2020/2021

1. Основные характеристики и классификация ЭВМ.

основные характеристики:

- отношение стоимость/производительность;

- надежность и отказоустойчивость;

- масштабируемость;

- совместимость и мобильность программного обеспечения.

Классификация ЭВМ:

* **Персональные компьютеры и рабочие станции**
  + Персональные компьютеры (ПК) появились в результате эволюции миникомпьютеров при переходе элементной базы машин с малой и средней степенью интеграции на большие и сверхбольшие интегральные схемы.
  + Миникомпьютеры стали прародителями и другого направления развития современных систем - 32-разрядных машин.
* X**-терминалы**

представляют собой комбинацию бездисковых рабочих станций и стандартных ASCII-терминалов. Бездисковые рабочие станции часто применялись в качестве дорогих дисплеев и в этом случае не полностью использовали локальную вычислительную мощь. Одновременно многие пользователи ASCII-терминалов хотели улучшить их характеристики, чтобы получить возможность работы в многооконной системе и графические возможности. Совсем недавно, как только стали доступными очень мощные графические рабочие станции, появилась тенденция применения "подчиненных" X-терминалов, которые используют рабочую станцию в качестве локального сервера

* **Серверы**

Прикладные многопользовательские коммерческие бизнес-системы, включающие системы управления базами данных и обработки транзакций, крупные издательские системы, сетевые приложения и системы обслуживания коммуникаций, разработку программного обеспечения и обработку изображений все более настойчиво требуют перехода к модели вычислений "клиент-сервер" и распределенной обработке. В распределенной модели "клиент-сервер" часть работы выполняет сервер, а часть пользовательский компьютер (в общем случае клиентская и пользовательская части могут работать и на одном компьютере). Существует несколько типов серверов, ориентированных на разные применения: файл-сервер, сервер базы данных, принт-сервер, вычислительный сервер, сервер приложений. Таким образом, тип сервера определяется видом ресурса, которым он владеет (файловая система, база данных, принтеры, процессоры или прикладные пакеты программ).

* **Мейнфрейм** - это синоним понятия "большая универсальная ЭВМ". Мейнфреймы и до сегодняшнего дня остаются наиболее мощными (не считая суперкомпьютеров) вычислительными системами общего назначения, обеспечивающими непрерывный круглосуточный режим эксплуатации. Они могут включать один или несколько процессоров, каждый из которых, в свою очередь, может оснащаться векторными сопроцессорами (ускорителями операций с суперкомпьютерной производительностью). В нашем сознании мейнфреймы все еще ассоциируются с большими по габаритам машинами, требующими специально оборудованных помещений с системами водяного охлаждения и кондиционирования. Однако это не совсем так. Прогресс в области элементно-конструкторской базы позволил существенно сократить габариты основных устройств. Наряду со сверхмощными мейнфреймами, требующими организации двухконтурной водяной системы охлаждения, имеются менее мощные модели, для охлаждения которых достаточно принудительной воздушной вентиляции, и модели, построенные по блочномодульному принципу и не требующие специальных помещений и кондиционеров.
* **Кластерные сиситемы**

наибольшее распространение в мире получили кластерные системы, благодаря тому, что они обеспечивают достаточно высокий уровень готовности систем при относительно низких затратах. Термин "кластеризация" на сегодня в компьютерной промышленности имеет много различных значений. Строгое определение могло бы звучать так: "реализация объединения машин, представляющегося единым целым для операционной системы, системного программного обеспечения, прикладных программ и пользователей". Машины, кластеризованные вместе таким способом могут при отказе одного процессора очень быстро перераспределить работу на другие процессоры внутри кластера. Это, возможно, наиболее важная задача многих поставщиков систем высокой готовности

1. Функциональная организация ЭВМ.

**Функциональная организация ЭВМ** - это абстрактная модель ЭВМ, описывающая функциональные возможности машины и предоставляемые ею услуги. Функциональная организация ЭВМ в значительной степени определяется предъявляемыми к ней требованиями, уровнем подготовки потенциальных пользователей, типом решаемых ими задач, потребностями в развитии компьютера (по емкости ЗУ, разрядности, составу периферийных устройств и др.).

Термин «**функциональная организация ЭВМ**» часто используют в качестве синонима (в некотором смысле) более широкого термина «архитектура ЭВМ».

**Архитектура ЭВМ** – это абстрактное представление ЭВМ, которое отражает ее структурную, схемотехническую и логическую организацию. Понятие архитектуры ЭВМ является комплексным и включает в себя:

– Структурную схему ЭВМ;

– Средства и способы доступа к элементам структурной схемы;

– Организацию и разрядность интерфейсов ЭВМ;

– Набор и доступность регистров;

– Организацию и способы адресации памяти;

– Способы представления и форматы данных ЭВМ;

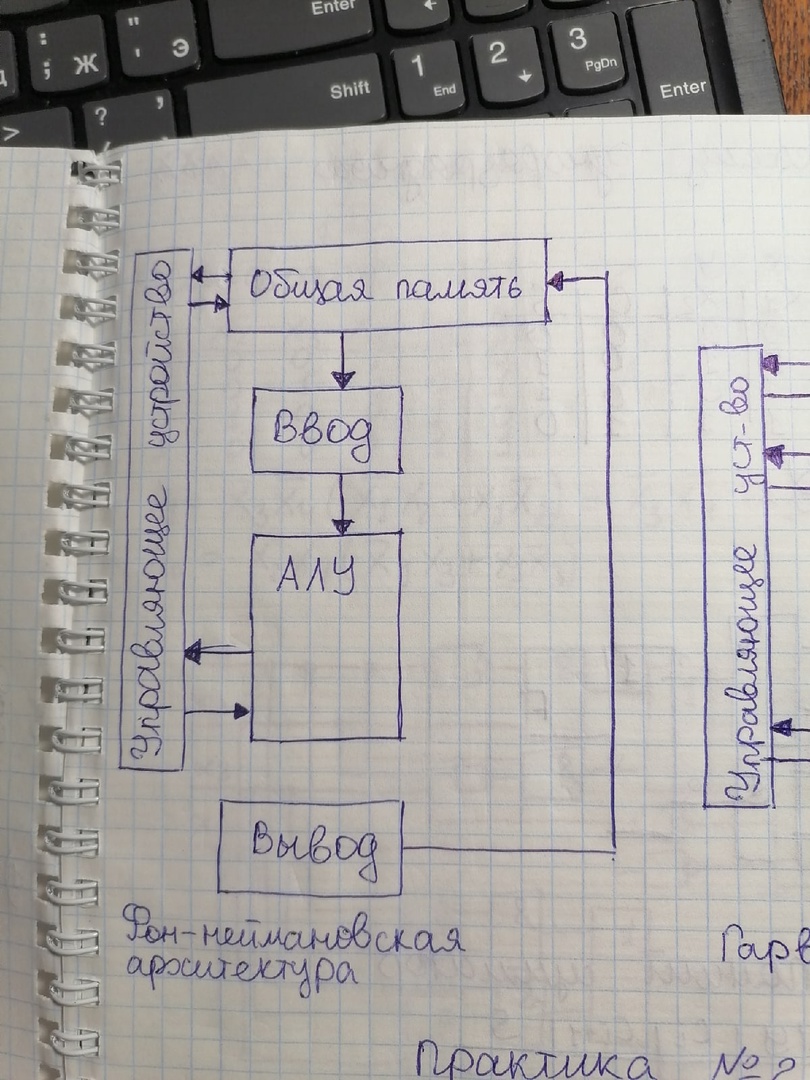
– Набор машинных команд ЭВМ;

– Форматы машинных команд;

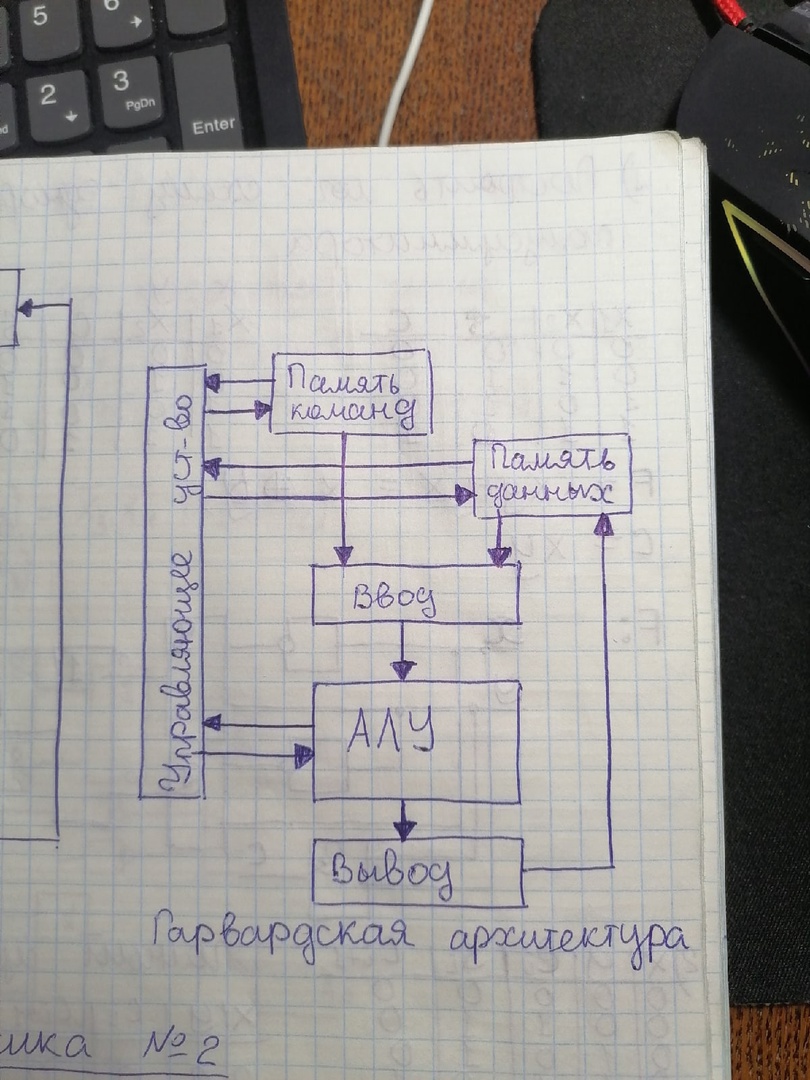
– Обработку нештатных ситуаций (прерываний).

1. Структурная организация ЭВМ. Фон-Неймановская и гарвардская архитектура.

В общем случае под **структурной организацией ЭВМ** понимается некоторая физическая модель, устанавливающая состав, порядок и принципы взаимодействия основных функциональных частей машины (без излишних деталей их технической реализации).



**Фон Неймановская (принстонская) архитектура** (предложена Джо фон Нейманом в 1945 г.) предполагает, что программа и данные находятся в общей памяти, доступ к которой производится по одной шине данных и команд. Основным преимуществом такого подхода является его гибкость, так как для изменения программы достаточно просто загрузить новый код в соответствующую область памяти. По существу, фон-неймановская архитектура, показанная на рис., состоит из центрального процессора (ЦПУ), памяти и общей шины (магистрали), по которой в обоих направлениях пересылаются данные. ЦПУ также должен взаимодействовать и с окружающим миром. При этом данные к/от соответствующих интерфейсных портов передаются по одной общей шине данных. Огромным преимуществом фон-неймановской архитектуры является ее простота, поэтому данная концепция легла в основу большинства компьютеров общего назначения. Однако использование общей шины означает, что в любой момент времени может выполняться только одна операция. Соответственно, пересылка данных между ЦПУ и памятью данных не может осуществляться одновременно с выборкой команды. Эта особенность называется фон-неймановским узким местом.



**Гарвардская архитектура** (реализована в 1944 г. в ЭВМ Гарвардского университета) соответствует структуре с разделенными устройствами памяти команд и данных и отдельными шинами команд и данных.

Такая концепция была более эффективной, чем фон-неймановская (принстонская) архитектура, поскольку код программы мог считываться из памяти программ дновременно с обменом между ЦПУ и памятью данных или с операциями ввода/вывода. Однако такие машины были намного сложнее и дороже в изготовлении. А с учетом уровня технического развития 40 - 50-х годов, высоких экономических затрат, они не получили широкого распространения. Однако с развитием больших интегральных схем и технологии гарвардская архитектура снова оказалась в центре внимания.

Две физически разделенные шины, используемые для передачи информации между ЦПУ и неперекрывающимися областями памяти. Каждая память имеет собственную шину адреса, поэтому адрес ячейки памяти программ никоим образом не связан с адресом ячейки памяти данных. В таком случае говорят, что обе области памяти находятся в различных адресных пространствах. Память данных иногда называют файловой памятью, в этом случае n-я ячейка обозначается как файл n. Фоннеймановская архитектура более экономно расходует аппаратные ресурсы, а гарвардская позволяет реализовать большее быстродействие.

1. Основные принципы повышения производительности ЭВМ.

* **На самом нижнем уровне - это передовая технология конструирования и изготовления быстродействующих элементов и плат с высокой плотностью монтажа.**
* Следующий шаг в направлении повышения быстродействия предполагает **уменьшение числа логических уровней** при реализации комбинационных схем.
* Следующий уровень охватывает **способы реализации основных операций**, таких как сложение, умножение и деление.
* Далее, быстродействие вычислительных систем может быть повышено за счет **реализации аппаратными или программно-аппаратными средствами встроенных сложных команд** соответствующих тем или иным функциям, встречающимся во многих практических вычислениях
* Еще один резерв, используемый для повышения эффективности работы процессора,- это **сокращение временных затрат при обращениях к памяти.**
* Наконец, мы подходим к структуре алгоритма, по которому работает система. На этом уровне основной подход к повышению быстродействия состоит в том, чтобы выполнять **одновременно несколько команд**.

1. Дайте определение ЭВМ и перечислите ее особенности.

**ЭВМ**— программируемое функциональное устройство, состоящее из одного или нескольких взаимосвязанных центральных процессоров, периферийных устройств, управление которыми осуществляется посредством программ, располагающимихся в оперативной памяти. Эта машина может производить большой объем вычислений, содержащих большое количество арифметических, логических и других операций без вмешательства пользователя в течение периода выполнения.

1. Какие устройства входят в состав ЭВМ и каково их назначение?

* **Микропроцессор**

Центром вычислительной системы является ее процессор. Это основное звено, или "мозг" компьютера. Именно процессор обладает способностью выполнять команды, составляющие компьютерную программу. Персональные компьютеры строятся на базе микропроцессоров, выполняемых в настоящее время на одном кристалле (чипе).

* **Основная (материнская) плата и шина**

Когда данные передаются внутри компьютерной системы, они проходят по общему каналу, к которому имеют доступ все компоненты системы. Этот путь получил название шины данных. Необходимо отметить, что понятие «шина данных» имеет общее значение, конкретно же и микропроцессор имеет свою шину данных и оперативная память. Когда нет специального уточнения, то речь идет, как правило, об общей шине, или иначе шине ввода-вывода. Эта шина формируется на сложной многослойной печатной плате - основной, или иначе, материнской.

* **Память**

Одним из основных элементов компьютера, позволяющим ему нормально функционировать, является память. Внутренняя память компьютера (оперативная память и кэш-память) - это место хранения информации, с которой он работает. Она является временным рабочим пространством. Информация во внутренней памяти не сохраняется при выключении питания, на диске же или дискете может храниться годами без потребления питания. В постоянной памяти (ROM) персонального компьютера записан набор программ базовой системы ввода-вывода (BIOS). Эта память энергонезависима и BIOS всегда готова к чтению при включении питания компьютера.

Основная (оперативная) память (RAM – Random Access Memory – память с произвольным доступом) компьютера отличается от прочих устройств памяти, прежде всего тем, что к любому ее месту можно обратиться одинаково быстро, даже если делать это в случайном (произвольном) порядке (random access).

* **Накопители на подвижном магнитном носителе**(винчестеры)
* **Накопители на гибких магнитных дисках (**дискеты**)**
* **Оптические диски**
* **Блоки расширения**

Они могут использоваться для подключения дополнительных устройств (адаптеров дисплея, контроллера дисков и т.п.).

1. Поясните назначение и основные параметры ЭВМ.

Электронная вычислительная машина, компьютер – комплекс технических средств, предназначенных для автоматической обработки информации в процессе решения вычислительных и информационных задач.

К основным параметрам ЭВМ относятся

* Производительность
* число разрядов машинного слова
* емкость оперативной памяти (ОП)
* максимальная скорость передачи информации между центральной частью ЭВМ (процессор (ПР) и ОП) и периферийными (внешними) устройствами
* надежность
* габаритные размеры
* потребляемая мощность.

1. Перечислите режимы работы ЭВМ и поясните их характерные особенности.

В современных ЭВМ можно выделить следующие режимы работы:

- **однопрограммный;**

Режим, при котором выполняется не более одной независимой

программы. При таком режиме работы ЭВМ решение задачи начинается с загрузки программы в ОП, после чего ЭВМ последовательно выполняет команды программы. При этом в каждый момент времени работает одно ее устройство, в то время как остальные простаивают в ожидании окончания ранее начатого действия. Значительные потери рабочего времени ЭВМ связаны смедленной работой устройства ввода-вывода по сравнению с работой быстродействующих устройств (АЛУ, ЦУУ, ОЗУ и т.д.).

**- мультипрограммный;**

Режим, при котором в памяти ЭВМ хранится несколько программ и выполнение одной программы может быть прервано для перехода к выполнению другой с последующим возвратом к прерванной программе. При совместном выполнении нескольких программ простои оборудования

уменьшаются, поскольку увеличивается вероятность того, что среди находящихся в ЭВМ программ имеется одна, готовая к использованию освободившегося оборудования. Для уменьшения простоев оборудования ЭВМ широко применяют метод организации параллельной работы устройства ЭВМ

за счет совмещения различных операций при работе ЭВМ. В целях более эффективного использования ЭВМ организуют мультипрограммную обработку информации на ЭВМ так, чтобы ею параллельно выполнялись команды, относящиеся к различным и независимым программам. Мультипрограммный режим повышает производительность ЭВМ за счет увеличения числа задач, решаемых ЭВМ в течение некоторого промежутка времени. При этом время решения отдельной задачи увеличивается по сравнению с временем решения ее в однопрограммном режиме.

**- пакетной обработки;**

В этом режиме задачи (программы и данные), подготовляемые многими пользователями ЭВМ, собираются в пачки-пакеты. Пакет состоит из заданий (не более 15), относящимся ко многим задачам, обработка которых занимает не менее часа машинного времени. Различают два режима пакетной обработки. В первом число задач, выполняемых одновременно, фиксируется, а во втором не фиксируется, но в

процессе обработки пакета ЭВМ оно может изменяться пакета ЭВМ оно может изменяться динамически. Пакет, предварительно записанный на том или ином носителе информации, вводится в ОЗУ ЭВМ. Когда пакет загружен, ЭВМ выбирает на обработку несколько задач и начинает выполнять их мультипрограммном режиме. Когда решение одной группы задач пакета закончено, из него выбирается для обработки следующая группа, это продолжается до тех пор, пока не будет обработана последняя группа задач пакета. После этого в ЭВМ вводится новый пакет задач.

Пакетная обработка данных позволяет увеличить производительность ЭВМ и уменьшить стоимость машинной обработки информации.

**- разделения во времени;**

Этот режим обеспечивает непосредственный и одновременный доступ к ЭВМ некоторому количеству пользователей чаще всего с дистанционно удаленных пунктов (терминалов). Терминал – периферийное устройство,

предназначенное для обслуживания одного человека, решающего задачи на ЭВМ.

Пользователи с помощью терминалов вводят в ЭВМ исходные данные и программы и получают результаты вычислений. ЭВМ предоставляет каждому активному терминалу квант времени, равный секундам и долям секунды. По

истечении этого времени ЭВМ переходит к обслуживанию следующего пользователя. За некоторый период времени ЭВМ обслуживает всех пользователей. При достаточно высоком быстродействии ЭВМ у отдельного пользователя создается иллюзия непрерывного контакта с ЭВМ.

