COURSE DESCRIPTION

Dept., Number  CSC 142  Course Title  Advanced Computer Organization
Semester hours  3  Course Coordinator  Behnam Arad
URL (if any):  http://gaia.ecs.csus.edu/~arad/

Catalog Description

Design and performance issues of computers: CPU, I/O interface, and memory. Design alternatives for arithmetic functions, CPU internal architecture, instruction set, instruction cycle, I/O, interrupt, direct memory access, and bus and memory hierarchy. CAD tools for schematic capture and simulations. Students will design and simulate a microcomputer. Prerequisite: At least a C- grade in CSC 137 or equivalent and full CSC or CPE major status. Cross-listed as CPE 142; only one may be counted for credit.

Textbook


References

Verilog hardware description language.

Course Goals

To provide students with an understanding of:

1. Arithmetic unit algorithm design tradeoffs.
2. Control unit design tradeoffs.
3. CPU instruction set design issues.
4. CPU data path design issues.
5. Input/output interface design.
6. Memory hierarchy.
7. Multiple processors.

Prerequisites by Topic

*Thorough understanding of:*

- Combinational circuit design.
- Sequential circuit design.
- Machine language programming.
- Instruction execution cycle.
- Memory internal origination.
Basic understanding of:
- CPU data path.
- Control unit design.
- Memory hierarchy.
- Bus cycle.
- I/O interface.
- Direct memory access.

Exposure to:
- Computer arithmetic.
- Instruction pipelining.
- Multiprocessor architecture.

Major Topics Covered in the Course

1. Introduction (signal timing and register clocking) (1 hr).
2. Introduction to Verilog HDL (3 hrs).
3. Arithmetic algorithm design (6 hrs).
4. CPU architecture: RISC/CISC, data path design: single cycle, multiple cycle, and pipelining; instruction set design; register windows; compiler optimization; branch prediction, VLIW, and EPIC; survey of contemporary architectures (11 hrs).
5. CPU control unit design (3 hrs).
7. Memory system design principles (1 hrs).
9. Interleaved memory architecture, memory bandwidth improvement (2 hrs).
10. Direct memory access, virtual memory and MMU architecture (3 hrs).
11. Introduction to arithmetic pipelining and multiprocessor architecture (3 hrs).
12. Review and exam (4 hrs).

Outcomes

Thorough understanding of:
- Instruction pipelining.
- Control unit design.
- Memory hierarchy and optimization.

Basic understanding of:
- Integer computer arithmetic.
- Floating point arithmetic.
- Branch prediction and speculative execution.
- Super scalar and super pipelined instruction data path.
- Improving memory bandwidth (memory architecture).
- Improving data transfer rate (bus architecture).
Exposure to:
- Compiler optimization for pipelined data path.
- Compiler optimization for better cache memory performance.
- Data coherency in multiprocessor systems.
- Introduction to parallel architecture.

Laboratory Projects

1. Each student will work on several projects in various areas covering topics from advanced computer arithmetic, micro programmed control, instruction pipeline design, cache memory design, etc. (6 weeks).
2. Other assignments would focus on performance and analysis.

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>0.5</td>
<td>2.5</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Oral and Written Communications

No significant component.

Social and Ethical Issues

No significant component.

Theoretical Content

This is the undergraduate advanced computer architecture/organization elective course for the computer science degree. The course covers the theory of space vs. time optimization of arithmetic algorithm design, RISC vs. CISC, instruction pipelining, parallel vs. pipelining, cache memory coherency, MMU, and system scalability.

Problem Analysis

Alternative arithmetic algorithms designs and their implementation area vs. time trade offs are analyzed; various alternative CPU data paths and their performance trade offs are analyzed; the pros and cons of CPU control design alternatives are analyzed; the effect of various memory system organizations on the average CPU memory access time is analyzed; impact of branch prediction and compiler optimization on CPI, and the analysis of various Verilog HDL design simulation results.
Solution Design

Various arithmetic algorithms (adder, multiplier, etc.) design; alternative control unit designs; cache memory design; design using Verilog HDL; design of data path (e.g., CPU, cache, etc.)

/aa