Error Analysis and Management for MLC NAND Flash Memory

Onur Mutlu
onur@cmu.edu

(joint work with Yu Cai, Gulay Yalcin, Erich Haratsch, Ken Mai, Adrian Cristal, Osman Unsal)

August 7, 2014
Flash Memory Summit 2014, Santa Clara, CA
Executive Summary

- Problem: MLC NAND flash memory reliability/endurance is a key challenge for satisfying future storage systems’ requirements.

- Our Goals: (1) Build reliable error models for NAND flash memory via experimental characterization, (2) Develop efficient techniques to improve reliability and endurance.

- This talk provides a “flash” summary of our recent results published in the past 3 years:
  - Experimental error and threshold voltage characterization [DATE’12&13]
  - Retention-aware error management [ICCD’12]
  - Program interference analysis and read reference V prediction [ICCD’13]
  - Neighbor-assisted error correction [SIGMETRICS’14]
Agenda

- Background, Motivation and Approach
- Experimental Characterization Methodology
- Error Analysis and Management
  - Characterization Results
  - Retention-Aware Error Management
  - Threshold Voltage and Program Interference Analysis
  - Read Reference Voltage Prediction
  - Neighbor-Assisted Error Correction
- Summary
Evolution of NAND Flash Memory

- Flash memory is widening its range of applications
  - Portable consumer devices, laptop PCs and enterprise servers

Seaung Suk Lee, “Emerging Challenges in NAND Flash Technology”, Flash Summit 2011 (Hynix)
Flash Challenges: Reliability and Endurance

P/E cycles (required)
A few thousand

P/E cycles (provided)

Writing the full capacity of the drive 10 times per day for 5 years (STEC)

> 50k P/E cycles

E. Grochowski et al., “Future technology challenges for NAND flash and HDD products”, Flash Memory Summit 2012
NAND Flash Memory is Increasingly Noisy
Future NAND Flash-based Storage Architecture

Our Goals:
Build reliable error models for NAND flash memory
Design efficient reliability mechanisms based on the model
NAND Flash Error Model

Experimentally characterize and model dominant errors

- Neighbor page program (c-to-c interference)
- Retention


Cai et al., “Neighbor-Cell Assisted Error Correction in MLC NAND Flash Memories”, SIGMETRICS 2014


Cai et al., “Error Analysis and Retention-Aware Error Management for NAND Flash Memory, ITJ 2013"
Our Goals and Approach

Goals:
- Understand error mechanisms and develop reliable predictive models for MLC NAND flash memory errors
- Develop efficient error management techniques to mitigate errors and improve flash reliability and endurance

Approach:
- Solid experimental analyses of errors in real MLC NAND flash memory → drive the understanding and models
- Understanding, models and creativity → drive the new techniques
Agenda

- Background, Motivation and Approach
- **Experimental Characterization Methodology**
- Error Analysis and Management
  - Main Characterization Results
  - Retention-Aware Error Management
  - Threshold Voltage and Program Interference Analysis
  - Read Reference Voltage Prediction
  - Neighbor-Assisted Error Correction
- Summary
Experimental Testing Platform


Cai et al., FPGA-based Solid-State Drive prototyping platform, FCCM 2011.
NAND Flash Usage and Error Model

- Erase Errors
- Program Errors
- Retention Errors
- Read Errors

Start

P/E cycle 0

... 

P/E cycle i

... 

P/E cycle n

End of life

Erase Block

Program Page

Retention1 (t_1 days)

Read Page

Retention j (t_j days)

Read Page

(Page0 - Page128)
Methodology: Error and ECC Analysis

- Characterized errors and error rates of 3x and 2y-nm MLC NAND flash using an experimental FPGA-based platform
  - [Cai+, DATE’12, ICCD’12, DATE’13, ITJ’13, ICCD’13, SIGMETRICS’14]

- Quantified Raw Bit Error Rate (RBER) at a given P/E cycle
  - Raw Bit Error Rate: Fraction of erroneous bits without any correction

- Quantified error correction capability (and area and power consumption) of various BCH-code implementations
  - Identified how much RBER each code can tolerate
  - how many P/E cycles (flash lifetime) each code can sustain
NAND Flash Error Types

