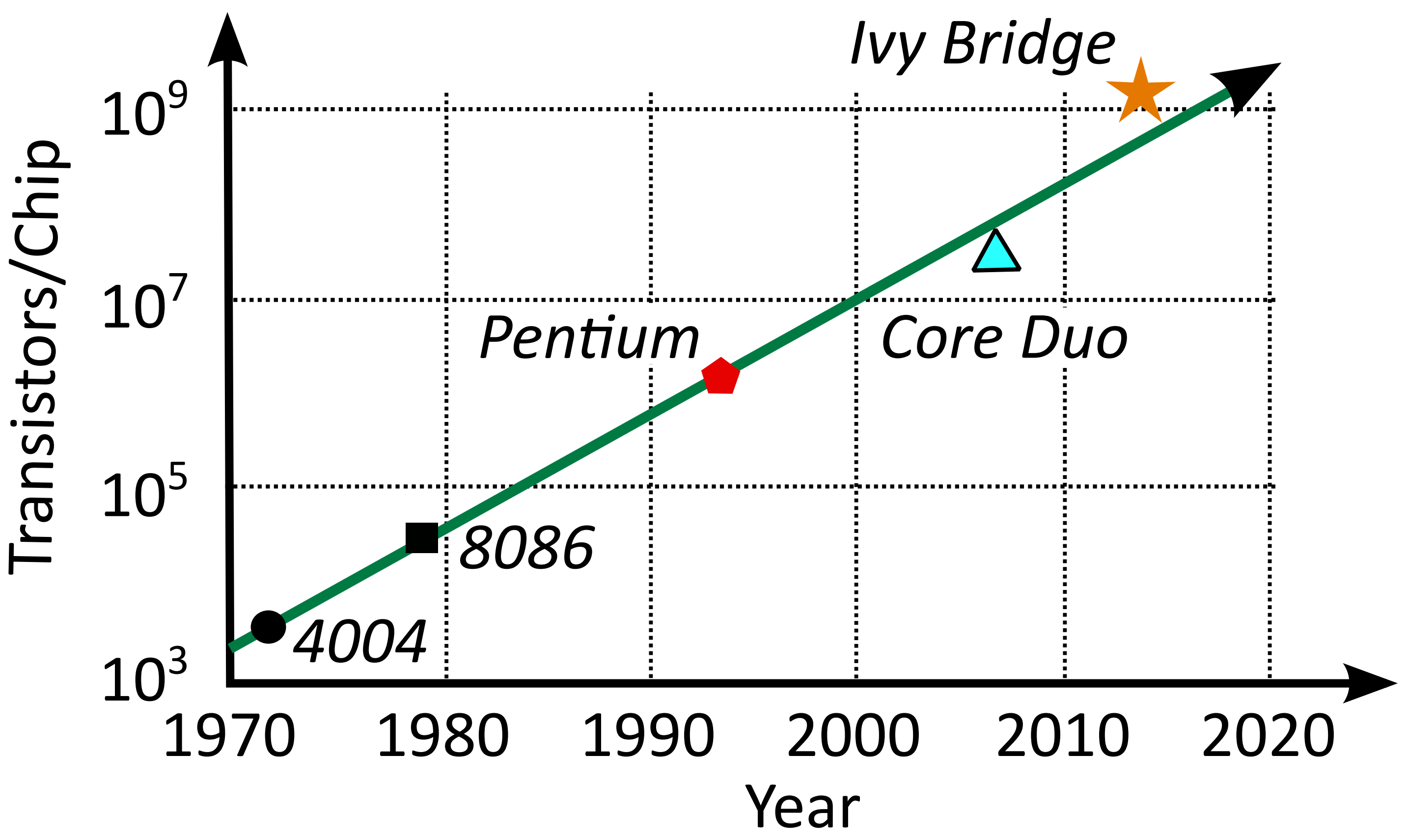
Learning to Live with Errors

Architectural Solutions for Memory Reliability at Extreme Scaling

Prashant J. Nair

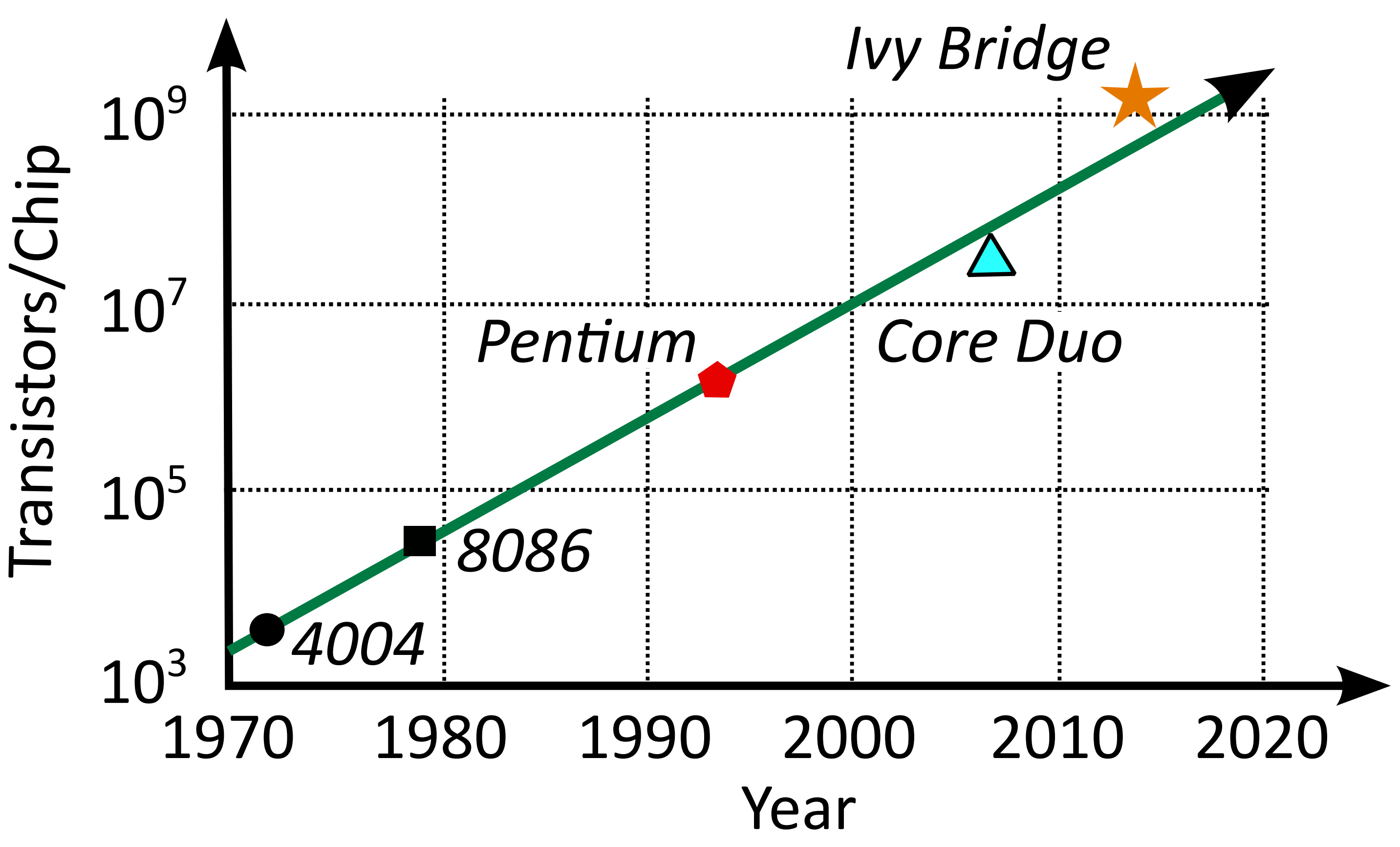
MOORE'S LAW IN COMPUTING

Technology Scaling → 2x transistors every ~2 years



MOORE'S LAW IN COMPUTING

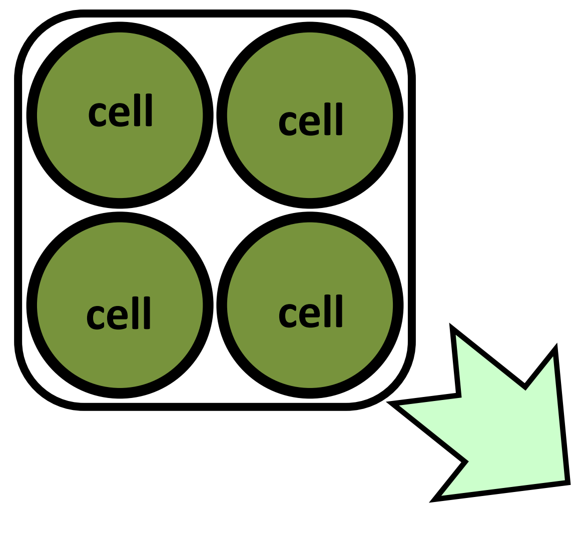
Technology Scaling → 2x transistors every ~2 years



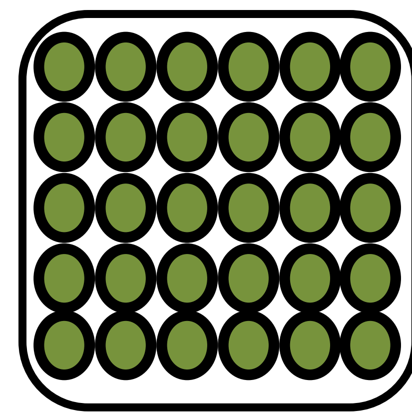
Improving the Performance of Computing Systems

MOORE'S LAW IN MEMORY SYSTEMS

Technology Scaling: A key driver for applications



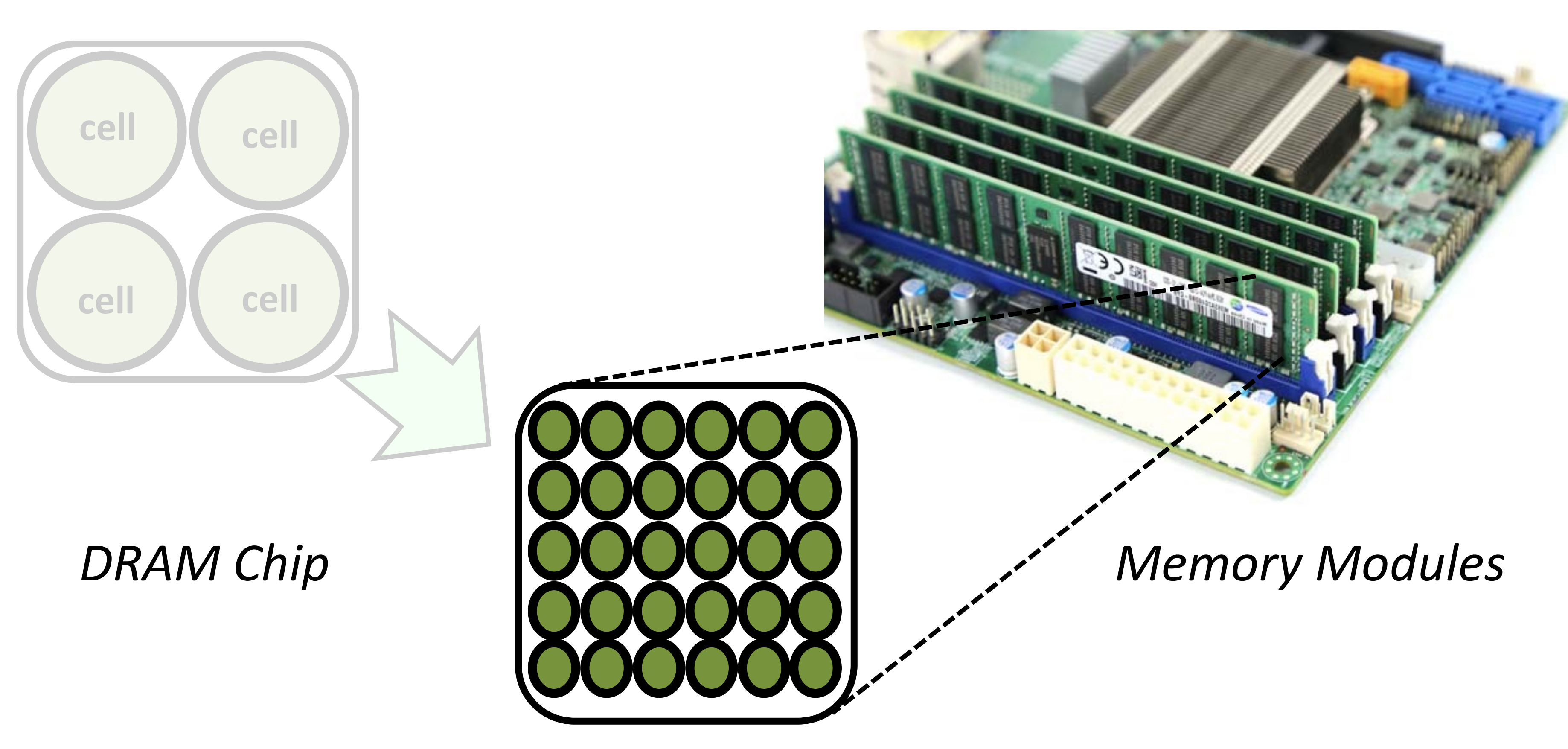
Technology Scaling



DRAM Chip

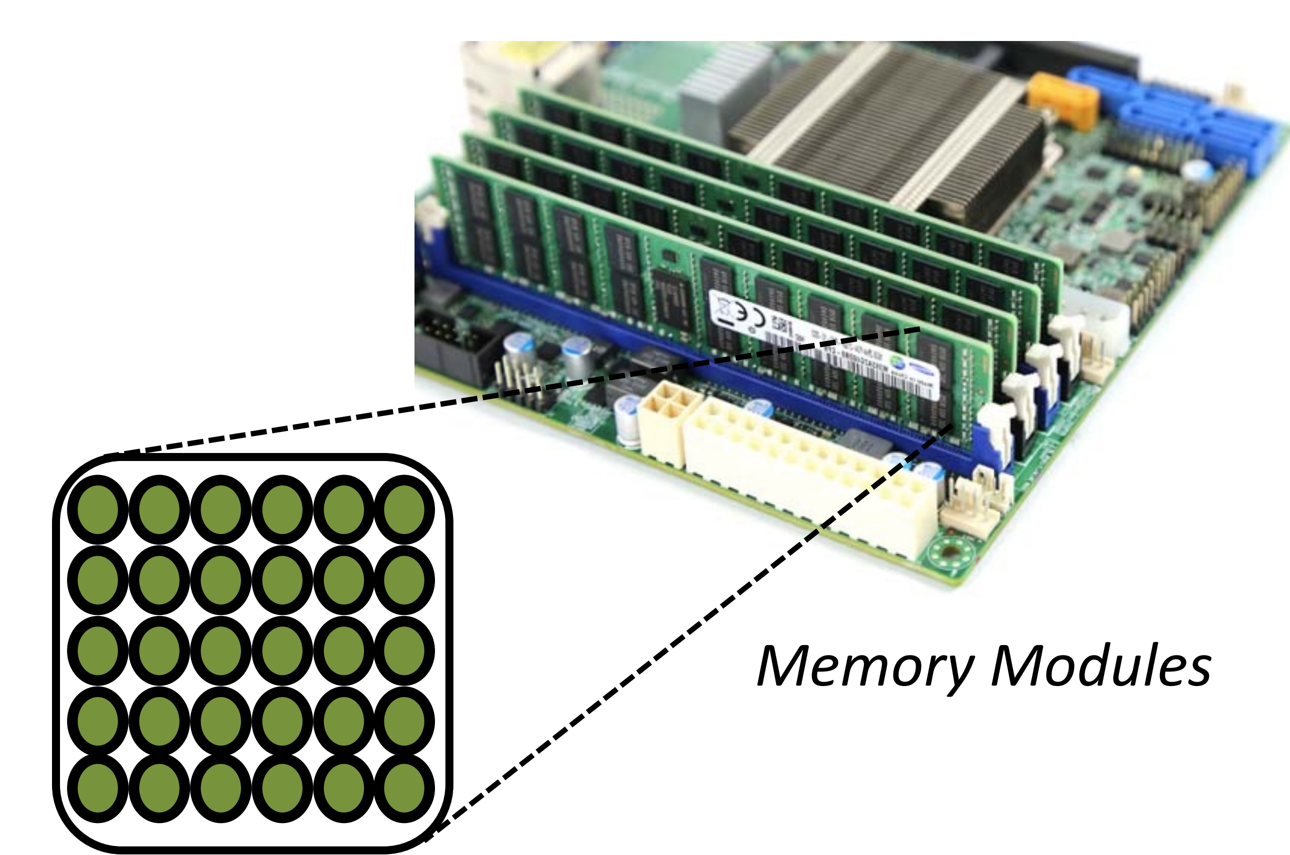
MOORE'S LAW IN MEMORY SYSTEMS

Technology Scaling: A key driver for applications



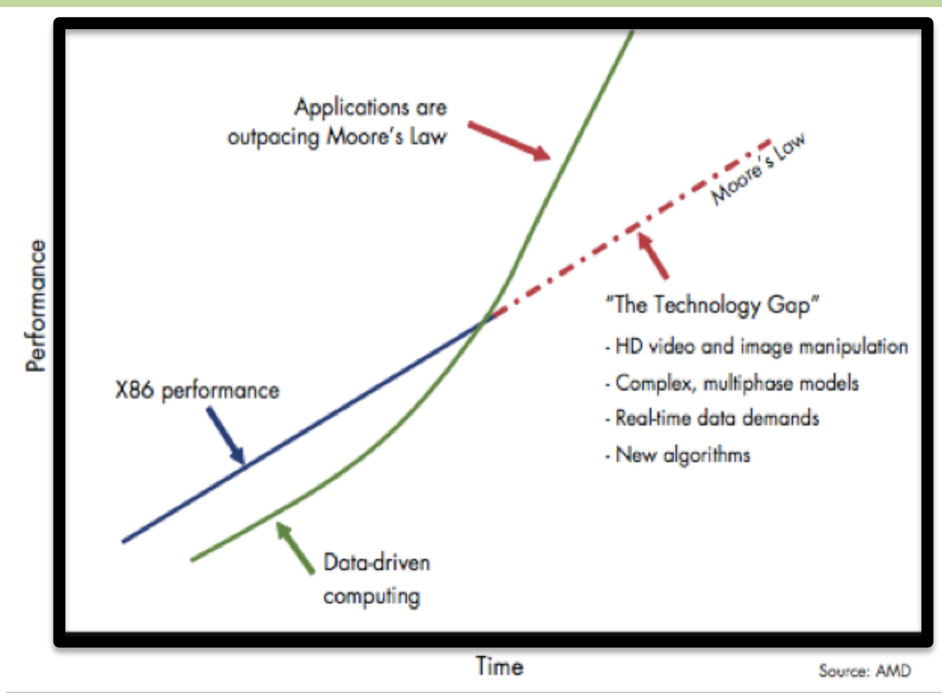
MOORE'S LAW IN MEMORY SYSTEMS

Technology Scaling: A key driver for applications

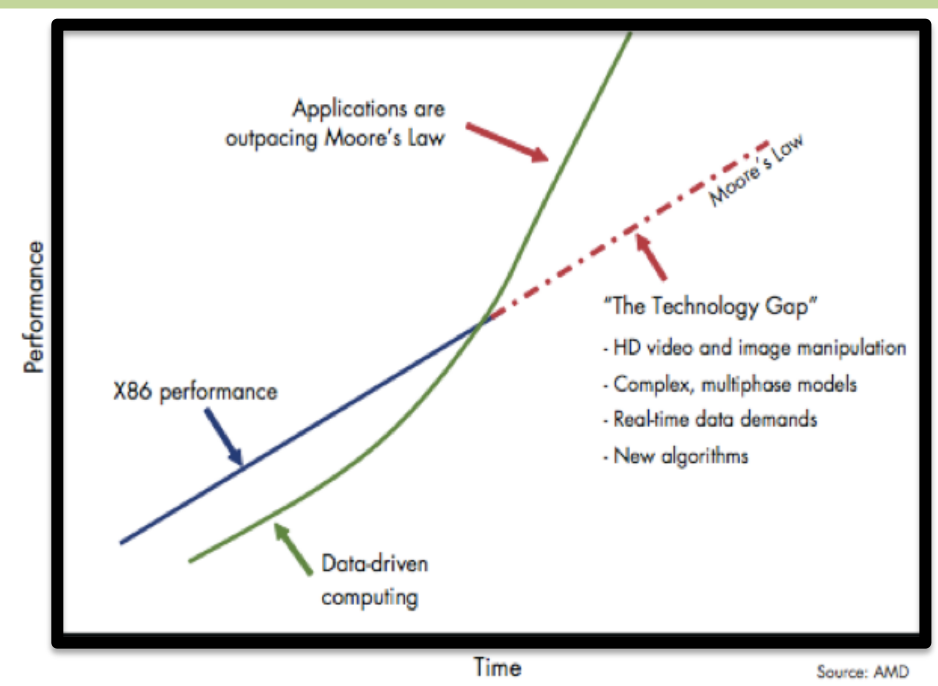


Moore's Law is vital for High-Density Memories

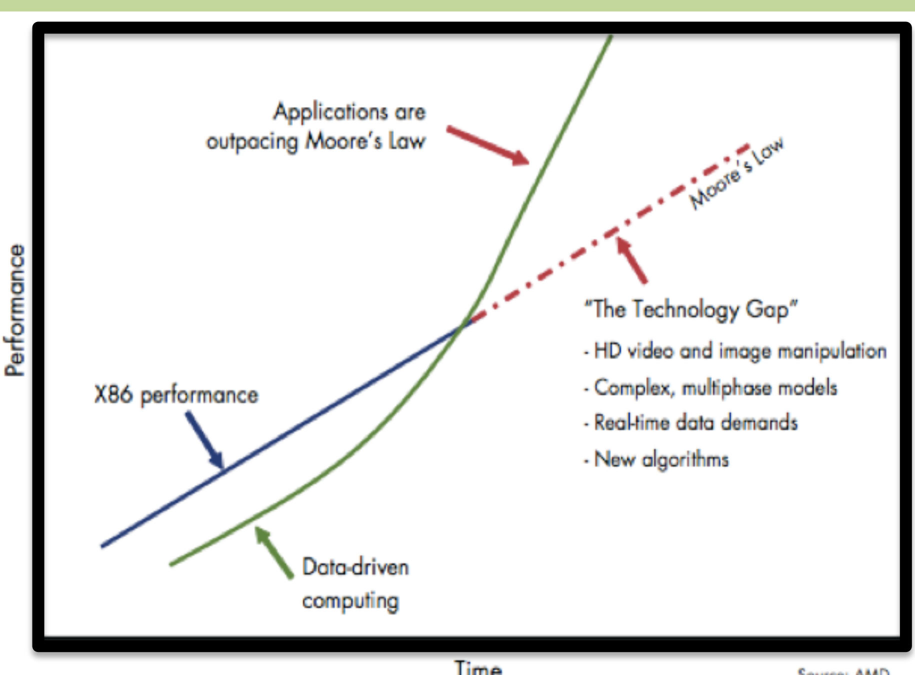
MEMORY SCALING: VITAL FOR ALL DOMAINS

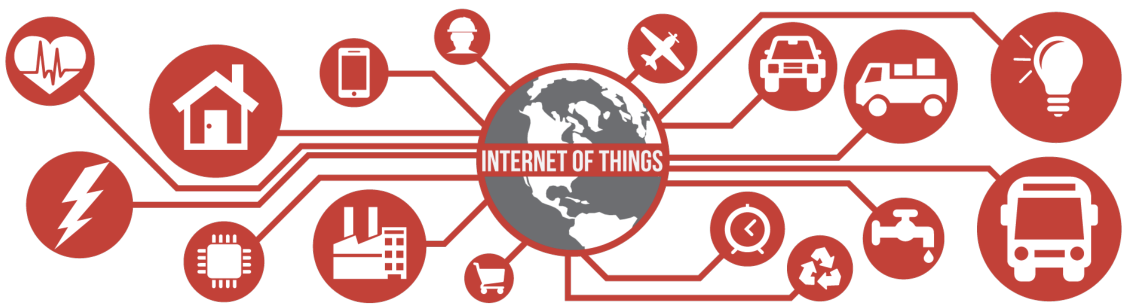


MEMORY SCALING: VITAL FOR ALL DOMAINS

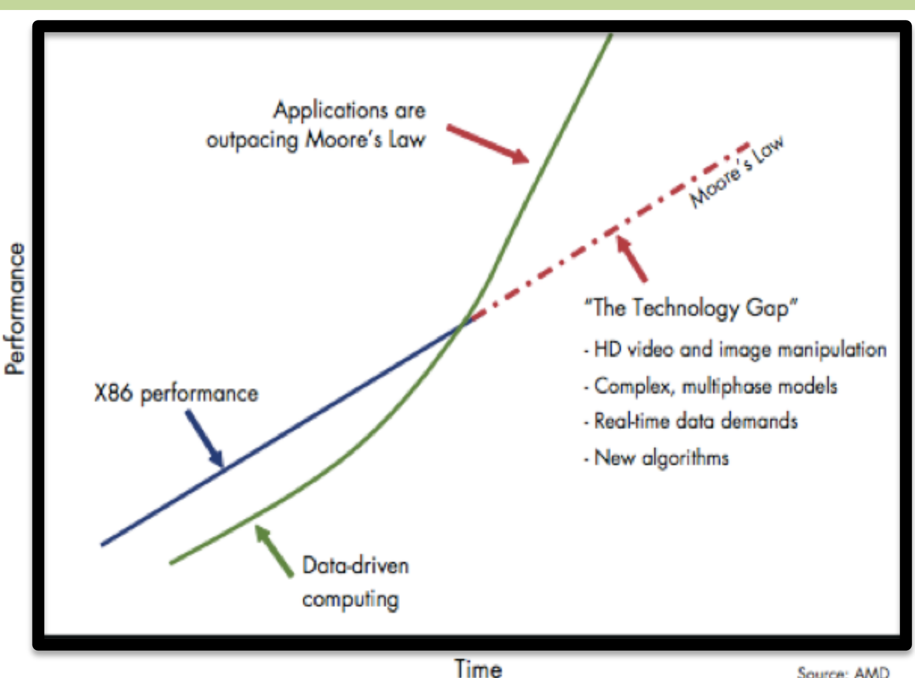


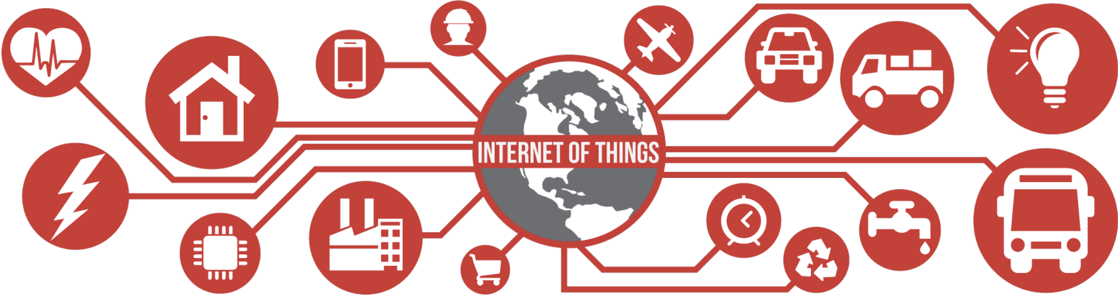
MEMORY SCALING: VITAL FOR ALL DOMAINS





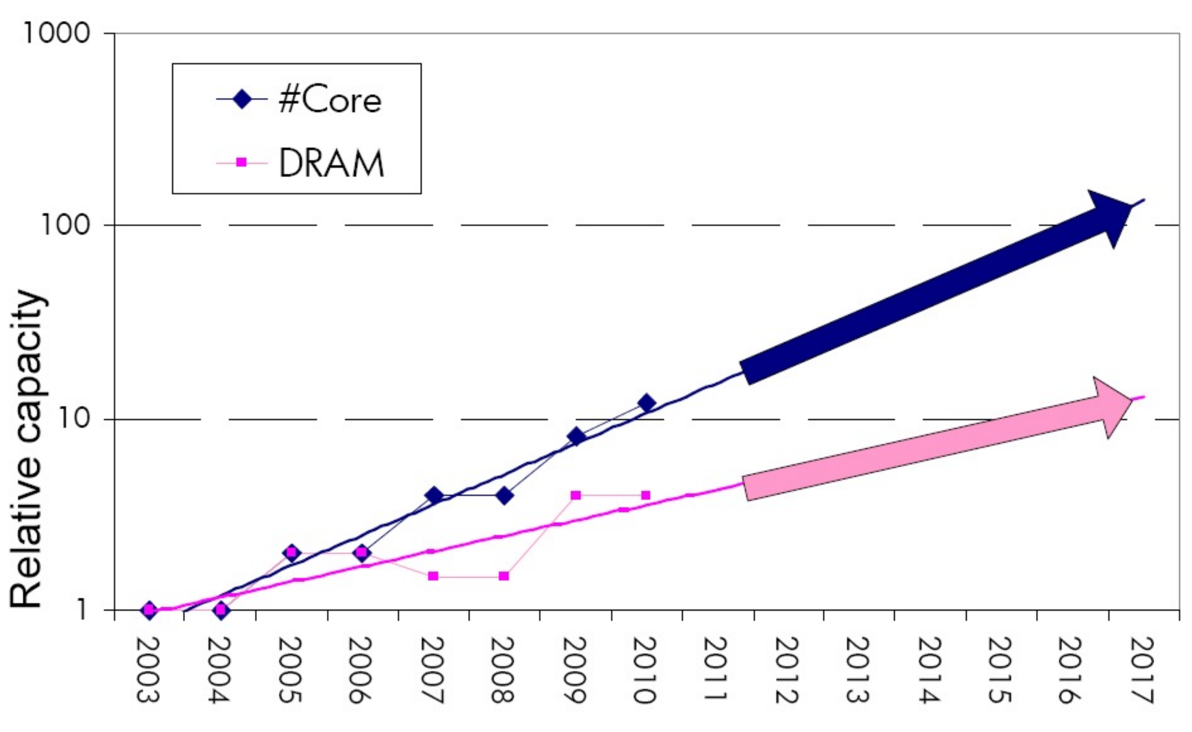
MEMORY SCALING: VITAL FOR ALL DOMAINS



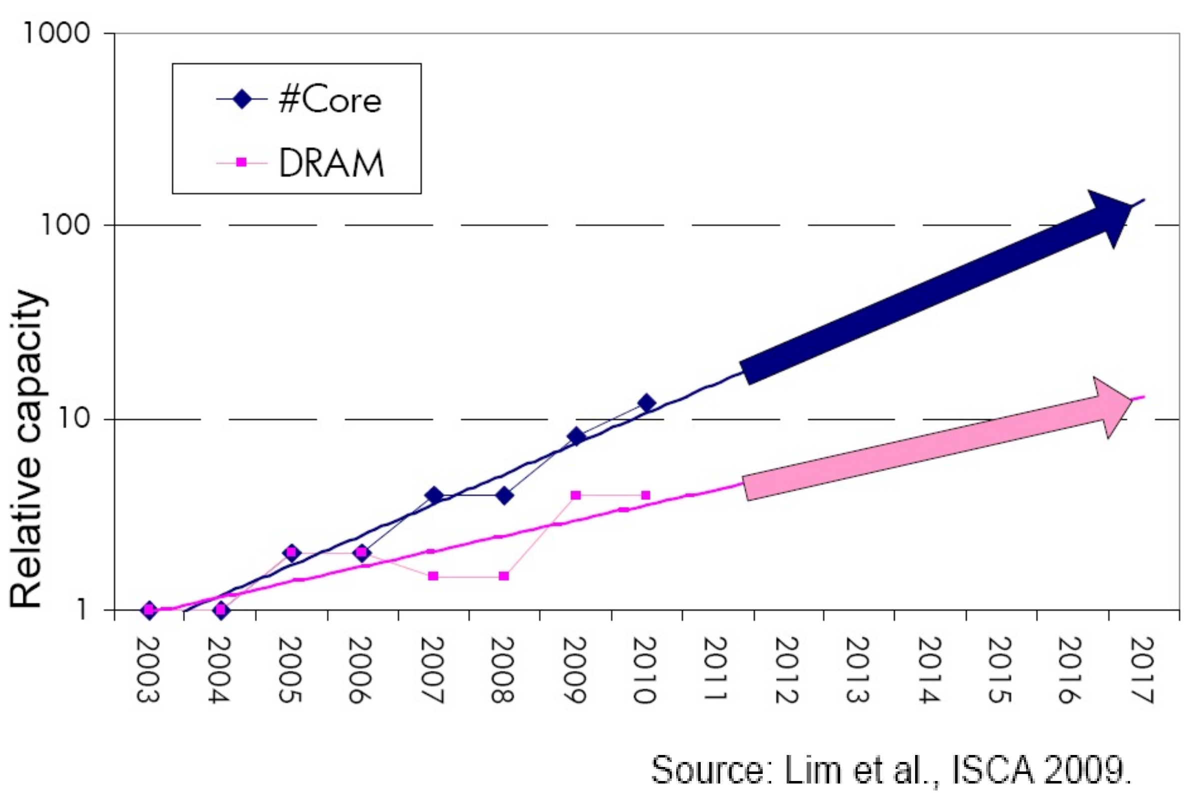


Client, Server and IoT devices → Scalable Memories

CHALLENGES IN MEMORY SCALING

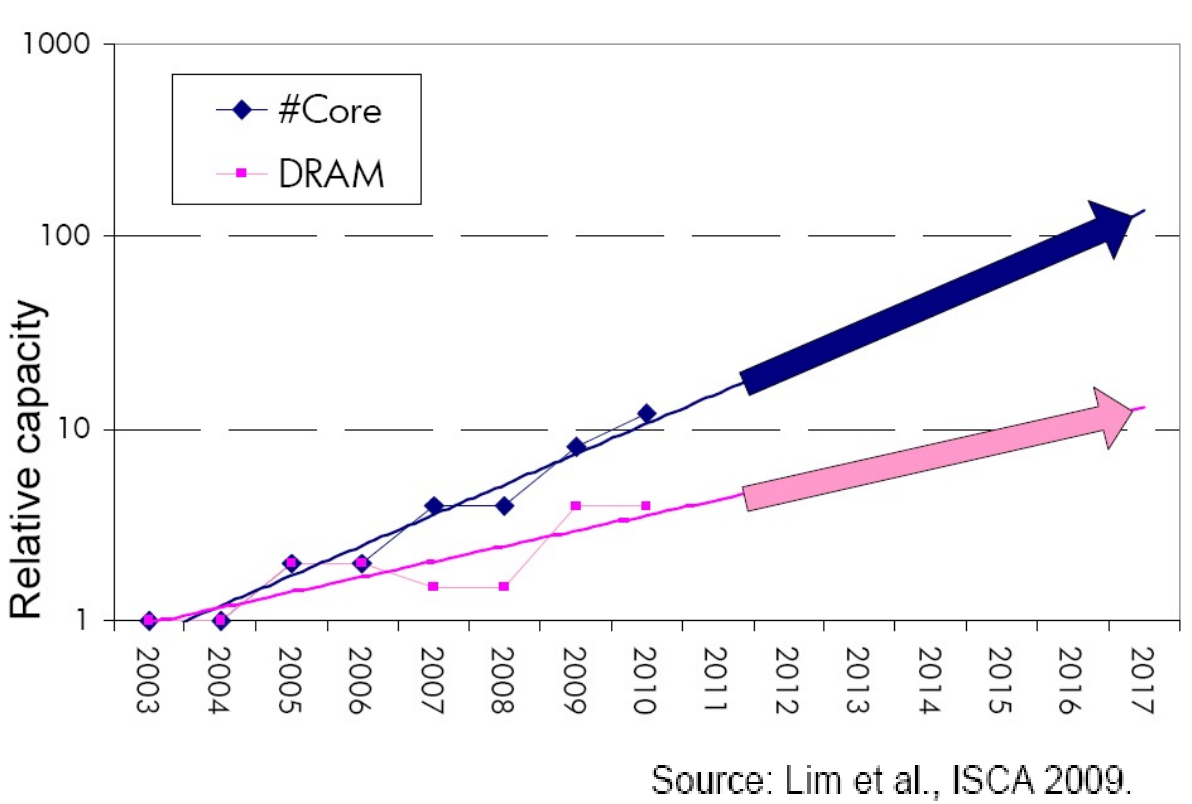


CHALLENGES IN MEMORY SCALING



Per-core DRAM capacity reduces by 30% every 2 years

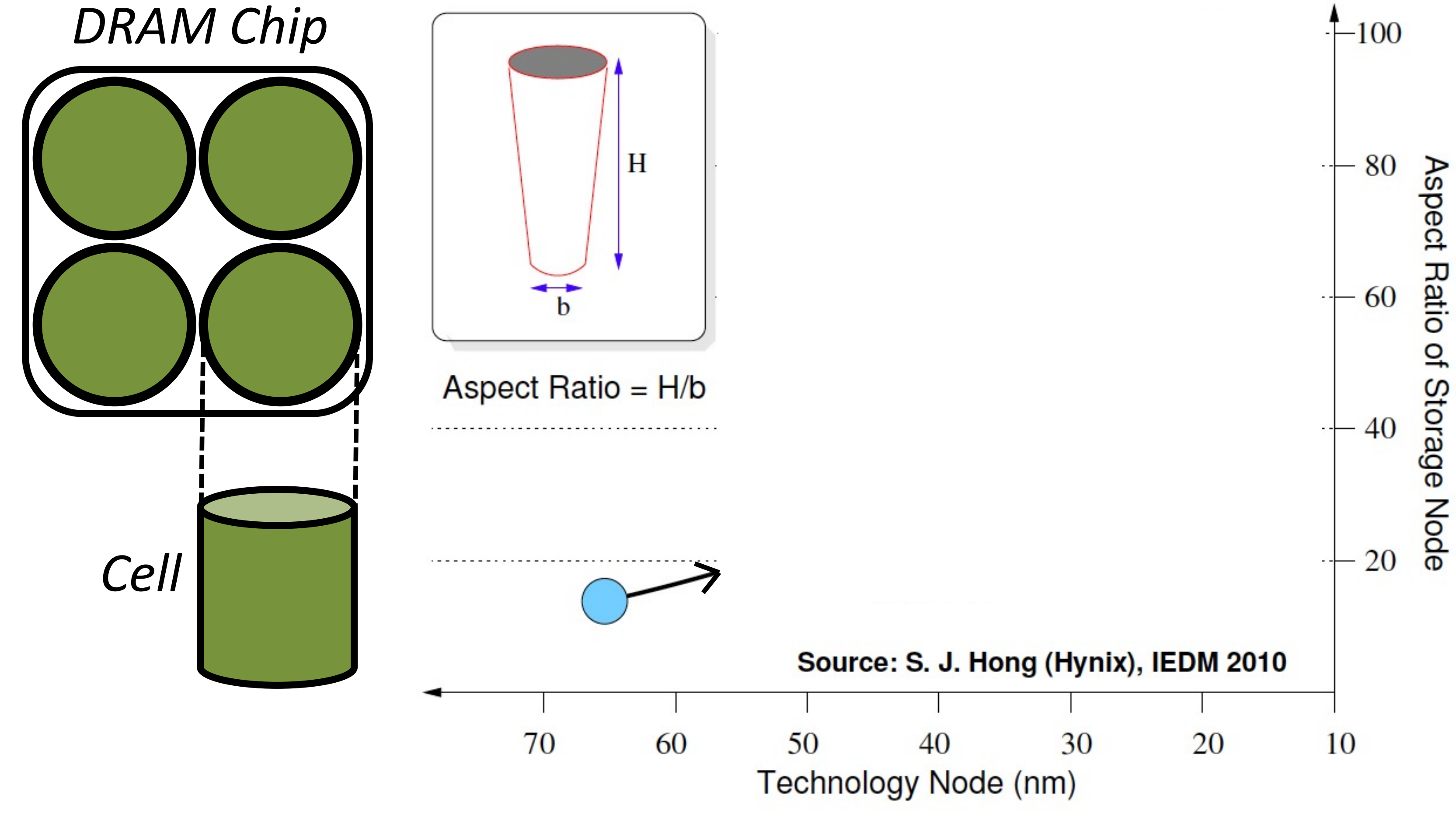
CHALLENGES IN MEMORY SCALING



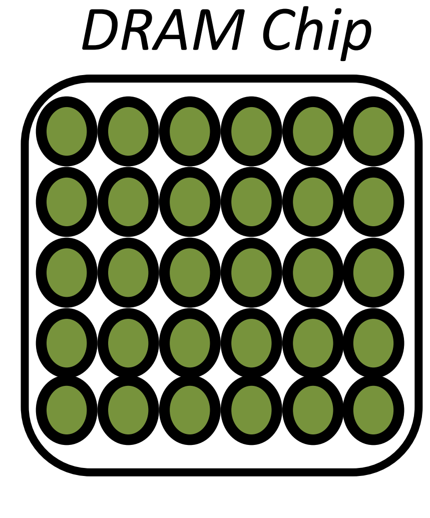
Per-core DRAM capacity reduces by 30% every 2 years

Moore's Law in Memory Systems → Scaling Wall

WHY IS DRAM SCALING DIFFICULT?



WHY IS DRAM SCALING DIFFICULT?



H
b
Aspect Ratio = H/b

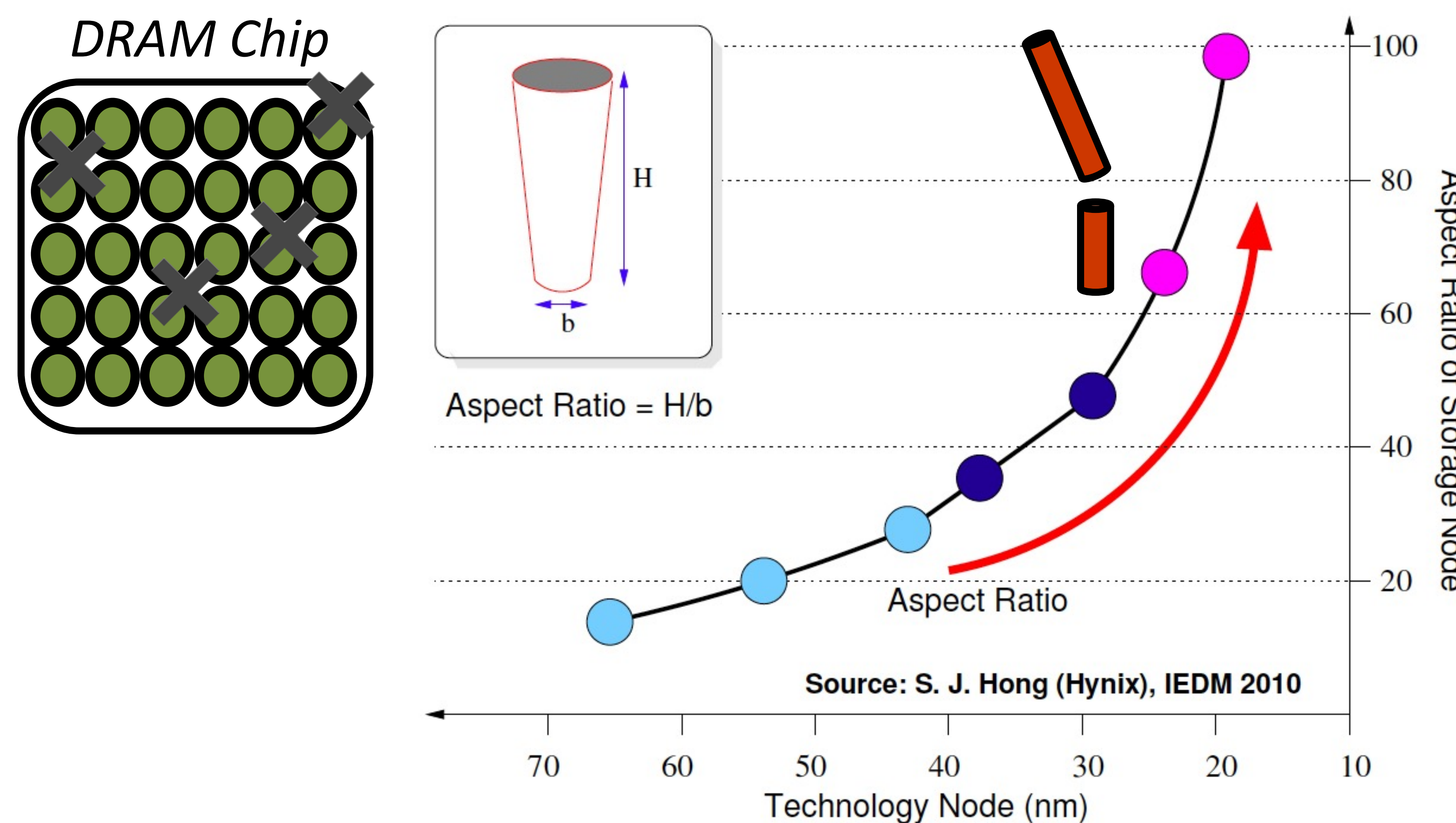
100
80
60
40
20
Aspect Ratio of Storage Node

Aspect Ratio

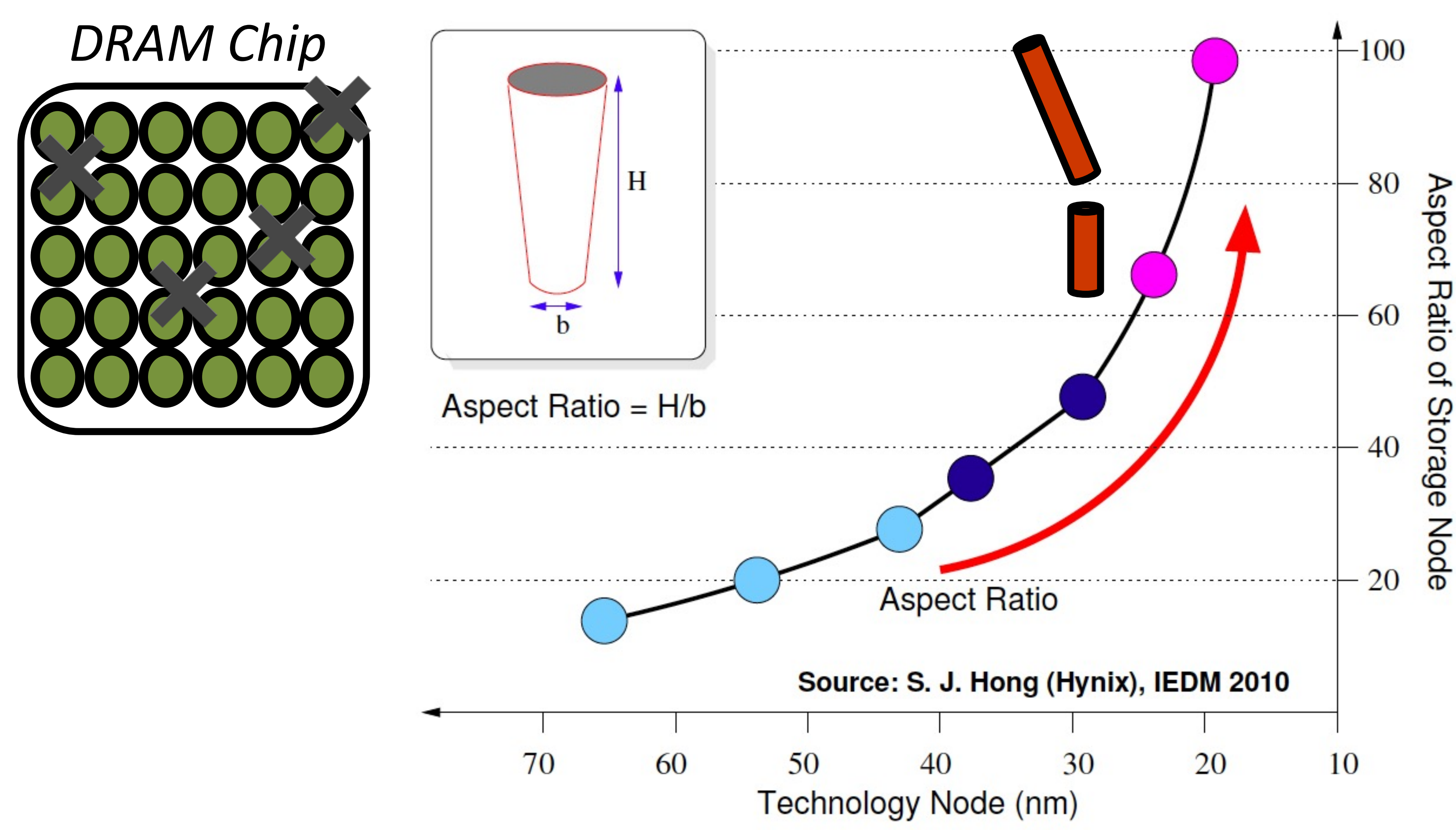
Source: S. J. Hong (Hynix), IEDM 2010

70 60 50 40 30 20 10
Technology Node (nm)

WHY IS DRAM SCALING DIFFICULT?



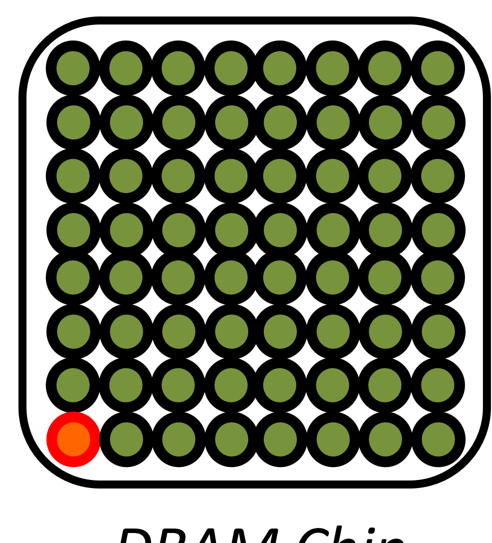
WHY IS DRAM SCALING DIFFICULT?



High aspect ratio → Faulty cells (Scaling Faults)

