NYU
New York University

CSCI-GA.3033-012
Multicore Processors: Architecture & Programming

Lecture 1: Multicore/Manycore Revolution

Mohamed Zahran (aka Z)
mzahran@cs.nyu.edu
http://www.mzahran.com

Who Am I?

- Mohamed Zahran (aka Z)
- http://www.mzahran.com
- Research interest:
  - computer architecture
  - hardware/software interaction
  - Biologically-inspired machines
- Office hours: Wed 4:00-6:00 pm
  - or by appointment
- Room: WWH 320

Formal Goals of This Course

- What are multicore/manycore processors?
- Why do we have them?
- What are the challenges in dealing with them?
- How to make the best use of them in our software?

Informal Goals of This Course

- Don't be afraid of hardware
- Understand the hardware/software interaction
- Enjoy the challenge of making the best use of hardware to the benefit of software
- Enhance your way of thinking about parallelism and parallel programming models
- Build a vision about technology and its future

The Course Web Page

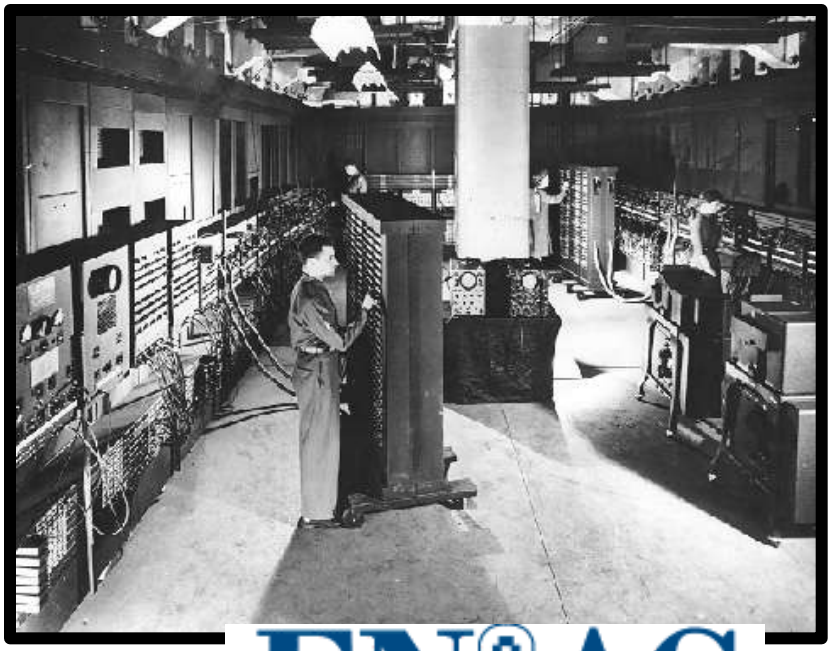
- Lecture slides
- Info about mailing list, labs, … .
- Useful links (tools, articles, …)

http://cs.nyu.edu/courses/fall12/CSCI-GA.3033-012/index.html

Grading

- Homework assignments 20%
- Programming assignments 20%
- Project 60%

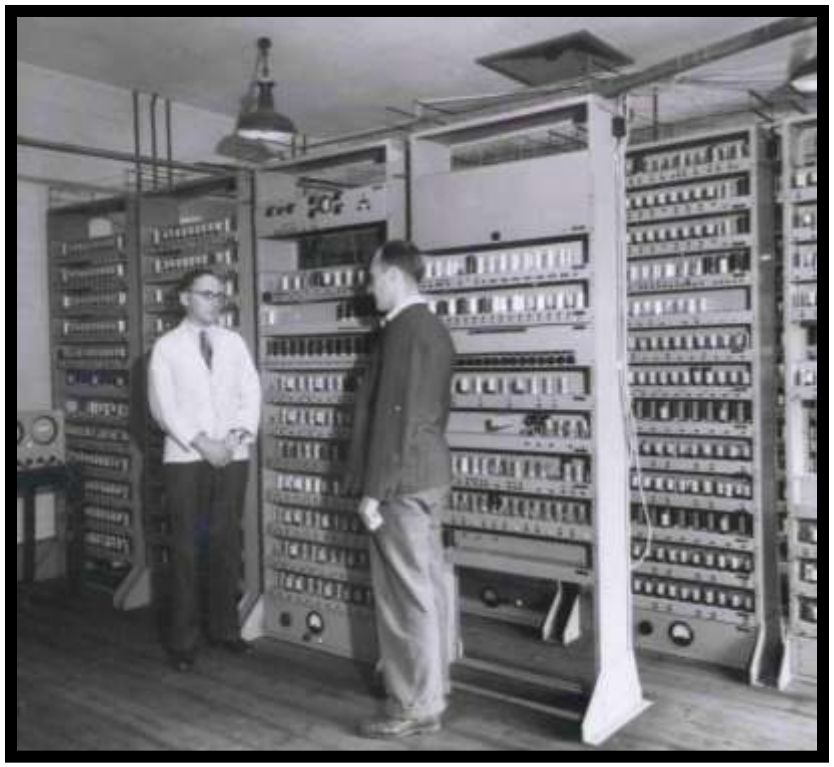
Computer History



Eckert and Mauchly

- 1st working electronic computer (1946)
- 18,000 Vacuum tubes
- 1,800 instructions/sec
- 3,000 ft^3

Computer History



EDSAC 1 (1949)

