Introduction

CSE 410, Spring 2009

Computer Systems

http://www.cs.washington.edu/410/

Reading and References

- Reading
  - » *Computer Organization and Design, Patterson and Hennessy*
    - Chapter 1 (background)
    - Chapter 2, secs. 2.1-2.5

Administrative

- Instructor:
  - » Hal Perkins
  - » perkins@cs.washington.edu, CSE548
- TA:
  - » Braden Pellett
- All class info is on the web site
  - » http://www.cs.washington.edu/410

Class Overview

- Provide an introduction to the inner workings of computer systems
- Levels of abstraction
  - » bits, bytes, assembly language
  - » operating system concepts
  - » higher level languages - C, C++, Java, …
  - » application programs

Goal

- You will understand
  - » what is actually happening when a computer system is running application programs
- So that you will be able to
  - » make good design choices as a developer, project manager, or system customer
  - » calibrate your hype-o-meter with facts

The structure of this class

- The hardware / software interface
  - » the elements of a computer system
  - » what parts are visible to the software
  - » instruction set architecture (ISA)
  - » what happens inside the CPU
- Operating systems
  - » services an OS performs for an application
  - » design of various OS components
  - » OS mechanisms and policies

Course Mechanics

- 3 Lectures/week
- Homeworks most weeks
  - » Written problems, small programming exercises
- Office hours tba, scattered through week
  - » Use them!
- Online discussion board to stay in touch between classes / office hours

Homework & Exams

- ≈ 6-7 assignments (50%)
- Midterm, tentatively Fri. May 1 (20%)
- Final, Tue. June 9, 2:30 (25%)
- Participation, citizenship, etc. (5%)

- Late policy: 4 “late days”, at most 2 on any assignment, counted in 24 hour chunks, otherwise no late assignments.
  - » Save late days for later!

Academic Integrity

- Policy on the course web. **Read it!**
- Do your own work – always explain any unconventional action on your part
- I trust you completely
- I have no sympathy for trust violations – nor should you
- Honest work is the most important feature of a university. It shows respect for your colleagues *and yourself*.

Computers

- Computers impact our lives in a huge number of ways:
  - » Computer-controlled brakes in your car
  - » You look up everything with Google
  - » You take a picture of a bad cut with your cell phone and email it to your doctor
  - » You download music for your MP3 player
- All this has been enabled by an incredible advance in microprocessor technology

