Enhancing CXL Memory RAS through a Forgotten Coding Theory

MEMORY FABRIC FORUM

OCP GLOBAL SUMMIT

OCT 15-17, 2024
SAN JOSE, CA

Enhancing CXL Memory RAS through a Forgotten Coding Theory

Tong Zhang, ScaleFlux Inc.

RS Codes for DRAM Fault Tolerance

- Data centers demand strong DRAM fault tolerance
  - ✓ **Higher reliability & stability** of data centers
  - ✓ **Lower system TCO** by accommodating less reliable, lower cost DRAM chips
  - ✓ **Higher resilience** to the security risk caused by DRAM RowHammer attacks
- Reed-Solomon (RS) code: One of the most widely used error correction codes (ECC)
  - ➢ Adopted by server CPUs for DRAM fault tolerance
- Standard RS decoding algorithms correct **up to** $t = \left\lfloor \frac{d-1}{2} \right\rfloor$ symbol errors, where d is the minimum distance between any two codewords

Beyond-Minimum-Distance Decoding

Correct **more-than-*t*** errors?!

- Professor Peter Elias at MIT in 1950s
  - Rationale: Minimum-distance-based decoding ***significantly under-utilizes*** the Euclidean code space

(80, 64) RS code: protects 64-byte data with 16-byte redundancy → (Minimum-distance decoding) → Utilize only $< \frac{1}{10^{25}}$ of the code space

↓

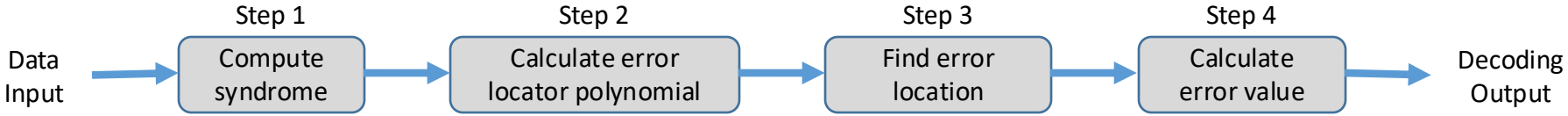
List decoding: **search** for possible decoding solutions in a larger space beyond the minimum distance

↓

How to **efficiently** search in the larger code space?

