Addressing Fast-Detrapping for Reliable 3D NAND Flash Design

Mustafa M. Shihab - The University of Texas at Dallas
Jie Zhang - Yonsei University
Myoungsoo Jung - KAIST
Mahmut Kandemir - Pennsylvania State University

UT DALLAS | Erik Jonsson School of Engineering and Computer Science

CAMELab.org
Computer Architecture and Memory Systems Laboratory

PennState College of Engineering

Outline

- Background
  - Paradigm shift from 2D to 3D
  - Floating-gate vs. Charge-trap Flash
  - 3D NAND fabrication
- Problem/Challenge
  - Fast-detrapping in CT Flash
  - Impact of fast-detrapping on 3D NAND flash
- Contributions
  - Analytic model for fast-detrapping
  - Counter-Mechanisms
    - Investigating a fast-drift aware V_{Ref} mechanism
    - Exploiting page organization to support stronger ECC
    - Using Reinforcement-Learning for efficient charge-refill
- Experimental Results

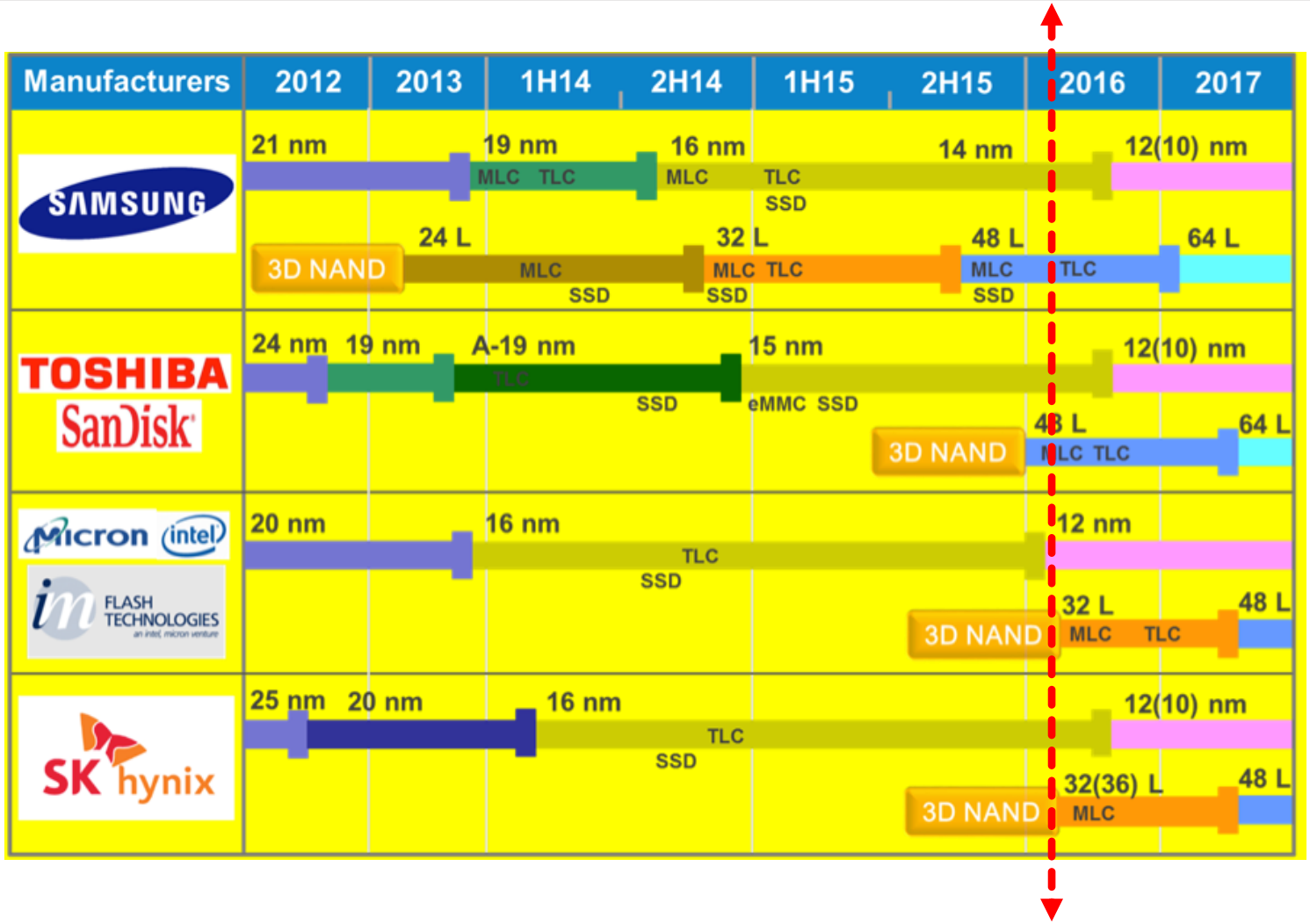
NAND Flash Paradigm Shift: From 2D To 3D

- For the last two decades, NAND flash is changing the perception of data storage
  - Diverse and successful incarnations as the preferred storage medium
    - From low-power mobile devices to high-performance computing

- There is a continuous demand for larger storage capacity and scalability has become a critical limitation for the planar NAND flash design
  - Insufficient number of electrons in the substrate
  - Excessive cell-to-cell interference
  - Prohibitively expensive fabrication process

