inst.eecs.berkeley.edu/~cs61c

CS61C : Machine Structures

Lecture #1 – Introduction

2010-01-20

There are two handouts today at the front and back of the room!

Lecturer SOE Dan Garcia

www.cs.berkeley.edu/~ddgarcia

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"

www.nytimes.com/2010/01/07/us/07calif.html

"I stand on the shoulders of giants…"

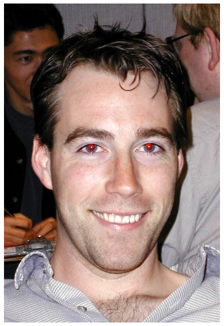
Prof
David Patterson

Prof
John Wawrznek

Lecturer SOE
Mike Clancy

Prof
David Culler

TA
Andy Carle



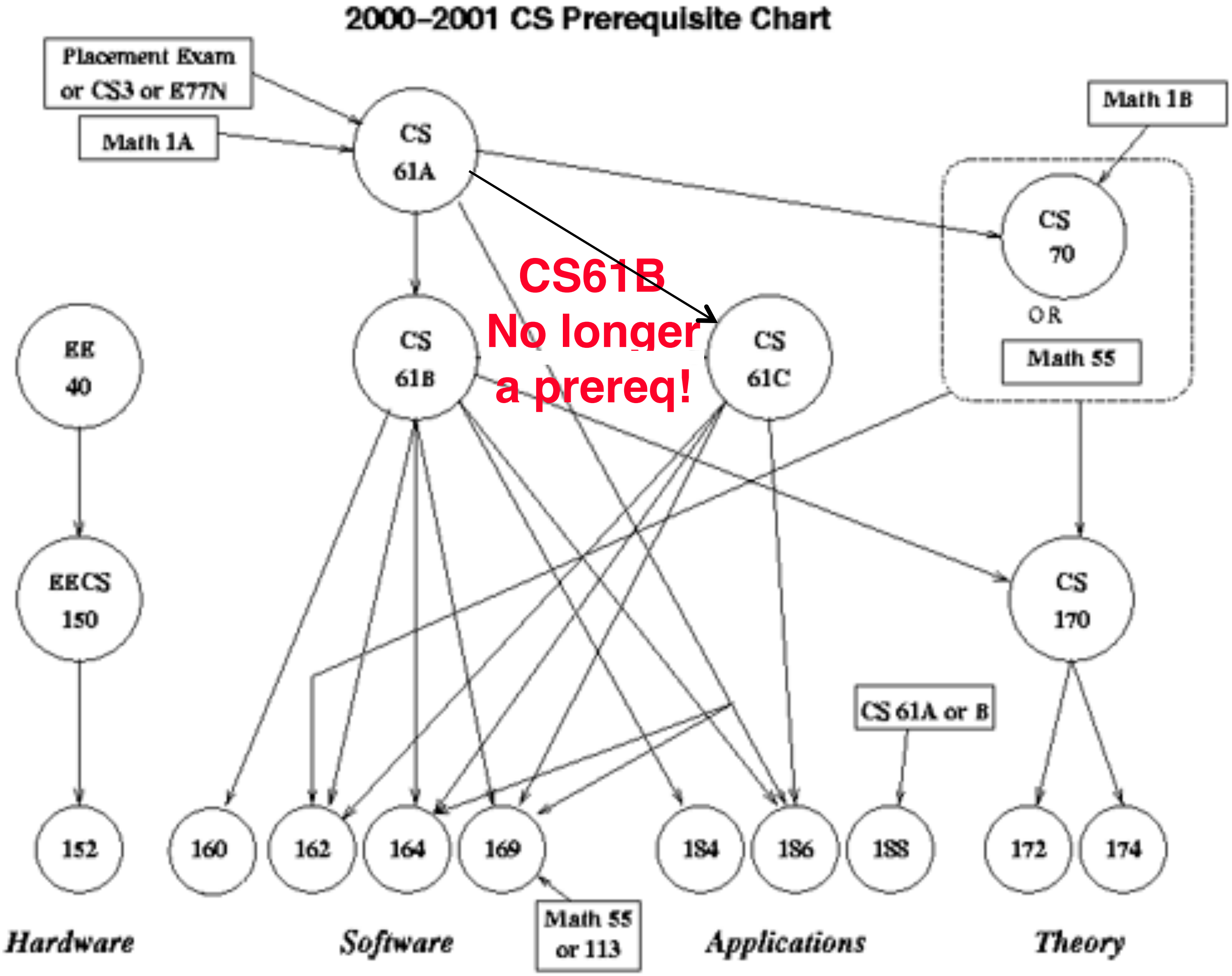
TA
Kurt Meinz

TA
David Jacobs

TA
Jeremy Huddleston

Thanks to these talented folks (& many others) whose contributions have helped make CS61C a really tremendous course!

Where does CS61C fit in?



