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

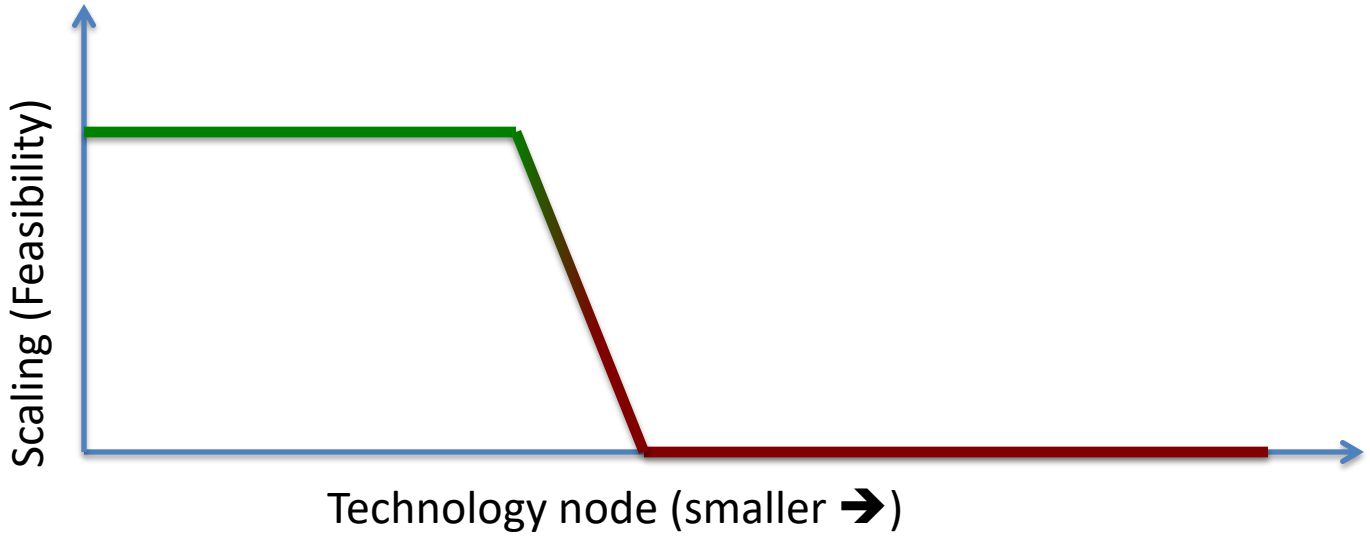
Prashant J. Nair

Dae-Hyun Kim

Moinuddin K. Qureshi

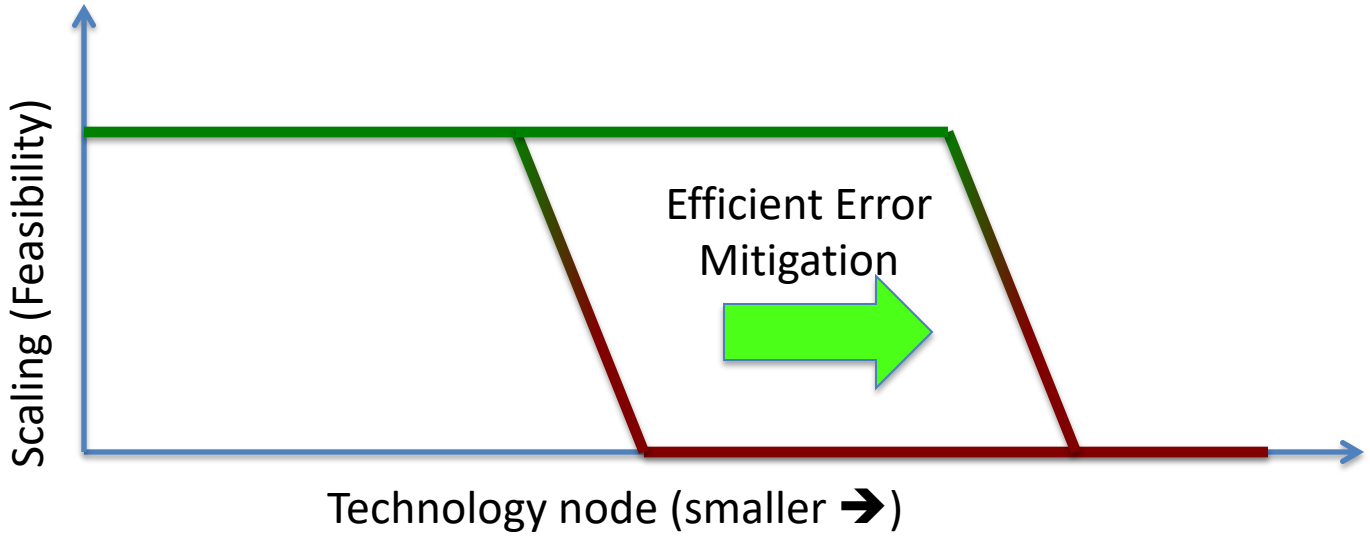
Introduction

- DRAM: Basic building block for main memory for four decades
- DRAM scaling provides higher memory capacity. Moving to smaller node provides ~2x capacity
- Shrinking DRAM cells becoming difficult ➔ threat to scaling



Introduction

- DRAM: Basic building block for main memory for four decades
- DRAM scaling provides higher memory capacity. Moving to smaller node provides ~2x capacity
- Shrinking DRAM cells becoming difficult ➔ threat to scaling



Efficient error handling can help DRAM technology scale

Why DRAM Scaling is Difficult?

Why DRAM Scaling is Difficult?

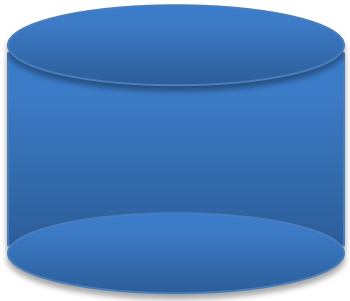
- Scaling is difficult.

Why DRAM Scaling is Difficult?

- Scaling is difficult. More so for DRAM cells.

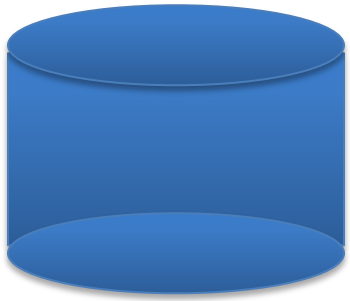
Why DRAM Scaling is Difficult?

- Scaling is difficult. More so for DRAM cells.
- Volume of capacitance must remain constant (25fF)



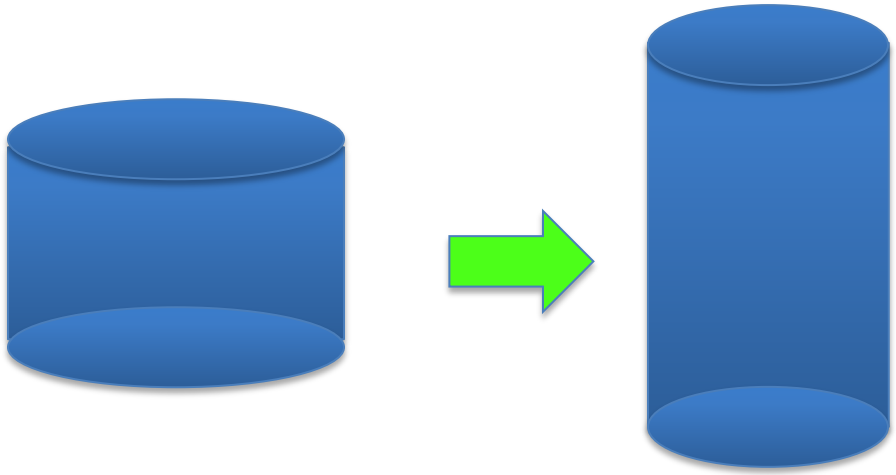
Why DRAM Scaling is Difficult?

- Scaling is difficult. More so for DRAM cells.
- Volume of capacitance must remain constant (25fF)
- Scaling: 0.7x dimension, 0.5x area ➔ 2x height



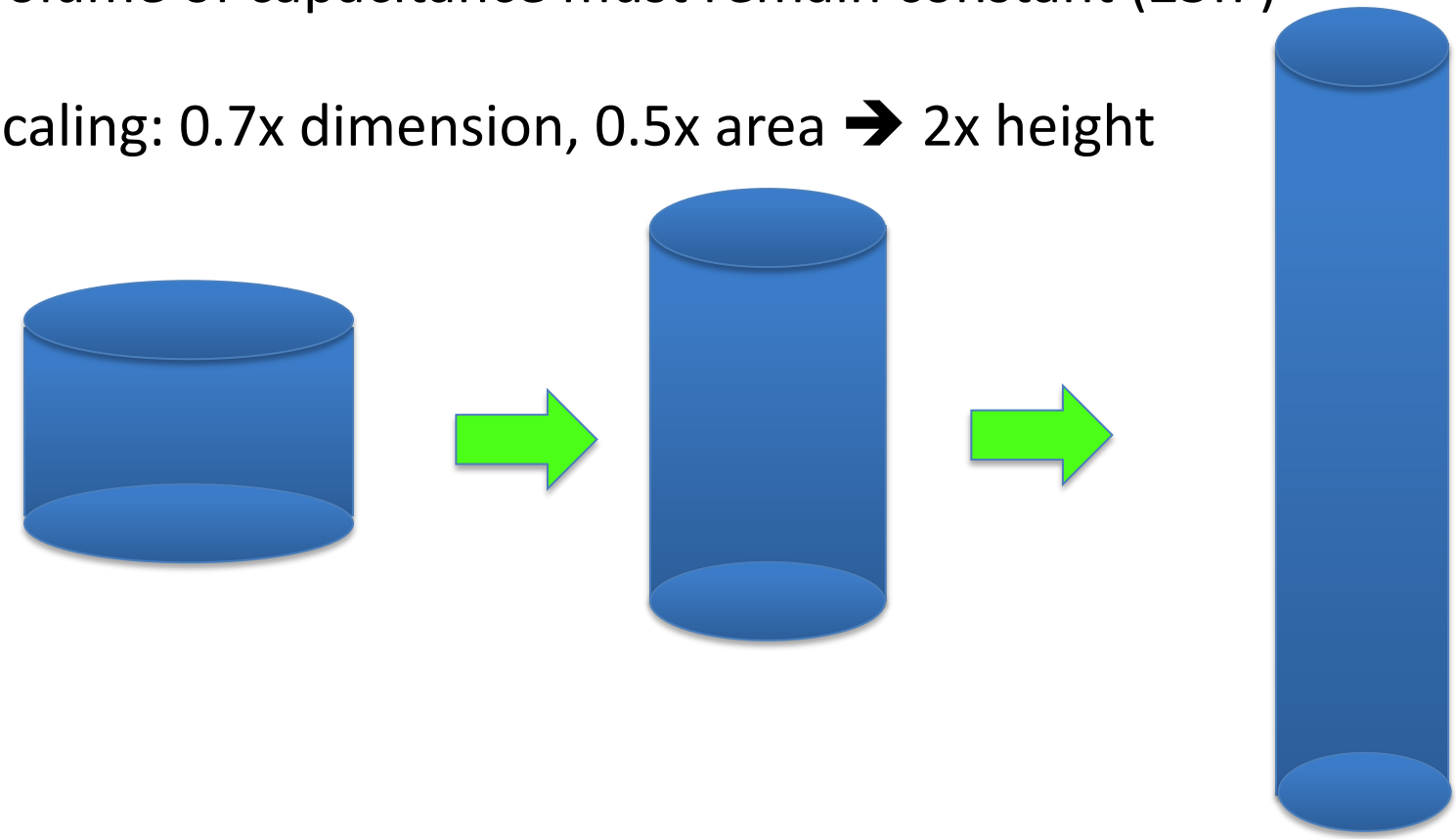
Why DRAM Scaling is Difficult?

- Scaling is difficult. More so for DRAM cells.
- Volume of capacitance must remain constant (25fF)
- Scaling: 0.7x dimension, 0.5x area ➔ 2x height



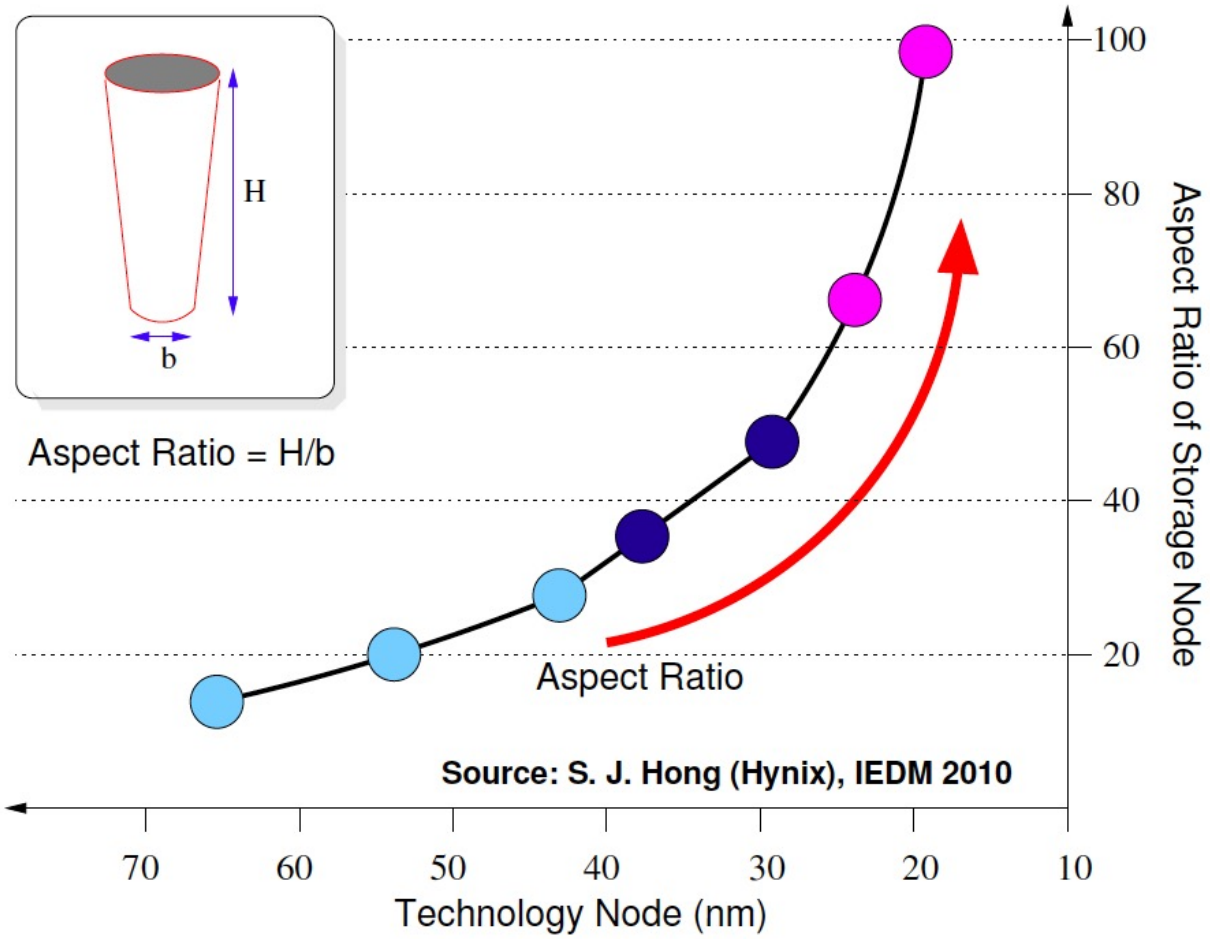
Why DRAM Scaling is Difficult?

- Scaling is difficult. More so for DRAM cells.
- Volume of capacitance must remain constant (25fF)
- Scaling: 0.7x dimension, 0.5x area ➔ 2x height



With scaling, DRAM cells not only become narrower but longer

DRAM: Aspect Ratio Trend



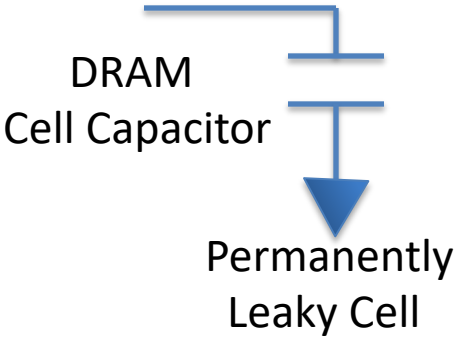
Narrow cylindrical cells are mechanically unstable ➔ breaks

More Reasons for DRAM Faults

- Unreliability of ultra-thin dielectric material
- In addition, DRAM cell failures also from:
  - Permanently leaky cells
  - Mechanically unstable cells
  - Broken links in the DRAM array

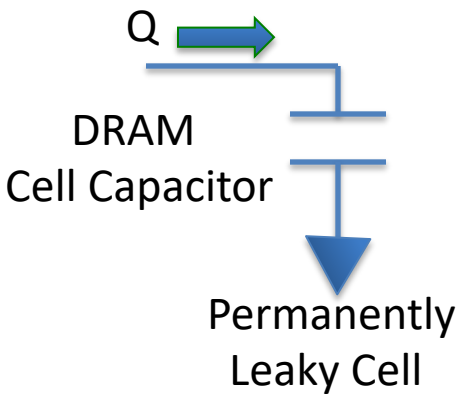
More Reasons for DRAM Faults

- Unreliability of ultra-thin dielectric material
- In addition, DRAM cell failures also from:
  - Permanently leaky cells
  - Mechanically unstable cells
  - Broken links in the DRAM array



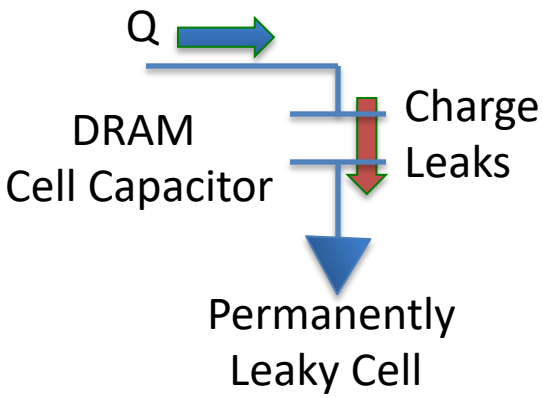
More Reasons for DRAM Faults

- Unreliability of ultra-thin dielectric material
- In addition, DRAM cell failures also from:
  - Permanently leaky cells
  - Mechanically unstable cells
  - Broken links in the DRAM array



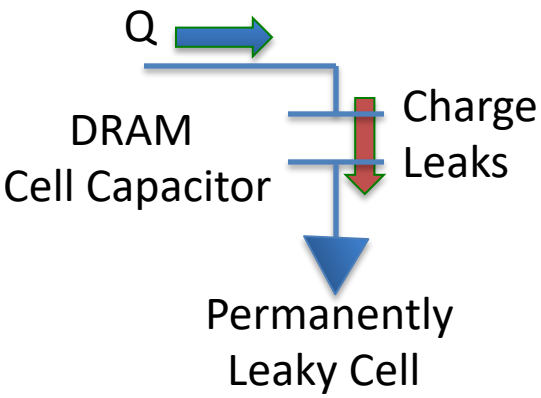
More Reasons for DRAM Faults

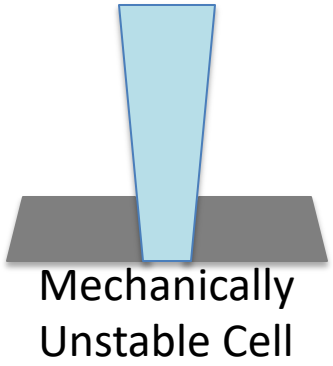
- Unreliability of ultra-thin dielectric material
- In addition, DRAM cell failures also from:
  - Permanently leaky cells
  - Mechanically unstable cells
  - Broken links in the DRAM array



More Reasons for DRAM Faults

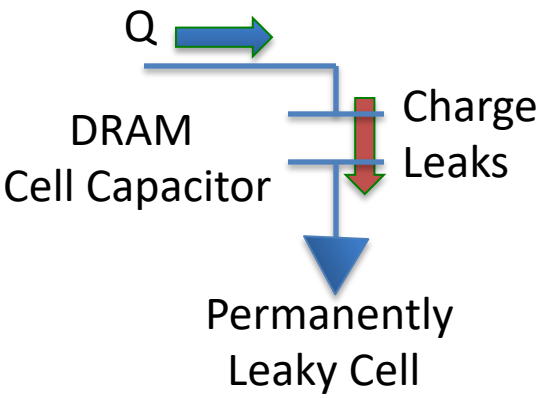
- Unreliability of ultra-thin dielectric material
- In addition, DRAM cell failures also from:
  - Permanently leaky cells
  - Mechanically unstable cells
  - Broken links in the DRAM array

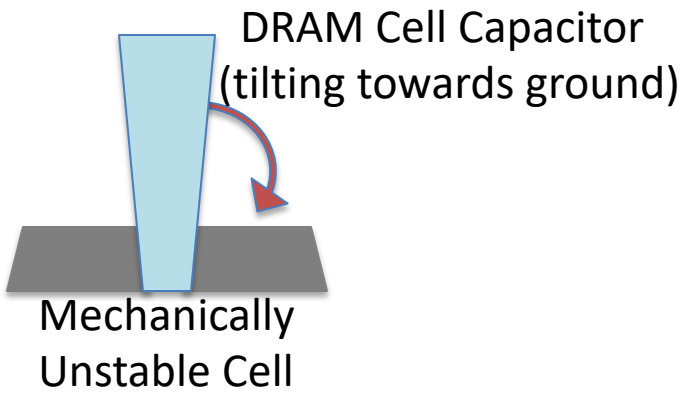




More Reasons for DRAM Faults

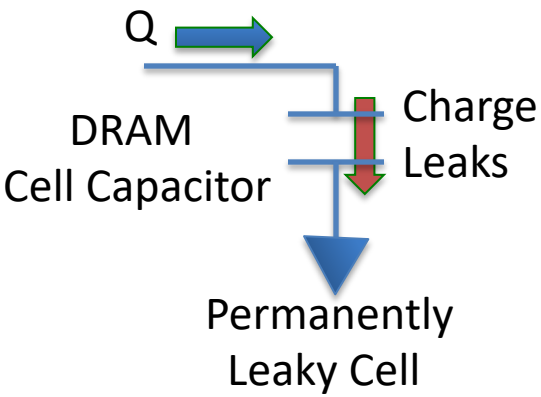
- Unreliability of ultra-thin dielectric material
- In addition, DRAM cell failures also from:
  - Permanently leaky cells
  - Mechanically unstable cells
  - Broken links in the DRAM array

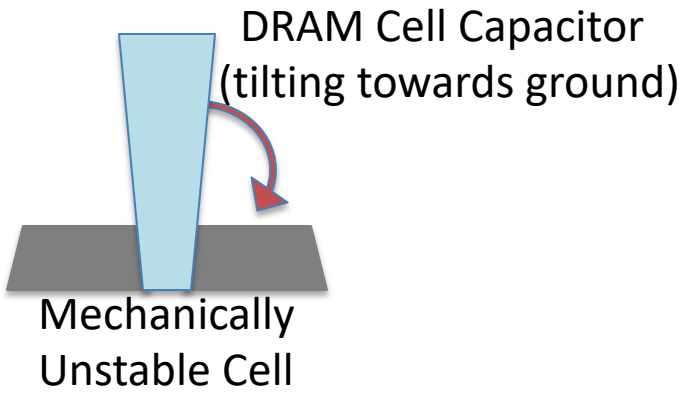


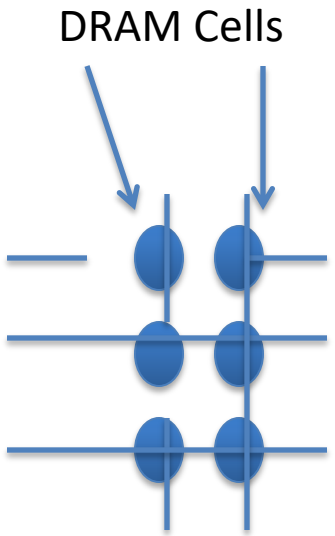


More Reasons for DRAM Faults

- Unreliability of ultra-thin dielectric material
- In addition, DRAM cell failures also from:
  - Permanently leaky cells
  - Mechanically unstable cells
  - Broken links in the DRAM array

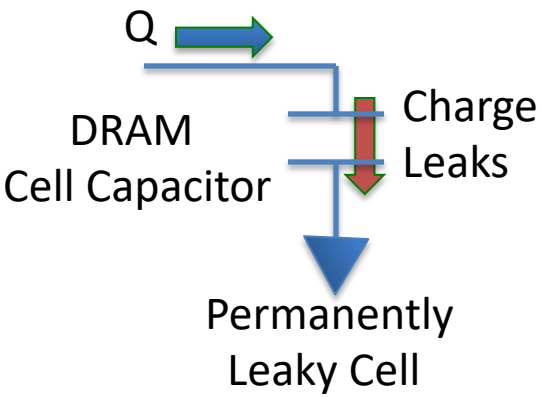


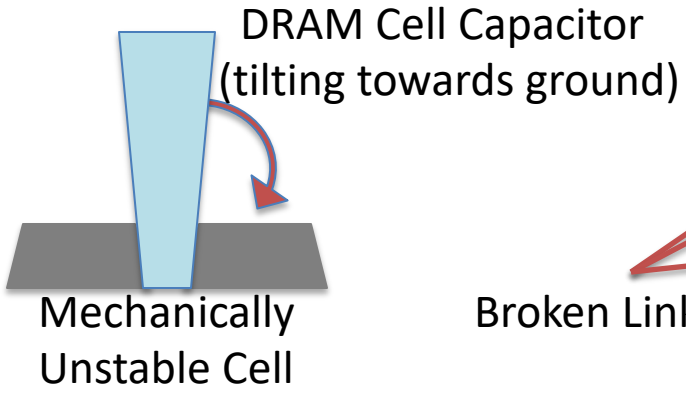


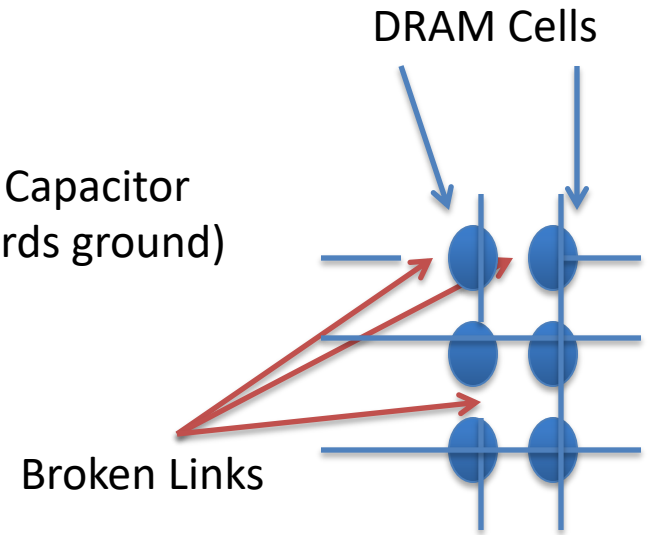


More Reasons for DRAM Faults

- Unreliability of ultra-thin dielectric material
- In addition, DRAM cell failures also from:
  - Permanently leaky cells
  - Mechanically unstable cells
  - Broken links in the DRAM array

