

Jin-Soo Kim
(jinsoo.kim@snu.ac.kr)

Systems Software &
Architecture Lab.

Seoul National University

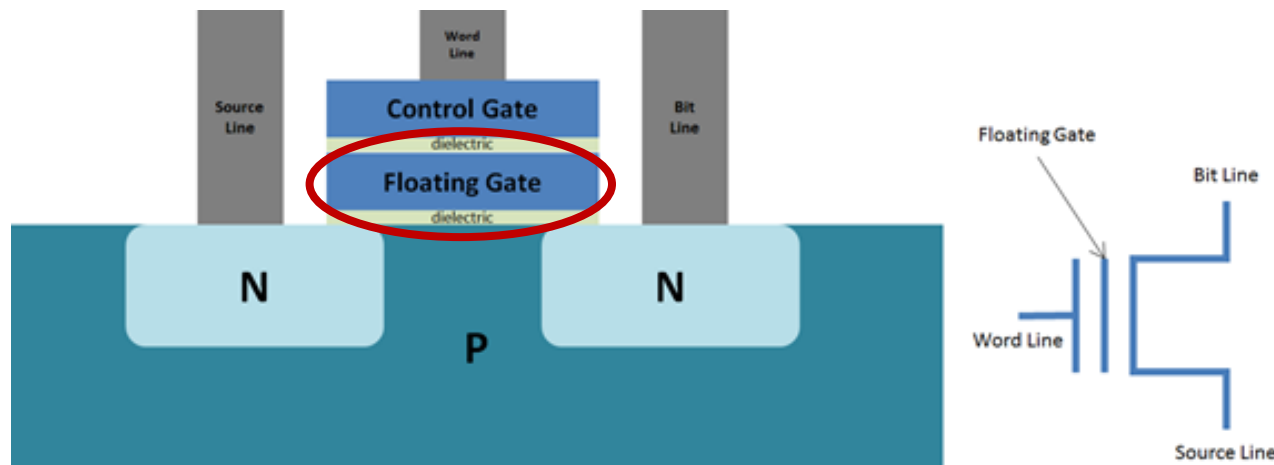
Fall 2023

Solid State Drives (SSDs)



Flash Memory Cell

- Transistor with floating gate
 - The floating gate is insulated all around with an oxide layer
 - Electrons trapped in the floating gate can remain for up to years



