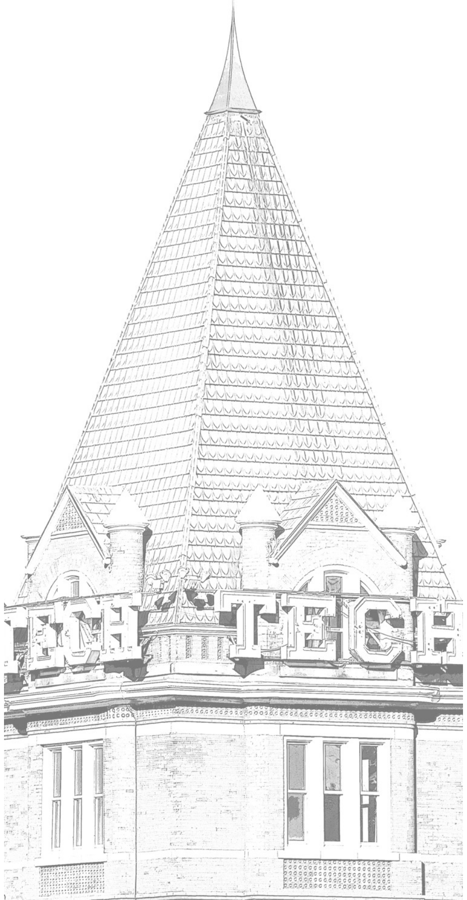
Citadel: Efficiently Protecting Stacked Memory From Large Granularity Failures



Dec 15th 2014

MICRO-47 Cambridge UK

Prashant Nair - Georgia Tech

David Roberts - AMD Research

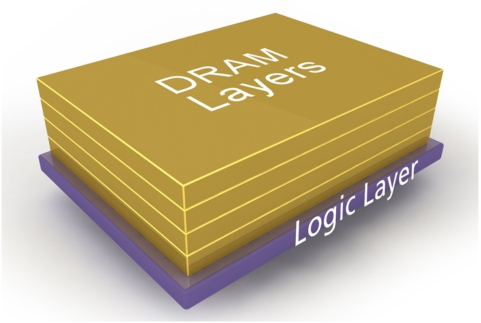
Moinuddin Qureshi - Georgia Tech

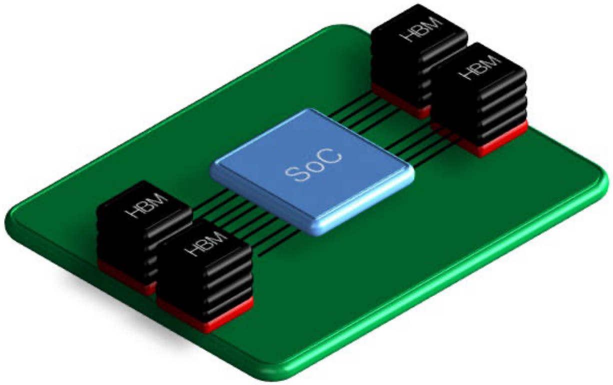
Georgia Institute of Technology

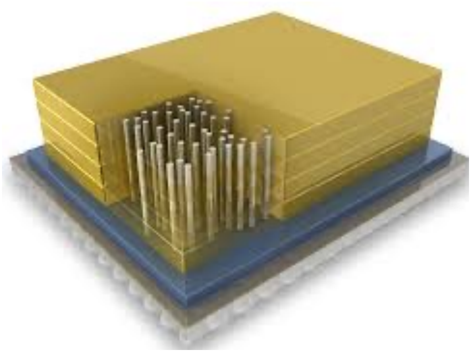
AMD

INTRODUCTION TO 3D DRAM

- DRAM systems face a bandwidth wall







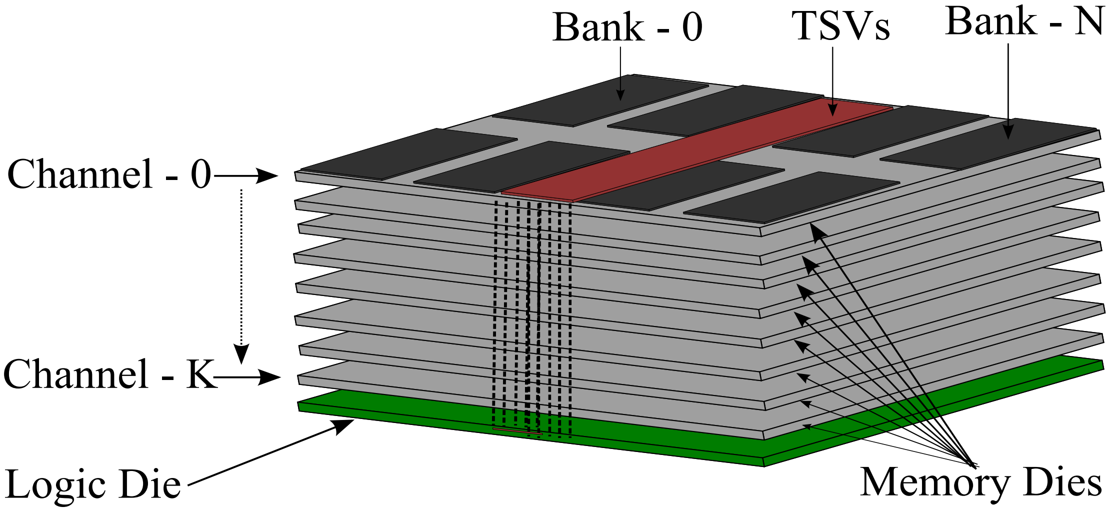
- Stack DRAM Dies over each other → 3D DRAM
- Use Through Silicon Vias (TSV) to connect Dies
- Higher density of TSV → Higher Bandwidth

