Solid-State Drives (SSDs)

Jin-Soo Kim (jinsookim@skku.edu)
Computer Systems Laboratory
Sungkyunkwan University
http://csl.skku.edu
Solid-State Drives (SSDs)
HDDs vs. SSDs (1)

2.5” HDD    Flash SSD
(101x70x9.3mm)

1.8” HDD    Flash SSD
(78.5x54x4.15mm)
## HDDs vs. SSDs (2)

<table>
<thead>
<tr>
<th>Feature</th>
<th>SSD (Samsung)</th>
<th>HDD (Seagate)</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Model</strong></td>
<td>MMDOE56G5MXP (PM800)</td>
<td>ST9500420AS (Momentus 7200.4)</td>
</tr>
<tr>
<td><strong>Capacity</strong></td>
<td>256GB (16Gb MLC x 128, 8 channels)</td>
<td>500GB (2 Discs, 4 Heads, 7200RPM)</td>
</tr>
<tr>
<td><strong>Form factor</strong></td>
<td>2.5” Weight: 84g</td>
<td>2.5” Weight: 110g</td>
</tr>
<tr>
<td><strong>Host interface</strong></td>
<td>Serial ATA-2 (3.0 Gbps)</td>
<td>Serial ATA-2 (3.0 Gbps)</td>
</tr>
<tr>
<td></td>
<td>Host transfer rate: 300MB</td>
<td>Host transfer rate: 300MB</td>
</tr>
<tr>
<td><strong>Power consumption</strong></td>
<td>Active: 0.26W</td>
<td>Active: 2.1W (Read), 2.2W (Write)</td>
</tr>
<tr>
<td></td>
<td>Idle/Standby/Sleep: 0.15W</td>
<td>Idle: 0.69W, Standby/Sleep: 0.2W</td>
</tr>
<tr>
<td><strong>Performance</strong></td>
<td>Sequential read: Up to 220 MB/s</td>
<td>Power-on to ready: 4.5 sec</td>
</tr>
<tr>
<td></td>
<td>Sequential write: Up to 185 MB/s</td>
<td>Average latency: 4.17 msec</td>
</tr>
<tr>
<td><strong>Measured performance</strong></td>
<td>Sequential read: 176.73 MB/s</td>
<td>Sequential read: 86.07 MB/s</td>
</tr>
<tr>
<td>(On MacBook Pro, 256KB for sequential, 4KB for random)</td>
<td>Sequential write: 159.98 MB/s</td>
<td>Sequential write: 84.64 MB/s</td>
</tr>
<tr>
<td></td>
<td>Random read: 10.56 MB/s</td>
<td>Random read: 0.61 MB/s</td>
</tr>
<tr>
<td></td>
<td>Random write: 2.93 MB/s</td>
<td>Random write: 1.28 MB/s</td>
</tr>
<tr>
<td><strong>Price</strong></td>
<td>539,190 won</td>
<td>80,400 won</td>
</tr>
</tbody>
</table>

NAND Constraints
No In-Place Update

- Can’t overwrite data
- Previous data should be erased
- The erase unit is much larger than the read/write unit
  - Read/write unit: page (4KB, 8KB)
  - Erase unit: block (64 ~ 128 pages)
Bit Errors

- Error correction codes (ECC) in spare area
- Hardware vs. software

<table>
<thead>
<tr>
<th>Error Correction Level</th>
<th>Bits Required in the NAND Flash Spare Area</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Hamming</td>
</tr>
<tr>
<td>1</td>
<td>13</td>
</tr>
<tr>
<td>2</td>
<td>N/A</td>
</tr>
<tr>
<td>3</td>
<td>N/A</td>
</tr>
<tr>
<td>4</td>
<td>N/A</td>
</tr>
<tr>
<td>5</td>
<td>N/A</td>
</tr>
<tr>
<td>6</td>
<td>N/A</td>
</tr>
<tr>
<td>7</td>
<td>N/A</td>
</tr>
<tr>
<td>8</td>
<td>N/A</td>
</tr>
<tr>
<td>9</td>
<td>N/A</td>
</tr>
<tr>
<td>10</td>
<td>N/A</td>
</tr>
</tbody>
</table>

Source: Micron Technology, Inc.
Bad Blocks

- Factory-marked bad blocks
- Run-time bad blocks
- Bad block table
- Bad block remapping
Limited Lifetime

- Limited program/erase cycles
- 100K for SLCs
- 5K ~ 10K for MLCs
- Wear-leveling
Page Programming

- **NOP**
  - The number of partial-page programming is limited
  - 1 / sector for most SLCs (4 for 2KB page)
  - 1 / page for most MLCs

- **Sequential page programming**
  - Pages should be programmed sequentially inside a block
  - For large block SLCs and MLCs
Paired Pages in MLCs

- Performance difference
- Interference

I/O-split MLCs

Page-split MLCs
Beauty and the Beast

▪ NAND Flash memory is beauty
  • Small, light-weight, robust, low-cost, low-power non-volatile device

▪ NAND Flash memory is a beast
  • Much slower program/erase operations
  • No in-place-update
  • Erase unit > write unit
  • Limited lifetime (5K ~ 100K P/E cycles)
  • Bad blocks, ...
Introduction to
Flash Translation Layers
Storage Abstraction

- Abstraction given by block device drivers:

  - Operations
    - Identify(): returns N
    - Read (start sector #, # of sectors)
    - Write (start sector #, # of sectors, data)

Source: Sang Lyul Min (Seoul National Univ.)
What is FTL?

