

CSCE 491: CAPSTONE COMPUTER SYSTEM PROJECT

Catalog Description:

491—Capstone Computer System Project. (3) (Prereq: CSCE 211, 212, 317) Advanced computer systems engineering. Digital systems design. Team projects.

Prerequisite(s) By Topic:

Digital Logic

Computer Architecture

Computer Networks

Textbook(s) and Other Required Material:

Michael Gast, *802.11 Wireless Networks: The Definitive Guide 2nd ed.*, O'Reilly Publishers, Inc., Sebastopol, CA, 2005.

(James P. Davis, *Digital Systems Design: A Model-Based Approach*, Pearson Prentice Hall Publishers, Inc., New York, NY, to be published 2006. Pre-publication lecture notes and chapters.)

Computing Platform: Unix

Course Objectives: {Assessment Methods Shown in Braces}

1. Requirements & Specification: Upon completion of this course, the student should have the ability to take a *requirements specification*, develop an understanding of this specification, and devise an effective plan for creating a digital electronics system that meets this specification; {project oral presentation, project demonstrations, project milestone deliverables, exams}
2. Modeling & Analysis: Upon completion of this course, the student should have the ability to use principles of computer engineering design to create a moderately complex *model* of a digital system using an appropriate *design representation language* formalism; {oral presentation, project demonstration, project milestone deliverables, exams}
3. Verification & Debugging: Upon completion of this course, the student should have the ability to construct effective and comprehensive *test plans* and execute *testing* and model *debugging* activities through the use of systems *simulation* and *verification* tools and methods; {demonstrations, project milestone deliverables, exams}
4. Engineering Knowledge: During participation in this course, the student should demonstrate working knowledge of *digital logic*, *computer architecture*, *software engineering*, and *algorithms* design in the creation of a system model using: *concurrency and parallelism*, *thread synchronization*, *handshaking and interrupts*, *pipelining*, *arbitration*, *coding/decoding*, *memory access*, *exception handling*, and other relevant *design patterns* employed in creating moderately complex computing systems. In addition, the student shall demonstrate, through the use of these engineering principles, the practice of good engineering design, so as to maximize *robustness* and *reuse* of the design artifacts, in addition to maximizing *throughput* of the system.
5. Engineering Knowledge: During participation in this course, the student should demonstrate the ability to acquire in-depth understanding of a moderately complex

computer engineering problem domain--where such knowledge allows the student to ask and answer questions about potential solutions in the *design space*, explore potential solutions to *design challenges* presented in this domain, and assess potential *design tradeoffs* through analysis and inquiry; { project milestone deliverables, exams }

6. **Performance Assessment:** During participation in this course, the student should demonstrate the ability to measure and assess system performance through simulation and analysis, demonstrating that the resultant design solution meets stated performance requirements; { project demonstration, project milestone deliverables }
7. **Project Documentation:** During participation in this course, the student should show ability to document and explain work on a project of significant complexity, through tracking of effort distribution, providing explanation of design assumptions, and reporting on results of project implementation activities; { project milestone deliverables }
8. **Team Participation:** During participation in this course, the student should show ability to work as a part of a small team and of a larger group to accomplish a large multi-step project. {oral presentation, project demonstration, project reports }

Topics Covered: Because of the project-oriented nature of this course, specific course hours are not given; the topics are interwoven.

1. High-level digital logic and systems design.
2. Requirements specification, verification, and validation.
3. Object-oriented hardware and software engineering design: architecture analysis, robustness analysis, detailed design, facilitating reuse.
4. Computer architecture (pertaining to serial data-oriented protocols and architectures).
5. IEEE 802.11 b/g wireless networking protocol and architecture.
6. Performance assessment and system verification (function, timing) using software simulation.
7. Design tradeoff analysis and design space exploration.

Laboratory Projects

The entire course is centered on students' work in the lab—designing a digital systems application of moderate complexity, verifying the functionality and timing of the model for this system, measuring the system's performance and conformance to stated performance objectives. The laboratory work is augmented with 15-20 hours of lecture on the relevant engineering knowledge areas required to carry out the design project (as listed above and below). The laboratory-based assessment is augmented through use of 2 in-class examinations that test student comprehension and understanding of the knowledge areas being used.

Syllabus Flexibility: Medium. The Undergraduate Committee approves the choice of textbook; project details vary every semester.

Relationship of Course to Program Outcomes:

The contribution of each course objective to meeting the program outcomes is indicated with the scale:

3 = major contributor, 2 = moderate contributor, 1 = minor contributor. Blank if not related.

Course Objectives	Program Outcomes										
	1. Logic & Math	2. Computing Fundamentals	3. Apply Computing Principles	4. Work on teams	5. Communicate Effectively	6. Liberal arts & Soc. Sciences	7. Basic Science and Lab Procedures	8. Learn New Tools & Processes	9. Employed upon Graduation	10. Application Area	11. Electronics and Digital Sys Design
1. Requirements & specification	2	2		2	2			2	2	3	2
2. Modeling & analysis		3	3	2	2			3	3	3	3
3. Verification & debugging	1	3	3	2	2		3	3	1	3	3
4. Exploiting background engineering knowledge	2	3	3	1	1		2		1	3	3
5. Acquiring new/novel engineering knowledge (lifelong learning)		2	2	1	2	1			1	3	2
6. Performance assessment		2	2	2			3		3	3	2
7. Project documentation				3	3	2	3		3		
8. Team participation				3	3	2			3		

Estimated Computing Category Content (Semester hours):

Computer Science majors do not take this course.

Estimated Information Systems Category Content (Semester hours):

Computer Information Systems majors do not take this course.

Oral and Written Communication:

For each stated project milestone deliverable, students must provide documentation on their designs, and they must draft a verification test plan and report of test results. The students, as members of teams, must be able to orally present, discuss and answer questions about their design artifacts. They must submit and present an end-of-project presentation (in PowerPoint), and provide a final report of all deliverables. The students must account for all time involved in project activities through regular submission of *effort distribution* reports showing task-related breakdown of effort on project activities.

Social and Ethical Issues: (1) Principles of competitive secrecy (non-disclosure), (2) Principles of design robustness, product quality and qualitative system reliability, (3) Professional behavior on project team.

Theoretical Content:

Object-oriented analysis and design methods

Digital logic, finite state machines (FSM) and register-transfer level (RTL) modeling of digital systems

Serial data transfer protocols (wired and wireless)

Analysis and Design:

Design and analysis of system, design verification of system functionality and timing behavior, and measuring and assessing performance

Class/Laboratory Schedule:

Lecture: 1 period of 75 minutes per week

Recitation: Generally one group meeting with instructor each week

Laboratory: 1 period of 75 minutes per week. In addition, students have free access to laboratory

Course Coordinator: Jim Davis

Modification and Approval History:

Initial description, January 2000

Revised, April 2001

Revised June 2005 by Jim Davis for content and by Caroline Eastman for format