

# ТЕМА 1: Основы построения ЭВМ

---

## Содержание:

1. Принципы фон Неймана
2. Шинная структура связей в ЭВМ
3. Виды архитектур МПС
4. Режимы обмена по магистрали

# Принципы фон Неймана

---

В 1946 году Д. фон Нейман, Г. Голдстайн и А. Беркс в своей совместной статье изложили новые принципы построения и функционирования ЭВМ:

## 1. Использование двоичной системы счисления

- Упрощается техническая реализация аппаратуры для обработки двоичных кодов.

## 2. Принцип программного управления

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

## 3. Принцип однородности памяти

- Как программы, так и данные хранятся в одной и той же памяти и кодируются в двоичной системе; над командами можно выполнять такие же действия, как и над данными.

## 4. Принцип адресности

- Память состоит из пронумерованных ячеек; процессору в произвольный момент времени доступна любая ячейка.

# Устройство ЭВМ по фон Нейману

В соответствии с принципами фон Неймана в состав ЭВМ входят:

1. Арифметико-логическое устройство

- выполняет арифметические и логические операции.

2. Устройство управления

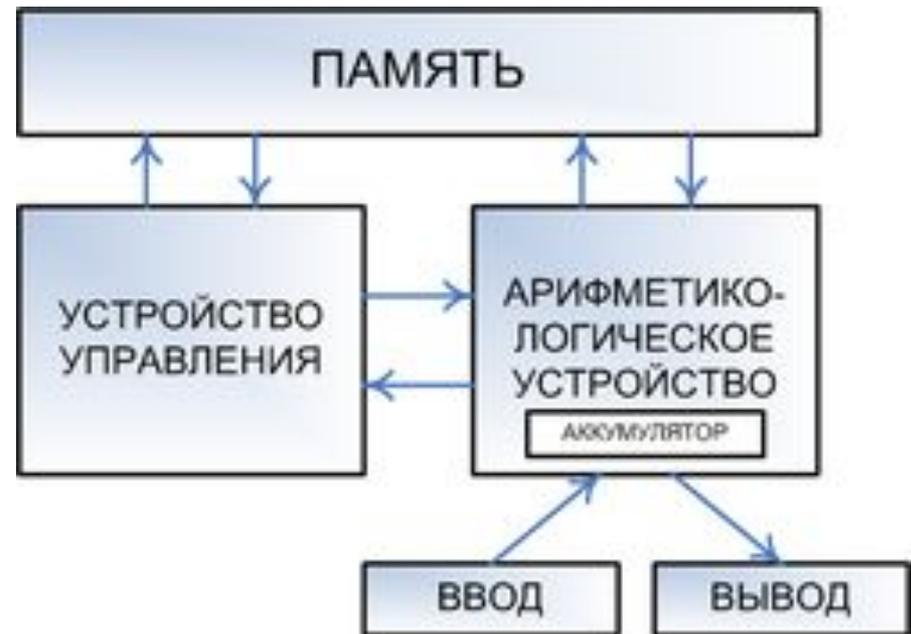
- организует процесс выполнения программ.

3. Запоминающее устройство

- хранит программы и данные.

4. Внешние устройства

- ввод-вывод информации.



Структура ВС, построенной на принципах фон Неймана

# Микропроцессор и его функции

---

Ядром любой современной ЭВМ является **микропроцессор** — электронный узел, в котором объединены арифметико-логическое устройство и устройство управления.

**Микропроцессор** выполняет следующие функции:

- ◆ арифметические операции (сложение, вычитание);
- ◆ логические операции (сдвиг, сравнение);
- ◆ временное хранение кодов (во внутренних регистрах);
- ◆ пересылку кодов;
- ◆ управление взаимодействием всех узлов ЭВМ.

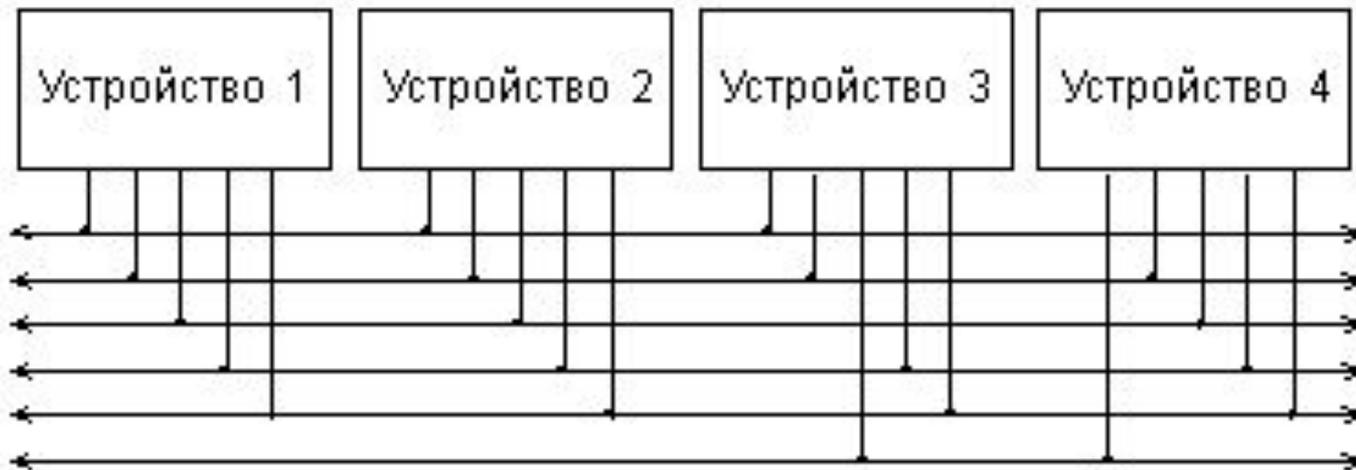
**Остальные узлы**, входящие в микропроцессорную систему выполняют вспомогательные функции:

- ◆ хранение информации (память),
- ◆ обмен информации с устройствами ввода-вывода.

