DESIGN, AUTOMATION & TEST IN EUROPE

25 – 27 March 2024 · Valencia, Spain

The European Event for Electronic
System Design & Test

SELCC:
Enhancing MLC Reliability and Endurance with Single Cell Error Correction Codes

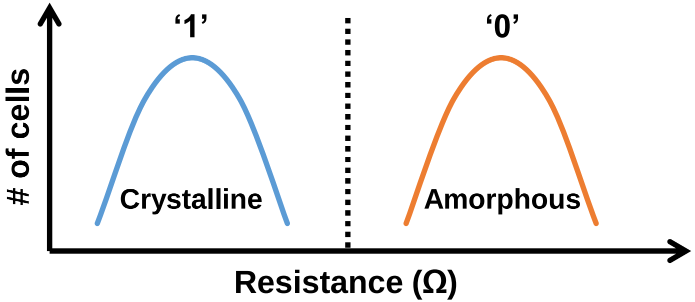
Yujin Lim, Dongwhee Kim, and Jungrae Kim

Sungkyunkwan University

01. Background

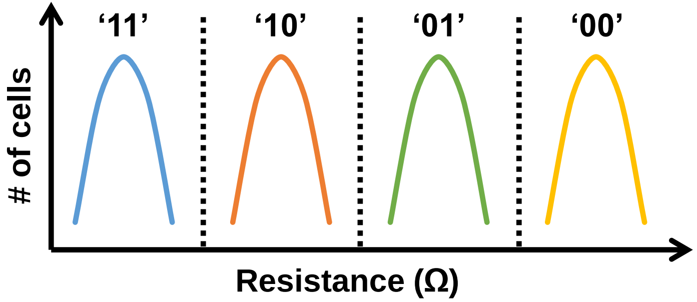
Phase Change Memory

- **Phase Change Memory (PCM)** is a emerging memory technology that can be a alternative to DRAM.
  - Currently, DRAM inherently faces several limitations.
    - voltality, static power consumption, limited scalability ...
  - PCM uses GST(Ge2Sb2Te5) alloy for storing data. This material has two different stable states and has a wide resistance range between the two stable states.

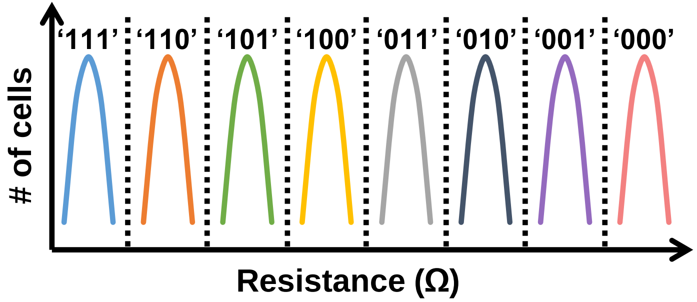


Multi-level Cell Memory

- PCM can provide non-volatility and higher capacity through **Multi-Level Cells (MLCs)**.
  - **Multi-level Cell (MLC)** memory divides the full resistance range into *n* levels so that it can store *log2(n)* bits in a single cell.



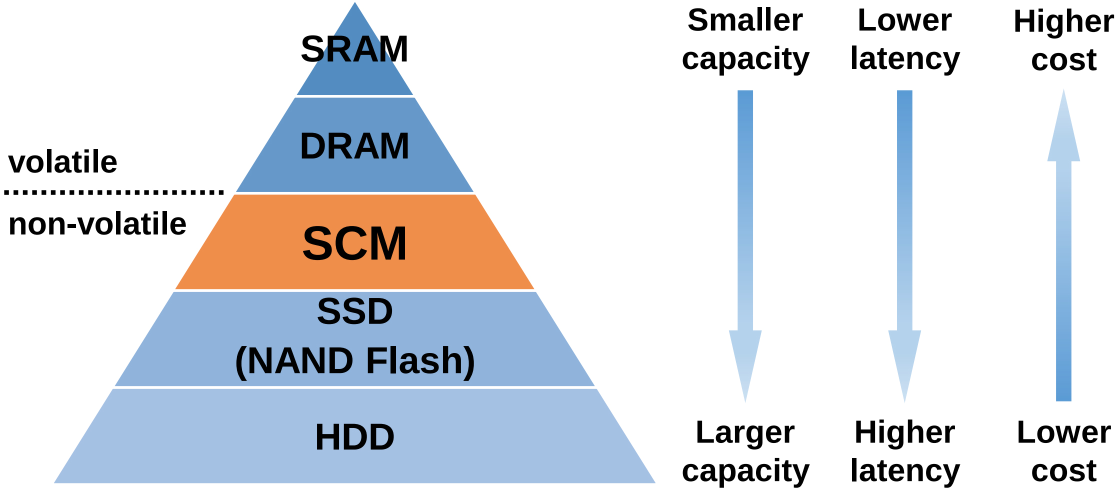
(a) 4-level cell (4LC)



(b) 8-level cell (8LC)

Use case of MLC PCM

- **Storage-Class Memory (SCM)** is the most promising use case of MLC PCM.
  - SCM bridges the gap between fast-but-volatile DRAM-based main memory and slower-but-non-volatile storage.



  - Using MLC PCM as SCM can boost the system performance with its high capacity.

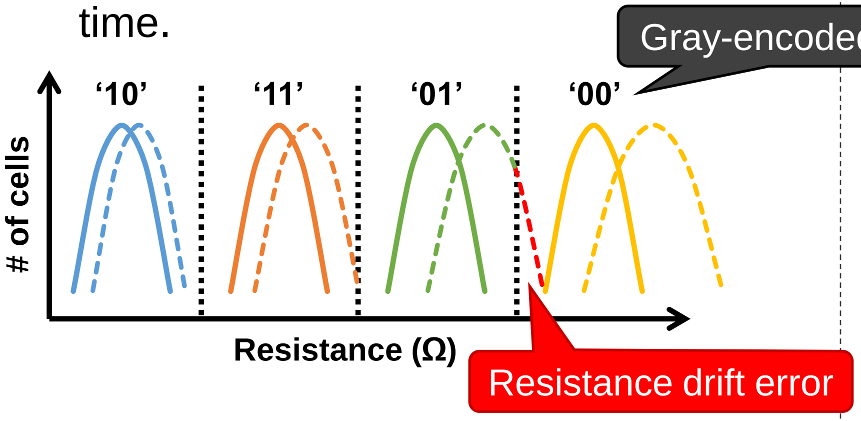
Challenges of MLC PCM

- However, utilizing MLC PCM as SCM poses several challenges.
  - MLC PCM faces issues of **reliability** and **reduced endurance** due to numerous physical phenomena.

MLC PCM Errors

① Limited Magnitude Errors

- **Resistance drift** : A cell's resistance value may increase with time.



