Аппаратные средства телекоммуникационных систем

Лекция 2. Введение в архитектуру процессорных устройств.

Последовательные системы с гибкой логикой – процессоры

Аппаратные средства телекоммуникационных систем.

Основные понятия и определения архитектуры вычислительной техники.

Последовательные устройства гибкой логики. Понятие процессор

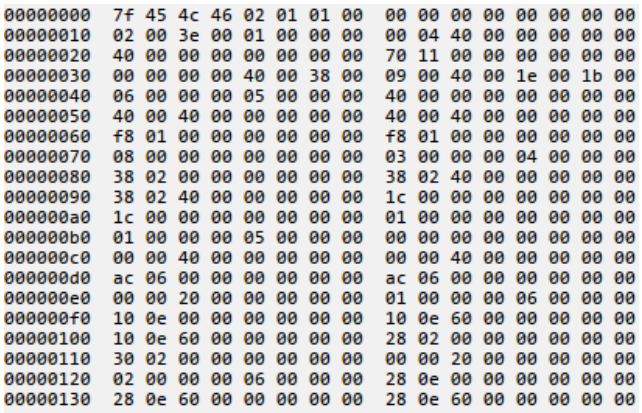
- ***Процессор*** – электронный блок либо интегральная схема (микропроцессор), исполняющая машинные инструкции (код программы), главная часть аппаратного обеспечения компьютера или программируемого логического контроллера.
  - *Устройства типа процессор подчинены т.н.* ***«принципу программного управления»****.*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

Принцип программного управления.

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

О машинном коде и языках программирования

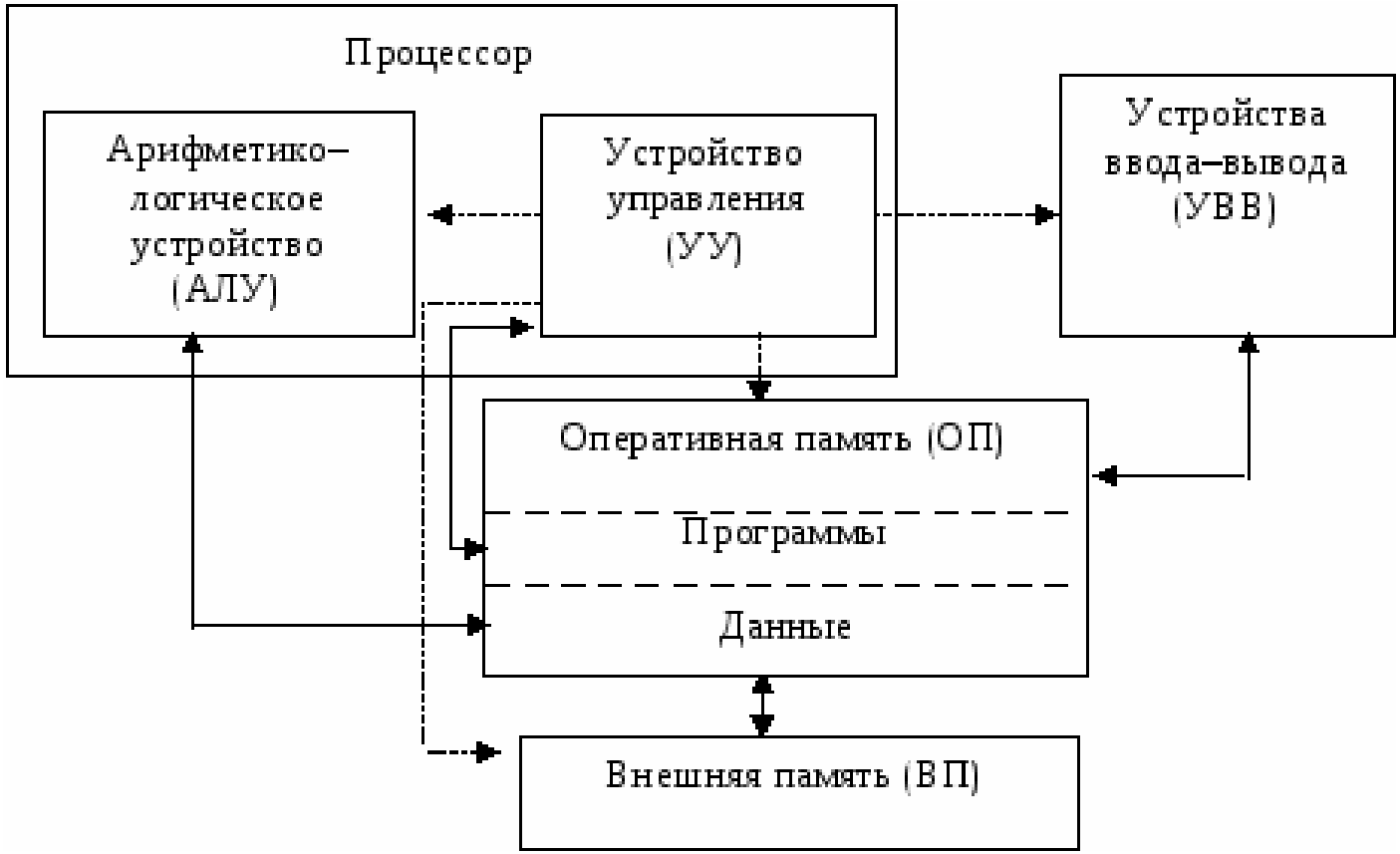
- На машинном уровне программа представляет собой набор аппаратно-выполняемых команд – **машинный код**.
  - Примеры таких команд – сложение, умножение, логическое или, перенос значения из одной ячейки памяти в другую.
- *Каждая команда имеет свой кодовый номер и адреса двух ячеек – данных, для выполнения 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
```

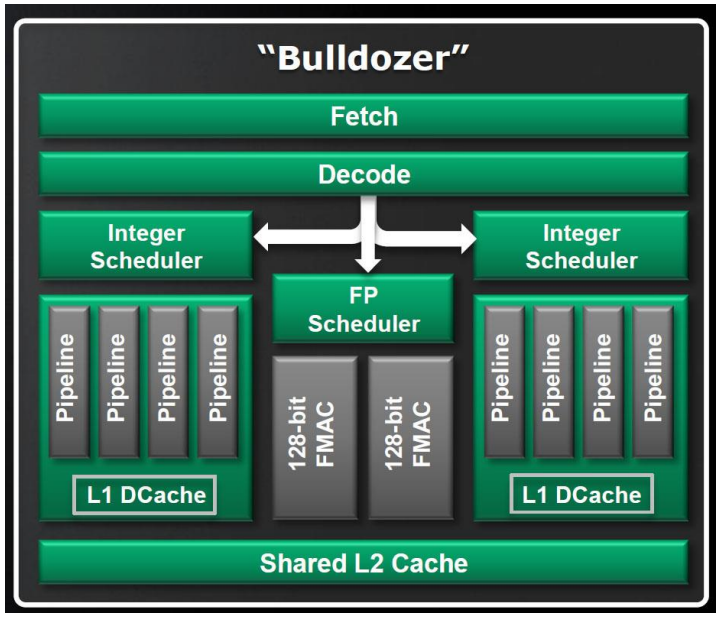
Принцип построения ЭВМ

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



Блок-схема современного процессора

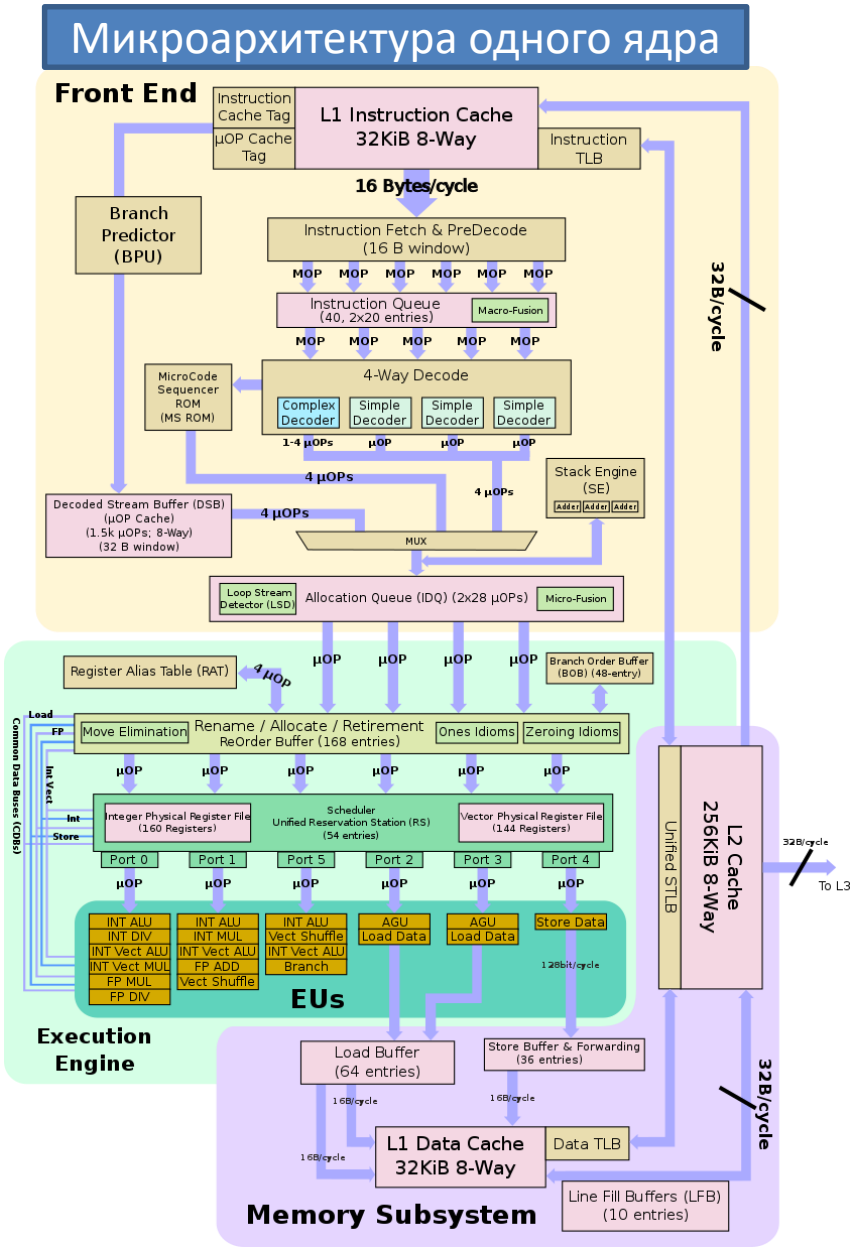
AMD Bulldozer (2003)



Блок-схема современного процессора

Ivy Bridge 2011

Кристалл процессора



Принцип построения ЭВМ

- ***Центральный процессор*** **(ЦП) включает:**
  - **АЛУ – Арифметико-Логическое Устройство**
  - **блок устройства управления (УУ), и регистры**
  - **Блок работы с памятью**
- ***ЦП*** *обладает принцип последовательной передачи управления.*
- Процессор имеет **набор регистров в устройстве управления (УУ)**
  - **Часть регистров доступна для работа с АЛУ:**
    - хранения операндов;
    - выполнения действий над операндами;
    - и формирования адреса инструкций и операндов в памяти.
  - ***Другая часть регистров используется процессором для служебных (системных) целей,***
    - *доступ к ним может быть ограничен (в т.ч. программно-невидимые регистры).*

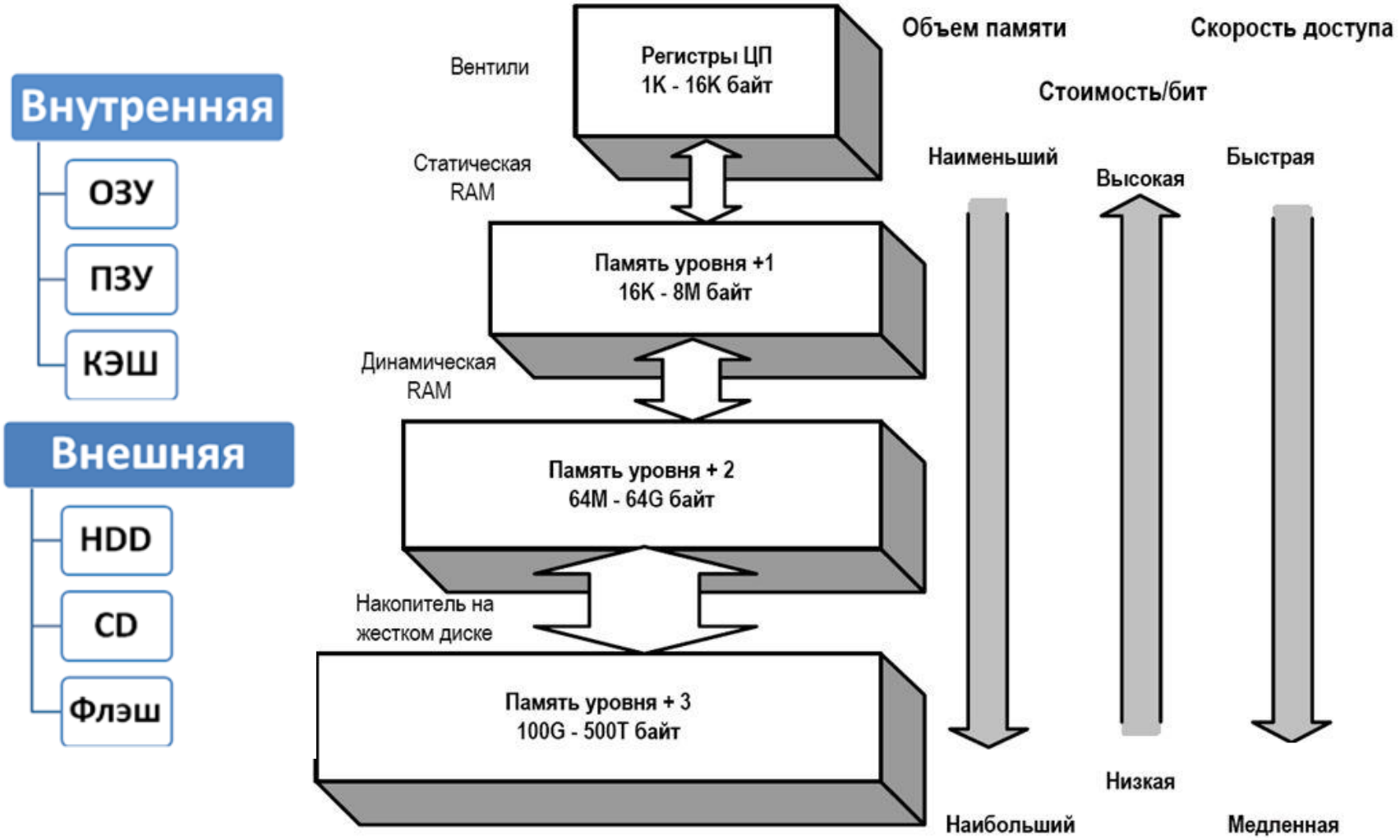
Принцип построения ЭВМ

- **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

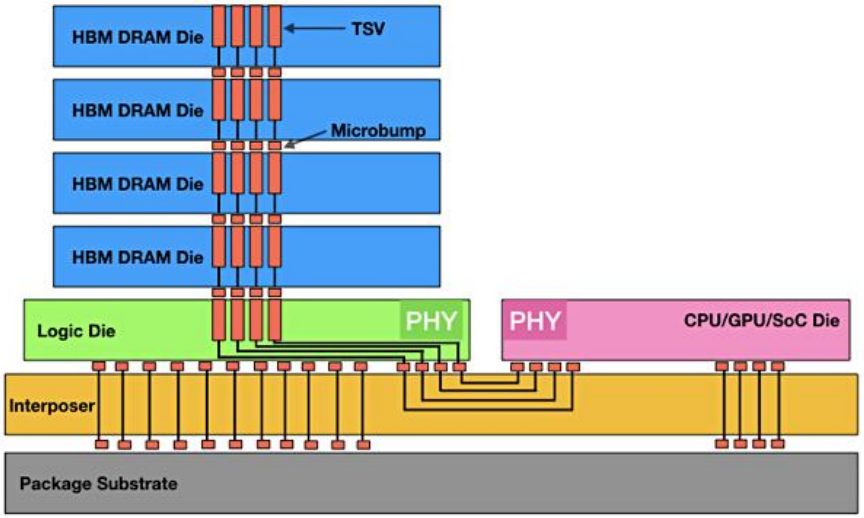


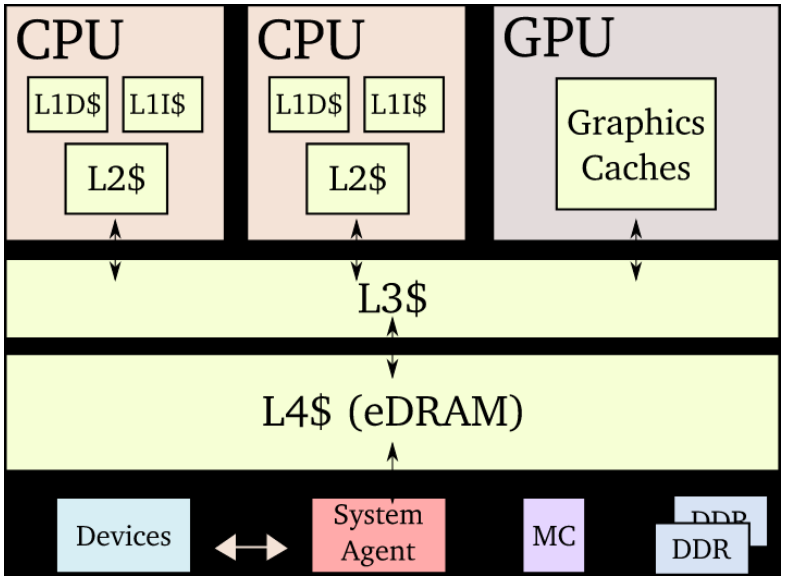
Оперативная память

- ***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**.
  - **Часть ОЗУ** может быть прикреплена к процессору и называться или **L4 или eDRAM, HBM** (зависит от технологии изготовления).
  - Иногда предлагается также **L5 уровень** – использование **SSD в качестве ОЗУ (**NVRAM, NVDIM**).**
  - *Также ЭВМ имеет некоторые виды специальной памяти (например, видеопамять графического адаптера).*
  - **Латентность** (время) доступа к ОЗУ наиболее узкое место работы процессора. Большинство решений реализованных в процессорах направленны на компенсацию латентности доступа к ОЗУ.
  - При решении проблем латентности всегда должен быть решен баланс между ценой/объемом/временем доступа.

Особенности современного КЭШ

- Увеличение объема КЭШ памяти и уровней Кэш-а, а также отдельные кэши данных и инструкций.
  - А также добавление eDRAM модулей, а также HBM модулей быстрой памяти и модулей энергонезависимой флэш памяти (NVRAM, NVDIM).
  - Данный тренд направлен на решение проблем основного узкого места ЦПУ латентности доступа к основной памяти. За счет таких многоуровневых асинхронных решений повышается вероятность т.н. попадания в Кэш – то есть вероятность того, что нужные данные в нужный момент будут в кэш памяти.





Иерархия памяти в компьютере

- ***Постоянная память*** **(ПЗУ)**, из нее можно только считывать команды и данные
- В любом компьютере есть ***энергонезависимая память***, в которой хранится программа начального запуска компьютера и минимально необходимый набор сервисов (FLASH, EEPROM, **ROM BIOS**).
  - Доступ к внутренней памяти осуществляется по одномерному (линейному) адресу, который представляет собой двоичное число. Доступна для процессора.
- ***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 ок*ружающим,* 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 PC 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*** гибрид 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

•Математические сопроцессоры (FPU) -операции с плавающей запятой (имеют 2 ЛУ для мантиссы и экспоненты);

• SIMD сопроцессоры – операции над линейными (иногда и двух-мерными) массивами данных.

•Графические (многоядерные, много АЛУ, мало команд другого профиля) ориентация на рендеринг – расчет текстуры по модели;

• Специализированные, например навигационные (с GPS); кодирование данных, медисопроцессоры, аудиосопроцессоры и др.

• Цифровые сигнальные процессоры –аппаратное решение задач умножения с накоплением в цикле.

•Коммуникационные (поддержка сетевых интерфейсов и протоколов). Например (Ethernet, или беспроводных, например WiFi и GPRS)

• ***Часть сопроцессоров может быть расположено внутри процессора или его ядер, другие могут быть вынесены наружу.***

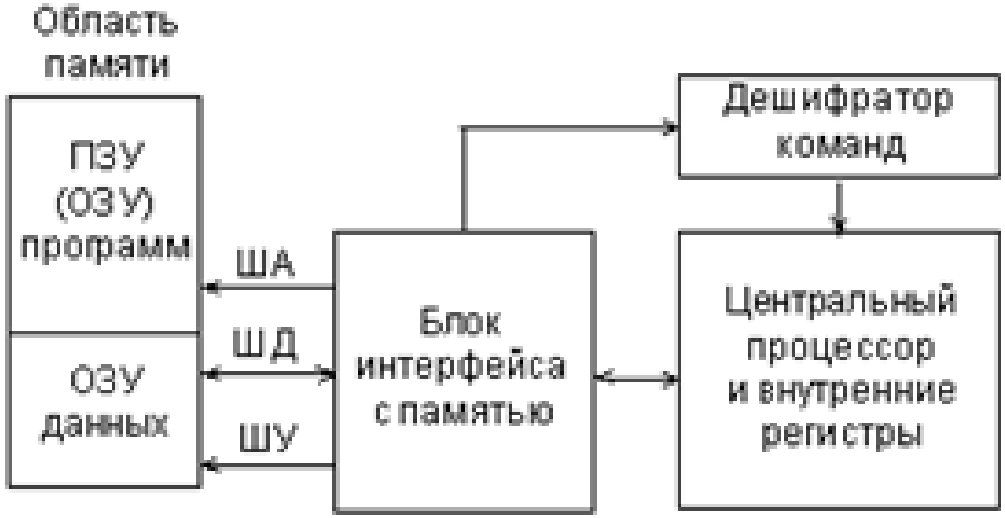
Типы архитектуры процессорных устройств по принципу разделения памяти

Аппаратные средства телекоммуникационных систем.

