Multi-Core Memory Hierarchies

Lecture 14:
Heterogeneous Memory System

Tong Dong

http://mprc.pku.edu.cn/~tongdong/MMH
Scratchpad Memory

- On-Chip Memory in Embedded processor, DSP, GPU, Network processor.
- User Address Space
- Direct Programming or Compiler Guided
- IBM Cell Processor
- SPM for Domain Specific Accelerator

## DRAM Technologies

<table>
<thead>
<tr>
<th></th>
<th>DDR4-2400</th>
<th>DDR3-1866</th>
<th>RLDRAM3</th>
<th>LPDRAM2</th>
<th>GDDR5</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Data Rate</strong></td>
<td>2400 MT/s</td>
<td>1866 MT/s</td>
<td>1600 MT/s</td>
<td>800 MT/s</td>
<td>8.0 Gb/s</td>
</tr>
<tr>
<td><strong>tRC=tRP+tRAS</strong></td>
<td>45.35ns</td>
<td>47.91ns</td>
<td><strong>12ns</strong></td>
<td>60ns</td>
<td>-</td>
</tr>
<tr>
<td><strong>tRCD-tRP-tCL</strong></td>
<td>16-16-16</td>
<td>13-13-13</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td><strong>tRCD</strong></td>
<td>13.32ns</td>
<td>13.91ns</td>
<td>-</td>
<td>18ns</td>
<td>-</td>
</tr>
<tr>
<td><strong>tRP</strong></td>
<td>13.32ns</td>
<td>13.91ns</td>
<td>-</td>
<td>18ns</td>
<td>-</td>
</tr>
<tr>
<td><strong>tCL (tCAS)</strong></td>
<td>13.32ns</td>
<td>13.91ns</td>
<td>10ns</td>
<td>18ns</td>
<td>-</td>
</tr>
<tr>
<td><strong>tRAS</strong></td>
<td>32ns</td>
<td>34ns</td>
<td>-</td>
<td>42ns</td>
<td>-</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th><strong>Voltage (V)</strong></th>
<th>SDRAM-133</th>
<th>DDR-400</th>
<th>DDR2-800</th>
<th>DDR3-800</th>
<th>DDR3-1066</th>
<th>DDR3-1333</th>
<th>DDR3-1600</th>
</tr>
</thead>
<tbody>
<tr>
<td>Max Capacity (per chip)</td>
<td>512Mb</td>
<td>1Gb</td>
<td>2Gb</td>
<td>4Gb</td>
<td>4Gb</td>
<td>2Gb</td>
<td>1Gb</td>
</tr>
<tr>
<td>Price (1GB UDIMM non-ECC)</td>
<td>$49</td>
<td>$49</td>
<td>$28</td>
<td>$75</td>
<td>$99</td>
<td>$129</td>
<td>$175</td>
</tr>
<tr>
<td>Bus Freq. (MHz)</td>
<td>133</td>
<td>200</td>
<td>400</td>
<td>6400</td>
<td>533</td>
<td>8533</td>
<td>667</td>
</tr>
<tr>
<td>BW (MB/s/channel)</td>
<td>1066</td>
<td>3200</td>
<td>6400</td>
<td>2.5</td>
<td>1.87</td>
<td>2.5</td>
<td>1.5</td>
</tr>
<tr>
<td>tCK (ns)</td>
<td>7.5</td>
<td>5</td>
<td>2.5</td>
<td>15</td>
<td>15</td>
<td>15</td>
<td>13.5</td>
</tr>
<tr>
<td>Timing(tCL) (memory cycle)</td>
<td>3</td>
<td>3</td>
<td>6</td>
<td>6</td>
<td>8</td>
<td>9</td>
<td>11</td>
</tr>
<tr>
<td>Burst Length (memory cycle)</td>
<td>8</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>4</td>
</tr>
<tr>
<td>Tpre,Tact,Tcol (ns)</td>
<td>22.5</td>
<td>15</td>
<td>15</td>
<td>15</td>
<td>15</td>
<td>15</td>
<td>13.5</td>
</tr>
<tr>
<td>T_{bl} (ns)</td>
<td>60</td>
<td>20</td>
<td>10</td>
<td>10</td>
<td>7.5</td>
<td>6</td>
<td>5</td>
</tr>
</tbody>
</table>

Table 1: Comparison of DRAM Technologies.
Heterogeneous DRAM Memory

- Observation: the first word in a cache line is typically most critical and benefits from being placed in a low-latency channel/DIMM.
- The pin-bandwidth of RL−DRAM3 is comparable to DDR3, its core latencies are extremely small, due to
  - the use of many small arrays.
  - the use of SRAM−style addressing within a single WRITE and READ command.
- Critical-word-first (CWF)

(a) A baseline DDR3 system, with a 72-bit wide Data + ECC bus and a 23-bit wide address/control bus. The processor has four instances of this configuration.

