



# **Empirical Analysis of Solid State Disk Data Retention when used with Contemporary Operating Systems**

**Christopher King (CERT)**

**Tim Vidas (CMU ECE)**

**DFRWS 2011**

**New Orleans, LA**



# Notices

---

© 2011 Carnegie Mellon University

Except for the U.S. government purposes described below, this material SHALL NOT be reproduced or used in any other manner without requesting formal permission from the Software Engineering Institute at [permission@sei.cmu.edu](mailto:permission@sei.cmu.edu).

This material was created in the performance of Federal Government Contract Number FA8721-05-C-0003 with Carnegie Mellon University for the operation of the Software Engineering Institute, a federally funded research and development center. The U.S. Government's rights to use, modify, reproduce, release, perform, display, or disclose this material are restricted by the Rights in Technical Data-Noncommercial Items clauses (DFAR 252-227.7013 and DFAR 252-227.7013 Alternate I) contained in the above identified contract. Any reproduction of this material or portions thereof marked with this legend must also reproduce the disclaimers contained on this slide.

Although the rights granted by contract do not require course attendance to use this material for U.S. Government purposes, the SEI recommends attendance to ensure proper understanding.

THE MATERIAL IS PROVIDED ON AN "AS IS" BASIS, AND CARNEGIE MELLON DISCLAIMS ANY AND ALL WARRANTIES, IMPLIED OR OTHERWISE (INCLUDING, BUT NOT LIMITED TO, WARRANTY OF FITNESS FOR A PARTICULAR PURPOSE, RESULTS OBTAINED FROM USE OF THE MATERIAL, MERCHANTABILITY, AND/OR NON-INFRINGEMENT).

# Outline

---

Introduction

Solid State Disk Background

Forensic Challenges

Testing Methodology

Results

Conclusion

# Introduction

---

Solid State Disks are the future of data storage.

- 15M units projected to be sold in 2011, 118% increase over last year.<sup>1</sup>
- Large increases in Read/Write speed over hard disk drives
  - 20x+ faster read<sup>2</sup>
    - ~170-300MB/s sequential read
    - ~18-77MB/s random read
  - 10x+ faster write<sup>2</sup>
    - ~40-250MB/s sequential write
    - ~5-51MB/s random write
  - Use less energy
  - No moving parts

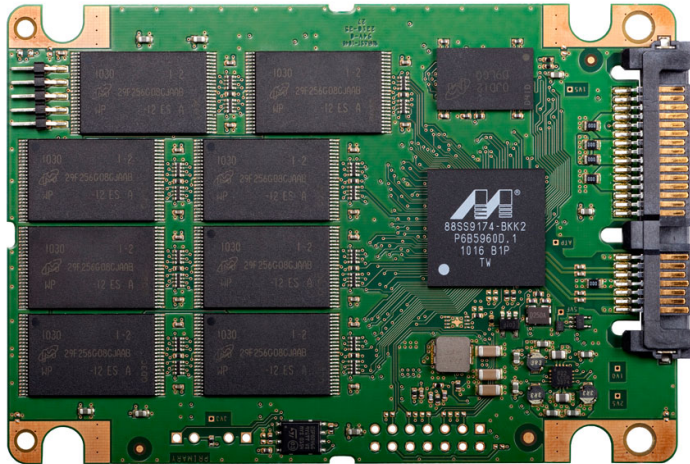
1 - [http://www.isuppli.com/Memory-and-Storage/MarketWatch/Pages/Solid-State-Drives-to-Boom-in-2011-and-Generate-Over-\\$4-Billion.aspx](http://www.isuppli.com/Memory-and-Storage/MarketWatch/Pages/Solid-State-Drives-to-Boom-in-2011-and-Generate-Over-$4-Billion.aspx)

2 - <http://www.anandtech.com/show/2944/1>

# Problem: Data Recovery

---

## Solid State Disk



Credit: Micron Technology, Inc.

## Hard Disk Drive



# Problem: Inconsistent Research Findings

---

- Early research in SSDs has resulted in conflicting or inconsistent results.
  - Data deleted over time (Bell and Boddington 2010)
  - Data stays remnant in the system (Wei et al. 2011)
- Increased need for reproducible results
  - *Bringing science to digital forensics with standardized forensic corpora*  
Simson Garfinkel, Paul Farrella, Vassil Roussevc, George Dinolta

# Prior Work

---

Bell and Boddington - *Solid State Drives: The Beginning of the End for Current Practice in Digital Forensic Recovery?*

- Showed SSD garbage collection was wiping unallocated space after a period of time.

