SC23 Denver, CO | i am hpc.

SUNG KYUN KWAN UNIVERSITY(SKKU) | SAL SCALABLE ARCHITECTURE LAB

Unity ECC

Unified Memory Protection Against Bit and Chip Errors

Dongwhee Kim†, Jaeyoon Lee†, Wonyeong Jung†,
Michael B. Sullivan††, Jungrae Kim†

†**Sungkyunkwan University**, ††NVIDIA Corporation

2023.11.15

SUNG KYUN KWAN UNIVERSITY(SKKU)

NVIDIA

Executive Summary

☑ Contributions

- Performance by 8.2% (Multi-core)
- DRAM energy consumption by 8.0%
- DRAM chip size by 6.9%

☑ Maintaining the same level of reliability

☑ Key idea

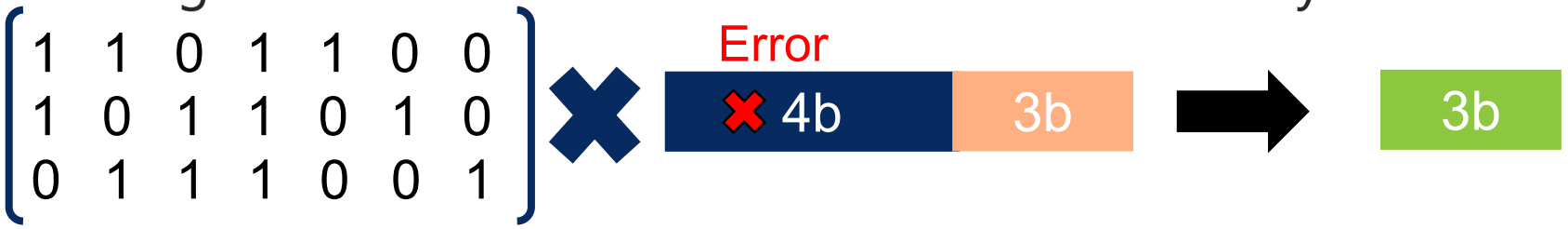
- Unity ECC can correct single-chip errors and double-bit errors

Contents

Error Correction Codes (ECC)

☑ ECC can detect and correct errors

- ECC generates codeword with additional redundancy



Syndrome	Error
(001)	(0000001)
(010)	(0000010)
(100)	(0000100)
(101)	(0100000)
(110)	(1000000)
(011)	(0010000)
(111)	(0001000)

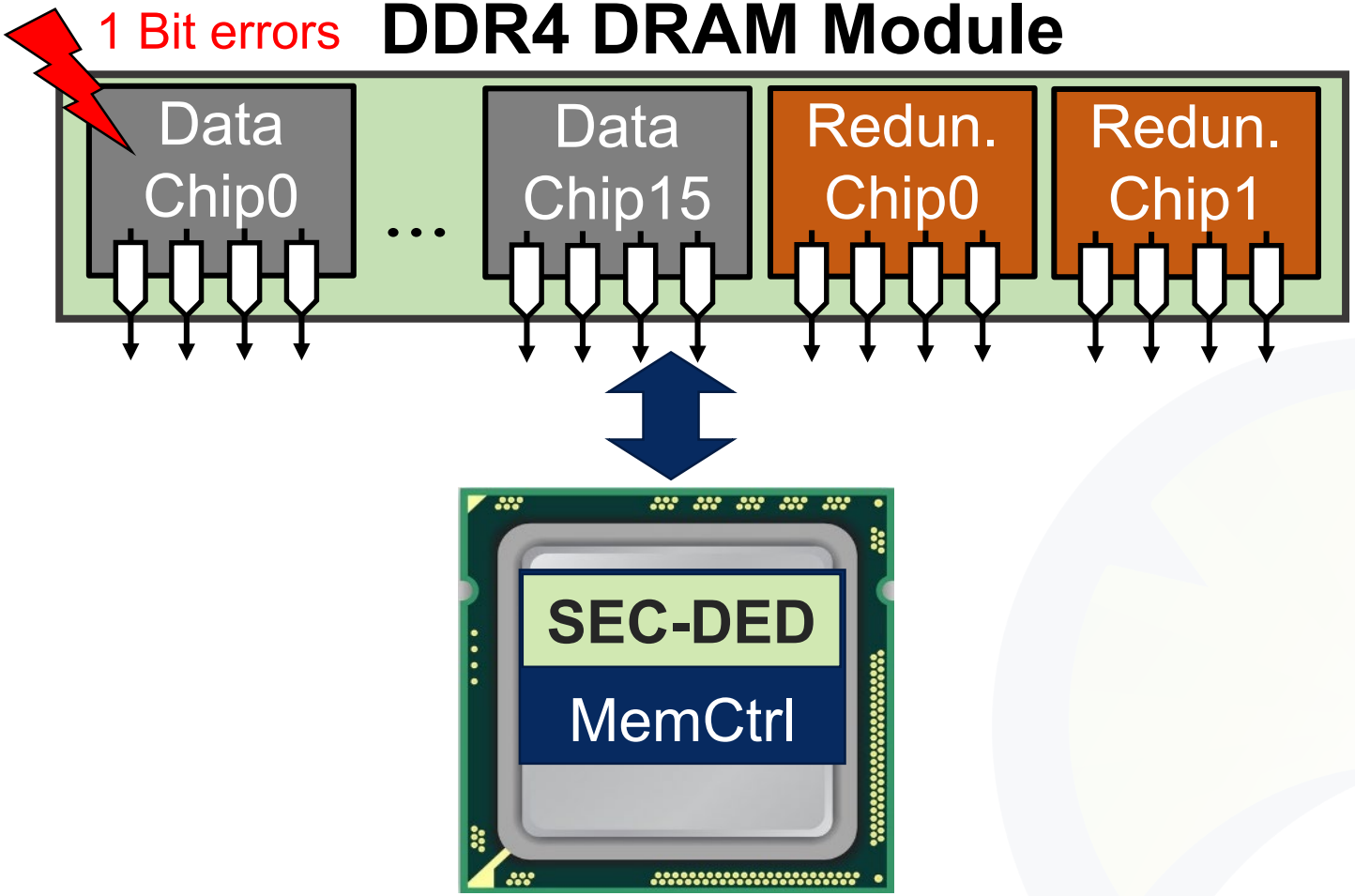
Codeword
(Error Corrected)

