EE 5123 --- Computer Architecture (Spring 2019)
Course Syllabus

Instructor:
Lide Duan
Assistant Professor
Department of Electrical and Computer Engineering
University of Texas at San Antonio
http://utsa-salsa.org/

Office: AET 2.376
Phone: 210-458-5208
Email: lide.duan@utsa.edu
Class Meeting Time: 8:30AM to 9:45AM (Tuesdays and Thursdays)
Class Location: MB 1.102
Office Hours: 10AM to 11:45AM (Tuesdays and Thursdays) or by appointment.

Website:
All class materials will be available electronically through Blackboard Learn. These include:

- Slides, HW, solutions, and other course materials distributed by the instructor.
- Notifications sent out by the instructor (please do check your emails!)
- HW/project submissions by students.

Course Description:
Computer architecture fundamentals, instruction set architectures, memory and cache hierarchy, microprocessor pipelining, instruction-level parallelism, data-level parallelism, and thread-level parallelism.

Prerequisite:
Graduate standing. Non-experience in computer architecture is assumed. However, course projects may require programming skills in high-level languages (e.g., C, C++, Java, Python).

Textbook:
Computer Architecture: A Quantitative Approach, 6th Edition.
Authors: John Hennessy, David Patterson
Paperback ISBN: 978-0-12-811905-1
eBook ISBN: 978-0-12-811906-8
Imprint: Morgan Kaufmann
Published Date: 23rd November 2017

Course Topics (tentative):

- Fundamentals of computer architecture
- Instruction set architectures (ISA): RISC-V
- Cache and memory hierarchy design
- Pipelining
- Instruction-level parallelism (ILP)
    - Branch predictions, dynamic scheduling, multiple issue and static scheduling, speculations, compiler techniques, ILP limitations, etc.
- Data-level parallelism (DLP)

    - Vector architectures, SIMD, GPUs.
- Thread-level parallelism (TLP)
    - Centralized shared-memory, distributed shared-memory, multiprocessor cache coherence, etc.

Grading Policy:

- Homework assignments: **30%**
    - 3 assignments of 10% each
- Exams: **35%**
    - Exam 1 (10%) + Exam 2 (10%) + Final (15%)
- Projects: **30%**
    - Project 1 (15%) + Project 2 (15%)
- Quizzes: **5%**
    - The instructor will randomly choose a few classes to ask the students to turn in an answer to a very simple question. These are mainly for checking attendance.
- Total: **100%**

About the Grading

- The final letter grades will be curved based on the scores and ranks. At least 50% of the students will receive an A (including A+, A, and A-).
- After the grade of each assignment/exam/project/quiz is posted, you have up to a week to contact me for any errors. After that, the grade is finalized.

About the HW Assignments

- There will be three HW assignments; each is assigned and due before an exam.
- Tentatively, the three assignments will be assigned in the 4th, 10th, and 14th week of the semester.
- All assignments must be submitted in pdf through Blackboard Learn before the specified deadlines. There will be no delay policy, and submissions missing the deadline will not be graded.
- No copying is allowed on any assignment. All parties involved (either copying or being copied) will lose all the points for the assignment.

About the Exams

- Tentatively, Exam 1 will be on **Tuesday 2/19** (class time); Exam 2 will be on **Tuesday 4/2** (class time); and Final Exam will be on **Friday 5/10**.
- All exams are open-books and open-notes. You can bring your laptop for accessing class materials, but Internet access and communication with others are not allowed.
- The final exam will be comprehensive, covering what have been taught in the whole semester.
- No make-up exams will be allowed (except for extremely special situations with legitimate proof and under discretion of the instructor).

About the Projects:

- There will be two projects. Project 1 will be well-defined by the instructor, and Project 2 will be proposed by students.
- Project teaming policy:
    - Students can form teams to conduct the projects; each team may have up to two students.

    - Project evaluation will take into account project workload, number of team members, project quality, etc.
    - **All team members will receive the same project scores.**
    - Teams must be the same for both projects.
- Project 1 is a cache performance simulator. You will be given a text file that has a list of memory addresses, and you will need to write a program to simulate cache hit/miss behavior based on a set of input parameters, including cache size, block size, associativity, and replacement policy.
    - Timeline: teams formed by the 3rd week; project 1 assigned in the 3rd week (1/31); due in the 7th week.
- Project 2 will be proposed by students.
    - Timeline: starts in the 8th week; topics proposed by the 10th week (after the spring break); due in the 15th week; presentations in the 16th week.
    - A few examples will be given by the instructor, but the students are highly encouraged to propose their own topics.
    - Some implementation must be included; a report must be submitted.
- Students have the flexibility to choose the programming language and development environment for implementing the projects. However, you should use a free environment in order for the instructor to reproduce the results. Therefore, developing under Linux using a free compiler is highly recommended. In your submission, you will need to include a readme file (in txt) that specifies the steps to compile and run your program. It is your responsibility to make sure that your program can be run successfully by following your readme file with reasonable effort.
- Copying source code (from your classmates or students who took this course before) is strictly prohibited. Automatic source code checking tools, in addition to manual effort, will be applied to check for coding plagiarism. All parties involved (either copying or being copied) will lose all the points for the project. Therefore, protect your source code!