# Operating System Practice

Che-Wei Chang

chewei@mail.cgu.edu.tw

Department of Computer Science and Information Engineering, Chang Gung University

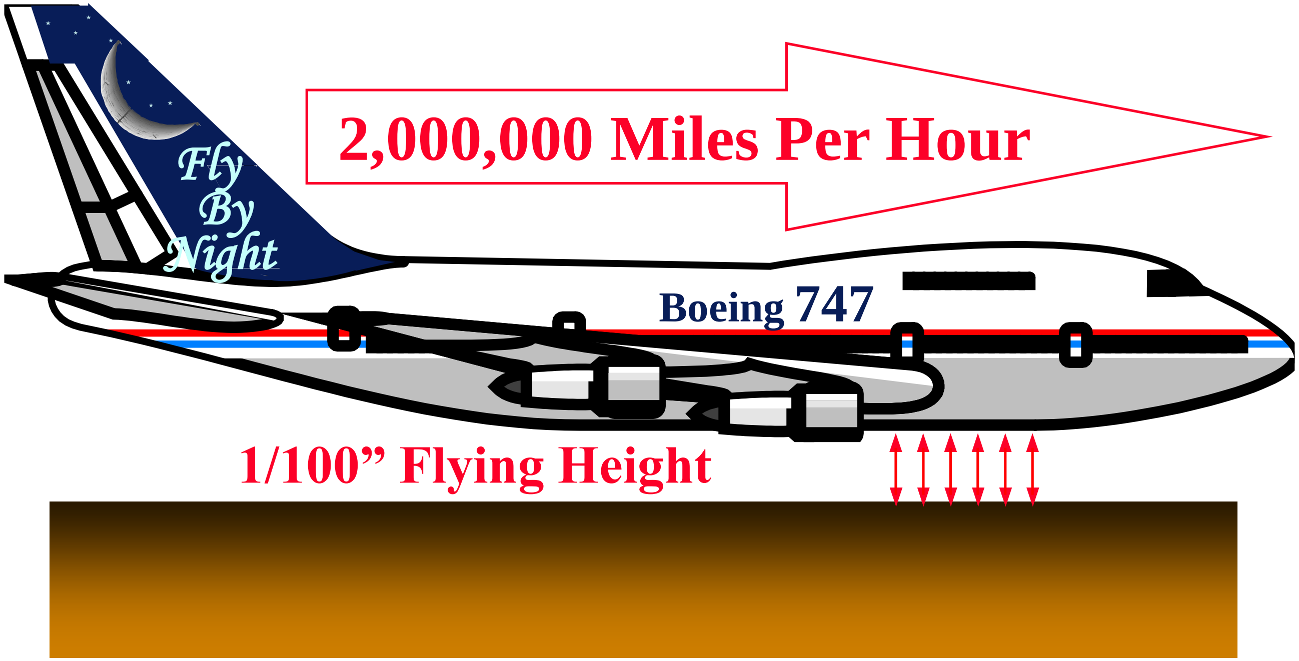
# Flash Memory and Phase Change Memory

Reference: Prof. Tei-Wei Kuo, NTU and Dr. Yuan-Hao Chang, Academia Sinica

# Trends – Market and Technology

- Diversified Application Domains
  - Portable Storage Devices
  - Consumer Electronics
  - Industrial Applications
- Competitiveness in the Price
  - Dropping Rate and the Price Gap with HDDs
- Technology Trend over the Market
  - Improved density
  - Degraded performance
  - Degraded reliability

# Trends – Storage Media



2,000,000 Miles Per Hour

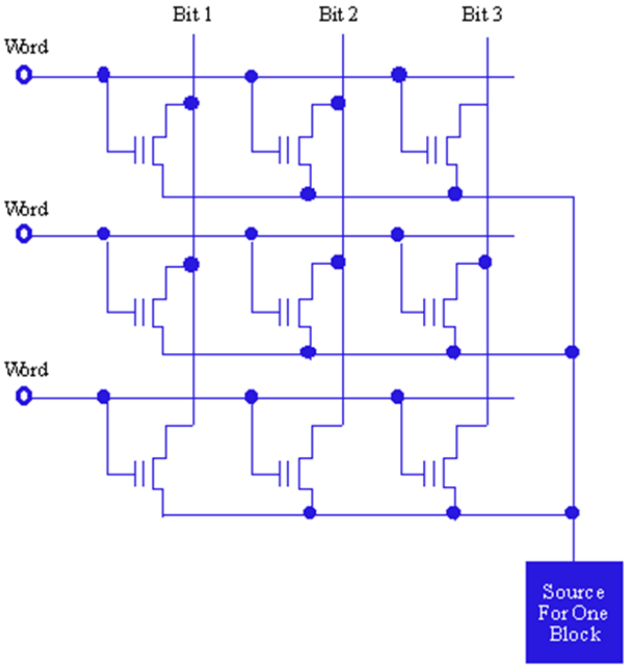
1/100” Flying Height

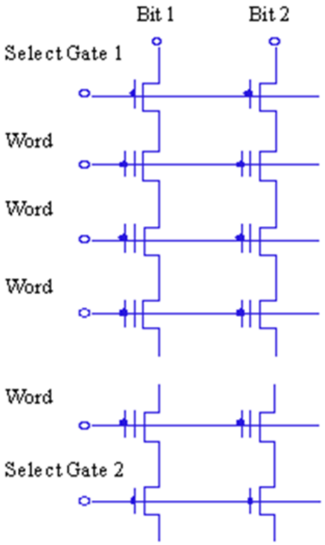
VS

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

# NOR and NAND Flash

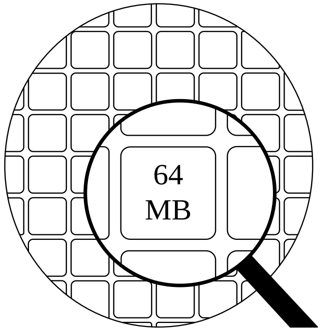
- NAND accesses each cell through adjacent cells, while NOR allows for individual access to each cell
- The cell size of NAND is almost half  the size of a NOR cell

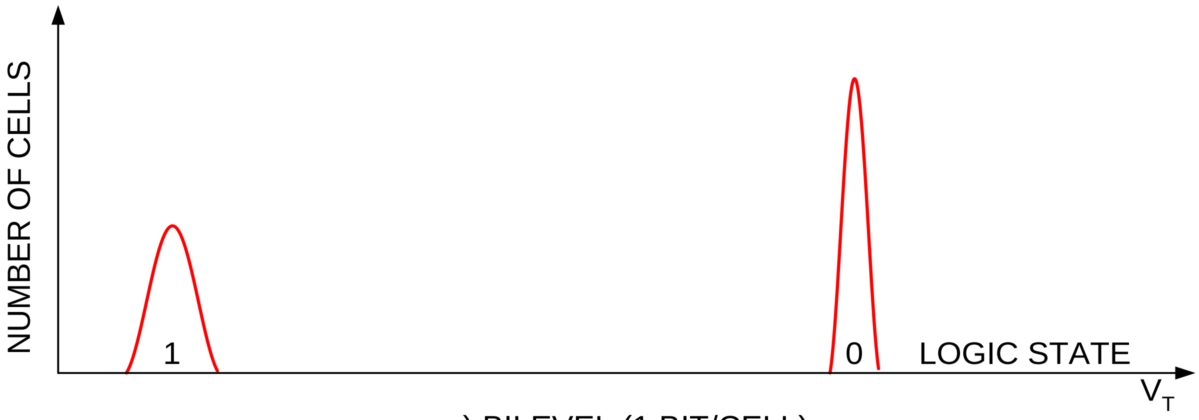




# Single-Level Cell (SLC) vs Multi-Level Cell (MLC) Flash

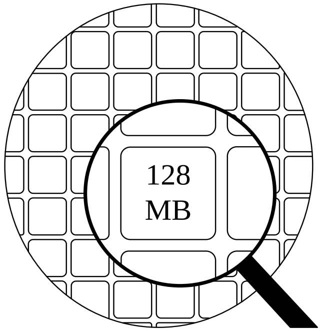
SLC Flash

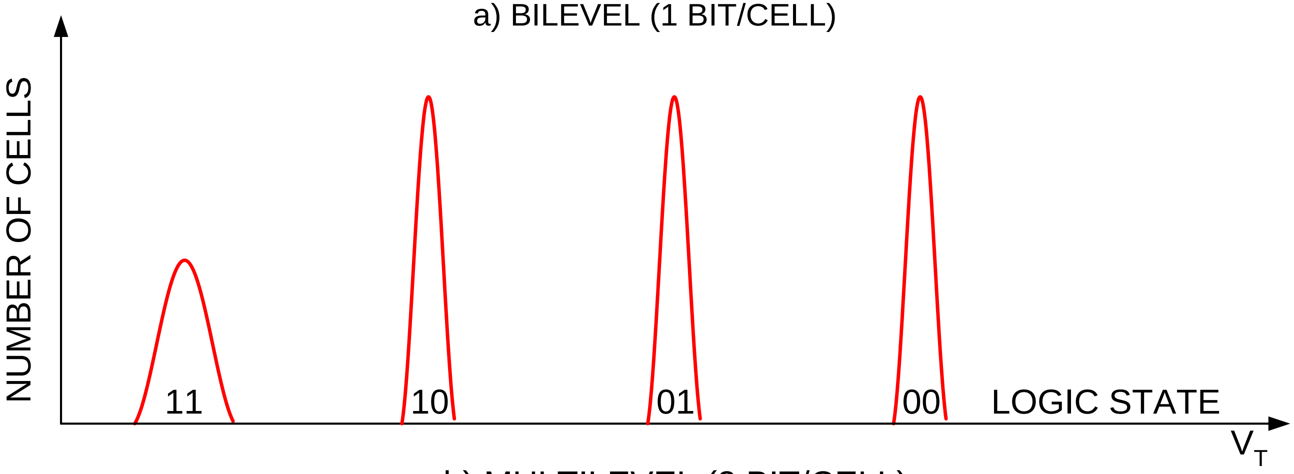




a) BILEVEL (1 BIT/CELL)

