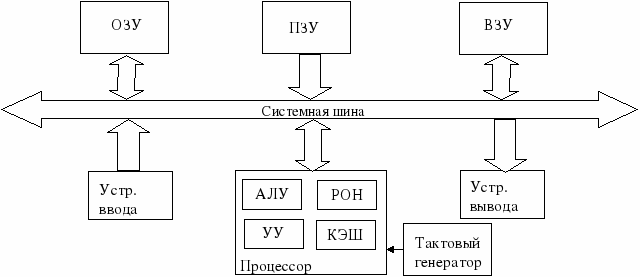
1.Понятие об ЭВМ. Структурная схема. Процессор, память, внешние устройства.

**ЭВМ** определяется как комплекс взаимосвязанных программно-управляемых технических устройств, предназначенных для автоматизированной обработки данных с целью получения результатов решения вычислительных и информационных задач.

Типичная ЭВМ состоит из **процессора, памяти и устройств ввода-вывода**.



**Процессор**  — [электронный блок](https://ru.wikipedia.org/wiki/%D0%AD%D0%BB%D0%B5%D0%BA%D1%82%D1%80%D0%BE%D0%BD%D0%BD%D0%BE%D0%B5_%D1%83%D1%81%D1%82%D1%80%D0%BE%D0%B9%D1%81%D1%82%D0%B2%D0%BE) либо [интегральная схема](https://ru.wikipedia.org/wiki/%D0%98%D0%BD%D1%82%D0%B5%D0%B3%D1%80%D0%B0%D0%BB%D1%8C%D0%BD%D0%B0%D1%8F_%D1%81%D1%85%D0%B5%D0%BC%D0%B0) ([микропроцессор](https://ru.wikipedia.org/wiki/%D0%9C%D0%B8%D0%BA%D1%80%D0%BE%D0%BF%D1%80%D0%BE%D1%86%D0%B5%D1%81%D1%81%D0%BE%D1%80)), **исполняющая** [машинные инструкции](https://ru.wikipedia.org/wiki/%D0%9C%D0%B0%D1%88%D0%B8%D0%BD%D0%BD%D1%8B%D0%B9_%D0%BA%D0%BE%D0%B4) (код программ), главная часть [аппаратного обеспечения](https://ru.wikipedia.org/wiki/%D0%90%D0%BF%D0%BF%D0%B0%D1%80%D0%B0%D1%82%D0%BD%D0%BE%D0%B5_%D0%BE%D0%B1%D0%B5%D1%81%D0%BF%D0%B5%D1%87%D0%B5%D0%BD%D0%B8%D0%B5) [компьютера](https://ru.wikipedia.org/wiki/%D0%9A%D0%BE%D0%BC%D0%BF%D1%8C%D1%8E%D1%82%D0%B5%D1%80) или [программируемого логического контроллера](https://ru.wikipedia.org/wiki/%D0%9F%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D1%83%D0%B5%D0%BC%D1%8B%D0%B9_%D0%BB%D0%BE%D0%B3%D0%B8%D1%87%D0%B5%D1%81%D0%BA%D0%B8%D0%B9_%D0%BA%D0%BE%D0%BD%D1%82%D1%80%D0%BE%D0%BB%D0%BB%D0%B5%D1%80).

**Устройство управления (УУ)** и **арифметико-логическое устройство (АЛУ)** в современных компьютерах объединены в один блок - процессор, являющийся преобразователем информации, поступающей из памяти и внешних устройств.  
**Память (ЗУ)** хранит информацию (данные) и программы. Запоминающее устройство у современных компьютеров "многоярусно" и включает **оперативное запоминающее устройство (ОЗУ)** и **внешние запоминающие устройства(ВЗУ).**  
**ОЗУ**- это устройство, хранящее ту информацию, с которой компьютер **работает непосредственно в данное время** (исполняемая программа, часть необходимых для нее данных, некоторые управляющие программы).  
**ВЗУ**-устройства гораздо большей емкости, чем ОЗУ, но существенно более медленны.

Память ЭВМ включает устройство, обеспечивающее хранение команд и данных. Это устройство состоит из блоков одинакового размера — ячеек памяти, предназначенных для хранения одного слова информации. В свою очередь, ячейка памяти состоит из элементов памяти, состояние каждого из которых соответствует одной двоичной цифре (0 или 1) Совокупность нулей и единиц. Хранящихся в элементах одной ячейки, представляет собой содержимое этой ячейки памяти.

Ячейки нумеруются числами 0. 1, 2, .. , называемыми адресом ячеек. Если необходимо записать в память слово, следует подать на шину адреса памяти сигналы, соответствующие адресу ячейки, в которую надо поместить записываемое слово, и подать само слово на шину записи. Память устроена так. Что заданное слово будет передано в ячейку с указанным адресом и может храниться там сколь угодно долго. В любой момент, обратившись к памяти, можно получить содержимое хранимого там слова. Для этого в память посылается адрес, определяющий местоположение требуемого слова, и она выдает по шине чтения копию слова. При считывании содержимое ячейки остается без изменения, так что. Один раз записав слово, можно сколько угодно раз получать его копии. Это аналогично записи песни на магнитофонную ленту Песню можно прослушивать с ленты (читать с ленты) сколько угодно раз. Но если на ее место записать другую мелодию, то первая будет стерта, однако время доступа к информации на магнитной ленте зависит от того, где записана эта информация (иногда надо перемотать почти всю ленту, чтобы прослушать какую-либо песню), тогда как время доступа к любой ячейке памяти всегда одинаково (не зависит от ее номера).

**Устройства ввода** обеспечивают считывание информации (исходных данных и программы решения задачи) с определенных носителей информации и ее представление в форме электрических сигналов, воспринимаемых другими устройствами ЭВМ (процессором или памятью)  
**Устройства вывода** представляют результаты обработки информации в форм,, удобной для визуального восприятия. При необходимости они обеспечивают запоминание результатов на носителях, с которых эти результаты могут быть снова введены в ЭВМ для дальнейшей обработки или передачу результатов на исполнительные органы управляемого объекта.

2. Принципы фон Неймана. Гарвардская и принстонская архитектуры. Примеры.

Машина фон Неймана является **абстрактной моделью** ЭВМ, однако, эта абстракция отличается от абстрактных исполнителей алгоритмов (например, машины Тьюринга). Если машину Тьюринга принципиально нельзя реализовать из-за входящей в её архитектуру бесконечной ленты, то машина фон Неймана **не поддаётся реализации**, так как многие детали в архитектуре этой машины не конкретизированы.



**ПАМЯТЬ.**

**Принцип линейности и однородности памяти.**Память – линейная (упорядоченная) однородная последовательность некоторых элементов, называемых ячейками. В любую ячейку памяти другие устройства машины могут записать и считать информацию, причём **время чтения/записи из/в любой ячейки одинаково для всех ячеек** (это и есть принцип однородности памяти).

Такая память в современных компьютерах называется памятью с произвольным доступом (Random Access Memory, RAM). На практике многие ЭВМ могут иметь участки памяти разных видов, одни из которых поддерживают только чтение информации (Read Only Memory, ROM), другие могут допускать запись, но за большее время, чем в остальную память (это так называемая полупостоянная память) и др.

Ячейки памяти в машине фон Неймана нумеруются от нуля до некоторого положительного числа N, которое обычно является степенью двойки. Адресом ячейки называется её номер.

Содержимое ячейки называется машинным словом. С точки зрения архитектуры, машинное слово – это минимальный объём данных, которым могут обмениваться различные узлы машины. Из каждой ячейки памяти можно считать копию машинного слова и передать её в другую часть памяти, при этом оригинал не меняется. При записи в память старое содержимое ячейки пропадает и заменяется новым машинным словом.

**Принцип неразличимости команд и данных.**

Машинное слово представляет собой либо команду, либо подлежащее обработке данное (для краткости – числа). Данный принцип фон Неймана заключается в том, что **числа и команды неотличимы друг от друга – в памяти и те и другое представляются некоторым набором разрядов, причём по внешнему виду машинного слова нельзя определить, что оно представляет – команду или число.**

**Принцип хранимой программы**. Этот принцип является очень важным, его суть состоит в том, что **программа хранится в памяти вместе с числами, а значит, может изменяться во время счёта этой программы**. Говорят также, что программа может самомодифицироваться во время счёта.

**УСТРОЙСТВО УПРАВЛЕНИЯ (УУ)**

**Принцип автоматической работы.** Машина, выполняя записанную в её памяти программу,

функционирует **автоматически, без участия человека**. Программа – набор записанных в памяти (не обязательно последовательно) машинных команд, описывающих шаги работы алгоритма. Таким образом, программа – это запись алгоритма на языке машины. Язык машины – набор всех возможных команд.

**Принцип последовательного выполнения команд.** УУ выполняет некоторую команду от начала до конца, а затем по определённому правилу выбирает следующую команду для выполнения, затем следующую и т.д. Этот процесс продолжается, пока не будет выполнена специальная команда останова, либо при выполнении очередной команды не возникнет аварийная ситуация (например, деление на ноль). Аварийная ситуация – это аналог безрезультативного останова алгоритма.

**АРИФМЕТИКО-ЛОГИЧЕСКОЕ УС-ВО (АЛУ)**В архитектуре машины фон Неймана АЛУ может выполнить следующие действия.

1. **Считать** содержимое некоторой ячейки памяти – поместить копию машинного слова из этой

ячейки в ячейку, расположенную в самом АЛУ.

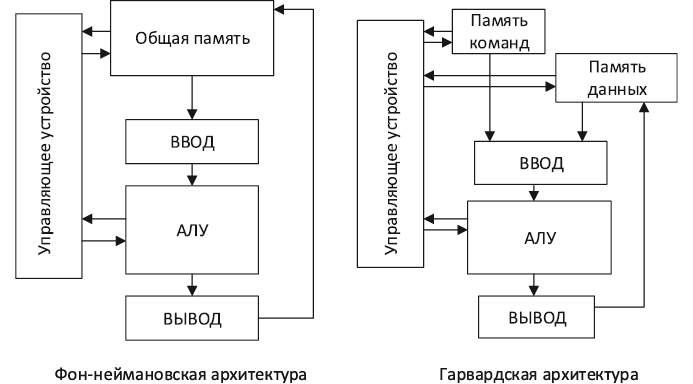
2. **Записать** в некоторую ячейку памяти – поместить копию содержимого регистра АЛУ в ячейку памяти.

3. АЛУ может также выполнять различные **операции над данными** в своих регистрах, например,

сложить содержимое двух регистров (обычно называемых регистрами первого R1 и второго R2

операндов), и поместить результат на третий регистр (называемый, как правило, сумматором S).

