CS352H: Computer Systems Architecture

Lecture 1: What Is Computer Architecture?

August 27, 2009

Goals

- Understand the “how” and “why” of computer system organization
  - Instruction set architecture
  - System organization (processor, memory, I/O)
  - Micro-architecture
- Learn methods of evaluating performance
  - Metrics
  - Benchmarks
- Learn how to make systems go fast
  - Pipelining, caching
  - Parallelism
- Learn how to make systems less power hungry (maybe)

Logistics

- Lectures: TTh 3:30-5:00, WEL 3.260
- Instructor: Don Fussell
  - Office: ACES 2.120
  - Office hours: TTh 2:00-3:00 or by appointment
  - Email: fussell@cs.utexas.edu
- TA: Dong Li
  - Office:
  - Office Hours:
  - Email:
- Grading:

Final exam	25%
Midterm	20%
~6 Homework + ~5 paper critiques	30%
Project	25%

  Homework/project late account - 3 days for the semester
- Course webpage: http://www.cs.utexas.edu/~fussell/courses/cs352h

Resources

- Text: Patterson & Hennessy
  Computer Organization & Design: The Hardware/Software Interface, 4th Edition
  (required)
- Reference : Hennessy & Patterson
  Computer Architecture: A Quantitative Approach
  (not required)
- Software packages
  - SPIM for MIPS assembly
  - Verilog for class project
  - You will need a CS department Unix account
- Using Blackboard:
  - Messages about the class (make sure your e-mail address is correct)
  - Other resources as needed in addition to the webpage
  - For your own discussion groups

What is expected of you

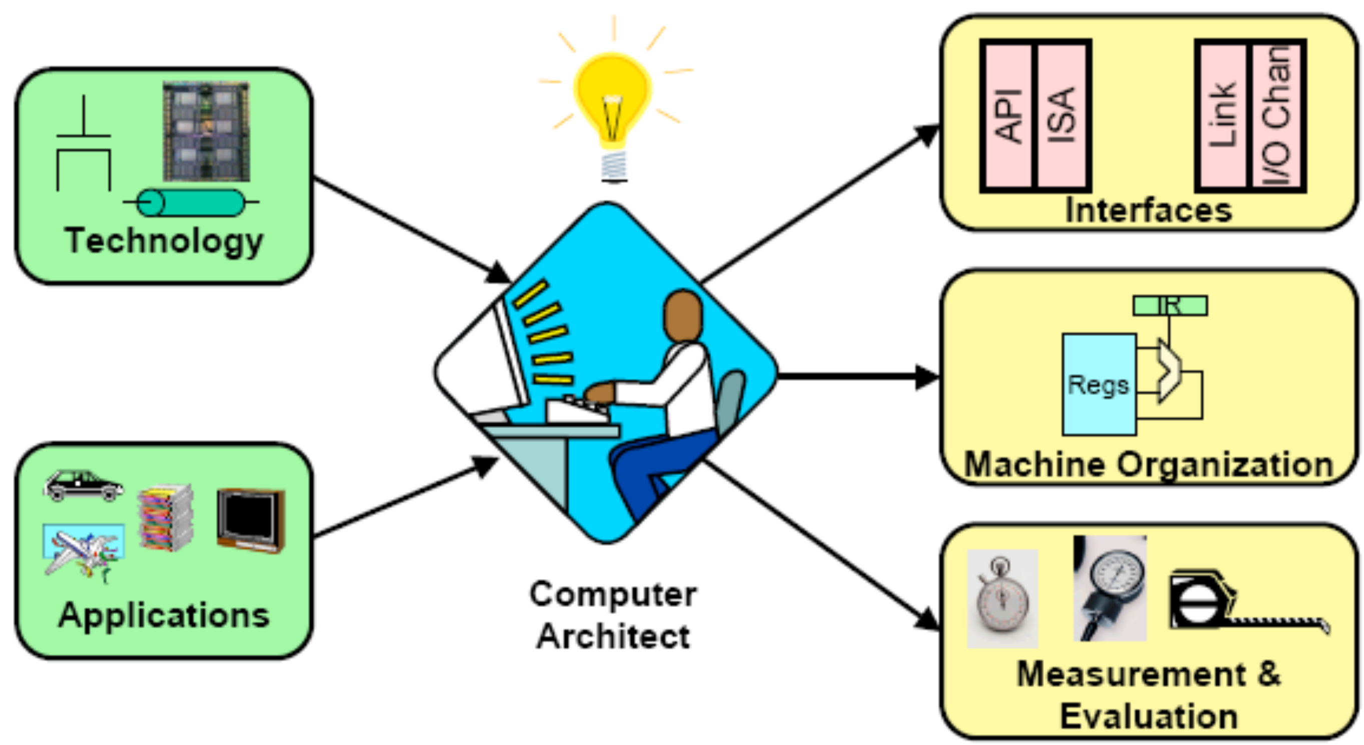
- Attend class and participate
  - Published lecture notes are just “notes”
  - Remember, Powerpoint was designed for content-free industrial presentations, there’s more here than fits into bullets
  - Discussions matter – they’re the best part of the course
  - Be a hero – ask stupid questions, you won’t be the only one who is wondering
- Do the work – the homework and project are more than half your grade for a reason, doing them is the heart of the course
- Things aren’t always as simple as they look – the devil is in the details, so sweat them
- Don’t procrastinate
- Take initiative to follow up on your interests
- Don’t cheat: follow code of conduct

