

Архитектура ЭВМ

АРБУЗОВА АНАСТАСИЯ ВИКТОРОВНА
СТАРШИЙ ПРЕПОДАВАТЕЛЬ КАФЕДРЫ АСУ
AARBUZOVA@LIST.RU

Несколько вопросов к аудитории

1. С какими ЭВМ вы сегодня столкнулись по дороге в университет?
2. Сколько времени в течение дня вы проводите с ЭВМ?
Каково их основное назначение?
3. Почему вы выбрали вашу специальность?

Система оценивания

Посещение лекции + прохождение опроса – 10 баллов

Практическое задание 1 – до 35 баллов

Практическое задание 2 – до 35 баллов

Итоговый тест – до 20 баллов

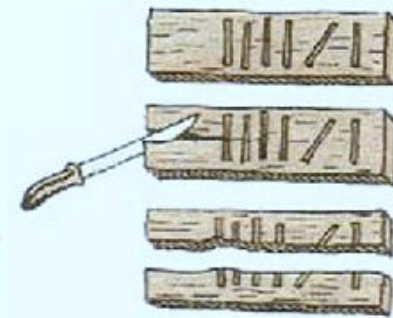
Студенты проходят тест в начале 2го практического занятия, строго в аудитории!

Архитектура ЭВМ





Пальцы рук и ног



Зарубки

"вестоничская кость", 30 тыс. лет до н.э.



Узелки



Абак, V век до н.э. по XVIII век н.э.

счёт осуществлялся с помощью размещённых на полосах камней или других подобных предметов



Счеты, около 500 г. н.э.

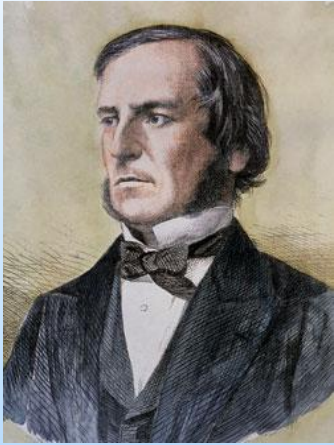
Вычислительная машина с программным управлением Чарльза Беббиджа (1830-1846 гг.)

Прототип современного компьютера, имеет схожую архитектуру, включает в себя:

- арифметическое устройство (на основе зубчатых колес);
- запоминающее устройство;
- устройство управление;
- устройство ввода-вывода (с использованием перфокарт)

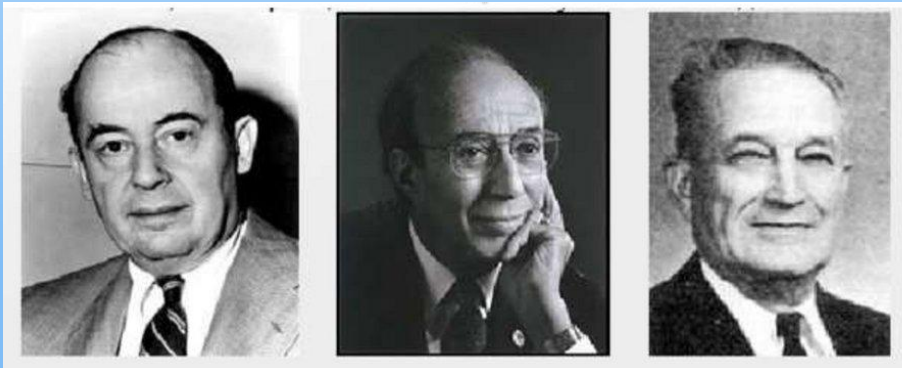
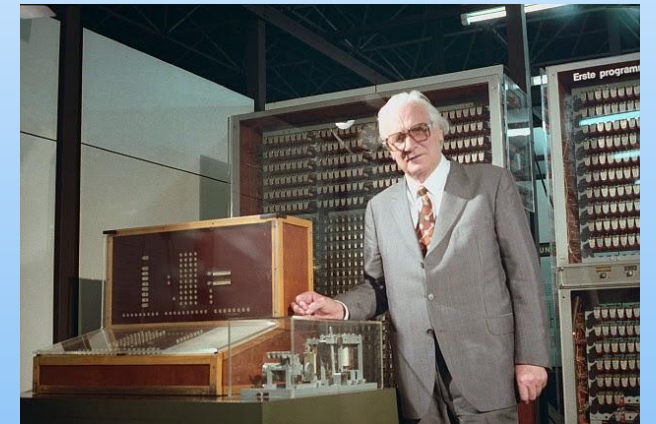


Программы для решения задач на машине Беббиджа, а также описание принципов ее работы были составлены Адой Лавлейс. В честь нее назван универсальный язык программирования Ада.



1848 и 1854 гг. Джон Буль – основатель математической логики, которая легла в основу всех современных вычислительных машин

1941 г. Конрад Цузе – создатель первого в мире действующего компьютера Z3



Джон фон
Нейман

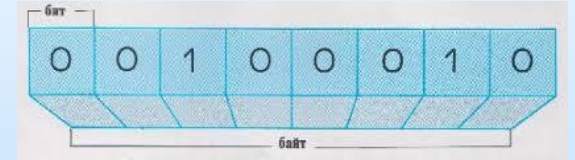
Герман
Голдстайн

Артур Беркс

1946 г.

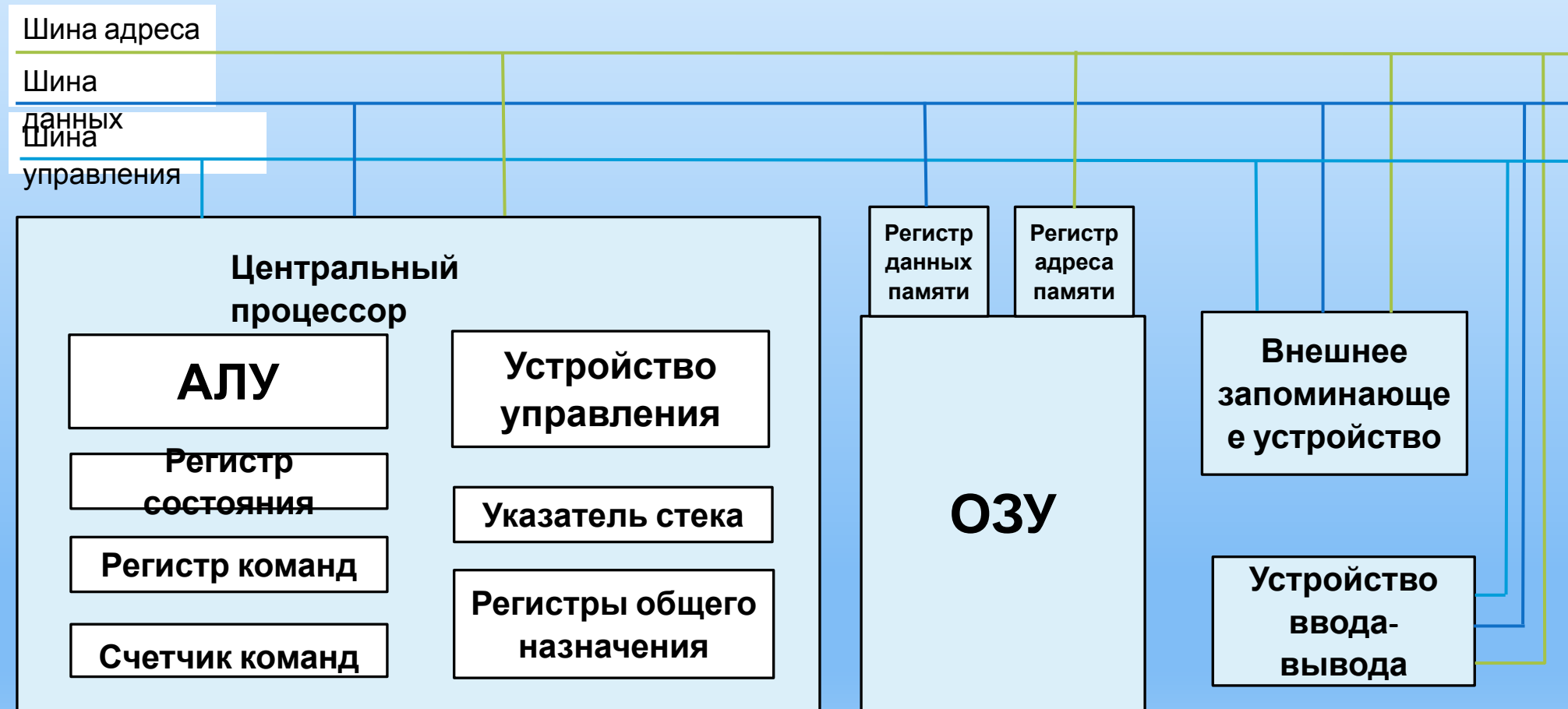
Джон фон Нейман, Герман Голдстайн, Артур Беркс
«Предварительное рассмотрение логического
конструирования электронного вычислительного
устройства»

