В О П Р О С Ы
к экзамену по курсу "Организация ЭВМ и систем"
для гр. Р3300, Р3301, Р3302, Р3304с, Р3310, Р3311, Р3317, Р3318, Р3375

1. Понятие вычислительной машины (компьютера).

   Компьютер – это прибор, способный производить вычисления и принимать решения в миллионы или даже в миллиарды раз быстрее человека. Компьютеры обрабатывают данные под управлением наборов команд, называемых компьютерными программами.

   Цифровой компьютер – это машина, которая может решать задачи, выполняя данные ей команды. Последовательность команд, описывающих решение определенной задачи, называется программой.

   ЭВМ – комплекс электронного оборудования, выполняющий интерпретацию программ в виде физических процессов, назначением которых является реализация математических операций над информацией, представляемой в цифровой форме.

   ЭВМ – искусственная (инженерная) система, предназначенная для выполнения вычислений на основе алгоритмов. Принципы построения ЭВМ определяются с одной стороны назначением ЭВМ и с другой – элементной базой (набором элементов, которые используются для создания ЭВМ). Основным назначением ЭВМ является выполнение вычислений на основе алгоритмов, и поэтому свойства алгоритмов предопределяют принципы построения ЭВМ или, точнее, ее архитектуру (организацию).

2. Вычислительный комплекс, вычислительная система.

ВЫЧИСЛИТЕЛЬНЫЙ КОМПЛЕКС - взаимосвязанная совокупность средств вычислительной техники, в которую входит не менее 2 процессоров, объединенных системой управления и имеющих общую память, единое математическое обеспечение и общие периферийные устройства.

3. Понятие архитектуры и организации ЭВМ. Виды архитектур.

   Под архитектурой ЭВМ обычно понимается ее представление и описание возможностей с точки зрения пользователя, разрабатывающего программы на машинно-ориентированном языке (ассемблере). Архитектура, как правило, отображает те аспекты структуры и принципов функционирования ЭВМ, которые являются видимыми для пользователя и, следовательно, для разрабатываемых им программ.

Термины архитектура ЭВМ и организация ЭВМ во многом являются подобными, в связи с чем многие специалисты используют их как синонимы. Одним из сторонников такого подхода является Э. Таненбаум: «Архитектура связана с аспектами, которые видны программисту. Например, сведения о том, сколько памяти можно использовать при написании программы, - часть архитектуры. Аспекты разработки (например, какая технология используется при создании памяти) не являются частью архитектуры. Изучение того, как разрабатываются те части компьютерной системы, которые видны программистам, называется изучением компьютерной архитектуры. Термины «компьютерная архитектура» и «компьютерная организация» означают в сущности одно и то же». Однако, существует и другой подход, при котором эти понятия, если не противопоставляются, то, по крайней мере, различаются. Это различие состоит в том, что если понятие архитектура ЭВМ определяет возможности ЭВМ, то понятие организация ЭВМ определяет, как эти возможности реализованы в рамках конкретных моделей ЭВМ.

Два вида архитектур:
- программная архитектура, которая включает в себя аспекты, видимые программистом.
- аппаратная архитектура, которая включает в себя аспекты, невидимые программистом (прозрачные как для программиста, так и для программ).

Программную архитектуру также можно разделить на два уровня: прикладную и системную.

4. Основные элементы архитектуры компьютера.

К основным элементам (аспектам) прикладной архитектуры ЭВМ, как правило, относятся:

1) типы, форматы и способы представления данных, аппаратно поддерживаемые в ЭВМ;
2) регистровая структура процессора;
3) адресная структура основной памяти и принципы размещения информации в ней, принципы формирования физического адреса;
4) режимы адресации;
5) структуры и форматы машинных команд;
6) система команд.

Все аспекты прикладной архитектуры естественным образом входят и в системную архитектуру.

К дополнительным аспектам системной архитектуры, как правило, относятся:

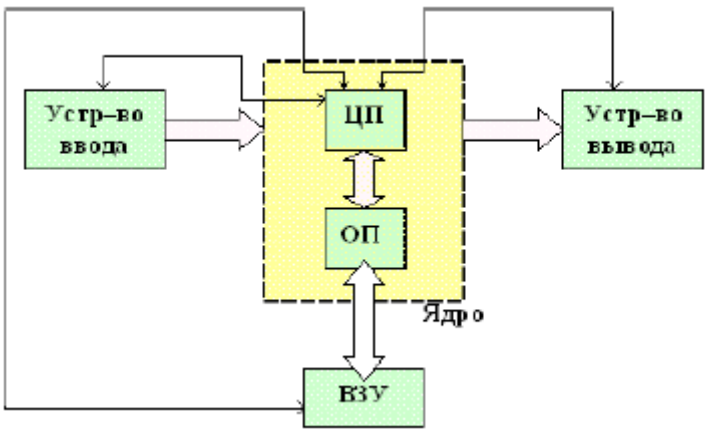
1) организация прерываний;
2) организация ввода/вывода;
3) организация виртуальной памяти (сегментная и страничная), принципы преобразования логического (виртуального) адреса в физический;
4) организация защиты памяти;
5) организация многозадачного (многопрограммного) режима работы ЭВМ, организация переключения задач (программ);
6) поддержка механизмов отладки программ на аппаратном уровне;
7) поддержка механизмов проверки (тестирования) отдельных блоков процессора на аппаратном уровне.

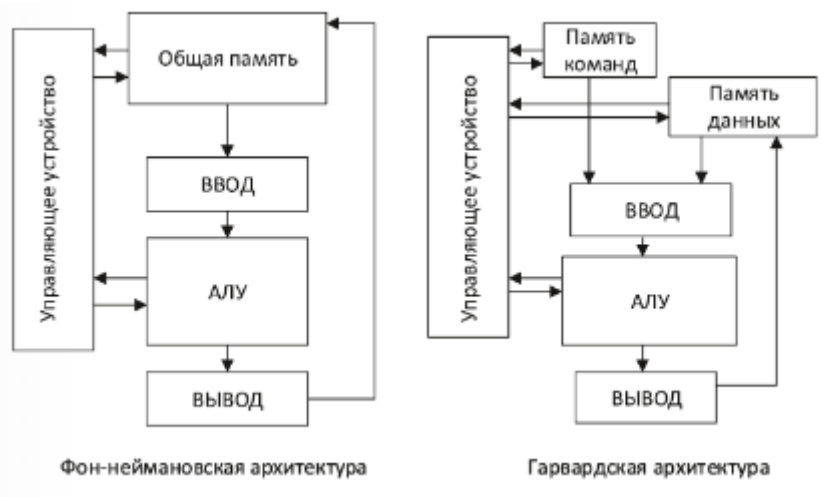
К основным аспектам аппаратной архитектуры, как правило, относятся:
1) структурная организация ЭВМ, включающая в себя номенклатуру устройств, входящих в состав ЭВМ, и организацию связей между устройствами на уровне аппаратных интерфейсов;
2) структурная организация процессора, включающая в себя реализацию конвейера команд и арифметико-логического устройства и принципы построения блока микропрограммного управления;
3) организация кэш□памяти;
4) организация основной памяти на физическом уровне и, в частности, принципы построения многомодульной памяти с расслоением обращений (чередованием адресов);
5) представление аппаратного интерфейса на физическом уровне.

5. Принцип программного управления.

Когда бы вы ни заглянули в память ЭВМ, в её ячейках хранятся наборы нулей и единиц. ЭВМ выполняет без участия человека не только одну команду, но и длинную последовательность команд (программу) . В этом и состоит один из основных принципов работы ЭВМ - принцип программного управления.

6. Каноническая структура компьютера. Принстонская и гарвардская архитектура ЭВМ.





Их основное отличие заключалось в том, что архитектура фон Неймана использовала единую память (общую шину данных), а гарвардская предполагала наличие нескольких шин (в оригинале две: шина данных и шина команд).

Преимущества машины фон Неймана оценили сразу, поскольку в ней содержалось значительно меньше проводников между арифметико-логическим устройством (АЛУ) и областью памяти, и на долгие годы она стала эталоном для создания ВС. Именно фон-неймановская архитектура с подачи Джона Кока являлась прародителем процессоров RISC (Reduced Instruction Set Computer – вычисления с сокращенным набором команд).

Появление процессоров на гарвардской архитектуре мировое сообщество восприняло прохладно, поскольку в начале 70-х годов не было программного обеспечения, способного реализовать его потенциал. Их за глаза называли процессорами «для бедных», поскольку они не могли работать на больших частотах.

Но все изменилось после появления персонального компьютера Apple I, в основе которого был восьмиразрядный процессор MOS 6502 на гарвардской архитектуре с операционной системой Apple DOS.

Простота ОС компенсировалась достаточно сложным процессором, названным впоследствии CISC (Complex Instruction Set Computer – вычисления с комплексным набором команд), с отдельной 16-разрядной адресной шиной ивозможностью произвольного манипулирования регистрами. Монолитная однопользовательская ОС позволила выжать из него небывалую по тем временам производительность при решении отдельных задачах.

7. Достоинства и недостатки неймановской архитектуры ЭВМ.

Архитектура фон Неймана имеет ряд важных достоинств.

Наличие общей памяти позволяет оперативно перераспределять ее объем для хранения отдельных массивов команд, данных и реализации стека в зависимости от решаемых задач. Таким образом, обеспечивается возможность более эффективного использования имеющегося объема оперативной памяти в каждом конкретном случае применения.
Использование общей шины для передачи команд и данных значительно упрощает отладку, тестирование и текущий контроль функционирования системы, повышает ее надежность.
Поэтому Принстонская архитектура в течение долгого времени доминировала в вычислительной технике.

Однако ей присущи и существенные недостатки. Основным из них является необходимость последовательной выборки команд и обрабатываемых данных по общей системной шине. При этом общая шина становится «узким местом» (bottleneck – «бутылочное горло»), которое ограничивает производительность цифровой системы.