Революционность идей фон Неймана заключалась в специализации: **каждое устройство отвечает за выполнение только своих функций**. Теперь память только хранила данные, АЛУ производило арифметико-логические операции над ними, устройство ввода только вводило данные из «внешнего мира» в память и т.д. Фон Нейман распределил функции между различными устройствами, что существенно упростило схему машины.

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


В чистой архитектуре фон Неймана процессор в каждый момент времени может либо читать инструкцию, либо читать/записывать единицу данных из/в памяти. Оба действия одновременно происходить не могут, поскольку инструкции и данные используют один и тот же поток (шину).

В компьютере с использованием гарвардской архитектуры процессор может считывать очередную команду и оперировать памятью данных одновременно и без использования кэш-памяти. Таким образом, компьютер с гарвардской архитектурой при определенной сложности схемы быстрее, чем компьютер с архитектурой фон Неймана, поскольку потоки команд и данных расположены на раздельных физически не связанных между собой аппаратных каналах.

Исходя из физического разделения шин команд и данных, разрядности этих шин (следовательно, и адресные пространства) могут различаться и физически не могут пересекаться.

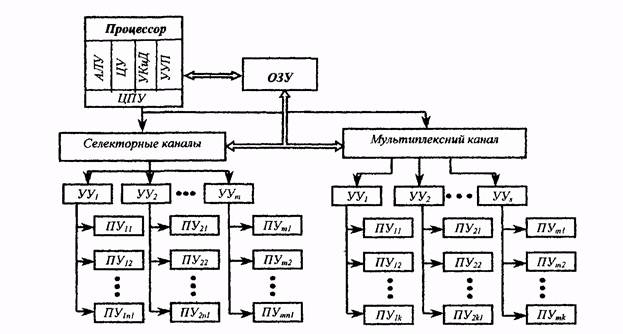
Основным преимуществом архитектуры Фон Неймана является то, что она упрощает устройство микропроцессора, так как реализует обращение только к одной общей памяти.

Гарвардская архитектура выполняет команды за меньшее количество так­тов, чем архитектура Фон Неймана. Это обусловлено тем, что в Гарвардской архитектуре больше возможностей для реализации параллельных операций. Выборка следующей команды может происходить одновременно с выполне­нием предыдущей команды, и нет необходимости останавливать процессор на время выборки команды.

ПРИМЕРЫ  
Гарвардская: микроконтроллеры (PIC, AVR), цифровой сигнальный процессор  
Принстонская: пк

**3. Структурная схема ЭВМ класса mainframe (сервер). Примеры**

В настоящее время различают архитектуру больших универсальных компьютеров-мейнфреймов (наиболее типичные представители – компы серии IBM 360/370 и их «потомки» ES9000)   
Их особенностью является **параллельная и асинхронная работа процессора и специализированных процессоров ввода-вывода – каналов ввода-вывода**. Каналы ввода-вывода полностью управляют всеми периферийными устройствам. Взаимодействие периферийных устройств с каналами и каналов с процессором обеспечивается **системой прерывания**. Если при выполнении программы возникает необходимость в работе периферийного устройства, то процессор инициализирует канал на выполнение данной операции, после чего продолжает выполнять основную программу. О завершении своей работы канал сообщает процессору прерыванием. Такая архитектура наиболее эффективна в понимании быстродействия, но требует больших аппаратных затрат (каналы ввода-вывода по своей архитектуре более сложные, чем процессор), сложного управления и имеют более низкую архитектурную надежность.



**Процессор имеет АЛУ, ус-во центрального управления (ЦУ), ус-во управления памятью (УУП) и ус-во контроля и диагностики (УКиД**). АЛУ выполняет арифметические и логические операции над двоичными и двоично-десятичными числами. ЦУ обеспечивает микропрограммное управление всего процессора, обработку прерываний и отсчет времени. УУП обеспечивает связь процессора и каналов ввода-вывода с ОЗУ, решение конфликтов при обращении к памяти и буферизацию информации, которая передается. УКиД обеспечивает текущий контроль функционирования компьютера при инициализации системы.  
Мультиплексный канал является специализированным процессором ввода-вывода и обеспечивает ввод-вывод информации из медленно-действующих периферийных ус-ств (ПУ). Он работает в мультиплексном режиме, т.е. после чтения/записи одного байта информации из одного ПУ возможен обмен байтом информации с другими более приоритетными устройствами.

Селекторные каналы также являются специализированными процессорами ввода-вывода, но они предназначены для работы с периферийными быстродействующими ус-вами, например, с ус-вами внешней памяти, накопителями на дисках. Селекторный канал работает в селекторном режиме, т.е. если начался обмен информацией с одним ус-вом, то он не может быть прерван другим, даже более приоритетным.

Все периферийные ус-ва подключаются к каналу через свои УУ, что обеспечивает стандартное подключение разнотипных ус-ств к каналам.

Используются в критически важных областях, где требуется высокая производительность и надежность.

Основное отличие суперкомпьютеров от мэйнфреймов: все ресурсы такого компьютера обычно направлены на то, чтобы решить одну или в крайнем случае несколько задач насколько возможно быстро, тогда как мэйнфреймы, как правило, выполняют довольно большое число задач, конкурирующих друг с другом.

**4. Структурная схема ЭВМ открытой шинной архитектуры. Системная плата. Примеры.**

Наличие интеллектуальных **контроллеров внешних устройств** стало важной отличительной чертой машин третьего и четвертого поколений.

**Контроллер** можно рассматривать как специализированный процессор, управляющий работой внешнего устройства. Такой процессор имеет собственную систему команд. **Результаты выполнения каждой операции заносятся во внутренние регистры памяти контроллера и могут быть в дальнейшем прочитаны центральным процессором.** Контроллер управляет своим устройством ввода-вывода и для этого регулирует доступ к шине. Задача контроллера состоит в том, чтобы разбить поток битов на фрагменты и записывать каждый такой фрагмент по мере накопления битов для него в память. Когда передача данных заканчивается, контроллер выдает прерывание, вынуждая центральный процессор приостановить работу текущей программы и начать выполнение особой процедуры. Эта процедура называется **программой обработки** прерываний и нужна она для того, чтобы проверить, нет ли ошибок, в случае их обнаружения произвести необходимые действия и сообщить операционной системе, что процесс ввода-вывода завершен. Когда программа обработки прерывания завершается, процессор возобновляет работу программы, которая была приостановлена в момент прерывания.

Центральный процессор при необходимости произвести обмен выдает задание на его осуществление контроллеру. Дальнейший обмен информацией может протекать под руководством контроллера без участия центрального процессора. Последний получает возможность «заниматься своим делом», т.е. выполнять программу дальше.



Из рисунка видно, что для связи между отдельными функциональными узлами ЭВМ используется **общая шина** (часто ее называют магистралью). Шина состоит из трех частей:

• **шина данных**, по которой передается информация;

• **шина адреса**, определяющая, куда передаются данные;

• **шина управления,** регулирующая процесс обмена информацией.

**Описанную схему легко пополнять новыми устройствами - это свойство называют открытостью архитектуры**. Для пользователя открытая архитектура означает возможность свободно выбирать состав внешних устройств для своего компьютера, т.е. конфигурировать его в зависимости от круга решаемых задач.

Обычно устройство представляет собой металлический корпус с большой интегральной схемой на дне, которая называется **материнской платой** . Материнская плата содержит микросхему процессора, несколько разъемов для модулей DIMM и различные вспомогательные микросхемы. Еще на материнской плате располагаются **шина** (она тянется вдоль платы) и **несколько разъемов для подсоединения устройств ввода-вывода.**

Структура типовой материнской платы:  
- процессор, установленный в специальный разъем и охлаждаемый радиатором

- микросхемы кэш-памяти второго уровня (внешней)

-слоты для установки модулей оперативной памяти

-слоты для установки карт расширения (видеоадаптера, звуковой карты, модема и проч) ОПРЕДЕЛЯЕТ ОТКРЫТУЮ АРХИТЕКТУРУ ПК

-микросхема перепрограммируемой памяти, в которой хранятся БИОС, программы загрузки ОС, драйверы устройств, начальные установки

-разъемы для подключения накопителей HDD, FDD

Примеры шин: ISA, PSI, PSIe

**5. Процессор. Регистры (команд, данных, адреса). УУ и АЛУ. Микропрограммная организация. Примеры.**

Центральный процессор — это мозг компьютера. Его задача — выполнять программы, находящиеся в основной памяти. Для этого он вызывает команды из памяти, определяет их тип, а затем выполняет одну за другой. Компоненты соединены шиной, представляющей собой набор параллельно связанных проводов для передачи адресов, данных и управляющих сигналов. Шины могут быть внешними (связывающими процессор с памятью и устройствами ввода-вывода) и внутренними. Современный компьютер использует несколько шин.

Процессор состоит из УУ, АЛУ, блок управления регистров (временное хранение обрабатываемой информации, представляет собой местную сверхбыстродействующую память), интерфейсная система микропроцессора (реализует сопряжение и связь с другими устройствами ПК), генератор тактовых импульсов (генерирует последовательность тактовых импульсов)

Основные характеристики: быстродействие (среднее число операций в секунду), тактовая частота (во многом определяет скорость работы компьютера, т.к. каждая операция выполняется за определенное кол-во тактов), разрядность (макс кол-во бит информации, которые могут обрабатываться и передаваться процессором одновременно)

Процессор состоит из нескольких частей. Блок управления отвечает за вызов команд из памяти и определение их типа. Арифметико-логическое устройство выполняет арифметические и логические операции (например, логическое И).

Внутри центрального процессора находится быстрая память небольшого объема для хранения промежуточных результатов и некоторых команд управления. Эта память состоит из нескольких регистров, каждый из которых выполняет определенную функцию. Обычно размер всех регистров одинаков. Каждый регистр содержит одно число в диапазоне, верхняя граница которого зависит от размера регистра. Операции чтения и записи с регистрами выполняются очень быстро, поскольку они находятся внутри центрального процессора.

Самый важный регистр — счетчик команд, который указывает, какую команду нужно выполнять следующей. Название «счетчик команд» выбрано неудачно, поскольку он ничего не считает, но этот термин употребляется повсеместно. Еще есть регистр команд, в котором находится выполняемая в данный момент команда. У большинства компьютеров имеются и другие регистры, одни из них многофункциональны, другие служат лишь какие-либо конкретным целям. Третьи регистры используются операционной системой для управления компьютером.

Внутреннее устройство тракта данных типичного фон-неймановского процессора иллюстрирует рис. 2.2. Тракт данных состоит из регистров (обычно от 1 до 32), арифметико-логического устройства (АЛУ) и нескольких соединительных шин.

**Регистры данных** программист может использовать по своему усмотрению для временного хранения любых объектов (данных или адресов) и выполнения над ними требуемых операций.

