CS 3410: Computer System Organization and Programming

Prof. Kavita Bala and Prof. Hakim Weatherspoon

CS 3410, Spring 2014

Computer Science

Cornell University

Course Objective

Bridge the gap between hardware and software

- How a processor works
- How a computer is organized

Establish a foundation for building higher-level applications

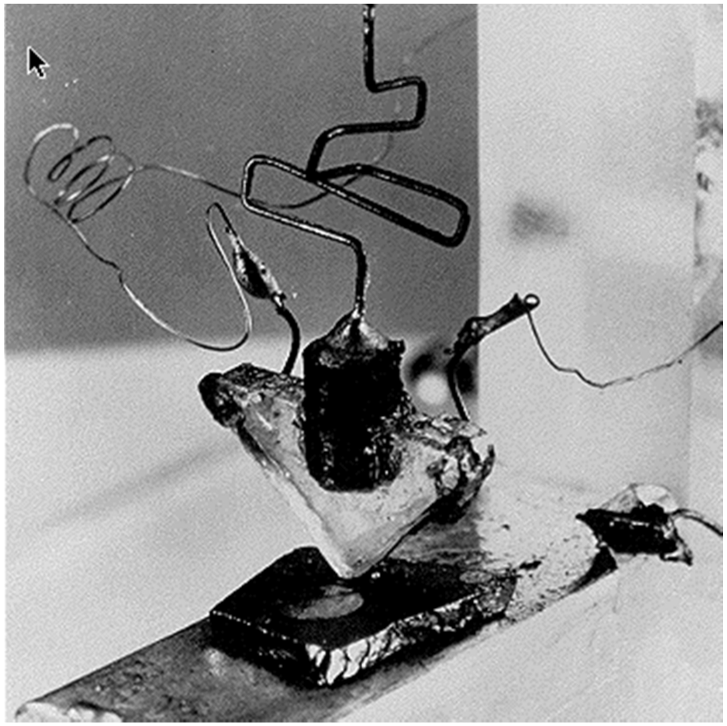
- How to understand program performance
- How to understand where the world is going

Where did it begin?

Electrical Switch

- On/Off
- Binary

Transistor



The first transistor on a workbench at AT&T Bell Labs in 1947

Moore’s Law

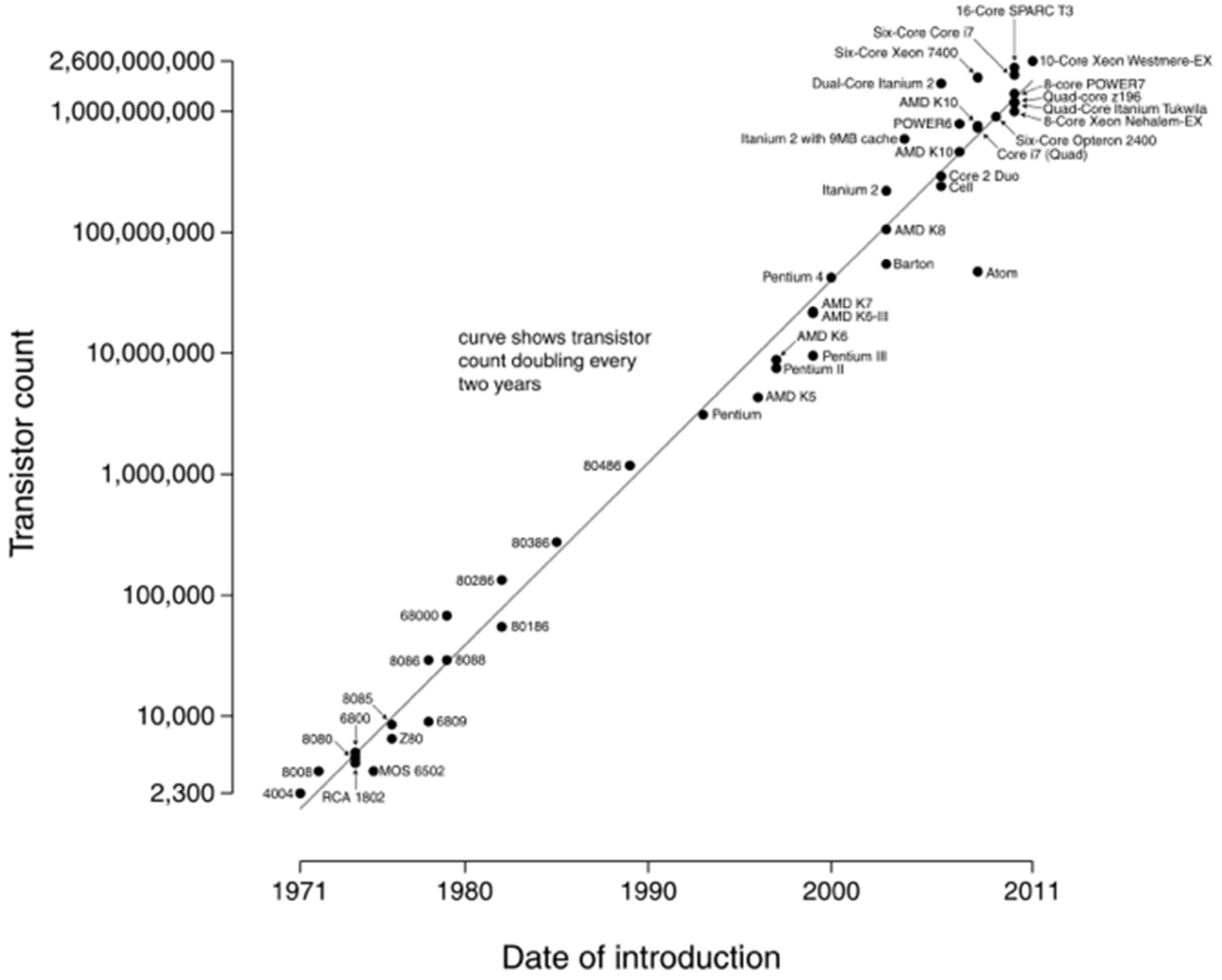
1965

- number of transistors that can be integrated on a die would double every 18 to 24 months (i.e., grow exponentially with time)

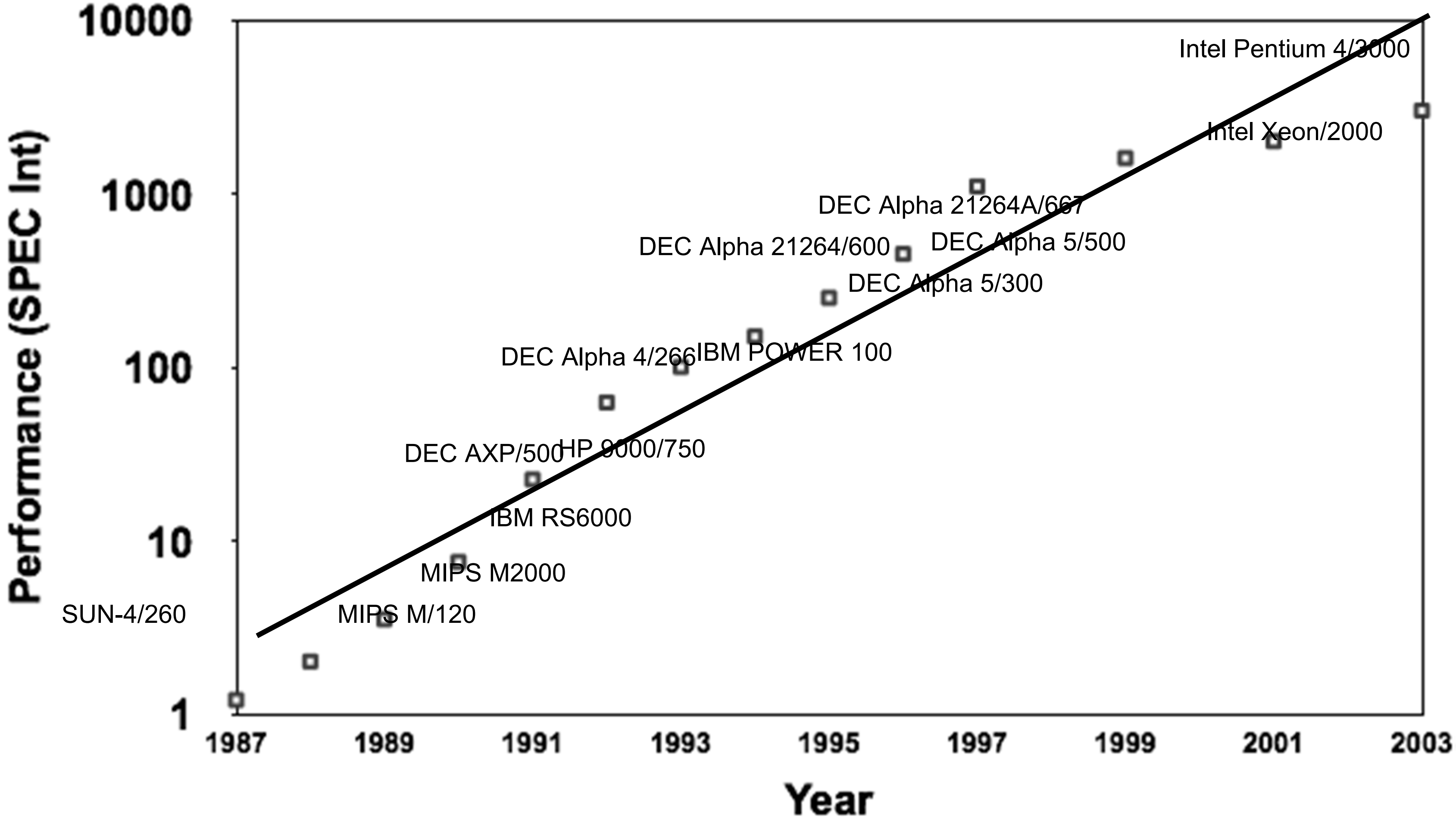
Amazingly visionary

- 2300 transistors, 1 MHz clock (Intel 4004) - 1971
- 16 Million transistors (Ultra Sparc III)
- 42 Million transistors, 2 GHz clock (Intel Xeon) – 2001
- 55 Million transistors, 3 GHz, 130nm technology, 250mm^2 die (Intel Pentium 4) – 2004
- 290+ Million transistors, 3 GHz (Intel Core 2 Duo) – 2007
- 721 Million transistors, 2 GHz (Nehalem) - 2009
- 1.4 Billion transistors, 3.4 GHz Intel Haswell (Quad core) – 2013

Microprocessor Transistor Counts 1971-2011 & Moore's Law



Processor Performance Increase



Moore’s Law

1965

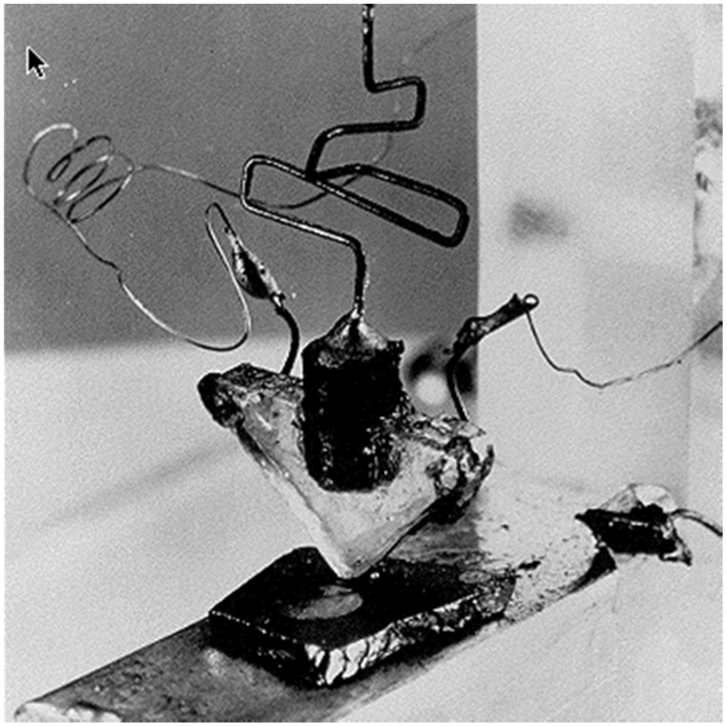
- number of transistors that can be integrated on a die would double every 18 to 24 months (i.e., grow exponentially with time)

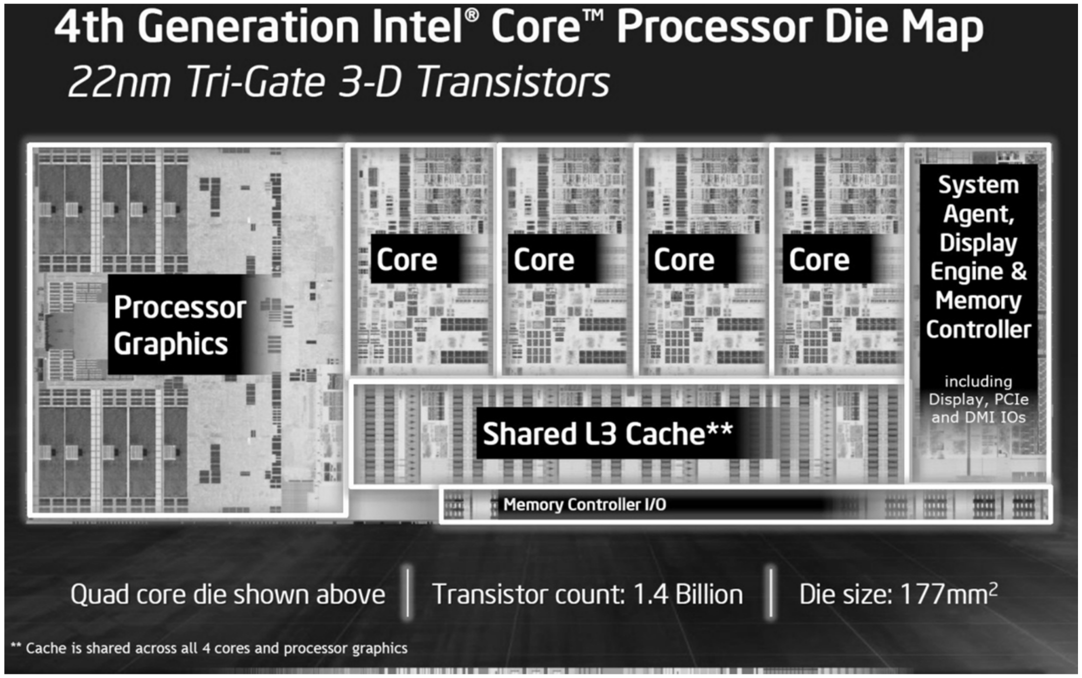
Amazingly visionary

- 2300 transistors, 1 MHz clock (Intel 4004) - 1971
- 16 Million transistors (Ultra Sparc III)
- 42 Million transistors, 2 GHz clock (Intel Xeon) – 2001
- 55 Million transistors, 3 GHz, 130nm technology, 250mm^2 die (Intel Pentium 4) – 2004
- 290+ Million transistors, 3 GHz (Intel Core 2 Duo) – 2007
- 721 Million transistors, 2 GHz (Nehalem) - 2009
- 1.4 Billion transistors, 3.4 GHz Intel Haswell (Quad core) – 2013

