NAND Flash-based Storage

Jin-Soo Kim (jinsookim@skku.edu)
Computer Systems Laboratory
Sungkyunkwan University
http://csl.skku.edu
Today’s Topics

- NAND flash memory
- Flash Translation Layer (FTL)
- OS implications
Flash Memory Characteristics

- **Flash memory**
  - Non-volatile, Updateable, High-density
  - Low cost, Low power consumption, High reliability

- **Erase-before-write**
  - Read
  - Write or Program: 1 $\rightarrow$ 0
  - Erase: 0 $\rightarrow$ 1

- **Read faster than write/erase**

- **Bulk erase**
  - Erase unit: block
  - Program unit: byte or word (NOR), page (NAND)
NOR Flash

- NOR flash
  - Random, direct access interface
  - Fast random reads
  - Slow erase and write
  - Mainly for code storage
  - Intel, Spansion, STMicro, ...
NAND Flash

- NAND flash
  - I/O mapped access
  - Smaller cell size
  - Lower cost
  - Smaller size erase blocks
  - Better performance for erase and write
  - Mainly for data storage

- Samsung, Toshiba, Hynix, ...
NAND Flash Architecture

- 2Gb NAND flash device organization

Source: Micron Technology, Inc.
NAND Flash Types (1)

- **SLC NAND Flash**
  - Small block (≤ 1Gb)
  - Large block (≥ 1Gb)

- **MLC NAND Flash**

- **TLC NAND Flash**

Source: Micron Technology, Inc.
## NAND Flash Types (2)

<table>
<thead>
<tr>
<th></th>
<th>SLC NAND(^1) (small block)</th>
<th>SLC NAND(^2) (large block)</th>
<th>MLC NAND(^3)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Page size (Bytes)</td>
<td>512+16</td>
<td>2,048+64</td>
<td>4,096+128</td>
</tr>
<tr>
<td>Pages / Block</td>
<td>32</td>
<td>64</td>
<td>128</td>
</tr>
<tr>
<td>Block size</td>
<td>16KB</td>
<td>128KB</td>
<td>512KB</td>
</tr>
<tr>
<td>(t_R) (read)</td>
<td>15 µs (max)</td>
<td>20 µs (max)</td>
<td>50 µs (max)</td>
</tr>
<tr>
<td>(t_{\text{PROG}}) (program)</td>
<td>200 µs (typ) 500 µs (max)</td>
<td>200 µs (typ) 700 µs (max)</td>
<td>600 µs (typ) 1,200 µs (max)</td>
</tr>
<tr>
<td>(t_{\text{BERS}}) (erase)</td>
<td>2 ms (typ) 3 ms (max)</td>
<td>1.5 ms (typ) 2 ms (max)</td>
<td>3 ms (typ)</td>
</tr>
<tr>
<td>NOP</td>
<td>1 (main), 2 (spare)</td>
<td>4</td>
<td>1</td>
</tr>
<tr>
<td>Endurance Cycles</td>
<td>100K</td>
<td>100K</td>
<td>10K</td>
</tr>
<tr>
<td>ECC (per 512Bytes)</td>
<td>1 bit ECC 2 bits EDC</td>
<td>1 bit ECC 2 bits EDC</td>
<td>4 bits ECC 5 bits EDC</td>
</tr>
</tbody>
</table>

\(^1\) Samsung K9F1208X0C (512Mb)  \(^2\) Samsung K9K8G08U0A (8Gb)  \(^3\) Micron Technology Inc.
NAND Applications

- Universal Flash Drives (UFDs)
- Flash cards
  - CompactFlash, MMC, SD, Memory stick, ...
- Embedded devices
  - Cell phones, MP3 players, PMPs, PDAs, Digital TVs, Set-top boxes, Car navigators, ...
- Hybrid HDDs
- Intel Turbo Memory
- SSDs (Solid-State Disks)
SSDs (1)

- HDDs vs. SSDs

2.5” HDD          Flash SSD  
(101x70x9.3mm)      

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

<table>
<thead>
<tr>
<th>Feature</th>
<th>SSD (Samsung)</th>
<th>HDD (Seagate)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Model</td>
<td>MMDOE56G5MXP (PM800)</td>
<td>ST9500420AS (Momentus 7200.4)</td>
</tr>
<tr>
<td>Capacity</td>
<td>256GB (16Gb MLC x 128, 8 channels)</td>
<td>500GB (2 Discs, 4 Heads, 7200RPM)</td>
</tr>
<tr>
<td>Form factor</td>
<td>2.5” Weight: 84g</td>
<td>2.5” Weight: 110g</td>
</tr>
<tr>
<td>Host interface</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>Power consumption</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>Performance</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></td>
<td>Average latency: 4.17 msec</td>
<td></td>
</tr>
<tr>
<td>Measured performance¹</td>
<td>Sequential read: 176.73 MB/s</td>
<td>Sequential read: 86.07 MB/s</td>
</tr>
<tr>
<td></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>Price²</td>
<td>583,770 won</td>
<td>88,800 won</td>
</tr>
</tbody>
</table>