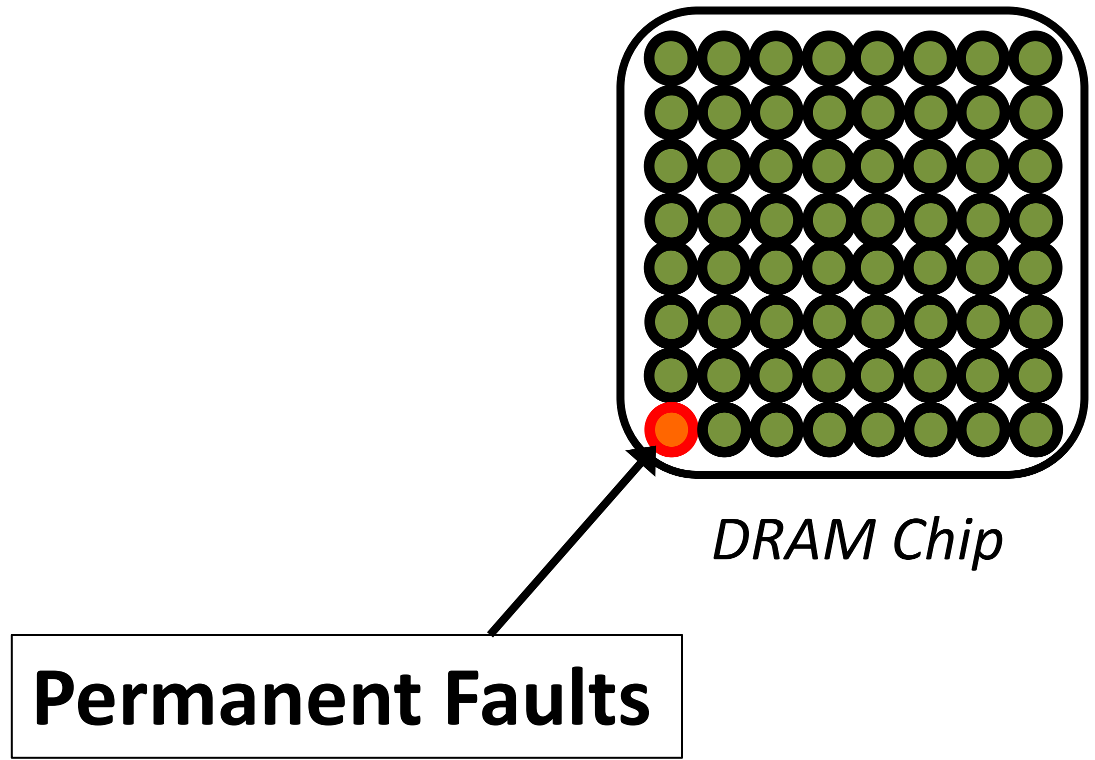
RUNTIME FAULTS

Faults that happen while the machine is operating

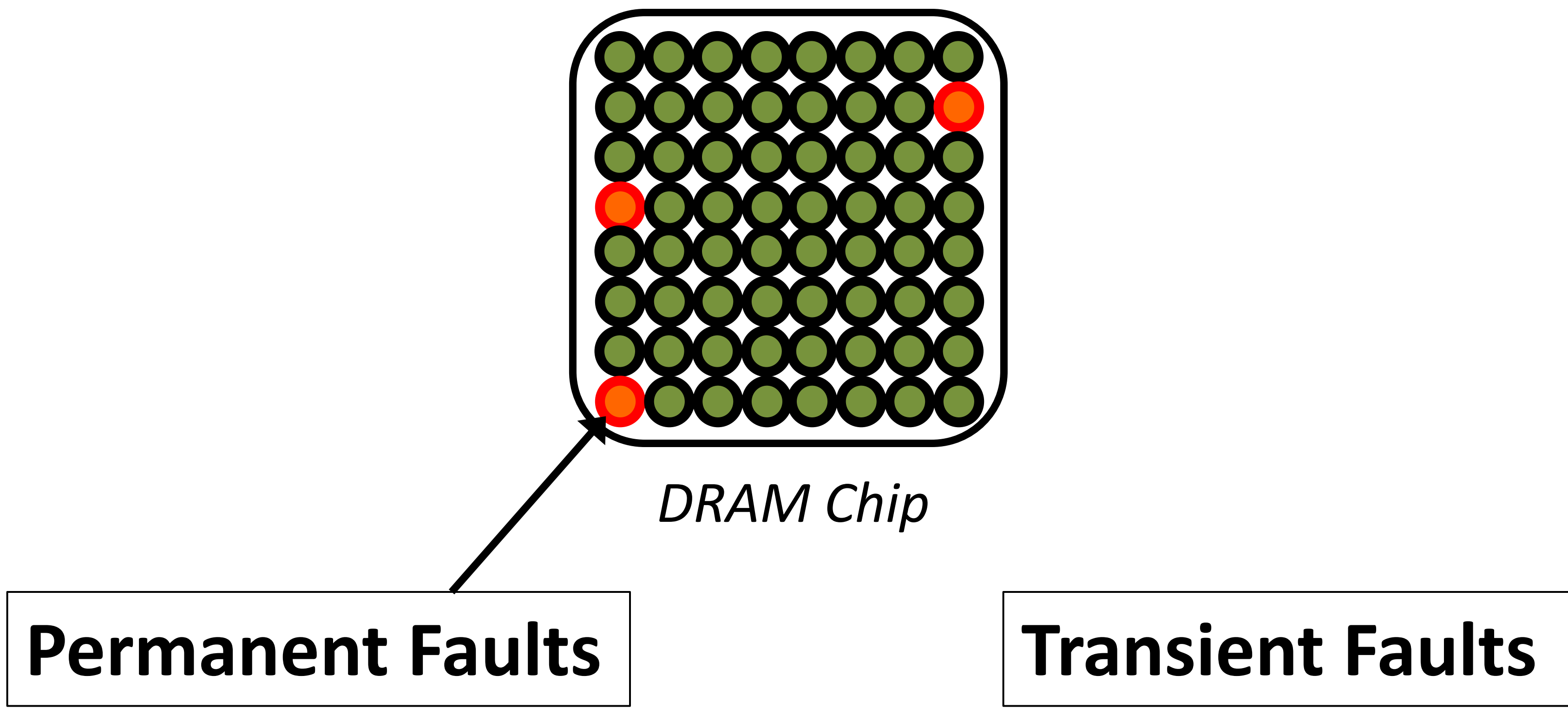


DRAM Chip

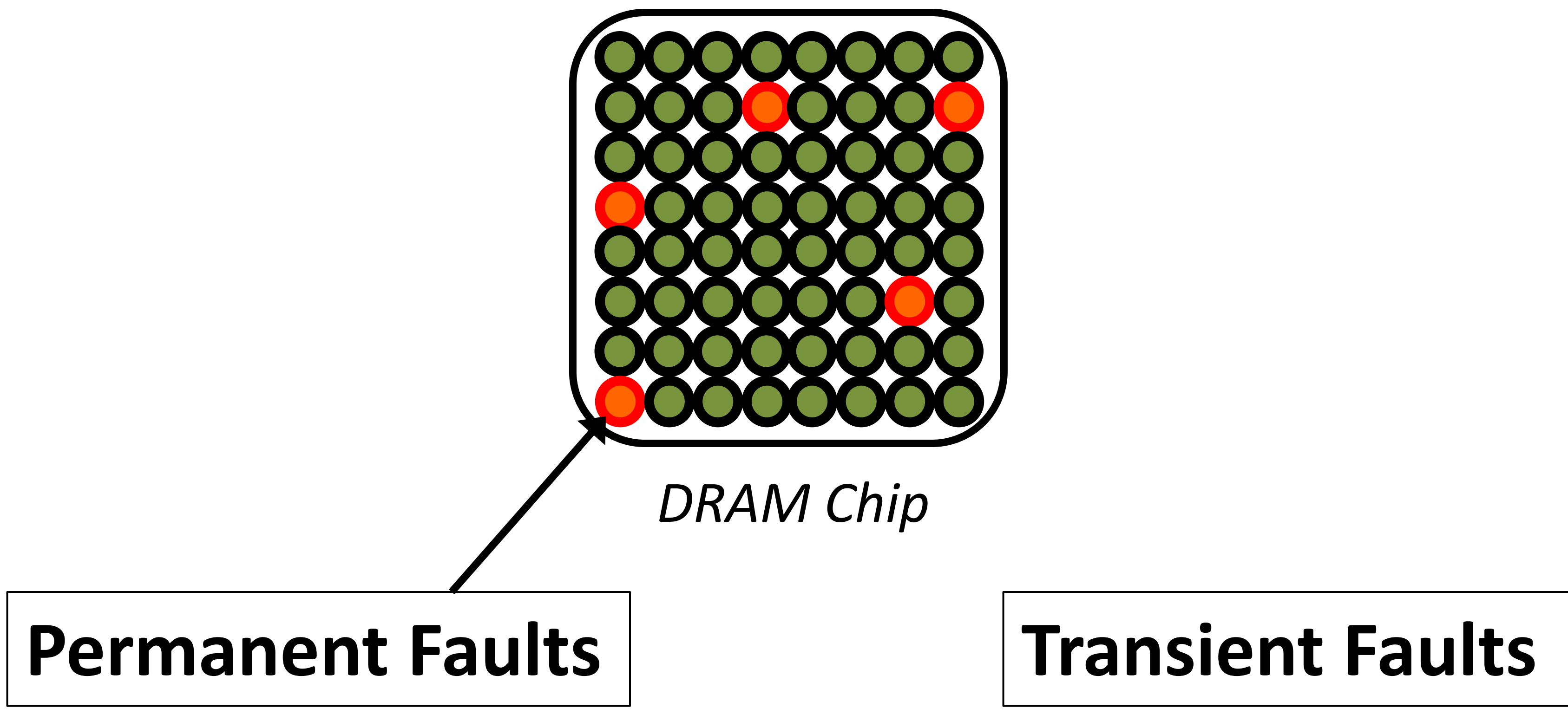
TYPES OF RUNTIME FAULTS



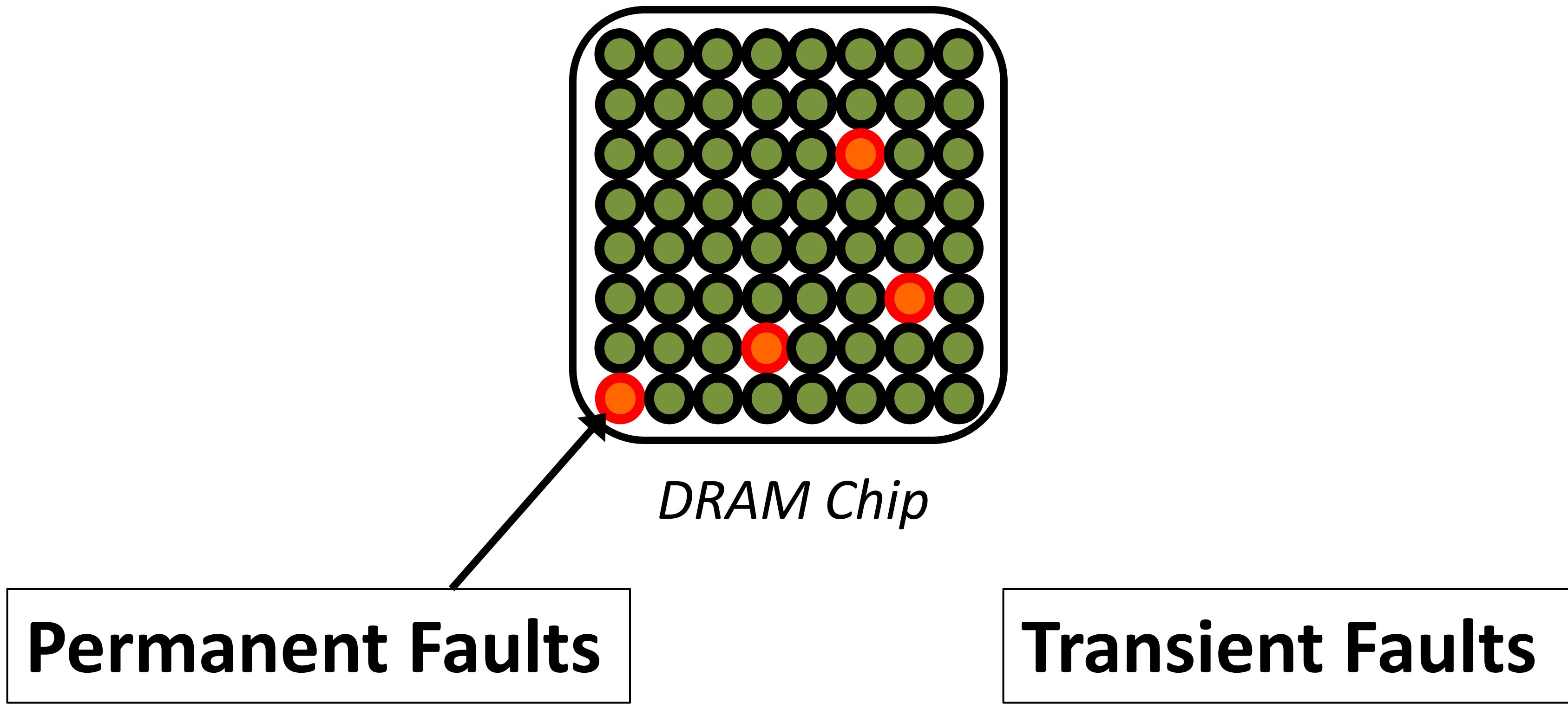
TYPES OF RUNTIME FAULTS



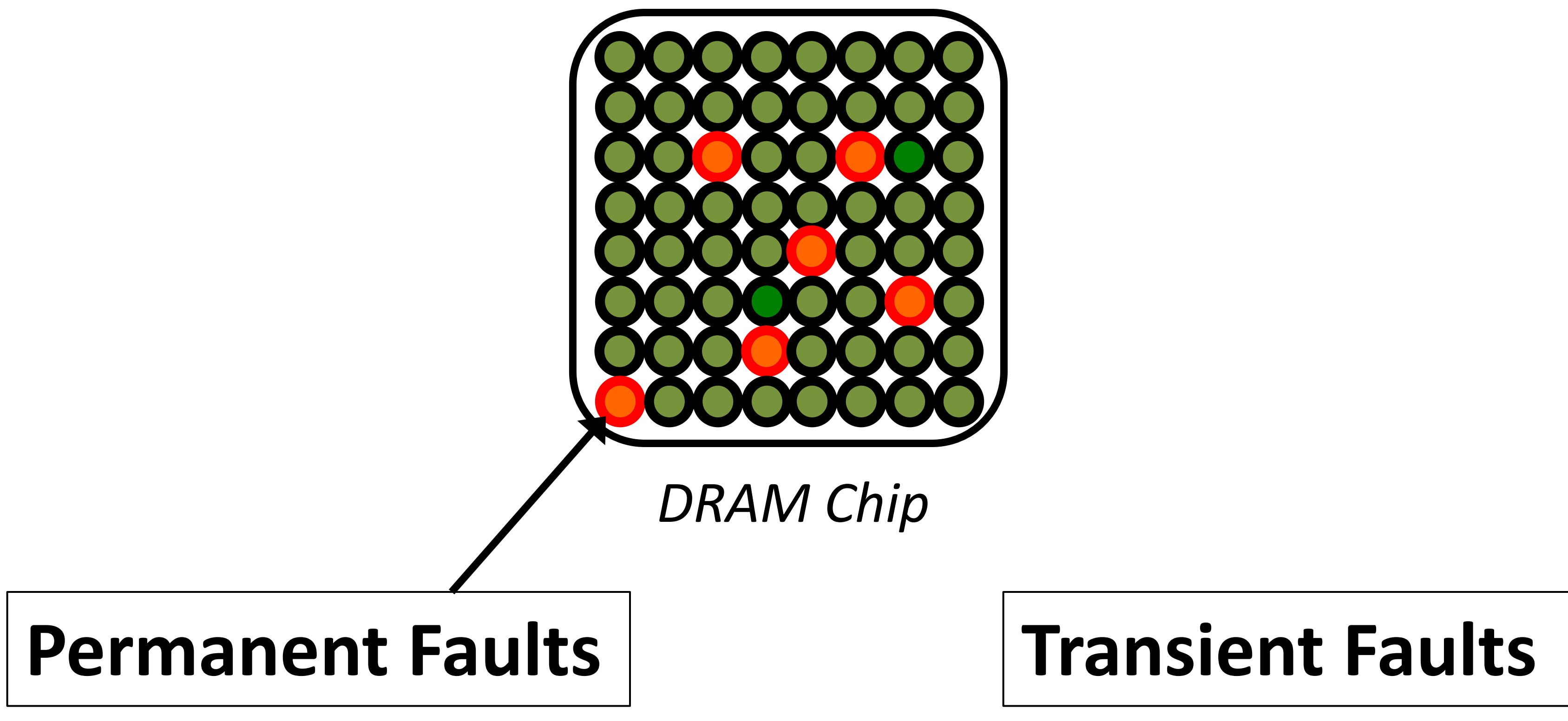
TYPES OF RUNTIME FAULTS



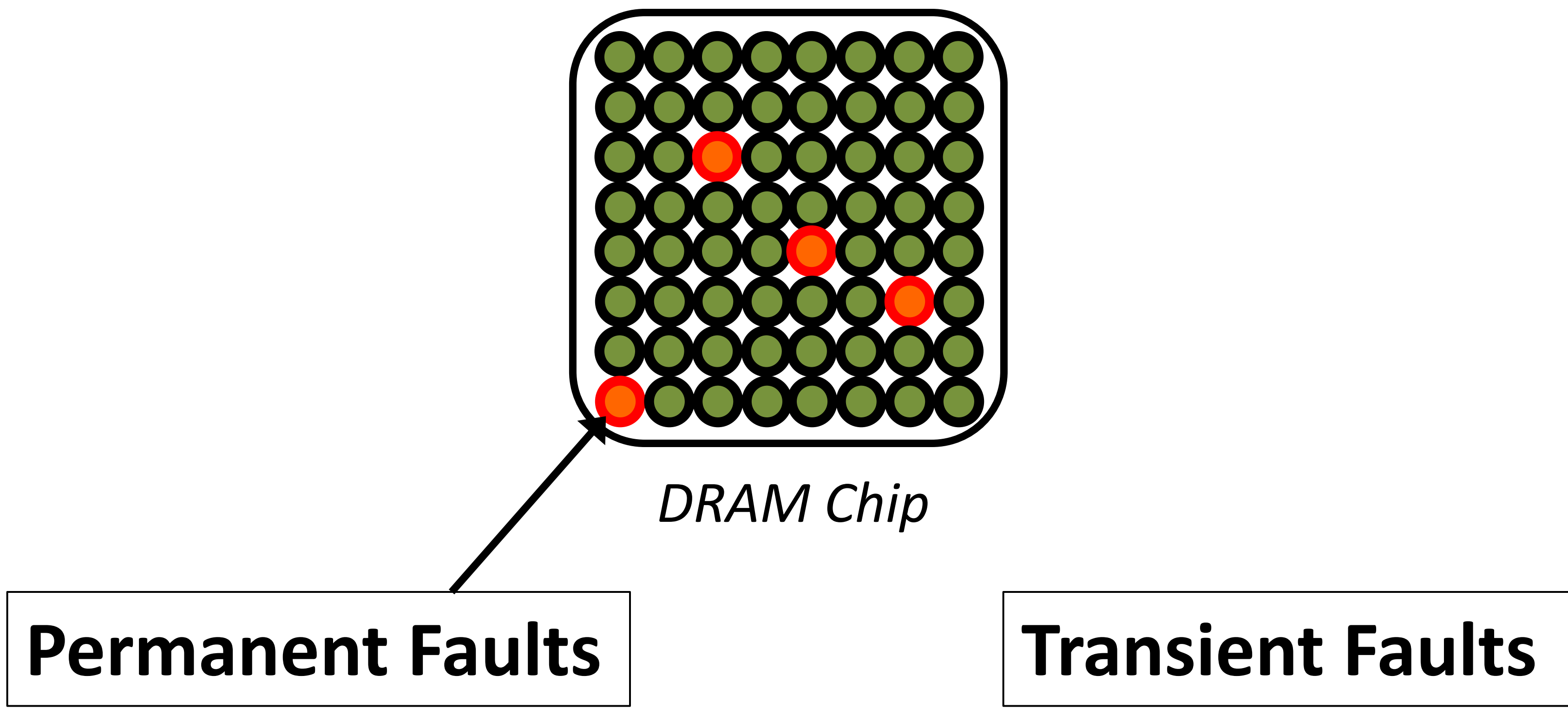
TYPES OF RUNTIME FAULTS



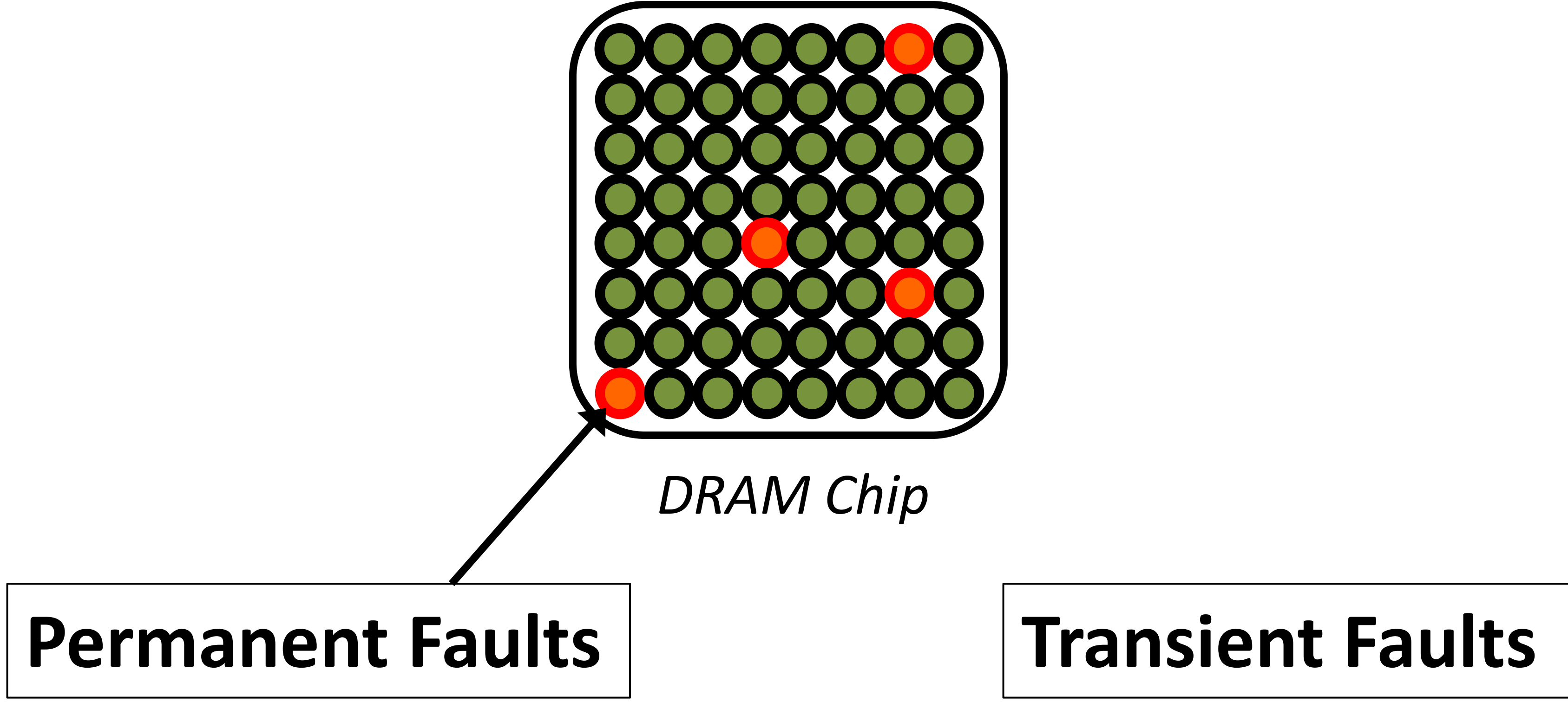
TYPES OF RUNTIME FAULTS



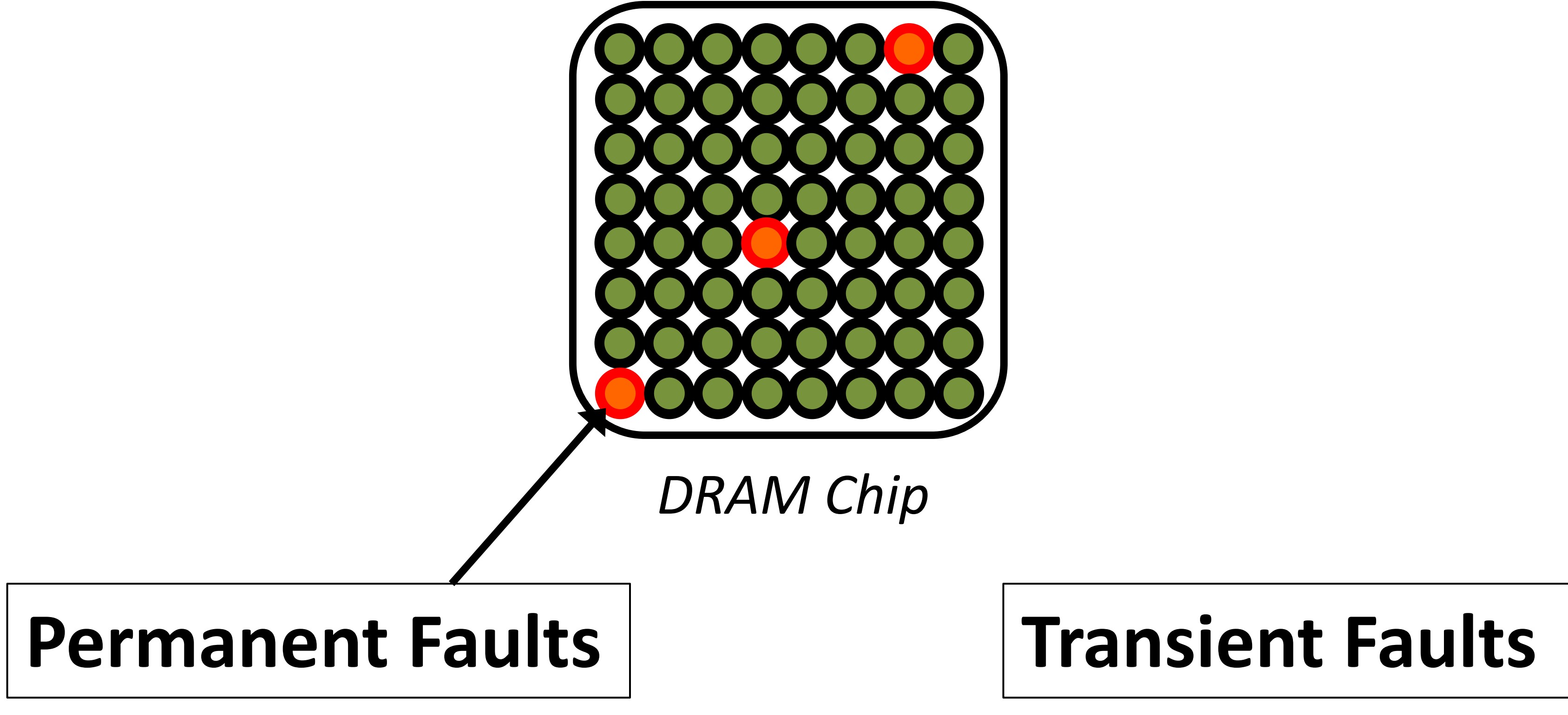
TYPES OF RUNTIME FAULTS



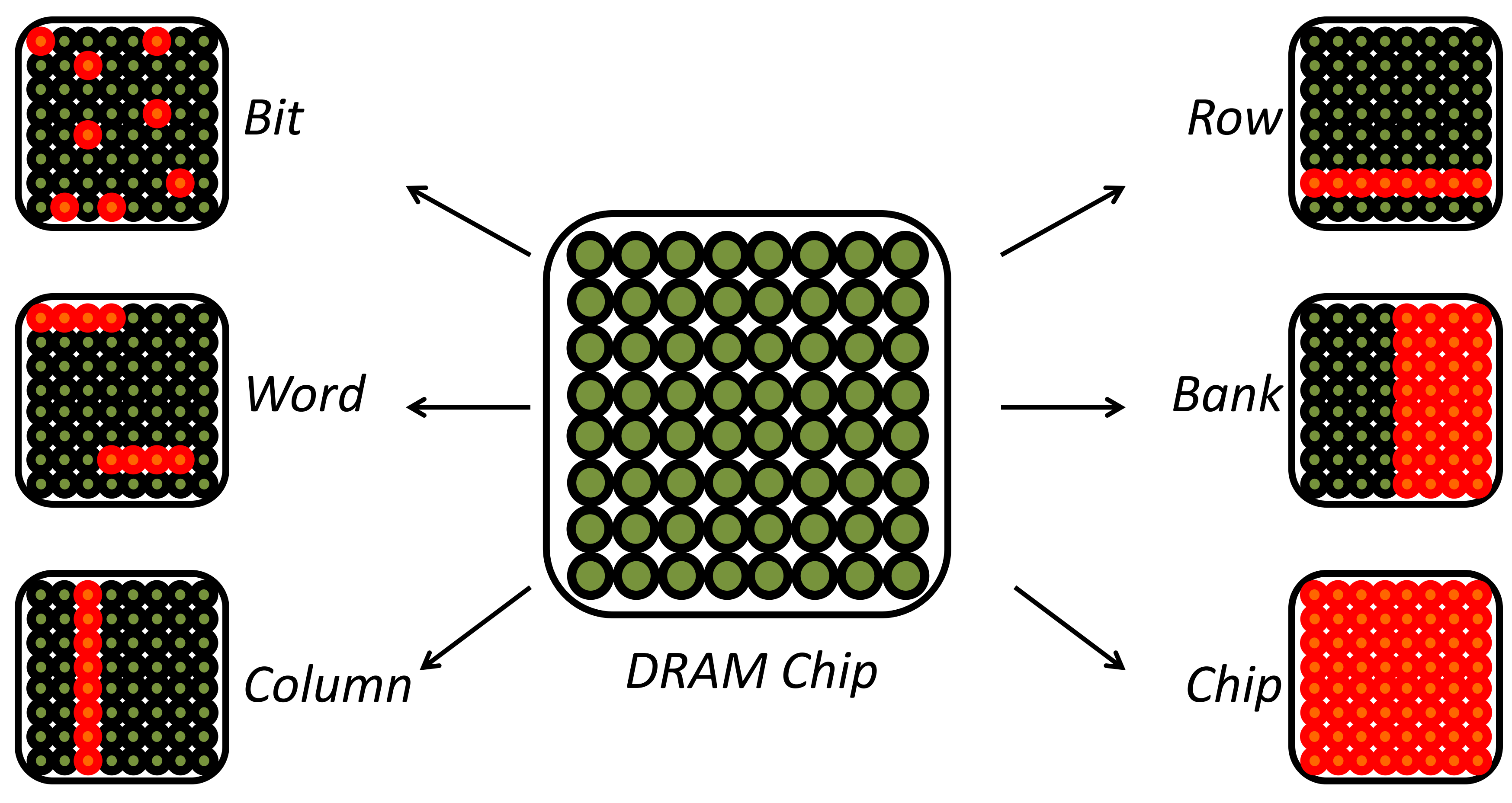
TYPES OF RUNTIME FAULTS



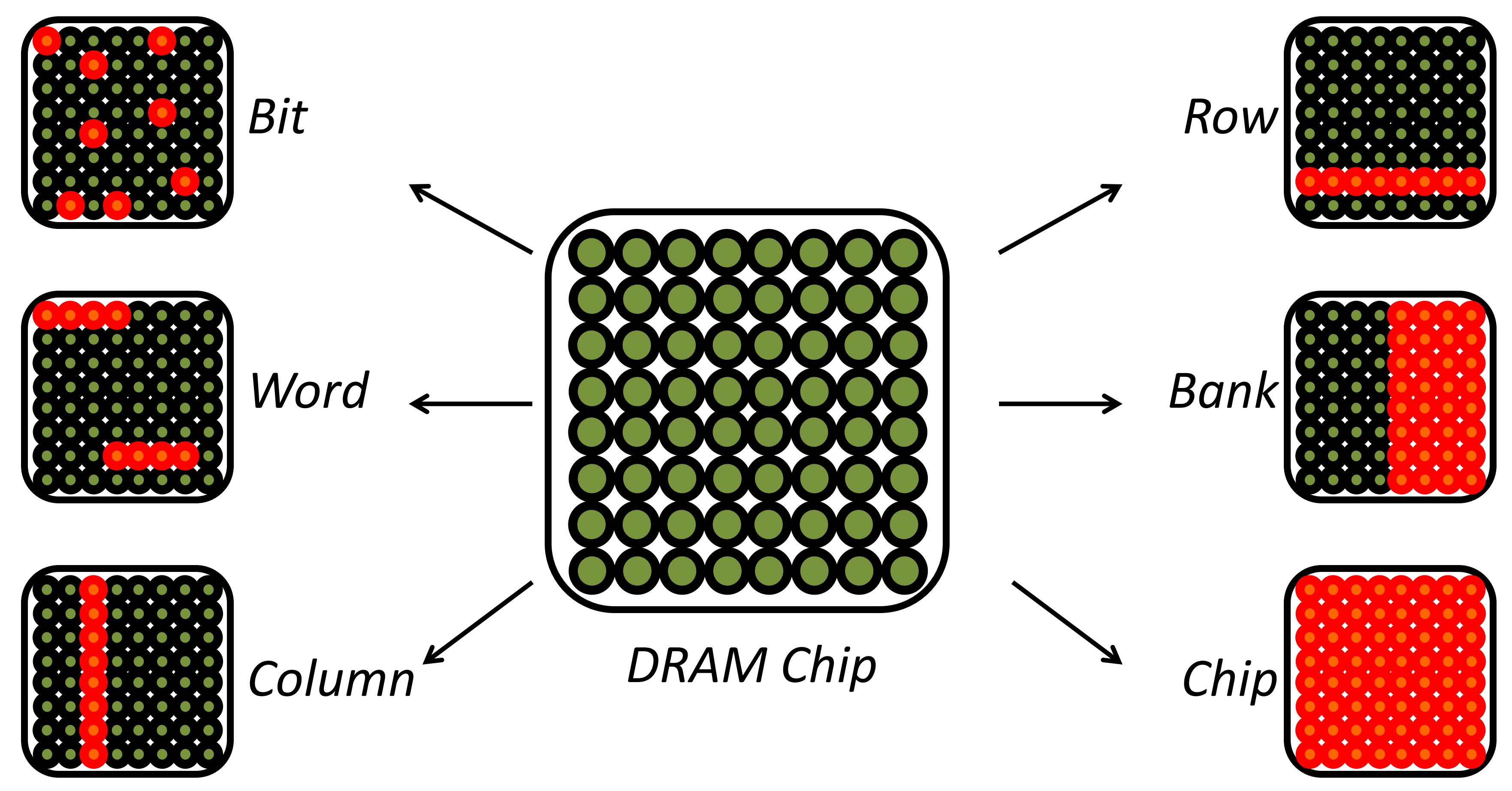
TYPES OF RUNTIME FAULTS



GRANULARITY OF FAULTS



GRANULARITY OF FAULTS



Two types of Runtime Faults @ Many Granularities

RUNTIME FAULTS ARE PERVASIVE

Efficient solutions to mitigate runtime failures

GOAL

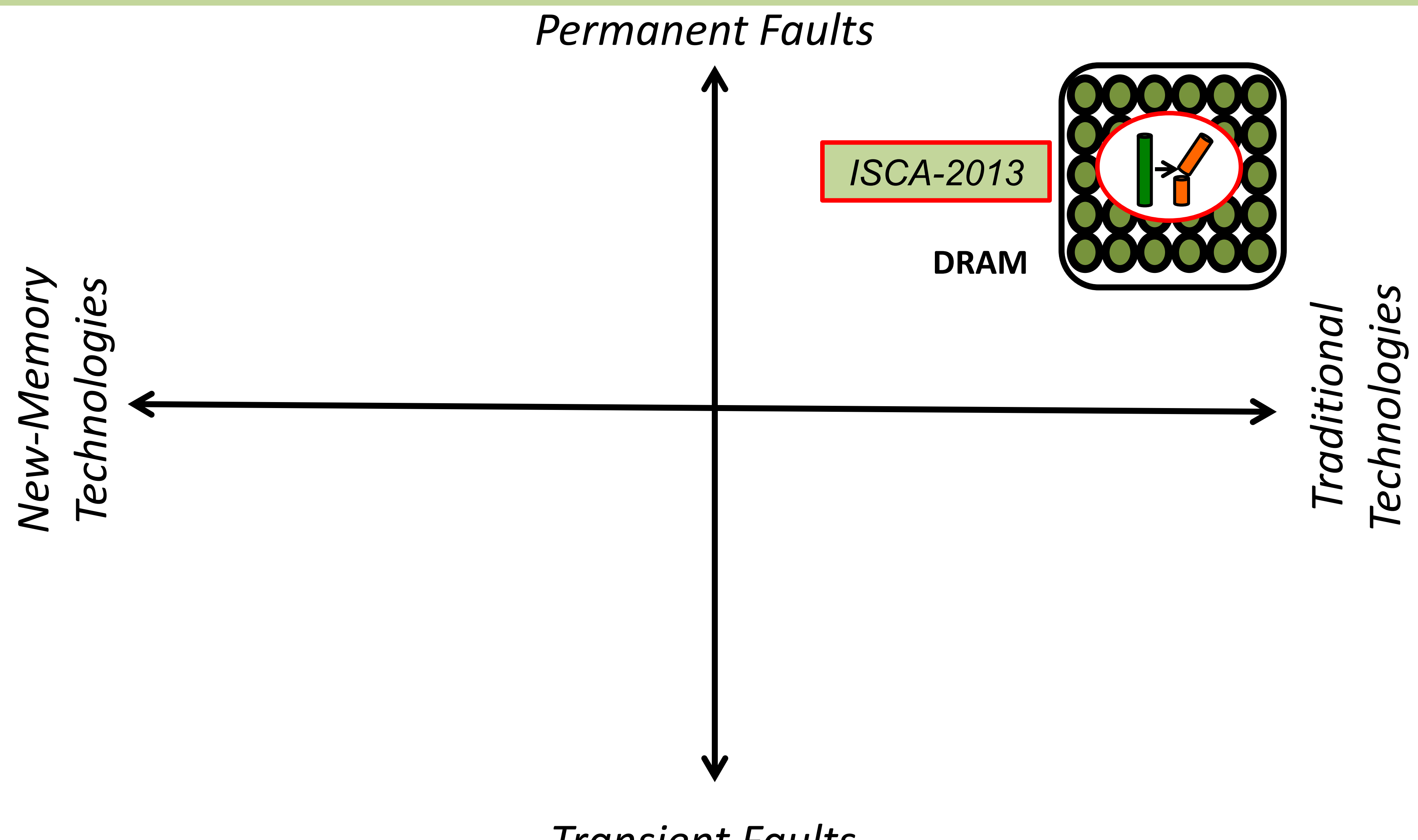
- Hurdles for Moore’s Law: Scaling & Runtime Faults
- Conventional techniques: Costly/Ineffective

GOAL

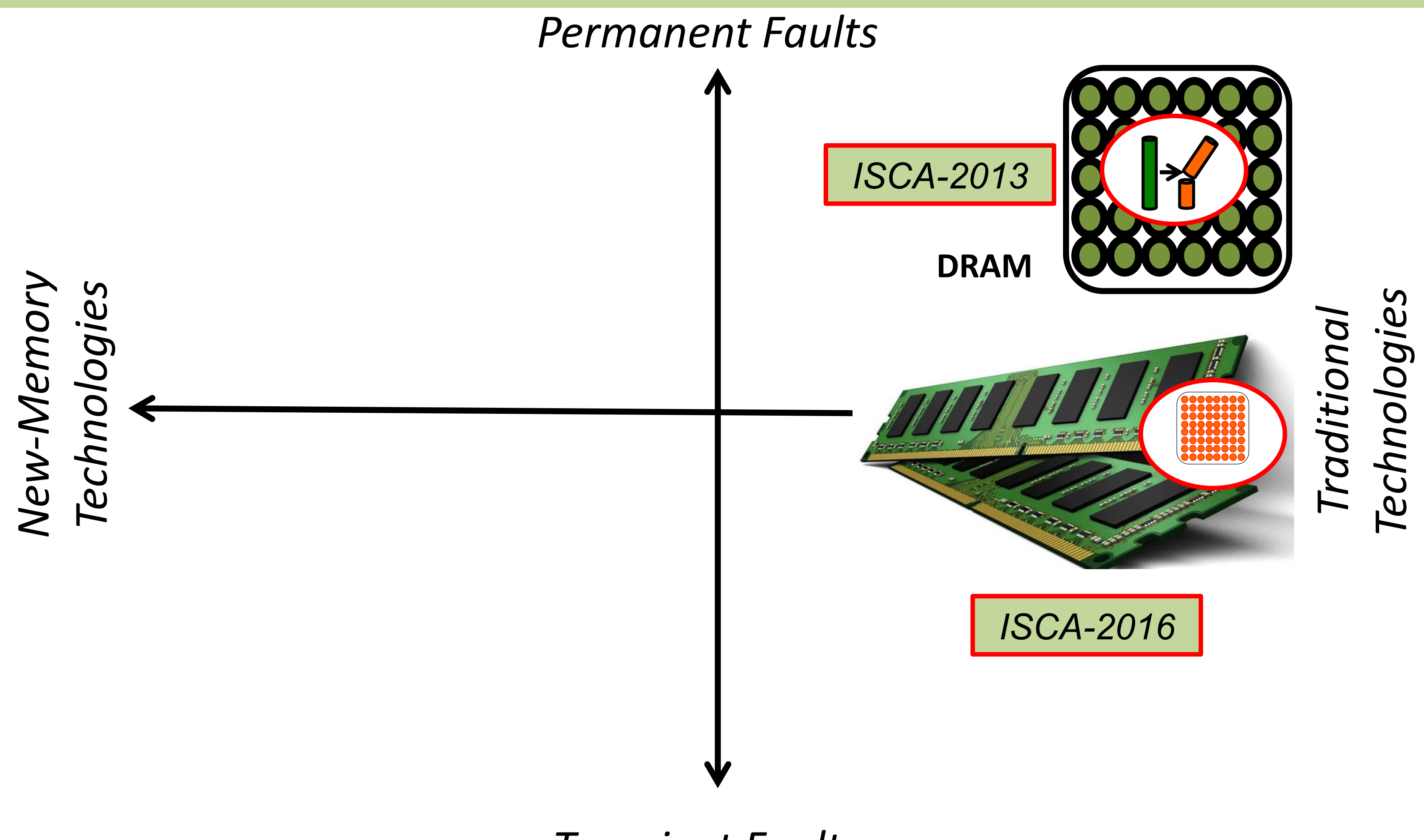
- Hurdles for Moore’s Law: Scaling & Runtime Faults
- Conventional techniques: Costly/Ineffective

Ultra low-cost solutions to sustain Moore’s Law in memories using architecture-level approaches

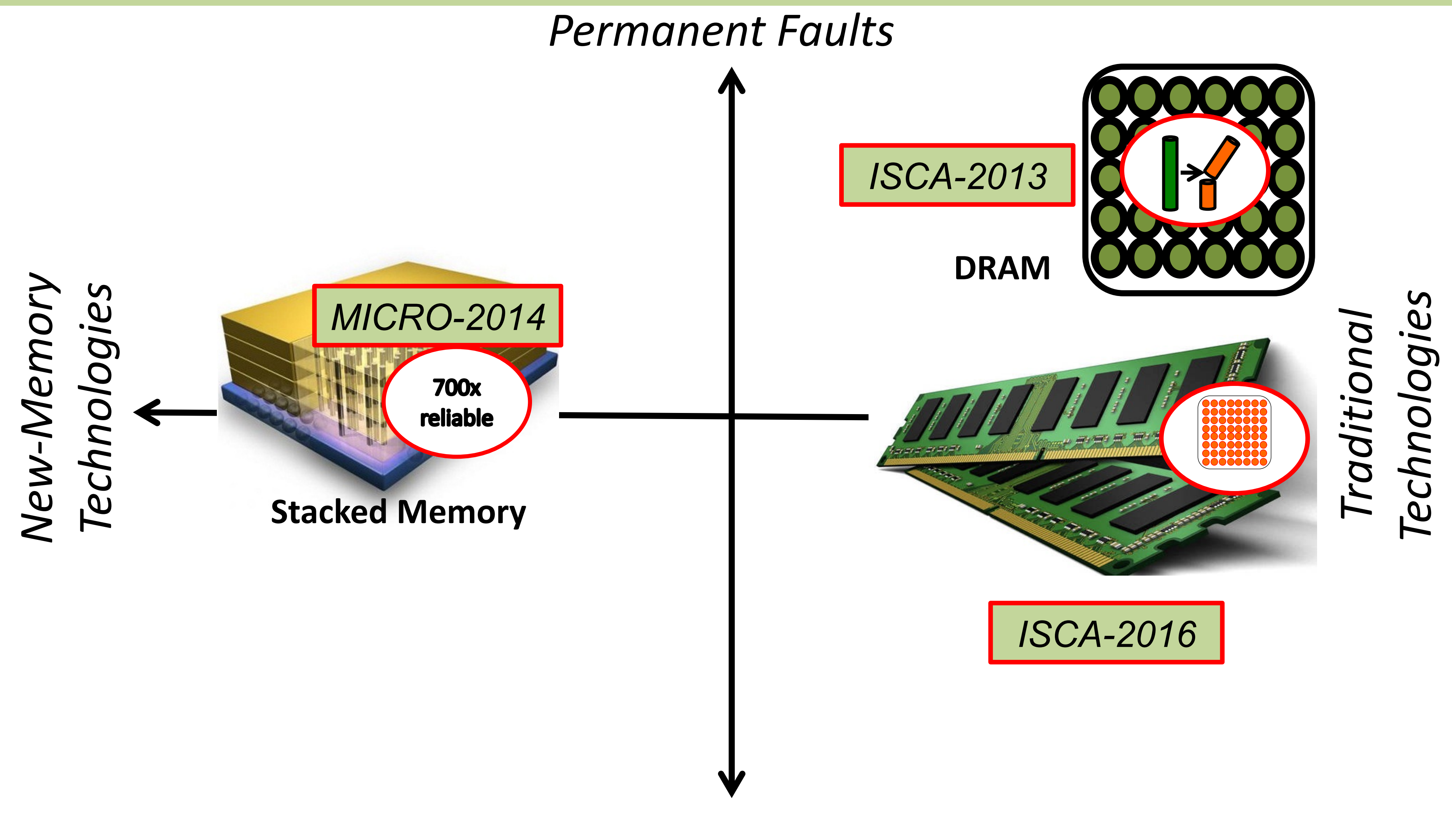
BROAD IMPACT AND SOLUTIONS



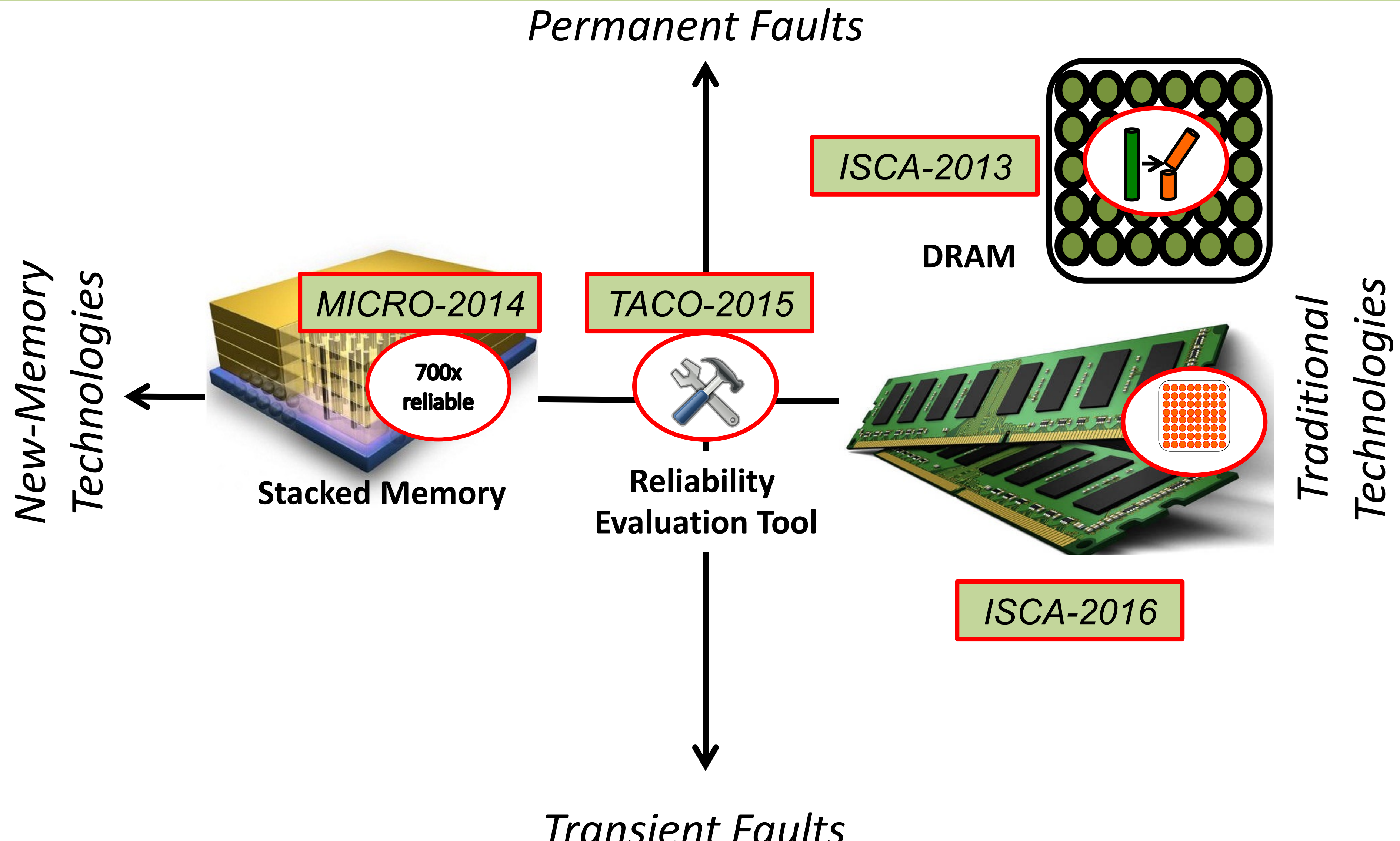
BROAD IMPACT AND SOLUTIONS



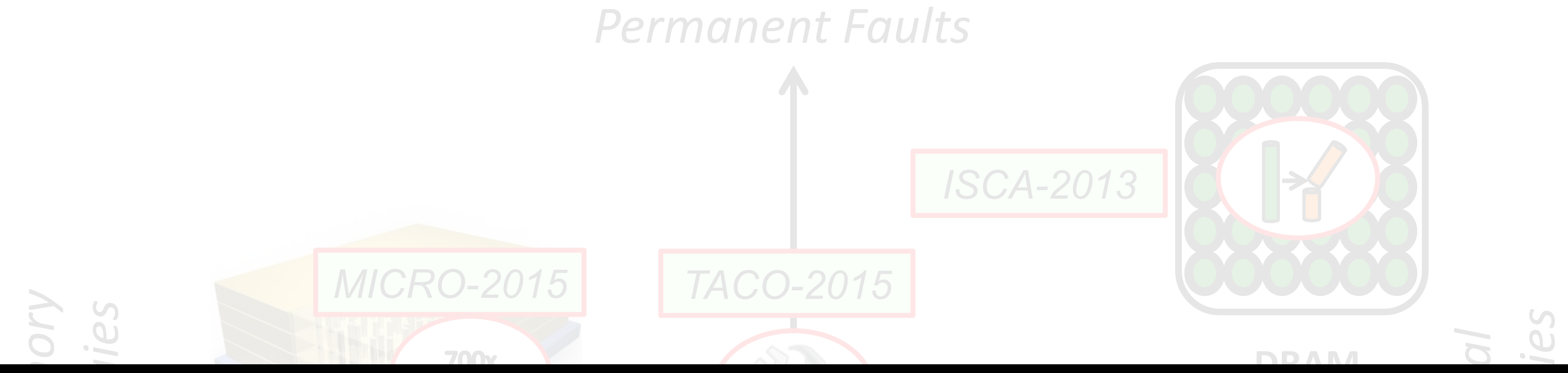
BROAD IMPACT AND SOLUTIONS



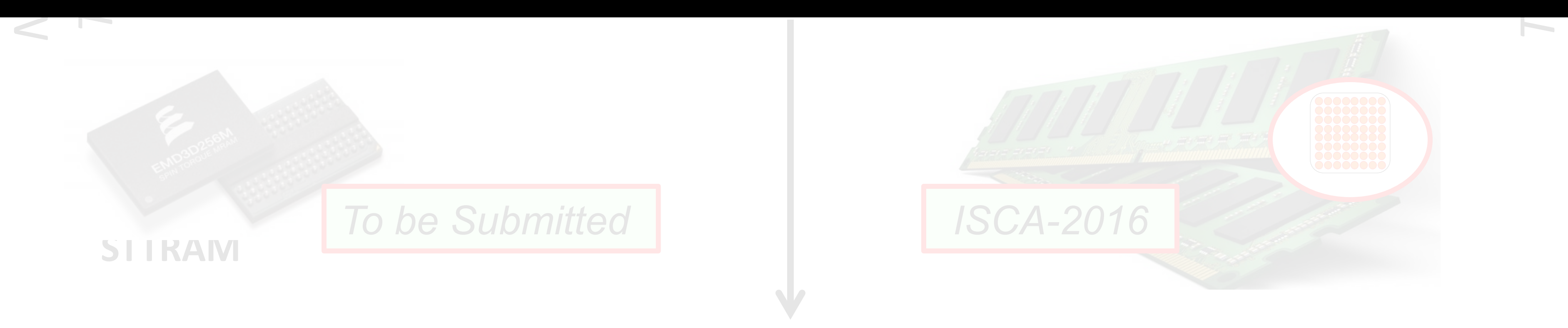
BROAD IMPACT AND SOLUTIONS



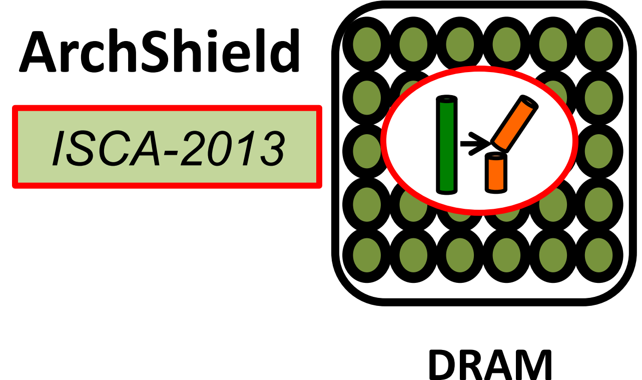
BROAD IMPACT AND SOLUTIONS

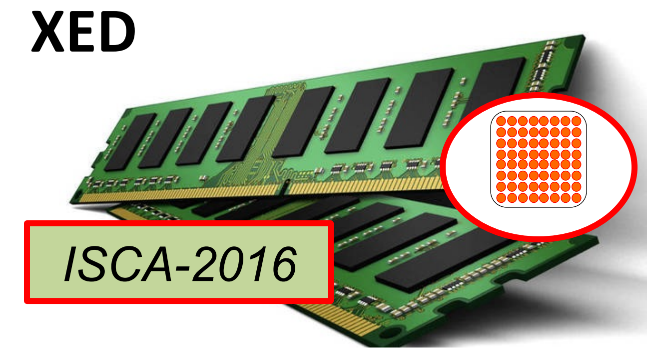


Low-cost architectural techniques to enable reliable and scalable memory systems → Sustain Moore's Law



THIS TALK





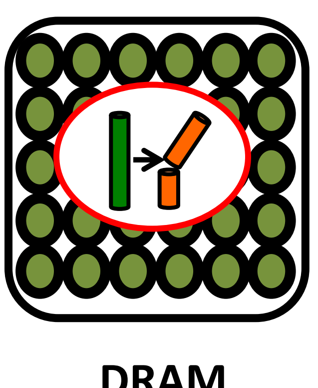
ArchShield: Architectural Framework for Assisting DRAM Scaling By Tolerating High Error-Rates

ISCA-2013

Prashant Nair

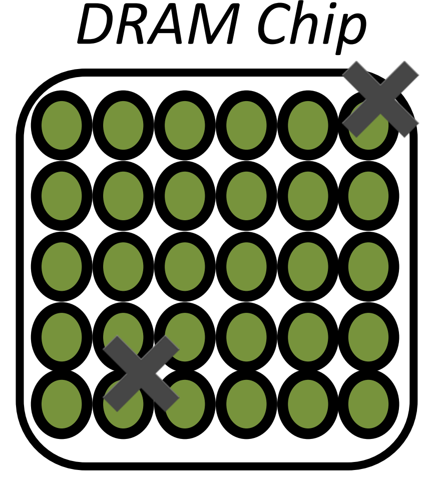
Daehyun Kim

Moinuddin Qureshi



PROBLEM: RELIABLE TECHNOLOGY SCALING

DRAM Chip



H

b

Aspect Ratio = H/b

Aspect Ratio

Source: S. J. Hong (Hynix), IEDM 2010

Aspect Ratio of Storage Node: 20, 40, 60, 80, 100

Technology Node (nm): 70, 60, 50, 40, 30, 20, 10

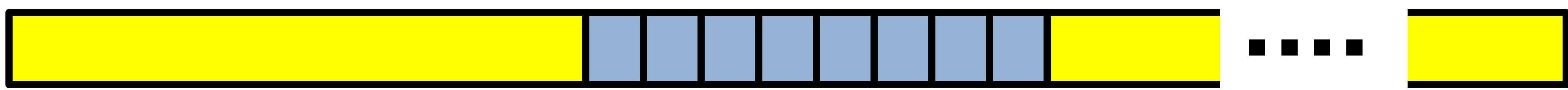
TERMINOLOGY: ROW, LINE, AND WORD

Activate a **Row** *of* **8KB**

....

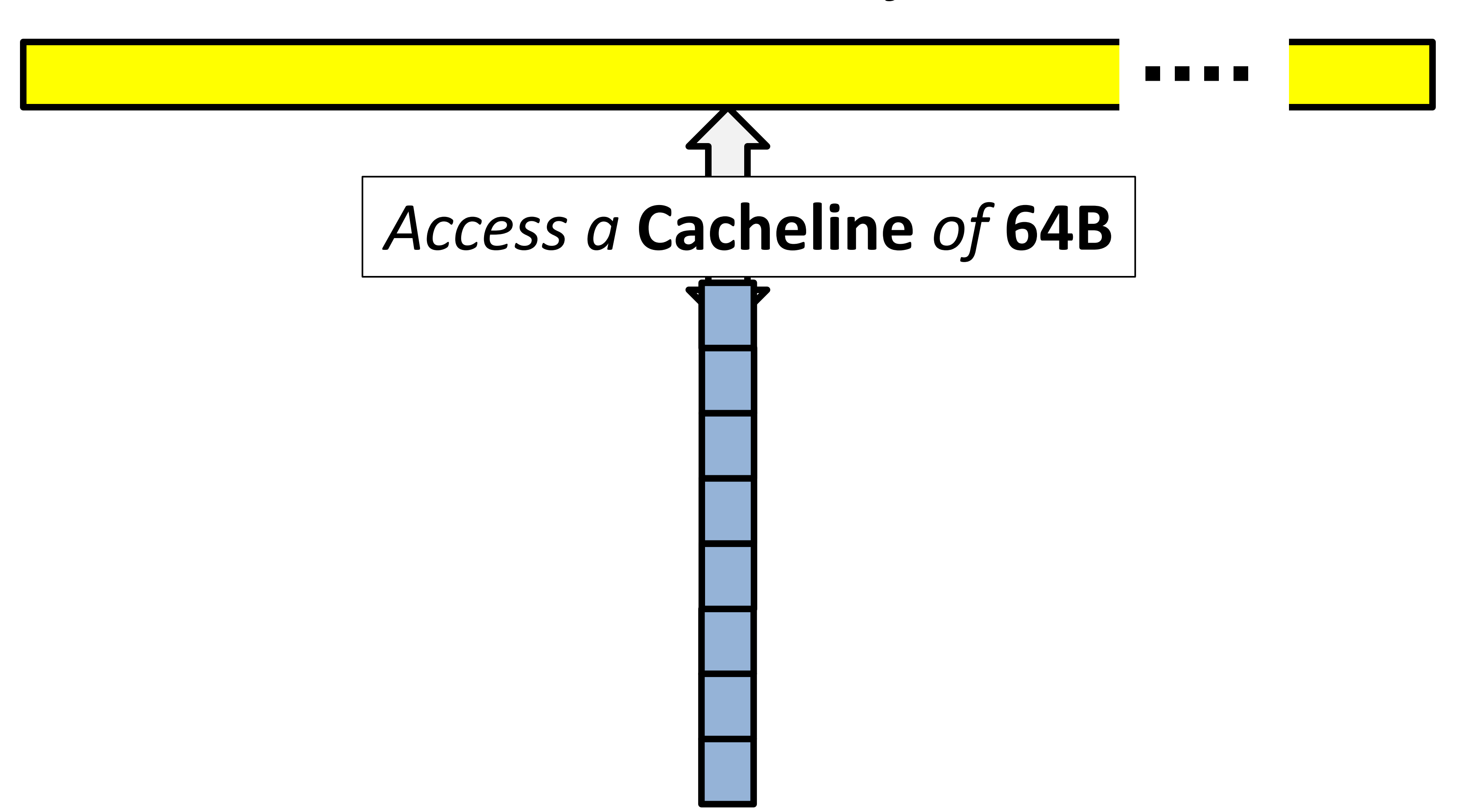
TERMINOLOGY: ROW, LINE, AND WORD

Activate a **Row** *of* **8KB**

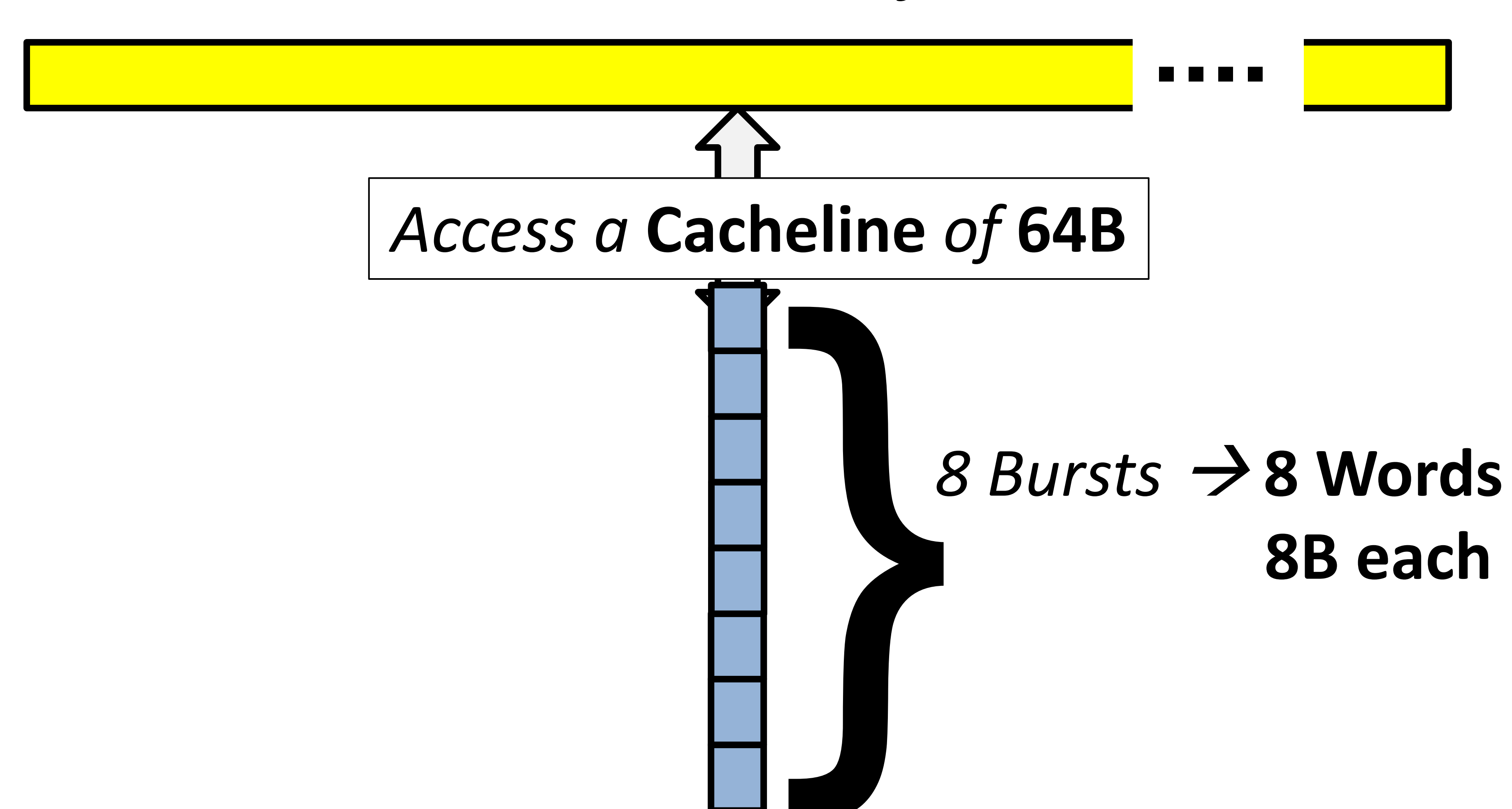


Access a **Cacheline** *of* **64B**

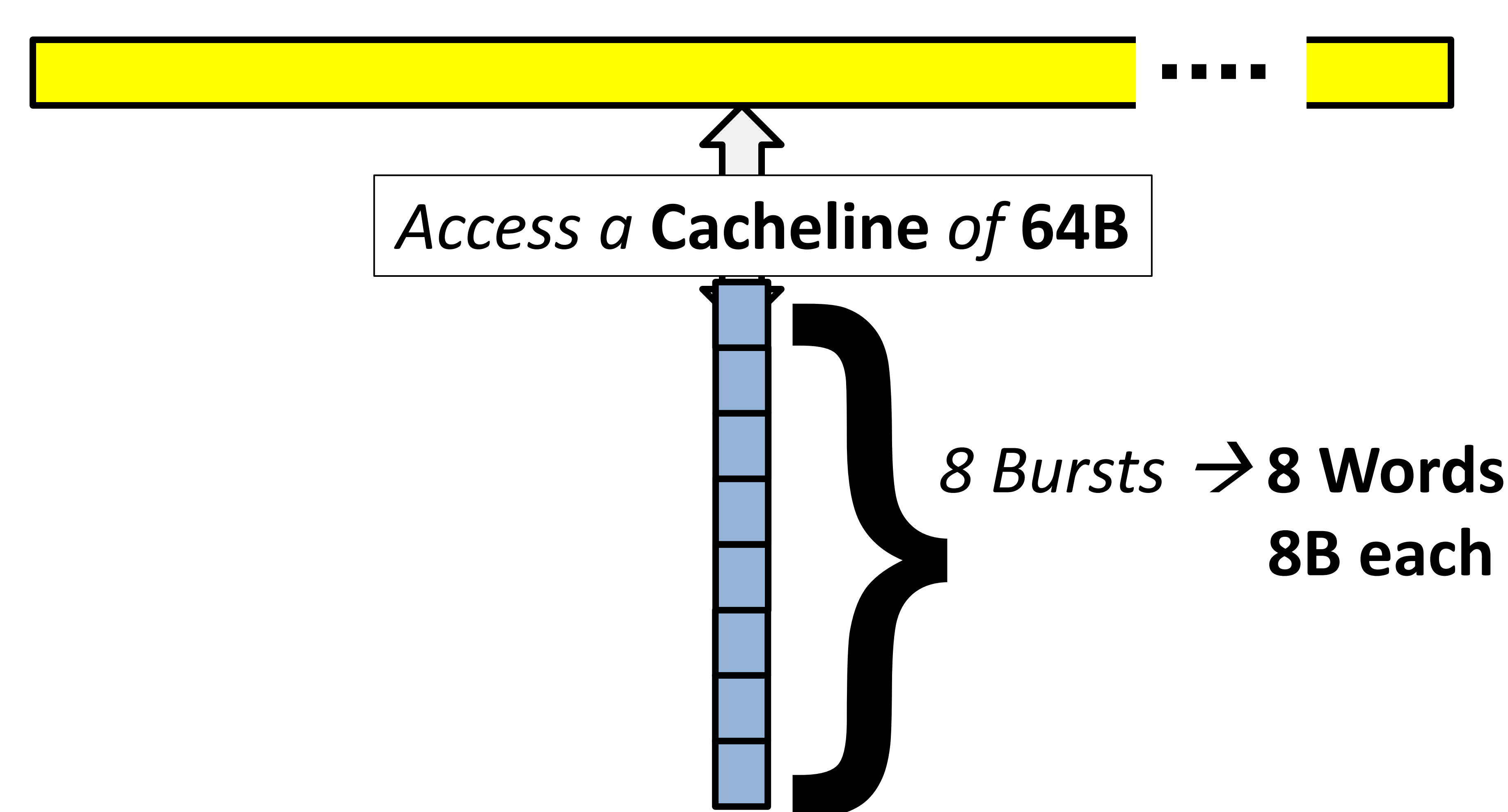
TERMINOLOGY: ROW, LINE, AND WORD



TERMINOLOGY: ROW, LINE, AND WORD

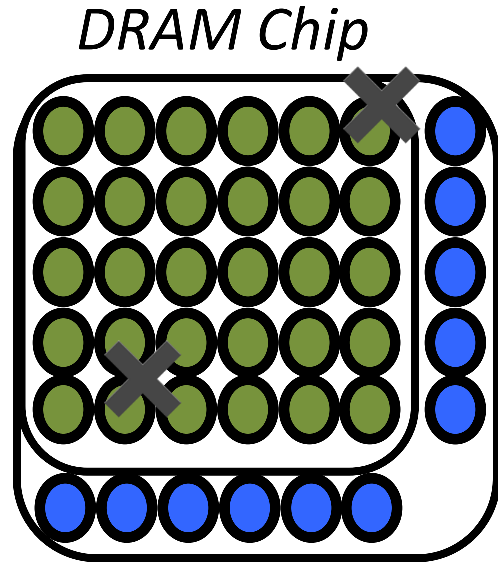


TERMINOLOGY: ROW, LINE, AND WORD



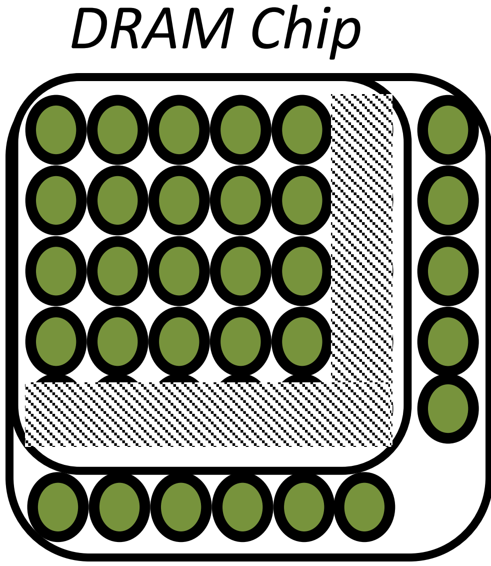
Memory is accessed in 64B Cachelines → 8 Words

SCALING OPTION 1: ROW/COL SPARING



Spare Rows and Columns

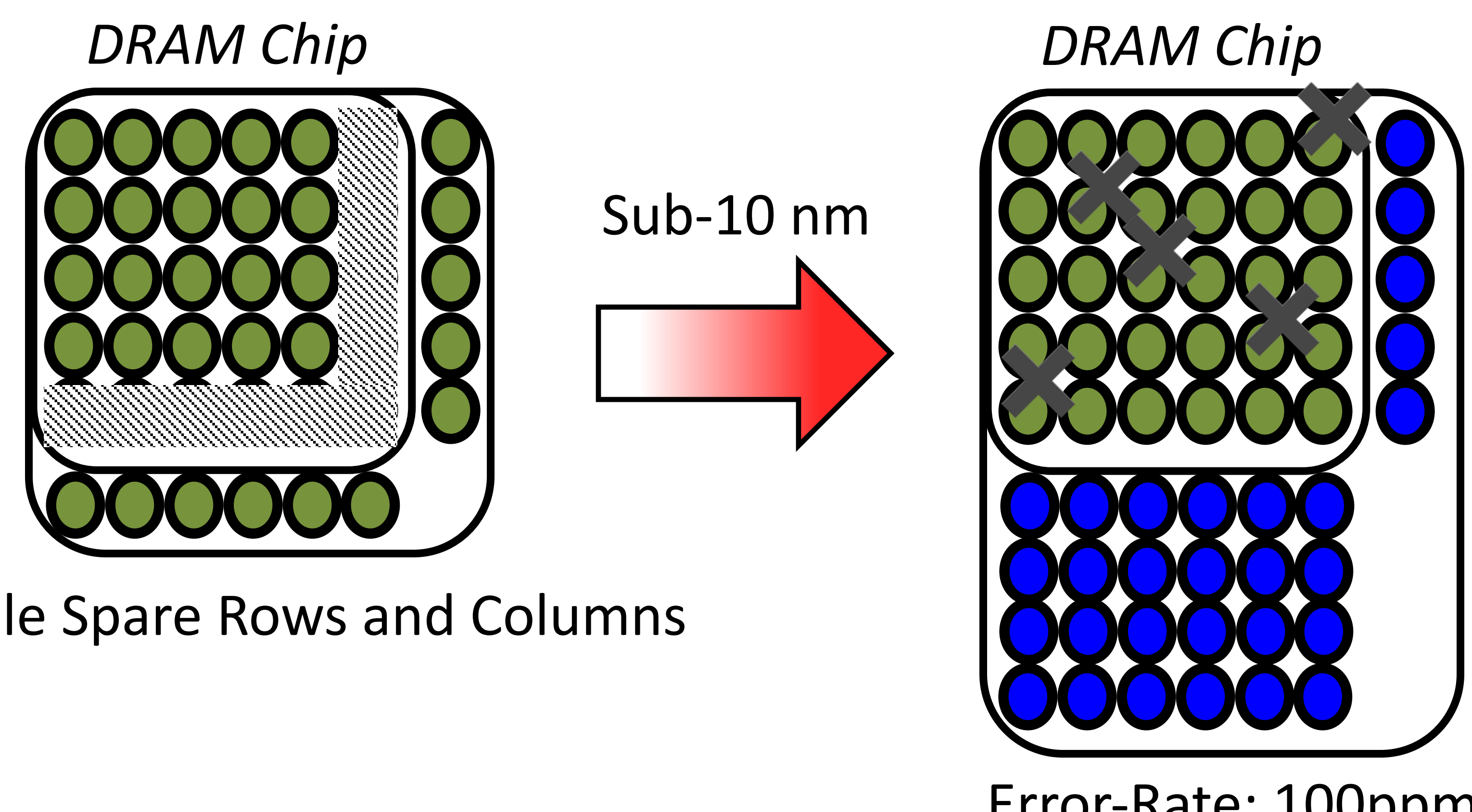
SCALING OPTION 1: ROW/COL SPARING



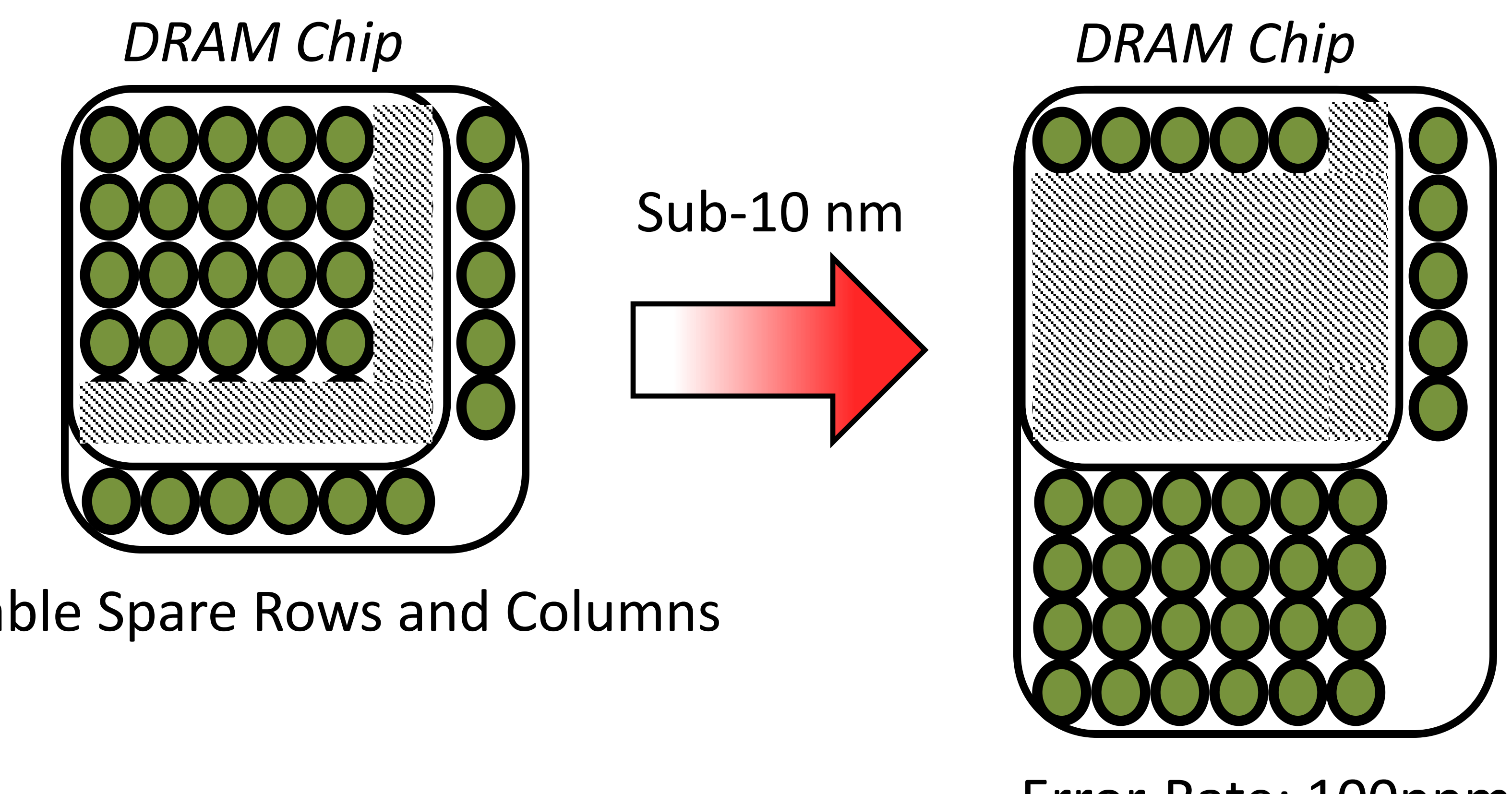
Enable Spare Rows and Columns

Entire row or column sacrificed for a few faulty cells

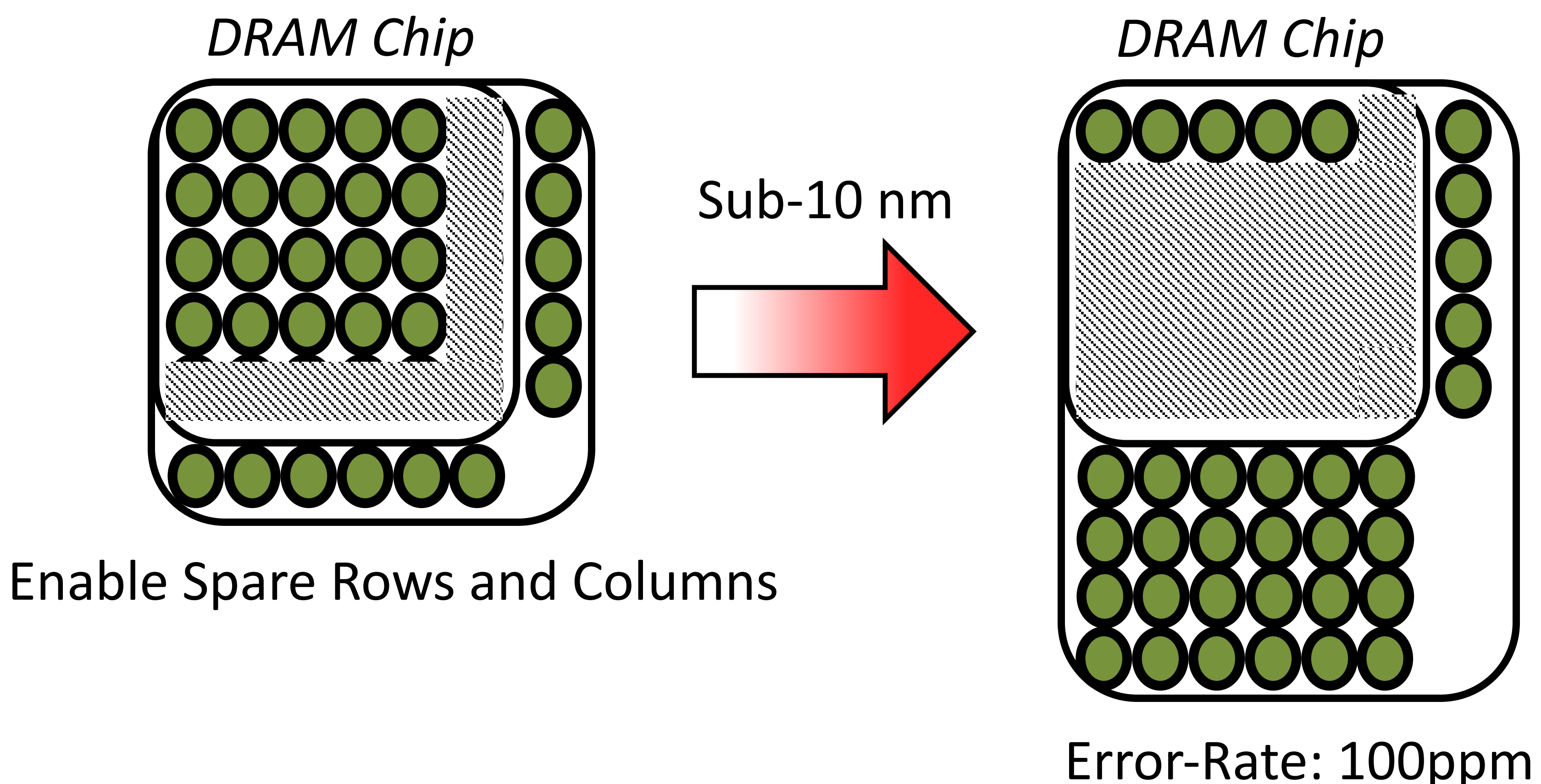
SCALING OPTION 1: ROW/COL SPARING



SCALING OPTION 1: ROW/COL SPARING



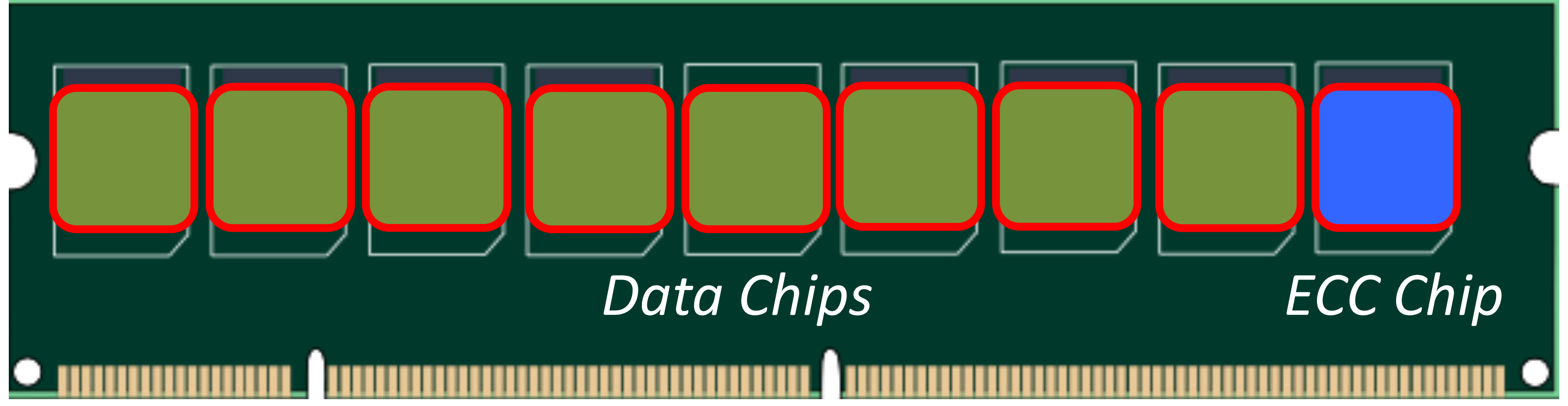
SCALING OPTION 1: ROW/COL SPARING



Row/Column sparing has large area overheads

SCALING OPTION 2: ECC-DIMM

Single Error Correct Double Error Detect (SECDED)