**Регистр команд**. Выбранная из памяти команда поступает по шине данных в РК, после чего начинается цикл выполнения команды, первым действием которого является ее дешифрация, обеспечиваемая УУ. В отличие от других регистров, РК только принимает данные, а посылать их на шину не может. Число разрядов РК зависит от состава команд микропроцессора (3 и более).

**Регистр адреса**. Для того чтобы выбрать очередную команду из памяти, содержимое счетчика команд передается по шине в регистр адреса РА. Выход этого регистра образует шину адреса, по которой числовое значение последнего поступает в блок памяти. В течение цикла выборки команды регистры РА и СК имеют одинаковые значения. После декодирования команды СК получает приращение, но содержимое РА не меняется. Если при выполнении команды появится необходимость получить данные из памяти, то выполняемую команду необходимо поместить в РА- адрес ячейки, хранящий требуемые данные.

Содержимое регистров поступает во входные регистры АЛУ, которые на рис. 2.2 обозначены буквами A и B. В них находятся входные данные АЛУ, пока АЛУ производит вычисления. Тракт данных — важная составная часть всех компьютеров, и мы обсудим его очень подробно.

АЛУ выполняет сложение, вычитание и другие простые операции над входными данными и помещает результат в выходной регистр. Содержимое этого выходного регистра может записываться обратно в один из регистров или сохраняться в памяти, если это необходимо. Не во всех архитектурах есть регистры A, B и выходные регистры. На рис. 2.2 представлена операция сложения, но АЛУ может выполнять и другие операции.

Большинство команд можно разделить на две группы: команды типа регистр-память и типа регистр-регистр. Команды первого типа вызывают слова из памяти, помещают их в регистры, где они используются в качестве входных данных АЛУ. Словом может быть целое число. Другие команды этого типа помещают регистры обратно в память. Команды второго типа вызывают два операнда из регистров, помещают их во входные регистры АЛУ, выполняют над ними какую-нибудь арифметическую или логическую операцию и переносят результат обратно в один из регистров. Этот процесс называется **циклом тракта данных**. В какой-то степени он определяет, что может делать машина. Современные компьютеры оснащаются несколькими АЛУ, работающими параллельно и специализирующимися на разных функциях. Чем быстрее происходит цикл тракта данных, тем быстрее компьютер работает.

Центральный процессор выполняет каждую команду за несколько шагов. Он делает следующее:

1. Вызывает следующую команду из памяти и переносит ее в регистр команд.

2. Меняет положение счетчика команд, который после этого указывает на следующую команду.

3. Определяет тип вызванной команды.

4. Если команда использует слово из памяти, определяет, где находится это слово.

5. Переносит слово, если это необходимо, в регистр центрального процессора.

6. Выполняет команду.

7. Переходит к шагу 1, чтобы начать выполнение следующей команды.

Такая последовательность шагов (**выборка — декодирование — исполнение)** является основой работы всех компьютеров.

**УУ** служит для управления работой всех узлов МП. **УУ**вырабатывает и передает другим компонентам МП и узлам ПК**управляющие и** **синхронизирующие сигналы**. Система синхронизации ПК базируется на кварцевом тактовом генераторе (ТГ). При включении ПК кварцевый генератор, имеющий строго определенные размеры, начинает вибрировать с постоянной частотой, достигающей до 3000Мгц и более. Каждое колебание ТГ генерирует **импульс напряжения**. Эти регулярно повторяющиеся импульсы совместно с другими сигналами управляющих схем задают темп работы всех компонент системной платы и обеспечивают синхронное срабатывание различных ее электронных элементов. Для передачи синхро и управляющих сигналов служит специальная шина управления (ШУ). Итак, блок УУ вырабатывает последовательность управляющих сигналов, запускающих выполнение соответствующей последовательности микрокоманд (МК). Эти микрокоманды реализуют текущую команду. Кроме этого УУ координирует работу всех устройств ЭВМ, посредством посылки управляющих сигналов, при этом обеспечивается обмен данными между ЦП и ОП, хранение и обработка информации, интерфейс с пользователем, тестирование, диагностика и т. д. Поэтому УУ можно рассматривать как отдельный блок ЦП. Однако на практике большинство управляющих схем распределены по всей ЭВМ. Они связаны между собой большим числом управляющих линий, по которым передаются сигналы для синхронизации операций во всех устройствах ЭВМ и принимаются сигналы о состоянии устройств ЭВМ.

**АЛУ предназначено для выполнения**арифметических и логических **операций над данными и адресами памяти,** которые хранятся в регистрах общего назначения (РОНах) и специальных регистрах. Регистры представляют собой сверх оперативную память, работающую со скоростью работы процессора. **Вся обрабатываемая и передаваемая информация** **передается по шине данных (ШД). Эта шина связывает все основные узлы** **системной платы, кроме ТГ**. АЛУ работает под управлением УУ. АЛУ выполняет арифметические операции над бинарными числами с фиксированной и плавающей точкой, над десятичными числами, производит обработку символьной информации над словами фиксированной и переменной длины и др. Логические операции производятся над отдельными битами, байтами и их последовательностями. **Тип выполняемой АЛУ операции определяется текущей командой,** **работающей в данный момент программы**. АЛУ служит для выполнения любой операции, которую задает УУ. В общем случае обрабатываемая информация состоит из слов, содержащих фиксированное число битов (например, 8, 16, 32, 64, 128 бит). В этом случае АЛУ должно иметь возможность производить операции над n – битными словами. Данные, которые в этом случае называются операнды, поступают из ОП на регистры АЛУ, а УУ указывает операции, которую надо произвести над операндами. **Результат каждой арифметическо-логической операции сохраняется в** **специальном регистре, который называется сумматором** и является основным регистром для арифметическо-логических операций. ЭВМ имеет несколько сумматоров. Если сумматоров больше четырех, они выделяются в специальную группу регистров общего назначения (РОНов).

Микропрограммная организация  
Основным компонентом любого компьютера является тракт данных. Он содержит несколько регистров, одну, две или три шины, а также один или несколько функциональных блоков, например АЛУ и схему сдвига. В основном цикле вызываются несколько операндов из регистров и передаются по шинам к АЛУ и другому функциональному блоку на исполнение. После завершения операции результаты вновь сохраняются в регистрах.

Тракт данных может управляться контроллером последовательности, который вызывает микрокоманды из управляющей памяти. Каждая микрокоманда содержит биты, управляющие трактом данных в течение одного цикла. Эти биты определяют, какие операнды нужно выбрать, какую операцию выполнить и что делать с результатами. Кроме того, каждая микрокоманда указывает на следующую микрокоманду (обычно в ней содержится ее адрес). Некоторые микрокоманды изменяют этот базовый адрес с помощью операции ИЛИ.

Существуют множество подходов к организации уровня микроархитектуры: в том числе 2- или 3-шинная структура, кодированные или декодированные поля микрокоманд, наличие или отсутствие предварительной выборки, конвейер с большим или небольшим количеством ступеней и т. д.

6. Понятие о различных архитектурах процессоров (CISC). Примеры

CISC  — тип [процессорной архитектуры](https://ru.wikipedia.org/wiki/%D0%9F%D1%80%D0%BE%D1%86%D0%B5%D1%81%D1%81%D0%BE%D1%80%D0%BD%D0%B0%D1%8F_%D0%B0%D1%80%D1%85%D0%B8%D1%82%D0%B5%D0%BA%D1%82%D1%83%D1%80%D0%B0), которая характеризуется следующим набором свойств:

* нефиксированное значение длины команды;
* арифметические действия кодируются в одной команде;
* небольшое число регистров, каждый из которых выполняет строго определённую функцию.

Методика построения системы команд CISC противостоит методике, применяемой в другом распространённом типе процессорных архитектур — [RISC](https://ru.wikipedia.org/wiki/RISC), где используется упрощённый набор инструкций.

Типичными представителями CISC-архитектуры являются процессоры на основе команд [x86](https://ru.wikipedia.org/wiki/X86), процессоры [Motorola](https://ru.wikipedia.org/wiki/Motorola" \o "Motorola) [MC680x0](https://ru.wikipedia.org/wiki/M68k), процессоры [мейнфреймов](https://ru.wikipedia.org/wiki/%D0%9C%D0%B5%D0%B9%D0%BD%D1%84%D1%80%D0%B5%D0%B9%D0%BC" \o "Мейнфрейм) [zSeries](https://ru.wikipedia.org/wiki/ZSeries" \o "ZSeries). Благодаря распространённости процессоров архитектур [x86](https://ru.wikipedia.org/wiki/X86) и [x86-64](https://ru.wikipedia.org/wiki/X86-64), CISC-системы являются самыми распространённым в мире вычислительной техники.

Основной недостаток CISC-архитектуры в сравнении с RISC — более сложный подход к распараллеливанию вычислений.

CPU первого типа являются традиционными, а их система команд включает большое количество команд для выполнения арифметических и логических операций, команд управления, пересылки и ввода-вывода данных. При считывании из операнда кода операции процессор обращается в ПЗУ микрокоманд и получает набор микроинструкций, реализующий алгоритм выполнения данной команды.

Такие CPU способны реализовывать любой алгоритм, который предварительно кодируется в системе команд данного CPU. Большинство универсальных CPU аппаратно поддерживают только целочисленную арифметику. Арифметика же с плавающей точкой реализуется программно во внутренних сопроцессорах. Сопроцессор расширяет набор команд ЭВМ. Когда основной процессор получает команду, которая не входит в его рабочий набор, он передает управление сопроцессору с целью ее выполнения.

Поначалу совершенствование процессоров было направлено на то, чтобы сконструировать по возможности более функциональный компьютер, который позволил бы выполнять как можно больше разных инструкций. Во-первых, так было удобнее для программистов (компиляторы языков высокого уровня еще только начинали развиваться, и все по-настоящему важные программы писались на ассемблере), а во-вторых, использование сложных инструкций зачастую позволяло сильно сократить размеры написанной на ассемблере программы. А где меньше инструкций – меньше и затраченное на исполнение программы время.

Подобные инструкции оказалось сложно не только выполнять, но и просто декодировать (выделять из машинного кода новую инструкцию и отправлять ее на исполнительные устройства). Чтобы машинный код CISC-компьютеров из-за сложных инструкций не разрастался до огромного размера, машинные инструкции в большинстве этих архитектур имели неоднородную структуру (разное расположение и размеры кода операции и ее операндов) и сильно отличающуюся длину (в x86, например, длина инструкций варьируется от 1 до 15 байт).

7. Понятие о различных архитектурах процессоров (RISC). Примеры

В то время когда разрабатывались эти простые процессоры, всеобщее внимание привлекало относительно небольшое количество команд (обычно около 50). Компьютер RISC (Reduced Instruction Set Computer — компьютер с сокращенным набором команд) противопоставлялся системе CISC (Complex Instruction Set Computer — компьютер с полным набором команд). На сегодняшний день мало кто считает, что размер набора команд так уж важен, но

названия сохранились до сих пор.

