

CS G242: COMPUTER ARCHITECTURE AND ORGANIZATION

Item	Value
Curriculum Committee Approval Date	05/02/2023
Top Code	070600 - Computer Science (Transfer)
Units	3 Total Units
Hours	54 Total Hours (Lecture Hours 54)
Total Outside of Class Hours	0
Course Credit Status	Credit: Degree Applicable (D)
Material Fee	No
Basic Skills	Not Basic Skills (N)
Repeatable	No
Grading Policy	Standard Letter (S)

Course Description

This course will cover 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. ADVISORY: CS G153 or CS G175. Transfer Credit: CSU; UC. C-ID: COMP 142. C-ID: COMP 142.

Course Level Student Learning Outcome(s)

1. Course Outcomes
2. Identify the fundamental components, both hardware and software, in the architectural and organizational design of a computer system.
3. Identify errors arising from binary representation.
4. Create assembly language segments with the correct data structure.

Course Objectives

- 1. Compare and contrast the impact on system performance of varying computer system architectures.
- 2. Describe the manner in which the architectural and organization components of computer system work, individually and collectively.
- 3. Solve problems involving operations of computer arithmetic and identify errors arising from binary representation.
- 4. Write the software code, mathematical formulas/expressions, and algorithms in the Assembly language.
- 5. Interact with input, output devices via the interrupts.

Lecture Content

Bits, bytes, and words Electronic hardware for bits and groups of bits Logic gates Logic circuits Mathematical equivalence of binary and decimal Memory storage Data formats ASCII (American Standard for Communication and Information Interchange) character code Numeric data representation and number bases Binary, octal, decimal, and hexadecimal representations Computer arithmetic Arithmetic errors

Overflow and underflow Basic data types Fixed and floating-point systems Fractions in binary Fixed point integers Floating-point format IEEE 754 floating-point standard Floating-point hardware Floating-point numerical accuracy Signed and twos-complement representations Data type ranges Signed data type arithmetic Representation of nonnumeric data (character codes, graphical data) ASCII character code Other codes Binary Coded Decimal (BCD) code Gray code Pixel memory representation Representation of records and arrays Memory addresses Data types and byte allocation Indexed memory locations Basic organization of the von Neumann machine Computer subsystems Subsystems interactions Control unit; instruction fetch, decode, and execution Central Processing Unit (CPU) CPU registers Interaction with memory and I/O (Input/Output) Instruction sets and types (data manipulation, control, I/O) 16-bit, 32-bit, 64-bit instruction sets CPU modes Assembly/machine language programming Assembler listings Linkers The assignment operator Addition and subtraction operators Program flow constructs Instruction formats General format of instructions Register extend prefix (REX) byte Scale/Index/Base (SIB) byte Mode/Register/Memory (ModeRM) byte Addressing modes Register direct Immediate data Base register plus offset Register instruction pointer (rip)-relative Indexed Subroutine call and return mechanisms Passing arguments Call stack Local variables on the call stack Return value and the frame pointer I/O and interrupts Hardware interrupts Software interrupts Exceptions Memory timing I/O device timing Bus timing I/O interfacing I/O ports Interrupt-driven I/O

Lab Content

No lab required

Method(s) of Instruction

- Lecture (02)
- DE Live Online Lecture (02S)
- DE Online Lecture (02X)

Reading Assignments

Textbook and instructor provided handouts.

Writing Assignments

Diagrams and reports on procedures for computer components assemble and utilize.

Out-of-class Assignments

Students will create Assembly solutions for problems assigned in class.

Demonstration of Critical Thinking

Students will analyze problem requirements and select hardware and/or data structures for efficient solution implementation. Testing and debugging will require students to perform data tracing and problem isolation during program execution.

Required Writing, Problem Solving, Skills Demonstration

Students will be required to complete Assembly projects presented to them in the form of business automation problems requiring solution implementation. Students will be required to write documentation for their hardware projects.

Eligible Disciplines

Computer science: Masters degree in computer science or computer engineering OR bachelors degree in either of the above AND masters degree in mathematics, cybernetics, business administration, accounting or engineering OR bachelors degree in engineering AND masters degree in cybernetics, engineering mathematics, or business administration OR

bachelors degree in mathematics AND masters degree in cybernetics, engineering mathematics, or business administration OR bachelors degree in any of the above AND a masters degree in information science, computer information systems, or information systems OR the equivalent. Note: Courses in the use of computer programs for application to a particular discipline may be classified, for the minimum qualification purposes, under the discipline of the application. Masters degree required.

Textbooks Resources

1. Required Robert G. Plantz, Ph.D. Introduction to Computer Organization: An Under the Hood Look at Hardware and x86-64 Assembly, latest ed. No Starch Press, 2022 2. Required Stallings, William. Computer Organization and Architecture, 11th (latest) ed. Pearson, 2019 Rationale: Textbook has comprehensive coverage for hardware topics and less for Assembly programming.