Вопросы по Экзамену МП

1. Микропроцессоры. Классификация по назначению, по конструкции.

Микропроцессор — 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).

Классификация микропроцессоров:

- По назначению:

Универсальные микропроцессоры могут быть применены для решения широкого круга разнообразных задач. При этом их эффективная производительность слабо зависит от проблемной специфики решаемых задач. Специализация МП, т.е. его проблемная ориентация на ускоренное выполнение определенных функций позволяет резко увеличить эффективную производительность при решении только определенных задач.

Среди специализированных микропроцессоров можно выделить различные микроконтроллеры, ориентированные на выполнение сложных последовательностей логических операций, математические МП, предназначенные для повышения производительности при выполнении арифметических операций за счет, например, матричных методов их выполнения, МП для обработки данных в различных областях применений и т.д. С помощью специализированных МП можно эффективно решать новые сложные задачи параллельной обработки данных.

2. Регистры общего назначения.

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

- AX — аккумулятор. Использовался для хранения операндов в командах умножения и деления, ввода-вывода, в некоторых командах обработки строк и других операциях;
- BX — регистр базы. Используется для хранения адреса или части адреса операнда, находящегося в памяти;

- CX — счётчик. Содержит количество повторений строковых операций, циклов и сдвигов;
- DX — регистр данных. Используется для косвенной адресации портов ввода-вывода, а также как «расширитель» аккумулятора в операциях удвоенной разрядности;
- SI — регистр адреса источника. Используется в строковых операциях, а также в качестве индексного регистра при обращении к операндам в памяти;
- DI — регистр адреса приёмника. Используется в строковых операциях, а также в качестве индексного регистра при обращении к операндам в памяти;
- BP — указатель кадра стека. Используется для адресации операндов, расположенных в стеке;
- SP — указатель стека. Используется при выполнении операций со стеком, но не для явной адресации операндов в стеке.

1. Микропроцессоры. Классификация по технологии изготовления, по быстродействию, по способу доступа к памяти, по разрядности, по способу реализации команд.

Классификация микропроцессоров:

- По технологии изготовления:
- По способу доступа к памяти:
- По разрядности:

Обрабатываемой информации МП могут быть 4, 8, 12, 16, 24, 32 - разрядными. На практике наибольшее распространение имеют 32 - разрядные МП (Pentium, Celeron, AMD). Все большее применение находят 64-разрядные МП фирмы AMD.

- По способу реализации команд:

Набор регистров процессора. Сегментные регистры.

Сегментные регистры — регистры, указывающие на сегменты.

Все сегментные регистры — 16-разрядные.

CS (англ. Code Segment), DS (англ. Data Segment), SS (англ. Stack Segment), ES (англ. Extra Segment), FS, GS

В реальном режиме работы процессора сегментные регистры содержат адрес начала 64Kb сегмента, смещённый вправо на 4 бита.

В защищённом режиме работы процессора сегментные регистры содержат селектор сегмента памяти, выделенного ОС.

CS — указатель на кодовый сегмент. Связка CS:IP(CS:EIP/CS:RIP — в защищённом/64-битном режиме) указывает на адрес в памяти следующей команды.

В 64-разрядном режиме сегментные регистры CS, DS, ES и SS в формировании линейного (непрерывного) адреса не участвуют, поскольку сегментация в этом режиме не поддерживается.

Структурная схема ПК. Принципы фон Неймана.

Структурно основная память состоит из пронумерованных ячеек; процессору в произвольный момент времени доступна любая ячейка. Отсюда следует возможность давать имена областям памяти, так, чтобы к хранящимся в них значениям можно было бы впоследствии обращаться или менять их в процессе выполнения программы с использованием присвоенных имен.

Принцип последовательного программного управления

Предполагает, что программа состоит из набора команд, которыевыполняются процессором автоматически друг за другом в определенной последовательности.

Принцип жесткости архитектуры

Неизменяемость в процессе работы топологии, архитектуры, списка команд.

Компьютеры, построенные на этих принципах, относят к типу фон-неймановских.

3. Описать архитектуру ОКОД. Описать архитектуру ОКМД.

Single Instruction, Single Data) или ОКОД (Одиночный поток Команд, Одиночный поток Данных) — архитектура компьютера, в которой один процессор выполняет один поток команд, оперируя одним потоком данных. Относится к фон-Неймановской архитектуре. Один из классов вычислительных систем в классификации Флинна.Один из классов вычислительных систем в классификации Флинна.

single instruction, multiple data — одиночный поток команд, множественный поток данных, ОКМД) — принцип компьютерных вычислений, позволяющий обеспечить параллелизм на уровне данных. ... Одним из преимуществ данной архитектуры считается то, что в этом случае более эффективно реализована логика вычислений.

4. Характеристики микропроцессоров.

Характеризуется:

1) тактовой частотой, определяющей максимальное время выполнения переключения элементов в ЭВМ;

2) разрядностью, т.е. максимальным числом одновременно обрабатываемых двоичных разрядов.

Разрядностть МП обозначается m/n/k/ и включает:

m - разрядность внутренних регистров, определяет принадлежность к тому или иному классу процессоров;

n - разрядность шины данных, определяет скорость передачи информации;

k - разрядность шины адреса, определяет размер адресного пространства.

Например, МП i8088 характеризуется значениями m/n/k=16/8/20;

3) архитектурой. Понятие архитектуры микропроцессора включает в себя систему команд и способы адресации, возможность совмещения выполнения команд во времени, наличие дополнительных устройств в составе микропроцессора, принципы и режимы его работы. Выделяют понятия микроархитектуры и макроархитектуры.

Микроархитектура микропроцессора - это аппаратная организация и логическая структура микропроцессора, регистры, управляющие схемы, арифметико-логические устройства, запоминающие устройства и связывающие их информационные магистрали.

Макроархитектура - это система команд, типы обрабатываемых данных, режимы адресации и принципы работы микропроцессора.

В общем случае под архитектурой ЭВМ понимается абстрактное представление машины в терминах основных функциональных модулей, языка ЭВМ, структуры данных.

8. Система команд микропроцессора. Классификация команд по функциям (выполняемым операциям), по направлению приема - передачи, по адресности.

Характеристика системы команд процессора - Система команд - это набор допустимых для данного процессора управляющих кодов и способов адресации данных. Система команд жестко связана с конкретным типом процессора, поскольку определяется аппаратной структурой блока дешифрации команд, и обычно не обладает переносимостью на другие типы процессоров (хотя может иметь место совместимость “снизу-вверх” в рамках серии процессоров, как, например, в серии i80x86). С физической точки зрения код команды ничем не отличается от обычных данных в двоичном коде, размещенных в памяти вычислителя. Конкретный двоичный код воспринимается и обрабатывается процессором как команда в том случае, когда он попадает в процессор в фазе чтения кода команды. С логической точки зрения в двоичном коде команды существуют группы разрядов – поля – с различным функциональным назначением

Способ адресации – это способ получения процессором адреса операнда или перехода на основании информации из адресной части команды. Различают следующие основные способы адресации:

· прямая – адрес операнда или перехода содержится в АЧ команды;

· непосредственная – в АЧ команды содержится значение операнда;

· регистровая – в коде команды содержится указание на один или два регистра процессора, являющихся источниками операндов или приемником результата;

· косвенная регистровая – в коде команды содержится указание на какой-либо регистр процессора, содержимое которого при выполнении команды интерпретируется процессором как адрес ячейки памяти, содержащей операнд;

· косвенная базовая (иногда – индексная) – адрес операнда формируется (вычисляется) процессором в ходе выполнения команды как сумма содержимого одного из регистров и смещения (числа), задаваемого в команде, либо как сумма содержимого двух регистров. Таким образом, базовая или индексная формы адресации также являются разновидностью косвенной адресации.

+Регистры, которые можно использовать для реализации косвенной адресации, часто называют указательнымирегистрами: в самом деле, они как бы “указывают” на ту ячейку памяти, в которой содержится операнд. Использование косвенной адресации более предпочтительно, поскольку такой подход позволяет создавать универсальные, легко перенастраиваемые (используется термин «переносимые»), и позиционно независимые программы. Применение прямой адресации “привязывает” программу к

конкретным ячейкам памяти, и при этом резко снижается возможность ее использования в различных проектах. Важнейшая особенность косвенной адресации заключается в том, что адрес операнда должен формироваться в процессе выполнения программы (в то время, как в случае использования прямой адресации адреса всех операндов должны быть определены и указаны при написания текста программы).

9. Основные этапы развития компьютеров. Поколения ЭВМ.

Основные этапы развития компьютеров:

- **Нулевое поколение — механические компьютеры (1642-1945)** - первым человеком, создавшим счетную машину, был французский ученый Блез Паскаль (1623-1662), в честь которого назван один из языков программирования.
- **Первое поколение — электронные лампы (1945-1955)** - стимулом к созданию электронного компьютера стала Вторая мировая война.
- **Второе поколение — транзисторы (1955-1965)** - транзистор был изобретен сотрудниками лаборатории Bell Laboratories Джоном Бардином Oohn Bardeen), Уолтером Браттейном (Walter Brattain) и Уильямом Шокли (William Shockley), за что в 1956 году они получили Нобелевскую премию в области физики. В течение десяти лет транзисторы совершили революцию в производстве компьютеров, и к концу 50-х годов компьютеры на вакуумных лампах уже безнадежно устарели. Первый компьютер на транзисторах был построен в лаборатории МТИ (Массачусетским Техническим Институтом). Он содержал слова из 16 бит, как и Whirlwind I. Компьютер назывался TX-0 (Transistorized experimental computer 0 — экспериментальная транзисторная вычислительная машина 0) и предназначался только для тестирования будущей машины TX-2.
- **Третье поколение — интегральные схемы (1965-1980)** - изобретение в 1958 году Робертом Нойсом (Robert Noyce) кремниевой интегральной схемы означало возможность размещения на одной небольшой микросхеме десятков транзисторов. Компьютеры на интегральных схемах были меньшего размера, работали быстрее и стоили дешевле, чем их предшественники на транзисторах.
- **Четвертое поколение — сверхбольшие интегральные схемы (1980-?)** - появление сверхбольших интегральных схем (СБИС) в 80-х годах позволило помещать на одну плату сначала десятки тысяч, затем сотни тысяч и, наконец, миллионы транзисторов. Это привело к созданию компьютеров меньшего размера и более быстродействующих.

