P&S Modern SSDs

Understanding and Designing Modern NAND Flash-Based Solid-State Drives

Dr. Jisung Park
Prof. Onur Mutlu
ETH Zürich

Spring 2021

5 May 2021

Today's Agenda

- Progress Review
- SSD Performance & Advanced NAND Flash Commands

Progress Review

- Refactoring MQSim
  - Push your modifications into the repository when ready
- Progress update
  - Marc: Host interface layer, Hong Chul: NAND flash model
- Any Questions?

P&S Modern SSDs

Meeting 7: SSD Performance & Advanced NAND Flash Commands

Dr. Jisung Park
Prof. Onur Mutlu
ETH Zürich
Spring 2021
28 April 2021

Recap: SSD & NAND Flash Memory

- SSD organization
  - SSD controller: Multicore CPU + per-channel flash controllers
  - DRAM: Metadata store, 0.1% of SSD capacity
  - NAND flash chips
    - Channel – Die (Chip) – Plane – Block – Page
- NAND flash characteristics
  - Erase-before-write, asymmetry in operation units (read/write: page, erase: block), limited endurance, retention loss…
- Basic NAND flash opeartions
  - Read/program/erase

SSD Performance

- Latency
  - The time delay until the request is returned
  - Average read latency (4 KiB): 67 us
  - Average write latency (4 KiB): 47 us

HDD: 5~8 ms

- Throughput
  - The number of requests that can be serviced per unit time
    - IOPS: Input/output Operations Per Second
  - Random read throughput: up to 500K IOPS
  - Random write throughput: up to 480K IOPS

HDD: > 1K IOPS

- Bandwidth
  - The amount of data that can be accessed per unit time
  - Sequential read bandwidth: up to 3,500 MB/s
  - Sequential write bandwidth: up to 3,000 MB/s

HDD: ~100 MB/s

Source: https://www.anandtech.com/show/16504/the-samsung-ssd-980-500gb-1tb-review

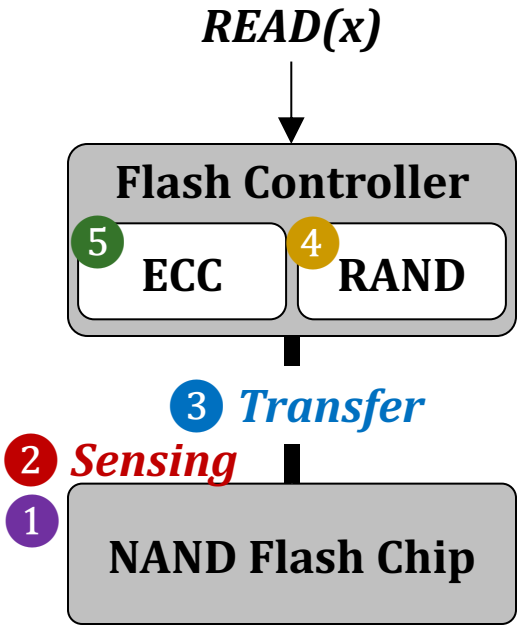
NAND Flash Chip Performance

- Chip operation latency
  - tR: Latency of reading data from the cells into the on-chip page buffer
  - tPROG: Latency of programming the cells with data in the page buffer
  - tBERS: Latency of erasing the cells (block)
  - Varies depending on the MLC technology, processing node, and microarchitecture
    - In 3D TLC NAND flash, tR/tPROG/tBERS ≈ 100us/700us/3ms
- I/O rate
  - Number of bits transferred via a single I/O pin per unit time
  - A typical flash chip transfers data in a byte granularity (i.e., via 8 I/O pins)
  - e.g., 1-Gb I/O rate & 16-KiB page size → tDMA = 16 us

NAND Flash Chip Performance (Cont.)

- tR, tPROG, and tBERS
  - Latencies for chip-level read/program/erase operations
  - tR: 50~100 us
  - tPROG: 700us~1000 us
  - tBERS: 3ms~5ms