Введение в архитектуру процессорных устройств.

Виды архитектура процессоров

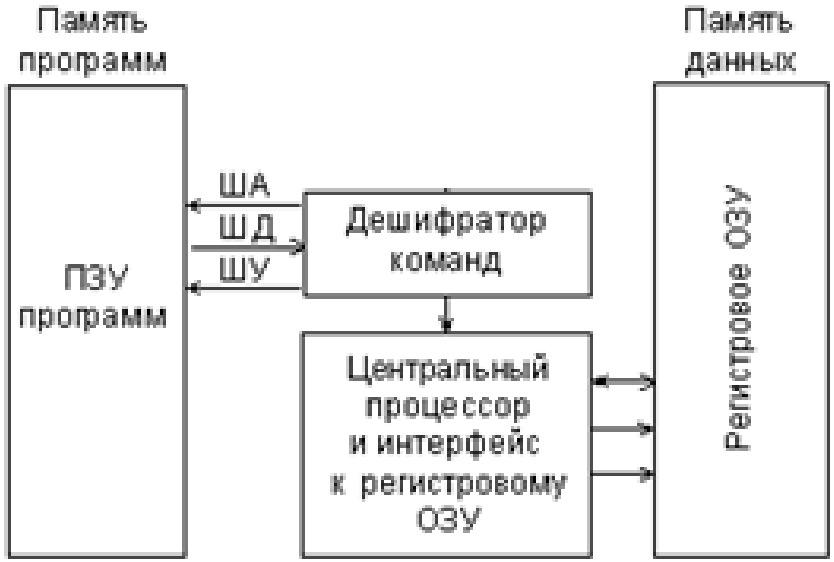
- ***Архитектура фон Неймана*** – предполагает хранение программ и данных в общей памяти.
- Наиболее характерна для процессоров, ориентированных на использование в компьютерах.
- Примером могут служить микропроцессоры семейства x86.



Современный вариант – **суперскалярная архитектура – несколько АЛУ и сопроцессоров параллельно**

Виды архитектура процессоров

- ***Гарвардская архитектура*** предполагает раздельное использование памяти программ и данных.
- **используют для повышения быстродействия** системы за счёт разделения путей доступа к памяти программ и данных.
  - Позволяет организовать конвейер данных.
- **Большинство специализированных микропроцессоров (особенно микроконтроллеры) имеют Гарвардскую архитектуру.**



Особенности гарвардской архитектуры - отдельной памяти данных

- **Сокращает длину *команд***
  - **Ускоряет поиска информации в *памяти данных*.**
    - Использование единого адресного пространства приводит к увеличению формата *команд* за счет увеличения числа разрядов для адресации операндов.
- **Возможность параллельного выполнения операций.**
  - **Выборка следующей команды может происходить одновременно с выполнением предыдущей**, и нет необходимости останавливать процессор на время выборки команды.
- 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.

Особенности архитектуры Фон-Неймана

- **Упрощение устройства процессора**,-обращение к одной общей памяти.
- **оперативное перераспределение ресурсов между областями программ и данных**, что повышает гибкость системы.
- **Архитектура 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

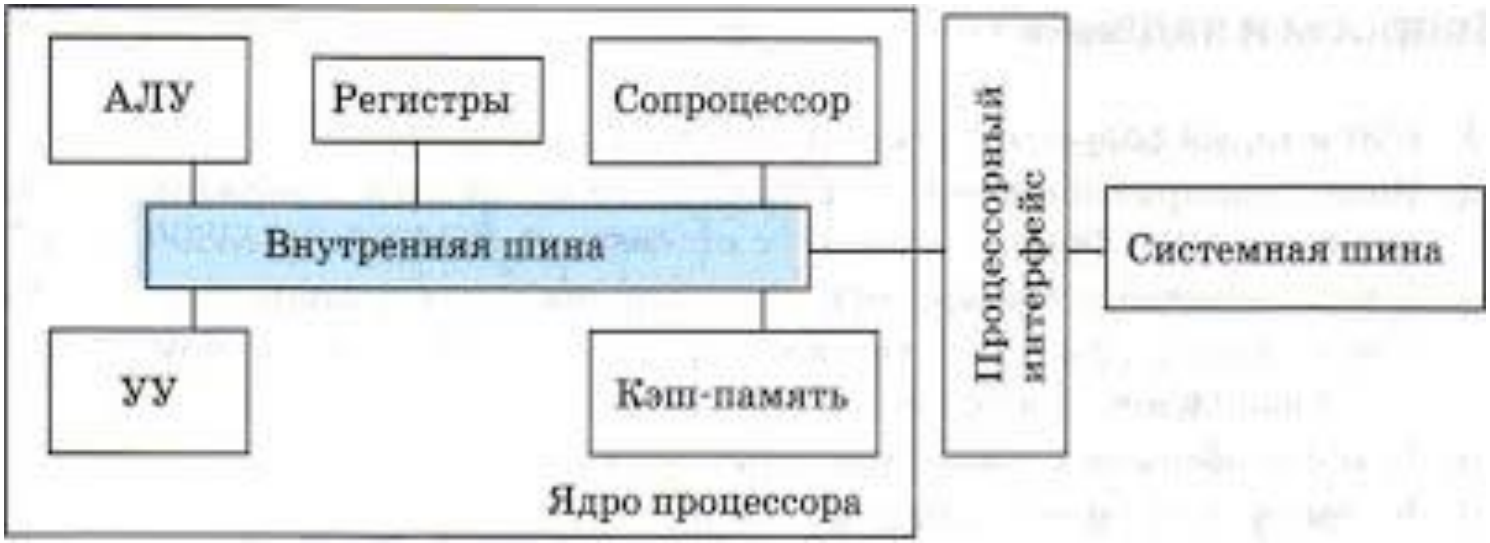
Особенности архитектуры Процессорных устройств

Аппаратные средства телекоммуникационных систем.

Основные понятия и определения архитектуры вычислительной техники.

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*.

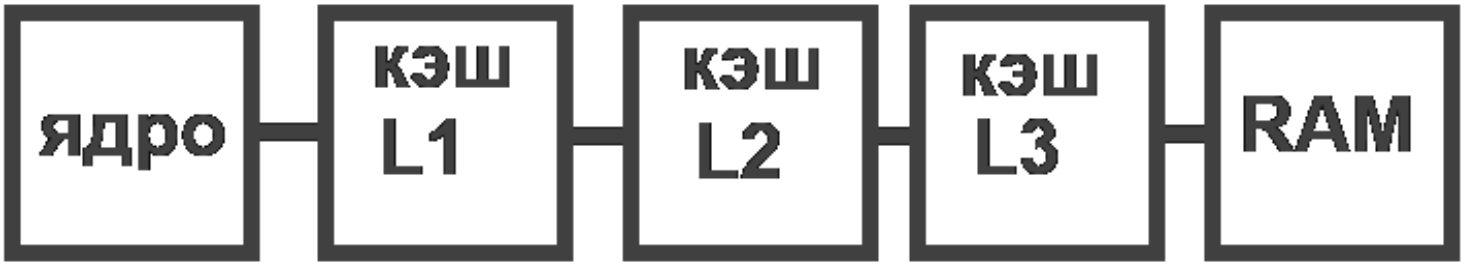


Магистральная организация процессов.

- ***Шина адреса/данных – передача адреса (например 24 линии) затем данных (например 32 линии).***
- ***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.

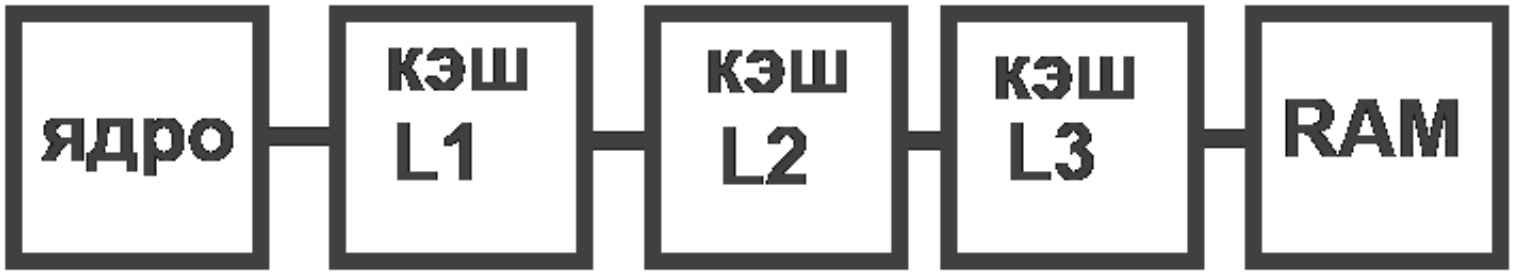
Кэш память.

- *Внутри центрального процессора быстрая память небольшого объема для хранения промежуточных результатов и команд управления –* ***КЭШ память****.*
- Операции чтения и записи с регистрами выполняются очень быстро, поскольку они находятся внутри центрального процессора.
  - ***Таким образом КЭШ решает проблему латентности данных.***
- Кэш-память позволяет держать наиболее часто используемые и последние слова внутри центрального процессора и избегать (медленных) обращений к основной памяти.
  - Кэш-память работает как асинхронный буфер по отношению к ОЗУ
  - Скорость высокая за счет **ассоциативного доступа** к памяти и ее небольшого объема, а также технологии SRAM.



Кэш память.

- **Кэш может быть многоуровневый.**
  - Часто кэш двухуровневый (L1, L2), причем L1 может быть разделен - для хранения команд L1C и отдельно для хранения данных L1D.
  - В ряде случаев вводят третий уровень (L3) для передачи данных между ядрами.
  - Бывают и более высокие уровни, последний часто называют Last level cache (LLC).
  - Часто используется victum стратегия, когда данные попадают в промежуточный кэш, а затем вытисняются в старший кэш.
  - Каждый уровень кэш-памяти дублирует предыдущий и дополняет его.
  - Также может быть L4 уровень eRAM – небольшой объем RAM прикреплённый к процессору.



Регистровая память.

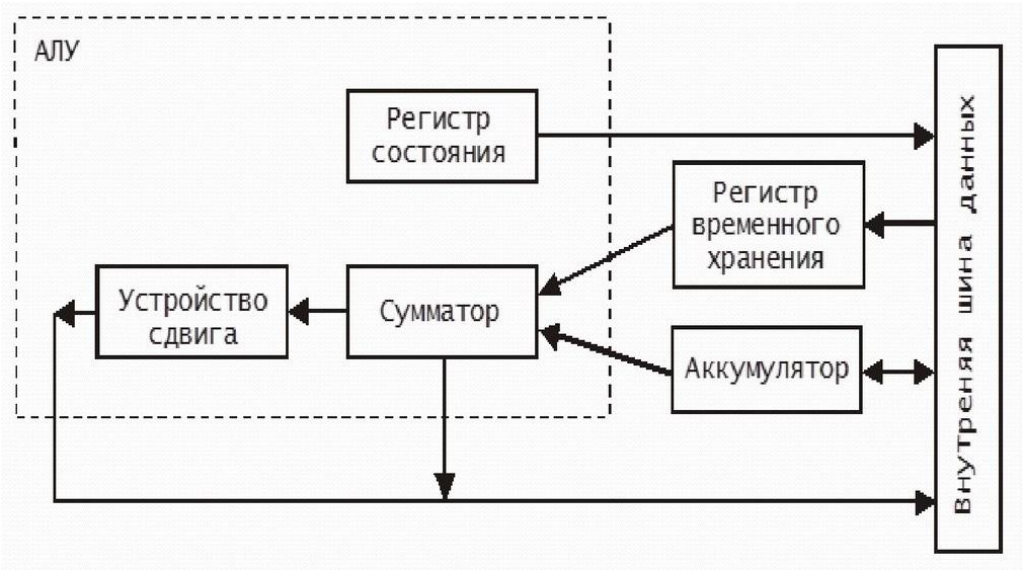
- Набор регистров - отвечают за настройки процессора, контроль его текущего состояния и хранение операндов для работы с ними. Например:
  - Регистр со значением конца диапазона памяти для команд (сегмент команд),
  - **Регистр нахождения процессора в прерывании (регистр флагов)**
  - **счетчик команд**, - какая по счету последовательная команда должна быть выполнена в настоящее время.
  - Регистр со значениями данных для текущей операции (регистры данных) и результата работы АЛУ (аккумулятор)
  - Специальные регистры указатели для работы со строками
  - Системные регистры GDTR, LDTR и IDTR для хранения базовых адресов таблиц дескрипторов - блоков памяти и прав доступа к ним ОС и приложений

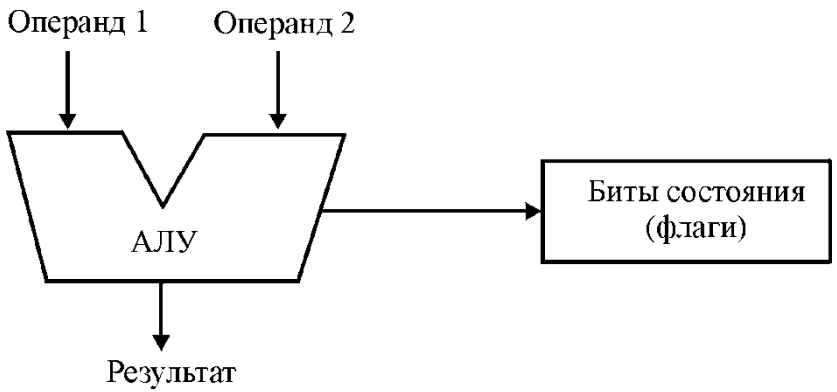
Функции устройство управления (УУ)

- Формирование адрес команды, которая должна быть выполнена
- Подача сигнала на чтение/запись содержимого ячейки запоминающего устройства (ЗУ).
- Формирование адресов операндов и управляющие сигналы для их чтения из ЗУ и передачи в арифметико-логическое устройство (АЛУ).
- Формирование признаков результата (знак, наличие переполнения, признак нуля и так далее) записываются во флаги.
  - Эта информация может использоваться при выполнении следующих команд программы, например команд условного перехода.

Арифметико-логическое устройство (ALU)

- АЛУ объединяет арифметические и логические операции.
  - Например, сложение, вычитание, сравнение величин, операции «И» и «ИЛИ».
- 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





Типы наборов команд процессоров

Аппаратные средства телекоммуникационных систем.

Основные понятия и определения архитектуры вычислительной техники.

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.

Организация наборов команд процессоров.

- ***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 на-бором 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).
- **Процессоры с разными типами команд несовместимы**

CISC система команд

CISC (англ. Complex Instruction Set Computer — «компьютер с полным набором 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 предназначены строго определенную функцию.
- одна команда может быть заменена ей аналогичной, либо группой команд, выполняющих ту же функцию.
- Характерный формат команды:

  «служебная инф.-Операция-операнд в памяти-операнд в регистре».

CISC система команд

- CISC система команд исторически появилась первой, по этому большинство процессоров CISC.
  - *Процессоры Intel, начиная с процессора 486, в RISC-ядро, которое выполняет простые (и обычно самые распространенные) команды за один машинный цикл,. 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.*
  - *Проблема CISC – часть CISC команд не используются компилятором, по этому в RISC от таких команд отказались.*

RISC система команд

RISC (англ. Reduced Instruction Set Computer — «компьютер с сокращённым набором команд») — архитектура процессора, в котором быстродействие увеличивается за счёт упрощения инструкций и их декодирования (упрощает УУ).

Примеры RISC-архитектур: PowerPC, серия архитектур ARM (ARM A7x, ARM5x, ARM, Cortex M); MISC; RISC-V (открытый набор команд) и многие другие –наиболее современный набор команд.

- Архитектура имеет постоянную длину команды и время выполнения один машинный цикл, работают только с регистрами.
  - Позволяет работать конвейером (то есть выполнять больше одной команды за один такт).

RISC система команд

RISC

RISC имеет большее количество регистров, которые могут не иметь назначения и быть переназначены в ходе работы – они представляют собой регистровый файл.

- Содержат набор простых, чаше встречающихся команд (по правилу 20-80) – то есть 20% от CISC команд.
- Основной недостаток RISC архитектуры — необходимость моделирования сложных команд.
  - Сборка сложных команд производится автоматическая из простых.
  - Формат инструкции
    - Служ. Инф. – операция – регистр операнд – регистр операнд – регистр результат

MISC система команд

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

- более простая архитектура чем RISC, используемая в первую очередь для уменьшения итоговой цены и энергопотребления и габоритов 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.

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.
  - Каждая команда состоит из RISC подобных суб-команд.
  - 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 и работает как часть Устройства управления.
  - Например, в следующем поколении максимальная длина «очень длинной команды» может из условных 256 бит стать 512 бит, и исчезнет совместимость.
  - Ключевым отличием от суперскалярных 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).

Параллелизм процессорных архитектур на уровне команд - Конвейерная архитектура

Аппаратные средства телекоммуникационных систем.

Основные понятия и определения архитектуры вычислительной техники.

Микроархитектура одноядерного процессора

- Выборка команд из кэш памяти команд.
- Декодирование команд.
  - Разбиение команд на примитивные микрокоманды,
    - 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
  - Команды подаются на исполнительные блоки а по мере готовности их операндов.
    - Неупорядоченное исполнение
    - При этом ведется аппаратный контроль зависимостей команд.

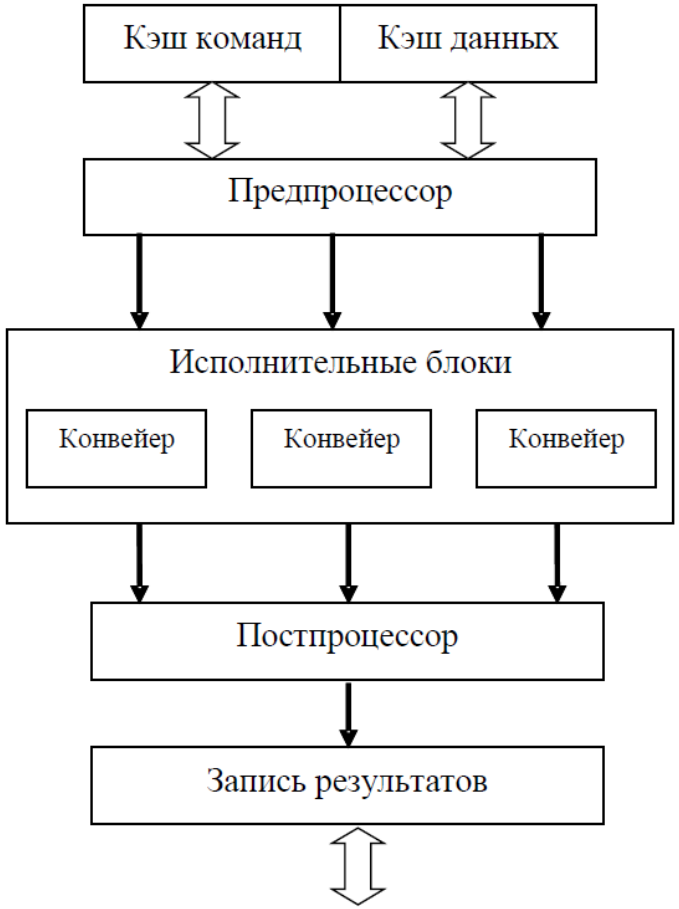


Схема классического процессора

Микроархитектура одноядерного процессора

- 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.
