Today’s Agenda

- SSD Organization & Request Handling
- NAND Flash Organization
- NAND Flash Operations
Recap: Modern SSD Architecture

- A modern SSD is a complicated system that consists of multiple cores, HW controllers, DRAM, and NAND flash memory chips

![SSD Diagram]

### Samsung PM853T 960GB Enterprise SSD

- **SSD Controller**
  - Core
  - Core
  - Core
  - HW Flash Ctrl.
  - Request Handler
  - ECC/Randomizer
  - Encryption Engine

- **LPDDR DRAM**
  - $0.001 \times 1,024 = 1 \text{ GB}$

- **NAND Packages**
  - $8 \times 128 \text{ GB} = 1 \text{ TB}$

[Source: Samsung PM853T 960GB Enterprise SSD (from https://www.tweaktown.com/reviews/6695/samsung-pm853t-960gb-enterprise-ssd-review/index.html)]
Another Overview

- **Host Interface Layer (HIL)**
- **Flash Translation Layer (FTL)**
  - Data Cache Management
  - Address Translation
  - GG/WL/Refresh/…
- **Flash Controller**
  - ECC
  - Randomizer
- **DRAM**
  - Host Request Queue
  - Write Buffer
  - Logical-to-Physical Mappings
  - Metadata (e.g., P/E Cycles)

Diagram showing the components and connections of a storage system.
Request Handling: Write

- Communication with the host operating system (receives & returns requests)
  - Via a certain interface (SATA or NVMe)

- A host I/O request includes
  - Request direction (read or write)
  - Offset (start sector address)
  - Size (number of sectors)
  - Typically aligned by 4 KiB
Request Handling: Write

- Buffering data to write (read from NAND flash memory)
  - Essential to reducing write latency
  - Enables flexible I/O scheduling
  - Helpful for improving lifetime (not so likely)
- Limited size (e.g., tens of MBs)
  - Needs to ensure data integrity even under sudden power-off
  - Most DRAM capacity is used for L2P mappings
Request Handling: Write

- Host Interface Layer (HIL)
- Flash Translation Layer (FTL)
  - Data Cache Management
  - Address Translation
  - GG/WL/Refresh/
- Flash Controller
  - ECC
  - Randomizer
- NAND Flash Package
- NAND Flash Package
- NAND Flash Package
- CTRL
- CTRL
- DRAM
  - Host Request Queue
  - Write Buffer
  - Logical-to-Physical Mappings
  - Metadata (e.g., P/E Cycles)

- Core functionality for out-of-place writes
  - To hide the erase-before-write property
- Needs to maintain L2P mappings
  - Logical Page Address (LPA) → Physical Page Address (PPA)
- Mapping granularity: 4 KiB
  - 4 Bytes for 4 KiB → 0.1% of SSD capacity
Request Handling: Write

Host Interface Layer (HIL)

Flash Translation Layer (FTL)
- Data Cache Management
- Address Translation
- GG/WL/Refresh/...

Flash Controller
- ECC
- Randomizer
- NAND Flash Package
- NAND Flash Package
- NAND Flash Package

DRAM
- Host Request Queue
- Write Buffer
- Logical-to-Physical Mappings
- Metadata (e.g., P/E Cycles)

- Garbage collection (GC)
  - Reclaims free pages
  - Selects a victim block → copies all valid pages → erase the victim block

- Wear-leveling (WL)
  - Evenly distributes P/E cycles across NAND flash blocks
  - Hot/cold swapping

- Data refresh
  - Refresh pages with long retention ages
Request Handling: Write

- Randomizer
  - Scrambling data to write
  - To avoid worst-case data patterns that can lead to significant errors

- Error-correcting codes (ECC)
  - Can detect/correct errors: e.g., 72 bits/1 KiB error-correction capability
  - Stores additional parity information together with raw data

- Issue NAND flash commands
Host Interface Layer (HIL)

Flash Translation Layer (FTL)
- Data Cache Management
- Address Translation
- GG/WL/Refresh/...

Flash Controller
- ECC
- Randomizer

DRAM
- Host Request Queue
- Write Buffer
- Logical-to-Physical Mappings
- Metadata (e.g., P/E Cycles)

- First checks if the request data exists in the write buffer
  - If so, returns the corresponding request immediately with the data

- A host read request can be involved with several pages
  - Such a request can be returned only after all the requested data is ready
Request Handling: Read

- Finds the PPA where the request data is stored from the L2P mapping table
Request Handling: Read

- First reads the raw data from the flash chip
- Performs ECC decoding
- Derandomizes the raw data
- ECC decoding can fail
  - Retries reading of the page w/ adjusted $V_{\text{REF}}$
  - Soft-decision ECC (LDPC)
Today’s Agenda

