

# Аппаратные средства телекоммуникационных систем

Введение в архитектуру  
процессорных устройств.

# Основные понятия архитектуры процессоров

Аппаратные средства  
телекоммуникационных систем.

Введение в архитектуру  
процессорных устройств.

# Понятие процессор

- **Процессор** – электронный блок либо интегральная схема (микропроцессор), исполняющая машинные инструкции (код программы), главная часть аппаратного обеспечения компьютера или программируемого логического контроллера.
  - *Устройства типа процессор подчинены т.н. «**принципу программного управления**».*
    - *Процесс реализации функции в устройстве описывается в форме алгоритма, называемого **программой**.*



# Принцип программного управления.

- любая функция, является последовательностью элементарных действий – **операций**.
- Каждая операция задается **специальной инструкцией или командой**, служащей для настройки процессора на выполнение заданного элементарного действия;
- Программа описывается в терминах команд и логических условий.
- Программа предварительно размещается в памяти устройства, а не вводится команда за командой в процессе его работы.



# Об машинном коде и языках программирования

- На машинном уровне программа представляет собой набор аппаратно-выполняемых команд – машинный код.
  - Примеры таких команд – сложение, умножение, логическое или, перенос значения из одной ячейки памяти в другую.
- *Каждая команда имеет свой кодовый номер и адреса двух ячеек – данных, для выполнения над ними определенного действия.*
  - Такие данные называются операндами.
- Любая команда программы уровня выше машинного (начиная от ассемблера и до современных абстрактных языков) интерпретируется в машинный код для ее выполнения.

```
00000000 7f 45 4c 46 02 01 01 00 00 00 00 00 00 00 00 00
00000010 02 00 3e 00 01 00 00 00 00 04 40 00 00 00 00 00
00000020 40 00 00 00 00 00 00 00 70 11 00 00 00 00 00 00
00000030 00 00 00 00 40 00 38 00 09 00 40 00 1e 00 1b 00
00000040 06 00 00 00 05 00 00 00 40 00 00 00 00 00 00 00
00000050 40 00 40 00 00 00 00 00 40 00 40 00 00 00 00 00
00000060 f8 01 00 00 00 00 00 00 f8 01 00 00 00 00 00 00
00000070 08 00 00 00 00 00 00 00 03 00 00 00 04 00 00 00
00000080 38 02 00 00 00 00 00 00 38 02 40 00 00 00 00 00
00000090 38 02 40 00 00 00 00 00 1c 00 00 00 00 00 00 00
000000a0 1c 00 00 00 00 00 00 00 01 00 00 00 00 00 00 00
000000b0 01 00 00 00 05 00 00 00 00 00 00 00 00 00 00 00
000000c0 00 00 40 00 00 00 00 00 00 00 40 00 00 00 00 00
000000d0 ac 06 00 00 00 00 00 00 ac 06 00 00 00 00 00 00
000000e0 00 00 20 00 00 00 00 00 01 00 00 00 06 00 00 00
000000f0 10 0e 00 00 00 00 00 00 10 0e 60 00 00 00 00 00
00000100 10 0e 60 00 00 00 00 00 28 02 00 00 00 00 00 00
00000110 30 02 00 00 00 00 00 00 00 00 20 00 00 00 00 00
00000120 02 00 00 00 06 00 00 00 28 0e 00 00 00 00 00 00
00000130 28 0e 60 00 00 00 00 00 28 0e 60 00 00 00 00 00
```

# Принцип построения ЭВМ

- Устройство, объединяющее процессор и периферийные модули называется электронно-вычислительной машиной (ЭВМ)

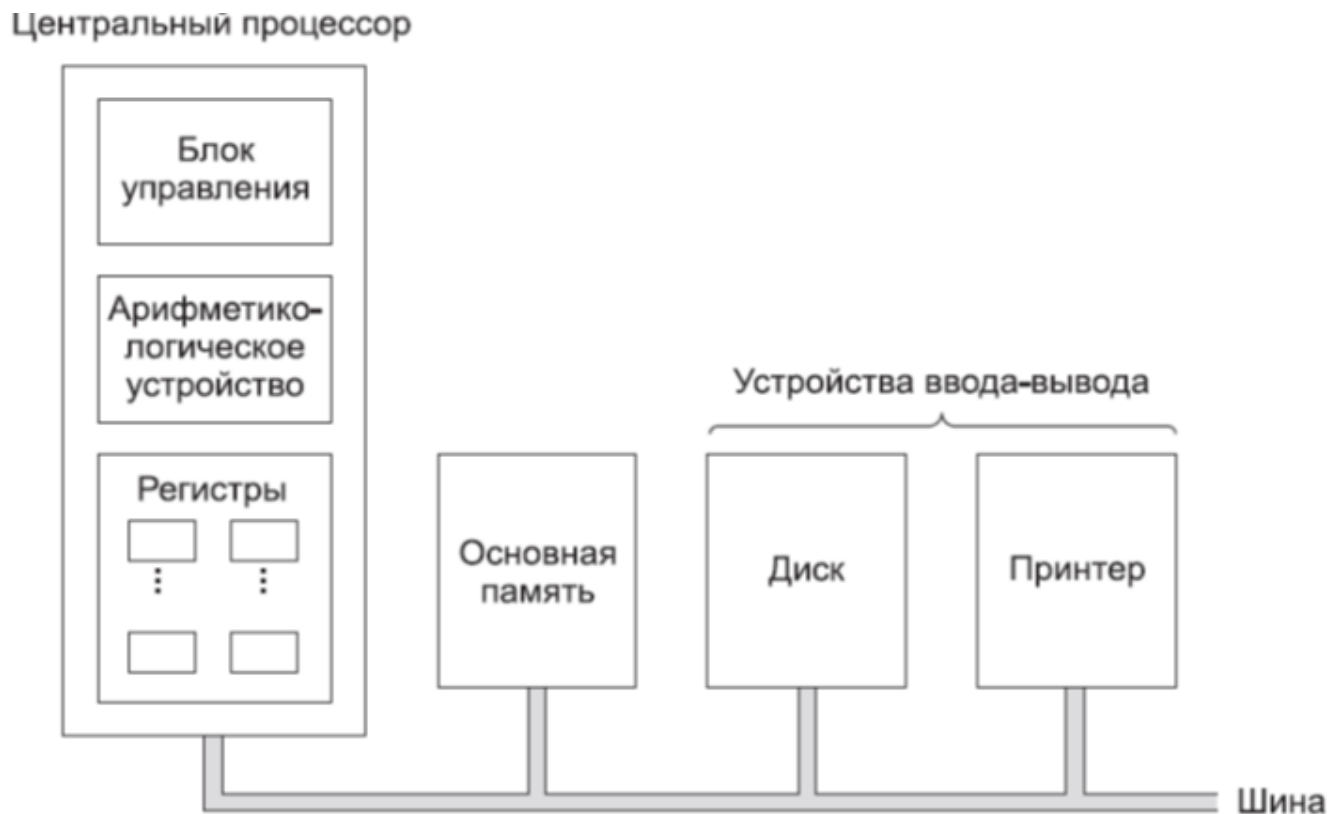


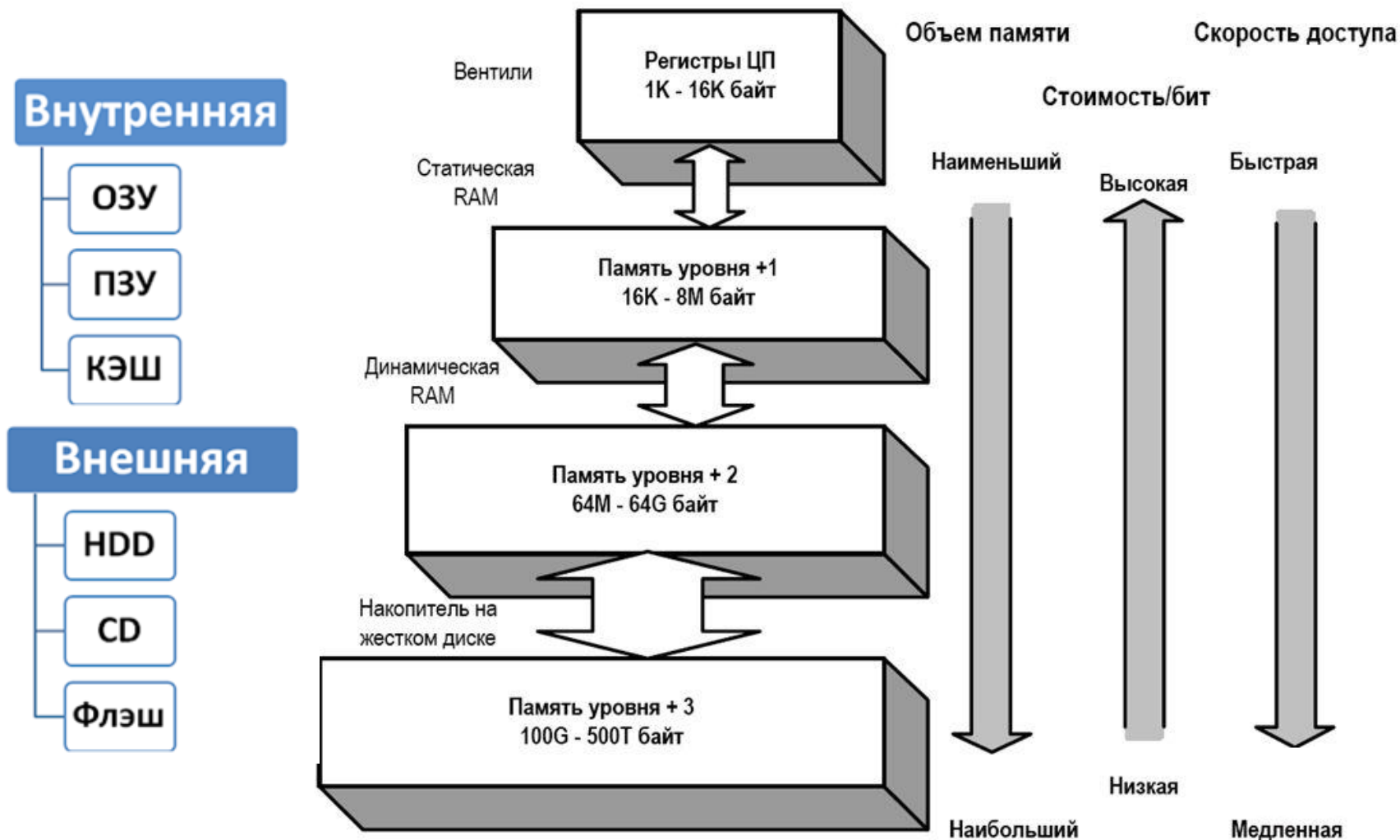
Схема компьютера с одним центральным процессором  
и двумя устройствами ввода-вывода

# Принцип построения ЭВМ

- **Центральный процессор (АЛУ с блоком устройства управления (УУ))** обладает принцип последовательной передачи управления.
- Набор арифметических, логических и прочих инструкций **АЛУ** насчитывает несколько сотен,
- Процессор имеет **набор регистров в устройстве управления (УУ)**
  - часть регистров доступна для хранения операндов, выполнения действий над ними и формирования адреса инструкций и операндов в памяти.
  - Другая часть регистров используется процессором для служебных (системных) целей,  
доступ к ним может быть ограничен (есть даже программно-невидимые регистры).
- **Все компоненты компьютера представляются для процессора в виде наборов ячеек памяти или/и портов ввода-вывода,**
  - В ячейки и порты в-в процессор может записывать и/или считывать содержимое.
- *Процессор (один или несколько), память и необходимые элементы, связывающие их между собой и с другими устройствами, называют центральной частью, или ядром, компьютера (или просто центром).*

# Иерархия памяти в компьютере

Самая важная характеристика памяти – латентность – время доступа к ячейки памяти





# Иерархия памяти в компьютере

- **Оперативная память (ОЗУ)** динамическая память с произвольным доступом.
  - Оперативная память вместе с кэшем всех уровней (в настоящее время — до трех) представляет собой единый массив памяти, доступный процессору для записи и чтения данных.
- **Постоянная память (ПЗУ)**, из нее можно только считывать команды и данные
- Также ЭВМ имеет некоторые виды специальной памяти (например, видеопамять графического адаптера).
- В любом компьютере есть **энергонезависимая память**, в которой хранится программа начального запуска компьютера и минимально необходимый набор сервисов (FLASH, ROM BIOS).
  - Доступ к внутренней памяти осуществляется по одномерному (линейному) адресу, который представляет собой двоичное число. Доступна для процессора.