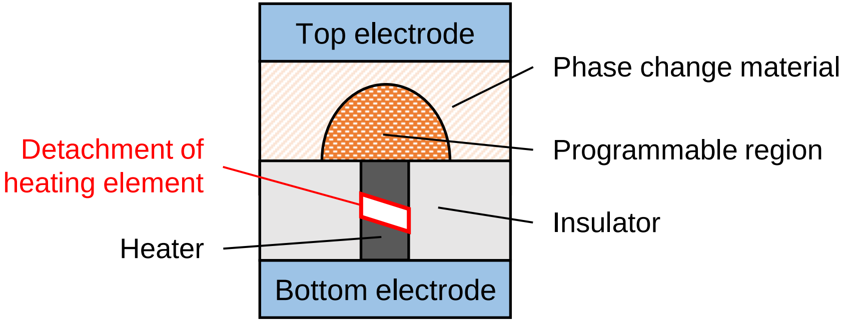
Gray-encoded data

Resistance drift error

- With Gray coding, single-level shift results in just a single-bit error.

② Unlimited Magnitude Errors

- **Cell wear-out** is the deterioration of a memory cell's ability to switch states due to repeated use.

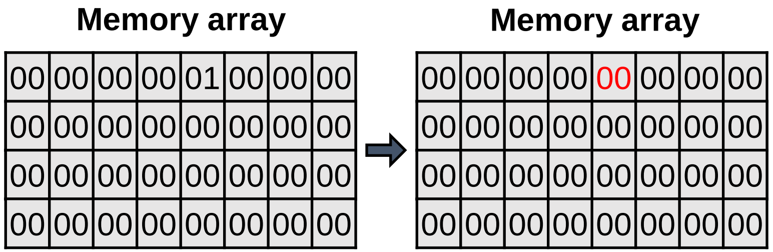


- When a cell wears out, its resistance transitions to the maximum level and lose its ability to store data.

The Impact of Errors on MLC Memory

① Reliability

- Drifts in a cell's resistance can compromise data integrity over time.



Memory array

00	00	00	00	01	00	00	00
00	00	00	00	00	00	00	00
00	00	00	00	00	00	00	00
00	00	00	00	00	00	00	00

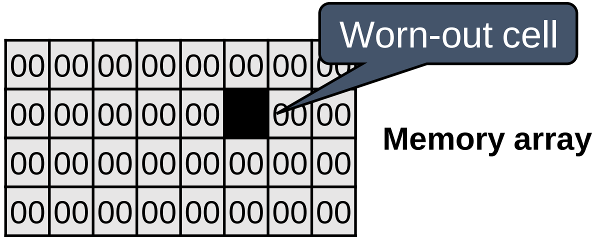
Memory array

00	00	00	00	00	00	00	00
00	00	00	00	00	00	00	00
00	00	00	00	00	00	00	00
00	00	00	00	00	00	00	00

- This issue can be exacerbated in Multi-Level Cell (MLC) memory, where the resistance ranges for each level are closer together.

② Endurance

- Over time, the repeated phase transitions can lead to wear-out of memory cells.



Worn-out cell

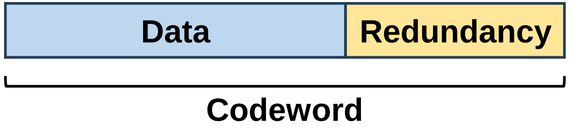
00	00	00	00	00	00	00	00
00	00	00	00	00	■	00	00
00	00	00	00	00	00	00	00
00	00	00	00	00	00	00	00

Memory array

- According to IRDS, PCM (SLC) can endure only 10^9 write cycles even in 2034.

Prior Solutions for MLC Memory Errors

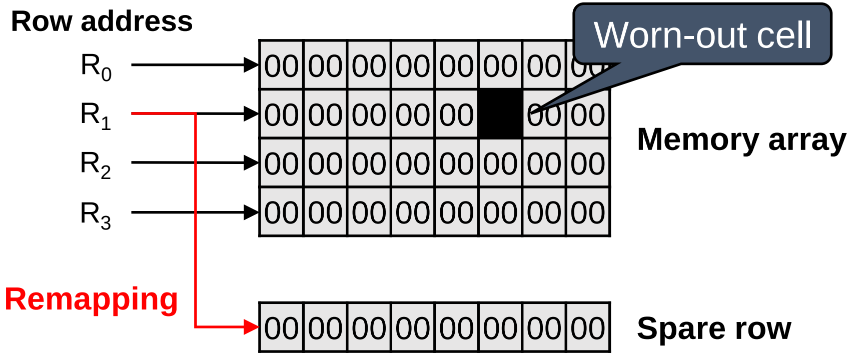
① Error Correction Codes (ECC)



- Conventional Single Error Correcting - Double Error Detecting (SEC-DED) ECC **lacks correction capability** for MLC memory.
- Low Density Parity Check (LDPC) codes, which is commonly used in MLC NAND, are inadequate for Storage Class Memory (SCM) due to its **long latency** from iterative decoding.

② Remapping

- When worn-out cells are identified, data is remapped to spare cells.
- The unit of remapping can be either a memory block or a row.



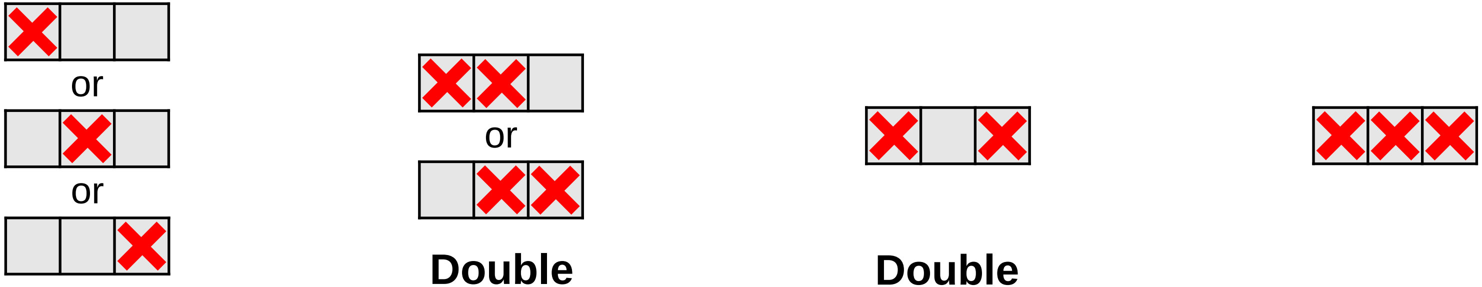
02. Proposed Method

Overview of Proposal

- Our approach, SELCC, **enhances both the reliability and endurance** of MLC memory **solely using Error Correction Codes (ECC)**.
  - SELCC uses the same redundancy ratio with previous ECC schemes.
    ➔ **No redundancy overheads**
  - SELCC can recover data from resistance drift with same redundancy as the traditional ECC schemes (e.g., SEC-DED). ➔ **Better reliability**
  - SELCC can also recover data of worn-out cells without degrading performance.
    ➔ **Higher endurance**

Motivation

- What is the most suitable Error Correction Codes (ECC) for MLC memory?
- Considering the **cell boundary** of MLC memory, it is important to **focus on correcting single-cell errors.**
    - It is rare for errors to cross a cell boundary.
- There are **7** possible single-cell error patterns in 8LC memory.