Data

Single Error Correction-Double Error Detection

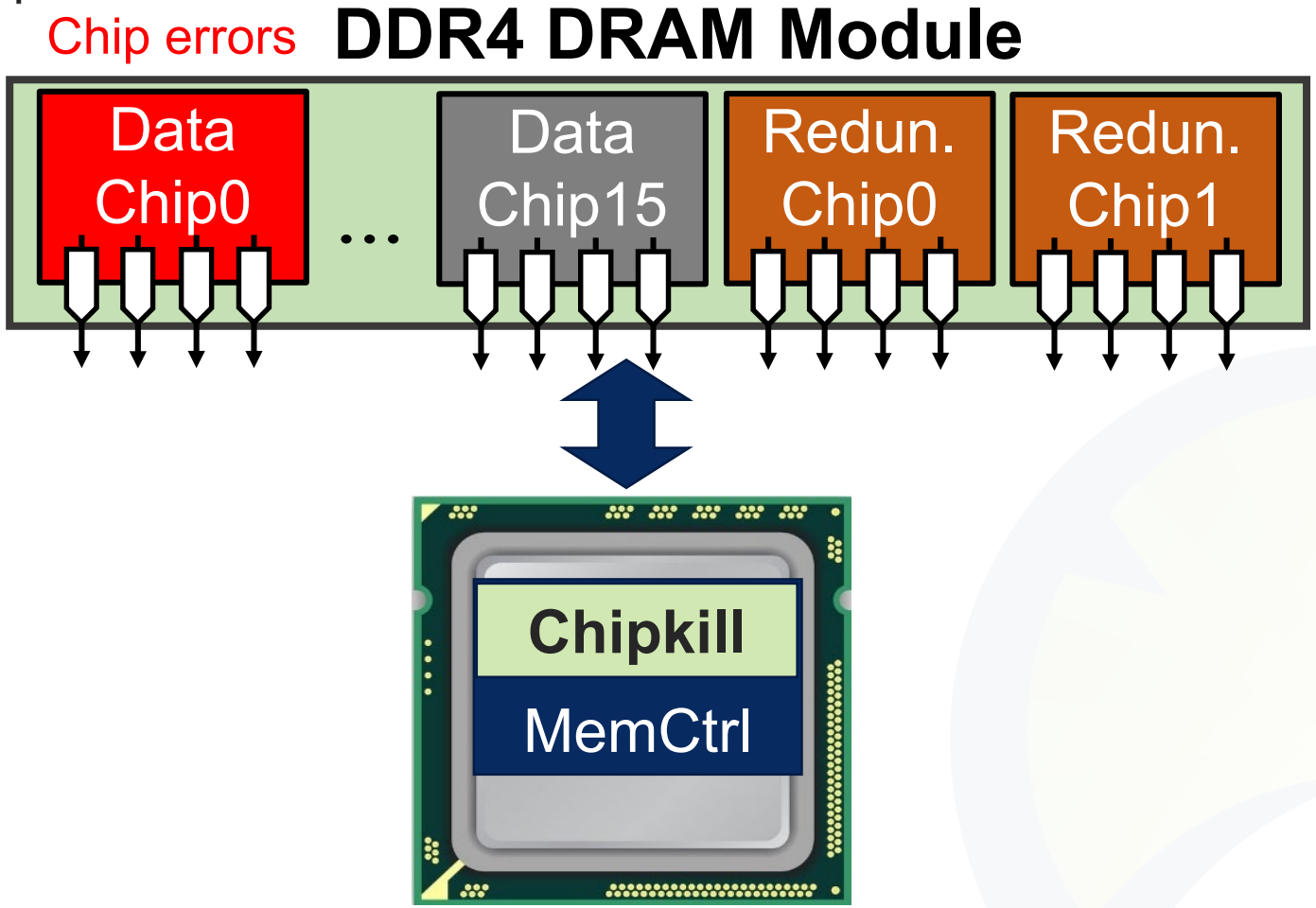
☑ SEC-DED can correct single bit and detect double bit errors

