



NAND Flash Reliability and Performance The Software Effect

Wes Prouty (waprouty@micron.com)

Applications Engineering Manager

Micron Technology, Inc.

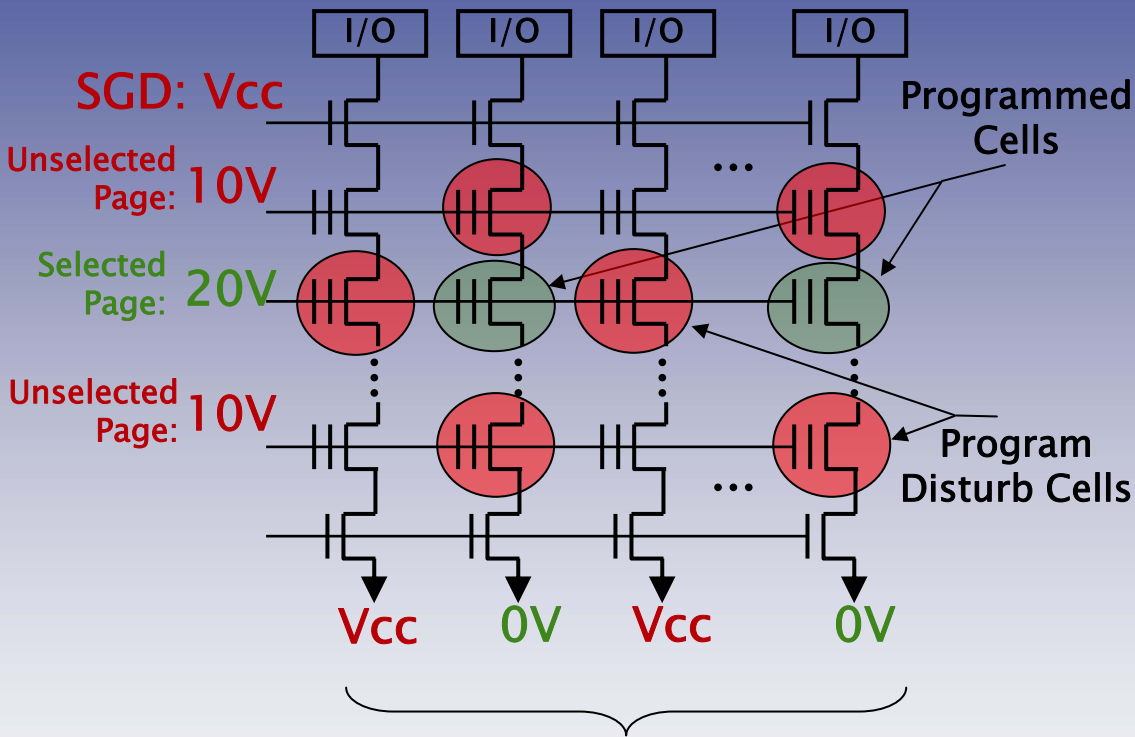
- Why change NAND Flash software?
- Review of NAND Flash error modes
- Review of NAND Flash performance enhancements
- Status of NAND software in embedded operating systems
- NAND Flash software stack variations
- How is software affected by MLC NAND?
- Is software the only solution?

Why Change NAND Flash Software?

- Embedded space has predominately used small-page, single-level cell NAND Flash
- Code storage is moving to NAND Flash in many applications
- As NAND Flash geometries shrink, software usage models must become more sensitive to NAND error modes
- New NAND devices include advanced performance features
- MLC NAND!

Program Disturb

- Stressed cells are limited to the block being programmed
- However, stressed cells can be in selected or unselected page
- Disturb occurs when charge collects on the floating gate, causing the cell to appear weakly programmed
- Does not damage cells; ERASE returns cells to undisturbed levels

