

P&S Modern SSDs

Basics of NAND Flash-Based SSDs

Dr. Jisung Park

Prof. Onur Mutlu

ETH Zürich

Spring 2022

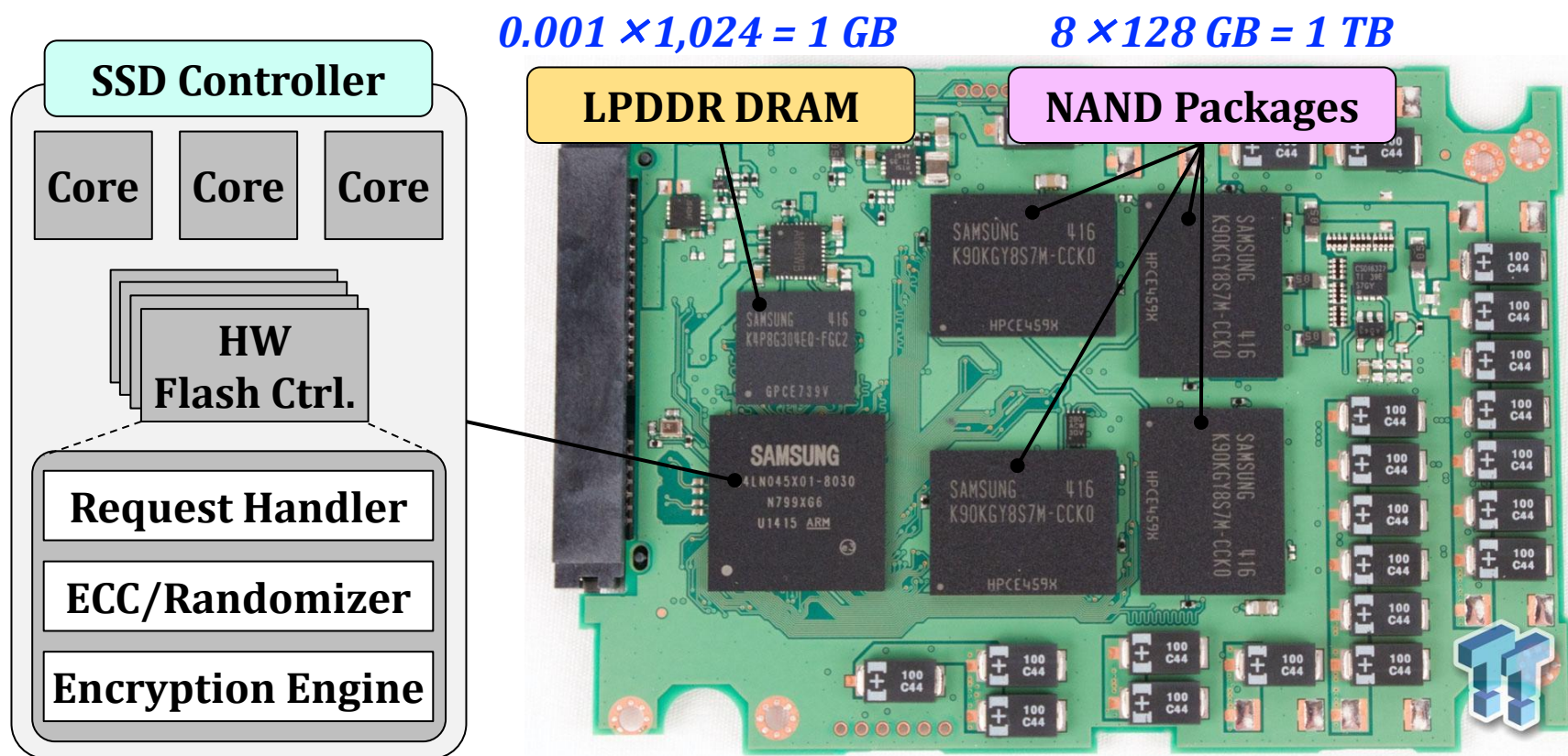
25 March 2021

Today's Agenda

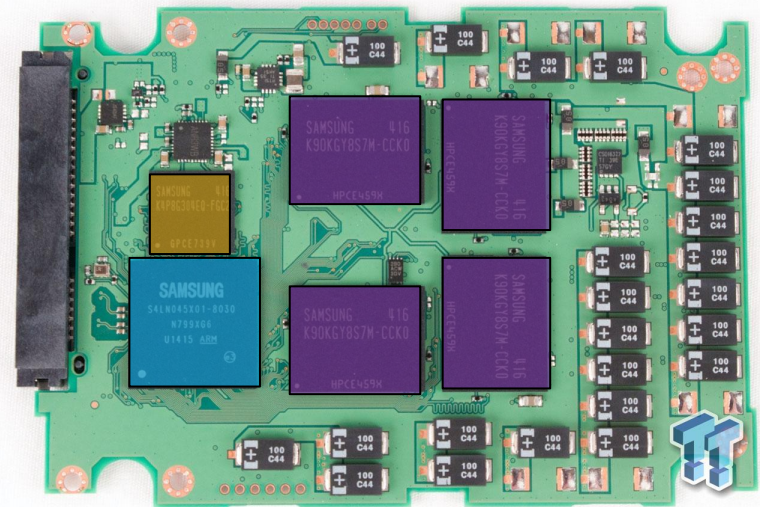
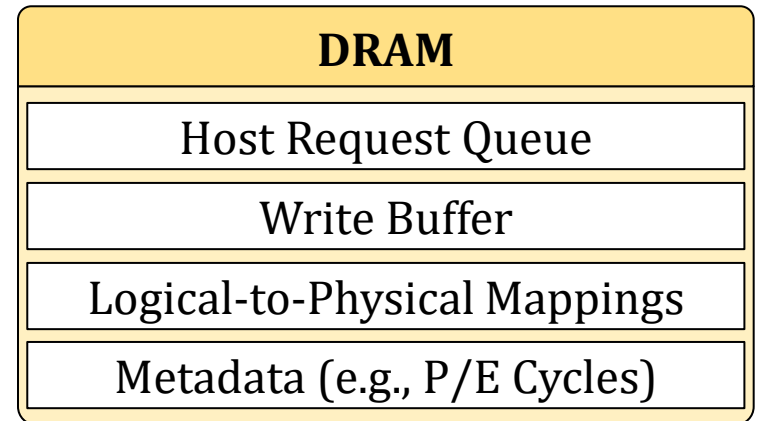
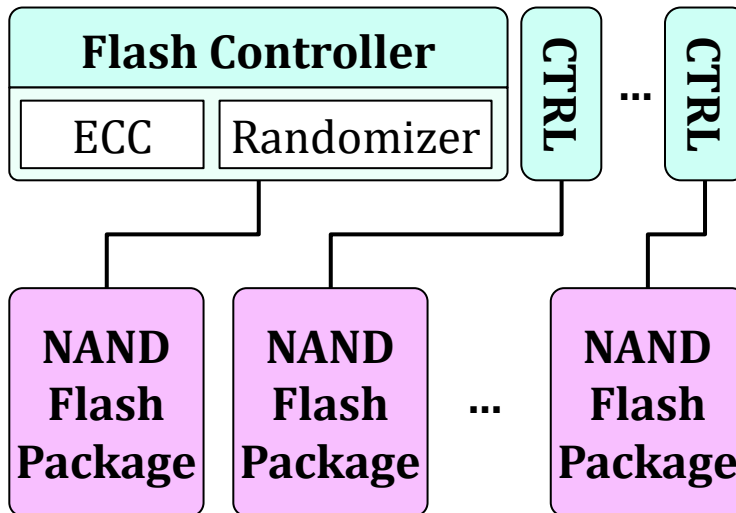
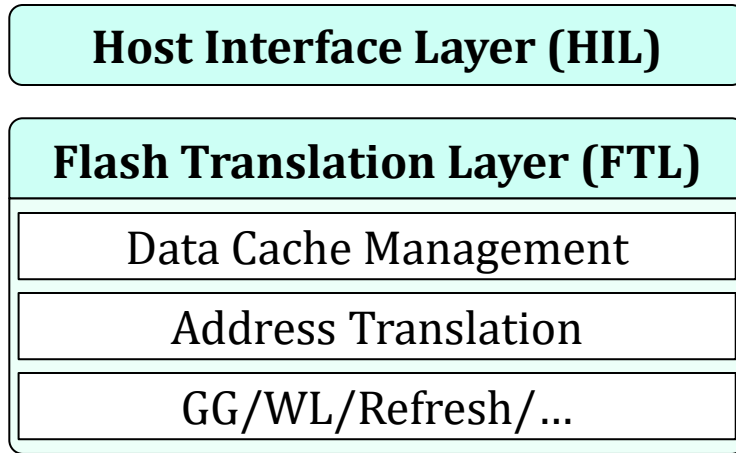
- SSD Organization & Request Handling
- NAND Flash Organization
- NAND Flash Operations

Modern SSD Architecture

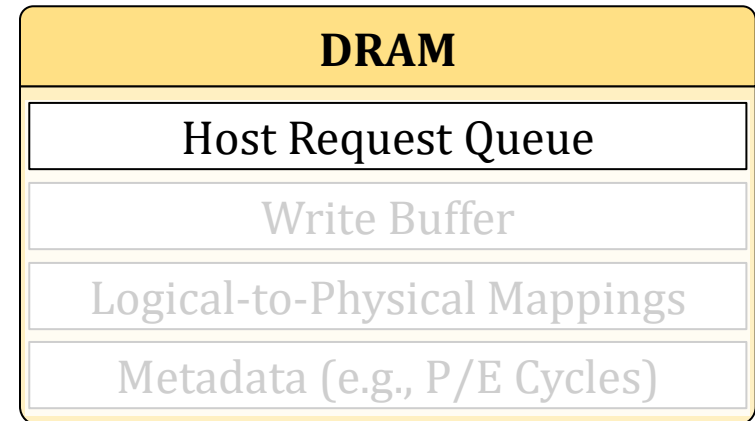
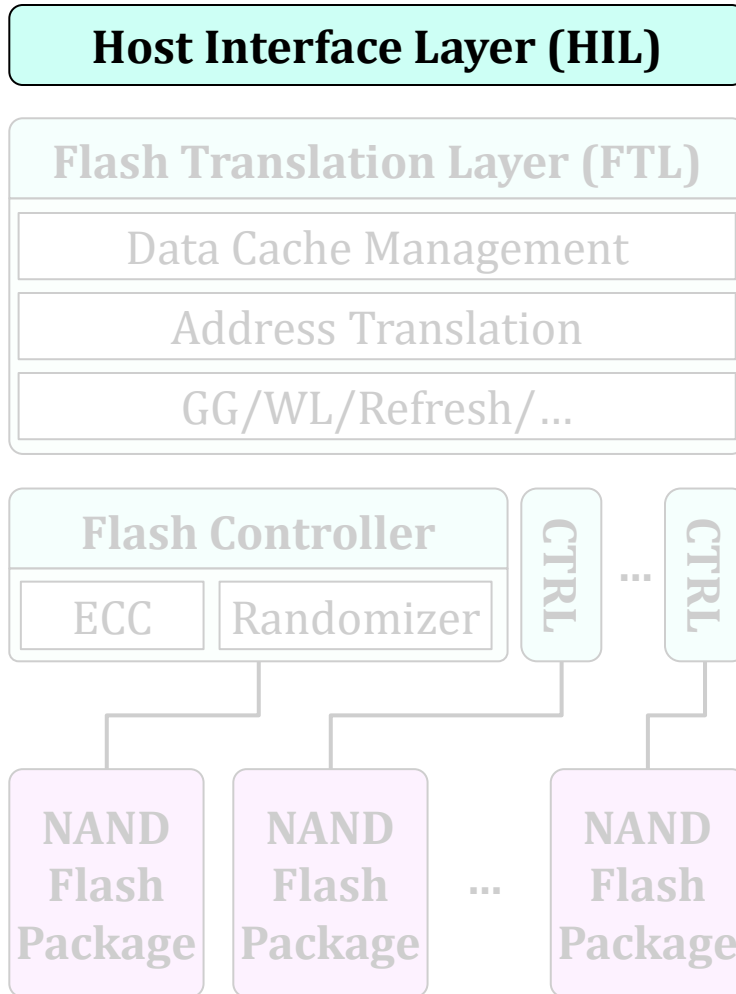
- A modern SSD is a complicated system that consists of multiple cores, HW controllers, DRAM, and NAND flash memory packages



Another Overview

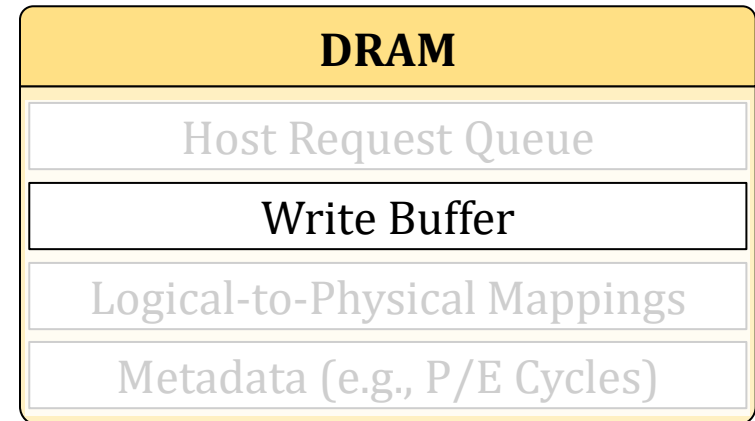
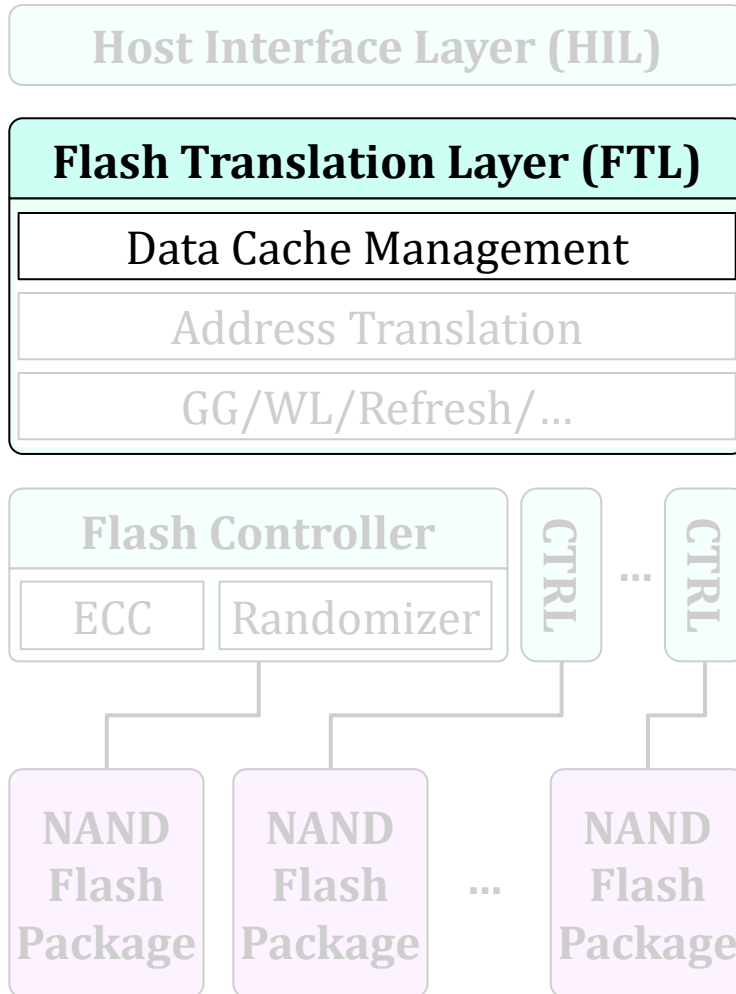


Request Handling: Write



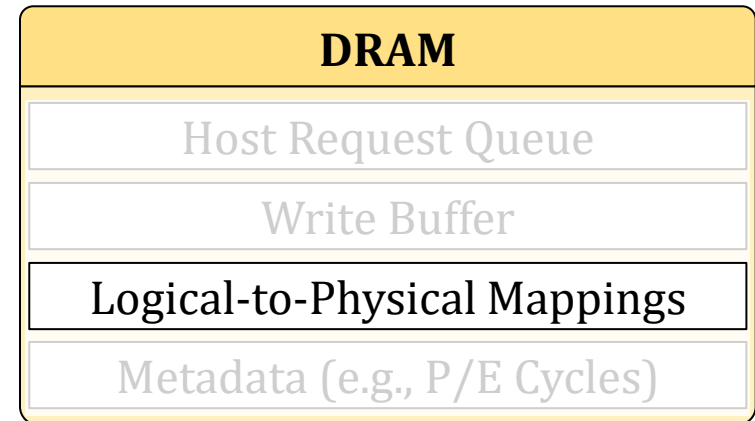
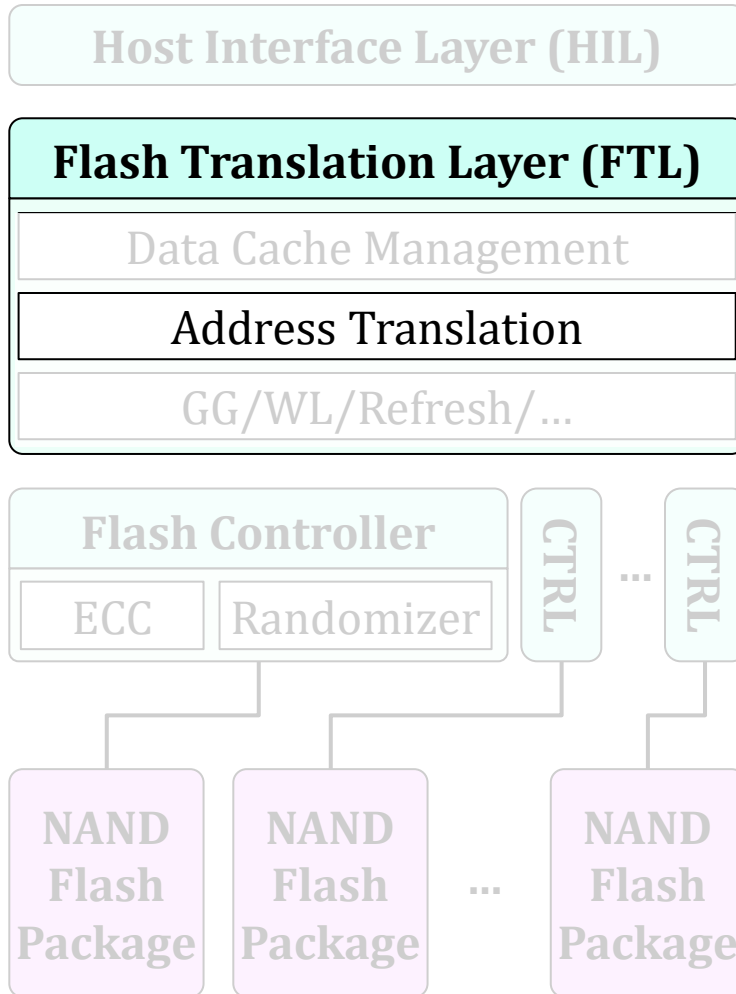
- Communication with the host operating system (receives & returns requests)
 - Via a certain interface (SATA or NVMe)
- A host I/O request includes
 - Request direction (read or write)
 - Offset (start sector address)
 - Size (number of sectors)
 - Typically aligned by 4 KiB