Принципы организации ЭВМ

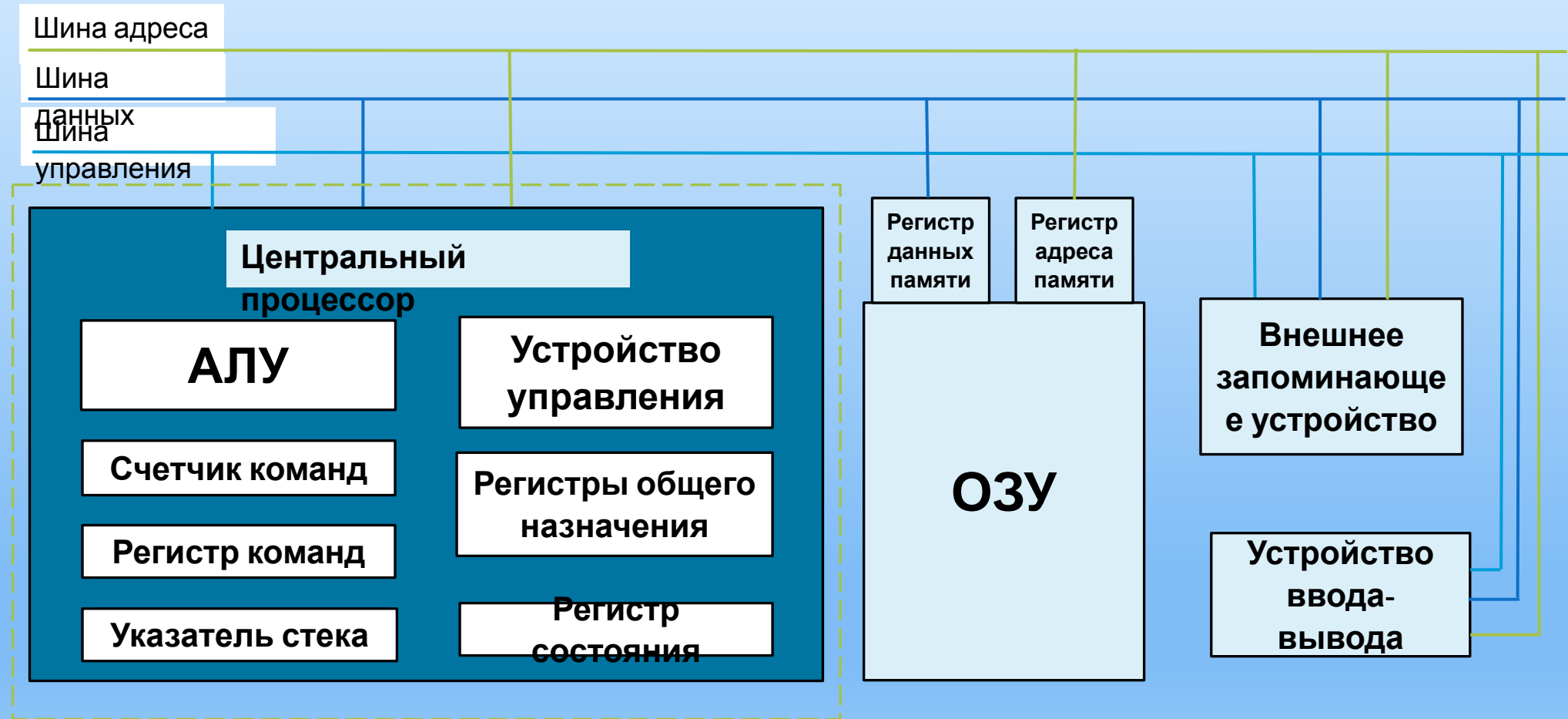


1. Принцип двоичного кодирования.
2. Принцип однородности памяти.
3. Принцип адресуемости памяти.
4. Принцип последовательного программного управления.
5. Принцип условного перехода.

Базовая структура ЭВМ Фон Неймана



Центральный процессор

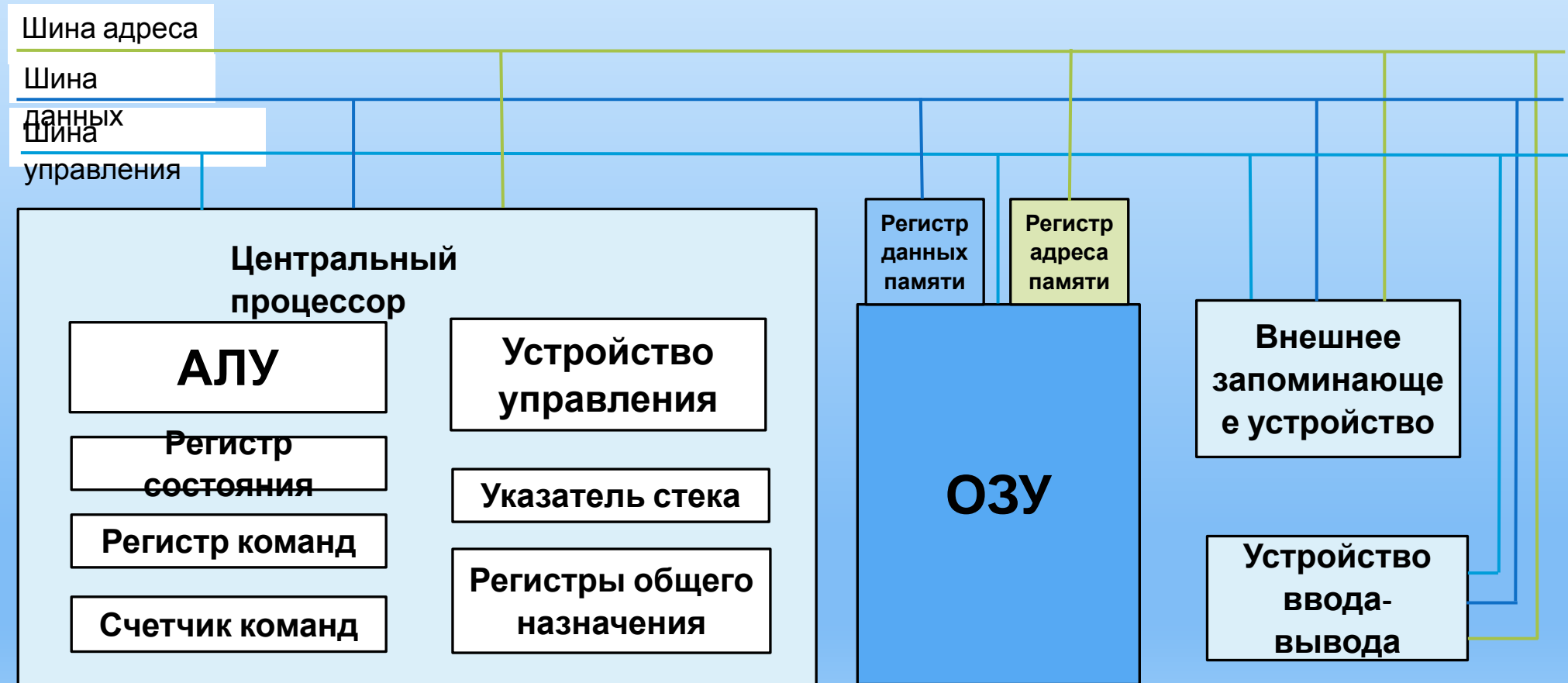


Характеристики процессора

- ☐ *разрядность процессора*
- ☐ *система команд процессора*
- ☐ *быстродействие процессора*



Оперативное запоминающее устройство



Периферийные устройства



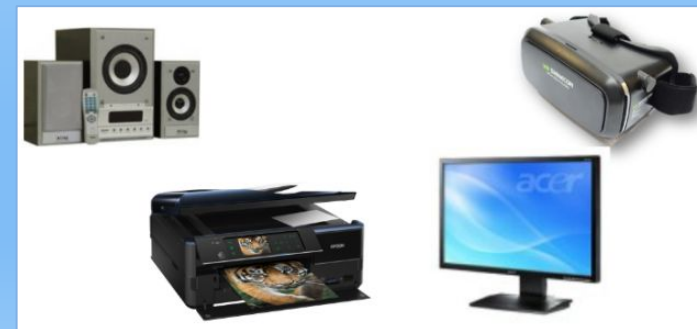
запоминающие
устройства:
HDD, SSD,
оптические
накопители,
flash-накопители
и др.;

устройства для
обмена
информацией
между ЭВМ:
модемы,
сетевые
интерфейсы.



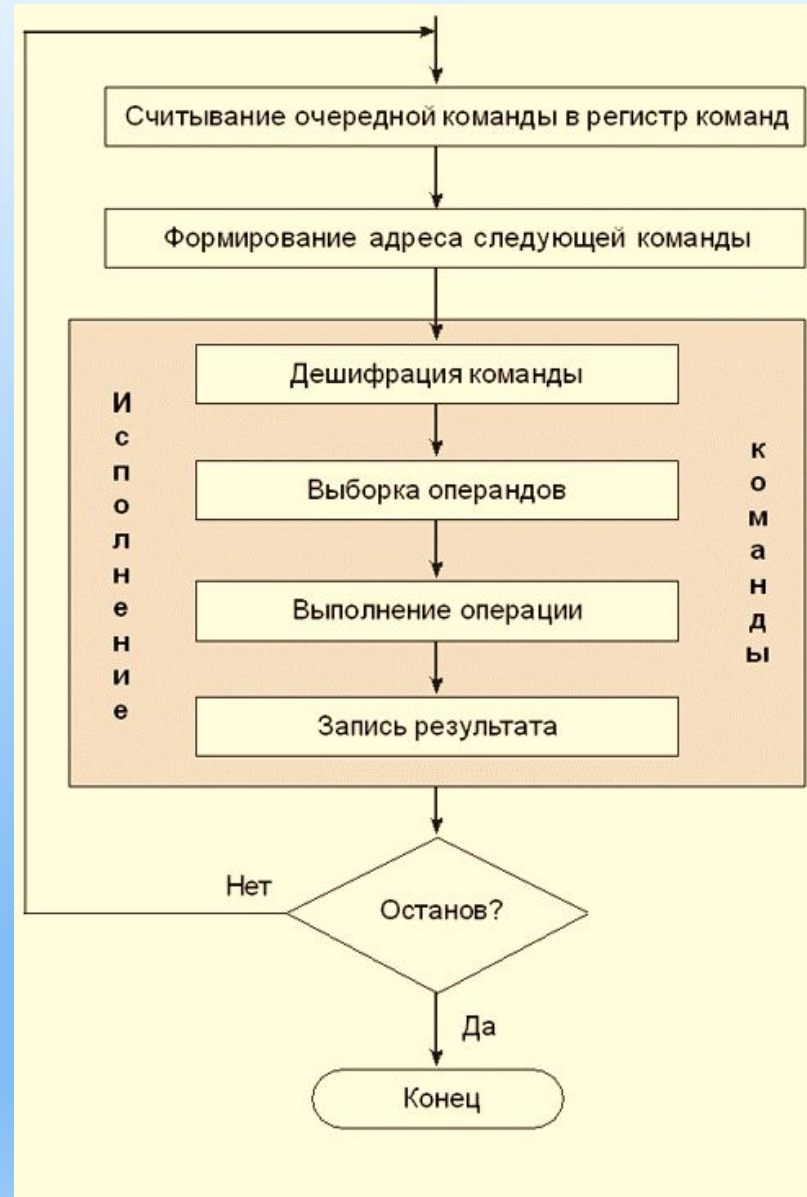
устройства
ввода
информации:
клавиатура,
мышь,
микрофон,
сканер и др.;

