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.

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.

EECS4201 Computer Architecture

- Text
  - Computer Architecture: A Quantitative Approach Patterson & Hennessey 5th Ed.
- Class Meeting
  - Tuesdays, Thursdays 10:00-11:30 CB122
- Office Hours
  - Tuesdays, Thursdays 1:00-3:00pm or by appointment

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
- SOC and MPSOC *if time permits*

Copyright © 2012, Elsevier Inc. All rights reserved. 4

Course Learning Outcomes

- 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

Copyright © 2012, Elsevier Inc. All rights reserved.

Grading EECS4201

- Grades are distributed as follows
  - HW/Assignments **0%**
  - Quizzes **15%**
  - Midterm **25%**
  - Paper review – groups of 2 **15%**
  - Final **45%**

Copyright © 2012, Elsevier Inc. All rights reserved.

Grading EECS5501

- Grades are distributed as follows
  - HW/Assignments **0%**
  - Quizzes **15%**
  - Midterm **25%**
  - Project **20%**
  - Final **40%**

Copyright © 2012, Elsevier Inc. All rights reserved. 7

Assumptions

- 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 !?

Copyright © 2012, Elsevier Inc. All rights reserved. 8

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

Copyright © 2012, Elsevier Inc. All rights reserved. 9

What is Computer Architcture

- The **interface** between hardware and software.
- Computer Architecture usually includes
  - Instruction set design
  - Where data is stored (registers) and processed (functional units).
  - Software conventions

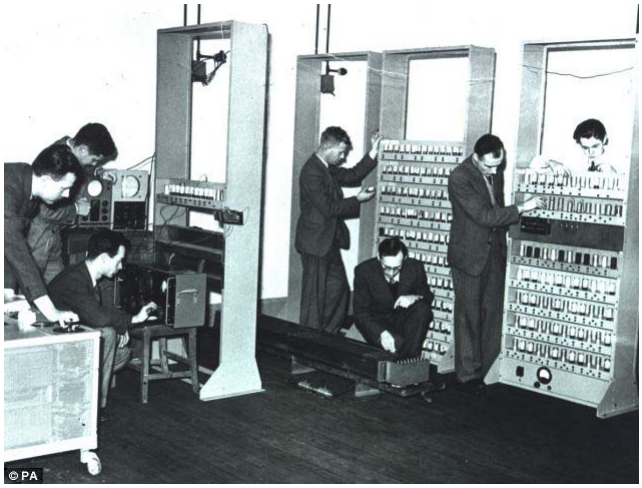
Copyright © 2012, Elsevier Inc. All rights reserved. 10

Computer Architect Job?

- How to utilize the existing resources to achieve the required goal.
- Could include adding more resources within some constraints
- The constraints are
  - Performance
  - Reliability and security (getting more important with smaller λ and more pervasive devices.
  - Power/energy consumption

Copyright © 2012, Elsevier Inc. All rights reserved. 11

Computers – A History

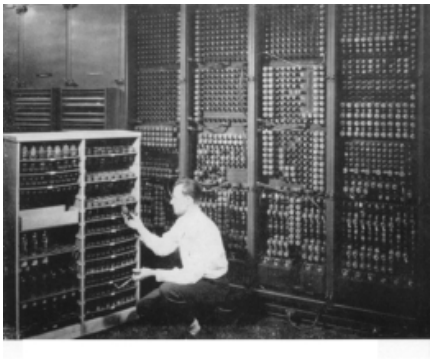


EDSAC – First programmable general purpose computer. University of Cambridge

http://www.dailymail.co.uk/sciencetech/article-2314063/Who-needs-silicon-valley-The-Cambridge-lab-created-programmable-celebrates-75th-birthday.html

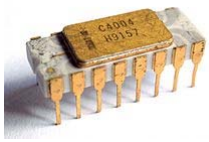
Copyright © 2012, Elsevier Inc. All rights reserved. 12

ENIAC



Copyright © 2012, Elsevier Inc. All rights reserved.

History



en.wikipedia.org/wiki/Intel_4004

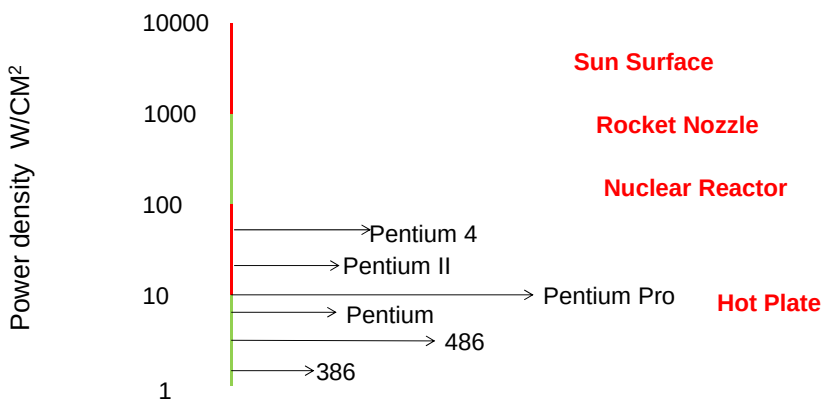
Copyright © 2012, Elsevier Inc. All rights reserved.