Request Handling: Write



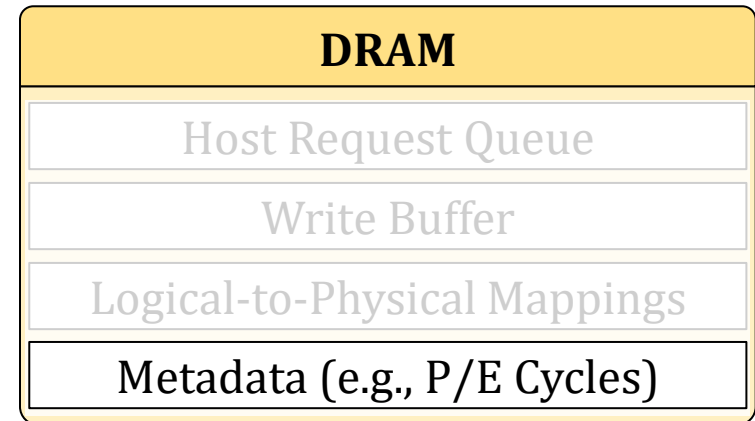
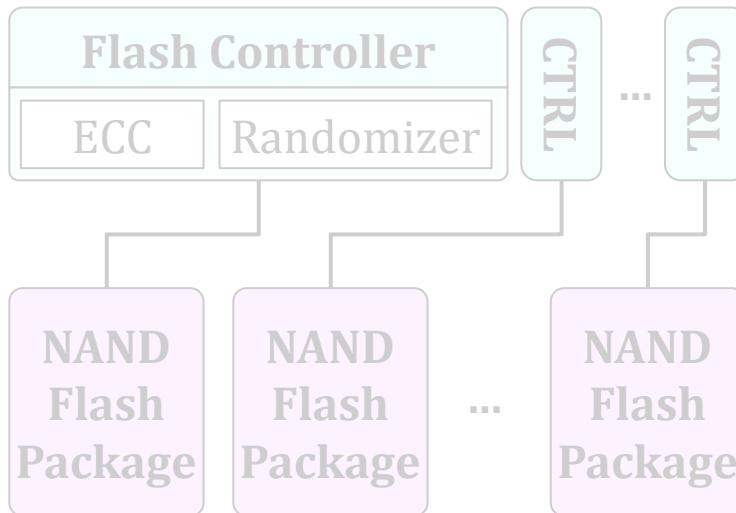
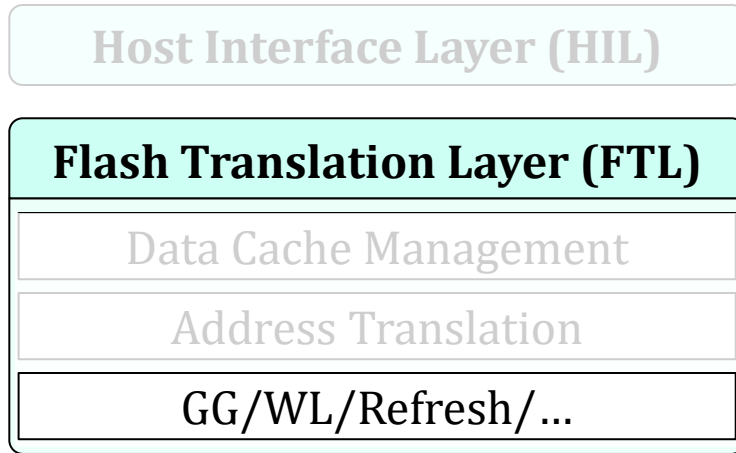
- Buffering data to write (read from NAND flash memory)
 - Essential to reducing write latency
 - Enables flexible I/O scheduling
 - Helpful for improving lifetime (not so likely)
- Limited size (e.g., tens of MBs)
 - Needs to ensure data integrity even under sudden power-off
 - Most DRAM capacity is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

Request Handling: Write



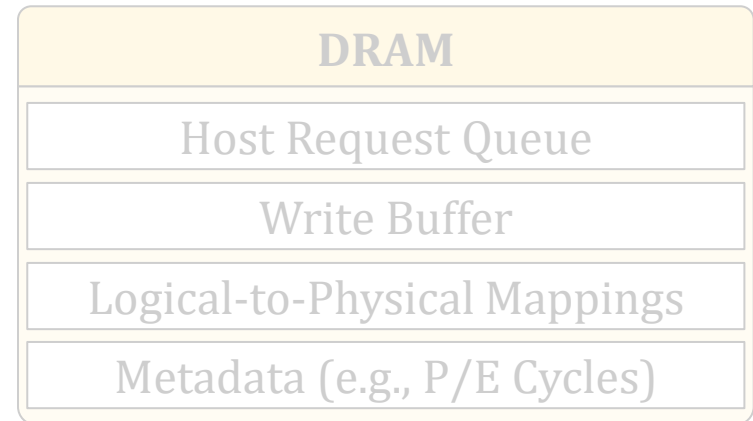
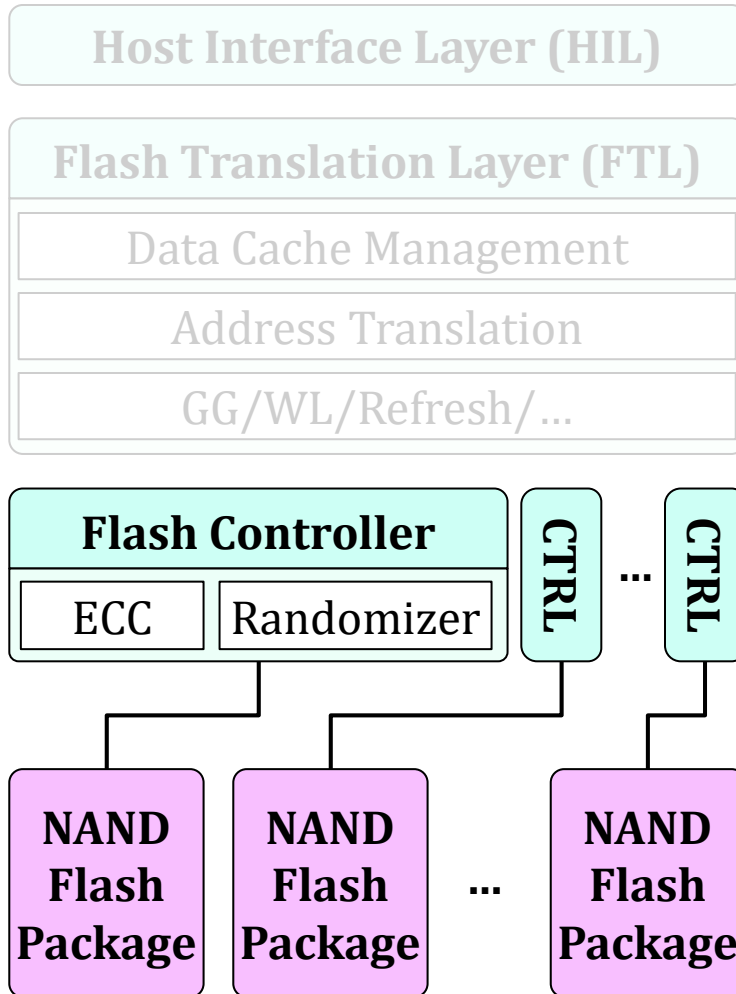
- Core functionality for out-of-place writes
 - To hide the erase-before-write property
- Needs to maintain L2P mappings
 - Logical Page Address (LPA)
→ Physical Page Address (PPA)
- Mapping granularity: 4 KiB
 - 4 Bytes for 4 KiB → 0.1% of SSD capacity

Request Handling: Write



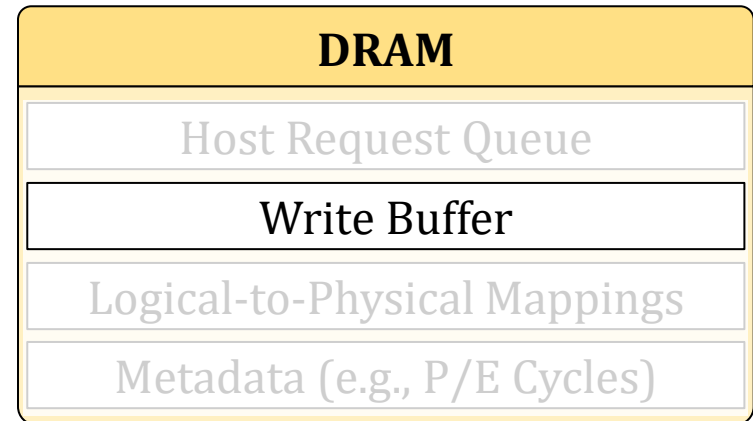
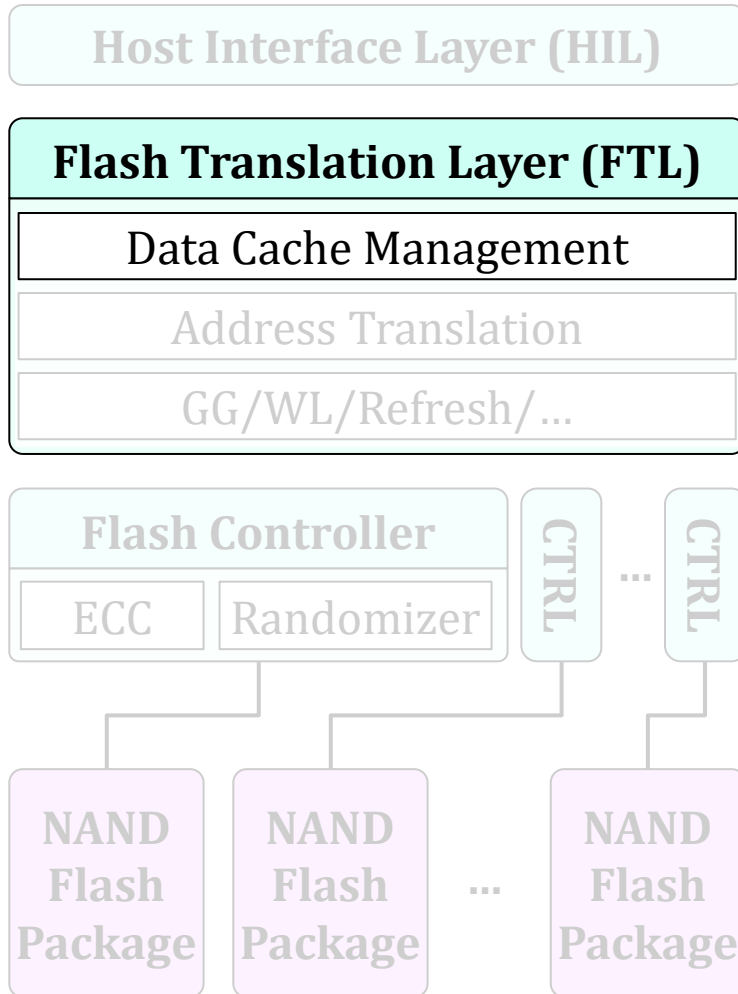
- Garbage collection (GC)
 - Reclaims free pages
 - Selects a victim block → copies all valid pages → erase the victim block
- Wear-leveling (WL)
 - Evenly distributes P/E cycles across NAND flash blocks
 - Hot/cold swapping
- Data refresh
 - Refresh pages with long retention ages

Request Handling: Write



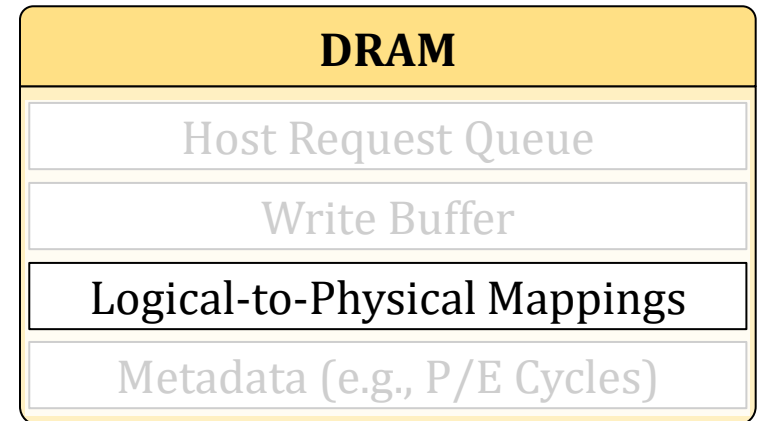
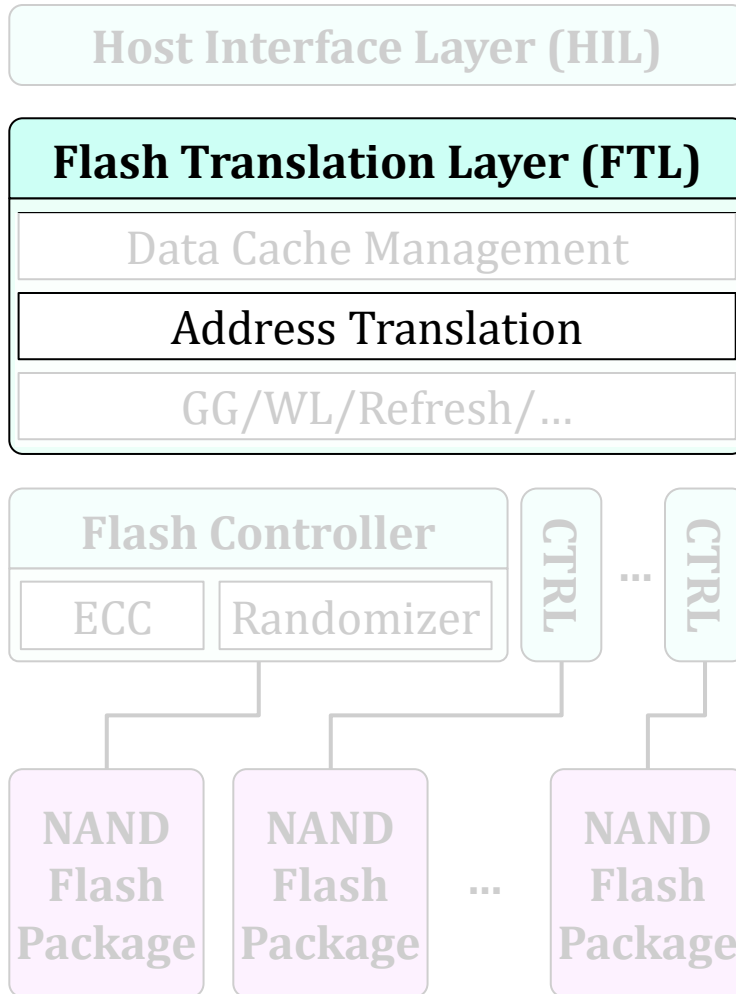
- Randomizer
 - Scrambling data to write
 - To avoid worst-case data patterns that can lead to significant errors
- Error-correcting codes (ECC)
 - Can detect/correct errors: e.g., 72 bits/1 KiB error-correction capability
 - Stores additional parity information together with raw data
- Issues NAND flash commands

Request Handling: Read



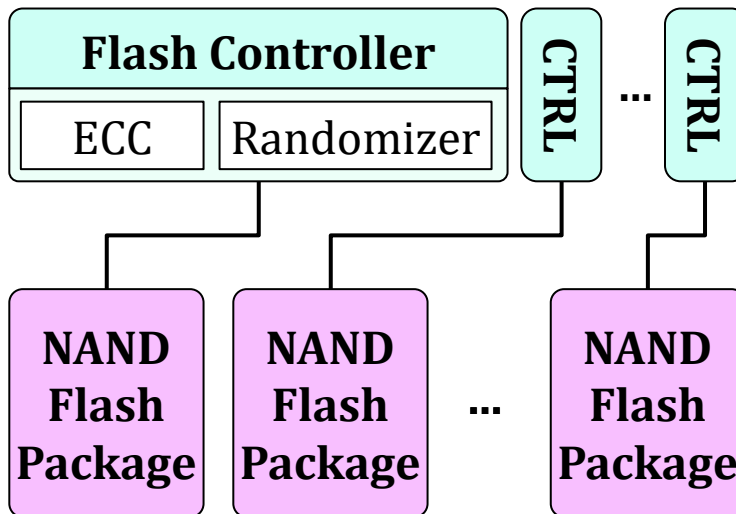
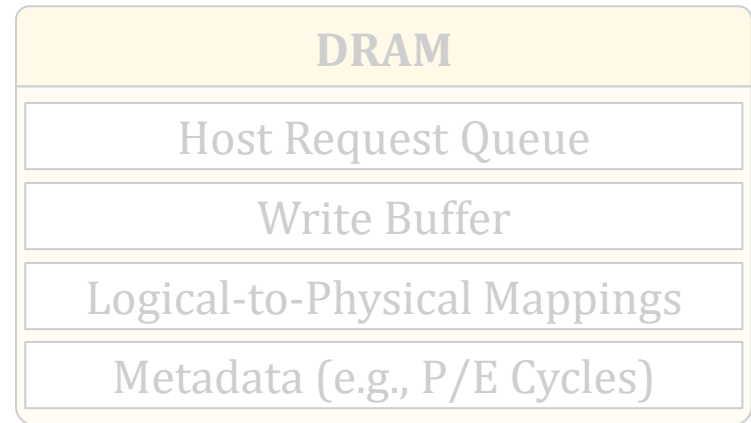
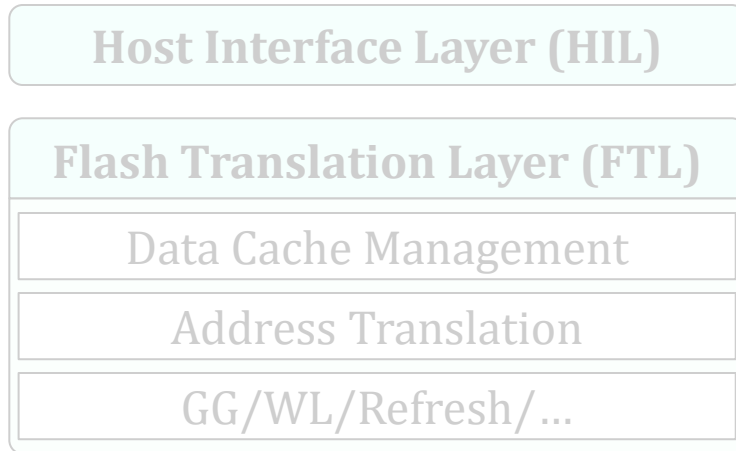
- First checks if the request data exists in the write buffer
 - If so, returns the corresponding request immediately with the data
- A host read request can be involved with several pages
 - Such a request can be returned only after all the requested data is ready

Request Handling: Read



- Finds the PPA where the request data is stored from the L2P mapping table

Request Handling: Read



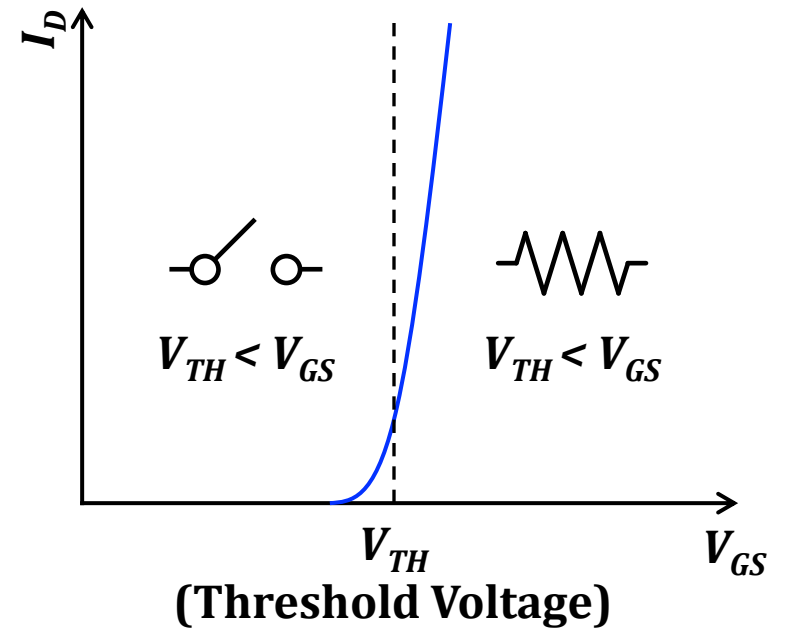
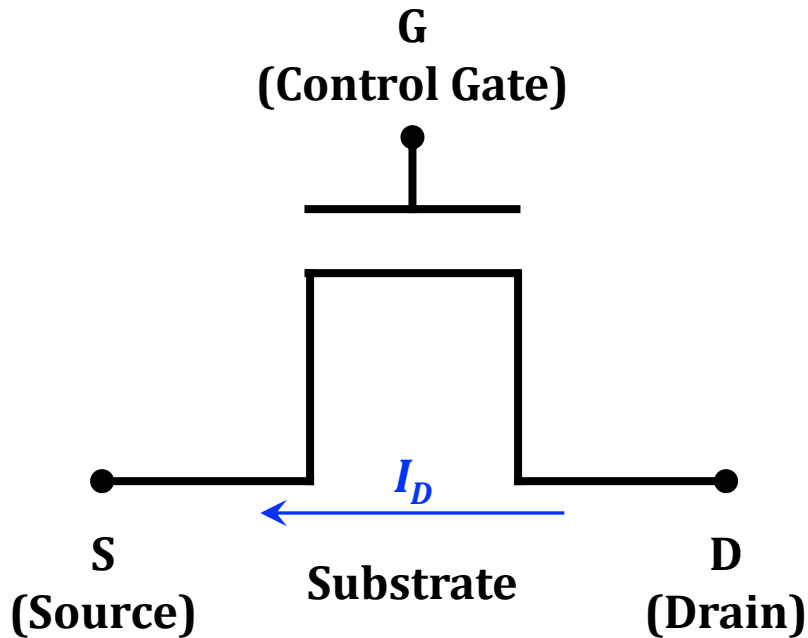
- First reads the raw data from the flash chip
- Performs ECC decoding
- Derandomizes the raw data
- ECC decoding can fail
 - Retries reading of the page w/ adjusted V_{REF}
 - Soft-decision ECC (e.g., LDPC)

Today's Agenda

- SSD Organization & Request Handling
- **NAND Flash Organization**
- NAND Flash Operation

