Мультиплексный КВВ?**

Сам канал быстродействующий, но обслуживает медленное периферийное устройство. При этом, подключившись к одному устройству, подаёт одно машинное слово, и после этого подключается к другому. Мультиплексный канал предназначен для одновременного обслуживания нескольких ПУ, каждое из которых работает в соответствии с отдельной канальной программой. В мультиплексном канале все оборудование последовательно используется несколькими внешними устройствами. К каналам такого типа возможно подключение до 256 внешних устройств, причем все они могут работать одновременно, пока суммарный поток данных находится в приемлемых границах. Максимальное количество внешних устройств, способных одновременно работать с данным каналом, определяется пропускной способностью канала. В свою очередь пропускная способность канала зависит от временных характеристик как внешних устройств, так и элементов, на которых собрано оборудование канала. Пропускная способность канала оценивается количеством символов, проходящих через оборудование канала в единицу времени.

*ПРАВИЛЬНЫЙ ОТВЕТ:* МКВВ позволяет одновременно обслуживать несколько параллельно работающих ПУ [3]. Каждое ПУ подключается к каналу на короткие промежутки времени, называемые сеансами связи. Для организации такого режима работы в ЭВС используется система прерываний. Сеансы связи с различными ПУ чередуются между собой в зависимости от сигнала готовности «Готов» со стороны ПУ. МКВВ предназначен для работы с медленнодействующими ПУ - принтерами, плоттерами, оперативными УВВ.

**Какое количество ПУ может обслуживать один Селекторный КВВ?**

Канал быстродействующий и обслуживает быстрые устройства. При этом подключившись к одному устройству, передаёт всю информацию, и после этого подключается к другому устройству.

*ПРАВИЛЬНЫЙ ОТВЕТ:* С помощью селекторного канала ввода-вывода (СКВВ) к центральным устройствам подключаются быстродействующие периферийные устройства, способные ожидать в очереди начала обмена [3]. В такой вычислительной системе информация обрабатывается в пакетном режиме, один пакет программы последовательно за другим.

**Перечислите основные узлы МКВВ**

В состав МКВВ входят следующие основные узлы (рисунок 1.5.1.).

1. Набор регистров, где во время сеанса связи с ПУ находятся текущие параметры активного подканала (для которого проводится данный сеанс связи):

РгКОУ содержит код и указатели операции ввода-вывода;

РгТАд содержит текущий адрес данных;

СчТд указывает количество байтов, которое осталось ввести в ОЗУ или вывести из него в данной операции;

РгАУСл определяет адрес очередного управляющего слова в цепи управляющих слов.

2. Память подканалов, представляющая собой внутреннюю память канала, предназначенную для хранения текущих параметров операций ввода-вывода, относящихся к пассивным подканалам. Каждому подканалу отведен в памяти участок, в котором хранится содержимое всех четырех перечисленных регистров РгКОУ, РгТАД, СчТд, РгАУСл, соответствующее последнему сеансу связи для данного подканала. Кроме того, в памяти подканалов предусматривается место для хранения других параметров операции, о которых будет сказано позже. Подканалы в памяти располагаются упорядоченно по возрастанию номера подканала, который таким образом может использоваться для адресации ячеек памяти подканалов. При формировании адресов ячеек памяти подканалов используется содержимое регистра номера активного подканала РгНАП.

З. Регистр связи с интерфейсом РгСвИ, куда поступает информация, получаемая из ПУ при вводе, и. откуда в ПУ выдается выводимая информация.

4. Обращение к ОЗУ осуществляется через блок связи с ОЗУ (БСОЗУ). Этот блок в соответствии с алгоритмом выполняемой операции выбирает один из регистров канала в качестве источника адреса, другой - в качестве источника или приемника информации и вырабатывает сигналы, необходимые для обмена информацией с ОЗУ.

5. Обмен информацией канала с ОЗУ производится словами, обмен с ПУ производится обычно более мелкими единицами информации, например, байтами. Поэтому при вводе канал формирует слова из поступающих в канал байтов, а при выводе - разворачивает слова в последовательность байтов. Для определения конца преобразования слов используется счетчик байтов СчБ, указывающий номер последнего обработанного байта в текущем слове данных. Содержимое СчБ и и слово данных, участвующее в обмене, запоминаются в соответствующем участке памяти подканалов.

6. Регистр команд ввода-вывода РгКВВ хранит код команды, поступающей в канал из процессора, когда процессор в соответствии с программой, называемой планировщиком, запускает новую операцию ввода-вывода.

**Перечислите основные узлы СКВВ**

СК содержит набор регистров, большинство из которых по своим функциям аналогичны соответствующим регистрам вышеописанного мультиплексного канала (МК). Это регистры: РгКВВ, РгКОУ, РгТАД, СчТД, РгАУСл, РгСвИ, СчБ. Регистр номера ПУ (РгНПУ), заполняемый процессором при начальной выборке, указывает, с каким из устройств селекторного канала (СК) проводится текущая операция. Регистр данных Ргд и регистр предварительного управляющего слова РгПУСл служат для уменьшения задержек в работе селекторного канала при обращениях к ОЗУ.

Слайд подчеркивает важное понятие: **КВВ** -– это функциональный элемент, служащий для организации связей и управления обменом между ПУ и внутренней памятью. На примере архитектуры мейнфрейма IBM показать способы организации КВВ и перечислить основные функции канала.

**Перечислите функции канала ввода-вывода**

КВВ и интерфейсы выполняют следующие функции:

1)Позволяют иметь машины с переменным составом периферийных устройств.  
2) Обеспечивают параллельную работу периферийных устройств как между собой, так и по отношению к процессору.  
3) Обеспечивают автоматическое распознавание и реакцию процессора на различные ситуации, возникающие в периферийных устройствах.

*ПРАВИЛЬНЫЙ ОТВЕТ:* Основные функции КВВ можно разделить на 3 группы:

установление логической связи между ПУ и ОЗУ (***образование канала***);

непосредственная ***передача данных*** между ПУ и ОЗУ;

завершение обмена (***разрушение канала).***

**Дайте определение техническим характеристикам КВВ**

***Номинальная пропускная способность*** КВВ – число бит, которые могут быть переданы через КВВ в ОЗУ в единицу времени при условии, что никакие другие устройства не мешают выполнению функций КВВ.

***Нагрузочная способность***  - наибольшее число ПУ, которое может обслуживать КВВ, не вызывая потерь информации и снижения скорости их работы.

N-число ПУ K-коэффициент размерности Vkвв-номинальная пропускная способность КВВ Тобсл-время обслуживания одного запроса. Сбзу-ёмкость буферного запоминающего устройства. n-глубина прерывания. (Программа прерывает другую).

Wmax-max пропускная способность канала связи N-нагрузочная способность КВВ., Где Wmax=F\*Ln(1+Pc/Pп) – Шеннона-Тайлера, и F-ширина полосы пропускания Pc-ср мощность полезного сигнала Pп-ср мощность сигнала помех

***Протокол*** – совокупность правил, устанавливающих единый порядок обмена информацией (данными) и управляющими сигналами.

**Сравните три способа организации связи между ЦП и ПУ (Типы и протоколы обмена данными между ЦП и ПУ)**

**Укажите плюсы и минусы каждого способа**

При программном способе организации связи между ЦП и ПУ инициативу по началу обмена информацией может проявить только ЦП, ПУ не может инициировать обмен. Реализация способа – только программная.