Wei et. al. - *Reliably erasing data from flash-based solid state drives*

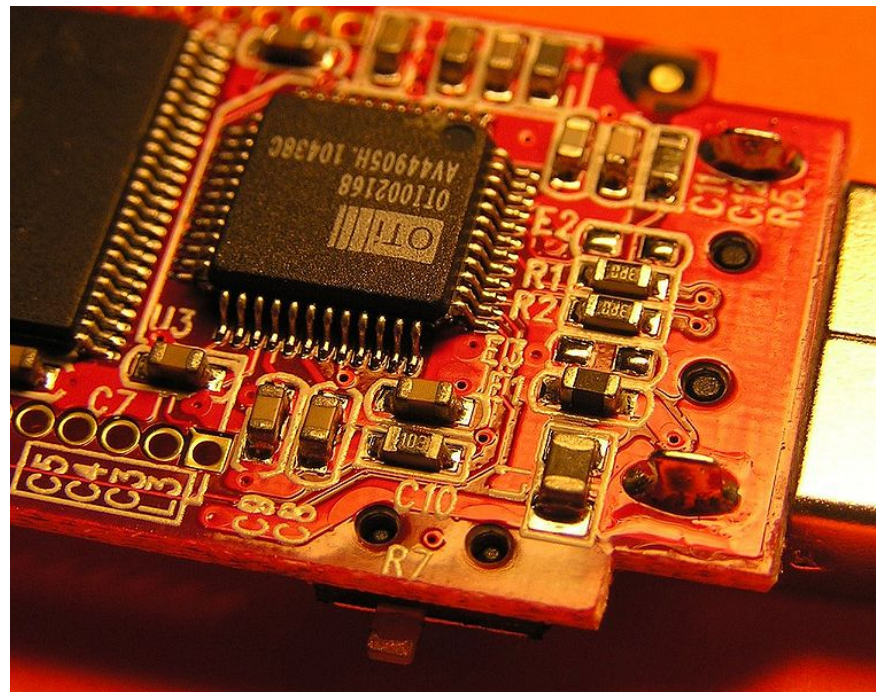
- Some data still resident on SSDs due to overprovisioning.

# SSDs: A Background

---

SSDs are Flash memory based disks, much like thumbdrives.

- They use NAND Flash in their operation.



Credit: John Fader, Creative Commons License

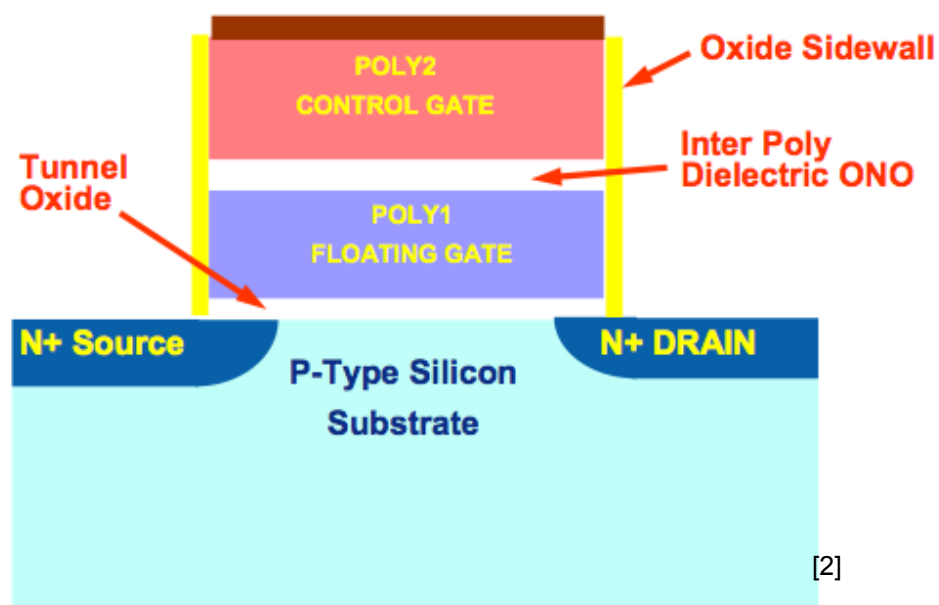
# NAND Flash Cell

A cell is the simplest structure in Flash memory.

When a charge is applied, the electrons tunnel into the cell through the dielectric barrier.