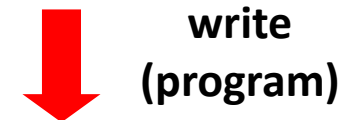
Flash Memory Characteristics

■ Erase-before-write

- Read
- Write or Program: 1 \rightarrow 0
- Erase: 0 \rightarrow 1

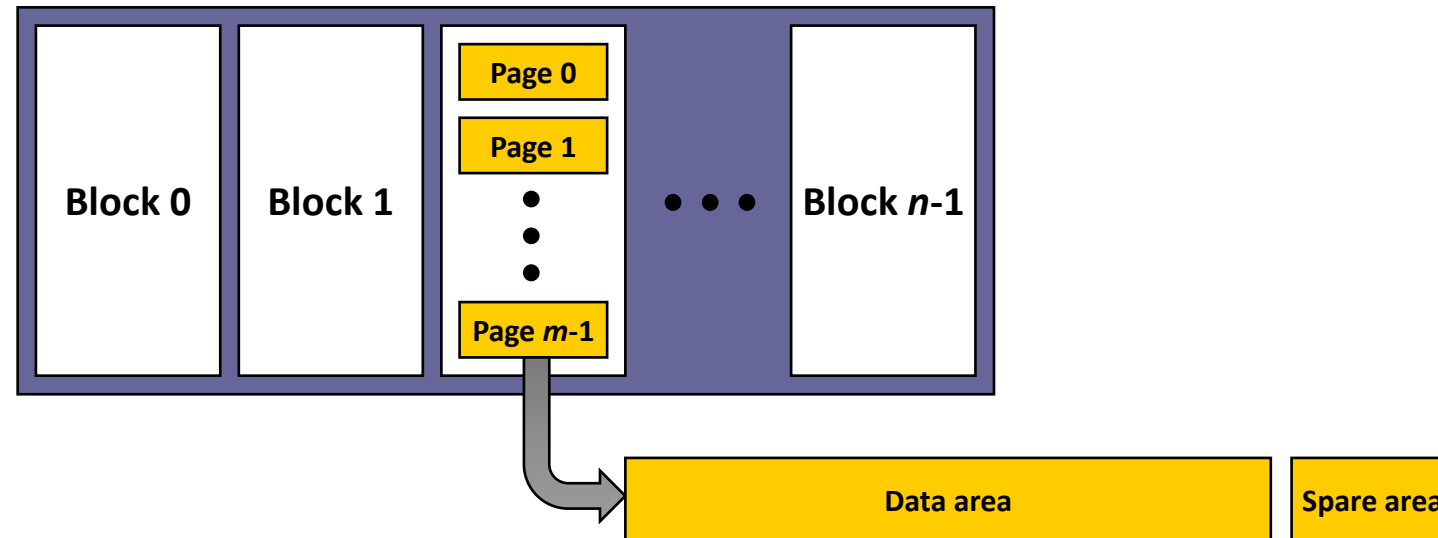
■ Bulk erase

- Program unit:
 - NOR: byte or word
 - NAND: page
- Erase unit: _____



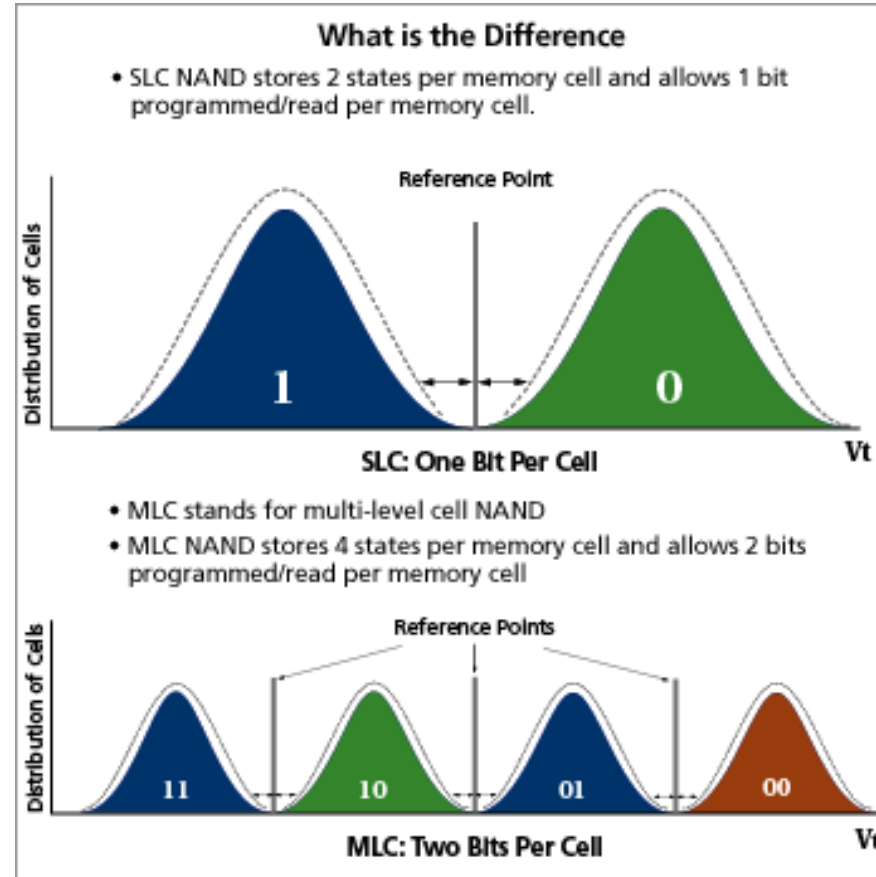
Logical View of NAND Flash

- A collection of **blocks**
- Each block has a number of **pages**
- The size of a block or a page depends on the technology (but, it's getting larger)



NAND Flash Types

- SLC NAND
 - Single Level Cell
 - 1 bit/cell
- MLC NAND
 - Multi Level Cell (misnomer)
 - 2 bits/cell
- TLC NAND
 - Triple Level Cell
 - 3 bits/cell
- 3D NAND



NAND Applications

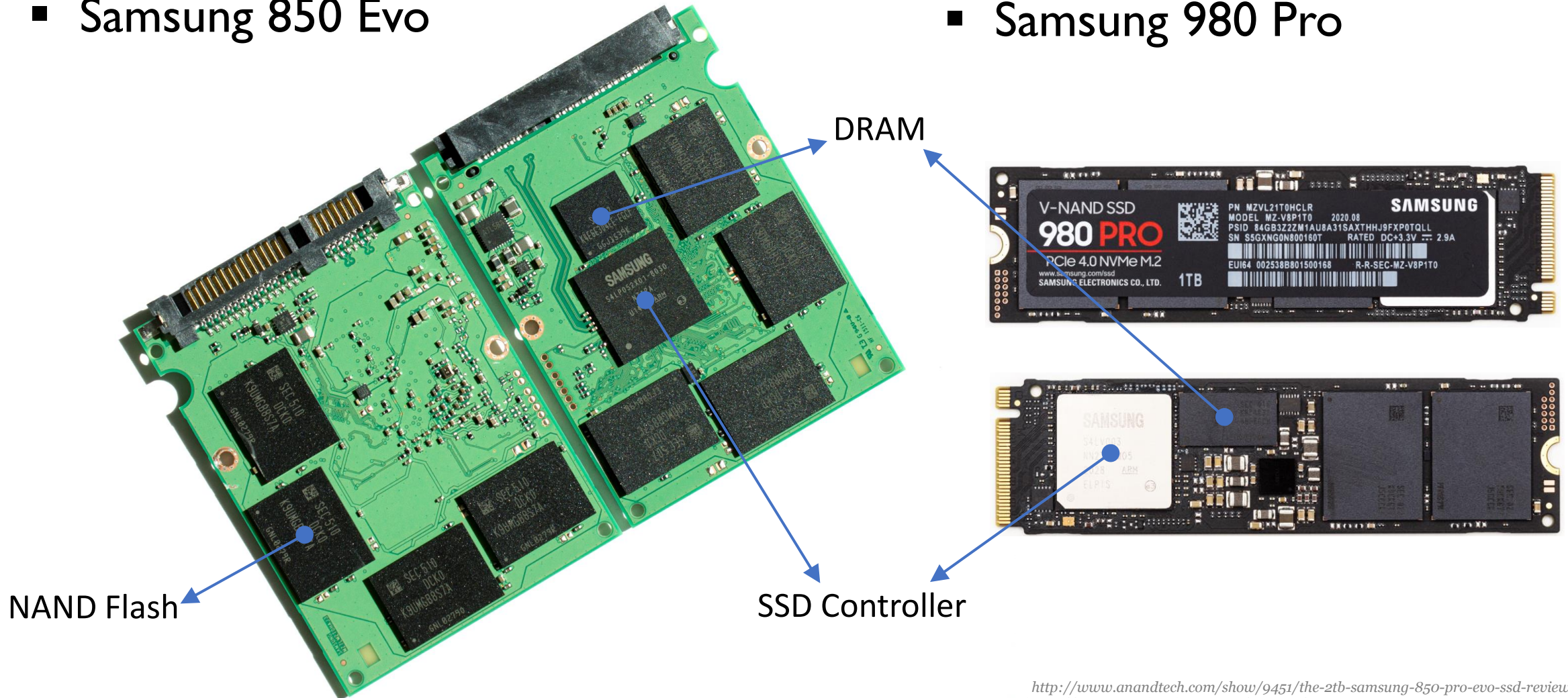
- Universal Flash Drives (UFDs)
- Flash cards
 - CompactFlash, MMC, SD, Memory stick, ...
- Smartphones
 - eMMC (Embedded MMC)
 - UFS (Universal Flash Storage)
- SSDs (Solid State Drives)
- Other embedded devices
 - MP3 players, Digital TVs, Set-top boxes, Car navigators, ...



