C335/A593

Computer Structures

Dr. David R. Surma
Department of Computer and Information Sciences
Introduction

- Outline, general topics, course objectives
- Course Syllabus
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.
  - Its instruction set is summarized on the reference sheet in the front of the textbook.

- 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.
What will we do in this course?

- 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.

Assembly Programming using MIPS R3000 CPU

What is MIPS R3000 Processor?

- A 32-bit RISC CPU developed by MIPS Technologies Inc.,
- Used for high-performance desktops such as workstations
- Many vendors manufacture the chip (NEC, IDT, Toshiba)

IDT = Integrated Device Technology
Why do we bother studying this?

- Because computer architecture is perhaps the most fundamental subject in computer science. Without computers the field of computer science does not exist!

- Both software and hardware affect performance. Understanding how they interact is essential.

- Knowing what is inside a computer and how it works will help you design, develop and implement applications better, faster, cheaper, and more efficiently.
Why do we bother studying this?

- Understanding how computers work helps us to become 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, transmission)
  - 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. Apply principles to new application.
What is “Computer Architecture/Structure”? 

* 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.
Class Logistics

Instructor: Dr. David R. Surma
Office: Northside Hall – 339
Phone: (574)520-4412
Email: dsurma@iusb.edu
Course Website: http://mypage.iu.edu/~dsurma

Class Meeting: 9:35am -- 11:25am (TR) – held via Zoom

Office Hours: 1:00pm – 3:00pm (W), 12:30pm – 2:30pm (R) (virtual)
Others by appointment

Course Management: The course will use Canvas (http://canvas.iu.edu) and the course website (pages.iu.edu/~dsurma).
Required textbook:

  - David Patterson & John Hennessy, Morgan Kaufmann publisher, 2013,
References:

- **Structured Computer Organization, Fifth Edition,**

- **The Essentials of Computer Organization and Architecture**
Course Syllabus

- Posted to Canvas and available via the course website.