Evolution of Intel CPU Speeds

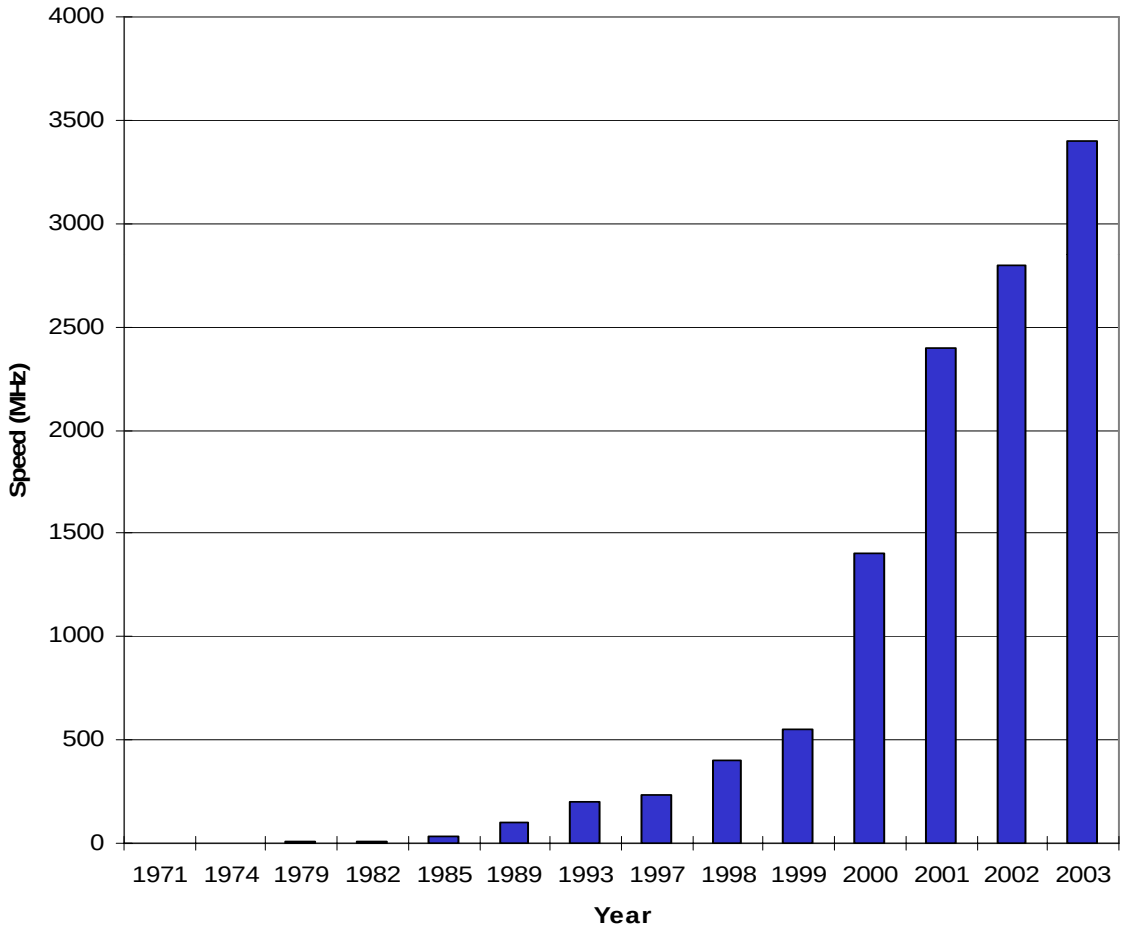
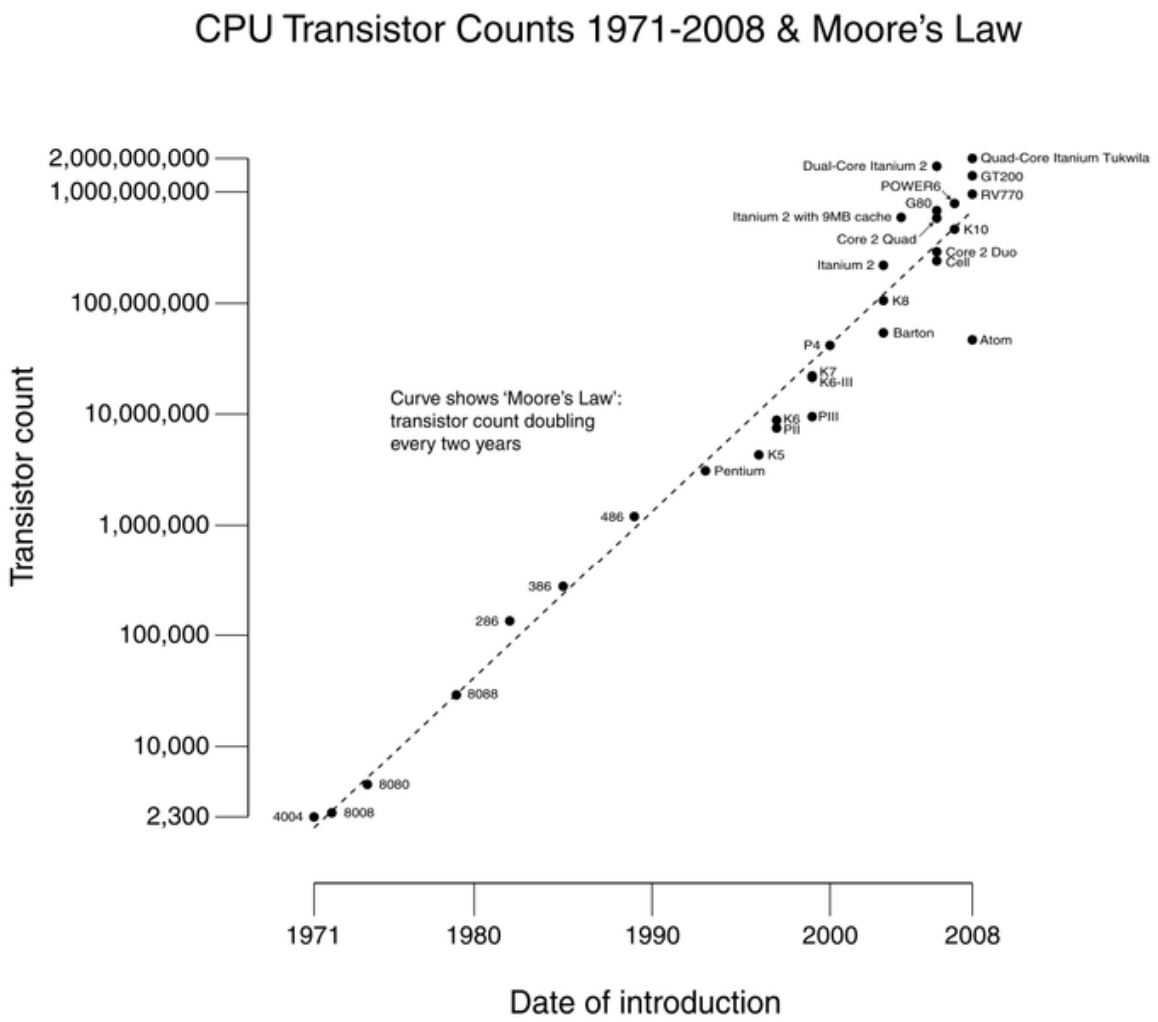