- SSD Organization & Request Handling
- NAND Flash Organization
- NAND Flash Operation
A Flash Cell

- Basically, it is a transistor

![Diagram of a Flash Cell with labels for Source (S), Substrate, Drain (D), Control Gate (G), Threshold Voltage (V_{TH}), Drain Current (I_D), and Gate Voltage (V_{GS})]
A Flash Cell

- Basically, it is a transistor
  - w/ a special material: Floating gate (2D) or Charge trap (3D)
A Flash Cell

- Basically, it is a transistor
  - w/ a special material: Floating gate (2D) or Charge trap (3D)
  - Can hold electrons in a non-volatile manner

\[ V_{PGM} = 20 \, V \]

- \( G \) (Control Gate)
- \( FG \) (Floating Gate)
- \( S \) (Source)
- \( GND \) (Substrate)
- \( D \) (Drain)

- Tunneling

\[ V_{TH} \]

\[ V_{GS} \]
A Flash Cell

- Basically, it is a transistor
  - w/ a special material: Floating gate (2D) or Charge trap (3D)
  - Can hold electrons in a non-volatile manner
  - Changes the cell’s threshold voltage ($V_{\text{TH}}$)

![Flash Cell Diagram]

- **GND**
- **G** (Control Gate)
- **FG** (Floating Gate)
- **S** (Source) 20 V Substrate
- **D** (Drain) T
- **Tunneling**

---

- $V_{\text{TH}} < V_{\text{REF}}$
- $V_{\text{TH}} < V_{\text{REF}}$
- ‘0’
- ‘1’

---

$V_{\text{GS}}$
Flash Cell Characteristics

- **Multi-leveling:** A cell can store multiple bits

  ![Program: Inject electrons](image)

  ![Erase: Eject electrons](image)

- **Retention loss:** A cell leaks electrons over time

  

  ![Retention error!](image)

- **Limited lifetime:** A cell wears out after P/E cycling

  

  ![Retention error!](image)
A NAND String

- Multiple (e.g., 128) flash cells are serially connected
Pages and Blocks

- A large number (> 100,000) of cells operate concurrently

Page = 16 + $\alpha$ KiB

Wordline

BL$^0$  BL$^1$  BL$^2$  BL$^3$  BL$^{132,095}$

Block = {(# of WL) × (# of bits per cell)} pages
Pages and Blocks (Continued)

- Program and erase: Unidirectional
  - Programming a cell → *Increasing* the cell’s $V_{TH}$
  - Eraseing a cell → *Decreasing* the cell’s $V_{TH}$

- Programming a page cannot change ‘0’ cells to ‘1’ cells
  → *Erase-before-write* property

- Erase unit: Block
  - Increase erase bandwidth
  - Makes in-place write on a page very inefficient
  → Out-of-place write & GC
A large number (> 1,000) of blocks share bitlines in a plane.
Planes

- A large number (> 1,000) of blocks share bitlines in a plane.
Planes and Dies

- A die (or chip) contains multiple (e.g., 2 – 4) planes

![Diagram of a 21-nm 2D NAND Flash Die]

- Planes share decoders: limits internal parallelism (only operations @ the same WL offset)
Today’s Agenda

- SSD Organization & Request Handling
- NAND Flash Organization
- NAND Flash Operation
Threshold Voltage Distribution

- $V_{TH}$ distribution of **cells** in a **programmed page/block/chip**

- **Why** distribution? **Variations** across the cells
  - Some cells are more easily programmed or erased

- **Why** (almost) the same shape?
  - Every data is stored after randomized for better reliability
  - In reality, $V_{TH}$ states’ shapes can be different, but there areas are almost the same

There are $y$ cells whose $V_{TH} = xV$
**V_{TH} Distribution of MLC NAND Flash**

- **Multi-level cell (MLC) technique**
  - $2^m V_{TH}$ states required to store $m$ bits in a single flash cell

- **Limited width** of the $V_{TH}$ window: Need to
  - Make each $V_{TH}$ state narrow
  - Guarantee sufficient margins b/w adjacent $V_{TH}$ states
V_{TH} Distribution of MLC NAND Flash

- Multi-level cell (MLC) technique
  - $2^m V_{TH}$ states required to store $m$ bits in a single flash cell

- Limited width of the $V_{TH}$ window: Need to
  - Make each $V_{TH}$ state narrow
  - Guarantee sufficient margins b/w adjacent $V_{TH}$ states
    - $V_{TH}$ changes over time after programmed
    - Narrower margins → Lower reliability
    - More bits per cell → higher density but lower reliability