History

	4004	Pentium 4	Intel core 2	Xeon
# of transistors	2300	125,000,000	291,000,000	2.6G
Feature size	10μ PMOS	90nm	65n	32nm
Year	1971	2005	2006	2011
Power	1W	≈ 50-60W	≈ 60W	≈ 90-130W
Area	12mm^2	112 mm^2	143mm^2	513mm^2
Data width	4	64	64	64
# of Cores	WHAT?	1	2	1-6
Clock	108K	3.8G	3 Ghz	2.4GHz

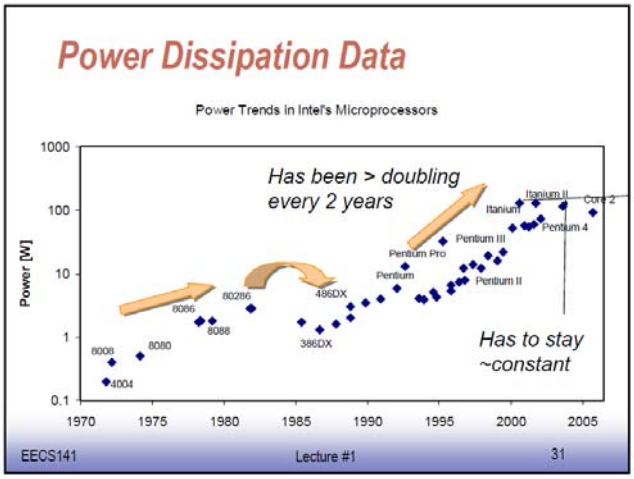
Copyright © 2012, Elsevier Inc. All rights reserved.

Power density



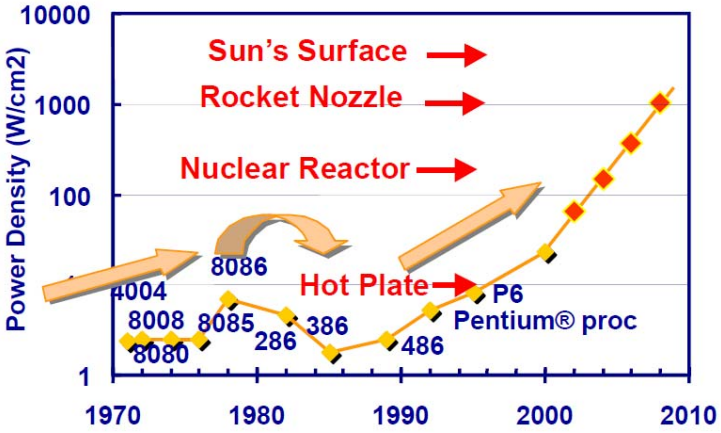
Copyright © 2012, Elsevier Inc. All rights reserved.

Power



Copyright © 2012, Elsevier Inc. All rights reserved.

Power Density



Copyright © 2012, Elsevier Inc. All rights reserved.

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

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 25-50% of total power consumption) increases with decreasing the transistor size (λ)
  - Scales with number of transistors
  - To reduce: power gating

Copyright © 2012, Elsevier Inc. All rights reserved.

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

Copyright © 2012, Elsevier Inc. All rights reserved.

Reducing Power

- Techniques for reducing power:
  - Do nothing well
  - Dynamic Voltage-Frequency Scaling
  - Design for typical case: for example PMD are idle most of the time, low power state for DRAM, disks
  - Overclocking, turning off cores

Copyright © 2012, Elsevier Inc. All rights reserved.

Example

- Dynamic Frequency Scaling DFS
- Dynamic Voltage and Frequency Scaling DVFS (if we reduce voltage, we have to decrease frequency).

Copyright © 2012, Elsevier Inc. All rights reserved.

