Implementation and Challenging Issues of Flash-Memory Storage Systems

Tei-Wei Kuo

Department of Computer Science &
Information Engineering
National Taiwan University

Agenda

- Introduction
- Management Issues
- Performance vs Overheads
- Other Challenging Issues
- Conclusion

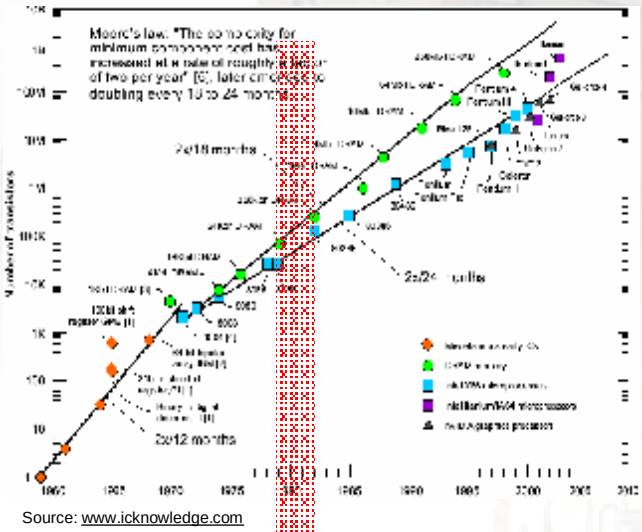
Introduction – Why Flash Memory

- Diversified Application Domains
  - Portable Storage Devices
  - Consumer Electronics
  - Industrial Applications
- SoC and Hybrid Devices
  - Critical System Components






Trends in VLSI Technology

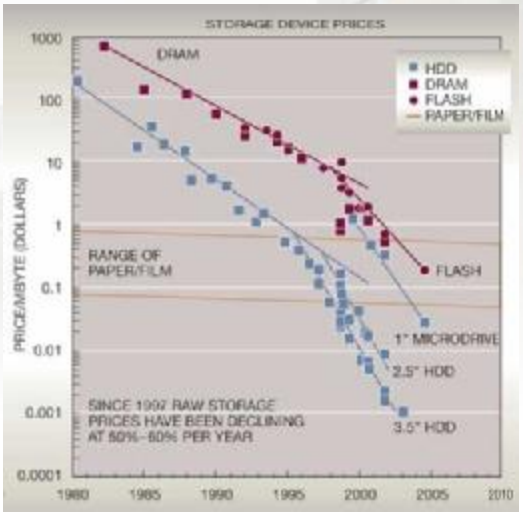


Source: www.icknowledge.com

* This slide was from the ASP-DAC'06 talk delivered by Prof. Sang L. Min from the Seoul National University

Introduction – Trends in Storage Technology

September 2006



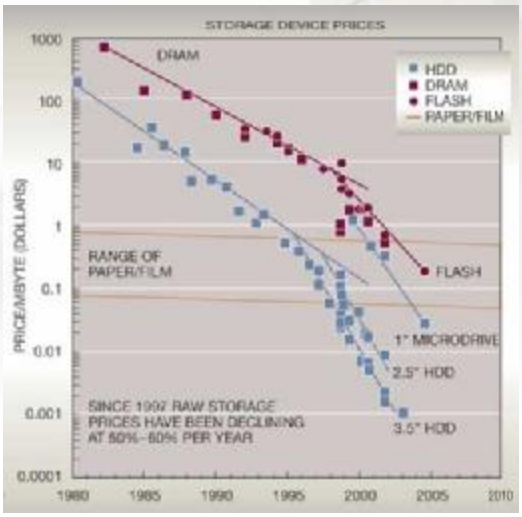
Samsung 2GB USB 2.0 Flash Drive
Price: $49.99
Less Rebate: - $25.00
Final Price: $24.99*

T-One 2GB Microdrive/3600RPM
$144.99

Source: Using multilevel cell NAND flash technology in consumer applications, Electronic Engineering Times, July ,2005

Introduction – Trends in Storage Technology

March 2007



Transcend 8GB
CompactFlash Card
Price: $84.85

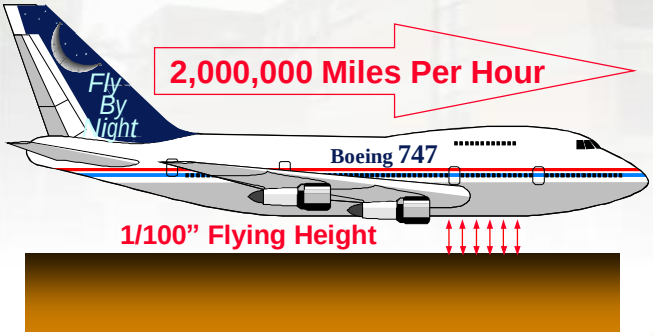
ScanDisk 4GB
CompactFlash Card
Price: $55.99

Microdrive 4GB
Compact Flash Type II
Price: $116

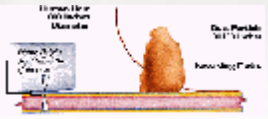
Source: Using multilevel cell NAND flash technology in consumer applications, Electronic Engineering Times, July ,2005. Amazon.com

The Ultimate Limit on Mechanical Devices

u A Microdrive Example



Source: Richard Lary, The New Storage Landscape: Forces shaping the storage economy, 2003.



Source: http://www.hitachigst.com/

* This slide was from the ASP-DAC'06 talk delivered by Prof. Sang L. Min from the Seoul National University.

