

COURSE DESCRIPTION

Department and Course Number CMPS 351 **Course Coordinator** Parvin Hashemian

Course Title Computer Organization and Assembly Language Programming **Total Credits** 3

URL <http://www.ucs.louisiana.edu/~pxh3930> **Semester hours** 3

Current Bulletin Description

Overview of computer organization. Performance evaluation, MIPS architecture, assembly and machine language, data representation, hardware/software interface, assembly and linking process, implementation of data path. FA, SP. Prereq: CMPS 260 and EECE 140 both with a minimum grade of C.

Textbook

John L. Hennessy, and David A. Patterson, *Computer Organization and Design, the Hardware/Software Interface*, 3rd Edition, Morgan Kaufmann Publishers, Inc., 2004.

References

1. Linda Null, and Julia Lobur, *The Essentials of Computer Organization and Architecture*, 3rd Edition, Jones & Bartlett Publishers, Inc., 2006.
2. William Stallings, *Computer Organization and Architecture: Designing for Performance*, 6th Edition, Prentice Hall, 2003.

Course Objectives

- To learn MIPS assembly language programming.
- To understand the fundamental organization of modern computers.

Course Outcomes

- To write programs in MIPS assembly language.
- To understand the process of translation from assembly language to machine
- Understand how the datapath and control unit operate inside a CPU.
- Measure and compare performances of different systems.

Prerequisites by Topic

- Number systems and conversions.
- Knowledge of a programming language.

Major Topics Covered in the Course

Classes

1. Computer abstractions and technology 1-2
2. Number systems (decimal, hexadecimal, binary) and basic operations 1
3. Introduction to MIPS architecture and SPIM simulator 1
4. MIPS assembly instructions;
 - arithmetic instructions (add, sub)

- data transfer instructions (lw, sw)
- branch instructions (beq, bne, j)
- comparison instructions – slt
- implementation of if-then-else, loop constructs into MIPS assembly instructions
- assembler directives
- logical operation 4
- 5. Translation from assembly language to machine language;
 - R- format
 - I- format
 - J- format 3
- 6. Compilation of high level language constructs into MIPS instructions
 - switch-case
 - function/procedure calls
 - recursion 3
- 7. Assembly, linking, and loading 2
- 8. Optimization of assembly code 1
- 9. Performance evaluation of computers 2-3
- 10. Adder design, ALU design 2
- 11. Implementation of a data-path 4
- 12. Introduction to Pipelining 1-2

Laboratory projects (specify number of weeks on each)

This course has several hand written and assembly language programming assignments. The students are required to implement a subset of MIPS assembly instructions in a high-level programming language (C++).

Oral and Written Communications

Every student is required to submit at least 0 written reports (not including exams, tests, quizzes, or commented programs) of typically 0 pages and to make 1 oral presentations of typically 10-15 minutes duration. Include only material that is graded for grammar, spelling, style, and so forth, as well as for technical content, completeness, and accuracy.

Social and Ethical Issues

Please list the topics that address the social and ethical implications of computing covered in all course sections. Estimate the class time spent on each topic. In what ways are the students in this course graded on their understanding of these topics (e.g., test questions, essays, oral presentations, and so forth)?

N.A.

Theoretical Content

Please list the types of theoretical material covered, and estimate the time devoted to such coverage.

About one tenth of the lectures are spent on performance evaluation of systems.

Problem Analysis

Please describe the analysis experiences common to all course sections.

The students are required to solve problems on performance evaluation of computer systems and to analyze the term project.

Solution Design

Please describe the design experiences common to all course sections.

The students should come up with their implementation of appropriate functions to implement the subset of MIPS instructions for the term project