(b) A heterogeneous Static-CWF memory system where word-0 is placed in a single-rank RLDRAM DIMM and other words are placed in a single-rank LPDRAM DIMM. The processor has four instances of this configuration.

(c) A low-cost heterogeneous Static-CWF memory system where the critical words of all 4 channels are aggregated on a 16-chip RLDRAM DIMM and other words are placed in four single-rank LPDRAM DIMMs.

3D-stacked Package Tecnologies

3D-stacked DRAM

2.5D-stacked DRAM
Hybrid DRAM Main Memory

Fig. 3. Illustration of the proposed heterogeneity-aware on-chip DRAM memory controller with optional migration controller.

Fig. 9. The improved version of N-1 Mode with live migration support: The additional F bit indicates that the corresponding on-package slot is under data movement; the associated bit map indicates which sub-block is ready for accessing.
- 3D-Stack DRAM + off-chip DRAM hybrid memory system
- PoM (part of memory) dynamically remap regions of memory based on their access patterns and expected performance benefits.
- Translation from Page Table Physical Address (PTPA) to DRAM Physical Address (DPA)
  - Segment Remapping Table
  - Segment Remapping Cache

Figure 3. Overview of the PoM architecture.
The Implementation Complexity of 3D DRAM Cache
- Tag, cache controller, inflexible, verification effort

- AMD APU
- NUMA
- Address Mapping
- Memory Controller
- Page Selection
  - History-based
  - First-touch
  - Hot Page Policy
  - First-touch hot-page (FTHP)
  - Hotness Threshold
  - Feedback-directed HMA
  - Low-cost Hot Page Detection

MemPod, a memory management mechanism for flat address space hybrid memories.

- MemPod monitors memory activity and periodically migrates the most frequently accessed memory pages to the faster memory.
  - Majority Element Algorithm (MEA)

MemPod’s partitioned architectural organization allows for efficient scaling with memory system capabilities.

---

**Algorithm 1: Majority Element Algorithm**

**Input:** X: Set of N elements  
**Input:** K: Number of elements to output  
**Data:** T: Map structure with K entries  
**Result:** Set of K majority elements

```
Initialization: T ← ∅

foreach i ∈ X do
  if i ∈ T then
    T[i] ← T[i] + 1;
  else if |T| < K - 1 then
    T[i] ← 1;
  else
    forall j ∈ T do
      T[j] ← T[j] - 1;
      if T[j] == 0 then T ← T \{j};
  end
end
```

---

**Figure 4:** MemPod high-level architecture

---

**Figure 5:** Major architectural Pod elements
Traditional Memory Hierarchies

- **On-chip memory** (SRAM): 1~30 cycles
- **Main memory** (DRAM): 100~300 cycles
- **Solid State Disk** (Flash Memory): 25000~2000000 cycles
- **HDD**: >5000000 cycles

Need more on-chip memory

Static power is dominating

Charge memory: scaling challenges
Emerging Memory Technologies

- Magnetic RAM (MRAM)
  - EverSpin (130nm, up to 16Mb)
- Spin-Torque-Transfer RAM (STTRAM)
  - Grandis (54nm, acquired by Samsung)
- Phase-Change RAM (PCRAM)
  - Samsung (20nm, diode, up to 8Gb)
  - Intel & Micron (3Dxpoint)
- Resistive RAM (RRAM)
  - Panasonic (180nm process, 4-layer xpoint)
  - Unity Semi (64MB, acquired by Rambus 2012.2)
- Racetrack Memory (RM)
  - Osaka Institute of Technology (1Mb chip)

They have some common features
Emerging Memory Technologies

(a) PCM

(b) STT-RAM

(c) ReRAM

(d) DWM/RM
PCM

Temperature

\[ T_m \]

\[ T_x \]

Time

RESET

SET
STT-RAM

(a) Inplane Magnetic Tunnel Junction (IMTJ)

(b) Perpendicular Magnetic Tunnel Junction (PMTJ)
ReRAM

- "0": High Resistance State (HRS)
- "1": Low Resistance State (LRS)
- HRS→LRS: SET
- LRS→HRS: RESET

Diagram:
- Voltage vs. Current (A)
- Voltage (V) scale: -3 to 2
- Current (A) scale: 10^-10 to 10^-3
- Set compliance = 100 µA
- Reset stop voltage = -3.3 V
Racetrack Memory/Domain Wall Memory