SCALING OPTION 2: ECC-DIMM

Single Error Correct Double Error Detect (SECDED)

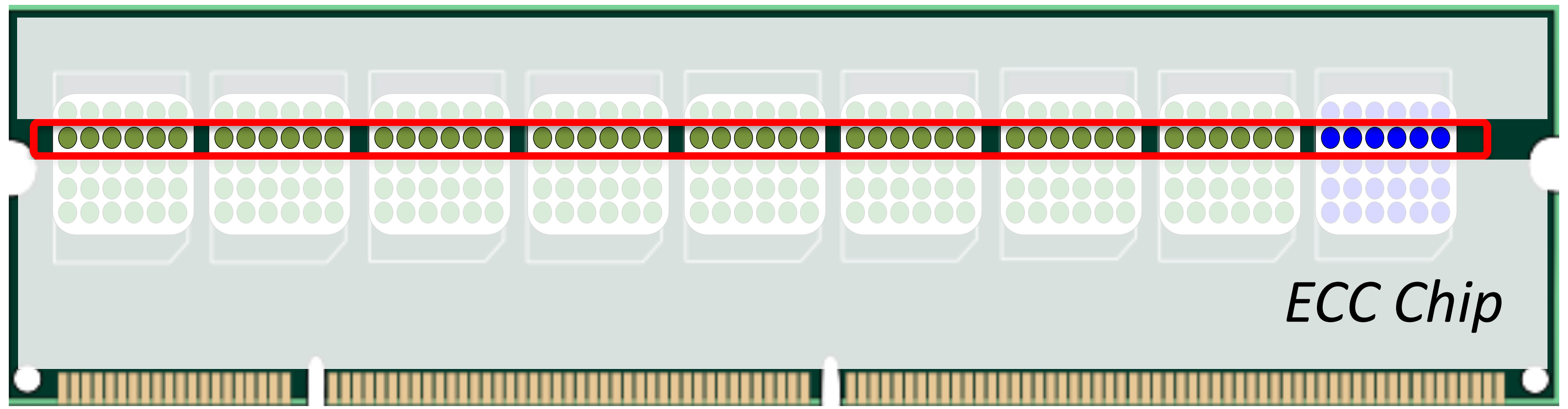
ECC Chip

Corrects single-bit fault in each word (8 Bytes)

For 8GB DIMM → 1 Billion words

SCALING OPTION 2: ECC-DIMM

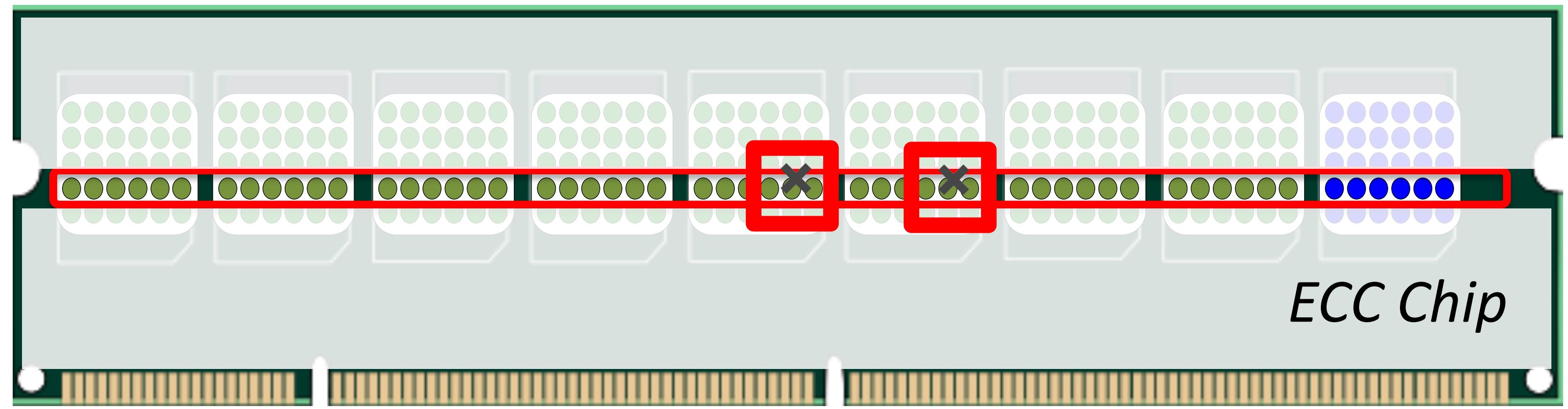
Single Error Correct Double Error Detect (SECDED)



Examining all the words…

SCALING OPTION 2: ECC-DIMM

Single Error Correct Double Error Detect (SECDED)



High chance of two faults in atleast one word

Birthday Paradox

SCALING OPTION 2: ECC-DIMM

Single Error Correct Double Error Detect (SECDED)

ECC Chip

High chance of two faults in atleast one word

Birthday Paradox

8GB DIMM → 1 billion words = N

*Expected faults till double-fault = 1.25*Sqrt(N) = 40K faults → 0.5 ppm*

SCALING OPTION 2: ECC-DIMM

Single Error Correct Double Error Detect (SECDED)

ECC Chip

High chance of two faults in atleast one word

Birthday Paradox

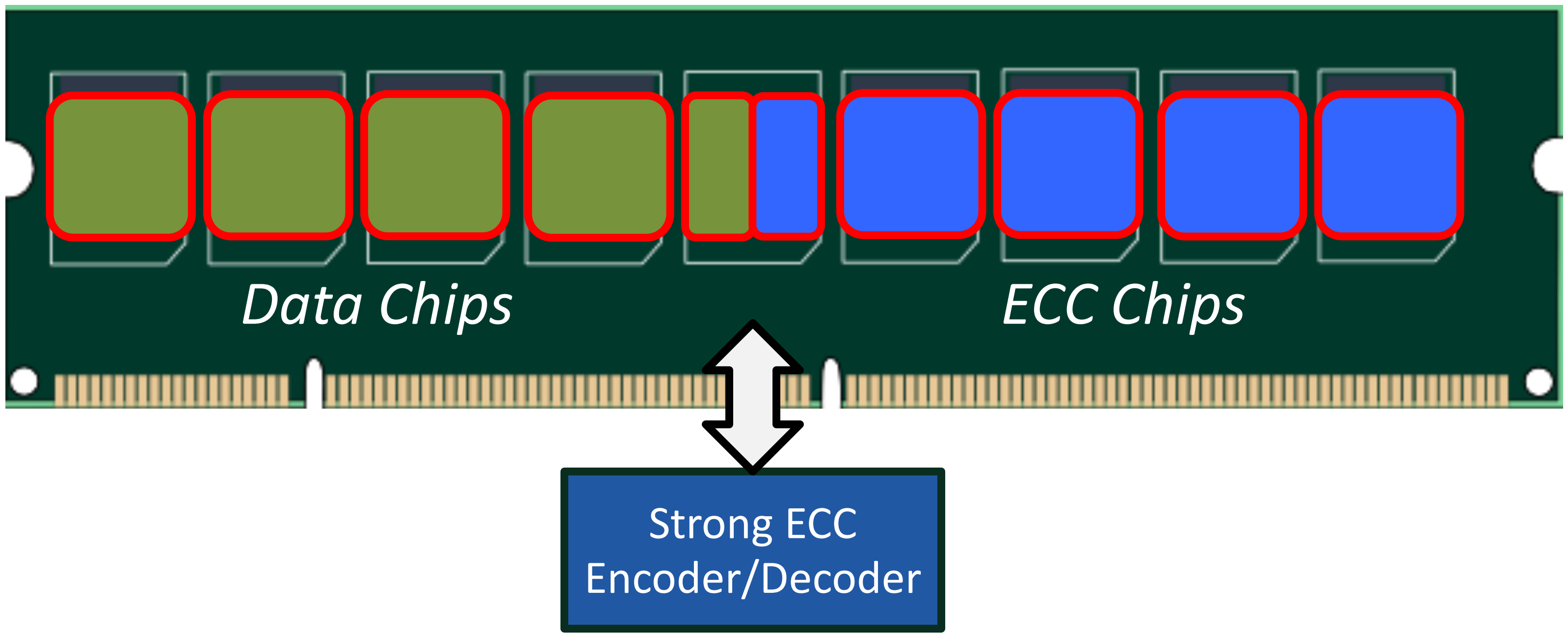
8GB DIMM → 1 billion words = N

*Expected faults till double-fault = 1.25*Sqrt(N) = 40K faults → 0.5 ppm*

SECDED alone cannot protect against scaling faults

SCALING OPTION 3: STRONG ECC

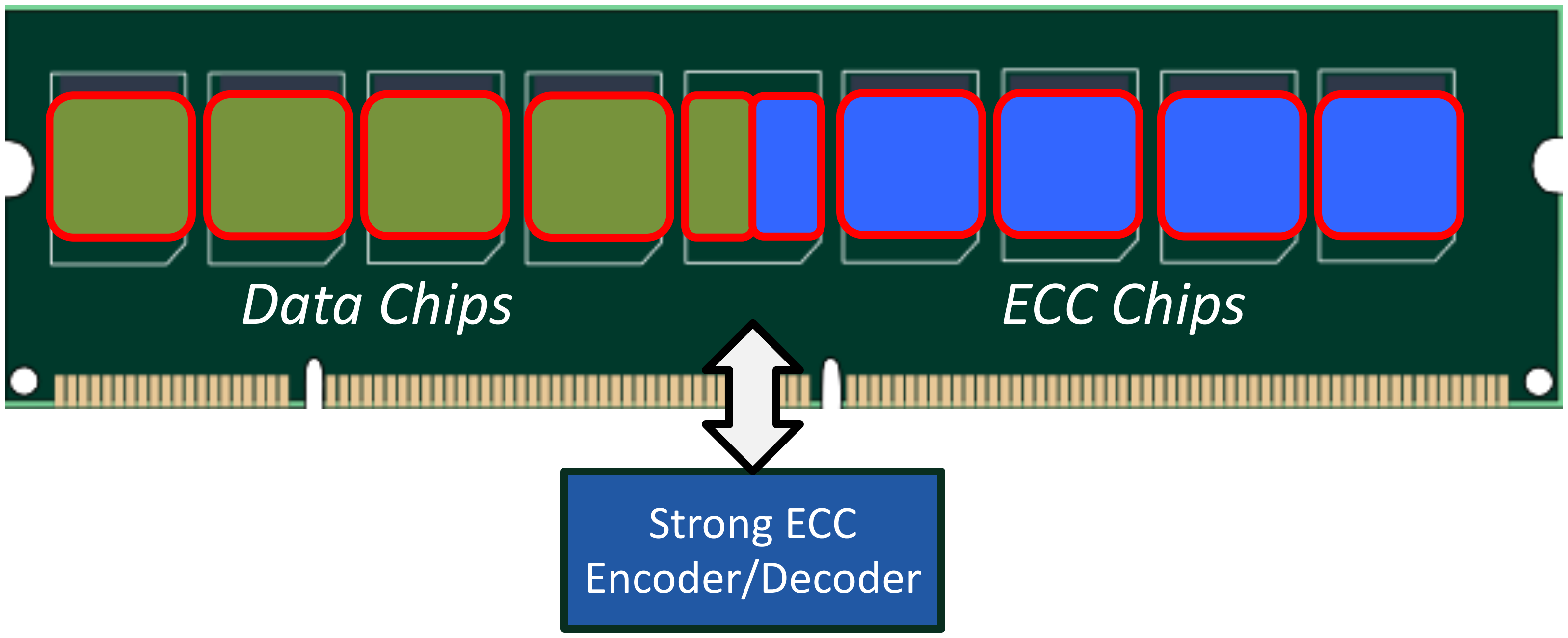
Strong ECC are robust, but complex and costly



Memory requests incur encoding/decoding latency

SCALING OPTION 3: STRONG ECC

Strong ECC are robust, but complex and costly

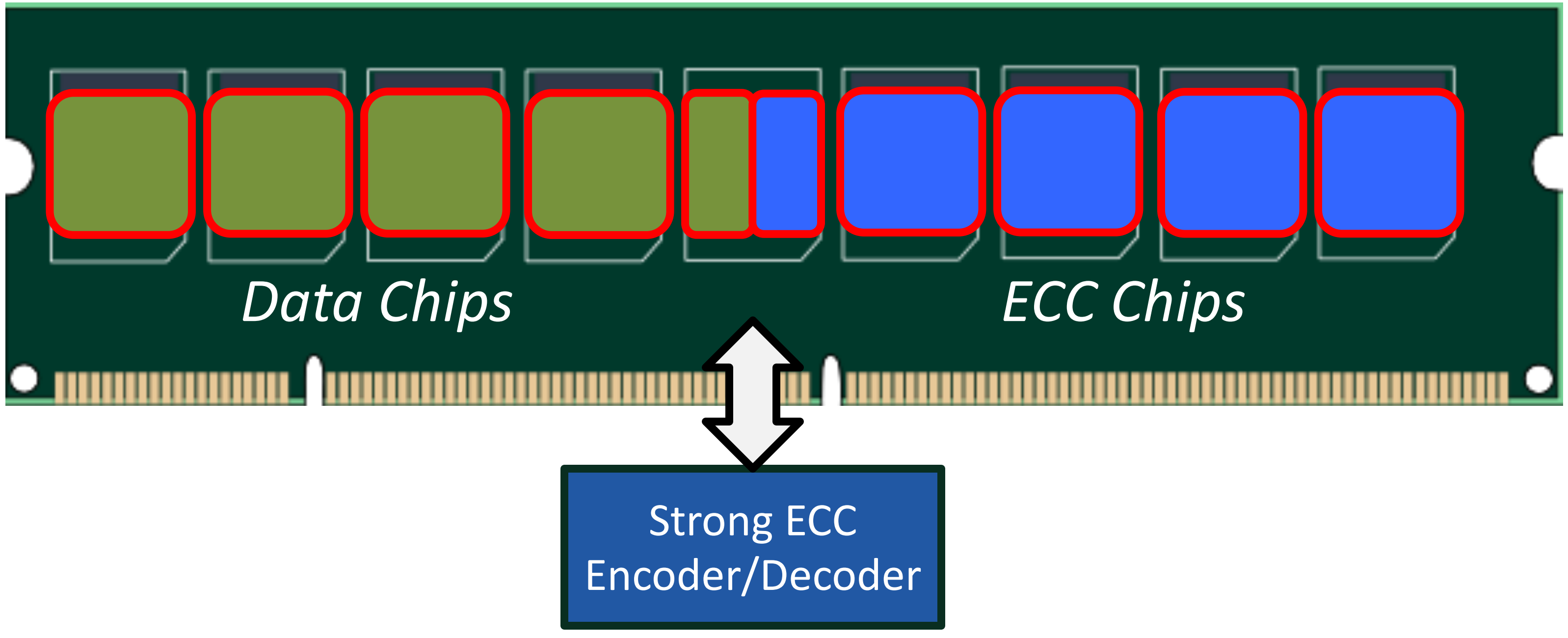


Memory requests incur encoding/decoding latency

Bit Error Rate (BER) of 100 ppm: ECC-4 → 50% storage overhead

SCALING OPTION 3: STRONG ECC

Strong ECC are robust, but complex and costly

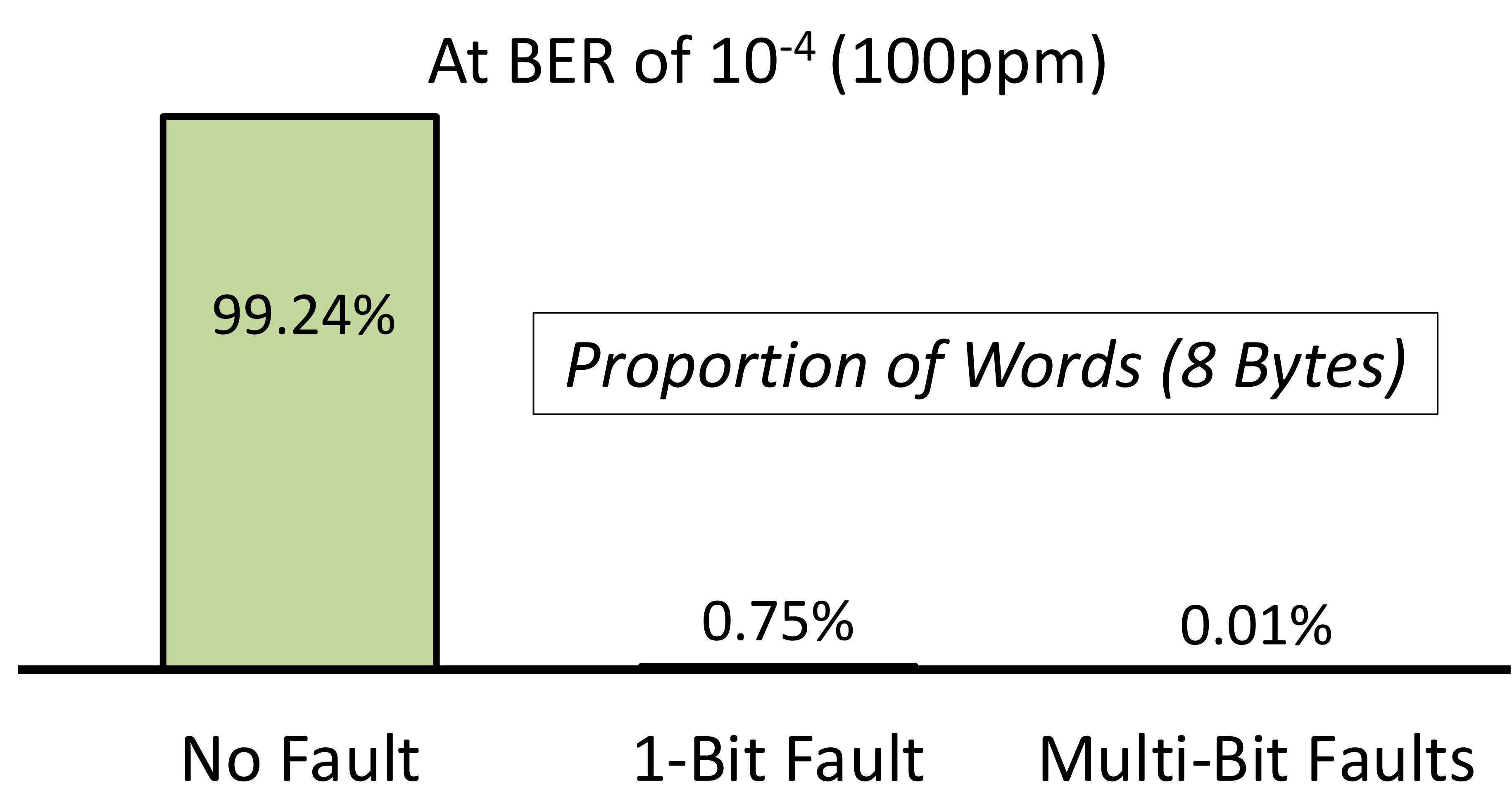


Memory requests incur encoding/decoding latency

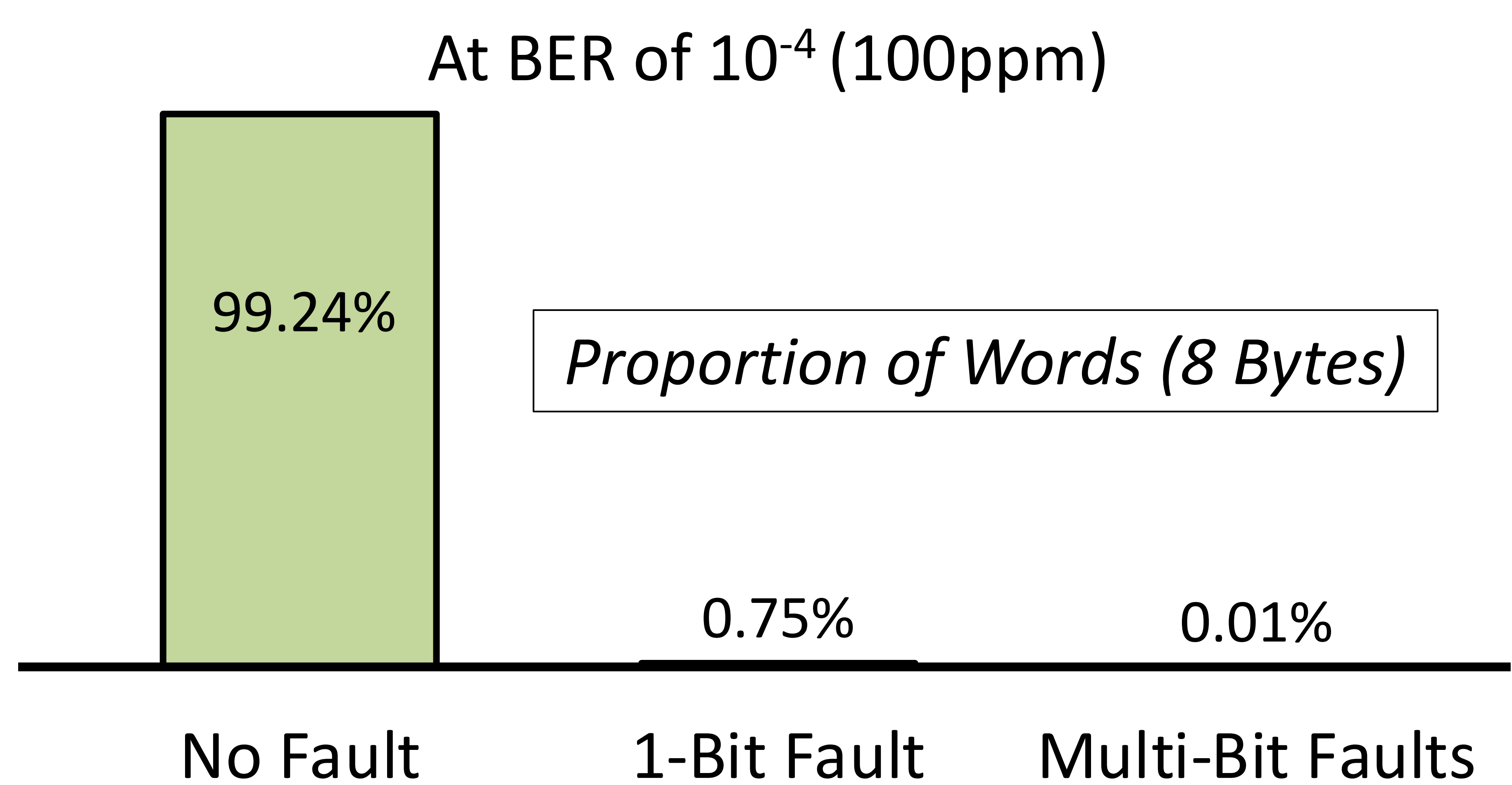
Bit Error Rate (BER) of 100 ppm: ECC-4 → 50% storage overhead

Strong ECC are inefficient for tolerating errors

DISSECTING FAULT PROBABILITIES



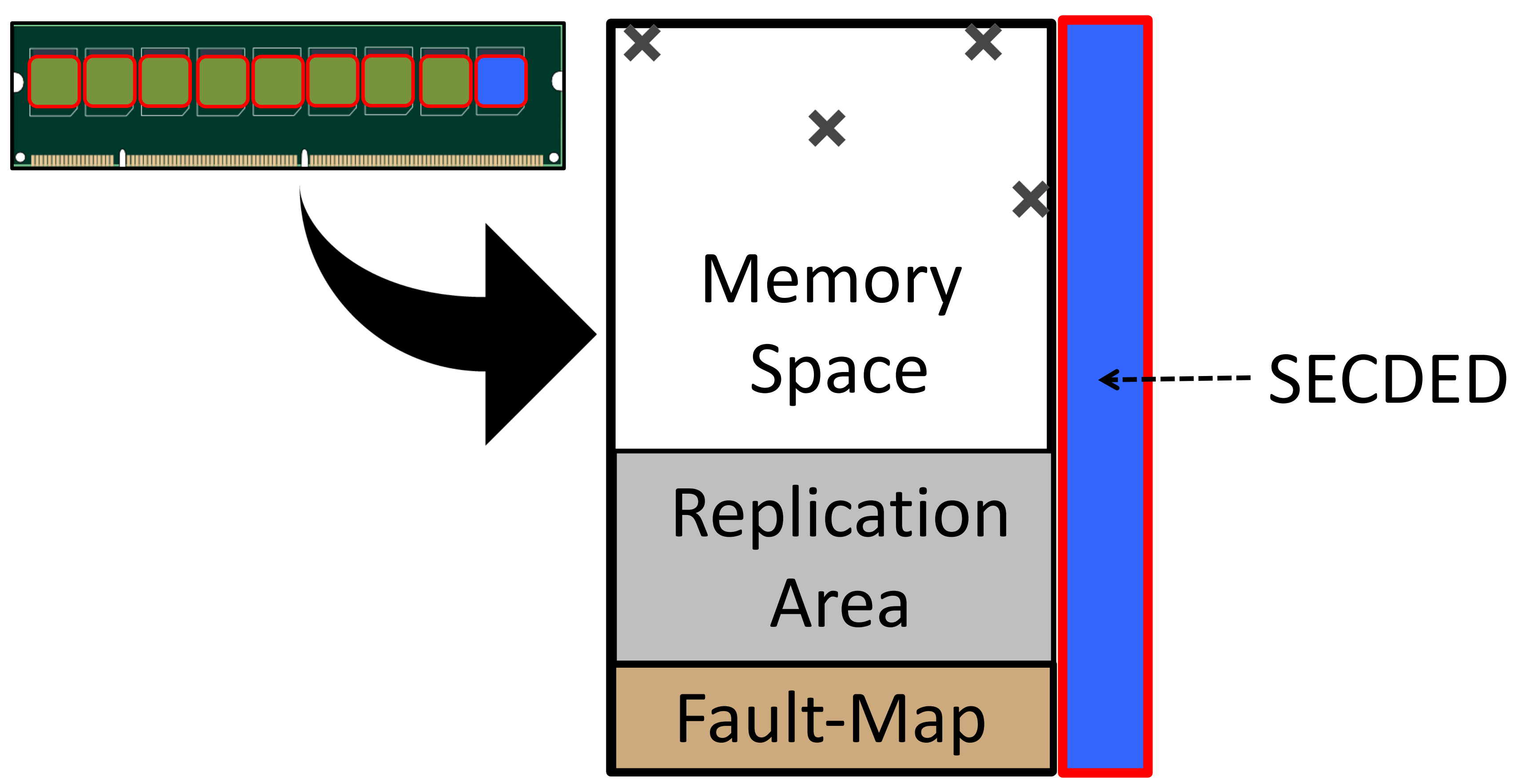
DISSECTING FAULT PROBABILITIES



Most faults are 1-bit: Exploit skew in probability

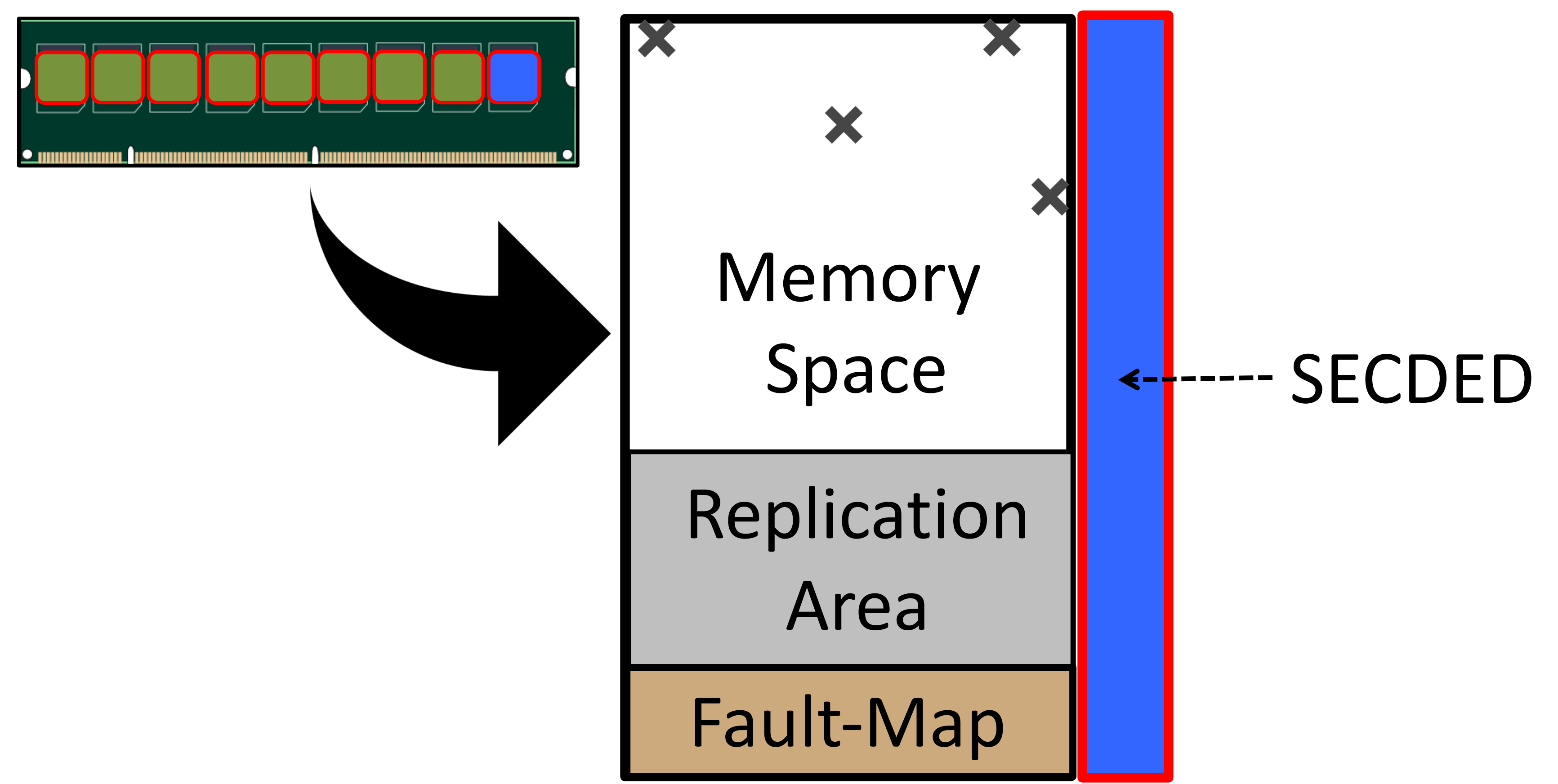
ARCHSHIELD: AN OVERVIEW

Inspired from SSDs to handle high rates of errors



ARCHSHIELD: AN OVERVIEW

Inspired from SSDs to handle high rates of errors

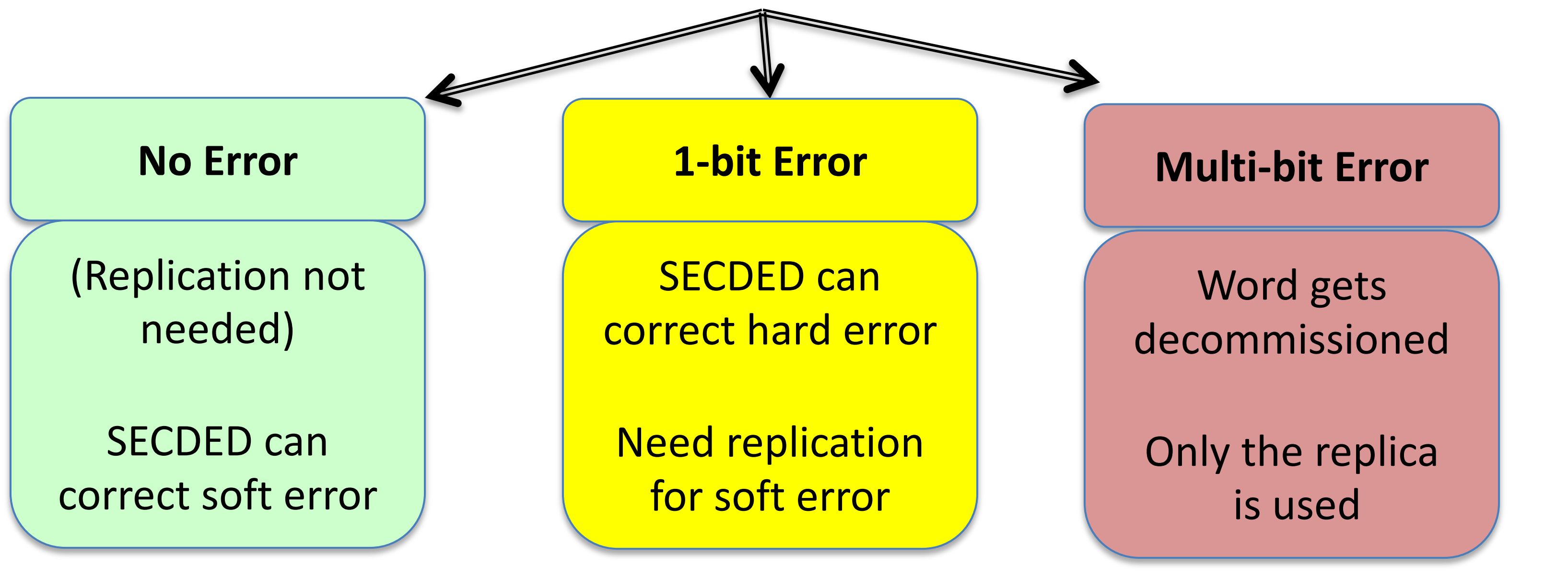


Use Replication Area and Fault-Map to handle faults

FAULTS: RUNTIME TESTING & CLASSIFICATION

During the first bootup, runtime testing is performed

Each 8B word gets classified into one of three types:

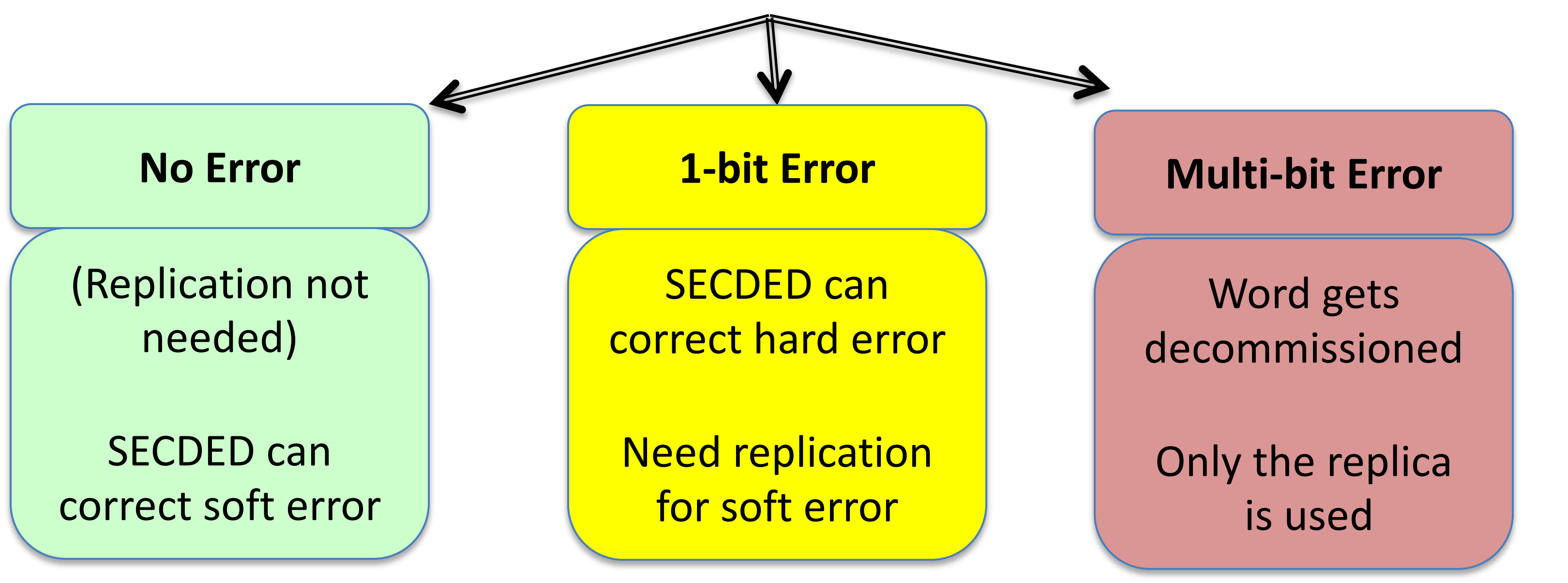


Faulty cell information: *Stored in hard drive for future use*

FAULTS: RUNTIME TESTING & CLASSIFICATION

During the first bootup, runtime testing is performed

Each 8B word gets classified into one of three types:

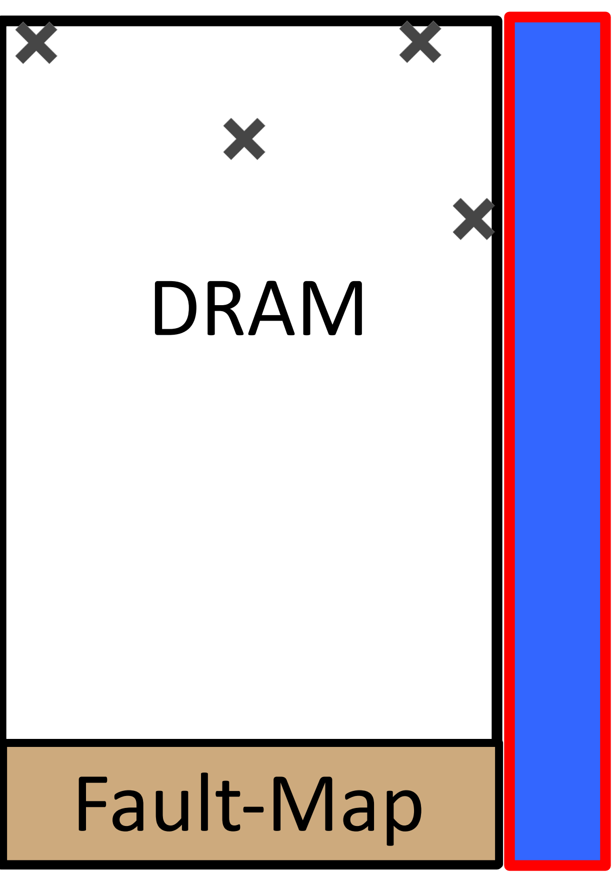


Faulty cell information: *Stored in hard drive for future use*

Identifies the faulty cells & decides type of correction

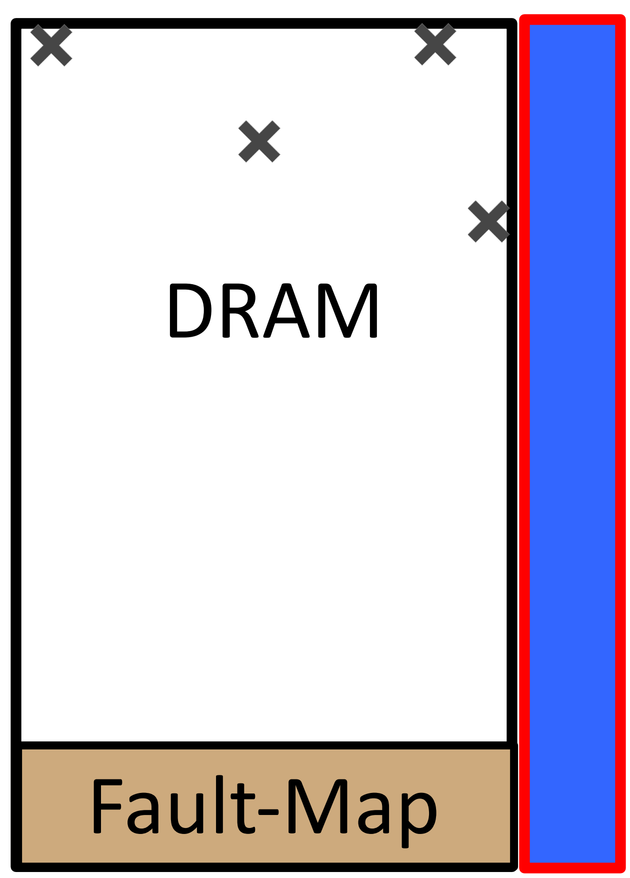
EXPOSE FAULTS USING A FAULT-MAP

A **map** to keep track of all words



EXPOSE FAULTS USING A FAULT-MAP

A **map** to keep track of all words

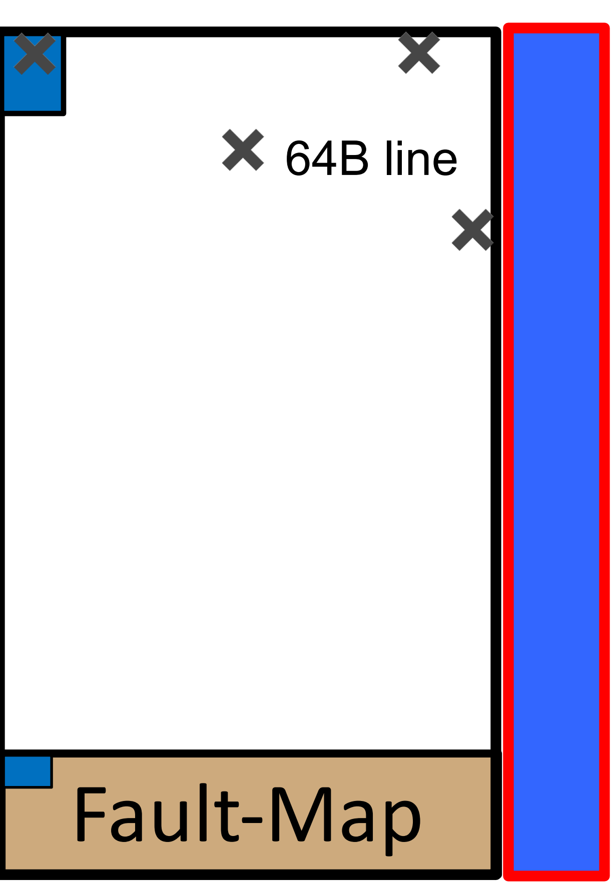


Fault-Map identifies faulty vs non-faulty words

FAULT-MAP: INFORMATION AND OVERHEAD

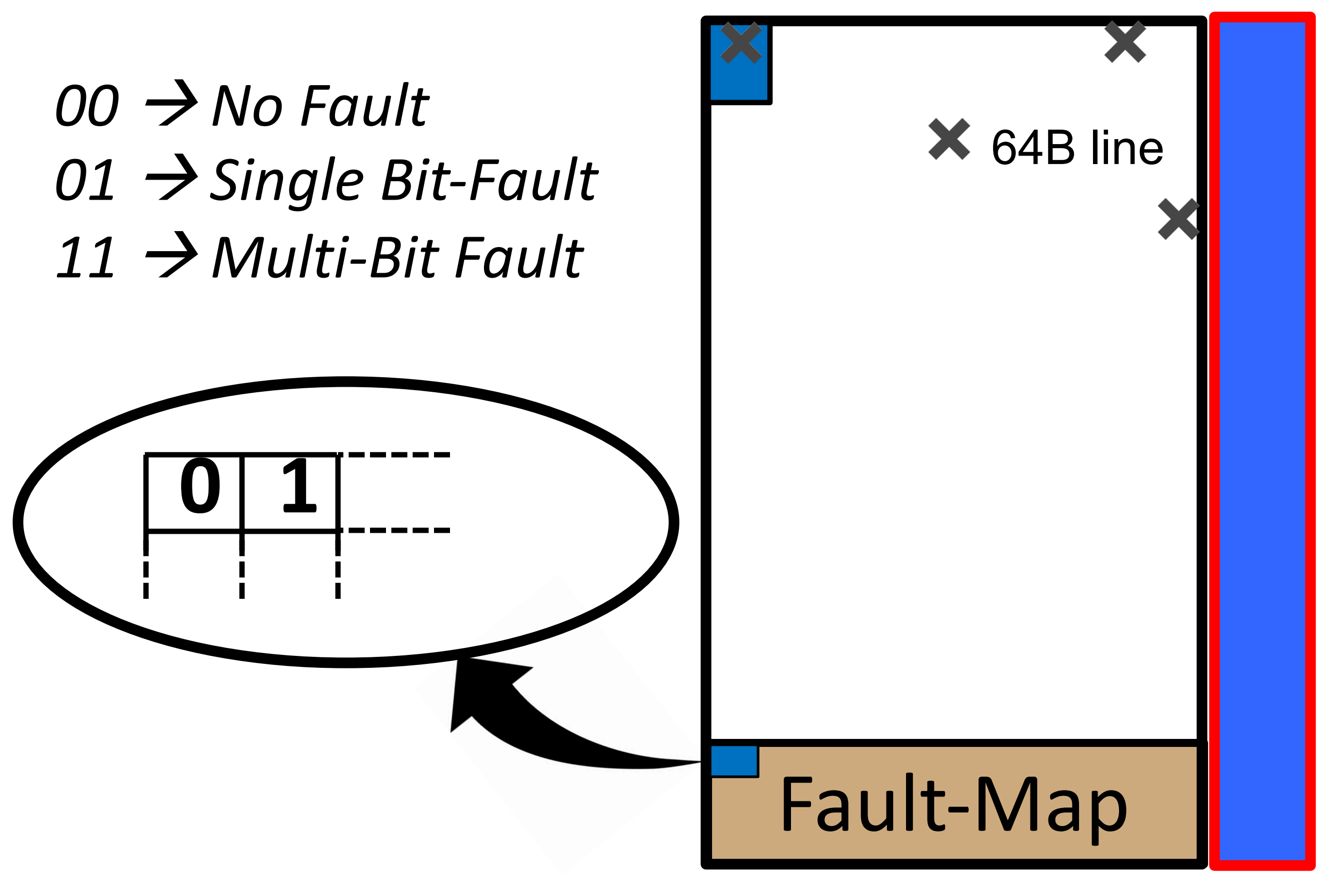
4bits (2-bits replicated) per **8B word**

00 → No Fault
01 → Single Bit-Fault
11 → Multi-Bit Fault



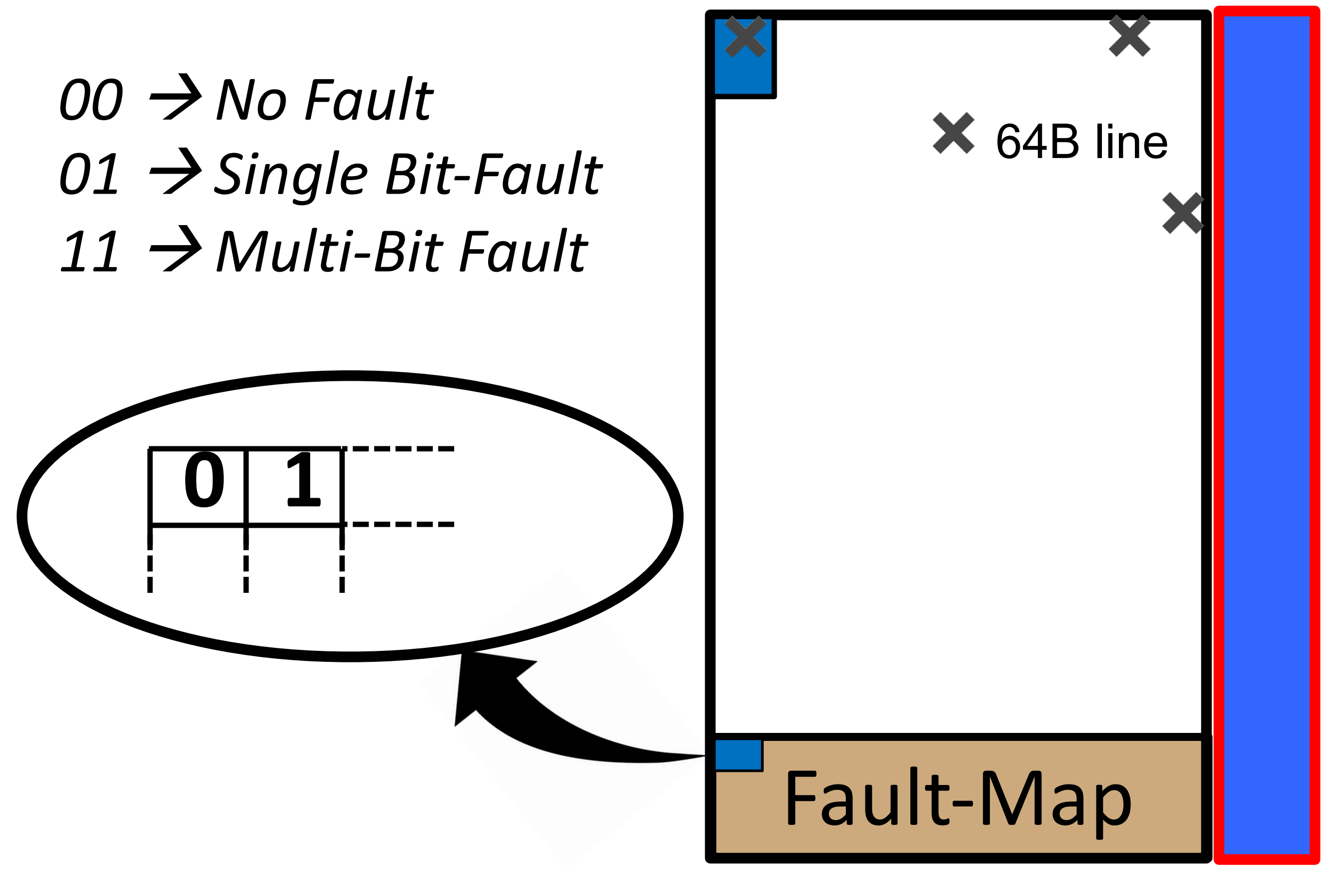
FAULT-MAP: INFORMATION AND OVERHEAD

4bits (2-bits replicated) per **8B word**



FAULT-MAP: INFORMATION AND OVERHEAD

4bits (2-bits replicated) per **8B word**

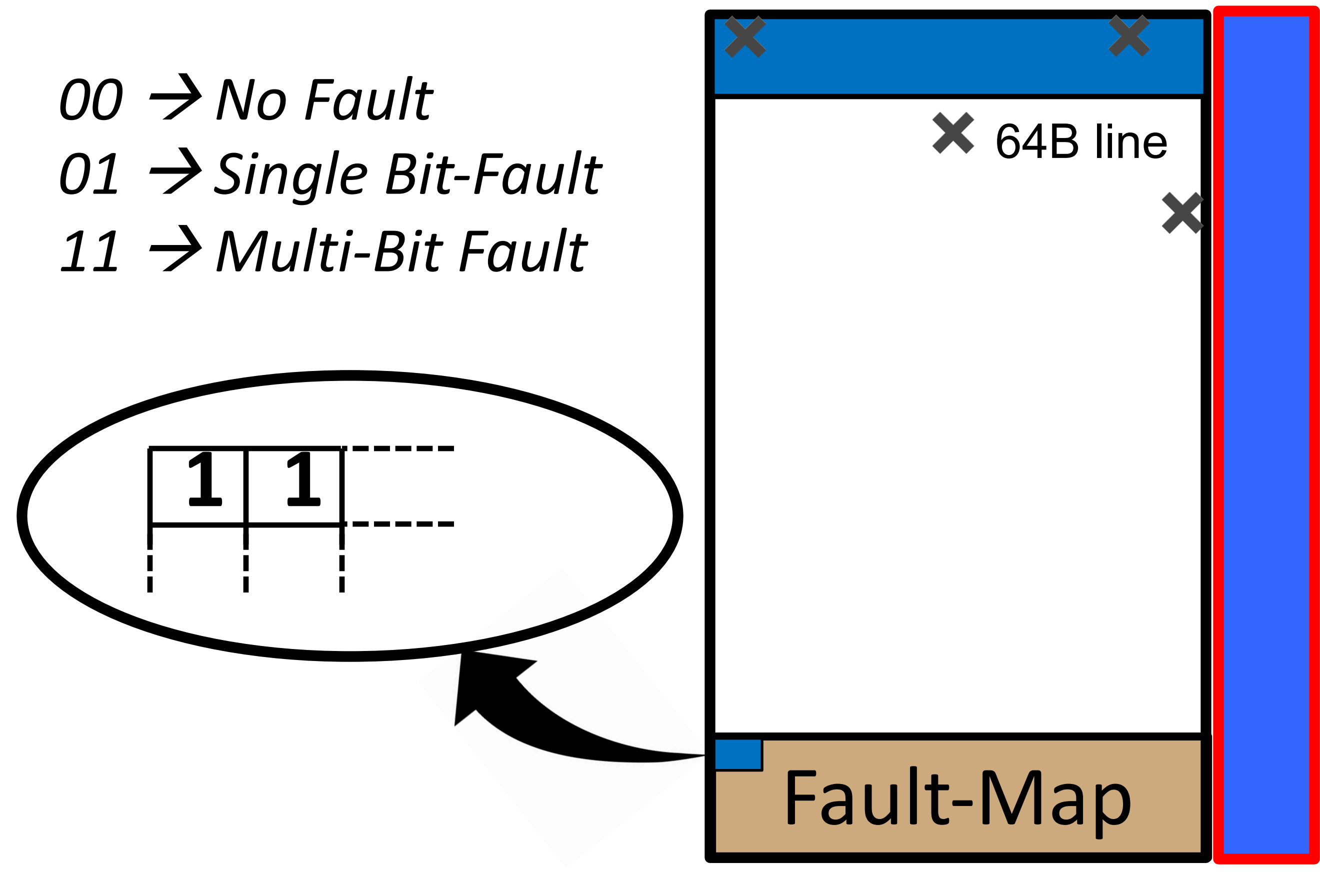


Per-word Fault-Map has an overhead of 6.4%

FAULT-MAP: INFORMATION AND OVERHEAD

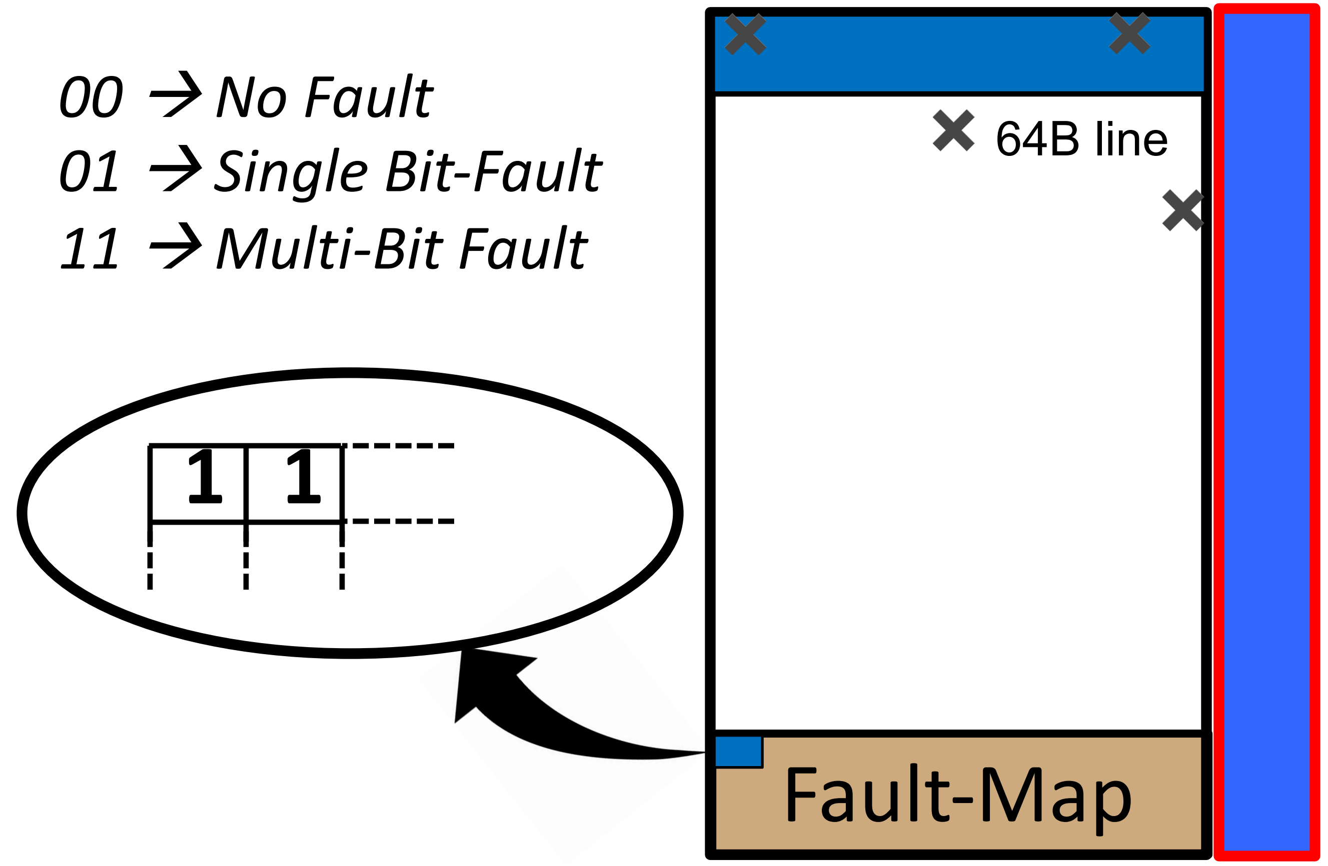
4bits (2-bits replicated) per **64B cacheline**

