CS 231

Computer Organization

Dr. William H. Robinson
August 25, 2004

http://eecs.vanderbilt.edu/courses/cs231/

Topics

“No! Layers! Onions have layers! Ogres have layers!
Onions have layers. You get it? We both have layers.”

– Shrek

• Icebreaker
  – Roll call. Tell one interesting fact about yourself.
• Course Syllabus
• Textbook
• Course Overview
• Assignment
  – Read Chapter 1 in Tanenbaum

CS 231: Computer Organization

Class Time and Location: MWF 9:10am – 10am, 306 FGH

| Instructor: | William H. Robinson |
| Office Address: | 248 Featheringill Hall |
| Office Phone: | 322-1507 |
| E-Mail: | william.h.robinson@vanderbilt.edu |
| Office Hours: | 10am – 11:30am, Mondays and Wednesdays |

| Teaching Assistant: | Jason Tan |
| Office Address: | 230 Featheringill Hall |
| Office Phone: | 322-8496 |
| E-Mail: | jason.tan@vanderbilt.edu |
| Office Hours: | 3pm – 4:30pm, Tuesday
  4pm – 5:30pm, Thursday |

• Prerequisite: CS 201
• Corequisite: EECE 116

Grading Policy

• 7% Homework and quizzes
• 28% Team Programming Assignments
• 15% Exam I
  – Wednesday, September 22
• 15% Exam II
  – Wednesday, October 13
• 15% Exam III
  – Wednesday, November 17
• 20% Cumulative Final Exam
  – Saturday, December 11 (9am – 12pm)

If necessary, each exam average is curved up to 75.
I will not curve down!
General Class Policy

- Class attendance is required. Students are responsible for all material covered in class and posted online.
- Quizzes can be both announced and unannounced. A missed quiz will be recorded as a zero.
- All assignments are due at the start of class on the specified due date. Late assignments will NOT be accepted.
- Exams are taken at the scheduled class time or at the scheduled final period. A missed exam will be recorded as a zero.
- The Honor Code of Vanderbilt University applies to all work done in CS 231.
- I encourage questions during lectures, office hours, or via e-mail.

“Write It Before Them…”

- **Required Text**

- **Reference Text**

The Big Picture

- Since 1946 (ENIAC I) all computers have had 5 components

<table>
<thead>
<tr>
<th>Processor</th>
<th>Memory</th>
<th>Input</th>
<th>Output</th>
</tr>
</thead>
<tbody>
<tr>
<td>Control</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Datapath</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Sample Organization

- Pentium III Chipset
  - Busses
  - Memory
  - Caches
  - I/O Devices: Disks, Displays, Keyboards, Networks
  - Controllers

- All have interfaces & organizations
- Communication is important

Adapted from John Kubiatowicz’s CS 152 lecture notes. Copyright © 2003 UCB.
Computer Architecture Is …

the attributes of a [computing] system as seen by the programmer, i.e., the conceptual structure and functional behavior, as distinct from the organization of the data flows and controls the logic design, and the physical implementation.

Amdahl, Blaaw, and Brooks, 1964
Computers Have Layers, Too

- **Structured Computer Organization**
  - Uses a series of abstractions
  - Each abstraction builds on the one below it

- **Course Objective**
  - Provide students with an understanding of the components of a computer and how they work

---

Structured Computer Organization

<table>
<thead>
<tr>
<th>Level</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td>Problem-oriented language level</td>
</tr>
<tr>
<td>4</td>
<td>Translation (compiler)</td>
</tr>
<tr>
<td>3</td>
<td>Assembly language level</td>
</tr>
<tr>
<td>2</td>
<td>Translation (assembler)</td>
</tr>
<tr>
<td>1</td>
<td>Operating system machine level</td>
</tr>
<tr>
<td>0</td>
<td>Partial interpretation (operating system)</td>
</tr>
<tr>
<td></td>
<td>Instruction set architecture level</td>
</tr>
<tr>
<td></td>
<td>Interpreted (microprogram) or direct execution</td>
</tr>
<tr>
<td></td>
<td>Machine level</td>
</tr>
<tr>
<td></td>
<td>Microarchitecture level</td>
</tr>
<tr>
<td></td>
<td>Hardware</td>
</tr>
<tr>
<td></td>
<td>Digital logic level</td>
</tr>
</tbody>
</table>