- GND, BL, SL, BLB, Nanowire (spintronic material)
- Domain
- Domain Wall
- RWL
- WWL
- R/W Port
- Overhead region
- Shift current
- Magnetization directions
A graphene-like material named MoS2, which produce a memristor effect in atomically thin insulating materials

——Nano Letters, Vol.1, 2018
Interesting Features

- **Pros:**
  - Non-volatility
  - High storage density
  - Low leakage
  - Immunity to particle-strike soft errors

- **Cons:**
  - Asymmetric read-write operations
    - Long write latency, High write energy
  - Limited lifetime
  - Unique vulnerability
  - Extra operations (e.g. shift for RM)
Heterogeneous Memory Hierarchies

- On-chip memory: 1~30 cycles
- Main memory: 100~300 cycles
- Solid State Disk: 25000~2000000 cycles
- HDD: >5000000 cycles

Our focus

- Where to use these memories in the memory hierarchy?
- Understand basics of these NVMs
- Circuit level modeling
- Directly Replacement and Optimization
- Hybrid Design with SRAM and DRAM
- Software-Hardware Cooperative Design
Comparison of Emerging Memories

- **STT-MRAM**: potential replacement of SRAM?
- **PCRAM**: potential replacement of DRAM/Nor Flash?
- **RRAM**: potential replacement of NAND Flash?
- **RM**: universal memory?

<table>
<thead>
<tr>
<th></th>
<th>SRAM</th>
<th>DRAM</th>
<th>FLASH</th>
<th>STT-RAM</th>
<th>PCRAM</th>
<th>RRAM</th>
<th>RM</th>
</tr>
</thead>
<tbody>
<tr>
<td>cell area (F²)</td>
<td>50-200</td>
<td>6</td>
<td>&lt;4 if 3D</td>
<td>6-20</td>
<td>4-10</td>
<td>&lt;4 if 3D</td>
<td>&lt;4</td>
</tr>
<tr>
<td>multi-bit</td>
<td>1</td>
<td>1</td>
<td>3</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td>scalability</td>
<td>&lt;20nm</td>
<td>&lt;20nm</td>
<td>&lt;20nm</td>
<td>&lt;10nm</td>
<td>&lt;20nm</td>
<td>&lt;10nm</td>
<td>&lt;10nm</td>
</tr>
<tr>
<td>voltage</td>
<td>&lt;1V</td>
<td>&lt;1V</td>
<td>&gt;20V</td>
<td>&lt;2V</td>
<td>&lt;3V</td>
<td>&lt;3V</td>
<td>&lt;2V</td>
</tr>
<tr>
<td>read speed</td>
<td>&lt;1ns</td>
<td>~10ns</td>
<td>~10µs</td>
<td>~ns</td>
<td>~50ns</td>
<td>&lt;10ns</td>
<td>~ns</td>
</tr>
<tr>
<td>energy/bit</td>
<td>~fJ</td>
<td>~pJ</td>
<td>~nJ</td>
<td>~0.1pJ</td>
<td>~10pJ</td>
<td>~0.1pJ</td>
<td>~0.1pJ</td>
</tr>
<tr>
<td>standby power</td>
<td>High</td>
<td>High</td>
<td>None</td>
<td>None</td>
<td>None</td>
<td>None</td>
<td>None</td>
</tr>
<tr>
<td>endurance</td>
<td>&gt;1E16</td>
<td>&gt;1E16</td>
<td>&lt;1E5</td>
<td>&gt;1E15</td>
<td>1E8-1E12</td>
<td>1E6-1E12</td>
<td>&gt;1E15</td>
</tr>
<tr>
<td>retention</td>
<td>volatile</td>
<td>~64ms</td>
<td>&gt;10 years</td>
<td>&gt;10 years</td>
<td>&gt;10 years</td>
<td>&gt;10 years</td>
<td>&gt;10 years</td>
</tr>
<tr>
<td>Vulnerability</td>
<td>High</td>
<td>High</td>
<td>Low</td>
<td>Low</td>
<td>Low</td>
<td>Low</td>
<td>Low</td>
</tr>
</tbody>
</table>

- **STT-MRAM**: potential replacement of SRAM?
- **PCRAM**: potential replacement of DRAM/Nor Flash?
- **RRAM**: potential replacement of NAND Flash?
- **RM**: universal memory?
IBM and Samsung achieve breakthrough on flash killer for wearables, mobile devices

MRAM could last virtually forever

By Lucas Mearian
Senior Reporter, Computerworld | JUL 12, 2016 1:18 PM PT
NVM Cache

- Reduce “effective write latency”
  - Reduce write intensities (replacement)
  - Hide long write latency (read-preemptive)

- Hybrid memory
  - Read intensive data -> STTRAM
  - Write intensive data -> SRAM