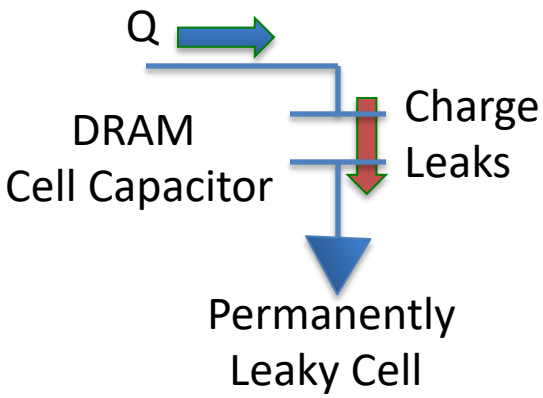


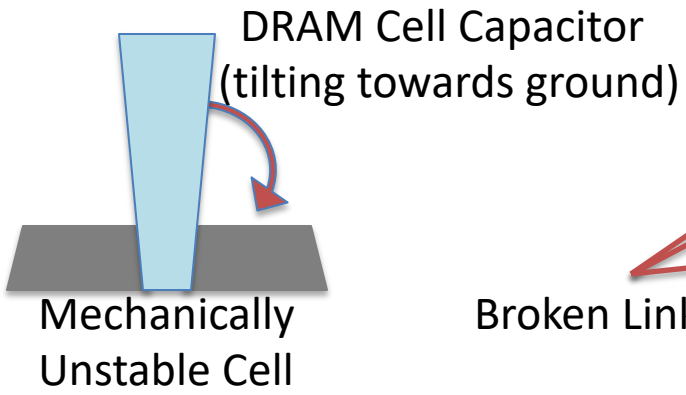


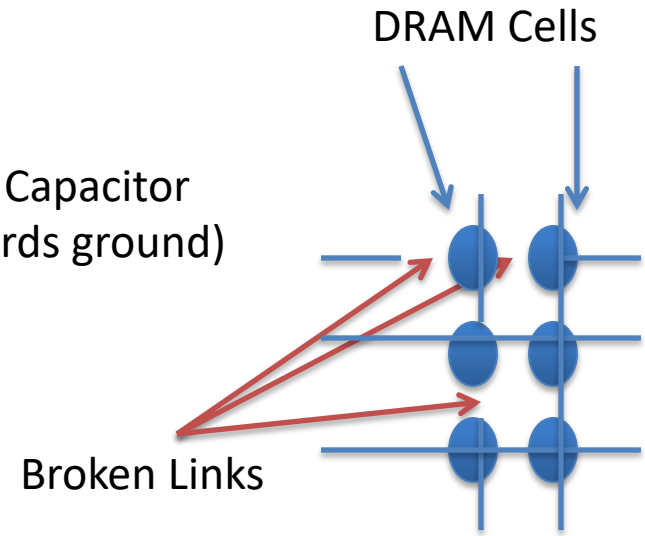


More Reasons for DRAM Faults

- Unreliability of ultra-thin dielectric material
- In addition, DRAM cell failures also from:
  - Permanently leaky cells
  - Mechanically unstable cells
  - Broken links in the DRAM array







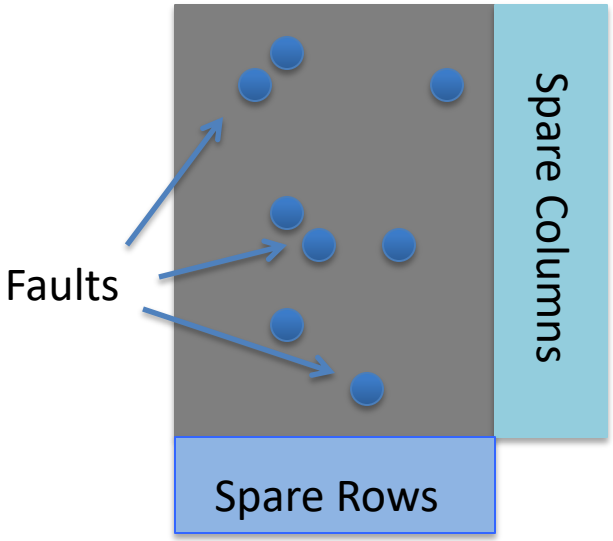
Permanent faults for future DRAMs expected to be much higher (we target an error rate as high as 100ppm)

Outline

- Introduction
- Current Schemes
- ArchShield
- Evaluation
- Summary

Row and Column Sparing

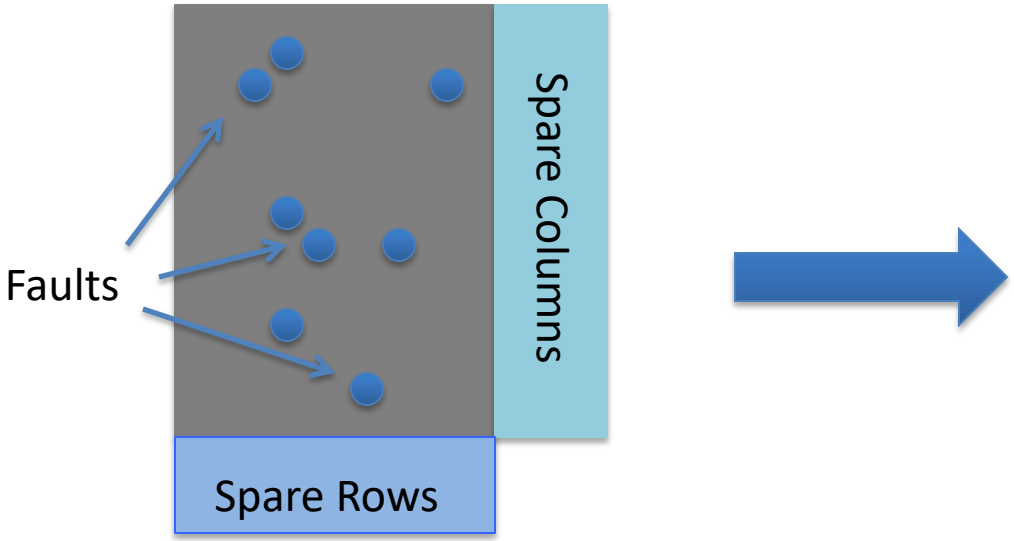
- DRAM chip (organized into rows and columns) have spares



DRAM Chip:
Before Row/Column Sparing

Row and Column Sparing

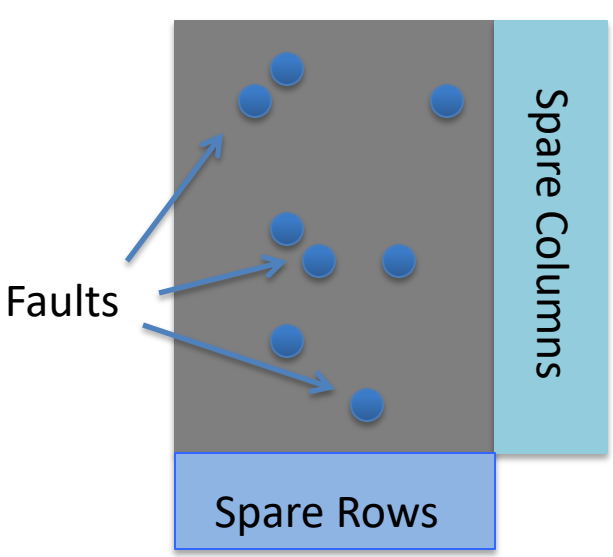
- DRAM chip (organized into rows and columns) have spares



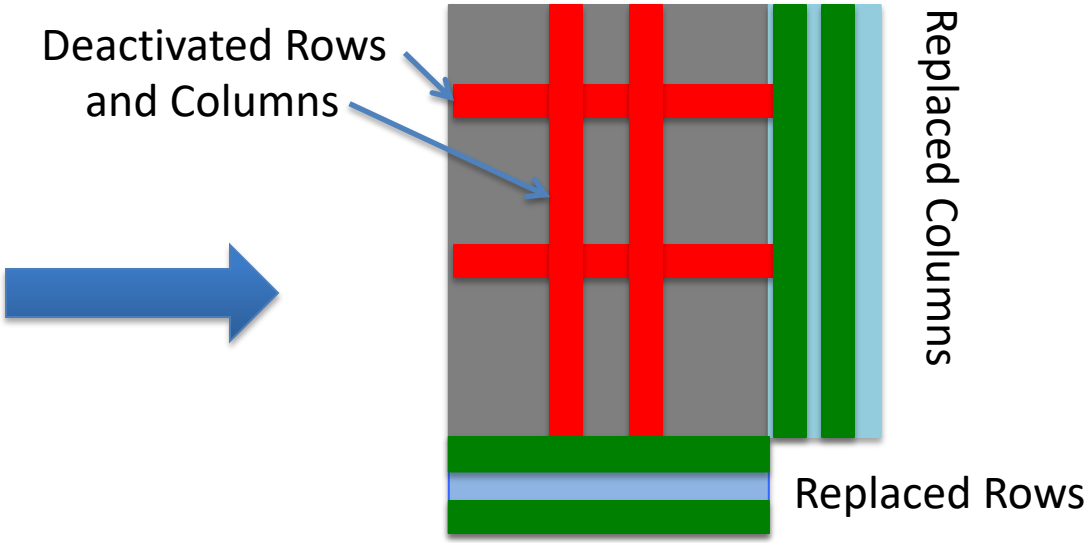
DRAM Chip:
Before Row/Column Sparing

Row and Column Sparing

- DRAM chip (organized into rows and columns) have spares



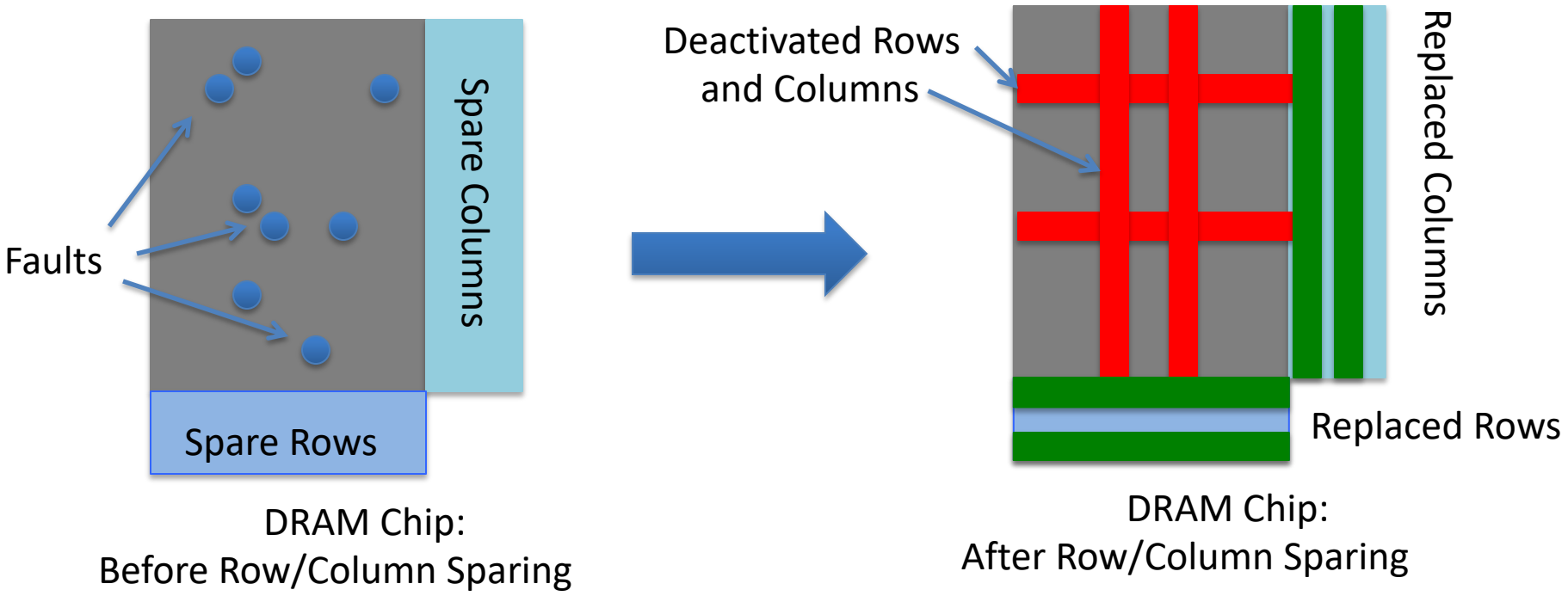
DRAM Chip:
Before Row/Column Sparing



DRAM Chip:
After Row/Column Sparing

Row and Column Sparing

- DRAM chip (organized into rows and columns) have spares



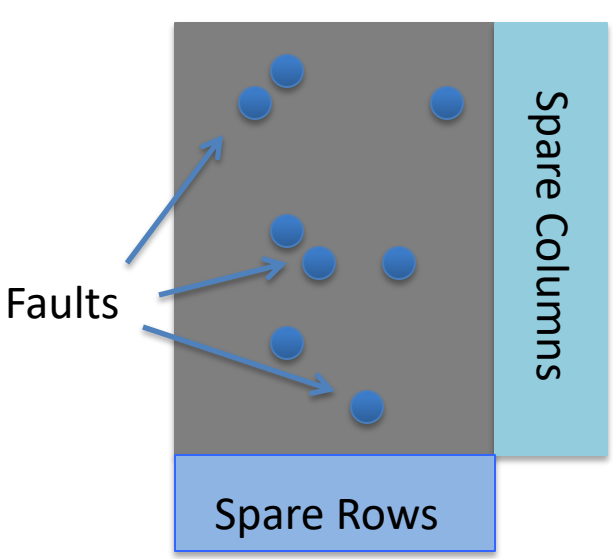
DRAM Chip:
Before Row/Column Sparing

DRAM Chip:
After Row/Column Sparing

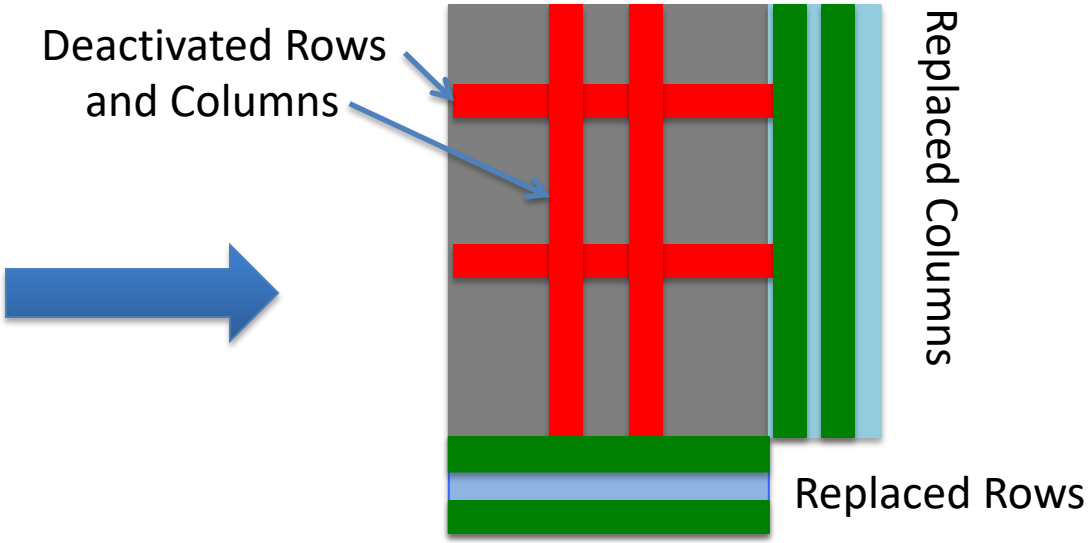
- Laser fuses enable spare rows/columns
- Entire row/column needs to be sacrificed for a few faulty cells

Row and Column Sparing

- DRAM chip (organized into rows and columns) have spares



DRAM Chip:
Before Row/Column Sparing



DRAM Chip:
After Row/Column Sparing

- Laser fuses enable spare rows/columns
- Entire row/column needs to be sacrificed for a few faulty cells

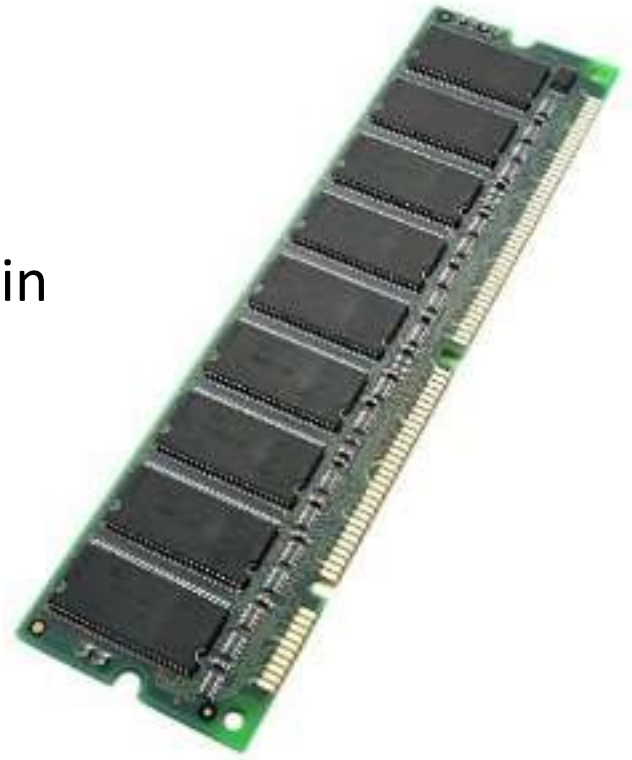
Row and Column Sparing Schemes have large area overheads

Commodity ECC-DIMM

- Commodity ECC DIMM with SECDED at 8 bytes (72,64)
- Mainly used for soft-error protection

Commodity ECC-DIMM

- Commodity ECC DIMM with SECDED at 8 bytes (72,64)
- Mainly used for soft-error protection
- For hard errors, high chance of two errors in same word (birthday paradox)



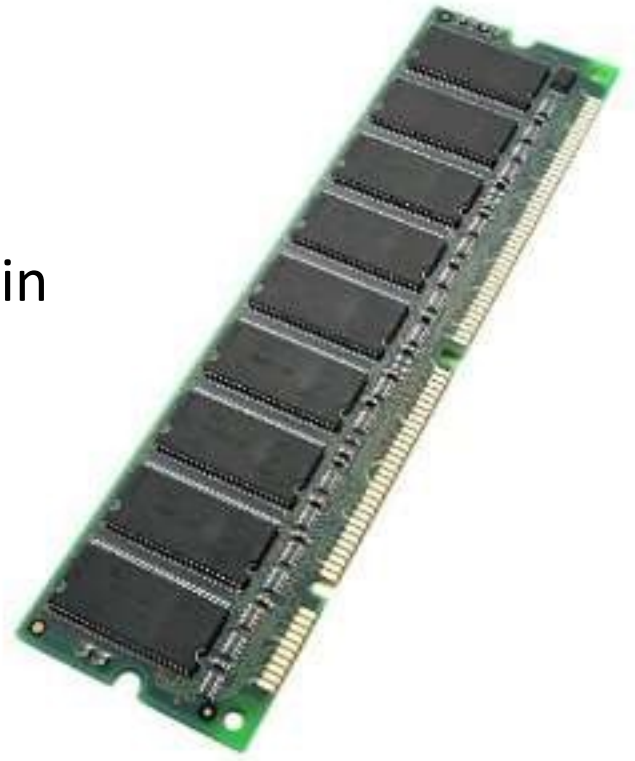
For 8GB DIMM ➔ 1 billion words

Expected errors till double-error word

= 1.25*Sqrt(N) = 40K errors ➔ 0.5 ppm

Commodity ECC-DIMM

- Commodity ECC DIMM with SECDED at 8 bytes (72,64)
- Mainly used for soft-error protection
- For hard errors, high chance of two errors in same word (birthday paradox)



For 8GB DIMM ➔ 1 billion words

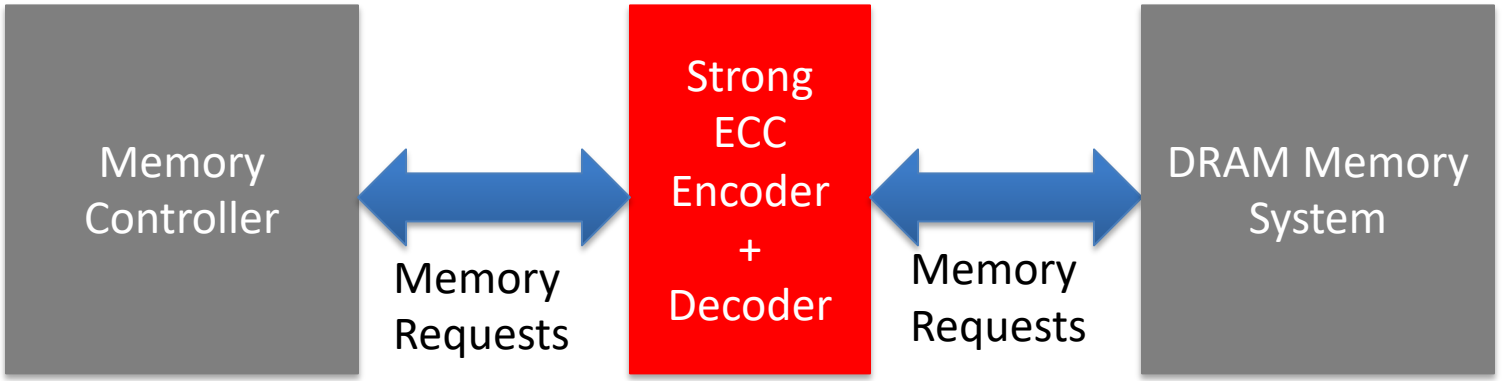
Expected errors till double-error word

= 1.25*Sqrt(N) = 40K errors ➔ 0.5 ppm

SECDED not enough for high error-rate (+ lost soft-error protection)

Strong ECC Codes

- Strong ECC (BCH) codes are robust, but complex and costly



- Each memory reference incurs encoding/decoding latency

- For BER of 100 ppm, we need ECC-4 ➔ 50% storage overhead

Strong ECC codes provide an inefficient solution for tolerating errors

Dissecting Fault Probabilities

At Bit Error Rate of 10^{-4} (100ppm) for an 8GB DIMM (1 billion words)

Faulty Bits per word (8B)	Probability	Num words in 8GB
0	99.3%	0.99 Billion
1	0.007	7.7 Million
2	26×10^{-6}	28 K
3	62×10^{-9}	67
4	10^{-10}	0.1

Dissecting Fault Probabilities

At Bit Error Rate of 10^{-4} (100ppm) for an 8GB DIMM (1 billion words)

Faulty Bits per word (8B)	Probability	Num words in 8GB
0	99.3%	0.99 Billion
1	0.007	7.7 Million
2	26×10^{-6}	28 K
3	62×10^{-9}	67
4	10^{-10}	0.1

Most faulty words have 1-bit error ➔ The skew in fault probability can be leveraged to develop low cost error resilience

Dissecting Fault Probabilities

At Bit Error Rate of 10^{-4} (100ppm) for an 8GB DIMM (1 billion words)

Faulty Bits per word (8B)	Probability	Num words in 8GB
0	99.3%	0.99 Billion
1	0.007	7.7 Million
2	26×10^{-6}	28 K
3	62×10^{-9}	67
4	10^{-10}	0.1

Most faulty words have 1-bit error ➔ The skew in fault probability can be leveraged to develop low cost error resilience

Goal: Tolerate high error rates with commodity ECC DIMM while retaining soft-error resilience