  - Если в ходе исполнения возникает ошибка, то процессор переигрывает команду заново

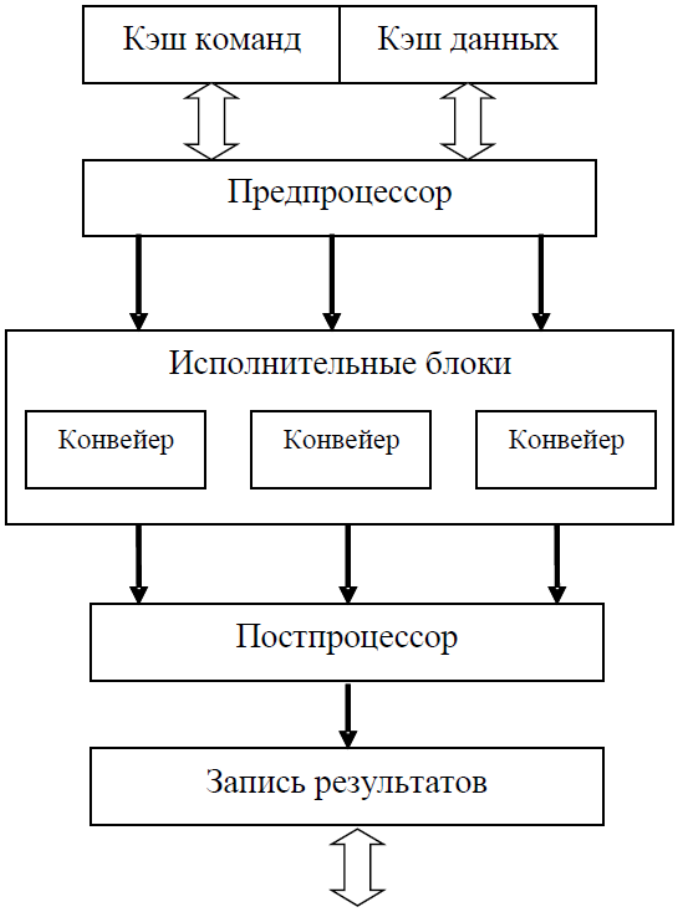


Схема классического процессора

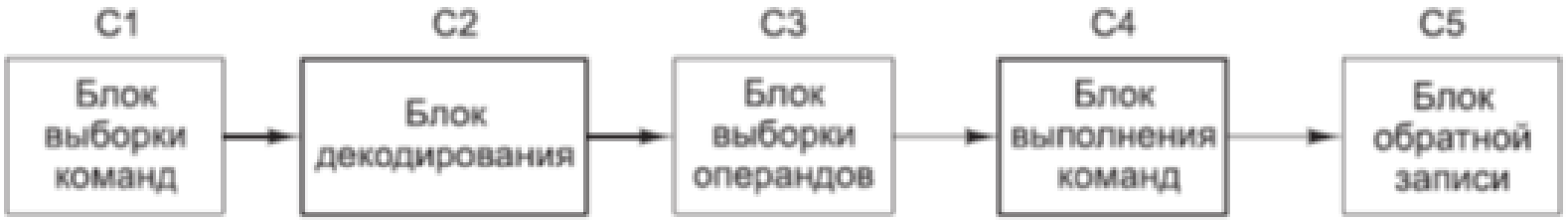
Конвейерная архитектура

- 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
  - В реальности CISC процессоры разбивают команды на несколько RISC микроопераций и выполняют их конвейером.
  - Реальные конвейеры могут содержать десятки суб-операций (суперконвейер)

Пятиступенчатый конвейер (*а*); состояние каждой ступени в зависимости от количества пройденных циклов (*б*). Показано 9 циклов

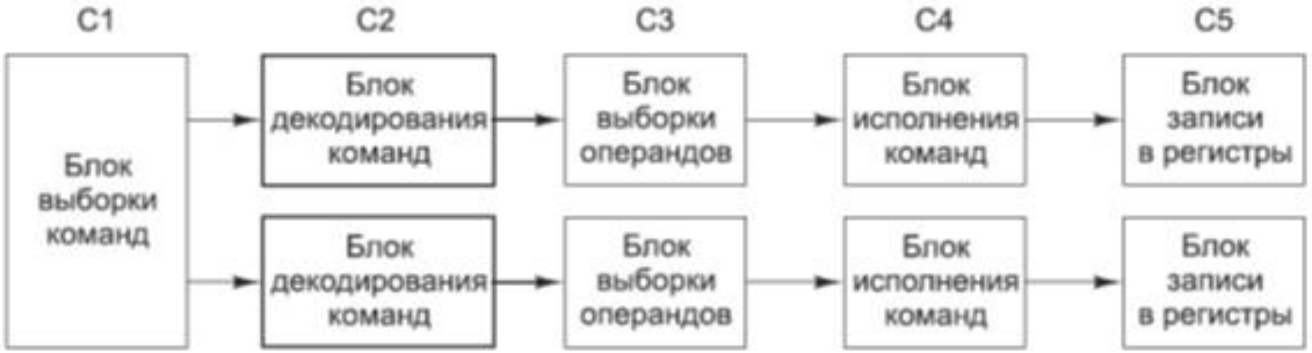
Конвейерная архитектура – предсказание переходов

- Конвейеризация – выполнение суб-операций одной операции так, чтобы все модули процессора были максимально заняты.
- Для работы конвейера нужно знать очередь операций, по этому если в программе есть условные переходы, то на каждом из них (если его результат неизвестен) нужно сбрасывать конвейер.
- Для того чтобы меньше сбрасывать конвейер переходы можно предсказывать или на уровне компилятора или динамически (подсчитывая вероятности переходов). Также могут быть исполнены сразу несколько ветвей переходов – т.н. спекулятивное исполнение инструкций.
- Чаще всего в наст. Время используется динамическое предсказание.
- Спекулятивный подход быстрее, но не безопасный.



Конвейерная архитектура многопоточность

- Многопоточность – выполнение операций в нескольких виртуальных ядрах.
  - многопоточность полезна когда одна из веток конвейера (или один из блоков в SMT) встала на длинной операции, вторая будет работать.
- Могут быть главный и зависимый конвейеры (дополнительный – может отключаться в силу небезопасности)

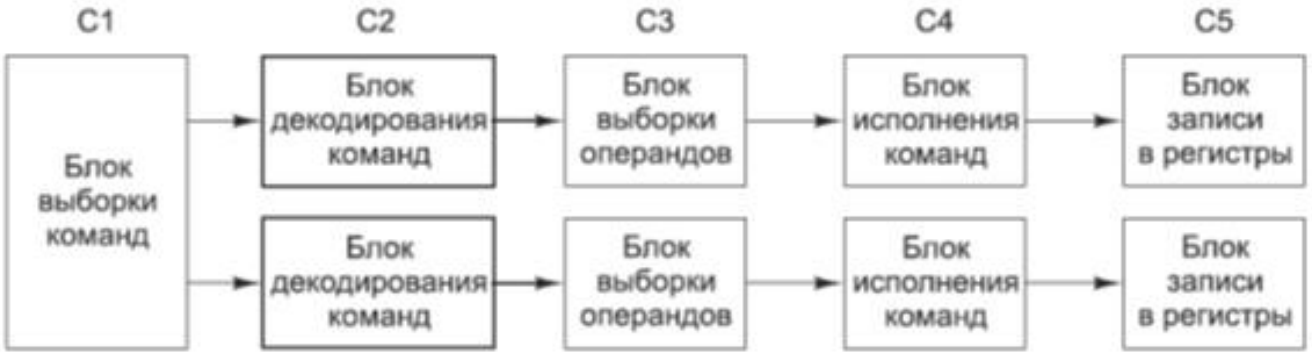


Сдвоенный пятиступенчатый конвейер с общим блоком выборки команд

Конвейерная архитектура многопоточность

- Каждый суб конвейер содержит свой набор АЛУ.
  - Команды не всегда выполнялись по заданному порядку – может быть внеочередное исполнение команд в порядке удобном процессору.

Многопоточность работает на уровне RISC микроопераций – то есть одна или несколько CISC команд могут быть рассмотрены как один поток.



Сдвоенный пятиступенчатый конвейер с общим блоком выборки команд

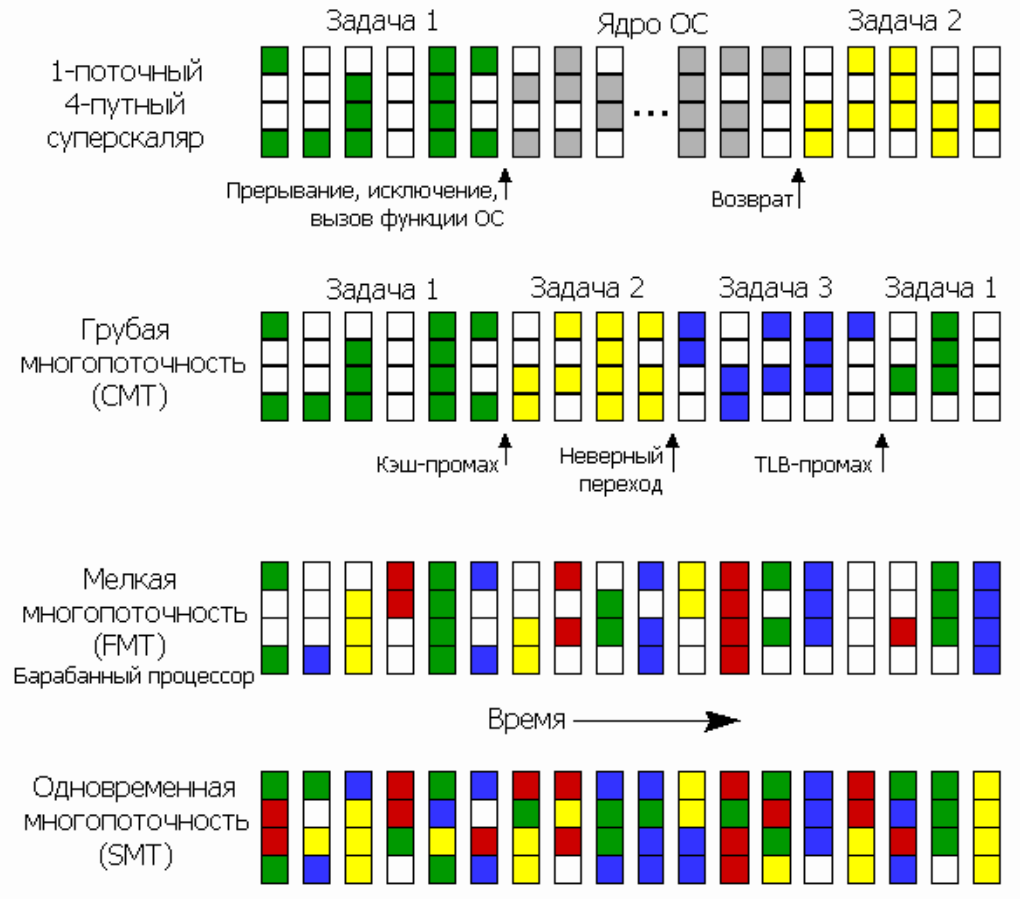
Конвейерная архитектура многопоточность

- Последовательная многопоточность
  - Грубо-зернистая – процессор выполняет поток пока не потребуется сложная суб-операция, затем процессор переключится на другой поток и будет ждать в нем возможности распараллелить суб-операции. – один поток на одном конвейере.
  - Мелко-зернистая – каждый такт процессор исполняет разные потоки (переключатеся между ними) – таким образом в среднем загрузка процессора максимальна – один поток может быть исполнен на нескольких конвейрах.
- Гиперпоточные или SMT (simultaneous multithreading) процессоры имеют общий блок выборки команд, который вызывает из памяти сразу по несколько команд из разных потоков. Распределение команд по конвейерам происходит динамически по мере того, когда они свободны.

Альтернативным к многопоточности являются однопоточный процессор (например суперскалярный или VLIW) и многопроцессорные скалярные однопоточные устройства.

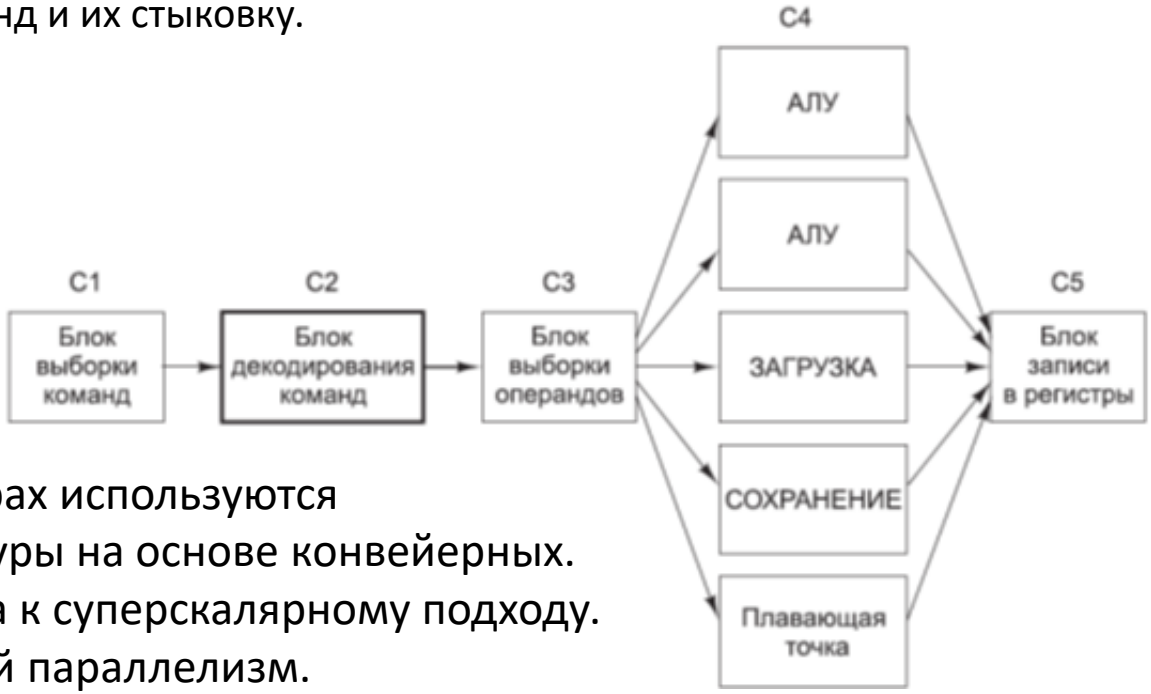
Конвейерная архитектура многопоточность

- Одновременная многопоточонсть может быть реализована только в суперскалярном процессоре и во VLIW процессоре.
- Последовательная многопоточность может быть реализована в любом процессоре с несколькими конвейерами или несколькими ядрами.
- Многоядерный процессор может быть рассмотрен как альтернатива многопоточному процессору – для процессора много потоков эквивалентны виртуальным ядрам.



Суперскалярная конвейерная архитектура

- Суперскалярными называют процессоры, способные запускать несколько команд (зачастую от четырех до шести) за один тактовый цикл.
- Параллельные команды не должны конфликтовать из-за ресурсов (например, регистров) и ни не должны зависеть от результата друг друга – для этого С1 блок выполняет контроль зависимостей команд и их стыковку.



В современных процессорах используются суперскалярные архитектуры на основе конвейерных.
VLIW подход альтернатива к суперскалярному подходу.
VLIW подход – статический параллелизм.
Суперскалярный подход– динамический параллелизм.
VLIW паралеллизм также использует многопоточный конвейер с предсказанием переходов, но наборы инструкций формируются компилятором.

Микроархитектура одноядерного процессора

Для большинства современных процессоров конвейер (микроархитектура) процессора включает

фронт-енд:

Захват CSIC инструкции (fetch) – красный блок

декодирование RISC микроопераций – фиолетовый блок).

И Бек-енд – datapath

– операции планирования инструкций в очередном и внеочередном порядке, разрешения зависимостей (бежевый блок)

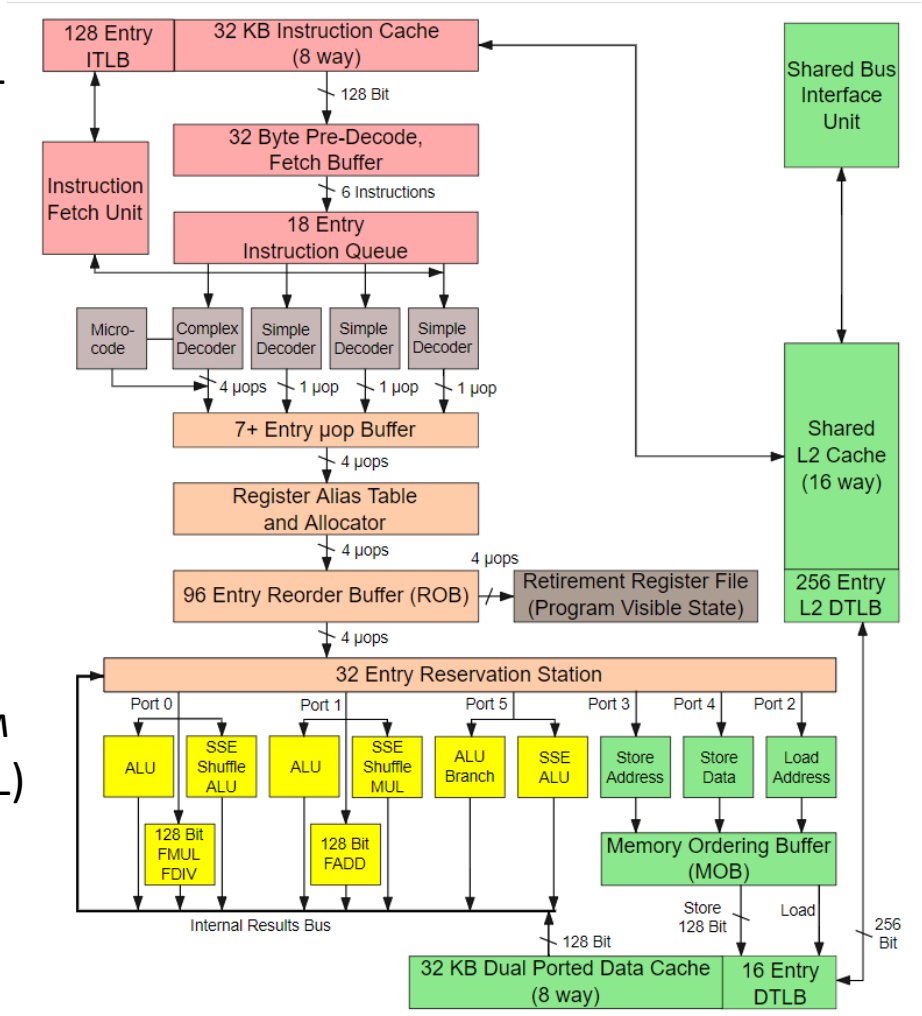
операции исполнения (желтый, в том числе сопроцессоры, напр. SSE, FMUL) или работы с памятью (зеленый).

Фрорнт-енд in-order

Бэк-енд суперскалярный out-of-order

Многопоточность может быть полная или только для бэк-енда

На иллюстрации один поток!



Микроархитектура intel core i7

Микроархитектура одноядерного процессора

Для большинства современных процессоров конвейер (микроархитектура) процессора включает фронт-енд

- Захват CSIC инструкции (fetch) – красный блок
- декодирование RISC микроопераций – фиолетовый блок).

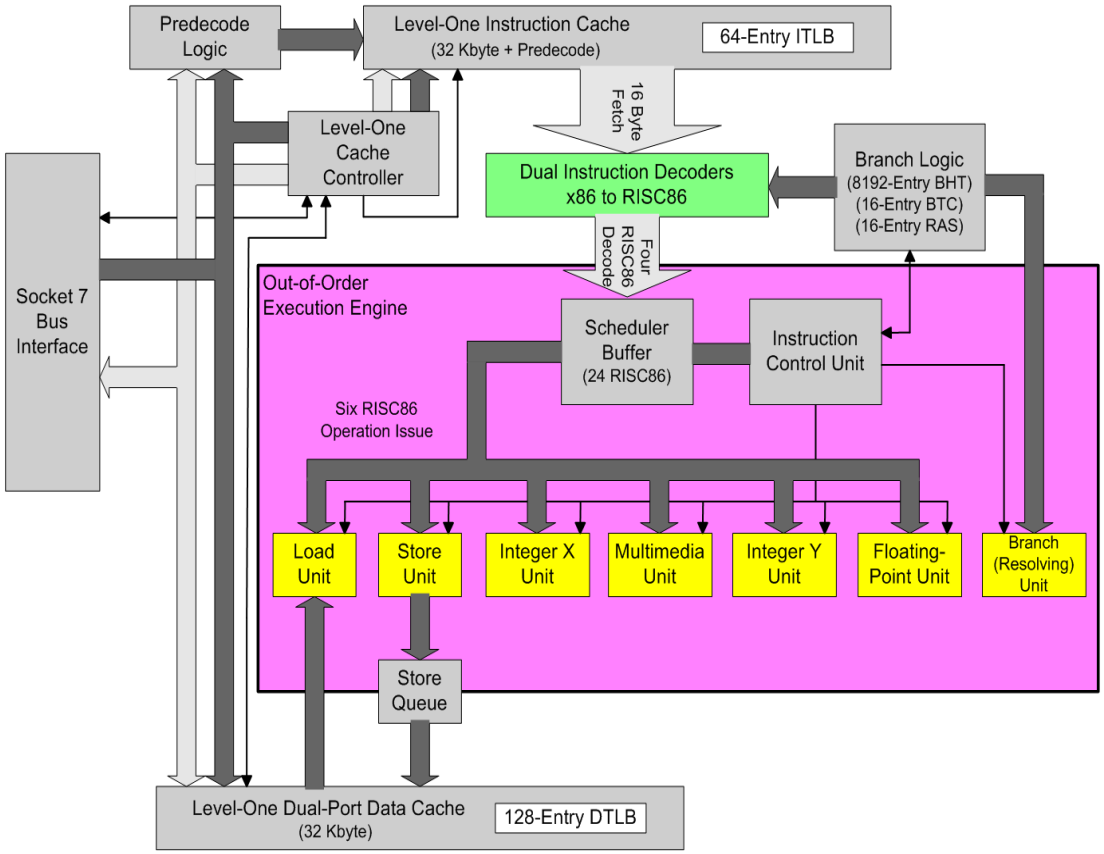
И Бек-енд – datapath

- – операции планирования инструкций в очередном и внеочередном порядке, разрешения зависимостей (бежевый блок)
- операции исполнения (желтый, в том числе сопроцессоры, напр. SSE, FMUL) или работы с памятью (зеленый).

Фрорнт энд in-order

Бэк-енд супрскалярный out-of-order

- Многопоточность может быть полная или только для бэк-енда
- На иллюстрации один поток!



Микроархитектура AMD K6

Параллелизм процессорных архитектур

Аппаратные средства телекоммуникационных систем.

Основные понятия и определения архитектуры вычислительной техники.

Архитектуры по степени параллелизма

Классификация Флинна – таксонометрия параллелизма на уровне данных.