Во втором способе – инициатива может исходить от ПУ. Запрос на прерывание (ЗП) выставляется на специальном входе ЦП, этот вход ЦП опрашивает после выполнения каждой команды. Если на этом входе появился сигнал, ЦП начинает отработку программы прерывания. Реализация способа – возможна и программная (простота) и аппаратная (быстродействие).

Третий способ вообще исключает «из игры» ЦП. Обмен информацией идет напрямую между ПУ и памятью Реализация способа – только аппаратная.

**Дайте определение приостановке и прерыванию**

**Как эти процедуры влияют на время цикла памяти**

*1) Приостановка* - процедура, при которой средства управления, работающие автономно от ЦП, задерживают его работу на время *цикла памяти* , при котором ОЗУ непосредственно занято приемом или выдачей информации. Механизм приостановок - процесс, позволяющий ПУ передавать или получать информацию из оперативной памяти (ОЗУ) без длительной операции обработки прерываний. Приостановка - занятие цикла ОЗУ, возникает в том случае, если к ОЗУ одновременно или попеременно обращаются ЦП и ПУ. Так как ОЗУ занято постоянно, запрос ЦП не удовлетворяется, следовательно, работа ЦП приостанавливается на то время, которое необходимо для освобождения ОЗУ от обрабатываемого процесса. Во время приостановок текущее состояние процессора не меняется, но выполнение программы (команды) задерживается до освобождения ОЗУ. Процесс приостановки работы ЦП показан на рисунке 1.6.1.

 Процесс приостановки работы центрального процессора на рис

Если обращение ЦП к ОЗУ произошло в момент , но ОЗУ занято выполнением чтения- записи данных от другого источника, то работа ЦП приостанавливается на время до момента освобождения ОЗУ. Во время приостановки ЦП никаких действий не выполняет. Приостановки, называемые также «*занятие цикла памяти*», широко используются при организации любых вычислительных систем. Они обеспечивают высокую степень совмещения операций обработки и ввода-вывода, которая тем выше, чем меньше длительность цикла памяти относительно длительности выполнения команды процессора .

Задержки, вызванные приостановками, обычно короче, чем задержки, вызванные прерываниями. Однако возможности приостановок ограничены непосредственной передачей данных при обращении к ОЗУ, поэтому для выполнения каких- либо действий по управлению обменом информацией со стороны ЦП необходимы прерывания.

Максимальное время задержки равно полному циклу обновления ОЗУ [7], а среднее время задержки - половине этого цикла. Эти задержки снижают *производительность ЦП*, однако, снижение значительно меньше, чем при прерываниях. При этом используется *буферизация* - для сокращения числа прерываний и приостановок используются специально выделенные буферные регистры, в которых хранятся принимаемые из ПУ байты информации до передачи в ОЗУ, при этом они объединяются в машинные слова - это уменьшает потери информации за счет увеличения допустимого времени ожидания при обращении к ОЗУ.

*2) Прерывание* - процесс переключения ЦП с одной программы на другую по внешнему сигналу (например, от ПУ) с сохранением информации для последующего возобновления работы прерванной программы. ПУ при возникновении события, требующего реакции со стороны ЦП, формирует сигнал, называемый запросом прерывания. *Запрос прерывания* (ЗП) может поступить в ЦП в произвольные моменты времени асинхронно по отношению к выполнению программы, поэтому ЗП запоминаются в специальном регистре, называемом регистром ЗП (РгЗП). В разрядах РгЗП в любой момент времени могут быть установлены «флаги» от внешних схем ПУ. Процессор производит опрос состояний разрядов регистра РгЗП в определенные моменты выполнения программы, при наличии хотя бы одной единицы ЦП передаёт управление в заранее определённую ячейку памяти для выполнения обработки прерывания. *Обработка прерывания* - запоминание содержимого регистров ЦП в специально выделенной области памяти и загрузка этих регистров новой информацией для выполнения новой программы. Интервал времени между моментом поступления сигнала ЗП в РгЗП и моментом начала обработки прерывания называют временем ожидания или *реакцией на прерывание* . *Обработка прерывания* включает в себя (см. рис. 1.6.) этапы запоминания состояния прерываемой программы и перехода к выполнению прерывающей программы ; собственно выполнения прерывающей программы и восстановления состояния прерванной программы и возврата к ее выполнению .

**Дайте определение понятия глубины прерывания.**

Глубина прерывания — максимальное число про­грамм, которые могут прерывать друг друга.

На рисунке приведена временная диаграмма процесса прерывания с глубинами прерывания, равными 1 и 3.

**Посмотрите на диаграммы работы процессора: Сколько программ выполняется? Какова здесь глубина прерывания?**

На рисунке приведена временная диаграмма процесса прерывания с глубинами прерывания, равными 1 и 3.

**Что делает арбитр?**

Схемы, позволяющие определить номер уровня со старшим приоритетом, называются схемами арбитража (арбитрами) .

*ДОПОЛНЕНИЕ 1:* Схема арбитра здесь имеет жесткую логическую структуру. Каналы с высокими приоритетами оттесняют низкоприоритетные, в результате чего они могут простаивать длительное время и даже полностью лишиться права доступа к магистрали. Для повышения гибкости работы арбитра необходимо периодически перестраивать приоритетную структуру, что достигается применением программного управления приоритетами.

*ДОПОЛНЕНИЕ: 2* На рис. приведена схема подключения арбитра к магистралям интерфейса типа «Общая шина» (ОШ) - Малтибас (MULTIBUS) - для однокристальных микропроцессорных наборов типа К580, К1810. Активные устройства, от которых может поступить ЗП, расположены в виде матрицы и разбиты на восемь групп в соответствии с восемью линиями связи интерфейса, по которым передается однобайтовое слово. Приоритет отдается устройствам верхней группы BRi - управляющие линии для передачи i =0…7 сигналов ЗП в арбитр. Линии, обозначенные на рисунке как BGi - управляющие линии для передачи сигналов разрешений от арбитра к активным устройствам. Линия, обозначенная на рисунке как SАСК – это оповещающая линия передачи ответа о подключении к магистрали того первоочередного претендента из активных устройств, у которого самый высокий приоритет.По получению сигнала SАСК арбитр прекращает выдачу сигнала разрешения. ВВSY - линия передачи сигнала, оповещающего все устройства о занятости (захвате) магистрали. По установлению режима захвата система переходит к прерыванию, затем к записи либо чтению. После ее освобождения очередник занимает линию, и сам формирует сигнал ВВSY. Линия R - вход сброса арбитра. Все ЗП складываются по схеме “монтажное ИЛИ” на линиях запроса BRi .Разрешение выдается арбитром группе с наименьшим номером. В группе разрешение получает устройство, которое ближе к арбитру. Пока идет обмен, арбитр готовит нового следующего очередника.

**Какие преимущества предоставляет программное управление приоритетами?**

Запросы от всех источников прерывания поступают на регистр запросов прерывания, устанавливая соответствующие его разряды в состояние 1, указывающее на факт наличия запроса прерывания определенного источника (причины).

Обычно в многоуровневых системах прерывания распознавание активного уровня прерывания производится аппаратными средствами, а конкретная причина прерывания определяется прерывающей программой по коду прерывания. Многоуровневое построение позволяет уменьшить объем аппаратуры, но приводит к замедлению работы системы прерывания. Одноуровневая система, в которой понятия запроса (причины) и уровня прерывания объединяются, позволяет достичь большего быстродействия в работе системы прерывания.

