SSD alternatives
Memory hierarchy

Fig. 1.2 Memory hierarchy (Created by Forward Insights)
• HDD manufacturers discovered ways to read more accurately than writing
• Physical read size < write size 😞
  – Applying magnetic field affects nearby tracks → erase band
  – Reading does no damage, can use narrower head
• Demagnetizing (erase) creates collateral damage
• Result: HDDs started using separate read/write heads
HDDs left behind

- Changes to HDD technology limited by physics
- Cant rotate too fast (heat)
- Bit size == write head size
- But tracks cant get infinitely narrow
  - cant magnetize too small grains
  - write heads become narrower
    → weaker magnetic fields
    → more error-susceptible
Lemonade out of lemons

- Shingled Magnetic Recording (SMR)
- Physical read size < write size < erase size? **Let's utilize it!**
- Write/erase not around track, but in direction that overlays next track!
Shingled Magnetic Recording Models, standardization, and Applications, SNIA’14
Q: what the problem? and what’s the design solution?
Indirection to the rescue

- Basic SMR design similar to block mapping in low-end SSDs
Problem: indirection makes random write suck 😐
Solution?
• Option 1: use log-structured file system on host
• Option 2: make smarter firmware
  – Shingle translation layer (STL)
Shingle translation layer

- Various options, similar ideas in SSD design
  - Persistent dedicated on-disk/flash journal cache (for random writes mostly)
  - Static mapping (block-mapped)
  - Fully associative designs
    - log bands, later merged to final location during GC
    - Lazy/aggressive GC
• Q: How can we tell what the STL is doing?
Skylight

• Q: How can we tell what the STL is doing?
  – Can look at firmware code
  – Deduce from response times
  – Can also drill a hole in disks to take a look
• Simple example: find location of disk cache
  – Randomly write small 1GB area
Memories (wish list)

- Infinite capacity
- Infinitely fast
- Non-volatile
- Minimal power required
- Reliable
- Cheap
Memories (real life)

• A lot of work on alternatives to flash
• Q: what is realistic?
Flash alternatives

• Faster than flash
  – Slower than DRAM
• Non-volatile
• Byte-addressable
• Cheaper than flash, ideally also from DRAM
• Where do these memories fit?

![Memory Hierarchy Diagram](image)

Fig. 1.2 Memory hierarchy (Created by Forward Insights)
Phase-change memories

• Most promising technology thus far
• Basic idea: use electrical resistance to encode data bits
  – Flash uses electrical charging
• Originally discovered in 1968
• Much research over the years
• PCM-based SSDs only coming out now...
• Cells contain special material
• Phase of material changes between two states
  – Crystalline (logical ‘1’) = small resistance
  – Amorphous (logical ‘0’) = large resistance
• Phase change induced by heat (650c)
• Phase-change material connected to heater element
• Two electrodes pass current

DyPhase: A Dynamic Phase Change Memory Architecture with Symmetric Write Latency, VLSI’17
• **RESET (0→1)**
  - Amorphous → Crystalline
  - Long medium-current electrical pulse heats amorphous material
  - Above certain threshold the material crystalizes (lower resistance)

• **SET (1→0)**
  - Crystalline → Amorphous
  - Short high current pulse heats material and melts it
  - After quick cool-off material is amorphous again
• **READ**
  – Short **low-voltage** current through cell
  – Measure resistance
Latencies

- Read 125ns
- SET 125ns
- RESET 250ns

- Q: Program latency?
MLC PCM

- Slower than SLC PCM
- Q: Why?
MLC PCM

- Slower than SLC PCM
- **Q: Why?**
  - One RESET, Multiple pulses of small SET operations
• Severely suffers from retention problem
  – Cells continue to crystalize even after heater stops

• **Q: Solution?**
• Severely suffers from retention problem
  – Cells continue to crystalize even after heater stops

• **Q: Solution?**
  – periodic refresh...
• Cell organization? depends on what you want...
• But generally speaking, multiple chips in parallel for performance
  – Updates performed in 512B/256B/64B units
PCM pros and cons

• **Pros:**
  – Non volatile
  – Low power versus flash
  – Byte addressable
    • Flash page-addressable
  – High P/E cycles endurance
    • $10^{16}$ DRAM > $10^8$ PCM > $10^5$ flash (SLC!)
  – Very dense, 8nm cell feature size
    • Better than flash, better than DRAM!
• **Cons:**
  – Higher latencies: 4-15x DRAM
  – 2-50x more power than DRAM
  – Endurance limits use as DRAM replacement
    • And reliability issue ("resistance drift")
  – MLC requires some refresh
PCM in real-life