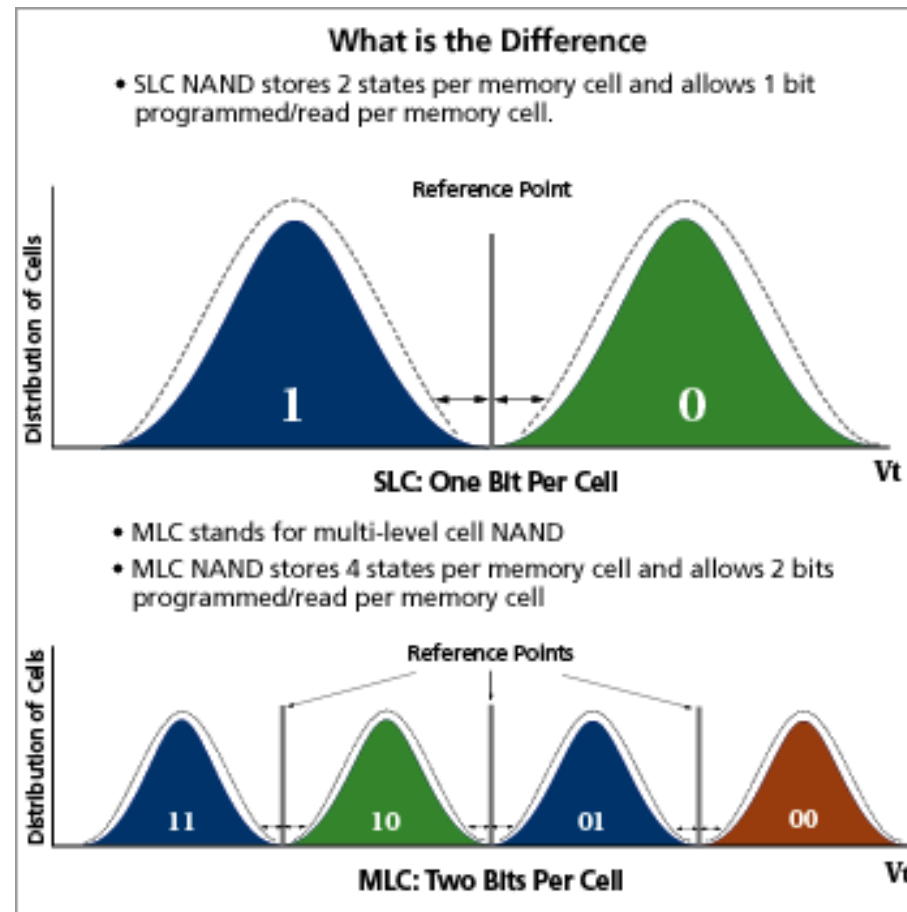
When the charge is stopped the electrons are trapped in the cell.

The resulting positive or negative charge can then be measured.



