EBERHARD KARLS UNIVERSITÄT TÜBINGEN

FACULTY OF SCIENCE

Communication Networks

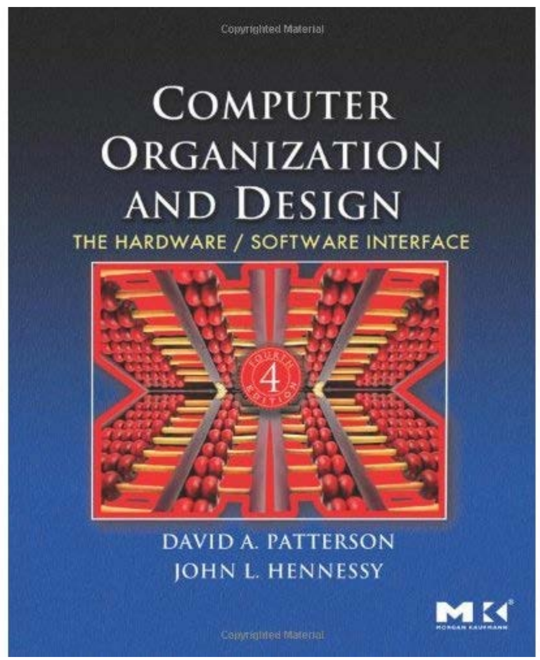
# Informatik der Systeme – Chapter 4: Computer Abstractions and Technology

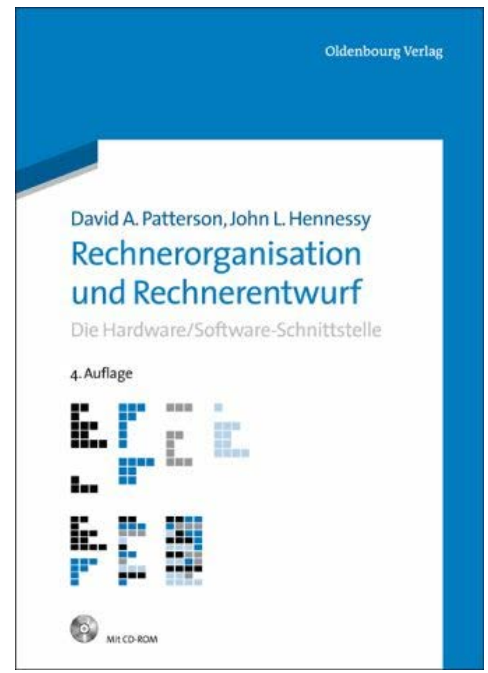
Prof. Dr. Michael Menth

*http://kn.inf.uni-tuebingen.de*

# Acknowledgements

- This set of slides is an adaptation of Prof. Mary Jane Irwin's lecture notes, http://www.cse.psu.edu/research/mdl/mji/
- Adapted from Patterson & Hennessy: "Computer Organization and Design", 4th Edition, © 2008, MK
- German translation: Patterson & Hennessy: "Rechnerorganisation und Rechnerentwurf"



# Overview

- Application binary interface
- Components of a computer
- Classes of computers
- Technological evolution
- Performance considerations

# Overview

- Application binary interface
- Components of a computer
- Classes of computers
- Technological evolution
- Performance considerations

# Layers of Software

- High-level language program (in C)

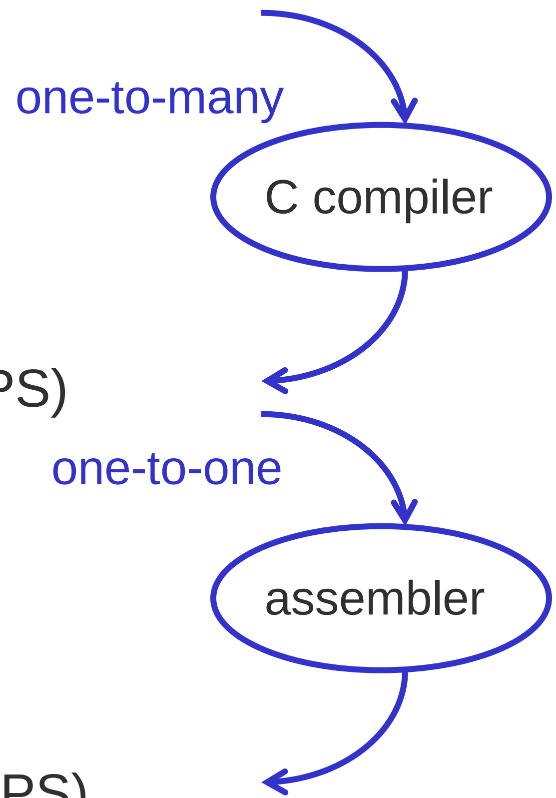
```
swap (int v[], int k)
(int temp;
      temp = v[k];
      v[k] = v[k+1];
      v[k+1] = temp;
)
```

- Assembly language program (for MIPS)

```
swap:  sll    $2, $5, 2
       add    $2, $4, $2
       lw     $15, 0($2)
       lw     $16, 4($2)
       sw     $16, 0($2)
       sw     $15, 4($2)
       jr     $31
```

- Machine (object, binary) code (for MIPS)