Implementation of Standard RS Decoding

Standard RS decoding algorithm **bounded** by minimum distance

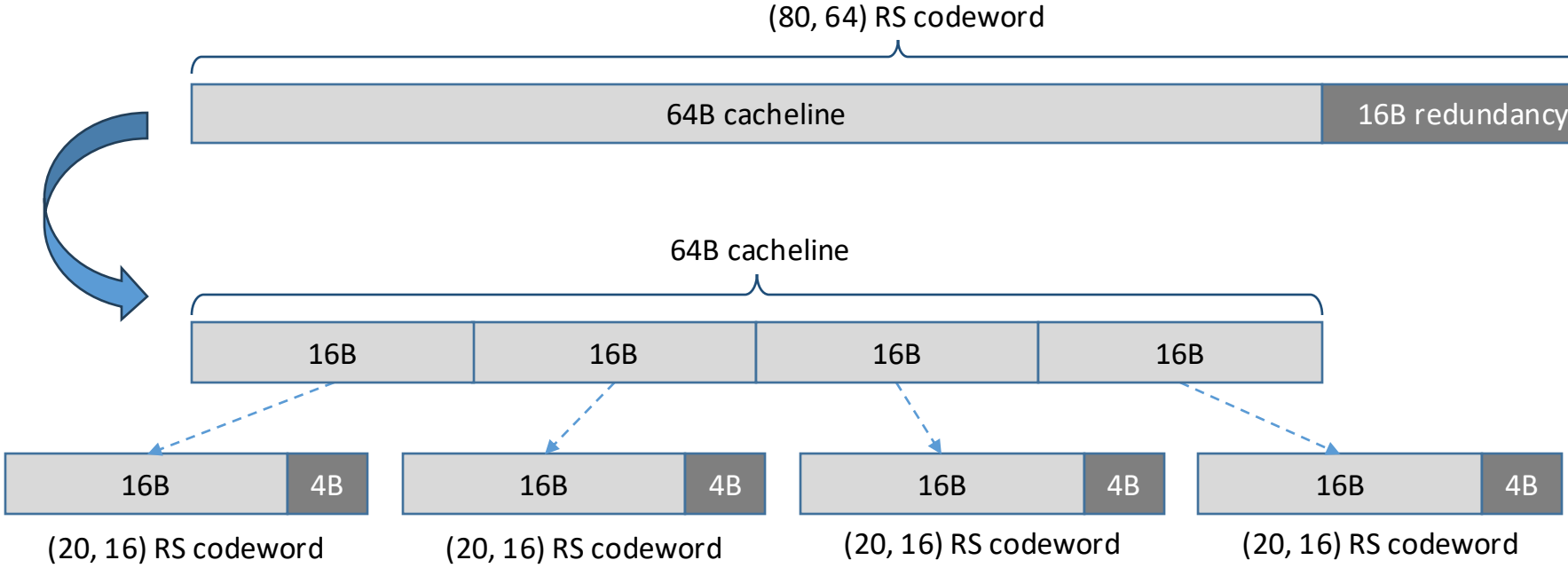


Circuit Implementation	Step 1	Step 2	Step 3	Step 4
Fully serial	$\propto n \cdot t$	$\propto t^2$	$\propto n \cdot t$	$\propto t^2$
Fully parallel	$\propto n \cdot t$	$\propto \boldsymbol{t^3}$	$\propto \boldsymbol{n \cdot t^2}$	$\propto t^2$

Computational Complexity

Implementation of ultra-low-latency standard RS decoder at reasonable silicon cost

Implementation of Standard RS Decoding: Codeword Interleaving

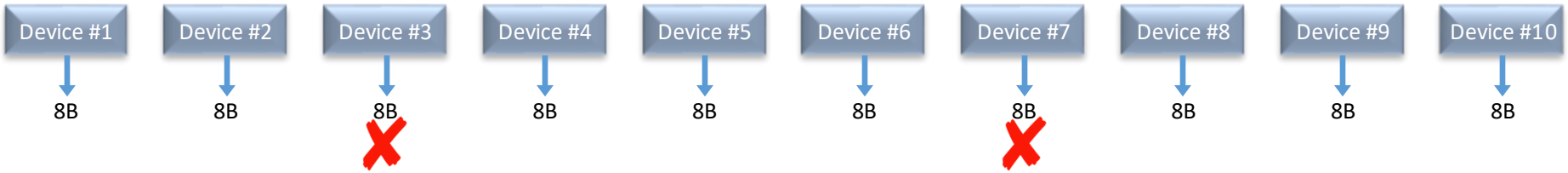


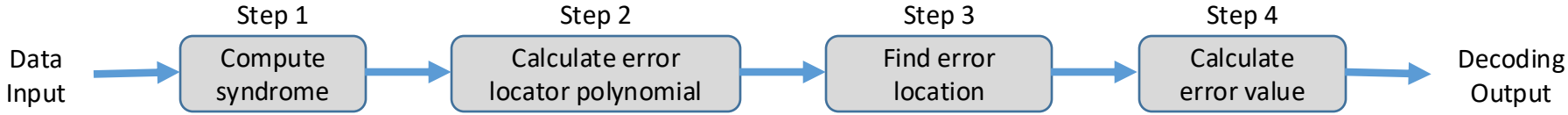
✓ **Short decoding latency at low silicon cost**