Parallelism

Then and Now





4th Generation Intel® Core™ Processor Die Map

22nm Tri-Gate 3-D Transistors

Processor Graphics
Core
Core
Core
Core
System Agent, Display Engine & Memory Controller
including Display, PCIe and DMI IOs
Shared L3 Cache**
Memory Controller I/O

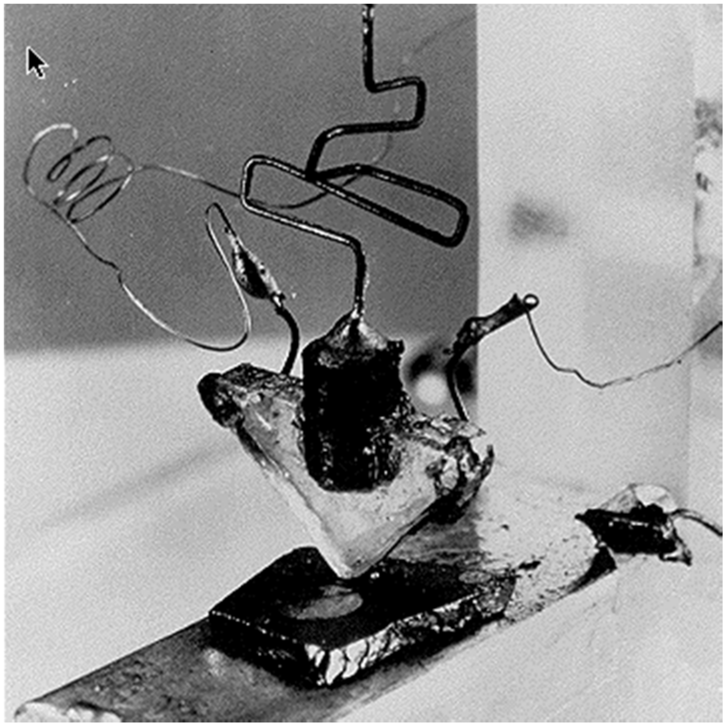
Quad core die shown above | Transistor count: 1.4 Billion | Die size: 177mm^2

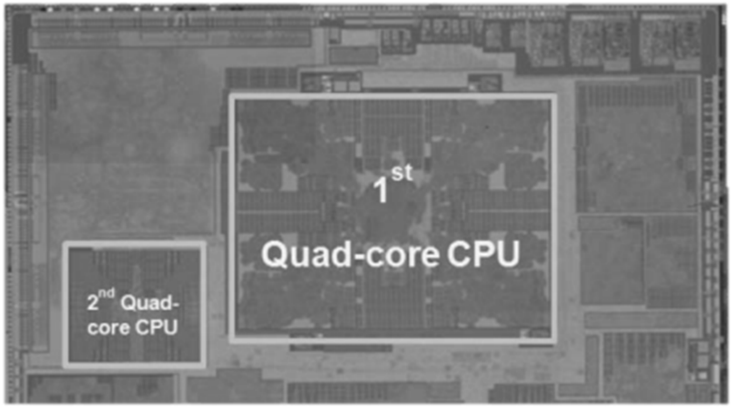
** Cache is shared across all 4 cores and processor graphics

http://techguru3d.com/4th-gen-intel-haswell-processors-architecture-and-lineup/

- The first transistor
  - One workbench at AT&T Bell Labs
  - 1947
  - Bardeen, Brattain, and Shockley
- An Intel Haswell
  - 1.4 billion transistors
  - 177 square millimeters
  - Four processing cores

Then and Now



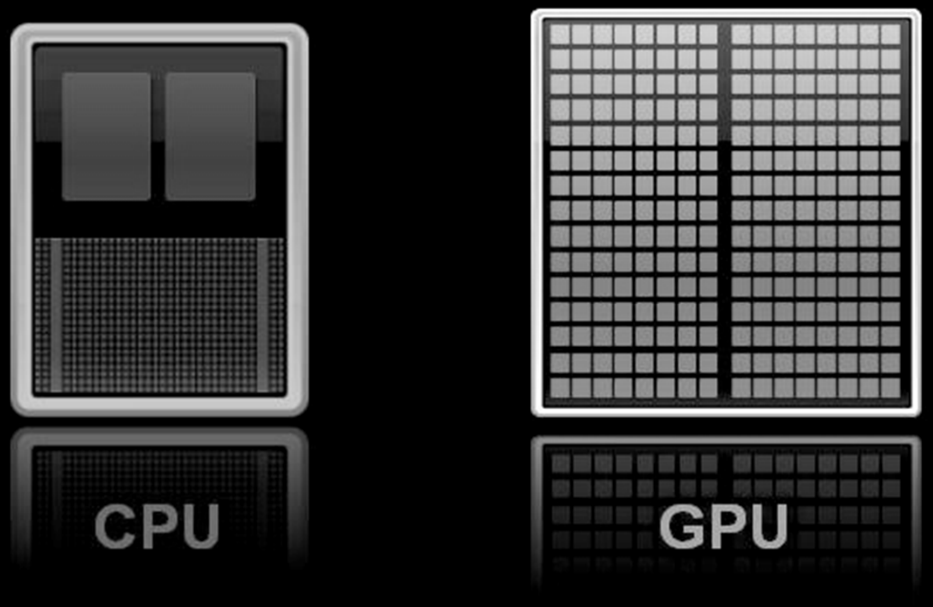


- The first transistor
  - One workbench at AT&T Bell Labs
  - 1947
  - Bardeen, Brattain, and Shockley
- Galaxy Note 3
  - 8 processing cores

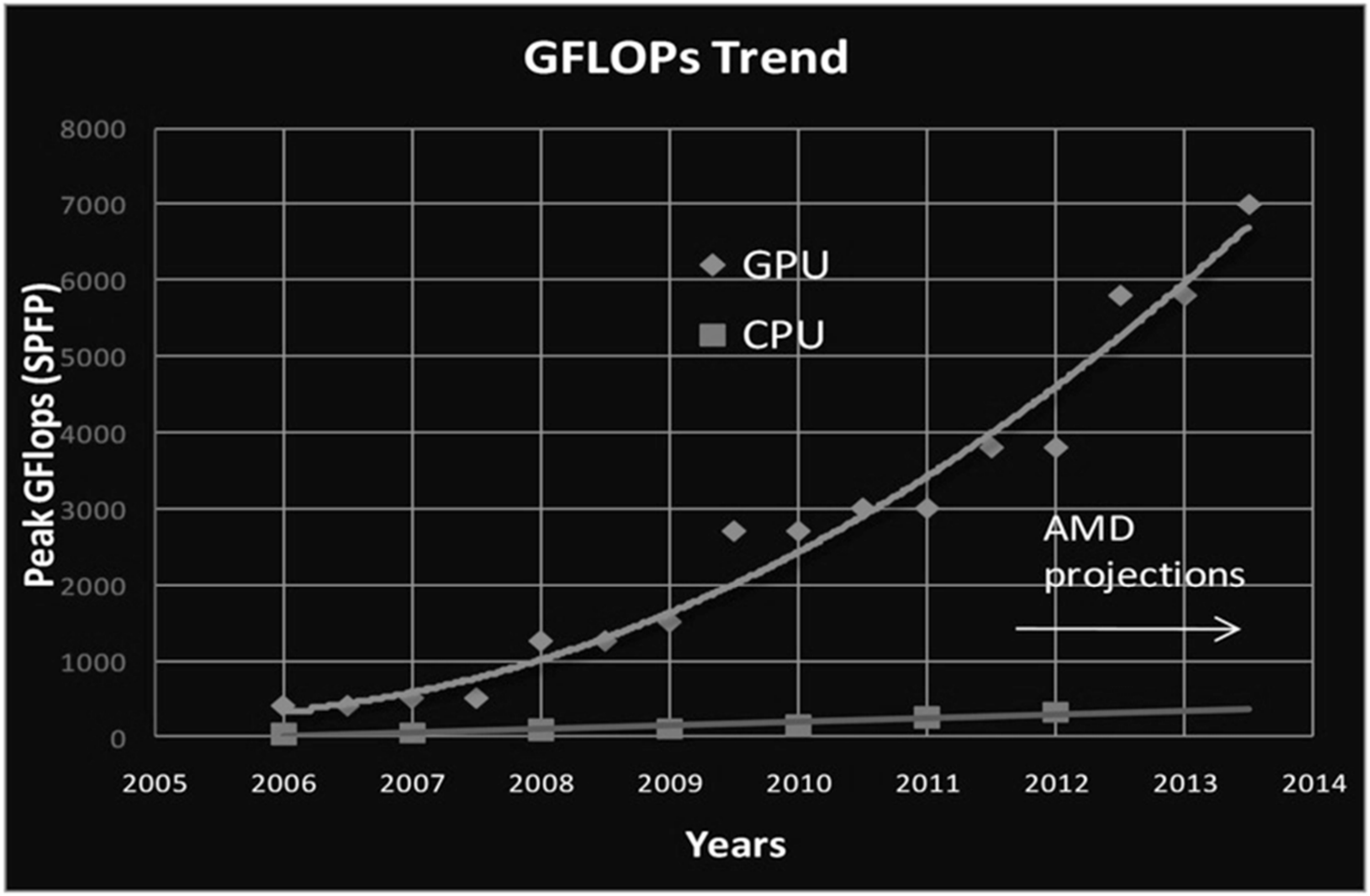
Parallelism

Central Processing Unit

Graphics Processing Unit



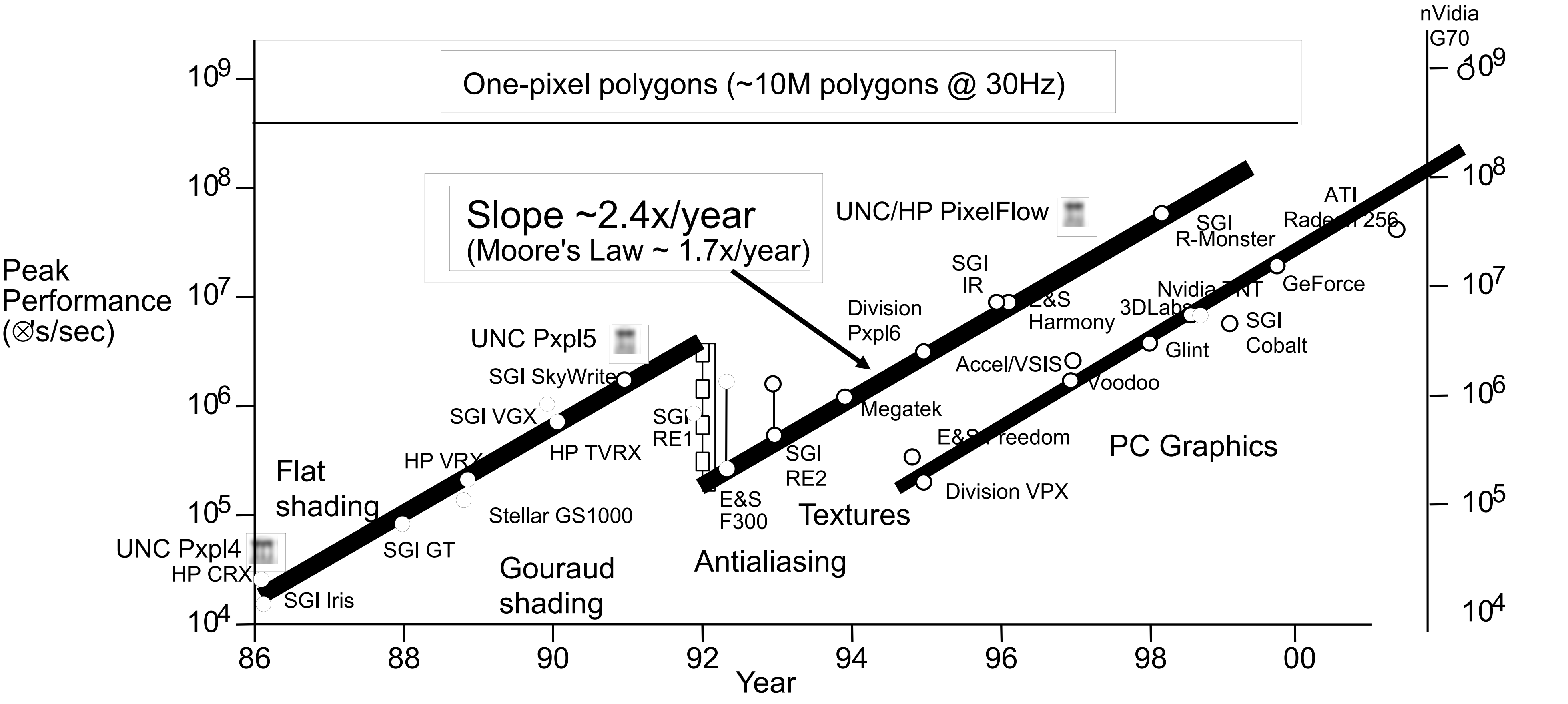
GPU-type computation offers higher GFlops



(Source: Sam Naffziger, AMD)