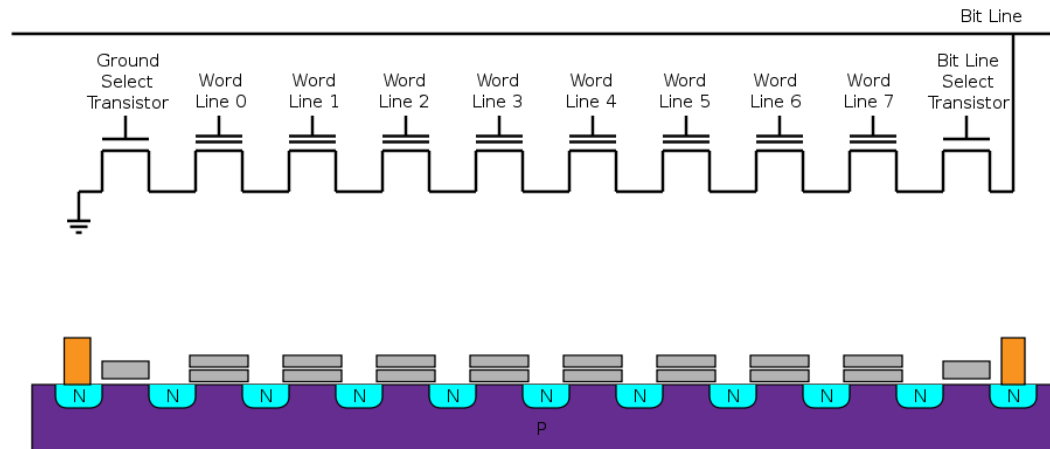
Credit: Anandtech.com

# Single Layer Cell vs. Multi Layer Cell



Credit: Micron

# NAND Flash page



Credit: Cyferz, Wikipedia Creative Commons License

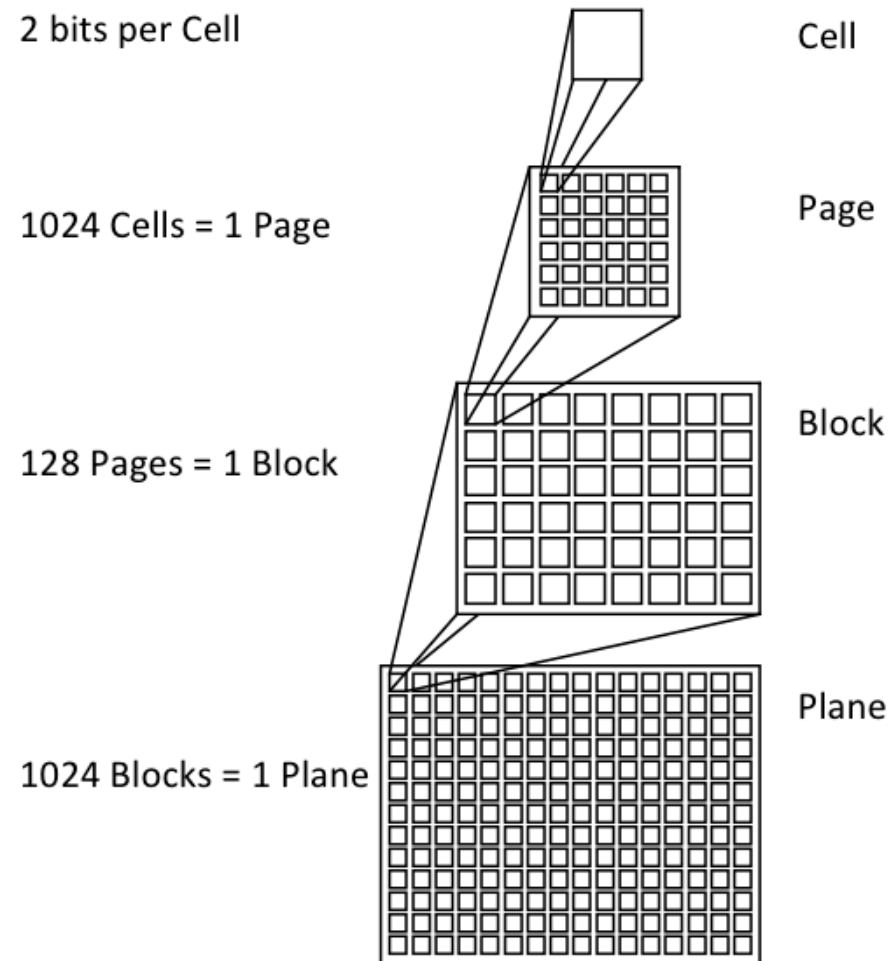
Series of floating gates and cells all connected.

A write occurs starting from the source, and writing down the entire page.

A read requires a measurement of the sink, which totals up the values of the floating gates.

# SSD Design

---



# Example: 4GB MLC Flash

---

Cell - 2 bits

Page – 4KB (8,192 cells)

Block - 256KB (64 pages)

Plane – 524MB (2048 blocks)

Chip/Die – 2GB (4 planes)

Drive – 4GB (2 chips/dies)

# Implementation Challenges

---

## Flash Cell Wear

- Maximum of 100,000 writes per cell.

## Delete-before-write

- Flash deletes in blocks of 128KB.
- Expensive operation
- Write amplification

## Legacy compatibility

- Drives must be compatible with major OS's that do not understand Flash file systems.

# Implementation Challenges - Solutions

---

## Flash Cell Wear

- Wear leveling

## Delete-before-write

- Garbage collection
- Overprovisioning
- TRIM

## Legacy compatibility

- Flash Transition Layer

# Review: Data Deletion on HDDs

---

The OS stores files in clusters, tracking the location of these clusters using a table with a pointer.

- ie) Master File Table (MFT) in NTFS, inodes in EXT2/3

When a file is deleted, the OS deletes the pointer in the table, marking that space available.

Until the OS uses those blocks again, the data remains on the disk.

- This data can be recovered using a number of tools.

# Data Deletion on SSDs

---

When a file is deleted, the pointer in the MFT is deleted and the OS sees that area as free.

- The SSD marks the data as invalid. The data stays where it is until that block needs to be overwritten or if the controller needs more blocks.

## Garbage Collection.

- The SSD keeps a number of “active” blocks available for writes (allocation pool).
- When this number falls below a certain threshold, the drive initiates block cleaning (or garbage collection).
- The SSD chooses the block with the highest ratio of invalid pages to total pages.
- The SSD then copies over any non-deleted data, deletes the pages, and puts the now clean block back into the pool.
- **A delete in NAND flash sets all the bits to 1.**

# Data Deletion on SSDs: TRIM

---

## TRIM Command

- Added to ATA8 specification to solve performance problems associated with garbage collection.
- OS tells disk what blocks are free.

## TRIM operation

- File is deleted
- The OS sends a TRIM command to the SSD with the LBA location of the deleted blocks.
- The SSD maps this to its own FTL blocks.
- It then copies any valid pages to memory and deletes the block.
- The block is rewritten with the valid data.
- This can happen immediately after delete instead of when blocks are needed.
- The drive controller software decides when to erase the blocks it knows are now “free.”

