XED: EXPOSING ON-DIE ERROR DETECTION INFORMATION FOR STRONG MEMORY RELIABILITY

Prashant Nair, Georgia Tech

Vilas Sridharan, AMD Inc.

Moinuddin Qureshi, Georgia Tech

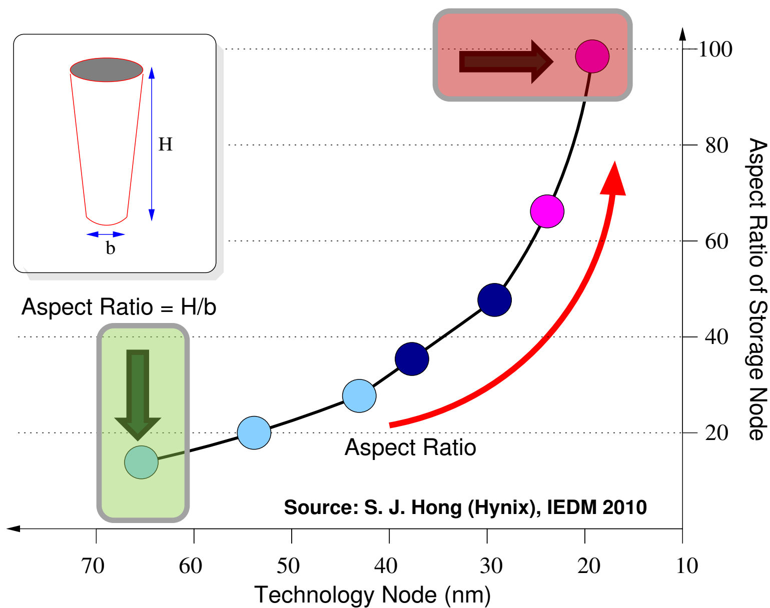
ISCA-43, June 20th 2016

Seoul, Republic of Korea

INTRODUCTION

DRAM Scaling → High Capacity Memories

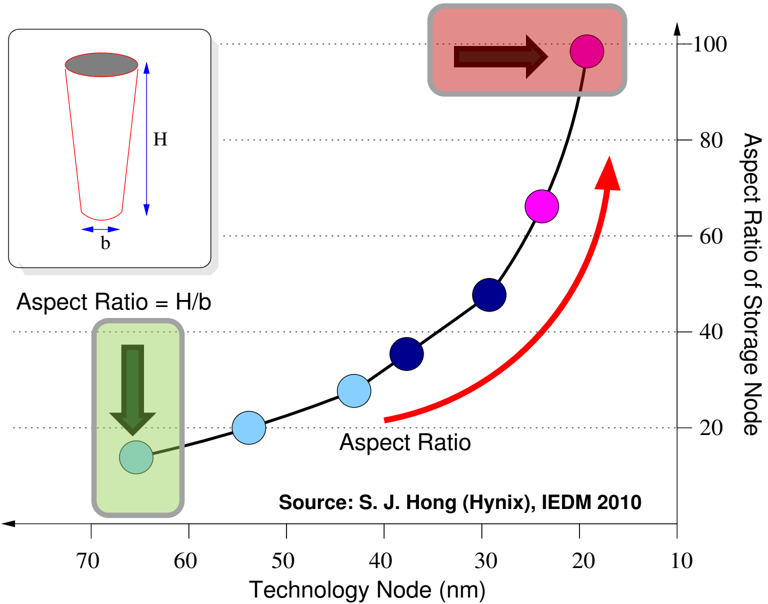
Two types of DRAM faults



INTRODUCTION

DRAM Scaling → High Capacity Memories
Two types of DRAM faults

Scaling Faults



[ArchShield ISCA'13, CiDRA HPCA'15]

Runtime Faults

Fault Mode	Transient Fault Rate (FIT)	Permanent Fault Rate (FIT)
Bit	14.2	18.6
Word	1.4	0.3
Column	1.4	5.6
Row	0.2	8.2
Bank	0.8	10
*Total	18	42.7

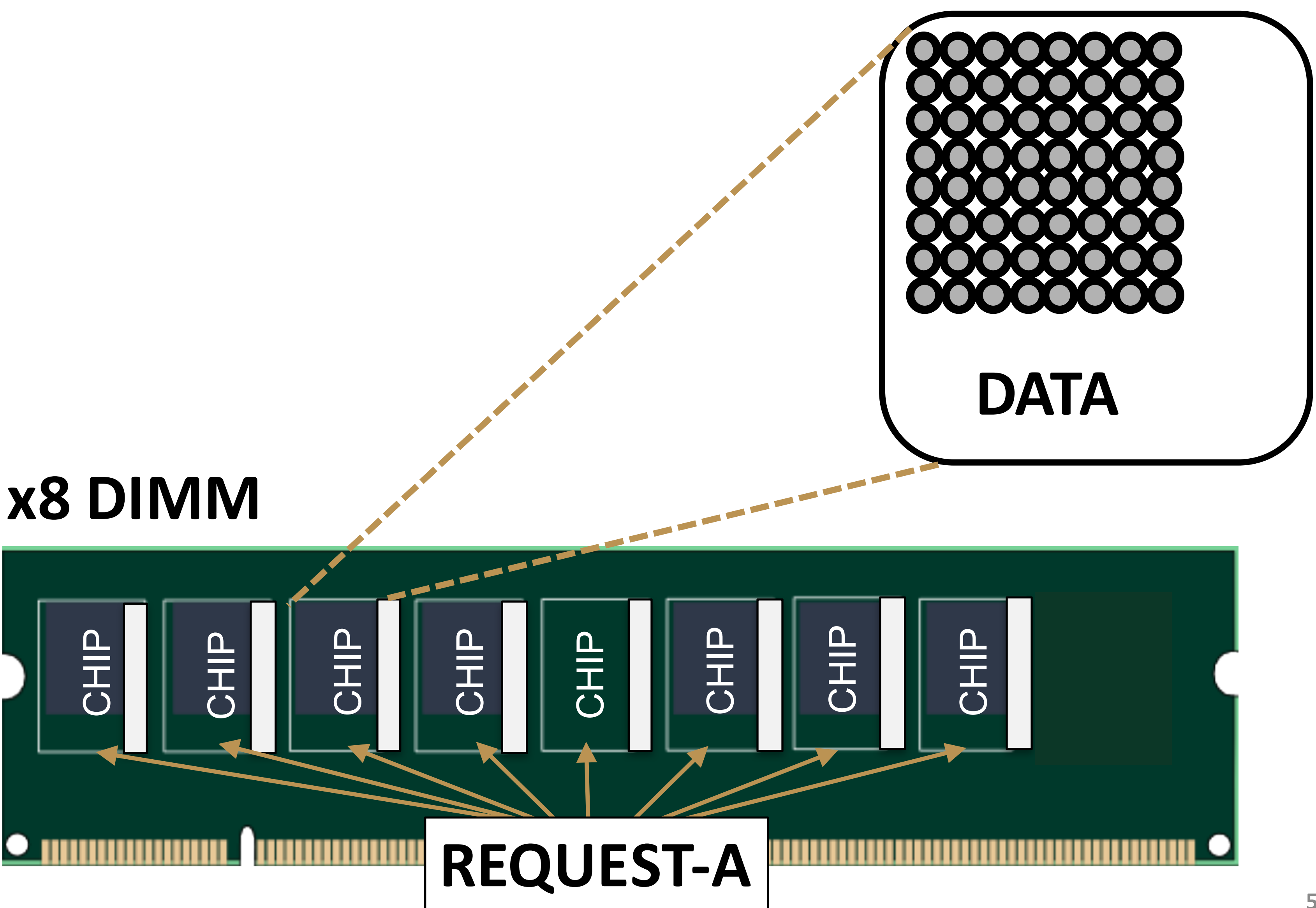
Sridharan et. al. SC13

ON-DIE ECC: MITIGATE SCALING FAULTS

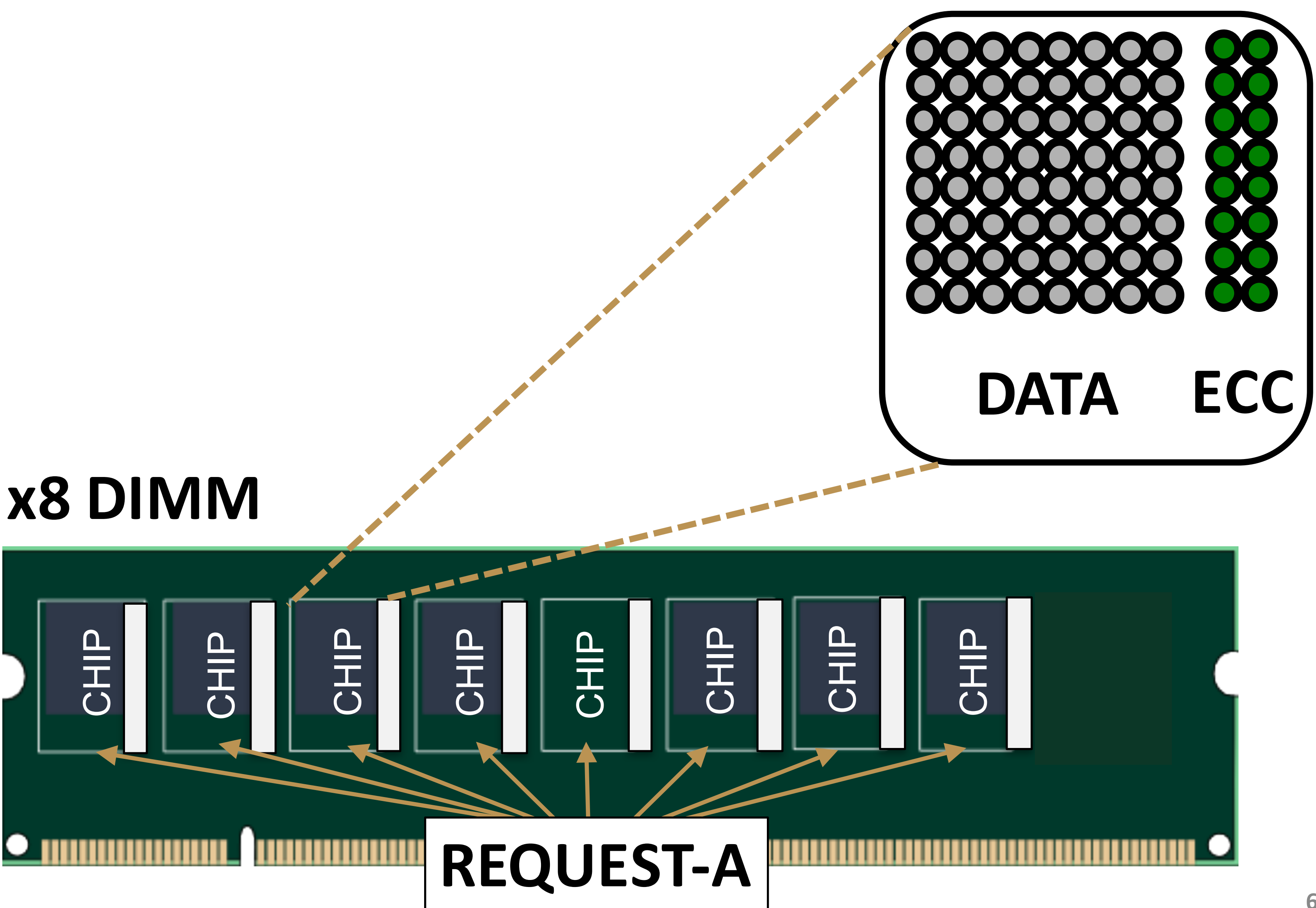
DRAM vendors plan to use "On-Die ECC"

- Mitigates scaling faults transparently
- Enables good DIMM with bad chips (yield)
- Part of: LPDDR4, DDR4, DDR5 (proposed)

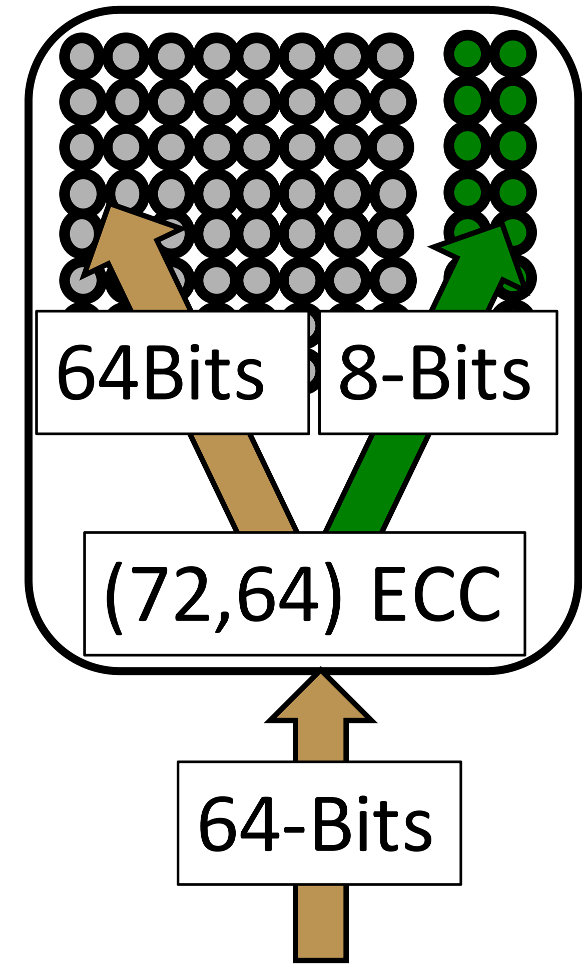
ON-DIE ECC: MITIGATE SCALING FAULTS



ON-DIE ECC: MITIGATE SCALING FAULTS

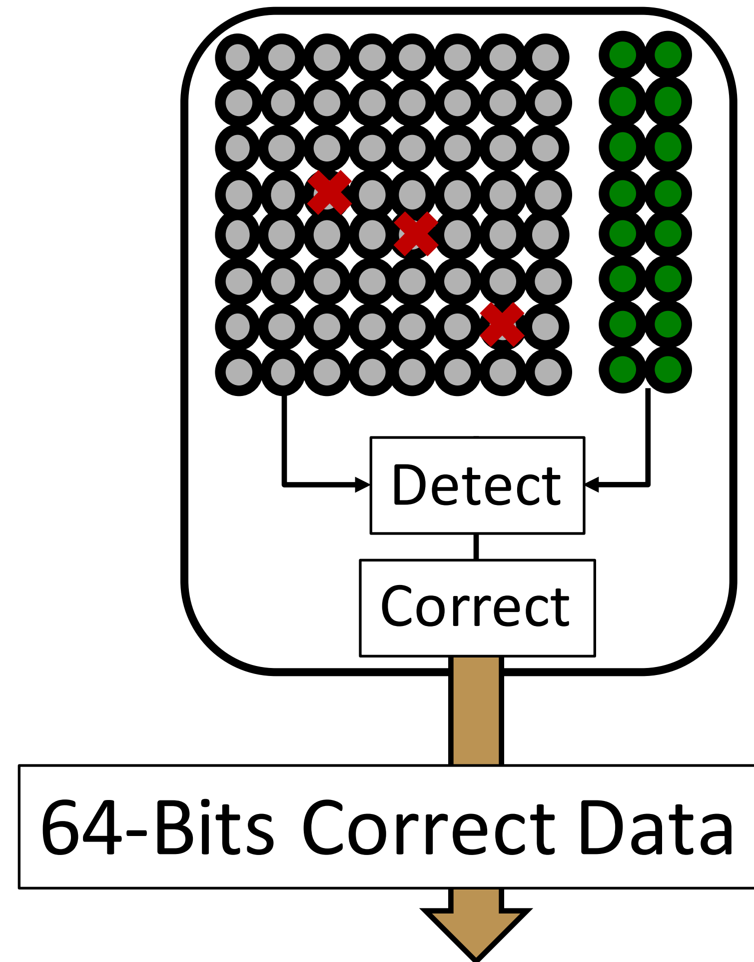


ON-DIE ECC: MITIGATE SCALING FAULTS



On-Die ECC: Single Error Correction, Double Error Detection Code (SECDED)