Go 3D to Scale Bandwidth Wall

Courtesy MICRON, Extremetech

FAILURES IN 3D DRAM

- 3D DRAM ➡ Communicate using TSVs

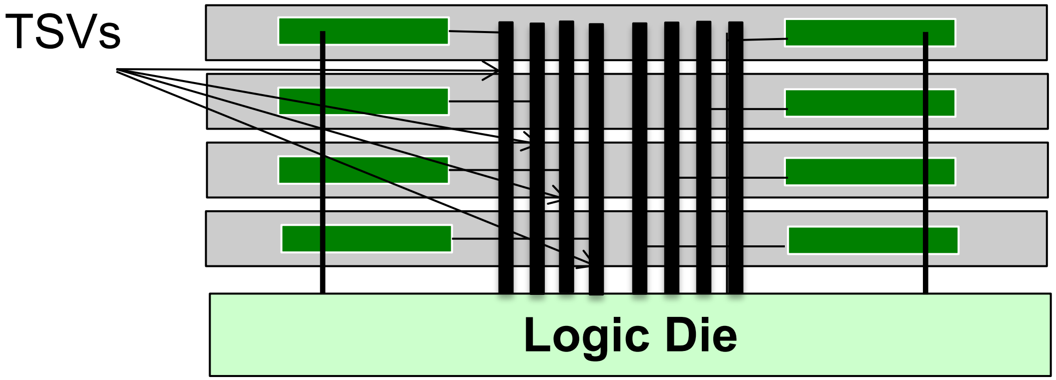


- A New Failure Mode: TSV Failures
- TSV Failures ➡ Large Granularity Failures

TSVs Present New Kind of Large Granularity Failures

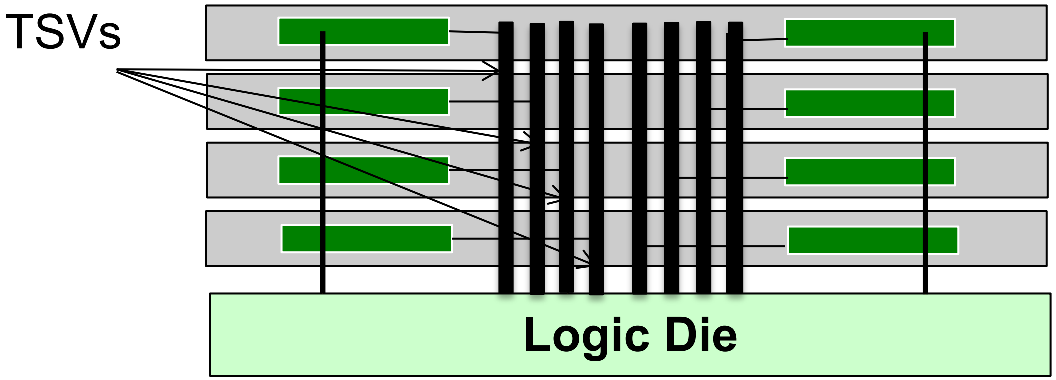
A NEW FAILURE MODE FROM TSVs

TSVs conduit for Address and Data



A NEW FAILURE MODE FROM TSVs

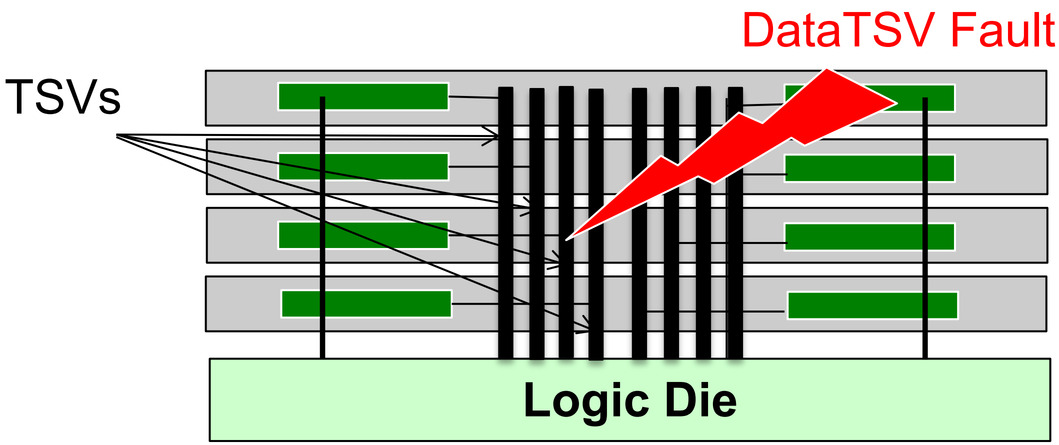
TSVs conduit for Address and Data



- Mainly Two Types TSV Faults

A NEW FAILURE MODE FROM TSVs

TSVs conduit for Address and Data

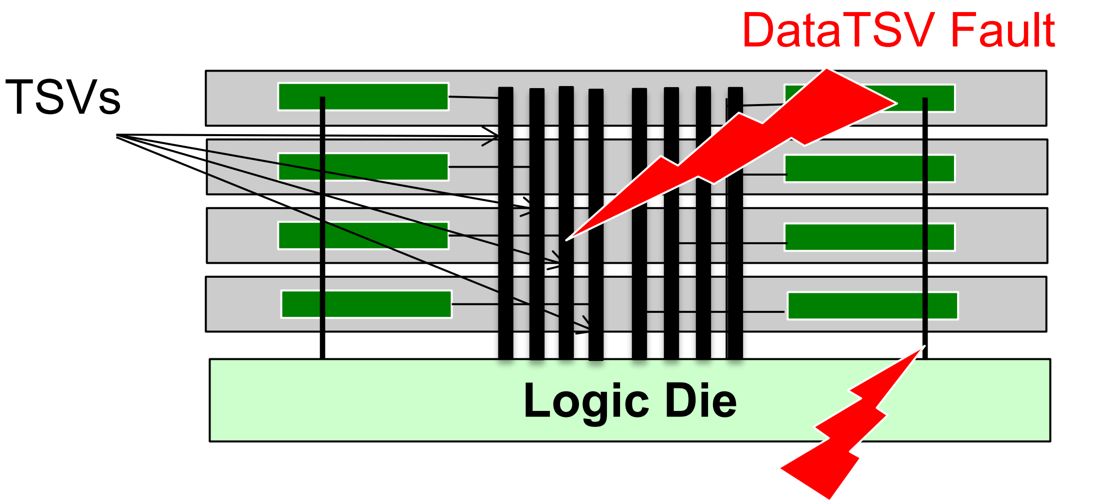


- Mainly Two Types TSV Faults

– Data (Incorrect Data fetched from DRAM Die)

A NEW FAILURE MODE FROM TSVs

TSVs conduit for Address and Data



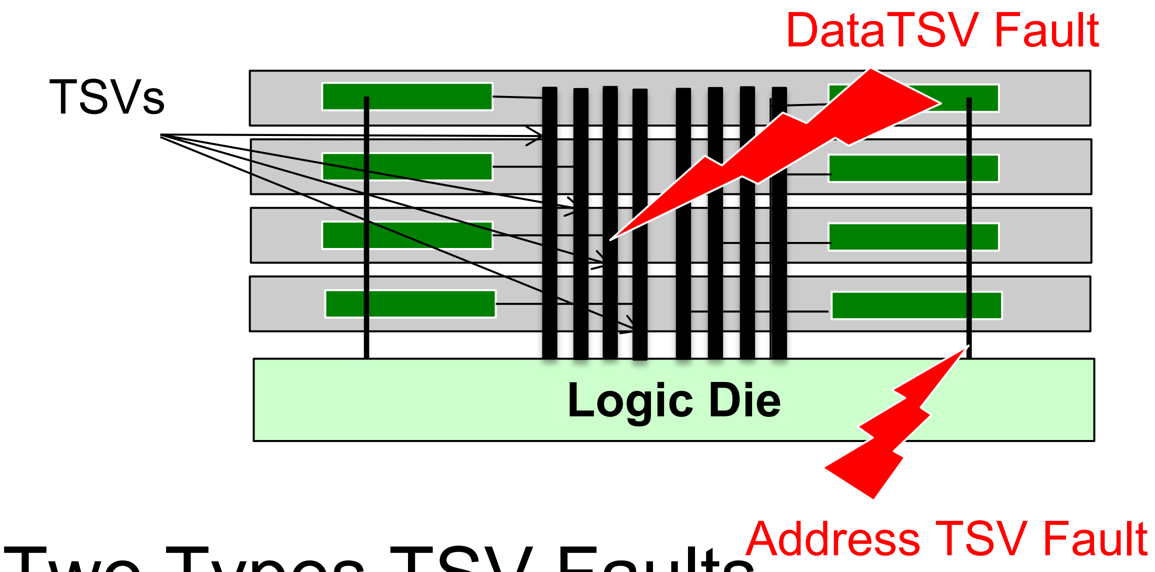
- Mainly Two Types TSV Faults

– Data (Incorrect Data fetched from DRAM Die)

– Address (Incorrect address presented to DRAM Die)

A NEW FAILURE MODE FROM TSVs

TSVs conduit for Address and Data



- Mainly Two Types TSV Faults

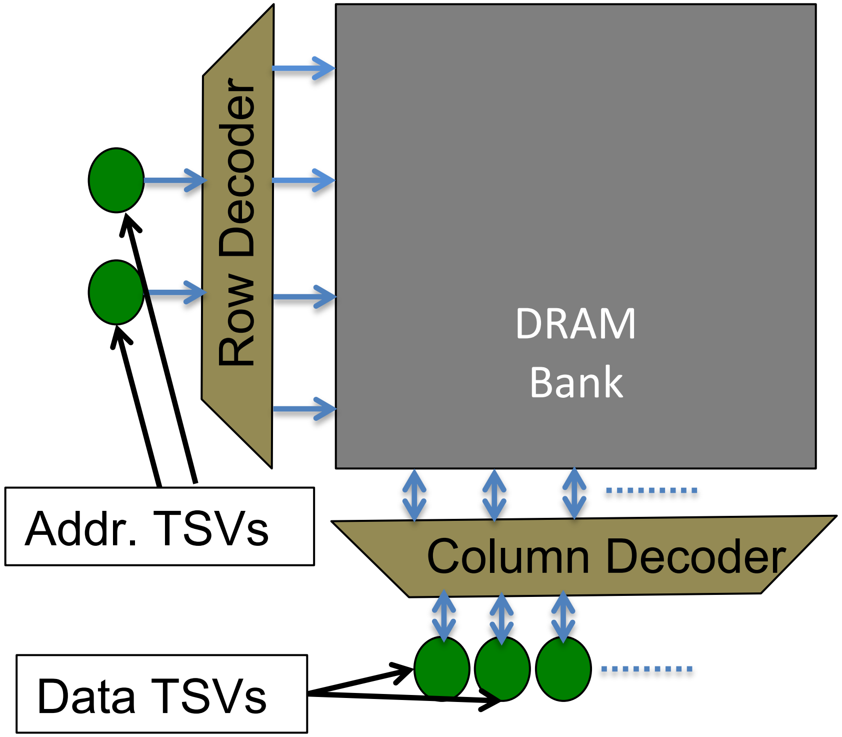
– Data (Incorrect Data fetched from DRAM Die)

– Address (Incorrect address presented to DRAM Die)

