



Уральский  
федеральный  
университет

# Параллельные и распределенные вычисления

## Введение

**Созыкин Андрей Владимирович**

К.Т.Н.

Заведующий кафедрой высокопроизводительных компьютерных технологий  
Институт математики и компьютерных наук

# Контакты

Созыкин Андрей Владимирович

Заведующий кафедрой высокопроизводительных  
компьютерных технологий ИМKN УрФУ

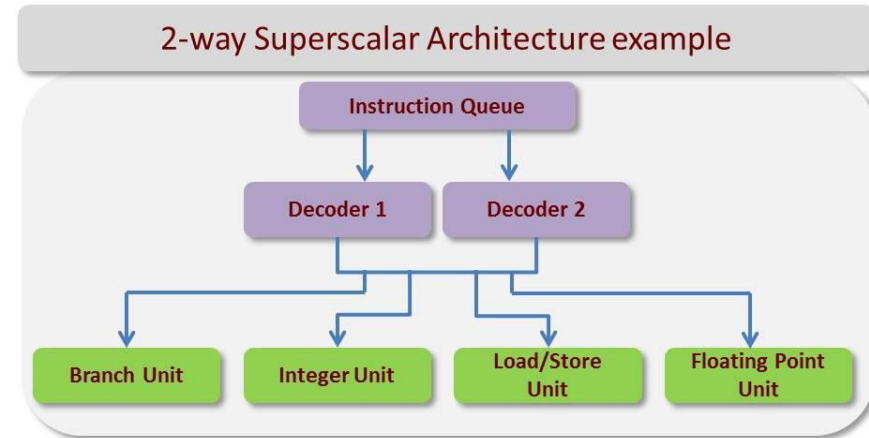
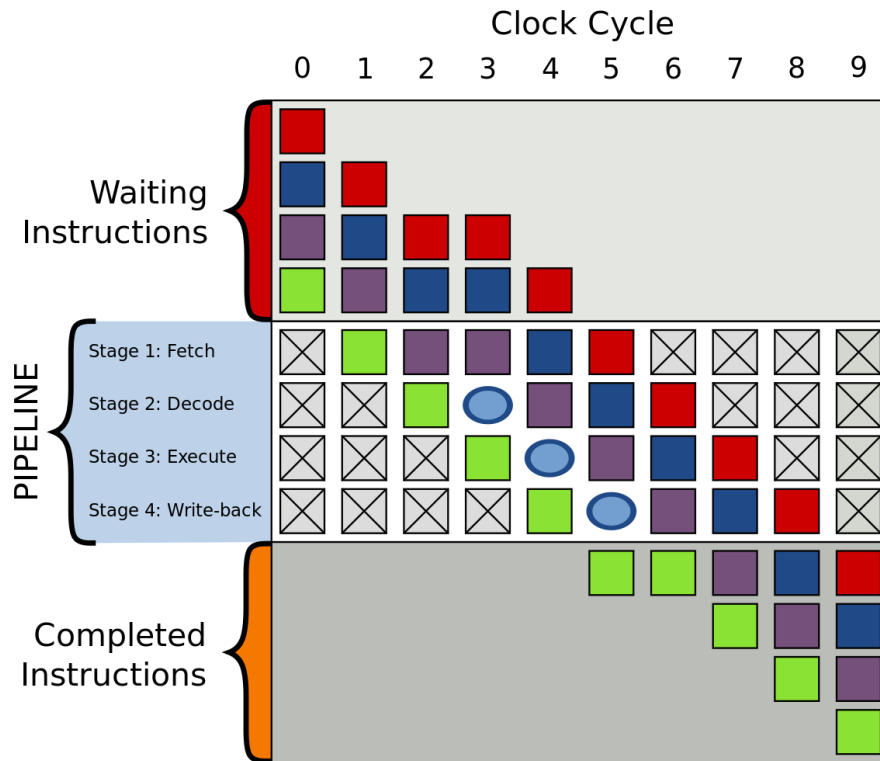
Email: [Andrej.Sozykin@urfu.ru](mailto:Andrej.Sozykin@urfu.ru)

[www.asozykin.ru](http://www.asozykin.ru)

