Challenges of SSD forensic analysis
Overview

- SSD Drive Technology Overview
- SSD Drive Components
  - NAND FLASH
  - Microcontroller
- SSD Drive Forensics
  - Challenges
Overview

- SSD’s are fairly new to the market
- Whereas HDD’s are well understood
- SSDs propose new challenges to forensics

Purpose

- Understand how SSDs function and understand the challenges of performing a forensics investigation.
SSD Overview

- Will replace HDD Drives
- Faster reads
- Faster writes
- Are small
- Use less energy
- Create less heat
- Are more expensive
  - The price has been decreasing
# SSD vs. HDD

<table>
<thead>
<tr>
<th></th>
<th>2.5” SATA 3.8Gbps SSD</th>
<th>2.5” SATA 3.8Gbps HDD</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Mechanism type</strong></td>
<td>Solid NAND Flash based</td>
<td>Magnetic rotating platters</td>
</tr>
<tr>
<td><strong>Density</strong></td>
<td>64GB</td>
<td>80GB</td>
</tr>
<tr>
<td><strong>Weight</strong></td>
<td>73g</td>
<td>365g</td>
</tr>
<tr>
<td><strong>Performance</strong></td>
<td>Read: 100MB/s Write: 80 MB/s</td>
<td>Read: 59MB/s Write: 60MB/s</td>
</tr>
<tr>
<td><strong>Active Power Consumption</strong></td>
<td>1W</td>
<td>3.86W</td>
</tr>
<tr>
<td><strong>Operating Vibration</strong></td>
<td>20G (10-2000Hz)</td>
<td>0.5G (22-350Hz)</td>
</tr>
<tr>
<td><strong>Shock Resistance</strong></td>
<td>1500G for 0.5ms</td>
<td>170G for 0.5ms</td>
</tr>
<tr>
<td><strong>Operating temperature</strong></td>
<td>0 °C– 70°C</td>
<td>5°C– 55°C</td>
</tr>
<tr>
<td><strong>Acoustic Noise</strong></td>
<td>None</td>
<td>0.3 dB</td>
</tr>
<tr>
<td><strong>Endurance</strong></td>
<td>MTBF &gt; 2M hours</td>
<td>MTBF &lt; 0.7M hours</td>
</tr>
</tbody>
</table>

*Source: [http://www.samsung.com]*
SSD Device Architecture

This can be Operating system File System
- FAT
- NTFS
- Journaling File System

Can be implemented in Hardware or Software
- SSDs do it in hardware (using Micro controller)
- xD Memory Cards (do it in the Driver)

Interface to Hardware through SATA or IDE

SSD Components

- Flash Memory
  - NOR Flash vs. NAND Flash
  - MLC vs. SLC
  - Limited Erase-write cycles
  - Read accuracy decreases after a certain number of reads.

- Implement techniques to overcome technology differences
  - COPYBACK (Read accuracy), ECC
  - Wear leveling (Limited Erase-write cycles)
NAND vs. NOR

NAND GATE

NOR GATE
## NAND vs. NOR

<table>
<thead>
<tr>
<th></th>
<th>SLC NAND Flash (x8)</th>
<th>MLC NAND FLASH (x8)</th>
<th>MLC NOR Flash (x16)</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Density</strong></td>
<td>512Mb – 4Gb</td>
<td>1Gbit to 16Gbit</td>
<td>16Mbit – 1Gbit</td>
</tr>
<tr>
<td><strong>Read Speed</strong></td>
<td>24 MB/s</td>
<td>18.6 MB/s</td>
<td>103MB/s</td>
</tr>
<tr>
<td><strong>Write Speed</strong></td>
<td>8.0 MB/s</td>
<td>2.4 MB/s</td>
<td>0.47 MB/s</td>
</tr>
<tr>
<td><strong>Erase Time</strong></td>
<td>2.0 msec.</td>
<td>2.0 msec.</td>
<td>900 msec.</td>
</tr>
<tr>
<td><strong>Interface</strong></td>
<td>I/O – indirect access</td>
<td>I/O – indirect access</td>
<td>Random access</td>
</tr>
<tr>
<td><strong>Application</strong></td>
<td>Program/Data mass storage</td>
<td>Program/Data mass storage</td>
<td>eXecute In Place (XIP)</td>
</tr>
</tbody>
</table>

# SLC vs. MLC

<table>
<thead>
<tr>
<th>Feature</th>
<th>SLC</th>
<th>MLC</th>
</tr>
</thead>
<tbody>
<tr>
<td>High Density</td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>Lower Cost Per Bit</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>Higher Endurance</td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>Greater Operating Temperature</td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>Range</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Lower Power Consumption</td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>Better Write/Erase Speeds</td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>Better Write/Erase Endurance</td>
<td></td>
<td>X</td>
</tr>
</tbody>
</table>

## What is the Difference

- **SLC**: NAND stores 2 states per memory cell and allows 1 bit programmed/read per memory cell.
- **MLC**: NAND stores 4 states per memory cell and allows 2 bits programmed/read per memory cell.