Anatomy of an SSD

- Samsung 850 Evo

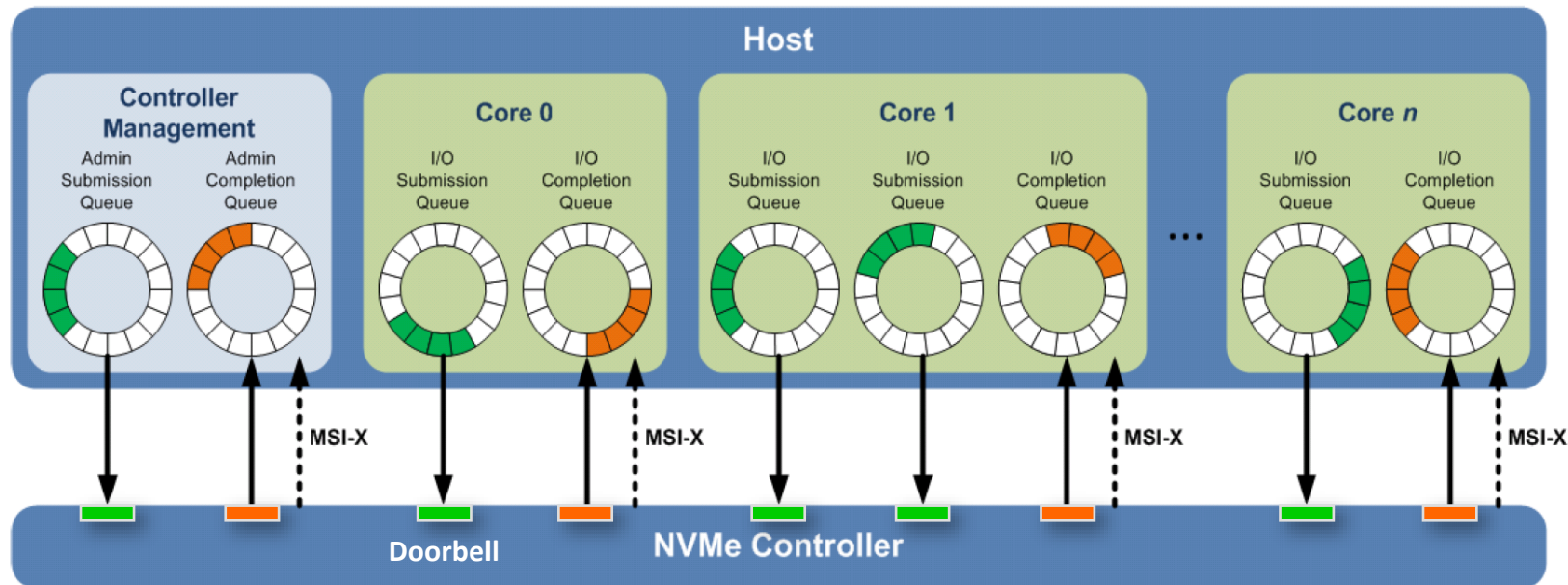
- Samsung 980 Pro



<http://www.anandtech.com/show/9451/the-2tb-samsung-850-pro-evo-ssd-review>
<https://www.tomshardware.com/reviews/samsung-980-pro-m-2-nvme-ssd-review>

NVMe SSD

- PCIe-based (PCIe Gen. 3: 1 GB/s per lane, up to 32 lanes)
- Deep queue: 64K commands per queue, up to 64K queues
- Streamlined command set: only 13 required commands
- One register write to issue a command (“doorbell”)



HDDs vs. SSDs

Feature	SSD (Samsung)	HDD (Seagate)
Model	MZ-V8P2T0 (980 Pro)	ST2000LM003 (SpinPoint M9T)
Capacity	2TB (512Gb 128-Layer 3D V-NAND TLC x 16 dies/chip x 2 chips)	2TB (3 Discs, 6 Heads, 5400 RPM)
Form factor	M.2 (2280), 55g	2.5", 130g
DRAM	2 GB	32 MB
Host interface	NVMe (PCIe 4.0 x 4, 8GB/s)	SATA-3 (600 MB/s)
Power consumption (Active / Idle / Sleep)	6.1 W / 0.035 W / 0.005 W	2.3 W / 0.7 W / 0.18 W
Performance	Sequential read: 7000 MB/s Sequential write: 5100 MB/s Random read: 1,000K IOPS (QD32) Random write: 1,000K IOPS (QD32) Random read: 22,000 IOPS (QD1) Random write: 60,000 IOPS (QD1)	Sequential read: 124 MB/s Sequential write: 124 MB/s Random read: 56 IOPS Random write: 98 IOPS Power-on to ready: 3.5 sec Average seek: 12/14 ms Average latency: 5.6 ms
Price ¹	246,050 won (123won/GB)	66,000 won (33won/GB)

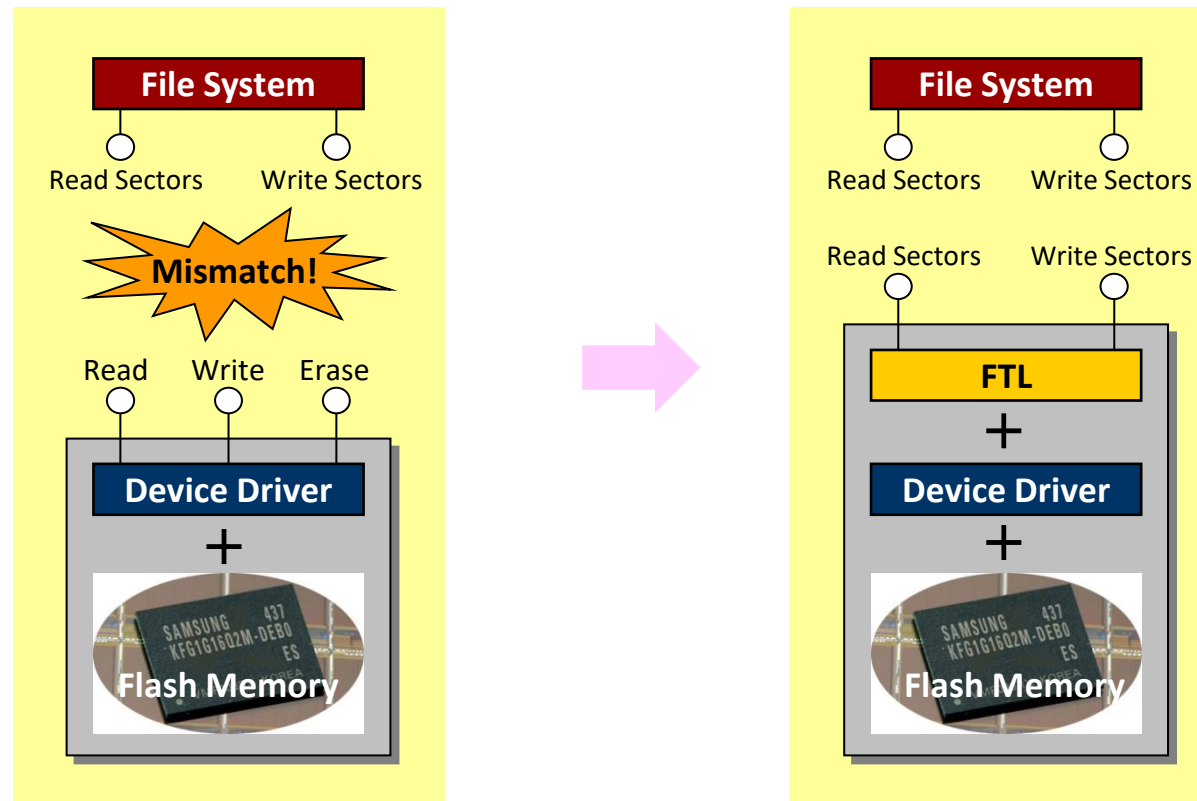
¹ <http://www.danawa.com> (As of Nov. 29, 2023)

