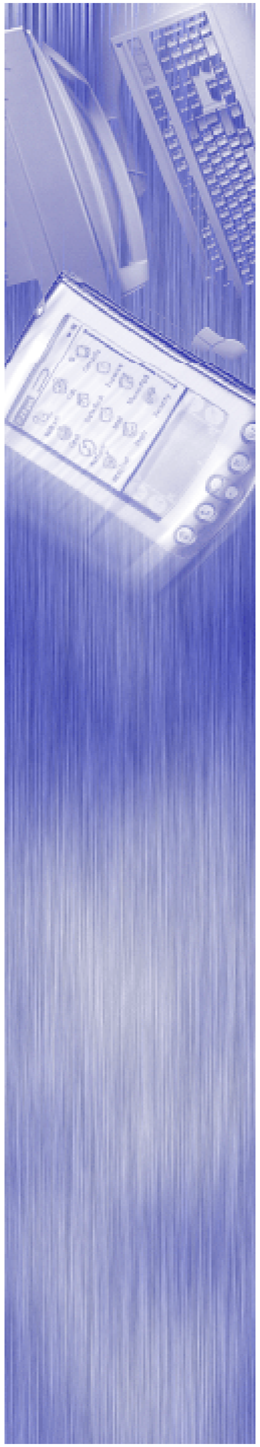


CS632/SEP564: Embedded Operating Systems (Fall 2008)

Flash Memory

KAIST

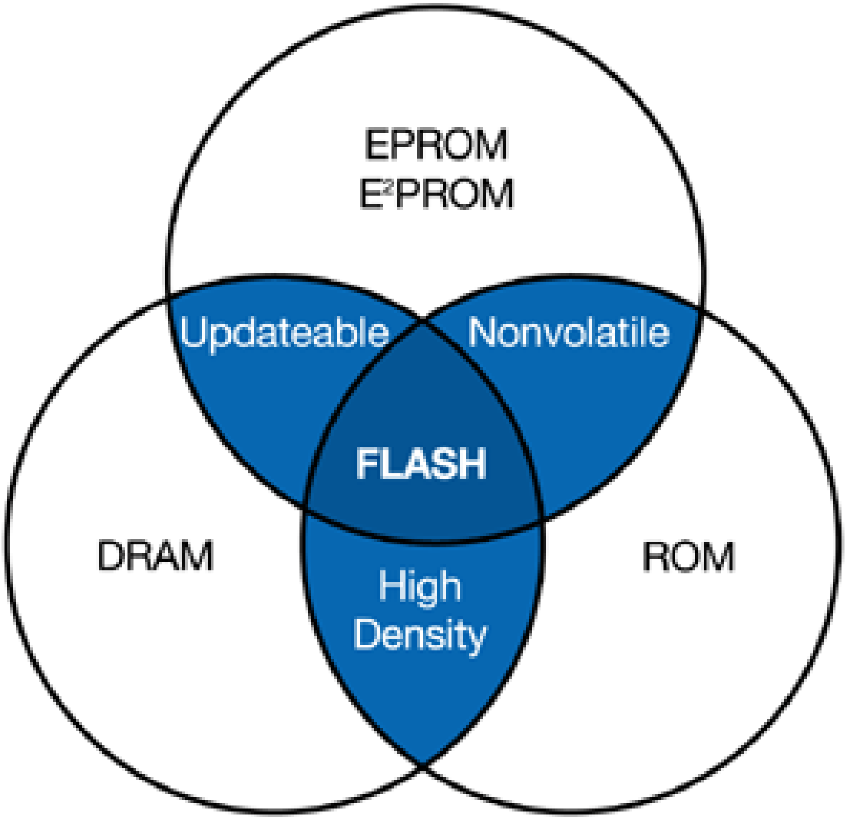
Memory Types

FLASH

- High-density
- Low-cost
- High-speed
- Low-power
- High reliability

DRAM

- High-density
- Low-cost
- High-speed
- High-power



EPROM

- Non-volatile
- High-density
- Ultraviolet light for erasure

EEPROM

- Non-volatile
- Lower reliability
- Higher cost
- Lowest density
- Electrically byte-erasable

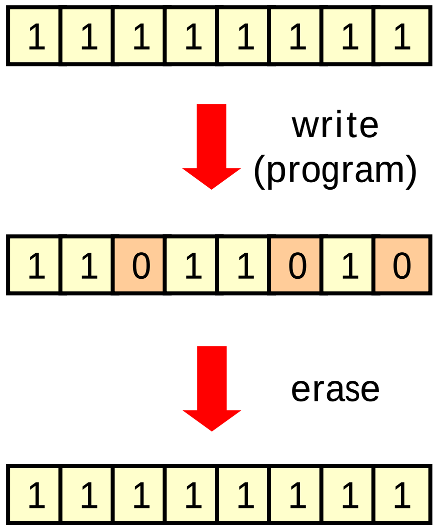
ROM

- High-density
- Reliable
- Low-cost
- Suitable for high production with stable code

Source: Intel Corporation.

Flash Memory Characteristics

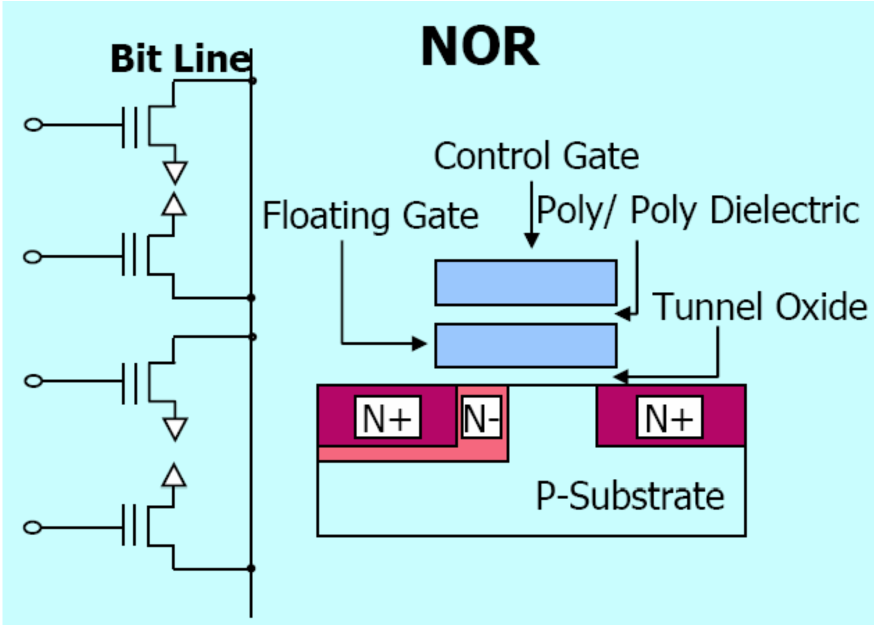
- **Erase-before-write**
  - Read
  - Write or Program – change state from 1 to 0
  - Erase – change state from 0 to 1



- **Bulk Erase**
  - Program unit:
    - NOR: byte or word
    - NAND: sector or page
  - Erase unit: block

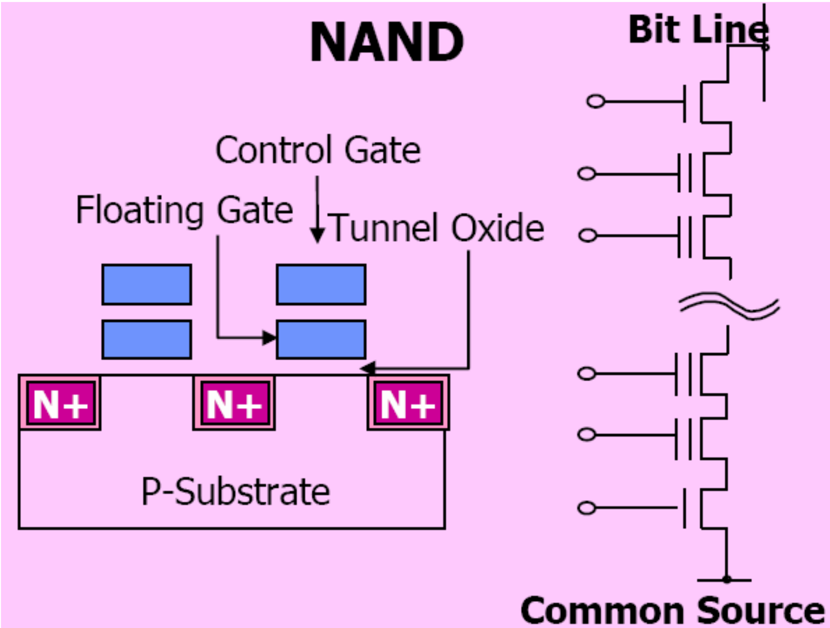
NOR Flash

- **NOR Flash**
  - Random, direct access interface
  - Fast random reads
  - Slow erase and write
  - Mainly for code storage
  - Intel, Spansion, STMicro, ...

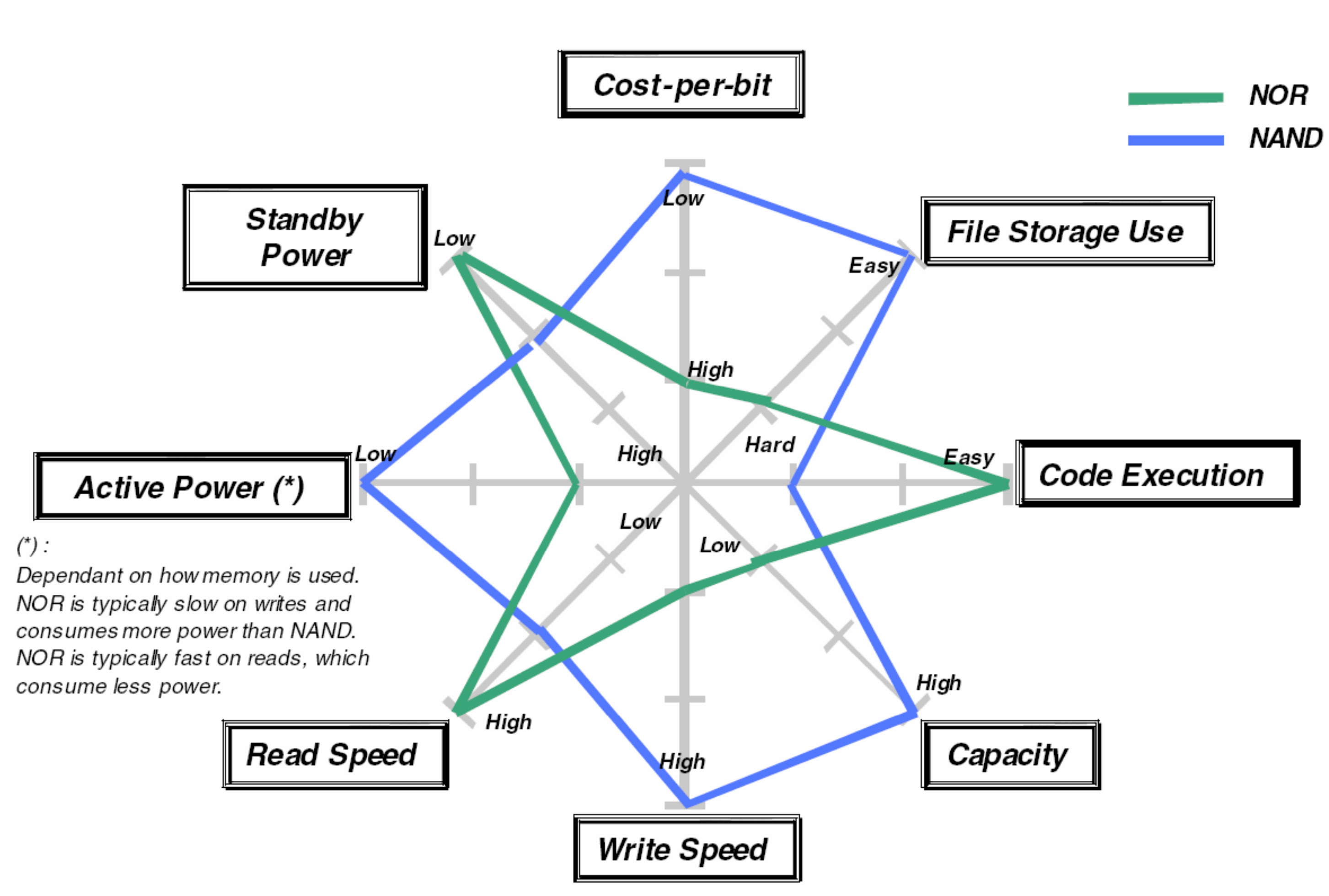


NAND Flash

- **NAND Flash**
  - I/O mapped access
  - Smaller cell size
  - Lower cost
  - Smaller size erase blocks
  - Better performance for erase and write
  - Mainly for data storage
  - Samsung, Toshiba, Hynix, ...