- Device + Architecture co-optimization
  - Trade retention for performance
  - Separate writing ‘0’ from writing ‘1’

- Reliability
  - Read/write disturbance
  - Shift errors
SRAM Caches vs. MRAM Caches

<table>
<thead>
<tr>
<th></th>
<th>Area (65nm)</th>
<th>3.66mm$^2$ SRAM</th>
<th>3.30mm$^2$ MRAM</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Capacity/Bank</strong></td>
<td>128KB</td>
<td>512KB</td>
<td></td>
</tr>
<tr>
<td><strong>Read latency</strong></td>
<td>2.25ns</td>
<td>2.32ns</td>
<td></td>
</tr>
<tr>
<td><strong>Write latency</strong></td>
<td>2.26ns</td>
<td>11.02ns</td>
<td></td>
</tr>
<tr>
<td><strong>Read energy</strong></td>
<td>0.90nJ</td>
<td>0.86nJ</td>
<td></td>
</tr>
<tr>
<td><strong>Write energy</strong></td>
<td>0.80nJ</td>
<td>5.00nJ</td>
<td></td>
</tr>
</tbody>
</table>

**Pros:** Low leakage power, high density.

**Cons:** Long write latency and large write energy.

*Replace SRAM caches with MRAM?*

---

Direct Replacement

- Good Miss Ratio and Lower static Power Consumption
- But, low performance due to the long write latency and high dynamic power due to the high write energy.

Figure 4. The comparison of L2 caches access miss rates for SRAM L2 cache and MRAM L2 cache that have similar area. Larger capacity of MRAM cache results in smaller cache miss rates.

Figure 5. IPC comparison of SRAM and MRAM L2 caches (Normalized by 2M SNUCA SRAM cache).

Figure 6. Power comparison of SRAM and MRAM L2 caches (Normalized by 2MB SNUCA SRAM cache).
Optimization

- **Read-Preemptive Write Buffer**
  - The read operation always has the higher priority in a competition for the execution right.
  - When a read request is blocked by a write retirement and the write buffer is not full, the read request can trap and stall the write retirement.

- **SRAM-MRAM Hybrid L2 Cache**
  - At write miss, first try to place the data in the SRAM cache ways.
  - Data in MRAM caches should be migrated to SRAM caches if the some cache lines are frequently written to.
Hybrid Cache ISCA 2009

- SRAM, MRAM, PCM, eDRAM
- Inter cache level hybrid cache (LHCA)
- Region based Intra-layer hybrid cache (RHCA)
  - Cache line migration policy: 1-bit sticky for fast region and 2-bit saturating counter for slow region
- 3D-stacking hybrid cache (3DHCA)
- LHCA 7%; RHCA 12%; 3DHCA 18%; 70% power reduction

Figure 5: Overview of hybrid cache design methodology.
Relaxing Non-Volatility STT-RAM Caches

- STT-RAM bottleneck: write latency and energy
- Lower the retention time by reducing the planar area of the cell, thereby reducing the write current
- DRAM style refresh policy to prevent data loss
- RAM-based L1 caches with reduced-retention STT-RAM L2 and L3 caches eliminates performance loss while still reducing the energy-delay product by more than 70%

Figure 2: Benefits of relaxed non-volatility STT-RAM

Figure 16: Impact of STT-RAM refreshing on the energy-delay product

Multi Retention Level STT-RAM Cache

- Lower Level Cache with Mixed High and Low Retention STT-RAM Cells

Figure 5: Dynamic counter-controlled refreshing scheme.

Multi Retention Level STT-RAM Cache

Figure 6: Hybrid lower-level cache migration policy: Flow graph (left). Diagram (right).
Adaptive Placement and Migration Policy

- Three categories of LLC writes
  - Core-write: dirty data evicted from the core cache and written back to the LLC
  - Prefetch-write: write from the LLC replacement caused by a prefetch miss
  - Demand-write: write from the LLC replacement caused by a demand miss

- Access Pattern Predictor
  - Zero-read-range prefetch write blocks: 26%, Bypassing
  - Immediateread-range prefetch write blocks: 56.9%, Burst to SRAM
  - Distant-read-range prefetch-write blocks: 17.5%, migrate to STT-RAM

[Diagram showing the adaptive placement and migration policy]
The key observation is that a significant amount of data written to last-level caches is not actually re-referenced again during the lifetime of the corresponding cache blocks.