² Source: [http://www.danawa.com](http://www.danawa.com) (As of Nov. 21, 2010)
NAND Constraints (1)

- **No in-place update**
  - Require sector remapping (or address translation)

- **Bit errors**
  - Require the use of error correction codes (ECC)

- **Bad blocks**
  - Factory-marked & run-time bad blocks
  - Require bad block remapping

- **Limited program/erase cycles**
  - < 100K for SLCs
  - < 5K for MLCs
  - Require wear-leveling
NAND Constraints (2)

- **Limited NOP (Number of Programming)**
  - 1 / sector for most SLCs (4 for 2KB page)
  - 1 / page for most MLCs

- **Sequential page programming**
  - For large block SLCs and MLCs

- **Pair-page programming in MLCs**
  - Two pages inside a block are linked together
  - Performance difference
  - Interference
What is FTL?

- A software layer to make NAND flash fully emulate traditional block devices (e.g., disks).

Diagram:

- File System
  - Read Sectors
  - Write Sectors

- Device Driver
  - Read
  - Write
  - Erase

- Flash Memory

Diagram:

- File System
  - Read Sectors
  - Write Sectors

- Device Driver
  - Read Sectors
  - Write Sectors

- FTL

- Flash Memory

Source: Zeen Info. Tech.
FTL (2)

- Flash cards internals

![Flash cards image](image-url)
FTL (3)

- SSD internals

Source: Indilinx
FTL (4)

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
FTL (5)

- For performance
  - Address translation
  - Garbage collection
  - Hot/cold data identification/separation
  - Interleaving over multiple channels & flash chips
  - Request scheduling
  - Buffer management
  - ...

FTL (6)

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

- Other features
  - Encryption
  - Compression
  - Deduplication
  - ...
## Sector Mapping (1)

### General page mapping

- Most flexible
- Efficient handling of small writes
- Large memory footprint
  - One mapping entry per page: 32MB for 32GB MLC (4KB page)
  - Bitmap for page validity
  - Per-block invalid page counter
- Sensitive to the amount of reserved blocks
- Performance affected as the system ages

\[ W = <1, 2, 8, 1, 2, 12, 13, 9> \]
Sector Mapping (2)

- Naïve block mapping
  - Each table entry maps one block
  - Small RAM usage
  - Inefficient handling of small writes

W = <4, 5, 6, 7, 1>
**Sector Mapping (3)**

- **Log block scheme** [IEEE TOCE 2002]
  - A small number of log blocks
  - 1+ log block(s) per data block
  - Page mapping for log blocks
  - Full/partial/switch merge
  - Switch merge for sequential updates
  - Low log block utilization

![](image)

\[ W = \langle 1, 2, 8, 1, 2, 12, 13, 9 \rangle \]
Sector Mapping (4)

- **FAST** [ACM TECS 2007]
  - Log blocks shared by all data blocks
  - Sequential/random log blocks
  - Improved log block utilization
  - Increased merge time

\[ W = \langle 1, 2, 8, 1, 2, 12, 13, 9 \rangle \]
### Sector Mapping (5)

- **Superblock FTL** [ACM EMSOFT 2006]
  - Superblock = logically adjacent N blocks
  - A superblock shares log blocks
  - Up to M log blocks per superblock
  - Page mapping within a superblock
  - Hot/cold pages separation
  - The amount of mapping information increased

\[
W = \langle 1, 2, 8, 1, 2, 12, 13, 9 \rangle
\]
Sector Mapping (6)

- **μ-FTL** [ACM EMSOFT 2008]
  - Page mapping
  - Multiple mapping granularities
    - Based on extents
    - Reduce the amount of mapping information
  - Requires more sophisticated index structure
    - μ-Tree is used to store the mapping information
  - Tunable memory footprint
    - Frequently accessed mapping information cached in memory

\[ W = <1, 2, 8, 1, 2, 12, 13, 9> \]
Performance (1)

- **Simulation environment**
  - 4GB flash memory
    - Large block SLC NAND (2KB page, 128KB block)
  - FTL schemes
    - Naïve block mapping
    - Replacement block
    - Log block
    - Superblock
  - Workload
    - Trace from PC using NTFS
Extra erase and write operations

- 256 extra blocks
OS Implications (1)

- NAND flash has different characteristics compared to disks
  - No seek time
  - Asymmetric read/write access times
  - No in-place-update
  - Good sequential read/sequential write/random read performance, but bad random write performance
  - Wear-leveling
  - ...
  - Traditional operating systems have been optimized for disks. What should be changed?
OS Implications (2)

- SSD support in Microsoft Windows 7
  - Turn off “defragmentation” for SSDs
  - New “TRIM” command
    - Remove-on-delete
  - Align file system partition with SSD layout
  - Larger block size proposal (4KB)
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 (10K~100K program/erase cycles)
  - Bad blocks, ...

- Software support for NAND flash memory is very important for performance & reliability.