С этого момента началась грандиозная идеологическая война между сторонниками RISC и «консерваторами» (VAX, Intel, мэйнфреймы IBM). По мнению первых, наилучший способ разработки компьютеров — включение туда небольшого количества простых команд, каждая из которых выполняется за один цикл тракта данных, то есть производит над парой регистров какую-либо арифметическую или логическую операцию (например, сложение или операцию

логического И) и помещает результат обратно в регистр. В качестве аргумента они утверждали, что даже если системе RISC приходится выполнять 4 или 5 команд вместо одной, которую выполняет CISC, RISC все равно выигрывает в скорости, так как RISC-команды выполняются в 10 раз быстрее (поскольку они не интерпретируются). Следует также отметить, что к этому времени быстродействие основной памяти приблизилась к быстродействию специальных командных ПЗУ, потому недостатки интерпретации были налицо, что еще более поднимало популярность компьютеров RISC.

Учитывая преимущества RISC в плане производительности, можно было предположить, что на рынке такие компьютеры, как UltraSPARC компании Sun, должны доминировать над компьютерами CISC (Intel Pentium и т. д.). Однако ничего подобного не произошло. Почему?

Во-первых, компьютеры RISC несовместимы с другими моделями, а многие компании вложили миллиарды долларов в программное обеспечение для продукции Intel. Во-вторых, как ни странно, компания Intel сумела воплотить те же идеи в архитектуре CISC. Процессоры Intel, начиная с процессора 486, содержат RISC-ядро, которое выполняет самые простые (и обычно самые распространенные) команды за один цикл тракта данных, а по обычной технологии CISC

интерпретируются более сложные команды. В результате обычные команды выполняются быстро, а более сложные и редкие — медленно. Хотя при таком «гибридном» подходе производительность ниже, чем в архитектуре RISC, новая архитектура CISC имеет ряд преимуществ, поскольку позволяет использовать старое программное обеспечение без изменений.

Основные особенности RISC-процессоров:

1. Сокращенный набор команд (от 80 до 150 команд).
2. Большинство команд выполняется за 1 такт.
3. Большое количество регистров общего назначения.
4. Наличие жестких многоступенчатых конвейеров.
5. Все команды имеют простой формат, и используются немногие способы адресации.
6. Наличие вместительной раздельной кэш-памяти.
7. Применение оптимизирующих компиляторов, которые анализируют исходный код и частично меняют порядок следования команд.

8. Понятие о различных архитектурах процессоров (VLIW). Примеры

VLIW — архитектура [процессоров](https://ru.wikipedia.org/wiki/%D0%A6%D0%B5%D0%BD%D1%82%D1%80%D0%B0%D0%BB%D1%8C%D0%BD%D1%8B%D0%B9_%D0%BF%D1%80%D0%BE%D1%86%D0%B5%D1%81%D1%81%D0%BE%D1%80) с несколькими [вычислительными устройствами](https://ru.wikipedia.org/wiki/%D0%90%D1%80%D0%B8%D1%84%D0%BC%D0%B5%D1%82%D0%B8%D1%87%D0%B5%D1%81%D0%BA%D0%BE-%D0%BB%D0%BE%D0%B3%D0%B8%D1%87%D0%B5%D1%81%D0%BA%D0%BE%D0%B5_%D1%83%D1%81%D1%82%D1%80%D0%BE%D0%B9%D1%81%D1%82%D0%B2%D0%BE). Характеризуется тем, что одна инструкция процессора содержит несколько операций, которые должны выполняться параллельно. Фактически это «видимое программисту» микропрограммное управление, когда машинный код представляет собой лишь немного свёрнутый [микрокод](https://ru.wikipedia.org/wiki/%D0%9C%D0%B8%D0%BA%D1%80%D0%BE%D0%BA%D0%BE%D0%B4) для непосредственного управления аппаратурой.

В процессорах VLIW задача распределения работы решается во время [компиляции](https://ru.wikipedia.org/wiki/%D0%9A%D0%BE%D0%BC%D0%BF%D0%B8%D0%BB%D1%8F%D1%82%D0%BE%D1%80) и в инструкциях явно указано, какое вычислительное устройство какую команду должно выполнять.

Процессоры с множественной выдачей инструкций (multiple-issue processors) ориентированы на исполнение нескольких инструкций за такт и бывают двух видов: суперскалярные процессоры (superscalar processors) и процессоры с архитектурой VLIW (Very Long Instruction Word).

VLIW процессоры исполняют фиксированное число независимых инструкций, сгруппированных в одну длинную инструкцию. В таком пакете инструкций параллелизм уровня инструкций обеспечивается статически на этапе компиляции. Компания Intel, например, именует такую методику явного распараллеливания инструкций – EPIC – Explicitly Parallel Instruction Computing.

Суть явного параллелизма заключается в том, что распределение команд между исполнительными узлами производится не процессором в ходе выполнения программы (динамически), а компилятором при формировании машинного кода (статически).

Таким образом, схемотехника VLIW процессора существенно упрощается (по сложности он сравним с суперскалярным процессором без поддержки неупорядоченного исполнения команд). Кроме того, в компиляторах алгоритмы выбора порядка исполнения команд могут быть существенно сложнее и эффективнее, чем алгоритмы аппаратного планирования инструкций (так как решение необходимо принимать в течение наносекунд).

Подход VLIW сильно упрощает архитектуру процессора, перекладывая задачу распределения вычислительных устройств на [компилятор](https://ru.wikipedia.org/wiki/%D0%9A%D0%BE%D0%BC%D0%BF%D0%B8%D0%BB%D1%8F%D1%82%D0%BE%D1%80). Поскольку отсутствуют большие и сложные узлы, сильно снижается энергопотребление. В то же время код для VLIW обладает невысокой плотностью. Из-за большого количества пустых инструкций для простаивающих устройств программы для VLIW-процессоров могут быть гораздо длиннее, чем аналогичные программы для традиционных архитектур. Архитектура VLIW выглядит довольно экзотической и непривычной для программиста. Из-за сложных внутренних зависимостей кода программирование вручную, на уровне машинных кодов для VLIW-архитектур, является достаточно сложным. Приходится полагаться на оптимизацию компилятора.

9. Понятие о различных архитектурах процессоров (векторно-конвейерная). Примеры

Первый векторно-конвейерный компьютер Cray-1 появился в 1976 году. Архитектура его оказалась настолько удачной, что он положил начало целому семейству компьютеров. Название этому семейству компьютеров дали два принципа, заложенные в архитектуре процессоров:

* конвейерная организация обработки потока команд
* введение в систему команд набора векторных операций, которые позволяют оперировать с целыми массивами данных [[2]](http://rsusu1.rnd.runnet.ru/tutor/method/m1/liter1.html#[2]).

Длина одновременно обрабатываемых векторов в современных векторных компьютерах составляет, как правило, 128 или 256 элементов. Очевидно, что векторные процессоры должны иметь гораздо более сложную структуру и по сути дела содержать множество арифметических устройств. Основное назначение векторных операций состоит в распараллеливании выполнения операторов цикла, в которых в основном и сосредоточена большая часть вычислительной работы. Для этого циклы подвергаются процедуре векторизации с тем, чтобы они могли реализовываться с использованием векторных команд. Как правило, это выполняется автоматически компиляторами при изготовлении ими исполнимого кода программы. Поэтому векторно-конвейерные компьютеры не требовали какой-то специальной технологии программирования, что и явилось решающим фактором в их успехе на компьютерном рынке. Тем не менее, требовалось соблюдение некоторых правил при написании циклов с тем, чтобы компилятор мог их эффективно векторизовать.

Векторно-конвейерная структура компьютера содержит конвейер операций, на котором обрабатываются параллельно элементы векторов и полученные результаты последовательно записываются в единую память. При этом отпадает необходимость в коммутаторе процессорных элементов, служащем камнем преткновения в матричных компьютерах.

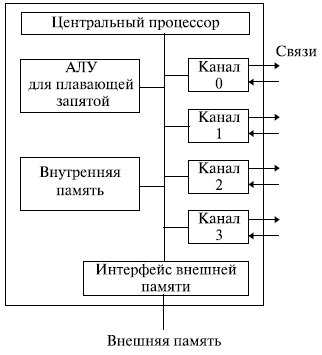
Векторно-конвейерную структуру имеют однопроцессорные супер-ЭВМ серии VP фирмы Fujitsu; серии S компании Hitachi; C90, М90, Т90 фирмы Cray Research; Сгау-3, Сгау-4 фирмы Cray Computer и т.д. Общим для всех векторных суперкомпьютеров является наличие в системе команд векторных операций, допускающих работу с векторами определенной длины, допустим, 64 элемента по 8 байт. В таких компьютерах операции с векторами обычно выполняются над векторными регистрами.

Векторный процессор — это [процессор](https://ru.wikipedia.org/wiki/%D0%9F%D1%80%D0%BE%D1%86%D0%B5%D1%81%D1%81%D0%BE%D1%80), в котором [операндами](https://ru.wikipedia.org/wiki/%D0%9E%D0%BF%D0%B5%D1%80%D0%B0%D0%BD%D0%B4) некоторых команд могут выступать упорядоченные массивы данных — [векторы](https://ru.wikipedia.org/wiki/%D0%92%D0%B5%D0%BA%D1%82%D0%BE%D1%80_(%D0%BC%D0%B0%D1%82%D0%B5%D0%BC%D0%B0%D1%82%D0%B8%D0%BA%D0%B0)). Отличается от [скалярных процессоров](https://ru.wikipedia.org/wiki/%D0%A1%D0%BA%D0%B0%D0%BB%D1%8F%D1%80%D0%BD%D1%8B%D0%B9_%D0%BF%D1%80%D0%BE%D1%86%D0%B5%D1%81%D1%81%D0%BE%D1%80), которые могут работать только с одним операндом в единицу времени. Абсолютное большинство [процессоров](https://ru.wikipedia.org/wiki/%D0%9F%D1%80%D0%BE%D1%86%D0%B5%D1%81%D1%81%D0%BE%D1%80) являются скалярными или близкими к ним. Векторные процессоры были распространены в сфере научных вычислений, где они являлись основой большинства [суперкомпьютеров](https://ru.wikipedia.org/wiki/%D0%A1%D1%83%D0%BF%D0%B5%D1%80%D0%BA%D0%BE%D0%BC%D0%BF%D1%8C%D1%8E%D1%82%D0%B5%D1%80) начиная с 1980-х до 1990-х. Но резкое увеличение производительности и активная разработка новых процессоров привели к вытеснению векторных процессоров из сферы повседневных [процессоров](https://ru.wikipedia.org/wiki/%D0%9F%D1%80%D0%BE%D1%86%D0%B5%D1%81%D1%81%D0%BE%D1%80).

