



## Carnegie Mellon University Course Syllabus

### 18-740: Computer Architecture

Fall Semester, 2015

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

#### Course Personnel:

##### *Instructor:*

- Onur Mutlu  
[onur@cmu.edu](mailto:onur@cmu.edu)  
<http://users.ece.cmu.edu/~omutlu/>  
Office Hours: Wed. 3-4PM (Pittsburgh time), or by appointment  
Office Location: CIC 4105 (Pittsburgh), B23 A116 (SV)

##### *Teaching Assistants:*

- |                      | Email  | Office Hours and Location |
|----------------------|--|---------------------------|
| • Nandita Vijaykumar | <a href="mailto:nandita@cmu.edu">nandita@cmu.edu</a> | TBD                       |

##### *Course Management Assistant:*

- SV: Stephanie Scott, Building 19 Room 1052, [scottsm@andrew.cmu.edu](mailto:scottsm@andrew.cmu.edu)
- Pittsburgh: Zara Collier, HH 1112, [zcollier@andrew.cmu.edu](mailto:zcollier@andrew.cmu.edu)

---

**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:

- Lectures (Pittsburgh): M & W 7:30pm to 9:20pm , Hamerschlag Hall 1107
- Recitation (Pittsburgh): T 7:30pm to 9:20pm, DH A302
  
- Lectures (SV): M & W 4:30pm to 6:20pm, B23 118
- Recitation (SV): T 4:30pm to 6:20pm, B19 1065

### **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 very useful as supplements to lectures. Some lecture readings will be assigned from them:

1. *Parallel Computer Organization and Design* by Dubois, Annavaram, and Stenstrom, Cambridge University Press
2. *Modern Processor Design: Fundamentals of Superscalar Processors*, by Shen and Lipasti.

Other useful readings could include the following:

1. *Computer Architecture: A Quantitative Approach*, Fifth Edition by Hennessy and Patterson, Morgan Kaufmann/Elsevier
2. *Computer Architecture and Implementation* by Harvey Cragon, Cambridge University Press
3. *Computer Organization* by Hamacher, Vranesic, Zaky, McGraw-Hill
4. *Structured Computer Organization* by Andrew Tanenbaum, Prentice Hall

5. *High Performance Computer Architecture* by Harold Stone, Prentice Hall (dated but good reference)
6. *Readings in Computer Architecture* by Hill, Jouppi, Sohi, Morgan Kaufmann (dated but good reference with reprints of papers)
7. *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:

35%	Critical Reviews, Participation, Presentations
50%	Research/Implementation Project
15%	Exam(s)
5%	Instructor's Evaluation of Your Performance

### Research Projects & Late Policy:

This course is a hands-on research oriented course. Research and critical thinking is the key emphasis. 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:

#### Tentative Course Calendar:

	Date	Day	Class Activity
<b>August/September</b>			
Week 1	8/31	Mon	<b><u>Lecture 1</u>: Introduction to Computer Architecture [JPS, OM]</b>
	9/1	Tues	Recitation 1: Overview of 18-740

	9/2	Wed	<b><u>Lecture 2: Review of Pipelined Processor Design [JPS]</u></b>
	9/7	Mon	<LABOR DAY HOLIDAY, NO CLASS>
Week 2	9/8	Tues	Recitation 2: Promising Research Topics in Computer Architecture
	9/9	Wed	<b><u>Lecture 3: From Pipelining to Superscalar (ILP) [JPS]</u></b>
	9/14	Mon	<b><u>Lecture 4: Instruction Flow Techniques [JPS]</u></b>
Week 3	9/15	Tues	Recitation 3
	9/16	Wed	<b><u>Lecture 5: Branch Prediction Techniques [OM]</u></b>
	9/21	Mon	<b><u>Lecture 6: Register Data Flow Techniques [JPS]</u></b>
Week 4	9/22	Tues	Recitation 4
	9/23	Wed	<b><u>Lecture 7: Dynamic Scheduling and Out-of-Order Execution [JPS]</u></b>
	9/28	Mon	<b><u>Lecture 8: Main Memory Overview [OM]</u></b>
Week 5	9/29	Tues	Recitation 5
	9/30	Wed	<b><u>Lecture 9: Emerging Memory Technologies and Architectures [OM]</u></b>
<b>October</b>			
	10/5	Mon	<b><u>Lecture 10: Memory Data Flow Techniques [JPS]</u></b>
Week 6	10/6	Tues	Recitation 6
	10/7	Wed	<b><u>Lecture 11: VLIW Architecture and Compiler (ILP &amp; LLP) [JPS]</u></b>
	10/12	Mon	<b><u>Special Lecture:</u></b> “ <i>Extreme System Integration via 3D Die Stacking</i> ” (Bryan Black)
Week 7	10/13	Tues	Recitation 7
	10/14	Wed	<b><u>Lecture 12: Multithreading Processors (from ILP to TLP) [OM]</u></b>
	10/19	Mon	Project Reviews
Week 8	10/20	Tues	Recitation 8
	10/21	Wed	Project Reviews
	10/26	Mon	<b><u>Lecture 13: Multi-Core Architecture (ILP &amp; TLP) [OM]</u></b>
Week 9	10/27	Tues	Recitation 9
	10/28	Wed	<b><u>Lecture 14: Multi-Core Memory Architectures &amp; Resource Management [OM]</u></b>

November			
Week 10	11/2	Mon	<b><u>Lecture 15: Memory Synchronization, Consistency &amp; Cache Coherence</u></b> [OM]
	11/3	Tues	Recitation 10
	11/4	Wed	<b><u>Lecture 16: Interconnects</u></b> [OM]
Week 11	11/9	Mon	<b><u>Lecture 17: Heterogeneous Systems</u></b> [OM]
	11/10	Tues	Recitation 11
	11/11	Wed	<b><u>Lecture 18: Prefetching and Latency Tolerance Techniques</u></b> [OM]
Week 12	11/16	Mon	<b><u>Lecture 19: SIMD and GPUs (DLP)</u></b> [OM]
	11/17	Tues	Recitation 12
	11/18	Wed	<b><u>Lecture 20: Performance and Power Iron Laws</u></b> [JPS]
Week 13	11/23	Mon	<b><u>Lecture 21: Power Reduction and Energy Efficiency</u></b> [JPS]
	11/24	Tues	Recitation 13
	11/25	Wed	<b><u>Special Lecture: “From Processor Design To NBA Technology”</u></b> [Quinn Jacobson]
December			
Week 14	11/30	Mon	<b><u>Lecture 22: War of Mobile Platforms and Ecosystems</u></b> [JPS]
	12/1	Tues	<b>NO CLASSES – THANKSGIVING</b>
	12/2	Wed	<b>NO CLASSES – THANKSGIVING</b>
Week 15	12/8	Mon	EXAM
	12/9	Tues	Project reviews
	12/10	Wed	<b>Project reviews</b>

**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 thee 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.*