8. Понятие ядра компьютера и периферии. Организация их взаимодействия.

   Центральную часть принято называть ядром. В ядро входят два основных устройства компьютера: ЦП и ОП.

   Периферийную часть можно условно представить устройствами трех типов:
   - внешние запоминающие устройства, которые образуют внешнюю память (к ним относятся накопители на магнитных дисках и на магнитных лентах);
   - устройства ввода;
   - устройства вывода.

   Обмен информацией между ядром и периферией, а так же между устройствами ядра осуществляется на уровне аппаратных интерфейсов.
   Организация обмена между ядром и периферийной частью компьютера возлагается на систему ввода/вывода (I/OS – Input/Output System).

9. Система ввода-вывода компьютера. Аппаратные и программные средства ввода-вывода

   Организация обмена между ядром и периферийной частью компьютера возлагается на систему ввода/вывода (I/OS – Input/Output System). Система ввода/вывода представляет собой аппаратно - программный комплекс.

   Аппаратная часть I/O S включает в себя:
   1. собственно периферийные устройства (разделённые на УВ/В и ВЗУ);
   2. контроллеры ПУ(устройства управления);
   3. контроллеры для организации обмена, в частности, контроллер DMA – Direct Memory Accses (ПДП-прямой доступ к памяти) PIC (Program Interrupt Controller – Программируемый Контроллер Прерываний);
   4. интерфейсы (шины);
   5. система прерываний.

   Программная часть I/O S включает в себя:
   1. супервизор (Supervisor) в/в;
   2. драйверы ВУ.

10. Способы адресации внешних устройств.

Различие способов адресации связанно с использованием раздельного или единого адресного пространства для памяти и портов ввода/вывода.

Использование раздельного адресного пространства предполагает, что одни и те же адреса могут применяться как для адресации ячеек памяти, так и для адресации портов ввода/вывода. При этом в системе команд процессора должны использоваться специальные команды для ввода/вывода, отличные от команд обмена с памятью. Достоинством подобного подхода принято считать возможность использования более короткого адреса порта ввода/вывода по сравнению с адресом ячейки памяти.
Так, например, в системе команд процессора Intel для адресации портов ввода/вывода может использоваться либо 1 байт (при прямой адресации порта), либо 2 байта (при косвенной адресации порта). При косвенном задании адреса используется регистр DX (неявно адресуемый), в котором и находится адрес порта ввода/вывода.

Использование единого адресного пространства существенно влияет как на систему команд процессора, так и на управление вводом/выводом на аппаратном уровне.
Некоторая область адресного пространства в старших адресах используется не для адресации памяти, а для адресации портов ввода/вывода. Подобная идея была впервые реализована в мини ЭВМ PDP – 11(DEC). Подобный способ хорошо вписывается в рамки так называемого магистрального интерфейса.

В системе команд отсутствуют специальные команды ввода/вывода, и обмен между памятью и портами ввода/вывода реализуется по аналогии с обменом между процессором и памятью с использованием обычной команды типа MOV.

Организация ввода/вывода с отображением на память обладает следующими достоинствами:

1) Не требуются специальные команды ввода/вывода (упрощение системы команд).
2) Возможность использования любых видов обработки (реализуется и/или логическими командами применительно к содержимому портов ввода/вывода).

Недостатками использование совмещенного адресного пространства являются:
1) Усложнение управления кэшированием, связанное с необходимостью запрета кэширования адресного пространства, выделенного для портов ввода/вывода.

2) Сложность реализации этого способа для многошинной архитектуры.

11. Понятие интерфейса. Виды интерфейсов, используемых в компьютерных системах.

В дальнейшем под интерфейсом будем понимать *аппаратный интерфейс*.
В литературе существует достаточно большое число определений интерфейса. В общем плане под интерфейсом принято понимать *способ сопряжения и взаимодействия между несколькими объектами и субъектами.*
В отношении ЭВМ принято рассматривать множество понятий интерфейсов, например, *аппаратный, программный, пользовательский* и т.д.

В обобщенном плане под *аппаратным интерфейсом* принято понимать совокупность линий и шин электрических схем и алгоритмов, предназначенную для осуществления обмена информацией между устройствами.

1. По способу соединения компонент:
- магистральный;
- радиальный;
- цепочный;
- комбинированный (смешанный).

С помощью *цепочного* интерфейса обычно реализуются линии управления, которые проходят последовательно через ряд подключенных к ним ВУ (в частности, сигналы разрешения). *Сигнал разрешения*, проходя последовательно через подключенные к интерфейсу ВУ, может быть заблокирован первым из ВУ на этой линии, которая предварительно послала запрос прерывания.
Реализация цепочного интерфейса предоставляет преимущество в обслуживании (приоритет) тем устройствам, которые находятся ближе по электрической связи к источнику сигнала разрешения. Этим источником, как правило, является **арбитр** – специализированный блок, входящий в состав ЦП. Как правило, линии цепочного интерфейса объединяются с линиями магистрального интерфейса, образуя комбинированный.

2. По способу передачи информации:
- параллельные;
- последовательные;
- параллельно-последовательные.

3. По принципу обмена:
- синхронный;
- асинхронный.

4. *По режиму передачи информации*:
- односторонний (симплексный);
- двухсторонний (дуплексный);
- двухсторонний - поочередный (полудуплексный).

5. По функциональному назначению:
- системные;
- интерфейсы периферийных устройств (малые интерфейсы);
- интерфейсы ввода/вывода;
- интерфейсы программно-управляемых модульных систем и приборов (приборные).

12. Понятие системного интерфейса. Уровни его представления.

системный интерфейс — Интерфейс, обеспечивающий сопряжение устройств центральной части электронной вычислительной машины.

1) ***логический***: определяет состав, наименование и предназначение линий (шин), а также порядок передачи информации (сигналов) по этим линиям (протокол обмена, который обычно представляется в виде временных диаграмм).
2) ***физический***: определяет параметры сигналов (электрических, оптических), переданных по линиям интерфейсов;
3) ***конструктивный:*** определяет физическую реализацию шин интерфейсов (печатные проводники, витая пара, коаксиальный кабель), а также виды разъемов и распределения линий интерфейсов по контактам разъемов.

13. Состав шин системного интерфейса.

Основные виды линий (шин) входящие в состав интерфейсов:
- шина адреса;
- шина данных;
- шина управления (для передачи сигналов управляющих обменом);
- линии синхронизации (для передачи сигналов, синхронизирующих обмен данных по интерфейсу);
- линии запросов прерывания (для передачи сигналов прерываний от ВУ подключенных к интерфейсу в ЦП или PIC);
- линии разрешения прерывания (для передачи сигналов разрешения от ЦП или PIC к ВУ);
- линии питания;
- линии заземления.

Общее число линий в современных интерфейсах составляет ~150-200.

14. Контроллеры внешних устройств. Параллельная и последовательная передача данных.

Контроллер внешнего устройства – подобие упрощенного процессора, который берет на себя часть работы по управлению внешними устройствами, таким образом, разгружает от работы по текущему обслуживанию внешних устройств - центральный процессор.

В телекоммуникации, последовательная передача — это последовательность передачи элементов сигнала, представляющих символ или другой объект данных. Цифровая последовательная передача — это последовательная отправка битов по одному проводу, частоте или оптическому пути. Так как это требует меньшей обработки сигнала и меньше вероятность ошибки, чем при параллельной передаче, то скорость передачи данных по каждому отдельному пути может быть быстрее. Этот механизм может использоваться на более дальних расстояниях, потому что легко может быть передана контрольная цифра или бит чётности.

Параллельной передачей в телекоммуникациях называется одновременная передача элементов сигнала одного символа или другого объекта данных. В цифровой связи

параллельной передачей называется одновременная передача соответствующих элементов сигнала по двум или большему числу путей. Используя множество электрических проводов можно передавать несколько бит одновременно, что позволяет достичь более высоких скоростей передачи, чем при последовательной передаче. Этот метод применяется внутри компьютера, например, во внутренних шинах данных, а иногда и во внешних устройствах, таких, как принтеры. Основной проблемой при этом является «перекос», потому что провода при параллельной передаче имеют немного разные свойства (не специально), поэтому некоторые биты могут прибыть раньше других, что может повредить сообщение. Бит чётности может способствовать сокращению ошибок. Тем не менее электрический провод при параллельной передаче данных менее надёжен на больших расстояниях, поскольку передача нарушается с гораздо более высокой вероятностью.

15. Структура компьютера с программно-управляемым интерфейсом.

В данной структуре на процессор возлагаются функции обеспечения ввода-вывода и ВУ подключаются через интерфейс ввода-вывода непосредственно к процессору. Команда ввода-вывода содержит код операции (КО) и адрес внешнего устройства (АВУ) и исполняется следующим образом:

1) Код операции КО и адрес АВУ посылается процессором в шину ввода-вывода;

2) Контроллеры ВУ сравнивают свои адреса с АВУ. Контроллер с заданным адресом принимает КО и инициирует работу ВУ;

3) При вводе данных СД предается по шине ввода-вывода в процессор, а при выводе – из процессора в контроллер ВУ, после чего команда ввода-выводасчитается выполненной.

(*СД*- слово данных пересылается специальным устройством ввода-выводамежду контроллером и регистром СД.)

Затем процессор обеспечивает обработку слова данных СД, например передачу слова в оперативную память. Передача сегмента данных, состоящего из последовательности слов, программируется совокупностью команд, в которой после команды ввода-вывода выполняются операции передачи СД в оперативную память (или в обратном направлении) и подсчет количества переданных слов.

Таким образом, ввод-вывод каждого слова данных программируется несколькими операциями процессора, и процессор непосредственно обслуживает процесс ввода-вывода.