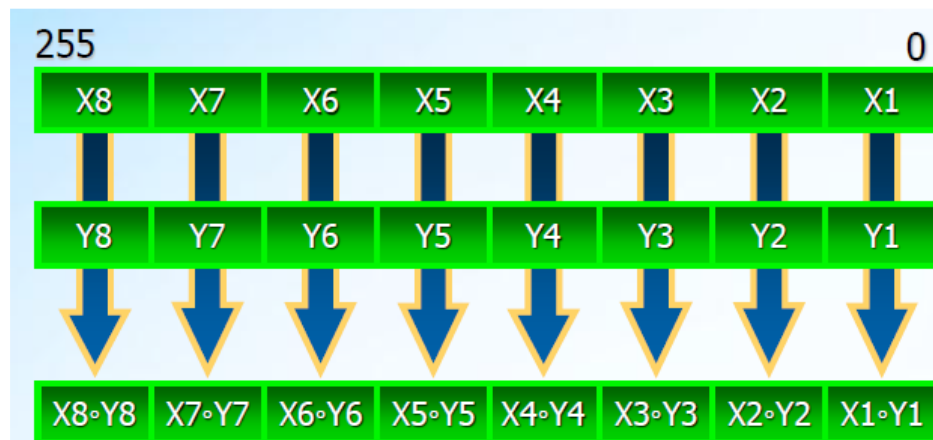
# Параллелизм в оборудовании



# Параллелизм инструкций (ILP)



# Параллелизм данных (DLP)



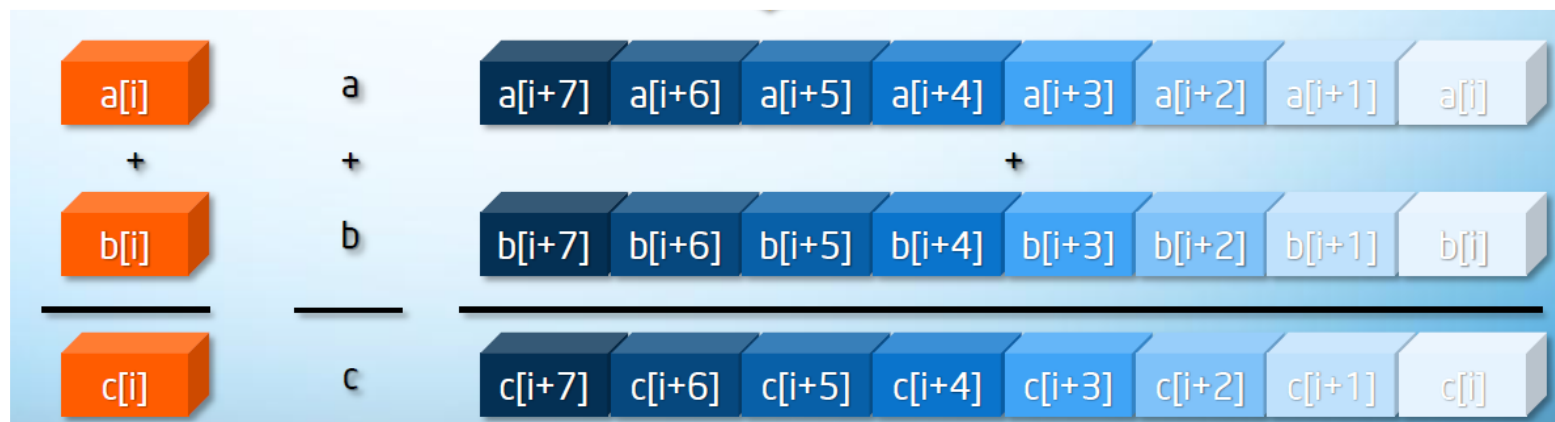
## AVX

Vector size: **256 bit**

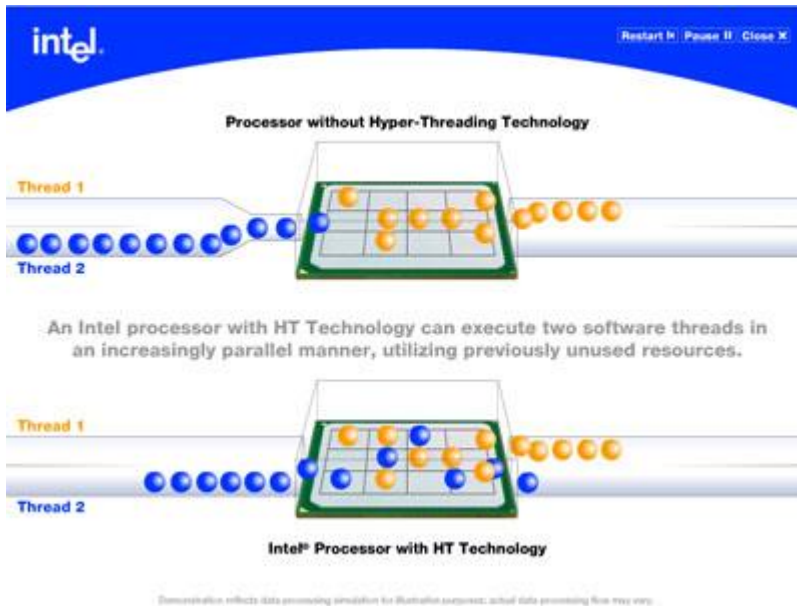
Data types:

- 32 and 64 bit float

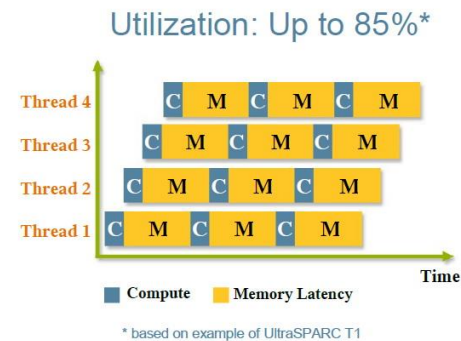
VL: 4, 8, 16



# Параллелизм потоков (TLP)



## Hardware Multi-Threading (HMT)



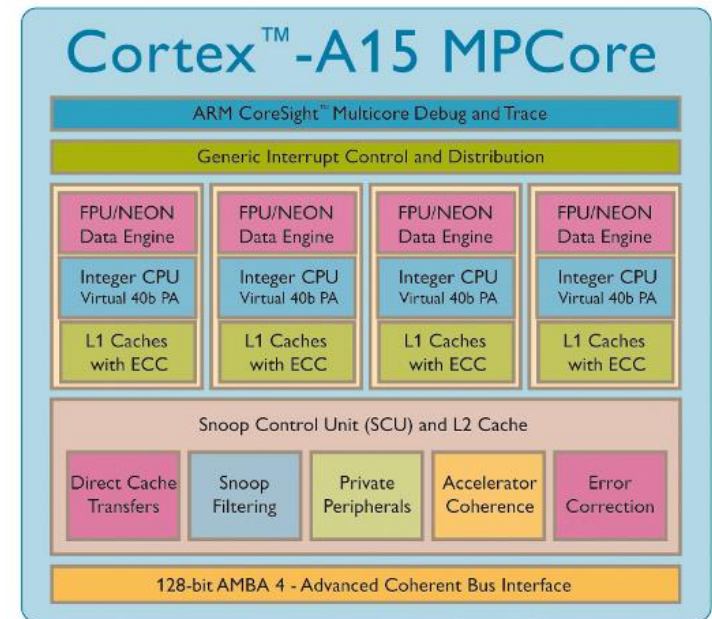
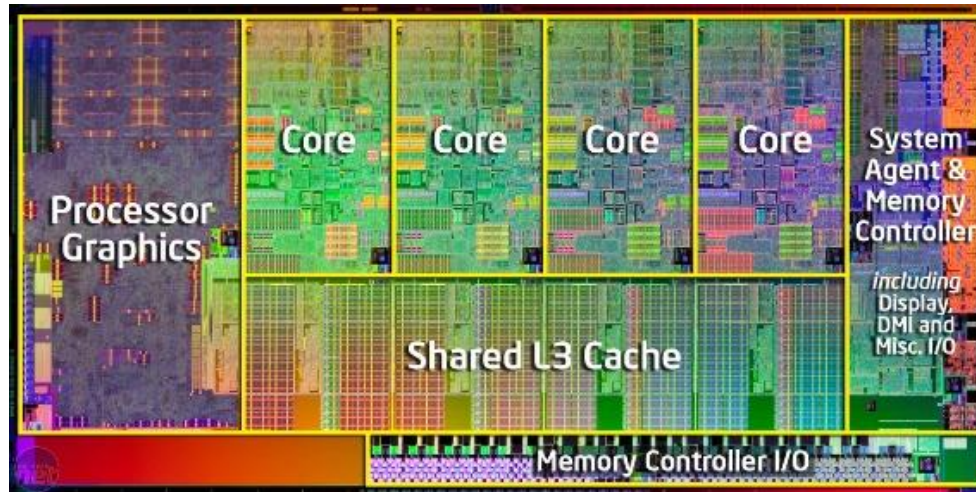
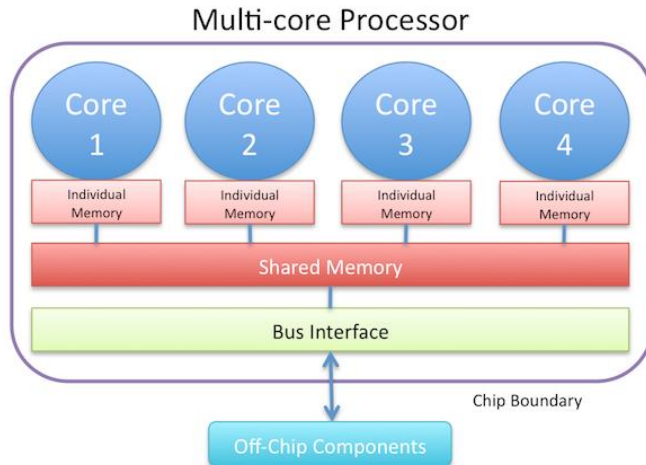
[www.OpenSPARC.net](http://www.OpenSPARC.net)

