



Carnegie Mellon University Course Syllabus

18-740: Computer Architecture

Fall Semester, 2013

<http://www.ece.cmu.edu/~ece740>

Course Personnel:

Instructor (ECE side):

- Onur Mutlu
onur@cmu.edu
<http://users.ece.cmu.edu/~omutlu/>
Office Hours: Wed. 2:30PM-3:30PM, or by appointment

<i>Teaching Assistants:</i>	Email	Office Hours and Location
• Tyler Huberty	thuberty@andrew.cmu.edu	TBD
• Brian Osbun	bosbun@andrew.cmu.edu	TBD
• Hongyi Xin	hxin@andrew.cmu.edu	TBD
• TBD	TBD	TBD

Course Management Assistant:

- Jennifer Loughran, jackson1@andrew.cmu.edu, Hamerschlag Hall D200

Course Description: Computer architecture is the science and art of designing, selecting and interconnecting hardware components and co-designing the hardware/software interface to create a computer that meets functional, performance, energy consumption, cost, and other specific goals. This course qualitatively and quantitatively examines fundamental computer design trade-offs, with the goal of developing an understanding that will enable students to perform cutting-edge research in computer architecture. We will learn, for example, how uniprocessors execute many instructions concurrently, how state-of-the-art memory systems deliver data into the processor and why they are so complex, and how/why multiple processors are interconnected to execute portions of a program or multiple programs in parallel, as done in modern multi-core processors. Examining trade-offs requires that you already know how to correctly design a computer, as is taught in the important prerequisite 18-447. This course will involve an at least two-month long research/implementation project in which students work in groups of 2-3.

Number of Units: 12

Pre-requisites: 18-447 or equivalent

Undergraduate Course Designation: Breadth, Depth, Coverage

Undergraduate Course Area: Computer Hardware Engineering

Class Schedule:

- Lecture: DNM (Does Not Meet – Offered Online – see webpage for instructions)
- Recitations:
 - M, 7:30pm to 9:20pm, Location TBD
 - W, 12:30pm to 2:20pm, Location TBD
 - F, 10:30am to 12:20pm, Location TBD
 - F, 3:00pm to 4:20pm, GHC 4303

What This Course is About:

This course has two major goals:

1. to familiarize computer architecture students and those interested in computer system design with both fundamental design tradeoffs and recent research issues/trends in processor, memory, and platform architectures in today's and future systems. There will be a strong emphasis in fundamentals and design tradeoffs.
2. to provide the necessary background and experience to advance the state-of-the-art in computer architecture by performing cutting-edge research. There will be a strong emphasis on critically evaluating research papers (through literature review assignments) and developing new mechanisms that advance the state of the art or evaluating aspects of modern architectures (through the course research project).

This course teaches students how to make design decisions in a computer platform (both in hardware and the hardware/software interface, and sometimes in software) and also how to make them better than past approaches. It is not an introduction to computer system design. The scope of the course material is general and fundamental: the techniques and the design decision process taught is fundamentally applicable to different type of computing platforms, including server, desktop, mobile, embedded, and sensor systems.

Readings and Textbooks:

Arguably, there is no perfect textbook for computer architecture. I encourage you to do your own research, consult multiple sources, question assumptions and statements, and talk with me and TAs whenever you have questions.

Lecture notes and videos will serve as the main source of information and they will provide the required references to textbooks or other reading material (such as research articles).

A good source of information on all covered topics is the *research articles* that introduced or built upon the covered topic. These articles are usually published in top conferences (such as ISCA, MICRO, ASPLOS, HPCA) or journals (such as IEEE or ACM Transactions). I strongly encourage you to dig out the original source of the covered topics as well as the research that builds upon it. This will help you become a successful and well-read researcher in computer architecture/systems. When in doubt, ask questions.