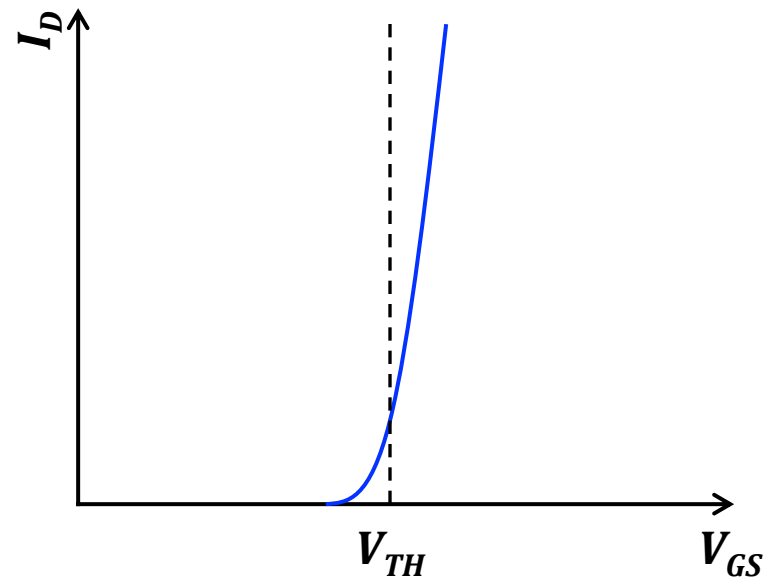
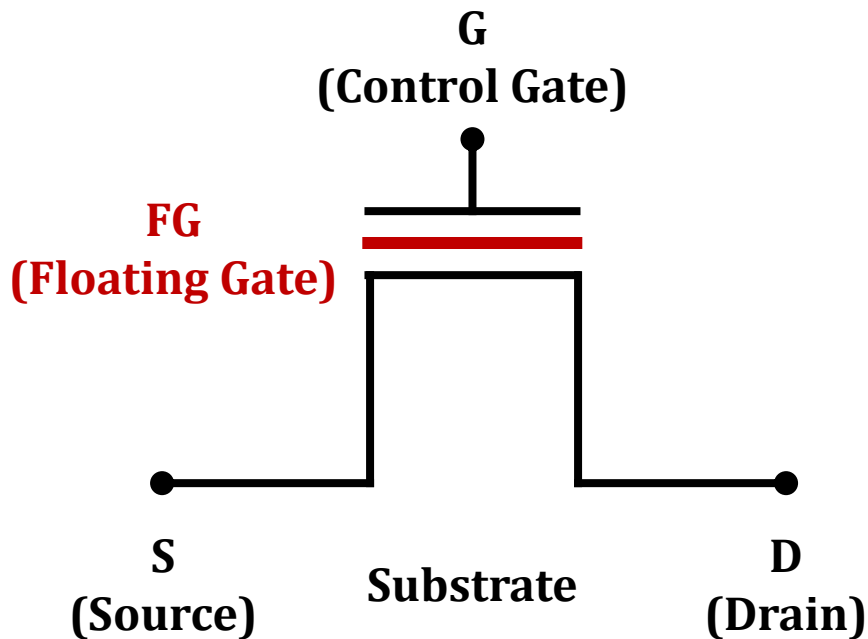
A Flash Cell

- Basically, it is a transistor



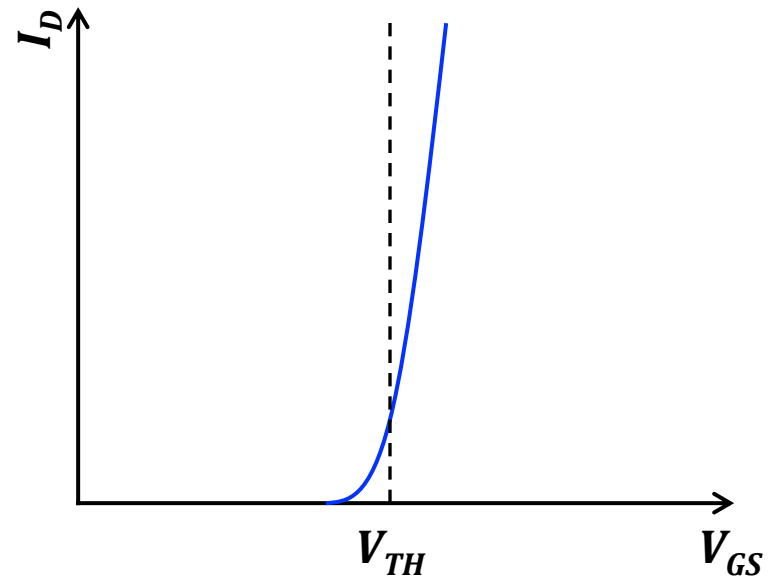
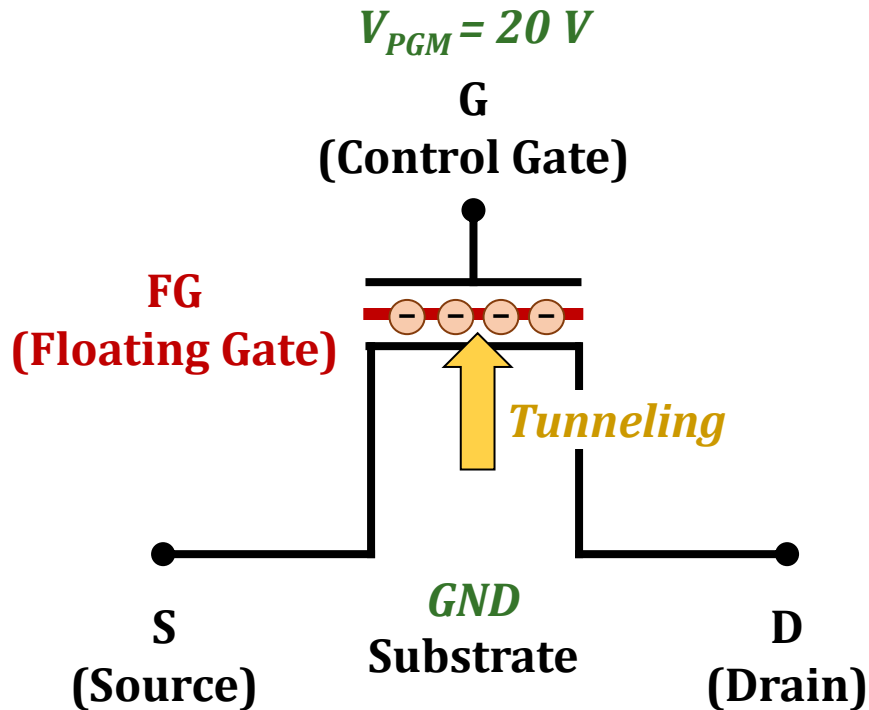
A Flash Cell

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



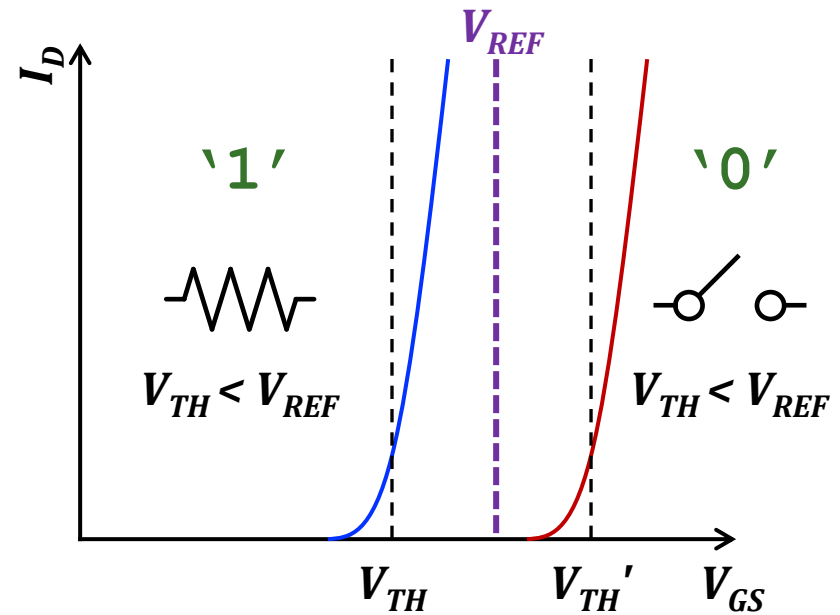
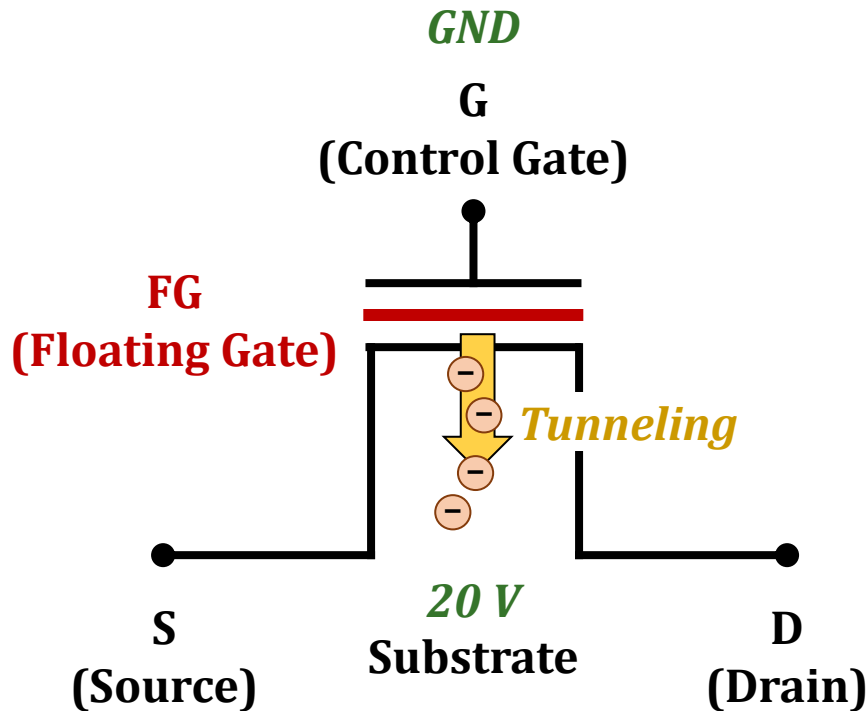
A Flash Cell

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



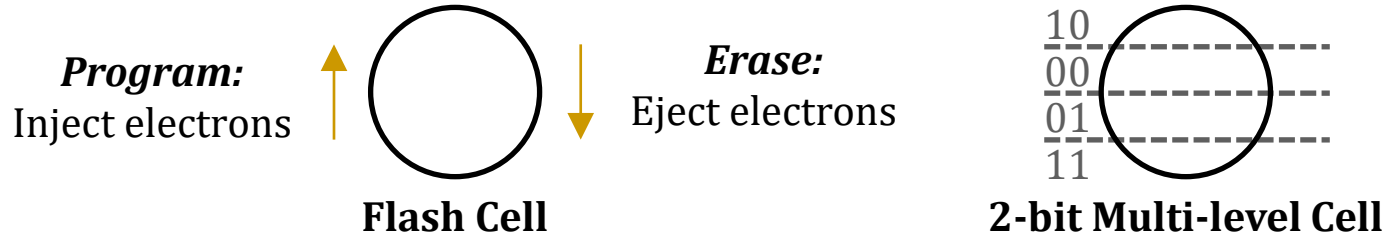
A Flash Cell

- 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})

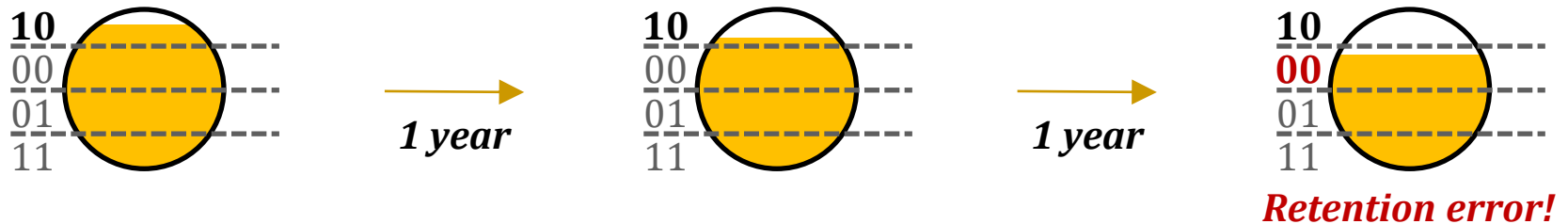


Flash Cell Characteristics

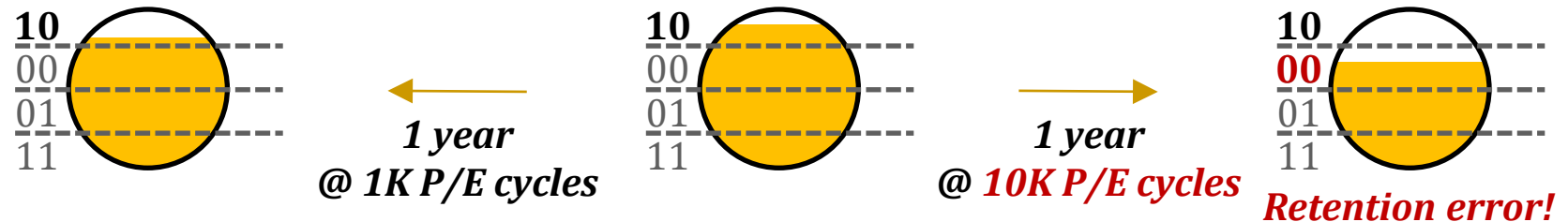
- Multi-leveling: A flash cell can store multiple bits



- Retention loss: A cell leaks electrons over time

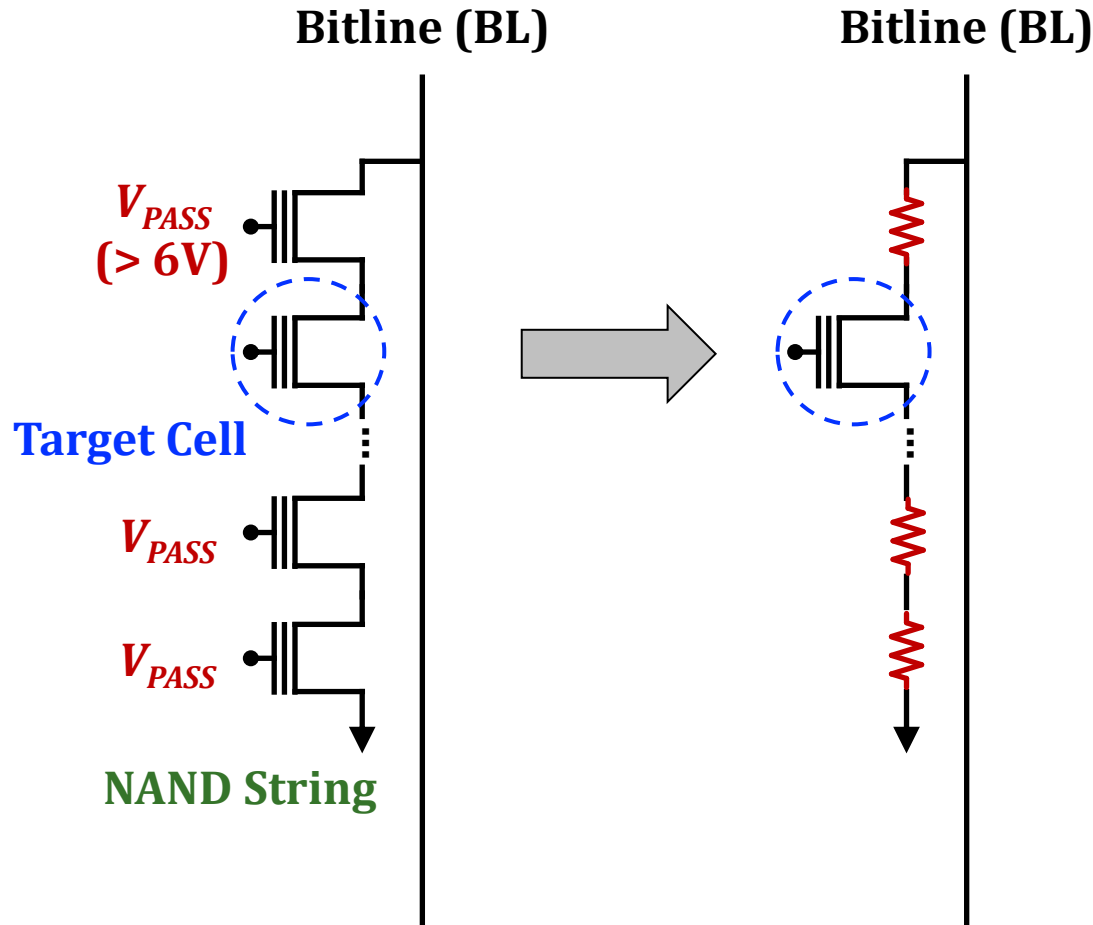


- Limited lifetime: A cell wears out after P/E cycling



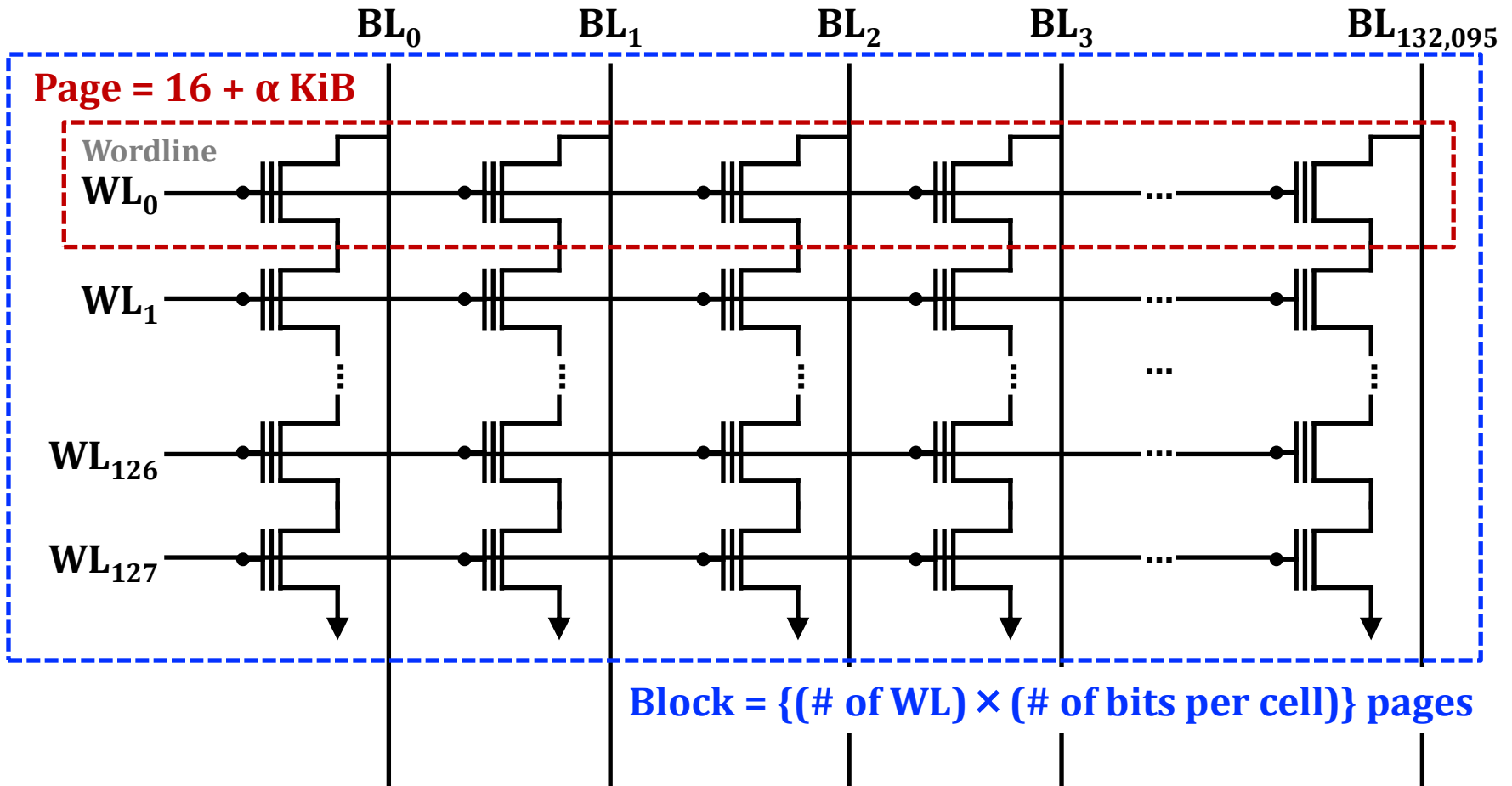
A NAND String

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



Pages and Blocks

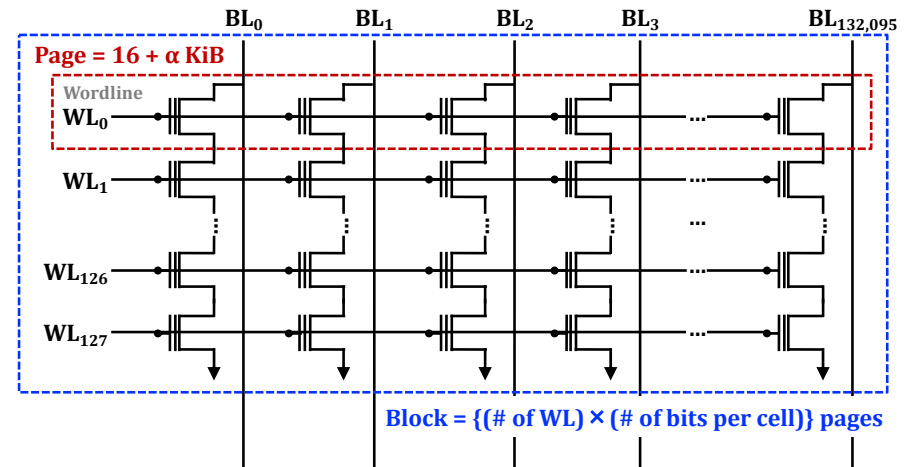
- A large number ($> 100,000$) of cells operate concurrently