# Иерархия памяти в компьютере

- **Внешняя память** каждая ее ячейка имеет свой адрес внутри *блока*, который, имеет многомерный адрес и может быть считан или записан только целиком.
  - В случае дискового накопителя физический адрес блока является трехмерным — он состоит из номера поверхности (головки), номера цилиндра и номера сектора, но виртуально линейным номером — логическим, адресом блока, а его преобразованием в физический адрес занимается внутренний контроллер накопителя

# Периферийные устройства ЭВМ

- **Устройства хранения данных** (устройства внешней памяти) — дисковые (магнитные, оптические, магнитооптические), твердотельные (карты, модули и флэш-память). Эти устройства используются для сохранения информации, на энергонезависимых носителях и загрузки этой информации в оперативную память.
- **Устройства ввода-вывода** служат для преобразования информации из внутреннего представления компьютера (биты и байты) в форму, понятную окружающим, и обратно. Под окружающими подразумеваются человек (и другие биологические объекты) и различные технические устройства
- **Коммуникационные устройства** служат для передачи информации между компьютерами и/или их частями. Сюда относят модемы (проводные, радио, оптические, инфракрасные...), адаптеры локальных и глобальных сетей.
- **Консоль.** Консолью компьютера называют его «выступающую часть», обращенную к пользователю. В РС стандартной консолью являются клавиатура (устройство ввода) и дисплей

# Классификация ЭВМ

- **Персональные ЭВМ**
  - Настольные персональные компьютеры.
  - Ноутбуки и нетбуки.
  - Однопалатные микрокомпьютеры.
  - Планшетные устройства и смартфоны.
  - Компьютеризированные устройства: фотоаппараты, mp3 плееры, диктофоны, игровые приставки.
- **Серверы:** промышленные серверы, Серверы на базе персональных компьютеров.
- **приемо-передающие устройства:** модемы, точки беспроводного и проводного доступа, устройства беспроводной связи.
- **Межсетевые узлы:** концентраторы, коммутаторы, мосты, шлюзы, маршрутизаторы, межсетевые экраны.
- **Устройства специального назначения.**
  - Бортовые компьютерные системы.
  - Встроенные системы.
  - Диагностические устройства.
  - Контрольно-кассовые аппараты.

# Классификация процессоров по видам

- **Центральные процессоры (CPU).** – пример CPU ПК.
- **Универсальные микропроцессоры** используются для построения вычислительных машин и систем связи. Такие компьютеры называются контроллерами. (пример Raspberry Pi, Siemens).
- **Микроконтроллеры (МК)** используются для управления малогабаритными и дешёвыми устройствами управления и связи. Они раньше назывались однокристальными микроЭВМ. В микроконтроллерах, в отличие от универсальных микропроцессоров, максимальное внимание уделяется именно габаритам, стоимости и потребляемой энергии.
- **Сигнальные процессоры (DSP)** используются для решения задач обработки сигналов. Аппаратная реализация сложных математических операций.
- **Медийные процессоры** – гибриды DSP и универсальных процессоров и предназначены для обработки аудио сигналов, графики, видеоизображений, а также для решения коммуникационных задач в мультимедиа-компьютерах, игровых приставках, бытовой технике и т.д.

# Примеры сопроцессоров

- **Основные виды сопроцессоров:**

- *Математические сопроцессоры (FPU) -операции с плавающей запятой (имеют 2 ЛУ для мантиссы и экспоненты);*
- *навигационные (с GPS);*
- *Графические (многоядерные, много АЛУ, мало команд другого профиля) ориентация на рендеринг – расчет текстуры по модели;*
- *Коммуникационные (поддержка сетевых интерфейсов и протоколов). Например (Ethernet, или беспроводных, например WiFi и GPRS)*

# Особенности архитектуры процессоров

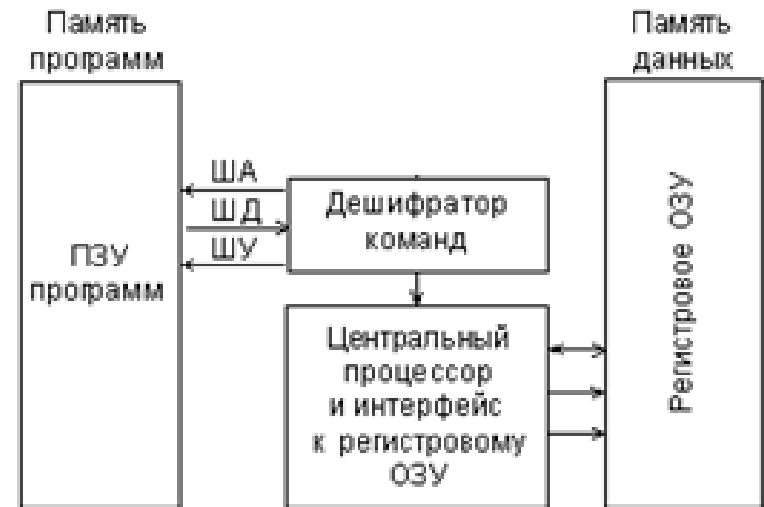
Аппаратные средства  
телекоммуникационных систем.

Введение в архитектуру  
процессорных устройств.

# Архитектуры процессоров.

## Виды архитектура процессоров

- **Гарвардская архитектура**  
предполагает раздельное использование памяти программ и данных.
- Обычно используют для повышения быстродействия системы за счёт разделения путей доступа к памяти программ и данных.
- Большинство специализированных микропроцессоров (особенно микроконтроллеры) имеют данную архитектуру.
- **Архитектура фон Неймана** – предполагает хранение программ и данных в общей памяти.
- Наиболее характерна для процессоров, ориентированных на использование в компьютерах.
- Примером могут служить микропроцессоры семейства x86.





# Архитектуры процессоров.

## Особенности гарвардской архитектуры

- Применение отдельной небольшой по объему *памяти данных* сокращает длину *команд* и ускоряет поиска информации в *памяти данных*. Использование единого адресного пространства приводит к увеличению формата *команд* за счет увеличения числа разрядов для адресации операндов.
- Гарвардская архитектура обеспечивает **потенциально более высокую скорость выполнения программы** по сравнению с фон-неймановской за счет возможности реализации параллельных операций. **Выборка следующей команды может происходить одновременно с выполнением предыдущей**, и нет необходимости останавливать процессор на время выборки команды.
- Этот метод реализации операций позволяет обеспечивать выполнение различных команд за одинаковое число тактов, что дает возможность более просто определить время выполнения циклов и критичных участков программы.
- Большинство производителей современных микроконтроллеров используют гарвардскую архитектуру.

# Архитектуры процессоров.

## Архитектура Фон-Неймана

- Упрощение устройства процессора, - обращение к одной общей памяти.
- Единая область памяти позволяет оперативно перераспределять ресурсы между областями программ и данных, что повышает гибкость системы.
- **Архитектура фон Неймана последовательная.** Выполняемые действия определяются блоком управления и АЛУ. Центральный процессор выбирает и исполняет команды из памяти последовательно, адрес очередной команды задается «счетчиком адреса» в блоке управления.
- Часто в процессоры встроены **сопроцессоры**, имеющие преимущества при решении определённого рода задач (например, для операций с плавающей запятой).



Блок-схема архитектуры центрального процессора

# Архитектуры процессоров. Магистральная организация процессов.

- **Магистраль или шина (Bus)** – группа линий передачи информации, объединенных общей функцией.
- В общем случае у процессору требуется 3 шины – шина адреса, шина данных и шина управления.
- Для снижения общего количества линий связи магистрали **часто применяется мультиплексирование шин адреса и данных** в разные моменты времени. Для фиксации этих моментов (стробирования) служат специальные сигналы на *шине управления*.
- **Шина управления (инструкций)** — это вспомогательная шина по которой передаются **управляющие и служебные сигналы**. Также сигналы с **внешних и внутренних источников**.
- **Основные функции шины управления - вызов прерываний.**
- На пример, в момент ввода с клавиатуры или достижение определенного значения внутреннего таймера. Предполагается выполнение определённых действий по сигналам прерываний.



# Архитектуры процессоров. Кэш память.

- Внутри центрального процессора находится быстрая память небольшого объема для хранения промежуточных результатов и некоторых команд управления – **КЭШ** память.
- Кэш может быть многоуровневый.
  - Часто двухуровневый, для хранения команд и отдельно для хранения данных или трех уровней с дополнительным уровнем для работы между ядрами.
- Операции чтения и записи с регистрами выполняются очень быстро, поскольку они находятся внутри центрального процессора.
- Кэш-память позволяет держать наиболее часто используемые слова внутри центрального процессора и избегать (медленных) обращений к основной памяти.
  - скорость работы процессора выше скорости операции обращения к памяти и получения от туда данных.
  - За работу Кэш память отвечает специальный контроллер внутри процессора.



# Архитектуры процессоров. Регистровая память.

- Каждый процессор для обеспечения гибкости работы имеет набор регистров, отвечающих за определенные настройки процессора (такие как, например, тактовая частота) (**Регистры**).
- Один из самых главных регистров – это **счетчик команд**, в нем указывается, какая по счету последовательная команда должна быть выполнена в настоящее время.
- Также в процессоре имеются регистры, содержащие код текущей команды и регистры операндов для текущей команды.