# Структура связей в ЭВМ

В ЭВМ реализована *шинная структура связей*, при которой все сигналы между устройствами передаются по одним и тем же линиям связи, но в разное время.

В результате такой организации количество линий связи существенно сокращается, а правила обмена информацией (протоколы) упрощаются. Группа линий связи, по которым передаются сигналы или коды одного назначения, называется шиной. Совокупность шин разного назначения называется **магистралью**.



*Шинная структура связей*

# Достоинства и недостатки

---

**Главное достоинство шинной структуры связей** состоит в том, что все устройства, подключенные к шине, должны принимать и передавать информацию по одним и тем же правилам (протоколам обмена информацией по шине). Соответственно, все узлы, отвечающие за обмен с шиной в этих устройствах, должны быть единообразны, унифицированы.

**Существенный недостаток шинной структуры** связан с тем, что все устройства подключаются к каждой линии связи параллельно. Поэтому любая неисправность любого устройства может вывести из строя всю систему, если она портит линию связи. По этой же причине отладка системы с шинной структурой связей довольно сложна и обычно требует специального оборудования.

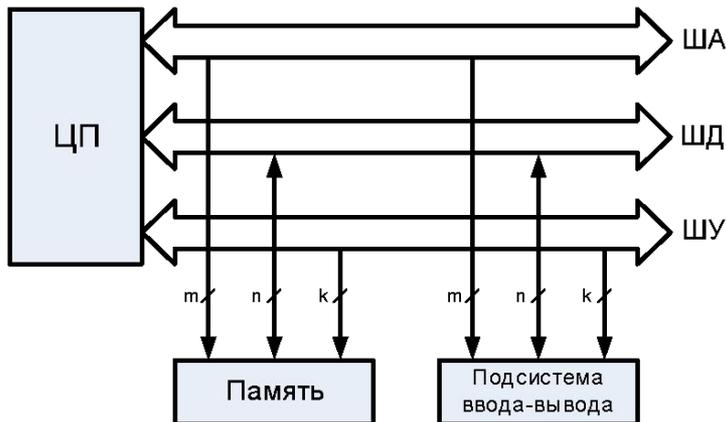
При шинной структуре связей вся информация передается по линиям связи **последовательно во времени**, по очереди, что *снижает быстроедействие системы* по сравнению с классической структурой связей.

# Структура магистрали

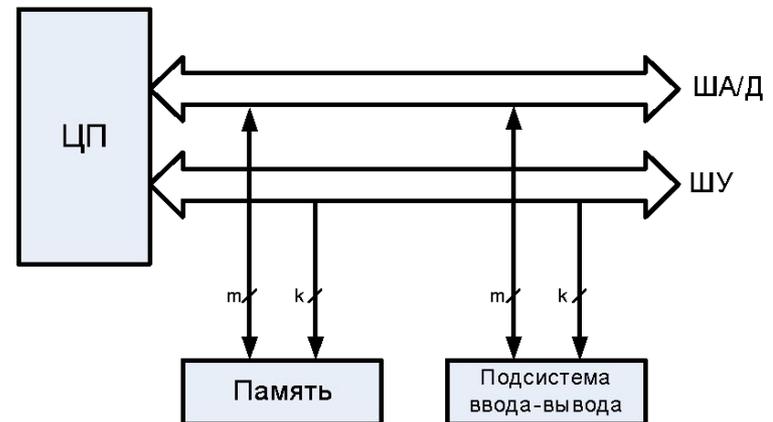
Магистраль представляет собой единый набор шин:

1. шина данных (ШД), для передачи данных;
2. шина адреса (ША), для передачи адресов ячеек памяти или портов;
3. шина управления (ШУ), для передачи сигналов управления работой системы.

Для сокращения числа проводников в магистрали адреса и данные могут передаваться по единой шине адреса-данных (ША/Д) по очереди. Такой подход называется мультиплексированием.



Магистраль с тремя шинами



Магистраль с двумя шинами

# Назначение шин магистрали

---

**Шина данных** — это основная шина, которая используется для передачи информационных кодов между всеми устройствами ЭВМ. Обычно в пересылке информации участвует процессор, который передает код данных в какое-то устройство или в ячейку памяти или же принимает код данных из какого-то устройства или из ячейки памяти. Но возможна также и передача информации между устройствами без участия процессора (режим прямого доступа к памяти).