Single Error (SE)

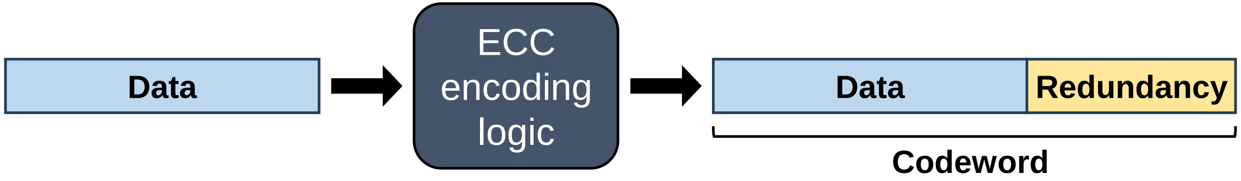
Double Adjacent Error (DAE)

Double Non-adjacent Error (DNE)

Triple Error (TE)

Error Correction Codes

- **Error Correction Codes (ECC)** can detect and corrects errors in data.
  - ECC adds redundant data (redundancy) to the original data.



  - There are various types of ECC, each with its own method of detecting and correcting errors in data.

ECC - Linear Block Codes

- **Error Correction Codes (ECC) - Linear Block Codes**
  - **ECC encoding** by generating a codeword.

*G-matrix: Generator matrix
*H-matrix: Parity check matrix

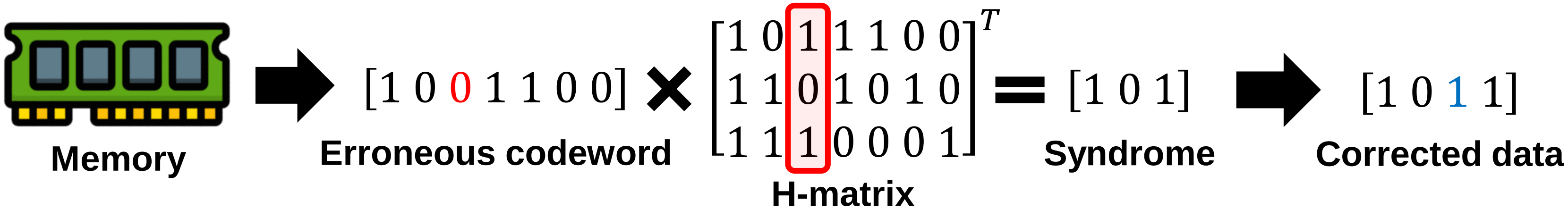
$$\underset{\text{Data}}{[1\ 0\ 1\ 1]} \times \underset{\text{G-matrix}}{\begin{bmatrix}1&0&0&0&1&1&1\\0&1&0&0&0&1&1\\0&0&1&0&1&0&1\\0&0&0&1&1&1&0\end{bmatrix}} = \underset{\text{Codeword}}{[\overbrace{1\ 0\ 1\ 1}^{\text{Data}}\ \overbrace{1\ 0\ 0}^{\text{Redundancy}}]} \rightarrow \text{Memory}$$

  - **ECC decoding** using syndromes.

$$\text{Memory} \rightarrow \underset{\text{Erroneous codeword}}{[1\ 0\ 0\ 1\ 1\ 0\ 0]} \times \underset{\text{H-matrix}}{\begin{bmatrix}1&0&1&1&1&0&0\\1&1&0&1&0&1&0\\1&1&1&0&0&0&1\end{bmatrix}^T} = \underset{\text{Syndrome}}{[1\ 0\ 1]} \rightarrow \underset{\text{Corrected data}}{[1\ 0\ 1\ 1]}$$

ECC Decoding - Linear Block Codes

- **Error Correction Codes (ECC) - Linear Block Codes (Decoding)**
  - An error in the nth bit produces a syndrome equal to the nth column in the H-matrix.



Memory →

$$[1\ 0\ 0\ 1\ 1\ 0\ 0] \times \begin{bmatrix}1&0&1&1&1&0&0\\1&1&0&1&0&1&0\\1&1&1&0&0&0&1\end{bmatrix}^T = [1\ 0\ 1] \rightarrow [1\ 0\ 1\ 1]$$

Memory

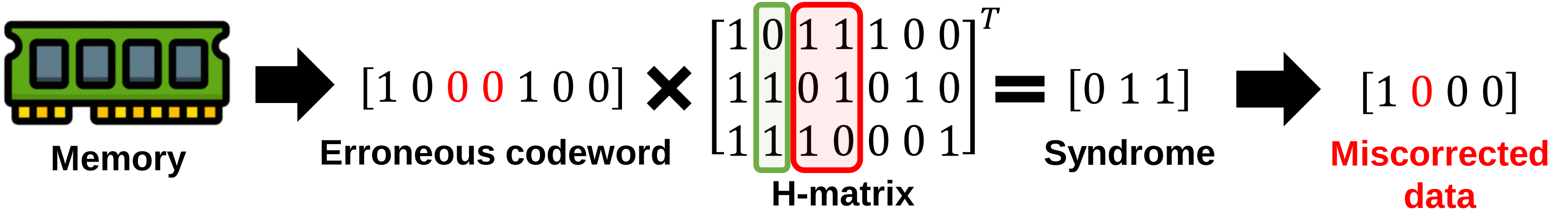
Erroneous codeword

H-matrix

Syndrome

Corrected data

  - Multi-bit error produces a syndrome equal to the **XOR of every column** in the error positions in the H-matrix.



Memory →

$$[1\ 0\ 0\ 0\ 1\ 0\ 0] \times \begin{bmatrix}1&0&1&1&1&0&0\\1&1&0&1&0&1&0\\1&1&1&0&0&0&1\end{bmatrix}^T = [0\ 1\ 1] \rightarrow [1\ 0\ 0\ 0]$$

Memory

Erroneous codeword

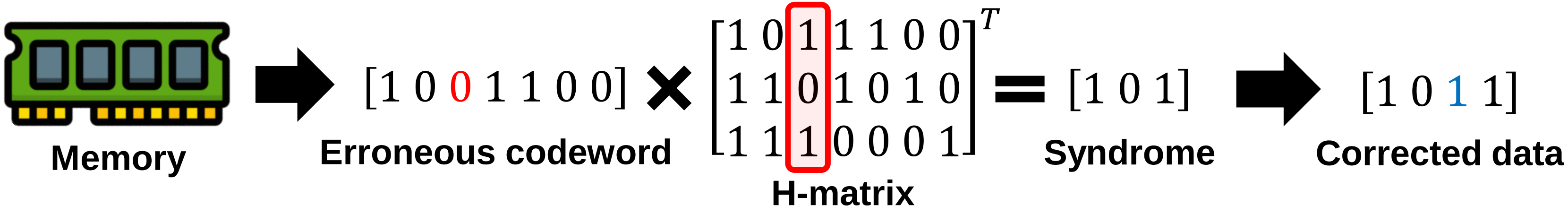
H-matrix

Syndrome

Miscorrected data

ECC Decoding - Linear Block Codes

- **Error Correction Codes (ECC) - Linear Block Codes**
  - An error in the nth bit produces a syndrome equal to the nth column in the H-matrix.



$$[1\ 0\ 0\ 1\ 1\ 0\ 0] \times \begin{bmatrix}1&0&1&1&1&0&0\\1&1&0&1&0&1&0\\1&1&1&0&0&0&1\end{bmatrix}^T = [1\ 0\ 1] \rightarrow [1\ 0\ 1\ 1]$$

  - Multi-bit error produces a syndrome equal to the **XOR of every column** in the error positions in the H-matrix.