SISD – скалярные процессоры

MISD – не используются

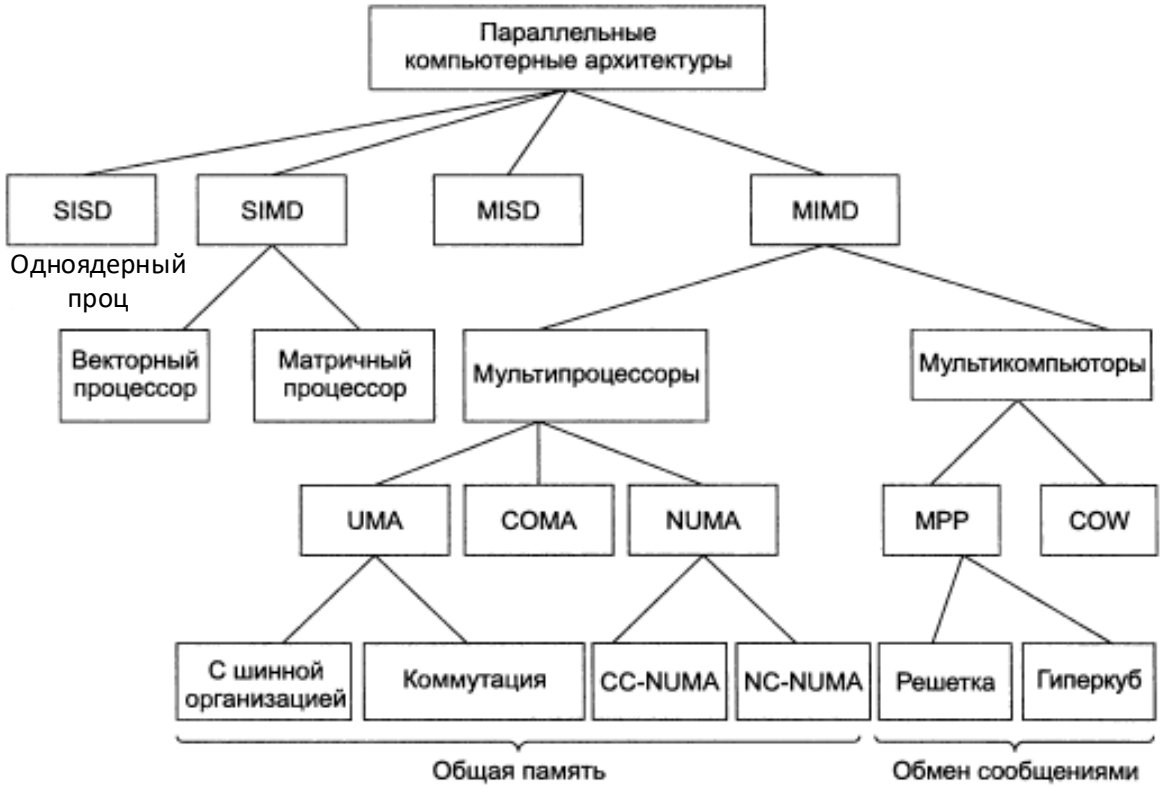
SIMD – векторная и матричная архитектуры

MIMD – многопроцессорные и многомашинные архитектуры

UMA— архитектура с однородным доступом к памяти - Часто представляет собой SMP (симметричный доступ к памяти процессоров на одной шине).

NUMA - с неоднородным доступом к памяти

COMA - с доступом только к кэш-памяти.



MPP— процессоры с массовым параллелизмом
COW – кластеры рабочих станций.
S – одно (single), M – много (many); D – данные (data), I – инструкции (instruction);
Например SIMD – single input many data

Архитектуры по степени параллелизма

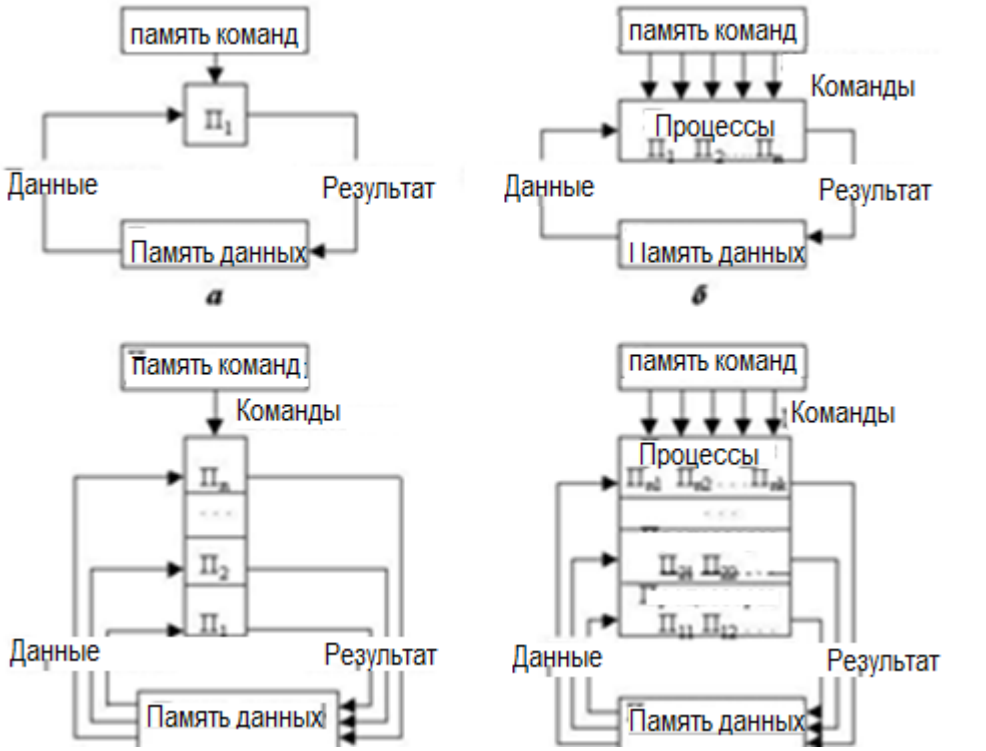
Классификация Флинна

SISD – скалярные процессоры, напр. «чистая архитектура Фон-Неймана».

MISD – не используются, только в некоторых спец. Применениях.

SIMD – векторная и матричные архитектуры (а также графические ускорители)

MIMD – многопроцессорные и многомашинные архитектуры

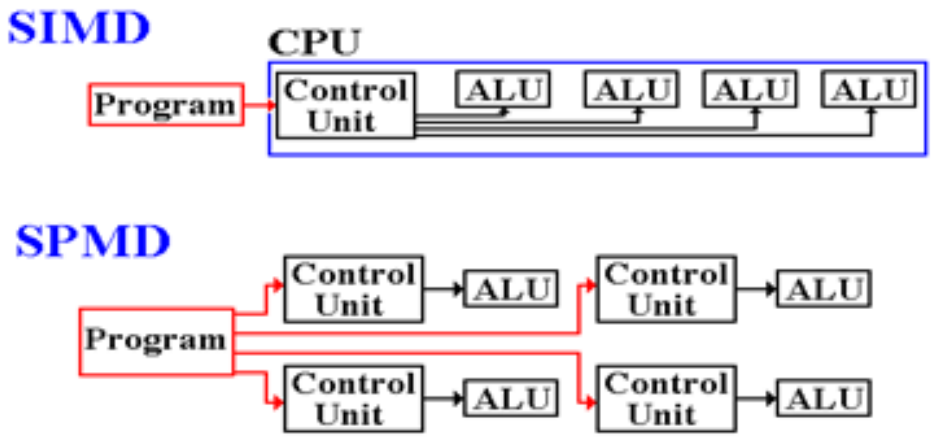


- *а* - SISD (однопроцессорная), *б* - MISD (конвейерная); *в* - SIMD (векторная); *г* - MIMD (матричная)

Параллельные процессоры

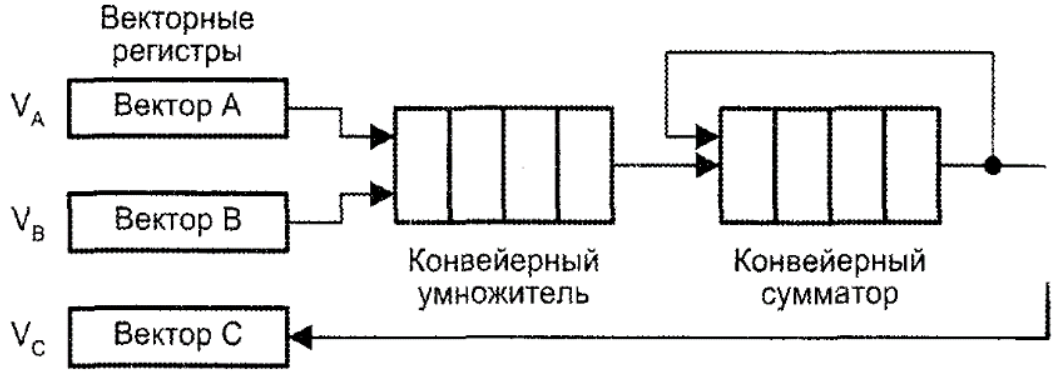
SIMD и SPMD-процессор (Single Instruction (Program)-stream Multiple Data-stream — один поток команд с несколькими потоками данных или программами) состоит из большого числа сходных процессоров, которые выполняют одну и ту же последовательность команд применительно к разным наборам данных.

- 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).



Параллельные процессоры

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.

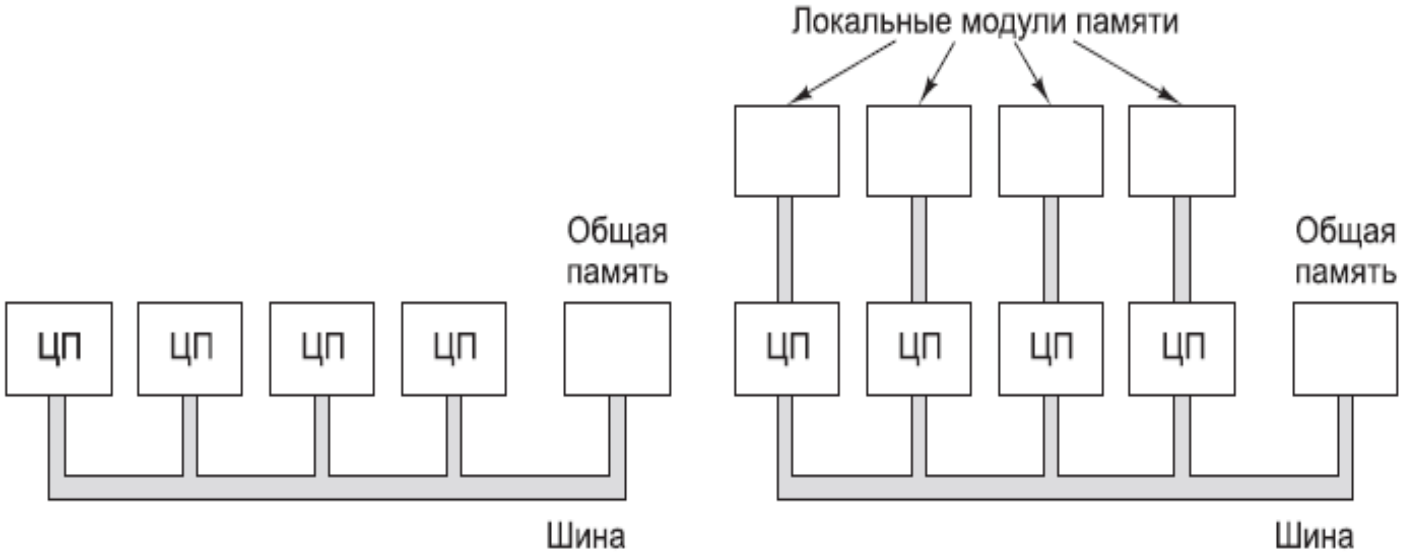


Мультипроцессоры

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 слаюо связанных компьютеров, у каждого из которых имеется собственная память называются **мультикомпьютерами**.



а мультипроцессор

б мультикомпьютер

Архитектуры по степени параллелизма

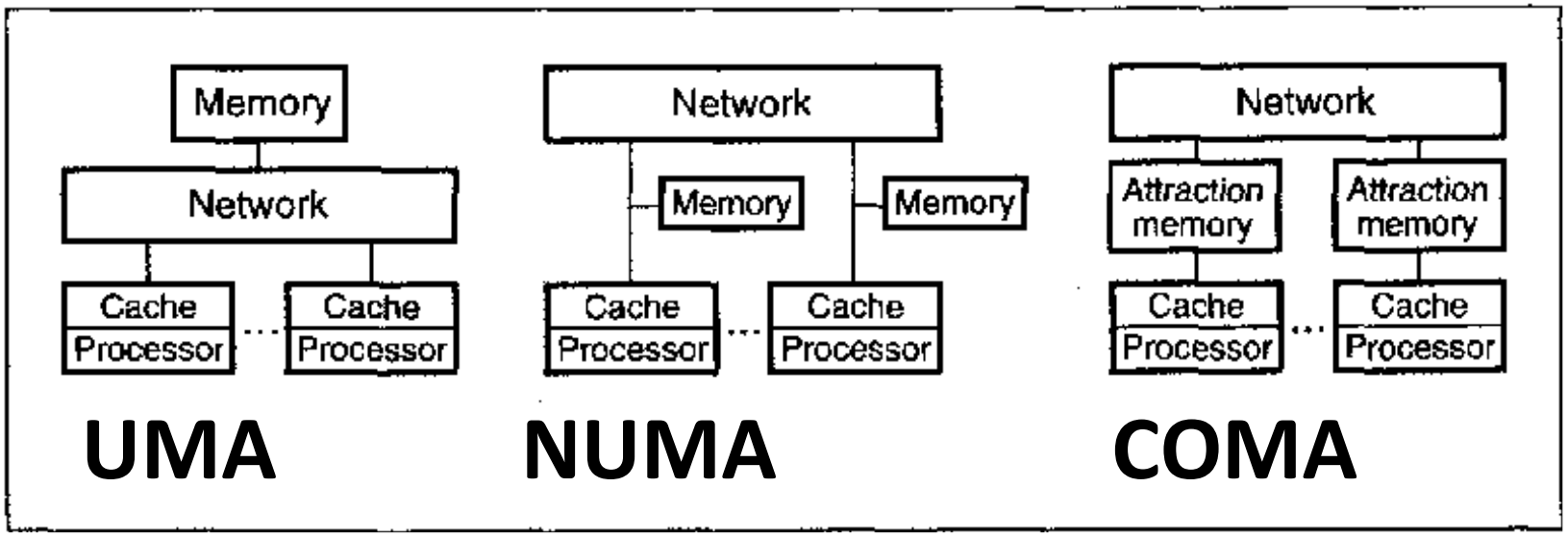
NUMA - Каждое ядро имеет свой кэш L1,L2, но все ядра объединены внутри процессорным интерфейсом и общим адресным пространством, а также общим кэшем – LLC (last level cache) (в данном случае L3).

CC-NUMA – все ядра обмениваются сообщениями об использовании данных в кэше с целью предотвращения ошибок доступа к данным.

Есть и другие варианты NUMA систем.

COMA системы представляют собой развитие CC-NUMA идей, но они не реализованы в коммерчески доступных системах,

UMA системы используются только в устаревших или дешевых устройствах (SMP).

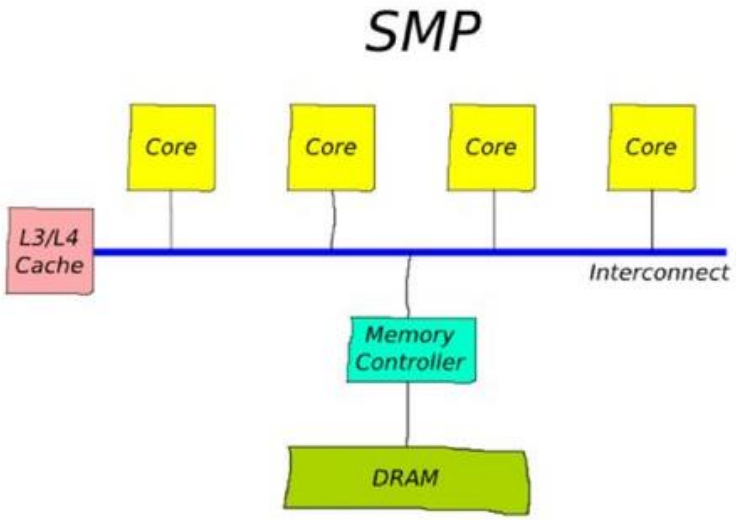


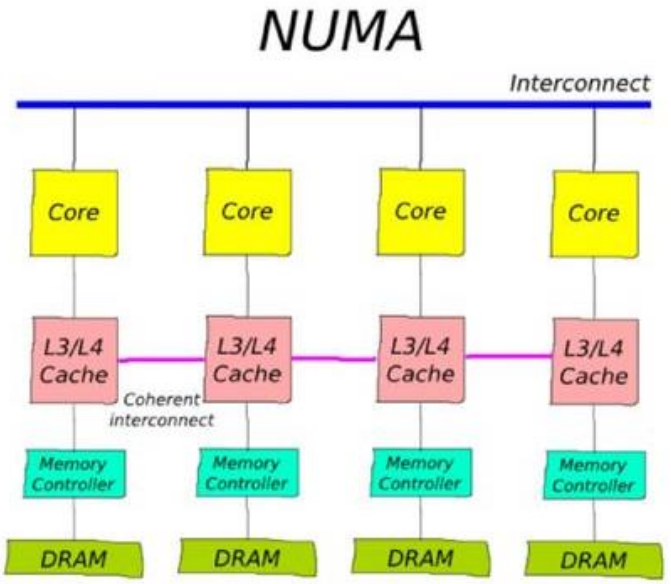
Архитектуры по степени параллелизма

На уровне многоядерного чипа современный процессоры CC- NUMA – MIMD многопроцессорные с системы с кэш-когерентным неоднородным доступом к памяти

Цель кэш-когерентности – предотвращение ошибок связанности данных, используемых в разных потоках/задачах/блоках инструкций – выполняемых разными устройствами.

Кэш-когерентность устанавливается на уровне аппаратного протокола обмена сообщениями между ядрами об используемом адресном пространстве.



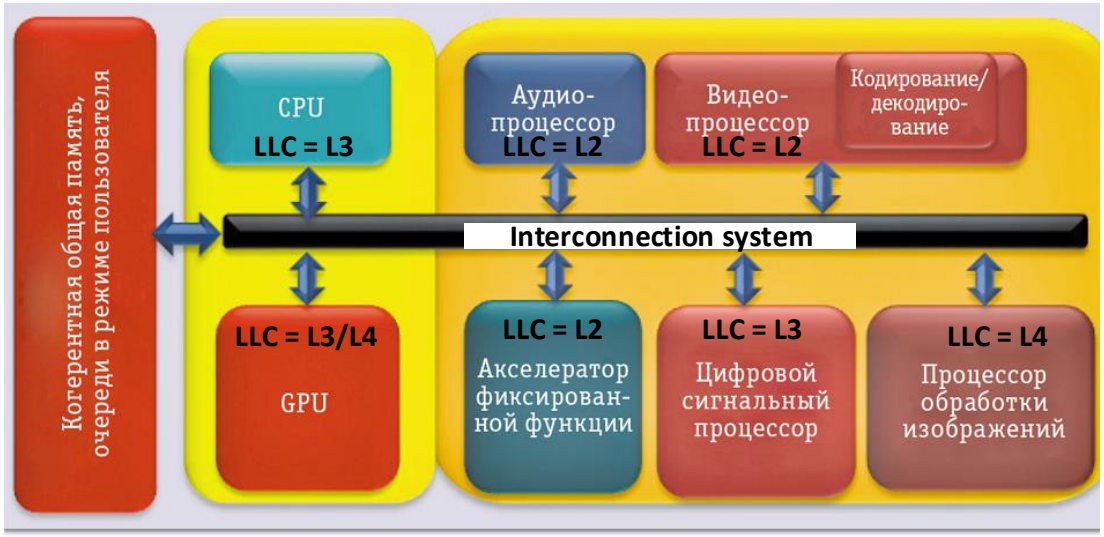


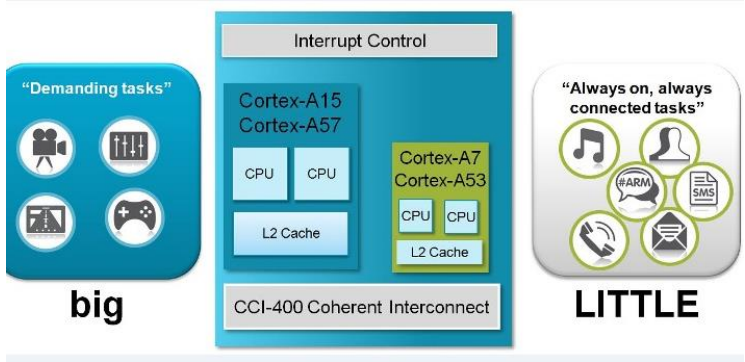
Гетерогенные NUMA архитектура

Часто современные процессоры представляют собой гетерогенные CC-NUMA системы – то есть системы в которых используется несколько блоков с различной структурой и с различными ISA. Например CPU и GPU, также в спец. устройствах DSP и сопроцессоры на основе FPGA – HAS гетерогенность.

В ряде процессоров big.Little гетерогенность, когда используется два типа ядер с одним ISA но разной структурой, например одно быстрое и энергопотребляющее для периодов большой загрузки, а другое медленное для периодов когда загрузки нет (используется в процессорах мобильных устройств).