OpenSPARC – Dec 2007 – Chicago

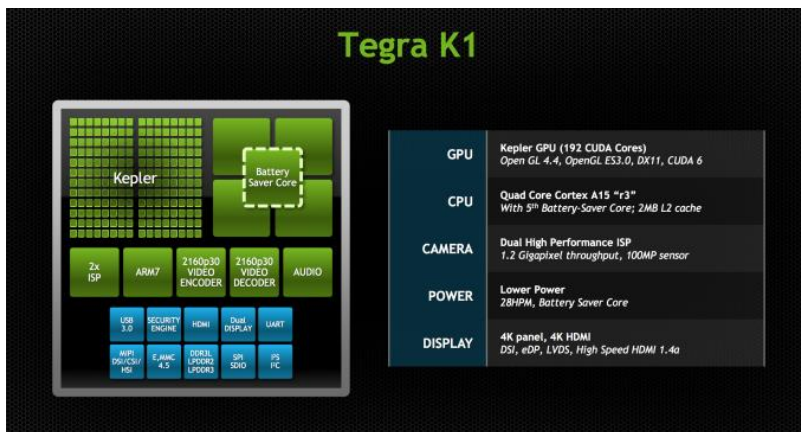
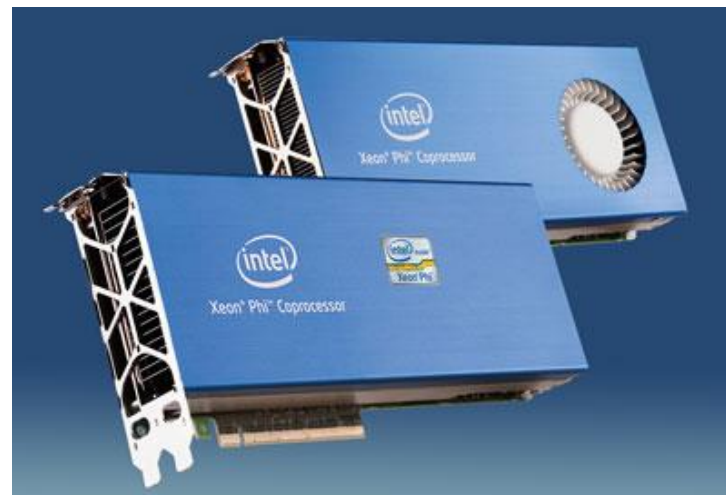
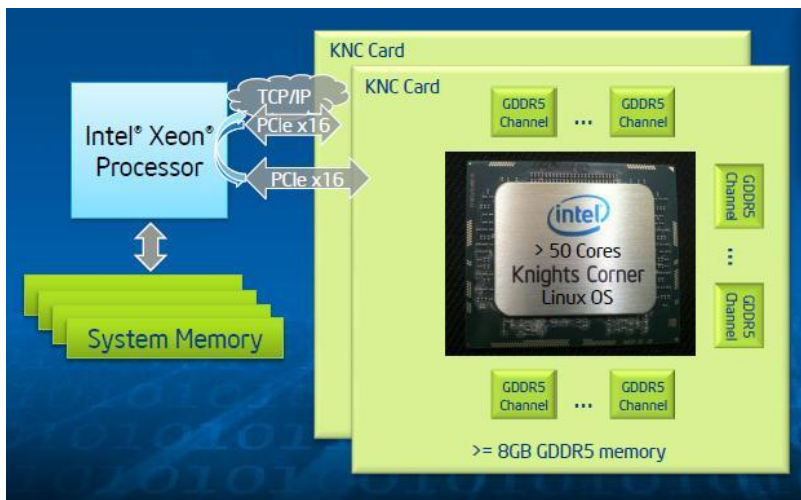
7