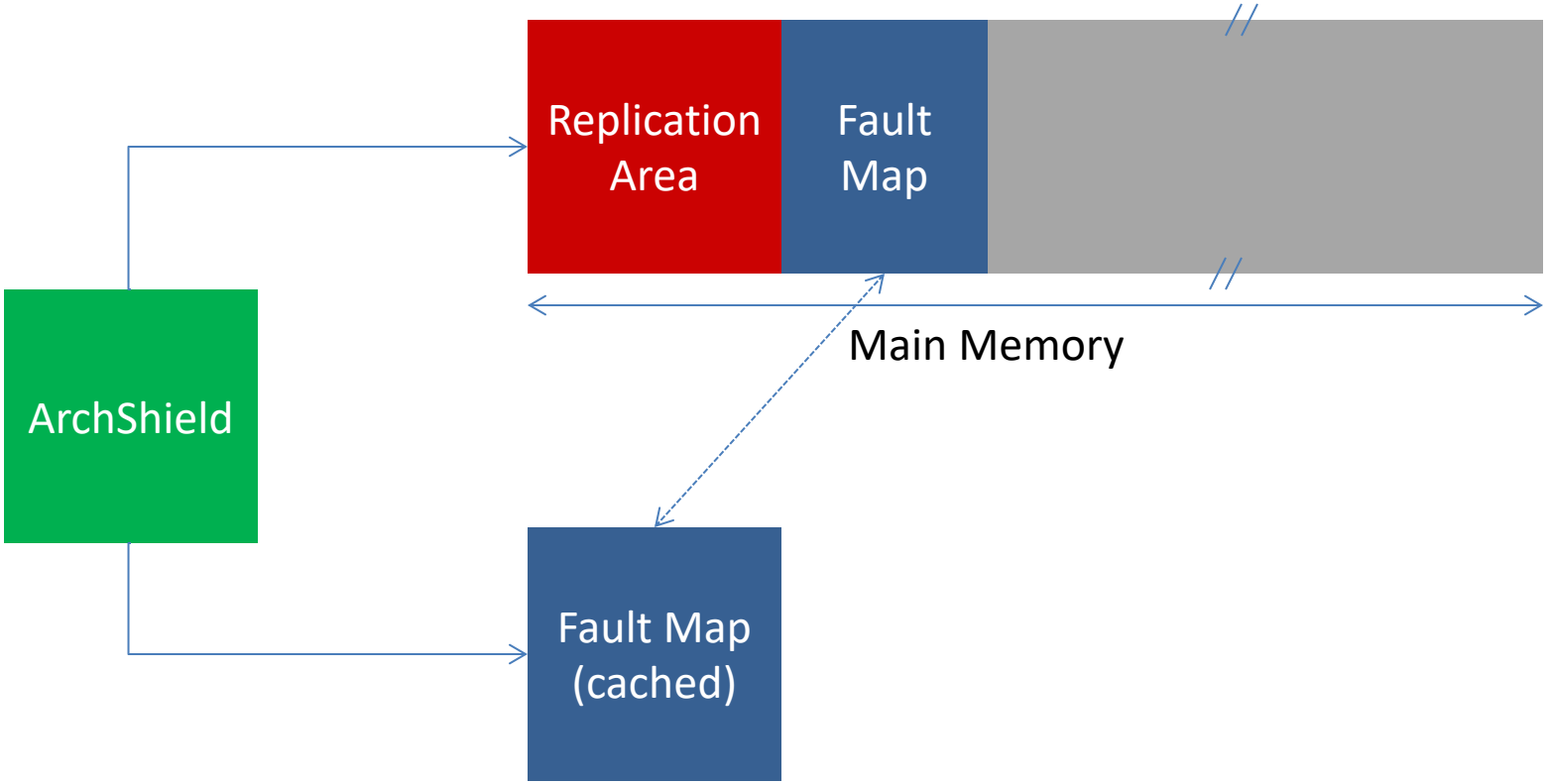
Outline

- Introduction
- Current Schemes
- ArchShield
- Evaluation
- Summary

ArchShield: Overview

Inspired from Solid State Drives (SSD) to tolerate high bit-error rate

Expose faulty cell information to Architecture layer via runtime testing

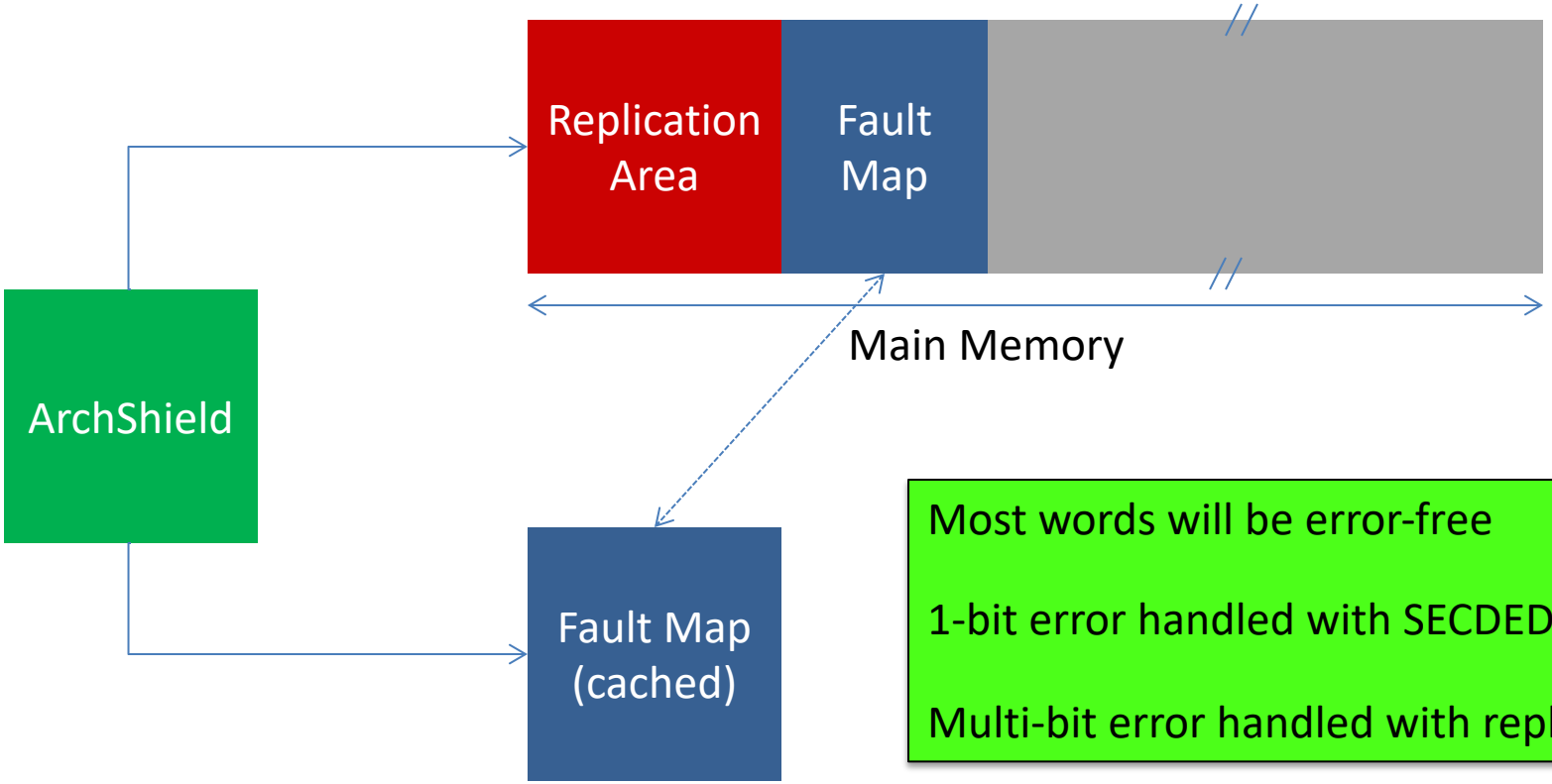


ArchShield stores the error mitigation information in memory

ArchShield: Overview

Inspired from Solid State Drives (SSD) to tolerate high bit-error rate

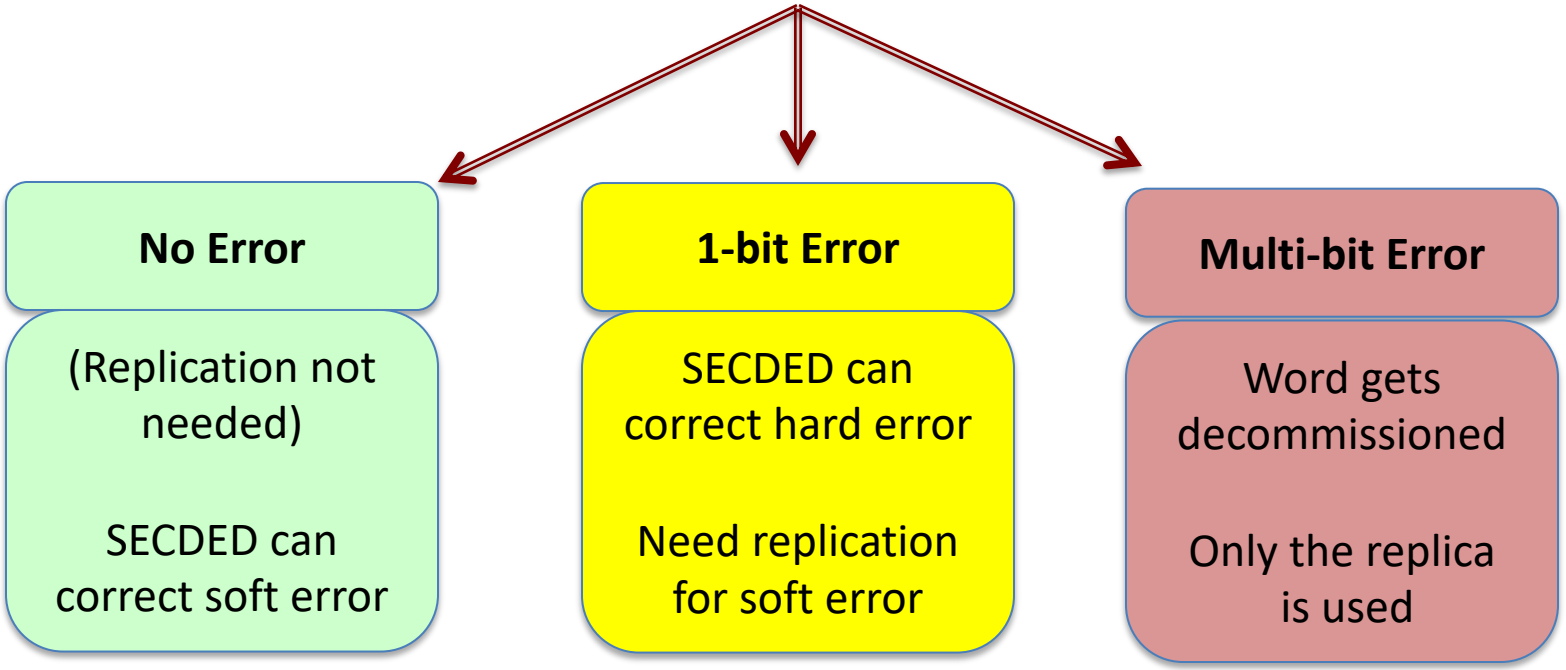
Expose faulty cell information to Architecture layer via runtime testing



ArchShield stores the error mitigation information in memory

ArchShield: Runtime Testing

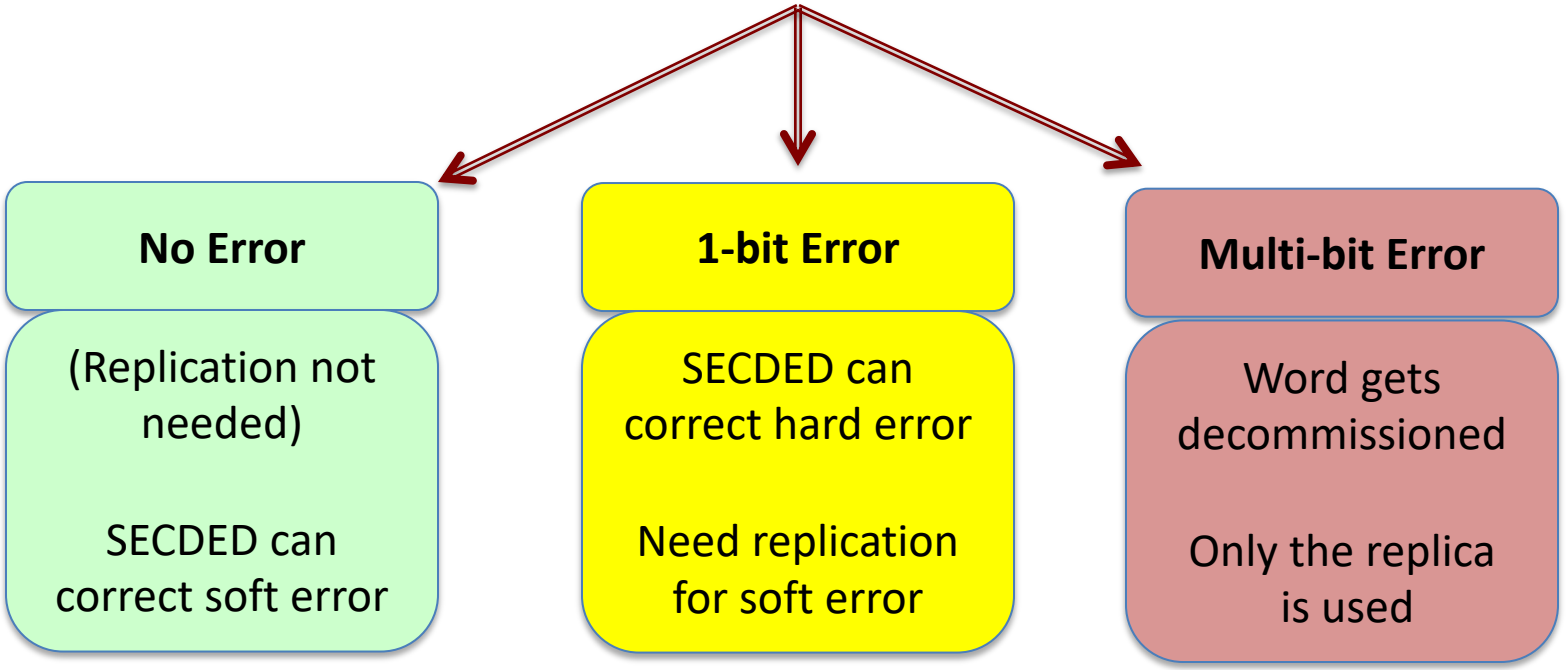
When DIMM is configured, runtime testing is performed. Each 8B word gets classified into one of three types:



(Information about faulty cells can be stored in hard drive for future use)

ArchShield: Runtime Testing

When DIMM is configured, runtime testing is performed. Each 8B word gets classified into one of three types:

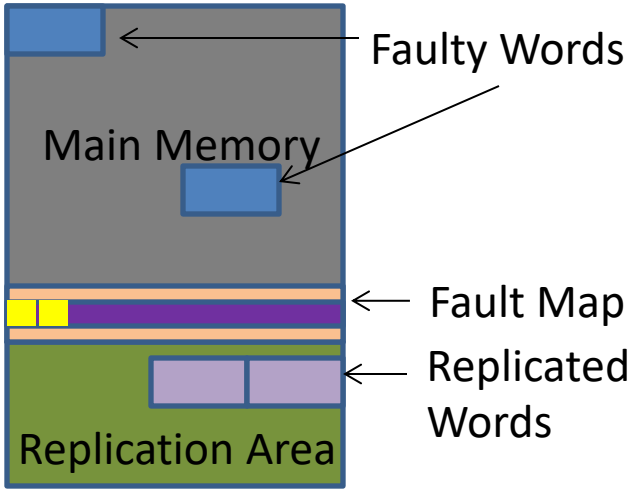


(Information about faulty cells can be stored in hard drive for future use)

Runtime testing identifies the faulty cells to decide correction

Architecting the Fault Map

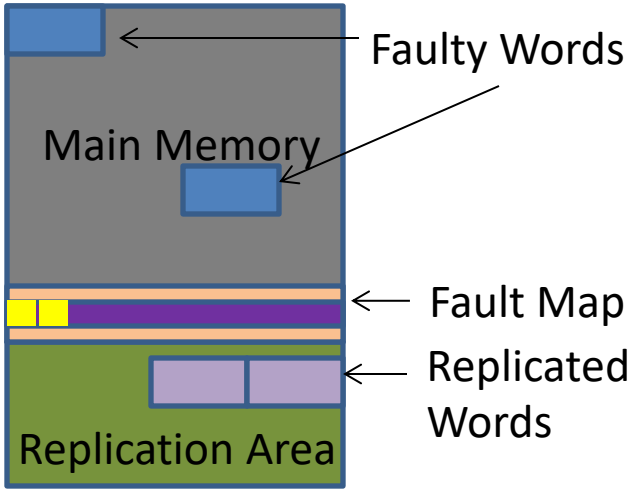
- Fault Map (FM) stores information about faulty cells
- Per word FM is expensive (for 8B, 2-bits or 4-bits with redundancy)
  ➔ Keep FM entry per line (4-bit per 64B)
- FM access method
  - Table lookup with Lineaddr



Fault Map – Organized at Line Granularity and is also cachable

Architecting the Fault Map

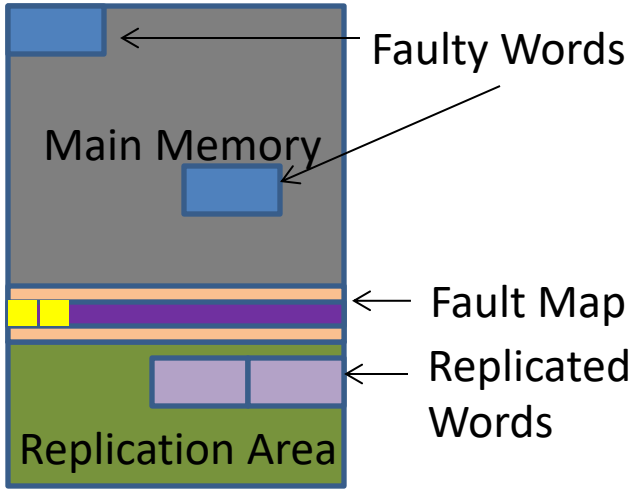
- Fault Map (FM) stores information about faulty cells
- Per word FM is expensive (for 8B, 2-bits or 4-bits with redundancy)
  ➔ Keep FM entry per line (4-bit per 64B)
- FM access method
  - Table lookup with Lineaddr
- Avoid dual memory access via
  - Caching FM entries in on-chip LLC
  - Each 64-byte line has 128 FM entries
  - Exploits spatial locality



Fault Map – Organized at Line Granularity and is also cachable

Architecting the Fault Map

- Fault Map (FM) stores information about faulty cells
- Per word FM is expensive (for 8B, 2-bits or 4-bits with redundancy)
  ➔ Keep FM entry per line (4-bit per 64B)
- FM access method
  - Table lookup with Lineaddr
- Avoid dual memory access via
  - Caching FM entries in on-chip LLC
  - Each 64-byte line has 128 FM entries
  - Exploits spatial locality



Fault Map – Organized at Line Granularity and is also cachable

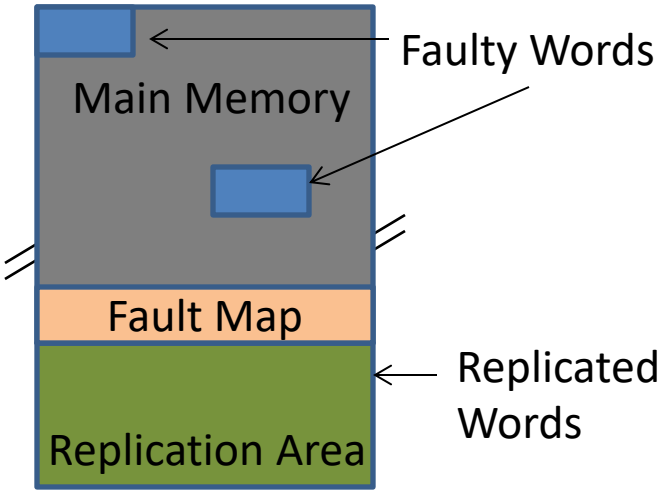
Line-Level Fault Map + Caching provides low storage and low latency

Architecting Replication Area

- Faulty cells replicated at word-granularity in Replication Area
- Fully associative Replication Area? Prohibitive latency
- Set associative Replication Area? Set overflow problem

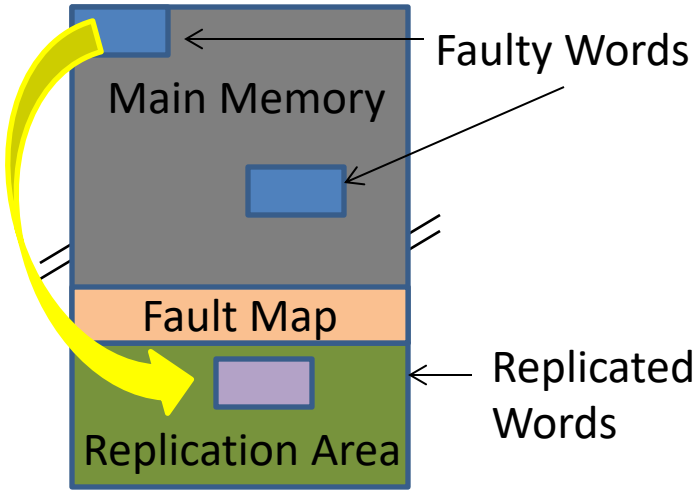
Architecting Replication Area

- Faulty cells replicated at word-granularity in Replication Area
- Fully associative Replication Area? Prohibitive latency
- Set associative Replication Area? Set overflow problem



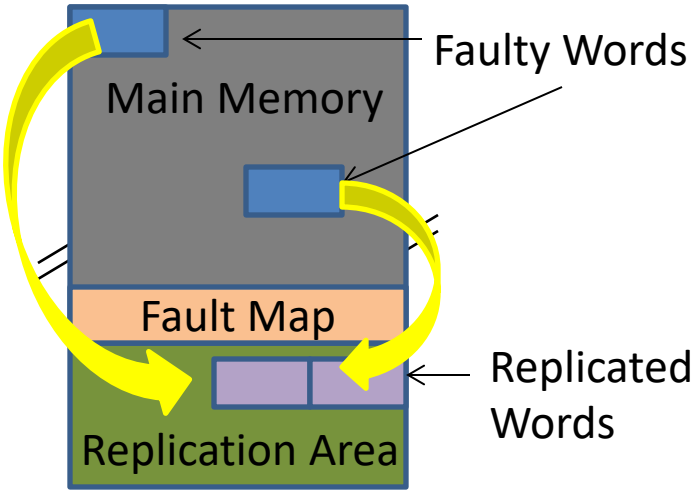
Architecting Replication Area

- Faulty cells replicated at word-granularity in Replication Area
- Fully associative Replication Area? Prohibitive latency
- Set associative Replication Area? Set overflow problem



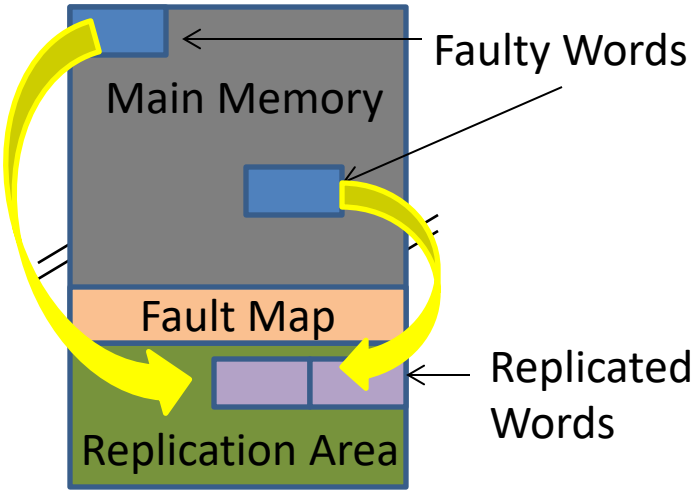
Architecting Replication Area

- Faulty cells replicated at word-granularity in Replication Area
- Fully associative Replication Area? Prohibitive latency
- Set associative Replication Area? Set overflow problem



Architecting Replication Area

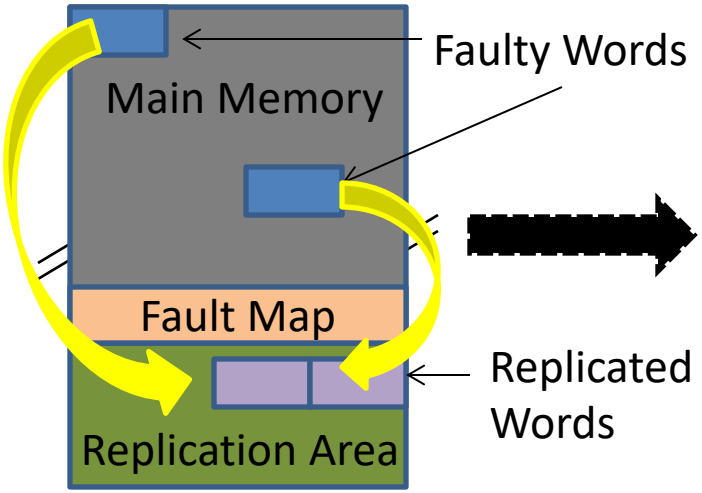
- Faulty cells replicated at word-granularity in Replication Area
- Fully associative Replication Area? Prohibitive latency
- Set associative Replication Area? Set overflow problem



Fully Associative Structure

Architecting Replication Area

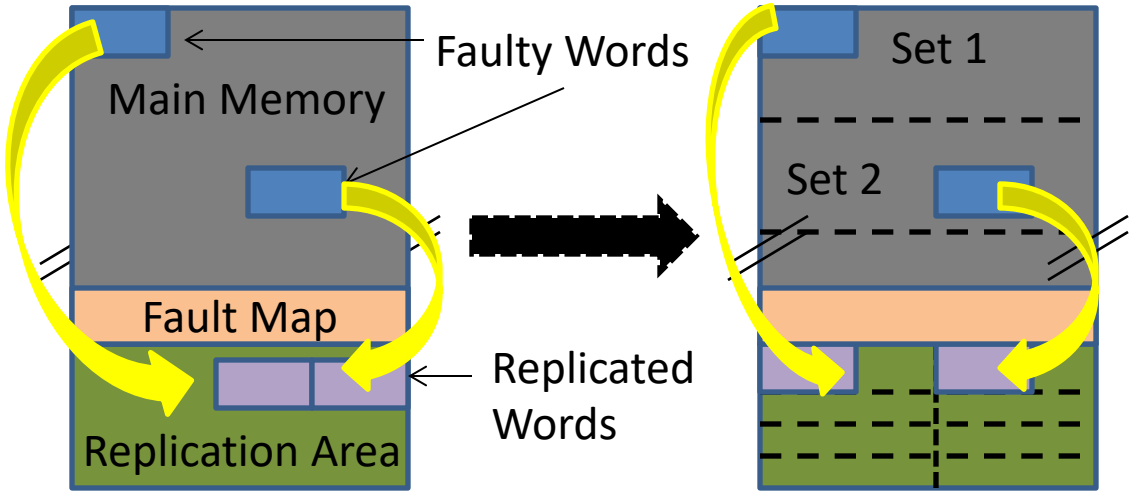
- Faulty cells replicated at word-granularity in Replication Area
- Fully associative Replication Area? Prohibitive latency
- Set associative Replication Area? Set overflow problem



Fully Associative Structure

Architecting Replication Area

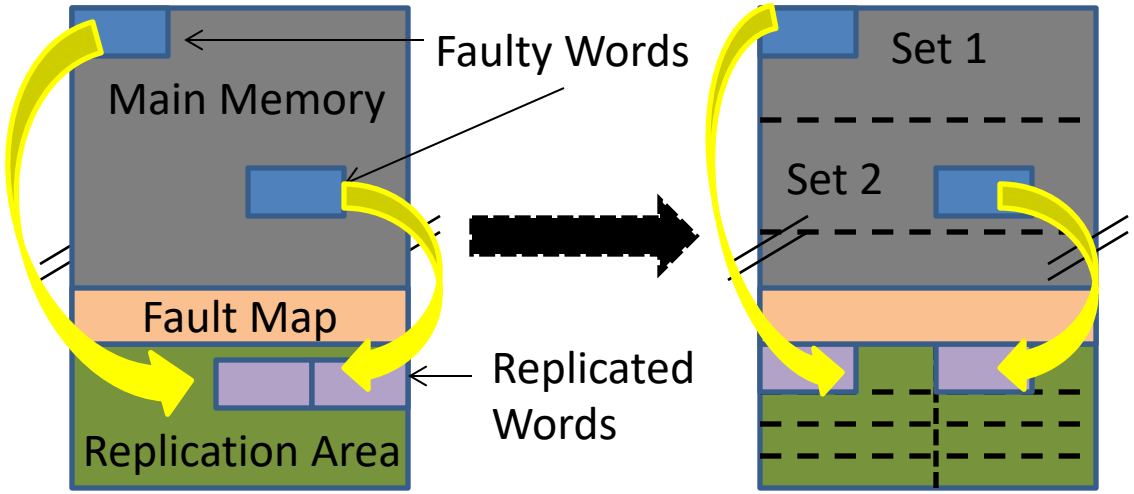
- Faulty cells replicated at word-granularity in Replication Area
- Fully associative Replication Area? Prohibitive latency
- Set associative Replication Area? Set overflow problem



