Lab 2 Overview

CSE141L
4/25/2007
## Course Overview

<table>
<thead>
<tr>
<th>Individual</th>
<th>Team</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Lab 1</strong></td>
<td><strong>Lab 2</strong></td>
</tr>
<tr>
<td>Know your tools</td>
<td>Design your own ISA</td>
</tr>
<tr>
<td>Implement a fetch unit step by step</td>
<td>Verify your ISA design by implementing an emulator with 3 small benchmarks</td>
</tr>
<tr>
<td></td>
<td>Implement the backend of the processor</td>
</tr>
<tr>
<td></td>
<td>Evaluate the performance of your processor</td>
</tr>
<tr>
<td></td>
<td>(extra credit) Implement a multicore microprocessor system</td>
</tr>
</tbody>
</table>
Team Policy

• You can work in a team of 2.
  – Highly recommended
  – However, if your team mate drops this class, you must finish the project alone.
  – Pick your partner wisely!

• Alternatively, you may work alone.

• Moderately higher expectations for teams of 2 over teams of 1.
Team Website

• Create your account
  – http://app2you.org/cse141lab/profile
  – Must use your UCSD account
• Create a team
• Select your team
• Make your team profile

Do it today!
### CSE 141L

#### Navigation
- Student Profile
- Team Proposal
- Team Selection
- Team Profile
- Team Details
- Team Gallery

---

#### Your changes were saved successfully.

### Team Name: Sample Team

### Team Members

<table>
<thead>
<tr>
<th>Name</th>
<th>Email</th>
</tr>
</thead>
<tbody>
<tr>
<td>DJ</td>
<td><a href="mailto:djanghwaen@hkmultis.com">djanghwaen@hkmultis.com</a></td>
</tr>
</tbody>
</table>

---

### Photos

#### Photo:

---

### Processor Description:
- Intel Core 2 Duo

### Your Interest Level:
- Real Intel Core 2 Duo
Dear Cob,

Our revenues are getting hit by the knock-off clones from our low-cost rival, Boatorola Corp. The business strategist from Underson Consulting says our only hope for survival is to patent every aspect of our product. However, since many of the components are standard, there is nothing to patent. We need to redesign each piece so that it explores non-standard parts of the design space, which being new, will give our patent attorneys fodder for patent filings. Then when Boatorola copies us, we litigate for patent infringement!
A Letter from Jake 2/2

We’re hoping that you can pull together a team that can provide us with a new instruction set architecture that looks novel enough that we will have ample material to patent. We suggest as a start that you use a 17-bit fixed instruction size and 34-bit word size; that combination may help you explore a new design space. Furthermore, Boatorola has access to the MIPS and SPARC patent portfolio, so try not to be too similar to them. Of course, our products still need to be competitive, so the ISA should be general purpose and reasonably efficient. Also, since the fuse is relatively short on this, you’ll need to keep it simple. Your compensation will be based on the correctness, novelty and efficiency of the end product. But most importantly, you must deliver a preliminary ISA by May 2 (1 week), the validated ISA by May 9 (1 weeks), and a final FPGA prototype by May 30 (3 weeks).

Thanks in advance,
Jake
ISA Goals

• General Purpose
  – Is it able to execute provided benchmark programs?

• Novelty
  – How much is it different from MIPS or SPARC?

• Performance
  – How long does it take to run benchmark programs on your processor?
ISA Requirements

• Size
  – 17 bit instruction
  – 34 bit data

• Make it be general purpose
  – Support function calls and recursions
  – Should be able to execute provided benchmark programs

• Include two IO instructions
  – in channel[3:0]
  – out channel[3:0]
IO Instructions

• Channels
  – 16 channels available for I/O
  – channel 0xF is designated for debugging

• in channel [3:0]
  – read 34bit data from the specified port
  – use blocking semantics
    • wait until data comes in

• out channel[3:0]
  – write 34bit data from the specified port
  – use blocking semantics
    • wait until the output port buffer is available
Preliminary Benchmarks for Lab 2a

- Merge Sort
  - Support recursion

- Program Loader
  - Copy instructions from data memory to instruction memory

- SuperGarbage
  - Support various operations
Lab 2a

• Due 5/2

• Make your ISA manual
  – instruction name, opcode, and RTL description
  – (not RTL verilog, but RTL)

• Write assembly programs for given benchmarks using your own ISA

• Webpage will be posted soon
Any Questions?

• Lab 2

• Lab 1c