Evaluation of Existing Architectures in IRAM Systems

Christoforos Kozyrakis, Ngeci Bowman, Neal Cardwell, Cynthia Romer and Helen Wang

Computer Science Division
University of California at Berkeley

http://iram.cs.berkeley.edu

Workshop on “Mixing Logic and DRAM”, ISCA ’97
Motivation

• Intelligent RAM promises:
  – high memory bandwidth (100x)
  – low memory latency (0.1x)
  – high energy efficiency (4x)
  – higher system integration

• Which microprocessor architecture can turn these advantages into significant application performance benefits?
Evolutionary IRAM Approach

• Use an **existing** processor architecture:
  simple RISC micro, superscalar or out-of-order execution organization

• Advantages:
  – Good knowledge of how to design and implement them
  – Performance trade-offs are well understood
  – “Out of the box” solutions both for system software and applications - software compatibility
  – Higher performance by tuning programs and compilers to new memory hierarchy characteristics

• **This work**: evaluate potential performance benefits of this approach
Outline

• IRAM Architectural Considerations
• Evaluation through Measurements and Extrapolations
• Evaluation through Simulation
• Conclusions
IRAM Architectural Considerations

- IRAM systems using existing DRAM technology:
  - 256Mbit DRAM 0.25μm CMOS process
  - 1/4 of die area for microprocessor
  - Up to 24MBytes of on-chip DRAM
- Memory access latency can be as low as 21ns
- Logic speed potentially 10% to 50% slower compared to conventional processors for initial implementations
- No level 2 cache necessary since on-chip DRAM can have comparable latency
- Memory bus as wide as cache line
Method I: Measurements and Extrapolation

- **Execution time analysis** of a simple (Alpha 21064) and a complex architecture (Pentium Pro) to predict performance of similar IRAM implementations.
- Used hardware counters for execution time measurements.
- Benchmarks: SPEC95Int, Mpeg_encode, Linpack1000, Sort.
- IRAM implementations: same architectures with 24MBytes of on-chip DRAM but no L2 caches; all benchmarks fit completely in on-chip memory.
- IRAM execution time model:

\[
ET = \frac{\text{computation}_\text{time}}{\text{clock}_\text{speedup}} + \frac{L1\text{miss}_\text{count}*\text{memory}_\text{access}_\text{time}}{\text{memory}_\text{access}_\text{speedup}}
\]
Method I: Processors Characteristics

<table>
<thead>
<tr>
<th></th>
<th>Alpha 21064</th>
<th>Pentium Pro</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pipeline</td>
<td>in-order</td>
<td>out-of-order</td>
</tr>
<tr>
<td>CPU Frequency</td>
<td>133 MHz</td>
<td>200MHz</td>
</tr>
<tr>
<td>Issue Rate</td>
<td>2-way</td>
<td>3-way</td>
</tr>
<tr>
<td>L1 Configuration</td>
<td>8KB I + 8KB D</td>
<td>8KB I + 8KB D</td>
</tr>
<tr>
<td>L1 Associativity</td>
<td>Direct map</td>
<td>4-way</td>
</tr>
<tr>
<td>L1 Access Time</td>
<td>22.5ns</td>
<td>15ns</td>
</tr>
<tr>
<td>L2 Configuration</td>
<td>512KB</td>
<td>256KB</td>
</tr>
<tr>
<td>L2 Associativity</td>
<td>Direct map</td>
<td>4-way</td>
</tr>
<tr>
<td>L2 Type</td>
<td>Off-chip SRAM</td>
<td>Off-chip SRAM</td>
</tr>
<tr>
<td>L2 Access Time</td>
<td>37.5ns</td>
<td>20ns</td>
</tr>
<tr>
<td>Memory</td>
<td>64MB EDO DRAM</td>
<td>64MB EDO DRAM</td>
</tr>
<tr>
<td>Total Latency</td>
<td>180ns</td>
<td>220ns</td>
</tr>
</tbody>
</table>
Execution Time Analysis of Conventional Systems

- Linpack1000 and Sort spend up to 50% of execution time in main memory
- SPEC and Mpeg_encode are CPU bound