**Шина адреса** служит для определения адреса устройства, с которым процессор обменивается информацией в данный момент. Каждому устройству (кроме процессора), каждой ячейке памяти в ЭВМ присваивается собственный адрес. Когда адрес, выставленный процессором на шине адреса, совпадает с адресом конкретного устройства, оно подключается к шине данных и происходит обмен информацией.

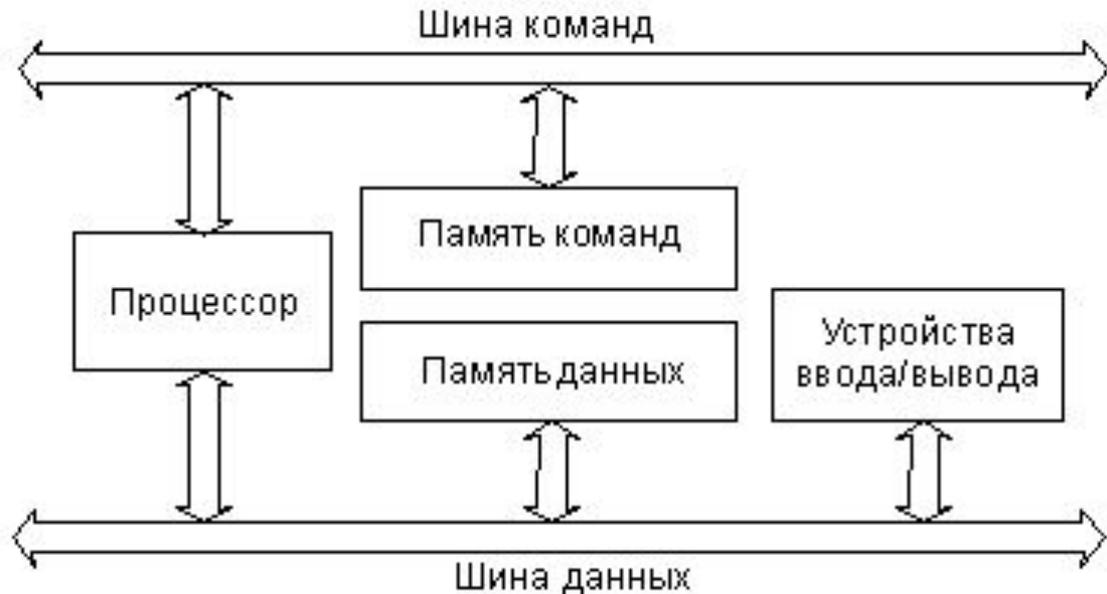
**Шина управления** в отличие от шины адреса и шины данных состоит из линий передачи отдельных управляющих сигналов. Каждый из этих сигналов во время обмена информацией выполняет свою функцию: стробирование передаваемых или принимаемых данных, подтверждение приема данных, для сброса всех устройств в исходное состояние, синхронизация работы устройств и т.д.

# Виды архитектур МПС

## 1. Одношинная или принстонская архитектура



## 2. Двухшинная или гарвардская архитектура



# **Одношинная (принстонская) архитектура**

---

Имеет общую, единую шину для данных и команд. Соответственно, в составе системы в этом случае присутствует одна общая память, как для данных, так и для команд.

Архитектура с общей шиной распространена гораздо больше, она применяется, например, в персональных компьютерах и в сложных микрокомпьютерах.

## ***Достоинства***

- ◆ Процессор обслуживает только одну шину.
- ◆ Наличие единой памяти большого объема (до нескольких гигабайт) позволяет гибко распределять ее объем между кодами данных и команд.
- ◆ Система способна решать самые сложные задачи.

## ***Недостатки***

- ◆ Пересылка данных и чтение команд происходит по одной магистрали последовательно во времени (нельзя делать одновременно).
- ◆ Быстродействие можно увеличить только за счет конвейерной технологии и быстрой кэшпамяти, повышения тактовой частоты, сокращение времени выполнения команд.

# ***Двухшинная (гарвардская) архитектура***

---

Имеет отдельные шины данных и команд. Эта архитектура предполагает наличие в системе отдельной памяти для данных и отдельной памяти для команд. Обмен процессора с каждым из двух типов памяти происходит по своей шине. Преимущества архитектуры реализуются внутри одной микросхемы, поэтому применяется в микроконтроллерах.

## ***Недостатки***

- ◆ Процессор должен работать одновременно с двумя потоками кодов, обслуживать обмен по двум шинам.
- ◆ Нет возможности гибкого перераспределения памяти - программа может размещаться только в памяти команд, данные - только в памяти данных.
- ◆ Память данных и память команд имеют не слишком большой объем, поэтому ограничен круг решаемых задач (не слишком сложные задачи).

## ***Достоинства***

- ◆ Обмен по обеим шинам может быть независимым, параллельным во времени.
- ◆ Структуры шин (разрядность, скорость обмена и т.д.) оптимизированы для круга решаемых задач.
- ◆ Реализуется максимальное быстродействие при заданной тактовой частоте.

# ***Основные режимы обмена по магистрали***

---

Любая развитая микропроцессорная система поддерживает ***три основных режима обмена*** по магистрали:

- ◆ программный обмен информацией;
- ◆ обмен с использованием прерываний (Interrupts);
- ◆ обмен с использованием прямого доступа к памяти (ПДП, DMA - Direct Memory Access).