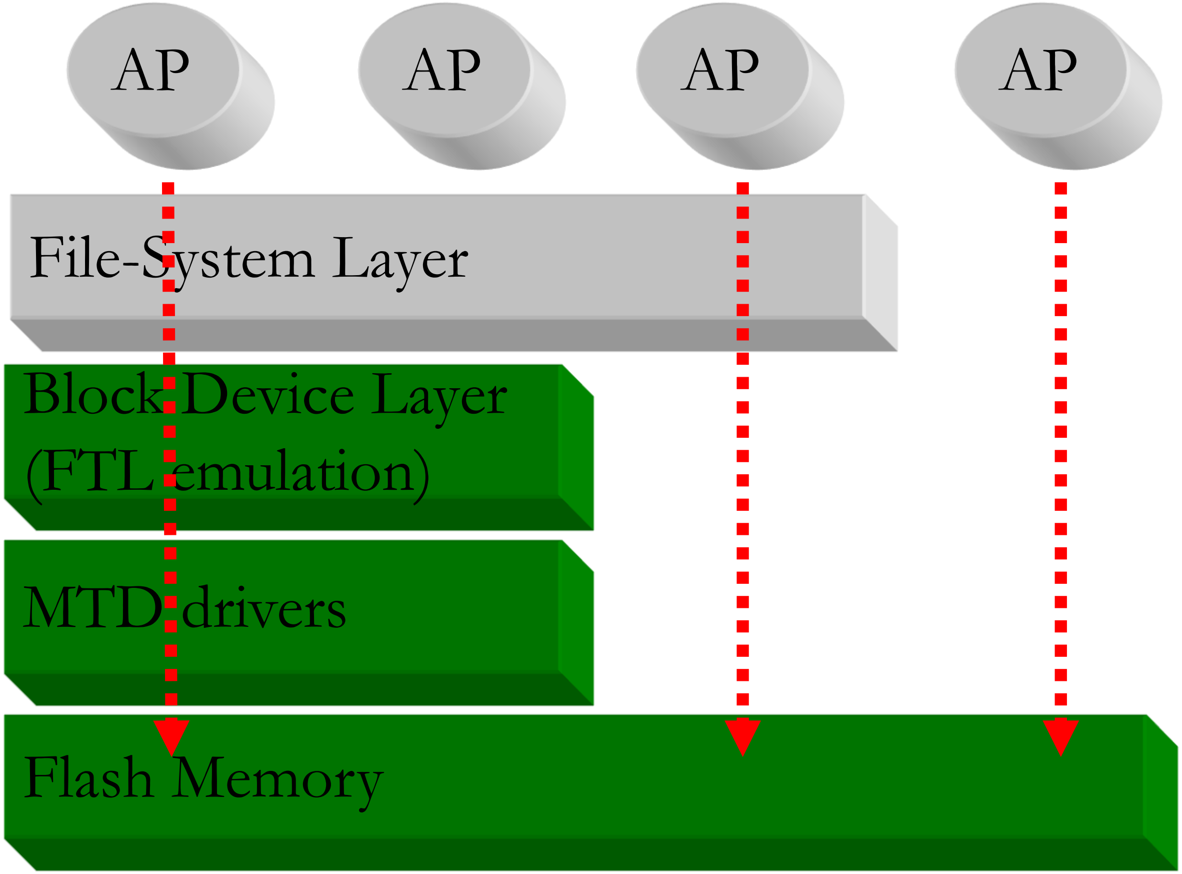
MLC Flash





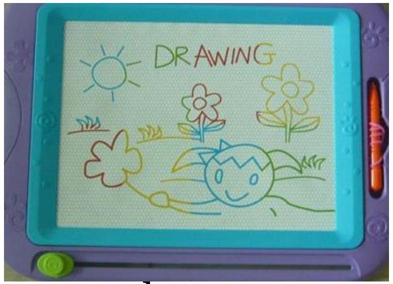
b) MULTILEVEL (2 BIT/CELL)

