Challenges and Solutions for Consumer Flash-Memory Devices

Tei-Wei Kuo, Po-Chun Huang, and Yuan-Hao Chang

Dept. of Computer Science & Info. Engr.

National Taiwan University, Taiwan

IIS and CITI, Academia Sinica, Taiwan

Agenda

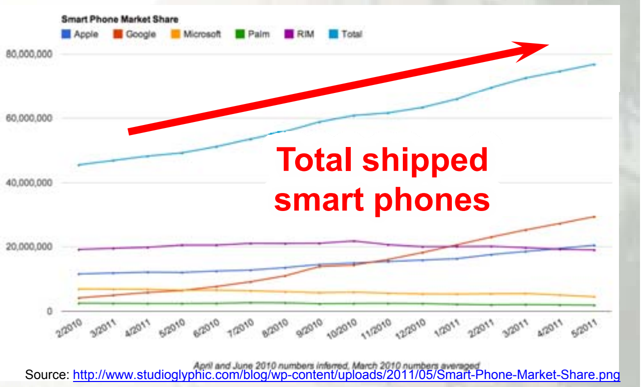
- Introduction
- Architecture and Design Issues
- Performance Issues
- Reliability/Endurance Issues
- Conclusion

Introduction

- Diversified Application Domains
  - Portable Storage Devices
  - Consumer Electronics
  - Servers and Storage Systems
  - Industrial Applications

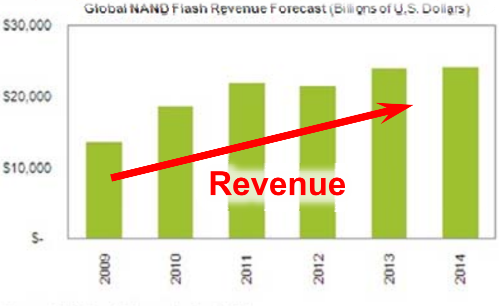
Trends – Market Growth

- **Mobile Devices**
  - Smartphones: 50%+ of growth from 2010/02 to 2011/05
  - Tablet PCs, e.g., iPad
  - Automotive navigation systems
- **Flash Memory**
  - 25% growth from 2011/01 to 2011/08

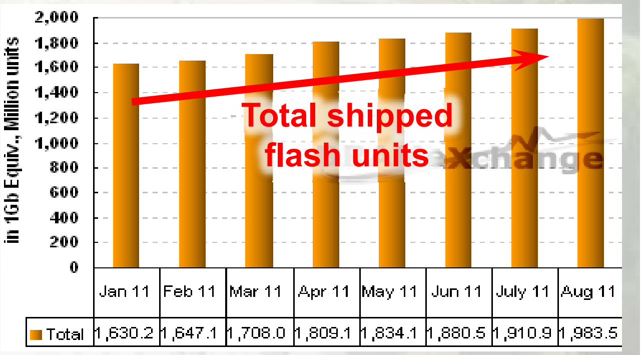


Source: http://www.studioglyphic.com/blog/wp-content/uploads/2011/05/Smart-Phone-Market-Share.png

Revenue: 22 billions in 2011



Source: IHS iSuppli Research, Jan. 2011

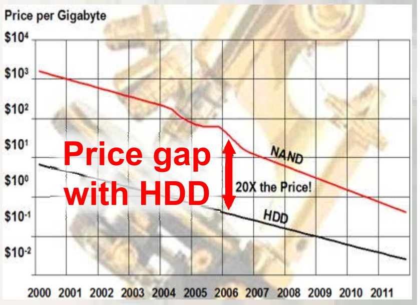


	Jan 11	Feb 11	Mar 11	Apr 11	May 11	Jun 11	July 11	Aug 11
Total	1,630.2	1,647.1	1,708.0	1,809.1	1,834.1	1,880.5	1,910.9	1,983.5

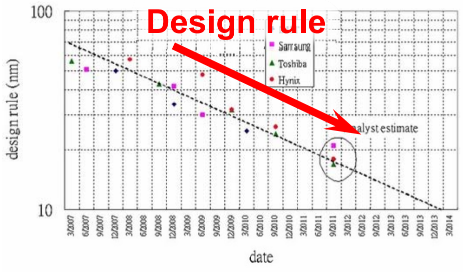
Source: http://www.dramexchange.com/images/monthlyproduction2009.gif

Trends – Market and Technology

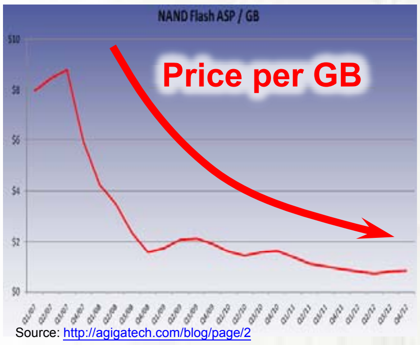
- **Competitiveness in the Price**
  - Dropping Rate and the Price Gap with HDDs
- **Technology Trend over the Market**
  - Improved density
  - Degraded performance
  - Degraded reliability
  - Worsened access constraints



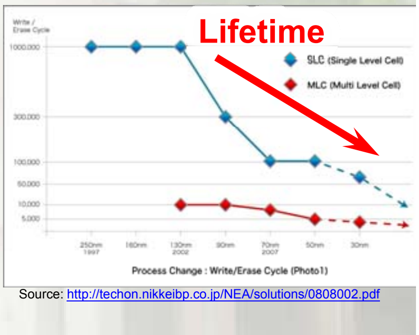
Source: http://agigatech.com/blog/wp-content/uploads/2009/12/Handy-HDD-SSD-Cost-Differential.jpg, from Understanding the NAND Market



Source: http://www.storagenewsletter.com/images/public/sites/StorageNewsletter.com/articles/icono7/intel_and_micron_20nm_f3_540.jpg



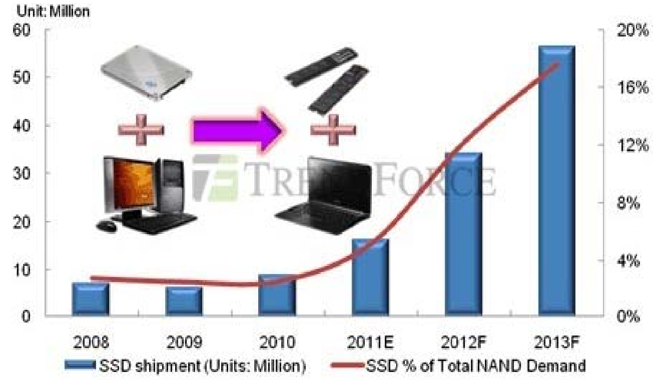
Source: http://agigatech.com/blog/page/2



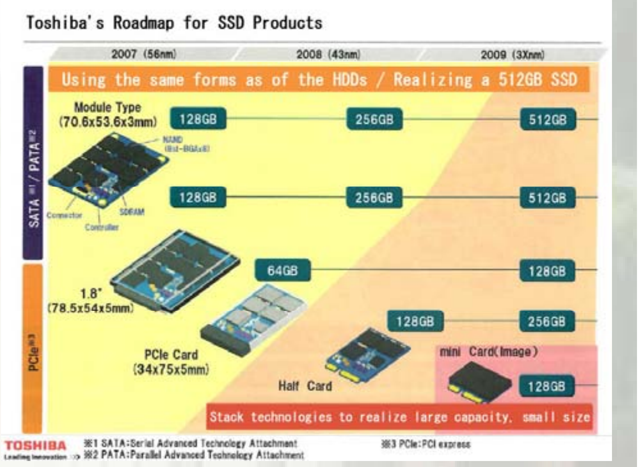
Source: http://techon.nikkeibp.co.jp/NEA/solutions/0808002.pdf

Trends – Solid-State Disks (SSDs)

- Flourishing in SSD Developments
  - Top 20 Vendors in 2010Q4: Fusion-io, SandForce, STEC, Violin Memory, Texas Memory Systems, OCZ, WD Solid State Storage, Pliant Technology, SanDisk, RunCore, ForeMay, Intel, Toshiba, SMART Modular Technologies, Seagate, Virident Systems, Kove, EMC, BiTMICRO, and DDRdrive



Source: DRAMeXchange, Aug., 2011


Toshiba's Roadmap for SSD Products
2007 (56nm)
2008 (43nm)
2009 (3Xnm)
Using the same forms as of the HDDs / Realizing a 512GB SSD
SATA ※1 / PATA ※2
Module Type (70.6x53.6x3mm)
128GB
256GB
512GB
128GB
256GB
512GB
1.8" (78.5x54x5mm)
64GB
128GB
PCIe ※3
128GB
256GB
PCIe Card (34x75x5mm)
Half Card
mini Card (Image)
128GB
Stack technologies to realize large capacity, small size
TOSHIBA
※1 SATA: Serial Advanced Technology Attachment
※2 PATA: Parallel Advanced Technology Attachment
※3 PCIe: PCI express

