Data Retention in MLC NAND Flash Memory: Characterization, Optimization, and Recovery

Yu Cai, Yixin Luo, Erich F. Haratsch,*
Ken Mai, Onur Mutlu

*Carnegie Mellon University, *LSI Corporation*

SAFARI

Carnegie Mellon

DSSC

You Probably Know

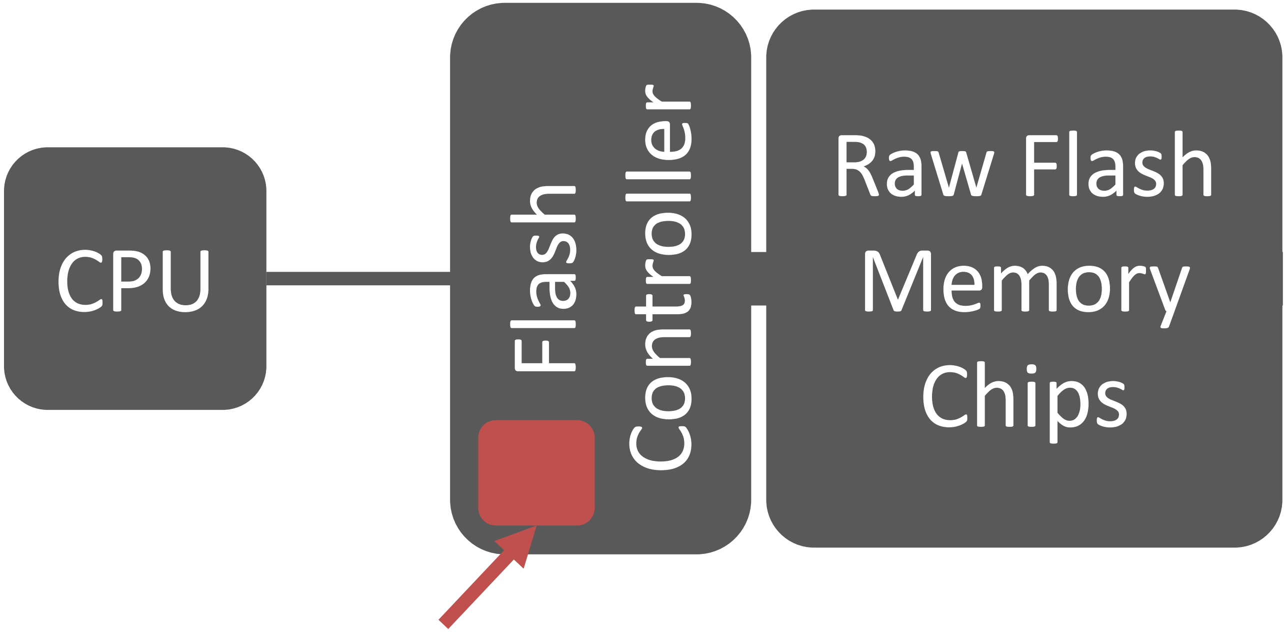
- *Many use cases:*

+ High performance, low energy consumption

NAND Flash Memory Challenges

– *Requires erase before program (write)*

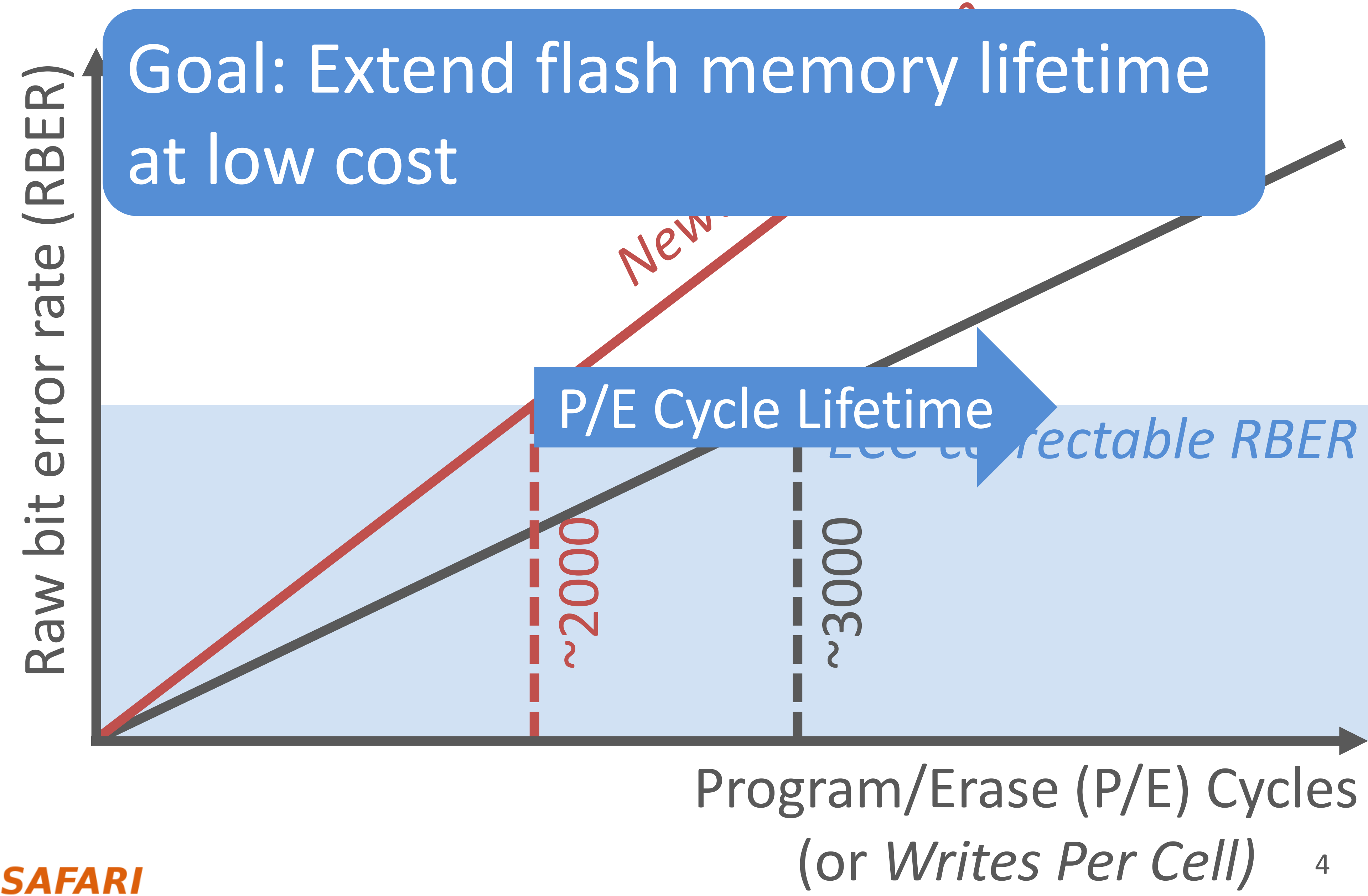
– *High raw bit error rate*



ECC Controller

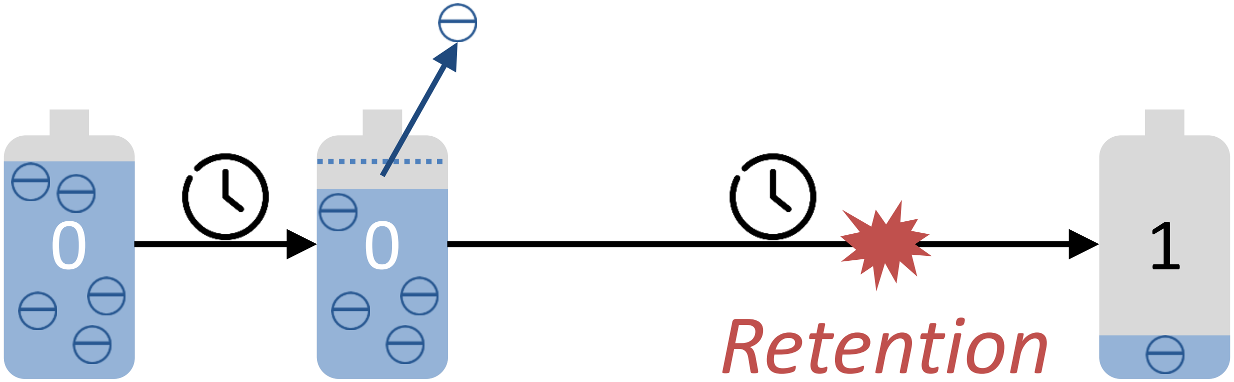
Limited Flash Memory Lifetime

Goal: Extend flash memory lifetime at low cost



Retention Loss

Charge leakage over time



Flash cell

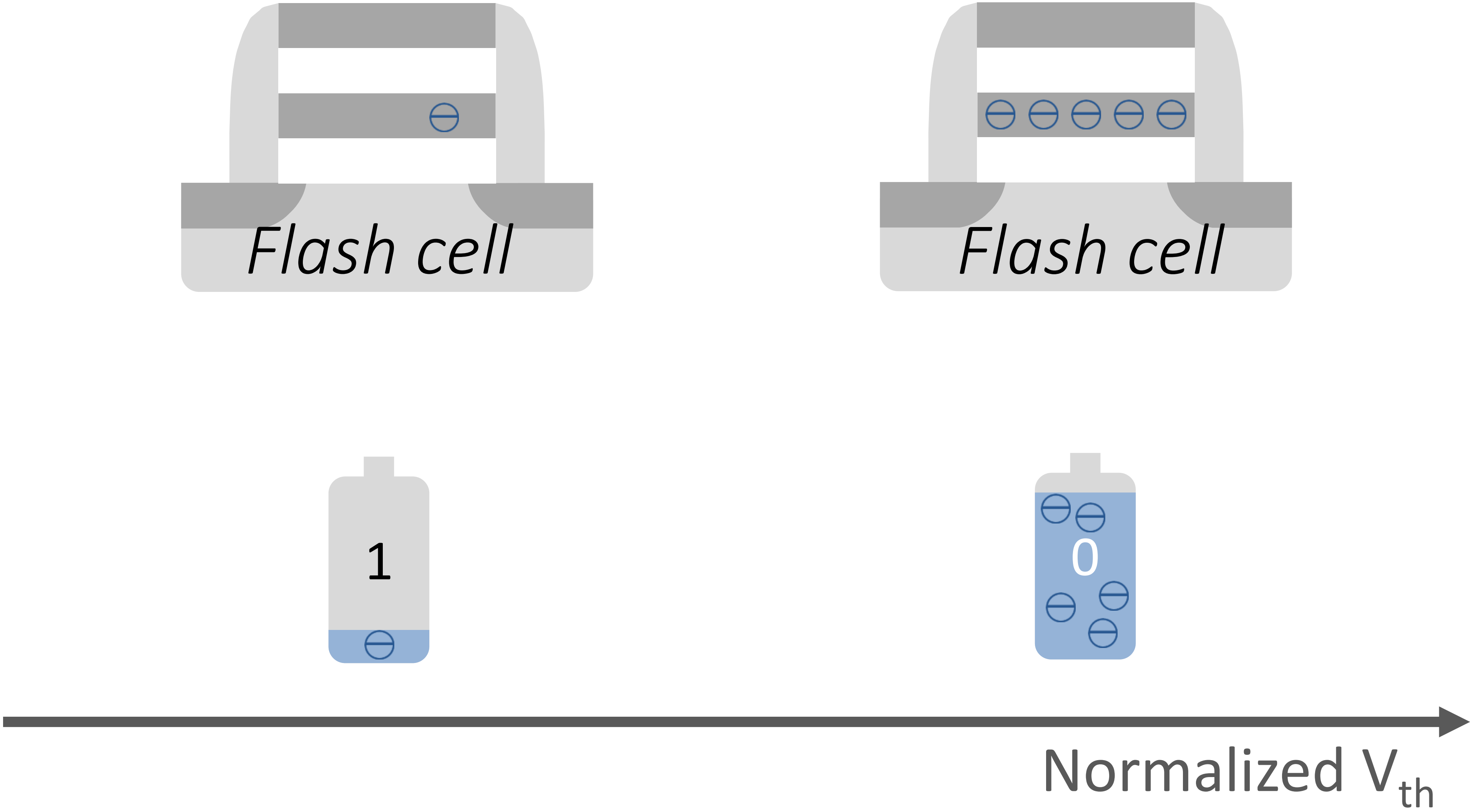
Retention error

One dominant source of flash memory errors [DATE ‘12, ICCD ‘12]

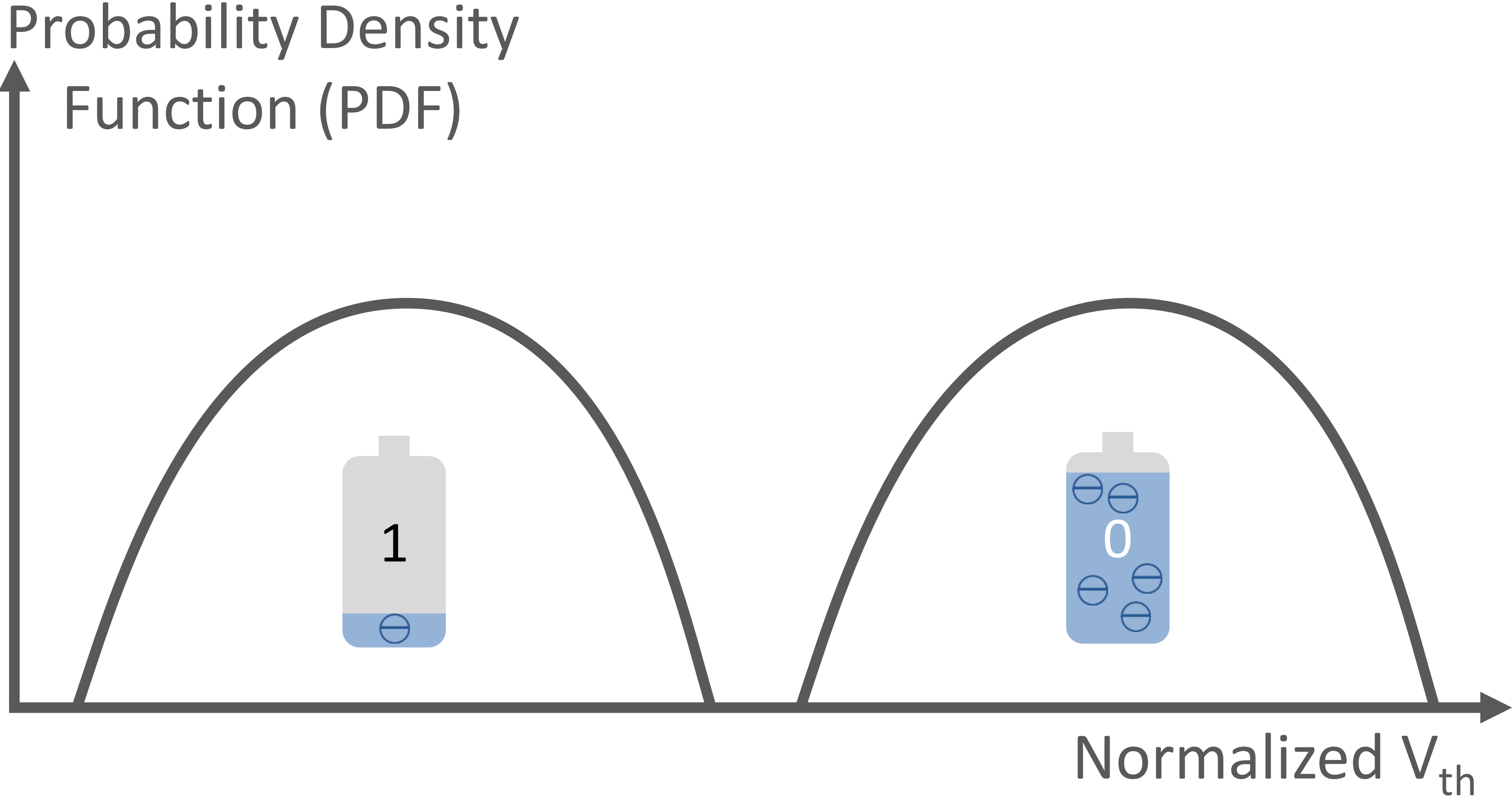
Before I show you
how we extend flash lifetime …

