California State University Dominguez Hills

Computer Science Department

Course Syllabus

CSC 531 Computer Architecture, 3 units, Fall 2016

Course Information Tuesdays and Thursdays, 5:30pm-6:45pm, LIB E127

Faculty Information Dr. Bin Tang
Assistant Professor of Computer Science

office NSM E-117
office hour Mondays 3pm-4pm, Wednesdays 7pm-8pm, and Thursdays 4pm-5pm, or by appointment

contact 310-243-2510, office phone
btang@csudh.edu, email address
Emergency phone number: 310-243-3398 (CS Office)

Course Information This course provides comprehensive study of computer architectures for achieving a proportional increase in performance with increasing system resources, which are scaled by the number of processors used, the memory capacity enlarged, the access latency tolerated, the I/O bandwidth required, etc. The main focus is on the quantitative analysis and cost-performance tradeoffs in instruction-set, pipeline, and memory design. Specific topics covered include: quantitative performance measures, instruction set design, parallelism (mainly pipeline and instruction-level), and memory organization. If time allows, related topics such as data centers and cloud computing will also be covered.

Prerequisites CSC 311 Data Structures and CSC 331 Computer Organization. Specifically, students should have mastery of Boolean algebra, logic equations, binary numbers (including negatives), fixed-point binary arithmetic, hexadecimal notation, powers-of-2, logarithms, exponential numbers. Student should also have familiarity with: logic gates & diagrams, processor organization & operation, memory devices & hierarchies, and processes.

Course content and instructional method This course consists of lectures, examinations, homework assignments and programming projects, and a term project. We will have four to five homeworks and two to three programming projects. They will all be done individually. The term project is done by groups of two students. For the programming projects, we will use QtSpim (http://pages.cs.wisc.edu/~larus/spim.html), a simulator that runs assembly language programs written for processors that implement the MIPS32 architecture. It runs on it runs on Microsoft Windows, Mac OS X, and Linux. In QtSpim website, you can also find many useful resources and documents to get started. We will discuss those materials in class when time comes.

Course Objectives and Student Outcomes

The course is to provide the knowledge needed to

- Understand organized approaches to determine the capabilities and performance of computing systems
- Understand the fundamentals of different instruction set architectures and their relationship to the CPU design
- Understand and effectively utilize modern advanced computer systems, and the operations of modern CPUs including pipelining and memory systems
- Understand different systems architectures and scalable computer systems

By the end of this course, students should be able to

- Identify trade-offs in design, implementation, and performance issues in computer systems
- Design and emulate a single cycle or pipelined CPU using QtSpim simulator
- Keep up with state-of-the-art industry practice and academia research, by writing reports and making presentations about their computer architecture projects

Required Textbooks Computer Architecture: A Quantitative Approach, Fifth Edition, Morgan-Kaufmann 2012

By John L. Hennessy and David A. Patterson, ISBN-10: 012383872X, ISBN-13: 978-0123838728

CSUDH's Affordable Learning Initiatives give students free online access of the book - google "csudh csc531".

Course Calendar and Lecture Schedule

WEEK	***DATE***	***READING***	***TOPICS***
1	8/23, 25	Appendix A	Instruction Set Architecture (ISA)
2	8/30, 9/1	Appendix A	Instruction Set Architecture (ISA)
3	9/6, 8	Chapter 1	Fundamentals of Quantitative Design & Analysis
4	9/13, 15	Chapter 1	Fundamentals of Quantitative Design & Analysis
5	9/20, 22	Appendix B	Review of Memory Hierarchy Design
6	9/27, 29	Appendix B	Review of Memory Hierarchy Design
7	10/4, 6	Appendix B	Review of Memory Hierarchy Design, **Midterm**
8	10/11, 13	Chapter 2	Memory Hierarchy Design
9	10/18, 20	Chapter 2	Memory Hierarchy Design
10	10/25, 27	Appendix C	Pipelining
11	11/1, 3	Appendix C	Pipelining
12	11/8, 10	Chapter 3	Instruction-Level Parallelism and Its Exploitation
13	11/15, 17	Chapter 3	Instruction-Level Parallelism and Its Exploitation
14	11/22	Chapter 3	Instruction-Level Parallelism and Its Exploitation
15	11/29-12/1	Handouts	Data Centers and Cloud Computing
16	12/6		Term Project Presentation

Examinations **Midterm Examination is scheduled on 10/6 Thursday, in class. It covers the Chapters 1 and Appendix A and B. Final Examination is scheduled on 12/8 Thursday, 5:30pm-7:30pm. It covers Chapter 2 and 3 and Appendix C.**

No class on 11/24 Thursday, Thanksgiving Holiday.