Agenda

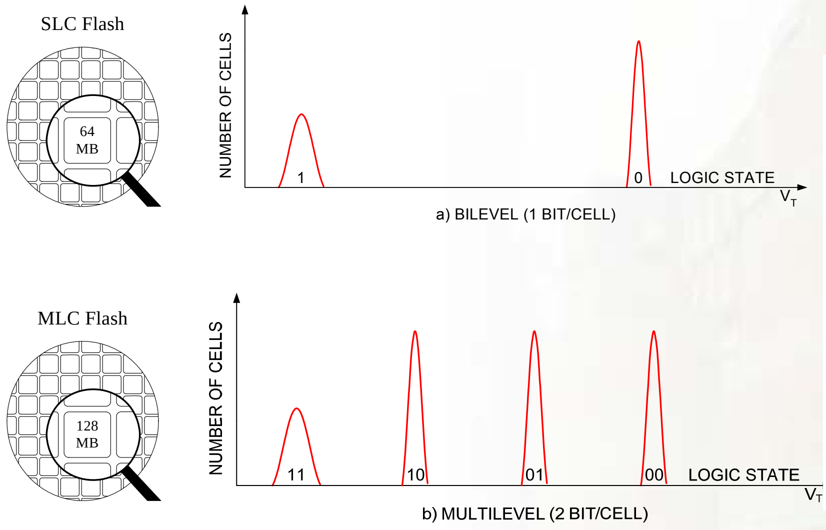
- Introduction
- Architecture and Design Issues
- Performance Issues
- Reliability/Endurance Issues
- Conclusion

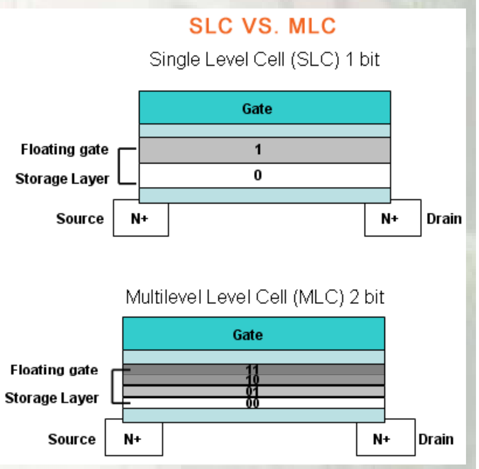
The Characteristics of Different Storage Media

Media	**Access time**		
	Read	**Write**	**Erase**
DRAM	13.9ns (1B) 7.12*us* (512B)	13.9ns (1B) 7.12*us* (512B)	N/A
NOR Flash	45ns (1B) 23.0*us* (512B)	14*us* (1B) 7.2ms (512B)	18ms (128KB)
PCM	115—135 ns (1B) 13*us* (512B)	115—135 ns (1B) 13*us* (512B)	N/A
NAND Flash (SLC)	15*us* (1B) MAX: 35*us* (8KB)	300*us* (1B) TYP: 350*us* (8KB) MAX: 500us (8KB)	TYP: 1.5ms (1MB) MAX:3ms (1MB)
DISK	15.8ms TYP: 8.2ms (512B)	6.06ms TYP: 9.2ms (512B)	N/A

Reference Devices/Modules: DRAM: Micron DDR3-1333. NOR Flash: Silicon Storage Technology SST39LF020. PCM: Micron P8P Parallel PCM, NAND Flash: Micron MT29F256G08AUAAAC5. Disk: Deskstar™ 7K3000 series.

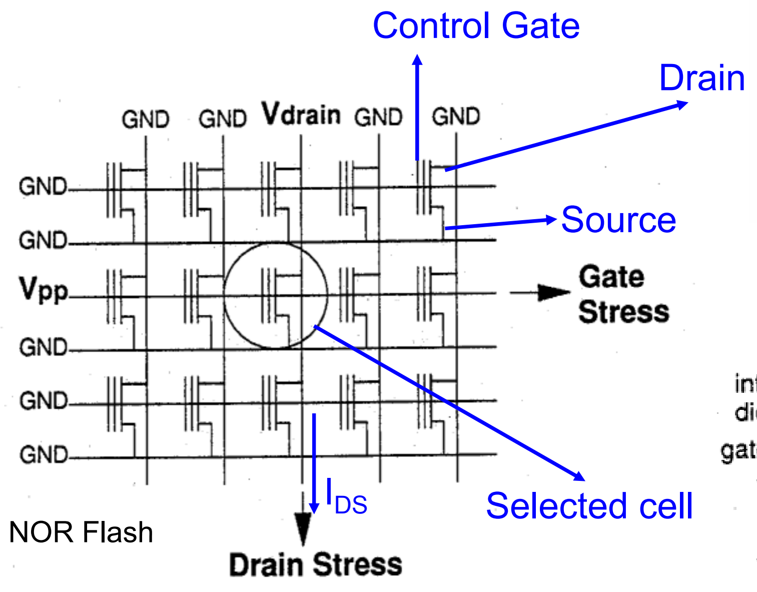
Single Level Cell (SLC) vs Multi-Level Cell (MLC)

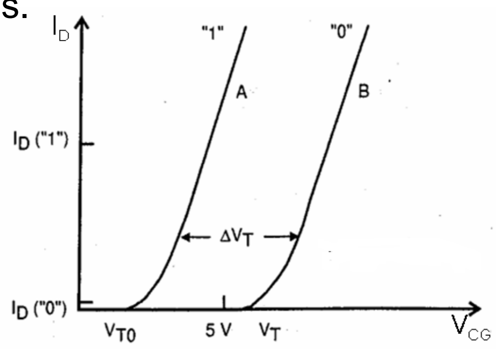


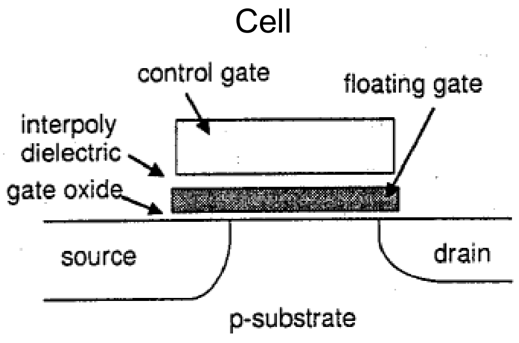


Single-Level Cell (SLC)

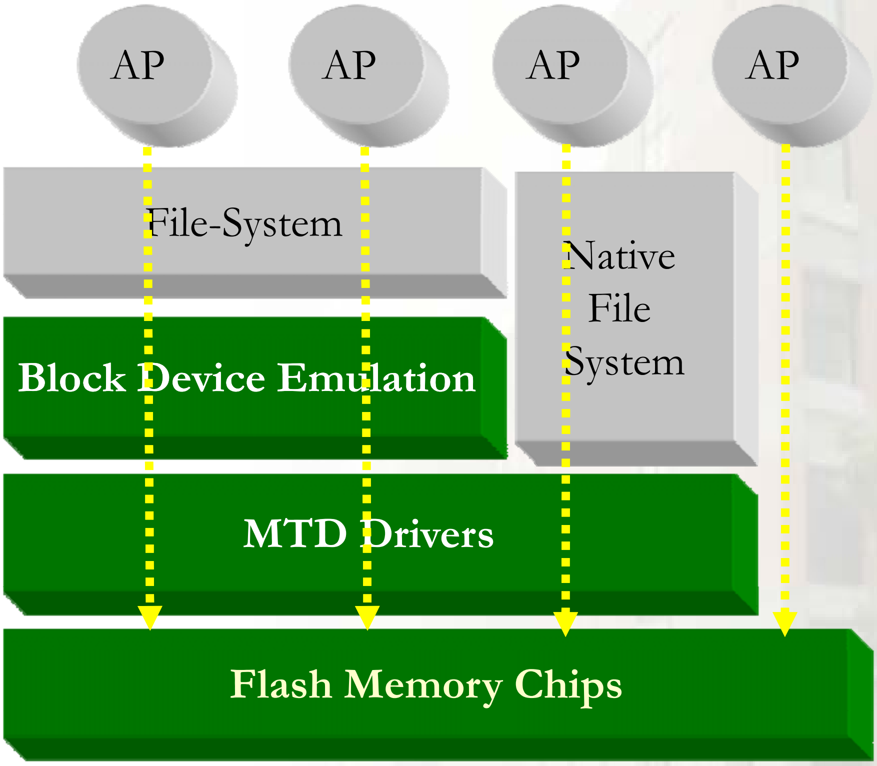
- Each Word-Line is connected to control gates.
- Each Bit-Line is connected to the drain.