http://www.cl.cam.ac.uk/UoCCL/misc/EDSAC99/

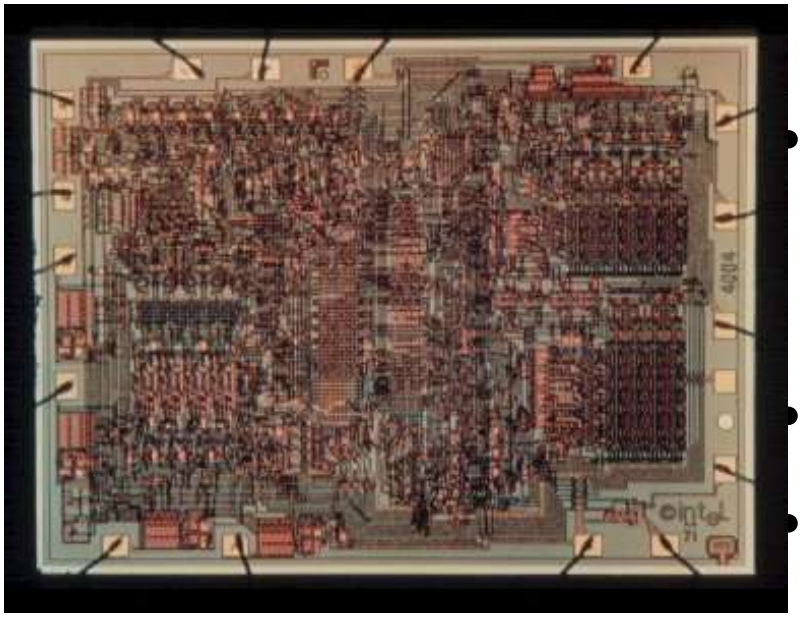
- Maurice Wilkes

1st stored program computer

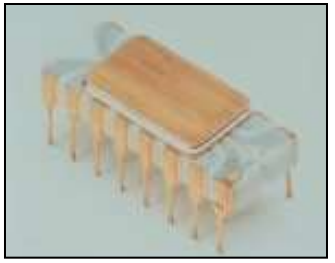
650 instructions/sec

1,400 ft^3

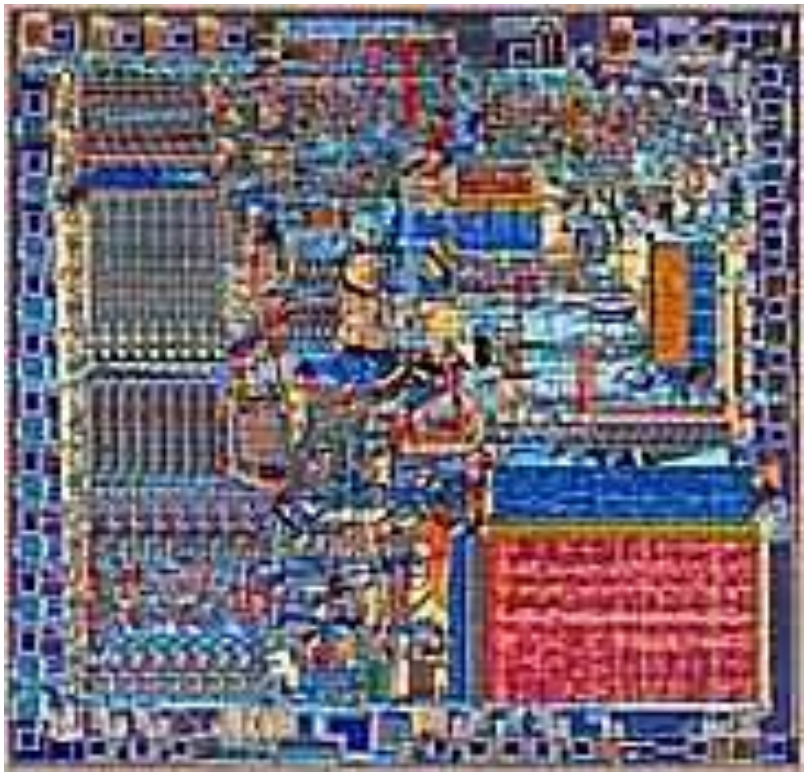
Intel 4004 Die Photo



- Introduced in 1970
  - First microprocessor
- 2,250 transistors
- 12 mm^2
- 108 KHz

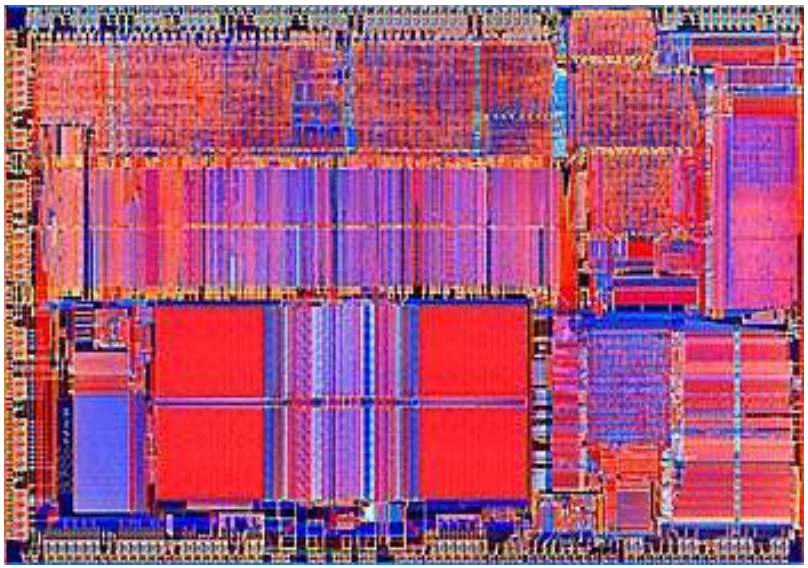


Intel 8086 Die Scan



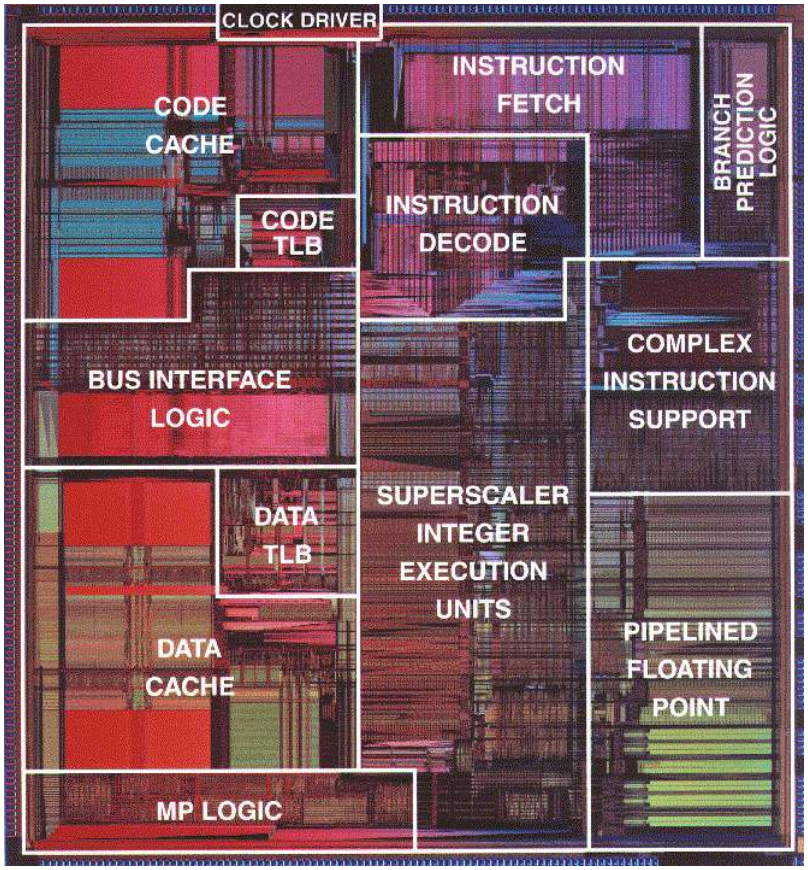
- 29,000 transistors
- 33 mm^2
- 5 MHz
- Introduced in 1979
  - Basic architecture of the IA32 PC

