**1.АППАРАТНАЯ ОСНОВА**

Аппаратная часть – технические средства обработки информации (средства вычислительной техники).

**2. ВЫЧИСЛИТЕЛЬНАЯ МАШИНА**

Вычислительная машина – это комплекс технических и программных средств, создающих возможность автоматизации обработки информации по заданному алгоритму и получения результата в необходимой форме.

ВМ – цифровое устройство, в котором определенным образом закодированные команды хранятся в памяти и выбираются из нее в момент исполнения.

**3. ВЫЧИСЛИТЕЛЬНАЯ СИСТЕМА КАК КОМПОНЕНТ IT-ИНФРАСТРУКТУРЫ**

Вычислительная система – совокупность нескольких взаимосвязанных и взаимодействующих процессов или вычислительных машин (компьютеров), организованная для совместного выполнения процессов обработки информации. (Объединенная коммуникационной средой)

Распределенная ВС ≡ Вычислительная сеть

IT-инфраструктура – это комплекс взаимосвязанных информационных систем и сервисов, обеспечивающих функционирование и развитие средств информационного взаимодействия предприятия.

Вычислительная система как один из ключевых компонентов IT-инфраструктуры обеспечивает необходимую пользователям производительность и доступность информационных систем.

**4. ПРЕДПОСЫЛКИ СОЗДАНИЯ ЭЛЕКТРОННЫХ ВЫЧИСЛИТЕЛЬНЫХ УСТРОЙСТВ**

1) Паскаль. Суммирующая машина. 1642 год

2) Однер. Арифмометр. 1874 год.

3) Машина Беббиджа

4) Холерит. Статистические вычисления. 1890 год.

5) Аналоговая вычислительная машина. В аналоговых вычислительных машинах обработка информации происходит с помощью специально подобранного физиче ского проце сса, моделирующего некоторую вычисляющую закономерность. (Дифференциальные уравнения, интегралы, производные).

6) Машина Тьюринга - основополагающая, теоретическая предпосылка для создания вычислительных машин. 1936 год Алан Тьюринг заложил основы теории алгоритмов и теории автоматов. Машина Тьюринга - автоматическое устройство, способное находиться в конечном числе внутренних состояний и снабженное бесконечной внутренней памятью. Среди всех состояний выделяют начальное и конечное.

7) 1938 год. Был создан программируемый вычислитель, названный Z1.

8) 1940 год. Американцы разработали модель электронного вычислительного устройства.

9) 1946 год. Была разработана ЭНИАК на электронных лампа

**5. ОСНОВОПОЛАГАЮЩИЕ СТРУКТУРНЫЕ ПРИНЦИПЫ ПОСТРОЕНИЯ СОВРЕМЕННЫХ СРЕДСТВ ВТ**

Модульность – любая система представляет собой набор модулей с известными параметрами и известными связями, что позволяет увеличить скорость разработки средств ВТ, используя готовые модули.

Магистральность – любой обмен данными между модулями происходит по единым магистралям с использованием наиболее быстрого алгоритма.

Микропрограммируемость – подразумевает возможность создания микропрограмм управления для блока управления, использующих идеологию управляющих автоматов с хранимой или программируемой логикой.

Модельность – одна и та же идеология средства ВТ имеет различные модели, различаемые по производительности, но имеющие одну и ту же целевую значимость для пользователя. Модель более высокого уровня может обрабатывать микропрограммы моделей более низкого уровня, что обеспечивает преемсьвенность о развитии.

**6. АРХИТЕКТУРА ЭВМ**

Архитектура ЭВМ – (абстрактное понятие) совокупность основных функциональных (аппаратных) блоков и схем их взаимодействия, определяющих функционально-логическую и структурную организацию вычислительной машины.

-структурная схема ЭВМ

-средства и способы доступа к элементам структурной схемы ЭВМ

-организация и разрядность интерфейсов ЭВМ

-организация и способы адресации памяти

-способы представления и форматы данных ЭВМ

-набор машинных команд ЭВМ

-обработка прерываний

**7. РАЗНОВИДНОСТИ АРХИТЕКТУР ВМ И УСТРОЙСТВ**

|  |  |  |
| --- | --- | --- |
| Тип архитектуры | Характеристика | Примеры архитектур |
| Вычислительная система | Взаимосвязь компонентов | Кластерная, массивно-параллельная |
| ЭВМ в целом | Взаимосвязь компонентов | Иерархическая, магистральная |
| Центральное устройство | Использование оперативной памяти | Принстонская, гарвардская |
| Микроархитектура процессора | Обработка микроопераций, кэш-память, конвейеры | Intel P5-P6, Nehalem, Skylake; AMD K7-K8 |
| Микроархитектура чипсета | Взаимосвязь компонентов системной платы ПК | «Северный мост – южный мост» |
| Система команд | Команды процессора | IA-32, IA-64, AMD64 |

**8. ТЕХНИКО-ЭКСПЛУАТАЦИОННЫЕ ХАРАКТЕРИСТИКИ ЭВМ**

-быстродействие

-разрядность

-формы представления чисел

-номенклатура и характеристики запоминающих устройств

-номенклатура и характеристики устройств ввода-вывода информации

-типы и характеристики внутренних и внешних интерфейсов

-наличие многопользовательских режимов и поддержка многозадачности

-типы и характеристики используемых ОС

-система команд и их структура

-функциональные возможности программного обеспечения и его наличие

-программная совместимость с другими типами ЭВМ

-срок эксплуатации

-условия эксплуатации

**9. ОСНОВНЫЕ КЛАССИФИКАЦИОННЫЕ ПРИЗНАКИ ВМ**

1.Принцип действия

-аналоговые

-цифровые

-гибридные

2.Используемая элементная база

3.Назначение

-универсальные

-проблемно-ориентированные (промышленные компьютеры со спец. конструктивным исполнением)

4.Размеры и вычислительная мощность

5.Особенности архитектуры

**10. ЦИФРОВЫЕ ВМ**

То же, что ЭВМ. Уточнение типа (ЦЭВМ) производится в случаях, когда это особо необходимо, например, в сложных вычислительных системах, включающих ЭВМ разных видов.

**11. АНАЛОГОВЫЕ ВМ**

Вычислительная машина непрерывного действия, обрабатывающая аналоговые данные. Предназначена для воспроизведения определенных соотношений между непрерывно изменяющимися физическими величинами. Основные области применения связаны с моделированием различных процессов и систем.

**12. ГИБРИДНЫЕ ВМ**

ЭВМ, сочетающая аналоговую и цифровую форму обработки данных.

**13. СПЕЦИАЛИЗИРОВАННЫЕ ВМ**

Контроллеры, управляющие несложными техническими устройствами и процессами, микропроцессоры специального назначения.

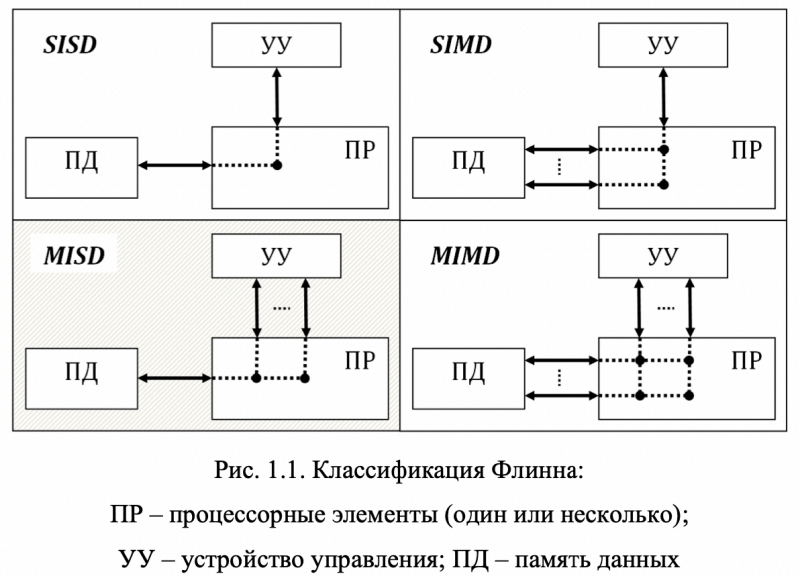
**14. УНИВЕРСАЛЬНЫЕ ВМ**

Универсальные вычислительные машины предназначены для решения широкого круга самых разнообразных математических задач Универсальные вычислительные машины имеют сумматоры, построенные по тем же арифметическим основам, что и счетчики электронных перфорационных вычислительных машин. Универсальная вычислительная машина состоит из ряда устройств - блоков, каждый из которых имеет специальное назначение.

**15. ПРОБЛЕМНО-ОРИЕНТИРОВАННЫЕ ВМ**

Промышленные компьютеры со специальным конструктивным исполнением.

**16. КЛАССИФИКАЦИЯ ФЛИННА**

-SISD (одиночный поток команд и одиночный поток данных. К этому классу относятся однопроцессорные одноядерные машины фон- Неймановского типа.)

-SIMD (одиночный поток команд и множественный поток данных. В архитектурах подобного рода сохраняется один поток команд, включающий, в отличие от предыдущего класса, векторные команды. Это позволяет выполнять одну арифметическую операцию сразу над многими данными – элементами вектора.)

-MISD (множественный поток команд и одиночный поток данных. Определение подразумевает наличие в архитектуре многих процессоров, обрабатывающих один и тот же поток данных. В настоящее время считается, что данный класс пуст)

-MIMD (множественный поток команд и множественный поток данных. Этот класс предполагает, что в вычислительной системе есть несколько процессоров, которые (как правило, асинхронно) выполняют разные команды над разными данными)

**17. ПЕРЕСПЕКТИВЫ ЭВОЛЮЦИИ АРХИТЕКТУРЫ ЭВМ И ВС (ДВА ПУТИ РАЗВИТИЯ)**

Создание ЭВМ пятого поколения традиционной (неймановской) архитектуры и апробации перспективных архитектур и схемотехнических решений. На формальном и прикладном уровнях исследуются архитектуры на основе параллельных абстрактных вычислителей (матричные и клеточные процессоры, систолические структуры, однородные вычислительные структуры, нейронные сети и др.). При этом ЭВМ должна обладать способностью к обучению, производить ассоциативную обработку информации и вести интеллектуальный диалог при решении конкретных задач. Но так же наряду с развитием архитектурных и системотехнических решений ведутся работы по совершенствованию технологий производства интегральных схем и по созданию принципиально новых элементных баз, основанных на оптических, молекулярных, квантовых и нейросетевых технологиях.

Новые архитектуры:

-оптические компьютеры

-квантовые компьютеры

-нейрокомпьютеры

-молекулярные компьютеры

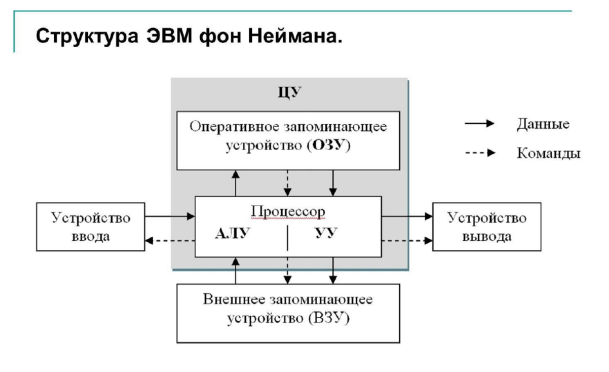
**18. ЗАКОН МУРА**

Согласно закону Мура, удвоение количества транзисторов, размещенных на кристалле, происходит в среднем каждые 2 года. Подобная тенденция способствует тому, что сейчас на одном кристалле можно расположить логические схемы всех компонентов процессора или даже несколько процессоров.

**19. КЛАСС SISD.**

SISD (single instruction stream / single data stream) – одиночный поток команд и одиночный поток данных. К этому классу относятся однопроцессорные одноядерные машины фон Неймановского типа. В таких машинах есть только один поток команд, все команды обрабатываются последовательно друг за другом и каждая команда инициирует одну операцию с одним потоком данных. При этом для увеличения скорости обработки команд и скорости выполнения арифметических операций может использоваться конвейерная обработка. Данный класс включает в себя ЭВМ с CISC и RISC архитектурой, а также ЭВМ с суперскалярной обработкой данных.

**20. АРХИТЕКТУРА ЭВМ ФОН НЕЙМАНА**



-принцип однородности (все данные хранятся в одной и той же памяти)

-принцип адресуемости (процессору в любой момент времени доступна любая ячейка)

-принцип последовательного программного управления (команды выполняются друг за другом в определенной последовательности)

-принцип двоичного кодирования

-принцип жесткости архитектуры (неизменяемость в процессе работы топологии, архитектуры, списка команд)

**21. ПРИНЦИПЫ ПОСТРОЕНИЯ ЭВМ С ХРАНИМОЙ В ПАМЯТИ ПРОГРАММОЙ**

ЭВМ Фон Неймана

**22. ПРИСТОНСКАЯ АРХИТЕКТУРА**

Также необходимо отметить условность принципа однородности памяти по отношению к ЭВМ в целом. Архитектура включает в себя ЦП, оперативку, устройства ввода/вывода и программы хранящиеся в оперативке. В архитектуре все инструкции и данные обрабатываются в одной и той же оперативной памяти. Данные и программы представлены в виде одинаковых структур – последовательности байтов. Последовательное выполнение инструкций.

**23. ГАРВАРДСКАЯ АРХИТЕКТУРА**

Практически одновременно с этим в Гарвардском университете была предложена иная модель, в которой ВМ имела отдельную память команд и отдельную память данных. Такой вид архитектуры называют «гарвардской» архитектурой. На протяжении многих лет принстонская архитектура оставалась основной, но так как она порождает проблемы пропускной способности шины «процессор-память», разработчики ЭВМ все чаще обращаются к гарвардской архитектуре. Например, идеи гарвардской архитектуры реализованы в современных процессорах, когда в кэш-памяти центрального процессора (кэшпамяти первого уровня) выделяется память команд (I-Cache) и память данных (D-Cache)

**24. АЛГОРИТМ. СВОЙСТВА АЛГОРИТМА**

Алгоритм – конечный набор предписаний, определяющий решение задачи посредством конечного количества операций.

