Computer Architecture
A Quantitative Approach, Fifth Edition

Chapter 1

Fundamentals of Quantitative Design and Analysis

These slides are based on the slides provided by the publisher.
The slides will be modified, annotated, explained on the board, and sometimes corrected in the class

Copyright © 2012, Elsevier Inc. All rights reserved. 1

EECS4201 Computer Architecture

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

Copyright © 2012, Elsevier Inc. All rights reserved. 2

EECS4201 Computer Architecture

- Text
  - Computer Architecture: A Quantitative Approach Patterson & Hennessey 5th Ed.
- Class Meeting
  - Tuesdays, Thursdays 10:11:30 BC 214
- Office Hours
  - Tuesdays, Thursdays 1:00-3:00pm or by appointment

MK MORGAN KAUFMANN Copyright © 2012, Elsevier Inc. All rights reserved. 3

EECS4201 Topics

- Introduction
- Instruction level parallelism
- Data level parallelism (SIMD and GPU)
- Thread level parallelism
- Memory hierarchy design
- Introduction to warehouse-scale computers

MK MORGAN KAUFMANN Copyright © 2012, Elsevier Inc. All rights reserved. 4

Grading EECS4201

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

Copyright © 2012, Elsevier Inc. All rights reserved. 5

Grading EECS5501

- Grades are distributed as follows
  - HW/Assignments **10%**
  - Quizzes **15%**
  - Midterm **20%**
  - Project **20%**
  - Final **35%**

Copyright © 2012, Elsevier Inc. All rights reserved. 6

Assumptions

- EECS2021 or equivalent
  - Assembly language
  - RISC architecture
  - ALU architecture
  - Pipelining and hazards
  - Memory hierarchy and cache organization !?

MK MORGAN KAUFMANN Copyright © 2012, Elsevier Inc. All rights reserved. 7

Computer Architecture

- Why study computer architecture
- Hardware/Architecture
  - Design better, faster, cheaper computers that use as little energy as possible
- Software
  - Understand the architecture to squeeze as much performance for your code as possible

MK MORGAN KAUFMANN Copyright © 2012, Elsevier Inc. All rights reserved. 8

Computer Technology

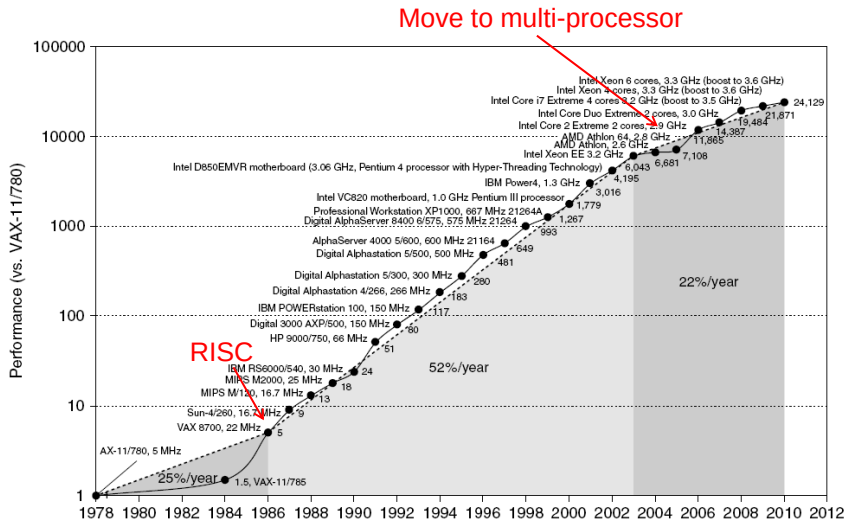
Introduction

- Performance improvements:
  - Improvements in semiconductor technology
    - Feature size, clock speed
  - Improvements in computer architectures
    - Enabled by HLL compilers, UNIX
    - Lead to RISC architectures
  - Together have enabled:
    - Lightweight computers
    - Productivity-based managed/interpreted programming languages

Copyright © 2012, Elsevier Inc. All rights reserved.

Single Processor Performance

Introduction



Copyright © 2012, Elsevier Inc. All rights reserved.

Current Trends in Architecture

- Cannot continue to leverage Instruction-Level parallelism (ILP)
  - Single processor performance improvement ended in 2003
- New models for performance:
  - Data-level parallelism (DLP)
  - Thread-level parallelism (TLP)
  - Request-level parallelism (RLP)
- These require explicit restructuring of the application

MK MORGAN KAUFMANN Copyright © 2012, Elsevier Inc. All rights reserved.

