C335
Computer Structures

Introduction

Dr. Liqiang Zhang

Department of Computer and Information Sciences

Agenda

- Outline, general topics, course objectives
- Class logistics

What will we do in this course?

- We will look at the design of an instruction set for a simple processor.
  - The processor is based on a “real” processor, the MIPS R2000/R3000.
  - It's instruction set is summarized on the reference sheet in the front of the book.
- We will see how logic relates to switching (and transistors) and how logic forms a calculus for designing digital circuits.
- We will construct the basic logic blocks required to build a simple computer.
- We will look at the internal structure of that simple processor, having a 32 bit instruction length and a 32 bit data word.
- We will design the processor, and add enhancements to improve the speed of execution of its instructions.

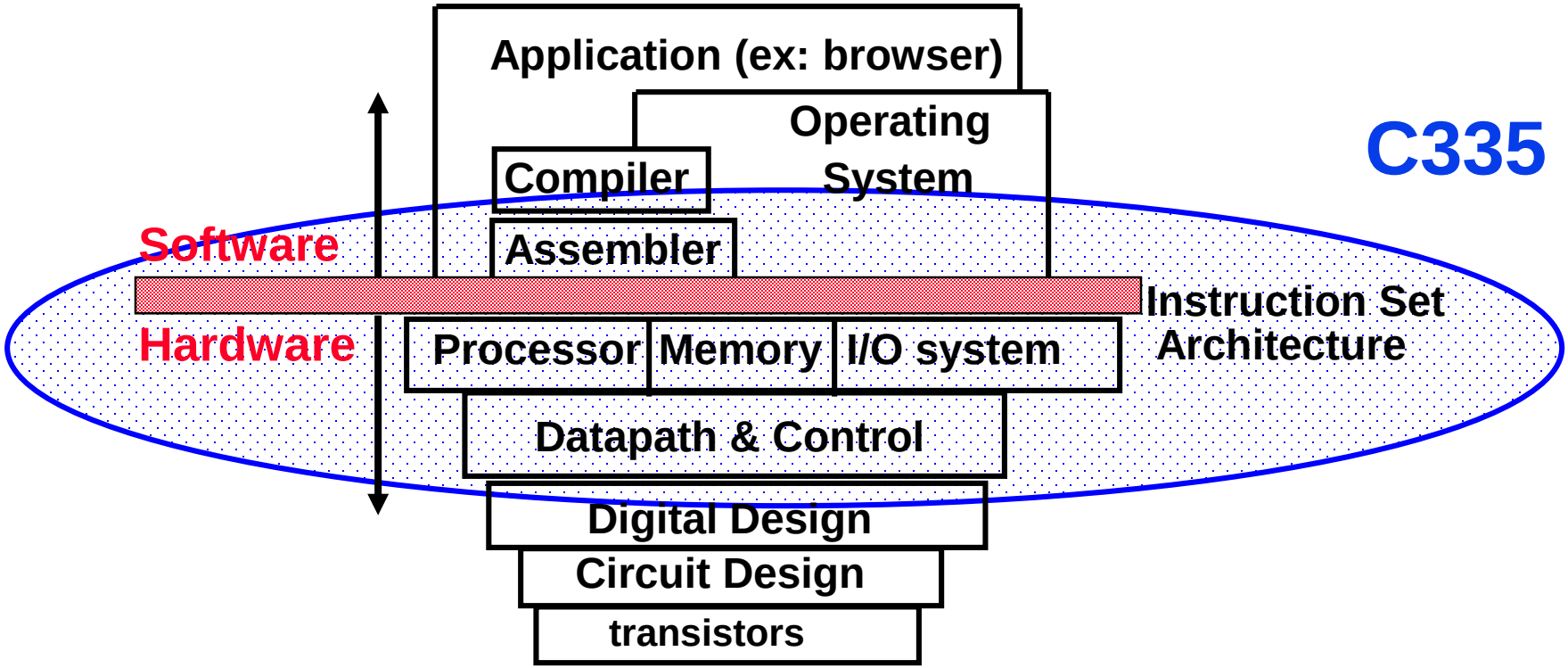
Why bother do we do this?