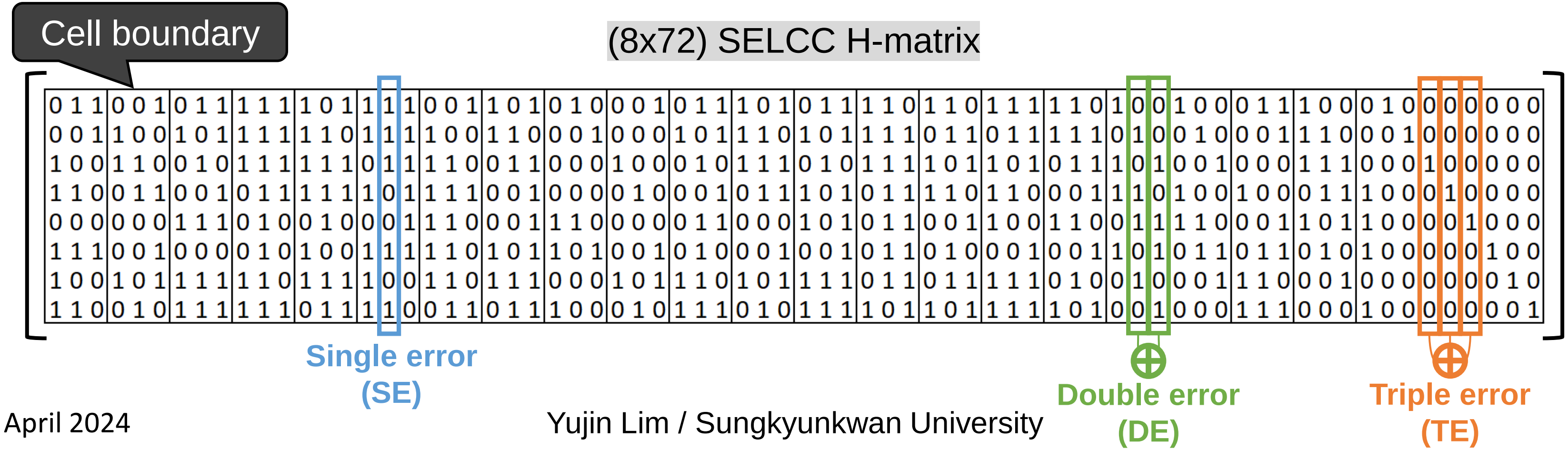
If the syndrome caused by a multi-bit error differs from that of a single-bit error and also unique, it can be identified.

Proposed Method: SELCC

- If **every single-cell error patterns has its own unique syndrome**, we can identify and correct the errors.
- Is it possible?
  - In 64-bit data + 8-bit redundancy configuration (8LC memory), we have 2^8 - 1 = **255** nonzero syndromes. (zero syndrome indicates no errors)
    - # of single-cell errors per cell = 7
    - # of 8LC cells in a 72-bit data block = 24
    - # of single-cell errors in a 72-bit data block = 7 x 24 = **168**
- Therefore, It is possible to map **168** error patterns onto unique syndromes among **255** nonzero syndromes.
  - 72 single errors (SEs), 72 double errors (DEs), and 24 triple errors (TEs)

H-matrix of SELCC

- The H-matrix should adhere to the following properties to correct all single-cell errors.
  1. Every column must be nonzero.
  2. Each column must be unique. ➔ **for correcting single errors (SEs) in a cell**
  3. The sum of two or three columns within any cell boundaries must be nonzero and unique. ➔ **for correcting double errors (DEs) or triple errors (TEs) in a cell**



Cell boundary

(8x72) SELCC H-matrix

$$\left[\begin{array}{ccc|ccc|ccc|ccc|ccc|ccc|ccc|ccc|ccc|ccc|ccc|ccc|ccc|ccc|ccc|ccc|ccc|ccc|ccc|ccc|ccc|ccc|ccc|ccc}
0&1&1&0&0&1&0&1&1&1&1&1&1&0&1&1&1&1&0&0&1&1&0&1&0&1&0&0&0&1&0&1&1&1&0&1&0&1&1&1&1&0&1&1&0&1&1&1&1&1&0&1&0&0&1&0&0&0&1&1&1&0&0&0&1&0&0&0&0&0&0&0\\
0&0&1&1&0&0&1&0&1&1&1&1&1&1&0&1&1&1&1&0&0&1&1&0&0&0&1&0&0&0&1&0&1&1&1&0&1&0&1&1&1&1&0&1&1&0&1&1&1&1&1&0&1&0&0&1&0&0&0&1&1&1&0&0&0&1&0&0&0&0&0&0\\
1&0&0&1&1&0&0&1&0&1&1&1&1&1&1&0&1&1&1&1&0&0&1&1&0&0&0&1&0&0&0&1&0&1&1&1&0&1&0&1&1&1&1&0&1&1&0&1&0&1&1&1&0&1&0&0&1&0&0&0&1&1&1&0&0&0&1&0&0&0&0&0\\
1&1&0&0&1&1&0&0&1&0&1&1&1&1&1&1&0&1&1&1&1&0&0&1&0&0&0&0&1&0&0&0&1&0&1&1&1&0&1&0&1&1&1&1&0&1&1&0&0&0&1&1&1&0&1&0&0&1&0&0&0&1&1&1&0&0&0&1&0&0&0&0\\
0&0&0&0&0&0&1&1&1&0&1&0&0&1&0&0&0&1&1&1&0&0&0&1&1&1&0&0&0&0&0&1&1&0&0&0&1&0&1&0&1&1&0&0&1&1&0&0&1&1&0&0&1&1&1&1&0&0&0&1&1&0&1&1&0&0&0&0&1&0&0&0\\
1&1&1&0&0&1&0&0&0&0&1&0&1&0&0&1&1&1&1&1&0&1&0&1&1&0&1&0&0&1&0&1&0&0&0&1&0&0&1&0&1&1&0&1&0&0&0&1&0&0&1&1&0&1&0&1&1&0&1&1&0&1&0&1&0&0&0&0&0&1&0&0\\
1&0&0&1&0&1&1&1&1&1&1&0&1&1&1&1&0&0&1&1&0&1&1&1&0&0&0&1&0&1&1&1&0&1&0&1&1&1&1&0&1&1&0&1&1&1&1&1&0&1&0&0&1&0&0&0&1&1&1&0&0&0&1&0&0&0&0&0&0&0&1&0\\
1&1&0&0&1&0&1&1&1&1&1&1&0&1&1&1&1&0&0&1&1&0&1&1&1&0&0&0&1&0&1&1&1&0&1&0&1&1&1&1&0&1&1&0&1&1&1&1&1&0&1&0&0&1&0&0&0&1&1&1&0&0&0&1&0&0&0&0&0&0&0&1
\end{array}\right]$$

Single error (SE)

