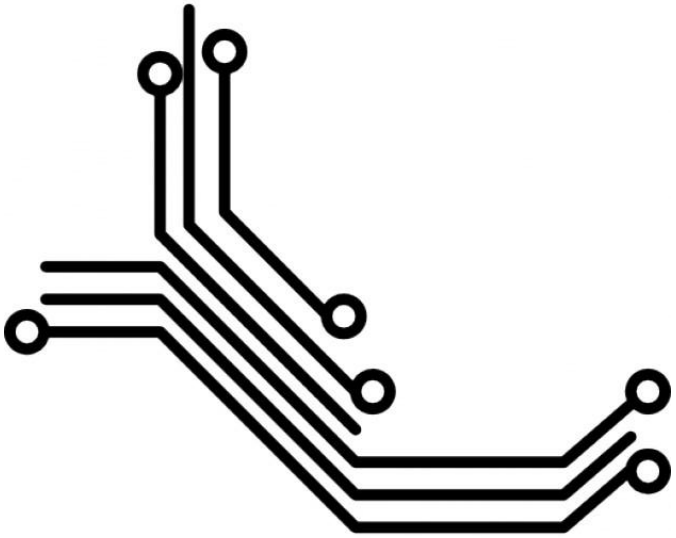


Высокопроизводительные вычисления

Минков В.И. 2016

Введение, закон Мура

- Рост производительности обеспечивался с помощью уменьшения размеров элементов микропроцессоров. При этом падало энергопотребление и росли частоты работы, компьютеры становились все быстрее, сохраняя, в общих чертах, свою архитектуру. Менялся техпроцесс производства микросхем и мегагерцы вырастали в гигагерцы.
- Оказалось, частоту дальше повышать нельзя — растут токи утечки, процессоры перегреваются и обойти это не получается.
- Закон Мура, по которому число транзисторов и связанная с ним производительность компьютеров удваивалась каждые полтора-два года оказался сомнительным.

$$Производительность = \frac{количество\ инструкций}{время\ выполнения}$$

$$Производительность = \left(\frac{количество\ инструкций}{количество\ тактов}\right) \times \left(\frac{количество\ тактов}{время\ выполнения}\right)$$

- Первая часть произведения — количество инструкций, выполняемых за такт (IPC, Instruction Per Clock), вторая – кол-во тактов процессора в единицу времени, тактовая частота.
  Для увеличения производительности нужно поднимать тактовую частоту/увеличивать кол-во инструкций за один такт.
  Рост частоты остановился -> увеличение количества исполняемых инструкций.

Параллельность

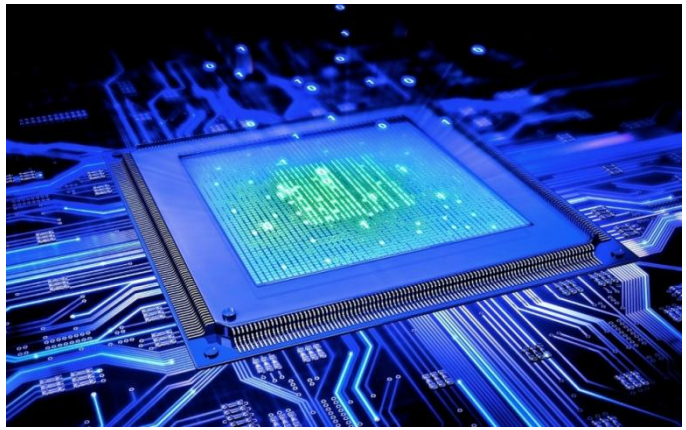
- Процессор, который умеет сам определять независимые и непротиворечащие друг другу инструкции и параллельно их выполнять, называется суперскалярным
  - *Пример:*
    - *A=1*
    - *B=2*
    - *C=A+B*
- EPIC (explicitly parallel instruction computing) — микропроцессорная архитектура с явным параллелизмом команд. …..?
- Hyper Threading …..?
- Технологии параллелизма на уровне инструкций активно развивались в 90е и первую половину 2000х годов, но в настоящее время их потенциал практически исчерпан

Параллельность

Под **параллельными вычислениями** понимают обработку данных, при которой одновременно выполняется несколько машинных операций.