устройства
вывода
информации:
монитор,
колонки,
принтер и др.;



Перерыв 10 минут

Цикл выполнения команды

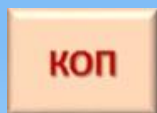
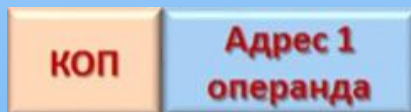
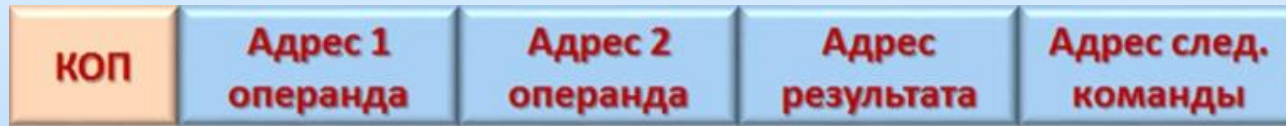


Система команд ЭВМ

По виду выполняемых операций различают следующие основные группы команд:

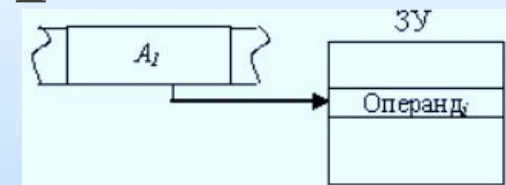
- ☐ команды арифметических операций;
- ☐ команды логических операций;
- ☐ команды пересылки данных;
- ☐ команды ветвления и переходов;
- ☐ команды управления, ввода/вывода и работы со стеком.

Структура команды

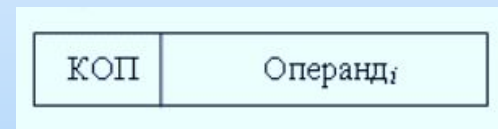


Способы адресации операндов

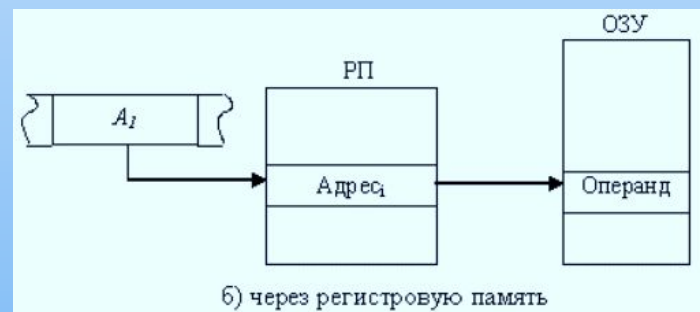
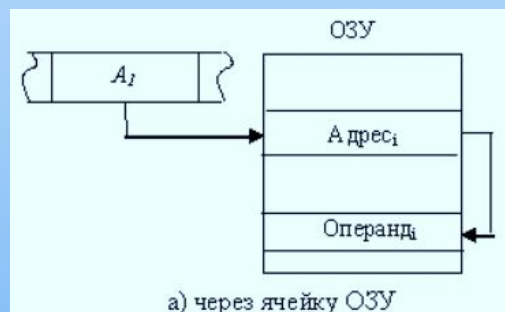
Прямая адресация. Физический адрес операнда совпадает с кодом в адресной части команды.



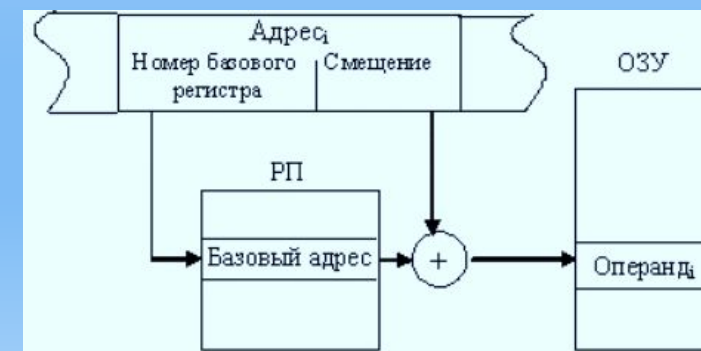
Непосредственная адресация. В команде содержится непосредственно сам операнд.



Косвенная адресация. В команде указывается адрес ячейки памяти (а), или номер регистра (б), в котором находится адрес операнда.



Относительная адресация. Память логически разбивается на сегменты. Адрес ячейки памяти состоит из адреса начала сегмента (базовый адрес) и смещение адреса операнда в сегменте.



Команды передачи управления

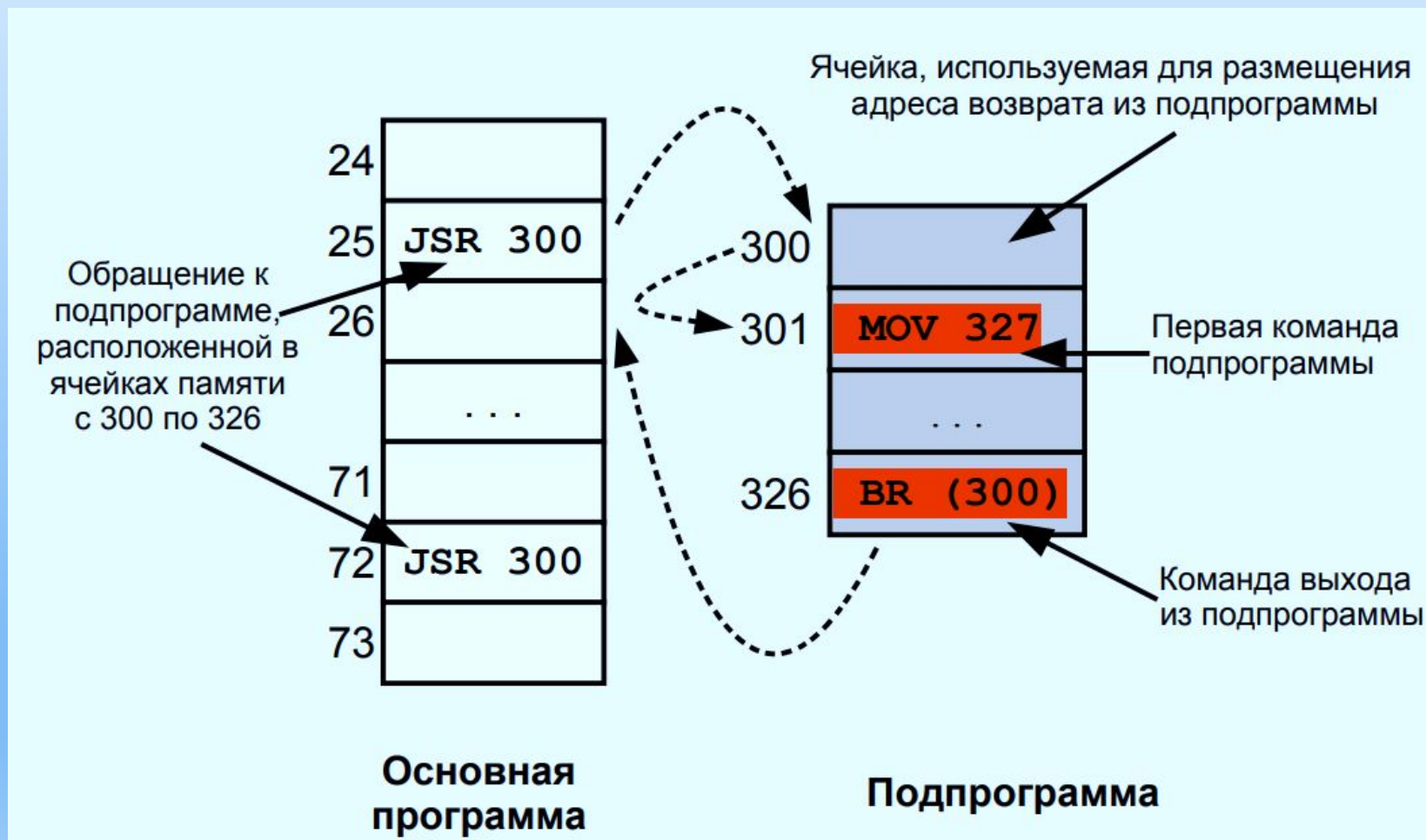
Команды передачи управления позволяют путем изменения содержимого регистра-счетчика команд процессора реализовать переходы в нужные точки программы.

Переходы могут быть **безусловными и условными**.

