А1-1 В чем преимущества и недостатки аккумуляторной архитектуры?

***Преимущества аккумуляторной архитектуры:***

*Эффективное использование энергии.*

*Подходит для мобильных устройств.*

*Хранение возобновляемой энергии.*

*Снижение выбросов парниковых газов.*

***Недостатки аккумуляторной архитектуры:***

*Ограниченная емкость и время работы.*

*Истощение аккумуляторов со временем.*

*Тяжелые и дорогостоящие.*

*Эффективность зарядки и разрядки.*

*Ограниченная жизнь аккумуляторов.*

А2-1 Какие преимущества стековой архитектуры вы можете перечислить? Какие средства используются для ускорения работы стека в ВМ со стековой архитектурой?

**Преимущества стековой архитектуры:**

*Компактность и универсальность: Стековая архитектура позволяет создавать компактные и универсальные виртуальные машины (ВМ).*

*Простота портирования: Программы для стековых ВМ могут быть легко портированы на разные аппаратные платформы.*

*Безопасность: Стековые ВМ обеспечивают изоляцию между приложениями и операционной системой, что улучшает безопасность.*

*Средства для ускорения работы стека в ВМ со стековой архитектурой включают:*

*JIT-компиляция: Динамическая компиляция байт-кода в нативный код для улучшения производительности.*

*Кэширование: Использование кэшей для хранения промежуточных результатов выполнения кода.*

*Оптимизация структур данных: Оптимизация использования структур данных для ускорения выполнения программ.*

*Многозадачность: Параллельное выполнение ВМ и оптимизация работы с несколькими потоками.*

*Аппаратное ускорение: Использование специализированных инструкций и аппаратных ресурсов для улучшения производительности ВМ.*

А3-1 Регистровая архитектура – преимущества и недостатки. Почему в регистровой архитектуре между АЛУ и массивом регистров должно быть минимум три шины?

***Преимущества регистровой архитектуры:***

*Быстрая работа: Регистры обеспечивают быстрый доступ к данным, что улучшает производительность.*

*Эффективное управление данными: Регистры позволяют удобно хранить промежуточные результаты вычислений.*

*Малое количество обращений к памяти: Это снижает нагрузку на память и улучшает скорость работы.*

***Недостатки регистровой архитектуры:***

*Ограниченное количество регистров: Количество доступных регистров ограничено, что может быть недостаточно для некоторых задач.*

*Большой размер команд: Регистровая архитектура может привести к увеличению размера команд и, следовательно, сложности декодирования.*

*Между АЛУ и массивом регистров должно быть минимум три шины (данных, адреса и управления), чтобы обеспечить эффективную передачу данных, адресации регистров и управление операциями в регистровой архитектуре. Это обеспечивает быстрый и эффективный обмен информацией между регистрами и АЛУ, что важно для производительности вычислений.*

В1-1 Почему в отличие от языка С, например, язык ассемблера не стандартизирован?

*Язык ассемблера не стандартизирован в том числе из-за своей сильной зависимости от аппаратной архитектуры. Каждая архитектура имеет свой собственный набор инструкций и соглашений о структуре кода, что делает стандартизацию на уровне языка ассемблера сложной. Язык С, например, абстрагирует аппаратные детали, что позволяет создавать переносимый код. В отличие от этого, ассемблер ближе к аппаратуре, и стандартизирование было бы менее практичным из-за разнообразия архитектур*

В2-1 Какие особенности характерны для CISC архитектуры?

***Особенности CISC (Complex Instruction Set Computer) архитектуры:***

*Большое количество сложных инструкций: CISC-процессоры предоставляют множество сложных инструкций, что позволяет выполнять более сложные операции в одной инструкции.*

*Использование памяти для операций: В CISC-архитектуре инструкции могут выполнять операции с памятью, что упрощает программирование, но может снижать производительность.*

*Вариативные форматы инструкций: Инструкции CISC могут иметь различные форматы и длину, что делает их более гибкими, но усложняет декодирование.*

*Оптимизация для программиста: CISC-архитектуры стараются предоставить более высокоуровневые инструкции, чтобы сделать программирование более удобным.*

*Наличие микрокодирования: CISC-процессоры могут использовать микрокодирование для выполнения сложных инструкций, что может увеличивать сложность процессора.*

*Кратко, CISC-архитектура характеризуется богатым набором сложных инструкций и упрощенным программированием, но может страдать от сниженной производительности из-за сложности декодирования и выполнения инструкций.*

В3-1 Какие основные принципы концепции фон-Неймана вы знаете?