# System Architectures for Flash Management



# 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

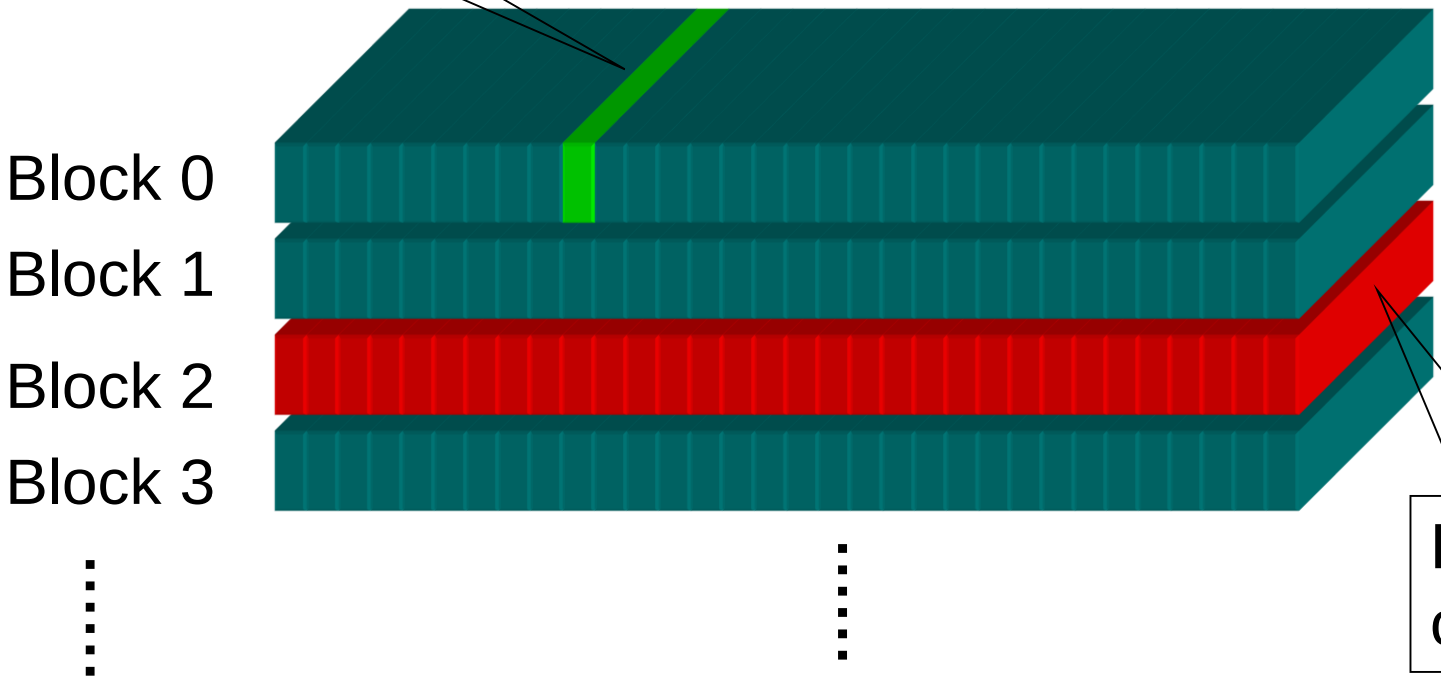


taobao.com

- Wear-Leveling
  - Each block has a limited lifetime in erasing counts

# Page Write and Block Erase

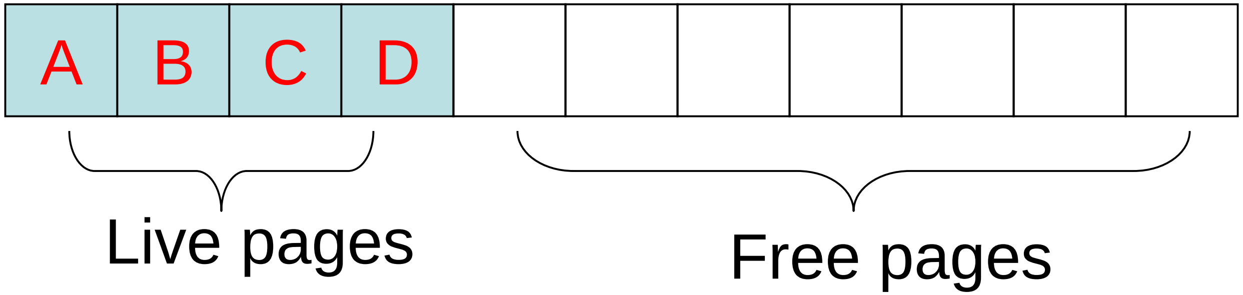
1 Page = 512B—4KB
1 Block = 32 pages



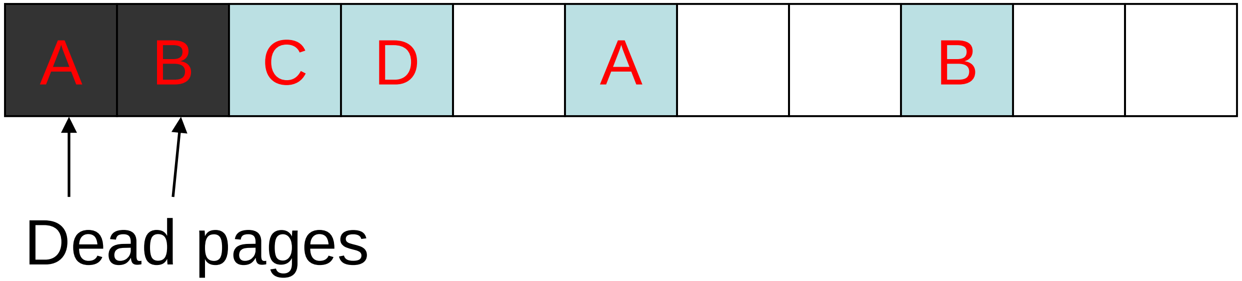
Write one page

Erase one block

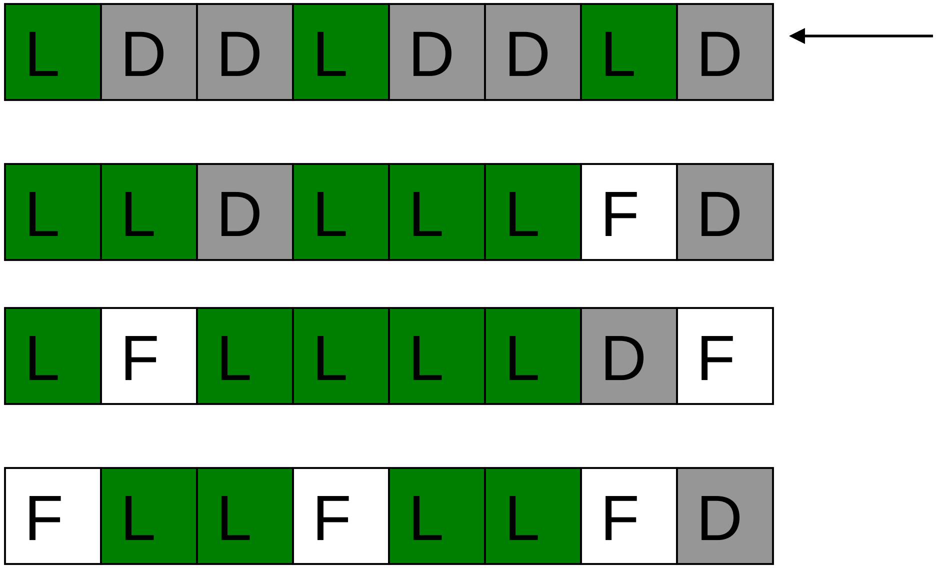
# Out-Place Update



**Suppose that we want to update data A and B…**



# Garbage Collection (1/3)

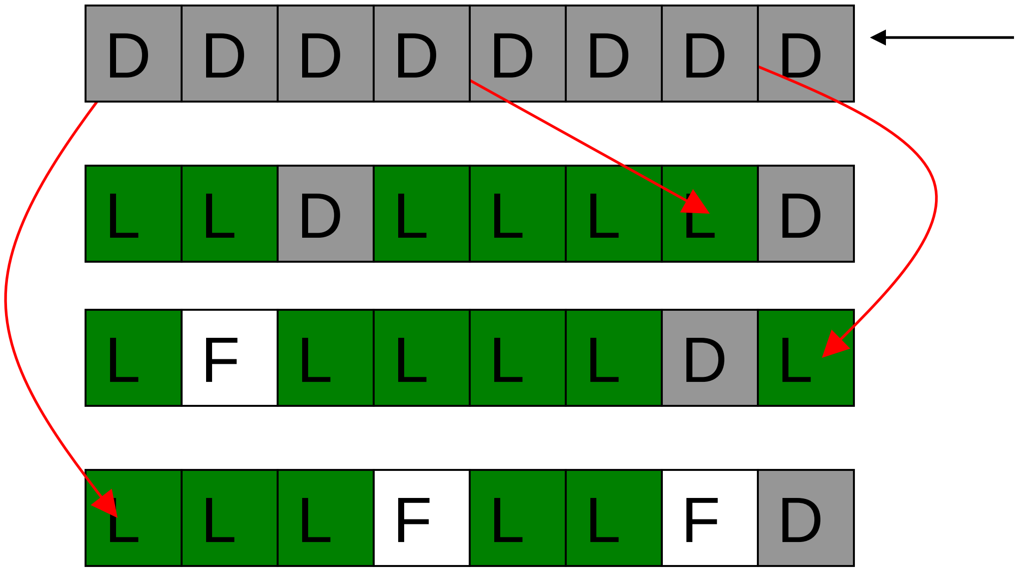


|  |  |  |  |  |  |  |  |
|---|---|---|---|---|---|---|---|
| L | D | D | L | D | D | L | D |
| L | L | D | L | L | L | F | D |
| L | F | L | L | L | L | D | F |
| F | L | L | F | L | L | F | D |

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

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

# Garbage Collection (2/3)



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

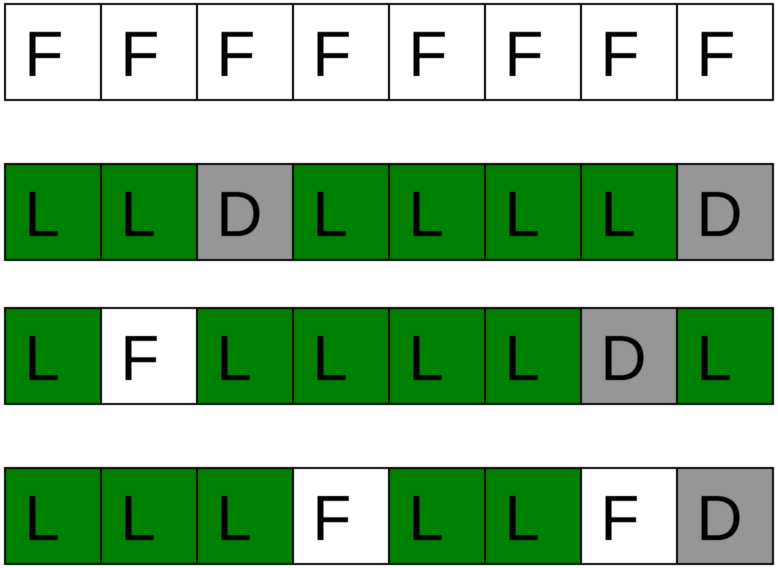
Live data are copied to somewhere else

🟩 A live page

▨ A dead page

□ A free page

# Garbage Collection (3/3)



| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 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

# Wear-Leveling

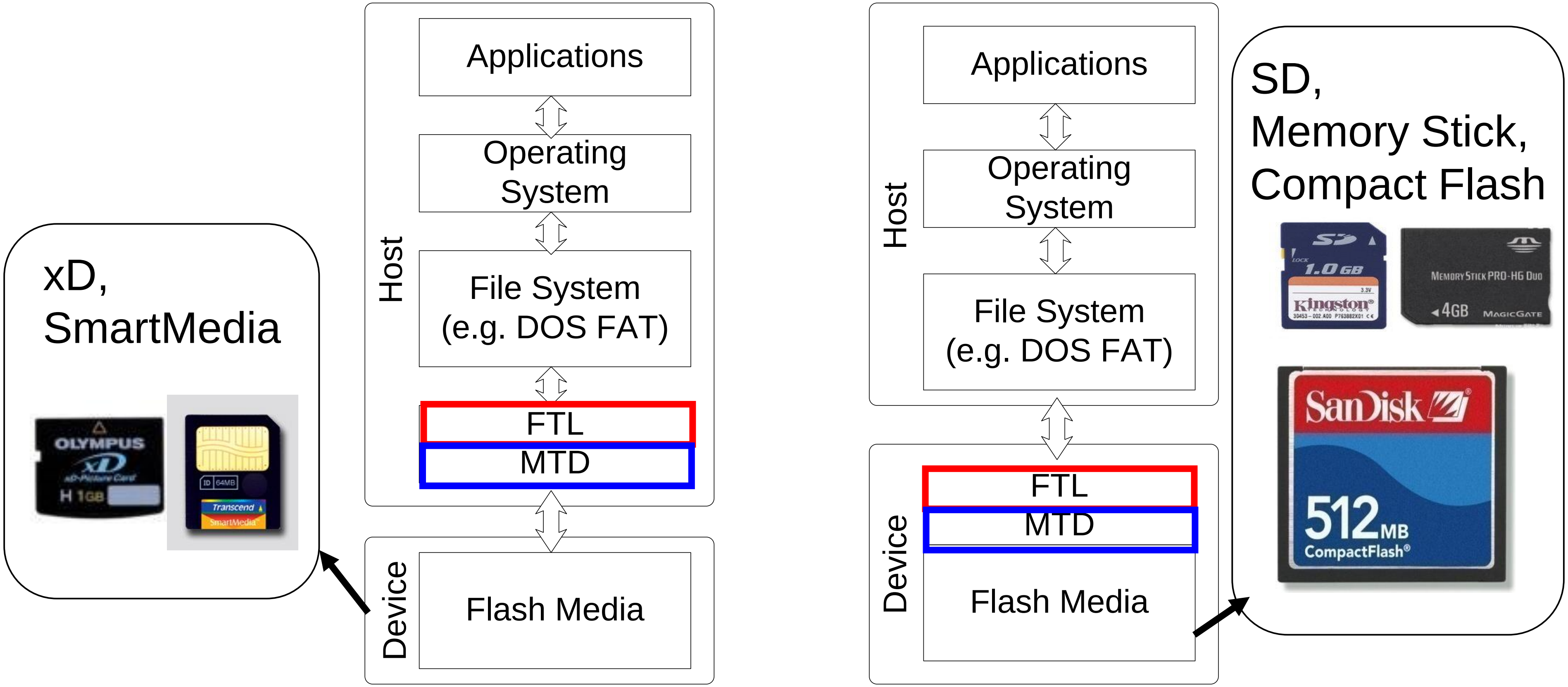
|  |  |  |  |  |  |  |  |  |  |
|---|---|---|---|---|---|---|---|---|---|
| 100 | L | D | D | L | D | D | L | D | A |
| 10 | L | L | D | L | L | L | F | D | B |
| 20 | L | F | L | L | L | L | D | F | C |
| 15 | F | L | L | F | L | L | F | D | D |

Erase cycle counts

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

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

# Flash Translation Layer



xD,
SmartMedia

Host

Applications

Operating System

File System (e.g. DOS FAT)

FTL

MTD

Device

Flash Media

Host

Applications

Operating System

File System (e.g. DOS FAT)