Процессор, исполняя команду ввода-вывода, может инициировать операции ввода-вывода и, не дожидаясь ее выполнения, продолжить исполнение программы. В этом случае процессор и ВУ работают параллельно во времени. Когда ВУ готово к передаче слова данных, оно формирует сигнал запроса, поступающий в процессор. По этому сигналу работа процессора прерывается с переходом к выполнению программы ввода-вывода, обслуживающей данное ВУ.

Достоинство рассматриваемой структуры – малые затраты оборудования в системе ввода-вывода за счет того, что процессор реализует функции по передаче вводимых/выводимых слов в оперативную память, и по подсчету числа переданных слов.

Однако структуре с программно-управляемым интерфейсом присущи следующие недостатки:

1) Снижение производительности процессора за счет:

· простоев процессора при ожидании передачи данных в ВУ (или обратно)

· выполнения процессором операций по обслуживанию передачи последовательности слов.

2) Невозможность подключения к интерфейсу ввода-вывода высоко скоростных ВУ, поскольку для передачи данных требуется несколько процессорных операций.

В связи с этим структура с программно управляемым интерфейсом используется в компьютерах с низкой интенсивностью ввода-вывода данных, например в компьютерах встраиваемых в измерительные приборы и другие технические устройства.

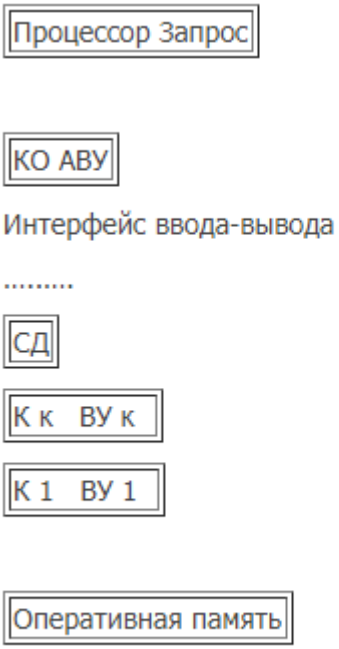
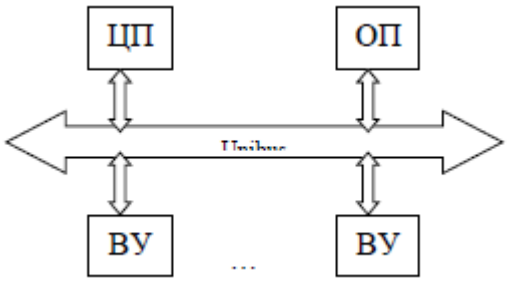


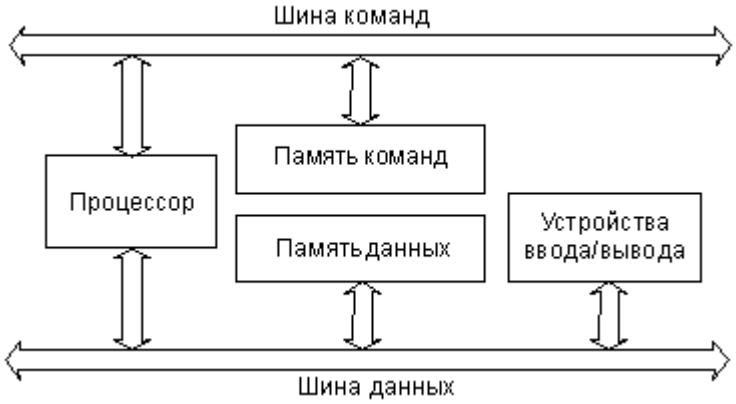
Рис. Структура компьютера с программно-управляемым интерфейсом ввода-вывода

16. Структура компьютера с общей шиной.



Основные особенности:
1. Для связи между любыми компонентами используются одни и те же линии интерфейса (ША, ШД, ШУ и т.п.), а также сигналы, управляющие передачей. Этот факт в значительной степени упрощает организации связи между устройством и обеспечивает простоту наращивания структуры путем подключения дополнительных устройств.
2. Использование единого интерфейса предполагает, что в любой момент времени по нему может быть организован обмен только между двумя устройствами, одно из которых является ведущим, а другое исполнительным, ведомым. Ведущим устройством не может быть ОП. Подобная структура является источником конфликтов между различными активными устройствами, требующими практически одновременного взаимодействия с шиной для передачи данных. Компьютеры с подобной структурой не предполагают значительного наращивания числа устройств. Подобная структура для увеличения производительности требует введения дополнительных шин для разгрузки основной.
3. В компьютерах с общей шиной могут быть реализованы различные способы организации В/В, такие как PIO, В/В по прерыванию, а также В/В в режиме DMA.

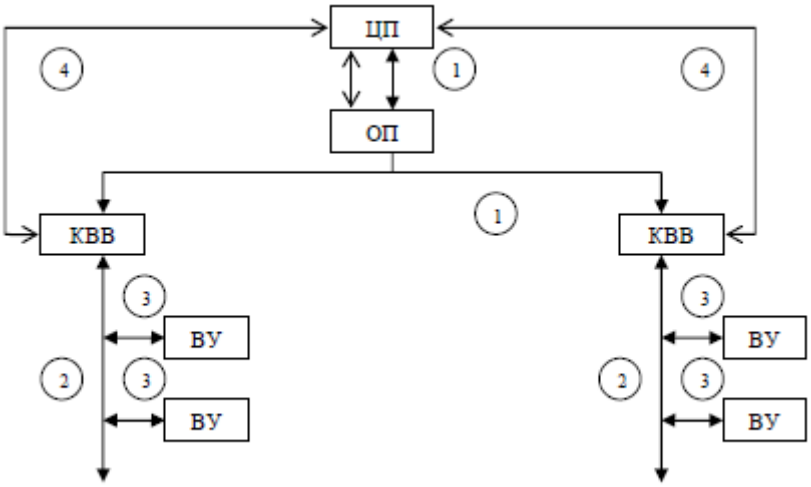
17. Структура компьютера с раздельными шинами.



Эта *архитектура* предполагает наличие в системе отдельной *памяти* для данных и отдельной *памяти* для команд (рис. 1.16). Обмен процессора с каждым из двух типов *памяти* происходит *по* своей *шине*.

Архитектура с общей *шиной* распространена гораздо больше, она применяется, например, в персональных компьютерах и в сложных *микрокомпьютерах*. *Архитектура* с раздельными *шинами* применяется в основном в однокристальных *микроконтроллерах*.

18. Структура компьютера с каналами ввода-вывода.



В структуре с каналами ввода/вывода используется *несколько видов интерфейсов*:
1. Интерфейс ОП – обеспечивает связь между основной памятью и центральным процессором с одной стороны, а также каналами ввода/вывода с другой стороны. Информационная ширина этого интерфейса определяется длиной слова ОП (шириной выборки из ОП). Для различных моделей ЭВМ IBM 360/370/390 типичными значениями ширины выборки являлись 16/32/64 бита.

2. Интерфейс ввода/вывода – предназначен для организации обмена между каналами ввода/вывода и ВУ, подключенного к ним. Посредником в организации этого обмена со стороны ВУ является устройство управления (контроллер ВУ).

3. Интерфейс внешних (периферийных) устройств.

4. Интерфейс ЦП – КВВ – по этому интерфейсу передача данных не осуществляется. По нему от ЦП к КВВ передаются управляющие сигналы, а в обратном направлении – осведомительные сигналы (сигналы о состоянии КВВ и ВУ, подключенных к нему).
Интерфейс ввода/вывода является стандартизованным, в свою очередь интерфейсы ОП и внешни устройств являются специализированными. Интерфейс ОП является моделезависимым, то есть ег характеристики изменяются от модели к модели. Интерфейсы периферийных устройств являютс разнообразными в зависимости от вида подключения по ним ВУ.

19. Функции канального процессора.

Основные функции КВВ:

1) Функции по установлению логической связи между ВУ и ОП.

а) прием и декодирование команд ввода/вывода от ЦП;
б) инициирование выполнения канальной программы при получении команды SIO (Start Input/Output) от ЦП;
в) проверка состояния ВУ, участвующего в обмене, и передача в ЦП информации о его готовности или неготовности к обмену;
2) Функции, связанные с непосредственной передачей данных между ВУ и ОП.
а) последовательная выборка команд канальной программы из ОП, их декодирование и выполнение;
б) обеспечение приема, передачи, контроля и промежуточного хранения данных при обмене между ОП и ВУ;
в) формирование текущих адресов ОП, по которым записываются или считываются передаваемые данные;

г) согласование форматов данных, передающихся по интерфейсу ввода/вывода, с форматом интерфейса ОП (как правило, ширина интерфейса ввода/вывода составляет 1, 2 или 4 байта, что меньше ширины интерфейса ОП: 4, 8, 16 байт);
д) подсчет числа передаваемых байт данных с целью определения момента завершения передачи блока данных;
е) выработка последовательности синхронизирующих и управляющих сигналов в соответствии со стандартом интерфейса ввода/вывода;
ж) анализ особых ситуаций в ВУ во время обмена (ошибка передаваемых данных, сбой устройства и т.п.) и информирование ЦП об этих ситуациях (с помощью запроса прерывания);
3) Функции, связанные с завершением обмена и разрушением логической связи между ВУ и ОП.
а) определение момента завершения в программе по организации обмена между ВУ и ОП;
б) передача в ЦП сигнала прерывания о завершении обмена.

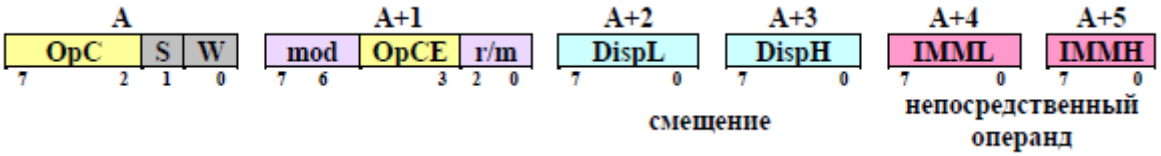
20. Система команд компьютера. Основные виды команд.