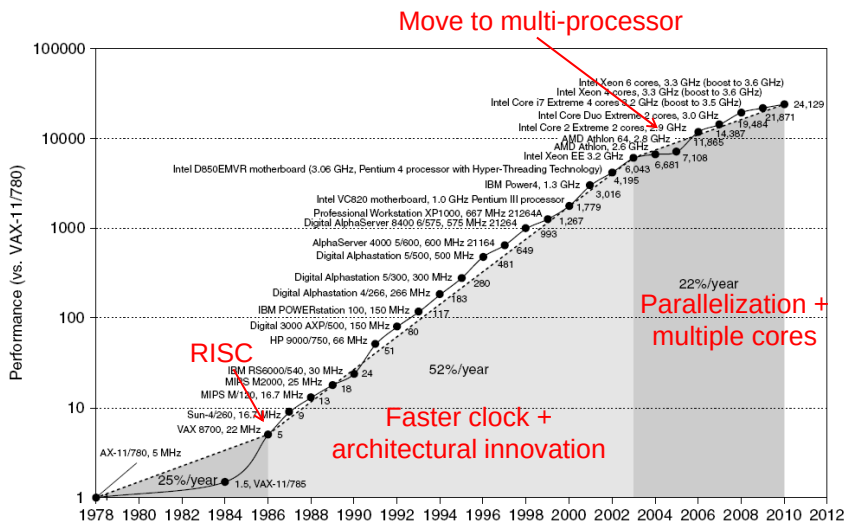
Modern Computer Systems

Application
O/S
ISA
Microarchitecture
Gates/RTL
Circuits
Transistors

Copyright © 2012, Elsevier Inc. All rights reserved.

Single Processor Performance

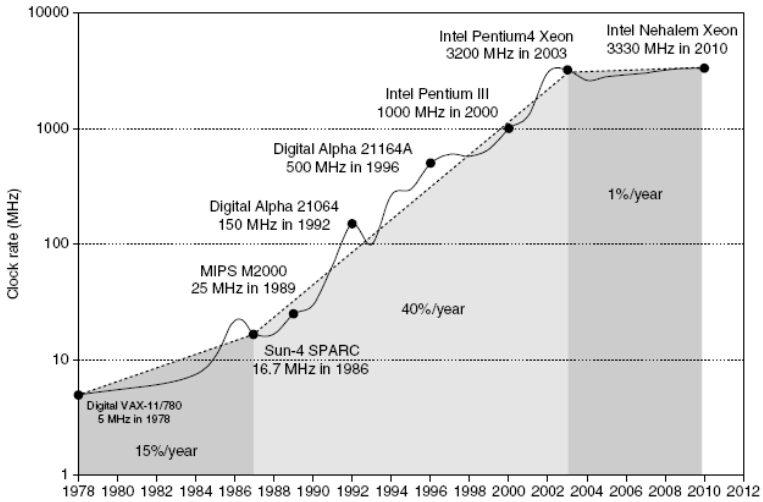
Introduction



Copyright © 2012, Elsevier Inc. All rights reserved.

Clock Speed

Introduction



Copyright © 2012, Elsevier Inc. All rights reserved.

Current Trends in Architecture

Introduction

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

Copyright © 2012, Elsevier Inc. All rights reserved.

Clases of Applications

- Scientific
  - Quantum physics, weather prediction, genome sequencing
  - Original applications computers were made for
- Commercial
  - Databases, e-business,
- Home and office
  - Many many applications
- Embedded and deep embedded
  - Appliances, cars, and disposable

Copyright © 2012, Elsevier Inc. All rights reserved.

Classes of Computers

- Supercomputers
  - Emphasis on performance
- 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

Copyright © 2012, Elsevier Inc. All rights reserved.

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 – Highly coupled
  - Request-Level Parallelism – Decoupled

Copyright © 2012, Elsevier Inc. All rights reserved.

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

Copyright © 2012, Elsevier Inc. All rights reserved.

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, hardware

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

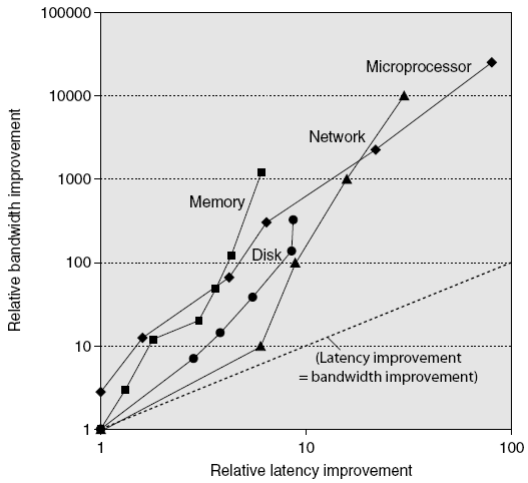
Copyright © 2012, Elsevier Inc. All rights reserved.

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.