CSCE 4610/5610: Computer Architecture

General Information
Instructor: Tong Shu  
Office: Discovery Park F277  
E-mail: Tong.Shu@unt.edu  
Office hours: 3:00 PM - 4:00 PM on Mon. & Wed.

Course Description
Study of performance issues and power requirements related to modern computer systems, including instruction level parallelism, out-of-order instruction scheduling, branch prediction, speculative execution, cache memory, and concurrency.

Prerequisites
C programming in Linux is required.  
CSCE 2610 - Assembly Language and Computer Organization  
or CSCE 3610 - Introduction to Computer Architecture  
or CSCE 3612 - Embedded Systems Design

Required Textbook

Reference Books

Evaluation
Grading components for CSCE 4610/5610:

<table>
<thead>
<tr>
<th>Component</th>
<th>Weight</th>
</tr>
</thead>
<tbody>
<tr>
<td>Homework (individual)</td>
<td>30%</td>
</tr>
<tr>
<td>Project (3-student team)</td>
<td>30%</td>
</tr>
<tr>
<td>Quiz (closed-book)</td>
<td>10%</td>
</tr>
<tr>
<td>Final exam (closed-book)</td>
<td>30%</td>
</tr>
</tbody>
</table>

4-5 homework assignments on problem solving  
Course project: selective from 2-3 project options (e.g., architectural simulators and parallel programming)  
   Project proposal (2 pages)
Project report (6 pages with at least 8 references) plus hands-on exercises (e.g. source code)

Note that the document format follows ACM proceedings template ( https://www.acm.org/publications/proceedings-template )

Grading scale for CSCE 5610*:

<table>
<thead>
<tr>
<th>Grade</th>
<th>Score</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>90 – 100</td>
</tr>
<tr>
<td>B</td>
<td>80 – 89</td>
</tr>
<tr>
<td>C</td>
<td>70 – 79</td>
</tr>
<tr>
<td>D</td>
<td>60 – 69</td>
</tr>
<tr>
<td>F</td>
<td>59 and below</td>
</tr>
</tbody>
</table>

*The grading scale for CSCE 4610 is 5 points lower than that for CSCE 5610.

**Final grades will not be curved unless necessary.

Late Policy:
Students are expected to complete work on schedule. Late work within 24 hours can only have 80% credit for delay penalty. Late work beyond 24 hours is not accepted with any excuse. For multiple submissions with partial delay, 80% credit is only applied to the delayed portion.

**Academic Integrity and Student Conduct**

*Plagiarism or cheating* behavior in any form is unethical and detrimental to proper education and will not be tolerated. All work submitted by a student (projects, programming assignments, lab assignments, quizzes, tests, etc.) is expected to be a student's own work. The plagiarism is incurred when any part of anybody else's work is passed as your own (no proper credit is listed to the sources in your own work) so the reader is led to believe it is therefore your own effort. Students are allowed and encouraged to discuss with each other and look up resources in the literature (including the internet) on their assignments, but appropriate references must be included for the materials consulted, and appropriate citations made when the material is taken verbatim.

This course follows the Department of Computer Science and Engineering Cheating Policy. If plagiarism or cheating occurs, the student will receive a failing grade on the assignment and (at the instructor’s discretion) a failing grade in the course. Specifically, students caught cheating or plagiarizing will receive a “0” for that particular assignment or exam for the first offense. Additionally, the incident may be reported to the Dean of Students, who may impose a further penalty. The second instance of cheating in this class will result in a grade of F in the class, and referral to the Department Chairperson and Dean of Engineering, whereby a dismissal hearing may be initiated by the Dean of Engineering.
Individual assignments, including laboratory exercises and programming assignments, in this course, must be the sole work of the individual student. You should not work with other students on shared program solutions or use solutions found on the Internet. Specifically, you should never copy someone else’s solution or code, and never let a classmate examine your code. If you are having trouble with an assignment, please consult with your instructor or TA/IA assigned to this course. Failure to adhere to these strict standards may be cause for disciplinary action even leading to expulsion from the University.

Students are responsible for being familiar with the university standard for academic integrity. In the case that the above description or any in-class discussion of appropriate and inappropriate collaboration does not answer all of your questions, please meet with your instructor and look at the university Student Rights and Responsibilities web page.

**Course Syllabus**

Chapter 1: Fundamentals of quantitative design and analysis (2.5 hours)
Appendix A: Instruction Set Principles (2.5 hours)
Appendix B: Review of Memory Hierarchy (2 hours)
Chapter 2: Memory hierarchy design (4.5 hours)
Appendix C: Pipelining: basic and intermediate concepts (2.5 hours)
Chapter 3: Instruction-level parallelism and its exploitation (6 hours)
Chapter 4: Data-level parallelism in vector, SIMD, and GPU architectures (6 hours)
Chapter 5: Thread-level parallelism (6 hours)
Chapter 6: Warehouse-scale computers to exploit request-level and data-level Parallelism (3 hours)

This syllabus is subject to change based on the needs of the class.