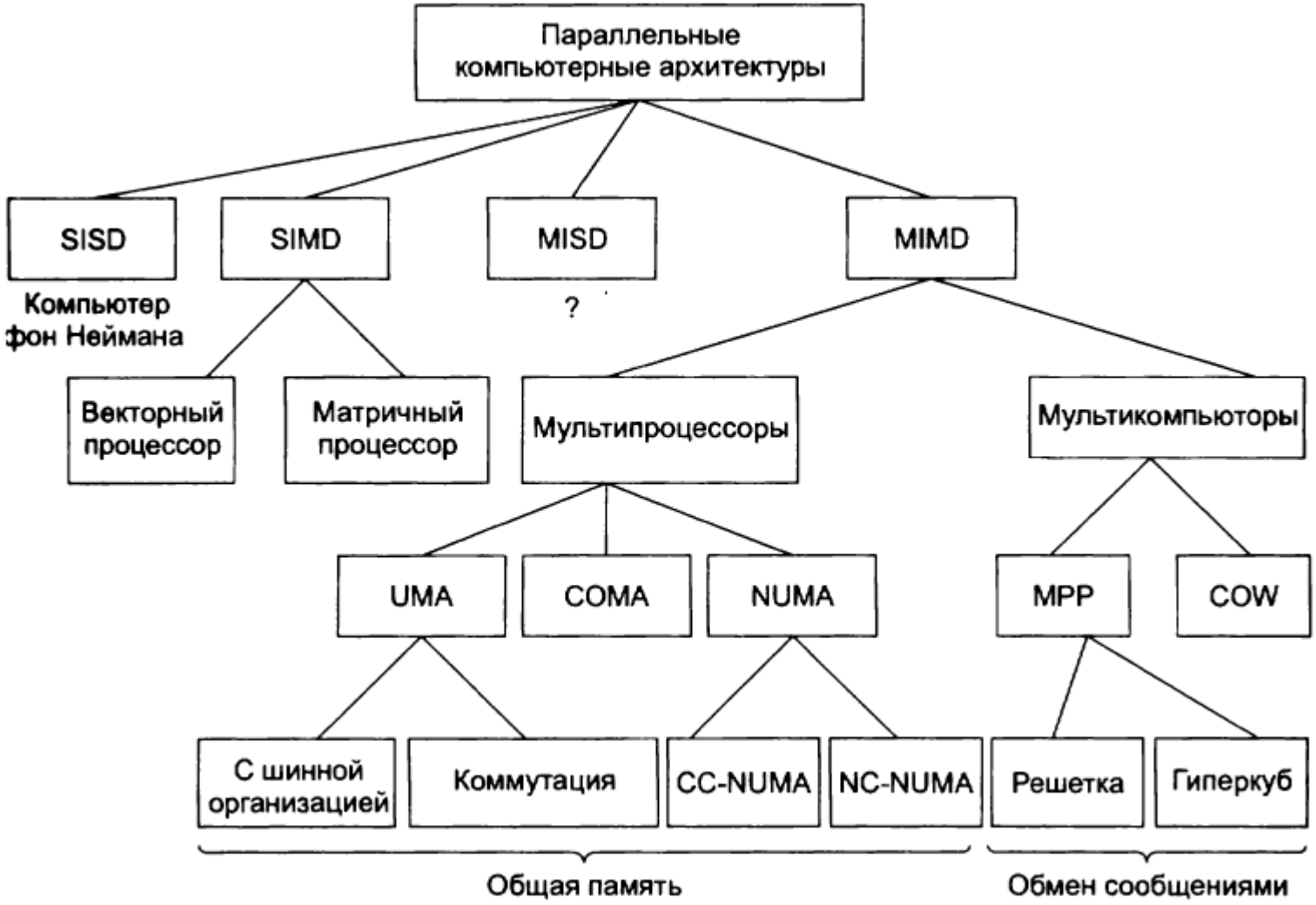
Классификация компьютерных архитектур по степени параллелизма