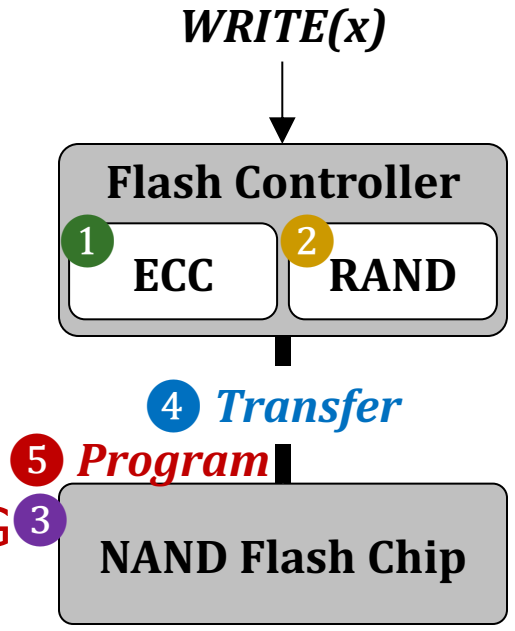
- Flash-controller level latency
  - 1-Gb I/O rate and 16-KiB page size
  - Read
    - (tCMD) + tR + tDMA + (tRND) + $tECC_{DEC}$
    - e.g., 100 + 16 + 20 = 136 us



NAND Flash Chip Performance (Cont.)

- tR, tPROG, and tBERS
  - Latencies for chip-level read/program/erase operations
  - tR: 50~100 us
  - tPROG: 700us~1000 us
  - tBERS: 3ms~5ms

- Flash-controller level latency
  - 1-Gb I/O rate and 16-KiB page size
  - Read
    - (tCMD) + tR + tDMA + (tRND) + $tECC_{DEC}$
    - e.g., 100 + 16 + 20 = 136 us
  - Program
    - $tECC_{ENC}$ + (tRND) + (tCMD) + tDMA + tPROG
    - e.g., 20 + 16 + 700 = 736 us



NAND Flash Chip Performance (Cont.)

- How about bandwidth?
  - Read
    - 16 KiB / 136 us ≈ 120 MB/s
  - Write
    - 16 KiB / 736 us ≈ 22 MB/s

WAIT!

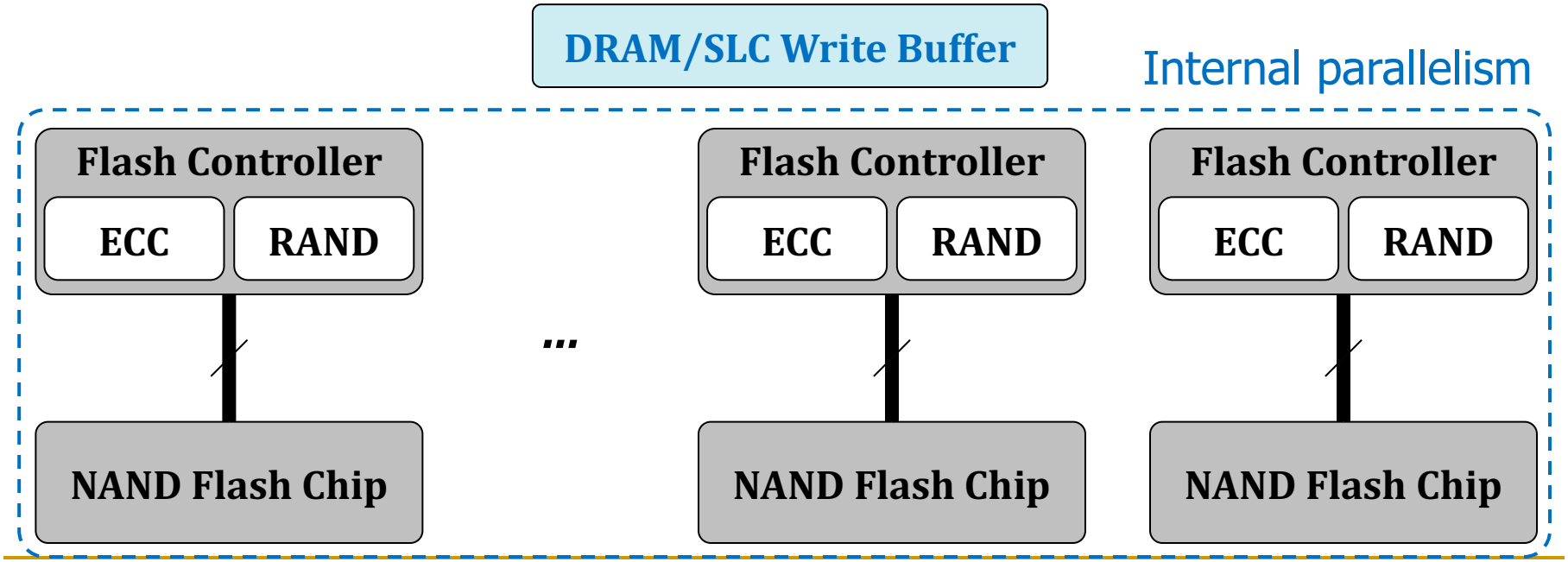
SSD read latency: 67 us

SSD read bandwidth: 3.5 GB/s

SSD write latency: 47 us

SSD write bandwidth: 3 GB/s

Optimizations w/ advanced commands



Advanced Commands for Small Reads

- Minimum I/O units in modern file systems: 4 KiB
  - Latency & bandwidth waste due to I/O-unit mismatch
  - e.g., A page read unnecessarily reads/transfers 12-KiB data