Дополнительной формой вычислений является конвейерная реализация обрабатывающих устройств.

При рассмотрении проблемы организации параллельных вычислений следует различать следующие возможные режимы выполнения независимых частей программы:



- *многозадачный режим (режим разделения времени)* - для выполнения процессов используется единственный процессор
- *параллельное выполнение* - в один и тот же момент времени может выполняться несколько команд обработки данных

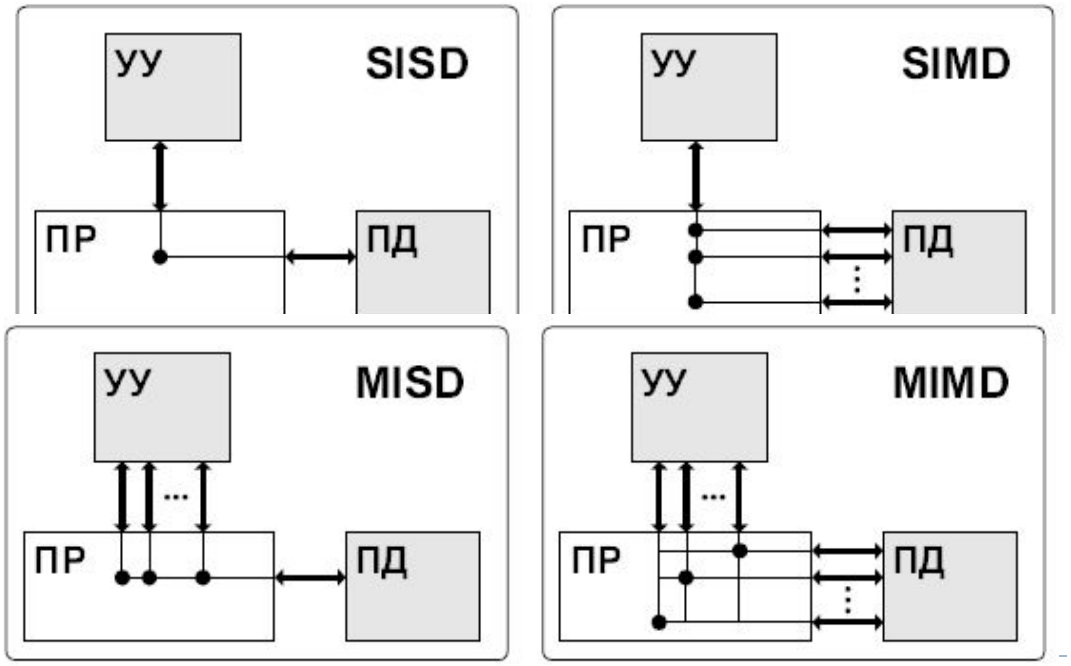
- *распределенные вычисления* - параллельная обработка данных, при которой используется несколько обрабатывающих устройств