Операционная система каждую миллисекунду меняет по определенному алгоритму приоритеты между восемью каналами связи с ПУ.

Количество причин прерывания может достигать от 16 в ПК до нескольких десятков и сотен в вычислительных системах и супер-ЭВМ. В таких случаях систему прерываний выполняют многоуровневой .

**В каких случаях систему прерываний выполняют многоуровневой?**

Запросы от всех источников прерывания поступают на регистр запросов прерывания, устанавливая соответствующие его разряды в состояние 1, указывающее на факт наличия запроса прерывания определенного источника (причины).

Обычно в многоуровневых системах прерывания распознавание активного уровня прерывания производится аппаратными средствами, а конкретная причина прерывания определяется прерывающей программой по коду прерывания. Многоуровневое построение позволяет уменьшить объем аппаратуры, но приводит к замедлению работы системы прерывания. Одноуровневая система, в которой понятия запроса (причины) и уровня прерывания объединяются, позволяет достичь большего быстродействия в работе системы прерывания.

Многоуровневая система прерывания характеризуется тем,что в системе может быть несколько КПр, которые могут одновременно послать запросы от ВУ на обслуживание прерывания.Очевидно, в МП должен быть предусмотрен механизм для приема в каждый момент времени единственного запроса для обслуживания прерывания. Этот механизм реализуется, как правило, аппаратными средствами и позволяет в первую очередь выбирать порядок обслуживания ВУ: либо первым удовлетворяется запрос прерывания от того контроллера, который включен ближе к МП в цепочке передачи сигналов «запрос прерывания - разрешение прерывания», либо первым удовлетворяется запрос прерывания с КПр, программно имеющего наивысший приоритет. Разделение всех ВУ в системе по уровням приоритетности обуславливается важностью формируемой ими информации. ВУ, в которых накопленная информация требует незамедлительной ее обработки, имеют наивысший или более высокий приоритет по отношению к МП. Те ВУ, в которых информация может храниться некоторое время без обработки, имеют средний или более низкий приоритет. Максимальное число программ, обслуживающих ВУ, которые могут прервать друг друга, характеризуют «уровень прерывания». При этом порядок обработки программ при одновременном их запросе определяется уровнем приоритета обслуживаемых ВУ.

В многоуровневых системах прерывания задание уровня

приоритета осуществляется тремя способами:

а) использованием в самом МП (или в контроллере прерываний) нескольких входов запросов на прерывание, каждый из которых имеет свой уровень приоритета;

б) с помощью 2 - 4-разрядной шины для подачи кода, несущего информацию об уровне приоритета того или иного КПр, запросившего прерывание;

в) использованием внешних аппаратных средств.

В многоуровневых системах прерываний программа обслуживания ВУ низкого уровня может быть прервана запросом на прерывание от Кпр, обслуживающего ВУ высокого уровня. Если же запрос на прерывание от КПр имеет тот ж уровень, что и обслуживаемое на момент запроса прерывание, или более низкий, то МП не реагирует на этот запрос до тех пор, пока не закончит обслуживание устройства.

**Что содержится в “коде прерывания”?**

Операционная система каждую миллисекунду меняет по определенному алгоритму приоритеты между восемью каналами связи с ПУ в зависимости от кода Q1 – Q7 на управляющих входах арбитров А (см. таблицу истинности схемы арбитра).

Количество причин прерывания может достигать от 16 в ПК до нескольких десятков и сотен в вычислительных системах и суперЭВМ. В таких случаях систему прерываний выполняют многоуровневой.

Запросы (I) от всех источников прерывания поступают на регистр запросов прерывания (РгЗП), устанавливая соответствующие его разряды в состояние 1, указывающее на факт наличия запроса прерывания определенного источника (причины). Запросы 1-го k-го уровней прерывания (ЗПУ) формируются элементами ИЛИ, объединяющими разряды РгЗП, относящиеся к соответствующим уровням. Еще одна схема ИЛИ формирует общий сигнал прерывания (ОСП), поступающий в устройство управления процессора. Значение сигнала ОСП определяется выражением:



Информация о действительной причине прерывания, породившей запрос данного уровня ЗПУ, содержится в “коде прерывания”, который отражает состояние разрядов РгЗП, относящихся к данному уровню прерывания. Обычно в многоуровневых системах прерывания распознавание активного уровня прерывания производится аппаратными средствами, а конкретная причина прерывания определяется прерывающей программой по коду прерывания. После принятия запроса прерывания на исполнение и передачи управления прерывающей программе соответствующий триггер РгЗП сбрасывается.

**Плюсы и минусы многоуровневого построения?**

Для оценки эффективности системы прерывания используются следующие характеристики системы

прерывания.Время реакции - время между появлением запроса прерывания и началом выполнения прерывающей программы. Для одного и того же запроса прерывания задержки в исполнении прерывающей программы зависят от того, сколько программ с более высоким приоритетом ВУ ждут обслуживания. По-

этому время реакции определяется для запроса с наивысшим приоритетом ВУ. Время реакции зависит от того, в какой момент допустимо прерывание. Большей частью прерывание допускается после окончания текущей команды. В этом случае время реакции определяется в основном длительностью выполнения команды. Оно

может оказаться недопустимо большим для систем, работающих в реальном масштабе времени. В таких системах часто допускается прерывание обрабатываемой программы после любого такта вы-

полнения команды. Однако при этом возрастает количество информации, подлежащей запоминанию и восстановлению при переключении программ, так как в этом случае необходимо сохранять также и состояния в момент прерывания счетчика тактов, регистра кода операции и некоторых других. Поэтому такая организация прерывания возможна в МПС с быстродействующей сверхоперативной памятью. Имеются ситуации, в которых желательно немедленное прерывание. Если аппаратура контроля обнаружила ошибку, то целесообразно сразу же прервать операцию, пока ошибка не оказала влияния на следующие такты работы системы. Затраты времени на переключение программ равны суммарному расходу времени на запоминание и восстановление состояния программы:

Глубина прерывания - максимальное число программ, которые могут прерывать друг друга. Глубина рерывания обычно совпадает с числом уровней приоритета в системе прерываний. Необходимо отметить, что, чем большее значение глубины прерывания имеет система, тем более быстрой реакцией на срочные запросы она обладает. Однако, если запрос окажется необслуженным к моменту прихода нового запроса от того же источника, то возникает так называемое насыщение системы прерывания. В этом случае предыдущий запрос прерывания от данного источника будет системой утрачен, что является недопустимым. Для устранения этого явления быстродействие МПС, характеристики системы прерывания, число источников прерывания и частота возникновения запросов должны быть согласованы таким образом, чтобы насыщение было невозможным.

**Для чего предназначен режим прямого доступа к памяти? Какие ПУ при этом обслуживаются?**

**Какую (Она чукча???) ввода и вывода информации обеспечивает DMA?**

Режим прямого доступа к памяти предназначен для передачи блоков данных между оперативной памятью и периферийным устройством (внешней платой) без участия ЦП. Этот протокол используется обычно для обслуживания быстродействующих ПУ. Обмен данными через канал DMA обеспечивает максимальную скорость ввода и вывода информации.