-> Дискретность – дискретные действия над дискретной информацией

-> Определенность – действия в алгоритме не должны восприниматься двояко

-> Массовость – применимость алгоритма множеству значений исходных данных

-> Результативность – возможность получения результата за конечное число шагов

**25. ЦЕНТРАЛЬНОЕ УСТРОЙСТВО (ЦУ)**

Центральный̆ процессор (ЦП; англ. Central processing unit, CPU – центральное обрабатывающее устройство) – электронный̆ блок либо интегральная схема (микропроцессор), исполняющая машинные инструкции (код программ), главная часть аппаратного обеспечения компьютера. Можно сказать, что это программно управляемое устройство, которое предназначено для обработки цифровой̆ информации и управления процессом этой обработки.

**26. АРИФМЕТИКО-ЛОГИЧЕСКОЕ УСТРОЙСТВО (АЛУ)**

АЛУ – часть процессора, выполняющая арифметические и логические операции над данными.

-АЛУ состоит из регистров, сумматора и блока управления выполняемым процессором.

-Помимо результата операции АЛУ формирует ряд признаком результата (флагов).

**27. УСТРОЙСТВО УПРАВЛЕНИЯ**

Устройство управления (УУ) координирует взаимодействие различных частей компьютера. Выполняет следующие основные функции:

• формирует и подает во все блоки машины в нужные моменты времени определенные сигналы управления (управляющие импульсы), обусловленные спецификой̆ выполнения различных операций;

• формирует адреса ячеек памяти, используемых выполняемой операцией, и передает эти адреса в соответствующие блоки компьютера;

• получает от генератора тактовых импульсов обратную последовательность импульсов.

**28. ЗУ В ЭВМ КЛАССИЧЕСКОГО ТИПА**

Памятью ЭВМ условимся называть всю совокупность устройств, функционально предназначенных для запоминания, хранения и выдачи информации (в форме числовых и нечисловых кодов). Отдельные устройства, входящие в эту совокупность, условимся называть устройствами памяти или запоминающими устройствами того или иного назначения, или типа.

Содержимое памяти:

1. коды команд (программ);
2. данные (константы, исходные, промежуточные и результаты);
3. адреса (ассоциативные признаки, критерии поиска);
4. состояние устройств ЭВМ, подсистем и вычислительной системы .

Информация, которую необходимо хранить в памяти ЭВМ это программные и числовые массивы слов, а также массивы адресов и состояний.

Основным требованием к ЗУ является надежное хранение информации без изменения ее начального содержания в течение продолжительного времени и своевременное предоставление по запросу.

Основные характеристики любого ЗУ – емкость и быстродействие.

**29. ОЗУ**

➢ ОЗУ(оперативное ЗУ) - быстрое запоминающее устройство не очень большого объема, непосредственно связанное с процессором и предназначенное для записи, считывания и хранения выполняемых программ и данных, обрабатываемых этими программами.

⇨ Можно только считывать и записывать

⇨ Модуль из микросхем на печатной плате. Вставляется в слот на материнской плате

**30. ПЗУ**

➢ ПЗУ (постоянное ЗУ)- энергонезависимое запоминающее устройство, изготовленное в виде микросхемы. Используется для хранения данных, не требующих изменения. Содержание памяти специальным образом "зашивается" в ПЗУ при изготовлении. В ПЗУ находятся программа управления работой самого процессора, программы управления дисплеем, клавиатурой, принтером, внешней памятью, программы запуска и остановки компьютера, тестирования устройств.

⇨ Можно только считывать

⇨ Микросхема, впаянная в материнскую плату

**31. РЕГИСТРЫ**

Регистры – специальные ячейки памяти, расположенные внутри процессора, доступ к которым осуществляется не по адресам, как к основной памяти, а по именам (сверхоперативная память). Они предназначены для временного хранения данных ограниченного размера.

**32. СТРУКТУРА ПРОСТЕЙШЕГО ЦЕНТРАЛЬНОГО УСТРОЙСТВА ЭВМ**

АЛУ + УУ

**33. ВНЕШНЯЯ ПАМЯТЬ**

Внешняя память (ВЗУ) – используется для долговременного хранения больших программ и массивов данных в ВМ.

-«Медленная»  
-«Дешевая»

-Энергозависимая.

**34. ФАЙЛ**

Файл – это идентифицированная совокупность экземпляров полностью описанного в конкретной программе типа данных, находящихся вне программы во внешней памяти и доступных программе посредством специальных операций.

**35. ВНЕШНИЕ УСТРОЙСТВА В ВМ РАЗЛИЧНОГО ТИПА (СПЕЦИАЛИЗИРОВАННЫХ, УНИВЕРСАЛЬНЫХ)**

Внешние устройства (ВУ) – обеспечивают эффективное взаимодействие компьютера с окружающей средой – пользователями, объектами управления, другими машинами.

В **специализированных** управляющих ЭВМ (технологические процессы, связи, ракеты и пр.):

-внешние устройства ввода – датчики (температуры, давления, расстояния и пр.)

-внешние устройства вывода – манипуляторы (гидро-, пневмо-, сервоприводы релей, вентилей и др.)

В **универсальных** ЭВМ (человеко-машинная обработка информации) в качестве ВУ выступают терминалы, принтеры и тп.

**36. АРХИТЕКТУРА ВМ С «НЕПОСРЕДСТВЕННЫМИ СВЯЗЯМИ». ДОСТОИНСТВА И НЕДОСТАТКИ**

Пристонская архитектура - структурная организация ВМ с непосредственными связями между устройствами.

-Совместное использование каналов передачи информации для памяти программ и памяти данных привело к ограничению пропускной способности канала.

-ВМ с непосредственными связями плохо поддавались оптимизации

-не обеспечивает высокий уровень параллелизма

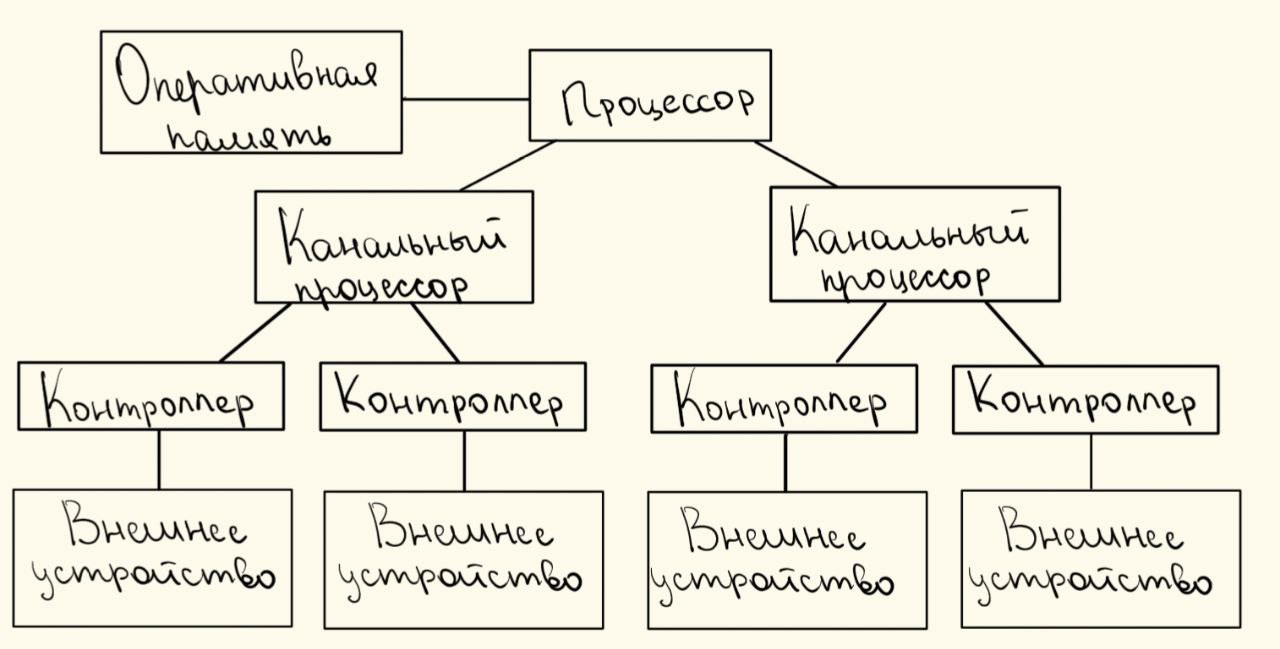
-использование единой памяти может привести к ограничениям в области доступа памяти

+простота (более удобная разработка программного обеспечения)

+эффективность

+универсальность (использование одинаковых ресурсов для разных целей)

**37. «ИЕРАРХИЧЕСКАЯ» АРХИТЕКТУРА ВМ.**



ЦУ соединено с периферийными процессорами, управляющими в свою очередь контроллерами, к которым подключены группы ВУ.

Канальные процессоры.

**38. МАГИСТРАЛЬНАЯ АРХИТЕКТУРА ВМ**

Магистральная архитектура предполагает организацию взаимодействия между процессором, оперативной памятью и внешними устройствами через общий канал связи. Все каналы передачи информации в ЭВМ обычно условно разделяют на две группы:

1. Внутренние – служат для соединения основных компонентов ЭВМ на материнской плате.

2. Внешние – служат для подключения дополнительных устройств, внешних по отношению к материнской плате. Это внешние интерфейсные соединения для подсоединения накопителей информации, устройств ввода-вывода, сетевого оборудования и других устройств.

**39. АРХИТЕКТУРА С ИЕРАРХИЕЙ ШИН**

**1 вариант**: Магистральная шина + несколько отдельных шин для различных устройств.

**2 вариант**: Магистральная шина + дополнительные шины (объединение однотипных устройств ввода/вывода)

-шины «процессор-память»

-шины ввода/вывода

-системные шины:

Подшина данных – происходит обмен информацией между устройствами ВМ.

Подшина адреса – передаются уникальные идентификационные адреса (коды) отдельных устройств, ячеек памяти, номера регистров ЦП, адреса «портов» ввода/вывода.

Подшина управления – передаются специальные служебные сигналы управления (синхронизация работы электронных компонентов ВМ).

**40. ИСТОРИЯ ПОЯВЛЕНИЯ ПК**

История появления персональных компьютеров (ПК) берет свое начало в середине 20 века и имеет множество ключевых моментов. Вот краткий обзор этой истории:

1940-е годы: В 40-х годах XX века, первые концепции электронных компьютеров начали появляться. Проекты, такие как MARK I и ENIAC, были разработаны в США в качестве ранних форм электронных вычислительных машин.

1950-е и 1960-е годы: В этот период произошло значительное развитие электронных компьютеров. Например, в 1950-е годы UNIVAC (Universal Automatic Computer) стал первым коммерчески доступным компьютером, который может быть использован для бизнеса. Этот период также связан с разработкой первых языков программирования, таких как FORTRAN и COBOL.

1970-е годы: В 70-е годы началось формирование рынка для персональных компьютеров. Компании, такие как Apple, IBM и Commodore, начали выпускать персональные компьютеры для потребительского использования. Многие из первых ПК были представлены с различными микропроцессорами, такими как Intel 8080, MOS Technology 6502 и Zilog Z80.

1980-е годы: В этот период произошел настоящий взрыв в индустрии ПК, с появлением компаний, таких как Microsoft и выходом на рынок первых версий операционных систем, таких как MS-DOS. Появление IBM PC и появление стандартизированных аппаратных компонентов также способствовало распространению ПК.

1990-е и 2000-е годы: В эти годы произошли многие технологические прорывы, включая значительное увеличение мощности и улучшение интерфейсов для пользователей. Кроме того, интернет стал все более важным аспектом компьютерной индустрии.

**41. ПРИНЦИП «ОТКРЫТОЙ» АРХИТЕКТУРЫ ВМ**

Открытая архитектура - архитектура компьютера, периферийного устройства или же программного обеспечения, на которые опубликованы спецификации, что позволяет другим производителям разрабатывать дополнительные устройства к системам с такой архитектурой. Примером такой архитектуры является IBM PC.

**42. БЛОЧНО-МОДУЛЬНАЯ КОМПАНОВКА ВМ**

-ПК собирается из отдельных унифицированных блоков.

-Для взаимодействия блоков между собой и с центральным процессором организуется приемно-передающий канал – системная шина.

-Для состыковки блоков между собой имеются специальные системные разъемы, к контактам которых подводятся сигналы системной шины.

Все функционирует на базе технических нормативов, описывающих конструкцию компьютера, рабочие режимы, протоколы обмена данными. Без подобных норм невозможен был бы успех IBM-совместимых компьютеров.

**43. СЕВЕРНЫЙ МОСТ**

Северный мост – системный контролер обеспечивает подключение ЦПУ к узлам, использующие высокопроизводительные шины. Северный мост выполняет функции контроля и направления потока данных из 4-ёх шин: шины связи с процессором или системной, шины связи с памятью, шины связи с графическим адаптером, шины связи с южным мостом.

К северному мосту обычно подключают наиболее производительные периферийные устройства (например, видеокарты), а менее производительные устройства (микросхема BIOS, устройства с шиной PCI) подключаются к «южному мосту» (SouthBridge), который соединяется с северным мостом специальной высокопроизводительной шиной.

**44. ЮЖНЫЙ МОСТ**

Южный мост – периферийный контроллер – содержит контроллеры периферийных устройств, контролеры шин для подключения периферийных устройств, контроллер клавиатуры и мыши. Южный мост отвечает за организацию взаимодействия с медленными компонентами ЭВМ:

➢ Платами расширения

➢ Периферийными устройствами

➢ Устройствами ввода-вывода

➢ Каналами межмашинного обмена и так далее.



