Physical NAND Flash Security: Preventing Recovery of Deleted Data

Michael Abraham
(mabraham@micron.com)
NAND Solutions Group Architect
Micron Technology, Inc.

Topics

- A brief history of data removal
- Data sanitization methods
- NAND Flash physical requirements
- Block management
- Data overwriting
- Secure erase
- Removing residual data
- Myths

A Brief History of Data Removal

- In early storage protocols, functions existed to read and write data, but not to remove it
- Deleted data typically remains on storage media after it is no longer needed
  - The file's index record is partially overwritten
  - The file's data is not overwritten
  - Whole industry exists to recover deleted files or to protect against accidental deletion
- Volume/partition reformatting and deletion also does not remove previous data
- Focus of these operating systems is not security, but rather to protect the user from unintentionally removing wanted data
- Data sanitization focuses on removing data permanently from the storage media

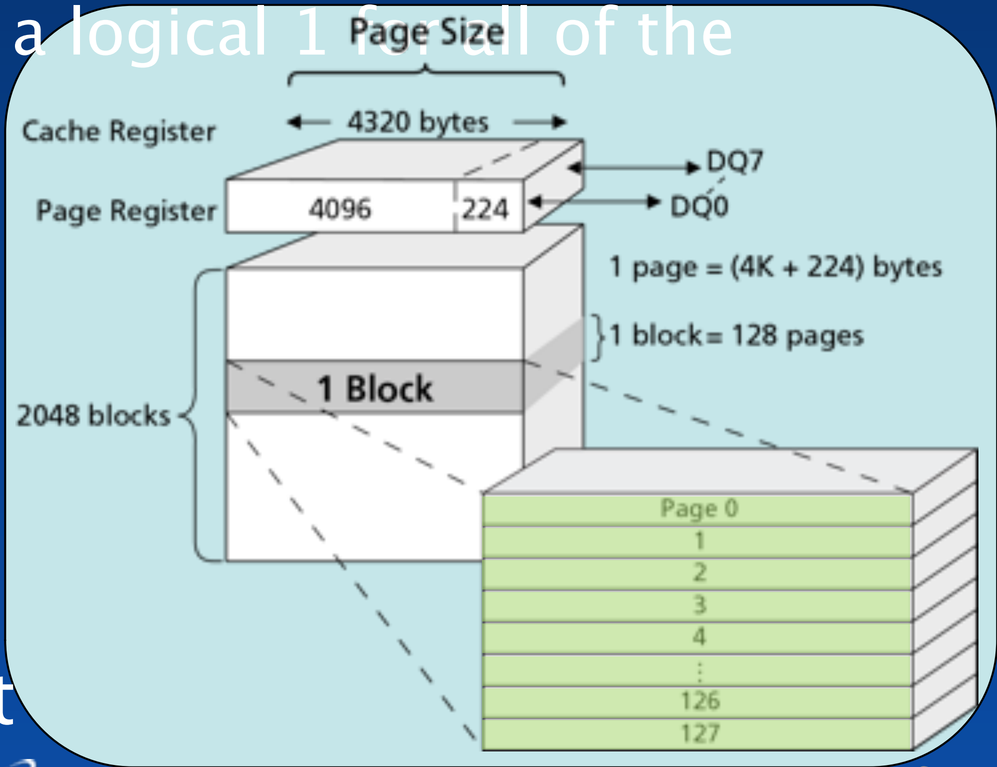
Data Sanitization Methods

- Clearing
  - Previous data may still be recoverable through laboratory attack
  - Data overwriting
  - Media is reusable
- Purging
  - Prevents laboratory attack to recover data
  - SECURE ERASE (for ATA disks)
  - Media may or may not be reusable
- Physical destruction
  - Disintegration, incineration, pulverization, melting, and shredding
  - Media is no longer reusable
- How do these methods relate to NAND Flash-based storage?