00 → No Fault
01 → Single Bit-Fault
11 → Multi-Bit Fault



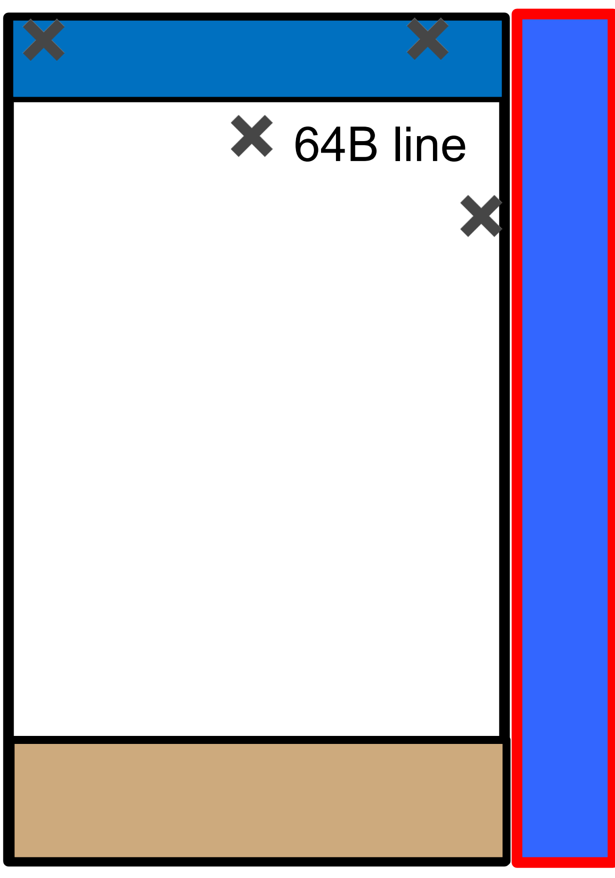
FAULT-MAP: INFORMATION AND OVERHEAD

4bits (2-bits replicated) per **64B cacheline**

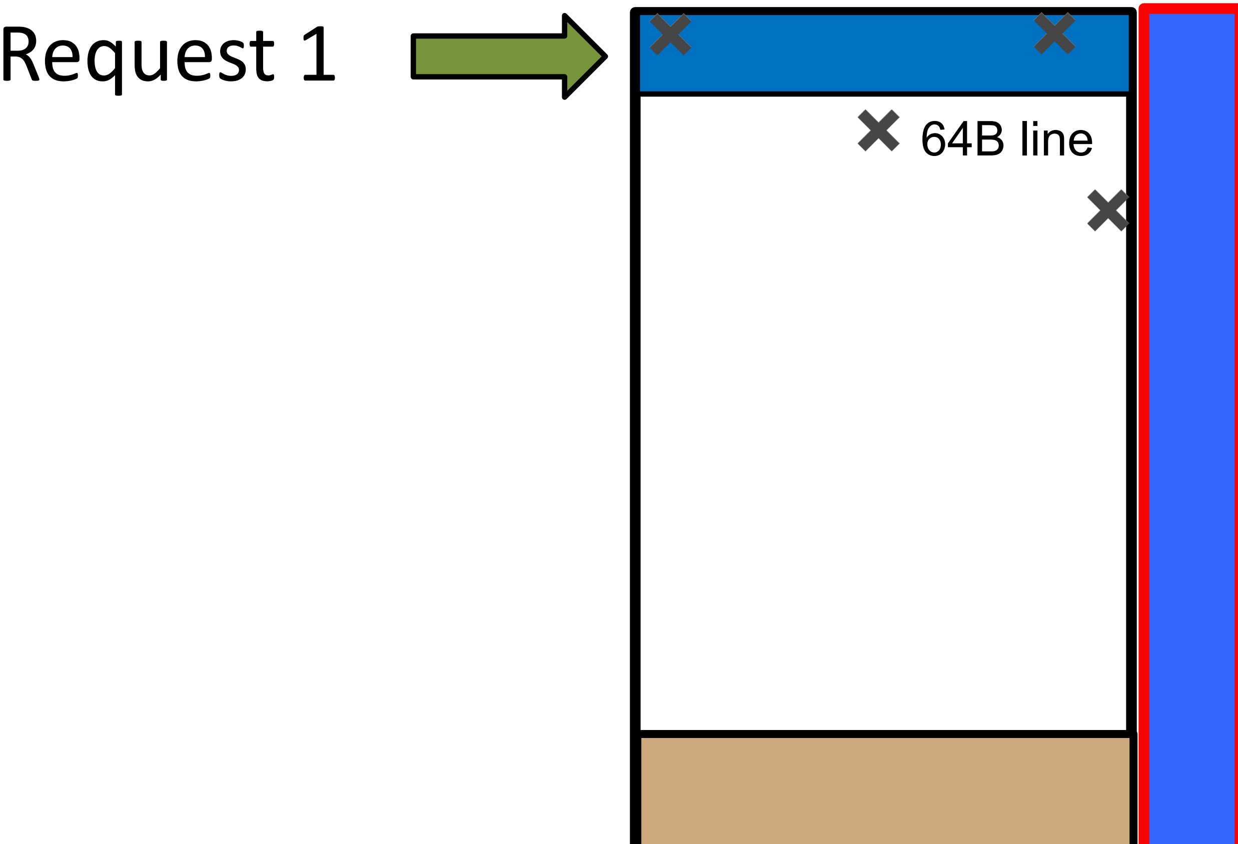


Per-line Fault-Map has an overhead of 0.8%

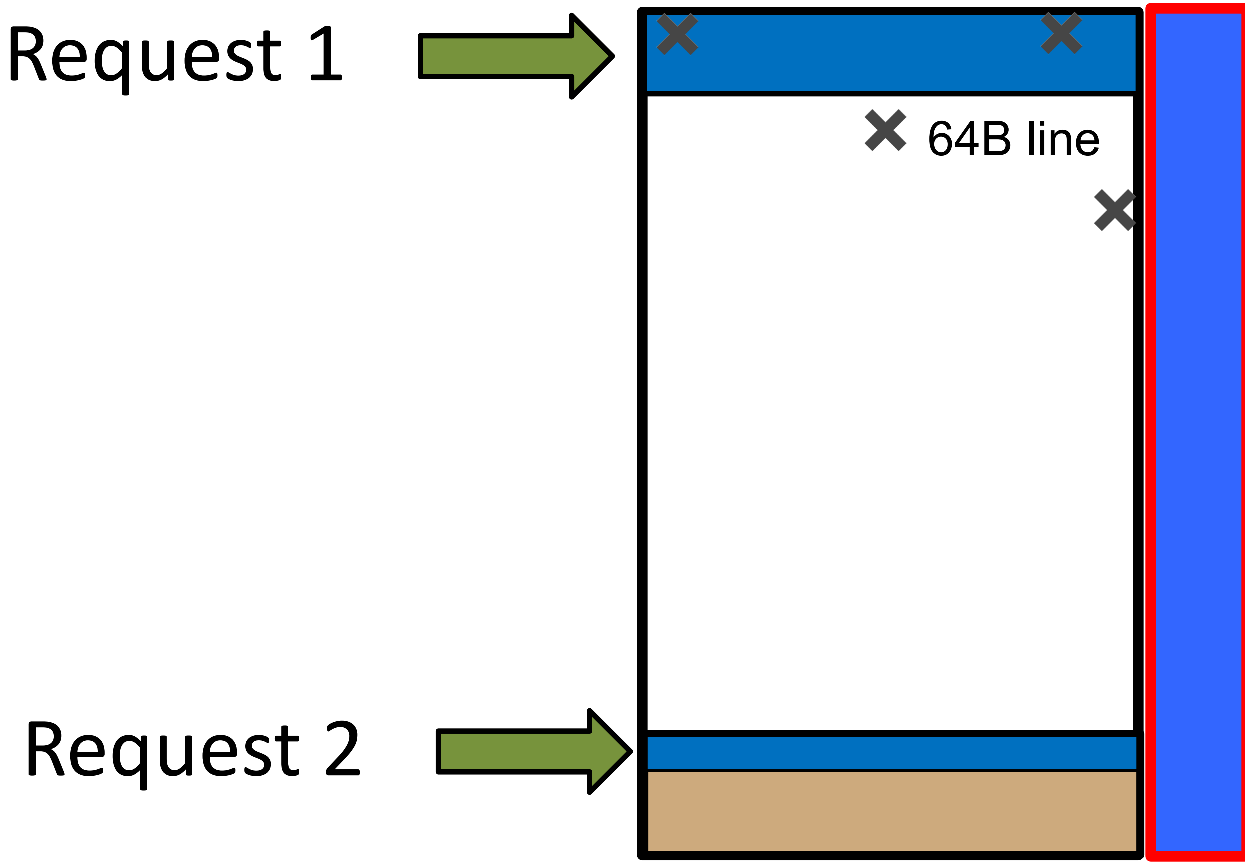
FAULT-MAP: OPERATION



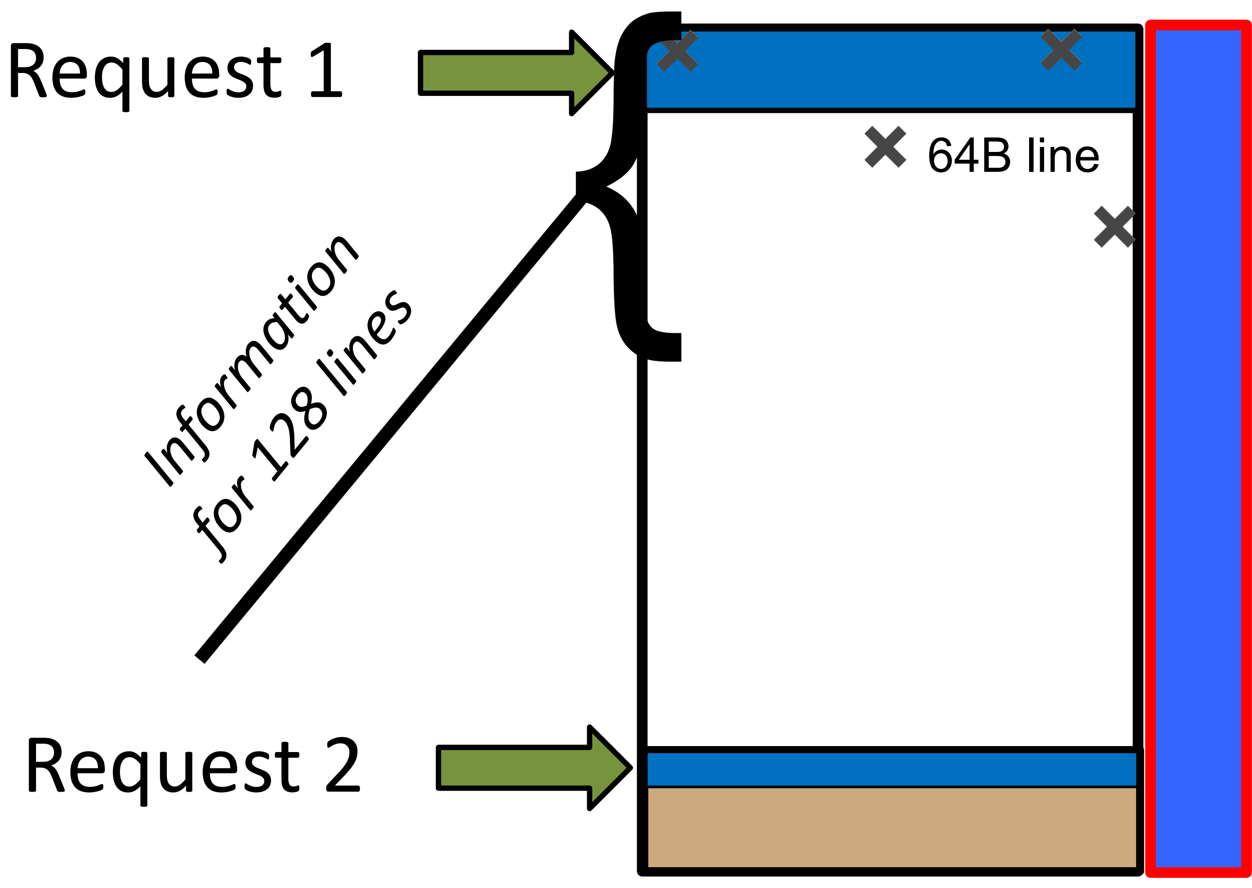
FAULT-MAP: OPERATION



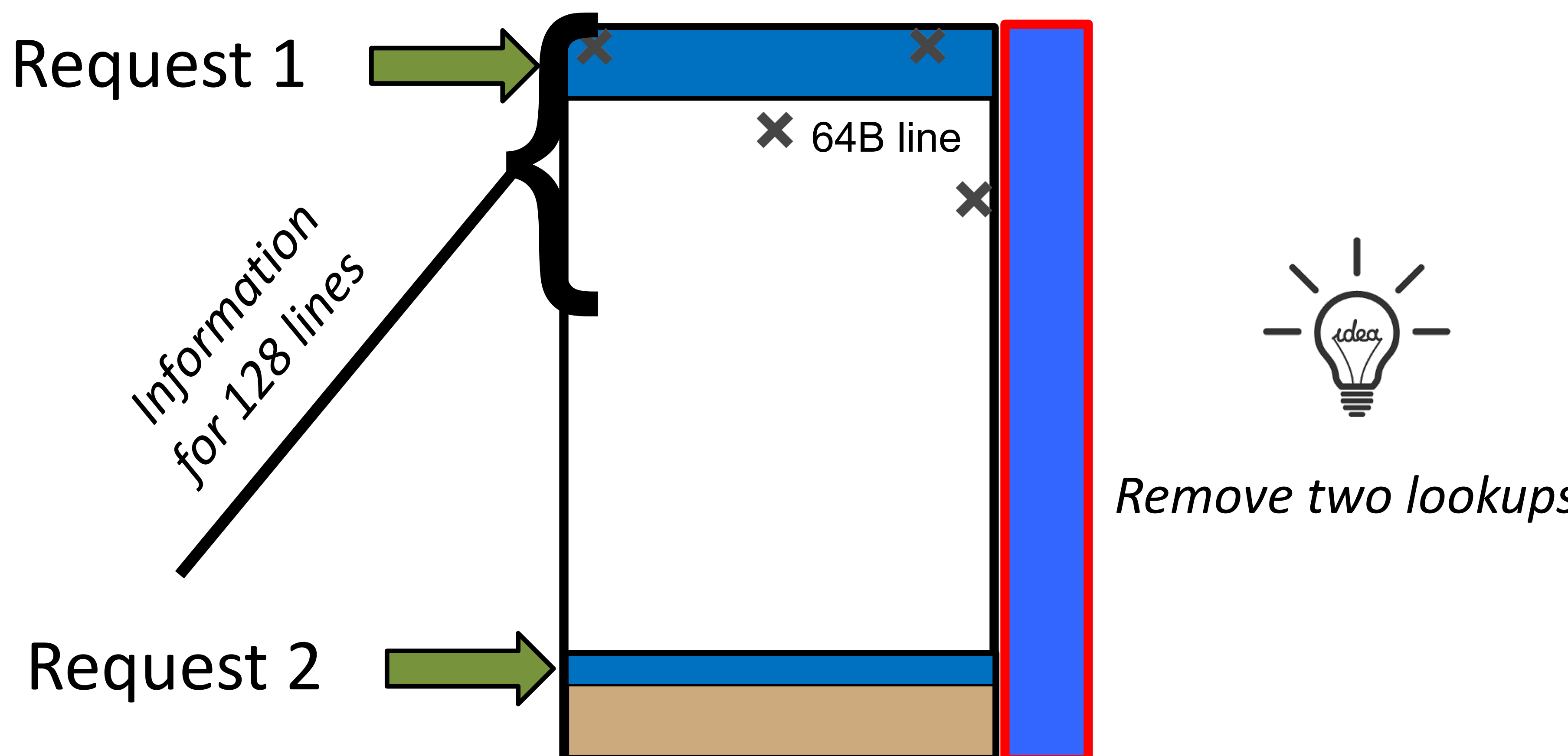
FAULT-MAP: OPERATION



FAULT-MAP: OPERATION

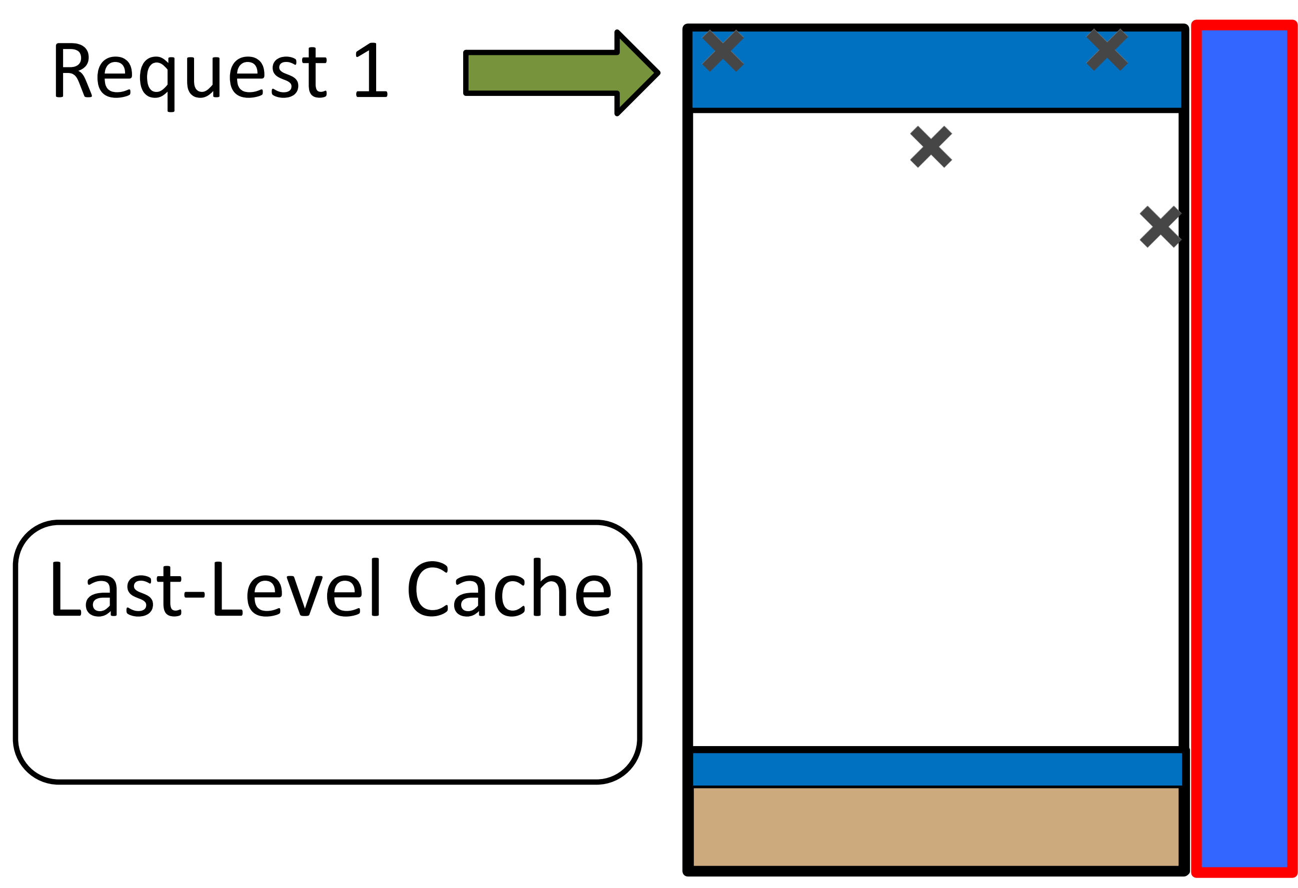


FAULT-MAP: OPERATION



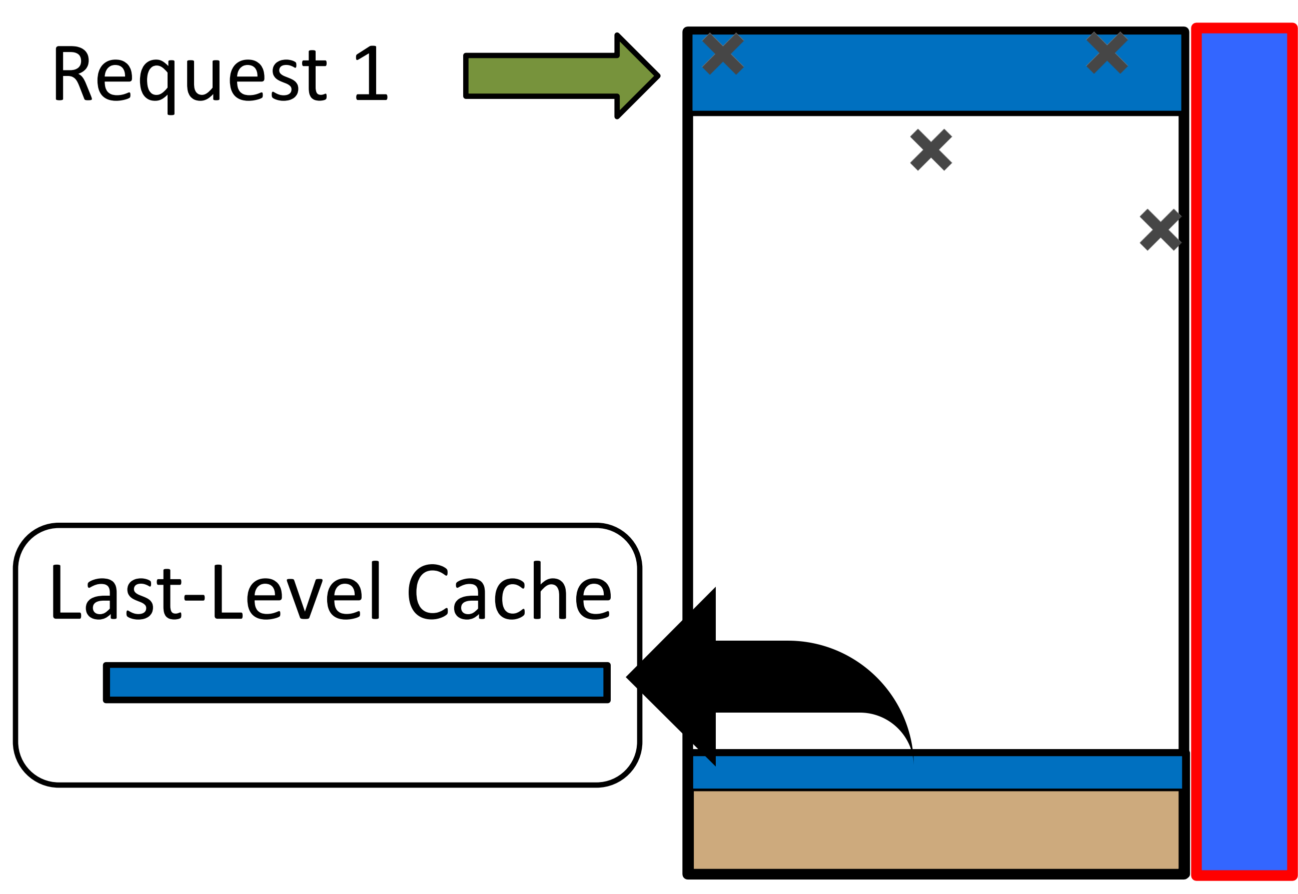
FAULT-MAP: OPERATION

Caching Fault-Map for Performance



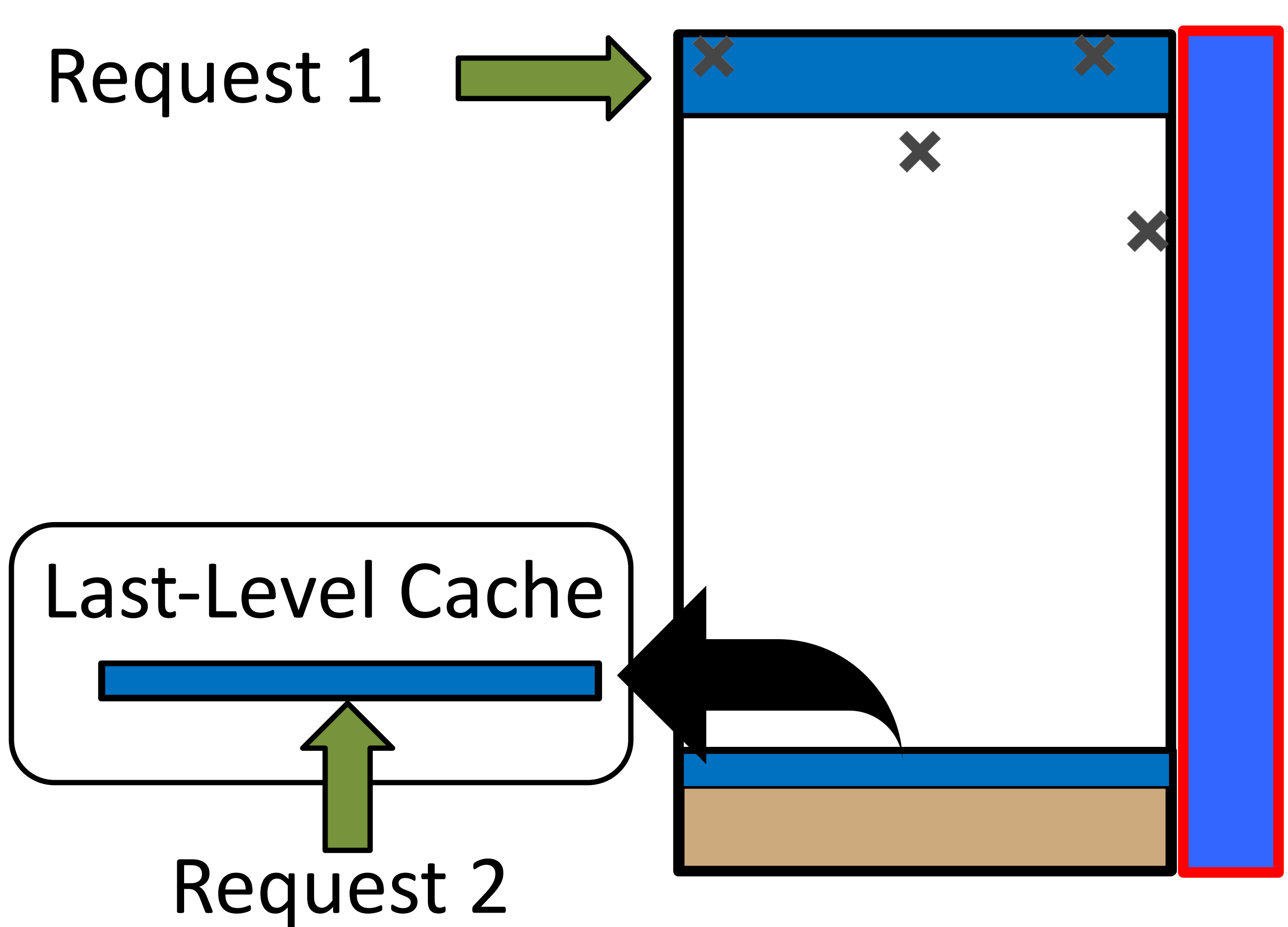
FAULT-MAP: OPERATION

Caching Fault-Map for Performance



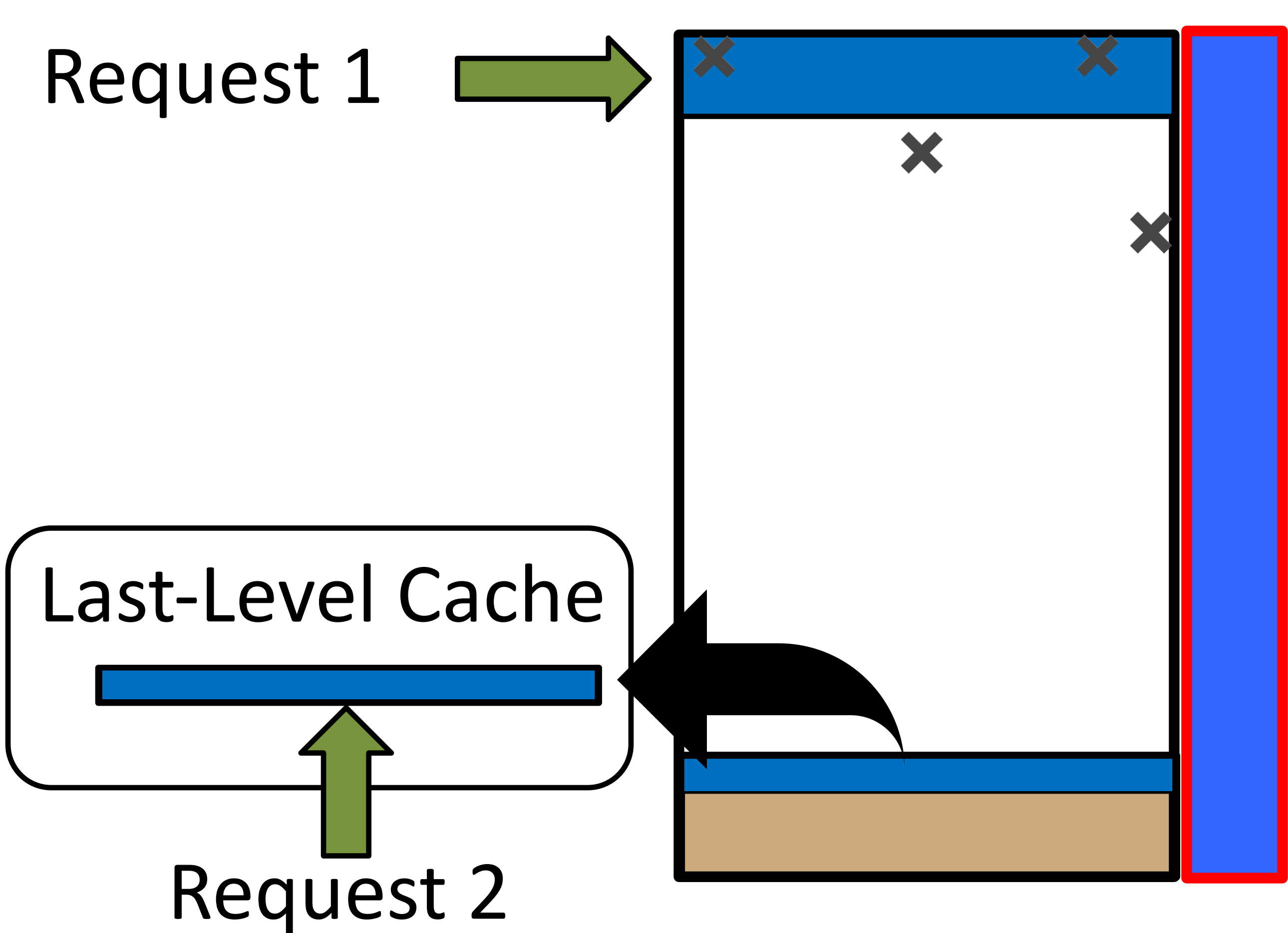
FAULT-MAP: OPERATION

Caching Fault-Map for Performance



FAULT-MAP: OPERATION

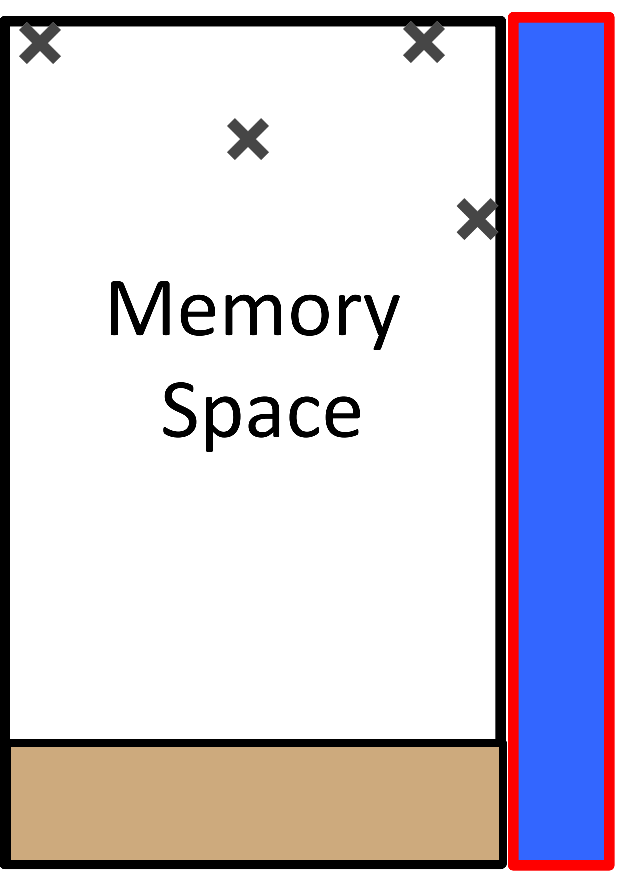
Caching Fault-Map for Performance



Cache Fault-Map line → Information of 128 lines

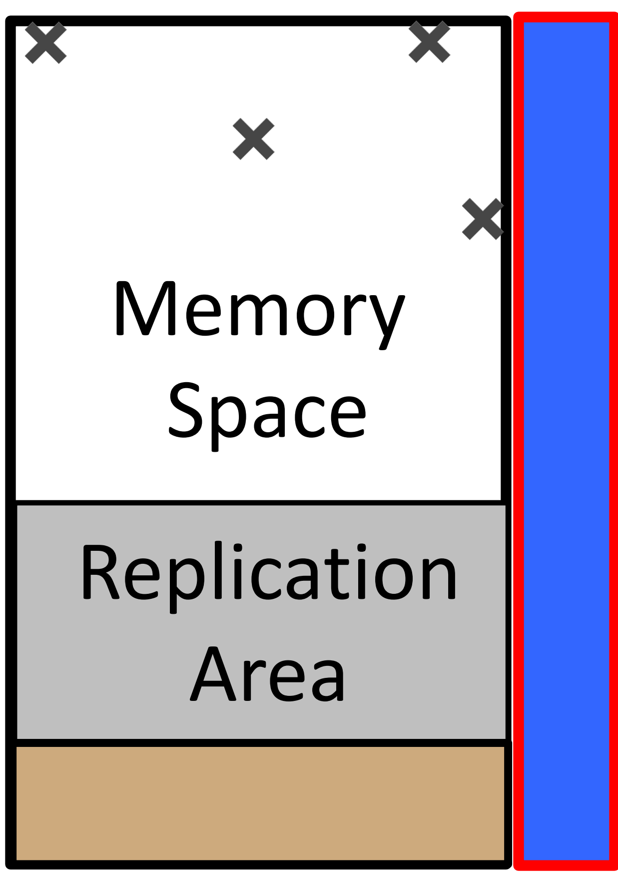
KEEP REPLICAS OF FAULTY WORDS

Valid data from faulty words are stored in replicas



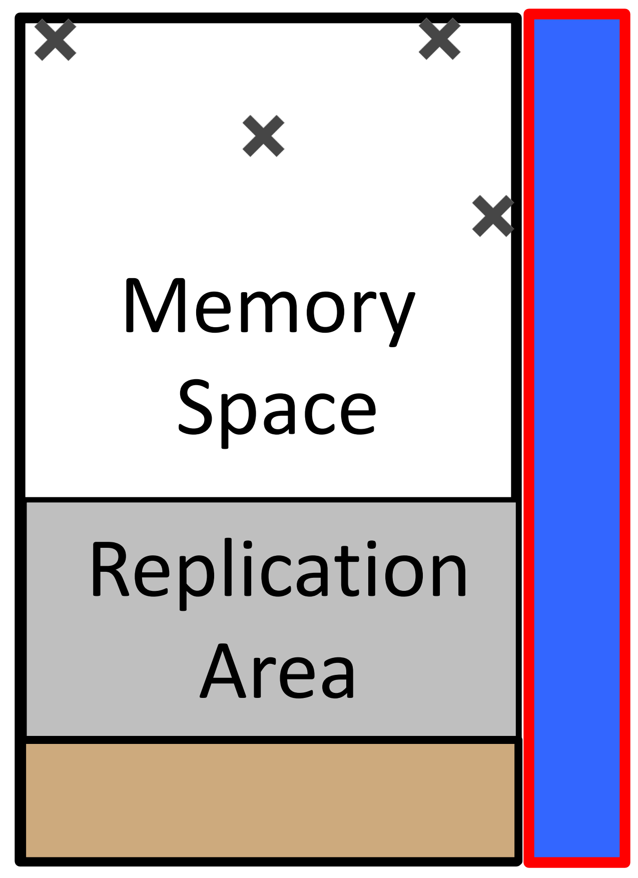
KEEP REPLICAS OF FAULTY WORDS

Valid data from faulty words are stored in replicas



KEEP REPLICAS OF FAULTY WORDS

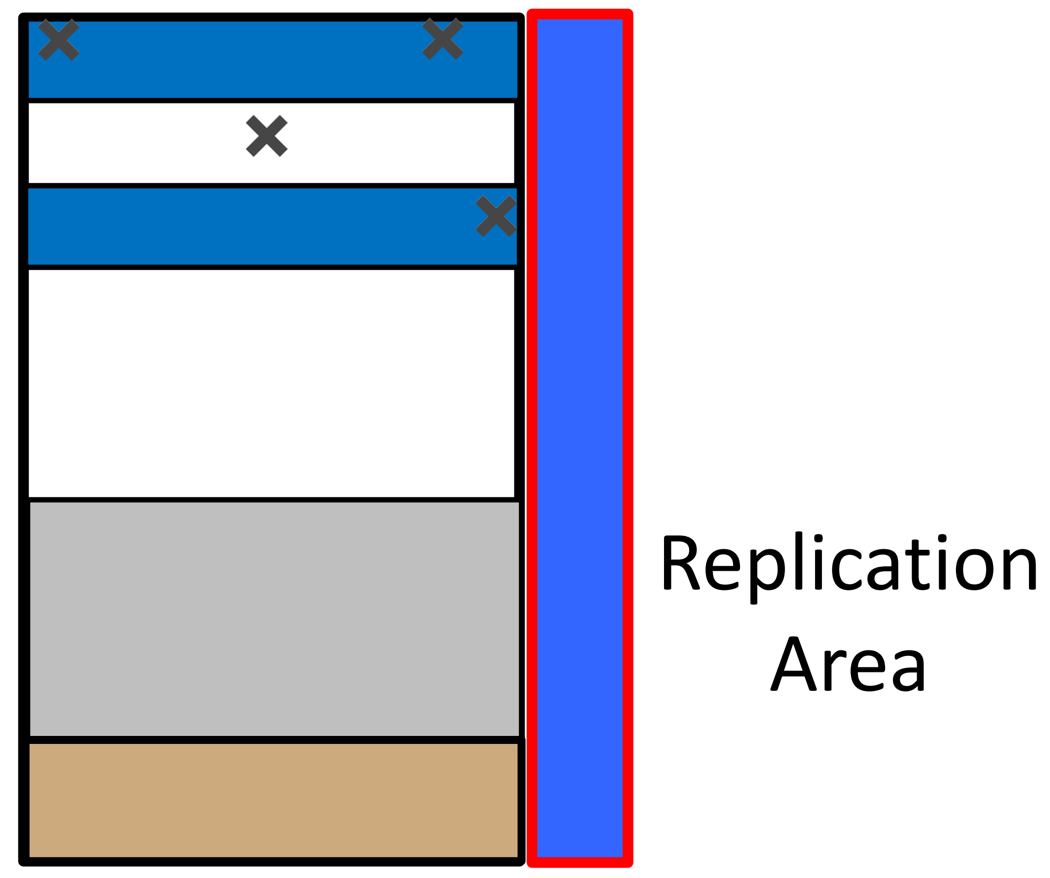
Valid data from faulty words are stored in replicas



Replication area stores valid data of faulty words

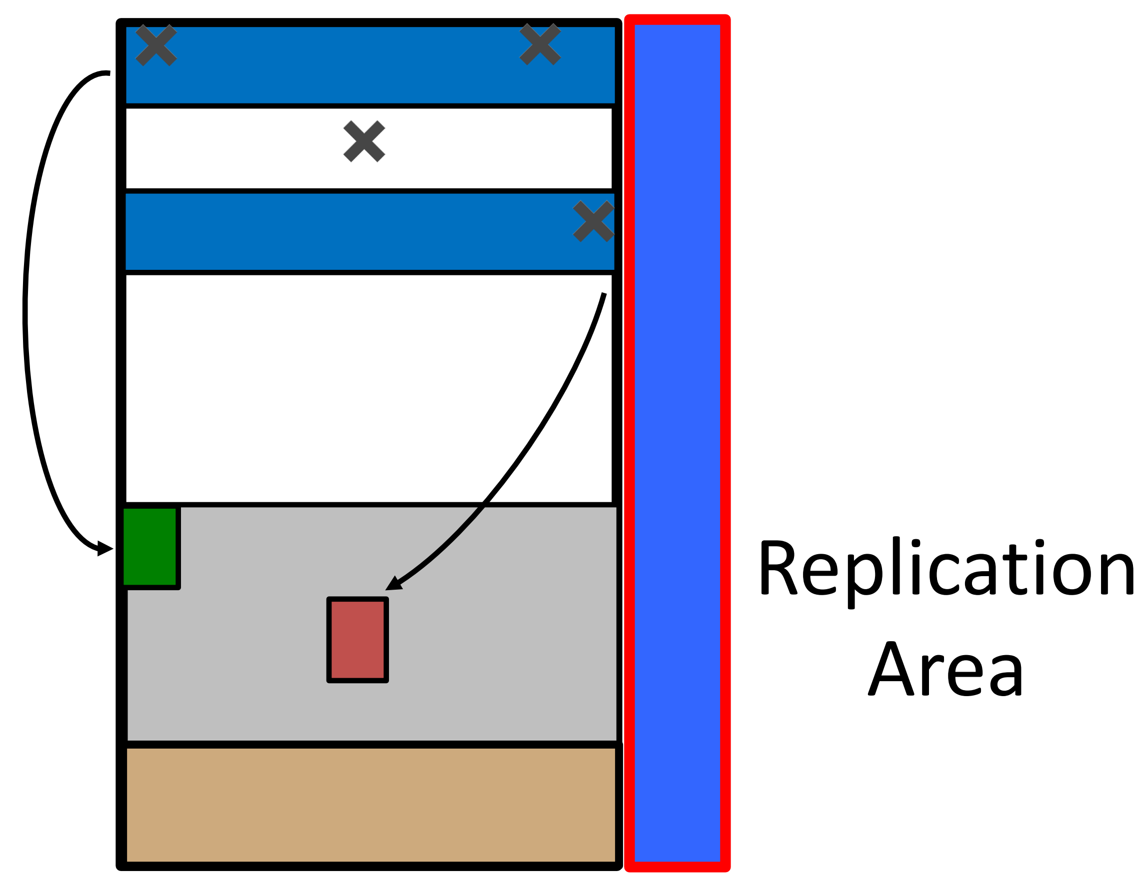
ARCHITECTING THE REPLICATION AREA

Replicas → Anywhere in a contiguous replication area



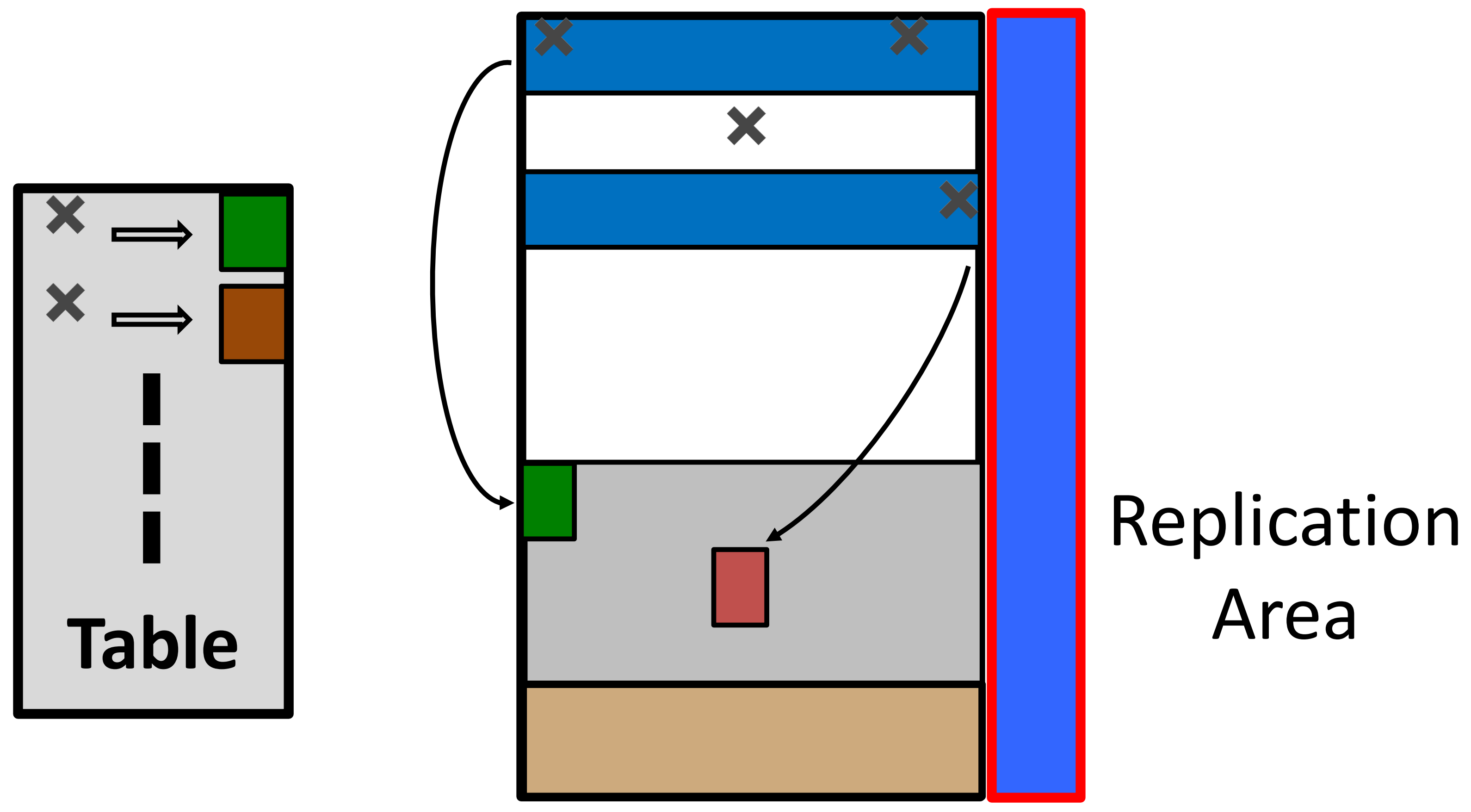
ARCHITECTING THE REPLICATION AREA

Replicas → Anywhere in a contiguous replication area



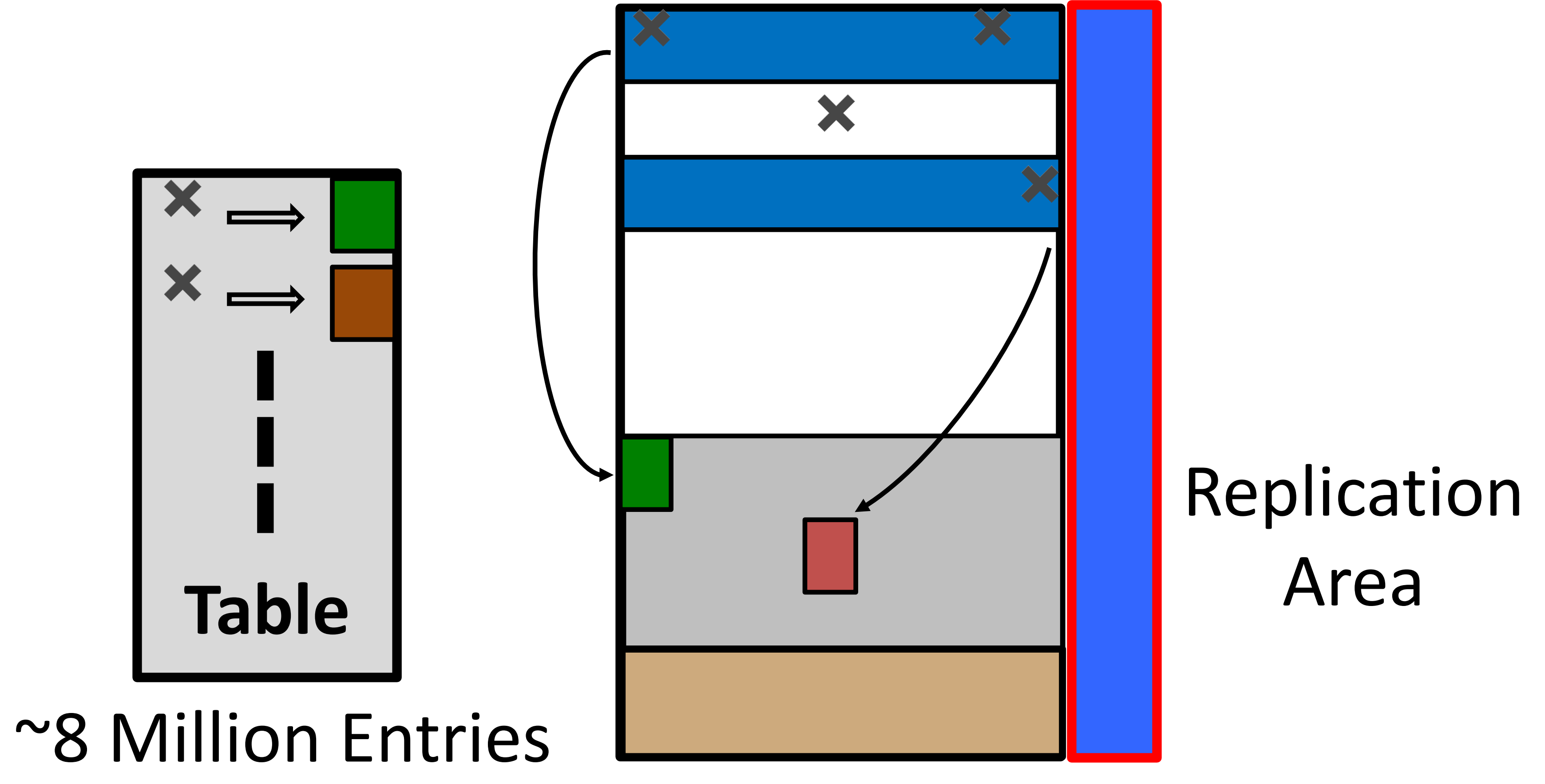
ARCHITECTING THE REPLICATION AREA

Replicas → Anywhere in a contiguous replication area



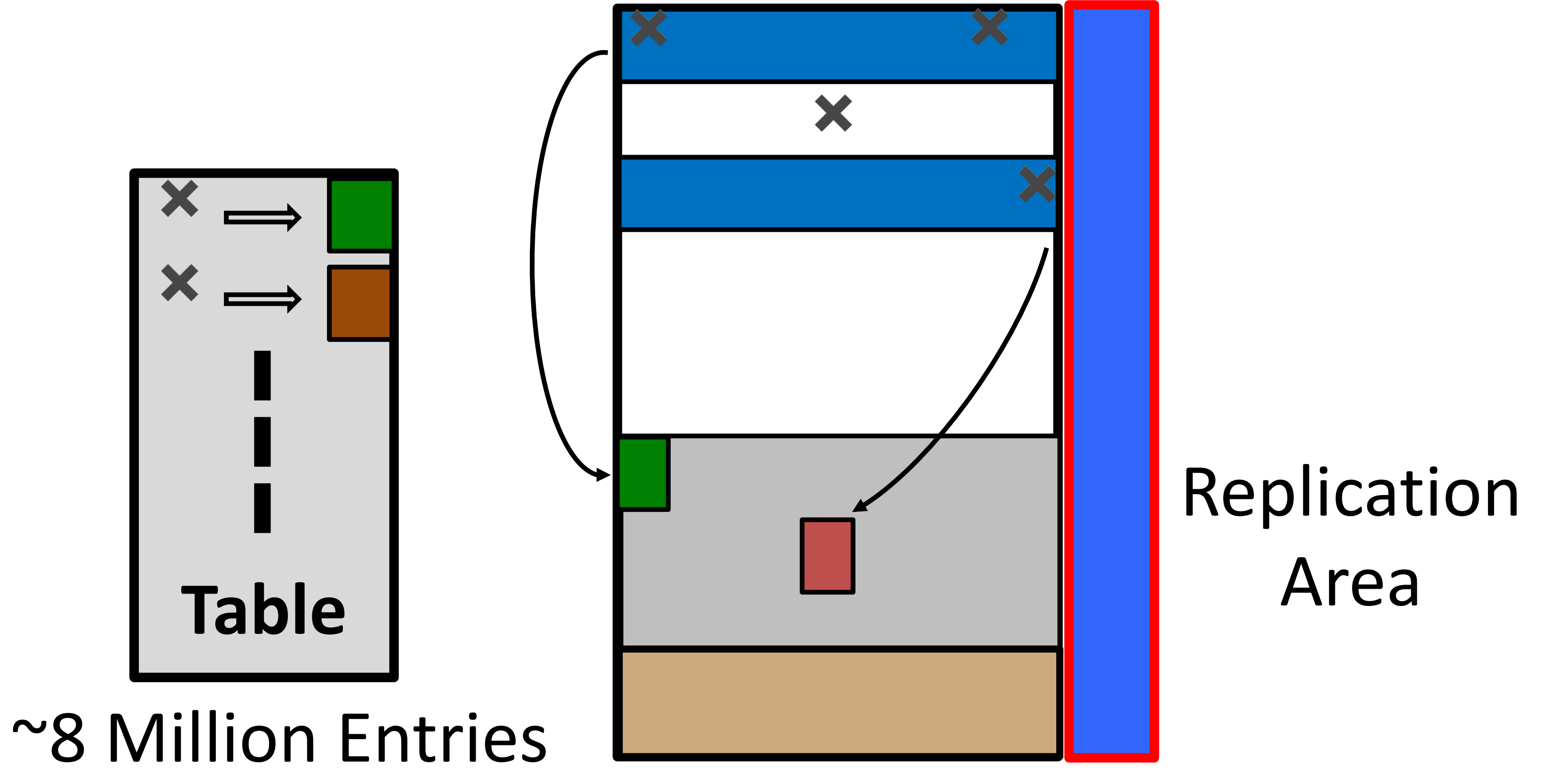
ARCHITECTING THE REPLICATION AREA

Replicas → Anywhere in a contiguous replication area



ARCHITECTING THE REPLICATION AREA

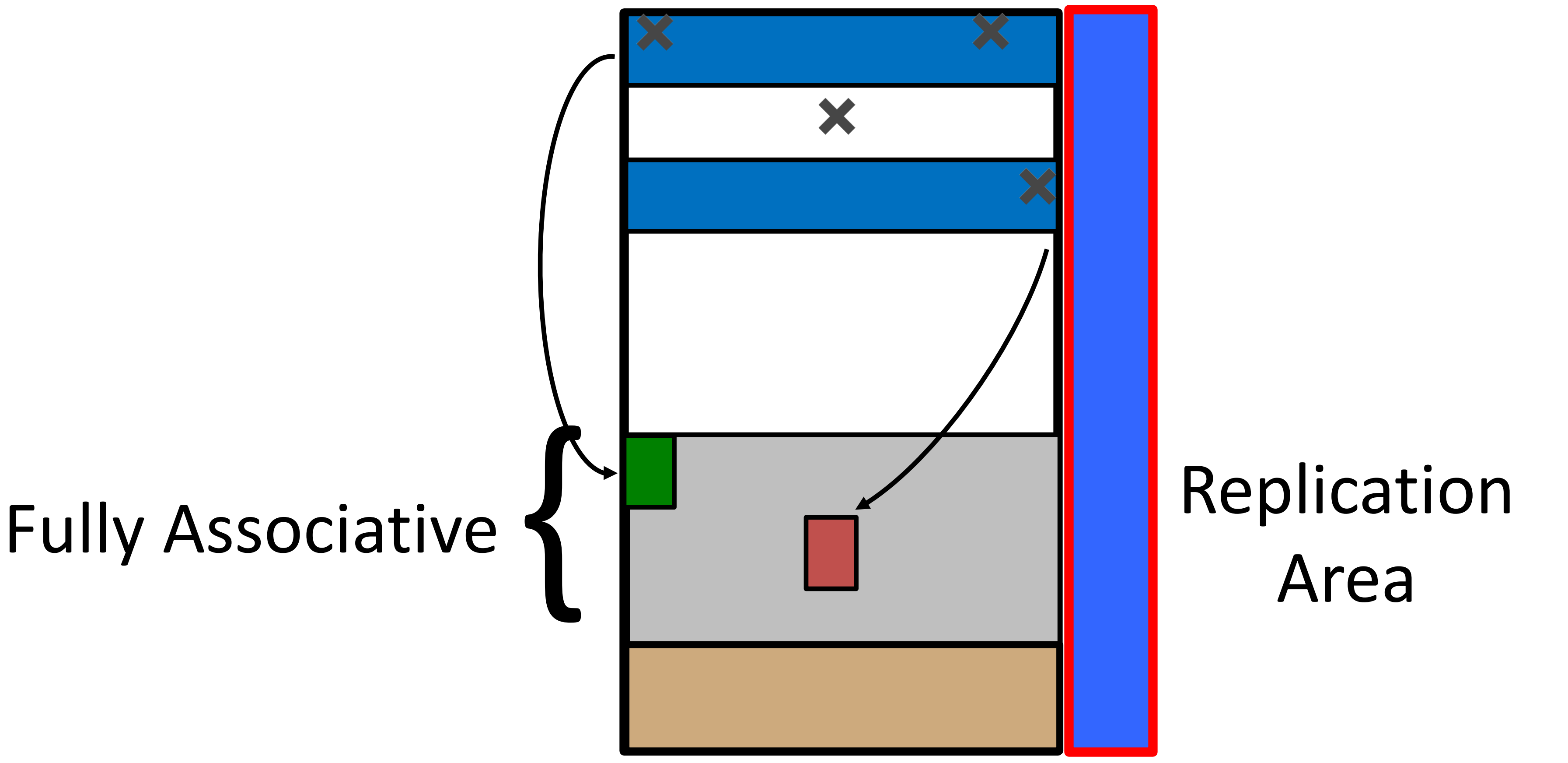
Replicas → Anywhere in a contiguous replication area



Looking up a Large Table → High Latency

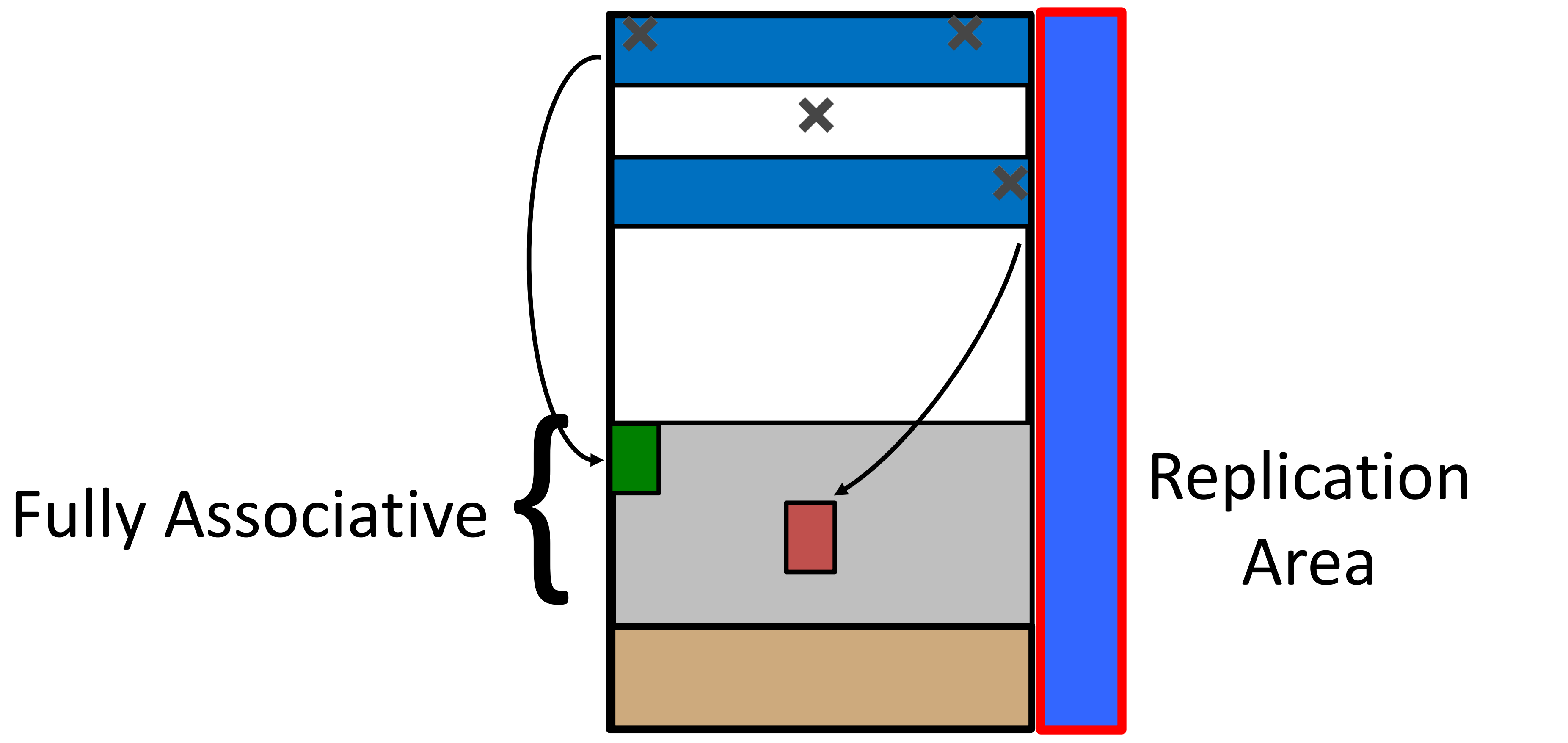
ARCHITECTING THE REPLICATION AREA

Replicas → Anywhere in a contiguous replication area



ARCHITECTING THE REPLICATION AREA

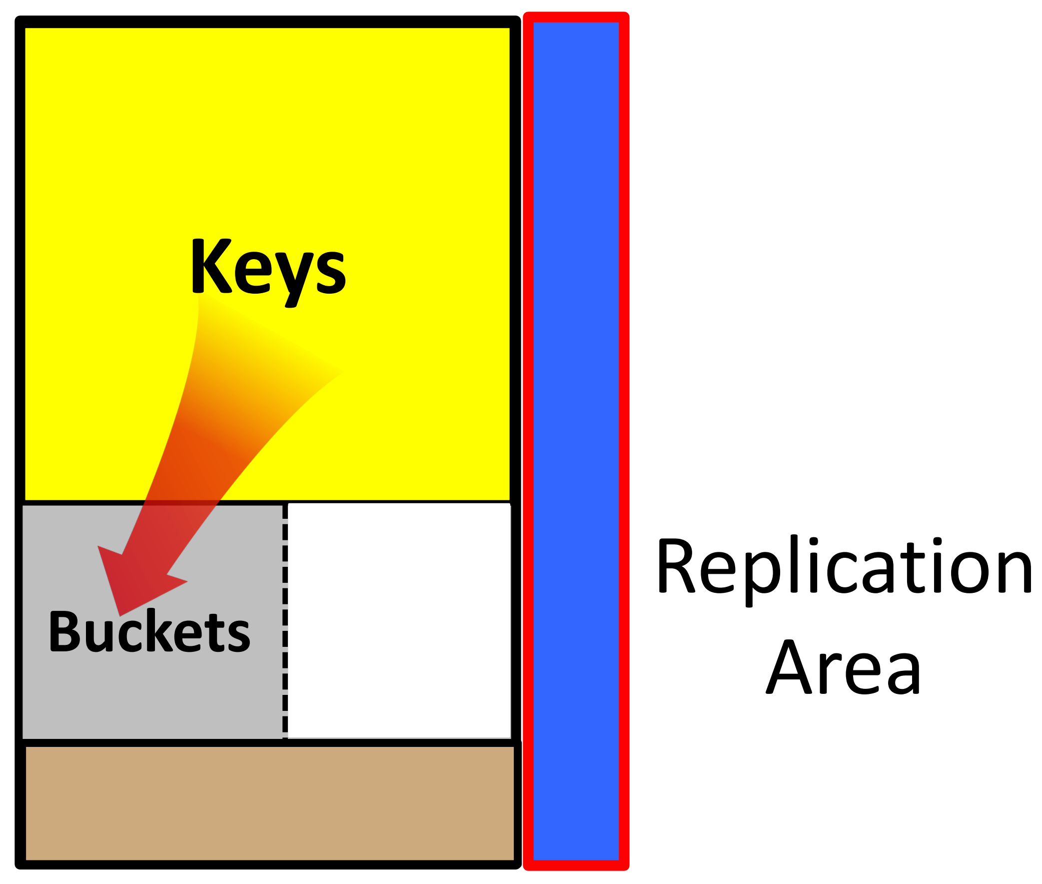
Replicas → Anywhere in a contiguous replication area



