EC413: Computer Organization – Fall, 2024

Instructor: Prof. Martin Herbordt – Office Hours: PHO333, TW 3:00-5:00 and by appointment
Office Phone: x3-9850 – herbordt@bu.edu

Course web page: http://learn.bu.edu

GTFs: Xiteng Yao, Shining Yang – {xtyao,shiningy}@bu.edu
UTFs: Eric Chen, Jason Li, Cristian Palencia, Ben Taylor, Kyla Wilson – {chene,jli3469,cris0211,betaylor,kylaw}@bu.edu

Discussion Sections: Monday 1:25-2:15, 4:40-5:30, 6:30-7:20 all in PHO 305

TF Lab Hours in PHO305/307: M: 5:30-6:30 / T: 5:30-8 / W: 6-8 / Th: 6-9 / Fr: 3-6

Mission Statements: *From gates to programs* and *How computers really work (how programs really run)*

Course Description: Introduction to the fundamentals and design of computer systems. Topics include computer instruction sets, assembly language programming, algorithmic and logic design of arithmetic operations, design of sequential logic with registers and buses, CPU design (data path, control, pipelining), performance evaluation, memory devices, memory systems (caching and virtual memory), and I/O. In parallel there is a lab where the focus is on in-depth understanding of most of the course topics. The lab is also a practicum in modern digital system design using HDLs (Verilog) and industrial EDA (Xilinx Tools).

Course Style: EC413 has both theoretical and practical aspects; you will gain both declarative and procedural knowledge.

Prerequisites: EC311, Introduction to Logic Design
Familiarity with Xilinx CAD tools
High-Level Language Programming, preferably in EC327. Literacy with **C** is helpful.

Texts: Patterson & Hennessy, *Computer Organization & Design: The Hardware/Software Interface*, 6th Edition, Morgan Kaufmann, 2020 – This is the MIPS edition.

Course Mechanics

- **Exams:** There are two midterms and a final. Exams are closed book/notes with a single standard-sized sheet of notes allowed (hand-written, front and back).
- **Quizzes:** Most semesters there are ~3-6 short quizzes. Their purpose is to help you keep up, especially at the start of a new topic. The quiz grade is computed from the N-1 best quizzes (including absences).
- **Attendance:** Attendance is essential – much of what we cover in this course will be found nowhere else.
- **Readings:** … are also essential. There is no way we can (or would want to) go through all you need to know about any topic during class. Good news - the textbook is a standard so you can expect your future colleagues to have learned computer organization in similar way.
- **Homework:** There are ~7-8 homework assignments. Unless stated otherwise, homework is due by *the beginning of class* on the date specified and turned in using the course web site. Homework will not be accepted after that. Homework must be readable, so typed is preferred. But scanning or photo (with conversion to pdf) is OK.
  Academic honesty wrt HWs: You are encouraged to work together to learn the material and to discuss approaches to solving homework problems. However, *you must come up with and write up the solutions on your own. See academic honesty policy.*

Labs and Discussion Sections

- **Overall:** There are ~9 labs. Much more about these later! Some of the labs have a substantial workload (esp. 5-7, but also 3,4,8), requiring perhaps 8 hours or more for many students. It is therefore essential that you get an early start and understand thoroughly the underlying material before you start to code. For most labs there is a *prelab* during discussion section; this is generally a to-be-turned-in assignment whose purpose is to help you get started.

- **Lab Practicalities:** The labs are open-ended in the sense that there is not a specified time during which the labs must be done. Rather, labs are assigned about a week before they are due and it is your responsibility to get them done on time, generally late on Friday afternoon. The TFs are in the lab about 15 hours during weeks when labs are due. Please note that the amount of time that it takes any particular student/group to complete any particular lab takes can vary substantially (from a few hours to much more than that). The greatest determinants of duration are how well prepared you are before you start coding and how well you understand the tools.
- **Lab grading mechanism:** A large part of each lab is the *demo*. Because there are many more students that TAs, you are urged to get the labs done early. This semester the imbalance is especially acute so we may need to have a sign up mechanism to prevent congestion near the due time.
  Academic honesty wrt labs: Since labs are done by group, the rules for collaboration are more strict than for homework: except for technical issues *labs are all to be done with your group alone.*
- **More lab grading:** From the programming lab on, there is 5% bonus for finishing by Thursday, 8% bonus for finishing Wednesday or earlier. There is a 10% penalty for being late one business day (usually the following Monday) and a larger penalty for being later than that.
- **Discussion Sections**: The discussion sections are run by the TFs and serve two purposes: a pre-lab or a pre-exam Q&A session. Attendance is not graded, but is *very strongly* recommended.

Grading

Grades are based on a weighted average of HW, quiz, exam, and lab scores. *Each category is curved independently*. Since we update all assignments every semester, we never know in advance exactly how the scores in each category will relate to a letter grade, but here is the general idea:

HWs and Labs. The expectation is that all students will make a significant effort on all HW assignments and will have working demos of all labs. There is also some opportunity for extra credit on the labs. Therefore the cut-offs between A/B, B/C, etc. are generally *higher* than the traditional 90, 80, etc. For example, for an A, a student should get at least 90s on HWs and 95s on the labs.