- Four types of errors [Cai+, DATE 2012]
  - Caused by common flash operations
    - Read errors
    - Erase errors
    - Program (interference) errors
  - Caused by flash cell losing charge over time
    - Retention errors
      - Whether an error happens depends on required retention time
      - Especially problematic in MLC flash because threshold voltage window to determine stored value is smaller
Agenda

- Background, Motivation and Approach
- Experimental Characterization Methodology
- Error Analysis and Management
  - Main Characterization Results
  - Retention-Aware Error Management
  - Threshold Voltage and Program Interference Analysis
  - Read Reference Voltage Prediction
  - Neighbor-Assisted Error Correction
- Summary
Observations: Flash Error Analysis

- Raw bit error rate increases exponentially with P/E cycles
- Retention errors are dominant (>99% for 1-year retention time)
- Retention errors increase with retention time requirement
Electron loss from the floating gate causes retention errors
- Cells with more programmed electrons suffer more from retention errors
- Threshold voltage is more likely to shift by one window than by multiple
Cells with more programmed electrons tend to suffer more from retention noise (i.e. 00 and 01)
Agenda

- Background, Motivation and Approach
- Experimental Characterization Methodology
- **Error Analysis and Management**
  - Main Characterization Results
  - *Retention-Aware Error Management*
  - Threshold Voltage and Program Interference Analysis
  - Read Reference Voltage Prediction
  - Neighbor-Assisted Error Correction
- Summary
Flash Correct-and-Refresh (FCR)

Key Observations:
- Retention errors are the dominant source of errors in flash memory [Cai+ DATE 2012][Tanakamaru+ ISSCC 2011] → limit flash lifetime as they increase over time
- Retention errors can be corrected by “refreshing” each flash page periodically

Key Idea:
- Periodically read each flash page,
- Correct its errors using “weak” ECC, and
- Either remap it to a new physical page or reprogram it in-place,
- Before the page accumulates more errors than ECC-correctable
- Optimization: Adapt refresh rate to endured P/E cycles

Cai et al., Flash Correct and Refresh, ICCD 2012.
FCR: Two Key Questions

- How to refresh?
  - **Remap** a page to another one
  - **Reprogram** a page (in-place)
  - **Hybrid** of remap and reprogram

- When to refresh?
  - **Fixed period**
  - **Adapt the period** to retention error severity
**In-Place Reprogramming of Flash Cells**

Floating Gate Voltage Distribution for each Stored Value

Retention errors are caused by cell voltage shifting to the left.

ISPP moves cell voltage to the right; fixes retention errors.

- **Pro:** No remapping needed → no additional erase operations
- **Con:** Increases the occurrence of program errors
Normalized Flash Memory Lifetime

- Base (No-Refresh)
- Remapping-Based FCR
- Hybrid FCR
- Adaptive FCR

Lifetime of FCR much higher than lifetime of stronger ECC
Energy Overhead

- Adaptive-rate refresh: <1.8% energy increase until daily refresh is triggered
Yu Cai, Gulay Yalcin, Onur Mutlu, Erich F. Haratsch, Adrian Cristal, Osman Unsal, and Ken Mai,
"Flash Correct-and-Refresh: Retention-Aware Error Management for Increased Flash Memory Lifetime"
Proceedings of the 30th IEEE International Conference on Computer Design (ICCD), Montreal, Quebec, Canada, September 2012.
Slides (ppt) (pdf)
Agenda

- Background, Motivation and Approach
- Experimental Characterization Methodology
- Error Analysis and Management
  - Main Characterization Results
  - Retention-Aware Error Management
  - Threshold Voltage and Program Interference Analysis
  - Read Reference Voltage Prediction
  - Neighbor-Assisted Error Correction
- Summary
Key Questions

- How does threshold voltage (Vth) distribution of different programmed states change over flash lifetime?

- Can we model it accurately and predict the Vth changes?

- Can we build mechanisms that can correct for Vth changes? (thereby reducing read error rates)
Threshold Voltage Distribution Model

Gaussian distribution with additive white noise

As P/E cycles increase ...

- Distribution shifts to the right
- Distribution becomes wider
Threshold Voltage Distribution Model

- **Vth distribution** can be modeled with ~95% accuracy as a Gaussian distribution with additive white noise.