- Simple design, yet not very reliable



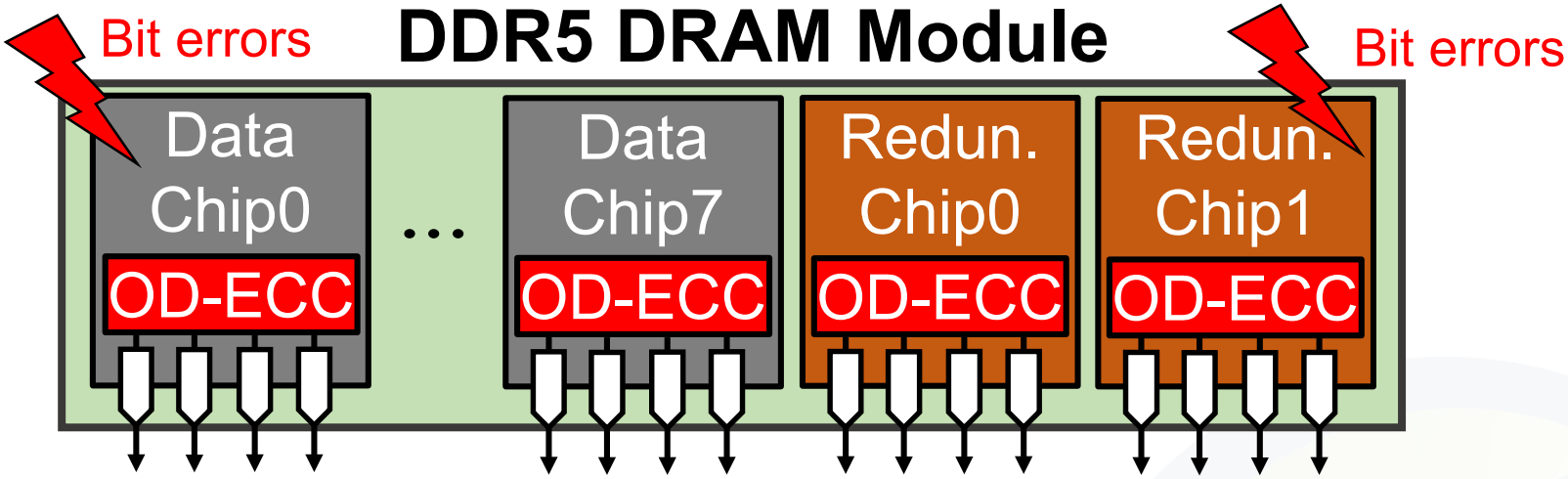
☑ To correct more severe chip errors

- Susceptible to random bit errors



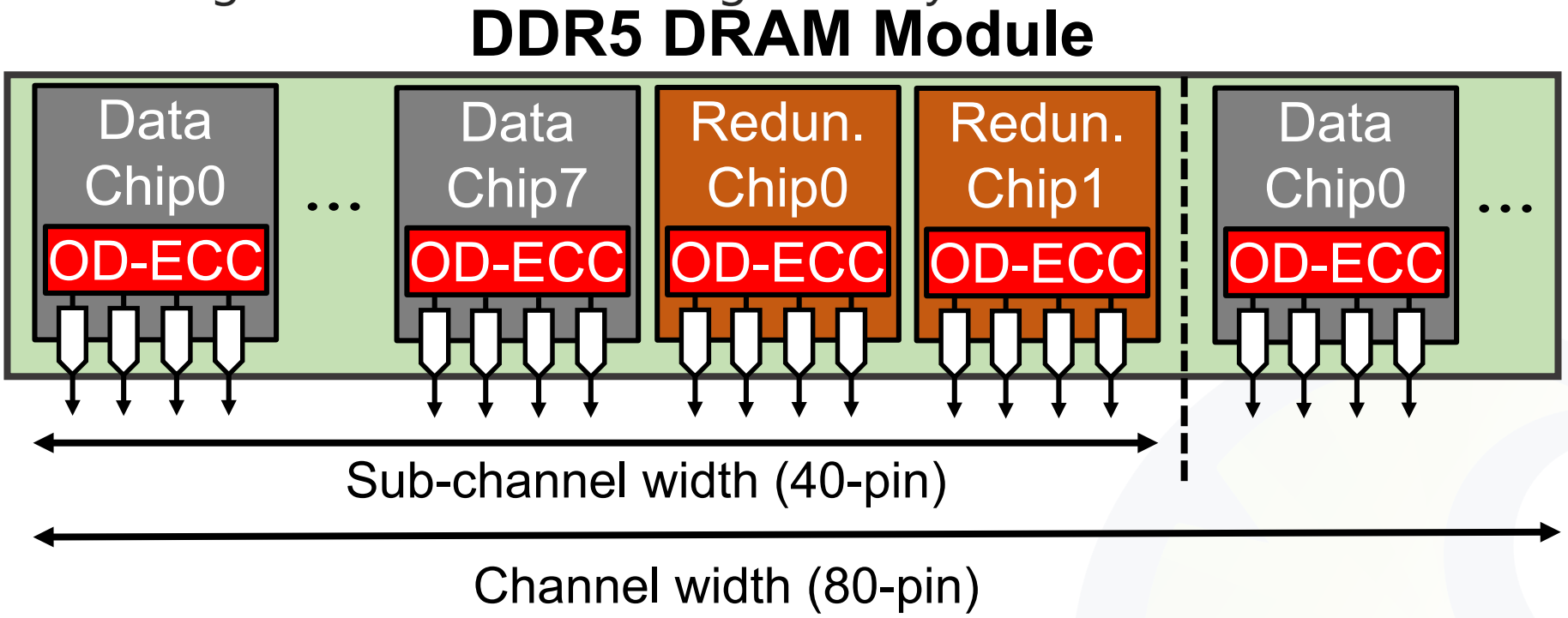
☑ On-Die ECC (OD-ECC)

- OD-ECC can correct random bit errors



☑ Sub-channel

- To align with the 64B cache granularity



Contents

Cost of Two-level Protection

☑ High redundancy ratio (32.8%)

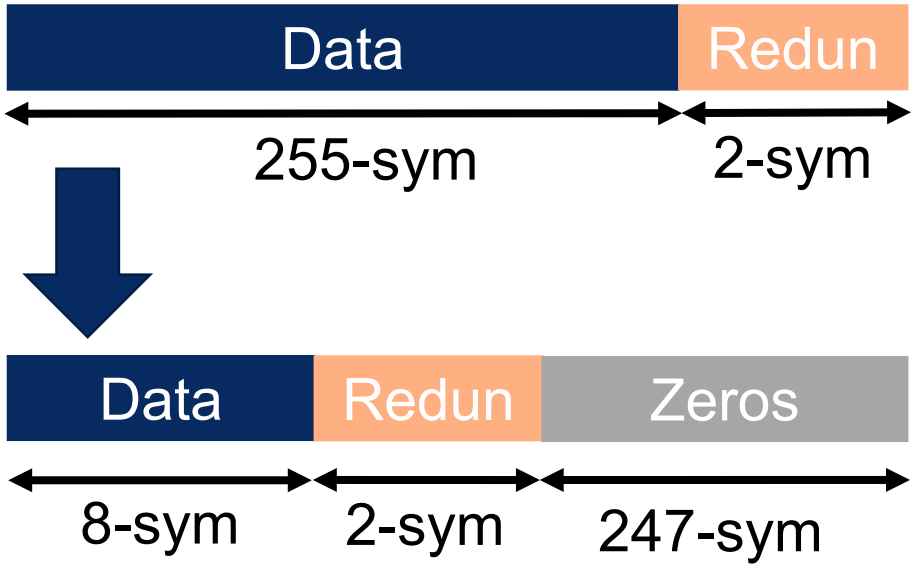
- OD-ECC (6.25%) X RL-ECC (25%)
  - RL-ECC (Rank-Level ECC): SEC-DED, Chipkill ...

☑ OD-ECC Overheads

- Performance
- DRAM energy
- DRAM chip area

☑ RL-ECC utilizes shortened code

- 8 bits per symbol (Reed-Solomon code)
- Unshortened code: **257** symbols (2,056 b)
- Shortened code: **10** symbols (80 b)



$$H = \left[\begin{array}{ccc|ccc|cc} \alpha^0 & \alpha^1 & \alpha^2 & \cdots & \alpha^{254} & \alpha^0 & 0 \\ \alpha^0 & \alpha^2 & \alpha^4 & \cdots & \alpha^{253} & 0 & \alpha^0 \end{array}\right]$$