TSV Faults cause unavailability of Data and Addresses

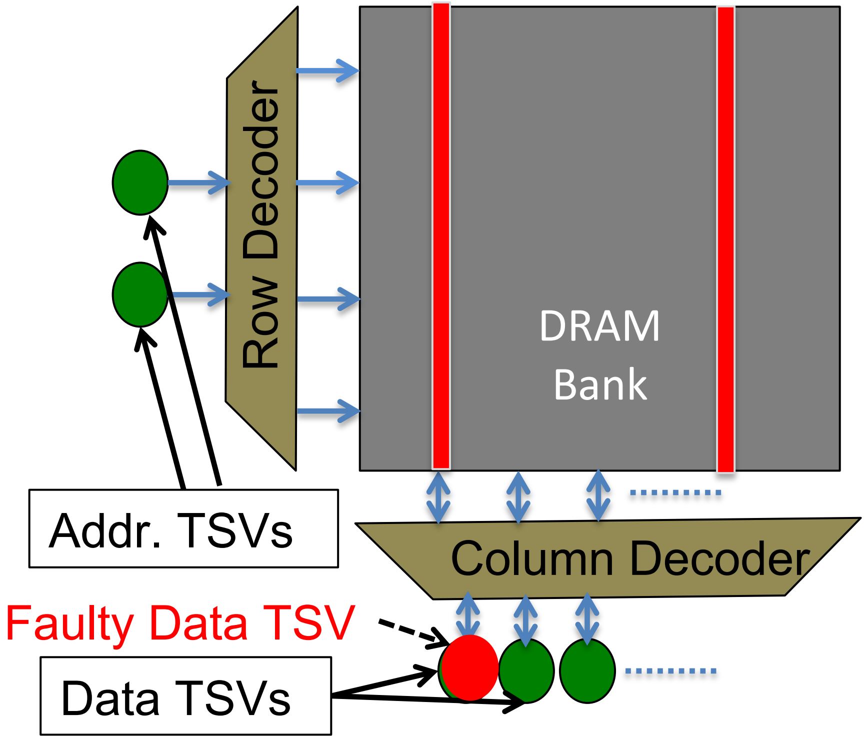
EFFECT OF TSV FAULTS

- Data TSV Fault ➡ Few Columns Faulty



EFFECT OF TSV FAULTS

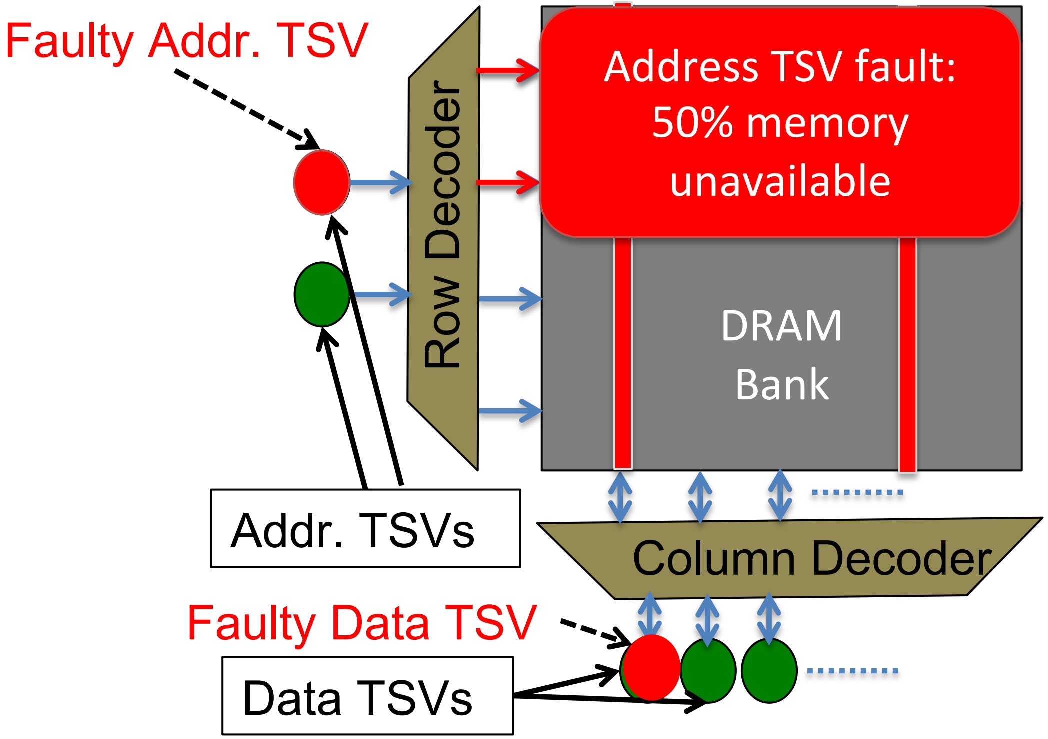
- Data TSV Fault ➡ Few Columns Faulty



EFFECT OF TSV FAULTS

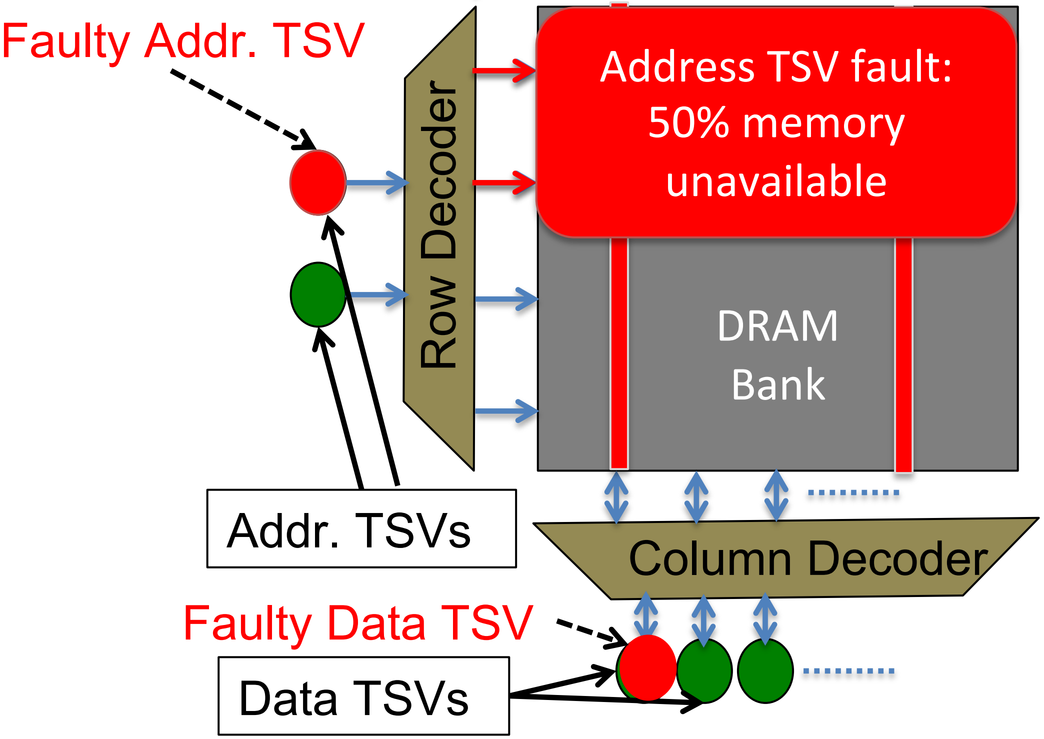
- Data TSV Fault ➡ Few Columns Faulty

- Address TSV Fault ➡ 50% Memory Loss



EFFECT OF TSV FAULTS

- Data TSV Fault ➡ Few Columns Faulty
- Address TSV Fault ➡ 50% Memory Loss

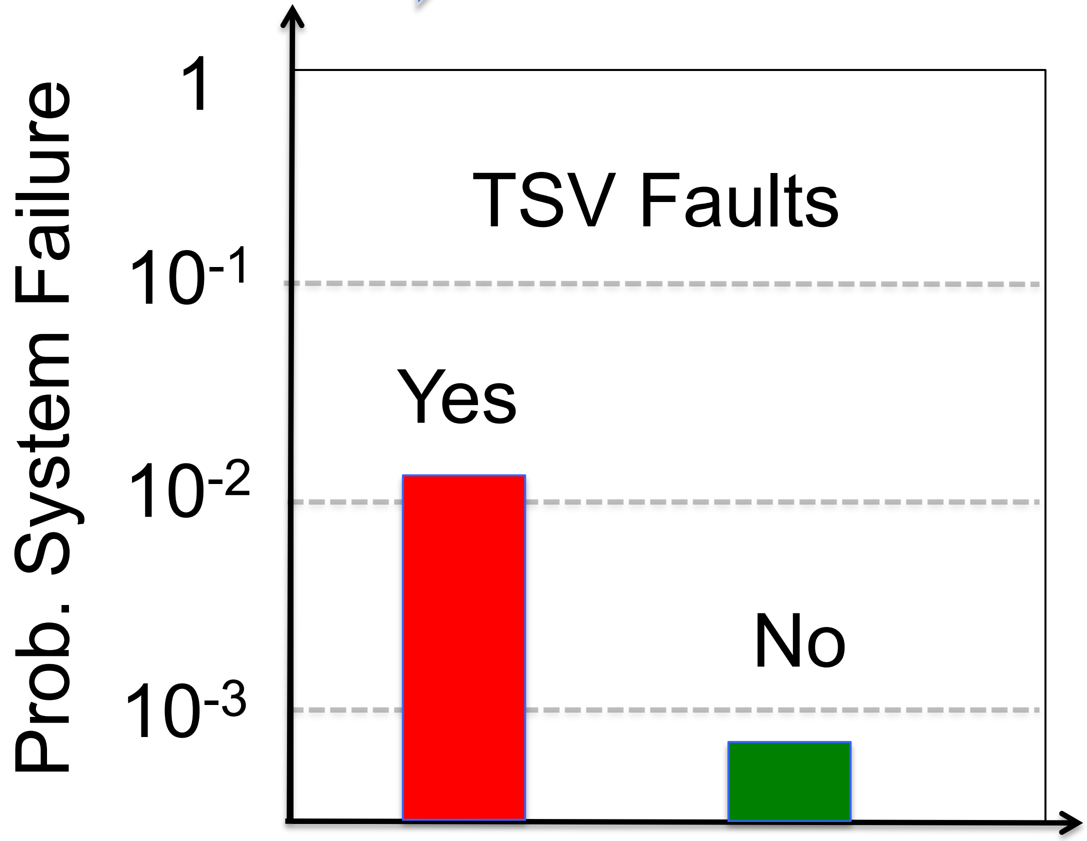


TSVs can cause failures at multiple granularities

IMPACT OF TSV FAULTS

System: 8GB Stacked Memory (HBM)

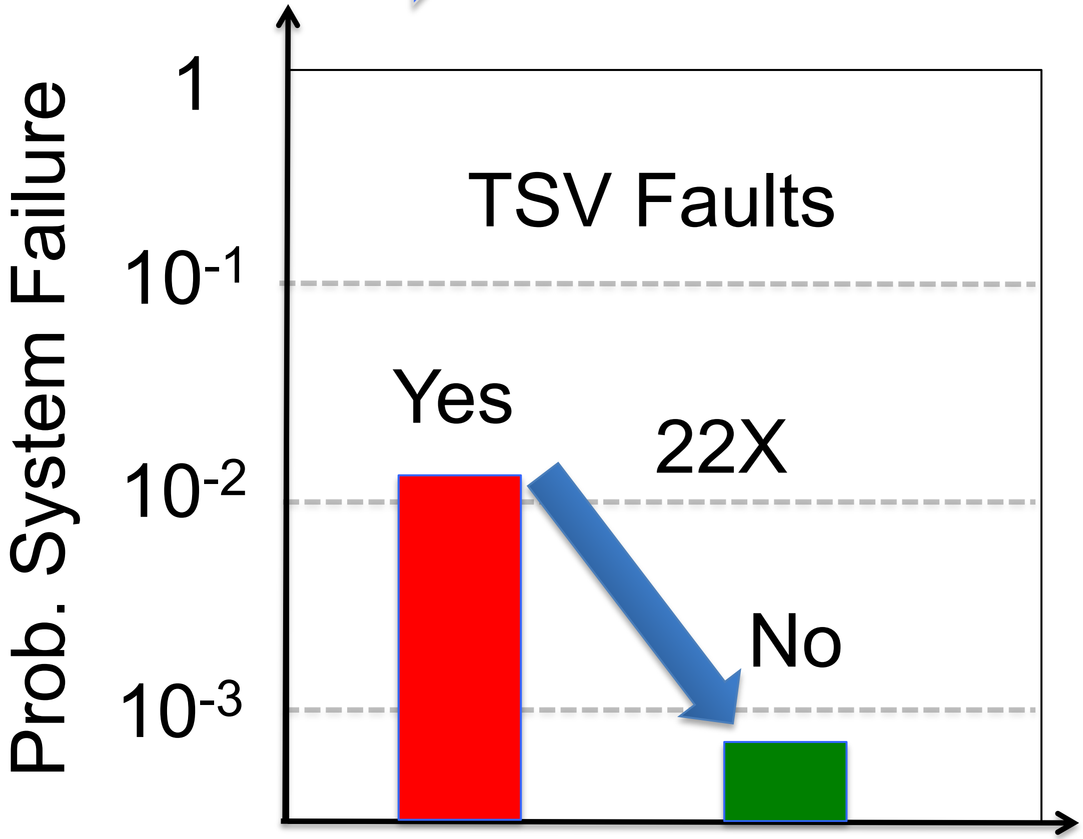
Prob. System Failure ➡ Prob(Uncorrectable Error)