System Architectures



AP

AP

AP

AP

File-System

Native File System

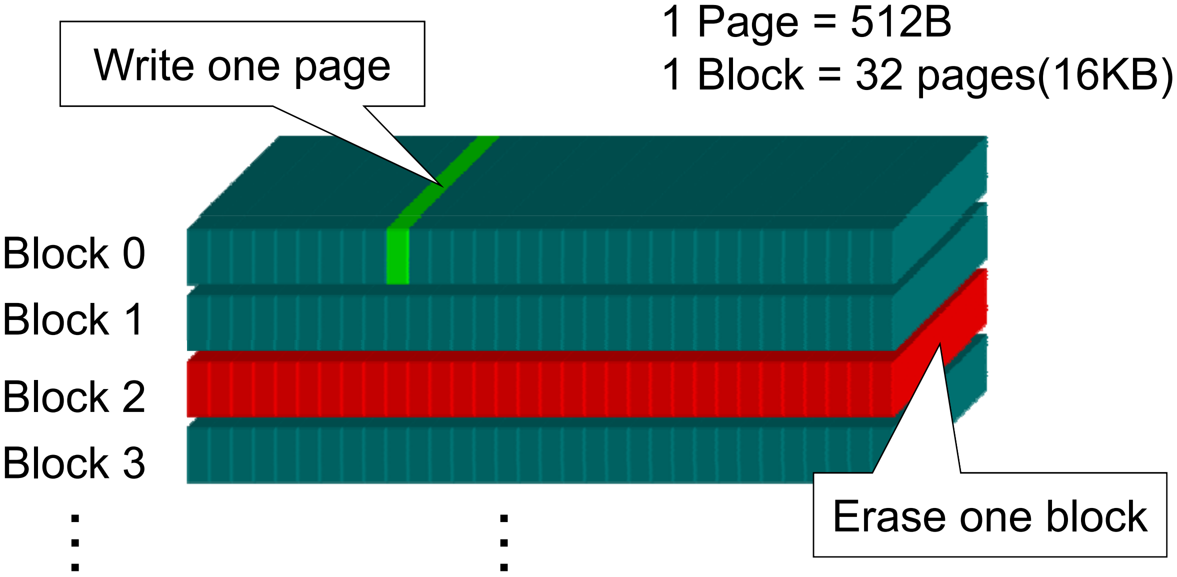
Block Device Emulation

MTD Drivers

Flash Memory Chips

Management Issues – Flash-Memory Characteristics

1 Page = 512B
1 Block = 32 pages(16KB)



Write one page

Block 0

Block 1

Block 2

Block 3

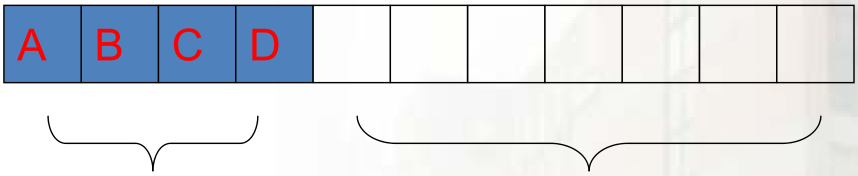
⋮

⋮

Erase one block

Management Issues – Flash-Memory Characteristics

- Example 1: Out-place Update



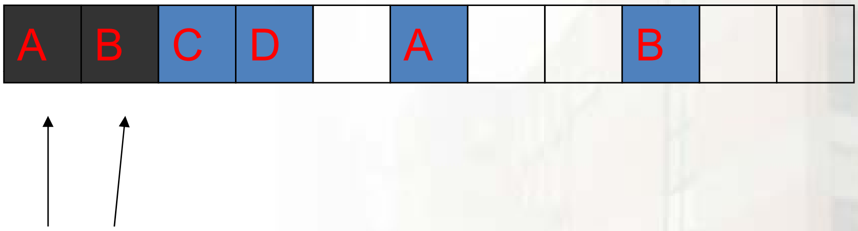
Live pages

Free pages

Suppose that we want to update data A and B…

Management Issues – Flash-Memory Characteristics

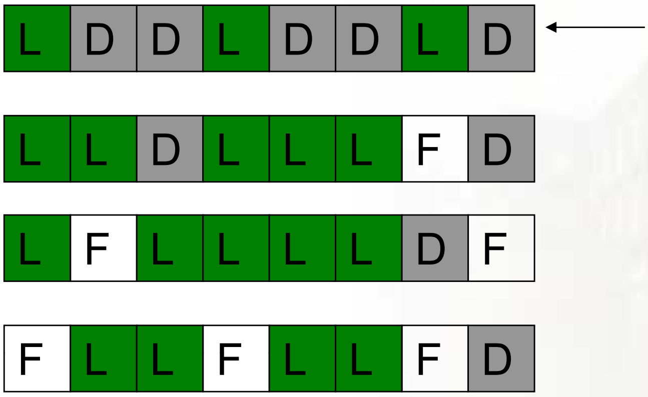
- Example 1: Out-place Update



Dead pages

Management Issues – Flash-Memory Characteristics

- Example 2: Garbage Collection



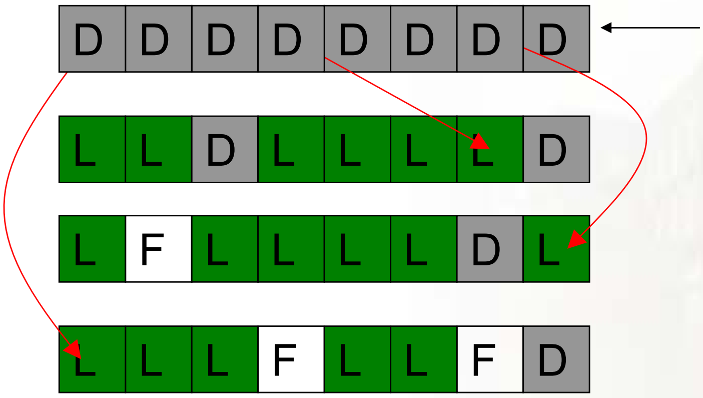
L	D	D	L	D	D	L	D
L	L	D	L	L	L	F	D
L	F	L	L	L	L	D	F
F	L	L	F	L	L	F	D

This block is to be recycled.
(3 live pages and 5 dead pages)

🟩 A live page
▨ A dead page
□ A free page

Management Issues – Flash-Memory Characteristics

- Example 2: Garbage Collection



D	D	D	D	D	D	D	D
L	L	D	L	L	L	L	D
L	F	L	L	L	L	D	L
L	L	L	F	L	L	F	D

Live data are copied to somewhere else.

🟩 A live page
▨ A dead page
□ A free page

Management Issues – Flash-Memory Characteristics

- Example 2: Garbage Collection

F	F	F	F	F	F	F	F
L	L	D	L	L	L	L	D
L	F	L	L	L	L	D	L
L	L	L	F	L	L	F	D

The block is then erased.

Overheads:
•live data copying
•block erasing.

🟩 A live page
▨ A dead page
□ A free page

Management Issues – Flash-Memory Characteristics

- Example 3: Wear-Leveling

100	L	D	D	L	D	D	L	D	A
10	L	L	D	L	L	L	F	D	B
20	L	F	L	L	L	L	D	F	C
15	F	L	L	F	L	L	F	D	D

↑
Erase cycle counts

Wear-leveling might interfere with the decisions of the block-recycling policy.

🟩 A live page
▨ A dead page
□ A free page

Management Issues – Flash-Memory Characteristics

- **SLC Flash Access Constraints**
  - **Write-Once**
    - No writing on the same page unless its residing block is erased!
    - Pages are classified as valid, invalid, and free pages.
  - **Bulk-Erasing**
    - Pages are erased in a block unit to recycle used but invalid pages.
  - **Wear-Leveling**
    - Each block has a limited lifetime in erasing counts.
- **Additional MLC Flash Access Constraints**
  - Prohibition of partial page programming
  - Serial page programming in a block

Comparisons of SLC and MLC

- **1-bit/cell SLC NAND flash**
  - 100,000 Program/Erase cycles (with ECC)[1]
  - 10 years Data Retention
- **2-bits/cell MLC NAND flash**
  - 3000—10,000 Program/Erase cycles (with ECC)[2]
  - 10 years Data Retention
- **3-bit/cell TLC NAND flash**
  - 250—500 Program/Erase cycles[3]
- **4-bits/cell QLC NAND flash** (2011—)
  - Developers: Intel, SanDisk, Micron, Toshiba and Samsung

[1] ST Micro-electronics NAND SLC large page datasheet (NAND08GW3B2A)
[2] ST Micro-electronics NAND MLC large page datasheet (NAND04GW3C2A)
[3] Spectek F??B74A61K3BAA??-AF/L