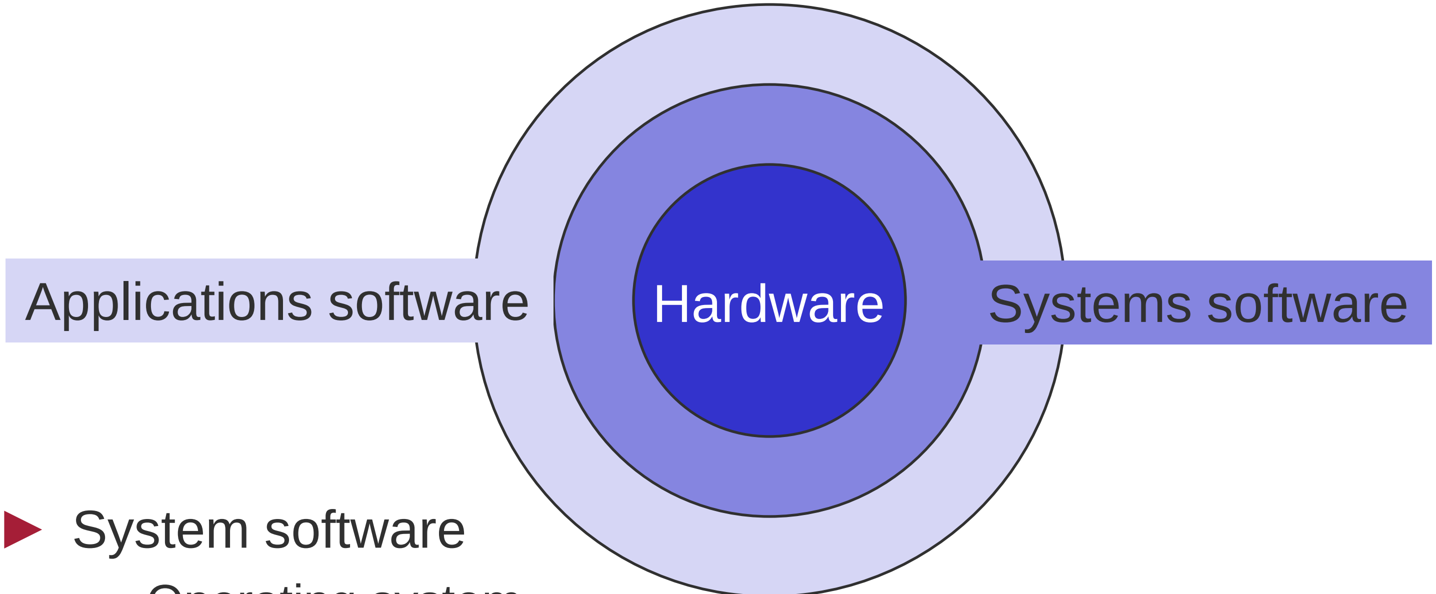
```
000000 00000 00101 0001000010000000
000000 00100 00010 0001000000100000
. . .
```



# Advantages of Higher-Level Languages

- Higher-level languages
  - Allow the programmer to think in a more natural language and for their intended use (Fortran for scientific computation, Cobol for business programming, Lisp for symbol manipulation, Java for web programming, …)
  - Improve programmer productivity – more understandable code that is easier to debug and validate
  - Improve program maintainability
  - Allow programs to be independent of the computer on which they are developed
    - Compilers and assemblers can translate high-level language programs to the binary instructions of any machine
  - Compilers produce very efficient assembly code optimized for the target machine
- As a result, very little programming is done today at the assembler level

# System Software




- System software
  - Operating system
    - Supervising program that interfaces the user's program with the hardware (e.g., Linux, MacOS, Windows)
    - Handles basic input and output operations
    - Allocates storage and memory
    - Provides for protected sharing among multiple applications
  - Compiler
    - Translates programs written in a high-level language (e.g., C, Java) into instructions that the hardware can execute

# ISA and ABI

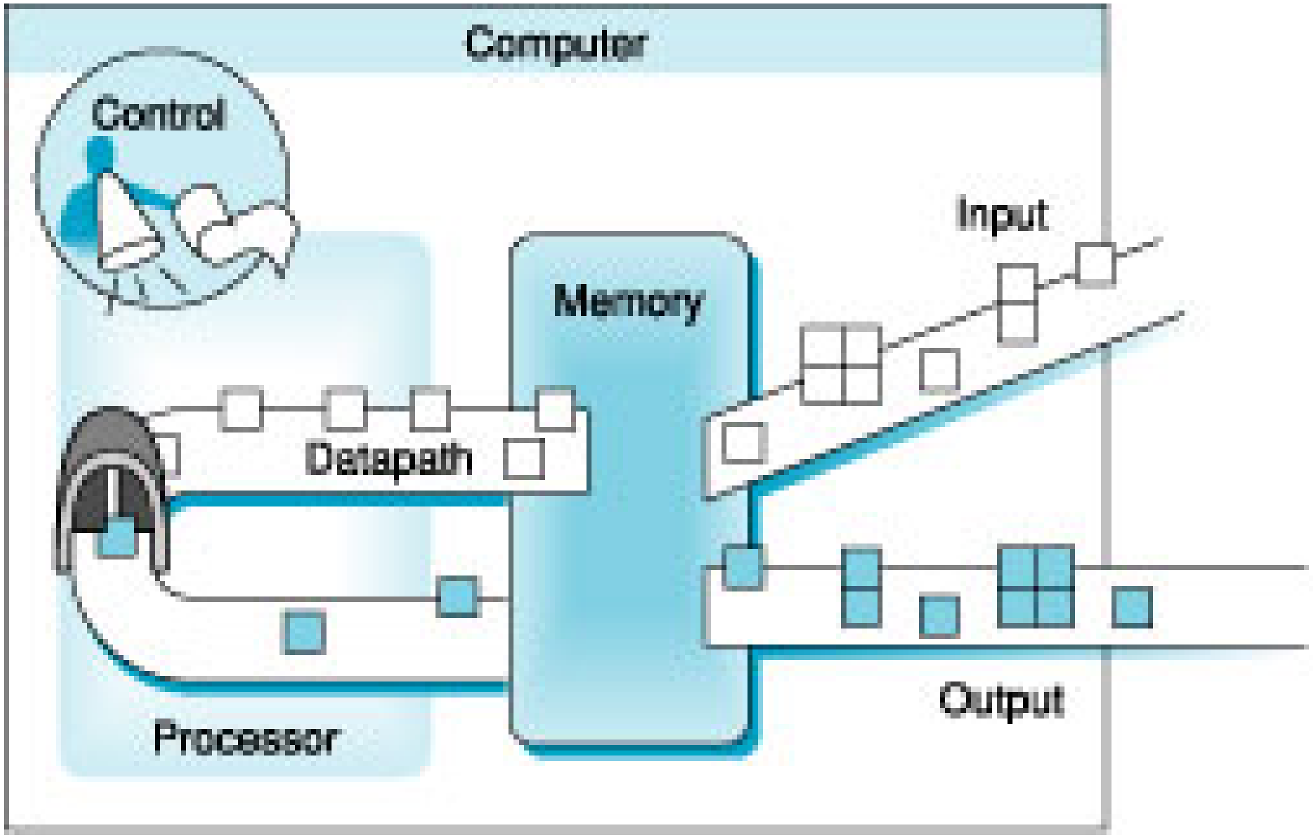
- Instruction set architecture (ISA)
  - Abstract interface between the hardware and the lowest level software (aka basic instruction set)
  - Encompasses all the information necessary to write a machine language program, including instructions, registers, memory access, I/O, …
  - Enables implementations of varying cost and performance to run identical software
- Application binary interface (ABI)
  - Combination of the user portion of the instruction set and the operating system interface
  - Used by application programmers
  - Defines a standard for binary portability across computers