NAND Flash 101

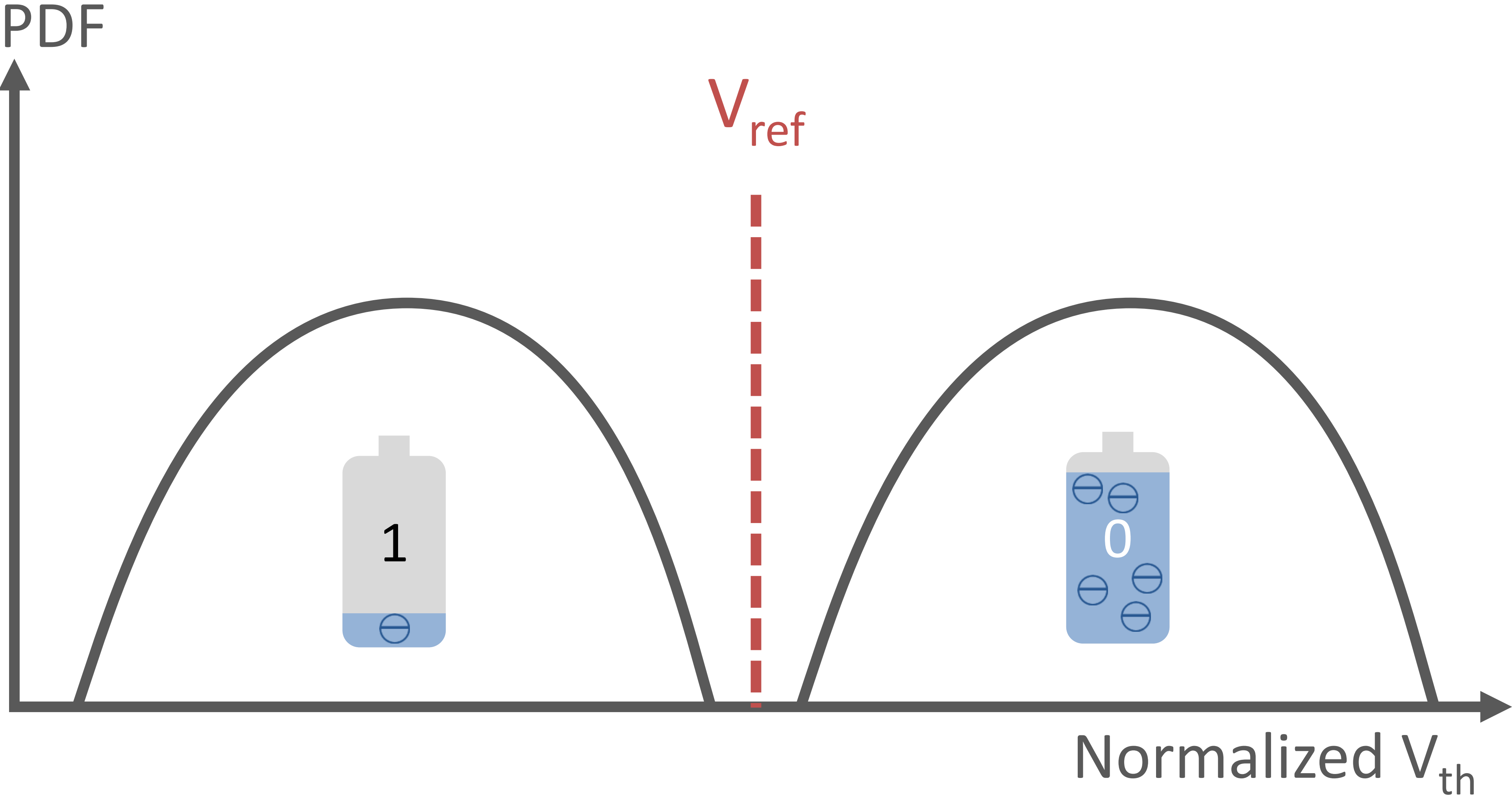
Threshold Voltage (V_{th})



Threshold Voltage (V_{th}) Distribution

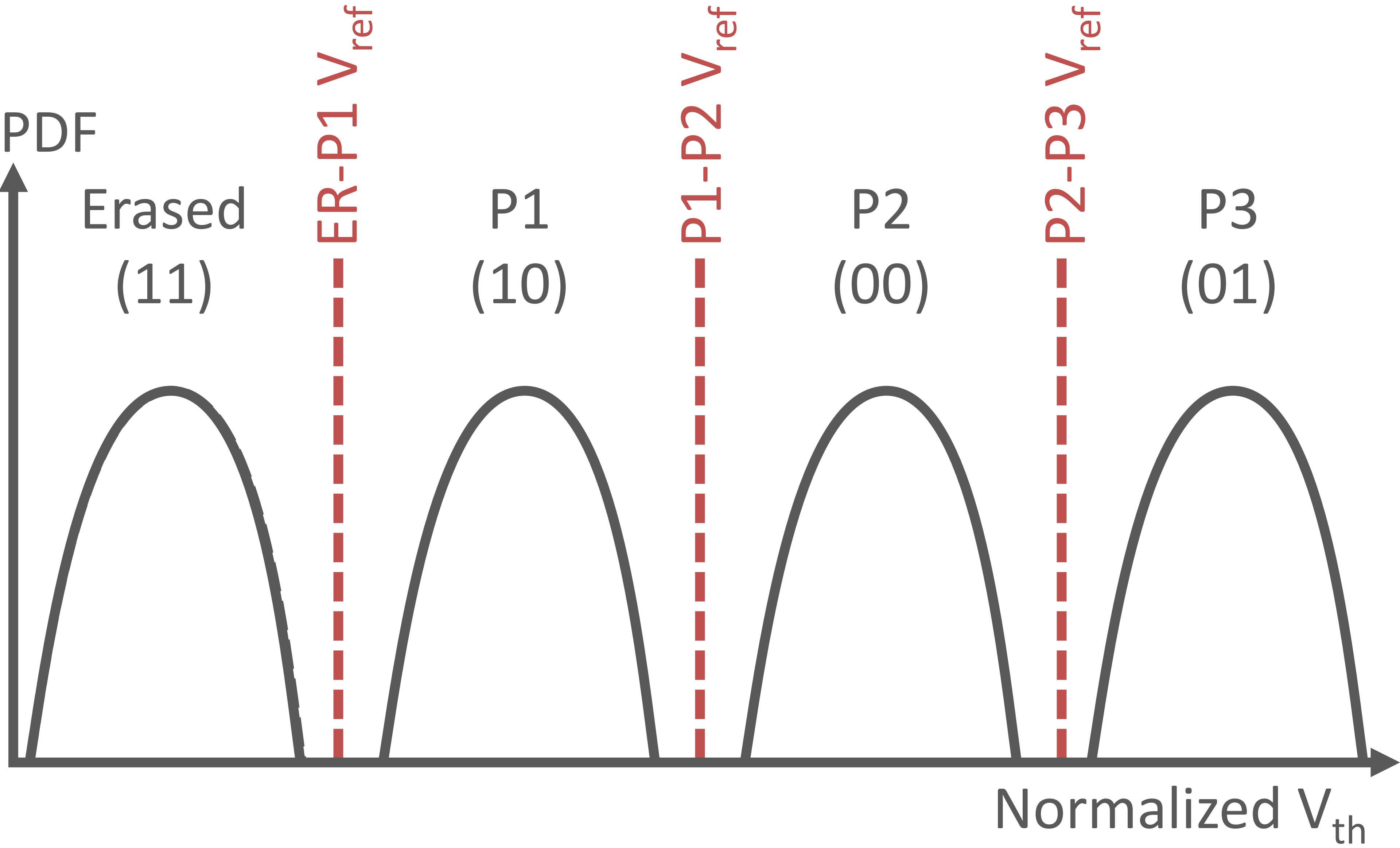


Read Reference Voltage (V_{ref})



SAFARI

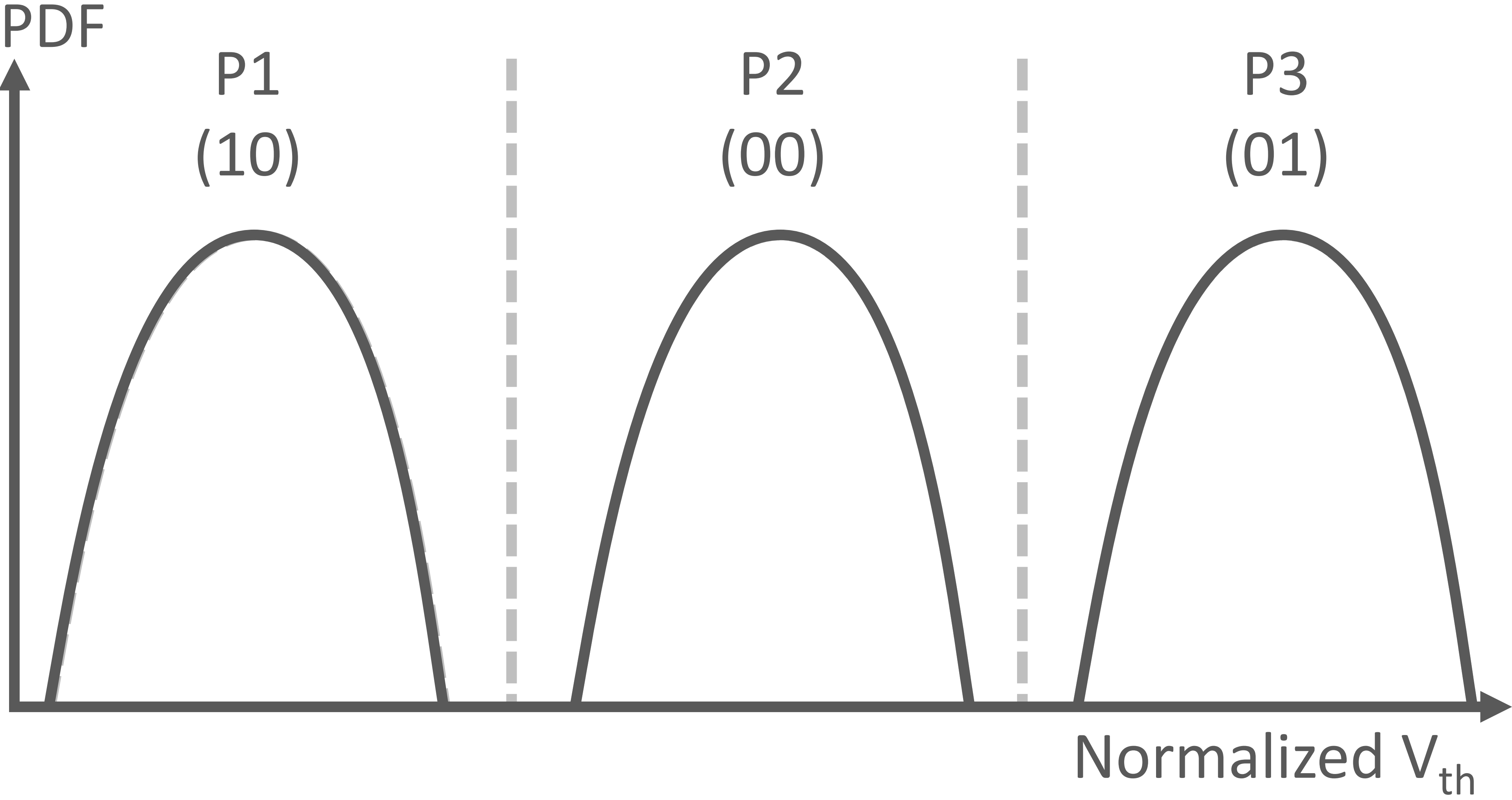
Multi-Level Cell (MLC)



SAFARI

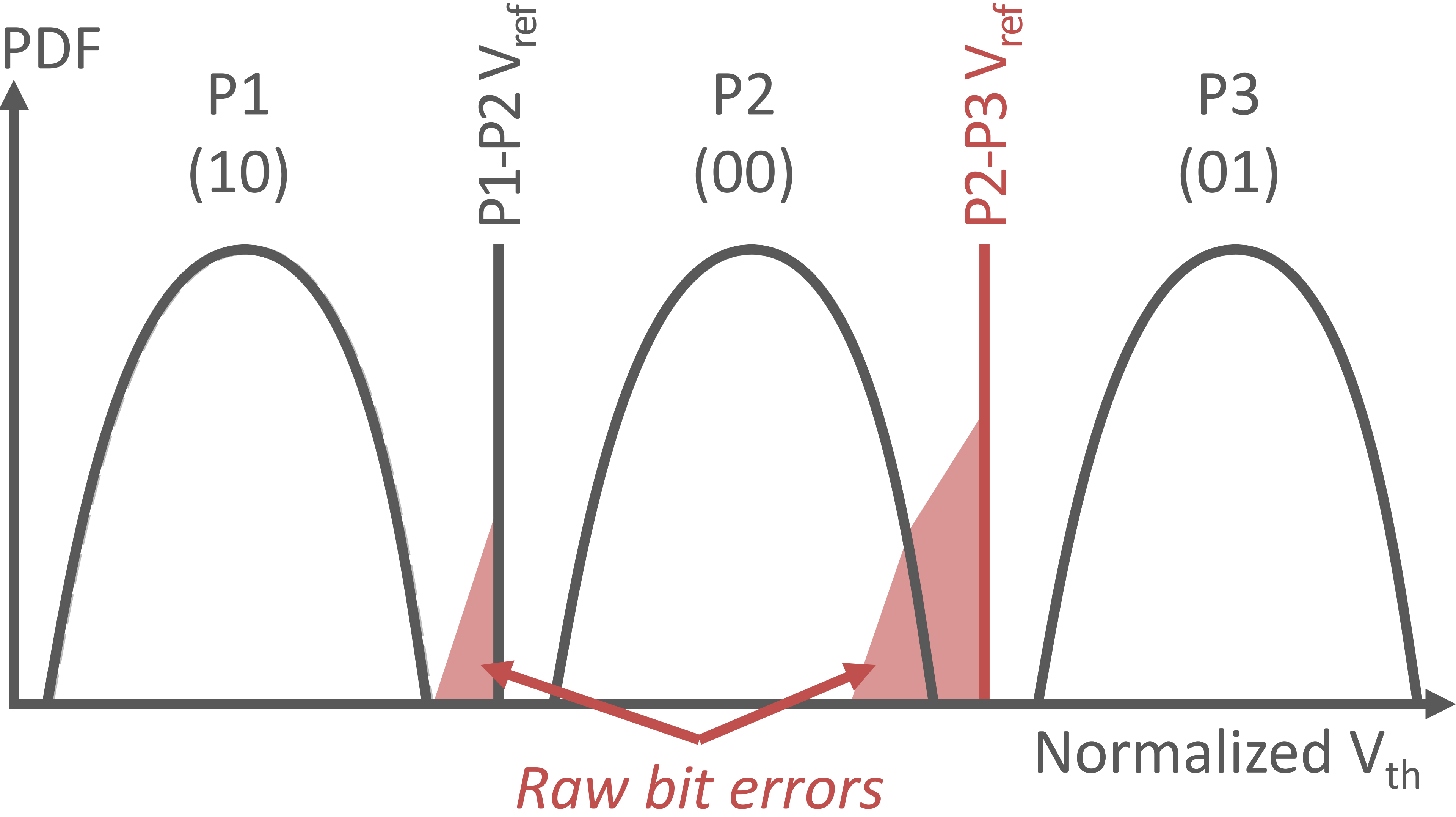
Threshold Voltage Reduces Over Time

After some retention loss:



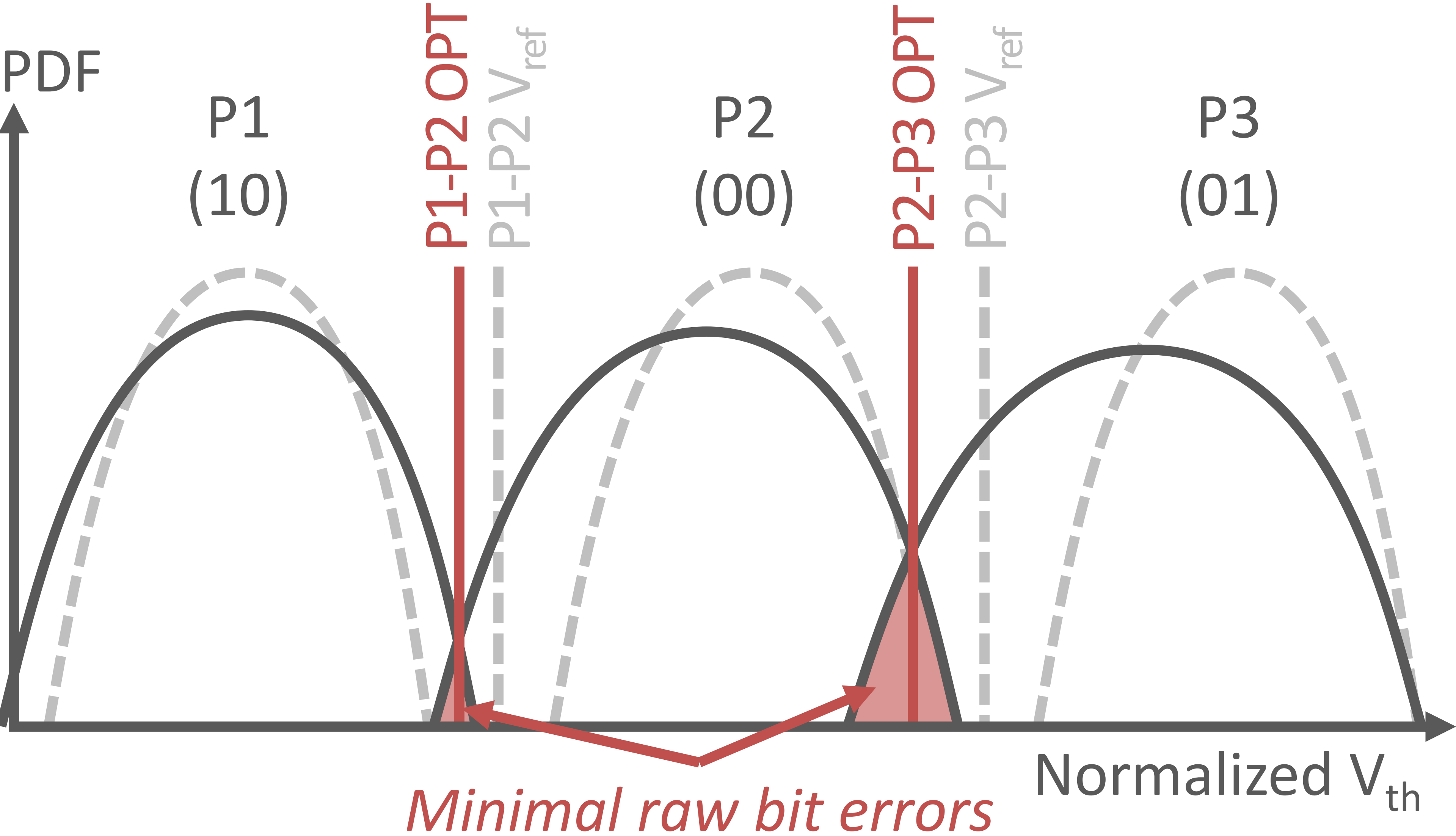
Fixed Read Reference Voltage Becomes Suboptimal