Classes of Computers

- Personal Mobile Device (PMD)
  - e.g. smart 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
- Embedded Computers
  - Emphasis: price

MK MORGAN KAUFMANN Copyright © 2012, Elsevier Inc. All rights reserved.

Parallelism

Classes of Computers

- 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 – Highly coupled
  - Request-Level Parallelism – Decoupled

MK MORGAN KAUFMANN Copyright © 2012, Elsevier Inc. All rights reserved. 13

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

MK MORGAN KAUFMANN Copyright © 2012, Elsevier Inc. All rights reserved. 14

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, hardware

MK MORGAN KAUFMANN Copyright © 2012, Elsevier Inc. All rights reserved.

Trends in Technology

- 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
  - 15-20X cheaper/bit than DRAM
- Magnetic disk technology: 40%/year
  - 15-25X cheaper/bit then Flash
  - 300-500X cheaper/bit than DRAM

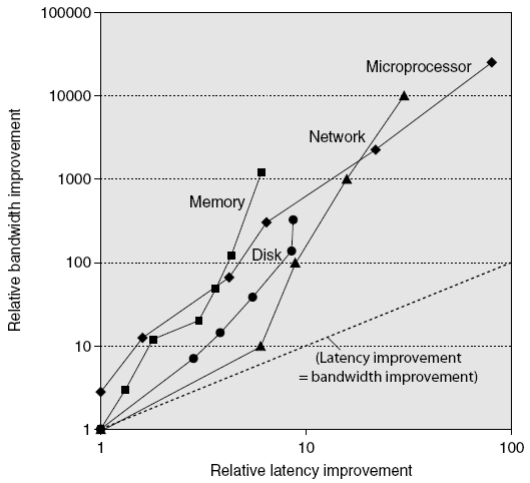
MK MORGAN KAUFMANN Copyright © 2012, Elsevier Inc. All rights reserved.

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
  - Time between start and completion of an event
  - 30-80X improvement for processors
  - 6-8X improvement for memory and disks

Copyright © 2012, Elsevier Inc. All rights reserved.

Bandwidth and Latency



Log-log plot of bandwidth and latency milestones

Copyright © 2012, Elsevier Inc. All rights reserved.

Transistors and Wires

Trends in Technology

- Feature size
  - Minimum size of transistor or wire in x or y dimension
  - 10 microns in 1971 to .032 microns in 2011
  - Transistor performance scales linearly
    - Wire delay does not improve with feature size!
  - Integration density scales quadratically

MK MORGAN KAUFMANN Copyright © 2012, Elsevier Inc. All rights reserved. 19

Power and Energy

Trends in Power and Energy

- Problem: Get power in, get power out
- Power vs. Energy: Which is more important?
- Thermal Design Power (TDP)
  - Characterizes sustained power consumption
  - 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

MK MORGAN KAUFMANN Copyright © 2012, Elsevier Inc. All rights reserved. 20

Dynamic Energy and Power

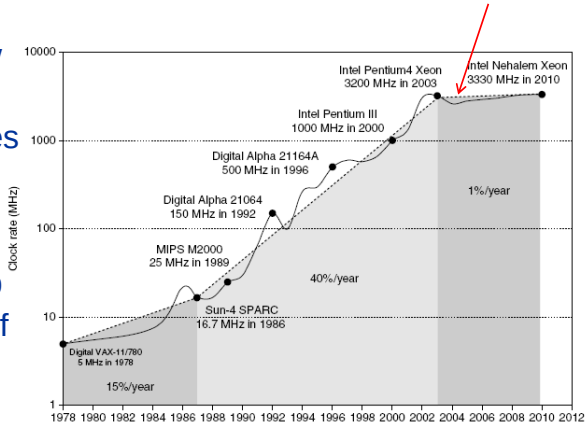
- Dynamic energy
  - Transistor switch from 0 → 1 or 1 → 0
  - ½ x Capacitive load x Voltage^2
- Dynamic power
  - ½ x Capacitive load x Voltage^2 x Frequency switched
- Reducing clock rate reduces power, not energy

Trends in Power and Energy

Copyright © 2012, Elsevier Inc. All rights reserved.

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
- Hot spot ?



Trends in Power and Energy

Copyright © 2012, Elsevier Inc. All rights reserved.

Reducing Power

- Techniques for reducing power:
  - Do nothing well
  - Dynamic Voltage-Frequency Scaling
  - Low power state for DRAM, disks
  - Overclocking, turning off cores

MK MORGAN KAUFMANN Copyright © 2012, Elsevier Inc. All rights reserved.