C. Kozyrakis, Evaluation of Existing Architectures in IRAM Systems
Method I: Results

- Equal clock speeds assumed for conventional and IRAM systems
- Maximum IRAM speedup compared to conventional:
  - Less than 2 for memory bound applications
  - 1.1 for CPU bound applications

*C. Kozyrakis, Evaluation of Existing Architectures in IRAM Systems*
Method II: Detailed System Simulations

• Used SimOS to simulate simple MIPS R4000-based IRAM and conventional architectures

• **Equal die size** comparison:
  – Area for on-chip DRAM in IRAM systems same as area for level 2 cache in conventional system

• **Wide memory bus** for IRAM systems

• Main simulation parameters:
  – On-chip DRAM access latency
  – Logic speed (CPU frequency)

• Benchmarks: SPEC95Int (compress, li, ijpeg, perl, gcc), SPEC95Fp (tomcatv, su2cor, wave5), Linpack1000
# Simulated Models

<table>
<thead>
<tr>
<th></th>
<th>IRAM</th>
<th>Conventional</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Pipeline</strong></td>
<td>Simple in-order</td>
<td>Simple in-order</td>
</tr>
<tr>
<td><strong>CPU Frequency</strong></td>
<td>333 or 500 MHz</td>
<td>500MHz</td>
</tr>
<tr>
<td><strong>Technology</strong></td>
<td>0.25μm DRAM</td>
<td>0.25μm logic</td>
</tr>
<tr>
<td><strong>L1 Configuration</strong></td>
<td>64KB I + 64KB D</td>
<td>64KB I + 64KB D</td>
</tr>
<tr>
<td><strong>L1 Associativity</strong></td>
<td>2-way</td>
<td>2-way</td>
</tr>
<tr>
<td><strong>L1 Block Size</strong></td>
<td>128B</td>
<td>64B I + 32B D</td>
</tr>
<tr>
<td><strong>L1 Type</strong></td>
<td>On-chip SRAM</td>
<td>On-chip SRAM</td>
</tr>
<tr>
<td><strong>L1 Access Time</strong></td>
<td>1 CPU cycle</td>
<td>1 CPU cycle</td>
</tr>
<tr>
<td><strong>L2 Configuration</strong></td>
<td>-</td>
<td>2MB unified</td>
</tr>
<tr>
<td><strong>L2 Associativity</strong></td>
<td>-</td>
<td>2-way</td>
</tr>
<tr>
<td><strong>L2 Block Size</strong></td>
<td>-</td>
<td>128B</td>
</tr>
<tr>
<td><strong>L2 Type</strong></td>
<td>-</td>
<td>On-chip SRAM</td>
</tr>
<tr>
<td><strong>L2 Access Time</strong></td>
<td>-</td>
<td>12 CPU cycles</td>
</tr>
<tr>
<td><strong>Memory Configuration</strong></td>
<td>24MB DRAM on-chip</td>
<td>24MB 166MHz SDRAM off-chip</td>
</tr>
<tr>
<td><strong>Memory Bus Width</strong></td>
<td>128B</td>
<td>16B</td>
</tr>
<tr>
<td><strong>Total Latency</strong></td>
<td>21 or 33ns</td>
<td>116ns</td>
</tr>
</tbody>
</table>
Method II: Results

- Execution times normalized to basic IRAM model (333MHz, 33ns memory latency)
- IRAM models up to 40% faster than conventional
Conclusions

• IRAM systems with existing processors provide only moderate performance benefits
• High bandwidth/low latency used to speed up memory accesses but not computation
• Reason: existing architectures developed under the assumption of a low bandwidth memory system
• Still attractive for portable/embedded domain
  – up to 4 times more energy efficient
  – higher system integration
Towards a Revolutionary Approach

• To provide significant performance benefits IRAM systems need microprocessor architectures that turn memory bandwidth into application performance

• Candidates:
  – Vector microprocessor
  – Multithreading architectures
  – Multiprocessor on a chip
  – Some hybrid combination?
  – Some new idea?