NOR vs. NAND Flash (1)



Source: Toshiba

NOR vs. NAND Flash (2)

Mass Storage-NAND

Memory Cards
(mobile computers)

Solid-State Disk
(rugged & reliable storage)

Digital Camera
(still & moving pictures)

Voice/Audio Recorder
(near CD quality)

- Low Cost and High Density
- Good P/E Cycling Endurance

Code Memory-NOR

BIOS/Networking
(PC/router/hub)

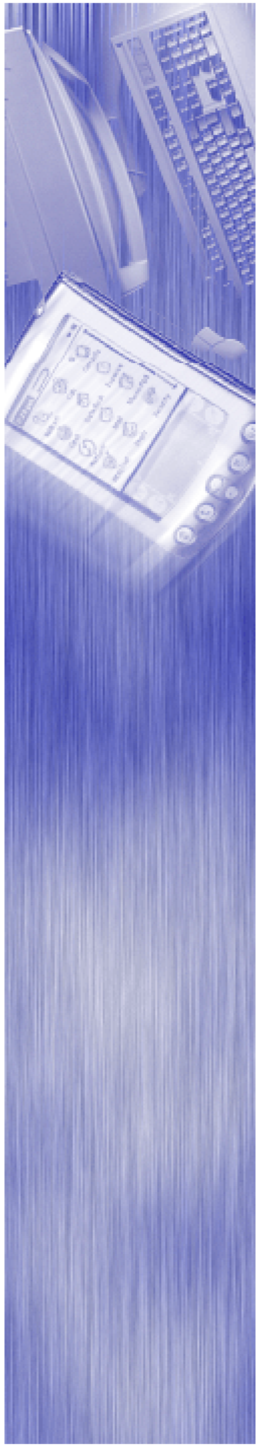
Telecommunications
(switcher)

Cellular Phone
(code & data)

POS / PDA / PCA
(code & data)

- Fast Random Access
- XIP

Source: Samsung Electronics



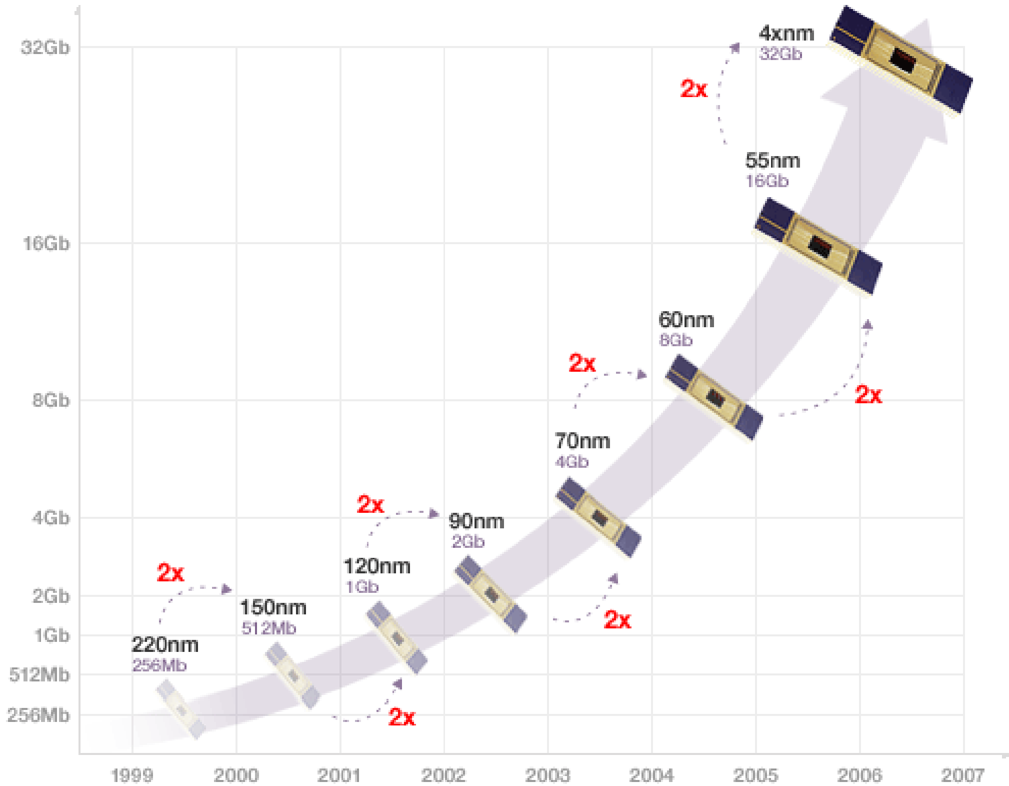
CS632/SEP564: Embedded Operating Systems (Fall 2008)

NAND Flash Memory

KAIST

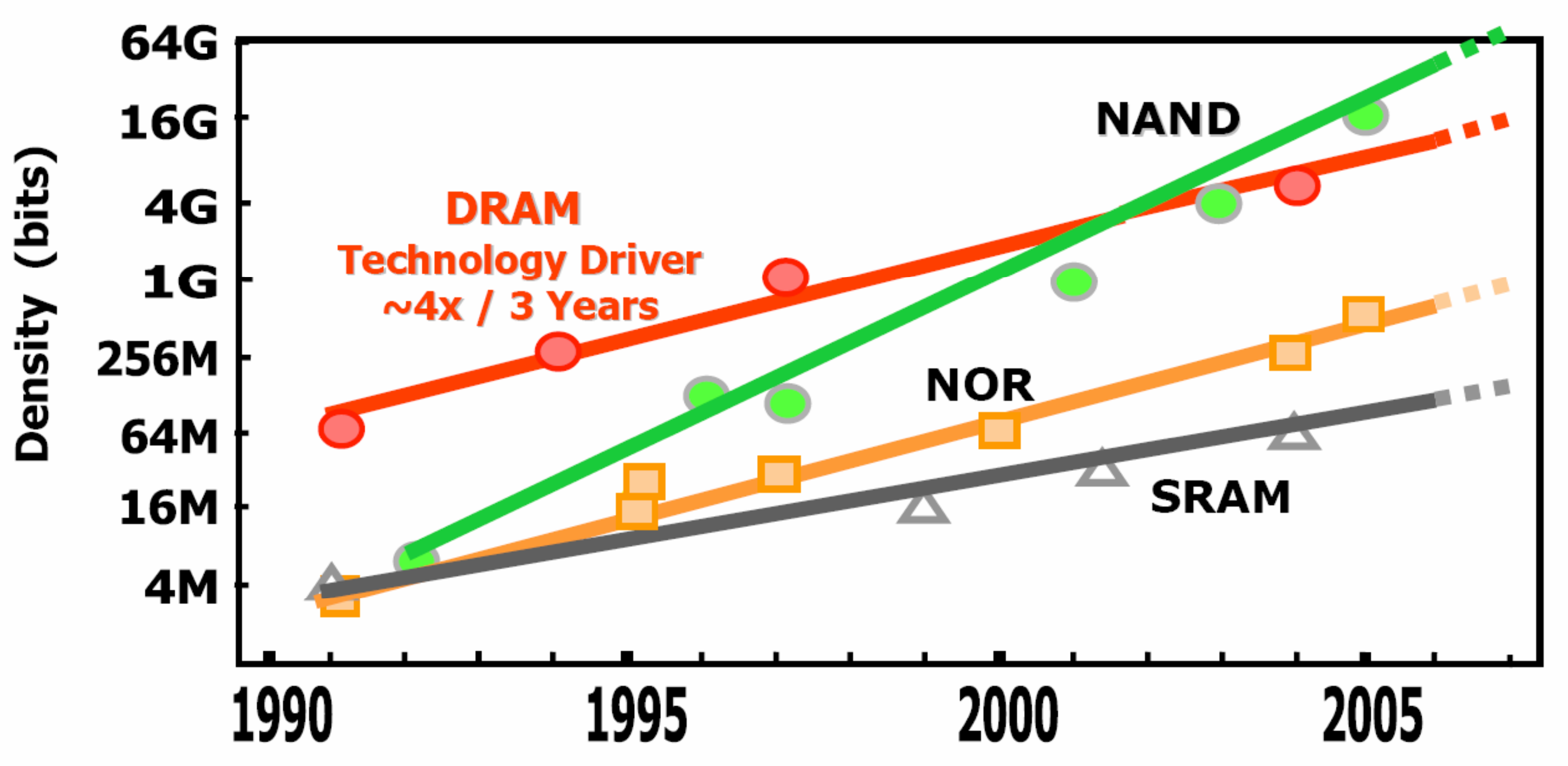
NAND Technology (1)

- **Hwang's Law**
  - The density of the top-of-the-line flash memory chips will double every 12 months



NAND Technology (2)

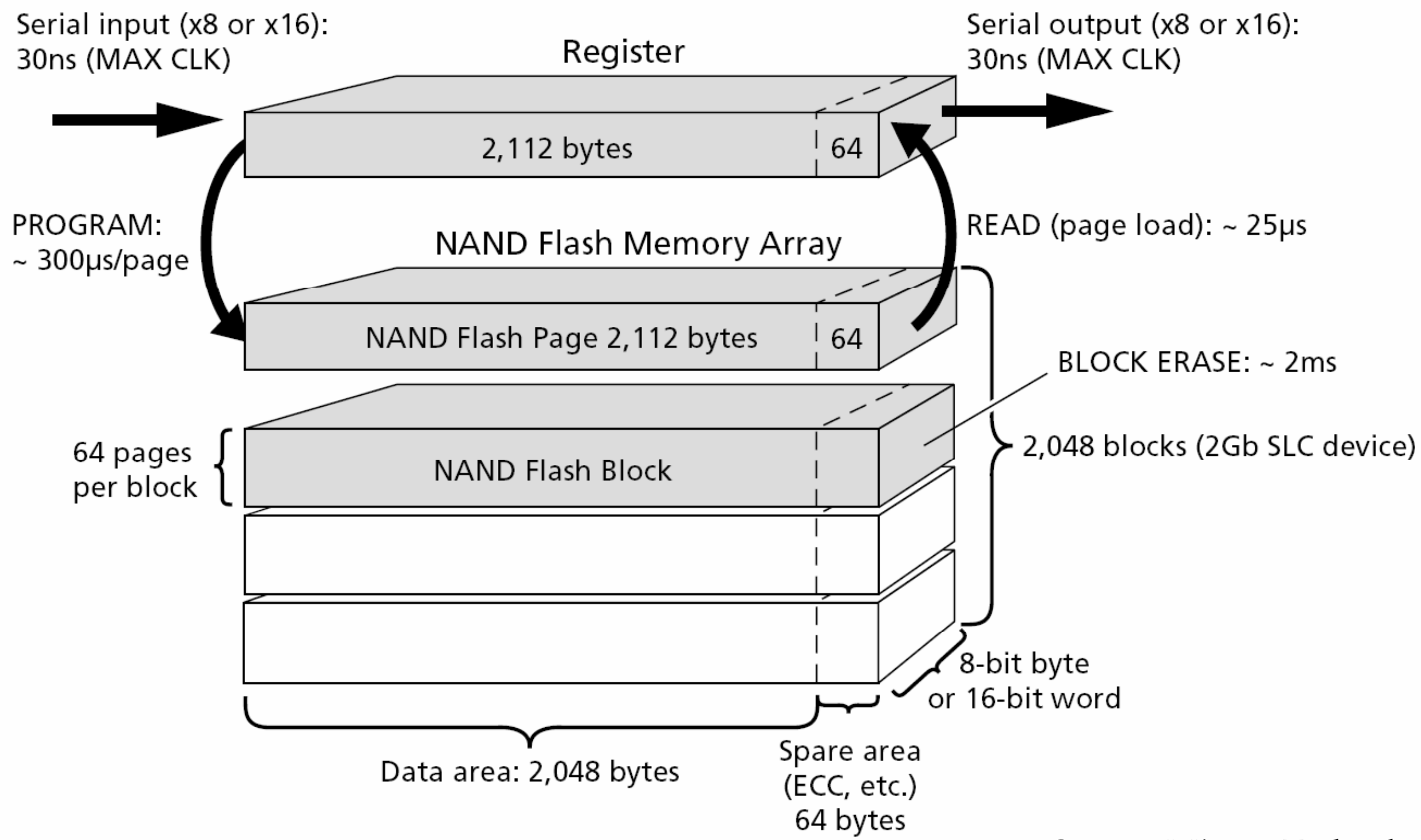
- **Density Growth**



Source: Samsung Electronics

NAND Flash Architecture (1)

- **2Gb NAND Flash Device Organization**

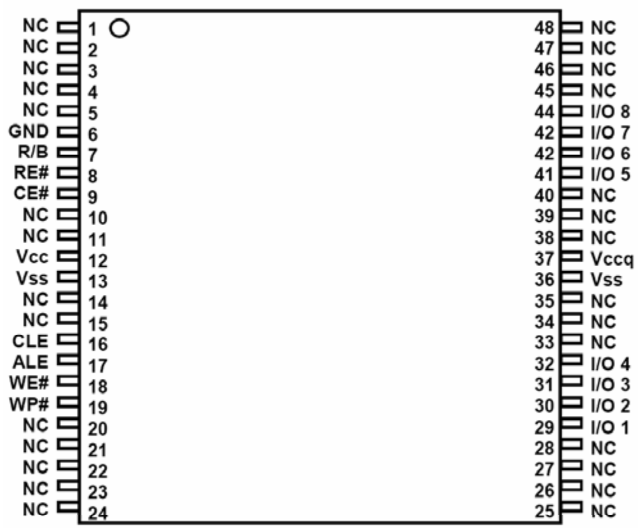


Source: Micron Technology, Inc.

NAND Flash Architecture (2)

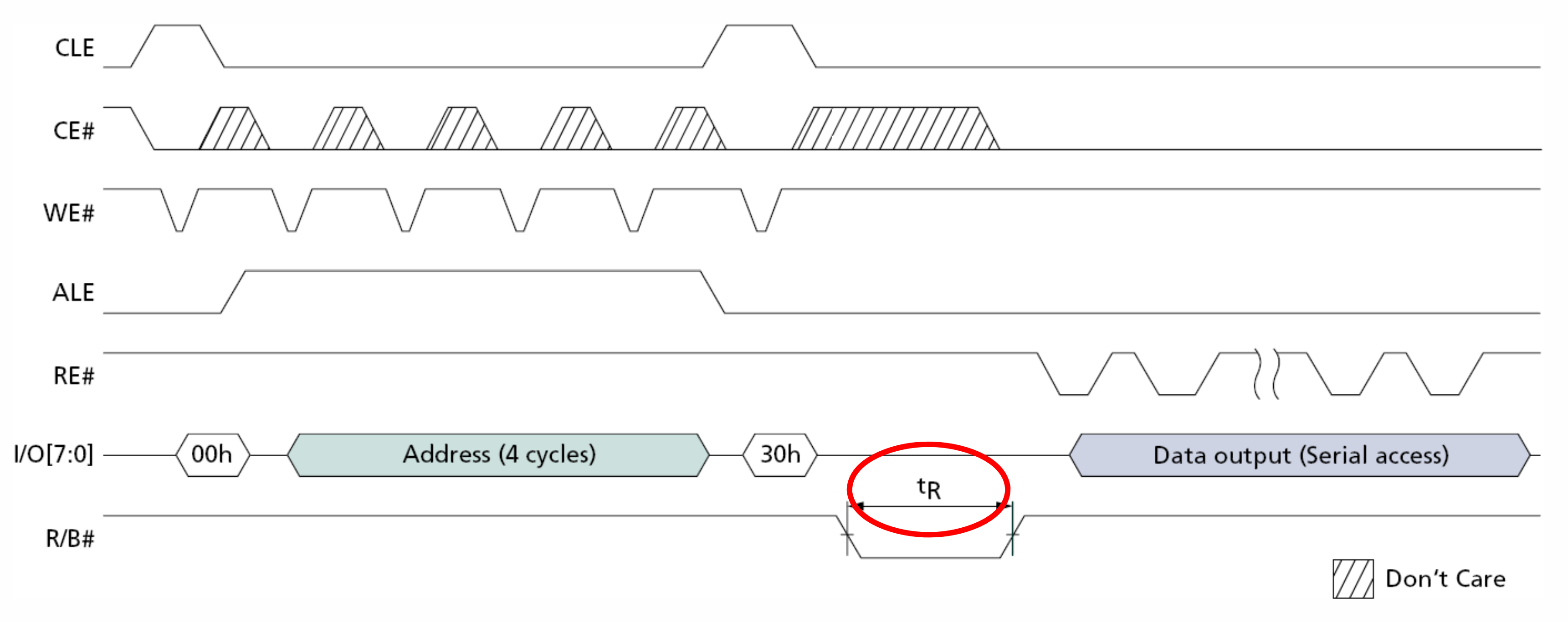
- ## NAND Flash Interface

Pin Name	Pin Function
$I/O_0 \sim I/O_7$	**DATA INPUS/OUTPUTS** Command, address, or data
CLE	**COMMAND LATCH ENABLE**
ALE	**ADDRESS LATCH ENABLE**
$\overline{CE}$	**CHIP ENABLE**
$\overline{RE}$	**READ ENABLE**
$\overline{WE}$	**WRITE ENABLE**
$\overline{WP}$	**WRITE PROTECT**
$R/\overline{B}$	**READY/BUSY OUTPUT**
VCC	**POWER**
VSS	**GROUND**



