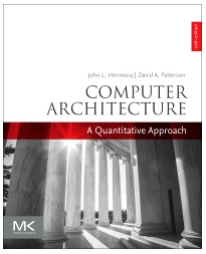
Computer Architecture
A Quantitative Approach, Sixth Edition



Chapter 1

Fundamentals of Quantitative Design and Analysis

 1

Logistics

Course Logistics

- EECS4201 Computer Architecture
- Instructor
  - Mokhtar Aboelaze
  - Office LAS2026 Phone ext: 40607
- Research interests
  - Computer Architecture
  - Low power architecture
  - Embedded systems
  - FPGA (in embedded applications)

 2

EECS 4201

Logistics

- EECS4201 Computer Architecture
- Text
  - Computer Architecture: A Quantitative Approach Patterson & Hennessey 6th Ed.
- Class Meeting
  - Tuesdays, Thursdays 10:00-11:30 HNE 031
- Office Hours
  - Tuesdays, Thursdays 12:00-2:00pm or by appointment

 3

Course Topics

Logistics

- Introduction
- Instruction level parallelism
- Data level parallelism (SIMD and GPU)
- Thread level parallelism
- Memory hierarchy design
- Introduction to warehouse-scale computers
- Domain-Specific Architecture *if time permits*

 4

Course Logistics

Logistics

- After successful completion of the course, students are expected to be able to:
  - Design cache, memory hierarchy, and virtual memory using different techniques to improve cost/performance ratio.
  - Demonstrate how dynamic scheduling and speculative execution can improve the system performance and explain how it is implemented in modern processors.
  - Evaluate different design alternatives and make quantitative/qualitative argument for one design over the other.
  - Identity the different types of parallelism (data, instruction, thread, transaction) for a given application.
  - Compare and evaluate different techniques (such as multithreading, multicore, or vector) to improve CPU performance

 5

Course Logistics

Logistics

- Grades are distributed as follows
  - HW/Assignments **10%**
  - Quizzes **15%**
  - Midterm **25%**
  - Paper review – groups of 2 **10%**
  - Final **40%**

 6

Assumptions

Logistics

- I assume that you already completed EECS2021 or equivalent (you know about these topics).
  - Assembly language
  - RISC architecture
  - ALU architecture
  - Pipelining and hazards
  - Memory hierarchy and cache organization !?

 7

Current Trends in Architecture

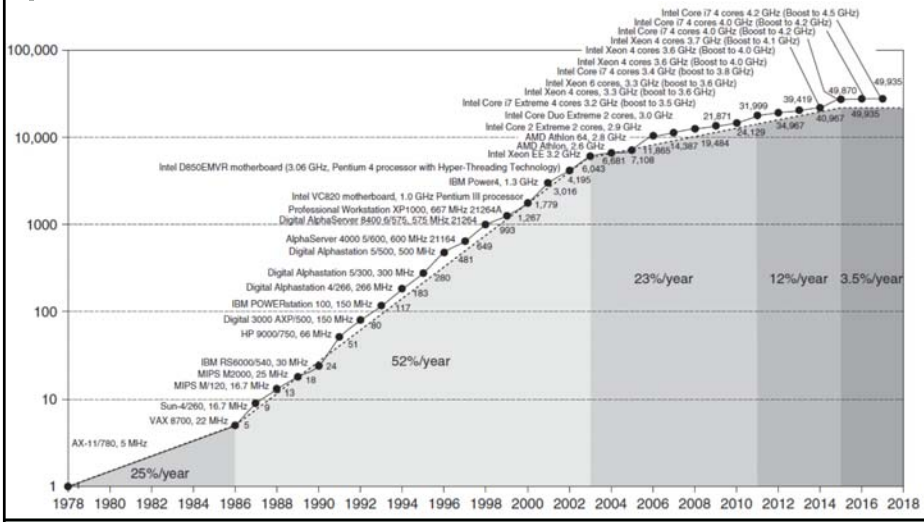
Introduction

- Cannot continue to leverage Instruction-Level parallelism (ILP)
  - Single processor performance improvement ended in 2003 (Dennard scaling ended in 2004)
- New models for performance:
  - Data-level parallelism (DLP)
  - Thread-level parallelism (TLP)
  - Request-level parallelism (RLP)
- These require explicit restructuring of the application (instead of just doing nothing and seeing your code performance improves by Moore's law).

 8

Single Processor Performance

Introduction



 9

Performance

Introduction

- With Moore's law ending (or at least slowing down). We need new ways to improve performance.
  - Multiple cores.
  - Specialization: processors may contain several domain specific cores that performs (very well) one class of computations.

 10

Classes of Computers

- 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
- 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
- Internet of Things/Embedded Computers
  - Emphasis: price, toys to network switches, 8-32 bits, $0.1-$100

Parallelism

- Classes of parallelism in applications:
  - Data-Level Parallelism (DLP)
  - Task-Level Parallelism (TLP)
- Classes of architectural parallelism:
  - Instruction-Level Parallelism (ILP)
  - Vector architectures/Graphic Processor Units (GPUs)
  - Thread-Level Parallelism
  - Request-Level Parallelism

Flynn's Taxonomy

Classes of Computers

- Single instruction stream, single data stream (SISD)
- Single instruction stream, multiple data streams (SIMD)
  - Vector architectures
  - Multimedia extensions
  - Graphics processor units
- Multiple instruction streams, single data stream (MISD)
  - No commercial implementation
- Multiple instruction streams, multiple data streams (MIMD)
  - Tightly-coupled MIMD
  - Loosely-coupled MIMD

 13

Defining Computer Architecture

Defining Computer Architecture

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

 14