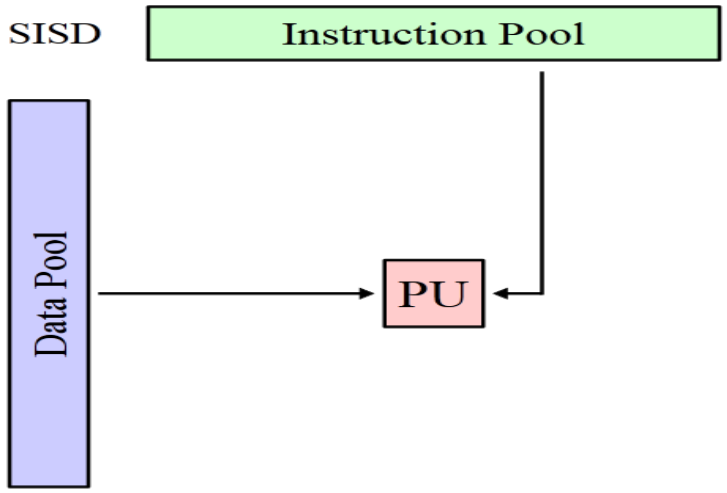
Классификация по способам обработки команд и данных (по способам наличия параллелизма)

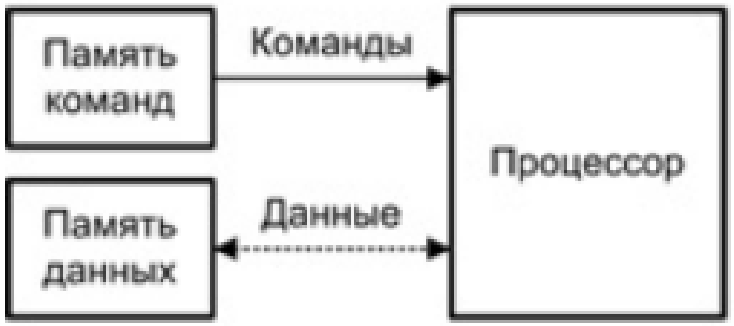
- В 1969 профессором Стэнфордского университета Майклом Флинном была предложена общая классификация архитектур ЭВМ по способам обработки команд и данных.

Параллельные архитектуры



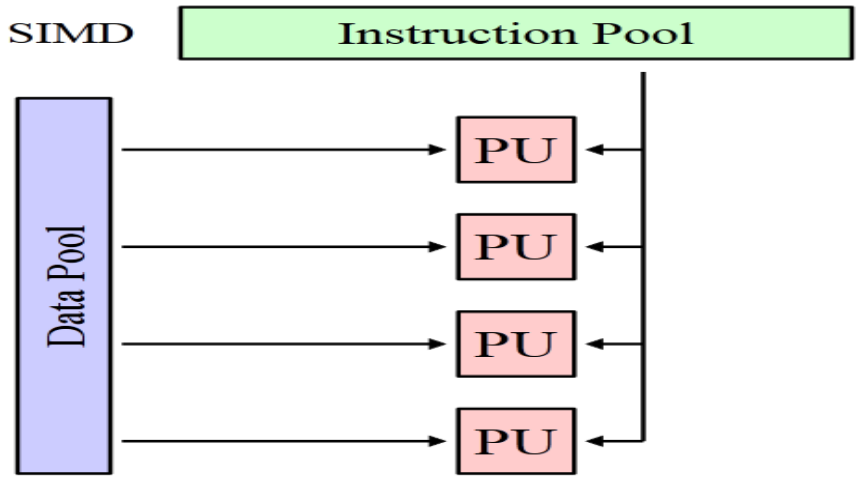
Архитектура SISD (*Single Instruction, Single Data)*

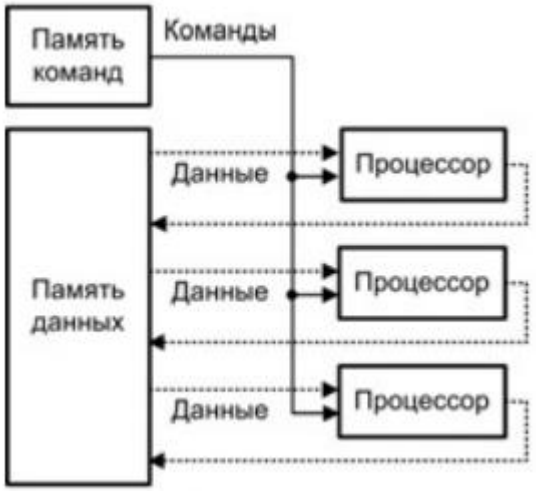




- Одиночный поток команд, одиночный поток данных
  - **Скалярные**
  - **Суперскалярные/конвейерные**
  - **Процессоры с длинными командами (VLIW)**