CS61B
No longer
a prereq!

http://hkn.eecs.berkeley.edu/student/cs-prereq-chart1.gif

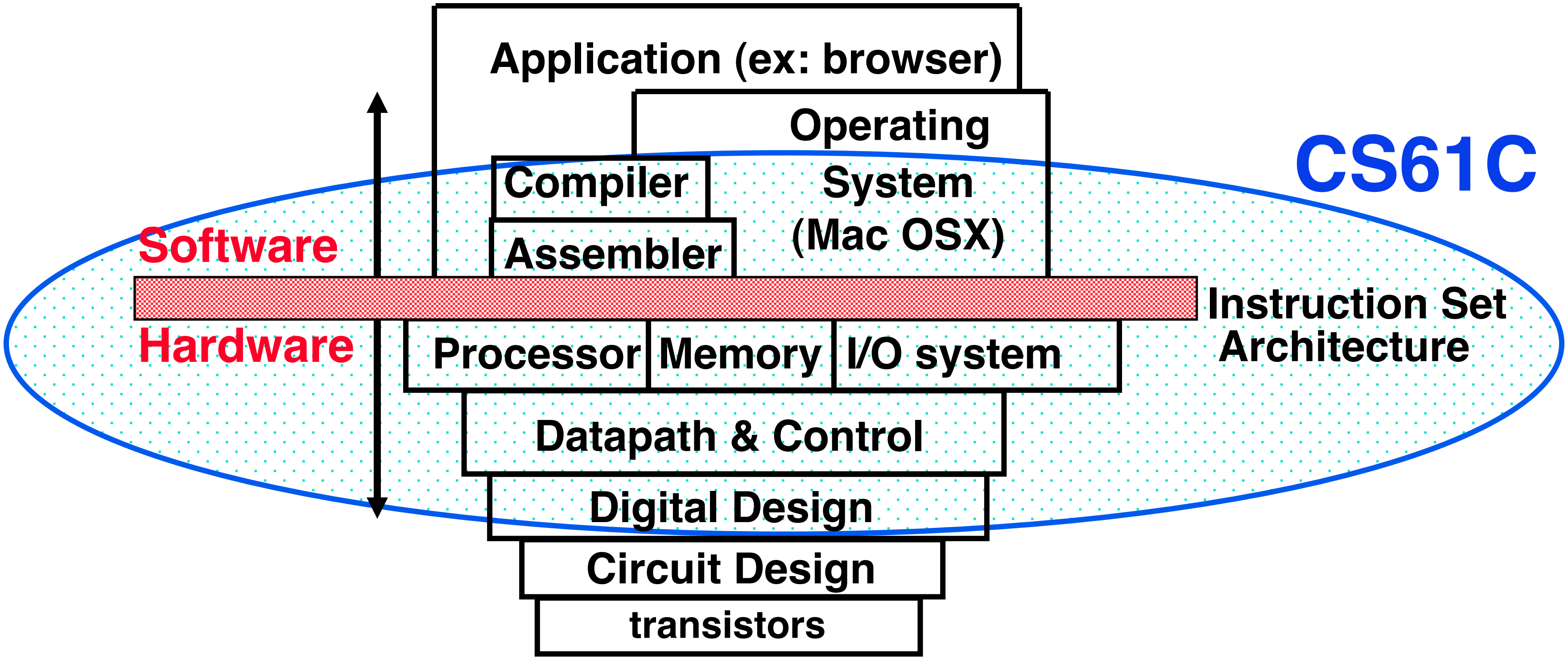
Are Computers Smart?

- 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?

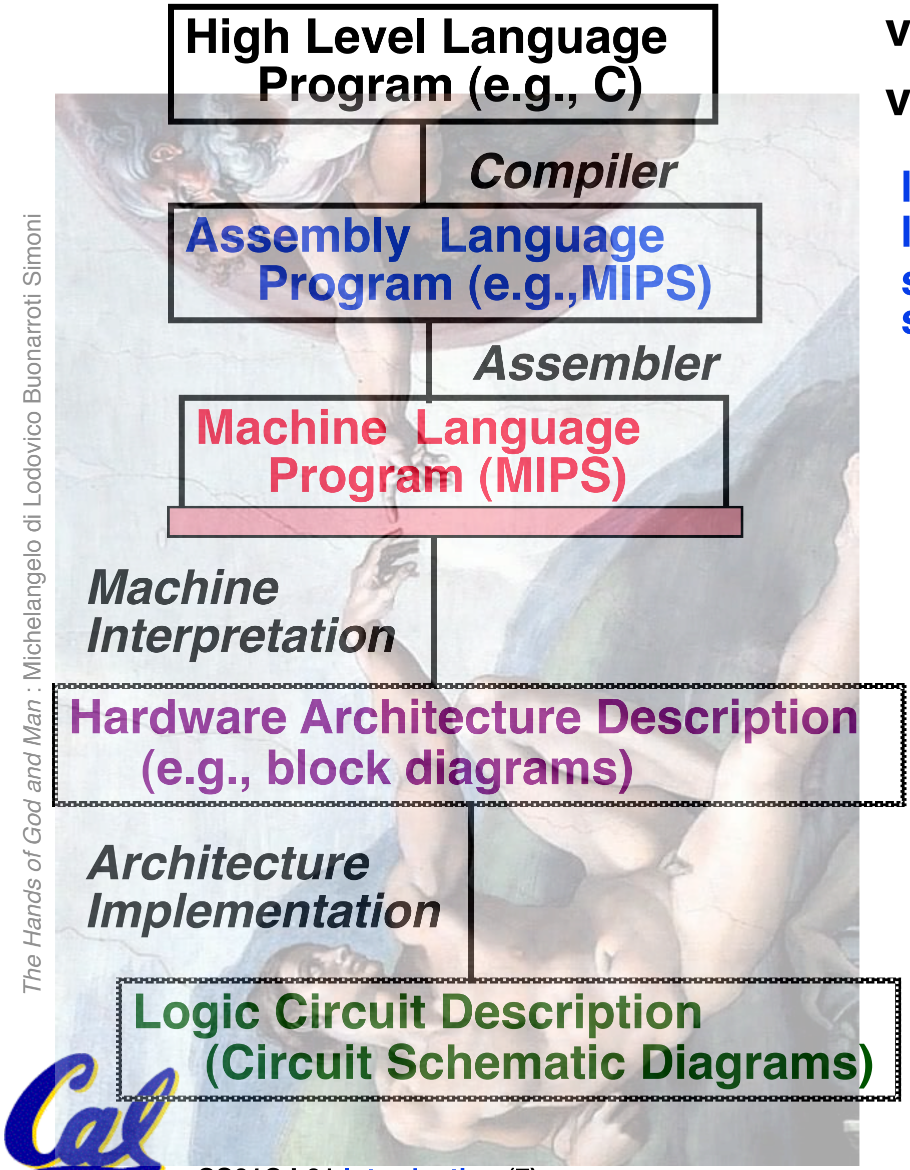
- In real life at the lowest level:
  - Only a handful of operations:
    - `{and, or, not}`
  - <u>No</u> automatic memory management.
  - At the lowest level, only 2 values:
    - {0, 1} or {low, high} or {off, on}

What are "Machine Structures"?