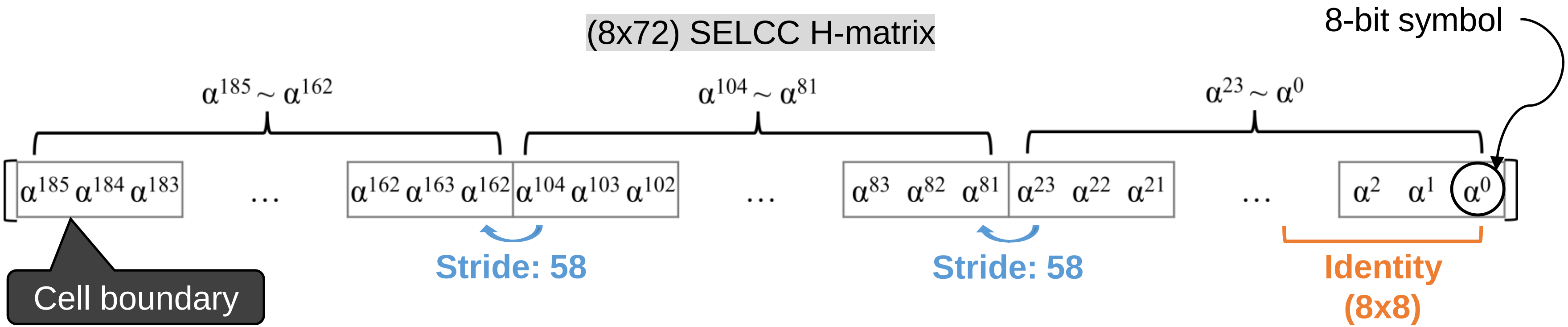
Double error (DE)

Triple error (TE)

H-matrix Construction

- Construction of **SELCC** H-matrix
  - Identity matrix at the right most side. (Systematic code)
  - Allocate consequent 8-bit symbols in a GF(2^8) with a specific size of stride to meet the properties.

(8x72) SELCC H-matrix


$\alpha^{185} \sim \alpha^{162}$
$\alpha^{104} \sim \alpha^{81}$
$\alpha^{23} \sim \alpha^{0}$
$\alpha^{185}\ \alpha^{184}\ \alpha^{183}$
…
$\alpha^{162}\ \alpha^{163}\ \alpha^{162}$
$\alpha^{104}\ \alpha^{103}\ \alpha^{102}$
…
$\alpha^{83}\ \alpha^{82}\ \alpha^{81}$
$\alpha^{23}\ \alpha^{22}\ \alpha^{21}$
…
$\alpha^{2}\ \alpha^{1}\ \alpha^{0}$

8-bit symbol

Cell boundary

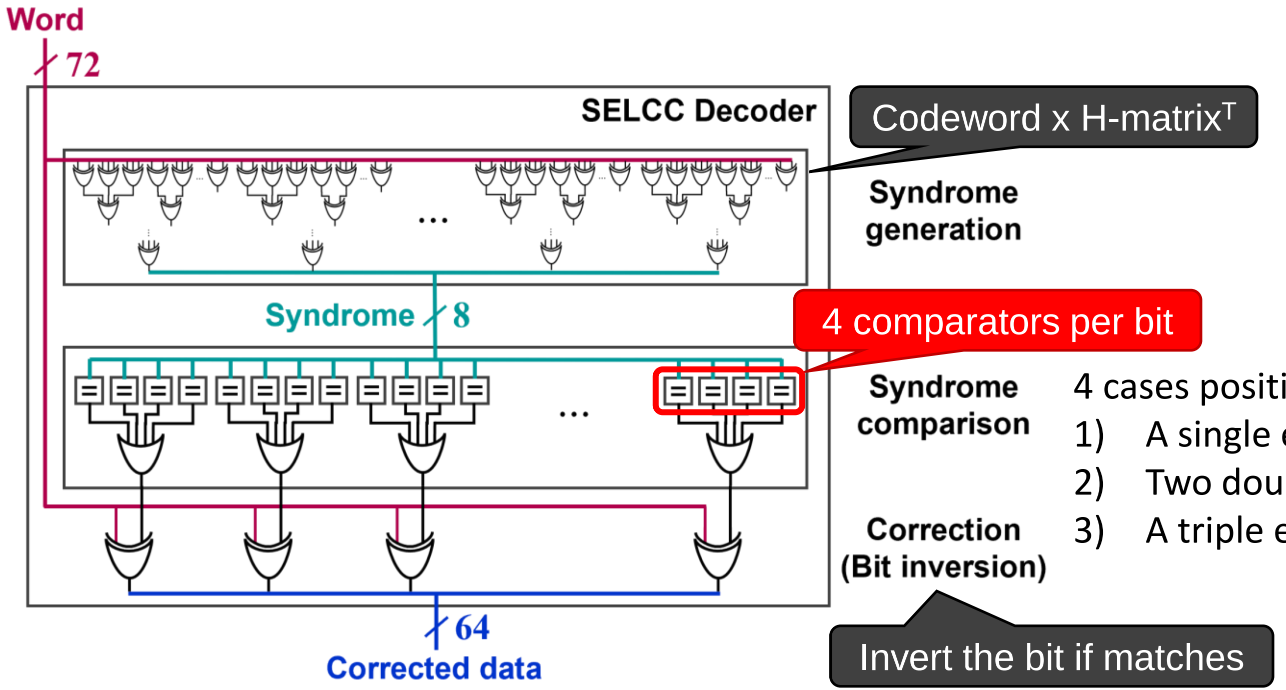
Stride: 58

Stride: 58

Identity
(8x8)

Hardware Implementation - Decoder

- Hardware implementation of **SELCC** decoder



Codeword x H-matrix^T

4 comparators per bit

Invert the bit if matches

4 cases position P needs to be corrected

1) A single error (SE) at P
2) Two double errors (DEs) including P
3) A triple error (TE) including P

Correction Capability of SELCC

- **S**ingle **E**ight-**L**evel **C**ell **C**orrecting **(SELCC)** ECC is tailored to **correct every single-cell error in 8LC memories.**

Scheme	Possible error patterns on a single cell				Redundancy ratio
	Single error (SE)	Double adjacent error (DAE)	Double non-adjacent error (DNE)	Triple adjacent error (TAE)	
SEC-DED	O	X	X	X	12.5%
SEC-DAEC	O	O	X		
SEC-DAEC-TAEC	O	O	X		
IP-DAEC	X	O	O	O	[illegible]
SELCC (proposed)	**O**	**O**	**O**	**O**	**12.5%**

The only scheme that can correct every single 8LC error with redundancy ratio of 12.5%!

03. Evaluation

Evaluation

- Evaluation metrics
  - Reliability
  - Endurance
  - Hardware overheads

Evaluation - Reliability

- Methodology
  - Injects errors at random positions within a 72-bit word (64-bit data + 8-bit redundancy)
  - The simulation introduces either a **single-cell error** or **a double-cell error**.
    - Note that the number of error bits is up to 3 in each cell.
  - Each error scenario underwent 1 million simulation iterations.

Evaluation - Reliability

- Correction capabilities (higher is better) against **single 8LC errors**.
  - Silent data corruption (SDC) probabilities (lower is better) are noted in parenthesis.