Параллелизм на уровне данных

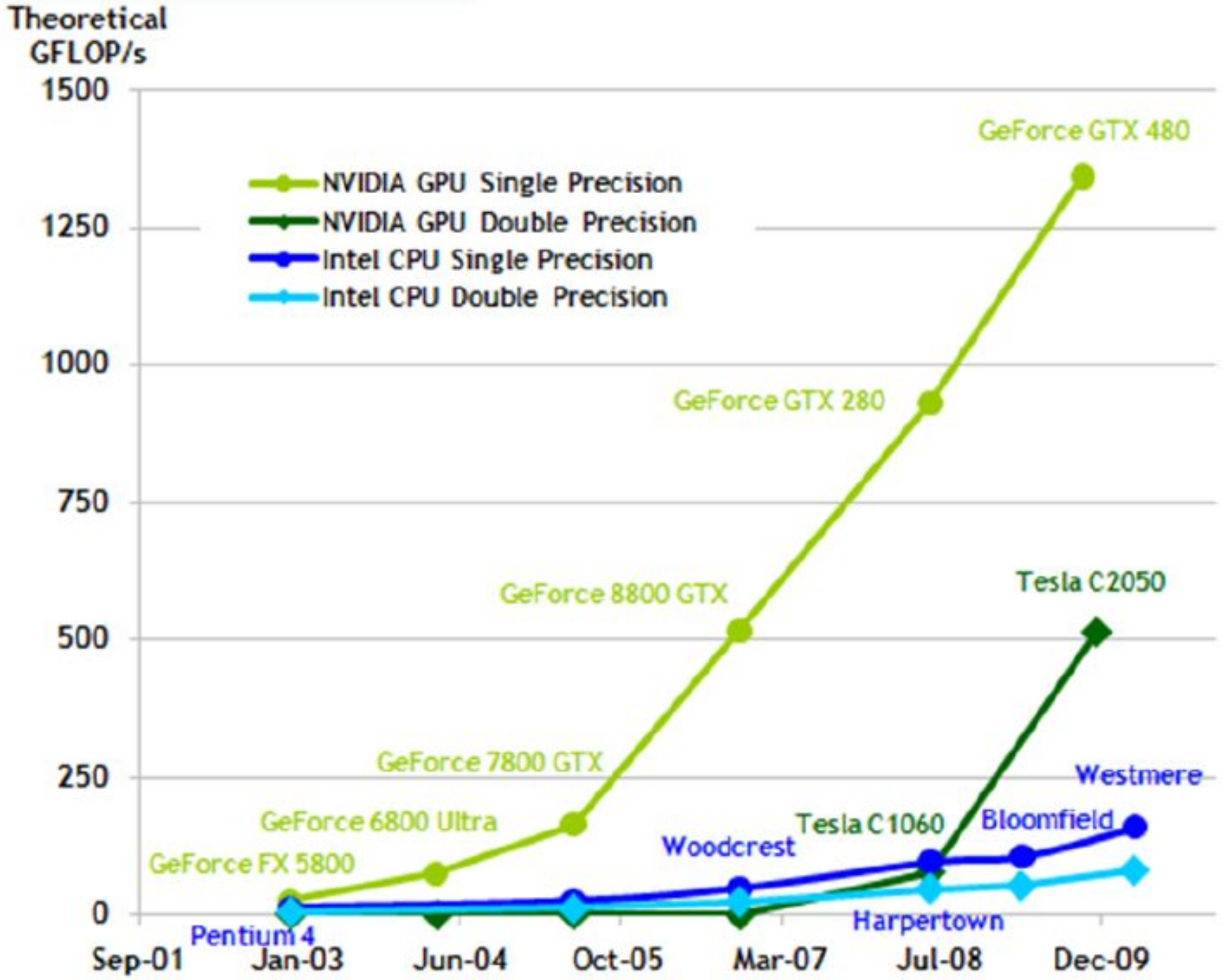
- Векторные процессоры
- относятся к **SIMD — (single instruction, multiple data — одиночный поток команд, множественный поток данных)**
- **Графические процессоры**
- **SIMT — (single instruction, multiple threads, одна инструкция — множество потоков)**. Так же как в SIMD операции производятся с массивами данных, но степеней свободы гораздо больше — для каждой ячейки обрабатываемых данных работает отдельная нить команд.

Классификация вычислительных систем

- **SISD** (Single Instruction, Single Data) - системы, в которых существует одиночный поток команд и одиночный поток данных;
- **SIMD** (Single Instruction, Multiple Data) - системы с одиночным потоком команд и множественным потоком данных;
- **MISD** (Multiple Instruction, Single Data) - системы, в которых существует множественный поток команд и одиночный поток данных;
- **MIMD** (Multiple Instruction, Multiple Data) - системы с множественным потоком команд и множественным потоком данных;