After some retention loss:



Optimal Read Reference Voltage (OPT)

After some retention loss:

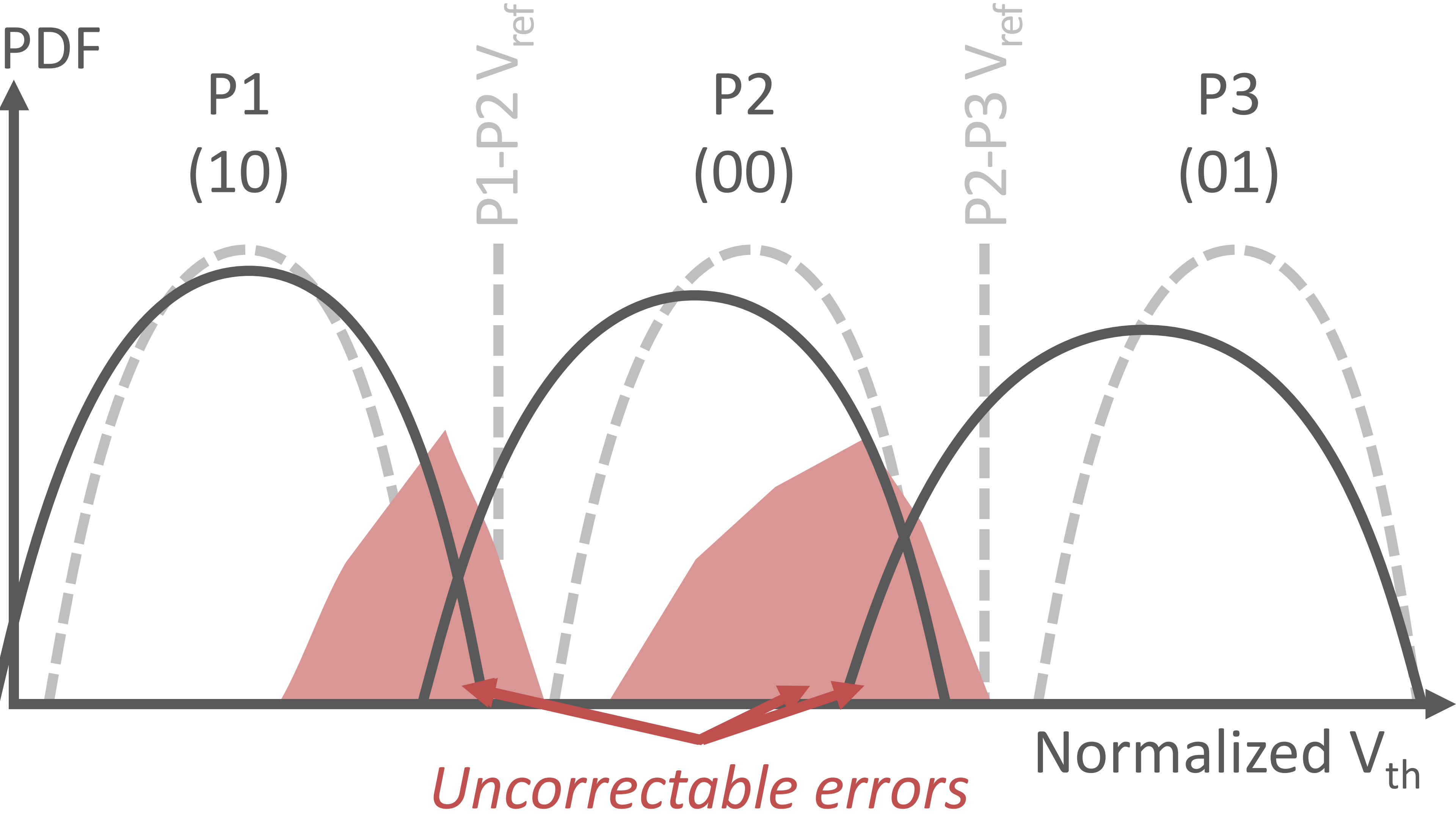


Minimal raw bit errors

Goal 1: Design a low-cost mechanism that dynamically finds the optimal read reference voltage

Retention Failure

After **significant** retention loss:



Goal 1: Design a low-cost mechanism that dynamically finds the optimal read reference voltage

Goal 2: Design an offline mechanism to recover data after detecting uncorrectable errors

To understand the effects of retention loss:

- Characterize retention loss using real chips

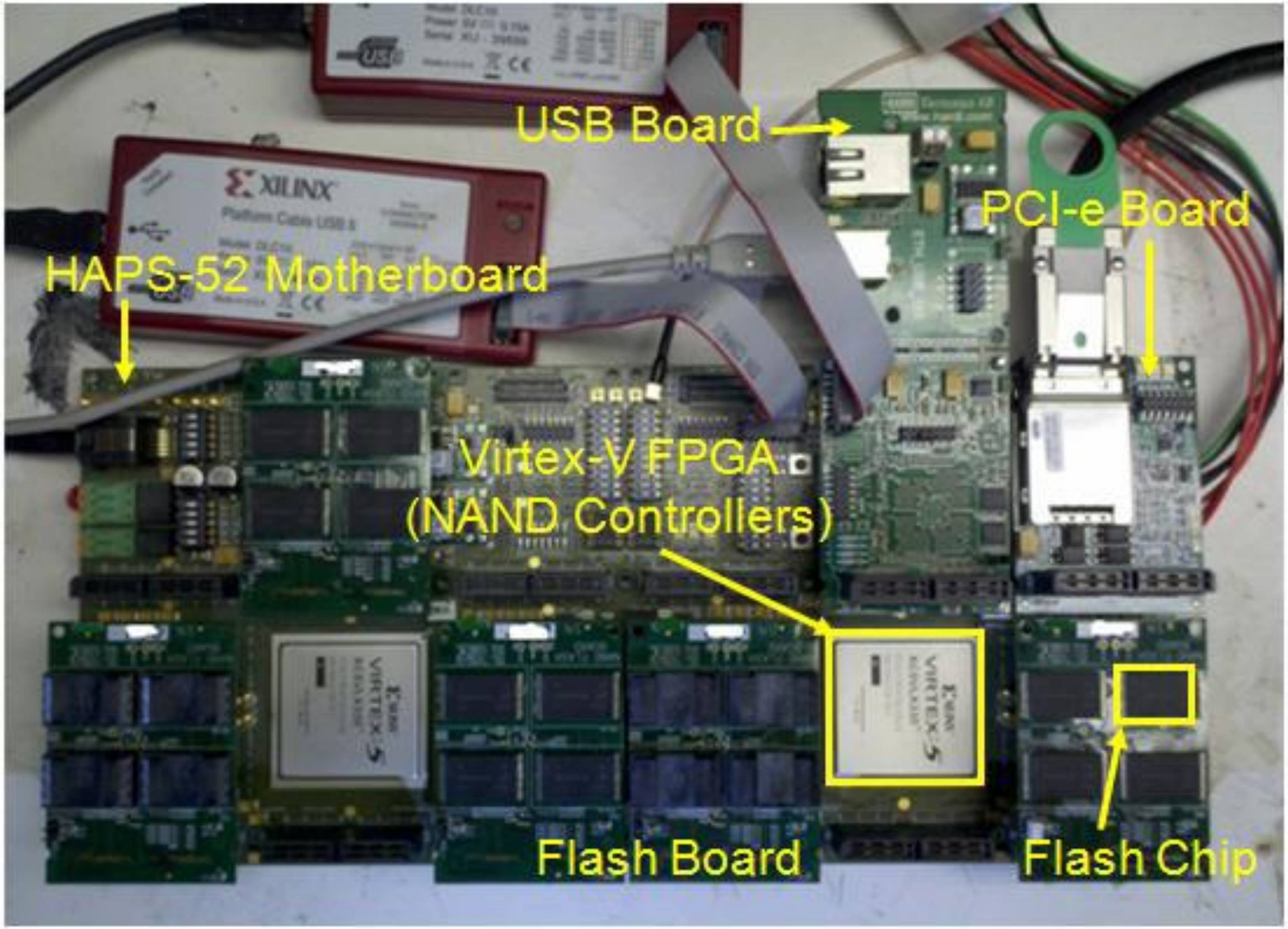
To understand the effects of retention loss:

- Characterize retention loss using real chips

Goal 1: Design a low-cost mechanism that dynamically finds the optimal read reference voltage

Goal 2: Design an offline mechanism to recover data after detecting uncorrectable errors

Characterization Methodology



FPGA-based flash memory testing platform [Cai+,FCCM '11]

Characterization Methodology

- *FPGA-based flash memory testing platform*
- ***Real** 20- to 24-nm MLC NAND flash chips*
- *0- to 40-day worth of retention loss*
- *Room temperature (20^{o}C)*
- *0 to 50k P/E Cycles*

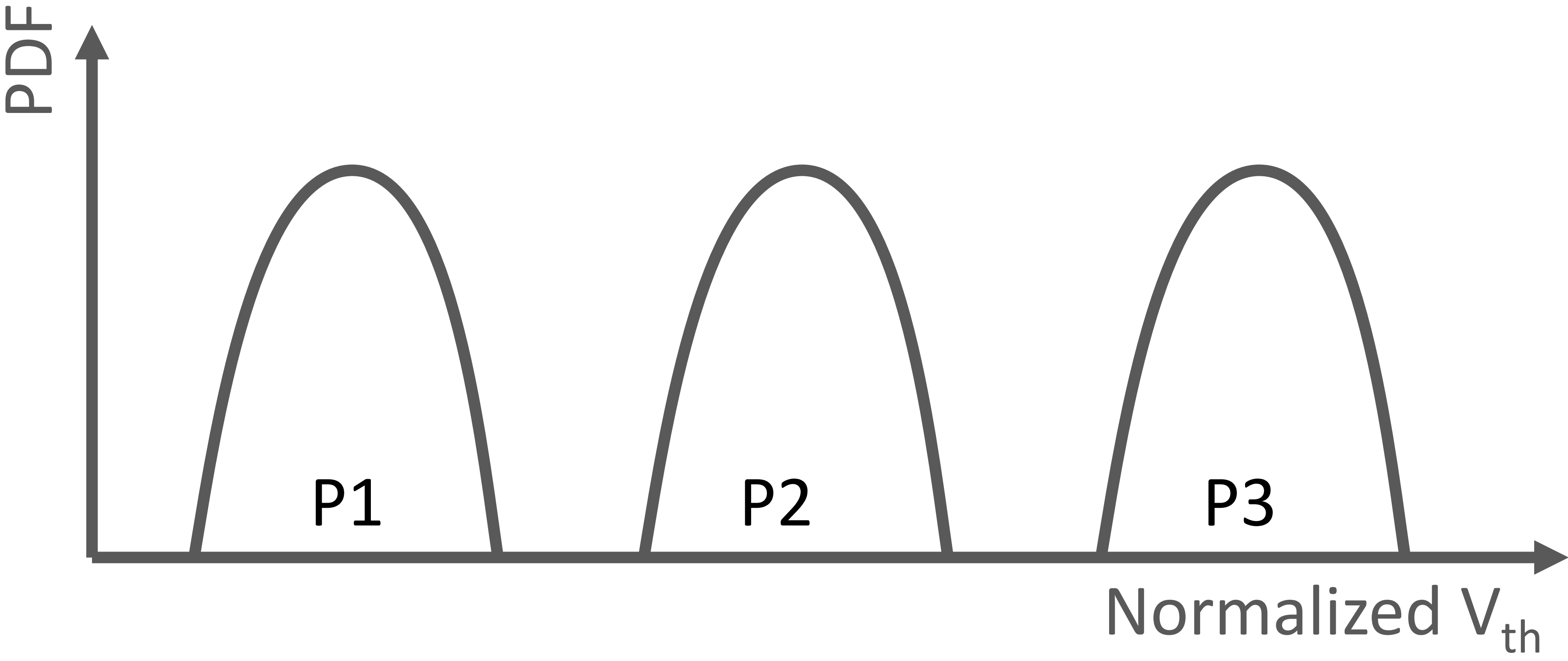
Characterize the effects of retention loss