Разделение времени позволяет устранить потери машинного времени, связанные с вмешательством оператора в работу ЭВМ из-за сравнительно низкой его скорости реакции, необходимости выполнения им определенных

действий вне ЭВМ и медленного ввода информации с пульта оператора. При мультипрограммной работе ЭВМ в промежуточные паузы работы одного оператора к ЭВМ имеют доступ другие, что позволяет обеспечить полную

загрузку внутренних устройств ЭВМ и тем самым поднять эффективность ее работы. Режим разделения времени совместим с режимом пакетной обработки данных, которая предусматривается в ЭВМ для решения задач в отдельные периоды времени, когда пользователи не загружают ЭВМ полностью.

**- диалоговый;**

Режим (режим “запрос-ответ”), при котором все программы

пользователей постоянно хранятся в памяти ЭВМ и пользователи имеют непосредственный доступ к ЭВМ. От пользователей в ЭВМ поступают входные данные и запросы с пультовых пишущих машинок или дисплеев. Ответ формируется по программе, соответствующей определенному запросу. Выбор допустимых запросов ограничен емкостью памяти. Каждый запрос имеет соответствующий приоритет и временные ограничения на срок обслуживания

**- режим реального времени**.

Режим, при котором ЭВМ управляет работой какого-либо объекта или технологического процесса. Особенностью работы в реальном масштабе времени является то, что, помимо арифметической и логической обработки,

выполняется слежение за работой объекта или прохождение некоторого процесса. Реализация этого режима привела к усложнению устройств и программного обеспечения ЭВМ.

1. Укажите основные направления развития и использования ЭВМ.

История компьютера тесным образом связана с попытками облегчить и автоматизировать большие объемы вычислений.

Даже простые арифметические операции с большими числами затруднительны для человеческого мозга. Поэтому уже в древности появилось простейшее счетное устройство - абак. В семнадцатом веке была изобретена логарифмическая линейка, облегчающая сложные математические расчеты. В 1820 году француз Шарль де Кольмар создал арифмометр, способный производить умножение и деление.

Основные идеи, которые лежат в основе работы компьютеров, были изложены еще в 1833 году английским математиком Чарльзом Бэббиджем.

В 1888 году американский инженер Герман Холлерит сконструировал первую электромеханическую счетную машину. Эта машина, названная табулятором, могла считывать и сортировать статистические записи, закодированные на перфокартах.

Основным активным элементом первого поколения ЭВМ являлась электронная лампа. Остальные компоненты электронной аппаратуры - это обычные резисторы, конденсаторы, трансформаторы. На смену электронным лампам в машинах второго поколения пришли транзисторы. В отличие от ламповых машин, транзисторные машины обладали большим быстродействием, емкостью ОП, надежностью. Существенно уменьшены размеры, масса и потребляемая электроэнергия. Большим достижением явилось применение печатного монтажа. Машины второго поколения обладали большими вычислительными и логическими возможностями. Появились машины для решения научно-технических задач, экономических задач, для управления производственными процессами и различными объектами (управляющие машины). Третье поколение ЭВМ ( с1962 г.) характеризовалось широким применением интегральных схем, заменивших большинство транзисторов и различных деталей. Благодаря интегральных схемам удалось существенно улучшить технические и эксплуатационные характеристики машины. Этому же способствовало применение многослойного печатного монтажа. Дальнейшее развитие получило программное обеспечение, особенно - операционные системы, которые обеспечивали работу ЭВМ в различных режимах - в режиме пакетной обработки, в режиме разделения времени, в режиме запрос-ответ и т. д. Четвертое поколение ЭВМ начало развиваться с 1970 г. Для них характерно применение больших интегральных схем (БИС). Высокая степень интеграции способствовала увеличению плотности компоновки электронной аппаратуры, повышению ее надежности и быстродействия, снижению стоимости.

Размеры машины и их стоимость настолько уменьшились, что появились их новые типы - от мини ЭВМ до персональных, предназначенных для индивидуального пользования.

Кратко основную концепцию ЭВМ пятого поколения можно сформулировать следующим образом: Компьютеры на сверхсложных микропроцессорах с параллельно-векторной структурой, одновременно выполняющих десятки последовательных инструкций программы. Компьютеры с многими сотнями параллельно работающих процессоров, позволяющих строить системы обработки данных и знаний, эффективные сетевые компьютерные системы. Шестое поколение ЭВМ. Электронные и оптоэлектронные многоядерные компьютеры с массовым параллелизмом, нейронной структурой, с распределенной сетью большого числа (десятки тысяч) микропроцессоров, моделирующих архитектуру нейронных биологических систем, распознающие сложные образы.

1. Дайте определение памяти ЭВМ.

**Память ЭВМ,** совокупность технических устройств и процессов, обеспечивающих запись, хранение и воспроизведение информации в ЭВМ. Память — основная часть любой вычислительной системы или отдельной вычислительной машины, она реализуется аппаратурно — в виде комплекса взаимосвязанных запоминающих устройств (ЗУ) — и программными средствами.

1. В чем заключается различие между машинным словом и словом памяти?

**Машинное слово** — это максимальное количеество бит, которыми может оперировать процессор за раз. (количество информации, которое может быть обработано за одну инструкцию).

**Слово памяти** — количество информации, представленной в 8-ми разрядной форме, которая может быть записана в память.

1. Назовите основные характеристики ЗУ.

К основным параметрам ЗУ относятся информационная емкость, потребляемая мощность, время хранения информации, быстродействие и др.

**Информационная емкость** определяется числом ячеек памяти ЗУ и указывает максимальный объем хранимой информации. Если ЗУ рассчитано на хранение n чисел (слов), каждое из которых имеет от разрядов, то информационная емкость N определяется выражением N = n· m.

**Потребляемая мощность** — мощность, потребляемая ЗУ в установившемся режиме работы.

**Время хранения информации** — интервал времени, в течение которого ЗУ сохраняет информацию в заданном режиме.

**Быстродействие** — промежуток времени, необходимый для записи или считывания информации.

**Стоимость хранения информации(редко)**

1. Поясните разницу в функционировании стековой и магазинной памяти.

Магазинная (стековая) память – вид памяти (ЗУ), являющийся аппаратной реализацией магазинного списка – стека, запись и считывание в котором осуществляются через одну и ту же ячейку – вершину стека. Стековая память обеспечивает такой режим работы, когда информация записывается и считывается по принципу «последним записан — первым считан\* (LIFO-Last In First Out). Память с подобной организацией широко применяется для запоминания и восстановления содержимого регистров процессора (контекста) при обработке подпрограмм и прерываний.

1. Поясните функции, выполняемые ОЗУ, СОЗУ, ЛЗУ, БЗУ и ВЗУ, образующими память ЭВМ.

**Оперативное ЗУ (ОЗУ)** является основным запоминающим устройством ЭВМ, в котором хранятся выполняемые в настоящий момент процессором программы и обрабатываемые данные, резидентные программы, модули

операционной системы и т.п.

**Сверхоперативные ЗУ (СОЗУ)** – служат для хранения результатов вычислительного устройства промежуточных операций. В микропроцессорах (МП) роль СОЗУ выполняет

регистровая память.

**Буферные ЗУ (БЗУ)** – предназначены для промежуточного хранения информации при ее обмене между устройствами, работающими с разной скоростью. Эту роль выполняют регистровые схемы или ОЗУ малого объема.

**Внешнее ЗУ(ВЗУ)** предназначено для сбора и долговременного хранения больших массивов информации и часто используется для увеличения объема ОП, однако быстродействие ВЗУ ниже быстродействия ОП.

1. Что понимают под виртуальной памятью?

Под виртуальной памятью понимается совокупность ячеек всех ОЗУ и ВЗУ, имеющих сквозную нумерацию

0, 1, 2, ..., (Q-1). Программирование процессов решения задач производится в терминах математических (виртуальных) адресов 0,1,2,..., (Q-1). При этом команды ссылаются на математические адреса в предположении, что слово, идентифицированное любым математическим адресом, является доступным для процессора. Таким образом, для программиста создается одноуровневая память емкостью Q слов. В физическом отношении виртуальная память — это совокупность оперативных и внешних ЗУ, охваченных средствами преобразования матических адресов в физические (действительные) адреса ячеек и автоматизации перемещения информации между устройствами памяти.

1. В чем разница между математическими и физическими адресами памяти ЭВМ?

**Виртуальные(математические)** адреса вырабатывает транслятор, переводящий программу на машинный язык. Так как во время трансляции в общем случае не известно, в какое место оперативной памяти будет загружена программа, то транслятор присваивает переменным и командам виртуальные (условные) адреса, обычно считая по умолчанию, что программа будет размещена, начиная с нулевого адреса.

**Физические адреса** соответствуют номерам ячеек оперативной памяти, где в действительности расположены или будут расположены переменные и команды.

1. Поясните метод страничной адресации виртуальной памяти.

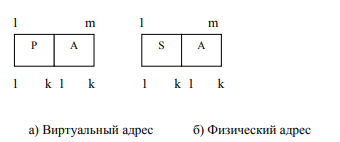


Рис. 1

Метод страничной адресации состоит в том, что виртуальная память (множество адресов) разделяется на страницы емкостью 2^K соседних адресов. Так, к странице с

адресом 0 относятся адреса 0, 1, 2, ... , (2^K −1) так далее. В результате адрес слова будет состоять из двух полей: Р, указывающих адрес страницы, и А — адрес слова в странице Р. (рис.1)

Если физическую память разделить, а сегменты, состоящие из 2^K соседних ячеек, то физические адреса в пределах одной виртуальной страницы по структуре будут полностью совпадать с математическими адресами (рис.1), где S - адрес сегмента, а А - адрес слова (ячейки) в сегменте S.

В процессе решения задачи страницы перемещаются между ОЗУ и ВЗУ. Если вычислительный процесс распределяется на страницу Р, то она вызывается в ОЗУ. Когда надобность в информации, размещенной на странице Р, отпадает, то она удаляется из ОЗУ в виртуальную память, освобождая место

для других страниц. В результате перемещения граница Р может быть помещена на любом сегменте S ОП.

Текущее состояние памяти ЭВМ характеризуется таблицей страниц (рис. 2). Отдельной странице виртуальной памяти Pi (i=1, 2, ..., Q-1) соответствует одна строка таблицы, в которой указываются параметры страницы Рi: Si — адрес сегмента ОЗУ, в котором размещается страница Рi, иначе говоря,

физический адрес страницы Pi; di - признак доступности страницы: при di=1 страница Pi, хранится в ОЗУ и недоступна для центрального процессора. Таблица страниц размещается в ОЗУ и в любое время доступна ЦП. Как видно

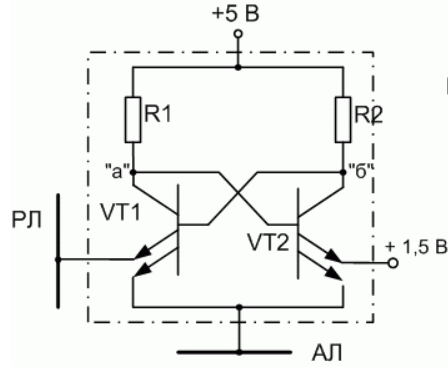
из рис. 2, 2048 страниц виртуальной памяти могут размещается в 128 сегментах ОЗУ и на магнитных дисках.

Каждой странице Рi (i=0, 1, 2, . . ., 2047) виртуальной памяти

соответствует свой сегмент, адрес которого определен в таблице страниц, и, следовательно, каждому слову присвоен свой физический адрес. Операционная система (ОС) обеспечивает преобразование математических адресов в физические адреса.



1. Поясните принцип записи и считывания информации в биполярном ОЗУ и динамическом МДП-ОЗУ.



**Запоминающий элемент статического биполярного ОЗУ**

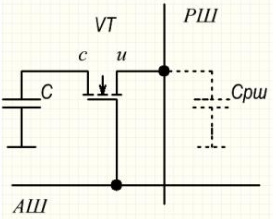
Данный ЗЭ представляет собой триггер, построенный на двух биполярных транзисторах, базы которых соединены с коллекторами "крест накрест", образуя положительную обратную связь. За уровень логического нуля принимается потенциал, близкий к потенциалу земли, а за уровень логической единицы - напряжение, близкое к + 5 В. К накопителю данный ЗЭ подключается адресной линией АЛ и разрядной линией РЛ.

запись информации -

* запись "0" - когда на разрядную линию РЛ подаётся логический 0. При этом транзистор VT 1 открыт, в точке "а" будет низкий потенциал, отличающийся от потен-циала земли на величину падения напряжения на открытом транзисторе (порядка 0,4 В). Этот низкий потенциал поступает на VT 2 и закрывает его. Таким образом, через VT 2 ток не протекает, падения напряжения на сопротивлении R 2 нет, поэтому в точке "б" будет потенциал, практически равный + 5 В. Он подается на базу транзистора VT 1 и подтверждает его открытое состояние. Триггер пришёл в устойчивое состояние: VT 1 открыт, VT 2 закрыт. Это состояние принимается за нулевое ;
* запись "1" - когда на разрядную линию РЛ подается логическая 1. VT 1 закрыт, ток через него не протекает, падения напряжения на сопротивлении R 1 нет. Следо-вательно, потенциал точки "а" будет практически равен + 5 В. Он подается на VT 2 и открывает его. Из-за разности потенциалов между + 5 В на коллекторе VT 2 и + 1,5 В на его верхнем эмиттере через открытый VT 2 протекает ток по цепи: + 5 В, сопротивление R 2, коллектор, база, верхний эмиттер VT 2. Основная часть падения напряжения в этой цепи в силу малого сопротивления открытого транзистора VT 2 приходится на сопротивление R 2. Поэтому в точке "б" будет низкий потенциал. Он поступает на базу транзистора VT 1 и подтверждает его закрытое состояние. Таким образом триггер пришёл в другое устойчивое состояние: VT 1 закрыт, VT 2 открыт. Оно принимается за единичное.

считывание информации - когда на РЛ подается проме-жуточный потенциал + 1,5 В. Доступ к ЗЭ по-прежнему обеспечивается подачей на адресную линию логической 1. При этом возможны варианты:

* чтение "0". Если ЗЭ находился в состоянии логического 0 ( VT 1 был открыт, а VT 2 закрыт) по РЛ потечёт больший ток, который преобразуется с помощью схем обрамления и на выход схемы подаётся как потенциал логического нуля;
* чтение "1". Если же в предыдущий момент времени ЗЭ находился в состоянии логической 1 ( VT 2 был открыт, а VT 1 закрыт) по РЛ потечёт меньший ток, который преобразуется с помощью схем обрамления и на выход схемы подаётся как потенциал логической единицы.



**Элемент памяти динамического типа на МДП-транзисторах.**

Запись

Подается высокий потенциал на адресную шину. На разрядная шину подается либо 0, либо 1, которые записываются в емкость С. Ток через транзистор может протекать в обоих направлениях, так как сток и исток равнозначны.

Считывание

Подается сигнал только на адресную шину, транзистор открывается. Напряжение с емкости С передается на разрядную шину, предварительно разряженную до Uопорного. Информация при считывании разрушается. Амплитуда сигнала на входе усилителя не превышает единиц милливольт. Требуется усилитель считывания.

1. Поясните назначение каждого блока полупроводникового OЗУ и принцип записи и считывания информации.
2. Какая оперативная память называется памятью с расслоением обращений?

Многоблочная ОП, в которой допускается совместное выполнение

нескольких обращений к разным блокам ОЗУ называется **ОП с расслоением обращений**. В такой ОП блоки ОЗУ функционируют параллельно во времени, что возможно, если последующие обращения к ОП адресованы к блокам, не запятым обслуживанием предшествующих запросов. Степень расслоения обращений характеризуется коэффициентом расслоения, равным среднему

числу обращений к ОП, которые могут быть приняты на обслуживание одновременно. Чем выше коэффициент расслоения, тем выше производительность.

1. Поясните достоинства СОЗУ с магазинной адресацией перед СОЗУ с прямой выборкой.

1. Поясните разницу между долговременными, программируемыми и перепрограммируемыми ПЗУ.

**ПЗУ** – Энергонезависимая память, используется для хранения данных, которые никогда не потребуют изменения. Содержание постоянной памяти "зашивается" в устройстве при его изготовлении для постоянного хранения. Из ПЗУ можно только читать. В постоянную память записывают программу управления работой самого процессора, программы управления дисплеем, клавиатурой, принтером, внешней памятью, программы запуска и остановки компьютера, программы тестирования устройств.

**ППЗУ** – Программирование этих микросхем осуществляется только с помощью специальных программаторов. Память представляет собой двумерный массив проводников (строк и столбцов) на пересечении которых находится специальная перемычка из металла (например, нихрома или титаново-вольфрамового сплава) или аморфного кремния. Программирование заключается в пропускании через соответствующую перемычку тока, который её расплавляет или испаряет. Восстановление расплавленных перемычек невозможно. Записанное в PROM, невозможно изменить. PROM содержит данные даже когда компьютер выключен.

**Перепрограммируемыми постоянными запоминающими устройствами** называются запоминающие устройства, в которых информация может стираться, вновь заноситься в ячейки памяти и сохраняться при отключении питания. Стирание и запись информации в ППЗУ могут осуществляться электрическим и неэлектрическим путем в зависимости от используемого элемента памяти.

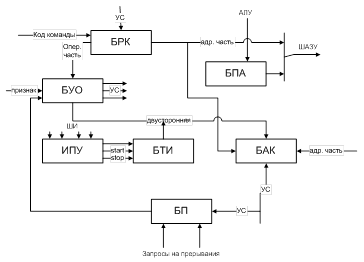
1. Какие вы знаете способы программирования ПЗУ?

Только с помощью специального устройства — **программатора ПЗУ** (как однократно, так и многократно прошиваемые), использование программатора необходимо, в частности, для подачи нестандартных и относительно высоких напряжений (до ±27 В) на специальные выводы.

**Внутрисхемно перепрограммируемые ПЗУ** (англ. in-system programming, ISP) — микросхемы, имеющие внутри генератор всех необходимых высоких напряжений, могут быть перепрошиты программным способом, то есть, без программатора и без выпайки из печатной платы.

1. Поясните назначение и структуру центрального устройства управления.

**Центральное устройство управления (ЦУУ)** - совокупность блоков и узлов процессора, обеспечивающая координирование работы всех устройств ЭВМ и управление ими для всех принятых в данной ЭВМ режимов работы.



БРК - Блок Регистра Команд

БПА - Блок Переадресации Адресов

БТИ - Блок Тактовых Импульсов

ИПУ - Инженерный Пульт Управления

БП - Блок Прерываний

БАК - Блок Адреса Команд

БУО - Блок Управления Операциями

УС - Управляющие Сигналы

1. Поясните последовательность выбора команды и операндов из ОЗУ.

Для реализации **выборки исполняемой команды** из ОЗУ необходимо код со счетчика команд (СК) = k передать в ОЗУ, обратиться в ячейку ОЗУ с адресом k и содержимое этой ячейки, являющееся кодом этой команды, передать на регистр команд.

**Выборка операнда**. Необходимо код из поля адреса операнда – из РК передать в ОЗУ, обратиться к ячейке с адресом в оперативной памяти и код этой ячейки передать в АЛУ.

1. Каково назначение базовых и индексных регистров в схеме обработки адресов ?

**Регистр (BX) является базовым регистром**. Это единственный регистр общего назначения, который может использоваться в качестве индекса для расширенной адресации. Регистр, предназначенный для хранения базового адреса

**Индексные регистры (SI и DI)** применяются для расширенной адресации и для использования в операциях сложения и вычитания. Регистр (SI) является индексом источника и применяется для некоторых операций над строками. Регистр (DI) является индексом назначения и применяется также для некоторых операций над строками.

1. Каково назначение микропрограммного управляющего автомата?

Управляющий автомат должен сформировать требуемую для операционного автомата последовательность микрокоманд F1, F2, F3 на основании сигналов X1, X2 логических условий, поступающих из операционного автомата.

В отличие от устройств на "жесткой" логике, принцип работы которых однозначно определяется используемыми элементами и способом их соединения, микропрограммные автоматы с помощью одной и той же схемы могут выполнять самые разные функции. То есть они гораздо более гибкие, чем схемы на "жесткой" логике. К тому же проектировать микропрограммные автоматы с точки зрения схемотехники довольно просто. Недостатком любого микропрограммного автомата по сравнению со схемами на "жесткой" логике является меньшее предельное быстродействие и необходимость составления карты прошивки ПЗУ с микропрограммами, часто довольно сложными.

