Lecture #1 – Introduction
2010-01-20
There are two handouts today at the front and back of the room!

Protests worked! ⇒ “Choosing UCs over prisons ... this is a historic and transforming realignment of California’s priorities” ... “The protests @ UCs were the tipping point ... our univ system is going to get the support it deserves”


“I stand on the shoulders of giants...”
Thanks to these talented folks (& many others) whose contributions have helped make CS61C a really tremendous course!

Where does CS61C fit in?

Are Computers Smart?

• To a programmer:
  • Very complex operations / functions:
    - (map (lambda (x) (* x x)) '(1 2 3 4))
  • Automatic memory management:
    - List l = new List;
  • “Basic” structures:
    - Integers, floats, strings, simple commands

Are Computers Smart?

• In real life at the lowest level:
  • Only a handful of operations:
    - {and, or, not}
  • No automatic memory management.
  • At the lowest level, only 2 values:
    - {0, 1} or {low, high} or {off, on}

Computers are dumb!

What are “Machine Structures”?

Coordination of many levels (layers) of abstraction
Overview of Physical Implementations

The hardware out of which we make systems.

• Integrated Circuits (ICs)
  • Combinational logic circuits, memory elements, analog interfaces.

• Printed Circuits (PC) boards
  • Substrate for ICs and interconnection, distribution of CLK, Vdd, and GND signals, heat dissipation.

• Power Supplies
  • Converts line AC voltage to regulated DC low voltage levels.

• Chassis (rack, card case, ...)
  • Holds boards, power supply, provides physical interface to user or other systems.

Connectors and Cables.

Integrated Circuits (2009 state-of-the-art)

• Primarily Crystalline Silicon
  • 1mm - 25mm on a side
  • 2009 feature size ~ 45 nm = 45 x 10^{-9} m (then 32, 22, and 16 [by yr 2013])
  • 100 - 1000M transistors
  • (25 - 100M “logic gates”)  
  • 3 - 10 conductive layers
  • “CMOS” (complementary metal oxide semiconductor) - most common.

• Package provides:
  • Spreading of chip-level signal paths to board-level
  • Heat dissipation.
  • Ceramic or plastic with gold wires.

Printed Circuit Boards

• fiberglass or ceramic
• 1-20 conductive layers
• 1-20 in on a side
• IC packages are soldered down.

• Provides:
  • Mechanical support
  • Distribution of power and heat.

Moore’s Law

Predicts: 2X Transistors / chip every 2 years

Gordon Moore
Intel Cofounder
B.S. Cal 1950!

en.wikipedia.org/wiki/Moore%27s_law
Technology Trends: Uniprocessor Performance (SPECint)

- VAX: 1.25x/year 1978 to 1986
- RISC + x86: 1.52x/year 1986 to 2002
- RISC + x86: 1.20x/year 2002 to present

"Sea change" in chip design: multiple "cores" or processors per chip

Performance (vs. VAX-11/780)

3X

Computer Technology - Growth!

You just learned the difference between (Kilo, Mega, ... ) and (Kibi, Mebi, ... )!

- Processor
  - Speed 2x / 1.5 years (since '85) [slowing!]
  - 100X performance last decade
  - When you graduate: 4 GHz, 32 Cores
- Memory (DRAM)
  - Capacity: 2x / 2 years (since '96)
  - 64x size last decade.
  - When you graduate: 128 GibiBytes
- Disk
  - Capacity: 2x / 1 year (since '97)
  - 250X size last decade.
  - When you graduate: 8 TeraBytes

...Not nec all on one disk

Computer Technology - Growth!

- VAX: 1.25x/year 1978 to 1986
- RISC + x86: 1.52x/year 1986 to 2002
- RISC + x86: 1.20x/year 2002 to present

"Sea change" in chip design: multiple "cores" or processors per chip

Performance (vs. VAX-11/780)

3X

CS61C: So, what's in it for me?

- Learn some of the big ideas in CS & Engineering:
  - Principle of abstraction
    - Used to build systems as layers
  - 5 Classic components of a Computer
  - Data can be anything
    - Integers, floating point, characters, ...
    - A program determines what it is
  - Principle of Locality
    - Exploited via a memory hierarchy (cache)
  - Greater performance by exploiting parallelism
  - Compilation v. interpretation through system layers
  - Principles / Pitfalls of Performance Measurement

Others Skills learned in 61C

- Learning C
  - If you know one, you should be able to learn another programming language largely on your own
  - If you know C++ or Java, it should be easy to pick up their ancestor, C

- Assembly Language Programming
  - This is a skill you will pick up, as a side effect of understanding the Big Ideas

- Hardware design
  - We'll learn just the basics of hardware design
  - CS 150, 152 teach this in more detail

Yoda says...

"Always in motion is the future..."