Introduction – The Characteristics of Storage Media

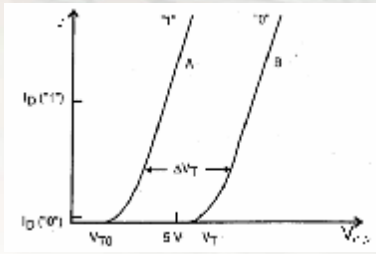
Media	Access time		
	Read	Write	Erase
DRAM	60ns (2B) 2.56*us* (512B)	60ns (2B) 2.56*us* (512B)	-
NOR FLASH 15X	150ns (1B) 14.4*us* (512B)	211*us* (1B) 3.52ms (512B)	1.2s (16KB)
NAND FLASH 100X 50X	10.2*us* (1B) 35.9*us* (512B)	201*us* (1B) 226*us* (512B)	2ms (16KB)
DISK 400X	12.4ms (512B) (average)	12.4 ms(512B) (average)	-

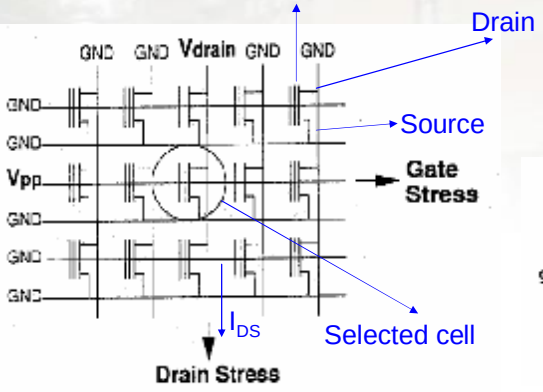
[Reference] DRAM:2-2-2 PC100 SDRAM. NOR FLASH: Intel 28F128J3A-150. NAND FLASH: Samsung K9F5608U0M. Disk: Segate Barracuda ATA II.[1]

1. J. Kim, J. M. Kim, S. H. Noh, S. L. Min, and Y. Cho. A space-efficient flash translation layer for compact-flash systems. *IEEE Transactions on Consumer Electronics*, 48(2):366–375, May 2002.

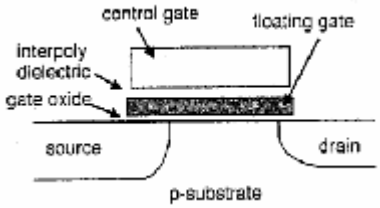
Introduction – Single-Level Cell (SLC)

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



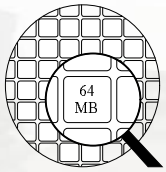


Cell



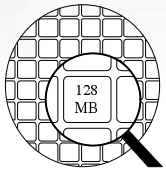
Introduction – Multi-Level Cell (MLC) vs SLC

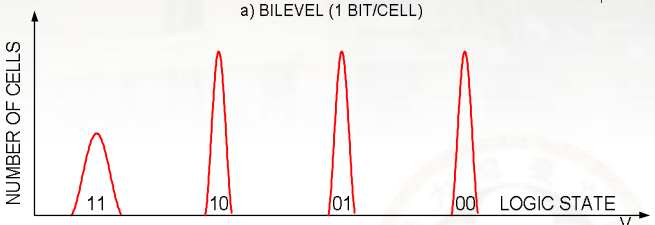
SLC Flash



a) BILEVEL (1 BIT/CELL)

MLC Flash





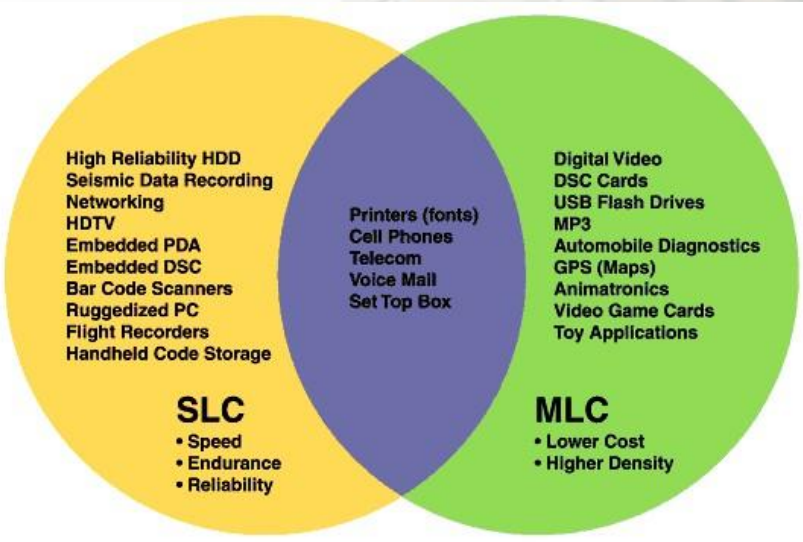
b) MULTILEVEL (2 BIT/CELL)

Comparison of SLC and MLC

- 1-bit/Cell SLC NAND Flash
  - 100,000 Program/Erase cycles (with ECC)[1]
  - 10 years Data Retention[1]
- 2-bits/Cell MLC NAND Flash
  - 10,000 Program/Erase cycles (with ECC) [2]
  - 10 years Data Retention[2]
- 4-bits/Cell MLC NAND FLASH Developers (2006)
  - M-systems, Intel, Samsung, and Toshiba

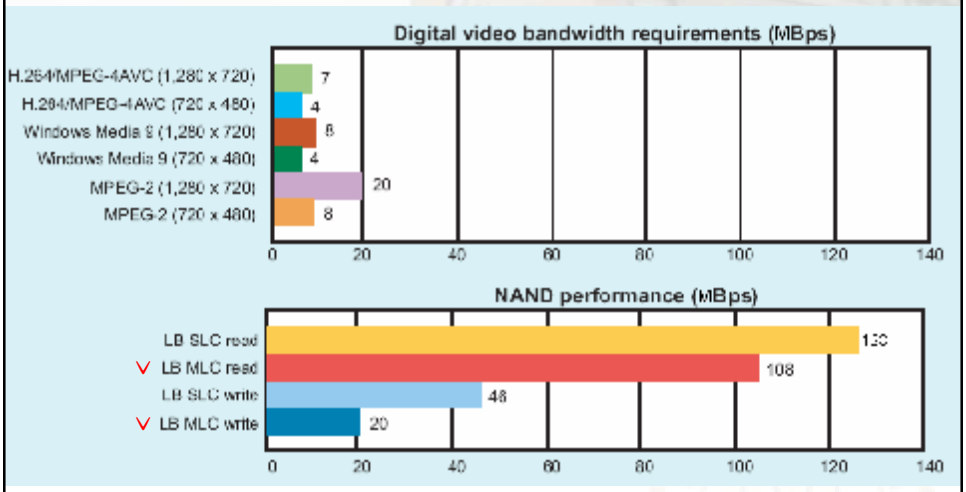
[1] ST Micro-electronics NAND SLC large page datasheet (NAND08GW3B2A)
[2] ST Micro-electronics NAND MLC large page datasheet (NAND04GW3C2A)
* USD34.65 per GB for NOR, USD6.79 per GB for NAND