IMPACT OF TSV FAULTS

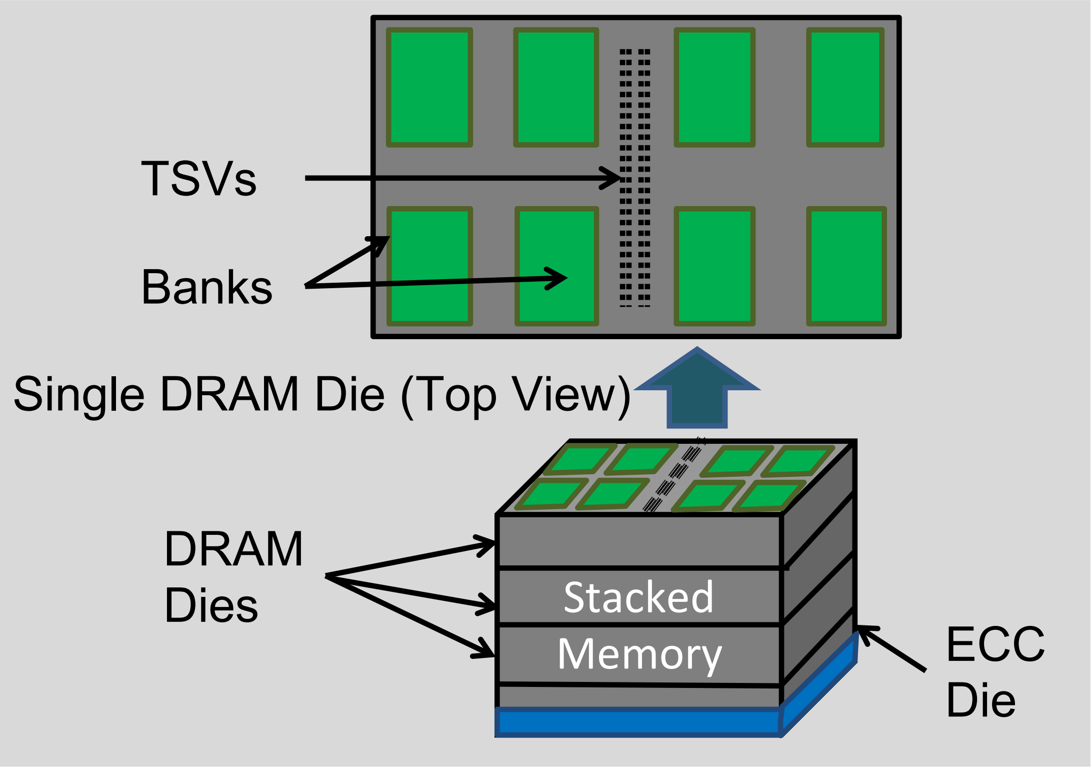
System: 8GB Stacked Memory (HBM)

Prob. System Failure → Prob(Uncorrectable Error)



Efficient Techniques to Mitigate TSV Faults

OTHER FAILURES STILL PRESENT



TSVs

Banks

Single DRAM Die (Top View)

DRAM Dies

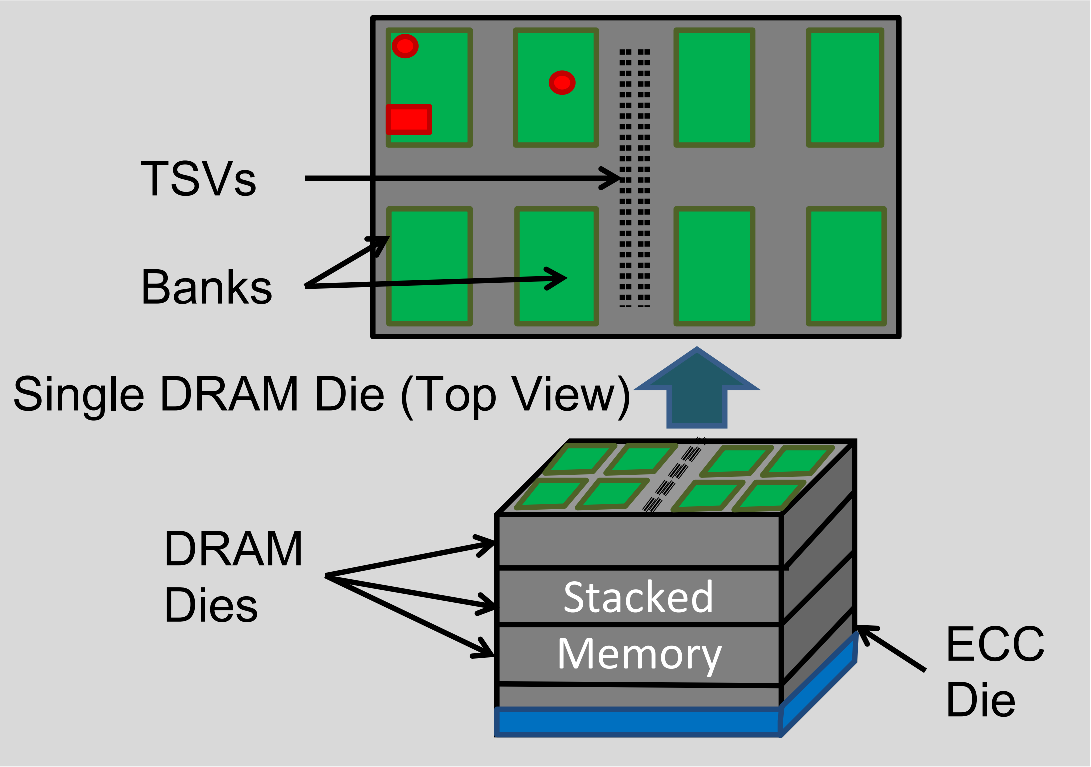
Stacked Memory

ECC Die

Apart from TSV Faults, 3D DRAM will also continue to have other multi-granularity failures

OTHER FAILURES STILL PRESENT

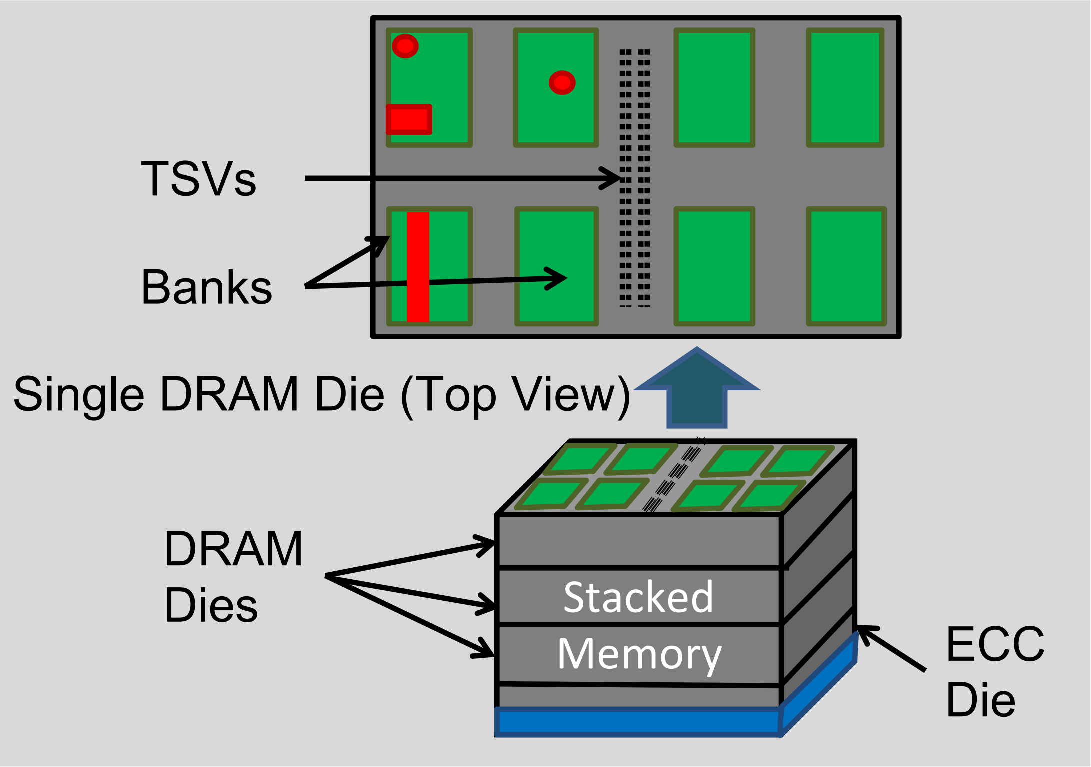
- Bit
- Word



Apart from TSV Faults, 3D DRAM will also continue to have other multi-granularity failures

OTHER FAILURES STILL PRESENT

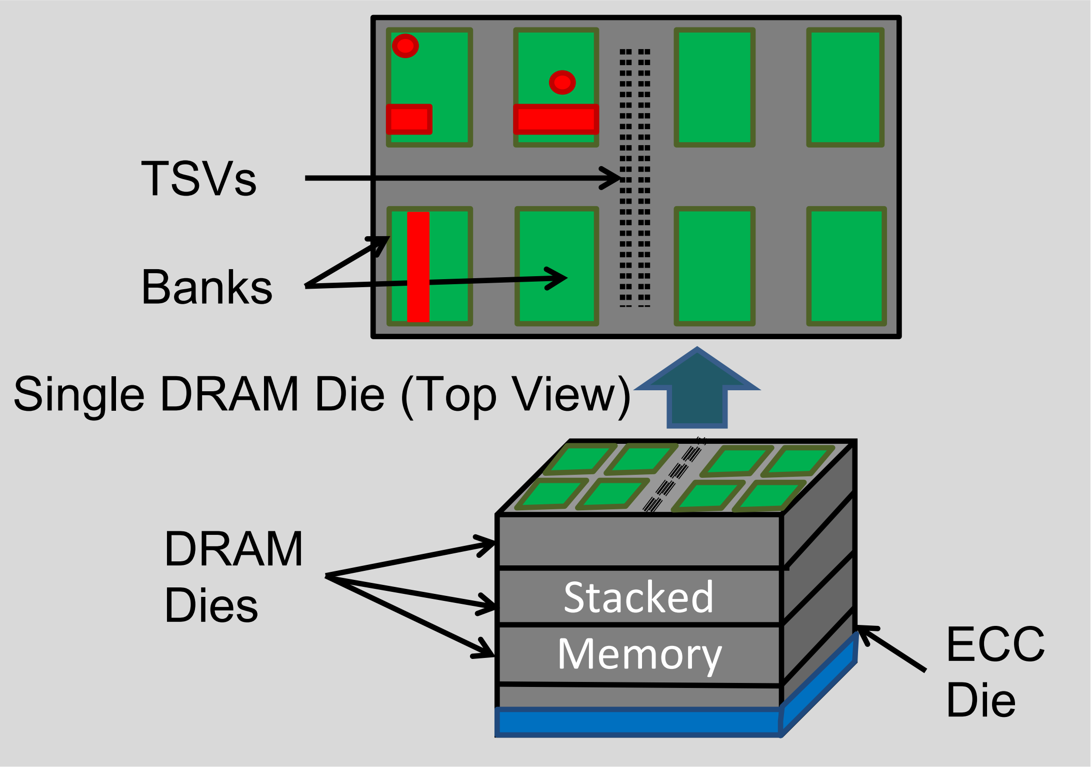
- Bit
- Word
- Column



Apart from TSV Faults, 3D DRAM will also continue to have other multi-granularity failures