Device

FTL

MTD

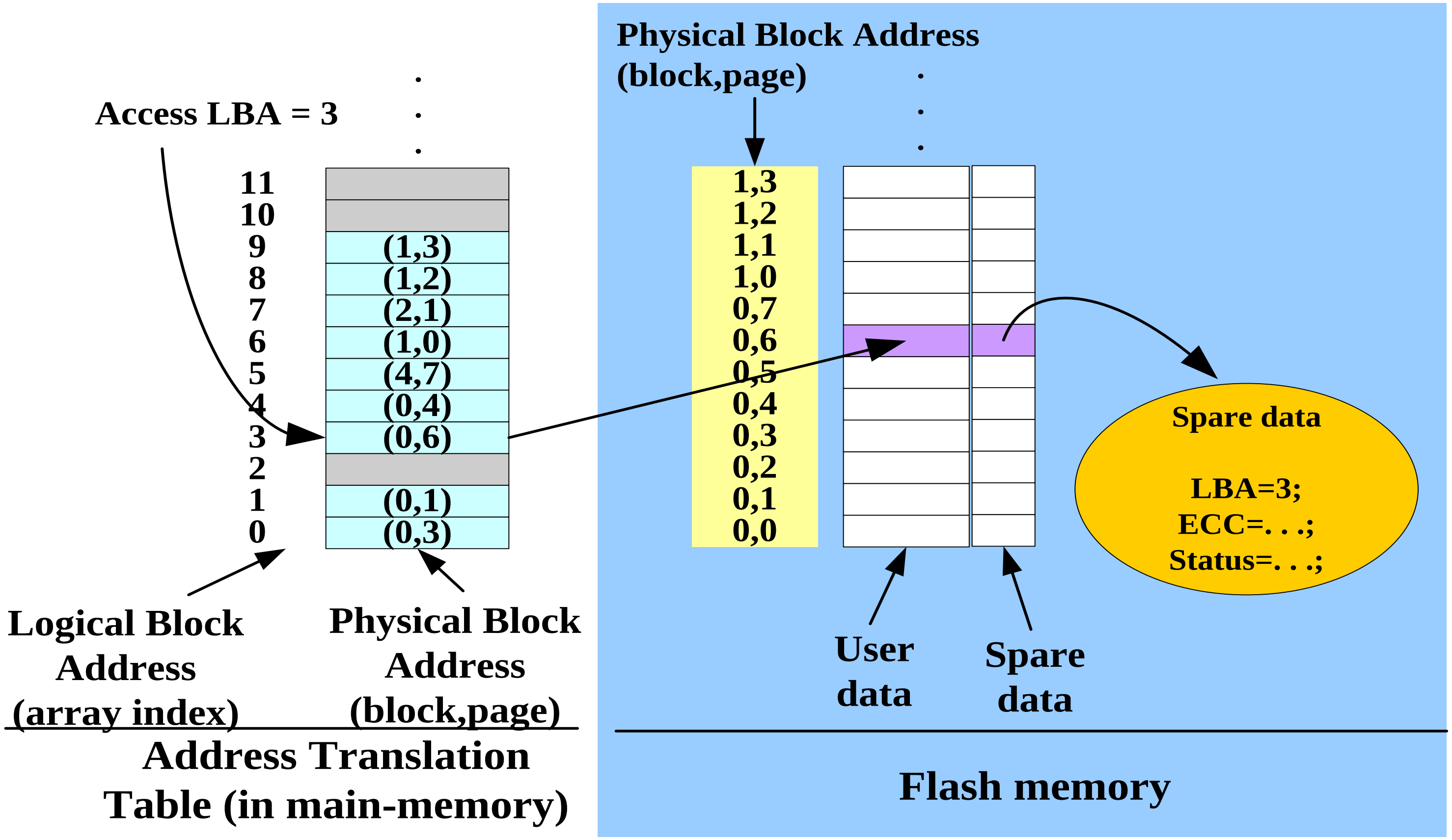
Flash Media

SD,
Memory Stick,
Compact Flash

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

# Policies – FTL

- FTL adopts a page-level address translation mechanism

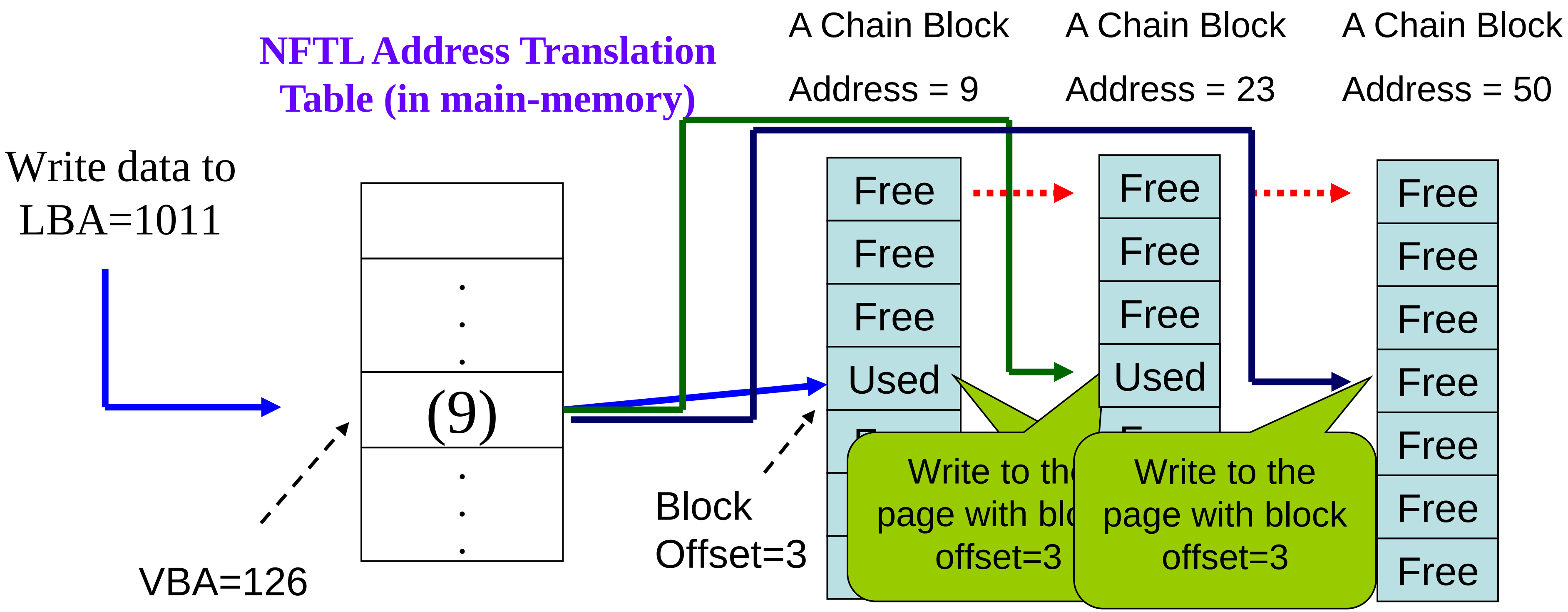


| Logical Block Address (array index) | Physical Block Address (block,page) |
|---|---|
| 11 | |
| 10 | |
| 9 | (1,3) |
| 8 | (1,2) |
| 7 | (2,1) |
| 6 | (1,0) |
| 5 | (4,7) |
| 4 | (0,4) |
| 3 | (0,6) |
| 2 | |
| 1 | (0,1) |
| 0 | (0,3) |

Address Translation Table (in main-memory)

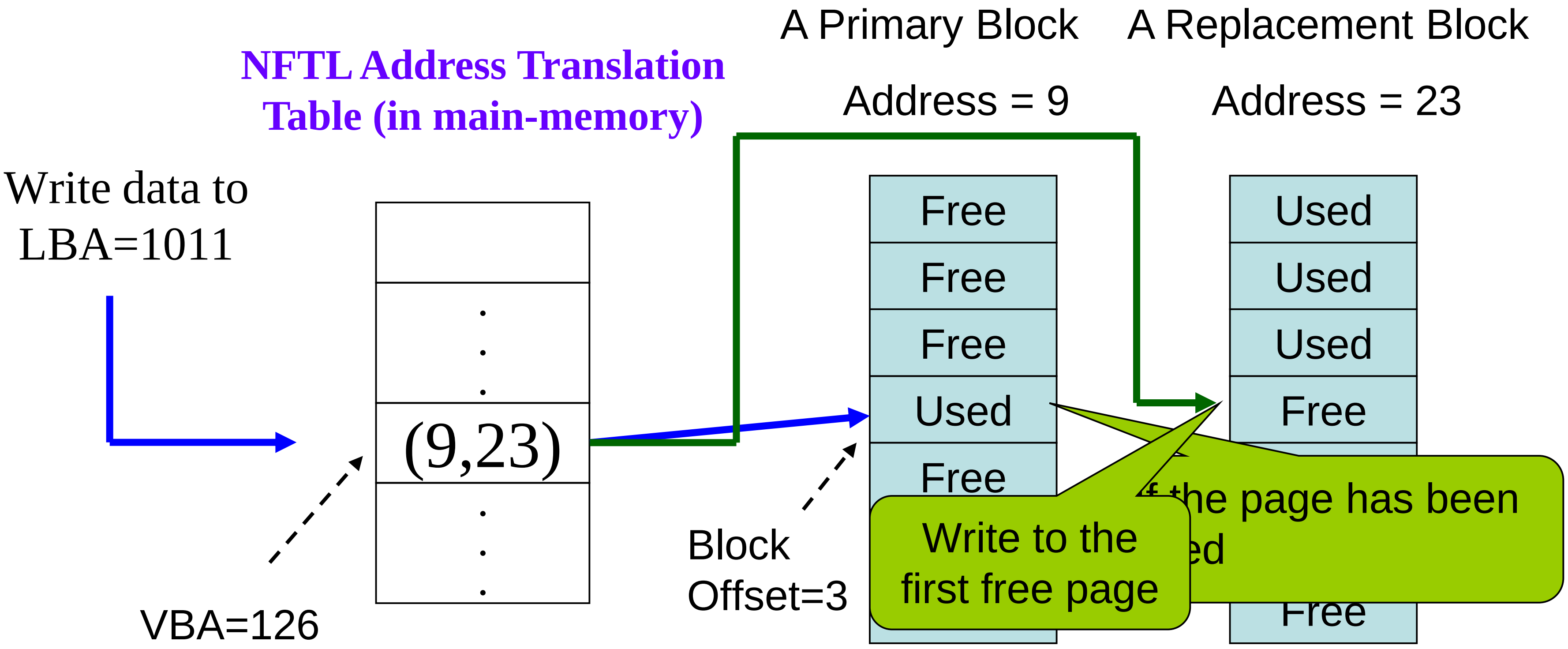
# Policies – NFTL (Type 1)

- 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


NFTL Address Translation Table (in main-memory)
A Chain Block Address = 9
A Chain Block Address = 23
A Chain Block Address = 50
Write data to LBA=1011
(9)
VBA=126
Block Offset=3
Free
Free
Free
Used
Free
Free
Free
Used
Free
Free
Free
Free
Free
Free
Free
Write to the page with block offset=3
Write to the page with block offset=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



Write data to LBA=1011

VBA=126

NFTL Address Translation Table (in main-memory)

(9,23)