# Overview

- Application binary interface
- Components of a computer
- Classes of computers
- Technological evolution
- Performance considerations

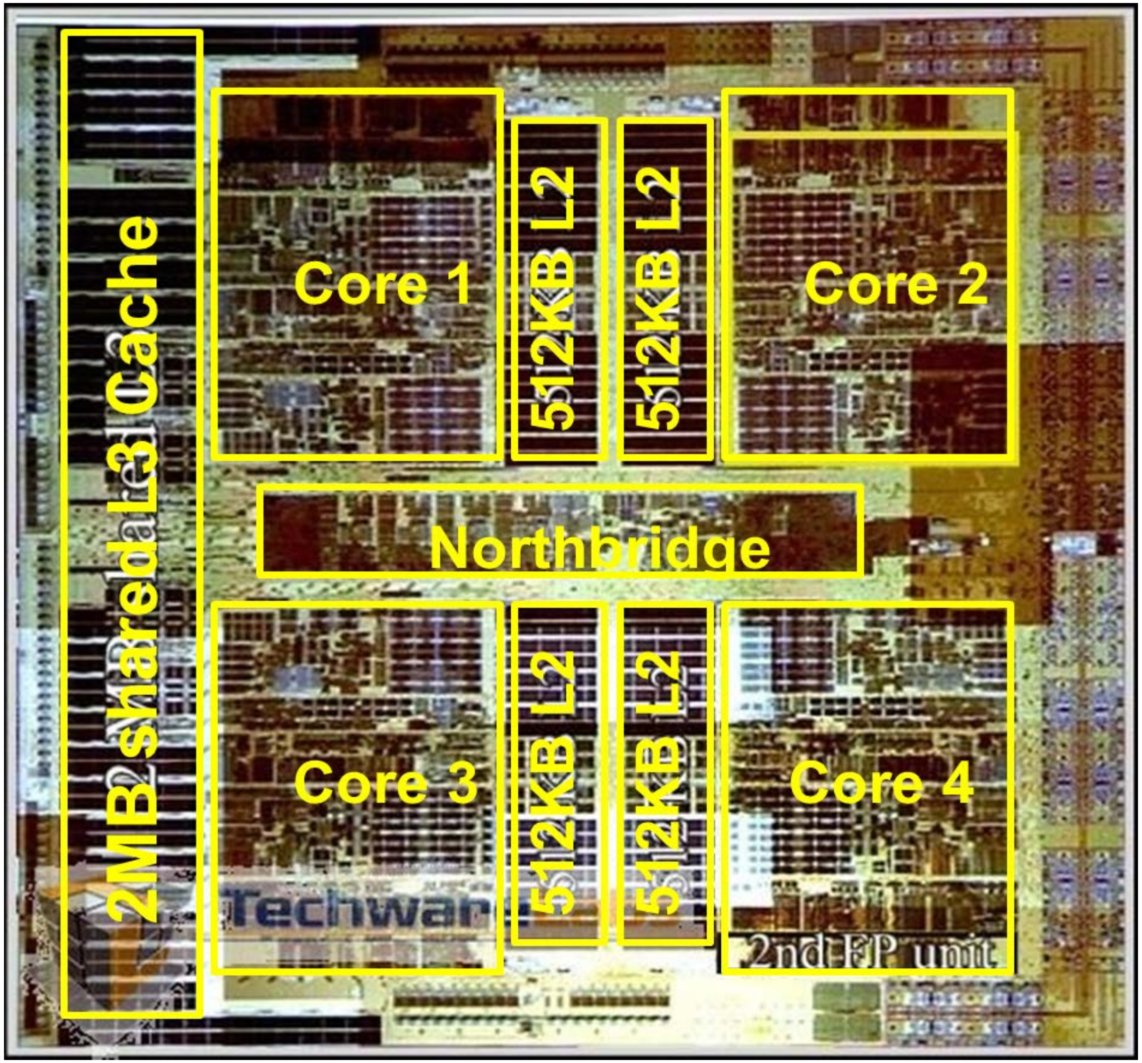
# Components of a Computer

- Five classic components of a computer
  - Input
  - Output
  - Memory
  - Datapath
  - Control




- Processor (CPU)
  - Control
  - Datapath

# AMD's Barcelona Multicore Chip




http://www.techwarelabs.com/reviews/processors/barcelona/

- Four cores on one chip
- 1.9 GHz clock rate
- 65 nm technology
- Three levels of caches (L1, L2, L3) on chip
- Integrated Northbridge