OTHER FAILURES STILL PRESENT

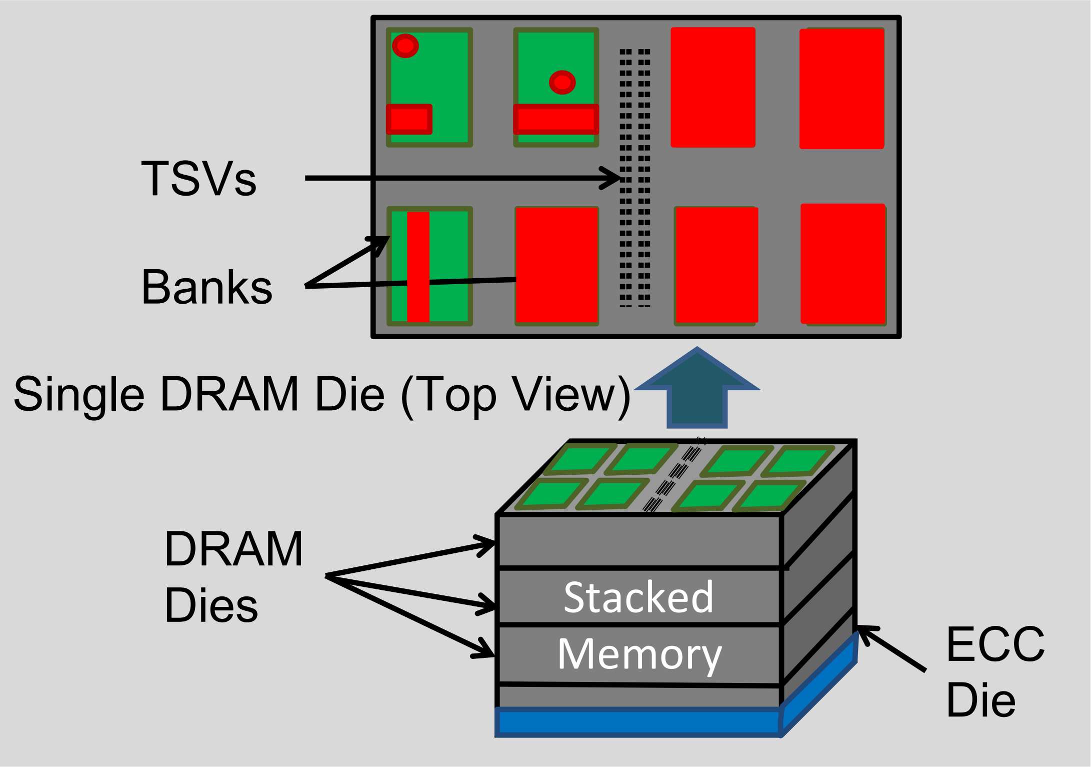
- Bit
- Word
- Column
- Row



Apart from TSV Faults, 3D DRAM will also continue to have other multi-granularity failures

OTHER FAILURES STILL PRESENT

- Bit
- Word
- Column
- Row
- Bank



Apart from TSV Faults, 3D DRAM will also continue to have other multi-granularity failures

3D DRAM: FAILURE RATE

Die Failure Mode	*Permanent Fault Rate (FIT)
Bit	148.8
Word	2.4
Column	10.5
Row	32.8
Bank	80

*Projected from Sridharan et. al. : DRAM Field Study

3D DRAM: FAILURE RATE

Die Failure Mode	*Permanent Fault Rate (FIT)
Bit	148.8
Word	2.4
Column	10.5
Row	32.8
Bank	80

= 125.7

*Projected from Sridharan et. al. : DRAM Field Study

3D DRAM: FAILURE RATE

Die Failure Mode	*Permanent Fault Rate (FIT)
Bit	148.8
Word	2.4
Column	10.5
Row	32.8
Bank	80

148.8 ✓ *SECDED*

$$\left.\begin{array}{l}2.4\\10.5\\32.8\\80\end{array}\right\} = \boxed{125.7}$$

✗ *SECDED*

*Projected from Sridharan et. al. : DRAM Field Study

3D DRAM: FAILURE RATE

Die Failure Mode	*Permanent Fault Rate (FIT)
Bit	148.8
Word	2.4
Column	10.5
Row	32.8
Bank	80

Bit 148.8: ✓ SECDED

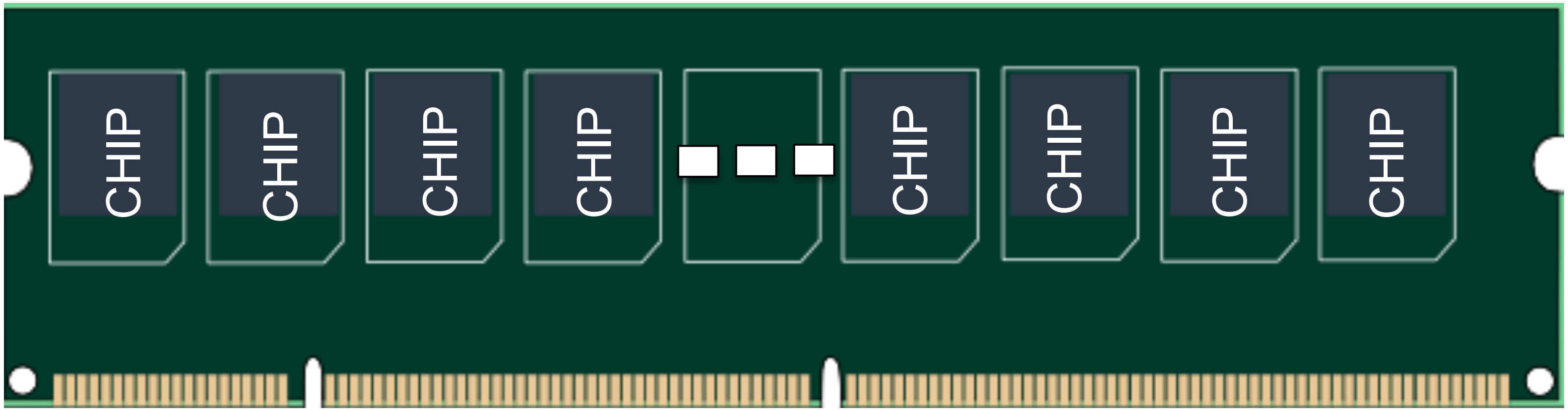
Word + Column + Row + Bank = 125.7: ✗ SECDED

1. Large Granularity Faults are as likely as Bit Faults
2. Low Cost Solutions Required For Large Faults

*Projected from Sridharan et. al. : DRAM Field Study

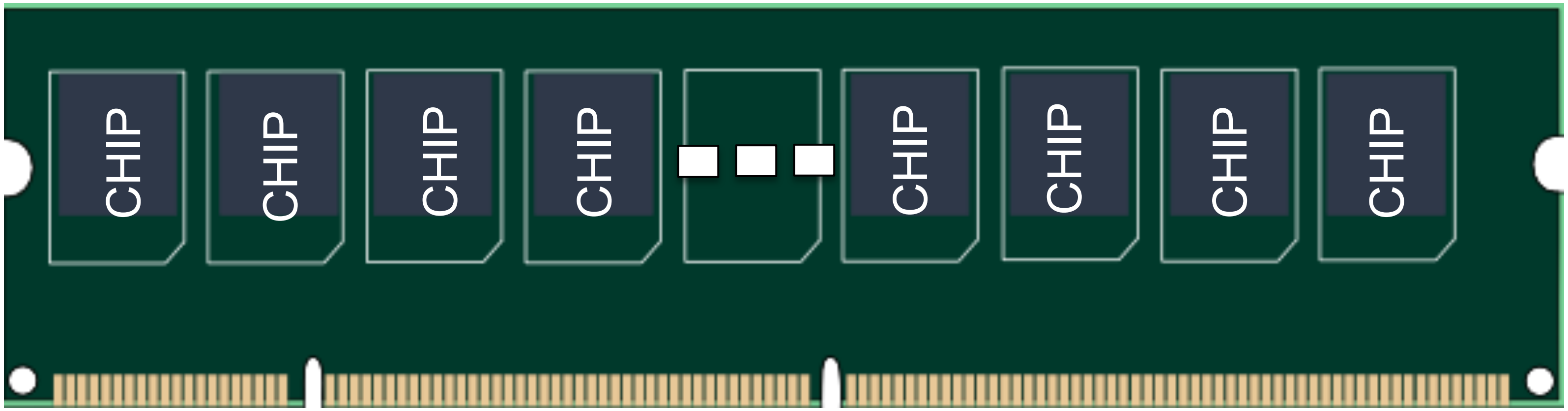
CONVENTIONAL SCHEMES

Current Systems Naturally Stripe Data Across Chips



CONVENTIONAL SCHEMES

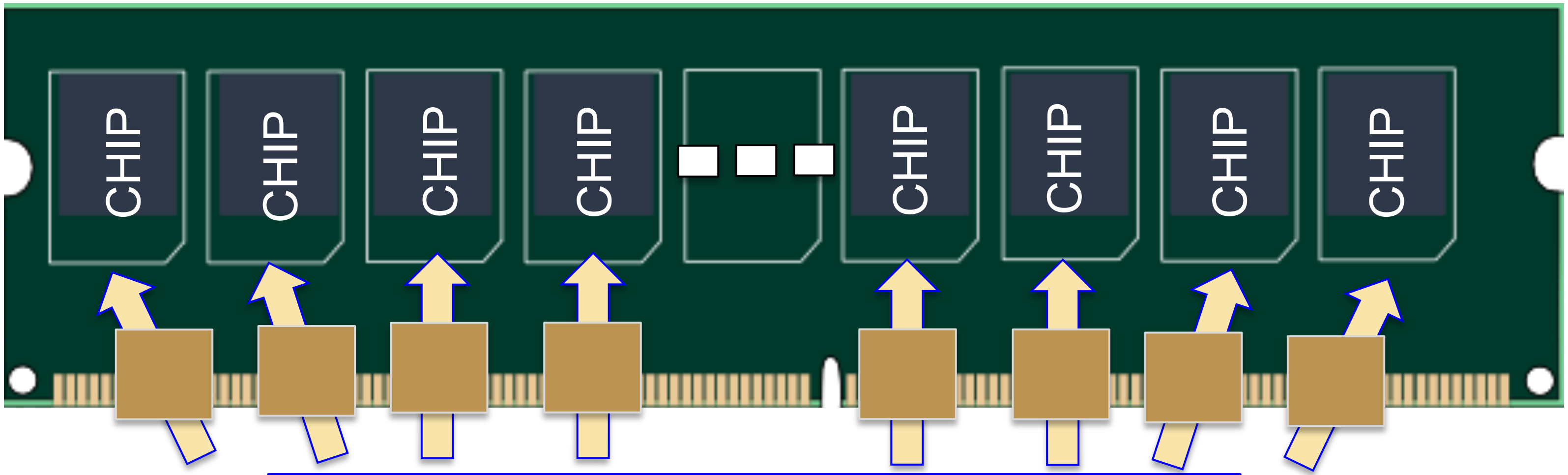
Current Systems Naturally Stripe Data Across Chips