Система команд компьютера включает в себя перечень машинных команд, реализуемых непосредственно аппаратными средствами. Именно система команд определяет возможности компьютера в плане решения разнообразных задач обработки данных.
Основной характеристикой системы команд является её мощность. Обычно под этим термино понимается количество разнообразных мнемокодов, используемых для символическог обозначения на ассемблере.

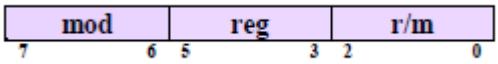
21. Структура и формат машинных команд.

Структура машинной команды задаёт основные части (поля) машинной команды и определяет их назначение. В свою очередь, формат машинной команды определяет разрядность, как всей команды, так и отдельных её полей. Под полем принято понимать совокупность последовательных битов формата (команды или данных), которые имеют определённое общее назначение. Примерами отдельных полей машинных команд могут являться
- **поле кода операции (OpC);**
- **адрес базового или индексного регистров (Base, Index);**
- **смещение(Disp);**
- **непосредственный операнд (IMM);**

Длина машинной команды без учета возможных префиксных байтов составляет от 1 до 6 байт. Префиксные байты имеют специальный код, отличный от кода операции и оказывают то или иное влияние на выполнение только одной машинной команды, следующей за префиксом. Виды префиксов : Rep (повторения), Seg (замена сегмента), Lock (блокировка шины).
В базовой модели Intel 8086 используется более 10 разнообразных форматов команд. Приме формата команды максимальной длины (двухоперандная команда с постбайтом адресации непосредственным операндом)



Специальные биты OpC - W(Word) и S(Sign Extended). Расширение Word определяет длину операндов (W=0 -> байт, W=1 -> слово). А S=1 - необходимость знакового расширения непосредственного операнда.
Знаковое расширение операнда имеет место только для комбинации S,W = 1,1, при этом в машинной команде задаётся только один младший байт (IMML) непосредственного операнда, т. к. операнд команды, задаваемый постбайтом адресации, является двухбайтным, то для приведения непосредственного операнда к тому же формату (слово) производится предварительное его расширение на старший байт. При этом операнды рассматриваются как целые знаковые числа, естественно, представленные в дополнительном коде. В связи с чем, старший байт IMMH заполняется нулями для положительного операнда или единицами для отрицательного. Фактически, все биты старшего байта становятся равными знаковому биту младшего байта, что и называется знаковым расширением операнда. При W=0, бит S становится неактуальным.
Постбайт адресации используется в команде для задания режимов адресации, в принципе, для обоих операндов. В приведенном примере формата, постбайт адресует только один операнд, поскольку второй из операндов задан непосредственно в команде.
Структура постбайта адресации для двухадресной команды имеет вид:



Т. к. поле reg в рассматриваемом формате не используется, то на его месте задается расширение кода операции. Назначение полей постбайта см. Эл. Консп.
Для операнда, находящегося в памяти, поле r/m (register/memory) задаёт компоненты эффективного адреса Base и/или Index неявным образом (см. таблицу кодирования). В базовой модели для базовой компоненты EA могут использоваться только регистры BX и BP. А для индексной компоненты только регистры SI и DI.

22. Понятие CISC и RISC-архитектуры.

Непосредственно с мощностью использования системы команд связаны два основных направления в архитектуре компьютера:
CISC – Complex Instruction Set Computer – компьютер с расширенной системой команд
RISC – Reduced Instruction Set Computer - компьютер с сокращенной системой команд.
Вся история развития компьютеров с CISC - архитектурой сопровождалась постоянным развитием и расширением их системы команд. Статистические исследования, широко проводимые в 70 –е годы в отношении частоты использования различных команд, позволили сформулировать достаточно актуальный в своё время принцип “80X20”.Суть которого в том, что на 20% машинных команд из общей системы команд процессор затрачивает порядка 80% своего бюджета, т. е. очень многие машинные команды оказываются практически невостребованными при разработке программных продуктов.

Сложность используемой системы команд в первую очередь сказывается на сложности устройства управления. При реализации CISC – процессора на одном кристалле в виде СБИС, на долю устройства управления приходится от 30% до 60% площади кристалла. В соответствии с реализацией принципов микропрограммного управления, используемого во всех CISC - процессорах, в состав устройства управления включена достаточно большая микропрограммная память для хранения микрокодов по реализации различных машинных команд. Например, в ЭВМ VAX – 11 ёмкость микропрограммной памяти составляет 50 Кбайт.

23. Особенности компьютеров RISC-архитектуры.

Основные особенности RISC – архитектуры.
Термин RISC был впервые введён в 1980 г. Дэвидом Паттерсоном, профессором Калифорнийского Университета.
1) Использование сравнительно небольшого множества машинных команд, наиболее часто используемых при решении задач обработки данных. Первые модели RISC – процессоров (середина и конец 80- ых г.г.) имели мощность системы команд менее 100, в современных моделях RISC – процессоров - примерно 150. Система команд современного RISC – процессоров включает в себя целочисленную арифметику, арифметику с плавающей точкой, мультимедийные расширения(векторные команды).
2) Стремление к выполнению большинства машинных команд за 1 машинный такт (машинный цикл).

Под одним машинным тактом понимается интервал времени между двумя последовательными синхросигналами, подаваемыми на вход процессора от генератора (как правило, внешняя микросхема). Величина, обратная длительности машинного такта, называется тактовой частотой, это одна из самых важных характеристик процессора, как и компьютера, определяющая его быстродействие (производительность).
За 1 машинный такт в процессоре выполняются элементарные действия, называемые микрооперациями, например, пересылка между двумя регистрами или выполнение элементарных операций в АЛУ, таких как сложение или логическое умножение. Управляющее слово, инициирующее выполнение одной или нескольких совместимых во времени микроопераций, называется микрокомандой.
Совокупность микрокоманд для реализации какой – нибудь сложной машинной команды называется микропрограммой. Микропрограммы составляются и отлаживаются при проектировании компьютера и хранятся в постоянной памяти, называемой памятью микропрограммы. Она входит в состав блока (устройства управления). В RISC – процессорах грань между машинной командой и микрокомандой практически стирается.
3) (как следствие из 2)) В RISC – процессорах для реализации устройства управления используется принцип жесткой логики как альтернатива принципа микропрограммной логики. Это означает, что устройство управления реализуется как схемный автомат с использованием автоматной модели Мили Мура. В виду упрощения устройства управления площадь, занимаемая им на кристалле для RISC – процессоров составляет 5 – 10 %, а для CISC – процессоров – 30 -50%

Освобождённая площадь кристалла используется для:
□ Увеличения внутренней регистровой памяти. В современных моделях RISC – процессорах число внутренних регистров может быть несколько сотен (до 500).

 Увеличение объёма внутренней Кэш – памяти. Типичный размер внутрикристальный Кэш первого уровня 64 – 128 Кбайт.

Замечание: в некоторых моделях используется внутрикристальная Кэш – память и второго уровня.
 Внедрение в кристалл дополнительных блоков для реализации векторной обработки (мультимедийные расширители).
4) Использование большого числа внутренних регистров создаёт дополнительную сложность при выходе на обработку прерывания, связанные с необходимостью сохранения контекста прерываемой программы. Для уменьшения времени издержек на переключение программ в RISC – процессорах зачастую используется механизм регистровых окон. Идея состоит в том, что каждой программе выделяется некоторое подмножество регистров, образующих окно.
5) Ограниченное число форматов машинных команд и используемых режимов адресации. Используется 3 – 5 форматов машинных команд фиксированной длины и 2 - 3 основных режима адресации (косвенная адресация в RISC – процессорах не используется). Всё это делается для максимального упрощения процесса декодирования команды и формирования адресов операндов, что в свою очередь приводит на минимизацию затрат на блок управления.
6) Все команды обработки данных реализуются только над регистровыми операндами (команды типа reg/reg). Для обмена с памятью используются специальные команды типа LOAD (memory -> reg) и STORE (reg -> memory).
7) Широкое использование принципов суперскалярной и суперконвейерной обработки.

24. Способы адресации, используемые в ЭВМ.

1. Прямая
памяти
 регистровая
2. Косвенная
 памяти
 регистровая
3. Относительная
 базовая
 индексная
 базово - индексная без смещения
 базово – индексная со смещением
 относительно текущего счетчика команд

25. Способы адресации с модификацией адреса.

С помощью постбайта адресации могут быть реализованы следующие режимы адресации:
 прямая регистровая;
 прямая адресация памяти EA = Disp;
 базовая EA = Bese + Disp;
 индексная EA = Index + Disp;
 базово – индексная без смещения EA = Base + Index;

- **базово – индексная со смещением EA = Base + Index + Disp;**
- **косвенная регистровая EA = [РОН], где РОН = BX, SI, DI.**

26. Стековая адресация. Выполнение вычислений в стековых ЭВМ (на примере).

Стековая память, реализующая безадресное задание операндов, является эффективным элементом современной архитектуры ЭВМ, особенно широко используемым в микропроцессорах, малых и микроЭВМ, а также в некоторых суперЭВМ. Учитывая своеобразие стековой адресации, ее рассмотрение выделено в отдельный параграф.

Стек представляет собой группу последовательно пронумерованных регистров (*аппаратурный стек*) или ячеек памяти, снабженных указателем стека (обычно регистром) (УС), в котором автоматически при записи и считывании устанавливается номер (адрес) последней занятой ячейки стека (вершины стека}. При операции записи заносимое в стек слово помещается в следующую по порядку свободную ячейку стека, а при считывании из стека извлекается последнее поступившее в него слово. Таким образом, в стеке реализуется правило *«последний пришел - первый ушел».*

Указанное правило при обращении к стеку реализуется автоматически, и поэтому при операциях со стеком возможно безадресное задание операнда — команда не содержит адреса ячейки стека, но содержит адрес (или он подразумевается) ячейки памяти или регистра, откуда слово передается в стек или куда помещается из стека.

