CS250
Computer Architecture

Fall 2012

Department of Computer Science
Purdue University

Course Information

Instructor:	Dongyan Xu (dxu@cs.purdue.edu)
	Associate Professor
Office:	LWSN 1173
Office hours:	Tuesday 1:00pm-3:00pm, or
	by appointment

TAs:

- Ashwin Jiwane (G)
- Meng-Lin Wu (G)
- Chunmeng Zhou (G)
- Seth A. Butts (U)
- Jason D. Salter (U)
- Sergei V. Uversky (U)

Course Information

- Course homepage:

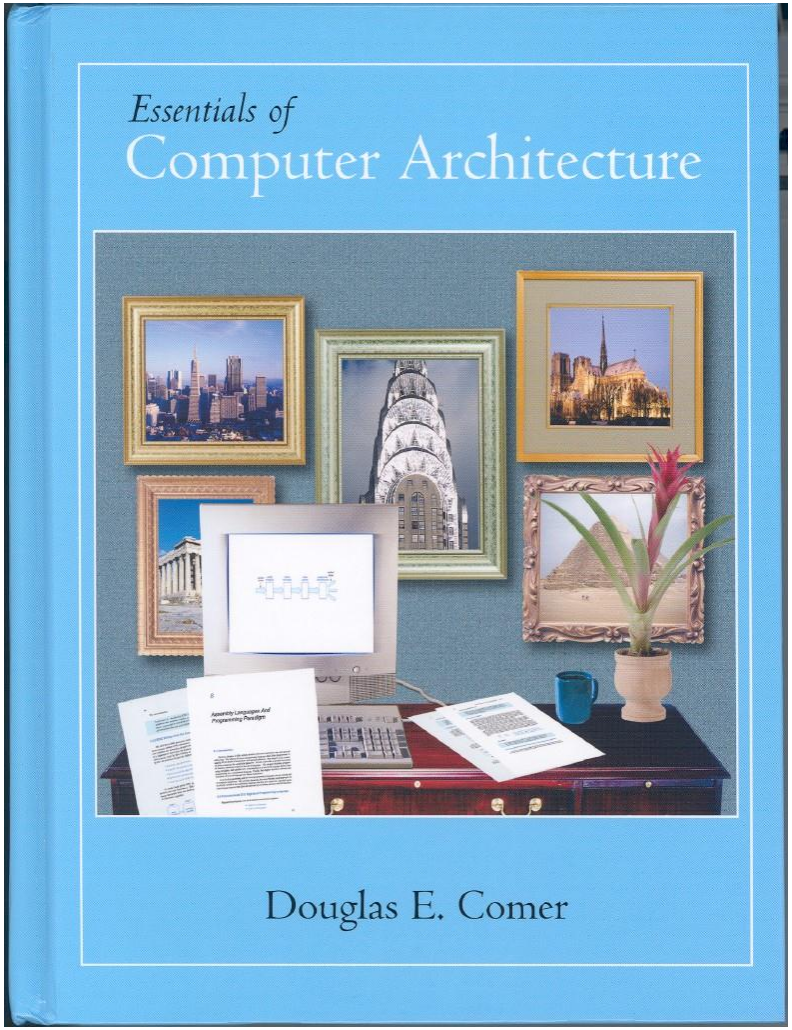
  http://www.cs.purdue.edu/homes/cs250

  - Syllabus
  - Course outline
  - Academic integrity policy
  - TAs schedule and emails

- Grading Policies:

Written homework:	5%
Lab assignments :	50% (plus bonus)
Midterm:	20%
Final:	25%
Quizzes:	3% (bonus)

Textbook



We will be using the electronic manuscript of the 2nd Edition (not available in stores)

Announcements

- There will be NO lab sessions this week.
- Lab sessions will start in Week 2 or Week 3 (TBA).

Acknowledgement

- The following people generously share their lecture slides:
  - Prof. Douglas Comer at Purdue University
  - Prof. Craig Zilles and Mr. Howie Huang at the University of Illinois at Urbana-Champaign
  - Mr. Hicham Elmongui at Purdue University
  - Prof. Chris Gniady at the University of Arizona

Ubiquity of Computers

- Originally used by the government
  - e.g. military applications
  - "I think there is a world market for maybe five computers." *Thomas Watson, chairman of IBM, 1943.*
  - "There is no reason anyone would want a computer in their home." *Ken Olson, president, chairman and founder of Digital Equipment Corp., 1977.*

- Now used by individuals in all sectors
  - Business, e.g. e-commerce, stock trading…
  - Research, e.g. drug discovery, grid, cloud computing…
  - Entertainment, e.g. games, film making, Xbox, Wii…
  - Personal, e.g. instant messaging, twitter, social networking…
  - Embedded, e.g. planes, cars, ZebraNet

What So Great About Computers?

