<u>Вариант А1-1:</u> В чём преимущества и недостатки аккумуляторной архитектуры?

Достоинства:

- Короткие команды;

- Простота декодирования команд.

Недостатки:

- Многократные обращения к памяти.

<u>Вариант А2-1:</u> Какие преимущества стековой архитектуры вы можете перечислить? Какие средства используются для ускорения работы стека в ВМ со стековой архитектурой?

Достоинства:

- стек может быть неоднороден (верхние ячейки быстрые, нижние медленные);

- сокращение адр. части команд;

- компактный код;

- простой компилятор;

- простое декодирование команд.

Для стековой архитектуры лучше (ускорения) всего подходит обратная польская нотация.

<u>Вариант А3-1:</u> Регистровая архитектура – достоинства, недостатки. Почему в регистровой архитектуре между АЛУ и массивом регистров должно быть минимум три шины?

Достоинства:

- компактность кода;

- высокая скорость.

Недостатки:

- более длинные инструкции (по сравнению с аккумуляторной архитектурой).

Так как АЛУ может выполнять операции над несколькими регистрами одновременно, то для передачи данных между АЛУ и массивом регистров должно быть не менее двух шин. *Одна шина используется для передачи данных из регистра в АЛУ, а другая - для передачи данных из АЛУ в регистр.*

Кроме того, третья шина может использоваться для передачи управляющих сигналов, таких как сигналы чтения и записи, которые контролируют доступ к регистрам.

<u>Вариант В1-1:</u> Почему в отличие от языка С, например, язык ассемблера не стандартизирован?

Язык ассемблера не стандартизирован, потому что он напрямую зависит от аппаратной архитектуры компьютера, для которой он предназначен.

<u>Вариант В2-1:</u> Какие особенности характерны для CISC архитектуры?

1. *Большой набор инструкций*: CISC-процессоры имеют большой набор инструкций, которые могут выполняться непосредственно процессором. Это позволяет программистам написать более эффективный код, который может выполняться быстрее.

2. *Использование памяти*: CISC-процессоры могут выполнять операции, которые требуют доступа к памяти, например, операции с плавающей точкой или операции с символами.

3. *Адресация памяти*: CISC-процессоры могут использовать различные методы адресации памяти, например, адресацию с индексом, адресацию с базовым указателем и адресацию с относительным смещением.

4. *Сложные инструкции*: CISC-процессоры могут выполнять сложные инструкции, которые объединяют несколько операций в одну инструкцию. Это может ускорить выполнение программы.

5. *Микрокод*: CISC-процессоры используют микрокод для выполнения инструкций. Микрокод - это набор инструкций, которые выполняются процессором для выполнения сложных инструкций.

<u>Вариант В3-1:</u> Какие основные принципы концепции фон-Неймана вы знаете?

Принципы Фон-Неймона:

- принцип хранимой в памяти программы;

- двоичное кодирование;

- линейное пространство памяти (адресность);

- однородность памяти (безразличие к целевому назначению данных) Принстон;

- программное управление (последовательное выполнение команд);

- принцип хранимой в памяти программы.

<u>Вариант А1-2:</u> Приведите последовательность операций, составляющих цикл выполнения команды.

<u>Вариант А2-2:</u> Сформулируйте понятие «программа для ЭВМ»?

Программа для ЭВМ – упорядоченная последовательность команд, подлежащая обработке.

<u>Вариант А3-2:</u> Дайте определение понятию «архитектура системы команд»

Архитектура системы команд (ISA) – интерфейс между всем выполняемым на машине ПО и аппаратным обеспечением, стандартизирует команды, шаблоны битов машинного языка и т.д.

<u>Вариант В1-2:</u> В чём заключается фон-Неймановский принцип «адресности»?

Принцип адресности Фон-Неймана:

- Память – набор пронумерованных ячеек
- В каждый момент доступна любая ячейка
- Двоичные коды данных и команд хранятся в единицах информации – словах
- Доступ к данным по номерам ячеек – адресам

<u>Вариант В2-2:</u> В чем достоинства и недостатки уровня абстракции «архитектура системы команд»? Зачем он нужен?

Архитектура системы команд (ISA) – интерфейс между всем выполняемым на машине ПО и аппаратным обеспечением, стандартизирует команды, шаблоны битов машинного языка и т.д.

Преимущества: позволяет реализовывать различные аппаратные варианты одной программной архитектуры.

Недостатки: периодически препятствует внедрению новых технологий и прочих инноваций.

<u>Вариант В3-2:</u> Почему разработчики процессоров в массе своей решили отказаться от уровня микропрограммной?

В 60-х-70-х годах количество микропрограмм сильно увеличилось. Однако они работали все медленнее и медленнее, поскольку требовали большого объема памяти (до 60-70% от всей памяти).