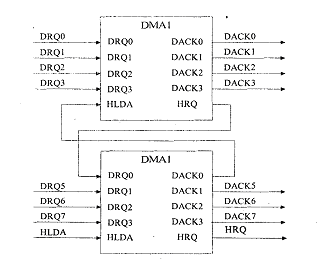
В этом режиме ПУ управляет работой общей шины, то есть, является задатчиком, а оперативная память - исполнителем. В компьютерах РС/АТ для реализации режима DMA используются 2 микросхемы 18237 (DMA), соединенные каскадно (рисунок 1.6.9.) с целью увеличения числа каналов.

Рисунок 1.6.9. Каскадное соединение DMA1 и DMA2

Работа канала прямого доступа DMA основывается на непосредственном обмене данных между памятью и ПУ без участия ЦП. Такой обмен осуществляется при откладывании выполнения основной программы на время обращения ПУ к памяти. Поскольку память отключена от ЦП и подключена к ПУ только на время записи или чтения, то можно говорить, что обмен происходит в режиме захвата цикла памяти.

**Перечислите последовательность основных действий устройств в режиме DMA при чтении байта данных из ОП.**

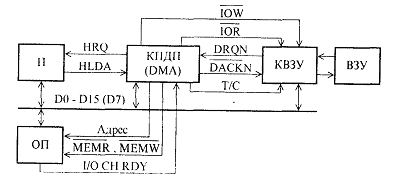


Рисунок 1.6.10 – Пример сопряжения устройств по каналу DMA

Последовательность основных действий перечисленных устройств в режиме DMA при чтении байта данных из ОП (рисунок 1.6.10) следующая:

* КНГМД формирует сигнал DRQ2;
* КПДП вырабатывает сигнал HRQ,
* ЦП освобождает шину и устанавливает сигнал HLDA.
* КПДП, захватив шину, передает адрес в ОП.
* КПДП формирует сигнал DACK2, поступающий в КНГМД;
* КПДП вырабатывает сигнал MEMR, осуществляющий чтение выбранной ячейки ОП.
* Содержимое ячейки ОП запоминается в регистре данных КНГМД по сигналу IOW, установленному КПДП.
* КПДП снимает сигнал HRQ (режим одиночной передачи).
* ЦП сбрасывает сигнал HLDA.
* КПДП увеличивает на 1 значение адреса и уменьшает на 1 число передаваемых байтов данных с появлением сигнала I/О СН RDY из ОП.

Если число передаваемых байтов не равно FFFFh, то последовательность действий повторяется до тех пор, пока не будет сформирован сигнал Т/С. Сигнал Т/С используется для формирования сигнала прерывания IRQ6 (при наличии сигнала разрешения прерывания), сообщающего ЦП о завершении передачи блока данных.

**Как решается проблема обслуживания нескольких запросов DMA, поступивших от нескольких различных ПУ? Познакомьтесь с назначением каждого канала DMA.**

Запрос DMA по сигналу DRQN устанавливается ПУ, и в этом мы видим прямую аналогию с запросами на прерывание. Проблема обслуживания нескольких запросов DMA, поступивших от нескольких различных ПУ, решается с помощью выставления приоритетов. Рассмотрим, как это организовано на примере реализации DMA в компьютерах серии PC/AT (рисунок 1.6.9).

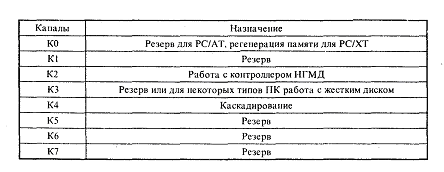
Каждая микросхема DMA содержит 4 канала (К0-К3), работающие в режиме разделения времени (в какой—либо момент времени только один канал является активным и обслуживает подключенное к нему периферийное устройство). Наивысшим приоритетом обладает канал 0 (К0). С увеличением номера канала приоритет уменьшается. Каналу соответствует сигнал DRQN (N=0-3), называемый запросом DMA. Периферийное устройство, требующее режима DMA, устанавливает этот сигнал. Именно здесь усматривается некоторая аналогия с сигналами запросов прерывания. Если одновременно поступает несколько сигналов запросов DMA, то внутренняя схема арбитража в DМА выбирает наиболее приоритетный сигнал DRQN. После этого DМА устанавливает общий для всех каналов сигнал HRQ (запрос шины) и после поступления от процессора сигнала HLDA подтверждения на захват шины, DМА становится «Задатчиком» и формирует для N-го периферийного устройства сигнал DАСК N (подтверждение DMA), разрешающий циклы прямого доступа к памяти. Для двух каскадно-соединенных DМА наивысший приоритет имеет сигнал DRQ0, а сигнал DRQ7 - наименьший. Кроме того, четвертый канал в DMA1 используется только для реализации каскадного соединения (запрограммирован на выполнение режима каскадирования). Назначение каналов DMA в компьютерах серии PС/АТ представлено в таблице 1.6.3.

Таблица 1.6.3. Назначение каналов DMA

Каналы К0-К3 используются для байтовых передач данных, а каналы К5-К7 --для передачи 2-х байтовых слов. Внешние платы (Задатчики) употребляют резервные каналы для получения доступа к системной шине. Для выполнения этой цели внешняя плата формирует сигнал DACK N и устанавливает сигнал МАSTER, блокирующий выход контроллера DMA на системную шину.

**Как организована адресация DMA?**

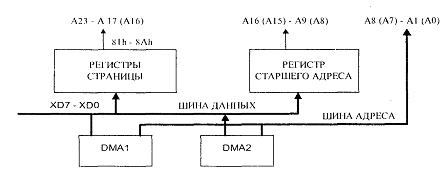
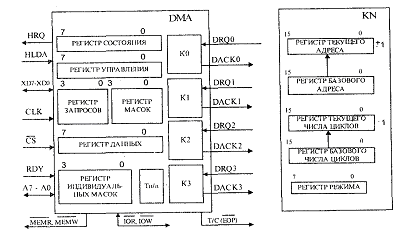
Рассмотрим схему формирования адреса (рисунок 1.6.11.) при прямом доступе к памяти.

Рисунок 1.6.11. – Схема формирования адреса при DMA

Схема формирования адреса содержит внешние по отношению к DМА регистры страницы и регистр старшего адреса. Регистры страницы - это байтовые порты с адресами из диапазона 81h-8Аh, загружаемые заранее процессором по шине данных ХD7 — ХD0 на этапе инициализации контроллера DMA.

Необходимость использования внешнего регистра старшего адреса связана с тем, что старшая часть адреса выводится из DМА по шине данных и запоминается в этом регистре по сигналам DМА. Младшая часть адреса выводится непосредственно по шине адреса DМА. На рисунке 1.6.11 показано, что в случае использования одного из каналов К5 —К7, связанных с 16-битными передачами, регистр страницы хранит адресные биты А23 —А17, регистр старшего адреса -- адресные биты А1б - А9, а по адресной шине выводятся адресные биты А8-А1. Адресный бит А0 в этом случае всегда равен 0, так как слова имеют четный адрес. Байтовые передачи данных связаны с обычным распределением битов адреса, которые указаны в скобках.

Внутренняя шина ХD7—ХD0 подключена к системной шине SD7-SD0 через соответствующий приемопередатчик, это подключение на рисунке 1.6.11 не показано.

**Перечислите основные части схемы DMA?**

Состав DMA и схема портов DМА изображена на рисунке 1.6.12.