The following textbooks could be useful as supplements to lectures:

1. *Parallel Computer Organization and Design* by Dubois, Annavaram, and Stenstrom, Cambridge University Press
2. *Computer Architecture: A Quantitative Approach*, Fifth Edition by Hennessy and Patterson, Morgan Kaufmann/Elsevier
3. *Computer Architecture and Implementation* by Harvey Cragon, Cambridge University Press
4. *Computer Organization* by Hamacher, Vranesic, Zaky, McGraw-Hill
5. *Structured Computer Organization* by Andrew Tanenbaum, Prentice Hall
6. *High Performance Computer Architecture* by Harold Stone, Prentice Hall (dated but good reference)
7. *Readings in Computer Architecture* by Hill, Jouppi, Sohi, Morgan Kaufmann (dated but good reference with reprints of papers)
8. *Parallel Computer Architecture: A Hardware/Software Approach*, by Gupta, Culler, Singh, Morgan Kaufmann

The following websites contain links to original articles as well as articles that refer to them:

1. Google and Google Scholar
2. ACM Digital Library
3. IEEE Explore
4. Microprocessor Report (The CMU E & S Library has a subscription)

Other reading material will be distributed in class and/or will be available on the website electronically.

Grading Algorithm:

20%	Reviews, Online Assignments, Participation, Presentations
40%	Research/Implementation Project
20%	Midterm 1 (can potentially be a Literature Review Assignment)
20%	Midterm 2
5%	Teaching Team's Evaluation of Your Performance

Research Projects & Late Policy:

This course is a hands-on research oriented course. You (in groups of two) are expected to propose, conduct, and experimentally evaluate a 2-3-month long research project whose goal is to advance the state-of-the-art and/or current understanding in computer architecture or a related subject. The topic of the project is flexible, but it must be approved by me. You will have to find a project partner and conduct research in groups of two. This is your chance to explore in depth a computer architecture topic that interests you and perhaps even publish your innovation in a top computer architecture conference. I strongly encourage you to start thinking about your project topic as early as possible and interacting with me and the teaching assistants to crystallize it over time.

A total of five late days will be allowed for the assignments and project proposal/reports. Once you use up all your late days, your submissions will be graded based on the instructor's discretion. No late submission will be allowed for the project final report.

Tentative Course Calendar and Topics:

Below is an ambitious list of topics we intend to cover. The order of topics might change and we may not be able to cover all topics. Please see the course website to view the most current schedule and to download lecture notes and handouts. The below topics are also very much fit for research projects.

1. Multi-core systems and multiprocessors
 - Parallel processing basics
 - Multi-core architectures: motivations, symmetric and asymmetric designs
 - Asymmetric multi-core, accelerators
 - Coherence and consistency (see 447 Spring 2013 video online)
 - Programming models
 - Speculation and transactional memory
2. Memory systems
 - Virtual memory (see 447 Spring 2013 videos online)
 - Caches and efficient caching
 - Interconnects and efficient interconnects

Main memory basics and scaling
Emerging technologies and hybrid memories
Memory interference and mitigation techniques
Non-volatile memory technologies and their scaling issues (if time permits)
Prefetching and latency tolerance methods

3. SIMD, GPUs, systolic architectures

SIMD processing and GPU basics
Systolic architectures

4. Instruction-level parallelism

Out-of-order execution
Runahead execution
Superscalar processing
VLIW and static scheduling
Advanced branch handling

5. Multithreading and Data flow

Models of multithreading
Data flow processing

6. Reliability and security issues in architecture

Education Objectives (Relationship of Course to Program Outcomes):

(a) an ability to apply knowledge of mathematics, science, and engineering: The students will apply classroom knowledge in the semester-long research project and reading assignments.

(b) an ability to design and conduct experiments, as well as to analyze and interpret data. The students will perform three tasks in the semester-long research project.

(c) an ability to design a system, component, or process to meet desired needs within realistic constraints such as economic, environmental, social, political, ethical, health and safety, manufacturability, and sustainability: The students will create designs to meet the design specifications of the projects they propose.