- A software layer to make NAND flash fully emulate traditional block devices (or disks)
Flash Cards Internals
SSD Internals
Implementing FTLs

Flash Cards, SSDs

Applications

Operating System

File Systems

Block Device Driver

Flash Translation Layer

NAND Controller

NAND Flash Memory

Embedded Flash Storage

Applications

Operating System

File Systems

Block Device Driver

Flash Translation Layer

NAND Controller

NAND Flash Memory
For Performance

- Indirect mapping (address translation)
- Garbage collection
- Over-provisioning
- Hot/cold data identification/separation
- Interleaving over multiple channels/flash chips/planes
- Request scheduling
- Buffer management, ...
For Reliability

- Bad block management
- Wear-leveling
- Power-off recovery
- Error correction code (ECC)
- ...
Other Features

▪ Encryption
▪ Compression
▪ Deduplication
▪ ...


Basic Mapping Techniques
Naïve Block Mapping

- Each table entry maps one block
- Small RAM usage
- Inefficient handling of small writes
Example (1)

- $W = \langle \{4, 5, 6, 7\}, \{1\} \rangle$
  - write ($\{4, 5, 6, 7\}$, ABCD)
Example (2)

- $W = \langle\{4,5,6,7\}, \{1\}\rangle$
  - write ($\{4,5,6,7\}$, ABCD)
  - write ($\{1\}$, E)
ANAND Scheme

- **M-Systems**
- **US Patent 5,937,425**
- Uses one or more replacement blocks
- Data is written to an unwritten physical page with the desired physical page offset

W = \langle \{1\}, \{4,5\}, \{8\}, \{10\}, \{8\} \rangle
Naïve Page Mapping

- Most flexible
- Efficient handling of small writes
- Large memory footprint
  - 32MB for 32GB MLC (4KB page)
- Sensitive to the amount of reserved blocks
- Performance affected as the system ages
Example (1)

- \( W = \langle \{1\}, \{2\}, \{8\}, \{1\}, \{2\}, \{12,13\}, \{9\} \rangle \)
  - write \( (\{1\}, A) \)
**Example (2)**

- \( W = \langle \{1\}, \{2\}, \{8\}, \{1\}, \{2\}, \{12,13\}, \{9\} \rangle \)
  - write (\{1\}, A)
  - write (\{2\}, B)

```
0
1
4
14
10
5
6
7
2
8
11
13
15
9
3
12
```
Example (3)

- $W = <\{1\}, \{2\}, \{8\}, \{1\}, \{2\}, \{12,13\}, \{9\}>
  - write ($\{1\}$, A)
  - write ($\{2\}$, B)
  - write ($\{8\}$, C)
Example (4)

- $W = \langle \{1\}, \{2\}, \{8\}, \{1\}, \{2\}, \{12,13\}, \{9\}\rangle$
  - write (\{1\}, A)
  - write (\{2\}, B)
  - write (\{8\}, C)
  - write (\{1\}, D)
Example (5)

- \( W = \langle \{1\}, \{2\}, \{8\}, \{1\}, \{2\}, \{12,13\}, \{9\} \rangle \)
  - write (\{1\}, A)
  - write (\{2\}, B)
  - write (\{8\}, C)
  - write (\{1\}, D)
  - write (\{2\}, E)
Example (6)

- $W = \langle \{1\}, \{2\}, \{8\}, \{1\}, \{2\}, \{12,13\}, \{9\} \rangle$
  - write ($\{1\}$, A)
  - write ($\{2\}$, B)
  - write ($\{8\}$, C)
  - write ($\{1\}$, D)
  - write ($\{2\}$, E)
  - write ($\{12,13\}$, FG)
Example (7)

- \( W = \langle \{1\}, \{2\}, \{8\}, \{1\}, \{2\}, \{12,13\}, \{9\} \rangle \)
  - write (\{1\}, A)
  - write (\{2\}, B)
  - write (\{8\}, C)
  - write (\{1\}, D)
  - write (\{2\}, E)
  - write (\{12,13\}, FG)
  - write (\{9\}, H)
Victim Selection Policies

▪ Greedy
  • A block with the largest amount of invalid data

▪ Cost-benefit [Kawaguchi 1995]
  • A block with the maximum \( \frac{\text{benefit}}{\text{cost}} = \frac{(1-u)}{2u} \times \text{age} \)
  • \( u \): utilization
  • \( \text{age} \): the time since the most recent modification

▪ Cost Age Times (CAT) [Chiang 1999]
  • A block with the minimum \( \frac{u}{(1-u) \times \text{age}} \times \text{count} \)
  • \( \text{count} \): erase count