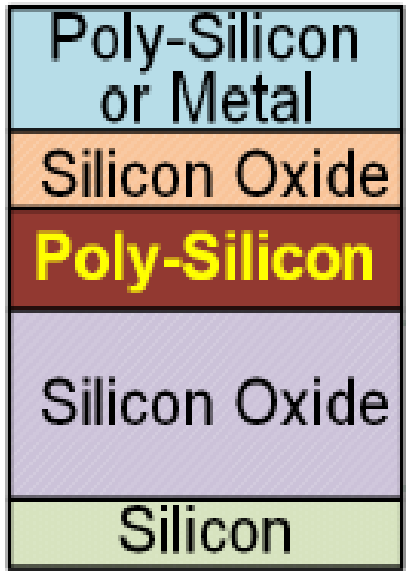
*Designers proposed to **vertically stack** the flash cells and expand storage capacity by constructing a three-dimensional NAND flash array*

NAND Flash Paradigm Shift: From 2D To 3D



Source: Comparison 1Y nanometer NAND architecture and beyond. SolidState Technology. 2015.

All NAND Flash Cells Are Not Made Equal



Floating-Gate (FG) NAND Flash

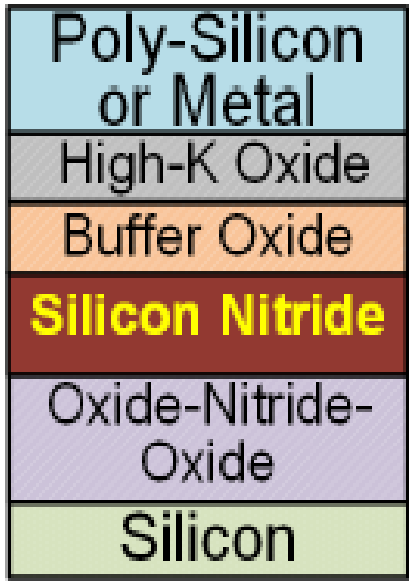
Control Gate

Gate Oxide

Charge Storage Layer

Tunnel Oxide

Channel



Charge-Trap (CT) NAND Flash

- A cell is divided into multiple layers -> charge storage layer (CSL) works as the storage core
- FG-flash has conducting poly-silicon CSL -> defect in the tunnel-oxide allows charge to leak out
  - Tunnel-oxide needs to be relatively thick
- CT-flash uses non-conductive silicone nitride CSL -> better tolerance to oxide defects
  - Can afford a thinner tunnel-oxide, but relatively expensive/difficult to fabricate

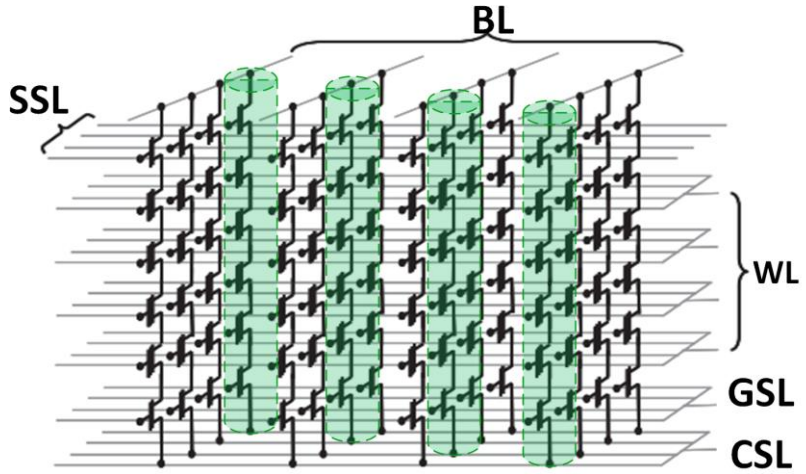
All NAND Flash Cells Are Not Made Equal

Floating-Gate cells were the predominant choice for conventional 2D NAND Flash,

But what about the 3D NANDs?

3D NAND Flash Architecture

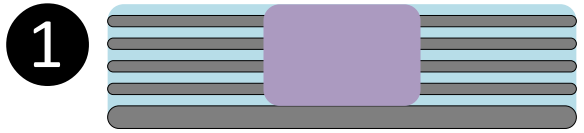
The **Terabit cell array transistor (TCAT)** is a popular 3D NAND flash design choice, and the first to be implemented in consumer products



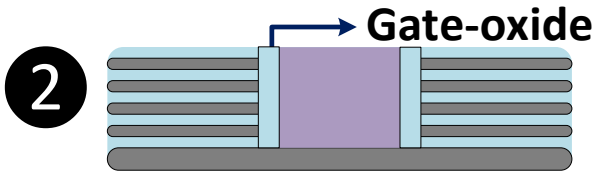
- Flash cells are vertically fabricated in cylindrical shapes known as *strings*
- Storage capacity can be increased by stacking more *layers*
- At each layer, cells are organized into *rows* and *columns*
  - *Wordlines* (WL) and *bitlines* (BL) connects all the cells in a row and a column, respectively
  - *String selec*t (SSL), *drain select* (DSL) and *ground select* (GSL) lines connect to the peripheral network