Архитектура SIMD (Single instruction, multiple data)

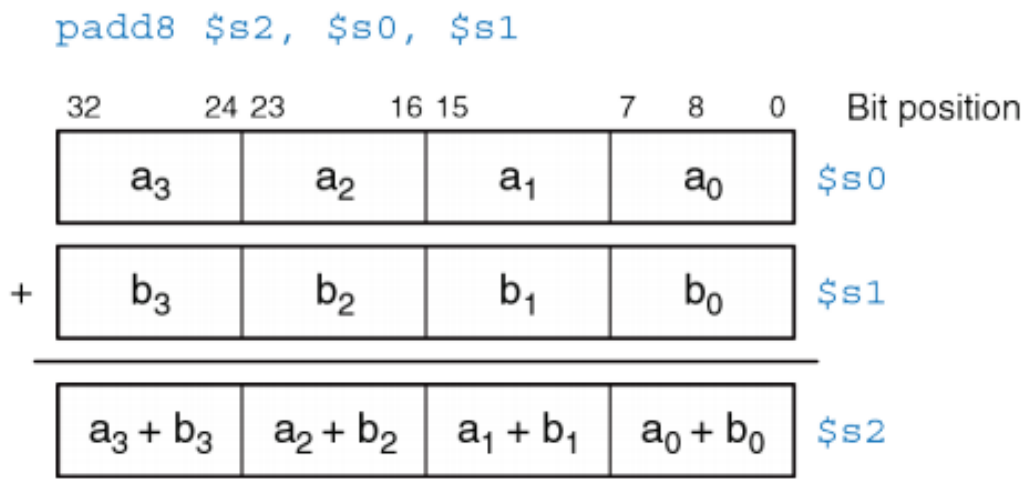




- Одиночный поток команд и множественным поток данных (векторные и матричные процессоры).
- Выполняют операции «пакованной арифметики»

Паковванная арифметика

- Выполнение арифметической операции над несколькими небольшими **независимыми** блоками данных
- Переносы между блоками отсутствуют



Паковванная арифметика: четыре одновременных операции

Векторный процессор

- пример по парного сложения двух наборов по 10 чисел

СКАЛЯРНЫЙ ПРОЦЕССОР

Повторить цикл 10 раз

1. прочитать следующую инструкцию и декодировать
2. получить первое слагаемое
3. получить второе слагаемое
4. сложить
5. сохранить результат