В гетерогенных системах все устройство объединены в единое адресное пространство на уровне LLC (может быть разным уровнем для разных модулей).





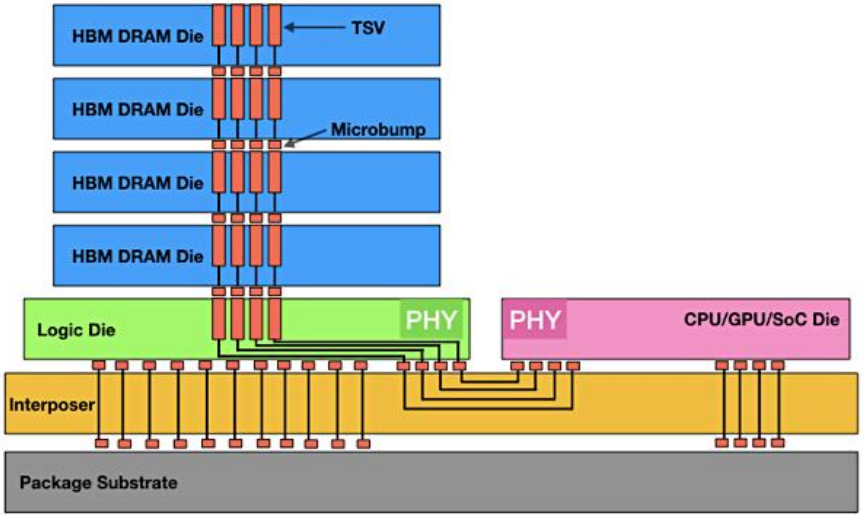
Особенности архитектуры современных процессоров

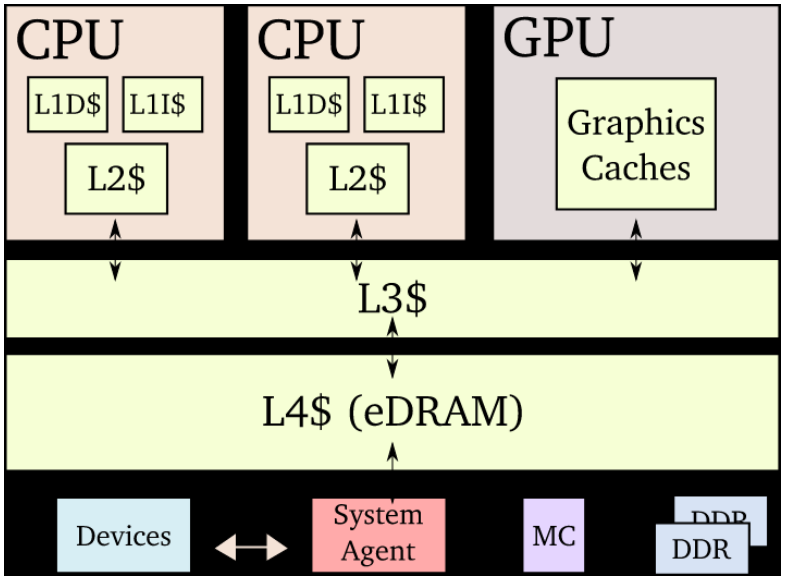
Аппаратные средства телекоммуникационных систем.

Основные понятия и определения архитектуры вычислительной техники.

Особенности современных архитектур процессоров

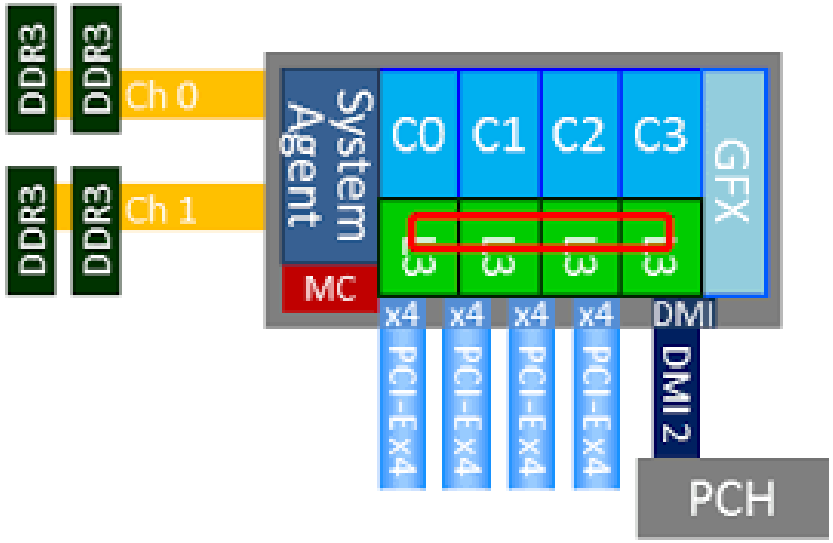
- Увеличение объема КЭШ памяти и уровней Кэш-а, а также отдельные кэши данных и инструкций.
  - А также добавление eDRAM модулей, а также HBM модулей быстрой памяти и модулей энергонезависимой флэш памяти (NVRAM, NVDIM).
  - Данный тренд направлен на решение проблем основного узкого места ЦПУ латентности доступа к основной памяти. За счет таких многоуровневых асинхронных решений повышается вероятность т.н. попадания в Кэш – то есть вероятность того, что нужные данные в нужный момент будут в кэш памяти.





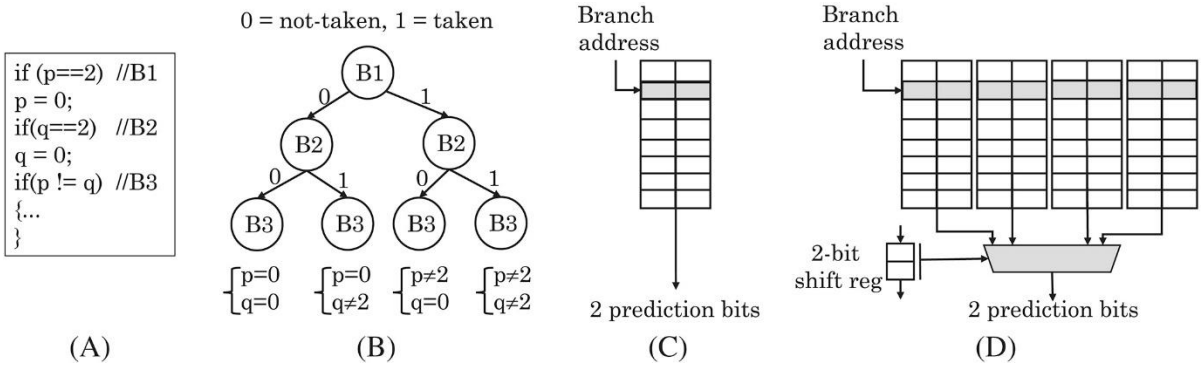
Особенности современных архитектур процессоров

- **Встроенные контроллеры доступа к памяти (MCU)** - оптимизация работы с ОЗУ.
- **Встроенный контроллер доступа к устройствам ввода-вывода и видеопамяти** (System Agent, uncore) - оптимизация работы с видеопамятью и другими внешними устройствами (напр. подключение дисплея), а также интеграция работы с твердотельными накопителями.
- **Использование гетерогенных CC-NUMA систем со специализированными модулям для широкого перечня задач**.



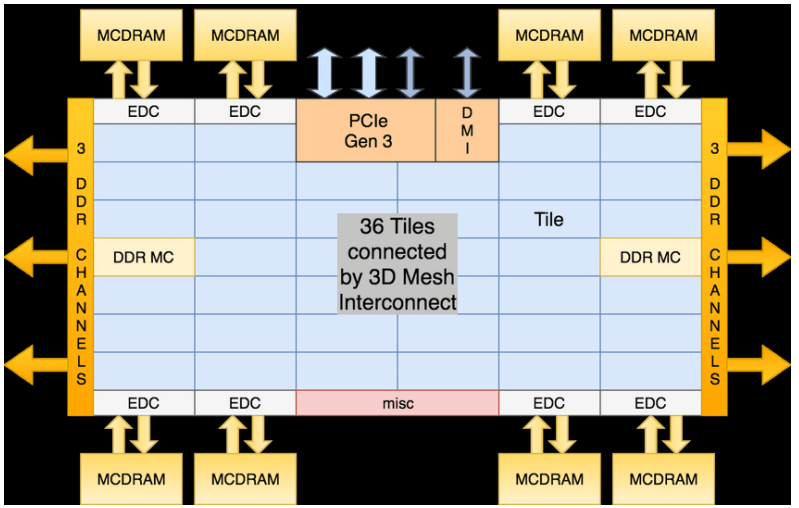
Особенности современных архитектур процессоров

- **Переход от спекулятивного выполнения команд к динамическому предсказанию ветвлений и повышения точности предсказаний.**
  - 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 актаки типа Spectra, meltdown и т.п.
  - Динамические предсказания ветвлений сегодня осуществляются при помощи нескольких предикторов, как правило построенных на принципах машинного обучения или простой статистики (в первых уровнях предсказаний).

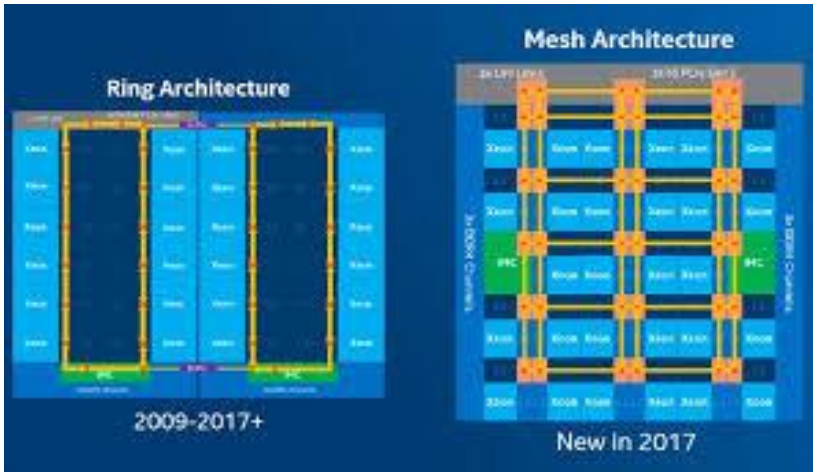


Особенности современных архитектур процессоров

- Уменьшение технологии производства p-n переходов
  - На данный момент технологии 5,7,10 нм.
- Увеличение количества ядер процессора.
  - На данный момент частота работы ядра почти на пределе, увеличение частоты ведет к повышению нагрева, необходимости повышения напряжения или снижению отношения сигнал-шум и сигнал-помех и другим эффектам, которые приводят к пробоям и помехам особенно для нанометровых p-n переходов.
  - Современные серверные процессоры могут иметь 16-72 ядер, каждое по 2-4 потока (до 288 потоков в Xenon Phi).



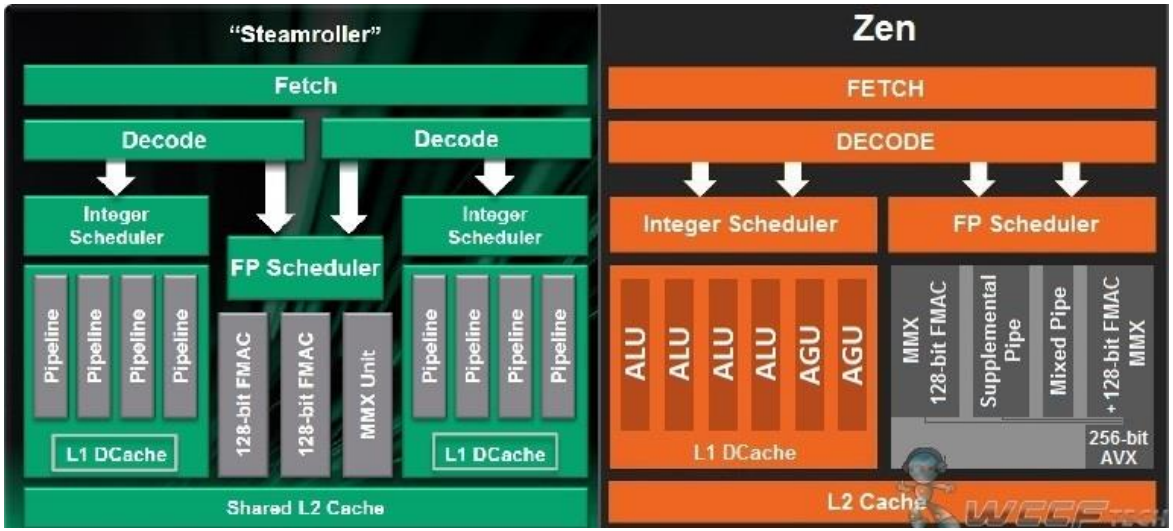
Intel Xenon Phi Knights Landing



Варианты Ring и Mesh архитектур

Особенности современных архитектур процессоров

- Повышение числа ступеней конвейеризации и функциональных модулей суперскалярной микроархитектуры.
- Одновременное выполнение двух-четырех 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.

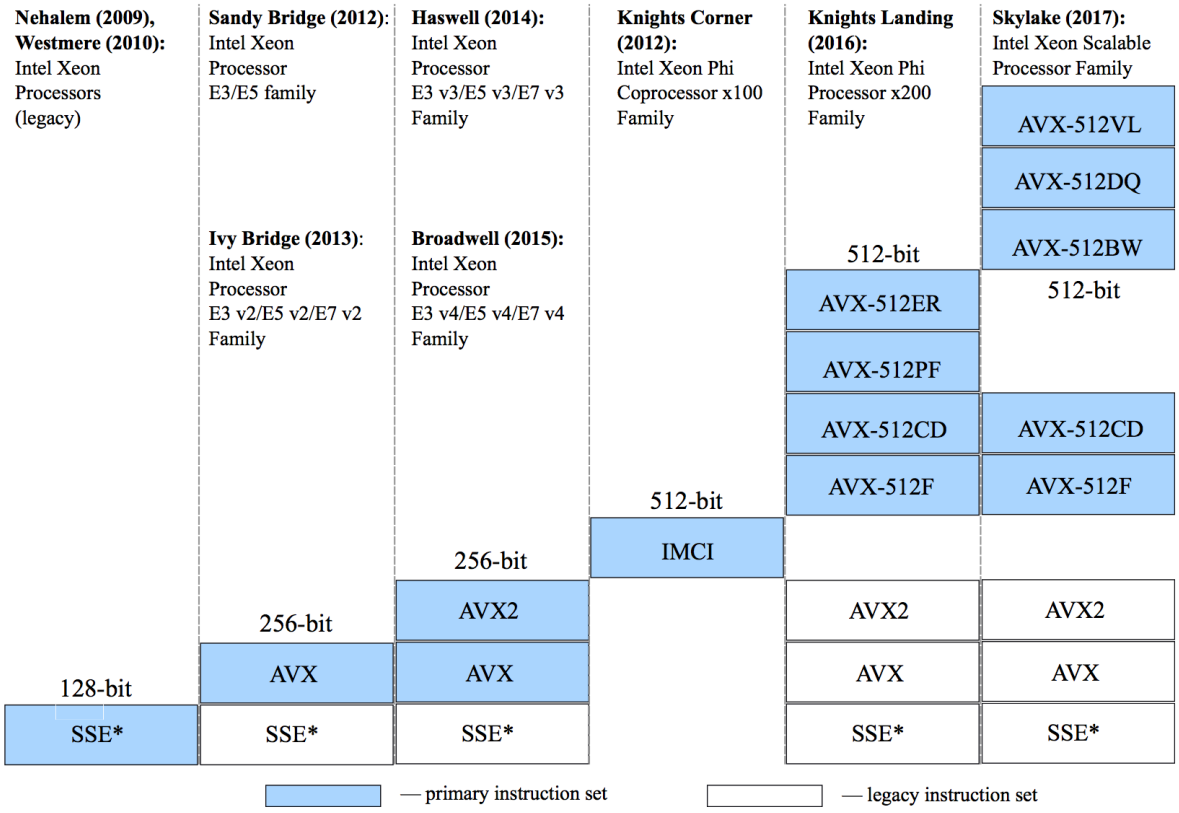


Суперскалярные архитектуры AMD Steamroller и Zen с двумя явными SMT потоками

Особенности современных архитектур процессоров

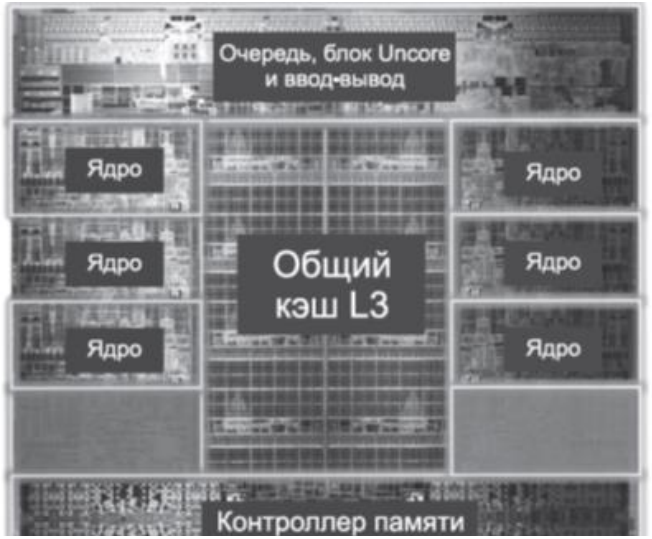
- Система команд X86-X64 (AMD x64) – расширенная система команд с 64 битной адресацией и расширенные системы команд SSE, XMM, FMA для ускорения вычислений с высокой степенью упорядоченности (для массивов)— например, обработки мультимедийных и научных данных.

В данным момент наиболее современные системы команд это AVX-256 и AVX-512, а также AMX (анонсированы на 2021 год).

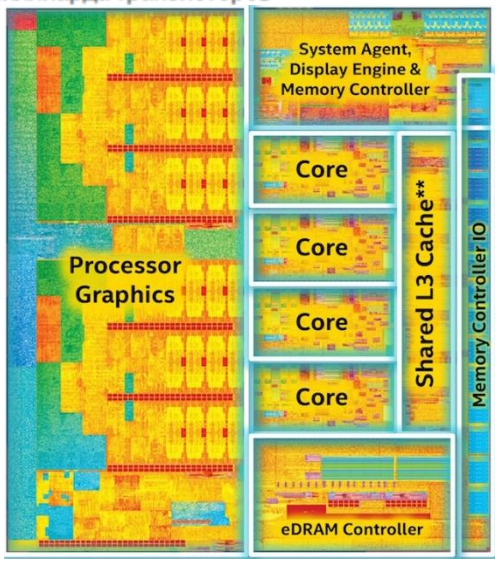


Архитектура современных процессоров Intel

- 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),
  - последовательная кэш-шина типа точка-точка для соединения процессоров и для передачи данных между процессором и системной платой.
  - Сегодня QPI заменен на UPI – расширенную версию QPI
  - управления энергопитанием (powermanagement),
  - встроенный графический контроллер.