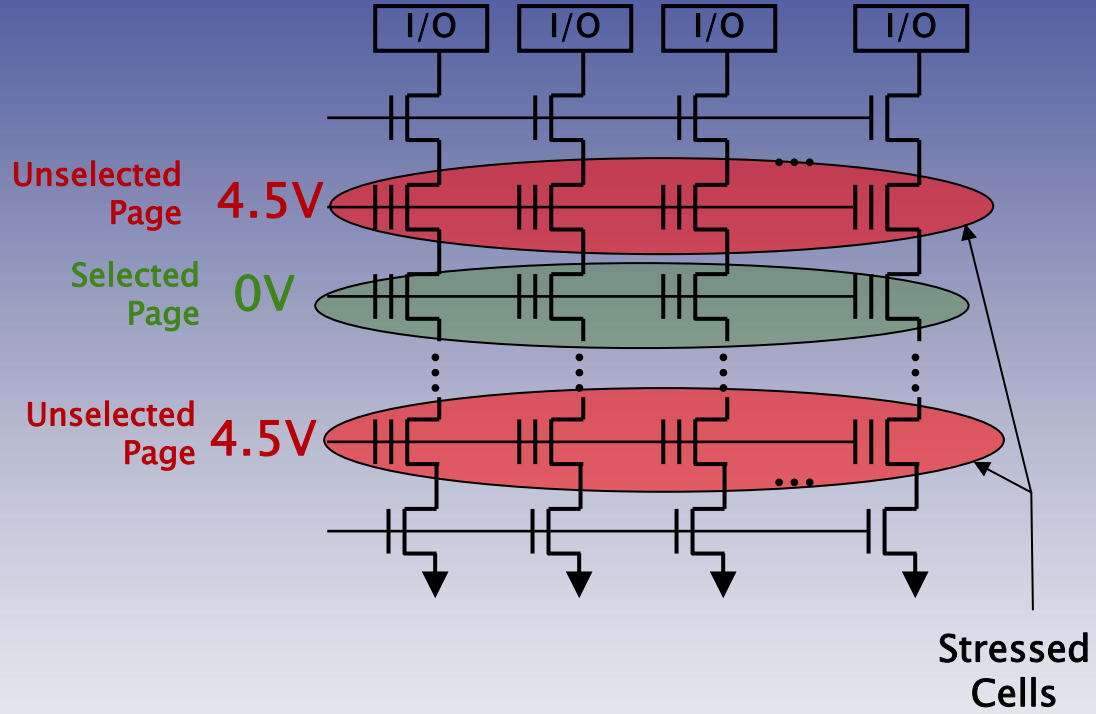


Strings being programmed are grounded. Others are at Vcc.

Note: Circuit structures and voltages are representative only. Details vary by manufacturer and technology node.

Read Disturb

- Stressed cells limited to the block being programmed
- Stressed cells are in unselected page
- Disturb occurs when charge collects on the floating gate, causing the cell to appear weakly programmed
- Does not damage cells; ERASE returns cells to undisturbed levels



Note: Circuit structures and voltages are representative only. Details vary by manufacturer and technology node

- Endurance – PROGRAM/ERASE cycles may cause charge to be trapped in the dielectric. This is a failed block that should be retired.
- Data Retention – Charge loss/gain occurs on the floating gate over time. Cell is undamaged, block can be reliably erased and reprogrammed.

