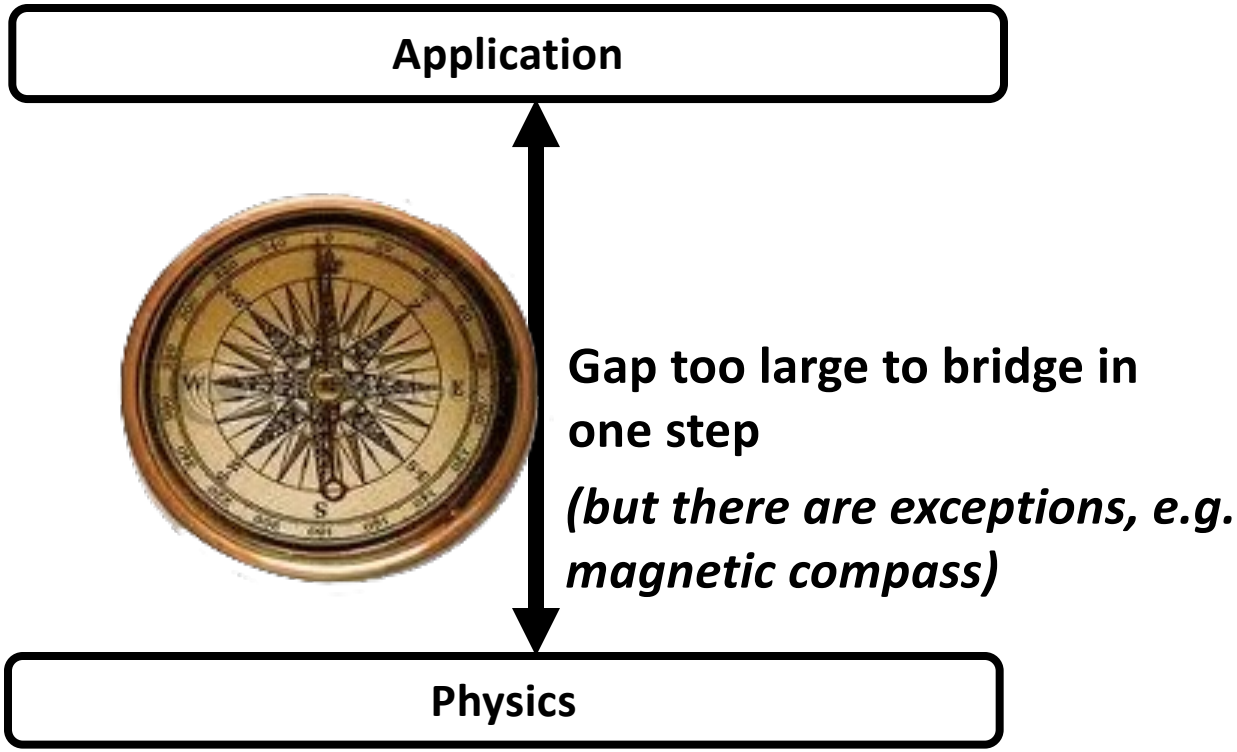
Introduction

Lecture 1
September 4th, 2023

Jae W. Lee (jaewlee@snu.ac.kr)
Computer Science and Engineering
Seoul National University

***Slide credits**: Instructor's slides from Elsevier Inc. and UC Berkeley CS 152*

What is Computer Architecture?



In its broadest definition, computer architecture is the *design of the abstraction layers* that allow us to implement information processing applications efficiently using available manufacturing technologies.

Abstraction Layers in Modern Systems

- Application
- Algorithm
- Programming Language
- Operating System/Virtual Machines
- Instruction Set Architecture (ISA)
- Microarchitecture
- Gates/Register-Transfer Level (RTL)
- Circuits
- Devices
- Physics

system programming / OS

Main topics of this course

Logic design

The End of the Uniprocessor Era

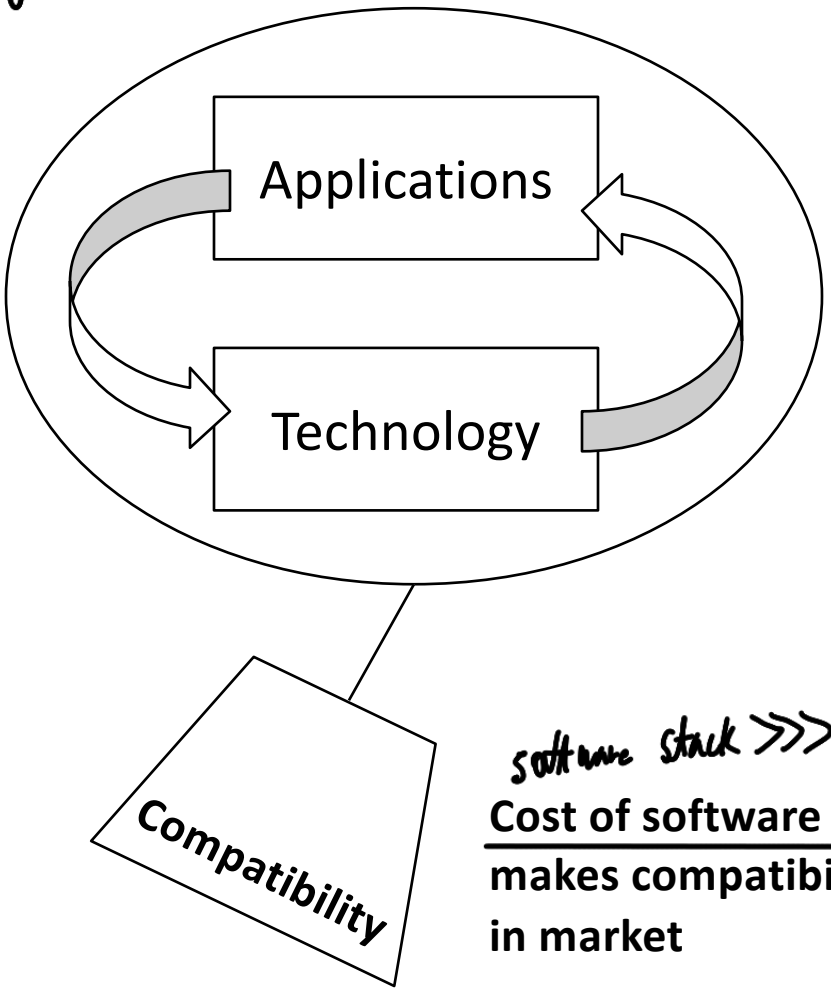
Single biggest change in the history of computing systems