**45. ТЕХНОЛОГИЧЕСКИЙ ПРОЦЕСС ПОЛУПРОВОДНИКОВОГО ПРОИЗВОДСТВА**

Технологический процесс полупроводникового производства – технологический процесс изготовления полупроводниковых (п/п) изделий.

Совершенствование технологии и пропорциональное уменьшение размеров п/п структур способствуют улучшению характеристик микросхем, процессоров, микроконтроллеров и т.д.:

-размеров

-энергопотребления

-рабочих частот

-стоимости

**46. СОСТАВ МИКРОПРОЦЕССОРА**

Современный процессор состоит из:

-одного или нескольких ядер, осуществляющих выполнение всех инструкций

-нескольких уровней КЭШ-памяти, ускоряющих взаимодействие процессора с ОЗУ

-контроллера ОЗУ

-контроллера системной шины

-регистры

-шина данных и шина адреса

-АЛУ

-УУ

**47. ХАРАКТЕРИСТИКИ МИКРОПРОЦЕССОРА**

1.Тип микроархитектуры. Например, Intel Core – это многоядерная микропроцессорная архитектура. Процессоры, созданные по этой архитектуре, входят в семейство Core2. Микроархитектуры Nehalem, Sandy Bridge, Ivy Bridge, Haswell, Broadwell, Skylake являются развитием микроархитектуры центральных процессоров Intel Core. Все эти микроархитектуры могут использоваться при создании процессоров различных ценовых диапазонов и для различных устройств, от настольных ПК, до ноутбуков и планшетов. Так, все эти микроархитектуры имеют реализации в процессорах семейства Intel Core i7 для «настольных» ПК.

2.Количество ядер.

3.Наборы инструкций, т.е. набор используемых процессором управляющих кодов и способов адресации данных. Чем больше набор используемых инструкций, тем качественнее и быстрее обрабатываются данные. (SSE, MMX и т.п.).

4.Разрядность – число двоичных разрядов, одновременно обрабатываемых при выполнении одной команды. Большинство современных процессоров – это 32- и 64- разрядные процессоры.

5.Тактовая частота – количество циклов работы устройства за единицу времени. Чем выше тактовая частота, тем выше производительность. В современных ПК этот показатель не превышает 4ГГц.

6. Наличие и размер кэш-памяти.

7. Техпроцесс – технологический процесс изготовления полупроводниковых (п/ п) изделий. При производстве п/п интегральных микросхем применяется фотолитография и литографическое оборудование. Разрешающая способность (в мкм. и нм.) этого оборудования и определяет название применяемого конкретного технологического процесса. Совершенствование технологии и пропорциональное уменьшение размеров п/п структур способствуют улучшению характеристик (размеры, энергопотребление, рабочие частоты, стоимость) полупроводниковых приборов (микросхем, процессоров, микроконтроллеров и т.д.). Так микропроцессоры на базе микроархитектуры Nehalem выпускаются по технологии 45 нм, а Skylake – 14 нм.Помимо всего перечисленного, так же важно обращать внимание на наличие встроенного контроллера памяти и графического ядра, уровень энергопотребления, тип поддерживаемой оперативной памяти и поддерживаемый объем этой памяти.

**48. ЯДРО МИКРОПРОЦЕССОРА (ОСНОВНЫЕ ФУНКЦИОНАЛЬНЫЕ БЛОКИ)**

1. Блок выборки инструкций

2. **Блоки декодирования инструкций –** занимаются декодированием инструкций, т.е. определяют, что надо сделать процессору, и какие дополнительные данные нужны для выполнения инструкции.

3. **Блок выборки данных** осуществляет выборку данных из КЭШ-памяти или ОЗУ, необходимых для выполнения текущих инструкций.

4. **Управляющий блок** на основании декодированных инструкций управляет работой блоков выполнения инструкций, разделяет нагрузку между ними, обеспечивает своевременное и верное выполнение инструкций.

5. **Блоки выполнения инструкций** включают в себя несколько разнотипных блоков: ALU, FPU, MMX, ATA, 3DNow!, AES.

6. **Блоки сохранения результатов** обеспечивают запись результата выполнения инструкции в ОЗУ по адресу, указанному в обрабатываемой инструкции.

7. **Блок работы с прерываниями.** Работа с прерываниями – одна из важнейших задач процессора, позволяющая ему своевременно реагировать на события прерывать ход работы программы и выполнять требуемые от него действия.

8. **ПЗУ, содержащий микрокод.** Микрокод – серия простых инструкций, в совокупности выполняющих то же действие, что и одна сложная инструкция. Набор микрокода прошит в ПЗУ, встроенном в процессоре.

9. **Набор регистров.** Регистры – сверхбыстрая оперативная память небольшого объема (несколько сотен байт), входящая в состав процессора для временного хранения промежуточных результатов выполнения инструкций. (Регистры процессора: регистры общего назначения, специальные регистры).

10. **Счетчик команд –** регистр, содержащий адрес команды, которую процессор начнет выполнять на следующем такте работы.

**49. РАСШИРЕНИЯ НАБОРА ИНСТРУКЦИЙ (SIMD-РАСШИРЕНИЯ АРХИТЕКТУРЫ IA-32)**

SIMD-расширения архитектуры IA-32 (расширения набора инструкций с одновременным выполнением множества данных) предоставляют возможность процессору выполнять одну и ту же операцию над несколькими данными параллельно. Некоторые из таких расширений включают в себя следующие инструкции:

1. MMX (Multimedia Extensions) - расширение, предназначенное для улучшения обработки мультимедийных данных, таких как звук и видео.

2. SSE (Streaming SIMD Extensions) - набор инструкций, который обеспечивает поддержку параллельной обработки множества данных, что позволяет ускорить выполнение операций над векторами и матрицами.

3. AVX (Advanced Vector Extensions) - расширение, которое предоставляет возможность более широкой (широкой) обработки данных в сравнении с SSE.

4. FMA (Fused Multiply-Add) - набор инструкций, позволяющих выполнять операции умножения и сложения для векторных данных за один такт.

Эти расширения позволяют современным процессорам выполнять ряд задач с высокой степенью параллелизма, что улучшает производительность вычислений, особенно в областях связанных с обработкой мультимедийных данных, научными вычислениями и симуляциями.

**50. ПРИНЦИП РАБОТЫ ЯДРА ПРОЦЕССОРА (УПРОЩЕННЫЙ ВИД)**

Принцип работы ядра процессора основан на цикле описанном Джоном фон Нейманом в 1946 году.

Скорость перехода от одного этапа цикла к другому определяется тактовой частотой процессора.

Время работы каждого этапа цикла и время полного выполнения одной инструкции, определяется устройством ядра процессора.

1. Блок выборки инструкций проверяет наличие прерываний. Если прерывание есть, то данные регистров и счетчика команд заносятся в стек, а в счетчик команд заносится адрес команды обработчика прерываний. По окончанию работы функции обработки прерываний, данные из стека будут восстановлены;

2. Блок выборки инструкций из счетчика команд считывает адрес команды, предназначенной для выполнения. По этому адресу из КЭШ-памяти или ОЗУ считывается команда. Полученные данные передаются в блок декодирования;

3. Блок декодирования команд расшифровывает команду, при необходимости используя для интерпретации команды записанный в ПЗУ микрокод. Если это команда перехода, то в счетчик команд записывается адрес перехода и управление передается в блок выборки инструкций (пункт 1), иначе счетчик команд увеличивается на размер команды (для процессора с длинной команды 32 бита – на 4) и передает управление в блок выборки данных;

4. Блок выборки данных считывает из КЭШ-памяти или ОЗУ требуемые для выполнения команды данные и передает управление планировщику;

5. Управляющий блок определяет, какому блоку выполнения инструкций обработать текущую задачу, и передает управление этому блоку;

6. Блоки выполнения инструкций выполняют требуемые командой действия и передают управление блоку сохранения результатов;

7. При необходимости сохранения результатов в ОЗУ, блок сохранения результатов выполняет требуемые для этого действия и передает управление блоку выборки инструкций (пункт 1).

**51. СПОСОБЫ ПОВЫШЕНИЯ ПРОИЗВОДИТЕЛЬНОСТИ ЯДРА ПРОЦЕССОРА**

1. Конвейеризация

2. Суперскалярность

3. Параллельная обработка данных

4. Технология Hyper-Threading

5. Технология Turbo Boost

**52. СУПЕРСКАЛЯРНОСТЬ**

Выполнение нескольких инструкций одновременно.

Суперскалярная архитектура позволяет поддерживать параллелизм на уровне инструкций, т.е. выполнять эти инструкции одновременно (за один такт) благодаря использованию нескольких одинаковых функциональных узлов.

Суперскалярность – архитектура вычислительного ядра, при которой наиболее нагруженные блоки могут входить в процессор в нескольких экземплярах.

Например, в ядре процессора блок выборки инструкций может нагружать сразу несколько блоков декодирования.

-Параллельное выполнение инструкций возможно, только если эти инструкции – независимые.

**53. ПАРАЛЛЕЛЬНАЯ ОБРАБОТКА ДАННЫХ**

Подавляющее большинство современных процессоров имеют два и более ядра.

Параллельные вычисления (параллельная обработка) – это использование нескольких или многих вычислительных устройств для одновременного выполнения разных частей одной программы (одного проекта).

Параллельные вычисления – вычисления, которые можно реализовать на многопроцессорных системах с использованием возможности одновременного выполнения многих действий, порождаемых процессом решения одной или многих задач.

Основная цель параллельных вычислений – уменьшение времени решения задачи. Многие необходимые для нужд практики задачи требуется решать в реальном времени или для их решения требуется очень большой объем вычислений.

Проблемы с приростом производительности:

-не все программы поддерживают распределение вычислений на несколько ядер

-усложняется работа с памятью, так как ядер много, и всем требуется доступ к ОЗУ

-возрастет энергопотребление, увеличивается тепловыделение и требуется мощная система охлаждения

**54. КОНВЕЙЕРИЗАЦИЯ**

Выполнение программ последовательно и параллельно. Может начать выполнять этап, не дожидаясь результатов предыдущего.

Конвейеризация предполагает использование конвейера инструкций. Условно, любой процессор, выполняя короткую программу, совершает следующие шаги:

1. Читает инструкцию.

2. Декодирует инструкцию.

3. Ищет все данные, необходимые для реализации инструкции.

4. Обрабатывает инструкцию.

5. Записывает результат.

Поскольку данные, необходимые для выполнения второго шага могут находиться и в оперативной памяти, и на жестком диске, возникают длительные, с т.з. процессора, задержки, связанные с передачей этих данных процессору. К тому же, каждая инструкция, выполняемая процессором последовательно, проходит все блоки ядра, в каждом из которых совершается своя часть действий, необходимых для выполнения инструкции. Если приступать к обработке новой инструкции только после завершения работы над первой инструкцией, то большая часть блоков ядра процессора в каждый момент времени будет простаивать. Одним из способов нивелировать задержки и снизить время простоя является многопоточное параллельное исполнение инструкций. Так, при конвейеризации по мере освобождения блоков ядра, они загружаются обработкой следующей инструкции, не дожидаясь пока предыдущая инструкция выполнится полностью.

**55. ТЕХНОЛОГИЯ HYPER-THREADING**

Технология Hyper-Threading реализует идеи одновременной многопоточности, когда каждое физическое ядро процессора видится операционной системой, как два логических, что позволяет выполнять две задачи одновременно. В ядре сохраняется состояние сразу двух потоков, у каждого логического ядра есть свой набор регистров, свой счетчик команд и свой блок работы с прерываниями для каждого потока. Однако остальные элементы ядра для обоих потоков – общие, и делятся между ними. Когда по какой-либо причине один из потоков освобождает процессорный блок (исполнительное устройство), другой поток его использует.

**56. ТЕХНОЛОГИЯ TURBO BOOST**

• Процессоры с этой технологией могут сами динамически, на короткий промежуток времени, повышать тактовую частоты, тем самым, увеличивая свою производительность.

• Процессор контролирует все параметры своей работы: напряжение, силу тока, температуру и тд не допуская сбоев и тем более выхода из строя.

Применение этой технологии позволяет значительно поднять производительность процессора, особенно, при работе с однопоточными приложениями. (Производительность большинства современных процессоров в домашних условиях можно немного поднять, попросту говоря разогнать – заставить работать на частотах, превышающих номинальную, т.е. заявленную производителем.

**57. НАПРАВЛЕНИЯ РАЗВИТИЯ АРХИТЕКТУР СОВРЕМЕННЫХ ПРОЦЕССОРОВ**

Развитие архитектуры процессоров идет в нескольких основных направлениях – увеличение числа ядер, оптимизация энергопотребления и совершенствование технологии кэширования.

1.Увеличение числа ядер

2.Увеличение частоты работы процессора

3.Использование специализированных инструкций и ускорителей

4.Оптимизация архитектуры для конкретных приложений

**58. ПРОЦЕССОРЫ CISC**

CISC (Complex instruction set computer) – это процессоры со сложным набором команд. Архитектура CISC характеризуется: сложными и многоплановыми инструкциями; большим набором различных инструкций; нефиксированной длиной инструкций; многообразием режимов адресации.

Выполнение сложных команд за один такт такта процессора, что позволяет выполнять более сложные операции, используя одну команду.

**59. ПРОЦЕССОРЫ RISC С СОКРАЩЕННЫМ НАБОРОМ КОМАНД**

RISK создан для более простых инструкций, что упрощает архитектуру процессора и облегчает его проектирование. Имеют набор базовых и простых команд. Для сложных операций требуется несколько простых команд.

RISC (Reduced Instruction Set Computer) – процессоры с сокращенным набором инструкций. Архитектура RISC характеризуется: фиксированной длиной инструкций; небольшим набором стандартизированных инструкций; большим количеством регистров общего назначения; отсутствием микрокода; меньшим энергопотреблением, по сравнению с CISC - процессорами аналогичной производительности; более простым внутренним устройством; меньшим количеством транзисторов, по сравнению с CISC - процессорами аналогичной производительности; отсутствием сложных специализированных блоков в ядре процессора.