Fully Associative Structure

Architecting Replication Area

- Faulty cells replicated at word-granularity in Replication Area
- Fully associative Replication Area? Prohibitive latency
- Set associative Replication Area? Set overflow problem

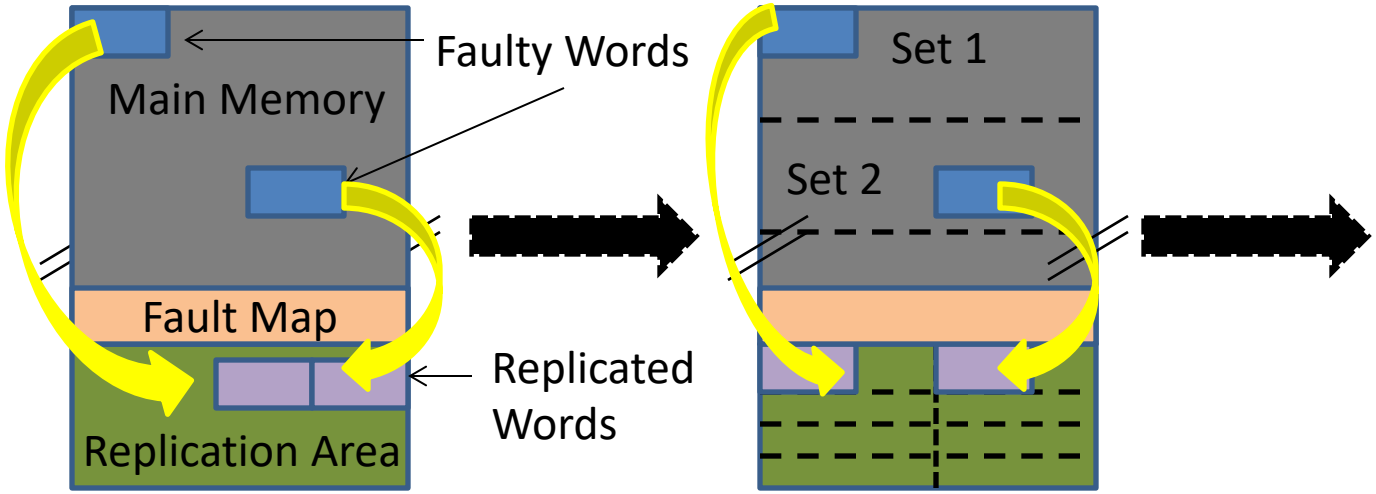


Fully Associative Structure

Set Associative Structure

Architecting Replication Area

- Faulty cells replicated at word-granularity in Replication Area
- Fully associative Replication Area? Prohibitive latency
- Set associative Replication Area? Set overflow problem

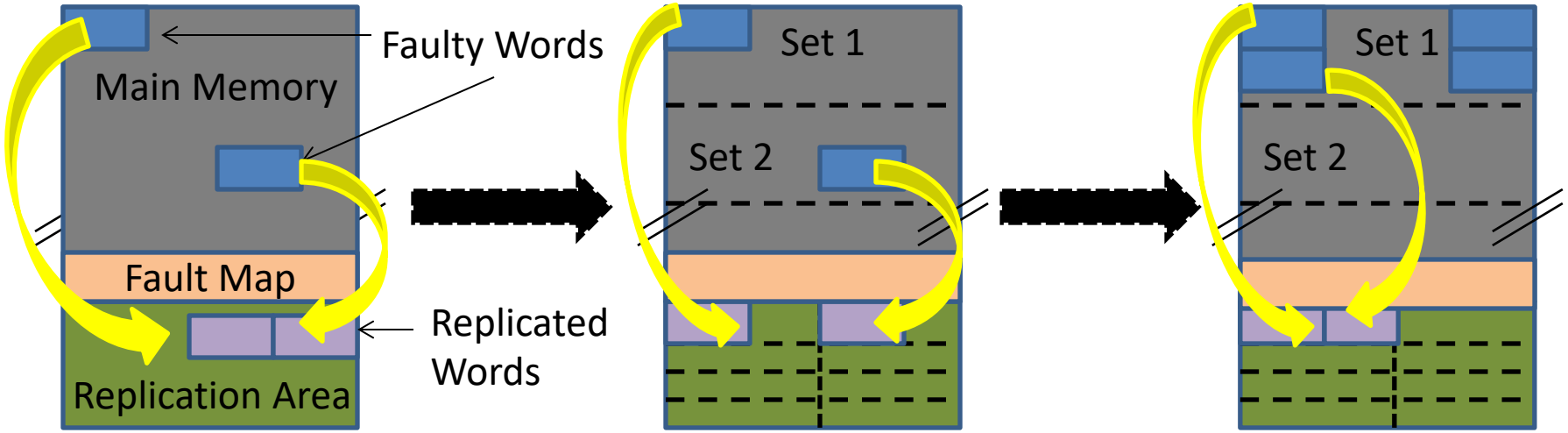


Fully Associative Structure

Set Associative Structure

Architecting Replication Area

- Faulty cells replicated at word-granularity in Replication Area
- Fully associative Replication Area? Prohibitive latency
- Set associative Replication Area? Set overflow problem

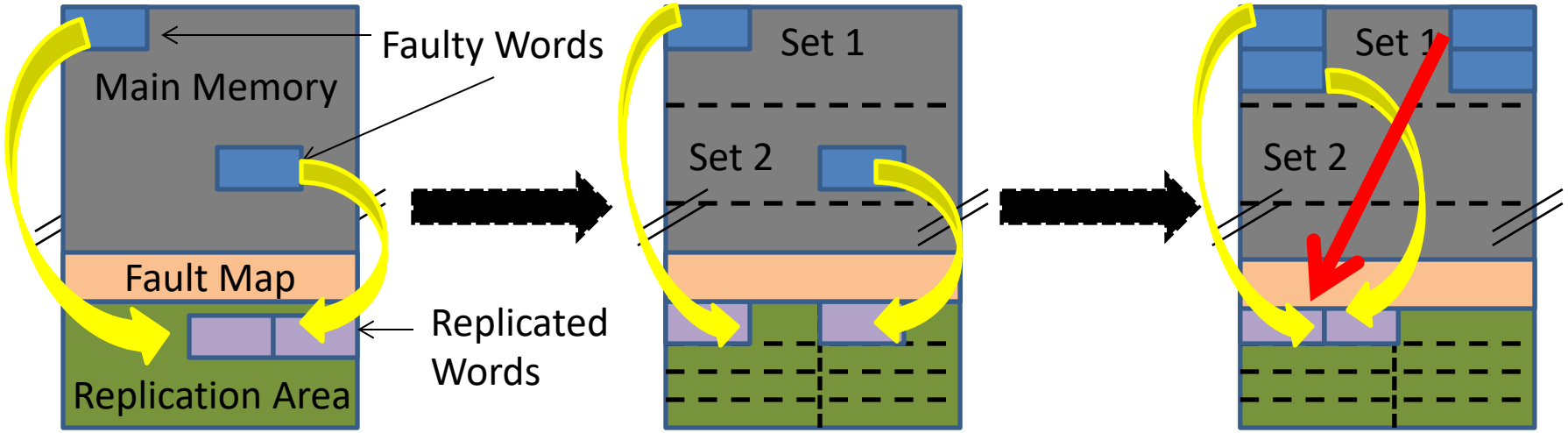


Fully Associative Structure

Set Associative Structure

Architecting Replication Area

- Faulty cells replicated at word-granularity in Replication Area
- Fully associative Replication Area? Prohibitive latency
- Set associative Replication Area? Set overflow problem

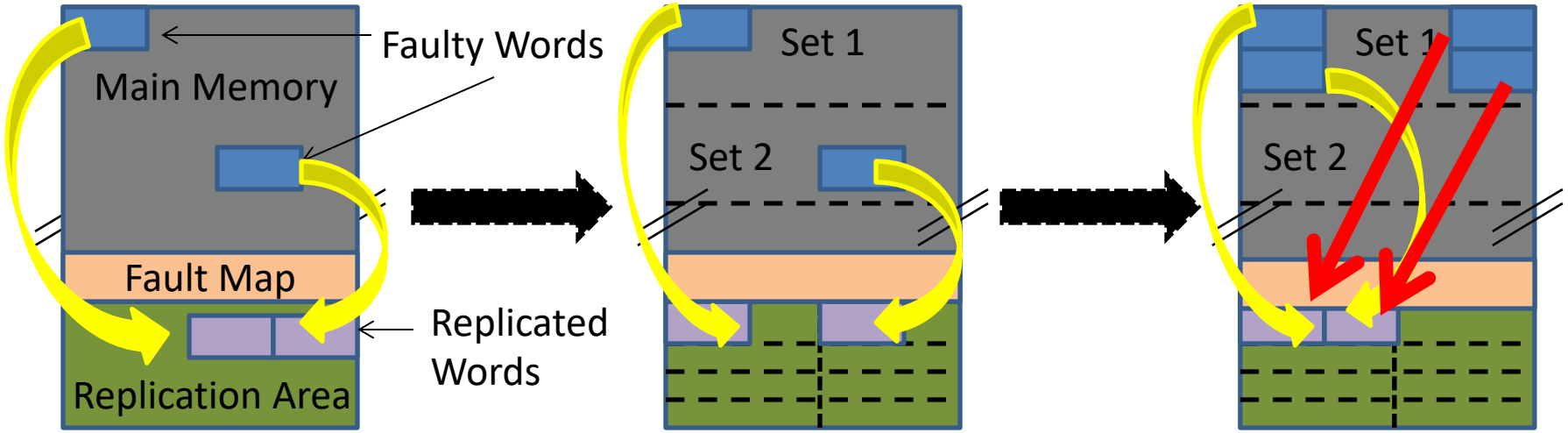


Fully Associative Structure

Set Associative Structure

Architecting Replication Area

- Faulty cells replicated at word-granularity in Replication Area
- Fully associative Replication Area? Prohibitive latency
- Set associative Replication Area? Set overflow problem

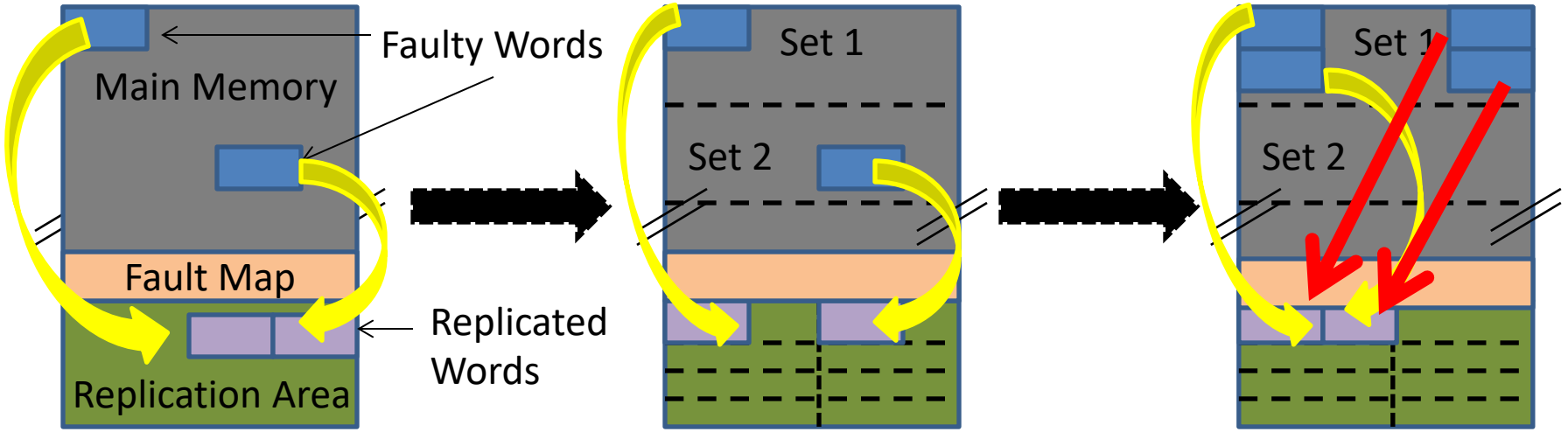


Fully Associative Structure

Set Associative Structure

Architecting Replication Area

- Faulty cells replicated at word-granularity in Replication Area
- Fully associative Replication Area? Prohibitive latency
- Set associative Replication Area? Set overflow problem



Fully Associative Structure

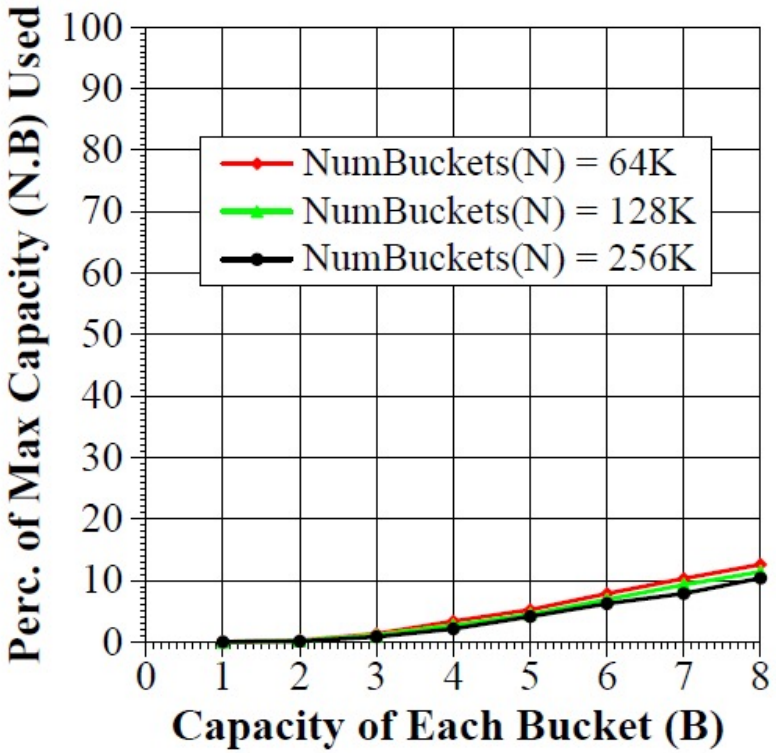
Set Associative Structure

Chances of Set Overflowing!

Overflow of Set-Associative RA

There are 10s/100s of thousand of sets ➔ Any set could overflow

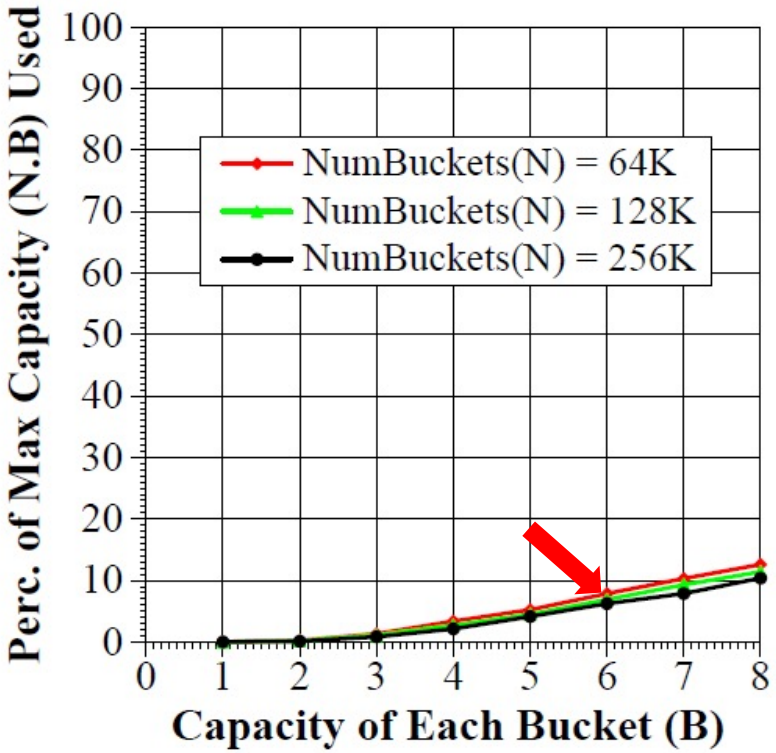
How many entries used before one set overflows? Buckets-and-Balls



Overflow of Set-Associative RA

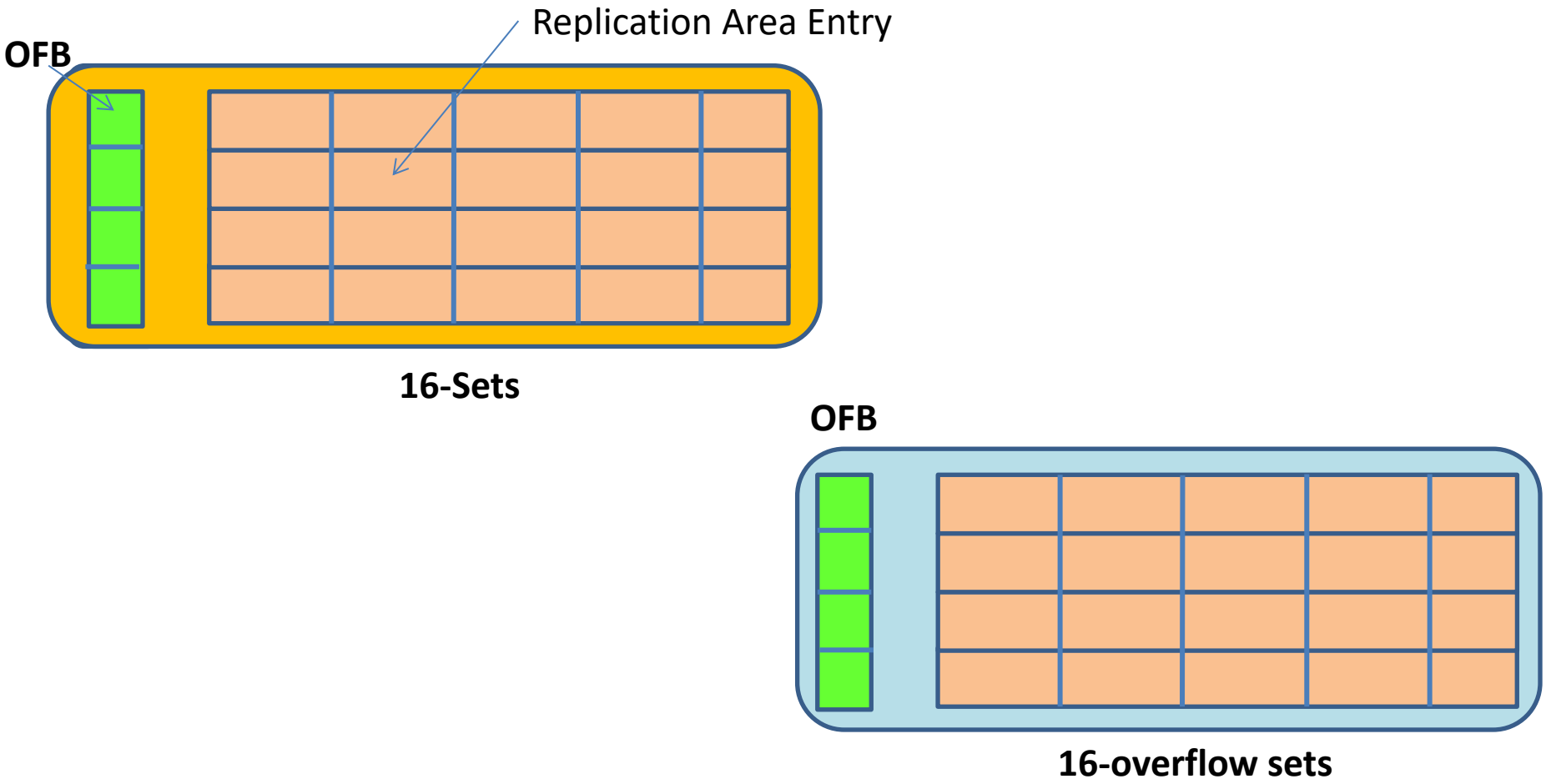
There are 10s/100s of thousand of sets ➔ Any set could overflow

How many entries used before one set overflows? Buckets-and-Balls

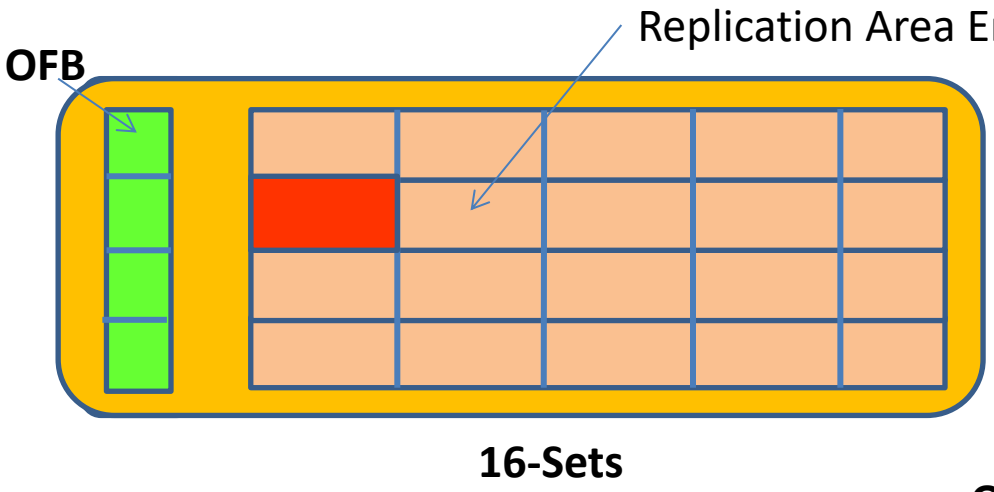


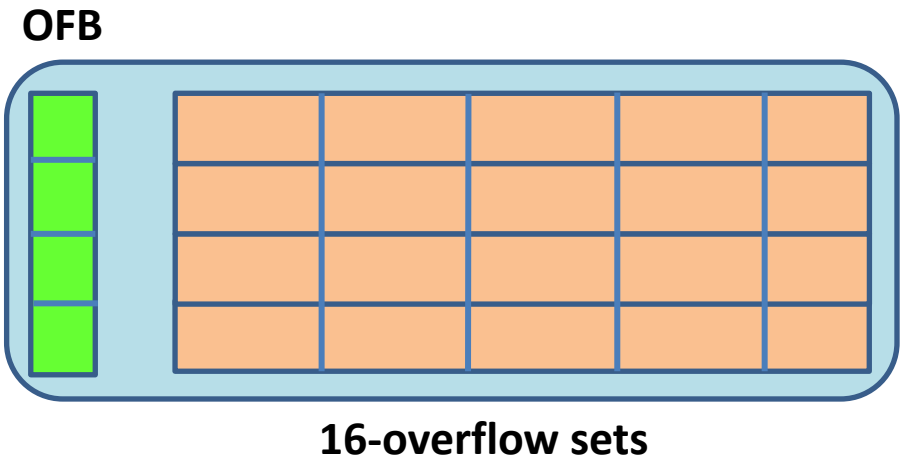
6-way table only 8% full when one set overflows ➔ Need 12x entries