Looking up the Replication Area → High Latency

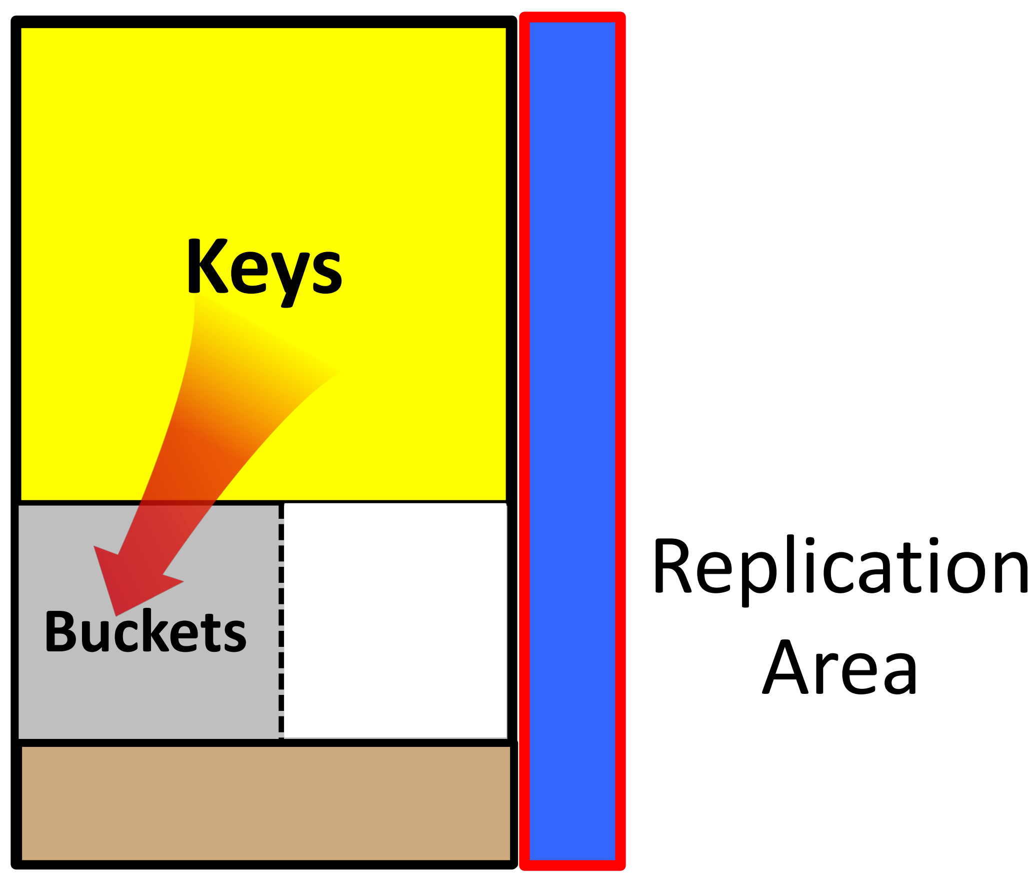
ARCHITECTING THE REPLICATION AREA

Taking inspiration from Hash-Tables



ARCHITECTING THE REPLICATION AREA

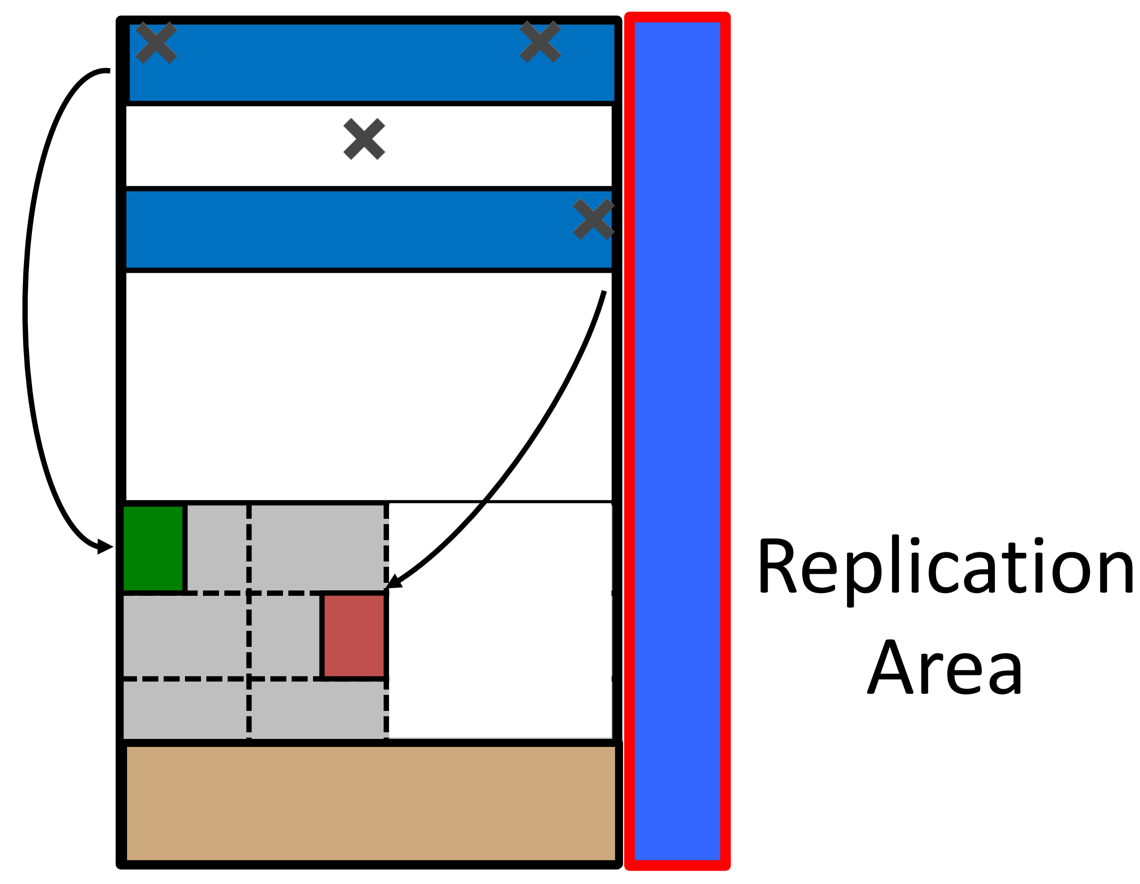
Taking inspiration from Hash-Tables



Hash-Table Lookup → Low Latency

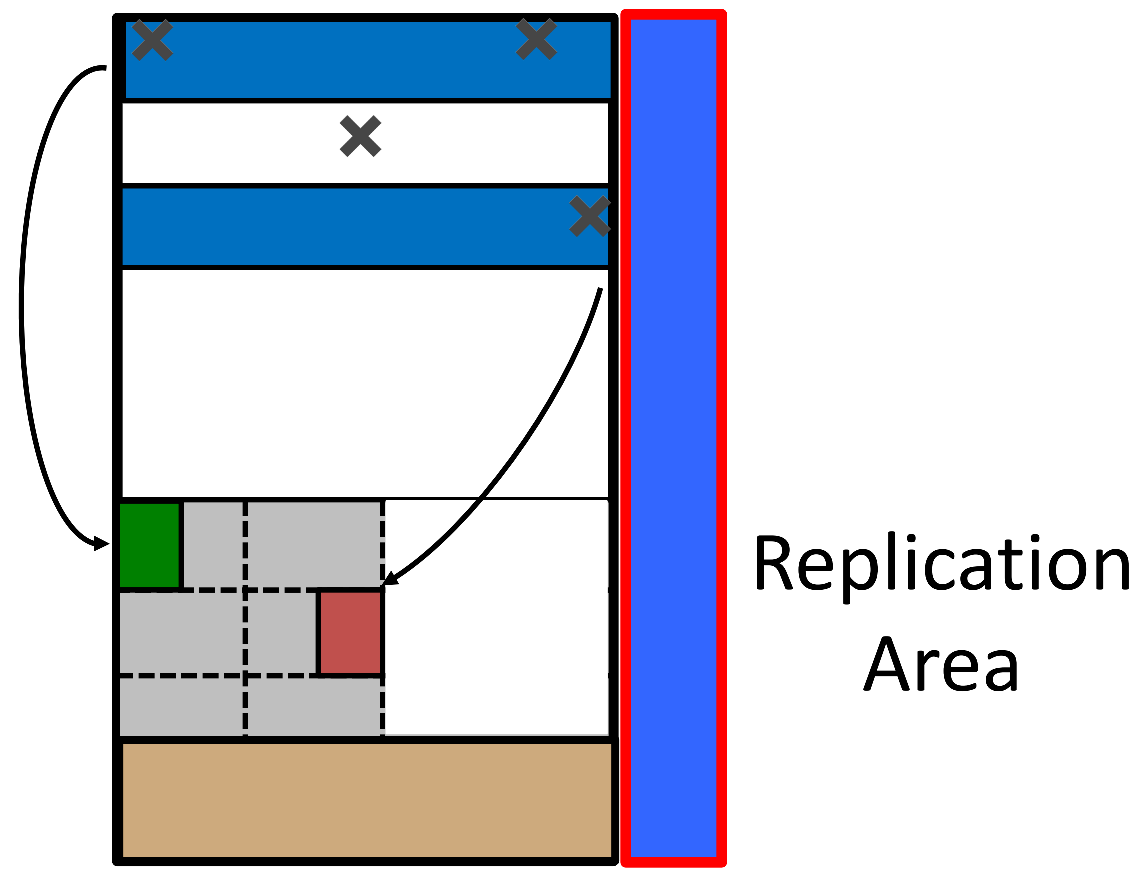
ARCHITECTING THE REPLICATION AREA

A Set Associative Area (Like a Hash Table)



ARCHITECTING THE REPLICATION AREA

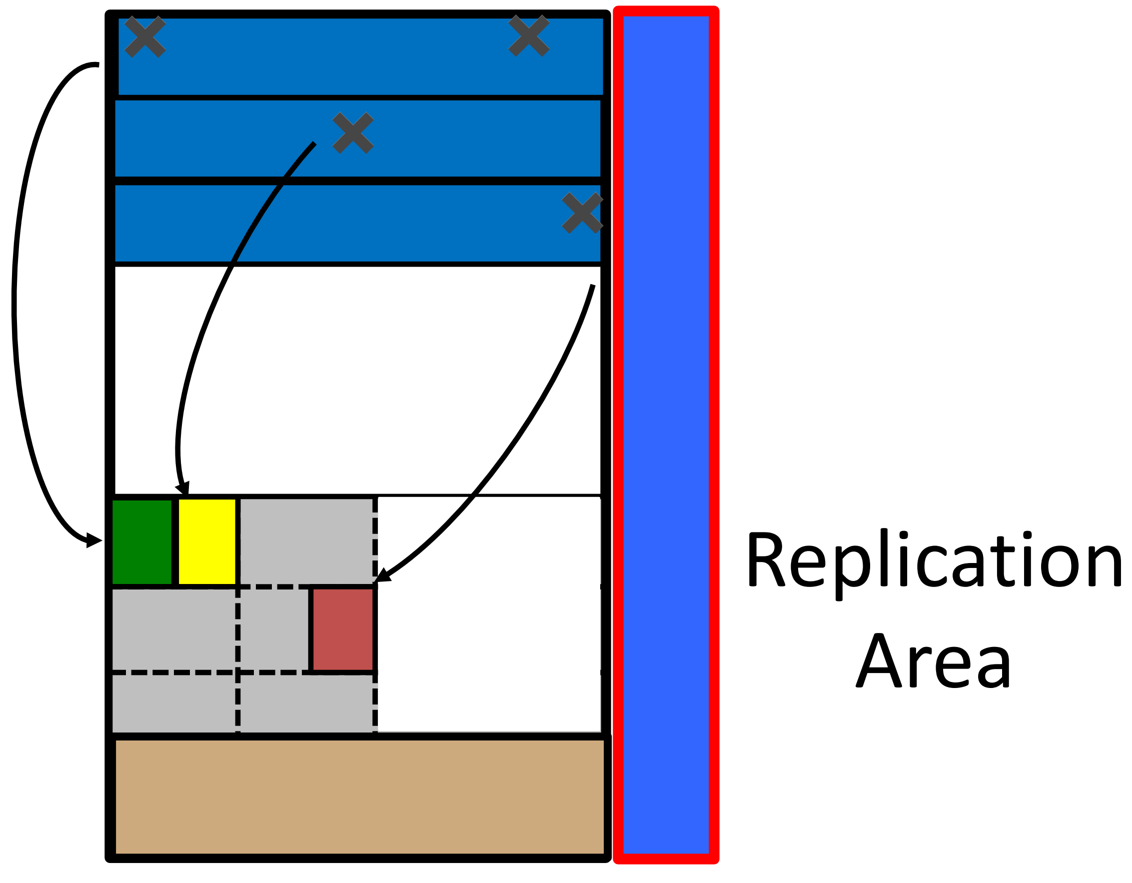
A Set Associative Area (Like a Hash Table)



Set-Associative Structure → Low Latency

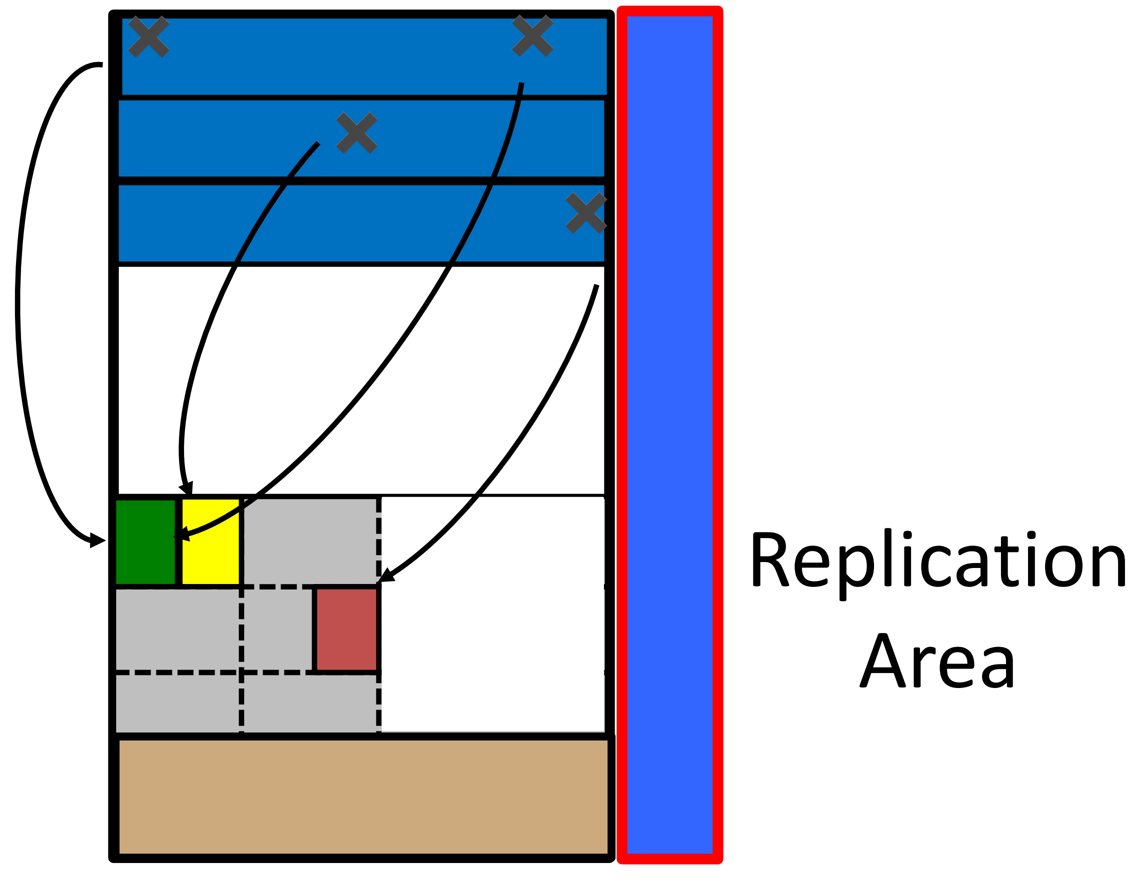
ARCHITECTING THE REPLICATION AREA

Set Associative: May not handle all faulty words



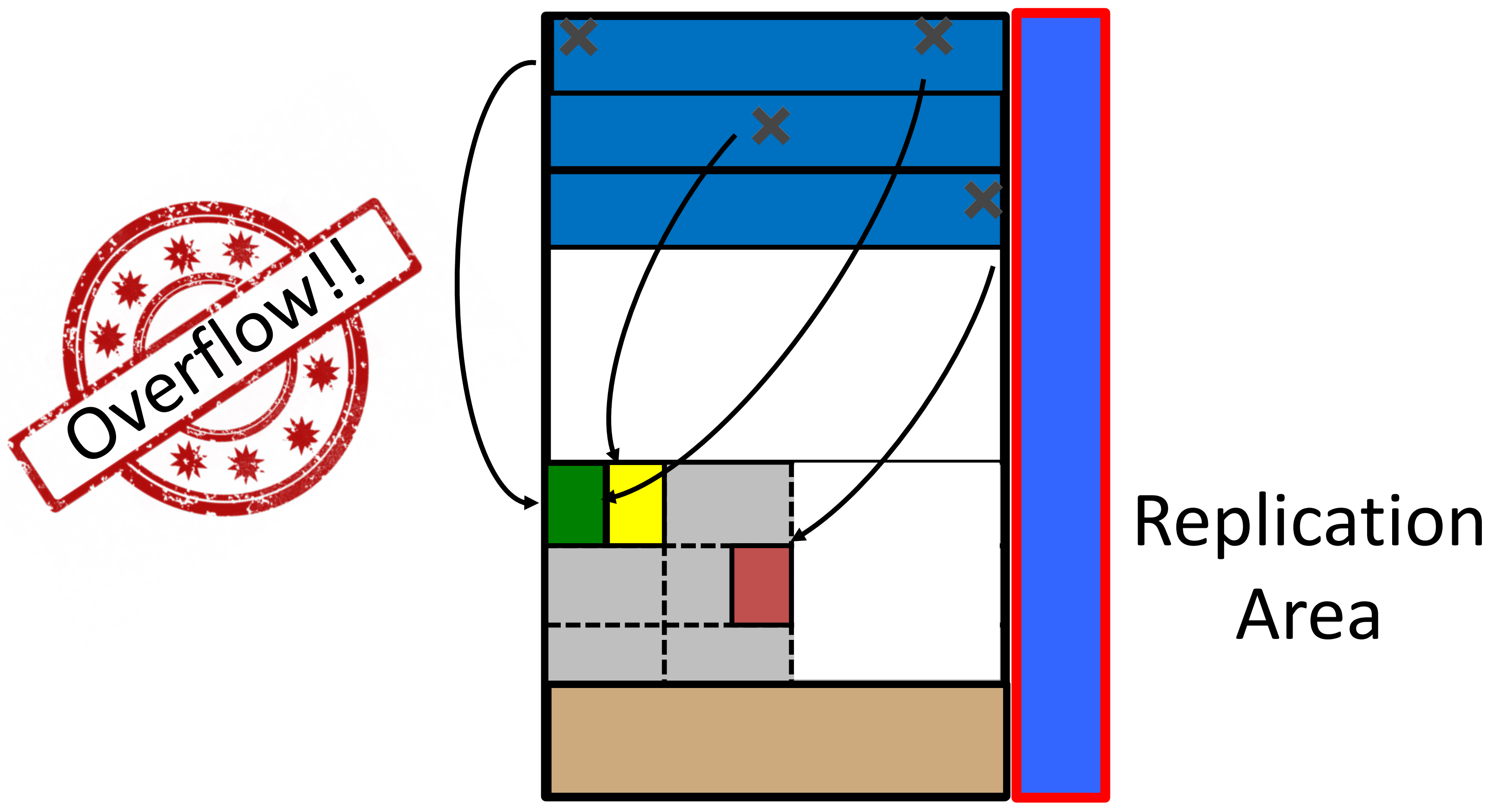
ARCHITECTING THE REPLICATION AREA

Set Associative: May not handle all faulty words



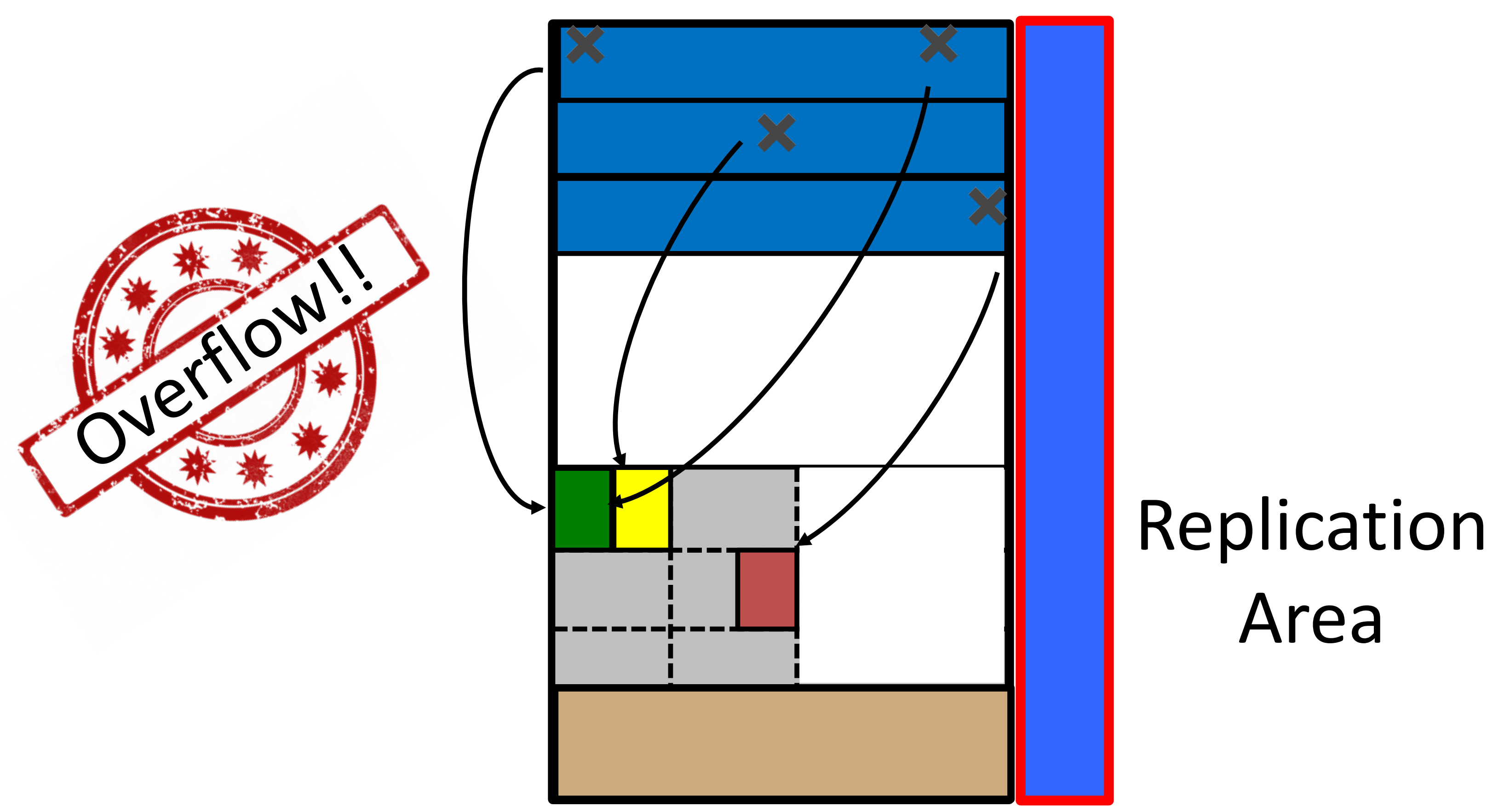
ARCHITECTING THE REPLICATION AREA

Set Associative: May not handle all faulty words



ARCHITECTING THE REPLICATION AREA

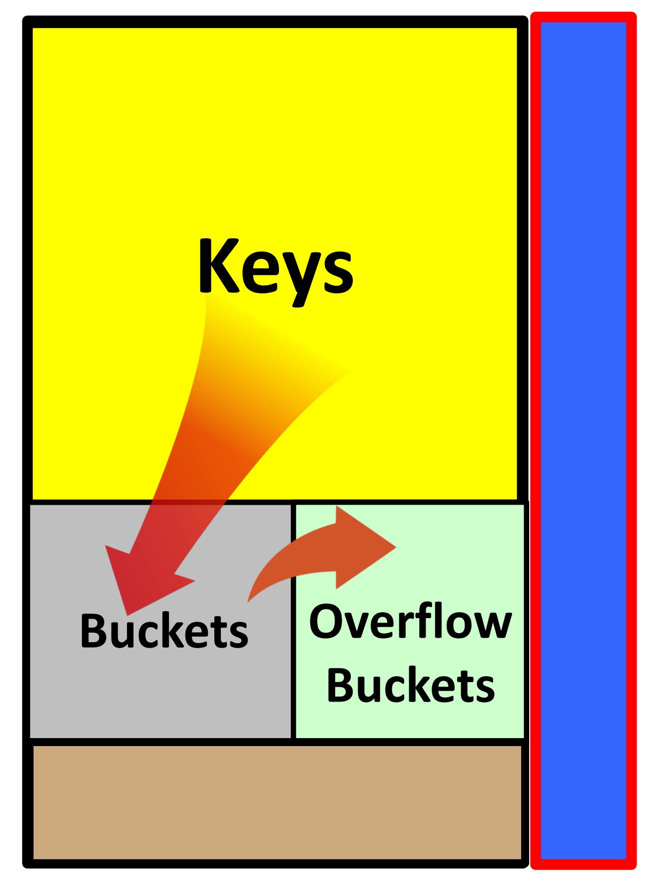
Set Associative: May not handle all faulty words



Set-Associative Structure → Can Overflow

ARCHITECTING THE REPLICATION AREA

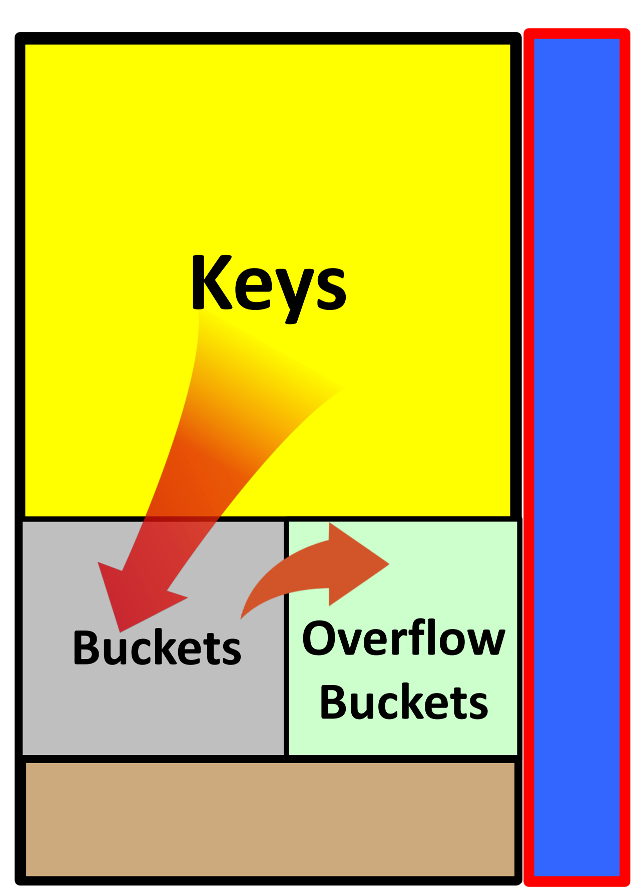
Taking inspiration from Hash-Tables with Chaining



Replication Area

ARCHITECTING THE REPLICATION AREA

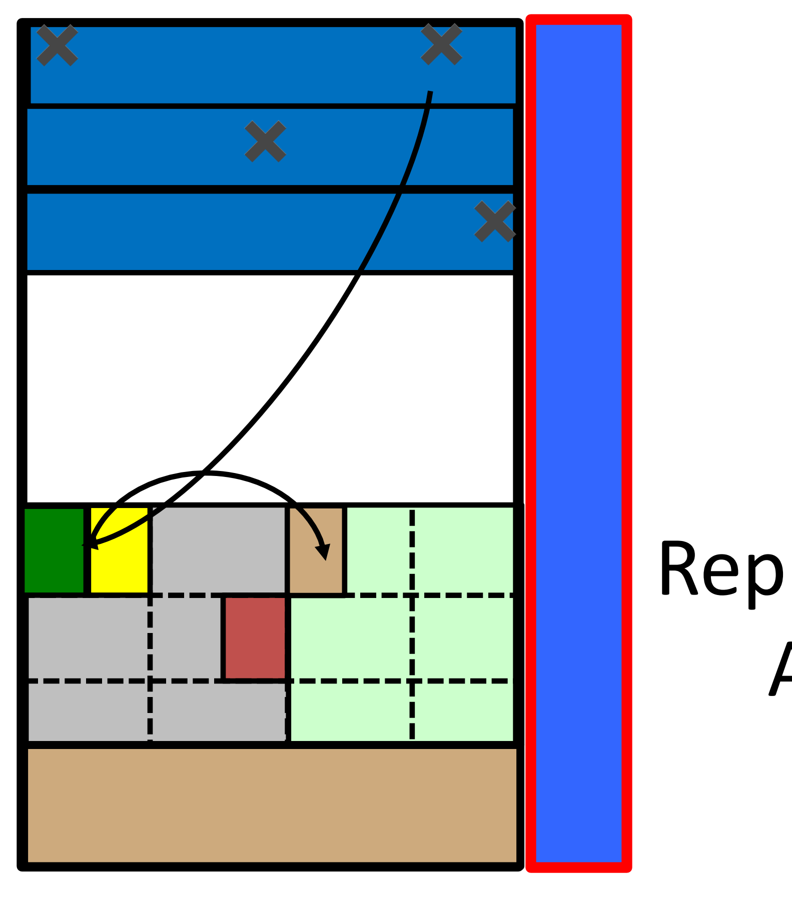
Taking inspiration from Hash-Tables with Chaining



Replication Area

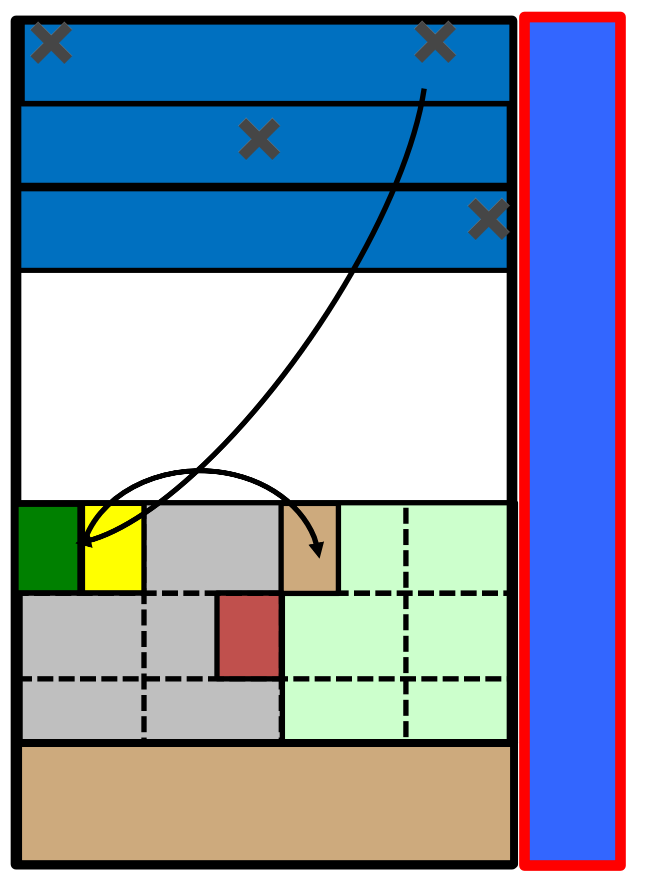
Hash-Table with Chaining → Mitigates Overflows

HANDLE SKEWS IN THE REPLICATION AREA



HANDLE SKEWS IN THE REPLICATION AREA

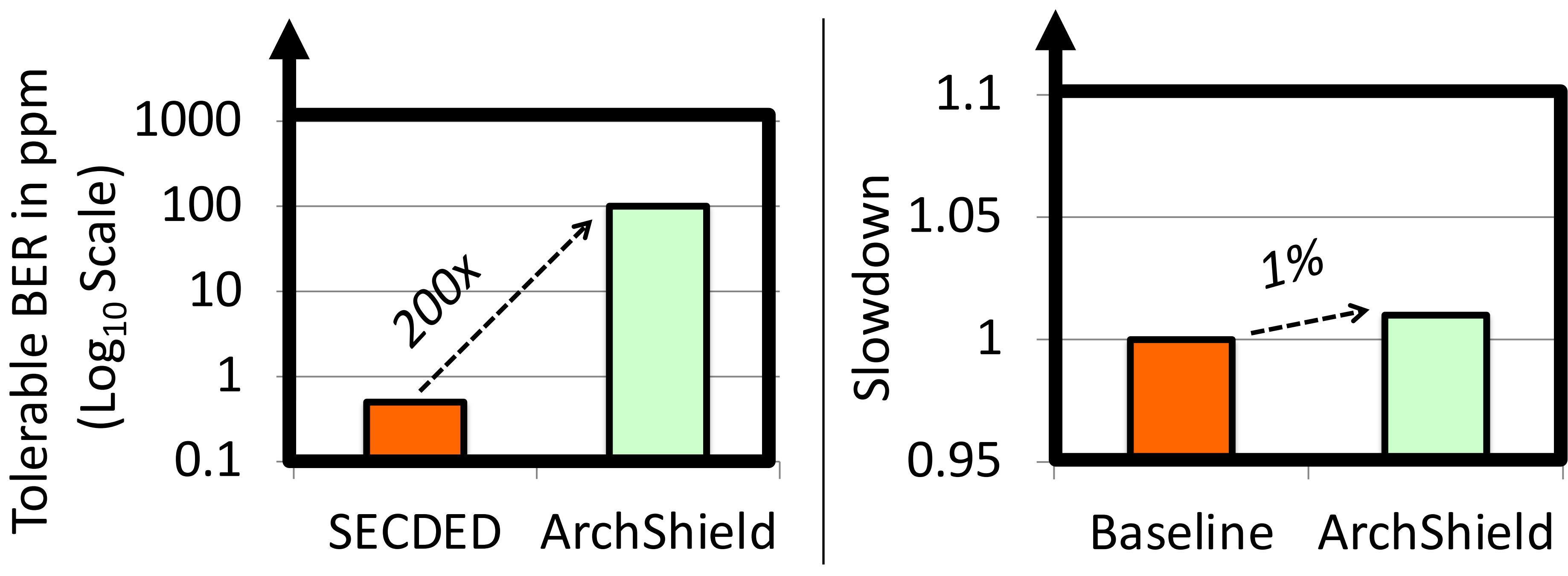
Provision Overflow Sets



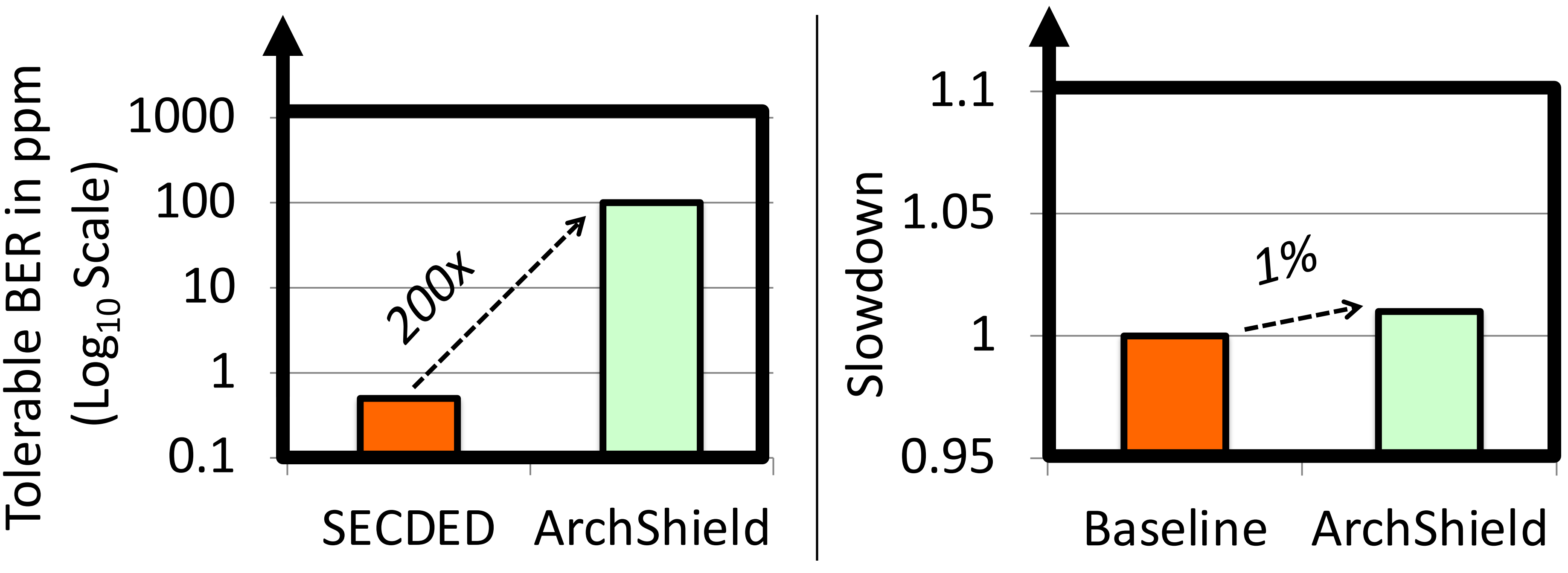
Replication Area

Overflow Sets handle skews in the Replication Area

ARCHSHIELD: RESULTS

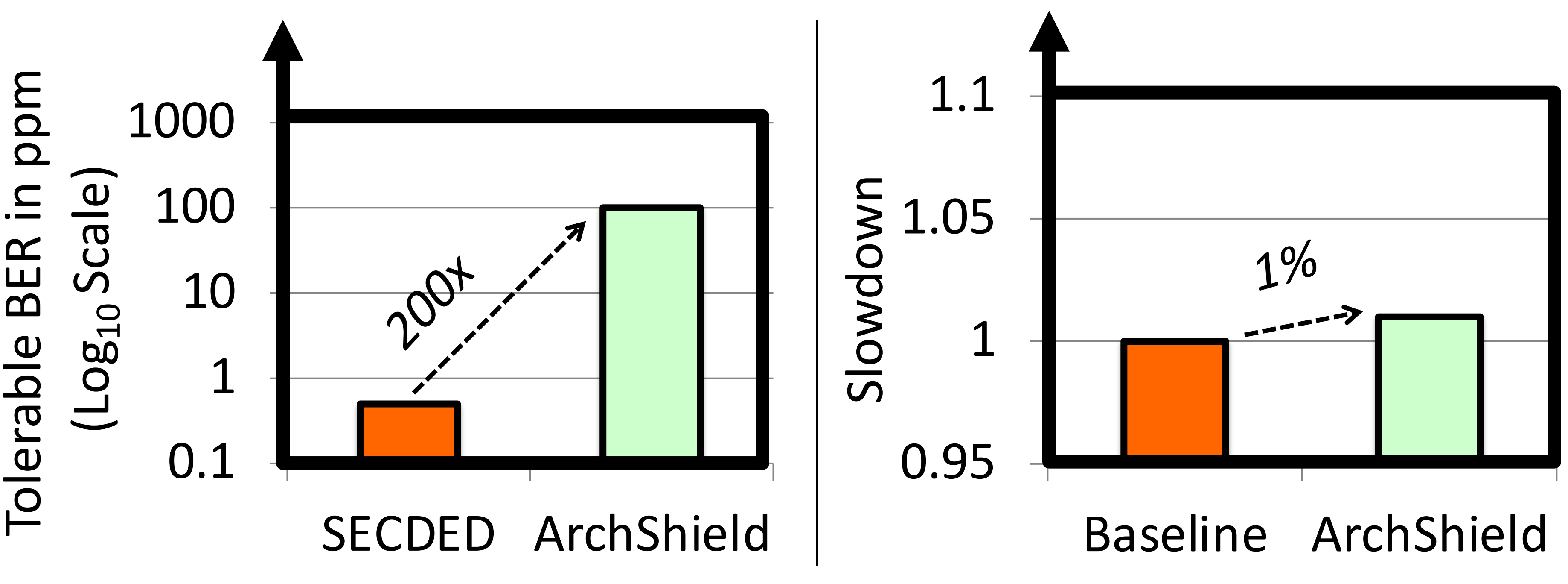


ARCHSHIELD: RESULTS



	Replication Area	*Fault-Map*
Area Overheads	3.2%	0.8%

ARCHSHIELD: RESULTS



	Replication Area	*Fault-Map*
Area Overheads	3.2%	0.8%

Tolerates 200x higher BER with only 1% slowdown

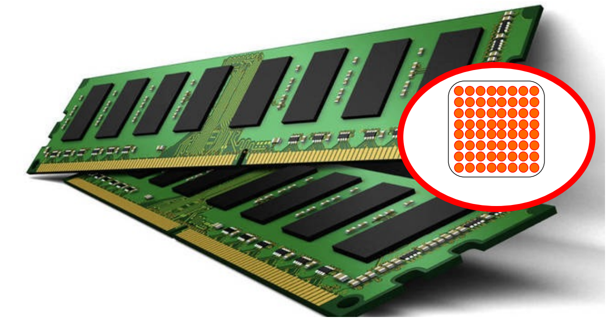
XED: Exposing On-Die Error Detection Information for Strong Memory Reliability

ISCA-2016

Prashant Nair

Vilas Sridharan

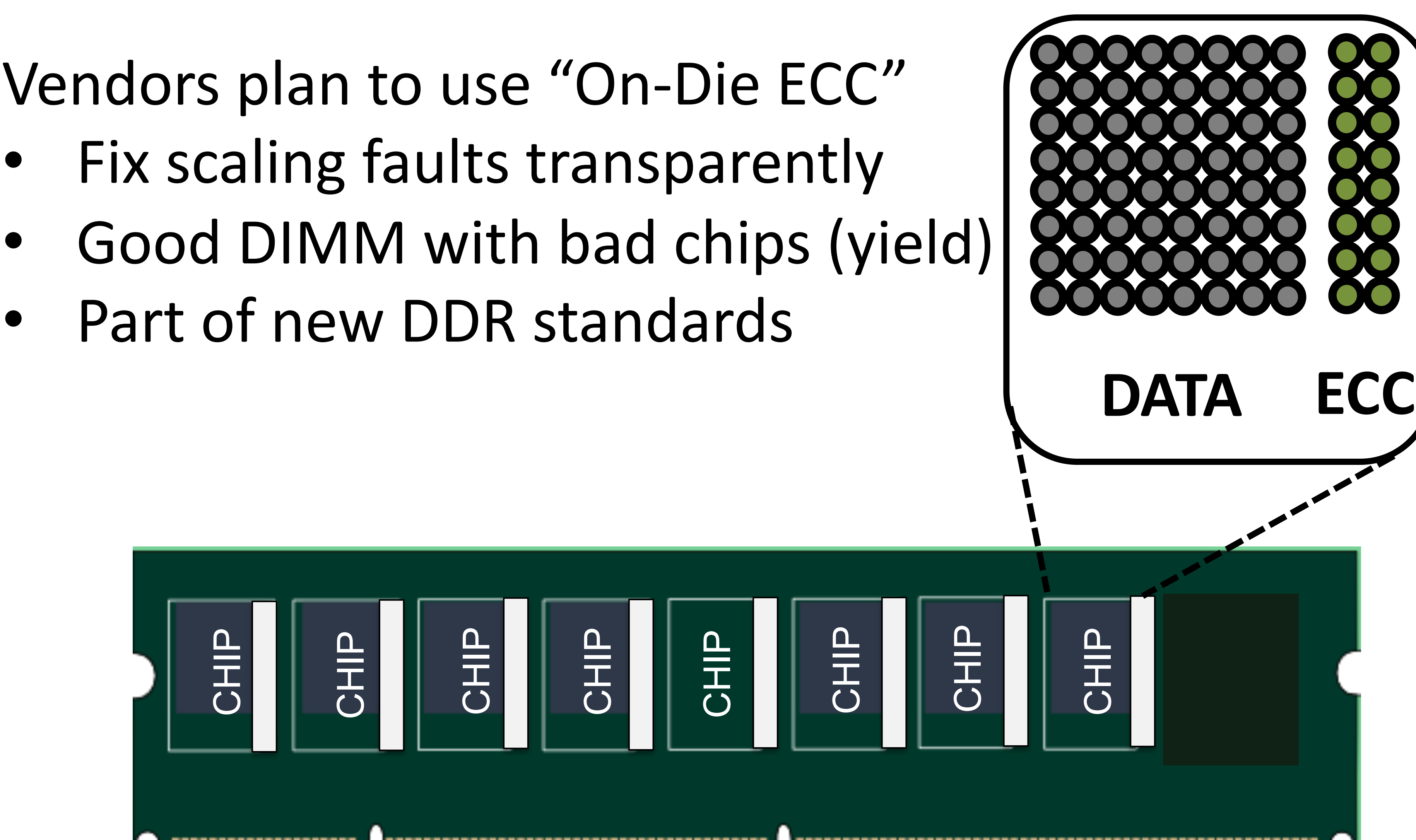
Moinuddin Qureshi



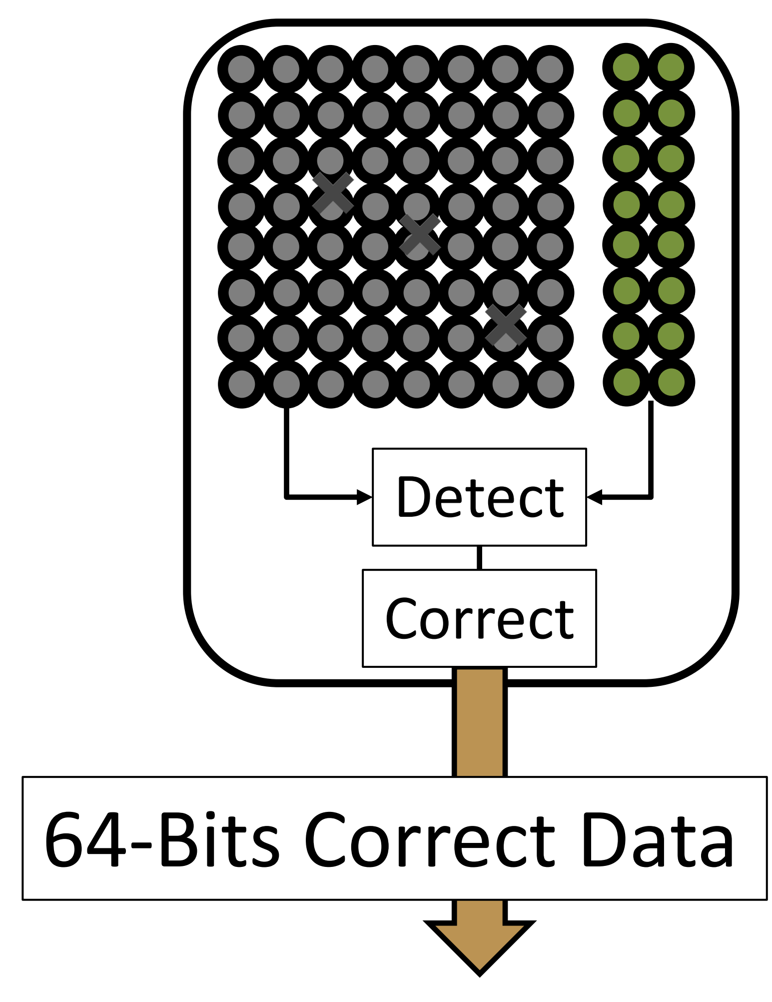
ON-DIE ECC: MITIGATE SCALING FAULTS

Vendors plan to use “On-Die ECC”

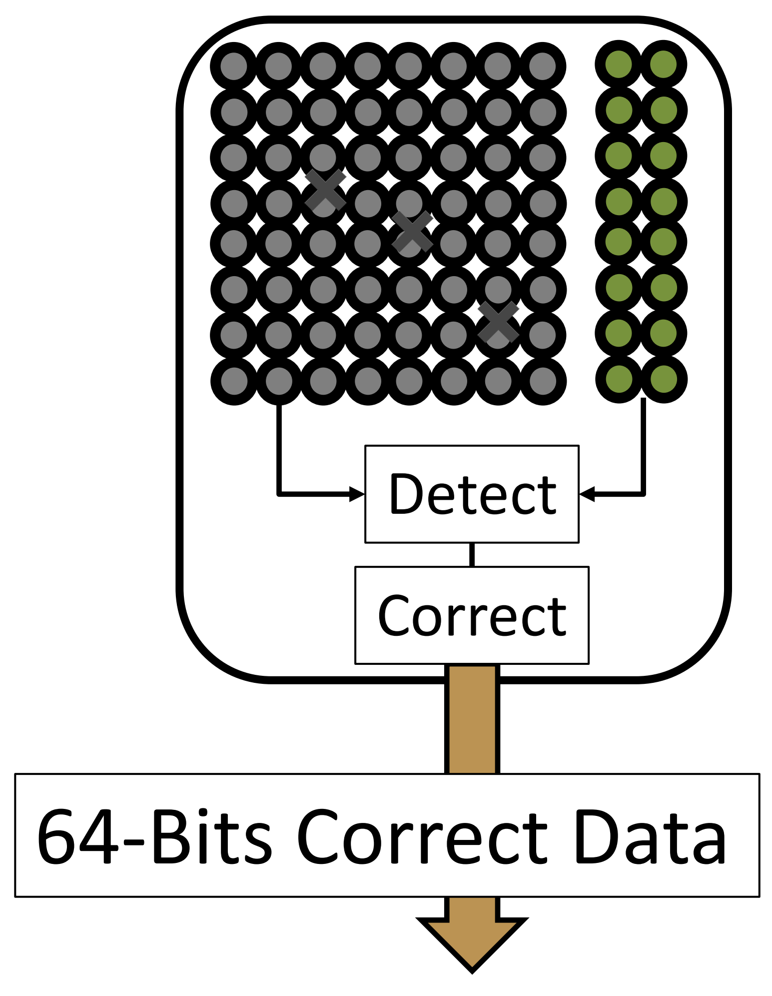
- Fix scaling faults transparently
- Good DIMM with bad chips (yield)
- Part of new DDR standards



ON-DIE ECC: MITIGATE SCALING FAULTS



ON-DIE ECC: MITIGATE SCALING FAULTS



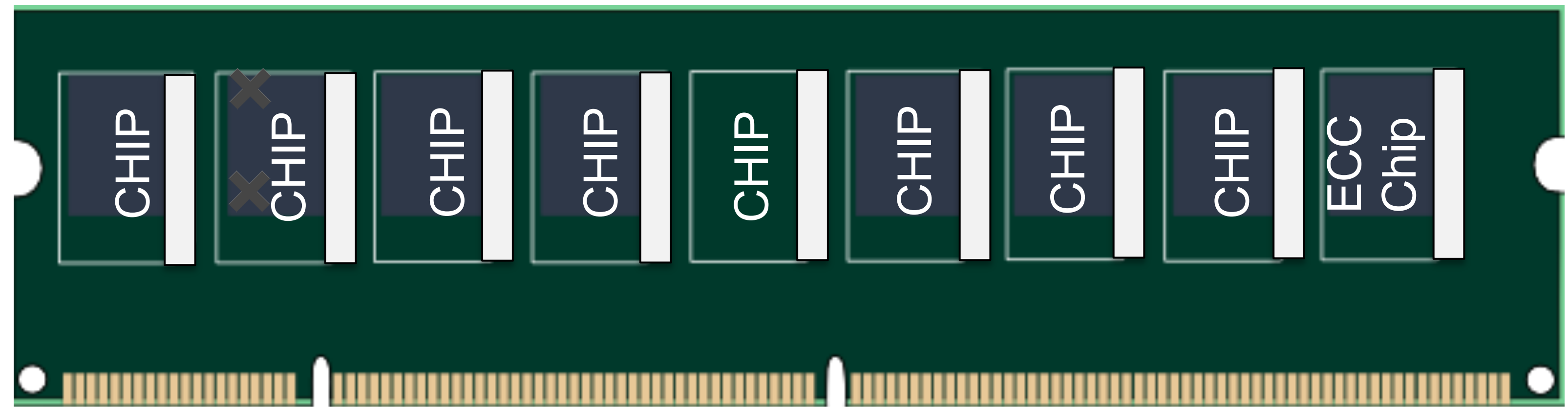
On-Die ECC fixes scaling faults invisibly

MITIGATING RUNTIME FAULTS

Runtime faults

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

ECC-DIMM (9-Chips)

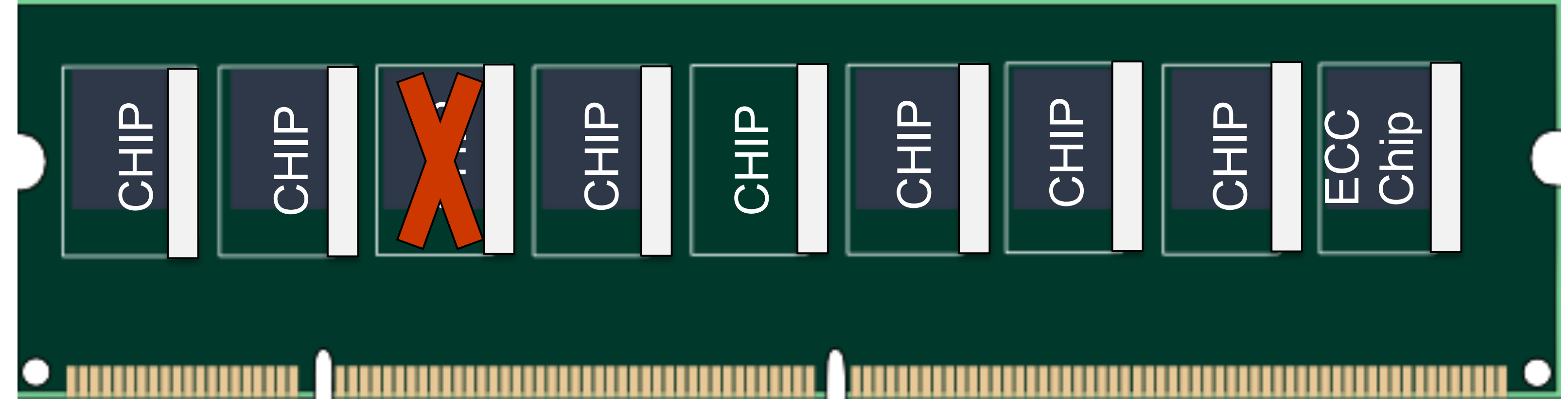


MITIGATING RUNTIME FAULTS

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

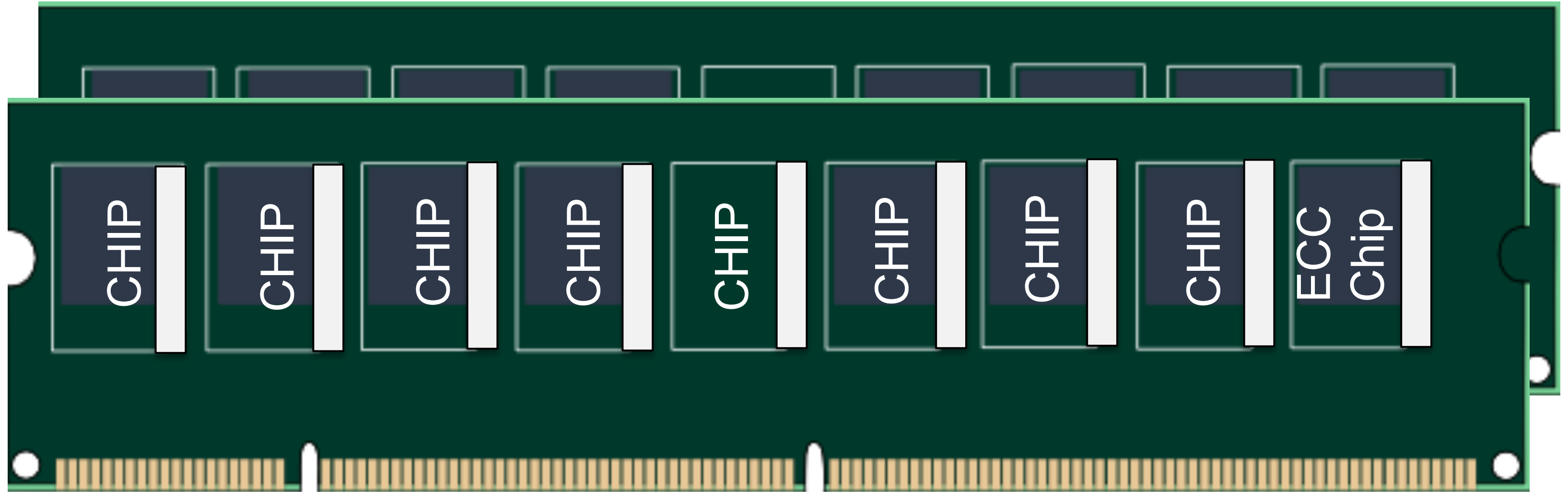


*Sridharan+ SC13

MITIGATING RUNTIME FAULTS

Runtime chip faults → Chipkill (strong ECC)

18 DRAM Chips

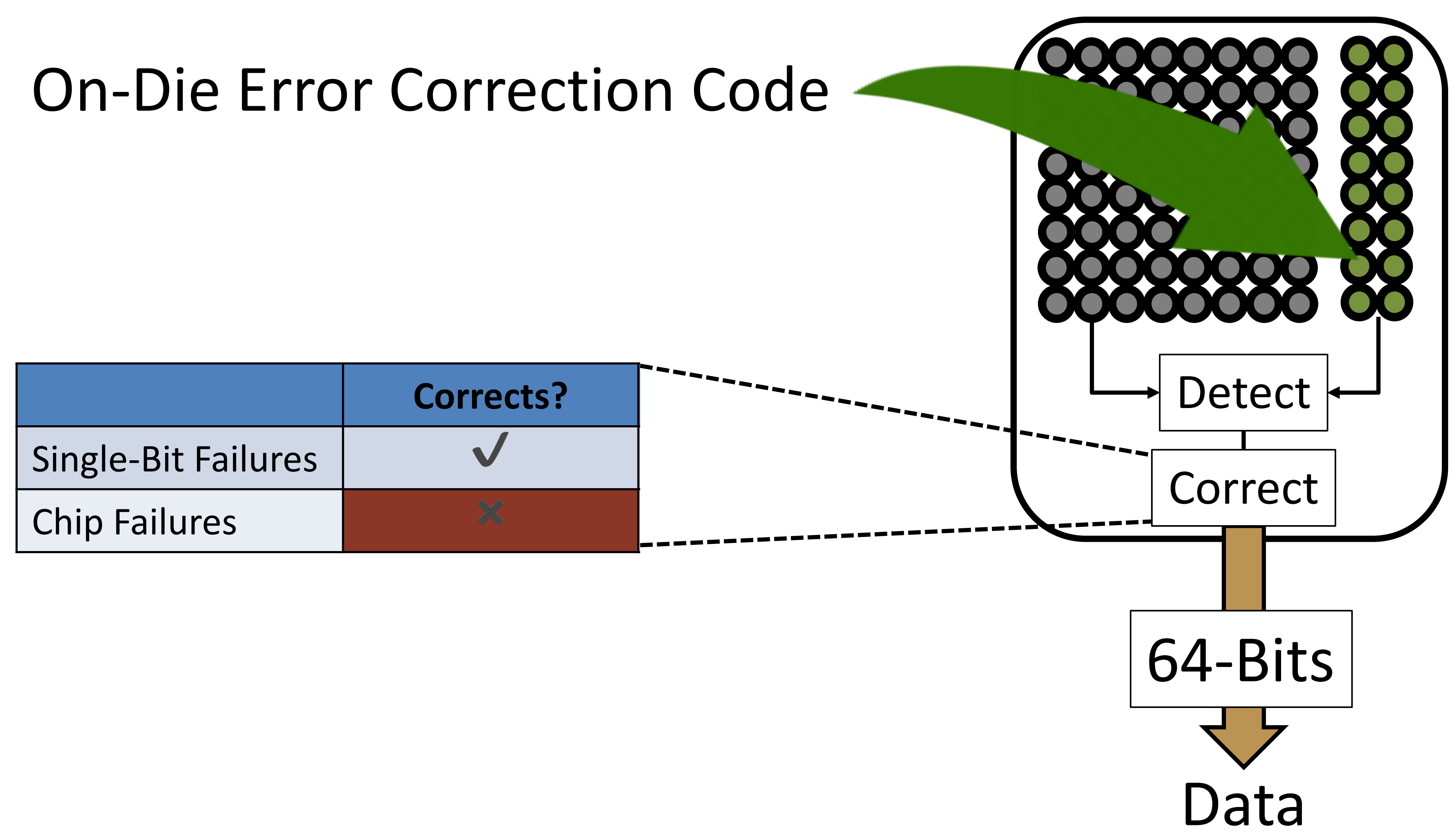


Cost: 18 Chips, Performance and Power Inefficient

GOAL

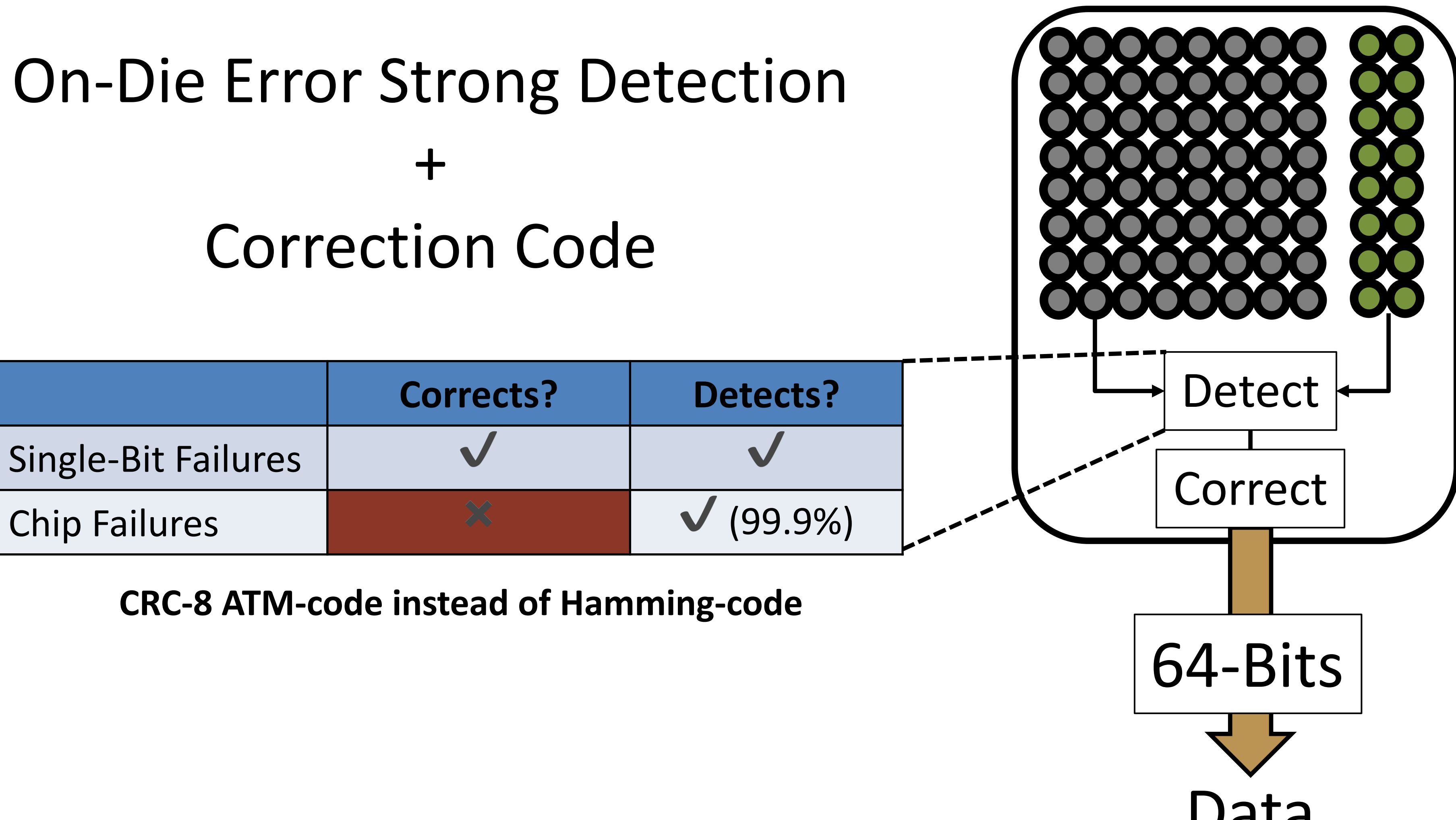
- Get "Chipkill-level" reliability with only 9 Chips
- Use On-Die ECC to enable Low-Cost Chipkill

XED: RE-PROVISION ON-DIE



	Corrects?
Single-Bit Failures	✓
Chip Failures	✗

XED: RE-PROVISION ON-DIE



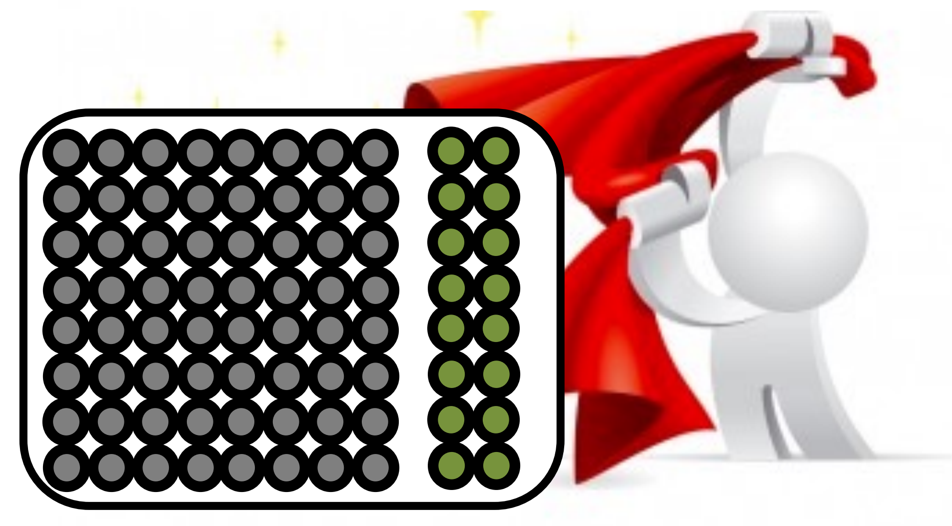
	Corrects?	Detects?
Single-Bit Failures	✓	✓
Chip Failures	✗	✓ (99.9%)