Intel 80486 Die Scan



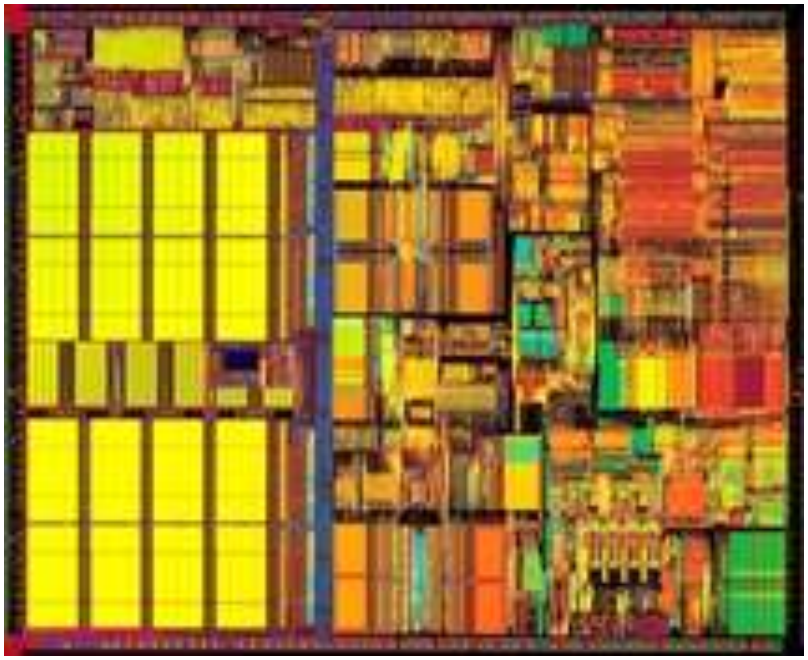
- 1,200,000 transistors
- 81 mm^2
- 25 MHz
- Introduced in 1989
  - 1st pipelined implementation of IA32

Pentium Die Photo



- 3,100,000 transistors
- 296 mm^2
- 60 MHz
- Introduced in 1993
  - 1st superscalar implementation of IA32

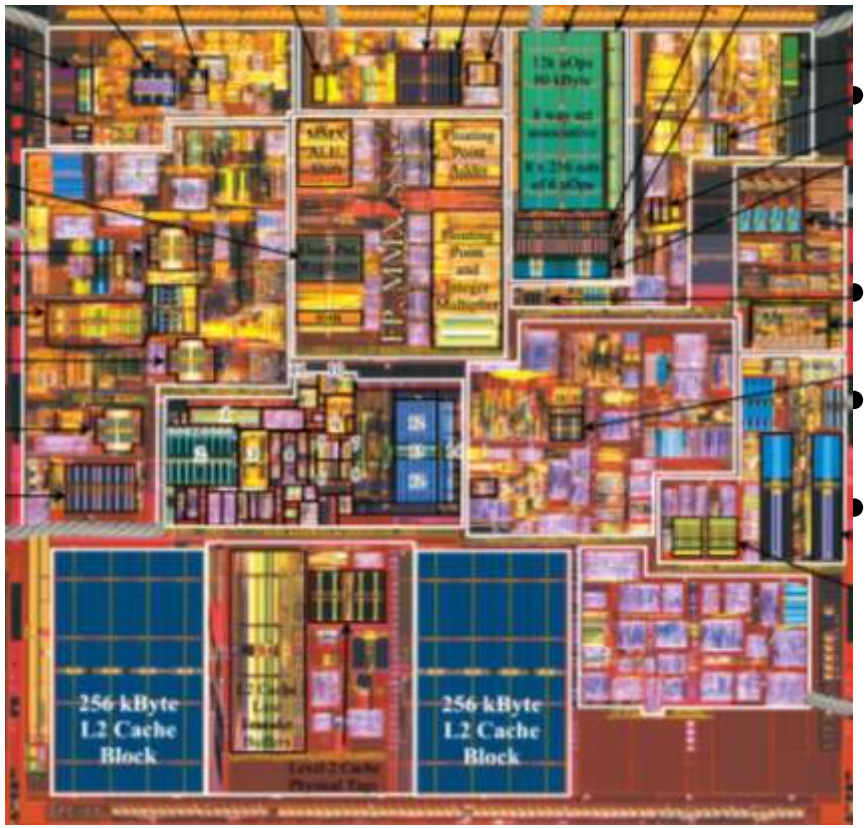
Pentium III



- 9,500,000 transistors
- 125 mm^2
- 450 MHz
- Introduced in 1999

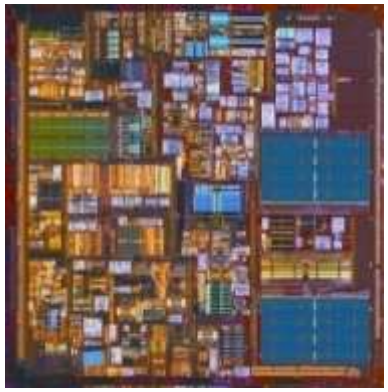
http://www.intel.com/intel/museum/25anniv/hof/hof_main.htm

Pentium 4

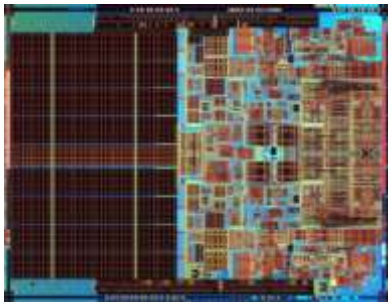


- 55,000,000 transistors
- 146 mm^2
- 3 GHz
- Introduced in 2000

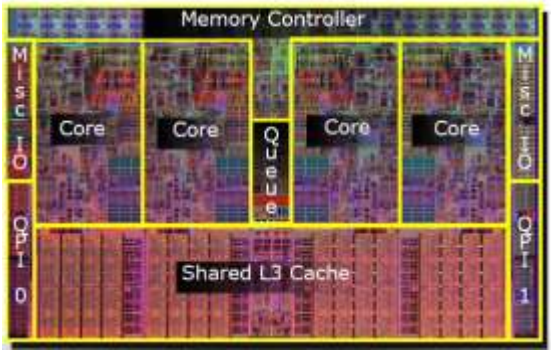
http://www.chip-architect.com



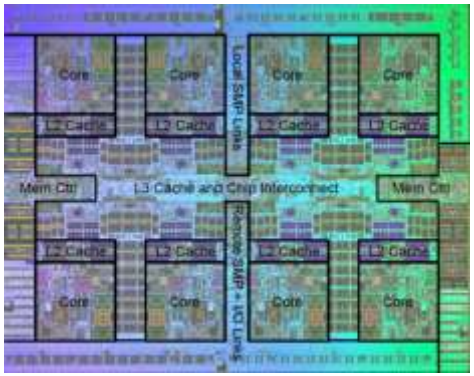
Pentium 4



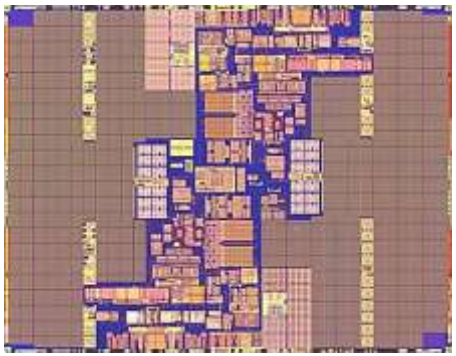
Core 2 Duo (Merom)