## Регистры данных

<b>AH</b>	<b>AL</b>	Аккумулятор
<b>BH</b>	<b>BL</b>	Базовый регистр
<b>CH</b>	<b>CL</b>	Счетчик
<b>DH</b>	<b>DL</b>	Регистр данных

## Регистры-указатели

<b>SI</b>	Индекс источника
<b>DI</b>	Индекс приемника
<b>BP</b>	Указатель базы
<b>SP</b>	Указатель стека

## Сегментные регистры

<b>CS</b>	Регистр сегмента команд
<b>DS</b>	Регистр сегмента данных
<b>ES</b>	Регистр дополнительного сегмента данных
<b>SS</b>	Регистр сегмента стека

## Прочие регистры

<b>IP</b>	Указатель команд
<b>FLAGS</b>	Регистр флагов

# Архитектуры процессоров. Устройство управления

- **Функции устройства управления (УУ)**
- формирует адрес команды, которая должна быть выполнена
- выдает управляющий сигнал на чтение содержимого соответствующей ячейки запоминающего устройства (ЗУ).
  - Считанная команда передается в УУ.
- По информации, содержащейся в адресных полях команды, УУ формирует адреса операндов и управляющие сигналы для их чтения из ЗУ и передачи в арифметико-логическое устройство (АЛУ).
  - После считывания операндов УУ по коду операции, содержащемуся в команде, выдает в АЛУ сигналы на выполнение операции.
- Полученный результат записывается в ЗУ по адресу приемника результата.
- Признаки результата (знак, наличие переполнения, признак нуля и так далее) поступают в УУ, где записываются в специальный регистр признаков.
- Эта информация может использоваться при выполнении следующих команд программы, например команд условного перехода.

Регистры данных		Регистры-указатели	Сегментные регистры	Прочие регистры
АH	АL	SI	CS	IP
ВH	ВL	DI	DS	FLAGS
СH	СL	BP	ES	
ДH	ДL	SP	SS	

# Архитектуры процессоров.

## Арифметико-логическое устройство (ALU)

- объединяет различные арифметические и логические операции в одном узле. Например, типичное АЛУ может выполнять сложение, вычитание, сравнение величин, операции «И» и «ИЛИ».

АЛУ имеет два регистра операндов

Результат работы АЛУ может быть подан на шины данных или обратно в АЛУ

АЛУ имеет ряд флаг, соответствующих определённым событиям, например переполнению.

Часто к АЛУ добавляют сопроцессор для работы с числами с плавающей запятой



# Архитектуры процессоров.

## Организация наборов команд процессоров.

**Программа** размещена в памяти команд (ПК). После запуска устройство управления (УУ) начинает выполнять три действия:

- 1) последовательную выборку команды из памяти команд;
  - 2) декодирование (интерпретацию) кода команды;
  - 3) выполнение операции, соответствующей команде в устройстве обработки (ОУ).
- **Команда или инструкция** (Command, Instruction) – двоичный код, служащий для настройки программно-управляемого устройства на выполнение заданной операции.
  - **Система команд** (Command set) – совокупность всех команд, допустимых для данного программного управляемого устройства.
  - **Программа** (Program) – последовательность инструкций (команд) и логических условий, реализующих заданный алгоритм.
  - **По типам команд процессоры делят на:**
    - **CISC** (Complex Instruction Set Computing) с полным набором команд;
    - **RISC** (Reduced Instruction Set Computing) с сокращенным набором команд;
    - **MISC** (Minimal Instruction Set Computer) с минимальным набором команд;
    - **VLIW** (Very Long Instruction Word) (одна команда выполняется параллельно на нескольких процессорах).



# Архитектуры процессоров.

## CISC система команд

CISC (англ. Complex Instruction Set Computer — «компьютер с полным набором команд») — полная система команд, подразумевает, что все необходимые для машинного языка команды выполняются на аппаратном уровне.

Самый яркий пример CISC архитектуры — это x86 (он же IA-32) и x86\_64 (он же AMD64).

- нефиксированная длина команд,
- небольшое число регистров, многие из которых выполняют строго определенную функцию.
- одна команда может быть заменена ей аналогичной, либо группой команд, выполняющих ту же функцию.
- CSIC и RISC процессоры несовместимы.
- CISC система команд исторически появилась первой, по этому большинство процессоров 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.
  - на выполнение даже самой короткой команды из системы CISC обычно тратится 4 такта.

# Архитектуры процессоров.

## RISC система команд

RISC (англ. Reduced Instruction Set Computer — «компьютер с сокращённым набором команд») — архитектура процессора, в котором быстродействие увеличивается за счёт упрощения инструкций декодирование становится более простым, а время выполнения — меньшим.

Первые RISC-процессоры не имели даже инструкций умножения и деления и не поддерживали работу с числами с плавающей запятой.

Примеры RISC-архитектур: PowerPC, серия архитектур ARM (ARM7, ARM9, ARM11, Cortex).

- архитектура имеет постоянную длину команды,
  - Позволяет работать параллельно и конвейером (то есть выполнять больше одной команды за один такт)
- меньшее количество схожих инструкций,
- большее количество регистров.
- содержат набор только простых, чаще всего встречающихся в программах команд (по правилу 20-80)
- Основной недостаток RISC архитектуры — необходимость моделирования сложных команд.
  - Сборка сложных команд производится автоматическая из простых.

# Архитектуры процессоров.

## MISC система команд

MISC (англ. Minimal Instruction Set Computer — «компьютер с минимальным набором команд»).

- более простая архитектура чем RISC, используемая в первую очередь для большего уменьшения итоговой цены и энергопотребления процессора.
- Архитектура MISC строится на стековой вычислительной модели с ограниченным числом команд (примерно 20—30 команд).
  - Может содержать в себе блок RISC, обрабатывающий в себе от 10 базовых команд (+, —, /, \*, if, else & etc), из которых формируются более сложные операции над значениями, методом ветвления полученных результатов в ПЗУ.
- Используется в IoT-сегменте и недорогих компьютерах, например, роутерах.
- Недостаток - сложность написания программ под различные процессоры.
  - Все нюансы по подбору методов вычисления и оптимизаций возлагались на плечи программистов.

# Архитектуры процессоров.

## VLIW система команд

VLIW (англ. Very Long Instruction Word — «очень длинная машинная команда») — архитектура процессоров с несколькими вычислительными устройствами

Архитектура VLIW в терминах Intel называется EPIC (на самом деле EPIC имеет отличия в организации параллелизма).

Примеры архитектуры: Intel Itanium (серверные процессоры Intel Core, архитектура IA-64), Эльбрус-3.

- одна инструкция процессора содержит несколько операций, которые должны выполняться параллельно.
  - По сути является архитектурой CISC со своим аналогом спекулятивного исполнения команд, спекуляция выполняется во время компиляции.
- Компиляторы для процессоров этой архитектуры сильно привязаны к конкретным процессорам.
  - Например, в следующем поколении максимальная длина «очень длинной команды» может из условных 256 бит стать 512 бит, и исчезнет совместимость.
  - Ключевым отличием от суперскалярных CISC-процессоров является то, что для них загрузкой исполнительных устройств занимается часть процессора (планировщик), а загрузкой вычислительных устройств для VLIW-процессора занимается компилятор, на что отводится существенно больше времени (качество загрузки и, соответственно, производительность теоретически выше).

# Архитектуры процессоров.

## Архитектуры по степени параллелизма

Классификация Флинна

SISD – скалярные процессоры

MISD – конвейерная архитектура

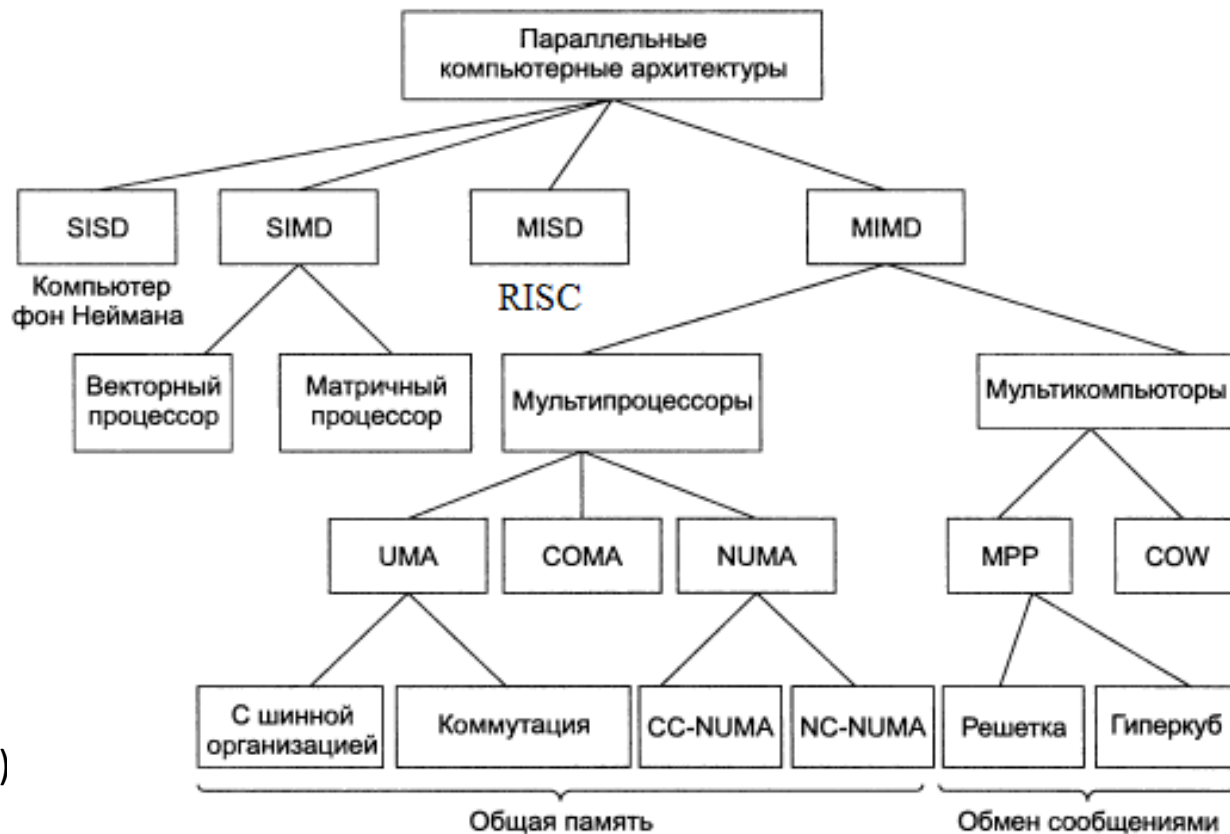
SIMD – векторная и суперскалярная архитектуры

MIMD – многопроцессорные и многомашинные архитектуры

UMA – архитектура с однородным доступом к памяти (с симметричным)

NUMA - с неоднородным доступом к памяти

COMA - с доступом только к кэш-памяти



MPP – процессоры с массовым параллелизмом

COW – кластеры рабочих станций.