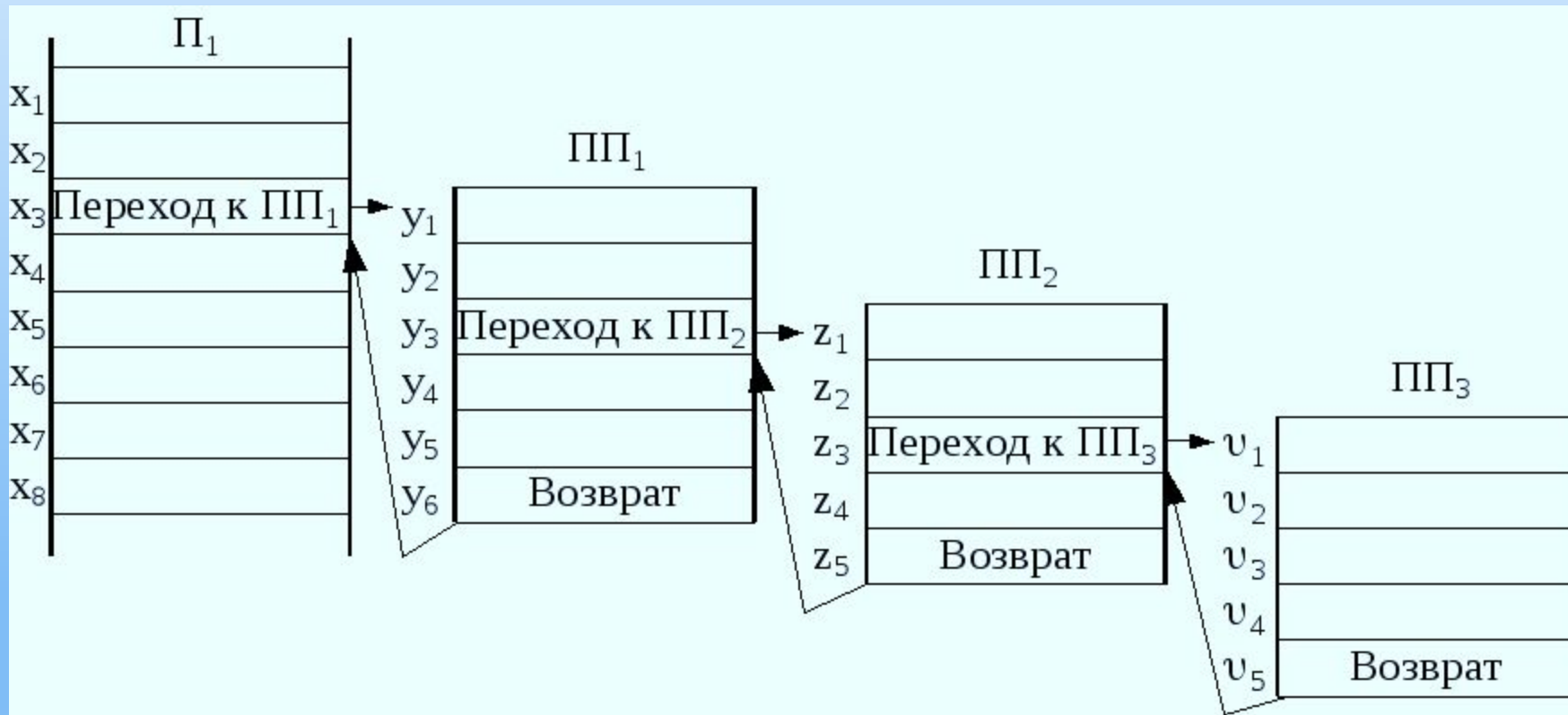
Механизм корректировки счетчика команд при реализации **условного** перехода:

- Анализ флажков регистра состояния процессора.
- При соответствии условию - изменение содержимое регистра счетчика-команд.
- При невыполнении условия - выборка следующей по порядку команды.

Вызов подпрограммы и возврат из нее



Реализация нескольких вложенных программ



Принцип работы стека



Особенности RISC-архитектуры процессора

CISC

- Большое количество команд
- Много типов данных
- Различная длина инструкций
- Небольшое количество регистров
- Ориентация на процессор

RISC

- Уменьшенное количество команд
- Только основные типы
- Фиксированная длина инструкций
- Большой регистровый файл
- Более глубокое использование компилятора

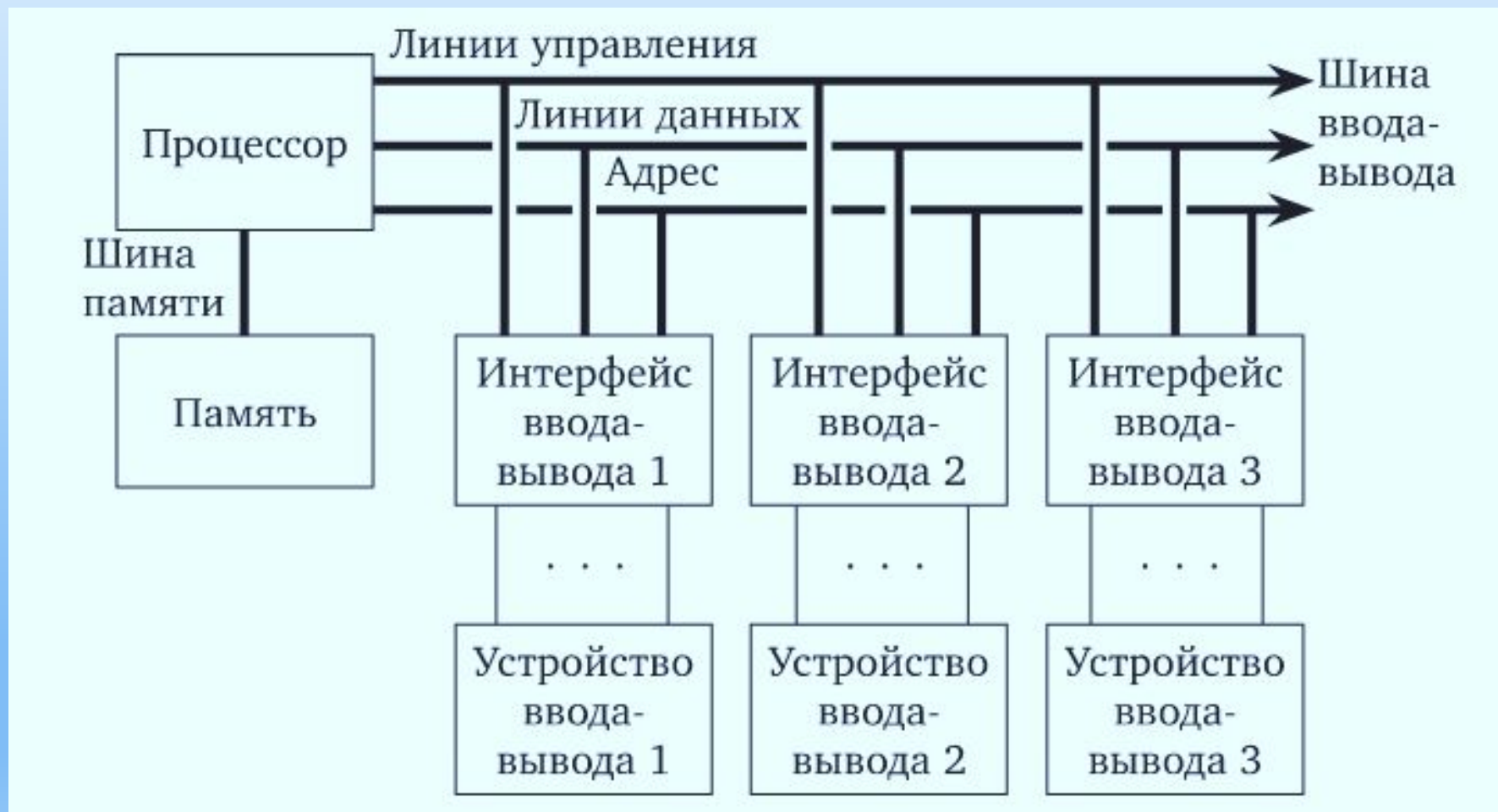
Перерыв 20 минут

Обмен данными с внешними устройствами

Возникающие проблемы

1. Разнообразие внешних устройств;
2. Различия в объемах и скорости передачи информации;
3. Момент передачи информации не определяется программой ЭВМ.