Fig. 4

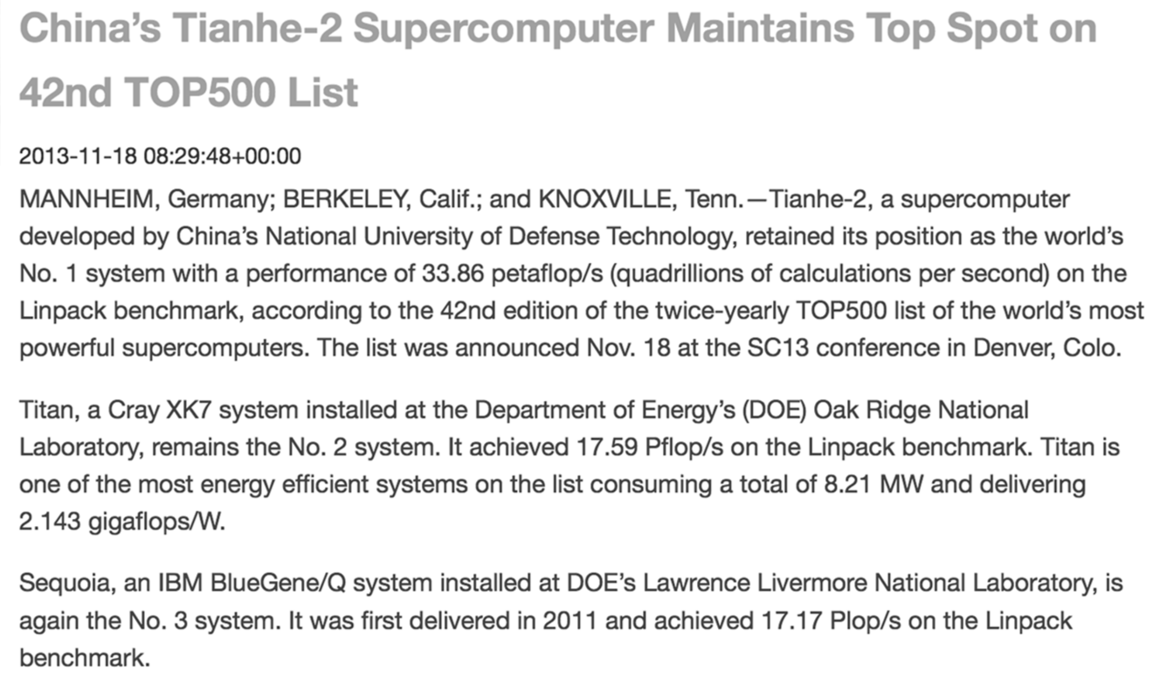
GPUs: Faster than Moore’s Law



Graph courtesy of Professor John Poulton (from Eric Haines)

Supercomputers

- Petaflops (10^{15})
  - GPUs/multicore/100s-1000s cores



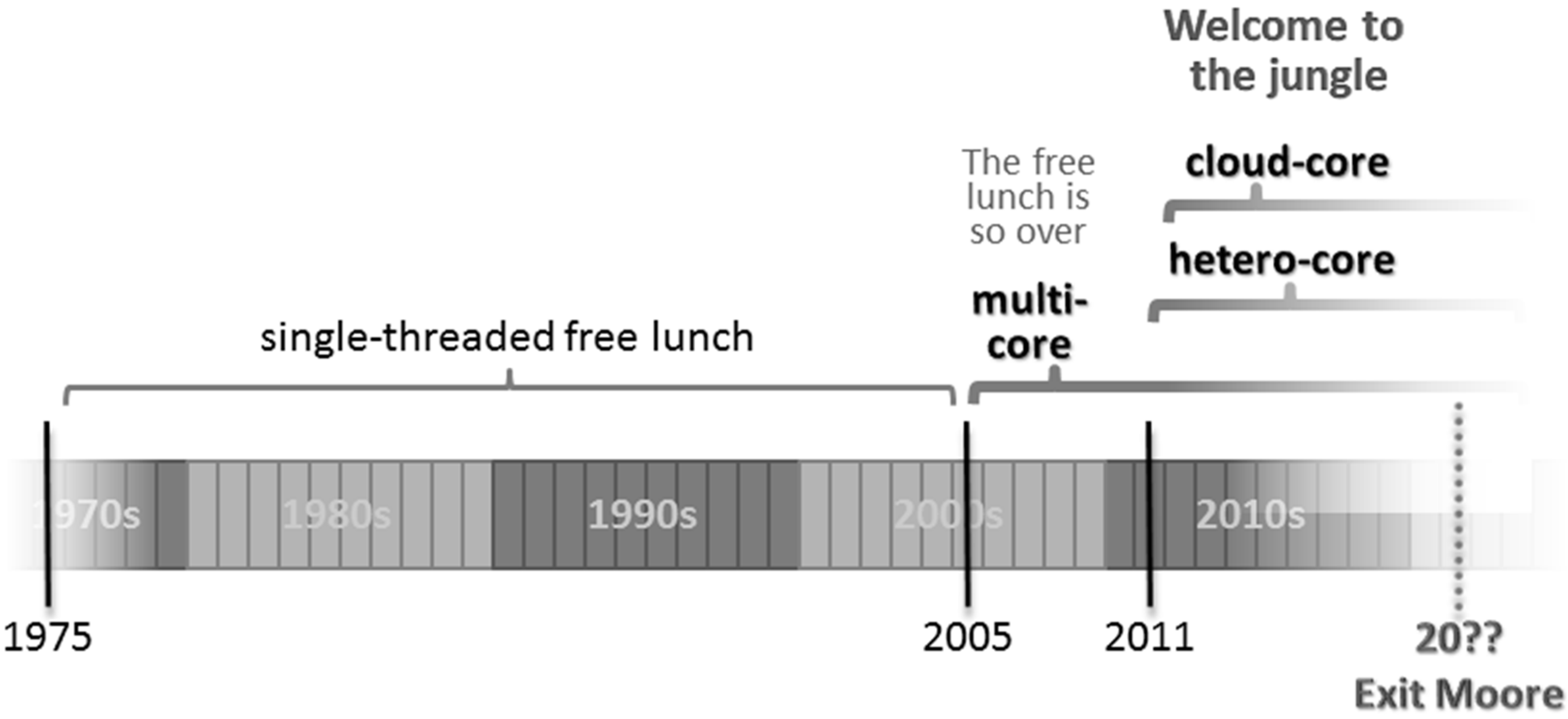
China’s Tianhe-2 Supercomputer Maintains Top Spot on 42nd TOP500 List

2013-11-18 08:29:48+00:00

MANNHEIM, Germany; BERKELEY, Calif.; and KNOXVILLE, Tenn.—Tianhe-2, a supercomputer developed by China’s National University of Defense Technology, retained its position as the world’s No. 1 system with a performance of 33.86 petaflop/s (quadrillions of calculations per second) on the Linpack benchmark, according to the 42nd edition of the twice-yearly TOP500 list of the world’s most powerful supercomputers. The list was announced Nov. 18 at the SC13 conference in Denver, Colo.

Titan, a Cray XK7 system installed at the Department of Energy’s (DOE) Oak Ridge National Laboratory, remains the No. 2 system. It achieved 17.59 Pflop/s on the Linpack benchmark. Titan is one of the most energy efficient systems on the list consuming a total of 8.21 MW and delivering 2.143 gigaflops/W.

Sequoia, an IBM BlueGene/Q system installed at DOE’s Lawrence Livermore National Laboratory, is again the No. 3 system. It was first delivered in 2011 and achieved 17.17 Plop/s on the Linpack benchmark.


Welcome to the jungle
cloud-core
The free lunch is so over
hetero-core
multi-core
single-threaded free lunch
1970s
1980s
1990s
2000s
2010s
1975
2005
2011
20??
Exit Moore

Why?

- Parallelism

- Pipelining

Programmable Hardware

- Started in 1999
- Flexible, programmable
  - – Vertex, Geometry, Fragment Shaders
- And much faster, of course
  - 1999 GeForce256: 0.35 Gigapixel peak fill rate
  - 2001 GeForce3: 0.8 Gigapixel peak fill rate
  - 2003 GeForceFX Ultra: 2.0 Gigapixel peak fill rate
  - ATI Radeon 9800 Pro : 3.0 Gigapixel peak fill rate
  - 2006 NV60: ... Gigapixel peak fill rate
  - 2009 GeForce GTX 285: 10 Gigapixel peak fill rate
  - 2011
    - – GeForce GTC 590: 56 Gigapixel peak fill rate
    - – Radeon HD 6990: 2x26.5
  - 2012
    - – GeForce GTC 690: 62 Gigapixel/s peak fill rate

Course Objective

Bridge the gap between hardware and software

- How a processor works
- How a computer is organized

Establish a foundation for building higher-level applications

- How to understand program performance
- How to understand where the world is going

How class is organized

Instructor: Kavita Bala and Hakim Weatherspoon

(kb@cs.cornell.edu, hweather@cs.cornell.edu)

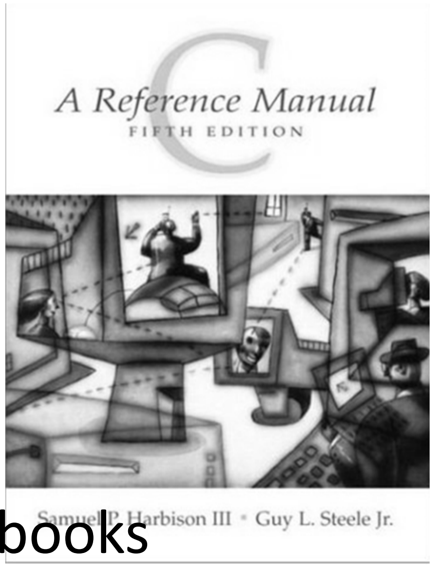
Lecture:

- Tu/Th 1:25-2:40
- Statler Auditorium

Lab sections:

- Start next week
- Carpenter 104 (Blue room)
- Carpenter 235 (Red room)
- Upson B7

COMPUTER ORGANIZATION AND DESIGN
THE HARDWARE/SOFTWARE INTERFACE
FIFTH EDITION
DAVID A. PATTERSON & JOHN L. HENNESSY
MK



books

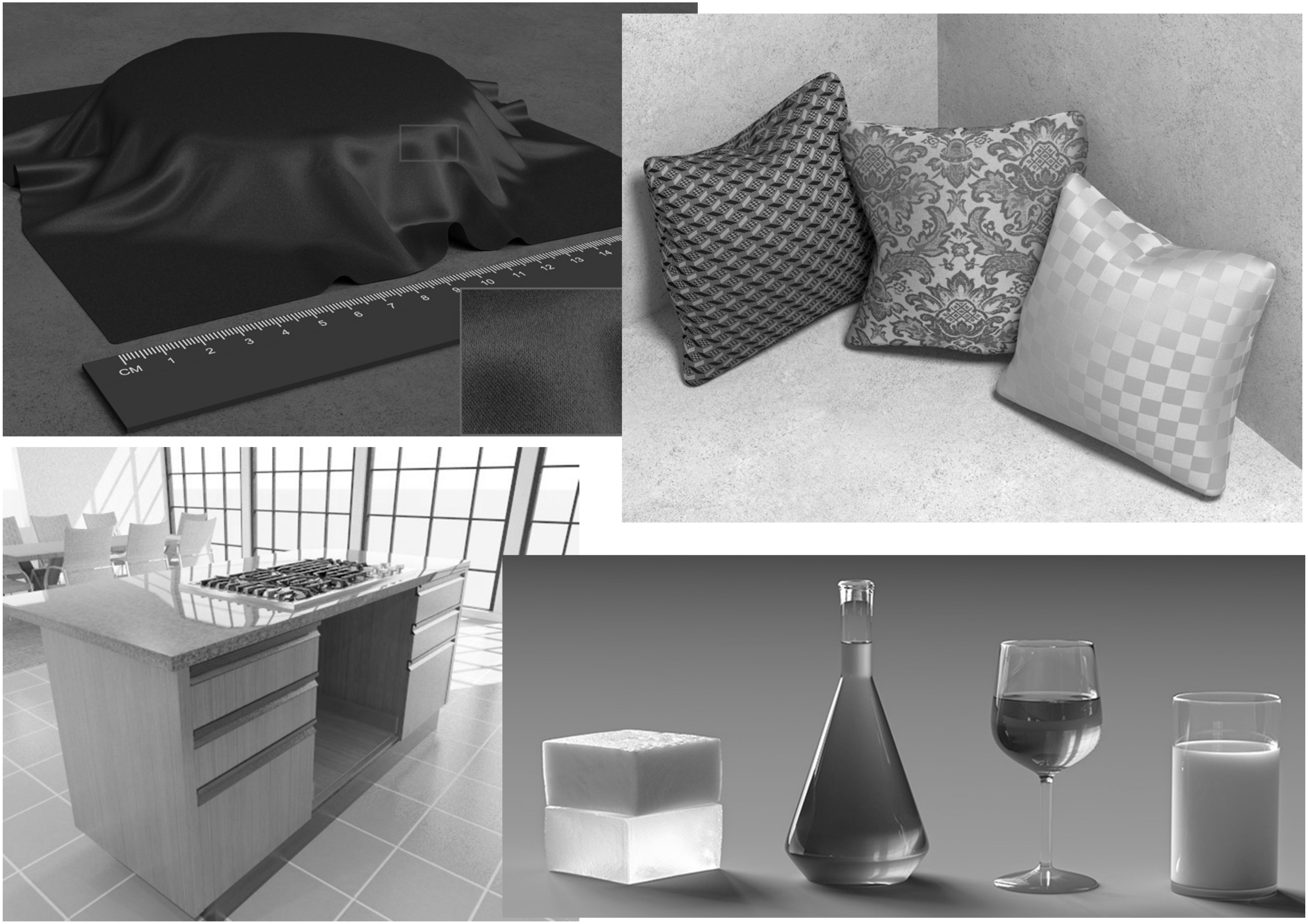
ESSENTIAL COMMANDS
2nd Edition
LINUX
POCKET GUIDE
O'REILLY