# Overview

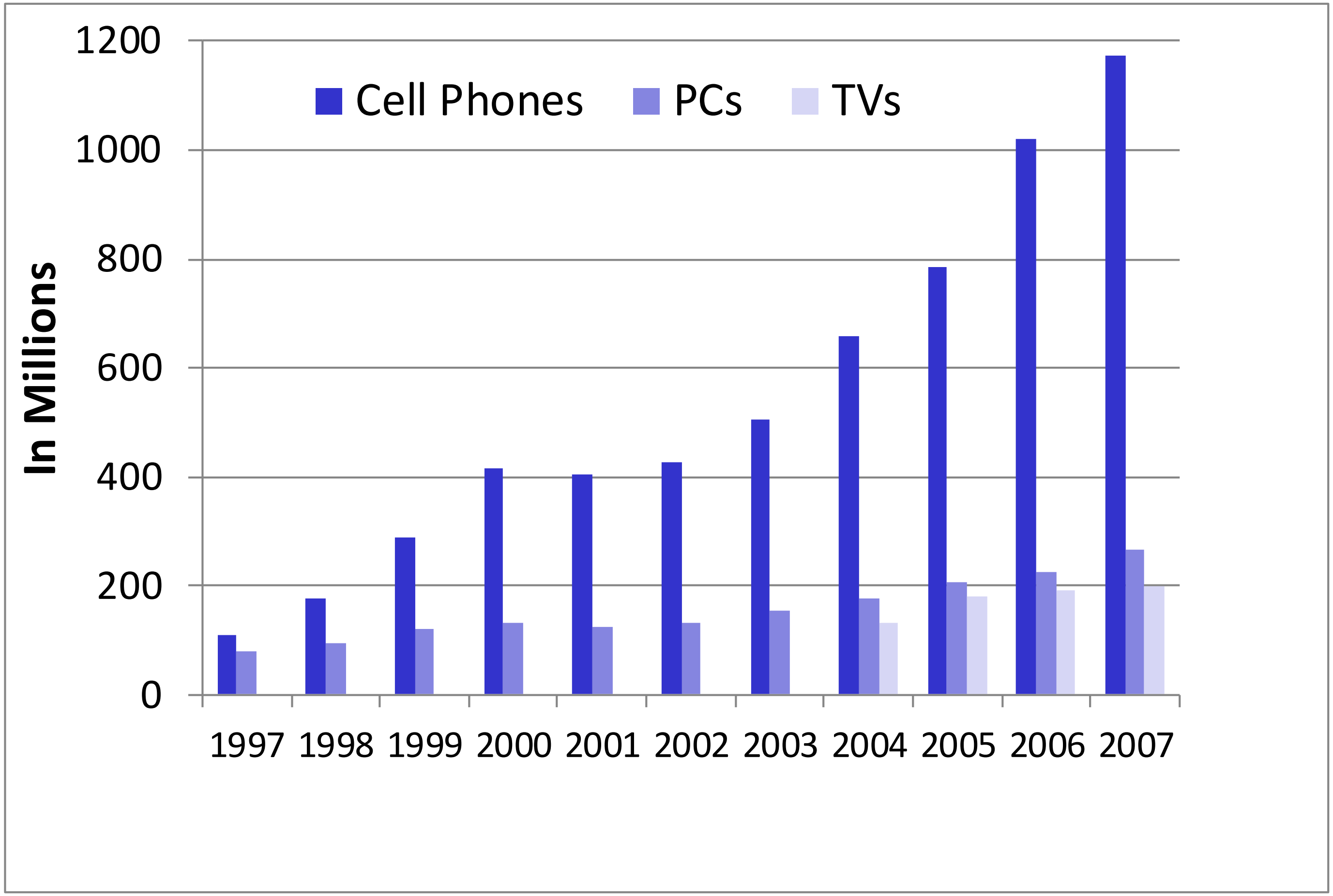
- Application binary interface
- Components of a computer
- Classes of computers
- Technological evolution
- Performance considerations

# Classes of Computers

- Desktop computers
  - Designed to deliver good performance to a single user at low cost usually executing 3rd party software, usually incorporating a graphics display, a keyboard, and a mouse
- Servers
  - Used to run larger programs for multiple, simultaneous users typically accessed only via a network and that places a greater emphasis on dependability and (often) security
- Supercomputers
  - A high performance, high cost class of servers with hundreds to thousands of processors, terabytes of memory and petabytes of storage that are used for high-end scientific and engineering applications
- Embedded computers (processors)
  - A computer inside another device used for running one predetermined application

# Growth in Cell Phone Sales (Embedded)

embedded growth >> desktop growth



# Embedded Processors

- The largest class of computers
- Span the widest range of applications and performance
- Often have
  - Minimum performance requirements
  - Stringent cost limitations
  - Stringent power consumption limitations
  - Low tolerance for failure