# Многоядерные процессоры (Multi Core CPU)



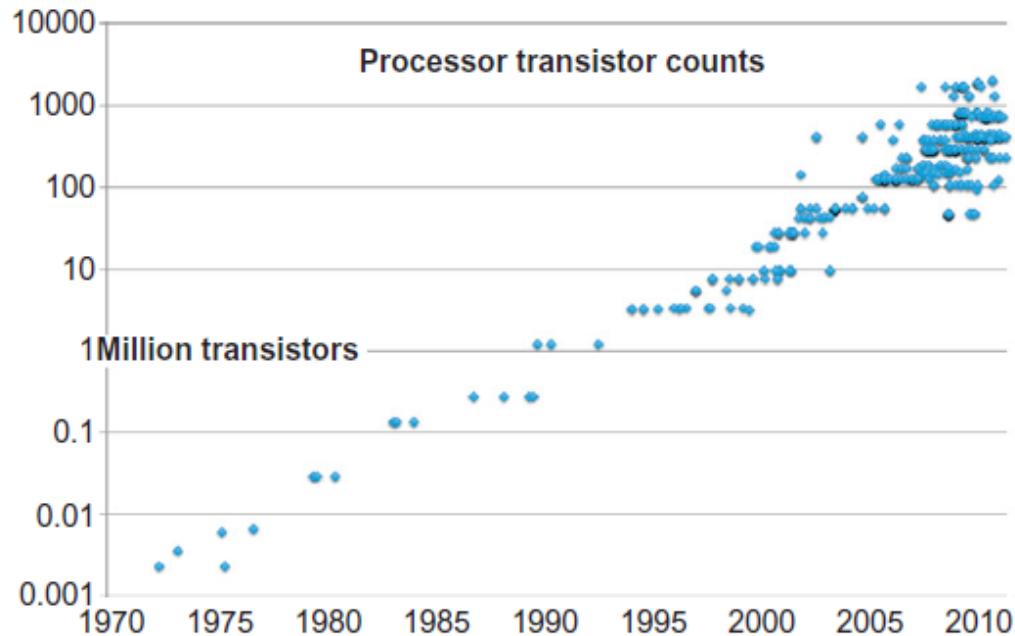
# Ускорители вычислений





# Почему так происходит?

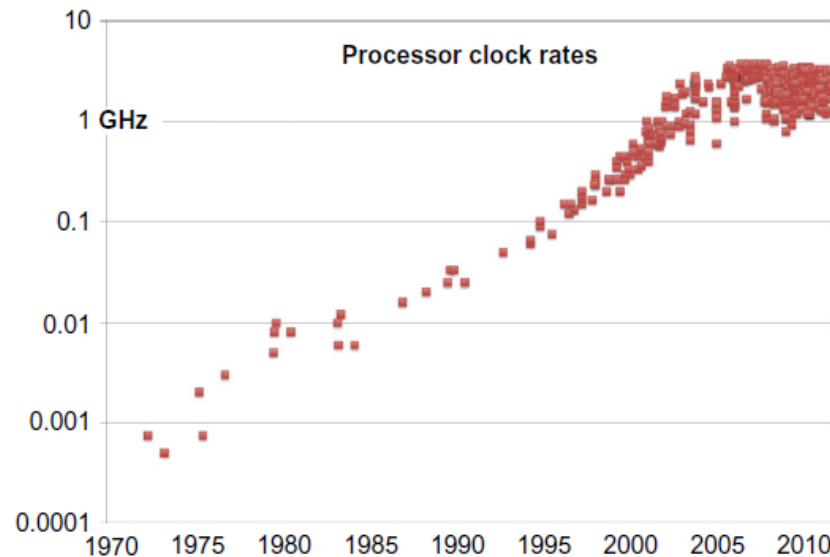
Закон Мура: количество транзисторов на кристалле удваивается примерно за 18 месяцев



M. McCool, J.Reinders, A.Robison. Structured Parallel Programming

# Тактовая частота процессоров не растет

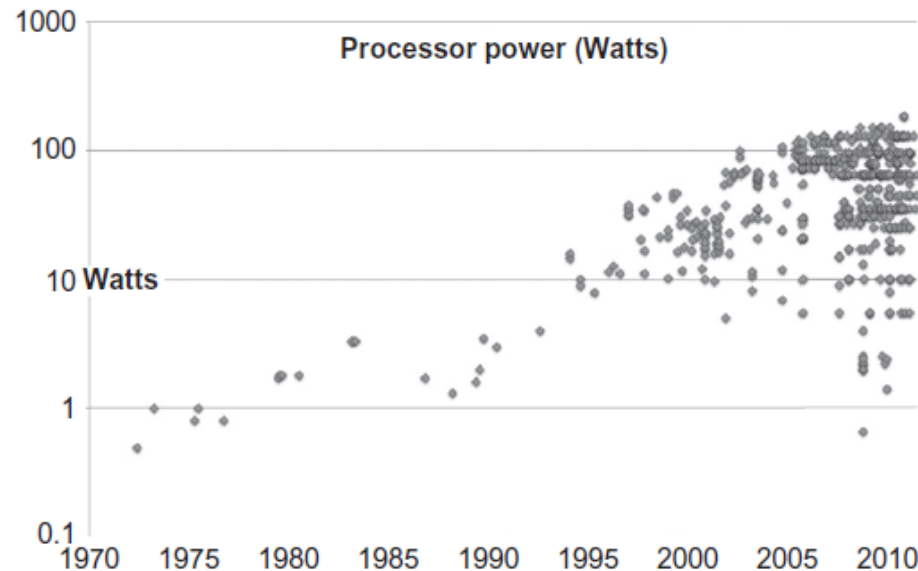
Закон Мура: количество транзисторов на кристалле удваивается примерно за 18 месяцев



M. McCool, J.Reinders, A.Robison. Structured Parallel Programming

# Power Wall

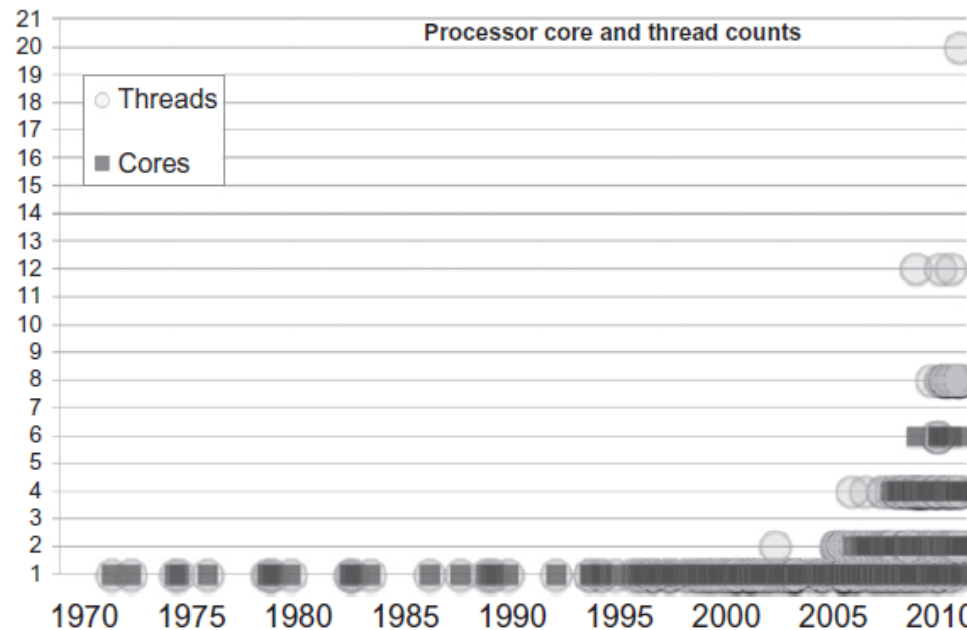
Неприемлемый рост энергопотребления при увеличении тактовой частоты ( $>130\text{Вт}$  нельзя охладить воздухом)