Illustration of Moore's Law



A modern CPU

- Recent Intel chips…
  - » 2-8 cores
  - » > 3.0 gigahertz
  - » >= 2 MB L2 cache
  - » up to 20-stage pipeline (P4, less for others)
  - » out-of-order instruction execution
  - » branch prediction
  - » 100s of instructions executing at once
  - » "hyper-threading" technology
  - » …….

What's next

- We're in trouble
  - » hard to go much faster – gets too hot!
  - » chips have gotten so big, it's a long way from one side to the other (in cycles)
  - » as chips get bigger, chance of errors in the chip goes up
  - » we need new ways to build faster computers
  - » these new ways usually involve adding more parallelism
- In a few years, every chip will have multiple CPUs on it (2-4 now, 16-64 soon) [called "multi-core"]
  - » (How will we take advantage of this? Open question…)

Layers of abstraction

- Abstraction
  - » defines a layer in terms of functions / interfaces
  - » isolates a layer from changes in the layer below
  - » improves developer productivity by reducing detail needed to accomplish a task
  - » helps define a single architecture that can be implemented with more than one organization
- Layers can be hardware, software, or a combination

Architecture and Organization

- Architecture (the boxes)
  - » defines elements and interfaces between layers
  - » ISA: instructions, registers, addressing
- Organization (inside the boxes)
  - » components and connections
  - » how instructions are implemented in hardware
  - » many different organizations can implement a single architecture
  - » One organization can support multiple architectures(!)

Computer Architecture

- Specification of how to program a specific computer family
  - » what instructions are available?
  - » how are the instructions formatted into bits?
  - » how many registers and what is their function?
  - » how is memory addressed?
  - » how does I/O work?
- The MIPS architecture is the basis for the first half of this course
  - » Why not a “real” computer? (e.g., x86)