Pages and Blocks (Continued)

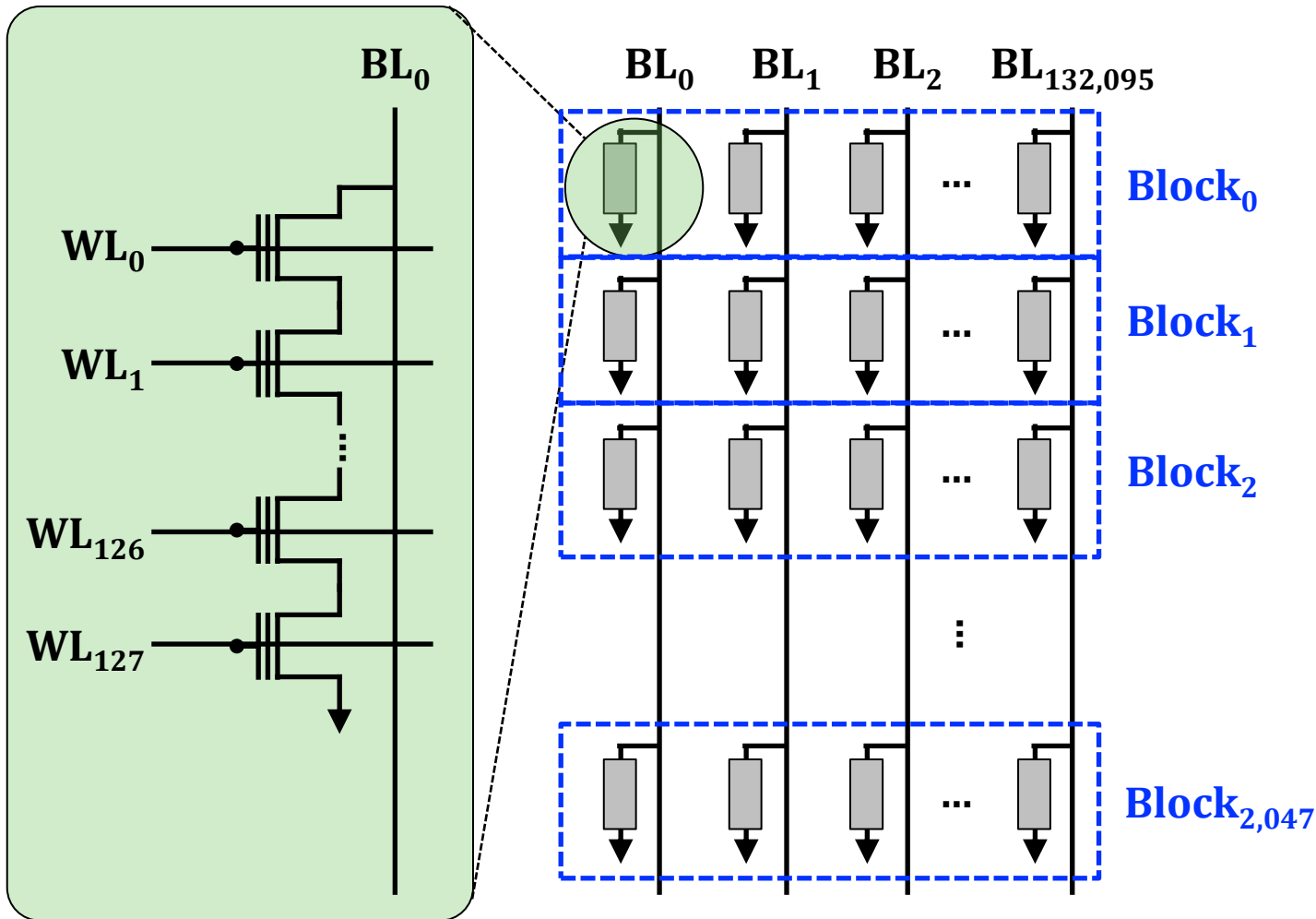
- Program and erase: Unidirectional
 - Programming a cell → Increasing the cell's V_{TH}
 - Erasing a cell → Decreasing the cell's V_{TH}
- Programming a page cannot change '0' cells to '1' cells
→ Erase-before-write property

- Erase unit: Block
 - Increase erase bandwidth
 - Makes in-place write on a page very inefficient
→ Out-of-place write & GC



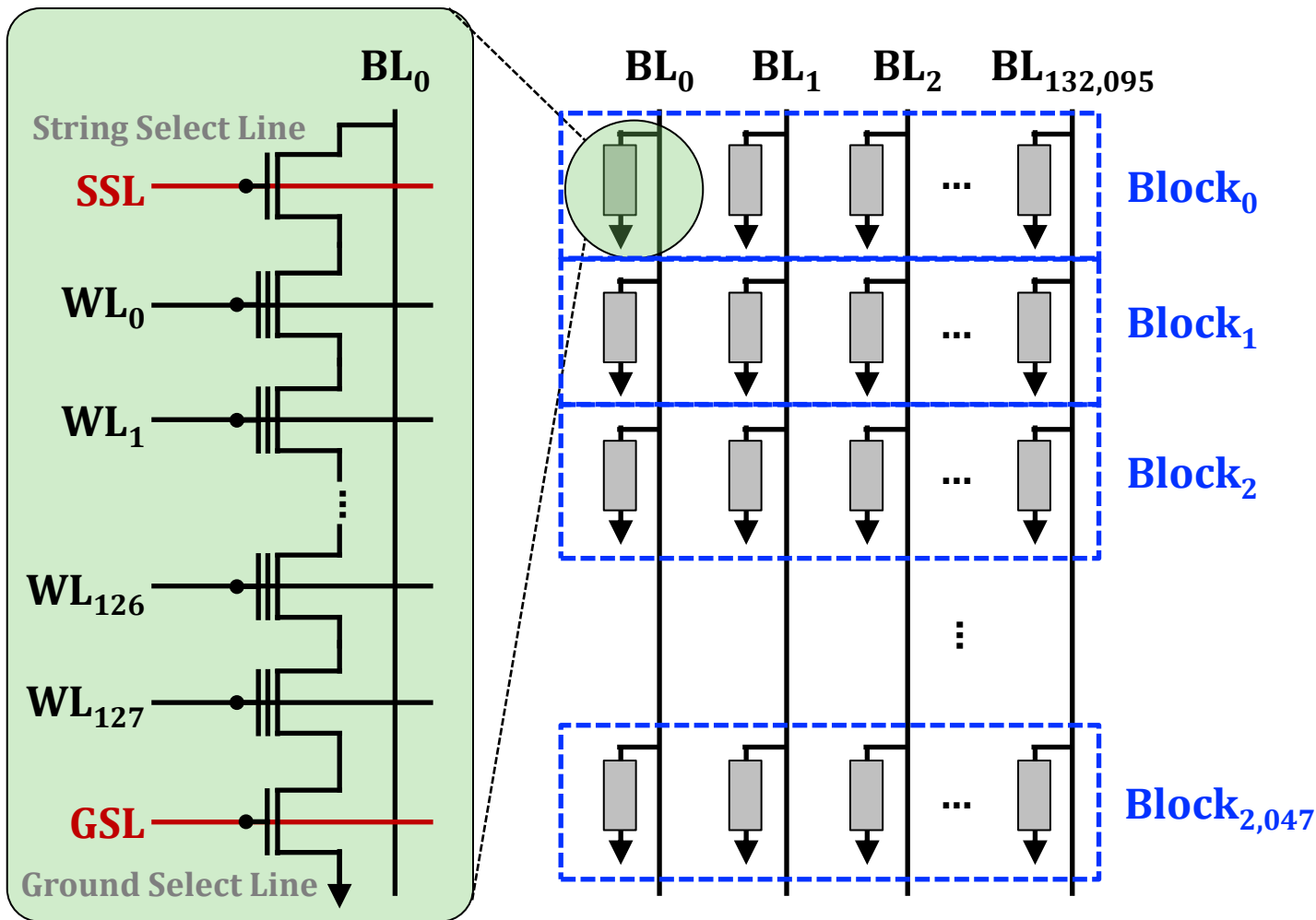
Planes

- A large number ($> 1,000$) of blocks share bitlines in a plane



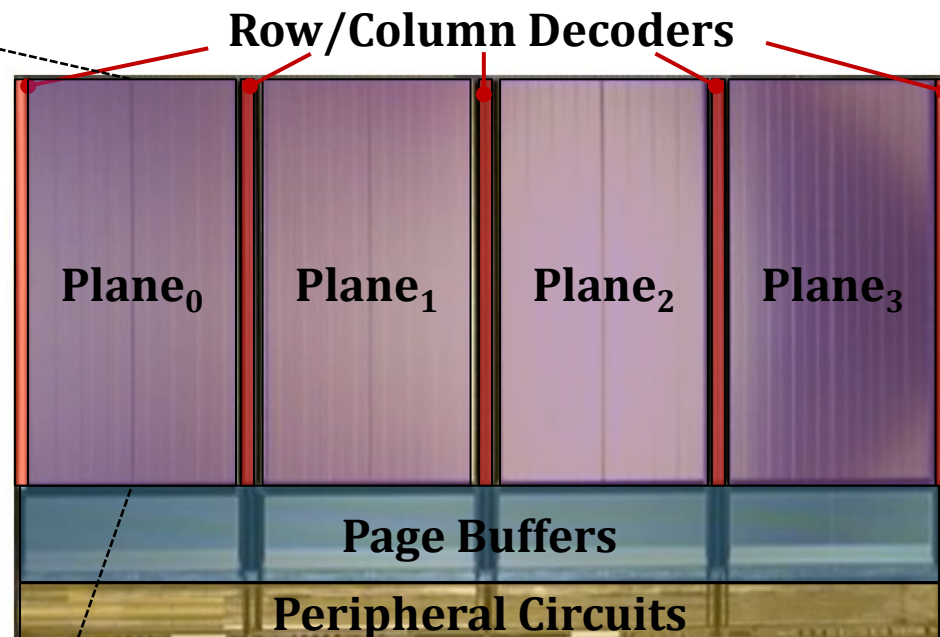
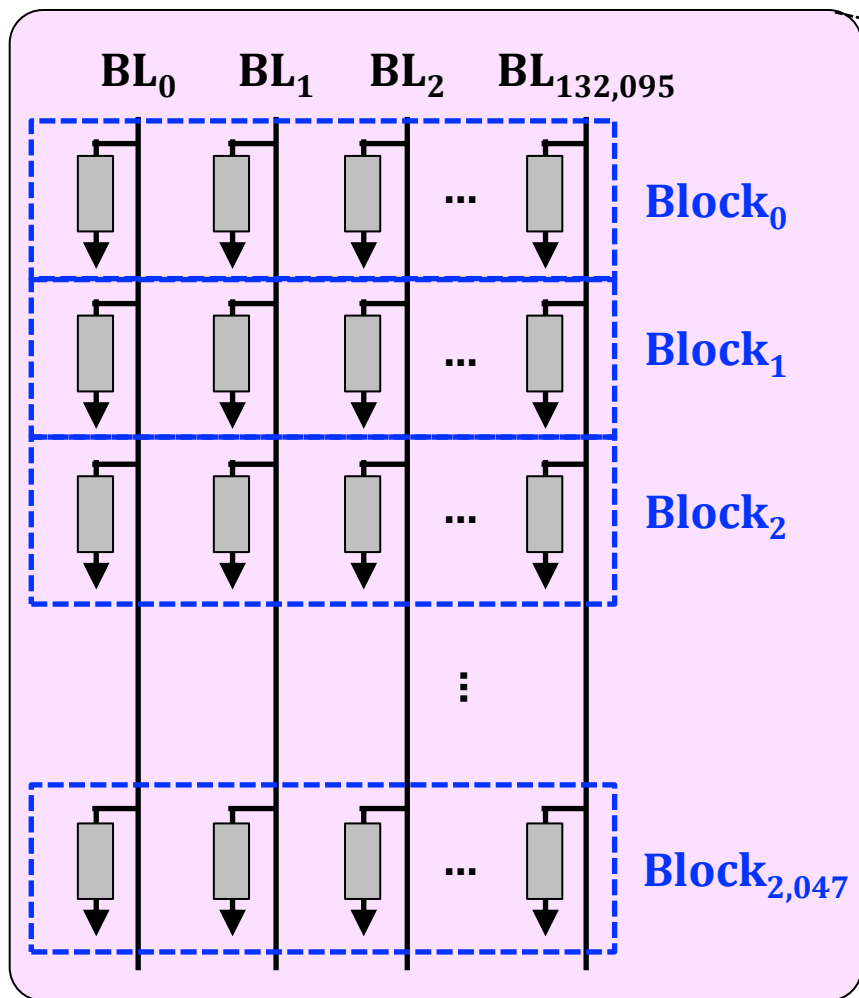
Planes

- A large number ($> 1,000$) of blocks share bitlines in a plane



Planes and Dies

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



A 21-nm 2D NAND Flash Die

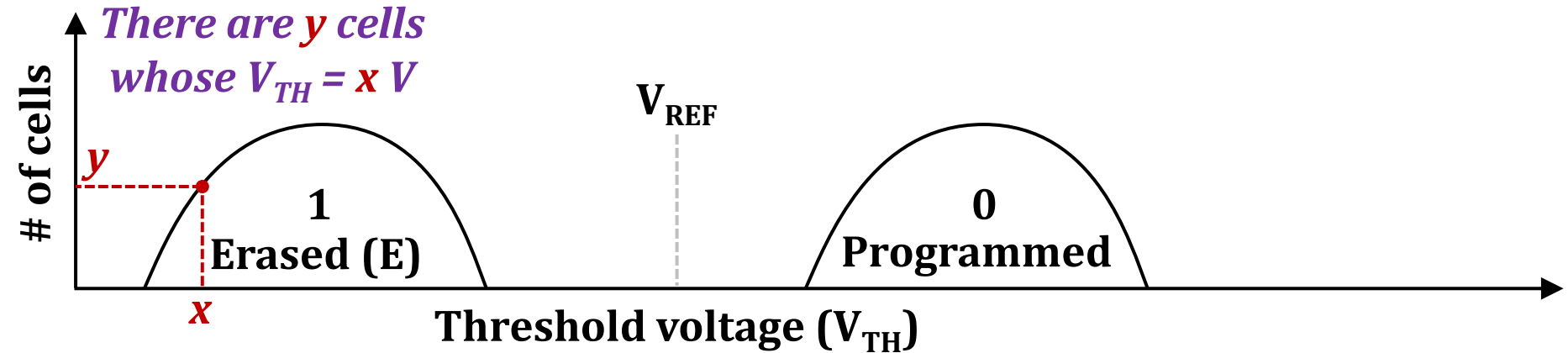
- Planes share decoders: limits internal parallelism (only operations @ the same WL offset)

Today's Agenda

- SSD Organization & Request Handling
- NAND Flash Organization
- **NAND Flash Operation**

Threshold Voltage Distribution

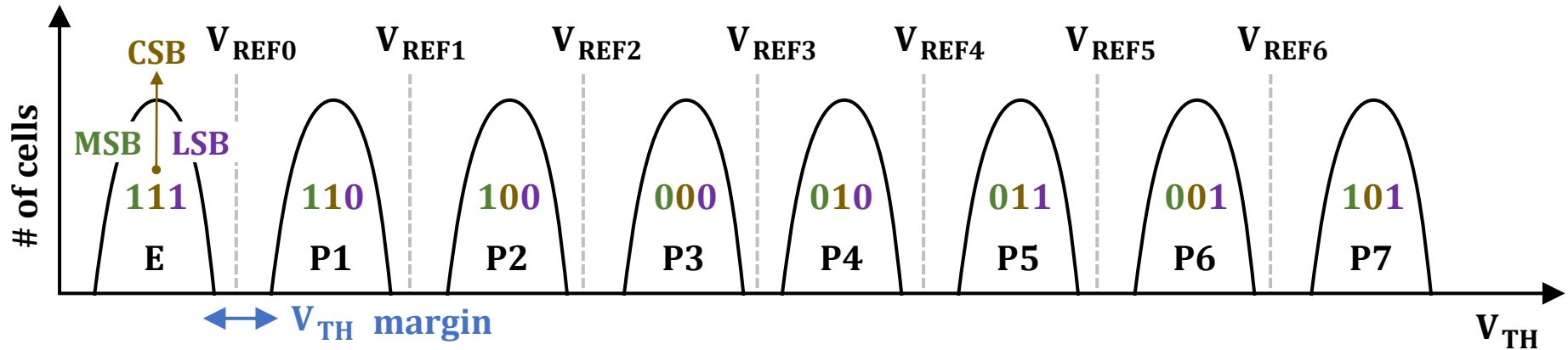
- V_{TH} distribution of **cells** in a **programmed page/block/chip**



- Why **distribution**? **Variations** across the cells
 - Some cells are more easily programmed or erased
- 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_{TH} Distribution of MLC NAND Flash

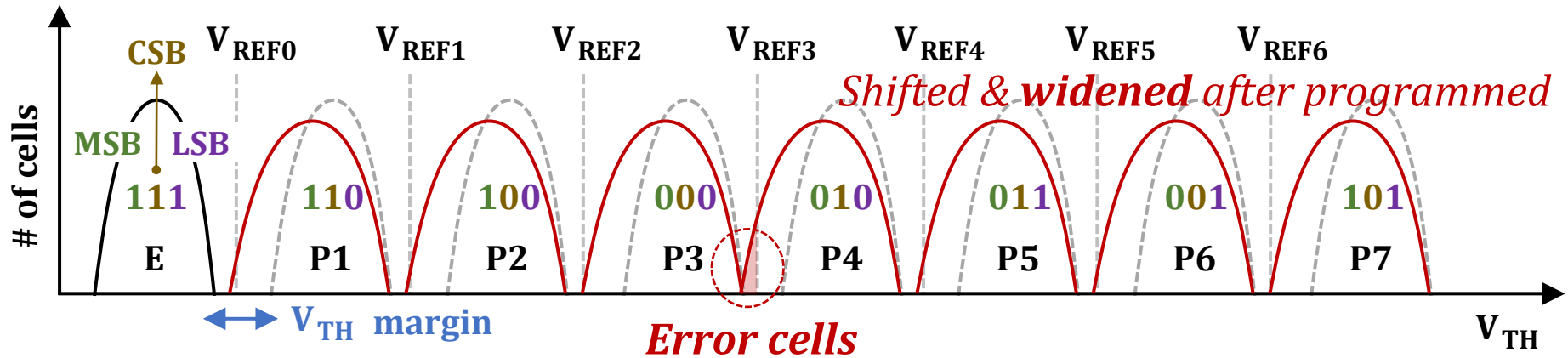
- Multi-level cell (MLC) technique
 - $2^m V_{TH}$ states required to store m bits in a single flash cell