Source: NIST 800-88, Table 2-1, p. 8

NAND Flash Physical Requirements

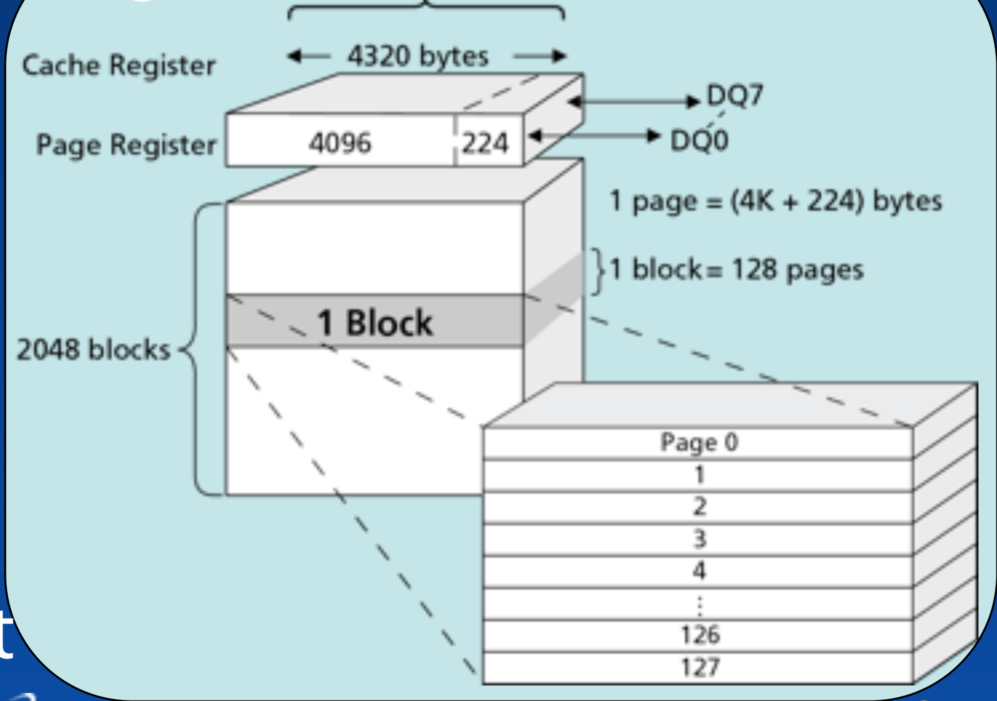
- NAND Flash requires block management
  - Erase blocks of data, consisting of multiple pages
  - Changes a logical 0 to a logical 1 for all of the cells in the block
  - Program pages within a block in sequential order
  - Changes a logical 1 to a logical 0
- When reusing a page, it must be erased first