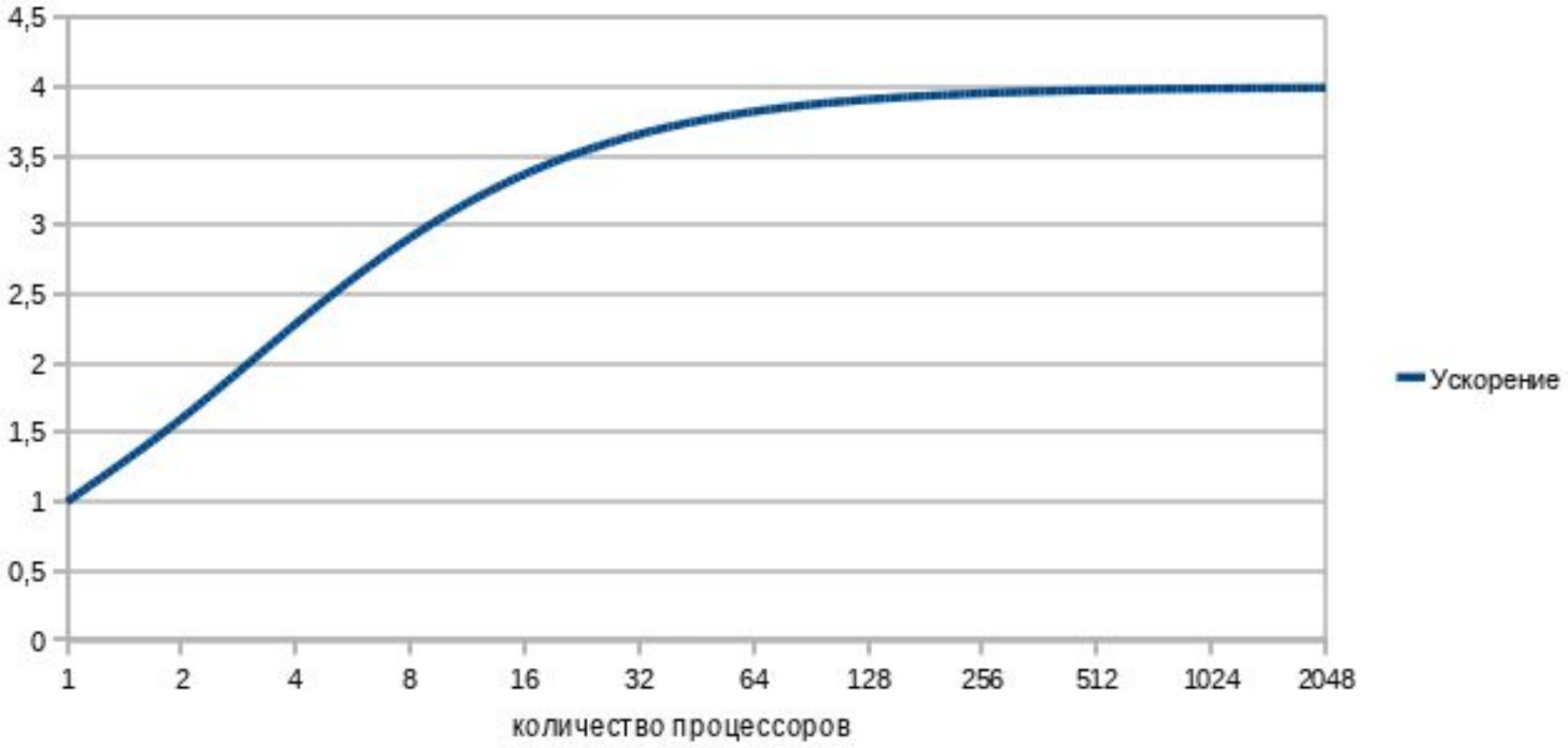
ПР — это один или несколько процессорных элементов,
УУ — устройство управления,
ПД — память данных


Theoretical GFLOP/s
1500
1250
1000
750
500
250
0
NVIDIA GPU Single Precision
NVIDIA GPU Double Precision
Intel CPU Single Precision
Intel CPU Double Precision
GeForce GTX 480
GeForce GTX 280
GeForce 8800 GTX
GeForce 7800 GTX
GeForce 6800 Ultra
GeForce FX 5800
Tesla C2050
Tesla C1060
Westmere
Bloomfield
Woodcrest
Harpertown
Pentium 4
Sep-01
Jan-03
Jun-04
Oct-05
Mar-07
Jul-08
Dec-09