8 columns 2 columns

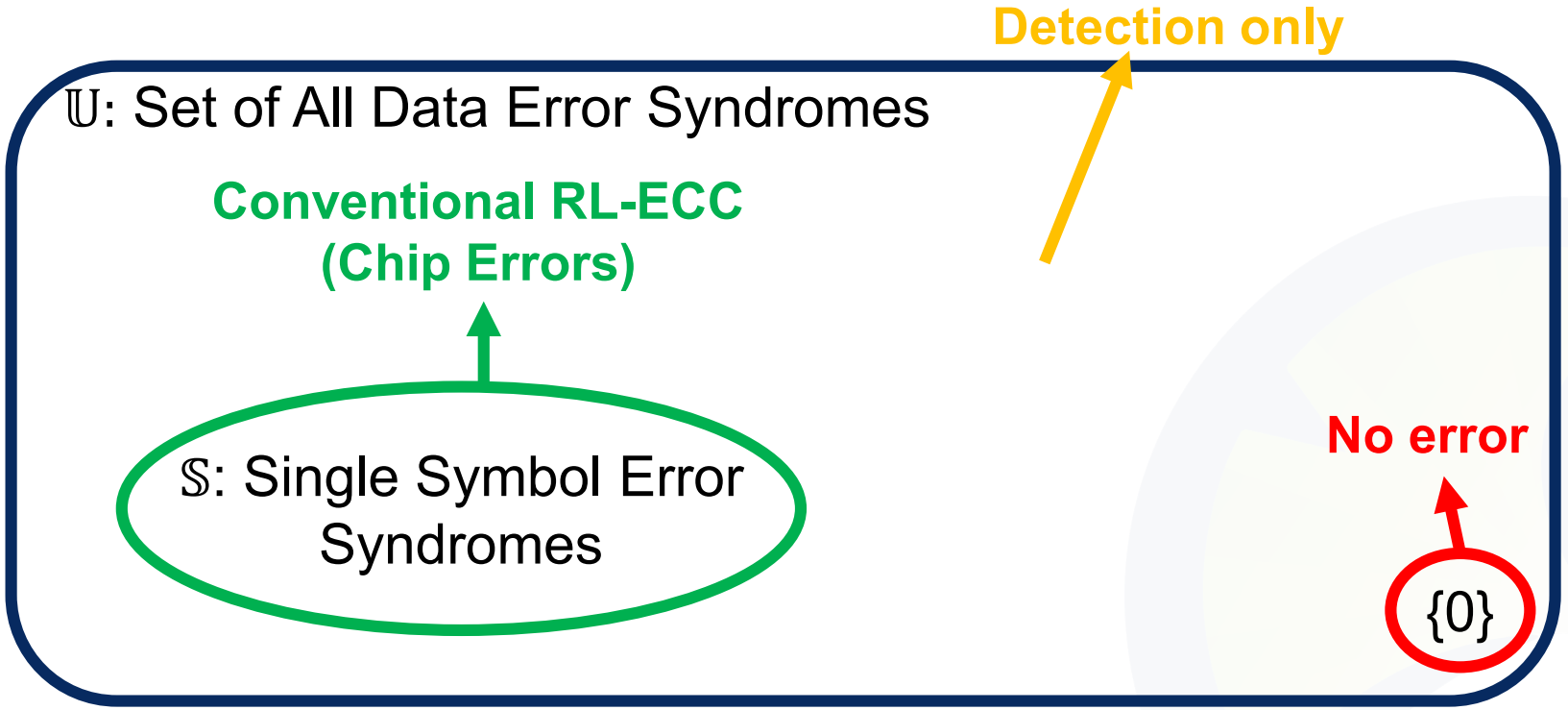
H-matrix
257 columns

$$H = \left[\begin{array}{ccccccc} \alpha^0 & \alpha^1 & \alpha^2 & \cdots & \alpha^7 & \alpha^0 & 0 \\ \alpha^0 & \alpha^2 & \alpha^4 & \cdots & \alpha^{14} & 0 & \alpha^0 \end{array}\right]$$

H-matrix
10 columns

☑ Syndrome spaces of DDR5 RL-ECC

- **Observation 1) 96.11% of syndromes are used only for detection**
- **Observation 2) Unused syndromes can be employed for OD-ECC**



Contents

☑ **Unity ECC is a single-level RL-ECC (also Chipkill)**

☑ **Unity ECC can correct single chip errors and double bit errors**

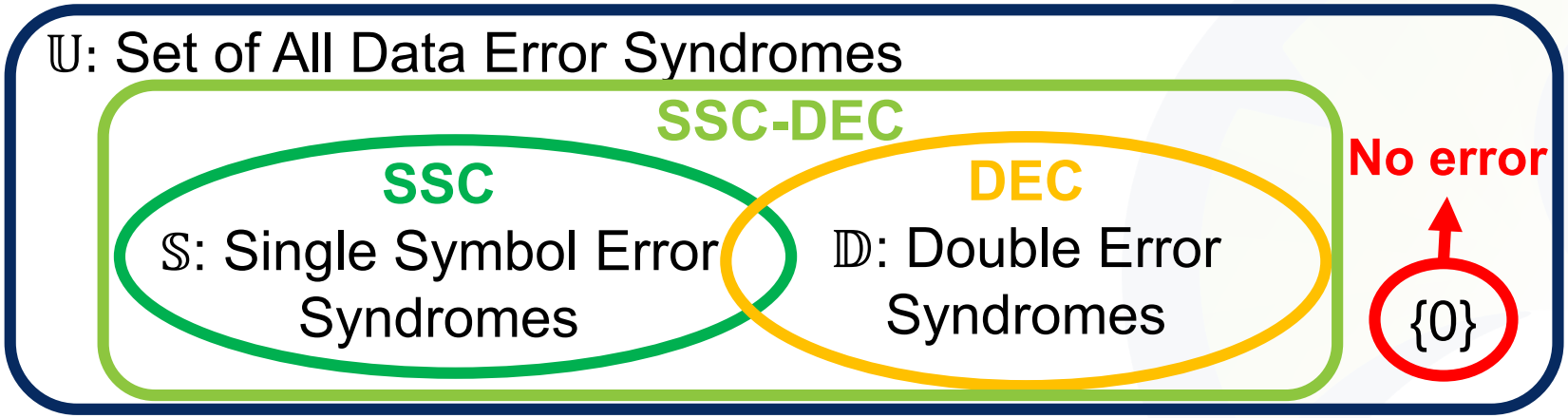
- SSC-DEC (Single Symbol Correction-Double Error Correction)

☑ **We propose flexible code construction algorithm**

Key idea: **Eliminate OD-ECC** by **remapping unused syndromes** in RL-ECC

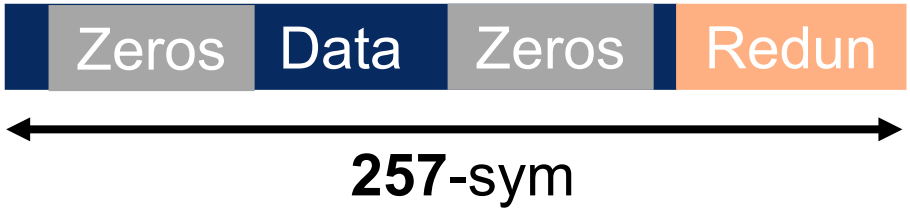
☑ H-matrix properties

- 1) All columns are **non-zero**
- 2) **DEC**: The sums (XOR operation) of any two columns are unique non-zero values
- 3) **SSC**: The sums (XOR operation) of all symbol-aligned columns are unique non-zero values
- 4) **SSC-DEC**: All sums from properties 2 and 3 should be unique



☑ Based on Reed-Solomon (RS) code

- Systematic code
- Greedy search



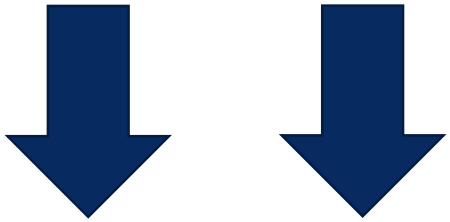
257 columns

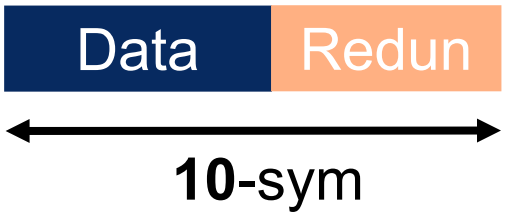
$$H = \begin{bmatrix} \alpha^0 & \alpha^1 & \alpha^2 & \cdots & \alpha^{254} & \alpha^0 & 0 \\ \alpha^0 & \alpha^2 & \alpha^4 & \cdots & \alpha^{253} & 0 & \alpha^0 \end{bmatrix}$$