1. Поясните по схеме последовательность выполнения микропрограммы, хранящейся в ПМК.
2. В чем преимущество программируемых логических матриц (ПЛМ) перед ПЗУ при использовании их в качестве ПМК.

Структурная схема ПЛМ состоит из двух программируемых матриц, матрицы «И» и матрицы «ИЛИ». в ПЗУ вместо матрицы «И» используется дешифратор n x 2n и матрица «монтажное ИЛИ» 2n x k. В ПЛМ m<<2n и вместо минтермов, создаваемых в ПЗУ дешифратором, в результате программирования матрицы «И» реализуются конъюнктивные термы Тq=Ki,j(ν), где ν=(X0, …, Хn-1). Таким образом, можно сказать, что ПЗУ реализуют произвольную логическую функцию, заданную в виде таблицы истинности, а ПЛМ – минимизированную логическую функцию. ПЛМ характеризуется тройкой чисел n, m и k (n – число логических переменных Х, m – число реализуемых произведений Т, k – число выходов (реализуемых функций).

Поскольку в ПЗУ используется дешифратор, то количество его выходов равно 2n (n – количество входов дешифратора, а, значит, и ПЗУ). Благодаря этому существует какой-либо выход дешифратора, соответствующий любому минтерму поданному на вход ПЗУ. Другими словами, используя ПЗУ, мы можем получить комбинацию двоичного числа на его выходе, соответствующую произвольному минтерму от заданного числа переменных (входов ПЗУ). Этот факт, безусловно, является большим преимуществом ПЗУ. Но он же является и его недостатком. Дело в том, что реализовать на одном кристалле дешифратор для большого количества входов представляется неразрешимой задачей (например, при количестве входов равном 10 количество выходов должно быть 1024). Именно поэтому ПЗУ с большим количеством входов не выпускается, т.е. количество переменных, с которыми работает ПЗУ, небольшое. Поэтому очень часто требуется аппаратное усложнение логического автомата путём использования нескольких ПЗУ именно из-за ограниченного количества входов.

Однако ситуации, когда используются все возможные комбинации переменных, крайне редки, т.е. возможности ПЗУ используются не полностью. Представьте себе, что на входы ПЗУ подаются сигналы от датчиков, сигнализирующих о состоянии технологического процесса. Наверняка многие комбинации состояния датчиков существовать просто не могут, или если они и существуют, то не представляют никакого интереса с точки зрения управления данным технологическим процессом (вспомните о «неопределённых условиях»).

Именно для таких ситуаций и были разработаны ПЛМ. Количество входов у этих микросхем больше, чем количество входов у ПЗУ, но поскольку количество выходов у матрицы «И» намного меньше, чем 2n, то реализовать можно не все возможные минтермы (а нам это в большинстве случаев и не надо). Зато при таком большом количестве входов мы можем значительно аппаратно упростить синтезируемый логический автомат.

Таким образом, преимущество ПЛМ заключается в большом количестве входов (что отсутствует у ПЗУ), а недостаток – в невозможности реализации всех функций от переменных поданных на эти входы.

1. Поясните структуру и функционирование блока прерываний.

Механизм прерываний позволяет автоматически реагировать на внешние и внутренние события (переполнение таймер/счетчиков, завершение последовательного обмена). Поступление запросов прерывания возможно от следующих источников: внешний вход INT0, таймер/счетчик T0, внешний вход INT1, таймер/счетчик T1, последовательный канал TI/RI.

1. Каково назначение блока защиты оперативной памяти?

Защита памяти — это способ управления правами доступа к отдельным регионам памяти. Используется большинством многозадачных операционных систем. Основной целью защиты памяти является запрет доступа процессу к той памяти, которая не выделена для этого процесса. Такие запреты повышают надежность работы как программ, так и операционных систем, так как ошибка в одной программе не может повлиять непосредственно на память других приложений. Следует различать общий принцип защиты памяти и технологии ASLR или NX-бит.

1. Каково назначение блока управления оперативной памятью?

Блок управления памятью или устройство управления памятью (англ. memory management unit, MMU) — компонент аппаратного обеспечения компьютера, отвечающий за управление доступом к памяти, запрашиваемым центральным процессором. Его функции заключаются в трансляции адресов виртуальной памяти в адреса физической памяти (то есть управление виртуальной памятью), защите памяти, управлении кеш-памятью, арбитражем шины и, в более простых компьютерных архитектурах (особенно 8-битных), переключением блоков памяти.

1. Поясните принцип управления многоканальной многоблочной оперативной памятью.

Оперативная память (ОП) — совокупность ОЗУ, объединенных в одну систему, управляемую процессором. Для обеспечения приспосабливаемости ЭВМ к конкретным потребностям пользователей применяют принцип блочного построения 0П. Так, например, на основе блоков 03У емкостью 128 и 256 Кслов можно построить ОП любой емкости. ОП заданной емкости, составленная из нескольких блоков ОЗУ, называется многоблочной 0П.

Устройства, подключенные к 0П, обращаются к ней независимо друг от друга. Принцип обслуживания запросов к ОП - приоритетный. Устройствам присваиваются приоритеты: низший — центральному процессору, более высший — ВЗУ. ОП обслуживает очередной запрос с наивысшим приоритетом, а все остальные запросы от других устройств ожидают момента окончания обслуживания. Такой принцип обслуживания объясняется тем, что ВЗУ не могут долго ждать, так как большое время ожидания приводит к потере информации, записываемой или считываемой с непрерывно движущегося носителя. ОП, ресурсы которой распределяются между несколькими потребителями, называют 0П с многоканальным доступом.

Многоблочная ОП, в которой допускается совместное выполнение нескольких обращений к разным блокам ОЗУ называется ОП с расслоением обращений. В такой ОП блоки ОЗУ функционируют параллельно во времени, что возможно, если последующие обращения к ОП адресованы к блокам, не запятым обслуживанием предшествующих запросов. Степень расслоения обращений характеризуется коэффициентом расслоения, равным среднему числу обращений к ОП, которые могут быть приняты на обслуживание одновременно. Чем выше коэффициент расслоения, тем выше производительность.

1. Поясните назначение пульта управления ЭВМ.

**Пульт управления** — устройство для контроля и управления работой устройств и процессов. В частности для управления ЭВМ. Пульт управления содержит ряд блоков управления: блок набора информации, блок управления, сигнальный блок и т. д.

1. По каким признакам классифицируют АЛУ.

АЛУ классифицируется по нескольким признакам:

• **по способу действия над операндами**:

* а) последовательные – операнды представлены в последовательном коде и операции над ними выполняются разряд за разрядом
* б) параллельные – операнды представлены в параллельном коде и операции выполняются одновременно со всеми разрядами

• **по способу представления чисел:**

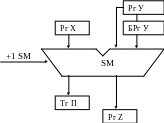
* а) АЛУ для чисел с фиксированной точкой
* б) АЛУ для чисел с плавающей точкой
* в) АЛУ для операций над десятичными числами

• **по характеру использования блоков АЛУ**:

* а) блочные – операции над числами с фиксированной и плавающей точкой, над десятичными числами и алфавитно-цифровыми полями выполняются в отдельных специализированных блоках (эти АЛУ – быстродействующие, т.к. блоки могут работать параллельно)
* б) многофункциональные – все операции выполняются одними и теми же блоками, которые коммутируются нужным образом

АЛУ управляются управляющими сигналами, которые инициируют выполнение определенных операций. Последовательность управляющих сигналов определяется кодом операции и осведомительными сигналами из АЛУ (из RG признаков).

1. Поясните назначение узлов операционного блока сложения и вычитания двоичных чисел с фиксированной запятой.



Модель АЛУ с максимальным количеством регистров для сложения и вычитания.

Рг Х, Рг У—регистры операндов (Х и У);

БРг У—буферный регистр У, используется для хранения обратного кода числа У при вычитании;

SM—сумматор, совершает операции сложения;

Рг Z—регистр результата(Z);

Тг П—триггер переполнения, используется, если результат больше операндов на разряд.

При сложении операнды поступают из ОЗУ в регистры операндов (Рг Х и Рг У). Затем операнды поступают в сумматор, складываются, и результат поступает в регистр результата (Рг Z). Если результат получился на разряд больше операндов, то триггер переполнения переходит в единичное состояние.

При вычитании операнды поступают из ОЗУ в регистры операндов. Затем вычитаемое из регистра У переходит в буферный регистр У в обратном коде. Затем операнды поступают в сумматор, происходит сложение и происходит подсуммирование 1 в младший разряд сумматора (+1 SM). Таким образом операция логического вычитания Z=X-Y сводится к изменению знака вычитаемого и операции логического сложения Z=X+(-Y).

1. В какой последовательности БМУ формирует управляющие сигналы, необходимые для выполнения операции вычитания операционным блоком?

Каждая машинная команда реализуется путем выполнения определенной микропрограммы, интерпретирующей алгоритм выполнения данной операции. Совокупность микропрограмм, необходимая для реализации систем команд ЭВМ, хранится в специальной **памяти микропрограмм**. Каждая **микропрограмма** состоит из определенной последовательности **микрокоманд**, которые после выборки из памяти преобразуются в **набор управляющих сигналов**. Эти сигналы воздействуют на все блоки **ЦП(АЛУ и др.),** обеспечивающие выполнение очередной команды и переход к следующей. Таким образом, использование в составе центрального устройства управления **БМУ** приводит к **двухуровневому принципу управления** процессом обработки данных.

* **Первый уровень** – это система команд ЭВМ(программное управление)
* **Второй** – микропрограммное управление.

1. Какие микрооперации выполняются операционным блоком под воздействием управляющих сигналов, выдаваемых БМУ?
2. Поясните структуру и назначение каждого узла операционного блока для обработки порядков.

Счетчик — накопительный узел ЭВМ, предназначенный для подсчета числа импульсов, поступивших на его вход. По структуре различают счетчики:

• с последовательным переносом;

• сквозным переносом;

• параллельным переносом;

• групповым переносом.

В зависимости от алгоритма реализации выделяют счетчики:

• суммирующие;

• вычитающие;

• реверсивные;

• с предустановкой.

В зависимости от модуля счета счетчики бывают:

• двоичные;

• десятичные.

К характеристикам счетчиков относят:

• коэффициент пересчета (число состояний счетчика) М (количество импульсов, поступивших на вход счетчика, которые переводят его в исходное состояние). Между числом разрядов счетчика п и коэффициентом пересчета М существует соотношение п > ^(М + 1);

• время реакции (регистрации) — интервал времени /рег между поступлением входного сигнала и окончанием самого длительного переходного процесса в счетчике;

• разрешающую способность — минимальный допустимый период (или максимальная частота) следования входных сигналов, при котором счетчик работает без сбоев.

1. В какой последовательности БМУ будет выдавать управляющие сигналы, необходимые для выполнения арифметических операций.

БМУ – блок местного управления, БМУ формирует управляющие сигналы, координирующие взаимодействие всех блоков АЛУ между собой.

Алгоритм операций АЛУ включает определенную последовательность элементарных действий (микроопераций). Основные микрооперации АЛУ:

1. прием кода операнда,

2. преобразование кода операции,

3. суммирование кодов операндов,

4. сдвиг кода операнда,

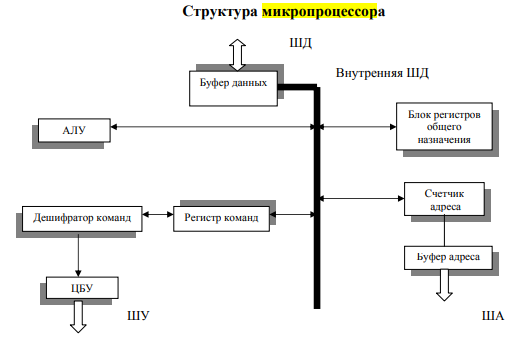
5. выдача кода результата. Из ОЗУ в блок микропрограммного управления (БМУ) поступает код команды, содержащий адрес первой микрокоманды (МК) микропрограммы выполнения данной команды. Эта микропрограмма записана в управляющей памяти, входящей в состав устройства управления. По указанному адресу считывается из управляющей памяти МК для всего микропроцессорного устройства. Эта МК содержит МК СРП БМУ, микрокоманду ОЗУ, МК устройства ввода - вывода.Сформированные МК передаются на соответствующие узлы микропро-цессорного устройства. БМУ служит для формирования адреса следующей МК. Если нет никаких условных переходов, то следующая МК считывается из следующей ячейки ПЗУ, на котором организована управляющая память. МК БМУ содержит поле условных переходов УА6...УА0, в котором записывается адрес следующей МК; поле управления признаками УФ3...УФ0 и поле управления загрузкой микрокоманды ЗМ.

1. Дайте определение процессор и микропроцессор.

**Процессор** — центральная часть ЭВМ, организующая ее работу по заданной программе. Процессор объединяет в себе АЛУ и ЦУУ, с помощью которых рабочая программа интерпретируется в вычислительный процесс.

**Микропроце́ссор** — процессор (устройство, отвечающее за выполнение арифметических, логических операций и операций управления, записанных в машинном коде), реализованный в виде одной микросхемы или комплекта из нескольких специализированных микросхем (в отличие от реализации процессора в виде электрической схемы на элементной базе общего назначения или в виде программной модели).

1. Базовая структура микропроцессора.



1. Основные отличия микропроцессоров.
2. Прямой доступ к памяти.

Для быстрого ввода-вывода блоков данных и разгрузки процессора от управления операциями ввода-вывода используют прямой доступ к памяти (DMA - Direct Memory Access).

Прямым доступом к памяти называется способ обмена данными, обеспечивающий автономно от процессора установление связи и передачу данных между основной памятью и внешним устройством.

В режиме прямого доступа к памяти используется специализированное устройство - контроллер прямого доступа к памяти, который перед началом обмена программируется с помощью центрального процессора: в него передаются адреса основной памяти и количество передаваемых данных. Затем центральный процессор от контроллера прямого доступа к памяти отключается, разрешив ему работать, и до окончания обмена может выполнять другую работу. Об окончании обмена контроллер прямого доступа к памяти сообщает процессору. В этом случае участие центрального процессора косвенное. Обмен ведет контроллер прямого доступа к памяти.

Прямой доступ к памяти (ПДП):

- освобождает процессор от управления операциями ввода-вывода;

- позволяет осуществлять параллельно во времени выполнение процессором программы с обменом данными между внешним устройством и основной памятью;

- производит обмен данными со скоростью, ограничиваемой только пропускной способностью основной памяти и внешним устройством.

ПДП разгружает процессор от обслуживания операций ввода-вывода, способствует увеличению общей производительности ЭВМ, дает возможность машине более приспособленнее работать в системах реального времени.

При работе в режиме прямого доступа к памяти контроллер ПДП выполняет следующие функции:

- принимает запрос на ПДП от внешнего устройства;

- формирует запрос микропроцессору на захват шин системной магистрали;

- принимает сигнал, подтверждающий вход микропроцессора в состояние захвата (перехода в Z-состояние, при котором процессор отключается от системной магистрали);

- формирует сигнал, сообщающий внешнему устройству о начале выполнения циклов ПДП;

- выдает на шину адреса системной магистрали адрес ячейки ОП, предназначенной для обмена;

- вырабатывает сигналы, обеспечивающие управление обменом данными;

- по окончании ПДП контроллер либо организует повторение цикла ПДП, либо прекращает режим ПДП, снимая запросы на него.

Циклы ПДП выполняются с последовательно расположенными ячейками памяти, поэтому контроллер ПДП имеет счетчик числа переданных байтов.

На рисунке приведена схема взаимодействия устройств микропроцессорной системы в режиме ПДП.

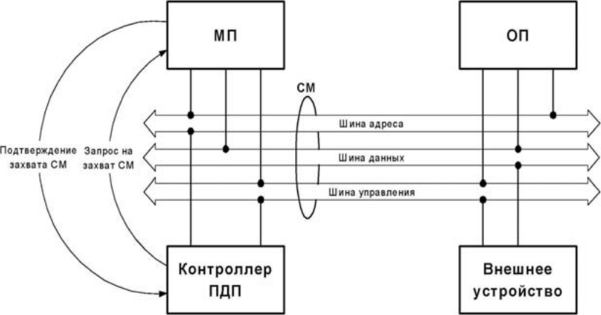


Рис. 1 Взаимодействие устройств в режиме ПДП

Контроллеры ПДП (КПДП) в IBM PC совместимы снизу вверх, т.е. программы, написанные с использованием КПДП ранних моделей ЭВМ, нормально работают и на более поздних моделях ЭВМ.

Каждый канал КПДП состоит из четырех 16-разрядных регистров (Рис. 2): регистр текущего адреса (CAR), регистр циклов ПДП (CWR), регистр хранения базового адреса (BAR), регистр хранения базового числа циклов ПДП (WCR) и 6-разрядный регистр режима (MR).

Регистр текущего адреса хранит текущий адрес ячейки памяти при выполнении цикла ПДП. После выполнения каждого цикла ПДП содержимое этого регистра увеличивается или уменьшается на единицу. Оно может быть Прочитано или загружено с помощью двух команд IN или соответственно OUT, работающих каждая только с одним байтом. В режиме авто инициализации содержимое регистра текущего адреса при обновлении принимает базовый адрес из регистра хранения базового адреса.

Регистр циклов ПДП хранит число слов, предназначенных для передачи. При выполнении циклов регистр работает в режиме вычитающего счетчика. При переходе из нулевого состояния в FFFFH вырабатывается управляющий сигнал для блока управления контроллером. Регистр может быть прочитан двумя командами IN. В него можно осуществить запись двумя командами OUT из микропроцессора или в режиме инициализации - из регистра хранения базового числа циклов ПДП.

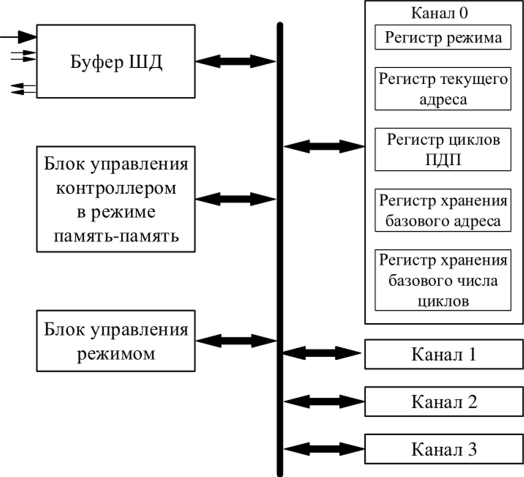


Рис. 2 Структурная схема контроллера ПДП

Регистры BAR и WCR предназначены для хранения констант - базового адреса и базового числа циклов. Они загружаются в режиме программирования КПДП одновременно с регистрами CAR и CWR. В процессе выполнения циклов ПДП их содержимое не изменяется. Прочитать состояние этих регистров невозможно.

Регистр режима определяет режим работы канала. Он содержит информацию о номере канала, типе цикла ПДП (чтение (ОП <— ВУ), запись (ОП—»ВУ), проверка), необходимости автоинициализации, режиме изменения регистра текущего адреса (CAR) - увеличение или уменьшение и режиме работы канала - передача по запросу, одиночная передача, блочная передача, каскадирование (работа в составе каскада КПДП).

Регистр команд блока управления режимом определяет основные параметры работы канала. Загружается при программировании КПДП микропроцессором.

Регистр условий хранит разрешение на ПДП каждому каналу (устанавливаемые программно) и запоминает факт перехода через 0 в регистре хранения базового числа циклов каждого канала.

Контроллер ПДП может работать в двух основных режимах:

1. В режиме программирования.

2. В режиме выполнения циклов ПДП.

В режиме программирования микропроцессор работает с КПДП, как с внешним устройством. После загрузки в КПДП управляющих слов контроллер переходит в пассивное состояние. В этом состоянии КПДП будет находиться до тех пор, пока не поступит запрос на ПДП от ВУ или от микропроцессора. Обнаружив запрос на ПДП, контроллер выставляет микропроцессору запрос на захват системной магистрали и ожидает от него подтверждения захвата (т.е. отключения МП от СМ, перехода его выходов в состояние высокого сопротивления, Z-состояния). При получении сигнала подтверждения захвата контроллер начинает выполнять циклы ПДП.

