**Вопросы к экзамену по «Архитектуре аппаратных средств»**

(2 СПО)

1. Понятие ЭВМ. Классификации ЭВМ, характеристики ЭВМ.

* ЭВМ— устройство или совокупность устройств, предназначенных для автоматического или механического процесса обработки информации.
* По типу решения задач:

Специальные – ориентированы на решение разного\одного рода задач

Общ. Назначения – служат для решения широкого класса задач, отличаются сложным алгоритмом и большим объемом обработки

Проблемно – ориентированные – для решения сложных задач, связанных как правила с управ. тех. объектами

* Относительно пользователь:

Основные показатели

Адекватное решение задач

Простой и удобный интерфейс

Доступность

* По условиям эксплуатации

Офисные – предназначены для решения широкого класса задач при нормальных условиях

Спец. ПК – служат для более узкого класса задач или одной задачи требующих многократного решения, и функционируют в определенных условиях

* Производительность и характер использования:

Микро-ПК – это ПК, в которых место процессора используется микро-ПЦ, нужны для решения разного рода задач

ПК — это микро-ПК универсального назначения рассчитанный на одного пользователя

* быстродействие, которое характеризуется числом команд, выполняемых ЭВМ за одну секунду.
* является емкость запо­минающих устройств. Этот показатель позволяет определить, какой набор программ и данных может быть одновременно размещен в памяти.
* Надежность - это способность ЭВМ при определенных условиях выпол­нять требуемые функции в течение заданного периода времени. Например, у современных HDD среднее время наработки на отказ достигает 500 тыс.ч. (около 60 лет).
* Точность - возможность различать почти равные значения. Точность получения результатов обработки в основном оп­ределяется разрядностью ЭВМ, а также используемыми структурными еди­ницами представления информации (байтом, словом, двойным словом). С помощью средств программирования языков высокого уровня этот диапазон может быть увеличен в несколько раз, что позволяет достигать очень высокой точности.
* Достоверность - свойство информации быть правильно воспринятой. Достоверность характеризуется вероятностью получения безошибочных ре­зультатов. Заданный уровень достоверности обеспечивается аппаратурно-программными средствами контроля самой ЭВМ. Возможны методы контроля достоверности путем решения эталонных задач и повторных расчетов. В особо ответственных случаях проводятся контрольные решения на других ЭВМ и сравнение результатов.

1. Этапы развития ЭВМ. Механический этап.

* Начался в середине 17в, длился до конца 19в.

Наиболее значимые результаты того времени это:

1623г. Шиккард – создаёт и описывает счётную машину, которая может делать 4 операции

1642 Паскаль - построил восьмиразрядную действующую модель счетной суммирующей машины. Впоследствии была создана серия из 50 таких машин, одна из которых являлась десятиразрядной.

1673 Лейбниц – создает арифмометр

1881 – серийное производство арифмометров

Чарльз Бэббидж выдвинул идею о создание программно-управляемой счетной машины, имеющей арифметическое устройство, устройство управления, ввода и печати.

1. Этапы развития ЭВМ. Электромеханический этап.

* Данный этап был самым коротким за все время, он длился всего 60 лет

1887 Холлерит – создал счётное – аналитический комплекс, который состоял из перфоратора, сортировочной машины и табулятора

1930 Буш - разрабатывает дифференциальный анализатор, использованный в дальнейшем в военных целях.

1937 Атанасов, Берри - создают электронную машину ABC.

1944 Айкен - разрабатывает и создает управляемую вычислительную машину MARK-1.

1957 - в СССР создана РВМ-I, которая эксплуатировалась до 1965 г.

1. Этапы развития ЭВМ. Электронный этап.

* Начало электронного этапа развития вычислительной техники связано с созданием электронной вычислительной машины ENIAC в США в 1945 году.

Выделяют несколько поколений ЭВМ, каждое из которых отличается элементной базой, архитектурой и программным обеспечением, а также быстродействием, объемом памяти и способами ввода и вывода данных.