Suggested
Textbook

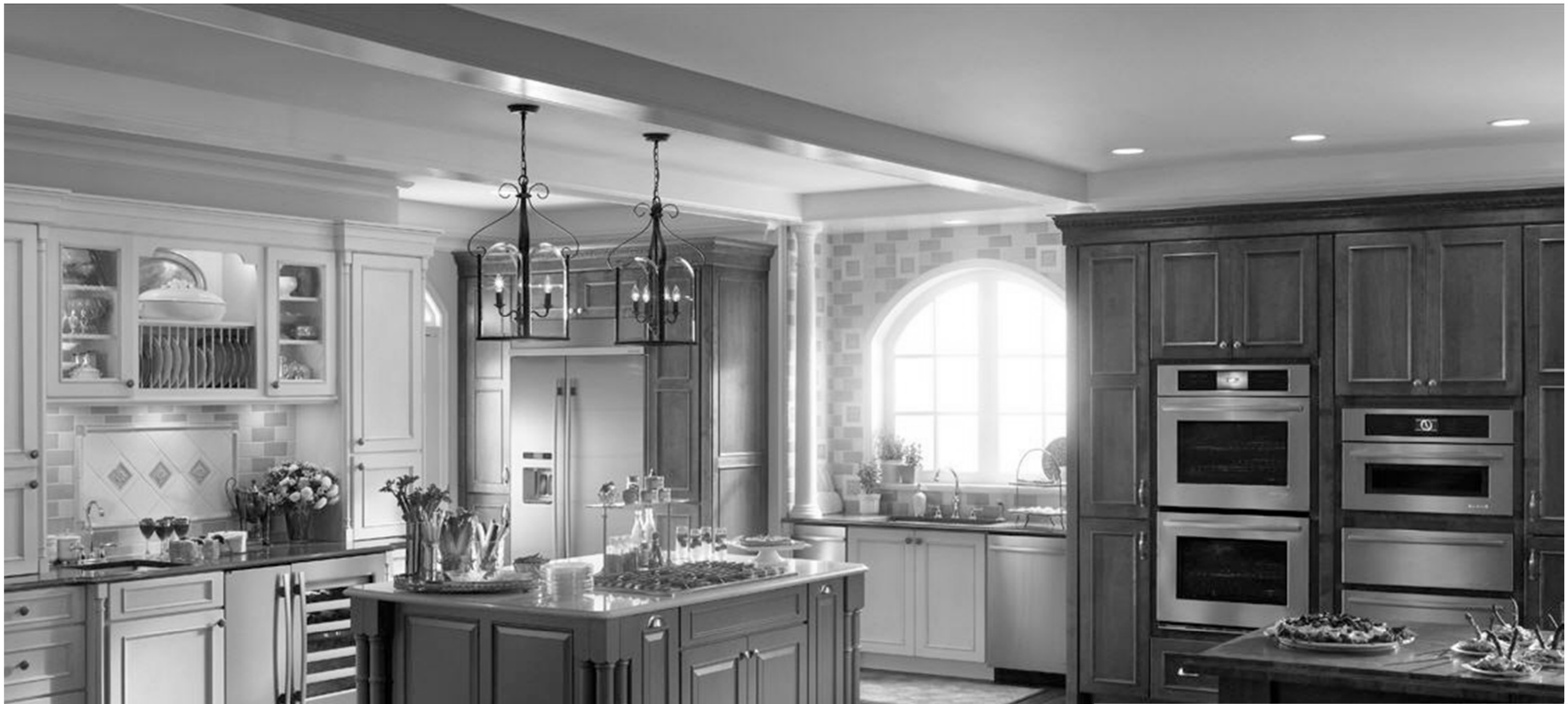
Who am I?

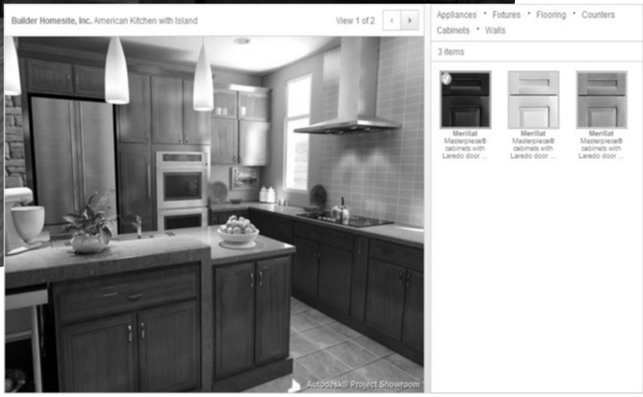
Prof. Kavita Bala

- Ugrad: IIT Bombay
- PhD: MIT
- Started in compilers and systems
- Moved to graphics
- Also work on parallel processing in graphics


CM
1
2
3
4
5
6
7
8
9
10
11
12
13
14

Autodesk 360 Cloud Render





Builder Homesite, Inc. American Kitchen with Island

View 1 of 2

Appliances • Fixtures • Flooring • Counters
Cabinets • Walls

3 items

Merillat Masterpiece® cabinets with Laredo door ...	**Merillat** Masterpiece® cabinets with Laredo door ...	**Merillat** Masterpiece® cabinets with Laredo door ...

Who am I?

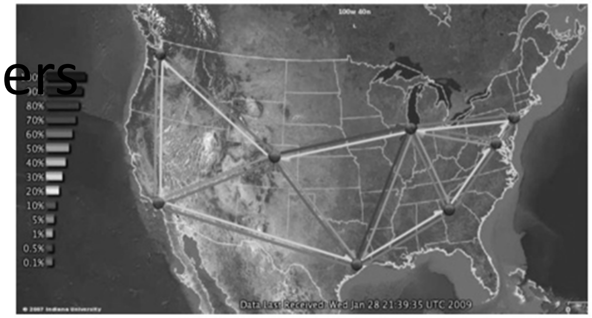
Prof. Hakim Weatherspoon

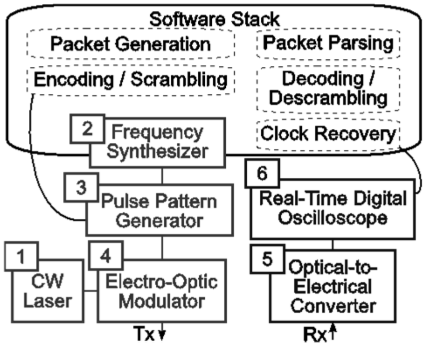
- (Hakim means Doctor, wise, or prof. in Arabic)
- Background in Education
  - Undergraduate University of Washington
    - Played Varsity Football
      - » Some teammates collectively make $100’s of millions
      - » I teach!!!
  - Graduate University of California, Berkeley
    - Some class mates collectively make $100’s of millions
    - I teach!!!
- Background in Operating Systems
  - Peer-to-Peer Storage
    - Antiquity project - Secure wide-area distributed system
    - OceanStore project – Store your data for 1000 years
  - Network overlays
    - Bamboo and Tapestry – Find your data around globe
  - Tiny OS
    - Early adopter in 1999, but ultimately chose P2P direction

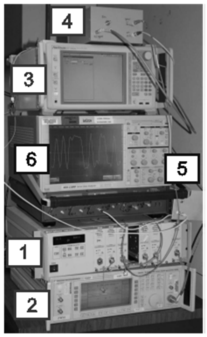
Who am I?

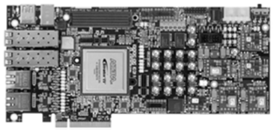
Cloud computing/storage

- Optimizing a global network of data centers


80%
70%
60%
50%
40%
30%
20%
10%
5%
1%
0.5%
0.1%
© 2007 Indiana University
Data Last Received: Wed Jan 28 21:39:35 UTC 2009







Course Staff

cs3410-staff-l@cs.cornell.edu

Lab/Homework TA's

Paul Upchurch	<paulu@cs.cornell.edu>	(PhD)
Zhiming Shen	<zshen@cs.cornell.edu>	(PhD)
Pu Zhang	<pz59@cornell.edu>	(PhD)
Andrew Hirsch	<akh95@cornell.edu>	(PhD)
Emma Kilfoyle	<efk23@cornell.edu>	(MEng)
Roman Averbukh	<raa89@cornell.edu>	(MEng)
Lydia Wang	<lw354@cornell.edu>	(MEng)
Favian Contreras	<fnc4@cornell.edu>	
Victoria Wu	<vw52@cornell.edu>	
Detian Shi	<ds629@cornell.edu>	
Maxwell Dergosits	<mad293@cornell.edu>	
Jimmy Zhu	<jhz22@cornell.edu>	
Antoine Pourchet	<app63@cornell.edu>	
Brady Jacobs	<bij4@cornell.edu>	
Kristen Tierney	<kjt54@cornell.edu>	
Gary Zibrat	<gdz4@cornell.edu>	
Naman Agarwal	<na298@cornell.edu>	
Sanyukta Inamdar	<sri7@cornell.edu>	
Sean Salmon	<ss2669@cornell.edu>	
Ari Karo	<aak82@cornell.edu>	
Brennan Chu	<bc385@cornell.edu>	

Administrative Assistant:

- Molly Trufant (mjt264@cs.cornell.edu)

Pre-requisites and scheduling

CS 2110 is required (Object-Oriented Programming and Data Structures)

- Must have satisfactorily completed CS 2110
- *Cannot take CS 2110 concurrently with CS 3410*

CS 3420 (ECE 3140) (Embedded Systems)

- Take either CS 3410 ***or*** CS 3420
  - both satisfy CS and ECE requirements
- *However, Need ENGRD 2300 to take CS 3420*

CS 3110 (Data Structures and Functional Programming)

- Not advised to take CS 3110 and 3410 together

Pre-requisites and scheduling

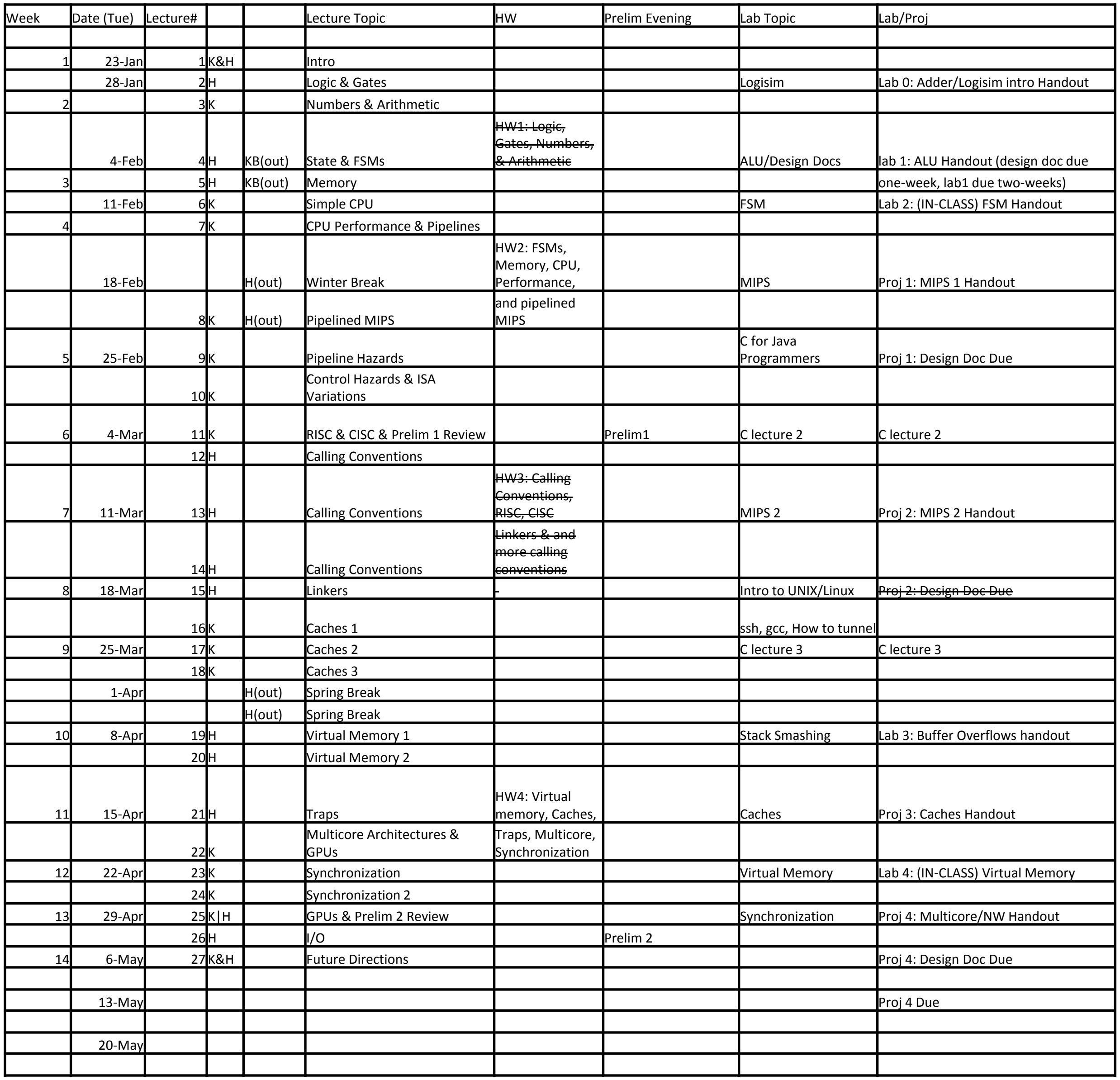
CS 2043 (UNIX Tools and Scripting)

- 2-credit course will greatly help with CS 3410.
- Meets Mon, Wed, Fri at 11:15am-12:05pm in Hollister (HLS) B14
- Class started yesterday and ends March 5th

CS 2022 (Introduction to C) and CS 2024 (C++)

- 1 to 2-credit course will greatly help with CS 3410
- *Unfortunately, offered in the fall, not spring*
- Instead, we will offer a primer to C during lab sections
  and include some C questions in homeworks

Schedule (subject to change)



Week	Date (Tue)	Lecture#			Lecture Topic	HW	Prelim Evening	Lab Topic	Lab/Proj
1	23-Jan	1	K&H		Intro				
	28-Jan	2	H		Logic & Gates			Logisim	Lab 0: Adder/Logisim intro Handout
2		3	K		Numbers & Arithmetic				
	4-Feb	4	H	KB(out)	State & FSMs	~~HW1: Logic, Gates, Numbers, & Arithmetic~~		ALU/Design Docs	lab 1: ALU Handout (design doc due
3		5	H	KB(out)	Memory				one-week, lab1 due two-weeks)
	11-Feb	6	K		Simple CPU			FSM	Lab 2: (IN-CLASS) FSM Handout
4		7	K		CPU Performance & Pipelines				
	18-Feb			H(out)	Winter Break	HW2: FSMs, Memory, CPU, Performance,		MIPS	Proj 1: MIPS 1 Handout
		8	K	H(out)	Pipelined MIPS	and pipelined MIPS			
5	25-Feb	9	K		Pipeline Hazards			C for Java Programmers	Proj 1: Design Doc Due
		10	K		Control Hazards & ISA Variations				
6	4-Mar	11	K		RISC & CISC & Prelim 1 Review		Prelim1	C lecture 2	C lecture 2
		12	H		Calling Conventions				
7	11-Mar	13	H		Calling Conventions	~~HW3: Calling Conventions, RISC, CISC~~		MIPS 2	Proj 2: MIPS 2 Handout
		14	H		Calling Conventions	~~Linkers & and more calling conventions~~			
8	18-Mar	15	H		Linkers	-		Intro to UNIX/Linux	~~Proj 2: Design Doc Due~~
		16	K		Caches 1			ssh, gcc, How to tunnel	
9	25-Mar	17	K		Caches 2			C lecture 3	C lecture 3
		18	K		Caches 3				
	1-Apr			H(out)	Spring Break				
				H(out)	Spring Break				
10	8-Apr	19	H		Virtual Memory 1			Stack Smashing	Lab 3: Buffer Overflows handout
		20	H		Virtual Memory 2				
11	15-Apr	21	H		Traps	HW4: Virtual memory, Caches,		Caches	Proj 3: Caches Handout
		22	K		Multicore Architectures & GPUs	Traps, Multicore, Synchronization			
12	22-Apr	23	K		Synchronization			Virtual Memory	Lab 4: (IN-CLASS) Virtual Memory
		24	K		Synchronization 2				
13	29-Apr	25	K|H		GPUs & Prelim 2 Review			Synchronization	Proj 4: Multicore/NW Handout
		26	H		I/O		Prelim 2		
14	6-May	27	K&H		Future Directions				Proj 4: Design Doc Due
	13-May								Proj 4 Due
	20-May								

Grading

Lab (50% approx.)