Coordination of many
levels (layers) of abstraction

CS61C Levels of Representation



High Level Language Program (e.g., C)

Compiler

Assembly Language Program (e.g.,MIPS)

Assembler

Machine Language Program (MIPS)

Machine Interpretation

Hardware Architecture Description (e.g., block diagrams)

Architecture Implementation

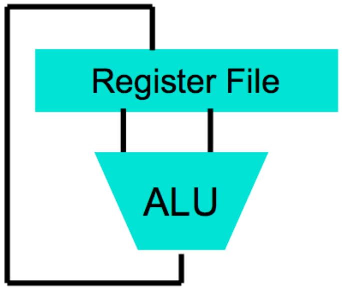
Logic Circuit Description (Circuit Schematic Diagrams)

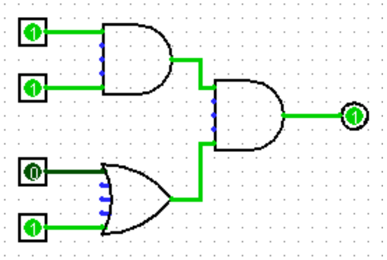
The Hands of God and Man : Michelangelo di Lodovico Buonarroti Simoni

```
temp = v[k];
v[k] = v[k+1];
v[k+1] = temp;
```

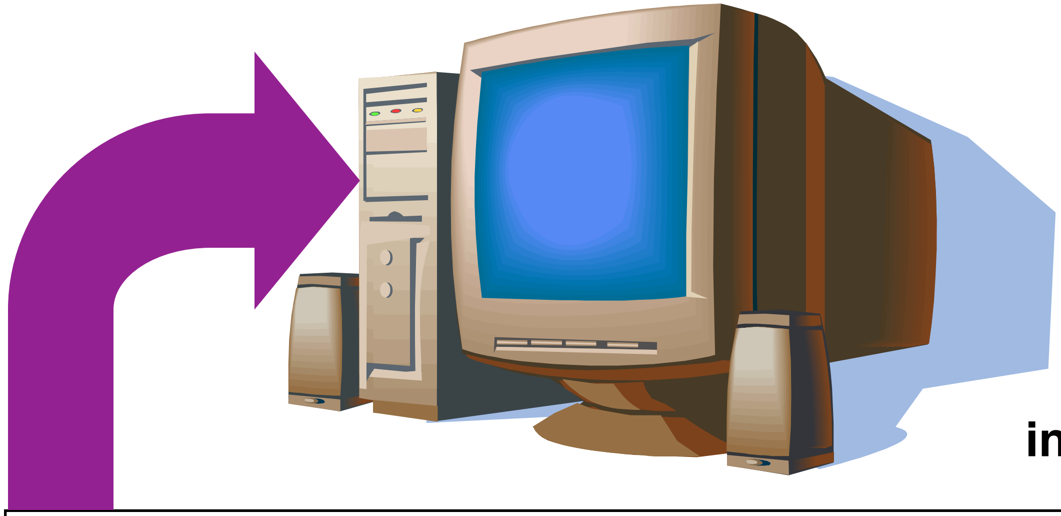
```
lw   $t0, 0($2)
lw   $t1, 4($2)
sw   $t1, 0($2)
sw   $t0, 4($2)
```

```
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
```