# Архитектуры процессоров.

## Архитектуры по степени параллелизма

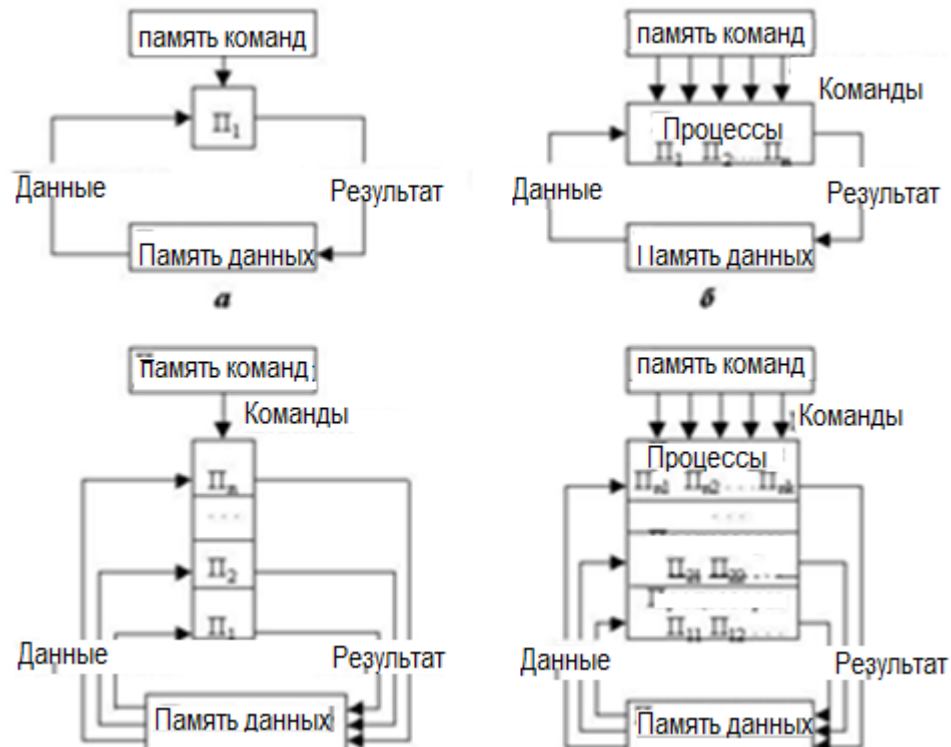
Классификация Флинна

SISD – скалярные процессоры

MISD – конвейерная архитектура

SIMD – векторная и  
суперскалярная архитектуры

MIMD – многопроцессорные и  
многомашинные архитектуры



- а - SISD (однопроцессорная), б - MISD (конвейерная);  
в - SIMD (векторная); г - MIMD (матричная)

# Архитектуры процессоров.

## Конвейерная архитектура

- Исполнение RISC команд за один такт позволяет выполнять их конвейером.
  - Например 3-х ступенчатый конвейер, за один такт происходят операции выполнения команды, считывания операндов следующей(2-й) команды и считывания кода следующей через одну (3-й) команды.
  - Другой пример 5 ступенчатый конвейер

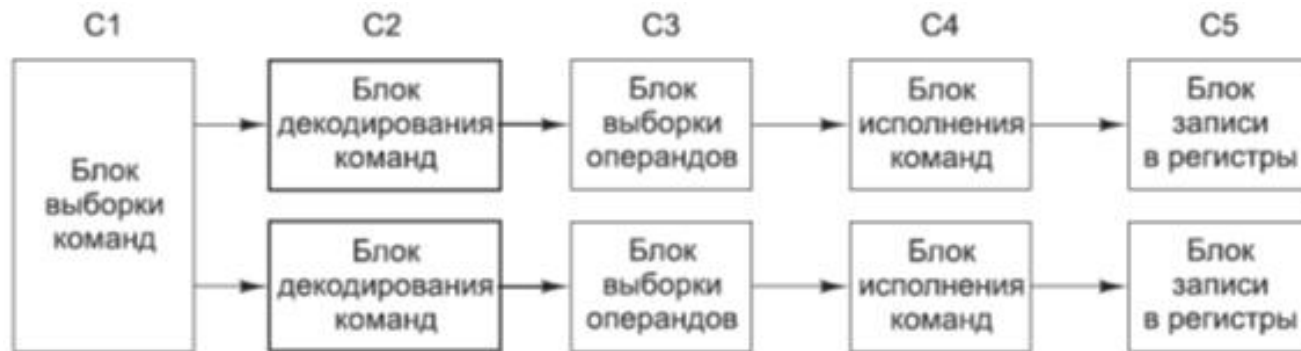


Пятиступенчатый конвейер (a); состояние каждой ступени в зависимости от количества пройденных циклов (b). Показано 9 циклов

# Архитектуры процессоров.

## Конвейерная архитектура

- В современных процессорах используются суперскалярные архитектуры на основе конвейерных.
- Имеют общий блок выборки команд, которые вызывает из памяти сразу по несколько команд и помещает каждую из них в один из конвейеров.
- Каждый конвейер содержит АЛУ.
  - параллельные команды не должны конфликтовать из-за ресурсов (например, регистров) и ни не должны зависеть от результата друг друга.
- Могут быть главный и зависимый конвейеры (дополнительный)
- Команды всегда выполнялись по порядку. Специальные компилятор объединяет совместимые команды в пары.



Сдвоенный пятиступенчатый конвейер с общим блоком выборки команд



# Архитектуры процессоров.

## Конвейерная архитектура

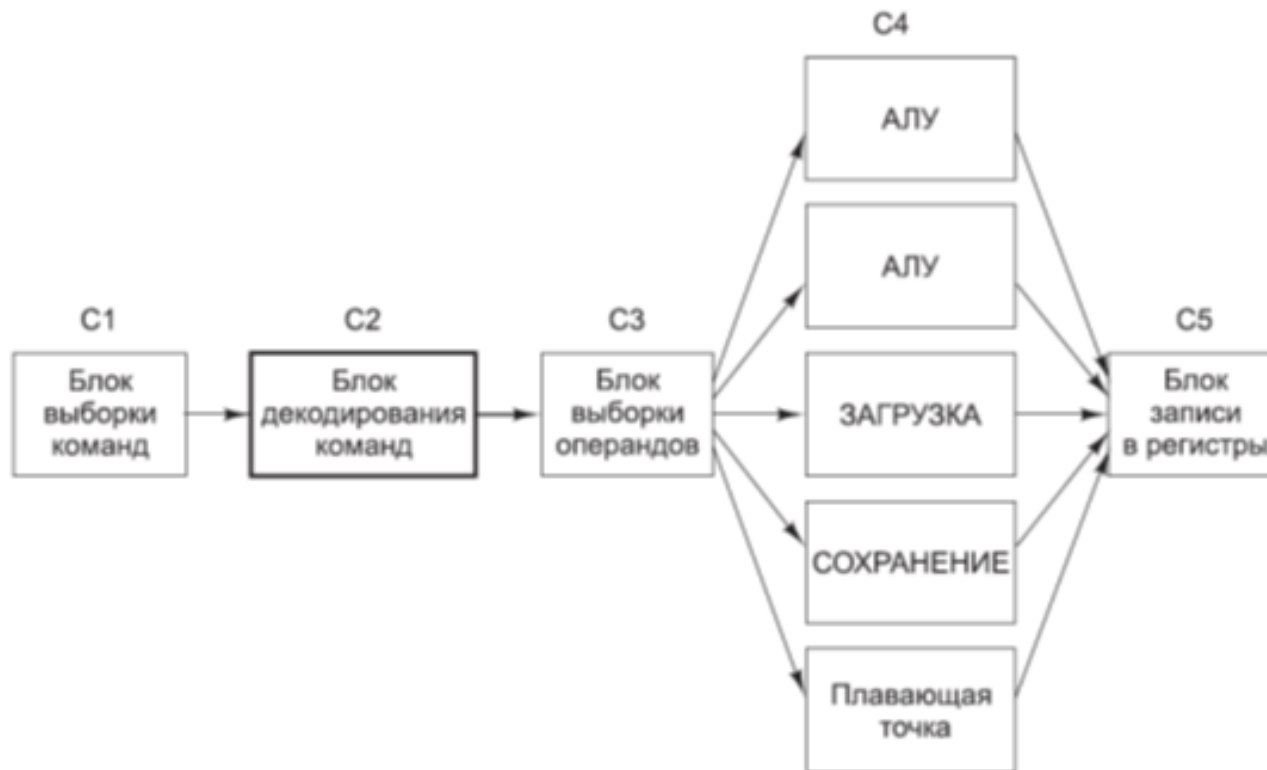
- В современных процессорах используются многоконвейерные архитектуры. Имеют общий блок выборки команд, которые вызывают из памяти сразу по несколько команд и помещает каждую из них в один из конвейеров.
- Каждый конвейер содержит АЛУ.
  - параллельные команды не должны конфликтовать из-за ресурсов (например, регистров) и ни не должны зависеть от результата друг друга.
- Могут быть главный и зависимый конвейеры (дополнительный)
- Команды всегда выполнялись по порядку. Специальные компилятор объединяет совместимые команды в пары.



# Архитектуры процессоров.

## Супер скалярная архитектура

- Суперскалярными называют процессоры, способные запускать несколько команд (зачастую от четырех до шести) за один тактовый цикл.
- В процессорах этого типа, как правило, предусматривается один конвейер.



Суперскалярный процессор с пятью функциональными блоками

# Архитектуры процессоров.

## Параллельные процессоры

**SIMD и SPMD-процессор** (Single Instruction (Program)-stream Multiple Data-stream — один поток команд с несколькими потоками данных или программами) состоит из большого числа сходных процессоров, которые выполняют одну и ту же последовательность команд применительно к разным наборам данных.

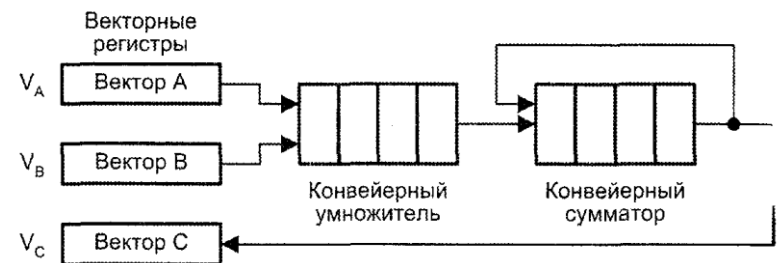
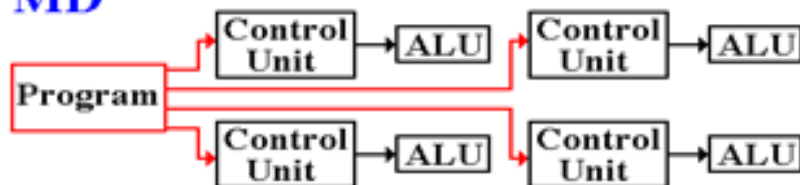
- Часто одни и те же вычисления многократно повторяются с разными наборами данных. Упорядоченность и структурированность программ, предназначенных для выполнения такого рода вычислений, очень удобны в плане ускорения вычислений за счет параллельной обработки команд (пример видеокарты).

**векторный процессор** (vector processor) также эффективен при выполнении последовательности операций над парами элементов данных. Отличие от SIMD-процессора, все операции сложения выполняются в одном блоке суммирования, который имеет конвейерную структуру.