конец цикла

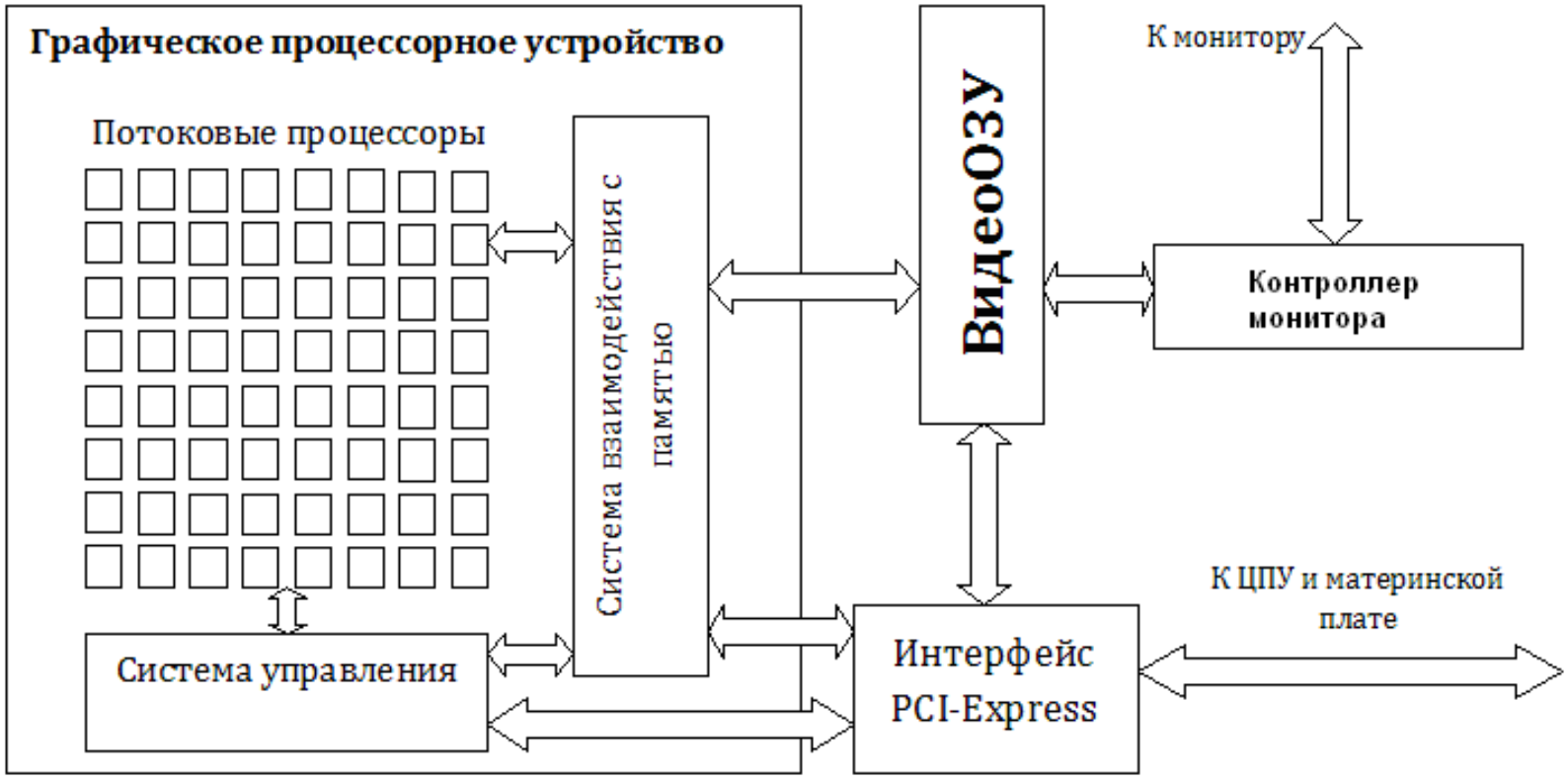
ВЕКТОРНЫЙ ПРОЦЕССОР

1. прочитать следующую инструкцию и декодировать
2. получить 10 первых слагаемых
3. получить 10 вторых слагаемых
4. сложить
5. сохранить результат

SIMD процессоры

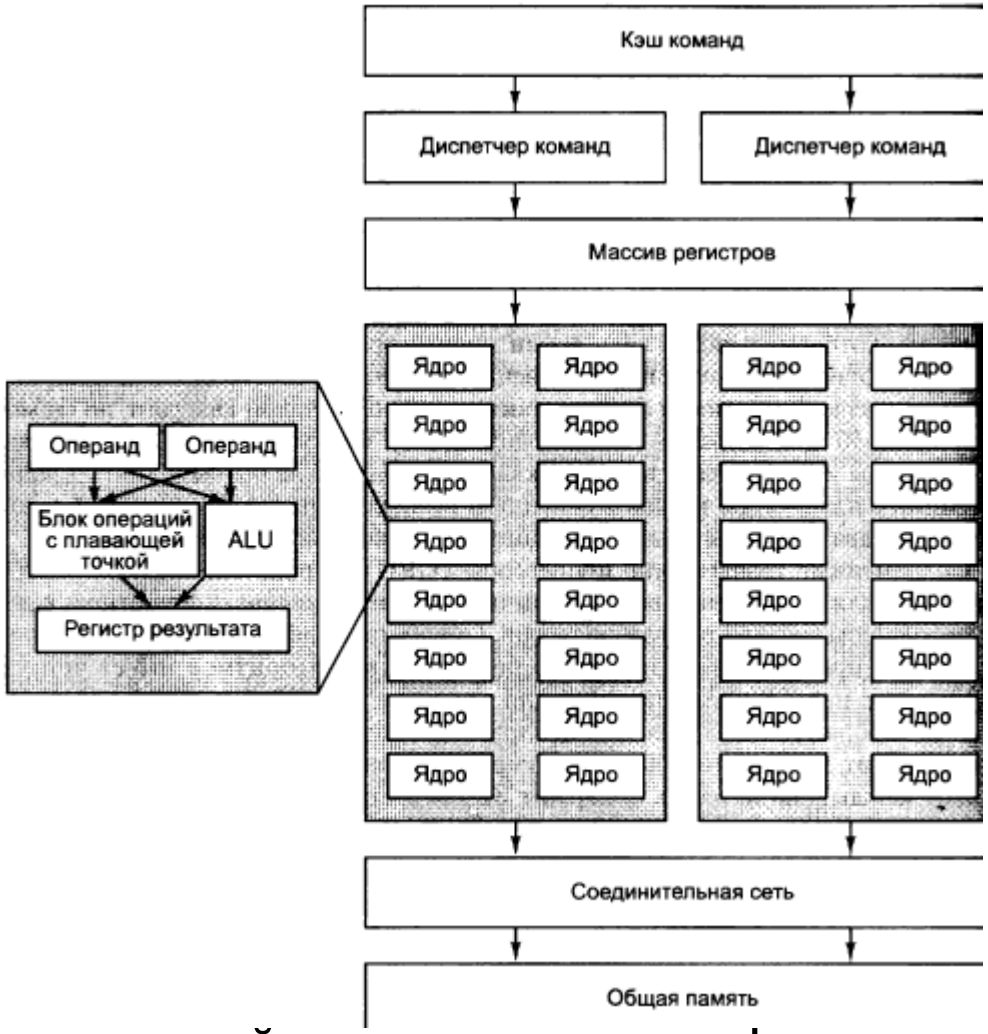
- Матричные (потоковые) реализуется в виде отдельного устройства
  - для выполнения попарных операций над каждыми элементами вектора используется отдельный специализированный упрощенный процессор .
- Векторные в виде набора дополнительных команд (MMX, iwMMXt, SSE, SSE2, SSE3, SSSE3, SSE4.x, AVX, AVX2
- AMD: 3DNow!
  - в векторных- элементы массивов загружаются в векторные (длинные) регистры над которыми в одном «стандартном» конвейерном процессоре выполняется операция