ЭВМ пятого поколения должны обеспечивать простоту использования, создавать программы по спецификациям на естественном языке и улучшать основные характеристики компьютеров

1. Понятие о многоуровневой архитектуре компьютера. Цифровой логический уровень (ЦЛУ). Понятие «вентиль». Вентили на физическом уровне. Обозначения вентилей на ЦЛУ.

* Такая архитектура состоит из различных уровней, каждый из которых соответствует отдельной функции. Таким образом, вносить изменения в каждый отдельный уровень проще, чем заниматься всей архитектурой. Разработчики могут создавать гибкие и повторно используемые приложения. Это значительно упрощает весь процесс управления системой.
* Цлу по-другому можно назвать аппаратным обеспечением Цифровые схемы этого уровня конструируются из небольшого числа простых элементов путем их сочетания в различных комбинациях.
* Вентиль – это устройство, которое выдает результат булевой операции от введенных в него данных (сигналов).
* Вентили на физическом уровне??
* ??

1. Виды микросхем компьютера (интегральные, комбинационные). Декодер. Кодер.

* Интеграл — это модуль, в котором собраны вентили, вентили размещаются на кремневой пластинке, которая помещается в пластиковый\кремневый корпус, вдоль длинных сторон корпуса располагаются два параллельных ряда выводов которые можно втыкать разъёмы или впаивать в печатную плату. Каждый вывод соед. С выходом\входом вентиля, или с источником питани.

Комби схема – схема с несколькими входами и выходами, в которой выходные сигналы определяются текущими входными сигналами

* Декодер - схема, которая получает на входе n-битное число и использует его для того, чтобы выбрать дугу из 2n выходных лимитов
* Кодер - Лог. Схема, функции которых схожи с декодерами, двоичные интераторы кодируют денные с 2 nвходных линий в n-разрядный код.

1. Мультиплексор. Демультиплексор.

* Схема с 2 n входами и n линиями управления, на которую одну из входных линий и одно соединение с выходом
* Схема, которая имеет n входов и 2 n выходов, на все выходы подается 0, но кроме n выхода на которую подается значения входа

1. Компаратор. Виды микросхем компьютера (арифметические). Схема сдвига.

* Компаратор - устройство, сравнивающее величины аналоговых сигналов. Для аналоговых сигналов компаратор является логическим элементом с релейным режимом
* Арифметическая микросхема — это комбинированные схемы использованные для выполнения арифметических операций
* Схема сдвига – входные данные представляют собой входные данные сдвинутые на один бит, сдвиг бывает арифметический и логический, как правило используется для деления или умножения целых чисел

1. Полусумматор. Полный сумматор. Арифметико-логическое устройство.

* Полусумматор – складывает два одноразрядных целых числа, складываемые числа подаются на выход «сумма»
* Полный сумматор – состоит из 2 полусумматоров, содержит входную линию «Вход переноса» на которую может поступить бит переноса из предыдущего разряда
* Оно выполняет некоторый набор арифметических и логических операций над входными словами (операндами) фиксированной разрядности, выдавая результат в виде выходного слова той же разрядности.

1. Понятие архитектуры ЭВМ. Классическая архитектура ЭВМ. Принципы фон Неймана 1 – 3.

* Архи.эвм –совокупность общих принципов организации аппаратно-программных средств и их характеристик, определяющая функциональные возможности ЭВМ при решении соответствующих классов задач.
* Принцип двоичности.

Для представления данных и команд используется двоичная система счисления.

* Принцип программного управления.

Программа состоит из набора команд, которые выполняются процессором друг за другом в определённой последовательности.

* Принцип однородности памяти.

Как программы (команды), так и данные хранятся в одной и той же памяти. Над командами можно выполнять такие же действия, как и над данными.

1. Классическая архитектура ЭВМ. Принципы фон Неймана 4 – 5. Функциональная структура ЭВМ.

* Принцип адресуемости памяти.

Структурно основная память состоит из пронумерованных ячеек; процессору в произвольный момент времени доступна любая ячейка.

* Принцип последовательного программного управления.

Все команды располагаются в памяти и выполняются последовательно, одна после завершения другой.