Cache Line = 64 Bytes

CONVENTIONAL SCHEMES

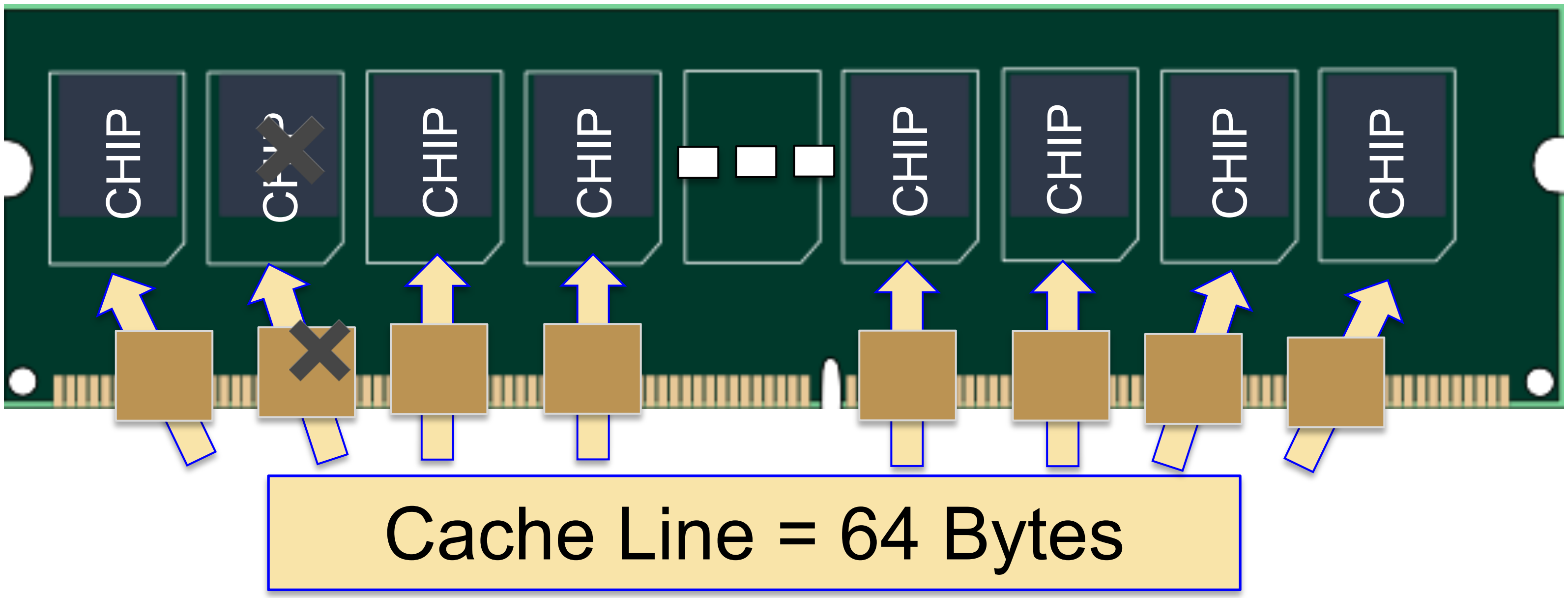
Current Systems Naturally Stripe Data Across Chips



Cache Line = 64 Bytes

CONVENTIONAL SCHEMES

Current Systems Naturally Stripe Data Across Chips

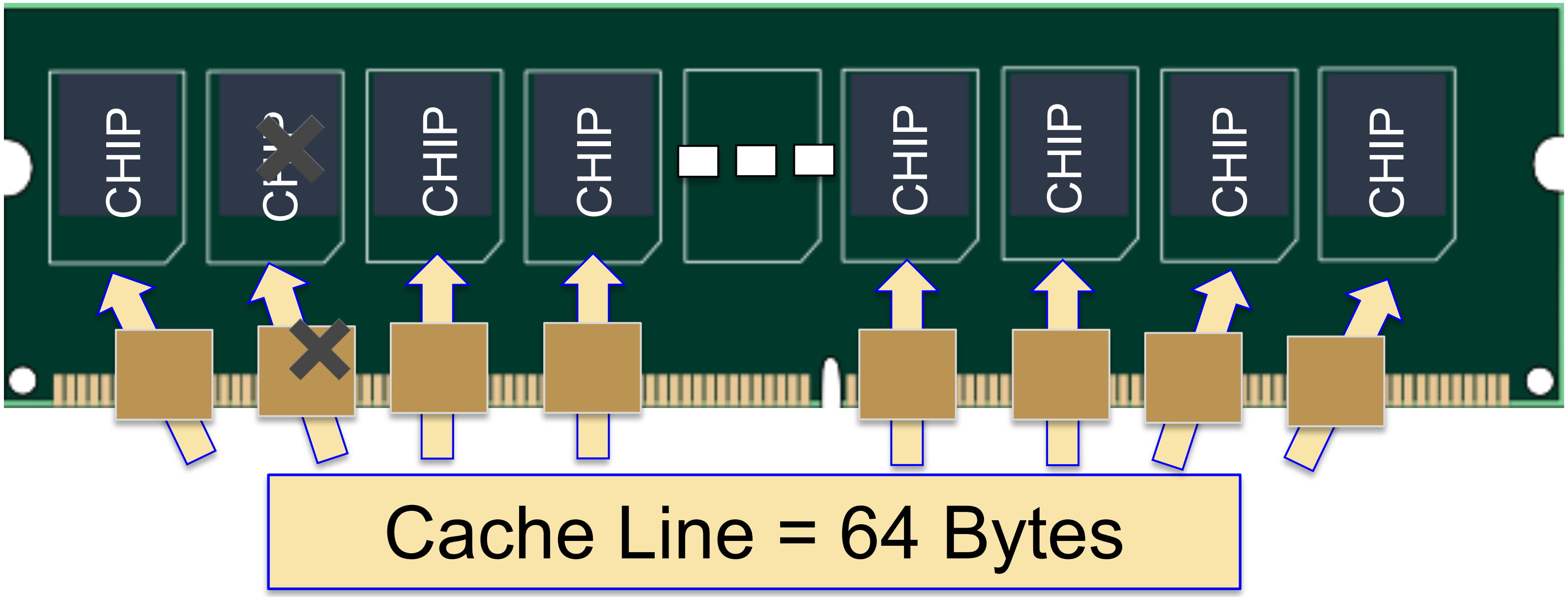


Cache Line = 64 Bytes

- ChipKill : Mitigate Large Failures (Whole Chip)

CONVENTIONAL SCHEMES

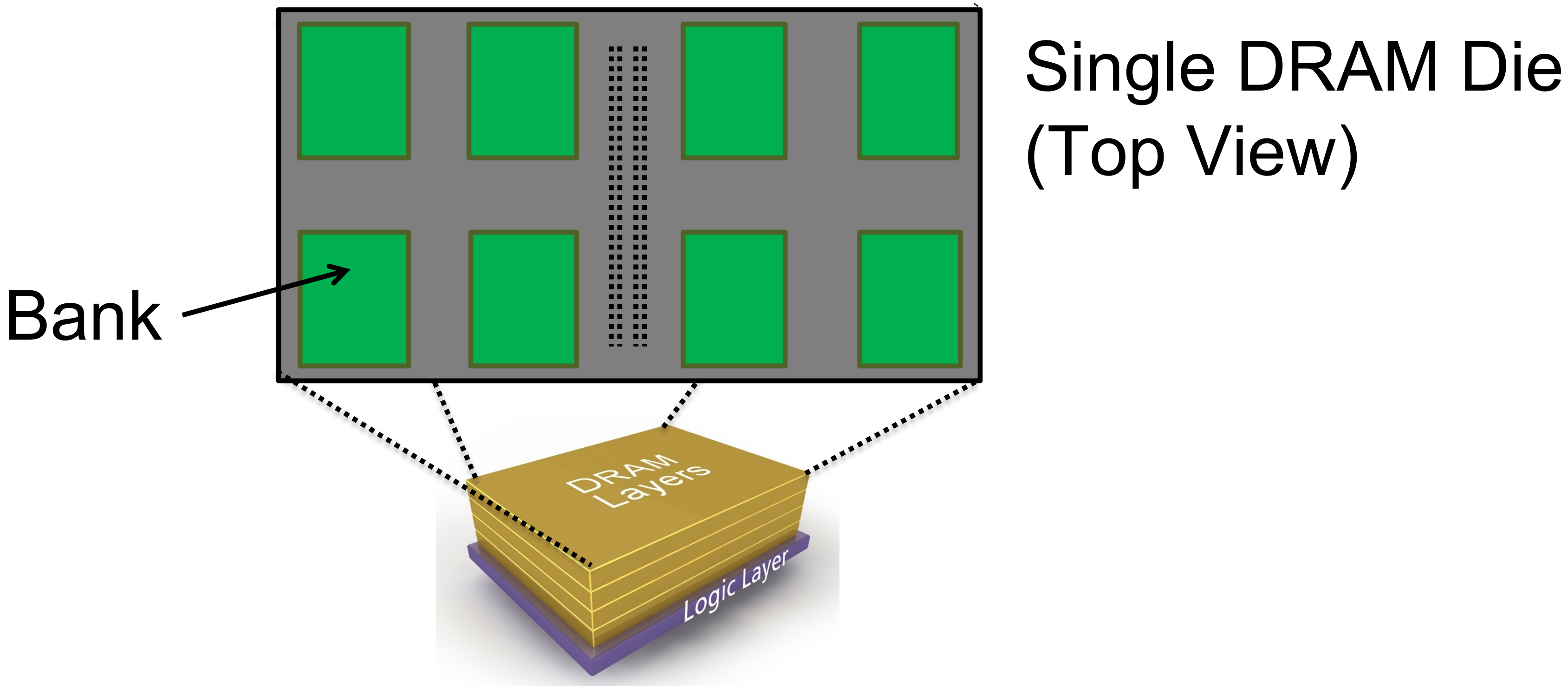
Current Systems Naturally Stripe Data Across Chips



- ChipKill : Mitigate Large Failures (Whole Chip)