Single to multi core

Architecture continually changing

interface between hardware and software

Applications suggest how to improve technology, provide revenue to fund development

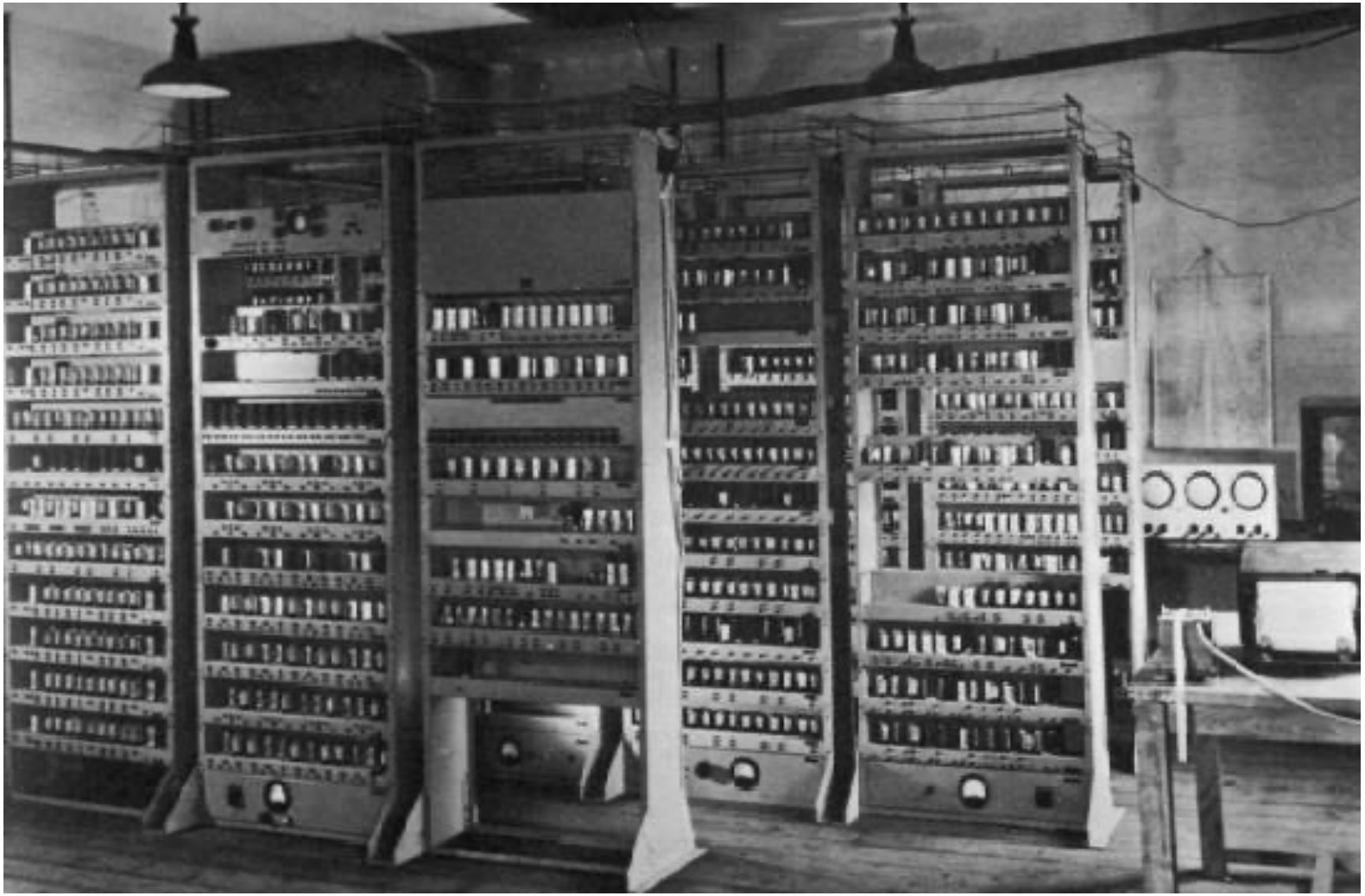


Improved technologies make new applications possible

software stack >>> hardware investment

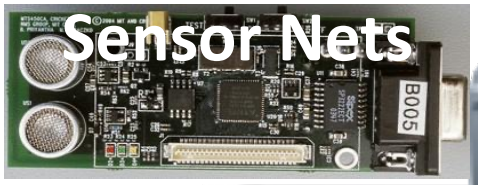
Cost of software development makes compatibility a major force in market