Мультиархитектуры

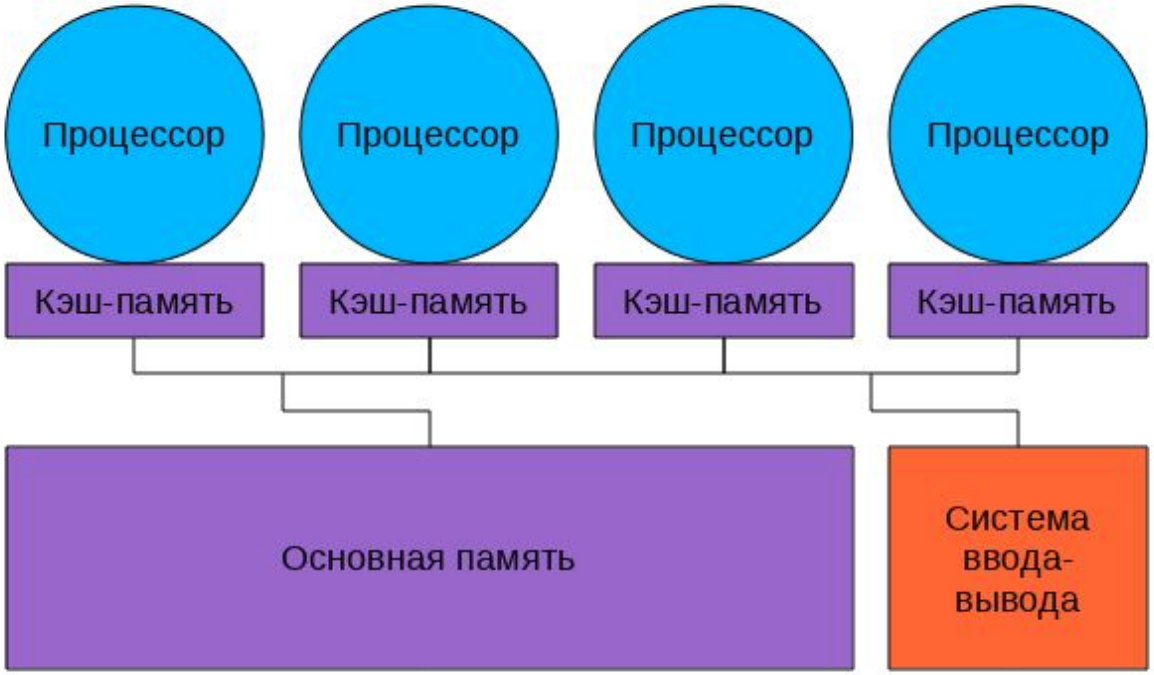
- **MIMD (Multiple Instruction stream, Multiple Data stream — Множественный поток Команд, Множественный поток Данных)**
- многопоточные программы
- Ускорение кода зависит от числа процессоров и параллельности кода согласно формуле

$$Ускорение = \frac{1}{Время\ выполнения\ последовательного\ кода + \frac{время\ выполнения\ параллельного\ кода}{количество\ процессоров}}$$


4,5
4
3,5
3
2,5
2
1,5
1
0,5
0
1
2
4
8
16
32
64
128
256
512
1024
2048
количество процессоров
Ускорение