### SIMD



### SPMD



# Архитектуры процессоров.

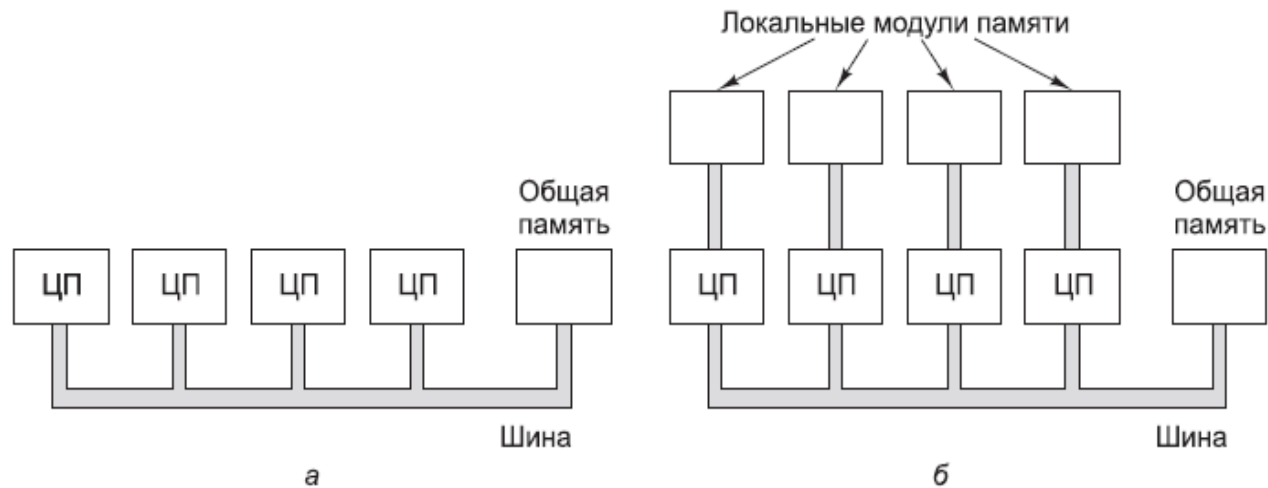
## Мультипроцессоры

Система из нескольких параллельных процессоров, имеющих общую память, называется **мультипроцессором**.

- Имеют единую память, их работа должна согласовываться программным обеспечением (сильно связанные процессоры).

Процессоры, состоящие из большого числа взаимосвязанных компьютеров, у каждого из которых имеется собственная память называются **мультикомпьютерами**.

- В них процессоры являются слабо связанными.



# Особенности архитектуры современных процессорных систем

Аппаратные средства  
телекоммуникационных систем.

Введение в архитектуру  
процессорных устройств.

# Архитектуры процессоров.

## Особенности современных архитектур процессоров

- Увеличение количества ядер микропроцессора коммутация между ядрами.
- Увеличение объема КЭШ памяти и уровней Кэш-а, а также отдельные кэши данных и инструкций.
- Конвейеризированные и суперскалярные архитектуры
- Спекулятивное выполнение команд.
  - Спекулятивное выполнение команд. Перераспределение команд в пределах одного блока (например цикл или if) и выполнение «тяжелых» команды раньше чем станет известно, понадобится ли она. Такие команды обрабатываются в период ожидания в основной ветке (например ожидания блока расчета float). Недостаток актами типа Spectra, meltdown и т.п.
- Встроенный контроллер доступа к памяти (MCU) - оптимизация работы с ОЗУ
- Система команд X86-X64 (AMD x64) – расширенная система команд с 64 битной адресацией и расширенные системы команд SSE
  - SSE для ускорения вычислений с высокой степенью упорядоченности — например, обработки мультимедийных и научных данных.
- Параллельное выполнение двух потоков инструкций ядром (hyper threading).
- Производительность зависит от тактовой частоты, IPC и энергопотребления (Instructions Per Clock)
  - IPC количество инструкций, исполняемых CPU за один так, зависит от логической структуры ядра.

# Архитектуры процессоров. Примеры.

## Микроархитектура одноядерного процессора

Процедура выборки. Команды программы хранятся в кэш памяти команд и извлекаются оттуда предпроцессором.  
декодирование команд. Разбиение команд на примитивные микрокоманды, которые воспринимаются функциональными устройствами процессора.

Микрокоманда получает из кэша данных свои операнды и готова к исполнению.

Декодированные микрокоманды образуют в предпроцессоре очередь к исполнительным блокам.  
Исполнительные блоки в виде конвейеров

При неупорядоченном исполнении команды подаются на исполнительные блоки не в порядке следования в программе, а по мере готовности их операндов.

Команды поступают в исполнительные блоки и выполняются. В силу различной скорости выполнения операций в конвейерах происходит переупорядочение команд и выдачи их результатов.

Постпроцессор следит за готовностью результатов на выходе исполнительных блоков и осуществляет возврат к естественной последовательности команд.

Результат данной команды считается готовым, если завершились все предыдущие команды и их результаты признаны готовыми.

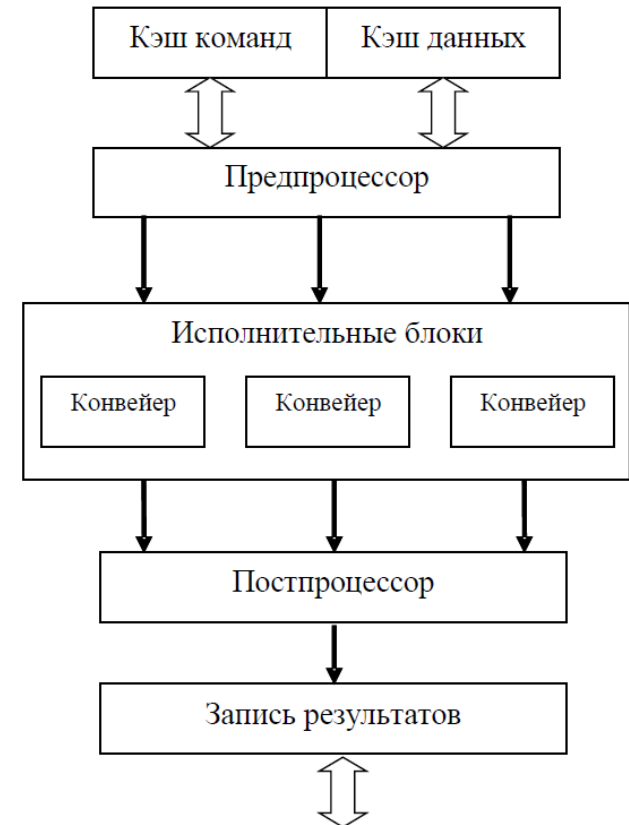
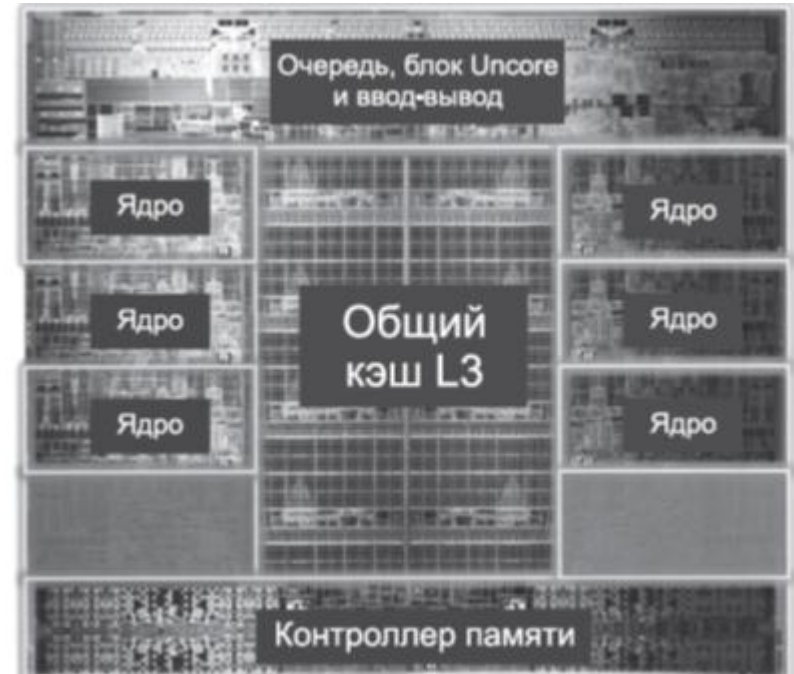


Схема классического процессора

# Архитектуры процессоров. Примеры.

## Архитектура современных процессоров Intel

- Каждое ядро имеет собственные кэши 1 и 2 уровня, но также имеется общий кэш 3 уровня (L3), используемый всеми
- *субъядро (uncore)* - компоненты, отвечающие за средства коммуникации :
  - контроллер памяти (memory controller),
  - интерконнект QuickPath (QuickPath links, QPI у INTEL, HyperTransport у AMD), последовательная кэш-шина типа точка-точка для соединения процессоров и для передачи данных между процессором и системной платой.
  - управления энергопитанием (powermanagement),
  - встроенный графический контроллер.



Микросхема Intel Core i7-3960X. Подложка имеет размеры 21×21 мм и содержит 2,27 миллиарда транзисторов

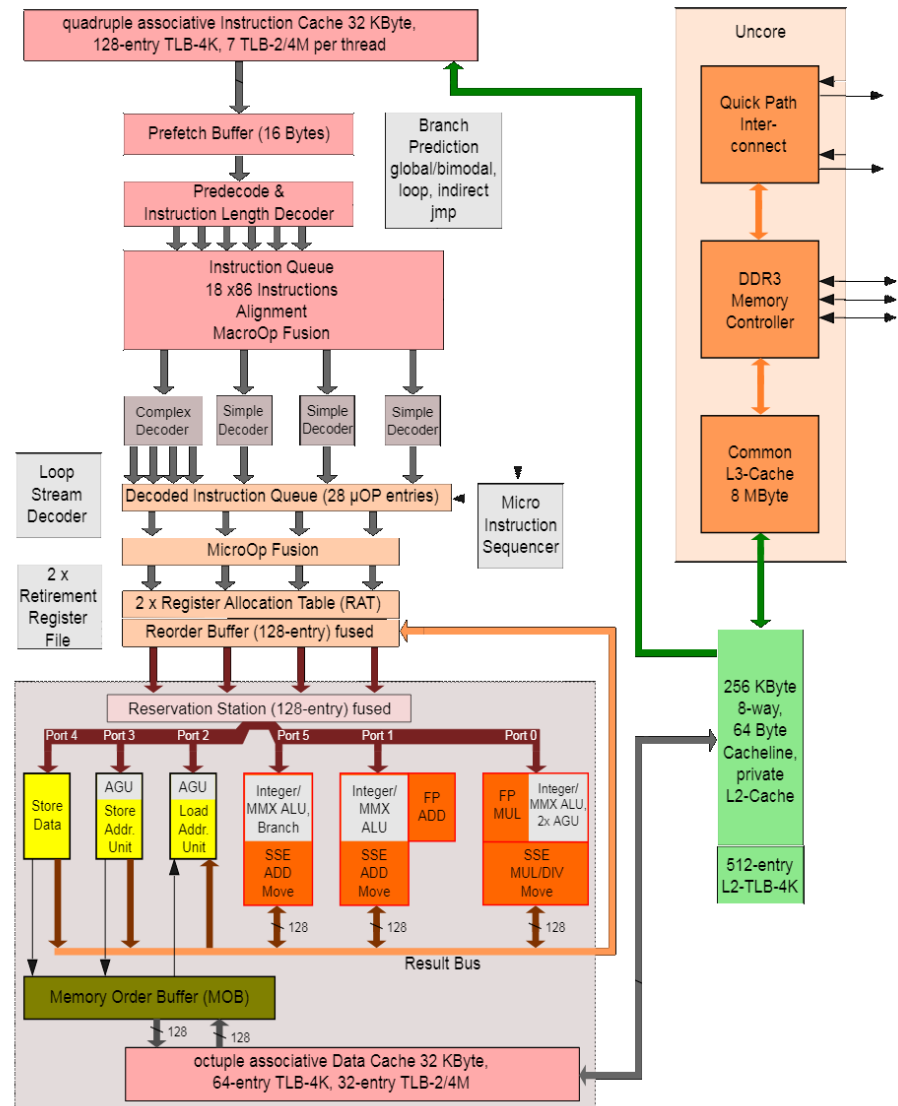
- Некоторые процессоры содержат блок - *Системный агент (System agent)* - содержит многоканальный контроллер памяти, «мосты» PCI-Express, DMI, дисплейные интерфейсы, блок аппаратного декодирования видео.