***Основные принципы концепции фон-Неймана:***

*Принцип хранимой программы: Программа хранится в памяти, и процессор может читать и выполнять инструкции из памяти.*

*Принцип однородности: Вся информация, включая данные и инструкции, представляется в виде битовых последовательностей и обрабатывается одним и тем же набором аппаратных средств*

*Принцип последовательной обработки: Инструкции выполняются последовательно, одна за другой, в линейной последовательности.*

*Принцип двоичной системы: Информация представлена в двоичной системе счисления (0 и 1), что обеспечивает стандартизацию и простоту обработки данных.*

*Принцип универсальности: Архитектура обеспечивает универсальность и способность выполнять различные программы, что делает ее универсальной и масштабируемой.*

А1-2 Приведите последовательность операций, составляющих цикл выполнения команды.

***Цикл выполнения команды (инструкции) в компьютере обычно включает в себя следующие операции:***

*Заборка (Fetch): Процессор извлекает инструкцию из памяти по адресу, указанному в счетчике команд (Program Counter).*

*Декодирование (Decode): Процессор анализирует извлеченную инструкцию, определяя её тип и операнды.*

*Выполнение (Execute): Процессор выполняет инструкцию, осуществляя необходимые операции, такие как арифметические вычисления, чтение/запись данных в регистры или память, управление выполнением программы и т. д.*

*Обратная запись (Write Back): Результат выполнения инструкции записывается обратно в регистры или в память, в зависимости от типа инструкции.*

*Эти операции составляют основу цикла выполнения команды в современных компьютерах*.

А2-2 Сформулируйте понятие «программа для ЭВМ».

*Программа для ЭВМ - это набор инструкций и алгоритмов, предназначенных для выполнения на компьютере, с целью решения конкретной задачи или выполнения определенных операций. Программа определяет последовательность действий, которые должны быть выполнены компьютером, чтобы достичь желаемого результата.*

А3-2 Дайте определение понятию «архитектура системы команд».

*Архитектура системы команд - это набор инструкций и форматов, определяющих функциональные возможности и структуру команд, которые может выполнять процессор компьютера. Эта архитектура определяет, какие операции могут быть выполнены процессором, как операнды обрабатываются и как они кодируются в машинном коде.*

В1-2 В чем заключается фон-Неймановский принцип «адресности»?

*Фон-Неймановский принцип "адресности" заключается в том, что в компьютерной системе данные и инструкции хранятся в памяти и имеют свои адреса. Это позволяет процессору обращаться к данным и инструкциям в памяти по их адресам, что обеспечивает универсальность и гибкость в выполнении различных операций и программ.*

В2-2 В чем достоинства и недостатки уровня абстракции «архитектура системы команд»? Зачем он нужен?

***Достоинства уровня абстракции "архитектура системы команд":***

*1. Упрощение программирования: Позволяет программистам писать код, используя более высокоуровневые инструкции, что делает процесс программирования более удобным и понятным.*

*2. Переносимость: Позволяет одной программе быть запущенной на разных аппаратных платформах с совместимой архитектурой.*

*3. Абстрагирование от аппаратуры: Скрывает детали аппаратной реализации, что упрощает разработку программ и делает их более независимыми от конкретной аппаратуры.*

***Недостатки уровня абстракции "архитектура системы команд":***

*1. Потеря производительности: Высокоуровневые инструкции могут быть менее эффективными, чем оптимизированный код, написанный на более низком уровне абстракции.*

*2. Ограничение доступа к аппаратурным ресурсам: Программист может не иметь полного контроля над аппаратурой, что может быть критичным в некоторых приложениях.*

*Уровень абстракции "архитектура системы команд" нужен для упрощения программирования, повышения переносимости и абстрагирования от аппаратуры, что делает разработку и использование компьютерных систем более удобными и гибкими.*

В3-2 Почему разработчики процессоров в массе своей решили отказаться от уровня микропрограммной?

