Chapter 1: Fundamentals of Quantitative Design and Analysis

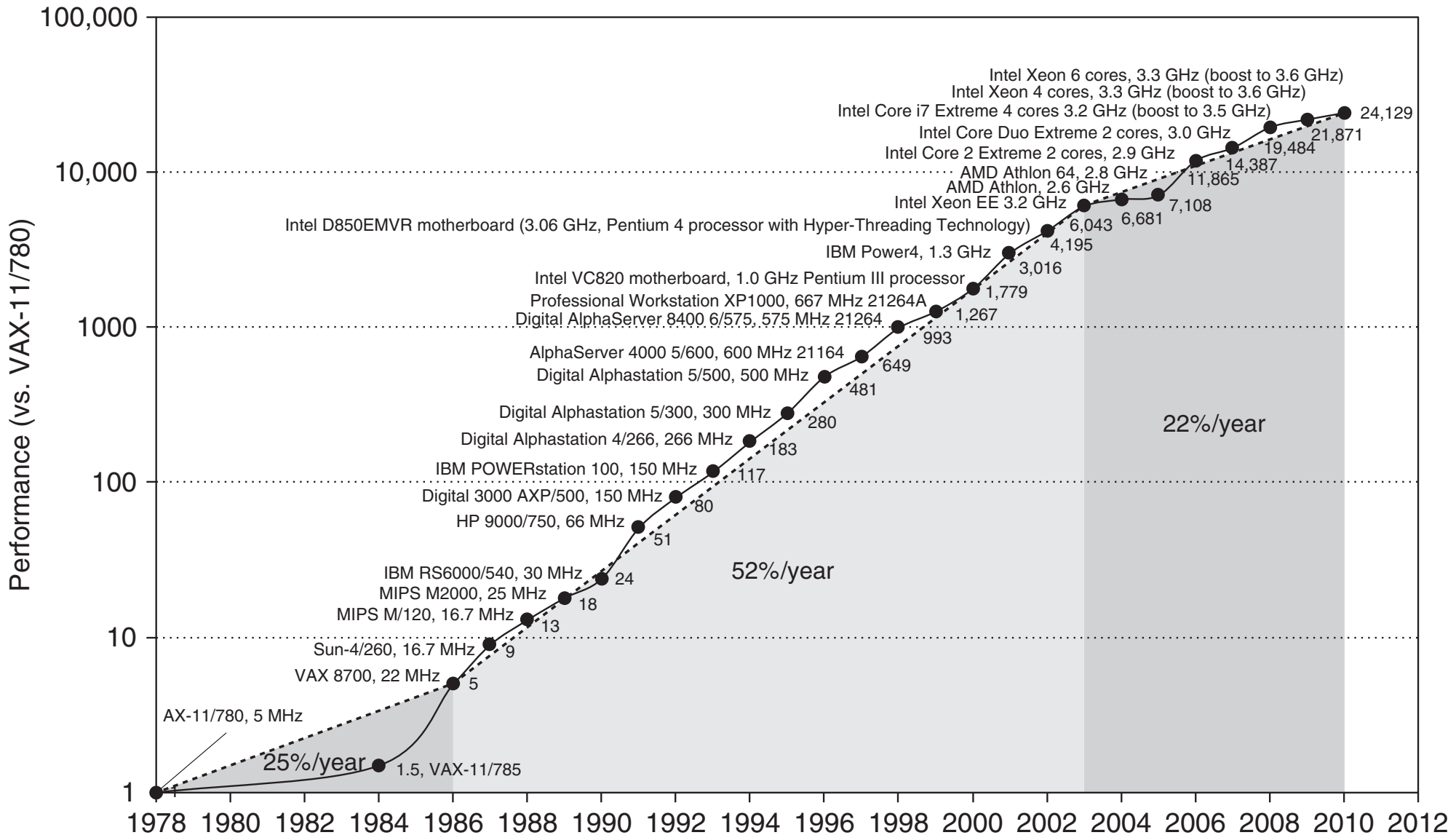
Be careful in this chapter. It contains a tremendous amount of information and data about the changes in computer architecture since the early 80's. Every time I read this chapter it triggers thoughts and reminders of key developments in computer systems. I encourage you to study it carefully and revisit it from time to time in your future years of study.