* Принцип условного перехода.

Команды из программы не всегда выполняются одна за другой. Возможно присутствие в программе команд условного перехода, которые изменяют последовательность выполнения команд в зависимости от значений данных.

* Функциональная структура ЭВМ — это описание основных компонентов и их связей, которые позволяют компьютеру выполнять задачи обработки информации. Она включает в себя процессор, память, ввод-вывод, шину связи и контроллеры, которые взаимодействуют друг с другом для обеспечения работы компьютера.

1. Понятие «память ЭВМ». Понятие «запоминающее устройство». основные операции в памяти. Характеристики памяти ЭВМ.

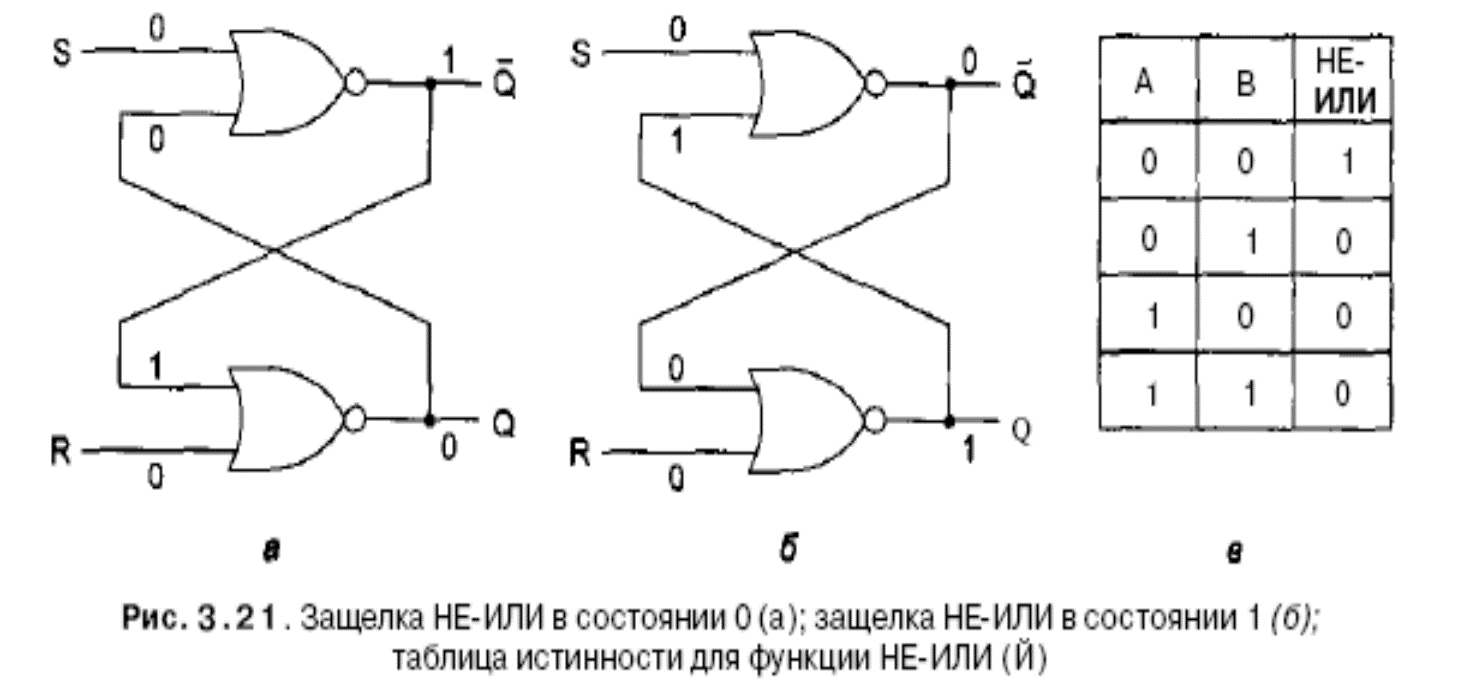
* Память ЭВМ – совокупность тех. Устройств и процессов, обеспечивающих запись, хранение и воспроизведение информаций в ЭВМ
* ЗУ – блок ВМ или самостоятельное устройство, предназначенное для записи, хранения и воспроизведения информации
* Основные операции с памятью:

Запись(занесение) инф. В память и считывание(выборка)инф. Из памяти

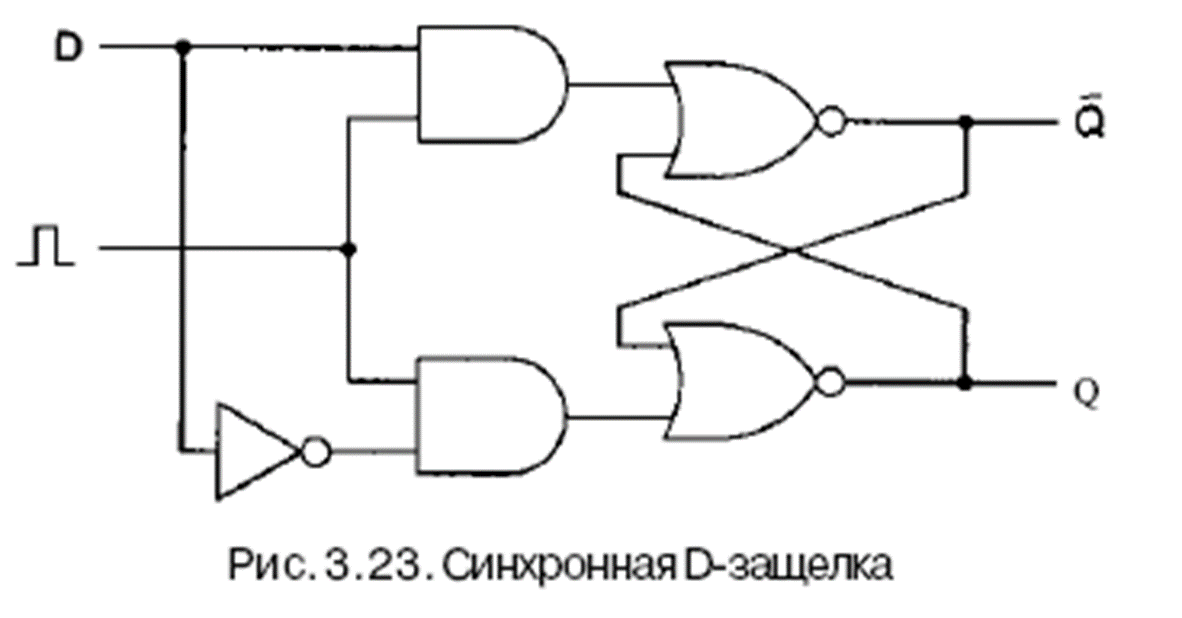
1. Понятие защелки и триггера. Асинхронная RS-защелка (схема, таблица переходов, принцип работы).

* Защелка – схема с двумя устойчивыми состояниями, где переход из одного состояния в другое происходит скачкообразно под воздействием уровня управляющегося сигнала, при этом изменяется уровень напряжения на выходе защелки

Триггер – схема с двумя устойчивыми состояниями, где переход из одного состояния в другое происходит скачкообразно под воздействиями фронта управляющего сигнала, также резко меняется уровень напряжения на выходе триггера

* 
* У асинхронной RS-защелки два входа s – установка и r – сброс, еще есть два дополнительных входа: Q и (не)Q, выходные сигналы защелки, как и триггера, не определяются текущими входными сигналами

1. Синхронная RS-защелка (схема, принцип работы). Синхронная D-защелка (схема, принцип работы).

* работает также что и асинхронная защелка, только при условии того, что на вход подается 1, если же на вход попадается 0, то защелка находится в режиме хранения, присутствует проблема запрещенного состояния, когда нельзя на оба входа подавать 1
* 
* Тоже что и асинхронная защелка только на вход подается 1, если же подается 0 то защелка остается в режиме хранения, отсутствует проблема запрещенного состояния

1. Триггеры. Импульсная схема (схема и график получения сигнала). D-триггер. Обозначения D-защелок и D-триггеров.
2. Понятие регистра. Память 4 на 3 (принцип работы).