# Архитектуры процессоров. Примеры.

## Архитектура современных процессоров Intel

- Каждое ядро имеет собственные кэши 1 и 2 уровня, но также имеется общий кэш 3 уровня (L3), используемый всеми

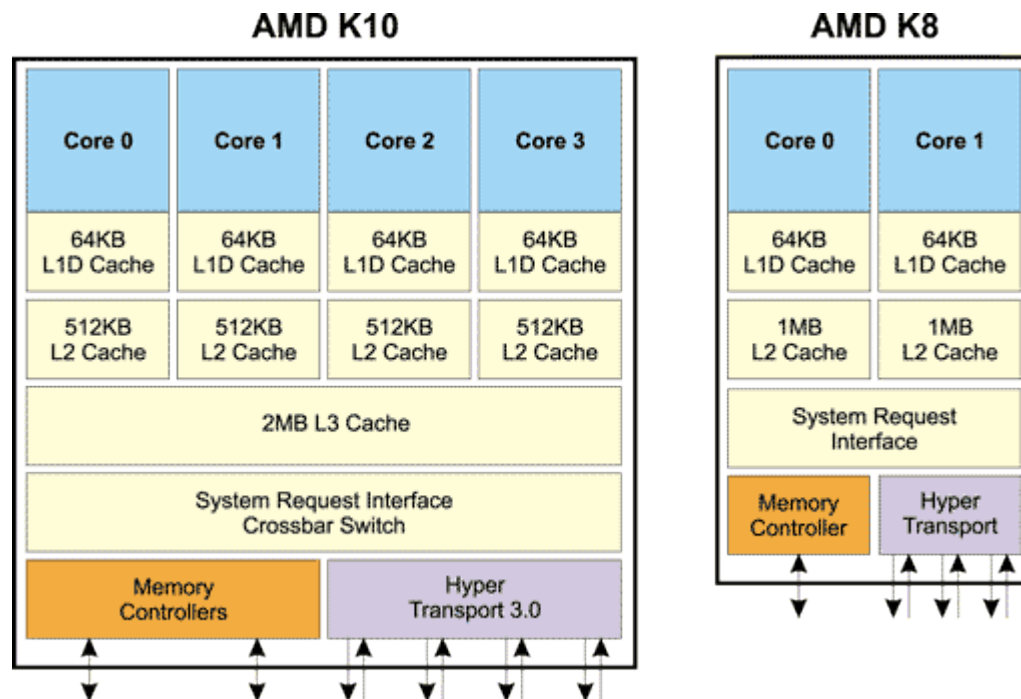


Архитектура ядра Intel Core i7

# Архитектуры процессоров. Примеры.

## Архитектура современных процессоров AMD

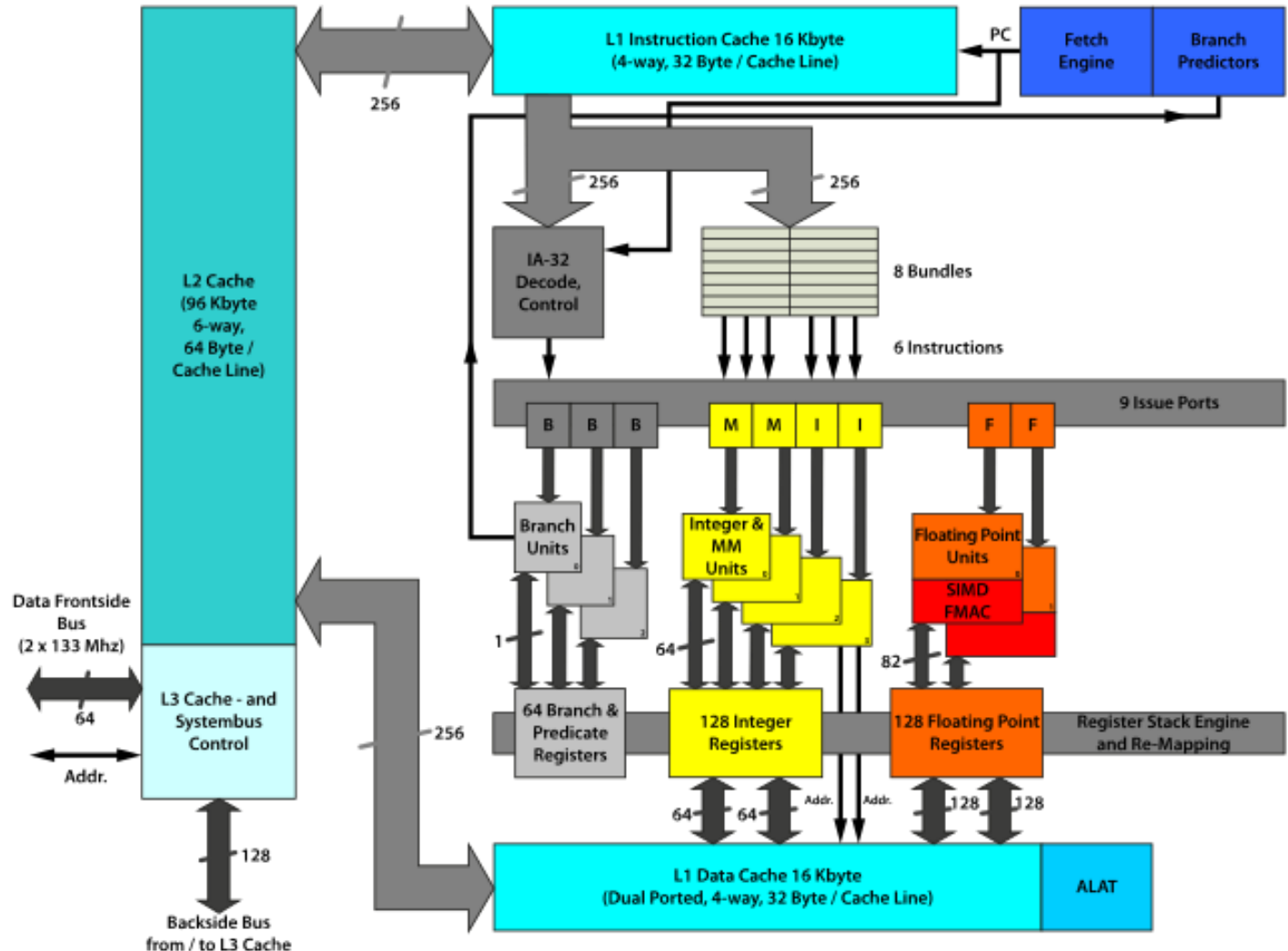
- технология HyperTransport для создания многопроцессорных системы, и объединять на одном кристалле несколько ядер (AMD Opteron).
- интегрированный контроллер памяти, не требуются промежуточные элементы (чипсет, который вносит задержку как минимум в такт).
- механизм предсказания косвенных переходов
- Каждое ядро работает на своей частоте.



# Архитектуры процессоров. Примеры.

## Архитектура процессоров Intel IA64

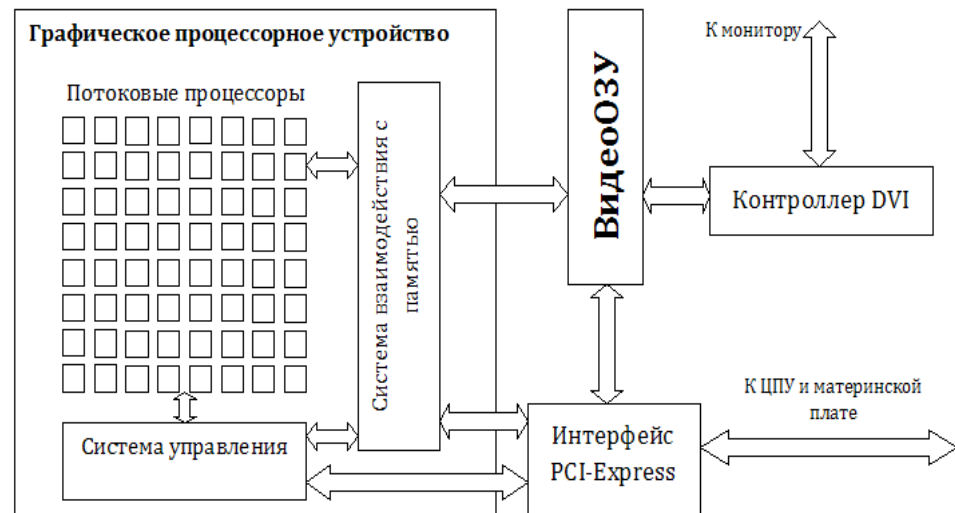
Используется  
в серверных  
процессорах  
Itanium  
(Архитектура  
EPIC)  
VILW  
архитектура  
Спекулятивны  
й КЭШ  
Исполнение  
простых  
команд по 3  
Несовместима  
с X86



# Архитектуры процессоров. Примеры.

## Архитектура современных графических процессоров

- Содержат набор одинаковых вычислительных устройств (поточковых процессоров, ПП), работающих с общей памятью ГПУ (видео ОЗУ) (SIMD архитектура).
- Все ПП синхронно исполняют один и тот же шейдер.
  - За один проход, являющийся этапом вычислений на ГПУ, шейдер выполняется для всех точек двумерного массива.
- Система команд ПП включает арифметические команды для вещественных и целочисленных вычислений и команды обращения к памяти.
- ГПУ выполняют операции асинхронно, в потоках и только с данными на регистрах. Из-за высоких задержек доступа к ОЗУ.
- За переключение потоков отвечает диспетчер потоков, который не является программируемым
- благодаря большому количеству поточковых процессоров высокая производительность.