  www.cs.utexas.edu/users/ear/CodeOfConduct.html

Computer Architecture

- "Computer architecture, like other architecture, is the art of determining the needs of the user of a structure and then designing to meet those needs as effectively as possible within economic and technological constraints." F.P. Brooks, Planning a Computer System, Project Stretch, 1962
- What does this "design" consist of?
  - A "programming" interface for software
    - Instructions
    - State changes
  - Organizational principles for processors, memory and I/O devices
    - Processors - microarchitecture
    - Overall - system design
  - In order to:
    - Meet functional and performance targets
    - Within constraints, such as cost and power
    - While taking advantage of advances in technology
- Architecture is about making tradeoffs

What Does a Computer Architect Do?



CS352H Topics

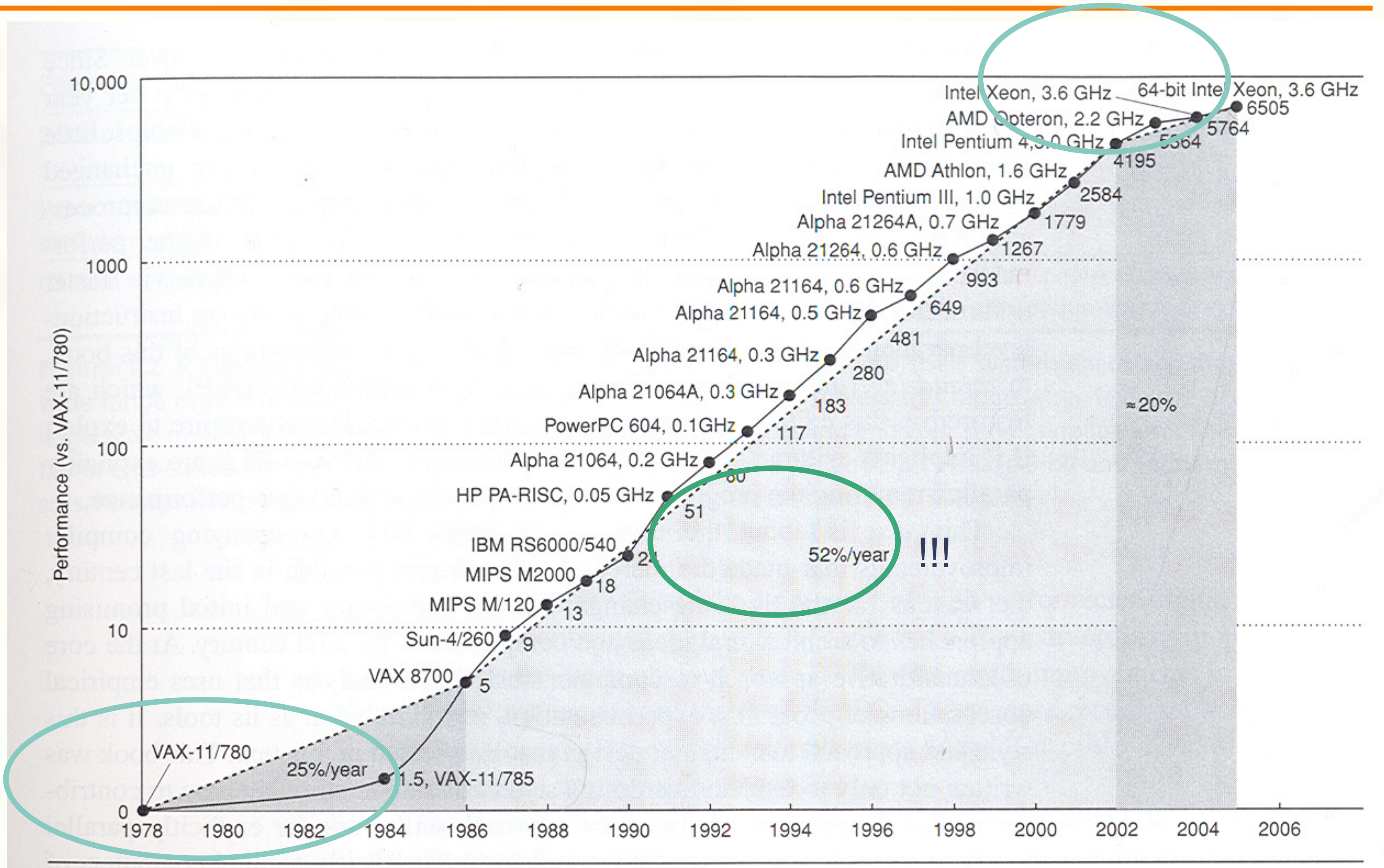
- Technology Trends
- Instruction set architectures
  - MIPS instruction set
- Designing for performance
  - Pipelining
  - Instruction level parallelism (ILP)
    - Static
    - Dynamic
- Memory hierarchies and caches
- Virtual memory
- Multiprocessors and multicore
- System measurement
- System implementation

Application constraints

- Applications drive machine balance
- Numerical simulation, scientific computing, 3-d graphics/games
  - Floating point performance
  - Memory bandwidth
- Transaction processing, databases
  - I/O throughput
  - Integer CPU performance
- Decision support
  - I/O bandwidth
- Embedded controllers
  - I/O timing, interfaces
  - Power
- Video processing
- Low precision pixel arithmetic

A Little Perspective: Processor Performance

Log scale!