# Overview

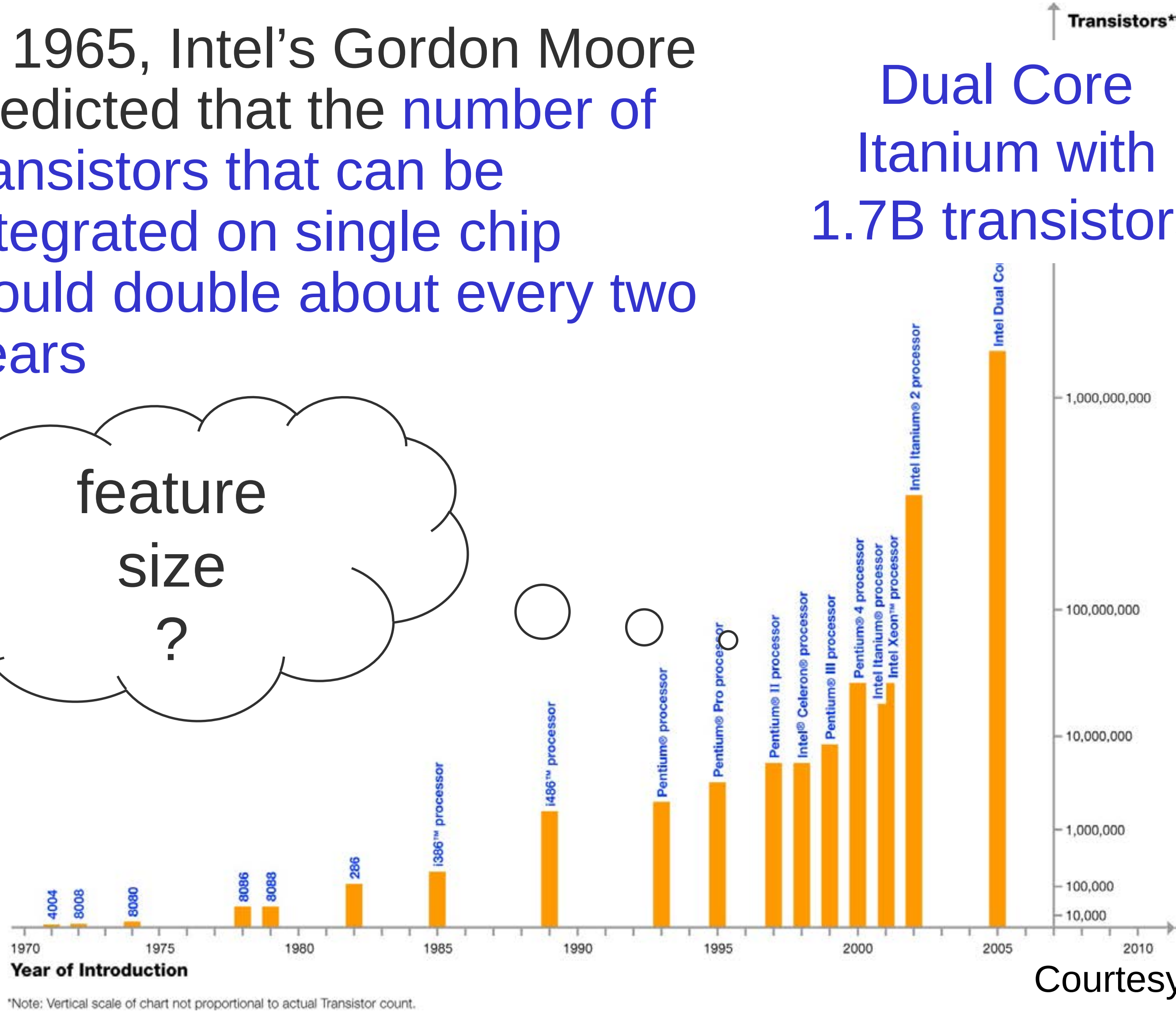
- Application binary interface
- Components of a computer
- Classes of computers
- Technological evolution
- Performance considerations

# Moore’s Law

- In 1965, Intel’s Gordon Moore predicted that the number of transistors that can be integrated on single chip would double about every two years

Dual Core Itanium with 1.7B transistors

feature size ?




