CS61C
Machine Structures
Lecture 1
August 30,2000

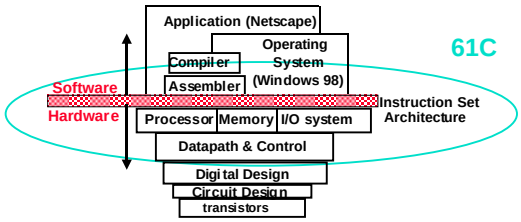
Dave Patterson
(http.cs.berkeley.edu/~patterson)

http://www-inst.eecs.berkeley.edu/~cs61c/

Overview

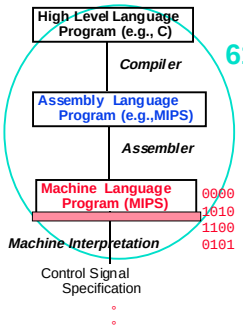
- Intro to Machine Structures (5 minutes)
- Organization and Anatomy of a Computer (10 min)
- Rapid Technological Change (5 min)
- Course Style, Philosophy and Structure (20 min)
- Conclusion (1 min)

What are "Machine Structures"?



- Coordination of many *levels of abstraction*

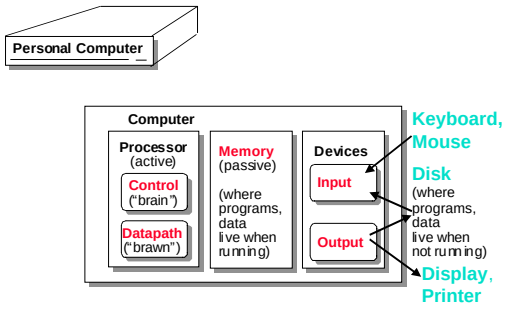
Levels of Representation




```
temp = v[k];
v[k] = v[k+1];
v[k+1] = temp;
lw   $to, 0($2)
lw   $t1, 4($2)
sw   $t1, 0($2)
sw   $t0, 4($2)
```

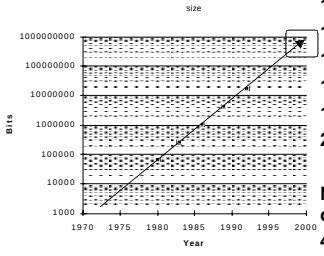
```
0000 1001 1100 0110 1010 1111 0101 1000
1010 1111 0101 1000 0000 1001 1100 0110
1100 0110 1010 1111 0101 1000 0000 1001
0101 1000 0000 1001 1100 0110 1010 1111
```


Anatomy: 5 components of any Computer



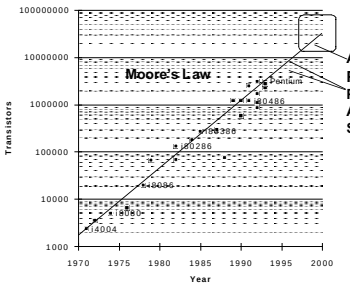
Technology Trends: Memory Capacity (1 Chip DRAM)

year	size(Megabit)
1980	0.0625
1983	0.25
1986	1
1989	4
1992	16
1996	64
2000	256



Now 1.4X/yr, or doubling every 2 years
4000X since 1980

Technology Trends: Microprocessor Capacity

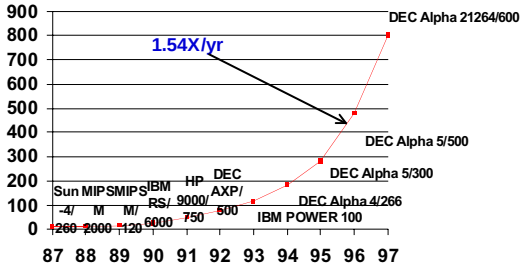


Alpha 21264: 15 million
Pentium Pro: 5.5 million
PowerPC 620: 6.9 million
Alpha 21164: 9.3 million
Sparc Ultra: 5.2 million

2X transistors/Chip Every 1.5 years

Called "Moore's Law"

Technology Trends: Processor Performance



Processor performance increase/year, mistakenly referred to as Moore's Law (transistors/chip)

Computer Technology => Dramatic Change

- Processor
  - 2X in speed every 1.5 years; 100X performance in last decade
- Memory
  - DRAM capacity: 2x / 2 years; 64X size in last decade
  - Cost per bit: improves about 25% per year
- Disk
  - capacity: > 2X in size every 1.0 years
  - Cost per bit: improves about 100% per year
  - 120X size in last decade

Computer Technology => Dramatic Change

- State-of-the-art PC when you graduate:
  - Processor clock speed: 4000 MegaHertz (4.0 GigaHertz)
  - Memory capacity: 1000 MegaByte (1.0 GigaBytes)
  - Disk capacity: 1000 GigaBytes (1.0 TeraBytes)
  - New units! Mega => Giga, Giga => Tera

Why Study Machine Structures?

- CHANGE; It's exciting!; It has never been more exciting!
- It impacts every other aspect of electrical engineering and computer

Bionics: Sensors in latex fingers instantly register hot and cold, and an electronic interface in his artificial limb stimulates the nerve endings in his upper arm, which then pass the information to his brain. The $3,000 system allows his hand to feel pressure and weight, so for the first time since losing his arms in a 1986 accident, he can pick up a can of soda without crushing it or having it slip through his fingers. *One Digital Day*

CS61C: So what's in it for me?

- Machine structures from a programmer's view
  - What the programmer writes
  - How it is converted to something the computer understands
  - How the computer interprets the program
  - What makes programs go slow

CS61C: So what's in it for me?