Introduction – Consumer Applications



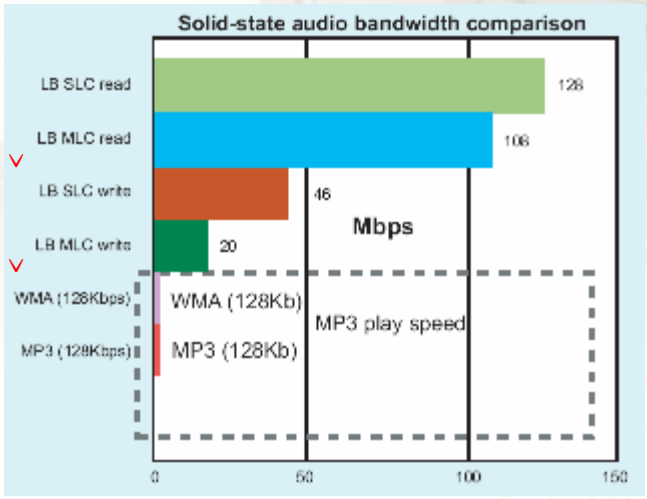
Electronic Engineering Times, July 2005

Bandwidth Requirements – Video



Electronic Engineering Times, July 2005

Bandwidth Requirements – Audio



Electronic Engineering Times, July 2005

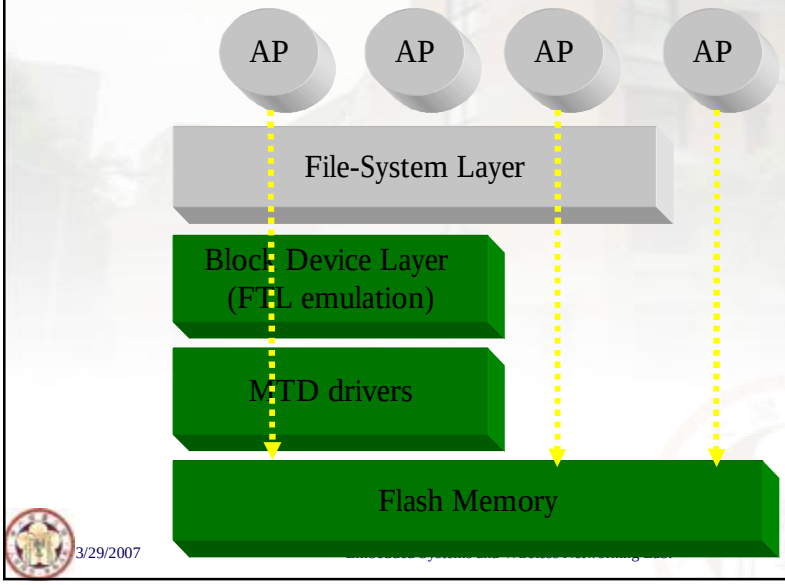
Introduction – Challenges in Flash-Memory Storage Designs

- Requirements in Good Performance
- Limited Cost per Unit
- Strong Demands in Reliability
- Increasing in Access Frequencies
- Tight Coupling with Other Components
- Low Compatibility among Vendors

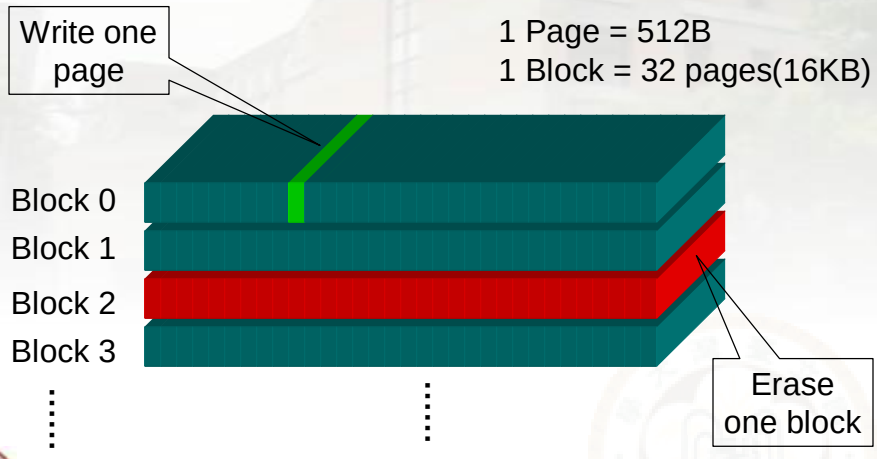
Agenda

- Introduction
- Management Issues
- Performance vs Overheads
- Other Challenging Issues
- Conclusion

Management Issues – System Architectures



Management Issues – Flash-Memory Characteristics

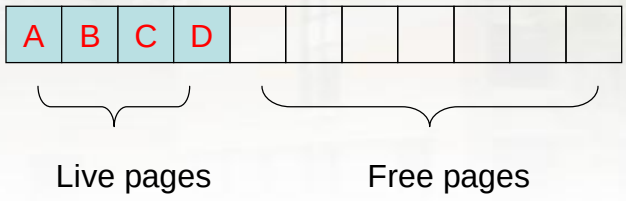


Management Issues – Flash-Memory Characteristics

- Write-Once
  - No writing on the same page unless its residing block is erased!
  - Pages are classified into 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.