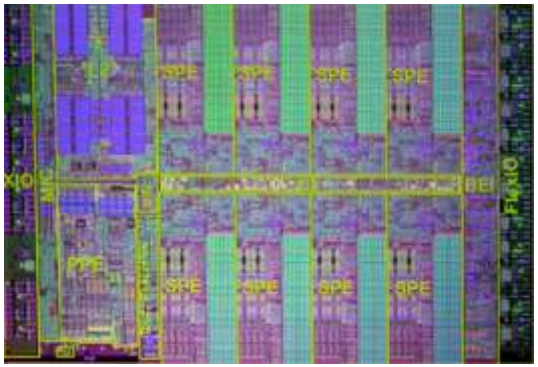
Intel Core i7 (Nehalem)



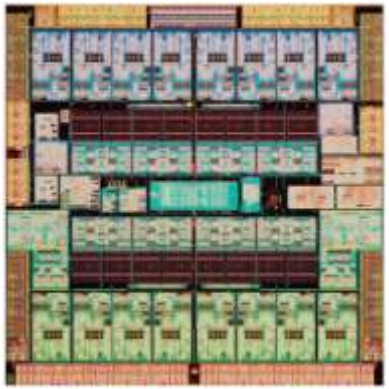
IBM Power 7



Montecito (Itanium 2)



Cell Processor

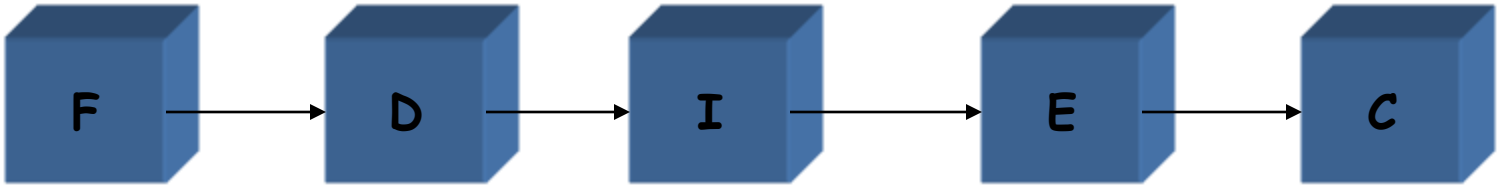


(SUN UltraSparc T3)

First Generation (1970s)

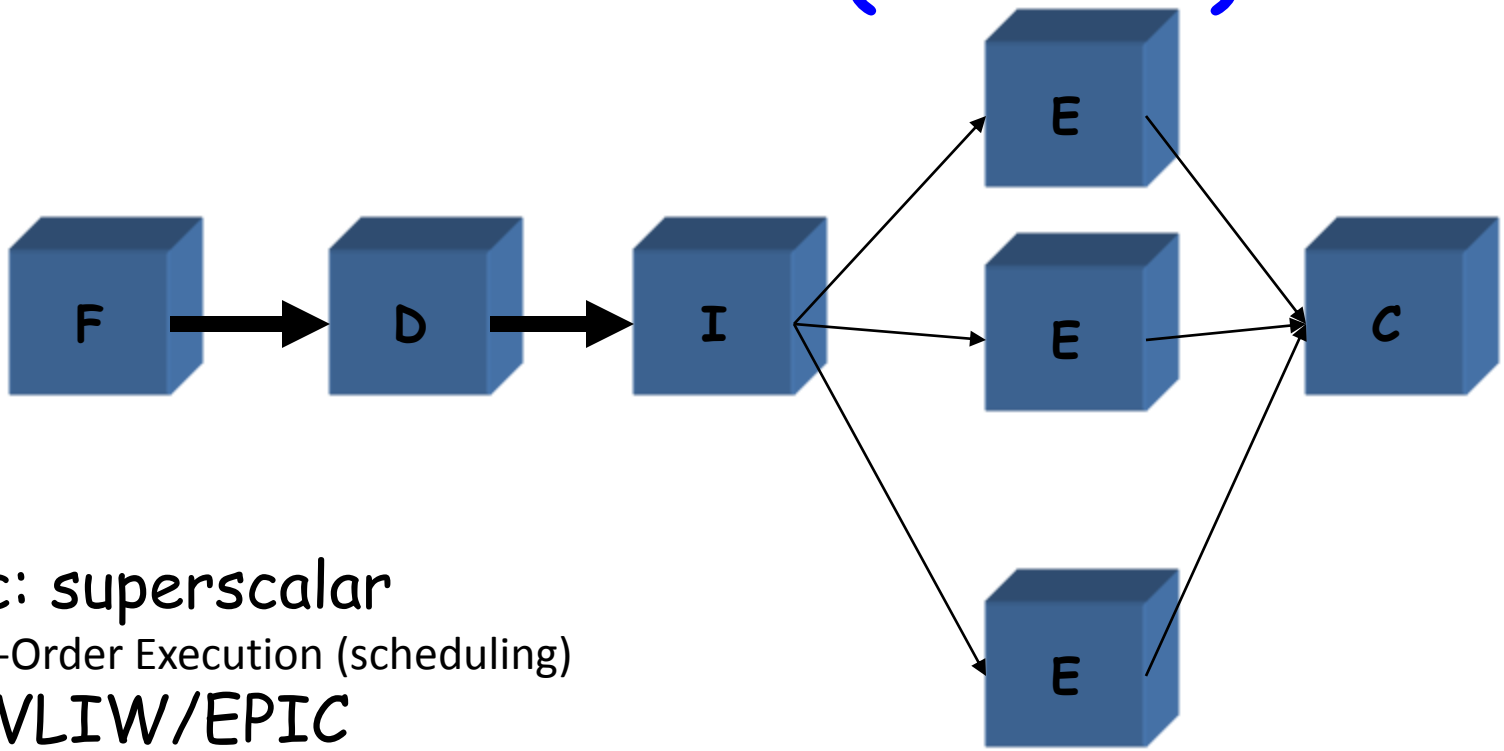
Single Cycle Implementation

Second Generation (1980s)