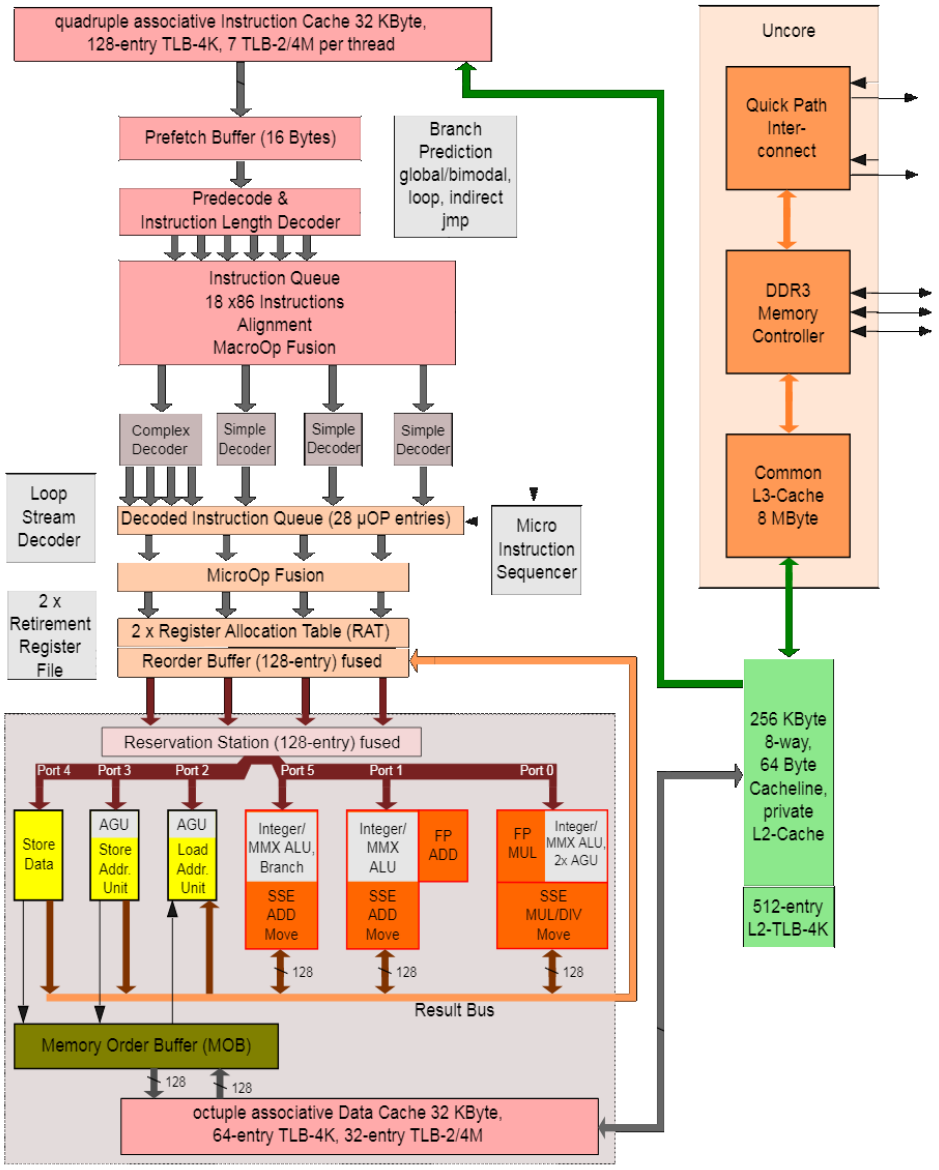
Микросхема Intel Core i7-3960X. Подложка имеет размеры 21×21 мм и содержит 2,27 миллиарда транзисторов



Intel 7-6x die with GPU

Архитектура современных процессоров Intel

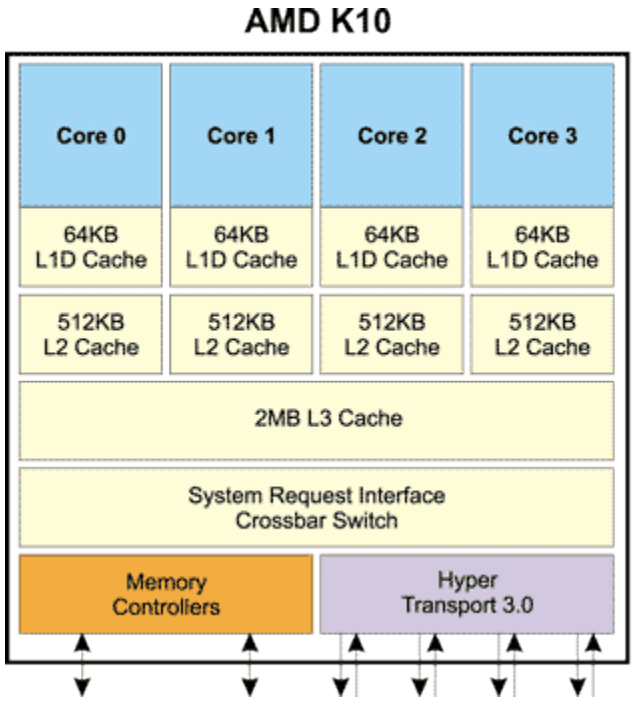
- Некоторые процессоры содержат блок - *Системный агент (System agent)* - содержит многоканальный контроллер памяти, «мосты» PCI-Express, DMI, дисплейные интерфейсы, блок аппаратного декодирования видео.

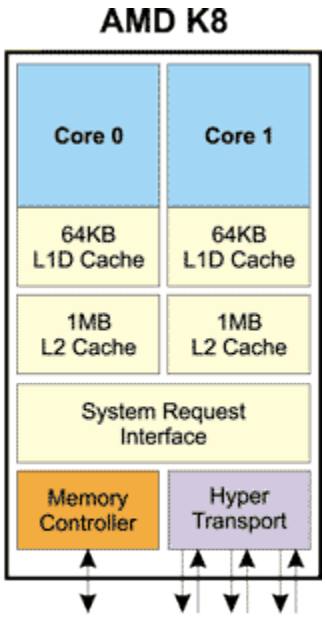


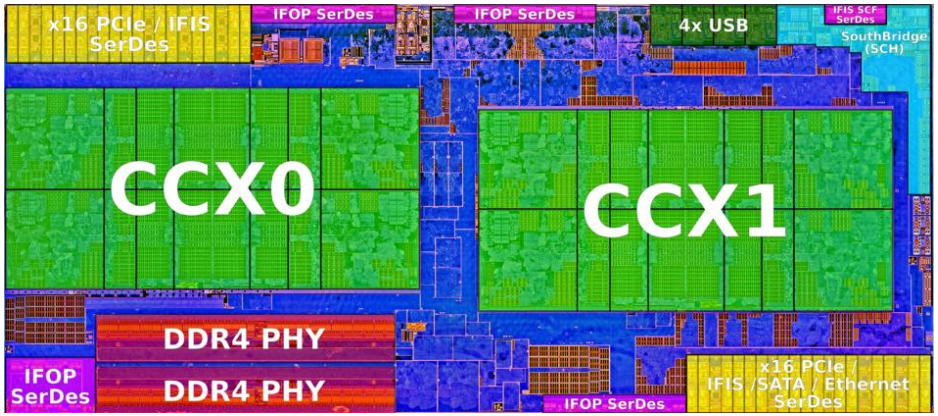
Архитектура ядра Intel Core I7-3x

Архитектура современных процессоров AMD

- технология HyperTransport (HT) для создания многопроцессорных системы, и объединять на одном кристалле несколько ядер (AMD Opteron).
  - –Аналог QPI от Intel, но последовательная (появилась раньше)
  - –Развитие HT – система **Infinity Fabric**
- Каждое ядро работает на своей частоте.
- Наборы ядер объединены в UMA CCX блоки. Каждый NUMA процессор это несколько CCX блоково.







AMD zen die

Архитектура процессоров Intel IA64

Используется в серверных процессорах Itanium

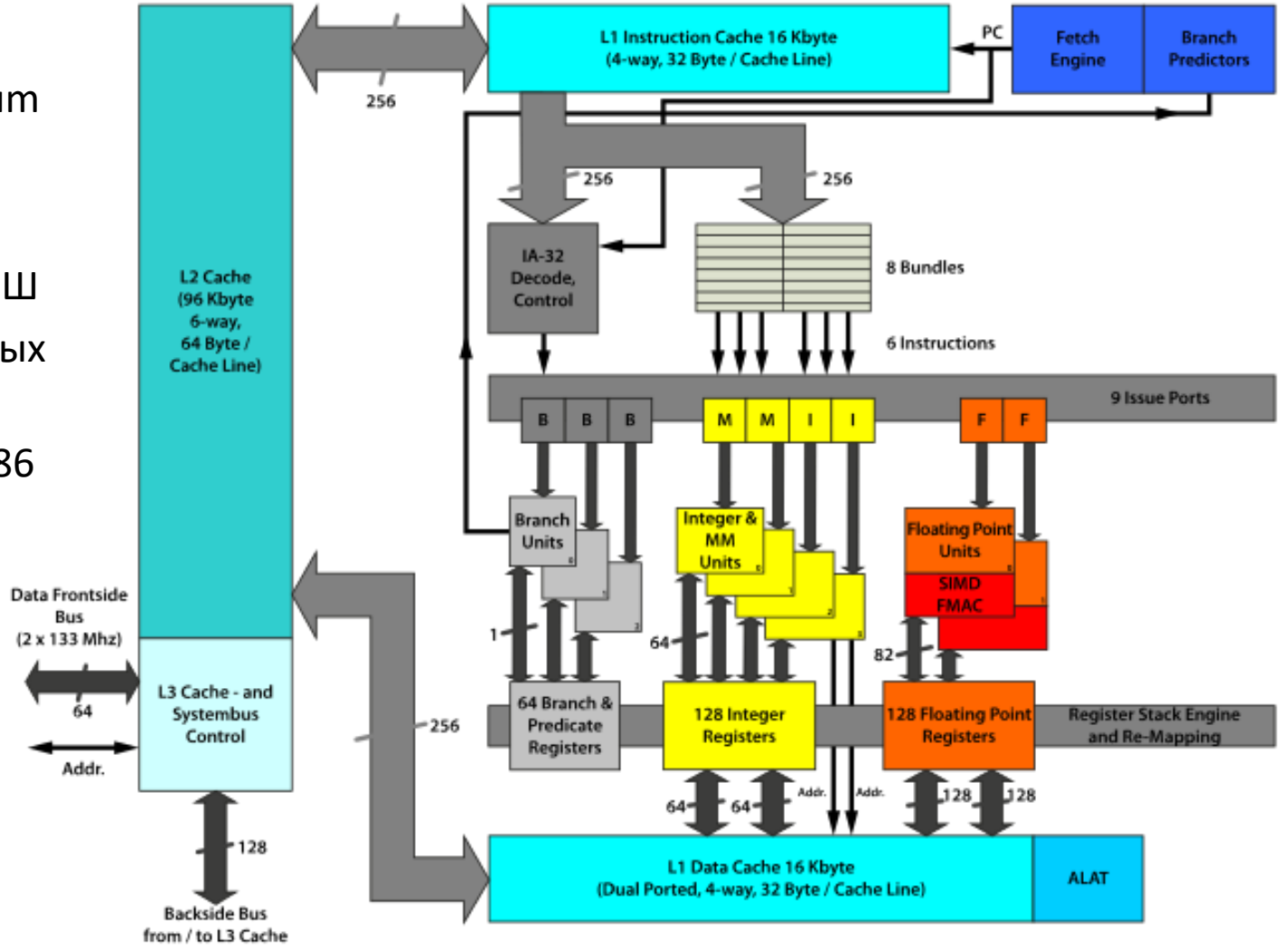
(Архитектура EPIC)

VILW архитектура

Спекулятивный КЭШ

Исполнение простых команд по 3

Несовместима с X86 и X64



Графические процессоры

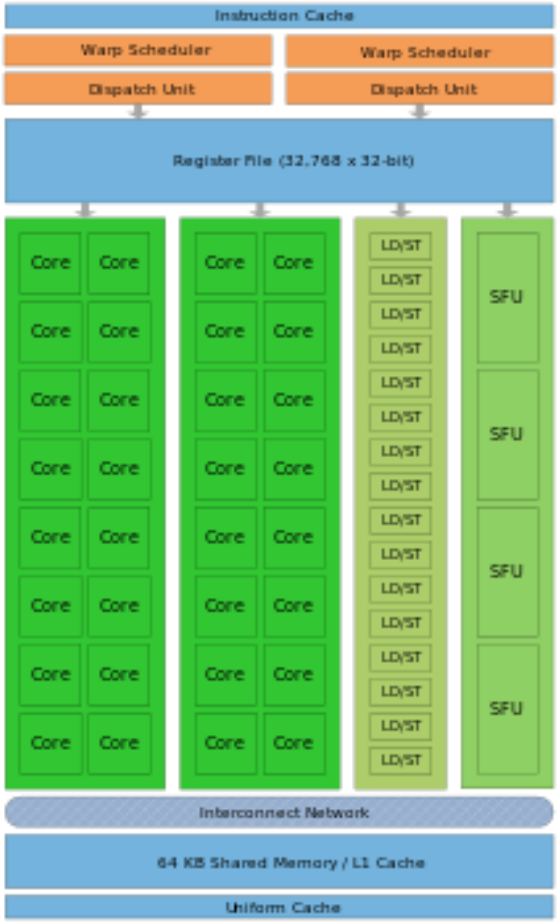
Аппаратные средства телекоммуникационных систем.

Основные понятия и определения архитектуры вычислительной техники.

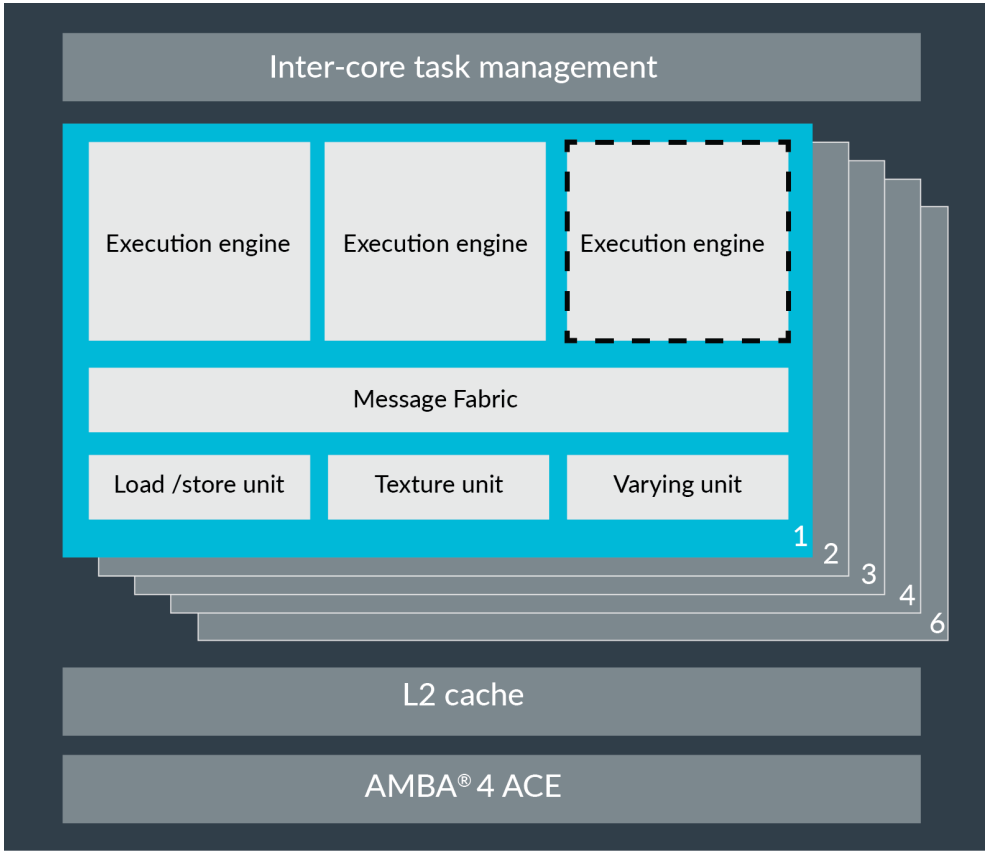
Архитектура графических процессоров

• Графический процессор (GPU) Содержат набор одинаковых вычислительных устройств (потоковых процессоров, ПП), работающих с общей памятью ГПУ (видео ОЗУ)

•



Nvidia Fermi



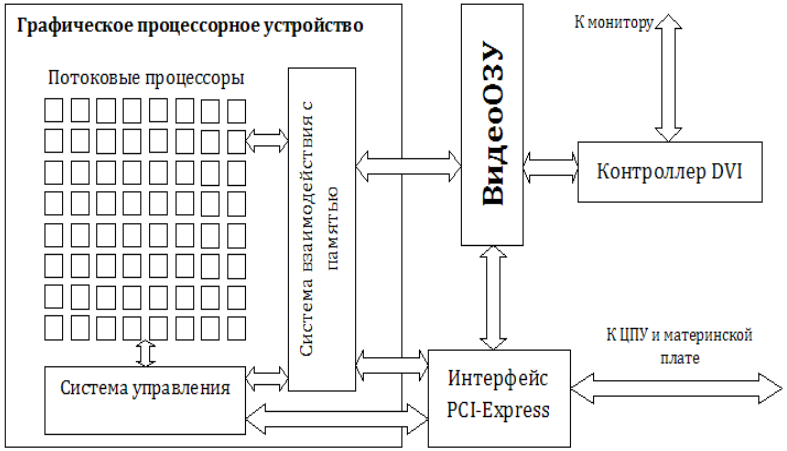
ARM G51 Mali

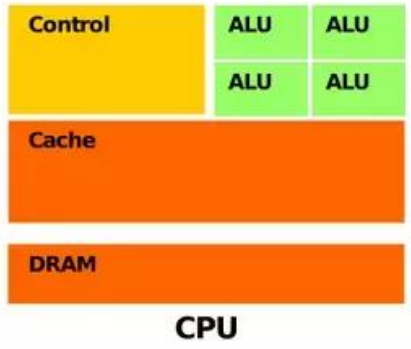
Архитектура графических процессоров

•Могут быть GPU обработки графики (текстуры, шейдеры, ray tracing) и т.н. GPU общего назначения (GPGPU) используемые для научных, инженерных расчетов и в задачах машинного обучения.

– И графические и научные операции как правило сводятся к операциям линейной алгебры, а те к операциям сложения с умножением массивов данных.

– Некоторые, особенно ранее GPU содержали специальные модули расчета шейдеров и текстур – в современных GPU часто их заменяют на доп. Блоки общей арифметики (сложение с умножением).

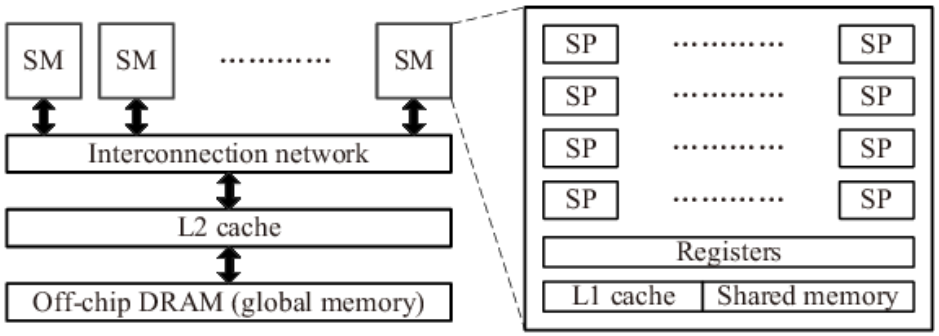




Архитектура графических процессоров

•Все потоковые процессоры (ПП) GPU синхронно исполняют один и тот же набор команд (конвейер, шейдер, умножение матриц, свертка).

- ГПУ выполняют операции асинхронно и только с данными во внутреннем ОЗУ (GDRAM или HBM иначе доступ к внешней ОЗУ все бы затормозил)
- Доступ к ОЗУ узкое место GPU – из за латентности доступа невыгодно использовать GPU для небольших массивов данных.
- Часто GPU содержат многоуровневый кэш для снижения латентности.
- В современных интегрированных GPU (в составе HSA CC-NUMA) LLC GPU объединён с CPU.
  - В устаревших версиях адресные пространства делились и приходилось выполнять доп. Операцию передачи данных в интегрированные GPU и обратно.
  - Компанией IBM поддерживается CC-NUMA интеграция с GPU NVIDIA за счет общего интерфейса для обоих систем NVLINK



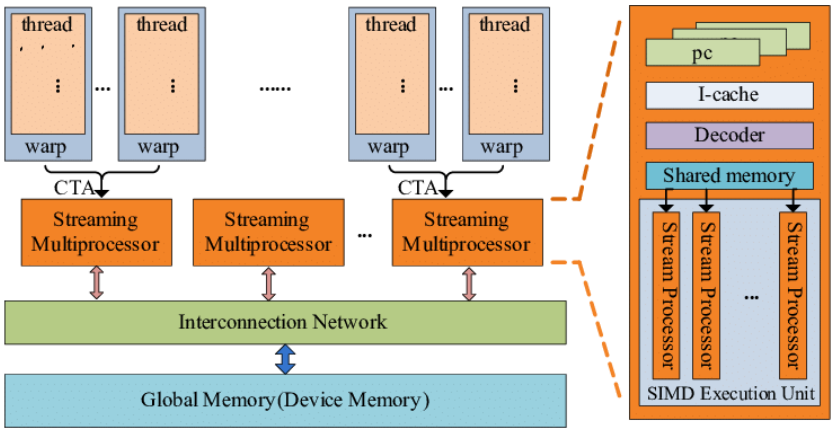
Архитектура графических процессоров

•По существу GPU это потоковая SIMD архитектура – т.н. SIMT архитектура.

•GPU всегда управляется при помощи CPU (GPU это именно сопроцессор). Задачи включают блоки команд и данные для них.

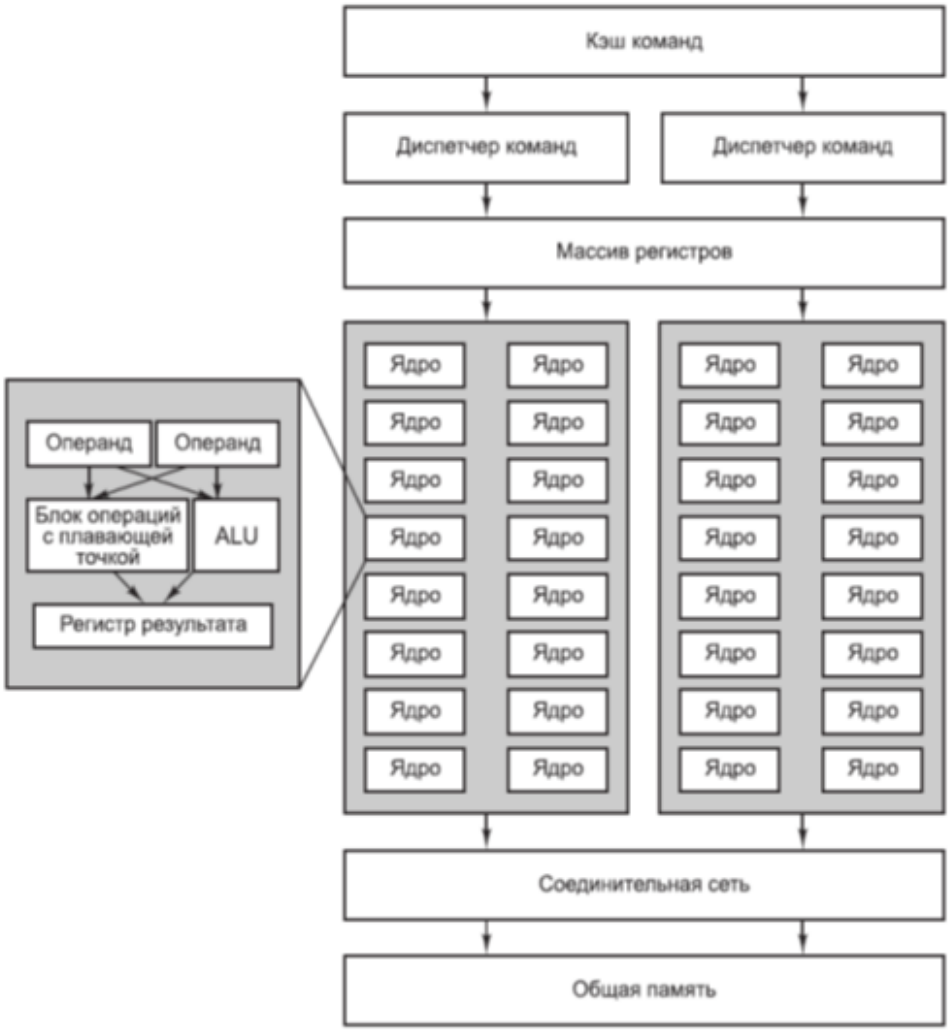
• Все ПП синхронно исполняют один и тот же набор команд (конвйер, шейдер).