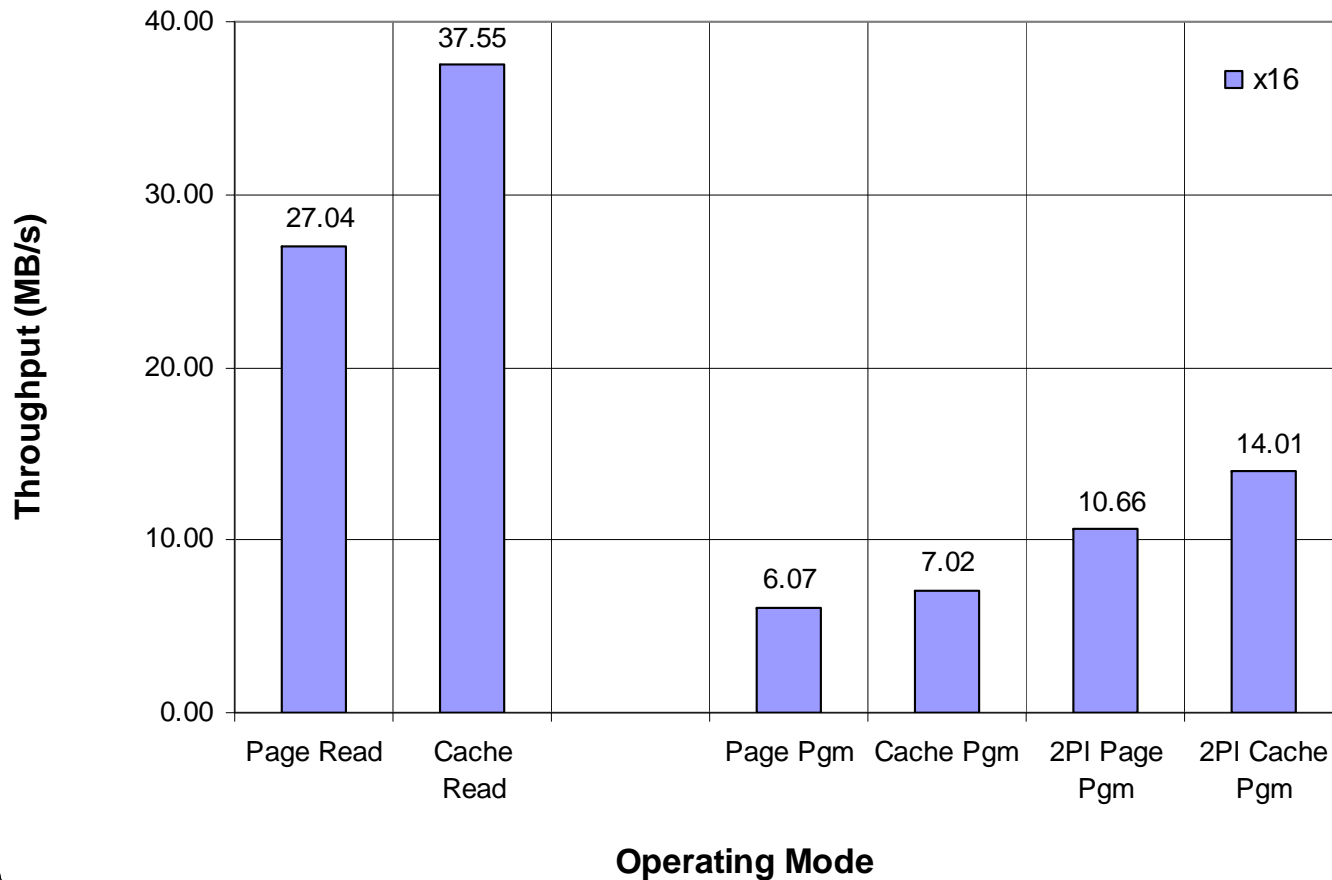
Reducing NAND Error Modes

- Program Disturb
 - Reduce partial page programming (NOP)
 - Programming the pages of each block in sequential order
- Read Disturb
 - Frequently read data should be cached to DRAM
 - Refresh data when exceeding read disturb recommendations
- Endurance and data retention
 - Retire bad blocks
 - Wear level across all blocks (code and data)
- Use ECC to manage disturbed bits!

- Embedded systems are primarily using 512-byte NAND Flash, which has lower performance than 2KB NAND Flash
- Those systems using 2KB NAND Flash only implement the “basic” commands
- NAND Flash enhanced feature set greatly improves NAND Flash performance:
 - Read performance can be increased by over 30%
 - Program performance can be increased by over 100%!