Block Offset=3

A Primary Block Address = 9

A Replacement Block Address = 23

f the page has been ed

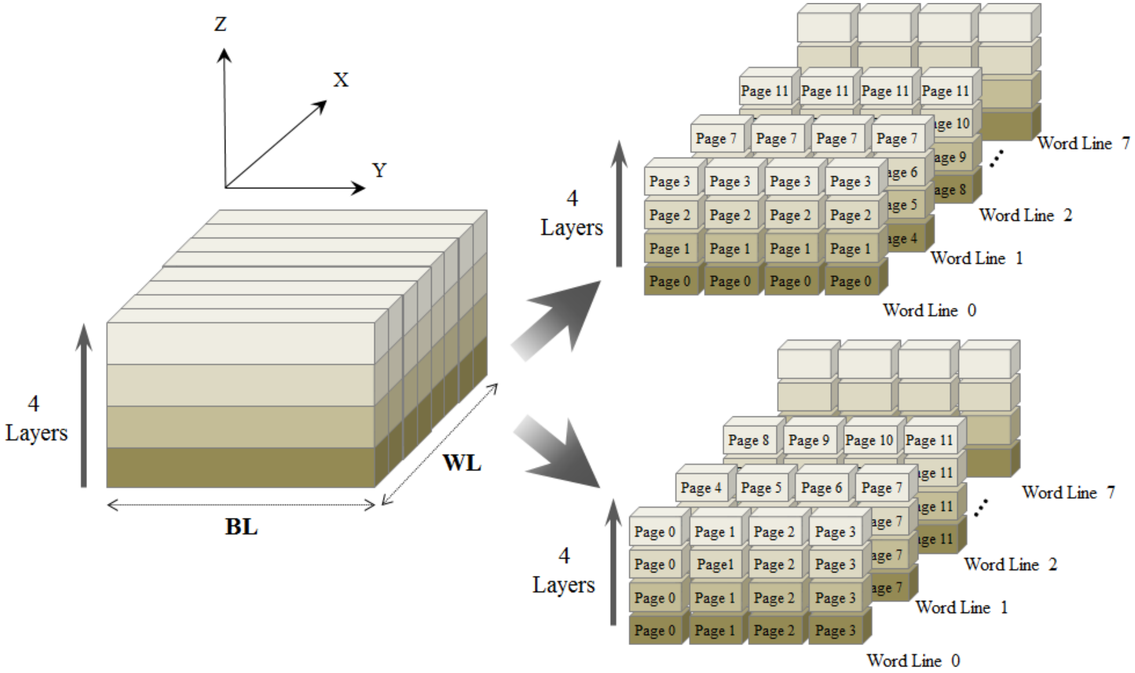
Write to the first free page

# Challenges and Research Topics of Flash Memory Designs

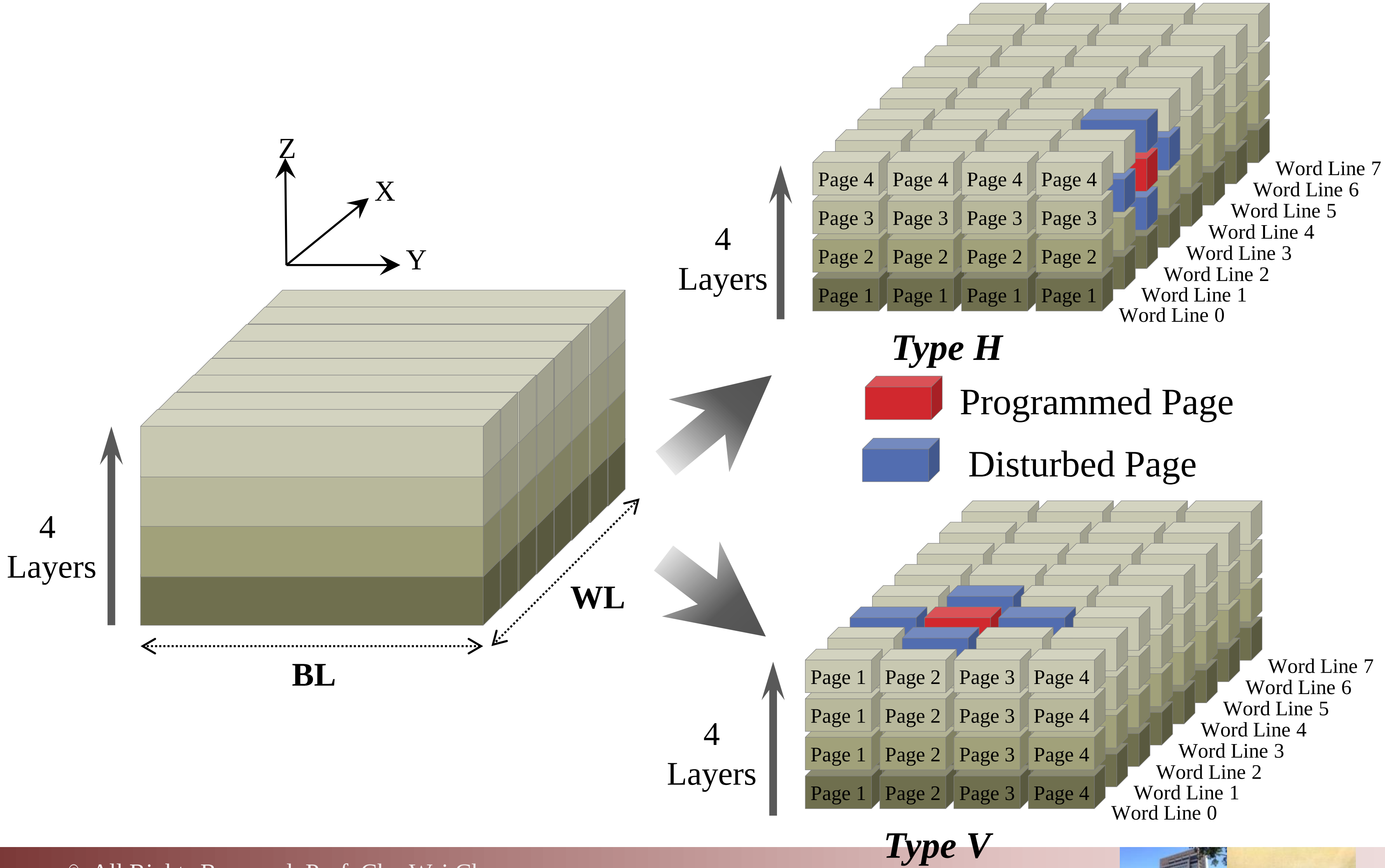
- Performance
  - Reduce the overheads of Flash management
  - Reduce the access time to data
  - Reduce the garbage collection time
- Reliability
  - Error correcting codes
  - Log systems
- Endurance
  - Dynamic wear-leveling
  - Static wear-leveling

# 3D Flash Memory

- 3D flash memory provides a good chance to further scale down the feature size and to reduce the bit cost.
  - Deliver very large storage space
  - Worsen program disturbance



# Deteriorated Disturb on 3D Flash

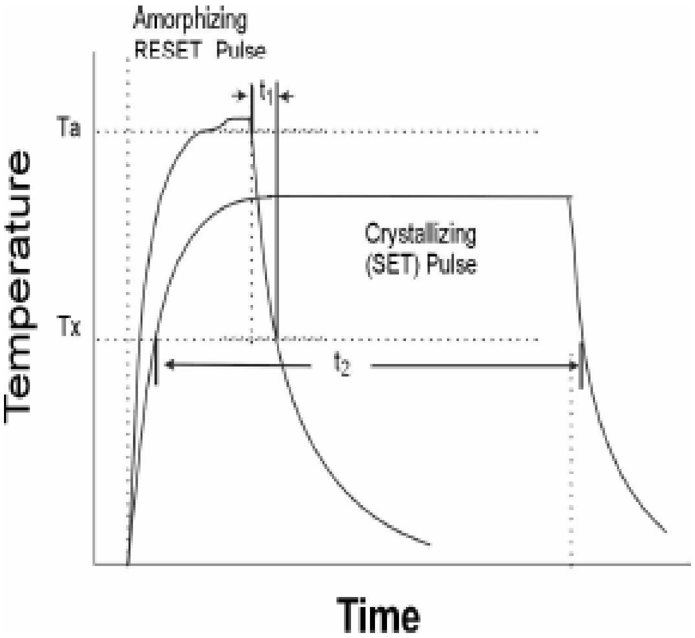


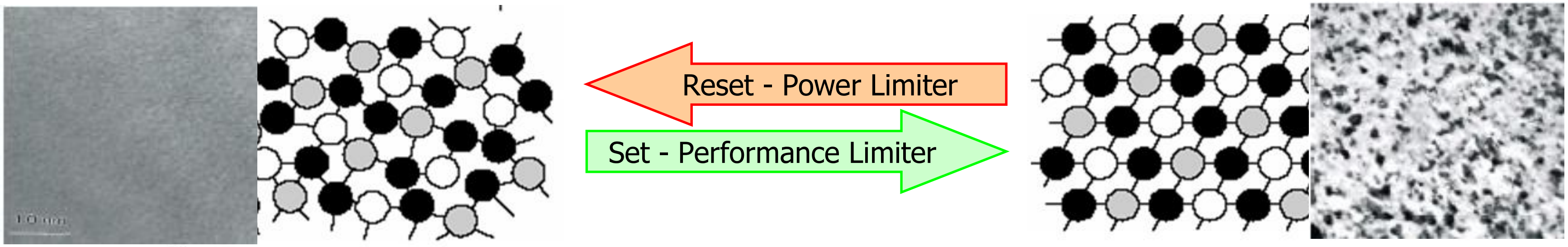
🟥 Programmed Page

🟦 Disturbed Page

# Phase Change Memory (PCM)

- PCM is a non-volatile memory (NVRAM)
- PCM employs a reversible phase change in materials to store information.
- PCM exploits differences in the electrical resistivity of a material in different phases