# Архитектуры процессоров. Примеры.

## Архитектура современных графических процессоров

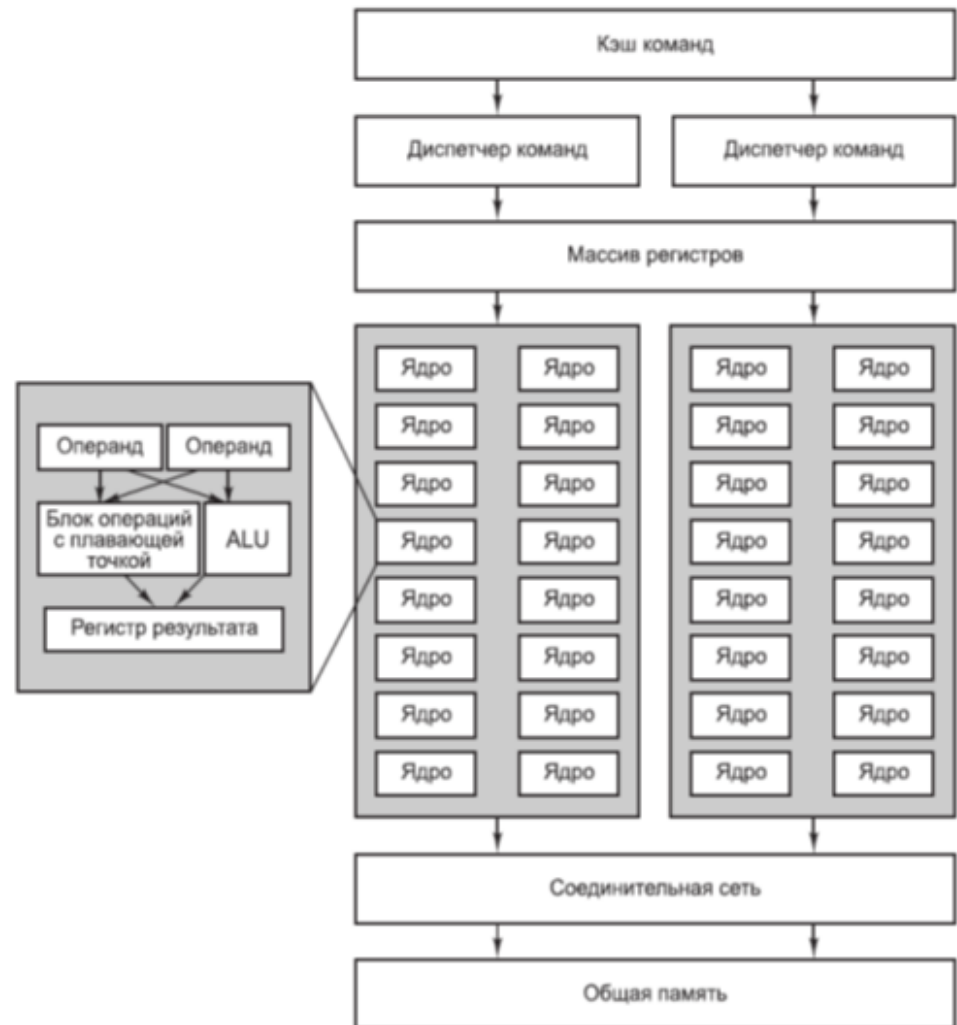
- Низкая скорость чтения из видео ОЗУ в шейдере на ГПУ (высокая латентность).

чтобы его компенсировать, требуется обрабатывать большое (10000 и более) количество элементов за 1 запуск.

Поддержка аппаратной многопоточности.

Разнородность архитектур.

Следует учесть, что оптимальные программы для NVidia и AMD будут сильно различаться.

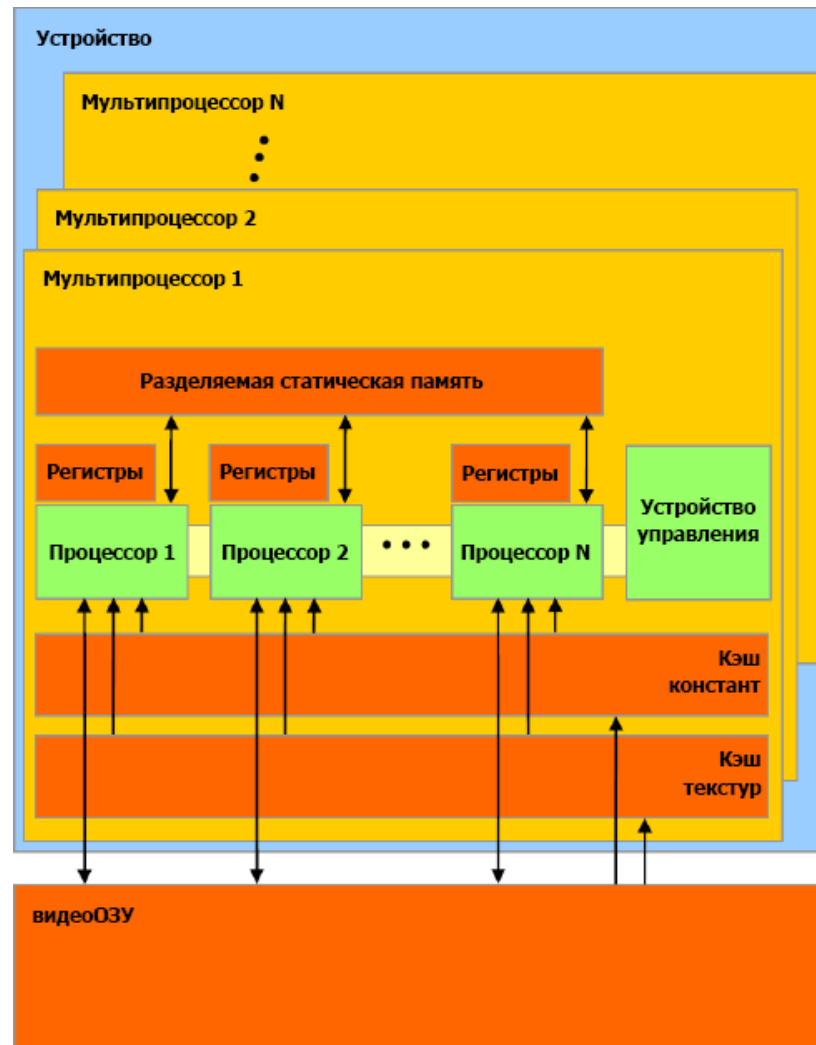


SIMD-ядро графического процессора Fermi

# Архитектуры процессоров. Примеры.

## Архитектура современных графических процессоров

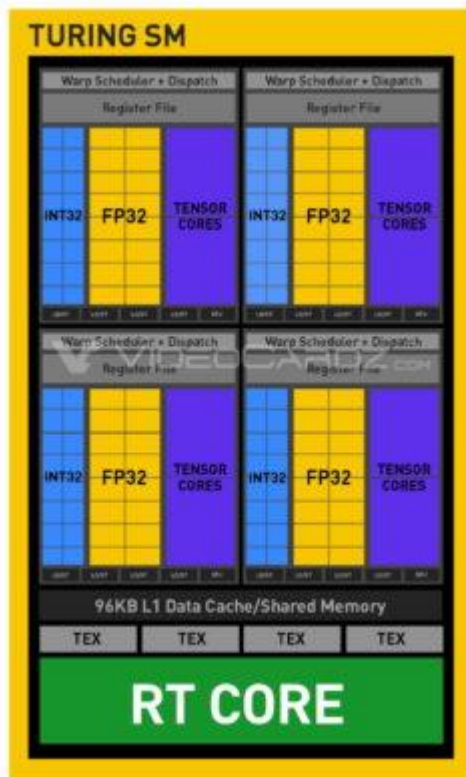
- 128 потоковых процессоров (ПП) объединены в SIMD-группы по 16 мультипроцессоров (МП),
- МП работают независимо друг от друга, хотя и исполняют один и тот же шейдер.
- Каждый ПП является суперскалярным устройством и может выполнять до 2 команд за такт.
- Каждому ПП доступна вся видео ОЗУ ему доступна вся память, как на чтение, так и на запись.
- Однако на практике, ввиду слабости средств синхронизации между различными МП, процесс обработки строится так, чтобы адреса записи не пересекались.



Архитектура GeForce 8x

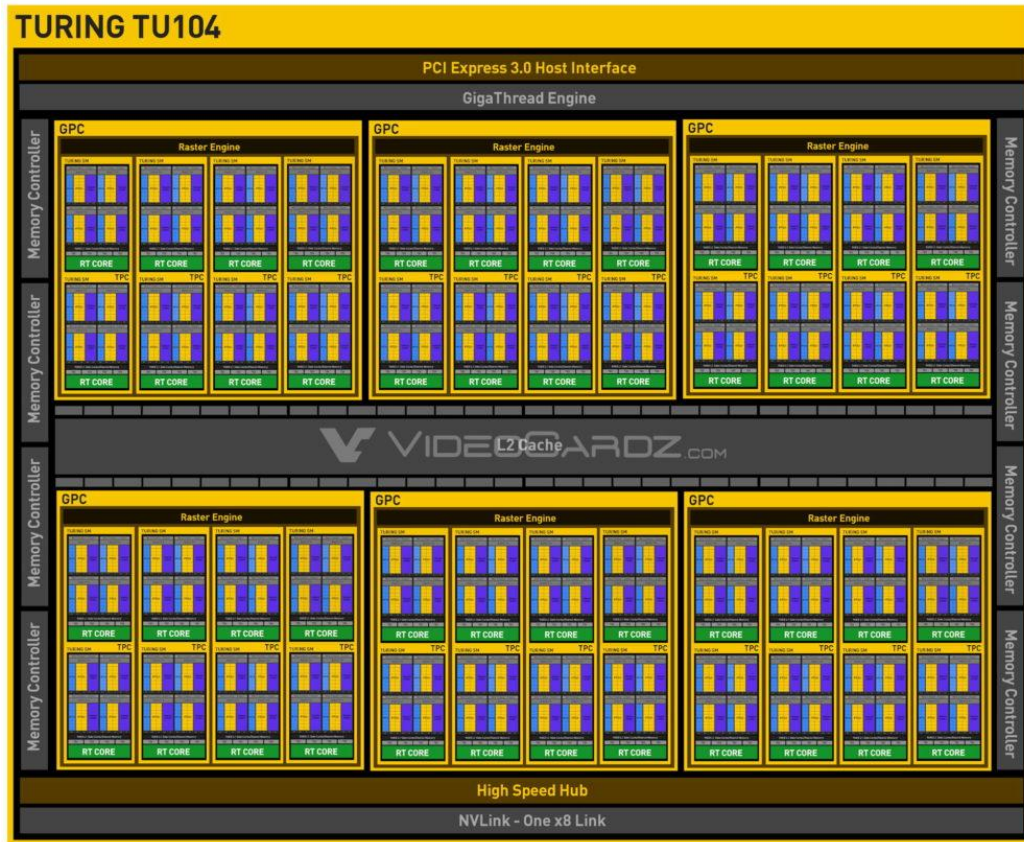
# Архитектуры процессоров. Примеры.