Системная магистраль (СМ) в режиме прямого доступа к памяти используется мультиплексно КПДП и микропроцессором, причем основное управление системной магистралью выполняет КПДП, а МП получает к ней доступ на очень короткие промежутки времени для обмена информацией с основной памятью.

Несмотря на большую самостоятельность ПДП, ведущее положение в МП-системе остается за микропроцессором, который, запустив обмен информацией в режиме ПДП, продолжает выполнять свою работу.

1. Дайте определение контроллер.

**Контроллер** — это электронное устройство, предназначенное для подключения к магистрали компьютера разных по принципу действия, интерфейсу и конструктивному исполнению периферийных устройств.

В современных организациях ЭВМ с магистральной и шинной организацией используются контроллеры.

Контроллер: в текущий момент времени выполняет одну команду ввода-вывода, получаемую от процессора или канала, и одновременно обслуживает одно внешнее устройство.

В его функции входит:

1. Опознание своего адреса выборки.
2. Подтверждение готовности внешних устройств.
3. Управление операцией во внешнем устройстве.
4. Согласование форматов данных.
5. Согласование скоростей передачи (буферизация).
6. Универсальные контролеры.
7. Контролеры IDE и SCSI.

Под термином **IDE** (Integrated Drive Electronics - электpоника, встpоенная в пpивод)), или ATA (AT Attachment - подключаемый к AT) понимается пpостой и недоpогой интеpфейс для PC. Все функции по упpавлению накопителем обеспечивает встpоенный контpоллеp. Пpостейший адаптеp IDE содеpжит только адpесный дешифpатоp. Адаптеpы IDE обычно не содеpжат собственного BIOS - все функции поддеpжки IDE встpоены в системный BIOS PC AT. Однако интеллектуальные или кэшиpующие контpоллеpы могут иметь собственный BIOS, подменяющий часть или все функции системного. Основной pежим pаботы устpойств IDE - пpогpаммный обмен (PIO) под упpавлением центpального пpоцессоpа.

Под термином **SCSI** - Small Computer System Interface (Интерфейс малых вычислительных систем) обычно понимается набор стандартов, разработанных Национальным институтом стандартов США (ANSI) и определяющих механизм реализации магистрали передачи данных между системной шиной компьютера и периферийными устройствами. Первоначально SCSI предназначался для использования в небольших дешевых системах и поэтому был ориентирован на достижение хороших результатов при низкой стоимости. Характерной его чертой является простота, особенно в части обеспечения гибкости конфигурирования периферийных устройств без изменения организации основного процессора. Главной особенностью подсистемы SCSI является размещение в периферийном оборудовании интеллектуального контроллера. Для достижения требуемого высокого уровня независимости от типов периферийных устройств в операционной системе основной машины, устройства SCSI представляются имеющими очень простую архитектуру. Например, геометрия дискового накопителя представляется в виде линейной последовательности одинаковых блоков, хотя в действительности любой диск имеет более сложную многомерную геометрию, содержащую поверхности, цилиндры, дорожки, характеристики плотности, таблицу дефектных блоков и множество других деталей. В этом случае само устройство или его контроллер несут ответственность за преобразование упрощенной SCSI модели в данные

для реального устройства.

1. Что такое шина?

**Шина** - это устройство, которое позволяет связать между собой несколько компонентов компьютера. Но к одной шине могут быть подключены несколько устройств и у каждой шины есть свой набор слотов для подключения кабелей или карт.

Фактически, шина - это набор электрических проводов, собранных в пучок, среди них есть провода питания, а также сигнальные провода для передачи данных. Шины также могут быть сделаны не в виде внешних проводов, а вмонтированы в схему материнской платы.

1. Системные и локальные шины.

Системная шина – совокупность линий передачи всех видов сигналов (в том числе данных, адресов и управления), идущих параллельно и имеющих одинаковое функциональное назначение, предназначенных для передачи информации между микропроцессором и остальными электронными устройствами компьютера

Некоторые компьютеры имеют единственную шину для памяти и устройств ввода/вывода. Такая шина часто называется **системной**.

**Локальной шиной** называется шина, электрически выходящая непосредственно на контакты микропроцессора. Она обычно объединяет процессор, память, схемы буферизации для системной шины и ее контроллер, а также некоторые вспомогательные схемы.

1. Транзакция шины.

Шинная транзакция включает в себя две части: посылку адреса и прием (или посылку) данных.

Шинные транзакции обычно определяются характером взаимодействия с памятью: транзакция типа "Чтение" передает данные из памяти (либо в ЦП, либо в устройство ввода/вывода), транзакция типа "Запись" записывает данные

в память. В транзакции типа "Чтение" по шине сначала посылается в память адрес вместе с соответствующими сигналами управления, индицирующими чтение. Память отвечает, возвращая на шину данные с соответствующими

сигналами управления. Транзакция типа "Запись" требует, чтобы ЦП или устройство в/в послало в память адрес и данные и не ожидает возврата данных. Обычно ЦП вынужден простаивать во время интервала между посылкой адреса

и получением данных при выполнении чтения, но часто он не ожидает завершения операции при записи данных в память

1. Основные локальные шины.

К основным типам локальных шин, используемых в ПК, относятся следующие.

* **VL-Bus** (локальная шина VESA)

Электрически выполнена в виде расширения локальной шины процессора - большинство входных и выходных сигналов процессора передаются непосредственно VLB-платам без промежуточной буферизации.

Из- за этого возрастает нагрузка на выходные каскады процессора, ухудшается качество сигналов на локальной шине и снижается надежность обмена по ней.

Поэтому VLB имеет жесткое ограничение на количество устанавливаемых устройств: при 33 MГц - три, 40 МГц - два, и при 50 МГц - одно, причем желательно - интегрированное в системную плату.

* **PCI**

РCI (Peripheral Component Interconnect - соединение внешних компонент) - развитие VLB. He совместима ни с какими другими, разрядность - 32/32 (расширенный вариант - 64/64), тактовая частота -до 33МГц, пропускная способность - до 132 Мб/с (264 Мб/сдля 32/32 на 66 МГц и 528 Мб/с для 64/64 на 66 МГц), поддержка автоконфигурации. Количество разъемов шины на одном сегменте ограничено четырьмя. Сегментов может быть несколько, они соединяются друг с другом посредством мостов (bridge). Сегменты могут объединяться в

различные топологии (дерево, звезда и т.п.). Разъем похож на MCA/VLB, но чуть длиннее (124 контакта). 64-разрядный разъем имеет дополнительную 64-контактную секцию с собственным ключом. Все разъемы и

карты к ним делятся на поддерживающие уровни сигналов 5В, 3.3 В и универсальные; первые два типа должны соответствовать друг другу, универсальные карты ставятся в любой разъем.

* **AGP**

ускоренный графический порт (AGP) это расширение шины PCI, чье назначение обработка больших массивов данных 3D графики. Если определить кратко, что такое AGP, то это - прямое соединение между графической подсистемой и системной памятью

1. Шина USB.

**Шина USB** предназначена для обеспечения обмена данными между компьютером (центральным процессором устройства) и подсоединенными к нему периферийными устройствами (ПУ) в условиях динамического (горячего)

изменения конфигурации системы.

Сама шина - это многоуровневая иерархическая система. На физическом уровне топология шины представляет собой корневидную структуру - многоуровневую звезду (в терминологии стандарта), при которой соединения

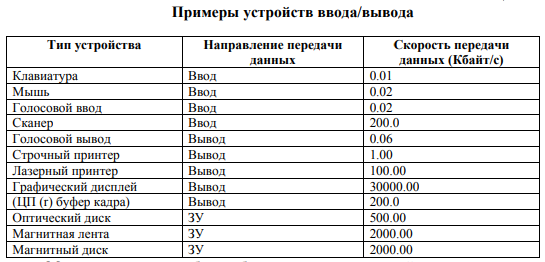
могут формировать цепочки и звезды. Закольцовка соединений в системе не допускается (этому, в частности, препятствует разная конструкция разъемов входного и выходного портов шины USB).

1. Основные особенности периферийных устройств.

Как правило, периферийные устройства компьютеров делятся на устройства ввода, устройства вывода и внешние запоминающие устройства (осуществляющие как ввод данных в машину, так и вывод данных из компьютера). Основной обобщающей характеристикой устройств ввода/вывода

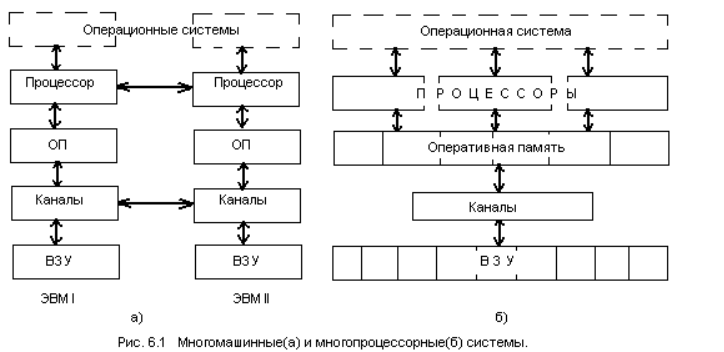
может служить скорость передачи данных (максимальная скорость, с которой данные могут передаваться между устройством ввода/вывода и основной памятью или процессором). В табл. представлены основные устройства

ввода/вывода, применяемые в современных компьютерах, а также указаны примерные скорости обмена данными, обеспечиваемые этими устройствами.



1. Каковы особенности работы мультипрограммных ЭВМ и ВС?
2. Поясните особенности матричных и конвейерных ВС.
3. Дайте сравнительную характеристику многомашинной и многопроцессорной ВС.

Различие понятий многомашинной и многопроцессорной ВС поясняет рис. Многомашинная ВС (ММС) содержит несколько ЭВМ, каждая из которых имеет свою ОП и работает под управлением своей операционной системы, а также средства обмена информацией между машинами. Реализация обмена информацией происходит, в конечном счете, путем взаимодействия операционных систем машин между собой. Это ухудшает динамические характеристики процессов межмашинного обмена данными. Применение многомашинных систем позволяет повысить надежность вычислительных комплексов. При отказе в одной машине обработку данных может продолжать другая машина комплекса. Однако можно заметить, что при этом оборудование комплекса недостаточно эффективно используется для этой цели. Достаточно в системе, изображенной на рис., а в каждой ЭВМ выйти из строя по одному устройству (даже разных типов), как вся система становится неработоспособной.



Этих недостатков лишены многопроцессорные системы (МПС). В таких системах (рис. 6.1,б) процессоры обретают статус рядовых агрегатов вычислительной системы, которые подобно другим агрегатам, таким, как модули памяти, каналы, периферийные устройства, включаются в состав системы в нужном количестве.

Вычислительная система называется многопроцессорной, если она содержит несколько процессоров, работающих с общей ОП (общее поле оперативной памяти) и управляется одной общей операционной системой. Часто в МПС организуется общее поле внешней памяти. МПС по сравнению с ММС достигается более быстрый обмен информацией между процессорами и поэтому может быть получена более высокая производительность, более быстрая реакция на ситуации, возникающие внутри системы и в ее внешней среде, и более высокие надежность и живучесть, так как система сохраняет работоспособность, пока работоспособны хотя бы по одному модулю каждого типа устройств.

Основные особенности построения многопроцессорных ВК заключаются в следующем:

система включает в себя один или несколько процессоров;

центральная память системы должна находиться в общем пользовании и к ней должен быть обеспечен доступ от всех процессоров системы;

система должна иметь общий доступ ко всем устройствам ввода-вывода, включая каналы; система должна иметь единую ОС, управляющую всеми аппаратными и программными средствами; в системе должно быть предусмотрено взаимодействие элементов аппаратного и программного обеспечения на всех уровнях: на уровне системного программного обеспечения, на программном уровне при решении задач пользователей (возможность перераспределения заданий), на уровне обмена данными и др. В многомашинных ВС связь может осуществляться только на информационном уровне.

1. Дайте классификацию ВС.

* **Однопроцессорные мультипрограммные ЭВМ**, высокая производительность которых была достигнута за счет распределения во времени основных устройств системы между программами.
* Дальнейшее повышение производительноти ЭВМ было достигнуто за счет мультиобработки программ (задач), т. е. за счет разбиения программ на отдельные блоки и параллельной обработки этих блоков на нескольких обрабатывающих устройствах, входящих в состав ВС. Первым типом ВС с мультиобработкой был многомашинный комплекс МК - **многомашинная ВС**. В состав МК объединялись различные ЭВМ с классической структурой, имеющие возможность обмениваться информацией.
* Следующим шагом в направлении дальнейшего увеличения производительности ВС явилось создание **многопроцессорных ВС** с мультиобработкой, в составе которых содержится два или несколько процессоров (ПР), работающих с единой ОП, общий набор каналов ввода-вывода (КВВ) и ВЗУ.
* Дальнейшее развитие идей мультиобработки привело к созданию крупных многопроцессорных систем высокой производительности, получивших назначение **высокопараллельных ВС**. Такие ВС в зависимости от ее структуры могут одновременно обрабатывать множественный поток данных или команд. Под потоком команд понимается последовательность команд, выполняемых ВС, а потоком данных - последовательность данных, обрабатываемых под управлением потока команд.

Многопроцессорные и многомашинные ВС классифицируют по различным признакам:

* По назначению ВС делятся на универсальные и специализированные. **Универсальные** ВС предназначены для решения широкого круга задач, специализированные — для решения определенного круга задач. **Специализированные** ВС, как правило, должны иметь аппаратурные и программные средства, предназначенные специально для этой системы.
* По типу оборудования ВС подразделяются на однородные и неоднородные. **Однородные** системы содержат несколько однотипных ЭВМ (или процессоров), **неоднородные** - разнотипные ЭВМ (или процессоры). Основной недостаток однородных ВС - неполная загруженность отдельных ЭВМ (процессоров) во время ее работы. В целях повышения эффективности использования ЭВМ (процессоров) используются неоднородные ВС.
* По типу структуры ВС разделяются с постоянной и переменной структурами. Под структурой ВС понимают состав системы и схемы функциональных и управляющих связей между ее элементами. В системах **с постоянной структурой** в процессе ее функционирования не изменяется состав функциональных и управляющих связей между ее элементами. **Переменную структуру** имеют адаптивные системы, т. е. такие, у которых структура изменяется на основе анализа текущей информации. Подобные системы позволяют достичь оптимального состояния в любых изменяющихся условиях функционирования.
* По степени централизации управления ВС разделяются на централизованные, децентрализованные и со смешанным управлением.

В **централизованных** ВС все функции управления сосредоточены в одном элементе, в качестве которого используется одна из ЭВМ, называемая машиной-директором, или центральный процессор. В **децентрализованной** ВС каждый процессор или ЭВМ действуют автономно, решая свои задачи. В системе **со смешанным управлением** ВС разбивается на группы взаимодействующих ЭВМ (или процессоров), в каждой из которых осуществляется централизированное управление, а между группами -

децентрализированное.

1. В каких режимах с целью повышения надежности могут работать двухмашинные ВС?

В настоящее время наиболее широко используют двухмашинные вычислительные комплексы, которые могут работать в одном из следующих режимов.

1. **100% - ное горячее резервирование.** Обе ЭМВ в этом режиме исправны и работают параллельно, выполняя одни те же операпни над одной и той же информацей (дуплексный режим). После выполнения каждой команды результаты преобразования сравниваются и при их совпадении процесс

вычислений продолжается. При этом в памяти обоих ЭВМ в каждый момент находится одна и та же информация. При обнаружении несовпадения в результатах обработки Неисправная ЭВМ выводится на ремонт, а исправная

ЭВМ продолжает работать под контролем встроенной в ЭВМ системы автоматического контроля.

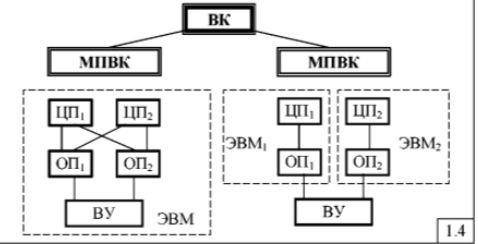
2. **0дна исправная ЭВМ** решает задачи без дублирования, а **другая ЭВМ находится в режиме «Профилактика»,** в котором осуществляется прогон контролирующих тестов. Если основная ЭВМ продолжает не в состоянии выполнить задачу, то резервная может прекратить "Профилактику" и начать работу параллельно с основной.

3. **Обе ЭВМ работают в автономном режиме** со своим набором ПфУ по автономным рабочим программам.

1. Поясните связи между ЭВМ в ВК.

ЭВМ (электронная вычислительная машина, компьютер) -совокупность технических средств, предназначенных для организации ввода, хранения, автоматической обработки по заданной программе и вывода данных (информации).

К техническим средствам относятся (рис. 1.3):



• центральный процессор (ЦП);

• оперативная (основная) память (ОП);

• внешние устройства (ВУ), включающие устройства ввода-вывода (УВВ) и внешние запоминающие устройства (ВЗУ);

• процессоры (каналы) ввода-вывода (ПВВ, КВВ).

Основной целью построения ВК является обеспечение высокой надежности и/или производительности, не достижимой для однопроцессорных ЭВМ.

Вычислительная система (ВС) - совокупность технических и программных средств, ориентированных на решение определенно60й совокупности задач

Вычислительный комплекс (ВК) - совокупность технических средств, содержащая несколько центральных процессоров и представляющая собой одну ЭВМ с несколькими ЦП (МПВК -многопроцессорный ВК)или объединение нескольких однопро­цессорных ЭВМ (ММВК -многомашинный ВК)

1. Какие технические средства связи используются в многомашинных ВК.

В многопроцессорных и многомашинных вычислительных системах используются в основном параллельные интерфейсы для сопряжения отдельных устройств в ЭВМ, и только в отдельных случаях применяются последовательные интерфейсы для подключения периферийных устройств. Параллельные интерфейсы обеспечивают в первую очередь передачу сигналов прерывания, а также отдельных слов (команд) и блоков данных между сопрягаемыми ЭВМ и устройствами.

1. Конвейеризация и параллельные вычисления.

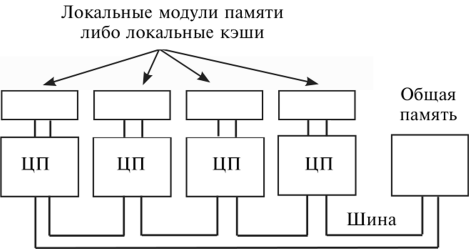
При параллелизме совмещение операций достигается путем воспроизведения в нескольких копиях аппаратной структуры. Высокая производительность достигается за счет одновременной работы всех элементов структур, осуществляющих решение различных частей задачи. Конвейеризация (или конвейерная обработка) в общем случае основана на разделении подлежащей исполнению функции на

более мелкие части, называемые ступенями, и выделении для каждой из них отдельного блока аппаратуры. Так обработку любой машинной команды можно разделить на несколько этапов (несколько ступеней), организовав передачу данных от одного этапа к следующему. При этом конвейерную обработку можно использовать для совмещения этапов выполнения разных команд. Производительность при этом возрастает благодаря тому, что одновременно на различных ступенях конвейера выполняются несколько команд. Конвейерная обработка такого рода широко применяется во всех современных быстродействующих процессорах.

1. Альтернативные виды архитектур ЭВМ. Потоковая архитектура.

Архитектура потоковых ЭВМ. Порядок выполнения команд определяется наличием исходной информации для выполнения каждой из них. Если несколько команд готовы к выполнению, то принципиально возможно их назначение для выполнения таким же количеством свободных процессоров. Говорят, что такие вычислительные системы управляются потоком данных (data flow). Порядок выполнения команд определяется наличием исходной информации для выполнения каждой из них.

Система из нескольких параллельных процессоров, имеющих общую память, называется мультипроцессором. Поскольку каждый процессор может записывать информацию в любую часть памяти и считывать информацию из любой части памяти, их работа должна согласовываться программным обеспечением. Компьютеры data flow имеют архитектуру, показанную на рис.