Our schedule may change slightly depending on some factors. This includes lectures, assignments & labs...
Tried-and-True Technique: Peer Instruction

- Increase real-time learning in lecture, test understanding of concepts vs. details
- As complete a “segment” ask multiple choice question
  - 1-2 minutes to decide yourself
  - 2 minutes in pairs/triples to reach consensus. Teach others!
  - 2 minute discussion of answers, questions, clarifications
- You’ll get transmitters from ASUC bookstore...

Extra Credit: EPA!

- **Effort**
  - Attending Dan’s and TA’s office hours, completing all assignments, turning in HW0, doing reading quizzes
- **Participation**
  - Attending lecture and voting using the PRS system
  - Asking great questions in discussion and lecture and making it more interactive
- **Altruism**
  - Helping others in lab or on the newsgroup
- EPA! extra credit points have the potential to bump students up to the next grade level! (but actual EPA! scores are internal)

Course Problems…Cheating

- What is cheating?
  - Studying together in groups is encouraged.
  - Turned-in work must be completely 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, ...
  - You’re not allowed to work on homework/projects/exams with anyone (other than ask Qs walking out of lecture)
  - Both “giver” and “receiver” are equally culpable
- Cheating points: 0 EPA, negative points for that assignment / project / exam (e.g., if it’s worth 10 pts, you get -10) in most cases, F in the course.
- Every offense will be referred to the Office of Student Judicial Affairs. [www.eecs.berkeley.edu/Policies/acad.dis.shtml](http://www.eecs.berkeley.edu/Policies/acad.dis.shtml)

My goal as an instructor

- To make your experience in CS61C as enjoyable & informative as possible
  - Humor, enthusiasm, graphics & technology-in-the-news in lecture
  - Fun, challenging projects & HW
  - Pro-student policies (exam clobbering)
- To maintain Cal & EECS standards of excellence
  - Your projects & exams will be just as rigorous as every year. Overall: B- avg
- To be an HKN “7.0” man
  - I know I speak fast when I get excited about material. I’m told every semester.
  - Please give me feedback so I improve! Why am I not 7.0 for you? I will listen!!

Summary

- Continued rapid improvement in computing
  - 2X every 2.0 years in memory size; every 1.5 years in processor speed; every 1.0 year in disk capacity;
  - Moore’s Law enables processor (2X transistors/chip ~1.5-2 yrs)
- 5 classic components of all computers
  - Control Datapath Memory Input Output
  - Processor

Teaching Assistants

- Scott Beamer (also Head TA)
- Eric Chang
- Michael Greenbaum
- Long Wei
- Bing Xia
Reference slides

You ARE responsible for the material on these slides (they're just taken from the reading anyway) -- we've moved them to the end and off-stage to give more breathing room to lecture!

Course Lecture Outline

- Basics
  - C-Language, Pointers
  - Memory management
- Machine Representations
  - Numbers (integers, reals)
  - Assembly Programming
  - Compilation, Assembly
- Processors & Hardware
  - Logic Circuit Design
  - CPU organization
  - Pipelining
- Memory Organization
  - Caches
  - Virtual Memory
- I/O
  - Interrupts
  - Disks, Networks
- Advanced Topics
  - Performance
  - Virtualization
  - Parallel Programming

Homeworks, Labs and Projects

- Lab exercises (every wk; due in that lab session unless extension given by TA) -- extra point if you finish in 1st hour!
- Homework exercises (~ every week; HW 0 out now, due in section next week)
- Projects (every 2 to 3 weeks)
  - All exercises, reading, homeworks, projects on course web page
  - We will DROP your lowest HW, Lab!
  - Only one {HW, Project, Midterm} / week

2 Course Exams

- Midterm: around 8th week @ 7-10pm
  - Give 3 hours for 2 hour exam
  - One "review sheet" allowed
  - Review session Sun beforehand, time/place TBA
- Final: Mon 2010-05-14 @ 8-11am (group 17)
  - You can clobber your midterm grade!
  - (students always LOVE this...)

Your final grade

- Grading (could change before 1st midterm)
  - 15pts = 5% Labs
  - 30pts = 10% Homework
  - 60pts = 20% Projects
  - 75pts = 25% Midterm* [can be clobbered by Final]
  - 120pts = 40% Final
  - Extra credit for EPA. What's EPA?
- Grade distributions
  - Similar to CS61[AB], in the absolute scale.
  - Perfect score is 300 points. 10-20-10 for A+, A, A-
  - Similar for Bs and Cs (40 pts per letter-grade)
    ... C+, C, C-, B, F (No D+ or D- distinction)
  - Differs: No F will be given if all-but-one (hw, lab), all projects submitted and all exams taken
  - We'll "ooch" grades up but never down

Texts

  The third edition is also accepted.
- Required: The C Programming Language, Kernighan and Ritchie (K&R), 2nd edition
- Reading assignments on web page