Monday, August 15, 2011

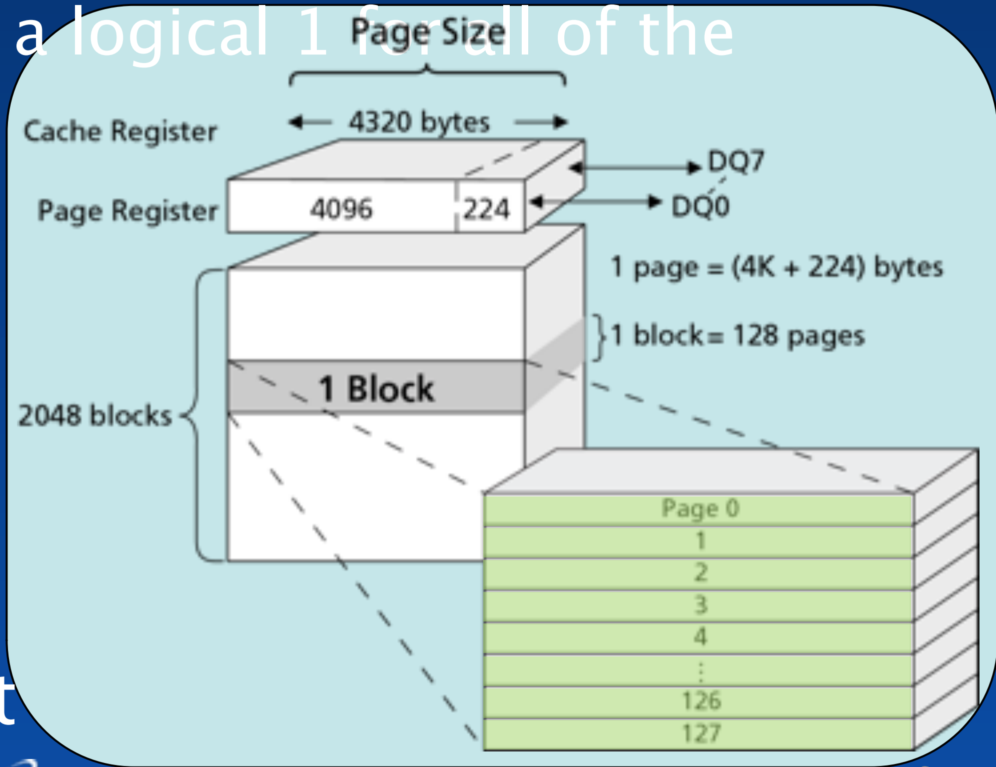
NAND Flash Physical Requirements

- NAND Flash requires block management
  - Erase blocks of data, consisting of multiple pages
  - Changes a logical 0 to a logical 1 for all of the cells in the block
  - Program pages within a block in sequential order
  - Changes a logical 1 to a logical 0
- When reusing a page, it must be erased first



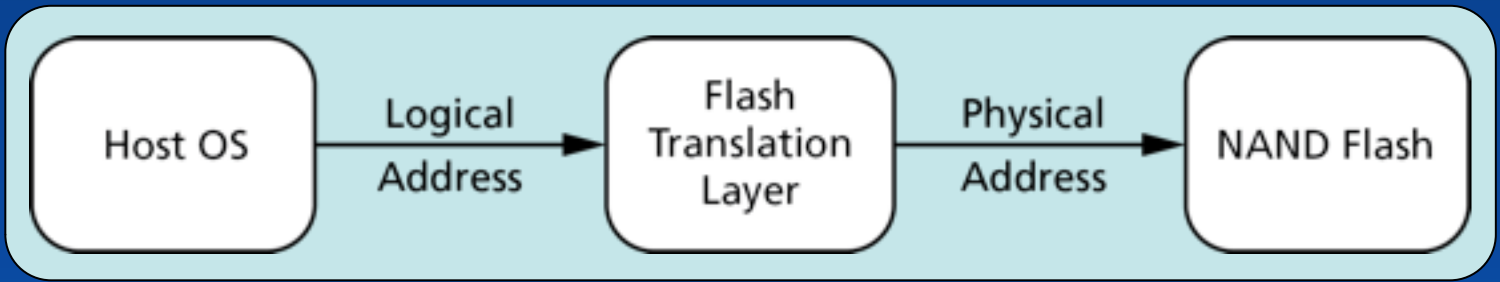
NAND Flash Physical Requirements

- NAND Flash requires block management
  - Erase blocks of data, consisting of multiple pages
  - Changes a logical 0 to a logical 1 for all of the cells in the block
  - Program pages within a block in sequential order
  - Changes a logical 1 to a logical 0
- When reusing a page, it must be erased first