- •То есть одна команда исполняется для целого массива данных.
- •Раньше шейдеры были аппаратными, теперь программные – представляют собой набор микрокоманд - конвейер.
- •ПП объединяются в блоки (связки, wrap, wavefront) по 16-64 ПП.
- •Каждый графический процессор запекает за один только одну связку.
- •Связка работает несколько циклов работы GPU.
- •Использование нескольких связок в одному GPU позволяет скрывать латентность доступа к ОЗУ/кэшу данных.



Особенности архитектуры графических процессоров

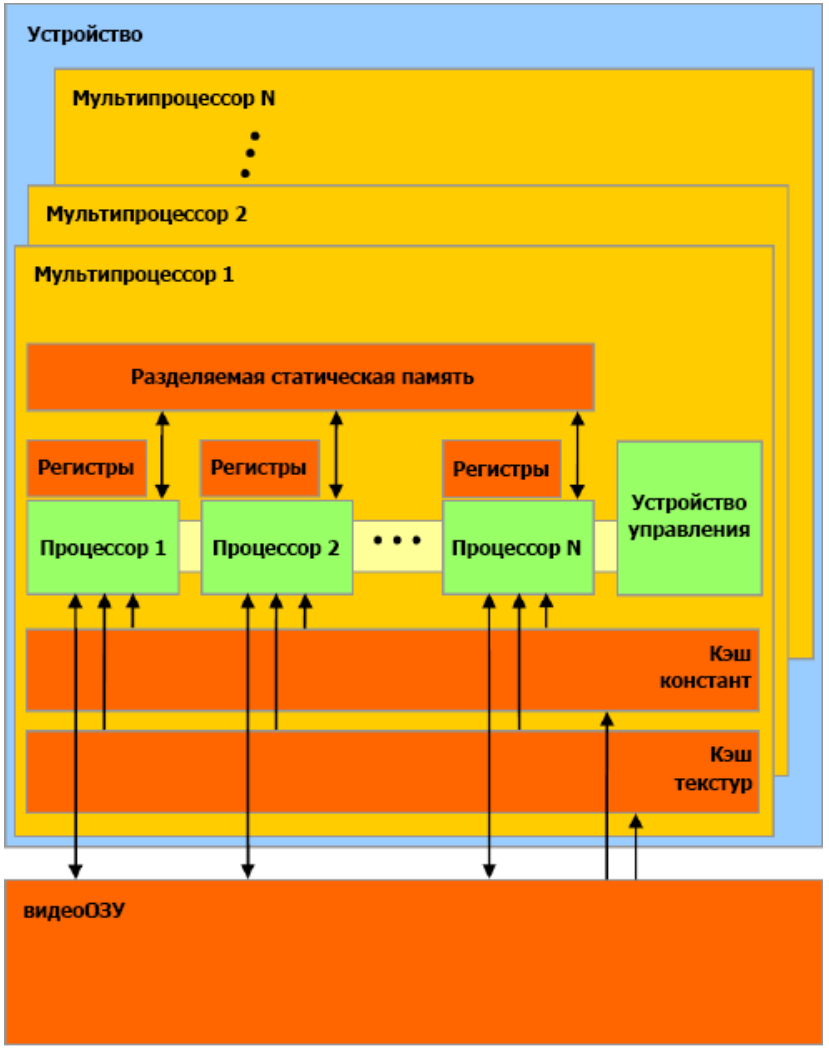
- Все ПП синхронно исполняют один и тот же набор команд
- Основная проблема ГПУ - Низкая скорость чтения из ОЗУ (высокая латентность).
  - чтобы его компенсировать, требуется обрабатывать большое (10000 и более) количество элементов за 1 запуск.
- Поддержка аппаратной многопоточности.
- Разнородность архитектур.
  - Следует учесть, что оптимальные программы для NVidia и AMD будут сильно различаться.



SIMD-ядро графического процессора Fermi

Архитектура современных графических процессоров

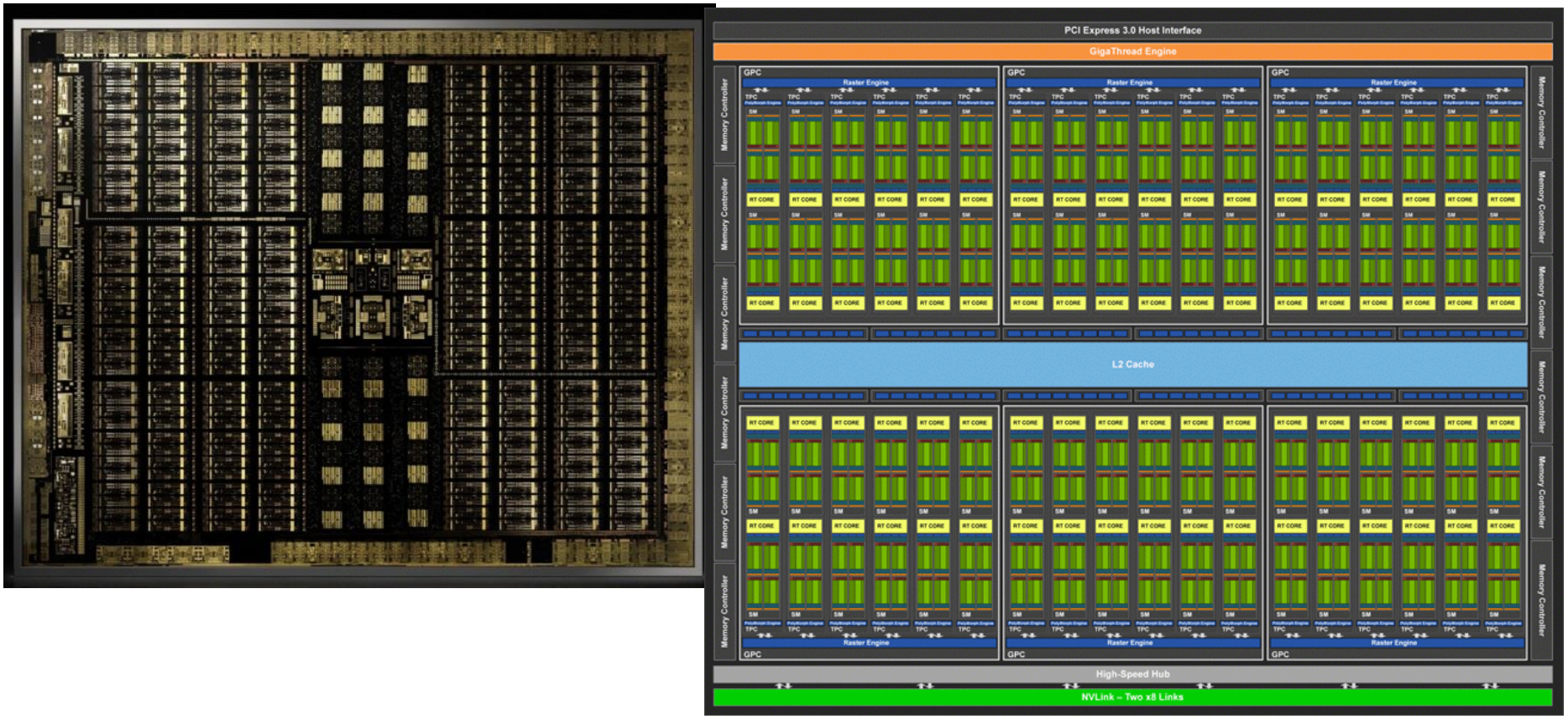
- 128 потоковых процессоров(ПП) объединены в SIMD-группы по 16 мультипроцессоров (МП),
  - –МП Также называется кластером или слайсом.
- МП работают независимо друг от друга, хотя и исполняют один и тот же набор команд.
- Каждый ПП является суперскалярным устройством и может конвейеризировано выполнять несколько команд.
- 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

Архитектура современных графических процессоров

.



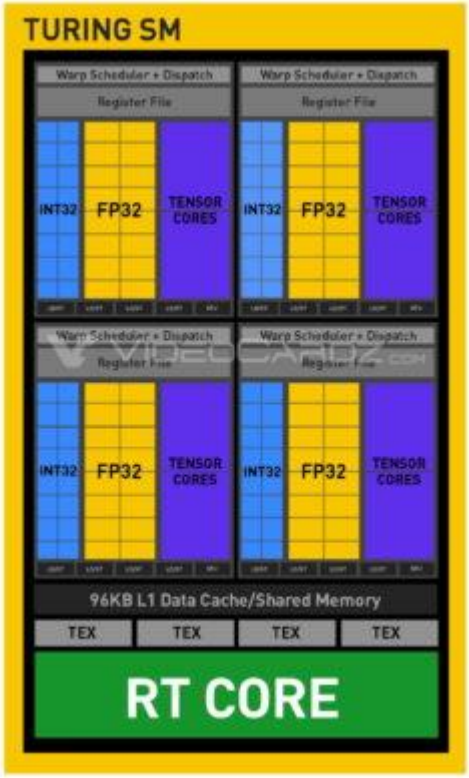
NVIDIA TU102 (*GeForce RTX 2080 Ti)* Фото кристалла и его схема

6 кластеров GPC, каждый по 6 текстурно-процессорных кластеров TPC, объединяющих мультипроцессорные блоки SM. Каждый SM-блок 64 вычислительных блока (CUDA-cores). Шина памяти GDDR6. Объем памяти 11 ГБ на уровне старого флагмана.
12 контроллеров памяти разрядностью 32 бита. Кэш L2 5632 КБ.

Архитектура современных графических процессоров

Схема мультипроцессорного блока NVidea Turing

Используются параллельные вычисления в FP32 и INT32, тензорные вычисления и блоки расчет текстур (tex), и лучей (RT core).
FP – векторные – SIMD вычисления.
INT – спец. Задачи – ветвления, переходы, расчет адресов операндов.
тензорные блоки поддерживают операции в форматах Int1, Int4, int8, Fp16, FP32, а также операции смешенной точности (FP16 умножение + Fp32 аккумулятор).
тензорные блоки используются в задачах машинного обучения для оптимизации ресурсов.



Архитектура NVidia Turing 104 (GeForce 20xx)

Каждое тензорное ядро выполняет до 64 операций с плавающей запятой, используя входные данные формата FP16.
Тензорные блоки позволяют работать с матицами или блоками матриц, например разменом 8x8.
Основная операция FMA (float multiply and Add).

Специальные типы процессоров

Аппаратные средства телекоммуникационных систем.

Основные понятия и определения архитектуры вычислительной техники.

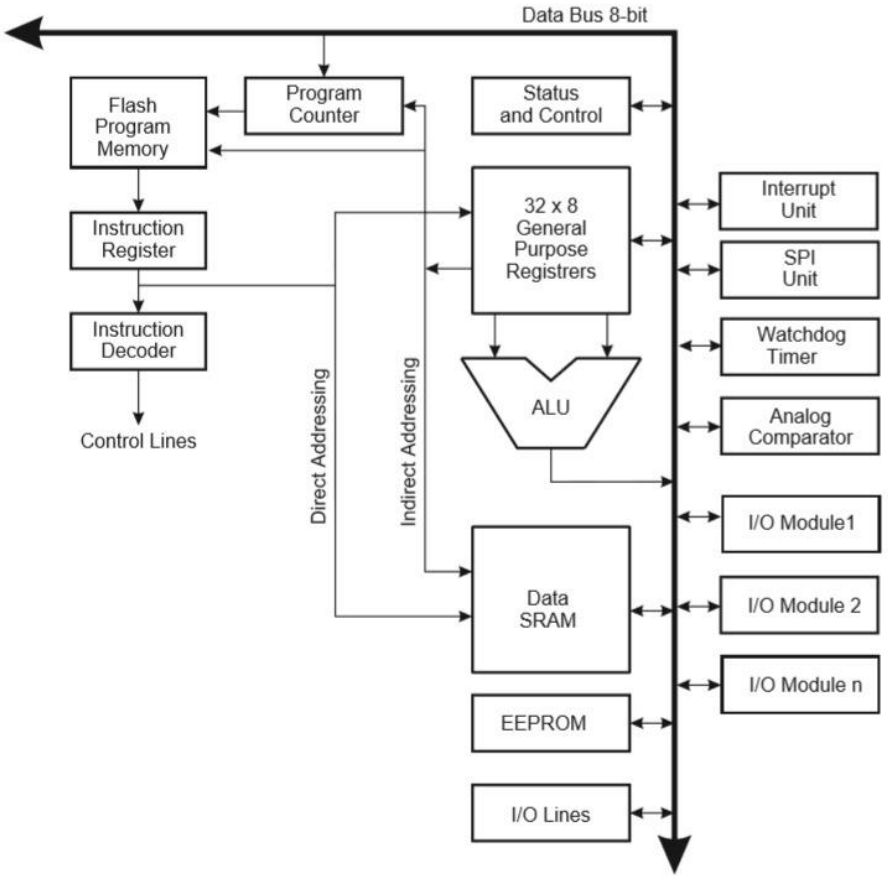
Архитектура современных микроконтроллеров

Микроконтроллер (МК) представляет собой урезанное ядро процессора общего научения с периферией общего и специального назначения, в т.ч. Цифровой, аналоговой и дискретной.

Также МК имеют. Спец периферию реального времени, такую как таймеры,

Модули обслуживания внешних прерываний и др.

Цель использования микроконтроллера – обработка информации измерительных устройств или устройств автоматизации в реальном времени.

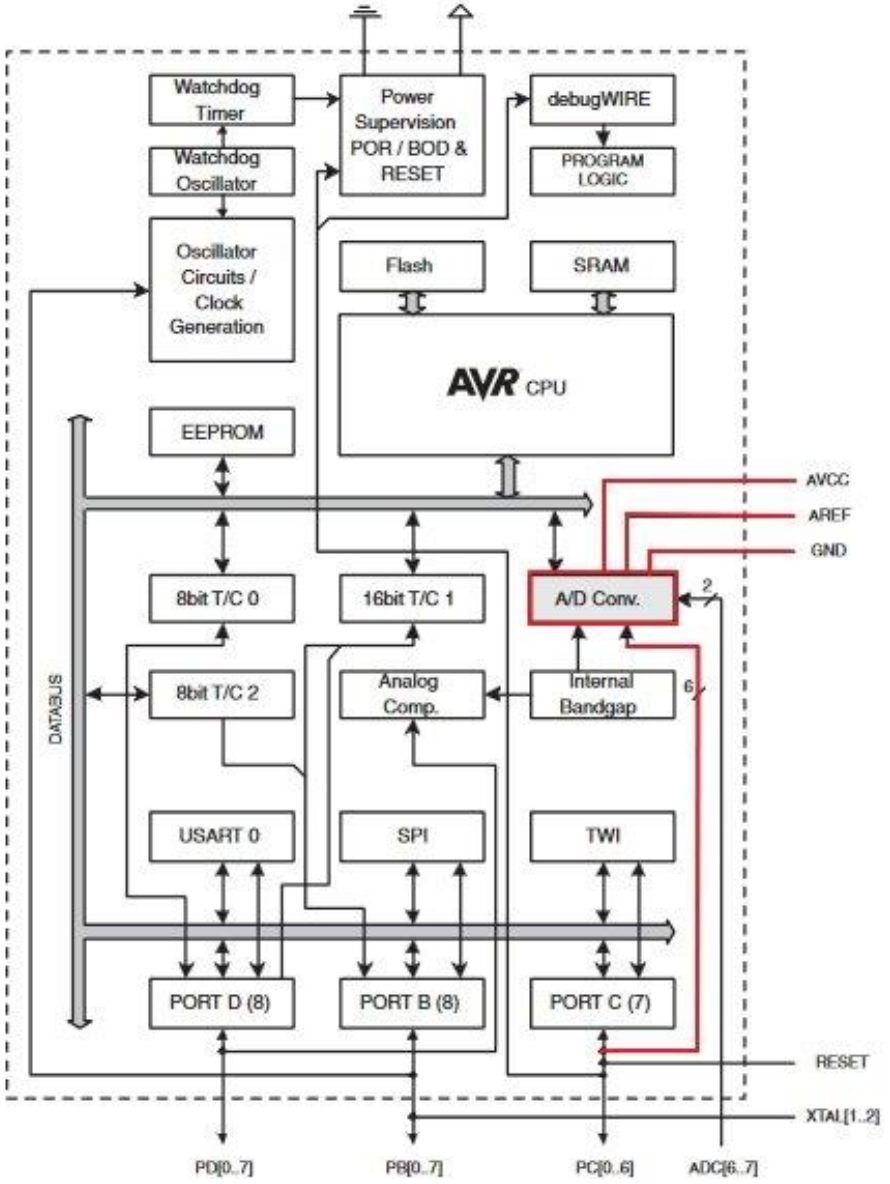


архитектура ядра AtMega
(в основе Arduino Uno)

Архитектура современных микроконтроллеров

МК как правило имеют урезанную Гарвардскую-RISC архитектуру ядра и весь необходимый обвес (SRAM, EEPROM ПЗУ, устройства ввода-вывода).
МК может содержать со-процессоры. Например для работы с USB, Ethernet, Bluetooth
или для поддержки, например, стандарта GPS.
А также сопроцессоры для операций цифровой обработки сигналов.

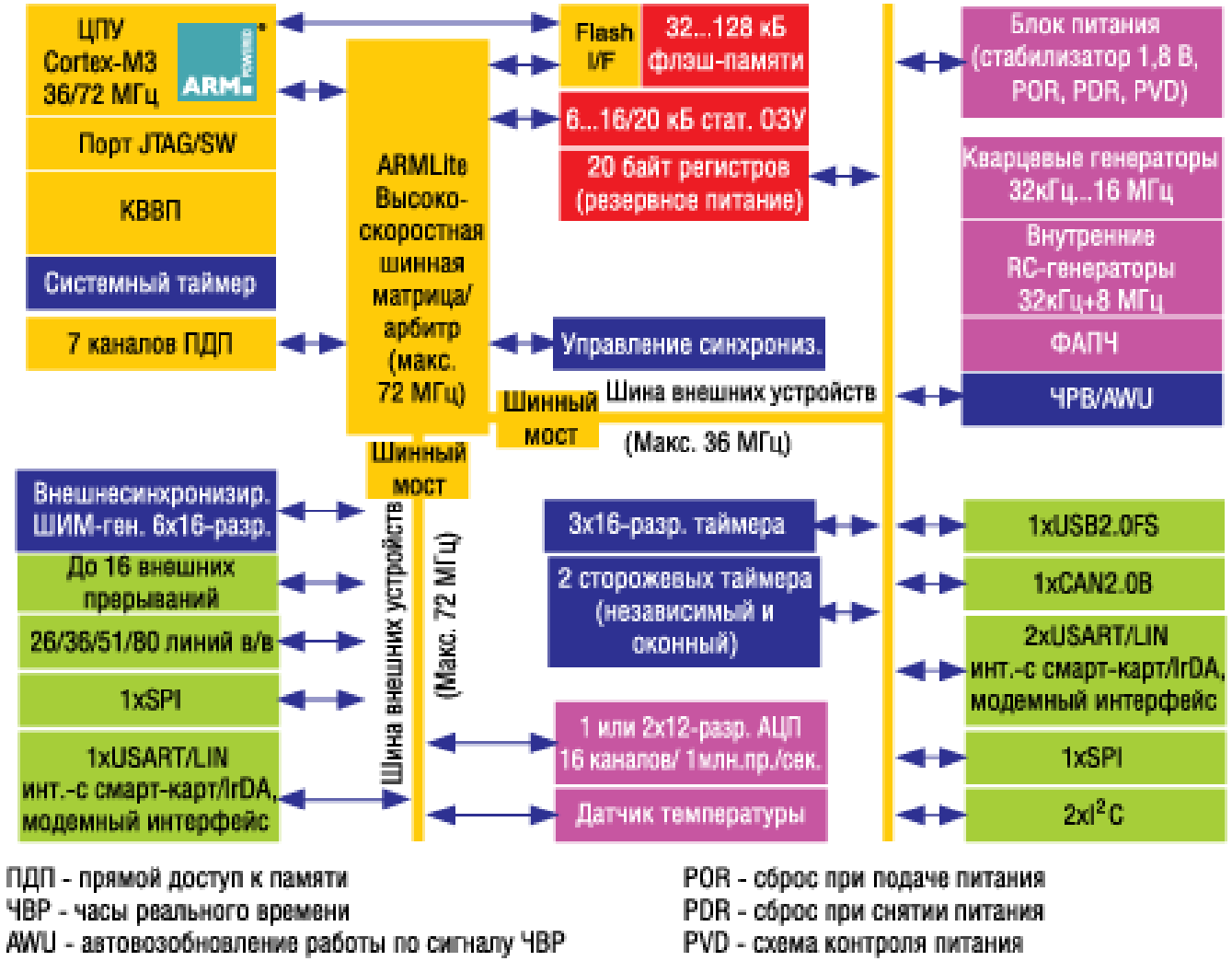
МК ориентированы на однозадачную работы без операционной системы и непосредственно с периферией (без дополнительных модулей типа чипсета и т.д.).