Fabrication Process for 3D NAND Flash

Interleaved layers of **oxide** and **polysilicon** are deposited on the Si substrate, and a hole is etched to the top of the substrate



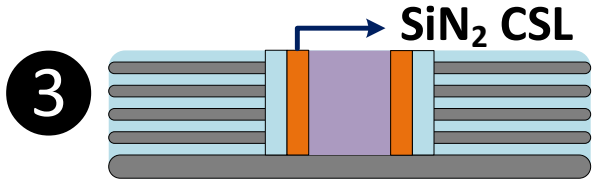
Horizontal Deposition and Etching



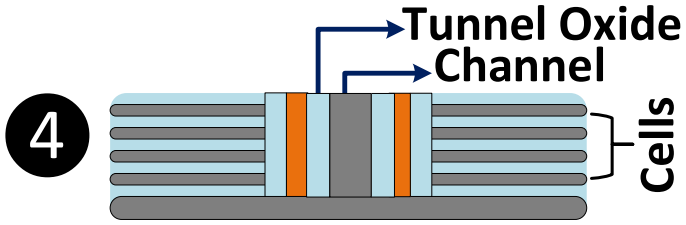
Vertical Gate-Ox Deposition

The wall of the hole is deposited with **gate-oxide**

The wall is then deposited with a layer of **silicon nitride**



Vertical CSL Deposition



Vertical Tunnel-Ox Deposition and channel fill up

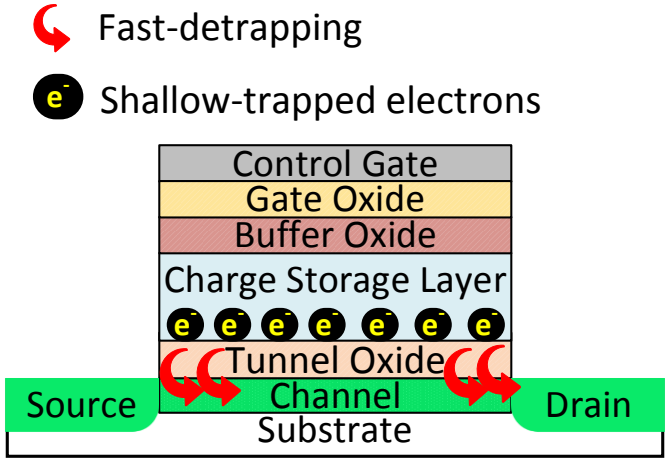
The **tunnel-oxide** is deposited on the nitride layer, and the remaining space in the hole is filled with **polysilicon channel**

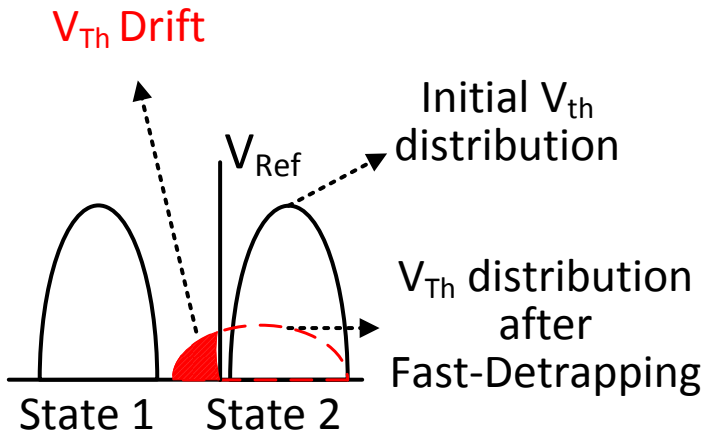
3D NAND's Choice of Flash Cell Type

- FG-flash requires the CSLs of the adjacent cells to be kept isolated
- CSLs in 3D NAND are deposited vertically - *like coats of paint* (❷, ❸, ❹)
  - Horizontal etching + deposition at each layer of each string is impractical
- CSLs of CT-flash does not require such CSL isolation

Most 3D NAND designs replaced FG-flash with CT-flash for a simplified and efficient fabrication process