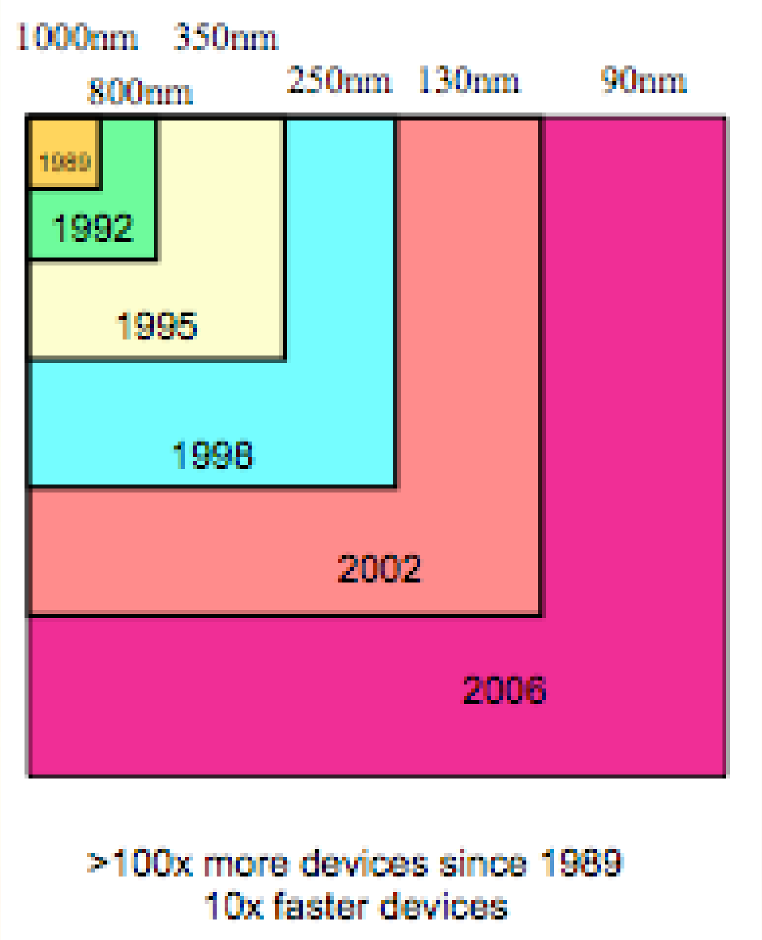


1Hz = 1 cycle per second. 1GHz = 10^9 cycles per second

Taken from: Hennessy & Patterson *Computer Architecture: A Quantitative Approach, 4th ed*

Technology scaling

- CPUs
  - Device density: 2x every 1.5 years (~60% per year)
  - Latency: 2x every 5 years (~15% per year)
- Memory (DRAM)
  - Capacity: 4x every 3 years (~60% per year) (2x every two years lately)
  - Latency: 1.5x every 10 years
  - Cost per bit: decreases about 25% per year
- Hard drives:
  - Capacity: 4x every 3 years (~60% per year)
  - Bandwidth: 2.5x every 4 years
  - Latency: 2x every 5 years
- Boards:
  - Wire density: 2x every 15 years
- Cables:
  - No change

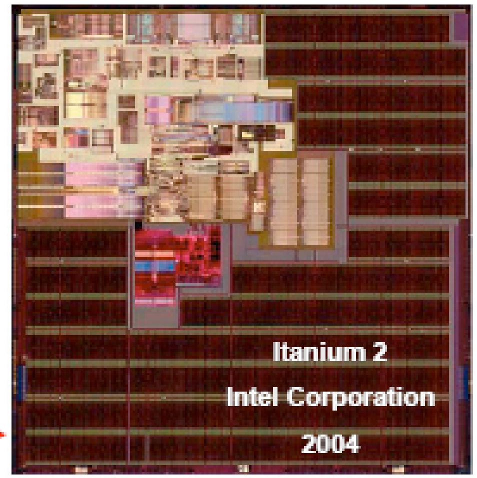


Machines Adapt to Changes in Technology

- 1970s
  - Multi-chip CPUs
  - Semiconductor memory is very expensive
  - Micro-coded control
  - Complex instruction sets
- 1980s
  - Single-chip CPUs
  - Some on-chip RAM
  - Simple, hard-wired control
  - Simple instruction sets
  - Small on-chip caches
- 1990s
  - Lots of transistors
  - Complex control to exploit ILP
  - Lots of on-chip memory
  - Multi-level caches
- 2000s
  - Approaching 1B transistors!
  - Slow wires
  - Power becoming expensive
  - Multicores