DASCA: Dead Write Prediction Assisted STT-RAM Cache Architecture

A novel classification of dead writes
- dead-on-arrival fills
- dead-value fills
- closing writes

Dead Write Predictor
- PC-based Sampling Dead Predictor
- The last-touch PC signature for closing write prediction
LAP: Loop-Block Aware Inclusion Properties for Energy-Efficient Asymmetric Last Level Caches

(a) Non-inclusive | (b) LAP | (c) Exclusive
---|---|---
✓ No redundant clean insertion | ✓ Insert only non-duplicate data | ✓ Redundant clean insertion
➢ Useless duplicate data | ✓ Duplicate only useful clean data | ✓ No duplicate data
➢ Redundant LLC data-fill | ✓ No redundant LLC data-fill | ✓ No redundant LLC data-fill

---

Clean victim

In L3? Y N

Dirty victim

L2 L3 hit

L3

Memory

No-invalidation on L3 hits & loop-block-aware replacement policy

<table>
<thead>
<tr>
<th>Invalidate on hit</th>
<th>L3 fill</th>
<th>Clean Writeback</th>
</tr>
</thead>
<tbody>
<tr>
<td>noni</td>
<td>N</td>
<td>Y</td>
</tr>
<tr>
<td>ex</td>
<td>Y</td>
<td>N</td>
</tr>
<tr>
<td>LAP</td>
<td>N</td>
<td>N</td>
</tr>
</tbody>
</table>
## Comparision

### Table 1: Comparison of various memory technologies for on-die caches.

<table>
<thead>
<tr>
<th></th>
<th>(A) SRAM</th>
<th>(B) STT-RAM</th>
<th>eDRAM</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Cell schematic</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Process</td>
<td>CMOS</td>
<td>CMOS + MTJ</td>
<td>CMOS + Cap</td>
</tr>
<tr>
<td>Cell size ($F^2$)</td>
<td>120 - 200</td>
<td>6 - 50</td>
<td>20 - 50</td>
</tr>
<tr>
<td>Data storage</td>
<td>Latch</td>
<td>Magnetization</td>
<td>Capacitor</td>
</tr>
<tr>
<td>Read time</td>
<td>Short</td>
<td>Short</td>
<td>Short</td>
</tr>
<tr>
<td>Write time</td>
<td>Short</td>
<td>Long</td>
<td>Short</td>
</tr>
<tr>
<td>Read energy</td>
<td>Low</td>
<td>Low</td>
<td>Low</td>
</tr>
<tr>
<td>Write energy</td>
<td>Low</td>
<td>High</td>
<td>Low</td>
</tr>
<tr>
<td>Leakage</td>
<td>High</td>
<td>Low</td>
<td>Low</td>
</tr>
<tr>
<td>Endurance</td>
<td>$10^{16}$</td>
<td>$&gt;10^{15}$</td>
<td>$10^{16}$</td>
</tr>
<tr>
<td>Retention time</td>
<td>-</td>
<td>-</td>
<td>$&lt;100$ us *</td>
</tr>
<tr>
<td>Features</td>
<td>(+) Fast</td>
<td>(+) Non-volatile</td>
<td>(+) Low leakage</td>
</tr>
<tr>
<td></td>
<td>(-) Large area</td>
<td>(+) Potential to scale</td>
<td>(+) Small area</td>
</tr>
<tr>
<td></td>
<td>(-) Leakage</td>
<td>(-) Extra process</td>
<td>(-) Extra process</td>
</tr>
<tr>
<td></td>
<td></td>
<td>(-) Long write time</td>
<td>(-) Destructive read</td>
</tr>
<tr>
<td></td>
<td></td>
<td>(-) High write energy</td>
<td>(-) Refresh</td>
</tr>
<tr>
<td></td>
<td></td>
<td>(-) Poor stability</td>
<td></td>
</tr>
</tbody>
</table>

* 32 nm technology node

Phase-Change Memory (PCM)

Intel/Micron 3D XPoint

INTEL AND MICRON PRODUCE BREAKTHROUGH MEMORY TECHNOLOGY

New Class of Memory Unleashes the Performance of PCs, Data Centers and More

NEWS HIGHLIGHTS

- Intel and Micron begin production on new class of non-volatile memory, creating the first new memory category in more than 25 years.
- New 3D XPoint™ technology brings non-volatile memory speeds up to 1,000 times faster than NAND, the most popular non-volatile memory in the marketplace today.
- The companies invented unique material compounds and a cross point architecture for a memory technology that is 10 times denser than conventional memory.
- New technology makes new innovations possible in applications ranging from machine learning to real-time tracking of diseases and immersive 8K gaming.
PCM Main Memory

- **Performance/Energy Improvement**
  - Hybrid memory
  - Asymmetric write
  - Coding

- **Lifetime Improvement**
  - Wear leveling
  - Write intensity reduction

- **Reliability/Security Improvement**
  - Secure wear-leveling
  - Specific ECC
  - Data encryption
PCM as a Candidate for Future Main Memory

- PCM has relative low latency (~100ns) compared to Flash and high density (~2X) compared to DRAM.
- The main challenges are the limited PCM endurance, longer access latencies, and higher dynamic power compared to the conventional DRAM technology.