Fast-Detrapping in CT NAND Flash



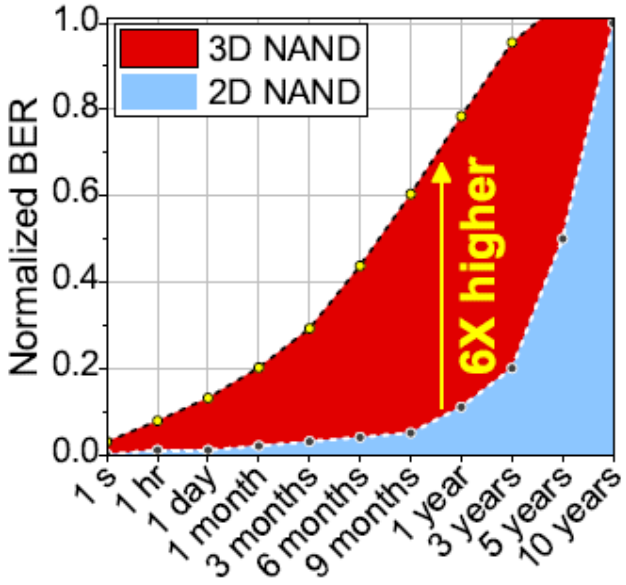


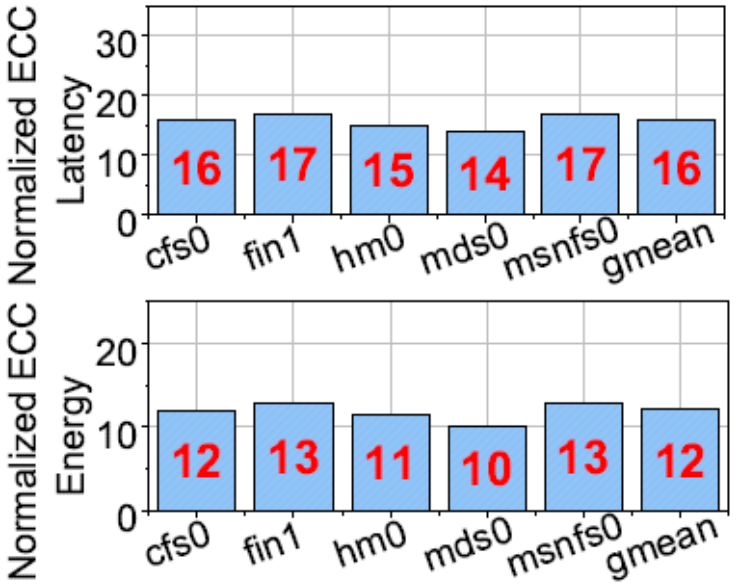
- Since the CSL is an insulator, during a program operation -
  - Not all injected electrons are plunged deep inside it
  - Large fraction of the electrons are shallowly trapped along the tunnel oxide-CSL boundary

- The **shallow-trapped** electrons can escape or **detrap** from the CSL soon after a program
  - Causes the threshold voltage (V_{Th}) to drift – commonly known as ***fast (threshold) drift***

The V_{Th} drift can spread beyond the threshold reference voltage (V_{Ref}) and generate error

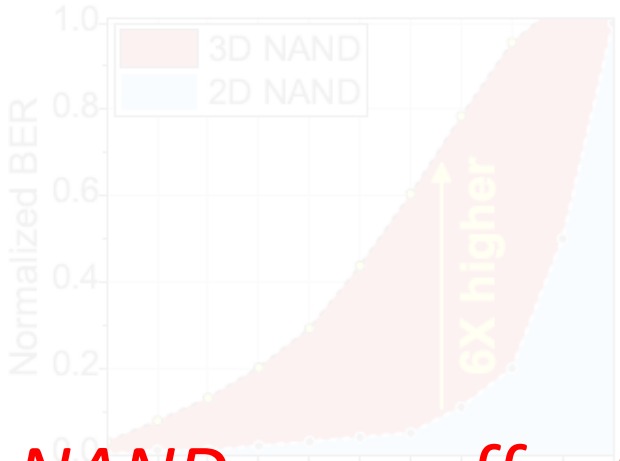
Impact Of Fast-Drift On 3D NAND Flash





- 2D NAND starts to suffer from high BER only near the end of its retention period
- But 3D NAND can experience around 70% of the peak BER only months after a program
  - Because of a sharp drift in V_{Th} soon after a program, due to fast- detrapping of charges
- Natural response could be to employ a stronger error-correcting code (ECC) scheme
  - Unfortunately, ECC overheads increase super-linearly with error rate
  - Compared to 2D NAND latency and energy can be 16X and 12X higher, respectively