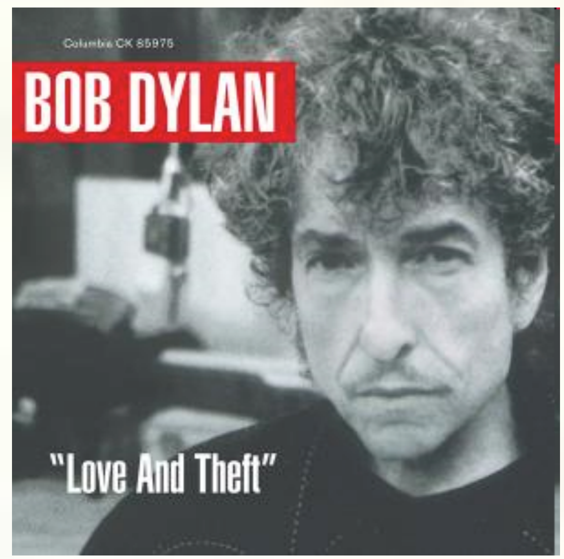
The Times They Are A-Changin'

36X larger



	4004 Processor	Itanium® 2 Processor
Year	1971	2004
Transistors	2300	592 M
Process	10 um	0.13 um
Die size	12 mm²	432 mm²
Frequency	108 kHz	1.7 GHz

1962



2001

What is a Computer?

- Supercomputer?
- Web server?
- Game console?
- Desktop?
- Notebook?
- Microcontroller?

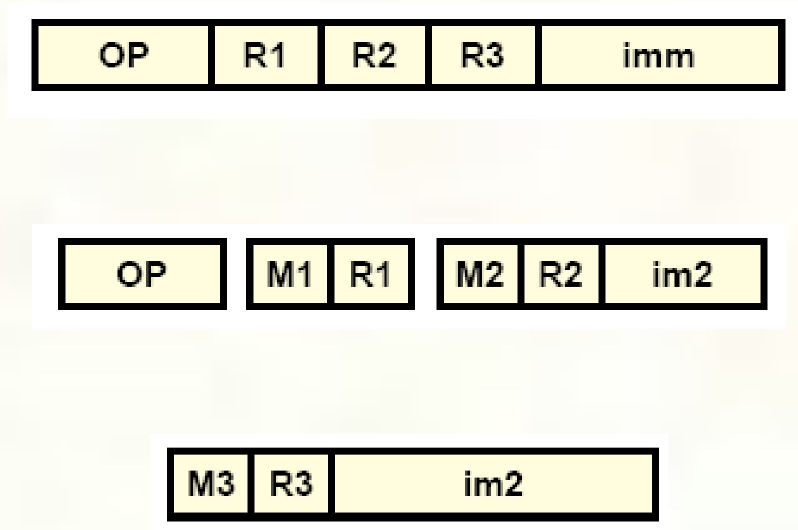
- Each of these “lives” in a different space with its own requirements and constraints
- We need to pick one for this course
  - Desktop
  - Using the MIPS ISA

Interface Design

- Interface vs. Implementation
  - Visibility
- Three types of interfaces:
  - Between layers
    - API, ISA
  - Between modules
    - SCSI, PCI
  - Standard representations
    - IEEE floating point, ASCII
- A good interface
  - Lasts through several generations of implementations
  - Is simple – ‘economy of mechanism’ (KISS)

Instruction-Set Architecture (ISA)

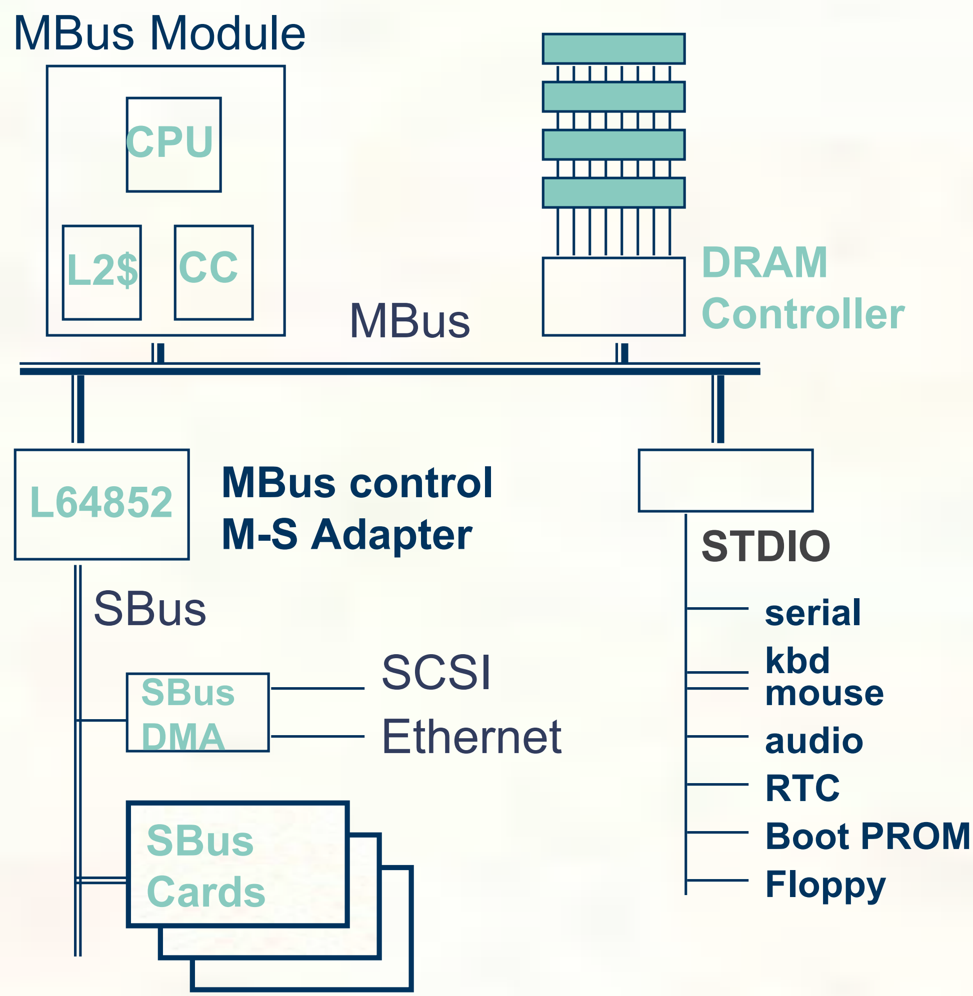
- HW/SW interface
- SW Impact
  - Support OS functions
    - Restartable instructions
    - Memory management
  - A good compiler target
    - Simple
    - Orthogonal
  - Dense
- HW Impact
  - Amenable to efficient implementation over time
  - Amenable to parallelization