- **Distortion in Vth** over P/E cycles can be modeled and predicted as an exponential function of P/E cycles.
  - With more than 95% accuracy.
More Detail on Threshold Voltage Model

Program Interference Errors

- When a cell is being programmed, **voltage level of a neighboring cell changes** (unintentionally) due to parasitic capacitance coupling
  - can change the data value stored

- Also called program interference error

- Causes neighboring cell voltage to increase (shift right)

- Once retention errors are minimized, these errors can become dominant
How Current Flash Cells are Programmed

- Programming 2-bit MLC NAND flash memory in two steps

Diagram:
- ER (11)
- Temp (0x)
- P1 (10)
- P2 (00)
- P3 (01)

Vth
Basics of Program Interference

WL<0>  WL<1>  WL<2>

(n-1,j-1)  (n-1,j)  (n-1,j+1)

∆V_{xy}  ∆V_x  ∆V_{xy}

(n,j)  (n+1,j)  (n+1,j+1)

∆V_{xy}  ∆V_y  ∆V_{xy}

∆V_{xy}  ∆V_y  ∆V_{xy}

∆V_{xy}  ∆V_y  ∆V_{xy}

Victim Cell

LSB:0  LSB:1  LSB:2
MSB:0  MSB:1  MSB:2
LSB:3  LSB:4  LSB:6
MSB:6
Traditional Model for Vth Change

Traditional model for victim cell threshold voltage change

\[ \Delta V_{\text{victim}} = \left( 2C_x \Delta V_x + C_y \Delta V_y + 2C_{xy} \Delta V_{xy} \right) / C_{\text{total}} \]

Not accurate and requires knowledge of coupling caps!
Our Goal and Idea

- Develop a new, more accurate and easier to implement model for program interference

Idea:
- Empirically characterize and model the effect of neighbor cell Vth changes on the Vth of the victim cell
- Fit neighbor Vth change to a linear regression model and find the coefficients of the model via empirical measurement

\[
\Delta V_{\text{victim}}(n, j) = \sum_{y=j-K}^{j+K} \sum_{x=n+1}^{n+M} \alpha(x, y) \Delta V_{\text{neighbor}}(x, y) + \alpha_{\text{V before victim}}(n, j)
\]

Can be measured
Developing a New Model via Empirical Measurement

- Feature extraction for $V_{th}$ changes based on characterization
  - Threshold voltage changes on aggressor cell
  - Original state of victim cell

- Enhanced linear regression model

\[
\Delta V_{\text{victim}}(n, j) = \sum_{y=j-K}^{j+K} \sum_{x=n}^{n+M} \alpha(x, y) \Delta V_{\text{neighbor}}(x, y) + \alpha_0 V_{\text{before}}^{\text{victim}}(n, j)
\]

\[
Y = X\alpha + \epsilon \quad \text{(vector expression)}
\]

- Maximum likelihood estimation of the model coefficients

\[
\arg\min_{\alpha} \left( \| X \times \alpha - Y \|_2^2 + \lambda \| \alpha \|_1 \right)
\]
Effect of Neighbor Voltages on the Victim

- Immediately-above cell interference is dominant
- Immediately-diagonal neighbor is the second dominant
- Far neighbor cell interference exists
- Victim cell’s Vth has negative effect on interference
New Model for Program Interference

\[
\Delta V_{\text{victim}}(n, j) = \sum_{y=j-K}^{j+K} \sum_{x=n+1}^{n+M} \alpha(x, y) \Delta V_{\text{neighbor}}(x, y) + \alpha_0 V_{\text{before}}^{\text{victim}}(n, j)
\]
Model Accuracy

Characterized on 2Y-nm chips using the read-retry feature

(x,y) = (measured before interference, measured after interference)

Interference causes systematic Vth shift

Ideal if no interference

Model corrects for the Vth shift: 96.8% acc.

(x,y) = (measured before interference, predicted with model)

Ideal if prediction is 100% accurate
Yu Cai, Onur Mutlu, Erich F. Haratsch, and Ken Mai, "Program Interference in MLC NAND Flash Memory: Characterization, Modeling, and Mitigation"

Agenda

- Background, Motivation and Approach
- Experimental Characterization Methodology
- Error Analysis and Management
  - Main Characterization Results
  - Retention-Aware Error Management
  - Threshold Voltage and Program Interference Analysis
  - Read Reference Voltage Prediction
  - Neighbor-Assisted Error Correction
- Summary
Mitigation: Applying the Model

- So, what can we do with the model?
- Goal: Mitigate the effects of program interference caused voltage shifts
Optimum Read Reference for Flash Memory