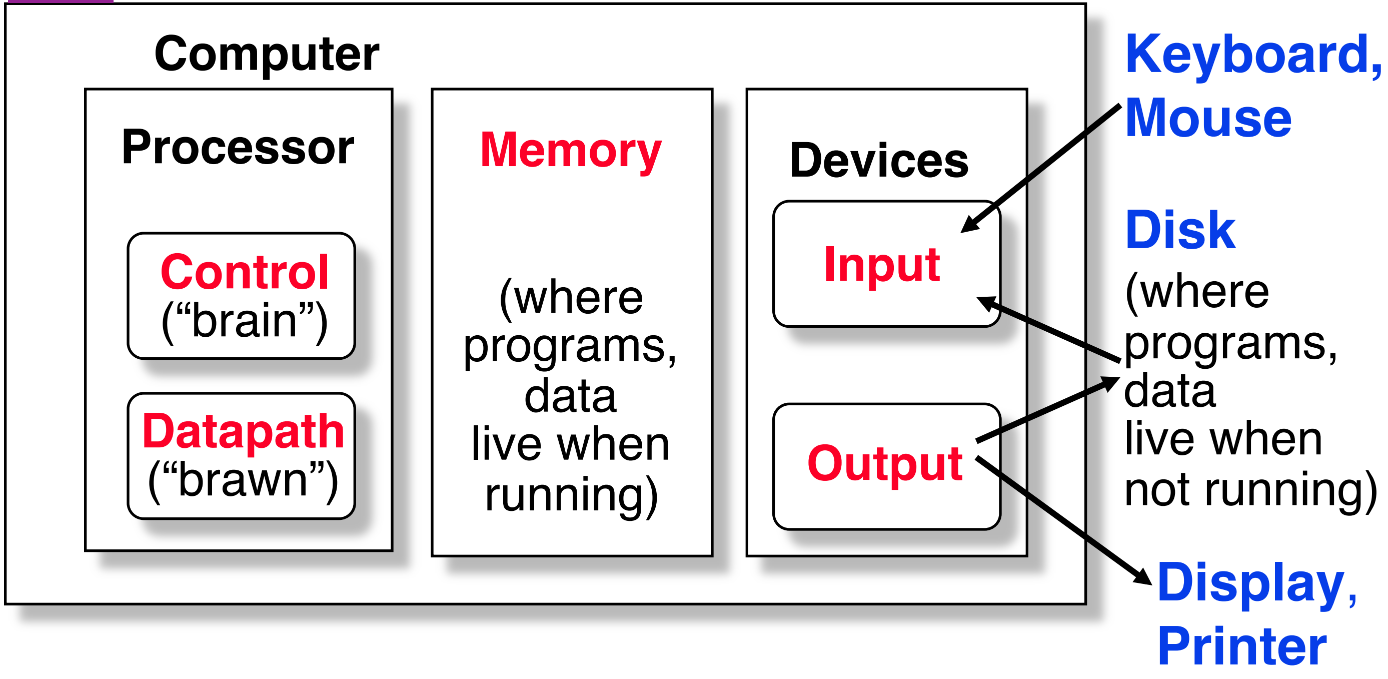




Anatomy: 5 components of any Computer



John von Neumann
invented this architecture



Computer

Processor

Control
("brain")

Datapath
("brawn")

Memory

(where programs, data live when running)

Devices

Input

Output

Keyboard, Mouse

Disk

(where programs, data live when not running)

Display, Printer

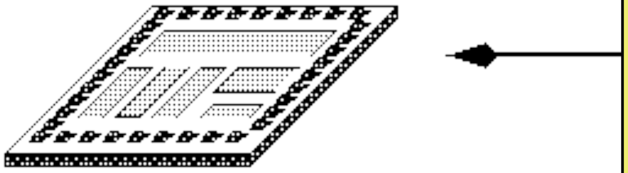
Overview of Physical Implementations

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.

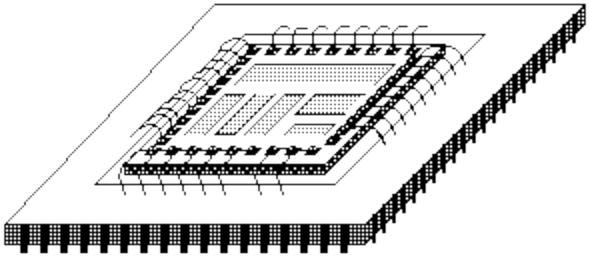
Integrated Circuits (2009 state-of-the-art)

Bare Die



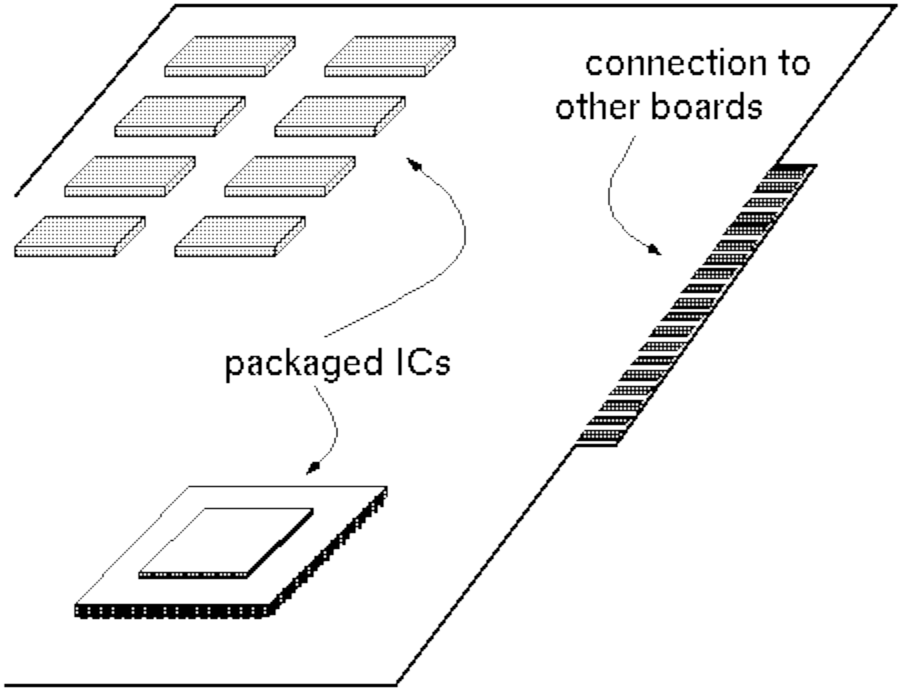
- Primarily Crystalline Silicon
- 1mm - 25mm on a side
- 2009 feature size ~ 45 nm = 45 x 10^{-9} m (then 32, 22, and 16 [by yr 2013])
- 100 - 1000M transistors
- (25 - 100M "logic gates")
- 3 - 10 conductive layers
- "CMOS" (complementary metal oxide semiconductor) - most common.

Chip in Package



- Package provides:
    - spreading of chip-level signal paths to board-level
    - heat dissipation.
- Ceramic or plastic with gold wires.