Term Project

Each group (of two students) is expected to do a literature survey of data center and cloud computing and come up with a suitable problem for the term project. The project should propose some sort of design improvement to CPU, ISA, pipelining, or memory hierarchy and evaluate the same using a simulator or experiments on existing systems. A final report and presentation are required. The list of papers is provided as blow:

1. The current state of understanding of the energy efficiency of cloud computing (Trust, Security and Privacy in Computing and Communications (TrustCom), 2012 IEEE 11th International Conference on)
2. Data Replication and Power Consumption in Data Grids (2nd IEEE International Conference on Cloud Computing Technology and Science)
3. Cloud Computing: Virtualization and Resiliency for Data Center Computing (Computer Design (ICCD), 2012 IEEE 30th International Conference on)
4. Data Center Network Virtualization: A Survey (Communications Surveys & Tutorials, IEEE)
5. Survey on Routing in Data Centers: Insights and Future Directions (IEEE Network)
6. Cloud Computing and Grid Computing 360-Degree Compared (Grid Computing Environments Workshop)
7. PACMan: Coordinated Memory Caching for Parallel Jobs (NSDI'12)
8. Resilient Distributed Datasets: A Fault-Tolerant Abstraction for In-Memory Cluster Computing (NSDI'12)
9. Joint Optimization of Idle and Cooling Power in Data Centers While Maintaining Response Time (ASPLOS'10)
10. Leveraging Stored Energy for Handling Power Emergencies in Aggressively Provisioned Datacenters (ASPLOS'12)
11. Revisiting Storage for Smartphones (FAST'12)
12. Self-constructive high-rate system energy modeling for battery-powered mobile systems (MobiSys'11)
13. More to come…

Other venues to find related papers include IEEE International Conference on Networking, Architecture and Storage (IEEE NAS) and IEEE Symposium on High Performance Computer Architecture (IEEE HPCA).

Course grade factors

20% Homework Assignments
10% Programming Projects
30% Midterm Exam
30% Final Exam
10% Term Project

Academic Integrity Policy

The mission of CSUDH includes cultivating in each student not only the academic skills that are required for a university degree, but also the characteristics of academic integrity that are integral to a CSUDH education. It is therefore part of the mission of the university to nurture in each student a

sense of moral responsibility consistent with the biblical teachings of honesty and accountability. Furthermore, a breach of academic integrity is viewed not merely as a private matter between the student and an instructor but rather as an act which is fundamentally inconsistent with the purpose and mission of the entire university.

Consequences for violations of academic integrity in this class will automatically receive an "F" in the course and may be in jeopardy of expulsion from the university.

Academic dishonesty includes:

Cheating – Intentionally using or attempting to use unauthorized materials, information, or study aids in any academic exercise.

- Students completing any examination should assume that external assistance (e.g., books, notes, calculators, and conversations with others) is prohibited unless specifically authorized by the instructor.
- Students may not allow others to conduct research or prepare work for them without advance authorization from the instructor.
- Substantial portions of the same academic work may not be submitted for credit in more than one course without authorization.

Fabrication – Intentional falsification or invention of any information or citation in an academic exercise.

Facilitating academic dishonesty – Intentionally or knowingly helping or attempting to help another commit an act of academic dishonesty.

Plagiarism – Intentionally or knowingly representing the words, ideas, or work of another as one's own in any academic exercise.

Disability services

Students in this course who have a disability that might prevent them from fully demonstrating their abilities should meet with an advisor in the Office of Special Services as soon as possible to initiate disability verification and discuss accommodations that may be necessary to ensure full participation in the successful completion of course requirements.

Bibliography

Computer Organization and Design, Fourth Edition: The Hardware/Software Interface (The Morgan Kaufmann Series in Computer Architecture and Design) Paperback – David A. Patterson, John L. Hennessy.

Reservation of Rights

The instructor reserves the right to change this syllabus, at his sole discretion, at any time, and without prior notice. The contents of the current undergraduate catalog are incorporated herein by reference. In the event of a conflict between this syllabus and the university catalog, the university catalog shall have precedence.