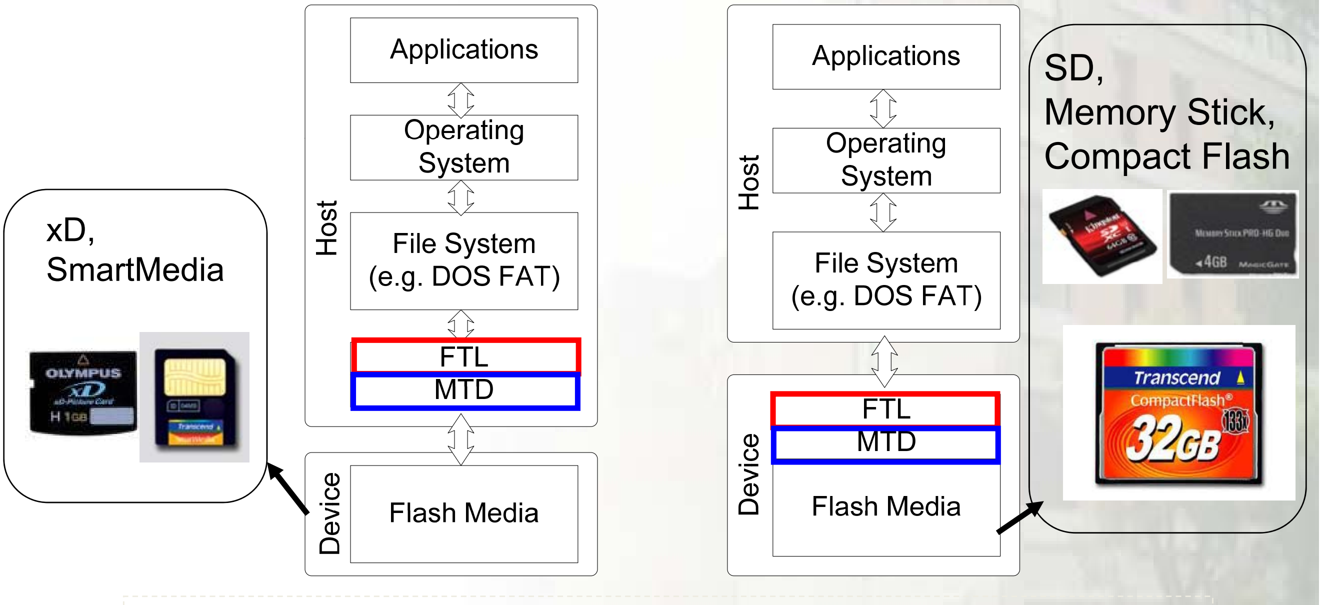
Management Issues – Challenges

- The write throughput drops significantly after garbage collection starts!
- The capacity of flash-memory storage systems increases very quickly such that memory space requirements grows quickly.
- Reliability becomes more and more critical when the manufacturing capacity increases!
- The significant increment of flash-memory access rates seriously exaggerates the Read/Program Disturb Problems!

Agenda

- Introduction
- Architecture and Design Issues
- Performance Issues
- Reliability/Endurance Issues
- Conclusion

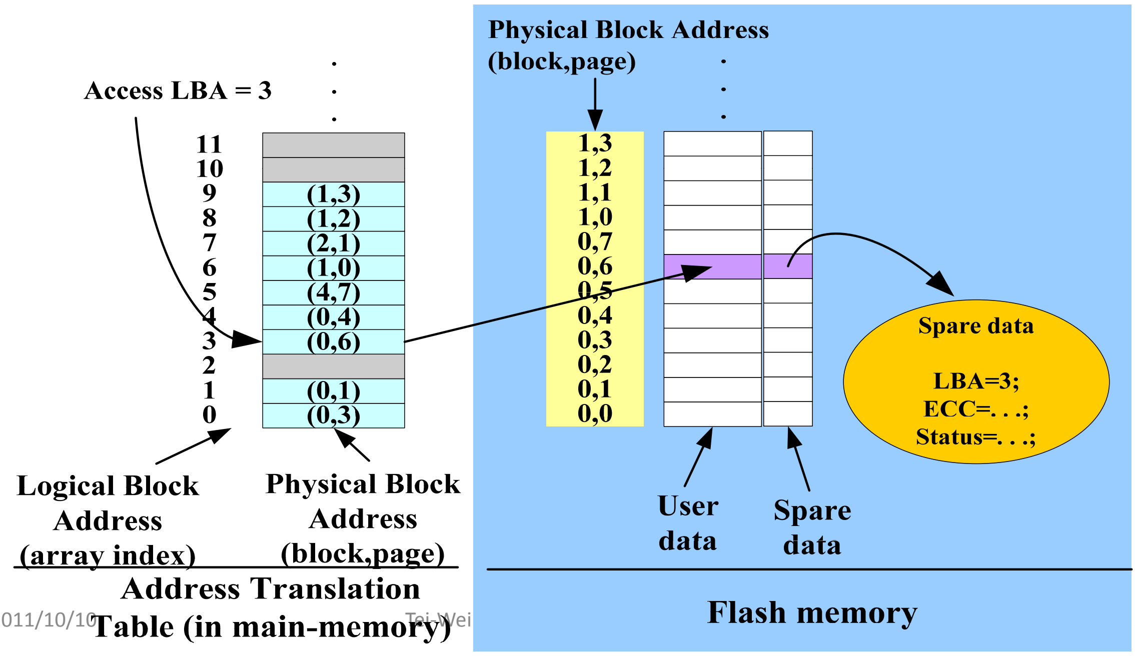
System Architecture – Layers



*FTL: Flash Translation Layer, MTD: Memory Technology Device

Example Address-Mapping Policies – FTL

- FTL adopts a page-level address translation mechanism.
  – The main problem of FTL is on large memory space requirements for storing the address translation information.



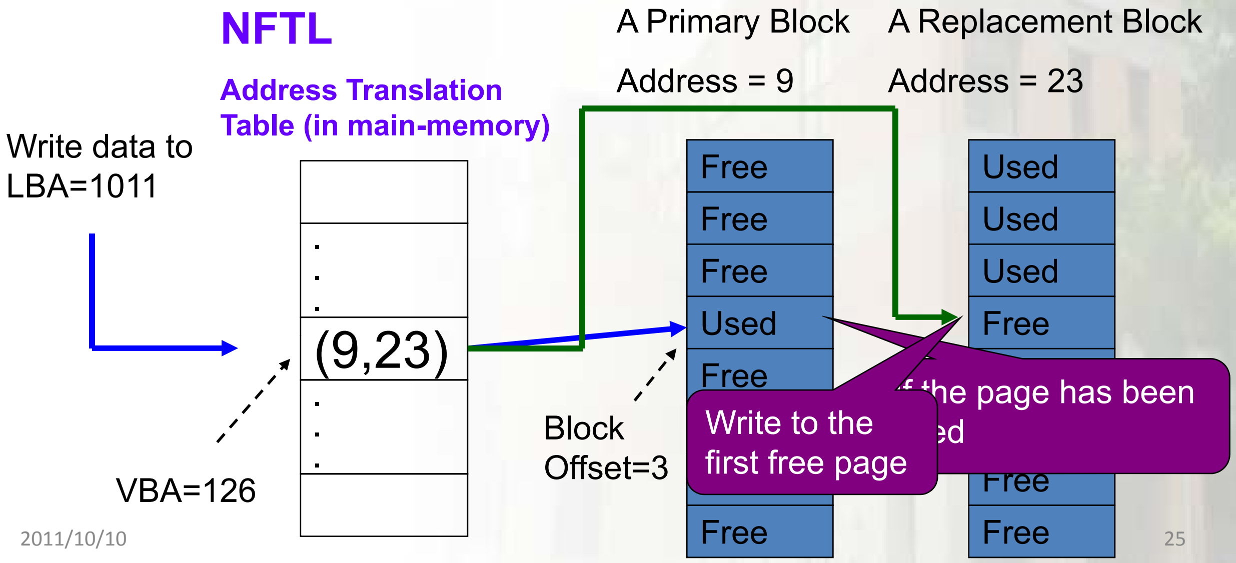
Logical Block Address (array index)	Physical Block Address (block,page)
11	
10	
9	(1,3)
8	(1,2)
7	(2,1)
6	(1,0)
5	(4,7)
4	(0,4)
3	(0,6)
2	
1	(0,1)
0	(0,3)

Address Translation Table (in main-memory)

1,3
1,2
1,1
1,0
0,7
0,6
0,5
0,4
0,3
0,2
0,1
0,0

Example Address-Mapping Policies – NFTL

- A logical address under NFTL is divided into a virtual block address and a block offset.
  - e.g., LBA=1011 => virtual block address (VBA) = 1011 / 8 = 126 and block offset = 1011 % 8 = 3


NFTL
Address Translation Table (in main-memory)
Write data to LBA=1011
(9,23)
VBA=126
A Primary Block Address = 9
A Replacement Block Address = 23
Free
Free
Free
Used
Free
Free
Block Offset=3
Used
Used
Used
Free
Free
Free
Write to the first free page
f the page has been
ed