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

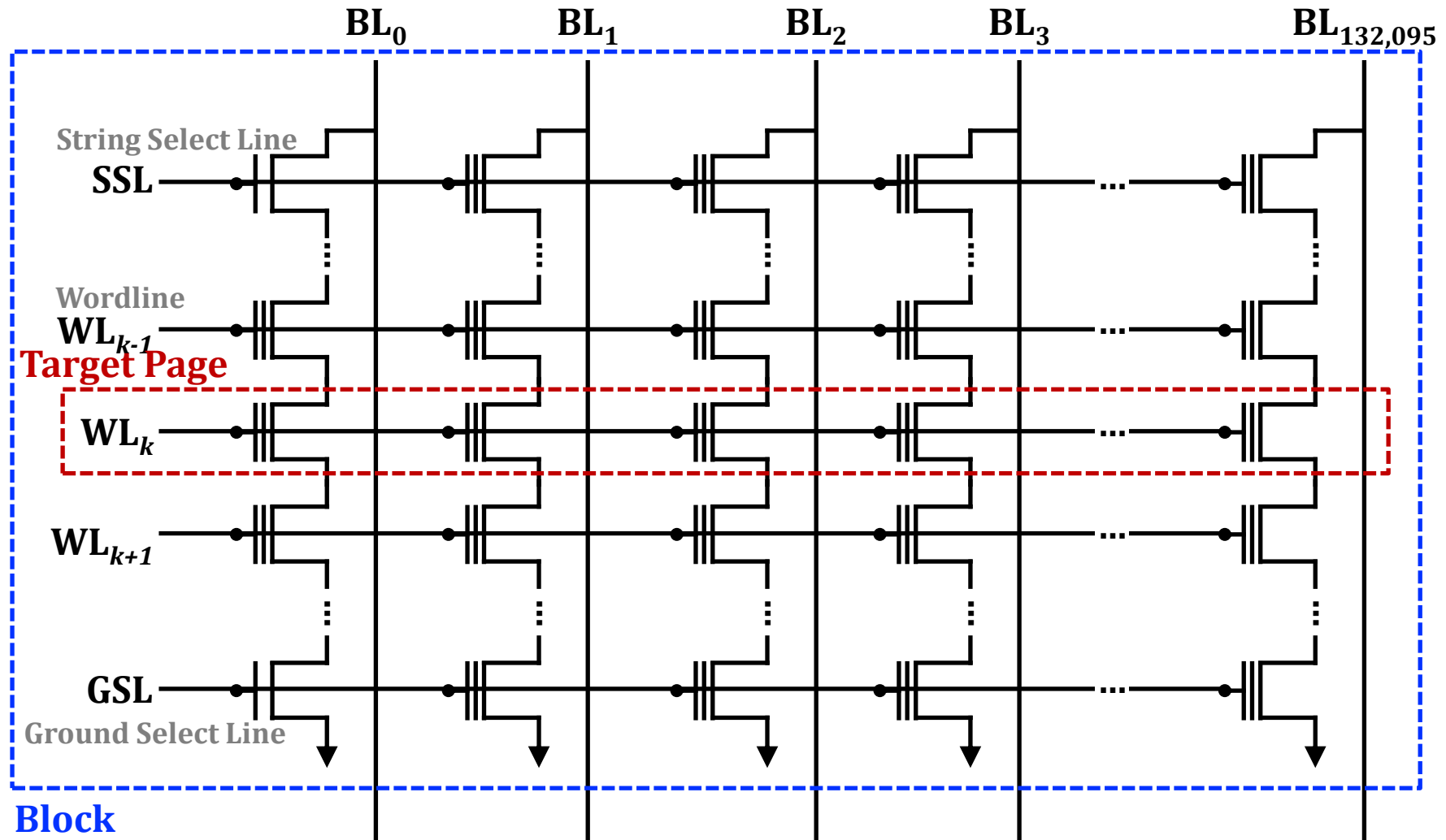
V_{TH} Distribution of MLC NAND Flash

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



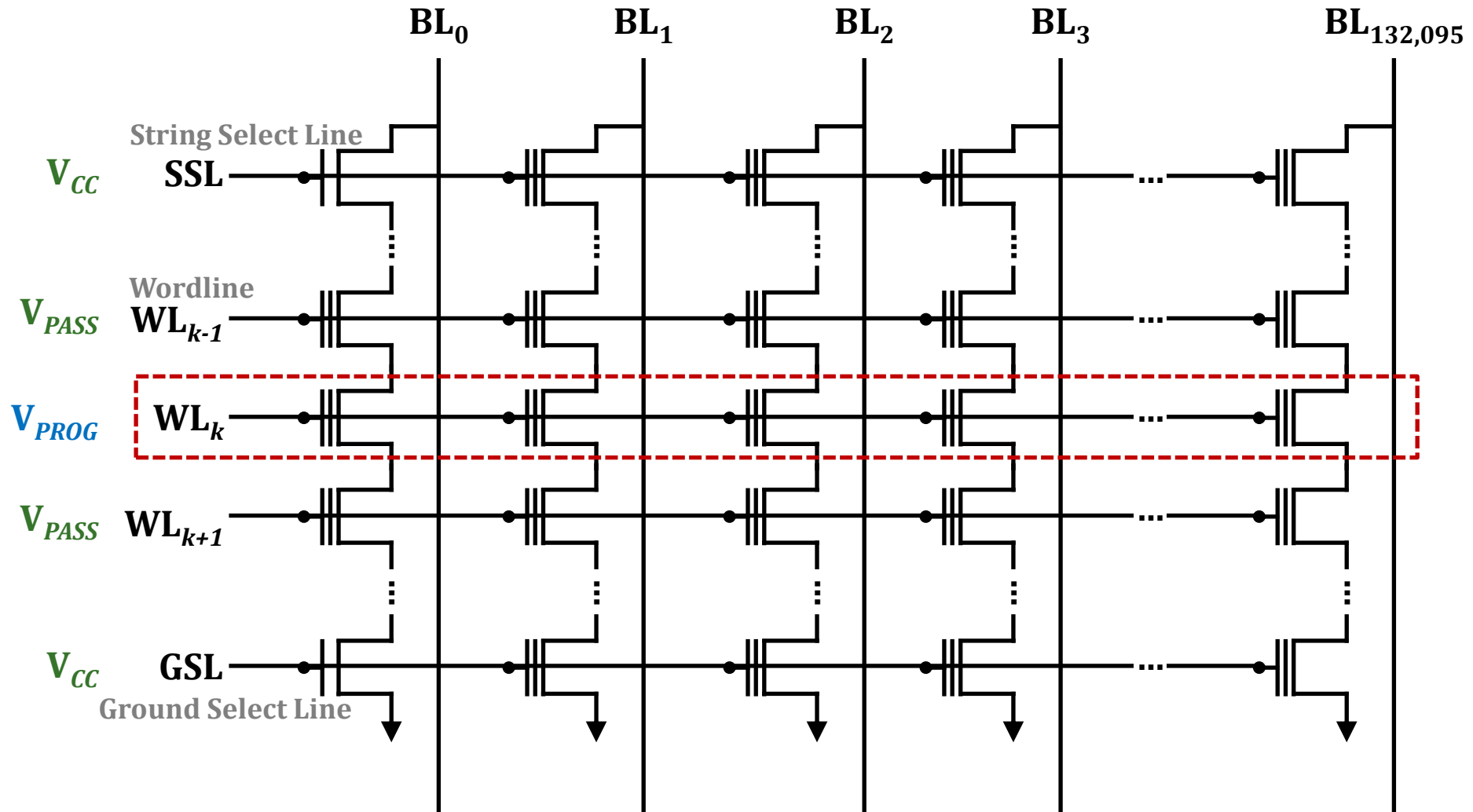
- Limited width of the V_{TH} window: Need to
 - Make each V_{TH} state narrow
 - Guarantee sufficient margins b/w adjacent V_{TH} states
 - V_{TH} changes over time after programmed
 - Narrower margins \rightarrow Lower reliability
 - More bits per cell \rightarrow higher density but lower reliability

Basic Operation: Page Program



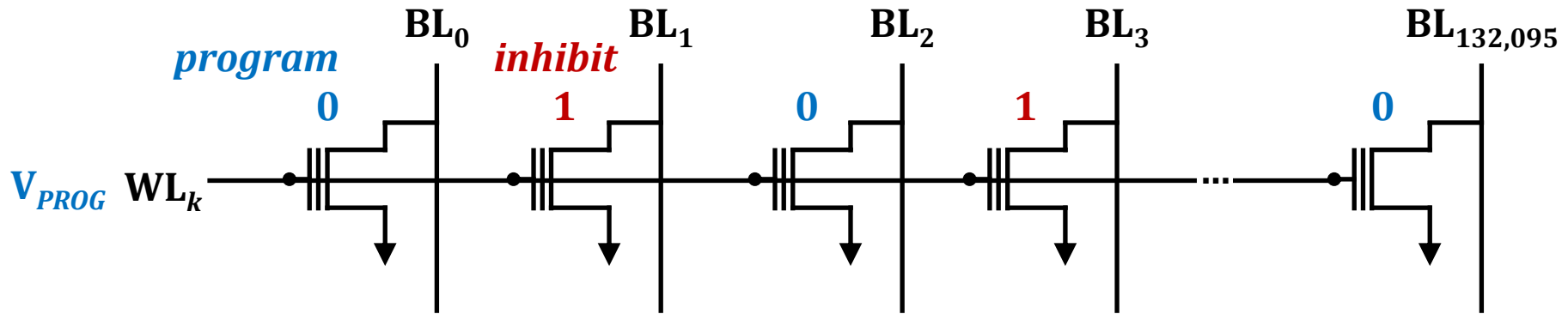
Basic Operation: Page Program

- WL control – All other cells operate as a resistance



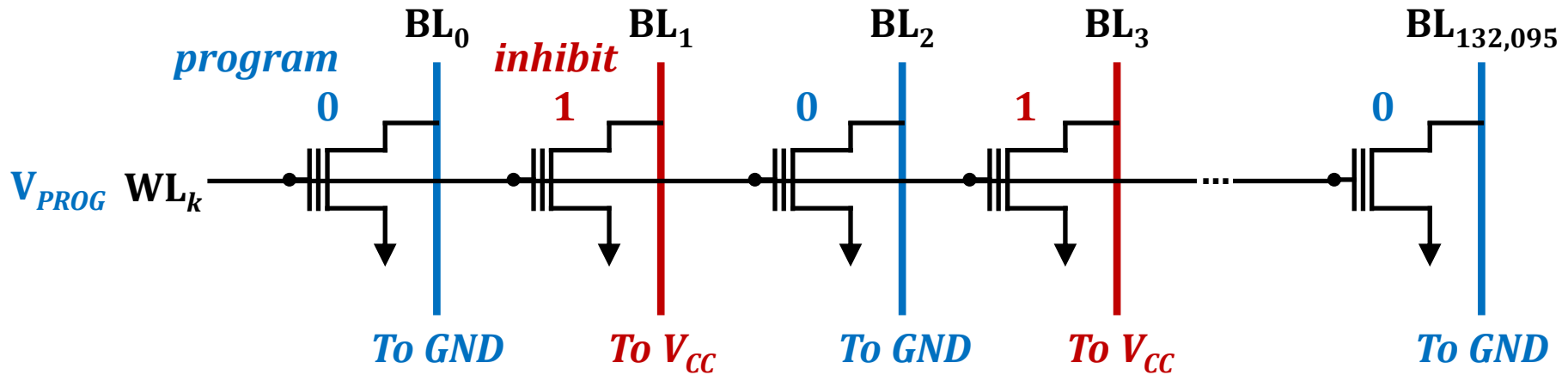
Basic Operation: Page Program

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

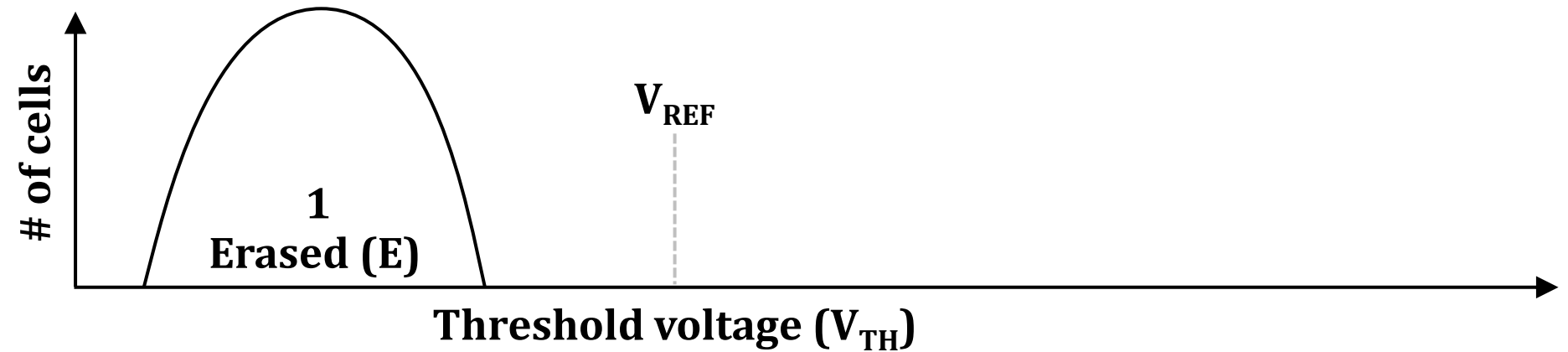
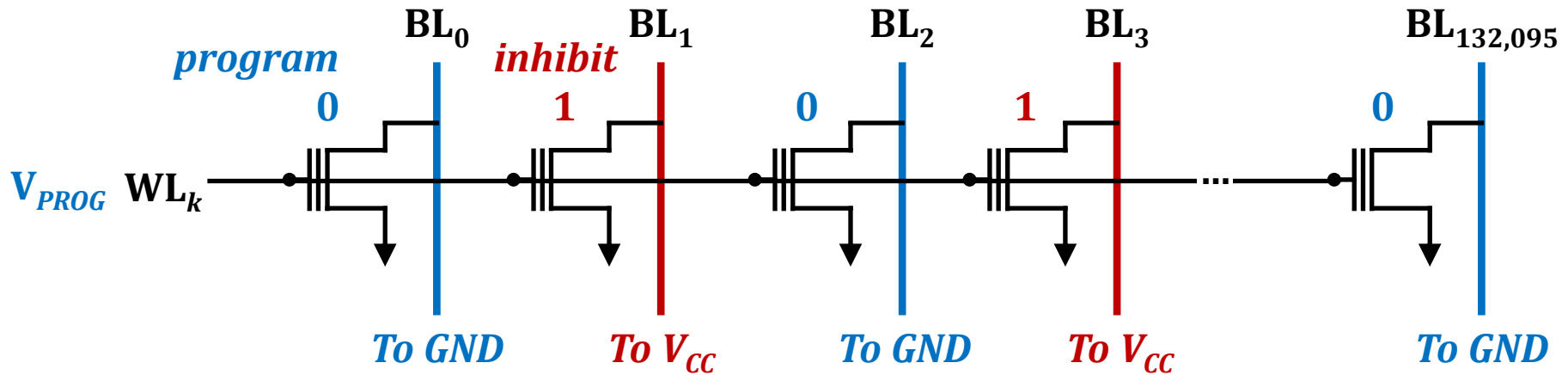


Basic Operation: Page Program

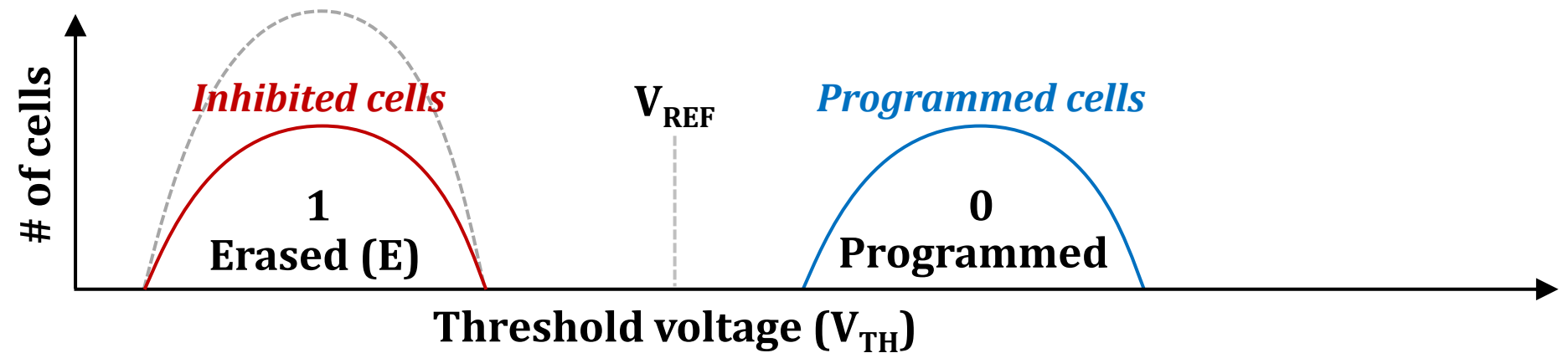
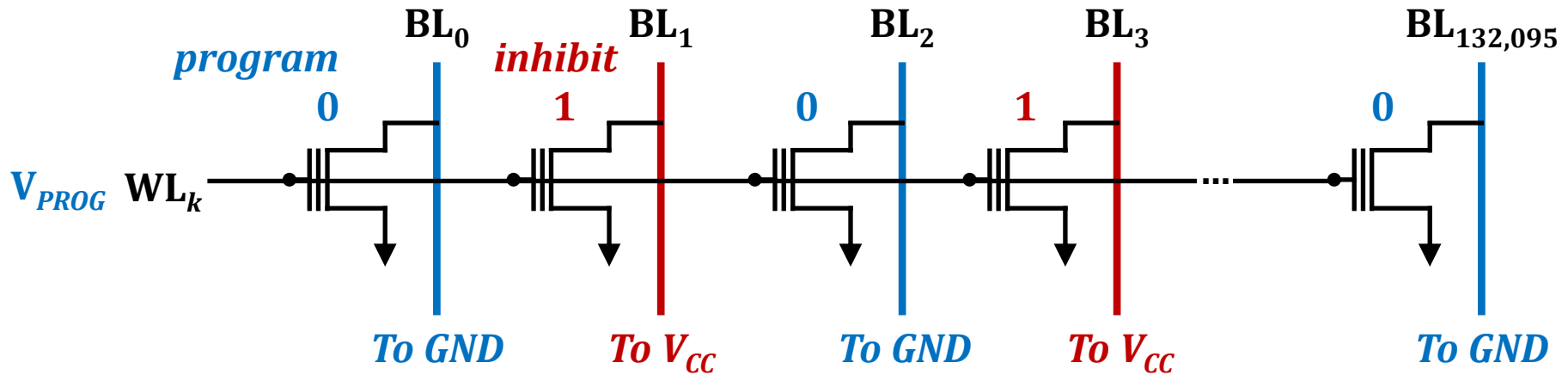
- BL control – **Inhibits cells** to not be programmed



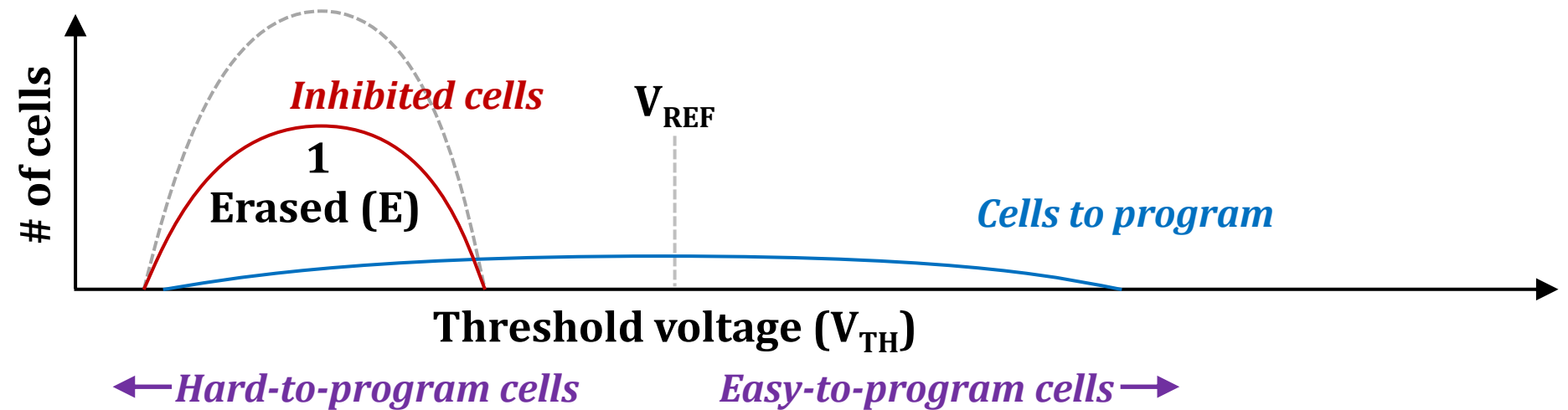
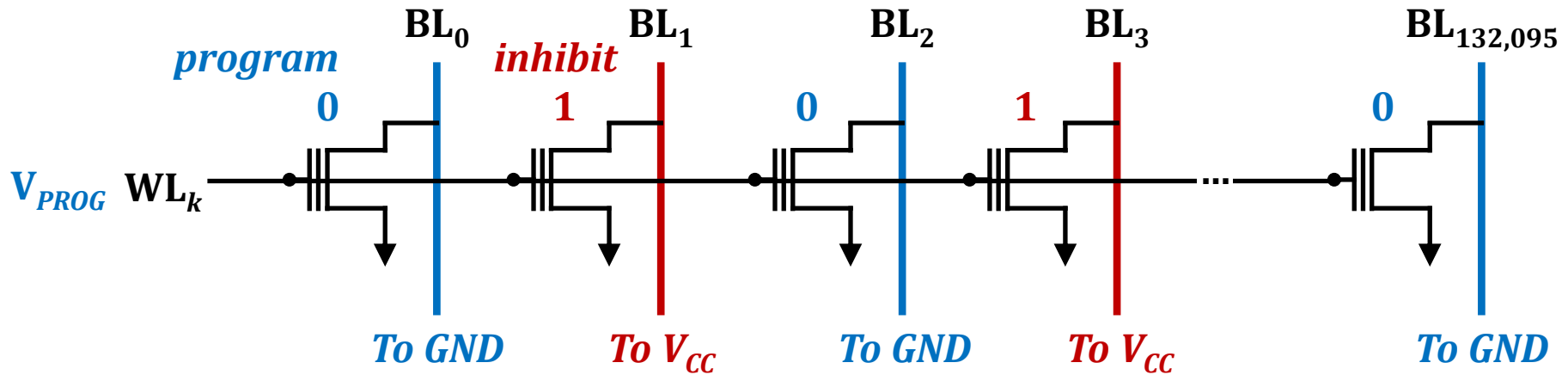
Basic Operation: Page Program