NAND Flash Performance

4Gb NAND (M50, 55nm) Performance

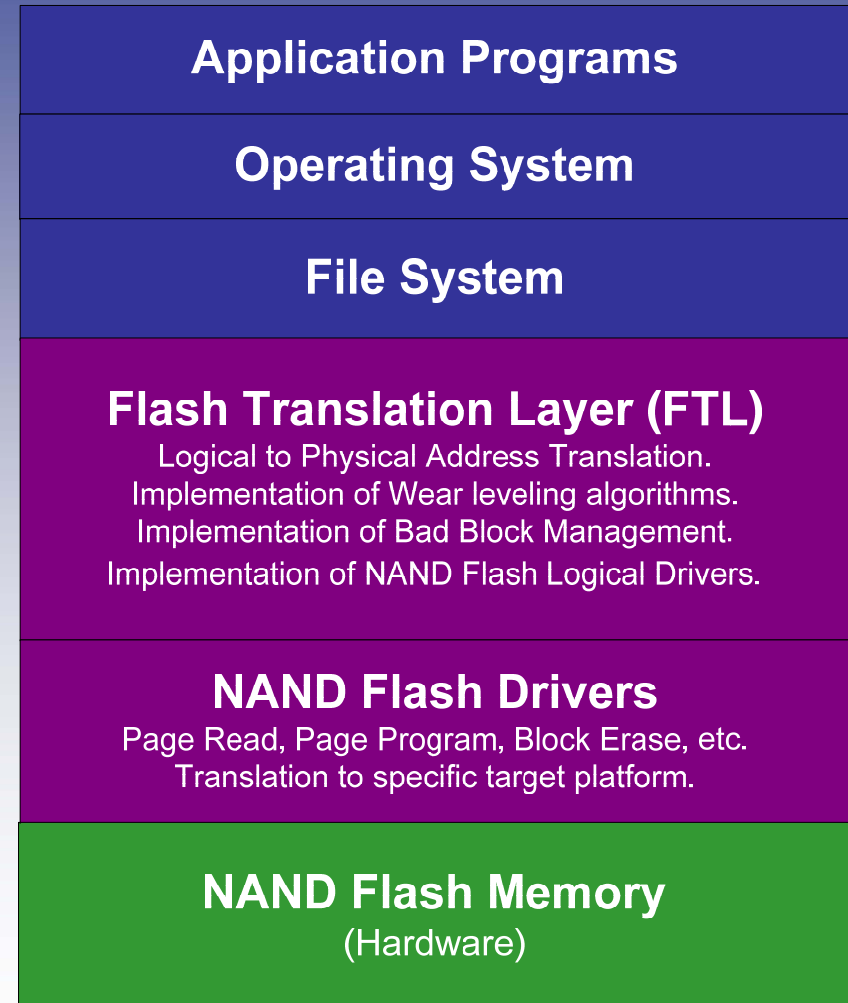


Status of NAND Software in Embedded Operating Systems

	ECC Support	Bad Block Management	Wear Leveling Data Storage	Wear Leveling Code Storage	Enhanced NAND Feature Support	MLC Support	NOP = 1	Data Refresh
OS #1	Yes	Yes	Yes	No	No	No	No	No
OS #2	Yes	Yes	Yes	No	No	No	No	No
OS #3	Yes	Yes	Yes	No	No	No	No	No

The FTL NAND Software Model

- The FTL model is used by Windows Mobile, Symbian, and many other embedded operating systems
- The NAND usage model is controlled by the FTL
- Need FTL changes to address:
 - NOPs
 - Enhanced performance features
 - Wear leveling of code storage
 - Data refresh
 - MLC NAND
- Need Flash driver for:
 - Cache modes
 - Two-plane operations

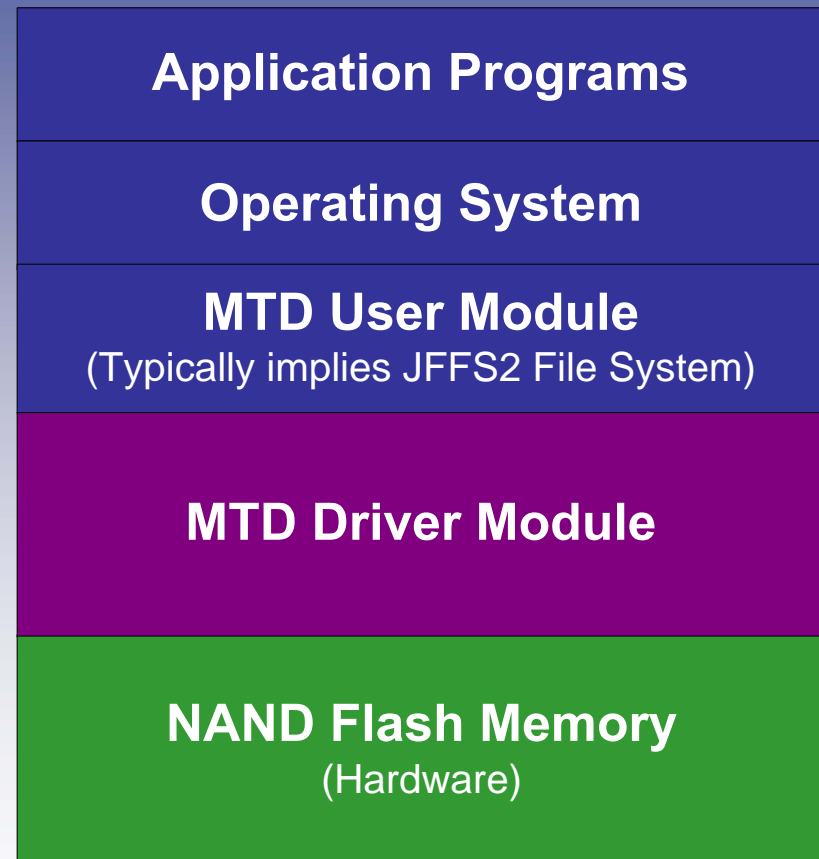


How Can We Drive NAND Usage Model Changes in FTL?