# Программный обмен информацией

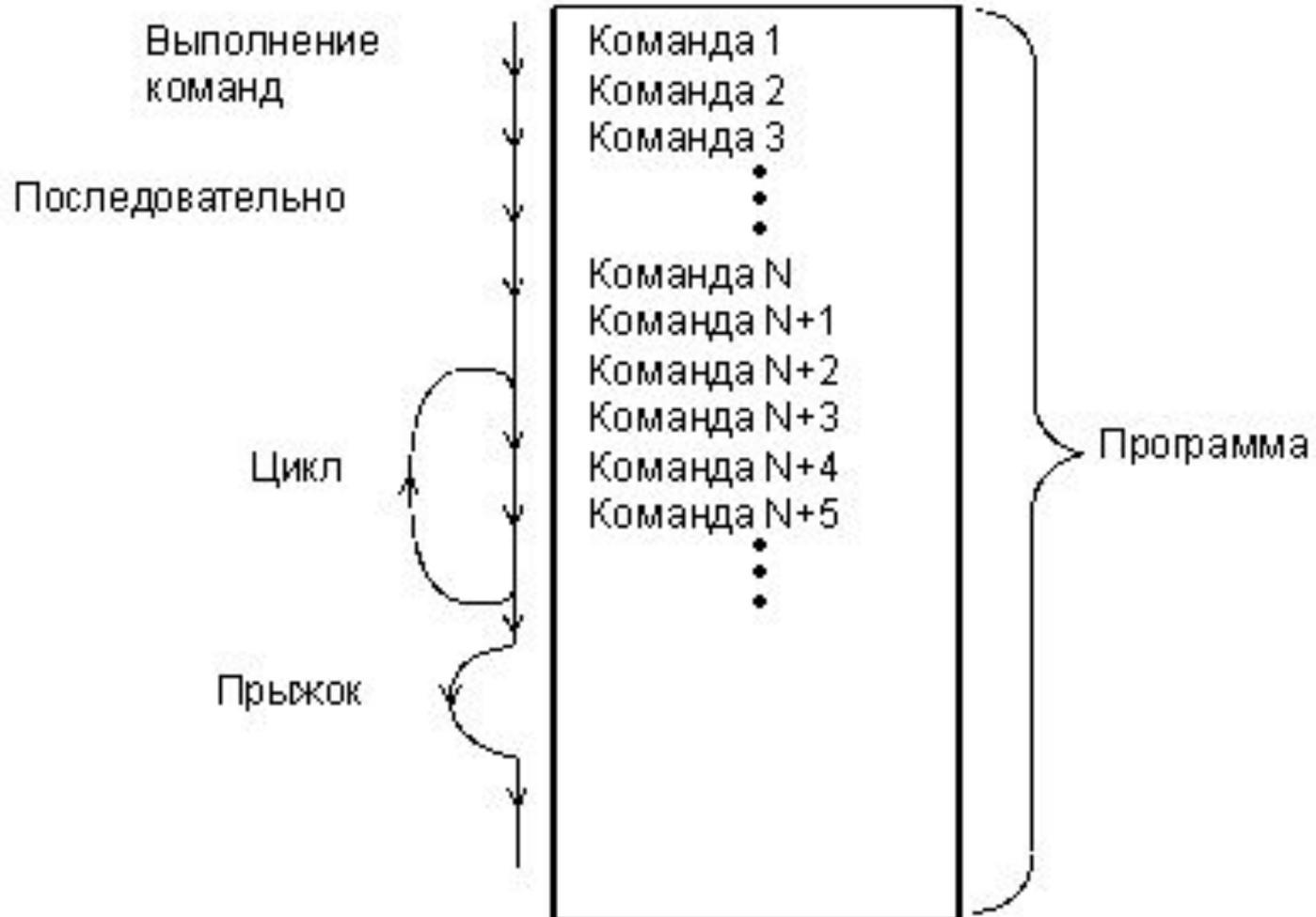
---

**Программный обмен информацией** является основным в любой микропроцессорной системе. Он предусмотрен всегда, без него невозможны другие режимы обмена. В этом режиме процессор является единоличным хозяином (или задатчиком, Master) системной магистрали. Все операции (циклы) обмена информацией в данном случае инициируются только процессором, все они выполняются строго в порядке, предписанном исполняемой программой.

Процессор выбирает из памяти коды команд и исполняет их, читая данные из памяти или из устройства ввода/вывода, обрабатывая их, записывая данные в память или передавая их в устройство ввода/вывода.

Путь процессора по программе может быть линейным, циклическим, может содержать переходы (прыжки), но он всегда непрерывен и полностью находится под контролем процессора. ***Ни на какие внешние события, не связанные с программой, процессор не реагирует.*** Все сигналы на магистрали в данном случае контролируются процессором.

# Схема программного обмена



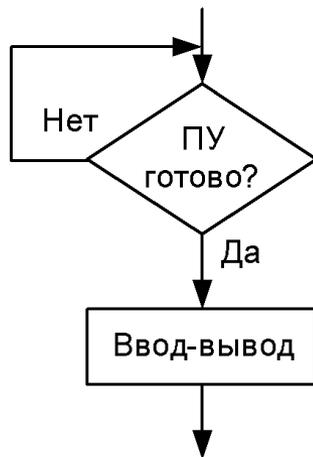
# Взаимодействие с ПУ в программном режиме

При программном обмене взаимодействие микропроцессора с периферийным устройством может быть **прямым** или **условным**.

