CS 242: COMPUTER ARCHITECTURE AND ORGANIZATION

Citrus College Course Outline of Record

Heading	Value
Effective Term:	Fall 2021
Credits:	3
Total Contact Hours:	54
Lecture Hours :	54
Lab Hours:	0
Hours Arranged:	0
Outside of Class Hours:	108
Prerequisite:	CS 225 (or concurrent enrollment).
Transferable to CSU:	Yes
Transferable to UC:	Yes - Approved
Grading Method:	Standard Letter

Catalog Course Description

The organization and behavior of real computer systems at the assembly-language level. The mapping of statements and constructs in a high-level language onto sequences of machine instructions is studied, as well as the internal representation of simple data types and structures. Numerical computation is examined, noting the various data representation errors and potential procedural errors. 54 lecture hours.

Course Objectives

- Design and implement assembly language programs using machine instructions, addressing modes, subroutines, macros and interrupts to solve problems of simple complexity.
- Investigate how fundamental high-level programming constructs are implemented at the machine-language level.
- Compare and contrast the basic relationship between hardware and software design.
- Analyze the basic computer architecture using engineering principles and quantitative cost/performance trade-offs.
- Design an ALU by understanding the basics of computer binary arithmetic and logic including addition, multiplication and division floating point representation.
- Illustrate the basics of logic design including gates, truth tables and combinatorial logic.

Major Course Content

1. Bits, bytes, and words
2. Numeric data representation and number bases
3. Fixed- and floating-point systems
4. Signed and twos-complement representations
5. Representation of nonnumeric data (character codes, graphical data)
6. Representation of records and arrays
7. Basic organization of the von Neumann machine
8. Control unit; instruction fetch, decode, and execution
9. Instruction sets and types (data manipulation, control, I/O)
10. Assembly/machine language programming
11. Instruction formats
12. Addressing modes
13. Subroutine call and return mechanisms
14. I/O and interrupts

Suggested Reading Other Than Required Textbook

The student will visit several programming online websites in order to read documentation about object oriented programming languages.

Examples of Required Writing Assignments

The student will create a flowchart and a pseudocode before implementing the programming code for any given assignment.

Examples of Outside Assignments

Students will be required to complete the following types of assignments outside of the regular class time:
- Study course concepts - Answer various programming questions - Practice skills (i.e., writing programs and creating flowcharts). - Read required materials - Solve programming problems - Create programs that apply assembly language programming techniques

Instruction Type(s)

Lecture, Online Education Lecture