- 5-6 Individual Labs
  - 2 out-of-class labs (5-10%)
  - 3-4 in-class labs (5-7.5%)
- 4 Group Projects (30-35%)
- Participation/Quizzes in lab (2.5%)

Lecture (50% approx.)

- 2 Prelims (35%)
  - Dates: March 4, May 1
- Homework (10%)
- Participation/Quizzes in lecture (5%)

Grading

Regrade policy

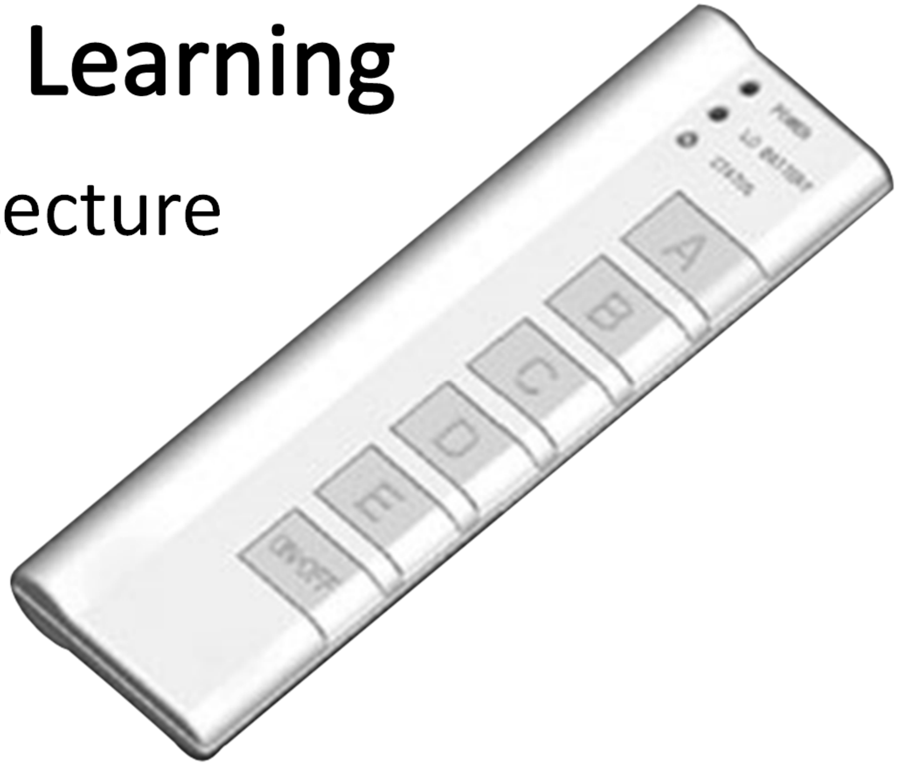
- Submit written request to lead TA,
  and lead TA will pick a different grader
- Submit another written request,
  lead TA will regrade directly
- Submit *yet* another written request for professor to regrade

Late Policy

- Each person has a total of ***four*** “slip days”
- Max of ***two*** slip days for any individual assignment
- For projects, slip days are deducted from all partners
- 25% deducted per day late after slip days are exhausted

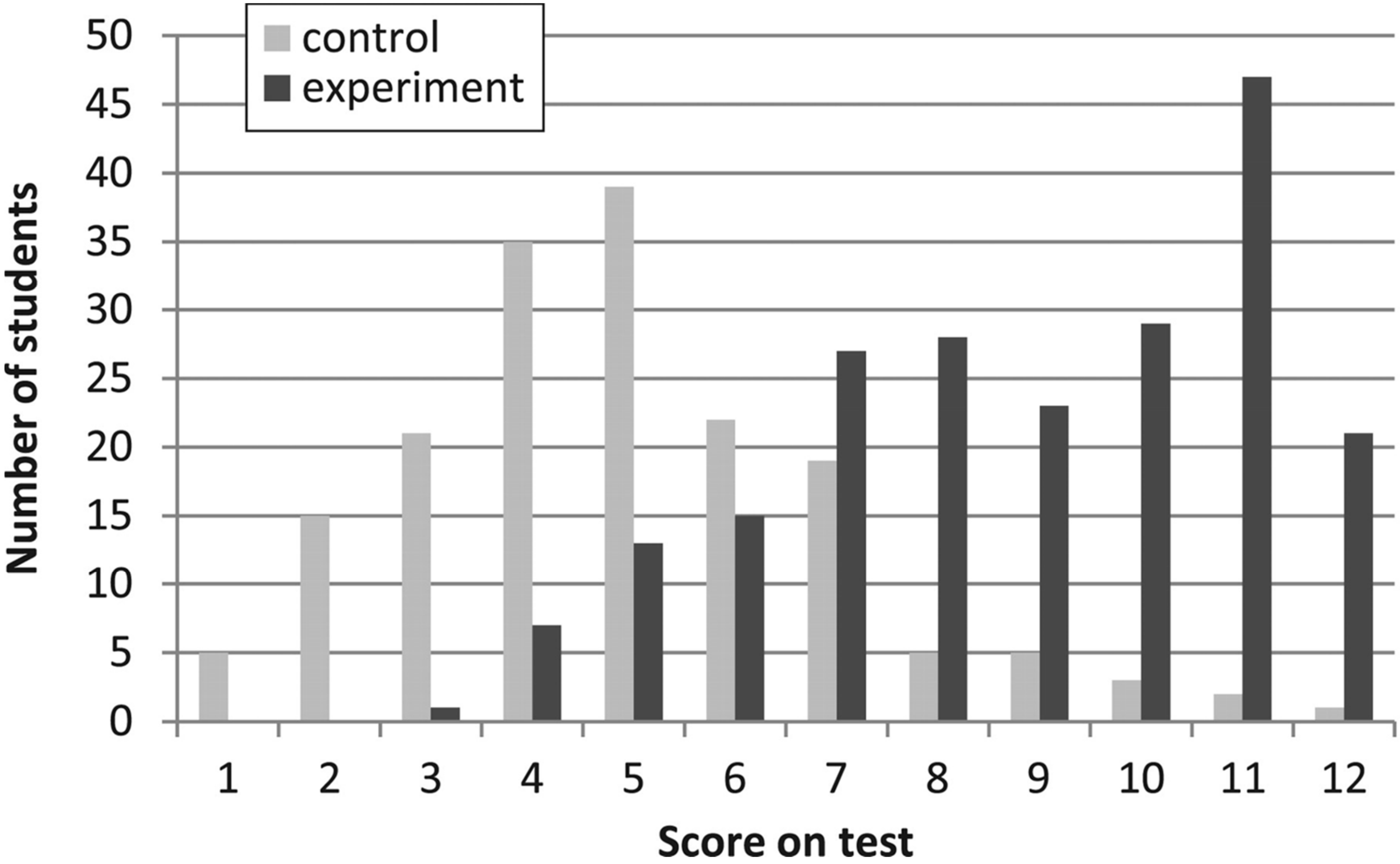
Active Learning

iClicker: Bring to every Lecture



Put all devices into ***Airplane Mode***

Active Learning



L Deslauriers et al. Science 2011;332:862-864

Fig. 1 Histogram of 270 physic student scores for the two sections: Experiment w/ quizzes and active learning. Control without.

Published by AAAS

Science
AAAS

Active Learning

Demo: What year are you in school?

a) Freshman

b) Sophomore

c) Junior

d) Senior

e) Other

Active Learning

Also, activity handouts will be available before class

In front of doors before you walk in

Administrivia

http://www.cs.cornell.edu/courses/cs3410/2014sp

- Office Hours / Consulting Hours
- Lecture slides, schedule, and Logisim
- CSUG lab access (esp. second half of course)

Lab Sections (start ***next week***)

T	2:55 – 4:10pm	Carpenter Hall 104 (Blue Room)
W	8:40—9:55am	Carpenter Hall 104 (Blue Room)
W	11:40am – 12:55pm	Carpenter Hall 104 (BlueRoom)
W	3:35 – 4:50pm	Carpenter Hall 104 (Blue Room)
W	7:30—8:45pm	Carpenter Hall 235 (Red Room)
R	8:40 – 9:55pm	Carpenter Hall 104 (Blue Room)
R	11:40 – 12:55pm	Carpenter Hall 104 (Blue Room)
R	2:55 – 4:10pm	Carpenter Hall 104 (Blue Room)
F	8:40 – 9:55am	Carpenter Hall 104 (Blue Room)
F	11:40am – 12:55pm	Upson B7
F	2:55 – 4:10pm	Carpenter Hall 104 (Blue Room)

- Labs are separate than lecture and homework
- Bring laptop to Labs
- ***Next*** week: intro to logisim and building an adder

Administrivia

http://www.cs.cornell.edu/courses/cs3410/2014sp

- Office Hours / Consulting Hours
- Lecture slides, schedule, and Logisim
- CSUG lab access (esp. second half of course)

Course Virtual Machine (VM)

- Identical to CSUG Linux machines
- Download and use for labs and projects
- **https://confluence.cornell.edu/display/coecis/CSUG+Lab+VM+Informatior**

Communication

Email

- cs3410-staff-l@cs.cornell.edu
- The email alias goes to me and the TAs, not to whole class

Assignments

- CMS: http://cms.csuglab.cornell.edu

Newsgroup

- http://www.piazza.com/cornell/spring2014/cs3410
- For students

iClicker

- http://atcsupport.cit.cornell.edu/pollsrvc/

Lab Sections, Projects, and Homeworks

Lab Sections start ***next*** week

- Intro to logisim and building an adder

Labs Assignments

- Individual
- One week to finish (usually Monday to Monday)

Projects

- two-person teams
- Find partner in same section

Homeworks

- One before each prelim
- Will be released a few weeks ahead of time
- Finish question after covered in lecture

Academic Integrity

All submitted work must be your own

- OK to study together, but do not share soln’s
- Cite your sources

Project groups submit joint work

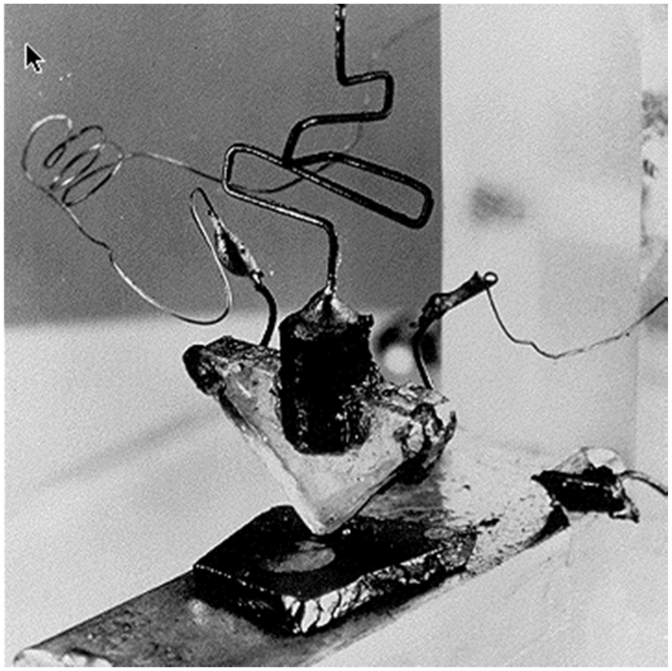
- Same rules apply to projects at the group level
- Cannot use of someone else’s soln

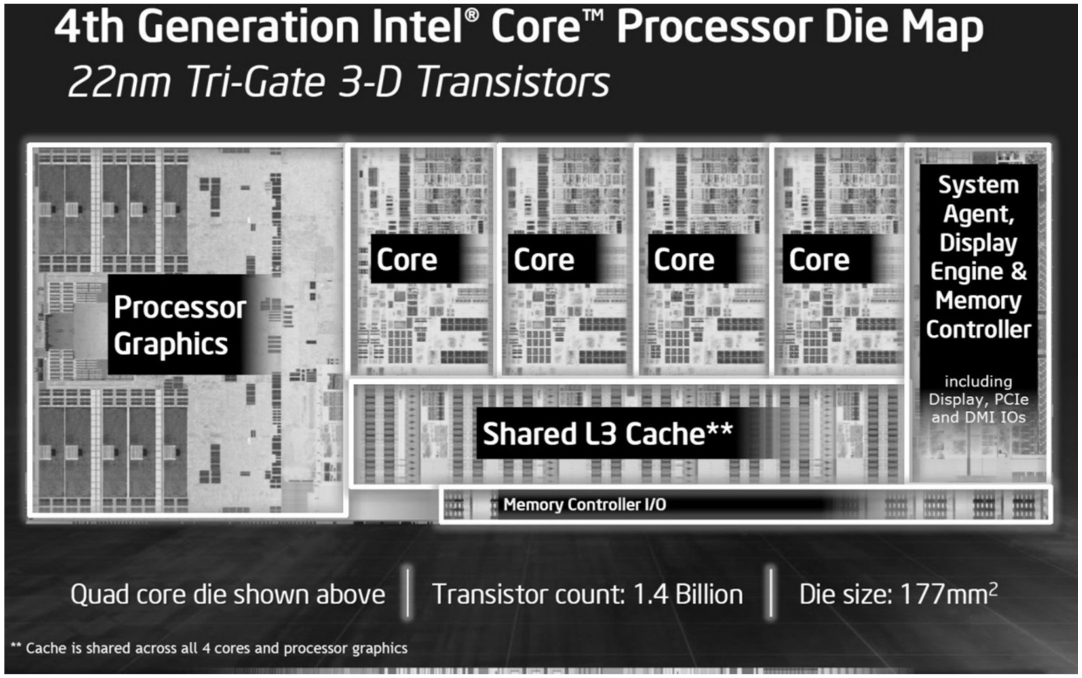
Closed-book exams, no calculators

- Stressed? Tempted? Lost?
  - Come see us before due date!