CRC-8 ATM-code instead of Hamming-code

On-Die ECC can detect chip-failures

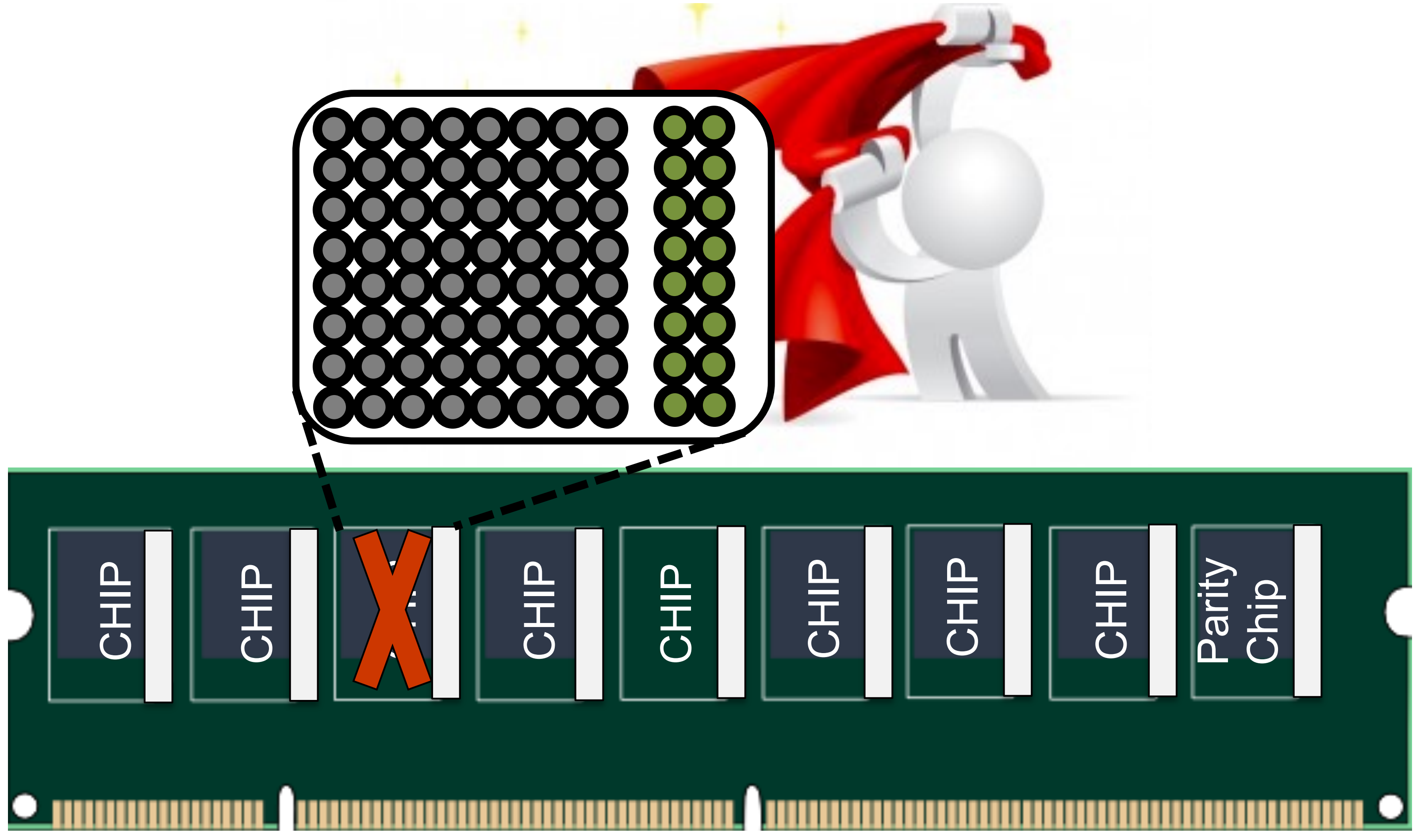
XED: RE-PROVISION ON-DIE ECC → RAID-3

Expose On-Die Error Detection → Chipkill with 9 Chips



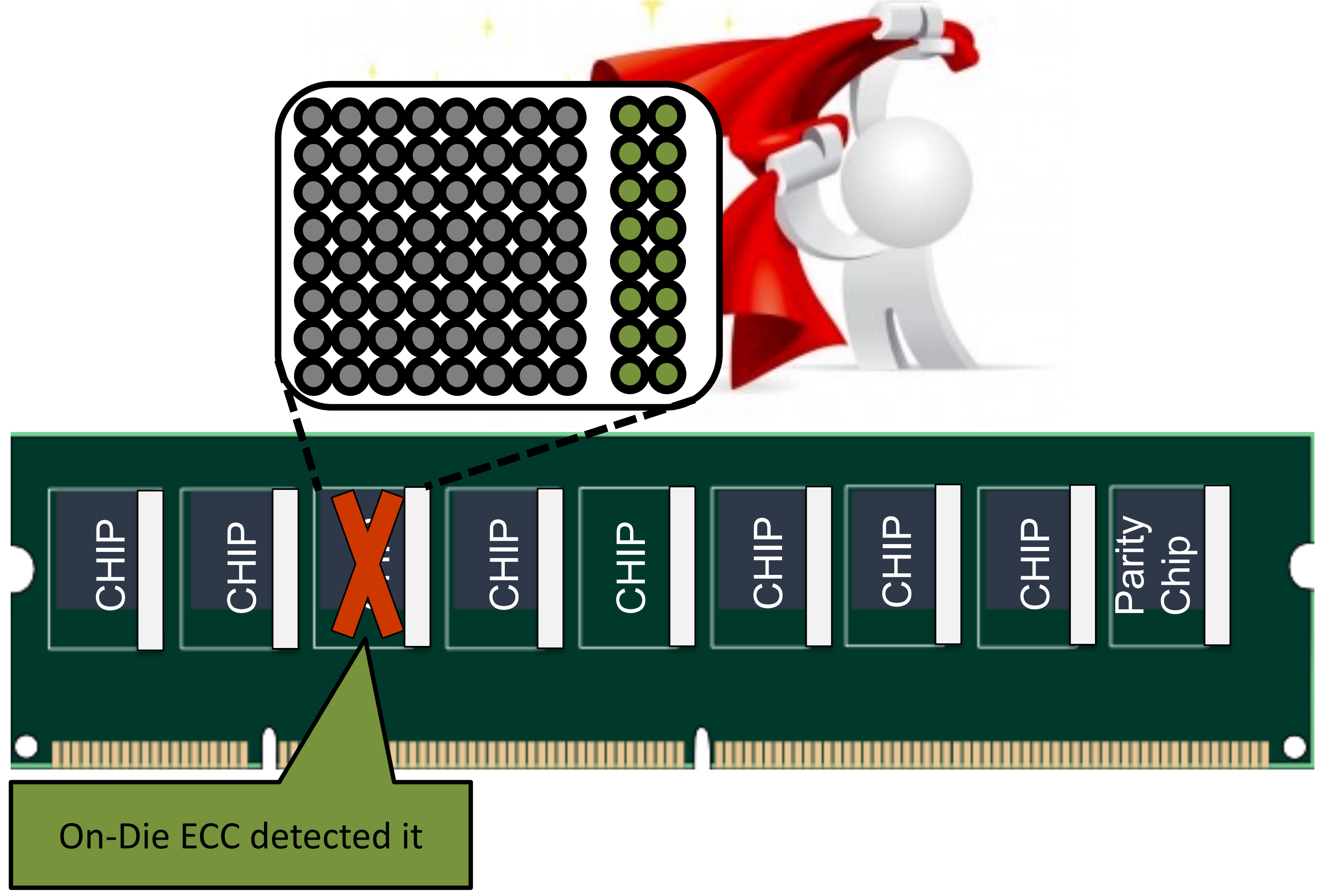
XED: RE-PROVISION ON-DIE ECC → RAID-3

Expose On-Die Error Detection → Chipkill with 9 Chips



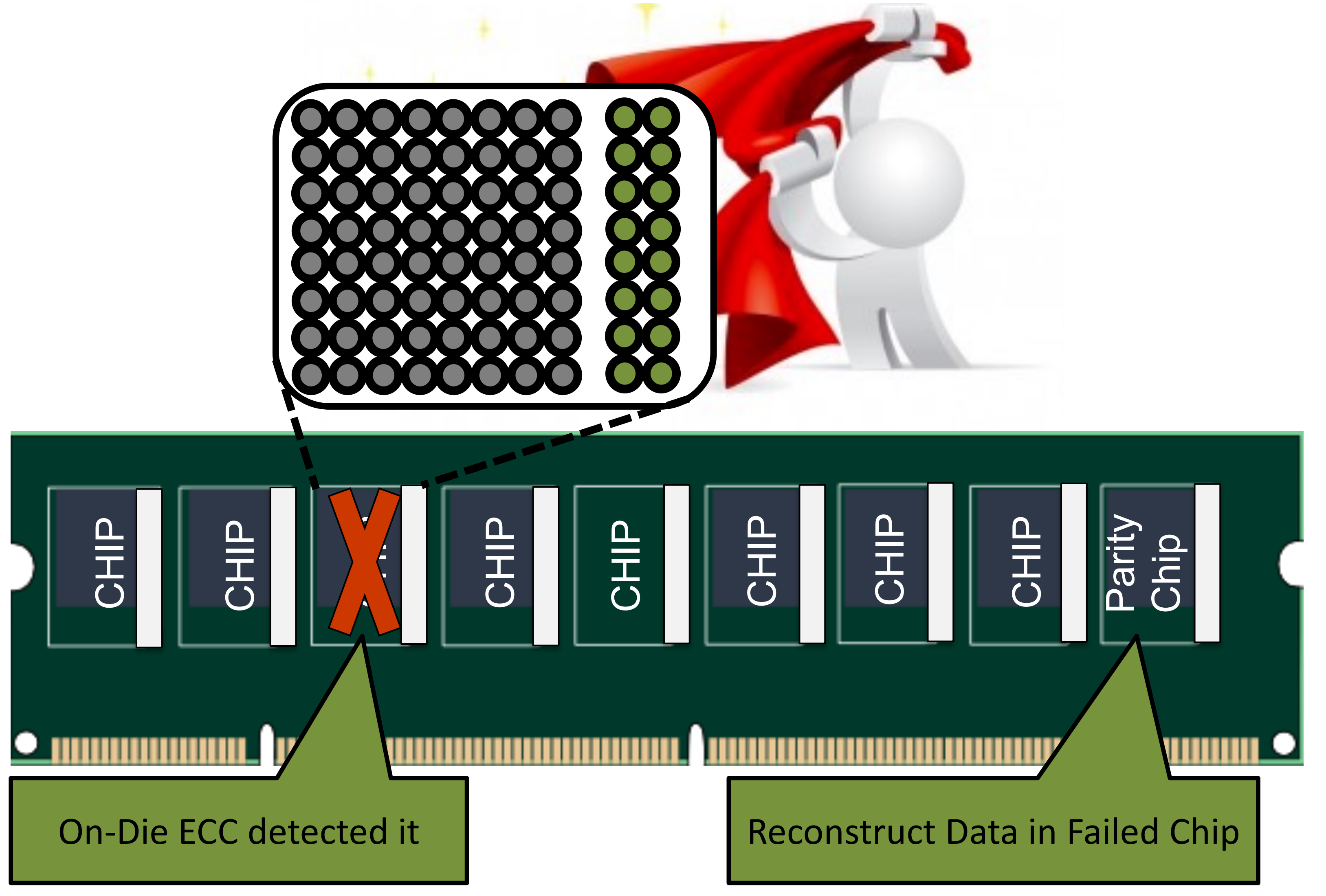
XED: RE-PROVISION ON-DIE ECC → RAID-3

Expose On-Die Error Detection → Chipkill with 9 Chips

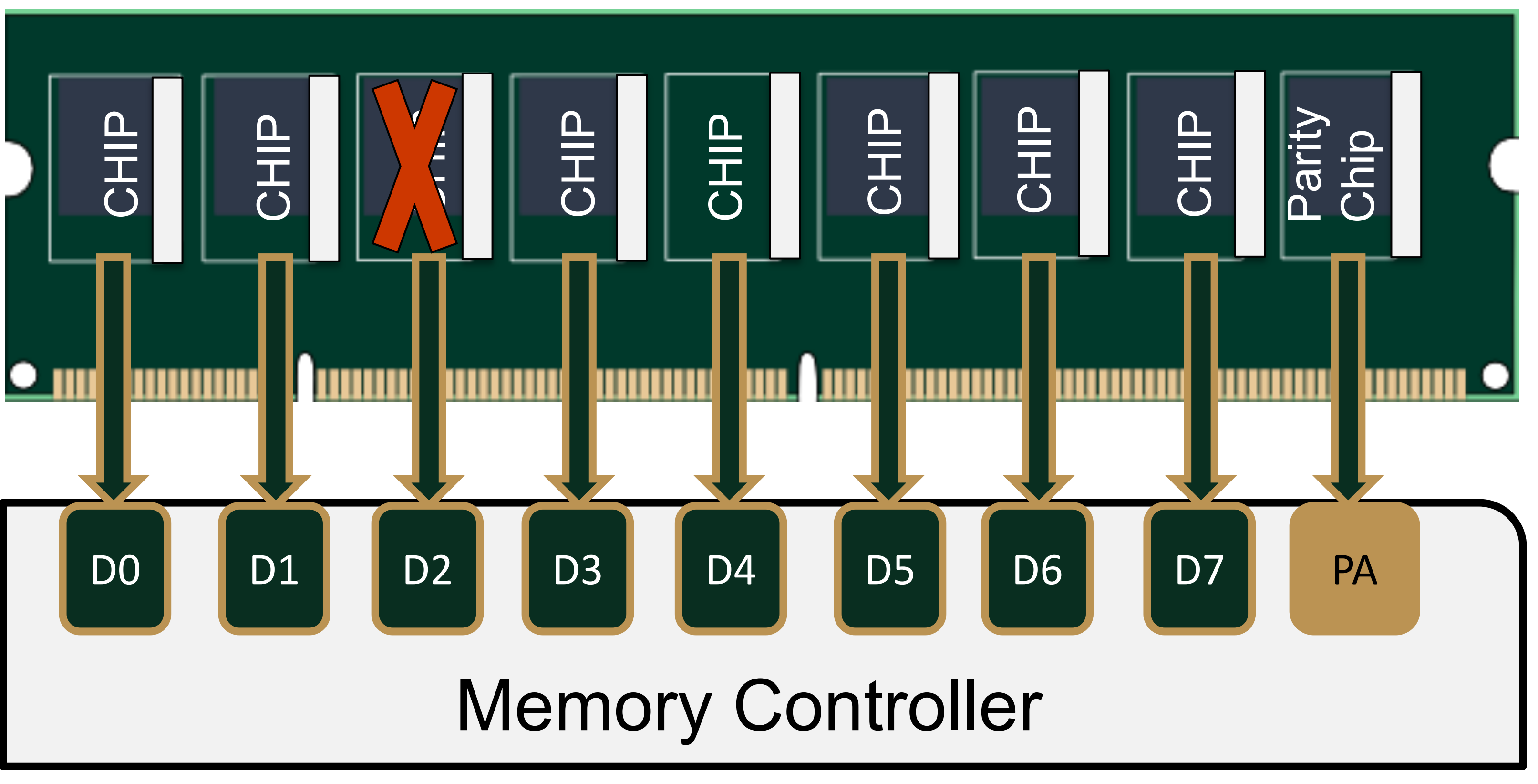


XED: RE-PROVISION ON-DIE ECC → RAID-3

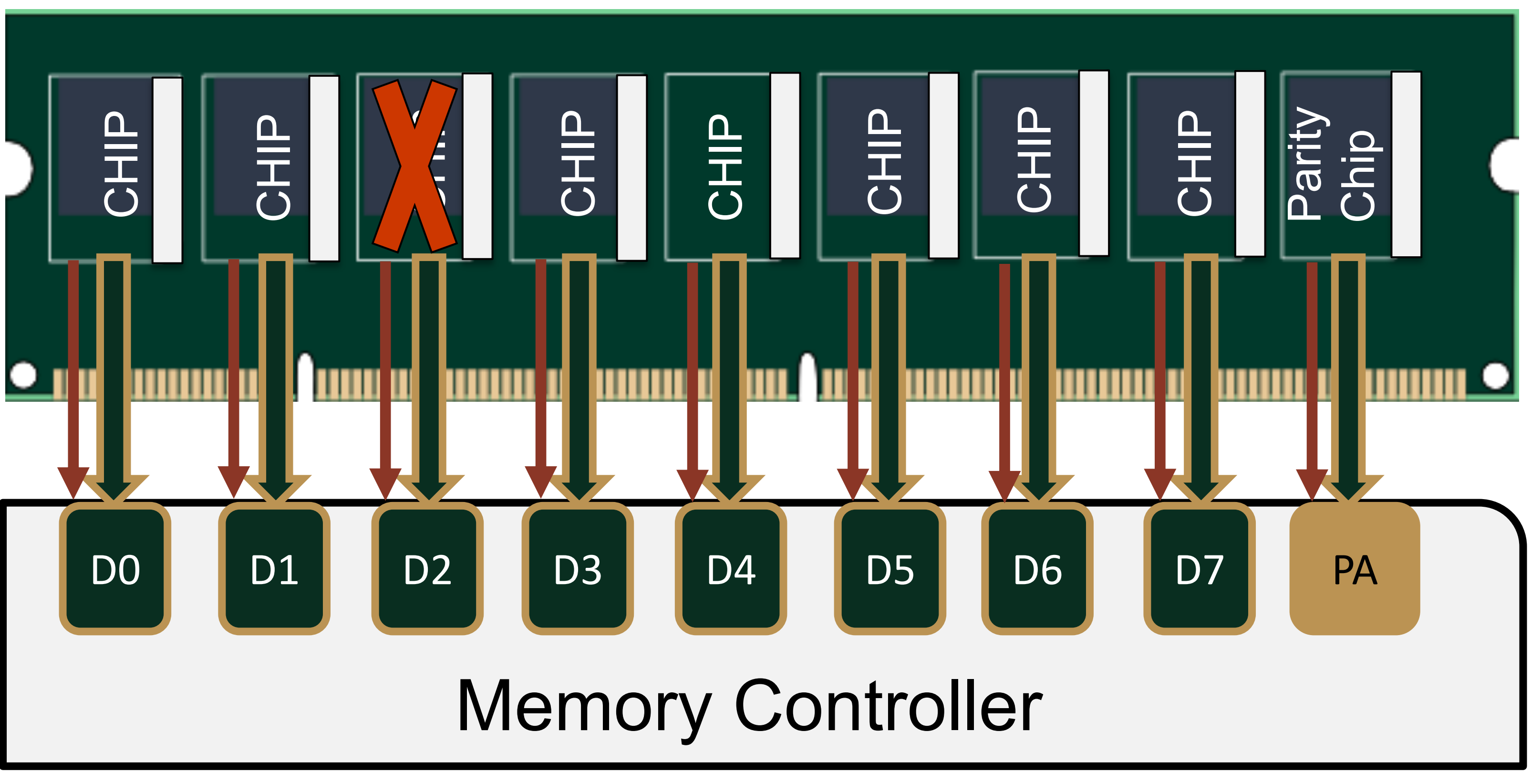
Expose On-Die Error Detection → Chipkill with 9 Chips



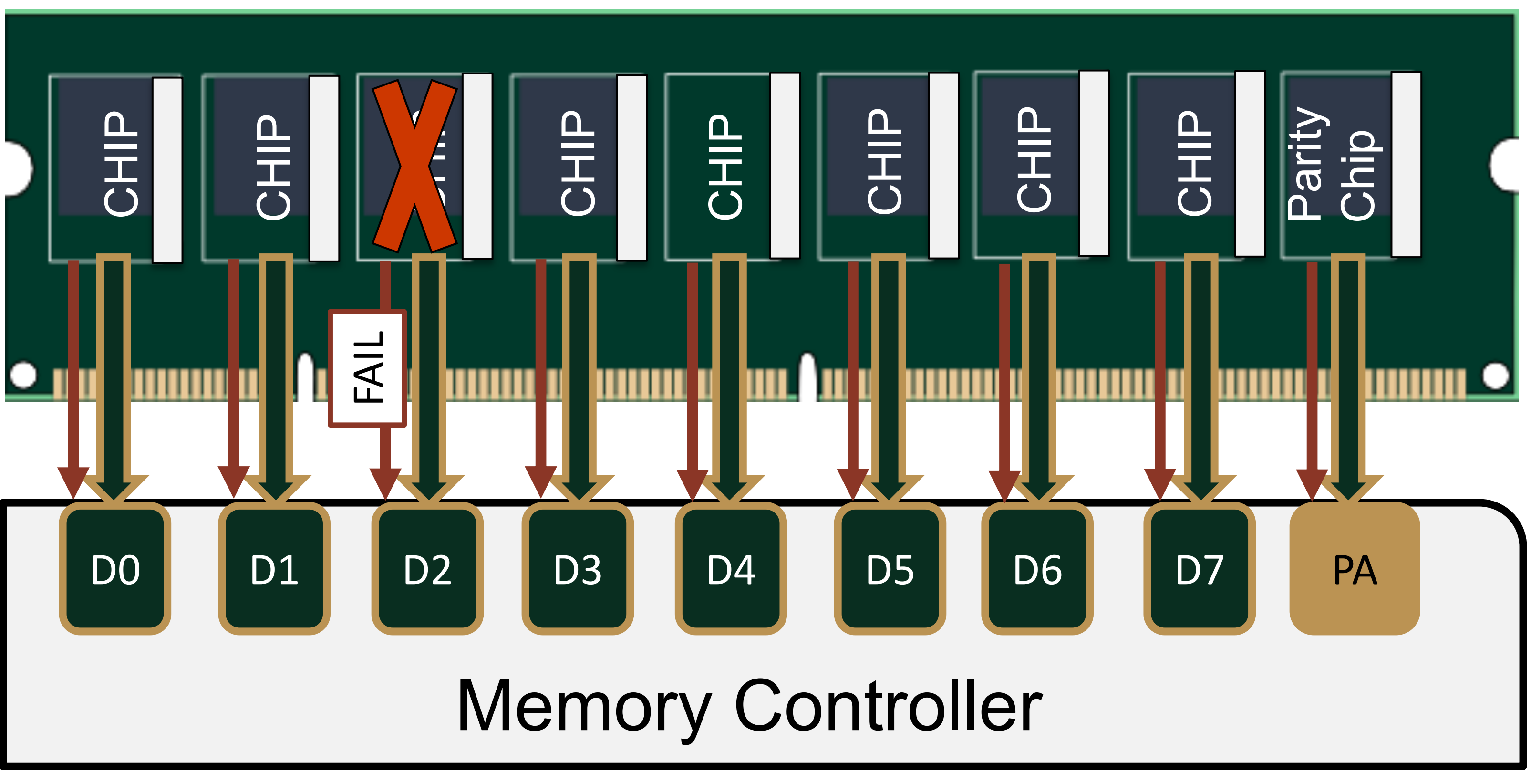
EXPOSE ON-DIE ERROR INFORMATION



EXPOSE ON-DIE ERROR INFORMATION



EXPOSE ON-DIE ERROR INFORMATION



EXPOSE ON-DIE ERROR INFORMATION

OPTION 1: Use additional wires

Incompatible with DDR memory standards

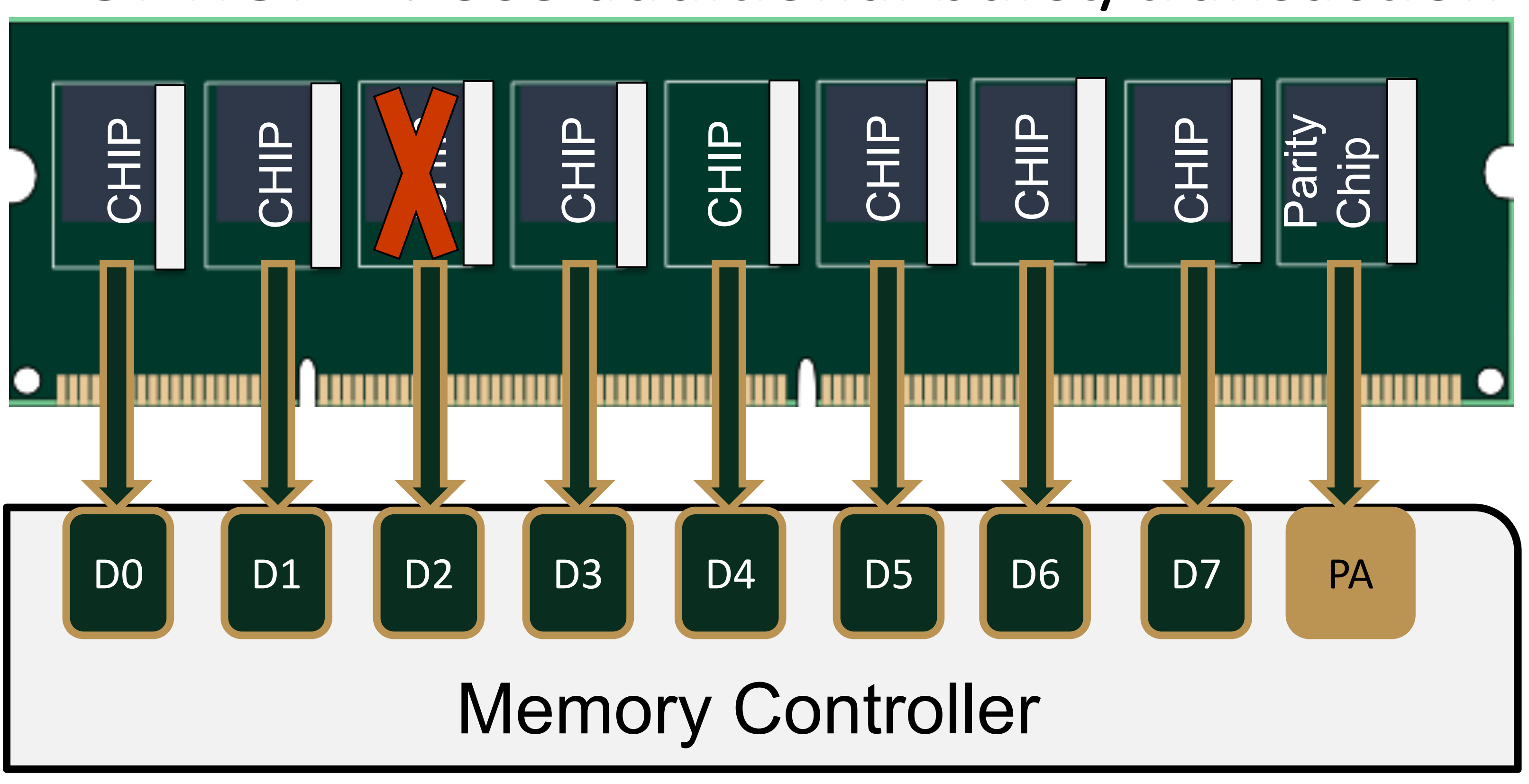
Needs a new protocol

Worse for pin-constrained future systems!

Memory Controller

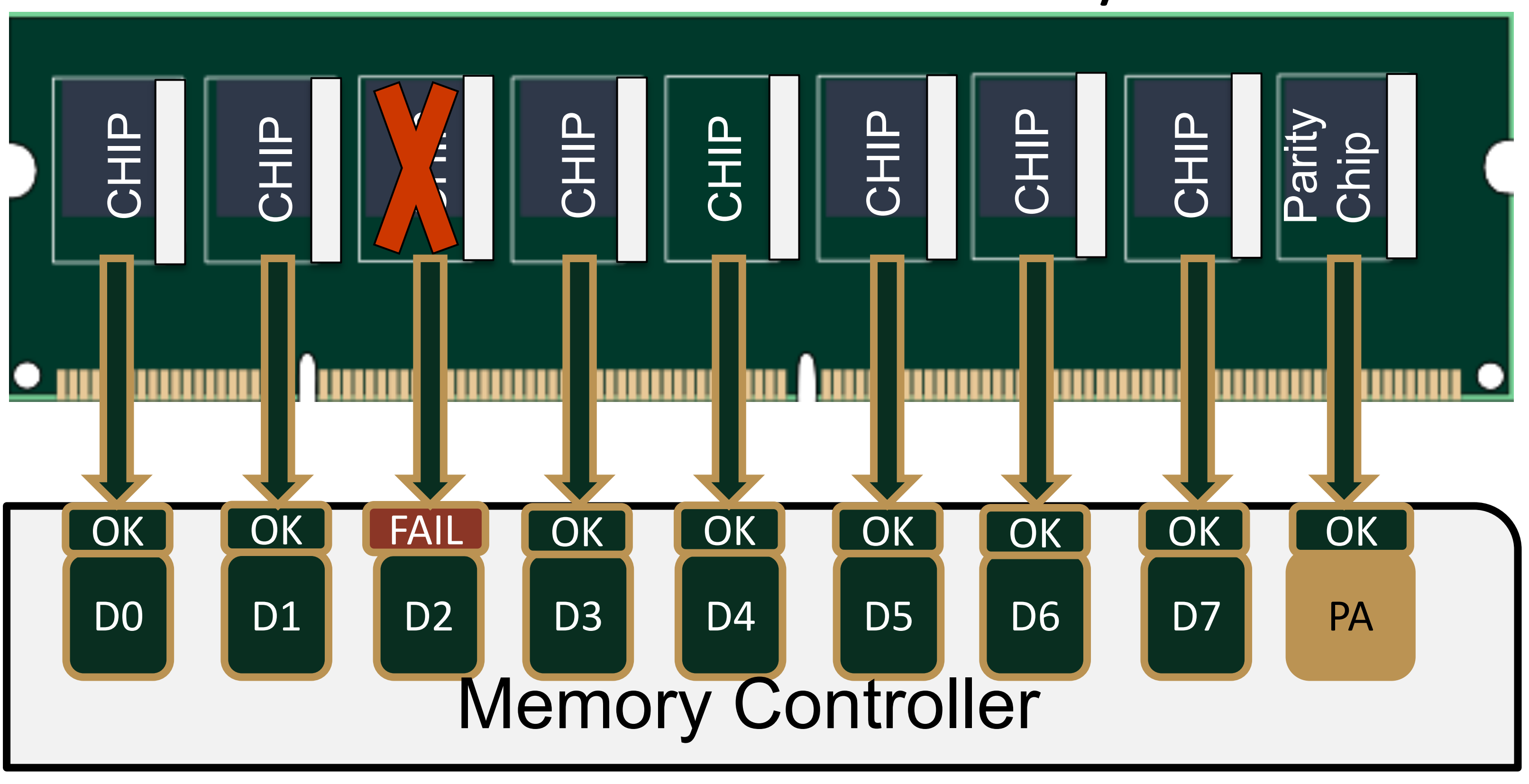
EXPOSE ON-DIE ERROR INFORMATION

OPTION 2: Use additional burst/transaction



EXPOSE ON-DIE ERROR INFORMATION

OPTION 2: Use additional burst/transaction



EXPOSE ON-DIE ERROR INFORMATION

OPTION 2: Use additional burst/transaction

CHIP CHIP CHIP CHIP CHIP CHIP CHIP CHIP Parity Chip

Additional 12.5% to 100% bandwidth overheads

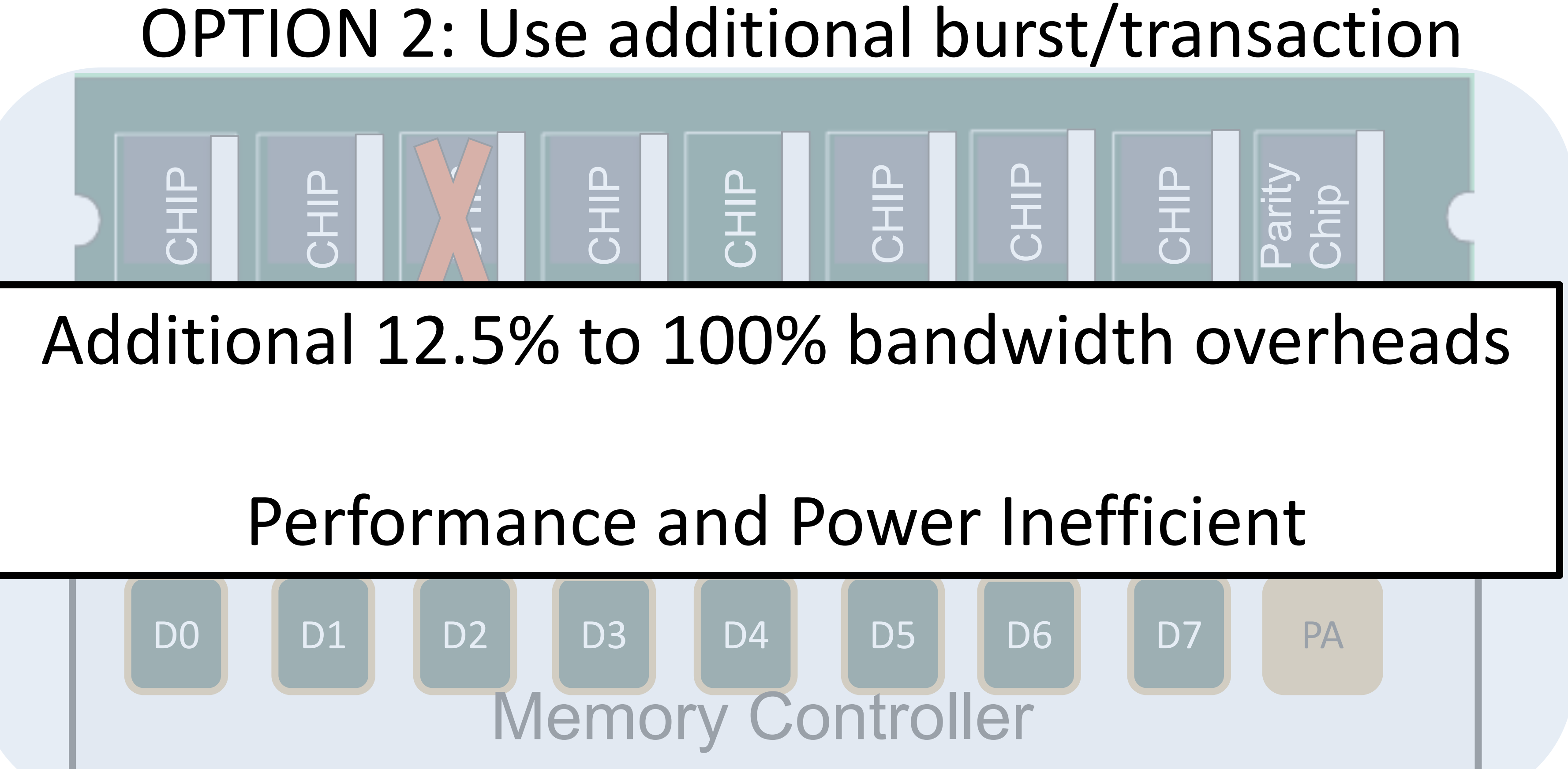
Performance and Power Inefficient

D0 D1 D2 D3 D4 D5 D6 D7 PA

Memory Controller

EXPOSE ON-DIE ERROR INFORMATION

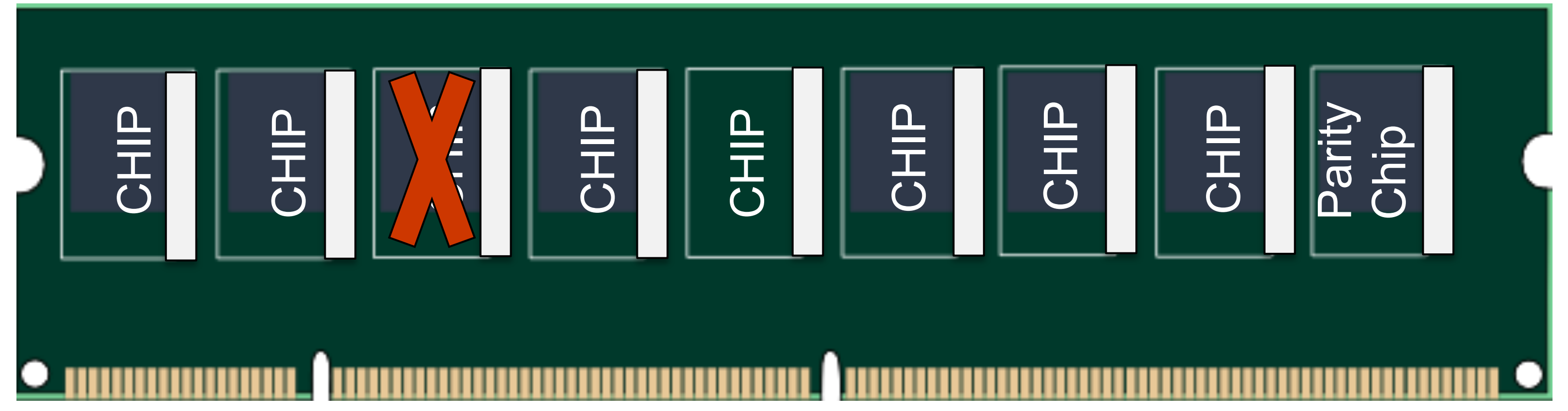
OPTION 2: Use additional burst/transaction



Expose On-Die error detection with minor changes

XED: ON-DIE ERROR INFORMATION FOR FREE

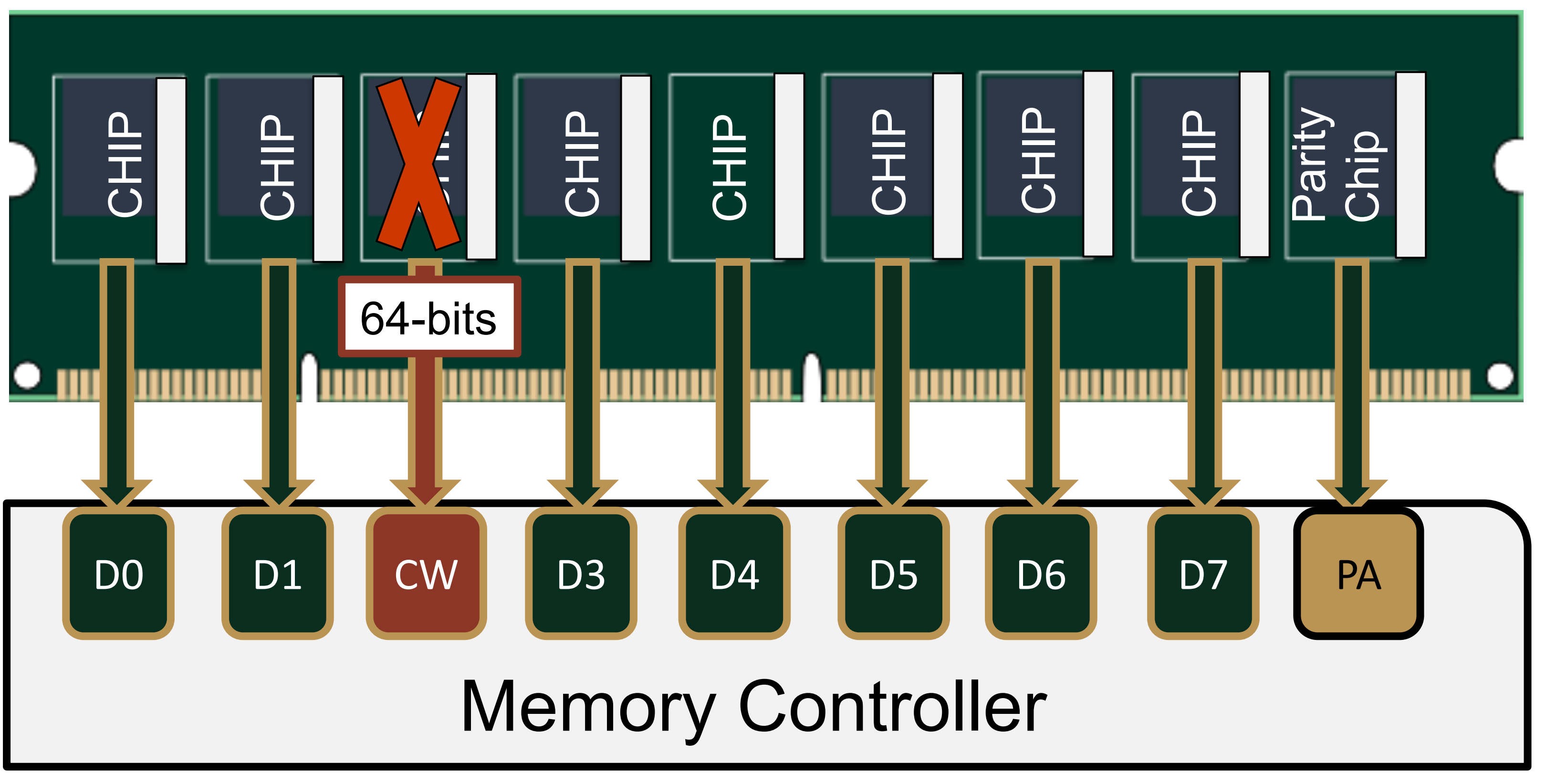
On detecting an error, the DRAM chip sends a 64-bit "Catch-Word" (CW) instead of data



Memory Controller

XED: ON-DIE ERROR INFORMATION FOR FREE

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



XED: ON-DIE ERROR INFORMATION 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

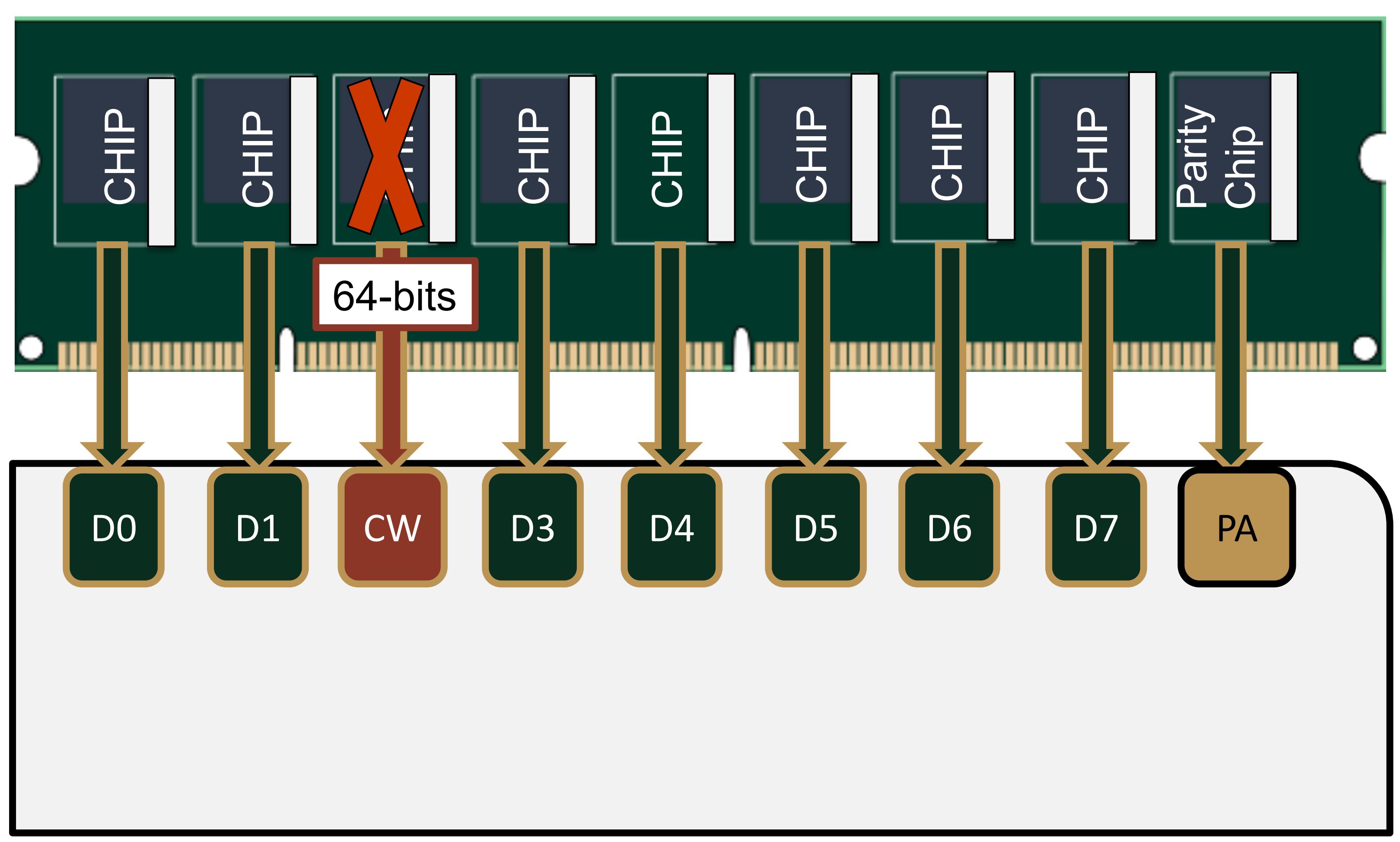
Compatible with existing memory protocols

Memory Controller

64-bit Catch-Words identify the faulty chip

WHY CATCH-WORDS WORK: SCENARIO-1

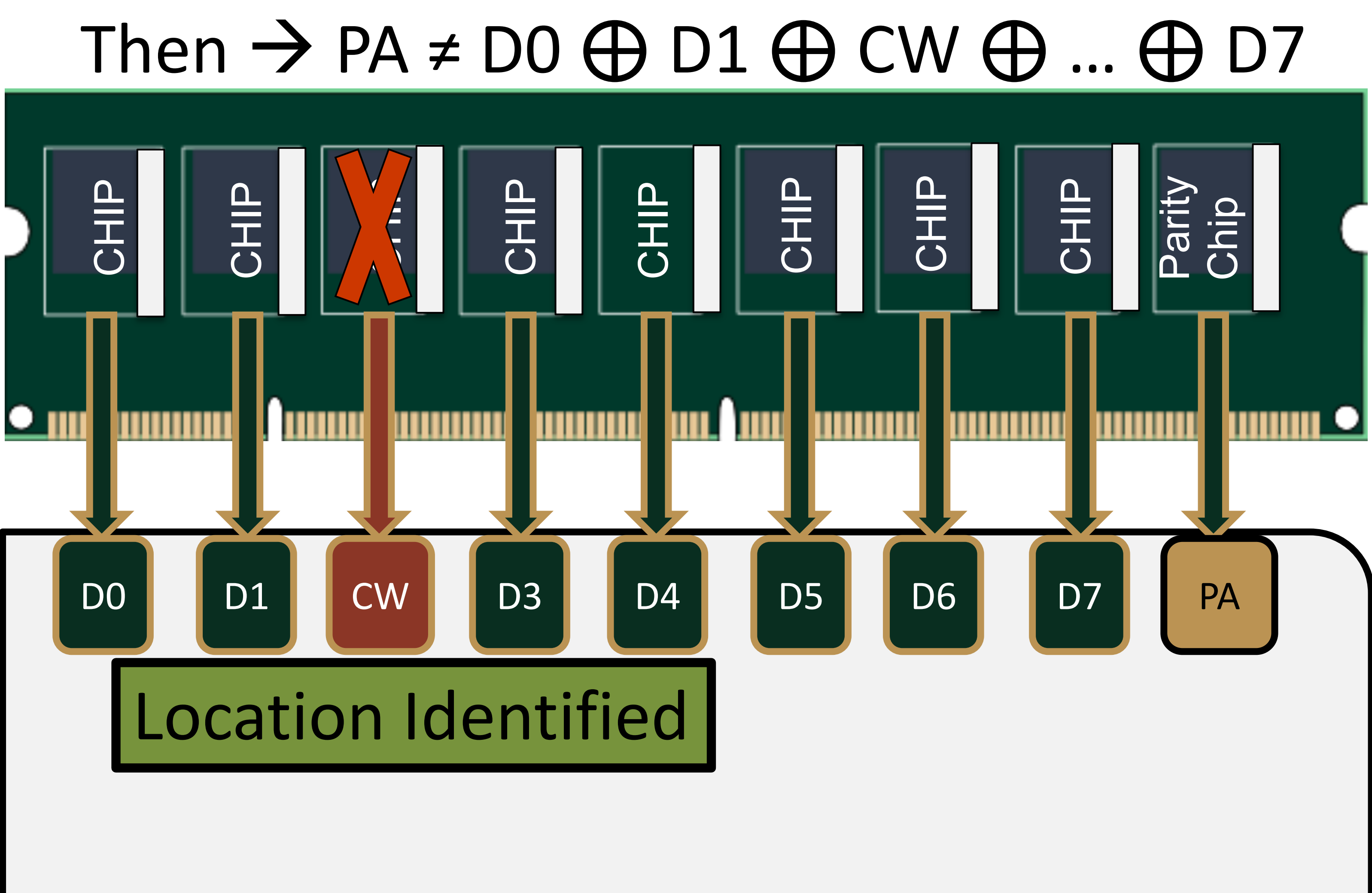
Catch Word (CW) ≠ Valid Data (D2)



WHY CATCH-WORDS WORK: SCENARIO-1

Catch Word (CW) ≠ Valid Data (D2)

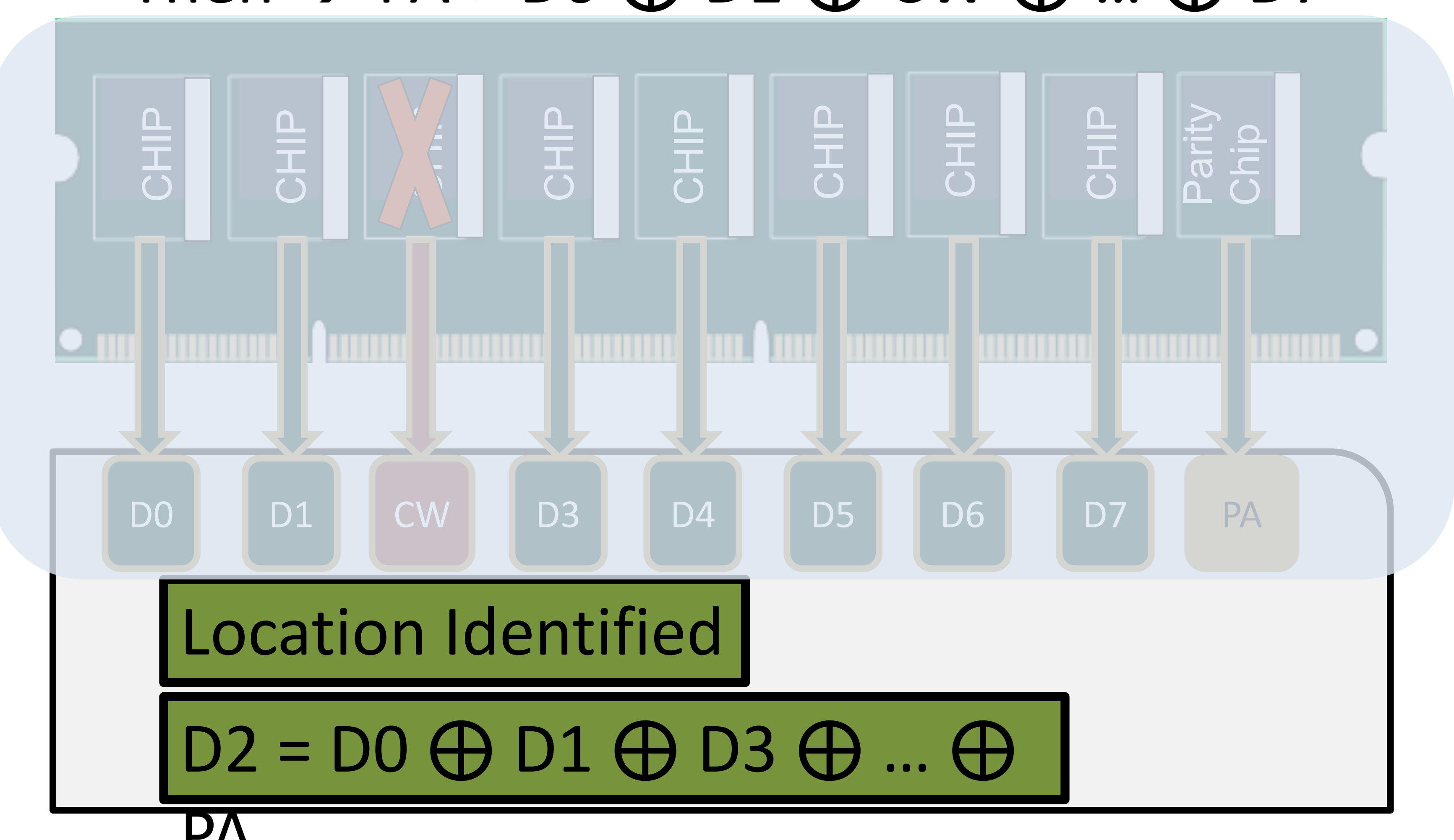
Then → PA ≠ D0 ⊕ D1 ⊕ CW ⊕ … ⊕ D7



WHY CATCH-WORDS WORK: SCENARIO-1

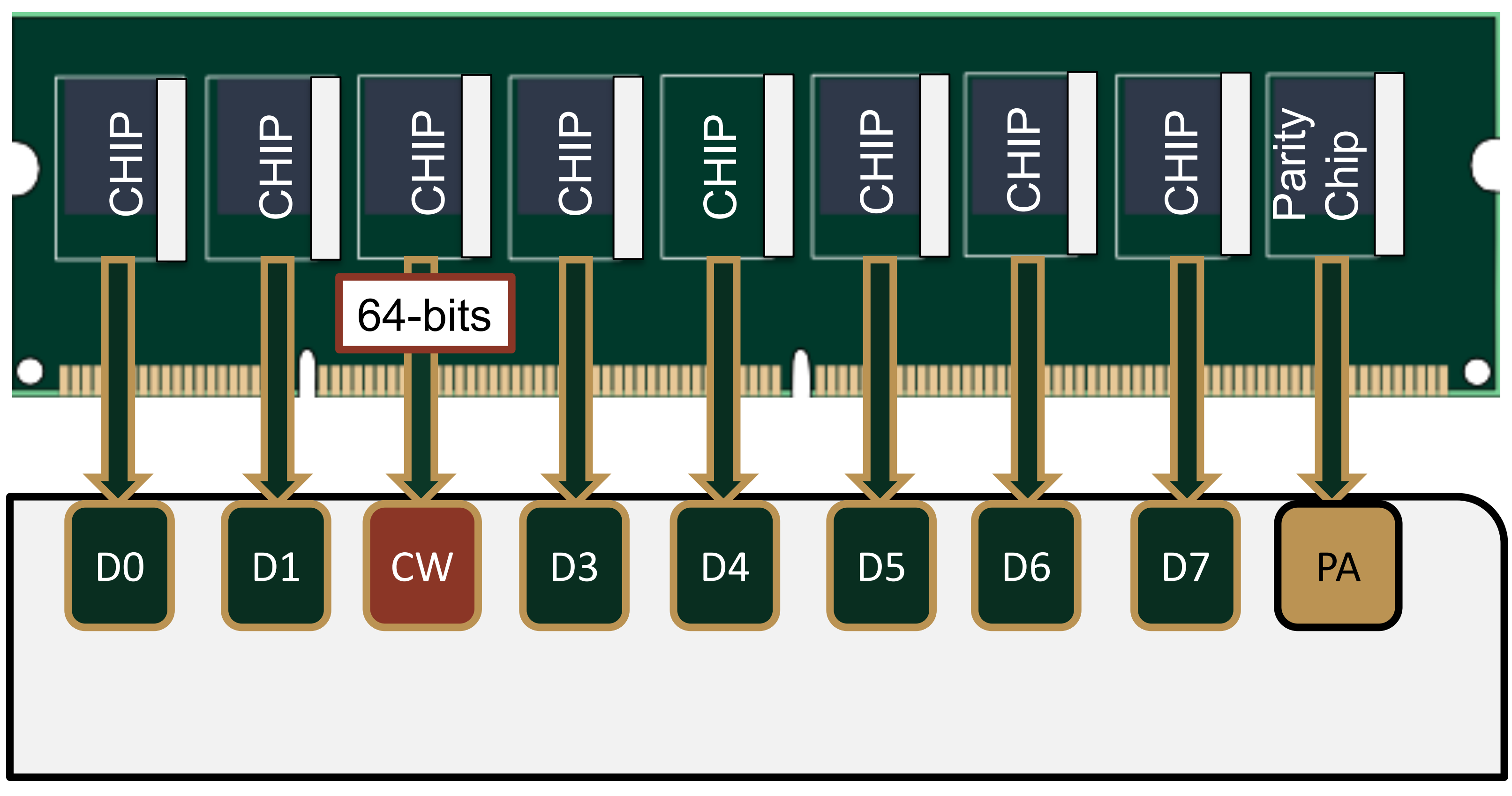
Catch Word (CW) ≠ Valid Data (D2)

Then → PA ≠ D0 ⊕ D1 ⊕ CW ⊕ … ⊕ D7



WHY CATCH-WORDS WORK: SCENARIO-2

Catch Word (CW) = Valid Data (D2)



WHY CATCH-WORDS WORK: SCENARIO-2

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

Then → PA = D0 ⊕ D1 ⊕ CW ⊕ … ⊕ D7

CHIP CHIP CHIP CHIP CHIP CHIP CHIP CHIP Parity Chip

D0 D1 CW D3 D4 D5 D6 D7 PA

No Error as Parity Matches

WHY CATCH-WORDS WORK: SCENARIO-2

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

Then → PA = D0 ⊕ D1 ⊕ CW ⊕ … ⊕ D7

CHIP CHIP CHIP CHIP CHIP CHIP CHIP CHIP Parity Chip

D0 D1 CW D3 D4 D5 D6 D7 PA

No Error as Parity Matches

Catch-Word collision: Doesn't affect correctness

CATCH-WORD COLLISIONS: NOT A PROBLEM

- A chip stores 64 bits/cache-line → 2^{64} combinations

CATCH-WORD COLLISIONS: NOT A PROBLEM

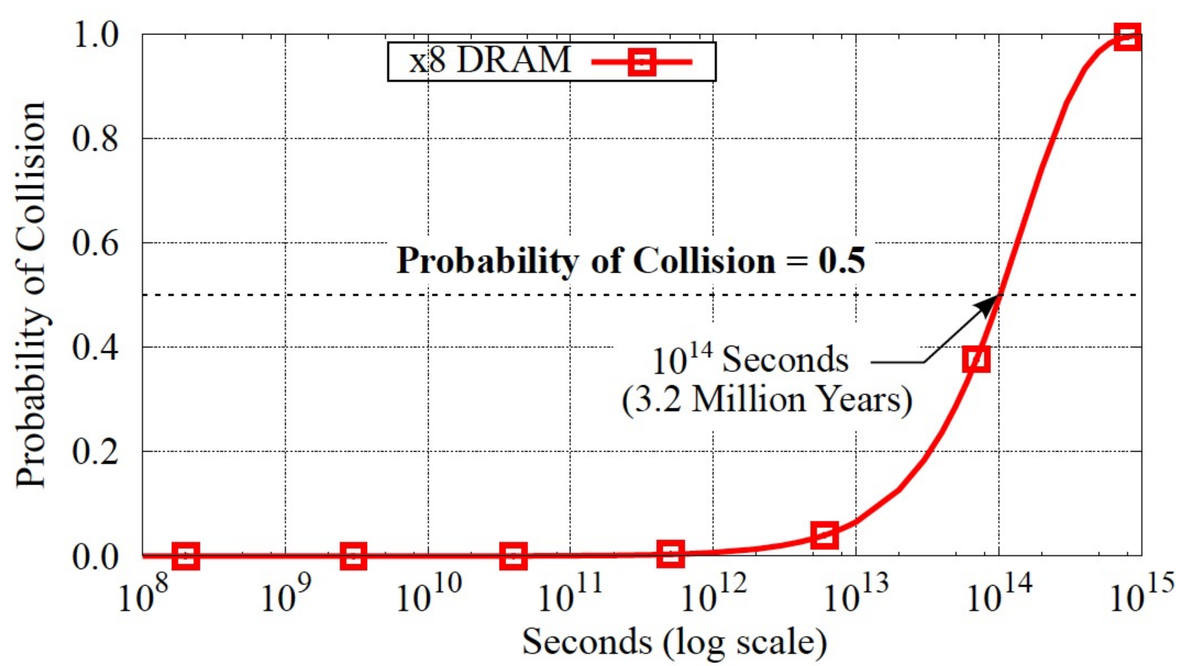
- A chip stores 64 bits/cache-line → 2^{64} combinations
- Even a 16Gb chip has only 2^{28} cachelines

CATCH-WORD COLLISIONS: NOT A PROBLEM

- A chip stores 64 bits/cache-line → 2^{64} combinations
- Even a 16Gb chip has only 2^{28} cachelines
- Nearly $2^{63.99}$ data combinations free!

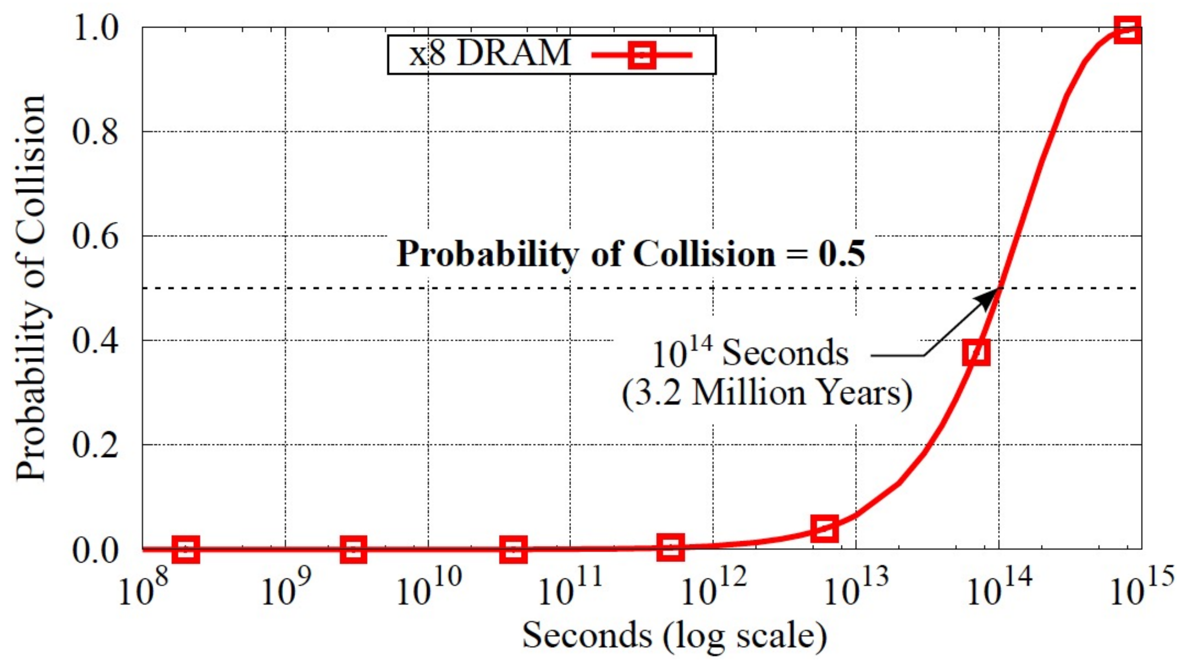
CATCH-WORD COLLISIONS: NOT A PROBLEM

- A chip stores 64 bits/cache-line → 2^{64} combinations
- Even a 16Gb chip has only 2^{28} cachelines
- Nearly $2^{63.99}$ data combinations free!



CATCH-WORD COLLISIONS: NOT A PROBLEM

- A chip stores 64 bits/cache-line → 2^{64} combinations
- Even a 16Gb chip has only 2^{28} cachelines
- Nearly $2^{63.99}$ data combinations free!