NAND Constraints

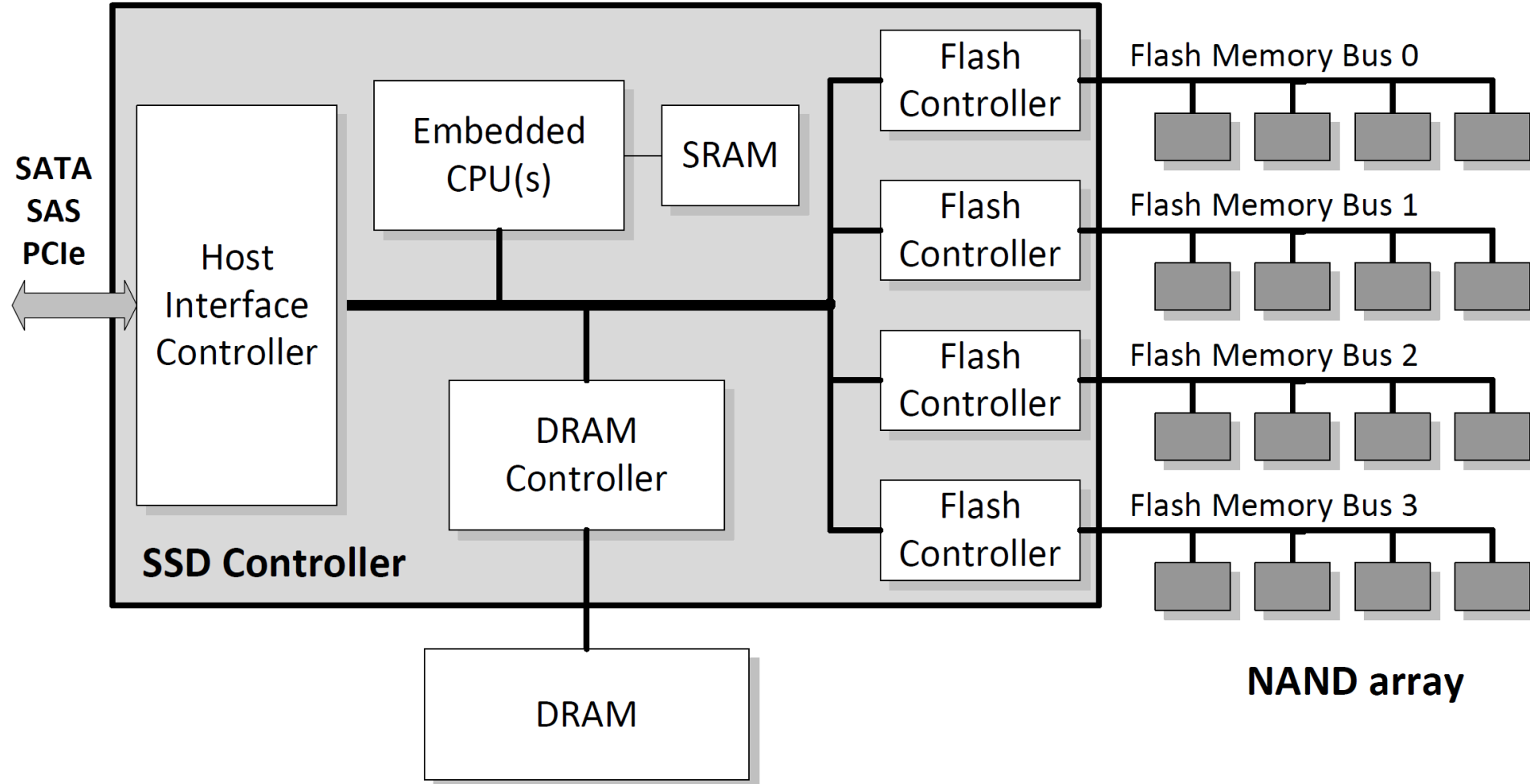
- Asymmetrical read/program time
- No in-place update
 - Erase unit (block) is much larger than read/program unit (page)
 - Require sector remapping (or address translation) and garbage collection
- Bit errors
 - Require the use of error correction codes (ECCs)
- Bad blocks
 - Require bad block remapping
- Limited program/erase cycles
 - $< 100K$ for SLCs, $< 3K$ for MLCs, $< 1K$ for TLCs
 - Require wear-leveling

Flash Translation Layer (FTL)

- A software layer to make NAND flash fully emulate traditional block devices (e.g., disks)

