CSCE 212 – Introduction to Computer Architecture

Pooyan Jamshidi

Spring 2022

Website: **https://pooyanjamshidi.github.io/csce212/** E-mail: pjamshid@cse.sc.edu
Office Hours: by appointment Class Hours: Async & Sync
Office: Storey Innovation Center #2207 Class Room: 300 Main St. Room: B213

Bulletin description

Computer architecture, components and organization; Microarchitecture; memory addressing; instruction sets; assembly-language programming.

Prerequisites

CSCE 211 and either 145 or 206

Course Description

The class provides a first introduction to computer architecture. It covers technical foundations of how a computing platform is designed from the bottom up. It introduces various execution paradigms and principles in computer architecture. The focus is on fundamental techniques employed in the design of modern microprocessors and their hardware/software interface.

Learning Outcomes

1. Learn how a modern computer works underneath, from the bottom up.
2. Evaluate tradeoffs of different designs and ideas.
3. Demonstrate the ability to program a microprocessor in assembly language.
4. Describe how conventional machine instructions operate in conjunction with the components of a computer.

Textbooks (optional but highly recommended)

Harris, David, and Sarah Harris. Digital design and computer architecture (2nd edition). Morgan Kaufmann, 2012.

Course Policy

Communication

We envision several routes of communication for this course:

- **Student-to-Instructor & Student-to-Student:** The only mode of communication between students, instructor, and TA will be all electronically through the following tools/services:
  - **Piazza**: It is intended for general questions about the course, clarifications about assignments, questions about research, discussions about the material, and so on. All students *must* enroll in Piazza (the link can be found on the course website).
  - **Gradescope**: We will use GradeScope for all assignments. All students are required to make an account at Gradescope and enroll using this code that will be provided on the course website.
  - **Email**: I would strongly encourage you to ask your question via Piazza so others can also benefit from the questions and learn from the answers. However, if you need to contact the instructor for something that cannot be shared with others (personal issues or something confidential), you can email me or the TA with the provided address. Before sending an email, please think whether it is really necessary or can be shared on Piazza (it is great and you must use it regularly throughout the course).
  - **Office Hours**: The instructor and TA will use Zoom or meet in person for one-2-one meetings or the project team if needed. These are all by appointment.
- **Student-to-Content:** All course materials (e.g., lectures, notes, project descriptions, assignments, syllabus, policies) can be accessed via this link: https://pooyanjamshidi.github.io/csce212/. Note that I will update these materials throughout the semester, so to find out about the latest versions please regularly visit this link. I will announce any update of content via Piazza too.

Academic Integrity

I would encourage you to discuss or brainstorm with other students or lecturers/TA about the assignments and projects, but submissions should acknowledge all collaborators and sources consulted. We will actively check for code and other kinds of plagiarism (both from current classmates and other available online sources). We trust you all to submit your own work, and you are expected to practice the highest possible standards of academic integrity. Any deviation from this expectation will result in a minimum academic penalty of your failing the assignment and will result in additional disciplinary measures including referring you to the Office of Academic Integrity. Violations of the University's Honor Code include, but are not limited to, plagiarism, cheating, falsification, complicity, and any other form of academic misrepresentation. For more information, see https://www.sa.sc.edu/academicintegrity/.

Disabilities Policy

Reasonable accommodations are available for students with a documented disability. If you have a disability and may need accommodations to fully participate in this class, contact the Student

Disability Resource Center: 803-777-6142, TDD 803-777-6744, email sadrc@mailbox.sc.edu, or stop by LeConte College Room 112A. All accommodations must be approved through the Student Disability Resource Center. See https://www.sa.sc.edu/sds/.

Late Work

All assignments and projects have due dates. No late work will be accepted. Submitting all assignments and the project is a necessary condition for passing this class.

Syllabus Change Policy

This syllabus is a guide and every attempt is made to provide an accurate overview of the course. However, circumstances and events may make it necessary for the instructor to modify the syllabus during the semester and may depend, in part, on the progress, needs, and experiences of the students. Changes to the syllabus will be made with advance notice.

Policies and Procedures

This section contains some general rules that will be enforced during this course. Please review these guidelines carefully. The course is governed by the policies and procedures of the university (http://www.sc.edu/policies/ppm/staf625.pdf). Violations of this code will be reported.

Grading Policy

- **Digital Systems Design (Prerequisite Test)**: 0%
- **Assisgnments (6)**: 50%; Based on course materials, including coding, analytical questions, etc. The lowest score will be dropped.
- **Midterm (1)**: 25%; Covers approximately half of the course.
- **Final (1)**: 25%; Cumulative

Grades are on the following **fixed scale**:

A	[90 – 100]
B+	[85 – 90)
B	[80 – 85)
C+	[75 – 80)
C	[70 – 75)
D	[60 – 70)
F	[0 – 60)

Course Schedule (Subject to change)

- **Week 1:** Introduction and Basics
- **Week 2:** Von Neumann Model, ISA, LC-3 and MIPS
- **Week 3** Programming I (MIPS Assembly)
- **Week 4** Programming II (Hardware Description Languages and Verilog)
- **Week 5:** Microarchitecture I
- **Week 6:** Microarchitecture II
- **Week 7:** Pipelining I
- **Week 8:** Pipelining II
- **Week 9:** Advanced Microarchitecture (Out-of-Order Execution)
- **Week 10:** Advanced Microarchitecture (Dataflow and Superscalar Execution)
- **Week 11:** Advanced Microarchitecture (Branch Prediction)
- **Week 12:** Advanced Microarchitecture (Systolic Arrays)
- **Week 13:** Advanced Microarchitecture (Prefetching)
- **Week 14:** Memory Hierarchy I
- **Week 15:** Memory Hierarchy II