

CS350 -Computer Organization and Assembly Language Programming

Last Updated - 03/11/02

Course Manager – Dr. Xiang-Yang Li, Assistant Professor

3 credit hours; required for CS & CPE (or ECE242 allowed for CPE); 100 min. lecture & 100 min. lab each week

Current Catalog Description - Introduction to the internal architecture of computer systems - including micro, mini-, and mainframe computer architectures. Focuses on the relationship between a computer's hardware, its native instruction set, and the implementation of high-level languages on that machine. Uses a set of assembly language programming exercises to explore and analyze microcomputer architecture. Prerequisite: CS 106 or CS 200. (2-2-3) (C)

Textbook

- D. Patterson and J. Hennessy, *Computer Organization and Design: The Hardware/Software Interface*, Second Edition, 1998 ISBN: 1-55860-428-6.
- J. Morris Chang, *CS350: Computer Organization and Assembly Language Laboratory Manual*, IIT CS department published

References - other textbooks or materials

- Y. Patt and S. Patel, *Introduction to Computing Systems: From Bits and Gates to C and Beyond*, 1999
- A. Tanenbaum, *Structure Computer Organization*, 1998
- Hennessy and Patterson, *Computer Architecture: A Quantitative Approach*, second edition
- William Stallings, *Computer Organization and Architecture*, fifth edition

Course Goals - Students should be able to:

- Explain the basic organization of the classical von Neumann machine and its major functional units.
- Explain how an instruction is executed in a classical von Neumann machine.
- Summarize how instructions are represented at both the machine level and in the context of a symbolic assembler.
- Explain different instruction formats, such as addresses per instruction and variable length vs. fixed length formats.
- Write simple assembly language program segments.
- Demonstrate how fundamental high-level programming constructs are implemented at the machine-language level.
- Explain the basic concepts of interrupts and I/O operations.
- Digital logic and digital systems
 - Use mathematical expressions to describe the functions of simple combinational and sequential circuits.
 - Discuss the physical limitations of electronic circuits.
- Machine level representation of data
 - Explain the pros and cons of using different formats to represent numerical data.
 - Convert numerical data from one format to another.
 - Discuss the effects of fixed-length number representations on accuracy and precision.
 - Describe the internal representation of characters, strings, records, and arrays.
- Memory system organization and architecture
 - Identify the main types of memory technology.
 - Explain the effect of memory latency on running time.
 - Explain the use of memory hierarchy to reduce the effective memory latency.
- Interfacing and communication
 - Identify various types of buses in a computer system.
 - Describe data access from a magnetic disk drive.
- Functional organization

- Compare alternative implementation of datapaths.
- Discuss the concept of control points and the generation of control signals using hardwired or microprogrammed implementations.
- Communicate, in written form, the design and experimental results for the project that integrates all topics covered in this course.

Prerequisites by Topic

- Basic knowledge of computers and programming languages

Major Topics Covered in Course

1. Computer organization, history and future of computer architecture	3
2. Number systems (decimal, hexadecimal, binary) and basic operations	3
3. ASCII code and Byte ordering	1
4. Introduction to MIPS Architecture	1
5. MIPS instructions -ALU instructions Add, Sub, Or, And; Memory Structure; Load Store Architecture - Lw, Sw, Lb, Sb; Branching Instructions - Beq, Bne, J, Jr, Jal; Comparison - Sl; Stack Implementation; Compilation of C structures into MIPS instructions - If, If-Else, For loop, Switch-Case, Procedure Calls, Recursion; Instruction formats - Rtype, Itype, Jtype; Calculation of Target addresses for Conditional and Unconditional Branches; Arrays vs. Pointers in memory accesses	16
6. CPU design (logic design, component design, datapath) - Logic gates - Not, And, Or, Nand, Nor, 15 Xor; 1bit Adder design; 8bit Ripple Carry Adder Design; 4bit Look Ahead Carry Adder Design 1bit ALU design; 8bit ALU design; Register File organization; Program Counter and Instruction Register design; Decoder design; Memory path - Instruction and Data; System Clock; Conditional and Unconditional branch control; Multiplication and Division Algorithms; Multiplication and Division Hardware design	12
7. CPU design project	2
8. Performance Issues	2
Project Presentation	5 hours
Class Admin, Midterm Exam #1, #2	-
Final Exam	60 hours

Laboratory projects (specify number of weeks on each)

- 10 labs (1 lab need about a week, each lab contains multiple design assignments), the student has to do the lab individually.
 - 1) Overview of PCSPIM; 2) Introduction to instruction sets and converting a programming language into assembly language; 3) Memory structure of MIPS, base-displacement addressing mode, and simulating some program (like “C”) into a PCSPIM program; 4) Array representation and management; 5) Base-displacement for structures (branch and jump); 6) Stacks in MIPS; 7) Procedure calls in MIPS; 8) 1 Bit Full Adder; 9) Four-bit ripple carry adder; 10) 8-bit ALU;
- 1 semester-ending project that integrates all topics covered in the course. The project is done in groups of up to 4 students. The project involves the design of a simple 8-bit processor that can perform basic arithmetic and logic operations based on RISC technology. Project detailed outline is provided on the CS 350 course web site. The students should write a report about their design and experimental results. Some of the groups will be selected to give oral presentation of their projects in the classroom.

Estimate CSAB Category Content in Credit Hours

	CORE	ADVANCED		CORE	ADVANCED
Data Structures	0		Computer Organization and Architecture	2.5	
Algorithms	0		Concepts of Programming Languages	0.5	

Oral and Written Communications - Every student is required to submit at least __1__ written reports (not including exams, tests, quizzes, or commented programs) of typically __5-6__ pages and to make __1__ oral presentations of typically __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.

- Communicate, in written form, the design and experimental results for the project that integrates all topics covered in this course.
- The oral presentation is only required for some selected groups when it is impossible to schedule all groups for presentations due to the large amount of students.

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)?

- none

Theoretical Foundations - Please list the types of theoretical material covered, and estimate the time devoted to such coverage in contact (lecture and lab) hours.

- Recursion, 2 hours

Problem Analysis - Please describe the problem analysis experiences common to all course sections.

- Analyze the design results obtained from the designing of different computer component and different methods.

Solution Design - Please describe the design experiences common to all course sections.

- There is a semester-ending project that integrates all topics covered in the course. The project is done by groups. Each group contains at most 4 students. They have to write a written report for their project and some groups have to give an oral presentation of their project.

Other Course Information

- Additional Suggested Course Assignments
 - 2 midterm exams (50 minutes each. The first mid-term contains around 90% programming questions, while the second mid-term contains about 50% programming questions and 50% architecture design questions.)
 - 1 final exam (120 minutes, around 20% programming questions and 80% architecture design questions. For architecture-related questions, there are three types of questions: computational questions, architecture-design questions, and architecture explanation questions)
- Planned Course Enhancements
 - None