NAND Operations (1)

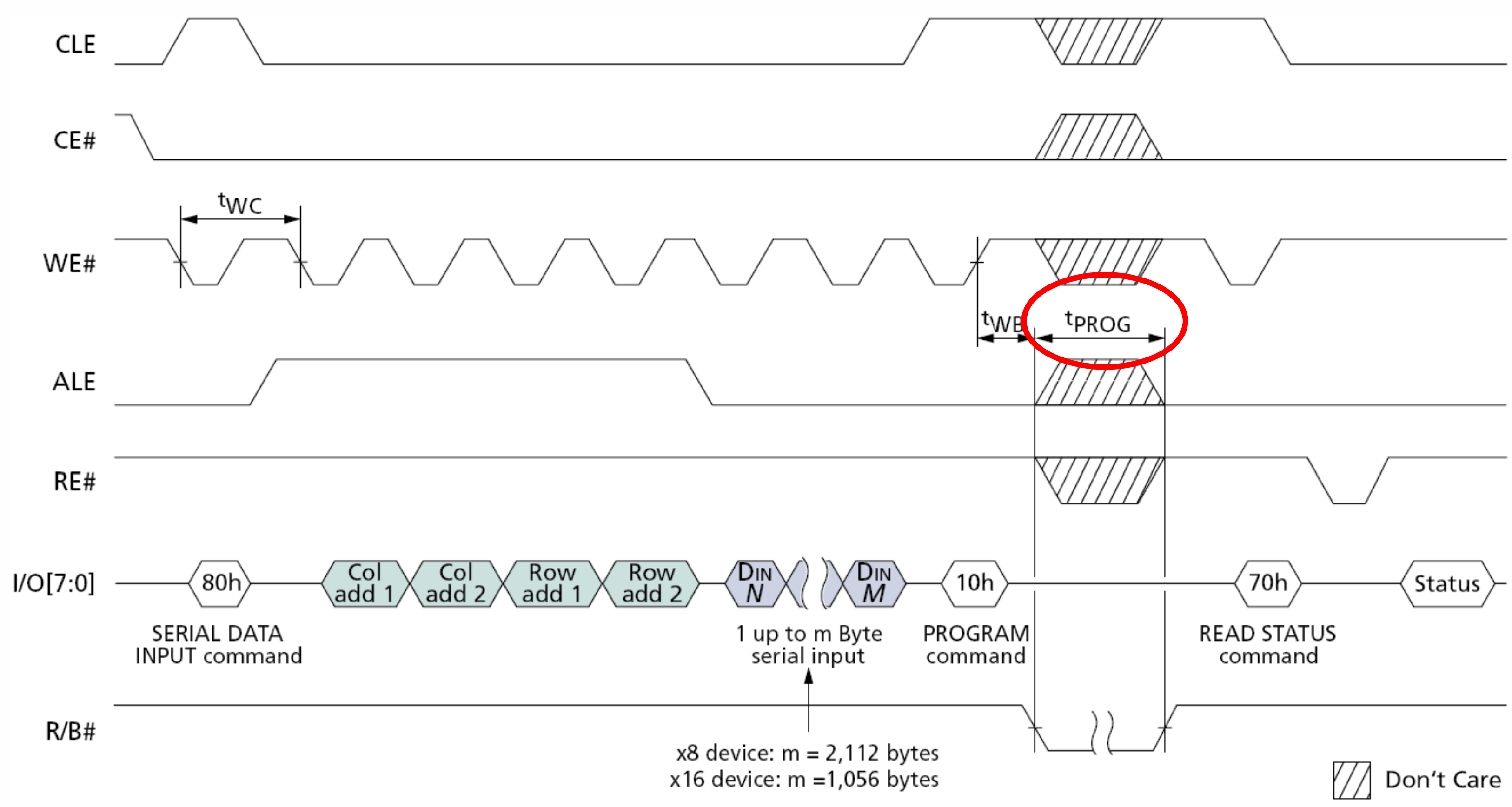
- **READ**



Source: Micron Technology, Inc.

NAND Operations (2)

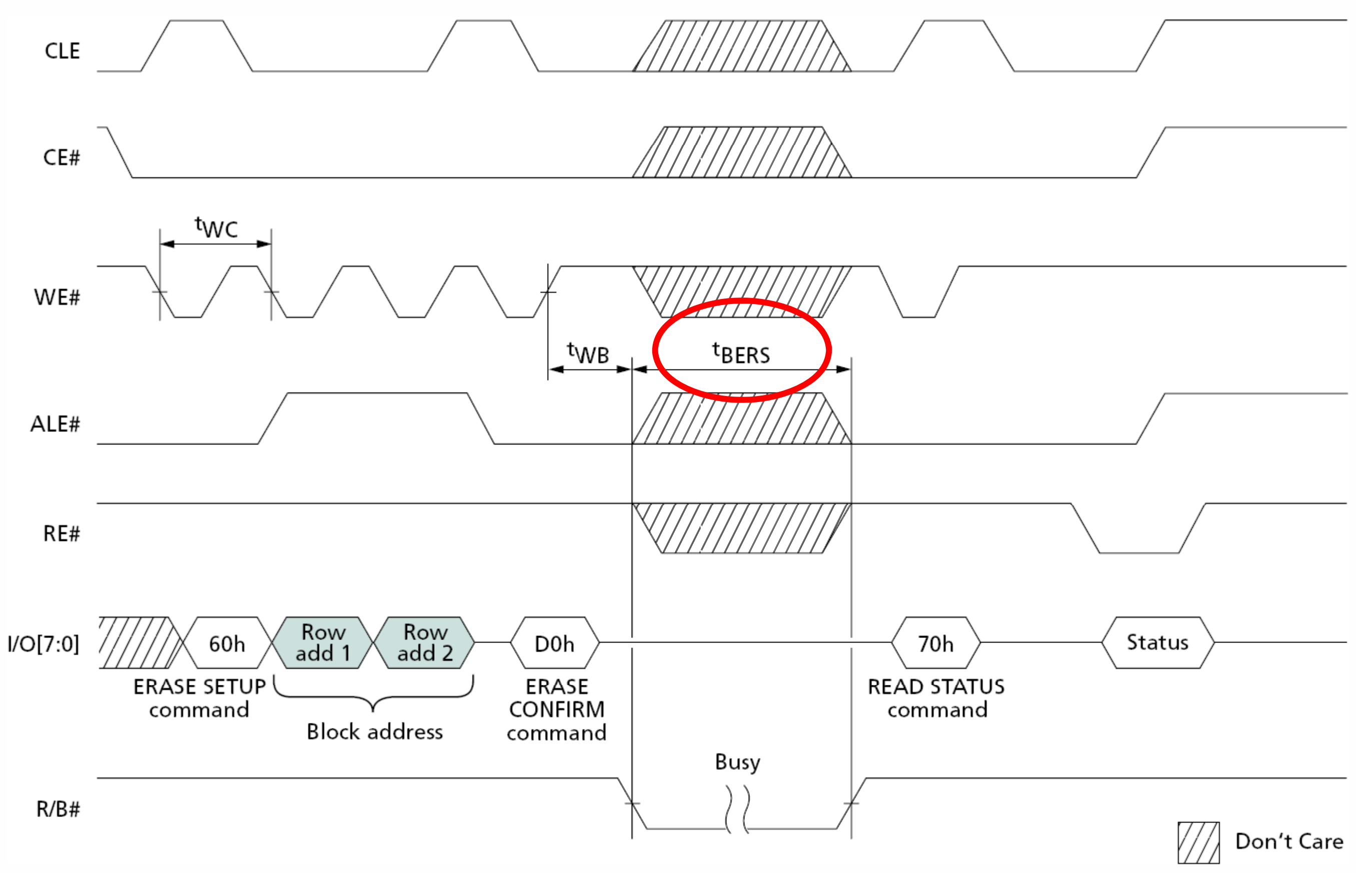
- **PROGRAM**



Source: Micron Technology, Inc.

NAND Operations (3)

- **BLOCK ERASE**

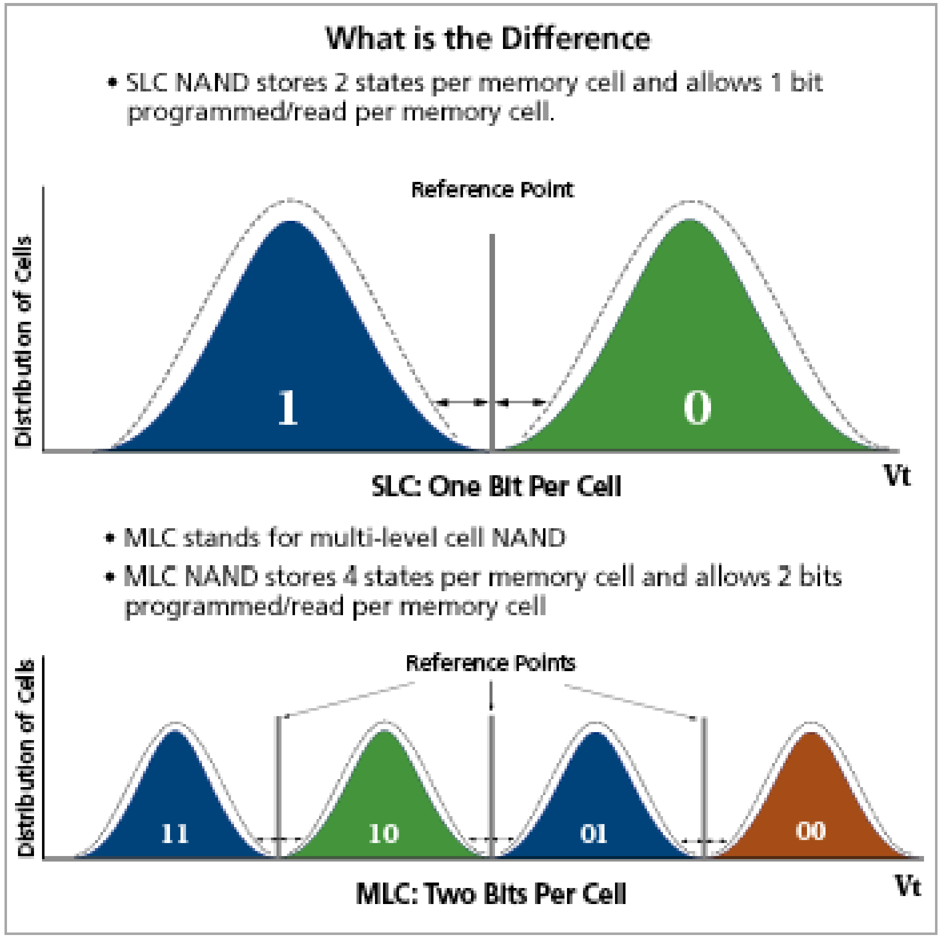


Source: Micron Technology, Inc.

NAND Flash Types (1)

- **SLC NAND Flash**
  - Small block (≤ 1Gb)
  - Large block (≥ 1Gb)

- **MLC NAND Flash**



Source: Micron Technology, Inc.