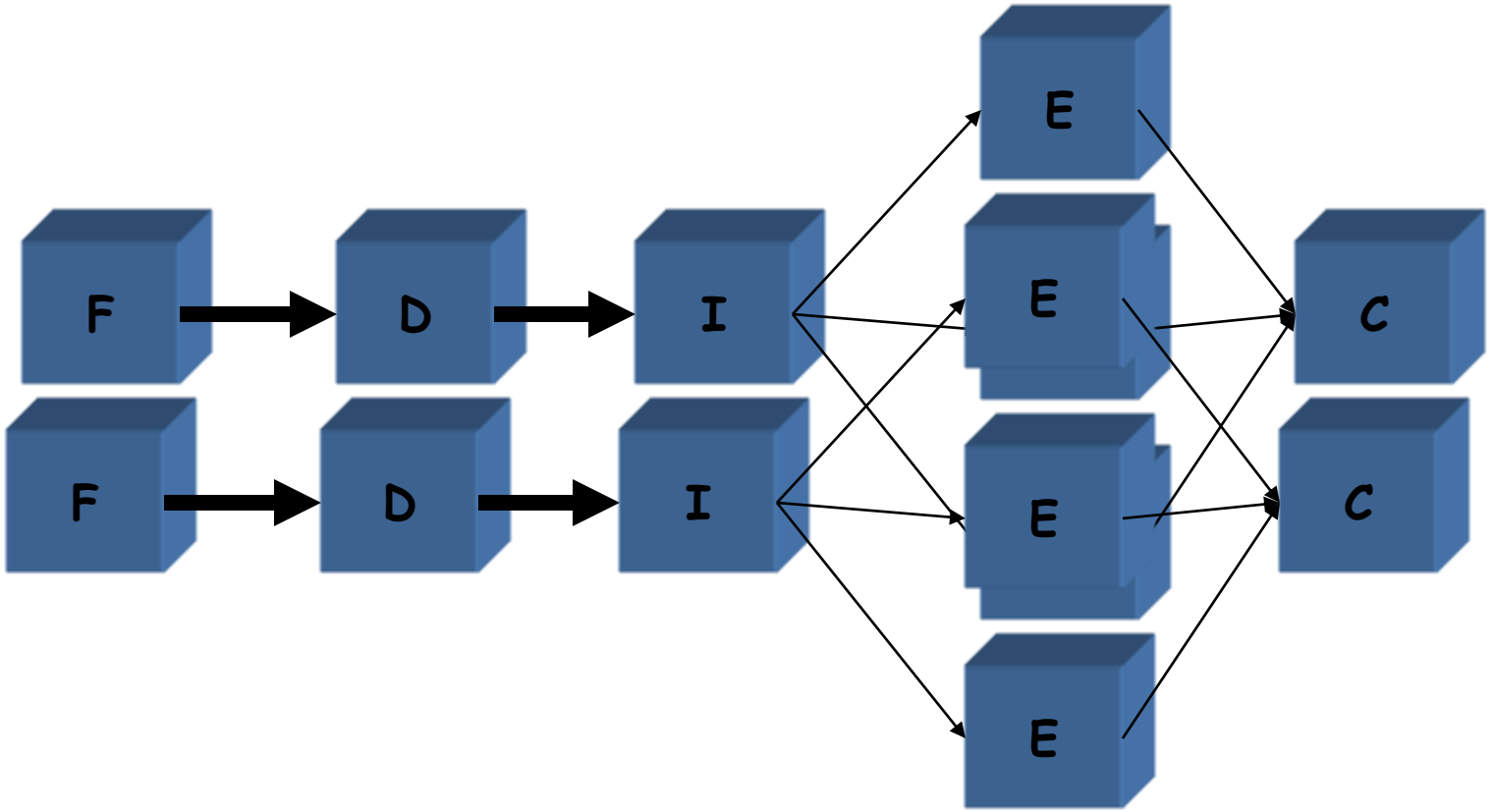
- Pipelinining: temporal parallelism
- Number of stages increase with each generation
- Maximum CPI = 1

Third Generation (1990s)



- ILP
  - Dynamic: superscalar
    - Out-Of-Order Execution (scheduling)
  - Static: VLIW/EPIC
- Spatial parallelism
- IPC not CPI
- Instruction window
- Speculative Execution (prediction)

Fourth Generation (2000s)



Simultaneous Multithreading (SMT)
(aka Hyperthreading Technology)

The Status-Quo

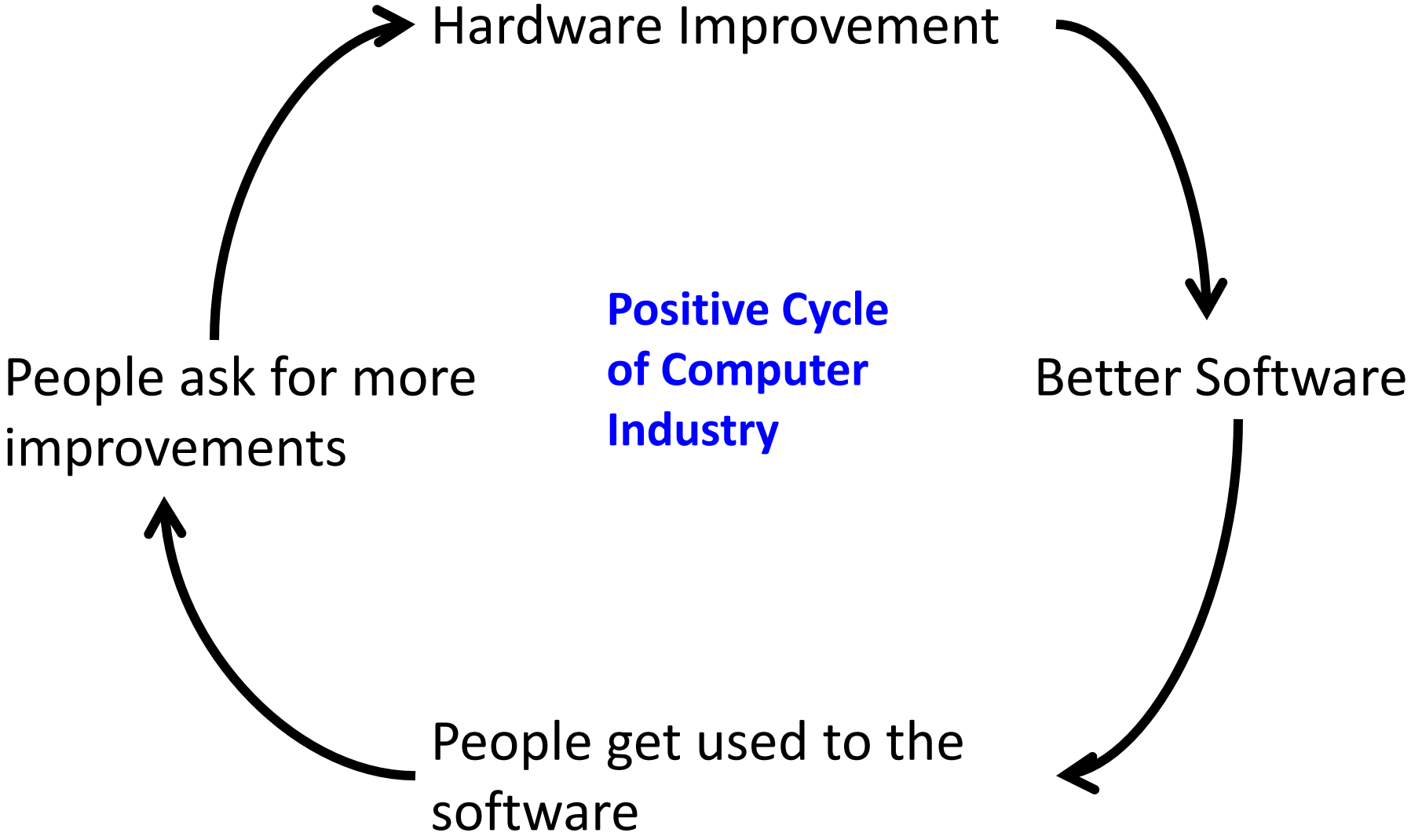
- We moved from single core to multicore to manycore:
  - for technological reasons
- Free lunch is over for software folks
  - The software will not become faster with every new generation of processors
- Not enough experience in parallel programming
  - Parallel programs of old days were restricted to some elite applications -> very few programmers
  - Now we need parallel programs for many different applications