Unshortened RS code H-matrix

✓ **Select 10 columns**

✓ Meet the H-matrix properties





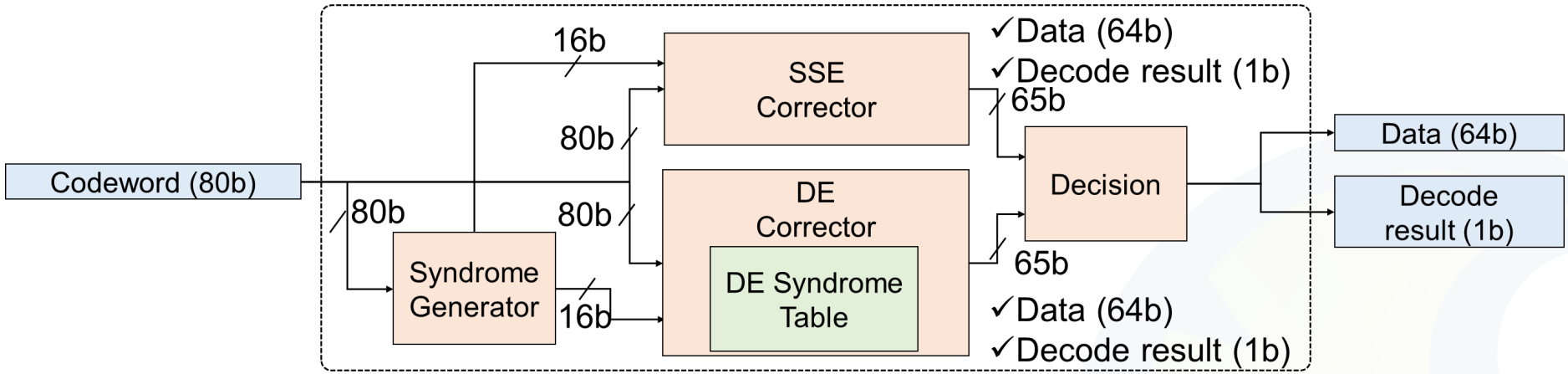
10 columns

$$H = \begin{bmatrix} \alpha^{25} & \alpha^{39} & \alpha^{63} & \alpha^{108} & \alpha^{141} & \alpha^{184} & \alpha^{215} & \alpha^{230} & \alpha^0 & 0 \\ \alpha^{50} & \alpha^{78} & \alpha^{126} & \alpha^{216} & \alpha^{27} & \alpha^{113} & \alpha^{175} & \alpha^{205} & 0 & \alpha^0 \end{bmatrix}$$

H-matrix example of (10, 8) **Unity ECC**

☑ Parallel decoder

- SSE Corrector (Single Symbol Error)
- DE Corrector (Double Error) [with DE Syndrome Table]



Contents

Four Evaluation Metrices

- ☑ **Performance**
- ☑ **DRAM energy**
- ☑ **Reliability**
- ☑ **Hardware overheads**
  - Synthesis (Synopsys Design Compiler, UMC 28nm)

Environmental Setup

☑ DRAM timing parameters

- Unity ECC eliminates OD-ECC **(lower DRAM timing parameters)**
- Baseline: OD-ECC + RL-ECC (Chipkill)
- Unity ECC: RL-ECC

Parameter	Baseline (OD-ECC + RL-ECC)	Unity ECC (RL-ECC)
Read latency (nCK)	40 (16.67ns)	**36 (15ns)**
tRCD (ns)	16.25	
tRP (ns)	16.25	
IDD4W (mA)	345	**240**

DRAM key timing parameters

Environmental Setup

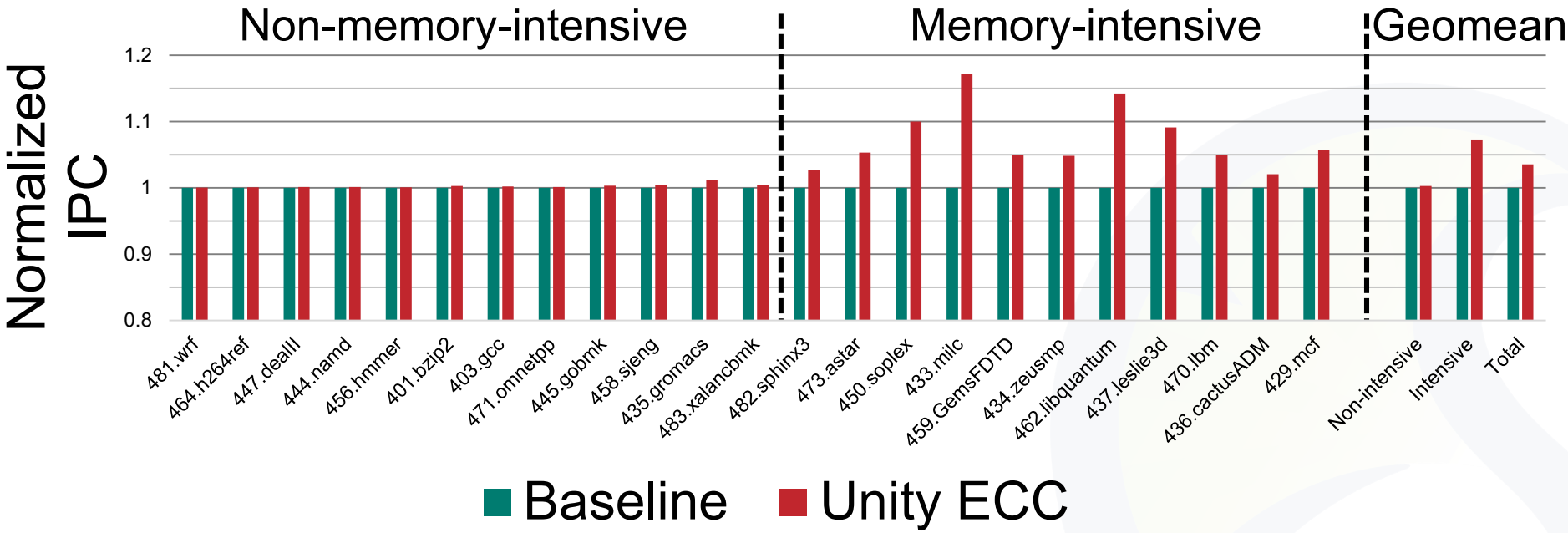
☑ Simulation configuration

- Using Ramulator **[1]** and DRAMPower **[2]**
- Single-core/4-cores
- DRAM: DDR5-4800B, 16Gb, x4 chip, 32 banks
- Benchmarks: SPEC CPU2006 **[4]**

Performance Results

Single-core performance improvement

- **17.2%** (max.)
- **7.27%** (memory-intensive geomean)
- 3.56% (total geomean)