*Разработчики процессоров отказались от уровня микропрограммной архитектуры, потому что это позволило увеличить производительность и снизить сложность процессоров, делая их более эффективными и экономичными в производстве и использовании.*

А1-3 Зачем нужны регистры процессора? Плюсы-минусы – если их много и плюсы-минусы – если их мало.

*Регистры процессора нужны для временного хранения данных и промежуточных результатов во время выполнения программ.*

***Плюсы многих регистров:***

*- Увеличивают производительность, так как уменьшают необходимость доступа к памяти.*

*- Позволяют более эффективно выполнять сложные вычисления и управлять данными.*

***Минусы многих регистров:***

*- Увеличивают сложность процессора и его стоимость.*

*- Могут требовать больше энергии и места на кристалле.*

***Плюсы небольшого числа регистров:***

*- Уменьшают сложность процессора и стоимость его производства.*

*- Могут быть достаточными для многих задач.*

***Минусы небольшого числа регистров:***

*- Могут снижать производительность, так как процессору приходится чаще обращаться к памяти.*

*- Ограничивают возможности для оптимизации выполнения программ.*

А2-3 Перечислите принципы концепции фон-Неймана.

***Принципы концепции фон-Неймана:***

*1. Принцип хранимой программы: Программа хранится в памяти и может быть изменена и выполняется последовательно.*

*2. Принцип однородности: Все данные и команды представляются в виде битовых последовательностей и обрабатываются одними и теми же средствами.*

*3. Принцип адресности: Данные и инструкции в памяти имеют уникальные адреса, по которым можно обращаться к ним.*

*4. Принцип последовательного выполнения: Инструкции выполняются последовательно и в линейной последовательности.*

*Эти принципы формируют основу для организации современных компьютерных систем и архитектур.*

А3-3 Какие АСК сейчас доминируют на рынке аппаратных средств и в каких нишах? По каким причинам?

*На рынке аппаратных средств доминируют следующие архитектуры:*

*1. x86 (и x86-64): Доминирует в ноутбуках, настольных компьютерах и серверах благодаря широкому распространению и совместимости.*

*2. ARM: Доминирует в мобильных устройствах (смартфоны, планшеты) и встроенных системах, таких как IoT-устройства, благодаря эффективности и энергоэффективности.*

*3. GPU (графические процессоры): Доминируют в вычислениях с высокой производительностью, включая гейминг и машинное обучение, благодаря своей параллельной архитектуре.*

*4. RISC-V: Набирает популярность как открытая архитектура с отличной масштабируемостью, используется во встроенных системах и исследовательских проектах.*

*Причины доминирования варьируются в зависимости от архитектуры, но включают в себя факторы, такие как производительность, энергоэффективность, доступность разработчиков и экосистемы, а также специфические требования для разных ниш рынка.*

В1-3 Перечислите плюсы и минусы микропрограммного уровня вычислительных машин.

***Плюсы микропрограммного уровня:***

*1. Гибкость: Микрокод может быть легко изменен для модификации архитектуры процессора без изменения аппаратуры.*

*2. Абстрагирование: Микрокод скрывает детали аппаратной реализации, облегчая разработку программ.*

*3. Удобство отладки: Микрокод может быть более понятным и легко отлаживаемым, чем аппаратная логика.*

***Минусы микропрограммного уровня:***

*1. Производительность: Микрокод может снижать производительность по сравнению с жесткой проводной логикой.*

*2. Дополнительные затраты: Требует дополнительных ресурсов для хранения и выполнения микрокода.*

*3. Сложность: Разработка и поддержка микрокода может быть более сложной и затратной.*

В2-3 Почему в языке ассемблера в отличие от, например, языка С возможно однозначное декодирование машинных кодов?

*В языке ассемблера машинные коды однозначно декодируются, потому что каждая инструкция ассемблера напрямую соответствует одной машинной инструкции процессора. В отличие от этого, в языке C одна строка кода на C может быть скомпилирована в несколько машинных инструкций, что усложняет однозначное декодирование.*

В3-3 На каких принципах базируется RISC-архитектура?