Address-Mapping Policies – Fine-Grained vs. Coarse Grained Ones

	FTL	NFTL
Memory Space Requirements	Larger	Smaller
Address Translation Time	Shorter	Longer
Garbage Collection Overhead	Less	More
Space Utilization	Higher	Lower

- The Memory Space Requirements for a 16GB NAND flash (4KB/Page, 4B/Table Entry, 128 Pages/Block)
  - FTL: 16MB (= 4*16G/4K)
  - NFTL: 128KB (= 4*16G/(4K*128))

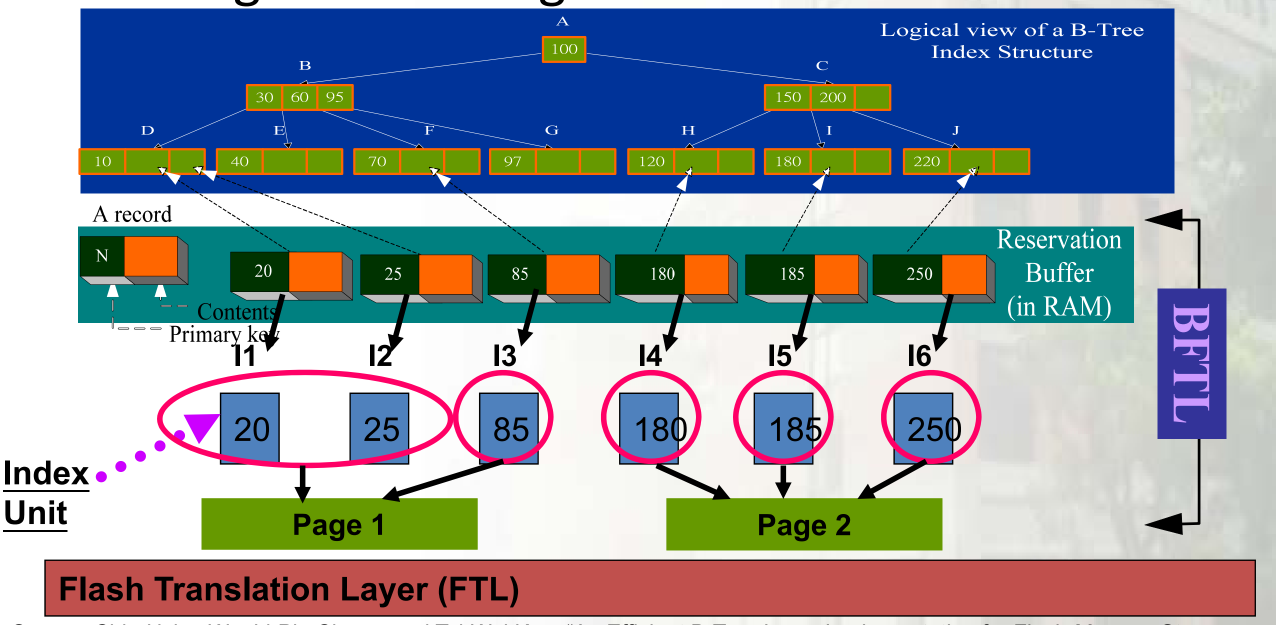
Remark: Each page of small-block(/large-block) SLC NAND can store 512B(/2KB) data, and there are 32(/64) pages per block.
Each page of MLCx2 NAND can store 4KB, and there are 128—256 pages per block.

Key Issues and Technologies

- **Address Translation**
  - Reduce the size of address translation information
  - Adjust address translation scheme with heterogeneous mapping granularities
  - Store address translation information over flash
- **Garbage Collection and Wear Leveling**
  - Cost versus Benefits
  - Needs in Performance/Real-time Constraints
- **Parallelism in Access**
  - Adaptive Striping and Architecture Designs

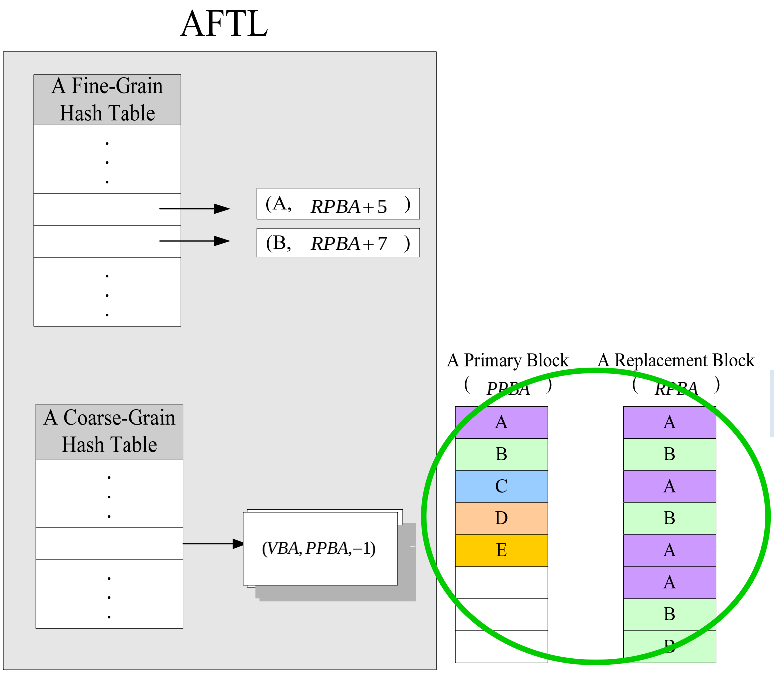
Address Translation – Indexing and Small Writes

- Packing and Caching



Source: Chin-Hsien Wu, Li-Pin Chang, and Tei-Wei Kuo, "An Efficient B-Tree Layer Implementation for Flash-Memory Storage Systems, " ACM Transactions on Embedded Computing Systems, Volume 6, Issue 3, July 2007

Address Translation – Adaptive Flash Translation Layer (AFTL)



1. AFTL doesn’t erase the two blocks immediately.
2. AFTL moves the mapping information of the replacement block to the fine-grained hash table by adding fine-grained slots.

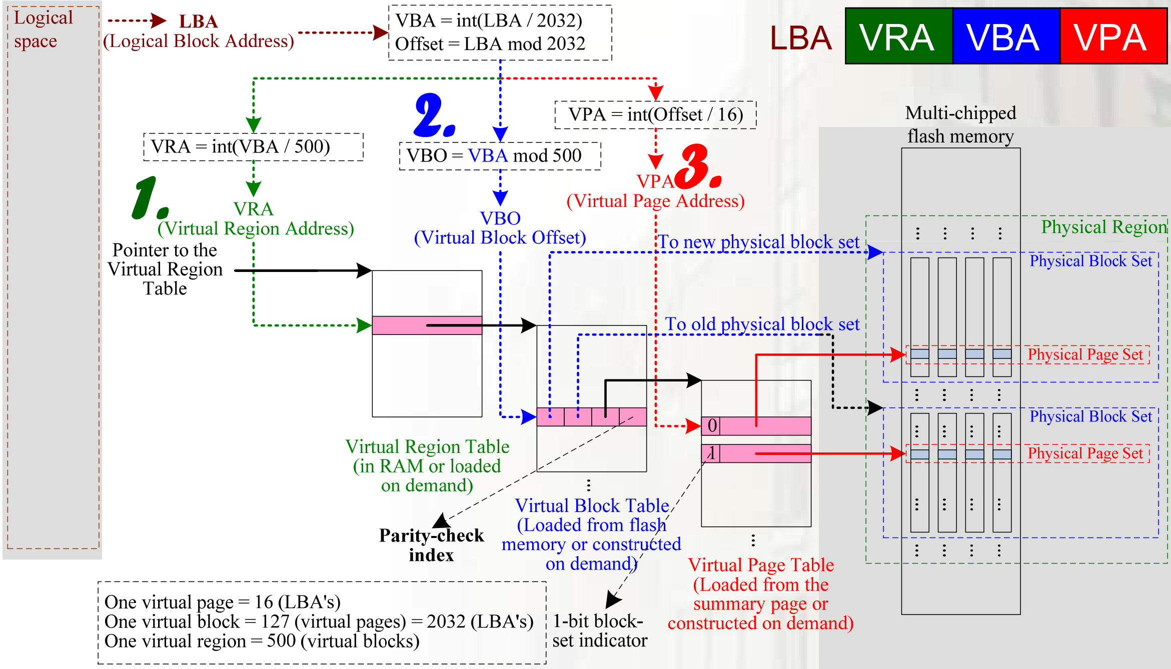
Coarse-to-Fine Switching

3. The RPBA field of the corresponding mapping information is nullified.

Chin-Hsien Wu and Tei-Wei Kuo, 2006, “An Adaptive Two-Level Management for the Flash Translation Layer in Embedded Systems,” IEEE/ACM 2006 International Conference on Computer-Aided Design (ICCAD), November 5-9, 2006.