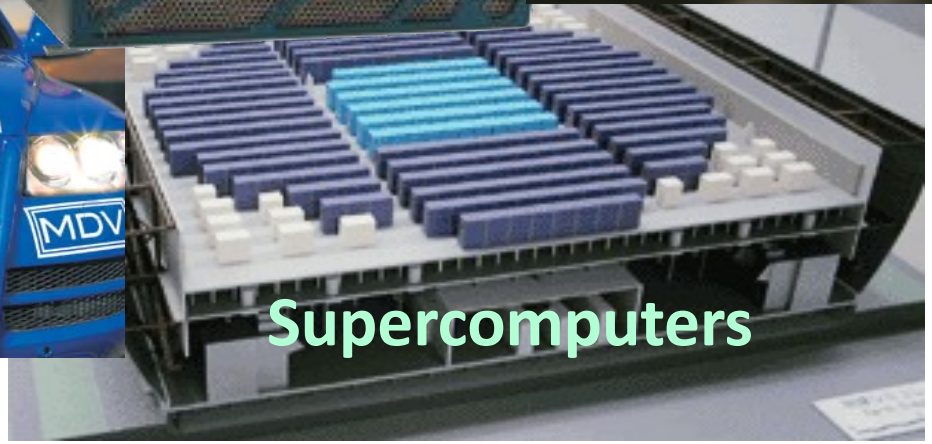
Computing Devices Then…



EDSAC, University of Cambridge, UK, 1949

Computing Devices Now

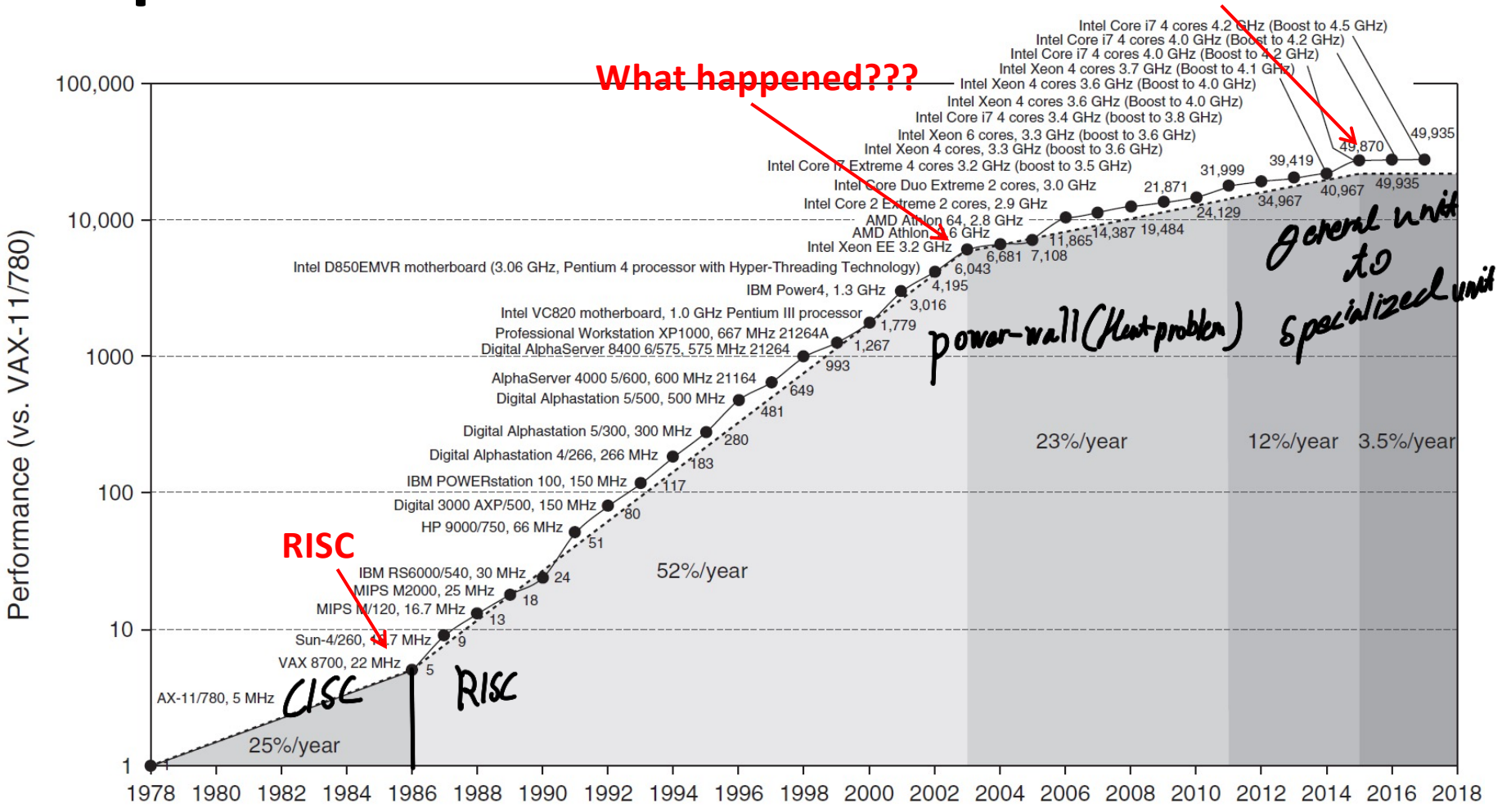




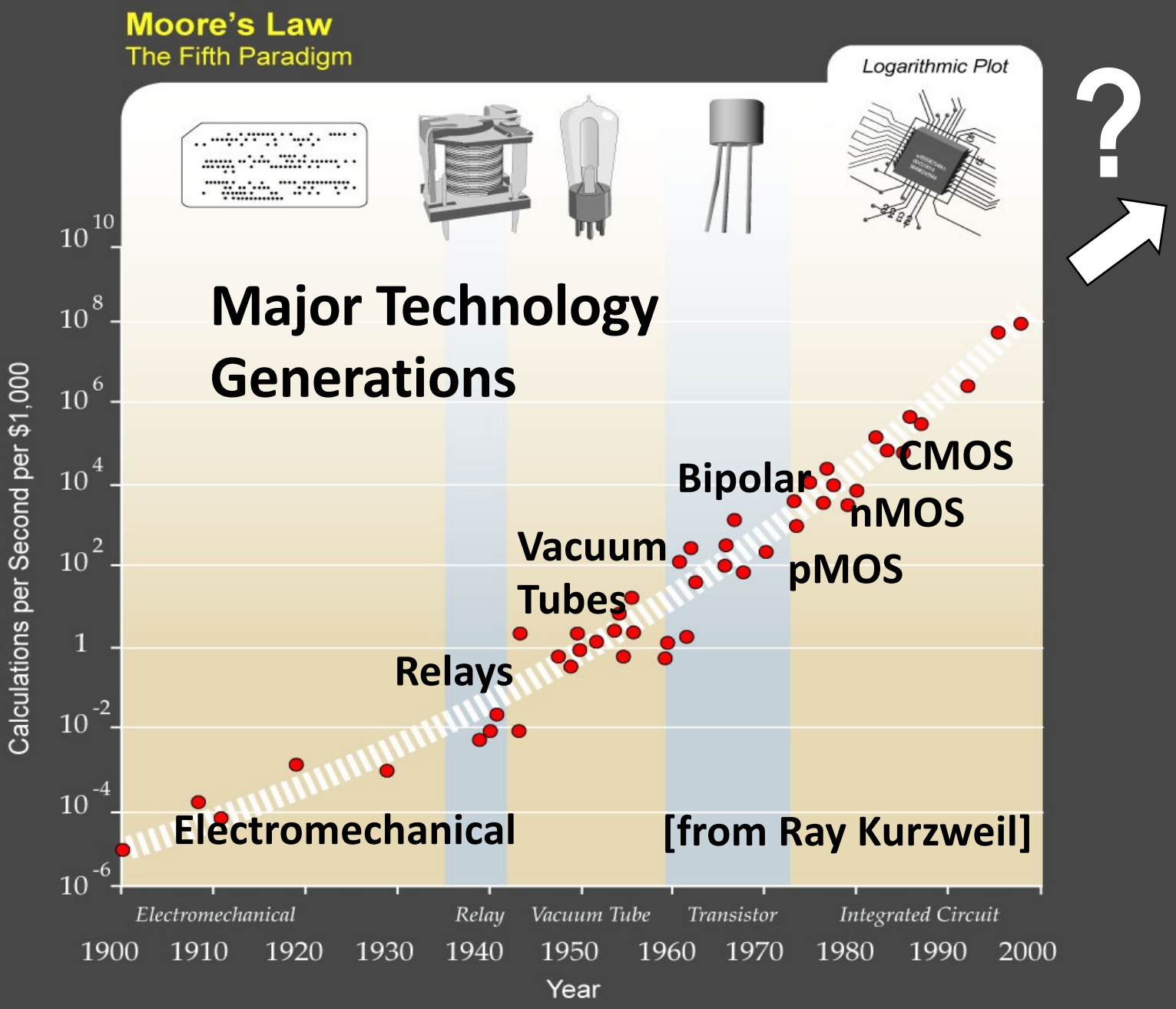
(Single-processor)