- Both software and hardware affect performance. Understanding how they interact is essential.
- Understanding how computers work helps us be better programmers.
- We may have to provide advice on which computer to purchase for some application.
- Computing performance has improved exponentially for over 50 years.
  - Why is the growth rate so fast?
  - How long can this continue?
  - How does this growth affect the programs I design?
  - How does it affect the value of hardware and software?
  - How does increased computation speed affect computer peripherals? (e.g., input/output devices.)

How do we learn?

- **This course is all about how computers work**
- **But what do we mean by a computer?**
  - Different types: embedded, laptop, desktop, server
  - Different uses: automobiles, graphics, finance, genomics…
  - Different manufacturers: Intel, Apple, IBM, Sony, Sun…
  - Different underlying technologies and different costs !
- **Analogy: Consider a course on "automotive vehicles"**
  - Many similarities from vehicle to vehicle (e.g., wheels)
  - Huge differences from vehicle to vehicle (e.g., gas vs. electric)
- **Best way to learn:**
  - Focus on a specific instance and learn how it works
  - While learning general principles and historical perspectives

What is “Computer Architecture/Structure”?



C335

* Coordination of many

levels (layers) of abstraction

C335 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 (Logisim, Verilog, etc.)

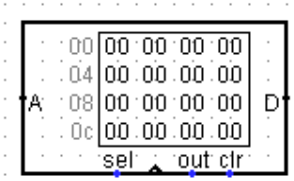
Architecture Implementation

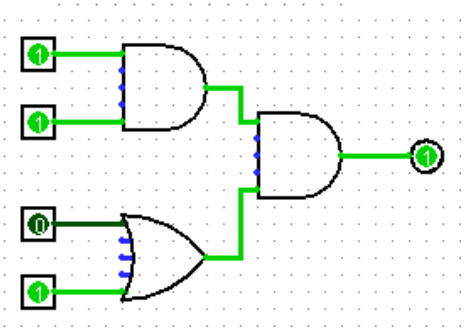
Logic Circuit Description (Logisim, etc.)

```
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
```





C335:So what's in it for me?

- In-depth understanding of the inner-workings of modern computers, their evolution, and trade-offs present at the hardware/software boundary.
  - Insight into fast/slow operations that are easy/hard to implement in hardware
- Experience with the *design process* in the context of a large complex (hardware) design.
  - Functional Specification --> Control & Datapath
- Learn how to design a correct single processor computer.
  - No magic required to design a computer
- Foundation for students aspiring to work in computer architecture.
- Others: solidifies an intuition about why hardware is as it is.

Tentative Topics (1/2)

- Computer Abstractions and Technology
- Instructions: Language of the Computer
  - introduction to the MIPS instruction set
  - addressing in the MIPS
  - relating architecture and instruction sets
- Assemblers, Linkers, and the SPIM Simulator
  - assemblers and linkers
  - procedure call conventions
  - The SPIM simulator
  - The MIPS assembly language
- logic design
  - combinational logic and logic design
  - sequential logic, memory, and state machines
- Arithmetic elements and the datapath
  - numbers and their representation (integers and reals)
  - design of simple datapath elements and an ALU

Tentative Topics (2/2)

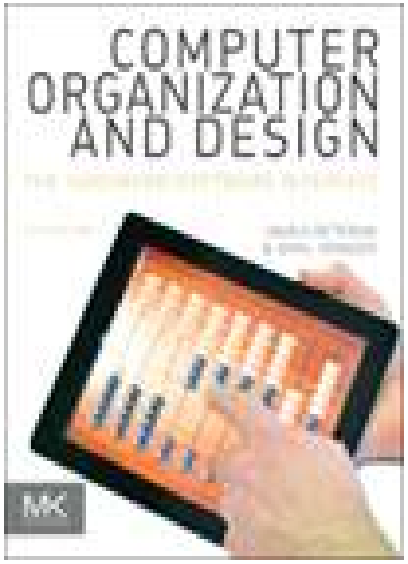
- Designing a processor
  - a single cycle datapath implementation
  - implementing the control unit
  - pipelining
  - data and resource dependencies
  - hazards
- Exploiting Memory Hierarchy
  - basics of cache
  - measuring and improving cache performance
- Basics of I/O
  - Exceptions and Interrupts
- Parallel Processors and GPUs

Class Logistics

Instructor:	Dr. Liqiang Zhang Office: Northside Hall – 331 Phone: (574)520-4297 Email: liqzhang@iusb.edu URL: http://www.cs.iusb.edu/~liqzhang
Class Meeting:	5:30pm -- 7:20pm (TR) NS-135
Office Hours:	1:30pm – 2:30pm, 3:45pm – 5:30pm (TR) **Virtual office hours**: 4:00pm – 5:00pm (TR) (**http://connect.iu.edu/officeonline**) Or by appointment
Course Management:	Login to **Canvas (http://canvas.iu.edu)** for schedule, announcements, lecture notes, assignments, due dates, grades and others.