Рисунок 1.6.12.- Схема портов DMA

В DMA применяется двунаправленная 8-битовая шина данных, по которой происходит запись выводимых из процессора байтов данных в 16-битовые (2-х байтовые) регистры выбранного канала. Для управления занесением байта данных в младшую часть (А7-А0) или старшую часть (А15-А8), например, регистра текущего адреса используется триггер первый/последний(Тп/п), который при нулевом значении разрешает запись байта данных в младшую часть регистра, а при единичном значении разрешает запись байта данных в старшую часть регистра. При инициализации DMA в начале программы следует один раз сбросить триггер (Тп/п), который затем автоматически устанавливается (сбрасывается) при передаче каждого байта.

В состав DMA входят следующие регистры:

* Регистр состояния доступен для чтения со стороны ЦП (8-ми битовый). Может быть выполнен как программно, так и аппаратно.
* Регистр управления (8-ми битовый) координирует работу DMA и доступен для записи со стороны ЦП.
* Регистр запросов (4-х битовый)
* Регистр масок(4-х битовый), к котором может быть установлен запрет DMA. Сброс установок осуществляется по сигналу RESET от порта 0Eh (DMA1) или DCh (DMA2).
* Регистр данныхDМА предназначен для временного хранения байта данных, передаваемого в режиме память-память. После завершения передачи байтов данных, последний байт может быть считан и введен в процессор.
* Регистр индивидуальных масок(4-х битовый), к котором может быть установлен индивидуальный запрет DMA для одного из каналов.

Кроме того, каждый из четырех каналов DМА содержит 8-ми битовый регистр режима и четыре 16-битовых регистра:

* ***Регистр текущего адреса*** после передачи каждого байта (слова) автоматически уменьшается или увеличивается на единицу. В циклах DMA биты А7-А0 выводятся по шине адреса DМА, а биты А8-А15 — по шине данных.
* ***Регистр текущего числа циклов*** (счетчик) задает количество передаваемых байтов (слов) и уменьшается на 1 после каждого цикла ПДП. Если содержимое этого регистра равно FFFFh, то это означает, что обмен данными завершен. Поэтому начальное значение, заносимое в регистр текущего числа циклов, должно быть уменьшено на 1.
* Для реализации режима автоматизации в каждом канале имеется ***регистр базового адреса*** и
* ***регистр базового числа циклов***, которые загружаются начальными значениями соответствующих текущих регистров и затем используются для повтора передачи данных.

**Какой из режимов наиболее употребительный?**

Рассмотрим режимы обслуживания DМА.

* ***Одиночная передача*** – наиболее употребительный режим, состоит в том, что после передачи очередного байта контроллер DMA освобождает системную шину и начинает затем запрос шины для управления передачей следующего байта. И так до тех пор, пока содержимое регистра текущего числа циклов не станет равным FFFFh.
* ***Блочная передача*** (монопольная) выполняется непрерывно без освобождения системной шины до тех пор, пока содержимое регистра текущего числа циклов в KN не станет равным FFFFh (обмен завершен).
* ***Передача по требованию*** осуществляется так же, как и блочная, за исключением того, что она выполняется, пока удерживается сигнал DRQN. Снятие сигнала DRQN останавливает управление передачей данных, которая возобновляется с той же точки при появлении сигнала DRQN.
* ***Псевдопередача*** (проверка)- заключается в том, что выполняется цикл записи (чтения), но без передачи данных.
* ***Режим каскадирования***, как уже отмечалось, используется для увеличения числа каналов КПДП путем соединения нескольких DМА.

**Какой из режимов выполняется без освобождения системной шины?**

Блочная передача (монопольная) выполняется непрерывно без освобождения системной шины до тех пор, пока содержимое регистра текущего числа циклов в KN не станет равным FFFFh (обмен завершен).

**Есть ли различия у драйверов для разных ОС?**

Да, драйвера для одного и того же устройства, написанные под DOS, под UNIX/Linux и под WINDOWS различны, но имеют общие свойства:

способ общения пользовательской программы с драйвером аналогичен общению с файлами;

способ общения драйвера с ОС строго стандартизирован;

драйвер – это легко заменяемая запасная часть операционной системы;

драйвер имеет возможность войти в режим ядра со своим программным кодом.

**Какая функция драйвера – Основная?**

Драйвер, как программный модуль, обеспечивающий согласованную работу ПУ и компьютера, реализует следующие функции :

Функция обмена с внешними устройствами (работа с таймерами для синхронизации процессов, управление аппаратурой прерываний, работа с файловой системой ввода-вывода)

Функция выделения памяти и буферизации

Функция управления задачами и подзадачами (запуск на выполнение, синхронизация обработки, прекращение и завершение).

Основная функция драйвера ПУ – это автоматическое распознавание и конфигурирование ПУ в момент подключения ПУ к ПК.

**В чем заключается принцип работы драйвера?**

Принцип работы драйвера заключается в том, что путем чтения из регистров или записи в регистры, он обеспечивает совместимость работы ПУ и ЦП и многих ПУ между собой.

**Назначение и функции каждого регистра**

Регистр данных

Регистр команд

Регистр состояния

Драйвер ПУ

Регистр состояния. Обычно считывается драйвером, когда тому необходимо получить информацию о текущем состоянии устройства.

Регистр управления (команд). Биты этого регистра управляют устройством некоторым разом, например, начиная или прекращая передачу данных. Драйвер обычно производит запись в такие регистры.

Регистры данных. Обычно такие регистры используются для передачи данных между устройством и драйвером.

**Изучите сравнительную таблицу драйверов для разных ОС.**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Драйвер** | **DOS Driver** | **UNIX Driver** | **Windows Legacy Driver** | **WDM (PnP)** |
| **Характеристика драйвера** |
| **Способ идентификации устройства** | Device ID в заголовке драйвера | Device number | Ntddk.h | Wdm.h |
| **Процедура инициализации** | Init | Init | DriverEntry | DriverEntry & AddDevice |
| **Процедуры обработки прерываний** | Interrupt Routine | Interrupt Service Routine | IRP пакеты | IRP пакеты |
| **IOCTL** | READ, WRITE, OPEN, CLOSE и т.д. | READ, WRITE, OPEN, CLOSE и т.д. | READ, WRITE, OPEN, CLOSE и т.д. | READ, WRITE, OPEN, CLOSE и т.д. |
| **Установка драйвера** | Запись в файл config.sys имени драйвера в формате \*.sys | Драйверы устройств в виде файлов хранятся в директории ‘/dev’ | С помощью файла \*.inf  Запись в Системном Реестре | С помощью файла \*.inf  Запись в Системном Реестре |

**Перечислите сходство и отличия драйвера и программы**

Драйвер – это «программный кусок», фрагмент кода операционной системы, общий для всех пользовательских программ, выполняющий функции ввода-вывода информации, позволяющий операционной системе общаться с аппаратурой.

Программа – это последовательность инструкций, предназначенная для исполнения устройством управления вычислительной машины

**Какой язык программирования может конкурировать с Ассемблером?**

Поскольку драйвер устройства – это сердце всех доступов к устройству, он должен быть запрограммирован так «плотно», как это возможно, чтобы сократить время выполнения и область памяти.

Формат драйвера строго определен и только язык Assembler дает вам требуемый контроль формата.