Uniprocessor Performance

Constrained by instruction-level parallelism, power, memory latency



- VAX : 25%/year 1978 to 1986
- RISC + x86 : 52%/year 1986 to 2002
- RISC + x86 : 23%/year 2002 to 2021


Moore's Law
The Fifth Paradigm
Logarithmic Plot
?
Major Technology Generations
CMOS
Bipolar
nMOS
Vacuum Tubes
pMOS
Relays
Electromechanical
[from Ray Kurzweil]
Calculations per Second per $1,000
10^10
10^8
10^6
10^4
10^2
1
10^-2
10^-4
10^-6
Electromechanical
Relay
Vacuum Tube
Transistor
Integrated Circuit
1900
1910
1920
1930
1940
1950
1960
1970
1980
1990
2000
Year

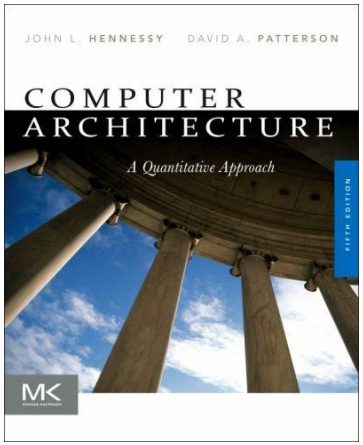
This Course

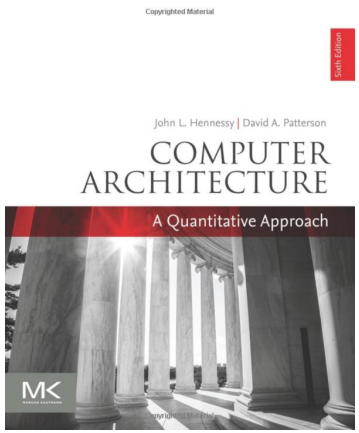
- **This course focuses on interaction of software and hardware**
  - More architecture and less digital engineering
  - More useful for OS developers, compiler writers, performance programmers

Textbook

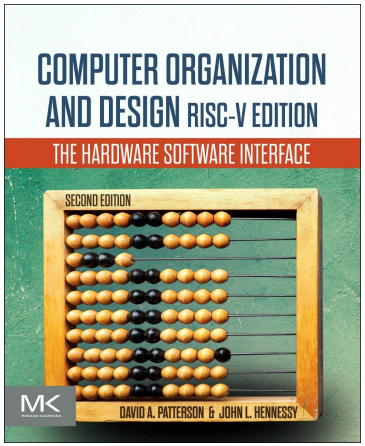
- We will faithfully cover the Hennessy and Patterson book.
- We assume you have already taken undergraduate computer architecture with Patterson and Hennessy book.

5th ed. 6th ed.





Main textbook (graduate level)
Hennessy & Patterson
(Either 5th or 6th Edition is fine)



Reference book (undergraduate level)
Patterson & Hennessy

Warning

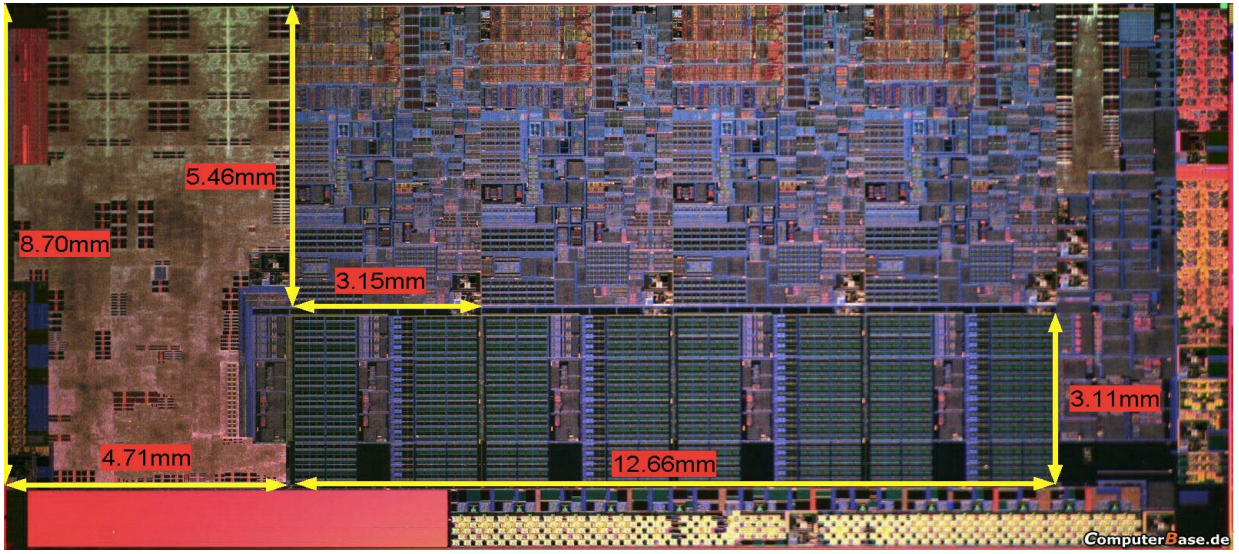
- *A*
- I
  c

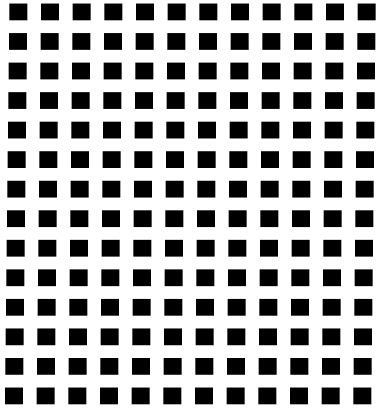

WARNING
YOU ARE NOT AUTHORIZED
TO VIEW THE REST OF THIS CONTENT

Course Executive Summary

The processor you built in undergrad computer architecture (MIPS 5-stage pipeline)

What you'll understand and experiment with in this course





Plus, the technology behind chip-scale multiprocessors (CMPs) and graphics processing units (GPUs)

Course Information

Instructor: Prof. Jae W. Lee, `jaewlee@snu.ac.kr`
Office: Engineering Building #301-506 or Zoom
Office Hours: Mon 2-3 PM or by appointment (Zoom)

Lectures: 11 AM – 12:15 PM (Mon/Wed) @ #301-101

Text: *Computer Architecture: A Quantitative Approach, Hennessey and Patterson, 5th Edition* (2012) / 6th Edition (2017)

Web page: All course materials will be distributed and announcements will be made through eTL (please check frequently).