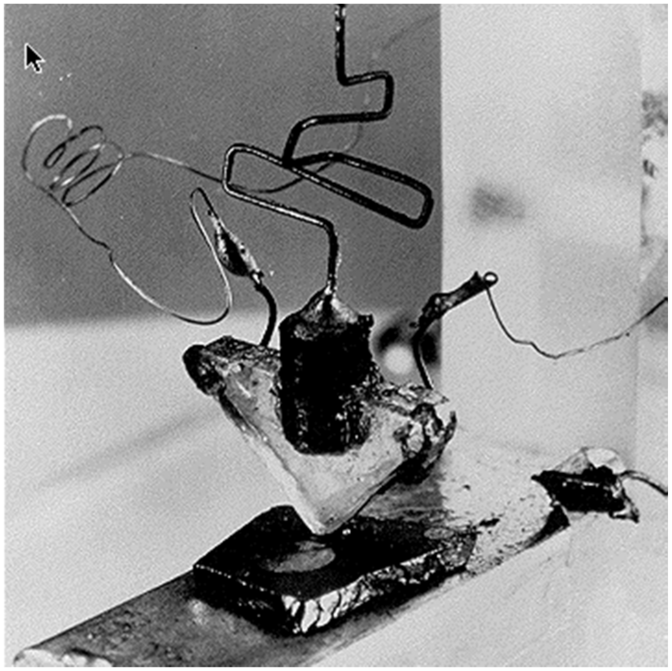
Plagiarism in any form will not be tolerated

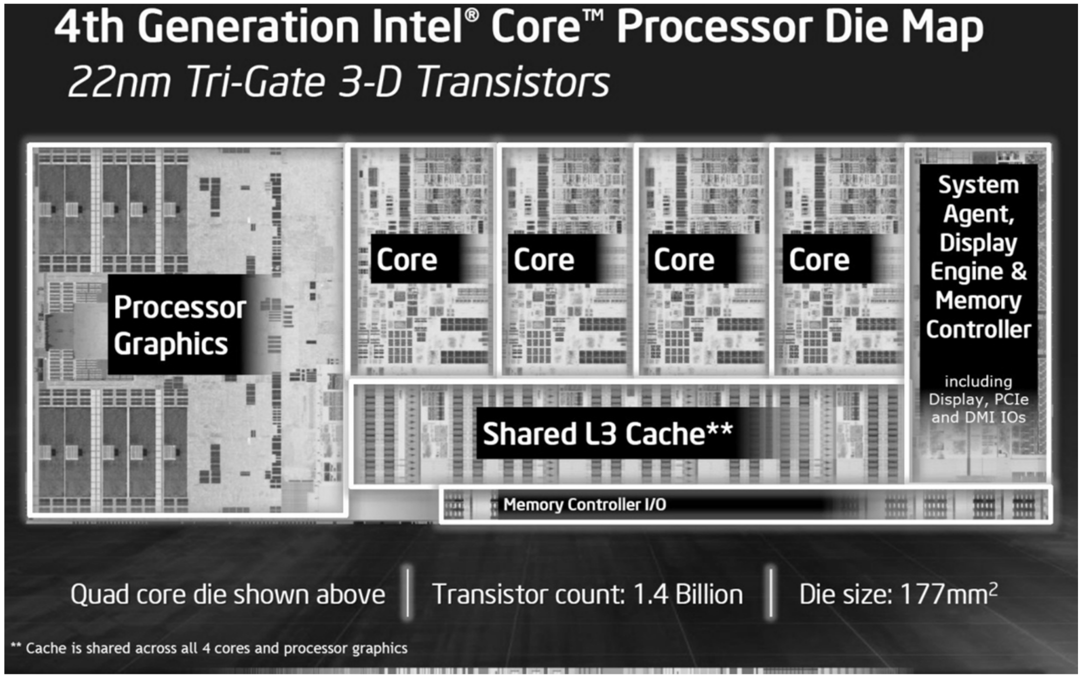
Why do CS Students Need Transistors?




4th Generation Intel® Core™ Processor Die Map
22nm Tri-Gate 3-D Transistors
Processor Graphics
Core
Core
Core
Core
System Agent, Display Engine & Memory Controller
including Display, PCIe and DMI IOs
Shared L3 Cache**
Memory Controller I/O
Quad core die shown above
Transistor count: 1.4 Billion
Die size: 177mm²
** Cache is shared across all 4 cores and processor graphics

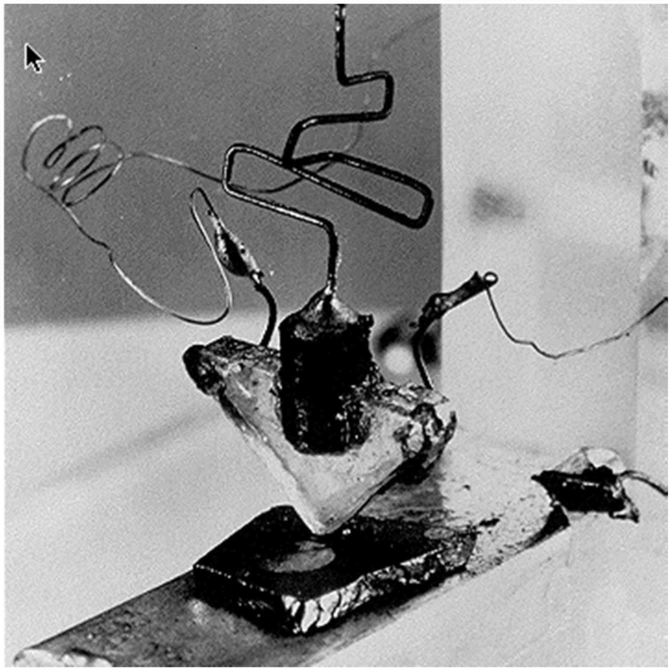
Why do CS Students Need Transistors?

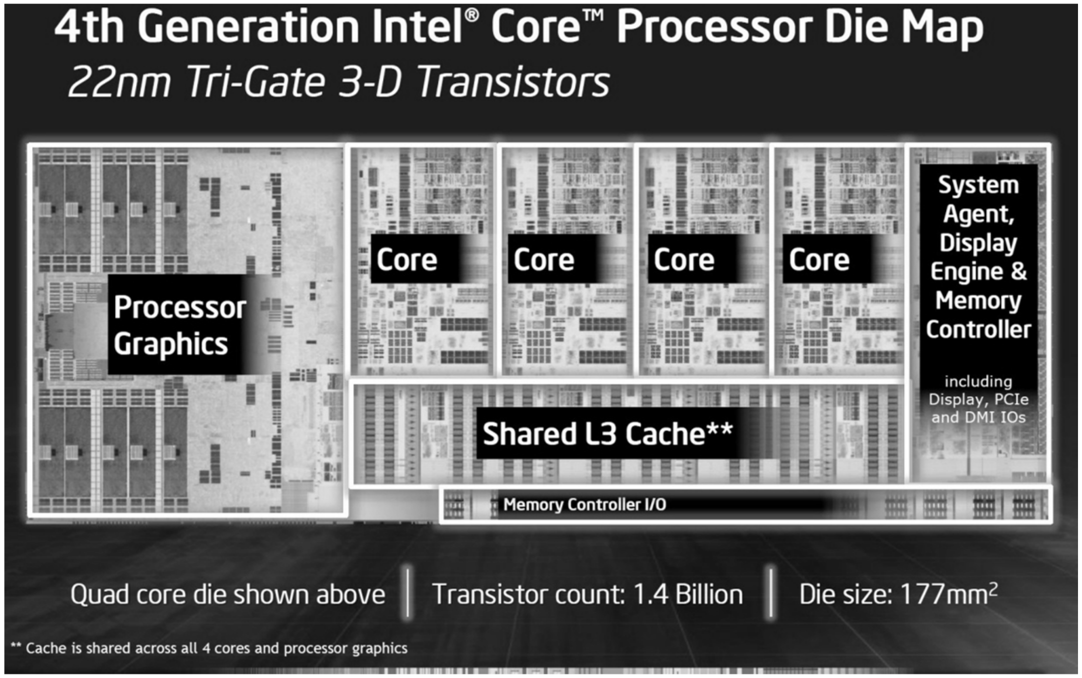



4th Generation Intel® Core™ Processor Die Map
22nm Tri-Gate 3-D Transistors
Processor Graphics
Core
Core
Core
Core
System Agent, Display Engine & Memory Controller
including Display, PCIe and DMI IOs
Shared L3 Cache**
Memory Controller I/O
Quad core die shown above
Transistor count: 1.4 Billion
Die size: 177mm²
** Cache is shared across all 4 cores and processor graphics

Functionality and Performance

Why do CS Students Need Transistors?

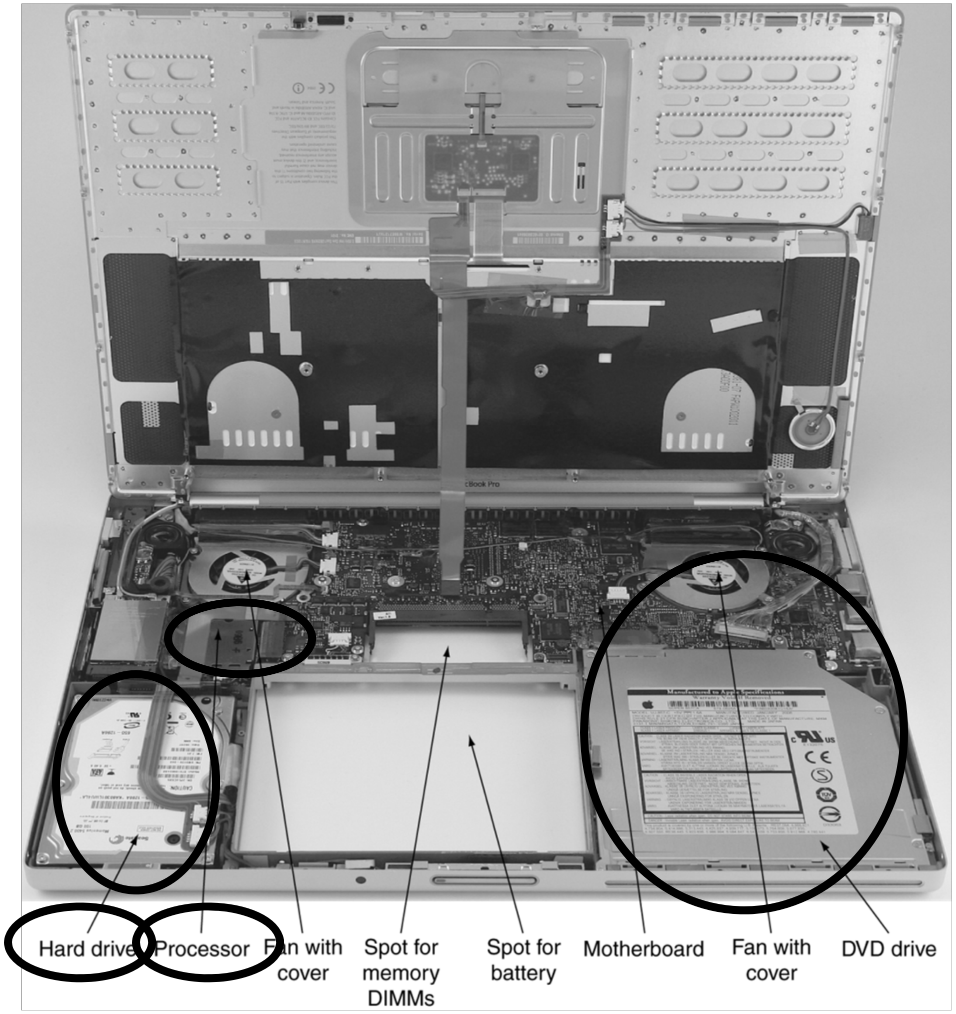


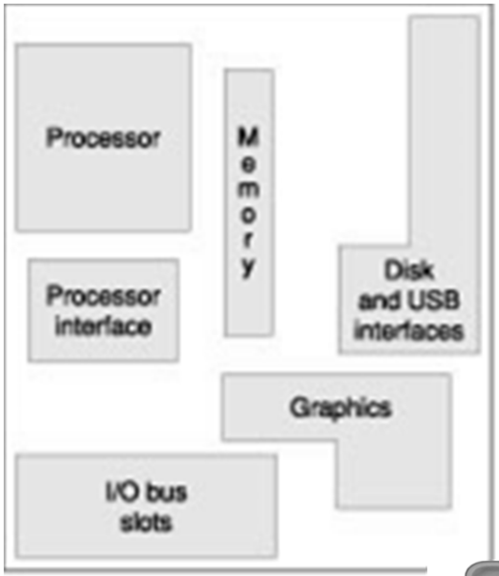

4th Generation Intel® Core™ Processor Die Map
22nm Tri-Gate 3-D Transistors
Processor Graphics
Core
Core
Core
Core
System Agent, Display Engine & Memory Controller
including Display, PCIe and DMI IOs
Shared L3 Cache**
Memory Controller I/O
Quad core die shown above
Transistor count: 1.4 Billion
Die size: 177mm²
** Cache is shared across all 4 cores and processor graphics

To be better Computer Scientists and Engineers

- Abstraction: simplifying complexity
- How is a computer system organized? How do I build it?
- How do I program it? How do I change it?
- How does its design/organization effect performance?

Computer System Organization





Computer System Organization

Computer System = ?