Basic Operation: Page Program

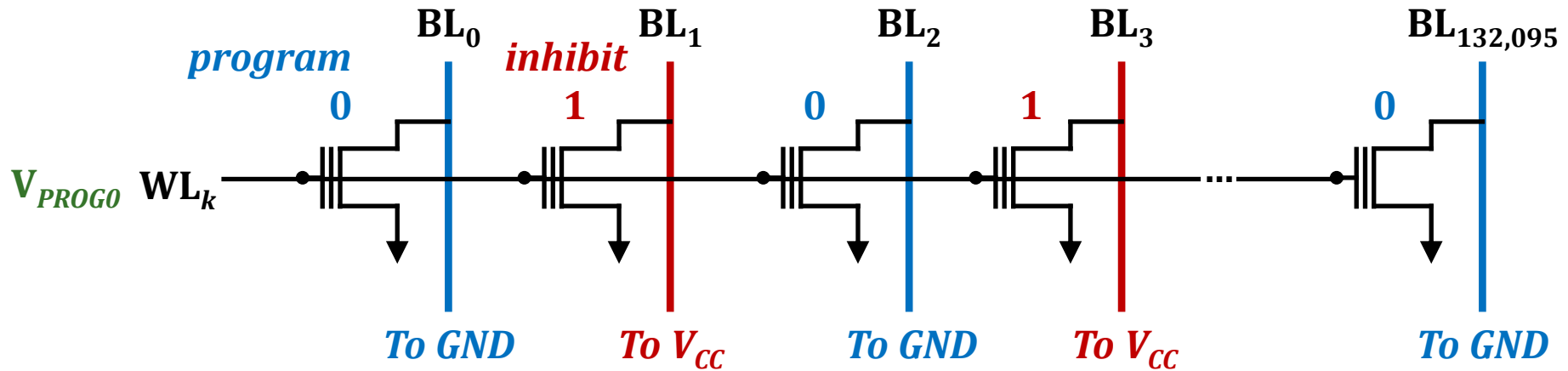


Basic Operation: Page Program

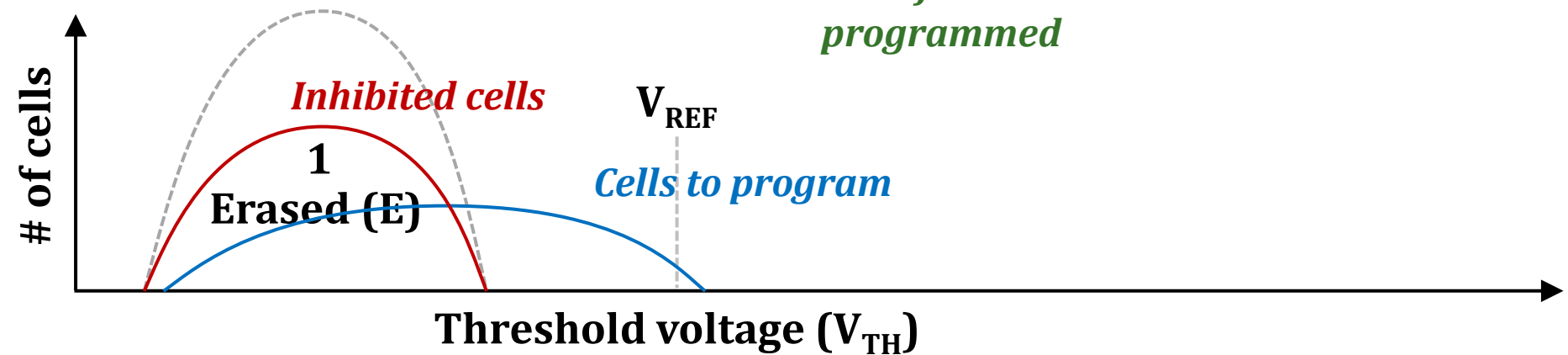


Basic Operation: Page Program

- Incremental Step-Pulse Programming (ISPP)

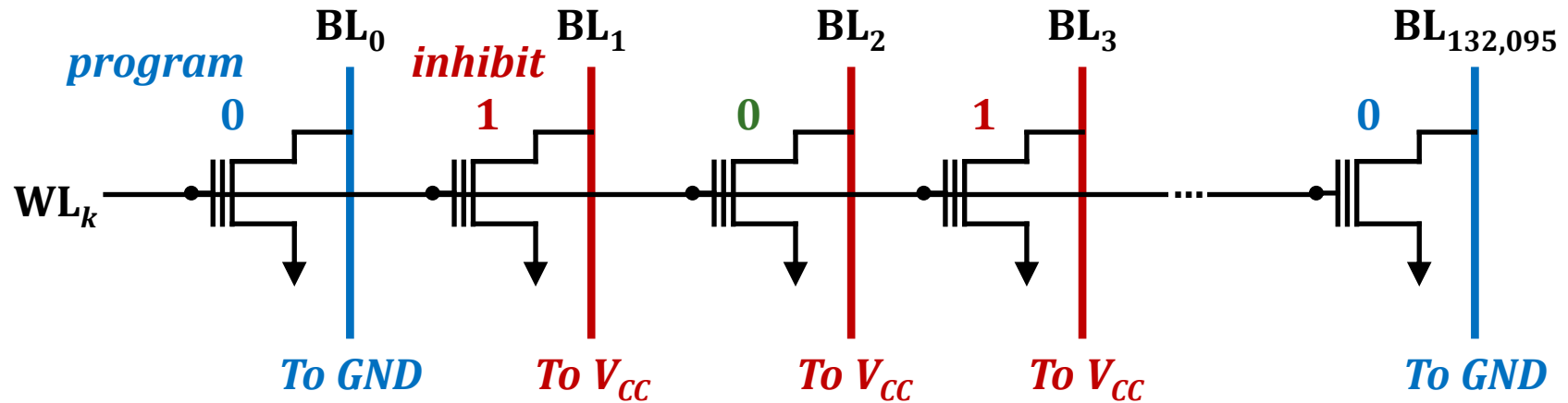


Verified as programmed



Basic Operation: Page Program

- Incremental Step-Pulse Programming (ISPP)



Inhibit

programmed cells

Inhibited cells

1
Erased (E)

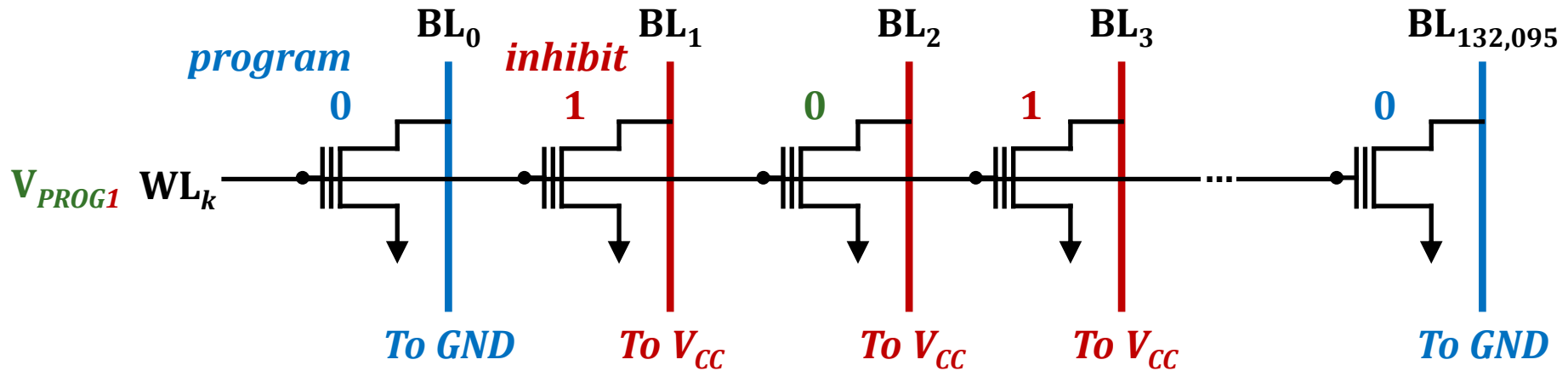
Cells to program

V_{REF}

Threshold voltage (V_{TH})

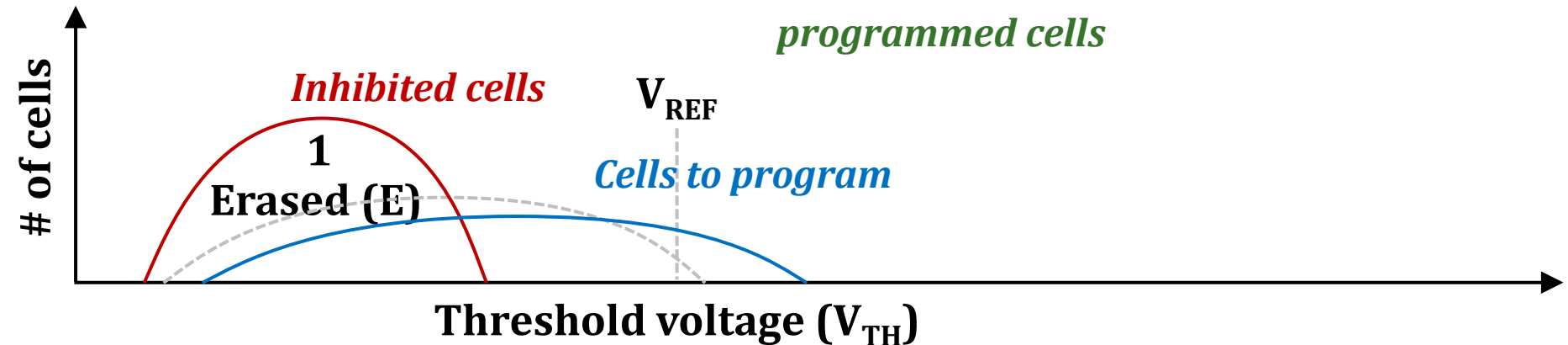
Basic Operation: Page Program

- Incremental Step-Pulse Programming (ISPP)



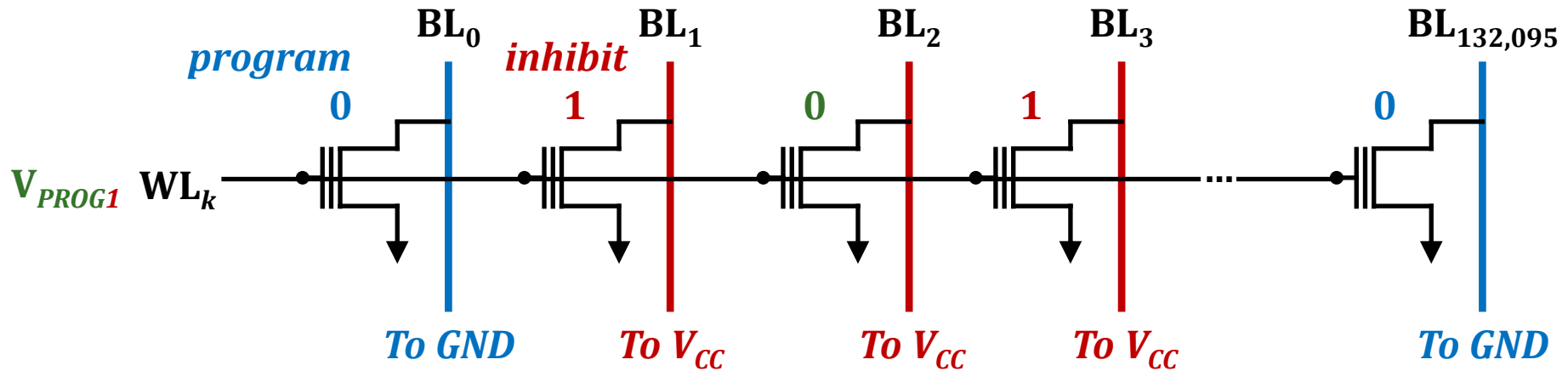
Inhibit

programmed cells



Basic Operation: Page Program

- Incremental Step-Pulse Programming (ISPP)



Inhibit

programmed cells

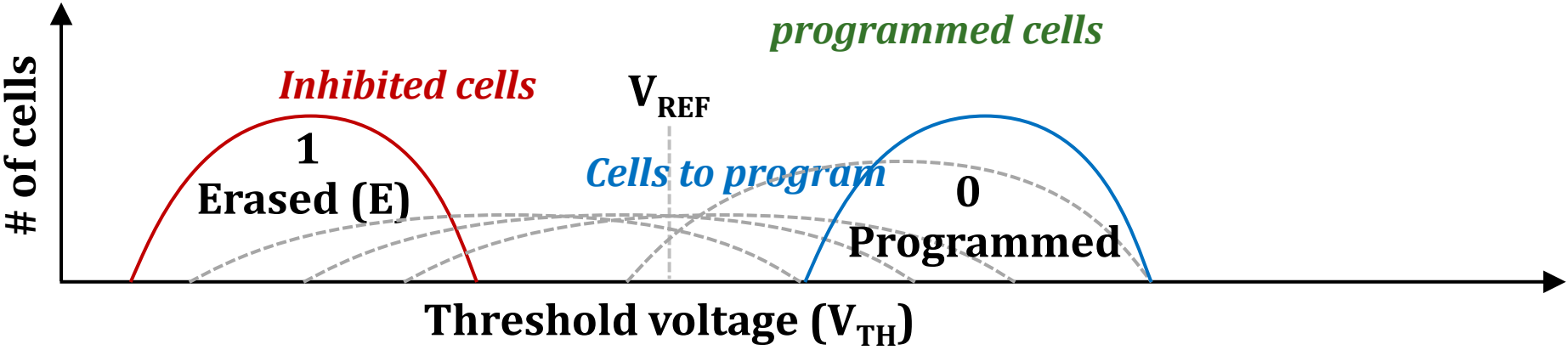
Inhibited cells

1 Erased (E)

Cells to program

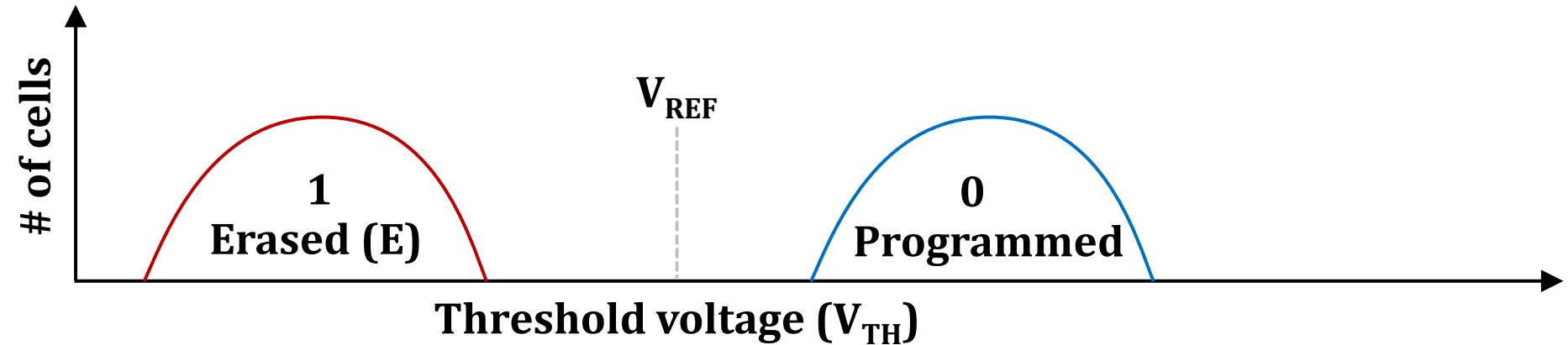
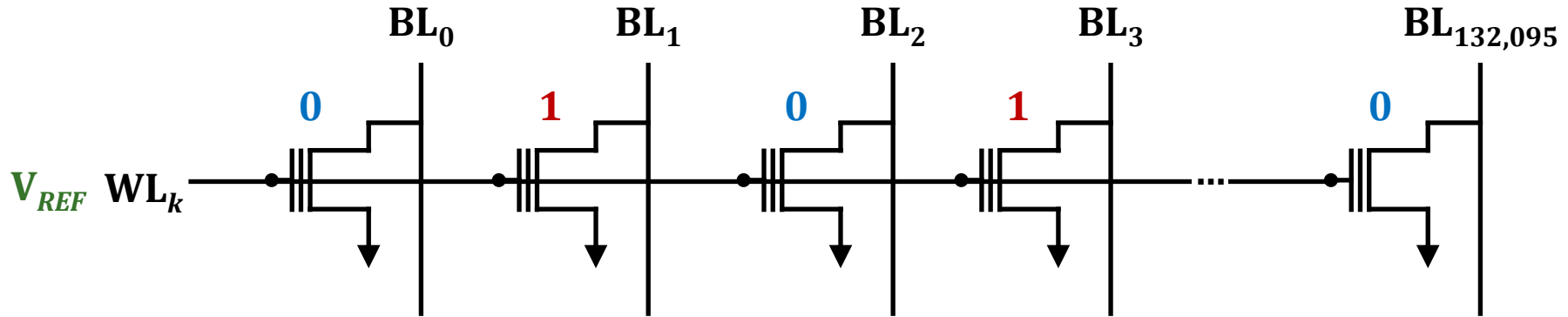
0 Programmed

Threshold voltage (V_{TH})



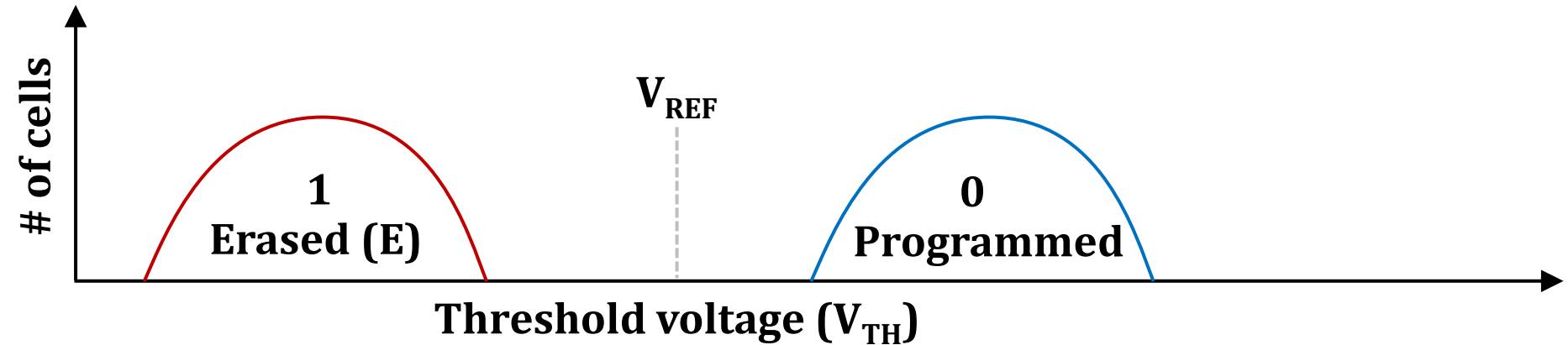
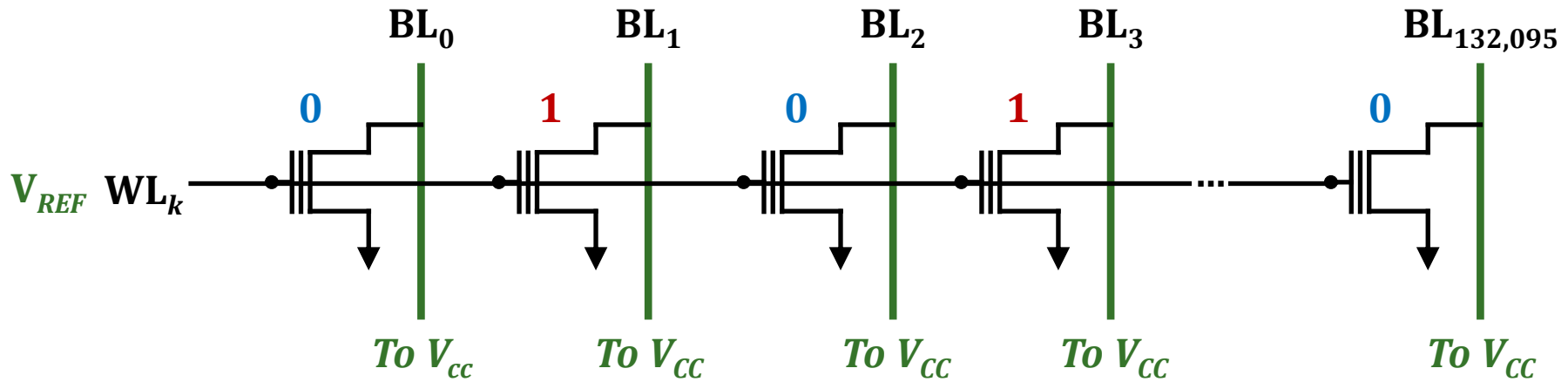
Basic Operation: Page Read