NAND Flash Types (2)

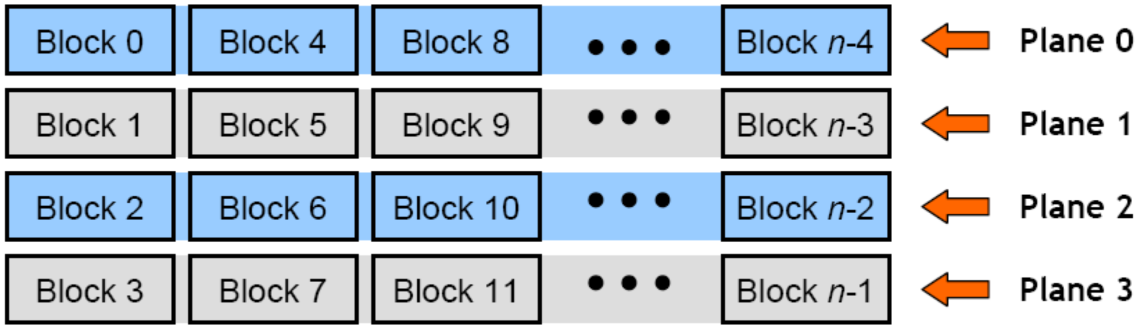
	SLC NAND[1] (small block)	SLC NAND[2] (large block)	MLC NAND[3]
Page size (Bytes)	512+16	2,048+64	4,096+128
Pages / Block	32	64	128
Block size	16KB	128KB	512KB
t_R	15 μs (max)	20 μs (max)	50 μs (max)
t_{PROG}	200 μs (typ) 500 μs (max)	200 μs (typ) 700 μs (max)	600 μs (typ) 1,200 μs (max)
t_{BERS}	2 ms (typ) 3 ms (max)	1.5 ms (typ) 2 ms (max)	3 ms (typ)
NOP	1 (main), 2 (spare)	4	1
Endurance Cycles	100K	100K	10K
ECC (per 512Bytes)	1 bit ECC 2 bits EDC	1 bit ECC 2 bits EDC	4 bits ECC 5 bits EDC

[1] *Samsung K9F1208X0C (512Mb)* [2] *Samsung K9K8G08U0A (8Gb)* [3] *Micron Technology Inc.*

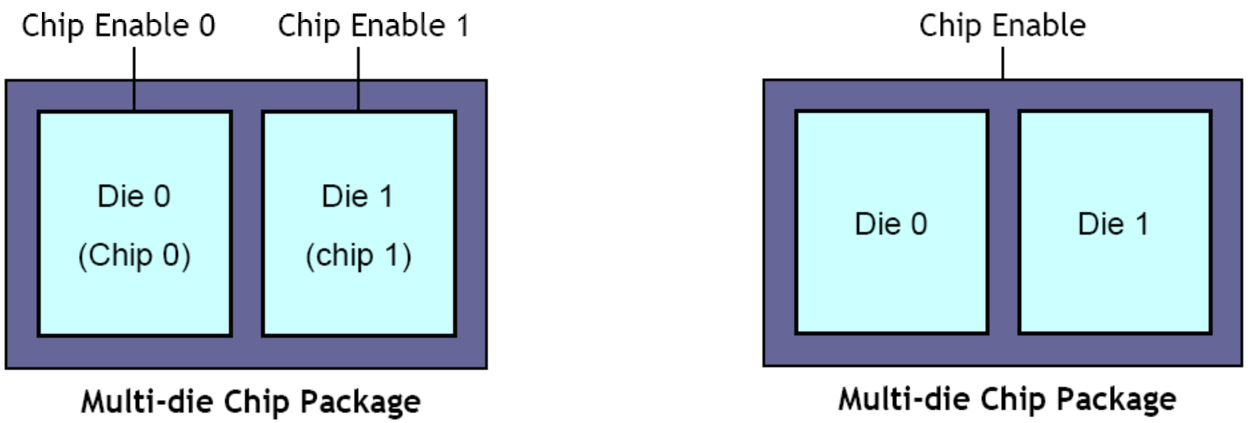
NAND Flash Types (3)

- **Extended NAND Flash Architecture**

Multi-plane



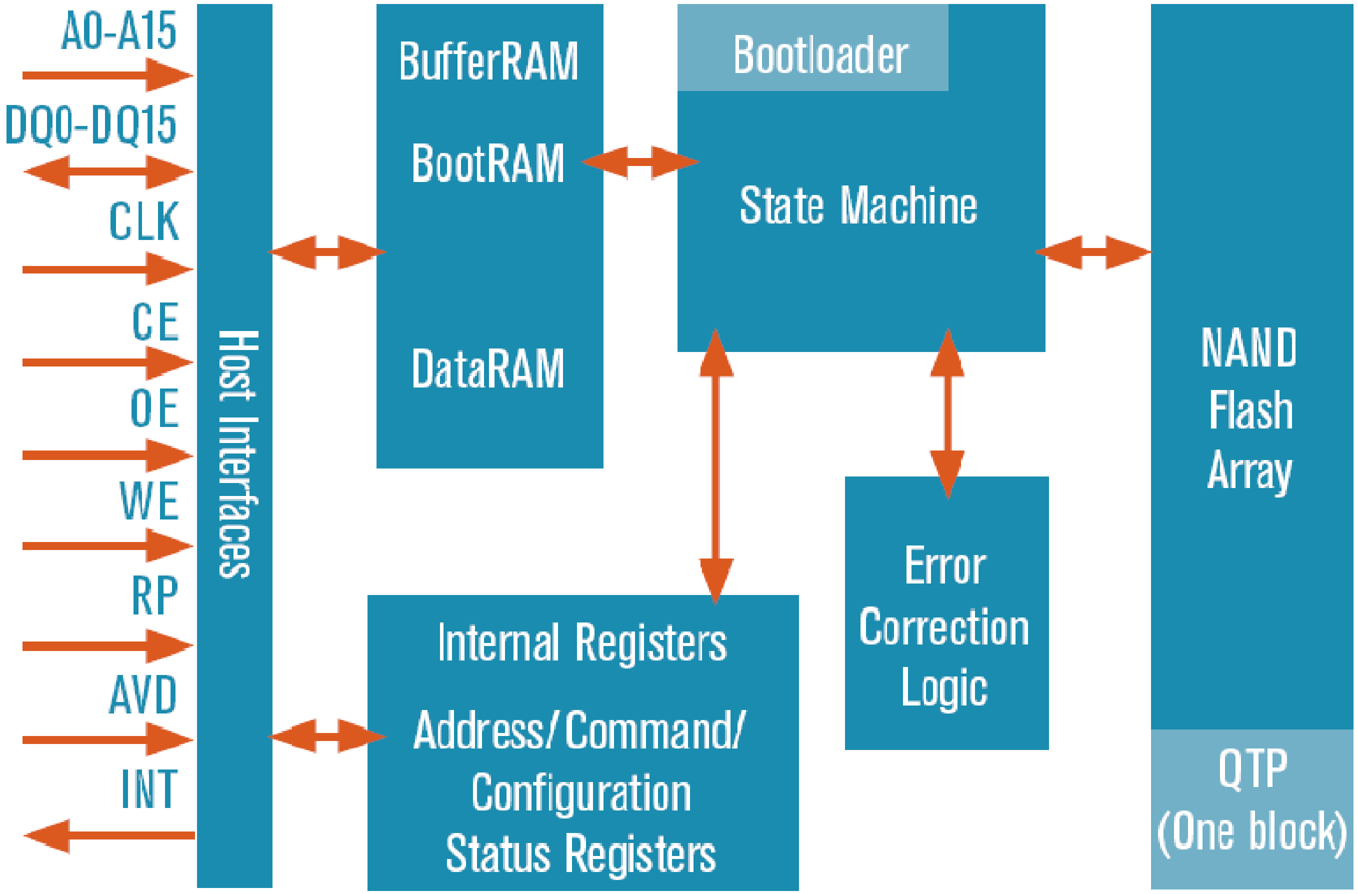
Multi-die



Source: Zeen Info. Tech.

NAND Flash Types (4)

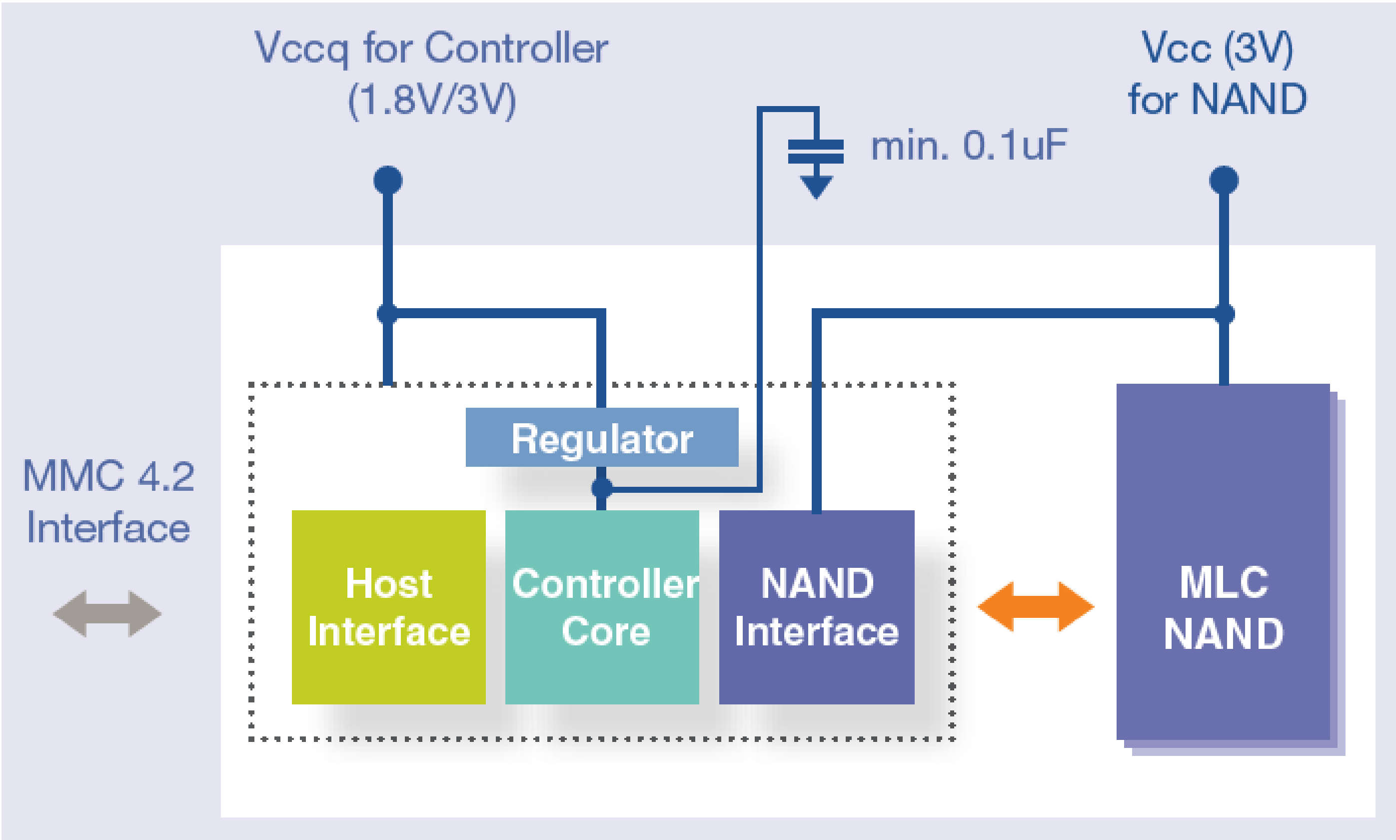
- **Samsung OneNAND™**



Source: Samsung's OneNAND ebrochure

NAND Flash Types (5)

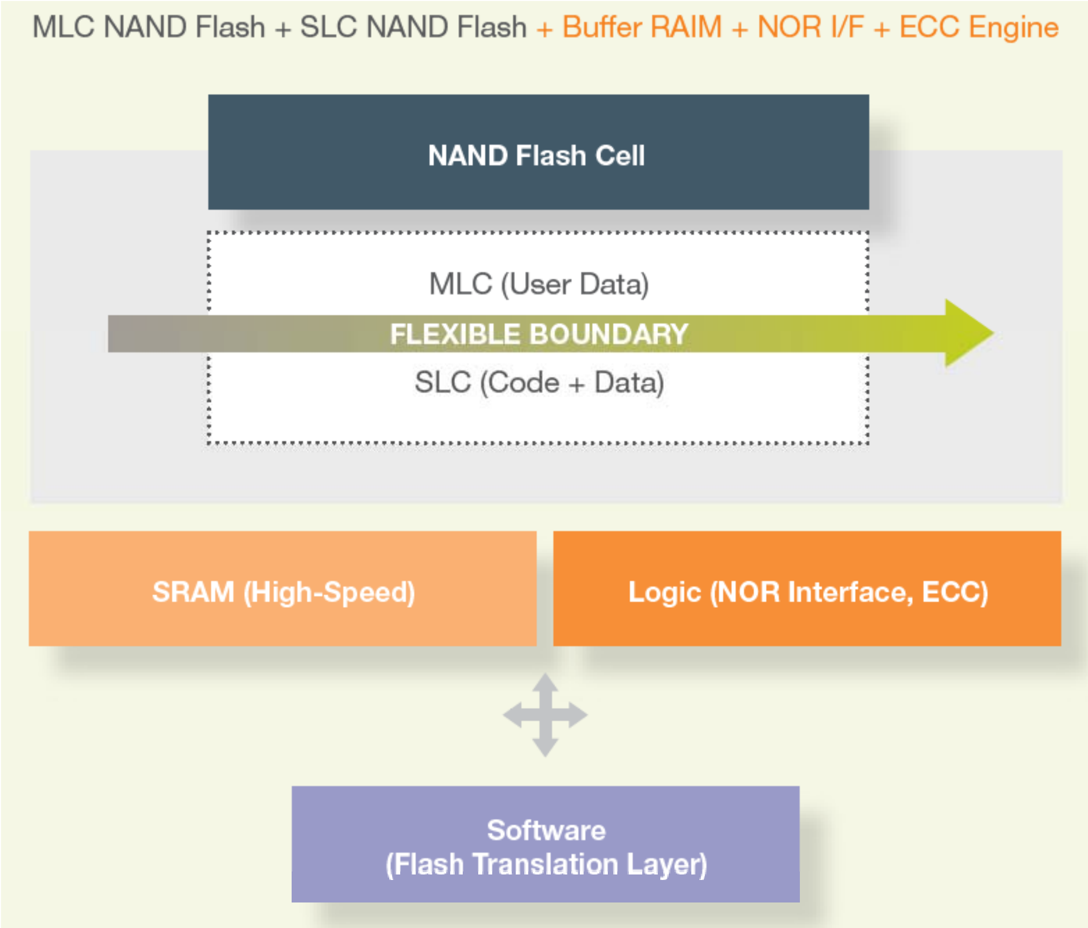
- **Samsung moviNAND™**



Source: Samsung Fusion Memory

NAND Flash Types (6)

- **Samsung Flex-OneNAND™**



Source: Samsung Fusion Memory

NAND Constraints (1)

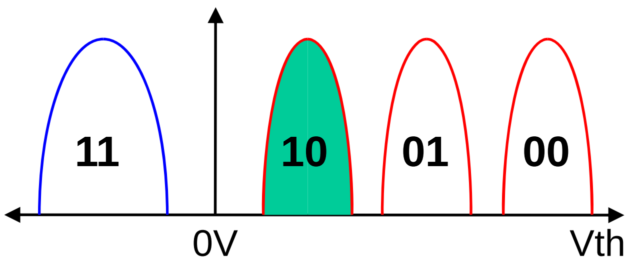
- **Bit Errors**
  - Error correction codes (ECC) in spare area
- **Bad Blocks**
  - Factory-marked bad blocks
  - Run-time bad blocks
  - Bad block management
- **Limited Program/Erase Cycles**
  - 100K for SLCs
  - 10K for MLCs
  - Wear-leveling

NAND Constraints (2)

- **NOP**
  - Partial-page programming
  - 1 / sector for most SLCs (4 for 2KB page)
  - 1 / page for most MLCs
- **Sequential Page Programming**
  - For large block SLCs and MLCs
  - From page 0 to page 63 for SLCs
  - From page 0 to page 127 for MLCs