Impact Of Fast-Drift On 3D NAND Flash

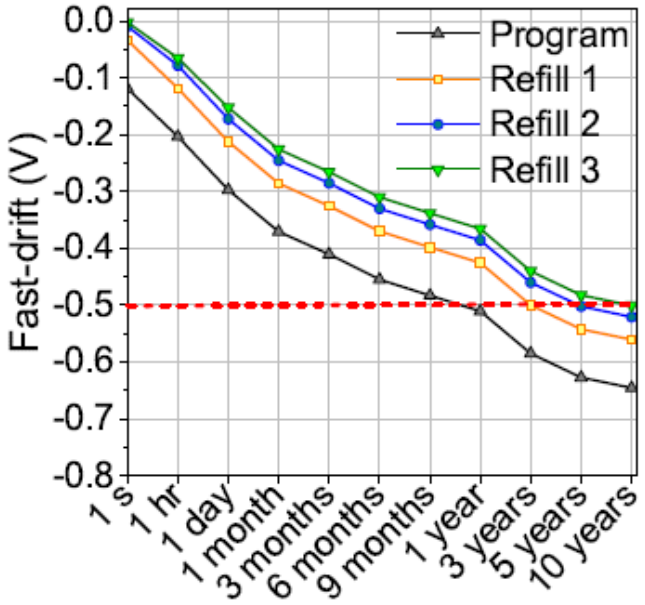


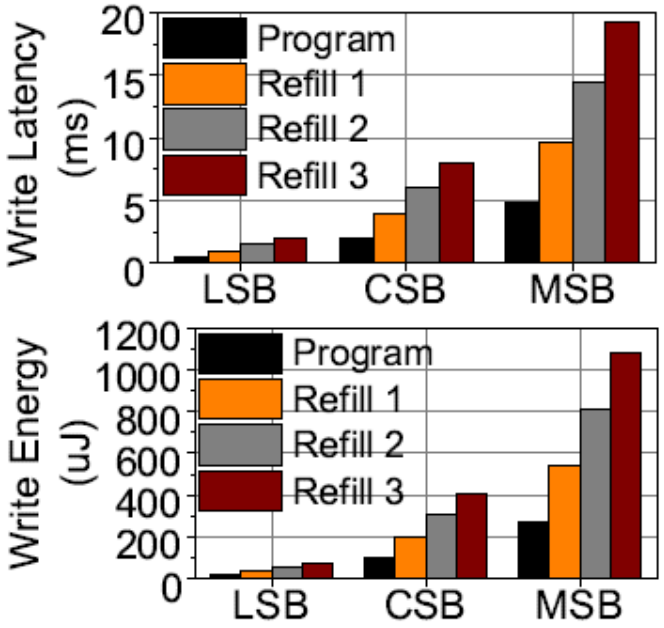
While 3D NAND can suffer from severe reliability problems without effective measures against fast-detrapping,
A brute-force attempt to correct the errors can also hurt the system

- 2D NAND starts to suffer from high BER only near the end of its retention period
- But 3D NAND can experience around 70% of the peak BER only months after a program
  - Because of a sharp drift in V_{Th} soon after a program, due to fast- detrapping of charges
- Natural response could be to employ a stronger error-correcting code (ECC) scheme
- Unfortunately, the ECC overheads increase super-linearly with error rate
  - The latency and energy overhead can be 16X and 12X higher, respectively

Charge-Refill: Benefit vs. Cost

Repeated in-place programming on CT-flash cells can refill the depleted charge and gradually diminish the impact of fast-drift





- Array-level simulation results confirm that, three extra charge-refill operations after a write can slow-down fast-drift sufficiently to ensure storage-class data retention

- Refill operations exceedingly amplify the overheads of each program operation
  - For TLC NAND flash, the latency and energy can increase by up to 9X and 15X, respectively

Charge-Refill: Benefit vs. Cost

Naively scheduling refill operations in 3D NAND can render it impractical for high-performance and low-power applications

But first, we need a mechanism to estimate/evaluate the impact of fast-drift on 3D NAND

Analytic Model for Fast-Drift

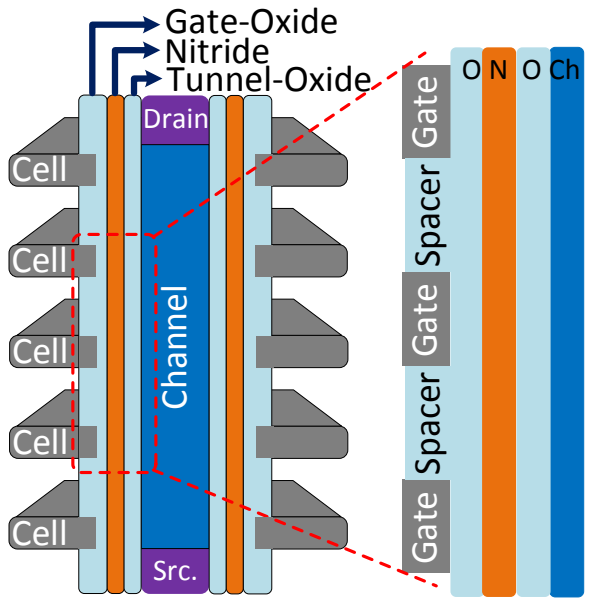
- Initiation and magnitude of fast-drift co-depend on certain design parameters and environmental conditions
- Leveraging the empirical data from prior work, we have developed the first publicly available analytic model to characterize fast-drift:

$$\Delta V_{Th}(t) = -[log(t) + \alpha][\frac{V_{Th,Init}}{\beta} + \theta\Delta T + \frac{\delta}{t_{buff-ox}}][\frac{1}{R+1}]$$

ΔV_{Th}	= Amount of fast-drift
T	= Elapsed time after a write
$V_{Th,\ Init}$	= Initially programmed V_{Th}
ΔT	= Operating temperature – Ideal room temperature
$t_{buff-ox}$	= thickness of the buffer-oxide
R	= Refill count
α, β, θ and δ	= Fitting constants

Extending the Model for 3D NAND Flash

- With shared oxide and CSL, 3D NAND can allow higher number of shallow-trapped electrons
  - The shared surface area in 3D-NAND increases with the additional stacked-layers
- 3D NAND flash cell's retention is affected by the inclusion of an immediate neighbor (layer), and is independent of other layers
- For a fixed programming voltage, fast-drift increases linearly