1. Threshold Voltage Distribution

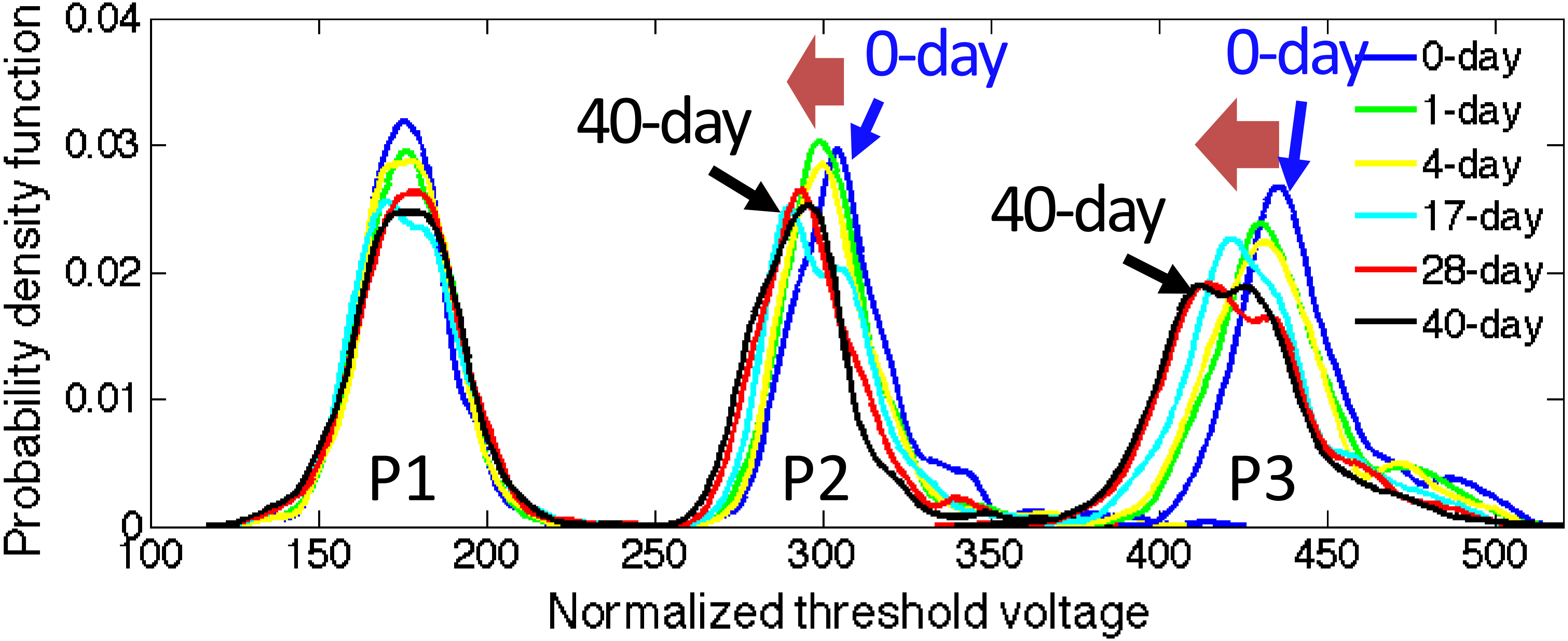
2. Optimal Read Reference Voltage

3. RBER and P/E Cycle Lifetime

1. Threshold Voltage (V_{th}) Distribution

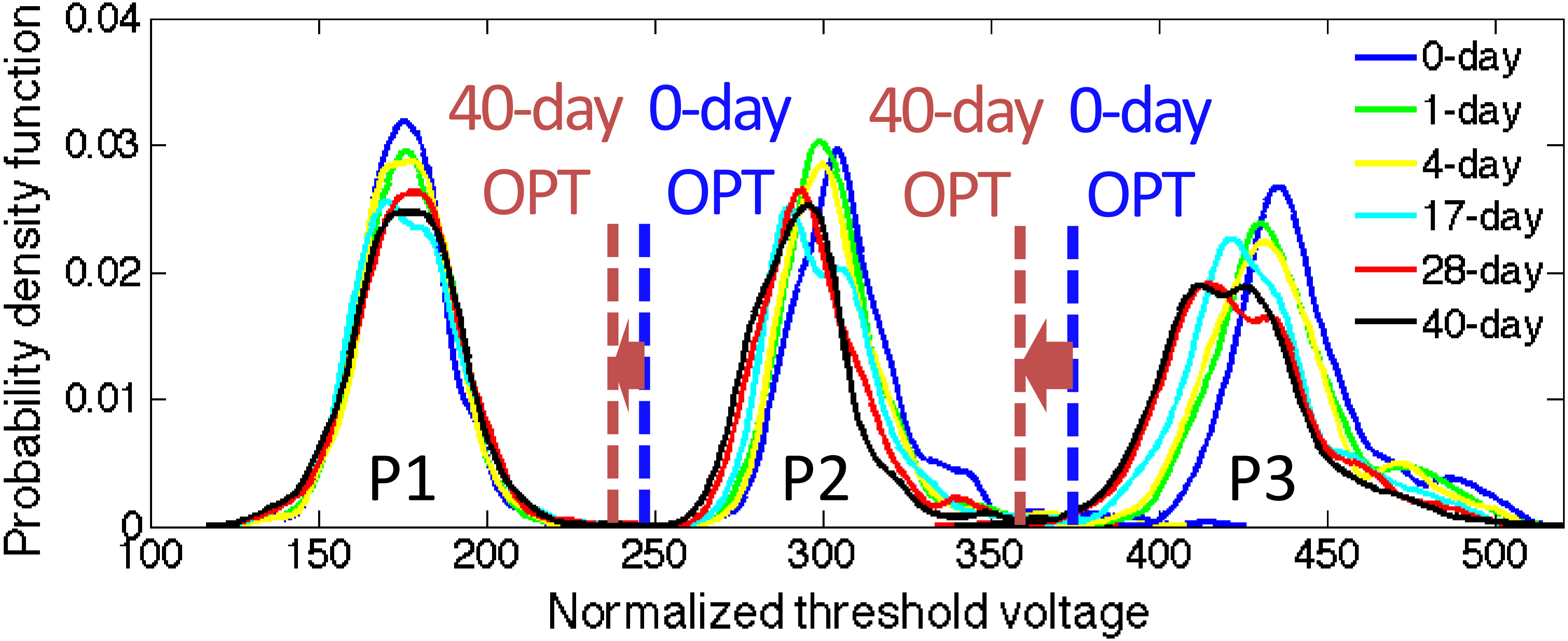


1. Threshold Voltage (V_{th}) Distribution



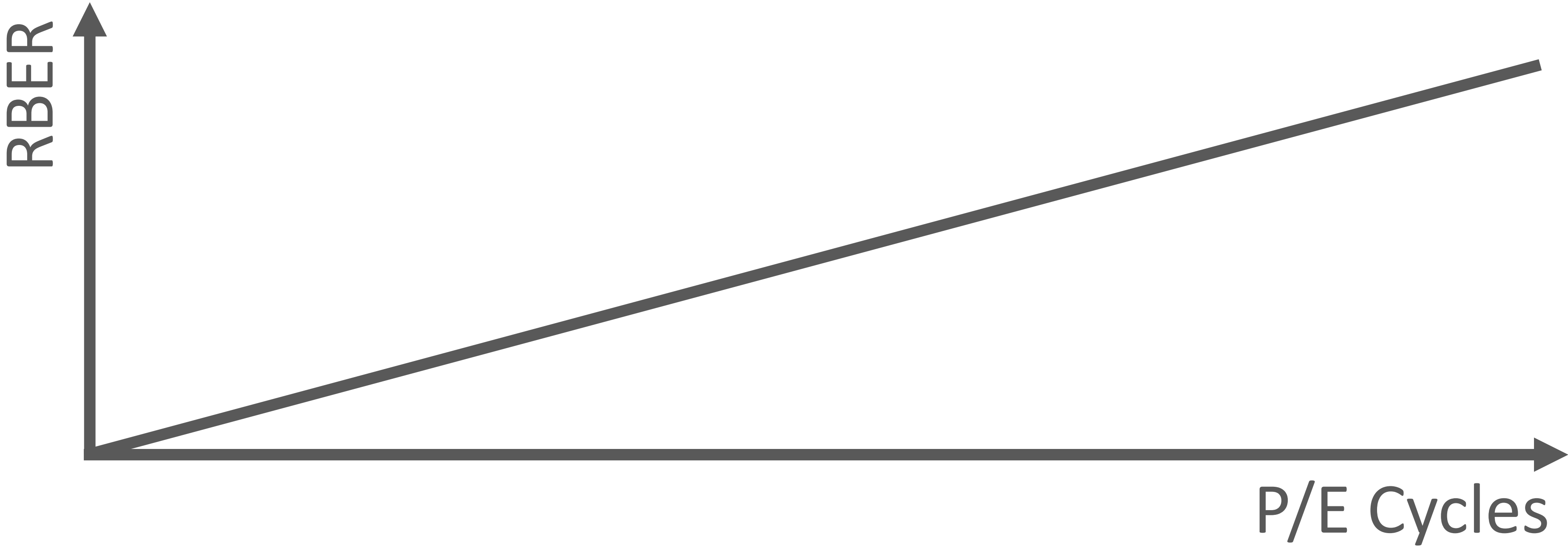
Finding: Cell’s threshold voltage decreases over time

2. Optimal Read Reference Voltage (OPT)



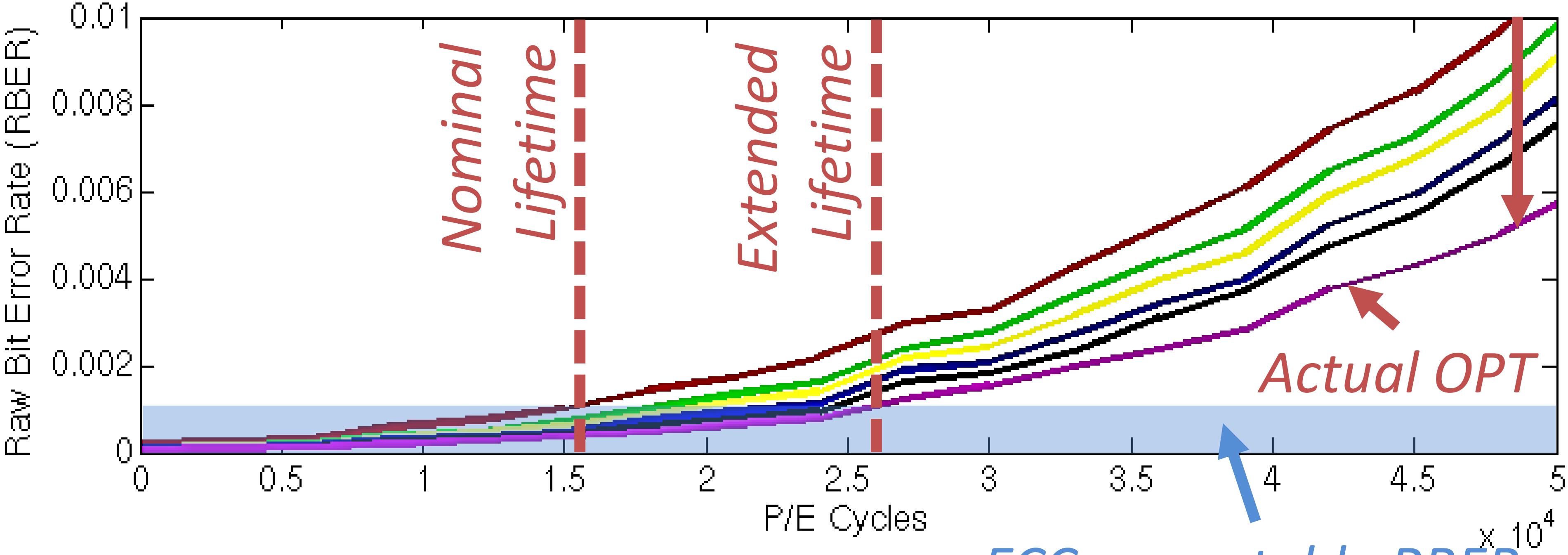
Finding: OPT decreases over time

3. RBER and P/E Cycle Lifetime



3. RBER and P/E Cycle Lifetime

Reading data with 7-day worth of retention loss.



V_{ref} closer to actual OPT

Nominal Lifetime

Extended Lifetime

Actual OPT

ECC-correctable RBER

Finding: Using actual OPT achieves the longest lifetime

Characterization Summary

*Due to **retention loss***

- **Cell's threshold voltage** (V_{th}) decreases over time
- **Optimal read reference voltage** (OPT) decreases over time

*Using the **actual OPT** for reading*

- Achieves the longest **lifetime**

To understand the effects of retention loss:
 - Characterize retention loss using real chips

Goal 1: Design a low-cost mechanism that dynamically finds the optimal read reference voltage

Goal 2: Design an offline mechanism to recover data after detecting uncorrectable errors

Naïve Solution: Sweeping V_{ref}

Key idea: Read the data multiple times with different read reference voltages until the raw bit errors are correctable by ECC

✓*Finds the optimal read reference voltage*

✘*Requires many read-retries → higher read latency*

Comparison of Flash Read Techniques

Flash Read Techniques	***Lifetime (P/E Cycle)***	***Performance (Read Latency)***
Fixed V_{ref}	✗	✓
Sweeping V_{ref}	✓	✗
Our Goal	✓	✓

SAFARI

Observations

1. The optimal read reference voltage gradually decreases over time

Key idea: Record the old OPT as a prediction (V_{pred}) of the actual OPT

Benefit: Close to actual OPT → Fewer read retries

2. The amount of retention loss is similar across pages within a flash block

Key idea: Record only one V_{pred} for each block

Benefit: Small storage overhead (768KB out of 512GB)

Retention Optimized Reading (ROR)

Components:

1. Online pre-optimization algorithm

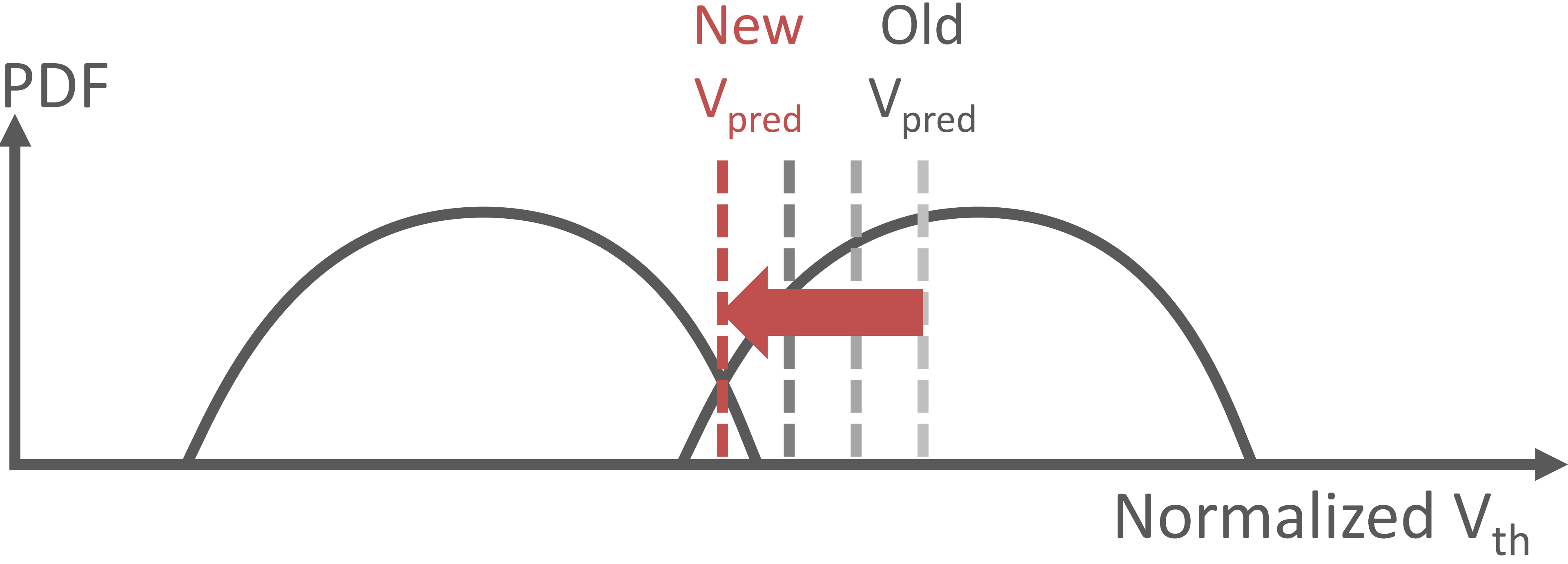
- Periodically records a V_{pred} for each block

2. Improved read-retry technique

- Utilizes the recorded V_{pred} to minimize read-retry count

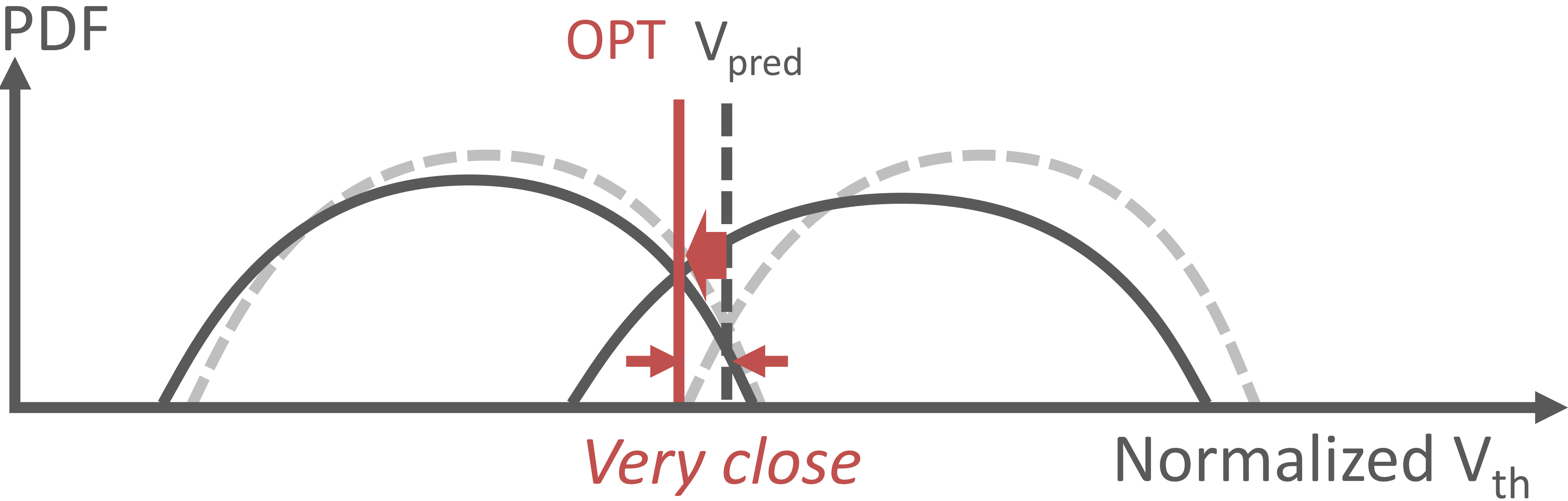
1. Online Pre-Optimization Algorithm

- *Triggered periodically (e.g., per day)*
- *Find and record an OPT as per-block V_{pred}*
- *Performed in background*
- *Small storage overhead*



2. Improved Read-Retry Technique

- *Performed as normal read*
- *V_{pred} already close to actual OPT*
- *Decrease V_{ref} if V_{pred} fails, and retry*



Retention Optimized Reading: Summary

Flash Read Techniques	***Lifetime (P/E Cycle)***	***Performance (Read Latency)***
Fixed V_{ref}	✗	✓
Sweeping V_{ref}	✓ 64% ↑	✗
ROR	✓ 64% ↑	✓ Nom. Life: 2.4% ↓ Ext. Life: 70.4% ↓