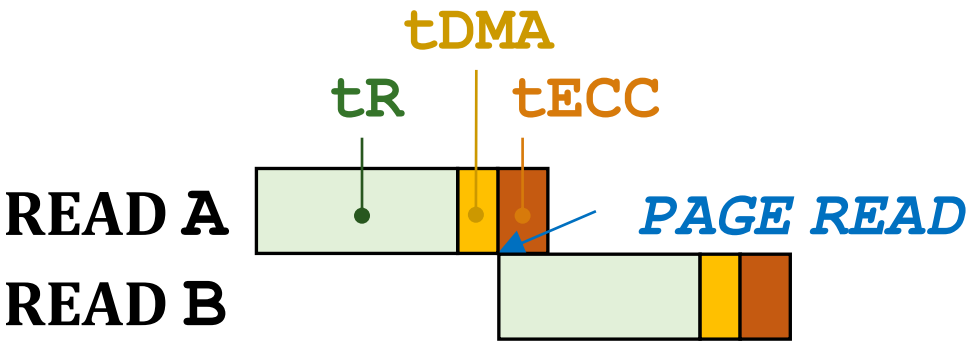
- Optimization 1: Sub-page sensing
  - e.g., Micron SNAP READ operation[1]
  - Microarchitecure-level optimization – directly reduces tR

- Optimization 2: Random Data Out (RDO)
  - Data transfer with an arbitrary offset and size
  - Reduce tDMA and $tECC_{DEC}$

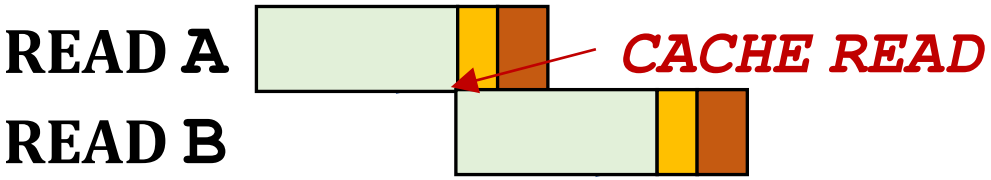
[1]https://media-www.micron.com/-/media/client/global/documents/products/technical-note/nand-flash/tn_2993_snap_read.pdf

CACHE READ Command

- Performs consecutive reads in a pipelined manner



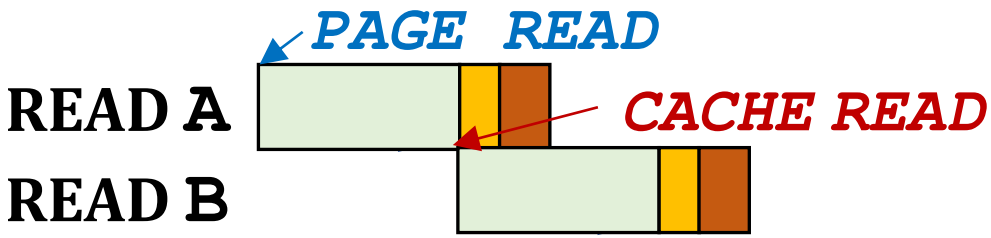
Regular PAGE READ:
Overlaps only tECC with tR