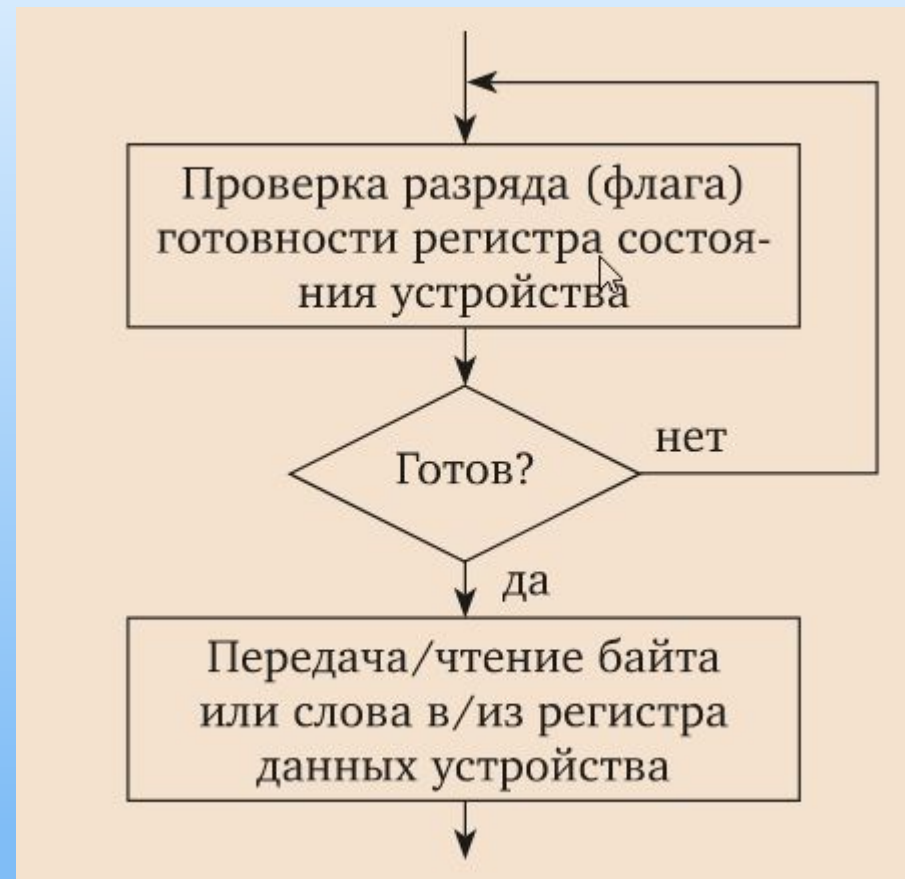
Принципы организации ввода-вывода



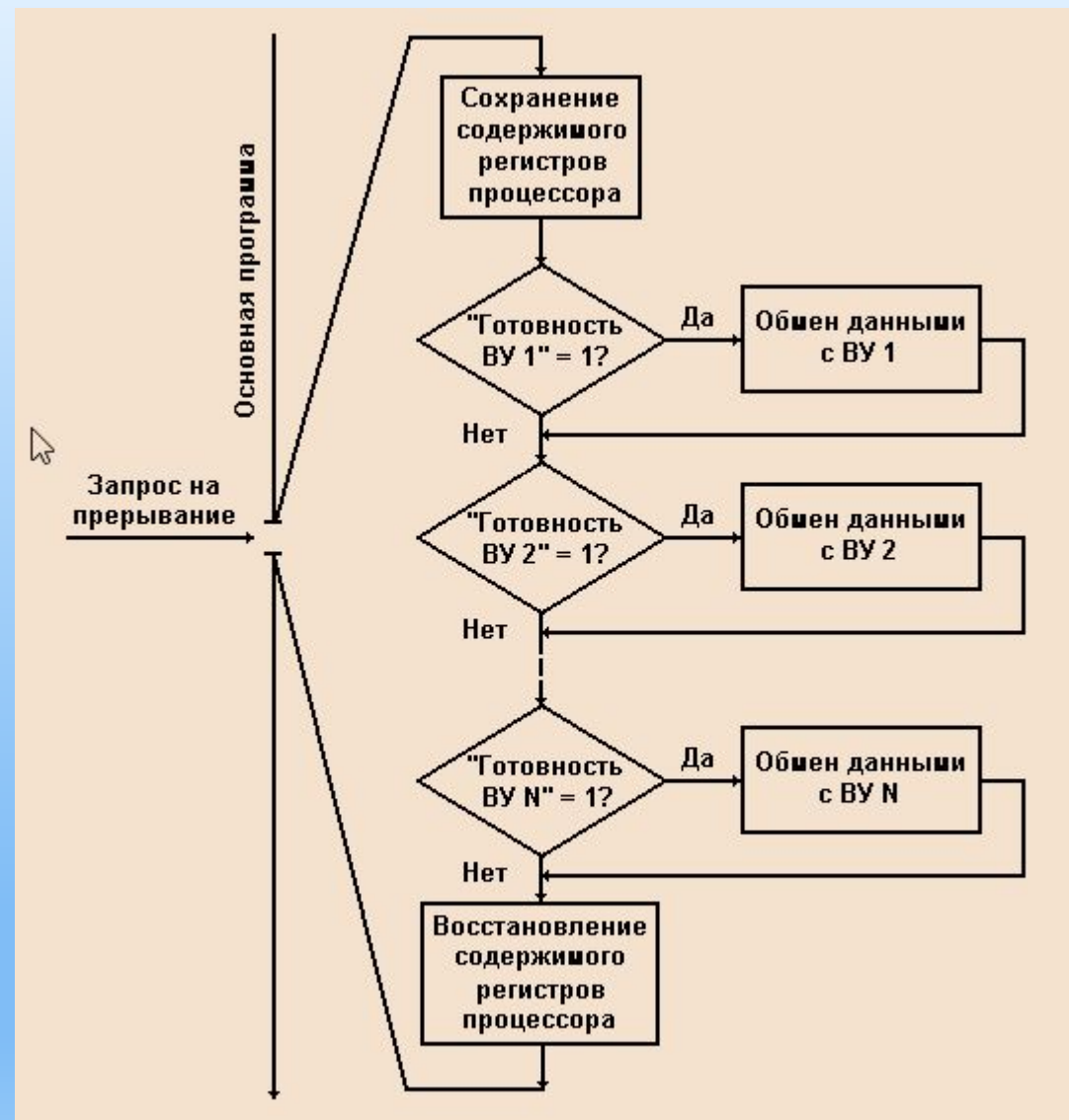
Способы обмена данными с внешними устройствами

1. Обмен данными *с опросом готовности устройства.*
2. Обмен данными *в режиме прерывания программы.*
3. Обмен данными *в режиме прямого доступа к памяти.*

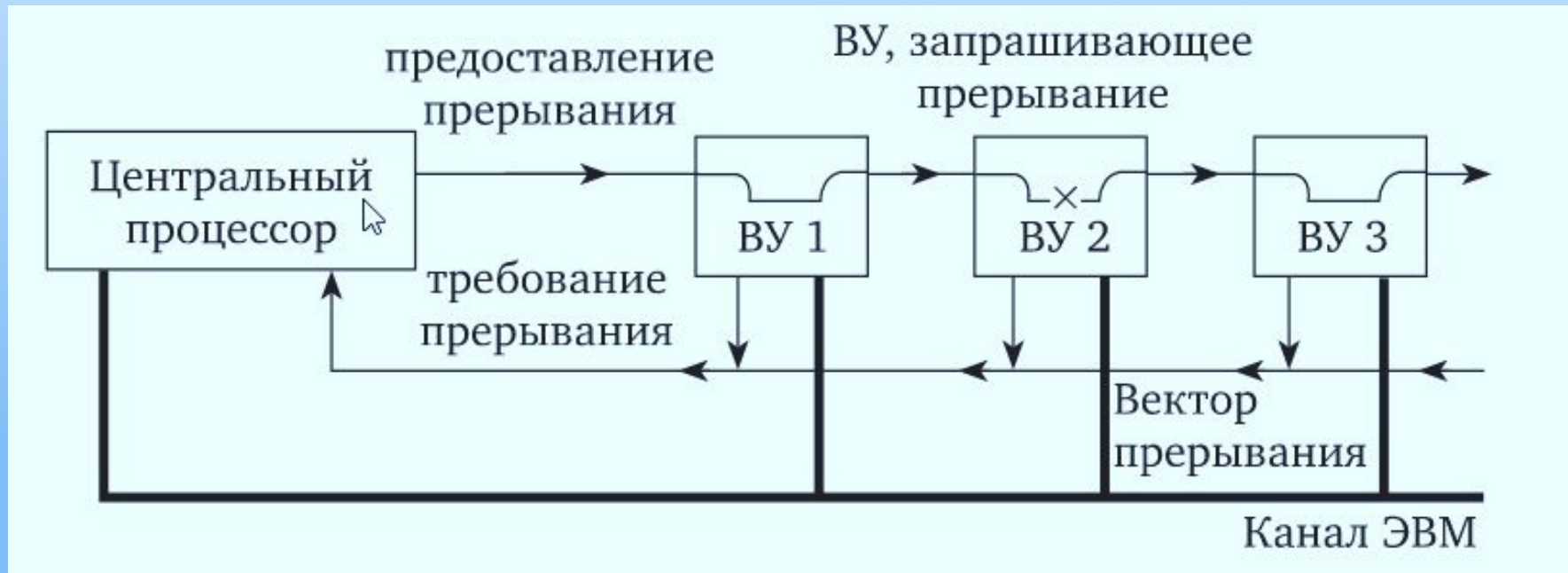
*Обмен данными
с опросом готовности
устройства*



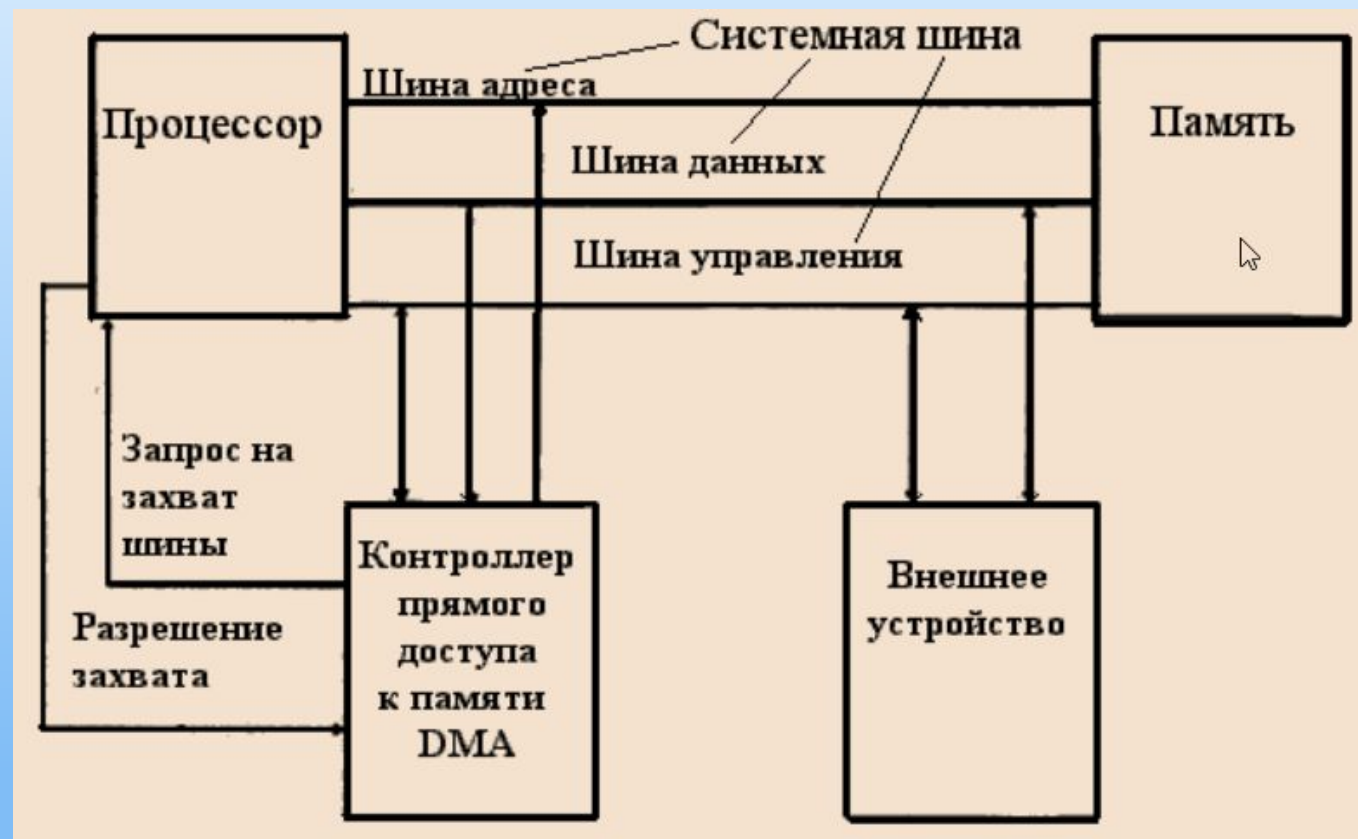
Обмен данными в режиме прерывания программы