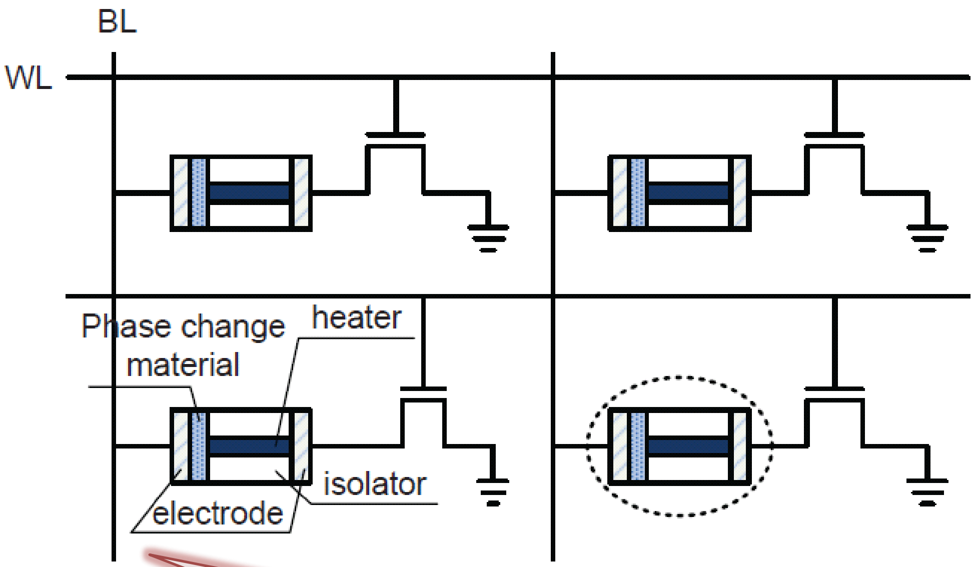


Reset - Power Limiter

Set - Performance Limiter

# PCM Cell Array and Characteristics

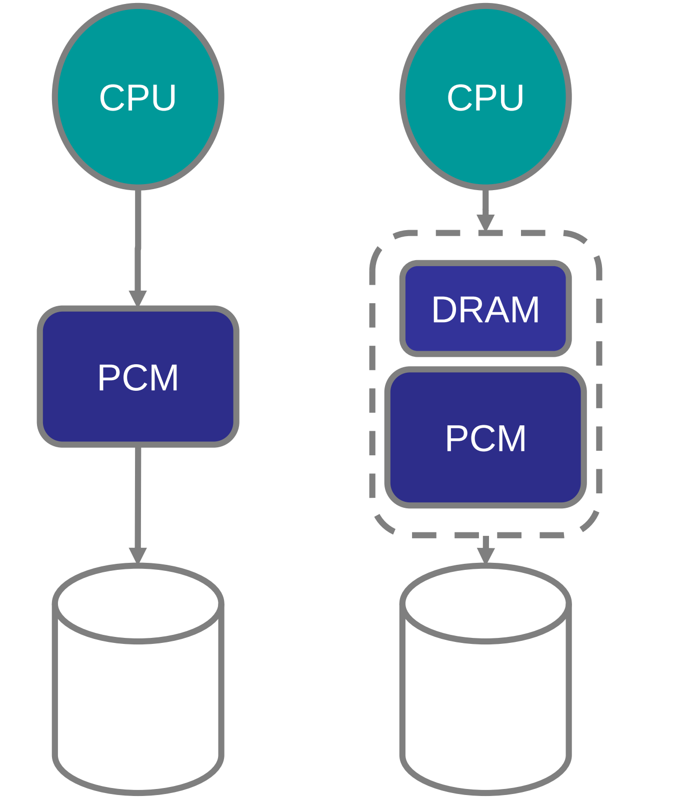
- Pros of PCM
  - Non-volatility
  - Bit-addressability
  - High scalability
  - No dynamic power
- Cons of PCM (compared to DRAM)
  - Low performance on writes
  - High energy consumption on writes
  - Low endurance



The read and write (SET and RESET) operations of a PCM cell require different current and voltage levels on the bitline, and take different amount of time to complete.

# PCM as Main Memory (1/2)

- Take advantage of its scalability and byte-addressability
- Challenges
  - Limited PCM endurance
  - Asymmetric read/write performance

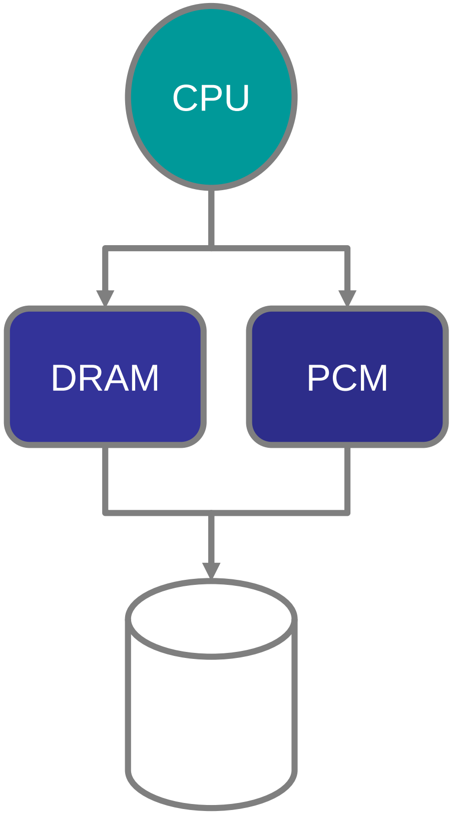


System with PCM

System with hybrid memory : DRAM as cache

# PCM as Main Memory (2/2)

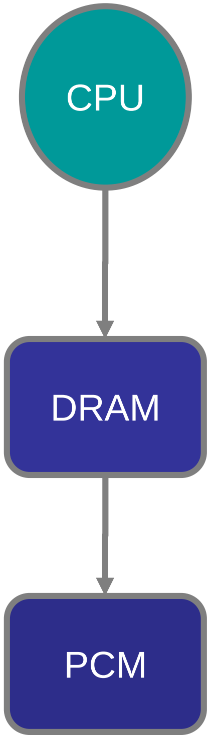
- Take advantage of its non-volatility and byte-addressability
- Challenges:
  - What data should be in DRAM
  - What data should be in PCM
  - How to reuse data after power-off



System with hybrid memory

# PCM as Storage

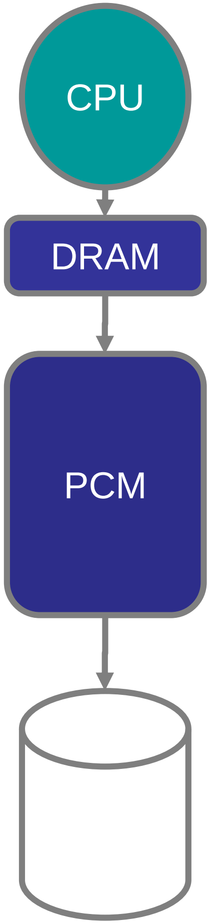
- Take advantage of its non-volatility and high performance
- Challenges
  - Modern file systems have been built around the assumption that persistent storage is accessed via block-based interface
  - How to exploit its properties of persistent, byte-addressable memory



System with PCM

# PCM as Storage Class Memory

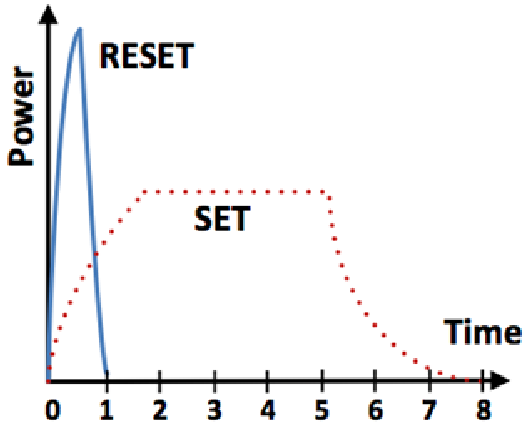
- IBM first proposed the idea of Storage Class Memory (SCM)
- PCM is the candidate of SCM
- SCM blurs the distinction between
  - Memory (fast, expensive, volatile) and
  - Storage (slow, cheap, non-volatile)

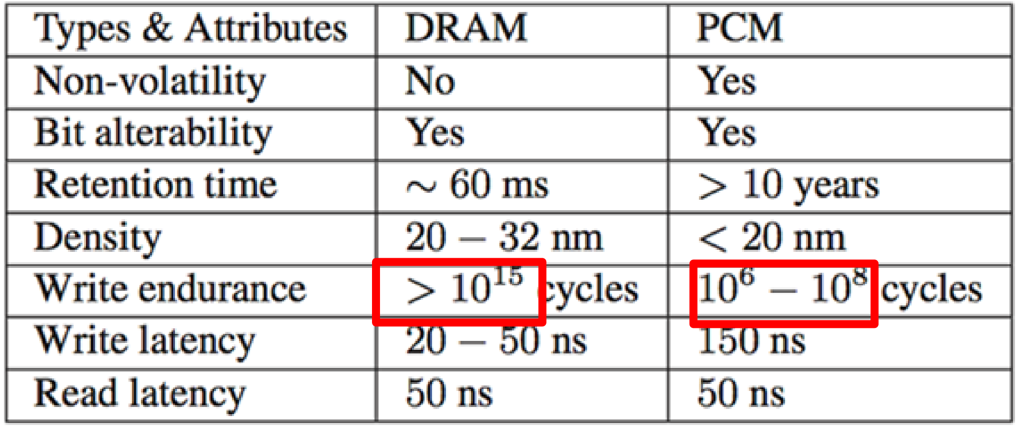


System with PCM as SCM

# Issues of Using PCM

- Write asymmetry
  - Reset
    - High instant  power with short time
  - Set
    - Low power with long time
- Write latency
- Endurance issue