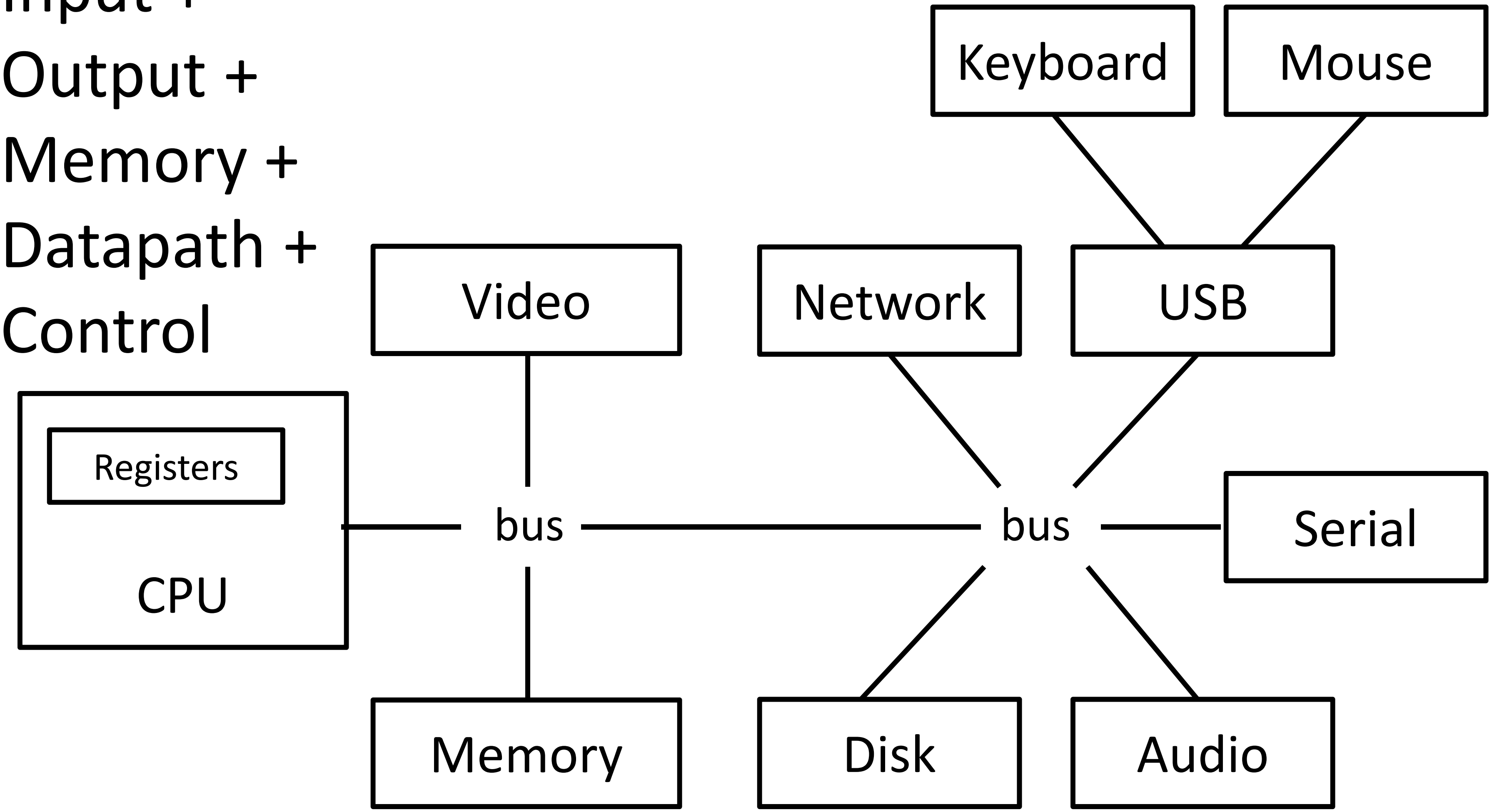
Input +

Output +

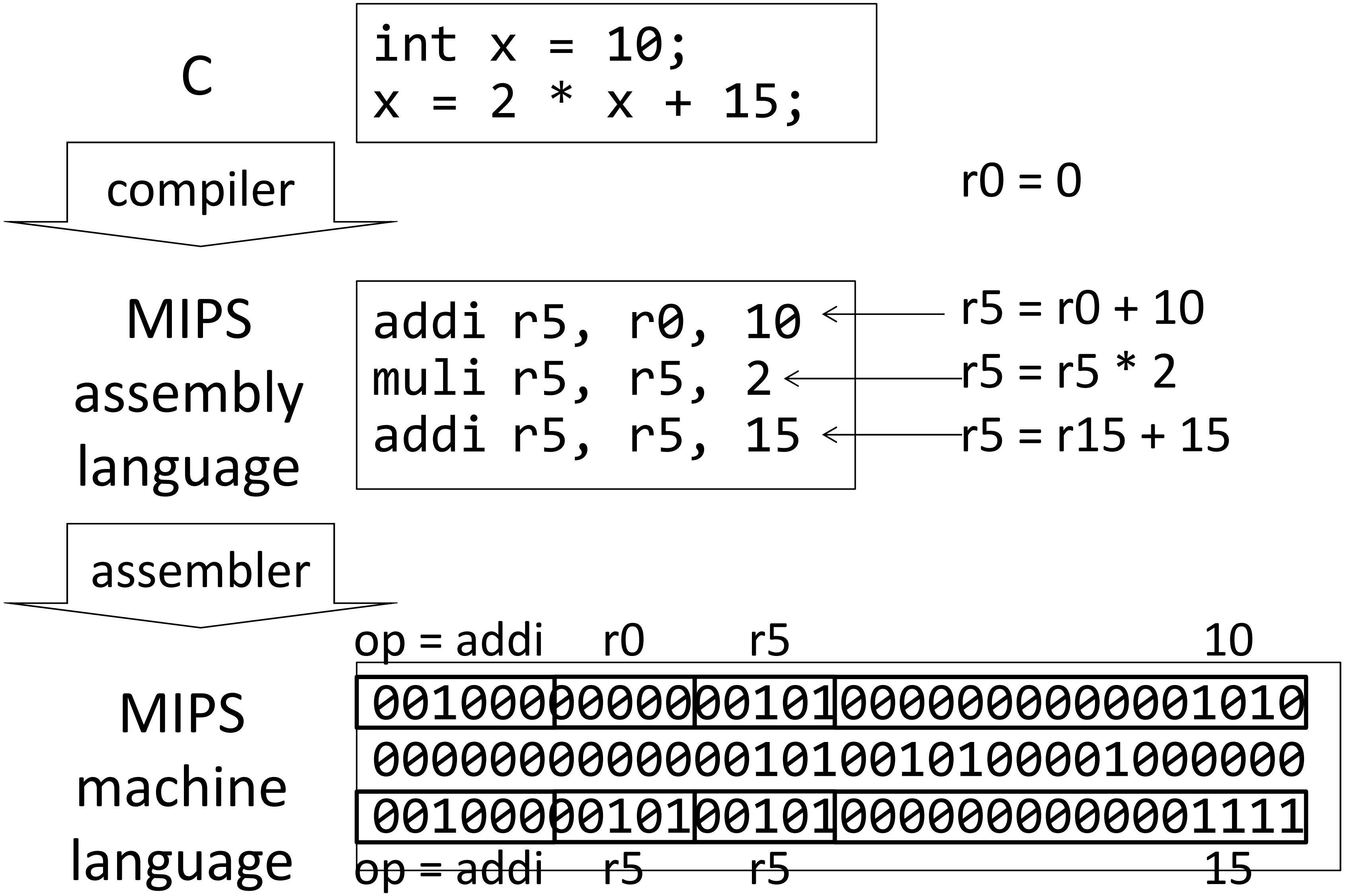
Memory +

Datapath +

Control



Compilers & Assemblers



C

```
int x = 10;
x = 2 * x + 15;
```

compiler

r0 = 0

MIPS assembly language

```
addi r5, r0, 10
muli r5, r5, 2
addi r5, r5, 15
```

r5 = r0 + 10

r5 = r5 * 2

r5 = r15 + 15

assembler

MIPS machine language

```
op = addi     r0        r5                          10
001000 00000 00101 0000000000001010
00000000000001010010100001000000
001000 00101 00101 0000000000001111
op = addi     r5        r5                          15
```

Instruction Set Architecture

ISA

- abstract interface between hardware and the lowest level software

- user portion of the instruction set plus the operating system interfaces used by application programmers

Basic Computer System

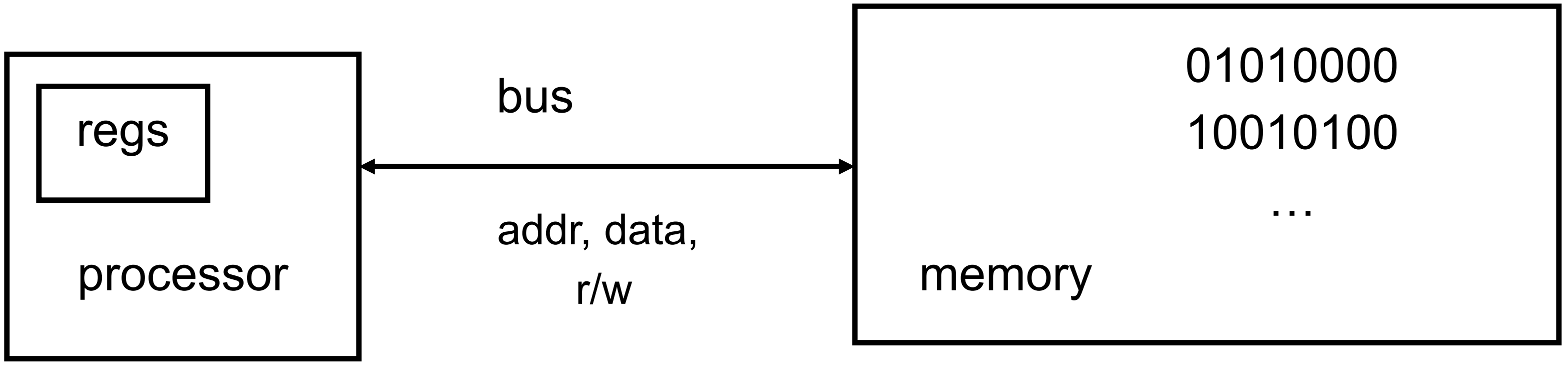
A processor executes instructions

- Processor has some internal state in storage elements (registers)

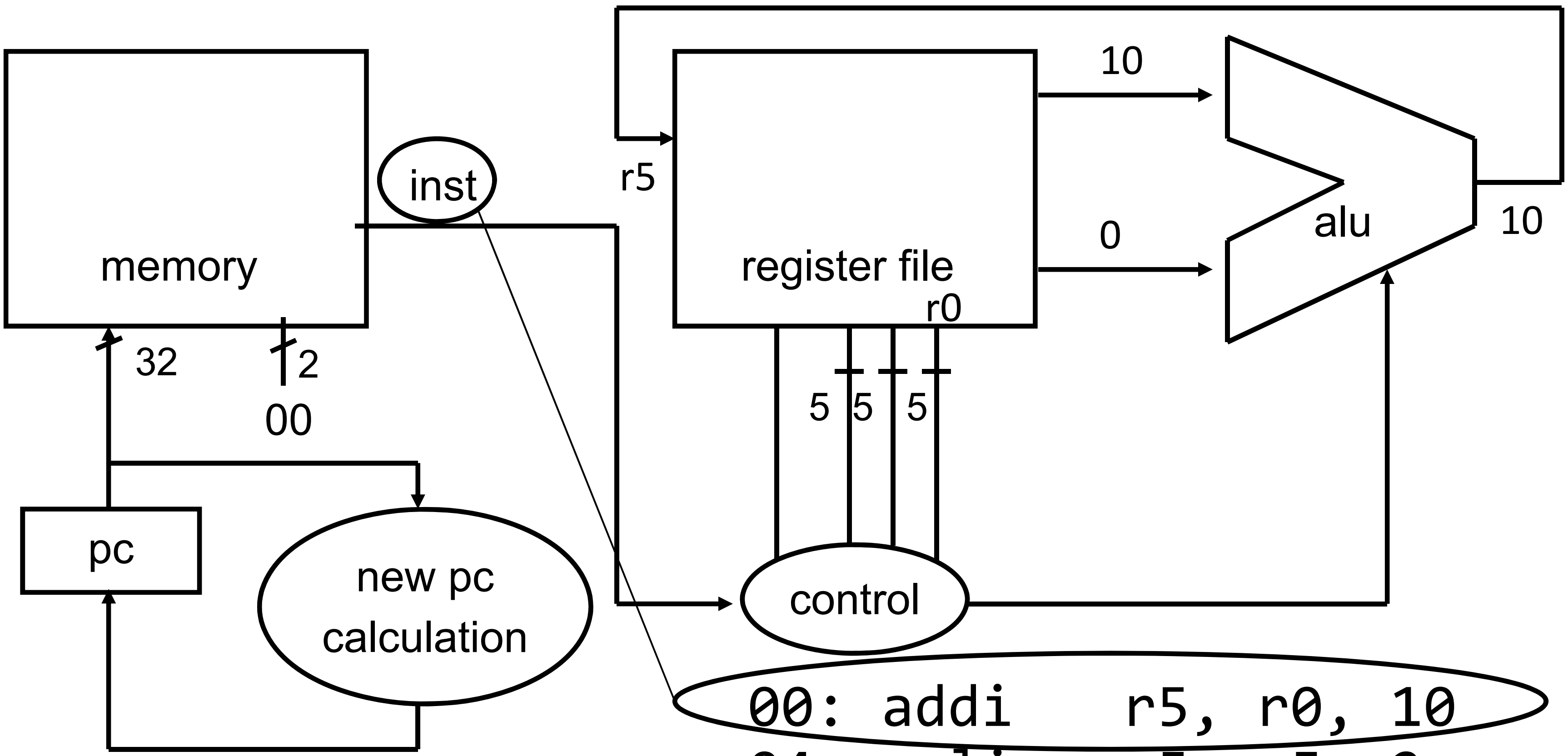
A memory holds instructions and data

- von Neumann architecture: combined inst and data

A bus connects the two



How to Design a Simple Processor



```
00: addi   r5, r0, 10
04: muli   r5, r5, 2
08: addi   r5, r5, 15
```