The Famous Moore's Law

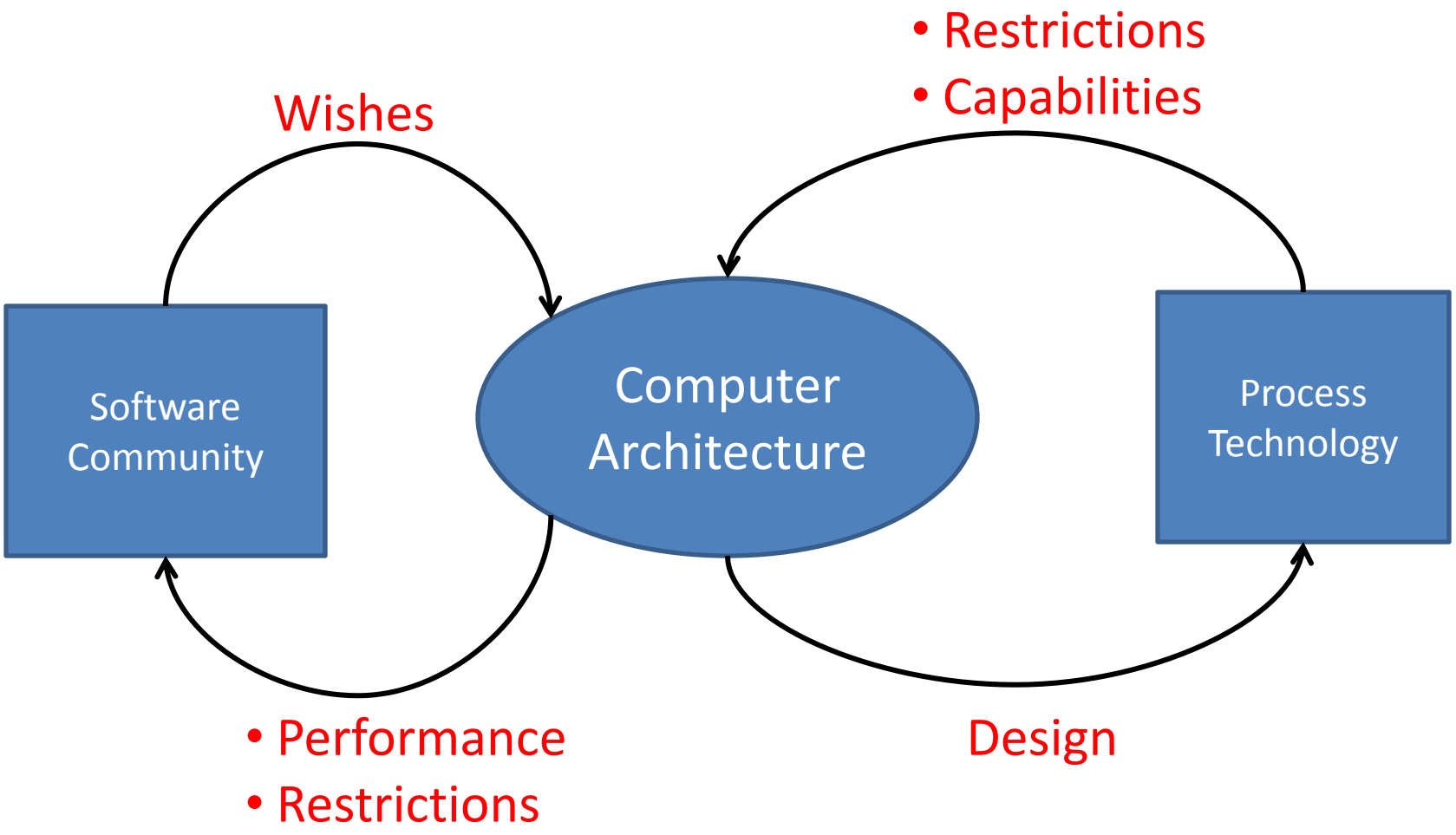
Moore's law works because of …

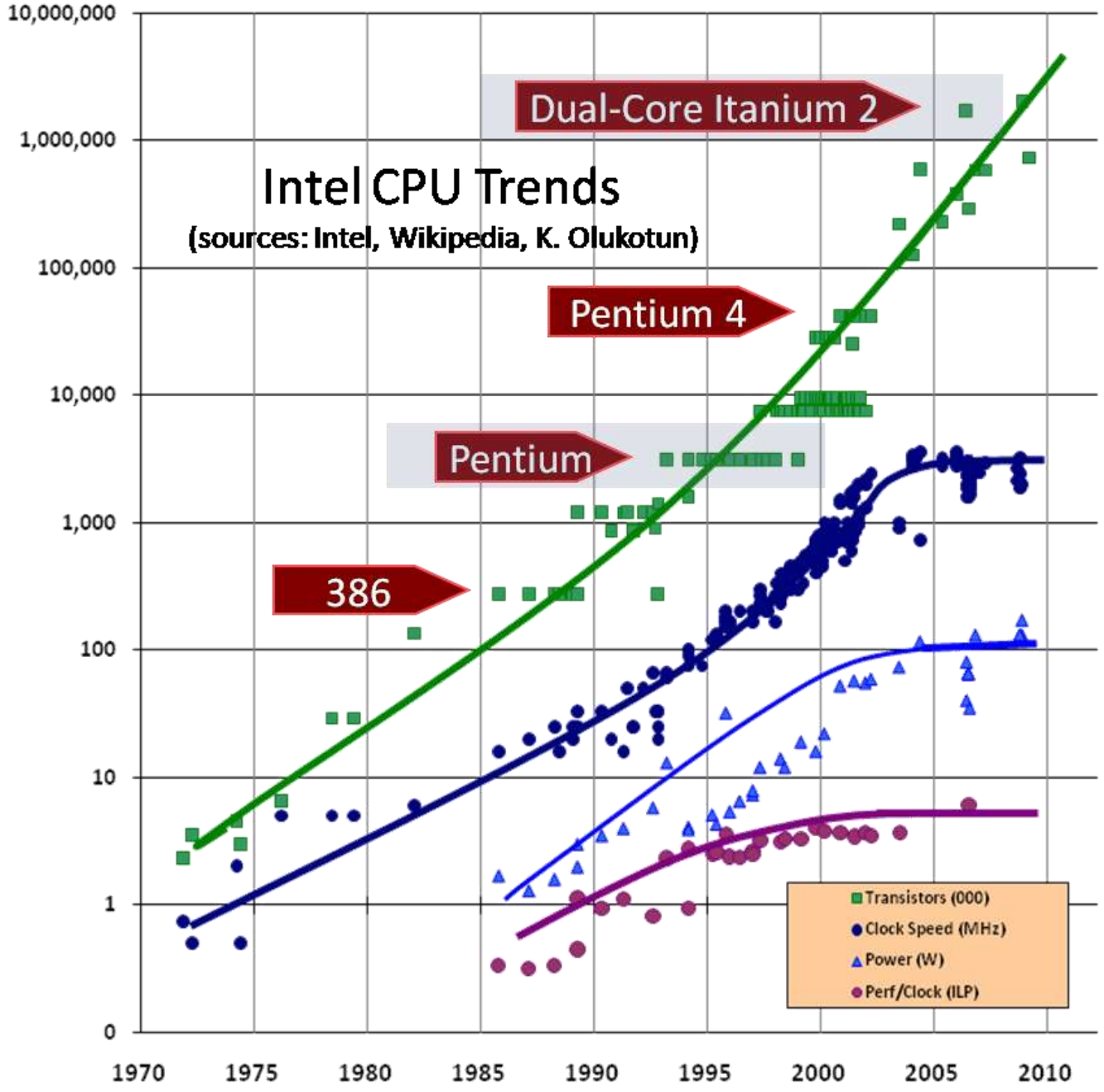
Dennard scaling

MOSFETs continue to function as voltage-controlled switches while all key figures of merit such as layout density, operating speed, and energy efficiency improve provided geometric dimensions, voltages, and doping concentrations are consistently scaled to maintain the same electric field.