Технология GPGPU



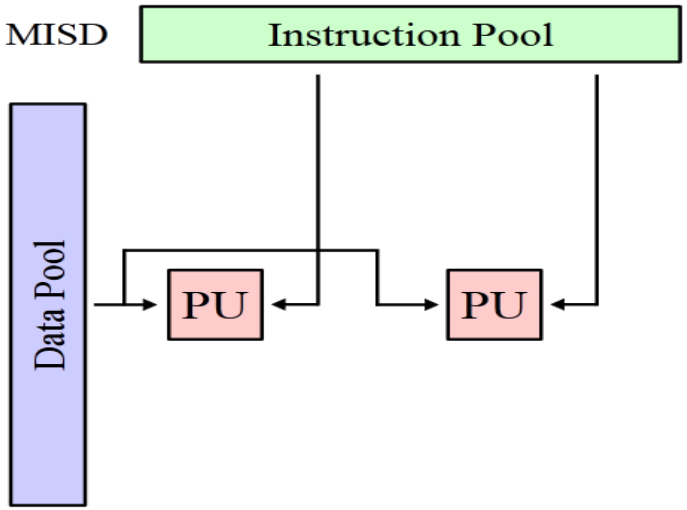
Технология GPGPU (general-purpose graphics processing unit – универсальный графический процессор) позволяет использовать графические процессоры для реализации различных вычислительных алгоритмов.

Матричный графический процессор NVIDIA



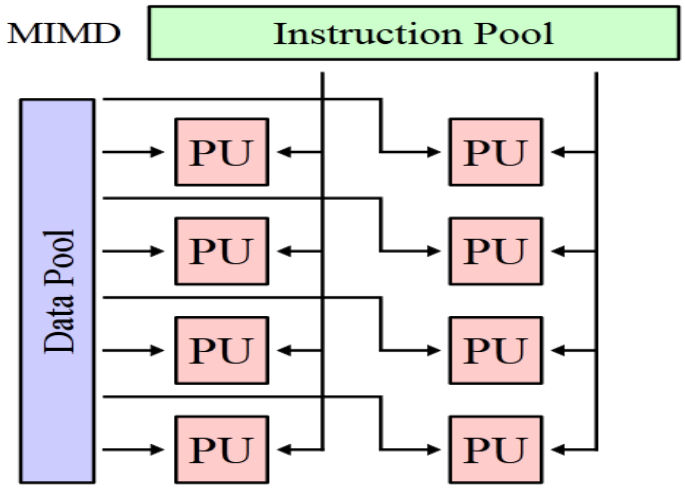
- Ядро выполняет простейшие операции с фиксированной и плавающей точкой (CUDA –ядро *Compute Unified Device Architectur*)
- Каждое ядро выполняет одинаковую команду для своего набора данных (над двумя элементами массива)