**60. ПРОЦЕССОРЫ MISC С МИНИМАЛЬНЫМ НАБОРОМ КОМАНД**

MISC (Minimal Instruction Set Computer) – дальнейшее развитие архитектуры RISС, основанное на еще большем упрощении инструкций и уменьшении их количества. Так, в среднем, в MISC-процессорах используется 20-30 простых инструкций. Такой подход позволил еще больше упростить устройство процессора, снизить энергопотребление и максимально использовать возможности параллельной обработки данных. Наилучшая эффективность и простота.

**61. ПРОЦЕССОРЫ VLIW С НАБОРОМ СВЕРХДЛИННЫХ КОМАНД**

VLIW (Very long Instruction word) - архитектура процессоров, использующая инструкции большой длины, содержащие сразу несколько операций, объединенных компилятором для параллельной обработки (длина инструкций может достигать 128-256 бит). Архитектура VLIW является дальнейшим усовершенствованием архитектуры RISC и MISC с углубленным параллелизмом.

**62. ПАМЯТЬ ВМ**

Память вычислительной машины - это ключевой элемент компьютерной системы, который используется для хранения данных и программ, необходимых для выполнения различных вычислительных задач. Память компьютера может быть разделена на несколько уровней, каждый из которых обеспечивает разные характеристики и применения.

1. Оперативная память (RAM): оперативная память представляет собой тип памяти, используемый для временного хранения данных и инструкций, которые в настоящее время используются компьютером. Она обеспечивает быстрый доступ к данным, но данные в ней удаляются при выключении устройства.

2. Виртуальная память: виртуальная память - это механизм, который используется компьютерной системой для управления объемом доступной памяти. Она позволяет компьютеру использовать дисковое пространство в качестве расширения оперативной памяти, когда физическая память исчерпана.

3. Кэш-память: это небольшие объемы очень быстрой памяти, используемые для временного хранения данных, которые компьютер часто использует, чтобы сократить время доступа к этим данным и повысить производительность.

**63. ИЕРАРХИЯ ЗАПОМИНАЮЩИХ УСТРОЙСТВ**



**64. ХАРАКТЕРИСТИКИ ЗУ**

Характеристики запоминающего устройства (ЗУ) определяют качество и целесообразность его применения в той или иной вычислительной машине или системе. Основными характеристиками ЗУ являются информационная ёмкость, быстродействие и надёжность.

*Информационная ёмкость ЗУ* определяется количеством двоичных единиц информации (бит), которое может храниться в нём (иногда ёмкость выражается в байтах. Обычно один байт равен восьми битам). Если ЗУ рассчитано на хранение *N* чисел, каждое из которых имеет *р* разрядов, то информационная ёмкость *М = N\*p*.

Возможность решения на ЭВМ той или иной задачи в значительной степени зависит от ёмкости ЗУ машины.

*Быстродействие ЗУ* характеризуется его временными характеристиками, к которым относятся: время обращения к ЗУ при записи и считывании информации, время записи информации, время считывания или выборки информации. Время обращения (время цикла) характеризуем максимальную частоту обращения к данному ЗУ при считывании или записи информации. Время считывания или выборки информации - интервал времени обращения к ЗУ до получения выходного сигнала от подачи сигнала считывания. Время записи информации - интервал времени от момента подачи сигнала обращения к ЗУ до момента готовности информации к считыванию.

*Надёжность ЗУ* определяется числовыми значениями параметров конструктивной и информационной надёжности. Под конструктивной, или элементной, надёжностью понимают вероятность безотказной работы всех элементов или устройства в заданном интервале времени и заданных условиях эксплуатации. Таким образом понятие конструктивной надёжности совпадает с общепринятым определением надёжности радиоэлектронных устройств.

**65. КЛАССИФИКАЦИЯ ЗУ**

Запоминающее устройство (ЗУ), блок вычислительной машины или самостоятельное устройство, предназначенное для записи, хранения и воспроизведения информации. Наибольшее распространение ЗУ получили в цифровых вычислительных машинах.

По устойчивости записи и возможности перезаписи ЗУ делятся на:

* Постоянные ЗУ (ПЗУ), содержание которых не может быть изменено конечным пользователем (например, BIOS). ПЗУ в рабочем режиме допускает только считывание информации.
* Записываемые ЗУ (ППЗУ), в которые конечный пользователь может записать информацию только один раз (например, CD-R).
* Многократно перезаписываемые ЗУ (ПППЗУ) (например, CD-RW).
* Оперативные ЗУ (ОЗУ) обеспечивают режим записи, хранения и считывания информации в процессе её обработки. Быстрые, но дорогие ОЗУ (SRAM) строят на триггерах, более медленные, но дешёвые разновидности ОЗУ — динамические ЗУ (DRAM) строят на конденсаторах. В обоих видах ЗУ информация исчезает после отключения от источника питания (например, тока).

По типу доступа ЗУ делятся на:

* Устройства с последовательным доступом (например, магнитные ленты).
* Устройства с произвольным доступом (RAM) (например, оперативная память).
* Устройства с прямым доступом (например, жесткие магнитные диски).
* Устройства с ассоциативным доступом (специальные устройства, для повышения производительности БД)

По геометрическому исполнению:

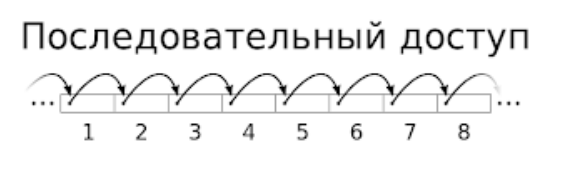
* дисковые (магнитные диски, оптические, магнитооптические);
* ленточные (магнитные ленты, перфоленты);
* барабанные (магнитные барабаны);
* карточные (магнитные карты, перфокарты, флэш-карты, и др.)
* печатные платы (карты DRAM, картриджи).

По форме записанной информации выделяют аналоговые и цифровые запоминающие устройства.

**66. ПОСЛЕДОВАТЕЛЬНЫЙ ДОСТУП К ПАМЯТИ**

Стриминг, ленты

Последовательный доступ означает, что доступ к группе элементов (например, данные в памяти, на диске или на магнитной ленте) осуществляется в заранее заданном порядке. Последовательный доступ иногда является единственным способом обратиться к данным, как, например, к записям на магнитной ленте. Кроме того, иногда это может быть всего лишь одним из методов доступа к данным, например, мы можем предпочесть этот способ если мы хотим обработать последовательность элементов данных по порядку.



**67. ПРЯМОЙ ДОСТУП К ПАМЯТИ**

Доступ к элементам жесткого диска

Каждая запись имеет уникальный адрес, отражающий ее фи­зическое размещение на носителе информации. Обращение осуществляется как адресный доступ к началу записи, с последующим последовательным досту­пом к определенной единице информации внутри записи. В результате время доступа к определенной позиции является величиной переменной. Такой ре­жим характерен для магнитных дисков.

**68. ПРОИЗВОЛЬНЫЙ ДОСТУП К ПАМЯТИ**

Произвольным доступом ( также называемым случайным доступом или прямым доступом, random access) понимают возможность обратиться к любому элементу последовательности за равные промежутки времени, не зависящие от размеров последовательности (в отличие от последовательного доступа, когда чем дальше расположен элемент, тем больше требуется времени для доступа).



**69. АССОЦИАТИВНЫЙ ДОСТУП К ПАМЯТИ**

Этот вид доступа позволяет выполнять поиск ячеек, содержащих такую информацию, в которой значение отдельных битов совпадает с состоянием одноименных битов в заданном образце. Сравнение осуще­ствляется параллельно для всех ячеек памяти, независимо от ее емкости. По ассоциативному принципу построены некоторые блоки кэш-памяти.

Способ доступа к данным в памяти, при котором операция чтения и записи может быть выполнена путем указания сопоставления адреса на хранимые данные. Адрес данных ассоциируется с реальным адресом памяти. Позволяет осуществлять доступ к данным не зная их фактического адреса.

**70. ОСНОВНАЯ ПАМЯТЬ**

Основная память (ОП) представляет собой единственный вид памяти, к которой ЦП может обращаться непосредственно (исключение составляют лишь регистры Центрального процессора). Информация, хранящаяся на внешних ЗУ, становится доступной процессору только после того, как будет переписана в основную память.

Основную память образуют запоминающие устройства с произвольным досту­пом. Такие ЗУ образованы как массив ячеек, а «произвольный доступ» означает, что обращение к любой ячейке занимает одно и то же время и может производиться произвольной последовательности. Каждая ячейка содержит фиксированное число поминающих элементов и имеет уникальный адрес, позволяющий различать ячейки при обращении к ним для выполнения операций записи и считывания.

Основная память может включать в себя два типа устройств: оперативные запоминающие устройства (ОЗУ) и постоянные запоминающие устройства (ПЗУ).

Преимущественную долю основной памяти образует ОЗУ, называемое оперативным, потому что оно допускает как запись, так и считывание информации, причем обе операции выполняются однотипно, практически с одной и той же скоростью, и производятся с помощью электрических сигналов. В англоязычной литературе ОЗУ соответствует аббревиатура RAM — Random Access Метощ, то есть «память с произвольным доступом», что не совсем корректно, поскольку памятью с произвольным доступом являются также ПЗУ и регистры процессора. Для большинства типов полупроводниковых ОЗУ характерна энергозависимость даже при кратковременном прерывании питания хранимая информация теряется. Микросхема ОЗУ должна быть постоянно подключена к источнику питания и поэтому может использоваться только как временная память.

Вторую группу полупроводниковых ЗУ основной памяти образуют энергоне­зависимые микросхемы ПЗУ (ROM — Read-Only Memory). ПЗУ обеспечивает счи­тывание информации, но не допускает ее изменения (в ряде случаев информация в ПЗУ может быть изменена, но этот процесс сильно отличается от считывания и требует значительно большего времени).

**71. СИНХРОННЫЕ ЗУ**

Синхронные запоминающие устройства (СЗУ) – это устройства, используемые для хранения информации в цифровых системах. Они работают на основе синхронных технологий и используют тактовый сигнал для синхронизации операций записи и чтения.

СЗУ имеют быстрый доступ к данным и широко используются в процессорах, компьютерах и других электронных устройствах. Они обеспечивают быстрый доступ к данным, что делает их идеальными для кэш-памяти и регистров процессоров.

Однако, СЗУ обычно имеют ограниченную емкость по сравнению с другими типами памяти, такими как динамическая оперативная память (DRAM) или флэш-память.

**72. АСИНХРОННЫЕ ЗУ**

Асинхронный принцип предполагает, что момент начала очередного действия определяется моментом завершения предшествующей операции. Перенося этот принцип на систему памяти, необходимо принимать во внимание, что контроллер памяти всегда работает синхронно. В асинхронных ЗУ цикл чтения начинается только при поступлении запроса от контроллера памяти, и если память не успевает выдать данные в текущем такте, контроллер может считать их только в следующем такте, поскольку очередной шаг контроллера начинается с приходом очередного тактового импульса. В последнее время асинхронная схема активно вытесняется синхронной.

**73. ДИНАМИЧЕСКАЯ ПАМЯТЬ DRAM**

Память в виде массивов конденсаторов.

Динамическая RAM (DRAM) - это тип полупроводниковой памяти, используемой в компьютерах и других электронных устройствах для хранения данных, которая отличается от статической RAM (SRAM) и построена по-другому.

DRAM использует конденсаторы для хранения каждого бита информации. Каждый бит хранится в виде заряда в конденсаторе. Периодически, данные в DRAM должны быть обновлены, так как конденсаторы теряют свой заряд из-за утечек, что делает DRAM более медленной, по сравнению с SRAM. Обновление происходит автоматически несколько раз в секунду, иначе данные могут быть утрачены.

Основное преимущество DRAM заключается в более высокой плотности упаковки данных, что позволяет создавать модули памяти большей емкости по сравнению с SRAM. Она обычно используется в оперативной памяти компьютеров (RAM), так как предоставляет более высокую емкость по сравнению с SRAM и более низкую стоимость производства.

Таким образом, DRAM является наиболее широко используемым типом памяти в компьютерах и других устройствах, основанных на полупроводниковых технологиях.

**74. СТАТИЧЕСКАЯ ПАМЯТЬ SRAM**

Статическая RAM (SRAM) - это тип полупроводниковой памяти, используемый для хранения данных в компьютерах и других электронных устройствах. Она отличается от динамической памяти (DRAM) своей способностью сохранять данные без необходимости периодической перезаписи, что делает ее быстрее, но более дорогой и вместимость этого типа памяти для хранения данных обычно меньше, чем у DRAM.

SRAM использует бистабильные триггеры для хранения каждого бита информации. Это означает, что данные сохраняются, пока питание подается на устройство, и не требуется освежение для их сохранения, как в DRAM. Это обеспечивает более быстрый доступ к данным и более низкий уровень задержки по сравнению с DRAM.

SRAM обычно используется для кэш-памяти процессоров и в других ситуациях, где требуется быстродействие и низкая задержка. Она имеет широкое применение в электронике, включая микроконтроллеры, микропроцессоры, кэши уровней 2 и 3 в процессорах, и другие интегральные микросхемы.

Память в виде массивов триггеров.

**75. КЭШ-ПАМЯТЬ**

Между ОП и процессором размещается небольшая, но быстродействующая буферная память, куда в процессе работы копируются те участки ОП, к которым производится обращение со стороны процессора. Выигрыш достигается за счет ранее рассмотренного свойства локальности – если скопировать содержимое участка ОП в более быстродействующую буферную память и переадресовать на нее все обращения в пределах скопированного участка, то можно существенно сократить среднее время доступа к информации. Упомянутая буферная память получила название кэш-память.