Figure 4: Candidate main memory organizations
(a) Traditional system (b) Aggressive system with Flash-based disk cache (c) System with PCM (d) System with Hybrid memory system.
DRAM can play as a buffer for PCM not only to speed up access, but also can hold some update data reducing writes to PCM (Lazy Write) to reduce write intensity.

**Figure 5: Lazy Write Organization**

Flip-N-Write MICRO 2009

To Improve PRAM Write Performance, Energy and Endurance, Flip-N-Write is

- (i) to replace a write operation with a read-modify-write operation in order to skip bit programming action if not needed (e.g., writing a “0” on “0”)
- (ii) to limit the maximum number of bits to program by introducing a “flip bit.” The flip bit indicates whether the associated PRAM word has been flipped or not.

Write Cancellation M. K. Qureshi, HPCA 2010

- **Write Cancellation**: can abort the processing of a scheduled write requests if a read request arrives to the same bank within a predetermined period.

- **Write Pausing**: exploits the iterative write algorithms used in PCM to pause at the end of each write iteration to service any pending reads.

- **46% Performance Improvement**

**PreSet** M. K. Qureshi, ISCA 2012

- PCM writes are slow only in one direction (SET operation) and are almost as fast as reads in the other direction (RESET operation).
- Pro-actively SET all the bits in a given memory line well in advance of the anticipated write to that memory line.
- Initiates a PreSET request for a memory line as soon as that line becomes dirty in the cache, thereby allowing a large window of time for the PreSET operation to complete.
- 34% Performance and 25% Energy Efficiency Improvement.

---

Observation: an unbalanced distribution of modified data bits among cell groups significantly increases PCM write time and hurts effective write bandwidth.

Double XOR mapping (D-XOR): distribute modified data bits among cell groups in a balanced way

45% write time reduction and 1.8x write throughput

With ECC, 51% write time reduction and 12% IPC improvement
Observation: there is a wide distribution in cell resistance in both the SET state and the RESET state, and that the read latency of PCM is designed conservatively to handle the worst case cell.

- **Early Read**: reads the data earlier than the specified time period. Using Berger codes to detect read data errors.
  - 25% read time reduction.

- **Turbo Read**: reduces the sensing time for read operations by pumping higher current, using Error Correction Codes (ECC) and Probabilistic Row Scrubbing (PRS) to maintain data integrity.

---

**Fig. 1.** (a) Timing and voltage for read and write operations (b) Variation in cell resistance for SET and RESET (c) Early Read lowers the reference resistance ($R_{r,ref}$) at the expense of some SET cells being read as RESET (d) Turbo Read increases $V_{rd}$ to lower $T_{rd}$, at the expense of accidentally flipping some RESET cells into SET state.
PCM Main Memory

- Performance/Energy Improvement
  - Hybrid memory
  - Asymmetric write
  - Coding

- Lifetime Improvement
  - Wear leveling
  - Write intensity reduction

- Reliability/Security Improvement
  - Secure wear-leveling
  - Specific ECC
  - Data encryption
Improving the Endurance Of PCM

- It is common that only a few bits are changed in a new memory block write (>=64Byte).
- We can simply first read the data and compare it with the new data. And only update the different bits. Thus, redundant writes are avoided.
- Wear-Leveling: In addition, if 50% of bits are to be updated. We can first shift the row and then update (need to record the shift bit).
- The PCM lifetime is increased from 176 days to 22 years

Table 1: Final lifetime (years) with redundant bit-write removal, row shifting, and segment swapping.