![Comparison Diagram]
NAND FLASH
Implementation Overview
TSOP 48 PINS

LGA-52 PAD

BGA-100

TSOP 48 is most common in most electronics, such as MP3 Players, USB sticks, Solid State Drives, Switches, Routers and the like.
Manufacturers can use any kind of interface that they want, but a group of companies created a consortium for NAND to standardize the industry.

Open NAND Flash Interface Working Group

- [http://onfi.org/](http://onfi.org/)

ONFI Specification 2.1

NAND Flash Specification

- Form Factor
- Memory Addressing
- Pin outs
- Timing
- Command Set
- This is good for Forensics Analysis
NAND Memory Organization

- Pages are basic programmable units of flash
- 512 bytes, but likely 2048 bytes
  - 16 bytes per 512 bytes for ECC and management
  - 2048 => 64 bytes
- 2112 bytes per page, but only 2048 usable
Blocks are basic erasable blocks
- Generally 64x pages per block
- 2048/64 * 64 bytes
  - 131072/4096 bytes per block
NAND Flash Organization

- 512 Bytes: Data Information
  - Can be addressed by the OS

- 2048 Bytes: Management & ECC
  - Not seen by OS

- 16 Bytes

- Block: 64 Pages
  - ...
NAND FLASH Operations

- **Reading**
  - Each read operation introduces a potential error
  - After several reads to the same location, there is some chance of error for consecutive reads from that location.

- **Writing**
  - Two types of operation to facilitate Writing
    - Set all bits to 1 (Erase)
    - Set bits from 1 to 0. (Program)
    - Cannot set bits from 0 to 1, must reset all bits to one (Erase)
  - To write a block, must erase (set all bits to 1), then program (set appropriate bits to 0).
  - To rewrite, must always use the erase program cycle.
Wear Leveling
Basic Algorithms

Source: Micron TN-29-42: Wear-Leveling Techniques in NAND Flash Devices
Wear Leveling (Micron)

- Wear leveling helps reduce premature wear in NAND Flash devices.
- Each Erase operation reduces life of device and makes it more vulnerable to read decay.
- Generally: 10,000 or 100,000 erase cycles
- Two primary wear leveling techniques
  - static
  - dynamic
Wear Leveling Importance

- Spread the drive wear across the entire drive
- Ensure that all blocks fail at approximately the same time
- Don’t allow some of the blocks to fail faster than others.
- Causes Severe Fragmentation
- Can dramatically reduce read and write speed of sequential reads.
Wear Leveling Implementation

<table>
<thead>
<tr>
<th>LBA</th>
<th>PBA</th>
</tr>
</thead>
<tbody>
<tr>
<td>0x00000000</td>
<td>0x00000000</td>
</tr>
<tr>
<td>0x00000001</td>
<td>0x00000001</td>
</tr>
<tr>
<td>0x00000002</td>
<td>0x20000002</td>
</tr>
<tr>
<td>0x00000003</td>
<td>0x00000003</td>
</tr>
<tr>
<td>0x00000004</td>
<td>0x20000004</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>0xFFFFFFFFD</td>
<td>0x2FFFFFFFFD</td>
</tr>
<tr>
<td>0xFFFFFFFFE</td>
<td>0x3FF000FE</td>
</tr>
<tr>
<td>0xFFFFFFFFF</td>
<td>0x4FF000FF</td>
</tr>
</tbody>
</table>

Host
- Use Logical Block Addressing (LBA)

Controller
- Has Look Up Table
- Translates between LBA and PBA

NAND Flash
- Uses Physical Block Addressing (PBA)
Wear Leveling: Dynamic vs. Static

- **Dynamic**
  - When allocating blocks, choose a least erased block from free list

- **Static**
  - When allocating blocks, choose a least erased block from free list
  - Occasionally
    - Move “static” non-free blocks with low erase count (below a threshold) to a block with a high erase count

- **Hybrid**
  - Allocate a portion of the drive for static wear leveling
  - Allocate a portion of the drive for dynamic wear leveling
## Choosing Wear-Leveling Methods