- Work with the major operating system companies to change FTL delivered with OS
 - Pro – covers the widest range of applications, NAND Flash code comes with OS
 - Con – software changes slow, may not cover all application requirements
- Work with software providers of FTL such as Datalight, Inc., or CMX Systems, Inc.
 - Pro – customized solutions, quicker development
 - Con – may require additional software integration work, may not be optimized for multiple memory vendor products
- Use FTL software provided by memory vendor
 - Pro – solution is highly optimized for a specific memory
 - Con – all memory vendors do not offer software, memory second source could be an issue

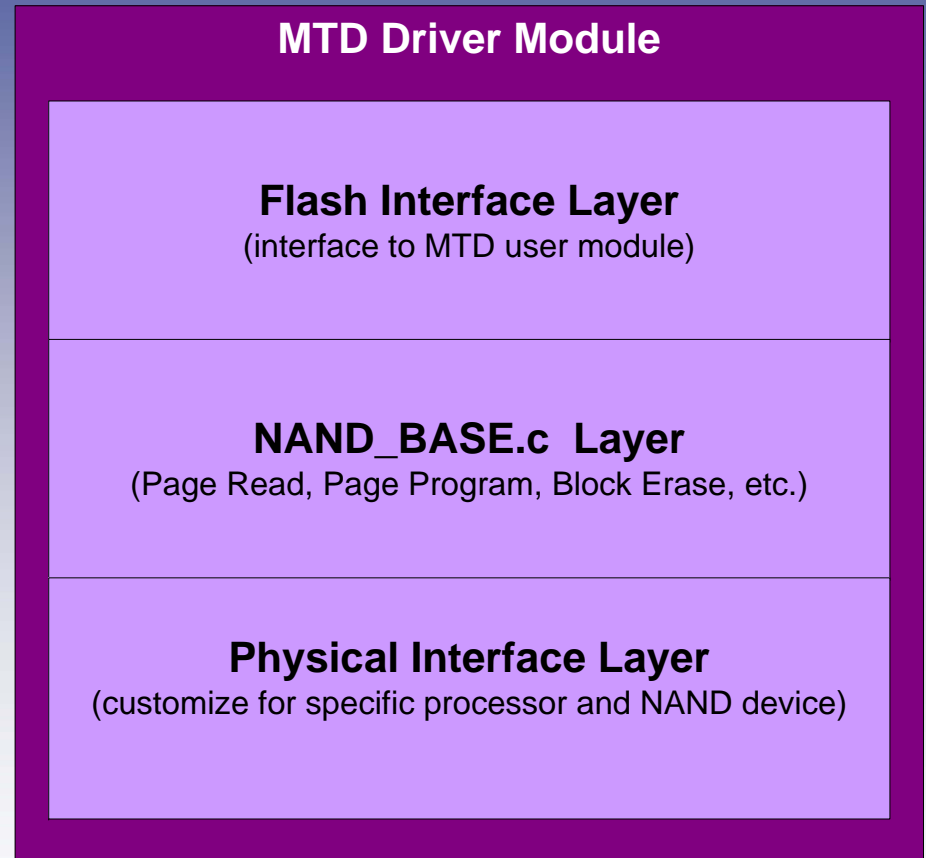
The MTD NAND Software Model

- The MTD model is used by Embedded Linux Operating Systems
- Note that the NAND Flash usage model is controlled by the file system in this model



The MTD NAND Software Model – More Detail

- MTD driver module changes required to implement enhanced NAND features
- Changes must be considered carefully so they don't force changes to upper layers (which are more difficult to implement)
- Easiest method is to make changes in physical interface layer:
 - Pro – changes confined to physical interface layer
 - Con – changes required for individual platforms



How Can We Drive NAND Usage Model Changes in MTD?

- Develop new code for MTD user and driver modules that addresses NAND usage model and performance issues:
 - Pro – addresses reliability and performance issues
 - Con – the required file system changes are not likely to be adopted quickly

- Update MTD driver code only and release to open source community:
 - Pro – quickest solution for adding enhanced performance
 - Con – does not address NAND usage model issues that are controlled by MTD user module (JFFS2)

How is Software Affected by MLC NAND?