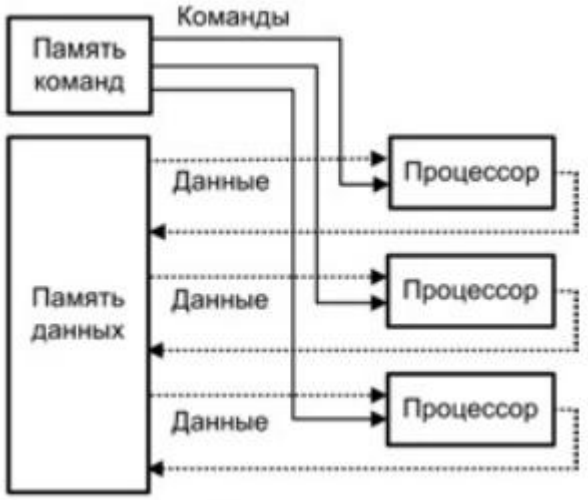
Архитектура MISD (Multiple Instruction stream, Single Data stream)



- **Множественный поток команд и одинарный поток данных** .
- Несколько процессоров выполняют различные операции над одними данными
- Практически не используется

Архитектура MIMD (Multiple Instruction stream, Multiple Data stream)





- Множественный поток команд и данных.
- Каждый процессор параллельно обрабатывает свой поток команд на своем потоке данных.
- Многопроцессорные, многоядерные, многопоточные процессоры, а также компьютерные кластеры.

Четыре стены

- Стена частоты (4-5 Ггц)
- Стена мощности
- Стена памяти
- Стена количества одновременно выполняемых команд

- Выход – создание многоядерных процессоров

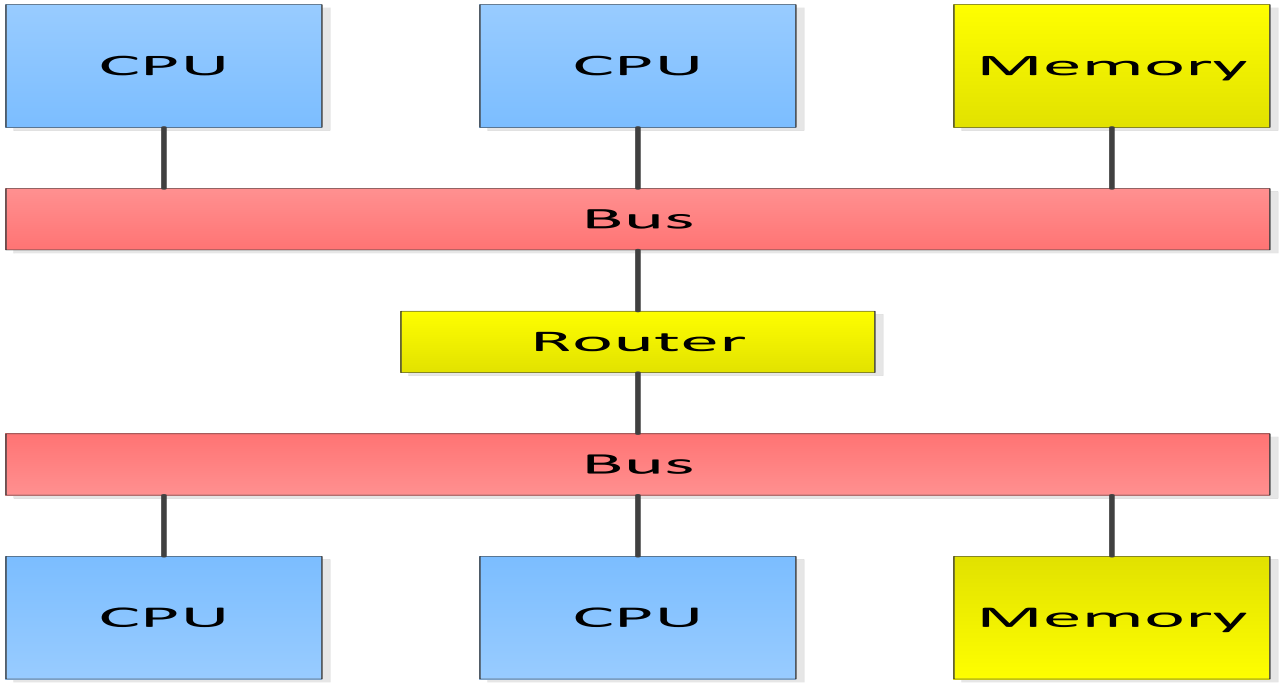
Процессоры/Ядра и память

Многопроцессорные архитектуры с однородным доступом к памяти (Uniform Memory Access), SMP-системы (Symmetric Multi-Processor Architecture)