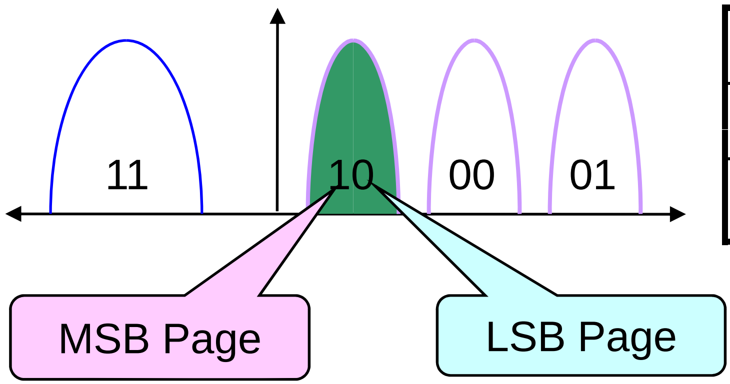
NAND Constraints (3)

- **Pair-page Programming in MLCs**
  - Performance difference
  - Interference



I/O-split MLCs

IO7	IO6	IO5	IO4	IO3	IO2	IO1	IO0	
1	0	0	1	0	0	1	1	Page 0

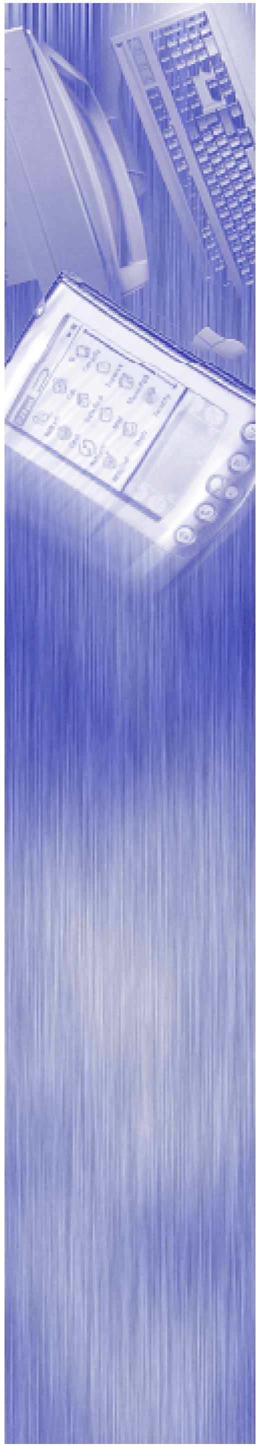


Page-split MLCs

IO7	IO6	IO5	IO4	IO3	IO2	IO1	IO0	
1	1	0	0	1	1	0	0	Page 0
1	0	0	1	1	0	0	1	Page 1

Beauty and the Beast

- **NAND Flash memory is beauty.**
  - Small, light-weight, robust, low-cost, low-power non-volatile device
- **NAND Flash memory is a beast.**
  - Much slower program/erase operations
  - No in-place-update
  - Erase unit > write unit
  - Limited lifetime (10K~100K program/erase cycles)
  - Bad blocks, …
- **Software support for NAND flash memory is very important for performance & reliability.**



CS632/SEP564: Embedded Operating Systems (Fall 2008)

Memory Architectures for Mobile Devices

Requirements

Code

Mobile

Consumer Electronics

Networking

Read	Writes	Density	Reliability
Fast Random	Medium	Small – Medium	No bad bits

Data

Cards

MP3

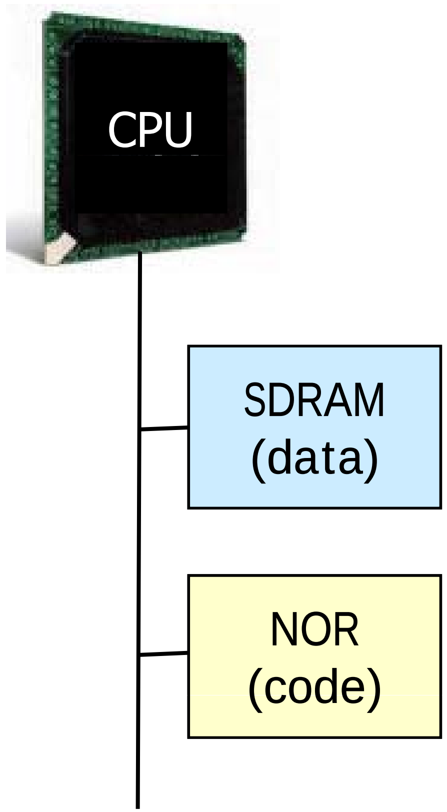
USB Drives

Read	Writes	Density	Reliability
Fast Sequential	Fast	Large	Bad bits allowed

Source: "Non-Volatile Memories", Intel Corp.

NOR XIP

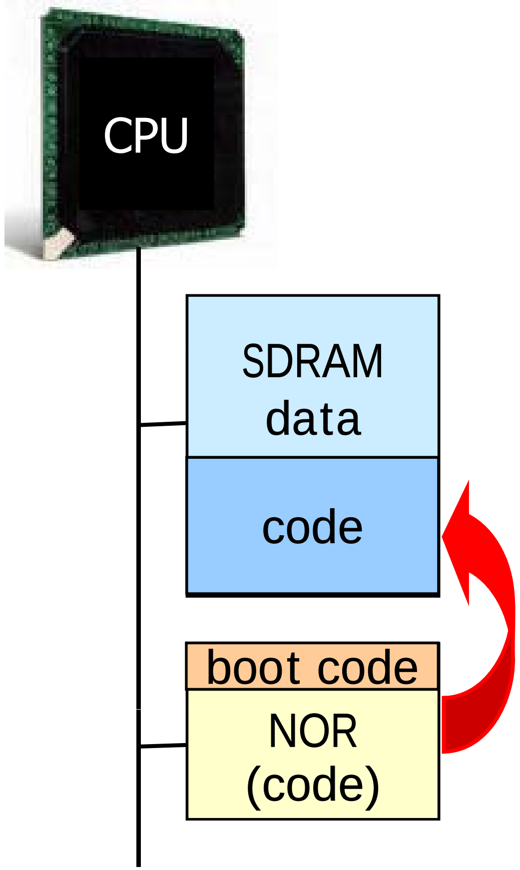
- **Pros**
  - Simple, easy to design
  - Execute-In-Place (XIP)
  - Predictable read latency
  - Code + Data in NOR
  - Firmware upgrades
- **Cons**
  - Slow read speed
  - Much slower write speed
  - The high cost of NOR



NOR Shadowing

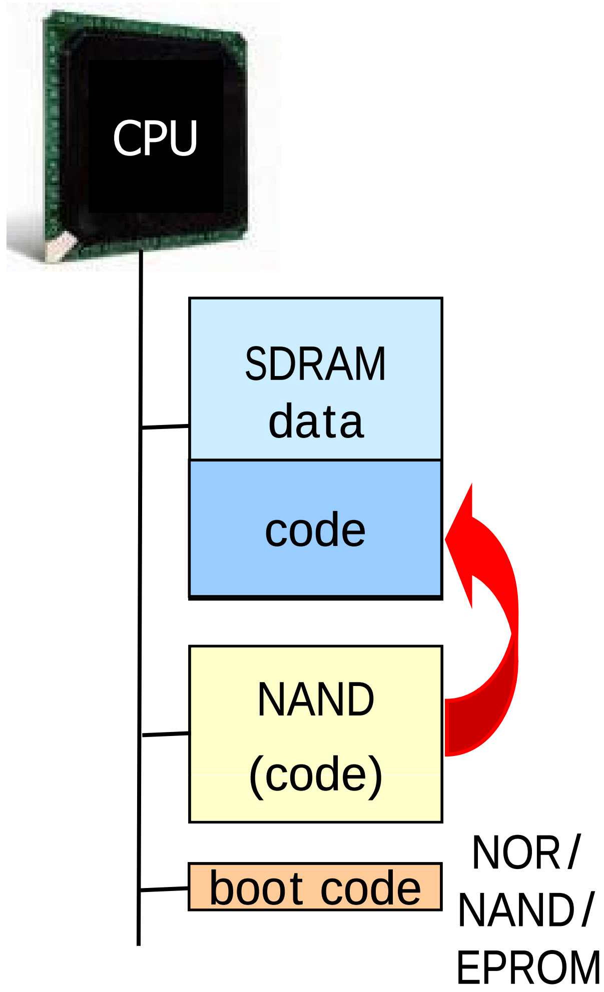
- **Pros**
  - Faster read and write
  - Easy boot-up
  - Use a relatively pricey NOR only to boot up the system
  - Code can be compressed

- **Cons**
  - Larger DRAM needed
  - Require more design time
  - Not energy efficient



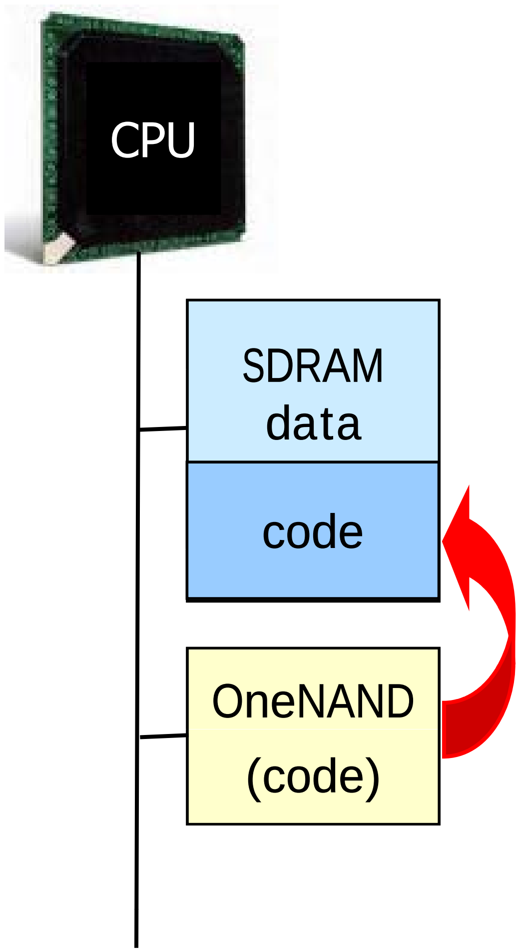
NAND Shadowing

- **Pros**
  - Faster read and write
  - Cost effective
  - NAND for both code and data storage
- **Cons**
  - Require a special boot mechanism
  - Extensive ECC for NAND
  - Larger DRAM needed
  - Require more design time
  - Not energy efficient



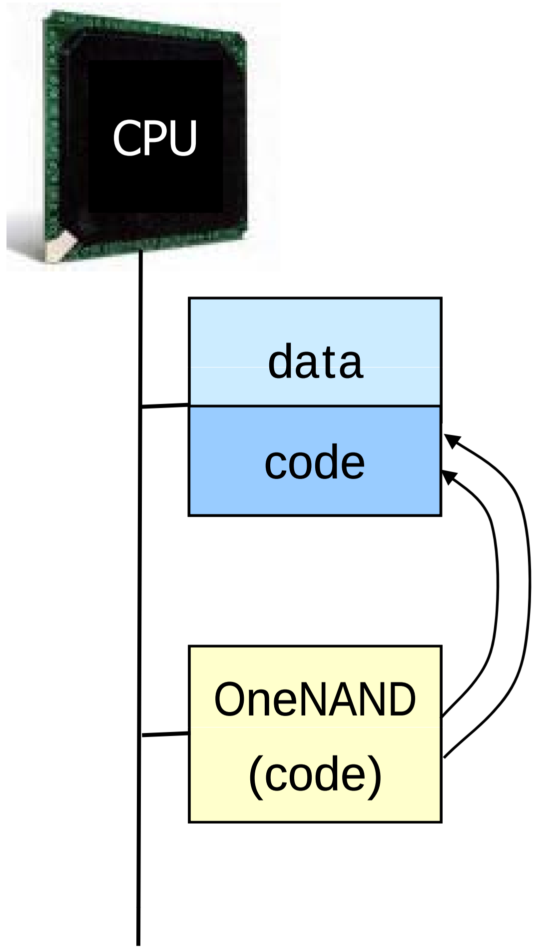
Hybrid NAND Shadowing

- **Pros**
  - Much faster read and write speed
  - ECC embedded
  - Cost effective
  - NAND for both code and data storage
- **Cons**
  - Larger DRAM needed
  - Not energy efficient



NAND Demand Paging

- **Pros**
  - Less DRAM required
  - Low cost
  - Energy efficient
  - NAND for both code and data storage



- **Cons**
  - Require MMU-enabled CPU
  - Unpredictable read latency
  - Complex to design and test

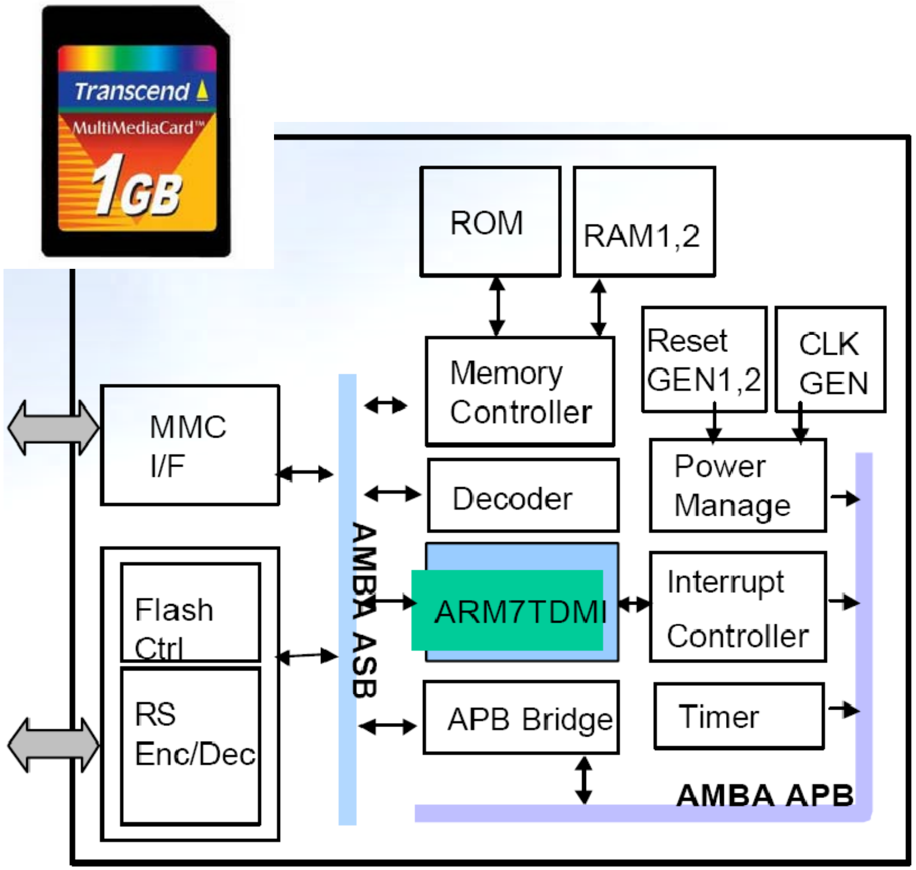
CS632/SEP564: Embedded Operating Systems (Fall 2008)