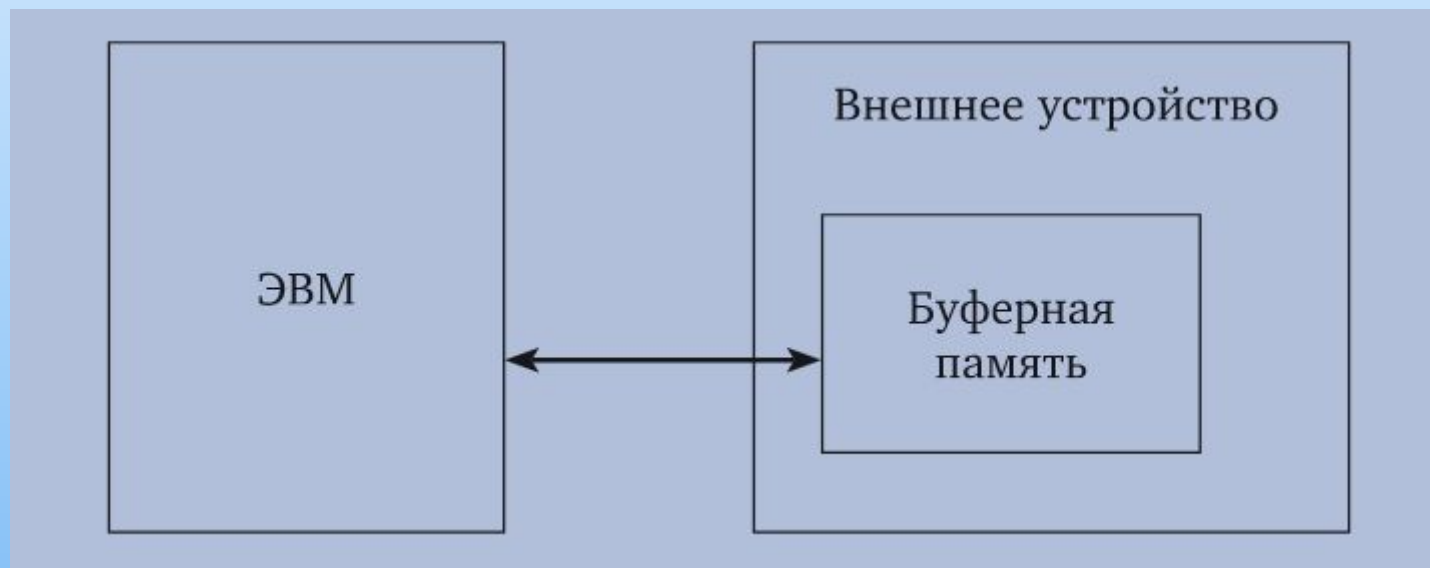
Устранение конфликтов при одновременном требовании прерываний



*Обмен данными с
использованием
прямого доступа
к памяти*

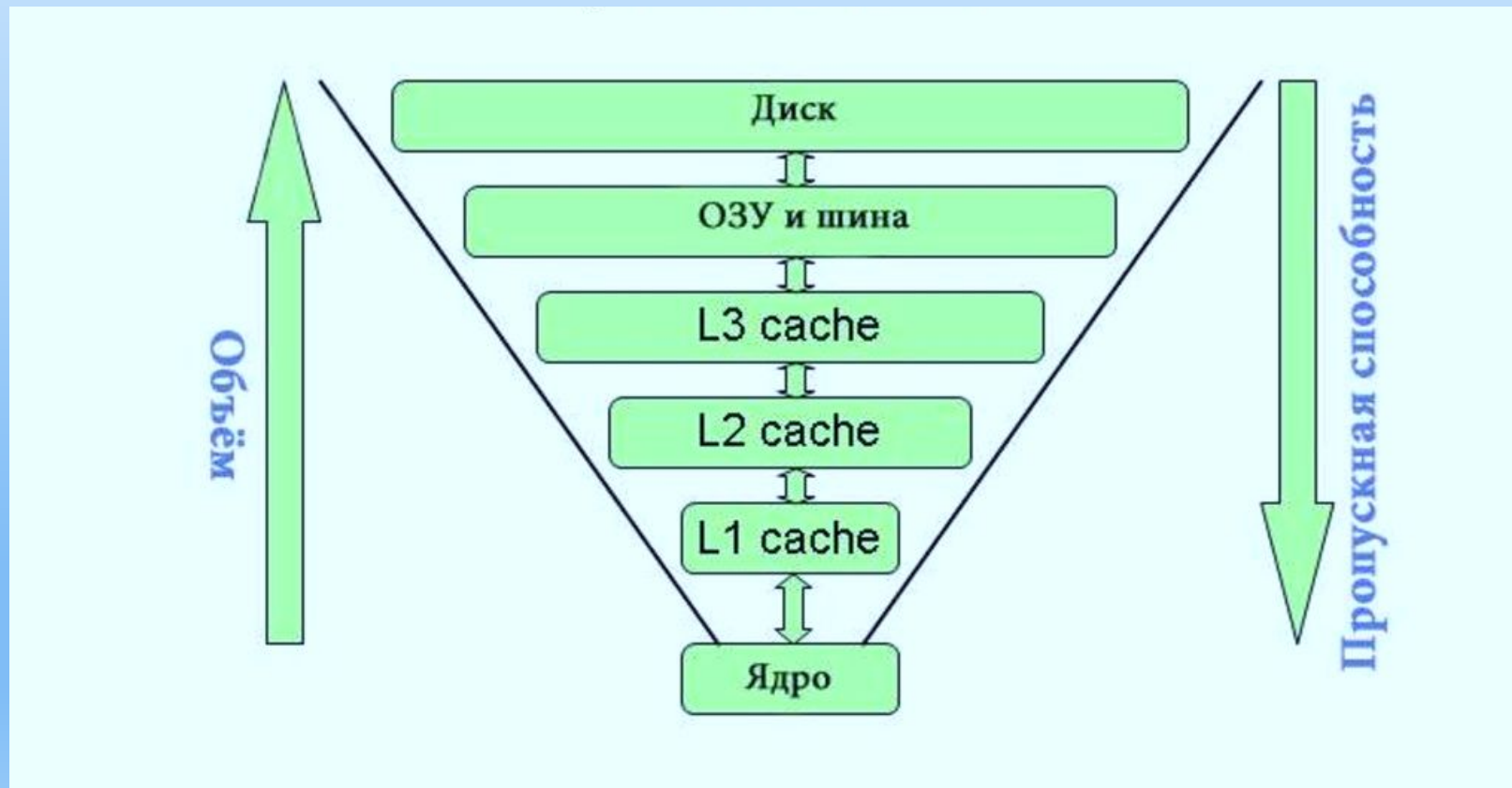


Буферизация данных в системах ввода–вывода

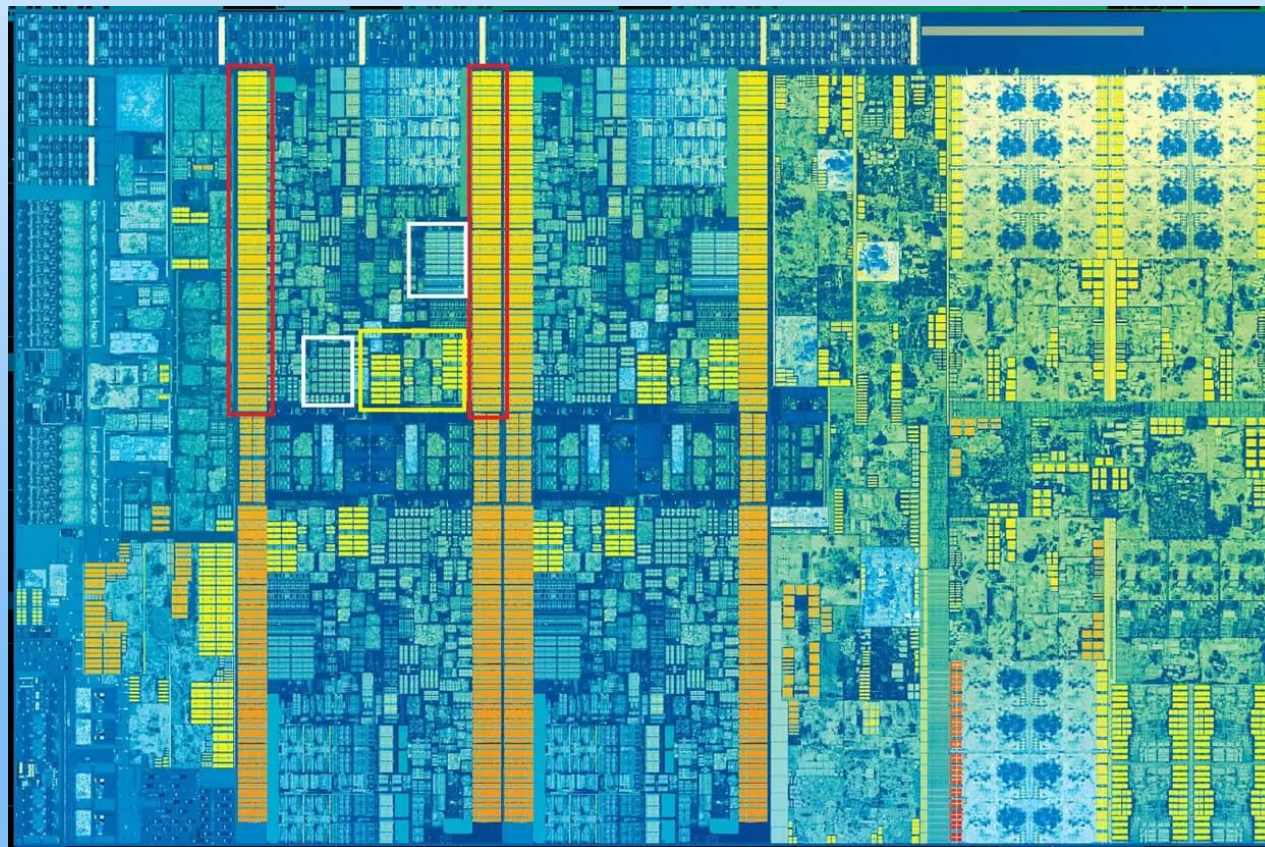


Буферная память – область памяти для запоминания информации при обмене данными между двумя устройствами или процессами с разной скоростью работой.