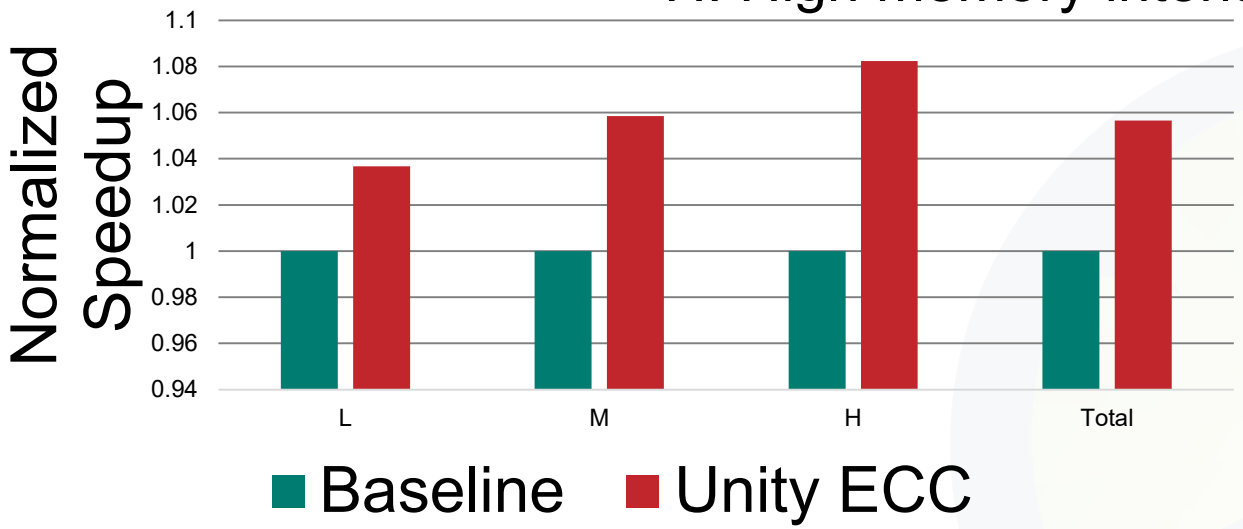


Performance Results

☑ Multi-core performance improvement

- **8.2%** (max.)
- 5.7% (total geomean)

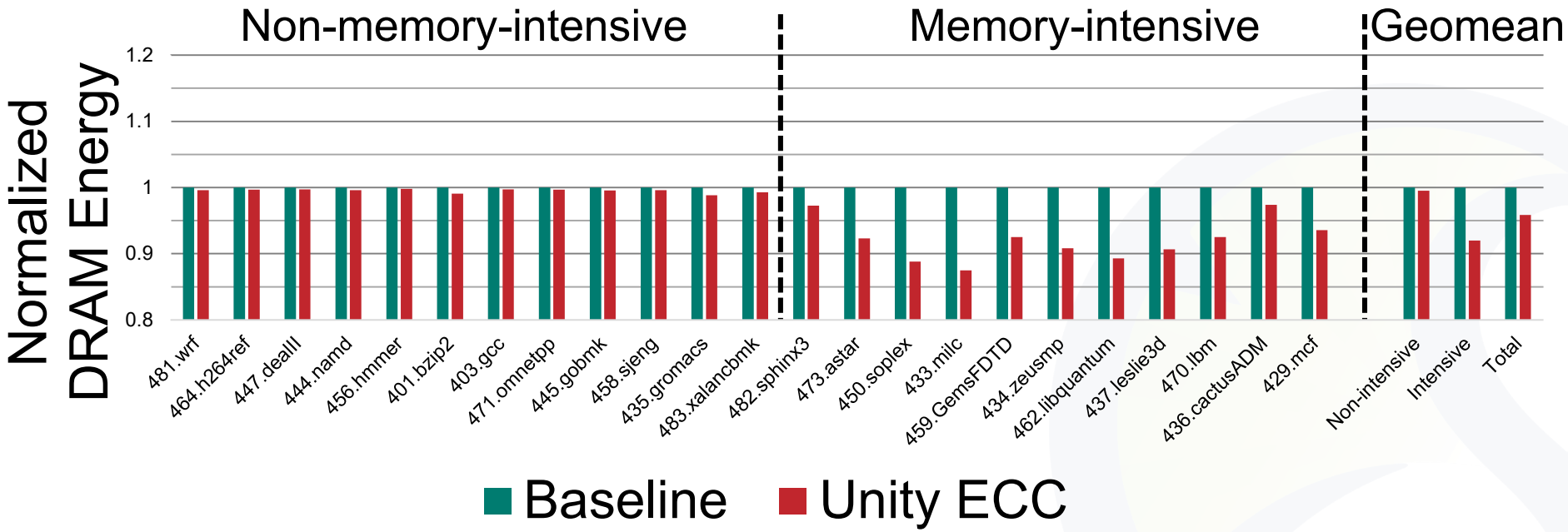
✓ L: Low memory intensity mix
✓ M: Medium memory intensity mix
✓ H: High memory intensity mix



DRAM Energy Results

☑ Single-core DRAM energy reduction

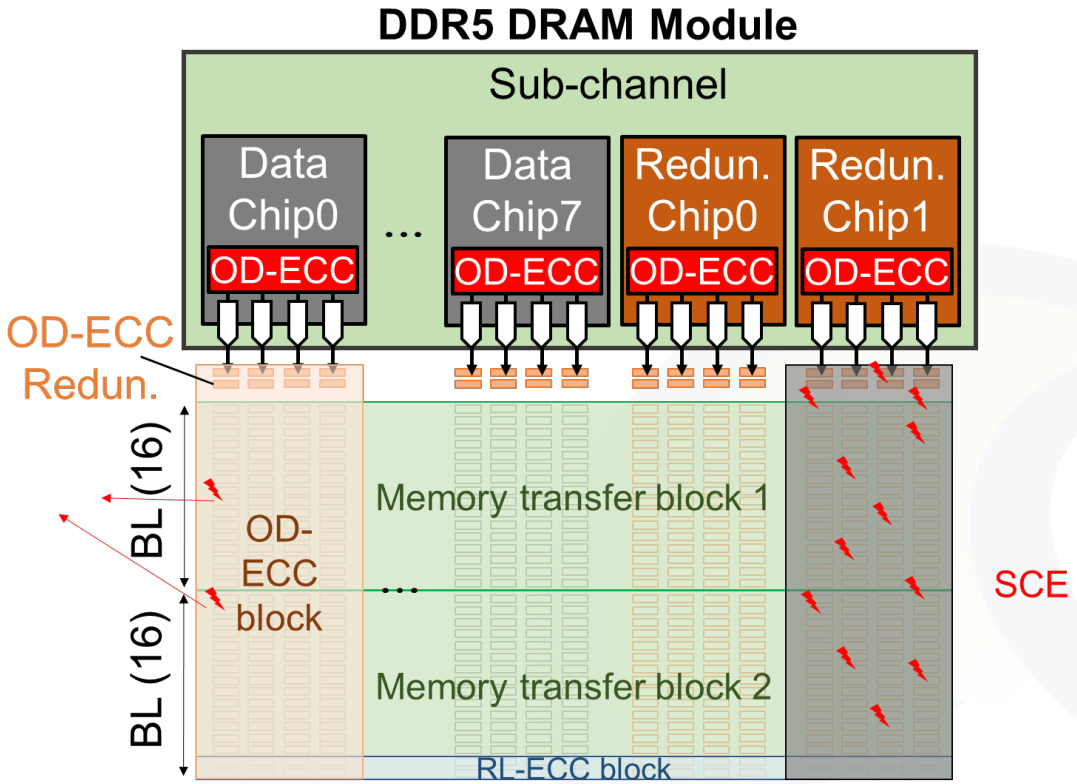
- **12.53%** (max.)
- **8.00%** (memory-intensive geomean)
- 4.16% (total geomean)