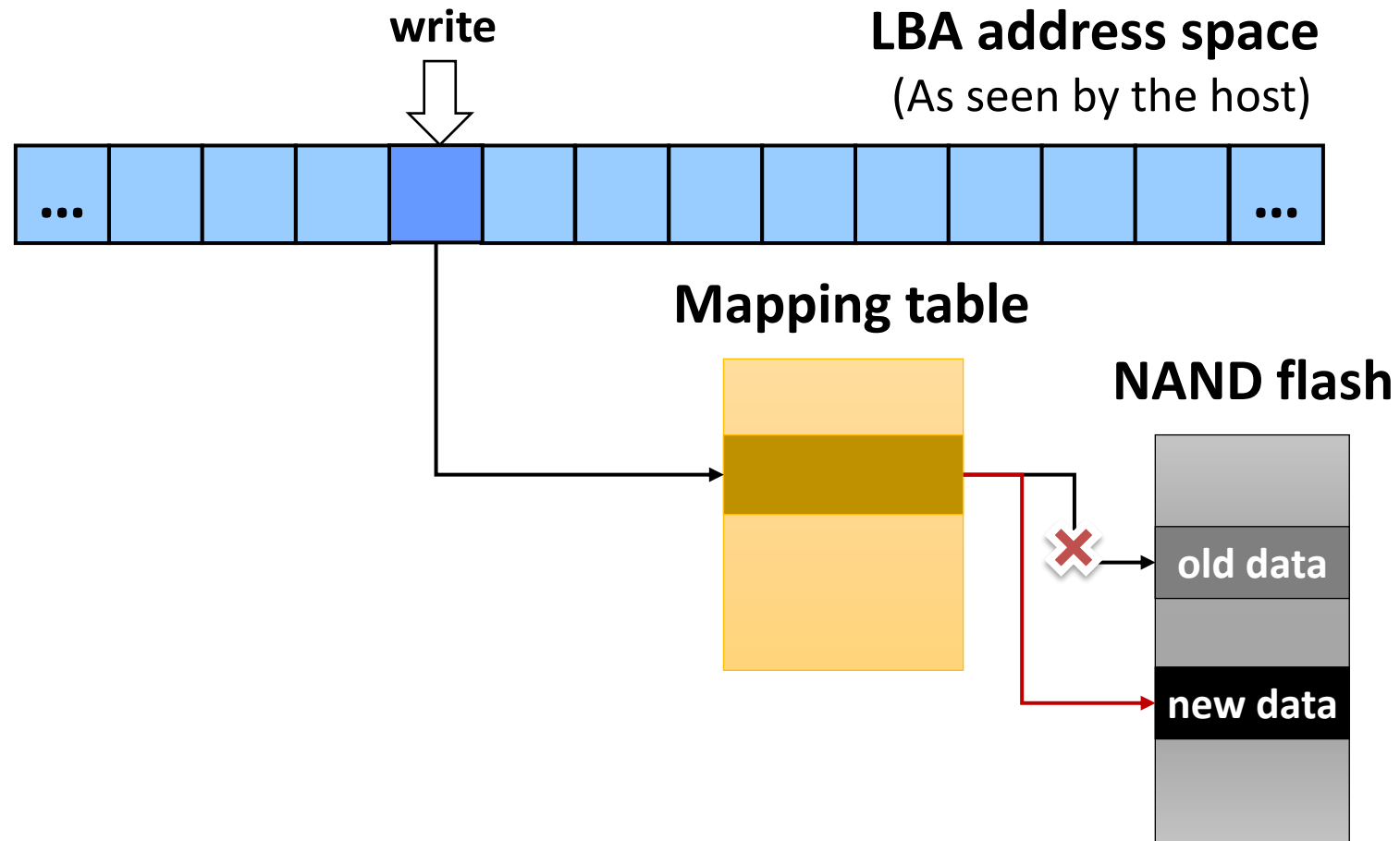


SSD Internals



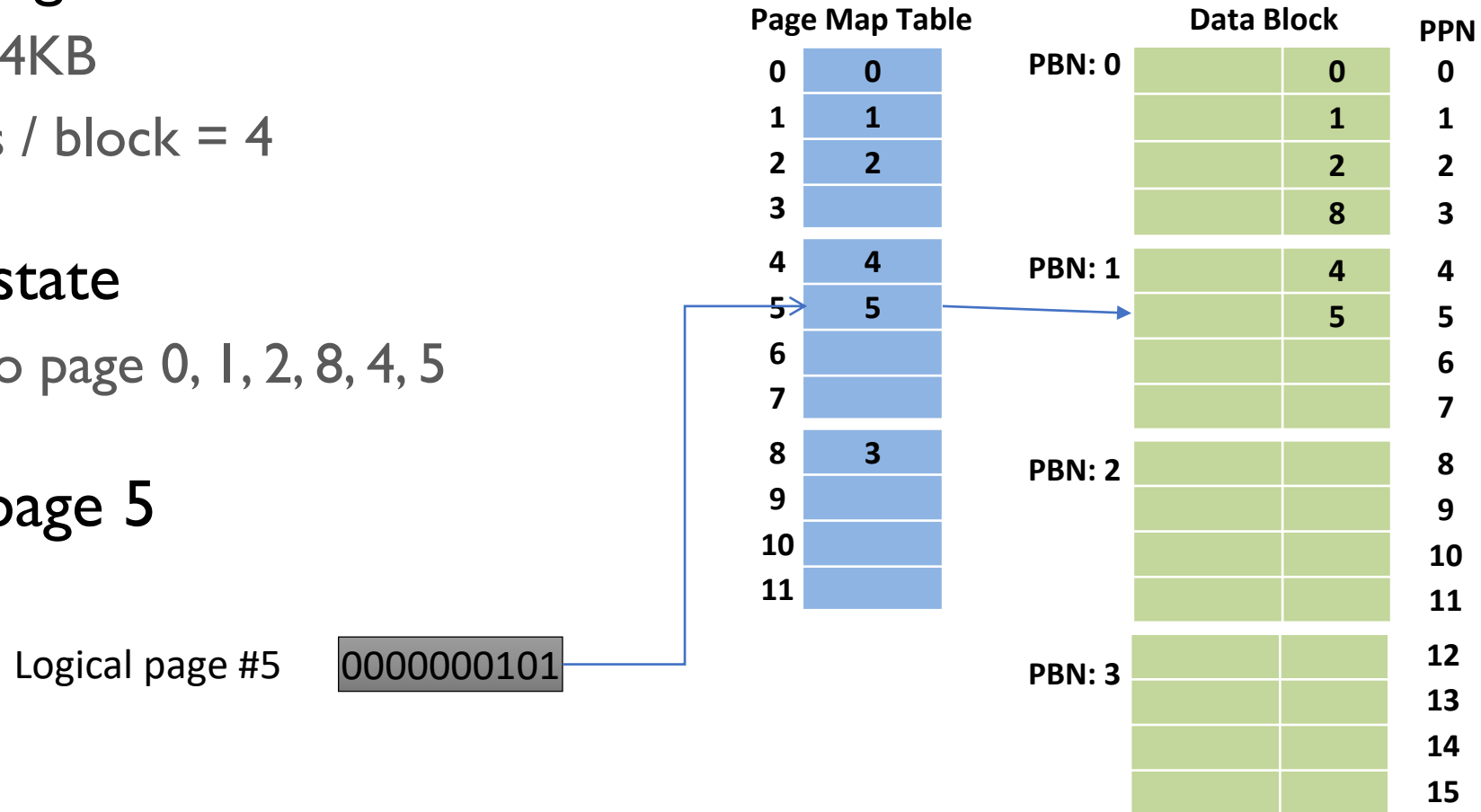
Address Mapping

- Required since flash pages cannot be overwritten



Example: Page Mapping

- Flash configuration
 - Page size: 4KB
 - # of pages / block = 4
- Current state
 - Written to page 0, 1, 2, 8, 4, 5
- Reading page 5



Example: Page Mapping

- Flash configuration
 - Page size: 4KB
 - # of pages / block = 4
- Current state
 - Written to page 0, 1, 2, 8, 4, 5
- New requests (in order)
 - Write to page 9
 - Write to page 3
 - Write to page 5

Page Map Table

0	0
1	1
2	2
3	
4	4
5	5
6	
7	
8	3
9	
10	
11	

Data Block

	Data Block	PPN
PBN: 0	0	0
	1	1
	2	2
	8	3
PBN: 1	4	4
	5	5
		6
		7
PBN: 2		8
		9
		10
		11
PBN: 3		12
		13
		14
		15

Example: Page Mapping

- Flash configuration
 - Page size: 4KB
 - # of pages / block = 4
- Current state
 - Written to page 0, 1, 2, 8, 4, 5
- New requests (in order)
 - Write to page 9
 - Write to page 3
 - Write to page 5

Page Map Table

0	0
1	1
2	2
3	
4	4
5	5
6	
7	
8	3
9	6
10	
11	

	Data Block	PPN
PBN: 0	0	0
	1	1
	2	2
	8	3
PBN: 1	4	4
	5	5
	9	6
		7
PBN: 2		8
		9
		10
		11
PBN: 3		12
		13
		14
		15

Example: Page Mapping

- Flash configuration
 - Page size: 4KB
 - # of pages / block = 4
- Current state
 - Written to page 0, 1, 2, 8, 4, 5
- New requests (in order)
 - Write to page 9
 - Write to page 3**
 - Write to page 5