ON-DIE ECC: MITIGATE SCALING FAULTS



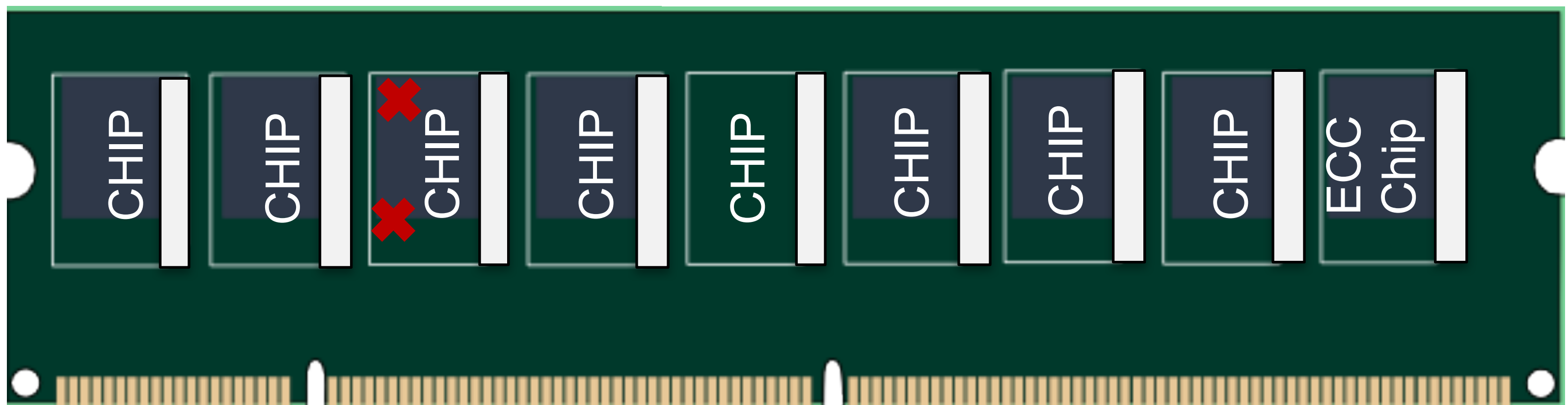
On-Die ECC fixes scaling faults invisibly

MITIGATING RUNTIME FAULTS

Runtime faults

Fault Mode	Transient Fault Rate (FIT)	Permanent Fault Rate (FIT)
Bit	14.2	18.6

ECC-DIMM (9-Chips)

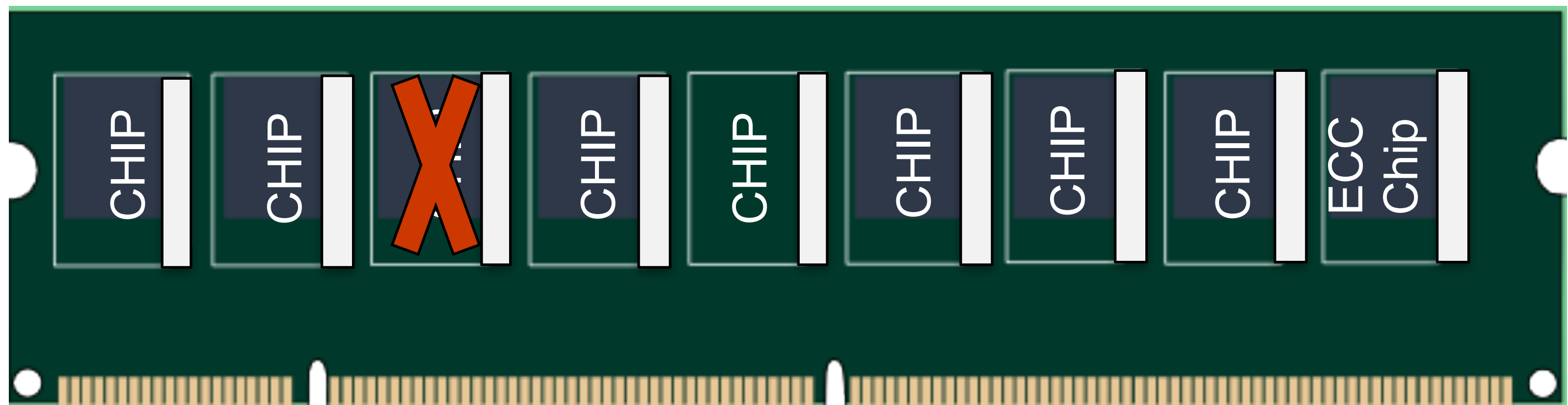


MITIGATING RUNTIME FAULTS

Runtime faults

- Chip faults common
- Need strong ECC

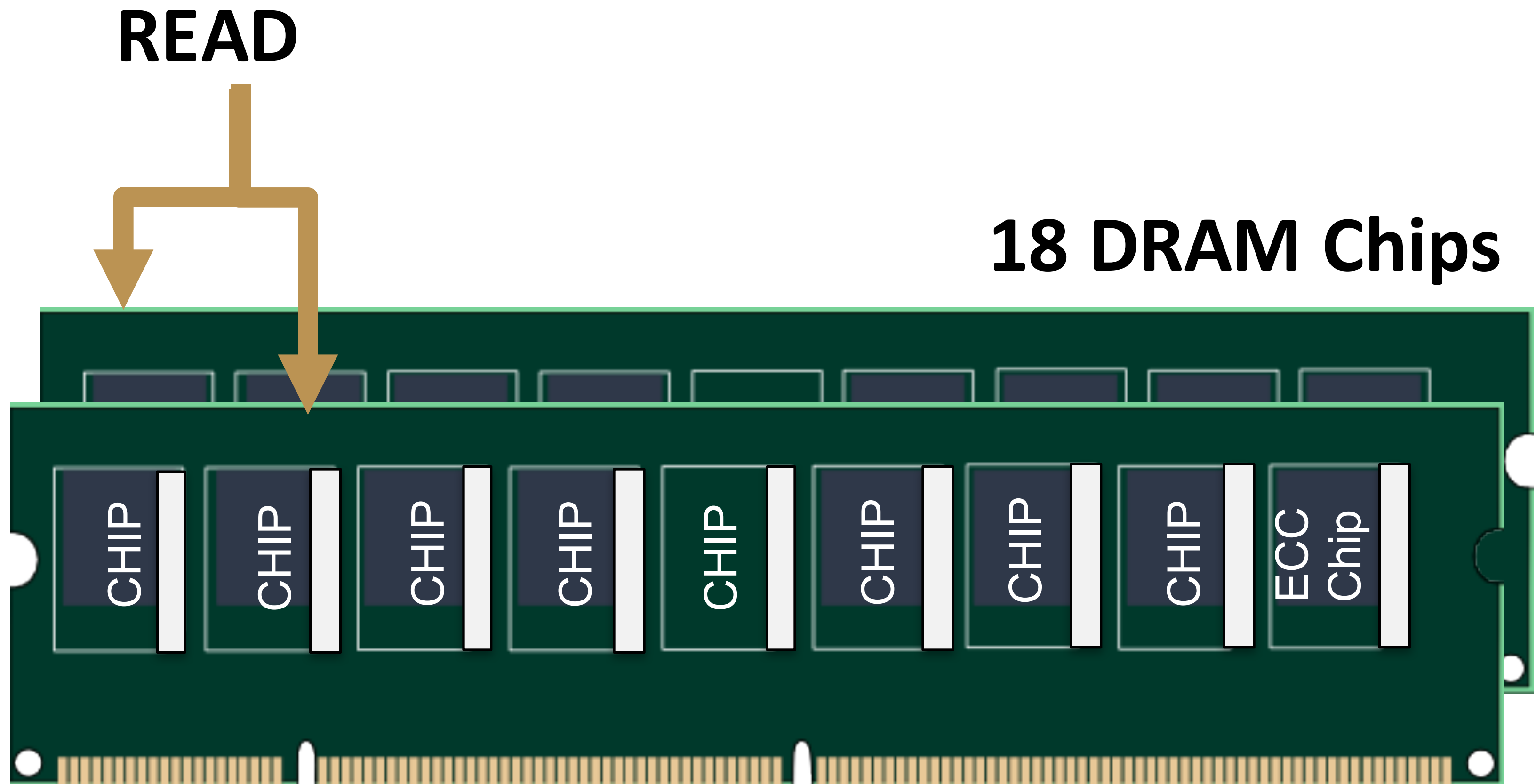
Fault Mode	Transient Fault Rate (FIT)	Permanent Fault Rate (FIT)
Bit	14.2	18.6
Word	1.4	0.3
Column	1.4	5.6
Row	0.2	8.2
Bank	0.8	10
*Total	18	42.7



*Sridharan+ SC13

MITIGATING RUNTIME FAULTS

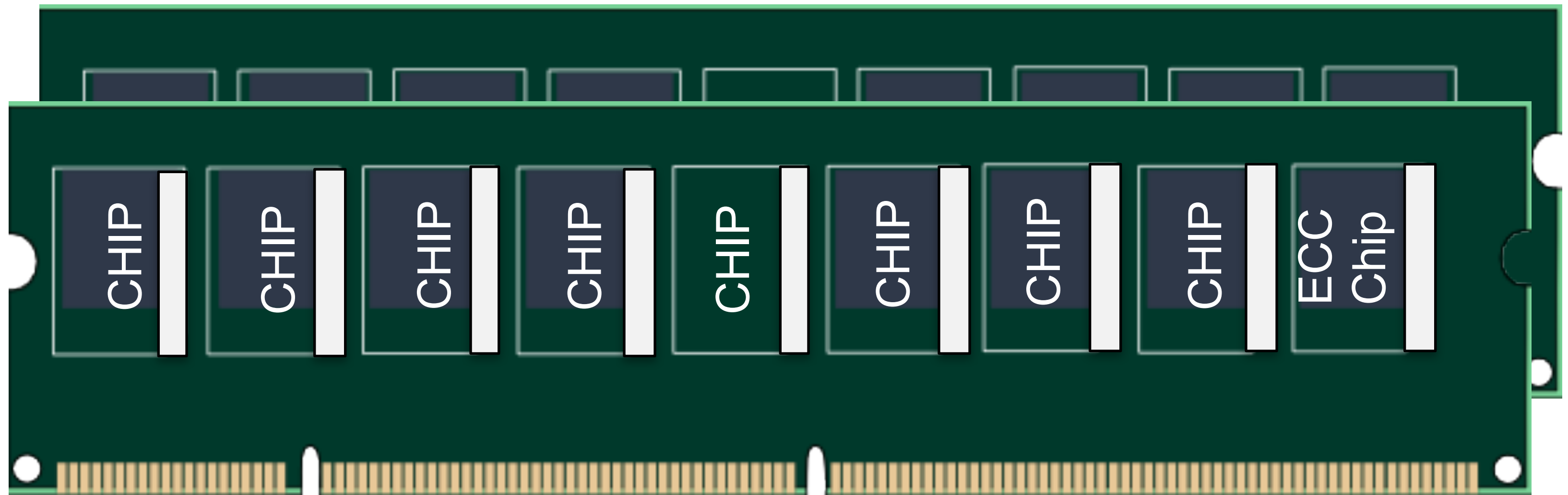
Runtime chip faults → Chipkill (strong ECC)



MITIGATING RUNTIME FAULTS

Runtime chip faults → Chipkill (strong ECC)

18 DRAM Chips



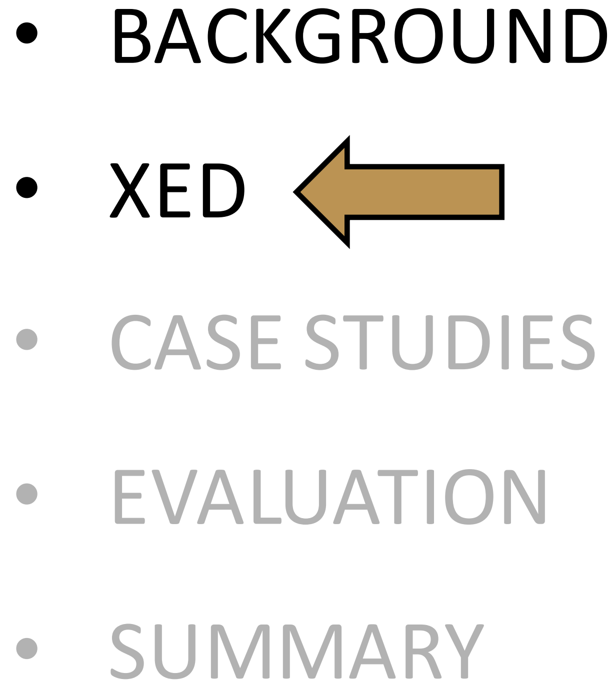
Cost: 18 Chips, Performance and Power Inefficient

GOAL AND CHALLENGE

GOAL: Use On-Die ECC to mitigate runtime faults
"Chipkill-level reliability using x8 ECC-DIMM"

CHALLENGE: On-Die ECC is invisible, expose it without changing the memory interface