Reliability Results

☑ Methodology

- Random injection of bit and chip errors
- **The worse** of the two memory transfer blocks (128B) becomes the final output

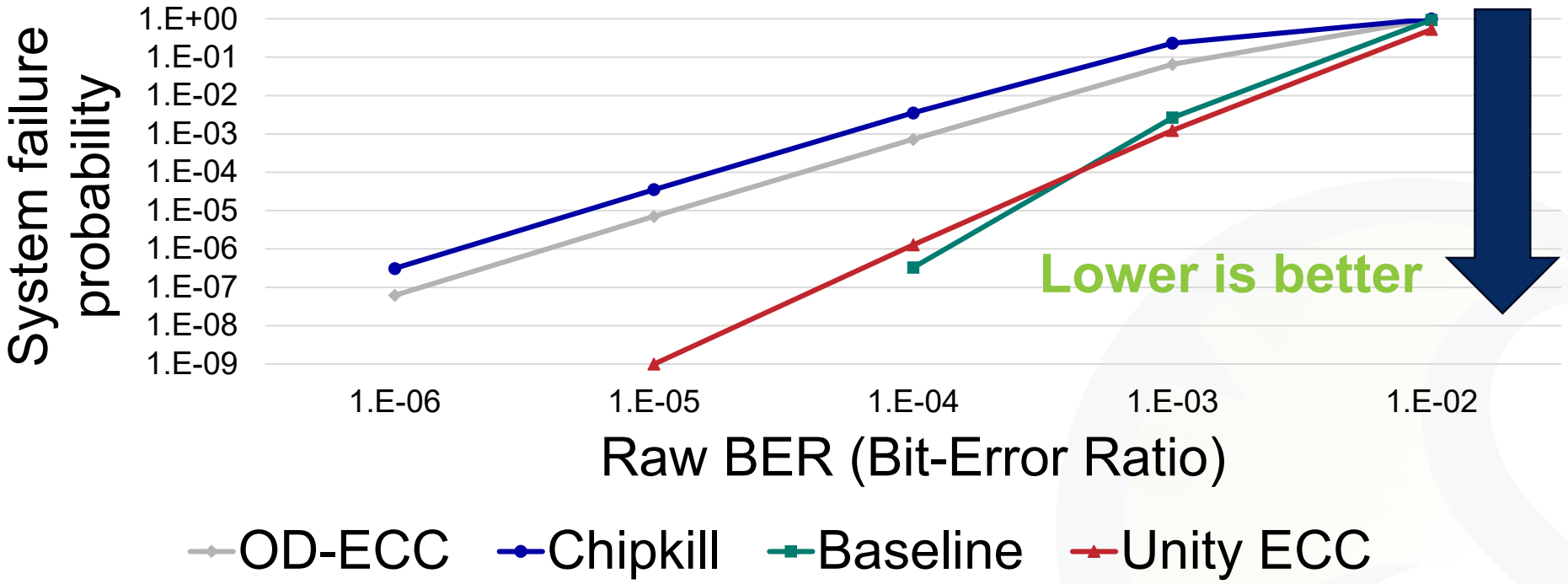


✓ DBE: Double-Bit Error
✓ SCE: Single-Chip Error

Reliability Results

☑ Bit errors

- Unity ECC ≈ Baseline >> OD-ECC > Chipkill
  - Unity ECC even surpasses Baseline when BER > 10^{-3}



☑ Bit-error scenarios

- Unity ECC is robust to bit errors
  - Higher correction-probability of multi-bit errors
  - **Higher Correctable Error (CE) is better**

A comparison of reliability against bit–error scenarios **(CE cases)**

Error Scenario	OD-ECC	Chipkill	Baseline	Unity ECC
SBE (%)	100			
DBE (%)	0.14	100	100	**100**
DBE + DBE (%)	0.01	12.30	8.89	**98.68**

✓ SBE: Single-Bit Error
✓ DBE: Double-Bit Error

Reliability Results

☑ Bit-error scenarios

- Unity ECC is robust to bit errors
  - Higher detection-probability of multi-bit errors
  - **Lower Silent Data Corruption (SDC) is better**

A comparison of reliability against bit–error scenarios **(SDC cases)**

Error Scenario	OD-ECC	Chipkill	Baseline	Unity ECC
DBE (%)	99.86	0	0	**0**
DBE + DBE (%)	99.99	1.03	1.75	**0.09**

✓ SBE: Single-Bit Error
✓ DBE: Double-Bit Error

Reliability Results

☑ Chip errors

- Unity ECC ≈ Baseline ≈ Chipkill >> OD-ECC
  - Unity ECC can correct all 1 chip errors
  - Unity ECC can detect most 2 chip errors (99.9999996%)

A comparison of reliability against chip–error scenarios

	Error Scenario	OD-ECC	Chipkill	Baseline	Unity ECC
CE cases	SCE (%)	0	100	100	**100**
	SCE + SCE (%)	0	0	0	0
SDC cases	SCE (%)	100	0	0	0
	SCE + SCE (%)	100	0	0	**0.0000004**

✓ SCE: Single-Chip Error

Reliability Results

☑ Chip errors

- Unity ECC ≈ Baseline ≈ Chipkill >> OD-ECC
  - Unity ECC can correct all 1 chip errors
  - Unity ECC can detect most 2 chip errors (99.9999996%)

More results can be found in our paper

Scenario				
SCE (%)	0	100	100	**100**
SCE + SCE (%)	0	0	0	0

A comparison of reliability against chip–error scenarios **(SDC cases)**

SCE (%)	100	0	0	0
SCE + SCE (%)	100	0	0	**0.0000004**

✓ SCE: Single-Chip Error

Hardware Overheads Results

☑ Unity ECC vs Baseline RL-ECC (Chipkill) – Latency

- Read latency slightly increases in rare error cases only
  - Doesn't affect normal read latency