В общем виде использование кэш-памяти (КП) поясним следующим образом. Когда ЦП пытается прочитать слово из основной памяти, сначала осуществляется поиск копии этого слова в КП. Если такая копия существует, обращение к ОП не производится, а в ЦП передается слово, извлеченное из кэш-памяти. Данную ситуацию принято называть успешным обращением или попаданием (hit). При отсутствии слова в кэш-памяти, то есть при неуспешном обращении (промахе miss), блок данных, содержащий это слово, сначала пересылается из ОП в кэш-память, а затем уже из КП затребованное слово передается в ЦП.

**76. ЛАТЕНТНОСТЬ ПАМЯТИ. ТАЙМИНГИ**

Латентность памяти и тайминги относятся к характеристикам оперативной памяти компьютера. Давайте разберем каждый из этих терминов более подробно:

**Латентность памяти:**

Латентность памяти представляет собой задержку между запросом на доступ к определенной части памяти и временем, когда эта информация действительно становится доступной для использования. Это время измеряется в тактах частоты памяти (чаще всего в наносекундах), и чем меньше значение латентности, тем быстрее будет доступ к памяти. Часто латентность памяти может быть обозначена как последовательность цифр - например, CL16 или CL19, где CL означает CAS (Column Address Strobe) Latency, или латентность столбца.

**Тайминги памяти:**

Тайминги памяти - это специфические параметры, описывающие задержки и частоту работы модуля оперативной памяти. Они включают в себя различные параметры, такие как CAS Latency, tRCD (Row Address to Column Address Delay), tRP (Row Precharge Time) и tRAS (Row Active Time). Каждый из этих параметров определяет различные аспекты задержек доступа к памяти и управления ею.

Оптимальные значения таймингов зависят от конкретного типа памяти и параметров работы системы, и могут быть настроены в BIOS компьютера для достижения наилучшей производительности оперативной памяти. затрачиваемое памятью на чтение данных.

**77. ПРИНЦИПЫ ОБМЕНА ДАННЫМИ В ЭВМ**

Обмен данных в электронно-вычислительных машинах (ЭВМ) основан на нескольких принципах:

1. Хранение данных: Данные хранятся в памяти ЭВМ, которая может быть как оперативной (RAM), так и постоянной (например, жесткий диск или флэш-память).

2. Ввод данных: Данные могут быть введены в ЭВМ с помощью различных устройств, таких как клавиатура, мышь, сканер или другие устройства ввода.

3. Обработка данных: После ввода данные могут быть обработаны с использованием процессора ЭВМ. Это может включать выполнение арифметических операций, логических операций, обработку данных в соответствии с программой и т.д.

4. Хранение результатов: Результаты обработки данных могут быть сохранены в памяти ЭВМ или переданы на устройства вывода для отображения или сохранения.

5. Вывод данных: Результаты обработки данных могут быть выведены на экран, напечатаны на принтере или переданы на другие устройства вывода.

6. Обмен данными между устройствами: Для обмена данными между различными устройствами (например, между процессором и памятью, между памятью и устройствами ввода/вывода) используются различные интерфейсы и протоколы обмена данными, такие как шины данных, протоколы передачи данных по сети и т.д.

Эти принципы обмена данных обеспечивают основу работы ЭВМ и позволяют им выполнять различные задачи, начиная от обработки текстовой информации до выполнения сложных вычислений.

**78. ВНУТРЕННИЕ И ВНЕШНИЕ КАНАЛЫ ПЕРЕДАЧИ ИНФОРМАЦИИ В ВМ**

Магистральная архитектура предполагает организацию взаимодействия между процессором, оперативной памятью и внешними устройствами через общий канал связи.

Все каналы передачи информации в ЭВМ обычно условно разделяют на две группы:

1. Внутренние – служат для соединения основных компонентов ЭВМ на материнской плате.

2. Внешние – служат для подключения дополнительных устройств, внешних по отношению к материнской плате. Это внешние интерфейсные соединения для подсоединения накопителей информации, устройств ввода-вывода, сетевого оборудования и других устройств.

**79. ШИНЫ ПОСЛЕДОВАТЕЛЬНЫЕ И ПАРАЛЛЕЛЬНЫЕ**

Операции по шине называются транзакциями, например, транзакции чтения-записи, ввода-вывода, если в обмене участвует устройство ввода-вывода. Транзакция включает в себя: посылку адреса или прием (посылку) данных.

В зависимости от способа передачи информации по шине различают шины последовательные и параллельные. Последовательная шина состоит из одной линии данных, при этом данные передаются по этой линии последовательно (бит за битом). Параллельная шина содержит несколько линий. Их число трактуется как разрядность («ширина») шины, которая определяет количество бит, параллельно (одновременно) проходящих через шину.

Количество информации, передаваемое по шине в единицу времени, называется ее пропускной способностью и измеряется в Мбит/с.

**80. ВНЕШНИЕ ИНТЕРФЕЙСНЫЕ СОЕДИНЕНИЯ**

Внешние интерфейсные соединения представляют собой способы подключения электронных устройств к другим устройствам или к внешним средам. Эти соединения позволяют обмениваться данными, энергией или управлять устройствами. Вот некоторые из наиболее распространенных внешних интерфейсных соединений:

1. USB (Universal Serial Bus): USB-порты используются для подключения различных устройств, таких как клавиатуры, мыши, принтеры, флэш-накопители, внешние жесткие диски и т.д. USB также используется для зарядки мобильных устройств.

2. HDMI (High Definition Multimedia Interface): Этот интерфейс используется для передачи аудио- и видеосигналов высокой четкости между устройствами, такими как компьютеры, мониторы, телевизоры, проекторы и др.

4. Ethernet: Этот интерфейс используется для соединения компьютеров с локальной сетью или Интернетом через сетевой кабель.

5. Bluetooth: Bluetooth - беспроводной интерфейс, который позволяет устройствам обмениваться данными на небольшие расстояния. Он широко используется для подключения клавиатур, мышей, наушников, аудиосистем и других устройств к компьютерам и мобильным устройствам.

6. Wi-Fi: Wi-Fi - беспроводной интерфейс, который позволяет устройствам подключаться к беспроводным сетям для доступа в Интернет или обмена данными.

**81. ПРЕРЫВАНИЯ АППАРАТНЫЕ И ПРОГРАММНЫЕ**

**Прерывание** - сигнал, сообщающий процессору о наступление какоголибо события. При этом выполнение текущей последовательности команд приостанавливается и управление передается обработчику прерываний, который выполняет работу по обработке события и возвращает управление в прерванный код.

**Аппаратное прерывание** - событие, генерируемое внешним по отношению к процессору устройством.

**Программное прерывание** — синхронное прерывание, которое может осуществить программы с помощью специальной инструкции. Программные прерывания не являются "истинными" прерываниями. Они возникают при выполнении определенной команды процессора и применяются в том случае, когда необходимо выполнить некоторые привилегированные действия (например обратится к порту компьютера).

**82. ПРЯМОЙ ДОСТУП К ПАМЯТИ (Direct Memory Access – DMA)**

Прямой доступ к памяти (DMA – Direct Memory Access). Режим обмена данными между устройствами компьютера или же между устройством и основной памятью без участия центрального процессора (ЦП). Используется для быстрой передачи больших объемов данных.

**83. ЗАХВАТ ШИНЫ (BUS MASTERING)**

В шинах PCI и еѐ концептуальных производных (AGP и PCI-X), используется иная реализация DMA – захват шины (bus mastering). Эти шины позволяют любому устройству заявить о возникновении потребности к захвату шины, таковая потребность удовлетворяется т. н. арбитром при первой возможности. Устройство, успешно осуществившее захват шины, самостоятельно выставляет на шину сигналы адреса и управления и исполняет в течение какого-то времени ту же ведущую роль на шине, что и ЦП. Доступ ЦП к шине при этом кратковременно блокируется.

Это способ обмена данными между устройствами и ЦП. Устройство может непосредственно управлять обменом данными на шине без необходимости физического вмешательства ЦП. Устройство может самостоятельно инициировать обмен данными с памятью или другими устройствами. Уменьшая нагрузку на процессор и улучшая производительность системы (функциональна для периферийных устройств).

**84. ТЕХНОЛОГИЯ PLUG AND PLAY**

Устройства, подключаемые по шине PCI – самонастраиваемые. Т.е. работают с использованием технологии Plug and Play, позволяющей определять и конфигурировать устройства в ЭВМ без участия пользователя. Каждое PCI-устройство имеет конфигурационное пространство в 256 байт с информацией о самом устройстве и о ресурсах, занимаемых этим устройством. После включения компьютера системное ПО сканирует настройки PCI-устройств и распределяет ресурсы, настраивает прерывания (IRQ). Обычно, система выделяет каждому устройству собственное прерывание, но если это не удается, предоставляет возможность работать с одной линией запроса прерывания нескольким устройствам.

**85. КЛАСС SIMD**

SIMD (single instruction stream / multiple data stream) – одиночный поток команд и множественный поток данных. В архитектурах подобного рода сохраняется один поток команд, включающий, в отличие от предыдущего класса, векторные команды. Это позволяет выполнять одну арифметическую операцию сразу над многими данными – элементами вектора.

**86. ВЕКТОРНО-КОНВЕЙЕРНЫЕ ВС. ИЕРАРХИЧЕСКАЯ СТРУКТУРА**

Векторно-конвейерные вычислительные системы относятся к классу SIMD-систем. Основные принципы, заложенные в архитектуру векторно-конвейерных систем:

· конвейерная организация обработки потока команд;

· введение в систему команд набора векторных операций, которые позволяют оперировать с целыми массивами данных.

Длина обрабатываемых векторов в современных векторно-конвейерных системах составляет, как правило, 128 или 256 элементов. Основное назначение векторных операций состоит в распараллеливании выполнения операторов цикла, в которых обычно сосредоточена большая часть вычислительной работы.

Первый векторно-конвейерный компьютер Cray-1 появился в 1976 году. Архитектура этого компьютера оказалась настолько удачной, что он положил начало целому семейству компьютеров.

**Современные векторно-конвейерные системы имеют иерархическую структуру:**

· на нижнем уровне иерархии расположены конвейеры операций (например, конвейер (pipeline) сложения вещественных чисел, конвейер умножения таких же чисел и т.п.);

· некоторая совокупность конвейеров операций объединяется в конвейерное функциональное устройство;

· векторно-конвейерный процессор содержит ряд конвейерных функциональных устройств;

· несколько векторно-конвейерных процессоров (2-16), объединенных общей памятью, образуют вычислительный узел;

· несколько таких узлов объединяются с помощью коммутаторов, образуя либо NUMA-систему либо MPP-систему.

Типичными представителями такой архитектуры являются компьютеры CRAY J90/T90, CRAY SV1, NEC SX-4/SX-5. Уровень развития микроэлектронных технологий не позволяет в настоящее время производить однокристальные векторно-конвейерные процессоры, поэтому эти системы довольно громоздки и чрезвычайно дороги.

Каждая часть конвейера операций называется ступенью конвейера операций, а общее число ступеней - длиной конвейера операций.

**87. КОНВЕЙЕРНОЕ ФУНКЦИОНАЛЬНОЕ УСТРОЙСТВО**

Конвейерное функциональное устройство – это принцип организации и управления выполнением инструкций процессором. Конвейер позволяет процессору одновременно выполнять несколько стадий обработки инструкций, что повышает производительность путем параллельной обработки нескольких инструкций.

Конвейерное-функциональное устройство разделяет выполнение инструкций на несколько этапов: извлечение инструкций из памяти, декодирование, выполнение, запись результата. Процессор выполняет разные стадии выполнения различных инструкций одновременно.

**88. КОНВЕЙЕР КОМАНД И КОНВЕЙЕР ОПЕРАЦИЙ**

**Конвейер команд**

Этапы выполнения команды:

1. Определение адреса команды (ОАК),(изменение счетчика команд).
2. Выборка команды (ВК), (в регистр команд из ячейки команды и кэш-памяти поступает код команды).
3. Дешифрация команды (ДКО).
4. Определение адреса операнда (ОАО).
5. Выборка операнда (ВО).
6. Арифметическая или логическая операция (АЛО).
7. Запись результатов (ЗР).

Эффективность конвейера тем выше, чем набольшее число этапов разбито выполнение команды. Наличие переходов, а особенно условных переходов, сразу нарушает работу конвейера, приводя его к холостым пробегам.

В современных конвейерах используется конвейеры команд и конвейеры арифметических операций.

**Конвейер операций**

Каждая часть конвейера операций называется *ступенью конвейера операций*, а общее число ступеней - *длиной конвейера операций*.

Эффективность конвейера тем выше:

- чем больше массив обрабатываемых данных;

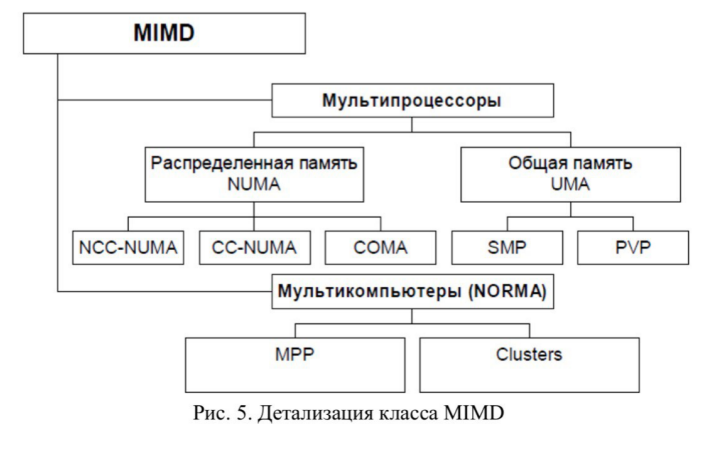
- чем набольшее число операционных блоков разбит процессор.