- WL control – All other cells operate as a resistance



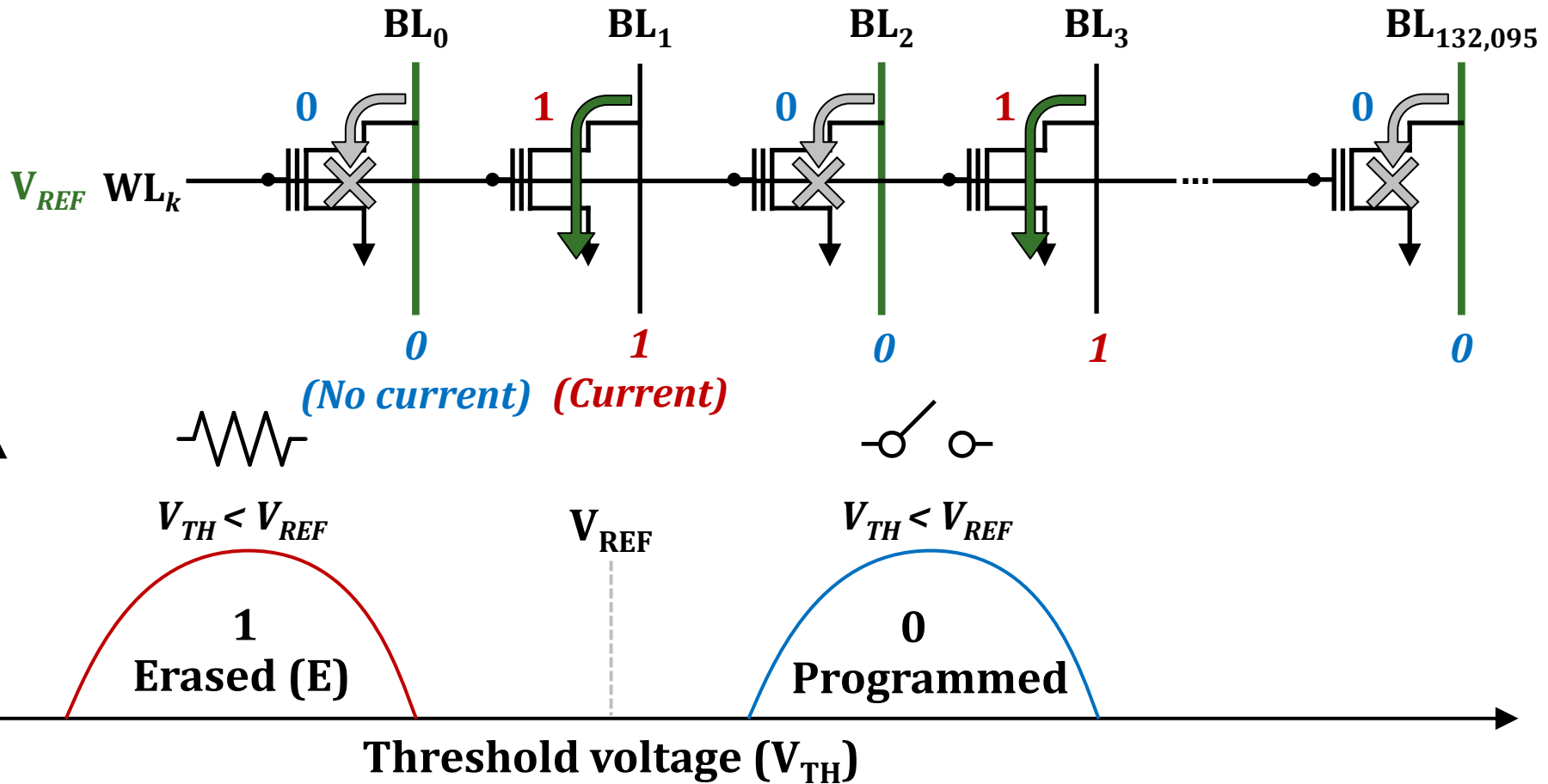
Basic Operation: Page Read

- BL control – Charge all BLs



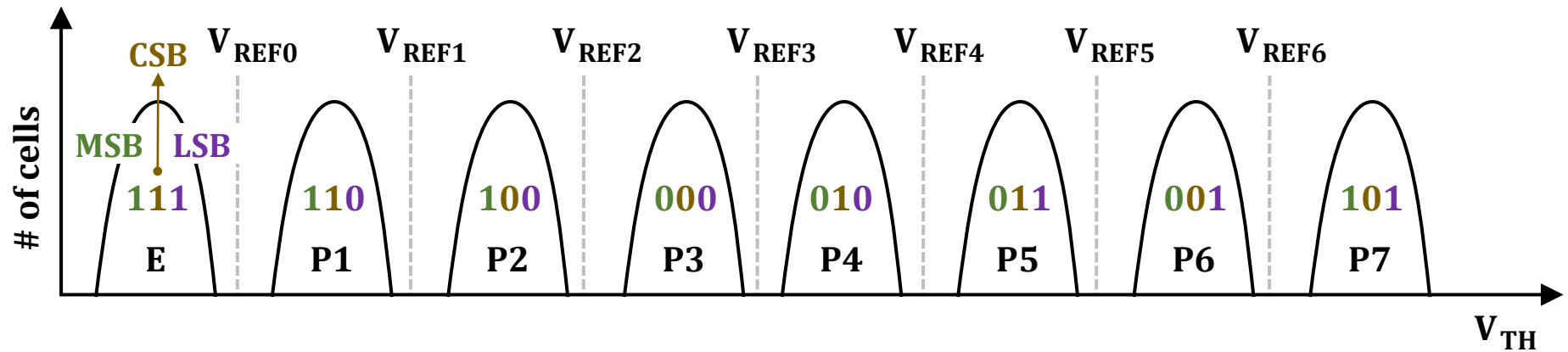
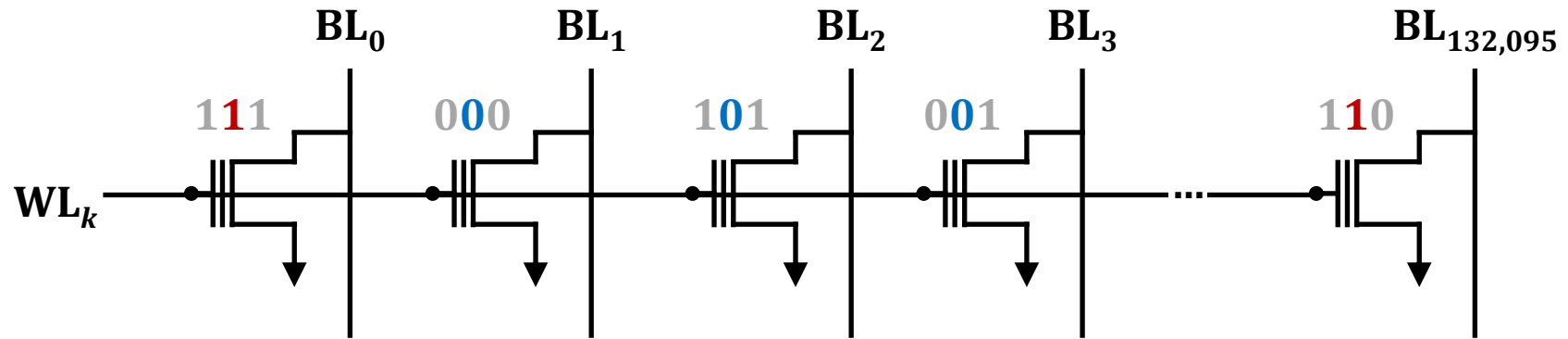
Basic Operation: Page Read

- Sensing the current through BLs



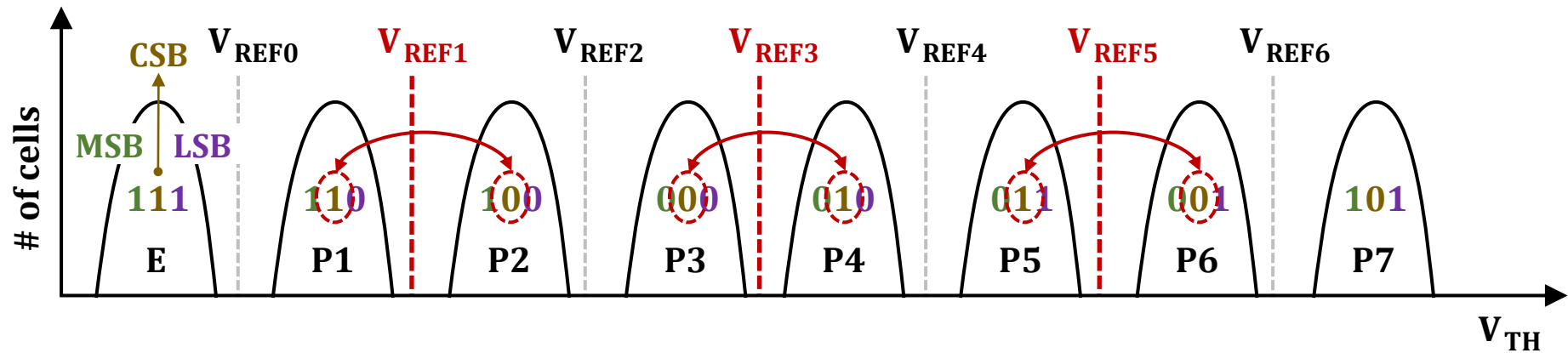
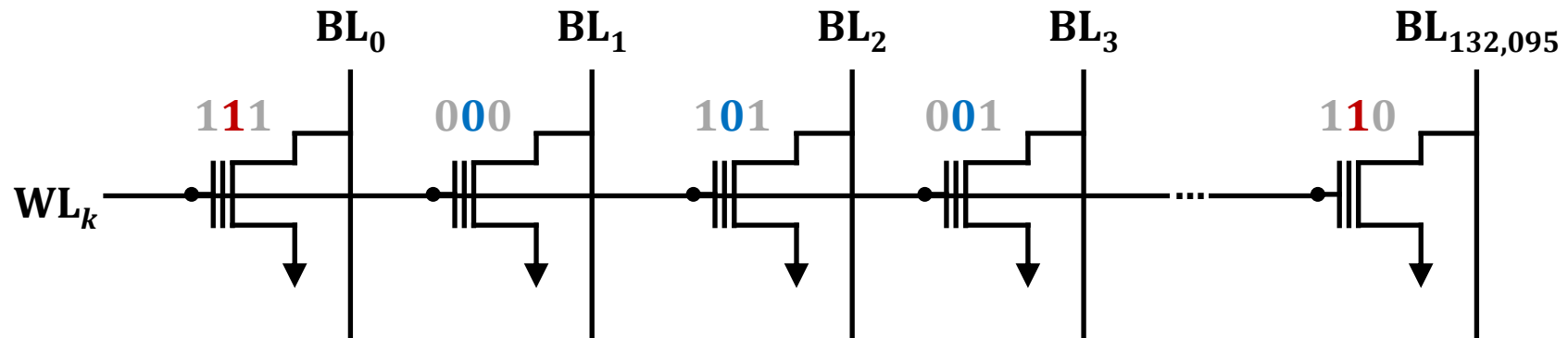
Basic Operation: Page Read - MLC

- Sensing the current through BLs



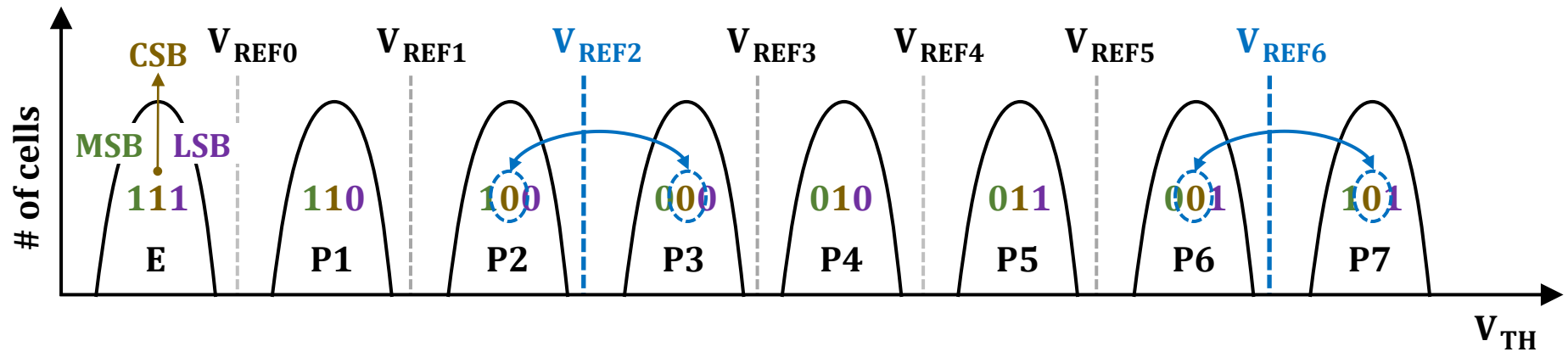
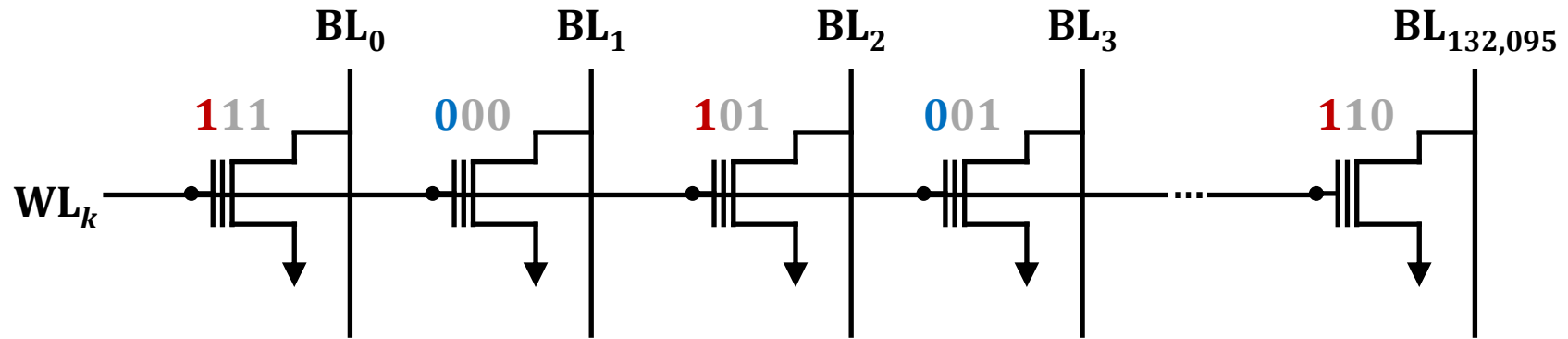
Basic Operation: Page Read - MLC

- Sensing the current through BLs



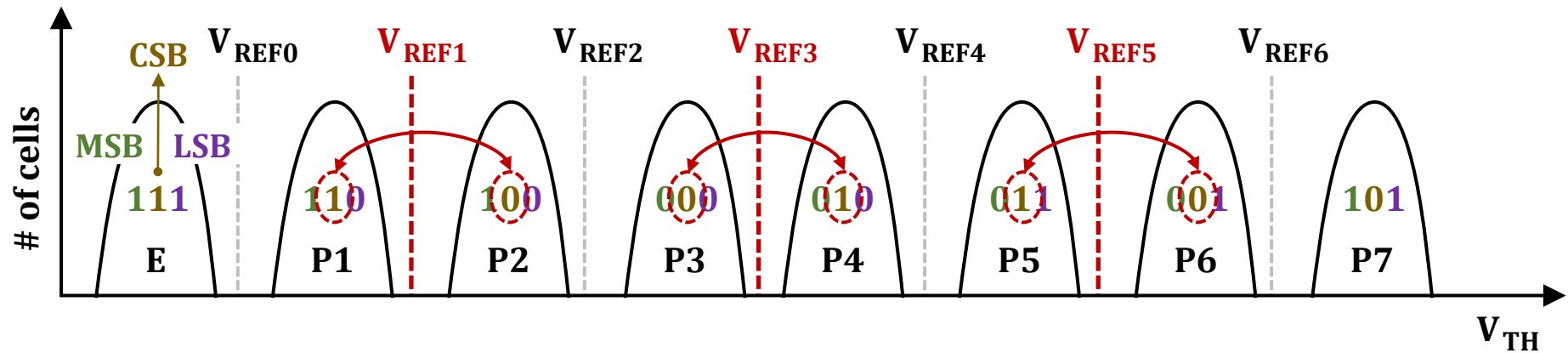
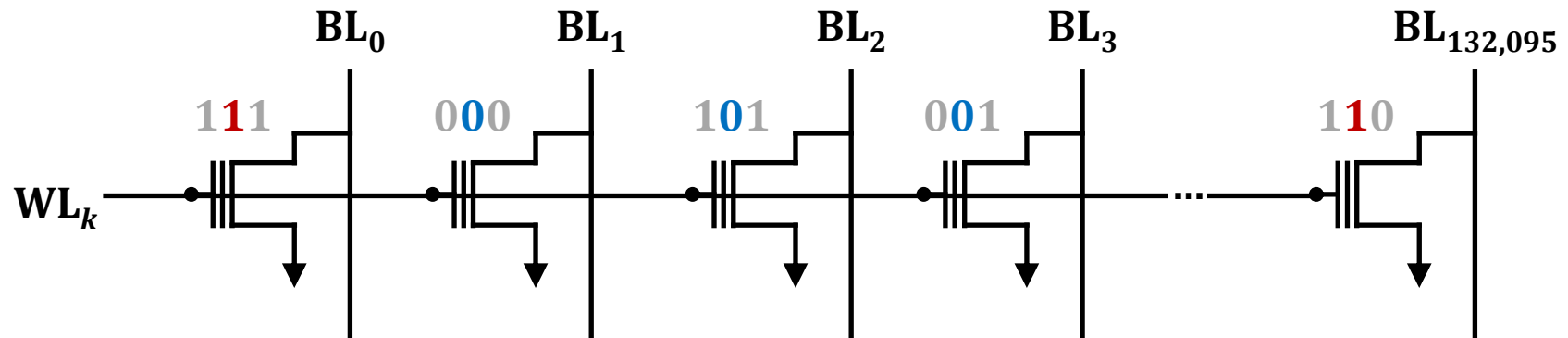
Basic Operation: Page Read - MLC

