Group Meetings

- Groups of 2-3 people
- Groups meet individually with Arvind, TA, Mentor weekly during assigned slot sometime 3-4:30pm Monday or Wednesday in Arvind’s office
- Reports due day before the meeting
- See schedule posted on website for which reports are due which weeks
Project Considerations

- Performance should matter
  - Otherwise you could just write software
- Must be testable
  - How will you verify your design works?
  - Is there accessible reference C code?
Past Projects

Posted on Website under Projects

2009

- Essence and Echo Hash Algorithms
- Whirlpool and FSB Hash Algorithms
- Gigabit Ethernet TCP Regular Expression Matcher
- Vigilance Multicore Processor

2008

- A Hardware Accelerator Store for Low Power Processors
- Ogg Vorbis Decoder Implementation in Bluespec
- JPEG Decoder
More Past Projects

2007

- A Load-Balanced Graphics Pipeline on Heterogeneous Multicore
- Processor Parameterization for Redistribution of Resource Among Pipeline Stages
- Reed-Solomon Decoder
- Re-Order Buffer for the SMIPSv2 Processor
- Difference of Gaussian Image Pyramids for Hardware-Accelerated SIFT
- GZIP
- Vorbis Decoder
- H.264 Decoder Area and Power Optimizations
- HASIM Timing Model for Smips
Even More Past Projects

2006

- Runahead Processor
- SMIPS Multimedia Extensions
- H.264 Encoder Design
- Partial H.264 Decoder
- S3DGProc: A Simple 3D Graphics Processor
- In-Order Superscalar SMIPS Processor
Multithreaded SMIPS

- Implement an SMIPS processor that interleaves the execution of multiple threads in hardware
- You can experiment with cores support 2-8 threads
- Implement fine-grain, coarse-grain, or simultaneous multithreading.

(http://www.realworldtech.com/page.cfm?articleid=RWT122600000000)
Cache Hierarchy Exploration with SMIPS

- Experiment with different types and levels of caching
- Try different: associativity, inclusivity, replacement policies

A Typical Memory Hierarchy c.2006
Other SMIPS Project Ideas

Out-of-order superscalar SMIPS Processor
For example, using Tomasulo’s algorithm for out-of-order execution with register renaming through reservation stations.
Other SMIPS Project Ideas

**Out-of-order superscalar SMIPS Processor**
For example, using Tomasulo’s algorithm for out-of-order execution with register renaming through reservation stations.

**SMIPS DSP Extensions**
Use the SMIPS coprocessor interface to add a DSP accelerator to a basic SMIPS processor. You will need to extend the SMIPS ISA and write appropriate test/benchmark codes. Compare performance against baseline SMIPS.
Other SMIPS Project Ideas

Prefetching
Try implementing a hardware prefetcher to bring values into cache before the processor requests them. Stream buffers are one technique which predicts the stride of regular accesses.

Compressed Memory Systems
Implement a compressed memory system, where cache lines are uncompressed when loaded into cache, and compressed again when evicted to main memory.
Other SMIPS Project Ideas

Prefetching
Try implementing a hardware prefetcher to bring values into cache before the processor requests them. Stream buffers are one technique which predicts the stride of regular accesses.

Compressed Memory Systems
Implement a compressed memory system, where cache lines are uncompressed when loaded into cache, and compressed again when evicted to main memory.
Modeling On Chip Networks

- Experiment with virtual channels, arbitration in 2D Mesh network.
- Processor elements could be: SMIPS, Special Processors, or just stubs

Resources = Cores = Processing Elements (P.E.)
Longest Prefix Match

Sparse tree representation

<table>
<thead>
<tr>
<th>IP address</th>
<th>Result</th>
<th>M Ref</th>
</tr>
</thead>
<tbody>
<tr>
<td>7.13.7.3</td>
<td>F</td>
<td>2</td>
</tr>
<tr>
<td>10.18.201.5</td>
<td>F</td>
<td>3</td>
</tr>
<tr>
<td>7.14.7.2</td>
<td></td>
<td></td>
</tr>
<tr>
<td>5.13.7.2</td>
<td>E</td>
<td>1</td>
</tr>
<tr>
<td>10.18.200.7</td>
<td>C</td>
<td>4</td>
</tr>
</tbody>
</table>

In this lecture:
Level 1: 16 bits
Level 2: 8 bits
Level 3: 8 bits
⇒ 1 to 3 memory accesses

February 22, 2010
http://csg.csail.mit.edu/6.375
L06-10
Generalized Sudoku Solver

Design Contest for 2009 International Conference on Field-Programmable Technology
(http://fpt09.cse.unsw.edu.au/competition.html)
SAT Solver

- Given Boolean formula in conjunctive normal form, figure out if any assignment of variables makes the formula true
- Satisfiability is NP-Complete

\[(A \lor B) \land (\neg B \lor C \lor \neg D) \land (D \lor \neg E)\]