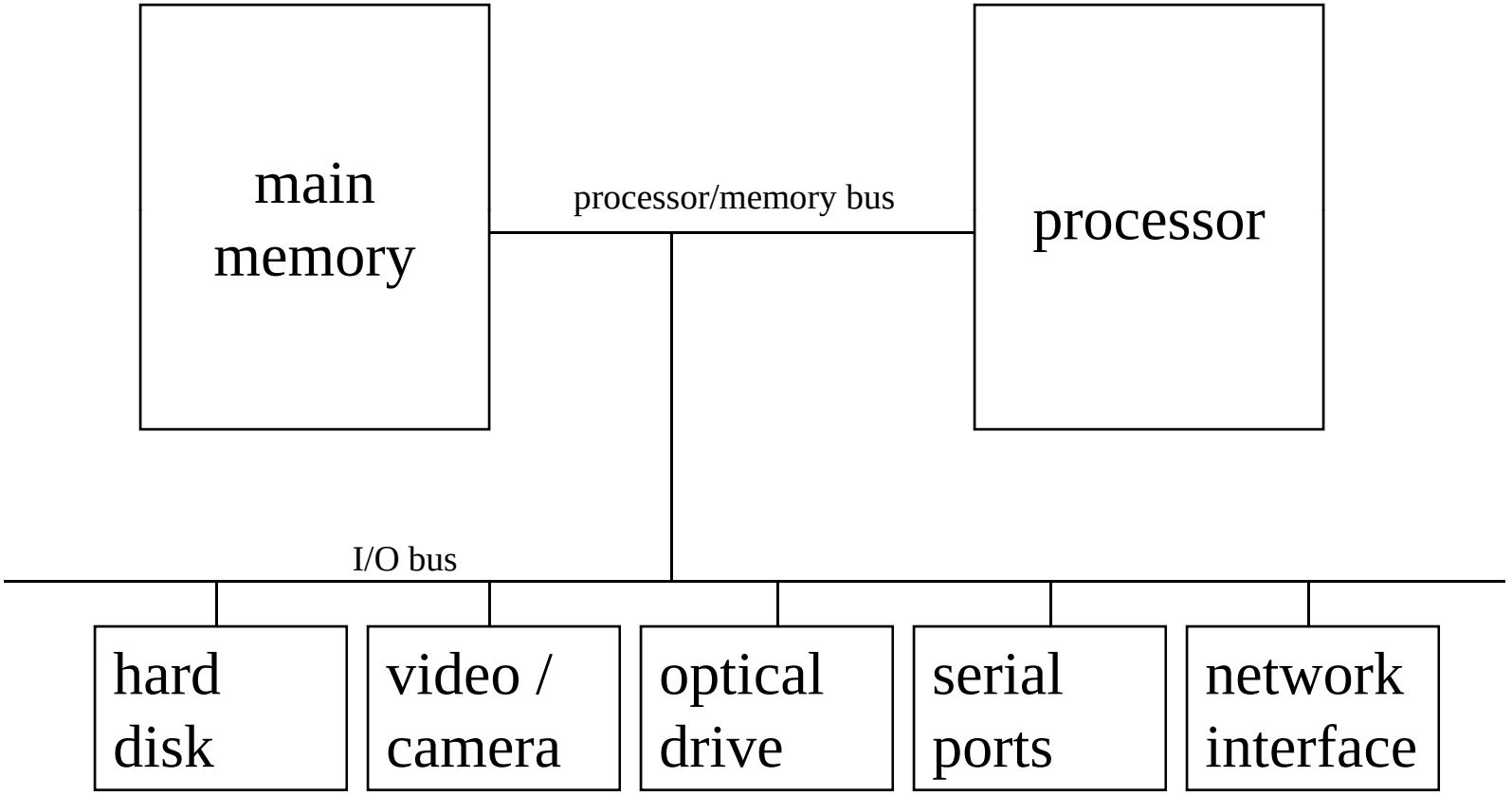
Architecture Families

- IBM 360, 370, … (the first computer family)
- PowerPC 601, 603, …
- DEC VAX, PDP-11
- Intel x86: 286, 386, 486, Pentium, P4, Core…
- Intel IA64 Itanium
- MIPS R2000, R3000, R4000, R5000, ...
- SUN Sparc
- ARM family

Computer Organization

- Processor
  - » datapath (functional units) manipulate the bits
  - » control hardware manages the manipulation
- Memory
  - » Registers – 100s of bytes, very fast, on the CPU
  - » cache memory – 1000s of bytes, fast, on the CPU
  - » main memory – millions of bytes, slower, off the CPU
- Input / Output
  - » interface to the rest of the world

A typical organization



Change Organization or Architecture?

- Theory
  - » Organization changes provide incremental changes in speed and cost for same software
  - » Architecture changes enable breakthrough changes in speed and cost for new software
- Real life
  - » incremental changes are very rapid (once a year)
  - » breakthrough changes are very costly (once a decade)