Механизм стековой адресации поясняется на рис.3.7. При выполнении команды передачи в стек слова из регистра или ячейки ОП сначала указатель стека увеличивается на 1 (в *перевернутом стеке* уменьшается на 1), а затем слово помещается в ячейку стека, указываемую УС. При команде загрузки из стека регистра или ячейки памяти сначала слово извлекается из вершины стека, а затем указатель стека уменьшается на 1 (в перевернутом стеке увеличивается на 1). Как это ни кажется на первый взгляд удивительным, но при соответствующем расположении операндов в стеке можно вычислять выражения полностью безадресными командами, указывающими только вид операции. Такая команда извлекает из стека в соответствии с кодом операции один или два операнда, выполняет над ними предписанную операцию и заносит результат в стек.

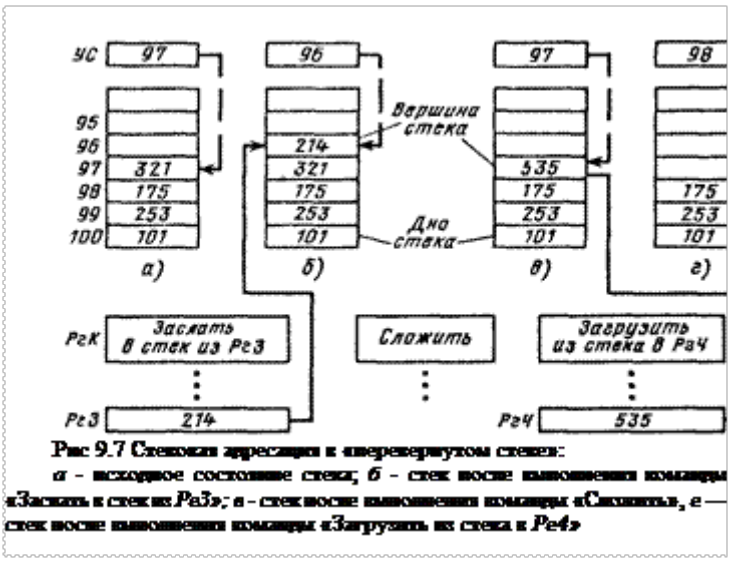


Рис 9.7 Стековая адресация и «перевернутый стек»:
а **– исходное состояние стека;** ***б*** **– стек после выполнения команды «Заслать в стек из Рг3»;** ***в*** **– стек после выполнения команды «Сложить»,** ***г*** **— стек после выполнения команды «Загрузить из стека в Рг4»**

Вычисления с использованием стековой памяти удобно описывать и программировать с помощью польской инверсной (бес скобочной) записи арифметических выражений ПОЛИЗ. Эта запись производится по следующему правилу: читаем арифметическое выражение слева направо и последовательно друг за другом выписываем встречающиеся операнды. Как только окажется, что все операнды некоторой операции выписаны, записываем знак этой операции и продолжаем выписывать операнды. Если операция имеет операндом результат некоторой предыдущей операции и знак последней выписан, то считаем этот операнд выписанным.

Например, выражение

(k + l - m) (p - s)

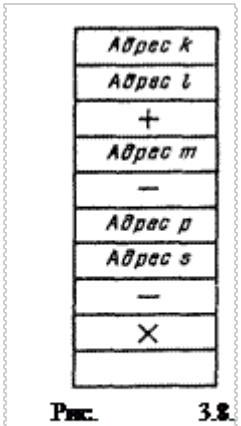


Рис. 3.8. Программа вычисления

в ПОЛИЗ имеет вид

*k l + m – p s - *.*

Выражение в ПОЛИЗ не содержит скобок, но порядок действий определяет однозначно. При использовании стековой памяти последовательность символов в выражении ПОЛИЗ, может рассматриваться как программа вычисления исходного арифметического выражения (рис. 3.8), если под буквами понимать команды засылки, содержащие только адреса в ОП соответствующих операндов, засылаемых в стек, а под знаками операции безадресные команды, содержащие только коды операций. Команда второго типа инициирует извлечение из стека двух (или одного) слов, выполнение над ними указанной в команде операции и засылку результата в вершину стека

Безадресные команды на основе стековой адресации предельно сокращают формат команд, экономят память и способствуют повышению производительности ЭВМ.

Однако при такой структуре команд возникают осложнения с построением команд передачи управления и работы с периферийными устройствами.

В современной архитектуре процессоров и микропроцессоров стек и стековая адресация широко ИСПОЛЬЗУЮТСЯ при организации переходов к подпрограммам и возврате от них, а также в системах прерывания. Весьма широко стеки и безадресные команды используются в вычислительном комплексе «Эльбрус».

27. Принцип иерархической организации памяти.

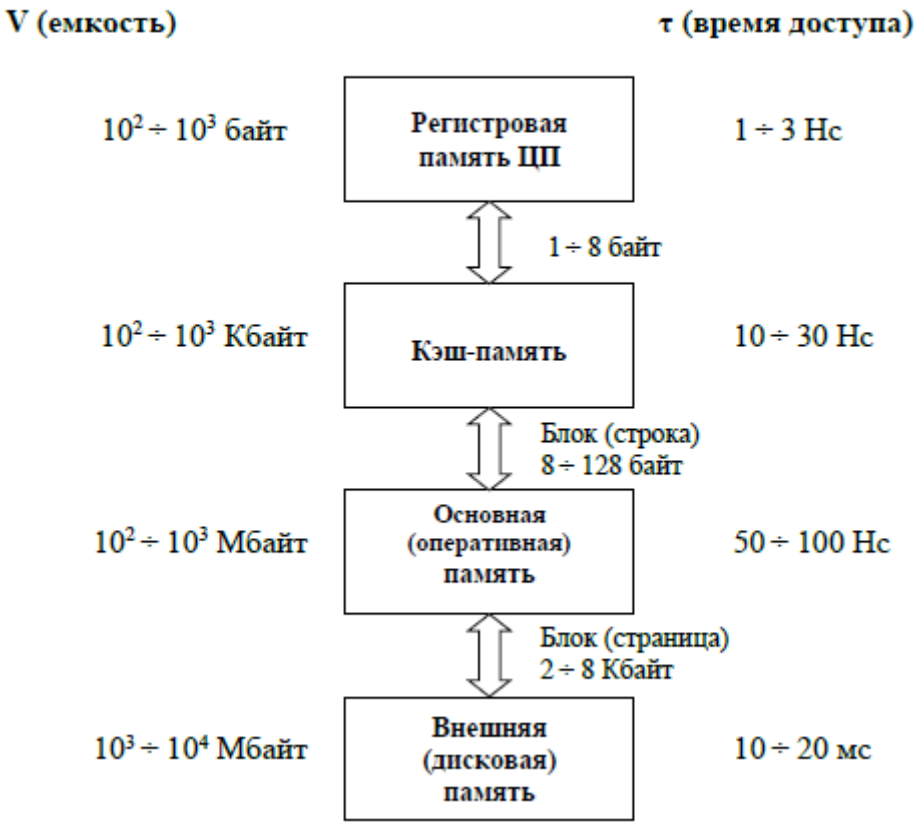
Память компьютера строится из достаточно большой совокупности разнообразных запоминающих устройств, обладающих различным принципом действия, элементной базой и характеристиками.
Основными характеристиками запоминающих устройств являются:
• емкость (объем);
• быстродействие (время доступа);
• удельная стоимость (стоимость хранения единицы информации).
Для построения памяти компьютера используется иерархический принцип (разделение ЗУ на ряд уровней), что обусловлено противоречивостью требований со стороны пользователей к различным характеристикам памяти (больше объема, меньше время доступа и дешевле).

Обмен данными в многоуровневой памяти осуществляется только между двумя соседними уровнями. В принципе, могут иметь место исключения из

этого правила, например, в виде пересылок между ОП и регистрами ЦП, минуя кэш-память.
Изменение характеристик от уровня к уровню имеет следующие закономерности:
• сверху вниз емкость памяти увеличивается;
• сверху вниз быстродействие и удельная стоимость уменьшаются.
Для двух соседних уровней все необходимые для верхнего уровня данные обязательно размещаются на нижнем уровне. Передача данных между уровнями инициируется в том случае, если на верхнем уровне при обращении к нему необходимых данных не обнаружено.
В отношении Кэш-памяти (в качестве примера) обнаружение требуемой информации называется ***cache-hit*** , отсутствие требуемой информации – ***cache-miss***.
Пересылка блоков между тремя верхними уровнями реализуется чисто аппаратными средствами, в то время как пересылка между основной и внешней памятью реализуется совместно аппаратными и программными средствами.
Эффективность использования модели иерархической памяти во многом, если не во всем, объясняется существованием так называемого ***принципа локальности обращений*** (локальности ссылок). Этот принцип рассматривается в двух аспектах: пространственном и временном в отношении как команд, так и данных.
Пространственный аспект принципа локальности в отношении команд означает, что вероятность выборки команды по следующему адресу по сравнению с адресом исполняемой команды намного больше, чем по любому другому адресу. Этот принцип проявляется на линейных участках программ. По статистике, средняя длина линейного участка составляет 5-8 машинных команд.
Пространственный аспект принципа локальности в отношении данных означает, что вероятность обращения к данным по следующему адресу по сравнению с предыдущим обращением намного больше вероятности обращения по любому другому адресу. Этот принцип наиболее ярко проявляется при обработке структур данных типа массив.
Временной аспект принципа локальности в отношении команд и данных состоит в большей вероятности повторных обращений по одним и тем же адресам за командами или данными в течение небольшого промежутка времени. В отношении команд этот принцип наиболее ярко проявляется в циклах, а в отношении данных – при повторной обработке одной и той же структуры (например, массива).