Inside the Processor

AMD Barcelona: 4 processor cores

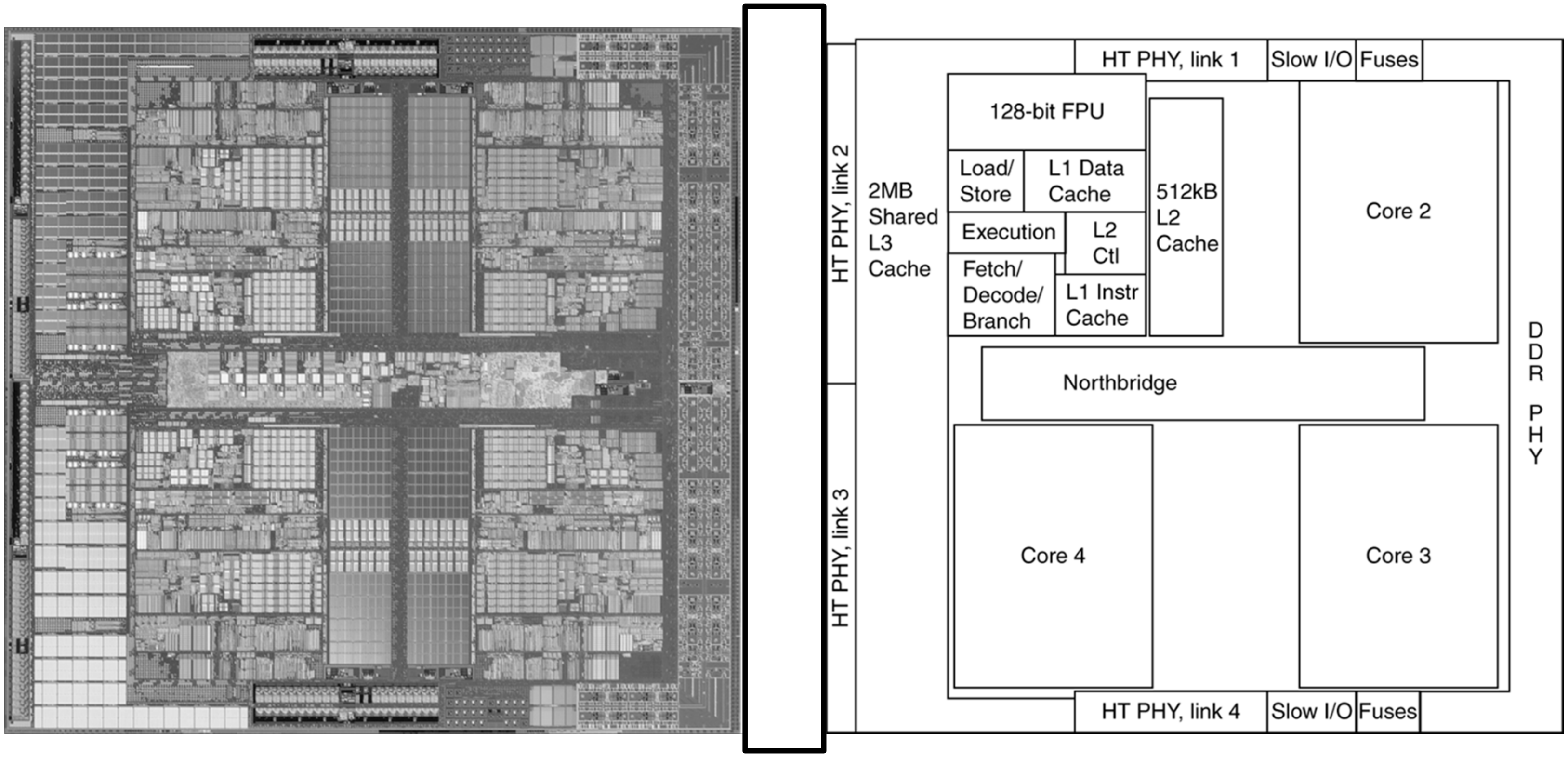
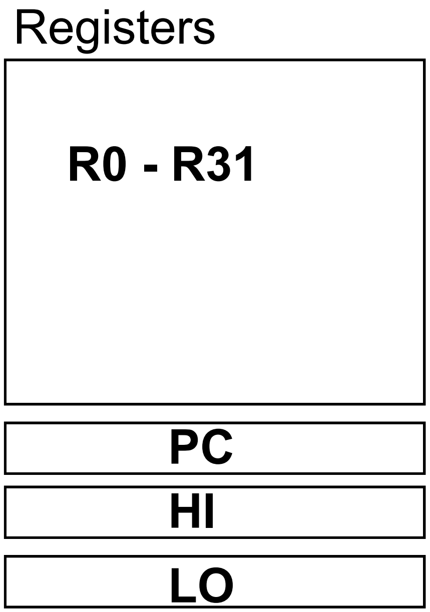


Figure from Patterson & Hennesssy, Computer Organization and Design, 4th Edition

How to Program the Processor: MIPS R3000 ISA

Instruction Categories

- Load/Store
- Computational
- Jump and Branch
- Floating Point
  - coprocessor
- Memory Management



Registers

R0 - R31

PC

HI

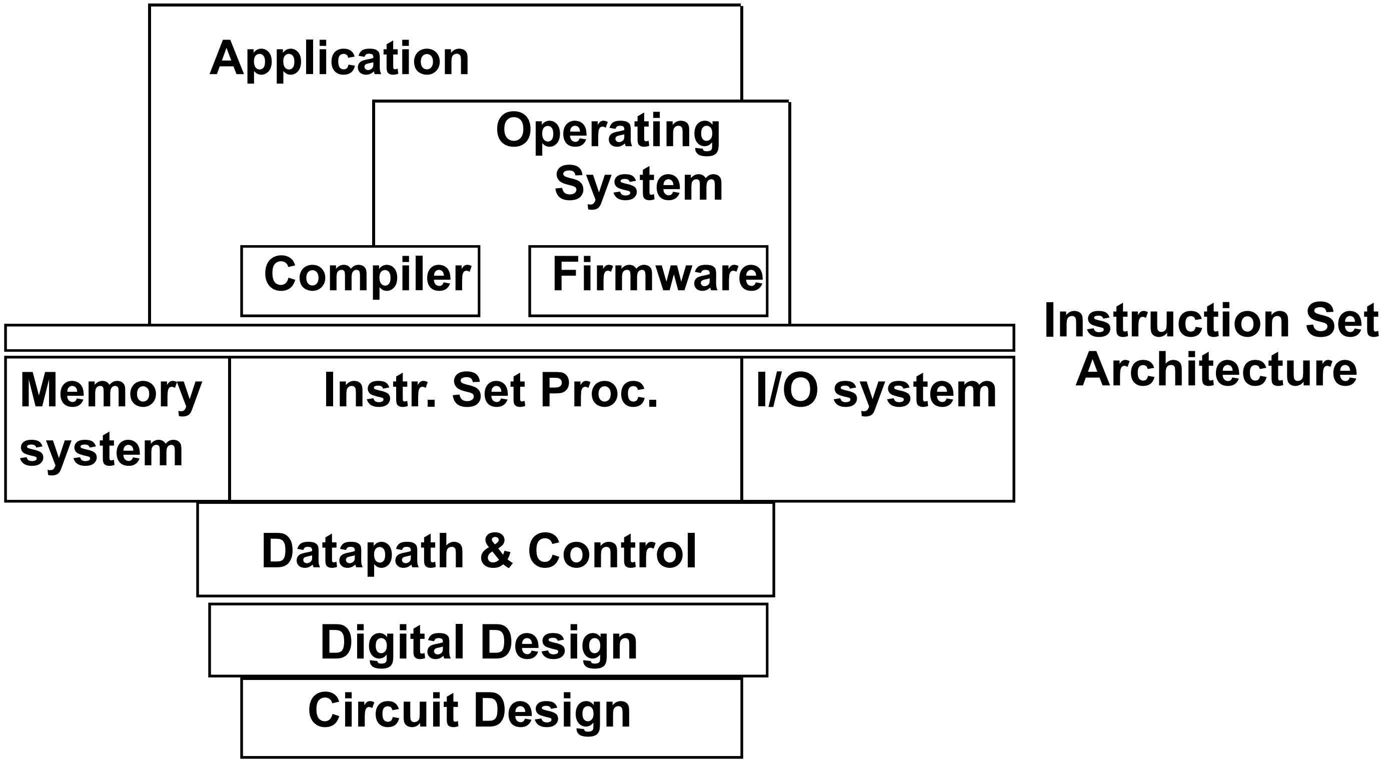
LO

OP	rs	rt	rd	sa	funct

OP	rs	rt	immediate

OP	jump target

Overview


Application
Operating System
Compiler
Firmware
Instruction Set Architecture
Memory system
Instr. Set Proc.
I/O system
Datapath & Control
Digital Design
Circuit Design

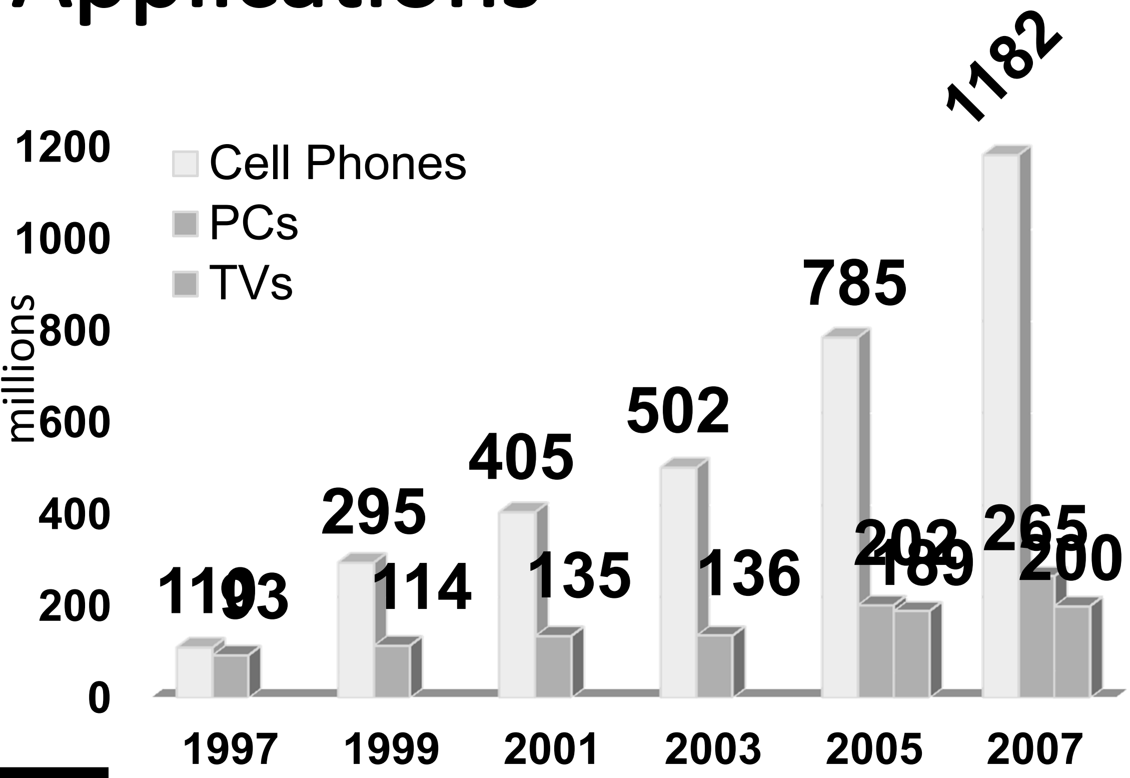
Applications

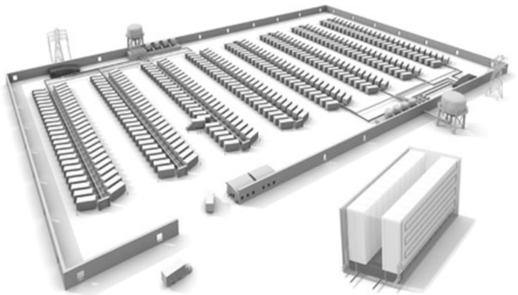
Everything these days!

- Phones, cars, televisions, games, computers,…

Applications

Xilinx FPGA





Cloud Computing

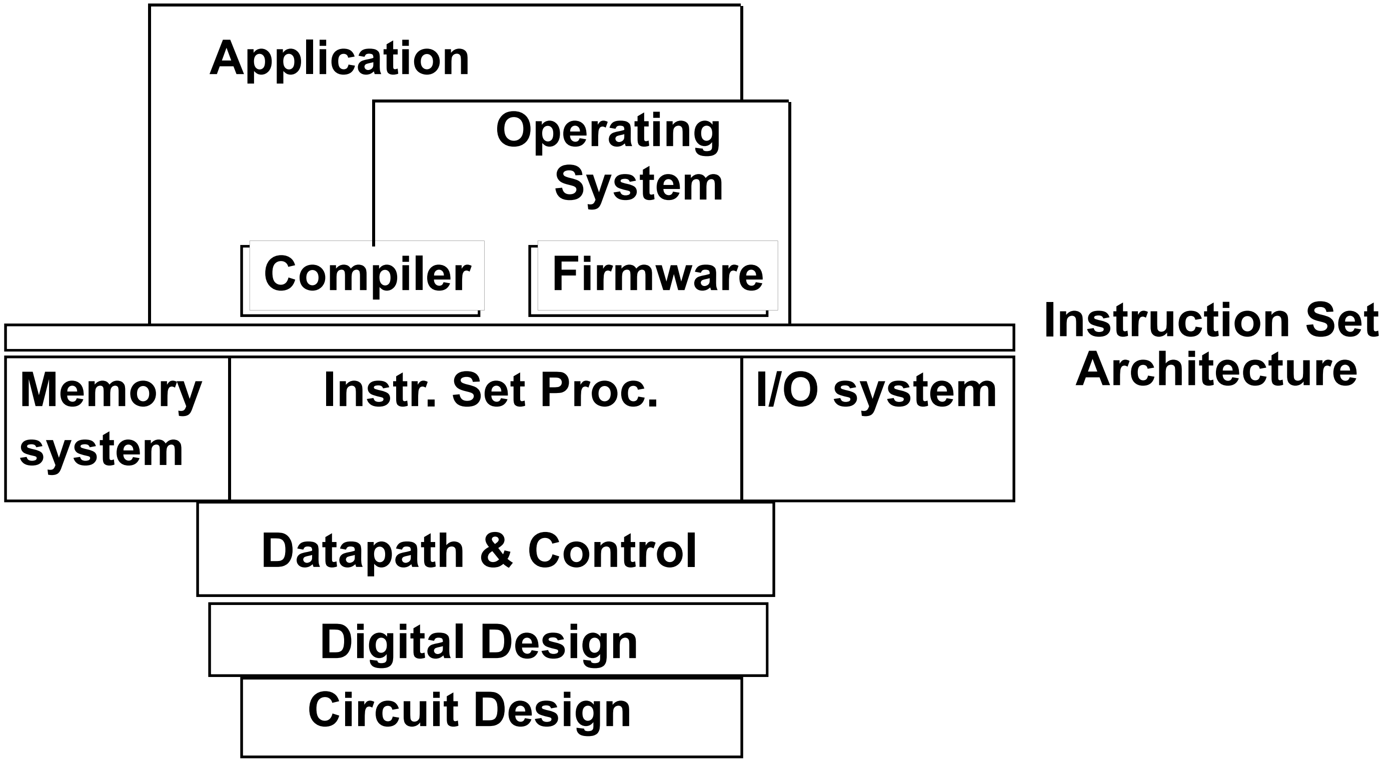
NVidia GPU

Berkeley mote

Cell Phone

Cars

Covered in this course



Reflect

Why take this course?

- Basic knowledge needed for *all* other areas of CS:

  operating systems, compilers, ...
- Levels are not independent

  hardware design ↔ software design ↔ performance
- Crossing boundaries is hard but important

  device drivers
- Good design techniques

  abstraction, layering, pipelining, parallel vs. serial, ...
- Understand where the world is going