Согласование пропускных способностей процессора и памяти ЭВМ

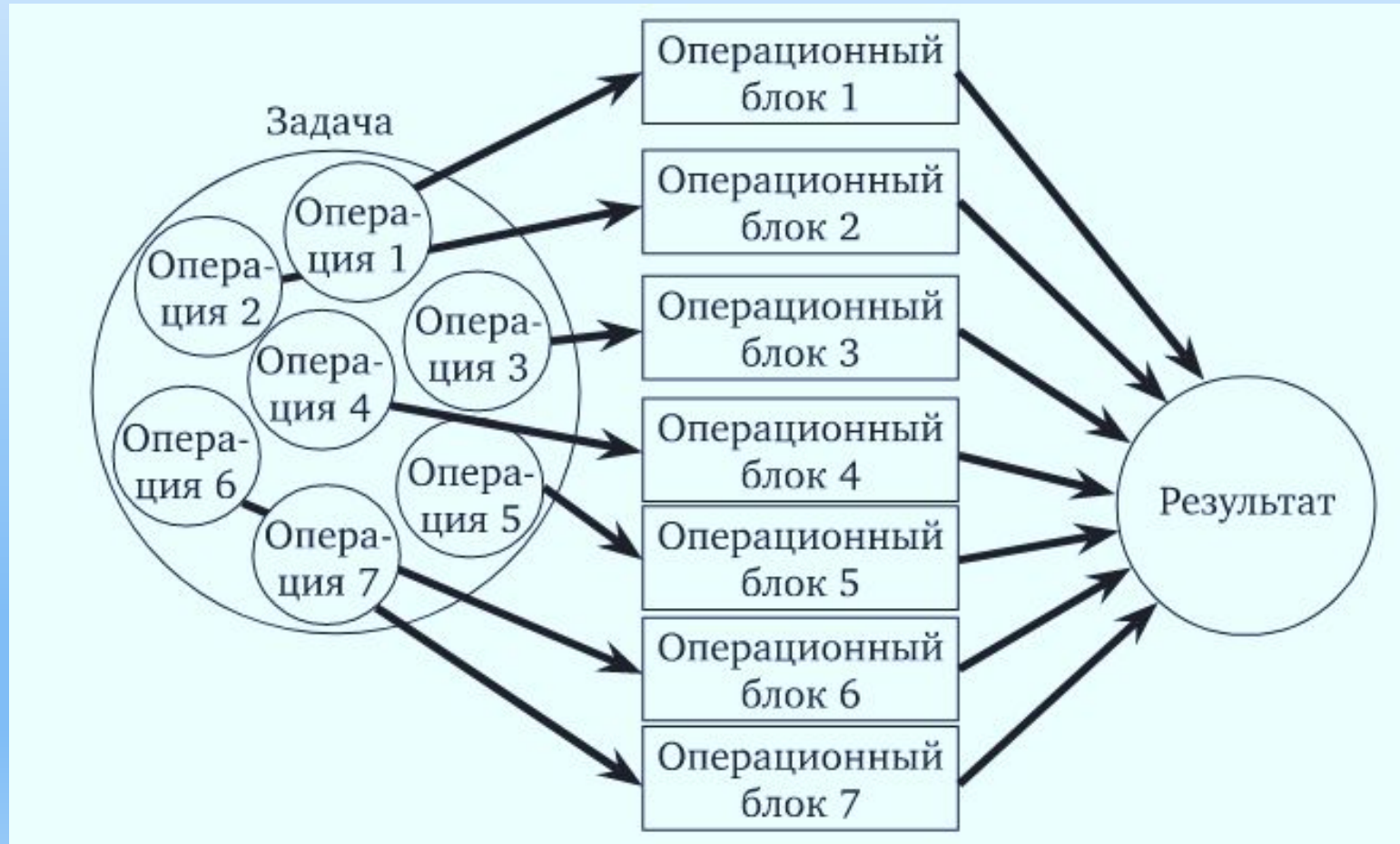


Процессор Intel Kaby Lake

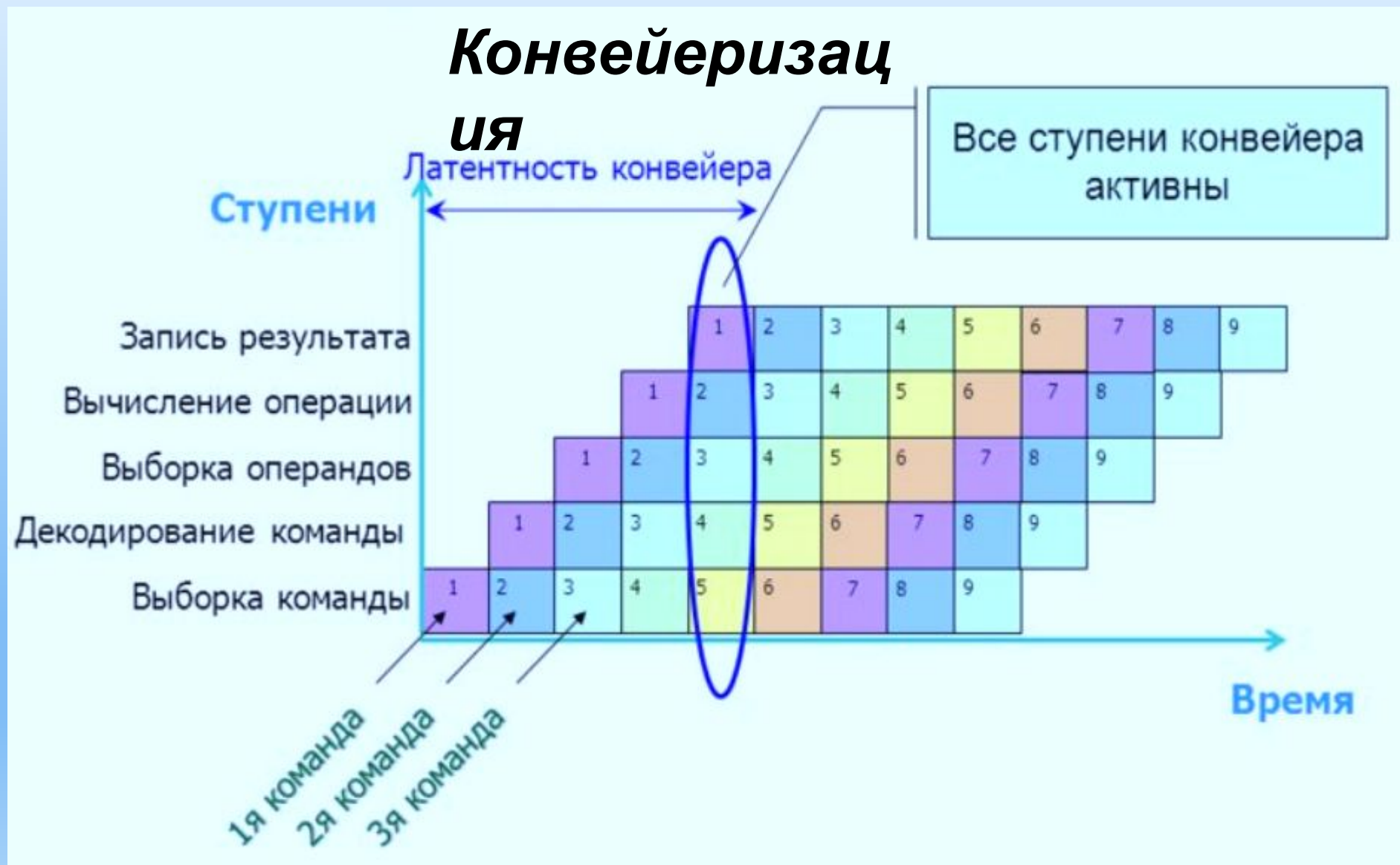


Пути повышения производительности ЭВМ

Распараллеливание



Пути повышения производительности ЭВМ



Перерыв 10 минут

Классификация вычислительных архитектур

Самой ранней и наиболее известной является классификация архитектур вычислительных систем, предложенная в 1966 году **М. Флинном**.

Классификация базируется на понятии *потока* — последовательности элементов, команд или данных, обрабатываемая процессором.