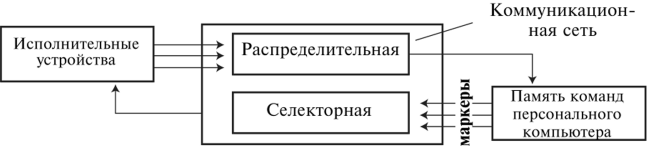


В таком компьютере каждый процессор кроме доступа к общей памяти по общей шине имеет собственную локальную память, недоступную для других процессоров. Эта память используется для тех программ и данных, которые не нужно разделять между несколькими процессорами. При доступе к локальной памяти основная шина не используется и, таким образом, объем передаваемой по ней информации становится меньше.

Программа или ее часть (сегмент) размещается в памяти команд ПК, состоящей из ячеек команд. Команды имеют структуру:

{код операции, операнд 1,операнд L, адрес результата 1, адрес результата М}

В командах проверки условия возможно альтернативное задание адреса результата, поэтому L < М. Адреса результатов являются адресами ПК, т.е. результаты выполнения одних команд в качестве операндов могут поступать в текст других команд. Команда не готова к выполнению, если в ее тексте отсутствует хотя бы один операнд. Ячейка, обладающая полным набором операндов, переходит в возбужденное состояние и передает в селекторную сеть информационный пакет (маркер), содержащий код операции и необходимую числовую и связную информацию. Он поступает по коммуникационной сети на одно из исполнительных устройств. Там же операция выполняется, и в распределительную сеть выдается результирующий пакет, содержащий результат вычислений и адреса назначения в ПК (возможно, за счет выбора альтернативы, т.е. такой выбор — тоже результат). Селекторная и распределительная сети образуют общую коммуникационную сеть вычислительных систем (рис.)

****

Сверхвысокая производительность архитектуры обеспечивается за счет одновременной и независимой активизации большого числа готовых команд, каждая из которых выполняется на отдельном процессоре.

1. Основные характеристики и классификация процессоров.

Двумя основными архитектурами набора команд, используемыми компьютерной промышленностью на современном этапе развития вычислительной техники являются архитектуры CISC и RISC. Основоположником CISC-архитектуры можно считать компанию IBM с ее базовой архитектурой /360, ядро которой используется с 1964 года и дошло до наших дней.

CISC - Complete Instruction Set Computer

RISC - Reduced Instruction Set Computer

**Для CISC-процессоров характерно**: сравнительно

небольшое число регистров общего назначения; большое количество машинных команд, некоторые из которых нагружены семантически аналогично операторам высокоуровневых языков программирования и выполняются за много тактов; большое количество методов адресации; большое количество форматов команд различной разрядности; преобладание двухадресного формата команд; наличие команд обработки типа регистр-память.

Основой архитектуры современных рабочих станций и серверов является архитектура компьютера с сокращенным набором команд (**RISC – Reduced Instruction Set Computer**). Для них характерно:

* Ограниченный набор простых команд, выполняемых за один такт
* Использование команд «регистр-регистр»
* Доступ к памяти с помощью команд загрузки и выгрузки регистров
* Аппаратное управление
* Фиксированный формат и длина команд
* Огрраниченное количество методов адресации
* Конвейерное выполнение команд
* Наличие большого числа регистров общего назначения

Еще процессоры можно классифицировать по двум параметрам: разрядности и быстродействию. Быстродействие процессора — довольно простой параметр. Оно измеряется в мегагерцах (МГц); 1 МГц равен миллиону тактов в секунду. Чем выше быстродействие, тем лучше (тем быстрее процессор). Разрядность процессора — параметр более сложный. В процессор входит три важных устройства, основной характеристикой которых является разрядность:

* шина ввода и вывода данных;
* внутренние регистры;
* шина адреса памяти.

Процессоры с тактовой частотой менее 16 МГц не имеют встроенной кэш-памяти. В системах до 486-го процессора быстрая кэш-память устанавливалась на системную плату. Начиная с процессоров 486, кэш-память первого уровня устанавливалась непосредственно в корпусе и работала на частоте процессора. А кэш-память на системной плате стали называть кэш-памятью второго уровня. Она работала уже на частотах, поддерживаемых системной платой.

1. Система команд и данных процессора.

В общем случае система команд процессора включает в себя следующие четыре основные группы команд:

команды пересылки данных;

* арифметические команды ;
* логические команды ;
* команды переходов.

Команды пересылки данных не требуют выполнения никаких операций над операндами. Операнды просто пересылаются (точнее, копируются) из источника (Source) в приемник (Destination). Источником и приемником могут быть внутренние регистры процессора, ячейки памяти или устройства ввода/вывода. АЛУ в данном случае не используется.

Арифметические команды выполняют операции сложения, вычитания, умножения, деления, увеличения на единицу (инкрементирования), уменьшения на единицу (декрементирования) и т.д. Этим командам требуется один или два входных операнда. Формируют команды один выходной операнд.

Логические команды производят над операндами логические операции, например, логическое И, логическое ИЛИ, исключающее ИЛИ, очистку, инверсию, разнообразные сдвиги (вправо, влево, арифметический сдвиг, циклический сдвиг). Этим командам, как и арифметическим, требуется один или два входных операнда, и формируют они один выходной операнд.

Наконец, команды переходов предназначены для изменения обычного порядка последовательного выполнения команд. С их помощью организуются переходы на подпрограммы и возвраты из них, всевозможные циклы, ветвления программ, пропуски фрагментов программ и т.д. Команды переходов всегда меняют содержимое счетчика команд. Переходы могут быть условными и безусловными. Именно эти команды позволяют строить сложные алгоритмы обработки информации.

В соответствии с результатом каждой выполненной команды устанавливаются или очищаются биты регистра состояния процессора ( PSW ). Но надо помнить, что не все команды изменяют все имеющиеся в PSW флаги. Это определяется особенностями каждого конкретного процессора.

У разных процессоров системы команд существенно различаются, но в основе своей они очень похожи. Количество команд у процессоров также различно. У современных мощных процессоров количество команд достигает нескольких сотен. В то же время существуют процессоры с сокращенным набором команд (так называемые RISC-процессоры), в которых за счет максимального сокращения количества команд достигается увеличение эффективности и скорости их выполнения.

Команды пересылки данных занимают очень важное место в системе команд любого процессора. Они выполняют следующие важнейшие функции:

* загрузка (запись) содержимого во внутренние регистры процессора;
* сохранение в памяти содержимого внутренних регистров процессора;
* копирование содержимого из одной области памяти в другую;
* запись в устройства ввода/вывода и чтение из устройств ввода/вывода.

В некоторых процессорах (например, Т-11) все эти функции выполняются одной единственной командой MOV (для байтовых пересылок — MOVB ) но с различными методами адресации операндов.

В других процессорах помимо команды MOV имеется еще несколько команд для выполнения перечисленных функций. Например, для загрузки регистров могут использоваться команды загрузки, причем для разных регистров — разные команды (их обозначения обычно строятся с использованием слова LOAD — загрузка). Часто выделяются специальные команды для сохранения в стеке и для извлечения из стека ( PUSH — сохранить в стеке, POP — извлечь из стека). Эти команды выполняют пересылку с автоинкрементной и с автодекрементной адресацией (даже если эти режимы адресации не предусмотрены в процессоре в явном виде).

Иногда в систему команд вводится специальная команда MOVS для строчной (или цепочечной) пересылки данных (например, в процессоре 8086). Эта команда пересылает не одно слово или байт, а заданное количество слов или байтов ( MOVSB ), то есть инициирует не один цикл обмена по магистрали, а несколько. При этом адрес памяти, с которым происходит взаимодействие, увеличивается на 1 или на 2 после каждого обращения или же уменьшается на 1 или на 2 после каждого обращения. То есть в неявном виде применяется автоинкрементная или автодекрементная адресация.

В некоторых процессорах (например, в процессоре 8086) специально выделяются функции обмена с устройствами ввода/вывода. Команда IN используется для ввода (чтения) информации из устройства ввода/вывода, а команда OUT используется для вывода (записи) в устройство ввода/вывода. Обмен информацией в этом случае производится между регистром-аккумулятором и устройством ввода/вывода. В более продвинутых процессорах этого же семейства (начиная с процессора 80286) добавлены команды строчного (цепочечного) ввода (команда INS ) и строчного вывода (команда OUTS ). Эти команды позволяют пересылать целый массив (строку) данных из памяти в устройство ввода/вывода ( OUTS ) или из устройства ввода/вывода в память ( INS ). Адрес памяти после каждого обращения увеличивается или уменьшается (как и в случае с командой MOVS ).

Также к командам пересылки данных относятся команды обмена информацией (их обозначение строится на основе слова Exchange ). Может быть предусмотрен обмен информацией между внутренними регистрами, между двумя половинами одного регистра ( SWAP ) или между регистром и ячейкой памяти.

1. Способы адресации данных.

Для расширения операционных возможностей и увеличения производительности процессора применяются различные способы адресации информации, отличающиеся порядком использования и обработки адресного поля в команде, посредством которого организуется доступ к информации, хранящейся в оперативной памяти (или в ПЗУ) ЭВМ. Рассмотрим способы адресации операндов и команд.

**Непосредственная адресация**

В коде команды (в коде одного или нескольких адресов) размещается непосредственный операнд, если число значащих цифр операнда не превышает длины адресной части команды. Такая адресация используется для хранения различного рода констант и находит широкое применение в универсальных ЭВМ в целях экономии ячеек ОП и уменьшения времени выполнения команды.

**Прямая адресация**

Исполнительный адрес - адрес ячейки ОП, в которой хранится адресуемое слово, совпадает с адресной частью команды. Этот метод используется в ЭВМ в комбинации с другими методами адресации.

**Прямая регистровая адресация**

В адресном поле команды содержится адрес R регистра СОЗУ процессора, в котором хранится операнд:

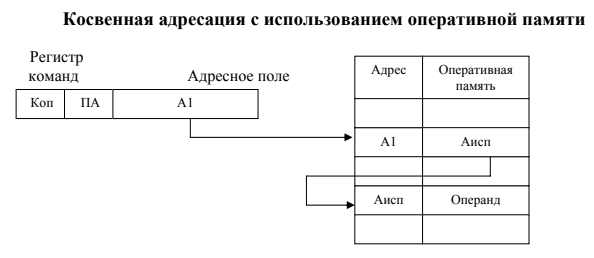
При таком способе адресации быстродействие ЭВМ повышается, так как нет необходимости извлекать операнды из ОП и команда имеет более короткий формат, так как регистров в СОЗУ обычно значительно меньше, чем ячеек в оперативной памяти. Прямая регистровая адресация используется для операндов, многократно используемых при выполнении программ.

**Подразумеваемая адресация**

При такой адресации в команде не содержится явных указаний об адресе операнда участвующего в операции, или адреса, по которому передается результат операции. Этот адрес подразумевается и фактически задается кодом операции. Например, в одноадресных командах адресом второго операнда или результата операции подразумевается адрес — содержимое специального регистра процессора, хранящего второй операнд или принимающего результат операции; в двухадресных командах подразумевается помещение операции по адресу одного из операндов.

**Косвенная адресация**

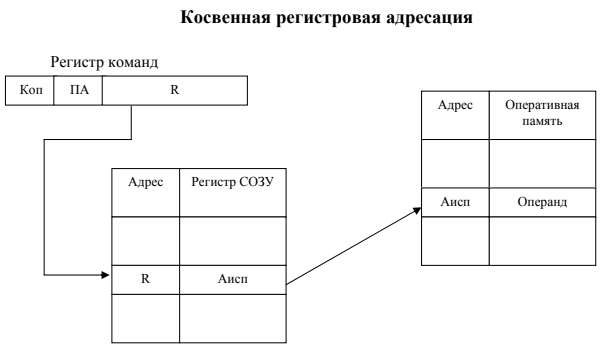
В адресном поле команды указывается адрес ячейки оперативной памяти, содержащей другой адрес, который может быть исполнительным Аисп или еще одним косвенным адресом (так называемая многоступенчатая косвенная адресация). Таким образом, косвенная адресация (рис.) может быть определена как «адресация адреса»:



Она используется в тех случаях, когда число разрядов в адресной части команды недостаточно для указания всех адресов оперативной памяти ЭВМ.

**Косвенная регистровая адресация**

При таком способе адресации в адресном поле команды число является адресом R регистра Рсозу, который содержит исполнительный адрес Аисп (рис.):



При такой адресации необходимо сначала загрузить регистр R, а поэтому ее используют тогда, когда программа многократно использует один и тот же адрес ячейки.

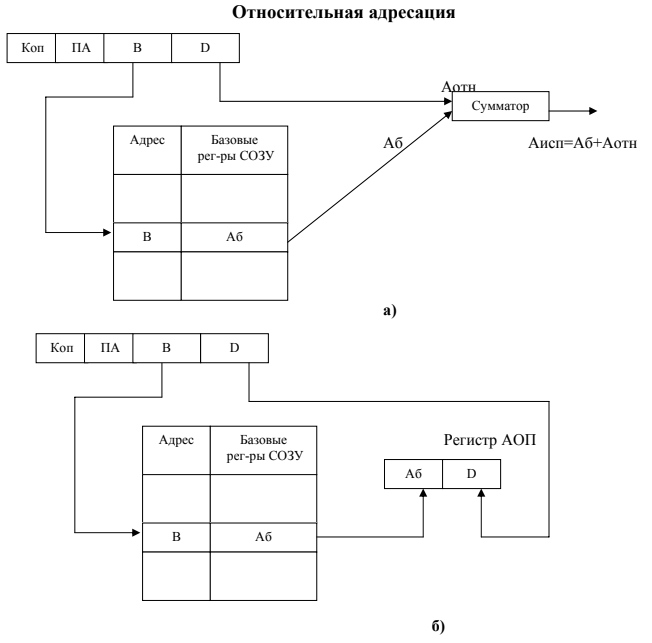
**Модификация адресов**

Рассмотренные методы адресации обеспечивают адресацию переменных и констант. При решении ряда задач на ЭВМ необходимо выполнять некоторые участки программ многократно (цикличность вычислительного процесса) над различными операндами, расположенными упорядоченно в массивах ОП. Поскольку операнды, обрабатываемые при повторениях цикла, имеют разные адреса, то каждый цикл в программе можно представить виде последовательности команд, отличающихся адресными частями. Однако при таком подходе программа решения задачи оказывается слишком длинной и ее составление чрезмерно трудоемким. Программирование вычислительных циклов существенно упрощается, если после каждого цикла обеспечить автоматическое изменение в соответствующих командах их адресных частей. Процедура изменения адреса в командах называется модификацией адреса. Модификация адресов команд основана на возможности выполнения над кодами команд или их частями арифметических и логических операций. В качестве операндов в командах вычислительного цикла могут фигурировать элементы массивов называемые переменными с индексами. Элемент массива представляется базовым адресом Аб и индекса i, указывающего, на сколько единиц должен быть изменен адрес команды перед ее выполнением. Программный способ модификации адресов в команде значительно замедляет процесс обработки переменных с индексом и требует для этих целен большой емкости оперативной памяти. В связи с этим в современных ЭВМ для модификации адресов используют аппаратные средства. В этом случае адрес в команде (рис. 6.5) представляется двумя полями. В поле В указывается базовый адрес массива Аб оперативной памяти. Поле Х называется индексом. Если Х=0, то адрес Аб не модифицируется, т.е. является исполнительным Аисп. Значение Х<>0 определяет адрес ячейки памяти индексов, в которой хранится индекс i. Модификация адреса сводится к вычислению исполнительного адреса Аисп=Аб+(Х), где (X) — содержимое ячейки Х индексной памяти. В качестве индексной памяти используют в процессоре так называемые индексные регистры СОЗУ. Суммирование производится или АЛБ процессора, пли в специальном сумматоре обработки адресов, что несколько только увеличивает объем процессора.



**Относительная адресация**

При динамическом распределении памяти базовые адреса массивов изменяются в процессе выполнения программы, в результате адреса не могут быть зафиксированы в программе. Для обеспечения динамического распределения памяти используют способ относительной адресации. Относительный адрес (рис. 6.6, а) состоит из двух полей: В, указывающего базовый адрес Аб массива D, представляющего собой относительный адрес Аотн. Поле D принято называть смещением D операнда относительно начала массива. Исполнительный адрес вычисляется по формуле Аисп= (В)+D. Для хранения базовых адресов в целях повышения быстродействия ЭВМ используют также так называемые базовые регистры СОЗУ. При формировании Аисп (рис. а) на суммирование в SМ затрачивается некоторое время. В целях уменьшения этого времени используют так называемый метод совмещения. В этом случае в базовом регистре В (рис. б) содержатся старшие разряды, а в поле С) записывают младшие разряды исполнительного адреса Аисп, которые выдаются непосредственно в регистр адреса оперативной памяти (РгАОП). При совмещении, очевидно, базовый адрес Аб не может принимать значение адреса любой ячейки ОП, а только тех адресов, которые содержат в младших разрядах нули, соответствующие количеству разрядов поля D.



В универсальных ЭВМ используют совместно относительную адресацию и модификацию адресов (рис. 6.7). В этом случае Аисп вычисляется по формуле Аисп=(В)+(X)+D, где (В) — базовый адрес Аб (содержимое ячейки В), (X) — индекс i (содержимое ячейки X); D — смещение операнда (относительный адрес).



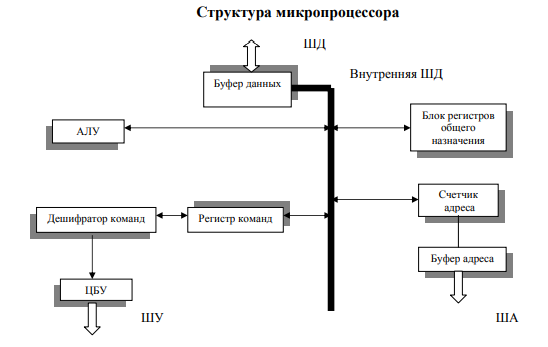
1. Структурная организация процессора.

Процессор — центральная часть ЭВМ, организующая ее работу по заданной программе. Процессор объединяет в себе АЛУ и ЦУУ, с помощью которых рабочая программа интерпретируется в вычислительный процесс.

Структура процессора зависит от принятой в ЭВМ системы счисления, формата данных и команд, системы команд, способов адресации и организации вычислительного процесса и принципа управления им, а также метода контроля и диагностики работы ЭВМ. Арифметическо-логическое устройство (АЛУ) — совокупность блоков и

узлов процессора, обеспечивающая выполнение арифметических и логических операций над операндами. Характер операции задается командой программы.

Центральное устройство управления (ЦУУ) - совокупность блоков и узлов процессора, обеспечивающая координирование работы всех устройств ЭВМ и управление ими для всех принятых в данной ЭВМ режимов работы.



Cache (запас) - обозначает быстродействующую буферную память между процессором и основной памятью (буфер данных, буфер адреса). Кэш служит для частичной компенсации разницы в скорости процессора и основной памяти - туда попадают наиболее часто используемые данные. Когда процессор первый раз обращается к ячейки памяти, ее содержимое параллельно копируется в кэш, и в случае повторного обращения в скором времени может быть с гораздо большей скоростью выбрано из кэша. При записи в память значение попадает в кэш, и либо одновременно копируется в память (схема Write Through - прямая

или сквозная запись), либо копируется через некоторое время (схема Write Back - отложенная или обратная запись). При обратной записи, называемой также буферизованной сквозной записью, значение копируется в память в первом же свободном такте, а при отложенной (Delayed Write) - когда для помещения в кэш нового значения не оказывается свободной области; при этом в память вытесняются наименее используемая область кэша. Вторая схема более эффективна, но и более сложна за счет необходимости поддержания

соответствия содержимого кэша и основной памяти.

1. Устройство управления. Жесткое и микропрограммное управление.

УСТРОЙСТВО УПРАВЛЕНИЯ ЭВМ - координирует совместную работу процессора, внешней памяти, устройств ввода-вывода и др. посредством управляющих сигналов, вырабатываемых устройством управления в соответствии с реализуемой программой.

Микропрограммное управление, вид иерархического управления работой цифровых вычислительных машин, при котором каждая команда является обращением к последовательности т. н. микрокоманд, обычно более низкого уровня, чем сама команда. Набор микрокоманд называется микропрограммой и обычно хранится в постоянной памяти ЦВМ, составляющей неотъемлемую часть устройства управления. Записанные в памяти микрокоманды определяют работу всех устройств машины, выбирая в каждом такте нужные совокупности элементарных машинных операций, а последовательность микрокоманд обеспечивает выполнение заданной команды. Микрокоманда может содержать три части: оперативную, в которой указываются управляющие входы всех исполнительных устройств машины; адресную, определяющую адрес следующей микрокоманды с учётом условий логических переходов (передач управления); временную, определяющую время выполнения микрокоманды. При этом код конкретной операции программы совпадает с адресом первой микрокоманды соответствующей микропрограммы.