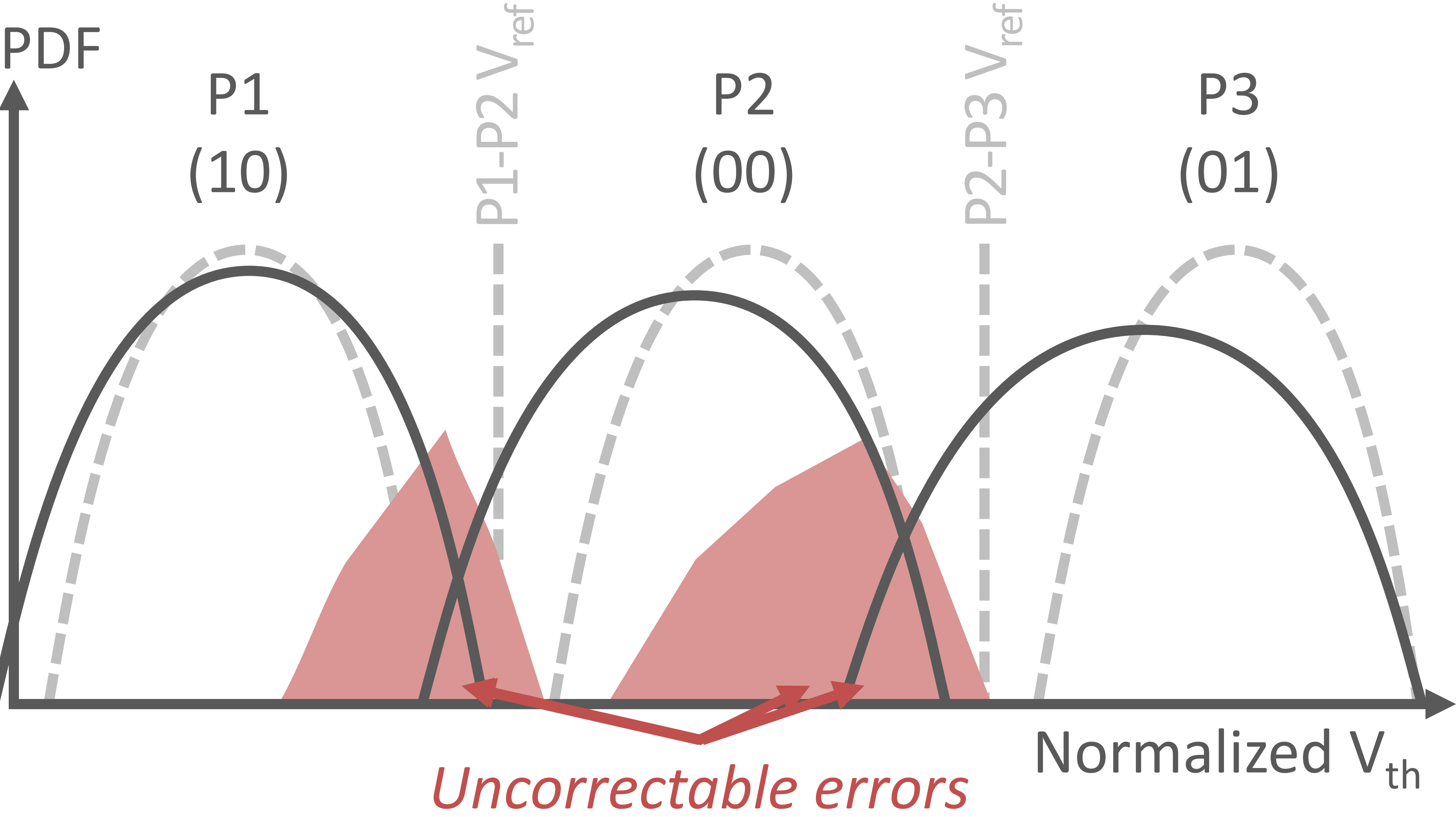
To understand the effects of retention loss:
- Characterize retention loss using real chips

Goal 1: Design a low-cost mechanism that dynamically finds the optimal read reference voltage

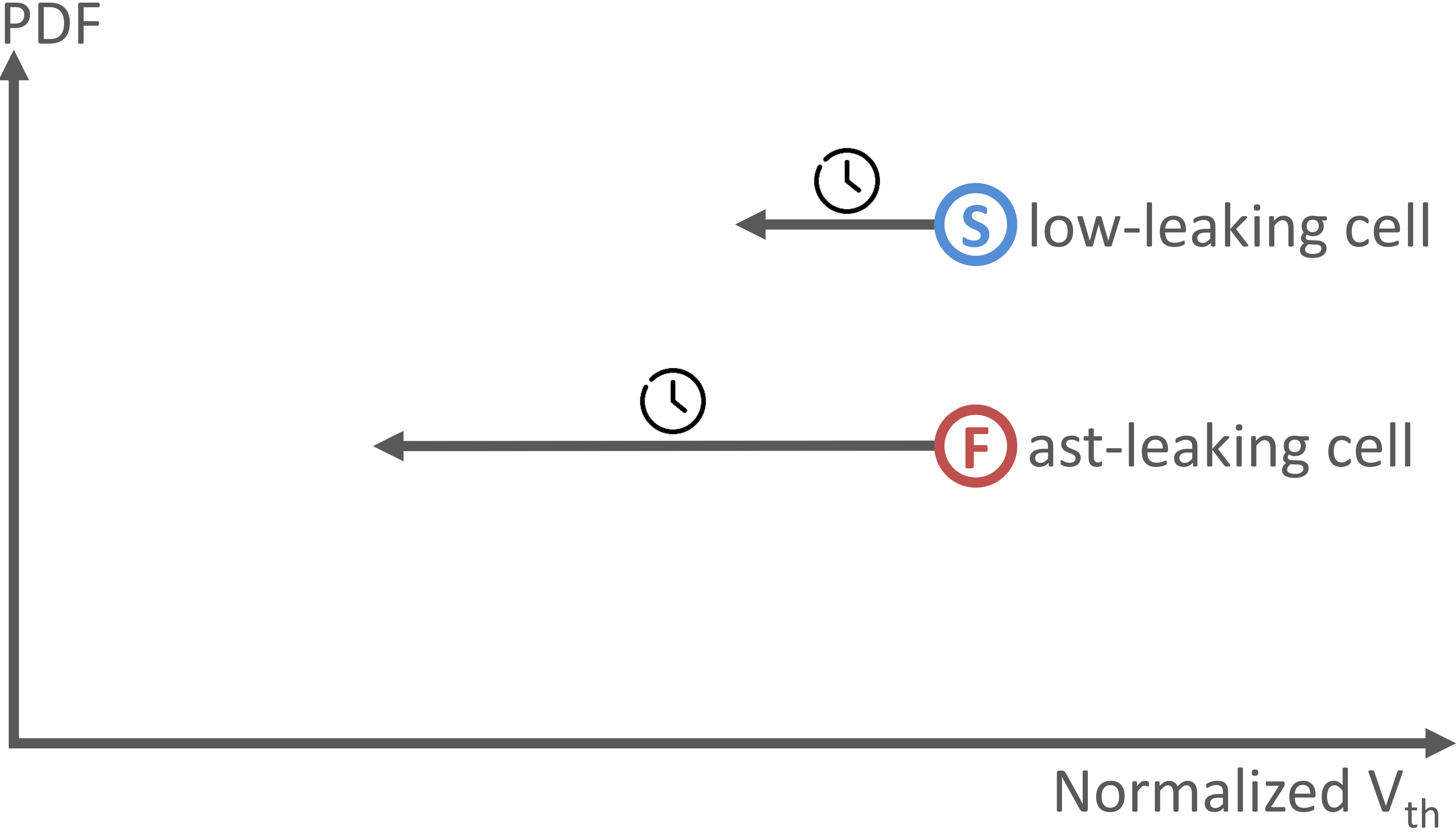
Goal 2: Design an offline mechanism to recover data after detecting uncorrectable errors

Retention Failure

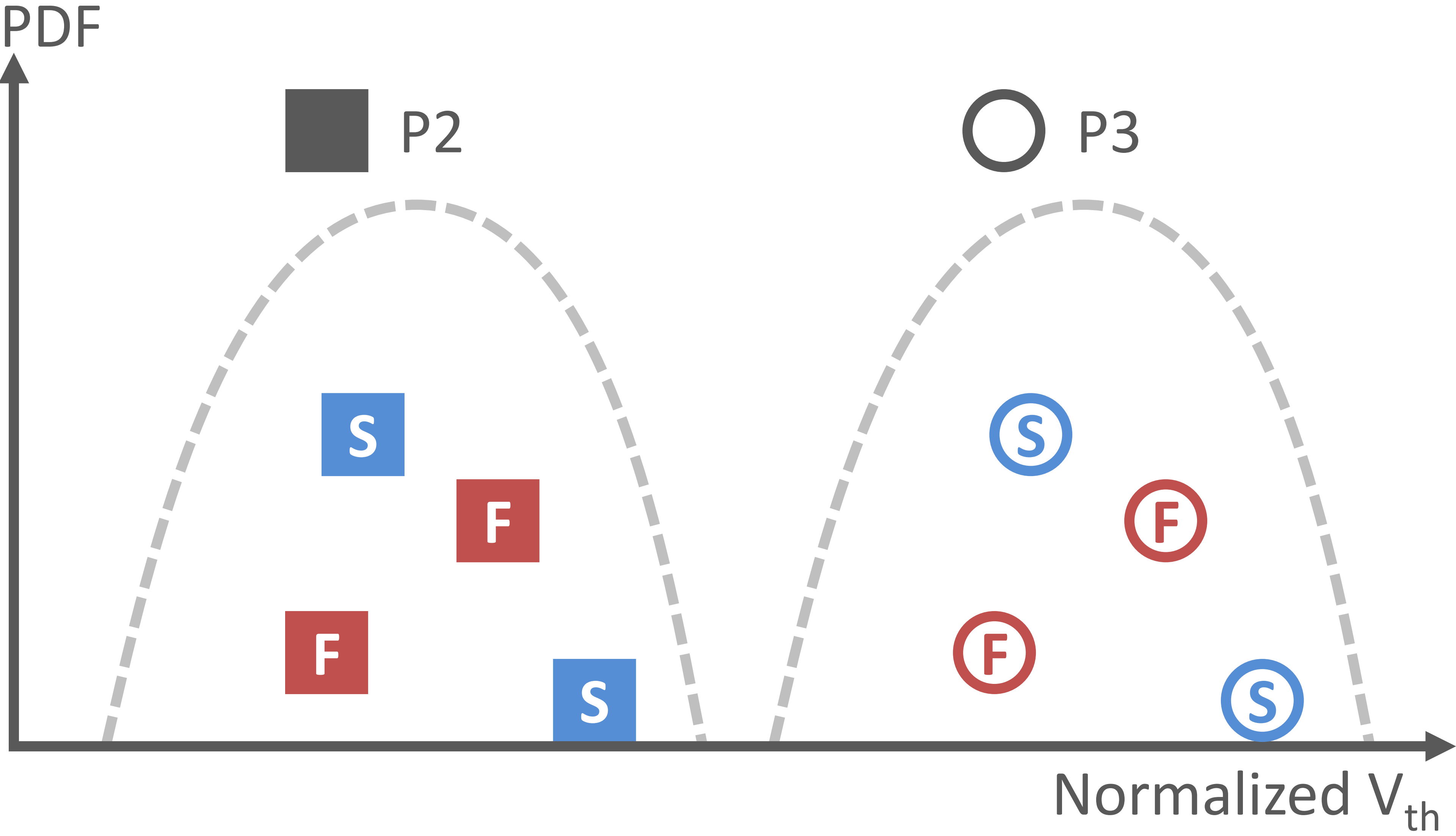
After **significant** retention loss:



Leakage Speed Variation



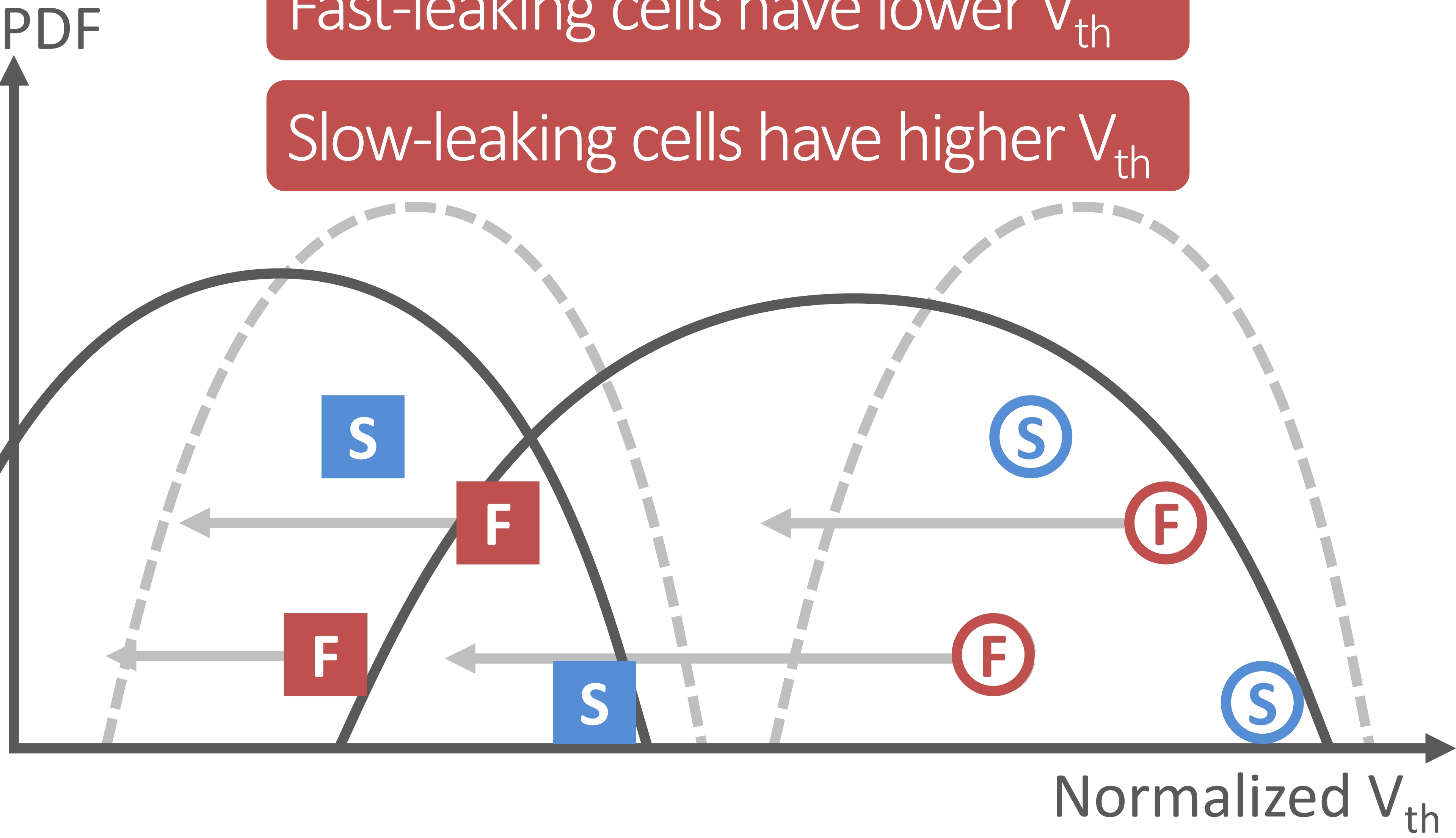
Initially, Right After Programming



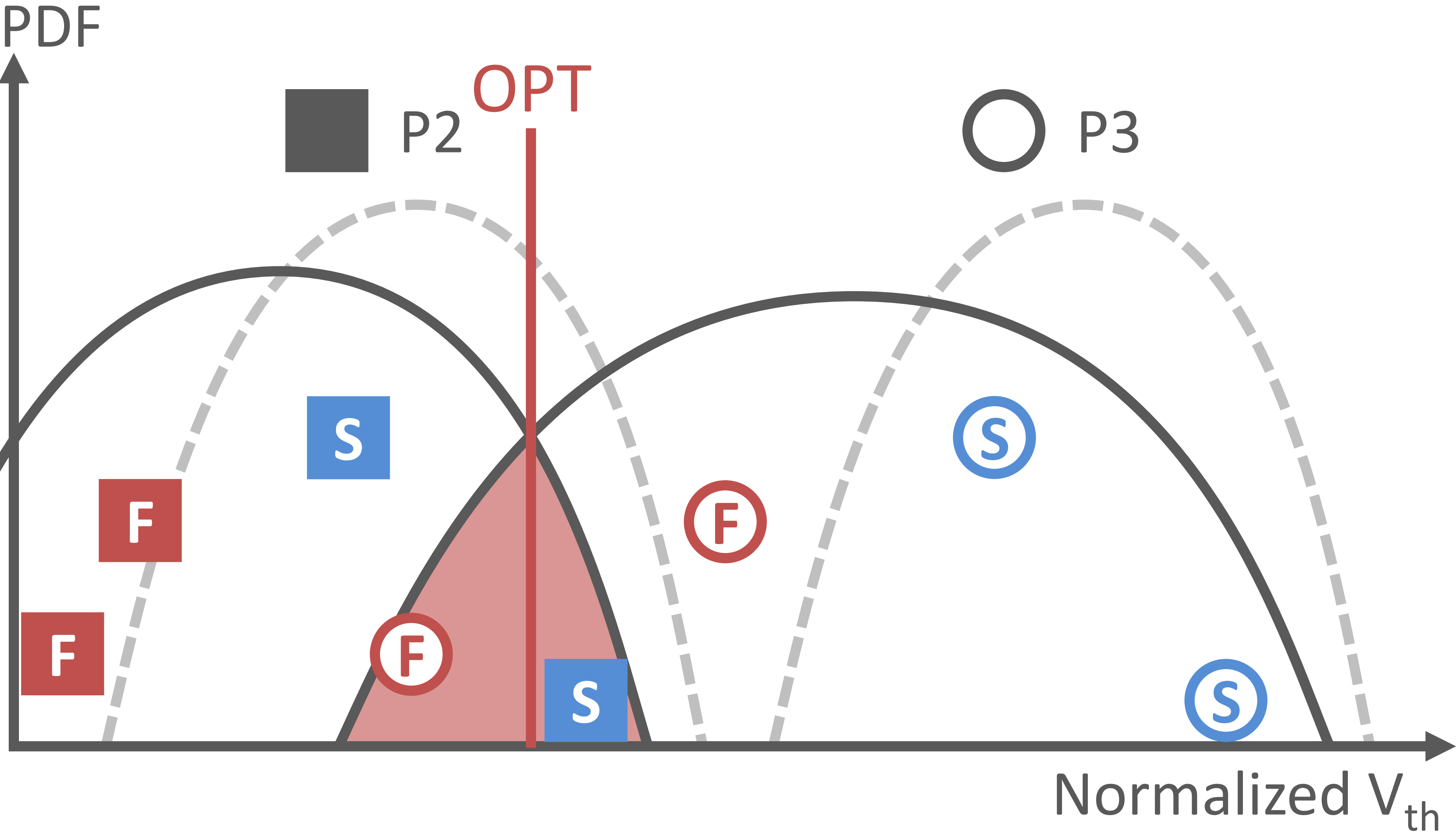
After Some Retention Loss

Fast-leaking cells have lower V_{th}

Slow-leaking cells have higher V_{th}



Eventually: Retention Failure



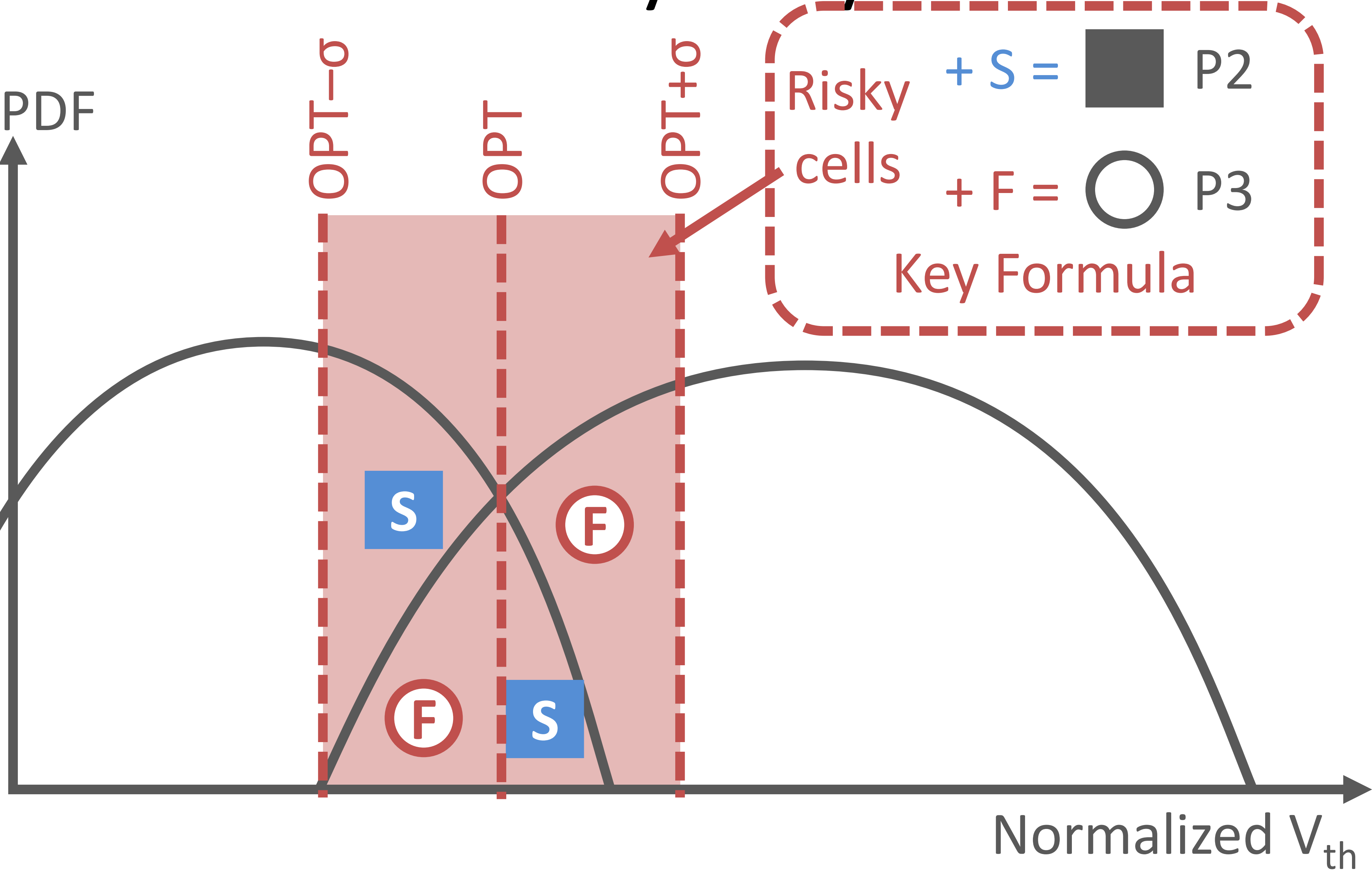
Retention Failure Recovery (RFR)

Key idea: *Guess original state of the cell from its leakage speed property*

Three steps

1. *Identify risky cells*
2. *Identify fast-/slow-leaking cells*
3. *Guess original states*

1. Identify Risky Cells

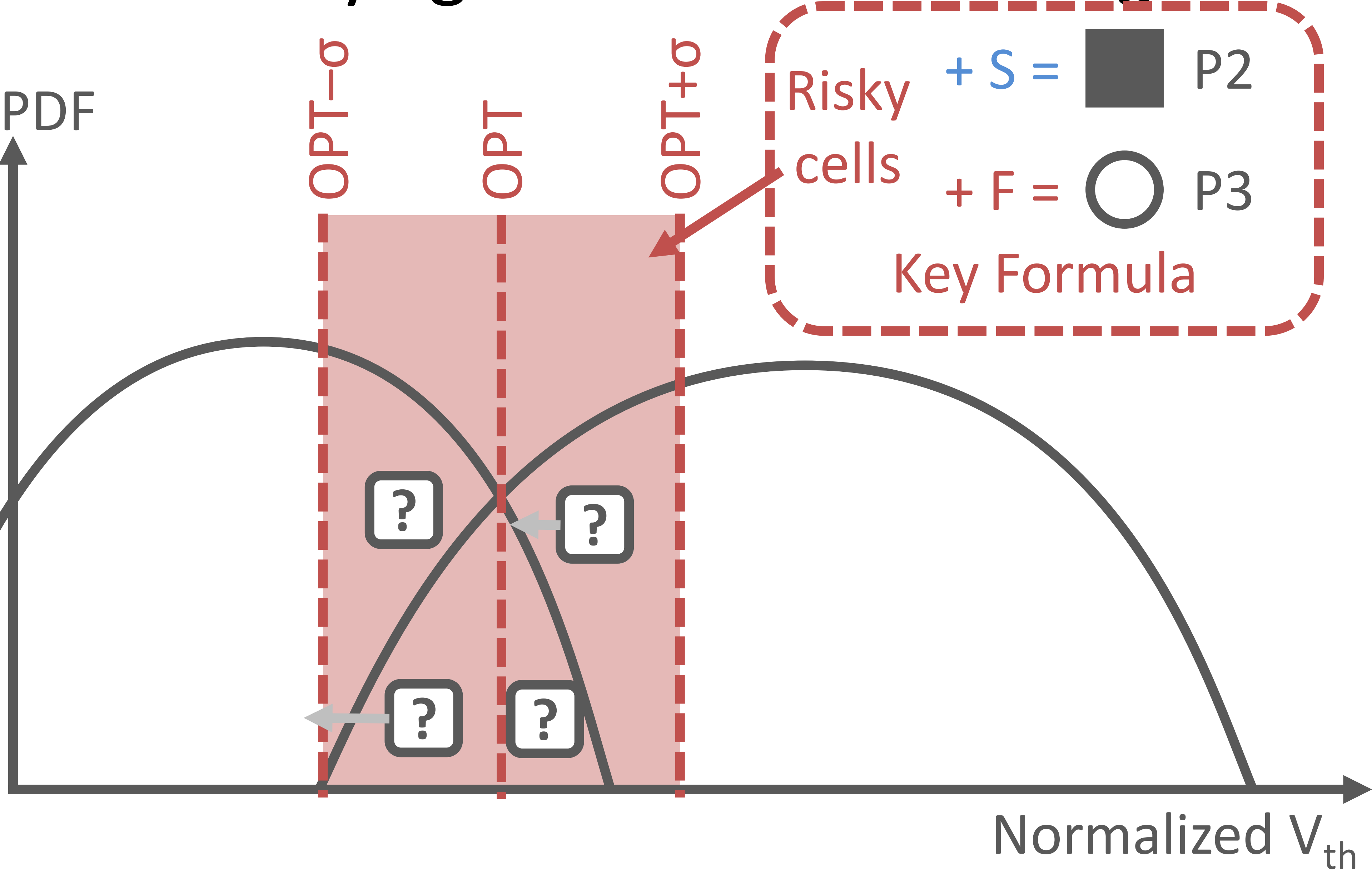


Risky cells + S = ■ P2

+ F = ○ P3

Key Formula

2. Identifying Fast- vs. Slow-Leaking Cells



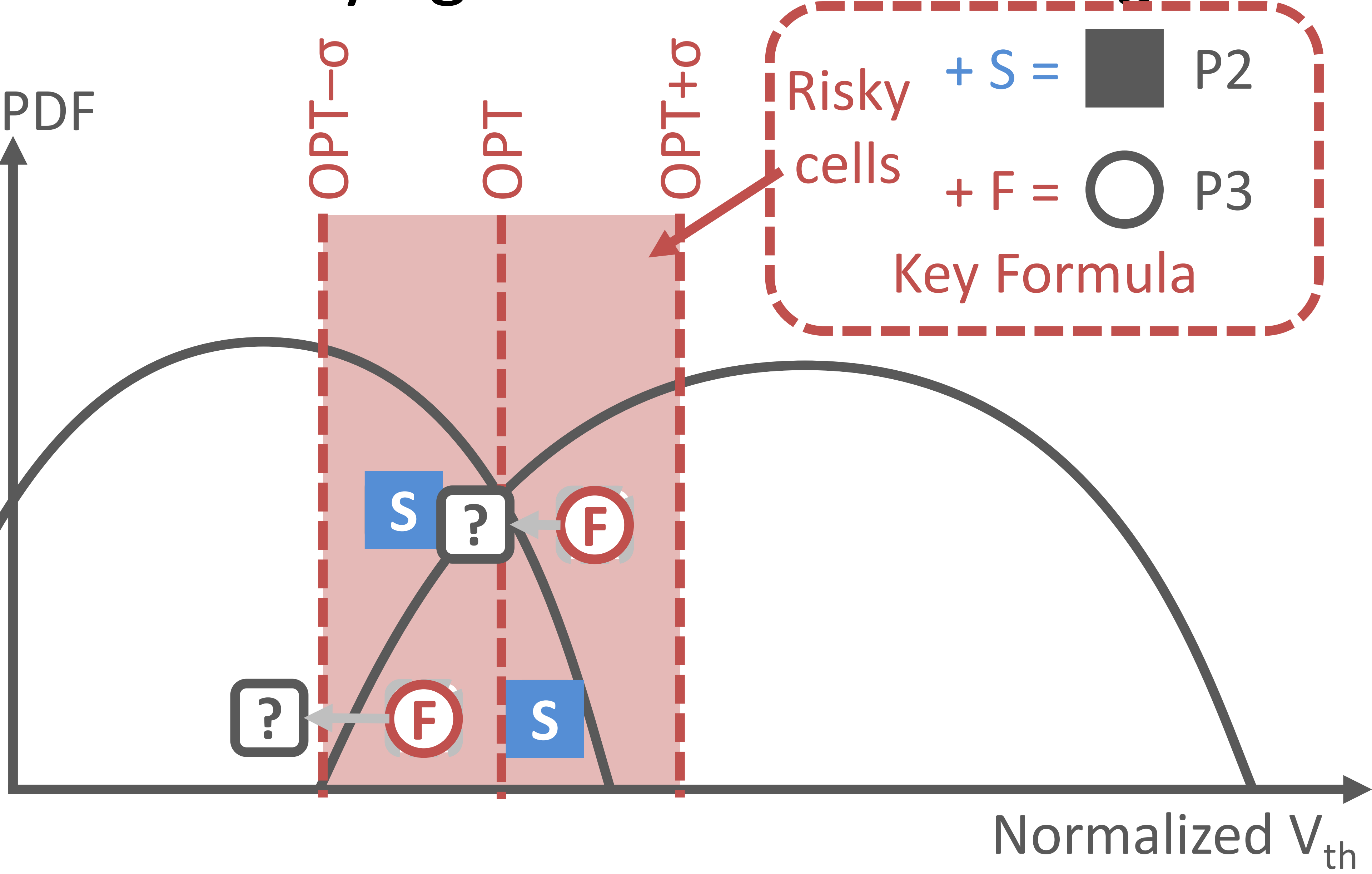
Risky cells

+ S = ■ P2

+ F = ○ P3

Key Formula

2. Identifying Fast- vs. Slow-Leaking Cells

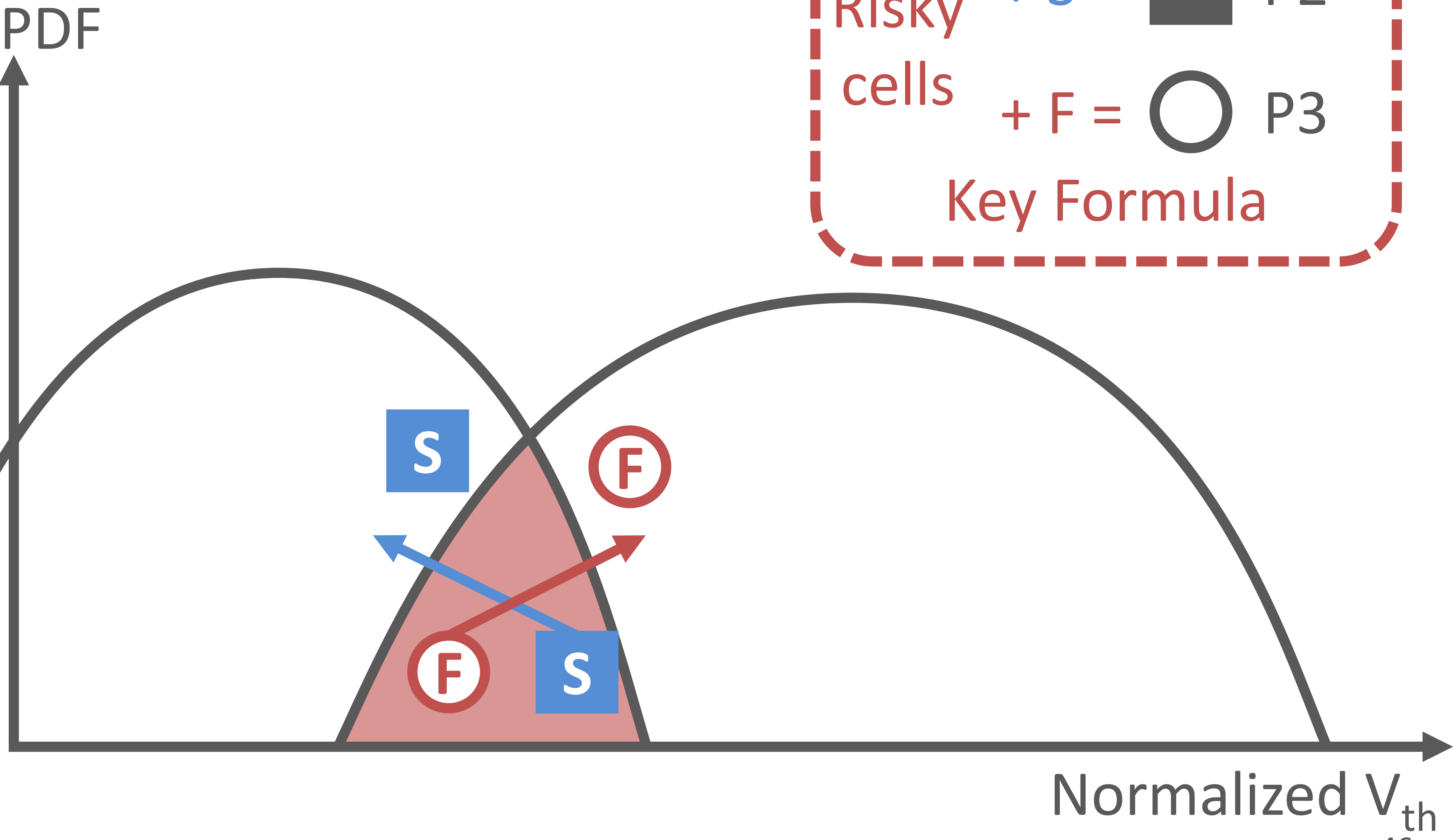


3. Guess Original States

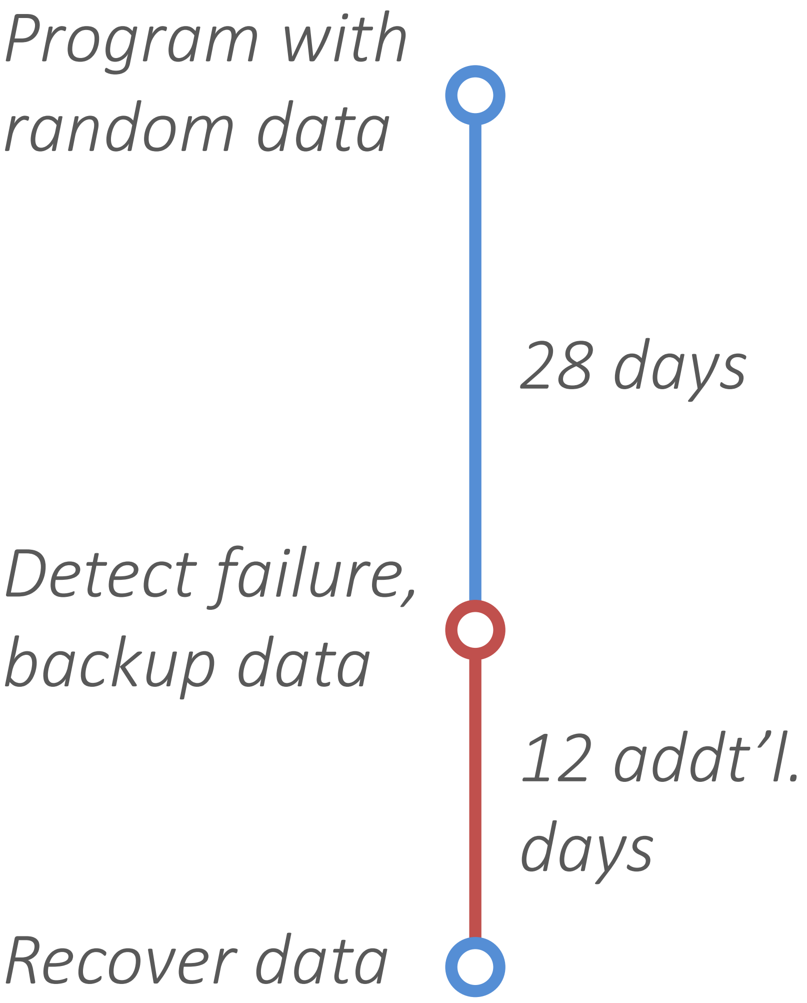
Risky cells + S = ■ P2

+ F = ○ P3

Key Formula



RFR Evaluation



Program with random data

28 days

Detect failure, backup data

12 addt'l. days

Recover data

- *Expect to eliminate 50% of raw bit errors*
- *ECC can correct remaining errors*

