Course Details

Instructor: Vivek Pathak
Web: [http://www.cs.stevens.edu/~vpathak/](http://www.cs.stevens.edu/~vpathak/)
Email: vpathak+488@cs.stevens.edu
Office: Lieb 307
Office Hours: Monday 2PM-3PM or by appointment.
Telephone: (201) 216-5048
Class: MRF 01:00-01:50PM in Babbio Center for Technology, Room 319

Prerequisites

CS 383 and co-requisite MA 222, or equivalent.

Objectives

This course provides an introduction to the functional level structure of modern pipelined processors and the empirical and analytic evaluation of their performance. Topics include: empirical and analytic techniques for measuring performance (use of various means, Amdahl’s Law, and benchmarks); tradeoff analysis; principles of instruction set design and evaluation (memory addressing, operations, types and sizes of operands, instruction set encoding, CISC vs. RISC, and related compilation issues); pipelining (basics, data hazards, and control hazards); and memory systems.

At the end of this course, students shall be able to:

1. State and use the basic quantitative principles of computer design and performance measurements.
2. State and consider the role of today’s technology and the factors affecting the cost of computer systems.
3. State and consider price-performance measurements of processors designed for desktop, server, and embedded systems, as well as the role of power efficiency in design of processors.
4. Design basic and intermediate RISC pipelines, including the instruction set, data paths, and ways of dealing with pipeline hazards.
5. Compare various pipeline designs and performance, while identifying the tradeoff between performance and hardware complexity.
6. Consider use of dynamic scheduling in processor design, and use of basic methods to implement dynamic scheduling.
7. Consider various techniques of instruction-level parallelism, including superscalar execution, branch prediction, and speculation, in design of high-performance processors.
8. Consider use of static approaches to exploiting instruction-level parallelism to improve performance by relying on more sophisticated compiler technology.

9. State and understand memory hierarchy design, memory access time formula, performance improvement techniques, and trade-offs.

10. Compare the effect of memory hierarchy (main and caches) designs using benchmark data.

11. State and compare various memory organizations, bandwidth optimizations, and the effect on their cost-performance.

12. State and compare properties of multiprocessor systems shared and distributed memory architectures, and their application domain, evaluating both organizational principles and effect on performance.

13. State and understand the cache coherency in multiprocessors and the protocols for achieving it.

14. State and understand the computer organization, including CPU-memory bus, I/O buses, interfaces, and communication.

15. State and understand the role of disk systems in providing reliability, availability, their MTTF and MTTR.

16. State and compare the characteristics of magnetic disks and their organization, evaluation of throughput based on speed and organization, etc.

17. Use Little's queuing theory to measure the performance of some basic I/O systems.

18. State the organization of a vector processor, its application domain, performance comparison with a uniprocessor, and their application domains.

Homework exercises are assigned and graded. Students are tested through exam questions.

Text

Title: Computer Architecture: A Quantitative Approach. 4th Ed.
Author: John L. Hennessy and David A. Patterson
ISBN: 0123704901
Publisher: Morgan Kaufmann

Grading

There will be several homework assignments. Each is like a take-home exam. These assignments account for a good percentage of your grade while also providing practice for the mid-term and final exams. Students are expected to work individually on assignments. Homework may have a different weight depending on the size of topics.

There will be two exams: one midterm exam, and one final exam. The final exam is comprehensive, but a greater amount of emphasis will be placed on the second part of the course. Grading will be assigned according to the following scheme:

- Assignments: Every week or couple of weeks with total weightage 50%
- Midterm: 20%
- Final: 30%

Outline

<table>
<thead>
<tr>
<th>Week</th>
<th>Topics</th>
<th>Reading</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>Introduction. Fundamental of Computer Design</td>
<td>Chapter 1</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Topic</td>
<td>Chapter/App.</td>
<td></td>
</tr>
<tr>
<td>---</td>
<td>------------------------------------------------------</td>
<td>--------------</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>Quantitative Approach to Design</td>
<td>Chapter 1</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>Classifying Instruction Set Architectures</td>
<td>Appendix B</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>Basic Concepts of Pipelining</td>
<td>Appendix A</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>Intermediate Concepts of Pipelining</td>
<td>Appendix A</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>Instruction Level Parallelism</td>
<td>Chapter 2</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>Midterm Exam</td>
<td></td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>Limits on Instruction Level Parallelism</td>
<td>Chapter 3</td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>Memory Hierarchy Design</td>
<td>Appendix C</td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>Memory Performance</td>
<td>Chapter 5</td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>Storage Systems</td>
<td>Chapter 6</td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>Multi-Processor and Thread-level Parallelism</td>
<td>Chapter 4</td>
<td></td>
</tr>
<tr>
<td>13</td>
<td>Multi-Processor and Thread-level Parallelism</td>
<td>Chapter 4</td>
<td></td>
</tr>
<tr>
<td>14</td>
<td>Vector Processors</td>
<td>Appendix F</td>
<td></td>
</tr>
</tbody>
</table>

On Chapters 1, 2 and Appendices A, B

_Last modified on January 16, 2011._