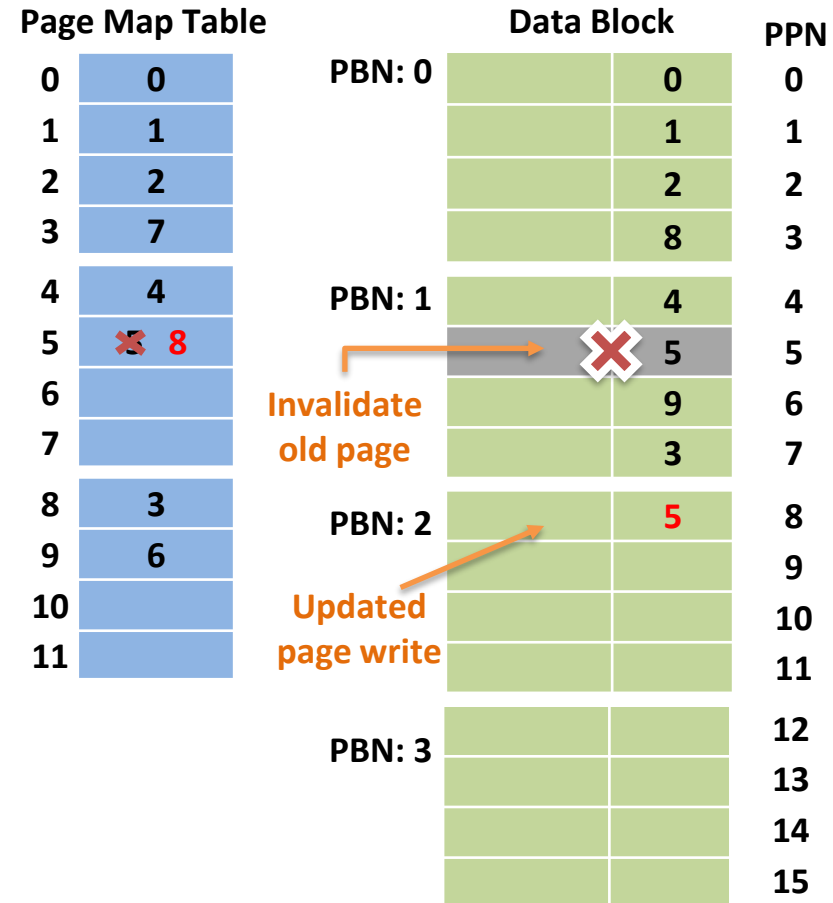
Page Map Table

0	0
1	1
2	2
3	7
4	4
5	5
6	
7	
8	3
9	6
10	
11	

	Data Block	PPN
PBN: 0	0	0
	1	1
	2	2
	8	3
PBN: 1	4	4
	5	5
	9	6
	3	7
PBN: 2		8
		9
		10
		11
PBN: 3		12
		13
		14
		15

Example: Page Mapping

- Flash configuration
 - Page size: 4KB
 - # of pages / block = 4
- Current state
 - Written to page 0, 1, 2, 8, 4, 5
- New requests (in order)
 - Write to page 9
 - Write to page 3
 - Write to page 5**



Garbage Collection

- **Garbage collection (GC)**
 - Eventually, FTL will run out of blocks to write to
 - GC must be performed to reclaim free space
 - Actual GC procedure depends on the mapping scheme

- **GC in page-mapping FTL**
 - Select victim block(s)
 - Copy all valid pages of victim block(s) to free block
 - Erase victim block(s)
 - Note: At least one free block should be reserved for GC

Example: GC in Page Mapping

■ Current state

- Written to page 0, 1, 2, 8, 4, 5
- Written to page 9, 3, 5

■ New requests (in order)

- Write to page 8
- Write to page 9
- Write to page 3
- Write to page 1
- Write to page 4

Page Map Table		Data Block		PPN
0	0	PBN: 0	0	0
1	1		1	1
2	2		2	2
3	7		8	3
4	4	PBN: 1	4	4
5	8		5	5
6			9	6
7			3	7
8	3	PBN: 2	5	8
9	6			9
10				10
11				11
		PBN: 3		12
				13
			Spare block	14
				15

Example: GC in Page Mapping

■ Current state

- Written to page 0, 1, 2, 8, 4, 5
- Written to page 9, 3, 5

■ New requests (in order)

- Write to page 8
- Write to page 9
- Write to page 3
- Write to page 1
- Write to page 4

Page Map Table		Data Block		PPN
0	0	PBN: 0	0	0
1	1		1	1
2	2		2	2
3	7		8	3
4	4	PBN: 1	4	4
5	8		5	5
6			9	6
7			3	7
8	9	PBN: 2	5	8
9	6		8	9
10				10
11				11
		PBN: 3		12
				13
			Spare block	14
				15

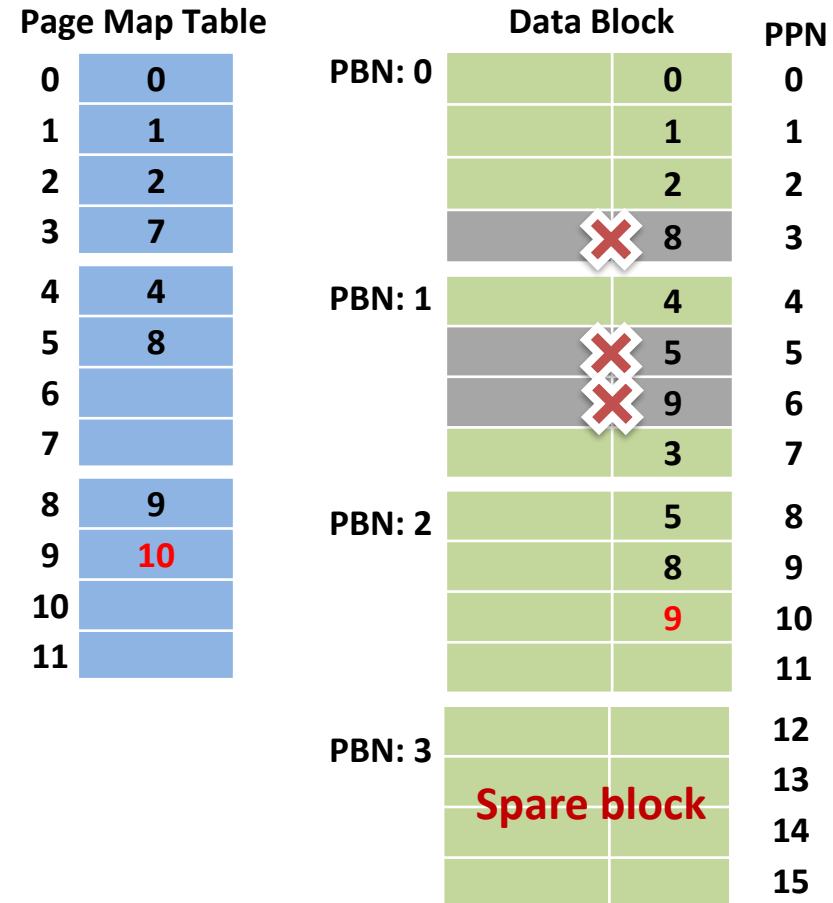
Example: GC in Page Mapping

■ Current state

- Written to page 0, 1, 2, 8, 4, 5
- Written to page 9, 3, 5

■ New requests (in order)