Flash Translation Layers (FTL)

KAIST

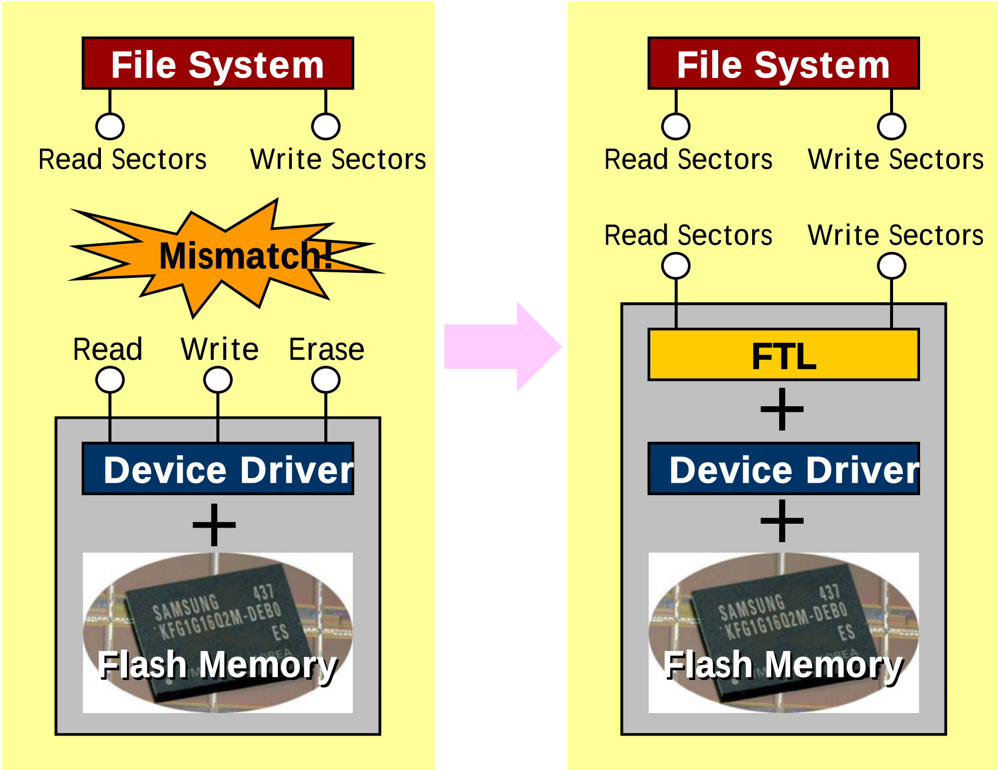
FTL (1)

- **Flash Cards Internals**



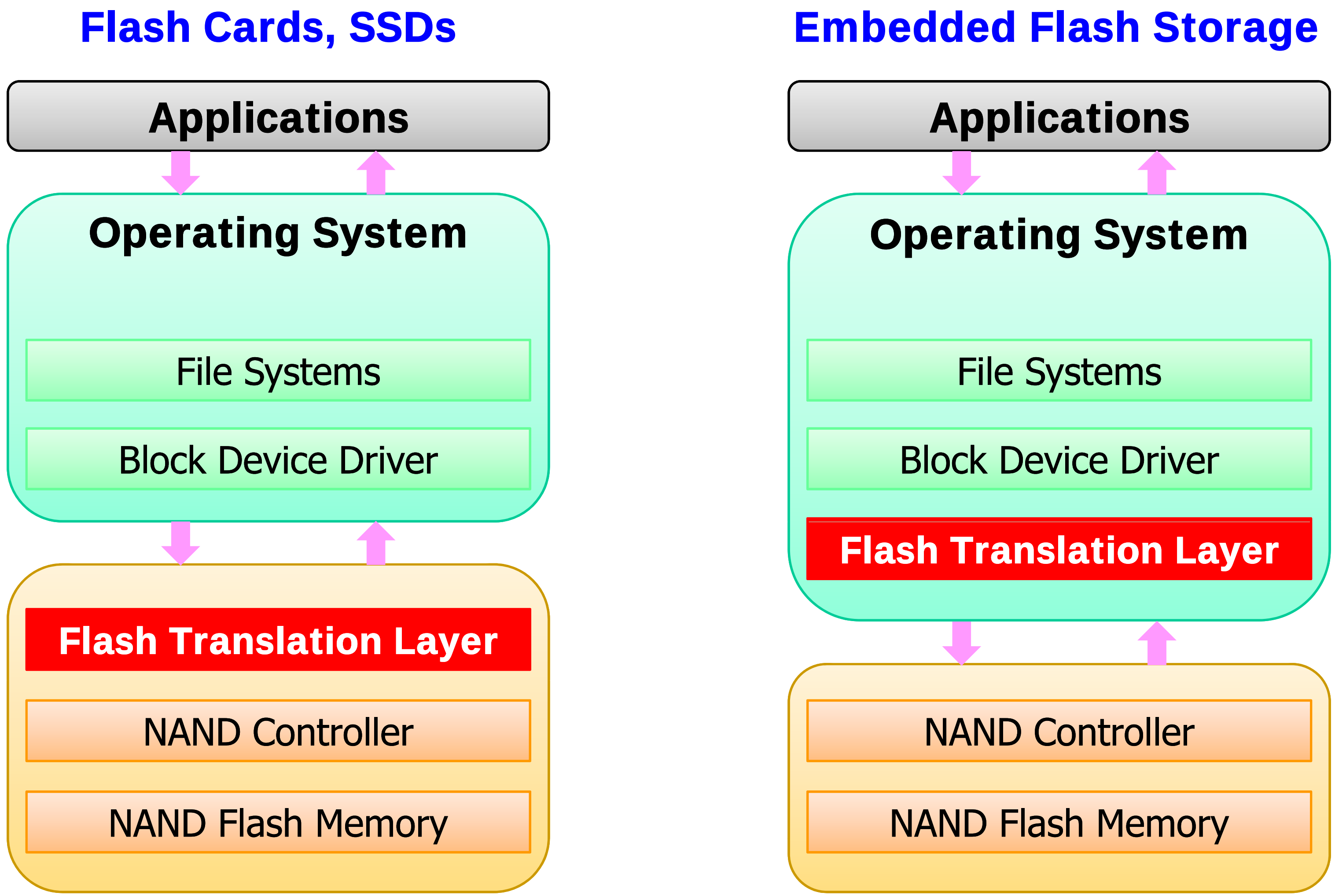
FTL (2)

- **FTL**
  - A software layer to make NAND flash fully emulate traditional block device (e.g., disks)
- **Why FTL?**
  - No in-place-update
  - Bulk erase
  - Asymmetry in read and write speeds



Source: Zeen Info. Tech.

FTL (3)

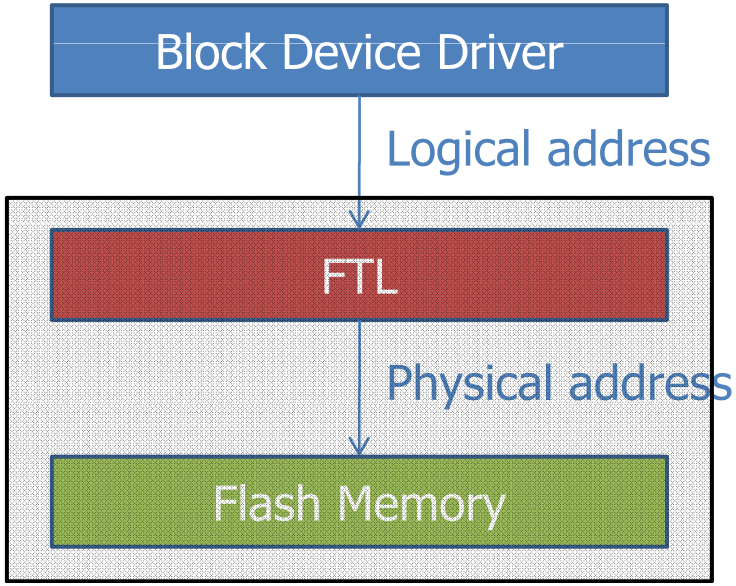

Flash Cards, SSDs
Applications
Operating System
File Systems
Block Device Driver
Flash Translation Layer
NAND Controller
NAND Flash Memory
Embedded Flash Storage
Applications
Operating System
File Systems
Block Device Driver
Flash Translation Layer
NAND Controller
NAND Flash Memory

FTL (4)

- **For Performance**
  - Sector mapping (address translation)
  - Garbage collection
- **For Reliability**
  - Power-off recovery
  - Wear-leveling
  - Bad block management
  - Error correction code (ECC)

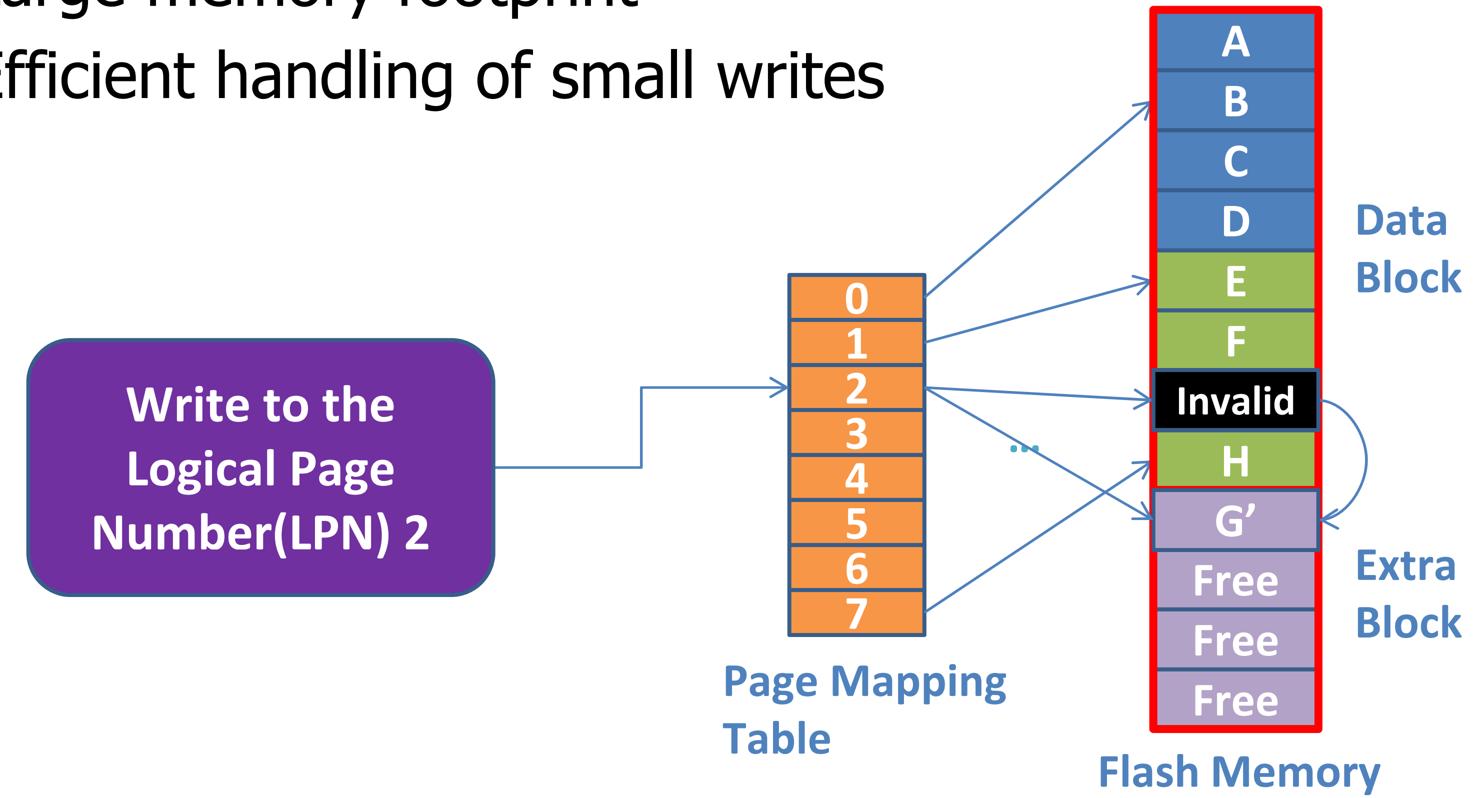
Address Translation

- **Mapping granularity**
  - Page mapping
  - Block mapping
  - Hybrid mapping
- **Block organization**
  - In-place
  - Out-of-place
- **Managing mapping info.**
  - Per block
  - Map block