С помощью языка Assembler Вы можете манипулировать точно определенными регистрами центрального процессора в определенные моменты времени, что трудно сделать из языка С.

Вывод – ассемблер вне конкуренции.

**Какие имена присваивают символьным устройствам?**

Символьным устройствам присвоены имена, например CON, AUX, LPT, которые могут быть длиной до восьми символов (как имя файла). 8-ми символьное ограничение появилось, потому что имя драйвера, содержащееся в заголовке устройства, имеет только 8 символов.

**Какие имена присваивают блоковым устройствам?**

Блоковым устройствам присваиваются буквы драйвера, и они становятся одним или более системными логическими драйверами, например A:, B:, C: и т.д.

**Как работают драйверы потоковых устройств?**

Потоковые драйверы разработаны для поддержки сетевых устройств. Для таких устройств, как сетевые карты есть понятия входного и выходного потоков. Входной поток посылает все пакеты, полученные картой, в ядро, а ядро, в свою очередь, посылает пакеты в сетевое устройство. Часто желательно, чтобы различные операции выполнялись над потоком: фильтрации, разбиения одного потока на два по определенному алгоритму, и т.д. Потоковые устройства – это абстрактные устройства, которые выполняют алгоритмы над потоками. Множество таких устройств могут быть ранжированы в определенном порядке для выполнения множества действий над потоком. Драйвер устройства говорит конкретному аппаратному обеспечению, какие манипуляции над потоком должны быть проделаны.

**Перечислите методы доступа к ПУ**

Портовый доступ. Обмен данными через порты данных. Управление через порты с использованием сигналов прерываний. По этому методу работают клавиатура, таймеры, устройства управления звуком, система управления аппаратными прерываниями, а также системные порты последовательной (COM) и параллельной (LPT) передачи данных.

Обмен данными с ЦП через адресацию в памяти (окно памяти). Управление через командный порт и порт программирования. По этому методу работают видеосистемы и системы управления верхними блоками памяти.

Передача данных через процессор прямого доступа к памяти (DMA - Direct Memory Access). Управление через порты программирования. По этому методу работают программы обмена для дисковых накопителей.

**Назначение каждого бита в регистрах параллельного интерфейса**

|  |  |  |  |
| --- | --- | --- | --- |
| Регистр | Тип доступа и номер бита | Назначение | Описание |
| 0 | Read/Write | Data (DR) | Байт данных, передаваемый через параллельный порт |
| 1 | Read only | Status (SR) | Текущее состояние порта |
|  | Биты 0—1  Бит 2  Бит З  Бит 4  Бит 5  Бит 6  Бит 7 | PIRQ  ERROR#  SELECT  OUT\_OF\_PAPER  ACK  BUSY | Зарезервированы  0 - прерывание было запрошено портом (т.е. если сигнал АСК вызвал прерывание)  0—произошла ошибка  1 — принтер выбран (включен)  1 — в принтере отсутствует бумага  отображает состояния линии Ас1с  0 — принтер занят (1 — разрешение на вы вод очередного байта) |
| 2 | Read/Write  Бит 0  Бит 1  Бит 2  Бит З  Бит 4  Биты 5—7 | Control (CR)  STROBE  AUTO\_LF  INIT#  SELECT\_IN  ENABLE\_INT | Команды, посылаемые в порт  1 — строб передачи данных в/из порта  1 — автоматическая подача строки  0 — инициализировать принтер  1 — выбрать принтер  1 — разрешает прерывания по спаду сигнала на линии АСК  зарезервированы |

**Два варианта программного доступа к регистрам ПУ. В чём отличие?**

1й вариант: Используем специфическую для данного процессора инструкцию ввода/вывода в порт, а значит, используем адрес порта ввода/вывода. Команды для доступа к пространству ввода/вывода - IN и OUT.

2й вариант: Адресам доступа к периферийному устройству предоставляются особые области в памяти. Запись по адресу в памяти означает перенос данных в устройство.

Команда для доступа к пространству памяти — MOV

Например, **доступ к видеопамяти** — доступ по специальным адресам с использованием стандартных обращений к памяти.

**Какой недостаток имеет HAL?**

Недостаток в том, что аналогов для работы со строками байт (а также 16-разрядных слов и 32-разрядных двойных слов), соответствующих инструкциям INSB, OUTSB, INSW, OUTSW, INSD и OUTSD - макроопределений HAL просто нет. Но когда они есть, они дают 2-3-х кратный выигрыш по скорости доступа, например, к параллельному порту. Поэтому иногда лучше использовать ассемблер вместо HAL.

**Два основных типа DMA операций, в чём различие?**

***Slave DМА*** = DMA с использованием системных контроллеров = Набор контроллеров на материнской плате

Плюс: количество аппаратной логики для реализации DMA в устройстве уменьшается

Плохие результаты при использовании его для двух высокоскоростных устройств

Пример - Контроллер флоппи дисководов

***Bus master DMA***

ПУ содержат собственные средства обеспечения ***DМА***

**Данные о ПУ, используемые драйвером. Когда драйвер их использует?**

В случае PnP устройств, идентификаторы производителя и типа устройства являются критерием выбора драйвера при загрузке системы или же при подключении устройства (если оно было подключено после загрузки).

**Какие ОС имеют задачи распределения памяти или защиты ввода-вывода?**

OS/2, Windows начиная с 95й, UNIX-подобные. Может ещё что-то, но эти точно.

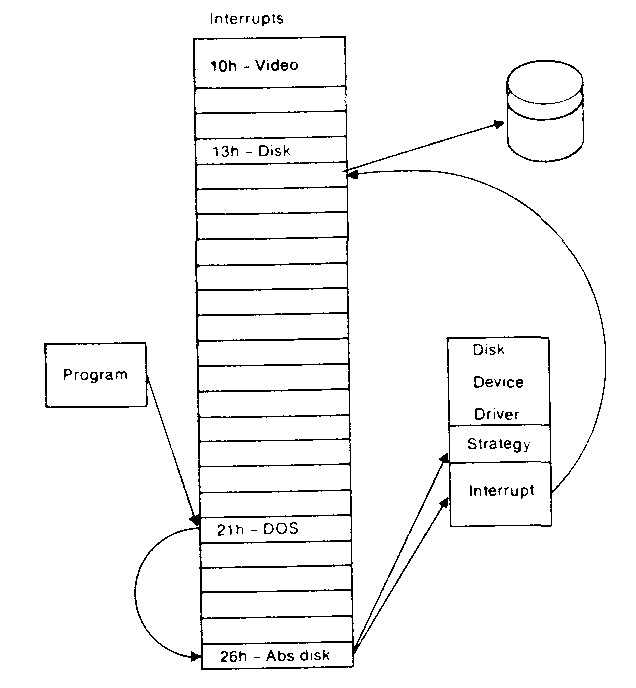
**Почему не бывает потоковых драйверов под DOS?**

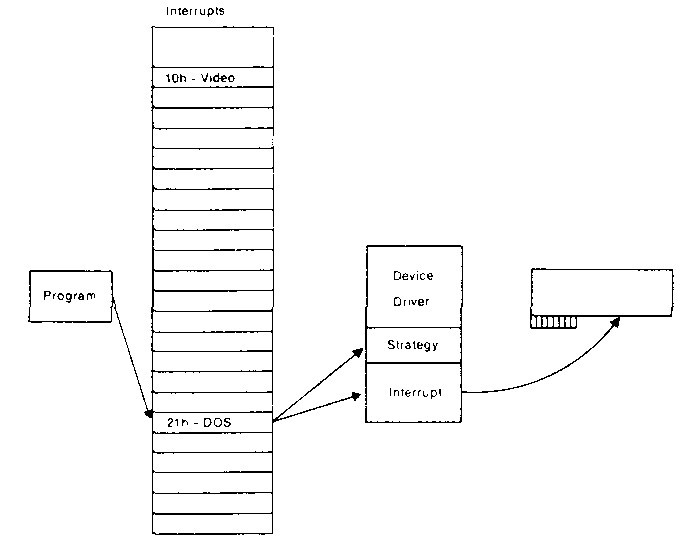
(в методах нету, пишу как думаю)

Что значит потоковый драйвер? (в основном используются для сетевых устройств) Это значит, что организуется как минимум 2 потока (восходящий и нисходящий). DOS, как известно, однопоточная ОС.