*RISC-архитектура (Reduced Instruction Set Computer) базируется на следующих принципах:*

*1. Упрощенные инструкции: Ограниченный набор простых инструкций, что упрощает декодирование и выполнение.*

*2. Фиксированная длина инструкций: Инструкции имеют фиксированную длину, что упрощает декодирование и сокращает сложность процессора.*

*3. Отсутствие микрокода: Отсутствие микропрограммы делает выполнение инструкций более быстрым и предсказуемым.*

*4. Отдельные регистры для данных и адресов: Отдельные наборы регистров для данных и адресов упрощают управление данными и адресами.*

*5. Параллелизм инструкций: Поддержка одновременного выполнения нескольких инструкций, что увеличивает производительность.*

*RISC-архитектура стремится к упрощению процессора, улучшению производительности и предсказуемости выполнения инструкций.*

А1-4 В чем различия в подходах по преодолению семантического разрыва в ВМ с CISC и RISC-архитектурами АСК?

*Семантический разрыв между архитектурой и исполнением инструкций может быть преодолен следующим образом:*

*1. CISC (Complex Instruction Set Computer): В CISC-архитектуре семантический разрыв уменьшается за счет внутренней микрокодированной архитектуры. Процессор CISC включает в себя микрокод, который выполняет сложные инструкции, переводя их в более простые микрооперации, что упрощает декодирование и выполнение.*

*2. RISC (Reduced Instruction Set Computer): В RISC-архитектуре семантический разрыв уменьшается за счет простых и однозначных инструкций, что делает их более предсказуемыми и управляемыми. Процессор RISC не использует микрокодирование для выполнения инструкций, и каждая инструкция выполняется непосредственно на аппаратуре.*

*Итак, CISC использует микрокодирование для уменьшения семантического разрыва, в то время как RISC стремится к его уменьшению за счет простых инструкций и отсутствия микрокода.*

А2-4 Что такое микропрограмма? Чем она отличается от программы на языке ассемблера?

*Микропрограмма - это последовательность микроопераций или микрокоманд, которая определяет внутренние действия процессора при выполнении машинных инструкций. Она представляет собой низкоуровневое управление внутренними ресурсами процессора.*

*Отличие от программы на языке ассемблера заключается в том, что микропрограмма работает на уровне аппаратуры и управляет внутренними элементами процессора, в то время как программа на языке ассемблера представляет собой набор инструкций, которые выполняются процессором для выполнения конкретных задач. Микропрограмма используется для управления выполнением инструкций и функционированием аппаратуры процессора.*

А3-4 В чем отличие команд в машинных кодах от команд языка ассемблера?

*Отличие команд в машинных кодах от команд языка ассемблера заключается в том, что команды в машинных кодах представлены в виде двоичных чисел и являются непосредственно понятными и исполняемыми процессором, в то время как команды в языке ассемблера представлены в текстовой форме и предназначены для удобства программиста, а затем переводятся в соответствующие машинные коды компилятором или ассемблером.*

В1-4 Перечислите плюсы и минусы большого количества РОН в ЭВМ с RISC-архитектурой АСК.

***Плюсы большого количества РОН (регистров общего назначения) в ЭВМ с RISC-архитектурой:***

*1. Улучшение производительности: Регистры позволяют хранить промежуточные результаты и операнды, уменьшая необходимость доступа к памяти.*

*2. Более эффективная компиляция: Компиляторы могут генерировать более оптимизированный код.*

*3. Параллельное выполнение: Большее количество регистров способствует параллельному выполнению инструкций, что увеличивает производительность.*

***Минусы большого количества РОН:***

*1. Увеличение размера процессора: Большее количество регистров может потребовать больше места на кристалле и могут увеличивать стоимость процессора.*

*2. Сложность управления регистрами: Увеличение числа регистров может сделать сложнее управление ими и управление ресурсами.*

*3. Усложнение архитектуры: Большее количество регистров может усложнить архитектуру и декодирование инструкций.*

В2-4 Перечислите известные вам уровни абстракции ЭВМ. Прокомментируйте их функции.

*Уровни абстракции ЭВМ:*

*1. Аппаратный уровень*

*2. Уровень микропрограммирования: Этот уровень представляет собой низкоуровневое управление аппаратурой, определяющее внутренние операции процессора*

*3. Уровень машинного кода: Этот уровень представляет машинные инструкции в виде двоичных кодов, которые прямо исполняются процессором.*

*4. Уровень ассемблера:*

*5. Уровень операционной системы:*

*6. Уровень высокоуровневых языков программирования*

В3-4 В чем состоит основная дилемма вычислительной техники? Объясните подробно. Каким образом ее решают?