Static Power

- Static power consumption
  - $\text{Current}_{\text{static}}$ x Voltage
  - Leakage current (power could be as high as 50% of total power consumption) increases with decreasing the transistor size (λ)
  - Scales with number of transistors
  - To reduce: power gating

MK MORGAN KAUFMANN Copyright © 2012, Elsevier Inc. All rights reserved.

Trends in Cost

- Cost driven down by learning curve
  - Yield
- DRAM: price closely tracks cost
- Microprocessors: price depends on volume
  - 10% less for each doubling of volume

Copyright © 2012, Elsevier Inc. All rights reserved.

Integrated Circuit Cost

- Integrated circuit

$$\text{Cost of integrated circuit} = \frac{\text{Cost of die} + \text{Cost of testing die} + \text{Cost of packaging and final test}}{\text{Final test yield}}$$

$$\text{Cost of die} = \frac{\text{Cost of wafer}}{\text{Dies per wafer} \times \text{Die yield}}$$

$$\text{Dies per wafer} = \frac{\pi \times (\text{Wafer diameter}/2)^2}{\text{Die area}} - \frac{\pi \times \text{Wafer diameter}}{\sqrt{2 \times \text{Die area}}}$$

- Bose-Einstein formula:

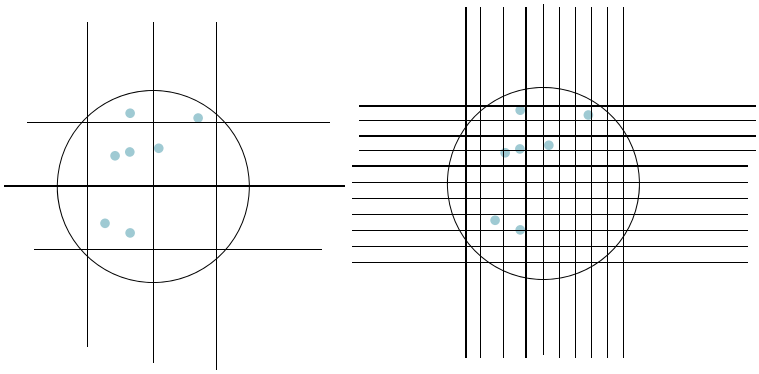
$$\text{Die yield} = \text{Wafer yield} \times 1/(1 + \text{Defects per unit area} \times \text{Die area})^N$$

- Defects per unit area = 0.016-0.057 defects per square cm (2010)
- N = process-complexity factor = 11.5-15.5 (40 nm, 2010)

Copyright © 2012, Elsevier Inc. All rights reserved.

Integrated Circuit Cost

Wafer Yield



Copyright © 2012, Elsevier Inc. All rights reserved.

27

Dependability

Dependability

- Module reliability
  - Mean time to failure (MTTF)
  - Mean time to repair (MTTR)
  - Mean time between failures (MTBF) = MTTF + MTTR
  - Availability = MTTF / (MTTF+MTTR)
- Cost of failure: varies hugely depending on applications

Copyright © 2012, Elsevier Inc. All rights reserved.

28

Measuring Performance

- Typical performance metrics:
  - Response time
  - Throughput
- Speedup of X relative to Y
  - Execution time$_Y$ / 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)

MK MORGAN KAUFMANN Copyright © 2012, Elsevier Inc. All rights reserved. 29

Reporting Performance

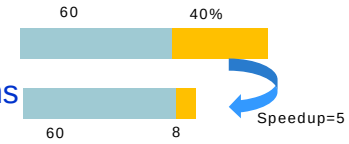
- Must be reproducible
- Complete description of the computer and compiler flags.
- Usually, compared to a standard machine execution time SPECRatioA = T_{ref}/T_A.
- Geometric mean

MK MORGAN KAUFMANN Copyright © 2012, Elsevier Inc. All rights reserved. 30

Principles of Computer Design

Principles

- 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

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

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

MK MORGAN KAUFMANN Copyright © 2012, Elsevier Inc. All rights reserved. 31

Principles of Computer Design

Principles

- The Processor Performance Equation

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

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

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

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

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

MK MORGAN KAUFMANN Copyright © 2012, Elsevier Inc. All rights reserved. 32

Principles of Computer Design

Principles

- Different instruction types having different CPIs

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

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

Copyright © 2012, Elsevier Inc. All rights reserved.

Example

Example

Copyright © 2012, Elsevier Inc. All rights reserved.

Example

Copyright © 2012, Elsevier Inc. All rights reserved.

Fallacies and Pitfalls

Copyright © 2012, Elsevier Inc. All rights reserved.