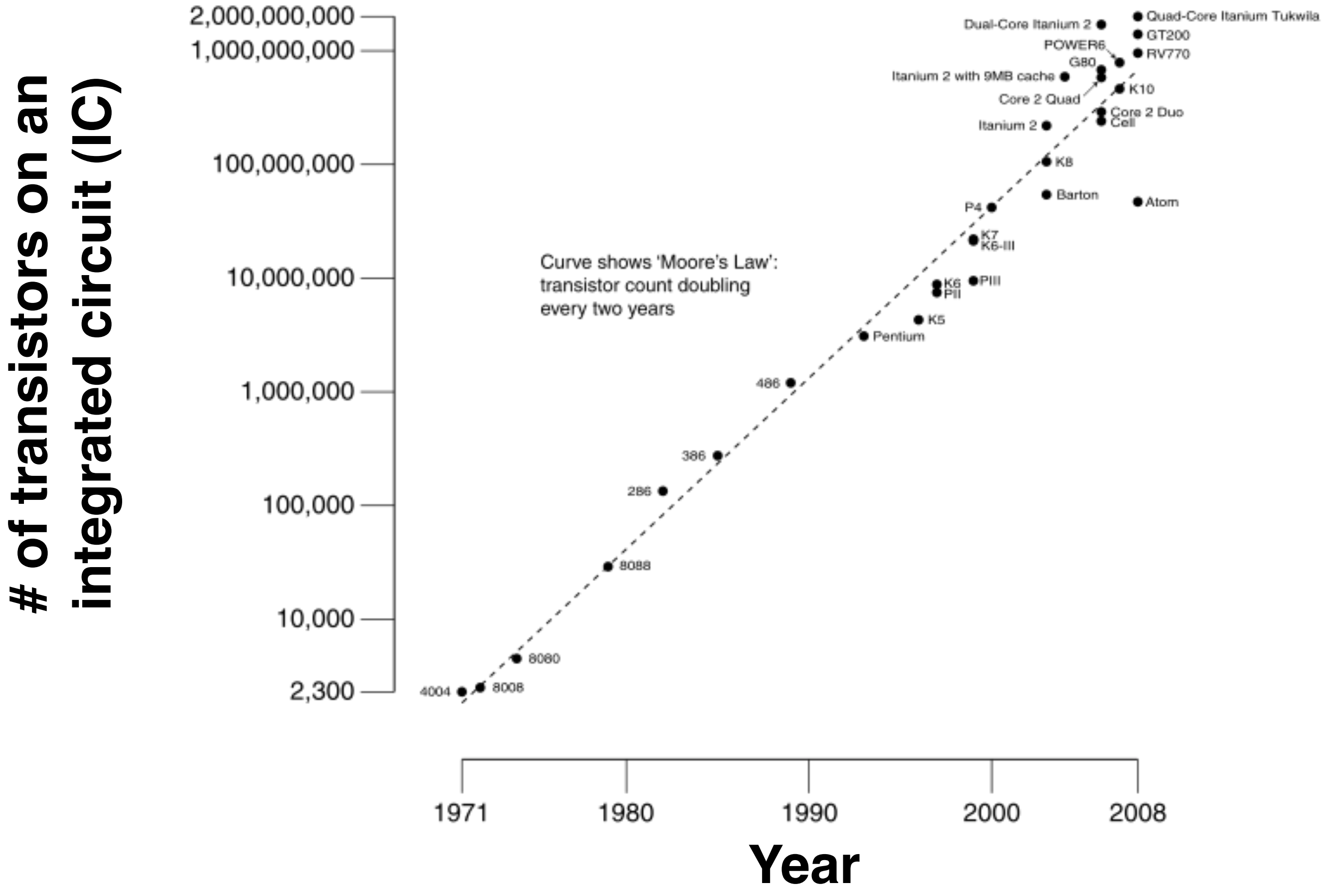
Printed Circuit Boards

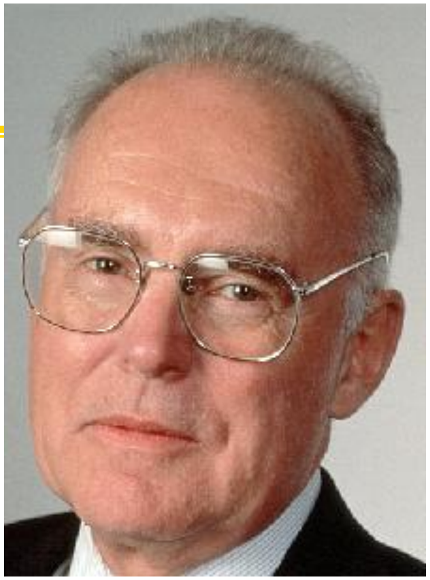


- fiberglass or ceramic
- 1-20 conductive layers
- 1-20 in on a side
- IC packages are soldered down.
- Provides:
  - Mechanical support
  - Distribution of power and heat.

