P&S Modern SSDs

Basics of NAND Flash-Based SSDs

Dr. Jisung Park
Prof. Onur Mutlu
ETH Zürich
Spring 2022
25 March 2021
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 Controller**
- Core
- Core
- Core

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

**LPDDR DRAM**

**NAND Packages**

\[0.001 \times 1,024 = 1 \text{ GB}\]

\[8 \times 128 \text{ GB} = 1 \text{ TB}\]

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)

NAND Flash Package

NAND Flash Package

NAND Flash Package
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
- GG/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

- 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

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

- **Flash Controller**
  - ECC
  - Randomizer

- **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**
Request Handling: Read

- **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 with adjusted $V_{REF}$
  - Soft-decision ECC (e.g., 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](image)

- **S** (Source)
- **D** (Drain)
- **G** (Control Gate)
- **I_D**
- **V_{TH} < V_{GS}**
- **V_{TH} < V_{GS}**

**Graph:**

- **I_D** vs **V_{GS}**
- **V_{TH}** (Threshold Voltage)
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)
- \( V_{TH} \)
- \( V_{GS} \)
- Tunneling
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 ($V_{TH}$)

\[ G \text{ (Control Gate)} \]

\[ S \text{ (Source)} \]

\[ D \text{ (Drain)} \]

\[ 20 \text{ V} \]

\[ \text{Floating Gate} \]

\[ \text{Tunneling} \]

\[ \text{Substrate} \]

\[ V \text{GS} \]

\[ I_D \]

\[ V_{TH} \]

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

\[ V_{TH}' \]

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

\[ V_{REF} \]
Flash Cell Characteristics

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

  ![Flash Cell Diagram]

  **Program**: Inject electrons
  **Erase**: Eject electrons
  

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

  ![Retention Loss Diagram]

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

  ![Lifetime Diagram]

Retention error! @ 1K P/E cycles
Retention error! @ 10K P/E cycles
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 + 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

Page = $16 + \alpha K_iB$
Block = {(# of WL) × (# of bits per cell)} pages
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 (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
**V\text{TH} Distribution of MLC NAND Flash**

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

- **Limited width of the V\text{TH} window**: Need to
  - Make each $V_{\text{TH}}$ state narrow
  - Guarantee sufficient margins b/w adjacent $V_{\text{TH}}$ 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 $\rightarrow$ Lower reliability
- More bits per cell $\rightarrow$ higher density but lower reliability

---

![Graph](image-url)
Basic Operation: Page Program

String Select Line

Wordline $WL_{k-1}$

Target Page

Wordline $WL_k$

Wordline $WL_{k+1}$

Ground Select Line

Block
Basic Operation: Page Program

- WL control – All other cells operate as a resistance

![Diagram showing the operation of the Page Program with WL control and voltage levels V_CC, V_PASS, V_PROG, SSL (String Select Line), and GSL (Ground Select Line).]
Basic Operation: Page Program

- BL control – Inhibits cells to not be programmed

![Diagram showing BL control for program and inhibit]

- $V_{PROG}$
- $WL_k$
- BL0: 0
- BL1: 1
- BL2: 0
- BL3: 1
- BL132,095: 0
Basic Operation: Page Program

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

![Diagram showing BL control](image)

- **V\text{PROG}**
- **WL\_k**
- **BL\_0**
- **BL\_1**
- **BL\_2**
- **BL\_3**
- **BL\_{132,095}**
- **0**
- **1**
- **To GND**
- **To V\text{CC}**

Program status:
- 0
- 1

Inhibit status:
- 0
- 1
Basic Operation: Page Program

The diagram illustrates the process of programming and inhibiting cells in memory. The voltage $V_{PROG}$ is applied to the word line $WL_k$, and the bit lines $BL_0$ to $BL_{132,095}$ are set to 0 or 1:

- **Program ($\text{program}$)**: The bit lines are set to 0 to program the cell.
- **Inhibit ($\text{inhibit}$)**: The bit lines are set to 1 to inhibit the cell.