Address Translation – Region-Based Mapping

- A three-level address translation architecture

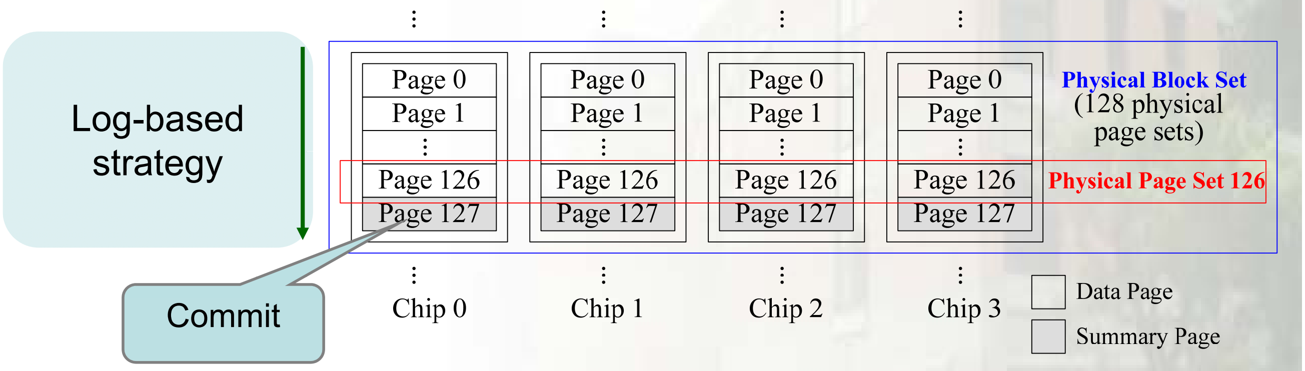


One virtual page = 16 (LBA's)
One virtual block = 127 (virtual pages) = 2032 (LBA's)
One virtual region = 500 (virtual blocks)

Yuan-Hao Chang and Tei-Wei Kuo, "A Commitment-based Management Strategy for the Performance and Reliability Enhancement of Flash-memory Storage Systems," the ACM/IEEE Design Automation Conference (DAC), San Francisco, Jul. 26-31, 2009.

Address Translation – Commitment-based Management

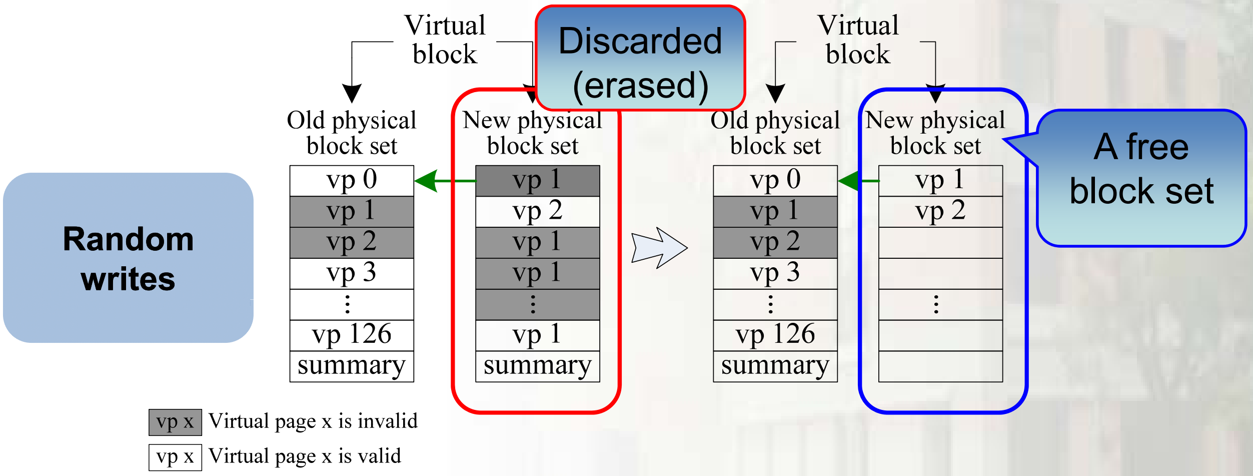
- An adaptive block mapping mechanism with a log-based strategy



Yuan-Hao Chang and Tei-Wei Kuo, "A Commitment-based Management Strategy for the Performance and Reliability Enhancement of Flash-memory Storage Systems," the ACM/IEEE Design Automation Conference (DAC), San Francisco, Jul. 26-31, 2009.

Address Translation – Adaptivity to Access Patterns

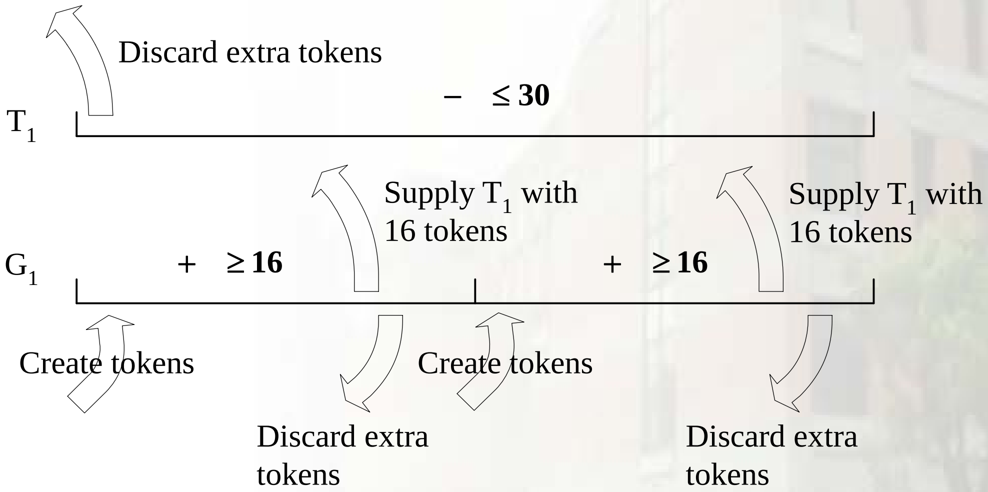
- Good performance to random writes and sequential writes
  - A virtual block → up to 2 physical block sets
    - Replace the block set with fewer valid data.
    - Set the remaining one as the old one.



Yuan-Hao Chang and Tei-Wei Kuo, "A Commitment-based Management Strategy for the Performance and Reliability Enhancement of Flash-memory Storage Systems," the ACM/IEEE Design Automation Conference (DAC), San Francisco, Jul. 26-31, 2009.

Garbage Collection – Real-Time Garbage Collection

- Garbage Collection and MLC Write Constraints



Discard extra tokens

T_1 $- \leq 30$

Supply T_1 with 16 tokens

Supply T_1 with 16 tokens

G_1 $+ \geq 16$ $+ \geq 16$

Create tokens

Create tokens

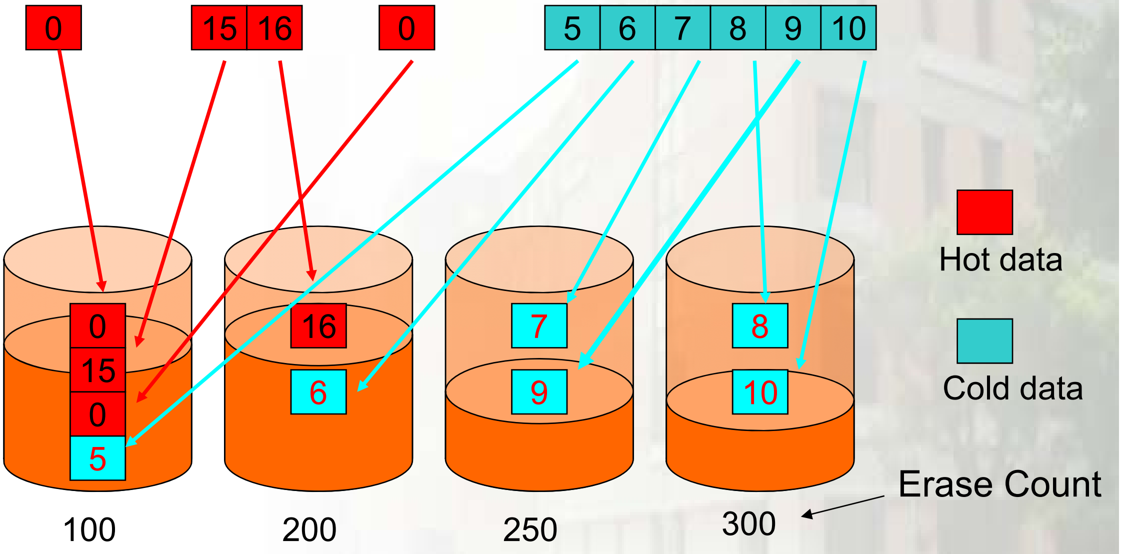
Discard extra tokens

Discard extra tokens

Source: Li-Ping Chang and Tei-Wei Kuo, “A Real-Time Garbage Collection Mechanism for Flash-Memory Storage Systems in Embedded Systems,” the 8th International Conference on Real-Time Computing Systems and Applications (RTCSA), Tokyo, Japan, March 2002