---

L0 – Digital Logic

- Composed of transistors
- Implements Boolean functions
- Known as “gates”

\[
\begin{align*}
F &= A \land B \\
F &= A + B \\
F &= A \oplus B
\end{align*}
\]

---

L1 – Microarchitecture

- Includes registers & ALU
- Forms a datapath
- Can perform simple arithmetic operations
L2 – Instruction Set Architecture

- ISA is the Hexadecimal column
- Sometimes called machine language

<table>
<thead>
<tr>
<th>hex</th>
<th>mnemonic</th>
<th>meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>0x10</td>
<td>BR/SH Byte</td>
<td>Push byte onto stack</td>
</tr>
<tr>
<td>0x08</td>
<td>CPR</td>
<td>Copy top 8 bits from stack</td>
</tr>
<tr>
<td>0x07</td>
<td>COC</td>
<td>Copy all contents of stack</td>
</tr>
<tr>
<td>0x06</td>
<td>DCOP</td>
<td>Pop two words from stack</td>
</tr>
<tr>
<td>0x03</td>
<td>JAN&lt;0</td>
<td>Pop two words from stack, push Boolean AND</td>
</tr>
<tr>
<td>0x02</td>
<td>JLL offset</td>
<td>Pop two words from stack, push branch if less than zero</td>
</tr>
<tr>
<td>0x01</td>
<td>JLS offset</td>
<td>Pop two words from stack, push branch if less than zero</td>
</tr>
<tr>
<td>0x00</td>
<td>JLSR offset</td>
<td>Pop two words from stack, branch if sign</td>
</tr>
<tr>
<td>0x84</td>
<td>LREG variable</td>
<td>Add a constant to a local variable</td>
</tr>
<tr>
<td>0x80</td>
<td>LJAC variable</td>
<td>Push current variable off stack</td>
</tr>
<tr>
<td>0x78</td>
<td>LDEC variable</td>
<td>Make a reference</td>
</tr>
<tr>
<td>0x60</td>
<td>LDR</td>
<td>Pop two words from stack, push Boolean OR</td>
</tr>
<tr>
<td>0x50</td>
<td>LLET variable</td>
<td>Pop from memory with a logical value</td>
</tr>
<tr>
<td>0x30</td>
<td>LSTORE variable</td>
<td>Pop two words from stack and store in local variable</td>
</tr>
<tr>
<td>0x10</td>
<td>LSUB</td>
<td>Pop two words from stack, push their difference</td>
</tr>
<tr>
<td>0x04</td>
<td>LCON</td>
<td>Push constant from constant pool onto stack</td>
</tr>
<tr>
<td>0x00</td>
<td>NOP</td>
<td>Do nothing</td>
</tr>
<tr>
<td>0x02</td>
<td>POP</td>
<td>Delete word off top of stack</td>
</tr>
<tr>
<td>0x0F</td>
<td>NEQ</td>
<td>Branch if two top values on the stack</td>
</tr>
<tr>
<td>0x04</td>
<td>WIDE</td>
<td>Push instruction; next instruction has a 16-bit index</td>
</tr>
</tbody>
</table>

Other Levels

- L3: operating system machine level
  - Not covered in the class
- L4: assembly language level
  - Will use with ISA
- L5: problem-oriented language level
  - May use for programming assignments

Course Content

- Basics of computer architecture
- Computer components
- Boolean Algebra
- Digital logic design
- Instruction Set Architecture (ISA) design
- Microcode
- Assembly language
- Etc. (as time permits)

What Skills Will You Learn?

- Identify the layers or levels of a computer used in structured computer organization
- Recognize the interconnected system of processors, memories, and input/output devices that comprise digital computers
- Describe the Boolean Algebra of the basic elements and fundamental circuits from which all computers are constructed
- Analyze the microcode that implements a particular Instruction Set Architecture (ISA)
- Evaluate the cost versus performance question as it pertains to microcode
- Describe methods used to improve system (primarily CPU and memory) performance
Take a Moment…

• Write your thoughts/reactions to our first day
  – Anonymous feedback

• I will stay a few minutes for additional questions

• Reading assignment for Friday:
  Tanenbaum Ch. 1