Процессоры имеют равноправный доступ к общей оперативной памяти через общую шину или коммутатор

«+» Хорошо подходит для многопоточных приложений.
«-» Сложная аппаратная реализация (наличие общих линий обращения к памяти).
«-» Снижение производительности с ростом количества процессоров/ядер (до 32процессоров)

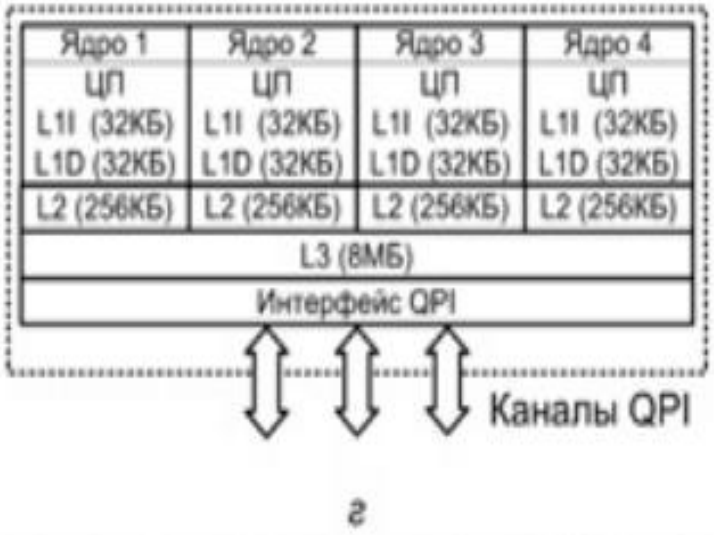
Гибридная архитектура с неоднородным доступом к памяти – NUMA (Non-Uniform Memory Access Architecture)



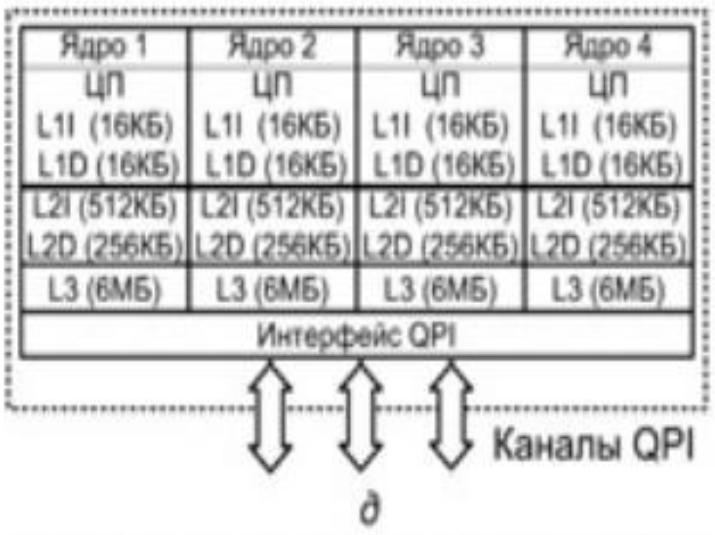
Каждый процессор может обращаться, как к своей локальной памяти, так и к памяти соседа. Память становится трехуровневой:

- кэш-память процессора;
- локальная оперативная память;
- удаленная оперативная память.
- противоречит принципам Фон-Неймана (сложнее программировать)