	Encoder		Decoder	
	Baseline RL-ECC	Unity ECC	Baseline RL-ECC	Unity ECC
Encoding latency (ns)	0.25	0.25		
Decoding latency (ns) - detection			0.25	**0.25**
Decoding latency (ns) - overall			0.31	**0.81**

Hardware Overheads Results

☑ Unity ECC vs Baseline RL-ECC (Chipkill) – Area and Power

- Decoder area slightly increases **0.009%** of the entire processor
- Power savings from DRAM can easily offset the 25.4mW
- Eliminating OD-ECC can reduce the DRAM chip size by **6.9% [5]**

	Encoder		Decoder	
	Baseline RL-ECC	Unity ECC	Baseline RL-ECC	Unity ECC
Area (um^2)	113.06	264.26	1184.74	**10135.78**
Total power (mW)	0.34	1.02	3.75	**29.13**

Summary

☑ Observations

- OD-ECC overheads & Shortened code in RL-ECC

☑ Idea: Unity ECC

- Eliminate OD-ECC while maintaining reliability levels by repurposing unused correction syndromes

☑ Evaluation

- **Pros:** Performance (8.2%), DRAM energy (8.0%) and DRAM chip size (6.9%)
- **Cons:** Reliability and Hardware overheads (0.009%)

Reliability Evaluation Open Source

☑ ECC-ExerSim [3]

- Github search ‘xyz123479/ECC-exercise‘
- CERN-OHL-S-2.0 license

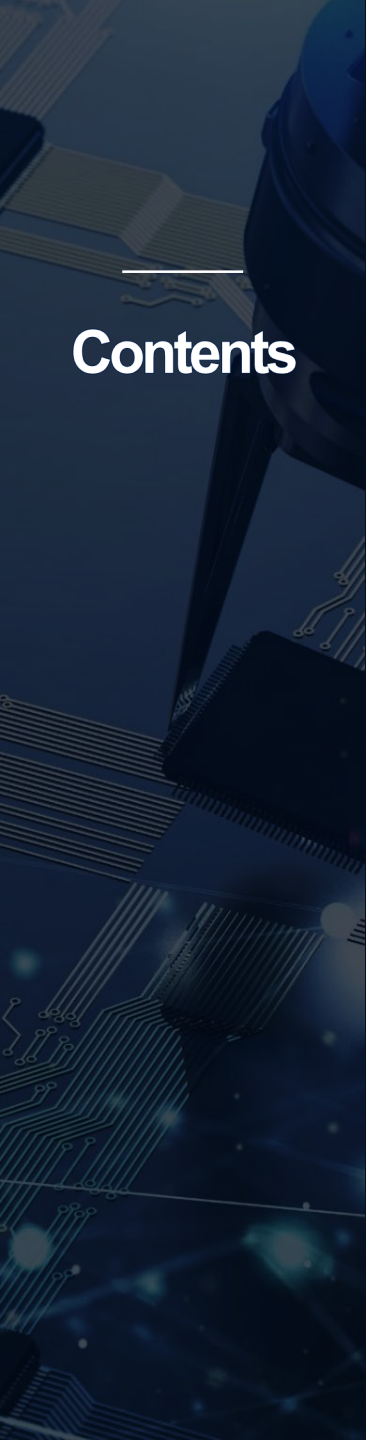
ECC-ExerSim

Thank you

Q&A

SUNG KYUN KWAN UNIVERSITY(SKKU)

SAL
SCALABLE ARCHITECTURE LAB



Contents

Related Works

☑ Comparison to Related Works

- Unity ECC has **high error correction capability**
- **No additional redundancy (vs Chipkill)**

✓ n: codeword length
✓ k: data length

Class	Error Correction			ECC word configs		
	SE	DE	SSE	n	k	Redundancy
SEC	O	X	X	136	128	6.25%
DEC	O	O	X	144	128	12.5%
	O	O	X	78	64	21.875%
SSC (Chipkill)	O	X	O	80	64	25%
DEC-SbEC	O	O	O	88	64	37.5%
Unity ECC	O	O	O	80	64	25%

✓ SEC (Single Error Correcting)
✓ DEC (Double Error Correcting)
✓ SSC (Single Symbol Correcting)
✓ DEC-SbEC (Double Bit Error Correcting – Single b-bit Byte Error Correcting)

DRAM timing parameters

☑ DRAM timing parameters

- Due to OD-ECC

Parameter	Baseline (OD-ECC + Chipkill)	Unity ECC
Read latency (nCK)	40 (16.67ns)	**36 (15ns)**
tRCD (ns)	16.25	
tRP (ns)	16.25	
tCCD_L_WR (nCK)	48	**24**
Write latency (nCK)	38	**34**
tCCD_S_WTR (nCK)	52	**48**
tCCD_L_WTR (nCK)	70	**66**
IDD0 (mA)	103	
IDD4W (mA)	345	**240**

Simulation Configuration

☑ Simulation configuration

- Single-core
- Multi-core (4 core)

Processor	1 or 4 core(s), 4GHz, 4-wide issue, 8 MSHRs per core 128-entry instruction window
LLC	64B cacheline, 8-way associative, 47 CPU-cycle latency, 8MB total capacity (2MB/core with 4 cores)
Memory Controller	FR-FCFS-Cap scheduling, timer-based row open policy, 64-entry read/write request queue
DRAM	1 channel, 1 rank, **DDR5**, 4800Mbps, 16Gb ×4 chip, 8 bank groups, 4 banks per bank group
Benchmarks	23 benchmarks from **SPEC CPU2006 [4]**

Hardware Overheads

☑ Synthesis setup

- Compiler: Synopsys Design Compiler
- Logic libraries: UMC 28nm SVT/LVT cells (Choose the worst one)
- Clock uncertainty 60% (40% margin)
- Activity factor switching: 10%
- Corner: ss
- PVT (28nm, 0.9V, 125’C)

References

[1] Yoongu Kim, Weikun Yang, and Onur Mutlu. 2015. Ramulator: A fast and extensible DRAM simulator. IEEE Computer architecture letters 15, 1 (2015), 45–49.

[2] Karthik Chandrasekar, ChristianWeis, Yonghui Li, Benny Akesson, NorbertWehn, and Kees Goossens. 2012. DRAMPower: Open-source DRAM power energy estimation tool. URL: http://www. drampower. info 22 (2012).

[3] https://github.com/xyz123479/ECC-exercise

[4] SPEC CPU2006. 2006. Standard performance evaluation corporation

References

[5] Sanguhn Cha, O Seongil, Hyunsung Shin, Sangjoon Hwang, Kwangil Park, Seong Jin Jang, Joo Sun Choi, Gyo Young Jin, Young Hoon Son, Hyunyoon Cho, et al. 2017. Defect analysis and cost-effective resilience architecture for future DRAM devices. In 2017 IEEE International Symposium on High Performance Computer Architecture (HPCA). IEEE, 61–72