Structure and Syllabus*

Schedule	Contents	Remarks
Weeks 1-4	Module 1: Advanced Processor Design (ISAs, unpipelined machines, Iron Law, simple pipelines, complex pipelining (score-boarding, out-of-order issue), speculative execution)	Chapters 1, 3, Appendices A, C
Weeks 5-7	Module 2: Memory hierarchy (DRAM, caches, optimizations, virtual memory systems, exceptions, interrupts)	Chapter 2 Appendix B
Week 8	**Midterm Exam.**	
Week 9-10	Module 3: Explicit parallel multiprocessors (Multicore processors, Multi-threading, Vector, VLIW, GPU)	Chapter 4
Weeks 11-13	Module 4: Multiprocessor architectures (memory models, cache coherence, synchronization)	Chapter 5
Weeks 14-15	Module 5: Domain-specific architectures	
Week 16	**Final Exam.**	

**** Schedule is subject to change.***

Course Components

- **10% Class Participation**
  - Mostly attendance but may give extra credits to active class participation
- **20% Paper Summary Assignments**
  - Assigned ~6 papers to read and summarize in 3 pages or less
  - Pass or fail – all or nothing
  - Can be asked about in midterm/final
- **30% Midterm**
- **40% Final**

fine in Korean

Overview

Textbook: Chapter 1

- **Classes of computers**
- **Trends in technology**
- **Measuring performance**
- **Principles of computer design**

Computer Technology

- **Performance improvements:**
  - Improvements in semiconductor technology (chip, gate)
    - Feature size, clock speed
  - Improvements in computer architectures CISC→RISC, pipeline(design)
    - Enabled by HLL compilers, UNIX
    - Lead to RISC architectures

  - Together have enabled:
    - Lightweight computers
    - Productivity-based managed/interpreted programming languages

Current Trends in Architecture

- **Cannot continue to leverage Instruction-Level parallelism (ILP)** — reduce CPI (run more instruction together)
  - Single processor performance improvement ended in 2003
    ⇒ n-stage pipelining / high dependency on application (limit exist)
- **New models for performance:**
  - Data-level parallelism (DLP) — Handle multiple data at a time (GPU, server)
  - Thread-level parallelism (TLP) - Multicore programming (map instruction, divide as task)
  - Request-level parallelism (RLP)
- **These require explicit restructuring of the application**

Classes of Computers: Scales

- **Personal Mobile Device (PMD)**
  - e.g. start phones, tablet computers
  - Emphasis on energy efficiency and real-time
- **Desktop Computing**
  - Emphasis on price-performance
- **Servers**
  - Emphasis on availability, scalability, throughput (Batching)
- **Clusters / Warehouse Scale Computers**
  - Used for "Software as a Service (SaaS)"
  - Emphasis on availability and price-performance
  - Sub-class: Supercomputers, emphasis: floating-point performance and fast internal networks
- **Embedded Computers**
  - Emphasis: price

Classes of Computers: Parallelism

- **Two kinds of parallelism in applications:** Soft
  - Data-Level Parallelism (DLP)
  - Task-Level Parallelism (TLP)

- **Computer hardware exploits parallelism as follows:** hard
  - Instruction-Level Parallelism (ILP)
  - Vector architectures/Graphic Processor Units (GPUs)
  - Thread-Level Parallelism
  - Request-Level Parallelism

Classes of Computers: Flynn's Taxonomy

- **Single instruction stream, single data stream (SISD)**
- **Single instruction stream, multiple data streams (SIMD)**
  - Vector architectures
  - Multimedia extensions
  - Graphics processor units

⇒ Inter i128 add

- **Multiple instruction streams, single data stream (MISD)**
  - No commercial implementation

Not reasonable

- **Multiple instruction streams, multiple data streams (MIMD)**
  - Tightly-coupled MIMD
  - Loosely-coupled MIMD

Defining Computer Architecture

- **"Old" (narrow) view of computer architecture:**
  - Instruction Set Architecture (ISA) design
  - i.e. decisions regarding:
    - registers, memory addressing, addressing modes, instruction operands, available operations, control flow instructions, instruction encoding

- **"Real" (broad view of) computer architecture:**
  - Specific requirements of the target machine
  - Design to maximize performance within constraints: cost, power, and availability
  - Includes ISA, microarchitecture, hardware