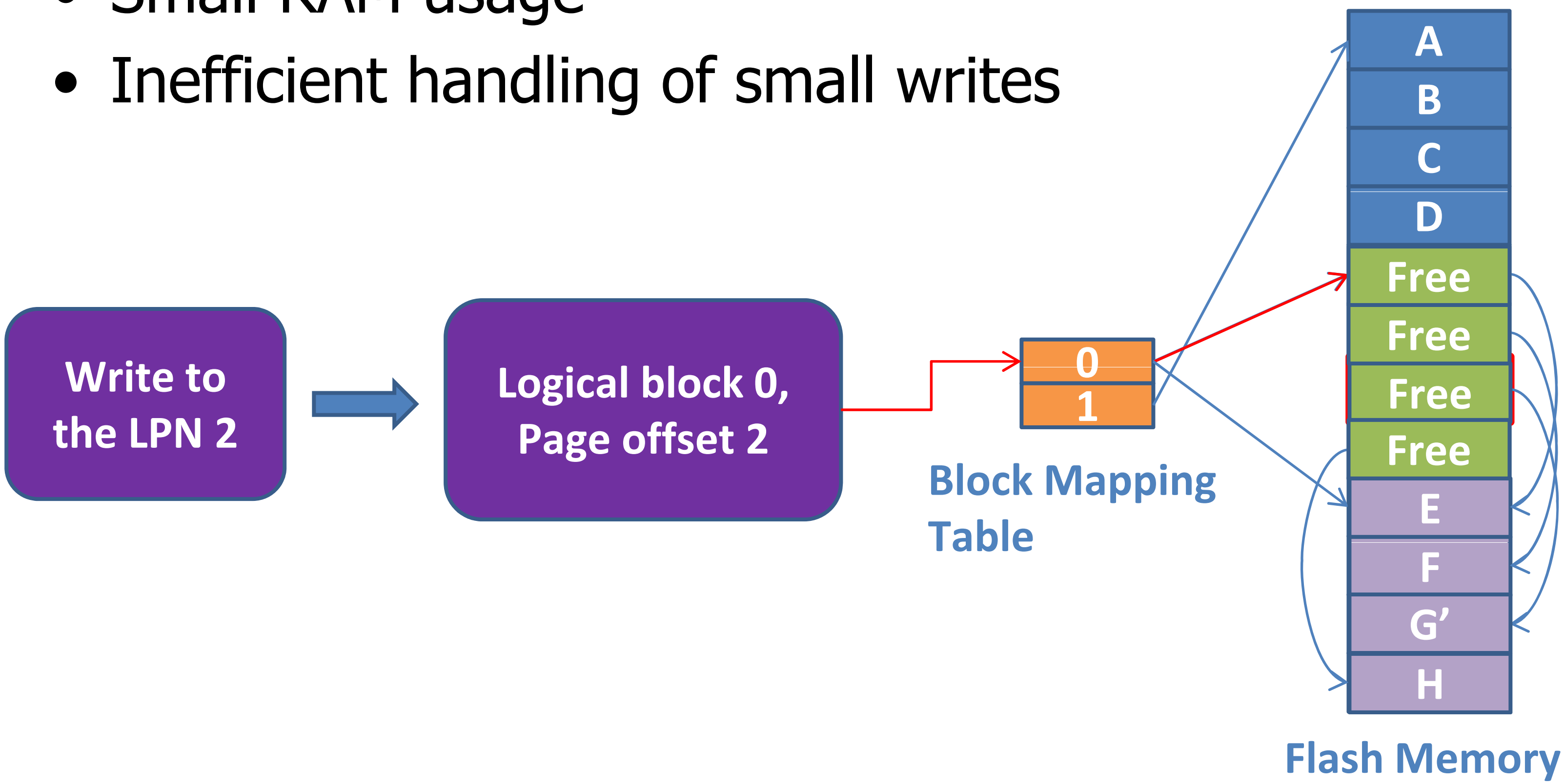
Page Mapping

- **Characteristics**
  - Each table entry maps one page
  - Large memory footprint
  - Efficient handling of small writes



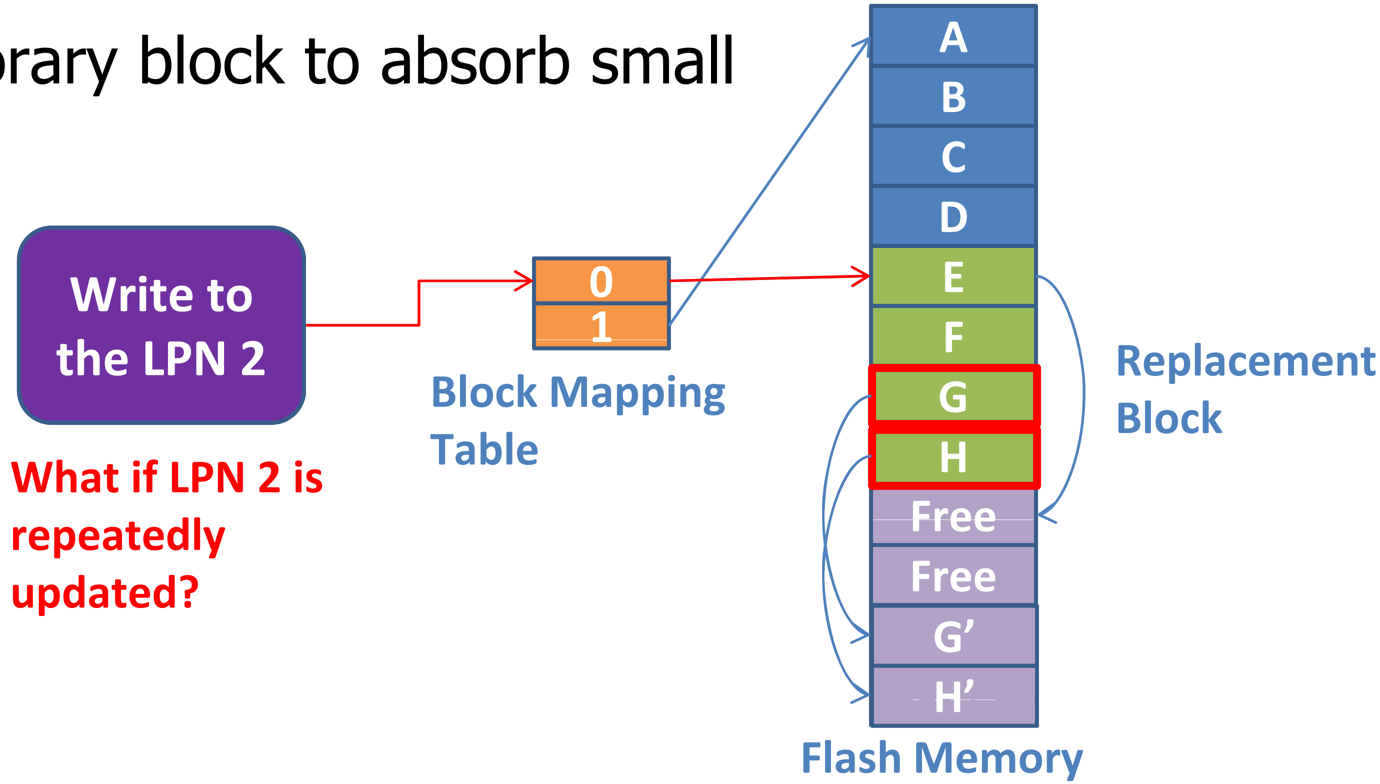
Naïve Block Mapping

- **Characteristics**
  - Each table entry maps one block
  - Small RAM usage
  - Inefficient handling of small writes



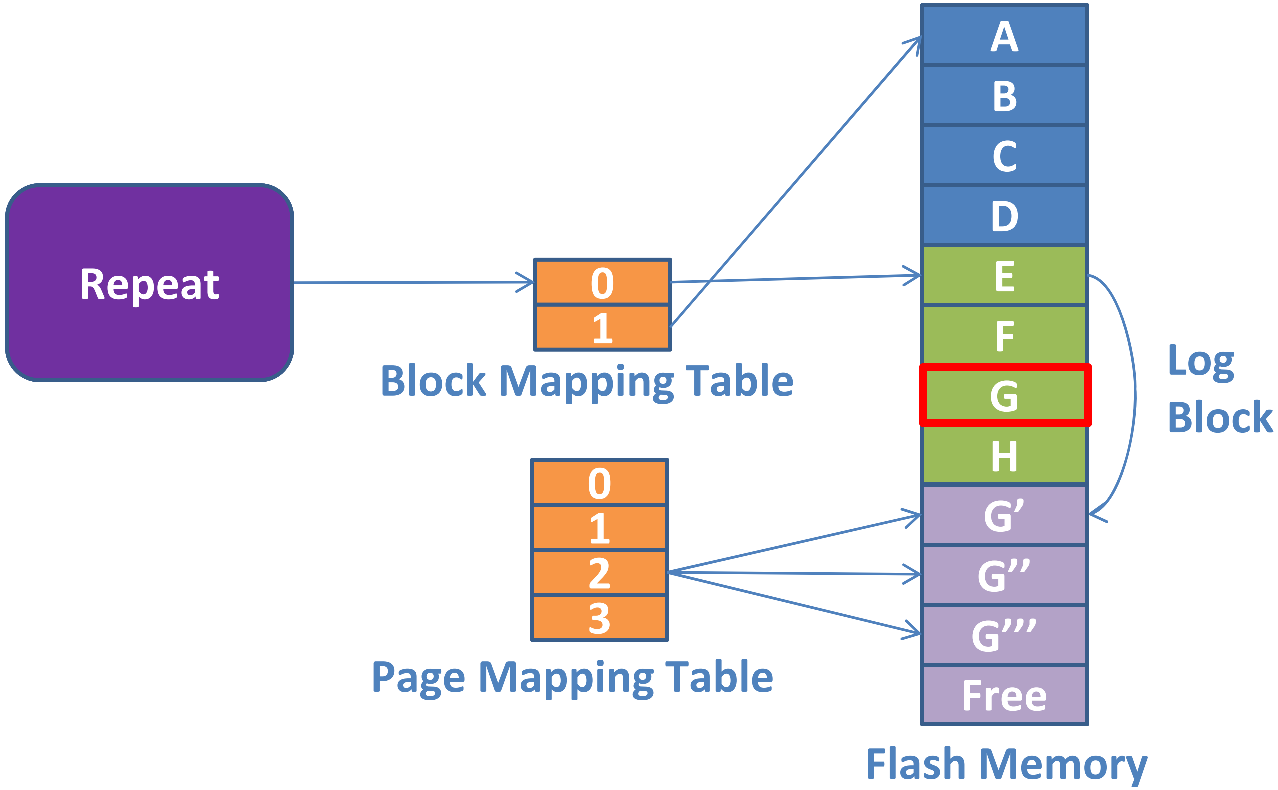
Replacement Block Scheme

- **Data Block**
  - Storage for an ordinary data
- **Replacement Block**
  - A temporary block to absorb small writes



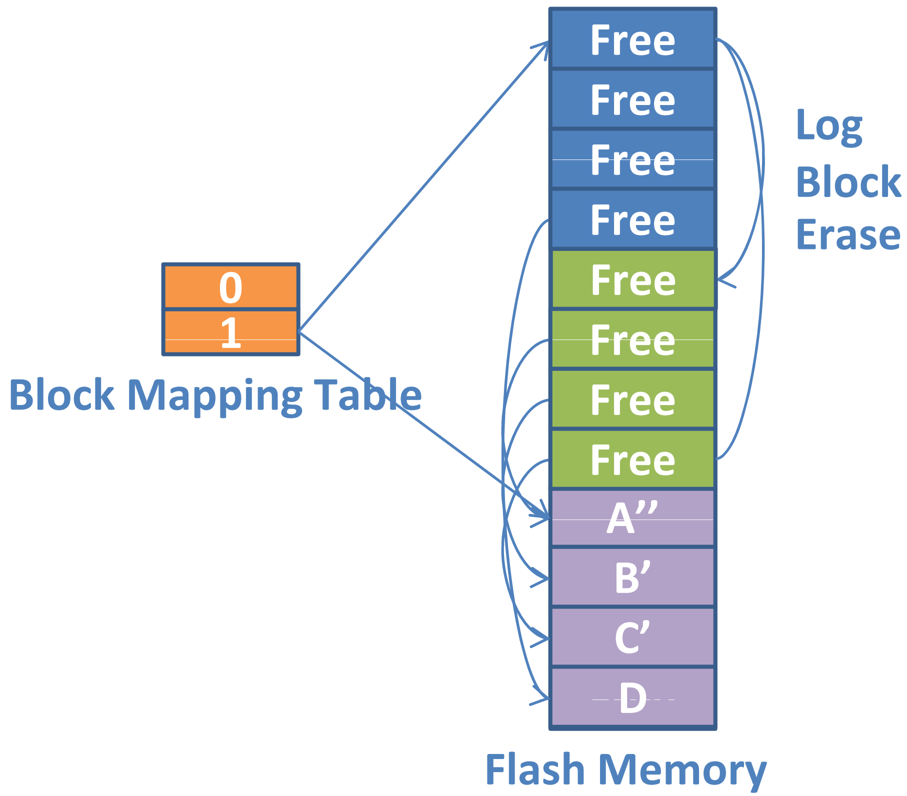
Log Block Scheme (1)

- **Log Block**
  - A temporary storage for small writes
  - Incremental updates from the first page



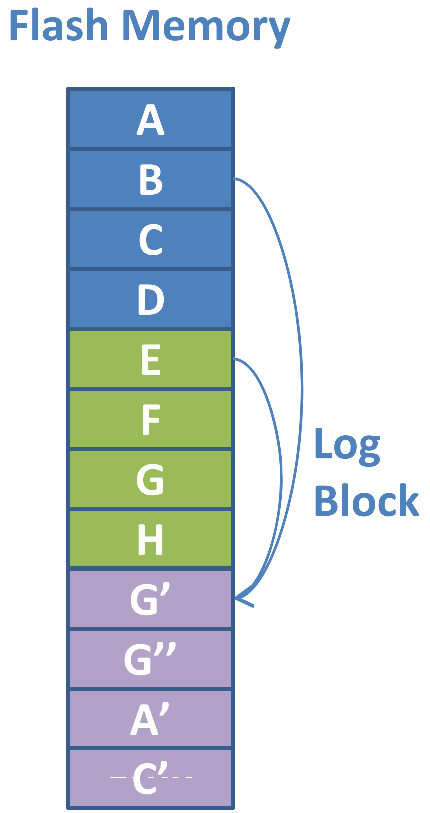
Log Block Scheme (2)

- **Block Merge**
  - When there is no free log block to allocate.
  - Allocate a free block and copy all the up-to-date pages