# Forensic Challenges

---

The sector the OS reports is not the physical location of the data.

Data deleted is gone forever (in some cases).

Garbage collection implementation varies among manufacturers.

# Research Questions

---

How will forensics investigators know if data can be recovered?

- Are these disks even similar enough that data CAN be recovered?

Are some SSDs better than others at data retention?

- Performance varies greatly between controller manufacturers – does this apply to data retention?

How does TRIM affect data recovery?

- This new ATA command seems to effect deletion of data.

How does the Operating System affect data recovery?

- Windows 7 was optimized for SSDs – what does that mean?

# Test Methodology

---

## Test Cases:

- Test 1: Data recovery from low-usage disk
- Test 2: Data recovery from high-usage disk
- Test 3: Data recovery from OS formatted disk

## Each test was tested using:

- Windows 7 Enterprise
- Windows XP Professional
- Ubuntu Linux Desktop 9.04

Data recovery was done using Backtrack 4, Sleuthkit, and custom scripts to compare individual blocks to reference files.

# Test Methodology

---

To ensure proper comparisons over many tests:

- Each drive was secure wiped before installing OS
- The OS was installed on to the disk
- Two reference files, a small 1 MB text file and a large, 650MB binary file were copied to the disk
- In the high usage test, the disk was filled with random files to simulate a filled disk.
- The starting and ending block locations of each file were extracted using Sleuthkit.
- The files were deleted normally and recovery attempted using a custom script to byte-wise compare the blocks.

# Test Methodology (cont.)

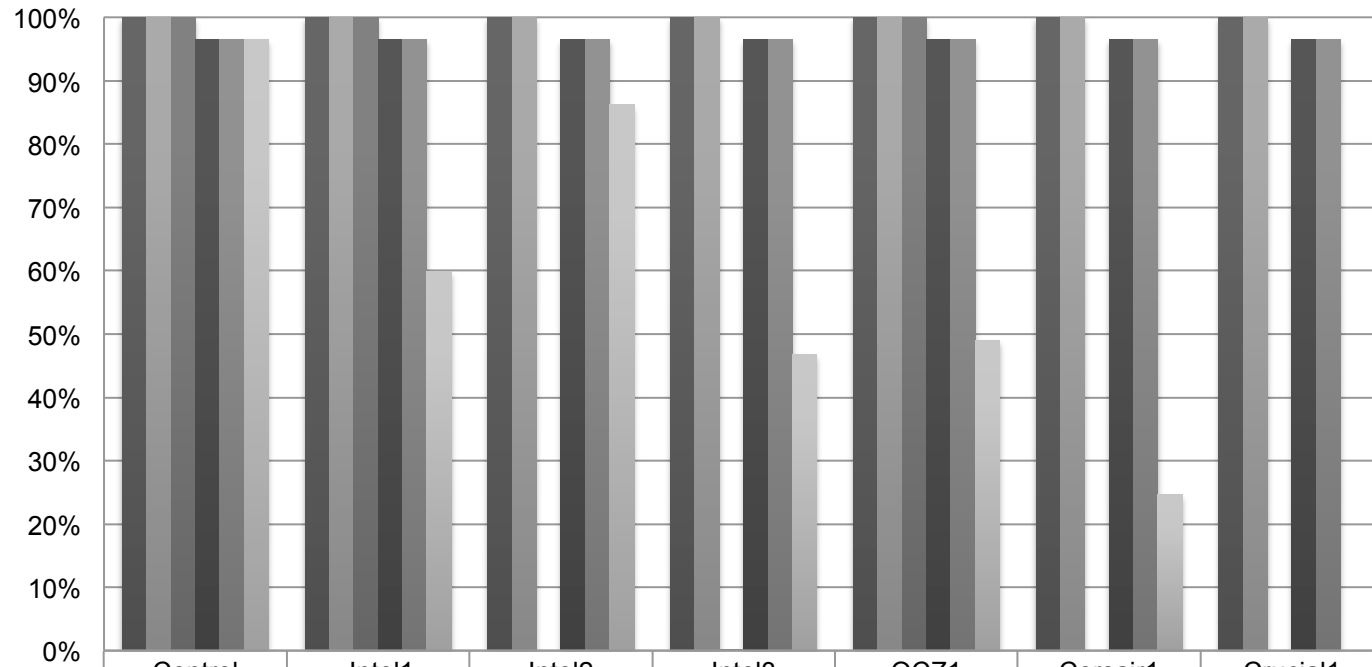
---

## Drive statistics

- 16 SSDs tested (1 HDD control)
- 10 different manufacturers
- 5 drives supporting TRIM
- 144 tests conducted