The threshold voltage $V_{TH}$ is determined by the number of cells in the erased state (E) with $V_{REF}$.

$V_{PROG}$ is used to program the cells to a state that is detectable, and $V_{CC}$ and GND are used to set the bit lines.

The number of cells and their threshold voltage $V_{TH}$ are plotted on the graph, showing the relationship between the number of cells and the threshold voltage.
Basic Operation: Page Program

V_{PROG} \quad WL_k

program

BL_0 \quad inhibit

BL_1

BL_2

BL_3

BL_{132,095}

# of cells

Threshold voltage (V_{TH})

Inhibited cells

Programmed cells

1

Erased (E)

0

Programmed

V_{REF}
Basic Operation: Page Program

$V_{PROG}$ $WL_k$

- **Program** $BL_0$
- **Inhibit** $BL_1$
- $BL_2$
- $BL_3$

- **To GND**
- **To $V_{CC}$**

**# of cells**

$V_{REF}$

**Threshold voltage ($V_{TH}$)**

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

**Cells to program**

**Hard-to-program cells**

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

- Incremental Step-Pulse Programming (ISPP)

![Diagram of ISPP](attachment://diagram.png)

- **Program**:
  - 0 on BL0
  - 1 on BL1

- **Inhibit**:
  - 0 on BL2
  - 1 on BL3

- **V_{PROG0}**
  - WL_k
  - To GND
  - To V_{CC}
  - To GND

- **Verifying**:
  - Cells verified as programmed

- **Threshold voltage (V_{TH})**
  - Erased (E)
  - Inhibited cells
  - Cells to program

- **V_{REF}**

- **# of cells**

- **Notes**:
  - BL0 to BL132,095
Basic Operation: Page Program

- Incremental Step-Pulse Programming (ISPP)

![Diagram showing incremental step-pulse programming (ISPP)]

- Program
- Inhibit

- WL_k
- BL_0
- BL_1
- BL_2
- BL_3
- BL_132,095

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

# of cells

Inhibited cells
Erased (E)
Cells to program

V_{REF}

Threshold voltage (V_{TH})

Inhibited programmed cells

Number of cells
Basic Operation: Page Program

- Incremental Step-Pulse Programming (ISPP)

![Diagram showing the process of programming and inhibiting cells.]

- **V<sub>PROG</sub>**
- **WL<sub>k</sub>**
- **BL<sub>0</sub>**
- **BL<sub>1</sub>**
- **BL<sub>2</sub>**
- **BL<sub>3</sub>**
- **BL<sub>132,095</sub>**

- **V<sub>REF</sub>**
- **Threshold voltage (V<sub>TH</sub>)**
- **Inhibited cells**
- **Erased (E)**
- **Cells to program**

- **To GND**
- **To V<sub>CC</sub>**

- **Inhibit programmed cells**

In the diagram, the process of programming and inhibiting cells is depicted. The voltage levels and the number of cells are highlighted to illustrate the ISPP process.
Basic Operation: Page Program

- Incremental Step-Pulse Programming (ISPP)

V_{PROG1} \quad \text{WL}_k \quad \text{BL}_0 \quad \text{BL}_1 \quad \text{BL}_2 \quad \text{BL}_3 \quad \text{BL}_{132,095}

- Program
- Inhibit

To GND \quad \text{To V}_{cc} \quad \text{To V}_{cc} \quad \text{To V}_{cc} \quad \text{To GND}

# of cells

Erased (E) \quad \text{Inhibited cells} \quad \text{Cells to program} \quad \text{Programmed}

Threshold voltage (V_{TH})

V_{REF}
Basic Operation: Page Read

- WL control – All other cells operate as a resistance

\[ V_{\text{REF}} \]  
\[ \text{WL}_k \]

\[ \begin{array}{c}
\text{BL}_0 \\
0 \\
\downarrow \\
\text{BL}_1 \\
1 \\
\downarrow \\
\text{BL}_2 \\
0 \\
\downarrow \\
\text{BL}_3 \\
1 \\
\downarrow \\
\text{BL}_{132,095} \\
0 \\
\downarrow \\
\end{array} \]