**89. ВЕКТОРНО-ПАРАЛЛЕЬНЫЕ ВС**

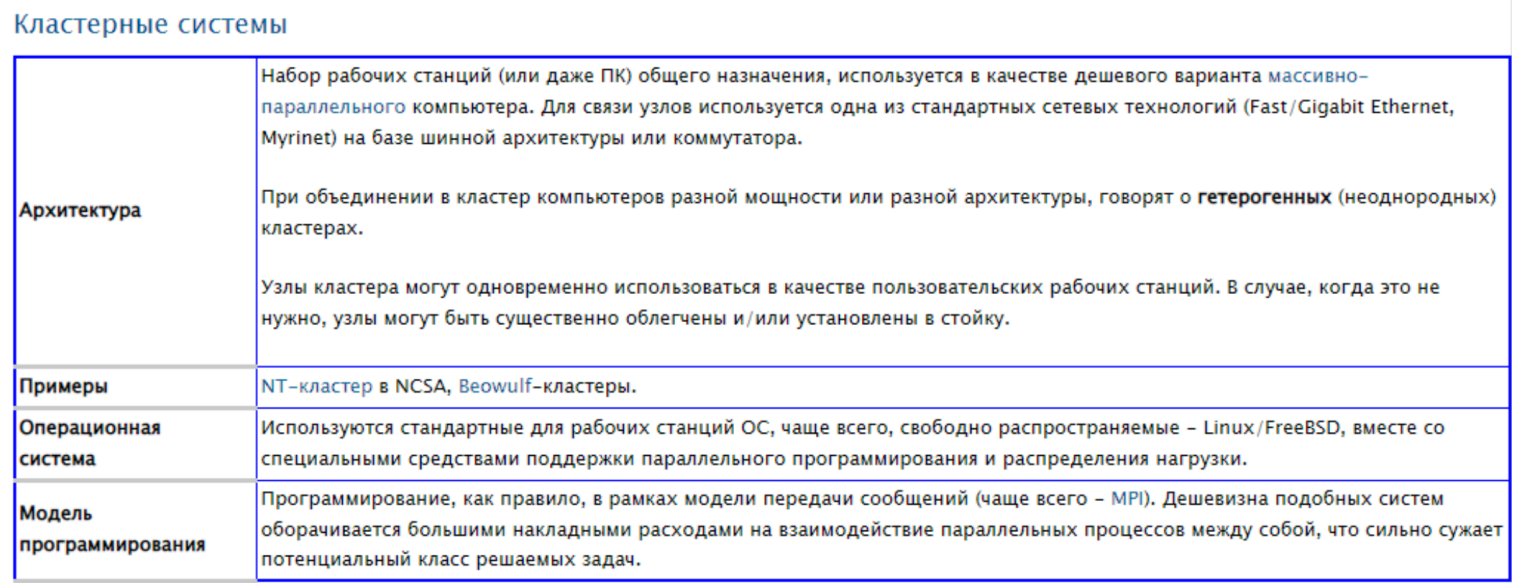


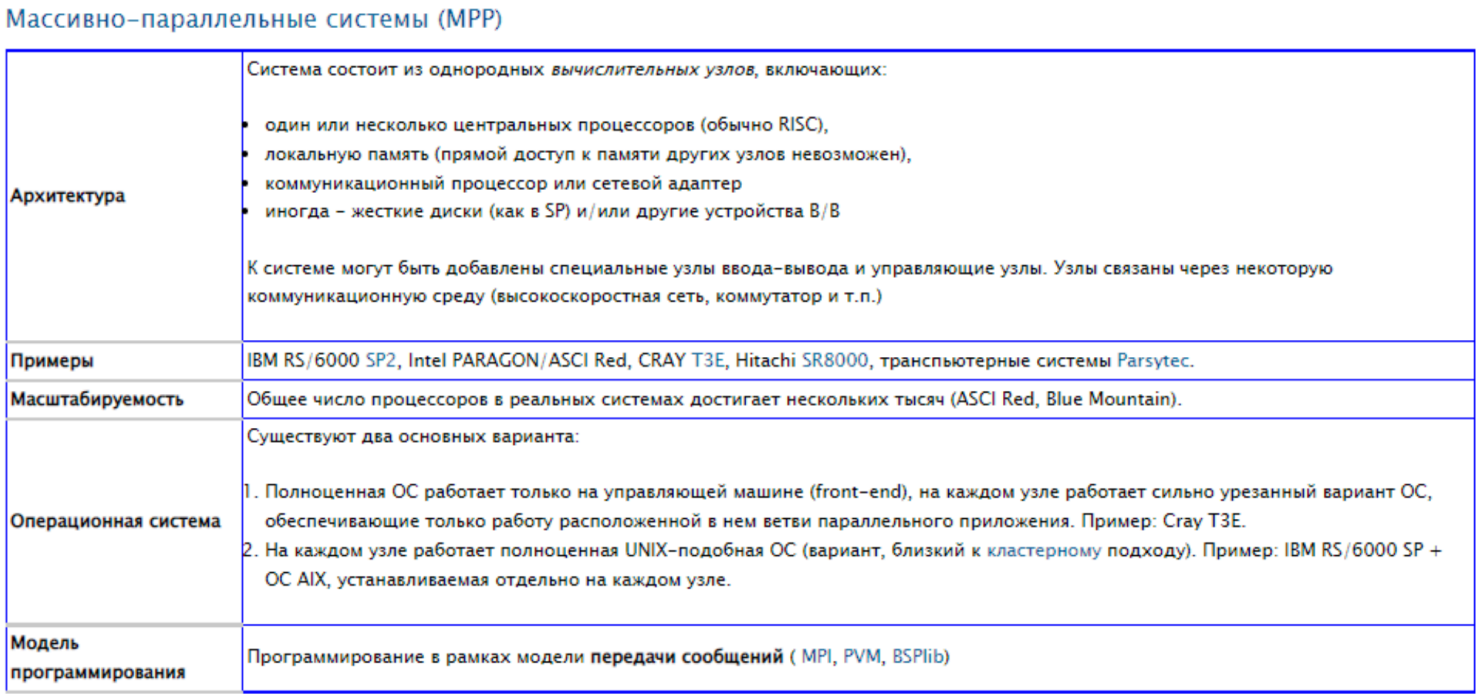
**90. КЛАСС МIMD**

MIMD (multiple instruction stream / multiple data stream) – множественный поток команд и множественный поток данных. Этот класс предполагает, что в вычислительной системе есть несколько процессоров, которые (как правило, асинхронно) выполняют разные команды над разными данными.

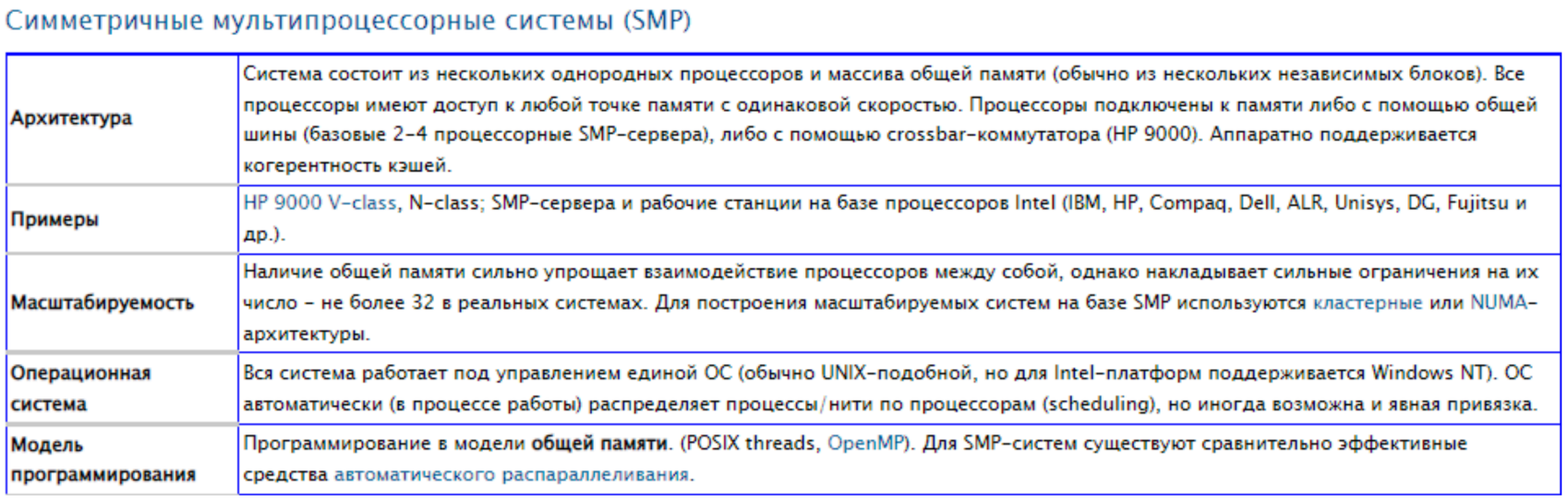


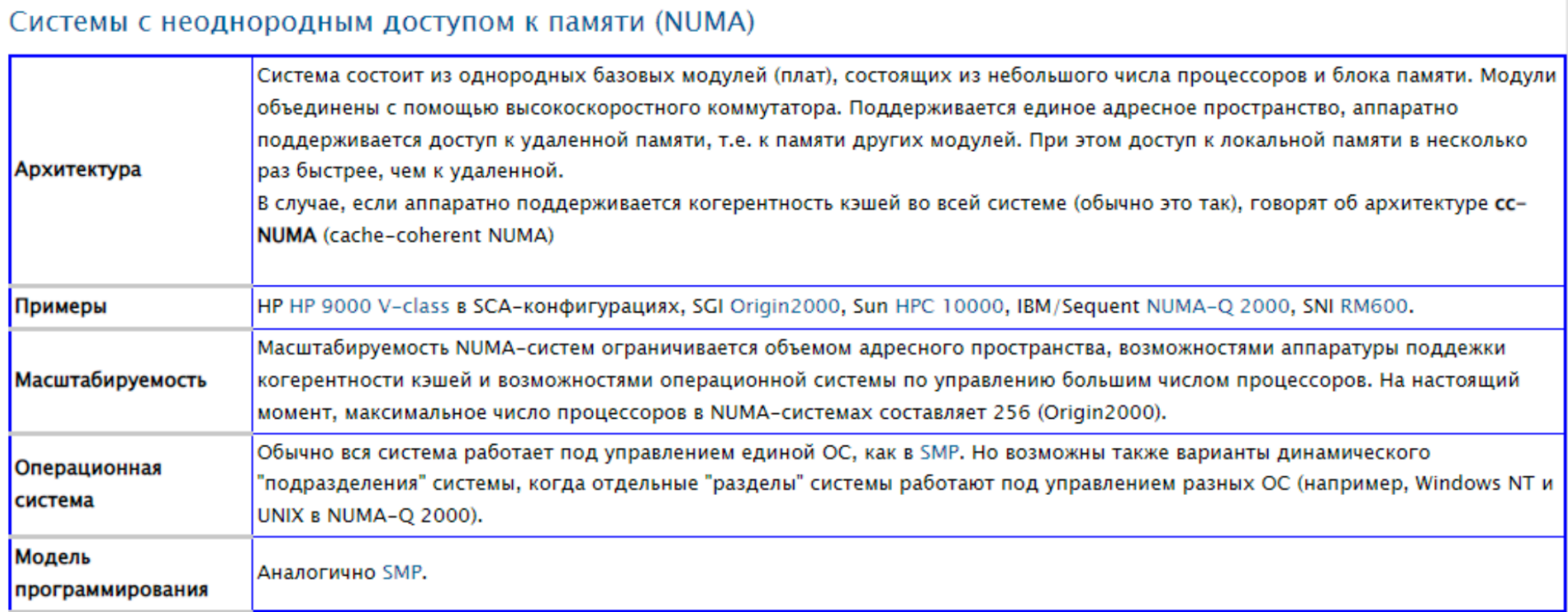
**91. МУЛЬТИКОМПЬЮТЕРЫ**

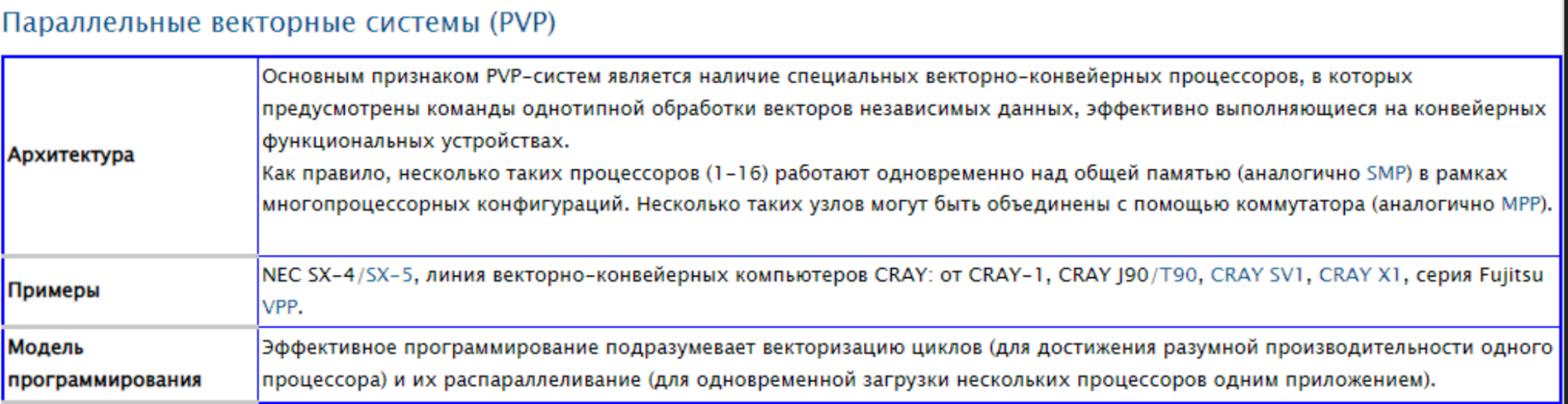




**92. МУЛЬТИПРОЦЕССОРЫ**







**93. NUMA. UMA. NORMA.**

**UMA**



В основе самых простых мультипроцессоров лежит одна шина. Если процессору нужно считать информацию из памяти, он проверяет, свободна ли шина.