<table>
<thead>
<tr>
<th>Benchmarks</th>
<th>SLC(segment swap only)</th>
<th>SLC</th>
<th>MLC-2</th>
<th>MLC-4</th>
</tr>
</thead>
<tbody>
<tr>
<td>ammp</td>
<td>4.9</td>
<td>32.5</td>
<td>22.7</td>
<td>19.3</td>
</tr>
<tr>
<td>art</td>
<td>1.3</td>
<td>24.8</td>
<td>18.2</td>
<td>15.4</td>
</tr>
<tr>
<td>lucas</td>
<td>1.1</td>
<td>15.0</td>
<td>12.7</td>
<td>11.2</td>
</tr>
<tr>
<td>mggrid</td>
<td>1.8</td>
<td>26.7</td>
<td>20.0</td>
<td>16.2</td>
</tr>
<tr>
<td>milc</td>
<td>2.3</td>
<td>36.3</td>
<td>32.8</td>
<td>31.5</td>
</tr>
<tr>
<td>ocean</td>
<td>52.8</td>
<td>224.4</td>
<td>161.2</td>
<td>133.1</td>
</tr>
<tr>
<td>sphinx3</td>
<td>5.5</td>
<td>586.2</td>
<td>583.3</td>
<td>580.9</td>
</tr>
<tr>
<td>swim</td>
<td>0.3</td>
<td>3.9</td>
<td>3.0</td>
<td>2.1</td>
</tr>
<tr>
<td>mcf</td>
<td>3.3</td>
<td>1285.4</td>
<td>889.0</td>
<td>719.5</td>
</tr>
<tr>
<td>specweb-banking</td>
<td>4.4</td>
<td>22.1</td>
<td>17.5</td>
<td>15.2</td>
</tr>
<tr>
<td>specweb-ecommerce</td>
<td>8.0</td>
<td>42.9</td>
<td>34.1</td>
<td>27.9</td>
</tr>
<tr>
<td>specweb-support</td>
<td>10.2</td>
<td>83.9</td>
<td>69.1</td>
<td>62.7</td>
</tr>
<tr>
<td>Amean</td>
<td>8.0</td>
<td>198.7</td>
<td>155.3</td>
<td>136.3</td>
</tr>
<tr>
<td>Hmean</td>
<td>1.7</td>
<td>22.1</td>
<td>17.1</td>
<td>13.4</td>
</tr>
</tbody>
</table>

Figure 7: Implementation of redundant bit-write removal and row shifting. Added hardware are circled.

The lifetime of a PCM system can be improved by making writes uniform throughout the entire memory space.

Start-Gap wear leveling that uses an algebraic mapping between logical addresses and physical addresses, and avoids tracking per-line write counts.

The achievable lifetime of the baseline 16GB PCM-based system is boosted from 5% (with no wear-leveling) to 97% of the theoretical maximum, while incurring a total storage overhead of less than 13 bytes and obviating the latency overhead of accessing large tables.

**Figure 7: Mapping of Logical Address to Physical Address.**

Randomized Start-Gap

**PCM Security**

- Static Randomizer
- Randomized Start-Gap
- Optimized Structure

**Random Invertible Binary (RIB) Matrix**

**Diagram:**
- LA (Logical Address)
- IA (Intermediate Address)
- PA (Physical Address)
- Stage 1: Key1, F1, EXOR
- Stage 2: Key2, F1
- Stage 3: Key3, F1

**Formulas:**

\[
\begin{bmatrix}
0 & 1 & 1 & 1 \\
1 & 0 & 1 & 0 \\
0 & 1 & 0 & 1 \\
0 & 1 & 1 & 0 \\
\end{bmatrix}
\times
\begin{bmatrix}
a \\
b \\
c \\
d \\
\end{bmatrix}
= 
\begin{bmatrix}
A \\
B \\
C \\
D \\
\end{bmatrix}
\]
The key idea of coset coding is that it performs a one-to-many mapping from each dataword to a coset of vectors, and having multiple possible vectors provides the flexibility to choose the vector to write that optimizes lifetime.

FlipMin: uses coset coding and, for each write, selects the vector that minimizes the number of bits that must flip.

Every element in S3 represents D3=10

If we wants to flip 0101(00) to (10) select 0010 from S3 to minimize bit flips

D1=00 maps to \( S_1 = \{0000, 0101, 1010, 1111\} \)
D2=01 maps to \( S_2 = \{0001, 0100, 1011, 1110\} \)
D3=10 maps to \( S_3 = \{0010, 0111, 1000, 1101\} \)
D4=11 maps to \( S_4 = \{0011, 0110, 1001, 1100\} \)

Word Level Compression Restricted Coset (WLCRC) code

- Encoding data blocks at small granularity to reduce write energy actually increases the write energy because of the auxiliary encoding bits.
  - Suitable codeword mappings that use fewer auxiliary bits
  - Word-Level Compression (WLC) which compresses more than 91%

**Figure 6**: (a) Word Level Compression (WLC), (b) Restricted coset coding at 16-bit granularity.

**Figure 7**: On-chip WLCRC architecture for 16-bit granularity.
PCM Main Memory

- Performance/Energy Improvement
  - Hybrid memory
  - Asymmetric write
  - Coding

- Lifetime Improvement
  - Wear leveling
  - Write intensity reduction

- Reliability/Security Improvement
  - Secure wear-leveling
  - Specific ECC
  - Data encryption
Unlike charge-based DRAM, the material state of PCM cells is not susceptible to particle-induced soft errors.

In “read-write-read”, a final read checks to ensure that the data were correctly written.