\( T_{\text{TH}} \)  

\[ \begin{array}{c}
\# \text{of cells} \\
\end{array} \]

\[ \begin{array}{c}
\text{Erased (E)} \\
1 \\
\end{array} \]

\[ \begin{array}{c}
\text{Programmed} \\
0 \\
\end{array} \]

\[ V_{\text{REF}} \]
Basic Operation: Page Read

- **BL control** – Charge all BLs

![Diagram showing BL control](image)

- **VREF**
- **WLk**
- **To Vcc**

<table>
<thead>
<tr>
<th>BL</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>BL0</td>
<td>0</td>
</tr>
<tr>
<td>BL1</td>
<td>1</td>
</tr>
<tr>
<td>BL2</td>
<td>0</td>
</tr>
<tr>
<td>BL3</td>
<td>1</td>
</tr>
<tr>
<td>BL132,095</td>
<td>0</td>
</tr>
</tbody>
</table>

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

- **Threshold voltage (V_TH)**
- **VREF**

- **1 Erased (E)**
- **0 Programmed**
Basic Operation: Page Read

- Sensing the current through BLs

![Diagram showing sensing the current through BLs with symbols for 0 (No current) and 1 (Current).]

- V_{REF}\ WL_k

- \# of cells:
  - Erased (E)
  - Programmed

- Threshold voltage (V_{TH}): V_{TH} < V_{REF}
Basic Operation: Page Read - MLC

- Sensing the current through BLs

![Diagram showing the sensing of current through BLs with MSB, LSB, V_REF0, V_REF1, V_REF2, V_REF3, V_REF4, V_REF5, V_REF6, and WLk.]
Basic Operation: Page Read - MLC

- Sensing the current through BLs

![Diagram showing sensing through BLs with WLk, BL0 to BL132,095, and voltage levels V_REF0 to V_REF6.](image-url)
Basic Operation: Page Read - MLC

- Sensing the current through BLs

![Diagram showing cell sensing through BLs with WL and VTH]

- Diagram illustrating cell sensing through BLs with WL and VTH

- Diagram showing cell sensing through BLs with WL and VTH

- Diagram showing cell sensing through BLs with WL and VTH

- Diagram showing cell sensing through BLs with WL and VTH
Basic Operation: Page Read - MLC

- Sensing the current through BLs

![Diagram of sensing current through BLs]

- The diagram shows a word line (WL_k) connected to bit lines (BL0 to BL132,095).
  - Each bit line represents a cell with its own voltage levels for sensing.
  - The MSB and LSB are indicated with arrows, and the voltage levels V_{REF0} to V_{REF6} are shown.

# of cells

- The figure illustrates how the voltage levels are used to distinguish between different states of the cells.

V_{TH}
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

```
<table>
<thead>
<tr>
<th>BL0</th>
<th>BL1</th>
<th>BL2</th>
<th>BL3</th>
<th>BL132,095</th>
</tr>
</thead>
<tbody>
<tr>
<td>111</td>
<td>000</td>
<td>101</td>
<td>001</td>
<td>110</td>
</tr>
</tbody>
</table>
```

- Voltages:
  - \( V_{TH} < V_{REF} \) for '1'
  - \( V_{TH} < V_{REF} \) for '0'

- VREF values:
  - \( V_{REF0} \)
  - \( V_{REF1} \)
  - \( V_{REF2} \)
  - \( V_{REF3} \)
  - \( V_{REF4} \)
  - \( V_{REF5} \)
  - \( V_{REF6} \)
Basic Operation: Page Read - MLC

- Sensing the current through BLs

![Diagram showing basic operation of page read in MLC with sensing through BLs.](Image)
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 showing voltage levels and bit encoding for MLC NAND flash memory](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
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
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

Basics of NAND Flash-Based SSDs

Dr. Jisung Park
Prof. Onur Mutlu

ETH Zürich
Spring 2022
25 March 2021