Многоядерные процессоры



г



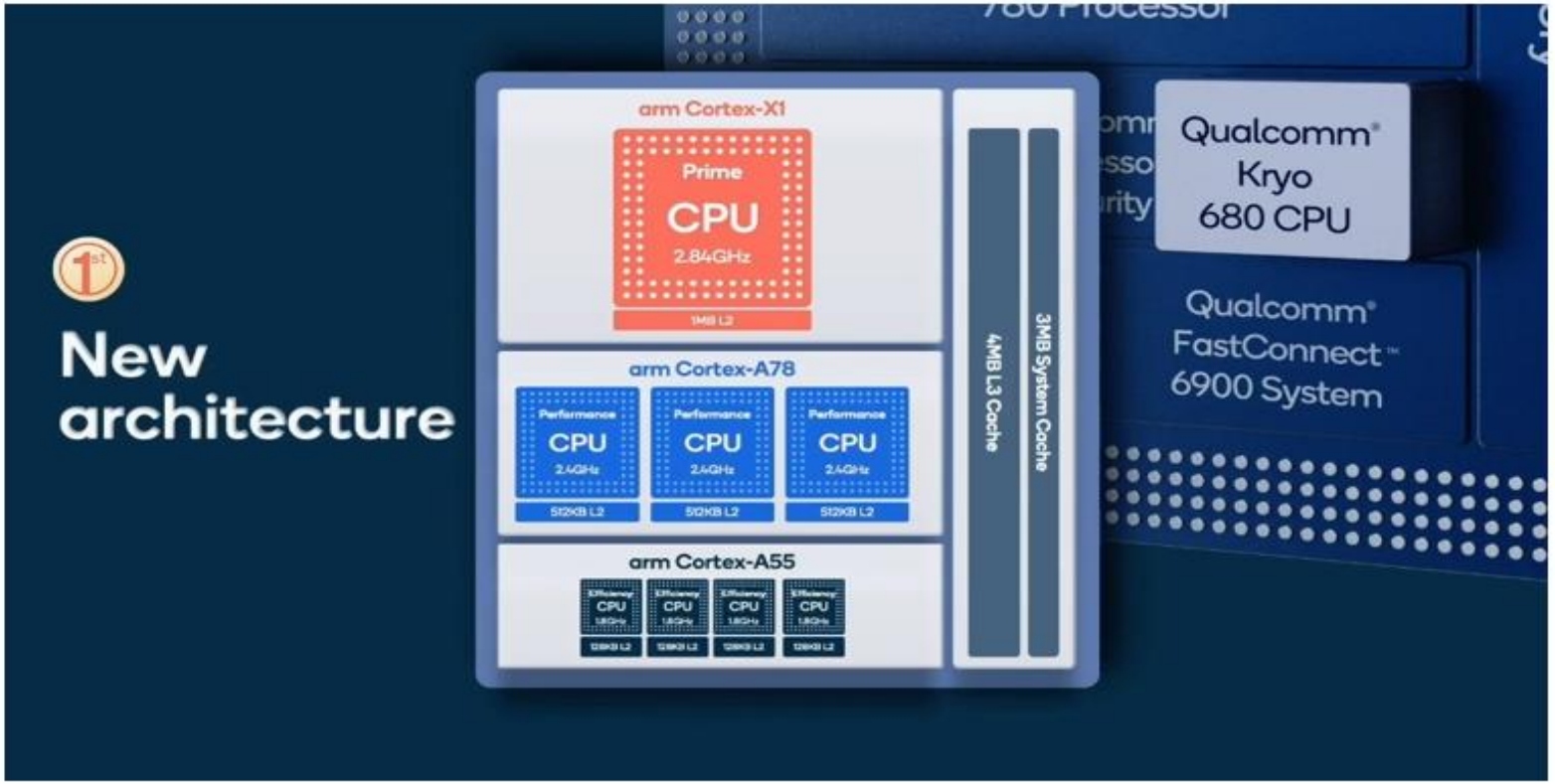
д

Проблемы многоядерности

- не все программы поддерживают распределение вычислений на несколько ядер.
- усложняется работа с памятью, так как ядер – много, и всем им требуется доступ к ОЗУ. Требуется сложный механизм, определяющий очередность доступа ядер процессора к памяти и к другим ресурсам ЭВМ (SMP и NUMA архитектуры).
- возрастает энергопотребление, а, следовательно, увеличивается тепловыделение и требуется более мощная система охлаждения и питания .
- себестоимость производства многоядерных процессоров – выше.

big.LITTLE

- Мобильный флагманский чип Snapdragon 888



- Сверхпроизводительное» ядро ARM Cortex-X1 – 1 шт
- Производительные («большие») ядра ARM Cortex-A78 – 3 шт
- Энергоэфективне («малые») ядра ARM Cortex-A55 – 4 шт