ECP, not ECC (Error-Correcting Codes)
- Handle permanent faults rather than soft errors
- Encodes and stores the addresses of failed cells and allocates additional cells to replace them.
Security Refresh ISCA 2010

- Prevent Malicious Wear-out and Increase Durability for Phase-Change Memory with Dynamically Randomized Address Mapping
- Must take the worst-case scenario into account with the presence of malicious exploits and a compromised OS to address the durability and security issues simultaneously.
- Security Refresh swaps data using random keys upon each refresh due based on Random-Based Start-Gap (RBSG) [Qureshi, MICRO’09]

![Diagrams](attachment:security_refresh_diagrams.png)

Figure 4: Security Refresh Terminology

SAFER exploits the key attribute that a failed cell with a stuck-at-value is still readable, making it possible to continue to use the failed cell to store data.

SAFER partitions a data block dynamically while ensuring that there is at most one fail bit per partition and uses single error correction techniques per partition for fail recovery.

Data Block Inversion of SAFER

---

Prior solutions are

- oblivious to soft errors (recently raised as a potential issue even for PCRAM) and
- incompatible with high-level fault tolerance techniques such as chipkill.
- need custom logic in PCM devices

Key observation that even if a block is deemed dead, it still has many functional bits that can store useful information.

FREE-p: fine-grained remapping with ECC and embedded pointers.

- Data/Pointer flag (D/P)
- Remapped pointer cache
- Index cache with hashing

Figure 1. Example of fine-grained remapping with an embedded pointer.

Figure 2. Chained remapping example (solid pointers) and limiting the number of hops (dashed pointer).
The key observation that only a few lines require high levels of hard-error correction.

PAYG: allocates error correction entries in proportion to the number of hard-faults in the line. PAYG splits the correction entries into two parts:

- Per-line Local Error Correction (LEC) that can correct up to 1 error per line (95%)
- Global Error Correction (GEC) pool that contains tagged ECP entries and provides error correction entries for lines that have more errors
Persistent Memory

- Storage Class Memory (SCM) promise user-level access to non-volatile storage through regular memory instructions and enable fast user-mode access to persistence, allowing regular in-memory data structures to survive system crashes.

- Persistent Memory: a OS abstraction enables programmers to make in-memory data structures persistent without first converting them to a serialized format.
  - Simple for programmer to declare data as persistent, `pstatic`
  - Support consistent modifications of data
  - Compatible with existing commodity processors

- Mnemosyne
  - Persistent memory regions
  - Persistence primitives
  - durable memory transaction
  - raw word log (RAWL)

- 37%-117% faster

---

Figure 2. Example of missing write detection in the RAWL. Entries written on the current pass through the log have a torn bit of ‘1’, while entries written the previous pass have a ’0’. A missing write is detected by an incorrect torn bit.

Persistent memory systems must update a set of programmer-defined nonvolatile locations in an atomic, consistent, and durable way to enforce crash consistency.

Kiln employs a nonvolatile LLC and a nonvolatile memory to construct a persistent memory hierarchy, allows a persistent memory system to directly update the real in-memory data structures, rather than performing high cost logging or copy-on-write (COW).
Heterogeneous Memory Management

- 3D/2.5 stacked DRAM
  - 2x-4x low capacity
  - 10x high bandwidth, 1.5x low latency

- PCM and STT-RAM
  - 4x-8x high capacity
  - 2x high read latency, 5x-8x high write latency

Figure 1: Memory system using DRAM and PCM DIMMs. DRAM and PCM arrays are slightly different (cells and sense amplifiers).
OS level page manager that utilizes the write frequency information provided by the hardware to perform uniform wear leveling across all the PRAM pages.

- Page Allocator
- Page Swapper
Emerging Memory System Architecture

- Overhead of Moving Data
  - ~200x times more than FP computing itself
  - Technology improvement does not help

Bill Daily, “The Path to ExaScale”, SC14
Shekhar Borkar, “Exascale Computing—a fact or a fiction?”, IPDPS’13
Reducing overhead of data moving

- Move more memory closer to CPU/GPU
  - eDRAM (such as IBM Power 8, 96MB eDRAM)
  - 2.5D/3D Stacked memory (such as AMD Fury GPU, 4GB on-package stacked DRAM)

- Move more computing operations into memory
  - Near-data-computing (NDC)
  - Processing-in-memory (PIM)
  - In/Near Memory Processing

- Make the communication efficient
  - Optical interconnect

- NVM-enable PIM: ReRAM (Memristor)
Summary

- Heterogeneous Memory System
- DRAM Based HMS
- 3D-DRAM Based HMS
- NVM Memory
  - Performance/Power (Write)
  - Endurance (Write)
  - Reliability (ECP)
  - Security
  - NVM Based HMS
- In/Near-Memory-Processing