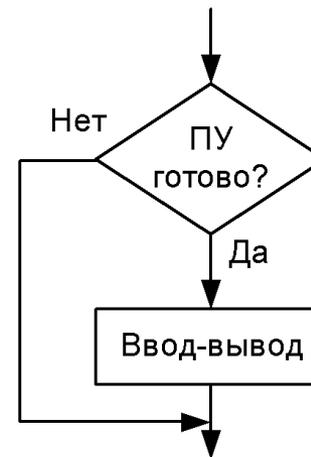
При **прямом** обмене в соответствии с программой процессор выполняет команды ввода или вывода не проверяя готовности ПУ.

При **условном** обмене программа периодически опрашивает регистр состояния ПУ, анализирует и на основе анализа принимает решение о готовности ПУ. При готовности ПУ к обмену процессор выполняет операции передачи данных. В случае неготовности ПУ возможны два режима:

*1- обмен с занятием цикла*



*2- совмещенный обмен*



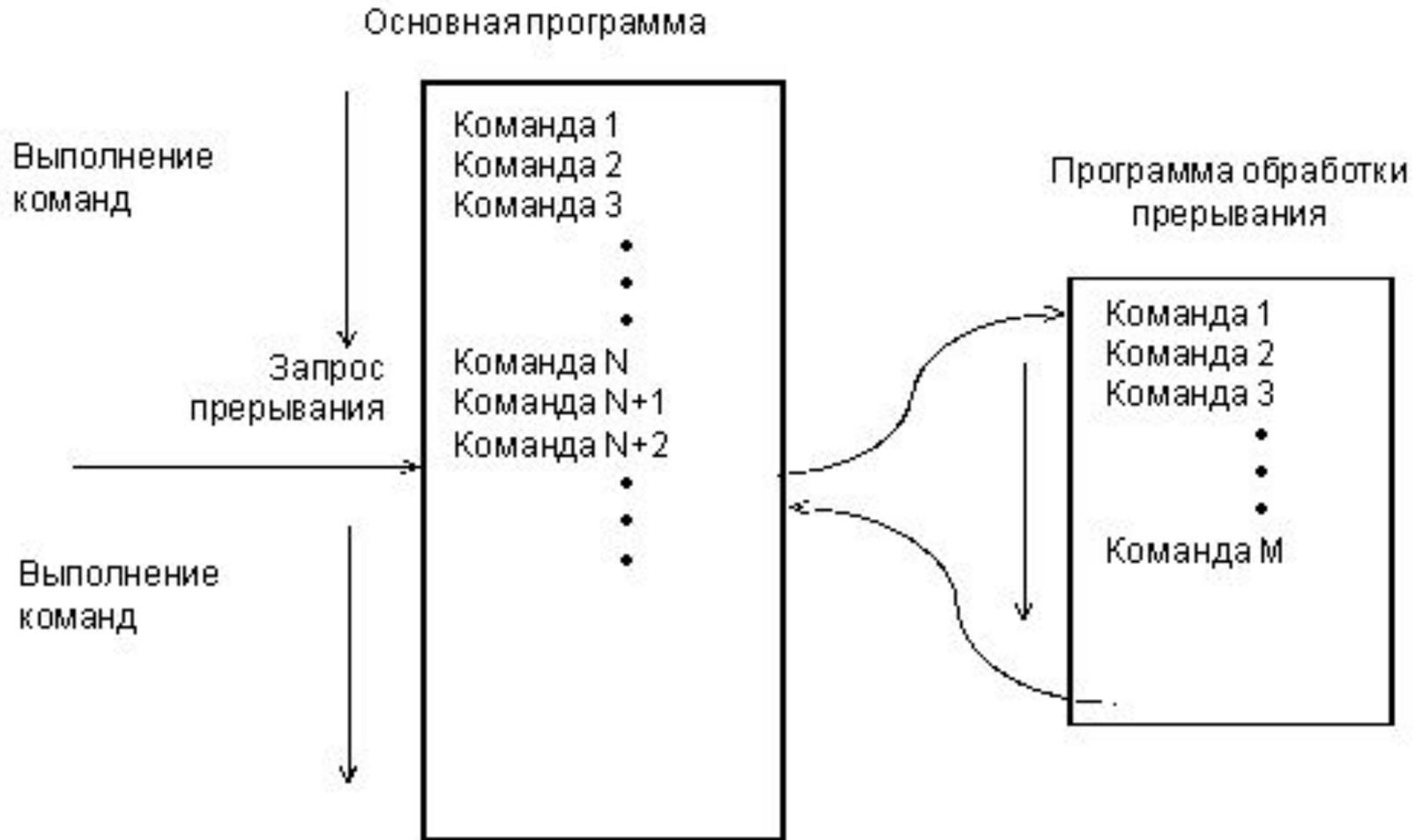
# Обмен с использованием прерываний

---

Обмен с использованием прерываний используется, когда необходима реакция микропроцессорной системы на какое-то внешнее событие, на приход внешнего сигнала. В случае компьютера внешним событием может быть, например, нажатие на клавишу клавиатуры или приход по локальной сети пакета данных. Компьютер должен реагировать на это, соответственно, выводом символа на экран или же чтением и обработкой принятого по сети пакета данных.