Dramatically improve our working performance

Last 50 year ≈ Last 500 years

Classes of Computers

▪ Supercomputer	$ 5M ~ $ 20M (Blue Waters)
▪ Mainframe	$ 1M ~ $ 4M
▪ Minicomputer	$ 3K ~ $ 200K
▪ PC/workstation	$ 1K ~ $ 3K
▪ Network Computer (thin client)	$ 300 ~ $ 1K
▪ Smartphone, tablets	$ 100 ~ $ 300
▪ Embedded	$ 1 ~ 10

Why Study Computer Design?

It is exciting! And it affects all aspects of CS, ECE, many scientists, our society and YOU!

A big player in IT revolution

Old machines obsolete fast!

- New technology, e.g. denser ICs
- New demand, e.g. Web 2.0, Cloud Computing, Search Engines, Social Networking, Mobile Apps…
- Cost changes

Learn to deal with complexity via abstraction

- Problems may take months and years to complete

CS 250: 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

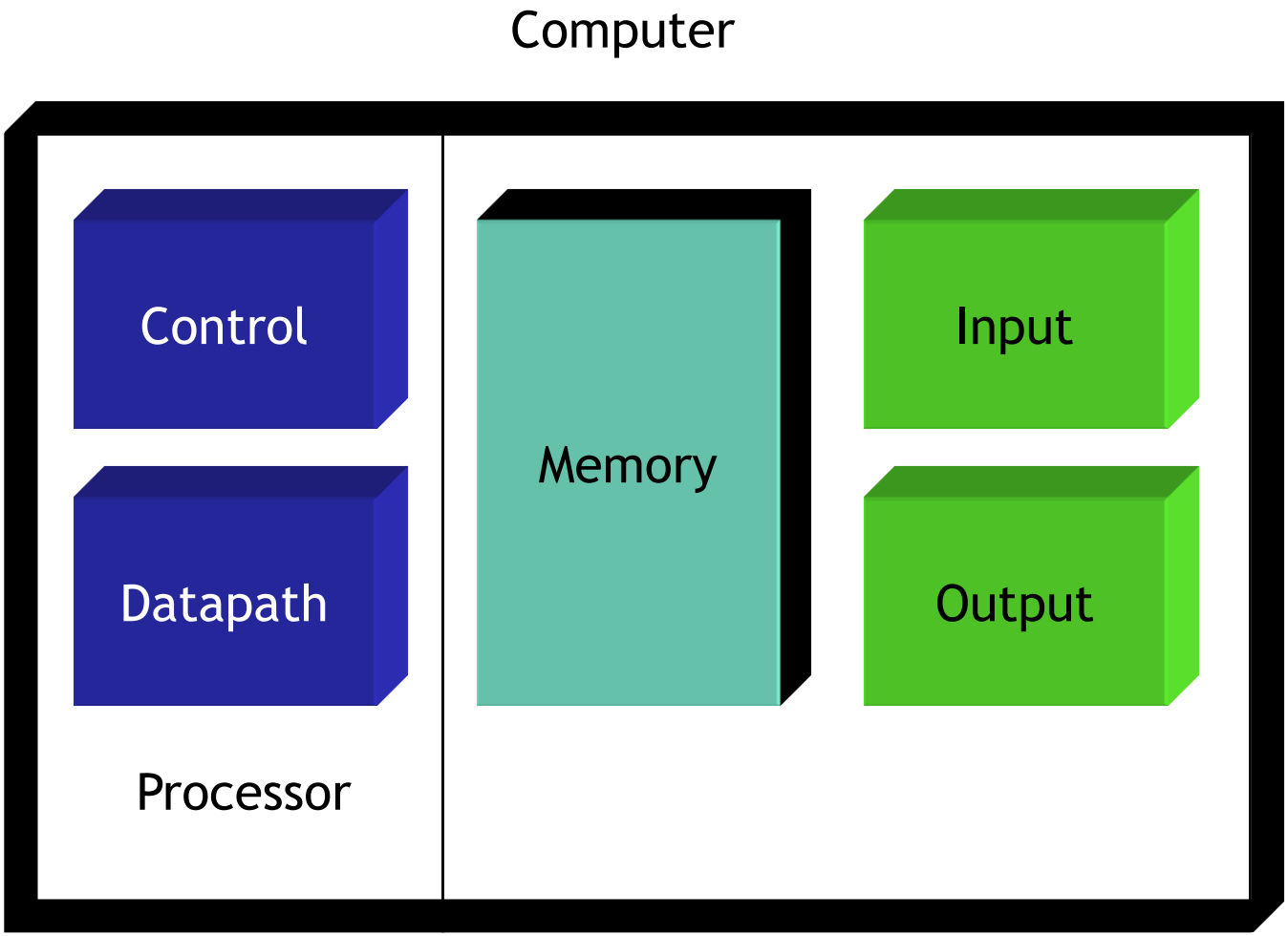
General hints to reach CS250 nirvana

- **Remember the big picture.**
  What are we trying to accomplish, and why?
- **Read the textbook and notes.**
  The book clear, well-organized, and well-written. The diagrams in the notes can be complex, but are definitely worth studying. Work through the examples and try some exercises on your own.
- **Talk to each other.**
  You can learn a lot from other CS250 students, both by asking and answering questions. But make sure to work on your written and lab assignments ***independently***.
- **Help us help you.**
  Come to lectures, labs and office hours. Send email or post on the Blackboard. Ask lots of questions! Check out the web page.