Log Scale

Courtesy, Intel ®

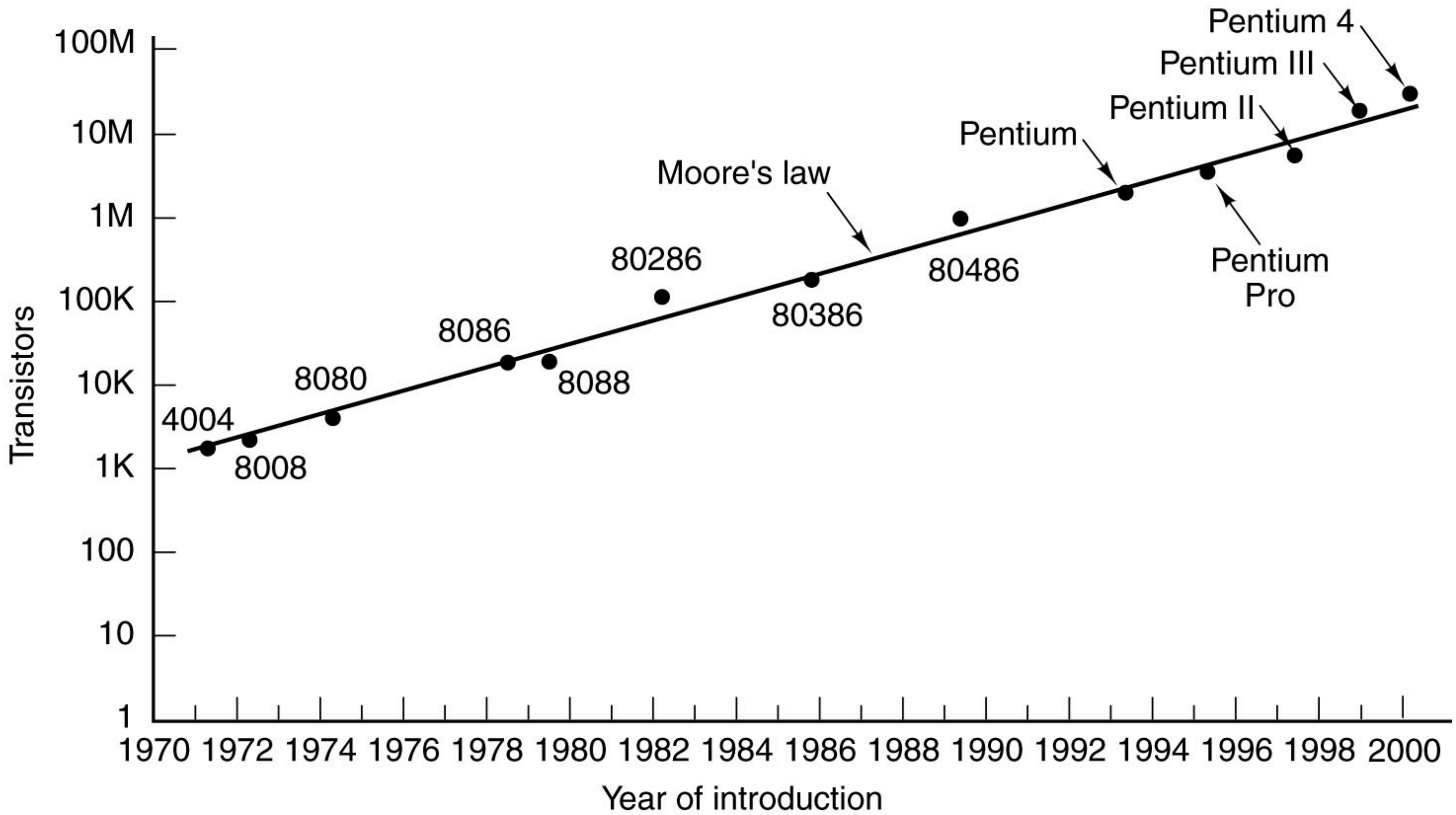
EBERHARD KARLS UNIVERSITÄT TÜBINGEN

# Technology Scaling Road Map (ITRS)

| Year | 2004 | 2006 | 2008 | 2010 | 2012 |
|---|---|---|---|---|---|
| Feature size (nm) | 90 | 65 | 45 | 32 | 22 |

- Fun facts about 45 nm transistors
  - 30 million can fit on the head of a pin
  - You could fit more than 2,000 across the width of a human hair
  - If car prices had fallen at the same rate as the price of a single transistor has since 1968, a new car today would cost about 1 cent

