Catalog Description:
611-Conceptual Modeling Tools for CAD. (3) (Prereq: CSCE 211, 245) Design techniques for logic systems; emphasis on higher-level CAD tools such as hardware description languages and conceptual modeling.

Textbook(s) and Other Required Material:
Course webpage:  http://www.cse.sc.edu/~jbakos/611 (tutorials and additional instruction for lab work)

Course Outcomes:
After completing this course students should be able to:
1. Design large-scale digital systems using VHDL
2. Perform behavioral verification using test benches and behavioral simulation
3. Write and simulate programs in MIPS assembly language using SPIM (an assembler and simulator)
4. Design a full microprocessor that implements the MIPS instruction set and interfaces a memory system

Topics Covered:
- VHDL digital design flow
- Design methodologies and techniques
- MIPS instruction set architecture
- Microarchitecture design
- Test bench design
- Memory models
- Bus models and interface design
- Exceptions and interrupts
- Memory hierarchy and cache subsystems
Assessment of Learning by Course-Outcome:

### Summary of Results

<table>
<thead>
<tr>
<th>Coursework</th>
<th>Topic</th>
<th>OC. 1</th>
<th>OC. 2</th>
<th>OC. 3</th>
<th>OC. 4</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>CE</td>
<td>EE</td>
<td>Grad</td>
<td>CE</td>
</tr>
<tr>
<td>Lab 1</td>
<td>Design of a 32-bit ALU and corresponding test bench and behavioral verification</td>
<td>94%</td>
<td>100%</td>
<td>99%</td>
<td>94%</td>
</tr>
<tr>
<td>Lab 2</td>
<td>MIPS instruction set architecture</td>
<td></td>
<td></td>
<td>100%</td>
<td>100%</td>
</tr>
<tr>
<td>Lab 3</td>
<td>Multi-cycle CPU design and verification</td>
<td>84%</td>
<td>85%</td>
<td>83%</td>
<td>84%</td>
</tr>
<tr>
<td>Lab 4</td>
<td>Memory model integration and interface design</td>
<td>87%</td>
<td>80%</td>
<td>84%</td>
<td>87%</td>
</tr>
<tr>
<td>Lab 5</td>
<td>Exceptions and interrupts (ugrad), Unified cache design (grad)</td>
<td>90%</td>
<td>38%</td>
<td>95%</td>
<td>90%</td>
</tr>
<tr>
<td>Final Exam</td>
<td>Design practicum</td>
<td>90%</td>
<td>74%</td>
<td>68%</td>
<td>90%</td>
</tr>
<tr>
<td><strong>Average</strong></td>
<td></td>
<td>89%</td>
<td>75%</td>
<td>86%</td>
<td>90%</td>
</tr>
</tbody>
</table>

* Averages based on 2 Computer Engineering, 0 Computer Science, 2 Electrical Engineering, 7 graduate students (or undergraduates taking for graduate credit) who completed the course.
Outcome 1. Design large-scale digital systems using VHDL

Coverage of Material

This outcome is covered through the following mechanisms:

1. **A series of lectures the span the term:** Lectures are given that cover behavioral logic design mythology, design flows, logic verification, test bench design, and the MIPS microarchitecture and memory system.

2. **A series of web-based design tutorials:** Students work through a series of tutorials that guide them through the VHDL behavioral design and verification flow.

3. **A series of guided lab assignments:** Students complete a series of lab assignments that develop their ability to use the VHDL behavioral design and verification flow to design large-scale digital systems, including a fully function CPU design that implements the MIPS instruction set architecture. Students are given approximately 40% of the class time to work on these assignments and be given individual guidance by the instructor. Students must also devote a significant amount of time outside of class to complete these lab assignments.

Measurements

This outcome is measured using labs 1, 3, 4, 5, and the final exam. The lab assignments are graded relative to the number of design requirements that were fulfilled.

Outcome 2. Perform behavioral verification using test benches and behavioral simulation

Coverage of Material

Each lab assignment requires that students verify the functionality of their designs using their own test bench designs along with behavioral simulation. Lab 1 introduces students to this technique, where they design a 32-bit ALU from online tutorials but must design their own custom test bench to verify the functionality of the ALU.

Measurements

Labs 3, 4, and 5 require the students to verify their designs with the test bench technique. The final exam requires that the students design a device that passes a test bench that is designed by the instructor.

Outcome 3. Write and simulate programs in MIPS assembly language using SPIM (an assembler and simulator)

Coverage of Material

The MIPS instruction set architecture is presented to the students through a series of lectures. The students use the MIPS assembly language and instruction set architecture as a set of specifications to design and verify their CPU.
Measurements

Lab 2 requires the students to write, assemble, and test a program in MIPS assembly language using the SPIM simulator.

Outcome 4. Design a full microprocessor that implements the MIPS instruction set and interfaces with a memory system

Coverage of Material

Approximately three quarters of the semester involve the incremental design and testing of a MIPS microprocessor.

Measurements

Labs 3, 4, and 5 measure this outcome, and consist of the piece-wise development of a CPU using the design and test methodology presented in this course.

Grade Distribution from Spring 2007:

<table>
<thead>
<tr>
<th>Grade</th>
<th>CE</th>
<th>EE</th>
<th>Grad</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>1</td>
<td>1</td>
<td>5</td>
</tr>
<tr>
<td>B+</td>
<td>1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>B</td>
<td></td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>C+</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>C</td>
<td>1</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>D+</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>D</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>I</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>F</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Differences Between Undergraduate and Graduate Work:

Undergraduates and graduates are assigned different final projects. The undergraduates are required to add exceptions and interrupts to their CPU, while the graduate students are required to design a unified primary cache. The undergraduate project requires less effort and scope as compared to the graduate project.

Changes

May 2007:
When I took over teaching this course, I performed an extensive redesign and update of the course. I am not familiar with the previous incarnation of the course.
-Jason D. Bakos

Major Changes
N/a.

**Minor Changes**

N/a.

**Planned Changes**

None.

**Effect on Outcomes**

N/a.