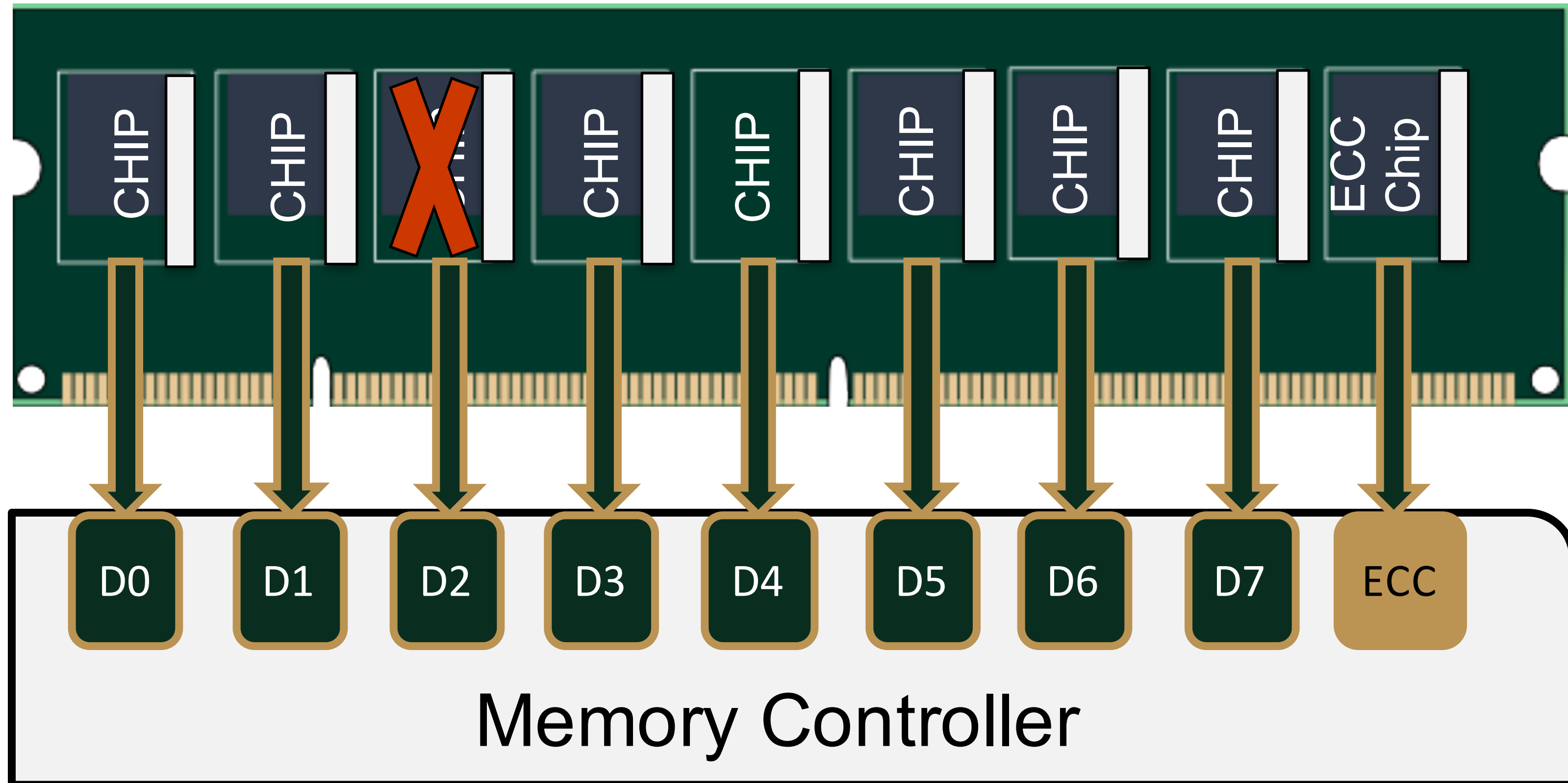
OUTLINE



- BACKGROUND
- XED
- CASE STUDIES
- EVALUATION
- SUMMARY

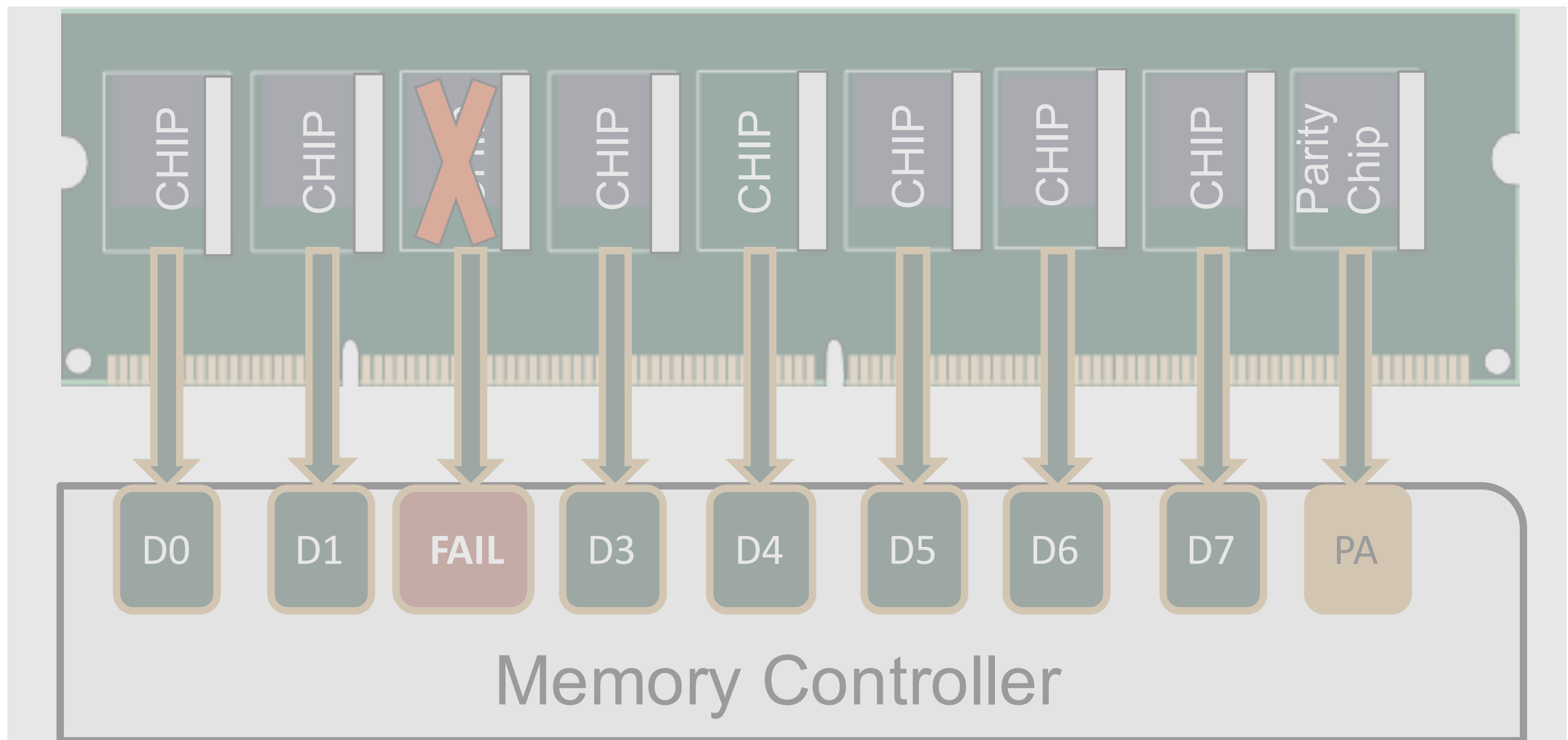
USING PARITY + FAILED LOCATION

What if the chip can inform that it failed?



USING PARITY + FAILED LOCATION

What if the chip can inform that it failed?



Parity + Location → Reconstruct Data for Faulty Chip

Fix chip-faults using only 9 Chips

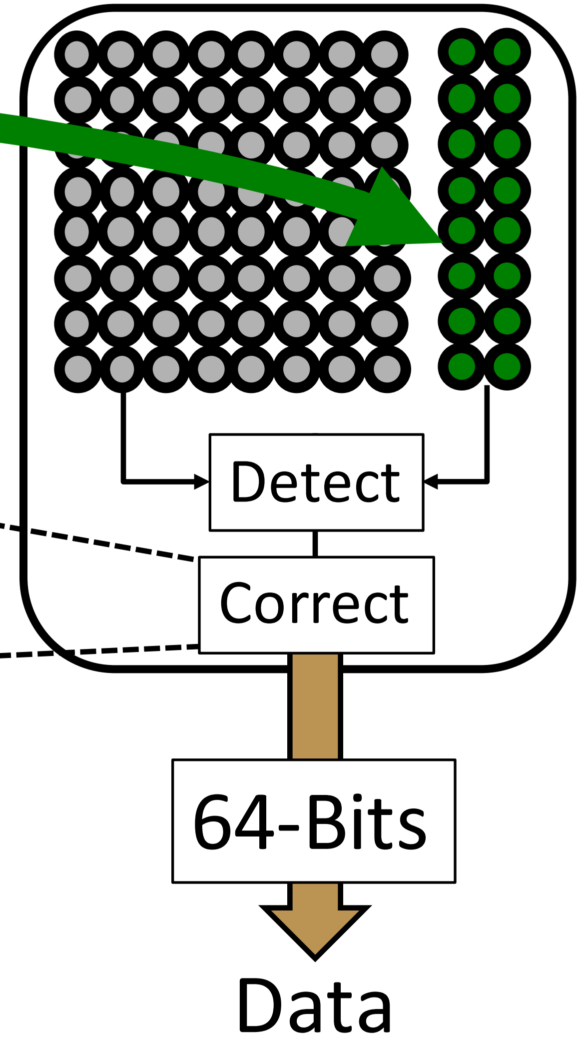
XED: EXPOSED ON-DIE ERROR DETECTION

XED consists of three components

- Strong detection in addition to SEC
- Parity-based correction
- Transparently identifying faulty chip

XED: ON-DIE ECC AS DETECTION CODE

On-Die Error Correction Code



	Corrects?
Single-Bit Failures	✔
Chip Failures	✖

XED: ON-DIE ECC AS DETECTION CODE

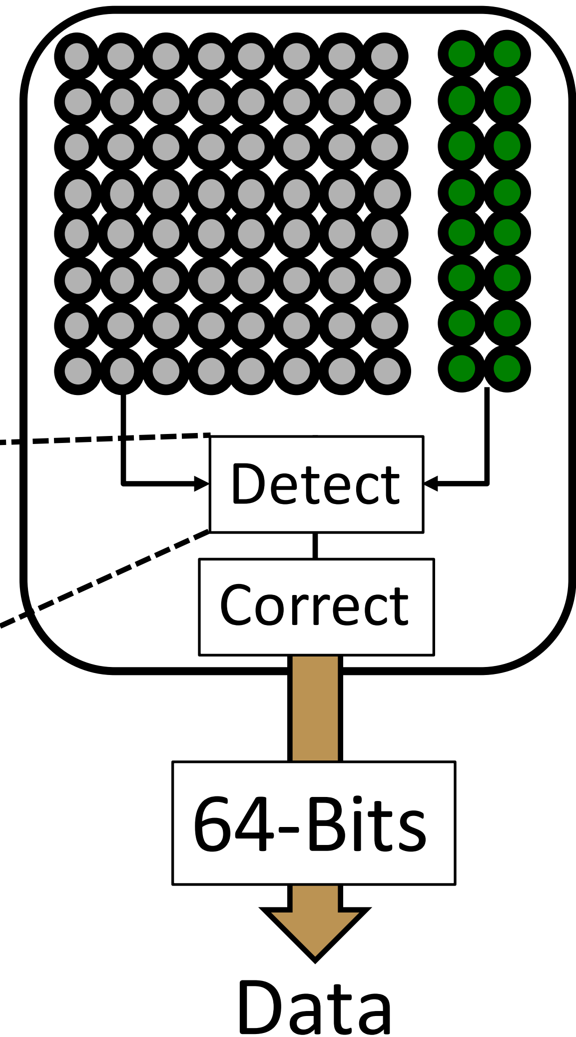
On-Die Error Strong Detection

+

Correction Code

	Corrects?	Detects?
Single-Bit Failures	✔	✔
Chip Failures	✖	✔ (99.9%)