В большинстве современных микропроцессоров имеются векторные расширения (см. [SSE](https://ru.wikipedia.org/wiki/SSE)). Кроме того, современные [видеокарты](https://ru.wikipedia.org/wiki/%D0%92%D0%B8%D0%B4%D0%B5%D0%BE%D0%BA%D0%B0%D1%80%D1%82%D0%B0) и физические ускорители можно рассматривать как векторные [сопроцессоры](https://ru.wikipedia.org/wiki/%D0%A1%D0%BE%D0%BF%D1%80%D0%BE%D1%86%D0%B5%D1%81%D1%81%D0%BE%D1%80).

10. Понятие о различных архитектурах процессоров (транспьютеры). Примеры

Транспьютер - это микропроцессор со встроенными средствами межпроцессорной коммуникации, предназначенной для построения многопроцессорных систем. Его название происходит от слов TRansfer (передатчик) и computer (вычислитель).

Транспьютер включает в себя средства для выполнения вычислений (ЦП, АЛУ с плавающей точкой, внутрикристальную память) и 4 канала для связи (линка) с другими транспьютерами и/или другими устройствами. Каждый линк представляет собой 2 однонаправленных последовательных канала передачи информации. Встроенныйинтерфейс позволяет подключать внешнюю память емкостью до 4 Гбайт

Использование такого подхода позволило организовать виртуальные каналы связи между процессами, которые могли размещаться как на единственном транспьютере, так и на нескольких транспьютерах, и виртуальные линки между процессами. Любой транспьютер может одновременно образовывать любое число параллельных процессов. Он имеет специальный планировщик, который производит распределение процессорного времени между этими процессами. Тем самым появляется возможность, имея всего лишь один транспьютер, написать параллельную программу, которая полностью выполняется на нем. Задача разбивается на ряд процессов, и все эти процессы параллельно протекают внутри одного транспьютера, периодически останавливаясь для получения данных друг от друга. Систему можно расширить другими транспьютерами и перенести на них ряд процессов. При этом нужно просто переопределить таблицу связей процессов, указав, на каком транспьютере теперь выполняется тот или иной процесс. Сама же программа изменений не претерпевает, а вычислительная мощность системы, естественно, увеличивается.

11. Понятие о различных архитектурах процессоров (DSP). Примеры

Характерные особенности ЦСП (DSP):

– ОЗУ, ПЗУ;

– последовательный/параллельный интерфейсы;

– схема обработки прерываний;

– ЦПУ оптимизировано для многократно повторяющихся математических операций в реальном масштабе времени.

Примеры: TMS320Cхххх, ADSP-21XX

Процессор цифровой обработки сигналов (ЦСП или DSP – digital signal processor) представляет из себя специализированный программируемый микропроцессор, предназначенный для манипулирования в реальном масштабе времени потоком цифровых данных. DSP-процессоры широко используются для обработки потоков графической информации, аудио- и видеосигналов.

DSP-процессор должен быть «узким специалистом». Его единственная задача — изменять поток цифровых сигналов, и делать это быстро. DSP-процессор состоит главным образом из высокоскоростных аппаратных схем, выполняющих арифметические функции и манипулирующих битами, оптимизированных для того, чтобы быстро изменять большие объемы данных.

В силу этого набор команд у DSP куда меньше, чем у центрального процес- сора настольного компьютера; их число не превышает 80. Это значит, что для DSP требуется облегченный декодер команд и гораздо меньшее число исполнительных устройств. Кроме того, все исполнительные устройства в конечном итоге должны поддерживать высокопроизводительные арифметические операции. Таким образом, типичный DSP-процессор состоит не более чем из нескольких сот тысяч транзисторов.

Используя алгоритмы, основанные на соответствующем математическом аппарате, DSP-процессор может воспринимать цифровой сигнал и выполнять операции свертки для усиления или подавления тех или иных свойств сигнала.

В силу того, что в DSP-процессорах значительно меньше транзисторов, чем в центральных процессорах, они потребляют меньше энергии, что позволяет использовать их в продуктах, работающих от батарей. Крайне упрощается и их производство, поэтому они находят себе применение в недорогих устройствах. Сочетание низкого энергопотребления и невысокая стоимость обусловливает применение DSP-процессоров в сотовых телефонах и в роботах-игрушках.

12. Выполнение программы в ЭВМ. Схема взаимодействия процессора (АЛУ, УУ) и памяти.

До момента, когда пользователь компьютера явно или неявно выдаст запрос на выполнение компьютерной программы, она обычно хранится в энергонезависимой памяти. При получении такого запроса программа посредством другой компьютерной программы, называющейся операционной системой, загружается в память с произвольным доступом, откуда её непосредственно может выполнять центральный процессор. После этого центральный процессор выполняет программу, инструкция за инструкцией, до её завершения. Выполняющаяся программа называется процессом. Завершение программы происходит либо по достижению её последней инструкции (обычно передающей управление операционной системе) либо по ошибке, программной или аппаратной.

Центральный процессор выполняет каждую команду за несколько шагов. Он делает следующее:

1. Вызывает следующую команду из памяти и переносит ее в регистр команд.

2. Меняет положение счетчика команд, который после этого указывает на следующую команду.

3. Определяет тип вызванной команды.

4. Если команда использует слово из памяти, определяет, где находится это слово.

5. Переносит слово, если это необходимо, в регистр центрального процессора.

6. Выполняет команду.

7. Переходит к шагу 1, чтобы начать выполнение следующей команды.

Такая последовательность шагов (**выборка — декодирование — исполнение)** является основой работы всех компьютеров.



АЛУ осуществляет обработку данных. Типичными операциями, выполняемыми АЛУ, являются сложение, вычитание, лог. сложение (ИЛИ), лог умножение (И), сложение по модулю (ИСКЛЮЧАЮЩЕЕ ИЛИ), инверсия, сдвиг, пересылка. Обычно АЛУ имеет два входа, которые называются входными портами, и один выход, или выходной порт. Данные на входные порты АЛУ поступают в внутренней шины данных из специального регистра, называемого аккумулятором, через буферные регистры, или регистры операндов, предназначенные для временного хранения данных.   
Работой АЛУ и внутренними регистрами управляет УУ, которое извлекает из регистра команд очередную команду, дешифрует ее и обеспечивает выполнение этой операции в АЛУ.

Любая задача решается по программе, которая представляет собой строгую последовательность нужных команд. Эту последовательность поступления команд обеспечивает регистр, называемый счетчиком команд. Счетчик команд может иметь большее число разрядов, чем длина слова данных. В результате можно записать команду в любую ячейку памяти.

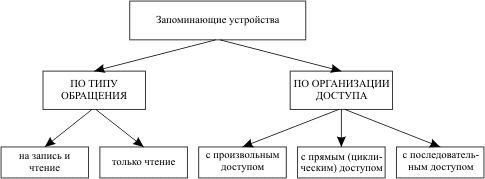
Перед выполнением программы в счетчик команд записывают число, которое определяет адрес первой программы, хранящееся в ЗУ. Затем это число из счетчика команд переписывается в 16-разрядный регистр адреса памяти. Из регистра адреса памяти по шине адреса (ША) адрес первой команды посылается в устройство управления памятью. По указанному адресу из ЗУ осуществляется считывание первой команды, которая переписывается в регистр команды. Рассмотренный цикл операций называют циклом выборки или фазой адресации.

После записи команды в регистр УУ осуществляется ее распознавание (декодирование) и в АЛУ поступают сигналы, стимулирующие выполнение данной команды. Цикл выборки совместно с циклом выполнения команды образуют цикл команд.

13. Запоминающее устройство. Основная память. Слово. Адрес. Бит. Байт. Примеры

**Памятью** ЭВМ называется совокупность устройств, служащих для запоминания, хранения и выдачи информации.

Отдельные устройства, входящие в эту совокупность, называются ***запоминающими устройствами*** ( *ЗУ* ) того или иного типа



По типу обращения *ЗУ* делятся на устройства, допускающие как чтение, так и запись информации, и постоянные *запоминающие устройства* (ПЗУ), предназначенные только для чтения записанных в них данных ( ***ROM*** - **read only memory** ). *ЗУ* первого типа используются в процессе работы процессора для хранения выполняемых программ, исходных данных, промежуточных и окончательных результатов. В *ПЗУ*, как правило, хранятся *системные программы*, необходимые для запуска компьютера в работу, а также *константы*. В некоторых ЭВМ, предназначенных, например, для работы в системах управления *по* одним и тем же неизменяемым алгоритмам, все *программное обеспечение* может храниться в *ПЗУ*.

В ***ЗУ с произвольным доступом*** ( ***RAM*** - **random access memory** ) время доступа не зависит от места расположения участка памяти (например, *ОЗУ* ).

В ***ЗУ с прямым (циклическим) доступом*** благодаря непрерывному вращению носителя информации (например, магнитный диск - МД) возможность обращения к некоторому участку носителя циклически повторяется. Время доступа здесь зависит от взаимного расположения этого участка и головок чтения/записи и во многом определяется скоростью вращения носителя.

В ***ЗУ с последовательным доступом*** производится последовательный просмотр участков носителя информации, пока нужный участок не займет некоторое нужное положение напротив головок чтения/записи (например, магнитные ленты - МЛ).

***Основная память*** - это запоминающее устройство, напрямую связанное с процессором и предназначенное для хранения выполняемых программ и данных непосредственно участвующих в операциях. Она имеет достаточное быстродействие, но ограниченный объем. Основная память делится на различные виды, основными из которых являются оперативная память (ОЗУ) и постоянное запоминающее устройство (ПЗУ) (рис.1).

         ОЗУ предназначено для хранения информации (программ и данных), непосредственно участвующей в вычислительном процессе на текущем этапе функционирования.

         ОЗУ служит для приема, хранения и выдачи информации. Именно в нем процессор «берет» программы и исходные данные для обработки, в нее он записывает полученные результаты. Название «оперативная» это память получила потому, что она работает очень быстро, так что процессору практически не приходится ждать при чтении данных из памяти и записи в память. Однако содержащие в ней данные сохраняются только пока компьютер включен. При выключении компьютера содержимое оперативной памяти стирается. Таким образом ОЗУ - энергозависимая память.



Часто для оперативной памяти используют обозначение RAM (random access memory, т.е. память с произвольным доступом). Под произвольным доступом понимают возможность непосредственного доступа к любой (произвольной) заданной ячейки памяти, причем время доступа для любой ячейки одинаково.