Moore’s Law

Predicts: 2X Transistors / chip every 2 years

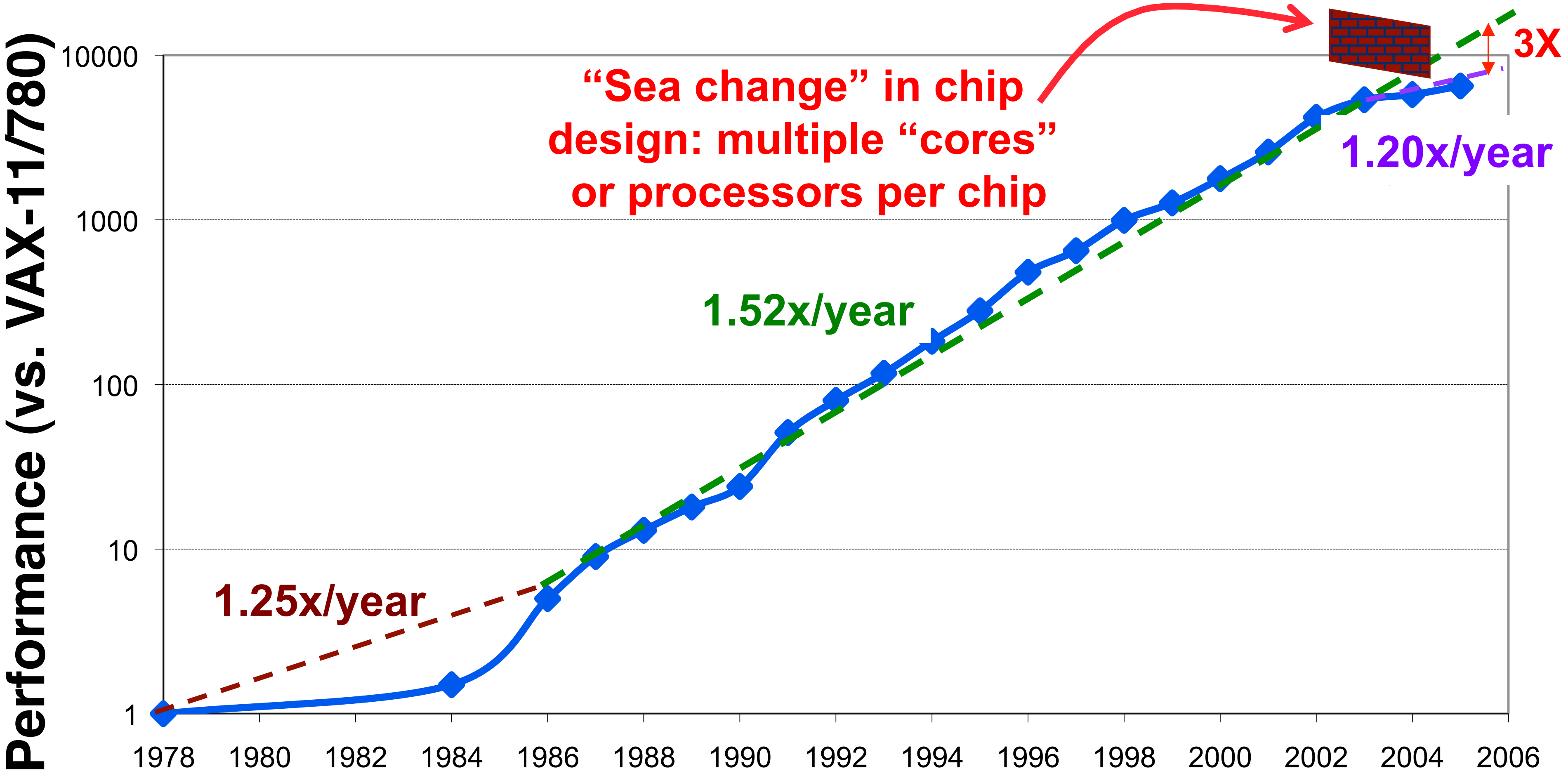




Gordon Moore
Intel Cofounder
B.S. Cal 1950!

`en.wikipedia.org/wiki/Moore's_law`

Technology Trends: Uniprocessor Performance (SPECint)



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

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

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

Kilo (10^3) & Kibi (2^{10})

↓

Mega (10^6) & Mebi (2^{20})

↓

Giga (10^9) & Gibi (2^{30})

↓

Tera (10^{12}) & Tebi (2^{40})

↓

Peta (10^{15}) & Pebi (2^{50})

↓

Exa (10^{18}) & Exbi (2^{60})

↓

Zetta (10^{21}) & Zebi (2^{70})

↓

Yotta (10^{24}) & Yobi (2^{80})

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
    - Stored program concept: instructions just data
  - 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