10. Классы процессоров CISC, RISC, MISC, VLIM.

CISC (Complete Instruction Set Computer) – полный набор команд микропроцессора.

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.

RISC - Reduced Instruction Set Computer – архитектура компьютера с сокращенным набором команд. RISC-архитектура предполагает реализацию в ЭВМ сокра-щенного набора простейших, но часто употребляемых команд, что позволяет упростить аппаратурные средства процессора и благодаря этому получить возможность повысить его быстро-действие.

Современные процессоры типа RISC характеризуются следующими особенностями:

упрощенный набор команд, имеющих одинаковую длину.

Отсутствуют макрокоманды, усложняющие структуру процессора и уменьшающую скорость его работы.

Взаимодействие с оперативной памятью ограничивается операциями пересылки данных.

Уменьшено число способов адресации

Используется конвейер команд.

Применяется высокоскоростная память.

VLIW – появилась в России. Она не попадает под принципы фон Неймана (нарушает принцип программного управления, т.е. последовательного выполнения команд).

Архитектура ЭВМ с длинным командным словом (VLIW – Very Long Instruction Word) позволяет сократить объем оборудования, требуемого для реализации параллельной выдачи несколько команд, базируется на множестве независимых командных устройств. Вместо того чтобы выдавать на эти устройства последовательные команды, операции упаковываются в

одну очень длинную команду. Ответственность за вывод параллельно выдаваемых для выполнения команд полностью ложится на компилятор.

В процессорах VLIW задача решается распределения работы между вычислительными модулями во время компиляции. Аппаратные средства, необходимые для реализации параллельной обработки, отсутствуют.

MISC, minimal instruction set computer — минимальный набор команд компьютера.

Увеличение разрядности процессоров привело к идее укладки нескольких команд в одно большое слово (связку, bound). Это позволило использовать возросшую производительность компьютера и его возможность обрабатывать одновременно несколько потоков данных. Кроме этого, MISC использует стековую модель вычислений.

Процессоры, образующие «компьютеры с минимальным набором команд» (MISC), как и процессоры RISC, характеризуются небольшим числом чаще всего встречающихся команд. Вместе с этим, принцип «очень длинных командных слов» (VLIW) обеспечивает выполнение группы непротиворечивых команд за один цикл работы процессора. Порядок выполнения команд распределяется таким образом, чтобы в максимальной степени загрузить маршруты, по которым проходят потоки данных. Таким образом архитектура MISC объединила вместе суперскалярную и VLIW-концепции. Компоненты процессора просты и работают на высоких частотах.

2. Фирмы – производители микропроцессоров. Микропроцессоры первого и второго поколений.

Фирмы:

- AMD.
- Intel.
- Silicon Graphics.
- Sun Microsystems.
- VIA.

Микропроцессоры первого и второго поколений:

- Первый МП был разработан фирмой INTEL и выпущен в 1971г. на основе p-МОП технологии (i4004). В 1972 и 1973 годах этой же фирмой были выпущены модели i4040, i8008. Фирма Rockwell выпустила модели МП PSS-4, PSS-8.

- МП второго поколения, которые отличались от МП-1 не только количественными характеристиками, но и качественно. В 1974г. был выпущен МП i8080, который стал первым и наиболее популярным МП второго поколения (МП-2). Он же положил начало семейству однокристальных МП, которому суждено было стать (и оставаться до настоящего времени) доминирующим на мировом рынке МП. Вслед за i8080 другими фирмами были выпущены МП со сходными (иногда несколько лучшими) характеристиками. Наиболее известными являются. Z80 фирмы Zilog и MS6800 (Motorola). Эти МП, как и i8080, имеют своих 16- и 32-разрядных "потомков".

3. Технология повышения производительности процессора. Конвейерная обработка. Суперскаляризация.

Конвейерная обработка команд. Суперскаляризация. Рассмотрим процесс выполнения процессором команды для коротких (арифметических с фиксированной запятой или логических) операций. Обработка команды, или цикл процессора, может быть разделена на несколько основных этапов, которые можно назвать микрокомандами, которых известно пять основных типов.

Каждая операция требует для своего выполнения времени, равного такту генератора процессора (tick of the internal clock). Отметим, что к длинным операциям (плавающая точка) это не имеет отношения — там другая арифметика. Очевидно, что при тактовой частоте в 100 МГц быстродействие составит 20 миллионов операций в секунду.

Все этапы команды задействуются только 1 раз и всегда в одном и том же порядке — одна за другой. Это, в частности, означает, что если схема первой микрокоманды выполнила свою работу и передала результаты второй, то для выполнения текущей команды она больше не понадобится, и, следовательно, может приступить к выполнению следующей команды.

Конвейеризация осуществляет многопоточную параллельную обработку команд, так что в каждый момент одна из команд считывается, другая декодируется и т. д., и всего в обработке одновременно находится пять команд. Таким образом, на выходе конвейера на каждом такте процессора появляется результат обработки одной команды (одна команда в один такт). Первая инструкция может считаться выполненной, когда завершат работу все пять микрокоманд.

Такая технология обработки команд носит название конвейерной (pipeline) обработки. Каждая часть устройства называется ступенью (стадией) конвейера, а общее число ступеней — длиной линии конвейера.

С ростом числа линий конвейера и увеличением числа ступеней на линии увеличивается пропускная способность процессора при неизменной тактовой частоте. Процессоры с несколькими линиями конвейера получили название суперскалярных. Pentium — первый суперскалярный процессор Intel.

4. Программирование на языке кодовых операций.

Программирование — процесс создания компьютерных программ с помощью языков программирования.

Язык программирования — формальная знаковая система, предназначенная для описания алгоритмов в форме программы, которая удобна для исполнителя (например, компьютера).

Единственный язык, напрямую выполняемый процессором — это машинный язык (также называемый машинным кодом). Изначально все программисты прорабатывали каждую мелочь в машинном коде, но сейчас эта трудная работа уже не делается. Вместо этого программисты пишут исходный код, и компьютер (используя компилятор, интерпретатор или ассемблер) транслирует его, в один или несколько этапов, уточняя все детали, в машинный код, готовый к исполнению на целевом процессоре. Даже если требуется полный низкоуровневый контроль над системой, программисты пишут на языке ассемблера, мнемонические инструкции которого преобразуются один к одному в соответствующие инструкции машинного языка целевого процессора.

Язык ассемблера — тип языка программирования низкого уровня, представляющий собой формат записи машинных команд, удобный для восприятия человеком.

5. Технология повышения производительности процессора. Динамическое исполнение.

Динамическое исполнение (Dynamic execution technology)- это совокупность технологий обработки данных в процессоре,

обеспечивающая более эффективную работу процессора за счет ма-нипулирования данными, а не простого исполнения списка инст-рукций.

Динамическое исполнение представляет собой-комбинацию трех методов обработки данных:

множественное предсказание ветвлений;

анализ потока данных;

спекулятивное (по предположению) исполнение. Впервые реализовано в процессоре Pentium Pro.

Множественное предсказание ветвлений. Пред-сказывается прохождение программы по нескольким ветвям: про-цессор может предвидеть разделение потока инструкций, используя алгоритм множественного предсказания ветвлений. С большой точ-ностью (более 90 %) он предсказывает, в какой области памяти можно найти следующие инструкции. Это оказывается возможным, поскольку в процессе исполнения иструкции процессор просматри-вает программу на несколько шагов вперед. Этот метод позволяет увеличить загруженность процессора.

Хотя BTB (Branch Target Buffer — буфер предсказания перехо-дов) и не может правильно предсказать абсолютно все переходы, но

большинство предсказаний оказывается точными, что обеспечивает

значительное повышение производительности. Например, про-граммный цикл, состоящий из пересылки, сравнения, сложения и перехода в 80486 DX выполняется за 6 тактов синхронизации, а в Pentium за 2 (команды пересылки и сложения, а также сравнения и перехода сочетаются и предсказывается переход).

Анализ потока данных. Анализирует и составляет гра-фик исполнения инструкций в оптимальной последовательности,

независимо от порядка их следования в тексте программы. Исполь-зуя анализ потока данных, процессор просматривает декодирован-ные инструкции и определяет, готовы ли они к непосредственному исполнению или зависят от результата других инструкций. Далее процессор определяет оптимальную последовательность выполне-ния и исполняет инструкции наиболее эффективным образом.

Спекулятивное выполнение. Повышает скорость вы-полнения, просматривая программу вперед и исполняя те инструк-ции, которые необходимы. Процессор выполняет инструкции (до пяти инструкций одновременно) по мере их поступления в оптими-зированной последовательности (спекулятивно). Поскольку выпол-нение инструкций происходит на основе предсказания ветвлений, результаты сохраняются как «спекулятивные». На конечном этапе порядок инструкций восстанавливается и переводится в обычное машинное состояние.

6. Многопроцессорные системы.

Многопроцессорная система (multiprocessor system) – вычислительная система, использующая для обработки данных более одного процессора. Примеры: двухпроцессорные персональные компьютеры, мощные серверы со множеством процессоров, суперкомпьютеры.

Большие инженерные и исследовательские приложения, выполняемые на суперкомпьютерах, обеспечивают прирост производительности за счет параллельной обработки данных на нескольких процессорах.

Коммерческие и научные организации используют многопроцессорные системы для повышения производительности, предоставления задачам достаточных ресурсов и достижения высокой надежности.