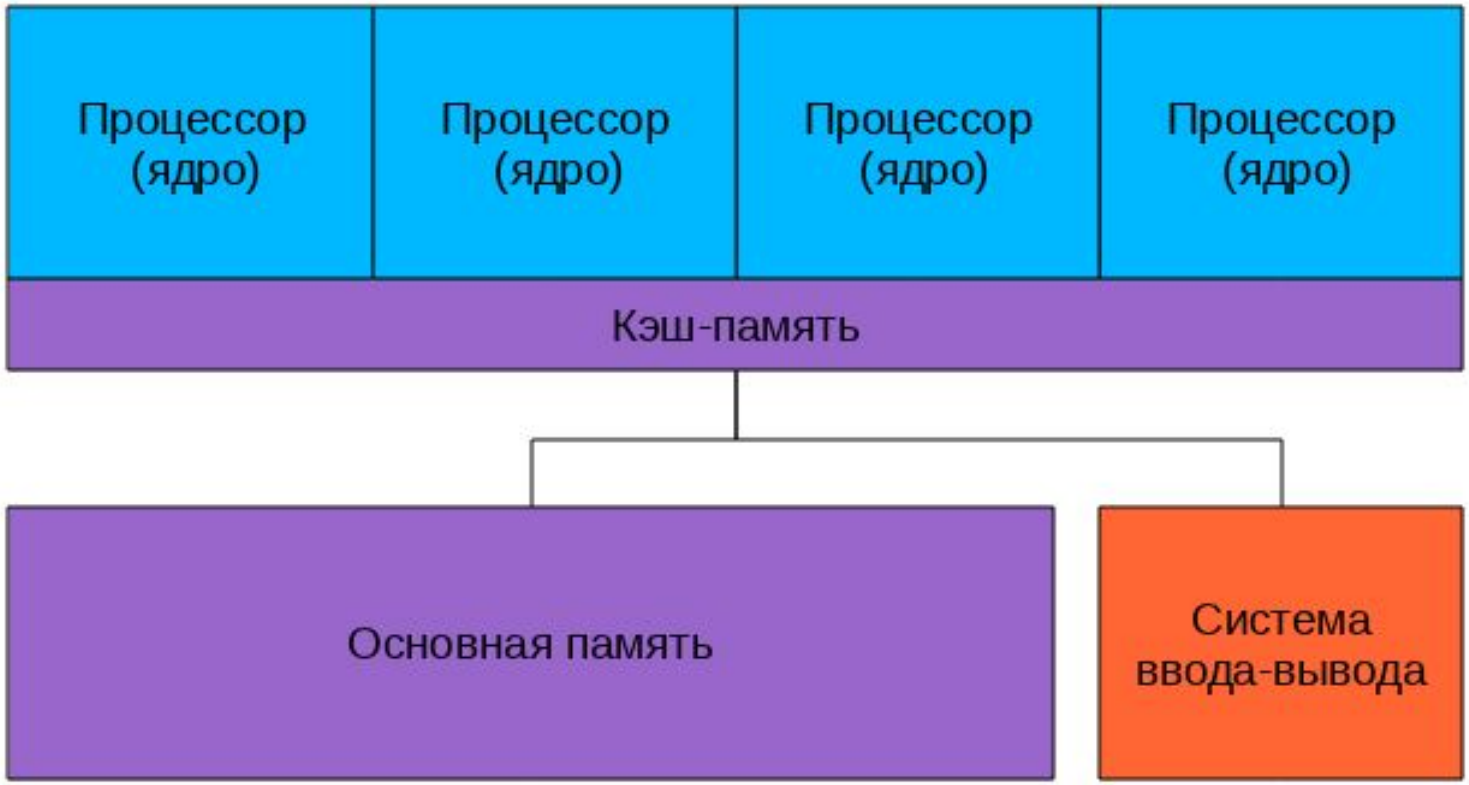
Мультипроцессор

- **Мультипроцессор** — это компьютерная система, которая содержит неск. процессоров и одно видимое для всех процессоров адресное пространство.
  Мультипроцессоры отличаются по организации работы с памятью.
- Системы с общей памятью