✗ **Much weaker DRAM fault tolerance strength**

ScaleFlux DRAM-Oriented RS List Decoding

One DIMM rank: 10 x4 DDR5 devices



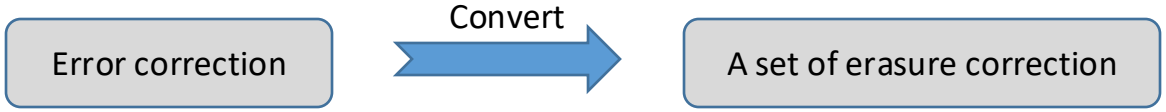


Circuit Implementation	Step 1	Step 2	Step 3	Step 4
Fully serial	$\propto n \cdot t$	$\propto t^2$	$\propto n \cdot t$	$\propto t^2$
Fully parallel	$\propto n \cdot t$	$\propto \boldsymbol{t^3}$	$\propto \boldsymbol{n \cdot t^2}$	$\propto t^2$

Computational Complexity

If error locations are known ➔ steps 2 & 3 will disappear

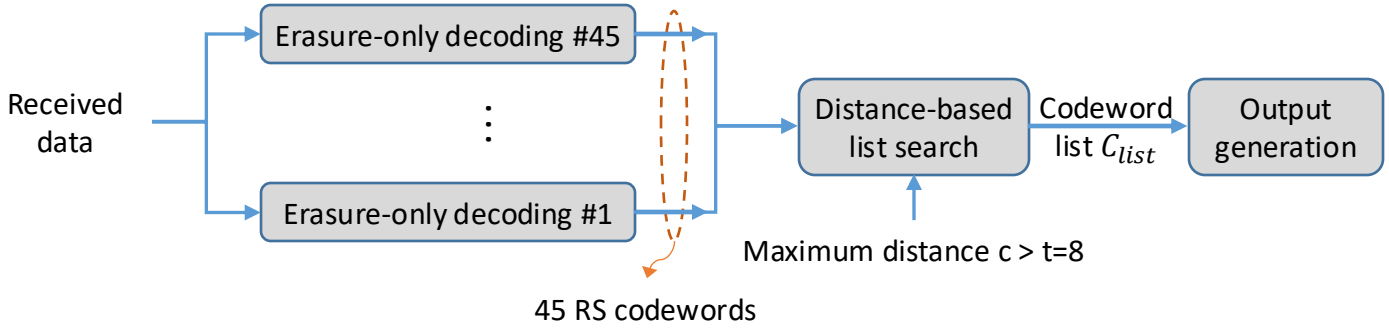
ScaleFlux DRAM-Oriented RS List Decoding



One DIMM rank: 10 x4 DDR5 devices

Device #1 | Device #2 | Device #3 | Device #4 | Device #5 | Device #6 | Device #7 | Device #8 | Device #9 | Device #10

- Define $\binom{10}{2} = 45$ erasure sets, where each erasure set covers two distinct DRAM devices
- Implement 45 fully parallel ultra-low-latency erasure decoders