ОС многопроцессорных ЭВМ должны дополнительно гарантировать, что:

Все процессоры загружены работой.

Процессы равномерно распределены в системе.

Выполнение взаимосвязанных процессов синхронизировано.

Процессы работают с состоятельными копиями данных, хранящихся в общей памяти.

Обеспечивается взаимное исключение для выхода из тупиковых ситуаций

7. Технология повышения производительности процессора. Предикация. Опережающее считывание.

Предикация (высказывание, утверждение) в лингвистике — одна из функций языкового выражения; предикация соотносит мысль к действительности: состоянию объекта к субъекту, событие к ситуации. Предикация — формальное установление связей между субъектом и предикатом

В предикации, которую можно считать актом высказывания, построения пропозиции, выделяются два этапа:

- незавершённая предикация (или предикация в узком смысле) — соединение смыслов более простых языковых выражений;
- завершённая предикация (или предикация в широком смысле) — выражение отношения говорящего к действительности, утверждение или отрицание пропозиции;

Опережающее чтение, или **интервал копирования** — приём набора, при котором выполняется чтение букв (слогов, слов, словосочетаний) наперёд. Этот приём повсеместно применяется наборщиками при перепечатке текста с оригинала. Согласно проведённым экспериментам, только в тех случаях, когда наборщику для предварительного просмотра доступно более 7 символов, он может показать увеличение средней скорости. С увеличением средней скорости эта зависимость увеличивается.

8. Вычислительные платформы.

В современной организации любая деятельность связана так или иначе с использованием информационных технологий. При грамотном построении ИТ-инфраструктуры эффективность бизнес процессов повышается, за счет

чего можно принимать оперативные управленческие решения. Сердцем ИТ-систем являются вычислительные платформы.

Вычислительная платформа — программно-аппаратный комплекс инфраструктуры, от которого зависит работа всей информационной системы. Благодаря внедрению вычислительных платформ можно достигнуть высоких результатов, решая задачи при использовании вычислительных систем.

9. Технология HyperTreading.

Технология Intel® Hyper-Threading— это инновационная аппаратная технология, позволяющая обрабатывать на каждом ядре процессора несколько потоков. ... Таким образом, каждому физическому ядру соответствуют два логических ядра, которые могут обрабатывать разные программные потоки.

10. Микропроцессоры шестого поколения (Pentium Pro, PII/III, Celeron, Xeon).

Pentium Pro (произносится: Пентиум Про) — процессор Intel шестого поколения, совместимый с архитектурой x86. Был анонсирован 1 ноября 1995 года, однако доступен стал несколько позже. Первоначально планировалось заменить этим процессором всю линейку Pentium, но в дальнейшем от этих планов Intel отказалась и процессор позиционировался, в основном, как процессор для серверов и рабочих станций.

Intel Pentium III (в русской разговорной речи — Интел Пентиум три, сниженный вариант — третий пень) — x86-совместимый микропроцессор архитектуры Intel P6, анонсированный 26 февраля 1999 года (в России Pentium III поступили в продажу летом того же года). Ядро Pentium III представляет собой модифицированное ядро Deschutes (которое использовалось в процессорах Pentium II). По сравнению с предшественником расширен набор команд (добавлен набор инструкций SSE) и оптимизирована работа с памятью.

Celeron — семейство микропроцессоров Intel, предназначенное для заполнения самой «низкобюджетной» ниши рынка. ЦП марки Celeron разрабатываются и выпускаются с шестого поколения микроархитектур Intel по настоящее время.

Xeon (произносится как «Зион», в русском языке часто употребляется как «Ксеон») — линейка серверных микропроцессоров производства Intel. Название остаётся неизменным для нескольких поколений процессоров. Название ранних моделей состояло из соответствующего названия из ряда настольных процессоров и слова Xeon, современные модели имеют в

названии только Xeon. В общих чертах серверная линейка процессоров отличается от процессоров для настольных ПК:

- увеличенным объёмом кэш-памяти,
- поддержкой многопроцессорных систем большой производительности,
- поддержкой большего объёма памяти и портов ввода-вывода,
- памяти с коррекцией ошибок.

11. Классификация памяти компьютера.

1) По способу доступа к информации (память с произвольным доступом(оперативная), память с прямым циклическим доступом (дисковая), память с последовательным доступом(ленточная)

2) По функциональному назначению:

ПЗУ - постоянные запоминающие устройства или ROM

СОЗУ- сверхоперативное запоминающее устройство

ОЗУ - оперативное запоминающее устройство или RAM (

3) По способу адресации:

Адресная память

Ассоциативная память

Стековая память

4) По способу хранения информации:

Статистическая память

Динамическая память

Постоянная память

Голографическая память

Биологическая память

Память на магнитных носителях

Оптическая память

12. Микропроцессоры седьмого поколения (Pentium 4).

Intel Pentium 4 — одноядерный x86-совместимый микропроцессор компании Intel, представленный 20 ноября

2000 года, ставший первым микропроцессором, в основе которого лежала принципиально новая по сравнению с предшественниками архитектура седьмого поколения (по классификации Intel) — NetBurst.

13. Система памяти. Иерархическая организация памяти.

Иерархия компьютерной памяти — концепция построения взаимосвязи классов разных уровней компьютерной памяти на основе иерархической структуры.

Сущность необходимости построения иерархической памяти — необходимость обеспечения вычислительной системы (отдельного компьютера или кластера) достаточным объёмом памяти, как оперативной, так и постоянной.

11. Микропроцессоры фирмы AMD.

Процессор AMD Ryzen 5 3600.

Процессор AMD Athlon 3000G.

Процессор AMD Ryzen 5 5600X.

Процессор AMD Ryzen 3 1200.

Процессор AMD Ryzen 7 3700X.

Процессор AMD Ryzen 5 3500X.

AMD Threadripper 3990X

AMD FX-9590

AMD Phenom II X6

AMD Phenom II X4

AMD A10

AMD A8

AMD A6

12. Кэш – память. Необходимость. Структура кэш - памяти. Основные характеристики современных моделей памяти (попадание, промах).

Кэш память (Cache) — массив сверхбыстрой оперативной памяти, являющейся буфером между контроллером системной памяти и процессором. В этом буфере сохраняются блоки данных, с которыми центральный процессор работает в данный момент, тем самым значительно уменьшается количество обращений процессора к медленной системной

памяти. Тем самым заметно увеличивается общая производительность процессора.

Кэш память первого уровня (L1) — самый быстрый, но по объему меньший, чем у остальных. С ним напрямую работает ядро процессора. Она имеет наименьшую латентность (время доступа).
Кэш память второго уровня (L2) – объем этой памяти значительно больше, чем L1.
Кэш память третьего уровня (L3) – с большим объемом, но более медленная, чем L2.

В классическом варианте существовало 2 уровня кэш-памяти – 1-ий и второй уровень. 3-ий уровень по организации отличается от L2. Если данные не обрабатывались или процессор должен обработать срочные данные, то для освобождения L2 данные перемещаются в кэш память 3-го уровня. L3 больше по размеру, однако, и медленнее, чем L2 (шина между L2 и L3 более узкая, чем шина между L1 и L2), но все же его скорость, намного выше скорость оперативной памяти.

Кэш-памятью управляет специальное устройство — контроллер, который, анализируя выполняемую программу, пытается предвидеть, какие данные и команды вероятнее всего понадобятся в ближайшее время процессору, и подкачивает их в кэш-память. При этом возможны как "попадания", так и "промахи". В случае попадания, то есть, если в кэш подкачаны нужные данные, извлечение их из памяти происходит без задержки. Если же требуемая информация в кэше отсутствует, то процессор считывает её непосредственно из оперативной памяти. Соотношение числа попаданий и промахов определяет эффективность кэширования.

13. Совместимость процессоров.

Если два процессора имеют одинаковую систему команд, то они полностью совместимы на программном уровне. Это означает, что программа, написанная для одного процессора, может исполняться и другим процессором. Процессоры, имеющие разные системы команд, как правило, несовместимы или ограниченно совместимы на программном уровне.

14. Кэш – память. Стратегия управления памятью (кэш с прямым отображением, частично ассоциативная, полностью ассоциативная).

Существует три основных способа размещения блоков (строк) основной памяти в кэше:

- кэш-память с прямым отображением (direct-mapped cache);
- частично ассоциативная (или множественно ассоциативная, set-associative cache) кэш-память;
- полностью ассоциативная кэш-память (fully associative cache).

Память с прямым отображением. В этом случае каждый блок основной памяти имеет только одно фиксированное место, на котором он может появиться в кэш-памяти. Это наиболее простая организация кэш-памяти. Все блоки основной памяти, имеющие одинаковые младшие разряды в своем адресе, попадают в один блок кэш-памяти. При таком подходе справедливо соотношение:

(Адрес блока кэш-памяти) = (Адрес блока основной памяти) mod (Число блоков в кэш-памяти).

Полностью ассоциативная память. Здесь некоторый блок основной памяти может располагаться на любом месте кэш-памяти. Кэш-память называется частично ассоциативной, если некоторый блок основной памяти может располагаться на ограниченном множестве мест.

В современных процессорах, как правило, используется либо кэш-память с прямым отображением, либо двух- (четырех-) канальная множественно ассоциативная кэш-память.

Стратегия замещения информации в кэше определяет блок, подлежащий замещению при возникновении промаха. Простота при использовании кэша с прямым отображением заключается в том, что аппаратные решения здесь наиболее простые: легко реализуется сама аппаратура, легко происходит замещение данных. При замещении просто нечего выбирать — на попадание проверяется только один блок, и только этот блок может быть замещен.

При полностью или частично ассоциативной организации кэш-памяти имеются несколько блоков, из которых надо выбрать кандидата в случае промаха. Как правило, для замещения блоков применяются две основные стратегии:

- случайная (Random) — блоки-кандидаты выбираются случайно (чтобы иметь равномерное распределение). В некоторых системах используют псевдослучайный алгоритм замещения;
- замещается тот блок, который не использовался дольше всех (LRU — Least - Recen11y Used). В этом случае, чтобы уменьшить вероятность удаления информации, которая скоро может потребоваться, все обращения к блокам фиксируются.