- Learn big ideas in CS and engineering
  - 5 Classic components of a Computer
  - Data can be anything (integers, floating point, characters): a program determines what it is
  - Stored program concept: instructions just data
  - Principle of Locality, exploited via a memory hierarchy (cache)
  - Greater performance by exploiting parallelism
  - Principle of abstraction, used to build systems as layers
  - Compilation v. interpretation thru system layers
  - Principles/Pitfalls of Performance Measurement

What 61C is not

- Learning C
  - If you know one, you should be able to learn another programming language on your own
  - Given that you know Java, should be easy to pick up its ancestor, C

C → C++ → Java

- Assembly Language Programming
  - This is a skill you will pick up, as a side effect of understanding the Big Ideas
- Hardware design
  - Hardware at abstract level, with only a little bit of physical logic to give things perspective
  - CS 150, 152 teach this

CS61B Prerequisite

- Students who have not taken 61B:
  - Will be dropped from class if enrolled or not promoted from wait list
- If you have taken 61B or the equivalent and you are on the list:
  - See Michael-David Sasson, 379 Soda, 643-6002, msasson@cs to straighten things out
- 61B Fall Semester meets in the same room, so it can easily add 100 people; more sections will be added as needed

Course Lecture Outline (COD chapters)

- 1 week: Computer Anatomy (Ch. 1)
- 4 weeks: C v. ASM languages (Ch. 3)
- 1.5 weeks:C v. ASM numbers (Ch. 4)
- 1.5 weeks: on I/O and interrupts (8)
- 1 week on Cache (COD Ch. 7)
- 1 week on Virtual Memory (Ch. 7)
- 2 weeks Processor Datapath, Pipelining (COD 5.1,6.1)
- 2 weeks on review of difficult topics (pointers, caches, interrupts)

Course Exams

- Reduce the pressure of taking exams
  - Midterm: Wednesday October 25
  - 3 hrs to take 1.5-hr test (5-8 PM, 1 Pimentel)
  - Our goal: test knowledge vs. speed writing
  - Review meetings: Sunday before
  - Can bring 1 page summary sheet
- Final: Wednesday December 12 (5-8 PM, 1 Pimentel)

Homework Assignments, Labs and Projects

- Lab exercises are to be done every week in lab section, and checked off by your lab TA or turned in at beginning of lab
- Homework exercises are to be handed in either online or to homework boxes in 283 Soda, due on Mondays at noon; teams 2-3 with 2nd exercise
  - 1st assignment: COD Exercises 1.1-1.16, 1.18, 1.21-1.23, 1.25, 1.27-1.30, 1.34-1.41,1.43-1.44, 1.56; Due Tuesday 9/5 noon
- Projects are larger programming assignments; individual and team

Homework Assignments, Labs and Project

- Must turn in survey, login and attend lab/discussion sections to be considered enrolled
  - Go to old and new sections to ask TA's to switch sections

Grading

- Grade breakdown
  - Midterm Exam: 25%
  - Final Exam: 35%
  - Homework Assignments 11%
  - Lab Exercises 11%
  - Projects 18%
- Scores posted on home page
  - Written/email request for changes to grades; work first with TA
  - Dec 4 deadline to correct online scores

Course Problems

- Can't make midterm, final
  - Tell early us and we will schedule alternate time before exam
- Forgot to turn in homework/ Dog ate computer
  - As a result of feedback, going to grade almost immediately so that can give results back quickly => late a hassle
- Get 2 days per semester to use up in emergencies; can use 1 day at a time

Course Problems

- What is cheating?
  - Studying together in groups is encouraged
  - Work must be your own
  - Common examples of cheating: running out of time on a assignment and then pick up output, take homework from box and copy, person asks to borrow solution "just to take a look", copying an exam question, ...
  - Better off to skip assignment (11 homeworks, 11 labs, 6 projects 40% of grade; how much can one assignment mean?)

Class decides on penalties for cheating

- Exercises (book):
  - 0 for problem
  - 0 for assignment
  - subtract full value for assignment
  - subtract 2X full value for assignment
- Labs (groups: only penalize individuals?)
  - 0 for problem
  - 0 for assignment
  - subtract full value for assignment
  - subtract 2X full value for assignment

Class decides on penalties; staff enforces

- Projects (groups: only penalize individuals?)
  - 0 for problem
  - 0 for assignment
  - subtract full value for assignment
  - subtract 2X full value for assignment
- Exams
  - 0 for problem
  - 0 for exam

Course Administration

- Instructor: David A. Patterson
  (patterson@cs) 635 Soda Hall
  Office Hours: Wed 1-2
- Labs: Class Accounts for 271 Soda
- Materials: http://www-inst.eecs/~cs61c
- Newsgroup: ucb.class.cs61c
- Text: *Computer Organization and Design: The Hardware/Software Interface, Second Edition*, Patterson and Hennessy

Typical Lecture Format

- 20-Minute Lecture
- 5-Minute Administrative Matters
- 25-Minute Lecture
- Instructor will come to class early & stay after to answer questions

And in Conclusion...

- 15 weeks to learn big ideas in CS&E
  - Principle of abstraction, used to build systems as layers
  - Pliable Data: a program determines what it is
  - Stored program concept: instructions are just data
  - Principle of Locality, exploited via a memory hierarchy (cache)
  - Greater performance by exploiting parallelism (pipeline)
  - Compilation v. interpretation to move down layers of system
  - Principles/Pitfalls of Performance Measurement

And in Conclusion...

- Continued rapid improvement in Computing
  - 2X every 1.5 years in processor speed; every 2.0 years in memory size; every 1.0 year in disk capacity; Moore's Law enables processor, memory (2X transistors/chip/ ~1.5 yrs)
- 5 classic components of all computers