- Write to page 8
- **Write to page 9**
- Write to page 3
- Write to page 1
- Write to page 4



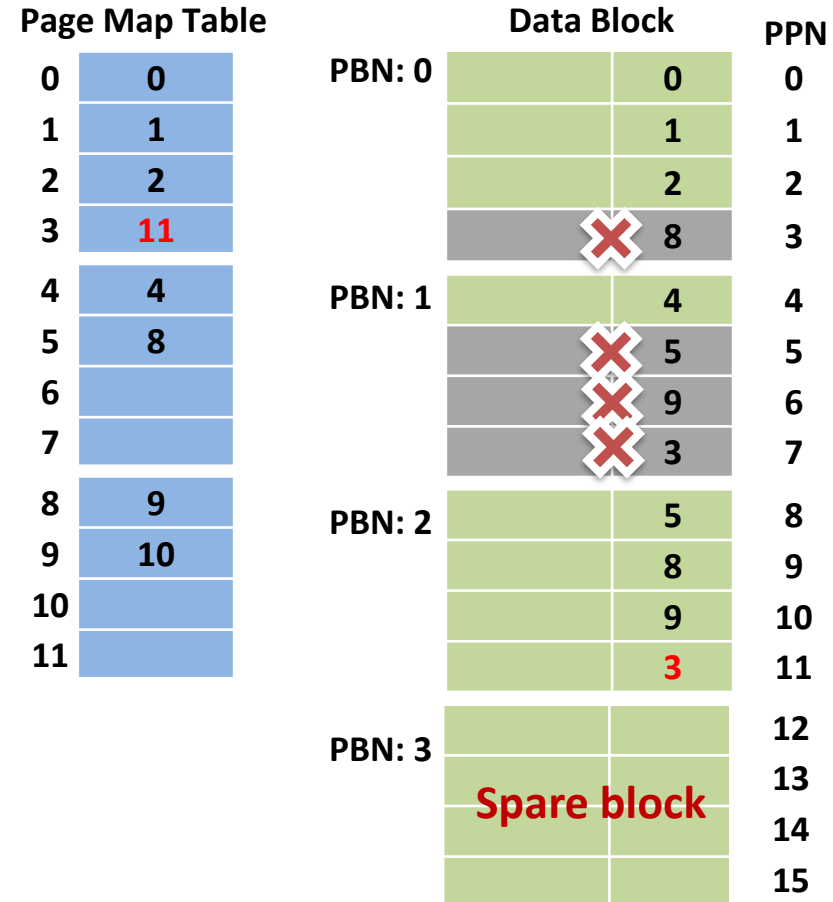
Example: GC in Page Mapping

■ Current state

- Written to page 0, 1, 2, 8, 4, 5
- Written to page 9, 3, 5

■ New requests (in order)

- Write to page 8
- Write to page 9
- **Write to page 3**
- Write to page 1
- Write to page 4



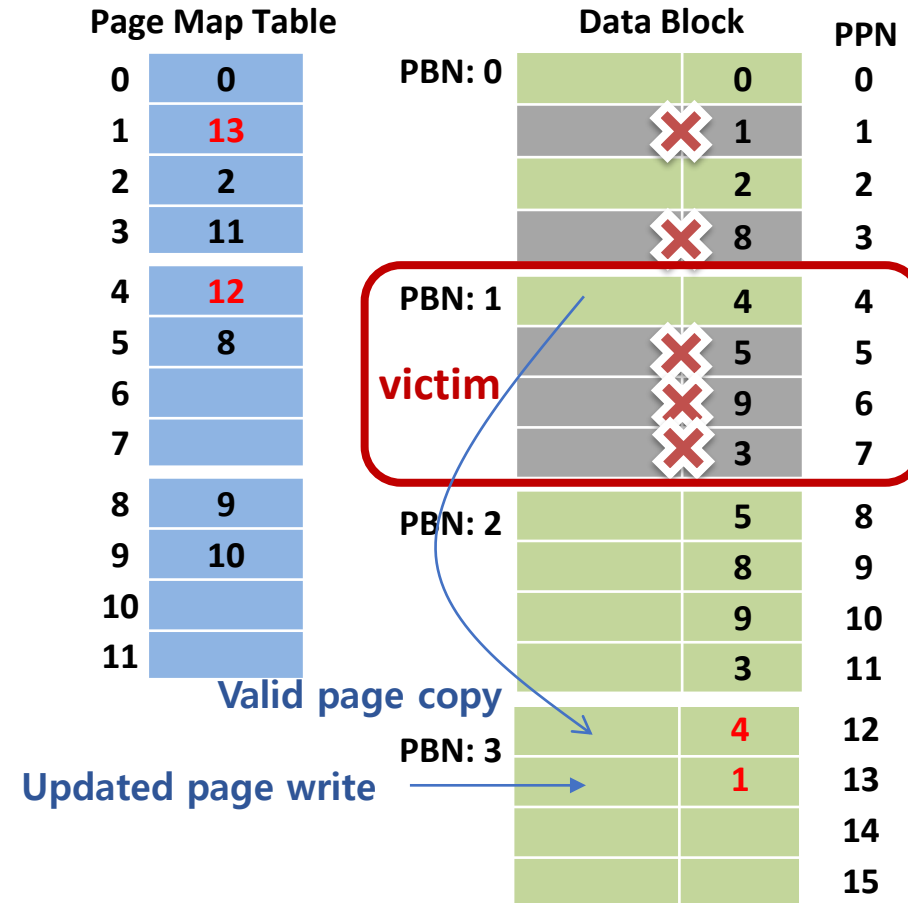
Example: GC in Page Mapping

■ Current state

- Written to page 0, 1, 2, 8, 4, 5
- Written to page 9, 3, 5

■ New requests (in order)

- Write to page 8
- Write to page 9
- Write to page 3
- **Write to page 1**
- Write to page 4



Example: GC in Page Mapping

■ Current state

- Written to page 0, 1, 2, 8, 4, 5
- Written to page 9, 3, 5

■ New requests (in order)

- Write to page 8
- Write to page 9
- Write to page 3
- Write to page 1
- **Write to page 4**