# of cells

- Shifted & widened after programmed

$V_{TH}$ margin

Error cells
Basic Operation: Page Program

- **Block**: The basic unit of memory in the page program.
- **Target Page**: The page to be programmed.
- **String Select Line (SSL)**: Selects the string of memory cells.
- **Wordline (WL)**: Selects the word (row) of memory cells.
  - WL$_k$ and WL$_{k+1}$: Select the current and next wordlines, respectively.
  - WL$_{k-1}$: Select the previous wordline.
- **Ground Select Line (GSL)**: Used to ground the selected memory cells.
Basic Operation: Page Program

- WL control – All other cells operate as a resistance

![Diagram of page program operation]

- String Select Line (SSL)
- Wordline (WL)
- Ground Select Line (GSL)

- $V_{CC}$
- $V_{PASS}$
- $V_{PROG}$
Basic Operation: Page Program

- BL control – Inhibits cells to not be programmed

![Diagram showing BL control for page program with program and inhibit states.]
Basic Operation: Page Program

- BL control – **Inhibits cells** to not be programmed

![Diagram showing basic operation of page program](image)

- **V\textsubscript{PROG}** and **WL\textsubscript{k}**
- **BL\textsubscript{0}** (program)
- **BL\textsubscript{1}** (inhibit)
- **BL\textsubscript{2}**
- **BL\textsubscript{3}**
- **BL\textsubscript{132,095}**
Basic Operation: Page Program

program

inhibit

$V_{PROG}$, $WL_k$

To $GND$

To $V_{CC}$

To $GND$

To $V_{CC}$

To $GND$

# of cells

Erased (E)

Threshold voltage ($V_{TH}$)

$V_{REF}$
Basic Operation: Page Program

**V_{PROG}** WL_k

- **program**
  - BL_0
    - 0
    - To GND
- **inhibit**
  - BL_1
    - 1
    - To V_{cc}
- BL_2
  - 0
  - To GND
- BL_3
  - 1
  - To V_{cc}
- BL_132,095
  - 0
  - To GND

---

Inhibited cells

Erased (E)

Programmed cells

Threshold voltage (V_{TH})

**V_{REF}**
Basic Operation: Page Program

**V**<sub>PROG</sub> **WL<sub>k</sub>**

- **program**
- **inhibit**

# of cells

**Threshold voltage (V<sub>TH</sub>)**

- **Inhibited cells**
- **Erased (E)**

- **V**<sub>REF</sub>

- **Cells to program**

- **Hard-to-program cells**
- **Easy-to-program cells**
Basic Operation: Page Program

- Incremental Step-Pulse Programming (ISPP)

![Diagram of ISPP process]

- Program
- Inhibit

# of cells

Threshold voltage ($V_{TH}$)

- Inhibited cells
- Cells to program

Erased (E)

Verified as programmed

Program inhibit

To GND

To $V_{CC}$

To GND

To $V_{CC}$

To GND

$V_{PROGO}$ $WL_k$
Basic Operation: Page Program

- Incremental Step-Pulse Programming (ISPP)

![Diagram of ISPP process](image)
Basic Operation: Page Program

- Incremental Step-Pulse Programming (ISPP)

![Diagram of ISPP]

- Incremental Step-Pulse Programming (ISPP)

- Program

- Inhibit

- $V_{\text{PROG}}$

- $W_{L_k}$

- BL$_0$

- BL$_1$

- BL$_2$

- BL$_3$

- BL$_{132,095}$

- To GND

- To $V_{CC}$

- Inhibit programmed cells

- Inhibited cells

- Erased (E)

- Cells to program

- Threshold voltage ($V_{TH}$)

- # of cells

- $V_{\text{REF}}$
Basic Operation: Page Program

- Incremental Step-Pulse Programming (ISPP)

V\text{PROG}_1 \text{ WL}_k \quad \text{BL}_0 \quad \text{BL}_1 \quad \text{BL}_2 \quad \text{BL}_3 \quad \text{BL}_{132,095}

\text{program} \quad 0 \quad \text{inhibit} \quad 1 \quad 0 \quad 1 \quad 0

To GND \quad \text{To } V\text{CC} \quad \text{To } V\text{CC} \quad \text{To } V\text{CC} \quad \text{To GND}

# of cells

Inhibited cells

Inhibited cells

Threshold voltage (V\text{TH})

Cells to program

Programmed

Erased (E)
Basic Operation: Page Read

- **WL control** – All other cells operate as a resistance

![Diagram showing WL control and threshold voltage (V_{TH})](image)
Basic Operation: Page Read

- **BL control – Charge all BLs**

![BL control diagram](image)

- **# of cells**
  - **Erased (E)**
  - **Programmed**

- **Threshold voltage (V_{TH})**
  - **V_{REF}**
  - **1** Erased (E)
  - **0** Programmed
Basic Operation: Page Read