Hardware Improvement
Better Software
People get used to the software
People ask for more improvements
Positive Cycle of Computer Industry

How Did These Advances Happen?





Performance in the past achieved by:

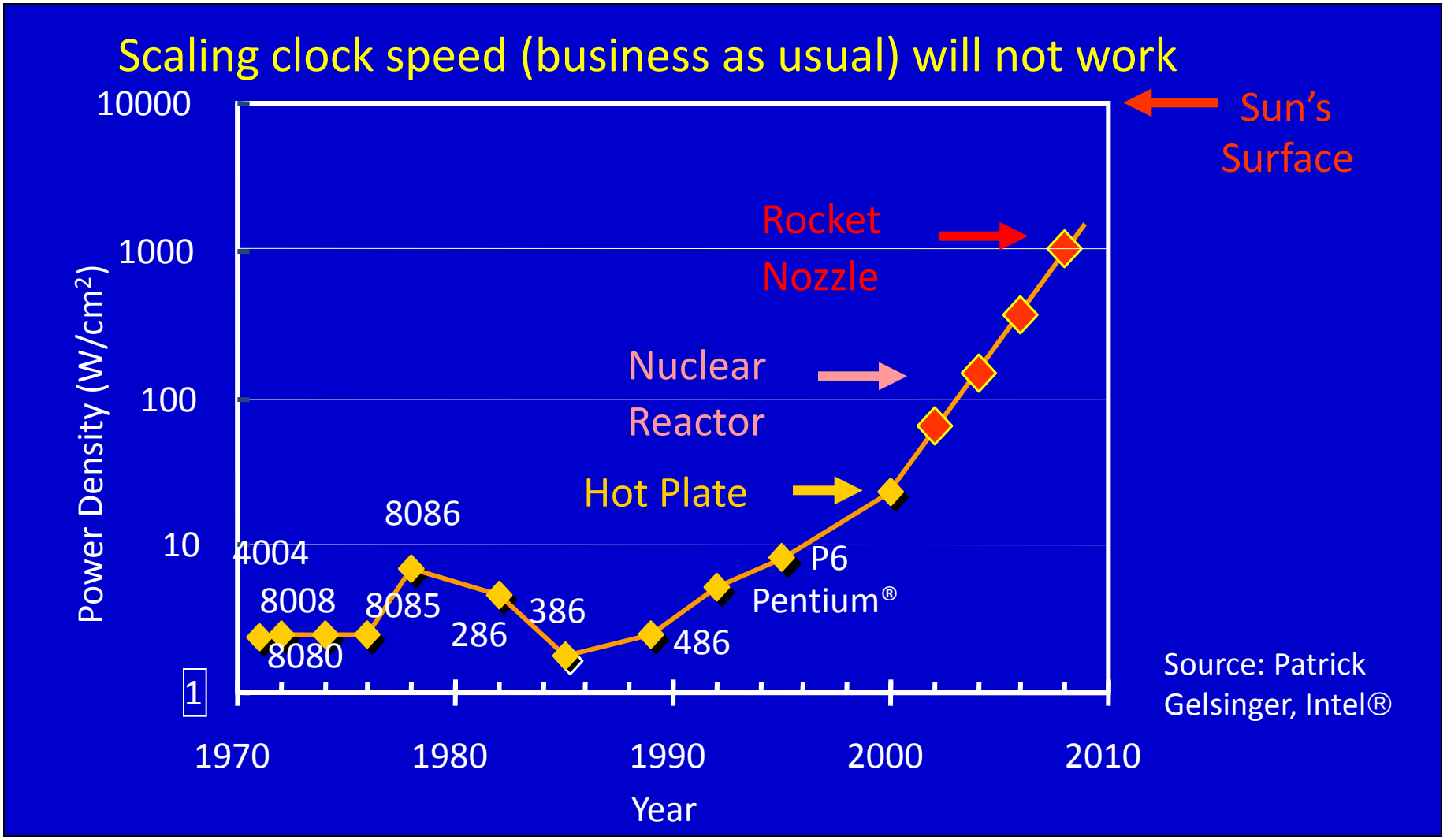
- clock speed
- execution optimization
- cache

Performance now achieved by:

- hyperthreading
- multicore
- cache

Power Density

Moore's law is giving us more transistors than we can afford!