Trends in Technology: An Overview

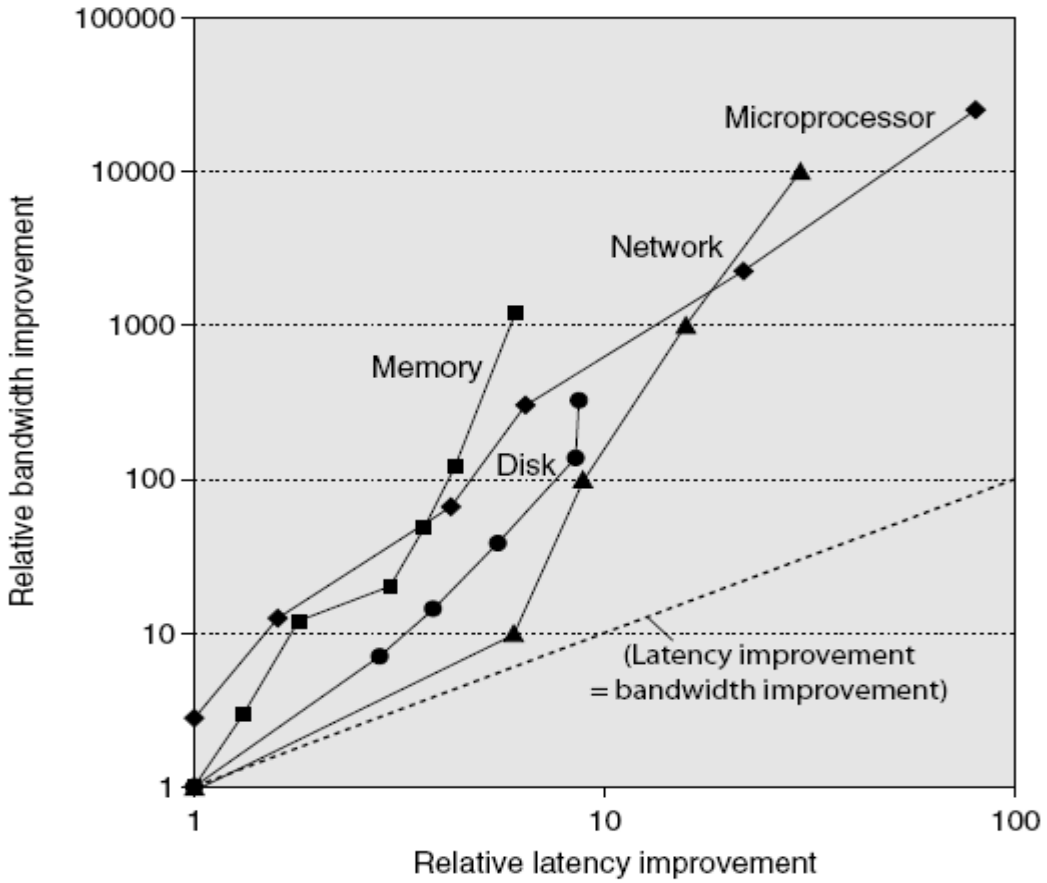
- **Integrated circuit technology**
  - Transistor density: 35%/year
  - Die size: 10-20%/year
  - Integration overall: 40-55%/year
- **DRAM capacity: 25-40%/year (slowing)**
- **Flash capacity: 50-60%/year** SSD
  - 15-20X cheaper/bit than DRAM
- **Magnetic disk technology: 40%/year** HDD
  - 15-25X cheaper/bit than Flash
  - 300-500X cheaper/bit than DRAM

Trends in Technology: Bandwidth and Latency

- **Bandwidth or throughput**
  - Total work done in a given time
  - 10,000-25,000X improvement for processors
  - 300-1200X improvement for memory and disks

- **Latency or response time** : much more difficult
  - Time between start and completion of an event
  - 30-80X improvement for processors
  - 6-8X improvement for memory and disks

Trends in Technology: Bandwidth and Latency (Cont)



Log-log plot of bandwidth and latency milestones

Trends in Technology: Transistors and Wires

- **Feature size**
  - Minimum size of transistor or wire in x or y dimension
  - 10 microns in 1971 to .005 microns in 2021
  - Transistor performance scales linearly
    - Wire delay does not improve with feature size!
  - Integration density scales quadratically

Trends in Technology: Power and Energy

- **Problem: Get power in, get power out** (+ Heat)
- **Thermal Design Power (TDP)**
  - Characterizes sustained power consumption (max power)
  - Used as target for power supply and cooling system
  - Lower than peak power, higher than average power consumption
- **Clock rate can be reduced dynamically to limit power consumption**
- **Energy per task is often a better measurement**

Trends in Technology: Dynamic Energy and Power

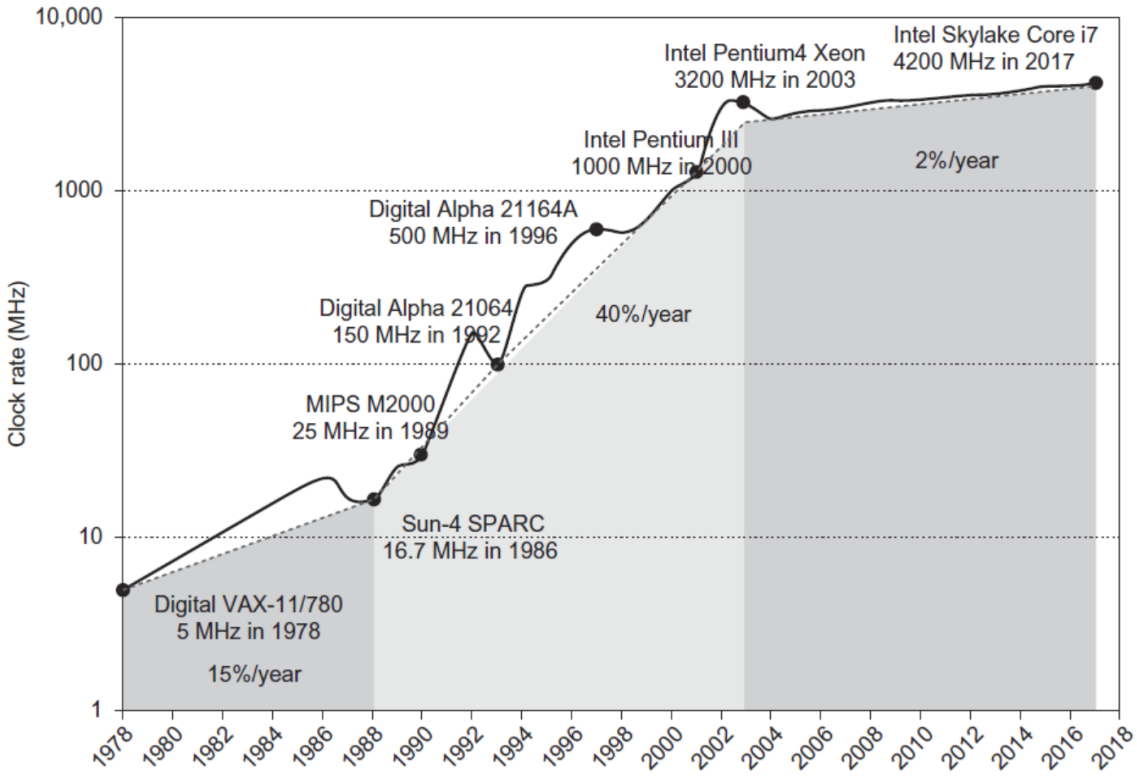
- **Dynamic energy**
  - Transistor switch from 0 -> 1 or 1 -> 0
  - ½ x Capacitive load x Voltage^2
- **Dynamic power**
  - ½ x Capacitive load x Voltage^2 x Frequency switched