Parallel access supports – Adaptive striping

- Stripping and Utilization:
  Distribute hot and cold data evenly over banks



Source: Li-Pin Chang and Tei-Wei Kuo, "An Adaptive Stripping Architecture for Flash Memory Storage Systems of Embedded Systems," IEEE Eighth Real-Time and Embedded Technology and Applications Symposium (RTAS), San Jose, USA, Sept 2002

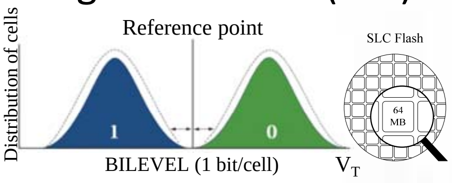
Agenda

- Introduction
- Architecture and Design Issues
- Performance Issues
- Reliability/Endurance Issues
- Conclusion

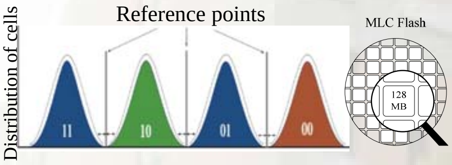
Reliability Challenges

- **Low Endurance**
- **Bad Data Retention**
- **High Bit Error Rate**
- **Serious Disturbing**

Single-Level-Cell (SLC)



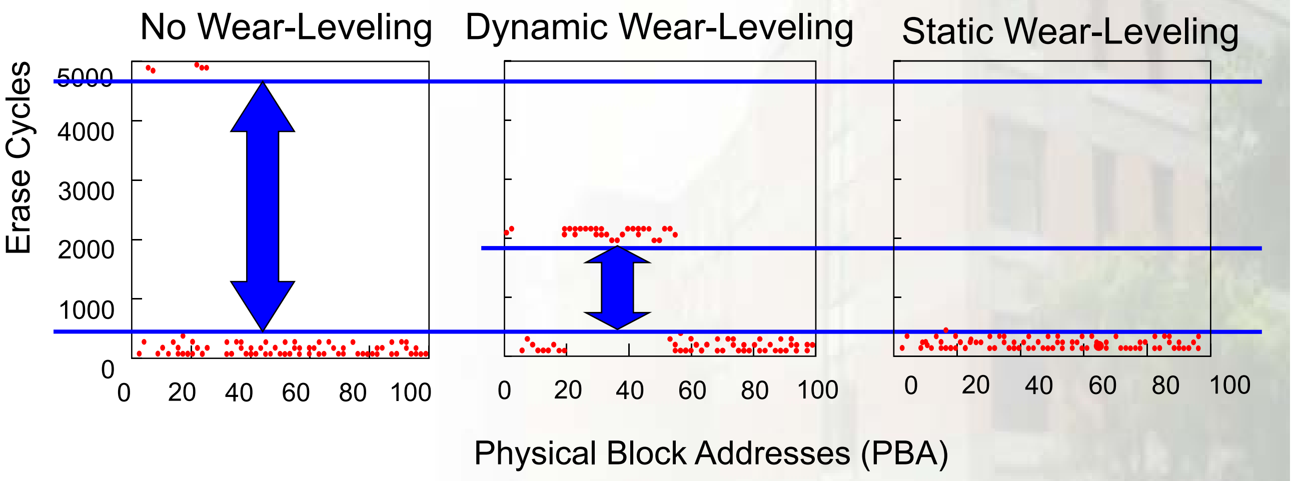
Multi-Level-Cell (MLC)[1]



	Read (Mbps)	Write (Mbps)	Endurability (Erase cycles)	Reliability (Bit error rate)	Cost (US $/GB)
SLC[1]	235	23	60,000	10^{-9}	0.813
MLC$_{x2}$[2]	109	**6.3**	≤ 3,000	10^{-6}	0.113[4]
TLC / MLC$_{x3}$[3]	27	**0.8**	**≤ 500**	**$\geq 10^{-5}$**	0.095[5]
MPEG-2 (1280x720)	20	20			
MPEG-4	6 - 7	6 - 7			

* 2010Q2: SLC 6.01USD/GB, MLCx2 1.70USD/GB, TLC 1.49USD/GB
[1][2] Micron MT29F256G08AUCAB & MT29F512G08CUAA; [3] Spectek FNNB63A (downgraded flash product); [4][5] DRAMExchange, October 2011

Wear-Leveling Technologies

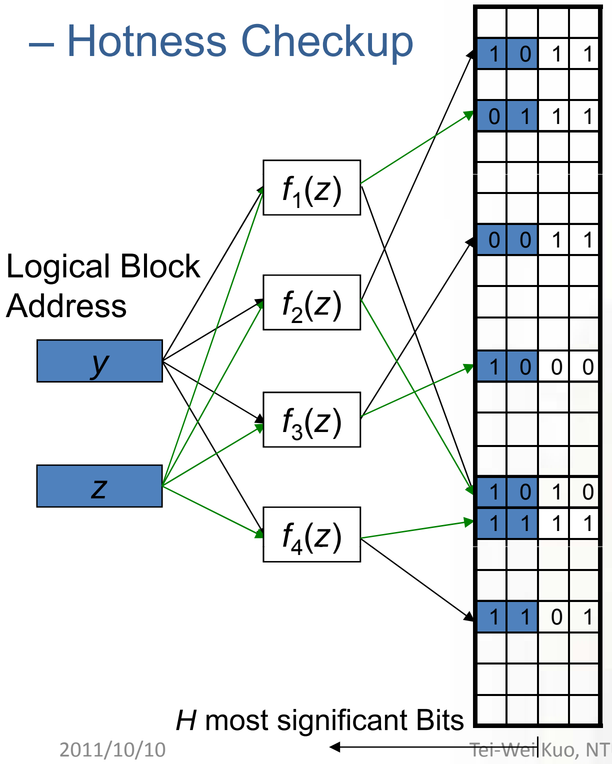


Key Issues and Technologies

- **Identification of Hot and Cold Data**
  - Locality in Access
  - Garbage Collection Performance
- **Wear Leveling**
  - Dynamic Wear Leveling
  - Static Wear Leveling
- **Reliability Enhancement**
  - Downgrading Designs
  - Reliability Enhancement at FTL/MTD/File-System Levels

Efficient Hot-Data Identification

– Hotness Checkup



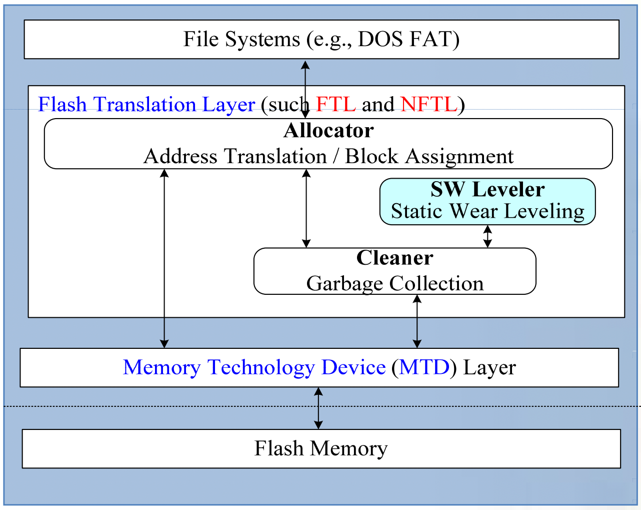
1. An LBA is to be verified as a location for hot data.
2. The corresponding LBA y is hashed simultaneously by K given hash functions.
3. Check if the H most significant bits of every counter of the K hashed values contain a non-zero bit value.

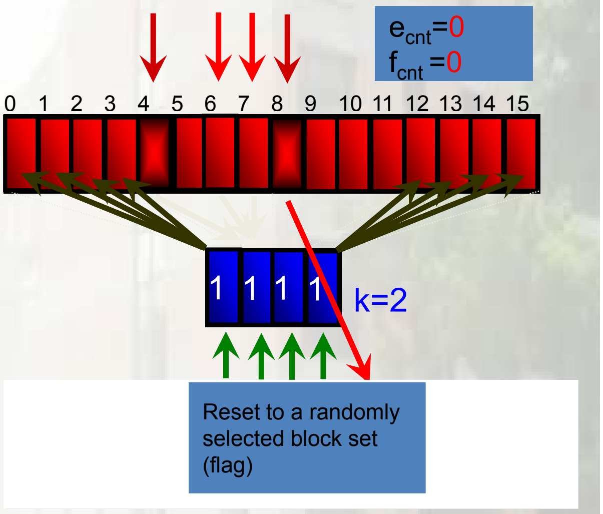
A Multi-Hash-Function Framework

Jen-Wei Hsieh, Tei-Wei Kuo, Li-Pin Chang, "Efficient Identification of Hot Data for Flash Memory Storage Systems," the ACM Transactions on Storage, Volume 2, Issue 1, pp.22-40, Feb 2006.