- MLC vs. SLC differences from a software standpoint:
 - Number of pages per block: SLC = 64, MLC = 128
 - Spare area in page may be larger for MLC
 - Bad block marking may change

- All performance and reliability discussions from this presentation are applicable to MLC
 - NOP = 1 is a hard requirement for MLC
 - Disturb mechanisms are heightened
 - Enhanced performance features are required to maintain performance

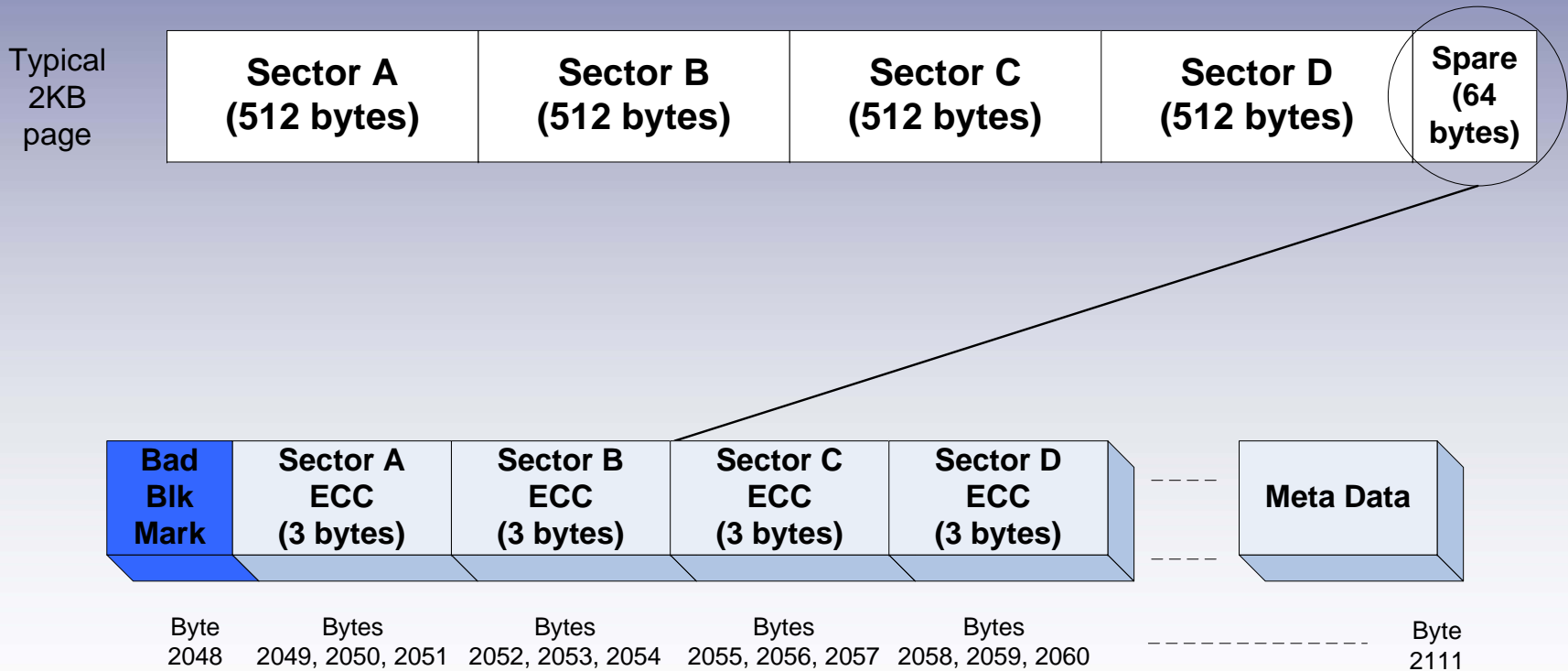
Effect of ECC Requirements on Page Size

Type	Hamming				Reed-Solomon				Binary BCH			
ECT ¹	Overhead Per Sector		Spare Area Usage		Overhead Per Sector		Spare Area Usage		Overhead Per Sector		Spare Area Usage	
	Bits	Bytes	64B	112B	Bits	Bytes	64B	112B	Bits	Bytes	64B	112B
1	13	2	13%	7%	18	3	19%	11%	13	2	13%	7%
2	-	-	-	-	36	5	31%	18%	26	4	25%	14%
4	-	-	-	-	72	9	56%	32%	52	7	44%	25%
8	-	-	-	-	144	18	113%	64%	104	13	81%	46%
10	-	-	-	-	180	23	144%	82%	130	17	106%	61%
14	-	-	-	-	252	32	200%	114%	182	23	144%	82%