* Регистр – это совокупность триггеров, связанных друг с другом определённым образом общей системой управления, предназначенная для хранения данных.
* Память4\*3 содержит 4 3-разразрядных регистра, где регистр состоит из 3 D-триггеров, структура памяти регулярная и на каждом ее регистре можно совершить операцию.

1. Буферные элементы. Микросхемы памяти (2 способа организации).

* Буф.елементы работают как переключатели и представляют собой устройство с 3 состояниями, помимо этого они могут усиливать сигнал поэтому они могут справляться с большим количеством сигналов
* Микросхемы сущ двух видов, первый

1. Классификация памяти ЭВМ по функциональному назначению. Оперативная память и ее виды.

* Оперативная память (ОП) — совокупность ОЗУ, объединенных в одну систему, управляемую процессором.

Существуют 4 типа оп: DDR, DDR2, DDR3, DDR4. Они также делятся на 2 форм фактора: DIMM – для компьютеров, SO-DIMM – для ноутбуков. Эти два типа абсолютно разные, их невозможно спутать, для компьютеров они вытянутые, для ноутбуков – короткие.

1. Классификация памяти ЭВМ по функциональному назначению. ПЗУ и ее виды. ВЗУ (понятие, виды).
2. Масочные – для изготовления нужен трафарет с опр. Набором битов, которые накладывается на фоточувствительный материал, а затем открытые или закрытые части поверхности вытравливаются
3. Однократно программируемые – в отличии от обычных пзу их можно программировать в условиях эксплуатации что позволяет сокротить время выполнения заказа
4. Стираемые программируемые пзу
   * Пзу с уф стиранием
   * Пзу с электрическим стиранием
5. Флэш память – стирается и записывается блоками

* ВЗУ – устройства которые используются для хранения и сохранения данных в сис.пк они предназначены для долгосрочного хранения инф.и обеспечивают доступ к данным вне пк. Виды внешней памяти: магнитная (дискеты, жесткие диски hdd)\оптическая(cd\DVD)\электронная(sdd\flash\ката памяти)

1. Структурная схема микропроцессора intel 80x86. Исполнительный блок. Устройство сопряжения с системной магистралью.
2. Структурная схема микропроцессора (МП) intel 80x86. Взаимодействие элементов при работе микропроцессора. Алгоритм работы МП.
3. Тракт данных. Цоколевка типичного МП. Характеристика выводов.
4. Классификация МП.
5. Определение, характеристики шин. Виды шин.

* Шина – это группа проводников, соединяющих различные устройства и обеспечивающая передачу данных в двоичном коде.
* 1. По выполняемым функциям:

Внутренние – служат для передачи данных от ОП из/в АЛУ.

Внешние – связывают процессор с внешней памятью и устройствами ввода-вывода.

Протокол шины – это четкие правила о том, как должна работать шина. И все устройства, связанные с ней, должны подчиняться им.

Протоколы и технические стандарты были введены, чтобы платы, производимые третьими лицами, подходили к системным шинам. Также существуют технические требования к шинам, чтобы платы от третьих лиц подходили к каркасу печатных плат и имели разъемы, соответствующие материнской плате по мощности, по синхронизации и т.д.

По синхронизации:

Синхронная шина содержит линию, которая запускается кварцевым генератором. Сигнал на этой линии представляет собой меандр с определенной частотой. Любое действие шины занимает целое число так называемых циклов шины.

Асинхронная шина не содержит задающего генератора. Циклы шины могут быть любой требуемой длины и необязательно одинаковы по отношению ко всем парам устройств.

Асинхронная шина не содержит задающего генератора. Циклы шины могут быть любой требуемой длины и необязательно одинаковы по отношению ко всем парам устройств.

1. Виды устройств, работающих с шиной. Вспомогательные устройства шины. Ширина шины (понятие).