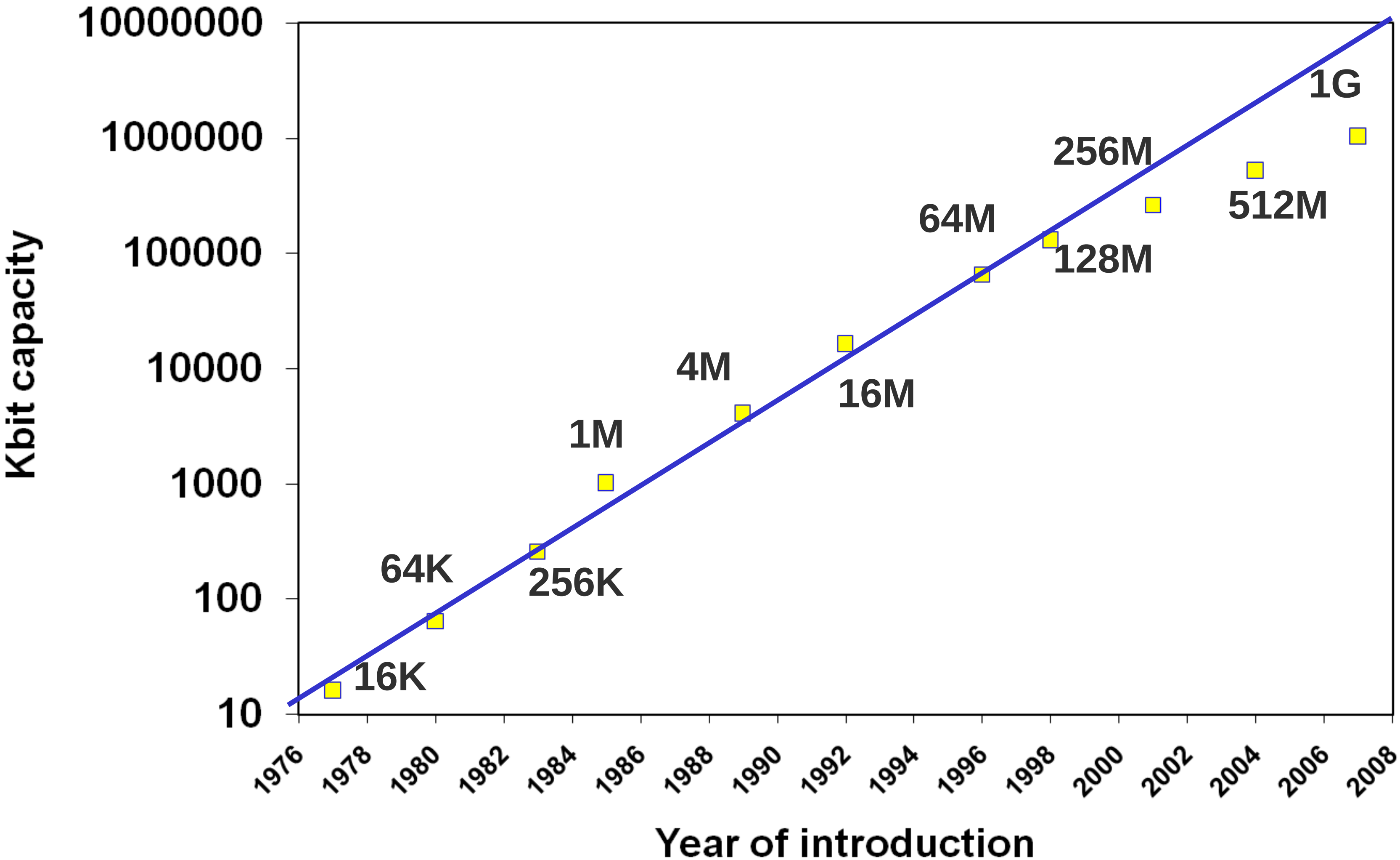
# Moore’s Law for (Intel) CPU Chips




Source: A. Tanenbaum, Structured Computer Organization, 5/E, © Pearson

# Another Example of Moore's Law Impact

DRAM capacity growth over 3 decades

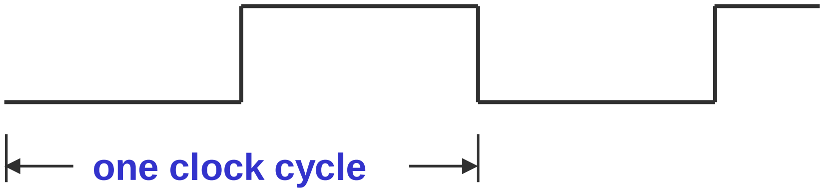


# Overview

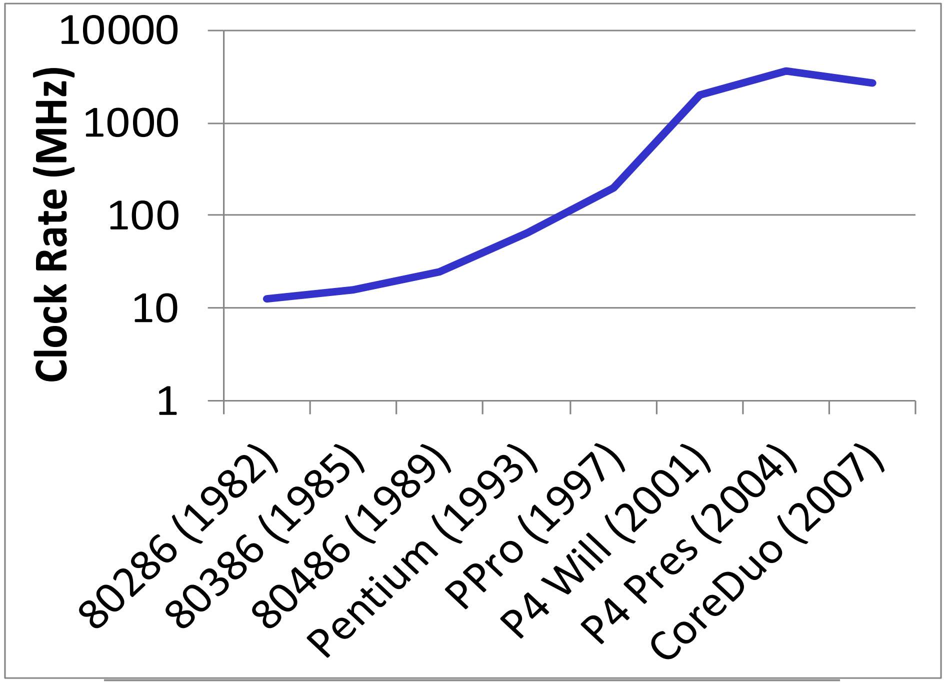
- Application binary interface
- Components of a computer
- Classes of computers
- Technological evolution
- Performance considerations

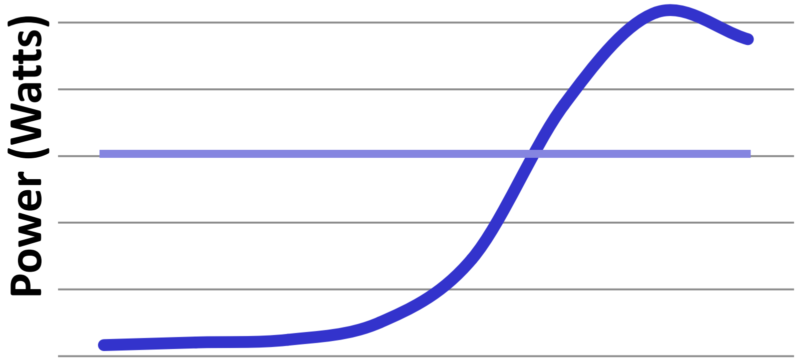
# Machine Clock Rate

- Machines proceed in clock cycles (CCs)
  - Given in ns, ps




- Clock rate = 1 / CC
  - Given in MHz, GHz
- Clock rates hit a “power wall”
  - Order of GHz not exceeded for several years for energy reasons



# A Sea Change is at Hand

| Product | AMD Barcelona | Intel Nehalem | IBM Power 6 | Sun Niagara 2 |
|---|---|---|---|---|
| Cores per chip | 4 | 4 | 2 | 8 |
| Clock rate | 2.5 GHz | ~2.5 GHz? | 4.7 GHz | 1.4 GHz |
| Power | 120 W | ~100 W? | ~100 W? | 94 W |

- The power challenge has forced a change in the design of microprocessors
  - Since 2002 the rate of improvement in the response time of programs on desktop computers has slowed from a factor of 1.5 per year to less than a factor of 1.2 per year
- As of 2006 all desktop and server companies are shipping microprocessors with multiple processors – cores – per chip
- Plan of record is to double the number of cores per chip per generation (about every two years)

# Clock Cycles per Instruction (CPI)

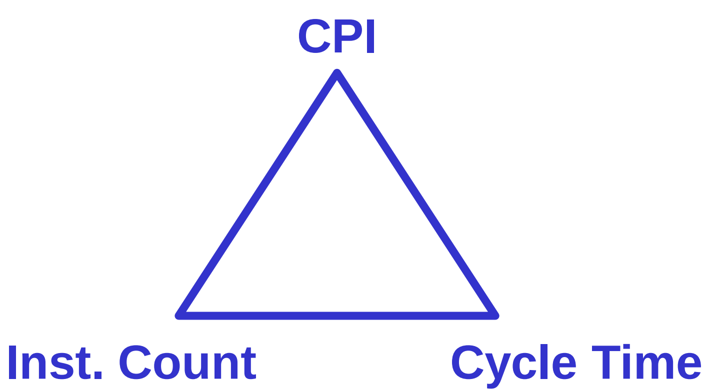
- Instruction set (IS)
  - Same IS may be supported by different machines
  - Performance may vary

- Clock cycles per instruction (CPI)
  - Depends on how IS is implemented on a machine
  - Complex instructions have large CPI
    - CPI is instruction-specific
    - Define instruction classes with similar CPI

| | Instruction class | | |
|---|---|---|---|
| | A | B | C |
| CPI (machine 1) | 1 | 2 | 3 |
| CPI (machine 2) | 2 | 3 | 5 |

