P&S Modern SSDs

Basics of NAND Flash-Based SSDs

Dr. Mohammad Sadrosadati
Prof. Onur Mutlu

ETH Zürich
Spring 2023
17 March 2023
Today’s Agenda

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

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

![SSD Architecture Diagram]

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

- **LPDDR DRAM**: $0.001 \times 1,024 = 1$ GB
- **NAND Packages**: $8 \times 128$ GB = 1 TB

Another Overview

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

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

- **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
- GC/WL/Refresh/…

**Flash Controller**
- ECC
- Randomizer

**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
- GC/WL/Refresh/...

Flash Controller
- ECC
- Randomizer
- CTRL

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

Host Interface Layer (HIL)
Flash Translation Layer (FTL)
- Data Cache Management
- Address Translation
- GC/WL/Refresh/...

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

Flash Controller
- ECC
- Randomizer

Flash Packages
- NAND Flash Package
- NAND Flash Package
- NAND Flash Package

- 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
- Issues NAND flash commands
Host Interface Layer (HIL)

Flash Translation Layer (FTL)
- Data Cache Management
- Address Translation
- GC/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 with adjusted $V_{REF}$
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 S (Source), D (Drain), G (Control Gate), and I_D (Current). There is also a graph showing the relationship between I_D and V_GS (Gate Source Voltage) with threshold voltage (V_TH) indicated.]
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

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

![Diagram of a Flash Cell](image)

- Source (S)
- Control Gate (G)
- Floating Gate (FG)
- Substrate (GND)
- Drain (D)
- 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 ($V_{TH}$)

![Diagram of Flash Cell]

- $V_{TH} < V_{REF}$ → ‘1’
- $V_{TH} > V_{REF}$ → ‘0’

Legend:
- FG (Floating Gate)
- G (Control Gate)
- S (Source)
- D (Drain)
- 20 V Substrate
- Tunneling
- $V_{GS}$
Flash Cell Characteristics

- **Multi-leveling**: A flash cell can store multiple bits
  
  ![Program: Inject electrons](image1) ![Erase: Eject electrons](image2)  
  **Flash Cell**

- **Retention loss**: A cell leaks electrons over time
  
  ![Retention error!](image3)  
  ![Retention error!](image4)  
  ![Retention error!](image5)  
  **1 year**  
  **1 year**  
  **1 year**

- **Limited lifetime**: A cell wears out after P/E cycling
  
  ![Retention error!](image6)  
  ![Retention error!](image7)  
  ![Retention error!](image8)  
  **1 year**  
  **1 year**  
  **@ 1K P/E cycles**  
  **@ 10K P/E cycles**
A NAND String

- Multiple (e.g., 128) flash cells are serially connected
A large number (> 100,000) of cells operate concurrently

Page = 16 + α KiB

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

- Program and erase: Unidirectional
  - Programming a cell → Increasing the cell’s \( V_{TH} \)
  - Erasing 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
Planes

- 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 contains multiple (e.g., 2 – 4) planes

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

There are $y$ cells whose $V_{TH} = xV$

- 1 Erased (E)
- 0 Programmed

Threshold voltage ($V_{TH}$)

# of cells
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
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 $\rightarrow$ Lower reliability
    - More bits per cell $\rightarrow$ higher density but lower reliability

- **Shifted & widened after programmed**

- **Error cells**
Basic Operation: Page Program

String Select Line
SSL

Wordline
WL_{k-1}
Target Page
WL_k
WL_{k+1}

Ground Select Line
GSL

Block

BL_0  BL_1  BL_2  BL_3  BL_{132,095}
Basic Operation: Page Program

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

![Diagram showing WL control and various voltage levels](image-url)
Basic Operation: Page Program

- **BL control** – Inhibits cells to not be programmed
Basic Operation: Page Program

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

![Diagram showing the basic operation of page program with BL control](image-url)
Basic Operation: Page Program

To GND  To $V_{cc}$  To GND  To $V_{cc}$  To GND

# of cells

Threshold voltage ($V_{TH}$)

$V_{REF}$

Erased (E)

$V_{PROG}$  $WL_k$

$BL_0$  0  1  $BL_1$  0  1  $BL_2$  0  $BL_3$  0  $BL_{132,095}$

program inhibit
Basic Operation: Page Program

- **Programmed cells**
  - Threshold voltage ($V_{TH}$)
  - Programmed (1)

- **Inhibited cells**
  - Threshold voltage ($V_{TH}$)
  - Erased (E)

- WL$_k$ connected to $V_{POWER}$
  - BL$_0$ connected to GND
  - BL$_1$ connected to $V_{CC}$
  - BL$_2$ connected to GND
  - BL$_3$ connected to $V_{CC}$
  - BL$_{132,095}$ connected to GND

- Number of cells:
  - Erased (E)
  - Programmed (1)
Basic Operation: Page Program

**Program**

- \( V_{PROG} \)
- \( WL_k \)
- \( \text{To GND} \)
- \( \text{To } V_{CC} \)

**Inhibit**

- \( BL_0 \)
- \( BL_1 \)
- \( BL_2 \)
- \( BL_3 \)
- \( BL_{132,095} \)
- \( \text{To GND} \)
- \( \text{To } V_{CC} \)

### # of cells

- Erased (E)
- Inhibited cells
- \( V_{REF} \)
- Cells to program
- Hard-to-program cells
- Easy-to-program cells