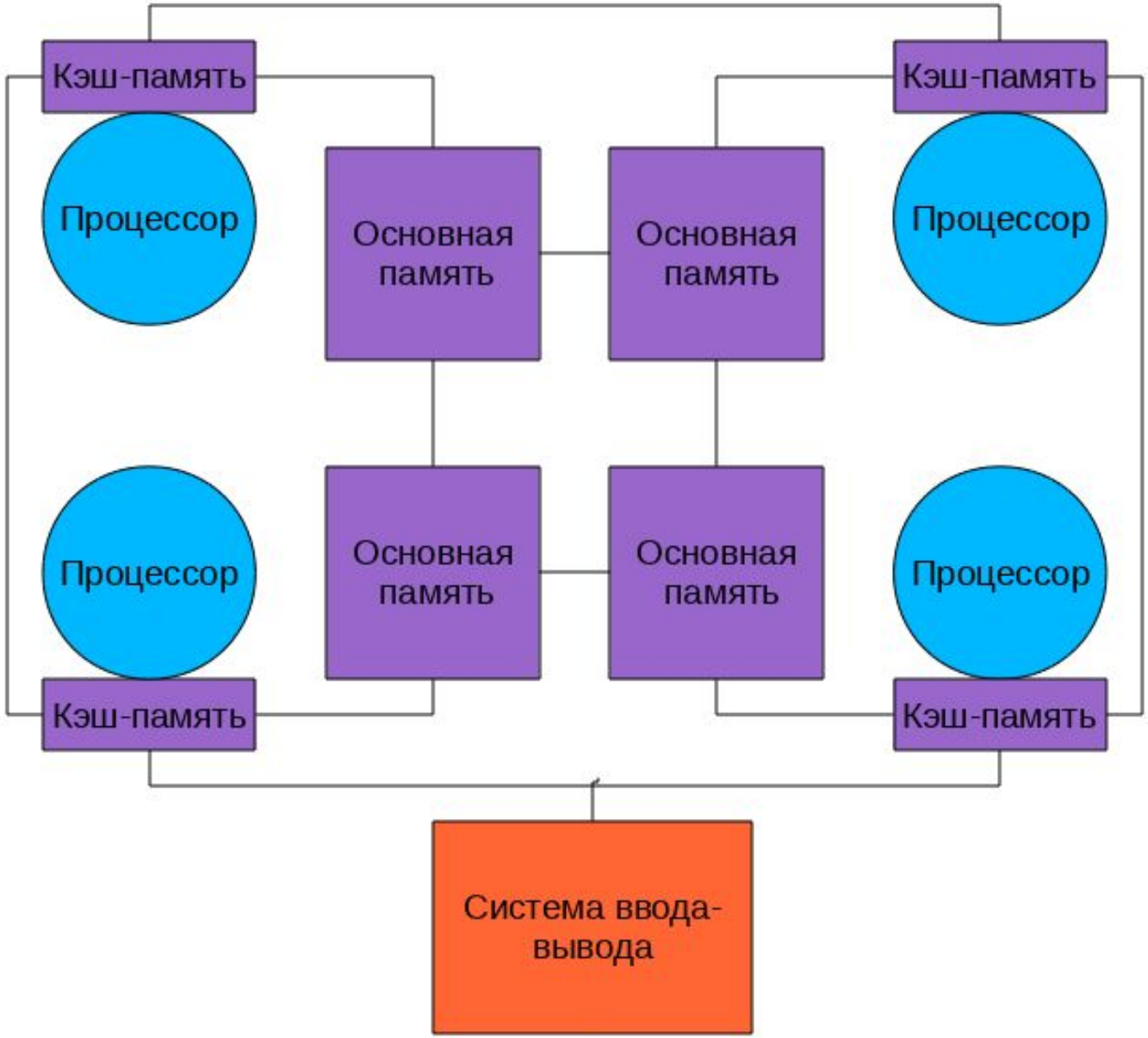
Многоядерные процессоры

- Общий кэш



NUMA

- **NUMA (Non-Uniform Memory Access — «неравномерный доступ к памяти» или Non-Uniform Memory Architecture — «Архитектура с неравномерной памятью») — архитектура, в которой, при общем адресном пространстве, скорость доступа к памяти зависит от ее расположения**


Кэш-память
Кэш-память
Процессор
Основная память
Основная память
Процессор
Процессор
Основная память
Основная память
Процессор
Кэш-память
Кэш-память
Система ввода-вывода