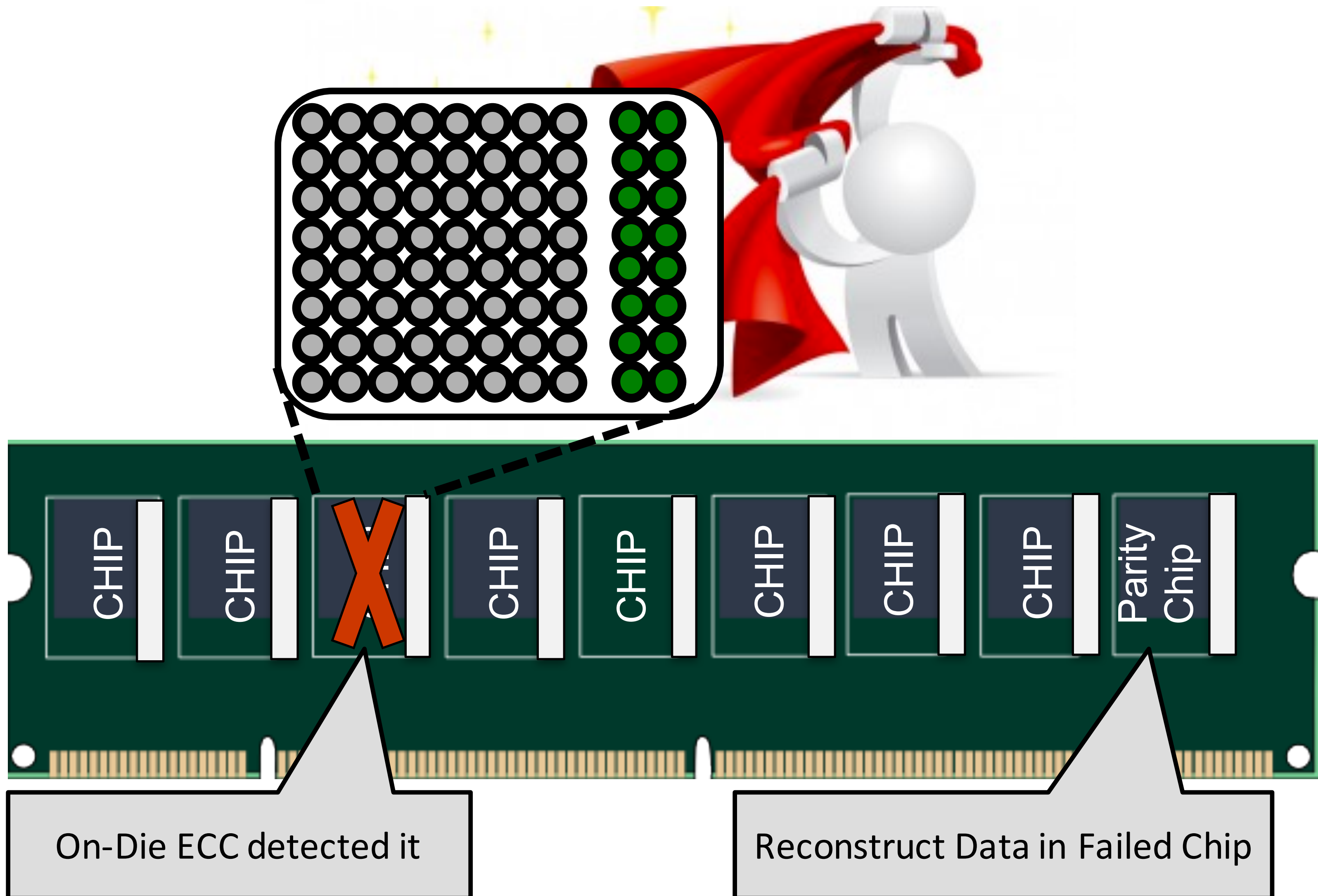
CRC-8 ATM-code instead of Hamming-code



On-Die ECC can detect chip-failures

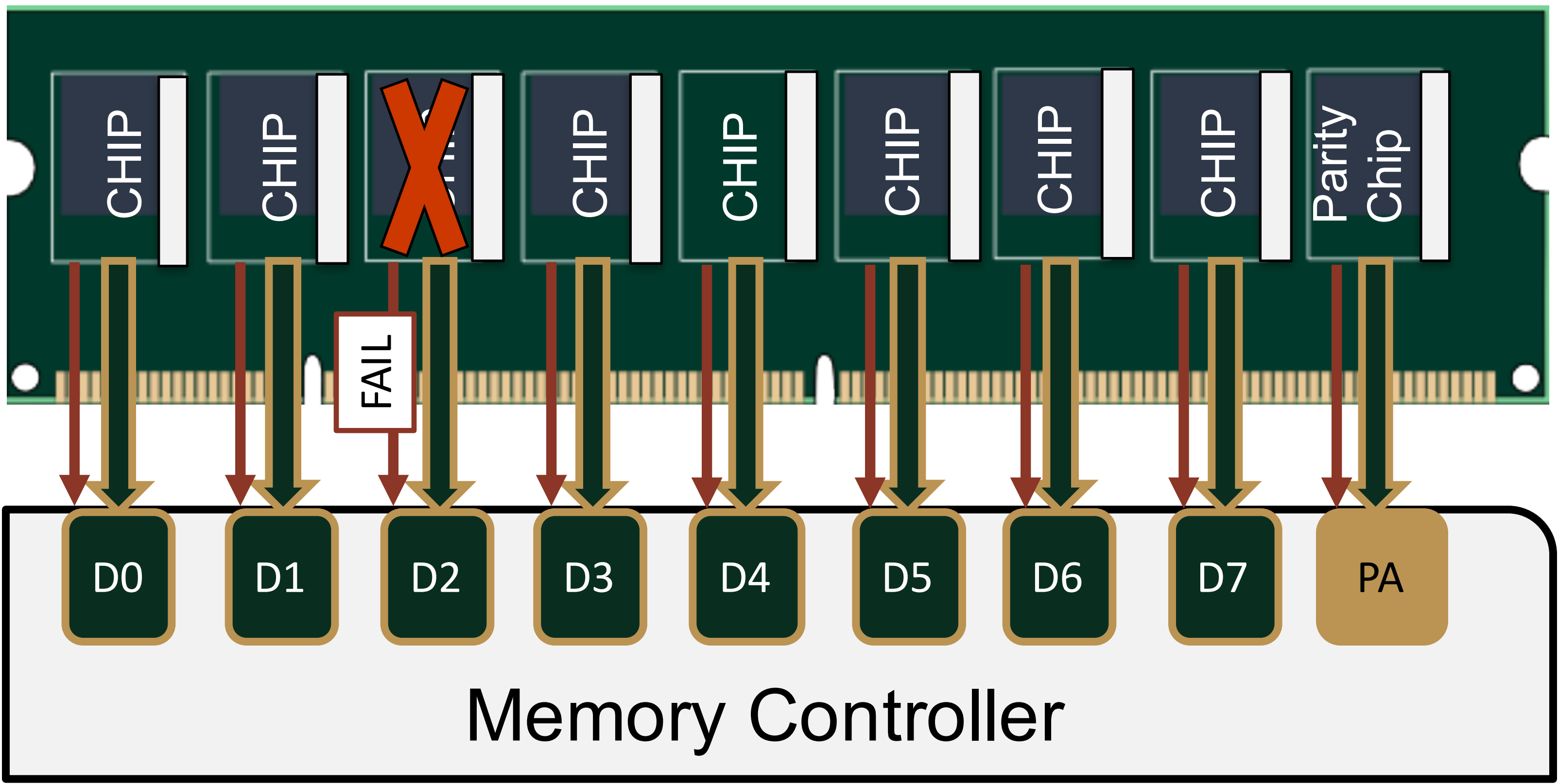
XED: RAID-3 BASED CORRECTION

If we could expose On-Die Error Detection → Chipkill



EXPOSE ON-DIE ERROR INFO

OPTION 1: Use additional wires



EXPOSE ON-DIE ERROR INFO

OPTION 1: Use additional wires

Incompatible with DDR memory standards

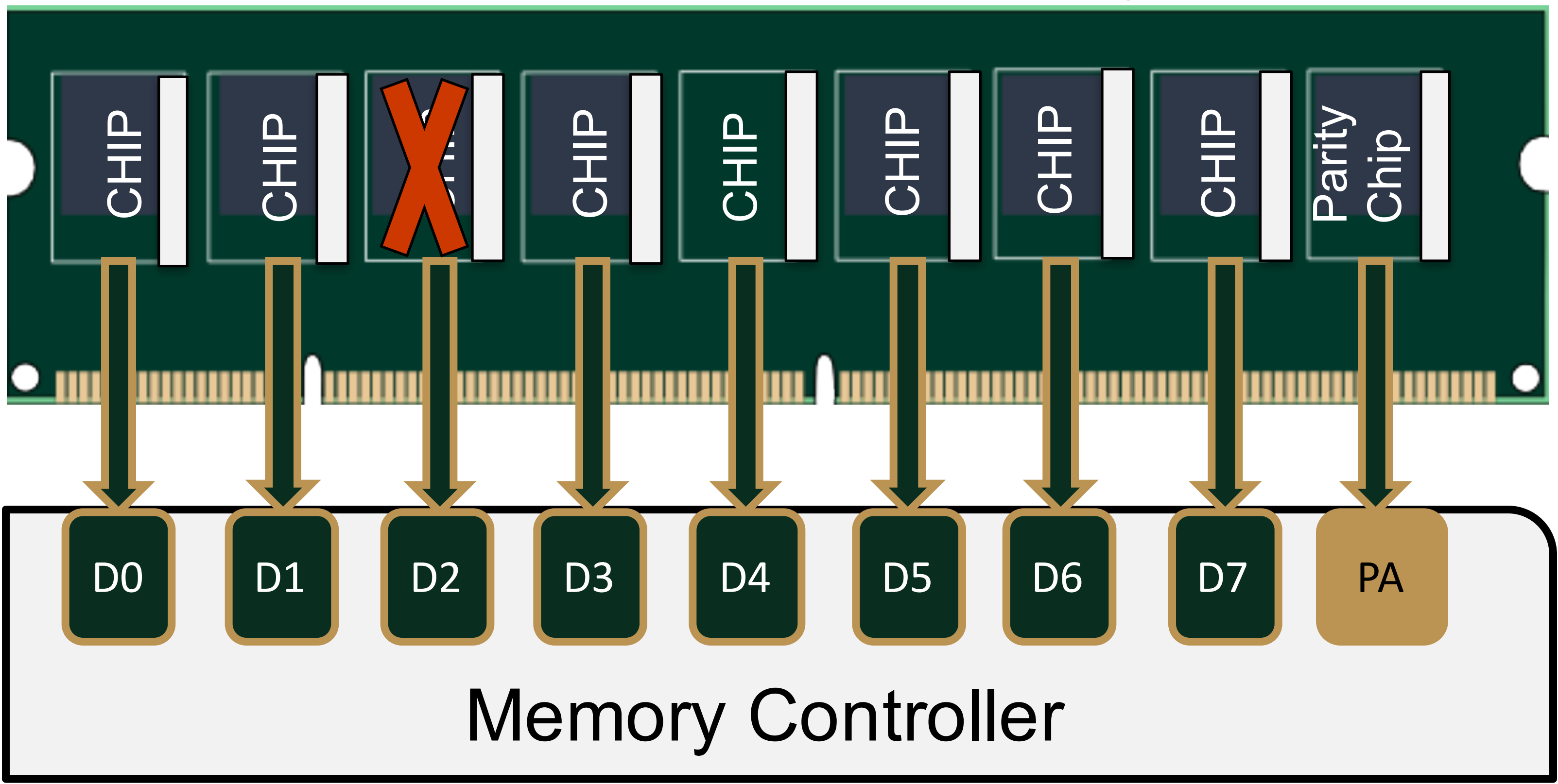
Needs a new protocol

Worse for pin-constrained future systems!

Memory Controller

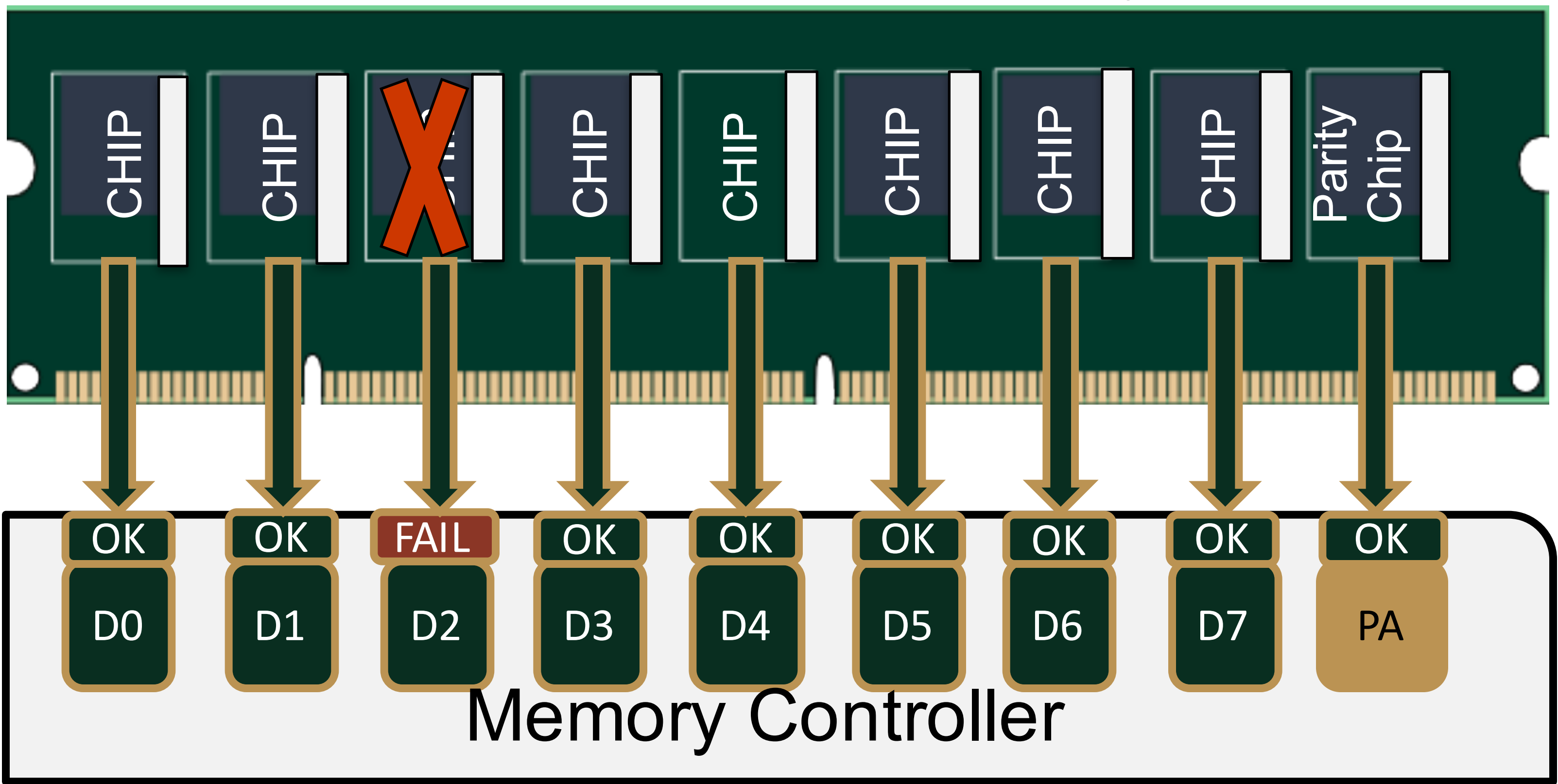
EXPOSE ON-DIE ERROR INFO

OPTION 2: Use additional burst/transaction



EXPOSE ON-DIE ERROR INFO

OPTION 2: Use additional burst/transaction

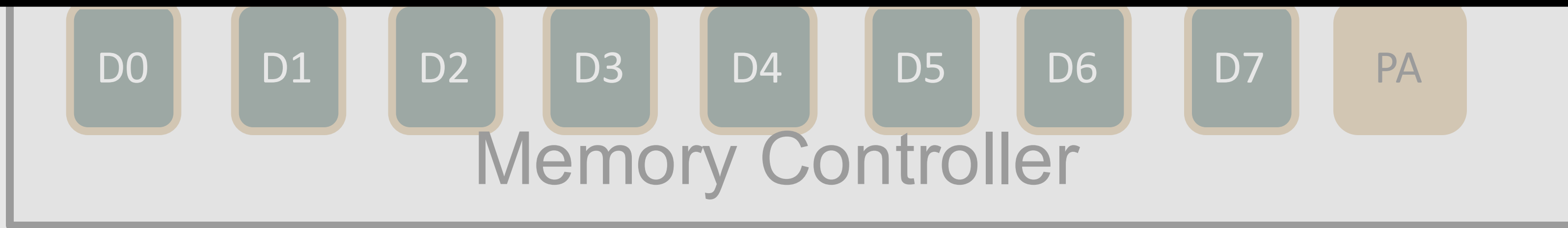


EXPOSE ON-DIE ERROR INFO

OPTION 2: Use additional burst/transaction

Additional 12.5% to 100% bandwidth overheads

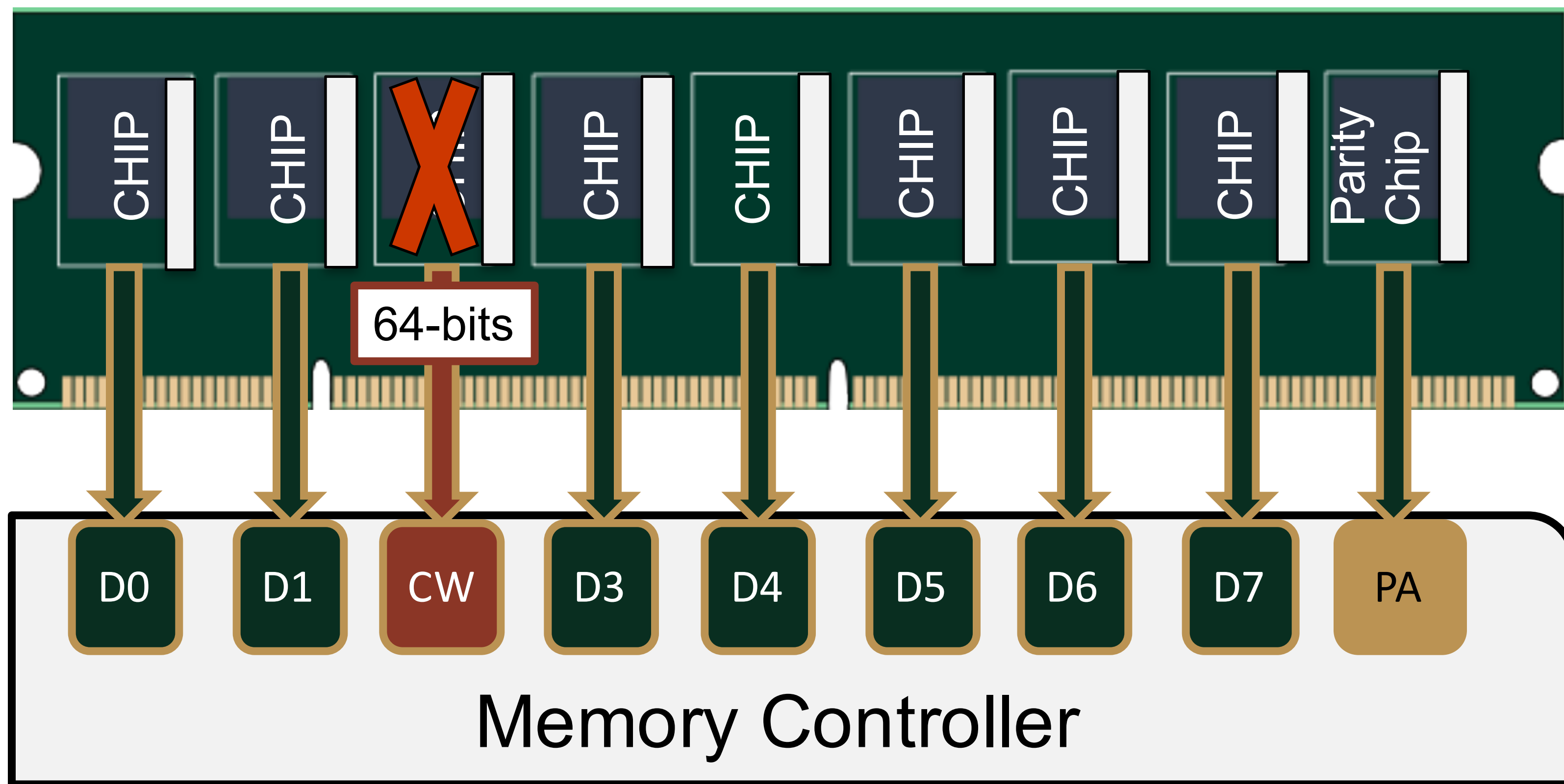
Performance and Power Inefficient



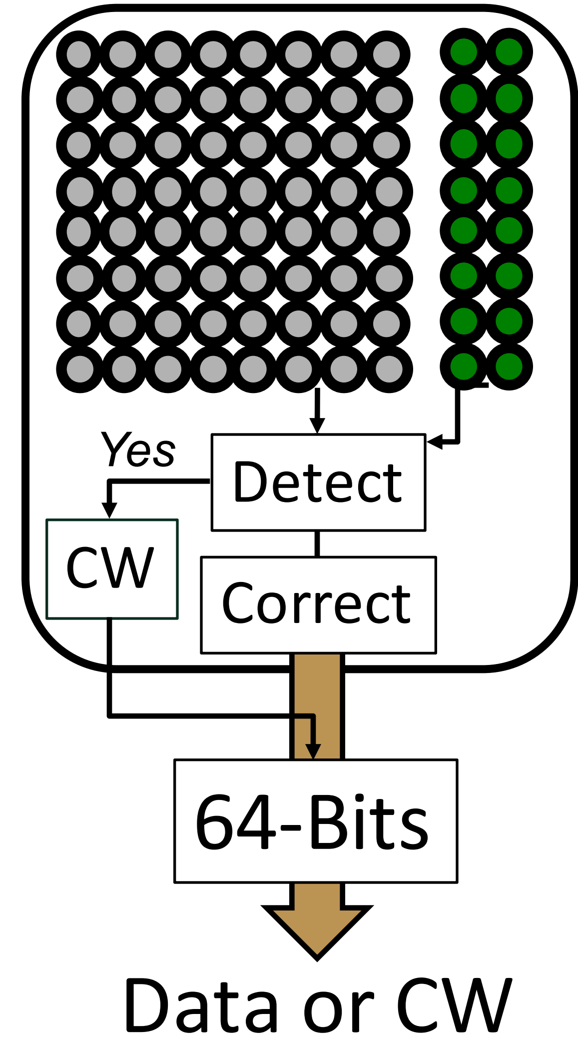
Expose On-Die error detection with minor changes