Management Issues – Flash-Memory Characteristics

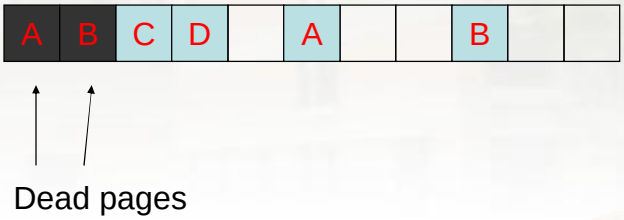
- Example 1: Out-place Update



Suppose that we want to update data A and B…

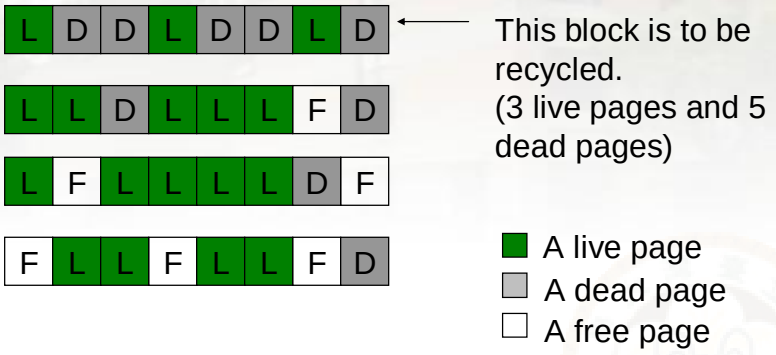
Management Issues – Flash-Memory Characteristics

- Example 1: Out-place Update



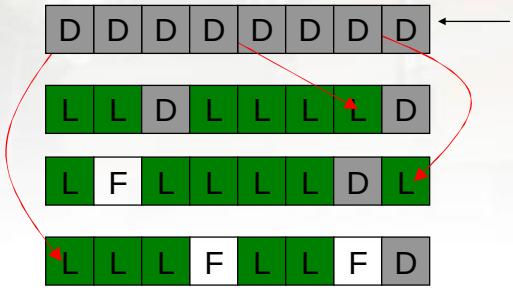
Management Issues – Flash-Memory Characteristics

- Example 2: Garbage Collection



Management Issues – Flash-Memory Characteristics

- Example 2: Garbage Collection

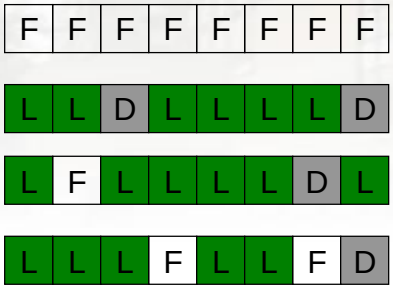


Live data are copied to somewhere else.

- A live page
- A dead page
- A free page

Management Issues – Flash-Memory Characteristics

- Example 2: Garbage Collection



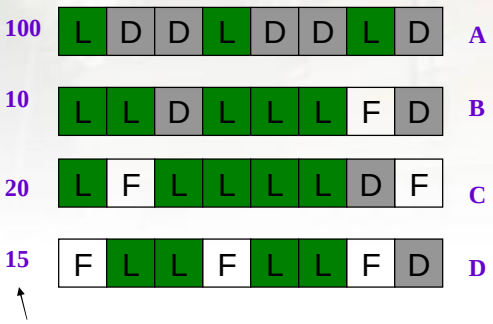
The block is then erased.

Overheads:
- live data copying
- block erasing.

- A live page
- A dead page
- A free page

Management Issues – Flash-Memory Characteristics

- Example 3: Wear-Leveling



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

- A live page
- A dead page
- A free page

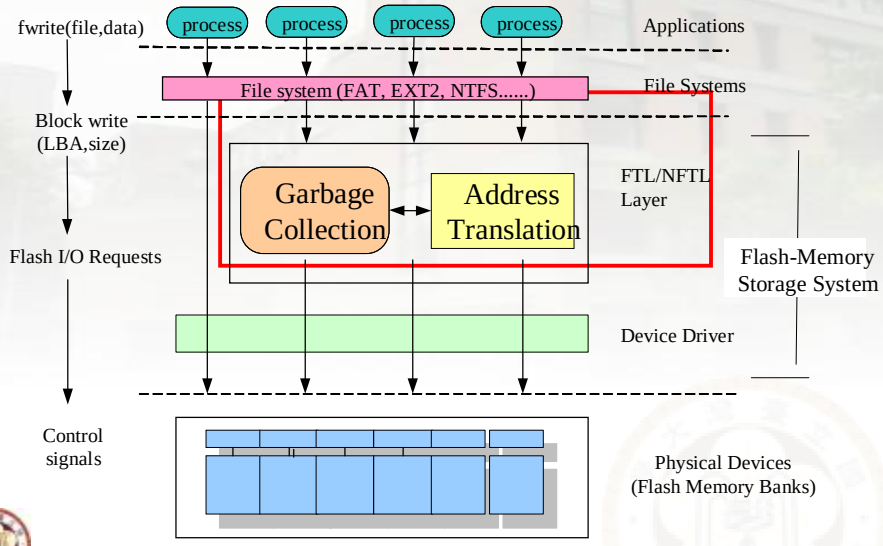
Management Issues – Challenges

- 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!

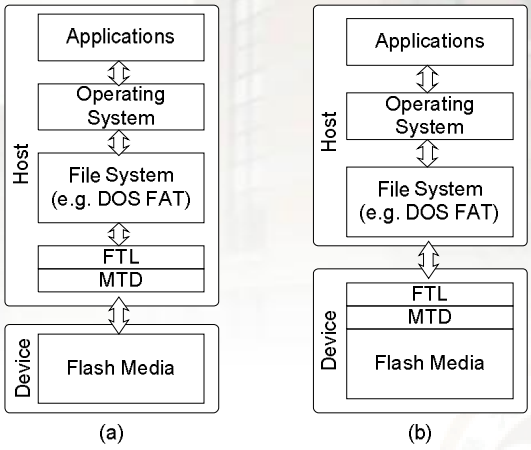
Agenda

- Introduction
- Management Issues
- Performance vs Overheads – FTL vs NFTL
- Other Challenging Issues
- Conclusion

System Architecture



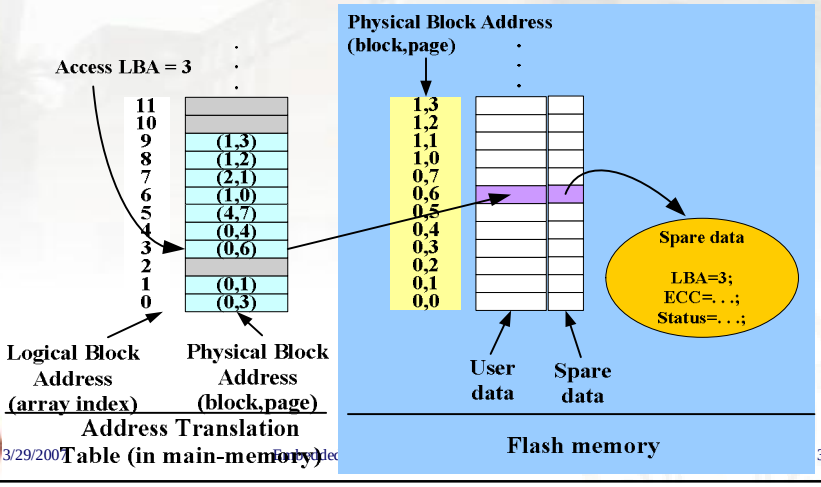
Management Issues – Flash-Memory Characteristics



