

CSCE 611: DIGITAL ELECTRONICS SYSTEMS DESIGN

Catalog Description:

611—Digital Electronics Systems Design. (3) (Prereq: CSCE 211, 212, 245) Design techniques and methods for digital electronic systems; emphasis on using higher-level EDA tools such as hardware description languages and conceptual modeling; emphasis on architecture level design exploration and tradeoff analysis.

Prerequisite(s) By Topic:

Digital logic
Computer architecture
Programming in a high level language

Textbook(s) and Other Required Material:

Wayne Wolf, *FPGA-Based System Design*, Prentice-Hall, Englewood Cliffs, NJ, 2004.

Computing Platform: Unix, Linux, Windows XP platforms; Nimbus, Synopsys, Xilinx EDA tools.

Course Objectives: {Assessment Methods Shown in Braces}

1. Modeling & Analysis: Upon completion of this course, the student should have the ability to use principles of computer engineering design to create complex *models* of digital circuits and subsystems using an appropriate *design representation language* formalism; {oral presentation, final report, exams}
2. 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 deliverables, final report, exams}
3. Circuit Synthesis and Layout: Upon completion, the student should have the ability to use tools to synthesize circuits from finite state machine and register-transfer notation constructs of algorithms and protocols, should have the ability to understand the gate-level structure of these circuits and to take the netlist of a synthesized circuit and place and route it onto a target programmable logic device, and should have the ability to obtain post-layout performance data and interpret the results in light of system design requirements and constraints.
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 employing: *concurrency and parallelism*, *thread synchronization*, *handshaking and interrupts*, *pipelining*, *arbitration*, *coding/decoding*, *memory access*, *exception handling*, and other relevant *design patterns* used 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 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—such as arithmetic, coding, and computing and control applications—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 (Pareto) 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 and post-layout Pareto-optimal solution analysis, demonstrating that the resultant design solution both meets stated performance requirements and is reasonably optimal when considering alternate design solutions; {project demonstration, project deliverables, final report}
7. **Project:** During this course, the student will demonstrate the ability to plan, coordinate and execute tasks allowing the completion of a significant digital circuit/systems design project; for undergraduates, this also entails working in a small team setting, whereas for graduate students, this entails conducting the project from an applied research frame of reference {project milestone deliverables, final report}
8. **Project Reporting and 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. For graduate students, this objective is that the student work should be of a quality that a paper could be subsequently written based on the work completed in this course; {project milestone deliverables, final report}

Topics Covered:

1. The digital systems design process (6)
2. Levels of abstraction: algorithm, register transfer, gate, and switch levels (6)
3. Using programmable logic (FPGAs) for realizing digital systems (9)
4. The Algorithmic State Machine (ASM) method (9)
5. Systems analysis for digital systems design: Performance metrics, Pareto optimal solutions analysis, and tradeoffs (9)
6. Review and examinations (3)

Laboratory Projects: Students complete one major VLSI design project in addition to homework assignments and examinations. Typical project domains: computer arithmetic circuits (Adders and Multipliers), CPU logic circuits (68000 instruction decoder), coding circuits (multi-bit resolution error correction codes CRC, Huffman and Reed-Solomon; MPEG compression code), and network protocol circuits (802.3, 802.11 PHY). Students who have completed the CSCE 491's 802.11 MAC project may take on one of several possible functional extensions to their earlier design.

Difference between Undergraduate and Graduate Work: Graduate students complete a more complex design project that involves a more rigorous investigation into a specific design application area; project reports written by graduate students are to be of sufficient depth and quality that they could be used as a basis for writing a conference publication. Graduate students must complete additional assignments as part of homework, and are

graded more rigorously on examinations. Graduate students generally must work alone on projects whereas undergraduates may work in teams of 2 students.

Syllabus Flexibility: High. Choice of textbook and design project determined by instructor

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. Modeling and analysis			3					3	3		3
2. Verification and debugging			3				1	3	2		3
3. Circuit synthesis and layout			3					3	2		3
4. Application of engineering knowledge			3						3	2	3
5. Acquisition of engineering knowledge (lifelong learning)	2	2	3					3	2	3	2
6. Design performance assessment			3				1	3	2	2	2
7. Project execution			3	1					3	1	2
8. Project reporting and documentation			1		2				3		1

Estimated Computing Category Content (Semester hours):

Area	Core	Advanced	Area	Core	Advanced
Algorithms			Data Structures		
Software Design		1	Programming Languages		
Computer Architecture		2			

Estimated Information Systems Category Content (Semester hours):

Computer Information Systems majors do not take this course.

Oral and Written Communication:

Project report and engineering documentation for major VLSI design project; end-of-project presentation (PowerPoint) and demonstration of project results

Social and Ethical Issues: None

Theoretical Content: Computer arithmetic (3), Pareto analysis (1)

Analysis and Design:

Design and analysis of system, design verification of system functionality and timing behavior, measuring and assessing performance, and evaluating design tradeoffs (Pareto optimal solution analysis)

Class/Laboratory Schedule:

Lecture: 3 periods of 50 minutes or 2 periods of 75 minutes per week

Course Coordinator: Jim Davis

Modification and Approval History:

Initial description April 1999

Revised, June 2001

Revised June 2005 by Caroline Eastman to modify format

Revised June 2005 by Jim Davis to modify textbook and content