XED: ON-DIE ERROR INFO FOR FREE

On detecting an error, the DRAM chip sends a 64-bit "Catch-Word" (CW) instead of data



XED: MUX TO SEND CATCH-WORDS



Simple MUX to chose between Data and Catch-Word

XED: ON-DIE ERROR INFO FOR FREE

On detecting an error, the DRAM chip sends a 64-bit “Catch-Word” (CW) instead of data

Chips provisioned with a unique Catch-Word

No additional wires/bandwidth overheads

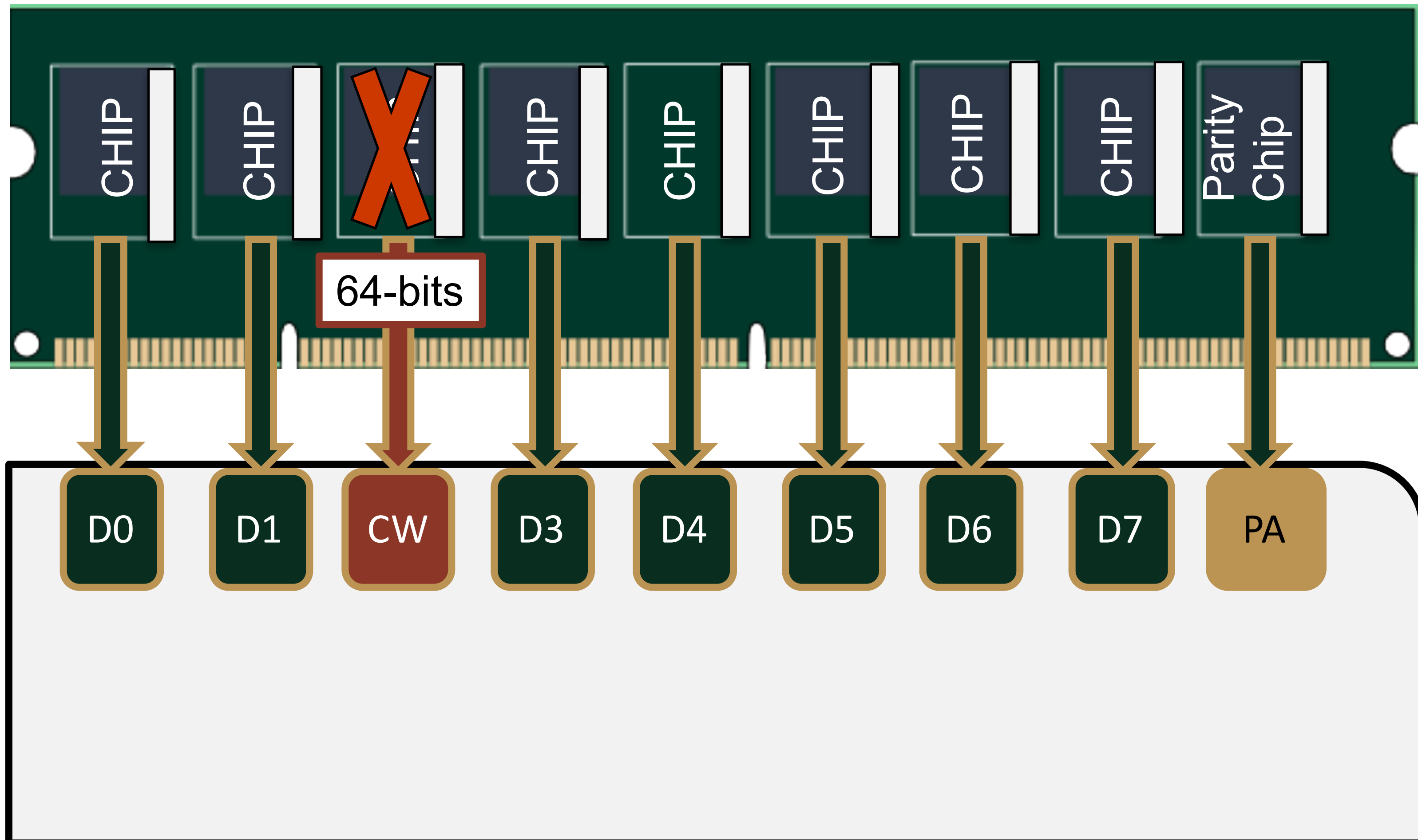
Compatible with existing memory protocols

Memory Controller

64-bit Catch-Words identify the faulty chip

WHY DO CATCH-WORDS WORK?

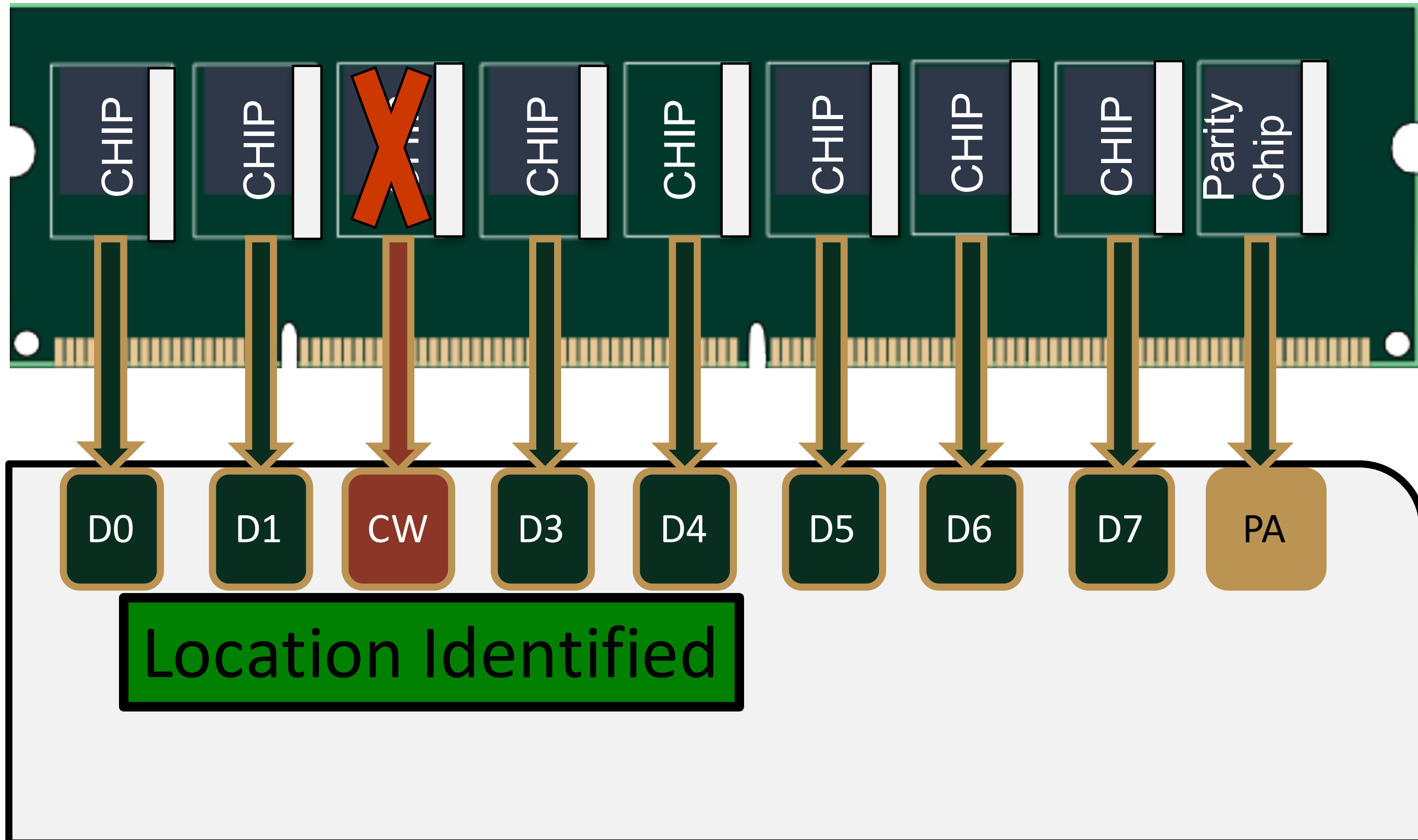
Catch Word (CW) ≠ Valid Data (D2)



WHY DO CATCH-WORDS WORK?

Catch Word (CW) ≠ Valid Data (D2)

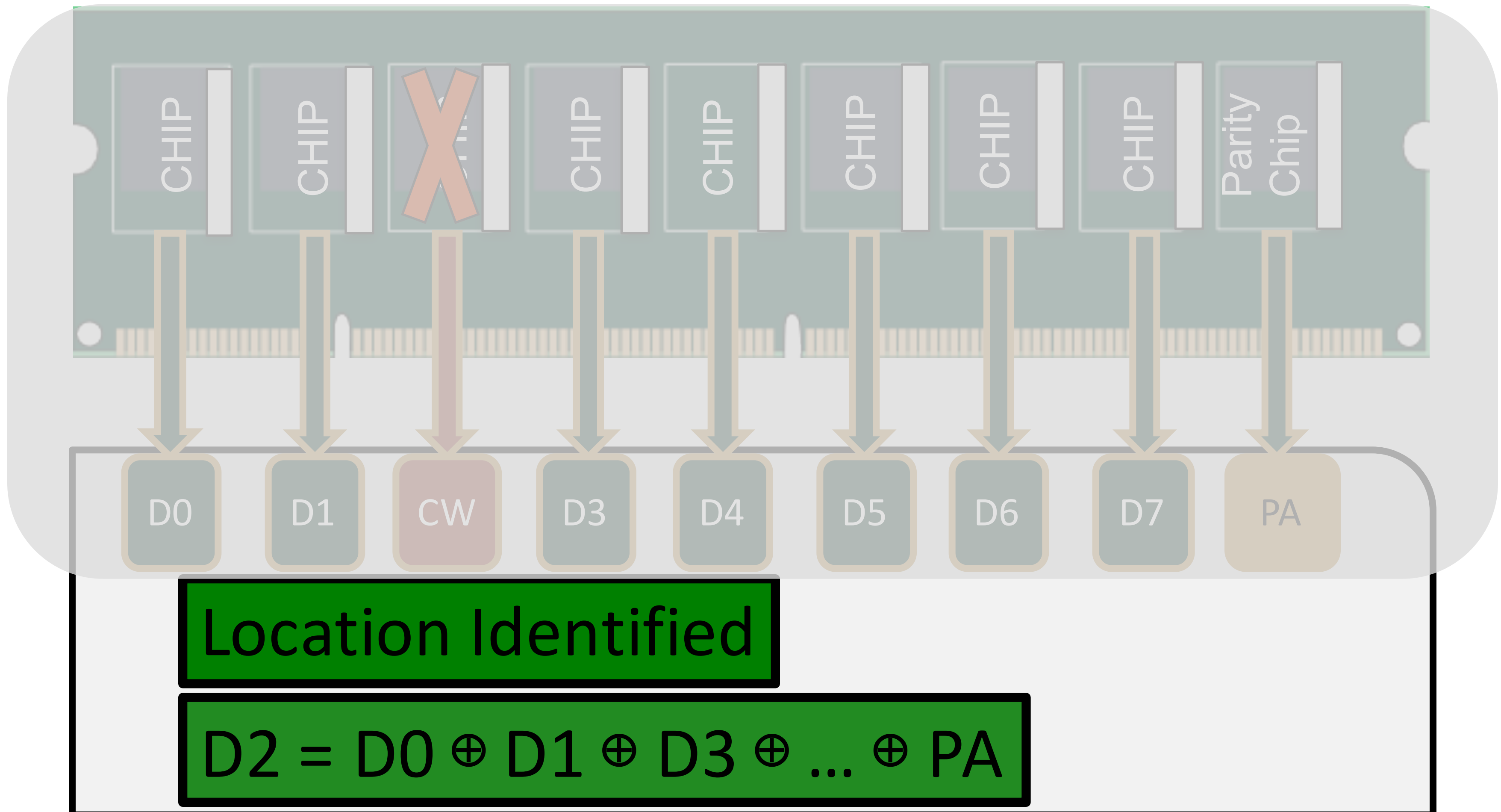
Then → PA ≠ D0 ⊕ D1 ⊕ CW ⊕ … ⊕ D7



WHY DO CATCH-WORDS WORK?