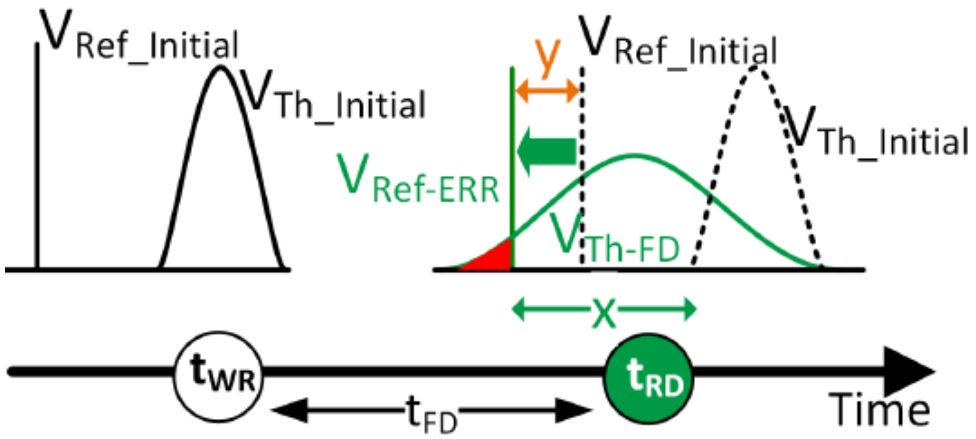
Impact of fast-drift is more critical for 3D NAND:

$$\Delta V_{Th-3D} = \left(1 + \frac{P}{100}(n-1)\right)\Delta V_{Th-Cell}$$

P	= % increase in fast-drift for each stacked layer
n	= Number of layers
$\Delta V_{Th-Cell}$	= Fast-drift for a single CT-flash cell

Countermeasure 1: Elastic Read Reference voltage (ERR)

- Fast-drift varies with the elapsed time between writing and reading a page
  - If the V_{Ref} is also adjusted proportionally, we can correctly read the affected

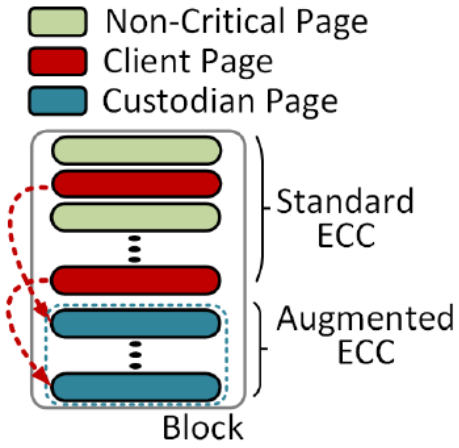


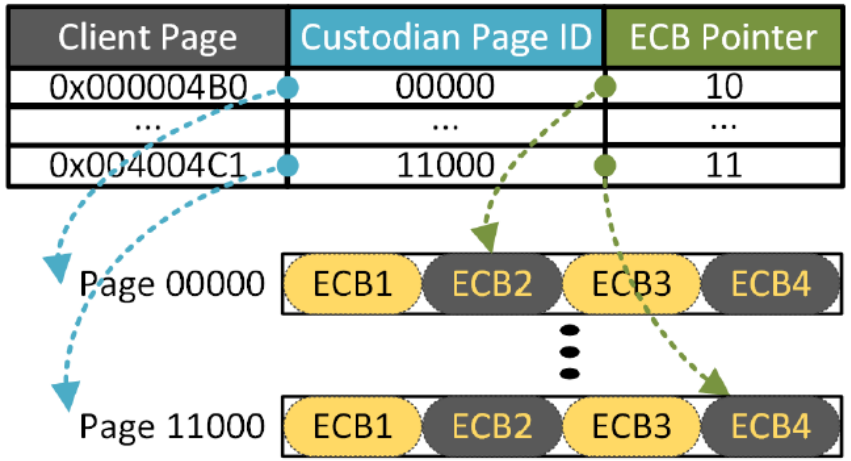
Time-bin for Est. Fast-drift (days)	ERR Assigned V_{Ref-FD} (V) [50°C]	$\|y\| = V_{Ref-FD} - V_{Ref-Init}$
$t_{FD} < 0.01$	$\|x\| = 0.425$	0.000
$0.01 < t_{FD} < 1$	$\|x\| = 0.45$	0.025
$1 < t_{FD} < 100$	$\|x\| = 0.475$	0.050
$t_{FD} > 100$	$\|x\| = 0.5$	0.075

- ERR *timebins* a set of V_{Ref}s, and dynamically assign one to each page read - based on the time that page was last written
  - Flash controller marks the time of a read request as the read-time (t_{RD}).
  - The time-stamp for the latest write on that page is set as the write-time (t_{WR}).
  - Effective elapsed time for fast-drift (t_{FD}) = $t_{RD} - t_{WR}$
  - Fast-drift is estimated using the analytic model and a suitable VRef –FD is assigned

