COURSE DESCRIPTION

**Department and Course Number:** CSc 242

**Coordinator:** Behnam S. Arad

**Course Title:** Computer-Aided Systems Design and Verification

**Units:** 3

**Catalog Description**

Design and verification methodology using hardware description and verification languages (HDVLs); advances in IC chip design; introduction to HDVLs such as SystemVerilog; HDVL language basics including data types, arrays, structures, unions, procedural blocks, tasks, functions, and interface concept; design hierarchy; verification planning and productivity; verification infrastructure; guidelines for efficient verification of large designs; assertion-based verification; comprehensive computer-related design projects.  Prerequisite: CSc 205.

**Textbook**


**References**


**Course Goals**

To learn:
- The latest techniques used to design and verify complex systems.
- HDVL enhancements for design and verification.
- Verification guidelines for HDVL’s.
- Assertion-based verification.

**Prerequisites by Topic**

*Thorough understanding of:*
- Application of standard combinational devices.
- Design and implementation of sequential circuits.
- Computer organization.
- Proficiency in High-level Programming Languages.

*Exposure to:*
- Object oriented programming.
- IC fabrication process.
- Hardware description languages.
Course Topics

- Overview of IC chip design process (2 hours).
- Review of design using hardware description languages (3 hours).
- Introduction to hardware description and verification languages (HDVLs) (1 hour).
- HDVL basics including data types, structures, unions, and arrays (5 hours).
- Procedural blocks (2 hours).
- Design hierarchy (3 hours).
- Interface concept and its application (6 hours).
- Verification planning and productivity (1 hour).
- Verification infrastructure (5 hours).
- Object-oriented approach in verification (6 hours).
- Inter-process communication in SystemVerilog (1 hour).
- Verification methodology using HDVLs (5 hours).
- Case studies involving computer design (5 hours).

Laboratory Projects

The following items describe suggested assignments that students should complete outside of class periods.

- Learning how to simulate a SystemVerilog design.
- Utilizing enhanced data types in System Verilog in a design.
- Programming experience with arrays, structures and unions.
- Developing SystemVerilog testbenches using the interface concepts.
- Programming experience to understand design hierarchy in SystemVerilog.
- Utilizing object programming features of SystemVerilog.
- A comprehensive project that involves design and verification of a system using SystemVerilog language and its verification guidelines.

Estimated Curriculum Category Content (Semester hours)

<table>
<thead>
<tr>
<th>Area</th>
<th>Core</th>
<th>Advanced</th>
<th>Area</th>
<th>Core</th>
<th>Advanced</th>
</tr>
</thead>
<tbody>
<tr>
<td>Algorithms</td>
<td></td>
<td></td>
<td>Data Structures</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Software Design</td>
<td></td>
<td></td>
<td>Prog. Languages</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Comp. Arch.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Oral and Written Communications

The term project requires a written report on design and verification of a complex system as well as an oral presentation.

Social and Ethical Issues

No significant component.

Theoretical Content
Introduction to assertion-based verification and formal verification in hardware design.

Analysis

Students study the impact of the coding style on verification efficiency.

Design

The course involves significant hardware design projects using a hardware description and verification language.