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

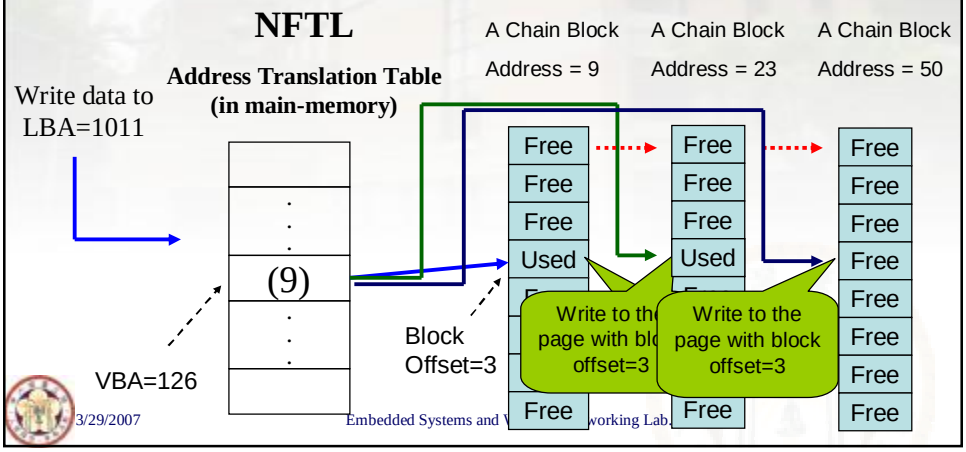
Policies – FTL

- 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.



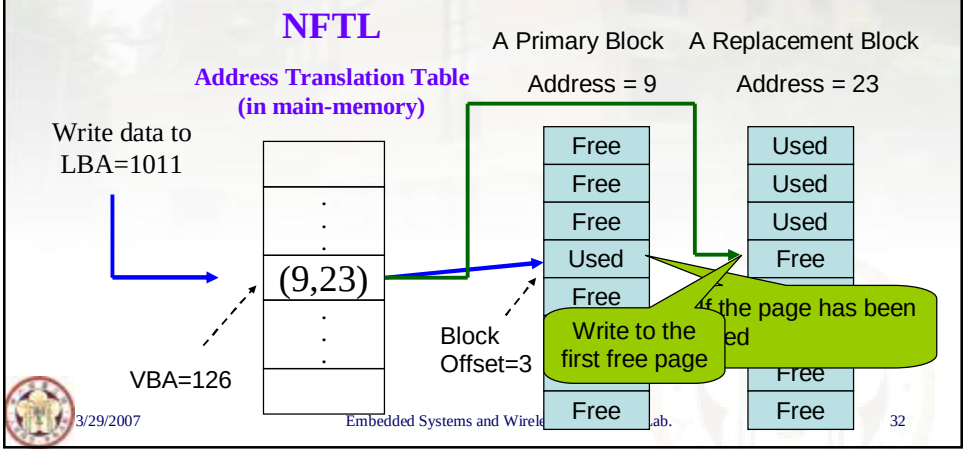
Policies – NFTL (Type 1)

- 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



Policies – NFTL (Type 2)

- 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



Policies – NFTL

- NFTL is proposed for the large-scale NAND flash storage systems because NFTL adopts a block-level address translation.
- However, the address translation performance of read and write requests might deteriorate, due to linear searches of address translation information in primary and replacement blocks.

Policies – FTL or NFTL

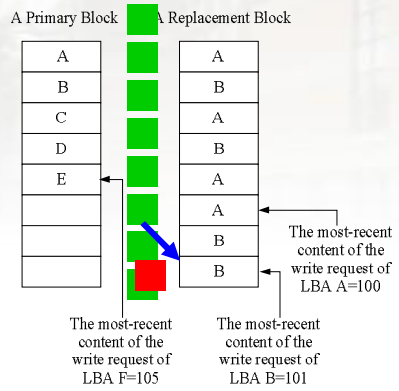
	FTL	NFTL
Memory Space Requirements	Large	Small
Address Translation Time	Short	Long
Garbage Collection Overhead	Less	More
Space Utilization	High	Low

- The Memory Space Requirements for one 1GB NAND (512B/Page, 4B/Table Entry, 32 Pages/Block)
  - FTL: 8,192KB (= 4*(1024*1024*1024)/512)
  - NFTL: 256KB (= 4*(1024*1024*1024)/(512*32))

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 2KB, and there are 128 pages per block.

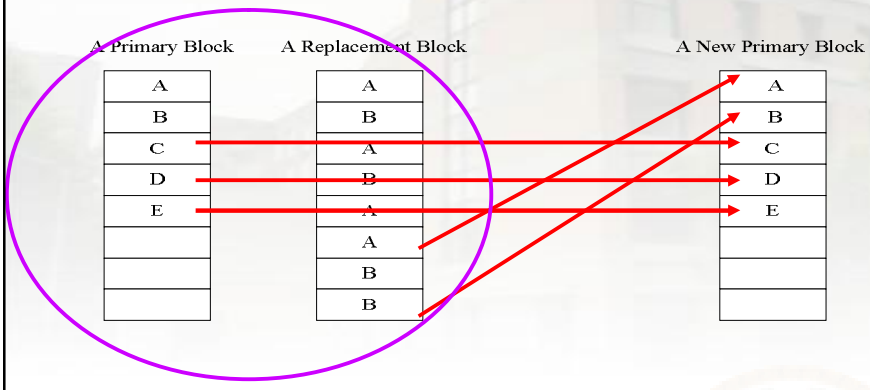
Address Translation Time - NFTL

- **The address translation performance of read and write requests can be deteriorated, due to linear searches of physical addresses.**