To understand the effects of retention loss:
- Characterize retention loss using real chips

Goal 1: Design a low-cost mechanism that dynamically finds the optimal read reference voltage

Goal 2: Design an offline mechanism to recover data after detecting uncorrectable errors

Conclusion

Problem: Retention loss reduces flash lifetime

Overall Goal: Extend flash lifetime at low cost

Flash Characterization: Developed an understanding of the effects of retention loss in real chips

Retention Optimized Reading: A low-cost mechanism that dynamically finds the optimal read reference voltage

- 64% lifetime ↑, 70.4% read latency ↓

Retention Failure Recovery: An offline mechanism that recovers data after detecting uncorrectable errors

- Raw bit error rate 50% ↓, reduces data loss

Data Retention in MLC NAND Flash Memory: Characterization, Optimization, and Recovery

Yu Cai, Yixin Luo, Erich F. Haratsch,
Ken Mai, Onur Mutlu*

*Carnegie Mellon University, *LSI Corporation*

Backup Slides

RFR Motivation

Data loss can happen in many ways

1. *High P/E cycle*
2. *High temperature → accelerates retention loss*
3. *High retention age (lost power for a long time)*

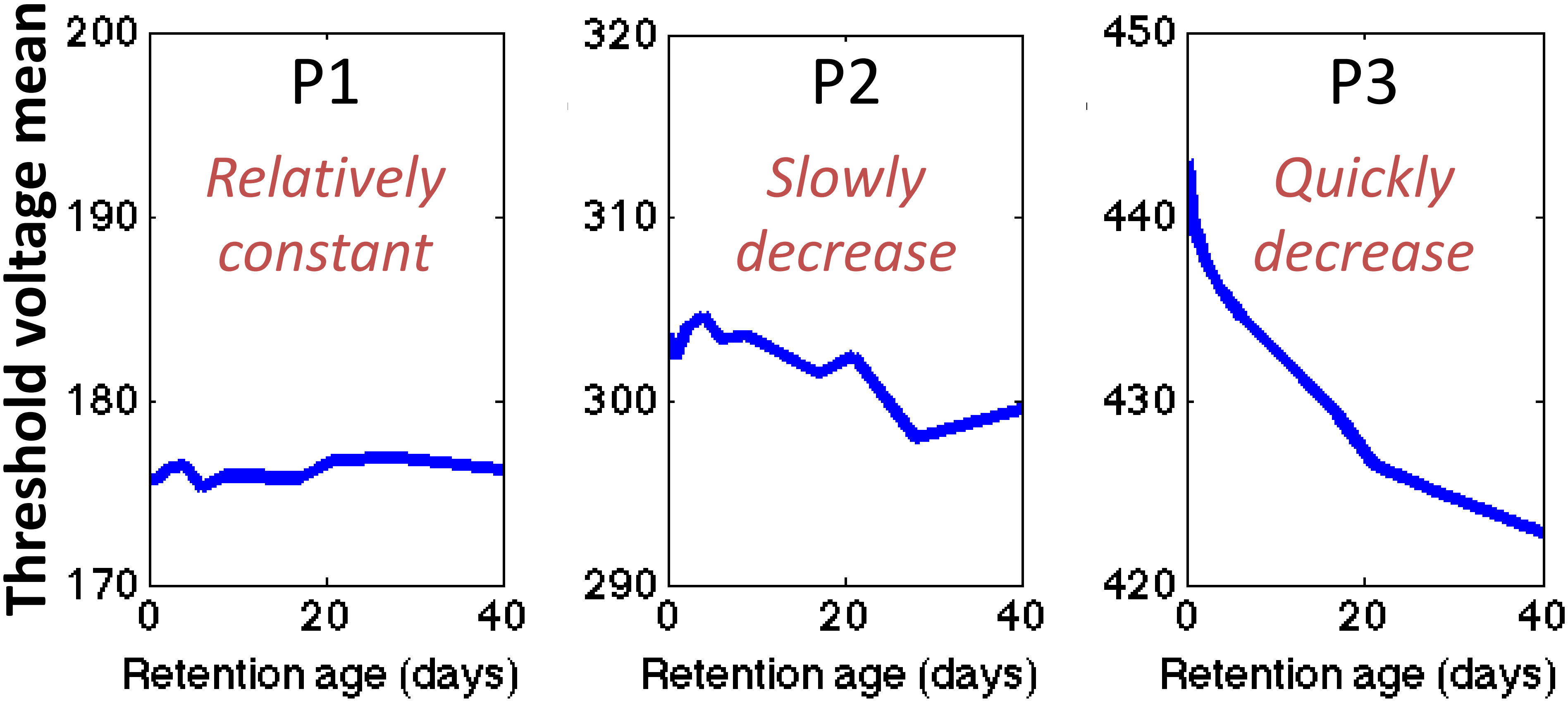
What if there are other errors?

Key: *RFR does not have to correct all errors*

Example:

- *ECC can correct 40 errors in a page*
- *Corrupted page has 20 retention errors, 25 other errors (45 total errors)*
- *After RFR: 10 retention errors, 30 other errors (40 total errors → ECC correctable)*

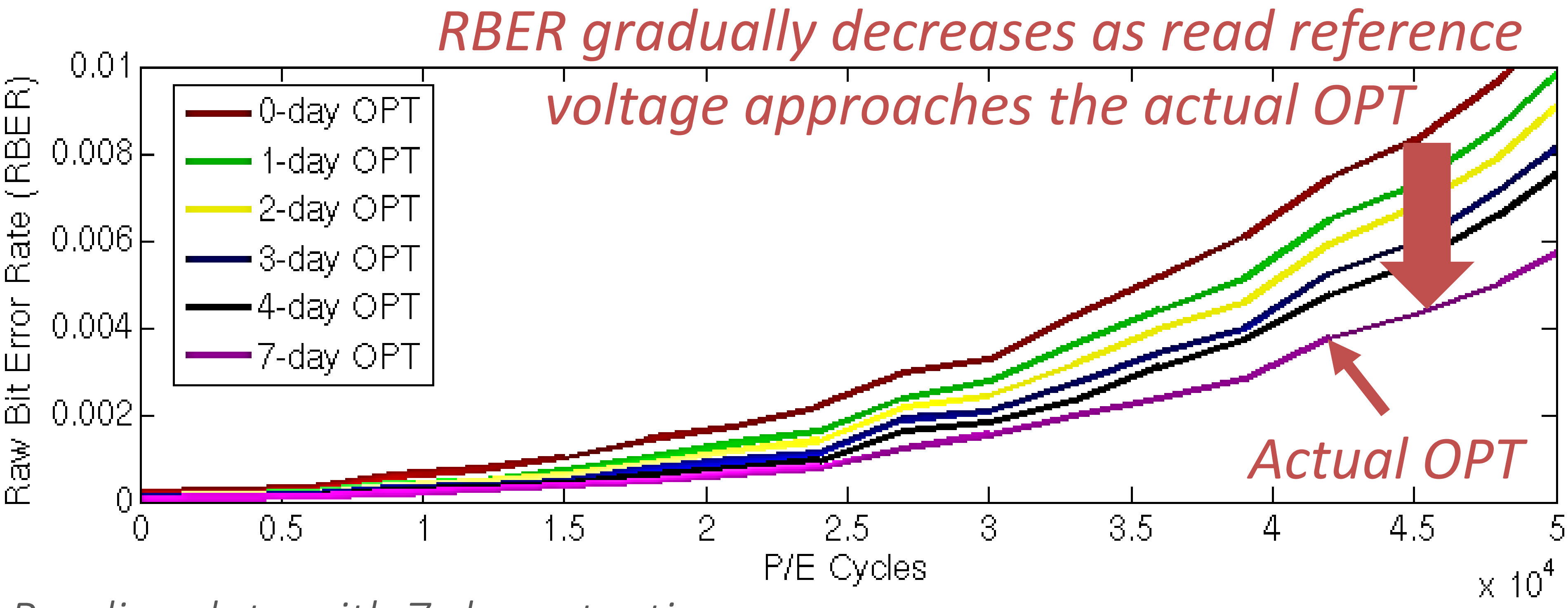
Threshold Voltage (V_{th}) Mean



Finding: V_{th} shifts faster in higher voltage states

SAFARI

Raw Bit Error Rate (RBER)



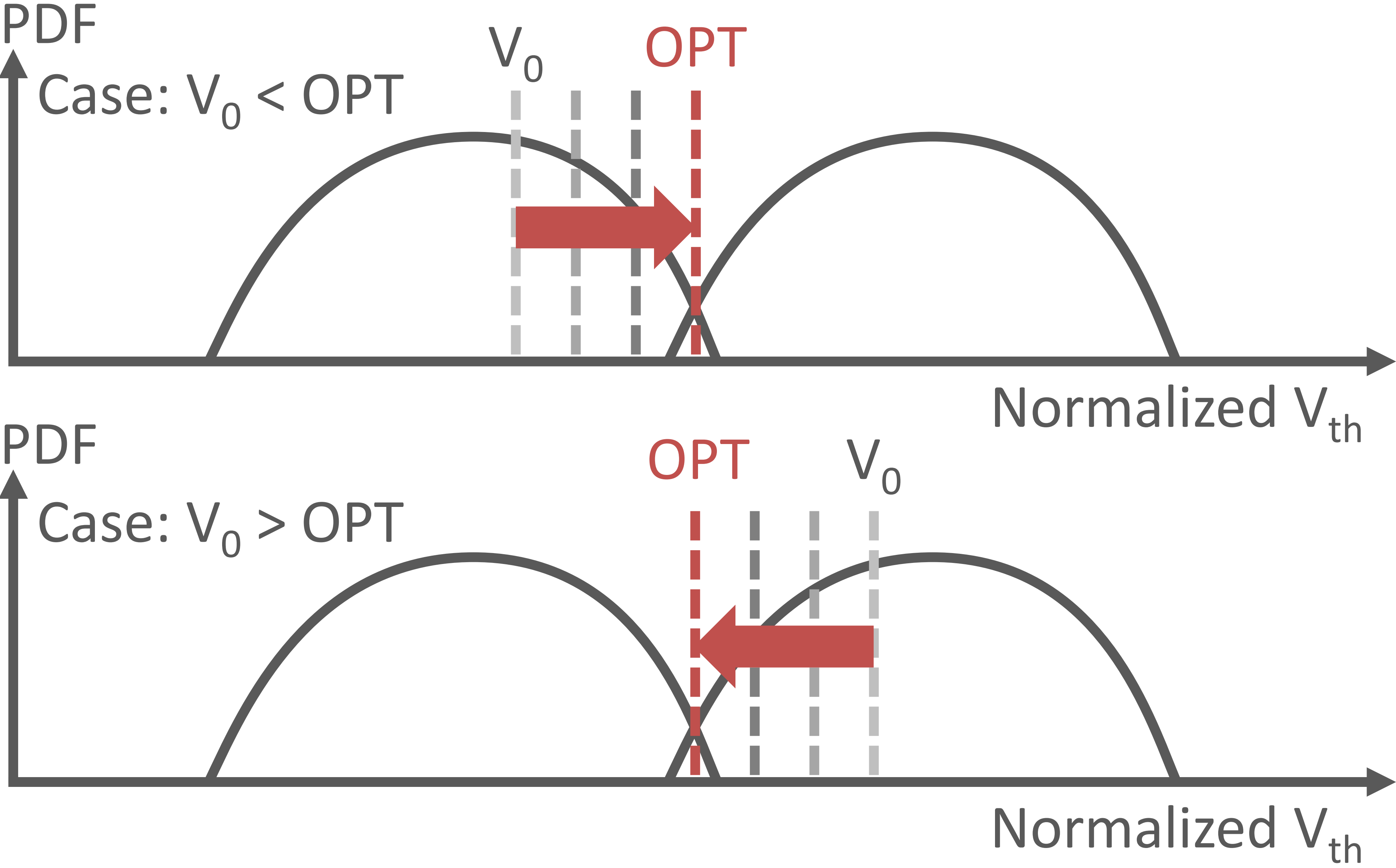
RBER gradually decreases as read reference voltage approaches the actual OPT

Actual OPT

Reading data with 7-day retention age.

Finding: The actual OPT achieves the lowest RBER

Online Pre-Optimization Algorithm



Online Pre-Optimization Algorithm

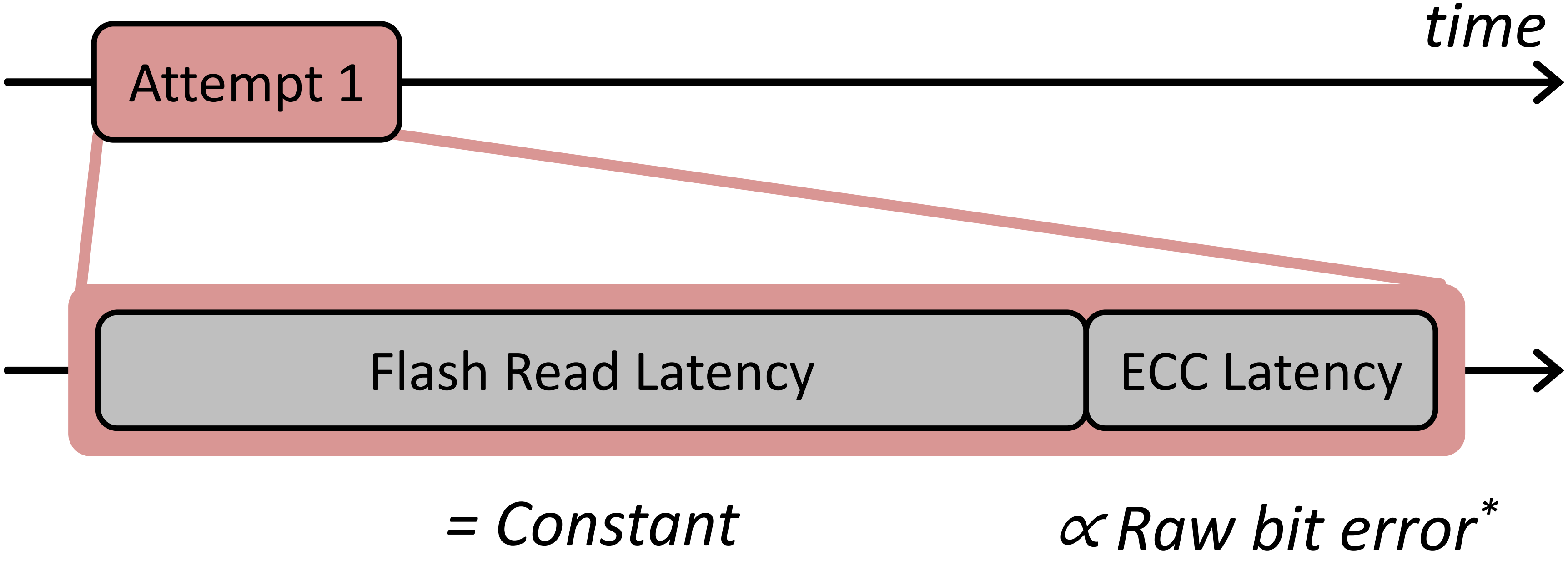
- ***Periodically** learn and record **OPT for page 255** as **per-block** starting read reference voltage (V_0)*
  - Page 255 has the shortest retention age
  - Other pages within the block have longer retention age and retention age will increase over time

- *Step 1:* *Read with V_{ref} = old $V_{0,}$ record RBER*
- *Step 2:* *Decrease $V_{ref} = V_{ref} - \Delta V^*$ compare RBER*
- *Step 3:* *Increase $V_{ref} = V_{ref} + \Delta V$ compare RBER*
- *Step 4:* *Record new $V_0 = V_{ref}$ | minimal RBER*

$^*\Delta V$ is the smallest step size for changing read reference voltage.