## Архитектура современных графических процессоров



Архитектура каждого процессора  
NVideo Turing 102, 104, 106

Ядро состоит из целочисленного блока  
Int32, блока float FP32 и блока  
матричных вычислений Tensor Cores



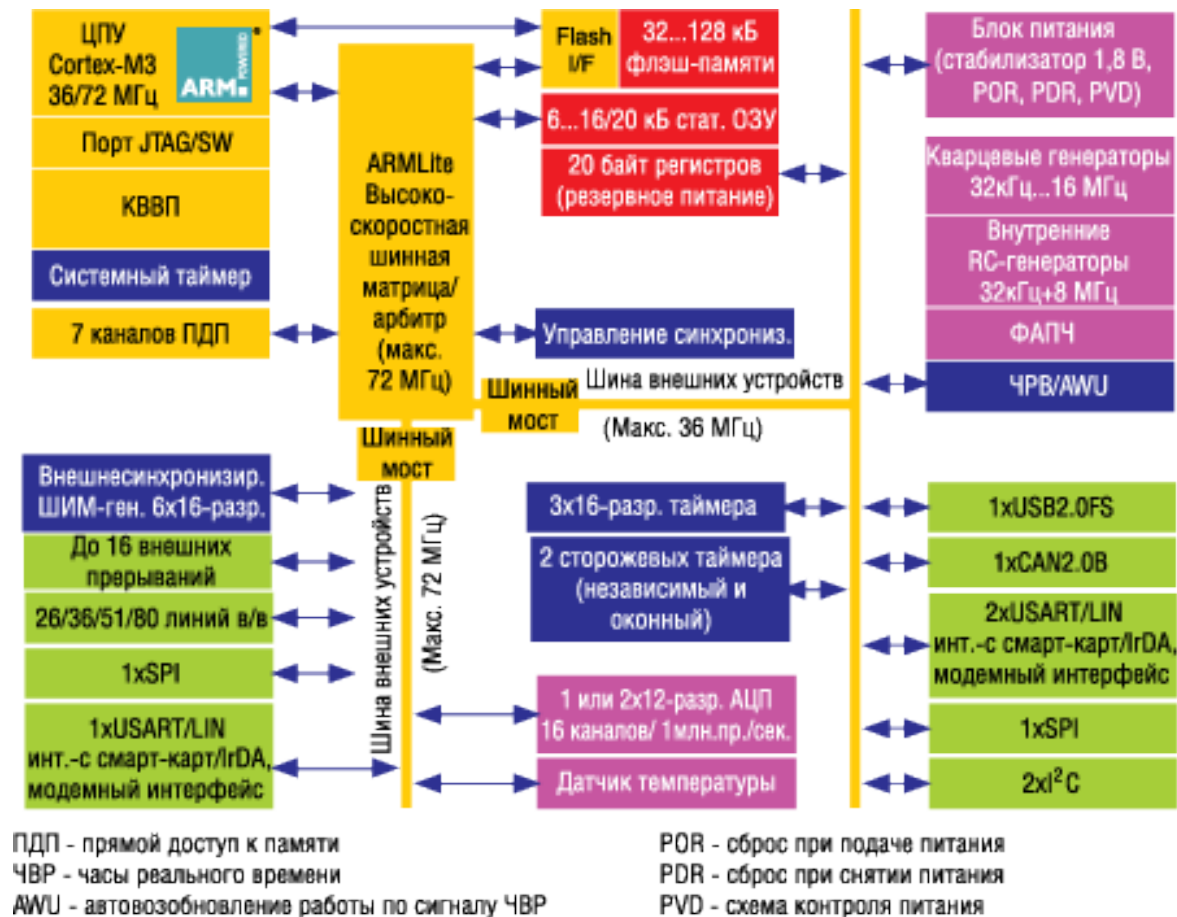
Архитектура NVideo Turing 104 (GeForce 20xx)

Используются параллельные вычисления в  
FP32 и INT32,  
Технология сжатия данных без потерь для  
увеличения пропускной способности шины



# Архитектуры процессоров. Примеры.

## Архитектура современных микроконтроллеров

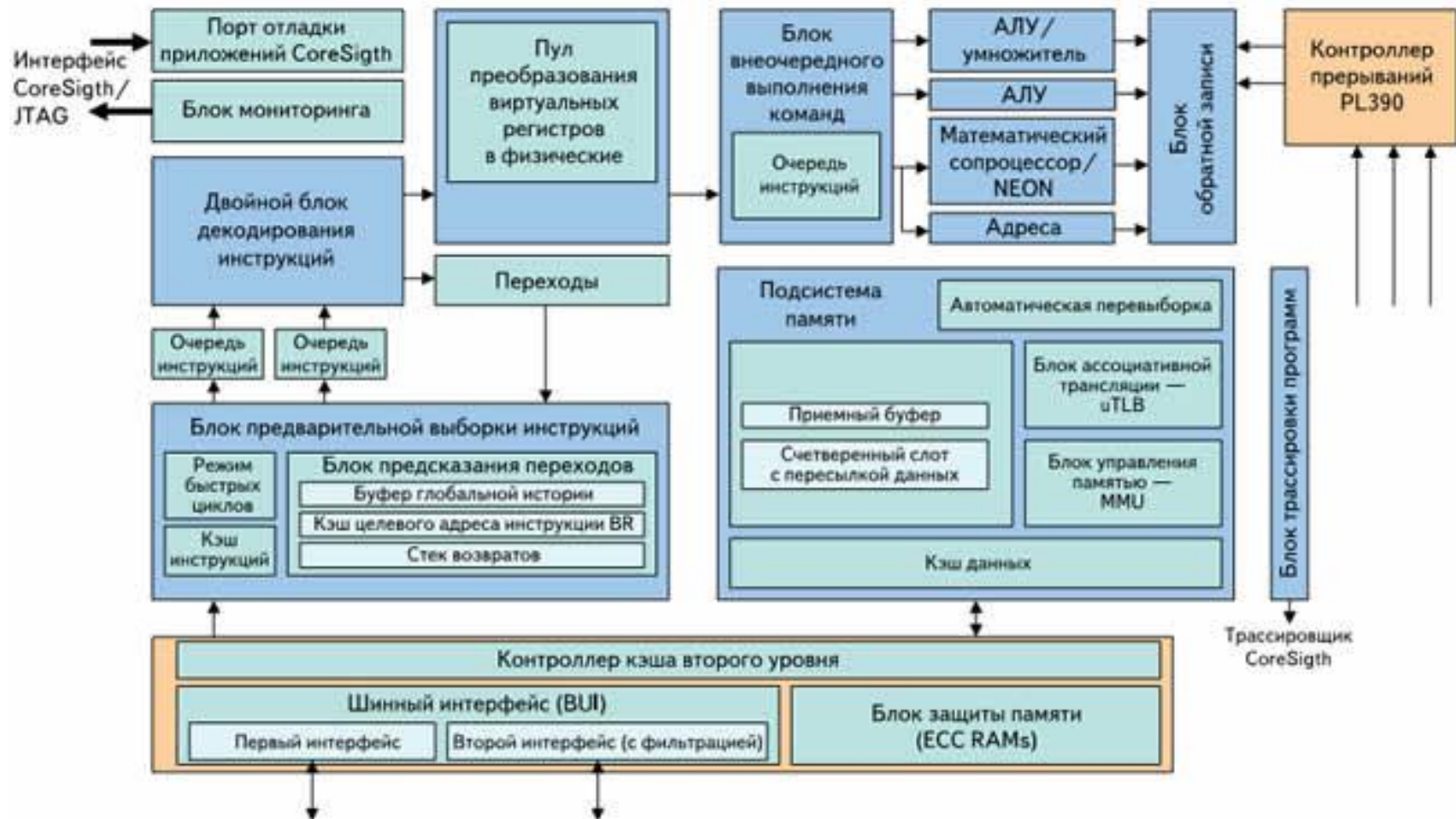


- ARM Cortex M 32битная система, сопроцессоры аппаратного деления и умножения, работа с float
- RISC система команд
- конвейерная архитектура



# Архитектуры процессоров. Примеры.

## Архитектура современных микропроцессоров



- ARM Cortex A9 (на базе Cortex M)
- Конвейерная многоканальная архитектура
- Многоуровневый спекулятивный КЭШ

# Архитектуры процессоров. Примеры.

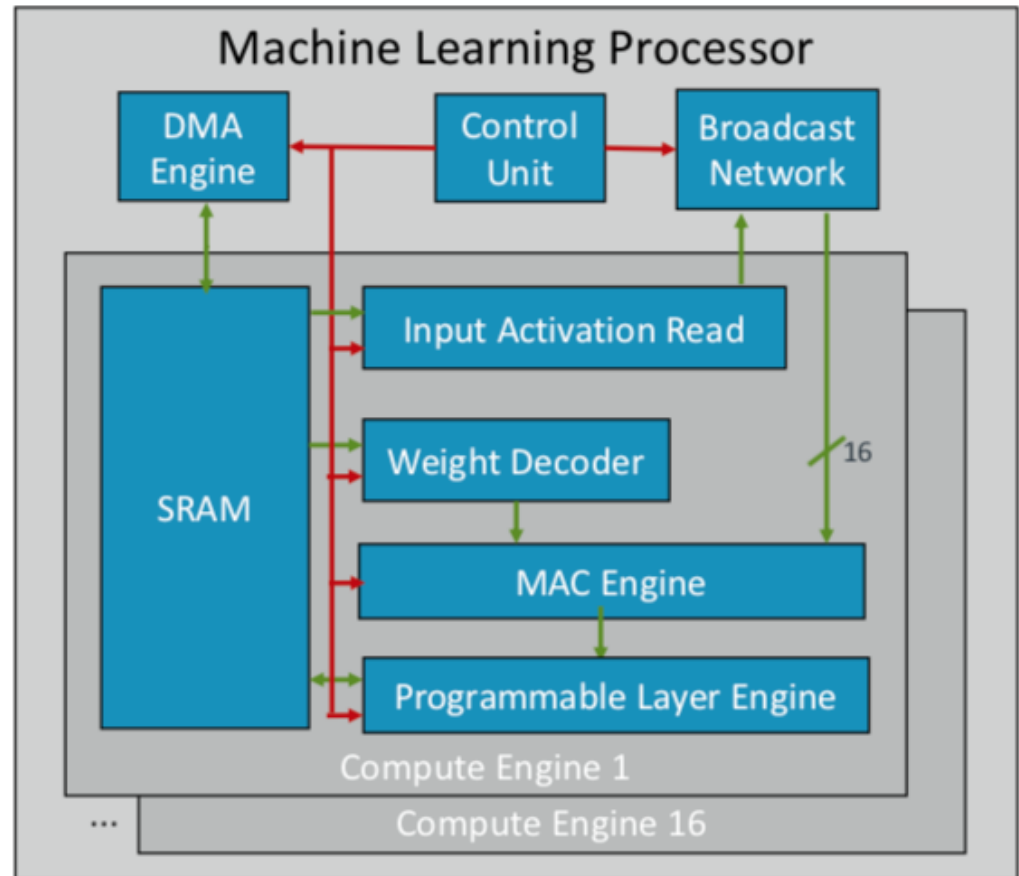
## Архитектура современных процессоров с ИИ

внедрение ИИ для определения приоритетов хранения, обработки и перемещения данных по всей системе.

Функции

прогнозирование,  
распределение ресурсов системы,  
каждый вычислительный модуль  
работает на отдельной карте  
признаков,  
электромагнитная адаптация.

Концепция Nvidia: создание  
выделенного «движка» глубокого  
обучения рядом с GPU для  
оптимизации обработки  
изображений и видео.

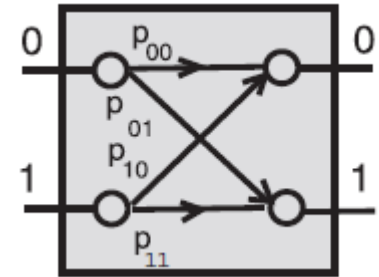
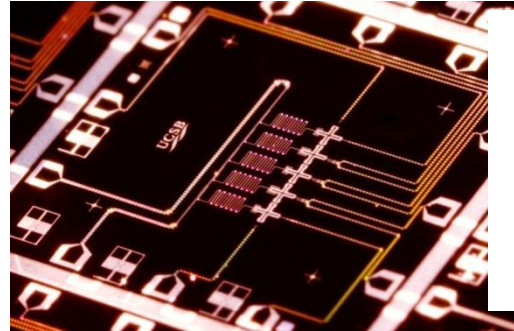


*Архитектура процессора ML от ARM*

# Архитектуры процессоров. Примеры.

## Архитектура квантовых процессоров

используются кубиты – состояния спинов элементарных частей, могут одновременно находиться в двух состояниях, 3 кубита одновременно в 8 состояниях. Высокая параллельность вычислений. Экспоненциальный рост производительности.



*вид квантового процессора Google, Квантовый элемент И-Не и схема квантового компьютера*