В конце концов исследователи осознали, что с устранением микропрограммы резко сократится количество команд и компьютеры станут работать быстрее.

<u>Вариант А1-3:</u> Зачем нужны регистры процессора? Плюсы-минусы – если их много, и плюсы-минусы – если их мало?

Регистр процессора – поле заданной длины во внутрипроцессорной сверхбыстрой оперативной памяти (СОЗУ). Они используются для хранения данных и выполнения операций непосредственно в самом процессоре. Они позволяют сохранять промежуточные результаты вычислений, обрабатывать данные и выполнять операции быстро и эффективно. Регистры также используются для передачи данных между различными частями процессора и для взаимодействия с памятью.

Меньшее количество регистров => меньше разрядов на кодирование номера регистра => короче команда.

<u>Вариант А2-3:</u> Перечислите основные принципы концепции фон-Неймана?

Принципы фон-Неймана (1945)

- принцип хранимой в памяти программы
- двоичное кодирование
- программное управление (последовательное выполнение команд)
- однородность памяти (безразличие к целевому назначению данных)
- линейное пространство памяти (принцип адресности)

<u>Вариант А3-3:</u> Какие АСК (Архитектуры систем команд) сейчас доминируют на рынке аппаратных средств и в каких нишах? По каким причинам?

Последние годы – повышение интереса к стековой архитектуре. Удобен для Java и Forth. (компактный код, простой компилятор и декодирование команд)

Регистровая АСК – преобладающая в настоящее время (все современные персоналки). Достоинства – компактность кода, высокая скорость.

<u>Вариант В1-3:</u> Перечислите плюсы и минусы микропрограммного уровня вычислительных машин.

Плюсы:

- сокращало количество ламп и увеличивало надежность ВМ.
- простота добавления новых команд.
- способность приостановить одну программу и начать другую, используя небольшое число команд (переключение процесса).
- доп. команды для ускорения работы ВМ.

Минусы:

- ёмкость микропрограммной памяти занимает до 60%.
- из-за большого количества команд, работа компьютера замедлилась.

<u>Вариант В2-3:</u> Почему в языке ассемблера в отличие от, например, языка С, возможно однозначное декодирование машинных кодов?

Это связано с тем, что язык ассемблера представляет собой низкоуровневый язык программирования, который напрямую соответствует инструкциям процессора. Каждая инструкция имеет определенный машинный код, который задается определенной последовательностью битов. При компиляции программы на языке ассемблера, эти инструкции преобразуются в соответствующие машинные коды, которые можно однозначно декодировать и выполнить на процессоре. В языке С же компилятор выполняет более сложные преобразования и оптимизации кода, что может привести к тому, что одному и тому же фрагменту кода на языке С могут соответствовать разные машинные коды.

<u>Вариант В3-3:</u> На каких принципах базируется RISC – архитектура?

RISC-архитектура базируется на принципах простоты и эффективности процессорной архитектуры. Она предполагает использование набора простых инструкций, каждая из которых выполняется за один такт процессора. Кроме

того, в RISC-архитектуре используется большое количество регистров, что позволяет уменьшить количество обращений к памяти и повысить скорость работы процессора. Также в RISC-архитектуре предполагается использование конвейерной обработки команд, что позволяет увеличить производительность процессора за счет параллельного выполнения нескольких команд.

<u>Вариант А1-4:</u> В чём различия в подходах по преодолению семантического разрыва, в ВМ с CISC- и RISC- архитектурами АСК?

В ВМ с CISC-архитектурой семантический разрыв преодолевается путем использования сложных инструкций, которые могут выполнять несколько операций одновременно. Это позволяет уменьшить количество инструкций, необходимых для выполнения определенной задачи, и повысить производительность процессора. Однако, такой подход может привести к увеличению сложности процессора и затруднить его разработку.
В ВМ с RISC-архитектурой семантический разрыв преодолевается путем использования простых инструкций, каждая из которых выполняется за один такт процессора. Это позволяет упростить процессорную архитектуру и ускорить выполнение команд. Кроме того, в RISC-архитектуре используется большое количество регистров, что позволяет уменьшить количество обращений к памяти и повысить скорость работы процессора. Однако, такой подход может привести к увеличению количества инструкций, необходимых для выполнения определенной задачи, и усложнить программирование.

<u>Вариант А2-4:</u> Что такое микропрограмма? Чем она отличается от программы на языке ассемблера?

Микропрограмма – встроенный неизменяемый интерпретатор, функция которого заключается в выполнении программ посредством интерпретации.
Отличие микропрограммы от программы на языке ассемблера заключается в том, что микропрограмма используется процессором для выполнения инструкций архитектуры, в то время как программа на языке ассемблера используется для написания прикладных программ. Кроме того, микропрограмма является низкоуровневой и специфичной для конкретной архитектуры процессора, в то время как программа на языке ассемблера может быть написана на различных архитектурах процессоров.