В отличие от программного обмена, когда процессор постоянно контролирует наличие события (так называемый метод опроса флага состояния ПУ), **в режиме прерывания** происходит следующее. Процессор, получив запрос прерывания от внешнего устройства (IRQ - Interrupt ReQuest), заканчивает выполнение текущей команды и переходит к программе обработки прерывания. Закончив выполнение программы обработки прерывания, он возвращается к прерванной программе с той точки, где его прервали.

# Схема обслуживания прерываний



# Функции процессора в режиме прерываний

---

В режиме обмена по прерываниям вся работа, выполняется самим процессором, т.е. все сигналы на магистрали выставляются процессором, то есть он полностью контролирует магистраль.

Для обслуживания прерываний в систему иногда вводится специальный модуль - **контроллер прерываний**. Этот контроллер в обмене информацией не участвует. Его задача состоит в том, чтобы упростить работу процессора с внешними запросами прерываний:

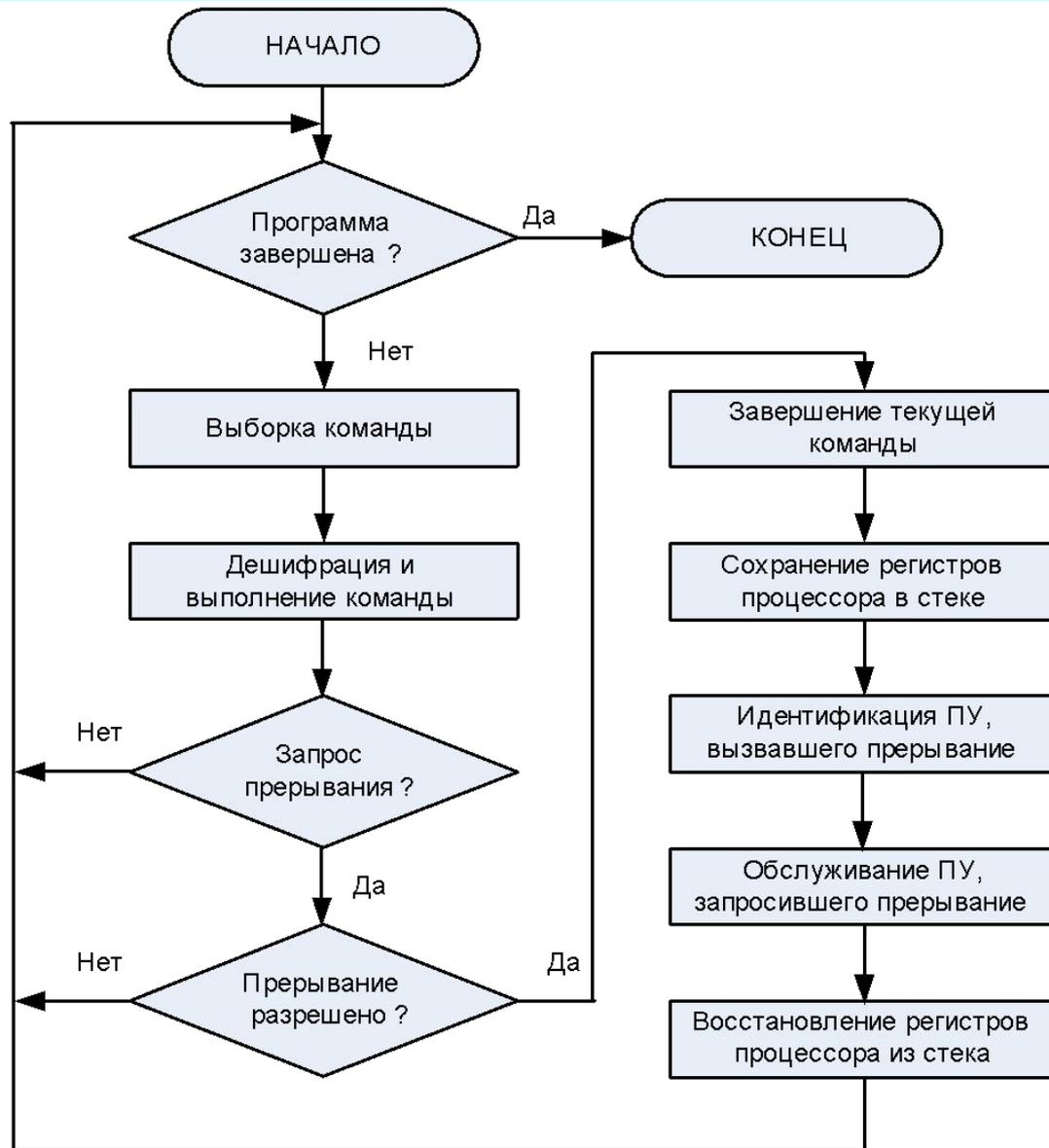
- ◆ определить источник запроса на прерывание;
- ◆ если запросов несколько, выбрать устройство с наибольшим приоритетом (выполнить арбитраж).

Работа контроллера прерываний программно управляется процессором по системной магистрали.