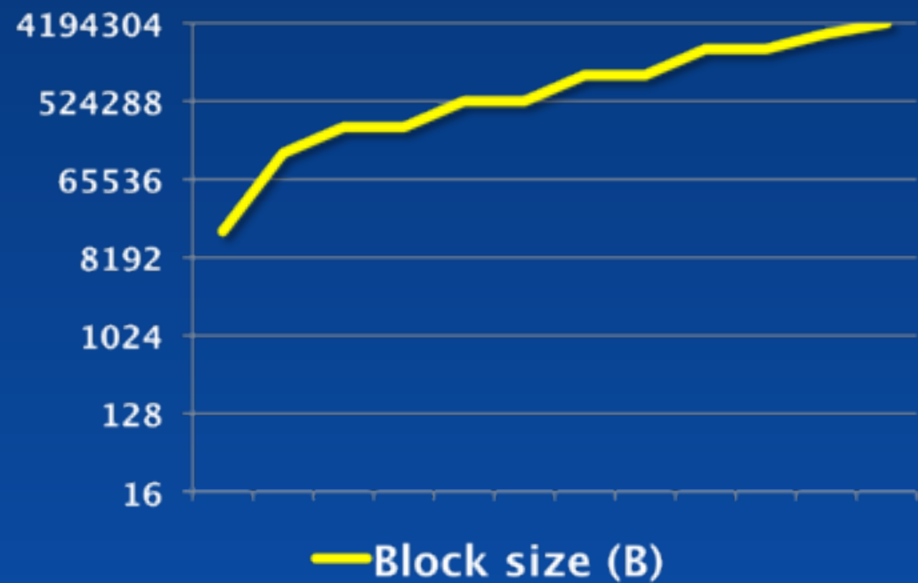
Block Management

- Block management is used to make sure NAND physical requirements are met
- The Flash translation layer translates host addresses (logical) to NAND addresses (physical)
  - Software running on a host operating system
  - Firmware running on a NAND controller



Block Management Today

- Block management algorithms have become more complex as block sizes have increased
  - Typical data sizes are still 512B or 4KB
  - Page sizes are moving to 8KB or 16KB for SSDs
  - Block management algorithms are optimized for sequential throughput and IOPS
- More data fragments remain on the media through use



Monday, August 15, 2011

Data Overwriting

- For hard disk drives (HDDs), the primary method to clear individual files was through data overwriting
- Sensitive data is overwritten with one or more data patterns to remove it
- Residual data may still be recoverable if overwritten only once (though unlikely with today's HDDs)
- Data overwriting applications/algorithms were developed around direct addressing—an LBA points to the same physical location for every write
- Securely deleting a file is typically a tradeoff of thoroughness (number of passes) and performance (time)

Data Overwriting with NAND Flash

- Because of block management, data overwriting is not effective to clear individual LBAs of data
- NAND Flash uses indirect addressing—an LBA points to a different physical location for every write
- This results in multiple copies of the data existing
  - Current version
  - Older versions
- Older versions are eventually discarded when the blocks they reside in no longer have valid data in them and are erased
- Data overwriting is mostly effective if the entire drive is overwritten, but some previous data typically remains in hidden blocks and can be recovered through laboratory attack
  - Typically less sophisticated to recover than an HDD data overwrite
  - Slow
  - Risk is proportional to the amount of overprovisioning

Secure Erase

- Secure erase was developed to permanently purge <u>all data</u> from HDDs, including areas not directly addressable
- Secure erase implemented on many SSDs
- e·MMC has adapted secure erase to purge data from portions of the memory device

Secure Erase and Block Management

- If purging a portion of the media, a host-issued SECURE ERASE command would be responsible for the following NAND functions
  - Identify all physical copies of data (current and previous) representing a host LBA or LBA range
  - Copy/move all good, valid data around the data to be purged to new locations
  - Properly erase the blocks

- Time is required to perform the block management function of identifying all copies of a particular LBA and consolidating valid data