Page Map Table

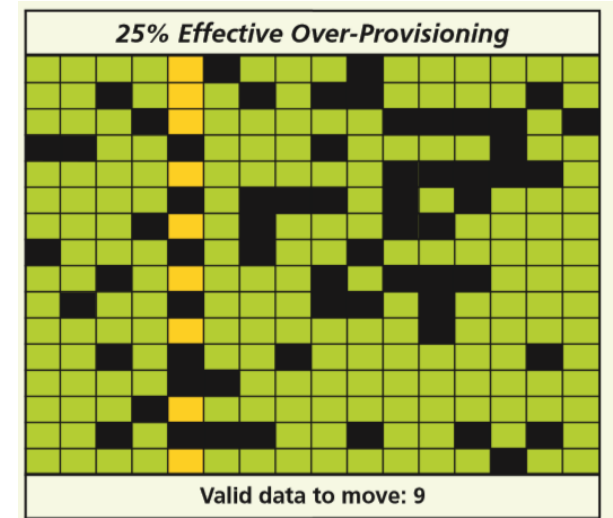
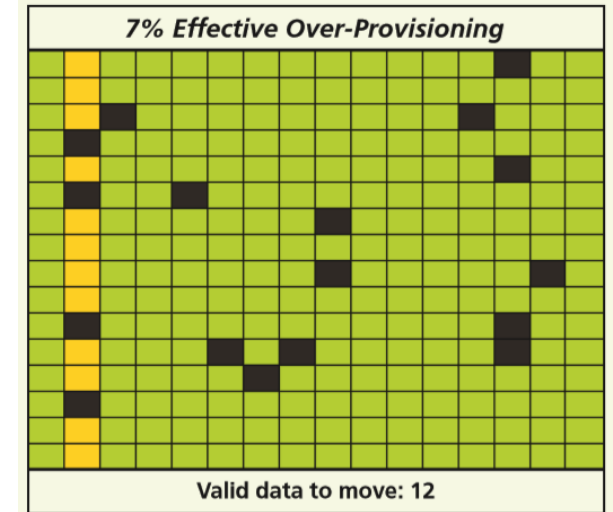
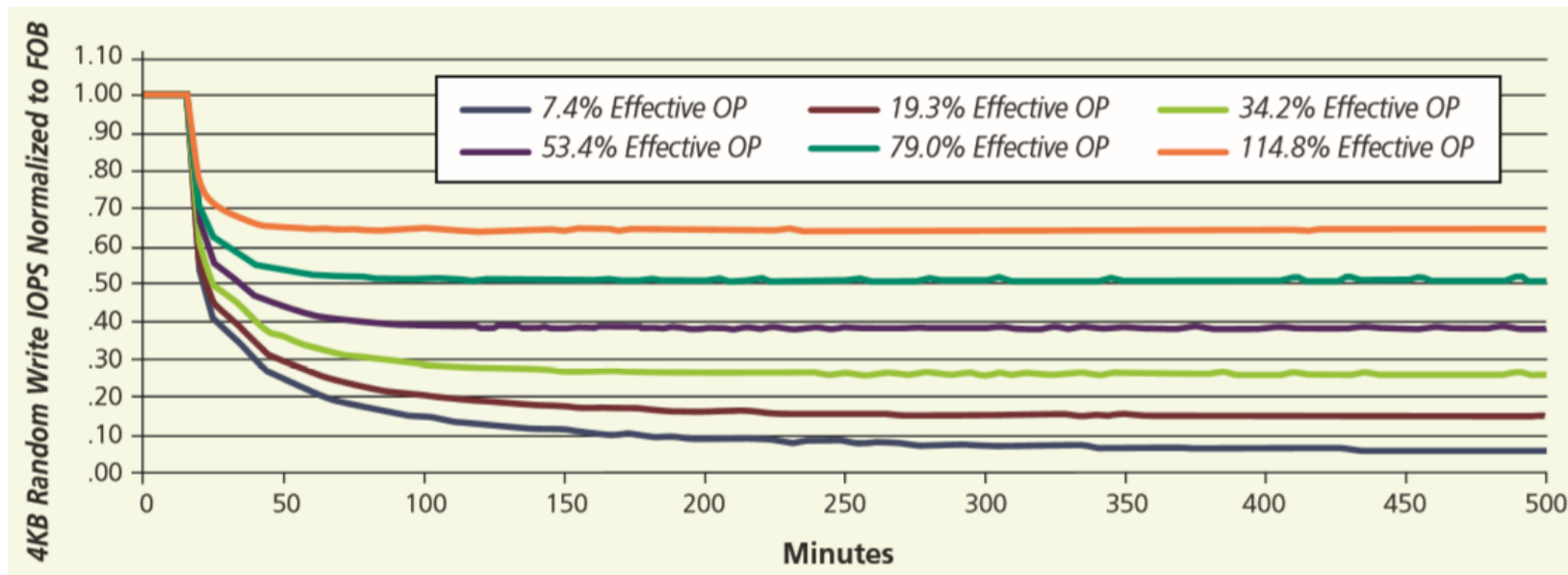
0	0
1	13
2	2
3	11
4	14
5	8
6	
7	
8	9
9	10
10	
11	

	Data Block	PPN
PBN: 0	0	0
	1	1
	2	2
	8	3
PBN: 1		4
	Spare block	5
		6
		7
PBN: 2	5	8
	8	9
	9	10
	3	11
PBN: 3	4	12
	1	13
	4	14
		15

Over-Provisioning and GC

- IOPS for random write workloads

- OP (Over-Provisioning) = $\frac{\text{Physical Capacity}}{\text{Logical Capacity}} - 1$
- What about for sequential write workloads?



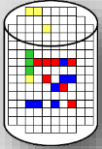

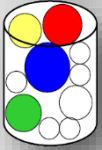
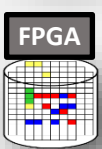

OS Implications

- NAND flash has different characteristics compared to disks
 - No seek time
 - Asymmetric read/write access times
 - No in-place-update
 - Good sequential read/write and random read performance, but bad random write performance
 - Wear-leveling
 - ...
- Traditional operating systems have been optimized for disks. What should be changed?

SSD Support in OS & Applications

- Align file system partition with SSD layout
- Larger block size (4KB)
- Turn off “defragmentation” for SSDs
- New “TRIM” command (remove-on-delete)
- Simpler & scalable I/O scheduler
- Flash-aware file systems (e.g., F2FS in Linux)
- New “multi-stream” interface
- User-level storage access (e.g., SPDK)
- Fairness, isolation, etc.

New NVMe SSD Proposals

		Interfaces	Data placement	Flash management	overwrite	Data movement	Host CPU utilization	Hardware complexity	Computation capability	Software modification
Conventional SSD		Block (SATA, SAS, NVMe)	Host	Device	Yes	-	-	-	No	-
ZNS SSD FDP SSD		Restricted Block (NVMe)	ZNS: Host FDP: Host + Device	ZNS: Host FDP: Host + Device	ZNS: No FDP: Yes	High	High	Low	No	High
KV-SSD		Key-value (NVMe)	Device	Device	Yes	Low	Low	High	Some	High
SmartSSD		FPGA + Block (NVMe)	Host	Device	Yes	Low	Low	High	Yes	Very High
Computational Storage		Compute + Block? (NVMe)	Host or device?	Device	Yes	Low	Low	High	Yes	Very High

Beauty and the Beast

- **NAND Flash memory is a beauty**
 - Small, light-weight, robust, low-cost, low-power, non-volatile device
- **NAND Flash memory is a beast**
 - No in-place-update
 - Much slower program/erase operations
 - Erase unit $>$ read/write unit
 - Bit errors
 - Limited lifetime etc.
- **Software support is essential for performance and reliability!**