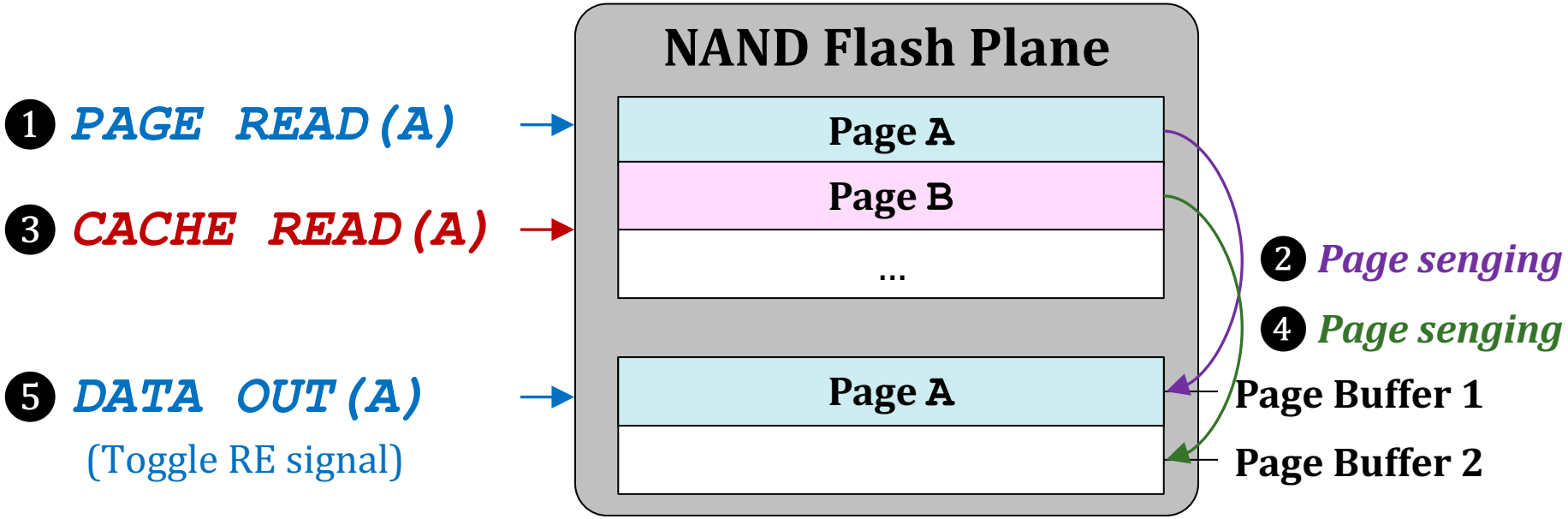


CACHE READ:
Overlaps tDMA & tECC with tR

Enabling the CACHE READ Command

- Needs additional on-chip page buffer



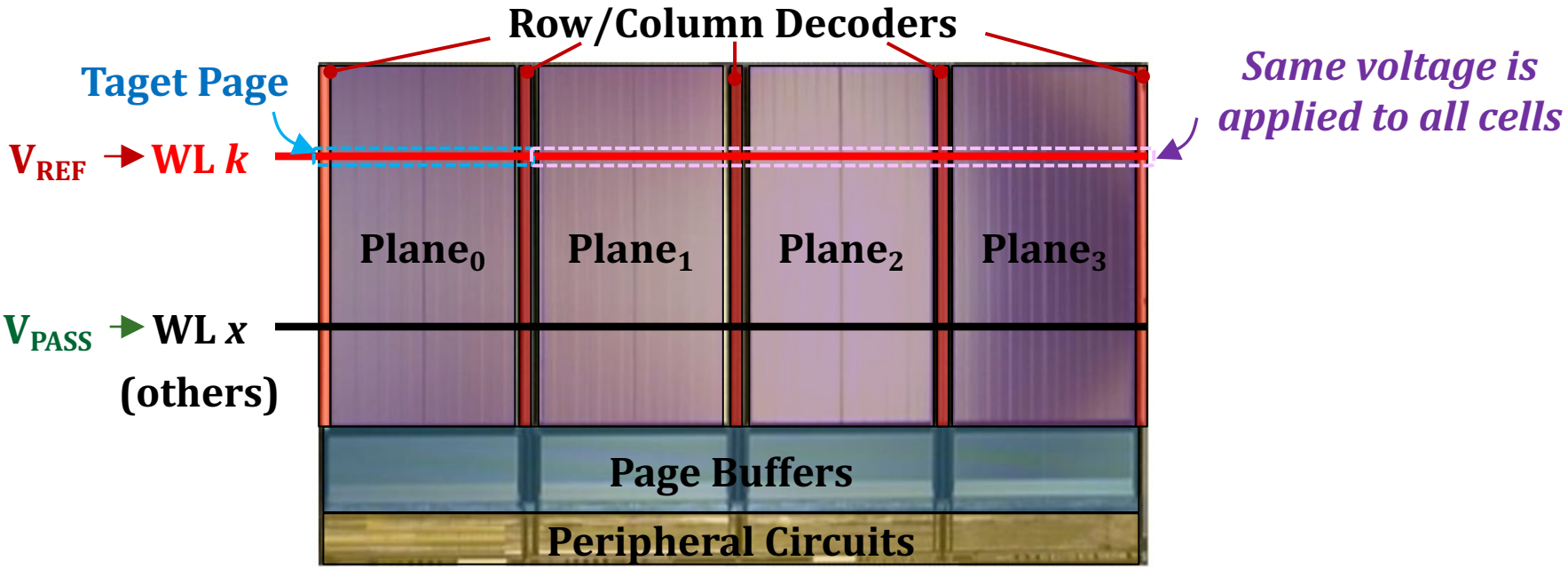


CACHE READ Command: Benefit

- Removes tDMA from the critical path
  - Increases throughput/bandwidth
  - Reduces effective latency
    - By reducing the time delay for a request being blocked by the previous request