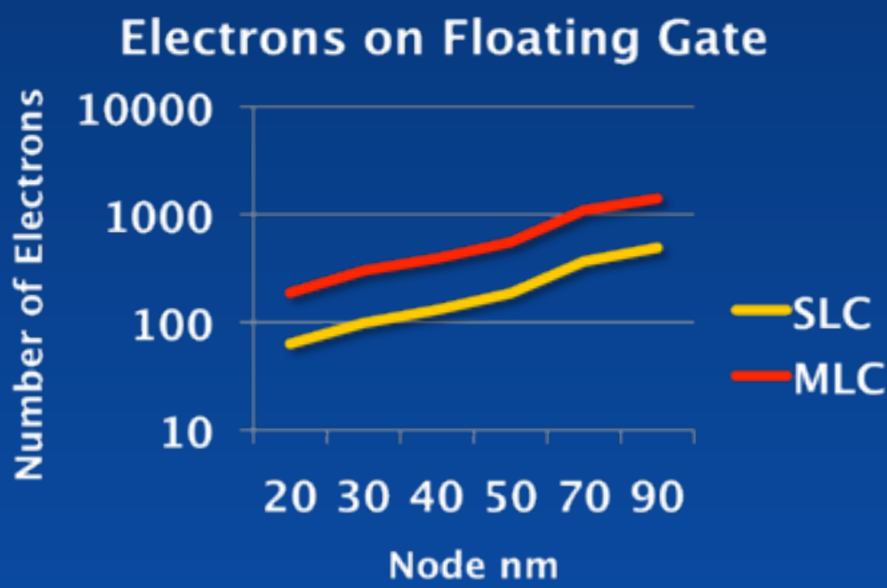
- If purging all drive data, then SECURE ERASE is fairly quick because no block consolidation needs to occur

Removing Residual Data

- Can data be purged from the NAND Flash to prevent recovery through a laboratory attack?
- This requires a better understanding of the NAND PAGE PROGRAM and BLOCK ERASE commands

NAND Flash Cell Fundamentals

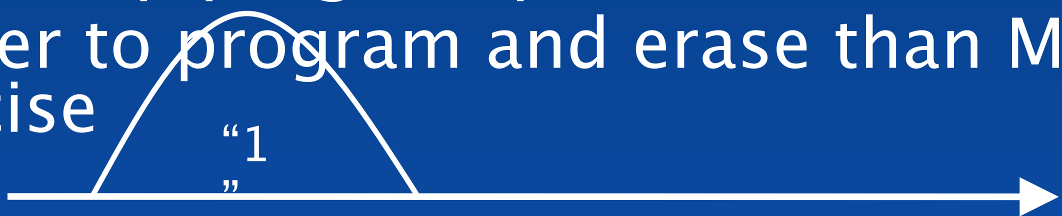
- 1s and 0s are represented by the number of electrons stored on the NAND floating gate
- Working number of electrons on each floating gate is decreasing as NAND process shrinks
- Becomes harder to discern bit states



Source: Micron Technology, Inc.

Single-Level Cell (SLC) NAND Flash

- Each NAND cell is mapped to one NAND page
- Needs fewer working electrons than MLC to distinguish bit states
- Programming increases effective voltage on cells
- Only two states represented: 1, 0
- Bits states use wider cell distributions than MLC
- One-step program process
- Faster to program and erase than MLC—less precise

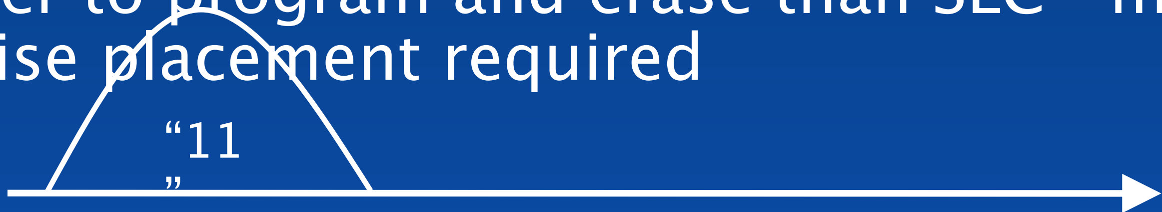


Single-Level Cell (SLC) NAND Flash

- Each NAND cell is mapped to one NAND page
- Needs fewer working electrons than MLC to distinguish bit states
- Programming increases effective voltage on cells
- Only two states represented: 1, 0
- Bits states use wider cell distributions than MLC
- One-step program process
- Faster to program and erase than MLC—less precise