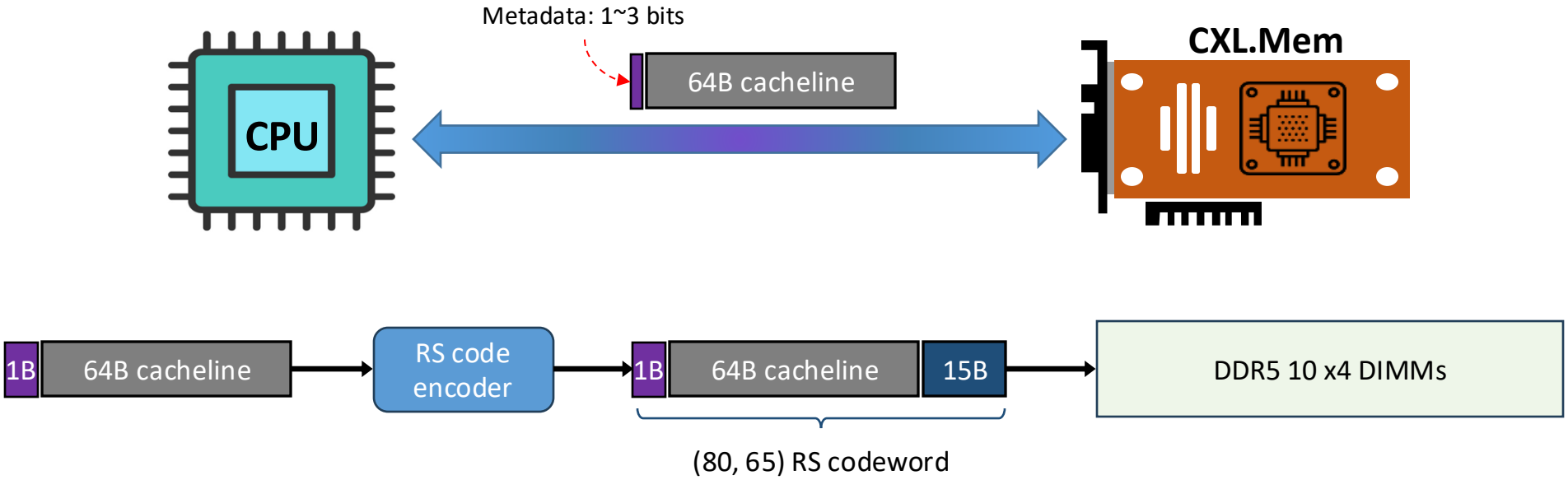


High computational complexity

Arithmetic optimization

Low-cost silicon implementation

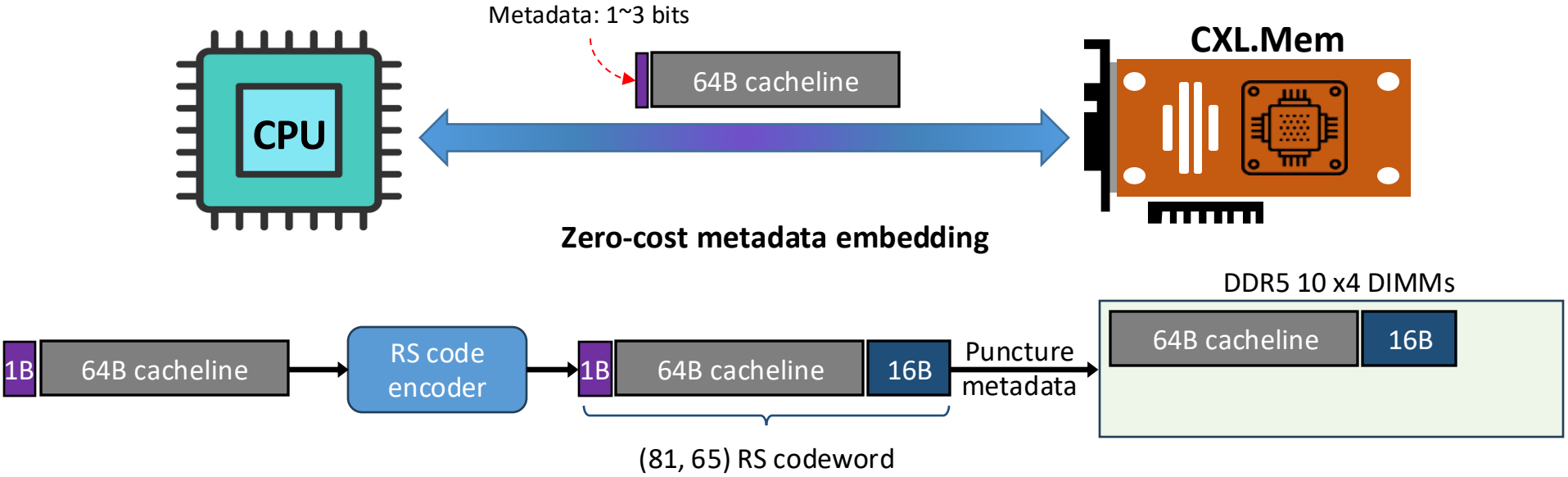
Application to CXL Memory



✗ **Waste 5~7 bits per cacheline storage**

✗ **Weaker error correction strength**

Application to CXL Memory



Metadata-aware list decoding: For each possible metadata content, do erasure-based list decoding!

