Judith Providence
Computer Architecture
CS 654
Outline

- Background/Motivation
- Multi-processors
- Larrabee Architecture
- Performance studies
- Evaluation
- Conclusion
Motivation: Trends Towards Many-core Processors

- Power
- Growth in HPC
- Decrease performance in uniprocessors
  - Limits on Instruction-Level Parallelism
    - Register renaming
    - Branch prediction
    - Jump prediction
    - Memory address Alias Analysis
    - Perfect caches
Larrabee: GPU or CPU?

- GPU
- PCI bus
- Only a minimum amount of memory available
- Only single-precision floating point performance

- Larrabee CPU
- It supports 4 threads
- Efficient inter-block communication
  - Ring network for full inter-processor communication
- Each Larrabee core is a complete x86 core that supports
  - Virtual memory and page swapping
  - Fully coherent caches at all levels
Larrabee:CPU

- Larrabee a in-order many-core x86 CPU
- Intel president in 2005 stated: We are dedicating all of our future product development to multi-core designs.
- Multi-core processors vs. many-core processors
- GPU-like capabilities
Motivation for an in-order CPU

- Comparison between a modern out-of-order CPU, the Intel Core2Duo processor, and an in-order test CPU design based on the Pentium processor with a 16-wide VPUs

<table>
<thead>
<tr>
<th></th>
<th>2 out-of-order</th>
<th>10 in-order</th>
</tr>
</thead>
<tbody>
<tr>
<td>#CPU cores:</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Instruction issue:</td>
<td>4 per clock</td>
<td>2 per clock</td>
</tr>
<tr>
<td>VPU per core:</td>
<td>4-wide SSE</td>
<td>16-wide</td>
</tr>
<tr>
<td>L2 cache size:</td>
<td>4 MB</td>
<td>4 MB</td>
</tr>
<tr>
<td>Single-stream:</td>
<td>4 per clock</td>
<td>2 per clock</td>
</tr>
<tr>
<td>Vector throughput:</td>
<td>8 per clock</td>
<td>160 per clock</td>
</tr>
</tbody>
</table>
Multi-processors

- **Inter-processor Communication**
  - Inter-processor Ring Network

- **Computation**
  - SIMD vector processing unit, mask register

- **Shared Memory**
  - Coherent cached memory hierarchy, MIMD Model

- **Synchronization Mechanisms**
  - Semaphores, Software locks
Larrabee Architecture

Figure 1: Schematic of the Larrabee many-core architecture: The number of CPU cores and the number and type of co-processors and I/O blocks are implementation-dependent, as are the positions of the CPU and non-CPU blocks on the chip.
Core Design of Larrabee

Larrabee CPU core and associated system blocks: the CPU is derived from the Pentium processor in-order design, plus 64-bit instructions, multi-threading and a wide VPU. Each core has fast access to its 256KB local subset of a coherent 2nd level cache. L1 cache sizes are 32KB for Icache and 32KB for Dcache. Ring network accesses pass through the L2 cache for coherency.
Inter-processor Ring Network

- Bi-directional
- Routing decisions made before messages are placed into the network
- Checks for data sharing
- Provides a path for the L2 cache to access memory
- Allows Fixed Function Logic agents to be accessed by the CPU cores
- Scaling to more than 16 cores
Wide Vector Processing Unit

- SIMD
- 16 lanes
- Executes integer and floating point instructions
- Scatter gather supports a maximum of 16 elements
Fixed Function Logic Unit

- Used for Graphical tasks
- Larrabee uses software in place of a fixed functional unit for some graphical tasks
- Cores pass commands to the texture unit through the L2 cache
- Texture filter logic
  - would be 12x to 40x longer in software
Advanced Applications

- Larrabee supports irregular data structures
- An efficient scatter-gather support for irregular data structures
- The SIMD vector processing unit can be programmed
- Intel’s auto-vectorization computer technology
Performance Study

- Spectral methods/Dense Linear algebra
- Data is in the frequency domain
- High Performance Kernel-3D-FFT
- Data that are dense matrices or vectors -BLAS-3
High Performance Computing Kernels

- Simulation results are based on Stanford’s PhysBam
  
  http://physbam.standford.edu/~fedkiw

- Amdahl’s Law: Speedup\text{ maximum} = \frac{1}{(1-\text{fraction enhanced})}
Evaluation of Larrabee for parallel applications

**con**
- Memory contention
- Lack of error correcting code (ECC) memory,
- Graphic double data rate
- Shortage of double precision floating point capability

**pro**
- Load balancing is accomplished by moving processes
- Supports irregular data structures
Conclusion-Relevance of Larrabee for the Future

- Amdahl’s Law - Limitations in parallelism make it difficult to achieve good speedup
- 1965 - Moore’s Law states that the number of transistors on a chip will double about every two years
- Need a Moore’s Law to handle software
- Solution: the establishment of academic communities