Chapter Overview

- Classes of Computers
- Defining Computer Architecture
- Trends
- Measuring, Reporting and Summarizing Performance
- Principles of Quantitative Design
- Putting it all together

Growth in Processor Performance

UNIVERSITY OF Cincinnati



Observations of Processor Growth

- RISC & Chip transistor count accelerate growth curve
- Transistor count facilitate translation of x86
- Since 2003 power and ILP limits negatively impact growth
- Transition from ILP to:
  - data-level parallelism (DLP)
  - thread-level parallelism (TLP)
  - request-level parallelism (RLP)

Classes of Computers

- Personal mobile devices (PMD): cost, energy, media perf, responsiveness
- Desktop: price-performance, energy, graphics perf
- Server: throughput, availability, scalability, energy
- Warehouse: price-performance, throughput, energy
- Embedded: price, energy application-targeted performance

Parallelism: Application/Architecture

Application: data-/task-level parallelism (DLP, TLP)

Architecture:

- Instruction Level Parallelism (ILP)
- Vector Machines/Graphics Processing Units (GPUs)
- Thread-level parallelism
- Request-level parallelism

Defining Computer Architecture

Architecture: Covering all aspects of computer design—instruction set architecture, organization (or micro-architecture), and hardware.

Overall the role of the architect is to design a computer (or system) to meet functional requirements and performance requirements where performance includes considerations of all significant requirements including price, power, etc.

Trends

- Technology: transistor density, memory, disk, and network
- Bandwidth and latency: trade latency for bandwidth often works
- Power and Energy: careful here
  - Do nothing well
  - Dynamic Voltage-Frequency Scaling (DVFS)
  - Design for the typical case
  - Overclocking/turbo mode
- Cost

Measuring and Reporting Performance

- Execution time is king. But what is “execution time”: wall-clock, time in CPU?
- Benchmarks: kernels, toy programs, synthetic benchmarks
- Dhrystone SPEC, TPC
- Reproducibility

- Locality: 90/10 rule
- Focus on the common case: drive evaluation of functional and performance requirements in this space.
- Amdahl's Law:

$$Speedup_{overall} = \frac{Execution\ time_{old}}{Execution\ time_{new}}$$

$$= \frac{1}{(1 - Fraction_{enhanced}) + \frac{Fraction_{enhanced}}{Speedup_{enhanced}}}$$

Putting it all together

Remember it is ultimately the system performance that matters. Optimizing one item and ignoring other (sometimes dominate) subsystems will easily lead to failure. For example, the CPU core is only one (often small) factor in the power budget.

Some Key Transition Points in Computing

UNIVERSITY OF Cincinnati

1. D. E. Knuth, “An empirical study of FORTRAN programs,” *Software: Practice and Experience*, vol 1, no 2, 105–133, 1971.
2. M. G. H. Katevenis, *Reduced Instruction Set Computer Architectures for VLSI*, The MIT Press, 1985.
3. Possibly not as fundamental as the two above, the next two are, nevertheless indicating the clear turning point toward multi-/many-core processing:
   - 3.1 A. Ghuloum, “Face the Inevitable, Embrace Parallelism,” *Communications of the ACM*, vol 52, no 9, 36–38, Sep 2009.
   - 3.2 K. Asanovic *et al*, “A View of the Parallel Computing Landscape,” *Communications of the ACM*, vol 52, no 10, 56–67, Oct 2009.
4. And, of course, the textbook for this course.
5. Things to watch:
   - 5.1 H. Esmaeilzadeh, E. Bleem, R. St Amant, K. Sankaralingam, D. Burger, “Dark Silicon and the End of Multicore Scaling”, *Proc of the 38th Int Symp on Computer Architecture* (ISCA ’11), JUne 2011.