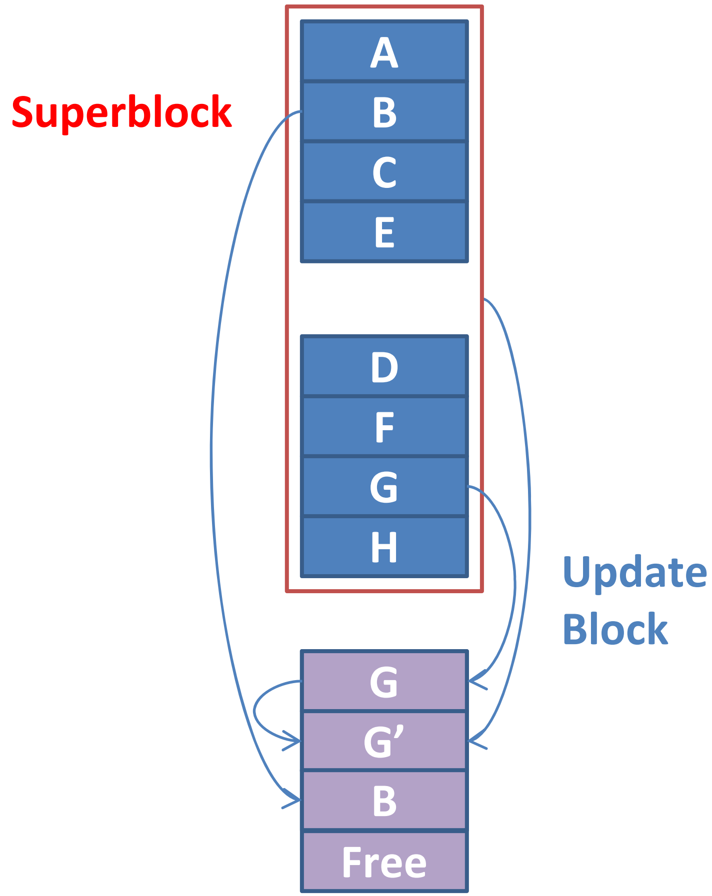
FAST

- **Fully Associative Sector Translation (FAST)**
  - Sharing log blocks
- **Pros**
  - Increase log block utilization
  - Reduces the number of erase operations
- **Cons**
  - Increased merge time



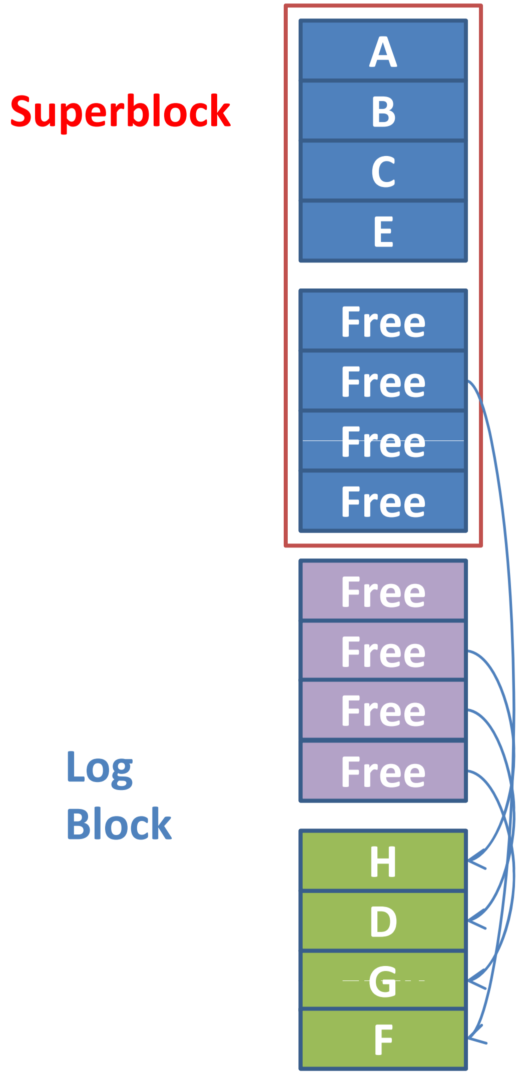
Superblock Scheme (1)

- **Superblock**
  - N logically adjacent blocks
  - Page-level address translation within a superblock
- **Update Block**
  - A superblock shares log blocks
  - More than one log blocks can be allocated to a superblock.



Superblock Scheme (2)

- **Merge Operation**
  - Gathers hot pages into the same block



Comparison

		Replacement block scheme	Log block scheme	FAST	Superblock scheme
Data blocks	Terminology	Data blocks	Data blocks	Data blocks	D-blocks
	Management scheme	In-place	In-place	In-place	Out-of-place
	The degree of sharing	1	1	1	N
Update blocks	Terminology	Replacement blocks	Log blocks	Random/ sequential log blocks	U-blocks
	Management scheme	In-place	Out-of-place	Out-of-place	Out-of-place
	The degree of sharing	1	1	P	N

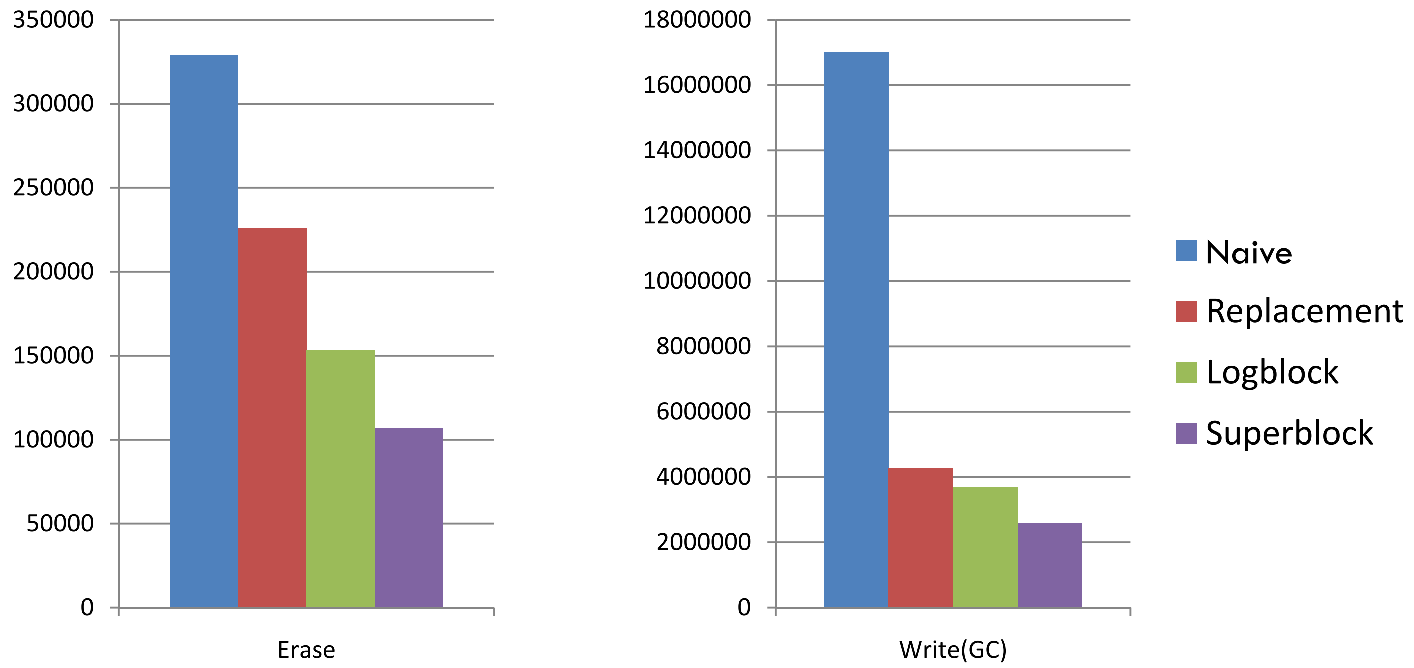
N: superblock size, P: the number of pages in a block

Performance (1)

- **Simulation Environment**
  - 4GB flash memory
    - Large block SLC NAND (2KB page, 128KB block)
  - FTL schemes
    - Naïve block mapping
    - Replacement block
    - Log block
    - Superblock
  - Workload
    - Trace from PC using NTFS

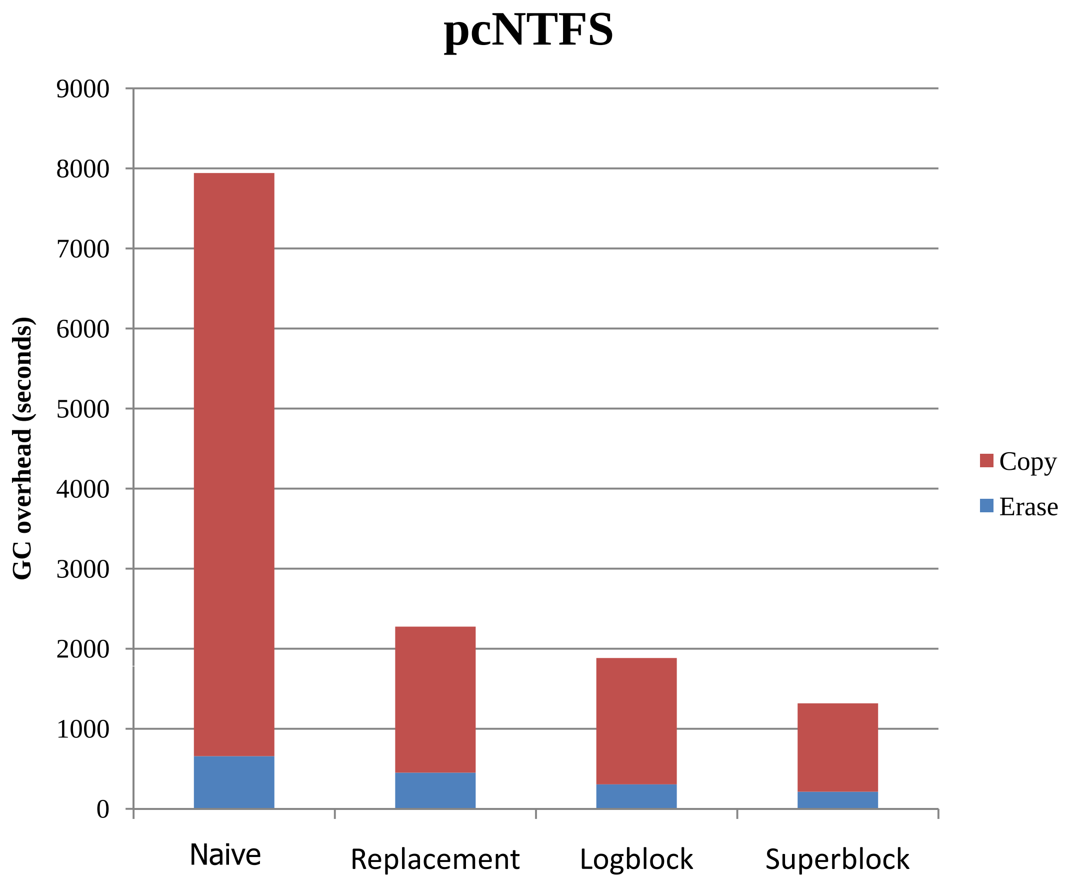
Performance (2)

- **Extra Erase and Write Operations**
  - 256 extra blocks



Performance (3)

- **Garbage Collection Overhead**



Bad Block Management (1)

- **Bad Blocks**
  - Factory-marked bad blocks
    - Marked "non-FF" at the first byte in the spare area (either 1st or 2nd page of every initial bad block)
    - < 2% of entire blocks
    - The first block is guaranteed to be good.
    - Should not erase or program factory-marked bad blocks.
  - Run-time bad blocks
    - Bit errors
    - Endurance cycle
    - PROGRAM/ERASE failures

Bad Block Management (2)

- **Issues**
  - Bad block table organization
    - Space
    - Bad block #, Remapped block #
    - Very sparse
  - Fast lookup
    - Time
  - Number of reserved blocks

Wear-Leveling (1)

- **Wear-leveling**
  - A software technique for balancing the erase counts of physical blocks to fully utilize the lifetime of NAND flash.

	With Wear-Leveling	Without Wear-Leveling
Feature	Maintain erase counts in both flash and RAM	Randomly select a block at run-time
Pros	Maximize the lifetime up to spec. value	No performance overhead
Cons	Performance degradation RAM overhead	Excessive wear-out in the specific area depending on the workload

Wear-Leveling (2)

- **Naïve Wear-leveling**
  - Use blocks in a round-robin fashion
  - Performance!
  - JFFS
- **Sophisticated Wear-leveling**
  - Keep erase count per block (in spare area)
  - Dynamic mapping
  - Consider erase counts during garbage collection
  - Block swapping

Wear-Leveling (3)

Hot-Cold Swapping

- Swap if $max(EC_i) - min(EC_i) > Threshold$

CAT (Cost, Age, Times)

- Erase the block with minimizes the following score:

$$Cleaning\ Cost_i \times \frac{1}{Age_i} \times Number\ of\ Cleaning_i = \frac{\mu_i \times \varepsilon_i}{(1-\mu_i) \times a_i(t)}$$

  - μ_i : utilization (the percentage of valid data)
  - $a_i(t)$: the elapsed time since the block was erased
  - ε_i : the erase count

Wear-Leveling (4)

- **Issues**
  - Maintaining erase counts
    - Per block
    - Separate area
  - Memory footprint
    - Min/Max erase count
    - All the erase counts
    - Building in-memory data structure
  - Performance degradation
    - Overhead due to wear-leveling

Error Correction Codes

- **ECC**
  - Hardware vs. Software

Error Correction Level	Bits Required in the NAND Flash Spare Area		
	Hamming	Reed-Solomon	BCH
1	13	18	13
2	N/A	36	26
3	N/A	54	39
4	N/A	72	52
5	N/A	90	65
6	N/A	108	78
7	N/A	126	91
8	N/A	144	104
9	N/A	162	117
10	N/A	180	130

Source: Micron Technology, Inc.

Power-Off Recovery

- **Power-Off Recovery**
  - For every modifying operations: write, reclaim
  - Atomicity (transactional operation)
  - Scan vs. checkpointing
    - Per Block
    - Map Block
  - File system level vs. FTL level
  - System-wide power-management scheme is necessary
    - On detecting power-off, system should notify devices drivers to safely close the current operation
  - Super capacitor?

Summary (1)

- **Small memory footprint**
  - Directly related to the SRAM size ($$$)
- **Handling random writes**
- **Garbage collection**
  - Minimize valid copy during garbage collection
  - Wear-leveling
- **Wear-leveling**
- **Power-off recovery**
- **Real-time performance**

Summary (2)

- **Physical Constraints**
  - Large block NAND
    - Sequential page programming
    - NOP 4
  - MLC (Multi-Level Cell) NAND
    - NOP 1
    - Less endurance cycles (~ 10K)
    - More bit errors
    - Most of spare area is dedicated to ECC/EDC
    - Pair-page programming