Multi-Plane Operations

- Concurrent operations on different planes
  - Recall: Planes share WLs and row/column decoders



- Opportunity: Planes can concurrently operate
- Constraints: Only for the same operations on the same page offset

Multi-Plane Operations: Benefit

- Increase the throughput/bandwidth linearly with # of planes that concurrently operate
  - Bandwidth with regular page programs:
    16 KiB / 736 us ≈ 22 MB/s
  - Bandwidth with multi-plane page programs (2 plane):
    32 KiB / 736 us ≈ 44 MB/s

- Per-operation latency increases
  - Regular page program: $tECC_{ENC}$ + tDMA + tPROG
  - Multi-plane page program: 2 × ($tECC_{ENC}$ + tDMA) + tPROG

- The benefits highly depend on the access pattern and FTL's data placement
  - Random-read-dominant vs. Random-write-dominant

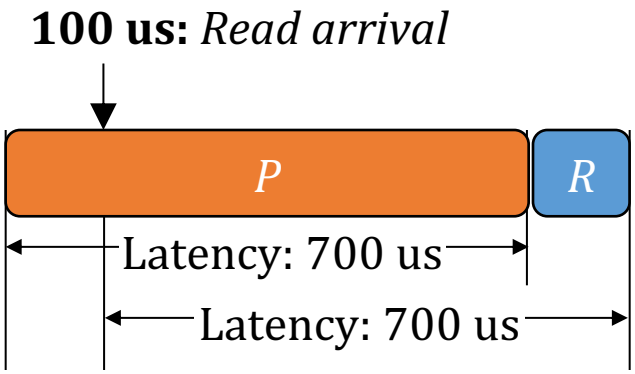
Program & Erase Suspensions

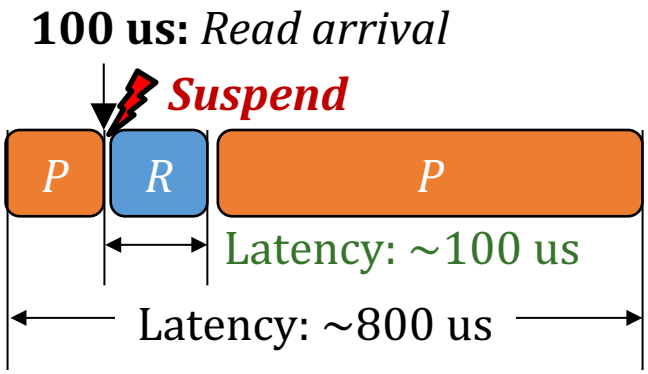
- Read performance is often more important
  - Writes can be done in an asynchronous manner using buffers
    - e.g., return a write request immediately after receiving the data (and storing it to the write buffer)
  - A read request can be returned only when the requested data is ready (after reading the data from the chip)

- Significant latency asymmetry
  - tR: 100 us, tPROG: 700 us, tBERS: 5 ms (TLC NAND flash)
    - If the chip is designed to program all the pages in the same WL at once, the actual program latency is 2,100 us
  - The worst-case chip-level read latency can be 50x longer than the best-case latency

Program & Erase Suspensions (Cont.)

- Suspends an on-going program (erase) operation once a read arrives





  - Pros: Significantly decreases the read latency
  - Cons
    - Additional page buffers (for data to program)
    - Complicated I/O scheduling (Until when can we suspend on-going program requests?)
    - Negative impact on the endurance

Recommend Materials

- Cache read & Read-retry
  - Jisung Park et al., "Reducing Solid-State Drive Read Latency by Optimizing Read-Retry," In ASPLOS 2021.

- Program & Erase Suspension
  - Guanying Wu et al., "Reducing SSD Read Latency via NAND Flash Program and Erase Suspension," In USENIX FAST 2012.
  - Shine Kim et al., "Practical Erase Suspension for Modern Low-latency SSDs," In USENIX ATC 2019.

Next Meetings

- We will provide more background on host request handling
- We will discuss your progress in last week
  - Please contact us whenever you have any questions

P&S Modern SSDs

Understanding and Designing Modern NAND Flash-Based Solid-State Drives

Dr. Jisung Park
Prof. Onur Mutlu
ETH Zürich

Spring 2021

5 May 2021