M. McCool, J.Reinders, A.Robison. Structured Parallel Programming

# Многоядерные процессоры

Вместо тактовой частоты увеличивается количество ядер/потоков



M. McCool, J.Reinders, A.Robison. Structured Parallel Programming

# Энергопотребление

$$\text{Power} = CV^2F$$

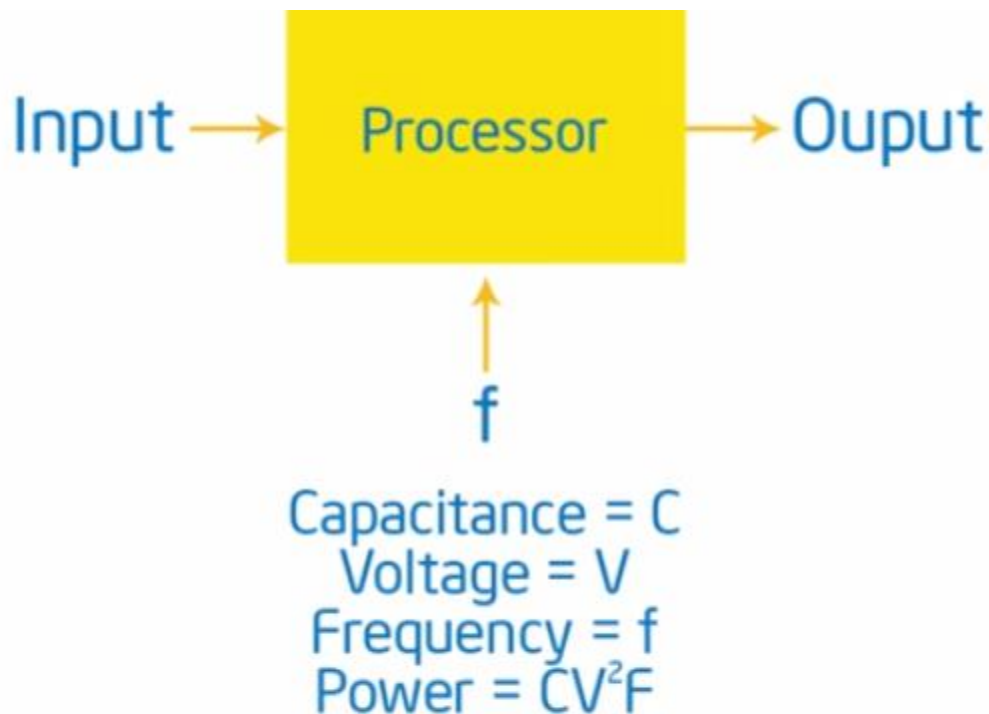
C – Capacitance, возможность микросхемы сохранять энергию