**Драйвер под DOS для стандартного и нестандартного ПУ. В чём различия?**

Сначала стандартный. Рисуем схемку

Когда прикладная программа вызывает функцию DOS Int21h, чтобы выполнить любую функцию ввода-вывода, драйверы устройства оказываются вовлеченными почти в каждом таком случае. Когда сервисная подпрограмма получает управление, она изменяет установки и вызывает требуемую функцию (к примеру рассмотрим Int26h (Absolute Disk Write). Команда Int26h устанавливает заголовок запроса (буфер команд для драйвера) в зарезервированной области памяти и вызывает стратегическую подпрограмму для драйвера устройства, которая работает с диском. Стратегическая подпрограмма просто сохраняет адреса требуемого заголовка и возвращает управление Обработчику прерываний.

Затем, DOS вызывает подпрограмму Interrupt - прерывающую часть драйвера (имя этой подпрограммы, как и имя стратегической подпрограммы, не отражают их действительные функции). Прерывающая часть драйвера – подпрограмма Interrupt читает заголовок запроса и из него определяет, что требуется. Прерывающая часть драйвера затем передает управление соответствующей внутренней подпрограмме и выполняет запись на диск, вызвав BIOS - функцию записи на диск – Int13h. Когда запись на диск завершена, управление возвращается по цепочке к программе приложения, коды статуса настраиваются для реагирования на каждую ожидаемую подпрограмму вызова.

Нестандартный.

Если устройство нестандартное, в BIOS нет программы, которая бы знала в деталях, как взаимодействовать с новым устройством, поэтому драйвер должен будет общаться с новым устройством напрямую, без использования стандартных функций BIOS. Рисуем схемку

Для добавляемого уникального устройства наш драйвер должен обращаться с аппаратурой напрямую во всех деталях.

Когда мы добавляем аппаратуру к системе и пишем драйвер для аппаратуры, надо учесть все детали взаимодействия.

**Перечислите основные части драйвера под DOS**

Структура драйвера показана на рисунке 3. Драйвер состоит из частей: Header (заголовок), Body (тело) и Init (процедура инициализации).

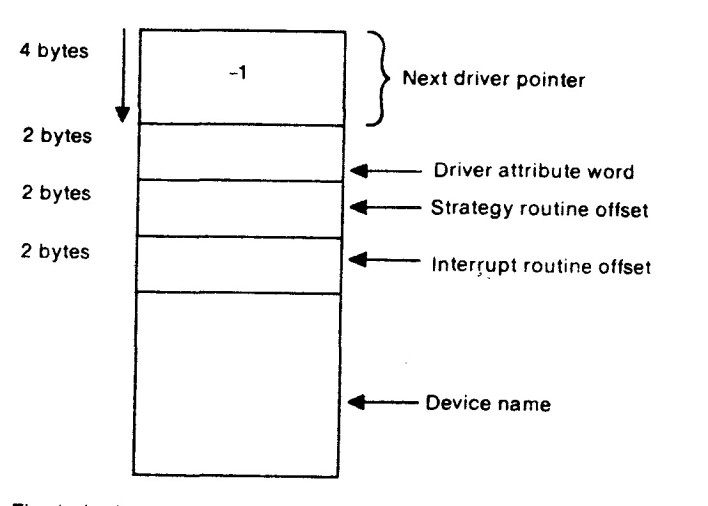
|  |
| --- |
| Header |
| Body |
| Init |

В свою очередь тело (Body) драйвера устройства содержит подпрограмму стратегии (Strategy) и подпрограмму прерывания (Interrupt).

Этот раздел рассматривает структуру драйвера. Посмотрим сначала на первую важную часть работающего драйвера – заголовок устройства.

**Пять полей заголовка Header.**

Рисуем схемку



***Next driver pointer.***

Указатель следующего драйвера. Четыре байта инициализированы как -1 (FFFFFFFFh). Указатель на следующий драйвер из списка драйверов. Последний драйвер в списке обозначен -1. (Только смещение\* этого указателя должно быть -1; сегмент может быть 0).

***Driver attribute word****.*

Список свойств драйвера. Два байта, которые определяют характеристики драйвера.

***Strategy routine offset.***

2-х байтовое смещение к подпрограмме стратегии внутри драйвера.

***Interrupt routine offset.***

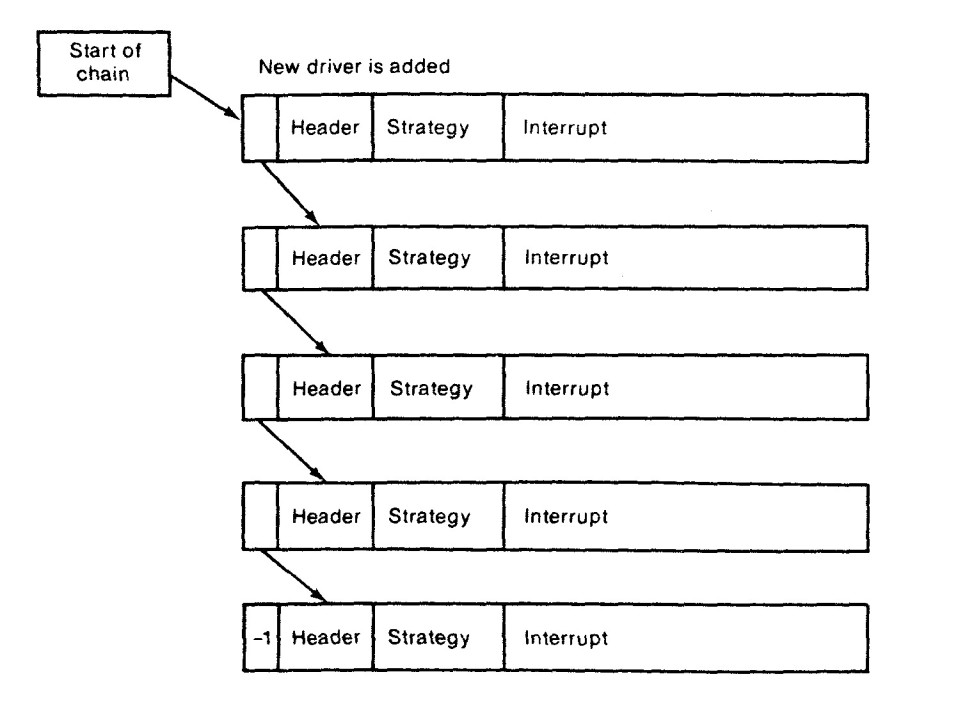
2-х байтовое смещение к подпрограмме прерывания внутри драйвера.