# Results: Drives without TRIM

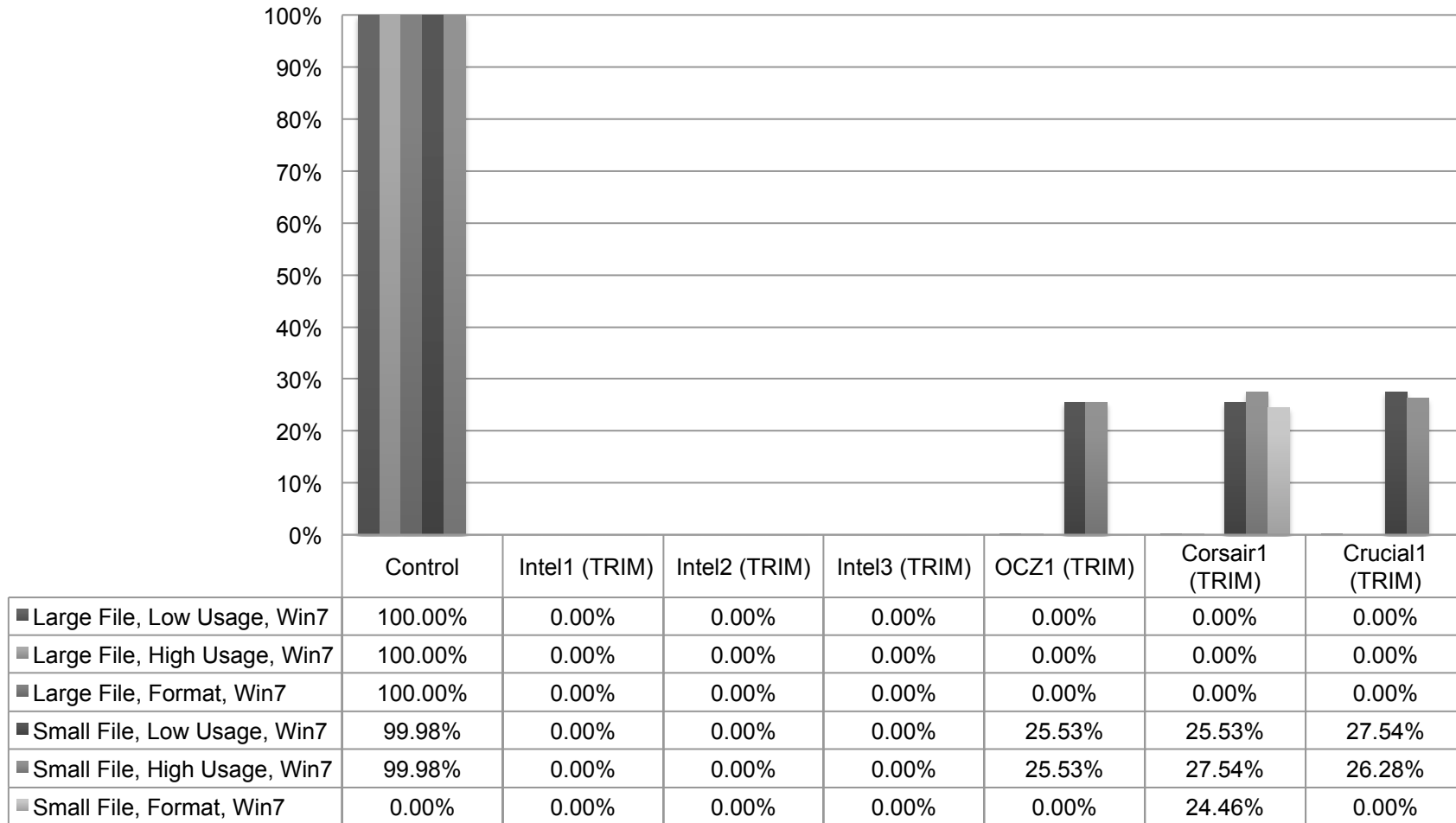
## Percent Blocks Recovered on Drives without TRIM