V – Voltage, напряжение

F – Frequency, частота

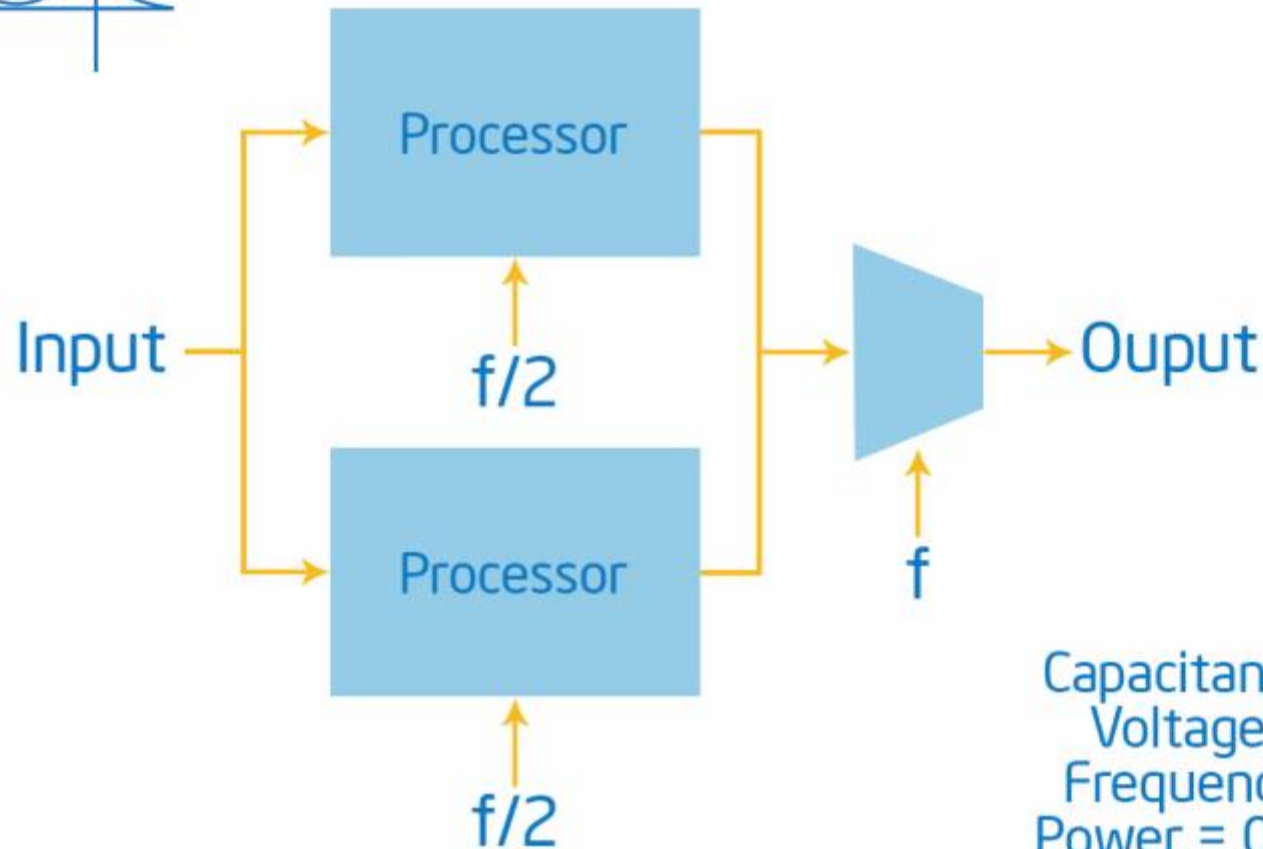


# Энергопотребление



Chandrakasan, A.P.; Potkonjak, M.; Mehra, R.; Rabaey, J.; Brodersen, R.W., "Optimizing power using transformations," IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, vol.14, no.1, pp.12-31, Jan 1995

# Энергопотребление



Chandrakasan, A.P.; Potkonjak, M.; Mehra, R.; Rabaey, J.; Brodersen, R.W., "Optimizing power using transformations," IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, vol.14, no.1, pp.12-31, Jan 1995

Source:  
Vishwani Agrawal

# Ядра и энергопотребление

Процессор с двумя ядрами:

- Производительность такая же, как и у одноядерного
- Энергопотребление – 40% от одноядерного

Как запускать программу на двух ядрах?

- Пусть об этом думают программисты!
- The problem was shifted
- “The Free Lunch Is Over”

*Herb Sutter*

<http://www.gotw.ca/publications/concurrency-ddj.htm>

# Другие «стены» тактовой частоты

## Memory wall:

- Скорость работы памяти намного меньше скорости процессора
- Промежуточное решение – кэш (уже занимает много места на кристалле процессора)
- Характеристики памяти: пропускная способность, задержка

## Instruction-level parallelism (ILP) wall:

- Ресурсы низкоуровневого параллелизма практически исчерпаны
- Количество исполнительных устройств (суперскалярность) – до 6 шт.
- Количество ступеней конвейера – до 10 шт. (максимум 31 шт.)

# Вычислительные кластеры



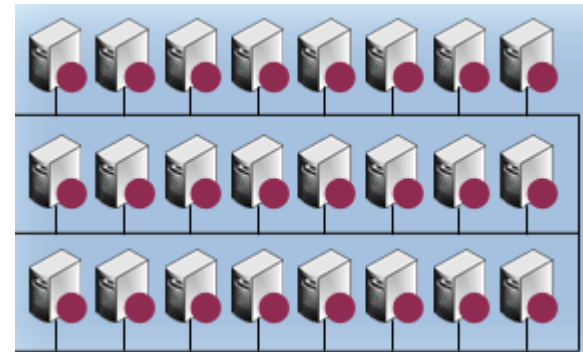


# Распределенные вычисления

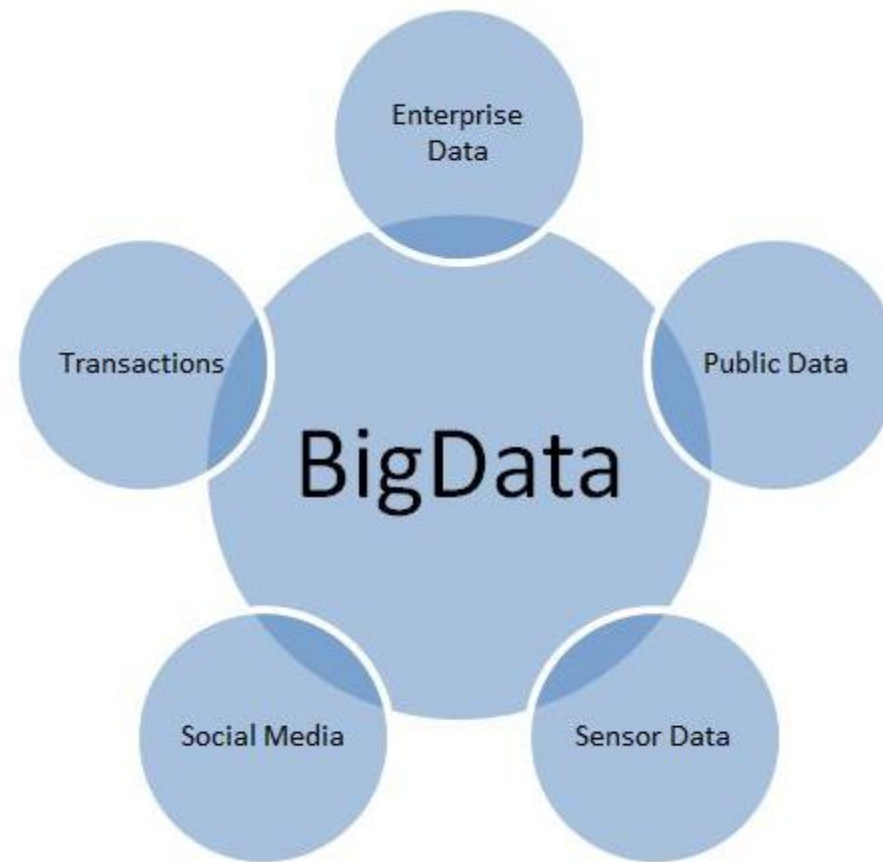
«If you were plowing a field, which would you rather use:  
Two strong oxen or 1024 chickens?»  
*Seymour Cray*



VS



# Большие данные



# Последовательные программы

Последовательное выполнение программы – это иллюзия!