ChipKill relies on data striping to tolerate large granularity failures

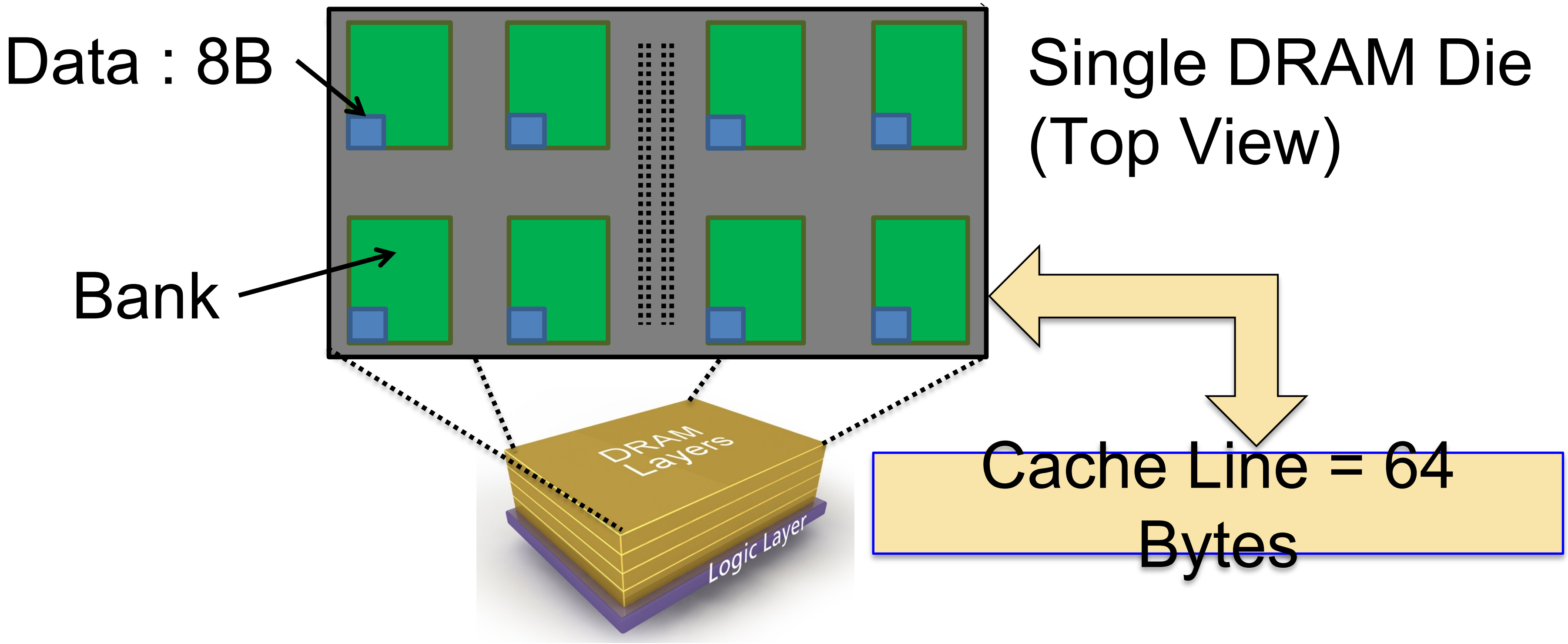
CHIPKILL IN STACKED MEMORY



Single DRAM Die (Top View)

Bank

CHIPKILL IN STACKED MEMORY



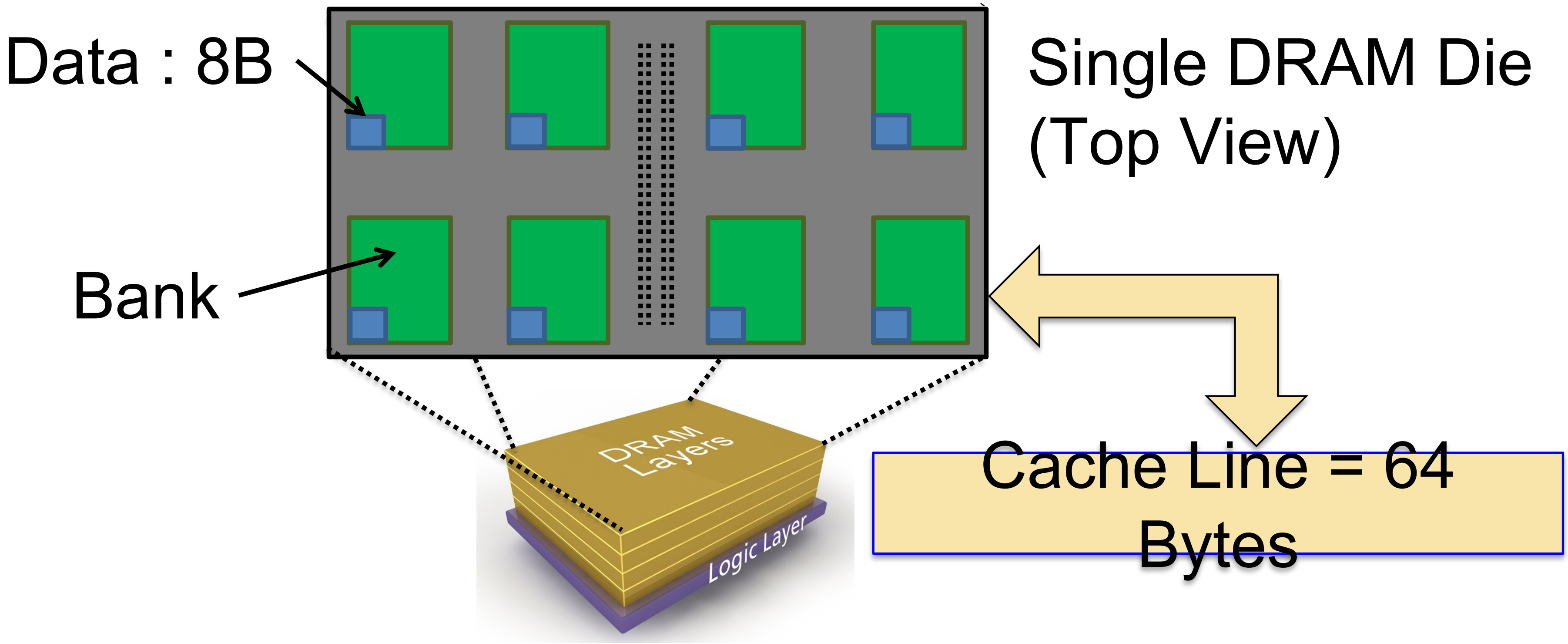
Data : 8B

Bank

Single DRAM Die
(Top View)

Cache Line = 64 Bytes

CHIPKILL IN STACKED MEMORY



Data : 8B

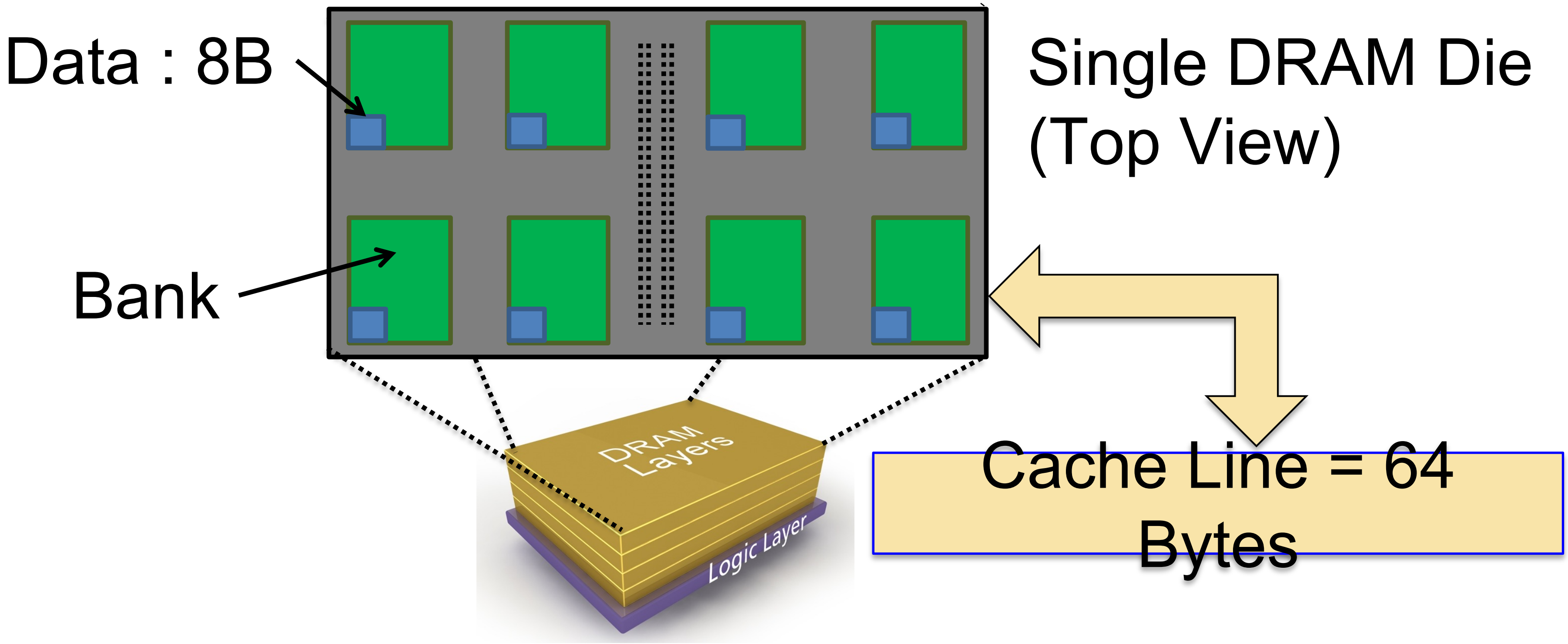
Bank

Single DRAM Die (Top View)