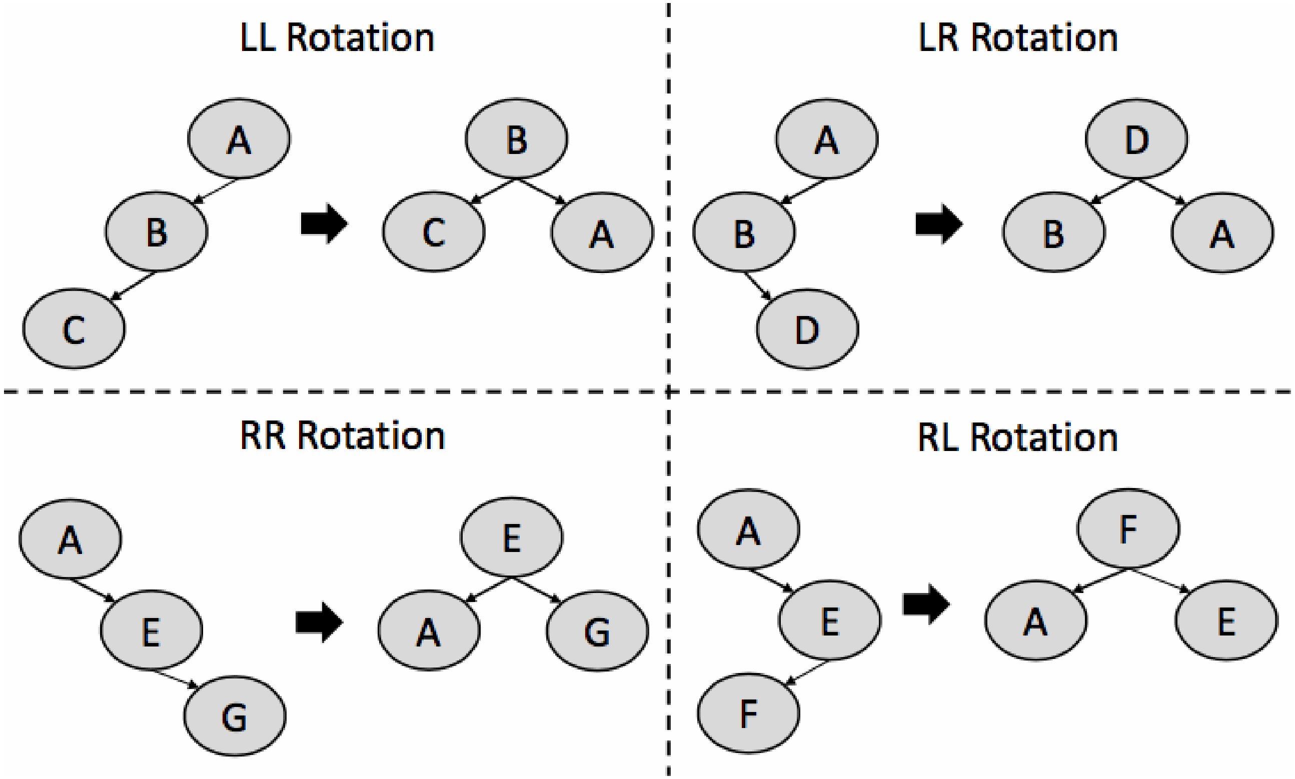


| Types & Attributes | DRAM | PCM |
|---|---|---|
| Non-volatility | No | Yes |
| Bit alterability | Yes | Yes |
| Retention time | $\sim$ 60 ms | $>$ 10 years |
| Density | 20 – 32 nm | $<$ 20 nm |
| Write endurance | $> 10^{15}$ cycles | $10^6 - 10^8$ cycles |
| Write latency | 20 – 50 ns | 150 ns |
| Read latency | 50 ns | 50 ns |

# Write Reduction on PCM

- Big/massive data applications demand extremely large main memory space for better performance
- PCM with low leakage power and high density is a promising candidate to replace DRAM
- Write endurance and latency are critical for using PCM
- Exiting studies improve the write mechanism to handle given write patterns on PCM
- Why don’t we improve fundamental data structures directly so as to generate more suitable write patterns for PCM

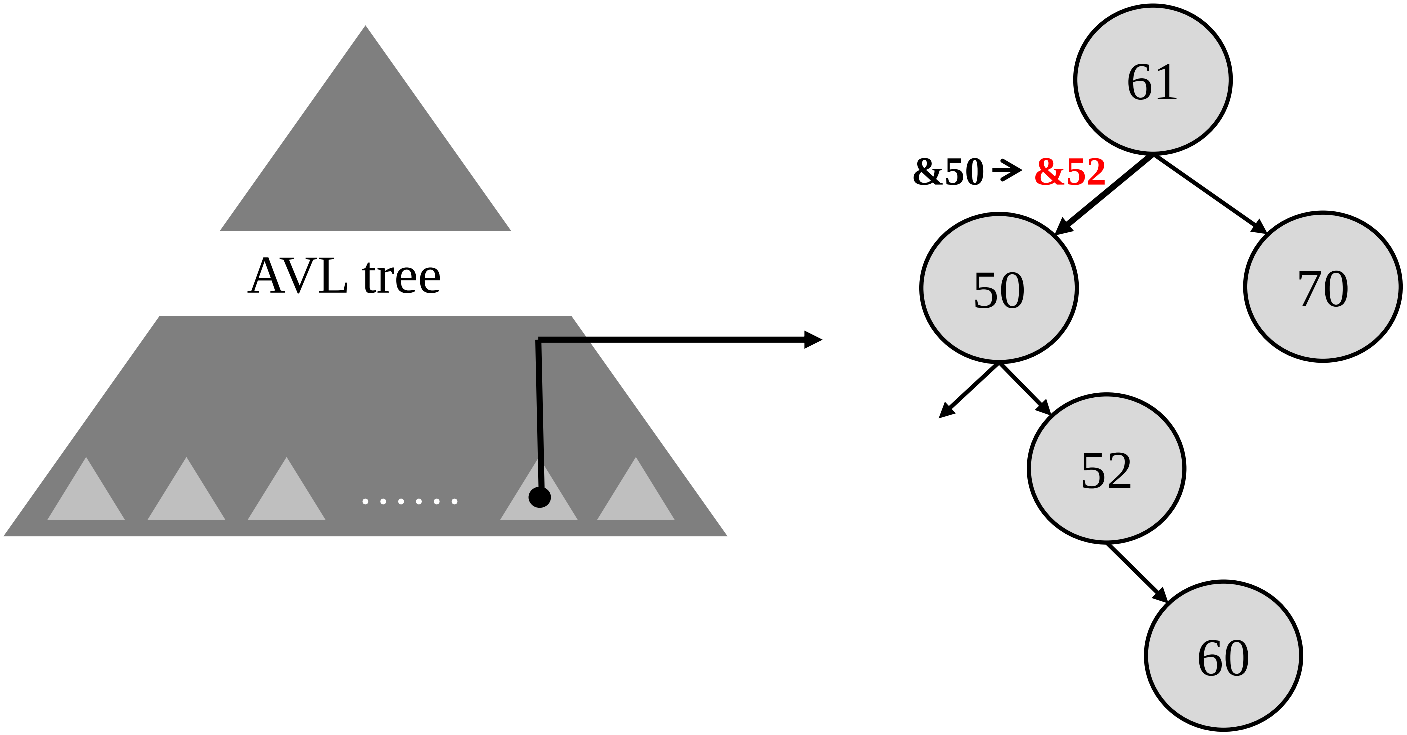
# Four Types of AVL Tree Rotations



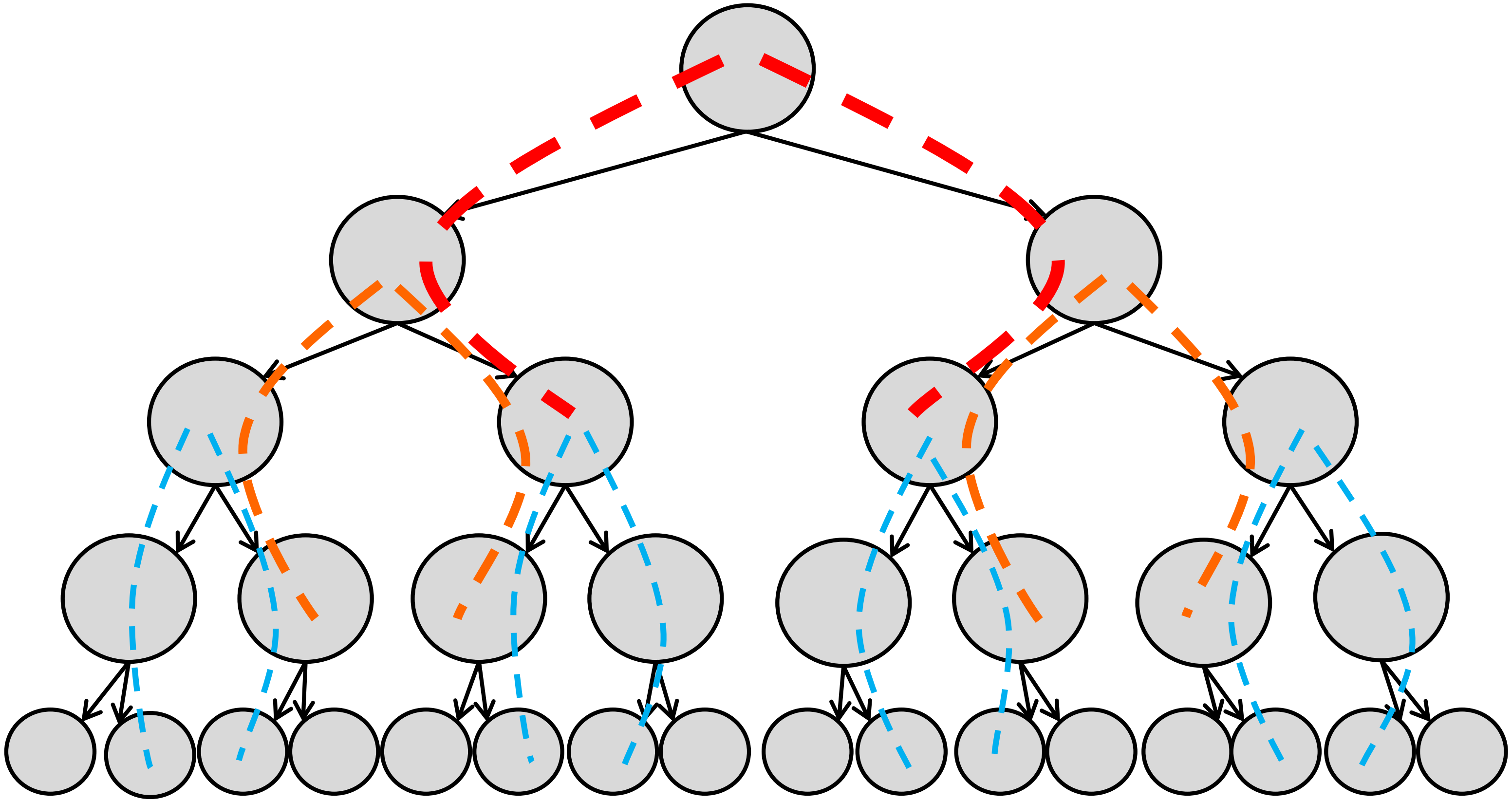
# Relation among Nodes in an RR Rotation