         Основу ОЗУ составляют большие интегральные схемы, содержащие матрицы полупроводниковых запоминающих элементов (триггеров). Запоминающие элементы расположены на пересечении вертикальных и горизонтальных шин матрицы; запись и считывание информации осуществляется подачей электрических импульсов по тем каналам матрицы, которые соединены с элементами, принадлежащими выбранной ячейке памяти.



**Машинное слово** — машиннозависимая и платформозависимая величина, [измеряемая](http://dic.academic.ru/dic.nsf/ruwiki/4910) в [битах](http://dic.academic.ru/dic.nsf/ruwiki/4904) или [байтах](http://dic.academic.ru/dic.nsf/ruwiki/4903)([тритах](http://dic.academic.ru/dic.nsf/ruwiki/448217) или [трайтах](http://dic.academic.ru/dic.nsf/ruwiki/1260335)), равная [разрядности](http://dic.academic.ru/dic.nsf/ruwiki/300216) регистров [процессора](http://dic.academic.ru/dic.nsf/ruwiki/1187537) и/или разрядности [шины данных](http://dic.academic.ru/dic.nsf/ruwiki/29590) (обычнонекоторая степень двойки).

Каждая ячейка памяти имеет адрес, который используется для ее нахождения. ***Адреса*** - это числа, начиная с нуля для первой ячейки, увеличивающиеся по направлению к последней ячейке памяти. Поскольку адреса - это те же числа, компьютер может использовать арифметические операции для вычисления адресов памяти.

**Бит** — это единица компьютерной информации. Дискретная. Может принимать ровно два значения: ноль и единица.

**Байт = 8 бит**. Другими словами, байт состоит из восьми ячеек, каждая из которых может принимать ровно два разных значения: нулик или единичку. По основам комбинаторики такая конструкция может принимать 28=256 значений. То есть, одним байтом можно закодировать 256 разных значений, от 0 до 255 включительно.

14. Запоминающее устройство. Кэш-память. Примеры

Кэш - промежуточный [буфер](https://ru.wikipedia.org/wiki/%D0%91%D1%83%D1%84%D0%B5%D1%80_(%D0%B8%D0%BD%D1%84%D0%BE%D1%80%D0%BC%D0%B0%D1%82%D0%B8%D0%BA%D0%B0)) с быстрым доступом, содержащий информацию, которая может быть запрошена с наибольшей вероятностью. Доступ к данным в кэше осуществляется быстрее, чем выборка исходных данных из более медленной памяти или удаленного источника, однако её объём существенно ограничен по сравнению с хранилищем исходных данных.

Кэш — это память с большей скоростью доступа, предназначенная для ускорения обращения к данным, содержащимся постоянно в памяти с меньшей скоростью доступа (далее «основная память»). Кэширование применяется [ЦПУ](https://ru.wikipedia.org/wiki/%D0%9F%D1%80%D0%BE%D1%86%D0%B5%D1%81%D1%81%D0%BE%D1%80), [жёсткими дисками](https://ru.wikipedia.org/wiki/%D0%96%D1%91%D1%81%D1%82%D0%BA%D0%B8%D0%B9_%D0%B4%D0%B8%D1%81%D0%BA), [браузерами](https://ru.wikipedia.org/wiki/%D0%91%D1%80%D0%B0%D1%83%D0%B7%D0%B5%D1%80), [веб-серверами](https://ru.wikipedia.org/wiki/%D0%92%D0%B5%D0%B1-%D1%81%D0%B5%D1%80%D0%B2%D0%B5%D1%80" \o "Веб-сервер), службами [DNS](https://ru.wikipedia.org/wiki/DNS) и [WINS](https://ru.wikipedia.org/wiki/WINS).

Когда клиент кэша (ЦПУ, веб-браузер, операционная система) обращается к данным, прежде всего исследуется кэш. Если в кэше найдена запись с идентификатором, совпадающим с идентификатором затребованного элемента данных, то используются элементы данных в кэше. Такой случай называется *попаданием кэша*. Если в кэше не найдена запись, содержащая затребованный элемент данных, то он читается из основной памяти в кэш, и становится доступным для последующих обращений. Такой случай называется *промахом кэша*. Процент обращений к кэшу, когда в нём найден результат, называется *уровнем попаданий*, или *коэффициентом попаданий* в кэш.

Многие периферийные устройства хранения данных используют внутренний кэш для ускорения работы, в частности, жёсткие диски используют кэш-память от 1 до 64 Мбайт, устройства чтения CD/DVD/BD-дисков также кэшируют прочитанную информацию для ускорения повторного обращения.

Операционная система также использует часть оперативной памяти в качестве кэша дисковых операций (например, для внешних устройств, не обладающих собственной кэш-памятью, в том числе жёстких дисков, flash-памяти и гибких дисков). Часто для кэширования жёстких дисков предоставляется вся свободная (не выделенная процессам) оперативная память.

15. Запоминающее устройство. Виртуальная память. Примеры

Суть концепции ***виртуальной памяти*** заключается в следующем. Информация, с которой работает активный процесс, должна располагаться в оперативной памяти. В схемах *виртуальной памяти* у процесса создается иллюзия того, что вся необходимая ему информация имеется в основной памяти. Для этого, во-первых, занимаемая процессом память разбивается на несколько частей, например страниц. Во-вторых, логический адрес (логическая страница), к которому обращается процесс, динамически транслируется в физический адрес (физическую страницу). И, наконец, в тех случаях, когда страница, к которой обращается процесс, не находится в физической памяти, нужно организовать ее подкачку с диска. Для контроля наличия страницы в памяти вводится специальный *бит присутствия*, входящий в состав атрибутов страницы в *таблице страниц* .

Таким образом, в наличии всех компонентов процесса в основной памяти необходимости нет. Важным следствием такой организации является то, что размер памяти, занимаемой процессом, может быть больше, чем размер оперативной памяти. *Принцип локальности*обеспечивает этой схеме нужную эффективность.

Но введение *виртуальной памяти* позволяет решать другую, не менее важную задачу – обеспечение контроля доступа к отдельным сегментам памяти и, в частности, защиту пользовательских программ друг от друга и защиту ОС от пользовательских программ. Каждый процесс работает со своими *виртуальными адресами*, трансляцию которых в физические выполняет аппаратура компьютера. Таким образом, *пользовательский процесс* лишен возможности напрямую обратиться к страницам основной памяти, занятым информацией, относящейся к другим процессам.

примеры

## Страничная организация виртуальной памяти

В большинстве современных операционных систем виртуальная память организуется с помощью страничной адресации. Оперативная память делится на страницы: области памяти фиксированной длины, которые являются минимальной единицей выделяемой памяти (то есть даже запрос на 1 байт от приложения приведёт к выделению ему страницы памяти). Исполняемый процессором пользовательский поток обращается к памяти с помощью адреса виртуальной памяти, который делится на номер страницы и смещение внутри страницы.

## Сегментная организация виртуальной памяти

Механизм организации виртуальной памяти, при котором виртуальное пространство делится на части произвольного размера — сегменты. Этот механизм позволяет, к примеру, разбить данные процесса на логические блоки.[[15]](https://ru.wikipedia.org/wiki/%D0%92%D0%B8%D1%80%D1%82%D1%83%D0%B0%D0%BB%D1%8C%D0%BD%D0%B0%D1%8F_%D0%BF%D0%B0%D0%BC%D1%8F%D1%82%D1%8C#cite_note-15) Для каждого сегмента, как и для страницы, могут быть назначены права доступа к нему пользователя и его процессов. При загрузке процесса часть сегментов помещается в оперативную память (при этом для каждого из этих сегментов операционная система подыскивает подходящий участок свободной памяти), а часть сегментов размещается в дисковой памяти.

16. Устройства ввода-вывода. Примеры

**Устройство [ввода-вы́вода](https://ru.wikipedia.org/wiki/%D0%92%D0%B2%D0%BE%D0%B4-%D0%B2%D1%8B%D0%B2%D0%BE%D0%B4" \o "Ввод-вывод)** — компонент типовой [архитектуры ЭВМ](https://ru.wikipedia.org/wiki/%D0%90%D1%80%D1%85%D0%B8%D1%82%D0%B5%D0%BA%D1%82%D1%83%D1%80%D0%B0_%D0%AD%D0%92%D0%9C), предоставляющий [компьютеру](https://ru.wikipedia.org/wiki/%D0%9A%D0%BE%D0%BC%D0%BF%D1%8C%D1%8E%D1%82%D0%B5%D1%80) возможность взаимодействия с внешним миром и, в частности, с [пользователями](https://ru.wikipedia.org/wiki/%D0%9F%D0%BE%D0%BB%D1%8C%D0%B7%D0%BE%D0%B2%D0%B0%D1%82%D0%B5%D0%BB%D1%8C).

Устройства ввода — это, в основном, [датчики](https://ru.wikipedia.org/wiki/%D0%94%D0%B0%D1%82%D1%87%D0%B8%D0%BA) преобразования неэлектрических величин (расположение в пространстве, [давление](https://ru.wikipedia.org/wiki/%D0%94%D0%B0%D0%B2%D0%BB%D0%B5%D0%BD%D0%B8%D0%B5), [вязкость](https://ru.wikipedia.org/wiki/%D0%92%D1%8F%D0%B7%D0%BA%D0%BE%D1%81%D1%82%D1%8C), [скорость](https://ru.wikipedia.org/wiki/%D0%A1%D0%BA%D0%BE%D1%80%D0%BE%D1%81%D1%82%D1%8C), [ускорение](https://ru.wikipedia.org/wiki/%D0%A3%D1%81%D0%BA%D0%BE%D1%80%D0%B5%D0%BD%D0%B8%D0%B5), [освещённость](https://ru.wikipedia.org/wiki/%D0%9E%D1%81%D0%B2%D0%B5%D1%89%D1%91%D0%BD%D0%BD%D0%BE%D1%81%D1%82%D1%8C), [температура](https://ru.wikipedia.org/wiki/%D0%A2%D0%B5%D0%BC%D0%BF%D0%B5%D1%80%D0%B0%D1%82%D1%83%D1%80%D0%B0), [влажность](https://ru.wikipedia.org/wiki/%D0%92%D0%BB%D0%B0%D0%B6%D0%BD%D0%BE%D1%81%D1%82%D1%8C), перемещение, количественные величины и т. п.) и электрических величин в электрические сигналы воспринимаемые процессором для дальнейшей их обработки в основном в цифровом виде.