Scalable Structure for RA

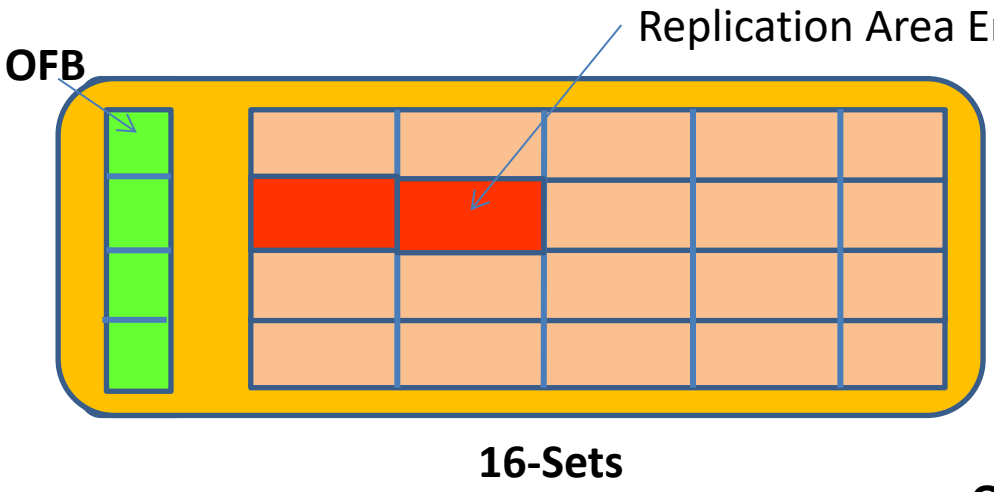


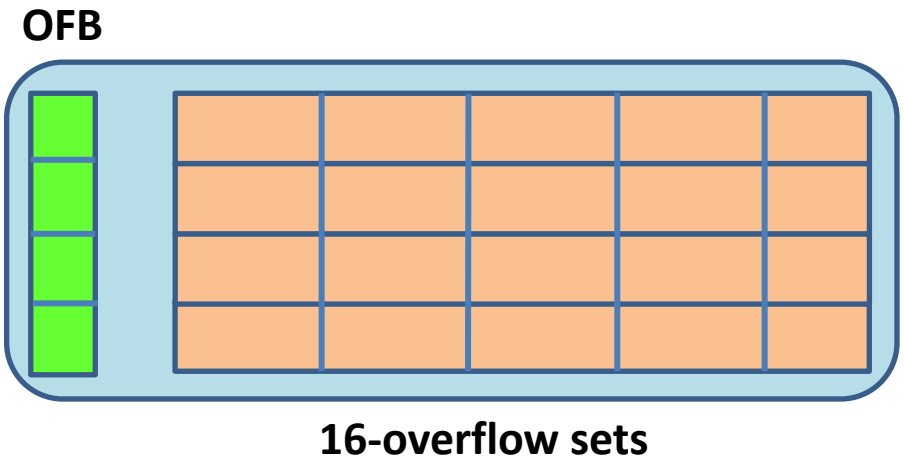
Scalable Structure for RA



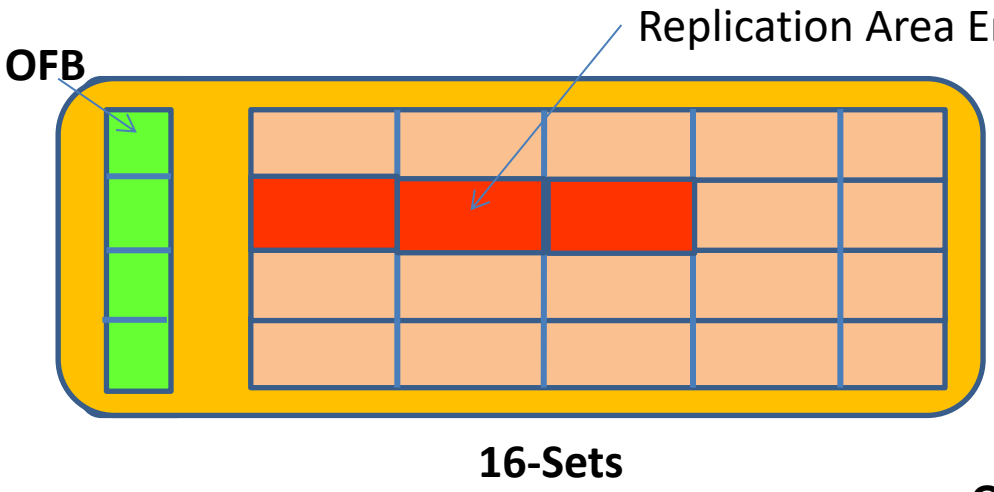


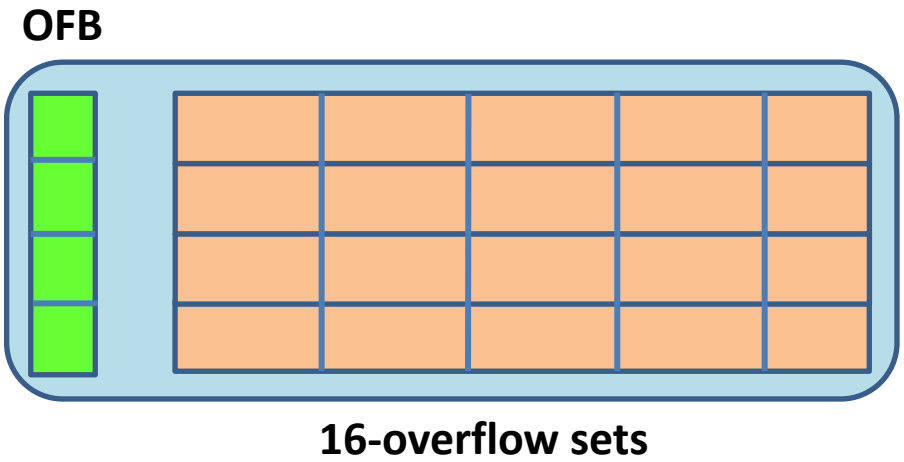
Scalable Structure for RA



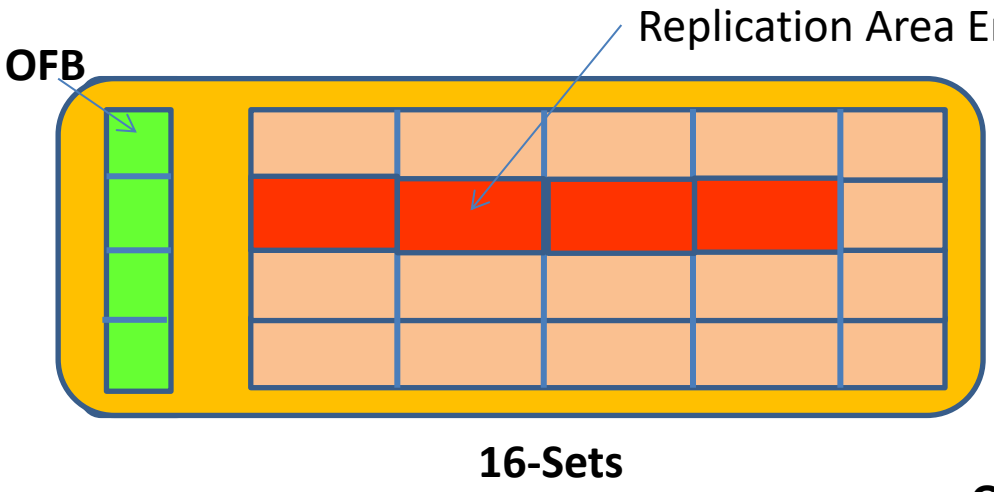


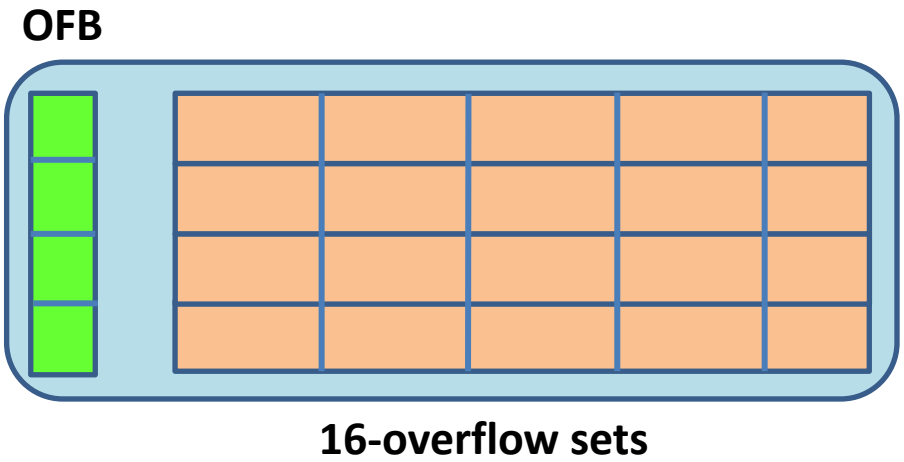
Scalable Structure for RA



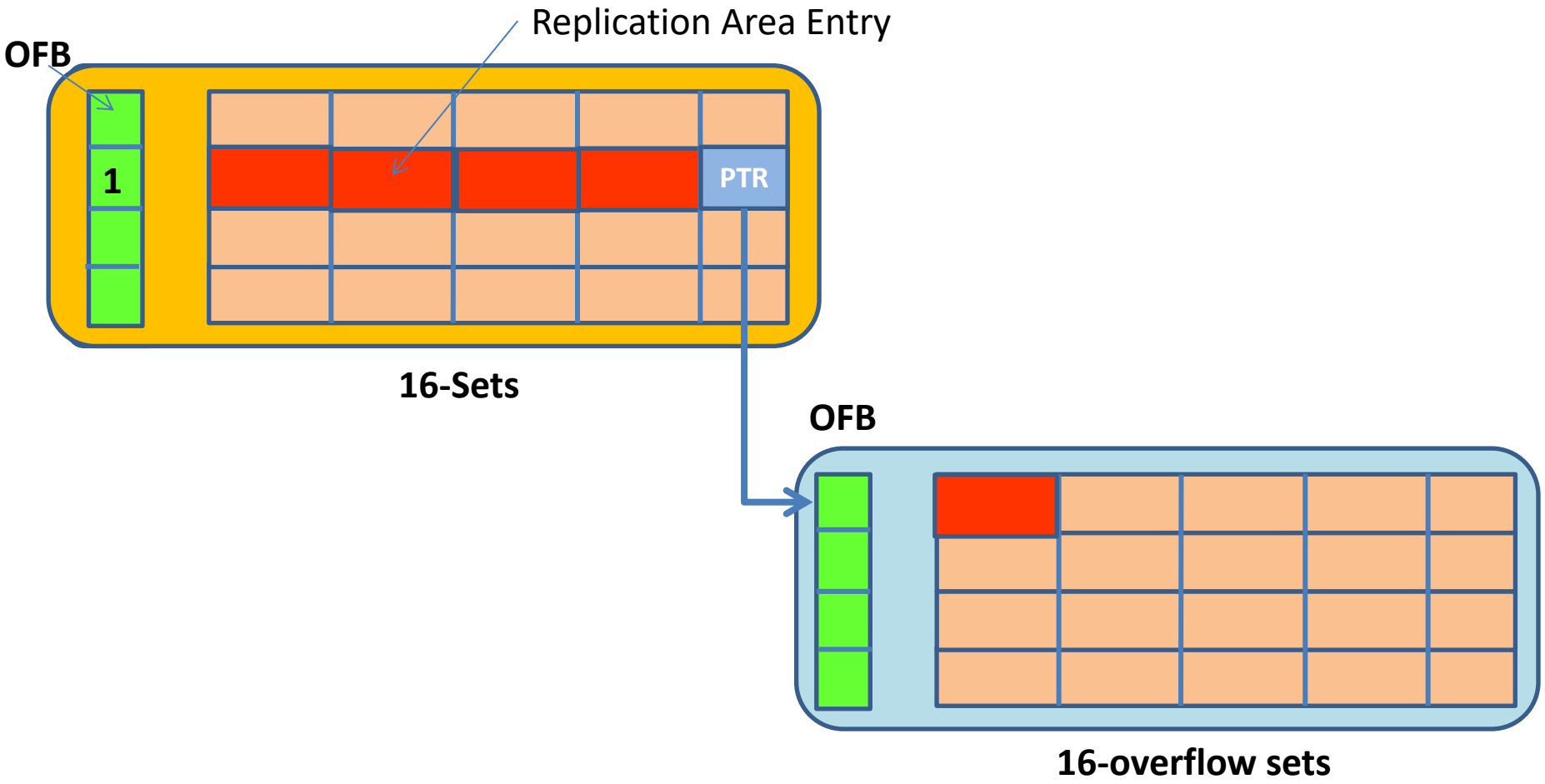


Scalable Structure for RA

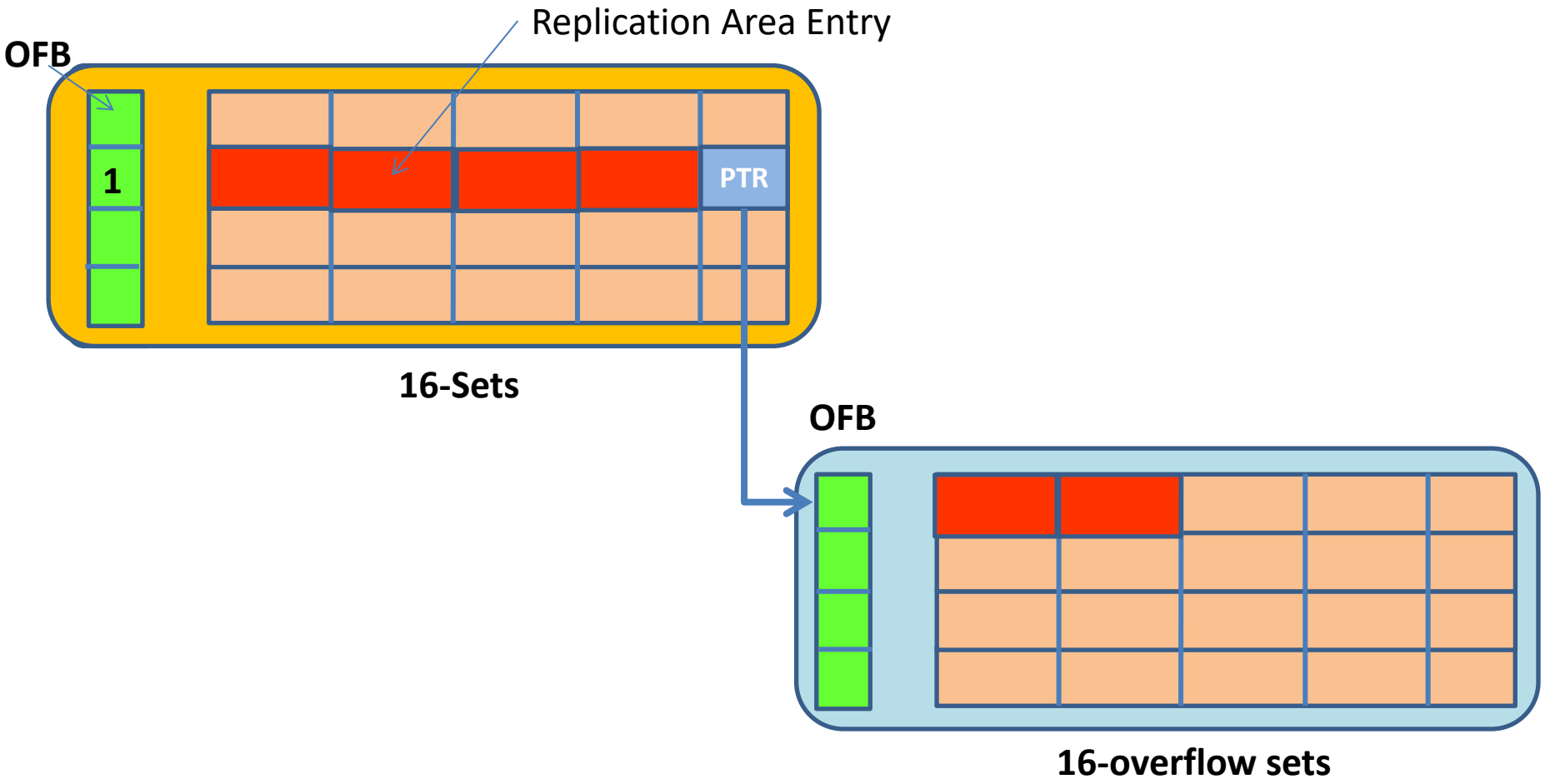




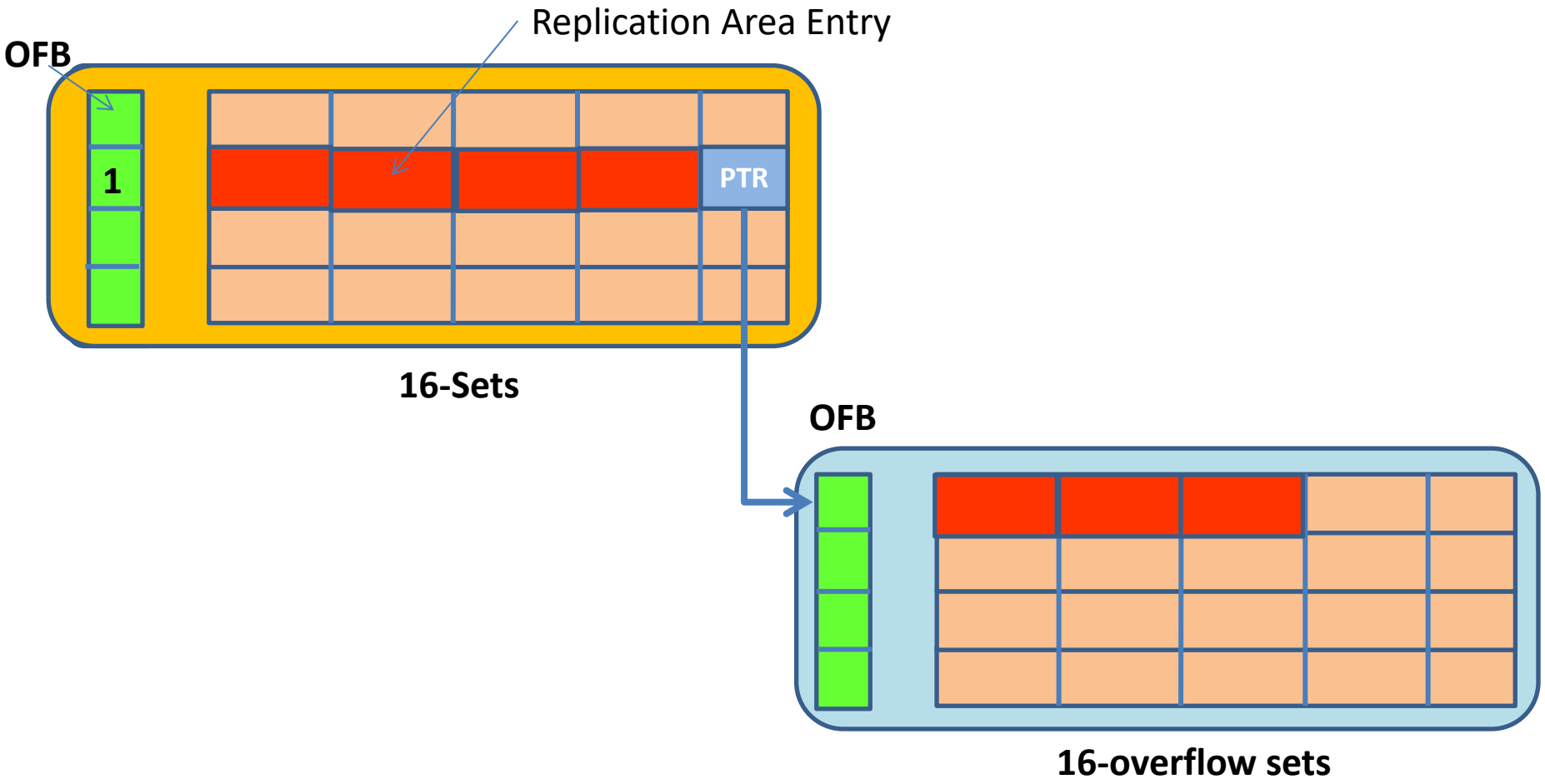
Scalable Structure for RA



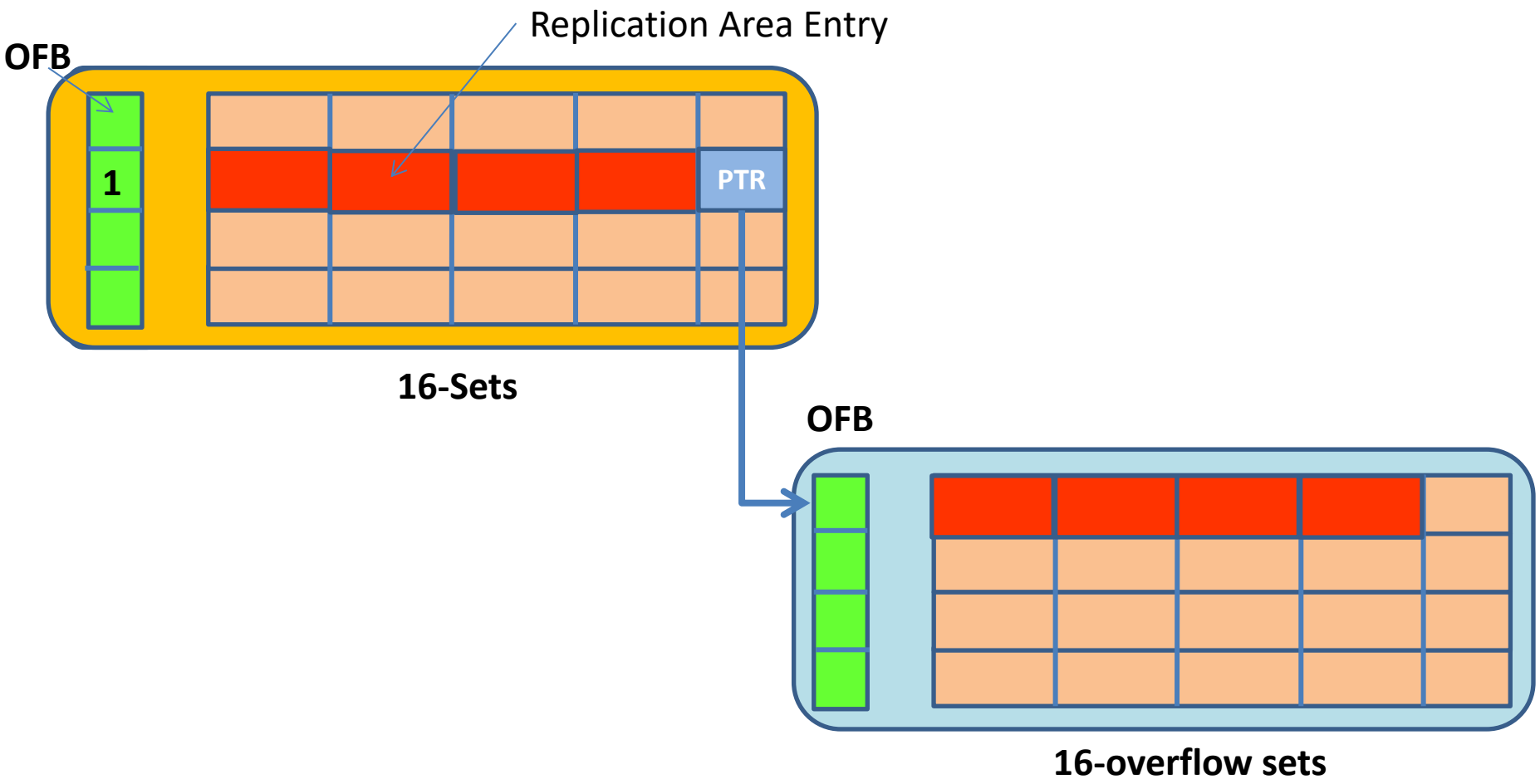
Scalable Structure for RA



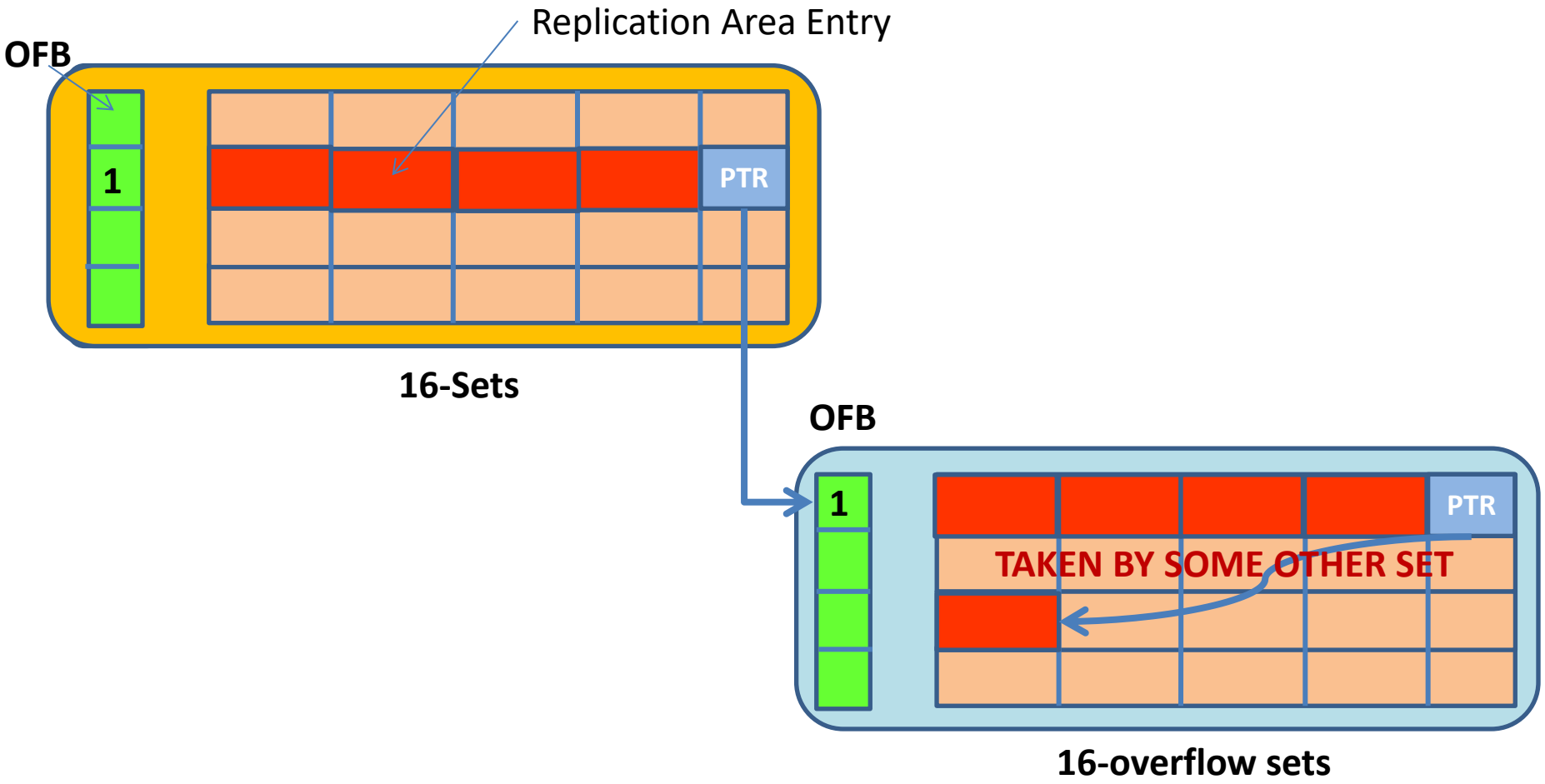
Scalable Structure for RA



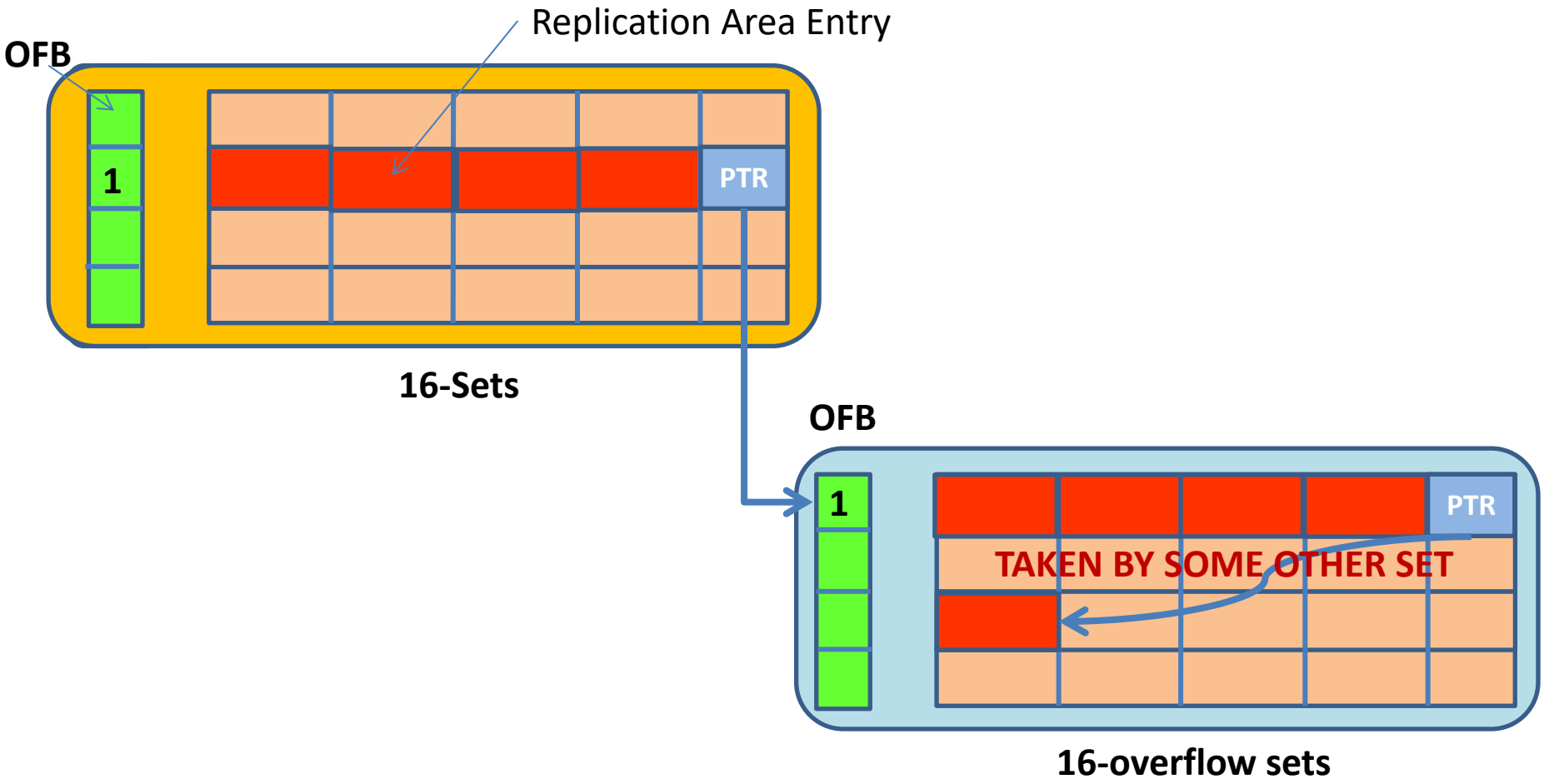
Scalable Structure for RA



Scalable Structure for RA

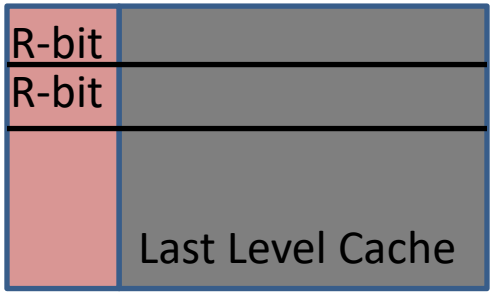


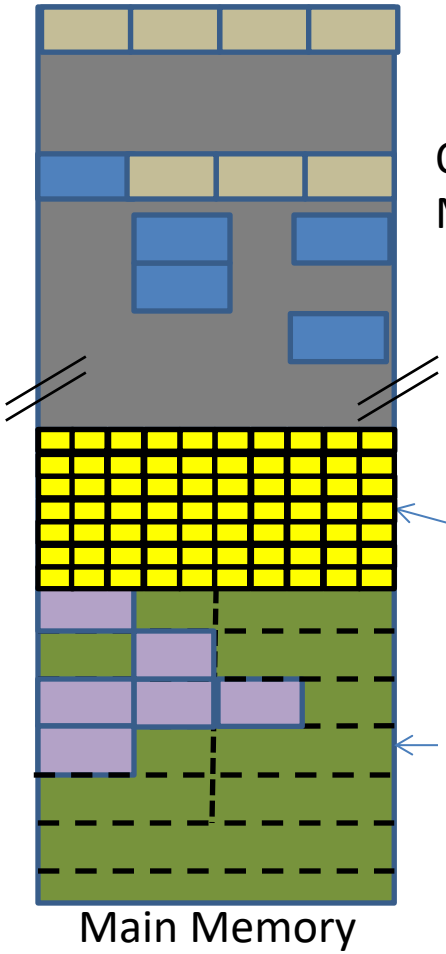
Scalable Structure for RA



With Overflow Sets, Replication Area can handle non uniformity

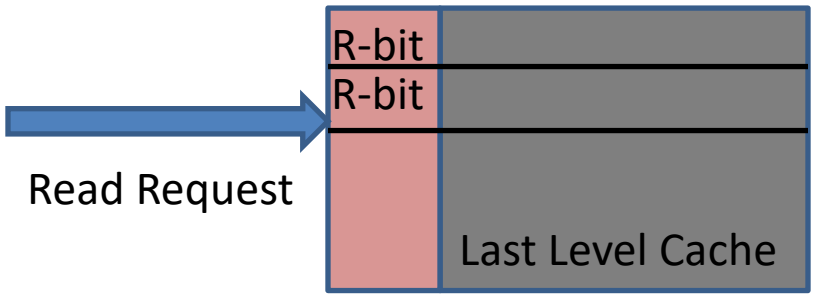
ArchShield: Operation

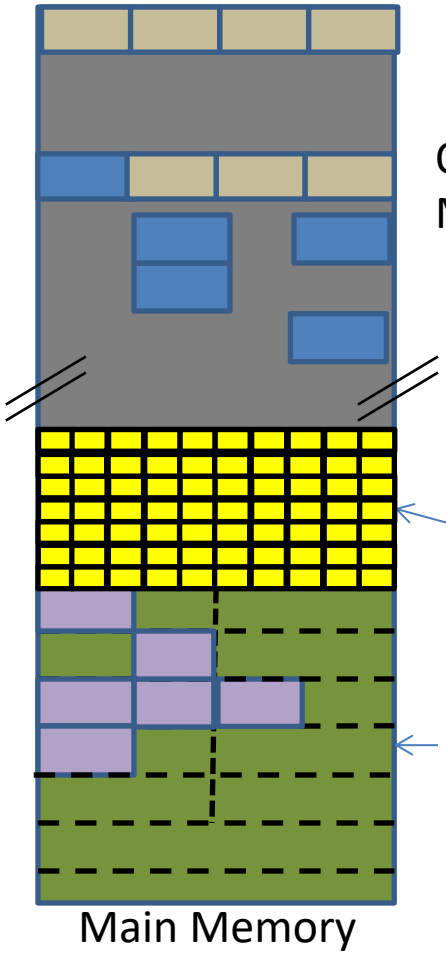


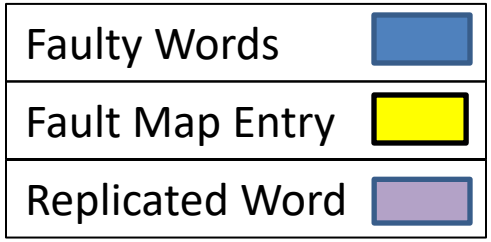


Faulty Words	
Fault Map Entry	
Replicated Word	

ArchShield: Operation

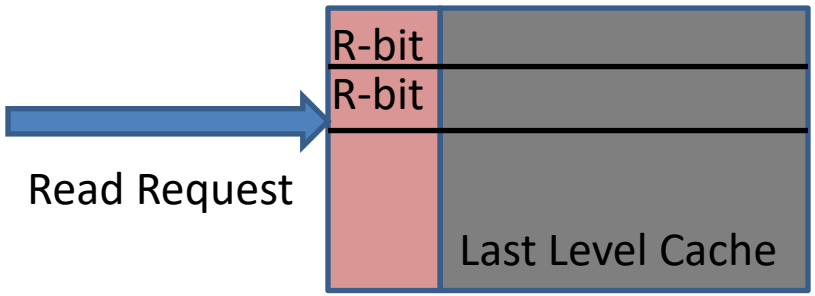


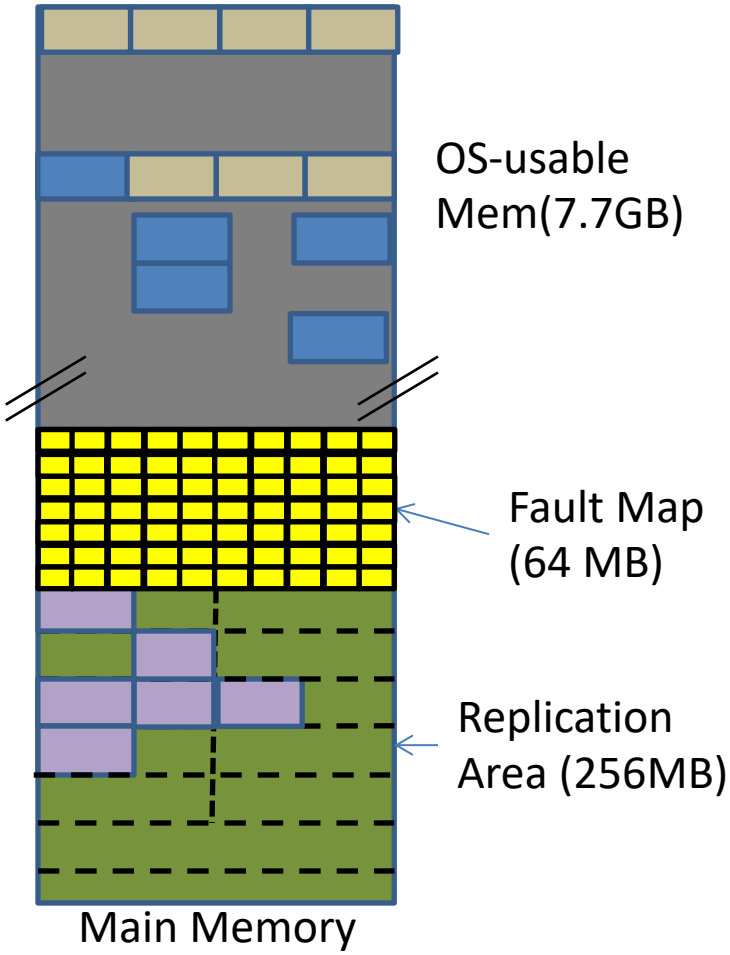




ArchShield: Operation

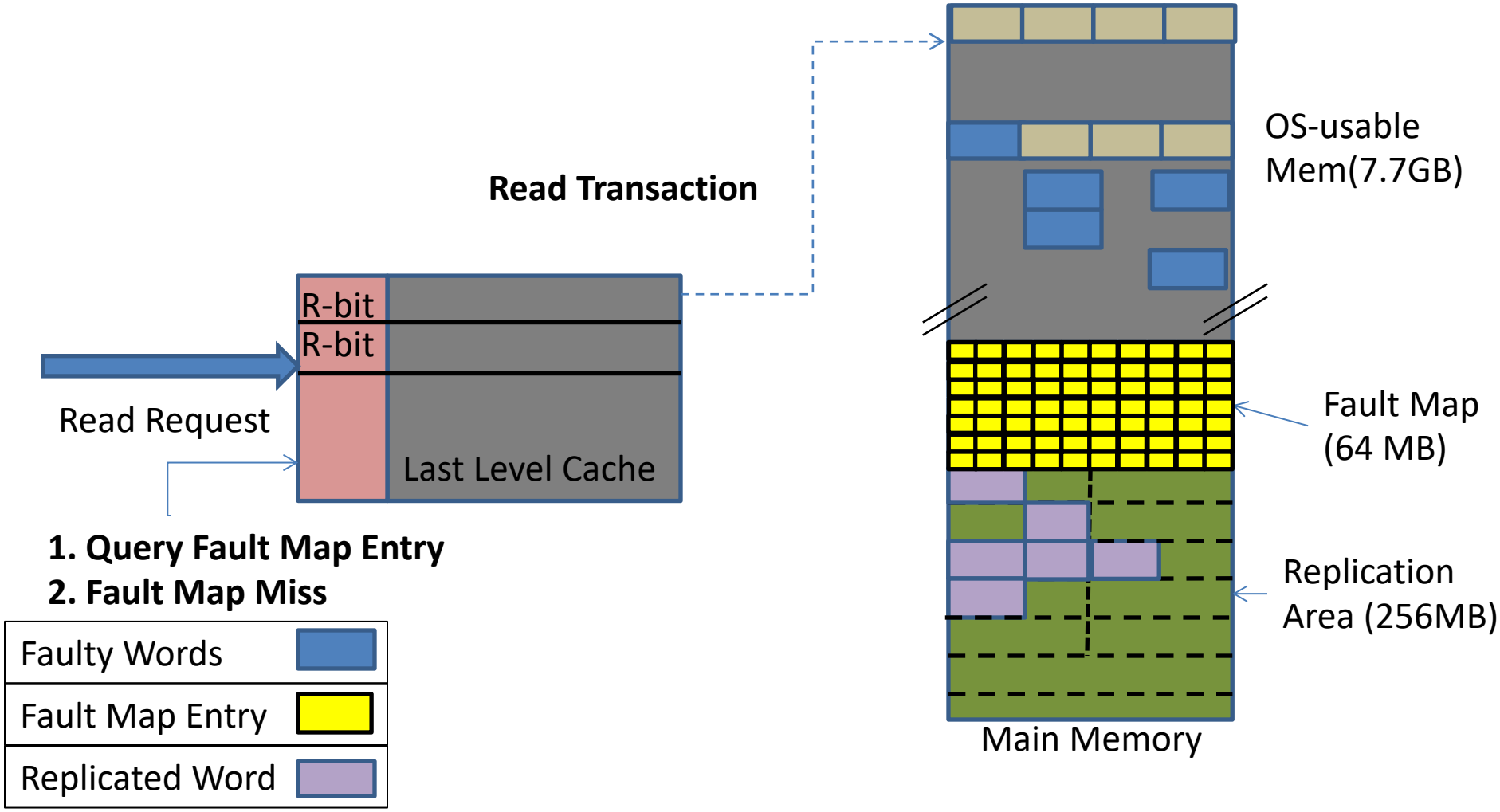
Last Level Cache Miss!