- Read reference voltage affects the raw bit error rate
  \[ BER_1 = \int_{v_{\text{ref}}}^{+\infty} f(x)dx + \int_{-\infty}^{v_{\text{ref}}} g(x)dx \]
  \[ BER_2 = \int_{v'_{\text{ref}}}^{+\infty} f(x)dx + \int_{-\infty}^{v'_{\text{ref}}} g(x)dx \]

- There exists an optimal read reference voltage
  - Predictable if the statistics (i.e. mean, variance) of threshold voltage distributions are characterized and modeled
Optimum Read Reference Voltage Prediction

- **Vth shift learning** (done every ~1k P/E cycles)
  - Program sample cells with known data pattern and test Vth
  - Program aggressor neighbor cells and test victim Vth after interference
  - Characterize the mean shift in Vth (i.e., program interference noise)

- **Optimum read reference voltage prediction**
  - Default read reference voltage + Predicted mean Vth shift by model
Effect of Read Reference Voltage Prediction

- Read reference voltage prediction reduces raw BER (by 64%) and increases the P/E cycle lifetime (by 30%).

32k-bit BCH Code (acceptable BER = 2x10⁻³)

- 30% lifetime improvement

Graph showing raw bit error rate vs. number of P/E cycles for both with and without read reference voltage prediction.
More on Read Reference Voltage Prediction

- Yu Cai, Onur Mutlu, Erich F. Haratsch, and Ken Mai, "Program Interference in MLC NAND Flash Memory: Characterization, Modeling, and Mitigation" 
Agenda

- Background, Motivation and Approach
- Experimental Characterization Methodology
- **Error Analysis and Management**
  - Main Characterization Results
  - Retention-Aware Error Management
  - Threshold Voltage and Program Interference Analysis
  - Read Reference Voltage Prediction
  - Neighbor-Assisted Error Correction
- Summary
Goal

- Develop a better error correction mechanism for cases where ECC fails to correct a page
Observations So Far

- Immediate neighbor cell has the most effect on the victim cell when programmed.

- A single set of read reference voltages is used to determine the value of the (victim) cell.

- The set of read reference voltages is determined based on the *overall threshold voltage distribution of all cells* in flash memory.
New Observations [Cai+ SIGMETRICS’14]

- Vth distributions of **cells with different-valued immediate-neighbor cells** are significantly different
  - Because neighbor value affects the amount of Vth shift

- **Corollary:** If we know the value of the immediate-neighbor, we can find a *more accurate set of read reference voltages* based on the “conditional” threshold voltage distribution

Victim WL before MSB page of aggressor WL are programmed

Victim WL after MSB page of aggressor WL are programmed
If We Knew the Immediate Neighbor …

- Then, we could choose a different read reference voltage to more accurately read the “victim” cell.
Overall vs Conditional Reading

- Using the optimum read reference voltage based on the overall distribution leads to more errors.

- Better to use the optimum read reference voltage based on the conditional distribution (i.e., value of the neighbor).
  - Conditional distributions of two states are farther apart from each other.
Measurement Results

Raw BER of conditional reading is much smaller than overall reading.
Idea: Neighbor Assisted Correction (NAC)

- Read a page with the read reference voltages based on overall Vth distribution (same as today) and buffer it.

- If ECC fails:
  - Read the immediate-neighbor page
  - Re-read the page using the read reference voltages corresponding to the voltage distribution assuming a particular immediate-neighbor value
  - Replace the buffered values of the cells with that particular immediate-neighbor cell value
  - Apply ECC again
Neighbor Assisted Correction Flow

- Trigger neighbor-assisted reading only when ECC fails
- Read neighbor values and use corresponding read reference voltages in a prioritized order until ECC passes

How to select next local optimum read reference voltage?
Lifetime Extension with NAC

ECC needs to correct 40 bits per 1k-Byte

33% lifetime improvement at no performance loss
Performance Analysis of NAC

No performance loss within nominal lifetime and with reasonable (1%) ECC fail rates
Yu Cai, Gulay Yalcin, Onur Mutlu, Eric Haratsch, Osman Unsal, Adrian Cristal, and Ken Mai, "Neighbor-Cell Assisted Error Correction for MLC NAND Flash Memories"