System-Level Organization

- Design at the level of processors, memories, …
- More important to application performance than CPU design
- Feeds and speeds
  - Constrained by pin counts and signaling rates
- System balance
  - Application-specific
- Driven by
  - Performance/cost goals
  - Available components
  - Technology constraints

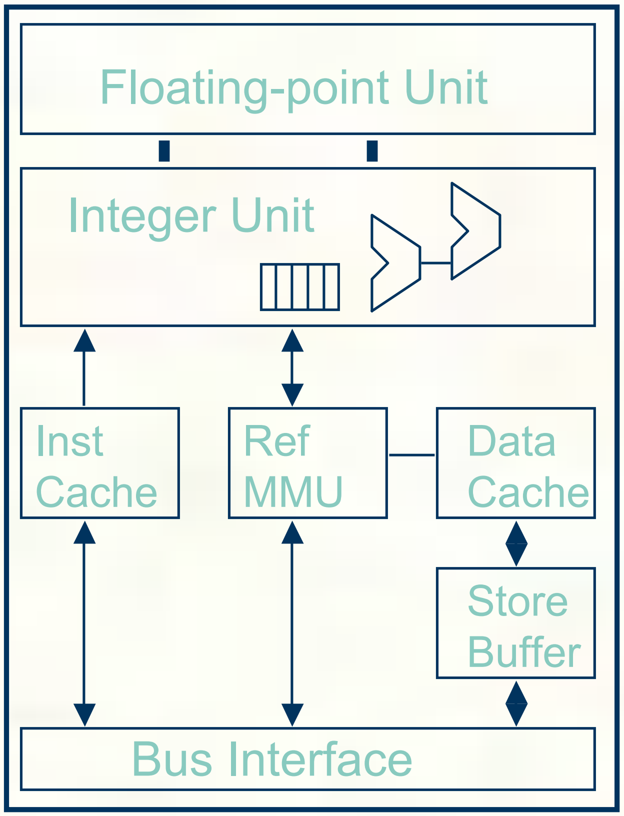
Sun SparcStation20



Micro-architecture

- Register-transfer-level (RTL)
- Implement instruction set
- Exploit technology capabilities
  - Locality & concurrency
- Iterative process
  - Generate architecture
  - Estimate cost
  - Evaluate performance
- Overcoming sequential nature of programs
  - Pipelining
  - Multiple issue
  - Dynamic scheduling
  - Branch prediction/speculation