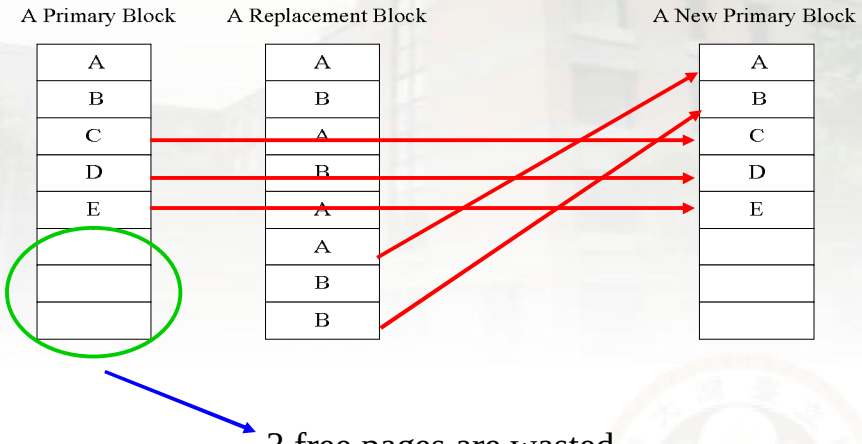
1. Assume that each block contains 8 pages.
2. Let LBA A, B, C, D, and E be written for 5, 5, 1, 1, and 1 times, respectively. Their data distribution could be like to what in the left figure.
3. For example, it might need to scan 9 spare areas for LBA B.

Garbage Collection Overhead - NFTL



1. Copy the most-recent content to the new primary block.
2. Erase the old primary block and the replacement block.
3. Overhead is 2 block erases and 5 page writes.

Space Utilization - NFTL



Agenda

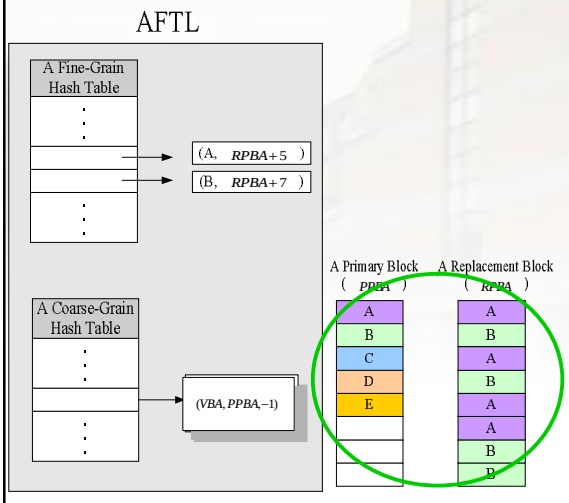
- Introduction
- Management Issues
- Performance vs Overheads – An Adaptive Two-Level Mapping Mechanism
- Other Challenging Issues
- Conclusion

Motivation

- **An adaptive two-level management design of a flash translation layer, called AFTL.**
  - Exploit the advantages of the fine-grained address mechanism and the coarse-grained address mechanism.

	FTL	NFTL	AFTL
Memory Space Requirements	**Large**	**Small**	**A little larger than NFTL**
Address Translation Time	**Short**	**Long**	**Much Better than NFTL**
Garbage Collection Overhead	**Less**	**More**	**Much Better than NFTL**
Space Utilization	**High**	**Low**	**Much Better than NFTL**

AFTL – Coarse-to-Fine Switching

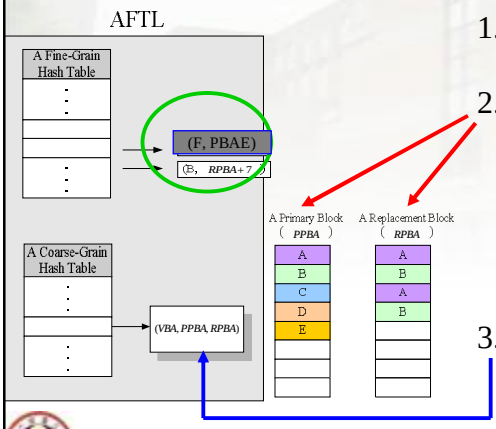


1. AFTL doesn't erase the two blocks immediately.
2. AFTL moves the mapping information of the replacement block to the fine-grained hash table by adding fine-grained slots.
3. The RPBA field of the corresponding mapping information is nullified.

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.

AFTL – Fine-to-Coarse Switching

- **The number of the fine-grained slots is limited.**
  - Some least recently used mapping information of fine-grained slots should be moved to the coarse-grained hash table.



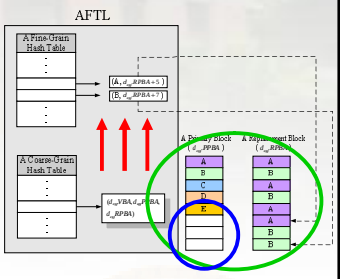
1. Assume that this fine-grained slot is to be replaced.
2. Data stored in the page with the given (physical) address are copied to the primary or replacement block of the corresponding coarse-grained slot, as defined by NFTL.
3. If there dose not exist any corresponding coarse-grained slot, a new one is created.

AFTL – Fine-to-Coarse Switching

- **Coarse-to-fine switches would introduce fine-to-coarse switches and overhead in valid page copying.**
  - It is because the number of the fine-grained slots is limited.
- **Stop any coarse-to-fine switch when some frequency bound in coarse-to-fine switches is reached.**
  - We set a parameter in the experiments to control the frequency of switches to explore the behavior of the proposed mechanism.

The Advantages of AFTL

- **Improve the address translation performance.**
  - It is because the moving of their mapping information to the fine-grained hash table.
- **Improve the garbage collection overhead.**
  - The delayed recycling of any replacement block reduces the potential number of valid data copyings and blocks erased.
- **Improve the space utilization.**
  - The delayed recycling of any primary block lets free pages of a primary block be likely used in the future.



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.

Agenda

- Introduction
- Management Issues
- Performance vs Overheads – Performance Evaluation
- Other Challenging Issues
- Conclusion

Performance Evaluation

Performance Setup

- The characteristics of the experiment trace was over a 20GB disk.