архитектура AtMega (в основе Arduino Uno)

Архитектура современных микроконтроллеров

- ARM Cortex M 32битная система, сопроцессоры аппаратного деления и умножения, работа с float
- RISC система команд
- конвейерная архитектура
- Развитая периферия
- Поддержка широкого набора промышленных интерфейсов

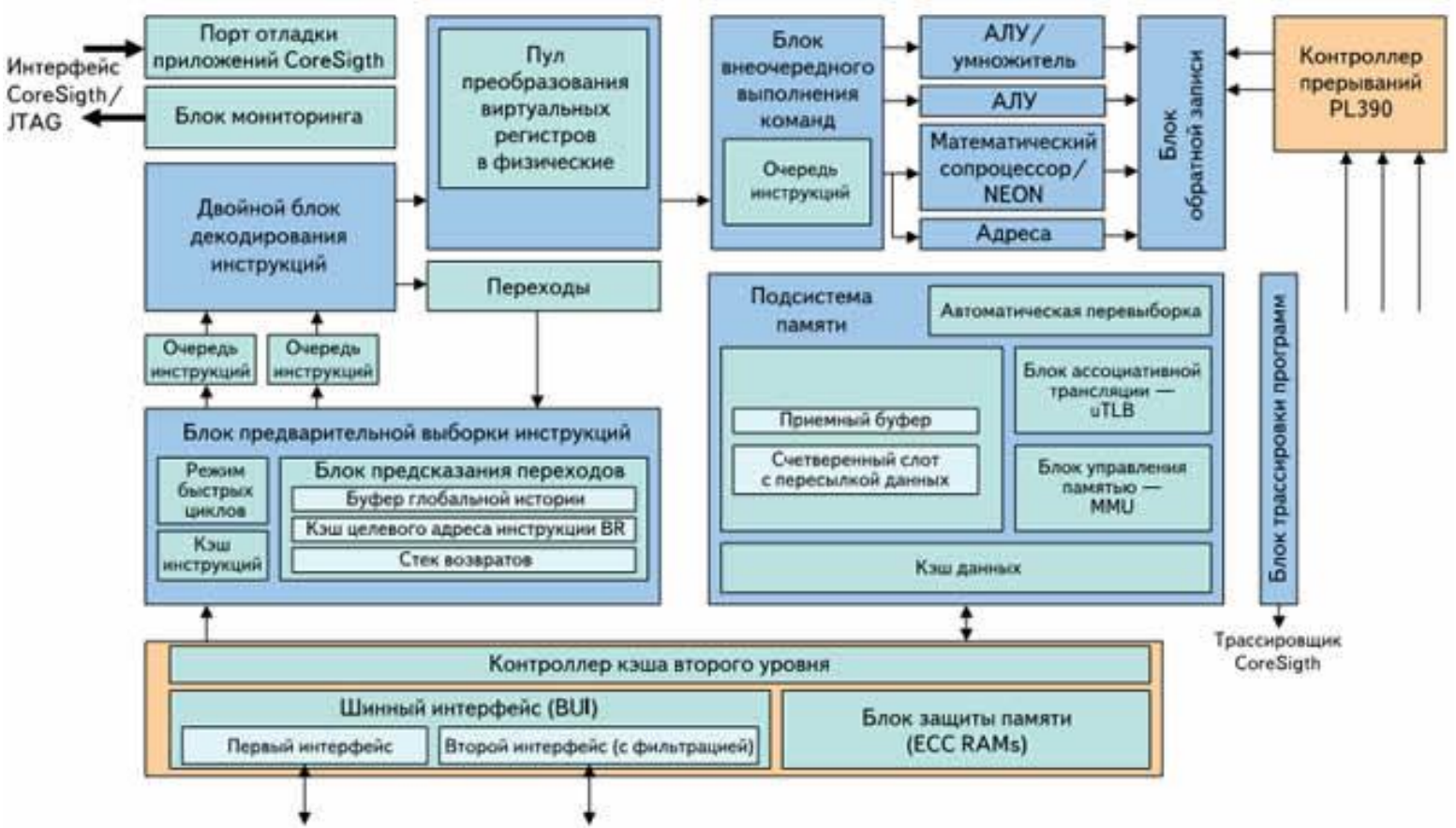


ПДП - прямой доступ к памяти
ЧВР - часы реального времени
AWU - автовозобновление работы по сигналу ЧВР

POR - сброс при подаче питания
PDR - сброс при снятии питания
PVD - схема контроля питания

- встроенный функционал работы с измерительной техникой, системами сбора данных и системами автоматизации.

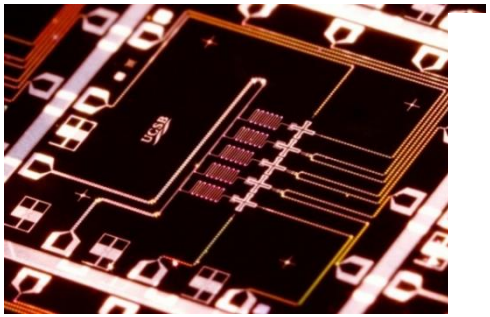
Архитектура современных микропроцессоров

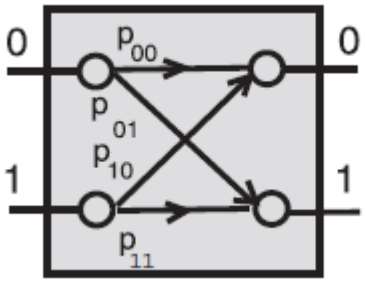


- ARM Cortex A9 процессор мобильных устройств на базе Cortex M (Гарвард, RISC).
- Расширенная конвейерная суперскалярная многоядерная архитектура, многопоточная ориентированная на работу с операционной системой.
- Многоуровневый спекулятивный КЭШ и др. особенности ЦПУ
- Ориентированы на низкое энерогопотребение.

Архитектура квантовых процессоров

Используются кубиты – состояния спинов элементарных частих, могут одновременно находится в двух состояниях, 3 кубита одновременно в 8 состояниях. Высокая параллельность вычислений. Экспоненциальный рост производительности.





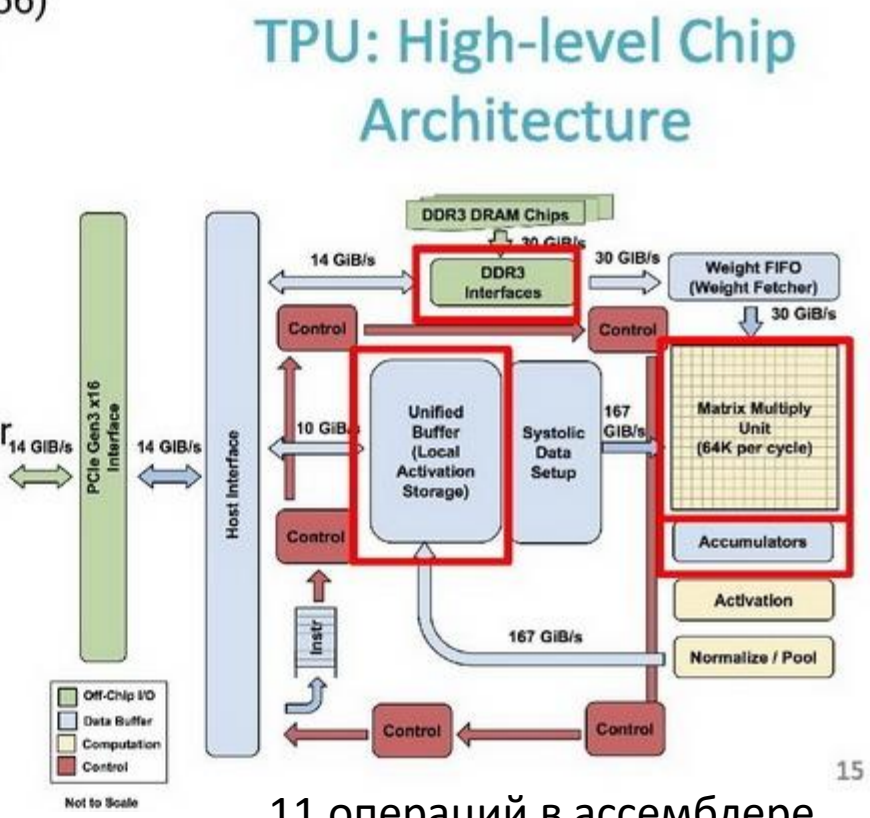
В последовательных приложениях квантовые компьютеры не имеют преимущества по сравнению с современными ЦПУ

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

Квантовые процессоры позволяют получить широкие возможности для параллелизма операций при их высокой скорости – что является одним из наиболее желаемых требований в задачах машинного обучения, криптографии и обработки массивов данных в научных и инженерных приложениях.

Архитектура тензорных процессоров

- The Matrix Unit: 65,536 (256x256) 8-bit multiply-accumulate units
- 700 MHz clock rate
- Peak: 92T operations/second
  - 65,536 * 2 * 700M
- >25X as many MACs vs GPU
- >100X as many MACs vs CPU
- 4 MiB of on-chip Accumulator memory
- 24 MiB of on-chip Unified Buffer (activation memory)
- 3.5X as much on-chip memory vs GPU
- Two 2133MHz DDR3 DRAM channels
- 8 GiB of off-chip weight DRAM memory

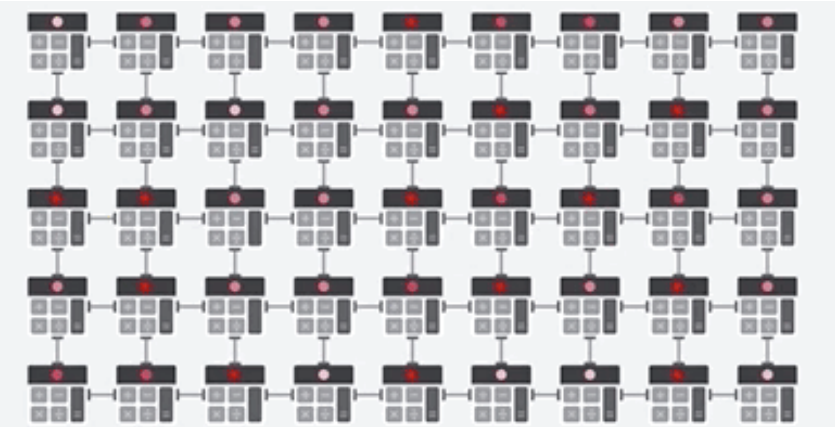


11 операций в ассемблере

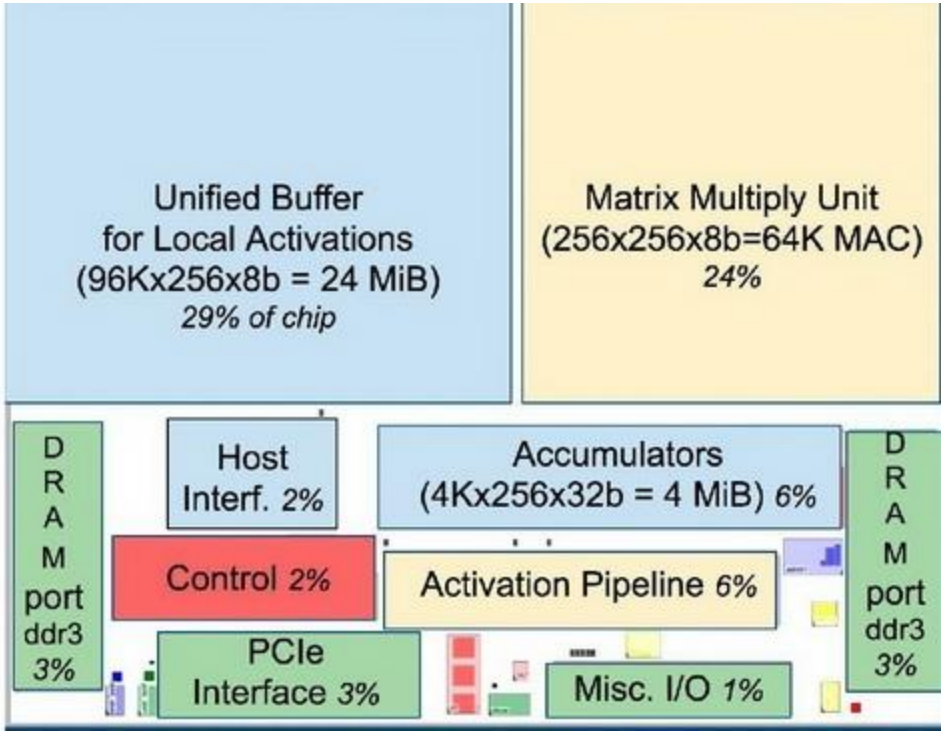
Архитектура Google TPU

Матрица умножения-сложения (matrix multiply-unit) 256 × 256, работает с 8-битными данными на частоте 700 МГц.
Пиковая производительность 92 триллионов операций в секунду.
Блоков MAC у TPU в 25 раз больше, нежели у современных GPU.

Архитектура тензорных процессоров



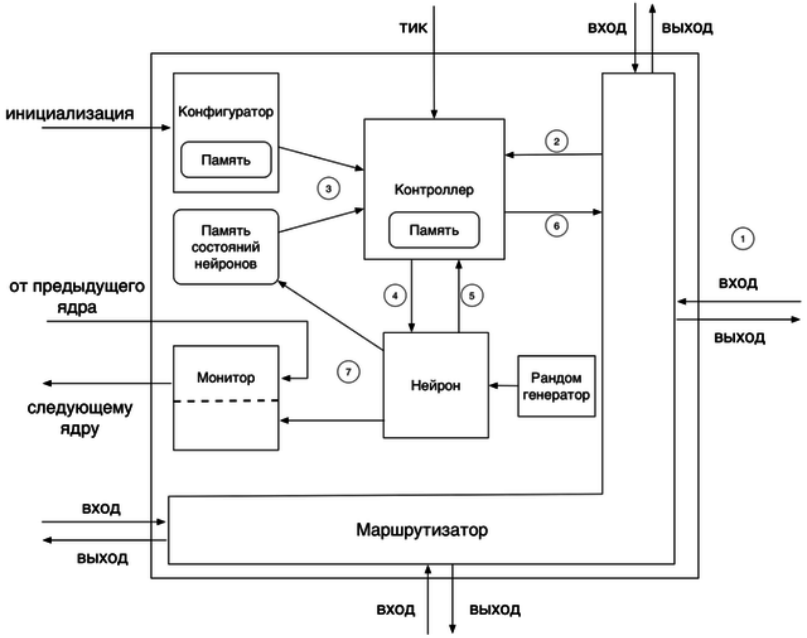
Архитектура типа систолический Массив (каждый элемент – TPU).



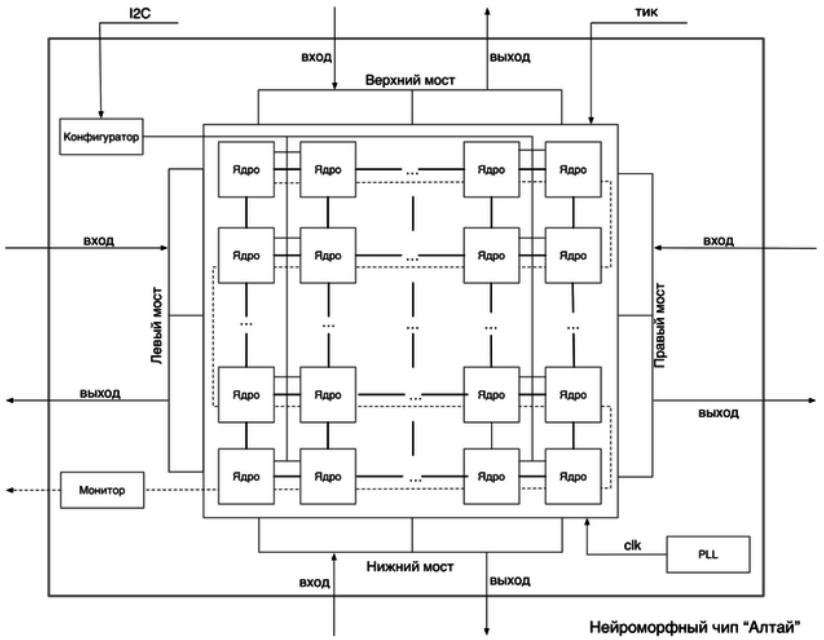
Архитектура Google TPU процессора

Тензорный процессор представлять собой т.н. систолический массив (Все АЛУ или тензорные АЛУ объединены в матрицу) – позволяет максимально ускорить обработку матриц данных, архитектура ориентирована на максимальное пере использование данных без обращения к внешнему ОУЗ за счет большого встроенного буфера.

Архитектура нейроморфных процессоров

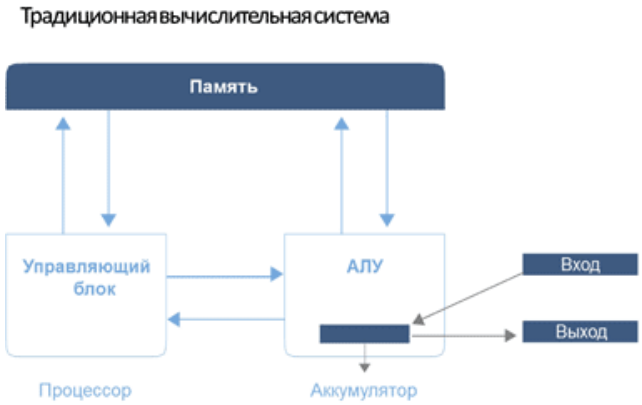


Ядро нейроморфного чипа “Алтай”

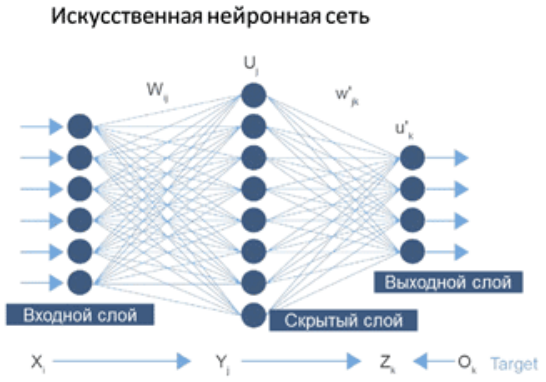


Нейроморфный чип “Алтай”

Процессор имеет параллельную коммуникационную архитектуру, отправляет множество небольших объёмов информации одновременно в различных направлениях



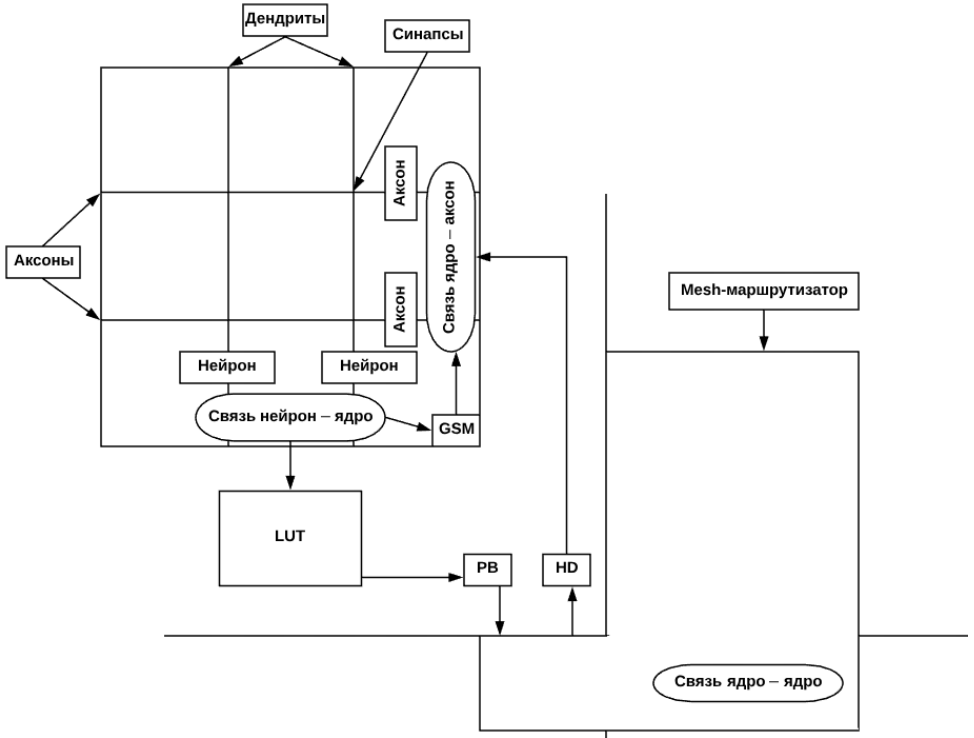
Оптимальная архитектура для последовательных вычислений



Распределенная, параллельная, однонаправленная

Архитектура нейроморфных процессоров

Традиционные процессоры состоят из обособленных блоков, выполняющих разные функции (вычислительные и периферийные блоки, память).
Нейроморфныепроцессоры имеют «однородную» структуру, включающую множества нейронов – одинаковых и относительно простых вычислительных ячеек со встроенной памятью



Чип IBM TrueNorth (архитектура NeuroMem)

Каждое ядро – нейрон, ядра объединены в слои, образующие нейронную сеть (предполагается, что это сверхточная НС), однако архитектура поддерживает и рекуррентный принцип работы НС

Традиционный и нейроморфный процессоры