TI SuperSPARC™ TMS390Z50

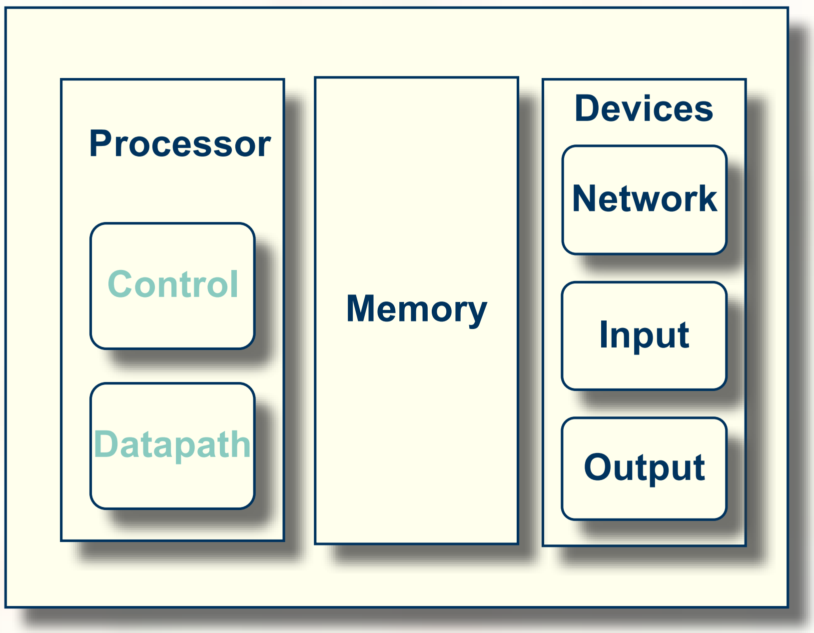


Performance Evaluation

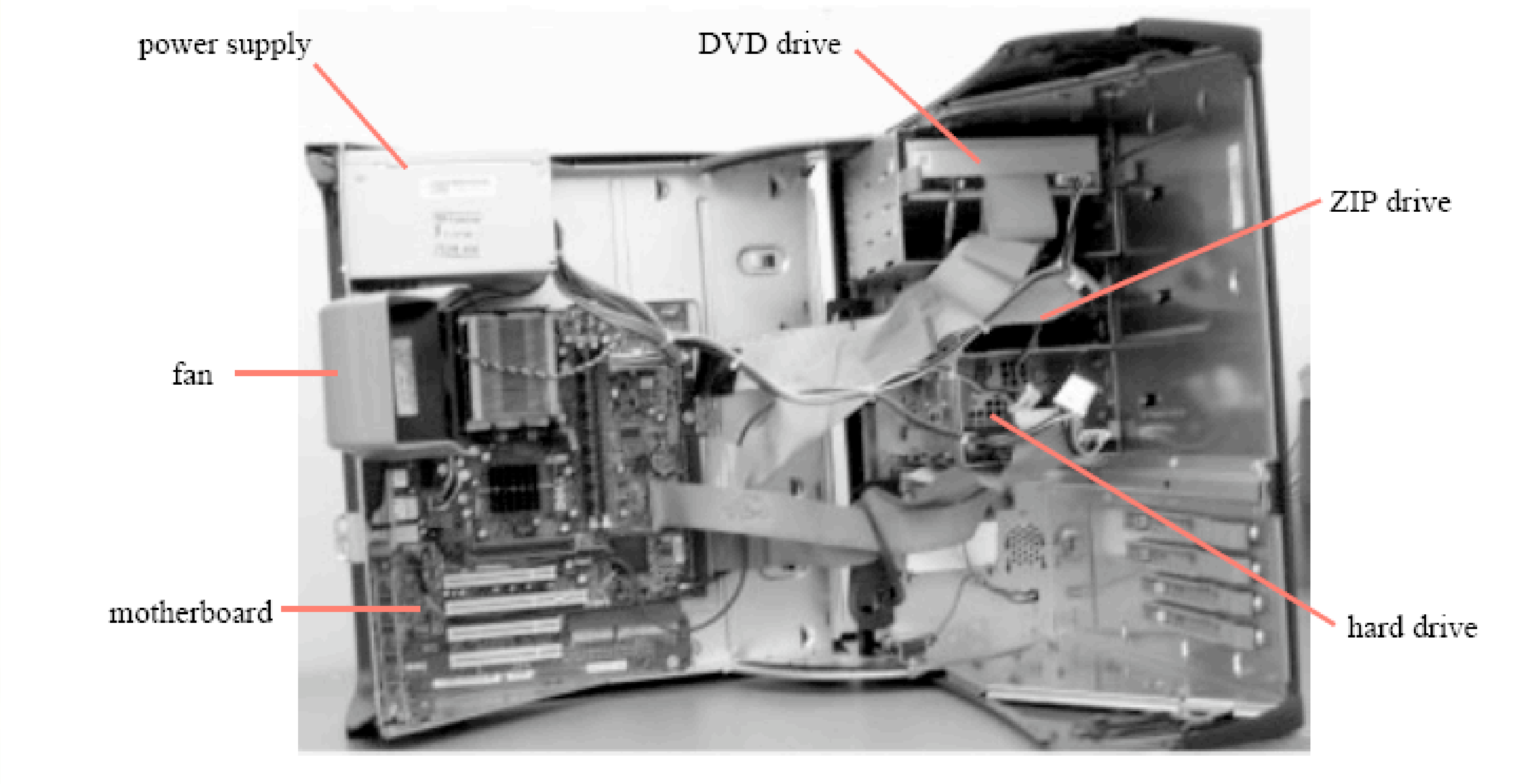
- What's the right measure?
  - Instruction count? cache b/w? I/O throughput?
  - Application response time? Application throughput?
- Measure what?
  - Benchmarks? Real applications? Traces?
- Measure how?
  - Simulation? Math models? Real tests?