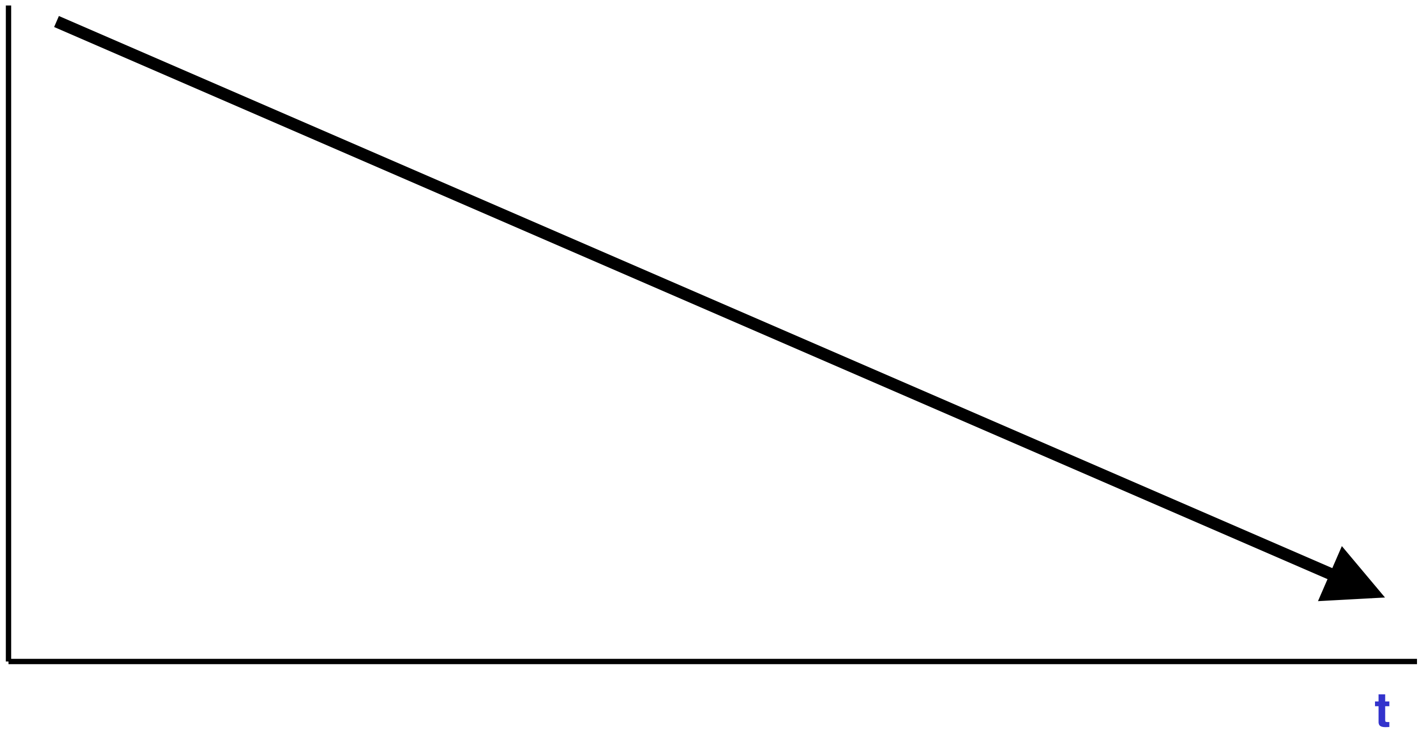
- Learning C
  - If you know one, you should be able to learn another programming language largely on your own
  - If you know C++ or Java, it should be easy to pick up their ancestor, C
- Assembly Language Programming
  - This is a skill you will pick up, as a side effect of understanding the Big Ideas
- 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…"

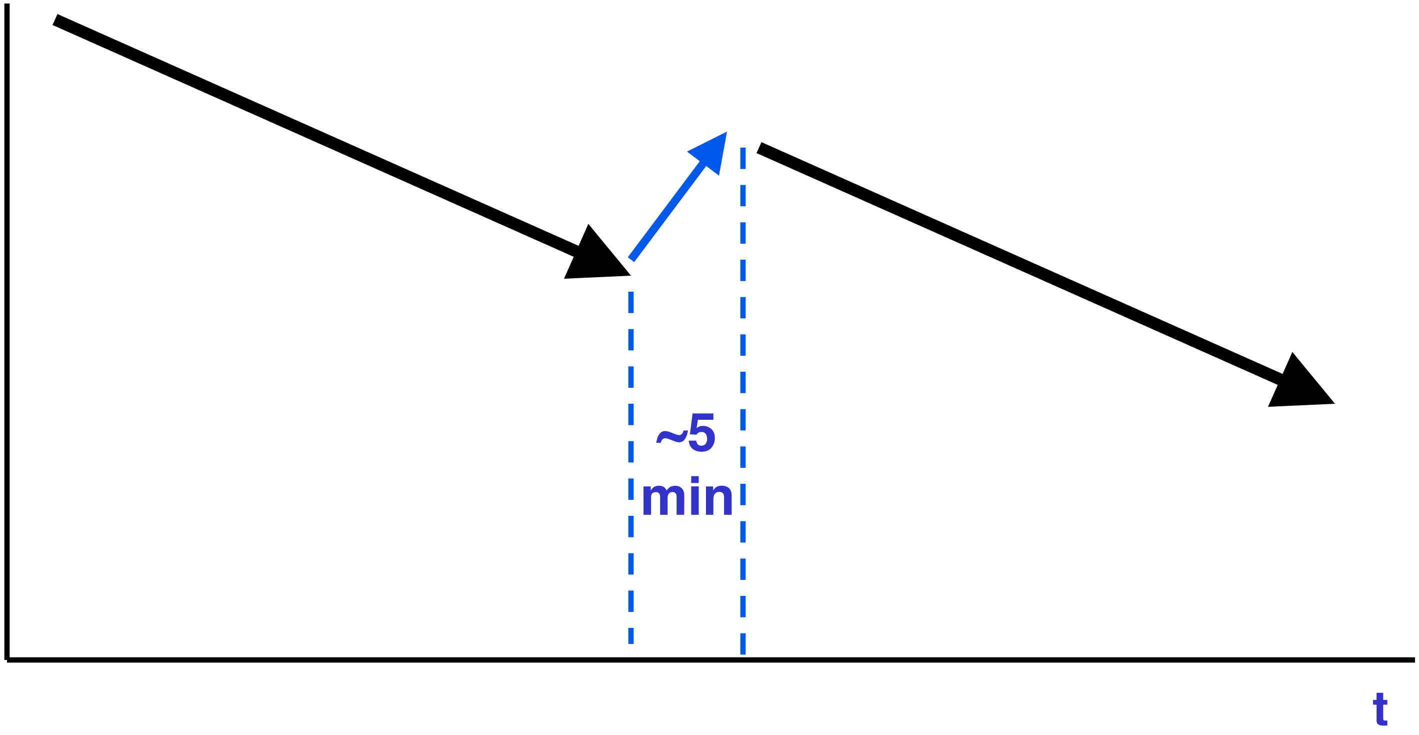
Our schedule may change slightly depending on some factors. This includes lectures, assignments & labs…

What is this?



Attention over time!

What is this?!



Attention over time!