# CPU Time

- Given: programm (instructions)
  - $IC$: instruction count
  - $IC_i$: instruction count per CPI class $i$
- Instruction mix
  - Influences $IC_i$
  - Depends on program
- Effective CPI
  - $n$ : number of CPI classes
  - $CPI_{eff} = \frac{\sum_{i=1}^{n} CPI_i \cdot IC_i}{IC}$
- CPU time
  - $t_{CPU} = \frac{\sum_{i=1}^{n} CPI_i \cdot IC_i}{clock\ rate}$
  - Three factors influenceing CPU time



# Factors for Instruction Count, $CPI_{eff}$, Clock Rate

| | Instruction count | $CPI_{eff}$ | Clock rate |
|---|---|---|---|
| Algorithm | **X** | **X** | |
| Programming language | **X** | **X** | |
| Compiler | **X** | **X** | |
| ISA | **X** | **X** | **X** |
| Core organization | | **X** | **X** |
| Technology | | | **X** |

# Performance

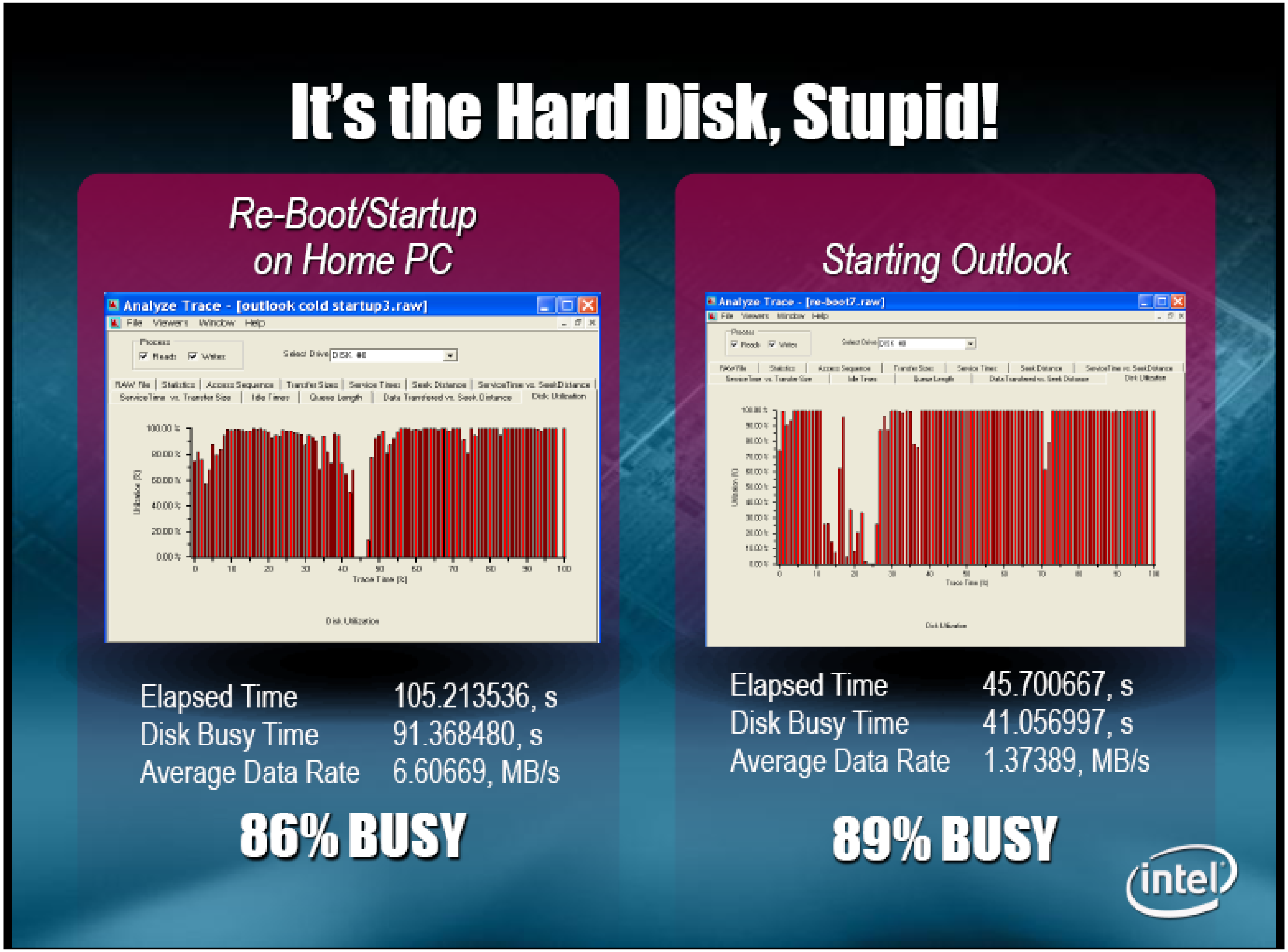
- Execution speed (CPU time), does not reflect idle times
  - Time to wait for I/O
  - Time to wait for CPU access on multiprocessor machines
  - Etc.

- Still we define the performance of machine $X$ for program $P$ as
  - $Performance(X,P) = \frac{1}{t_{CPU}(X,P)}$
  - Faster is better
  - Comparison of two machines $X$ and $Y$ for specific program $P$
    - $\frac{Performance(X,P)}{Performance(Y,P)} = \frac{t_{CPU}(Y,P)}{t_{CPU}(X,P)} = n$ means
      - Performance of $X$ is n times larger than the one of $Y$
      - $X$ is n times faster than $Y$

# Workloads and Benchmarks

- Benchmark
  - A set of programs that form a “workload” specifically chosen to measure performance
- SPEC (System Performance Evaluation Cooperative)
  - Creates standard sets of benchmarks
  - www.spec.org

# Response Time Matters




Justin Rattner's ISCA'08 Keynote (VP and CTO of Intel)

# Other Performance Metrics than CPU Time

- Response time (overall execution time)
  - Time between the start and the completion of a task
  - Important to individual users
- Throughput
  - Total amount of work done in a given time
  - Important to data center managers
- Energy efficiency
  - Power consumption especially important for power-limited applications, e.g., in the embedded market where battery life matters