- Sensing the current through BLs



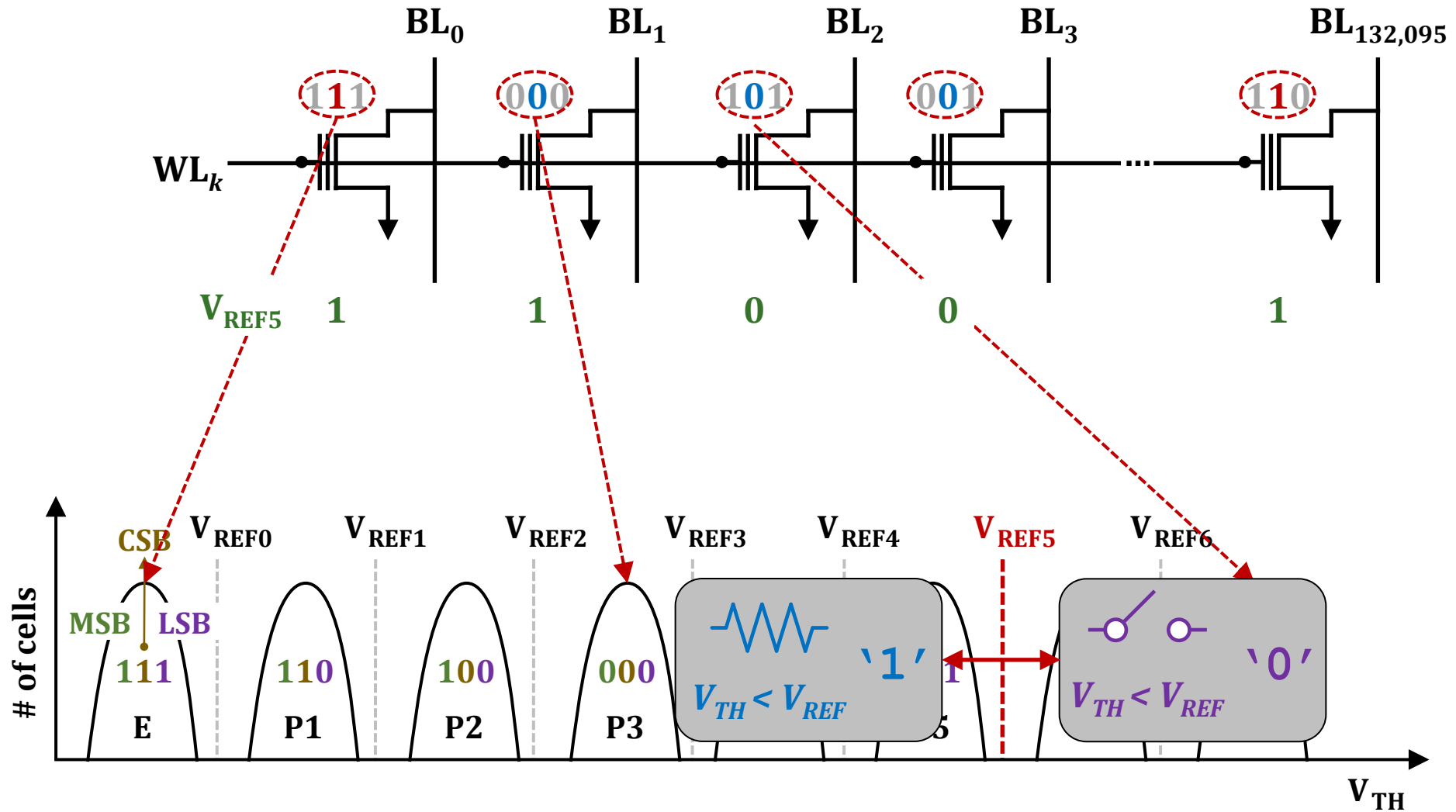
Basic Operation: Page Read - MLC

- Sensing the current through BLs



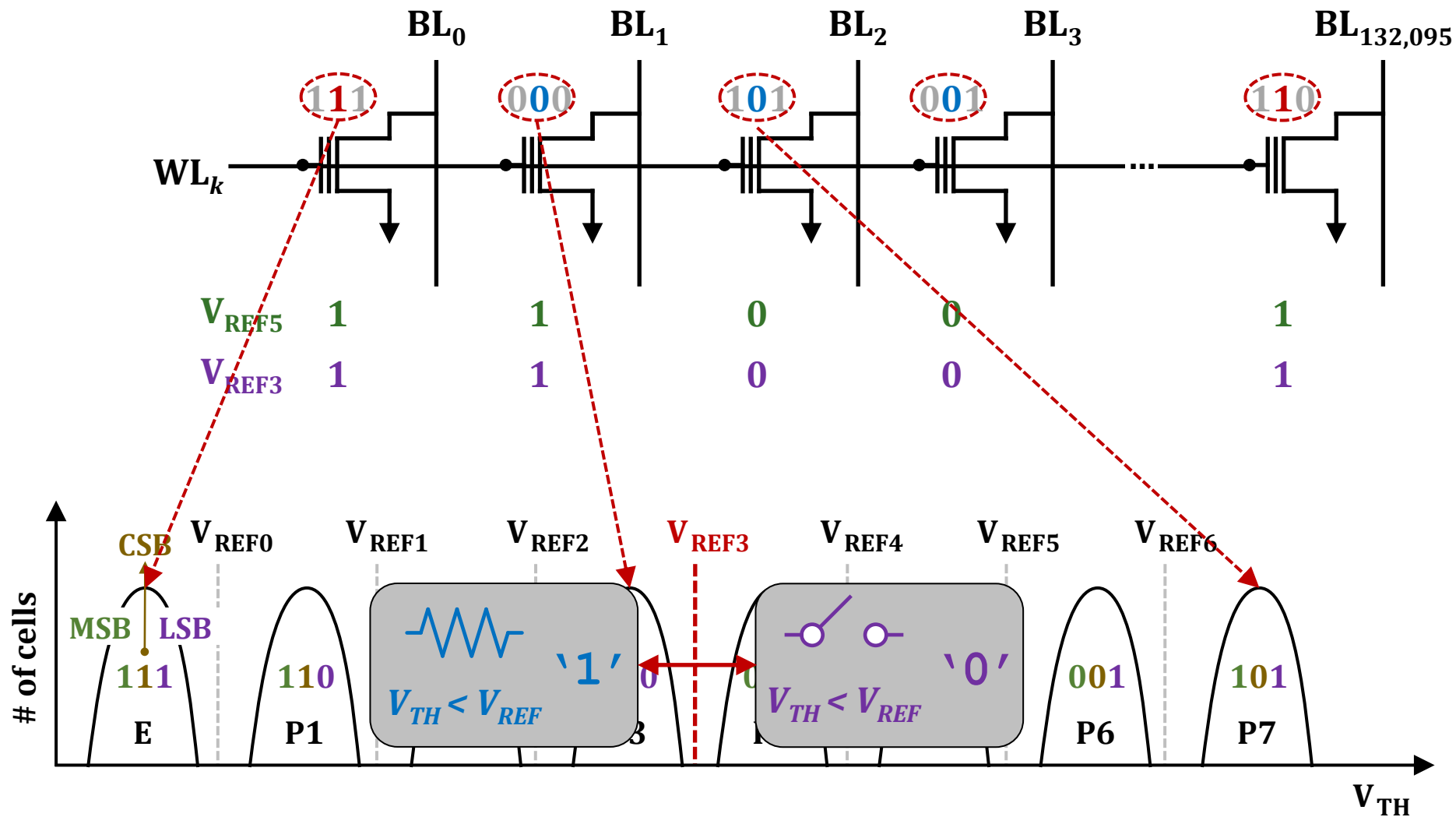
Basic Operation: Page Read - MLC

- Sensing the current through BLs



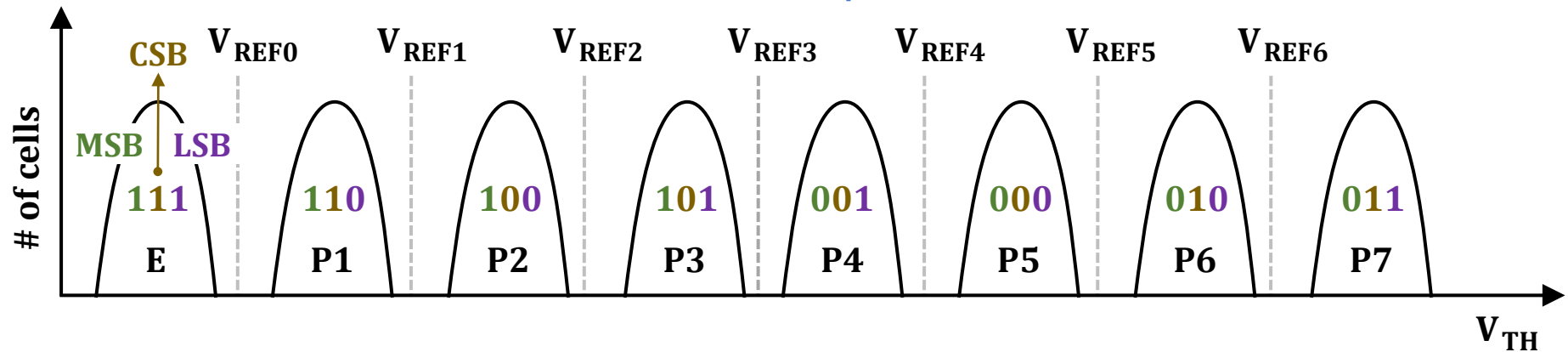
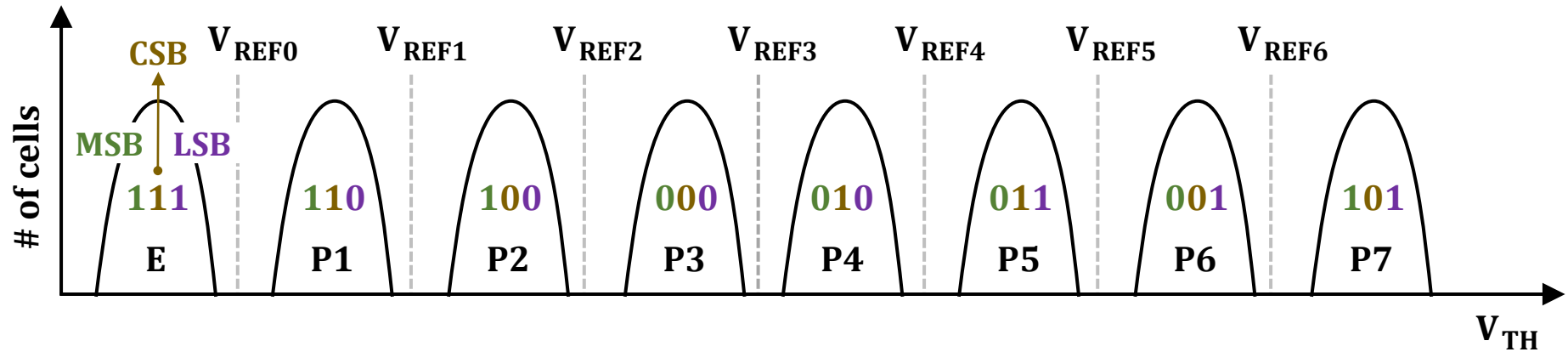
Basic Operation: Page Read - MLC

- Sensing the current through BLs



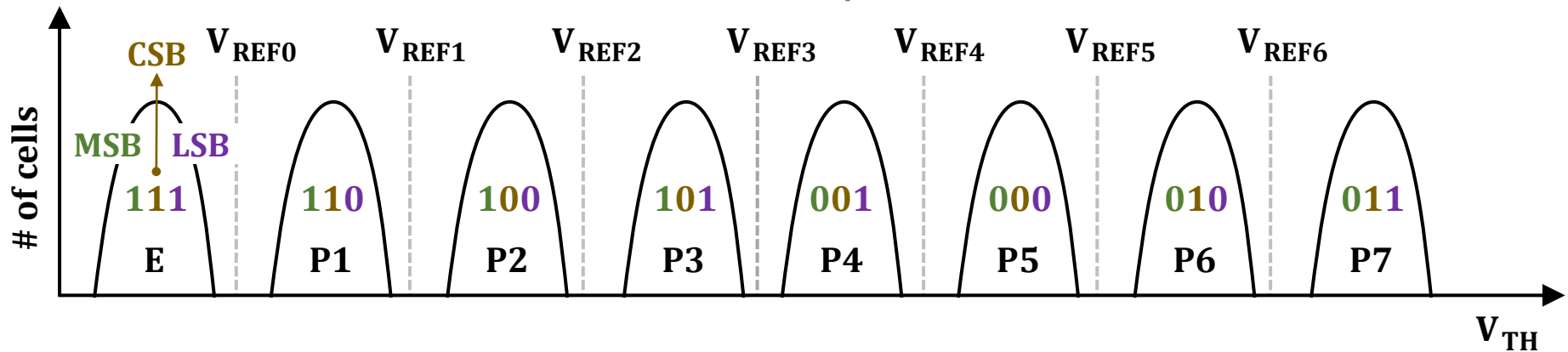
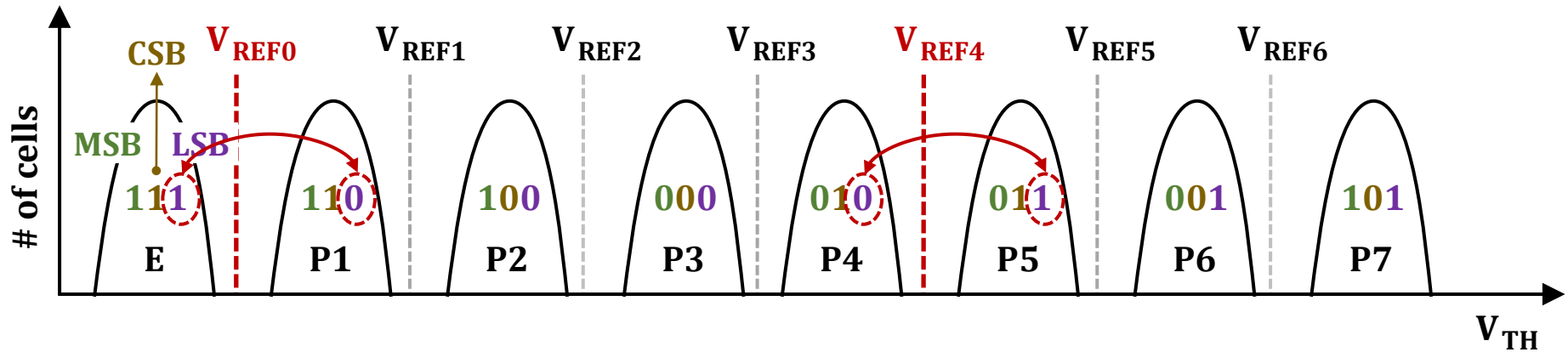
Basic Operation: Page Read – Takeaways

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



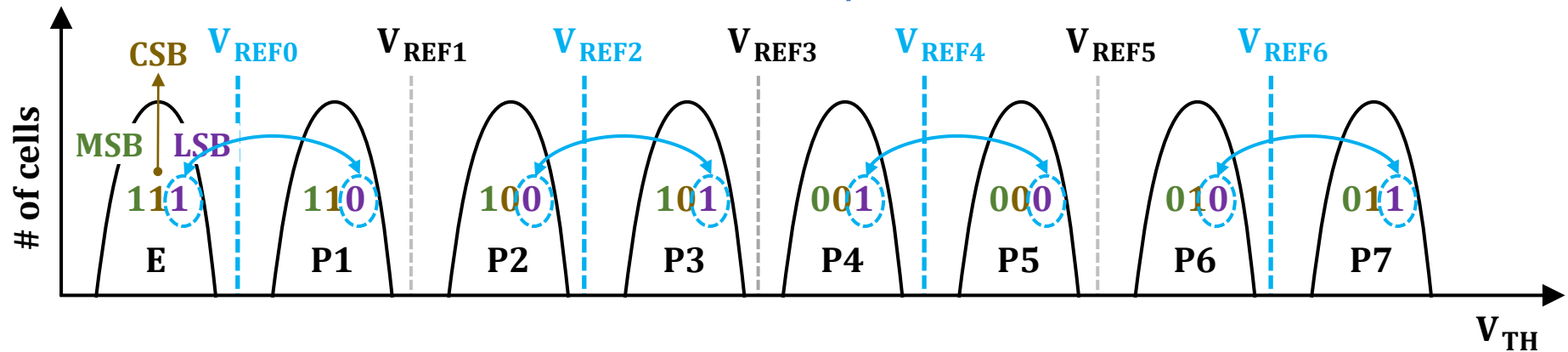
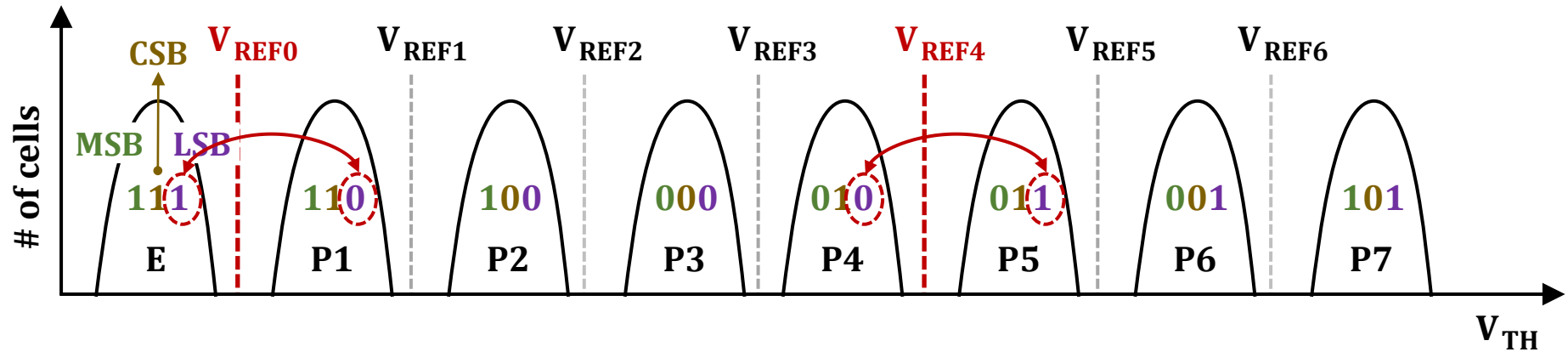
Basic Operation: Page Read – Takeaways

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



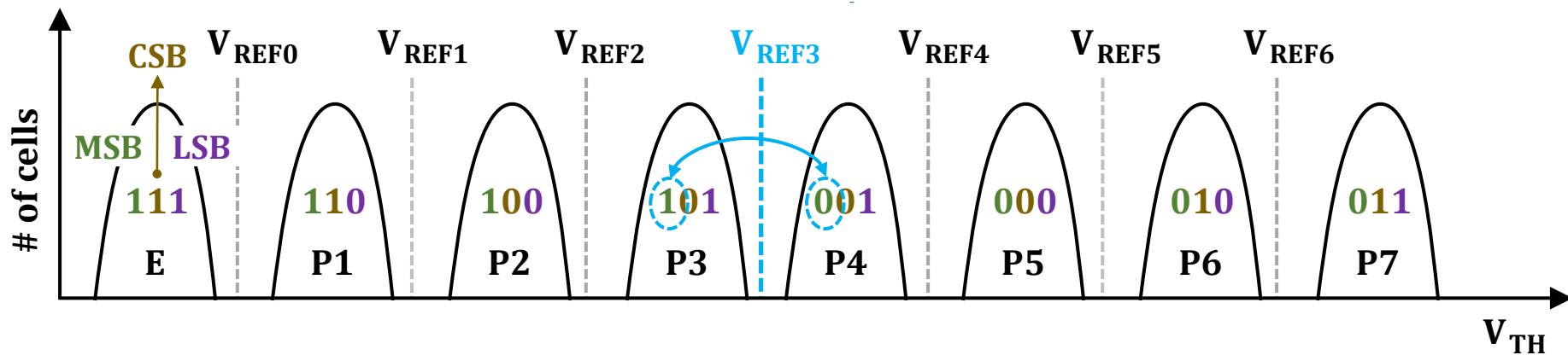
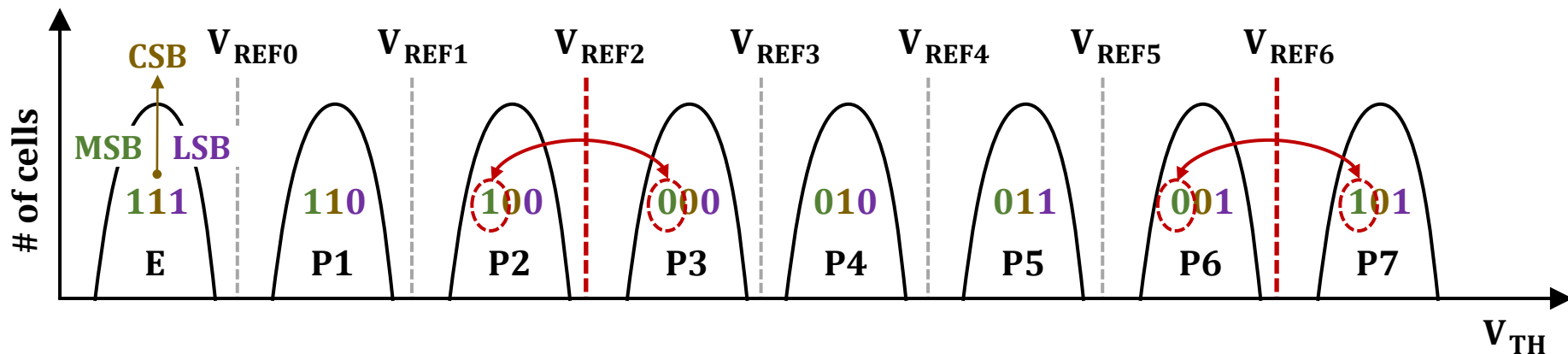
Basic Operation: Page Read – Takeaways

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



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

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