CSE 6421 (Approved): Computer Architecture

Course Description

Principles and tradeoffs behind the design of modern computer architectures, including instruction-level parallelism, memory system design, advanced cache architectures, cache coherence, multiprocessors, energy-efficient and embedded architectures.

Prior Course Number: CSE 775
Transcript Abbreviation: Comp Arch
Grading Plan: Letter Grade
Course Deliveries: Classroom
Course Levels: Graduate
Student Ranks: Senior, Masters, Doctoral
Course Offerings: Autumn, Spring
Flex Scheduled Course: Never
Course Frequency: Every Year
Course Length: 14 Week
Credits: 3.0
Repeatable: No
Time Distribution: 3.0 hr Lec
Expected out-of-class hours per week: 6.0
Graded Component: Lecture
Credit by Examination: No
Admission Condition: No
Off Campus: Never
Campus Locations: Columbus
Prerequisites and Co-requisites: (CSE 3421 or CSE 5421 or CSE 675 or ECE 5362 or ECE 662) and (CSE 3431 or CSE 5431 or CSE 660)
Exclusions: Not open to students with credit for CSE 775
Cross-Listings:

The course is required for this unit's degrees, majors, and/or minors: No
The course is a GEC: No
The course is an elective (for this or other units) or is a service course for other units: Yes

Subject/CIP Code: 14.0901
Subsidy Level: Doctoral Course

Programs

Abbreviation	Description
BS CSE	BS Computer Science and Engineering
MS CSE	MS Computer Science and Engineering
PhD CSE	PhD Computer Science and Engineering

Course Goals

Master quantitative and qualitative design issues in modern architectures.
Master techniques for exploiting instruction-level parallelism.
Be familiar with instruction set architecture design principles.
Be familiar with multiprocessors and thread-level parallelism.

Be familiar with memory system design.
Master advanced cache architectures and cache coherence.
Be exposed to energy-efficient microprocessor design.
Be exposed to vector and VLIW architectures.
Be exposed to emerging directions in computer architecture.

Course Topics

Topic	Lec	Rec	Lab	Cli	IS	Sem	FE	Wor
Quantitative and qualitative design principles and introduction to modern computer architectures	5.0							
Instruction set design principles	2.0							
Techniques for exploiting instruction-level parallelism	11.0							
Multiprocessors and thread-level parallelism	6.0							
Memory system design, advanced cache architectures, cache coherence	8.5							
Energy-efficient microprocessor design	3.0							
Vector and VLIW architectures	3.0							
Architectures for embedded systems	2.0							
Emerging directions in computer architecture	1.5							

Representative Assignments

Homework assignments from the textbook and other sources.
Laboratory assignments: - Out-of-order pipeline simulation and optimization - Cache system simulation and optimization - Multiprocessor system simulation

Grades

Aspect	Percent
Homework assignments	20%
Laboratory assignments	20%
Midterm	25%
Final exam	35%

Representative Textbooks and Other Course Materials

Title	Author
Computer Architecture: A Quantitative Approach	John Hennessy and David Patterson, Morgan Kauffman, 2007

ABET-EAC Criterion 3 Outcomes

Course Contribution		College Outcome
***	a	An ability to apply knowledge of mathematics, science, and engineering.
***	b	An ability to design and conduct experiments, as well as to analyze and interpret data.
***	c	An ability to design a system, component, or process to meet desired needs.

Course Contribution		College Outcome
*	d	An ability to function on multi-disciplinary teams.
*	e	An ability to identify, formulate, and solve engineering problems.
	f	An understanding of professional and ethical responsibility.
*	g	An ability to communicate effectively.
	h	The broad education necessary to understand the impact of engineering solutions in a global and societal context.
*	i	A recognition of the need for, and an ability to engage in life-long learning.
*	j	A knowledge of contemporary issues.
*	k	An ability to use the techniques, skills, and modern engineering tools necessary for engineering practice.

BS CSE Program Outcomes

Course Contribution		Program Outcome
***	a	an ability to apply knowledge of computing, mathematics including discrete mathematics as well as probability and statistics, science, and engineering;
***	b	an ability to design and conduct experiments, as well as to analyze and interpret data;
***	c	an ability to design, implement, and evaluate a software or a software/hardware system, component, or process to meet desired needs within realistic constraints such as memory, runtime efficiency, as well as appropriate constraints related to economic, environmental, social, political, ethical, health and safety, manufacturability, and sustainability considerations;
*	d	an ability to function on multi-disciplinary teams;
*	e	an ability to identify, formulate, and solve engineering problems;
	f	an understanding of professional, ethical, legal, security and social issues and responsibilities;
*	g	an ability to communicate effectively with a range of audiences;
	h	an ability to analyze the local and global impact of computing on individuals, organizations, and society;
*	i	a recognition of the need for, and an ability to engage in life-long learning and continuing professional development;
*	j	a knowledge of contemporary issues;
*	k	an ability to use the techniques, skills, and modern engineering tools necessary for practice as a CSE professional;
***	l	an ability to analyze a problem, and identify and define the computing requirements appropriate to its solution;
**	m	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;
*	n	an ability to apply design and development principles in the construction of software systems of varying complexity.

Prepared by: Mircea-Radu Teodorescu