OAKLAND UNIVERSITY, School of Engineering and Computer Science
CSE 564 Computer Architecture, Summer 2017
https://passlab.github.io/CSE564/

Instructor: Yonghong Yan (www.secs.oakland.edu/~yan), yan@oakland.edu
Office: 534 Engineering Center; **Tel:** (248) 370 4087
Office Hours: Monday 11:00 - 12:30PM, after class or by appointment
Meeting Time: 5:30PM – 8:50 PM Monday and Wednesday; Check http://www.oakland.edu/important-dates for other important date/deadline
Class Room: EC 566
Website: https://passlab.github.io/CSE564/

Course Catalog Description:

The course explores current trends and future directions in processor micro-architecture as well as various hardware and software techniques in high-performance computing. A review of pipeline and memory hierarchies is followed by advanced topics including branch prediction, dynamic scheduling, superscalar techniques, speculative execution, prefetching, high-speed I/O, VLIW, multi-threaded processors, and application-specific processors such as those for embedded and multimedia systems. This course is cross listed with an undergraduate course. Credit cannot be received for both CSE 564 and CSE 464.
4.000 Credit hours
4.000 Lecture hours
Levels: Graduate, Post Bachelor, Graduate Professional Development, Doctoral & Ed Specialist, Undergraduate

Requirements:

Minimum requirements include good reasoning and analytical skills, skills of C programming, e.g. the use of macro, pointer, array, struct, union, function pointer, and library for memory allocation and de-allocation (malloc and free), and knowledge of high-level languages such as Java or Scala. Knowledge or undergraduate courses of computer architecture (memory hierarchy, cache, virtual address) and data structures will be necessary for preforming well for the class. Knowledge or courses of programming languages and compilers, and operating systems will also help. Familiarity with Linux environment will be important for the assignments.

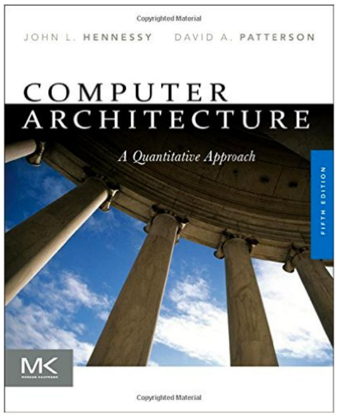
Course Objectives:

Upon completion of course, successful students shall be able to understand all the major concepts used in modern microprocessors and able to design basic computer architectures using high-level hardware description languages

- Understand in-depth how software interacts with hardware which include topics of instruction set and system software (compiler)
- Explain key concepts in computer architectures including processor architecture, memory hierarchy and cache coherence, CPU pipeline and out-of-order execution, instruction, data and thread level parallelism
- Perform quantitative design and analysis of computer architecture for computer programming
- Follow advanced and emerging technology and architectures

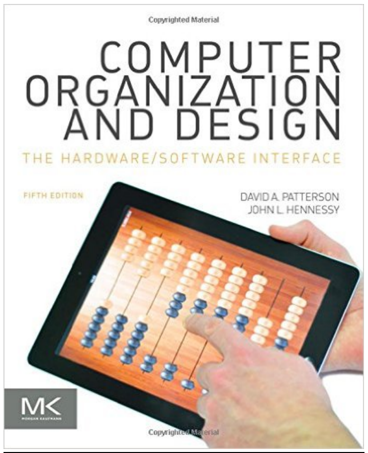
Textbook and materials:

Required Textbook:



Computer Architecture A Quantitative Approach, 5th Edition 2011, by John L. Hennessy and David A. Patterson

Reference Textbook:



Computer Organization and Design, Fifth Edition: The Hardware/Software Interface, 2013, by John L. Hennessy and David A. Patterson

Grade Assessment:

Course grade will be a combination of 4 assignment (50%) and two exams (50%). Additional bonus points may be included in some of the assignments. Exam are close book and close notes.

Assignments

In those assignments, you will apply theory and practice programming skills for the answers and implementations of processor design. They require both correct programming and good description and discussion in the written report, which should present the approach you take and the results you find in your program. All assignments and class activities will be completed **individually**.

Behavioral Contract: (For lectures, assignments and exams)

- This class is offered in classroom with PCs and will include both lecture and lab practices; please read the following carefully, students will be asked to leave the labs if they do not follow this contract:
  - Digital devices such as: cell phones, tablets, ipod, mp3 players… etc need to be placed out of sight and must be set to silent. (I will keep my cell phone avail, for emergency preparedness)
  - No headphones are allowed.
  - Students are expected to use the PCs only for this class's material and applications.
  - Surfing the net for Social sites and being on Social Media are not allowed **during lecture hours**. Anyone who is not following this rule will be asked to leave the laboratory/lecture.

Attendance

- Attendance is NOT required in all of the classes, but highly recommended since we will cover lots of design and implementation activities during the class.

Late Policy: (PLEASE READ CAREFULLY)

Each assignment has a due date and a cutoff date, which is 3 days after due date. You will have a total of 7 "slip days" throughout the semester that you can use at your discretion to turn in programming assignments past the posted due date. Slip days are used in whole day increments. Once your slip days are consumed, late assignments will be penalized at 10% per day. Assignment submission will not be accepted after the cutoff date.

Moodle:

A session specific website is located at https://moodle.oakland.edu/moodle. This website will include all the course materials (notes, schedules, assignments, etc) for our course. Assignment will be available for download from this site, and they should be submitted using Moodle only. Please check this site often for updates.

Academic Conduct: *Expected conduct on assignments and exams*

Although students may discuss an assignment, each student should complete his or her assignment individually. Copying of another's assignment is not permitted. It is assumed that ALL work throughout the term is your own. Discussions during an exam or quiz are not permitted. Cheating during an exam or quiz is not permitted. It is assumed that ALL WORK THROUGHOUT THE TERM IS YOUR OWN! **Discussion of lab assignments are permitted but copying of assignments is not! Handing in a lab assignment or exam that was essentially copied from someone else does constitute as cheating**. All of the tests are closed book unless it was told otherwise. Obtaining help from notes, another individual or from hand held computing devices during an exam is regarded as cheating. The Oakland University Academic Conduct Policy can be found at http://www4.oakland.edu/?id=1610&sid=75.Cheating on examinations, plagiarism, falsifying reports/records, and unauthorized collaboration, access, or modifying of computer programs are considered serious breaches of academic conduct. The Oakland University policy on academic conduct will be strictly followed with no exceptions. See catalog under Academic Policies and Procedures.

Contributions of Course Objectives to Program Outcomes:

a) An ability to apply knowledge of computing and mathematics appropriate to the discipline;

b) An ability to analyze a problem, and identify and define the computing requirements appropriate to its solution;

c) An ability to design, implement and evaluate a computer-based system, process, component, or program to meet desired needs;

d) An ability to function effectively on teams to accomplish a common goal;

e) An understanding of professional, ethical, legal, security, and social issues and responsibilities;

f) An ability to communicate effectively with a range of audiences;

g) An ability to analyze the local and global impact of computing on individuals, organizations and society;

h) Recognition of the need for, and an ability to engage in, continuing professional development;

i) An ability to use current techniques, skills, and tools necessary for computing practice;

j) An ability to apply mathematical foundations, algorithmic principles, and computer science theory in the modeling and design of computer-based systems in a way that demonstrates comprehension of the tradeoffs involved in design choices;

k) An ability to apply design and development principles in the construction of software systems of varying complexity.