Она обеспечивается:

- Компилятором
- Операционной системой

# Последовательные программы

Последовательное выполнение программы – это иллюзия!

Она обеспечивается:

- Компилятором
- Операционной системой

Зачем от нас скрывают правду?

# Параллельное программирование сложно

“Unnatural, error-prone, and untestable”

*R.K. Treiber, Coping with Parallelism, 1986*

“Too hard for most programmers to use”

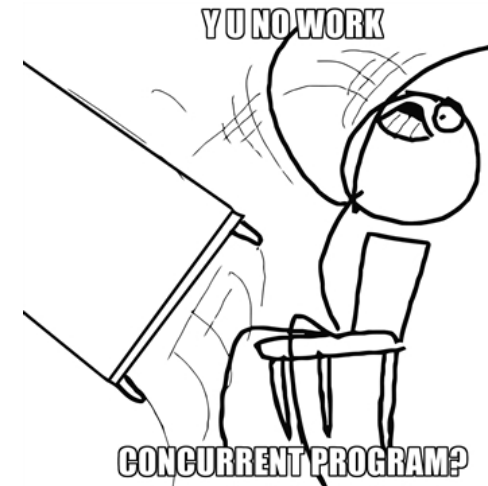
*Osterhout, Why Threads are a Bad Idea, 1995*

“It is widely acknowledged that concurrent programming is difficult”

*Edward Lee, The Problem with Threads, 2006*

“Is Parallel Programming Hard, And, If So, What Can You Do About It?”

*Paul E. McKenney, 2014 (книга, 509 страниц)*





# Проблемы параллельного программирования

## Недетерминированность

- Разные трассы выдают разные результаты
- Невозможно тестировать традиционными средствами
- Сложно отлаживать

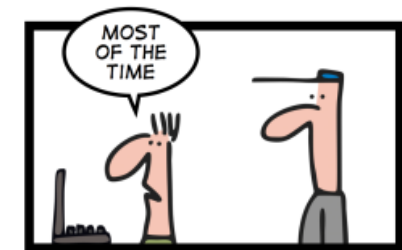
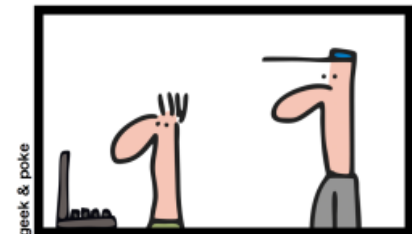
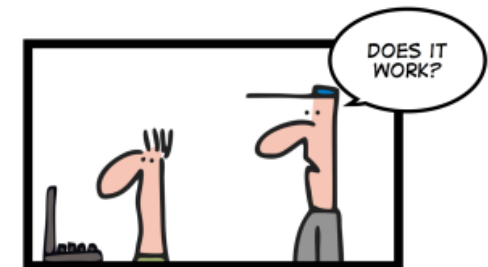
## Условия гонок

## Взаимоблокировки

## Масштабирование

## Балансировка нагрузки

SIMPLY EXPLAINED



CONCURRENCY

# Параллельное программирование сложно

Параллельное программирование требует **другого образа мышления** по сравнению с последовательным

- Недостаточно просто выучить новые конструкции

**Сложность** освоения параллельного программирования **сравнима** со сложностью освоения **объектно-ориентированного программирования**

# Содержание курса

1. Многопоточное программирование
2. Параллельное программирование:
  - OpenMP
  - Векторизация
  - MPI
3. Распределенные вычисления и BigData:
  - MapReduce и Hadoop
  - Экосистема Hadoop (Hive, Pig, Mahout, HBase и т.п.)

Языки программирования:

- C/C++: многопоточность (C++11), OpenMP, MPI
- Java: MapReduce

# Структура курса

Лекции: примерно 12 шт.

Домашние задания:

- Многопоточное программирование (2 задачи)
- Параллельное программирование (2 задачи)
- Распределенные вычисления (2 задачи)

Первое домашнее задание на своих компьютерах, второе и третье на кластерах ИММ УрО РАН

Дедлайны:

- Для каждого ДЗ будет дедлайн (2-3 недели), после которого задачи не принимаются

# Экзамен

Экзамен ставится по результатам домашних заданий:

- 6 задач, максимум 5 баллов за каждую задачу

Оценки за экзамен:

- 25 баллов - отлично
- 21 балл - хорошо
- 17 баллов – зачет (удовлетворительно)



# Материалы лекций и ДЗ

Страница курса:

- <http://www.asozykin.ru/courses/parallel-computing>

Домашние задания:

- <https://clck.ru/8rEVG>

Примеры кода:

- <https://github.com/sozykin/parallel-computing-course>

Видеолекции московского ШАД:

- <http://www.youtube.com/playlist?list=PLJOzdkh8T5krFksX90QkuntWC6vflDZZU>

# Вопросы?