28. Основные уровни иерархии системы памяти.

Упрощенная схема иерархии памяти



29. Дополнительные уровни иерархии памяти.

Дополнение к упрощенной структуре иерархической памяти

1) Кэш-память, как правило, сама является многоуровневой (двух- или трехуровневой): L_1, L_2, L_3.
2) Использование дискового КЭШа как промежуточного уровня между ОП и ВП.

Возможные реализации:

• программная, путем выделения некоторого буфера в ОП;

- аппаратная, в виде самостоятельного ЗУ, включаемого в состав НМД (накопителя на магнитном диске).

30. Основные характеристики системы памяти.

Основными характеристиками запоминающих устройств являются:
• емкость (объем);
• быстродействие (время доступа);
• удельная стоимость (стоимость хранения единицы информации).

31. Энергозависимая и энергонезависимая память.

Энергонезависимая память, **NVRAM** (англ. *Non Volatile Random Access Memory*) — разновидность запоминающих устройств с произвольным доступом, которые способны хранить данные при отсутствии электрического питания. Может состоять из модуля SRAM, соединённого со своей собственной батарейкой. В другом случае SRAM может действовать в связке с EEPROM, например, флеш-памятью[1].

В более общем смысле, **энергонезависимая память** — любое устройство компьютерной памяти, или его часть, сохраняющее данные вне зависимости от подачи питающего напряжения. Однако подпадающие под это определение носители информации, ПЗУ, ППЗУ, устройства с подвижным носителем информации (диски, ленты) и другие носят свои, более точные названия.

Поэтому термин «энергонезависимая память» чаще всего употребляется более узко, по отношению к полупроводниковой БИС запоминающего устройства, которая обычно выполняется энергозависимой, и содержимое которой при выключении обычно пропадает.

Энéргозави́симая пáмять (англ. *Volatile memory*) — компьютерная память, которая требует постоянного использования электропитания для возможности удерживать записанную на неё информацию. Эта особенность является ключевым отличием энергозависимой памяти от ***энергонезависимой*** — последняя сохраняет записанную на неё информацию даже после прекращения подачи электропитания на неё. Энергозависимая память также изредка называется ***врéменной памятью*** (англ. *temporary memory*).

Подавляющее большинство современных видов оперативной памяти с произвольным доступом являются энергозависимыми. Сюда относятся динамическая (DRAM) и статическая (SRAM) память с произвольным доступом. Ассоциативная память и DPRAM как правило реализуются через энергозависимую память. К ранним технологиям энергозависимой памяти относятся память на линиях задержки и запоминающая электронно-лучевая трубка.

32. ROM и RAM память.

Основной функцией памяти является хранение информации и обеспечение селективного доступа к ней. Основная память представляет собой в основном память типа RAM (Random Access Memory – с произвольным доступом).
Произвольность доступа означает, что время обращения к любой ячейке памяти по заданному адресу инвариантно к этому адресу. Память типа RAM является энергозависимой.
Для сохранения наиболее важной информации часть адресного пространства ОП реализуется в виде ROM (Read Only Memory – постоянная память).
Память ROM с возможностью перезаписи называется PROM (Programming ROM – полупостоянная память).

Операти́вная па́мять (англ. *Random Access Memory, RAM*, память с произвольным доступом) или **операти́вное запомина́ющее устро́йство** (**ОЗУ**); комп. жарг. *па́мять*, *операти́вка* — энергозависимая часть системы компьютерной памяти, в которой во время работы компьютера хранится выполняемый машинный код (программы), а также входные, выходные и промежуточные данные, обрабатываемые процессором.

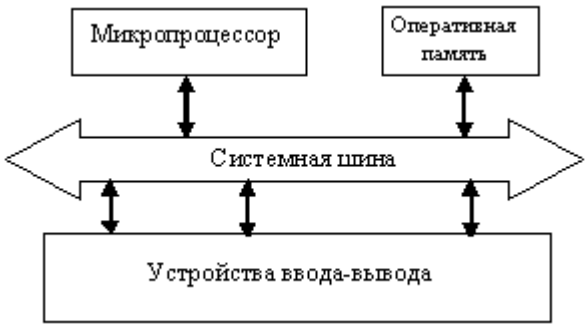
Обмен данными между процессором и оперативной памятью производится:

- непосредственно;
- через сверхбыструю память 0-го уровня — регистры в АЛУ, либо при наличии аппаратного кэша процессора — через кэш.

Содержащиеся в современной полупроводниковой оперативной памяти данные доступны и сохраняются только тогда, когда на модули памяти подаётся напряжение. Выключение питания оперативной памяти, даже кратковременное, приводит к искажению либо полному разрушению хранимой информации.

Энергосберегающие режимы работы материнской платы компьютера позволяют переводить его в режим *сна*, что значительно сокращает уровень потребления компьютером электроэнергии. В режиме *гибернации* питание ОЗУ отключается. В этом случае для сохранения содержимого ОЗУ операционная система (ОС) перед отключением питания записывает содержимое ОЗУ на устройство постоянного хранения данных (как правило, жёсткий диск). Например, в ОС Windows XP содержимое памяти сохраняется в файл `hiberfil.sys`, в ОС семейства Unix — на специальный swap-раздел жёсткого диска.

В общем случае, ОЗУ содержит программы и данные ОС и запущенные прикладные программы пользователя и данные этих программ, поэтому от объёма оперативной памяти зависит количество задач, которые одновременно может выполнять компьютер под управлением ОС.



33. Организация оперативной памяти по принципу 1D, 2D, 3D.

Различные варианты организации памяти с М-поискомсвязаны, прежде всего, с различными способами построения массива ЗЭ и декодирования адреса. С этой точки зрения выделяют память типа 1D, 2D, 2,5D, 3D, 4D – по количеству измерений массива ЗЭ. В памяти типа 1D массив имеет 1 измерение, то есть адресуется каждый бит памяти. При достаточно большом объеме ЗУ это приводит к сложным схемам дешифраторов и огромному количеству служебных линий, трассировка которых внутри кристалла вызывает проблемы, а площадь, занимаемая ими, сопоставима

с площадью массива самих ЗЭ. В2D-памяти(рис. 2.2) адресуются не отдельные биты, а слова, что улучшает общую картину.

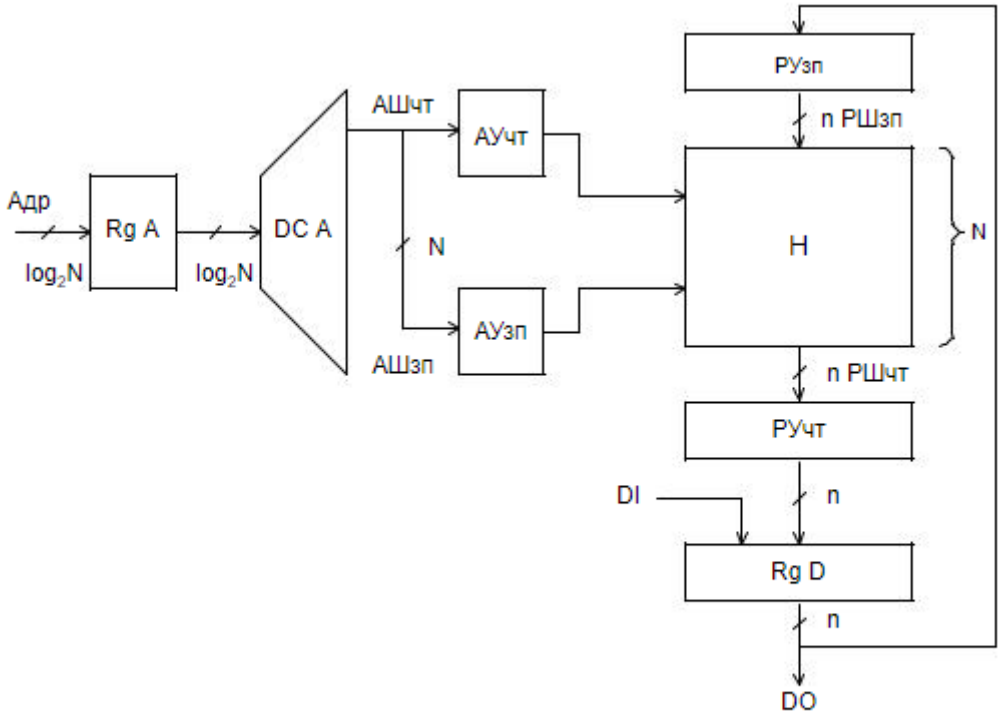


Рис. 2.2

Для большей экономии кристалла необходимо использовать слова еще большей разрядности, что входит в противоречие с разрядностью шин данных ВС, и создает дополнительные неудобства. Для их преодоления используют организацию типа 2,5D (рис. 2.3), при которой слова системной разрядности (16, 32, 64 и т.д.) объединяются в группы, адрес ячейки при декодировании в ЗУ делится на 2 части, большая из них используется для выбора группы, а меньшая – для выбора слова внутри группы.

При 3D организации (рис. 2.4) массив ЗЭ имеет два измерения, то есть выбор ячейки (слова) осуществляется по двум координатам, при этом адрес ячейки делится на две равные части, каждая из которых используется для
выбора одной из линий по одной из двух координат. В результате и количество линий, и сложность адресных дешифраторов уменьшается. Дальнейшее развитие такого подхода приводит к памяти с организацией 4D и т.д.