Decoding algorithm formulation + Arithmetic optimization → Area-efficient parallel decoder VLSI architecture

Evaluation and Implementation

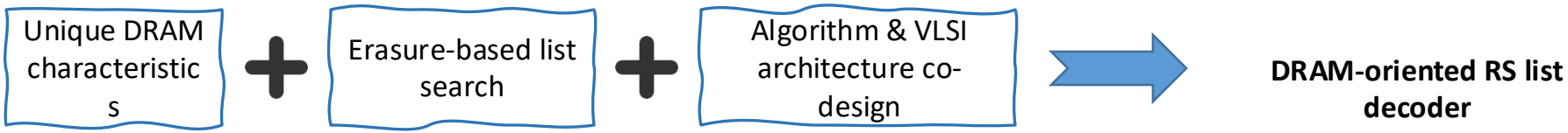
- Three possible outcome of decoding:
  1. **Success**: indeed outputs the *correct* codeword (**good**)
  2. **Detected failure**: fails to find a valid codeword and declares a decoding failure (**bad**)
  3. **Mis-correction**: outputs an incorrect but valid codeword (**really bad**)
- FPGA-based platform to confirm the superior error correction power
  - Given e symbol errors, define $P_{suc}(e)$, $P_{df}(e)$, $P_{mc}(e)$ as the probability of success, detected failure, mis-correction

(80, 64) RS	$e \in [0,8]$	$e = 9$	$e = 10$	$e = 11$	$e = 12$
SFX List decoding	$P_{suc}(e) = 1$ $P_{df}(e) = 0$ $P_{mc}(e) = 0$	$\boldsymbol{P_{suc}(e) \approx 1}$ $P_{df}(e) \approx 0$ $P_{mc}(e) = 0$	$\boldsymbol{P_{suc}(e) \approx 1}$ $P_{df}(e) \approx 0$ $P_{mc}(e) = 0$	$\boldsymbol{P_{suc}(e) \approx 1}$ $P_{df}(e) \approx 2 \times 10^{-11}$ $P_{mc}(e) \approx 0$	$\boldsymbol{P_{suc}(e) \approx 1}$ $P_{df}(e) \approx 6 \times 10^{-11}$ $P_{mc}(e) \approx 2 \times 10^{-11}$
Standard decoding		$\boldsymbol{P_{succ}(e) = 0}, P_{mc}(e) \approx 0, P_{df}(e) \approx 1$			

- Silicon implementation: one decoder @ 38GB/s decoding throughput and 5ns decoding latency
  - Integrated into ScaleFlux CXL 3.1 memory controller (launch in early 2025)

Summary

For the very first time, bring the beyond-minimum-distance error correction into DRAM fault tolerance



- ❑ One RS list decoder: 38GB/s decoding throughput and 5ns decoding latency
- ❑ First use case: CXL memory controller (launch early 2025)
  - ➢ Support both DDR4 and DDR5 with zero-cost per-cacheline metadata embedding

Raise the Standard of DRAM Fault Tolerance

Call to Action

- How to fully exploit such stronger-than-normal error correction?
  - ✓ System reliability: the impact on the overall system reliability
  - ✓ Memory cost: deploy less reliable, lower cost DRAM chips
  - ✓ System security: mitigate security risk caused by DRAM RowHammer attacks
- Industry-wide collaboration across the device, hardware, and software stacks

Thank you!

MEMORY FABRIC FORUM

OCP GLOBAL SUMMIT

OCT 15-17, 2024
SAN JOSE, CA