(d) an ability to function on multi-disciplinary teams: The students will work in teams of 2 or 3 in the research projects.

(e) an ability to identify, formulate, and solve engineering problems: The students will develop original solutions to open-ended problems in the research projects, exams, and assignments.

(f) an understanding of professional and ethical responsibility. The students will work in teams, which require this responsibility.

(g) an ability to communicate effectively: The students will provide written reports, presentations, posters for the research project. Interactive discussions planned into the recitations will encourage the students to communicate questions and answers verbally.

(h) a knowledge of contemporary issues: The students will be introduced to contemporary developments and topics in the computer architecture field.

(i) an ability to use the techniques, skills, and modern engineering tools necessary for engineering practice: The students will use industry-standard design tools and methodologies in the research projects as well as assignments.

(j) a knowledge of contemporary issues: the students will be exposed to state-of-the-art problems in computer architecture throughout the lectures, readings, projects

(k) an ability to use the techniques, skills, and modern engineering tools necessary for engineering practice: the students will learn and use modern tools in research projects.

Academic Integrity Policy (<http://www.ece.cmu.edu/student/integrity.html>)

The Department of Electrical and Computer Engineering adheres to the academic integrity policies set forth by Carnegie Mellon University and by the College of Engineering. ECE students should review fully and carefully Carnegie Mellon University's policies regarding *Cheating and Plagiarism*; *Undergraduate Academic Discipline*; and *Graduate Academic Discipline*. ECE graduate students should further review the *Penalties for Graduate Student Academic Integrity Violations* in CIT outlined in the CIT Policy on *Graduate Student Academic Integrity Violations*. In addition to the above university and college-level policies, it is ECE's policy that an ECE graduate student may not drop a course in which a disciplinary action is assessed or pending without the course instructor's explicit approval. Further, an ECE course instructor may set his/her own course-specific academic integrity policies that do not conflict with university and college-level policies; course-specific policies should be made available to the students in writing in the first week of class.

In 18-447, discussions about projects and review papers are encouraged. Each members of a project group has to contribute equally to all project assignments, including reviews, posters, papers. Exams and online assignments must be completed individually unless otherwise specified.

Carnegie Mellon University's Policy on Cheating and Plagiarism (<http://www.cmu.edu/policies/documents/Cheating.html>) states the following,

Students at Carnegie Mellon are engaged in preparation for professional activity of the highest standards. Each profession constrains its members with both ethical responsibilities and disciplinary limits. To assure the validity of the learning experience a university establishes clear standards for student work.

In any presentation, creative, artistic, or research, it is the ethical responsibility of each student to identify the conceptual sources of the work submitted. Failure to do so is dishonest and is the basis for a charge of cheating or plagiarism, which is subject to disciplinary action.

Cheating includes but is not necessarily limited to:

1. Plagiarism, explained below.
2. Submission of work that is not the student's own for papers, assignments or exams.
3. Submission or use of falsified data.
4. Theft of or unauthorized access to an exam.
5. Use of an alternate, stand-in or proxy during an examination.
6. Use of unauthorized material including textbooks, notes or computer programs in the preparation of an assignment or during an examination.
7. Supplying or communicating in any way unauthorized information to another student for the preparation of an assignment or during an examination.

8. Collaboration in the preparation of an assignment. Unless specifically permitted or required by the instructor, collaboration will usually be viewed by the university as cheating. Each student, therefore, is responsible for understanding the policies of the department offering any course as they refer to the amount of help and collaboration permitted in preparation of assignments.
9. Submission of the same work for credit in two courses without obtaining the permission of the instructors beforehand.

Plagiarism includes, but is not limited to, failure to indicate the source with quotation marks or footnotes where appropriate if any of the following are reproduced in the work submitted by a student:

1. A phrase, written or musical.
2. A graphic element.
3. A proof.
4. Specific language.
5. An idea derived from the work, published or unpublished, of another person.

This policy applies, in all respects, to 18-740.