Course Outline

Introduction to Computer Systems COMP 273

(Winter 2012; TJ 4:05:-5:25; ENGTR 100)

Instructor:	Professor Michael Langer
Office:	McConnell Engineering, rm. 329
Tel:	514-398-3740
Email:	langer@cs.mcgill.ca
Course Web Page:	www.cim.mcgill.ca/~langer/273.html
Office Hours:	TBD

Teaching Assistants (T.A.)	TBD (see Course Web Page)
Office hours:	TBD

Introduction

The course gives a bottom up view of how a computer works. It begins with a overview of digital logic, and then builds up the main architectural and system elements of a typical modern computer. We use a specific RISC computer architecture and assembly language, MIPS, to illustrate the main concepts.

Topics and Lecture Schedule

The course material is divided into three parts.

1. Digital Logic (7 lectures)
   - Number representations
     - binary, twos complement, floating point, hexadecimal
   - Combinational logic
     - truth tables, gates, adders, encoders, decoders, multiplexors, ROM
   - Sequential logic
     - latches, flop flops, registers, integer multiplication and division.
2. MIPS assembly language and CPU architecture (10 lectures)
   - instruction representations, absolute vs. immediate addressing
   - procedures, the stack, recursive procedures
   - SPIM simulator: directives, pseudoinstructions
   - multiplication, division, floating point instructions

- datapath and control
- fetch-execute, microinstructions vs. pipelining
- exceptions and the kernel

3. Memory and I/O (7 lectures)
    - RAM,
    - virtual memory, page tables, TLB
    - cache
    - system bus
    - interrupts and exceptions, memory mapped I/0, direct memory access
    - synchronous vs. asynchronous I/0

Lecture Notes

All material covered in the lectures will be made available as PDFs on the course web page.

Reference Textbooks

There is no textbook for the course. If you wish to do further background reading , then I would recommend the following which is available on *two hour reserve in the Schulich Library.* Call Numbers can be found from the McGill libraries website (see MUSE, Course Reserves).

For further details on MIPS, see:

- "Computer organization and design: the hardware/software interface" by David A. Patterson and John L. Hennesey. *This is the usual textbook for COMP 273.* A fourth edition that came out this year.
- Sweetman, Dominic "See MIPS run" San Francisco, Calif. : Morgan Kaufmann Publishers, 2002.

For further details on other topics covered in the course, see:

- Katz, Randy H., "Contemporary logic design" Redwood City, Calif. : Benjamin/Cummings, 1994
- Williams, Rob, "Computer systems architecture : a networking approach" Harlow, England ; New York : Addison-Wesley, 2001.
- Heuring, Vincent P. "Computer systems design and architecture" Upper Saddle River, N.J. : Prentice Hall, 2004.
- Carpinelli, John D. "Computer systems organization & architecture" Boston ; London : Addison-Wesley, 2001.

Prerequisites

- COMP 206 *Introduction to Software Systems*

This is the official prerequisite for the course. If you have not taken this course but you have some experience with C or C++ programming (or if you did very well in COMP 250), then you should be ok without it.

Evaluation

- 20 % In-class test (first week of February)
  - covers Part 1 (Digital Logic). Practice exercises and solutions will made available.
- 30 % Programming Assignments ($3 \times 10\%$ – in February)
  - There will be three programming assignments using the MIPS assembly language, specifically the SPIM simulator.
- 50 % Final Exam (during Final Examination Period in April)
  - Final Exam concentrates on MIPS and and Memory/I0.
  - There will be *no* 100 % Final Exam option.

In accord with McGill University's Charter of Students' Rights, students in this course have the right to submit in English or in French any written work that is to be graded.

Policy on collaboration

We greatly encourage you to discuss the assignment problems with each other. However, this discussion must not go so far that you are revealing the solutions to each other (and it certainly should not go so far that you are sharing code). Try to follow a simple guideline: any discussion you have about an assignment should be *open* in the sense that you would be 100% comfortable if anyone else *including the instructor* were listening in.

McGill policy on academic integrity

McGill University values academic integrity. Therefore, all students must understand the meaning and consequences of cheating, plagiarism and other academic offenses under the Code of Student Conduct and Disciplinary Procedures. See `http://www.mcgill.ca/students/srr/honest/` *for more information*