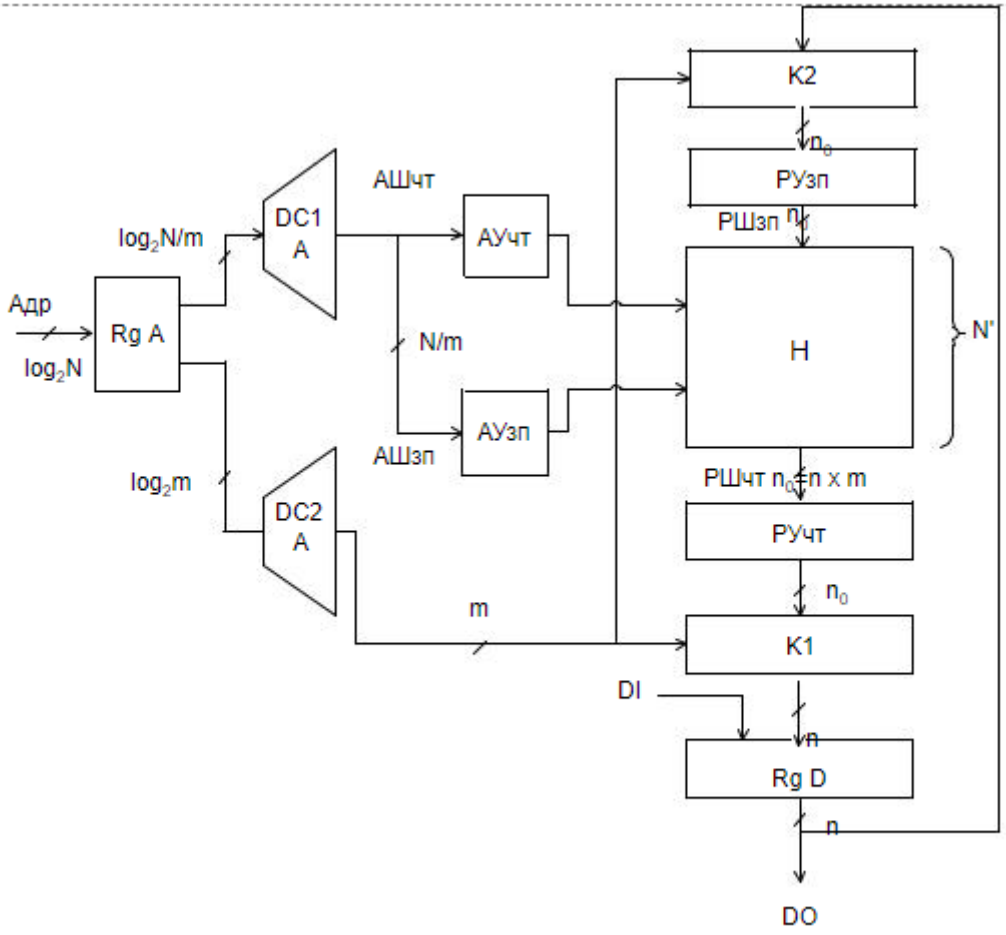

K2
n_0
РУзп
РШзп n
log2N/m
DC1
A
АШчт
АУчт
Адр
Rg A
log2N
N/m
H
N'
АУзп
АШзп
log2m
DC2
A
РШчт n_0=n x m
РУчт
n_0
m
K1
DI
n
Rg D
n
DO

Рис. 2.3

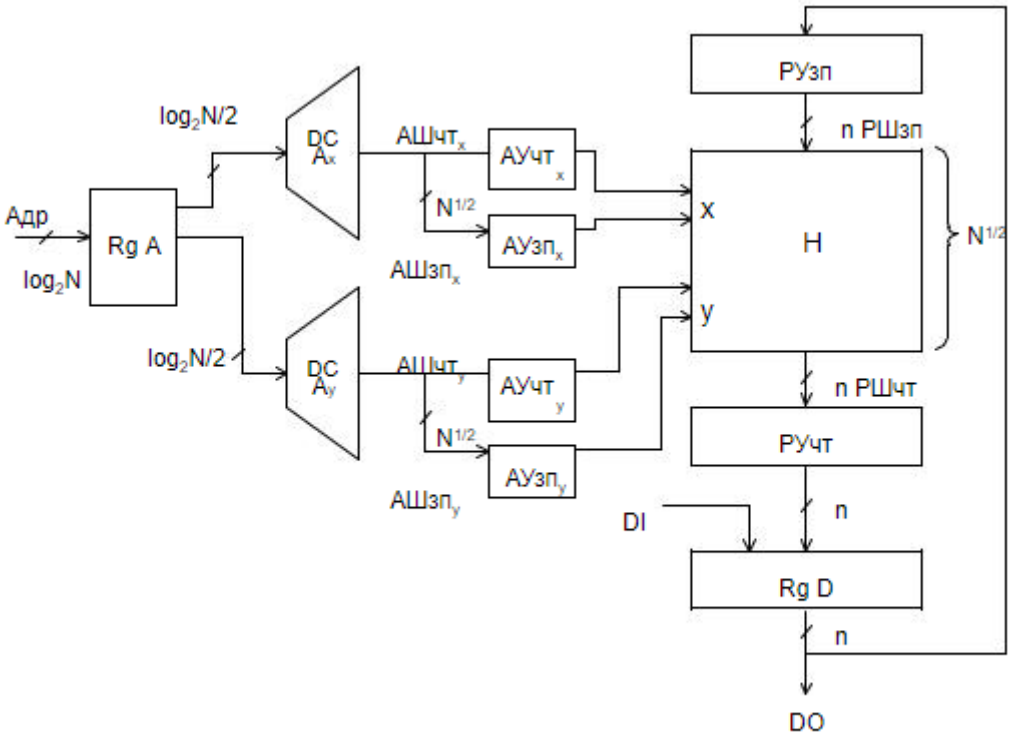


Рис. 2.4.

34. SRAM и DRAM память.

В качестве таковой в современных компьютерах используется *сверхбольшие интегральные схемы* (СБИС) на основе *CMOS технологий* (комплементарная МОП технология).
В качестве элементарной ячейки элементов памяти используется элемент типа **DRAM – Dynamic RAM**.
Основу элементов DRAM составляют один транзистор и конденсатор. Наличие заряда на конденсаторе соответствует хранению 1, отсутствие- 0.
Вследствие способности конденсатора к разрядке, содержимое динамической памяти требует постоянной перезаписи (Refresh).
Средний интервал между двумя последовательными перезарядками составляет 8-64 мксек.
Альтернативой динамических элементов памяти DRAM являются статические элементы **SRAM (Static RAM),** основу которых составляет статический триггер, создаваемый на 4-х или 6-ти транзисторах.

Если сравнивать элементы SRAM и DRAM по основным характеристикам, то можно утверждать следующее:
1) элементы DRAM значительно дешевле;

2) элементы SRAM значительно более скоростные;

3) *элементы DRAM характеризуются большей простотой и соответственно гораздо большей плотностью упаковки на кристалле*.

Областью применения элементов DRAM в современных компьютерах является основная память, в то время как на элементах SRAM реализуется кэш-память.

35. Привилегированные операции и состояния процессора.

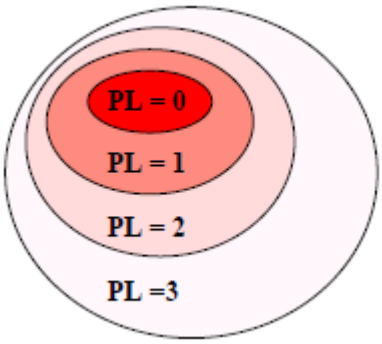
В целях разграничения доступа к системным ресурсам со стороны прикладных и системных программ, в современных моделях процессоров в том или ином виде существует два основных режима функционирования:

прикладной
- системный.
В системном режиме допускается исполнение любых машинных команд. В прикладном режиме не допускается исполнение так называемых привилегированных команд. В простейшем случае режим работы процессора задаётся с помощью специального бита, находящегося в каком – либо системном регистре. Например, в процессоре IBM 370 бит режима находится в слове состояния программы PSW (Program Status Word).Два альтернативных состояния процессора называются Task/Supervisor.
Аналогичный бит режима System/User имеет место в моделях системы VAX(DEC), который находится в PS (Processor Status).
В процессорах семейства 80X86 Pentium используется более сложная интерпретация (способ) для задания режима работы процессора в плане разделения системного и пользовательского режимов. С помощью специального бита PE (Protect Enabled) в управляющем регистре CR0 задаётся или реальный режим (PE = 0) или защищённый режим (PE = 1).
При использовании защищённого режима вступают в действие различные средства защиты. Одним из средств защиты, поддерживающимся на аппаратном уровне, является защита по уровням привилегий (кольцам защиты).
Аппаратно поддерживаются 4 уровня привилегий для сегментов и 2 уровня привилегий для страниц. Идея защиты по уровням привилегий состоит в присвоении различным сегментным объектам, в частности сегментам кода, данных, стека, определённого уровня привилегий. Этот уровень отражается соответствующим двухбитным полем DPL(Descriptor Privilege Level), размещаемом в дескрипторе (описателе сегмента). Уровень привилегий определяет степень важности и доступности сегмента. Наивысшим уровнем привилегий является PL = 0, и он присваивается программам ядра, наинизший уровень PL = 3, он присваивается прикладным программам.

Общее правило защиты предполагает возможность обращений или доступа из внутренних колец Во внешние. Попытки обращений из внешних колец во внутренние в общем случае пресекаются средствами защиты с выходом на прерывание специального типа Тип 13 (“Нарушение общей защиты”).

Пользовательский режим ассоциируется с уровнем привилегии PL = 3, системный режим с уровнем привилегий PL = 0.
Современные операционные системы, в частности Windows и Unix поддерживают только 2 уровня привилегий (User/Supervisor). Соответствующие биты для двух уровней используются на страничном уровне и размещаются в страничных дескрипторах. В процессорах семейства Intel к привилегированным командам относятся:
1) Команды загрузки и сохранения системных регистров.
2) Команды манипуляции флагом IF: CLI (0 - > IF), STI (1 - >IF).
3) Команды ввода/вывода: IN/OUT, INS/OUTS.
4) Команда останова процессора - HLT.