Multicore Processors Save Power

Power = C * V^2 * F

Performance = Cores * F

Let's have two cores

Power = 2*C * V^2 * F

Performance = 2*Cores * F

But decrease frequency by 50%

Power = 2*C * V^2/4 * F/2

Performance = 2*Cores * F/2

Power = C * V^2/4 * F

Performance = Cores * F

Late 20th Century	The New Reality
Moore's Law — 2× transistors/chip every 18-24 months	Transistor count still 2× every 18-24 months, but see below
Dennard Scaling — near-constant power/chip	Gone. Not viable for power/chip to double (with 2× transistors/chip growth)
The modest levels of transistor unreliability easily hidden (e.g., via ECC)	Transistor reliability worsening, no longer easy to hide
Focus on computation over communication	Restricted inter-chip, inter-device, inter-machine communication (e.g. Rent's Rule, 3G, GigE); communication more expensive than computation
One-time (non-recurring engineering) costs growing, but amortizable for mass-market parts	Expensive to design, verify, fabricate, and test, especially for specialized-market platforms

A Case for Multicore Processors

- Can exploit different types of parallelism
- Reduces power
- An effective way to hide memory latency
- Simpler cores = easier to design and test = higher yield = lower cost

The Need for Parallel Programming

Parallel computing: using multiple processors in parallel to solve problems more quickly than with a single processor

Examples of parallel machines:

- A **cluster computer** that contains multiple PCs combined together with a high speed network
- A **shared memory multiprocessor** (SMP) by connecting multiple processors to a single memory system
- A **Chip Multi-Processor** (CMP) contains multiple processors (called cores) on a single chip

Cost and Challenges of Parallel Execution

- Communication cost
- Synchronization cost
- Not all problems are amenable to parallelization
- Hard to think in parallel
- Hard to debug

Attempts to Make Multicore Programming Easy

- 1st idea: The right computer language would make parallel programming straightforward
  - **Result so far**: Some languages made parallel programming easier, but none has made it as fast, efficient, and flexible as traditional sequential programming.

Attempts to Make Multicore Programming Easy

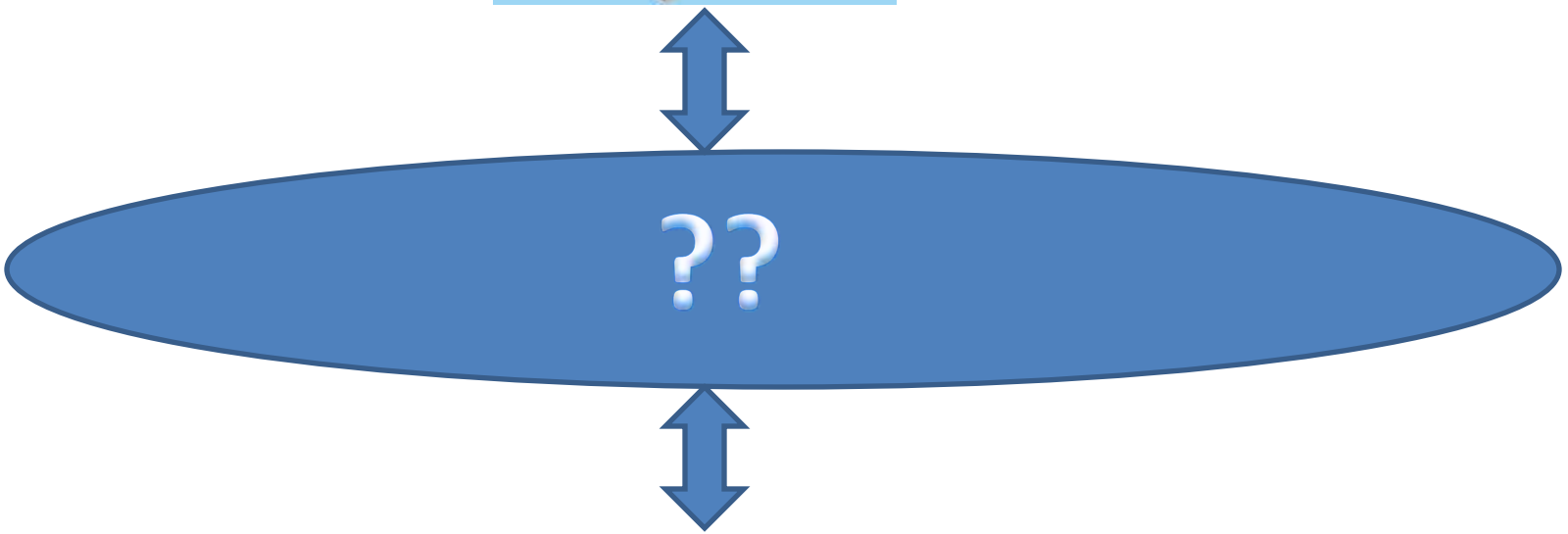
- 2nd idea: If you just design the hardware properly, parallel programming would become easy.
  - **Result so far**: no one has yet succeeded!

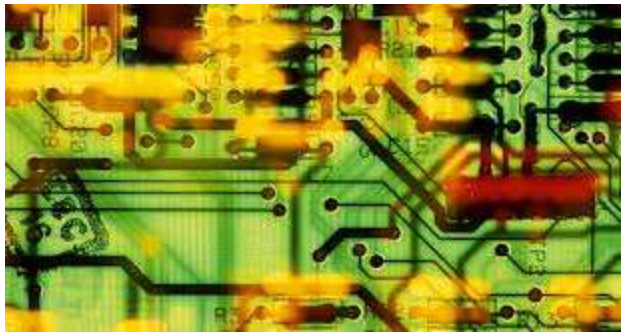
Attempts to Make Multicore Programming Easy

- 3rd idea: Write software that will automatically parallelize existing sequential programs.
  - **Result so far**: Success here is inversely proportional to the number of cores!

The Multicore Software Triad

Programming Model





The Real Hardware

Conclusions

- The free lunch is over.
- Mulicore/Manycore processors are here to stay, so we have to deal with them.
- Knowing about the hardware will make you way more efficient in software!