EECS 470: Computer Architecture

Winter 2025

Course Overview

Computer architecture is the science and art of selecting and interconnecting hardware components to create a computer that meets functional, performance and cost goals. This course qualitatively and quantitatively examines computer design trade-offs. We will learn, for example, how uniprocessors execute many instructions concurrently and why state-of-the-art memory systems are nearly as complex as processors.

EECS 470 is an advanced undergraduate/introductory graduate-level course in computer architecture. This course is intended to do two things: to give you a solid, detailed understanding of how computers are designed and implemented, including the central processor and memory; and to make you aware of the numerous tradeoffs in deexsign and implementation, their interaction, their realization in both historical and state-of-the-art systems, and trends that will affect them in future systems. We will cover pipelining (including basic pipelining, multiple-instruction-per-cycle machines, out-of-order instruction execution, and vector processing), memory systems (including caches and virtual memory), operating system issues, and basic multiprocessor systems.

A central part of EECS 470 is the detailed design of major portions of a substantial processor using the Verilog hardware design language (HDL). Portions of this work will be done individually as homeworks; the bulk of the work will be done in groups of five to six as a term project. You will learn to use modern commercial CAD tools to develop your design. This project represents a significant investment of time on your part, and is a significant portion of your grade in this class. However, in computer architecture it is particularly true that "the devil is in the details," and you will gain important experience and knowledge by coming face to face with that devil.

Upon satisfactory completion of this course, you will be able to describe and model the detailed mechanics of modern microarchitectures. You will be able to implement arbitrary hardware specifications using industry standard tools and have several concrete strategies for verifying them. You will be well equipped to begin reading publications in top-tier computer architecture conferences and make substantive critiques.

Course Components

Lectures are offered via in-person and recorded formats. Attendance is not required, but strongly encouraged!

Labs are held in-person and involve graded assignments. Attendance is required.

Projects must be completed individually and submitted to the autograder.

Exams will be administered in-person

Office hours will be offered in-person and virtually.

Prerequisites

EECS 470 assumes that you are familiar with the following material:

- Basic digital logic design (EECS 270 or equivalent)
- Basic machine organization (EECS 370 or equivalent)
- Assembly language programming: opcodes, operands, etc.
- High-level languages and data structures
- Verilog hardware description (covered in discussion, but prior experience is helpful)

Quick Links

Administrative Requests	Class Projects	Exams
Lecture Format	Turning in Projects	Textbook
Discussion Format	Doing Your Own Project	Right to Revise
Grading Policy	Labs	

Administrative Requests

Please let us know as soon as possible if you experience interruptions to your studies and will need accommodations for homeworks, projects, or exams. **Students do not need to submit requests for lecture absences, as attendance is not required or graded.**

For assignment extension requests (given for medical / personal emergencies, or approved conflicts communicated in advance, **must be made at least 24 hours before deadline** unless an emergency prevents prompt communication):

Extension Requests: *https://forms.gle/Vs8WE85VpCzVdaxv8*

For Exam Conflicts (Deadline February 4th):

Exam Conflict Requests: *https://forms.gle/tCQW25rF9u8MnfUH6*

Email eecs470staff@umich.edu for ***urgent issues*** and please include "EECS 470" in the subject line

Lecture Format

Lectures will be held live with recordings posted on the course website. Attendance is not required.

Lab Format

Labs will be in-person. Slides and assignments will be released beforehand. Labs will include assignments that must be checked off (in-person or remotely) by the posted deadline.

Grading Policy

Final grades will be based on the total points earned on homework, projects and exams. The grade distribution is as follows:

Category	#	%	Notes
Homework Problem Set	5	10%	Drop Lowest
Individual Projects	3	10%	2% + 3% + 5%
Labs	7	5%	5 with assignments
Exams	2	40%	20% + 20%
Group Project	1	35%	

The average grade in the course is expected to be a B+. Final grades will be based on a straight scale, but will be curved up if assignments are more difficult than expected and the average falls below this point.

Homework Problem Set

All assignments will be available on the course home page. Your solutions to the assigned problems will be submitted via GradeScope. Homework submitted up to 24 hours late will receive a 10% penalty.

Class Projects

Three individual projects will be assigned during the term, each of which will require a substantial time commitment on your part. Each project will involve writing a design in Verilog to meet provided specifications. Specifications will be released for each project on the course website once available. Labs will provide more details on implementation. Students may attend professor or staff office hours for extra help on the projects. There will be one group project, project 4, which will be done in teams. Teams will ultimately be formed by course staff, however, students can communicate who they would like to work with.

Turning in Projects

Projects are due at **11:59pm Eastern time exactly on the due date**.

You will be submitting your projects electronically by following directions in the spec. Your projects will be graded automatically using an autograder program. You are allowed to submit your programs as frequently as you wish. For each project, your final score will be derived from your **most recent submission** to the autograder. This may or may not be the highest scoring submission.

Academic Integrity

We encourage collaboration in EECS 470, especially on concepts, tool, specifications, and strategies.

For projects, you are welcome to use any code provided in lecture or labs. While you are allowed to show another student your project code to discuss concepts (see below), you may not copy code.

See below for examples of approved collaboration:

Encouraged Collaboration	Unacceptable collaboration
Sharing high-level design strategies, e.g., helper function organization or data structure choices	Walking through an important piece of code step-by-step, sharing pseudocode, sharing comments

Helping others understand the spec or project nuances	Providing your code as a reference
Helping someone debug	Debugging someone's code for them
Explaining a runtime error to someone	Fixing a runtime error for someone
Discussing test strategies	Sharing test code to verify someone else's design, even if test cases are not submitted
Brainstorming edge cases for testing	Discussing specifics about what tests exposed instructor bugs on the autograder
Using starter code provided with a project or based on examples shown in lecture	Copying code in whole or in part, even if the code is modified Writing original code for someone else, or paying someone to write your project
Looking at someone else's code to understand concepts or help someone debug	Sharing your code in a way that could be copied, e.g., sending code over email or taking a picture of code

You are still responsible for following these rules even after finishing the course.

If you are unsure about what constitutes an honor code violation, please contact the course staff with questions.

Labs

There will be seven lab sessions (five with assignments) during the semester. Each lab consists of an assignment. Assignments must be checked off by staff in lab or in office hours by the specified due date, however labs are intended to be finished within the 2 hour lab period.

Exams

There will be **two** in-person exams this semester and will be delivered outside normal lecture time.

You are expected to take the exams at the scheduled times. If you do not take an exam without verifying a documented medical or personal emergency causing you to miss an exam, you will receive a zero for that exam. If you anticipate conflicts with the exam time, declare your conflicts using the link at the top of the syllabus. The exam dates are announced at the beginning of the semester so you can avoid scheduling job interviews or other commitments on exam days.

We may not be able to accommodate requests submitted after the deadline. For last-minute emergency accommodations (*e.g.,* documented illness), please contact the staff at eecs470staff@umich.edu.

Textbook

The textbook for the class is "Computer Architecture: A Quantitative Approach" by Hennessy and Patterson. The text is available online for free if you are within the university subnet.

Right to Revise

The course staff reserve the right to make changes to the syllabus at any time, as they see fit. When a revision occurs, it will be announced through Piazza, and it is your responsibility to be informed of such.