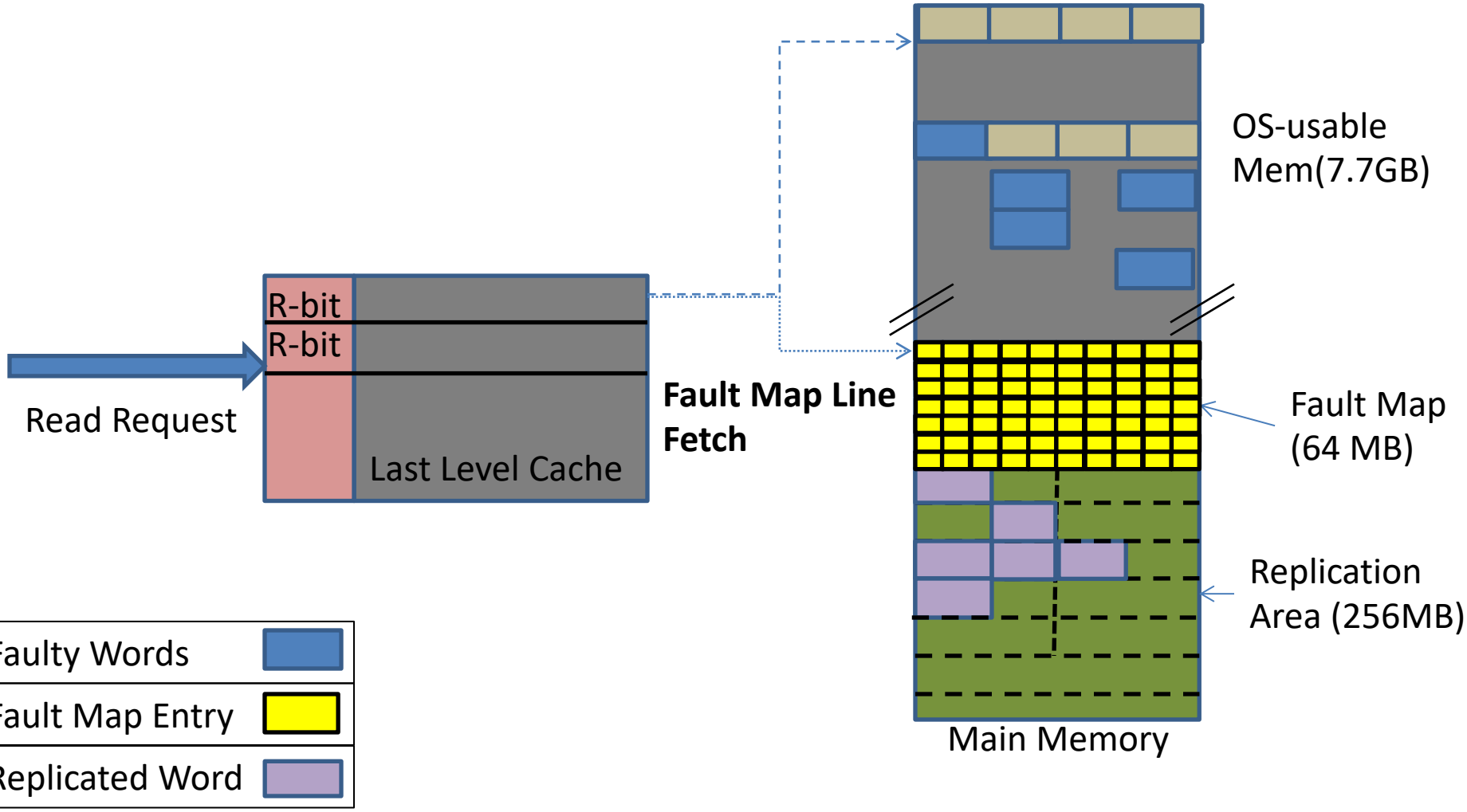


Faulty Words
Fault Map Entry
Replicated Word

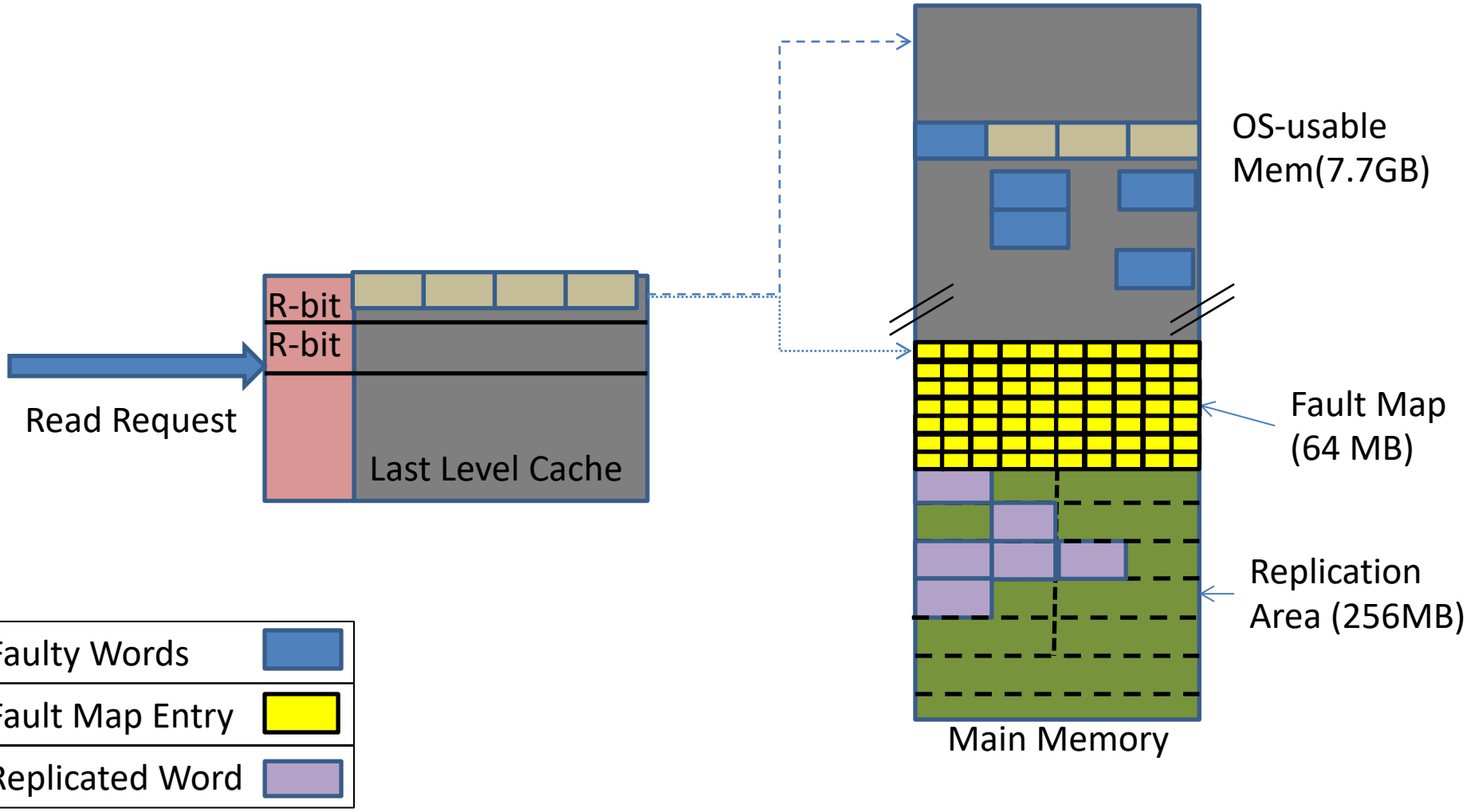
ArchShield: Operation



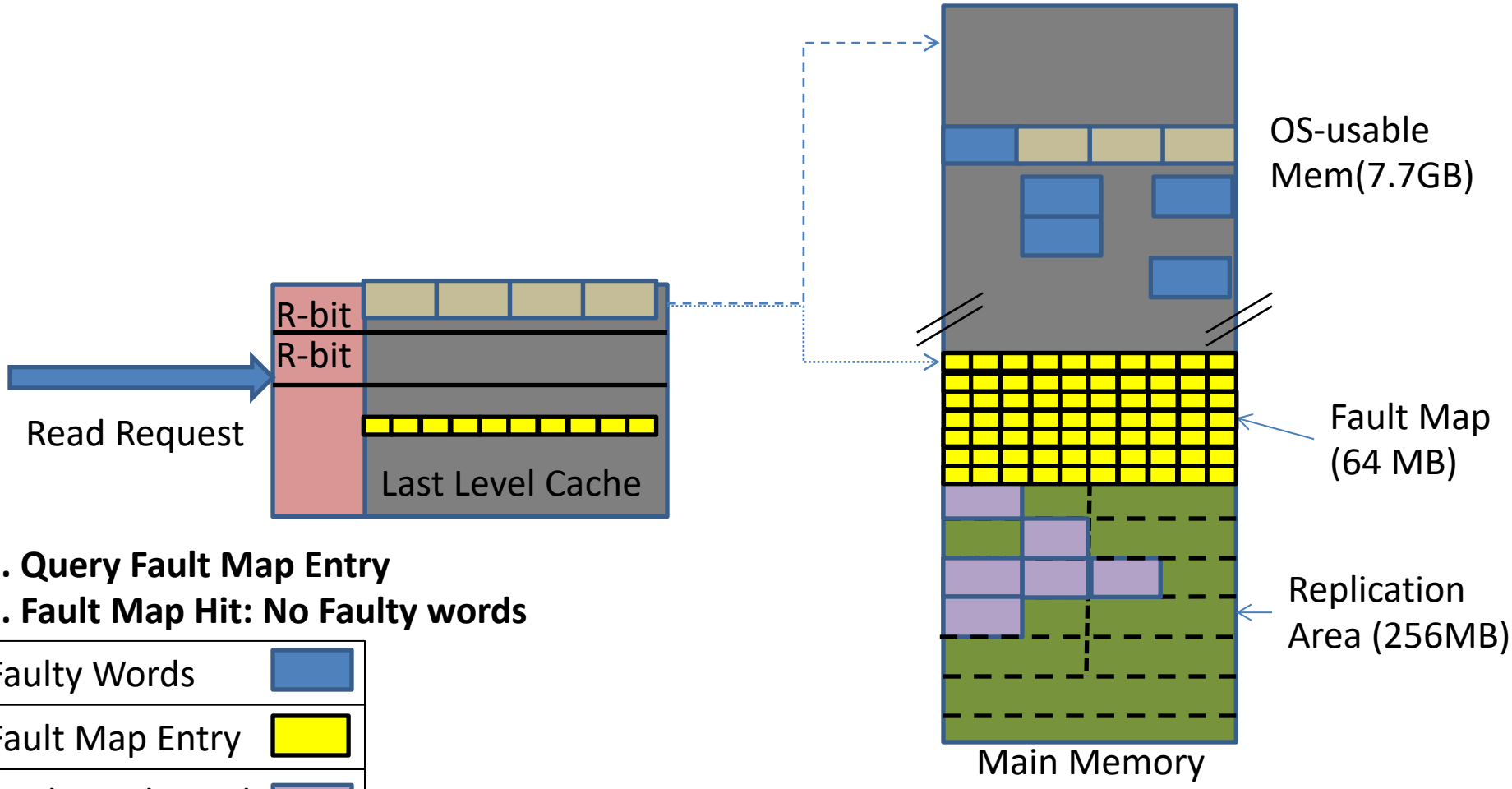
ArchShield: Operation



ArchShield: Operation



ArchShield: Operation

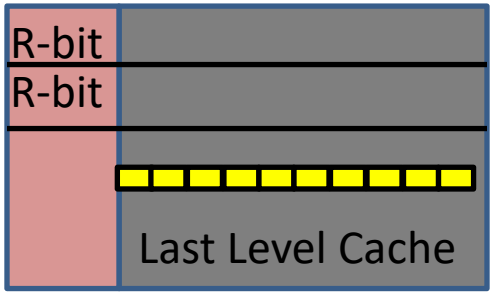


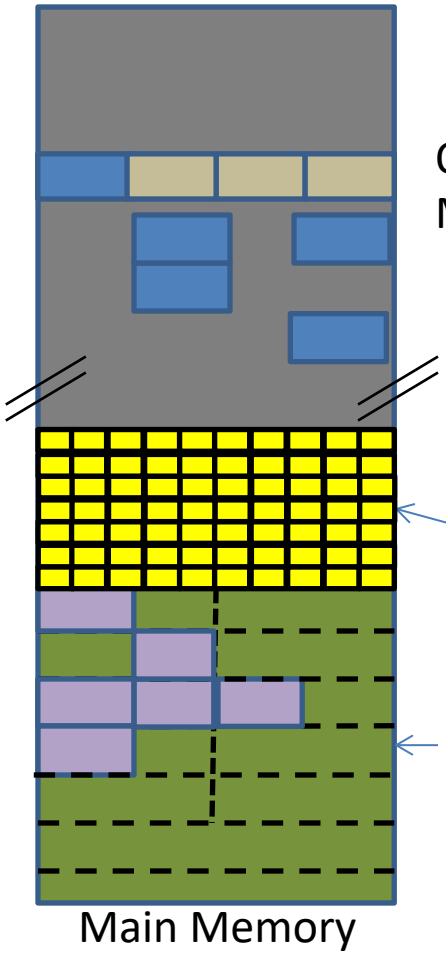
1. Query Fault Map Entry

2. Fault Map Hit: No Faulty words

Faulty Words	
Fault Map Entry	
Replicated Word	

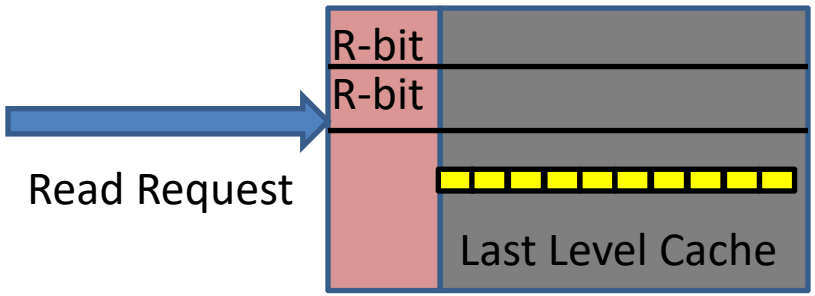
ArchShield: Operation

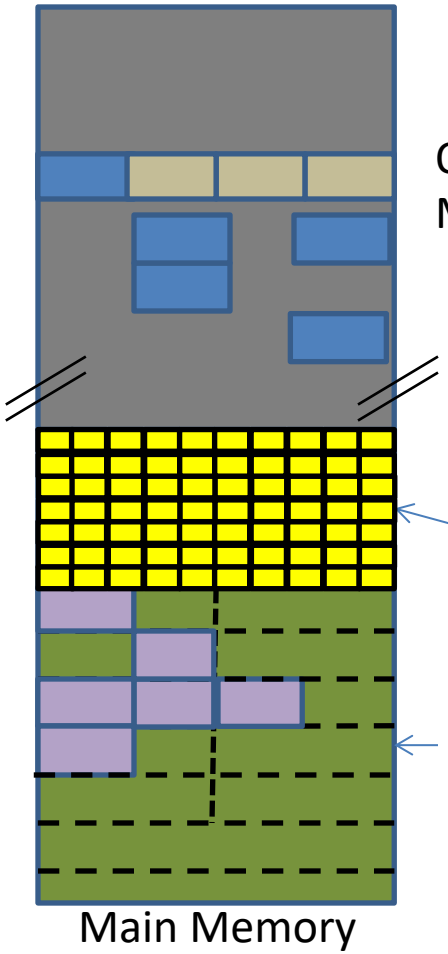


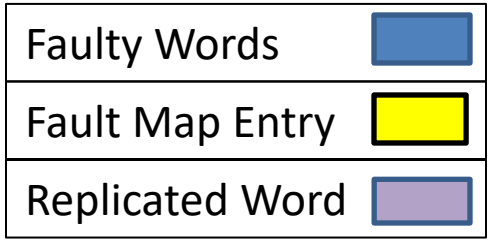


Faulty Words
Fault Map Entry
Replicated Word

ArchShield: Operation

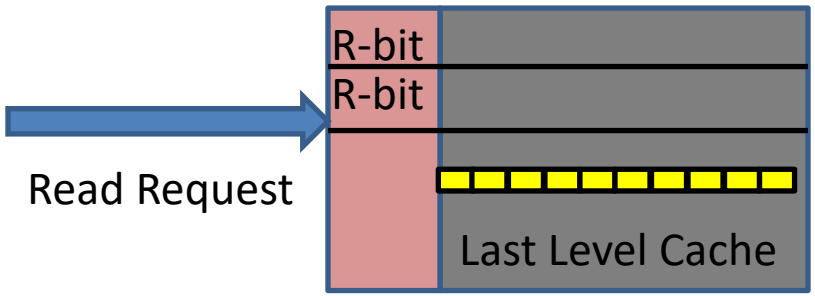


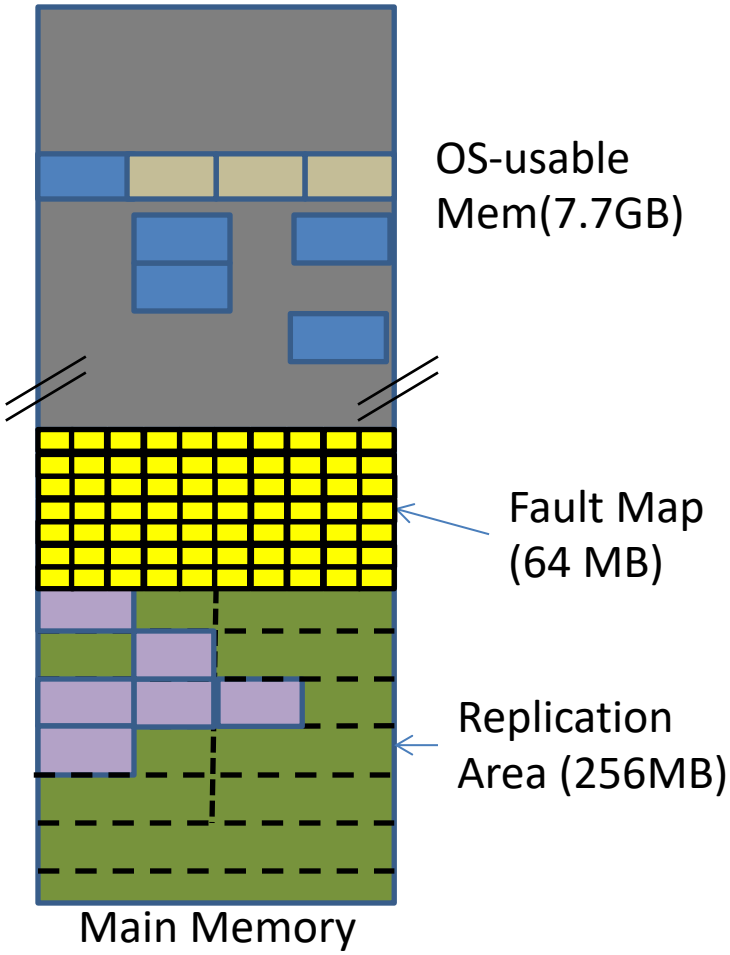




ArchShield: Operation

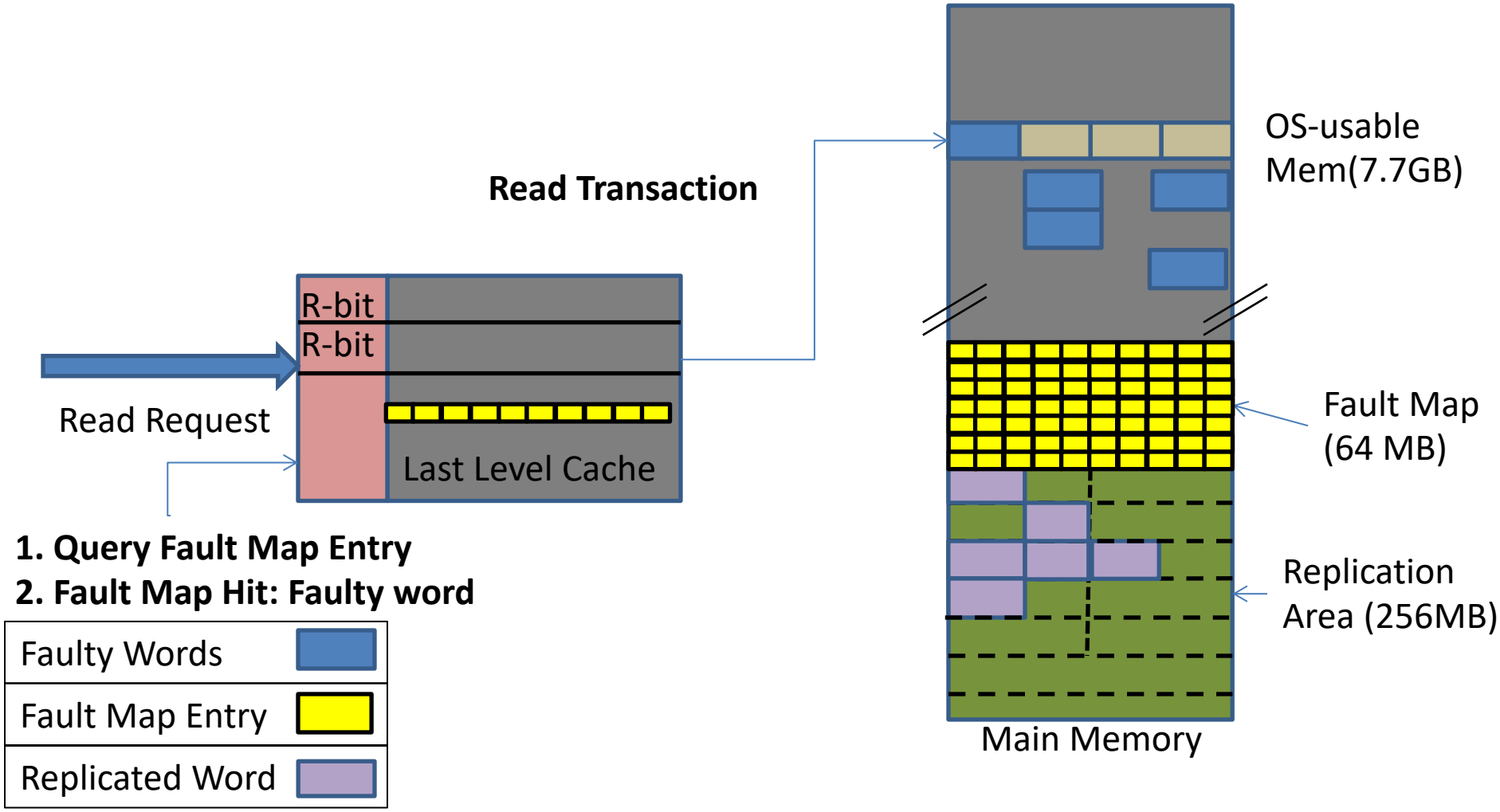
Last Level Cache Miss!