<table>
<thead>
<tr>
<th>Method</th>
<th>Advantages</th>
<th>Disadvantages</th>
</tr>
</thead>
<tbody>
<tr>
<td>Static</td>
<td>• Maximizes device life</td>
<td>• Requires more controller overhead</td>
</tr>
<tr>
<td></td>
<td>• Most robust wear-leveling method</td>
<td>• Can slow WRITE operations</td>
</tr>
<tr>
<td></td>
<td>• Most efficient use of memory array</td>
<td>• Higher power consumption</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• More complicated to implement than dynamic wear leveling</td>
</tr>
<tr>
<td>Dynamic</td>
<td>• Improves device life vs. no wear leveling</td>
<td>• May not optimize device life</td>
</tr>
<tr>
<td></td>
<td>• Easier to implement than static wear leveling</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• No impact on device performance</td>
<td></td>
</tr>
</tbody>
</table>
Intel x25 SSD
Intel x25m

- Certified for 25 MB/s read speed and 70 MB/s write speed.
- MLS SSD standard.
- 10-channel memory controller. Each channel is “responsible” for two memory chips.
- This controller works like a RAID 0 system but with flash memory.
SSD Architecture

Solid State Drive

Internal Architecture

Permanent Storage

SATA Data Interface

SATA Power Interface

SPI Flash Cache

Microcontroller

NVRAM

Flash Chip
Flash Chip
Flash Chip
Flash Chip
Flash Chip
Flash Chip
Flash Chip
Flash Chip
Flash Chip
Flash Chip
**Flash Storage**

Intel

\[29F32G08CAMCI\]
(Suspected Micron MT29F32G08)

- io83815 (8/4) [front/back]
- io83817 (2/6) [front/back]

**Description:**
Single Supply 32GbX8 NAND Flash
2048 + 64 byte pages (2112)
64 pages per block
32 blocks per chip
(4GB storage with 8-bit access)

---

**IO Microcontroller**

Intel PC29AS21AA0
(Unknown Chip Specification)
i0837

**Description:**
Possibly the Intel 8051 architecture
Possibly 8 bit architecture
Possibly like the SST

---

**SPI Flash Cache**

Winbond 25X40AVNIG

**Description:**
512KB SPI NAND Flash
Serial data access
8x64KB blocks
128x4KB sectors
2048x256B pages
256 pages per block
16 sectors per block

---

**NV RAM**

Samsung 843
K4S281632K-UC60
16MB SDRAM

---

**Serial ATA**

---

**Power**
Quick Analysis

- Microcontroller
  - Likely based on 8051 architecture by Intel
  - Probably has Internal ROM
- SPI Flash most likely contains either
  - ROM
  - Tables for keeping track of Wear Leveling.
- Flash Storage
  - More than likely Intel outsourced from Micron
- NVRAM
  - Likely used for quick writes or writes to the same blocks.
  - A similar setup is recommended by Micron in one of it’s whitepapers.
Challenges for Forensics
Challenges

- IDE interface allows logical data reads, but hides the internal data structures.
- Internals not well understood - may contain hidden data useful in forensics.
- No accepted standards
  - Every manufacturer does what it wants
- Manufacturers protect their implementation details to prevent data reads.
Challenges

- Wear leveling algorithms fragment data on the drive, but in an unpredictable way (non-standard)
- NAND flash technology in SSDs, doesn’t allow for the same forensics tricks to be used as with HDDs
  - Drive has spare blocks which cannot be read. X25 - 7% to 8%. Even more in enterprise version.
  - Mostly no slack space as entire block erased before write
  - Garbage collection clears blocks marked for “deletion”
    - Turn the hard drive off to prevent garbage collection
As soon as you do a write, a block gets allocated and always stays allocated.

When you delete a file, it is only "deleted" in the file system, but the hard drive block stays allocated.

The trim command is proposed so that the hard drive microcontroller knows to also deallocate the block (free it.)
Challenges

- Once you DISCARD/UNMAP sector X, the device can return any state on the next read of that sector, but must continue to return that data until sector is rewritten
- Latest draft:
  - If TPRZ bit set then the return for an unmapped block is always zero.
  - If TPRZ isn't set, it's undefined but consistent.
Challenges

- Hard to read data off chips directly
- Even if you do, hard to make sense of it
- Requires some very sophisticated carving technology
- Needs to be content based
- SmartCarving?
SmartCarving - Advantages

- Recovers multi-fragmented files
- Scales to millions of blocks
- Keys to success:
  - Collation
  - Matching metrics
  - Linear time heuristics for reassembly

For more – www.digital-assembly.com