Countermeasure 2: Hitch-Hike

- Hitch-Hike can provide a stronger ECC to the error-prone 3D NAND flash

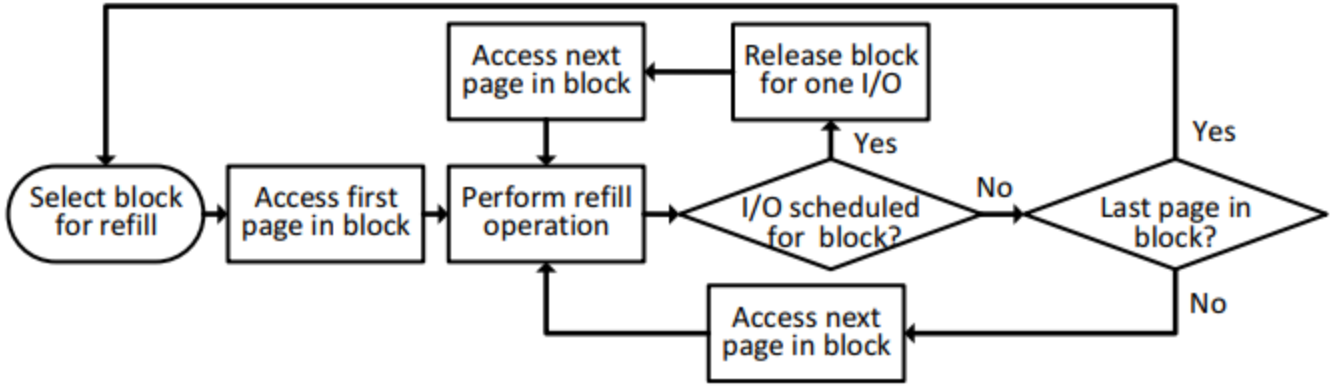




- Most pages in a block are for regular data storage and are encoded with the regular ECC
- A fraction of the pages are set as custodian pages for storing the error correction bits (ECB)
- When a page retains data for a prolonged period and is expected to be vulnerable to fast-drift:
  - Hitch-hike controller marks them as client pages
  - Client pages are read in the background and encoded using an augmented ECC codec
  - The ECB for this enhanced ECC encoding is stored in a custodian page
  - When a read is assigned for that client page, the controller accesses both the client page and its corresponding custodian page, and decodes the data using the stored ECB

Countermeasure 3: iRefill

- Intelligent charge-refill scheme that leverages reinforcement-learning to reduce the number of refills, which in turn can allow 3D NAND to attain storage-class retention with minimum overhead

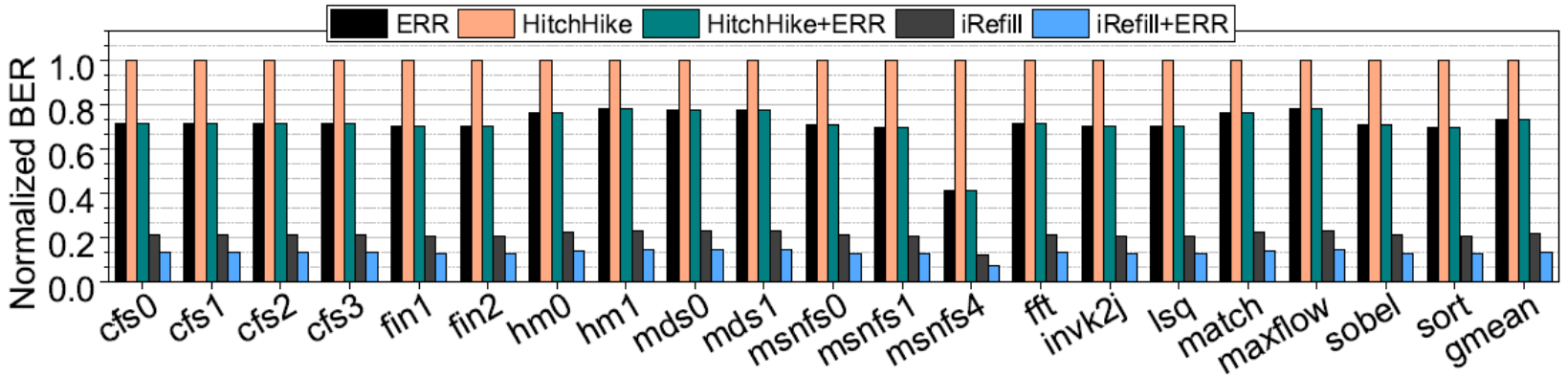


- Controller collects state and reward information from 3D NAND, and assigns an action for the next state
  - *State functions: current refill count, elapsed time since last write/refill, and current BER*
  - *Action functions: assigning a refill operation or, continuing with the regular operations*
  - *Immediate reward: maintain the BER permitted by the ECC scheme*
  - *Long-term reward: minimize the refill frequency and maximize I/O throughput*
- iRefill schedules refill operations at a block-level granularity to minimize potential resource overheads
- If regular I/O occurs while refilling a block, iRefill interleaves refills and I/Os

Evaluation Setup

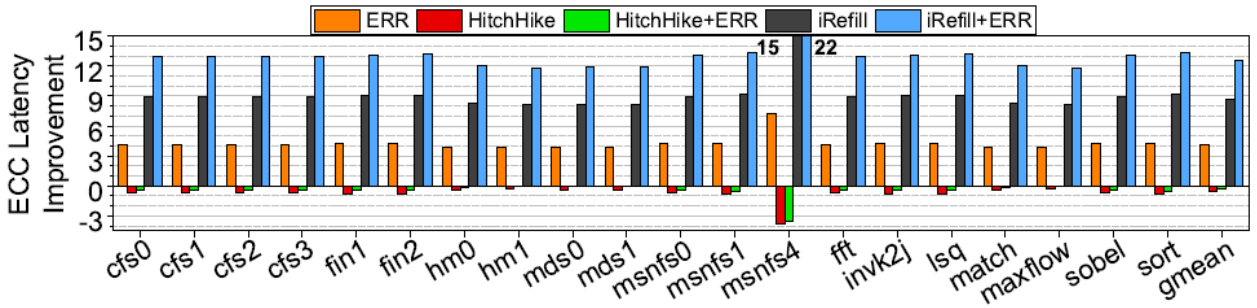
- Designed an in-house simulator based on the proposed fast-drift model
- Simulated raw BER for:
  - 256GB 3D NAND flash
  - 40 nmprocess technology
  - Maximum operating temperature of 70◦C
- Considered various configurations executing a wide range of real-life workload traces
- Calculated corresponding ECC latency and energy overheads for a 2.0 bit LDPC scheme