* [Клавиатура](https://ru.wikipedia.org/wiki/%D0%9A%D0%BB%D0%B0%D0%B2%D0%B8%D0%B0%D1%82%D1%83%D1%80%D0%B0_%D0%BA%D0%BE%D0%BC%D0%BF%D1%8C%D1%8E%D1%82%D0%B5%D1%80%D0%B0)
* [Мышь](https://ru.wikipedia.org/wiki/%D0%9C%D1%8B%D1%88%D1%8C_(%D0%BA%D0%BE%D0%BC%D0%BF%D1%8C%D1%8E%D1%82%D0%B5%D1%80)) и [тачпад](https://ru.wikipedia.org/wiki/%D0%A2%D0%B0%D1%87%D0%BF%D0%B0%D0%B4" \o "Тачпад)
* [Планшет](https://ru.wikipedia.org/wiki/%D0%93%D1%80%D0%B0%D1%84%D0%B8%D1%87%D0%B5%D1%81%D0%BA%D0%B8%D0%B9_%D0%BF%D0%BB%D0%B0%D0%BD%D1%88%D0%B5%D1%82)
* [Джойстик](https://ru.wikipedia.org/wiki/%D0%94%D0%B6%D0%BE%D0%B9%D1%81%D1%82%D0%B8%D0%BA)
* [Сканер](https://ru.wikipedia.org/wiki/%D0%A1%D0%BA%D0%B0%D0%BD%D0%B5%D1%80_%D0%B8%D0%B7%D0%BE%D0%B1%D1%80%D0%B0%D0%B6%D0%B5%D0%BD%D0%B8%D0%B9)
* [Цифровые фото](https://ru.wikipedia.org/wiki/%D0%A6%D0%B8%D1%84%D1%80%D0%BE%D0%B2%D0%BE%D0%B9_%D1%84%D0%BE%D1%82%D0%BE%D0%B0%D0%BF%D0%BF%D0%B0%D1%80%D0%B0%D1%82), [видеокамеры](https://ru.wikipedia.org/wiki/%D0%92%D0%B8%D0%B4%D0%B5%D0%BE%D0%BA%D0%B0%D0%BC%D0%B5%D1%80%D0%B0), [веб-камеры](https://ru.wikipedia.org/wiki/%D0%92%D0%B5%D0%B1-%D0%BA%D0%B0%D0%BC%D0%B5%D1%80%D0%B0" \o "Веб-камера)
* [Микрофон](https://ru.wikipedia.org/wiki/%D0%9C%D0%B8%D0%BA%D1%80%D0%BE%D1%84%D0%BE%D0%BD)

Устройства вывода — это преобразователи электрической цифровой информации в вид необходимый для получения требуемого результата, могущего быть как неэлектрической природы (механические, тепловые, оптические, звуковые), так и электрической природы ([трансформаторы](https://ru.wikipedia.org/wiki/%D0%A2%D1%80%D0%B0%D0%BD%D1%81%D1%84%D0%BE%D1%80%D0%BC%D0%B0%D1%82%D0%BE%D1%80%D1%8B), нагреватели, [электродвигатели](https://ru.wikipedia.org/wiki/%D0%AD%D0%BB%D0%B5%D0%BA%D1%82%D1%80%D0%BE%D0%B4%D0%B2%D0%B8%D0%B3%D0%B0%D1%82%D0%B5%D0%BB%D0%B8" \o "Электродвигатели),[реле](https://ru.wikipedia.org/wiki/%D0%A0%D0%B5%D0%BB%D0%B5)).

* [Монитор](https://ru.wikipedia.org/wiki/%D0%9C%D0%BE%D0%BD%D0%B8%D1%82%D0%BE%D1%80_(%D1%83%D1%81%D1%82%D1%80%D0%BE%D0%B9%D1%81%D1%82%D0%B2%D0%BE))
* [Графопостроитель](https://ru.wikipedia.org/wiki/%D0%93%D1%80%D0%B0%D1%84%D0%BE%D0%BF%D0%BE%D1%81%D1%82%D1%80%D0%BE%D0%B8%D1%82%D0%B5%D0%BB%D1%8C)
* [Принтер](https://ru.wikipedia.org/wiki/%D0%9F%D1%80%D0%B8%D0%BD%D1%82%D0%B5%D1%80)
* [Акустическая система](https://ru.wikipedia.org/wiki/%D0%90%D0%BA%D1%83%D1%81%D1%82%D0%B8%D1%87%D0%B5%D1%81%D0%BA%D0%B0%D1%8F_%D1%81%D0%B8%D1%81%D1%82%D0%B5%D0%BC%D0%B0)

17. Устройства внешней памяти. Примеры

Внешняя память (ВЗУ) предназначена для длительного хранения программ и данных, и целостность её содержимого не зависит от того, включен или выключен компьютер.

Одной из основных характеристик ВЗУ является общий объем хранимой информации, или емкость ВЗУ, обычно измеряемая в байтах.

Из-за большого различия быстродействия оперативной памяти и ВЗУ обращения к внешней памяти вызывают потери производительности ПК. Поэтому быстродействие ВЗУ является показателем не менее важным, чем его емкость.

Соответственно быстродействие ВЗУ определяется двумя показателями: временем доступа и скоростью передачи информации. Различают среднее и максимальное время доступа.

В связи с определенными техническими особенностями магнитных носителей информации, на них нельзя записать и с них нельзя считать отдельный байт. Запись и считывание информации могут производиться только группами байт строго определенного размера блоками.

Основные характеристики ВЗУ прямо зависят от плотности записи информации на носитель.

Поверхностная плотность записи информации является произведением продольной плотности записи на поперечную плотность записи.

Продольная плотность записи равна числу бит, записываемых на единицу длины дорожки (бит/мм, бит/см или бит/дюйм).Поперечная плотность записи равна числу дорожек, приходящихся на единицу длины в направлении, перпендикулярном движению носителя (дорожек/мм, дорожек/см или дорожек/дюйм). Поверхностная плотность записи, таким образом, измеряется числом бит на квадратный миллиметр, квадратный сантиметр или квадратный дюйм.

Увеличения поперечной плотности записи можно достигнуть уменьшением ширины дорожки и расстояния между центрами дорожек. Минимальная ширина дорожки ограничена технологическими трудностями изготовления головок. При уменьшении расстояния между центрами дорожек увеличиваются перекрестные электромагнитные наводки в головках.

Для уменьшения зазора используются различные аэродинамические эффекты, создающие между головками и диском воздушную подушку толщиной в несколько микрометров.

В отличие от оперативной памяти, внешняя память не имеет прямой связи с процессором. Для ускорения обмена с дисками широко применяется кэширование, принцип которого близок к принципу кэширования оперативной памяти. Точно так же кэширование диска позволяет за счет использования более быстрой электронной памяти, чем дисковая память, существенно увеличить среднюю скорость обмена с диском.

Особенно эффективно кэширование при оптимизации жесткого диска (его дефрагментации), когда каждый файл расположен в группе секторов, следующих друг за другом. Кэш-память диска обычно располагается на плате специального кэш-контроллера дисковода, и ее объем может достигать 16 Мбайт.

В состав внешней памяти компьютера входят:

* накопители на жёстких магнитных дисках;
* накопители на гибких магнитных дисках;
* накопители на компакт-дисках;
* накопители на магнито-оптических компакт-дисках;
* накопители на магнитной ленте (стримеры) и др.

Накопители на гибких магнитных дисках

18. Устройства межмашинной связи. Примеры.

Межмашинное взаимодействие (машинно-машинное взаимодействие, [англ.](https://ru.wikipedia.org/wiki/%D0%90%D0%BD%D0%B3%D0%BB%D0%B8%D0%B9%D1%81%D0%BA%D0%B8%D0%B9_%D1%8F%D0%B7%D1%8B%D0%BA) Machine-to-Machine, M2M) — общее название технологий, которые позволяют машинам обмениваться информацией друг с другом, или же передавать её в одностороннем порядке. Это могут быть проводные и беспроводные системы мониторинга датчиков или каких-либо параметров устройств ([температура](https://ru.wikipedia.org/wiki/%D0%A2%D0%B5%D0%BC%D0%BF%D0%B5%D1%80%D0%B0%D1%82%D1%83%D1%80%D0%B0), уровень запасов, местоположение и т. д.). К примеру, банкоматы или платёжные терминалы могут автоматически передавать информацию по [GSM](https://ru.wikipedia.org/wiki/GSM)-сетям, а также если у них закончилась чековая бумага или наличность, или же наоборот, что наличности слишком много и требуется приезд инкассаторов. M2M также активно используется в системах безопасности и охраны, вендинге, системах здравоохранения, промышленных телеметрических системах (производство, энергетика, ЖКХ и др.) и системах позиционирования подвижных объектов на основе систем [ГЛОНАСС](https://ru.wikipedia.org/wiki/%D0%93%D0%9B%D0%9E%D0%9D%D0%90%D0%A1%D0%A1)/[GPS](https://ru.wikipedia.org/wiki/GPS)[[1]](https://ru.wikipedia.org/wiki/%D0%9C%D0%B5%D0%B6%D0%BC%D0%B0%D1%88%D0%B8%D0%BD%D0%BD%D0%BE%D0%B5_%D0%B2%D0%B7%D0%B0%D0%B8%D0%BC%D0%BE%D0%B4%D0%B5%D0%B9%D1%81%D1%82%D0%B2%D0%B8%D0%B5#cite_note-1). Одним из подклассов M2M является межмашинное взаимодействие с использованием мобильных решений, для него также может использоваться аббревиатура M2M ([англ.](https://ru.wikipedia.org/wiki/%D0%90%D0%BD%D0%B3%D0%BB%D0%B8%D0%B9%D1%81%D0%BA%D0%B8%D0%B9_%D1%8F%D0%B7%D1%8B%D0%BA) Mobile-to-Mobile).

4G-роутеры промышленного типа используются во многих М2М-приложениях:

* телеметрия;
* удаленное управление производством, инженерными сетями;
* видеонаблюдение;
* системы энергоучета;
* платежные терминалы, автоматы, банкоматы;
* контроль перевозок.

**Межмашинная связь на уровне внешних устройств** используется главным образом для организации общего поля внешней памяти. Такая связь организуется через каналы ввода-вывода этих устройств и шинные интерфейсы. При этом обычно часть ВЗУ остается в индивидуальном пользовании отдельных машин. Преимуществом комплексирования на уровне ВЗУ является значительное увеличение объемов информации (данных и программ), одновременно доступных процессорам ММВС.

Наибольшее распространение получила **взаимодействие вычислительных средств на уровне канал-канал** через адаптер канал-канал. Адаптер подключается к двум каналам, причем функционально он рассматривается как устройство управления ввода-вывода для каждого из каналов, а каждая из связанных адаптером машин по отношению друг к другу является внешним устройством. В отличие от любого другого устройства управления внешними устройствами адаптер не управляет устройствами ввода-вывода, а только осуществляет связь между каналами и синхронизирует их работу. Адаптер обеспечивает быстрый обмен информацией между каналами, а следовательно, и между ОП взаимодействующих процессоров, если общее поле оперативной памяти не организовано.