*Основная дилемма вычислительной техники заключается в балансе между производительностью и энергоэффективностью.*

*Производительность требует мощных процессоров и быстрой памяти для выполнения сложных вычислений, но это может привести к высокому энергопотреблению и нагреву компьютера.*

*Энергоэффективность, напротив, стремится к уменьшению энергопотребления и тепловыделения, что важно для мобильных устройств и экологической устойчивости, но может снижать производительность.*

*Для решения этой дилеммы разработчики применяют различные методы, такие как оптимизация аппаратной архитектуры, создание более эффективных процессоров и использование технологий управления энергопотреблением. Также разрабатываются новые материалы и компоненты, которые способствуют более эффективной работе компьютеров.*

А1-5 Благодаря какому их принципов фон-Неймановской концепции стали возможными интерпретация и трансляция? Объясните почему.

*Интерпретация и трансляция стали возможными благодаря принципу хранимой программы в фон-Неймановской концепции. По этому принципу, программа хранится в памяти и может быть интерпретирована или транслирована в машинный код процессором. Это позволяет выполнять различные программы без необходимости изменения аппаратной части компьютера, что делает компьютеры более гибкими и универсальными.*

А2-5 В чем отличие трансляции от интерпретации? В чем принципиальное отличие калькулятора от ЭВМ?

***Отличие между трансляцией и интерпретацией:***

*- Трансляция: Процесс преобразования исходного кода программы (например, на языке высокого уровня) в машинный код или другой низкоуровневый код перед выполнением. Результатом трансляции является независимая от исходного кода исполняемая программа.*

*- Интерпретация: Процесс выполнения программы напрямую из исходного кода без предварительной трансляции. Интерпретатор читает и анализирует исходный код и выполняет соответствующие операции на ходу.*

***Принципиальное отличие калькулятора от ЭВМ:***

*- Калькулятор: Устройство, специализированное для выполнения арифметических и математических операций. Калькуляторы обычно не обладают возможностью программирования и не могут выполнять сложные общепрограммные задачи.*

*- ЭВМ (Электронная вычислительная машина): Универсальное вычислительное устройство, способное выполнять разнообразные операции и программы, включая арифметические вычисления, управление данными и выполнение программ. ЭВМ обладают возможностью программирования и обеспечивают более широкий спектр функциональности.*

А3-5 Почему тракт «ЦП-память» считается узким местом Принстонской архитектуры, в то время как для Гарвардской он не столь критичен?

*Тракт "ЦП-память" считается узким местом Принстонской архитектуры, потому что в этой архитектуре ЦП и память используют общую шину для обмена данными, что может создавать узкое горлышко в пропускной способности, особенно при выполнении сложных операций.*

*В Гарвардской архитектуре ЦП и память имеют раздельные шины для инструкций и данных, что позволяет более эффективно обрабатывать инструкции и данные параллельно, и, следовательно, тракт "ЦП-память" не так критичен.*

В1-5 Дайте определение понятия «язык ассемблера».

*Язык ассемблера - это низкоуровневый текстовый язык программирования, который предоставляет символьное представление машинных инструкций и операций для конкретной архитектуры процессора.*

В2-5 В чем заключается принцип двоичного кодирования концепции фон Неймана?

*Принцип двоичного кодирования в концепции фон Неймана заключается в том, что все данные и инструкции в компьютере представлены в виде двоичных чисел (нулей и единиц), что позволяет компьютеру легко обрабатывать и хранить информацию в цифровой форме. Это делает выполнение различных операций и алгоритмов более единообразными и универсальными.*

В3-5 Какие особенности аккумуляторной архитектуры можно считать ее достоинствами, а какие – недостатками?

***Достоинства аккумуляторной архитектуры:***

*1. Простота: Она проще в реализации и понимании, особенно для начинающих программистов.*

*2. Эффективность использования памяти: Требуется меньше регистров, что экономит ресурсы.*

***Недостатки аккумуляторной архитектуры****:*

*1. Ограниченность: Могут возникнуть сложности при реализации некоторых операций, требующих временного хранения нескольких промежуточных результатов.*

*2. Усложнение оптимизации: Может быть сложнее создавать эффективный код из-за ограничений в доступных регистрах.*