Before RR Rotation

After RR Rotation

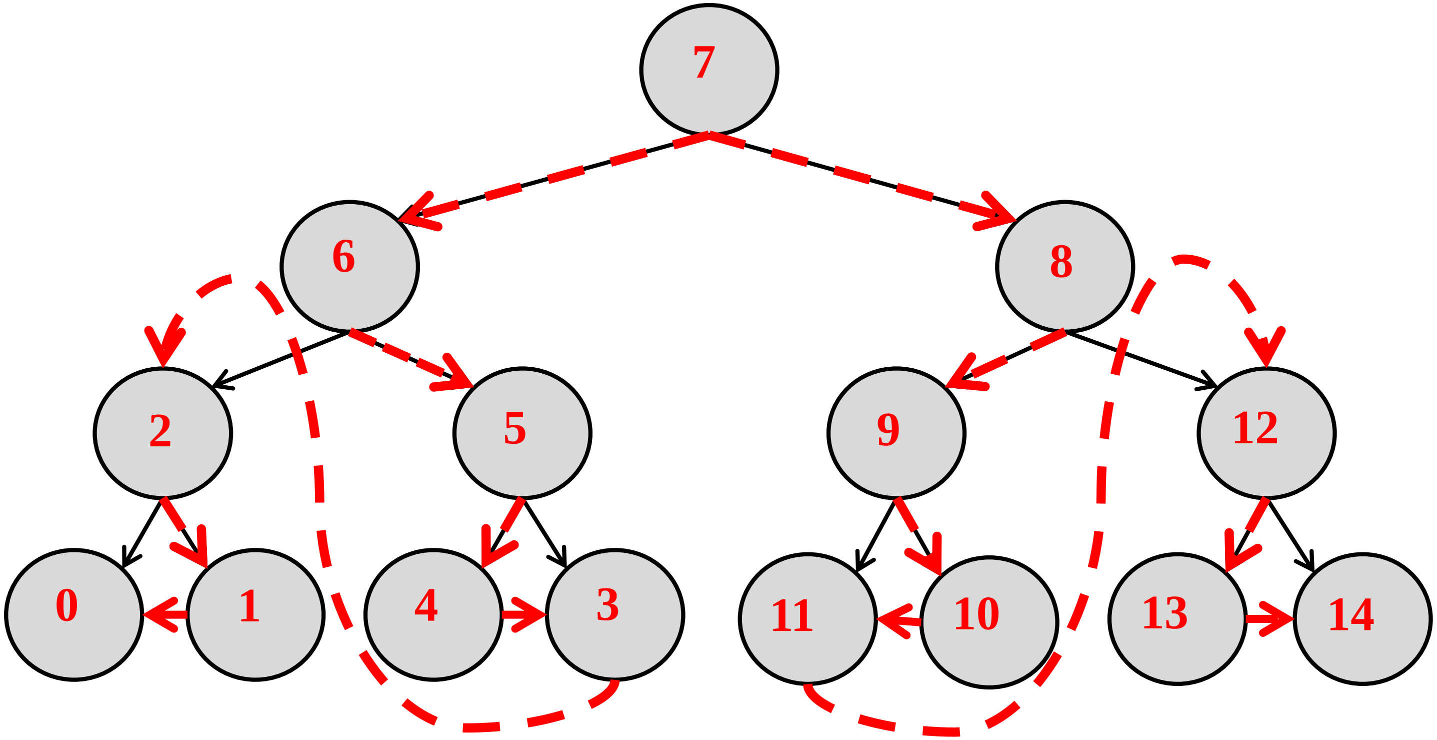


# Relation Binding of Tree Nodes



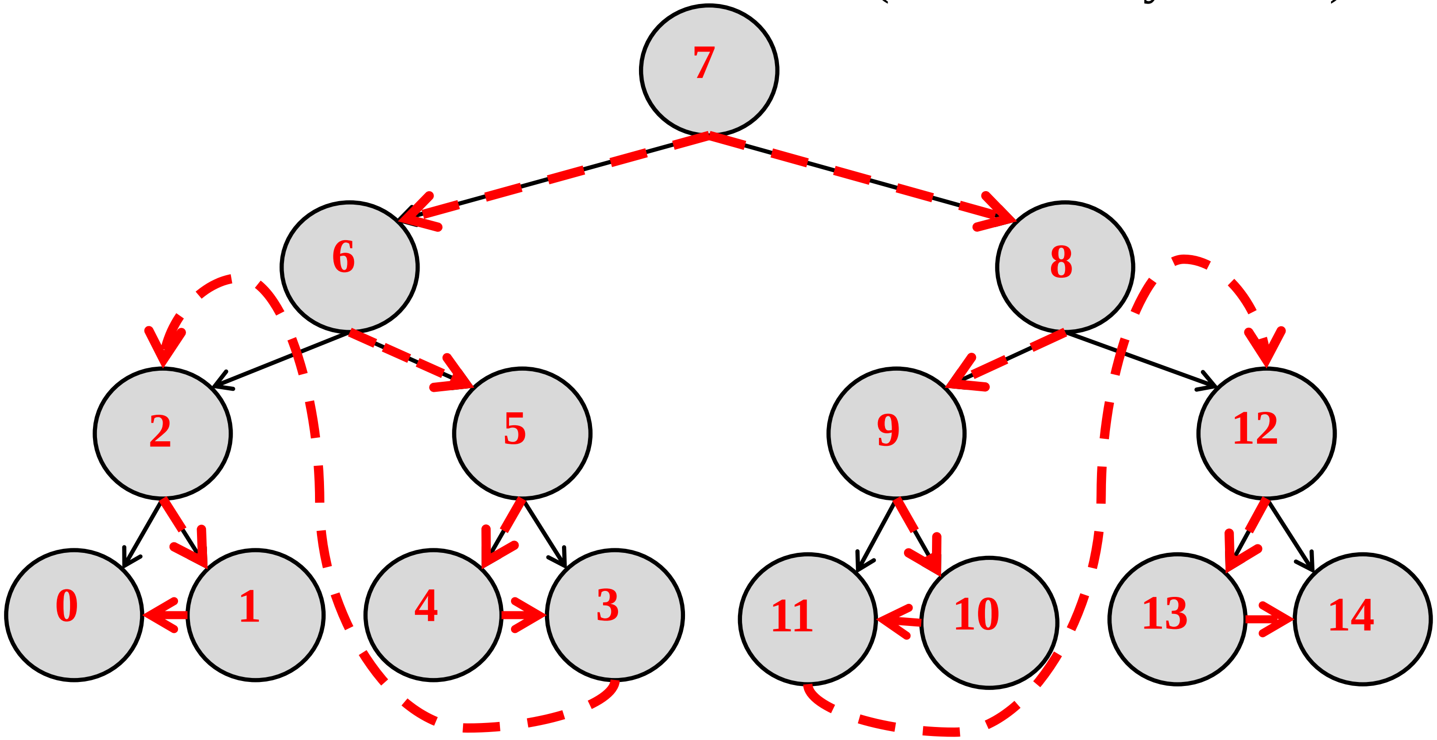
# Depth-First-Alternating Traversal (DFAT)

- A systematic approach for indexing all nodes, where nodes having stronger relations will be assigned closer indexes



# Leveraging Gray Code on DFAT

- Gray code: An ordering of the binary numeral system such that two successive values have the shortest distance (differ in only one bit)

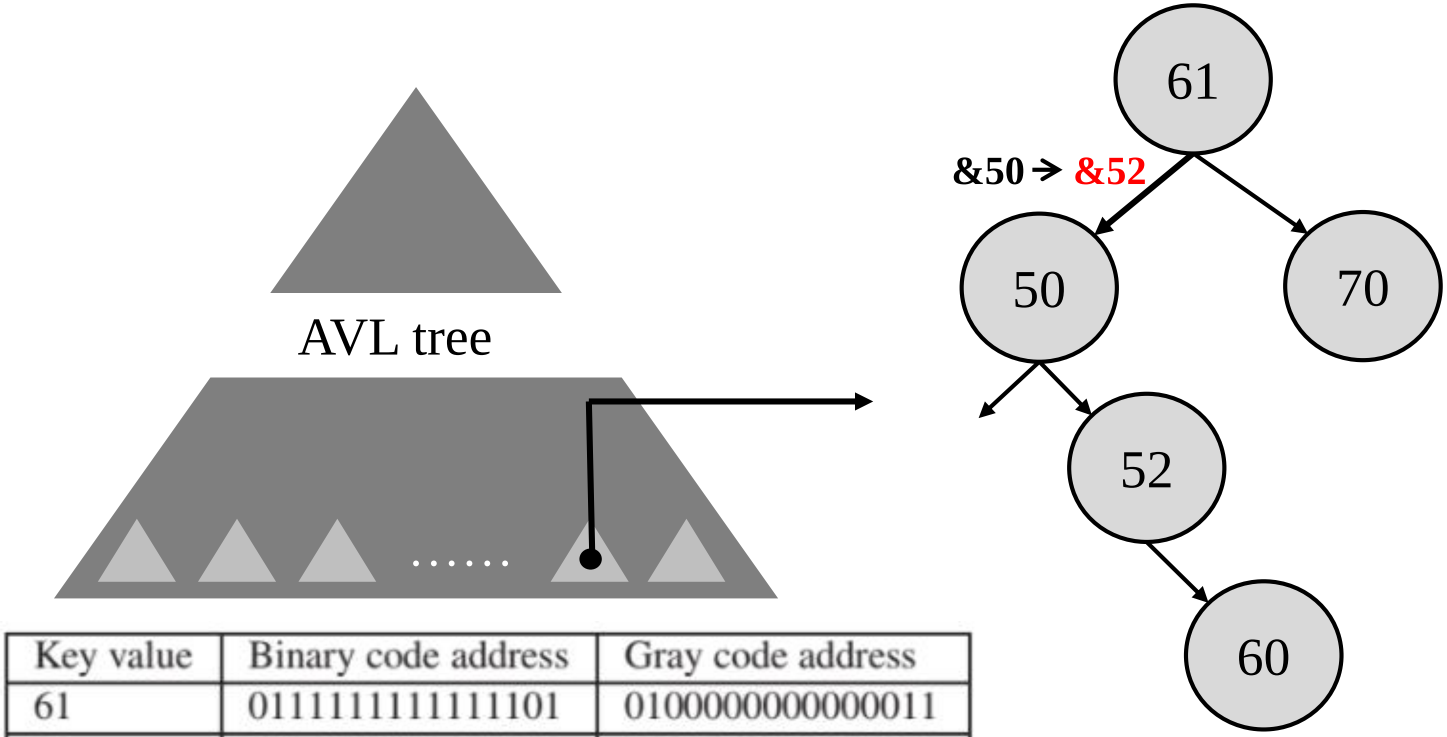


| **Index** | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **Gray Code** | 0000 | 0001 | 0010 | 0011 | 0100 | 0101 | 0110 | 0111 | 1000 | 1001 | 1010 | 1011 | 1100 | 1101 | 1110 |

# An Example of Running DFAT with Gray Code

Before

After RR Rotation



| Key value | Binary code address | Gray code address |
|---|---|---|
| 61 | 0111111111111101 | 0100000000000011 |
| 70 | 0111111111111110 | 0100000000000001 |
| *50* | *0111111111111111* | *0100000000000000* |
| *52* | *1000000000000000* | *1100000000000000* |
| 60 | 1000000000000001 | 1100000000000001 |