Major Components of a Computer

- Desktop design target
  - Processor: 25% of cost
  - Memory: 25% of cost
  - Rest (I/O devices, power supply, enclosure): 50%

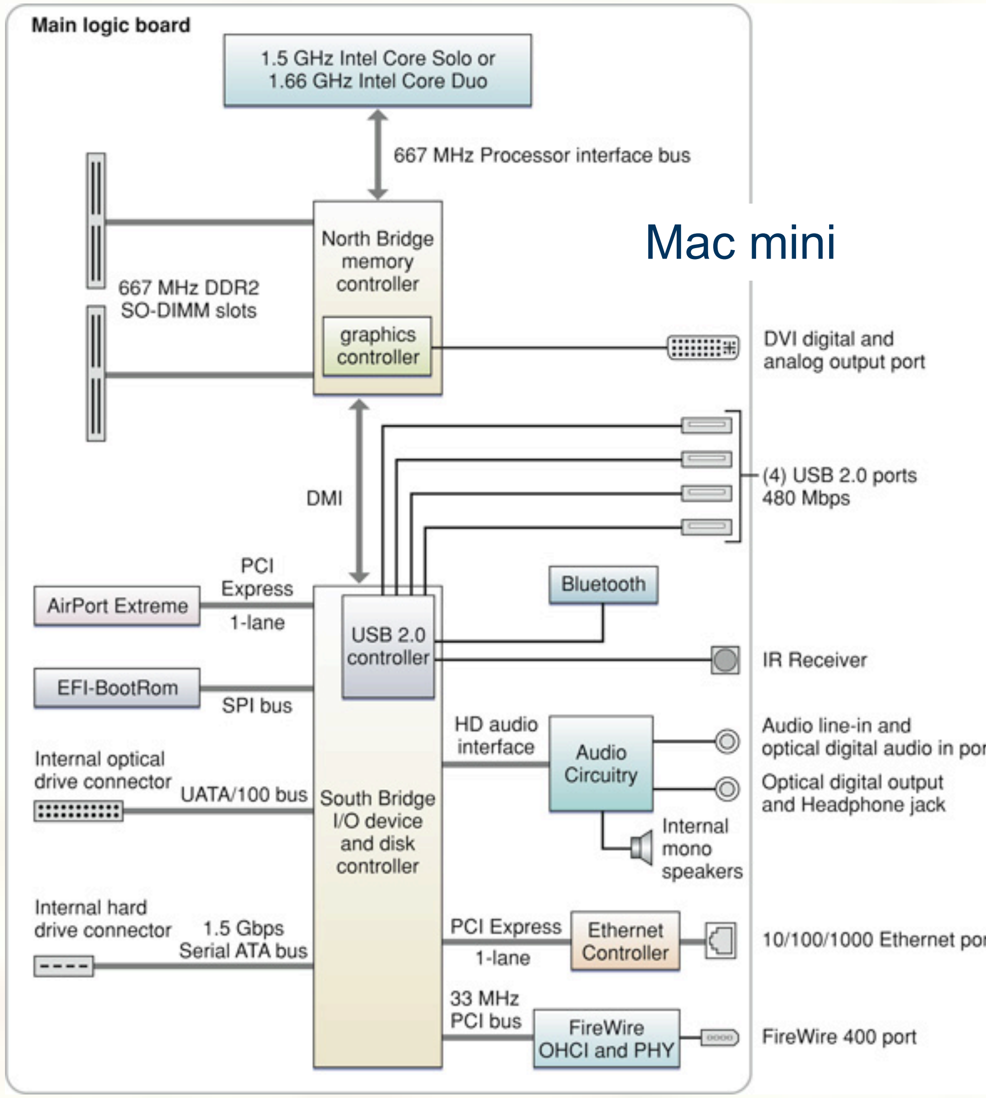


PC Chassis



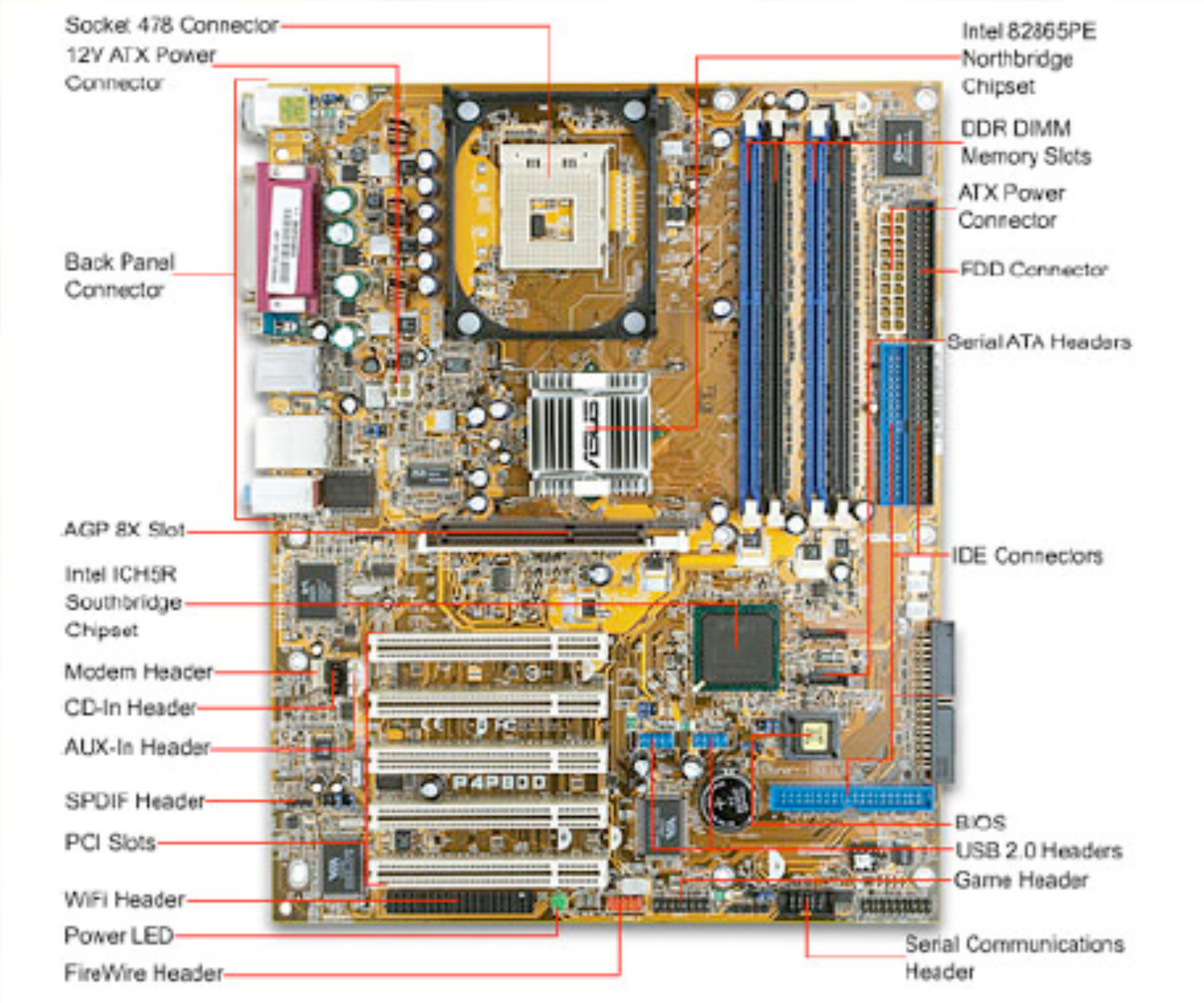
power supply
DVD drive
ZIP drive
fan
motherboard
hard drive

