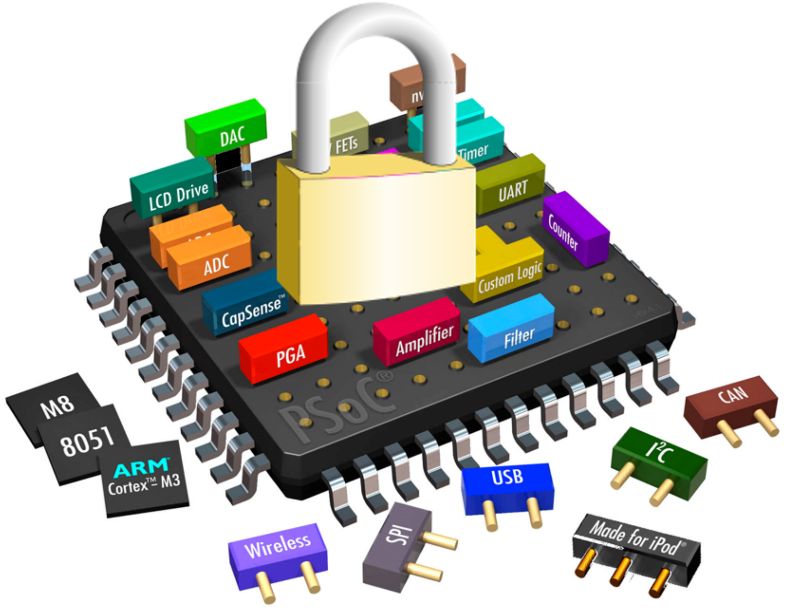
KOREA UNIVERSITY

Advanced Computer Architecture

Fundamentals of Quantitative Design and Analysis

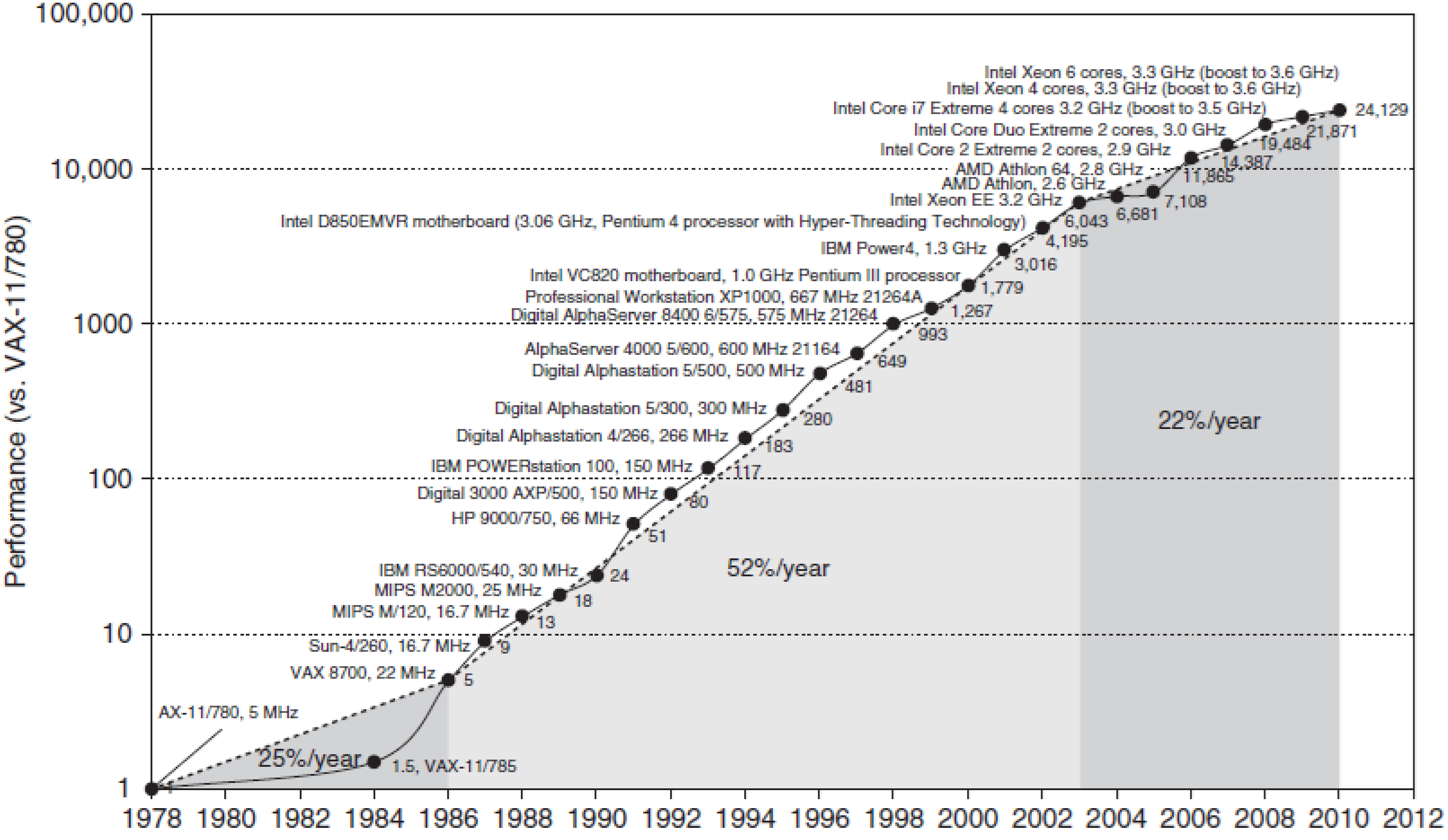


Personal Computers

"It's fair to say that personal computers have become the most empowering tool we've ever created. They're tools of communication, they're tools of creativity, and they can be shaped by their user."

- Bill Gates, February 24, 2004

The Evolution of Computer Performance



Computer technology has made incredible progress in roughly 65 years. Today, less than $500 will purchase a mobile computer with more performance, memory, and storage than a $1 million computer from 1985.

As shown in the graph, processor performance grew at about 25% annually until the mid-1980s, then accelerated to 52% per year until 2003, creating a 25,000-fold improvement since 1978. Since 2003, single-processor performance improvement has dropped to less than 22% per year due to power limitations and lack of instruction-level parallelism to exploit efficiently.

Architectural Innovation

1970s

Early electronic computers delivered performance improvements of about 25% per year through both technological and architectural advances.

Early 1980s

RISC (Reduced Instruction Set Computer) architectures emerged, focusing on instruction-level parallelism and sophisticated cache usage.

1

2

3

4

Late 1970s

Emergence of the microprocessor, riding improvements in integrated circuit technology, led to higher performance growth of roughly 35% per year.

1990s-2003

Combination of architectural and organizational enhancements led to 17 years of sustained growth at over 50% annually—unprecedented in the industry.

Two significant marketplace changes enabled new architectures to succeed: virtual elimination of assembly language programming (reducing need for object-code compatibility) and standardized operating systems like UNIX and Linux.

Four Major Impacts

Enhanced Computing Capability