<u>Вариант А3-4:</u> В чём отличие команд в машинных кодах от команд языка ассемблера?

Команды в машинных кодах и команды в языке ассемблера выполняют одни и те же операции, но различаются в формате представления. Команды в машинных кодах представлены битовыми последовательностями, которые понимает процессор. Команды в языке ассемблера представлены мнемониками, которые понятны человеку и затем переводятся в соответствующие битовые последовательности машинного кода. Команды в языке ассемблера обеспечивают более удобный и понятный способ написания программ, чем написание программ на машинном коде.

<u>Вариант В1-4:</u> Перечислите плюсы и минусы большого количества РОН в ЭВМ с RISC АСК.

Плюсы:
1. Большое количество РОН позволяет обрабатывать большие объемы данных и выполнять сложные вычисления.
2. RISC архитектура обеспечивает быстродействие и эффективность работы процессора, так как использует простые и быстрые команды.
3. АСК обеспечивает более эффективное использование ресурсов компьютера, так как позволяет выполнять несколько операций одновременно.
Минусы:
1. Большое количество РОН требует большого объема оперативной памяти и энергопотребления.
2. RISC архитектура может быть не совместима с некоторыми программами, написанными для других архитектур.
3. АСК может быть сложной для программистов, так как требует более тщательного управления ресурсами и оптимизации кода.

<u>Вариант В2-4:</u> Перечислите известные вам уровни абстракции ЭВМ. Прокомментируйте их функции.

1. Уровень аппаратуры - это низший уровень абстракции, который описывает физические компоненты компьютера и их взаимодействие. На этом уровне происходит выполнение операций в машинном коде.

2. Уровень микропрограммирования - это уровень, на котором происходит управление работой процессора, его регистрами и оперативной памятью. На этом уровне используются микрокоды, которые представляют собой набор инструкций, управляющих работой процессора.

3. Уровень ассемблера - это уровень, на котором программа описывается на языке ассемблера, который представляет собой более понятный для человека язык, чем машинный код. На этом уровне происходит трансляция программы из языка ассемблера в машинный код.

4. Уровень языка высокого уровня - это уровень, на котором программа описывается на языке программирования, который предоставляет более высокий уровень абстракции, чем язык ассемблера. На этом уровне программист может использовать более сложные конструкции и алгоритмы, чем на уровне ассемблера.

5. Уровень системного программного обеспечения - это уровень, на котором создаются операционные системы, драйвера устройств и другие системные приложения. На этом уровне используются различные абстракции, такие как файловые системы, сетевые протоколы и т.д.

6. Уровень прикладного программного обеспечения - это уровень, на котором создаются приложения для конечных пользователей, такие как текстовые редакторы, браузеры, игры и т.д. На этом уровне используются различные библиотеки и фреймворки для упрощения разработки приложений.

Вариант В3-4: В чем состоит основная дилемма вычислительной техники? Каким образом её решают?

Дилемма: алгоритм решения задачи для человека ≠ алгоритму решения задачи для ВМ.

Решение: построение ряда уровней абстракций, каждая из которых надстраивается над абстракцией более низкого уровня.

Вариант А1-5: Благодаря какому из принципов фон-Неймановской концепции стали возможными интерпретация и трансляция? Объясните – почему?

Интерпретация и трансляция стали возможными благодаря принципу однородности памяти. Это объясняется тем, что команды одной программы могут быть получены как результат работы другой программы.

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

Основное отличие в том, что при трансляции программа переводится из более высокоуровневого языка L1 в более низкоуровневый L0, хранится в памяти ВМ и затем выполняется. При интерпретации же каждая команда программы на L1 перекодируется в L0 и сразу же выполняется.

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

В Принстонской архитектуре тракт «ЦП-память» является узким местом, из-за его пропускной способности, поскольку процессор и память конкурируют за доступ к одной шине, что означает, что ЦП и память не могут одновременно работать на полную мощность. В Гарвардской архитектуре процессор и память имеют разные шины для обмена данными, что позволяет им работать на полную мощность одновременно.

Вариант В1-5: Дайте определение понятия «язык ассемблера».

Язык ассемблера – это низкоуровневый язык программирования, который транслируется непосредственно в инструкции машинного языка. Он представляет собой формат записи машинных команд, удобный для восприятия человеком.

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

Принцип двоичного кодирования заключается в представлении всей информации (команды, данные) в бинарном виде. При этом каждый тип информации имеет свой формат. Таким образом образуется последовательность битов, связанных одним смыслом, называемым полем.

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

Достоинства аккумуляторной архитектуры:

- Короткие команды
- Простота декодирования команд

Недостатки аккумуляторной архитектуры:

- Многократные обращения к памяти