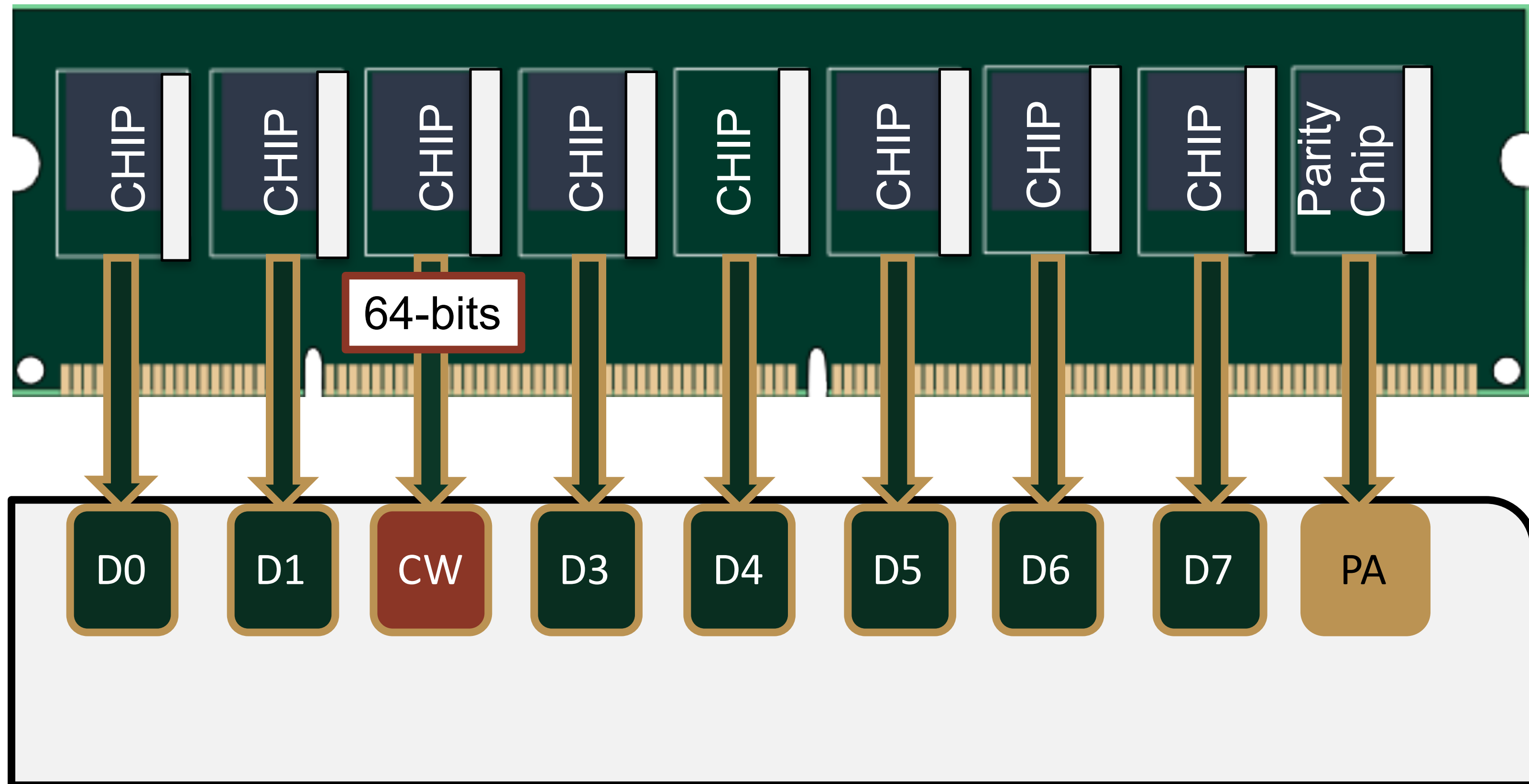
Catch Word (CW) ≠ Valid Data (D2)

Then → $PA \neq D0 \oplus D1 \oplus CW \oplus \ldots \oplus D7$



WHY DO CATCH-WORDS WORK?

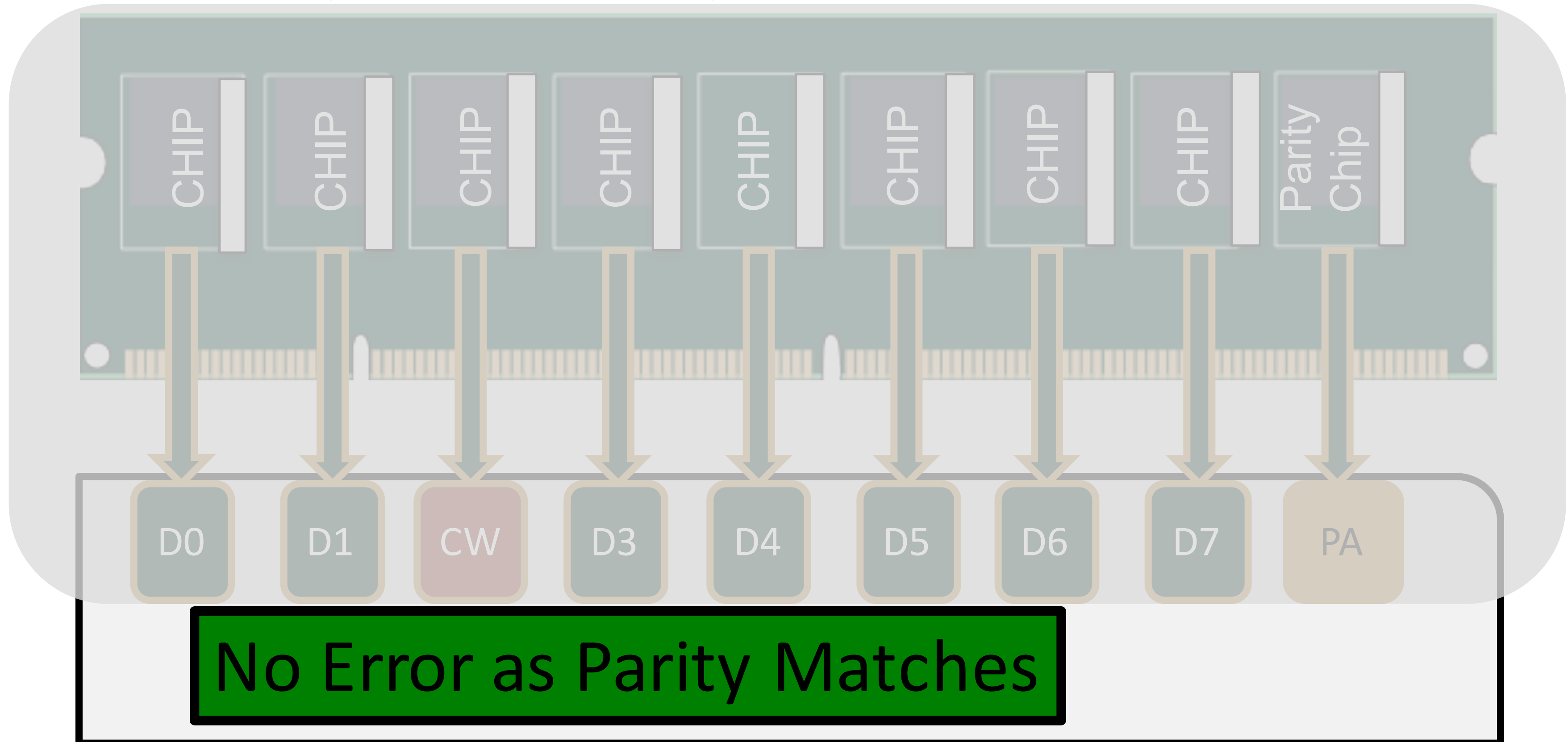
Catch Word (CW) = Valid Data (D2)



WHY DO CATCH-WORDS WORK?

Catch Word (CW) = Valid Data (D2) [*Collision*]

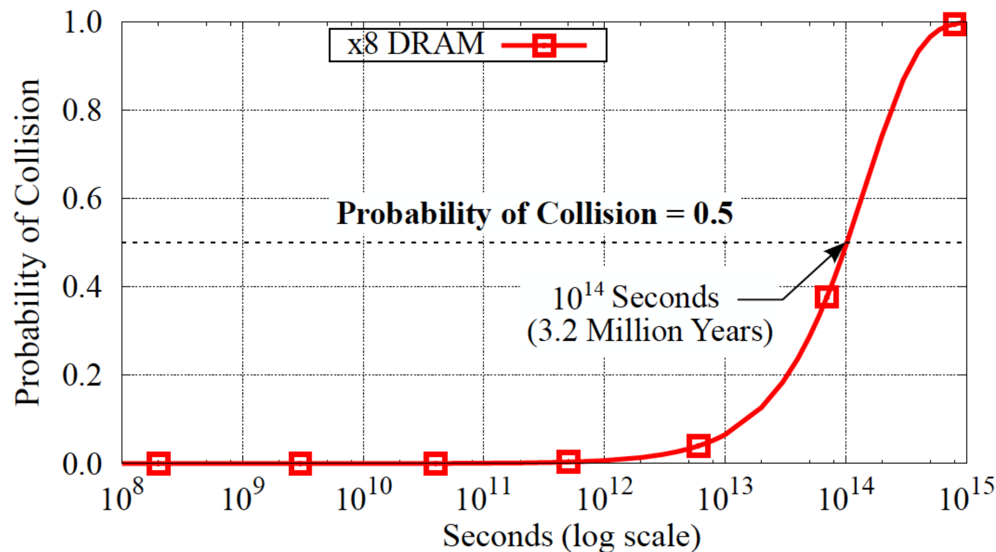
Then → PA = D0 ⊕ D1 ⊕ CW ⊕ … ⊕ D7



Catch-Word collision: Doesn't affect correctness

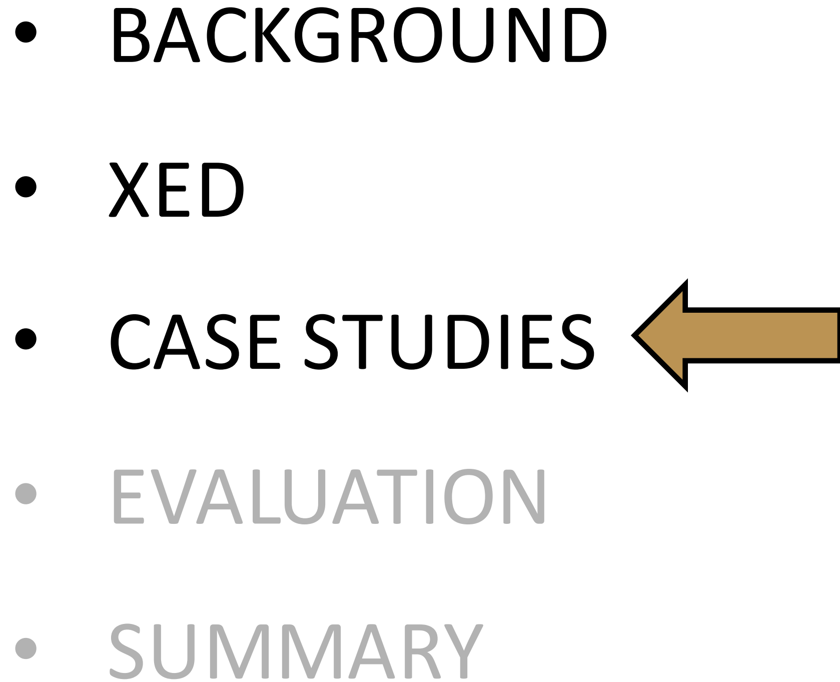
COLLISIONS: NOT A PROBLEM

- A chip stores 64 bits/cache-line → 2^{64} combinations
- However even a 16Gb chip has only 2^{28} cachelines
- Even if this entire chip contained different data there are nearly $2^{63.99}$ data combinations free!



The catch-word will most likely not collide

OUTLINE



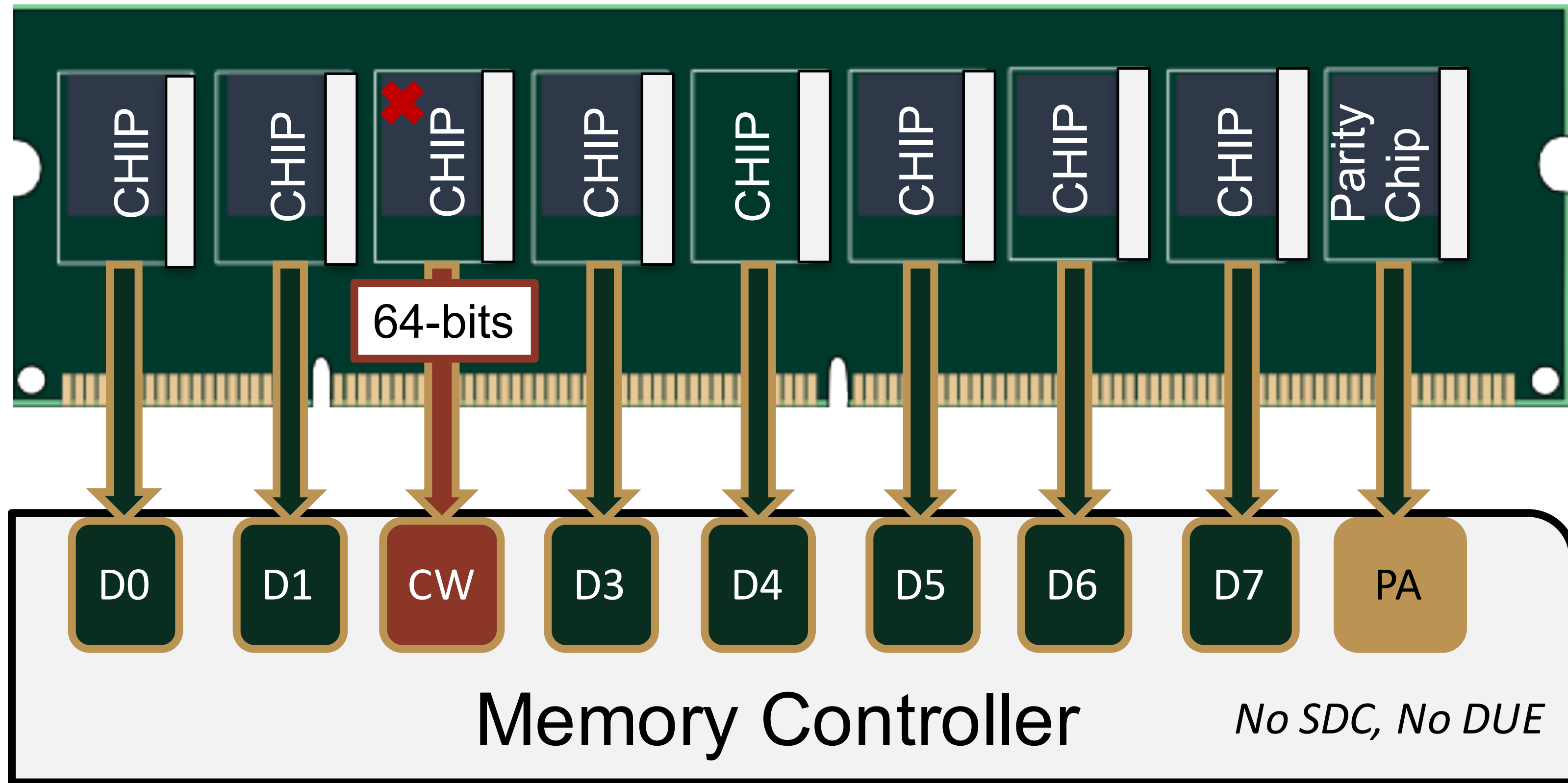
XED FOR SCALING ERRORS

On-Die ECC

- Single Error Correction
- Always detects scaling errors (single-bit)