Naive Read-Retry Latency Diagnosis II

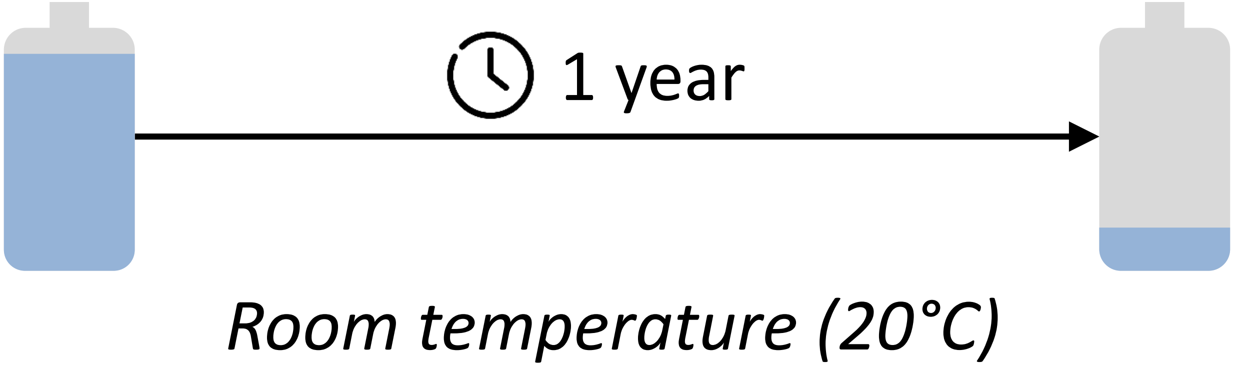
Read page A (Stage-0):



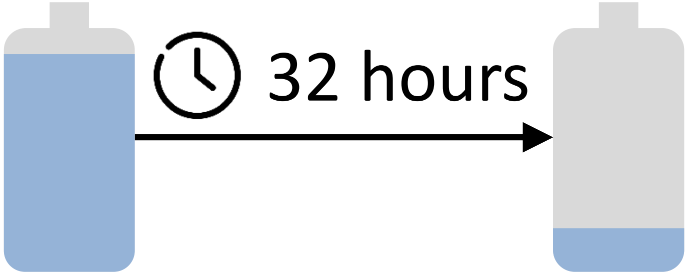
Observation: Average ECC latency ∝ RBER

*We provide detailed analysis of ECC latency in the paper.

Arrhenius Law



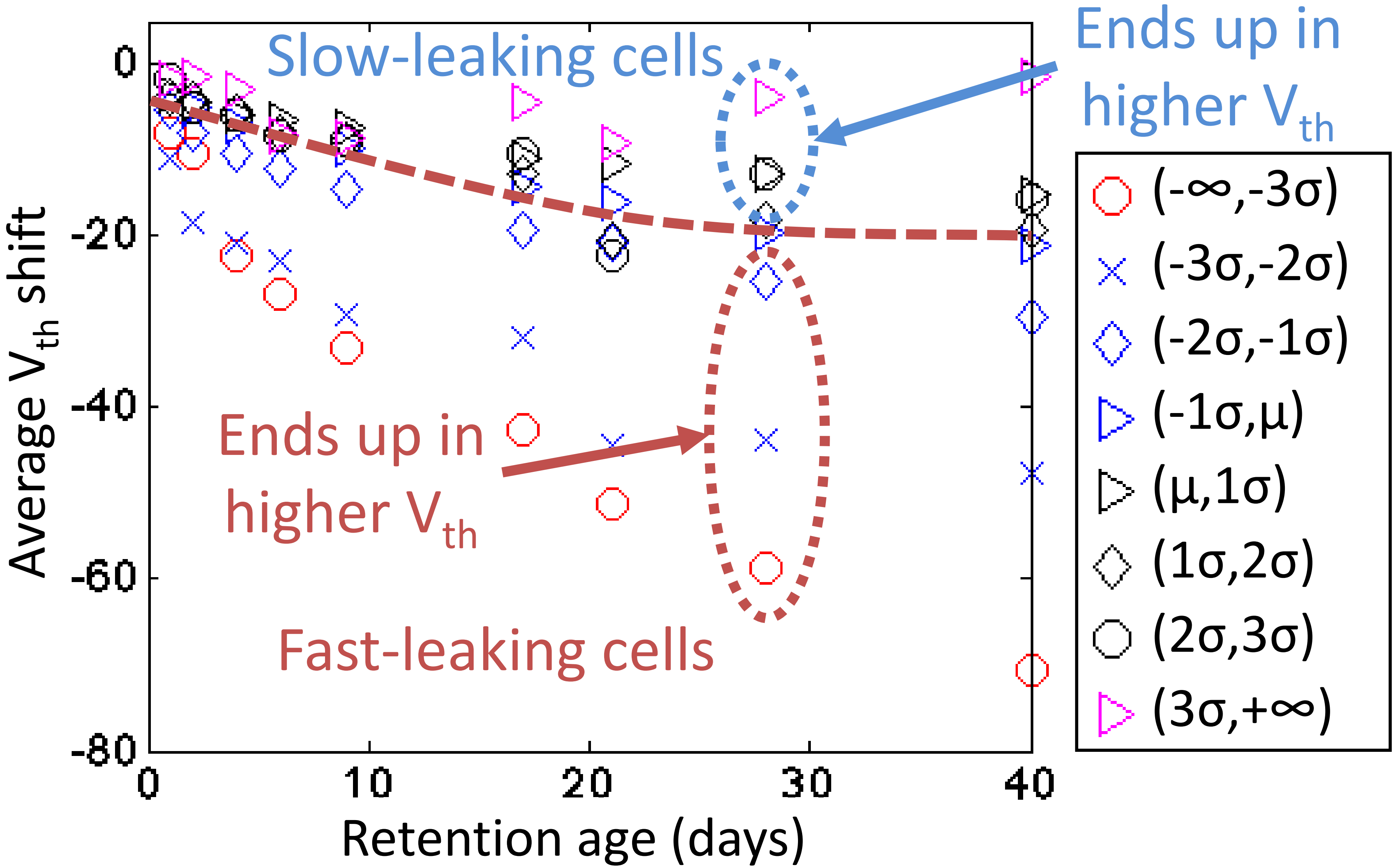
Room temperature (20°C)



High temperature (70°C)

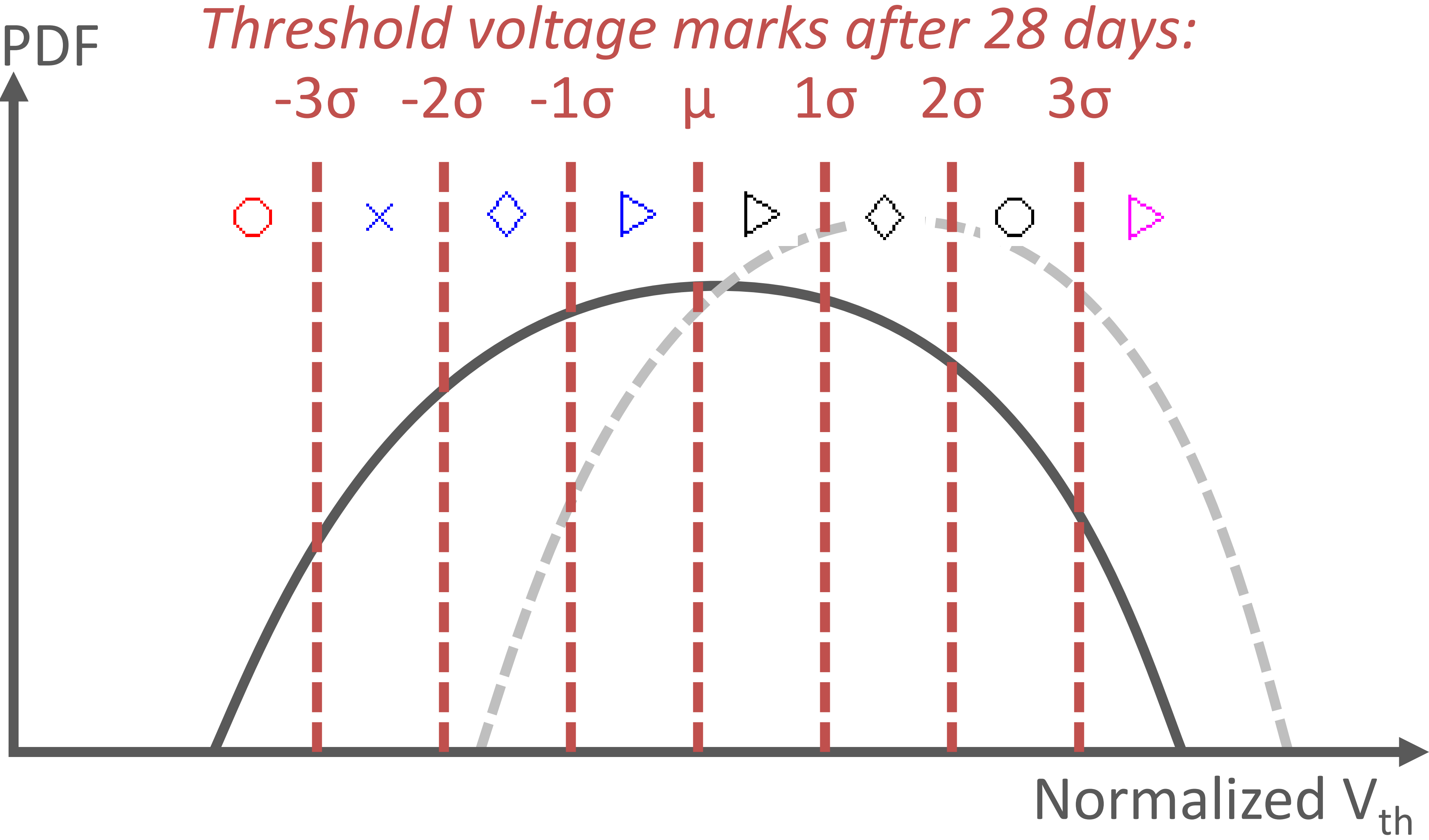
High temperature accelerates retention loss

Fast- and Slow-Leaking Cells

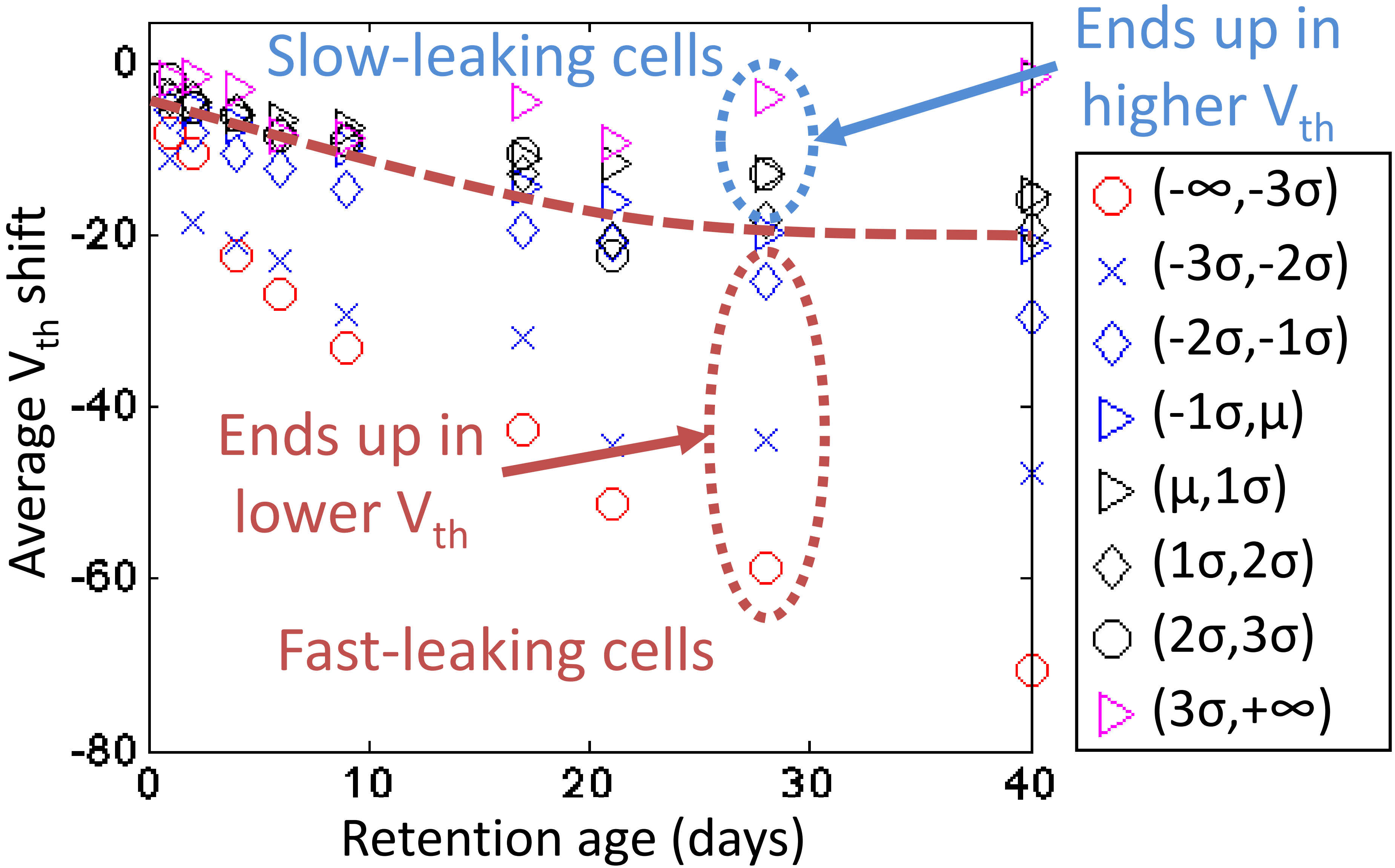


*Similar trends are found in P2 state, as shown in the paper.

Fast- and Slow-Leaking Cells



Fast- and Slow-Leaking Cells



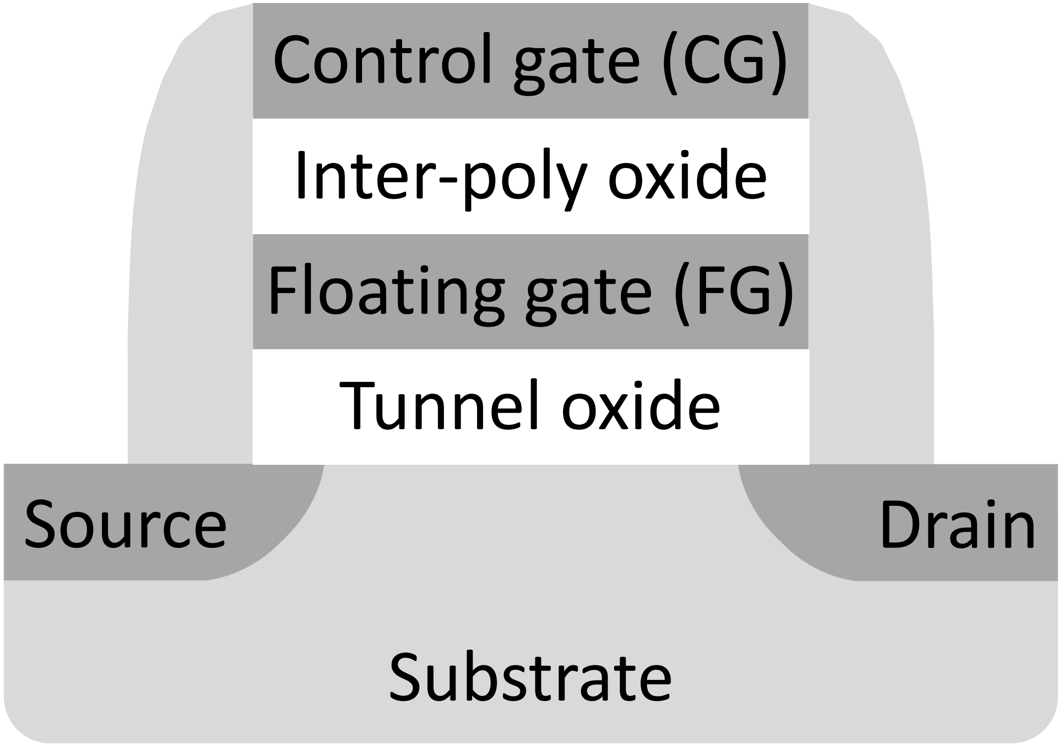
Slow-leaking cells

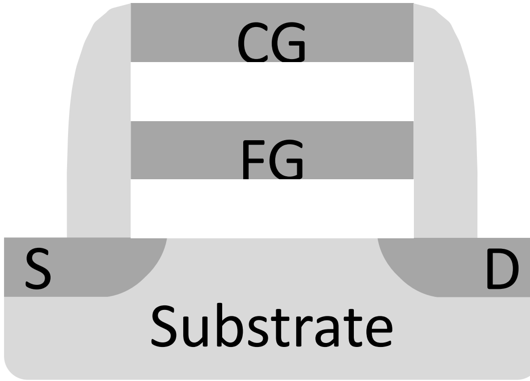
Ends up in higher V_{th}

Ends up in lower V_{th}

Fast-leaking cells

*Similar trends are found in P2 state, as shown in the paper.


Control gate (CG)
Inter-poly oxide
Floating gate (FG)
Tunnel oxide
Source
Drain
Substrate


CG
FG
S
D
Substrate