примеры: спутниковая связь, локальный шнур, лазерная связь

19. Терминальные устройства: текстовые и Х-терминалы, сетевые компьютеры, нулевые клиенты, сравнение с ПЭВМ. Примеры

**Компью́терный термина́л**, **оконе́чное устро́йство** — устройство, используемое для взаимодействия пользователя (или оператора) с [компьютером](https://ru.wikipedia.org/wiki/%D0%9A%D0%BE%D0%BC%D0%BF%D1%8C%D1%8E%D1%82%D0%B5%D1%80) или компьютерной системой, локальной или удалённой[[1]](https://ru.wikipedia.org/wiki/%D0%9A%D0%BE%D0%BC%D0%BF%D1%8C%D1%8E%D1%82%D0%B5%D1%80%D0%BD%D1%8B%D0%B9_%D1%82%D0%B5%D1%80%D0%BC%D0%B8%D0%BD%D0%B0%D0%BB#cite_note-.D0.92.D0.BE.D1.80.D0.BE.D0.B9.D1.81.D0.BA.D0.B8.D0.B9.E2.80.942003.E2.80.94.E2.80.94132-1)[[2]](https://ru.wikipedia.org/wiki/%D0%9A%D0%BE%D0%BC%D0%BF%D1%8C%D1%8E%D1%82%D0%B5%D1%80%D0%BD%D1%8B%D0%B9_%D1%82%D0%B5%D1%80%D0%BC%D0%B8%D0%BD%D0%B0%D0%BB#cite_note-2). Выводимая терминалом информация может быть как [текстовой](https://ru.wikipedia.org/wiki/%D0%A2%D0%B5%D0%BA%D1%81%D1%82%D0%BE%D0%B2%D0%B0%D1%8F_%D0%B8%D0%BD%D1%84%D0%BE%D1%80%D0%BC%D0%B0%D1%86%D0%B8%D1%8F), так и [графической](https://ru.wikipedia.org/wiki/%D0%93%D1%80%D0%B0%D1%84%D0%B8%D1%87%D0%B5%D1%81%D0%BA%D0%B0%D1%8F_%D0%B8%D0%BD%D1%84%D0%BE%D1%80%D0%BC%D0%B0%D1%86%D0%B8%D1%8F). Терминал с возможностями обработки данных и памятью назывался ***интеллектуальным*** ([англ.](https://ru.wikipedia.org/wiki/%D0%90%D0%BD%D0%B3%D0%BB%D0%B8%D0%B9%D1%81%D0%BA%D0%B8%D0%B9_%D1%8F%D0%B7%D1%8B%D0%BA) *intelligent terminal*)[[3]](https://ru.wikipedia.org/wiki/%D0%9A%D0%BE%D0%BC%D0%BF%D1%8C%D1%8E%D1%82%D0%B5%D1%80%D0%BD%D1%8B%D0%B9_%D1%82%D0%B5%D1%80%D0%BC%D0%B8%D0%BD%D0%B0%D0%BB#cite_note-3).

К одному компьютеру может быть подключено несколько терминалов. Такую вычислительную систему называют ***многотерминальной***[[4]](https://ru.wikipedia.org/wiki/%D0%9A%D0%BE%D0%BC%D0%BF%D1%8C%D1%8E%D1%82%D0%B5%D1%80%D0%BD%D1%8B%D0%B9_%D1%82%D0%B5%D1%80%D0%BC%D0%B8%D0%BD%D0%B0%D0%BB#cite_note-.D0.92.D0.BE.D1.80.D0.BE.D0.B9.D1.81.D0.BA.D0.B8.D0.B9.E2.80.942003.E2.80.94.E2.80.94127-4).

Терминал — это [устройство ввода-вывода](https://ru.wikipedia.org/wiki/%D0%A3%D1%81%D1%82%D1%80%D0%BE%D0%B9%D1%81%D1%82%D0%B2%D0%BE_%D0%B2%D0%B2%D0%BE%D0%B4%D0%B0-%D0%B2%D1%8B%D0%B2%D0%BE%D0%B4%D0%B0), его основные функции заключаются в отображении и вводе данных. Устройство со значительным объёмом обработки данных называют *smart terminal* или «[толстый клиент](https://ru.wikipedia.org/wiki/%D0%A2%D0%BE%D0%BB%D1%81%D1%82%D1%8B%D0%B9_%D0%BA%D0%BB%D0%B8%D0%B5%D0%BD%D1%82)» ([англ.](https://ru.wikipedia.org/wiki/%D0%90%D0%BD%D0%B3%D0%BB%D0%B8%D0%B9%D1%81%D0%BA%D0%B8%D0%B9_%D1%8F%D0%B7%D1%8B%D0%BA) *fat client*). Терминал, сильно зависящий от своей хост-машины, на которой выполняются основные вычисления, называют [тонким клиентом](https://ru.wikipedia.org/wiki/%D0%A2%D0%BE%D0%BD%D0%BA%D0%B8%D0%B9_%D0%BA%D0%BB%D0%B8%D0%B5%D0%BD%D1%82) ([англ.](https://ru.wikipedia.org/wiki/%D0%90%D0%BD%D0%B3%D0%BB%D0%B8%D0%B9%D1%81%D0%BA%D0%B8%D0%B9_%D1%8F%D0%B7%D1%8B%D0%BA) *thin client*).

**Текстовый терминал** (или чаще просто **терминал**, иногда также **текстовая консоль**) — это интерфейс компьютера для последовательной передачи данных — ввода и изображения текста. Информация представляется в виде массива предопределённых знаков.

X-терминалы представляют собой комбинацию бездисковых рабочих станций и стандартных ASCII-терминалов. Бездисковые рабочие станции часто применялись в качестве дорогих дисплеев и в этом случае не полностью использовали локальную вычислительную мощь. Одновременно многие пользователи ASCII-терминалов хотели улучшить их характеристики, чтобы получить возможность работы в многооконной системе и графические возможности. Совсем недавно, как только стали доступными очень мощные графические рабочие станции, появилась тенденция применения "подчиненных" X-терминалов, которые используют рабочую станцию в качестве локального сервера.

На компьютерном рынке X-терминалы занимают промежуточное положение между персональными компьютерами и рабочими станциями.

**Сетево́й компью́тер** — [компьютер](https://ru.wikipedia.org/wiki/%D0%9A%D0%BE%D0%BC%D0%BF%D1%8C%D1%8E%D1%82%D0%B5%D1%80), являющийся компонентом архитектуры компьютер-сеть[[*источник не указан 2743 дня*]](https://ru.wikipedia.org/wiki/%D0%92%D0%B8%D0%BA%D0%B8%D0%BF%D0%B5%D0%B4%D0%B8%D1%8F:%D0%A1%D1%81%D1%8B%D0%BB%D0%BA%D0%B8_%D0%BD%D0%B0_%D0%B8%D1%81%D1%82%D0%BE%D1%87%D0%BD%D0%B8%D0%BA%D0%B8) и имеющий упрощённые структуры в отличие от персонального компьютера(небольшой объём памяти, возможно отсутствие дисковода и т. п.). Это аппаратная часть для выполнения программы сетевого компьютерного [терминала](https://ru.wikipedia.org/wiki/%D0%9A%D0%BE%D0%BC%D0%BF%D1%8C%D1%8E%D1%82%D0%B5%D1%80%D0%BD%D1%8B%D0%B9_%D1%82%D0%B5%D1%80%D0%BC%D0%B8%D0%BD%D0%B0%D0%BB). В качестве сетевого компьютера могут использоваться устаревшие модели [персональных компьютеров](https://ru.wikipedia.org/wiki/%D0%9F%D0%B5%D1%80%D1%81%D0%BE%D0%BD%D0%B0%D0%BB%D1%8C%D0%BD%D1%8B%D0%B9_%D0%BA%D0%BE%D0%BC%D0%BF%D1%8C%D1%8E%D1%82%D0%B5%D1%80) объединённые в большую иерархичную сеть [грид-вычислений](https://ru.wikipedia.org/wiki/%D0%93%D1%80%D0%B8%D0%B4" \o "Грид), в которой также присутствуют [сервера](https://ru.wikipedia.org/wiki/%D0%A1%D0%B5%D1%80%D0%B2%D0%B5%D1%80). В сетевом компьютере может отображаться экран входа в [учётную запись](https://ru.wikipedia.org/wiki/%D0%A3%D1%87%D1%91%D1%82%D0%BD%D0%B0%D1%8F_%D0%B7%D0%B0%D0%BF%D0%B8%D1%81%D1%8C) пользователя в операционную систему с дальнейшим отображением рабочего стола. А сама [операционная система](https://ru.wikipedia.org/wiki/%D0%9E%D0%BF%D0%B5%D1%80%D0%B0%D1%86%D0%B8%D0%BE%D0%BD%D0%BD%D0%B0%D1%8F_%D1%81%D0%B8%D1%81%D1%82%D0%B5%D0%BC%D0%B0) установлена на сервере.

Zero Client не имеет собственного программного обеспечения, операционной системы и драйверов. Отсутствуют также центральный процессор и память, но при этом обеспечивается полная функциональность настольного компьютера.   
Образно говоря, ПК разрезается на две части, которые соединяются по IP сети. На рабочем месте остается только дисплей, клавиатура и мышь, а вся вычислительная мощность располагается на сервере.

Преимущества и недостатки:   
• Стоимость устройства сведена к минимуму: нет процессора, нет памяти, нет дисков, нет оптического привода, что обеспечивает: 

* Минимальное количество оборудования на рабочем месте пользователя.
* Минимальное энергопотребление и тепловыделение.  
  • Каждый пользователь работает с собственной виртуальной машиной, на которой установ-лена стандартная ОС и приложения, поэтому:
* Нет необходимости в адаптации приложений.
* Сбой виртуальной машины одного пользователя не затрагивает работу остальных
* Нулевой клиент легко заменить на другой, при этом текущее состояние рабочего места пользователя сохранится (состояние ОС и активных приложений пользователя остается без изменений). Такая оперативная замена возможна, поскольку виртуальная машина пользователя продолжает работать на сервере, даже когда она не используется нулевым клиентом. К тому же, в нулевом клиенте полностью отсутствуют какие-либо настройки (в том числе сетевые), подключение происходит автоматически.
* Управление нулевыми клиентами существенно упрощается, так как все рабочие места это виртуальные машины, работающие на одном сервере, что обеспечивает:
* Централизованное администрирование - все рабочие места управляются при помощи централизованной консоли управления виртуальными машинами.
* Централизованная установка и обновление ОС и приложений. Быстрое развертывание - новое виртуальное рабочее место может быть оперативно создано путем клонирования уже существующего шаблона виртуальной машины.
* Легко изменить конфигурацию рабочего места путем изменения настроек виртуальной машины.
* Наращивание вычислительных мощностей, происходит только за счет добавления/модернизации компонентов сервера.