Agenda

- Background, Motivation and Approach
- Experimental Characterization Methodology
- Error Analysis and Management
  - Main Characterization Results
  - Retention-Aware Error Management
  - Threshold Voltage and Program Interference Analysis
  - Read Reference Voltage Prediction
  - Neighbor-Assisted Error Correction
- Summary
Executive Summary

- Problem: MLC NAND flash memory reliability/endurance is a key challenge for satisfying future storage systems’ requirements

- We are: (1) Building reliable error models for NAND flash memory via experimental characterization, (2) Developing efficient techniques to improve reliability and endurance

- This talk provided a “flash” summary of our recent results published in the past 3 years:
  - Experimental error and threshold voltage characterization [DATE’12&13]
  - Retention-aware error management [ICCD’12]
  - Program interference analysis and read reference V prediction [ICCD’13]
  - Neighbor-assisted error correction [SIGMETRICS’14]
Readings (I)


Readings (II)


Referenced Papers

- All are available at
  http://users.ece.cmu.edu/~omutlu/projects.htm
Related Videos and Course Materials

- Computer Architecture Lecture Videos on Youtube
  - [https://www.youtube.com/playlist?list=PL5PHm2jkkXmidJOd59REog9jDnPDTG6IJ](https://www.youtube.com/playlist?list=PL5PHm2jkkXmidJOd59REog9jDnPDTG6IJ)

- Computer Architecture Course Materials

- Advanced Computer Architecture Course Materials

- Advanced Computer Architecture Lecture Videos on Youtube
  - [https://www.youtube.com/playlist?list=PL5PHm2jkkXmgDN1PLwOY_tGtUlynnyV6D](https://www.youtube.com/playlist?list=PL5PHm2jkkXmgDN1PLwOY_tGtUlynnyV6D)
Thank you.

Feel free to email me with any questions & feedback

onur@cmu.edu
http://users.ece.cmu.edu/~omutlu/
Error Analysis and Management for MLC NAND Flash Memory

Onur Mutlu
onur@cmu.edu
(joint work with Yu Cai, Gulay Yalcin, Eric Haratsch, Ken Mai, Adrian Cristal, Osman Unsal)

August 7, 2014
Flash Memory Summit 2014, Santa Clara, CA
Additional Slides
Error Types and Testing Methodology

- **Erase errors**
  - Count the number of cells that fail to be erased to “11” state

- **Program interference errors**
  - Compare the data immediately after page programming and the data after the whole block being programmed

- **Read errors**
  - Continuously read a given block and compare the data between consecutive read read sequences

- **Retention errors**
  - Compare the data read after an amount of time to data written
    - Characterize short term retention errors under room temperature
    - Characterize long term retention errors by baking in the oven under 125°C
Improving Flash Lifetime with Strong ECC

- Lifetime improvement comparison of various BCH codes

![Bar chart showing P/E cycle endurance for different BCH codes: 512b-BCH, 1k-BCH, 2k-BCH, 4k-BCH, 8k-BCH, and 32k-BCH. The 32k-BCH shows a 4X lifetime improvement and 71X power consumption, 85X area consumption.]

Strong ECC is very inefficient at improving lifetime.
Our Goal

Develop new techniques to improve flash lifetime without relying on stronger ECC
### FCR Intuition

#### Errors with No refresh

<table>
<thead>
<tr>
<th>Event</th>
<th>Program Page</th>
<th>After time T</th>
<th>After time 2T</th>
<th>After time 3T</th>
</tr>
</thead>
<tbody>
<tr>
<td>Program Page</td>
<td>✗</td>
<td>✗</td>
<td>✗</td>
<td></td>
</tr>
<tr>
<td>After time T</td>
<td>✗</td>
<td>✗</td>
<td>✗</td>
<td>✗</td>
</tr>
<tr>
<td>After time 2T</td>
<td>✗</td>
<td>✗</td>
<td>✗</td>
<td>✗</td>
</tr>
<tr>
<td>After time 3T</td>
<td>✗</td>
<td>✗</td>
<td>✗</td>
<td>✗</td>
</tr>
</tbody>
</table>