Достоинство случайного способа заключается в том, что его проще реализовать в аппаратуре. Когда количество блоков увеличивается, алгоритм LRU становится все более дорогим и часто только приближенным.

15. Идентификация процессоров (инструкция CPUID).

CPUID (CPU Identification) - это дополнительная инструкция процессоров архитектуры x86, которая используется для получения информации о процессоре. С ее помощью можно определить тип процессора и его характеристики. Она впервые была представлена корпорацией Intel в 1993 году примерно в то же время, когда этой же корпорацией представила торговую марку процессоров x86 Pentium и улучшенные процессоры i486. Код инструкции CPUID - 0F A2 (если задавать двумя байтами, если словом - A20Fh). Во время исполнения инструкции задействован регистр EAX, и в некоторых случаях - ECX.

16. Память с безадресным обращением. Стековая память. Применение.

Стековые ЗУ - Стековая память состоит из ячеек, связанных друг с другом разрядными цепями передачи слов. Обмен информацией всегда выполняется только через верхнюю ячейку – *вершину стека.* При записи нового слова (команды, числа, символа) все ранее записанные слова сдвигаются на одну ячейку вниз, а новое слово помещается на вершину стека. Считывание возможно только с вершины стека и производится с удалением или без удаления считываемого слова. Такую память часто называют памятью типа LIFO (Last – In First – Out последним вошел, первым вышел). Аппаратная реализация стека сложна и обычно стек моделируют программно. При этом в качестве стека обычно используется часть адресной памяти.

В стековой памяти (памяти магазинного типа, организованной по принципу «Последним вошел - первым вышел» - LIFO - "Last In - First Out") все операции чтения и записи 32 осуществляются относительно указателя стека (SP-stack pointer). Указатель стека указывает на ячейку памяти, содержащую последнее внесенное в стек слово. Стековая память может организовываться программно-аппаратным или аппаратным способом. Команды обращения к стеку не содержат адресной части, либо эта часть является относительной величиной, прибавляемой к указателю. Это позволяет сократить длину программы, так как нет необходимости указывать достаточно длинные адреса, а также - упростить схему ЗУ при аппаратной реализации стека. В то же время при работе со стековой памятью приходится осуществлять фактически последовательный доступ, кроме того, может происходить т.н. переполнение стека - при попытке записать в полностью заполненный стек очередное значение, либо при считывании из пустого стека. Использование стековой памяти будет более эффективным, если процессор, работающий со стеком, будет поддерживать специальные стековые команды - не только «занести в стек» и «считать из стека», но и

такие, как «сложить два числа на вершине стека», «переставить элементы стека» и т.д. Такие команды часто используются в RISC-процессорах, в микроконтроллерах, управляющих ЭВМ.

17. Структура процессора. Устройство Управления (УУ), Операционное Устройство (ОУ). Микрооперации, микрокоманды, микропрограммы.

Управляющее устройство (УУ) координирует действия узлов операционного устройства; оно вырабатывает в некоторой временной последовательности управляющие сигналы, под действием которых в узлах операционного устройства выполняются требуемые действия.

Операционное устройство (ОУ) - устройство, в котором выполняются операции. Оно включает в качестве узлов регистры, сумматоры, арифметико–логическое устройство (АЛУ), каналы передачи информации, мультиплексоры для коммутации каналов, шифраторы, дешифраторы и т.д.

Некоторые логические действия (поразрядно выполняемые операции конъюнкции, дизъюнкции и др.). Каждое такое элементарное действие, выполняемое в одном из узлов ОУ в течение одного тактового периода, называется **микрооперацией**.

В определенные тактовые периоды одновременно могут выполняться несколько микроопераций, например R2 <– 0, Сч <– (Сч) - 1. Такая совокупность одновременно выполняемых микроопераций называется **микрокомандой**, а весь набор микрокоманд, предназначенный для решения определенной задачи, - **микропрограммой**.

18. Память с безадресным обращением. Ассоциативная память. Применение.

Ассоциативная память (АП) или Ассоциативное запоминающее устройство (АЗУ) является особым видом машинной памяти, используемой в приложениях очень быстрого поиска. Известна также как память, адресуемая по содержимому, ассоциативное запоминающее устройство, контентно-адресуемая память или ассоциативный массив, хотя последний термин чаще используется в программировании для обозначения структуры данных.

Адресуемая содержанием память часто используется в компьютерных сетевых устройствах. Например, когда сетевой коммутатор (switch) получает

фрейм данных на один из его портов, это обновляет внутреннюю таблицу с источником MAC-адреса фрейма и порта, на который он был получен. Потом он ищет MAC-адрес назначения в таблице, чтобы определить, на какой порт фрейм должен быть отправлен, и отсылает его на этот порт. Таблица MAC-адресов обычно реализована на двоичной АП, таким образом порт назначения может быть найден очень быстро, уменьшая время ожидания коммутатора.

19. Два подхода к построению процессоров. Принцип схемной логики. Преимущества, недостатки.

Логические элементы — устройства, предназначенные для обработки информации в цифровой форме. Физически логические элементы могут быть выполнены механическими, электромеханическими (на электромагнитных реле), электронными (в частности, на диодах или транзисторах), пневматическими, гидравлическими, оптическими и другими.

Среди преимуществ можно выделить следующие:

1. при линейном построении текстовой информации нередко бываетсложно определить структуру изучаемого явления, выделить существенные связи между его компонентами. Это затруднение в значительной мере преодолевается при замене словесного описания таблицами, а лучше - схемами;
2. такое преобразование учебного текста представляет собой эффективный прием, активизирующий мышление учащегося;
3. способ схематической визуализации информации помогает более глубокому овладению дисциплинами, способствует формированию более рациональных приемов работы с учебным материалом вообще;
4. в ряде психологических исследований выявлено, что структурирование (под структурированием понимается установление взаимного расположения частей составляющих целое) и схематизации (под схематизацией понимается изображение или описание чего-либо в основных чертах) текстовой информации являются важнейшими компонентами мнемического действия (действие, целью которого является произвольное запоминание того или иного материала), составляющего основу процесса запоминания;
5. наглядно - образная форма представления информации способствует лучшему ее запоминанию;
6. как показывает опыт, представление учебной информации в системе структурно - логических схем выступает достаточно эффективным

средством организации и активизации самостоятельной работы обучающихся;

7. структурирование материала помогает быстрее сформировать у учащегося целостную картину изучаемого предмета, что создает основу для дальнейшей организации процесса усвоения учебного предмета до необходимой глубины.

Среди недостатков можно выделить следующие:

1. любой схематизм способствует некоторой упрощенности понимания чего-либо. Это может создать иллюзию у учащихся, что для изучения предмета вполне достаточно изображенного материала;
2. абсолютизация учебных пособий, построенных по принципу логико - структурного моделирования, может негативно повлиять на формирование профессионального мышления и языка;
3. отдельные части материала (особенно гуманитарных дисциплин) очень трудно «поддаются» структуризации, что затрудняет разработку целостного курса с помощью схем;
4. схематическая форма представления учебного материала может не в полной мере соответствовать его «кодируемому» содержанию. Например, знания о процессах изучаемых явлений (формированиея навыка, развитие мышления и т.п.) «требуют» другой формы схематизации, чем просто знания о фактах, явлениях, их свойствах и т.п.

20. Память с безадресным обращением. Пример ассоциативного ЗУ – продажа калькуляторов.

В настоящее время ассоциативные запоминающие устройства (АЗУ) в основном используются как вспомогательные устройства в некоторых вычислительных системах и позволяют повысить их производительность, обеспечивая процессору более простой доступ к операндам. В АЗУ поиск нужной информации производится не по адресу, а по ее содержанию, по ассоциативному признаку. При этом поиск по ассоциативному признаку или последовательно по отдельным разрядам этого признака может быть осуществлен для всех ячеек запоминающего массива как последовательно, так и параллельно во времени. Во многих случаях ассоциативный поиск позволяет существенно упростить и ускорить обработку данных. Это достигается за счет того, что в памяти этого типа операция считывания совмещена с выполнением ряда логических операций

Рассмотрим работу АЗУ на примере продажи микрокалькуляторов в специализированном магазине вычислительной техники. Покупателю предоставляется выбор микрокалькуляторов по четырем признакам:

-область применения - бухгалтерские расчеты, словари, инженерные расчеты;

-набор выполняемых операций - количество операций;

-тип источника питания - от сети, от аккумуляторной батареи;