Basic Division of Hardware

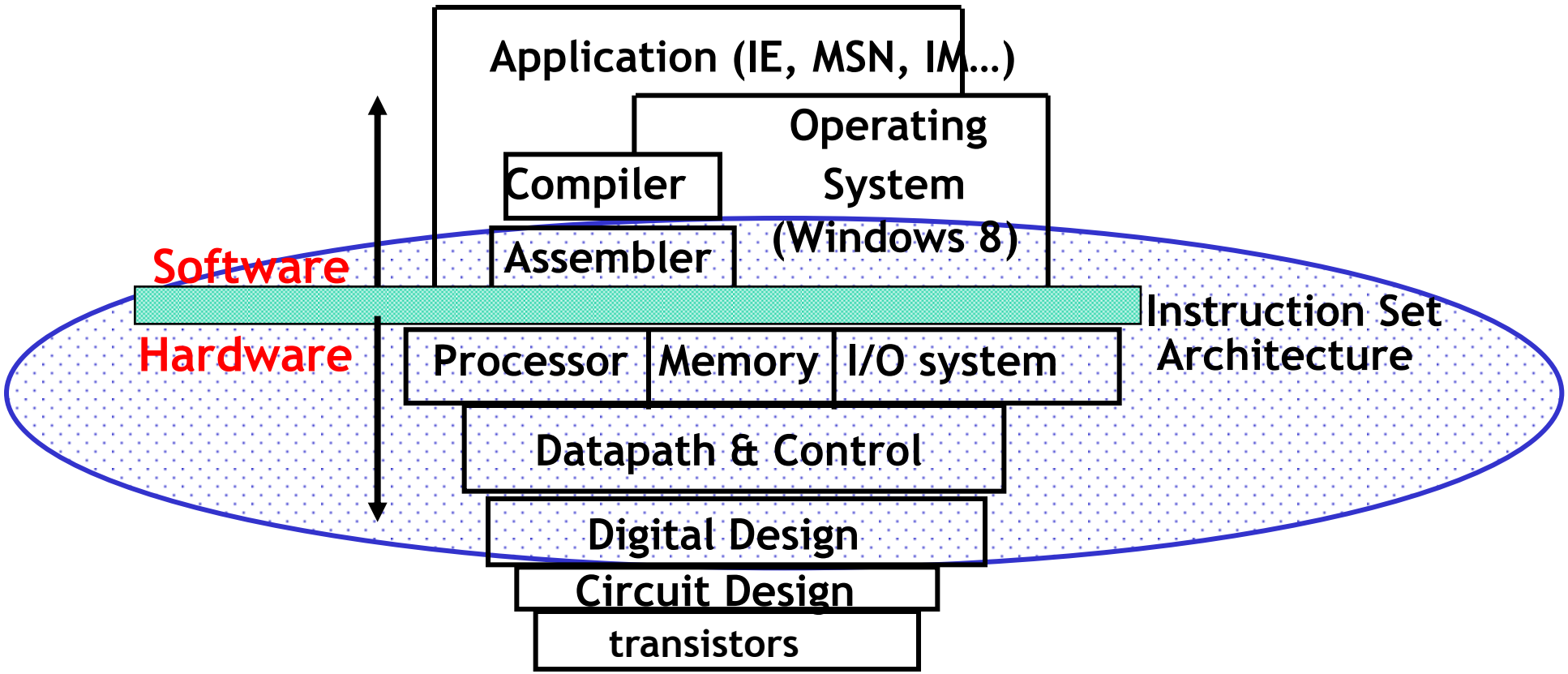
Five major components

- Control
- Datapath
- Memory
- Input
- Output



Computer, Structure, and Software/Hardware Interface

- Coordination of many *levels of abstraction:*



A Hierarchy of Abstractions

- Look at the computer system from top to bottom
  - Applications
    - Microsoft words
    - Games
  - Systems software
    - Compiler
    - Operating system
  - Hardware
    - Main topic of this course

One Important Abstraction in Computer Design

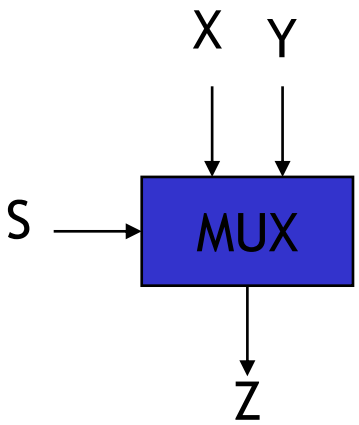
Instruction Set Architecture (ISA)

- Interfaces between hardware and software
- Hide (low-level) implementation details from high-level
- Why?