Exams and Quizzes. On the other hand, exams and quizzes are often challenging and grades are hard to predict, so the reverse is true here. Last year, the A-/B+ cut-off for the mid-terms was an 82 and for the final a 76.

Weighting the individual grades in final grade (for reference only, these may change).

- HWs, Quizzes = ~1pt each (about 10%-12% total)
- Labs = 3-4pts each (about 30%-32% total)
- Exams = 18pts, 18pts, 22pts

Please note: *You are not competing with each other on grades*. There have been semesters when half the class has gotten A- or better. FWIW, the median is often around a B, but this is purely coincidental. No student is "average"!

Administration

- **Office Hours:** Other than my office hours, the best time to catch me (otherwise) is between class halves or after class; the worst time is right before class when I am setting up. On occasion I may be traveling and so unable to keep office hours. I will announce this well in advance. *If I am away, the TFs will be available.*
- **Email:** You should periodically check your email since that is how some notifications (assignments and coreeccttions) are distributed. Questions via email are always good. If the question/answer has general interest, I will send it to the class (leaving the questioner anonymous); if the solution is very involved, we may need to go over it in person.
- **Course Web Site:** I use the course web site to post class notes, annotations, lab and homework assignments, homework solution sketches, additional readings, and other course information.
- **Incompletes:** Incompletes are granted only in accordance with university policy, which (broadly) requires a major non-academic difficulty near the end of the semester.
- **If you have an academic issue of any kind:** These happen – please let me know. And note that they are much easier to resolve or mitigate if you talk to me early – don't wait.
- **Academic Conduct:** https://www.bu.edu/academics/policies/academic-conduct-code - Please read the university academic honesty policy. If something is not clear, then ask. In particular, plagiarism is really serious!
- **Course Notes:** Because of the nature of the material, Computer Organization is always taught with slides, although we mix things up by doing examples, often on the slides themselves. A version of the slides will always available well before class, although the final version might not be available until just before class.

- **Distractions:** What was that? Except for annotating notes, please keep all electronic devices off during class. This will help me, you, and the students sitting around you.
- **Punctuality:** I start class promptly so please be here on time (see "distractions").
- **Instructer Errrosr:** Please don't be shy! If you see me make a mistake, please let me know right away. If you are not sure, that's even better – it might give me a chance to clarify something.
- **Your success** is something I really care about! All job markets are incredibly competitive, but students who do well in this class have a very high probability of working at the best companies (and becoming researchers, doctors, executives, etc.).

For all courses at BU – university policies

- **Inclusion:** I consider this classroom to be a place where you will be treated with respect, and I welcome individuals of all ages, backgrounds, beliefs, ethnicities, genders, gender identities, gender expressions, national origins, religious affiliations, sexual orientations, ability – and other visible and nonvisible differences. All members of this class are expected to contribute to a respectful, welcoming, and inclusive environment for every other member of the class.
- **Accommodations for Students with Documented Disabilities:** If you are a student with a disability or believe you might have a disability that requires accommodations, requests for accommodations must be made in a timely fashion to Disability & Access Services, 25 Buick St, Suite 300, Boston, MA 02215; 617-353-3658 (Voice/TTY). Students seeking academic accommodations must submit appropriate medical documentation and comply with the established policies and procedures http://www.bu.edu/disability/accommodations/

Keys to success

- **Attendance.** You must attend class. Much of the material (and perspective) will be found nowhere else.
- **Do the readings.** Other material is found only in the readings.
- **Read actively**. Work out the examples as you read. If you are not positive that you understand something completely, try inventing and solving your own problems.
- **Take notes.** In particular, annotate the lecture notes during class. This practice has been found to significantly improve retention of new material, whether on paper or a tablet
- **Reread the notes.** Perhaps surprisingly (or not), retention is better from reading than from listening, and even better for writing, and still better when you explain it to someone else.
- **Participate.** Ask questions; talk with your fellow students. Be active. This really helps me too!
- **Keep up.** Before each class, (at least) study the notes from the previous class. This course comprises a number of disparate topics, each with its own terminology and axioms.
- **Allocate enough time!** Much of the material is time-consuming to master. There is often a big difference between thinking you understanding a subject and *really* understanding it. In computer organization and architecture topics interact in subtle ways; mastery of any one requires a good understanding of all of them.
- **How do you know that you know the material?** A good metric is whether you would feel comfortable standing in front of a class explaining it. Or during a job interview!

Practical Goals (partial list)

Starting points: logic design, basic knowledge of HDLs and EDA tools, and high-level language programming.
Targets:
(1) Create a working CPU with 5-stage pipeline from basic logic.
(2) Program that CPU in assembly language.
(3) Create digital systems using standard design practices including hierarchy, parameterization, multiple design styles, and testing.
(4) Evaluate computer system design options at multiple levels.
(5) Understand the interaction of software and hardware and how this affects software performance – especially for AI applications.