#### Errors with Periodic refresh

<table>
<thead>
<tr>
<th>Event</th>
<th>Program Page</th>
<th>After time T</th>
<th>After time 2T</th>
<th>After time 3T</th>
</tr>
</thead>
<tbody>
<tr>
<td>Program Page</td>
<td>✗</td>
<td>✗</td>
<td>✗</td>
<td></td>
</tr>
<tr>
<td>After time T</td>
<td>✗</td>
<td>✗</td>
<td>✗</td>
<td>✗</td>
</tr>
<tr>
<td>After time 2T</td>
<td>✗</td>
<td>✗</td>
<td>✗</td>
<td>✗</td>
</tr>
<tr>
<td>After time 3T</td>
<td>✗</td>
<td>✗</td>
<td>✗</td>
<td>✗</td>
</tr>
</tbody>
</table>

- ✗ Retention Error
- ✗ Program Error
FCR Lifetime Evaluation Takeaways

- **Significant average lifetime improvement over no refresh**
  - Adaptive-rate FCR: 46X
  - Hybrid reprogramming/remapping based FCR: 31X
  - Remapping based FCR: 9X

- **FCR lifetime improvement larger than that of stronger ECC**
  - 46X vs. 4X with 32-kbit ECC (over 512-bit ECC)
  - FCR is less complex and less costly than stronger ECC

- **Lifetime on all workloads improves with Hybrid FCR**
  - Remapping based FCR can degrade lifetime on read-heavy WL
  - Lifetime improvement highest in write-heavy workloads
- **Read-retry feature of new NAND flash**
  - Tune read reference voltage and check which $V_{\text{th}}$ region of cells
- **Characterize the threshold voltage distribution of flash cells in programmed states through Monte-Carlo emulation**
Parametric Distribution Learning

- **Parametric distribution**
  - Closed-form formula, only a few number of parameters to be stored
- **Exponential distribution family**
  - Distribution parameter vector
  \[
p(x|\eta) = h(x)g(\eta) \exp \left\{ \eta^T u(x) \right\}
\]
- **Maximum likelihood estimation (MLE) to learn parameters**

  **Observed testing data**

  **Likelihood Function**
  \[
p(X|\eta) = \left( \prod_{n=1}^{N} h(x_n) \right) g(\eta)^N \exp \left\{ \eta^T \sum_{n=1}^{N} u(x_n) \right\}
\]

  **Goal of MLE:** Find distribution parameters to maximize likelihood function
## Selected Distributions

<table>
<thead>
<tr>
<th>Distribution</th>
<th>$p(x \mid \eta)$</th>
<th>Parameters</th>
</tr>
</thead>
</table>
| Gaussian     | \[
\frac{1}{\sigma\sqrt{2\pi}} \exp\left\{-\frac{1}{2} \left(\frac{x - \mu}{\sigma}\right)^2\right\}\] | Mean: $\mu$  
Var: $\sigma^2$ |
| Beta         | \[
\frac{\Gamma(\alpha + \beta)}{\Gamma(\alpha)\Gamma(\beta)} x^{\alpha-1}(1-x)^{\beta-1}\] | Mean: $\alpha/(\alpha+\beta)$  
Var: $\alpha\beta/((\alpha+\beta)^2(\alpha+\beta+1))$ |
| Gamma        | \[
\frac{1}{\theta^k} \frac{1}{\Gamma(k)} x^{k-1} e^{-\frac{x}{\theta}}\] | Mean: $k\theta$  
Var: $k\theta^2$ |
| Log-normal   | \[
\frac{1}{x\sigma\sqrt{2\pi}} \exp\left\{-\frac{(\ln x - \mu)^2}{2\sigma^2}\right\}\] | Mean: $\exp(\mu+\sigma^2/2)$  
Var: $(\exp(\sigma^2)-1)\exp(2\mu+\sigma^2)$ |
| Weibull      | \[
\frac{k}{\lambda} \left(\frac{x}{\lambda}\right)^{k-1} \exp\left\{-\left(\frac{x}{\lambda}\right)^k\right\}\] | Mean: $\lambda \Gamma(1+1/k)$  
Var: $\lambda^2 \Gamma(1+2/k) - \mu^2$ |
Distribution Exploration

Distribution can be approx. modeled as Gaussian distribution

<table>
<thead>
<tr>
<th></th>
<th>Beta</th>
<th>Gamma</th>
<th>Gaussian</th>
<th>Log-normal</th>
<th>Weibull</th>
</tr>
</thead>
<tbody>
<tr>
<td>RMSE</td>
<td>19.5%</td>
<td>20.3%</td>
<td>22.1%</td>
<td>24.8%</td>
<td>28.6%</td>
</tr>
</tbody>
</table>