+ Designs at both levels can be independent
+ Make lives easier

Abstraction vs. Implementation

- Abstraction tells what
  - e.g. 2-to-1 MUX

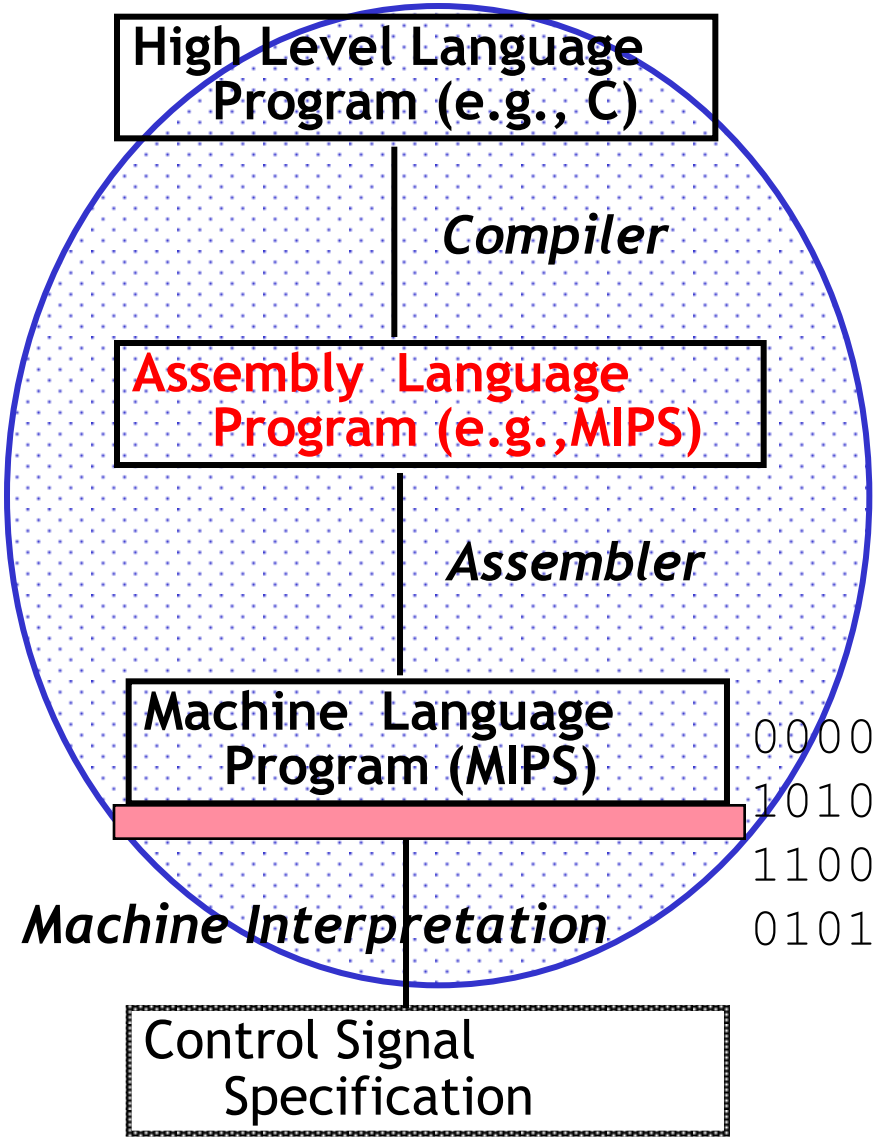


S	Z
0	X
1	Y

- Implementation tells how
  - One way to implement MUX: pass transistor

Remember: Abstraction ≠ Implementation

Computer Program: 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
```

How Does a Computer Work?

Example: (Components involved are in parenthesis)

- Load program (*input*, *memory*)
- Fetch instruction from memory (*control*, *datapath*, *memory**)
- Decode instruction (*control*, *datapath*)
- Fetch data (*control*, *datapath* , *memory**)
- Execute instruction (*control*, *datapath*)
- Store results (*control*, *datapath* , *memory**)
- File write (*memory*, *output*)

Why Performance Is Improved?

Improvement in technology

- Billions of transistors
- Higher clock frequencies
- Denser ICs

Improvement in *computer design* (based on *existing technologies*)