Error scenario	SEC-DED	SEC-DAEC	SEC-DAEC-TAEC	IP-DAEC	**SELCC**
Single error	100% (0%)			67.97% (0%)	**100% (0%)**
Double error	0% (0%)	66.69% (16.69%)	66.63% (29.21%)	100% (0%)	**100% (0%)**
Triple error	0% (0%)	0% (66.2%)	100% (0%)		**100% (0%)**

SELCC provides **100%** correction capability & **0%** SDC probability at every error scenario.

Evaluation - Reliability

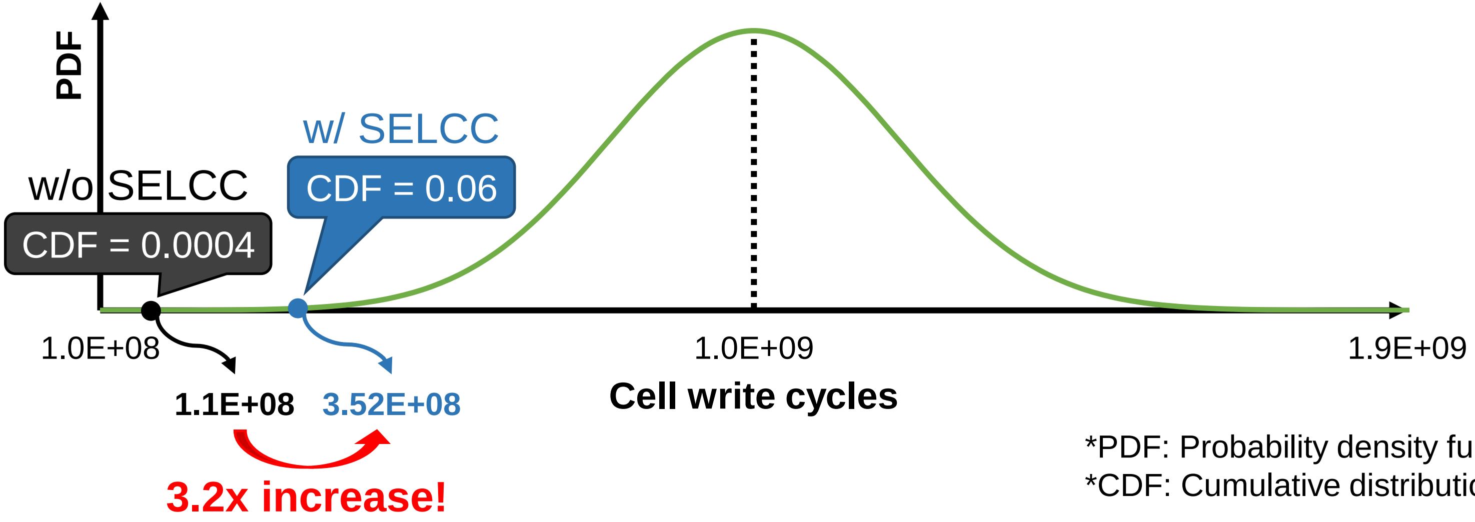
- Silent data corruption (SDC) probabilities (lower is better) against **double 8LC errors**.
  - The portion of correctable errors is either zero or negligible.

Error scenario	SEC-DED	SEC-DAEC	SEC-DAEC-TAEC	IP-DAEC	**SELCC**
Single error + Single error	0%	54.19%	83.91%	81.32%	**65.48%**
Single error + Triple error	0%	55.63%	83.67%	71.74%	**66.85%**
Double error + Triple error	71.98%	56.15%	82.40%	58.17%	**62.74%**

SELCC shows reasonable SDC rate considering 100% correction capability against single cell errors

Evaluation - Endurance

- **SELCC** provides **3.2x** increase in endurance.
  - We assume that each MLC can withstand **10^9** write cycles on average, nuanced by cell variation. (Coefficient of variation (COV) at 20%)
  - Find maximum cell write cycles where **99.99%** of 72-bit words have no defective cells.



*PDF: Probability density function
*CDF: Cumulative distribution function

Evaluation - Hardware Overheads

- **SELCC** incurs similar hardware overheads to existing ECCs.

		SEC-DED	SEC-DAEC	SEC-DAEC-TAEC	IP-DAEC	**SELCC**
Encoder	Latency (ns)	0.40	0.40	0.40	0.40	**0.40**
	Area (um^2)	115	135	103	95	**147**
	Power (mW)	0.21	0.25	0.18	0.18	**0.29**
Decoder	Latency (ns)	0.43	0.49	0.48	0.46	**0.52**
	Area (um^2)	1020	1841	2589	1684	**2704**
	Power (mW)	2.28	6.63	10.68	6.71	**10.60**

Negligible overheads

Summary

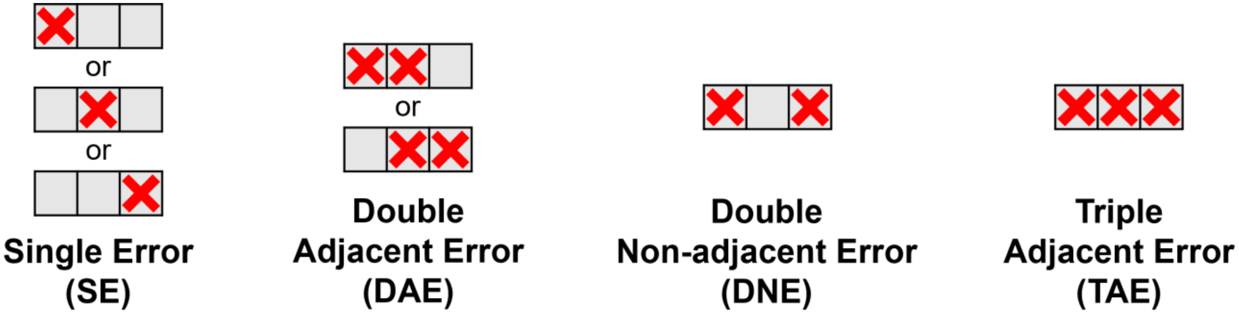
- **Observation**
  - Using **MLC PCM** as SCM can boost the system performance with its high capacity, however, it suffers from reduced endurance and reliability.
- **Proposed Idea: SELCC**
  - **SELCC can correct all single-cell errors in 8LC memory**, whether arising from a slight drift in resistance or a complete cell wear-out, by leveraging the unused syndromes in SEC-DED. **(No redundancy overheads!)**
- **Contribution**
  - **SELCC can significantly enhance the reliability** of 8LC memory, surpassing previous ECC solutions even with same redundancy.
  - **SELCC can also increase the endurance** of 8LC memory by **3.2 times**, while incurring negligible overheads.