- Sensing the current through BLs

\[ V_{\text{REF}} \]  \( W_{L_k} \)

\[ \text{BL}_0 \quad 0 \]  \( \rightarrow \)  \[ \text{BL}_1 \quad 1 \]  \( \rightarrow \)  \[ \text{BL}_2 \quad 0 \]  \( \rightarrow \)  \[ \text{BL}_3 \quad 1 \]  \( \rightarrow \)  \[ \text{BL}_{132,095} \quad 0 \]

(No current) (Current)

\[ V_{TH} < V_{REF} \]

\[ V_{REF} \]

\[ V_{TH} < V_{REF} \]

1 Erased (E)

0 Programmed

Threshold voltage (\( V_{TH} \))

# of cells
Basic Operation: Page Read - MLC

- Sensing the current through BLs

\[ \text{WL}_k \]

\[ \text{BL}_0 \rightarrow \text{BL}_1 \rightarrow \text{BL}_2 \rightarrow \text{BL}_3 \rightarrow \text{BL}_{132,095} \]

# of cells

\[ \text{V}_{\text{REF}0} \quad \text{V}_{\text{REF}1} \quad \text{V}_{\text{REF}2} \quad \text{V}_{\text{REF}3} \quad \text{V}_{\text{REF}4} \quad \text{V}_{\text{REF}5} \quad \text{V}_{\text{REF}6} \]

\[ \text{CSB} \quad \text{LSB} \quad \text{MSB} \]

\[ \text{E} \quad \text{P1} \quad \text{P2} \quad \text{P3} \quad \text{P4} \quad \text{P5} \quad \text{P6} \quad \text{P7} \]

\[ \text{V}_{\text{TH}} \]
Basic Operation: Page Read - MLC

- Sensing the current through BLs

![Diagram of sensing the current through BLs]

- # of cells

- V_{TH}

- V_{REF0} to V_{REF6}

- MSB, LSB, CSB

- WL_k
Basic Operation: Page Read - MLC

- Sensing the current through BLs

![Diagram showing the sensing of current through BLs with WLk, MSB, LSB, CSB, and VTH.]
Basic Operation: Page Read - MLC

- Sensing the current through BLs

![Diagram showing the sensing of current through BLs](image)
Basic Operation: Page Read - MLC

- Sensing the current through BLs

![Diagram showing sensing the current through BLs]
Basic Operation: Page Read - MLC

- Sensing the current through BLs

![Diagram showing the sensing of current through BLs and the comparison with reference voltage levels](image)
Basic Operation: Page Read - MLC

- Sensing the current through BLs

![Diagram showing the sensing of current through BLs](image-url)
Basic Operation: Page Read – Takeaways

- MLC NAND flash memory requires an **in-chip XOR logic**
- Bit-encoding affects the read latency!
  - Compare # of sensing for LSB

![Diagram of V_TH vs. # of cells with reference voltages (V_REF0 to V_REF6) and bit patterns (111, 110, 100, 000, 010, 011, 001, 101)]
MLC NAND flash memory requires an in-chip XOR logic

Bit-encoding affects the read latency!
- Compare # of sensing for LSB
Basic Operation: Page Read – Takeaways

- MLC NAND flash memory requires an in-chip XOR logic
- Bit-encoding affects the read latency!
  - Compare # of sensing for LSB
Basic Operation: Page Read – Takeaways

- MLC NAND flash memory requires an *in-chip XOR logic*
- Bit-encoding affects the read latency!
  - Compare # of sensing for LSB

![Diagram showing voltage levels and sensing points](image-url)
Required Material

- Yu Cai, Saugata Ghose, Erich F. Haratsch, Yixin Luo, and Onur Mutlu,
  “Errors in Flash-Memory-Based Solid-State Drives: Analysis, Mitigation, and Recovery,“
  Invited Book Chapter in Inside Solid State Drives, 2018 - Introduction and Section 1

- Jisung Park, Myungsuk Kim, Myoungjun Chun, Lois Orosa, Jihong Kim, and Onur Mutlu,
  “Reducing Solid-State Drive Read Latency by Optimizing Read-Retry,“ In ASPLOS, 2021
Recommended Material

- Arash Tavakkol, Mohammad Sadrosadati, Saugata Ghose, Jeremie Kim, Yixin Luo, Yaohua Wang, Nika Mansouri Ghiasi, Lois Orosa, Juan Gómez Luna, and Onur Mutlu, “FLIN: Enabling Fairness and Enhancing Performance in Modern NVMe Solid State Drives,” In ISCA, 2018

- Bryan S. Kim, Hyun Suk Yang, and Sang Lyul Min, “AutoSSD: an Autonomic SSD Architecture,” In USENIX ATC, 2018
P&S Modern SSDs

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

Dr. Jisung Park
Prof. Onur Mutlu

ETH Zürich
Fall 2021
6 October 2021