2-Bit Multiple-level Cell (MLC) NAND Flash

- Each NAND cell is mapped to two NAND pages
- Needs more working electrons than SLC to distinguish bit states
- Programming increases effective voltage on cells
- Four states represented: 11, 10, 00, 01
- Bits states use narrower cell distributions than SLC
- Two-step program process: fast page then slow page
- Slower to program and erase than SLC—more precise placement required

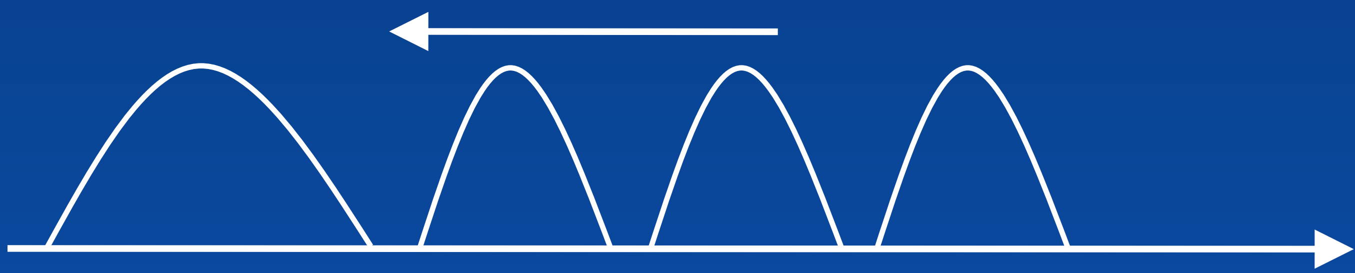


2-Bit Multiple-level Cell (MLC) NAND Flash

- Each NAND cell is mapped to two NAND pages
- Needs more working electrons than SLC to distinguish bit states
- Programming increases effective voltage on cells
- Four states represented: 11, 10, 00, 01
- Bits states use narrower cell distributions than SLC
- Two-step program process: fast page then slow page
- Slower to program and erase than SLC—more precise placement required

Goals of a BLOCK ERASE Operation

- Decrease effective voltage on all cells by removing electrons from the floating gate
- Increase effective voltages on deeply erased cells
  - Tightens erase distribution
  - Significantly reduces possibility of data recovery through laboratory attack



Monday, August 15, 2011

Goals of a BLOCK ERASE Operation

- Decrease effective voltage on all cells by removing electrons from the floating gate
- Increase effective voltages on deeply erased cells
  - Tightens erase distribution
  - Significantly reduces possibility of data recovery through laboratory attack

Goals of a BLOCK ERASE Operation

- Decrease effective voltage on all cells by removing electrons from the floating gate
- Increase effective voltages on deeply erased cells
  - Tightens erase distribution
  - Significantly reduces possibility of data recovery through laboratory attack

MLC NAND Flash Block Erase Algorithm

- MLC NAND Flash block erase algorithms already prevent laboratory attack, especially on the latest process nodes
  - Pre-erase data compaction brings all cells in the block to a close level
  - Post-erase data compaction adds electrons to deeply erased cells

- Does not require host involvement and occurs during the typical ^{t}BERS

- Note: Not all NAND vendors erase MLC NAND Flash with the same algorithms

SLC NAND Flash Block Erase Algorithm

- The SLC block erase algorithm is typically shorter than the MLC block erase, though moving toward MLC algorithms
- The typical SLC block erase for 20-30nm process nodes should adequately purge the cells within a NAND block
- For older NAND technology, the host may need to assist the NAND in purging residual data
  - Erase the block (optional)
  - Program all of the pages in the block to solid zeros
  - Erase the block
  - Can add an additional ~40ms to erase time