Попытка выполнения привилегированных команд в пользовательском режиме (PL = 3 приводит к выводу на прерывание 13. Почти все привилегированные команды, за исключение команд ввода/вывода и манипуляций флагом IF требует для своего выполнения наивысшег уровня привилегий PL = 0.



36. Организация прерывания программ. Источники прерываний.

37. Основные сведения об организации ввода/вывода информации. Программно-управляемая передача данных.

Программно-управляемая передача данных осуществляется при непосредственном участии и под управлением процессора. Например, при пересылке блока данных из периферийного устройства в оперативную память процессор должен выполнить следующую последовательность шагов:

1. сформировать начальный адрес области обмена ОП;
2. занести длину передаваемого массива данных в один из внутренних регистров, который будет играть роль счетчика;
3. выдать команду чтения информации из УВВ; при этом на шину адреса из МП выдается адрес УВВ, на шину управления - сигнал чтения данных из УВВ, а считанные данные заносятся во внутренний регистр МП;
4. выдать команду записи информации в ОП; при этом на шину адреса из МП выдается адрес ячейки оперативной памяти, на шину управления - сигнал записи данных в ОП, а на шину данных выставляются данные из регистра МП, в который они были помещены при чтении из УВВ;
5. модифицировать регистр, содержащий адрес оперативной памяти;
6. уменьшить счетчик длины массива на длину переданных данных;
7. если переданы не все данные, то повторить шаги 3-6, в противном случае закончить обмен.

Как видно, *программно-управляемый обмен* ведет к нерациональному использованию мощности микропроцессора, который вынужден выполнять большое количество относительно простых операций, приостанавливая работу над основной программой. При этом действия, связанные с обращением к оперативной памяти и к периферийному устройству, обычно требуют

удлиненного цикла работы микропроцессора из-за их более медленной по сравнению с микропроцессором работы, что приводит к еще более существенным потерям производительности ЭВМ.

38. Режим прямого доступа к памяти.

Альтернативой *программно-управляемому обмену* служит ***прямой доступ к памяти*** - способ быстродействующего подключения внешнего устройства, при котором оно обращается к оперативной памяти, не прерывая работы процессора. Такой обмен происходит под управлением отдельного устройства - ***контроллера прямого доступа к памяти (КПДП)***.

Структура ЭВМ, имеющей в своем составе *КПДП*, представлена на рис. 9.2

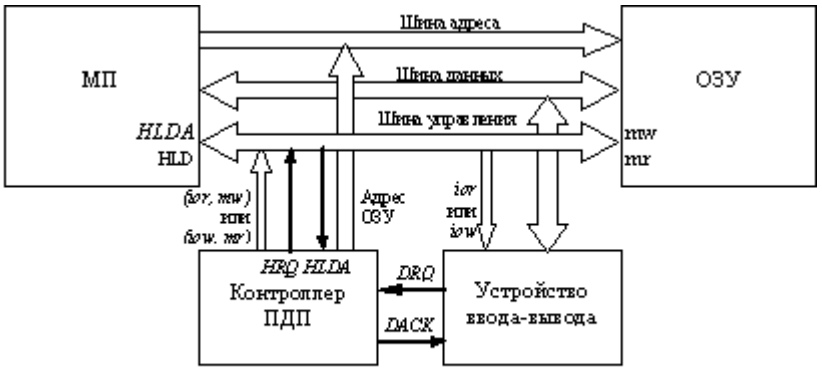


Рис. 11.2 Обмен данными в режиме прямого доступа к памяти

Перед началом работы *контроллер ПДП* необходимо инициализировать: занести начальный адрес области ОП, с которой производится обмен, и длину передаваемого массива данных. В дальнейшем по сигналу запроса прямого доступа контроллер фактически выполняет все те действия, которые обеспечивал микропроцессор при *программно-управляемой передаче*.

Последовательность действий *КПДП* при запросе на *прямой доступ к памяти* со стороны устройства ввода-вывода следующая:

1. Принять запрос на *ПДП* (сигнал DRQ) от УВВ.
2. Сформировать запрос к МП на захват шин (сигнал HRQ).
3. Принять сигнал от МП (HLDA), подтверждающий факт перевода микропроцессором своих шин в третье состояние.
4. Сформировать сигнал, сообщающий устройству ввода-вывода о начале выполнения циклов *прямого доступа к памяти*(DACK).
5. Сформировать на шине адреса компьютера адрес ячейки памяти, предназначенной для обмена.

6. Выработать сигналы, обеспечивающие управление обменом (IOR, MW для передачи данных из УВВ в оперативную память и IOW, MR для передачи данных из оперативной памяти в УВВ).
7. Уменьшить значение в счетчике данных на длину переданных данных.
8. Проверить условие окончания сеанса прямого доступа (обнуление счетчика данных или снятие сигнала запроса на *ПДП*). Если условие окончания не выполнено, то изменить адрес в регистре текущего адреса на длину переданных данных и повторить шаги 5-8.

Прямой доступ к памяти позволяет осуществлять параллельно во времени выполнение процессором программы и обмен данными между периферийным устройством и оперативной памятью.

Обычно программно-управляемый обмен используется в ЭВМ для операций ввода-вывода отдельных байт (слов), которые выполняются быстрее, чем при *ПДП*, так как исключаются потери времени на инициализацию *контроллера ПДП*, а в качестве основного способа осуществления операций ввода-вывода используют *ПДП*. Например, в стандартной конфигурации персональной ЭВМ обмен между накопителями на магнитных дисках и оперативной памятью происходит в режиме *прямого доступа*.

39. Организация обмена в режиме прямого доступа к памяти.

    См предыдущий вопрос

40. Функции контроллера ПДП.
41. Сходство и различие канального обмена и режима ПДП.

Многие специалисты сопоставляют (как синонимы) канальный В/В и DMA. Аналогия между ними состоит в том, что оба этих способа организации В/В реализуются практически без участия ЦП. Так же как и для DMA КВВ требует некоторого участия ЦП лишь на этапе инициализации В/В, при этом из ЦП в КВВ передается начальный адрес программы в памяти, а также при особых ситуациях в работе канала или ВУ. Существенным отличием КВВ от DMA является программная реализация первого и чисто аппаратная второго.

42. Сходство и различие программно-управляемого обмена и канального обмена.

Так как канал В/В осуществляет организацию обмена с ВУ по собственной программе, то КВВ следует считать программно управляемым, однако, в отличие от PIO программу, связанную с обменом, выполняет не ЦП, а специализированной процессор – КВВ.

43. Организация синхронного обмена.
44. Организация асинхронного обмена.
45. Организация обмена по прерыванию.
46. Концепция системы прерываний.

47. Основные виды источников (причин) прерывания программы.
48. Организация системы прерываний. Вектор прерывания. Понятие глубины прерывания. Уровни прерывания.
49. Функции системы прерываний.
50. Понятие приоритета прерываний. Абсолютный и относительный приоритет. Организация обработки запросов на прерывание.
51. Программирование приоритетов по маске.
52. Программирование приоритетов по порогу.
53. Программируемый контроллер прерываний (PIC).
54. Основные функции PIC.
55. Основные режимы работы PIC.
56. Принцип микропрограммного управления. Операционный и управляющий автоматы, их взаимодействие.
57. Микрооперация. Микрокоманда. Микропрограмма.
58. Принудительная адресация микрокоманд.
59. Естественная адресация микрокоманд.
60. Операционные микрокоманды.
61. Горизонтальное кодирование микрокоманд.
62. Вертикальное кодирование микрокоманд.
63. Смешанное кодирование микрокоманд.
64. Управляющие микрокоманды.
65. Управляющий автомат с хранимой микропрограммой.
66. Управляющий автомат с жесткой логикой.
67. Каноническая структура процессора.
68. Основные операционные блоки процессора.
69. Цикл выполнения машинных команд и его фазы.
70. Синхронный конвейер команд. Оценка его производительности.
71. Причины снижения производительности при конвейерном режиме обработки команд.
72. Способы повышения производительности при конвейерной обработке команд.
73. Особенности организации конвейера команд в процессорах Pentium. Структура процессора. Понятие суперскалярной архитектуры.
74. Понятие конвейера операций. Способы его организации.
75. Принцип многофункциональной обработки данных.
76. Классификация вычислительных систем по Флинну.
77. Классификация параллельных вычислительных систем.
78. Векторные и матричные процессоры.
79. Мультипроцессорные вычислительные системы
80. Мультикомпьютерные вычислительные системы.
81. Закон Амдала.
82. Способы организации параллельных вычислений

83. Понятия микропроцессора, микропроцессорной системы, микропроцессорного комплекта.
84. Микроконтроллер. Его основные особенности.
85. Классификация микропроцессоров по области применения.
86. Особенности VLIW-архитектуры.
87. Сравнение суперскалярной и EPIC-архитектуры.
88. Классификация компьютеров по области применения.
89. Понятие семантического разрыва.
90. Условия спаривания команд при их исполнении в конвейерах команд Pentium.
91. Принцип предварительной (спекулятивной загрузки данных) в архитектуре EPIC.
92. Основные функциональные блоки современных процессоров.
93. Последствия семантического разрыва.
94. Принцип использования явного параллелизма в машинном коде. (архитектура EPIC).
95. Характеристики быстродействия и производительности компьютеров.
96. FPU –назначение и особенности функционирования.
97. Основные особенности реализации суперскалярной обработки в процессорах P6.
98. Принцип масштабируемости архитектуры EPIC.
99. Основные виды аппаратных интерфейсов, используемых в современных ПК.
100. Тэговые компьютеры.
101. Процессор Pentium Pro (модель с архитектурой P6).
102. Pairing Rules (правила спаривания)