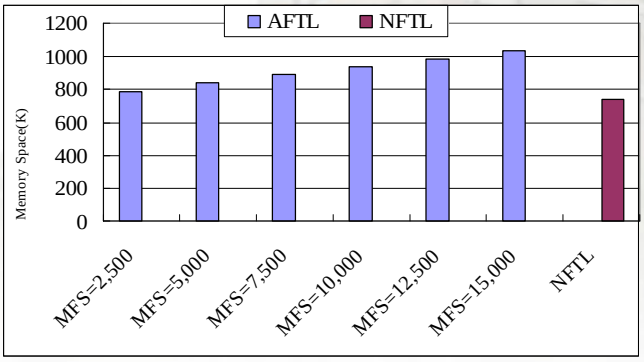
CPU	**Intel Celeron 750MHz**
RAM	**320 MB**
OS	**Windows XP**
File Systems	**NTFS**
Applications	**Web Applications, E-mail Clients, MP3 Player, MSN Messenger, Word, Excel, PowerPoint, Media, Player, Programming, and Virtual Memory Activities**
Durations	**One week**
Total Write / Read Requests	**13,198,805 / 2,797,996 sectors**
Different LBA's	**1,669,228**

Performance Evaluation

Performance Setup

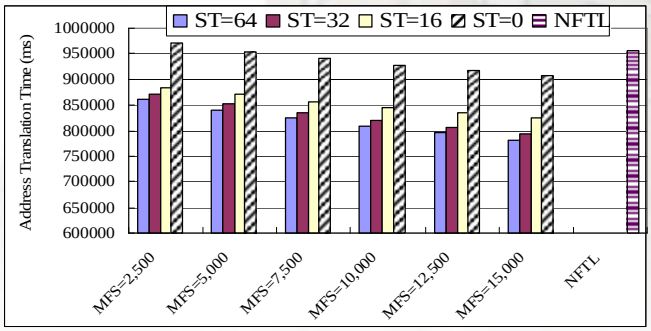
- The maximum number of fine-grained slots is controlled by a parameter ***MFS***.
- A parameter ***ST*** controls the frequency of switches between the two address translation mechanisms – n/***ST***.
  - ST=0 => No constraint on the number of switches.
  - Smaller ST => More switches.
  - Larger ST => Less switches.

Memory Space Requirements



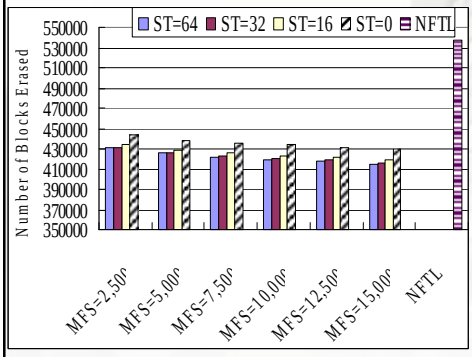
1. *MFS* **ranged from 2,500, 5,000, 7,500, 10,000, 12,500, to 15,000.**
2. **AFTL uses a little more memory space than NFTL.**

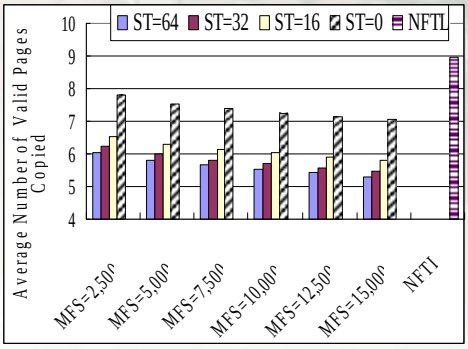
Address Translation Performance



1. Larger ***MFS*** => smaller address translation time
   - More address translations going through the fine-grained address translation mechanism.
2. Smaller ***ST*** => longer address translation time
   - More coarse-to-fine switches

Garbage Collection Overhead

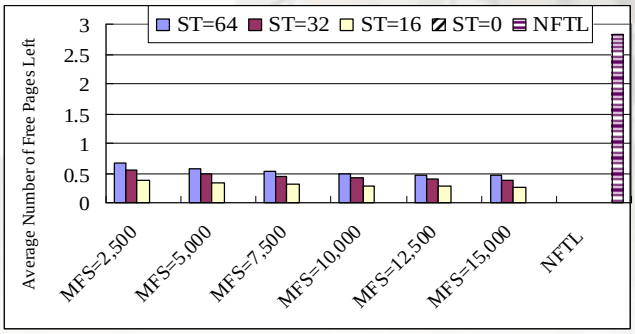




AFTL outperforms NFTL.

- Coarse-to-fine switches can avoid immediate recycling of their primary and replacement blocks and related valid data copyings.

Space Utilization

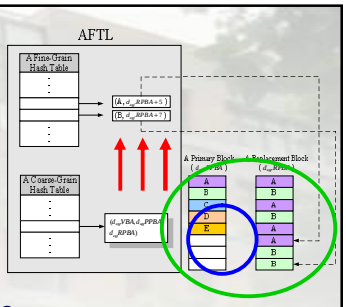


The Space utilization might be better under AFTL.

- Coarse-to-fine switches can delay the recycling of replacement blocks.

- Free pages of primary blocks might be used in the future..

Summary



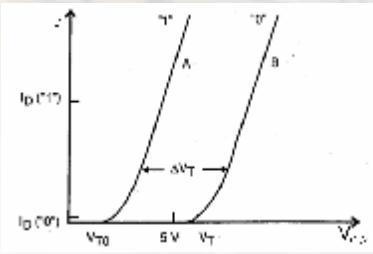
- **AFTL is proposed to**
  - exploit the advantages of fine-grained/coarse-grained address translation mechanisms, and to
  - switch dynamically and adaptively the mapping information between the two address translation mechanisms.
- **AFTL does provide good performance in address mapping and space utilization and have garbage collection overhead and memory space requirements under proper management.**

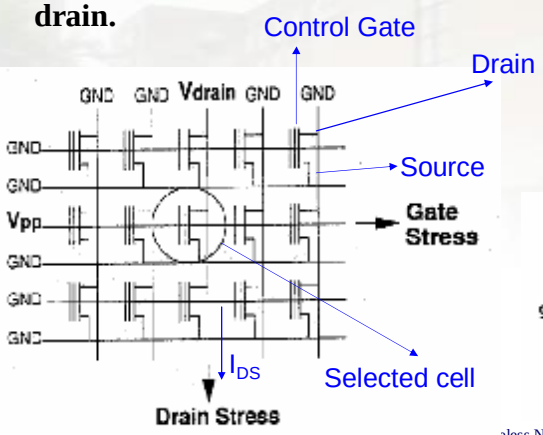
Agenda

- Introduction
- Management Issues
- Performance vs Overheads
- Other Challenging Issues
- Conclusion