The catch-word will most likely not collide

EVALUATION

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

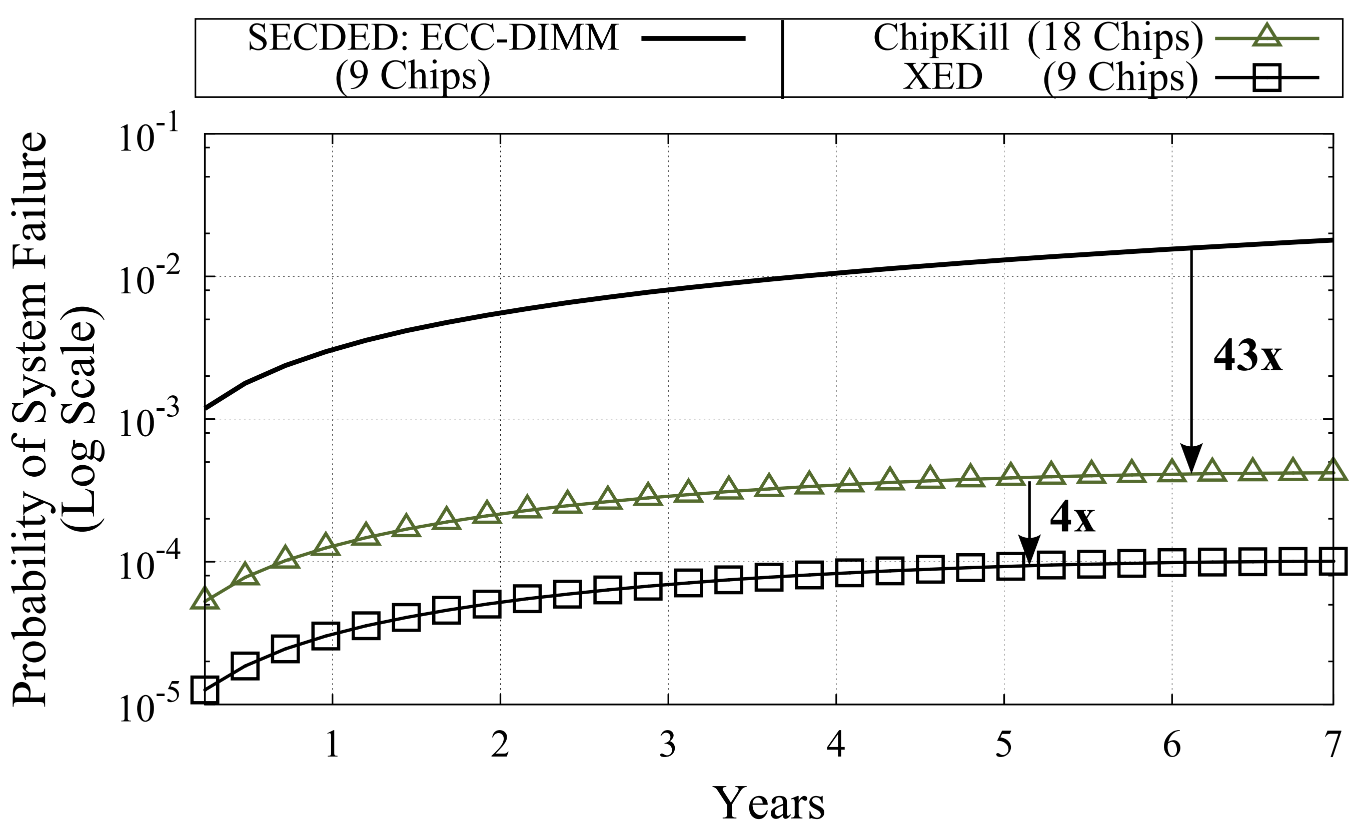
FaultSim*: Memory Reliability Simulator

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

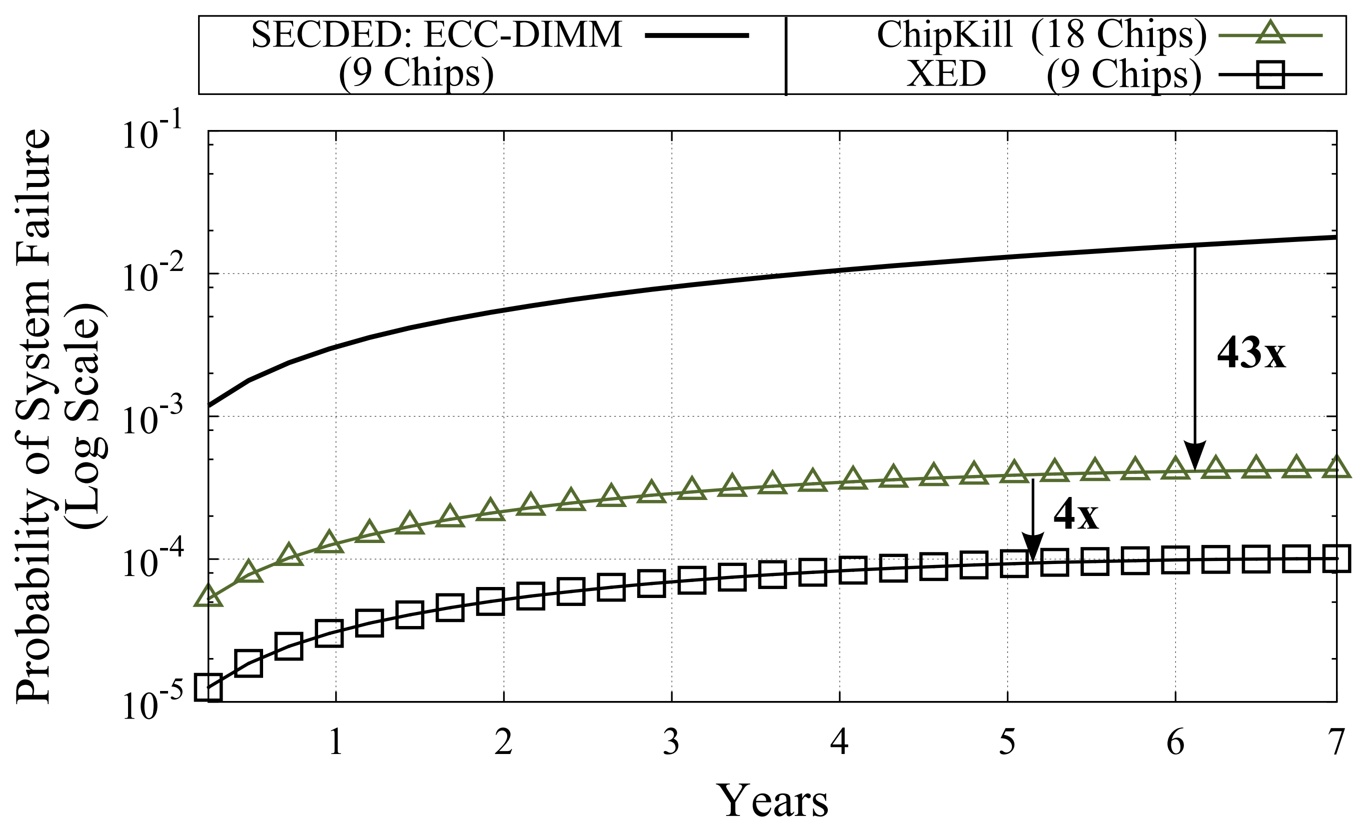
* Nair et. al. HiPEAC 2016

RESULTS: RELIABILITY

XED vs Commercial ECC schemes

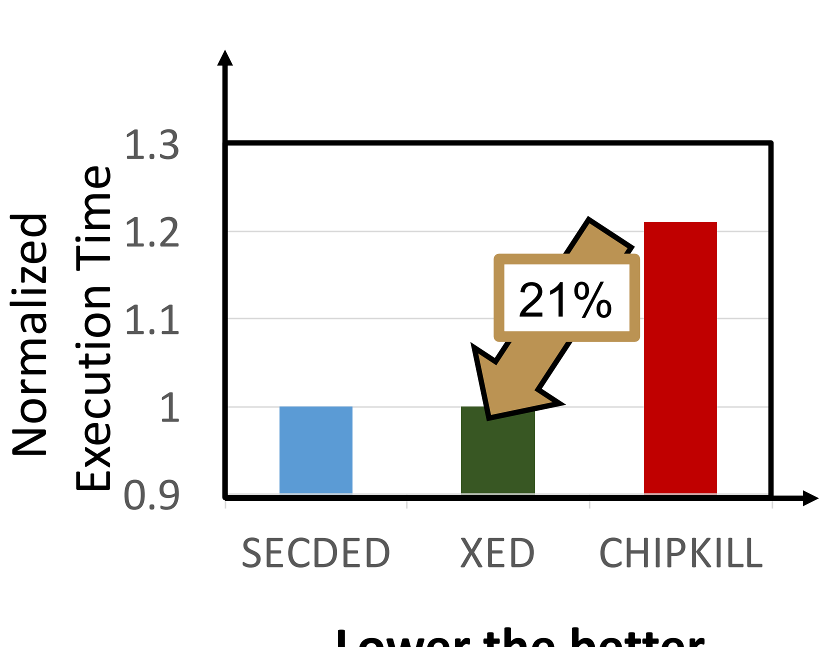


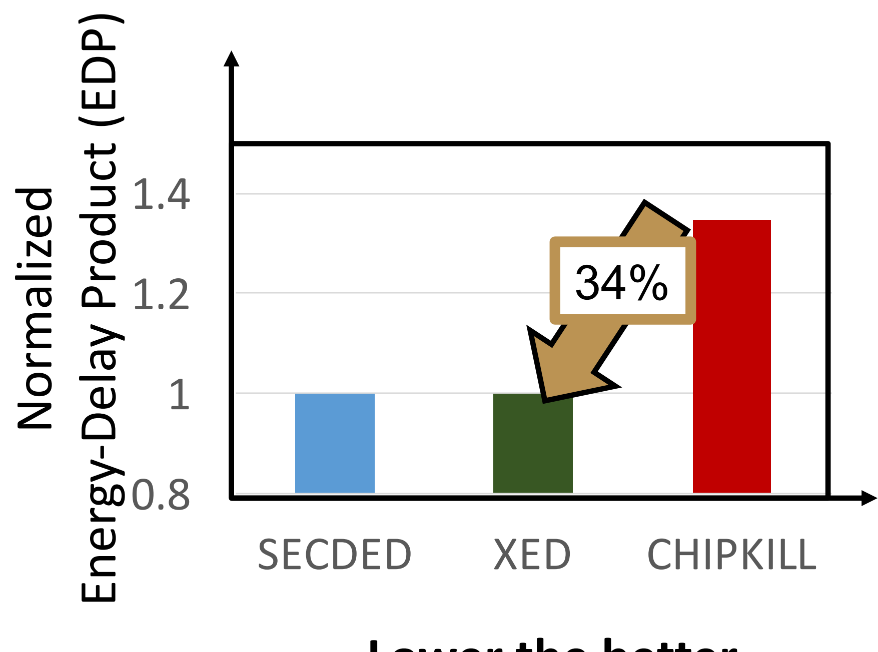
RESULTS: RELIABILITY



XED provides 172x higher reliability

RESULTS: PERFORMANCE AND EDP





XED enables Chipkill with a single DIMM
Significant performance & power benefits

SUMMARY

MEMORY SYSTEMS AND MOORE’S LAW

- Hurdles for Moore’s Law: Scaling & Runtime Faults

MEMORY SYSTEMS AND MOORE'S LAW

- Hurdles for Moore's Law: Scaling & Runtime Faults

- Current techniques are costly/ineffective

MEMORY SYSTEMS AND MOORE'S LAW

- Hurdles for Moore's Law: Scaling & Runtime Faults
- Current techniques are costly/ineffective

Low-cost architectural techniques can enable reliable and scalable memory systems → Sustain Moore's Law

MEMORY SYSTEMS AND MOORE'S LAW

- Hurdles for Moore's Law: Scaling & Runtime Faults

- Current techniques are costly/ineffective

Low-cost architectural techniques can enable reliable and scalable memory systems → Sustain Moore's Law

- **100-1000x** higher reliability with minimal overheads

AREAS INVESTIGATED

Published in ISCA, MICRO, ASPLOS, HPCA, DSN, HiPEAC and CAL

Reliability

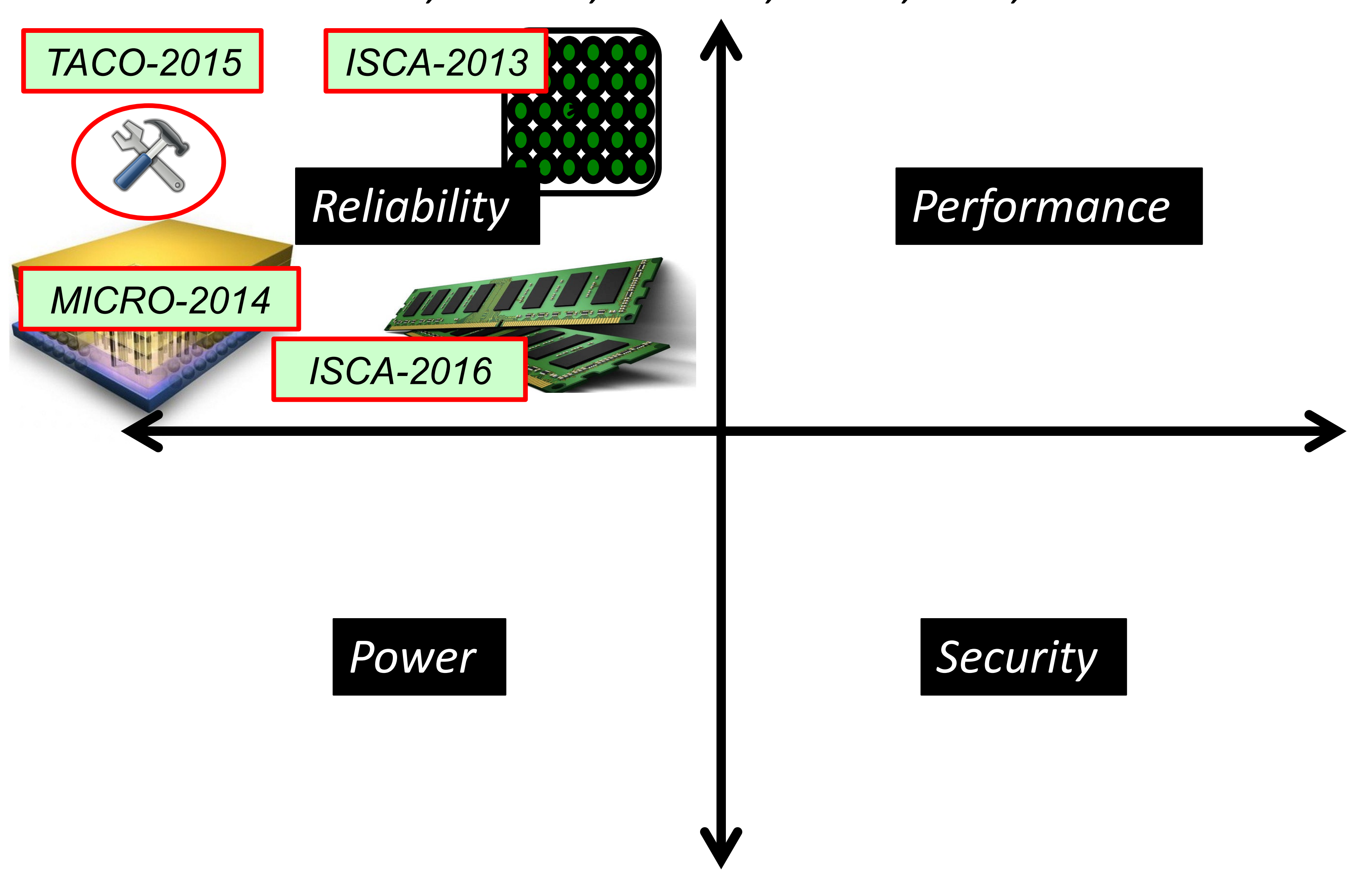
Performance

Power

Security

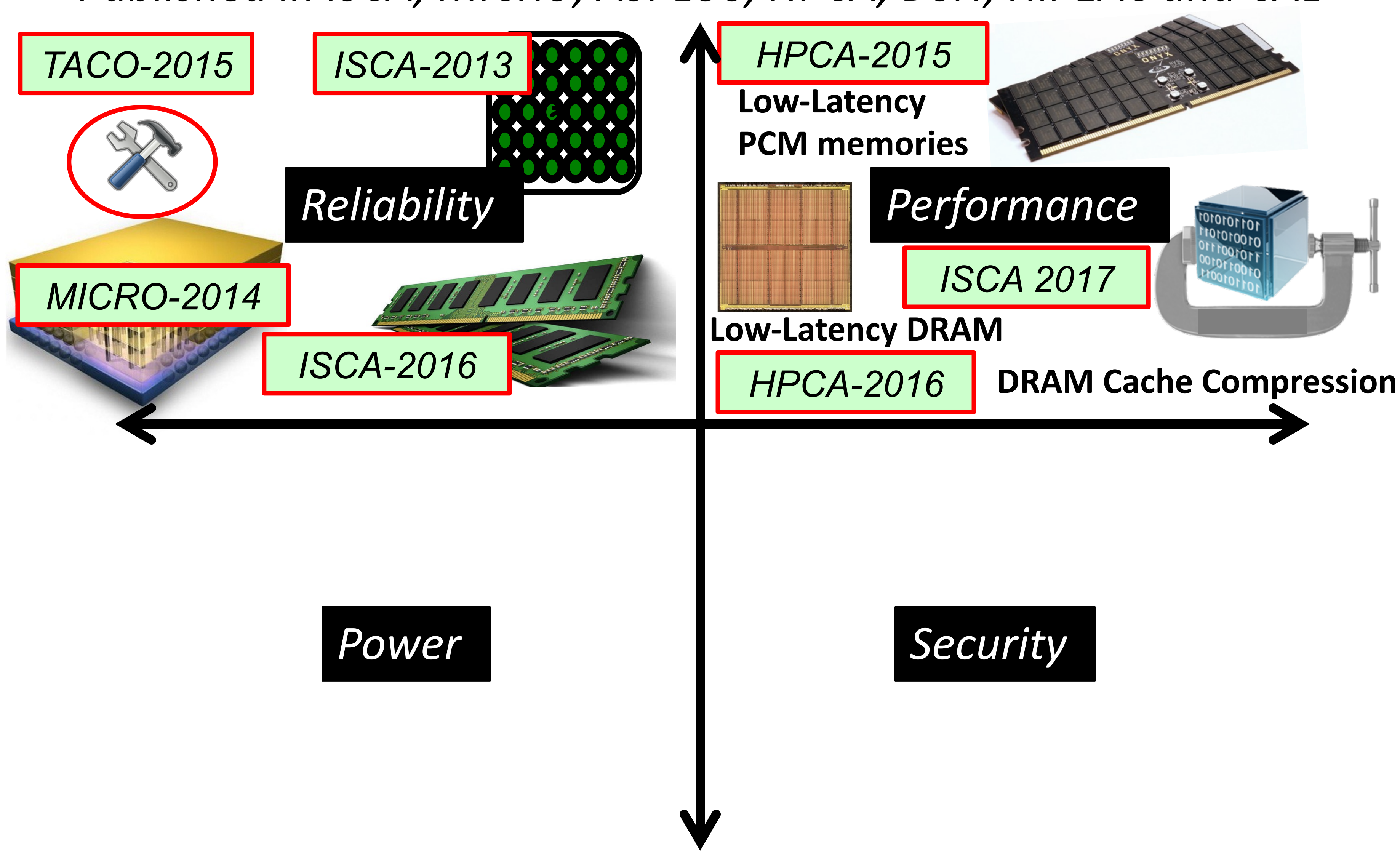
AREAS INVESTIGATED

Published in ISCA, MICRO, ASPLOS, HPCA, DSN, HiPEAC and CAL

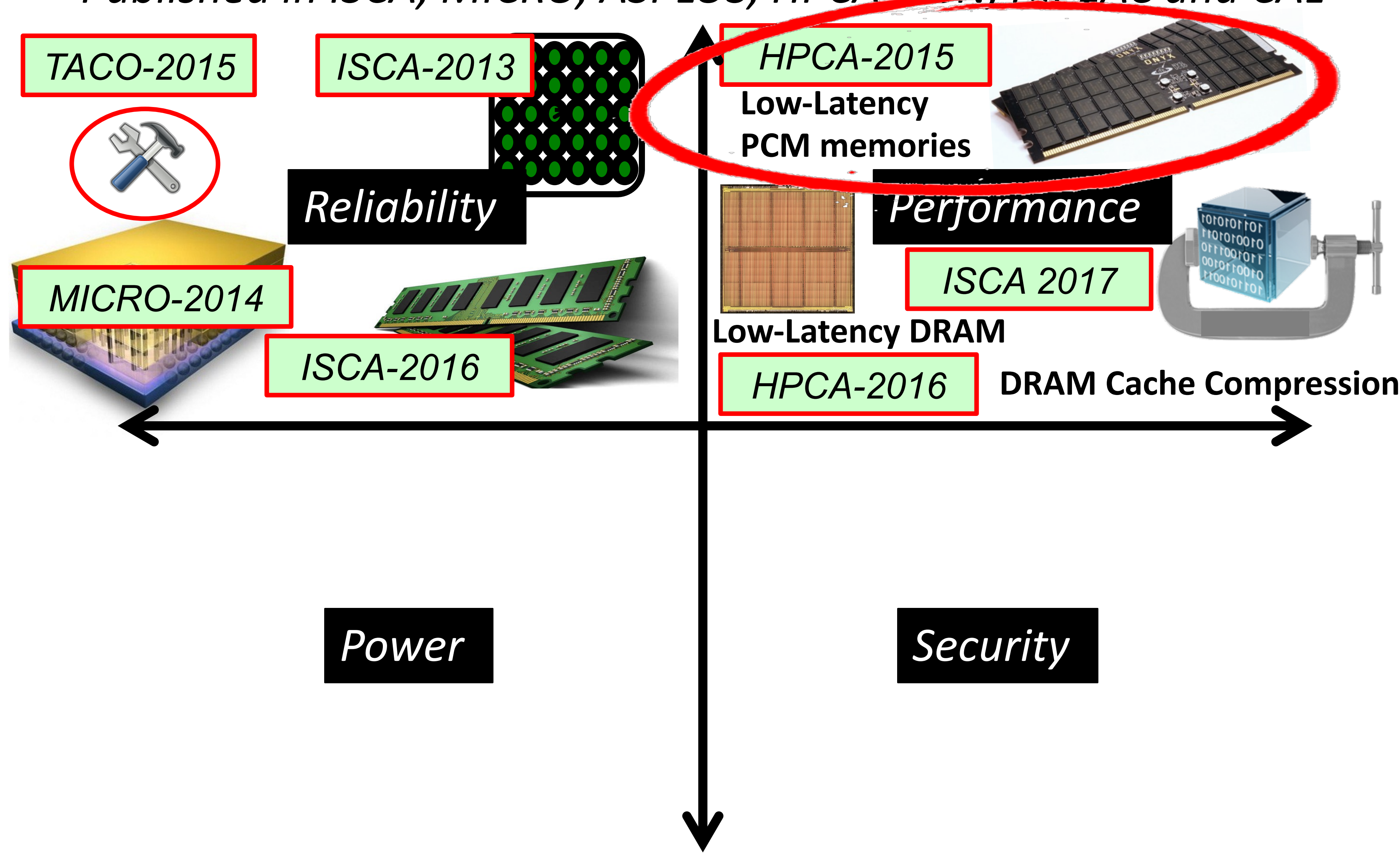


AREAS INVESTIGATED

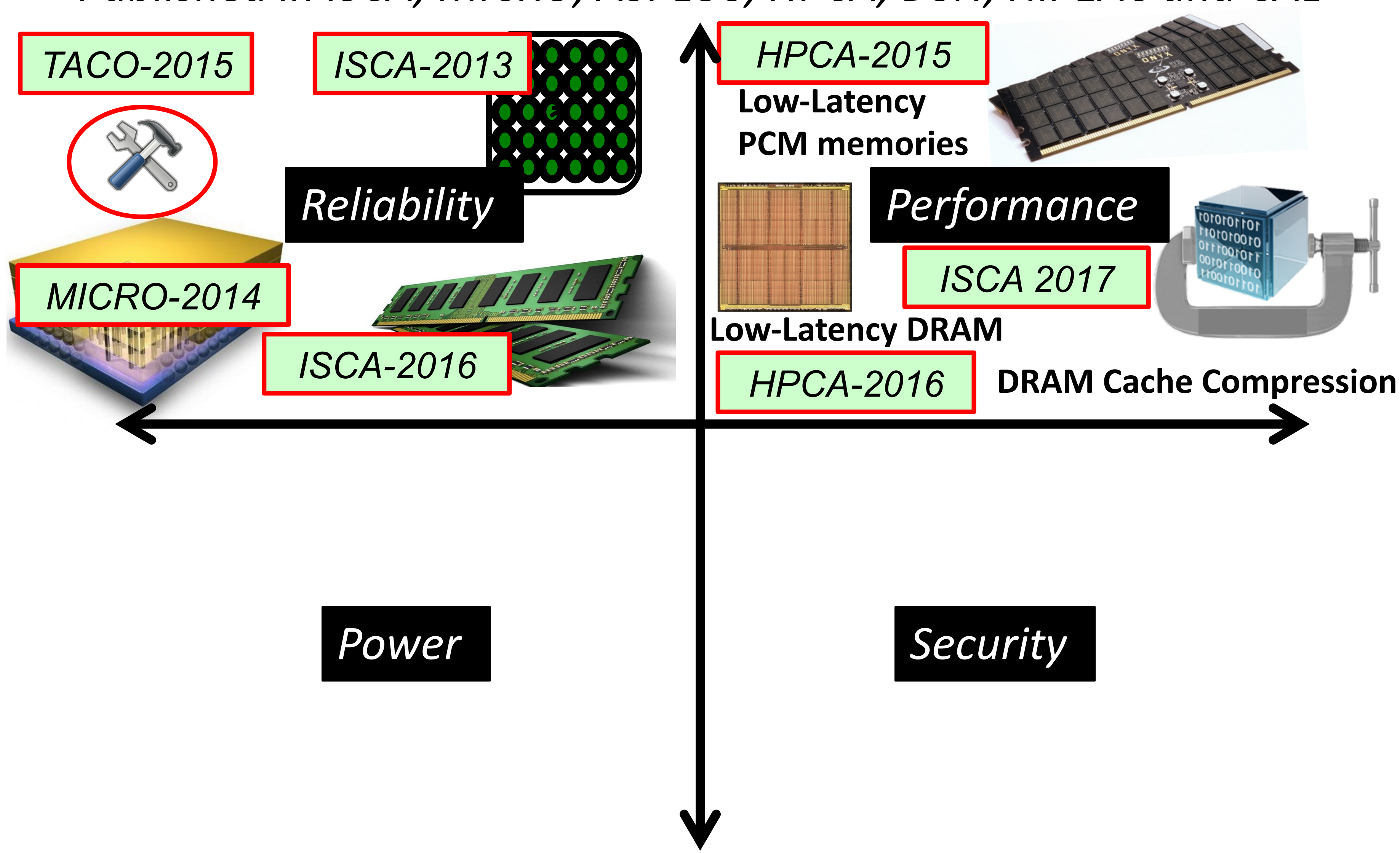
Published in ISCA, MICRO, ASPLOS, HPCA, DSN, HiPEAC and CAL



AREAS INVESTIGATED

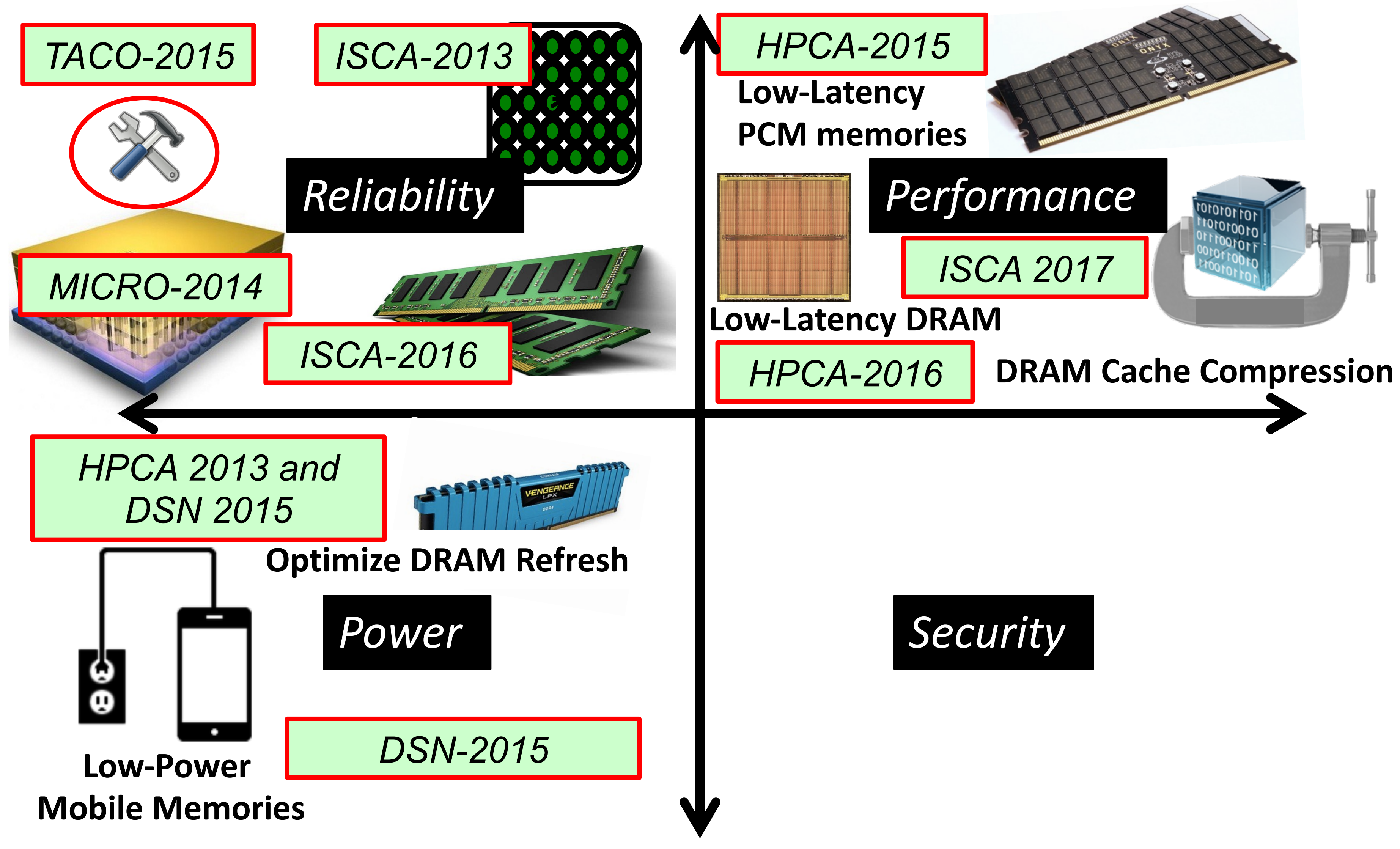


AREAS INVESTIGATED



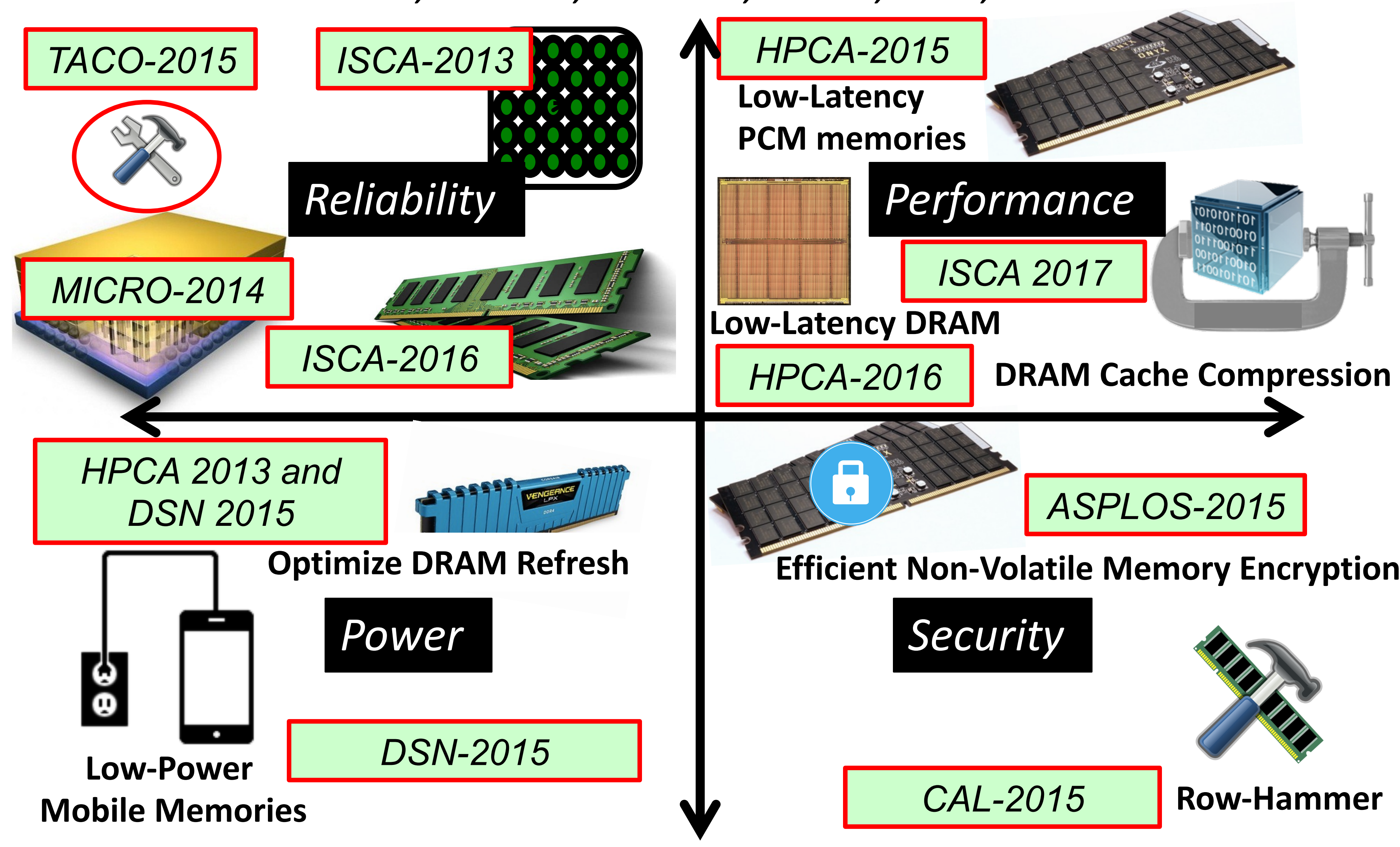
AREAS INVESTIGATED

Published in ISCA, MICRO, ASPLOS, HPCA, DSN, HiPEAC and CAL



AREAS INVESTIGATED

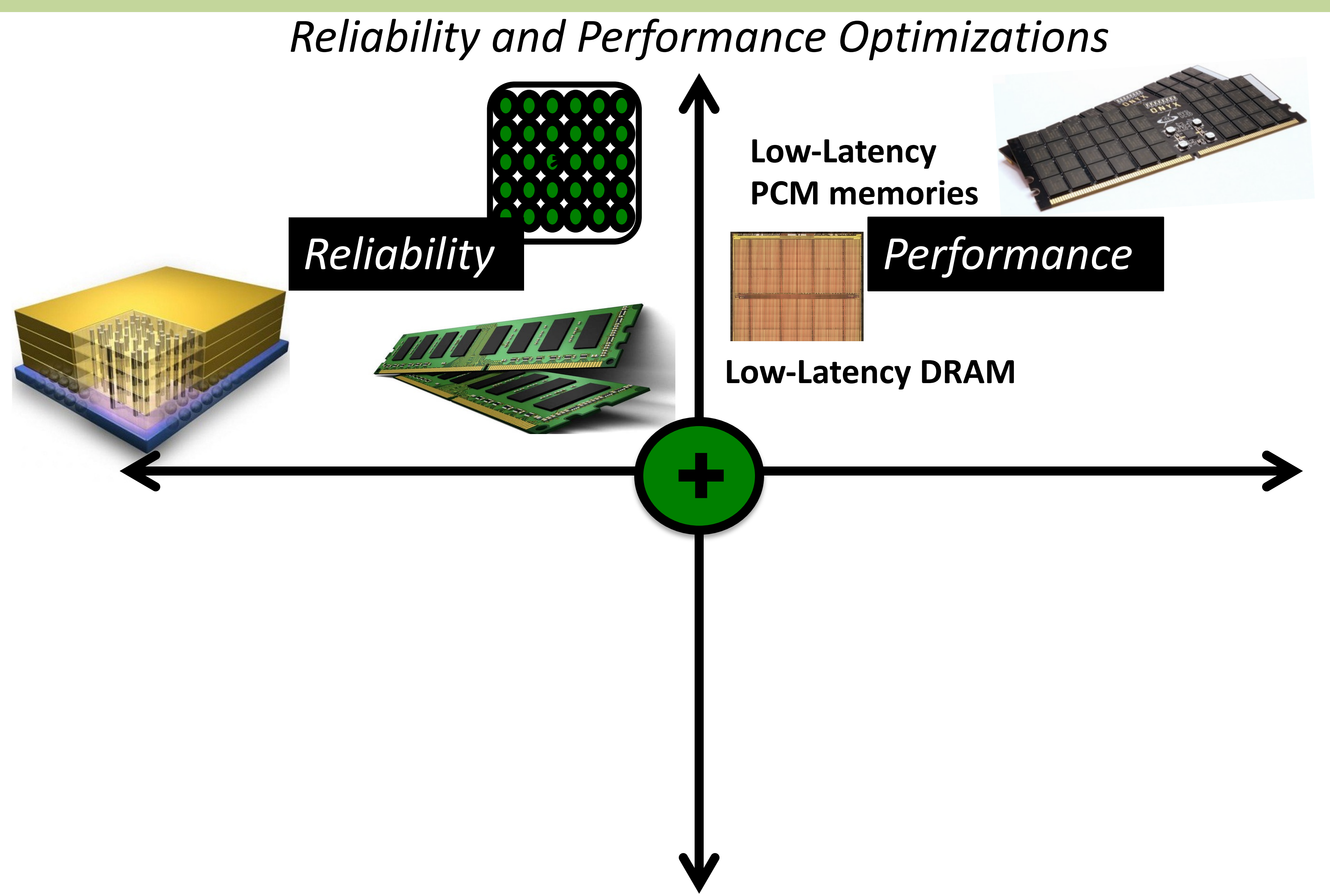
Published in ISCA, MICRO, ASPLOS, HPCA, DSN, HiPEAC and CAL



FUTURE RESEARCH VECTORS

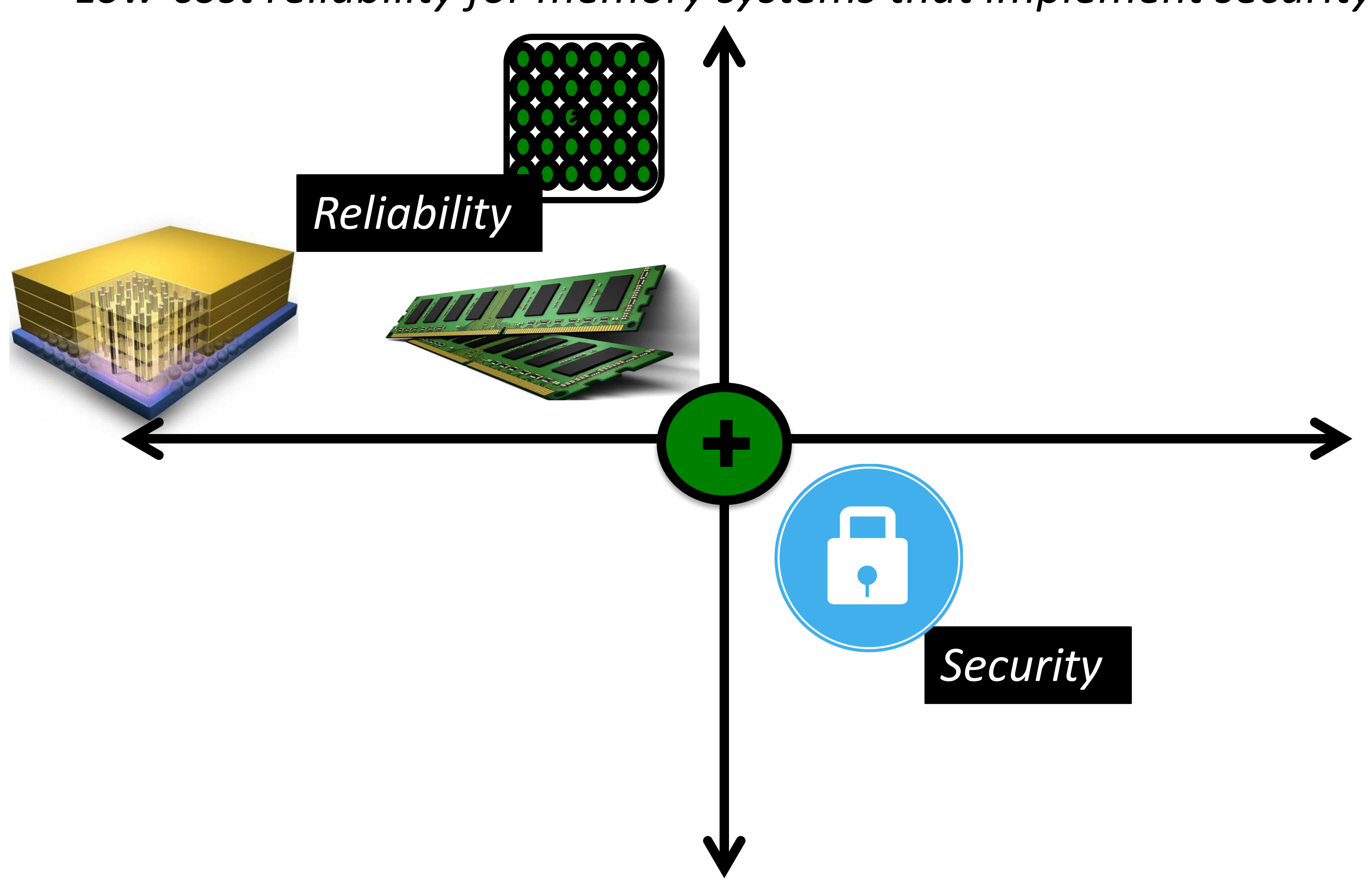
RESEARCH VECTOR: HYBRID MEMORY SYSTEM

Reliability and Performance Optimizations



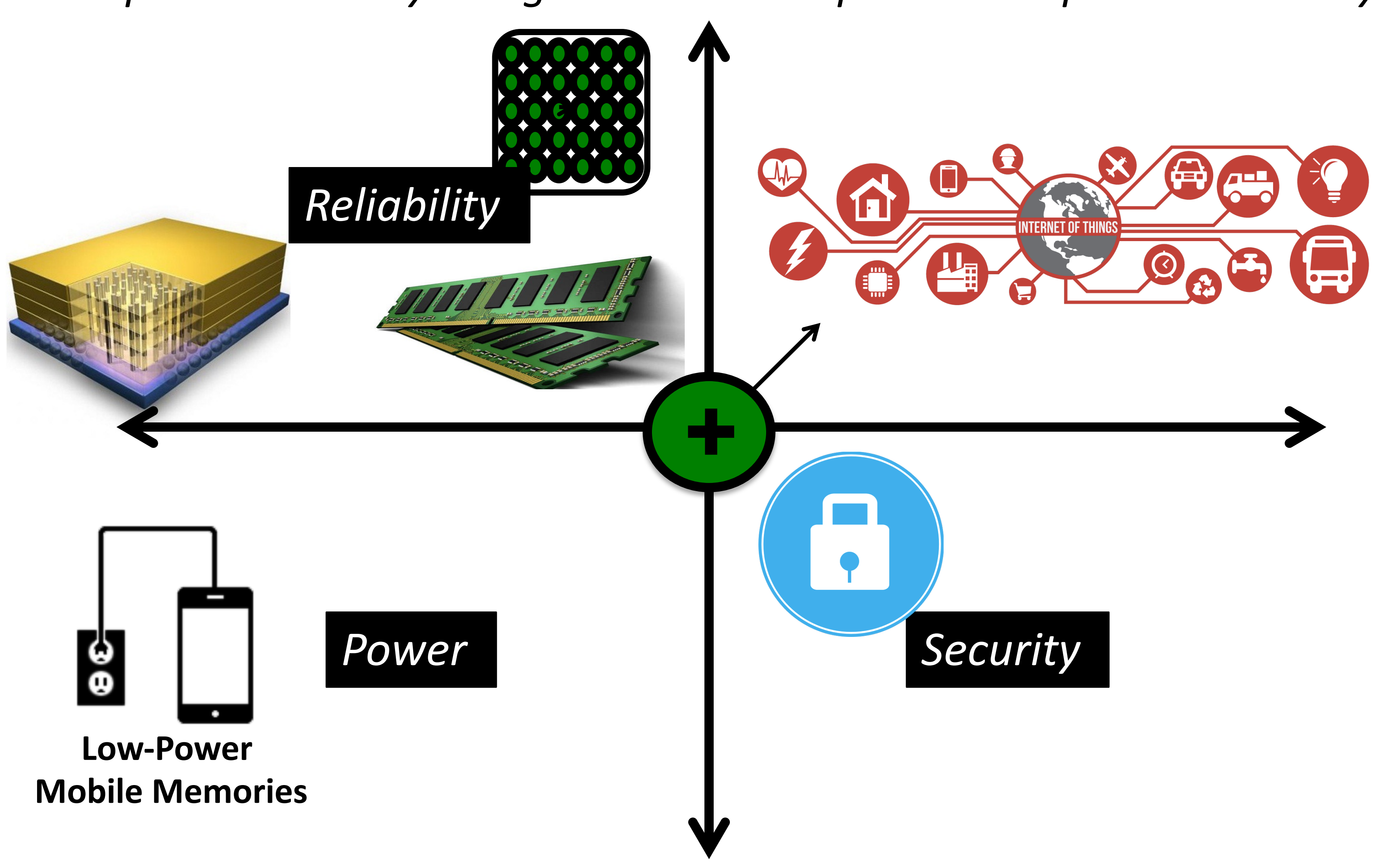
RESEARCH VECTOR: RELIABILITY + SECURITY

Low-cost reliability for memory systems that implement security

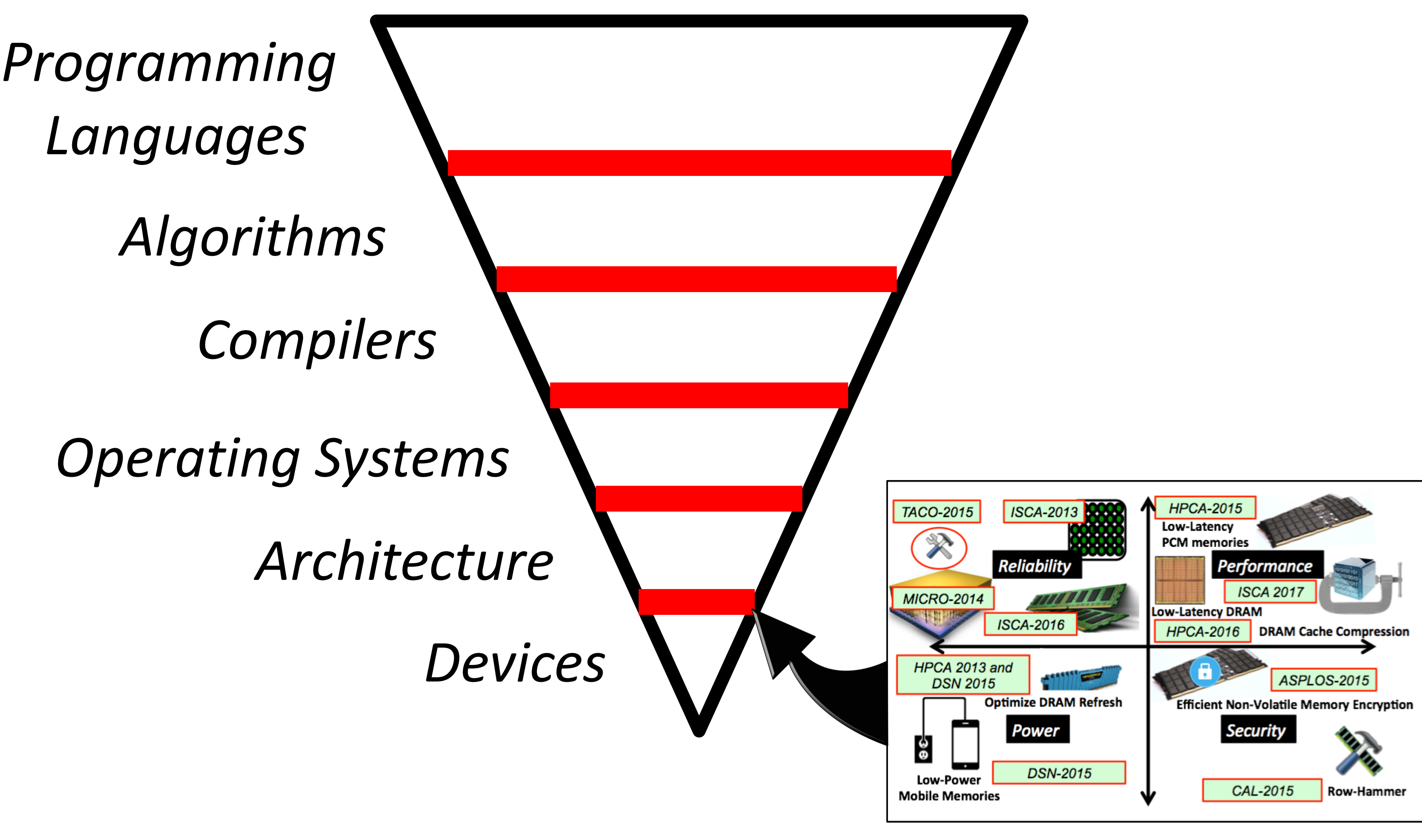


RESEARCH VECTOR: OPTIMIZED IoT

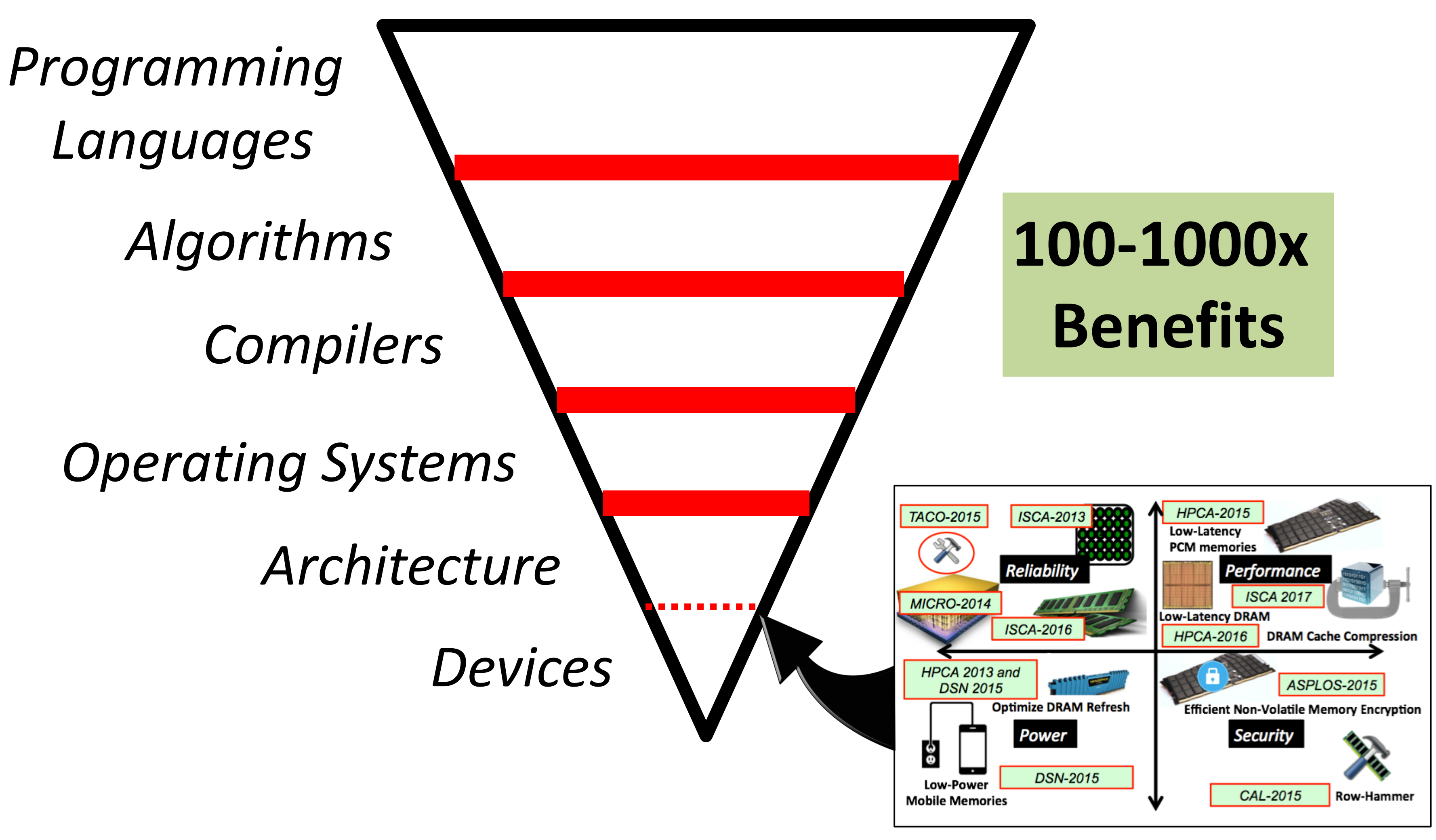
IoT optimizations by using codes to save power and provide security



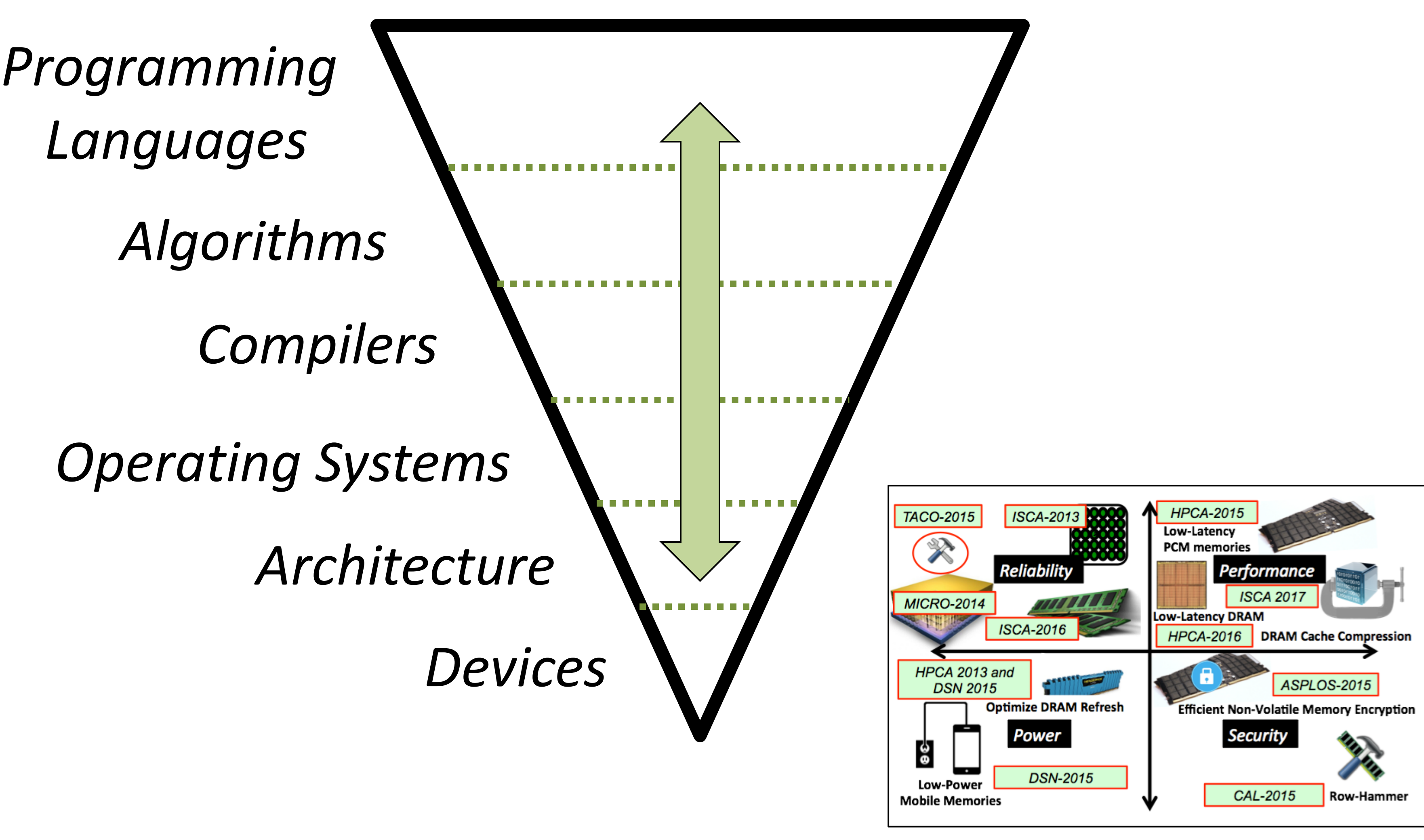
RESEARCH VECTOR: CROSS-STACK STRATEGIES



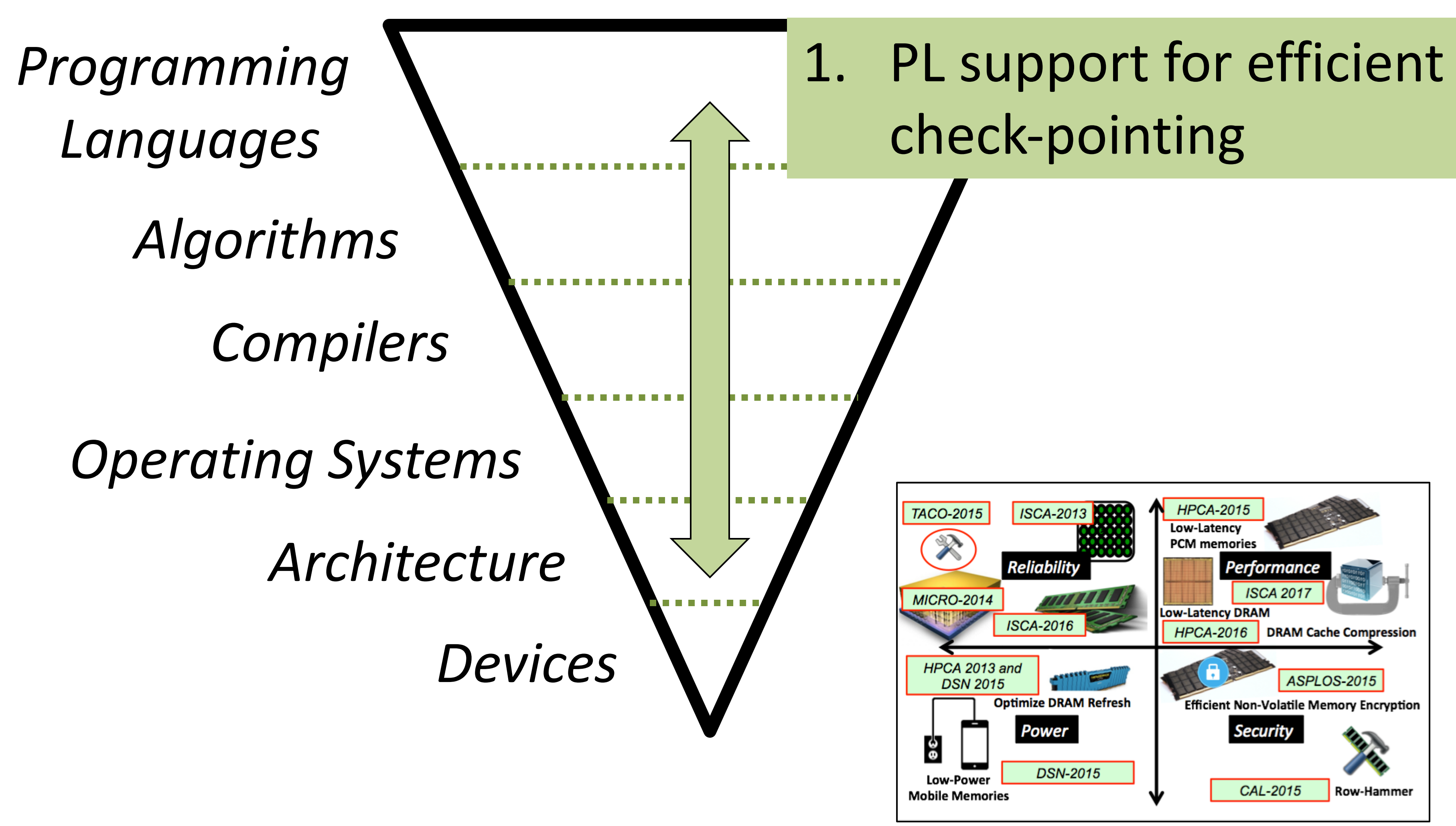
RESEARCH VECTOR: CROSS-STACK STRATEGIES



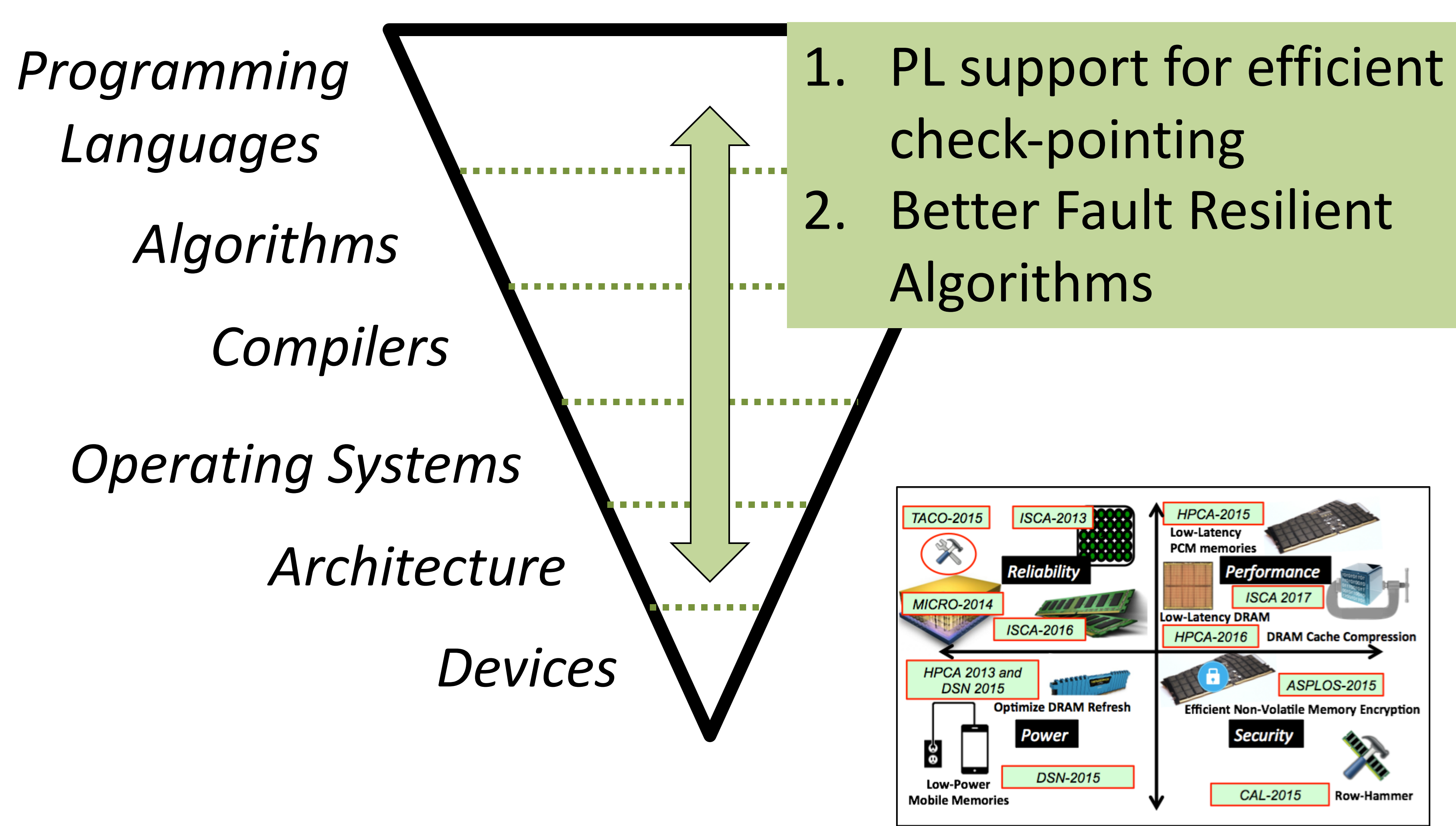
RESEARCH VECTOR: CROSS-STACK STRATEGIES



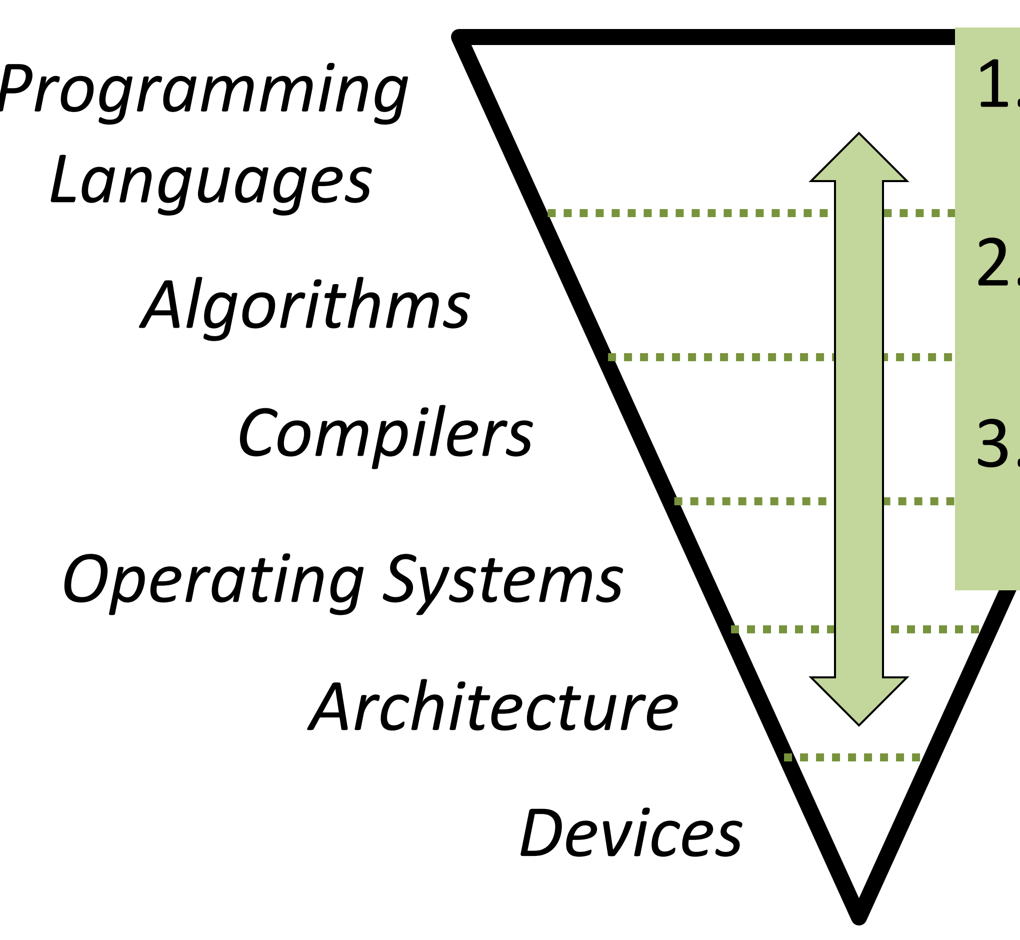
RESEARCH VECTOR: CROSS-STACK STRATEGIES



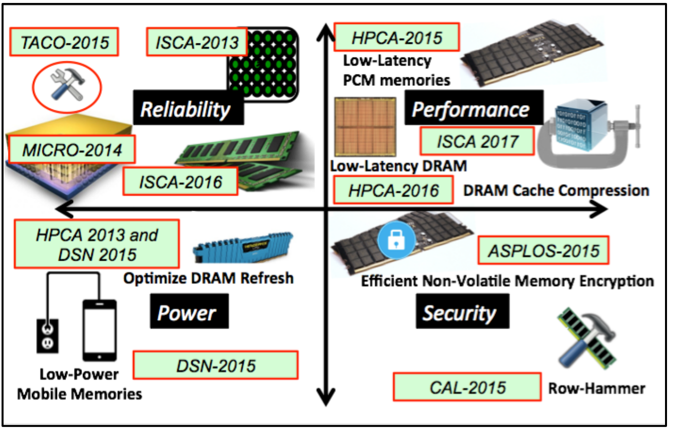
RESEARCH VECTOR: CROSS-STACK STRATEGIES



RESEARCH VECTOR: CROSS-STACK STRATEGIES



1. PL support for efficient check-pointing
2. Better Fault Resilient Algorithms
3. Strong systems with OS-level reliability