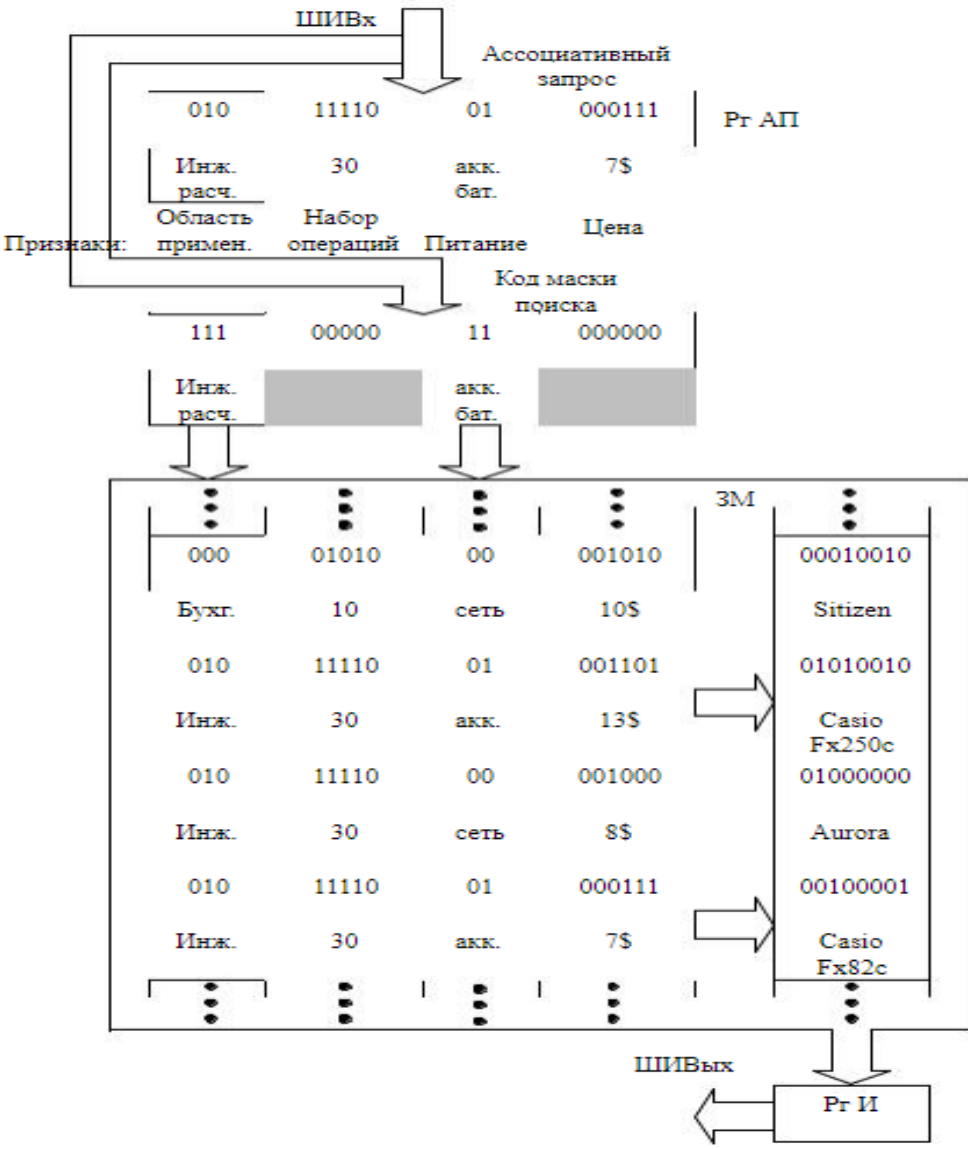
-цена.

Информация о калькуляторах хранится в запоминающем модуле ЗМ (рисунок). Работа по автоматизированному поиску типа калькулятора, удовлетворяющего требованиям покупателя по четырем техническим параметрам в порядке их приоритета, выполняется в следующем порядке:

1. Покупатель вводит информацию о необходимой ему покупке. Это соответствует появлению на входной адресной шине ШАВх и в регистре ассоциативного признака РгАП двоичного кода 4-х признаков выбора микрокалькулятора.

2.Покупатель информирует ЭВМ о том, что вначале выбор будет осуществляться только по двум признакам: область применения и источник питания, при этом в РгМаски появляется соответствующий код маски.

3.Производится операция считывания из ЗМ в информационный регистр РгИ всех слов, удовлетворяющих ассоциативному признаку, например, поочередное считывание в порядке возрастания цены. В такой последовательности на экран дисплея будет выдаваться информация о тех типах калькуляторов, которые отвечают введенным признакам запроса.



21. Синтез устройства, выполняющего умножение двоичных чисел.

22. Динамическое распределение памяти. Два способа динамического распределение памяти. Использование базовых регистров. Недостатки.

Один из способов динамического распределения памяти ос-нован на использовании базовых регистров. Операционная система каждой пользовательской программе ставит в соответ-ствие свой базовый адрес. Базовые адреса обрабатываемых программ находятся в общих регистрах. При выполнении про-граммы реальный или физический адрес образуется суммирова-нием базового и относительного адресов. При динамическом распределении памяти с помощью базовых регистров программа (или, по крайней мере, та часть ее, адрес которой преобразуется с помощью одного и того же базового адреса) должна распола-гаться в последовательных ячейках

и вводиться в ОП целиком, хотя в ближайшем цикле активности может потребоваться лишь небольшой фрагмент программы.

При рассмотренном способе динамического распределения памяти *свободная память* может состоять из *несвязных областей* (*фрагментация* памяти) и для ввода нужной программы может понадобиться сдвиг содержимого памяти. Это можно проиллюстрировать примером на рисунке.

+Первоначально ОП распределяется между программами A, B, C, D. Программы A и D в данный момент наименее активны и могут рассматриваться как кандидаты на удаление во внешнюю память. Если вновь вводимая программа E больше любой из программ A и D, то для ее размещения в памяти необходимо сдвинуть программы B и C "вверх" или "вниз". Это перемещение связано с потерей времени. Более того, в ряде прежних операционных систем такое перемещение требовало выполнения заново операции редактирования связей в программе и новой загрузки программы.

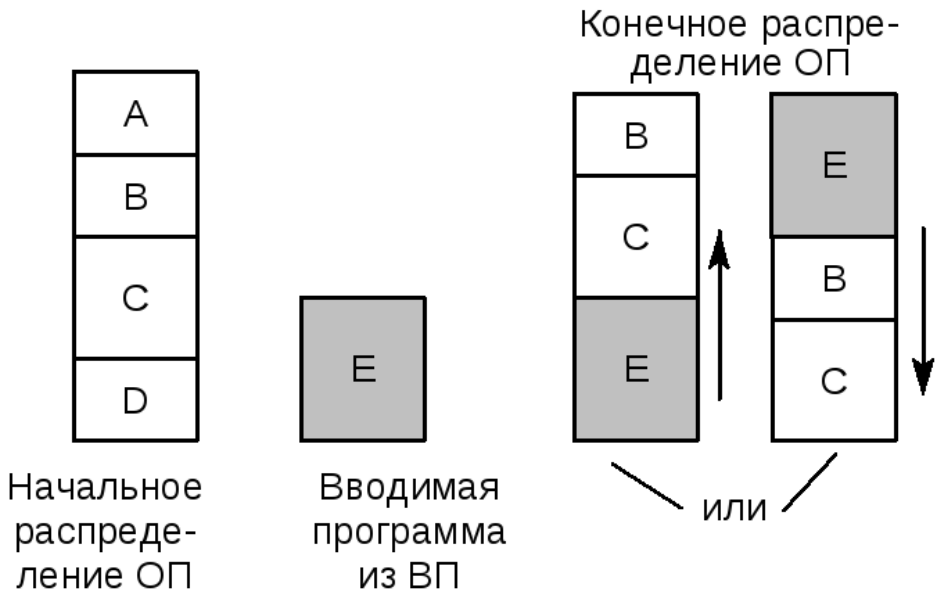


Рис. 9.11. Пример распределения памяти

23. Цифровые автоматы. Автоматы Мура, автоматы Мили.

Автомат Мура:

Автомат Мура (абстрактный автомат второго рода) в теории вычислений — конечный автомат, выходное значение сигнала в котором зависит лишь от текущего состояния данного автомата, и не зависит напрямую, в отличие от автомата Мили, от входных значений. Автомат Мура назван в честь описавшего его свойства Эдварда Ф. Мура.

Автомат Мили:

Автомат Мили (Mealy machine) — конечный автомат, выходная последовательность которого (в отличие от автомата Мура) зависит от состояния автомата и входных сигналов. Это означает, что в графе состояний каждому ребру соответствует некоторое значение (выходной символ). В вершины графа автомата Мили записываются выходящие сигналы, а дугам графа приписывают условие перехода из одного состояния в другое, а также входящие сигналы. Назван именем Джорджа Мили, учёного в области математики и компьютерных наук, придумавшего этот автомат.

24. Динамическое распределение памяти. Два способа динамического распределение памяти. Использование виртуальной памяти. Достоинства.

Динамическое распределение памяти — способ выделения оперативной памяти компьютера для объектов в программе, при котором выделение памяти под объект осуществляется во время выполнения программы.

Виртуальная память представляет собой совмещение оперативной памяти и временного хранилища файлов на жестком диске. В случае если памяти ОЗУ недостаточно, данные перемещаются во временное хранилище, называемое файлом подкачки. Для выполняющейся программы данный метод полностью прозрачен и не требует дополнительных усилий со стороны, однако реализация этого метода требует как аппаратной поддержки, так и поддержки со стороны операционной системы

Преимущества виртуальной памяти

- Освобождает программиста от необходимости вручную управлять загрузкой частей программы в память и согласовывать использование памяти с другими программами;
- Предоставляет программам больше памяти, чем физически установлено в системе;
- В многозадачных системах изолирует выполняющиеся программы друг от друга путём назначения им непересекающихся адресных пространств;
- Приложения исполняются в своём адресном пространстве и не мешают друг другу;
- Повышается безопасность за счёт защиты памяти.

25. Синтез устройства управления в форме автомата Мура. Синтез устройства управления в форме автомата Мили.

26. Виртуальная память со страничной организацией. Достоинства и недостатки.