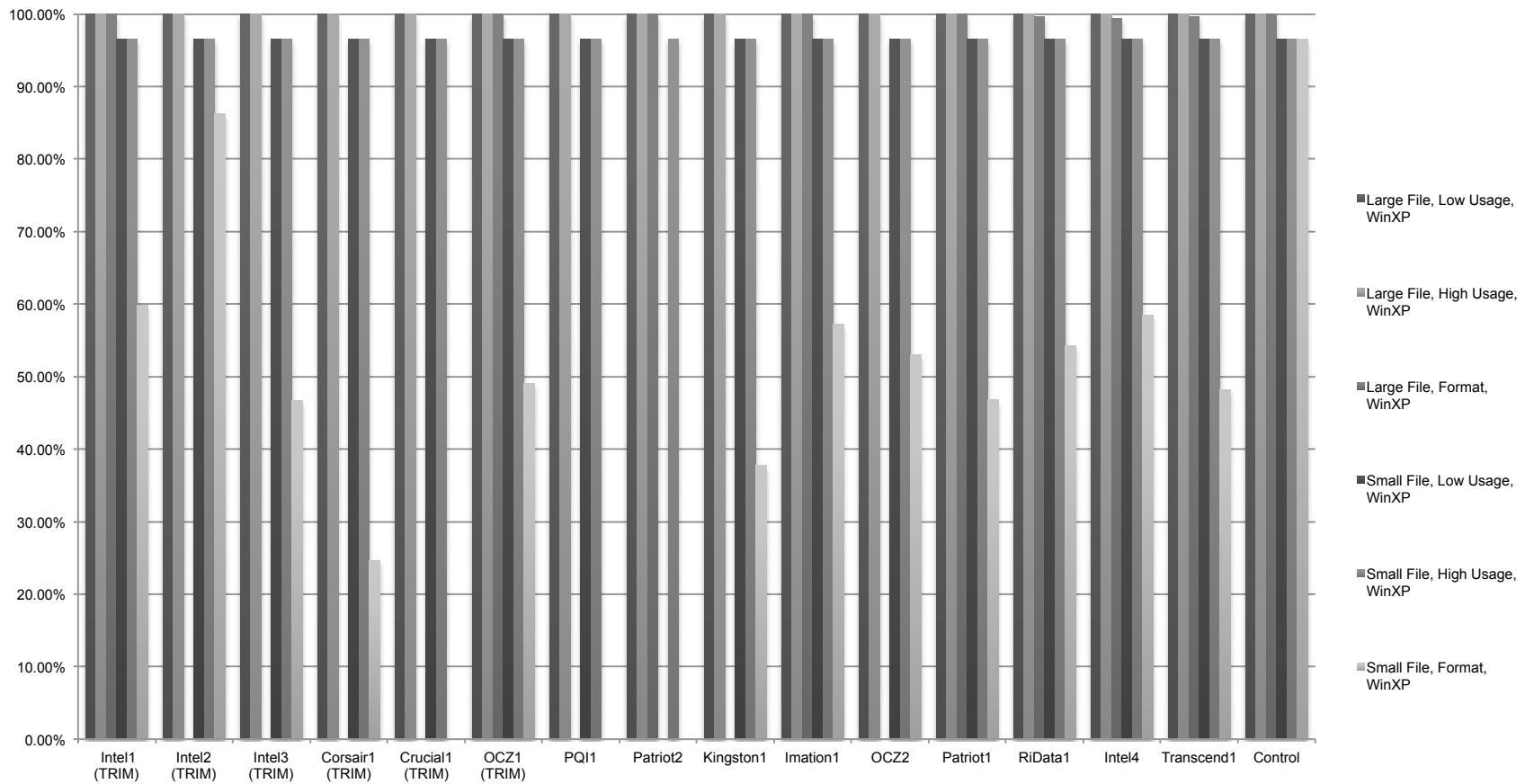
	Control	Intel1	Intel2	Intel3	OCZ1	Corsair1	Crucial1
■ Large File, Low Usage, WinXP	99.99%	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%
■ Large File, High Usage, WinXP	99.99%	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%
■ Large File, Format, WinXP	99.99%	100.00%	0.00%	0.13%	100.00%	0.00%	0.00%
■ Small File, Low Usage, WinXP	96.57%	96.57%	96.57%	96.57%	96.57%	96.57%	96.57%
■ Small File, High Usage, WinXP	96.57%	96.57%	96.57%	96.57%	96.57%	96.57%	96.57%
■ Small File, Format, WinXP	96.57%	59.84%	86.20%	46.76%	49.03%	24.69%	0.00%