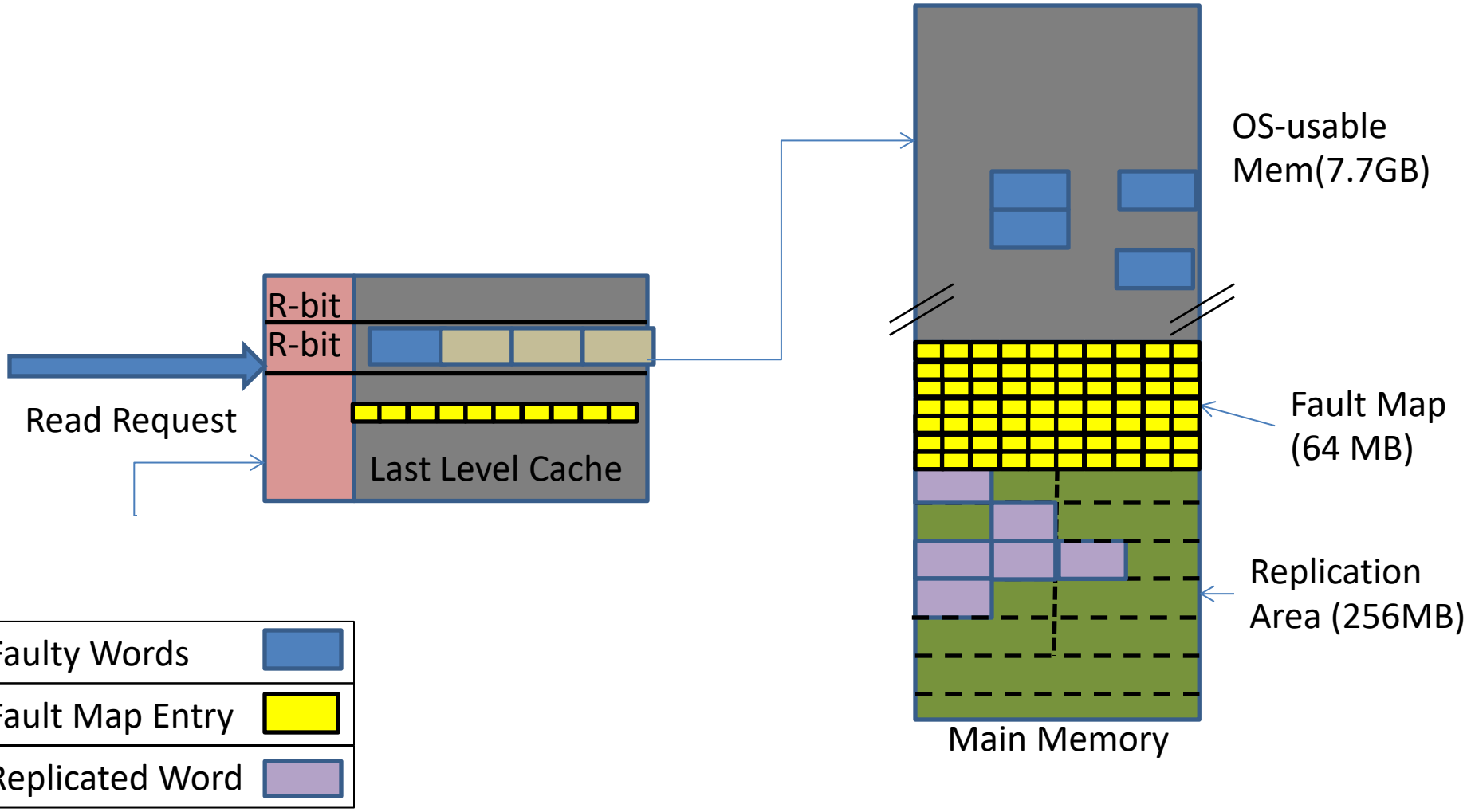


Faulty Words
Fault Map Entry
Replicated Word

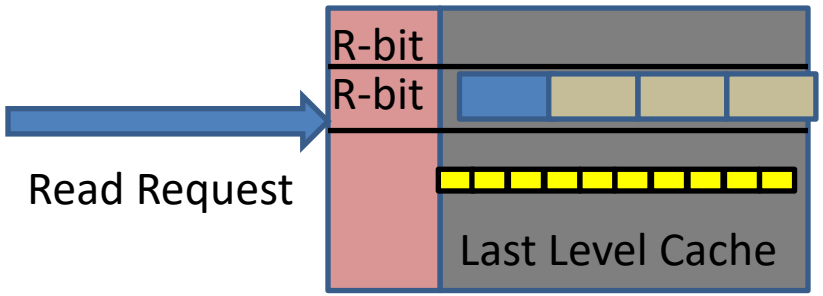
ArchShield: Operation

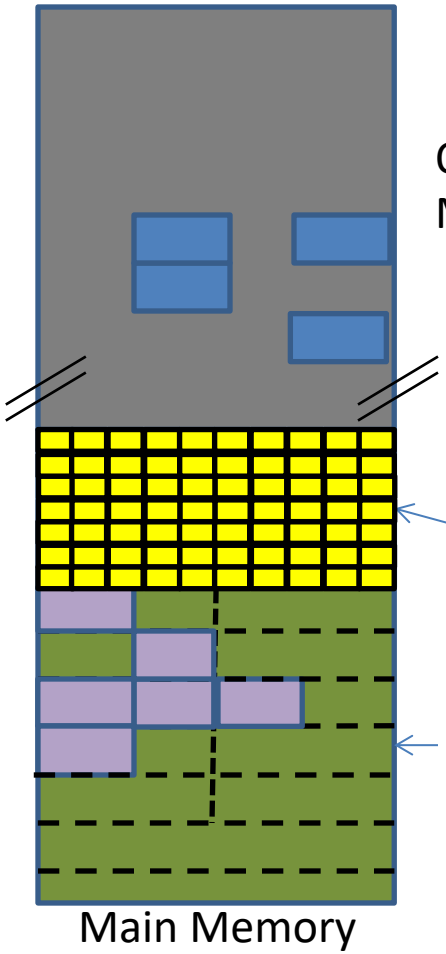


ArchShield: Operation



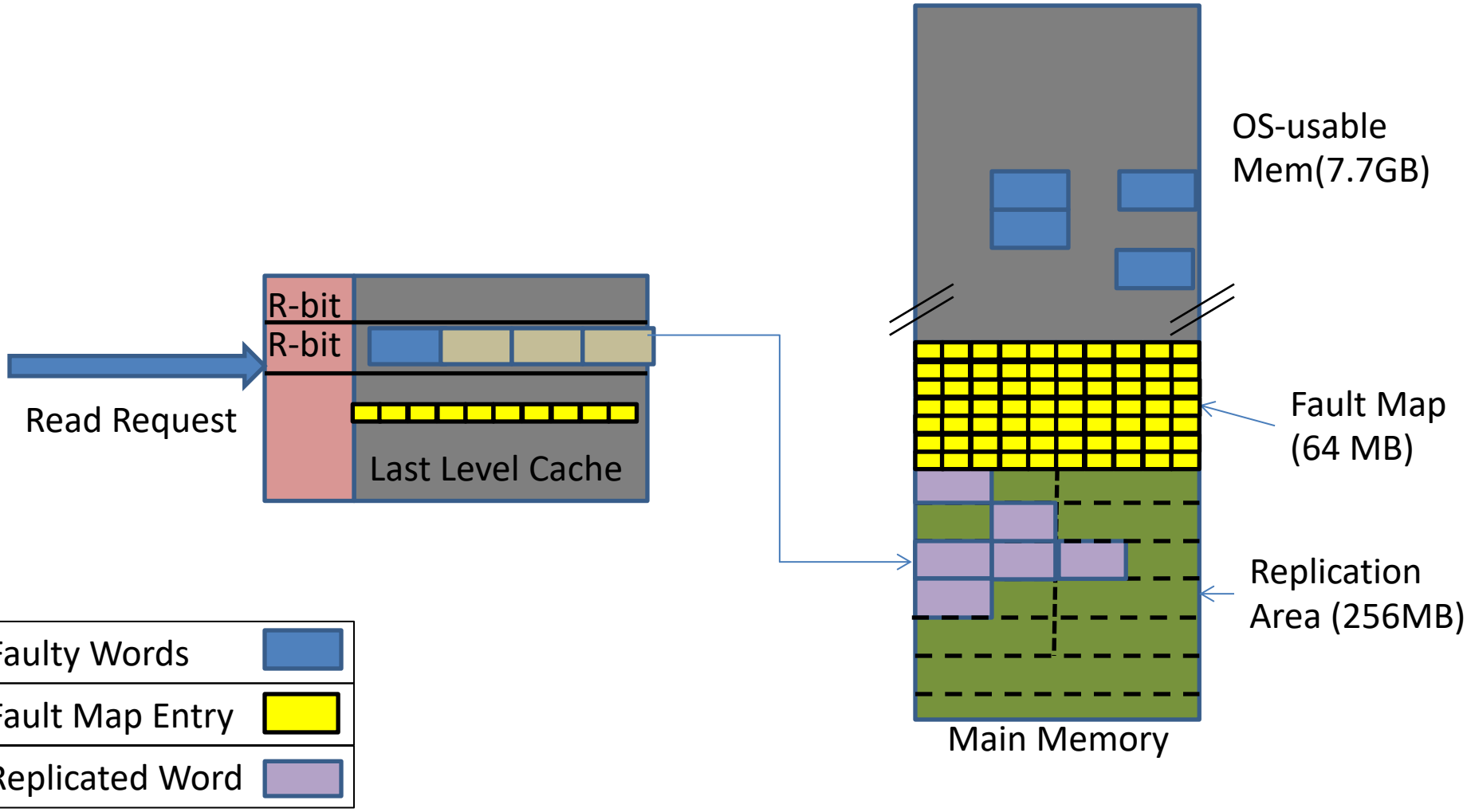
ArchShield: Operation



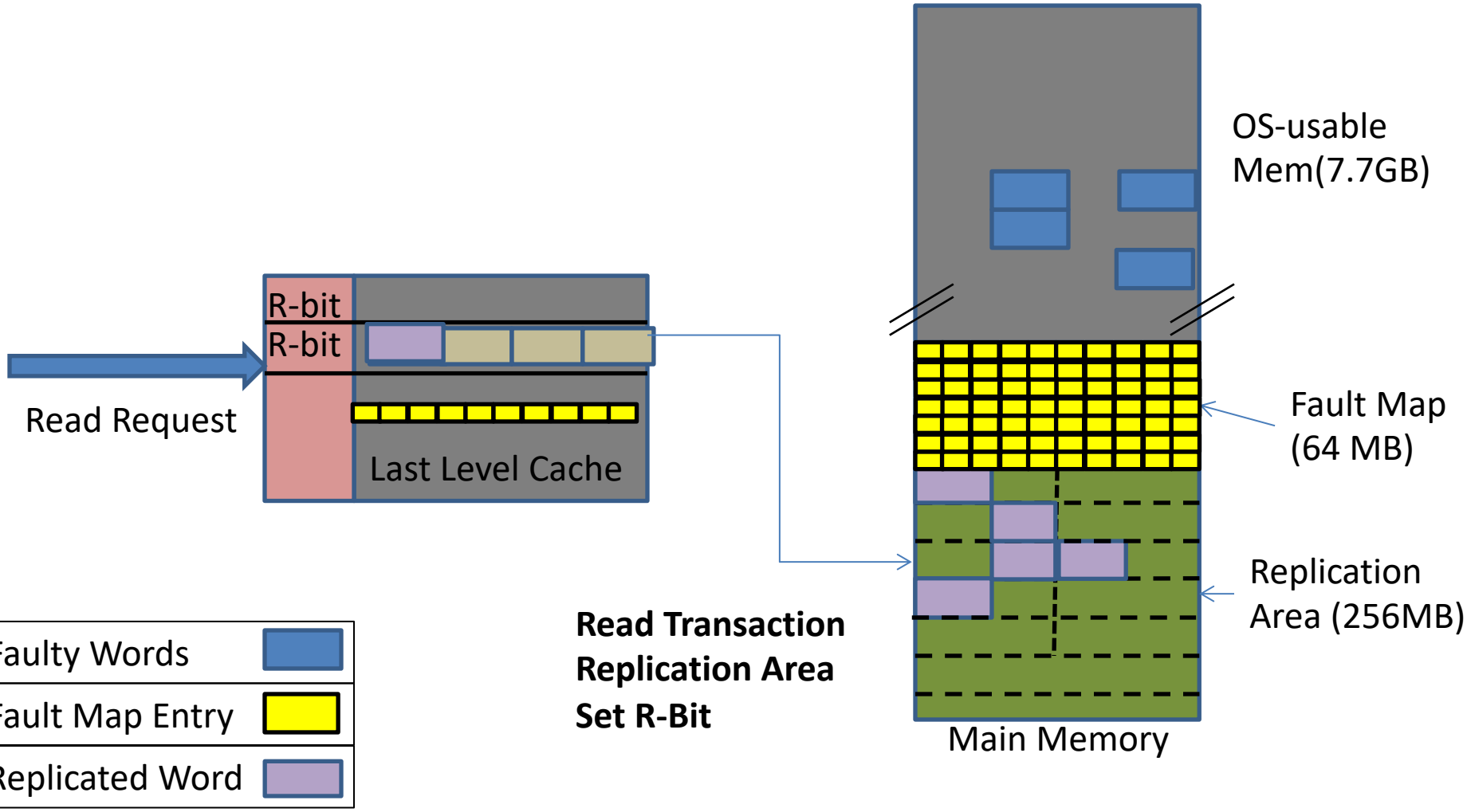


Faulty Words
Fault Map Entry
Replicated Word

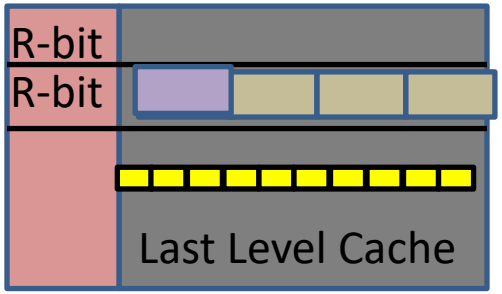
ArchShield: Operation

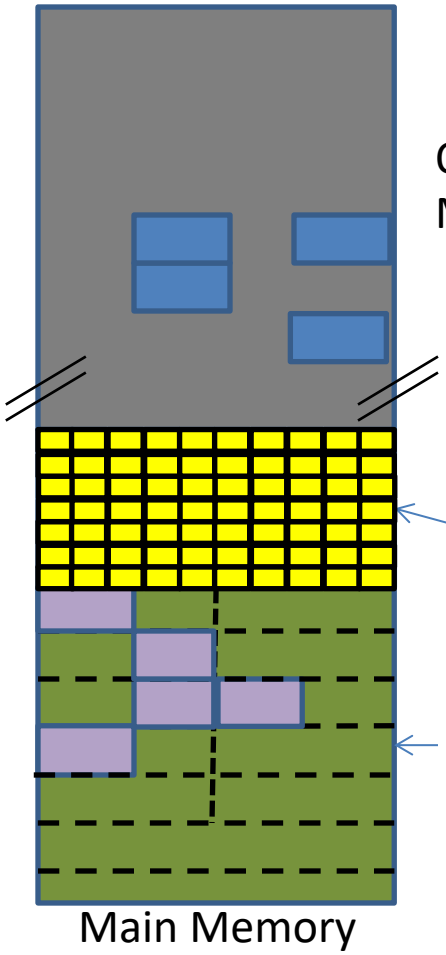


ArchShield: Operation



ArchShield: Operation



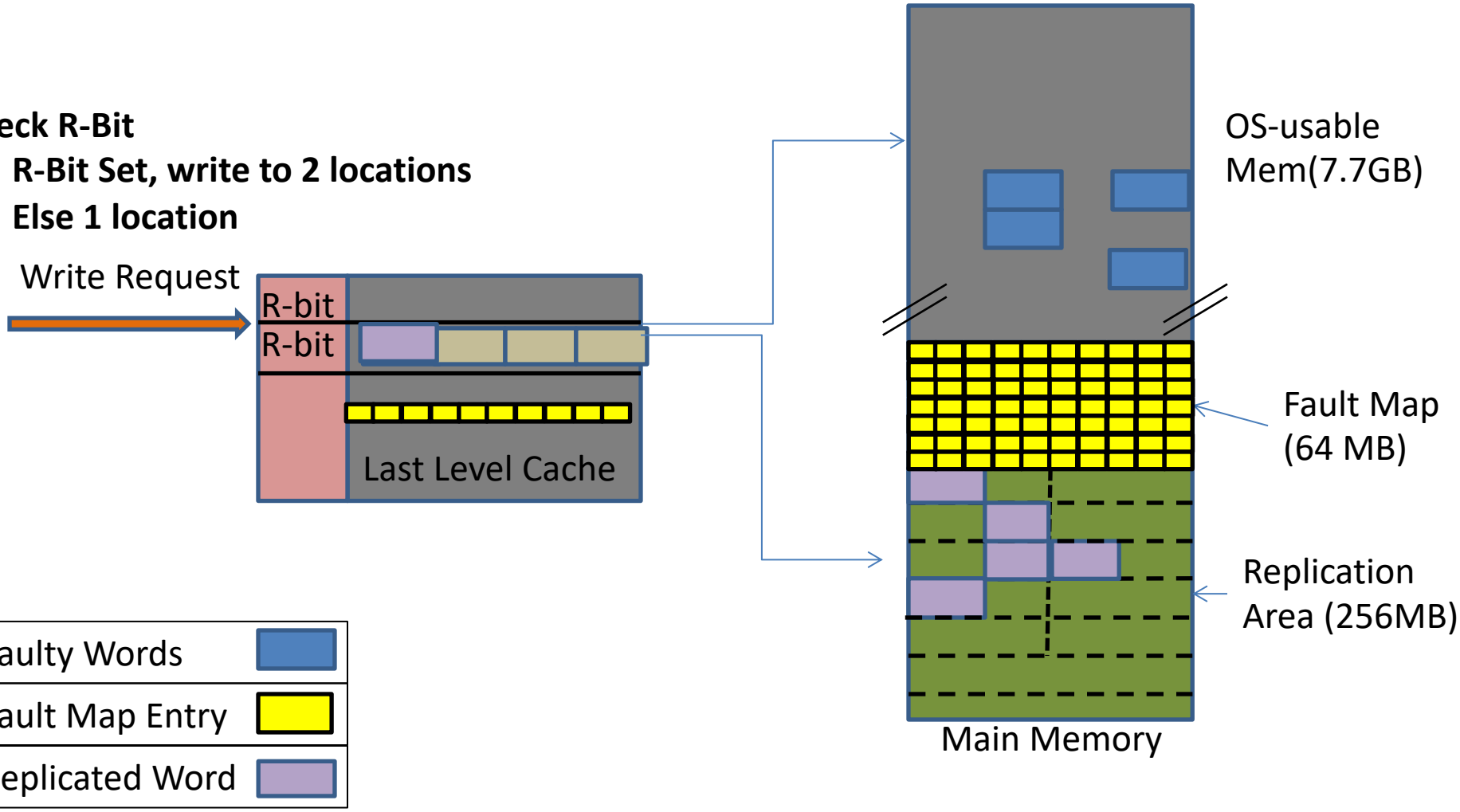


Faulty Words
Fault Map Entry
Replicated Word

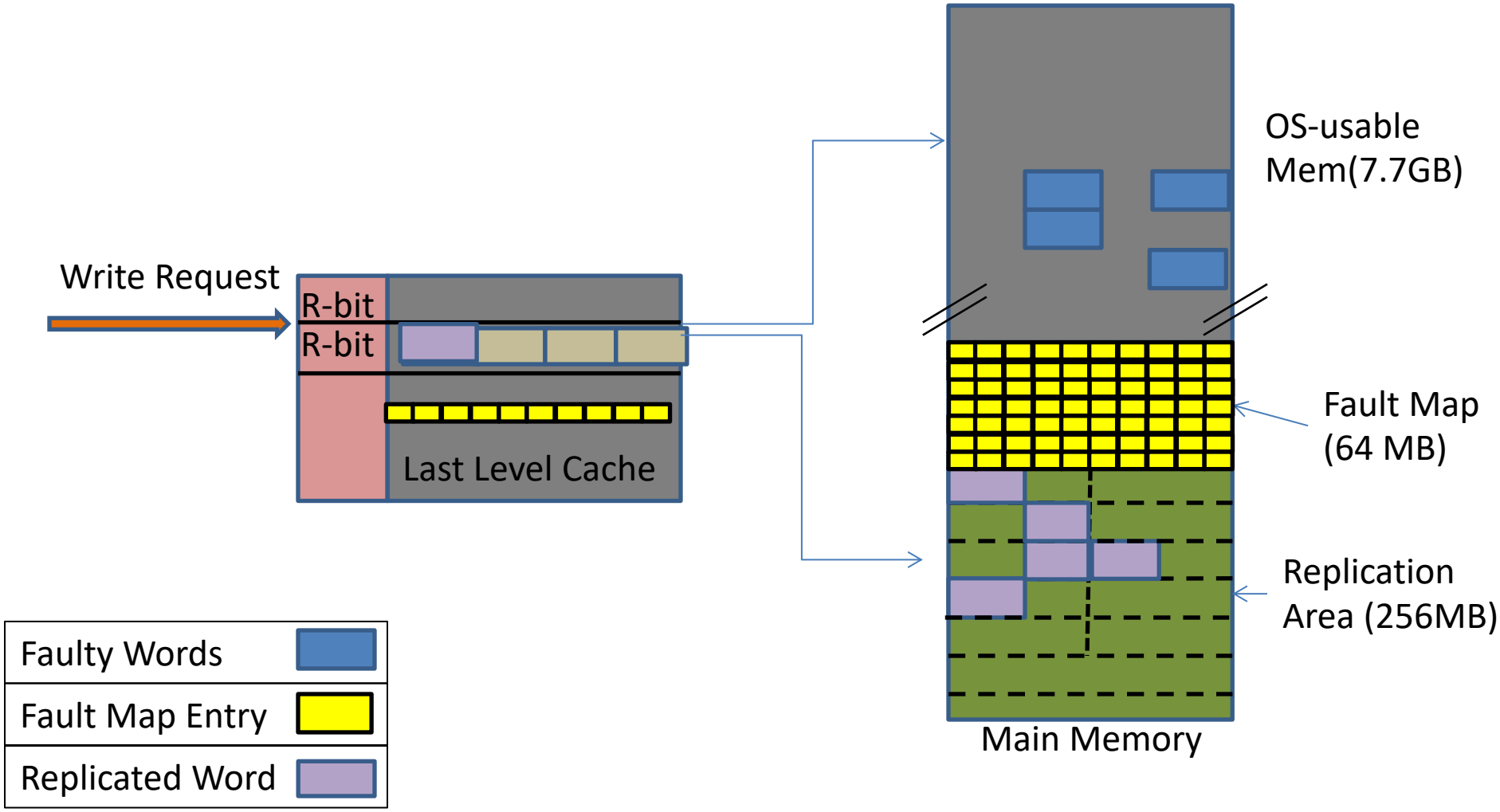
ArchShield: Operation

Check R-Bit

1. **R-Bit Set, write to 2 locations**
2. **Else 1 location**



ArchShield: Operation



Outline

- Introduction
- Current Schemes
- ArchShield
- Evaluation
- Summary

Experimental Evaluation

Configuration:

8-core CMP with 8MB LLC (shared)

8GB DIMM, two channels DDR3-1600

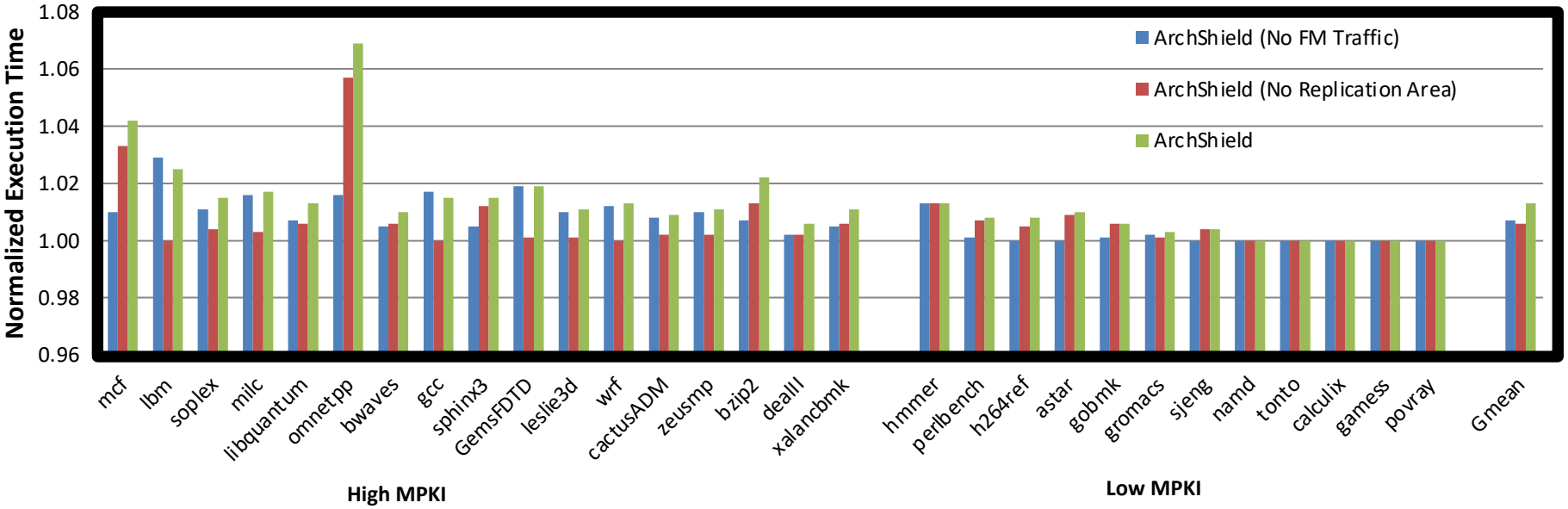
Workloads: SPEC CPU 2006 suite in rate mode

Assumptions: Bit error rate of 100ppm (random faults)

Performance Metric: Execution time norm to fault free baseline

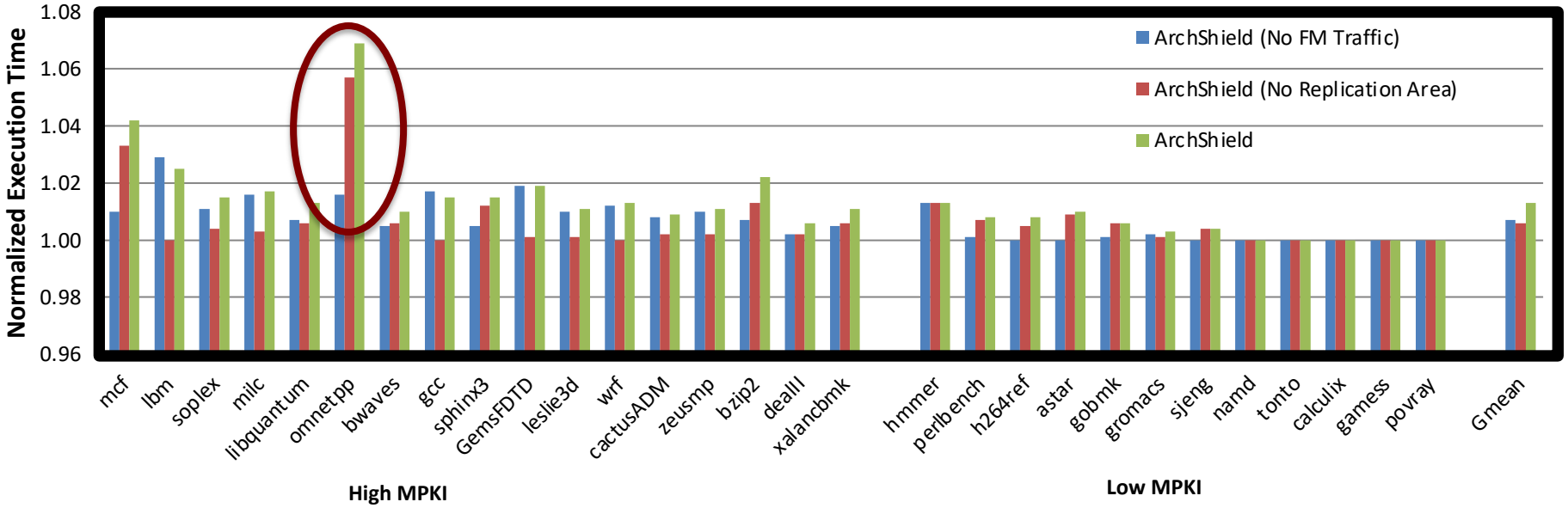
Execution Time

Two sources of slowdown:
Fault Map access and Replication Area access



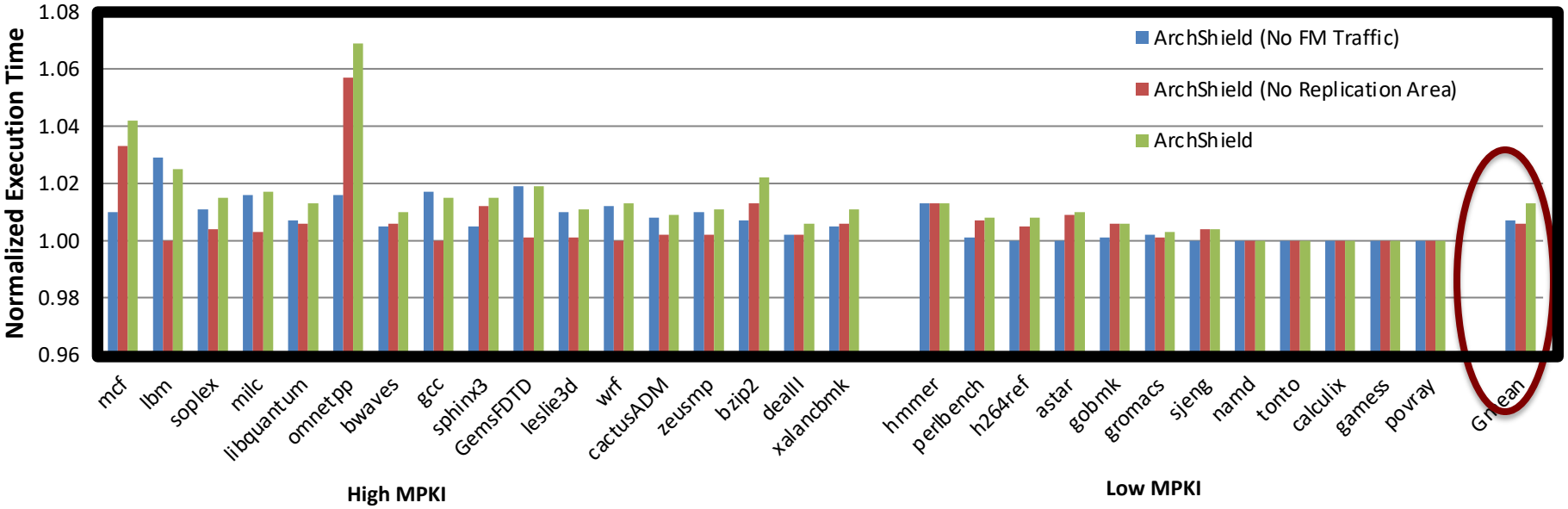
Execution Time

Two sources of slowdown:
Fault Map access and Replication Area access



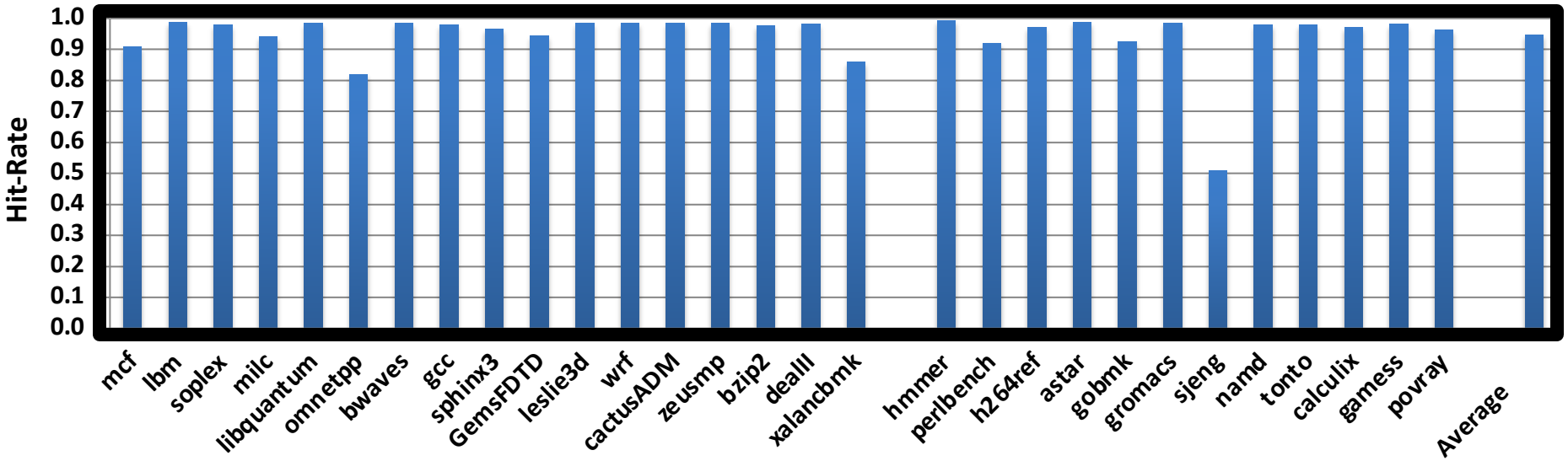
Execution Time

Two sources of slowdown:
Fault Map access and Replication Area access



On average, ArchShield causes 1% slowdown

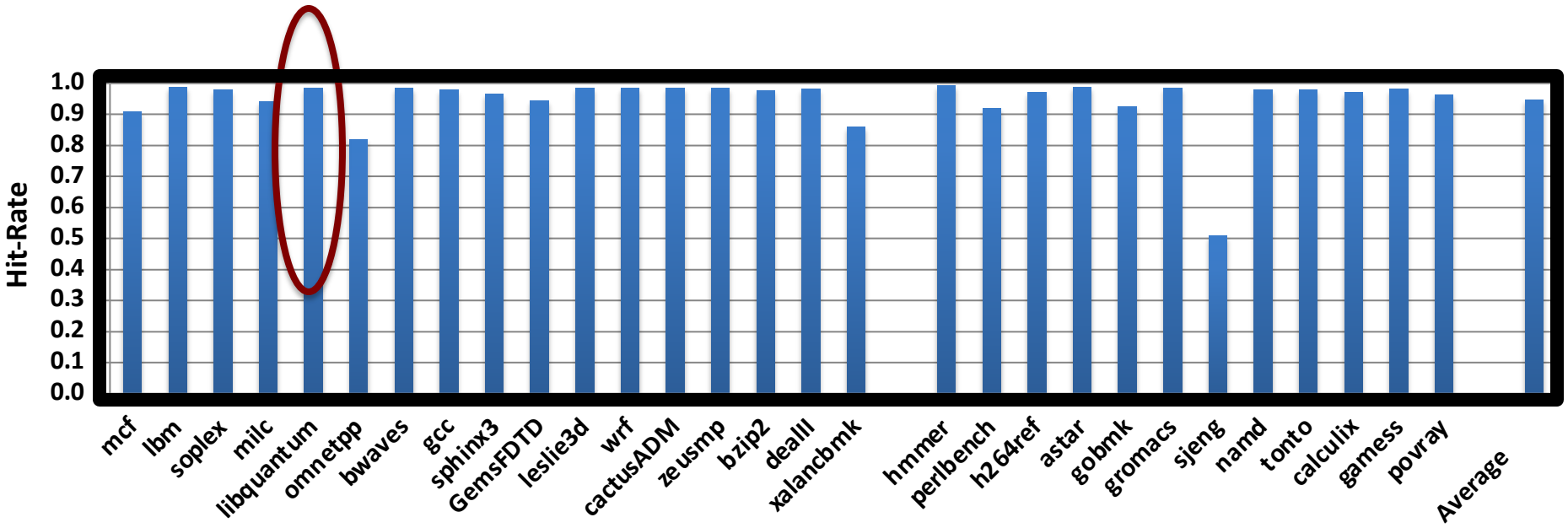