Myths: TRIM or "Super Voltage"

- TRIM is a command tells a drive which LBAs are no longer needed on a drive
  - Drives can discard unneeded data during block management
    - Improves performance
    - Reduces write amplification
  - Method of implementation on the NAND physical level is controller/firmware specific
  - TRIM does not guarantee old data removal because of its indeterminate nature
- Occasionally I get questions on destroying NAND Flash using a "Super Voltage"
  - Not guaranteed to destroy NAND Flash or to eliminate data in the array

Sanitization Summary for NAND Flash

- For sanitization of all data
  - Data overwriting clears data, but some data may remain in NAND blocks used for overprovisioning
  - Secure erase (if implemented at host interface) purges data from all NAND blocks with user data

- For sanitization of individual files
  - Data overwriting is ineffective—very likely to keep previous copies of data to be removed, especially if the drive is not already full
  - Secure erase or secure delete only works if block management consolidates good data from data to be removed and uses block erase to purge the old data

- Effectiveness of data purging is also dependent on the effectiveness of the BLOCK ERASE command
  - Latest 20-30nm SLC and MLC process nodes sufficiently purge data
  - Laboratory attack may be possible on older NAND process nodes

Questions?

Revisit the Micron FMS presentations at www.micron.com/fms

References

- Caulfield, Adrian, et al, "Secure Erase of Flash Memory," speech given at Flash Memory Summit, Santa Clara, California, August 11, 2009, http://www.flashmemorysummit.com/English/Collaterals/Proceedings/2009/20090811_F1A_Caulfield.pdf, accessed on December 16, 2009.

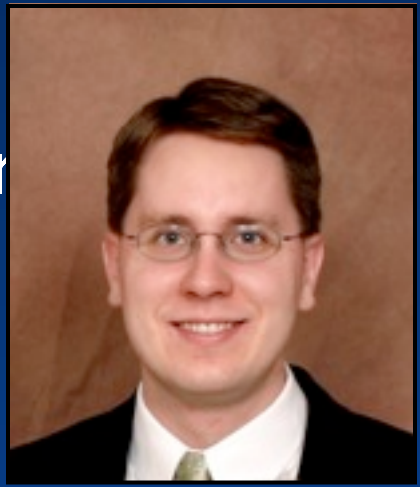
- De Nardi, Christophe, et al, "Direct Measurements of Charge in Floating Gate Transistor Channels of Flash Memories Using Scanning Capacitance Microscopy," paper presented at 32nd International Symposium for Testing and Failure Analysis, November 2006, http://www.multiprobe.com/technology/technologyassets/S05_1_direct_measurements_of_charge_in_floating_gate.pdf, accessed on March 16, 2010.

- Hughes, Gordon, "CMRR-Secure Erase," http://cmrr.ucsd.edu/people/Hughes/SecureErase.shtml, accessed on July 14, 2011.

- JEDEC Solid State Technology Association, "Embedded MultiMediaCard(e·MMC) e·MMC/Card Product Standard, High Capacity, including Reliable Write, Boot, Sleep Modes, Dual Data Rate, Multiple Partitions Supports, Security Enhancement, Background Operation and High Priority Interrupt (MMCA, 4.41)," JEDEC Standard No. 84-A441, March 2010, http://www.jedec.org/standards-documents/docs/jesd84-a441, accessed on July 14, 2011.

- Kissell, Richard, et al, "Guidelines for Media Sanitization: Recommendations of the

About Michael Abraham



- Architect in the NAND Solutions Group at Micron
- Covers advanced NAND and PCM interfaces and system solutions
- BS degree in Computer Engineering from Brigham Young University