- On paper sounds great
  - 1M P/E cycles
  - Sub us accesses
  - Byte-addressable
  - Can replace DRAM altogether (or most of it)
- Flash is DONE! that was said already in 2008...
• Intel/Micron only released 1st high-capacity PCM-based SSD in 2017
• Uses NVMe SSD interfaces (not optimal!)
• Chips support slower write throughput than promised
  – Technical limitations writing multiple bits concurrently
  – 20-30 MB/s for single chip
  – Improvement via parallelism

• Cost: $2.5/GB
  – 5x more than flash

• Endurance: 30 DWPD
  – Typical high-end flash SSDs = 3-7 DWPD

• What about performance?

<table>
<thead>
<tr>
<th>Next generation PCM chip</th>
<th>50%</th>
<th>99%</th>
<th>99.9%</th>
</tr>
</thead>
<tbody>
<tr>
<td>128B Write</td>
<td>2.9 us</td>
<td>3.1 us</td>
<td>4.1 us</td>
</tr>
<tr>
<td>128B Read</td>
<td>3.7 us</td>
<td>3.9 us</td>
<td>4.7 us</td>
</tr>
</tbody>
</table>

Controller Architecture for Low-latency Access to Phase-Change Memory in OpenPOWER Systems, FPL’16
Sequential performance

Intel Optane SSD 900P Review: 3D XPoint
Unleashed, Tom’s Hardware 2017
Q: is it all hype? What’s missing?
Mixed workloads

- Achilles hill of flash
More alternatives

- STT-RAM=Spin-Transfer Torque RAM
- Manipulate magnetic orientation of special material layer
- BUT magnetic orientation changes the electrical resistance of cell
• Endurance = same as DRAM ($10^{16}$)
• Latencies similar to DRAM
• Still too pricey, difficult producing large chips
• Technical difficulty making MLC cells
• Relatively energy hungry
And more alternatives...

<table>
<thead>
<tr>
<th>Attribute</th>
<th>FRAM</th>
<th>MRAM</th>
</tr>
</thead>
<tbody>
<tr>
<td>Technology</td>
<td>1 transistor – 1 ferroelectric capacitor (1T – 1FRC)</td>
<td>1 transistor – 1 magnetic tunnel junction (1T – MTJ)</td>
</tr>
<tr>
<td>Read</td>
<td>Destructive</td>
<td>Non-destructive</td>
</tr>
<tr>
<td>Read Cycle Time</td>
<td>90ns</td>
<td>35ns</td>
</tr>
<tr>
<td>Read Access Time</td>
<td>60ns</td>
<td>35 ns</td>
</tr>
<tr>
<td>Write Cycle Time</td>
<td>90ns</td>
<td>35ns</td>
</tr>
<tr>
<td>Write (CE)</td>
<td>60ns</td>
<td>35ns</td>
</tr>
<tr>
<td>Fatigue</td>
<td>Known wear-out mechanisms 10^{14} Read / Write</td>
<td>No wear-out. Infinite Read / Write</td>
</tr>
</tbody>
</table>

Source: everspin technologies
The future

• So far: operating systems (mostly) assumed either:
  1. Slow non-volatile storage (SSD, HDD) on slow buses, or...
  2. Fast volatile memory on fast bus (close to processor)

• The real benefit may come when PCM-SSDs are used on the memory bus

• No more compromise? NV-RAM can change how operating systems are designed
• May unknowingly relax persistence
• If used in tandem with DRAM we better expressively state the persistence of memory allocations (or lack of)
Replacing DRAM altogether

• Simply using as DRAM replacement in memory-intensive workloads endurance can be a problem
• Q: Why? How to resolve?

Figure 2: Raw lifetime of PCM main memory.

A Durable and Energy Efficient Main Memory Using Phase Change Memory Technology, ISCA'09
• But can be resolved by adapting memory accesses
  – Sub-row updates to reduce wear $\rightarrow$ 5x better
    • Didn’t matter for DRAM
  – Wear-levelling
    • E.g., periodic replacement+remapping of large segments (e.g., 1MB)
• Performance
  – Cycle-accurate simulation results
    • Measures cycles per instruction (CPI)
  – **PCM slower, but overall performance similar!**
  – Existing programs may be insensitive to slower main memory...

*Figure 15: Memory latency impact on CPI.*
What about storage?

- No need for slow-bus storage anymore
- May need to re-think file systems altogether
- Re-design for Storage Class Memories (SCM)

SCMFS: A File System for Storage Class Memory, SC’11
SCM file system

- “Storage” directly attached to CPU
- Part of physical memory contains page maps (as usual)
- Part of memory dedicated for:
  - superblocl + inode
  - Files (!)