Достоинства Микропрограммное управление состоят в том, что оно обеспечивает операционную гибкость ЦВМ и возможность изменения системы команд и состава машинных операций в зависимости от особенностей решаемых задач и условий применения машины; позволяет сравнительно престо реализовать различные сложные операции при значительной экономии машинного времени; даёт возможность строить диагностические микротесты для определения с большой точностью места неисправности в машине. Основной недостаток, обусловливающий ограниченное распространение Микропрограммное управление, — необходимость применения быстродействующих запоминающих устройств небольшого объёма (несколько тыс. слов) с временем обращения, соизмеримым с временем выполнения элементарных операций в исполнительных устройствах. В вычислительных машинах 3-го поколения широко используется также метод управления, при котором микропрограмма реализуется с помощью системы устройств, а не в виде команд, записанных в памяти ЭВМ; высокое быстродействие, большие объёмы оперативной памяти и богатое математическое обеспечение этих машин позволяют сделать управление более эффективным, чем при Микропрограммное управление в ЦВМ 2-го поколения.

Управляющий автомат, назначением которого является реализация микропрограммы (т. е. управление ее выполнением), называется микропрограммным автоматом (МПА).

Отличие МПА с “жесткой” логикой от МПА с программируемой логикой состоит в том, что в МПА с “жесткой” логикой закон функционирования определяется способом соединения логических элементов, в то время как в МПА с программируемой логикой он задается программой, хранимой в ячейках памяти, называемых управляющей памятью, либо памятью МК.

1. Особенности RISC и СISC –процессоров.

**Для CISC-процессоров характерно**: сравнительнонебольшое число регистров общего назначения; большое количество машинных команд, некоторые из которых нагружены семантически аналогично операторам высокоуровневых языков программирования и выполняются за много тактов; большое количество методов адресации; большое количество форматов команд различной разрядности; преобладание двухадресного формата команд; наличие команд обработки типа регистр-память.

Основой архитектуры современных рабочих станций и серверов является архитектура компьютера с сокращенным набором команд (**RISC – Reduced Instruction Set Computer**). Для них характерно:

* Ограниченный набор простых команд, выполняемых за один такт
* Использование команд «регистр-регистр»
* Доступ к памяти с помощью команд загрузки и выгрузки регистров
* Аппаратное управление
* Фиксированный формат и длина команд
* Огрраниченное количество методов адресации
* Конвейерное выполнение команд
* Наличие большого числа регистров общего назначения

1. Суперскалярная архитектура процессора.

Суперскалярным называется центральный процессор (ЦП), который одновременно выполняет более чем одну скалярную команду. Это достигается за счет включения в состав ЦП нескольких самостоятельных функциональных (исполнительных) блоков, каждый из которых отвечает за свой класс операций и может присутствовать в процессоре в нескольких экземплярах. Процессор включает в себя шесть блоков: выборки команд, декодирования команд, диспетчеризации команд, распределения команд по функциональным блокам, блок исполнения и блок обновления состояния

1. Регистры общего назначения процессора. Регистры управления процессором.

Регистры общего назначения предназначены для хранения операндов арифметико-логических инструкций, а также адресов или отдельных компонентов адресов ячеек памяти.

Регистром управления является объединенный 32-битный регистр инструкций и 32-битный регистр флагов (регистр процессора, отражающий его текущее состояние). Многие инструкции включают в себя операции сравнения и математические вычисления, которые способны изменить состояния флагов, а некоторые другие условные инструкции проверяют значения флагов состояния, чтобы перенести поток управления в другое место.

1. Назначение и функционирование математического сопроцессора.

Математический сопроцессор – устройство для обработки числовых данных с плавающей точкой. Называется сопроцессором из-за того, что предназначен для расширения вычислительных возможностей основного процессора (сам по себе полноценным процессором не является) и изначально размещался на отдельной плате (сейчас уже сопроцессор встраивают в основной). Процессор и сопроцессор имеют свои раздельные системы команд и форматы обрабатываемых данных. Несмотря на то, что сопроцессор архитектурно представляет собой отдельное вычислительное устройство, он не может существовать отдельно от основного процессора. Процессор и сопроцессор, являясь двумя самостоятельными вычислительными устройствами, могут работать параллельно. Но это распараллеливание распространяется только на выполнение команд. Оба процессора подключены к общей системной шине и имеют доступ к одной и той же информации. Инициирует процесс выборки очередной команды всегда основной процессор. После выборки команда попадает одновременно в оба процессора. Любая команда сопроцессора имеет код операции: первые пять бит имеют значение 11011. Сопроцессор, определив по первым пяти битам, что очередная команда принадлежит его системе команд, начинает ее исполнение. Если команда требует операнды из памяти, то сопроцессор обращается к шине данных за чтением содержимого ячейки памяти, которое к этому моменту предоставляется основным процессором.

1. Регистры и стек сопроцессора.

В программной модели сопроцессора выделяют 3 группы регистров:

1. **Восемь регистров r0…r7**, которые составляют основу программной модели – его стек. Размер каждого регистра – 80 бит
2. **Три служебных регистра**

* Регистр состояния сопроцессора SWR (Status word register) - отражает информацию о текущем состоянии сопроцессора
* Управляющий регистр сопроцессора CWR (Control word register) - управляет режимами работы сопроцессора
* Регистр тегов TWR (Tags word register — слово тегов) — используется для контроля за состоянием каждого из регистров стека

1. **Два регистра указателей**

* Указатель данных DPR (Data Point Register)

· Указатель команд IPR (Instruction Point Register)

Они предназначены для запоминания информации об адресе команды, вызвавшей исключительную ситуацию и адресе ее операнда. Эти указатели используются при обработке исключительных ситуаций (но не для всех команд).

1. Cистема команд и данных сопроцессора. Управление сопроцессором.

Система команд сопроцессора включает около 80 машинных команд. Их можно разделить на следующие группы:

* **Команды передачи данных** (предназначены для организации обмена между регистрами стека, вершиной стека сопроцессора и ячейками оперативной памяти)
* **Команды сравнения данных** (выполняют сравнение значений числа в вершине стека и операнда, указанного в команде)
* **Арифметические команды** (реализуют четыре основные арифметические операции — сложение, вычитание, умножение и деление)
* **Команды трансцендентных функций** (предназначены для вычисления тригонометрических, логарифмических и показательных функций)
* **Команды управления сопроцессором** (предназначены для общего управления работой сопроцессора типа очистка регистра, синхронизация с центральным процессором, считать слово управления/состояния в память)

Минимальная длина команды сопроцессора — 2 байта. Все команды сопроцессора оперируют регистрами стека сопроцессора. Команды работают с целыми двоичными, целыми десятичными и вещественными числами.

1. Векторные команды. Особенности ММХ-технологии.

**Вектором** называют совокупность данных (массив, матрицу, группу), расположенных в ячейках памяти с упорядоченными адресами. При векторной обработке данных с помощью одной команды выполняется операция над всеми элементами вектора. Для реализации векторной обработки в состав процессора вводят дополнительные средства (векторные регистры, векторные команды и т.д.).

(Как я понял, векторы используются при работе с графикой и всякими мультимедиа, тут-то и связь с MMX, которая на такие данные и завязана)

Технология MMX разработана **для ускорения мультимедиа и коммуникационных программ**. Она включает в себя новые команды и типы данных, что позволяет создавать приложения нового уровня. Технология основана на параллельной обработке данных. При этом сохраняется полная совместимость с существующими операционными системами и программным обеспечением. ММХ-технологии поддерживает новую арифметику, называемую арифметикой с насыщением (Saturation arithmetic).

В основе ММХ лежит принцип SIMD (Single Instruction Multiple Data), т.е. одной командой можно обработать сразу несколько единиц информации.

(Инфа выше из методички, инфа ниже из интернета (там более понятно написано на мой взгляд))

MMX представляет собой 57 дополнительных команд и восемь 64-битных регистров, позволяющих процессору за одну машинную операцию обрабатывать 64-битное бинарное слово, которое может включать сразу несколько более мелких однотипных слов. За счет этого в программах, разработанных с учетом поддержки MMX, процессору для обработки большого массива однотипных данных (которыми, по сути, и являются мультимедийные данные) требуется значительно меньше времени.

Например, при работе с цифровым изображением с глубиной цвета 16 бит процессор с MMX может одновременно обрабатывать сразу 4 пиксела, а не один.

Недостатком MMX является то, что она может быть использована процессором только для обработки целых чисел. Кроме того, **MMX имеет общие с сопроцессором регистры, так что одновременно задействовать инструкции MMX и инструкции сопроцессора процессор не может.**

1. Иерархическая структура памяти ЭВМ. Принцип кэширования.



Иерархия памяти ЭВМ

По мере продвижения по структуре сверху вниз возрастают 2 параметра:

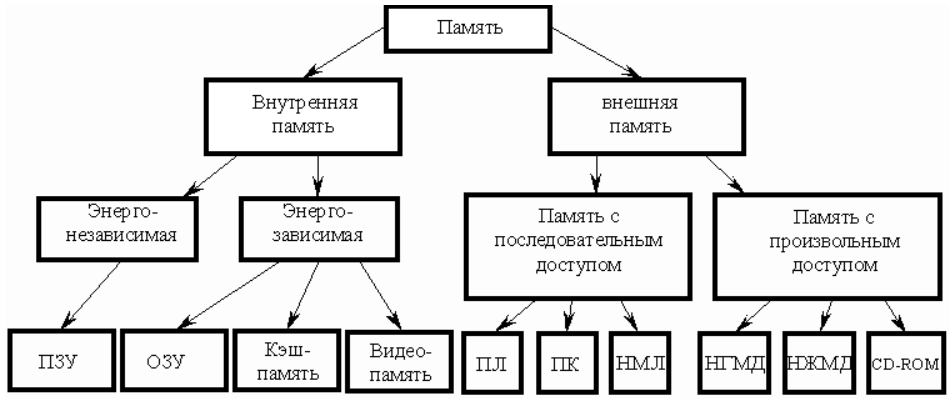
Во-первых, **увеличивается время доступа**. Доступ к регистрам занимает несколько наносекунд, доступ к кэш-памяти — немного больше, доступ к основной памяти — несколько десятков наносекунд. Дальше идет большой разрыв: доступ к дискам занимает по крайней мере 10 мкс, а время доступа к внешним накопителям вообще может измеряться в секундах

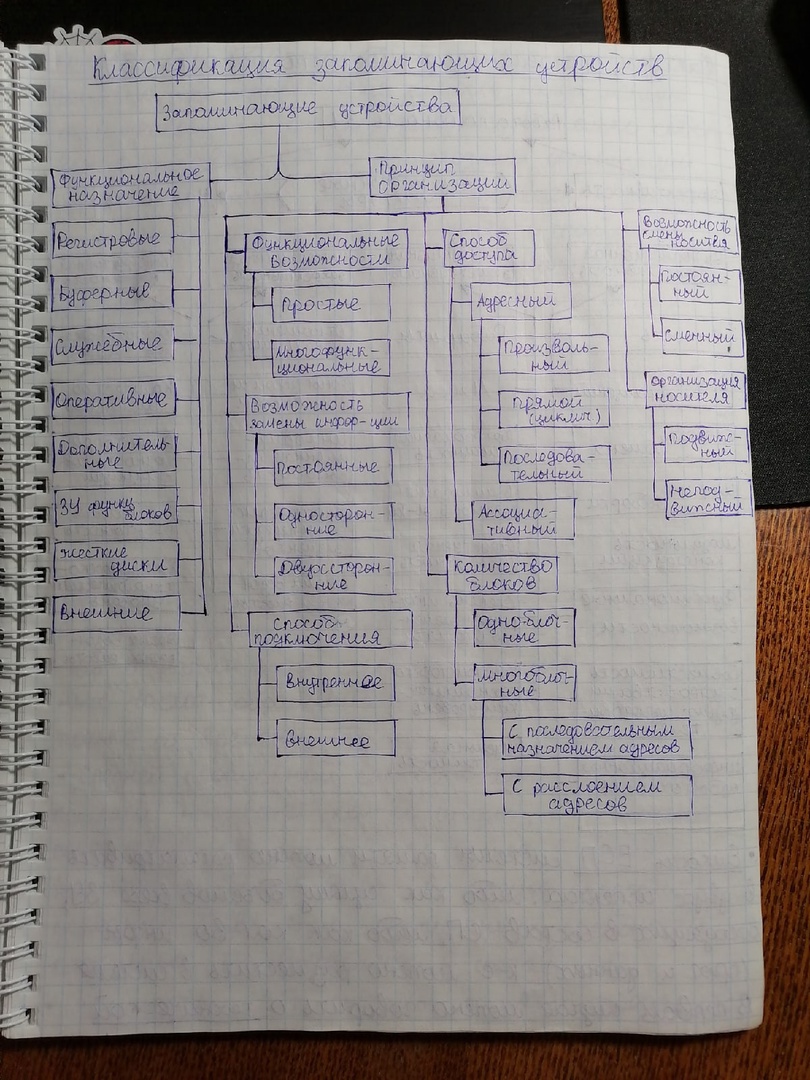
Во-вторых, **увеличивается объем памяти**. Регистры могут содержать в лучшем случае 128 байтов, кэш-память — несколько мегабайтов, основная память — десятки гигабайтов и т.д.

**Кэш (cache - запас)** - **быстродействующая буферная память** между процессором и основной памятью (в общем случае - между двумя любыми запоминающими устройствами с разным временем доступа). Кэш служит для частичной компенсации разницы в скорости процессора и основной памяти - туда попадают наиболее часто используемые данные. Когда процессор первый раз обращается к ячейке памяти, ее содержимое параллельно копируется в кэш, и в случае повторного обращения в скором времени может быть с гораздо большей скоростью выбрано из кэша.

1. Адресное пространство ЭВМ. Виды памяти.

**Адресное пространство** представляет собой множество кодов 0, 1, 2, … (2n-1), где n – число адресных линий. **Адресное пространство** определяет число возможных отличимых друг от друга кодовых комбинаций (адресов), которое может выдать на адресную шину активное устройство.





1. Организация памяти с произвольным доступом. Банки памяти. Модули памяти.

**Оперативная память (ОП)** — совокупность ОЗУ, объединенных в одну систему, управляемую процессором. совокупность ОЗУ, объединенных в одну систему, управляемую процессором. Для обеспечения приспосабливаемости ЭВМ к конкретным потребностям пользователей применяют принцип блочного построения 0П(многоблочная ОП). Функциональном отношении многоблочная ОП рассматривается как одно

ОЗУ с емкостью, равной сумме емкостей блоков, и быстродействием, примерно равным быстродействию отдельного блока. Адрес ячеек такой 0П содержит адрес блока и адрес ячейки памяти в заданном блоке ОЗУ. Принцип обслуживания запросов к ОП - приоритетный. Устройствам присваиваются приоритеты: низший — центральному процессору, более высший — ВЗУ.

Из микросхем, памяти RAM (- Random Access Memory, память с

произвольным доступом) используется два основных типа: статическая (SRAM - Static RAM) и динамическая (DRAM - Dynamic RAM).

Назовем, упорядоченную последовательность информационных и управляющих слов образует массив. Количество ячеек памяти, используемое для представления массива в ЭВМ, называется длиной массива. Группа ячеек памяти с последовательными номерами Аб, Аб+1, Аб+2, ... , Аб+n, представляющая массив длиной (n+1), рассматривается как массив ячеек памяти с базовым адресом Аб.

|  |  |
| --- | --- |
| **Статическое** распределение памяти основано на выделении ячеек ОП для массивов в процессе анализа и составления программы, т. е. до начала решения задачи и при выполнении программы базисные адреса сохраняют постоянные значения. | **Динамическое** распределение памяти основано на выделении ячеек памяти для массивов с учетом их длин в порядке их появления в процессе  решения задачи. Оно используется для экономии ячеек памяти в пределах одной программы и при мультипрограммной работе ЭВМ для распределения памяти между программами. |

Обычные виды SRAM и DRAM называют также асинхронными, потому что установка адреса, подача управляющих сигналов и чтение/запись данных могут выполняться в произвольные моменты времени - необходимо только соблюдение временных соотношений между этими сигналами. Ячейки микросхем динамической памяти

организованы в виде прямоугольной (обычно - квадратной) матрицы; при обращении к микросхеме на ее входы вначале подается адрес строки матрицы, сопровождаемый сигналом RAS (Row Address Strobe - строб адреса строки), затем, через некоторое время - адрес столбца, сопровождаемый сигналом CAS (Column Address Strobe - строб адреса столбца).

|  |  |
| --- | --- |
| **Виды памяти** | **Описание** |
| SDRAM (Synchronous DRAM - синхронная динамическая память) | – память с синхронным доступом, работающая быстрее обычной асинхронной. Помимо синхронного метода доступа, SDRAM использует внутреннее разделение массива памяти на два независимых банка, что позволяет совмещать выборку из одного банка с установкой адреса в другом банке. SDRAM также поддерживает блочный обмен. Основная выгода от использования SDRAM состоит в поддержке последовательного доступа в синхронном режиме, где не требуется дополнительных тактов ожидания. |
| FPM DRAM (Fast Page Mode DRAM - динамическая память с быстрым страничным доступом) | Память со страничным доступом отличается от обычной динамической памяти  тем, что после выбора строки матрицы и удержании RAS допускает многократную установку адреса столбца, стробируемого CAS, и также быструю регенерацию по схем.е "CAS прежде RAS". |
| ED0 (Entended Data Out - расширенное время удержания данных на выходе) | фактически представляют собой обычные микросхемы FРМ, на выходе которых установлены регистры-защелки данных. При страничном обмене такие  микросхемы работают в режиме простого конвейера: удерживают на выходах  данных содержимое последней выбранной ячейки, в то время как на их входы уже подается адрес следующей выбираемой ячейки. |
| BEDO (Burst EDO - EDO с блочным доступом) | - память на основе EDO, работающая не одиночными, а пакетными циклами чтения/записи. Современные процессоры, благодаря внутреннему и внешнему кэшированию команд и данных, обмениваются с основной памятью преимущественно блоками слов максимальной ширины. В случае памяти BEDO отпадает необходимость постоянной подачи последовательных адресов на входы микросхем с соблюдением необходимых временных задержек - достаточно стробировать переход к очередному слову отдельным сигналом. |
| PB SRAM (Pipelined Burst SRAM - статическая память с блочным конвейерным, доступом) | - разновидность синхронных SRAM с внутренней конвейеризацией, за счет которой примерно вдвое повышается скорость обмена  опоками данных. |

|  |  |
| --- | --- |
| **Тип модуля памяти** | **Описание** |
| DIP (Dual In line Package - корпус с двумя рядами  выводов) | - классические микросхемы, сейчас применяются в блоках кэш-памяти. |
| SIP (Single In line  Package - корпус с одним рядом выводов) | - микросхема с одним рядом выводов, устанавливаемая вертикально. |
| SIPP (Single In line Pinned Package - модуль с  одним рядом проволочных выводов) | - модуль памяти, вставляемый в панель наподобие микросхем DIP/SIP. |
| SIMM (Single In line Memory Module) - модуль памяти с одним рядом контактов) | - модуль памяти, вставляемый в зажимающий разъем; применяется во всех современных платах, а также во многих адаптерах, принтерах и прочих устройствах. SIMM имеет контакты с двух сторон модуля, но все они соединены между собой, образуя как бы один ряд контактов. |
| DIMM (Dual In line Memory Module - модуль памяти с двумя рядами контактов) | - модуль памяти, похожий на SIMM, но с раздельными контактами, за счет чего увеличивается разрядность или число банков памяти в модуле |

Расположенные на системной плате и модулях памяти микросхемы (DIP, SIMM, SIPP и DIMM) организуются в банки памяти. Оперативная память монтируется в так называемых банках (банк – комплект микросхем, обеспечивающий требуемую разрядность). Если работает 32-разрядная системная шина (486), а микросхема имеет 8 линий ввода/вывода, то необходимо 4 микросхемы (4х8=32 линий), и они, если установлены на одной пластине, составляют банк, у Pentium – 64 линий и потому потребуется 8 микросхем без проверки на четность, 9 - с проверкой. Банк - это минимальный объем памяти, с которым может работать процессор за одно обращение. Банк может быть заполненным и незаполненным, частичное его заполнение не допускается. Если модули памяти находятся на пластине, которая крепится на материнской плате, то комплект разъемов, в которых будут располагаться пластины с микросхемами оперативной памяти, обеспечивающие требуемую разрядность, также называется банком. Банк может содержать как один, так и несколько разъемов на материнской плате. Внутри банка микросхемы должны подбираться одного вида и одного объема, желательно одного производителя. Разные банки по своим характеристикам могут отличаться и желательно различия минимизировать, иначе они не смогут согласованно работать. Современные компьютеры имеют оперативную память, в которой один банк расположен на одной пластине.