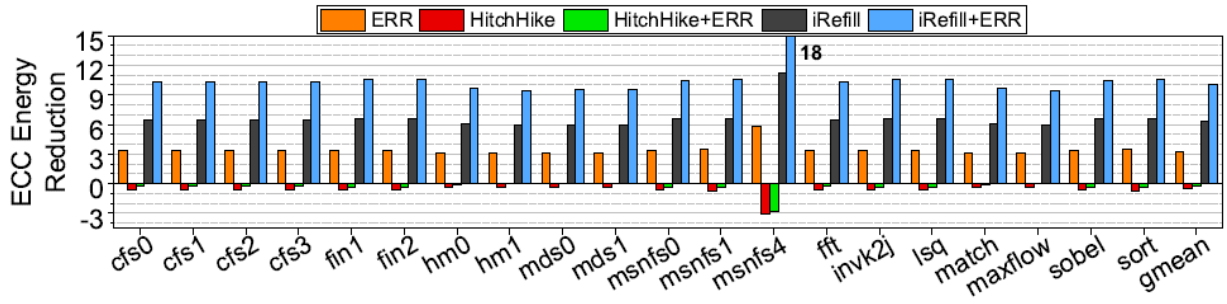
Evaluation Results – BER Reduction



- `ERR` attains an average BER improvement of 26% over the Baseline
- `HitchHike` and `HitchHike+ERR` do not show additional BER reduction, since the hitch-hike scheme is not designed to reduce error, but to correct more of them
- `iRefill` attains a significant improvement of 78% over the Baseline, on average
- `iRefill+ERR` demonstrates the optimum reliability rating with an average BER improvement of 87%
  - Combined impact of reducing fast-drift through iRefill, and correcting more errors with ERR, allows to achieve excellent reliability

Evaluation Results – ECC Latency and Power





- ECC overhead is proportional to the number of errors experienced by the system
  - Reducing BER can significantly lower the ECC latency and power consumption

- With the lowest number of error bits to correct among all the configurations, `iRefill+ERR` produces a 13X latency improvement over the Baseline, on average

- The combined effort also reduces the 3D NAND's average ECC energy consumption by 10X

Outline

- Background
  - Paradigm shift from 2D to 3D
  - Floating-gate vs. Charge-trap Flash
  - 3D NAND fabrication
- Problem/Challenge
  - Fast-detrapping in CT Flash
  - Impact of fast-detrapping on 3D NAND flash
- Contributions
  - Analytic model for fast-detrapping
  - Counter-Mechanisms
    - Investigating a fast-drift aware V_{Ref} mechanism
    - Exploiting page organization to support stronger ECC
    - Using Reinforcement-Learning for efficient charge-refill
- Experimental Results

References

[1] Chih-Ping Chen et al. 2010. Study of fast initial charge loss and it's impact on the programmed states Vt distribution of charge-trapping NAND Flash. In IEEE IEDM.

[2] Bongsik Choi et al. 2016. Comprehensive evaluation of early retention characteristics in tube-type 3-D NAND Flash memory. In IEEE VLSI Technology.

[3] Laura M Grupp, John D Davis, and Steven Swanson. 2012. The bleak future of NAND flash memory. In USENIX FAST.

[4] Jaehoon Jang et al. 2009. Vertical cell array using TCAT technology for ultra high density NAND flash memory. In IEEE VLSI Technology.

[5] Jonghong Kim et al. 2012. Low-energy error correction of NAND Flash memory through soft-decision decoding. EURASIP JASP 1 (2012), 195.

[6] Xinkai Li et al. 2014. Investigation of charge loss mechanisms in 3D TANOS cylindrical junction-less charge trapping memory. In IEEE ICSICT.

[7] HT Lue et al. 2005. Novel soft erase and re-fill methods for a P+ poly gate nitride trapping NVM device with excellent endurance and retention properties. In IRPS.

[8] Dushyanth Narayanan et al. 2009. Migrating server storage to SSDs: analysis of tradeoffs. In ACM ECCS.

[9] Ki-Tae Park et al. 2014. Three-dimensional 128Gb MLC vertical NAND flash-memory with 24-WL stacked layers and 50MB/s high-speed programming. In IEEE ISSCC.

[10] Mustafa M. Shihab et al. 2018. ReveNAND: A fast-drift-aware resilient 3d NAND flash design. ACM Transactions on Architecture and Code Optimization 15, 2 (2018), 17.

[11] Richard S Sutton and Andrew G Barto. 1998. Reinforcement learning: An introduction. MIT Press Cambridge.

[12] SungJin Whang et al. 2010. Novel 3-dimensional Dual Control-gate with Surrounding Floating-gate (DC-SF) NAND flash cell for 1Tb file storage application. In IEEE IEDM.

[13] Doe Hyun Yoon and Mattan Erez. 2010. Virtualized and flexible ECC for main memory. In ACM SIGARCH Computer Architecture News, Vol. 38. 397–408.

Thank You!