Виртуальная память - это совокупность программно-аппаратных средств, позволяющих пользователям писать программы, размер которых превосходит имеющуюся оперативную память, В наиболее простом и наиболее часто используемом случае страничной виртуальной памяти виртуальная память каждого процесса и физическая основная память представляются состоящими из наборов блоков или страниц одинакового размера. В большинстве современных компьютеров со страничной организацией виртуальной памяти все таблицы страниц хранятся в основной памяти, а быстрота доступа к элементам таблицы текущей виртуальной памяти достигается за счет наличия сверхбыстродействующей буферной памяти (кэша). Во-первых одним из преимуществ ВП с СО является достаточно большой объём прямо адресуемой памяти. Действительно объем памяти может исчисляться сотнями мегабайт (и даже гигабайтами). Размер виртуальной памяти целиком зависит от объёма накопителя на [жестком] магнитном диске. Созданный SWAPфайл размещается на диске и эмулирует оперативную память. При этом пользователь не задумывается о том куда будет помещен “кусок” его программы с которой он только что отработал. Таким образом, ещё одним преимуществом ВП с СО является то, что программы пользователя могут размещаться в любых свободных страницах. И наконец, одним из важнейших преимуществ ВП с СО (то, ради чего, собственно и была изобретена виртуальная память) повышение уровня мультипрограммной работы. Как было сказано выше, эта цель была одной из самых главных. С организацией ВП с СО пользователь получил реальную возможность загружать в память большее количество программ для того чтобы машина обрабатывала программы сразу (в действительности процессор устанавливает приоритет для каждой программы, находящейся в памяти, и далее в соответствии с приоритетом выделяет определённое количество времени на реализацию каждой программы или команды). Сам процессор постоянно “занят” каждый машинный такт выполняет определённую программу. Метод организации виртуальной памяти со страничной организацией значительно повысил эффективность работы с машиной.А недостатком виртуальной памяти пожалуй является то количество времени, которое машина тратит на обращение к внешней памяти. Извлечь необходимую информацию из ячеек оперативной памяти не представляет особого труда и больших затрат времени. Совсем иначе обстоит дело с диском: для того чтобы найти необходимую информацию, нужно сначала “раскрутить” диск, потом найти необходимую дорожку, в дорожке найти сектор, кластер, далее считать побитовую информацию в ОП. Все это требует времени и, порой если при методе случайного удаления страниц *,

процессору понадобятся сразу несколько страниц, хранящихся во внешней памяти, большого времени. Следующий недостаток скорее относится к вопросу о технической характеристике компьютера: наличие сверхоперативной памяти (СОП). Как было сказано выше, СОП.

Методы своппирования страниц имеют большую ёмкость и достаточно высокое быстродействие. СОП используется для хранения управляющей информации, служебных кодов, а также информации к которой осуществляется наиболее частое обращение в процессе выполнения программы. Этот недостаток в работе с ВП к счастью можно ликвидировать. Что касается технической характеристики есть ли в микросхемах оперативной памяти дополнительные интегральные схемы, которые являются запоминающими устройствами СОП? Если есть, то проблема с СОП решена, а если нет..? Тогда, благодаря достижениям в области компьютерной технологии, могут использоваться драйверы, резервирующие маленькую область ОП для имитирования СОП(стандартные операционные процедуры).

27. Динамическое распределение памяти. Способы управления виртуальной памятью

Наиболее распространенными реализациями виртуальной памяти является страничное, сегментное и странично-сегментное распределение памяти, а также свопинг.

Страничное распределение

Виртуальное адресное пространство каждого процесса делится на части одинакового, фиксированного для данной системы размера, называемые виртуальными страницами. В общем случае размер виртуального адресного пространства не является кратным размеру страницы, поэтому последняя страница каждого процесса дополняется фиктивной областью.

Сегментное распределение

При страничной организации виртуальное адресное пространство процесса делится механически на равные части. Это не позволяет дифференцировать способы доступа к разным частям программы (сегментам), а это свойство часто бывает очень полезным. Например, можно запретить обращаться с операциями записи и чтения в кодовый сегмент программы, а для сегмента данных разрешить только чтение. Кроме того, разбиение программы на "осмысленные" части делает принципиально возможным разделение одного сегмента несколькими процессами. Например, если два процесса используют одну и ту же математическую подпрограмму, то в оперативную память может быть загружена только одна копия этой подпрограммы. Виртуальное адресное пространство процесса делится на сегменты, размер которых

определяется программистом с учетом смыслового значения содержащейся в них информации. Отдельный сегмент может представлять собой подпрограмму, массив данных и т.п. Иногда сегментация программы выполняется по умолчанию компилятором.

Странично-сегментное распределение

Как видно из названия, данный метод представляет собой комбинацию страничного и сегментного распределения памяти и, вследствие этого, сочетает в себе достоинства обоих подходов. Виртуальное пространство процесса делится на сегменты, а каждый сегмент в свою очередь делится на виртуальные страницы, которые нумеруются в пределах сегмента. Оперативная память делится на физические страницы. Загрузка процесса выполняется операционной системой постранично, при этом часть страниц размещается в оперативной памяти, а часть на диске. Для каждого сегмента создается своя таблица страниц, структура которой полностью совпадает со структурой таблицы страниц, используемой при страничном распределении. Для каждого процесса создается таблица сегментов, в которой указываются адреса таблиц страниц для всех сегментов данного процесса. Адрес таблицы сегментов загружается в специальный регистр процессора, когда активизируется соответствующий процесс.

Свопинг

При свопинге, в отличие от рассмотренных ранее методов реализации виртуальной памяти, процесс перемещается между памятью и диском целиком, то есть в течение некоторого времени процесс может полностью отсутствовать в оперативной памяти. Существуют различные алгоритмы выбора процессов на загрузку и выгрузку, а также различные способы выделения оперативной и дисковой памяти загружаемому процессу.

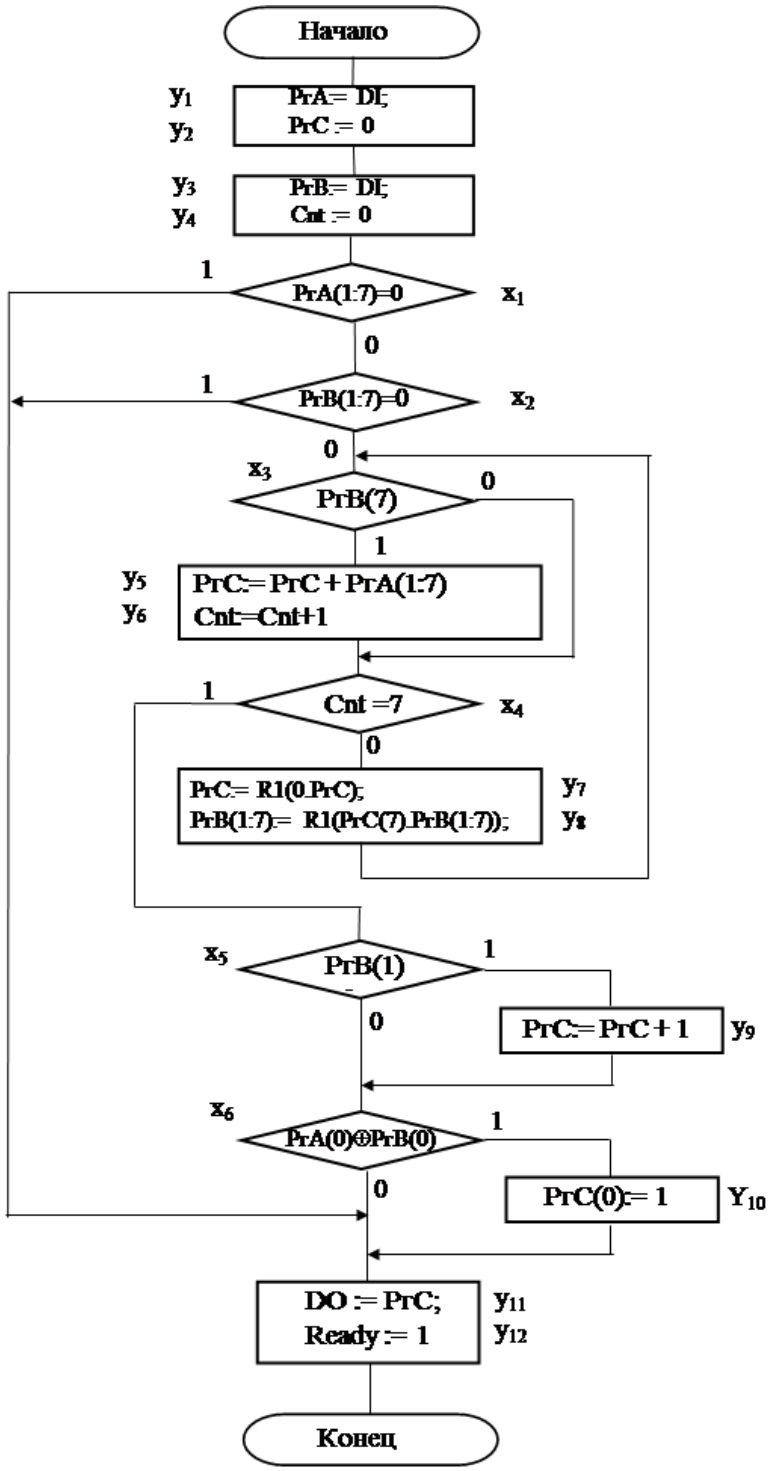
28. Два подхода к построению процессоров. Принцип программируемой логики. Преимущества, недостатки.

Устройства, основанные на таком принципе схемной логики, способны обеспечивать наивысшее быстродействие при заданном типе технологии элементов. Недостаток этого принципа построения процессора состоит в трудности использо-вания БИС и СБИС. Это связано с тем, что при использования схемного принципа каждый разрабатываемый процессор окажется индивидуаль-ным по схемному построению и потребует изготовления индивидуаль-ного типа БИС. Тогда выпускаемые промышленностью БИС окажутся узкоспециализированными, число выпускаемых типов БИС будет боль-шим, а потребность в каждом типе БИС окажется низкой. Выпуск многих типов БИС малыми сериями по каждому типу для промышлен-ности окажется экономически невыгодным.

Эти обстоятельства заставляют обратиться к другому подходу в проектировании цифровых устройств, основанному на использовании принципа программируемой логики. Этот подход предполагает постро-ение с использованием одной или нескольких БИС некоторого универ-сального устройства, в котором требуемое функционирование (т.е. специализация устройства на выполнение определенных функций) обеспечивается занесением в память устройства определенной программы (или микропрограммы). В зависимости от введенной программы такое универсальное управляющее устройство способно обеспечивать требуе-мое управление операционным устройством при решении самых разно-образных задач. В этом случае число типов БИС, необходимых для построения управляющего устройства, окажется небольшим, а потреб-ность в БИС каждого типа высокой, что обеспечит целесообразность их выпуска промышленностью.