Fault Map Hit-Rate



High MPKI

Low MPKI

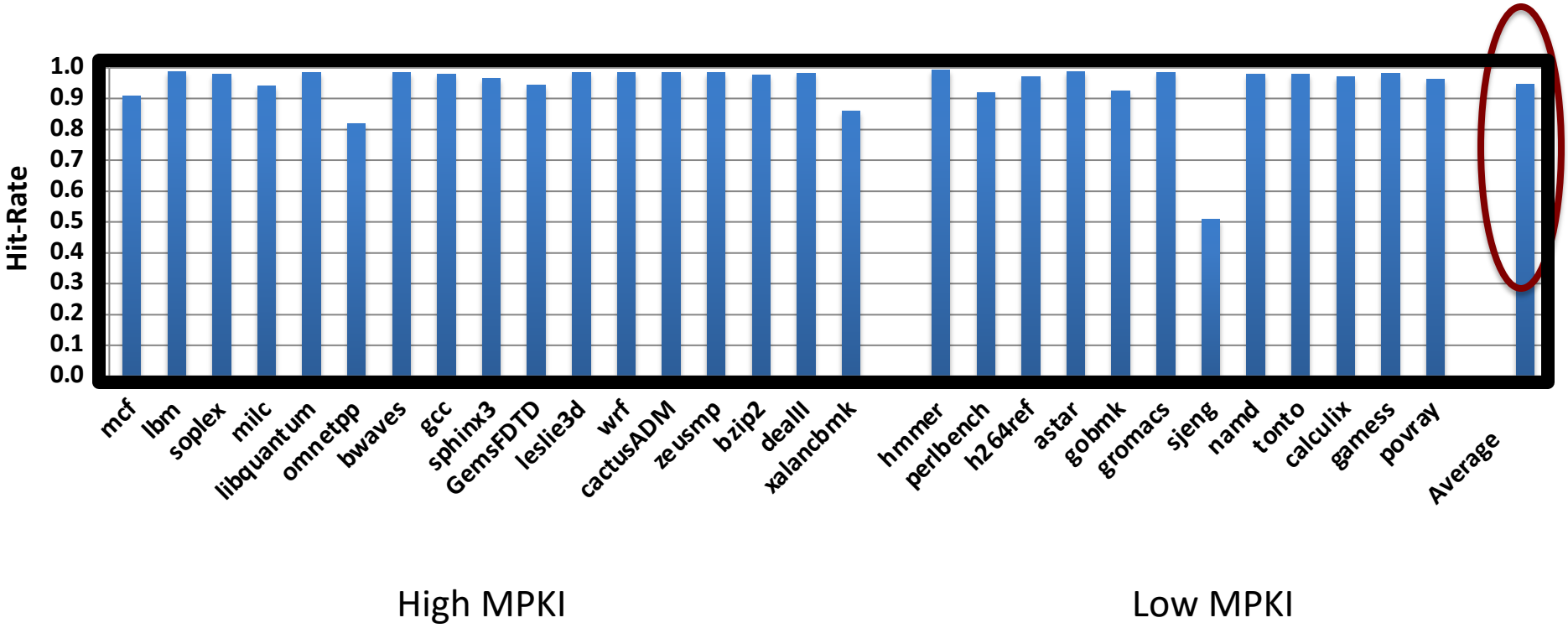
Fault Map Hit-Rate



High MPKI

Low MPKI

Fault Map Hit-Rate



High MPKI

Low MPKI

Hit rate of Fault Map in LLC is high, on average 95%

Analysis of Memory Operations

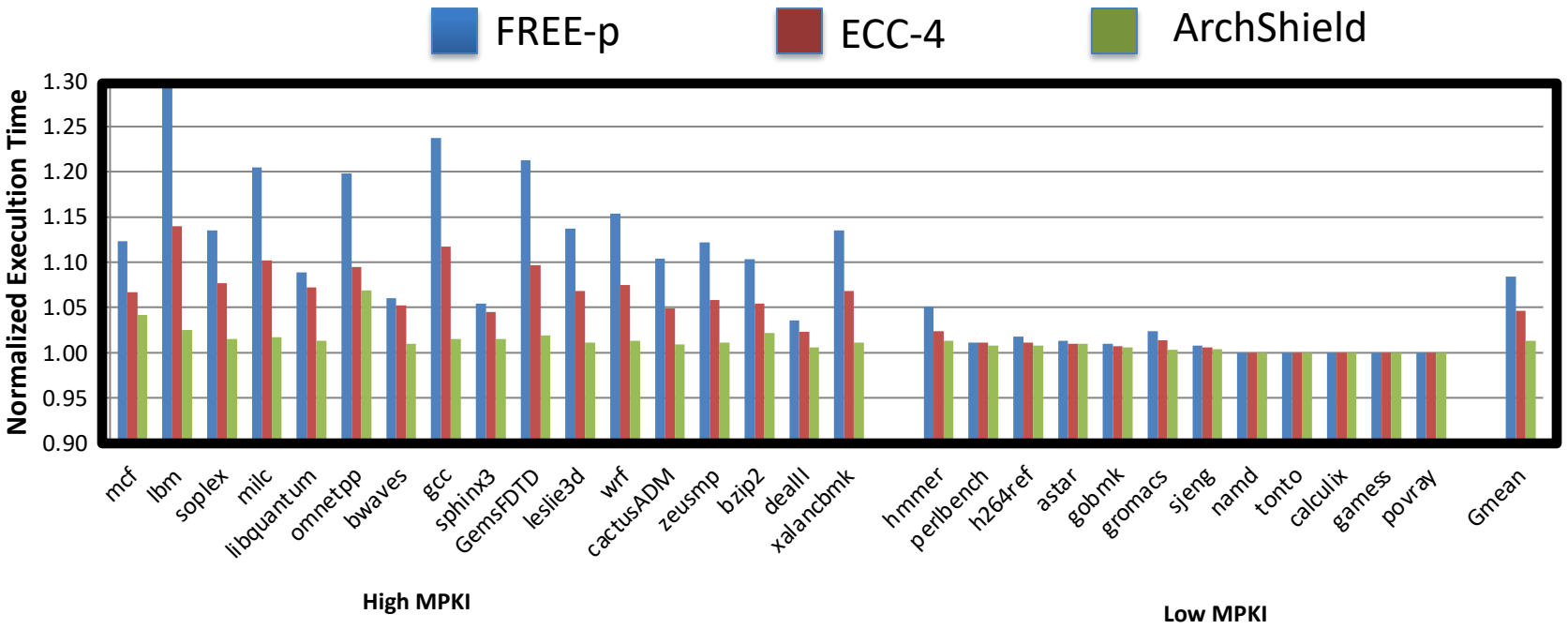
Transaction	1 Access(%)	2 Access (%)	3 Access (%)
Reads	72.1	0.02	~0
Writes	22.1	1.2	0.05
Fault Map	4.5	N/A	N/A
Overall	98.75	1.2	0.05

Analysis of Memory Operations

Transaction	1 Access(%)	2 Access (%)	3 Access (%)
Reads	72.1	0.02	~0
Writes	22.1	1.2	0.05
Fault Map	4.5	N/A	N/A
Overall	98.75	1.2	0.05

1. Only 1.2% of the total accesses use the Replication Area
2. Fault Map Traffic accounts for <5 % of all traffic

Comparison With Other Schemes



1. The read-before-write of Free-p + strong ECC ➔ high latency
2. ECC-4 incurs decoding delay
3. The impact of execution time is minimum in ArchShield

Outline

- Introduction
- Current Schemes
- ArchShield
- Evaluation
- Summary

Summary

- DRAM scaling challenge: High fault rate, current schemes limited
- We propose to expose DRAM errors to Architecture ➔ ArchShield
- ArchShiled uses efficient Fault Map and Selective Word Replication
- ArchShield handles a Bit Error Rate of 100ppm with less than 4% storage overhead and 1% slowdown

Summary

- DRAM scaling challenge: High fault rate, current schemes limited
- We propose to expose DRAM errors to Architecture ➔ ArchShield
- ArchShiled uses efficient Fault Map and Selective Word Replication
- ArchShield handles a Bit Error Rate of 100ppm with less than 4% storage overhead and 1% slowdown
- ArchShield can be used to reduce DRAM refresh by 16x (to 1 second)