Thank You

Q&A

Prior Works

- (72,64) Single error correction - Double error detection (SEC-DED)
  - **cannot correct DAE/DNE/TAE.**
- (72,64) Single error correction - Double adjacent error correction (SEC-DAEC)
  - **cannot correct DNE/TAE.**
- (72,64) Single error correction - Double adjacent error correction - Triple adjacent error correction (SEC-DAEC-TAEC)
  - **cannot correct DNE.**



References of Prior Works

- SEC-DED
  - J. A. Maestro, P. Reviriego, L. Xiao, S. Liu, and A. Sanchez-Macian, “Odd-weight-column SEC–DED–TAED Codes,” Electronics Letters, vol. 52, 2016.
- SEC-DAEC
  - S. Tripathi, J. Jana, J. Samanta, A. Anand, C. Kumar, and G. Raj, “FPGA and Asic Implementation of SEC-DED-DAEC Codes for SRAM Applications,” in the 2nd ICCDC, 2019.
- SEC-DAEC-TAEC
  - L.-J. Saiz-Adalid, P. Reviriego, P. Gil, S. Pontarelli, and J. A. Maestro, “MCU Tolerance in SRAMs Through Low-Redundancy Triple Adjacent Error Correction,” IEEE TVLSI, vol. 23, 2015.
- IP-DAEC
  - S. Liu, P. Reviriego, and F. Lombardi, “Codes for Limited Magnitude Error Correction in Multilevel Cell Memories,” IEEE Transactions on Circuits and Systems I: Regular Papers, vol. 67, 2020.

An Example of SELCC H-matrix

- (8x72) H-matrix in binary form.

```
0 1 1 0 0 1 0 1 1 1 1 1 1 0 1 1 1 1 0 0 1 1 0 1 0 1 0 0 0 1 0 1 1 1 0 1 0 1 1 1 1 0 1 1 0 1 1 1 1 1 0 1 0 0 1 0 0 0 1 1 1 0 0 0 1 0 0 0 0 0 0 0
0 0 1 1 0 0 1 0 1 1 1 1 1 1 0 1 1 1 1 0 0 1 1 0 0 0 1 0 0 0 1 0 1 1 1 0 1 0 1 1 1 1 0 1 1 0 1 1 1 1 1 0 1 0 0 1 0 0 0 1 1 1 0 0 0 1 0 0 0 0 0 0
1 0 0 1 1 0 0 1 0 1 1 1 1 1 1 0 1 1 1 1 0 0 1 1 0 0 0 1 0 0 0 1 0 1 1 1 0 1 0 1 1 1 1 0 1 1 0 1 0 1 1 1 0 1 0 0 1 0 0 0 1 1 1 0 0 0 1 0 0 0 0 0
1 1 0 0 1 1 0 0 1 0 1 1 1 1 1 1 0 1 1 1 1 0 0 1 0 0 0 0 1 0 0 0 1 0 1 1 1 0 1 0 1 1 1 1 0 1 1 0 0 0 1 1 1 0 1 0 0 1 0 0 0 1 1 1 0 0 0 1 0 0 0 0
0 0 0 0 0 0 1 1 1 0 1 0 0 1 0 0 0 1 1 1 0 0 0 1 1 1 0 0 0 0 0 1 1 0 0 0 1 0 1 0 1 1 0 0 1 1 0 0 1 1 0 0 1 1 1 1 0 0 0 1 1 0 1 1 0 0 0 0 1 0 0 0
1 1 1 0 0 1 0 0 0 0 1 0 1 0 0 1 1 1 1 1 0 1 0 1 1 0 1 0 0 1 0 1 0 0 0 1 0 0 1 0 1 1 0 1 0 0 0 1 0 0 1 1 0 1 0 1 1 0 1 1 0 1 0 1 0 0 0 0 0 1 0 0
1 0 0 1 0 1 1 1 1 1 1 0 1 1 1 1 0 0 1 1 0 1 1 1 0 0 0 1 0 1 1 1 0 1 0 1 1 1 1 0 1 1 0 1 1 1 1 1 0 1 0 0 1 0 0 0 1 1 1 0 0 0 1 0 0 0 0 0 0 0 1 0
1 1 0 0 1 0 1 1 1 1 1 1 0 1 1 1 1 0 0 1 1 0 1 1 1 0 0 0 1 0 1 1 1 0 1 0 1 1 1 1 0 1 1 0 1 1 1 1 1 0 1 0 0 1 0 0 0 1 1 1 0 0 0 1 0 0 0 0 0 0 0 1
```

Evaluation - Reliability

- Reliability against **single 8LC errors.**

Error scenario		SEC-DED	SEC-DAEC	SEC-DAEC-TAEC	IP-DAEC	**SELCC**
SE	CE	100.0%	100.0%	100.0%	68.0%	**100.0%**
	DUE	0.0%	0.0%	0.0%	32.0%	**0.0%**
	SDC	0.0%	0.0%	0.0%	0.0%	**0.0%**
DE	CE	0.0%	66.7%	66.6%	100.0%	**100.0%**
	DUE	100.0%	16.6%	4.2%	0.0%	**0.0%**
	SDC	0.0%	16.7%	29.2%	0.0%	**0.0%**
TE	CE	0.0%	0.0%	100.0%	100.0%	**100.0%**
	DUE	100.0%	33.4%	0.0%	0.0%	**0.0%**
	SDC	0.0%	66.6%	0.0%	0.0%	**0.0%**

Evaluation - Reliability

- Reliabi

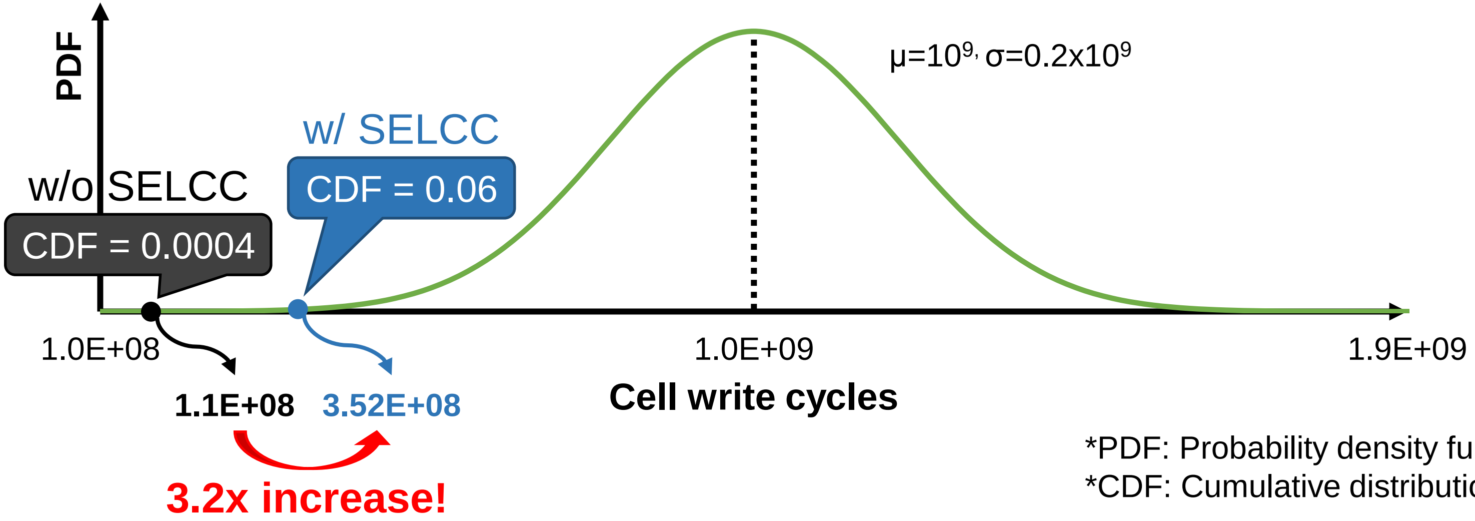
Error scenario		SEC-DED	SEC-DAEC	SEC-DAEC-TAEC	IP-DAEC	**SELCC**
SE+SE	CE	0.0%	0.9%	0.9%	0.0	**0.0**
	DUE	100.0%	44.9%	45.2%	18.7	**34.5**
	SDC	0.0%	54.2%	83.9%	81.3	**65.5**
SE+DE	CE	0.0%	0.0%	0.9%	0.0	**0.0**
	DUE	49.5%	45.6%	15.8%	28.1	**34.5**
	SDC	50.5%	54.4%	83.3%	71.9	**65.5**
SE+TE	CE	0.0%	0.0%	0.0%	0.0	**0.0**
	DUE	100.0%	44.4%	16.3%	28.3	**33.1**
	SDC	0.0%	55.6%	83.7%	71.7	**66.9**
DE+DE	CE	0.0%	0.0%	0.0%	0.0	**0.0**
	DUE	99.0%	46.4%	16.4%	41.6	**33.6**
	SDC	1.0%	53.6%	83.6%	58.4	**66.4**
DE+TE	CE	0.0%	0.0%	0.0%	0.0	**0.0**
	DUE	28.0%	43.8%	17.6%	41.8	**37.3**
	SDC	72.0%	56.2%	82.4%	58.2	**62.7**
TE+TE	CE	0.0%	0.0%	0.0%	0.0	**0.0**
	DUE	87.3%	44.5%	18.5%	42.7	**38.4**
	SDC	12.7%	55.5%	81.5%	57.3	**61.6**