# Results: Drives with TRIM

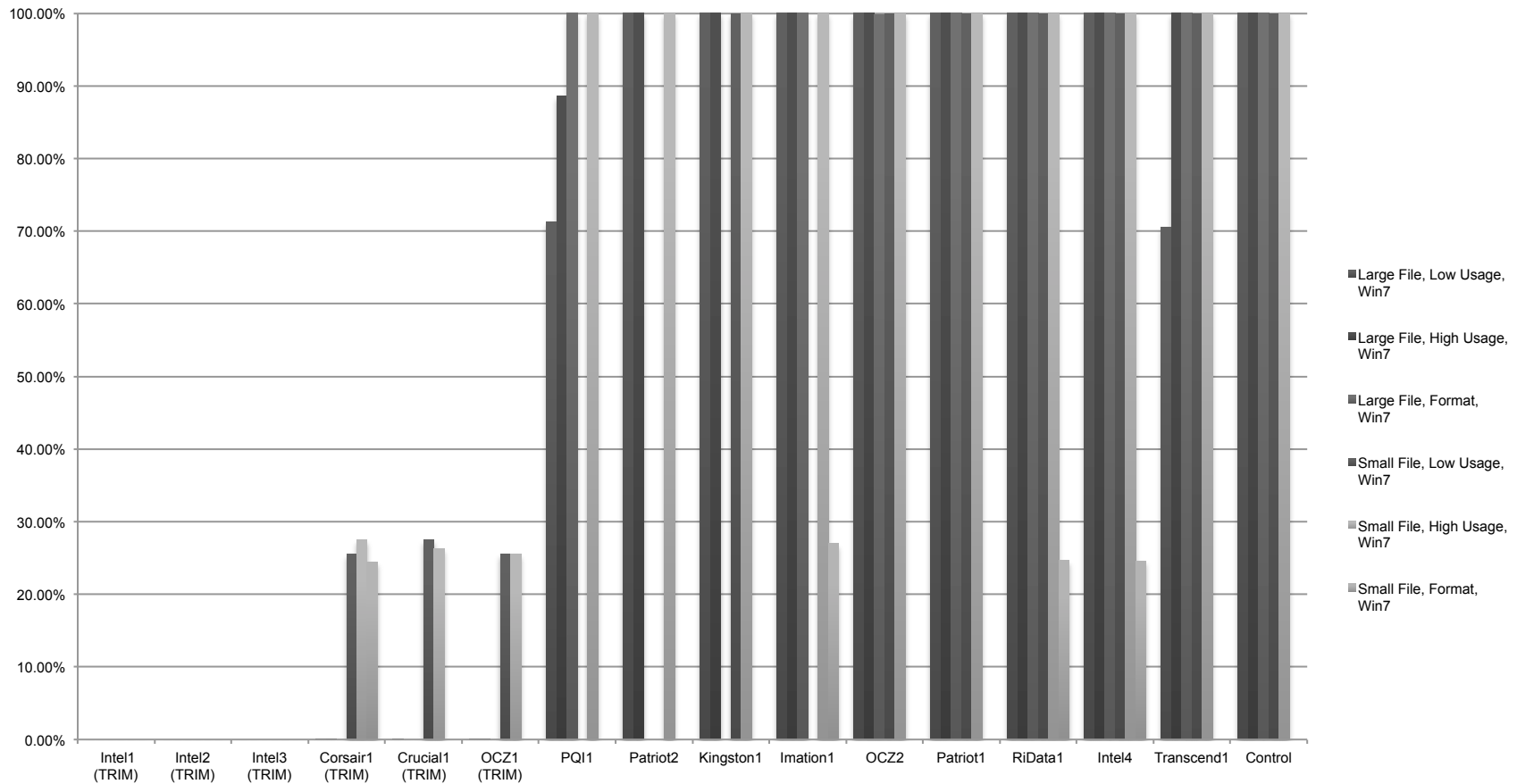
## Percent Blocks Recovered on Drives using TRIM



# Percent blocks recovered in Windows XP across all tests



# Percent blocks recovered in Windows 7 across all tests



# Findings

---

Data deleted on a SSD that supports TRIM, on a TRIM-enabled OS, is seemingly impossible to recover.

High usage disks have a greater chance of data retention over low usage disks.

- Windows 7: up to 17% more than low usage, 100% more with small files

Formatted disks using default settings result in partial data loss for all OS's and full data loss for Windows 7 with TRIM.

Manufacturer variance in garbage collection seems to change the recoverability of files on the drive.

# Manual TRIM Test

---

SSDs can be used for anti-forensics. A single click can tell the drive to TRIM itself, destroying evidence.

- A modified hdparm script caused two test SSDs to perform the equivalent of a secure wipe.
- Roughly 10 seconds to perform zeroing of drive. Further modifications to the code could reduce the time even further.
- Running drive was also TRIMed – drive remained operational until reboot, in which drive was fully zeroed.

# Implications

---

SSDs should not be considered the same as an HDD when encountered in the field.

- The manufacturer and model of a drive is an important consideration.
- Most new SSDs are being built with TRIM support.

Investigators should not expect substantial results with data recovery on drives supporting TRIM and using TRIM-enabled OS's.

- The OS type is very important for making a decision on whether to perform static or volatile analysis.

Drive manufacturer and firmware does matter for recoverability.

# Future Research

---

When do different OS's send TRIM commands?

Data recovery from Linux ext4 and TRIM-supported kernels.

Data recovery from SSDs in OSX.

Analysis of newer firmware and drives (Sandforce)

More comprehensive temporal analysis



**Questions?**