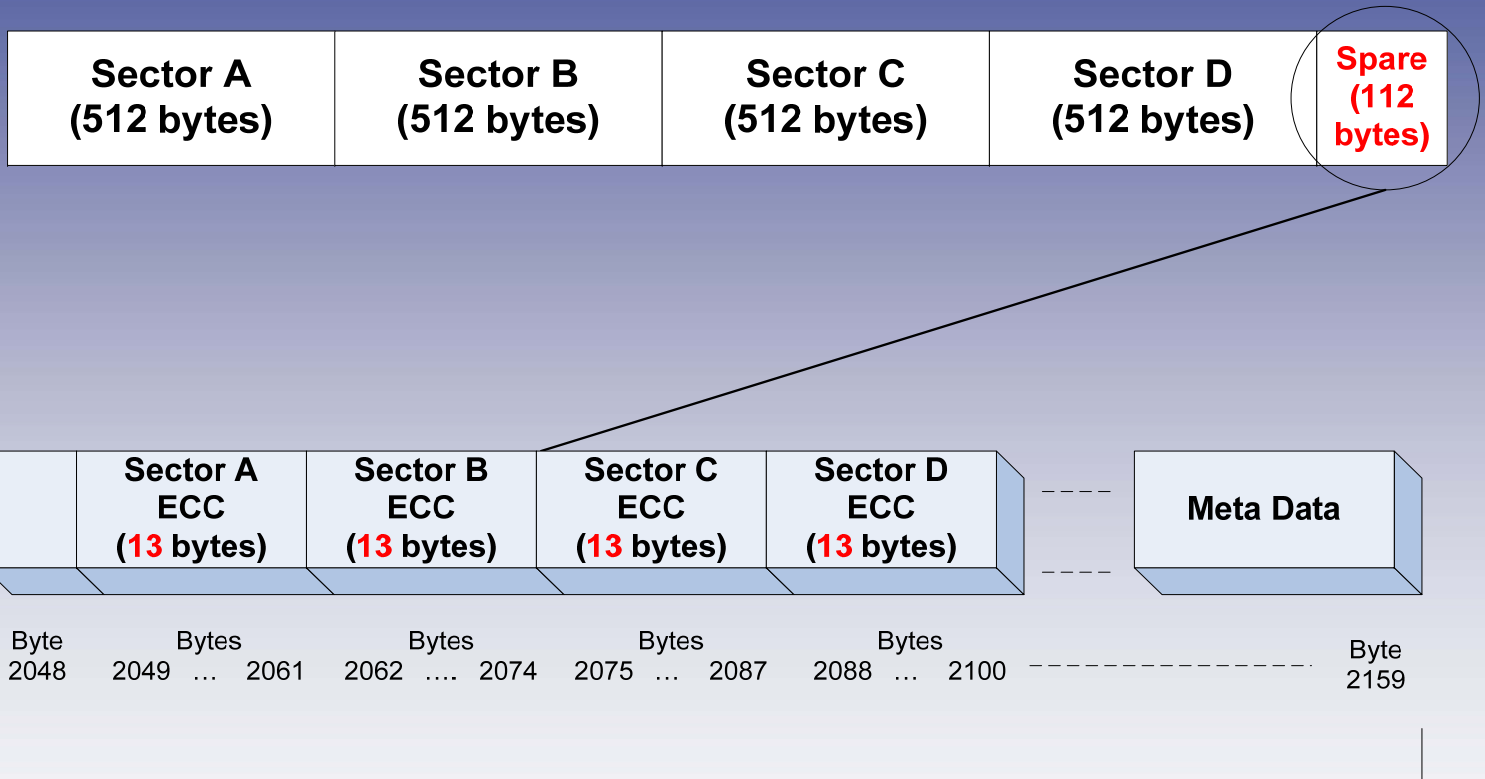
Notes: 1) ECT = error correction threshold; number of correctable bits per 512-byte sector.

Page Layout for SLC 2KB NAND Flash

SLC bad blocks identified by writing a non-FF value to first byte (2048) of spare area in page 0 or 1 of the block