* Устп. Работающие с шиной: активные – могут инициировать передачу инф. По шине; пассивные – ждут запроса

1. Проблемы выбора ширины шины.

* Проблемы:

Для памяти большой емкости необходимо много адресных линий. Для широких шин требуется больше проводов, чем для узких. Они занимают больше физического пространства, и для них нужны разъемы большего размера. Все эти факторы делают шину дорогостоящей. Следовательно, необходим компромисс между максимальным размером памяти и стоимостью системы.

1. Арбитраж шины. Централизованный и децентрализованный арбитраж.

* Арбитраж шины это -механизм регулирующий порядок обращений при одновременном обращение к шине
* Централизованный арбитраж - простейший механизм, при котором есть только 1 арбитр шины, чаще всего он вшит в микросхему проц. Она имеет одну линию запроса и может запускать одно или несколько устройств
* Децентрализованный – отсутствует специальная микросхема арбитра, в этом случае пк может содержать некоторое кол-во приоритетных линий запроса шины, когда устройству нужна шина, он запускает свою линию запроса

1. Механизм прерываний.

* **Прерывание** – это способность процессора прерывать свою работу.
* Прерывание происходит при возникновении внешнего события, которое обрабатывается МП. Часто эти события генерируются внешними устройствами. Для сигнала прерывания нужна шина.

1. Виды прерываний.

* Существует два вида прерывания:

1. По типу объекта, генерирующего прерывание:
2. **Аппаратные** – вырабатывающие устройствами пк, когда возникает необходимость их обслуживать, например по прерыванию таймера соответствующий обработчик прерывания увеличивает, содержимое ячеек памяти, используемых для хранения времени. Аппаратные прерывания всегда происходят асинхронно по отношению к выполняющимся программ.
3. **Логические –** генерируются самим процессором. Это прерывания с номерами с 0 по 3. Прерывание с номером 0 генерируется при ошибке деления на ноль, с номером 1 используется при управлении работой процессора и пошаговом выполнении программ, с номером 2 используется при отладке программ, с номером 3 генерируется при возникновении переполнений.
4. **Программные** – вызываются запланировано самой прикладной программой, когда она должна вызвать подпрограмму с помощью прерывания.
5. По зависимости от состояния флага IF
6. **Маскируемые.** Их обработка зависит от флага разрешения прерывания IF. Если IF=1, то прерывание обрабатывается и осуществляется переход на выполнение соответствующей подпрограммы. Если IF=0, то прерывание обрабатываться не будет. Обычно программы всегда разрешают прерывания, но в критических случаях можно их запретить, тогда никто не сможет прервать данную программу.
7. **Немаскируемое.** Не зависит от состояния флага IF. Его невозможно запретить.
8. Ассемблер, как машинно-ориентированный язык. Основные понятия. Редактор связей (Компоновщик).
9. Связывающий загрузчик и его функции. Кросс-системы.
10. Язык Ассемблера (в презентации слайды 10-12).

* **Язык Ассемблера** — система записи программы с детализацией до отдельной машинной команды, позволяющая использовать мнемоническое обозначение команд и символическое задание адресов.

Поскольку в разных аппаратных архитектурах разные программно-доступные компоненты (система команд, регистры, способы адресации), **язык Ассемблера аппаратно-зависимый**. Программы, написанные на языке Ассемблера, могут быть перенесены только на вычислительную систему той же архитектуры.

1. Понятие о макропрограммировании.

* **Макропроцессор (препроцессор)** — модуль системного ПО, позволяющий расширить возможности языка Ассемблера за счет предварительной обработки исходного текста программы.
* **Макровызов или макрокоманда или макрос** — оператор программы, который подлежит обработке Макропроцессором (Макропроцессор обрабатывает не все операторы, а только ему адресованные).
* **Макроопределение** — описание того, как должна обрабатываться макрокоманда, макроопределение может находиться в том же исходном модуле, что и макрокоманда или в библиотеке макроопределений.
* **Макрорасширение** — результат выполнения макровызова, представляющий собой один или несколько операторов языка Ассемблера, подставляемых в исходный модуль вместо оператора макровызова.