CASE STUDY 1: SINGLE SCALING FAULT

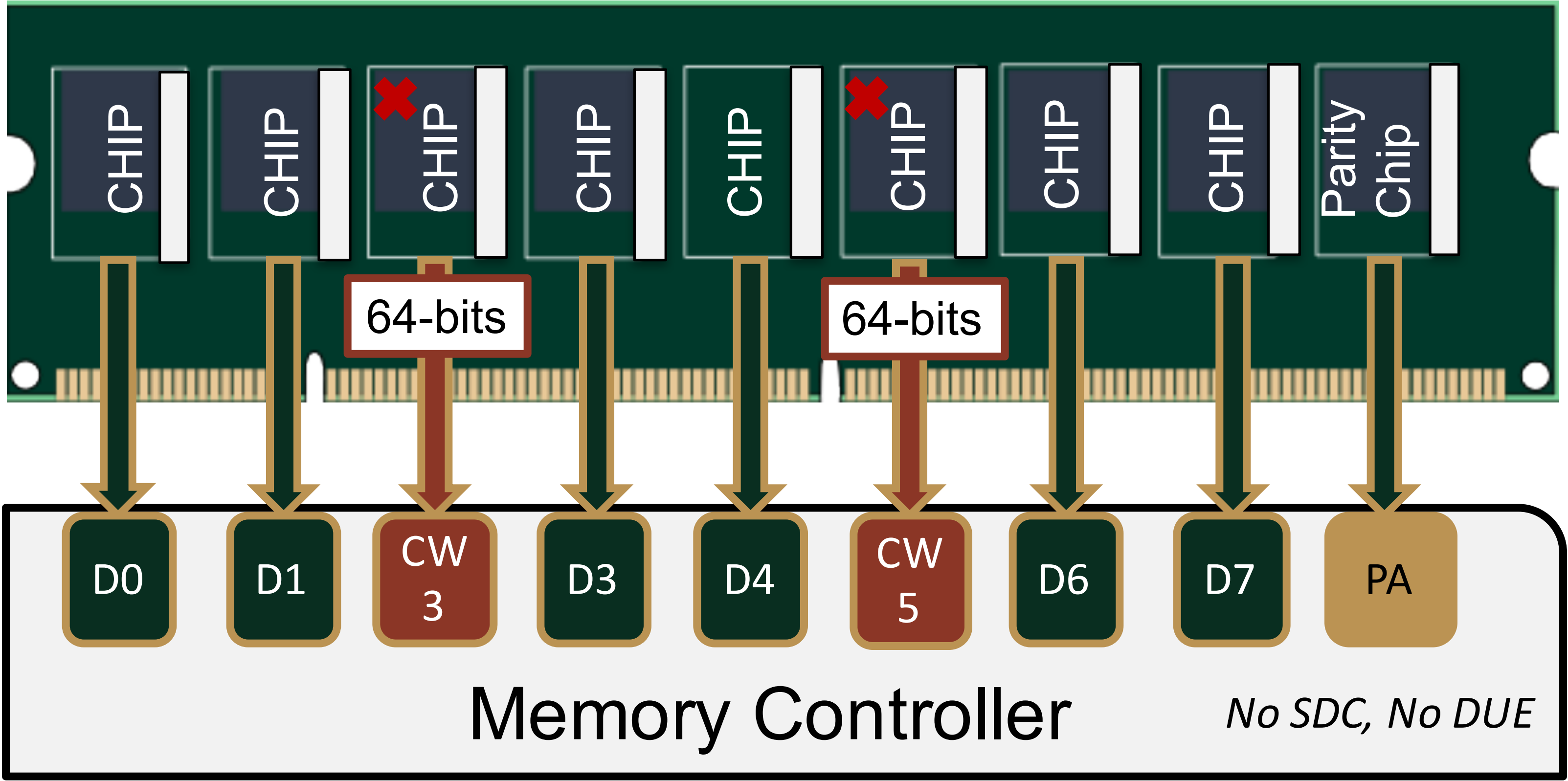
Scaling fault within a single chip



Parity reconstructs data from chip with scaling error

CASE STUDY 2: MULTIPLE SCALING FAULTS

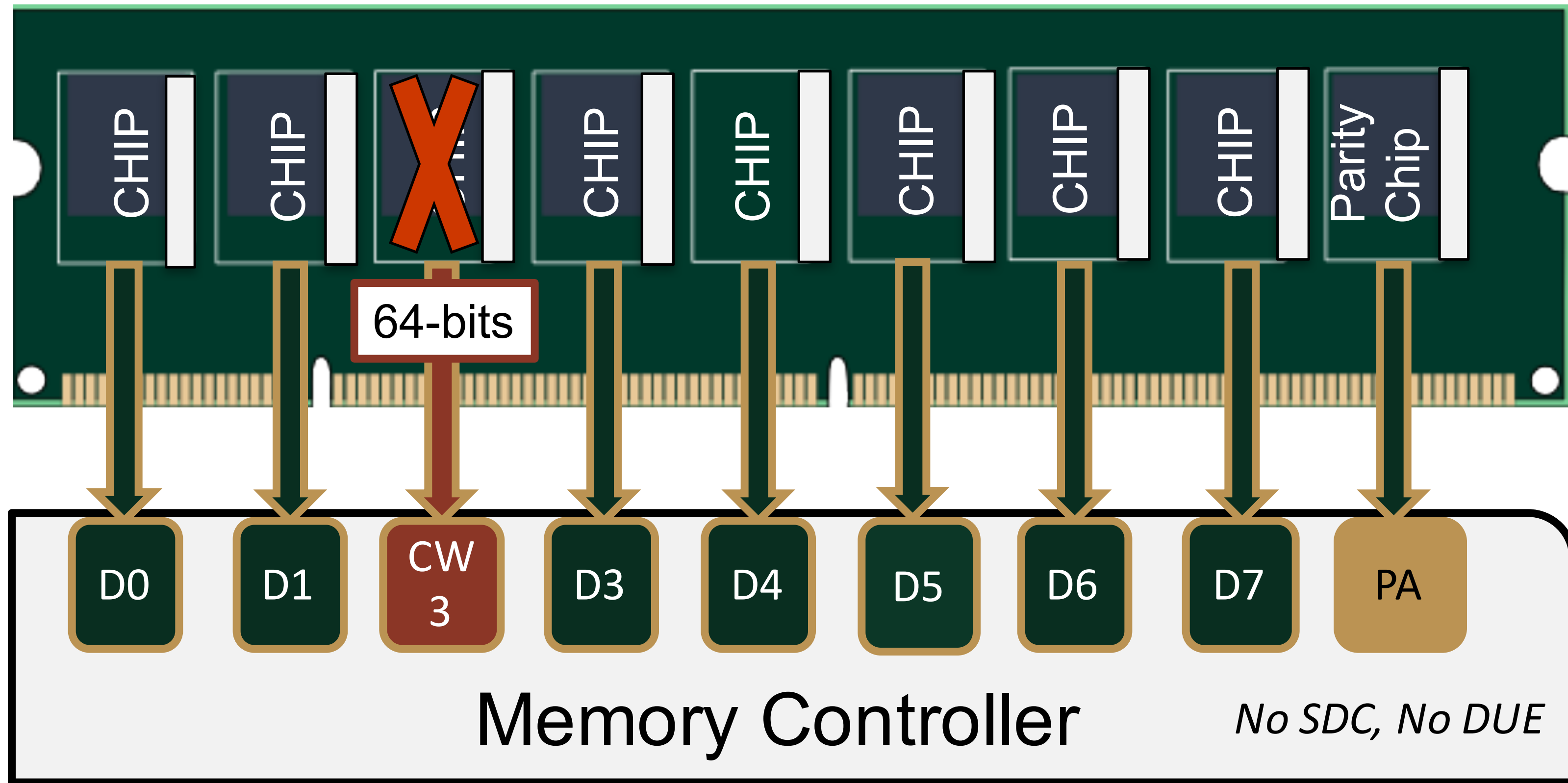
Scaling faults within multiple chips



Disable XED + Retry

CASE STUDY 3: CHIP FAULT

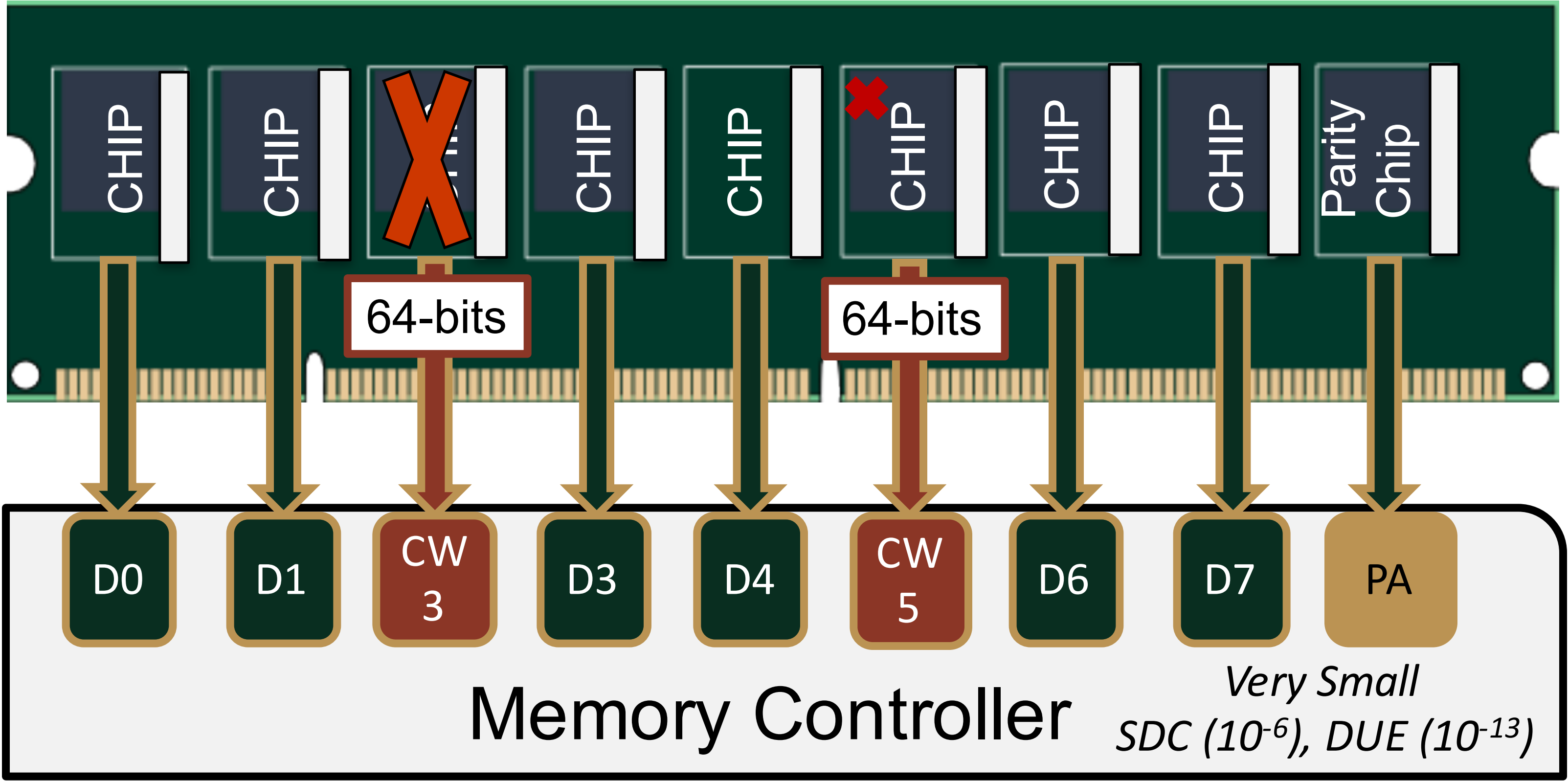
Catch-Word identifies the faulty chip



Parity reconstructs data from failed chip

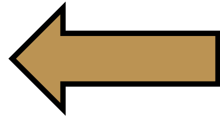
CASE STUDY 4: CHIP + SCALING FAULT

Parity detects error even after retry → Chip Failure



Disable XED + Diagnosis to locate chip failure

OUTLINE

- BACKGROUND
- XED
- CASE STUDIES
- EVALUATION 
- SUMMARY

EVALUATION

USIMM : 8 Cores, 4 Channels, 2 Ranks, 8 Banks

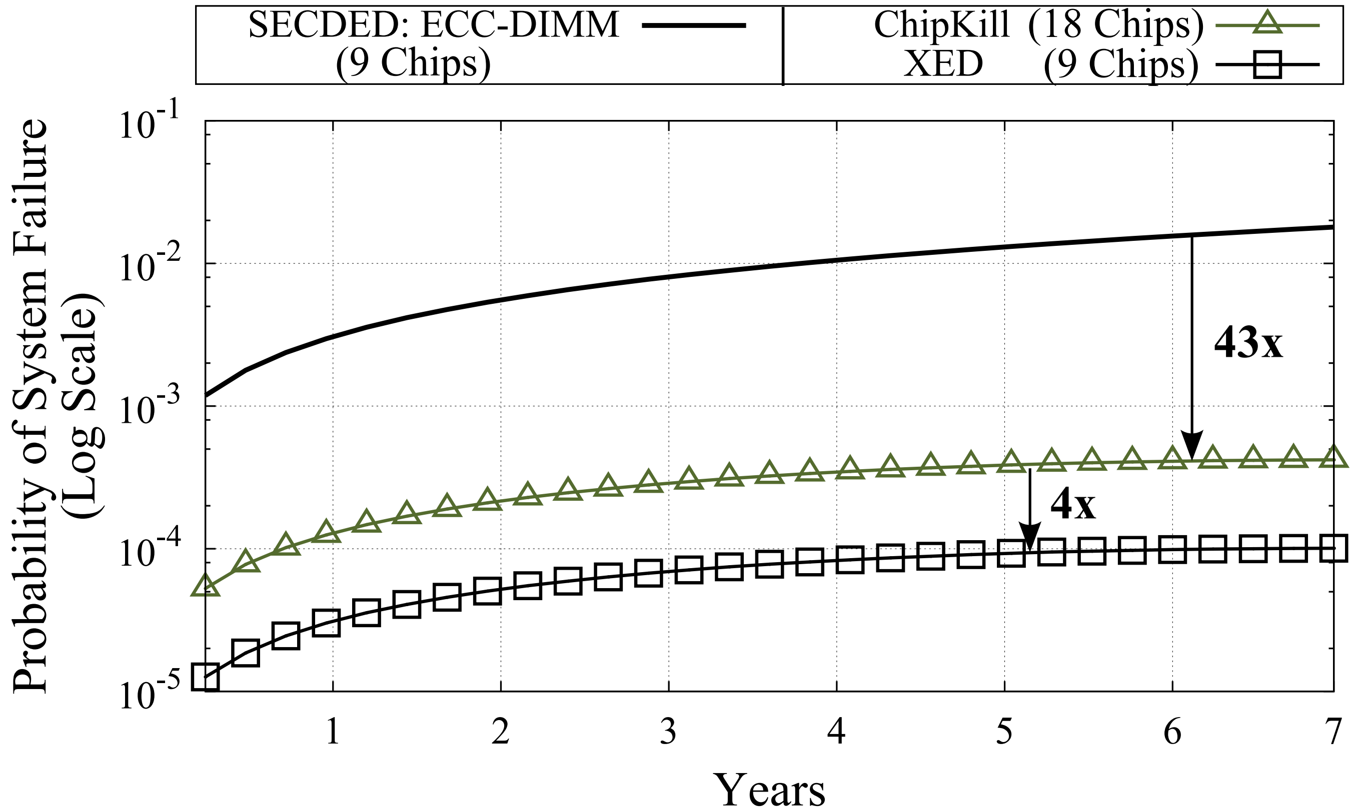
FaultSim*: Memory Reliability Simulator

- Real World Fault Data
- 7 year system lifetime,
- Billion Monte-Carlo Trails
- Metric: Probability of System Failure
- Scaling Fault-Rate: 10^{-4}