1. Организация постоянной и полупостоянной памяти ЭВМ.

**Постоянная память** используется для энергозависимого хранения системной информации— BIOS(программа начальной загрузки), таблиц знакогенераторов и т.д. Обычно при работе компьютера эта память только считывается, но имеется и режим программирования, при котором можно изменить содержимое. В ПК в качетсве постоянной памяти используют флеш-память, и ее объем составляет около 6 Мб. Быстродействие постоянной памяти гораздо меньше оперативной, поэтому для нее создается и используется ее теневая копия в оперативной памяти.

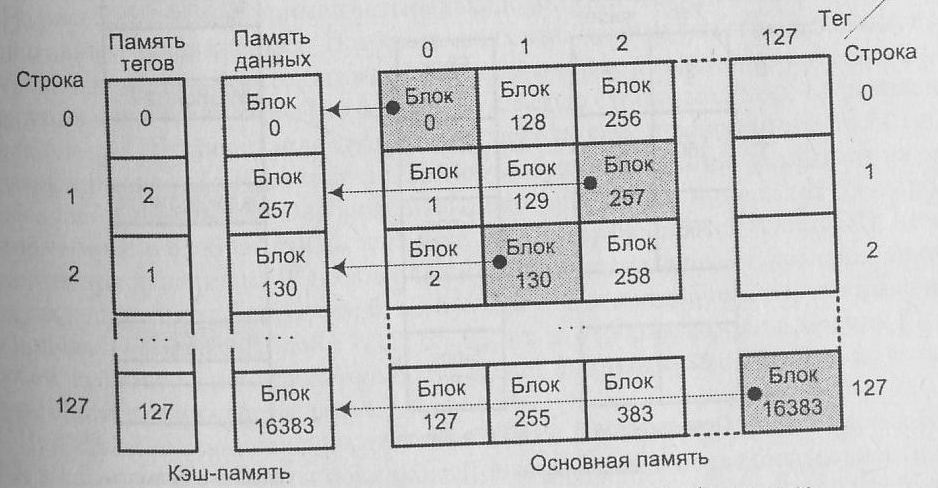
**Полупостоянная память** в основном используется для хранения информации о конфигурации (настроек) компьютера. Память конфигурации выполнена вместе с системными часами. Объем данной памяти составляет несколько десятков байт, а сохранность данных при отсутствии питания обеспечивается небольшой внутренней батарейкой.

1. Организация кэш-памяти. Кэш-память с прямым отображением, ассоциативная и наборно-ассоциативная кэш-память.

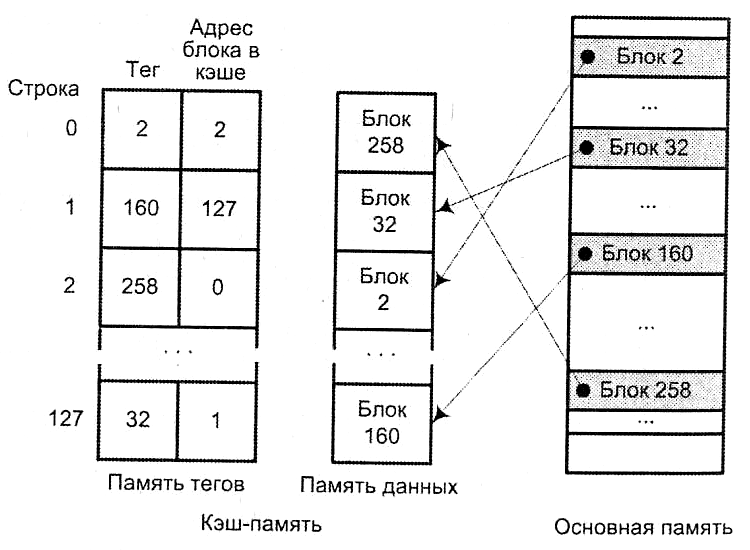
Принципы размещения блоков в кэш-памяти определяют три основных типа их организации:

- Если каждый блок основной памяти имеет только одно фиксированное место, на котором он может появиться в кэш-памяти, то такая кэш-память называется кэшем **с прямым отображением (direct mapped).** Это наиболее простая организация кэш-памяти, при которой для отображение адресов блоков основной памяти на адреса кэш-памяти просто используются младшие разряды адреса блока. Таким образом, все блоки основной памяти, имеющие одинаковые младшие разряды в своем адресе, попадают в один блок кэш-памяти, т.е.





- Если некоторый блок основной памяти может располагаться на любом месте кэш-памяти, то кэш называется полностью ассоциативным (fully associative).



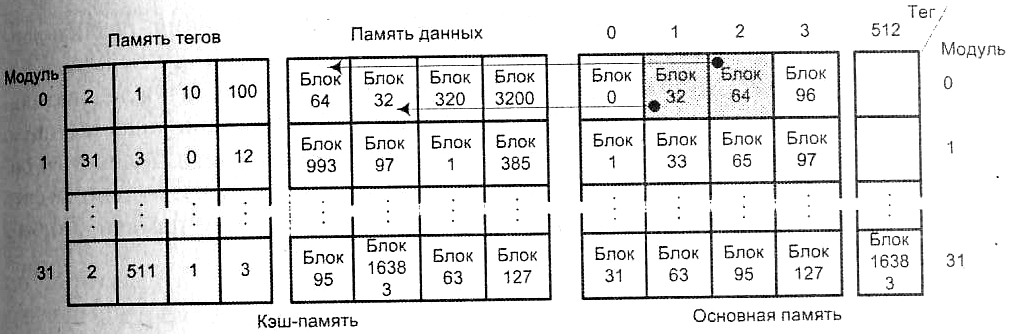
- Если некоторый блок основной памяти может располагаться на

ограниченном множестве мест в кэш-памяти, то кэш называется

множественно-ассоциативным (set associative). Обычно множество

представляет собой группу из двух или большего числа блоков в кэше. Если множество состоит из n блоков, то такое размещение называется множественно-ассоциативным с n каналами (n-way set associative). Для размещения блока прежде всего необходимо определить множество. Множество определяется младшими разрядами адреса блока памяти (индексом):





У каждого блока в кэш-памяти имеется адресный тег, указывающий, какой блок в основной памяти данный блок кэш-памяти представляет. Эти теги обычно одновременно сравниваются с выработанным процессором адресом блока памяти. Кроме того, необходим способ определения того, что блок кэш-памяти содержит достоверную или пригодную для использования информацию. Наиболее общим способом решения этой проблемы является добавление к тегу так называемого бита достоверности (valid bit). Адресация множественно-ассоциативной кэш-памяти осуществляется путем деления адреса, поступающего из процессора, на три части: поле смещения используется для выбора байта внутри блока кэш-памяти, поле индекса определяет номер множества, а поле тега используется для сравнения. Если общий размер кэш-памяти зафиксировать, то увеличение степени

ассоциативности приводит к увеличению количества блоков в множестве, при этом уменьшается размер индекса и увеличивается размер тега. При возникновении промаха, контроллер кэш-памяти должен выбрать подлежащий замещению блок. Польза от использования организации с прямым отображением заключается в том, что аппаратные решения здесь наиболее простые. Выбирать просто нечего: на попадание проверяется только один блок

и только этот блок может быть замещен. При полностью ассоциативной или множественно-ассоциативной организации кэш-памяти имеются несколько блоков, из которых надо выбрать кандидата в случае промаха. Как правило для замещения блоков применяются две основных стратегии: случайная и LRU.

В первом случае, чтобы иметь равномерное распределение, блоки-кандидаты выбираются случайно. В некоторых системах, чтобы получить воспроизводимое поведение, которое особенно полезно во время отладки аппаратуры, используют псевдослучайный алгоритм замещения. Во втором случае, чтобы уменьшить вероятность выбрасывания информации, которая скоро может потребоваться, все обращения к блокам фиксируются. Заменяется тот блок, который не использовался дольше всех (LRU - Least-Recently Used).

1. Методы записи в кэш-память. Cквозная запись, буферизированная сквозная запись, обратная запись.

Именно процессор определяет размер записи (обычно от 1 до 8

байтов) и только эта часть блока может быть изменена. В общем случае это подразумевает выполнение над блоком последовательности операций чтение-модификация-запись: чтение оригинала блока, модификацию его части и запись нового значения блока. Более того, модификация блока не может начинаться до тех пор, пока проверяется тег, чтобы убедиться в том, что обращение является попаданием. Поскольку проверка тегов не может выполняться параллельно с другой работой, то операции записи отнимают больше времени, чем операции чтения.

Очень часто организация кэш-памяти в разных машинах отличается

именно стратегией выполнения записи. Когда выполняется запись в кэш-память имеются две базовые возможности:

- **сквозная запись** (write through, store through) - информация

записывается в два места: в блок кэш-памяти и в блок более низкого уровня памяти.

- **буферизованная сквозная запись** – информация задерживается в кеш-буфере перед записью в ОП и переписывается в ОП в те циклы, когда ЦП к ней не обращается.

- **запись с обратным копированием** (write back, copy back, store in) - информация записывается только в блок кэш-памяти.

Модифицированный блок кэш-памяти записывается в основную

память только когда он замещается. Для сокращения частоты

копирования блоков при замещении обычно с каждым блоком кэш-памяти связывается так называемый бит модификации (dirty bit). Этот бит состояния показывает был ли модифицирован блок, находящийся в кэш-памяти. Если он не модифицировался, то обратное копирование отменяется, поскольку более низкий уровень содержит ту же самую информацию, что и кэш-память.

При записи с обратным копированием операции записи

выполняются со скоростью кэш-памяти, и несколько записей в один и тот же блок требуют только одной записи в память более низкого уровня. Поскольку в этом случае обращения к основной памяти происходят реже, вообще говоря требуется меньшая полоса пропускания памяти, что очень привлекательно для мультипроцессорных систем. При сквозной записи промахи по чтению не влияют на записи в более высокий уровень, и, кроме того, сквозная запись проще для реализации, чем запись с обратным копированием. Сквозная запись имеет также преимущество в том, что основная память имеет наиболее свежую

копию данных. Это важно в мультипроцессорных системах, а также для организации ввода/вывода. Когда процессор ожидает завершения записи при выполнении сквозной записи, то говорят, что он приостанавливается для записи (write stall). Общий прием минимизации остановов по записи связан с использованием буфера записи (write buffer), который позволяет процессору продолжить выполнение команд во время обновления содержимого памяти. Следует отметить, что остановы по записи могут возникать и при наличии буфера записи. При промахе во время записи имеются две дополнительные возможности:

- разместить запись в кэш-памяти (write allocate) (называется также

выборкой при записи (fetch on write)). Блок загружается в кэш-память, вслед за чем выполняются действия аналогичные выполняющимся при выполнении записи с попаданием. Это похоже на промах при чтении.

- не размещать запись в кэш-памяти (называется также записью в

окружение (write around)). Блок модифицируется на более низком

уровне и не загружается в кэш-память.

Обычно в кэш-памяти, реализующей запись с обратным копированием, используется размещение записи в кэш-памяти (в надежде, что последующая запись в этот блок будет перехвачена), а в кэш-памяти со сквозной записью размещение записи в кэш-памяти часто не используется (поскольку последующая запись в этот блок все равно пойдет в память).

Формула для среднего времени доступа к памяти в системах с кэш-памятью выглядит следующим образом:



1. Cегментная организация основной памяти. Дескрипторы. Таблицы дескрипторов.

Этот подход к организации памяти опирается на тот факт, что программы обычно разделяются на отдельные области-сегменты. Каждый сегмент представляет собой отдельную логическую единицу информации, содержащую совокупность данных или программ и расположенную в адресном пространстве пользователя. Сегменты создаются пользователями, которые могут обращаться к ним по символическому имени. В каждом сегменте устанавливается своя собственная нумерация слов, начиная с нуля.

Обычно в подобных системах обмен информацией между

пользователями строится на базе сегментов. Поэтому сегменты являются отдельными логическими единицами информации, которые необходимо защищать, и именно на этом уровне вводятся различные режимы доступа к

сегментам. Можно выделить два основных типа сегментов: программные сегменты и сегменты данных (сегменты стека являются частным случаем сегментов данных). Поскольку общие программы должны обладать свойством повторной входимости, то из программных сегментов допускается только

выборка команд и чтение констант. Запись в программные сегменты может рассматриваться как незаконная и запрещаться системой. Выборка команд из сегментов данных также может считаться незаконной и любой сегмент данных

может быть защищен от обращений по записи или по чтению.

Для реализации сегментации было предложено несколько схем, которые отличаются деталями реализации, но основаны на одних и тех же принципах.

В системах с сегментацией памяти каждое слово в адресном пространстве пользователя определяется виртуальным адресом, состоящим из двух частей: старшие разряды адреса рассматриваются как номер сегмента, а младшие - как

номер слова внутри сегмента. Наряду с сегментацией может также использоваться страничная организация памяти. В этом случае виртуальный адрес слова состоит из трех частей: старшие разряды адреса определяют номер сегмента, средние - номер страницы внутри сегмента, а младшие - номер слова

внутри страницы. Как и в случае страничной организации, необходимо обеспечить преобразование виртуального адреса в реальный физический адрес основной памяти. С этой целью для каждого пользователя операционная система должна

сформировать таблицу сегментов. Каждый элемент таблицы сегментов содержит описатель (дескриптор) сегмента (поля базы, границы и индикаторов режима доступа). При отсутствии страничной организации поле базы определяет адрес начала сегмента в основной памяти, а граница - длину

сегмента. При наличии страничной организации поле базы определяет адрес начала таблицы страниц данного сегмента, а граница - число страниц в сегменте. Поле индикаторов режима доступа представляет собой некоторую комбинацию признаков блокировки чтения, записи и выполнения.

Таблицы сегментов различных пользователей операционная система хранит в основной памяти. Для определения расположения таблицы сегментов выполняющейся программы используется специальный регистр защиты, который загружается операционной системой перед началом ее выполнения. Этот регистр содержит дескриптор таблицы сегментов (базу и границу), причем база содержит адрес начала таблицы сегментов выполняющейся программы, а

граница - длину этой таблицы сегментов. Разряды номера сегмента виртуального адреса используются в качестве индекса для поиска в таблице сегментов. Таким образом, наличие базово-граничных пар в дескрипторе таблицы сегментов и элементах таблицы сегментов предотвращает

возможность обращения программы пользователя к таблицам сегментов и страниц, с которыми она не связана. Наличие в элементах таблицы сегментов индикаторов режима доступа позволяет осуществить необходимый режим доступа к сегменту со стороны данной программы. Для повышения

эффективности схемы используется ассоциативная кэш-память. Отметим, что в описанной схеме сегментации таблица сегментов с индикаторами доступа предоставляет всем программам, являющимся частями некоторой задачи, одинаковые возможности доступа, т. е. она определяет

единственную область (домен) защиты. Однако для создания защищенных подсистем в рамках одной задачи для того, чтобы изменять возможности доступа, когда точка выполнения переходит через различные программы,

управляющие ее решением, необходимо связать с каждой задачей множество доменов защиты. Реализация защищенных подсистем требует разработки некоторых специальных аппаратных средств.

1. Страничная организация памяти. Каталог страниц.

Основное применение страничного преобразования адреса – организация виртуальной памяти. Виртуальная память позволяет использовать программам больший объем памяти, чем установленный на компьютере физический объем памяти. Остальная информация может быть сброшена на внешний носитель.

Управление страничным разбиением памяти обычно возлагается на специальную микросхему MMU (Memory Management Unit – устройство управления памятью).

Как и сегментация, страничная организация памяти связана с преобразованием виртуального адреса (в данном случае линейного) в физический. В страничном преобразовании базовым объектом памяти является блок фиксированного размера, называемый страницей (page). Размер страницы - 4 Кбайт.

Страничное управление (Paging) является средством организации виртуальной памяти с подкачкой страниц по запросу (Demand-Paged Virtual Memory). В отличие от сегментации, которая организует программы и данные в модули различного размера, страничная организация оперирует с памятью, как с набором страниц одинакового размера. В момент обращения страница может присутствовать в физической оперативной памяти, а может быть выгруженной на внешнюю (дисковую) память. При обращении к выгруженной странице памяти процессор вырабатывает исключение #PF – отказ страницы, а программный обработчик исключения (часть ОС) получит необходимую информацию для свопинга – «подкачки» отсутствующей страницы с диска.

СТРУКТУРА СТРАНИЦ.

При разрешенном страничном преобразовании физическая память компьютера разбивается на страницы. Иногда страницы называют страничными кадрами – page frame – размером 4 Кбайт. Поскольку часть страниц находится вне физической памяти, предусмотрен механизм замены страниц по требованию. Это позволяет программам использовать для своих нужд все линейное адресное пространство, не заботясь о том, занята ли физическая память другими процессами.

Границы сегментов и страниц могут не совпадать. Однако желательно, для повышения производительности системы, выравнивать границы сегментов на границы страничного кадра.

В отличие от сегмента, для страниц есть только два уровня привилегий:

* пользовательский (User) – системный программный уровень 3
* супервизора (Supervisor) – системный программный уровень 0,1,2

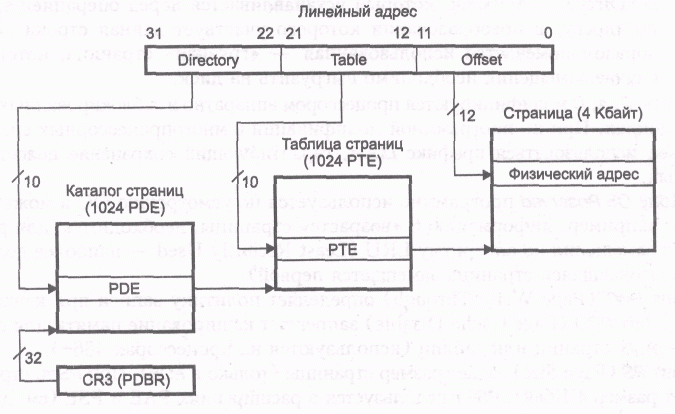
ФОРМИРОВАНИЕ АДРЕСА ПРИ СТРАНИЧНОМ ПРЕОБРАЗОВАНИИ.

Для уменьшения размера таблицы страниц предусмотрена двухуровневая схема преобразования адреса. Основой страничного преобразования служит регистр управления CR3, содержащий 20-ти битный физический базовый адрес каталога страниц текущей задачи. Предполагается, что каталог выровнен по границе страничного кадра, постоянно находится в памяти и не участвует в свопинге.

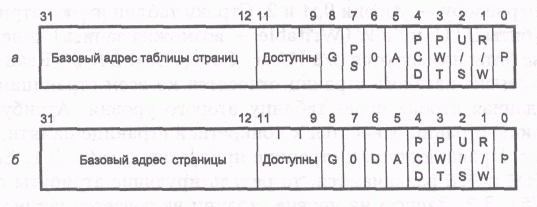
Механизм имеет три части:

* каталог страниц (Page Directory) – корневая страница, которая содержит 1024 32-х битных дескриптора, называемых элементами каталога страниц – PDE (Page Directory Entry). Каждый из них адресует подчиненную таблицу страниц.
* таблицы страниц (Page Table) – каждая из этих таблиц содержит 1024 32-х битных дескриптора, называемых элементами таблицы страниц – PTE (Page Directory Entry). Каждый PTE содержит адрес страничного кадра в физической памяти.
* Страницы (Page Frame).