increase

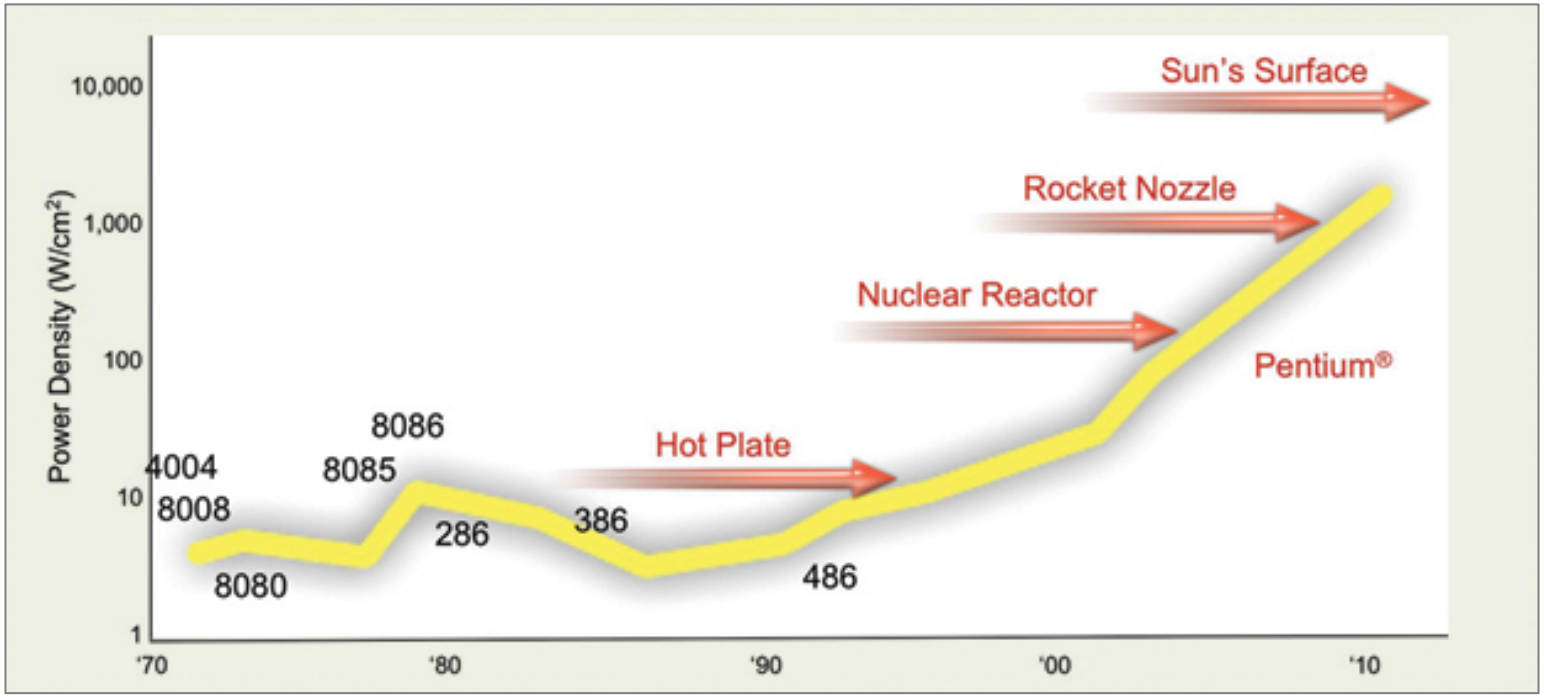
- **Reducing clock rate reduces power, not energy**

Trends in Technology: Power

- Intel 80386 consumed ~ 2 W
- 3.3 GHz Intel Core i7 consumes 130 W
- Heat must be dissipated from 1.5 x 1.5 cm chip
- This is the limit of what can be cooled by air



Trends in Technology: Power Density

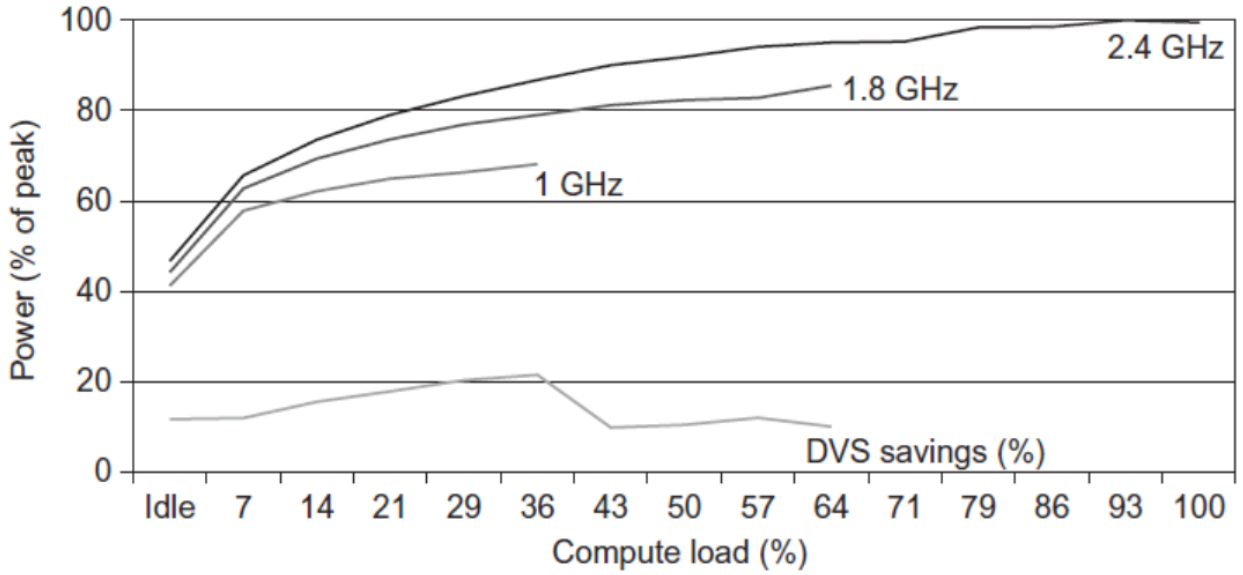


Courtesy of Pat Gelsinger, Intel Developer Forum, Spring 2004.