Если шина занята, процессор ждет. При наличии двух или трех процессоров доступ к шине вполне управляем. При наличии большего числа процессоров (32, 64 и т.д.) производительность системы полностью ограничивается пропускной способностью шины, а большинство процессоров будут простаивать.

Для разрешения этой проблемы необходимо добавить кэш-память к каждому процессору. Поскольку теперь слова можно получать из кэш-памяти, то и движения в шине будет меньше и система сможет поддержать большее число процессоров. Следующая разработка – каждый процессор имеет не только кэш- память, но и свою локальную память, к которой он получает доступ через локальную шину.

Каждый элемент кэш-памяти может находится в одном из четырех состояний(протокол MESI):

1. Invalid – элемент кэш-памяти содержит недействительные данные.

2. Shared – несколько кэшей могут содержать данную строку, основная память обновлена.

3. Exclusive – никакой другой кэш не содержит эту строку, основная память обновлена.

4. Modified – элемент действителен, основная память недействительна, копий элемента не существует.

UMA-мультипроцессоры с перекрестной коммутацией Имеются 4 адресные шины, которые используются для отслеживания строк в кэш- памяти. Каждая шина использует 1⁄4 часть физического адресного пространства. Для выбора шины используется 2 адресных бита.

UMA-мультипроцессоры с многоступенчатой коммутацией Коммутатор содержит два входа и два выхода. Сообщения, приходящие на один из входов могут переключаться на любой выход. Коммутаторы 2 х 2 можно компоновать различными способами и получать многоступенчатые сети.

**NUMA**

****

Мультипроцессоры NUMA (Non Uniform Memory Access – с неоднородным доступом к памяти). Как и мультипроцессоры UMA, они обеспечивают единое адресное пространство для всех процессоров, но в отличие от машин UMA, доступ к локальным модулям памяти происходит быстрее, чем к удаленным. Следовательно, все программы UMA будут работать без изменений на машинах NUMA, но производительность будет хуже, чем на машинах UMA с той же тактовой частотой.

Машины NUMA три ключевые характеристики, которыми все они обладают и которые в совокупности отличают их от других мультипроцессоров.

1. Существует одно адресное пространство, видимое для всех процессоров.

2. Доступ к удаленной памяти осуществляется с использованием команд load и store.

3. Доступ к удаленной памяти происходит медленнее, чем доступ к локальной памяти.

**NORMA (No Remote Memory Access)** – Архитектура без прямого доступа к удаленной памяти.

Суперкомпьютер короче

Отдельная ОС на каждом узле, из-за этого легче масштабировать.Так как нет проблемы с узким горлышком, у каждого узла своё адресное пространство, поэтому шина адреса не перегрузится.

Есть MPP и Кластеры, кластеры дешевле, MPP лучше масштабируются.

**94. КЛАСТЕРНЫЕ И ОТКАЗОУСТОЙЧИВЫЕ СИСТЕМЫ**

Классификация кластеров:

• Однородные кластерные системы

• Гетерогенные кластерные системы

Узлы вычислительного кластера:

• Полнофункциональные компьютеры

• Не полнофункциональные компьютеры

Преимущества кластеризации:

• Абсолютная масштабируемость

• Наращиваемая масштабируемость

• Высокий коэффициент готовности

• Превосходное соотношение цена/производительность

Кластер – это группа взаимно соединенных вычислительных систем (узлов), работающих совместно, составляя единый вычислительный ресурс и создавая иллюзию наличия единственной ВМ.

• В качестве узла кластера может выступать как однопроцессорная ВМ, так и ВС типа SMP или MPP.

• Каждый узел в состоянии функционировать самостоятельно и отдельно от кластера.

• Архитектура кластерных вычислений сводится к объединению нескольких узлов высокоскоростной сетью.

• Неотъемлемая частью кластера специализированное ПО.

**95. ВЫЧИСЛИТЕЛЬНЫ КЛАСТЕРЫ (HP)**

Вычислительный кластер - это набор компьютеров (вычислительных узлов), объединенных некоторой коммуникационной сетью. Каждый вычислительный узел имеет свою оперативную память и работает под управлением своей операционной системы. Наиболее распространенным является использование однородных кластеров, то есть таких, где все узлы абсолютно одинаковы по своей архитектуре и производительности.

**96. ОТКАЗОУСТОЙЧЕВЫЕ КЛАСТЕРЫ (HA)**

Отказоустойчивый кластер – это группа независимых компьютеров, которые работают совместно в целях повышения доступности и масштабируемости кластерных ролей (ранее называемых кластерными приложениями и службами). Кластерные сервисы (называемые «узлы») соединены физическими кабелями и программным обеспечением. При сбое на одном узле кластера его функции немедленно передаются другим узлам (этот процесс называется отработкой отказа). Кроме того, за кластерными ролями ведется профилактическое наблюдение, чтобы обеспечить их правильную работу. Если они не работают, выполняется перезагрузка или перемещение на другой узел.

**97. КЛАСТЕРЫ С БАЛАНСИРОВКОЙ НАГРУЗКИ (Load balancing clusters)**

Кластер с балансировкой нагрузки представляет собой совокупность серверов, компьютеров или других вычислительных узлов, объединенных в единую систему, которая автоматически распределяет приходящие запросы или задачи между узлами с целью равномерного распределения рабочей нагрузки. Это позволяет обеспечить более эффективное использование ресурсов и предотвратить перегрузки на отдельных серверах или узлах. Кластер с балансировкой нагрузки обычно используется для повышения отказоустойчивости и производительности систем, таких как веб-серверы, базы данных и приложения, где высокая надежность и быстродействие играют важную роль. Такой подход также способствует масштабируемости системы и улучшению общего качества обслуживания пользователей.

**98. ПРЕИМУЩЕСТВА КЛАСТЕРЗАЦИИ**

**Преимущества кластеризации**

* Абсолютная масштабируемость. Возможно создание больших кластеров, превосходящих по вычислительной мощности даже самые производительные одиночные ВМ. Кластер в состоянии содержать десятки узлов, каждый из которых представляет собой мультипроцессор.
* Наращиваемая масштабируемость. Кластер строится так, что его можно наращивать, добавляя новые узлы небольшими порциями. Таким образом, пользователь может начать с умеренной системы, расширяя ее по мере необходимости.
* Высокий коэффициент готовности. Поскольку каждый узел кластера — самостоятельная ВМ или ВС, отказ одного из узлов не приводит к потере работоспособности кластера. Во многих системах отказоустойчивость автоматически поддерживается программным обеспечением.
* Превосходное соотношение цена/производительность. Кластер любой производительности можно создать, соединяя стандартные «строительные блоки», при этом его стоимость будет ниже, чем у одиночной ВМ с эквивалентной вычислительной мощностью.

**99. ТИПЫ КОММУНИКАЦИОННЫХ СЕТЕЙ В ТИПОВОЙ КОНФИГУРАЦИИ КЛАСТЕРА**

**100. ТОПОЛОГИИ КЛАСТЕРОВ (КЛАСТЕРНЫХ ПАР)**

Топология кластерных пар находит применение при организации двух- или четырехузловых кластеров.

Узлы группируются попарно. Дисковые массивы присоединяются к обоим узлам пары, причем каждый узел имеет доступ ко всем дисковым массивам своей пары. Один из узлов является резервным для другого.

Четырехузловая кластерная «пара» представляет собой простое расширение двухузловой топологии. Обе кластерные пары с точки зрения администрирования и настройки рассматриваются как единое целое.

Эта топология подходит для организации кластеров с высокой готовностью данных, но отказоустойчивость реализуется только в пределах пары, так как принадлежащие ей устройства хранения информации не имеют физического соединения с другой парой.

**101. ЦЕНТРЫ ОБРАБОТКИ ДАННЫХ (ЦОД)**

Центр обработки данных (ЦОД) - это отказоустойчивая комплексная централизованная система, обеспечивающая автоматизацию бизнес-процессов с высоким уровнем производительности и качеством предоставляемых сервисов.

Назначение ЦОД. обеспечение гарантированной безотказной работы информационной системы предприятия с заданными уровнями доступности, надежности, безопасности и управляемости.

- Корпоративные ЦОД, которые работают в рамках конкретной компании;

- ЦОД, предоставляющие сервисы сторонним пользователям.

Центр обработки данных включает:

-высоконадежное серверное оборуование

-системы хранения и передачи данных

-программное обеспечение

-архитектурно-технические решения

-обеспечивающую инженерную инфраструктуру

-физическую защиту помещений

-комплекс организационных мероприятий

-систему мониторинга и управления

**102. СЕРВЕРА ЛЕЗВИЙНОЙ АРХИТЕКТУРЫ**

Архитектура "сервера лезвийного типа" (blade server architecture) представляет собой концепцию построения высокопроизводительных вычислительных систем на основе унифицированных модулей, известных как "лезвия". Эти лезвии являются небольшими, высокоплотными и высокоинтегрированными устройствами, которые могут помещаться в шасси и обеспечивать несколько вычислительных узлов в одном стандартизированном корпусе.

Основные особенности серверов лезвийной архитектуры включают в себя:

1. Высокую плотность: Лезвии обеспечивают высокую плотность вычислительных ресурсов, позволяя размещать множество серверов в едином шасси. Это позволяет сэкономить место в серверной комнате и уменьшить потребление энергии.

2. Простоту управления: Лезвийные сервера позволяют централизованно управлять несколькими вычислительными узлами через общую систему управления.

3. Гибкость: Возможность быстрой замены или добавления лезвий, что обеспечивает гибкость в настройке вычислительных ресурсов под конкретные потребности.

4. Упрощение инфраструктуры: Использование лезвий позволяет упростить инфраструктуру серверной комнаты, уменьшив количество кабелей, снижая потребление энергии и упрощая резервное копирование и обслуживание.

**103. СХД. ТОПОЛОГИИ СХД (DAS. NAS. SAN.)**



• DAS (Direct Attached Storage) – прямое подключение

• NAS (Network Attached Storage) – хранилище, подключенное по сети. Это файловый сервер, устройство подключенное в локальную сеть и предоставляющее доступ к своим дискам. NAS работает поверх локальной сети, используя обычное сетевое оборудование. Работает преимущественно с файлами. Позволяет коллективно использовать информацию на дисках.

**Плюсы**: дешево, доступно, простота коллективного использования ресурсов.

**Минусы**: невозможно исследовать не файловые методы доступа, доступ через сетевые протоколы медленнее, чем локальный доступ. • SAN (Storage Area Networks) – сеть хранения данных. Работает на уровне блоков данных, это могут быть как файлы, так и базы данных. Коллективное использование информации не всегда возможно.

**Плюсы:** возможность использовать блочные методы доступа, низкоуровневый доступ к диску, быстрый доступ к данным, многие приложения работают только с локальными дисками и не поддерживают обмен данными по сети.

**Минусы**: дорого, требуется специальное ПО.

**104. ДИСКОВЫЕ RAID-МАССИВЫ**

RAID — технология виртуализации данных, которая объединяет несколько дисков в логический элемент для избыточности и повышения производительности. Аббревиатура «RAID» изначально расшифровывалась как «redundant array of inexpensive disks» («избыточный (резервный) массив недорогих дисков», так как они были гораздо дешевле дисков SLED (. Именно так был представлен RAID его создателями Петтерсоном (David A. Patterson), Гибсоном (Garth A. Gibson) и Катцом (Randy H. Katz) в 1987 году. Со временем «RAID» стали расшифровывать как «redundant array of independent disks» , потому что для массивов приходилось использовать и дорогое оборудование (под недорогими дисками подразумевались диски для ПЭВМ).

Калифорнийский университет в Беркли представил следующие уровни спецификации RAID, которые были приняты как стандарт де-факто:

·RAID 0 — дисковый массив повышенной производительности с чередованием, без отказоустойчивости;

·RAID 1 — зеркальный дисковый массив;

·RAID 2 — зарезервирован для массивов, которые применяют код Хемминга;

·RAID 3 и 4 — дисковые массивы с чередованием и выделенным диском чётности;

·RAID 5 — дисковый массив с чередованием и «невыделенным диском чётности»;

**105. ВИРТУАЛИЗАЦИЯ ПРОГРАММНАЯ И АППАРАТНАЯ**

Вир**т**уализация — технология, которая позволяет использовать вычислительные ресурсы одной физической машины, например, сервера, для создания нескольких изолированных друг от друга виртуальных машин — со своими ресурсами. Иначе говоря, с ее помощью можно создавать отделенные друг от друг среды, которые используют мощности одного физического компьютера или сервера. При этом виртуальные машины могут иметь любые конфигурации (значения памяти, частоту процессора и др.) — конечно, в пределах максимальных мощностей физического устройства.

Программная виртуализация — та, в которой деление аппаратных ресурсов физического сервера осуществляется посредством операционной системы. Из-за этого все виртуальные машины используют общее программное ядро. Из этого следует главное требование для программной виртуализации — ОС виртуальных машин должна поддерживать программное ядро операционной системы физического сервера.

Аппаратная — виртуализация, при которой ресурсы физического сервера изолированно разделяются друг от друга на виртуальные машины, в каждой может быть своя полноценная ОС и свое ядро этой операционной системы.

**106. ВИРТУАЛЬНАЯ МАШИНА. ХОСТ**

Виртуальная машина (ВМ) — это программное обеспечение или эмулятор, который создает виртуальную среду, эмулирующую физическую компьютерную аппаратную платформу. ВМ позволяет запускать и использовать несколько операционных систем и приложений на одном и том же физическом компьютере одновременно.