Today's highest-performance microprocessors outperform supercomputers from less than 10 years ago, dramatically expanding what users can accomplish.

New Classes of Computers

Cost-performance improvements enabled personal computers and workstations in the 1980s, followed by smart phones and tablets that are now primary computing platforms for many.

Microprocessor Dominance

Semiconductor manufacturing improvements led to microprocessor-based computers dominating the entire range of computer design, from embedded systems to supercomputers.

Software Development Revolution

Performance gains allowed programmers to trade speed for productivity, leading to managed languages like Java and C#, scripting languages like Python, and new deployment models like Software as a Service.

End of the Hardware Renaissance

- Since 2003, single-processor performance improvement has dropped to less than 22% per year due to two major hurdles:
  - Maximum power dissipation limits of air-cooled chips
  - Lack of more instruction-level parallelism to exploit efficiently
- In 2004, Intel canceled its high-performance uniprocessor projects, signaling a historic shift from relying solely on instruction-level parallelism (ILP) to multiple processors per chip leveraging data-level parallelism (DLP) and thread-level parallelism (TLP).

Shift to Explicit Parallelism

Instruction-Level Parallelism (ILP)

Compiler and hardware work together to exploit parallelism implicitly without programmer attention. Primary focus of computer architecture through early 2000s.

Explicit Parallelism

Requires restructuring applications to exploit parallelism explicitly, creating major new burdens for programmers.

New Parallelism Types

Data-level parallelism (DLP), thread-level parallelism (TLP), and request-level parallelism (RLP) in warehouse-scale computers.

This shift represents one of the most significant challenges in computer architecture, requiring new approaches from both hardware designers and software developers.