Applications for SELCC

- on-die ecc? system-level ecc?

Endurance Calculation

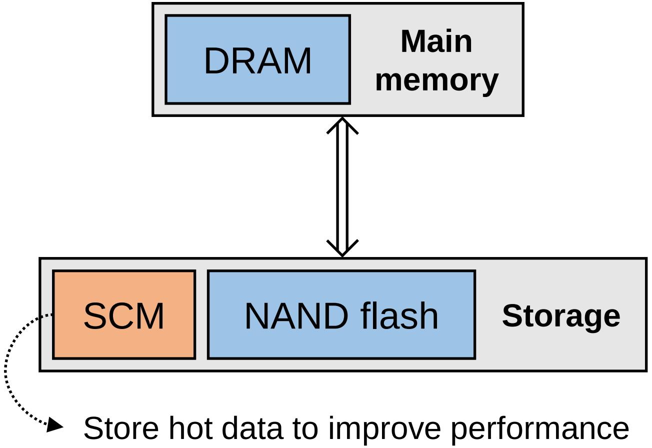
- Find maximum write cycles where **99.99%** of codewords have no defective cells.
  - w/o SELCC, all 24 cells in a codeword must be correct : (**99.9996%**)^{24} > 99.99%
  - w/ SELCC, 23 out of 24 cells in a codeword must be correct : ${}_{24}C_{23}$(**99.94%**)^{23}(0.06%)^{1} + ${}_{24}C_{24}$(**99.94%**)^{24}(0.06%)^{0} > 99.99%



*PDF: Probability density function
*CDF: Cumulative distribution function

Applications for Storage Class Memory

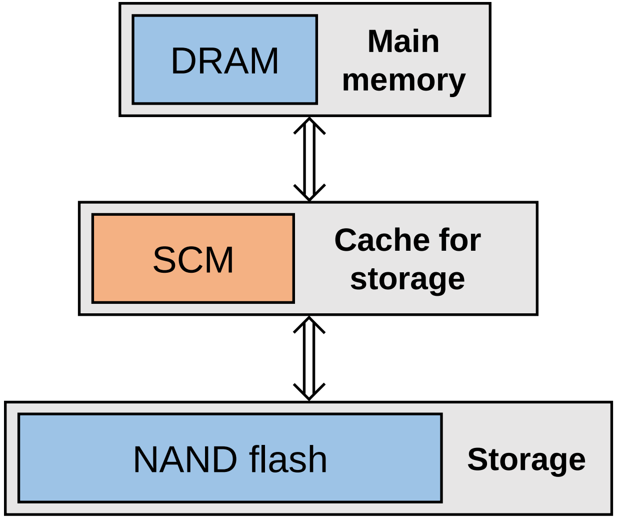
1. Tiering at the Storage-side
   - Place the hottest data to the SCM, and relegate the rest to SSDs.



❖ ***Dell Technologies. 2020. "Three Use Cases for Storage Class Memory (SCM)."***

Applications for Storage Class Memory

2. Caching at the Storage-side
   - SCM operates as a high-capacity cache between DRAM and the NAND flash drives.

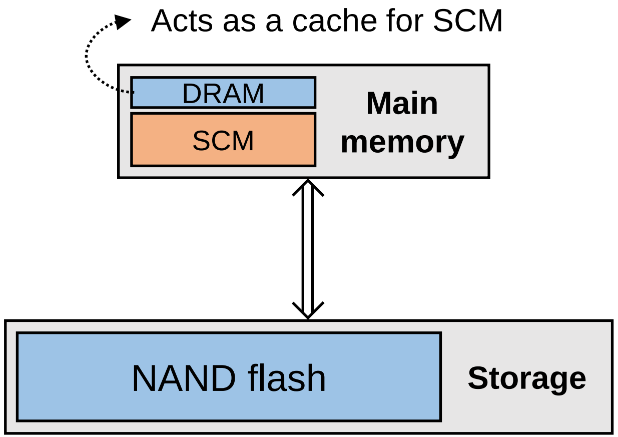


❖ ***Dell Technologies. 2020. "Three Use Cases for Storage Class Memory (SCM)."***

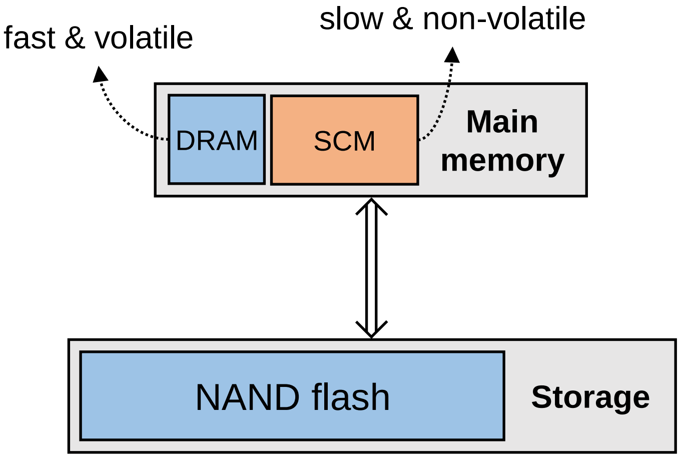
Applications for Storage Class Memory

3. Persistent Memory at the Server-side

① DRAM serves as an L4 cache for the SCM, extending the capacity of the existing DRAM.



② SCM is used as a second tier of memory next to DRAM.



❖ ***Dell Technologies. 2020. "Three Use Cases for Storage Class Memory (SCM)."***