Static Wear Leveling

- A modular design for compatibility considerations
  - An unevenness level (e_{cnt} / f_{cnt}) >= T → Triggering of the Static Wear Leveler


File Systems (e.g., DOS FAT)
Flash Translation Layer (such FTL and NFTL)
Allocator
Address Translation / Block Assignment
SW Leveler
Static Wear Leveling
Cleaner
Garbage Collection
Memory Technology Device (MTD) Layer
Flash Memory

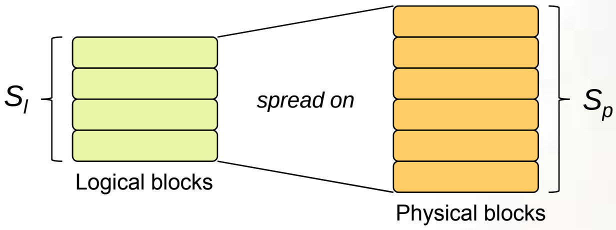


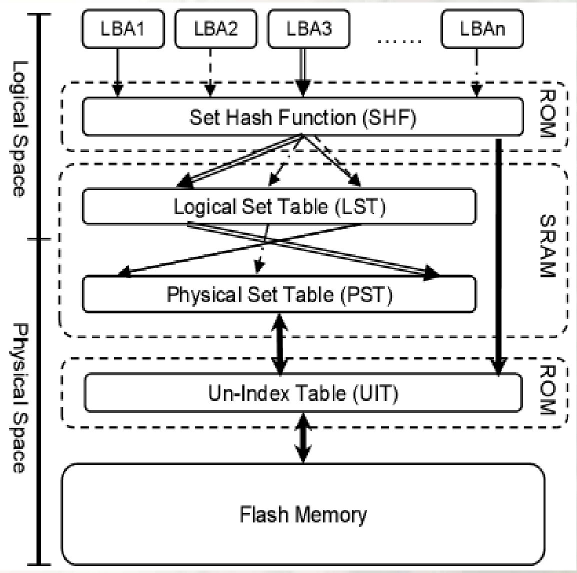
Reset to a randomly selected block set (flag)

Yuan-Hao Chang, Jen-Wei Hseuh, and Tei-Wei Kuo, 2007, "Endurance Enhancement of Flash-Memory Storage Systems: An Efficient Static Wear Leveling Design," ACM/IEEE 44-th Design Automation Conference (DAC), San Diego, USA, June 2007. [Best Paper Nomination]

A Set-Based Mapping Strategy for Downgraded Flash

- An efficient set-based mapping strategy is proposed

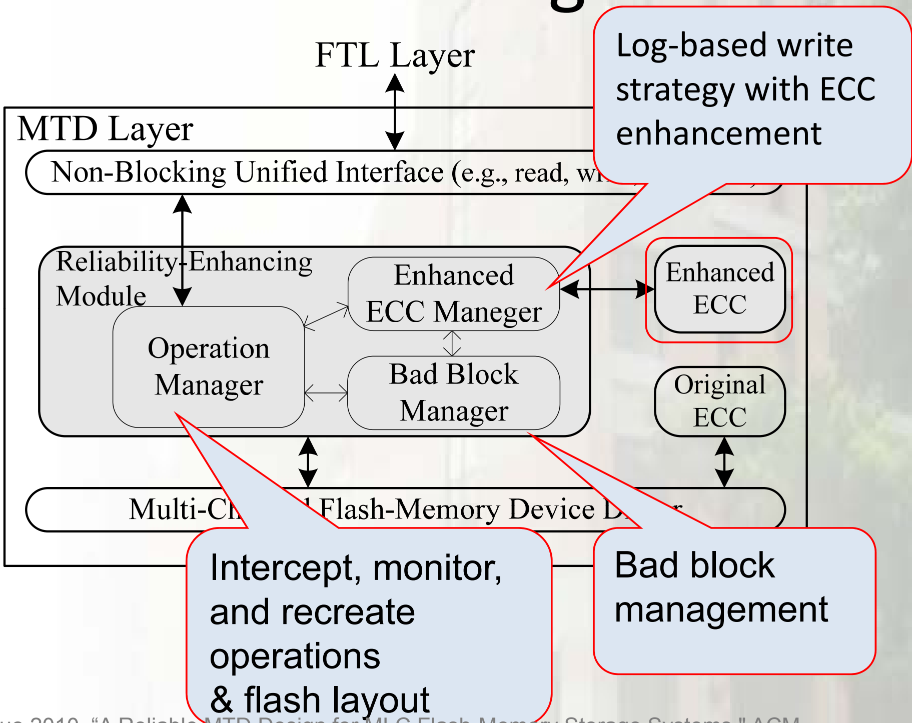




Yuan-Sheng Chu, Jen-Wei Hsieh, Yuan-Hao Chang, and Tei-Wei Kuo, 2009, "A Set-Based Mapping Strategy for Flash-Memory Reliability Enhancement," the ACM/IEEE 12th Conference of Design, Automation, and Test in Europe (DATE), Nice, France, April 20-24, 2009.

Reliability Enhancement – A Reliable MTD Design

- Segment-based mirroring with bad block replacement
- Log-based write strategy with ECC enhancement


FTL Layer
MTD Layer
Non-Blocking Unified Interface (e.g., read, wr
Reliability-Enhancing Module
Enhanced ECC Maneger
Operation Manager
Bad Block Manager
Enhanced ECC
Original ECC
Multi-Ch l Flash-Memory Device D r
Log-based write strategy with ECC enhancement
Intercept, monitor, and recreate operations & flash layout
Bad block management

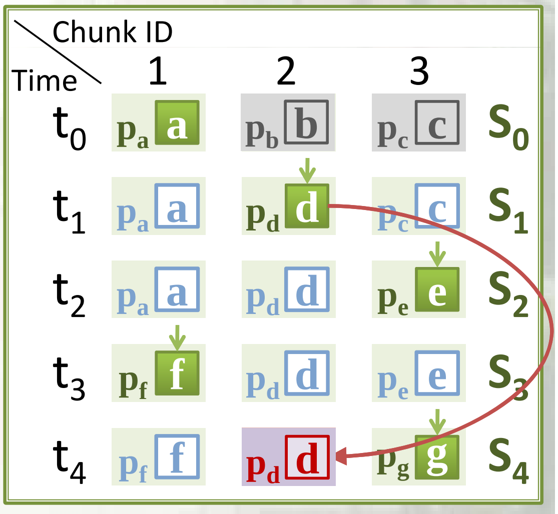
Yuan-Hao Chang and Tei-Wei Kuo 2010, "A Reliable MTD Design for MLC Flash-Memory Storage Systems," ACM International Conference on Embedded Software (EMSOFT), Scottsdale, Arizona, USA, Oct. 24-29, 2010

Reliability Enhancement – Forward Copying in a Native File System

- Duplicate data of the latest version of chunks which affected by the invalidation.

Which chunks need to be forward-copied ?

- Consider the co-existent relation
  - Chunks whose the latest out-of-date version **only co-exist** with the **invalidated page.**
  - Chunks which are latest updated **between the time points** that the invalidated page and the latest out-of-date version.



Time \ Chunk ID	1	2	3	
t_0	p_a a	p_b b	p_c c	S_0
t_1	p_a a	p_d d	p_c c	S_1
t_2	p_a a	p_d d	p_e e	S_2
t_3	p_f f	p_d d	p_e e	S_3
t_4	p_f f	p_d d	p_g g	S_4

Pei-Han Hsu, Yuan-Hao Chang, Po-Chun Huang, Tei-Wei Kuo, David Du, “A Version-based Strategy for Reliability Enhancement of Flash File Systems”, ACM/IEEE DAC 2011.

Conclusion

- What Is Happening?
  - Solid-State Storage Devices
  - New Designs in the Memory Hierarchy
  - Flash-Powered Storage Servers
  - More Applications in Components and Products
- Challenging Issues: Performance, Cost, and Reliability
  - Scalability Technology
  - Reliability Technology
  - Customization Technology

Contact Information

- **Professor Tei-Wei Kuo**
  - ktw@csie.ntu.edu.tw
  - URL: http://csie.ntu.edu.tw/~ktw
  - Flash Research: http://newslab.csie.ntu.edu.tw/~flash/
  - Office: +886-2-23625336-257
  - Fax: +886-2-23628167
  - Address:

    Dept. of Computer Science & Information Engr.
    National Taiwan University, Taipei, Taiwan 106

Questions or Comments?