Figure 4: Memory space layout
• Re-use MMU for mapping virtual addresses to physical addresses
• Files allocated contiguous logical addresses
  – File represented by <start address, size>
  – We’ve already seen this before...remember?
• Physical addresses obtained via MMU
File edits

• Can result in
  – Many memory operations
  – Memory fragmentation

• Reduce by maintaining “null” files
  – Ready-to-use
  – Can recycle deleted files

• Don’t deallocate shrunk files
  – When space runs low use garbage collection to reclaim space
New technology, new (old) problems

• Data written to CPU cache before memory
  – LRU-flushed to memory (for performance)
  – LRU = order of writes?
  – CPU cache non-volatile

• Write ordering becomes an issue
  – May write out-of-order updates
  – E.g., inode in CPU cache, data in memory

• Regular file systems faced similar problem
  – Changes in (relatively large) memory not reflected on (much larger) disk
  – Solved by force flushing metadata to disk
• Same solution (sort of) applies here

• Metadata consistency
  – Critical information (superblock, inodes, directories...)
  – Use instruction set calls for serialization (MFENCE) and invalidation of cache lines containing specific addresses on every metadata update

• Data consistency
  – Periodic flush of cache
Figure 9: Iozone results with multi-thread (Random workload)

- SCMFS uses virtual address space (no large page optimizations \(\rightarrow\) less efficient TLB use)
Can flash keep up?

- Z-NAND is a new Samsung flash technology
  - SLC (probably)? Better controller? More DRAM?
  - Not much details yet
  - Only datasheet numbers...

- **Q**: does it keep up?

---

<table>
<thead>
<tr>
<th>Tech</th>
<th>Capacity</th>
<th>R/W IOPS</th>
<th>R/W bandwidth</th>
<th>Life</th>
<th>R/W Latency</th>
</tr>
</thead>
<tbody>
<tr>
<td>Optane P4800X</td>
<td>750GB</td>
<td>550K</td>
<td>2.4/2.0 GB/sec</td>
<td>41 PBW</td>
<td>10/10μs</td>
</tr>
<tr>
<td>SZ985</td>
<td>800GB</td>
<td>750K/170K</td>
<td>3.2/3.2 GB/sec</td>
<td>42.7 PBW</td>
<td>12-20/16μs</td>
</tr>
</tbody>
</table>

A challenger appears: Specs for Samsung's potential Optane killer, TheRegister 2017
Flash can also increase capacity

**Intel, Micron's first-ever QLC NAND flash: Cheaper, denser SSD storage is coming**

Intel and Micron’s first Quad-Level Cell NAND flash memory should bring higher-density SSD storage.

*By Liam Tung | May 22, 2018 -- 13:19 GMT (14:19 BST) | Topic: Storage*

- Or encode data in more efficient ways...
• Q: can flash write data differently? smaller than page chunks? In-place?
• Already tried for representing 2 bits in 3 cells with one rewrite

<table>
<thead>
<tr>
<th>Plain bits</th>
<th>1st gen</th>
<th>2nd gen</th>
</tr>
</thead>
<tbody>
<tr>
<td>00</td>
<td>111</td>
<td>000</td>
</tr>
<tr>
<td>01</td>
<td>110</td>
<td>001</td>
</tr>
<tr>
<td>10</td>
<td>101</td>
<td>010</td>
</tr>
<tr>
<td>11</td>
<td>011</td>
<td>100</td>
</tr>
</tbody>
</table>

• Use only on young hot pages (small % of pages)
• Can improve performance and lifetime

Write Once, Get 50% Free:
Saving SSD Erase Costs Using WOM Codes, FAST’15
Flash may have run its course
The graph illustrates the trend of Flash Chip Price ($/Gbit) over time from 2008 to 2023. Three types of Flash memory are compared: SLC, MLC, and TLC. The price decreases sharply over the years, with SLC showing the highest initial price and a slower decrease compared to MLC and TLC, which have a more rapid decrease but start at a higher price point.
How long can flash keep up?
How long can flash keep up?

- 2006: SSDs developed by Samsung cost $45/GB
  
  - 1TB = $45K 😞

Loser: Too Little, Too Soon

Samsung’s solid-state disks will be puny, pricey, and impractical

By Harry Goldstein

“the cost per gigabyte of flash, while falling 30 to 40 percent per year, has stayed sky-high relative to that for hard drives and will remain so for the foreseeable future”, IEEE Spectrum