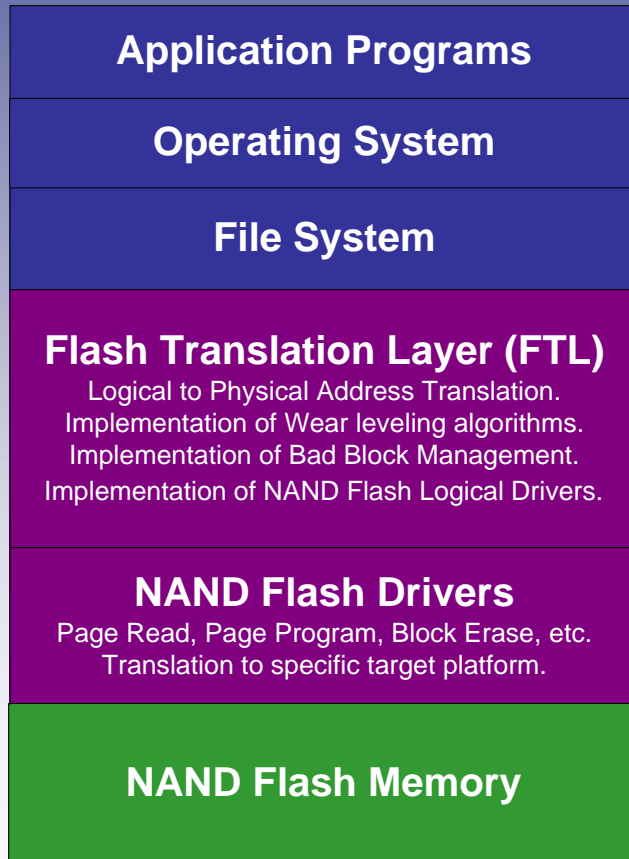
Page Layout for SLC 2KB NAND Flash



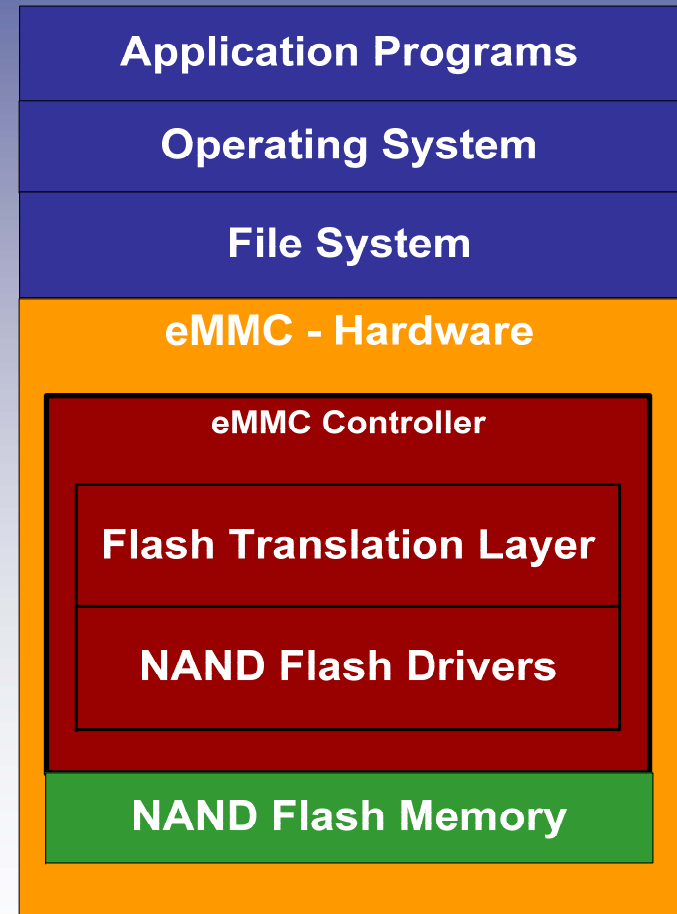
For ONFI MLC devices factory bad block mark can be 00h in any byte of spare area.

Is Software the Only Solution?

Software Solution



Hardware Solution



- Shrinking NAND Flash geometries require software adjustments to ensure hardware reliability
- New NAND Flash performance features require software changes to enhance system performance
- Different operating systems require different NAND Flash software solutions
- Addressing performance and reliability issues can give software a head start on MLC NAND support
- The NAND Flash usage model can be handled in hardware in eMMC

- Micron NAND Flash: www.micron.com/nand/
- Linux MTD Driver: www.linux-mtd.infradead.org/source.html
- Microsoft Windows Mobile: www.microsoft.com/windowsmobile
- Symbian Operating System: www.symbian.com/
- Datalight: www.datalight.com/
- CMX Systems: www.cmx.com/

About Wes Prouty

- Embedded Applications Engineering Manager for Micron Technology
- BSME from University of Idaho; MSEE from Boise State University
- 10 years experience in design and test of embedded applications and memory devices
- waprouty@micron.com
- (208) 363-2283

©2007 Micron Technology, Inc. All rights reserved. Products are warranted only to meet Micron's production data sheet specifications. Information, products and/or specifications are subject to change without notice. All information is provided on an "AS IS" basis without warranties of any kind. Dates are estimates only. Drawings not to scale. Micron and the Micron logo are trademarks of Micron Technology, Inc. All other trademarks are the property of their respective owners.