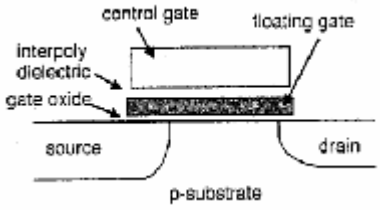
Challenging Issues – Reliability

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



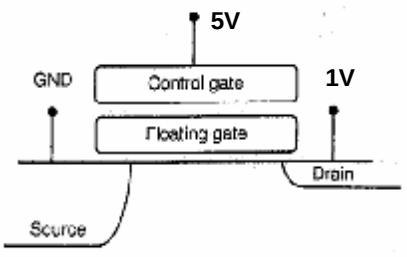


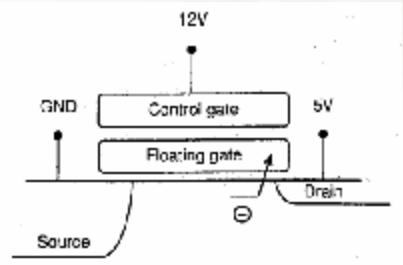
Cell



Challenging Issues – Reliability

- Read Operation
  - When the floating gate is not charged with electrons, there is current I_D (100*uA)* if a reading voltage is applied. ("1" state)
- Program Operation
  - Electrons are moved into the floating gate, and the threshold voltage is thus raised.





Challenging Issues – Reliability

- Over-Erasing Problems
  - Fast Erasing Bits **à** All of the cells connected to the same bit line of a depleted cell would be read as "1", regardless of their values.
- Read/Program Disturb Problems
  - DC erasing of a programmed cell, DC programming of a non-programmed cell, drain disturb, etc.
  - Flash memory that has thin gate oxide makes disturb problems more serious!
- Data Retention Problems
  - Electrons stored in a floating gate might be lost such that the lost of electrons will sooner or later affects the charging status of the gate!

Challenging Issues – Observations

- 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!
- Wear-leveling technology is even more critical when flash memory is adopted in many system components or might survive in products for a long life time!

Wear Leveling versus Product Lifetime

- Settings
  - File system: FAT16 file system with 8KB cluster size
  - Flash memory: 256MB small-block flash memory with 100K erase cycles
- Updating of a 16MB file repeatedly with the throughput: 0.1MBs
  - The file requires 2K clusters = 16MB ÷ 8KB(cluster size)
  - The FAT size of this file is 4KB (2K(clusters) x 2 bytes)
  - The 20% of blocks in flash memory joins the dynamic wear leveling
  - Data of a 16MB file is stored in 1K blocks (16MB ÷ 16KB(block size))
  - Suppose flash memory is managed in the block level
    - File systems update the FAT in each cluster writing so that FAT is updated 2K times for a 16MB file
    - Writing of a 16MB incurs 1K block erases because of the reclaiming of invalid space.

Wear Leveling versus Product Lifetime

- Ways in Data Updates
  - In-Place-Updates: Rewriting on the Same Page
  - Dynamic Wear Leveling: Rewriting over Another Free Page with Erasing over Blocks with Dead Pages
  - Static Wear Leveling: Rewriting over Another Free Page with Erasing over Any Blocks
- Expected Lifetime of

$$\text{NO Wear Leveling (in-place update)} = \frac{16(\text{MB})}{0.1(\text{MB/second})} \times \frac{100K}{2K \times 24 \times 60 \times 60} \approx 0.09(\text{days})$$

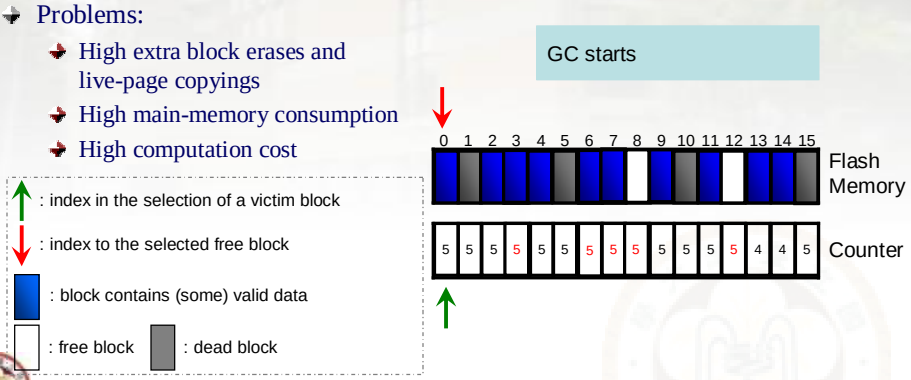
$$\text{Dynamic Wear Leveling} = \frac{16(\text{MB})}{0.1(\text{MB/second})} \times \frac{16K(\text{blocks}) \times 20\% \times 100K}{(2K+1K) \times 24 \times 60 \times 60} \approx 197.5(\text{days})$$

$$\text{Static Wear Leveling} = \frac{16(\text{MB})}{0.1(\text{MB/second})} \times \frac{16K(\text{blocks}) \times 100\% \times 100K}{(2K+1K) \times 24 \times 60 \times 60} \approx 987.5(\text{days})$$

Wear Leveling versus Product Lifetime

Static Wear Leveling – Block-Level Mapping

- Use a counter for each block
- The garbage collector always finds the block with the least erase count.
  - Some heuristic approach erases a block to maintain 2 free blocks when the garbage collector finds the erase count of the block is over a given threshold.
- Problems:
  - High extra block erases and live-page copyings
  - High main-memory consumption
  - High computation cost



Conclusion

- Summary
  - The Characteristics of Flash Memory and Management Issues
  - Popular Implementations: FTL vs NFTL
  - Adaptive Two-Level Address Translation
  - Performance, Cost, and Reliability Challenges

Conclusion

- What Is Happening?
  - Solid-State Storage Devices
  - New Designs in the Memory Hierarchy
  - More Applications in System 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


Q & A