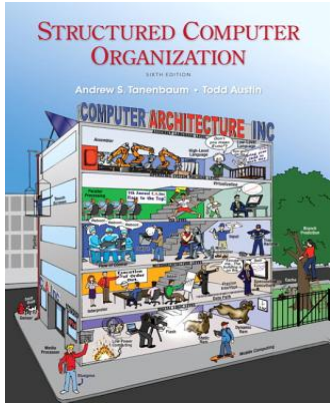
Class Logistics

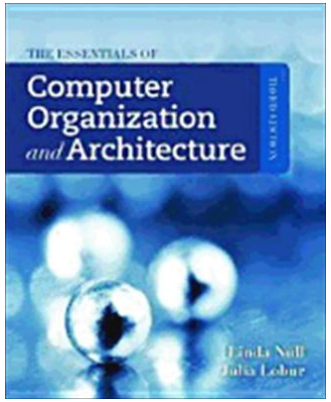
- **Required textbook:**
  - ***Computer Organization & Design, The Hardware/Software Interface, Fifth Edition,***
    - ***David Patterson & John Hennessy, Morgan Kaufmann publisher, 2013,***
    - ***ISBN-13: 978-0124077263***



Class Logistics

- **References:**
  - ***Structured Computer Organization,*** *Fifth Edition*,
    - Sixth Edition, Andrew Tanenbaum and Todd Austin, Prentice Hall, 2012, ISBN: 978-0-13-291652-3





  - ***The Essentials of Computer Organization and Architecture*** *Third Edition, Linda Null & Julia Lobur, Jones and Bartlett Publishers, 2010, ISBN: 978-1-44-960006-8*

Class Logistics

- **Grading scheme:**
  - Your final grade will be based on the sum of following parts:

Homework & project	30%
Quizzes	5%
Two Midterms	40%
Final Exam	20%
Attendance	5%
Total	**100%**

  - The final grade will be distributed as:

[90-100%]	A
[80-90%]	B
[70-80%]	C
[60-70%]	D
[0-60%]	F

Class Policies

- **Attendance:** attend all classes! why?
  - The lecture notes are not a complete study aid. In short, reading the course notes is not a substitute for class attendance!
  - Important info will be given in class

- Be active in class
  - Ask and answer questions, why?
  - Important part of attendance
  - Possible final grade curve/bonus

Class Policies

- **Homework assignments:** submit on time!

Within 24 hours after the due date (counted from the start of the class):	-10%
After 24 hours of the due date, but within 48 hours after the due date:	-20%
Beyond 48 hours after the due date:	You lose all the points for that assignment!

- **Reading assignments:**

Class Policies

- **Policy on Incompletes:**
  - The granting of an Incomplete in this course is highly discouraged. Thus, it will be done only in very rare cases and will conform fully to the college policy. Keep up with the course and notify the instructor promptly of any problems. You will receive more help if you act early.

Class Policies

- **Use of cell phones, notebook computers and other electronic devices:**
  - During lectures and labs all cell phones and portable electronic devices must be turned off or put in “silent” mode. The use of computers and phones of any kind during lectures is not allowed unless special permission has been granted by the Instructor.

Class Policies

- **Exams/Quizzes make-ups:**

- **Academic integrity:** NO cheating!
  - Please read all the related issues at http://www.iusb.edu/~judicial/.

- **Disabilities notice:**

For Online Students

- **Lecture Videos**
- **Time Management**
- **Participation**
- **IU Ready (http://ready.iusb.edu)**

Introducing yourself

- Introduce yourself (name, major, and else…)
- Any experience related to assembly programming / hardware that you want to share…
- Mind to share your study/career plan for the future? How do you think this class will help that?
- Any questions regarding the course?