### Threshold voltage (\( V_{TH} \))
Basic Operation: Page Program

- Incremental Step-Pulse Programming (ISPP)

![Diagram showing the process of programming and inhibiting cells in a memory array. The diagram includes labels for program, inhibit, To GND, To VCC, and a graph showing the relationship between the number of cells and threshold voltage (V_TH).]
Basic Operation: Page Program

- Incremental Step-Pulse Programming (ISPP)

**Diagram:***

- **WL<sub>k</sub>:** Word Line
- **BL<sub>0</sub>-<sub>3</sub>:** Bit Lines
- **V<sub>cc</sub>:** Supply Voltage
- **GND:** Ground

**Legend:**

- **Program:** BL<sub>0</sub> and BL<sub>1</sub>
- **Inhibit:** BL<sub>1</sub>
- **Inhibit programmed cells:** BL<sub>2</sub> and BL<sub>3</sub>

**Graph:**

- **Threshold voltage (V<sub>TH</sub>):**
- **Inhibited cells:** Cells with a threshold voltage between V<sub>REF</sub> and V<sub>TH</sub>
- **Erased (E):** Cells with a threshold voltage below V<sub>REF</sub>
- **Cells to program:** Cells with a threshold voltage above V<sub>REF</sub>
Basic Operation: Page Program

- Incremental Step-Pulse Programming (ISPP)

Diagram:
- Program 0 to GND
- Inhibit 1 to Vcc
- Program 0 to Vcc
- Program 1 to Vcc
- Program 0 to GND

Inhibited cells

Erased (E)

Cells to program

Threshold voltage (V_{TH})

Inhibited cells

Cells to program

V_{REF}

# of cells
Basic Operation: Page Program

- Incremental Step-Pulse Programming (ISPP)

**Diagram:**
- **V_{PROG1}** and **WL_k**
- **BL_0**, **BL_1**, **BL_2**, **BL_3**, **BL_{132,095}**
- **Program** and **Inhibit**
- **To GND**, **To V_{CC}**
- **Inhibited cells** and **Programmed cells**
- **Threshold voltage (V_{TH})**
- **# of cells**
- **V_{REF}**
- **Erased (E)**
- **Cells to program**
- **Programmed**

**Notes:**
- # of cells
- Erased (E)
- Cells to program
- Programmed
Basic Operation: Page Read

- WL control – All other cells operate as a resistance

![Diagram showing WL control and BL lines with different states](image)

V_{REF} WL_k

---

<table>
<thead>
<tr>
<th># of cells</th>
<th>Erased (E)</th>
<th>Programmed</th>
</tr>
</thead>
</table>

Threshold voltage (V_{TH})
Basic Operation: Page Read

- BL control – Charge all BLs

![Diagram of BL control](image)

- To $V_{cc}$

# of cells

$V_{REF}$

Threshold voltage ($V_{TH}$)

Threshold voltage ($V_{TH}$)

- Erased (E)

- Programmed

- $V_{REF}$
Basic Operation: Page Read

- Sensing the current through BLs

\[ V_{REF} \rightarrow WL_k \rightarrow BL_0 \rightarrow BL_1 \rightarrow BL_2 \rightarrow BL_3 \rightarrow \ldots \rightarrow BL_{132,095} \]

\[ \begin{align*}
&\text{(No current)} \\
&\text{(Current)} \\
&V_{TH} < V_{REF} \\
&V_{TH} > V_{REF}
\end{align*} \]
Basic Operation: Page Read - MLC

- Sensing the current through BLs

![Diagram of MLC sensing through BLs with WLk and various BLs showing different currents and voltages](image-url)
Basic Operation: Page Read - MLC

- Sensing the current through BLs

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

- Sensing the current through BLs

![Diagram showing sensing the current through BLs with WLk, BL0 to BL132,095, and the corresponding voltage levels VREF0 to VREF6.](image)

- MSB
- LSB
- VTH
- CSB
- # of cells

Example:

- WLk
- BL0: 111
- BL1: 000
- BL2: 101
- BL3: 001
- BL132,095: 110

- VREF0
- VREF1
- VREF2
- VREF3
- VREF4
- VREF5
- VREF6

- E
- P1
- P2
- P3
- P4
- P5
- P6
- P7
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 sensing through BLs with specific current values and reference voltages](image-url)
Basic Operation: Page Read - MLC

- Sensing the current through BLs

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

- Sensing the current through BLs

\[ \text{WL}_k \]

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

\[ \begin{array}{c}
\text{V}_{\text{REF}0} \\
\text{V}_{\text{REF}1} \\
\text{V}_{\text{REF}2} \\
\text{V}_{\text{REF}3} \\
\text{V}_{\text{REF}4} \\
\text{V}_{\text{REF}5} \\
\end{array} \]

\[ \text{CSB} \quad \text{XOR} \]

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

\[ \begin{array}{ccccccccc}
0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
\end{array} \]
Basic Operation: Page Read – Takeaways

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

![Diagram of V_{TH} vs. number of cells showing different reference voltages and binary values](image-url)
Basic Operation: Page Read – Takeaways

- MLC NAND flash memory requires an on-chip XOR logic
- Bit-encoding affects the read latency!
  - Compare # of sensing for LSB
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
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
P&S Modern SSDs

Basics of NAND Flash-Based SSDs

Dr. Mohammad Sadrosadati
Prof. Onur Mutlu

ETH Zürich
Spring 2023
17 March 2023