Trends in Technology: Reducing Power

- **Techniques for reducing power:**
  - "Do nothing" well (i.e., minimize waste)
  - Dynamic Voltage-Frequency Scaling (DVFS)
  - Low power state for DRAM, disks
  - Overclocking, turning off cores

Memory is passive element
40% of total



Trends in Technology: Static Power

- **Static power consumption**
  - $\text{Current}_{\text{static}}$ x Voltage
  - Scales with number of transistors
  - To reduce: power gating

Measuring Performance

- **Typical performance metrics:**
  - Response time
  - Throughput
- **Speedup of X relative to Y**
  - $\frac{\text{Execution time}_Y}{\text{Execution time}_X}$
- **Execution time**
  - Wall clock time: includes all system overheads
  - CPU time: only computation time
- **Benchmarks**
  - Kernels (e.g. matrix multiply)
  - Toy programs (e.g. sorting)
  - Synthetic benchmarks (e.g. Dhrystone)
  - Benchmark suites (e.g. SPEC06fp, TPC-C)

Principles of Computer Design (1)

- **Take Advantage of Parallelism**
  - e.g. multiple processors, disks, memory banks, pipelining, multiple functional units
- **Principle of Locality**
  - Reuse of data and instructions
- **Focus on the Common Case**
  - Amdahl's Law

$$\text{Execution time}_{\text{new}} = \text{Execution time}_{\text{old}} \times \left((1 - \text{Fraction}_{\text{enhanced}}) + \frac{\text{Fraction}_{\text{enhanced}}}{\text{Speedup}_{\text{enhanced}}} \right)$$

$$\text{Speedup}_{\text{overall}} = \frac{\text{Execution time}_{\text{old}}}{\text{Execution time}_{\text{new}}} = \frac{1}{(1 - \text{Fraction}_{\text{enhanced}}) + \frac{\text{Fraction}_{\text{enhanced}}}{\text{Speedup}_{\text{enhanced}}}}$$

Principles of Computer Design (2)

- **The Processor Performance Equation**

$$\text{CPU time} = \text{CPU clock cycles for a program} \times \text{Clock cycle time}$$

$$\text{CPU time} = \frac{\text{CPU clock cycles for a program}}{\text{Clock rate}}$$

$$\text{CPI} = \frac{\text{CPU clock cycles for a program}}{\text{Instruction count}}$$

$$\text{CPU time} = \text{Instruction count} \times \text{Cycles per instruction} \times \text{Clock cycle time}$$

$$\frac{\text{Instructions}}{\text{Program}} \times \frac{\text{Clock cycles}}{\text{Instruction}} \times \frac{\text{Seconds}}{\text{Clock cycle}} = \frac{\text{Seconds}}{\text{Program}} = \text{CPU time}$$

Better Algorithm (Compiler)

CPI (Computer design)

electronic improvement

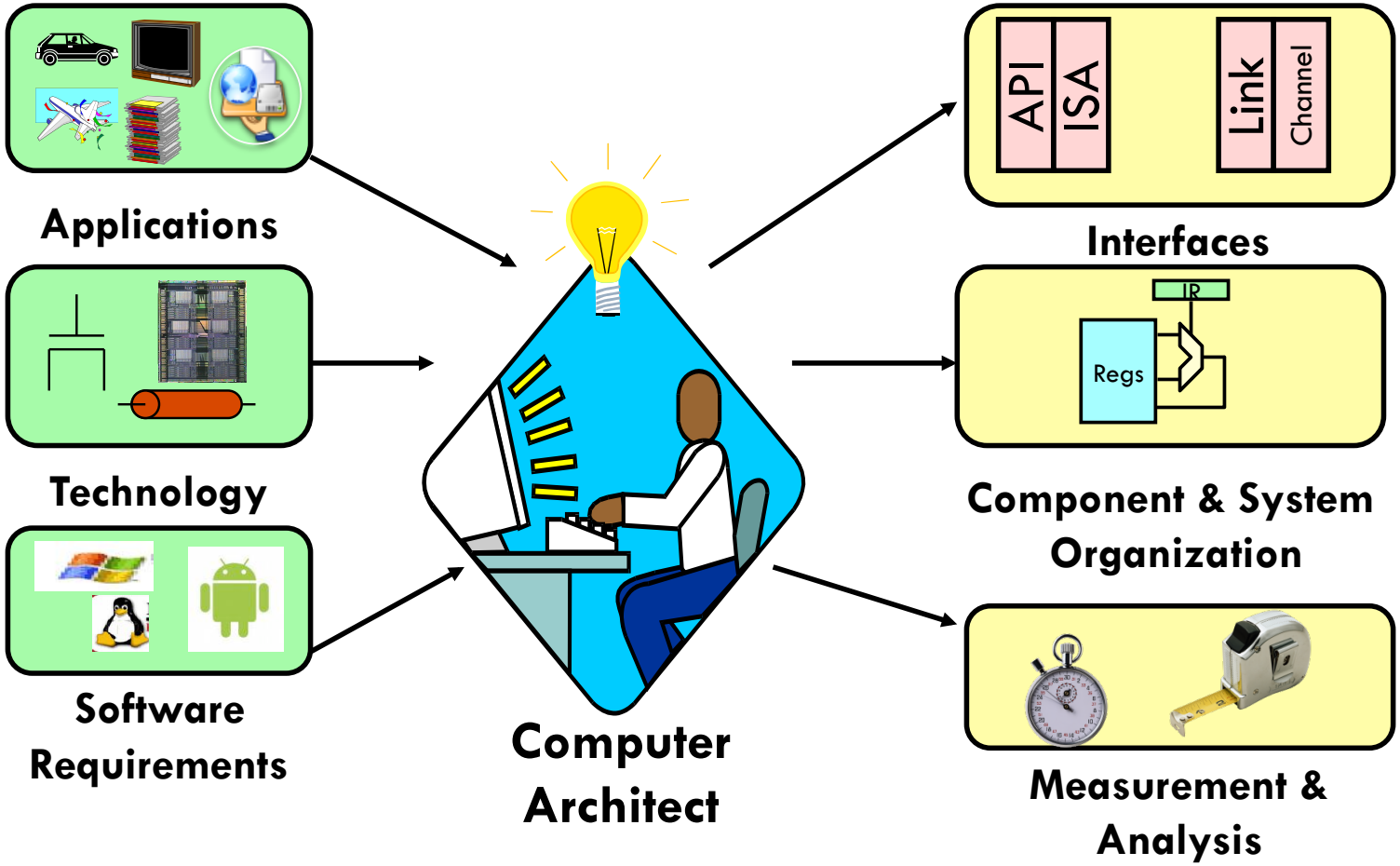
Principles of Computer Design (3)

- **Different instruction types having different CPIs**

$$\text{CPU clock cycles} = \sum_{i=1}^{n} \text{IC}_i \times \text{CPI}_i$$

$$\text{CPU time} = \left(\sum_{i=1}^{n} \text{IC}_i \times \text{CPI}_i \right) \times \text{Clock cycle time}$$

Summary - What Computer Architects Do?



Source: Stanford EE282 (Prof. C. Kozyrakis)