Реакция на внешнее событие по прерыванию в общем случае медленнее, чем при программном режиме.

Естественно, никакого ускорения работы системы режим прерывания не дает. Его применение позволяет только отказаться от постоянного опроса флага внешнего события.

# Алгоритм обмена по прерываниям



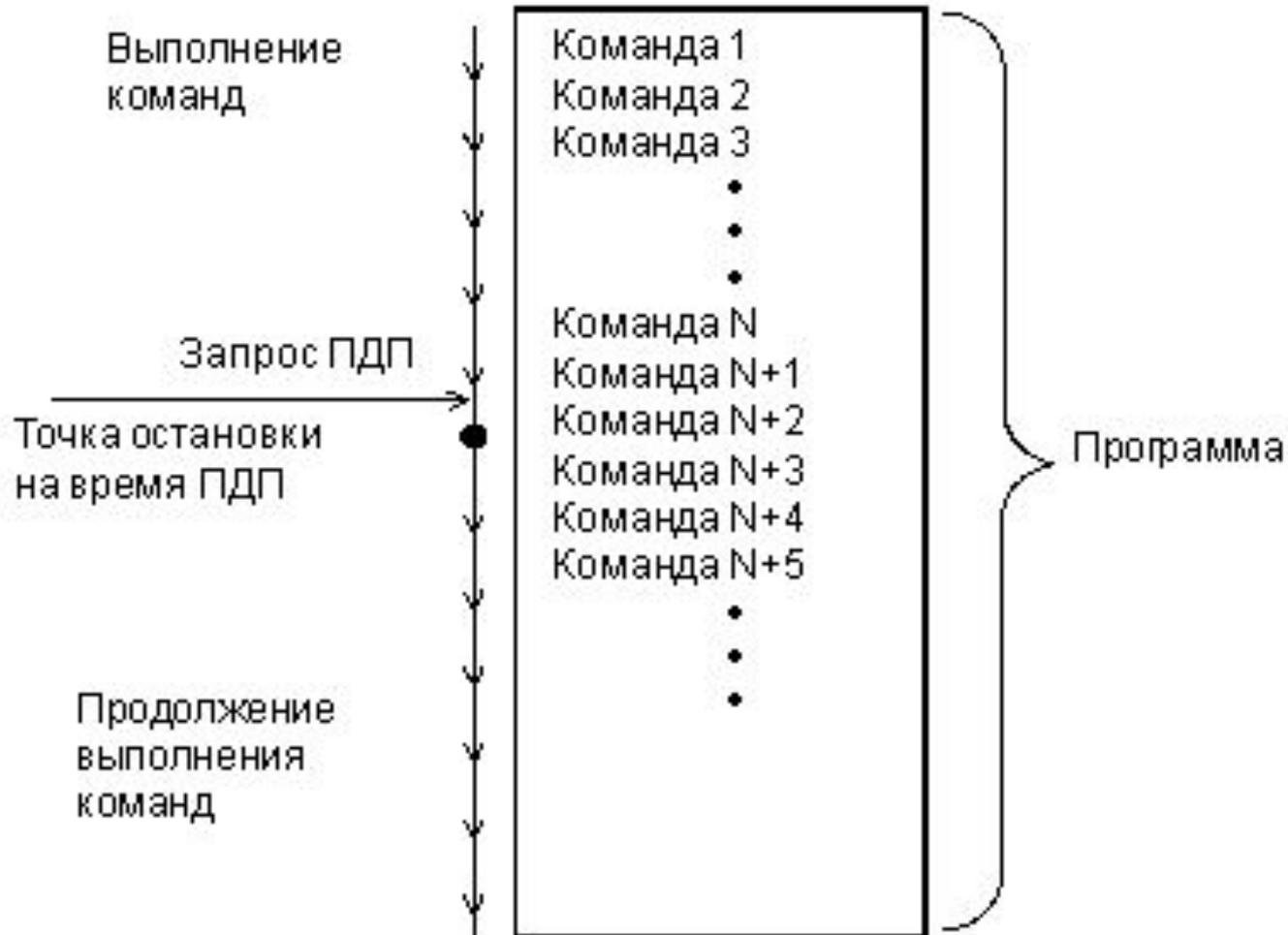
# Прямой доступ к памяти

---

*Прямой доступ к памяти* (ПДП или DMA) — это режим, принципиально отличающийся от двух ранее рассмотренных режимов тем, что обмен по системной шине идет без участия процессора. Внешнее устройство, требующее обслуживания, сигнализирует процессору, что необходим режим ПДП. В ответ на запрос процессор заканчивает выполнение текущей команды и отключается от всех шин, сигнализируя запросившему устройству, что обмен в режиме ПДП можно начинать.

Операция ПДП сводится к пересылке информации из устройства ввода/ вывода в память или же из памяти в устройство ввода/вывода. Когда пересылка информации будет закончена, процессор вновь возвращается к прерванной программе, продолжая ее с той точки, где его прервали. Это похоже на режим обслуживания прерываний, но в данном случае процессор не участвует в обмене. Как и в случае прерываний, реакция на внешнее событие при ПДП медленнее, чем в программном режиме.

# Схема обслуживания ПДП



# Функции контроллера ПДП

---

Для реализации ПДП требуется введение в систему дополнительного устройства - **контроллера ПДП**, которое будет осуществлять полноценный обмен по системной магистрали без участия процессора.

