A Coherent Hybrid SRAM and STT-RAM L1 Cache Architecture for Shared Memory Multicores

Jianxing Wang, Yenni Tim Zhong-Liang Ong and Weng-Fai Wong  
School of Computing  
National University of Singapore

Zhenyu Sun and Hai (Helen) Li  
Swanson School of Engineering  
University of Pittsburgh
Outline

• STT-RAM Basics
  • Cell structure and advantages
  • Challenges and motivations
• Hybrid L1 Cache Architecture
  • Naïve solution
  • The MESI protocol
  • Block transfer mechanisms
• Evaluation
  • Performance and energy
  • STT-RAM endurance
• Conclusion
STT-RAM Basics

- Magnetic Tunnel Junction (MTJ)
  - Two ferromagnetic layers separated by a barrier

![Diagram of MTJ and associated circuit elements]
STT-RAM Basics (cont.)

• Advantages
  • Non-volatile, near zero leakage energy
  • As fast as SRAM (read)
  • As dense as DRAM
  • Multi-level cell capability (stacking MTJs)
  • CMOS-compatible
  • Universal memory
Motivations of Hybrid Cache

- Expensive write operation of STT-RAM
  - High latency (10ns+)
  - High energy
  - Compensated by relaxed non-volatility \([\text{Smullen et al. 11}]\)
    - Refresh
  - Endurance
- Intense writes in L1
  - *bodytrack*: \(\frac{\text{L1(s)}}{\text{L2}} = \sim 29!\)
  - Additional synchronous operations under multi-core environment
Proposed Hybrid Cache Hierarchy
Cache Block Management

• Naïve solution
  • Based on temporal locality
  • Simple but not good enough
    • > 3% IPC degradation
The MIESI Coherent Protocol

- Developed by University of Illinois
  - Illinois MIESI
- For each cache block
  - M (modified) state – data dirty, exclusive copy
  - E (exclusive) state – data clean, exclusive copy
  - S (shared) state – data clean, multiple copies
  - I (invalid) state
- Common event bus
  - Local (processor) read/write
  - Remote (snoop / bus) read/write
Cache Block Management (cont.)

- Immediate transfer policy (IT)
  - Place dirty data (M state) block in SRAM
  - Place clean data (E/S state) block in STT-RAM
  - Transfer cache block when coherent state changes
  - DO NOT need extra information (built-in by MESI)
Immediate Transfer Policy (IT)

SRAM

I M

Write Miss

Remote Read

Local Write

STT-RAM

S E I

Read Miss
Cache Block Management (cont.)

• Delayed transfer policy (DT)
  • IT could be too aggressive
    • Coherent state “ping-pong” between M and S
  • Relax state restriction
  • Consider request history in prediction
  • Extra information required
Delayed Transfer Policy (DT)

1st Remote Read

SRAM

2nd Remote Read

STT-RAM

2nd Local Write (consecutive)

1st Local Write
Evaluation

- PARSEC on MARSSx86 \([Patel et al. 11]\)
  - IPC (Instruction Per Cycle)
- NVSim \([Dong et al. 12]\)
  - Latency, area and energy numbers (32nm)
- Configuration
  - Quadcore machine with two-level cache hierarchy
  - Relaxed STT-RAM’s non-volatility with a \(26.5\mu s\) retention period \([Sun et al. 11]\)
  - Various cache size combinations within the baseline area budget (64KB SRAM)
Normalized Energy (IT policy)

![Normalized Energy Chart]

- 4KB SRAM + 64KB STT-RAM
- 8KB SRAM + 64KB STT-RAM
- 16KB SRAM + 64KB STT-RAM
- 4KB SRAM + 128KB STT-RAM
Comparison of Transfer Policies

![Comparison of Transfer Policies](chart)

- Pure 64KB STT-RAM
- 4KB SRAM + 64KB STT-RAM
- 8KB SRAM + 64KB STT-RAM
- 16KB SRAM + 64KB STT-RAM
- 4KB SRAM + 128KB STT-RAM

<table>
<thead>
<tr>
<th>IPC (%)</th>
<th>Energy (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Naïve</td>
<td>IT</td>
</tr>
</tbody>
</table>

- IPC: Instruction Per Cycle (%)
- Energy (%)

- Naïve: Simplest approach
- IT: Incremental Transfer
- DT: Dual Transfer
- Energy: Complex Energy Optimization
STT-RAM Endurance

- Lifespan programming cycles
  - SRAM and DRAM: $10^{16}$
  - STT-RAM prediction [Tabrizi 07]: $10^{15}$
  - STT-RAM reported [Diao et al. 07]: $10^{13}$
  - SLC NAND flash: $10^5$

- Writes in L1 cache
  - High intensity
  - Non-even distributed
    - *bodytrack*: ~35% writes on one cache partition
    - *facesim*: ~50% writes on the same cache partition, ~15% on the same block!
STT-RAM Endurance (cont.)

- *facesim*

<table>
<thead>
<tr>
<th></th>
<th>Perfect distributed</th>
<th>Worst Partition</th>
<th>Worst Block</th>
</tr>
</thead>
<tbody>
<tr>
<td>Baseline SRAM</td>
<td>1,300+ years</td>
<td>300+ years</td>
<td>&lt; 360 hrs</td>
</tr>
<tr>
<td>Baseline STT-RAM</td>
<td>1.3 years</td>
<td>0.3 years</td>
<td>&lt; 22 mins</td>
</tr>
<tr>
<td>Hybrid Naïve</td>
<td>3.5 years</td>
<td>1.0 year</td>
<td>0.9 hr</td>
</tr>
<tr>
<td>Hybrid IT</td>
<td>41.2 years</td>
<td>6.9 years</td>
<td>51.6 hrs</td>
</tr>
<tr>
<td>Hybrid DT</td>
<td>32.9 years</td>
<td>7.0 years</td>
<td>54.3 hrs</td>
</tr>
</tbody>
</table>

150x lifespan increases for the worst block!
Conclusion

• Deploy STT-RAM as L1 cache
  • Expensive write (latency, energy and endurance)
• Architecture solution: hybrid cache
  • “big.LITTLE” model
• MESI-based Hybrid L1 Cache Architecture
  • Small SRAM partition + large STT-RAM partition
  • Using built-in information from coherent protocol
  • Performance maintained with less energy, and extended lifespan
THANK YOU!

Q & A