Механизм включается установкой бита PG=1 в регистре CR0. Регистр CR2 хранит линейный адрес отказа (Page Fault Linear Address) – адрес памяти, по которому был обнаружен последний отказ страницы. Регистр CR3 хранит физический адрес каталога страниц (Page Directory Physical Base Address). Его младшие 12 бит всегда нулевые, т.к. каталог выравнивается по границе страницы.



Каталог страниц, размером 4 Кбайт содержит 1024 32-х битных строки PDE. Каждая строка PDE (см.рис.) содержит 20 старших бит адреса таблицы следующего уровня (младшие биты этого адреса всегда нулевые) и признаки (атрибуты) этой таблицы. Индексом поиска в каталоге страниц являются 10 старших бит линейного адреса (А22-А31)



Рисунок, где а – строка каталога - PDE, б – строка таблицы - PTE

Каждая таблица страниц также имеет 1024 строки PTE аналогичного формата (см.рис.), но эти строки содержат базовый физический адрес (Page Frame Address) и атрибуты самих страниц. Индексом поиска в таблице являются биты А12-А21 линейного адреса.

Физический адрес получается из адреса страницы, взятого из таблицы, и младших 12 бит линейного адреса.

Строки каталога и таблиц имеют следующие биты атрибутов:

P (Present) – бит присутствия. Разрешает использования таблицы страниц или кадра страницы при P=1. Если же P=0, то обращение к соответствующему разделу или странице запрещено и попытка их использования вызовет прерывание типа 14 (отсутствие доступа к странице). Отметим, что при P=0 остальные биты доступны операционной системе и могут использоваться для получения информации о местонахождении данной таблицы.

R/W – бит чтения/записи и U/S – бит пользователя/супервизора. Эти биты определяют права доступа к соответствующему разделу или странице для программ пользователя, имеющих минимальные уровни привилегий (системный программный уровень 3). Если осуществляется запрос с уровнем привилегий 3 (программы пользователя), то при значении U/S=0 ему запрещается доступ к соответствующему разделу или кадру страницы. Если U/S=1, то при значении R/W=0 разрешается только чтение раздела или страницы, а при R/W=1 – и чтение, и запись. При запросах с бОльшими привилегиями (системные программные уровни 0,1,2) допускается запись и чтение разделов и страниц при любых значения U/S, R/W.

PWT и PCD –эти биты используются для управления работой кэш-памяти при страничной адресации. (Эти биты отсутствуют в указателях, используемых микропроцессором 80386).

PWT (Page Write Through) – бит обратной записи - определяет метод обновления содержимого внешней кэш-памяти. При PWT=1 для текущей страницы обновление реализуется методом сквозной записи (как для внутренней кэш-памяти), при PWT=0 – методом обратной записи.

PCD (Page Cache Disable) – бит запрещения кэширования страницы - запрещает кэширование памяти для обслуживаемых страниц или таблиц.

A (Accessed) – бит доступа. Автоматически устанавливается микропроцессором в состояние А=1 при обращении к данному разделу или странице для записи или чтения информации.

D (Dirty) – бит модификации. Этот бит в указателе кадра страницы устанавливается в состояние D=1 при записи на данную страницу. Таким образом, помечается использованная, «грязная» страница, которую в случае замещения необходимо выгрузить на диск.

PS (Page Size) – задает размер страницы (только в POE (Power Over Ethernet) – технология, позволяющая передавать сетевую информацию и одновременно обеспечивать питание Ethernet устройств). При PS=0 имеет размер 4 Кбайт, PS=1 используется в расширениях PAE\* и PSE\*.

OS Reserved – доступно. Зарезервированы для операционной системы, которая может использовать их для размещения информации о времени последнего обращения к данному разделу или странице. Эта информация используется для определения разделов и страниц, подлежащих замене из внешней памяти.

Бит G (Global) - появившийся в Р6, определяет глобальность страницы. Он анализируется только в строке, указывающей на страницу физической памяти (в РТЕ для страниц в 4 Кбайт, в PDE — для страниц 2 Мбайт или 4 Мбайт). Этот бит, управляемый только программно, позволяет пометить страницы гло­бального использования (например, ядра ОС). При установленном бите PGE в регистре CR4 строки с указателями на глобальные таблицы не будут аннулиро­ваться в TLB при загрузке CR3 или переключении задач, что снижает издержки обслуживания виртуальной памяти.

\* PAE - (Physical Address Extension – расширение физических адресов) – аппаратная технология, с помощью которой приложения Windows на 32-разрядных серверах с процессорами IA-32 могут адресовать физическую память свыше 4 Гбайт. Для создания дополнительной адресуемой памяти в PAE используется 36 разрядов. Работая с PAE, диспетчер памяти операционной системы использует трехуровневую схему преобразования адресов для обращения к памяти свыше 2 или 3 Гбайт, доступных на серверах без PAE. Как и любой механизм, с помощью которого операционная система может предоставить дополнительную память для приложений, PAE сокращает потребность в записи страниц из памяти в системный файл подкачки.

1. Система защиты памяти в защищенном режиме.
2. Организация многозадачной работы ЭВМ в защищенном режиме.
3. Классификация прерываний. Прерывания и исключения. Аппаратные и программные прерывания.

Для обеспечения перехода от одной программы к другой в

мультипрограммной ЭВМ вводится так называемый режим прерывания программ.

**Прерывание программы** — способность процессора прекращать выполнение текущей программы и ее управление при возникновении определенных условий. Сигналы, вызывающие прерывание программы, называются сигналами прерывания или запросами прерывания.В зависимости от условий возникновения сигналы прерывания подразделяют на пять классов:

1. **Прерывание от схем контроля, или машинное прерывание(исключения)**, возникает в случае обнаружения ошибок в каких-либо блоках и устройствах ЭВМ. При этом происходит переключение к диагностической программе, позволяющей локализовать место неисправности.
2. **Программное прерывание, или прерывание из-за ошибок программы**, возникает при обнаружении ошибок в программе или при появлении необычных ситуаций при ее выполнении. Например, при переполнении разрядной сетки, делении на нуль и т. п.
3. **Внешнее прерывание** возникает в случае появления сигналов от внешних объектов: датчиков времени (электронных часов), кнопок запросов на пультах инженера и оператора и других ЭВМ, подключенных к данной ЭВМ; аппаратуры передачи данных по линиям связи, от датчиков технологических процессов и т. п.
4. **Прерывание от устройств ввода-вывода** позволяет процессору получать информацию о состоянии каналов и периферийных устройств и отвечать на эти сигналы. Это прерывание от устройств ввода-вывода сформируется, когда канал и ПфУ не могут выполнить за данную операцию; при возникновении особой ситуации в процессе выполнения операции ввода-вывода (ошибка в информации, обрыв перфоленты и бумаги, замятие перфокарты и др.); в момент окончания операции ввода-вывода.
5. **Прерывание при обращении к управляющим подпрограммам-диспетчеру** в случаях, когда предусмотрено выполнение каких-либо действиипо управлению ЭВМ и ВС. Например, если при выполнении рабочей программы нужно выполнить действие, выходящее за пределы возможностей системы команд ЭВМ, то процессор формирует запрос на прерывание в виде обращения к соответствующей подпрограмме, которая реализует затребованное действие, после чего управление вновь передается прерванной программе. Путем обращения к управляющим программам организуется ввод-вывод, перераспределение памяти, прекращается выполнение программы.
6. Организация обработки аппаратных прерываний. Контроллер прерываний, его регистры и установка режимов работы.
7. Организация работы системы прерываний в реальном режиме.

В реальном режиме работы в системе прерываний используется понятие вектора прерывания, поскольку для указания адреса программы обработки прерывания здесь требуется не одно значение, а два (значение для сегментного регистра кода и значение для указателя команд), то есть мы имеем дело не со скалярной величи­ной, а с «векторной», состоящей из двух скалярных.

1. Организация работы системы прерываний в защищенном режиме. Шлюзы.
2. Организация программно-управляемого ввода-вывода ЭВМ.

Программно управляемая передача данных осуществляется при участии и под непосредственным управлением процессора. Данные между памятью и периферийными устройствами пересылаются через процессор.

При вводе- выводе в режиме ПДП процессор не участвует в обмене и либо приостанавливает свою работу на время обмена, либо выполняет параллельно с обменом обработку команд и данных, не требующих обращения к ОШ. Обмен данными между УВВ и ОП осуществляется напрямую, минуя процессор. Ввод-вывод в режиме ПДП является (принципиально) более быстродействующим, нежели программно управляемый.

Программный способ осуществляется либо по прерыванию, либо без прерывания. При вводе-выводе с прерыванием программы инициализация ввода-вывода осуществляется сигналом запрос на прерывание от ПУ. При вводе-выводе без прерывания его инициализация осуществляется текущей командой программы. Ввод-вывод без прерывания бывает синхронным и асинхронным. При синхронном вводе-выводе готовность ПУ к обмену не проверяется, при асинхронном- проверяется.

Для исключения потерь информации при асинхронном вводе- выводе, процессор при начале обмена проверяет готовность устройства ввода-вывода к обмену, путем считывания его регистра состояния (статус - регистра). Регистр состояния должен содержать информацию о функционировании устройства ввода-вывода и является дополнительным регистром ПУ. Регистры данных и состояния подключаются к шине данных и каждому из них присваиваются уникальные адреса. При неготовности ПУ к обмену процессор выполняет другие действия.

Асинхронный ввод-вывод приводит к непроизводительным затратам времени работы процессора или к простоям ПУ. Увеличение в программе количества точек анализа состояния ПУ увеличивает длину программы и время работы процессора, уменьшение- к увеличению простоя ПУ. Тем не менее, такой способ ввода-вывода широко используется в ЭВМ

1. Организация ввода-вывода с использованием системы прерываний.

Чтобы сигнализировать процессору об окончании работы, устройство инициализирует прерывание, выставляя сигнал на выделенную устройству линию шины (а не выделенный провод).

Контроллер прерываний - обслуживает поступающие прерывания от устройств.

Если необработанных прерываний нет, прерывание выполняется немедленно.

Если необработанных прерываний есть, контроллер игнорирует прерывание. Но устройство продолжает удерживать сигнал прерывания на шине до тех пор, пока оно не будет обработано.

Алгоритм работы:

Устройство выставляет сигнал прерывания

Контроллер прерываний инициирует прерывание, указывая номер устройства

Процессор начинает выполнять обработку прерывания, вызывая процедуру

Эта процедура подтверждает получение прерывания контроллеру прерываний

1. Организация ввода-вывода с использованием прямого доступа в память.

Прямой доступ к памяти реализуется с помощью DMA - контроллера.

Контроллер содержит несколько регистров:

регистр адреса памяти

счетчик байтов

управляющие регистры, могут содержать:

- порт ввода-вывода

- чтение или запись

- единицы переноса (побайтно или пословно)

Без контроллера происходит следующее:

Процессор дает команду дисковому контроллеру прочитать данные в буфер,

Считываются данные в буфер, контроллер проверяет

контрольную сумму считанных данных (проверка на ошибки). Процессор, до прерывания, переключается на другие задания.

Контроллер диска инициирует прерывание

Операционная система начинает работать и может считывать из буфера данные в память

С контроллером происходит следующее:

Процессор программирует контроллер (какие данные и куда переместить)

Процессор дает команду дисковому контроллеру прочитать данные в буфер

Считываются данные в буфер, контроллер диска проверяет контрольную сумму считанных данных, (процессор, до прерывания, переключается на другие задания).

Контроллер DMA посылает запрос на чтение дисковому контроллеру

Контроллер диска поставляет данные на шину, адрес памяти уже находится на шине, происходит запись данных в память

Когда запись закончена, контроллер диска посылает подтверждение DMA контроллеру

DMA контроллер увеличивает используемый адрес и уменьшает значение счетчика байтов

Все повторяется с пункта 4, пока значение счетчика не станет равной нулю.

Контроллер DMA инициирует прерывание

Операционной системе не нужно копировать данные в память, они уже там.

1. Контроллер ПДП и его регистры.

Контроллер осуществляет двунаправленный обмен данными между памятью и ВУ (по требованию ВУ), при этом в адресном канале микропроцессорной систе­мы формируются параметры заданного массива адресов ячеек памяти (началь­ный адрес и число циклов) и управляющие сигналы. Каждый из четырех каналов контроллера обеспечивает адресацию (путем инкрементирования выработанного адреса) внешней памяти массивами объемом до 16К байт с возможностью зада­ния любого из 64К начальных адресов.

Есть 2 регистра.

1. Организация и управление режимом работы ППИ.

Программируемый параллельный ин­терфейс (ППИ) позволяет организовать обмен 8–разрядными операндами между микропроцессором и внешними устройствами (ВУ) по трем каналам. Направле­ние обмена и режим работы каждого канала задаются программно путем записи управляющего слова. Программируемый параллельный интерфейс (ППИ) предназначен для следующих целей:

· двунаправленного побайтного обмена информацией с внешними и внутренними устройствами,

· управления исполнительными механизмами, · управления аналого-цифровыми преобразователями и вводом информации в МПС от них,

· обслуживания контактных датчиков, управления внутри МП системы, in,

· переключения страниц памяти,

· обслуживания стандартной периферии (принтер, сканер, плоттер) и ряд других применений. В ППИ можно выделить следующие составные части :

● двунаправленный 8–разрядный буфер канала данных (БКД) с тремя состояниями

● три 8–разрядных канала связи (КА, КВ, КС) с ВУ, причем канал КС может быть разделен на два канала КС1, КС2. В канале КА предусмотрено два регистра, один из которых используется для передачи данных со стороны микропро­цессора в ВУ, другой — в обратном направлении.

● регистр управляющего слова (РУС), предназначенный для организации обме­на данными между регистрами каналов и ВУ. На его вход подается сигнал сброса R.

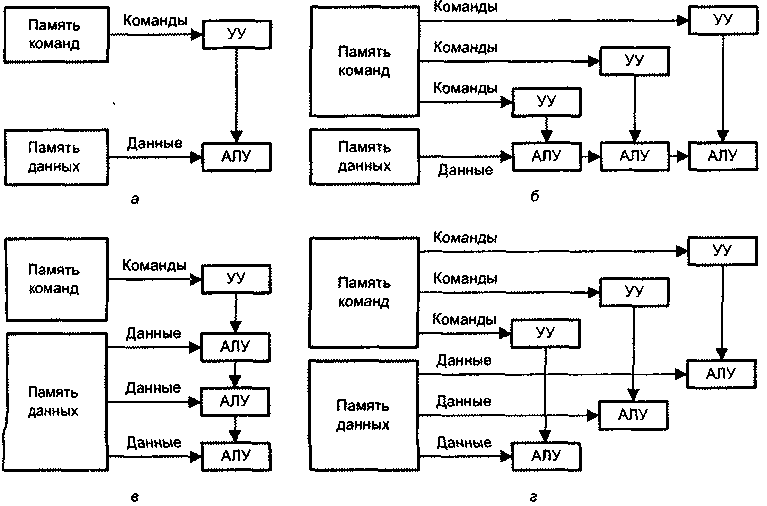
● устройство управления (УУ), предназначенное для выбора канала и обеспечения требуемых режимов работы.

1. Организация и управление режимом работы программируемого таймера.

https://www.computer-museum.ru/technlgy/i8253.htm

1. Классификация вычислительных систем. Архитектура SIMD, MIMD.

Среди всех рассматриваемых систем классификации ВС наибольшее признание получила классификация, предложенная в 1966 году М. Флинном. В ее основу положено понятие потока, под которым понимается последовательность элементов, команд или данных , обрабатываемая процессором. В зависимости от количества потоков команд и потоков данных Флинн выделяет четыре класса архитектур: SISD, MISD, SIMD, MIMD.



Архитектура вычислительных систем по Флинну: а — SISD; б — MISD; в — SIMD; г-MIMD

**SIMD** (Single Instruction Stream/Multiple Data Stream) — одиночный поток команд и множественный поток данных ВМ данной архитектуры позволяют выполнять одну арифметическую операцию сразу над многими данными — элементами вектора. Бесспорными представителями класса SIMD считаются матрицы процессоров, где единое управляющее устройство контролирует множество процессорных элементов. Все процессорные элементы получают от устройства управ­ления одинаковую команду и выполняют ее над своими локальными данными. В принципе в этот класс можно включить и векторно-конвейерные ВС, если каждый элемент вектора рассматривать как отдельный элемент потока данных.

**MIMD** (Multiple Instruction Stream/Multiple Data Stream) — множественный поток команд и множественный поток данных. Класс предполагает наличие в вычислительной системе множества устройств обработки команд, объе­диненных в единый комплекс и работающих каждое со своим потоком команд и данных. Класс MIMD чрезвычайно широк, поскольку включает в себя всевоз­можные мультипроцессорные системы. Кроме того, приобщение к классу MIMD зависит от трактовки. Так, ранее упоминавшиеся векторно-конвейерные ВС можно вполне отнести и к классу MIMD, если конвейерную обработку рассматривать как выполнение множества команд (операций ступеней конвейера) над множе­ственным скалярным потоком.

Схема классификации Флинна вплоть до настоящего времени является наиболее распространенной при первоначальной оценке той или иной ВС, поскольку позволяет сразу оценить базовый принцип работы системы, чего часто бывает достаточно. Однако у классификации Флинна имеются и очевидные недостатки, например неспособность однозначно отнести некоторые архитектуры к тому или иному классу. Другая слабость — это чрезмерная насыщенность класса MIMD. Все это породило множественные попытки либо модифицировать классификацию Флинна, либо предложить иную систему классификации.

1. Конвейерная организация вычислений. Структурные конфликты, конфликты по данным.
2. Организация многопроцессорных ВС с общей и распределенной памятью.
3. Когерентность кэш-памяти.

Когерентность памяти (memory coherence) - свойство вычислительных систем, в которых два или более процессора имеют доступ к общей области памяти.

Оснащение каждого процессора локальной кэш-памятью приводит к так называемой проблеме когерентности или обеспечения согласованности кэш-памяти. Система является когерентной, если каждая операция чтения по какому-либо адресу, выполненная любым из процессоров, возвращает значение, занесенное в ходе последней операции записи по этому адресу, вне зависимости от того, какой из процессоров производил запись последним.

Поддержание согласованности требует, чтобы при изменении элемента данных одним из процессоров соответствующие изменения были проведены в кэш-памяти остальных процессоров, где есть копия измененного элемента данных, а также в общей памяти.

В решении проблемы когерентности выделяются два подхода: программный и аппаратный. В некоторых системах применяют стратегии, совмещающие оба подхода.

1. Организация многомашинных ВС. Особенности организации ВС на основе кластерной связи.

Многомашинная ВС - вычислительная система на основе целых компьютеров.

Вычислительные системы бывают:однородные;неоднородные.Однородная ВС строится на основе однотипных компьютеров или процессоров, позволяет использовать стандартные наборы программных средств, типовые протоколы (процедуры) сопряжения устройств. Их организация значительно проще, облегчается обслуживание систем и их модернизация.Неоднородная ВСвключает в свой состав различные типы компьютеров или процессоров.Припостроениисистемыприходитсяучитыватьихразличныетехнические и функциональные характеристики, что существенно усложняет создание и обслуживание таких систем.

1. Организация матричных ВС. Систолические и волновые матрицы.

Ключевые компоненты матричных ВС:

- Набор процессорных элементов

- Набор банков памяти

- Коммутационная среда

- Блоки локального управления

- Блок общего управления

Наиболее общая структура МВС содержит наборы идентичных ПЭ и банков памяти. Каждый ПЭ может непосредственно соединяться со своей секцией памяти. Связь между ПЭ и присвоенным ему банком памяти осуществляется посредством коммутационной среды, которая объединяет все ПЭ. Коммутационная среда изображена как отдельный компонент, подключаемый к ПЭ с помощью входов и выходов. В общем смысле при такой организации коммутирующая сеть обычно распределена между различными процессорами.

В другой часто встречающейся конфигурации используется более сложная коммутационная среда: коммутатор устанавливается между процессорами и банками памяти. Такая конфигурация устраняет ограничения на связь ПЭ только с одним банком памяти и позволяет объединить в матрицу различное число банков памяти и ПЭ.