29. Структура виртуальной памяти при сегментном распределении. Построение микропрограммы для операции умножения.

Сегментное распределение обеспечивает большее удобство при программировании и повышение эффективности использования памяти особенно в случае программ, состоящих из нескольких массивов - подпрограмм, одной или нескольких секций данных



30. Сегментно – страничная организация памяти.

Данный метод представляет собой комбинацию страничного и сегментного распределения памяти и направлен на реализацию достоинств обоих подходов.

Виртуальное пространство процесса делится на сегменты, что позволяет определять разные права доступа к разным частям кодов и данных программы.

Перемещение между памятью и диском осуществляется страницами. Для этого каждый виртуальный сегмент и физическая память делятся на страницы равного размера. Виртуальные страницы нумеруются в пределах виртуального адресного пространства каждого процесса.

Таким образом, виртуальный адрес содержит указание на номер соответствующего сегмента (s) и смещение относительно начала сегмента, которое, в свою очередь, может состоять из двух полей: страница (p – Page) и индекс (I– Index). То есть весь адрес состоит из трех компонентов: s, p, I.

Физические страницы нумеруются в пределах оперативной памяти.

При создании процесса в память загружается только часть страниц, остальные загружаются по мере необходимости. Время от времени система выгружает уже ненужные страницы, освобождая память для новых. Операционная система ведет для каждого процесса таблицу страниц, в которой указывается соответствие виртуальных страниц физическим.

Начальные адреса таблицы сегментов и таблицы страниц процесса являются частью его контекста. При активизации процесса эти адреса загружаются в специальные регистры процессора и используются механизмом преобразования адресов.

Преобразование виртуального адреса в физический происходит в два этапа:

1) Работает механизм сегментации. Исходный виртуальный адрес, заданный в виде пары (№ сегмента, смещение), преобразуется в линейный виртуальный адрес. Для этого на основании базового адреса таблицы сегментов и номера сегмента вычисляется адрес дескриптора сегмента. Анализируются поля дескриптора, и выполняется проверка возможности выполнения заданной операции. Если доступ к сегменту разрешен, то вычисляется линейный виртуальный адрес путем сложения начального адреса сегмента (извлеченного из дескриптора) и смещения (заданного в исходном виртуальном адресе).

2)Работает страничный механизм. Полученный линейный виртуальный адрес преобразуется в искомый физический адрес.

31. Сравнение быстродействия управляющих устройств, построенных по принципу программируемой логики.

Устройства, построенные на таком принципе системной логики, способны обеспечивать наивысшее быстродействие при заданном типе технологии элементов. Недостаток этого принципа построения МКУ состоит в

невозможности "перестройки" структуры устройств и систем при необходимости изменения или расширения их функциональных возможностей.

32. Вычислительные системы. Определение. Два класса архитектур

Вычислительная система (ВС) — совокупность взаимосвязанных и взаимодействующих процессоров или ЭВМ, периферийного оборудования и программного обеспечения, предназначенная для сбора, хранения, обработки и распределения информации.

Классы архитектуры:

- Классическая архитектура (архитектура Дж. фон Неймана) — одно АЛУ.
- Многопроцессорная архитектура. Наличие в компьютере нескольких процессоров означает, что параллельно может быть организовано много потоков данных и команд, т. е. могут выполняться несколько фрагментов одной задачи.
- Многомашинная вычислительная система. Несколько процессоров, входящих в вычислительную систему, не имеют общей оперативной памяти, а каждый имеет свою локальную. Каждый компьютер в многомашинной системе имеет классическую архитектуру, и такая система применяется достаточно широко. Однако эффект от применения многомашинной системы может быть получен только при решении задач, имеющих специальную структуру, которая должна разбиваться на столько слабосвязанных подзадач, сколько компьютеров в системе.

33. Базовые представления об архитектуре ЭВМ. Типы архитектур. Архитектура “звезда”.

Архитектурой компьютера считается его представление на неко-тором общем уровне, включающее описание пользовательских воз-можностей программирования, системы команд, системы адреса-ции, организации памяти и т. д. Архитектура определяет принципы действия, информационные связи и взаимное соединение основных логических узлов компьютера: процессора, оперативного запоминаю-щего устройства (ОЗУ, ОП), внешних ЗУ и периферийных устройств. Общность архитектуры разных компьютеров обеспечивает их со-вместимость с точки зрения пользователя.

Структура компьютера — это совокупность его функциональ-ных элементов и связей между ними. Элементами могут быть самые различные устройства — от основных логических узлов компьютера до простейших схем. Структура компьютера графически представля-ется в виде структурных схем, с помощью которых можно дать опи-сание компьютера на любом уровне детализации

Типы архитектур:

- CISC (англ. complex instruction set computing) — архитектура с полным набором команд. Такие процессоры выполняют все команды, простые и сложные, за большое количество тактов. Команд в таких процессорах много, и компиляторы верхнего уровня редко используют все команды.
- RISC (англ. reduced instruction set computing) — архитектура с сокращённым набором команд. Такие процессоры работают быстрее, чем с CISC-архитектурой, за счёт упрощения архитектуры и сокращения количества команд, но для выполнения сложной команды она составляется из набора простых, что увеличивает время выполнения команды (за большее количество тактов).
- MISC (англ. minimal instruction set computing) — архитектура с минимальным набором команд. Такие процессоры имеют минимальное количество команд, все команды простые и требуют небольшого количества тактов на выполнение, но если выполняются сложные вычисления, например, с числами с плавающей запятой, то такие команды выполняются за большое количество тактов, превышающее CISC- и RISC-архитектуры.
- VLIW (англ. very long instruction word — «очень длинная машинная команда») — архитектура с длинной машинной командой, в которой указывается параллельность выполнения вычислений. Такие процессоры получили широкое применение в цифровой обработке сигналов.

Архитектура “звезда”:

Звезда — базовая топология компьютерной сети, в которой все компьютеры сети присоединены к центральному узлу (обычно коммутатор), образуя физический сегмент сети. Подобный сегмент сети может функционировать как отдельно, так и в составе сложной сетевой топологии (как правило, «дерево»). Весь обмен информацией идет исключительно через центральный компьютер или агрегат, на который таким способом возлагается очень большая нагрузка, поэтому ничем другим, кроме сети, он заниматься

не может. Как правило, именно центральный компьютер или агрегат является самым мощным в сетевом отношении, и именно на него возлагаются все функции по управлению сетью и передаче данных.

Активная звезда:

В центре сети находится сервер или другое активное сетевое оборудование.

Пассивная звезда:

В центре сети с данной топологией содержится концентратор, или коммутатор все пользователи в сети равноправны.

34. Микроконтроллеры. Назначение, области применения.

Микроконтроллеры предназначены для управления разными электронными приборами и устройствами. Они используются не только в компьютерах, но и в различной бытовой технике, в роботах на производстве, в телевизорах, в оборонной промышленности.

35. Базовые представления об архитектуре ЭВМ. Типы архитектур. Иерархическая архитектура.

Иерархическая архитектура (рисунок*)* — ЦУ соединено с периферийными процессорами (вспомогательными процессорами, каналами, канальными процессорами), управляющими в свою очередь контроллерами, к которым подключены группы ВУ (системы IBM 360-375, ЕС ЭВМ);

36. Вычислительные системы. Определение. Цели создания.

Вычислительные системы - совокупность аппаратно-программных средств, образующих единую среду, предназначенную для решения задач обработки информации (вычислений).

Создание ВС преследует следующие основные цели:

- повышение производительности системы за счет ускорения процессов обработки данных;

- повышение надежности и достоверности вычислений;
- предоставление пользователям дополнительных сервисных услуг и т. д;

37. Базовые представления об архитектуре ЭВМ. Типы архитектур. Магистральная архитектура.

Магистральная структура (общая шина — unibas, рисунок) — процессор (процессоры) и блоки памяти (ОП) взаимодействуют между собой и с ВУ (контроллерами ВУ) через внутренний канал, общий для всех устройств (машины DEC, ПЭВМ IBM PC-совместимые).

38. Система прерываний. Виды прерываний. Источники прерываний.

Система прерываний позволяет компьютеру приостановить текущие действия и переключиться на другие в ответ на поступивший запрос, например, на нажатие клавиши на клавиатуре.

Виды и источники делятся на:

- асинхронные, или внешние (аппаратные) — события, которые исходят от внешних аппаратных устройств (например, периферийных устройств) и могут произойти в любой произвольный момент: сигнал от таймера, сетевой карты или дискового накопителя, нажатие клавиш клавиатуры, движение мыши. Факт возникновения в системе такого прерывания трактуется как запрос на прерывание (Interrupt request, IRQ) - устройства сообщают, что они требуют внимания со стороны ОС;
- синхронные, или внутренние — события в самом процессоре как результат нарушения каких-то условий при исполнении машинного кода: деление на ноль или переполнение стека, обращение к недопустимым адресам памяти или недопустимый код операции;
- программные (частный случай внутреннего прерывания) — инициируются исполнением специальной инструкции в коде программы. Программные прерывания, как правило, используются для

обращения к функциям встроенного программного обеспечения (firmware), драйверов и операционной системы.

39. Абстрактное центральное устройство. Цикл процессора, такт работы процессора.

Цикл процессора — период времени, за который осуществляется выполнение команды исходной программы в машинном виде; состоит из нескольких *тактов.*

Такт работы процессора — промежуток времени между соседними импульсами (tick of the internal clock) *генератора тактовых импульсов*, частота которых есть *тактовая частота процессора. Такт процессора (такт синхронизации)* — квант времени, в течение которого осуществляется элементарная операция — выборка, сравнение, пересылка данных.