Motherboard organization

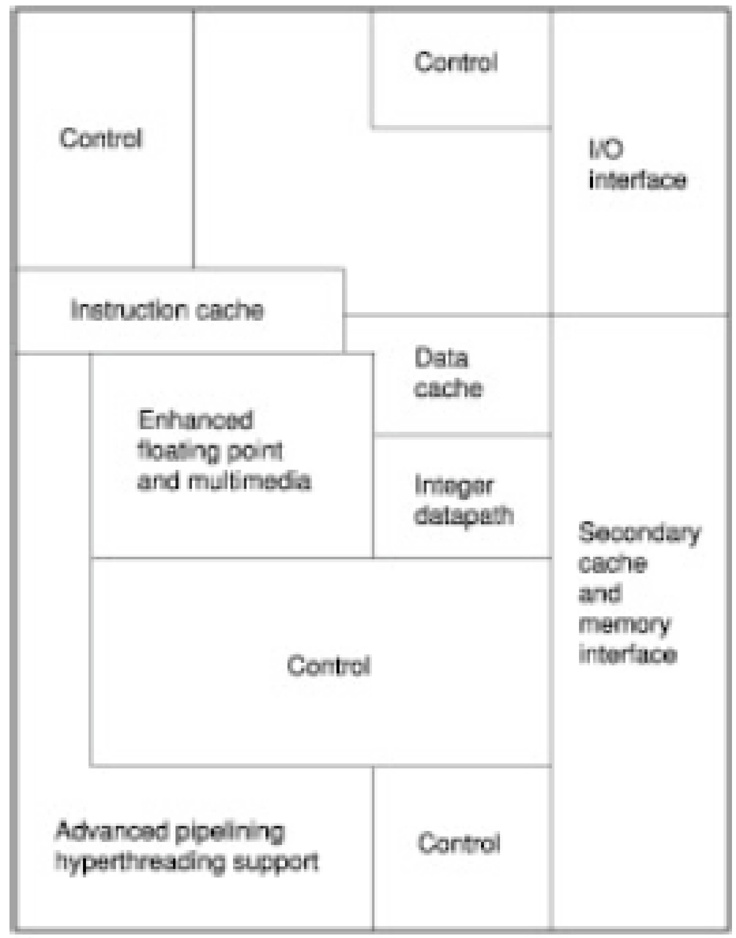


Mac mini

PC Motherboard



Pentium 4 Chip



Summary

- Architecture: Structure of “things”
- Interplay between technology and architecture
- Computer architect uses quantitative methods to:
  - Design interfaces
  - Develop machine organization: system-level & micro-architecture
- This course:
  - Instruction-Set Architectures (ISAs)
  - Performance evaluation (a tiny bit)
  - Pipelining
  - Memory
  - System organization
  - Parallelism
  - Power

First Assignment

- Read the Moore paper (see webpage)
- Write a one page critique
  - Typewritten, font size 10-12
- Critique format
  - Headline message: a one sentence tag line for the paper
  - Elevator pitch: a one paragraph summary of the paper's key messages
  - A couple of paragraphs to support and analyze the elevator pitch
- Hand in your one page review at the beginning of next class
- You should use this format for all other papers as well

Next Time

- Hand in a hard copy of your one-page typewritten review of the Moore paper at the beginning of the class
- We'll discuss the paper
- Then, instruction set architectures, specifically the MIPS ISA
- Read Chap 1 for background
- Read Chap 2.1-2.10 for class prep