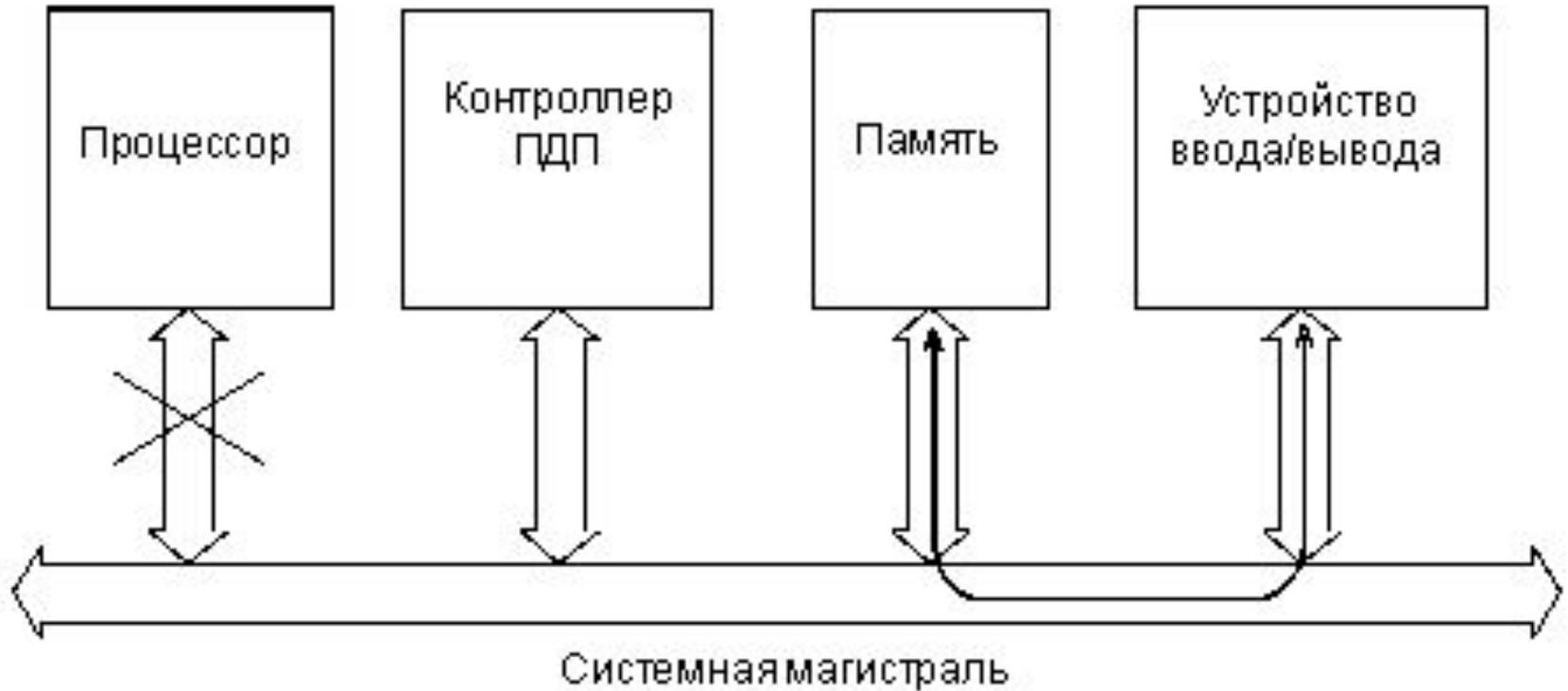
Перед началом обмена в режиме ПДП процессор инициализирует контроллер ПДП, т.е. передает в его регистры сведения откуда следует брать информацию и куда ее помещать. К основным регистрам контроллера ПДП относятся:

- ◆ регистр начального адреса размещения в оперативной памяти пересылаемого массива;
- ◆ регистр длины массива пересылаемых слов данных.

В процессе обмена после пересылки очередного слова регистр адреса увеличивает свое значение на 1, а регистр длины массива уменьшает на 1 свое значение. Признаком окончания обмена является обнуление регистра длины массива. После завершения обмена контроллер ПДП передает управление процессору.

**Контроллер ПДП можно считать специализированным процессором**, который отличается тем, что сам не участвует в обмене, т.е. не принимает в себя информацию и не выдает ее.

# Информационные потоки в режиме ПДП



# Реализации режима ПДП

---

Обмен в режиме ПДП может обеспечить более высокую скорость передачи информации, чем программный обмен, так как процессор передает данные медленнее, чем специализированный контроллер ПДП. Однако скорость обмена в режиме ПДП обычно ограничена пропускной способностью магистрали:

- ◆ шина ISA – 16 Мбайт/сек;
- ◆ шина PCI – 132 Мбайт/сек;
- ◆ шина PCI Express (x32) – 6400 Мбайт/сек.

Контроллер ПДП может входить в состав устройства ввода/вывода, которому необходим режим ПДП или даже в состав нескольких устройств ввода/вывода.

Если в системе уже имеется самостоятельный контроллер ПДП, то это может в ряде случаев существенно упростить аппаратуру устройств ввода/вывода, работающих в режиме ПДП.