Syllabus

<table>
<tr><td>Course code</td><td>ECE 650 (graduate)</td><td>Course name</td><td colspan="2">Advanced Computer Architecture</td><td>credits</td><td>3</td></tr>
<tr><td rowspan="3">Instructor</td><td colspan="3">Name : Hyokeun Lee</td><td colspan="3">Homepage : https://relacslab.github.io/</td></tr>
<tr><td colspan="3">E-Mail : hyokeunlee@ajou.ac.kr</td><td colspan="3">Office : 원천관 403</td></tr>
<tr><td colspan="6">Office hour : appointment is recommended</td></tr>
<tr><td>1. Goals</td><td colspan="6">Well defining a computer architecture is a key to achieve a robust and high-performance system. Based on the knowledge you learn in “Computer Organization and Architecture” class, this lecture will take you to the road on various advanced computer architecture technologies (e.g. out-of-order, prediction, coherence...). Although these technologies will be taught based on general-purpose processors, they are ubiquitously extended in various computing platforms.</td></tr>
<tr><td>2. Textbooks</td><td colspan="6"><u>Main</u>: Patterson and Hennessy, “Computer Architecture: A Quantitative Approach”
<u>Sub</u>: A. Gonzalez et al., “Processor Microarchitecture: An Implementation Perspective”</td></tr>
<tr><td>3. Prerequisites</td><td colspan="6">▪ Computer Organization and Architecture
▪ ECE Programming (C/C++)</td></tr>
<tr><td rowspan="2">4. Ratings (%)</td><td>Attendance</td><td>Homework</td><td>Mid-term</td><td>Final-term</td><td>Project</td><td>Others</td><td>Overall</td></tr>
<tr><td>5</td><td>0</td><td>35</td><td>35</td><td>25</td><td>0</td><td>100</td></tr>
<tr><td rowspan="17">5. Agenda</td><td>Week</td><td colspan="5">Contents</td></tr>
<tr><td>1</td><td colspan="5">Introduction to Computer Architecture</td></tr>
<tr><td>2</td><td colspan="5">Recap of ISA and Simple Microarchitecture</td></tr>
<tr><td>3</td><td colspan="5">Out-of-Order Microarchitecture (1)</td></tr>
<tr><td>4</td><td colspan="5">Out-of-Order Microarchitecture (2)</td></tr>
<tr><td>5</td><td colspan="5">Out-of-Order Microarchitecture (3)</td></tr>
<tr><td>6</td><td colspan="5">Branch Prediction</td></tr>
<tr><td>7</td><td colspan="5">Cache and Data Prefetch</td></tr>
<tr><td>8</td><td colspan="5">Mid-Term Exam</td></tr>
<tr><td>9</td><td colspan="5">Shared Memory: Cache Coherence (1)</td></tr>
<tr><td>10</td><td colspan="5">Shared Memory: Cache Coherence (2)</td></tr>
<tr><td>11</td><td colspan="5">Shared Memory: Memory Consistency (1)</td></tr>
<tr><td>12</td><td colspan="5">Shared Memory: Memory Consistency (2)</td></tr>
<tr><td>13</td><td colspan="5">Virtual Memory: Improving PTW Performance</td></tr>
<tr><td>14</td><td colspan="5">Virtual Memory: Enhancements for Various Domains</td></tr>
<tr><td>15</td><td colspan="5">Final-Term Exam</td></tr>
<tr><td>6. Notes for students</td><td colspan="6">▪ F will be given if cheating is caught no matter what case is
▪ One grade lower if not taking either mid-term or final-term exam
▪ This class has a Verilog project, please refrain from taking this class if you have not taken “Digital System Design” class</td></tr>
</table>