Distribution can be approx. modeled as Gaussian distribution
Cycling Noise Modeling

Mean value ($\mu$) increases with P/E cycles

\[
\begin{align*}
\text{Exponential model} & \quad V_{th}^{\text{mean, std}}(\text{PEcycle}) \\
& \quad = A + B \times e^{C \times \text{PEcycle}} \\
\text{Linear model} & \quad V_{th}^{\text{mean, std}}(\text{PEcycle}) \\
& \quad = D + E \times \text{PEcycle} 
\end{align*}
\]
Conclusion & Future Work

- **P/E operations modeled as signal passing thru AWGN channel**
  - Approximately Gaussian with 22% distortion
  - P/E noise is white noise

- **P/E cycling noise affects threshold voltage distributions**
  - Distribution shifts to the right and widens around the mean value
  - Statistics (mean/variance) can be modeled as exponential correlation with P/E cycles with 95% accuracy

- **Future work**
  - Characterization and models for retention noise
  - Characterization and models for program interference noise
Program Interference: Key Findings

- **Methodology:** Extensive experimentation with real 2Y-nm MLC NAND Flash chips

- Amount of **program interference** is dependent on
  - Location of cells (programmed and victim)
  - Data values of cells (programmed and victim)
  - Programming order of pages

- Our **new model** can predict the amount of program interference with 96.8% prediction accuracy

- Our **new read reference voltage prediction technique** can improve flash lifetime by 30%
NAC: Executive Summary

Problem: Cell-to-cell Program interference causes threshold voltage of flash cells to be distorted even they are originally programmed correctly

Our Goal: Develop techniques to overcome cell-to-cell program interference

- Analyze the threshold voltage distributions of flash cells conditionally upon the values of immediately neighboring cells
- Devise new error correction mechanisms that can take advantage of the values of neighboring cells to reduce error rates over conventional ECC

Observations: Wide overall distribution can be decoupled into multiple narrower conditional distributions which can be separated easily

Solution: Neighbor-cell Assisted Correction (NAC)

- Re-read a flash memory page that initially failed ECC with a set of read reference voltages corresponding to the conditional threshold voltage distribution
- Use the re-read values to correct the cells that have neighbors with that value
- Prioritize reading assuming neighbor cell values that cause largest or smallest cell-to-cell interference to allow ECC correct errors with less re-reads

Results: NAC improves flash memory lifetime by 39%

- Within nominal lifetime: no performance degradation
- In extended lifetime: less than 5% performance degradation
Overall vs Conditional Vth Distributions

- Overall distribution: \( p(x) \)
- Conditional distribution: \( p(x, z=m) \)
  - \( m \) could be 11, 00, 10 and 01 for 2-bit MLC all-bit-line flash
- Overall distribution is the sum of all conditional distributions

\[
p(x) = \sum_{m=1}^{2^n} p(x, z = m)
\]
Dominant errors are caused by the overlap of lower state interfered by high neighbor interference and the higher state interfered by low neighbor interference.
Procedure of NAC

- Online learning
  - Periodically (e.g., every 100 P/E cycles) measure and learn the overall and conditional threshold voltage distribution statistics (e.g. mean, standard deviation and corresponding optimum read reference voltage)

- NAC procedure
  - Step 1: Once ECC fails reading with overall distribution, load the failed data and corresponding neighbor LSB/MSB data into NAC
  - Step 2: Read the failed page with the local optimum read reference voltage for cells with neighbor programmed as 11
  - Step 3: Fix the value for cells with neighbor 11 in step 1
  - Step 4: Send fixed data for ECC correction. If succeed, exit. Otherwise, go to step 2 and try to read with the local optimum read reference voltage 10, 01 and 00 respectively
Microarchitecture of NAC (Initialization)

Page-to-be-Corrected Buffer

Pass Circuit Vector

Local-Optimum-Read Buffer

Comparator Vector

Bit1

Bit2

Neighbor LSB Page Buffer

Neighbor MSB Page Buffer

SAFARI
NAC (Fixing cells with neighbor 11)

Page-to-be-Corrected Buffer

Pass Circuit Vector

Local-Optimum-Read Buffer

Comparator Vector

Bit1

Bit2

Neighbor LSB Page Buffer

Neighbor MSB Page Buffer

SAFARI