RESEARCH VECTOR: CROSS-STACK STRATEGIES

Programming Languages

Algorithms

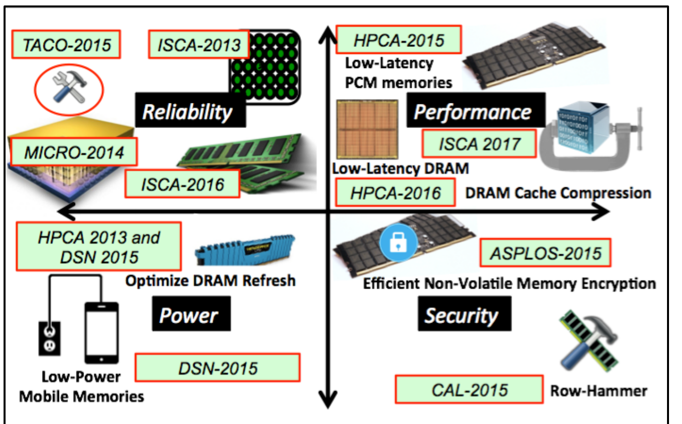
Compilers

Operating Systems

Architecture

Devices

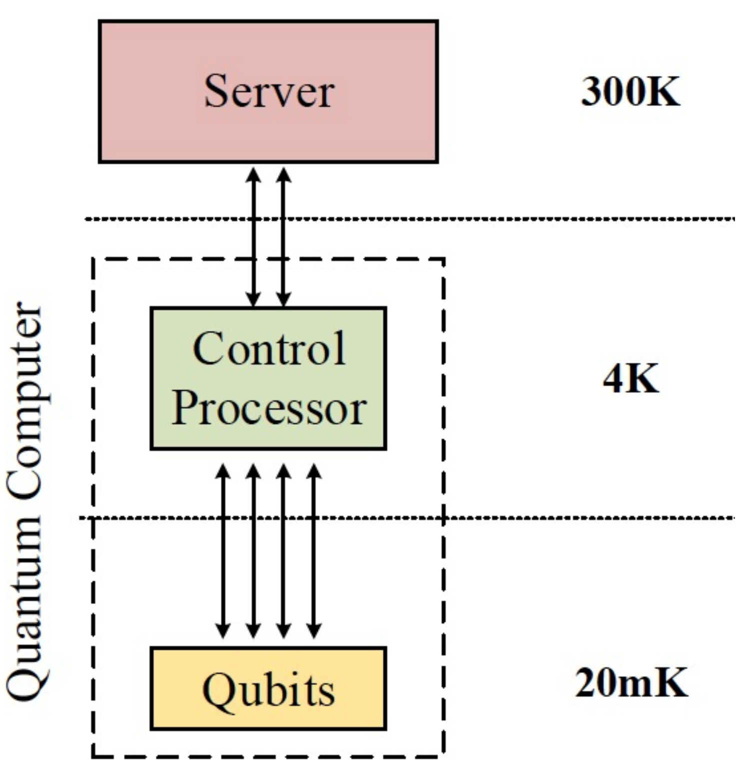
1. PL support for efficient check-pointing
2. Better Fault Resilient Algorithms
3. Strong systems with OS-level reliability



Unlock more benefits with cross-stack solutions

RESEARCH VECTOR: QUANTUM COMPUTERS

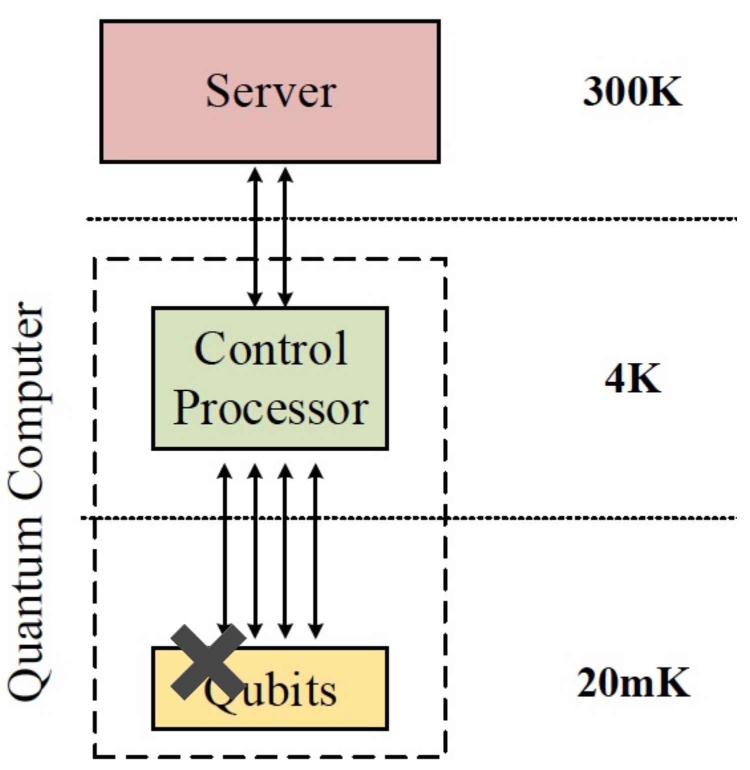
Enabling Practical and Scalable Quantum Computers



Quantum Substrate has a high error-rate

RESEARCH VECTOR: QUANTUM COMPUTERS

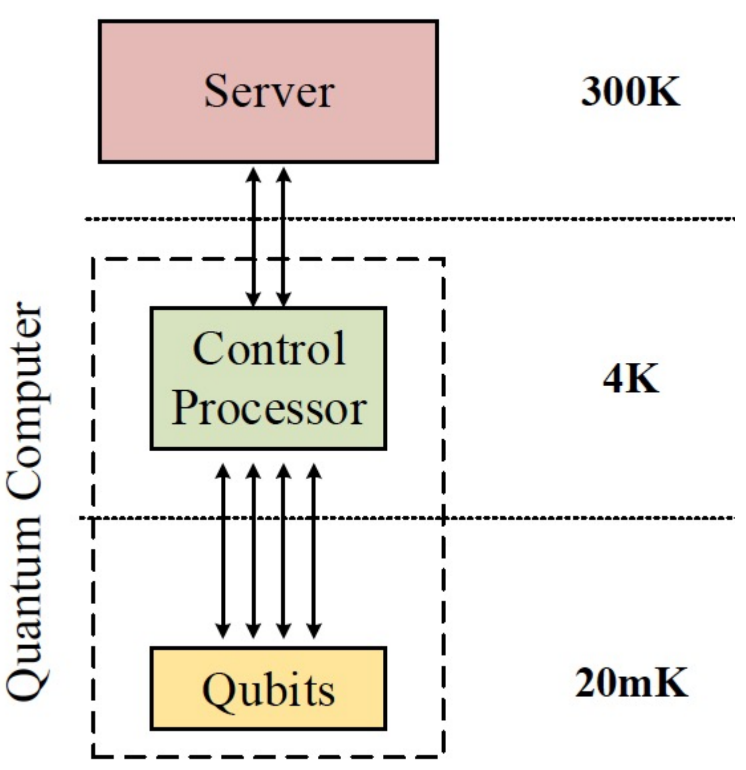
Enabling Practical and Scalable Quantum Computers



Quantum Substrate has a high error-rate

RESEARCH VECTOR: QUANTUM COMPUTERS

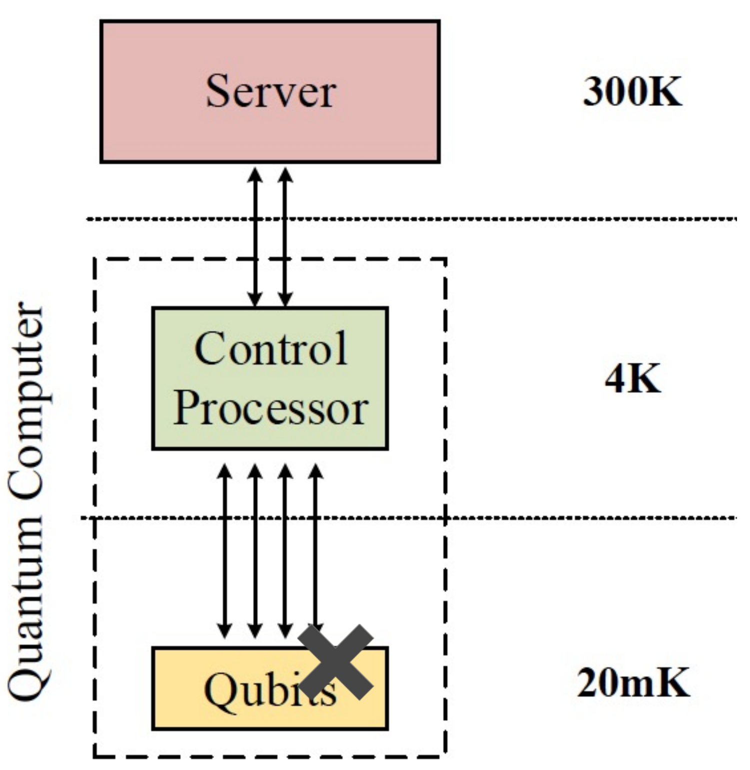
Enabling Practical and Scalable Quantum Computers



Quantum Substrate has a high error-rate

RESEARCH VECTOR: QUANTUM COMPUTERS

Enabling Practical and Scalable Quantum Computers



Quantum Substrate has a high error-rate

RESEARCH VECTOR: QUANTUM COMPUTERS

Enabling Practical and Scalable Quantum Computers

Temperature

Server 300K

Quantum Computer

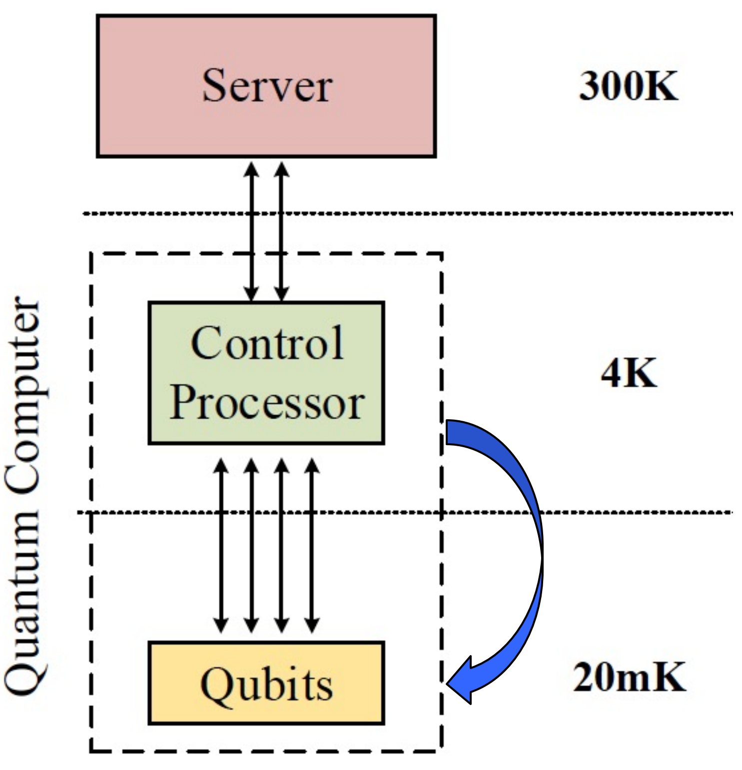
Control Processor 4K

Qubits 20mK

1000x higher bandwidth overhead due to error correction

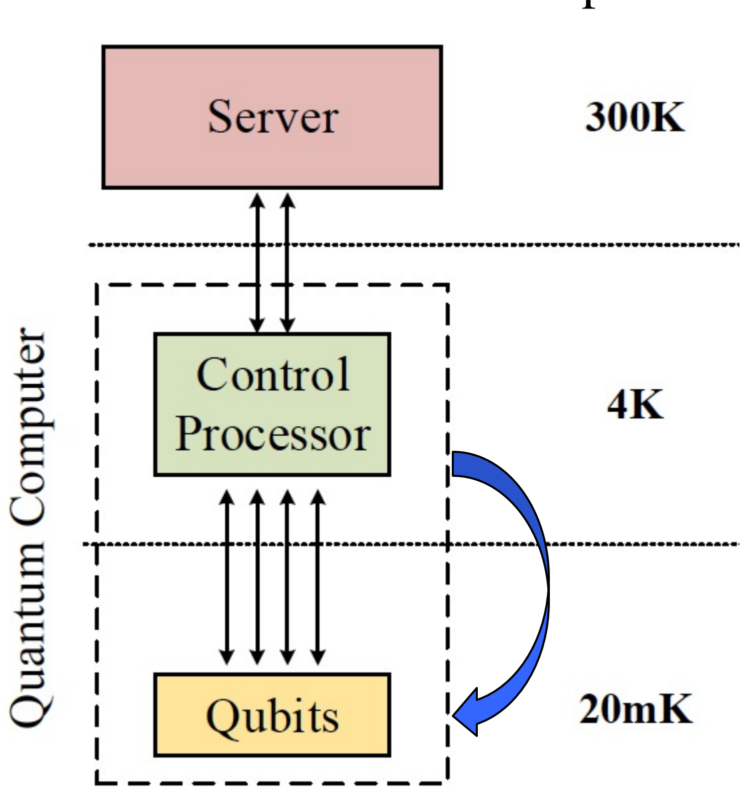
RESEARCH VECTOR: QUANTUM COMPUTERS

Enabling Practical and Scalable Quantum Computers



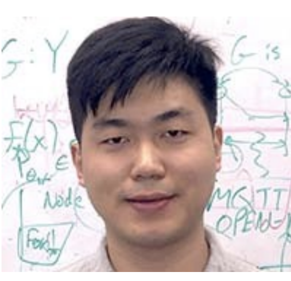
RESEARCH VECTOR: QUANTUM COMPUTERS

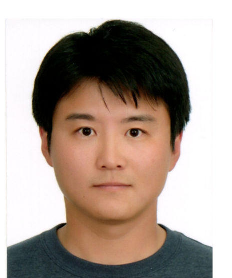
Enabling Practical and Scalable Quantum Computers

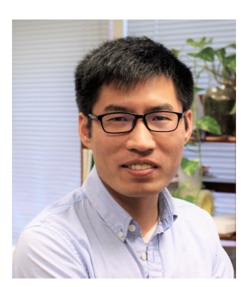


Ways to delegate ECC near the quantum substrate

COLLABORATORS

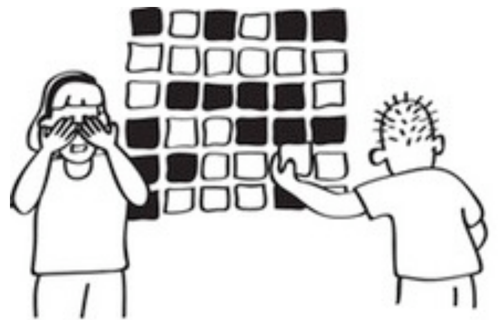


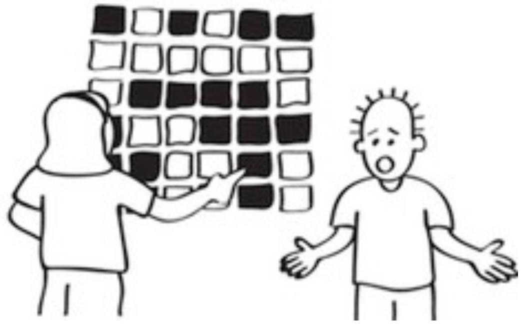




THANK YOU

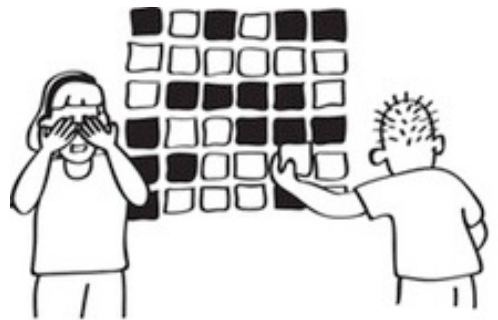
Always welcome to reach out to me at pnair6@gatech.edu

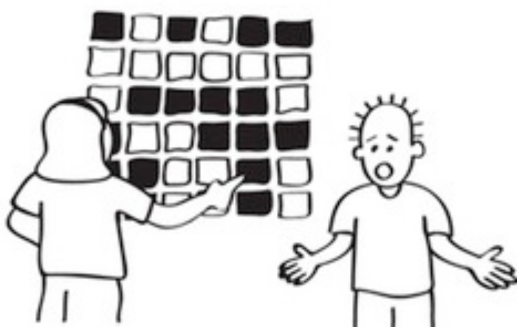




THANK YOU

Always welcome to reach out to me at pnair6@gatech.edu





BACKUP

Reducing Read Latency of Phase Change Memory via Early Read and Turbo Read

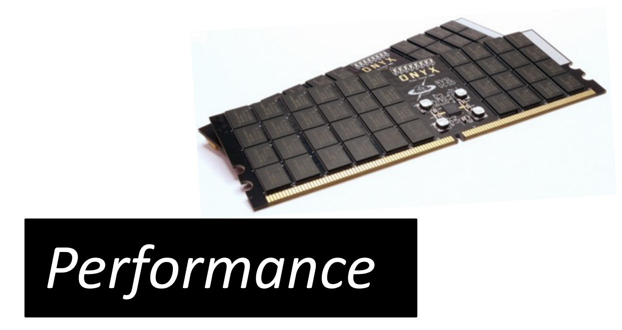
HPCA-2015

Prashant Nair

Chia-Chen Chou

Bipin Rajendran

Moinuddin Qureshi

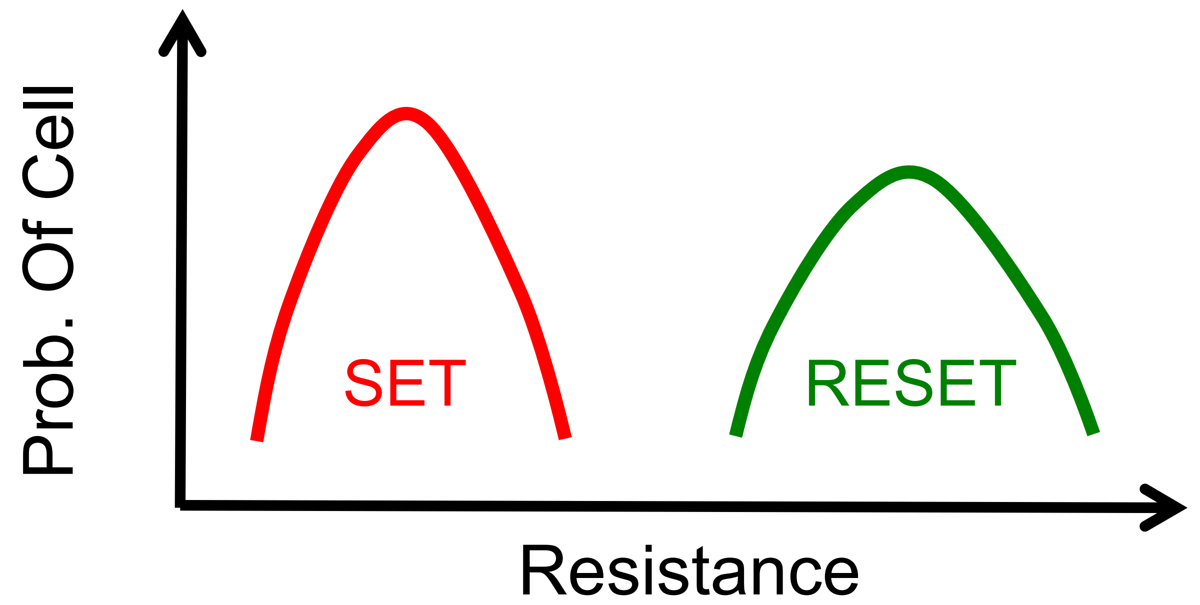


Performance

"Low-Latency Non-Volatile memories using Error Codes"

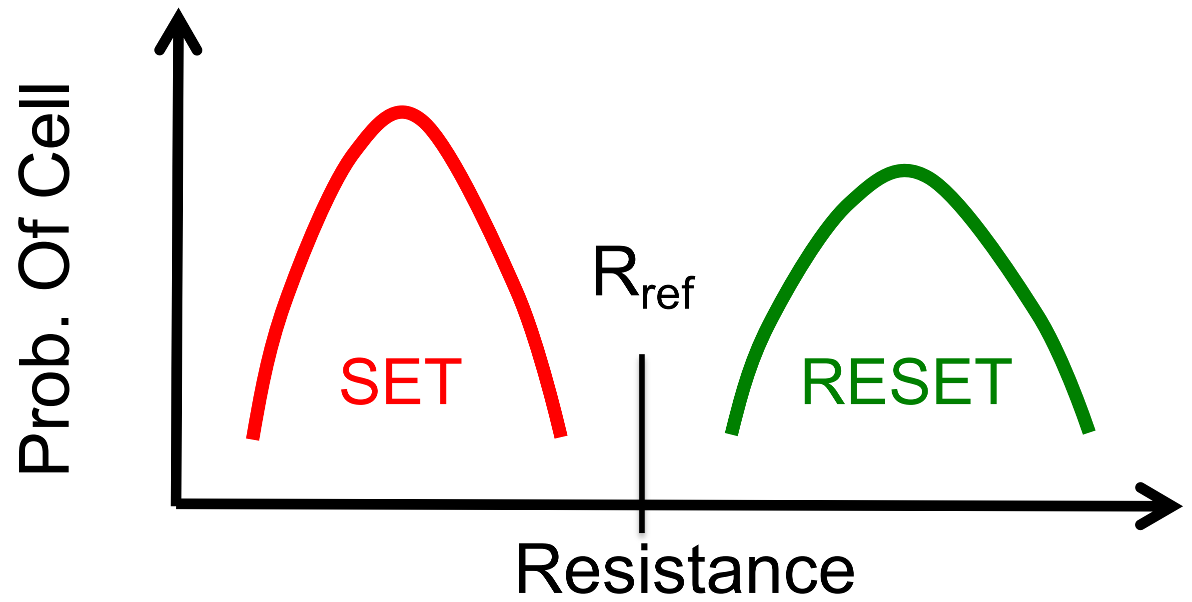
EARLY & TURBO READ: VARIABILITY IN PCM

- Low (SET) and High (RESET) resistance states



EARLY & TURBO READ: VARIABILITY IN PCM

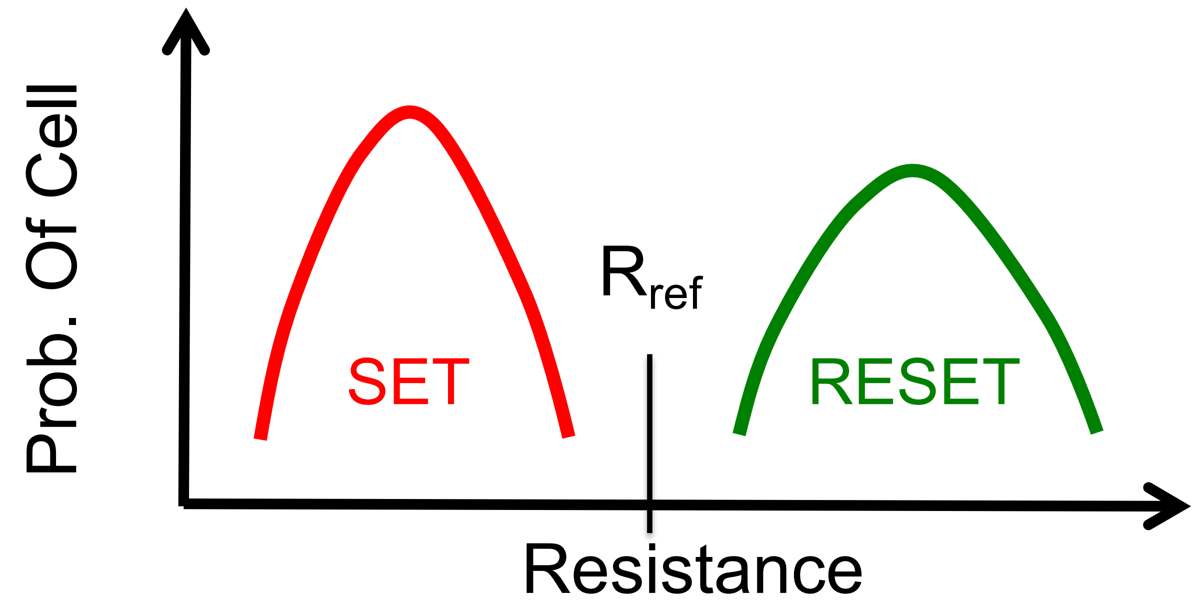
- Low (SET) and High (RESET) resistance states



- Cell states are compared to reference resistance

EARLY & TURBO READ: VARIABILITY IN PCM

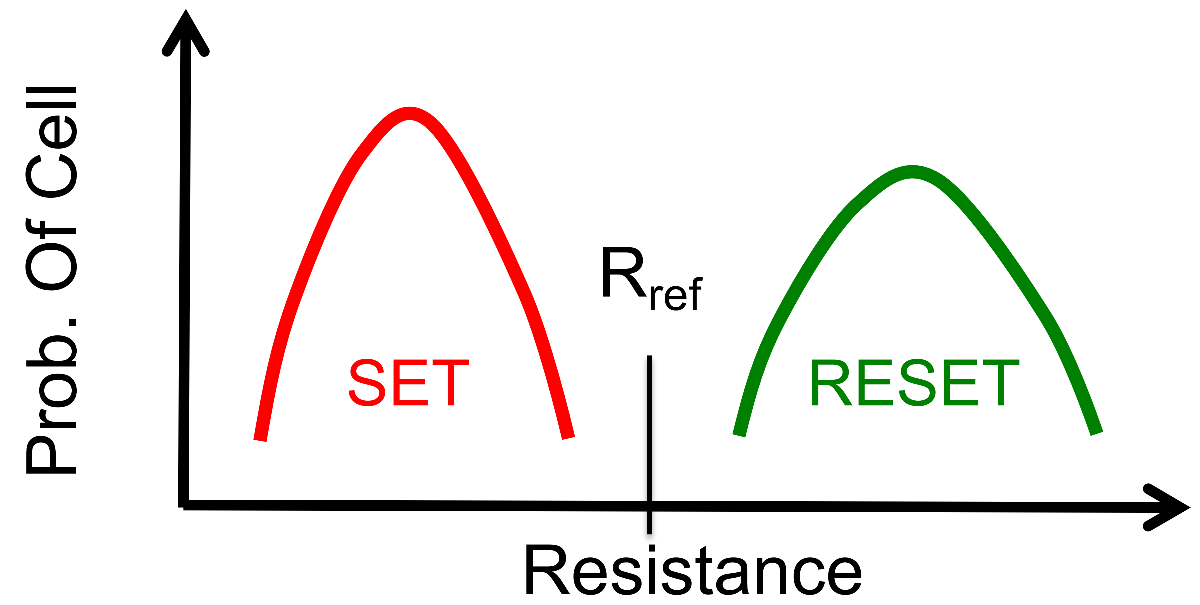
- Low (SET) and High (RESET) resistance states



- Cell states are compared to reference resistance
- The states correspond to binary values of 0 and 1

EARLY & TURBO READ: VARIABILITY IN PCM

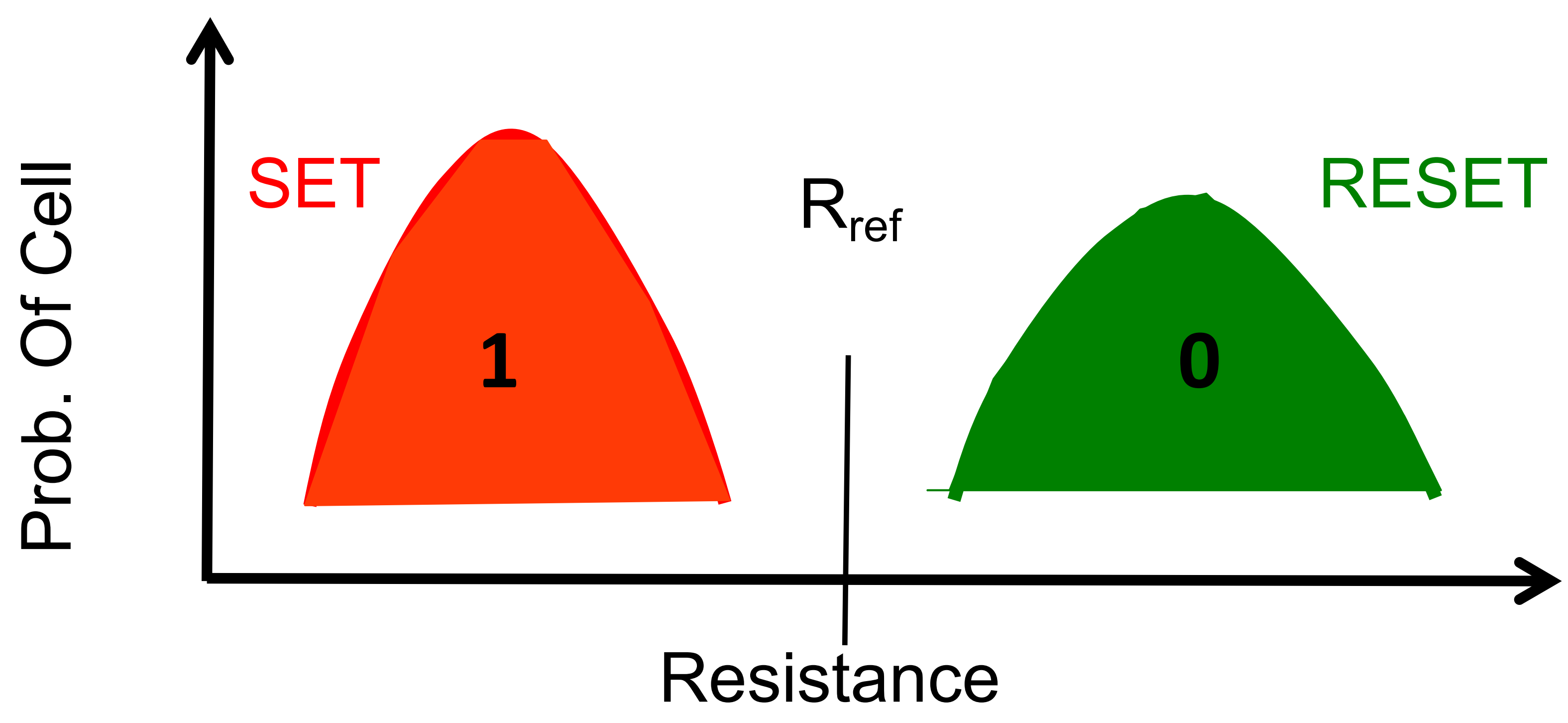
- Low (SET) and High (RESET) resistance states



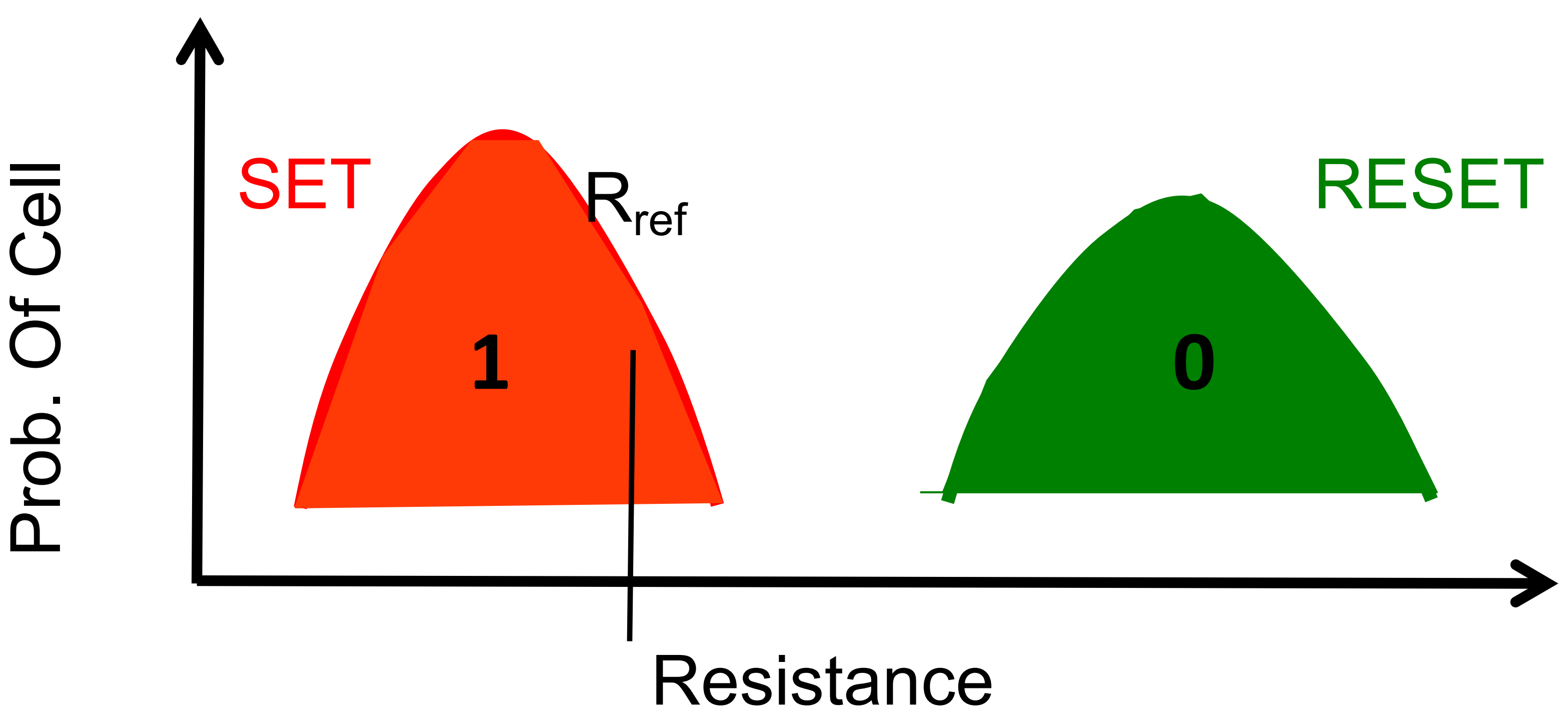
- Cell states are compared to reference resistance
- The states correspond to binary values of 0 and 1

The read latency of PCM depends on value of R_{ref}

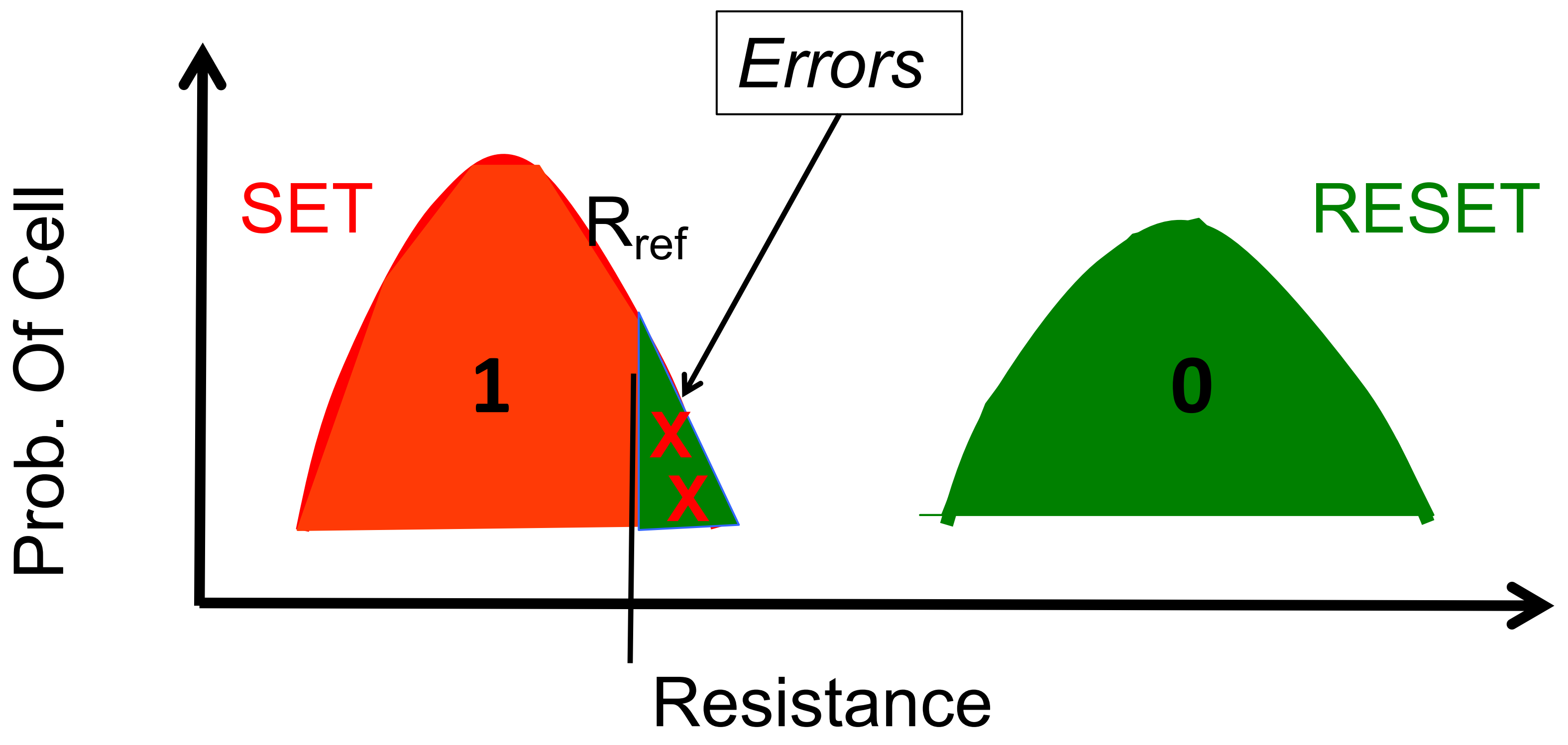
EARLY & TURBO READ: SENSING EARLIER



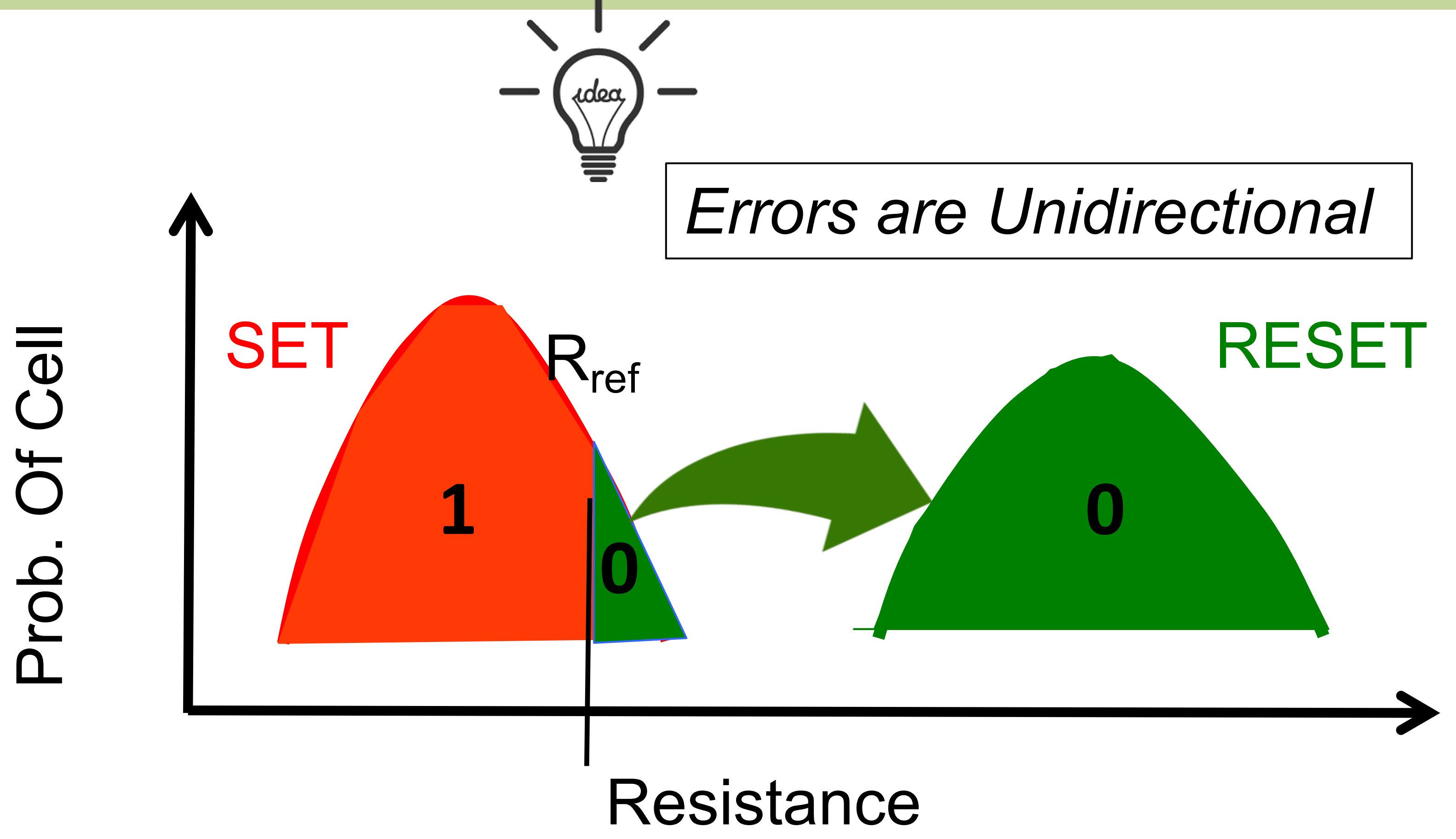
EARLY & TURBO READ: SENSING EARLIER



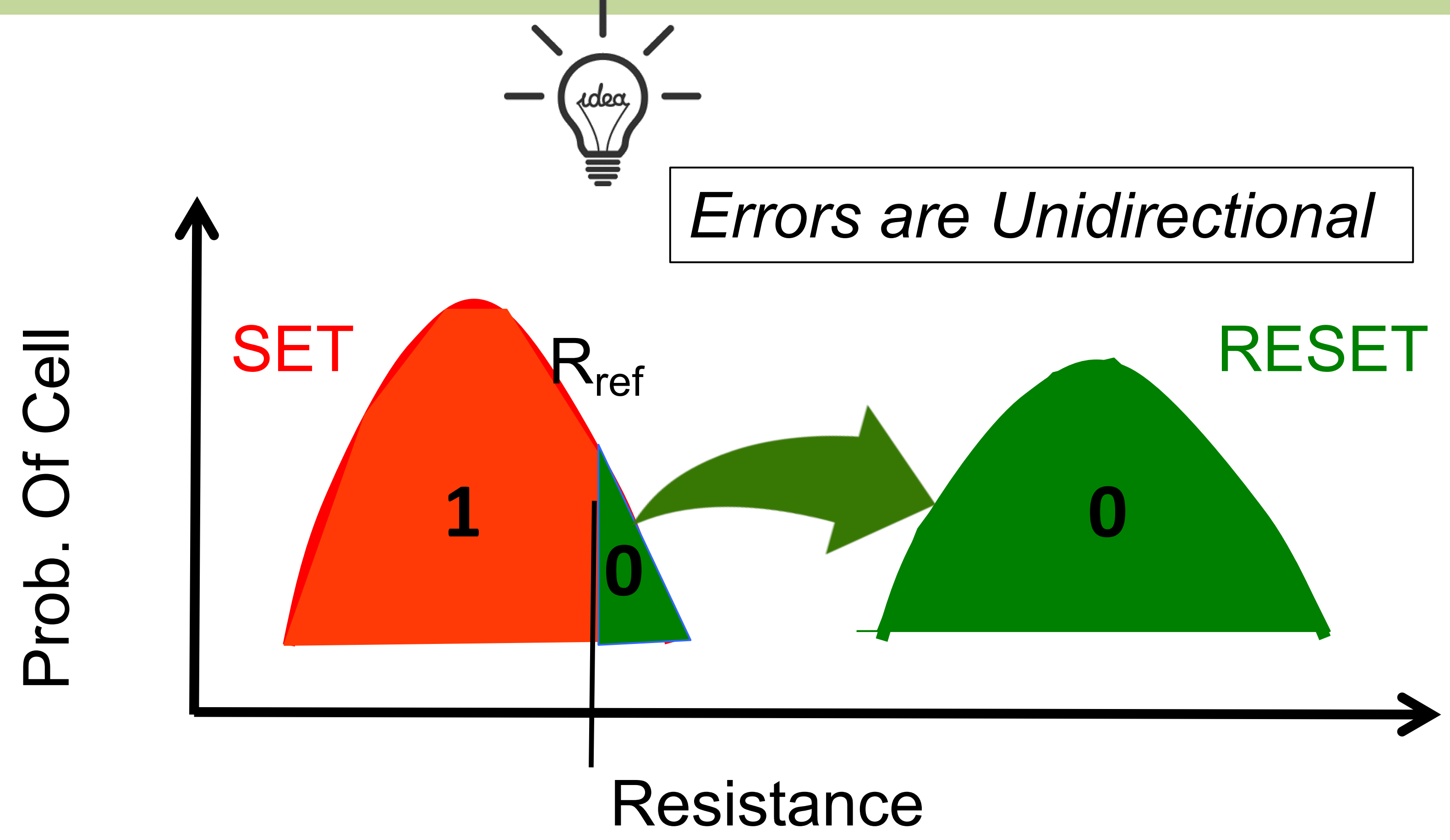
EARLY & TURBO READ: SENSING EARLIER



EARLY & TURBO READ: SENSING EARLIER

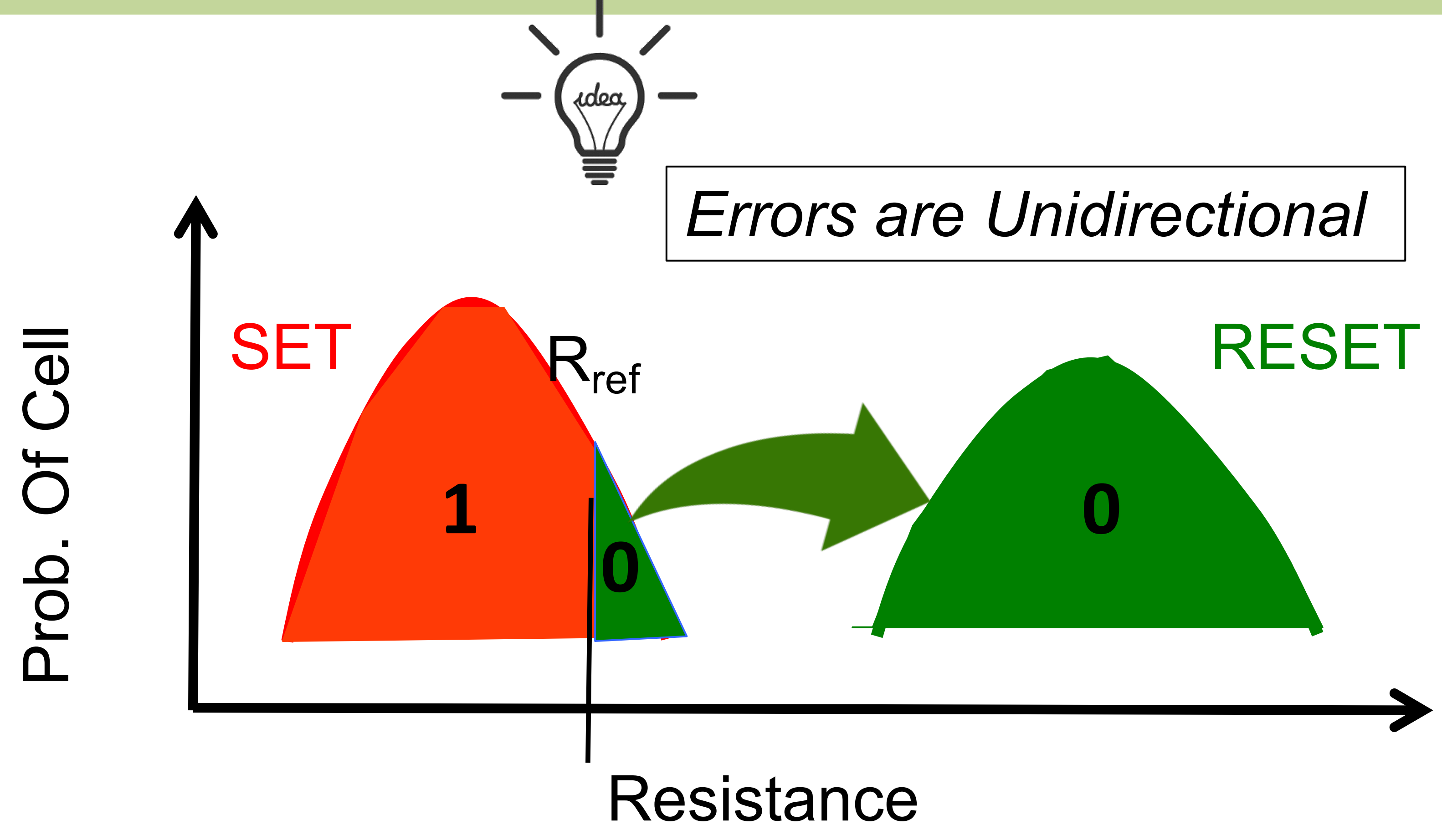


EARLY & TURBO READ: SENSING EARLIER



Detect with Berger Code, Retry on error

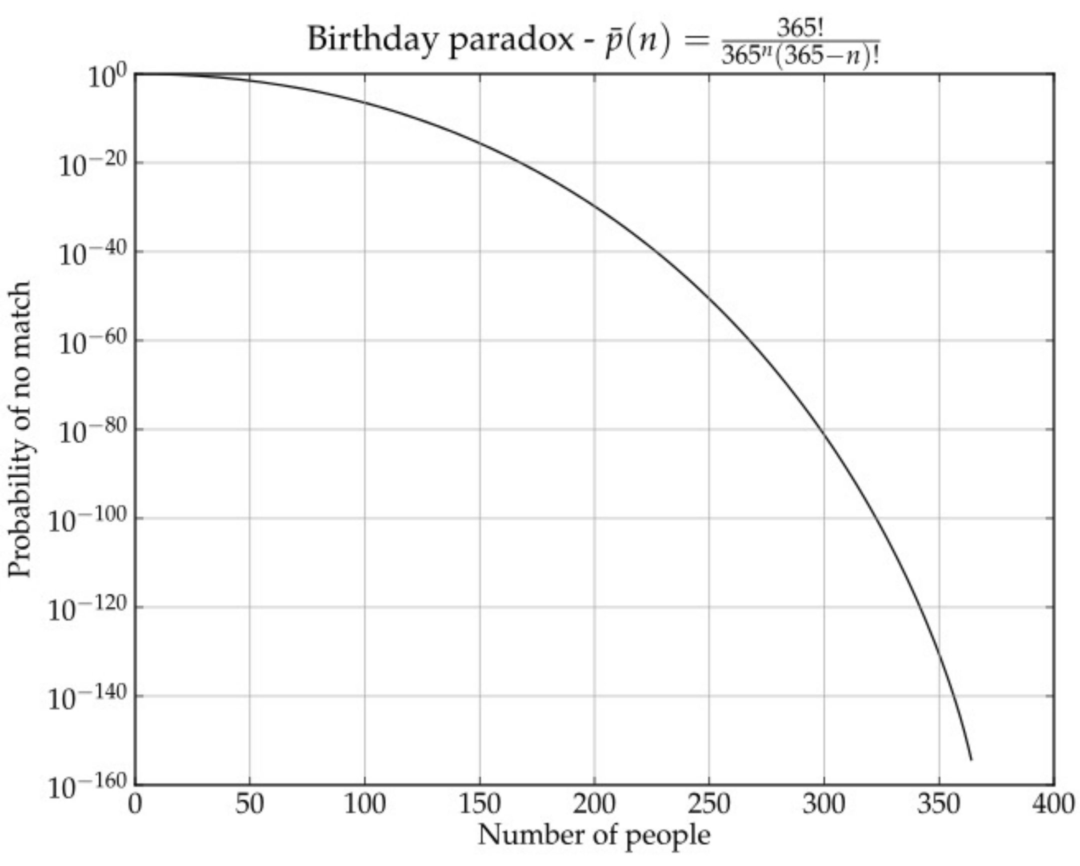
EARLY & TURBO READ: SENSING EARLIER



Detect with Berger Code, Retry on error

Read latency 30% ↓ → Performance 26% ↑

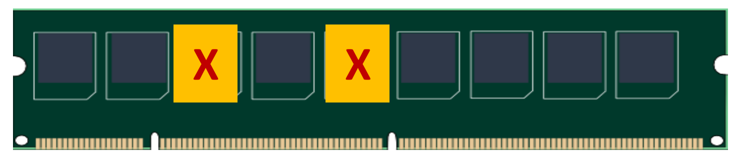
BIRTHDAY PROBLEM

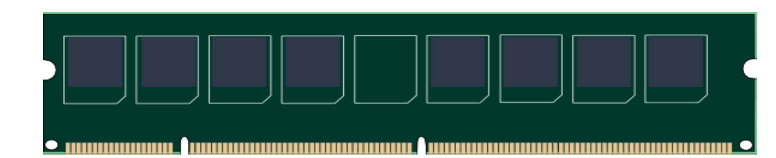


MTTF: XED VS CHIPKILL

2-Chip Failures

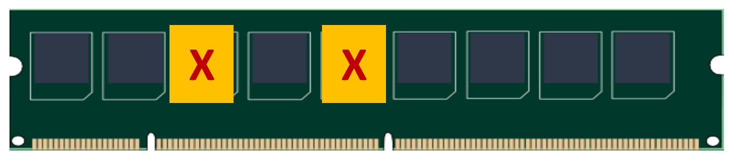
XED (9-chips)

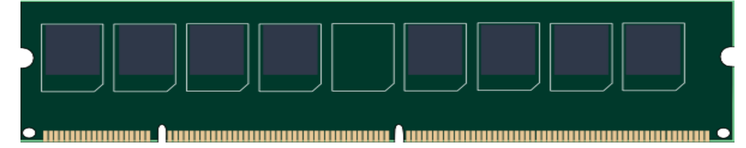




Chipkill (18-chips)

FAILED

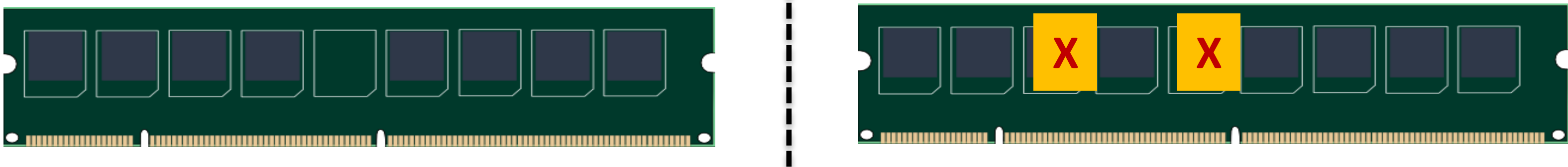




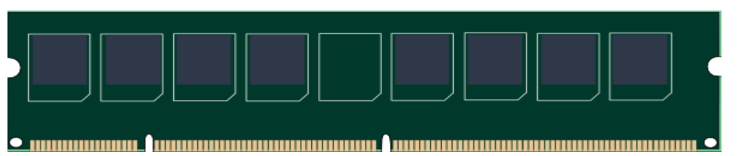
MTTF: XED VS CHIPKILL

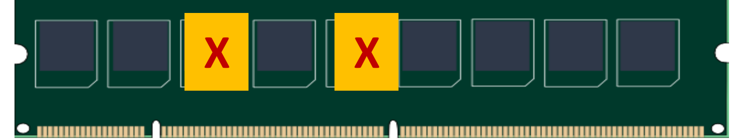
2-Chip Failures

XED (9-chips)



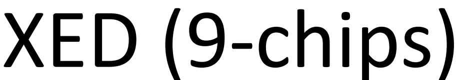
Chipkill

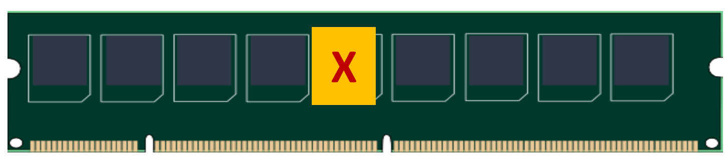


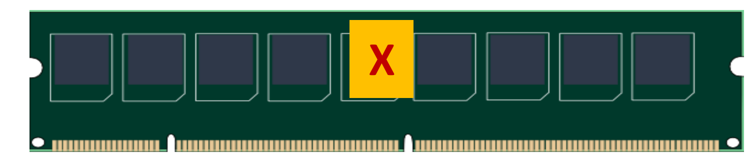


MTTF: XED VS CHIPKILL

2-Chip Failures

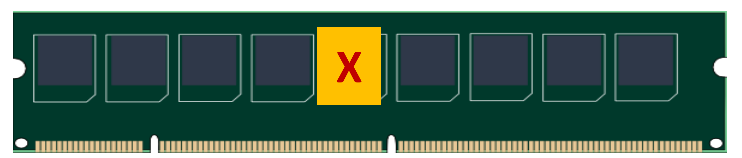


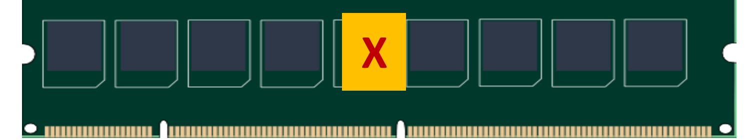




Chipkill

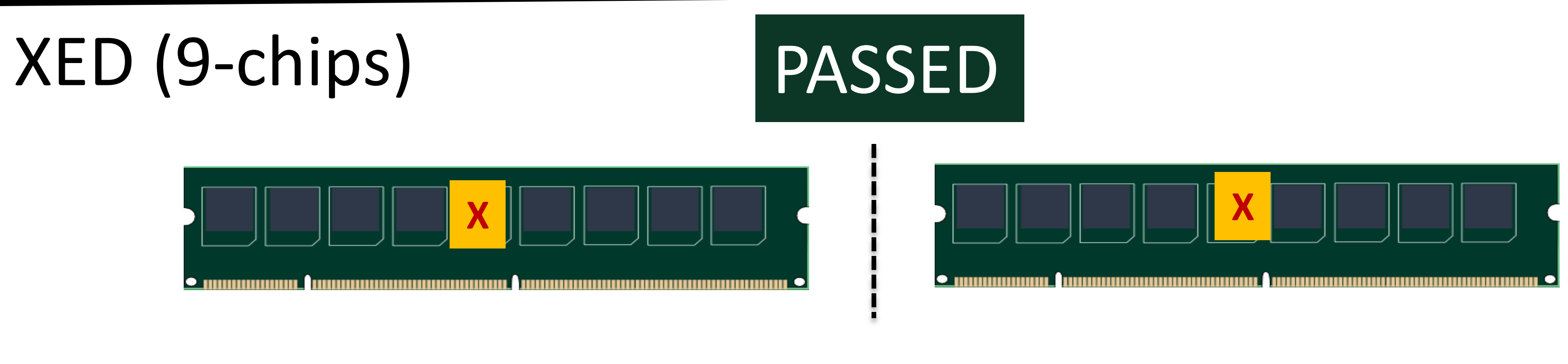
FAILED



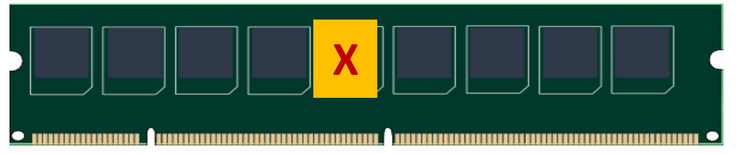


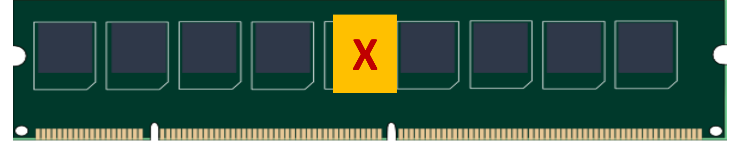
MTTF: XED VS CHIPKILL

2-Chip Failures → Extend to Multi-Chip Failures



Chipkill





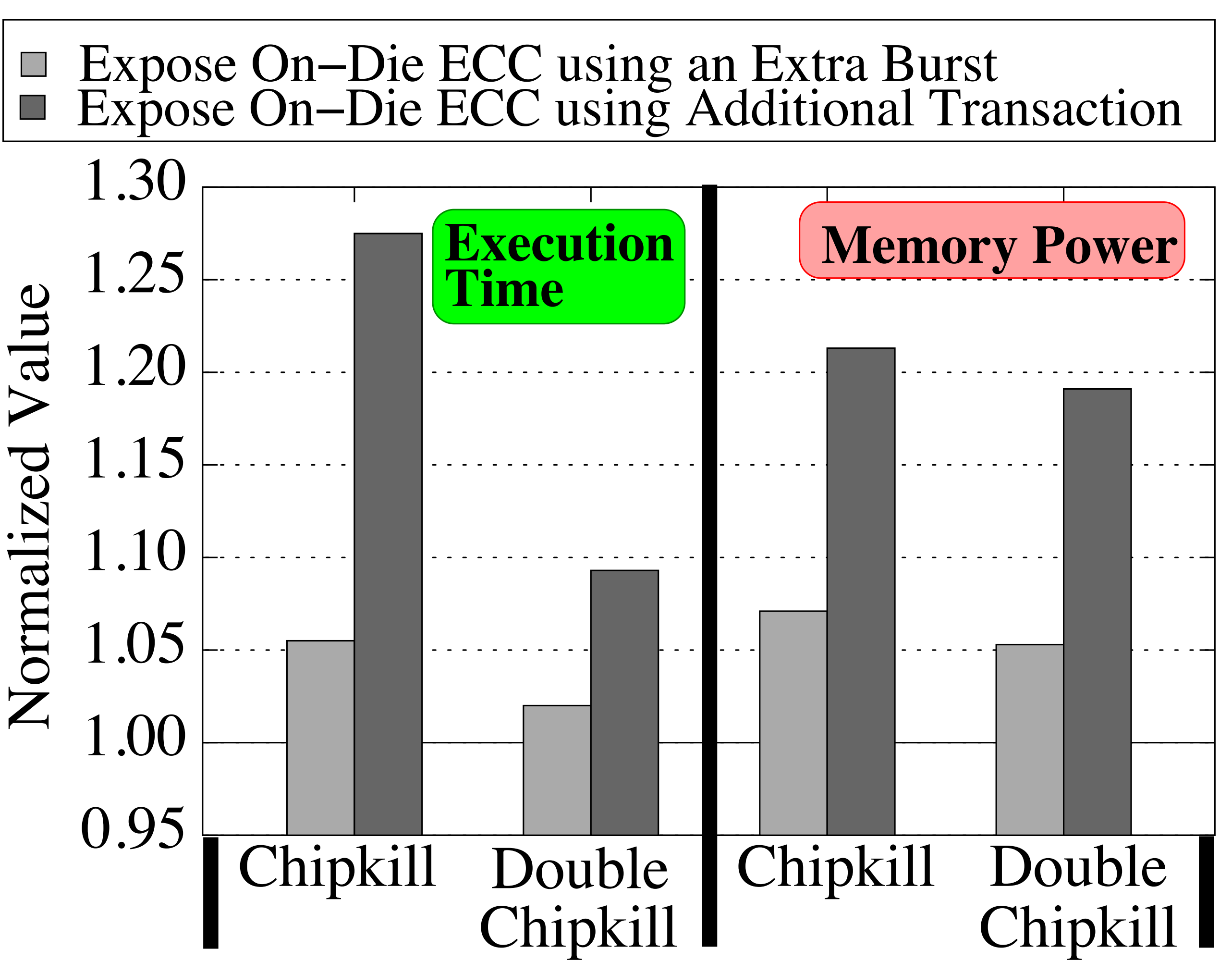
SDC AND DUE

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