***Device name.***

Если устройство *символьное,* 8-символьное слева направо заполненное в бланке имя устройства появится следом. Если имя такое же, как имя существующего устройства, новый драйвер замещает существующее устройство. Если устройство *блоковое*, первый байт в поле - это номер логического устройства, связанного с драйвером, остальные байты игнорируются.

**Какая системная команда используется для указания на первый драйвер DOS?**

Используется внутренняя функция 52h прерывания 21h. После её использования в ES:BX заносится адрес векторной таблицы связи, где по смещению 04 лежит указатель на первый драйвер.

**Где в стеке располагается последний установленный драйвер ПУ?**

Последний драйвер в списке обозначен -1. (Только смещение\* этого указателя должно быть -1; сегмент может быть 0).

**Состав раздела драйвера Body. Цель каждой подпрограммы**

Тело (Body) драйвера устройства содержит *подпрограмму стратегии* (Strategy) и *подпрограмму прерывания* (Interrupt).

**Подпрограмма стратегии** не имеет ничего общего со словом «стратегия», она не старается найти лучший путь для управления устройством или что-то в этом роде. Вы можете написать подпрограмму стратегии из пяти строчек, ее цель – «запомнить», где в памяти операционной системы было определено пространство для запроса заголовка устройства (*request header - RH).*

*RH* выполняет две функции:

* Задание области данных для внутренних операций DOS.
* Задание области, в которой DOS говорит драйверу, что делать, и где драйвер отчитывается о результате своих действий.

Здесь устанавливаются указатели на соответствующие процедуры в коде драйвера.

### Итак, запросы к драйверу устройства посылаются в виде структуры данных, названной ‘request header’. Ядро DOS создает такой ‘request header’, делает его доступным, используя регистры DS:BX, и вызывает процедуру стратегии, а затем процедуру прерывания.

### Подпрограмма прерываний

Главная часть драйвера, называемая подпрограммой прерываний, выполняет всю работу. Она плохо названа, потому что она работает не как прерывание, и она заканчивается с RET чаще, чем с IRET.

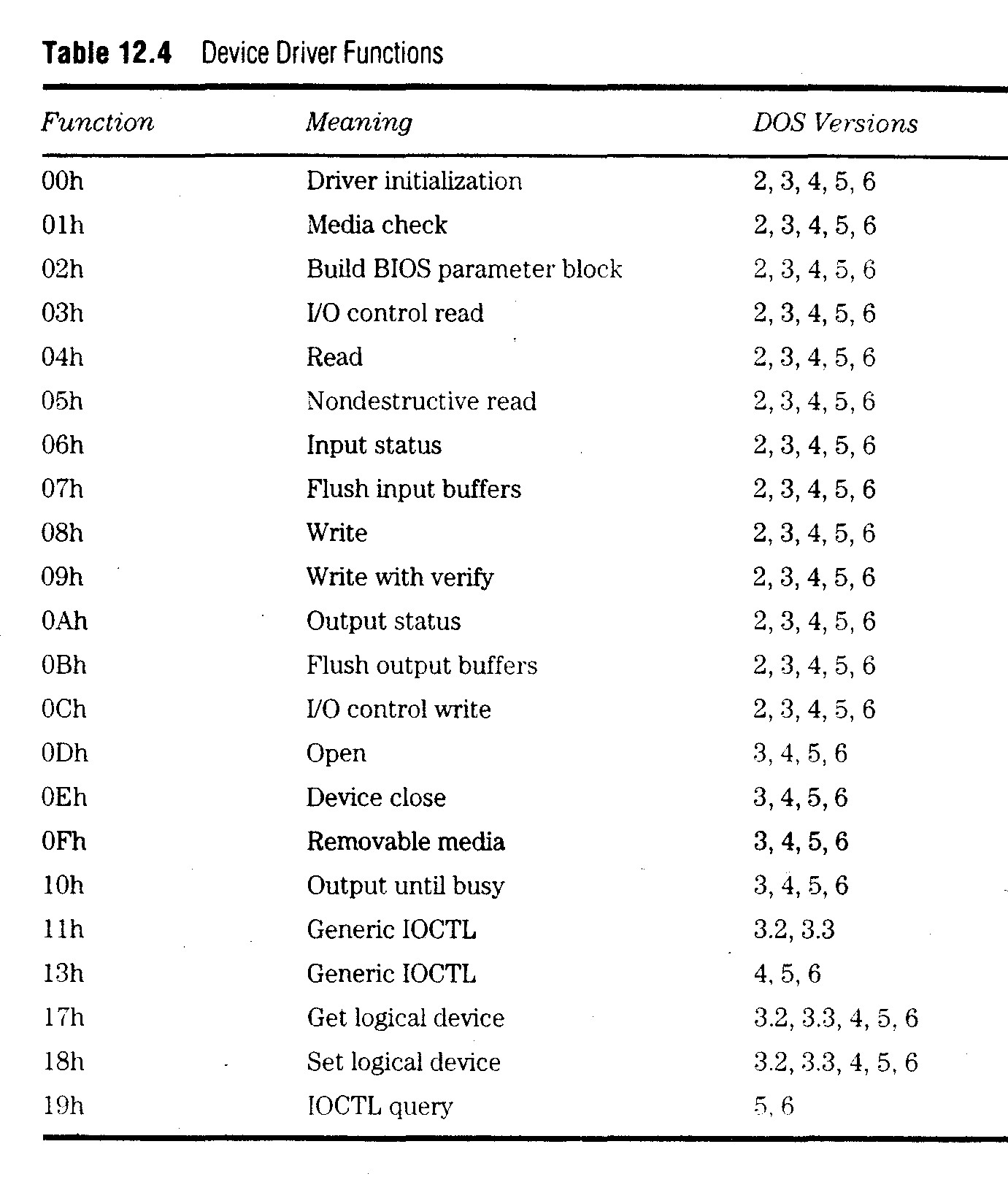
Подпрограмма прерываний содержит код для 21 функции, требуемой системой DOS v 5.0. Как бы ни назывался драйвер устройства, он достает адреса заголовка запроса и смотрит на байты смещения 02h в заголовке, чтобы найти код команды, которая показывает, какую функцию драйвер должен выполнить.

Процедура прерывания, которую также называют функцией обслуживания, анализирует команду в блоке запроса. Результат анализа выражается в соответствующем взаимодействии драйвера с аппаратурой.

Процедуры стратегии и обслуживания обе требуют цепочки различных задач, таких как вызов других подпрограмм. Для этого набор программ в цепочке, называемой драйверы устройств (по умолчанию) не всегда используются. Вместо них локальная цепочка используется, указанная в самой программе драйвера. Две главные процедуры должны управлять переключением глобальной и локальной цепочек (stack), их стартом и финишем.

**Познакомьтесь с функциями драйвера под DOS**

Знакомимся - таблица



**Перечислите последовательность транслирования и установки драйвера под DOS**

MASM 6.0

запустить macro assembler

запустить linker,

затем программу EXE2BIN, чтобы создать драйвер в виде исполняемого модуля

Отредактировать файл CONFIG.SYS, включив следующую строку:

DEVICE=A:DRVR.SYS