* Nair et. al. HiPEAC 2016

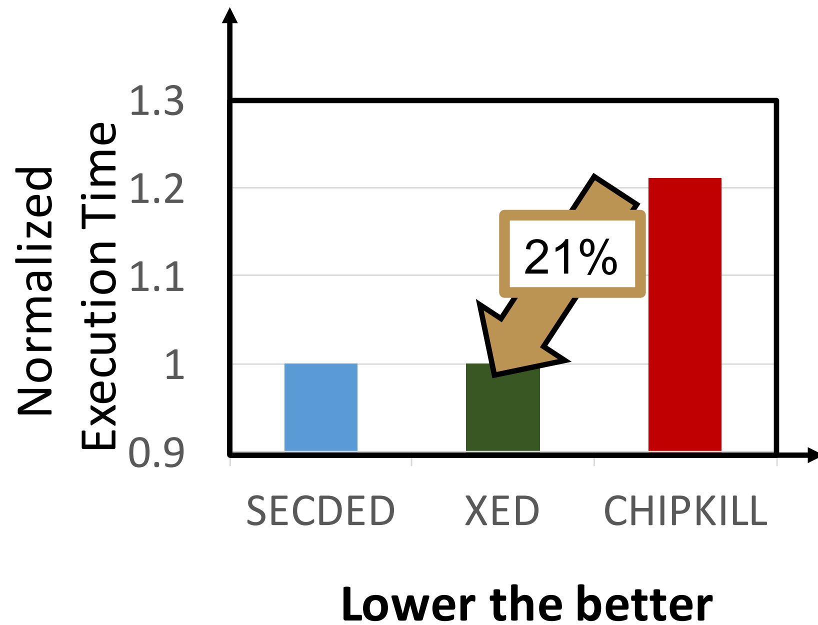
RESULTS: RELIABILITY

XED vs Commercial ECC schemes

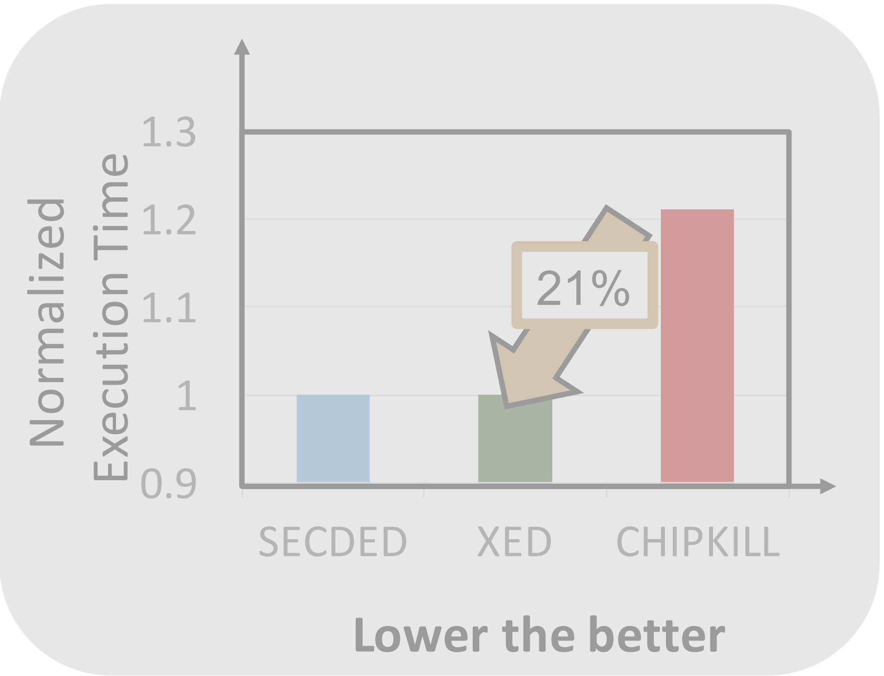


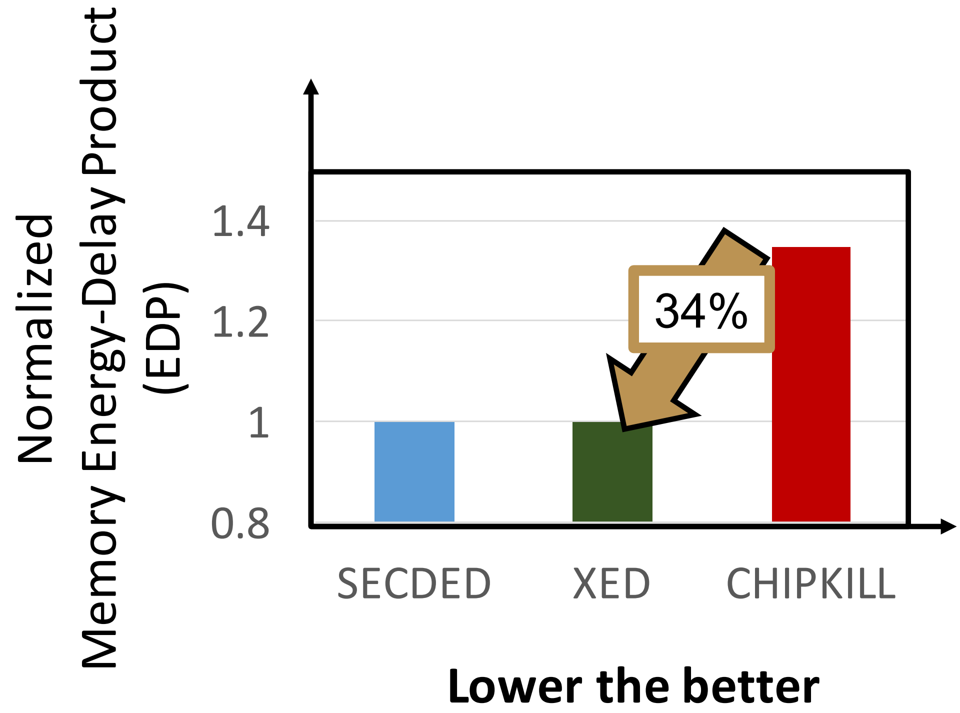
XED provides strong reliability while using fewer chips

RESULTS: PERFORMANCE AND EDP



RESULTS: PERFORMANCE AND EDP

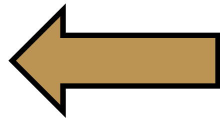




Execution time: 21% ↓, EDP : 34% ↓

OUTLINE

- BACKGROUND
- XED
- CASE STUDIES
- EVALUATION
- SUMMARY



SUMMARY

- DRAM Scaling introduces errors → On-Die ECC
- On-Die ECC is invisible to the memory system
- Exposing On-Die ECC: Efficient Runtime ECC
- XED
  - Exposes On-Die Error Detection using Catch-Words
  - 2X fewer chips as compared to Chipkill
  - 4X higher reliability as compared to Chipkill
  - 21% lower execution time as compared to Chipkill
- XED → No change in memory protocols

THANK YOU

"You are in a pitiable condition, if you have to conceal what you wish to tell"

- Publilius Syrus

BACKUP

RANDOM DATA?

- What if only half the data is random
  1. Then average time for collision increases by 2x
     (3.2 Million Years → 6.4 Million Years)
  2. Less random data increases collision time

- DIMMs today store scrambled (randomized) data
  1. To equalize the number of 1's and 0's
  2. Reduce Bit Error Rate on the bus
  3. Scrambling using address based hash

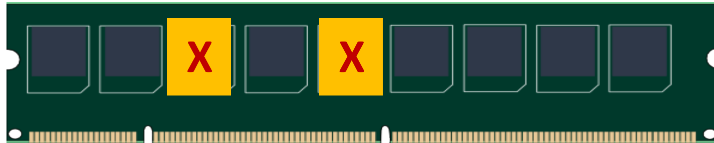
1. Lower randomization → Longer time till collision
2. Current systems anyway scramble data for fidelity

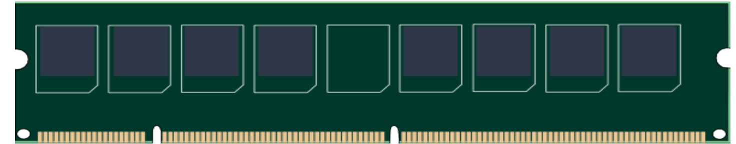
MTTF: XED VS CHIPKILL

2-Chip Failures

XED (9-chips)

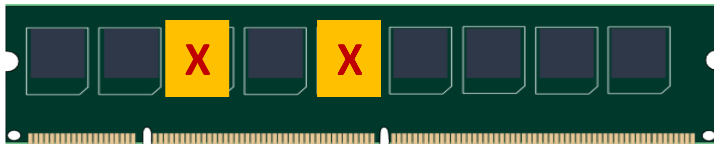
FAILED

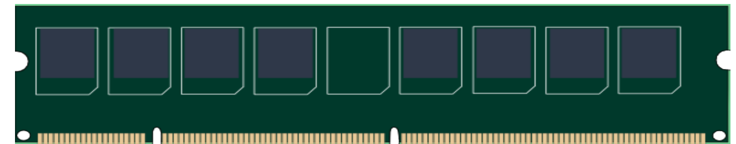




Chipkill (18-chips)

FAILED



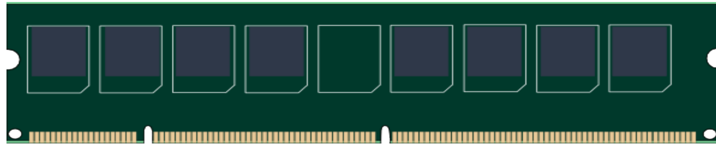


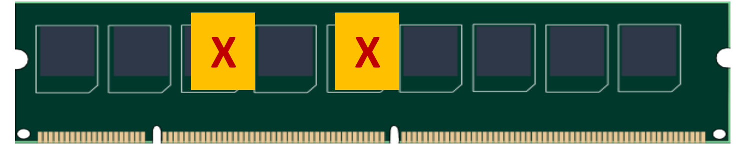
MTTF: XED VS CHIPKILL

2-Chip Failures

XED (9-chips)

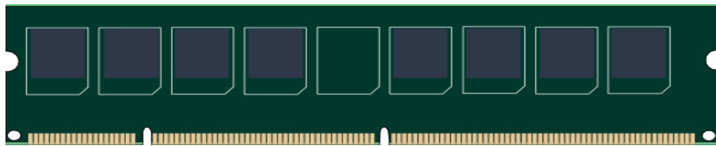
FAILED

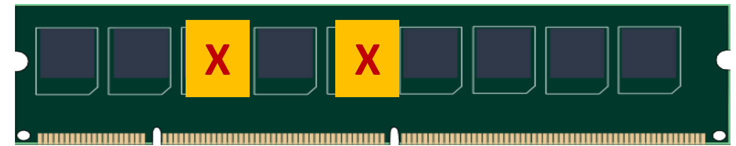




Chipkill

FAILED



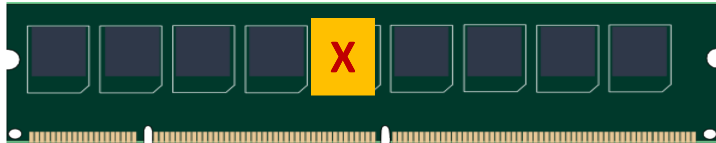


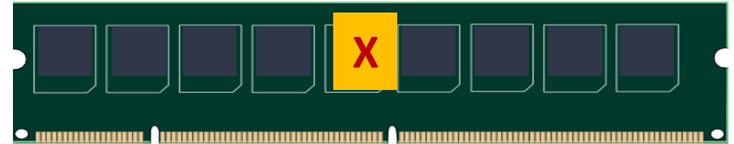
MTTF: XED VS CHIPKILL

2-Chip Failures → Extend to Multi-Chip Failures

XED (9-chips)

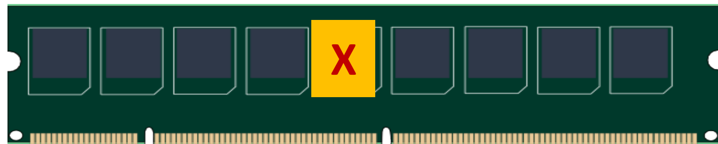
PASSED

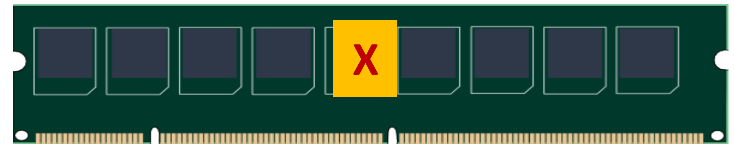




Chipkill

FAILED





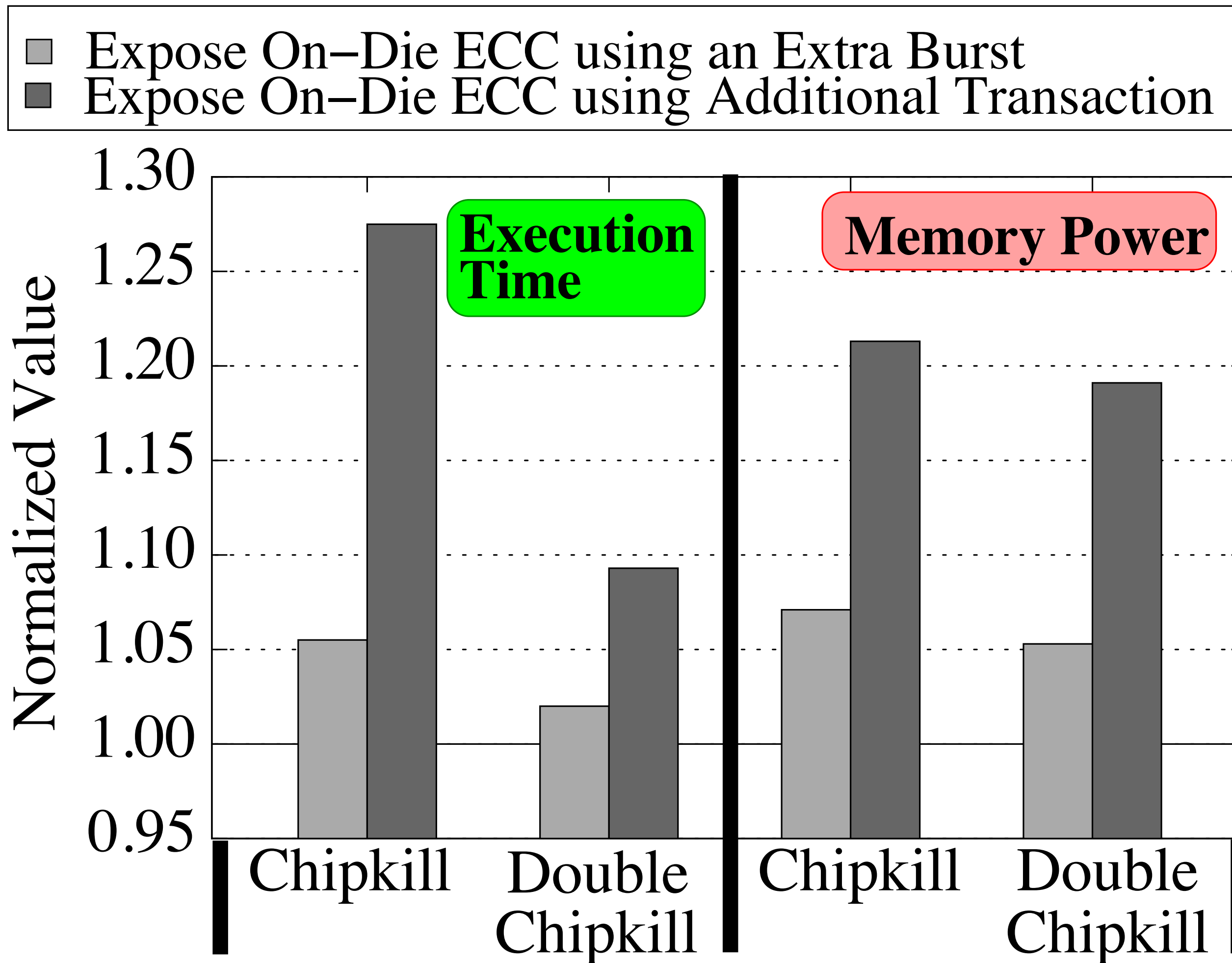
SDC AND DUE

SDC and DUE Rate of XED

Source of Vulnerability	Rate over 7 years
XED: Scaling-Related Faults	**No SDC or DUE**
XED: Row/ Column/ Bank Failure	1.4×10^{-13} (**SDC**)
XED: Word Failure	6.1×10^{-6} (**DUE**)

Data Loss from Multi-Chip Failures	5.8×10^{-4}

ADDITIONAL BURST/TRANSACTION



XED VS LOT-ECC