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`

My goal as an instructor

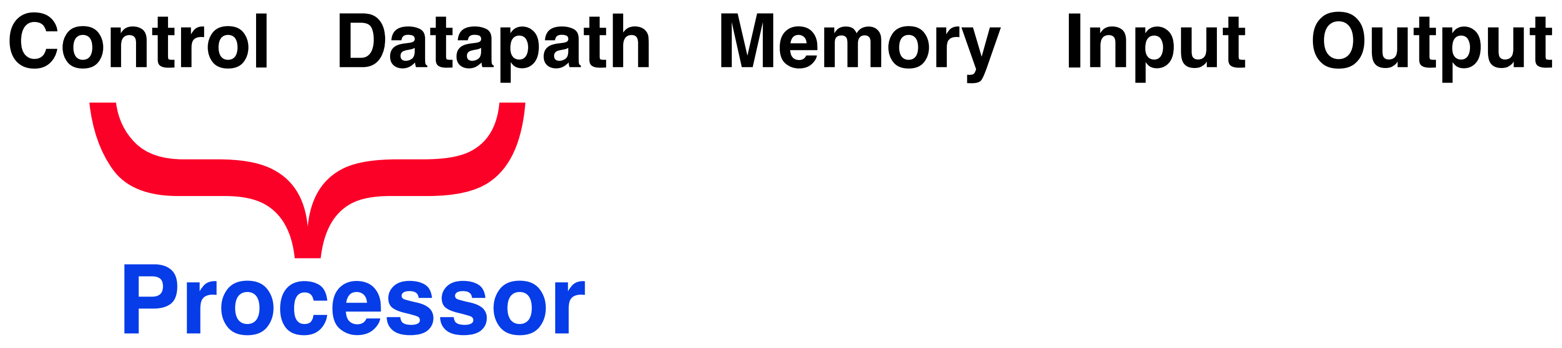
- To make your experience in CS61C as enjoyable & informative as possible
  - Humor, enthusiasm, graphics & technology-in-the-news in lecture
  - Fun, challenging projects & HW
  - Pro-student policies (exam clobbering)
- To maintain Cal & EECS standards of excellence
  - Your projects & exams will be 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. Help me slow down when I go toooo fast.
  - Please give me feedback so I improve! Why am I not 7.0 for you? I will listen!!

Teaching Assistants

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

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

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

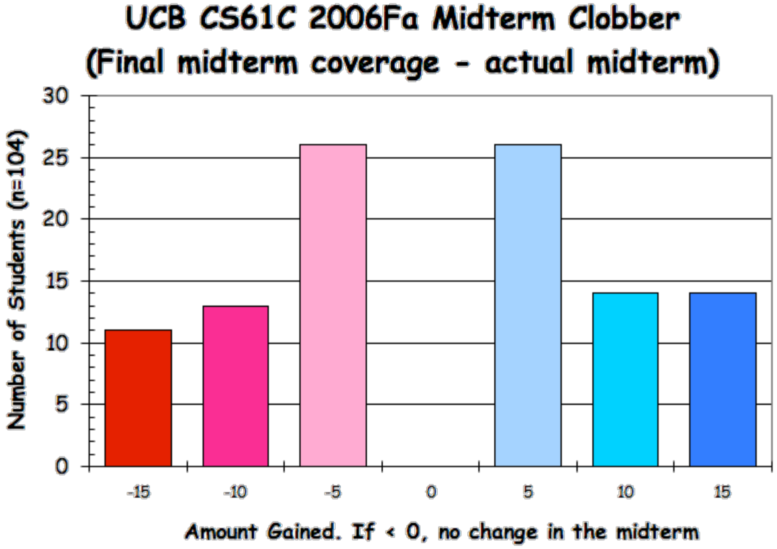
- Basics
  - C-Language, Pointers
  - Memory management
- Machine Representations
  - Numbers (integers, reals)
  - Assembly Programming
  - Compilation, Assembly
- Processors & Hardware
  - Logic Circuit Design
  - CPU organization
  - Pipelining
- Memory Organization
  - Caches
  - Virtual Memory
- 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

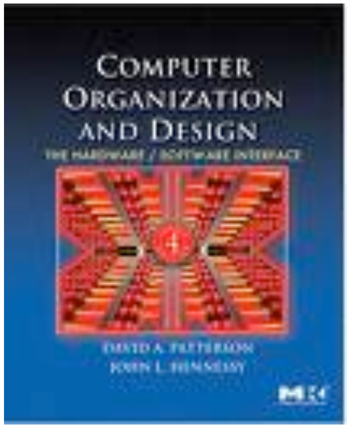
- Midterm: around 8th week @ 7-10pm
  - Give 3 hours for 2 hour exam
  - One "review sheet" allowed
  - Review session Sun beforehand, time/place TBA
- Final: Mon 2010-05-14 @ 8-11am (group 17)
  - You can *clobber* your midterm grade!
  - (students always LOVE this…)

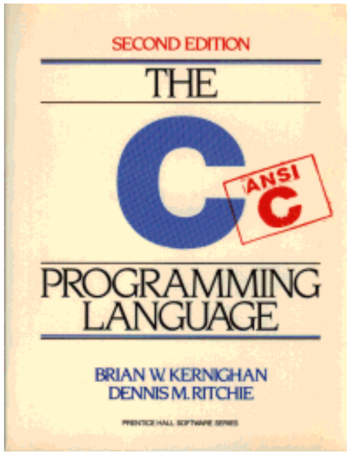


Your final grade

- 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-, D, 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





- Required: *Computer Organization and Design: The Hardware/Software Interface*, *Fourth Edition*, Patterson and Hennessy (COD).
  The third edition is also accepted.
- Required: *The C Programming Language*, Kernighan and Ritchie (K&R), 2nd edition
- Reading assignments on web page