Виртуальная машина изолирует программное обеспечение от физической аппаратной платформы, что обеспечивает уровень абстракции и позволяет эффективно использовать ресурсы, такие как процессорное время, память и хранилище данных.

Хост виртуализации (родительский сервер) — физический сервер с запущенным на нем гипервизором.

**107. ОБЛАСТИ ПРИМЕНЕНИЯ ВИРТУАЛИЗАЦИИ. ДОСТОИНСТВА И НЕДОСТАТКИ**

Виртуализация - это технология, которая позволяет создавать виртуальные версии аппаратных ресурсов, таких как серверы, сети, хранилища данных и операционные системы. Эти виртуальные ресурсы могут быть использованы для запуска нескольких приложений и операционных систем на одном физическом оборудовании. Вот некоторые из основных областей применения виртуализации:

1. Централизованное управление: Виртуализация позволяет централизованно управлять ресурсами, такими как серверы, сети и хранилища данных, что упрощает администрирование и повышает эффективность использования ресурсов.

2. Консолидация серверов: Виртуализация позволяет запускать несколько виртуальных серверов на одном физическом сервере, что помогает сократить количество физических серверов, уменьшить затраты на оборудование и упростить управление.

3. Тестирование и разработка: Виртуализация предоставляет изолированную среду для тестирования новых приложений и обновлений операционных систем без влияния на рабочие системы и приложения.

4. Безопасность: Виртуализация позволяет изолировать различные приложения и операционные системы друг от друга, что повышает безопасность и защиту от вредоносных атак.

5. Высокая доступность и отказоустойчивость: Виртуализация позволяет создавать резервные копии виртуальных машин и быстро восстанавливать их в случае сбоев или отказов оборудования.

6. Экономия энергии: Виртуализация позволяет лучше использовать вычислительные ресурсы, что приводит к снижению энергопотребления и экономии электроэнергии.

Достоинства виртуализации:

- Эффективное использование аппаратных ресурсов

- Упрощенное управление и администрирование

- Увеличение гибкости и масштабируемости

- Улучшение безопасности и отказоустойчивости

Недостатки виртуализации:

- Необходимость дополнительных инвестиций в оборудование и программное обеспечение

- Возможное снижение производительности из-за накладных расходов виртуализации

- Усложнение конфигурации и управления виртуальными средами

**108. ГИПЕРВИЗОР**

Гипервизор – специальное программное обеспечение, позволяющее создавать виртуальные машины и управлять ими.

**109. ОБЛАЧНЫЕ ТЕХНОЛОГИИ. ДОСТОИНСТВА И НЕДОСТАТКИ**

**Облачные вычисления** (cloud computing) - это способ обеспечения удаленного доступа к вычислительным ресурсам через интернет.

**Преимущества облачных технологий:**

· Вся информация доступна с любого устройства, - будь то ПК, планшет, смартфон и т.д. - подключенного к интернету. Здесь и плюс в том, что пользователь не привязан к определенному рабочему месту.

· Сокращение расходов на приобретение дорогостоящих мощных компьютеров, серверов, нет надобности оплачивать работу ИТ-специалиста для обслуживания локального дата-центра.

· Необходимые инструменты для работы предоставляются автоматически веб-сервисом.

· Высокий уровень технологичности вычислительных мощностей, который предоставляется пользователю, позволяет хранить, анализировать и обрабатывать данные.

· Оплачиваются сервисы только по мере необходимости их использования, при этом оплата происходит только за требуемый пакет услуг.

· Современные облачные вычисления могут обеспечивать наивысшую надежность, к тому же, лишь небольшое количество организаций могут позволить себе содержать полноценный дата-центр.

**Недостатки облачных технологий:**

· Для работы с «облаком» требуется постоянное подключение к интернету.

· Пользователь не всегда может настроить используемое программное обеспечение под личные нужды.

· Чтобы создать собственное «облако» потребуются очень большие затраты, что не целесообразно для новых предприятий.

· «Облако» - хранилище данных, к которым, используя уязвимости системы, могут получить доступ злоумышленники.

**110. ТИПЫ ОБЛАКОВ**

**Частное облако** (англ. private cloud) — инфраструктура, предназначенная для использования одной организацией, включающей несколько потребителей (например, подразделений одной организации), возможно также клиентами и подрядчиками данной организации. Частное облако может находиться в собственности, управлении и эксплуатации как самой организации, так и третьей стороны (или какой-либо их комбинации), и оно может физически существовать как внутри, так и вне юрисдикции владельца.

**Публичное облако** (англ. public cloud) — инфраструктура, предназначенная для свободного использования широкой публикой. Публичное облако может находиться в собственности, управлении и эксплуатации коммерческих, научных и правительственных организаций (или какой-либо их комбинации). Публичное облако физически существует в юрисдикции владельца — поставщика услуг.

**Общественное облако** (англ. community cloud) — вид инфраструктуры, предназначенный для использования конкретным сообществом потребителей из организаций, имеющих общие задачи (например, миссии, требований безопасности, политики, и соответствия различным требованиям). Общественное облако может находиться в кооперативной (совместной) собственности, управлении и эксплуатации одной или более из организаций сообщества или третьей стороны (или какой-либо их комбинации), и оно может физически существовать как внутри, так и вне юрисдикции владельца. Гибридное облако[править | править код]

**Гибридное облако** (англ. hybrid cloud) — это комбинация из двух или более различных облачных инфраструктур (частных, публичных или общественных), остающихся уникальными объектами, но связанных между собой стандартизованными или частными технологиями передачи данных и приложений (например, кратковременное использование ресурсов публичных облаков для балансировки нагрузки между облаками).

**111. ТРИ МОДЕЛИ ОБСЛУЖИВАНИЯ ОБЛАЧНЫХ ВЫЧИСЛЕНИЙ**

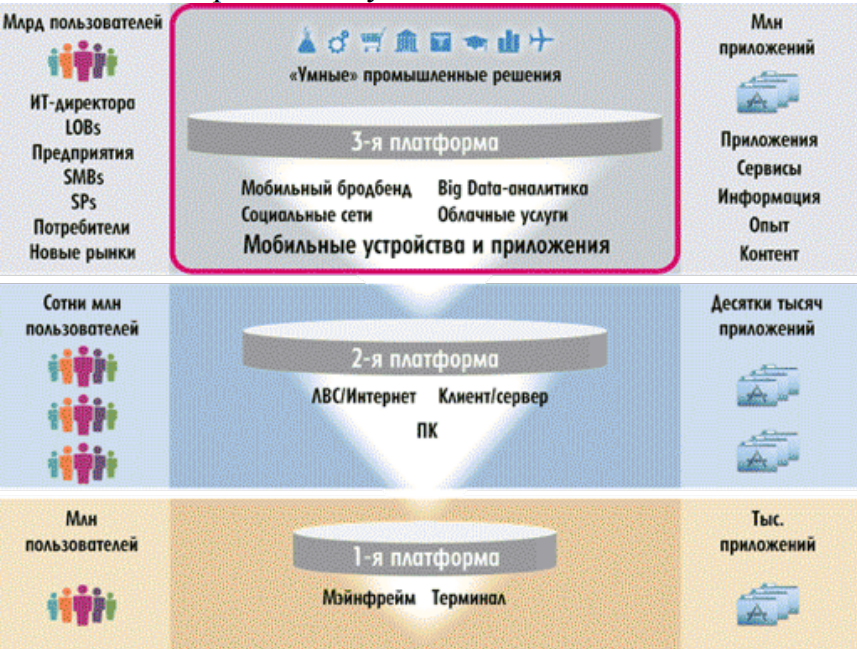
**Программное обеспечение как услуга**(SaaS,Softare-as-a-Service )- модель, в которой потребителю предоставляется возможность использования прикладного программного обеспечения провайдера, работающего в облачной инфраструктуре и доступного из различных клиентских устройств или посредством тонкого клиента, например, из браузера(например, веб-почта) или посредством интерфейса программы. Контроль и управление основной физической и виртуальной инфраструктурой облака, в том числе сети, серверов, операционных систем, хранения, или даже индивидуальных возможностей приложения (за исключением ограниченного набора пользовательских настроек конфигурации приложения) осуществляется облачным провайдером.

**Платформа как услуга**(PaaS,Platform-as-a-Service)— модель, когда потребителю предоставляется возможность использования облачной инфраструктуры для размещения базового программного обеспечения для последующего размещения на нём новых или существующих приложений (собственных, разработанных на заказ или приобретённых тиражируемых приложений). В состав таких платформ входят инструментальные средства создания, тестирования и выполнения прикладного программного обеспечения —системы управления базами данных,связующее программное обеспечение, среды исполнения языков программирования— предоставляемые облачным провайдером. Контроль и управление основной физической и виртуальной Программная Аппаратная инфраструктурой облака, в том числе сети, серверов, операционных систем, хранения осуществляется облачным провайдером, за исключением разработанных или установленных приложений, а также, по возможности, параметров конфигурации среды (платформы).

**Инфраструктура как услуга**(IaaS,IaaS or Infrastructure-as-a-Service) предоставляется как возможность использования облачной инфраструктуры для самостоятельного управления ресурсами обработки, хранения, сетями и другими фундаментальными вычислительными ресурсами, например, потребитель может устанавливать и запускать произвольное программное обеспечение, которое может включать в себя операционные системы, платформенное и прикладное программное обеспечение. Потребитель может контролировать операционные системы, виртуальные системы хранения данных и установленные приложения, а также обладать ограниченным контролем за набором доступных сетевых сервисов (например,межсетевым экраном,DNS). Контроль и управление основной физической и виртуальной инфраструктурой облака, в том числе сети, серверов, типов используемых операционных систем, систем хранения осуществляется облачным провайдером.

**112. ТРЕТЬЯ ПЛАТФОРМА. ПЕРСПЕКТИВЫ РАЗВИТИЯ ВС И IT-ТЕХНОЛОГИЙ**

«Третья платформа» названа так потому, что следует третьей в ряду платформ. Первая — на базе мэйнфреймов и терминалов с тысячами приложений и миллионами пользователей, вторая — на базе ПК и клиент-серверной архитектуры с сотнями тысяч приложений и миллиардами пользователей, а третья платформа — на базе различных, в том числе мобильных устройств, мобильного Интернета, социальных сетей, облачных технологий и построения всевозможных решений «умной» экономики.



Почему же обозначенные четыре составляющие (облако, мобильность, соцсети и «большие данные») объединены в одну платформу? Ответ на данный вопрос можно найти на рис. 2, который был приведен в докладе менеджера по исследованиям IDC Александра Прохорова на конференции IDC по управлению контентом в сентябре текущего года. Взаимосвязь указанных технологий можно проследить не только по горизонтали, как это указано на рисунке, но и в любых сочетаниях.



Действительно, пользователи растущего числа мобильных устройств производят всё больше контента, который удобно хранить в облаках. Контент, помещенный в облако, может «раздаваться» владельцам различных мобильных платформ в соответствующих форматах. Количество мобильных устройств непрерывно растет, поэтому потребность в облачном хранилище постоянно увеличивается. По данным IDC, количество смартфонов в 2015 году превысит 1,25 млрд штук, а количество планшетов — 300 млн, опередив количество ноутбуков.

Пользователи мобильных устройств все чаще обращаются к социальным сервисам, и именно за счет мобильных устройств повышается активность пользователей в соцсетях. Непрерывная совместная работа в реальном времени генерирует массу важной для корпорации информации. Это и мнения пользователей о бренде, и пожелания усовершенствования продуктов, и указания на недостатки продуктов, и поиск товаров. Вся эта информация нужна корпорациям, а для ее обработки требуются новые, всё более мощные средства анализа — средства анализа больших данных. Но под «большими данными» подразумевают не только системы аналитической обработки, ведь технологии предоставления социальных сетей для миллиардов пользователей — это тоже технологии «больших данных». Facebook, LinkedIn представляют собой сервисы, обеспечивающие высокоскоростную обработку миллиардов учетных записей и ассоциированной с ними информации.

Обмен информацией в социальных сетях позволяет устанавливать неформальные связи с партнерами и привлекать новых. Это приводит к лавинообразному росту информации, причем растущая ее часть оказывается вне зоны внимания организации. Собрать и проанализировать ее — одна из задач технологии Big Data (рис. 4). Технологию «больших данных» обычно представляют как технологию, характеризующуюся тремя «V»: скорость (Velocity), разнообразие (Variety) и объем (Volume). Скорость указывает на необходимость обработки большого объема данных при их быстром поступлении, как, например, при обработке потоковых данных. Вариативность говорит о способности системы обрабатывать большие массивы данных, поступающие из разных источников в различных форматах. Объем свидетельствует о больших объемах данных. Авторы ресурса Watalan.com представляют эволюцию развития концепции «больших данных» и отмечают, что технология Big Data напрямую связана с возникновением облачных технологий (предоставление эластичных ресурсов для обработки и хранения больших объемов данных).

Очевидно, что задачи «больших данных» требуют больших расходов на оборудование и программное обеспечение, а с помощью облачных технологий можно перевести капитальные расходы в операционные и снизить таким образом траты на владение ИТ. Облачные технологии предлагают новый тип услуг — предоставление хранилища и базы данных в рамках одной услуги. Согласно данным Watalon.com, связь технологий Big Data и облачных вычислений можно проследить на различных уровнях.

Аналитики IDC подчеркивают, что способность использования технологии «третьей платформы» для современных корпораций будет в существенной мере определять их конкурентоспособность. Следует отметить, что необходимость ориентации на технологии «третьей платформы» для ИТ-вендоров — это неоспоримый факт, причем важность данного процесса трудно оценить абсолютными цифрами и даже темпами роста отдельных направлений. Этот переход следует рассматривать как большое преобразование ИТ-индустрии. Для большинства ИТ-вендоров вопрос об использовании указанных технологий определяет не только возможность увеличить свою прибыль, но и вероятность того, насколько вендор впишется в модель поставки ИТ в ближайшие пятьдесять лет и в принципе сохранит ли свое место на рынке.