52. Классификация архитектуры вычислительных систем с параллельной обработкой данных. ОКОД.

Вычислительная система с **о**диночным потоком **к**оманд и **о**диночным потоком **д**анных
(SISD, **S**ingle **I**nstruction stream over a **S**ingle **D**ata stream).

SISD (англ. *Single Instruction, Single Data*) или **ОКОД** (*Одиночный поток Команд, Одиночный поток Данных*) — архитектура компьютера, в которой один процессор выполняет один поток команд, оперируя одним потоком данных. Относится к фон-Неймановской архитектуре. Один из классов вычислительных систем в классификации Флинна.

SISD-компьютеры — это обычные, «традиционные» последовательные компьютеры, в которых в каждый момент времени выполняется лишь одна операция над одним элементом данных (числовым или каким-либо другим значением). Большинство персональных ЭВМ до последнего времени, например, попадает именно в эту категорию. Иногда сюда относят и некоторые типы векторных компьютеров, это зависит от того, что понимать под потоком данных.

53. Абстрактное центральное устройство. Цикл выполнения команды

Цикл выполнения команды – это последовательность действий, которая совершается процессором при выполнении одной машинной команды. При выполнении каждой машинной команды процессор должен выполнить как минимум три действия: выборку, декодирование и выполнение.Если в

команде используется операнд, расположенный в оперативной памяти, то процессору придётся выполнить ещё две операции: выборку операнда из памяти и запись результата в память. Ниже описаны эти пять операций.

- **Выборка команды.** Блок управления извлекает команду из памяти (из очереди команд), копирует её во внутреннюю память процессора и увеличивает значение счётчика команд на длину этой команды (разные команды могут иметь разный размер).
- **Декодирование команды.** Блок управления определяет тип выполняемой команды, пересылает указанные в ней операнды в АЛУ и генерирует электрические сигналы управления АЛУ, которые соответствуют типу выполняемой операции.
- **Выборка операндов.** Если в команде используется операнд, расположенный в оперативной памяти, то блок управления начинает операцию по его выборке из памяти.
- **Выполнение команды.** АЛУ выполняет указанную в команде операцию, сохраняет полученный результат в заданном месте и обновляет состояние флагов, по значению которых программа может судить о результате выполнения команды.
- **Запись результата в память.** Если результат выполнения команды должен быть сохранён в памяти, блок управления начинает операцию сохранения данных в памяти.

54. Классификация архитектуры вычислительных систем с параллельной обработкой данных. ОКМД.

ОКМД — Вычислительная система с **о**диночным потоком **к**оманд и **м**ножественным потоком **д**анных (SIMD, **S**ingle **I**nstruction, **M**ultiple **D**ata).

SIMD (англ. *single instruction, multiple data* — **одиночный поток команд, множественный поток данных**, **ОКМД**) — принцип компьютерных вычислений, позволяющий обеспечить параллелизм на уровне данных. Один из классов вычислительных систем в классификации Флинна.

SIMD-компьютеры состоят из одного командного процессора (управляющего модуля), называемого контроллером, и нескольких модулей обработки данных, называемых процессорными элементами. Управляющий модуль принимает, анализирует и выполняет команды. Если в команде встречаются данные, контроллер рассылает на все процессорные элементы команду, и эта команда выполняется на нескольких или на всех процессорных элементах. Каждый процессорный элемент имеет свою

собственную память для хранения данных. Одним из преимуществ данной архитектуры считается то, что в этом случае более эффективно реализована логика вычислений. До половины логических инструкций обычного процессора связано с управлением выполнением машинных команд, а остальная их часть относится к работе с внутренней памятью процессора и выполнению арифметических операций. В SIMD-компьютере управление выполняется контроллером, а «арифметика» отдана процессорным элементам.

Векторные процессоры также использовали принцип SIMD, одной командой могли обрабатываться векторы размером до нескольких тысяч элементов.

55. Режимы работы процессора. Граф переходов между режимами процессора.

Режим работы процессора (CPU mode) — состояние процессора, определяющее его поведение при выполнении различных команд и возможность доступа к различным данным.

По способу адресации памяти (на примере x86):

- Реальный режим: обращение к оперативной памяти происходит по реальным (действительным) адресам. Набор доступных операций не ограничен, защита памяти не используется.
- Защищённый режим: обращение к памяти происходит по виртуальным адресам с использованием механизмов защиты памяти. Набор доступных операций определяется уровнем привилегий.

По уровню привилегий:

- Режим пользователя (прикладной): минимальный уровень привилегий, разрешены только операции с данными и переходы в пределах адресного пространства пользователя. Все остальные операции либо игнорируются, либо с помощью механизма обработки исключений вызывают переключение в привилегированный режим и передачу управления ядру операционной системы для выполнения специальных функций (например, отображения данных на дисплее) или аварийного завершения потока управления.
- Привилегированный режим (режим ядра): наравне с операциями режима пользователя, разрешены дополнительные операции — запрет или разрешение прерываний, доступ к портам ввода-вывода,

специальным регистрам процессора (например, для настройки блока управления памятью).

Защищённый режим (режим защищённой виртуальной адресации) — режим работы x86-совместимых процессоров. Частично был реализован уже в процессоре 80286, но там существенно отличался способ работы с памятью, так как процессоры ещё были 16-битными и не была реализована страничная организация памяти. Первая 32-битная реализация защищённого режима — процессор Intel 80386. Применяется в совместимых процессорах других производителей. Данный режим используется в современных многозадачных операционных системах, Windows, Linux, macOS.

56. Классификация архитектуры вычислительных систем с параллельной обработкой данных. МКОД.

МКОД — Вычислительная система со множественным потоком команд и одиночным потоком данных (MISD, Multiple Instruction Single Data).

MISD-Архитектура (англ. ***M**ultiple **I**nstruction stream, **S**ingle **D**ata stream*, **М**ножественный поток **К**оманд, **О**диночный поток Данных, **МКОД**) — тип архитектуры параллельных вычислений, где несколько функциональных модулей (два или более) выполняют различные операции над одними данными. Один из классов вычислительных систем в классификации Флинна.

Отказоустойчивые компьютеры, выполняющие одни и те же команды избыточно с целью обнаружения ошибок, как следует из определения, принадлежат к этому типу. К этому типу иногда относят конвейерную архитектуру, но не все с этим согласны, так как данные будут различаться после обработки на каждой стадии в конвейере. Некоторые относят систолический массив процессоров к архитектуре MISD.

Было создано немного ЭВМ с MISD-архитектурой, поскольку MIMD и SIMD чаще всего являются более подходящими для общих методик параллельных данных. Они обеспечивают лучшее масштабирование и использование вычислительных ресурсов, чем архитектура MISD.

57. Режим SMM. Управление энергопотреблением.

Режим системного управления (System Management Mode, SMM) — режим исполнения на процессорах x86/x86-64, при котором приостанавливается исполнение другого кода (включая операционные

системы и гипервизор), и запускается специальная программа, хранящаяся в SMRAM в наиболее привилегированном режиме.

Среди возможных применений SMM:

- Обработка системных ошибок, таких как ошибки памяти и чипсета;
- Функции защиты, например выключение процессоров при сильном перегреве;
- Глубокие уровни энергосбережения;
- Управление питанием — например, схемами изменения напряжения;
- Эмуляция периферии, которая не была реализована на материнской плате или реализация которой содержит ошибки;
- Эмуляция мыши и клавиатуры PS/2 при использовании таких же устройств с интерфейсом USB;
- Централизованная конфигурация системы, например на ноутбуках Toshiba и IBM;
- Обход систем защиты, встроенных в ОС;
- Запуск высокопривилегированных руткитов, как было предложено на Black Hat 2008;
- Эмуляция или передача вызовов на модуль Trusted Platform Module (TPM);

Система управления энергопотреблением (EMS) компании Schneider Electric предоставляет операторам систем питания более подробную информацию о системах питания и распределения, а также возможность работы либо с автономной системой, либо с системой, полностью интегрированной в систему управления распределительными сетями (ADMS) компании Schneider Electric. С помощью приложений для оценки состояния, распределения нагрузки, оптимального потока мощности, анализа последствий аварийных ситуаций, расчета сбоев, оптимального изменения конфигураций, показателей производительности и стабильности напряжения система управления энергопотреблением (EMS) компании Schneider Electric предоставляет коммунальным компаниям возможность улучшения визуализации, эксплуатации, оптимизации и обслуживания сетей питания и распределения.

58. Классификация архитектуры вычислительных систем с параллельной обработкой данных. МКМД.

МКМД — Вычислительная система со множественным потоком команд и множественным потоком данных.(MIMD, Multiple Instruction Multiple Data).

MIMD (англ. Multiple Instruction stream, Multiple Data stream — Множественный поток Команд, Множественный поток Данных, сокращённо МКМД) — концепция архитектуры компьютера, используемая для достижения параллелизма вычислений. Один из классов вычислительных систем в классификации Флинна.

Машины имеют несколько процессоров, которые функционируют асинхронно и независимо. В любой момент различные процессоры могут выполнять различные команды над различными частями данных. MIMD-архитектуры могут быть использованы в целом ряде областей, таких, как системы автоматизированного проектирования / автоматизированное производство, моделирование, а также коммуникатор связей (англ. communication switches). MIMD-машины могут быть либо с общей памятью, либо с распределяемой памятью. Эта классификация основана на том, как MIMD-процессоры получают доступ к памяти. Этот класс предполагает, что в вычислительной системе есть несколько устройств обработки команд, объединённых в единый комплекс и работающих каждое со своим потоком команд и данных.

Обработка разделена на несколько потоков, каждый с собственным аппаратным состоянием процессора, в рамках единственного определённого программным обеспечением процесса или в пределах множественных процессов. Поскольку система имеет несколько потоков, ожидающих выполнения (системные или пользовательские потоки), эта архитектура эффективно использует аппаратные ресурсы.