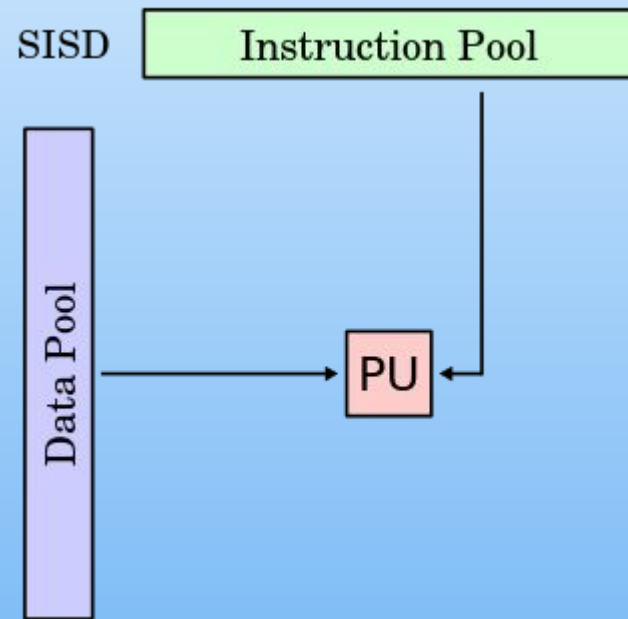
Два основных потока, по которым и производится классификация:

- Поток данных — физические схемы, передающие биты информации данных
- Поток команд — физические схемы, реализующие поток вычислений

Классификация Флинна

	Одиночный поток команд (single instruction)	Множество потоков команд (multiple instruction)
Одиночный поток данных (single data)	SISD (ОКОД)	MISD (МКОД)
Множество потоков данных (multiple data)	SIMD (ОКМД)	MIMD (МКМД)

Одиночный поток команд с одиночным потоком данных

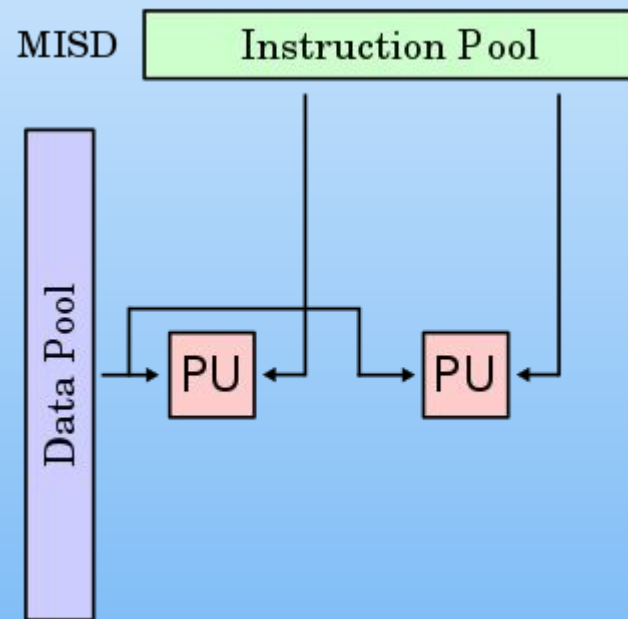


SISD (англ. **S**ingle **I**nstruction stream, **S**ingle **D**ata stream) - традиционный компьютер фон-Неймановской (точнее, Гарвардской) архитектуры с одним процессором, выполняющий последовательно одну команду за другой, работая с одним потоком данных.

Примеры SISD архитектур

- Простейшие встраиваемые системы (игровые автоматы)
- Одноядерные компьютеры и приставки
- Элементы SISD в устройстве самого процессора с кэшем данных и команд
- Элементы внешних устройств (сетевые, звуковые карты)
- Простейшее сетевое оборудование
- Процессоры игровых контроллеров
- Процессоры простых телефонов

Множественный поток команд с одиночным потоком данных

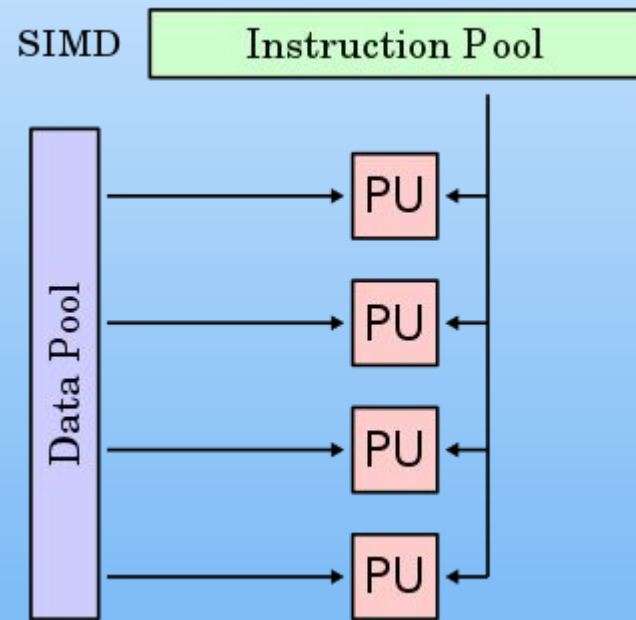


MISD (англ. **M**ultiple **I**nstruction stream, **S**ingle **D**ata stream) — по сути, гипотетический класс, поскольку реальных систем-представителей данного типа пока не существует. Некоторые исследователи относят к нему конвейерные ЭВМ. Отказоустойчивые системы, где несколько процессоров, выполняют один набор команд.

Примеры MISD архитектур

- На космическом корабле Буран многие вычислительные контуры дублировались
- В критически важных железнодорожных приложениях одинаковый расчет производится на нескольких серверах с разным аппаратным обеспечением

Одиночный поток команд с множественным потоком данных



SIMD (англ. **S**ingle **I**nstruction stream, **M**ultiple **D**ata stream, ОКМД) — один из наиболее распространённых типов параллельных ЭВМ. В такой системе имеется несколько одинаковых процессоров, каждый из которых параллельно обрабатывает свои собственные данные по одному и тому же алгоритму.

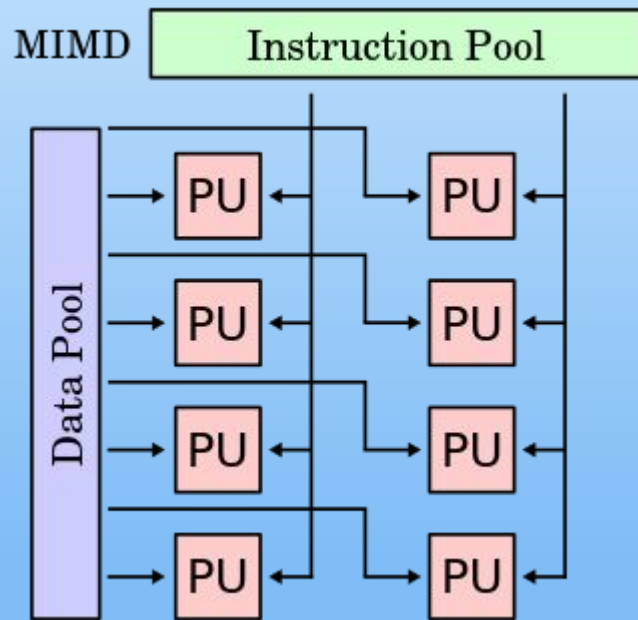
Расширение классификации Флинна

- **SM-SIMD** (shared memory SIMD) — подкласс SIMD с общей памятью. Векторные процессоры и векторные инструкции (SSE, AVX)
- **DM-SIMD** (distributed memory SIMD) — подкласс SIMD с распределённой памятью. Матричные процессоры — особый подвид с большим количеством процессоров. Видеокарты, программируемые вентильные матрицы — типичные представители данного класса.

Примеры устройств

- Встроенные видеопроцессоры в Intel Core старше 3 модели
- Все видеокарты всех производителей и моделей
- Мобильные видеокарты смартфонов
- Векторные расширения процессоров
- Семейство компьютеров системы Cray
- Wii U с процессором AMD RV770, PS 4 с приблизительным аналогом Radeon HD7850/7870, Xbox One с процессором AMD Radeon GPU в составе APU (Accelerated Processing Unit)

Множественный поток команд с множественным потоком данных



MIMD (англ. *Multiple Instruction stream, Multiple Data stream*) — ещё один распространённый тип параллельных ЭВМ. Включает в себя многопроцессорные системы, в которых процессоры обрабатывают множественные потоки данных. Сюда, как правило, относят традиционные мультипроцессорные машины, многоядерные и многопоточные процессоры, а также компьютерные кластеры.

Расширение классификации Флинна

- **SM-MIMD** (shared memory MIMD) — подкласс MIMD с общей памятью. Мультипроцессорные машины и многоядерные процессоры с общей памятью. Мультипроцессоры легко программировать, поддержка SMP (симметричной мультипроцессорности) давно присутствует во всех основных операционных системах.
- **DM-MIMD** (distributed memory MIMD) — подкласс MIMD с распределённой памятью. Многопроцессорные ЭВМ с распределённой памятью и компьютерные кластеры. Локальная память отдельно взятого процессора не видна другим. У каждого процессора своя задача. Если же ему необходимы данные из памяти другого процессора, он обменивается с ним сообщениями.

Примеры архитектур

- Wii U с процессором IBM Power 750 «Espresso»
- PS 4 с процессором AMD Jaguar
- Xbox One с процессором AMD Jaguar
- Большинство процессоров мобильных телефонов: Qualcomm Snapdragon, MediaTek, Apple Aх и так далее
- Семейство Intel Xeon Phi
- Ваш ПК/Ноутбук с процессором _____

Практические задания

Задание 1

Построение модели процессора с архитектурой фон Неймана

Построить процессор с архитектурой фон Неймана, реализующего выполнение простейших команд (чтение, запись, арифметические операции, команда безусловного перехода) в программе Logisim (или любой другой программе для моделирования цифровых электрических схем).

Описать ход выполнения работы, подкрепив скриншотами смоделированных устройств.

Сделать выводы о результатах моделирования.

Задание 2

Влияние уровней памяти на производительность компьютера

Применив заданный программный код (C++), провести временные замеры при использовании различных уровней памяти (кэш, оперативной и внешней).

Построить графики зависимостей времени выполнения операций от объема данных. На каждом из графиков произвести сравнение производительности компьютера:

- при использовании кэш-памяти и ОЗУ (на первом графике);
- при использовании внутренней и внешней памяти (на втором графике).

Объяснить полученные результаты, сделать выводы по каждому графику.

Спасибо за внимание!

Ваши вопросы?