- Pipelined architecture
- Caching
- Parallelism
- More…

Exponential Advances in Speed and Capacity

(source: CS232 classnotes of Dr. S. Adve at UIUC)

Technology	Capacity	Speed
Logic transistors	4x to 5x in 3 years	4x in 3 years
DRAM	4x in 3 to 4 years	1.5x in 10 years
Disk	4x in 3 years	1.5x in 10 years

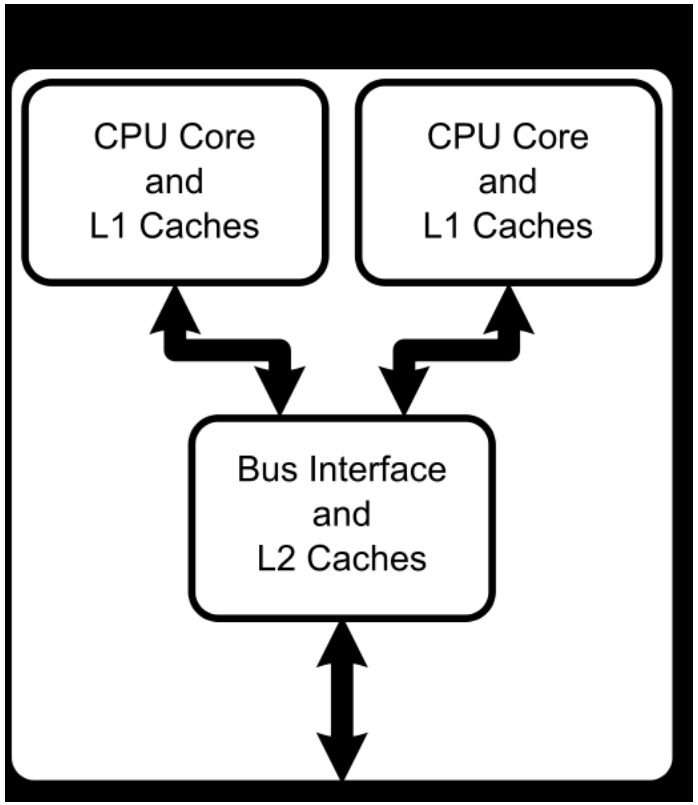
Performance Increase Over Last Decade

Moore's Law:

http://upload.wikimedia.org/wikipedia/commons/0/00/Transistor_Count_and_Moore%27s_Law_-_2008.svg

http://www.gotw.ca/images/concurrency-ddj.gif

Multi-core:



2 X 3GHz (=, >, or <) 6GHz

Buddy of Performance: Cost!

- Everything comes with a price
- One driving force behind the evolution of computer
- Evaluation should include both cost and performance
  - Largest machines ≠ the best cost/performance
- Any examples of cost/performance?

Summary

Computer architects must know both software & hardware

Abstraction in computer design

Cost and Performance