Cache Line = 64 Bytes

- A request activates at least 8 Banks or 8 Channels

CHIPKILL IN STACKED MEMORY



Data : 8B

Bank

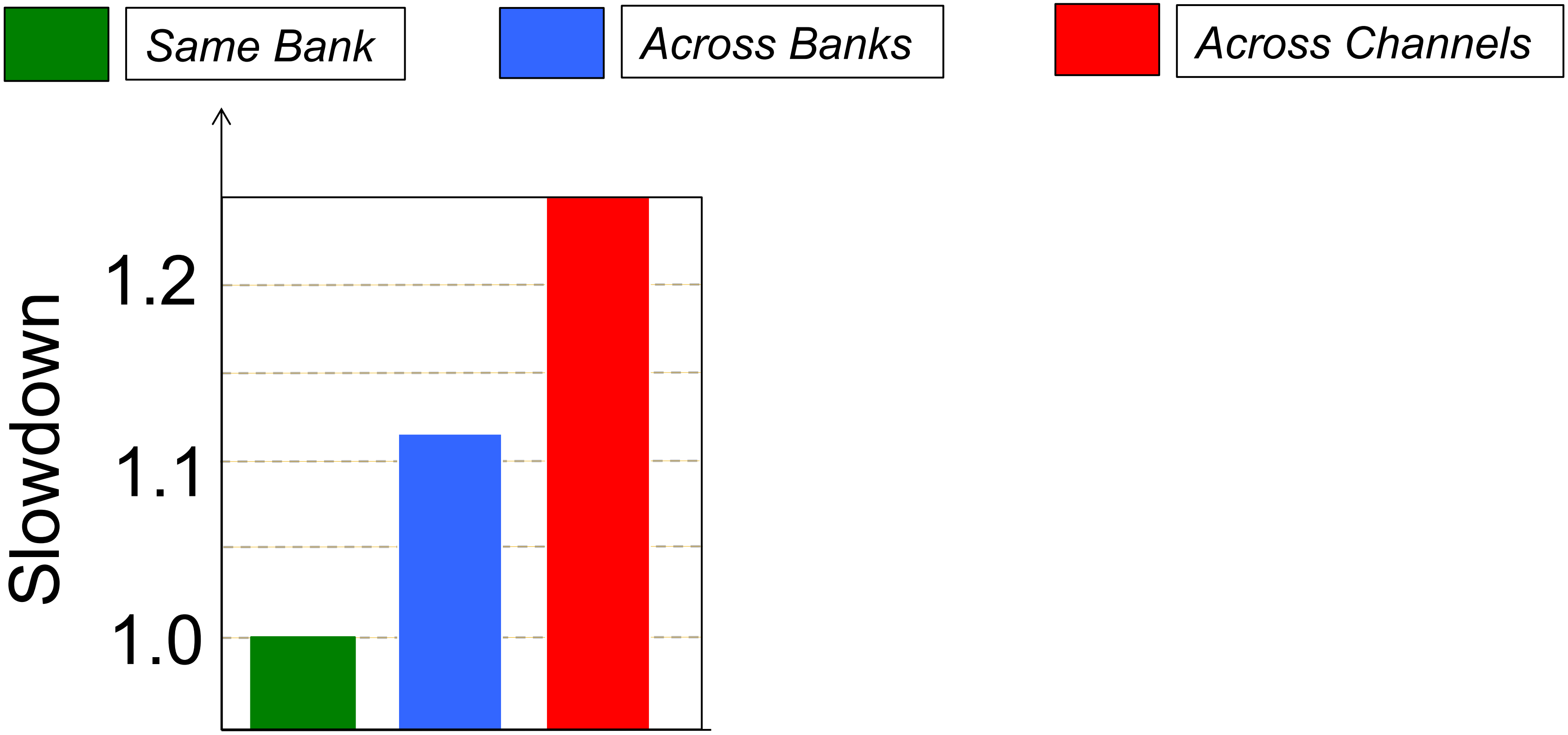
Single DRAM Die
(Top View)

Cache Line = 64 Bytes

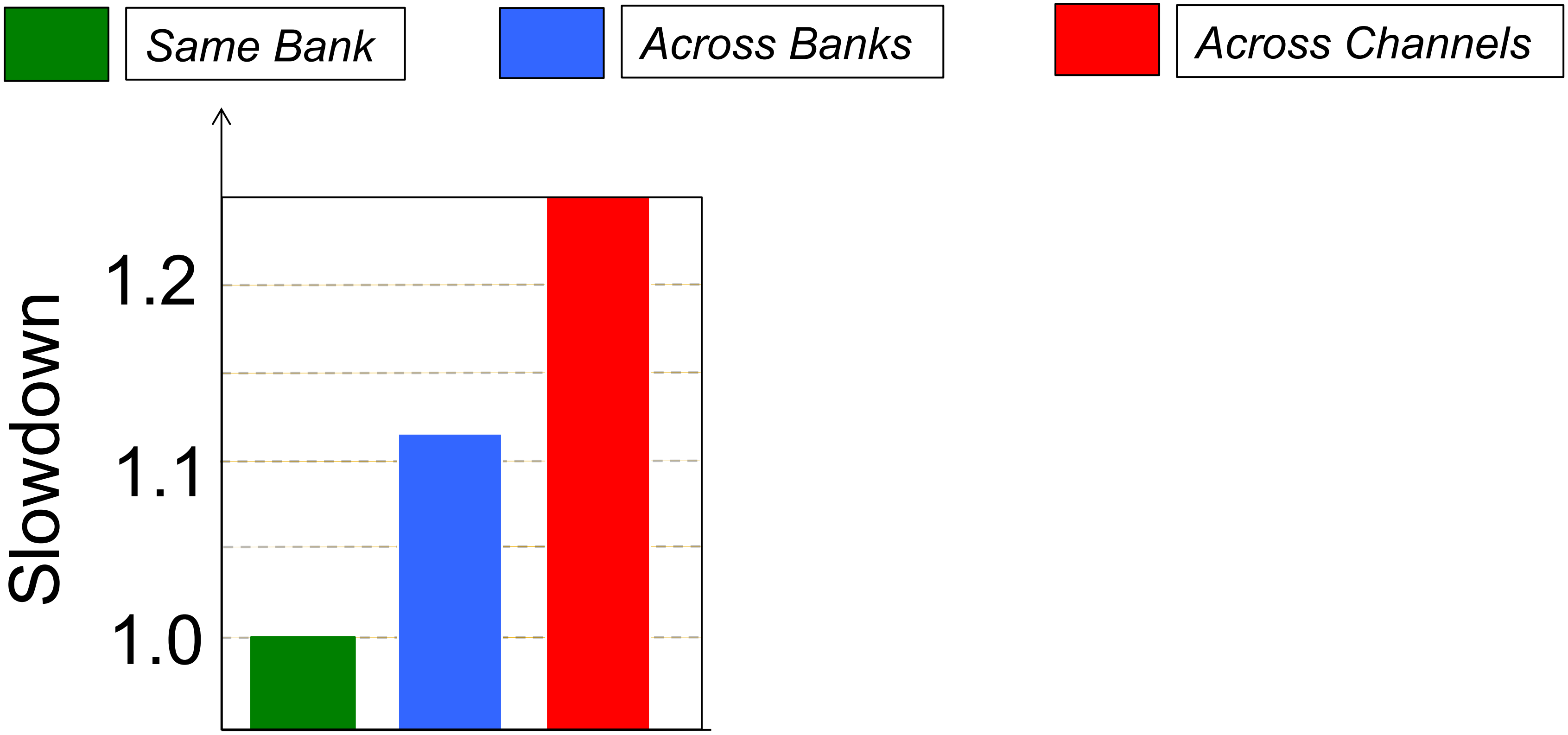
- A request activates at least 8 Banks or 8 Channels

At least 8X↑ activation power, 8X↓ DRAM parallelism

COST OF STRIPING IN 3D DRAM

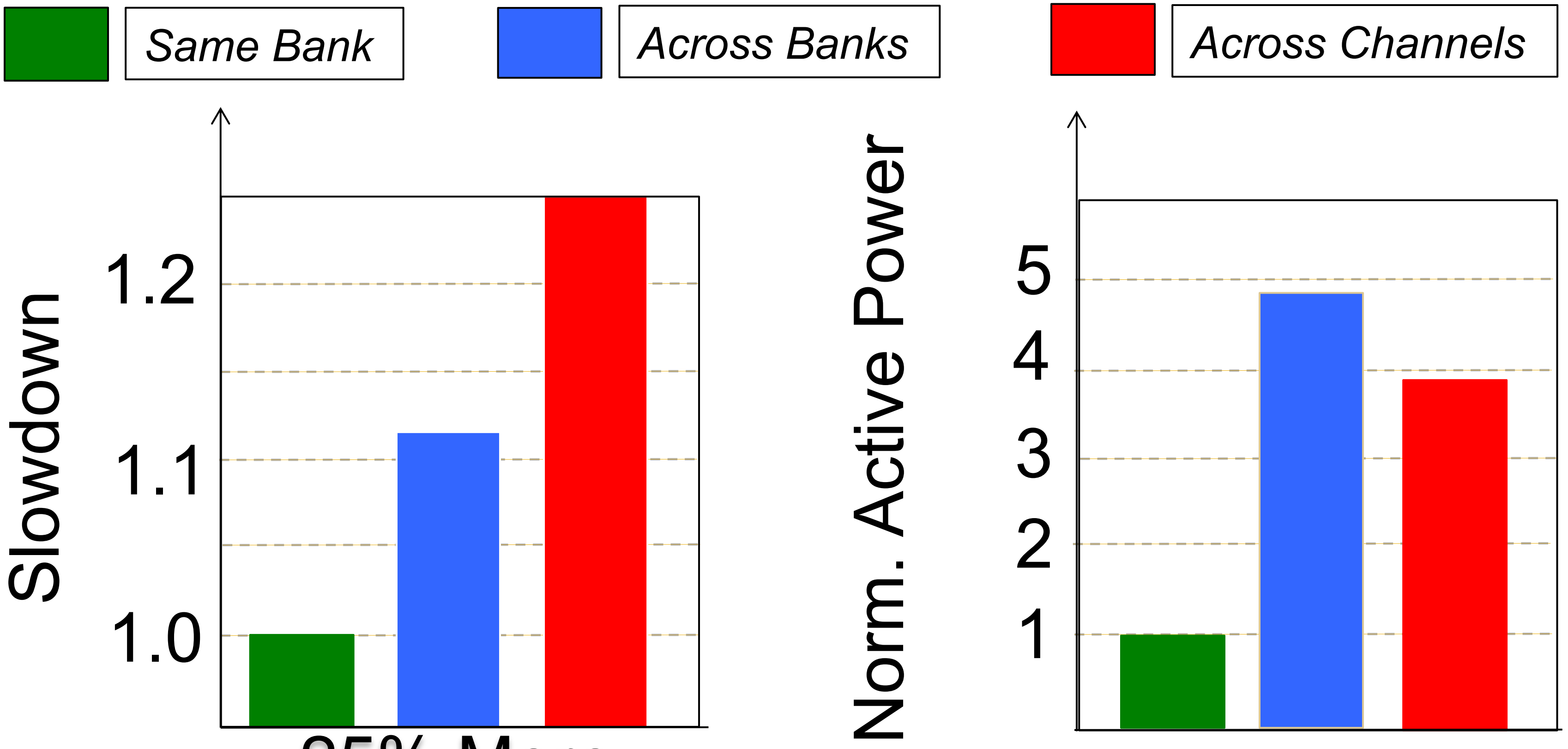


COST OF STRIPING IN 3D DRAM



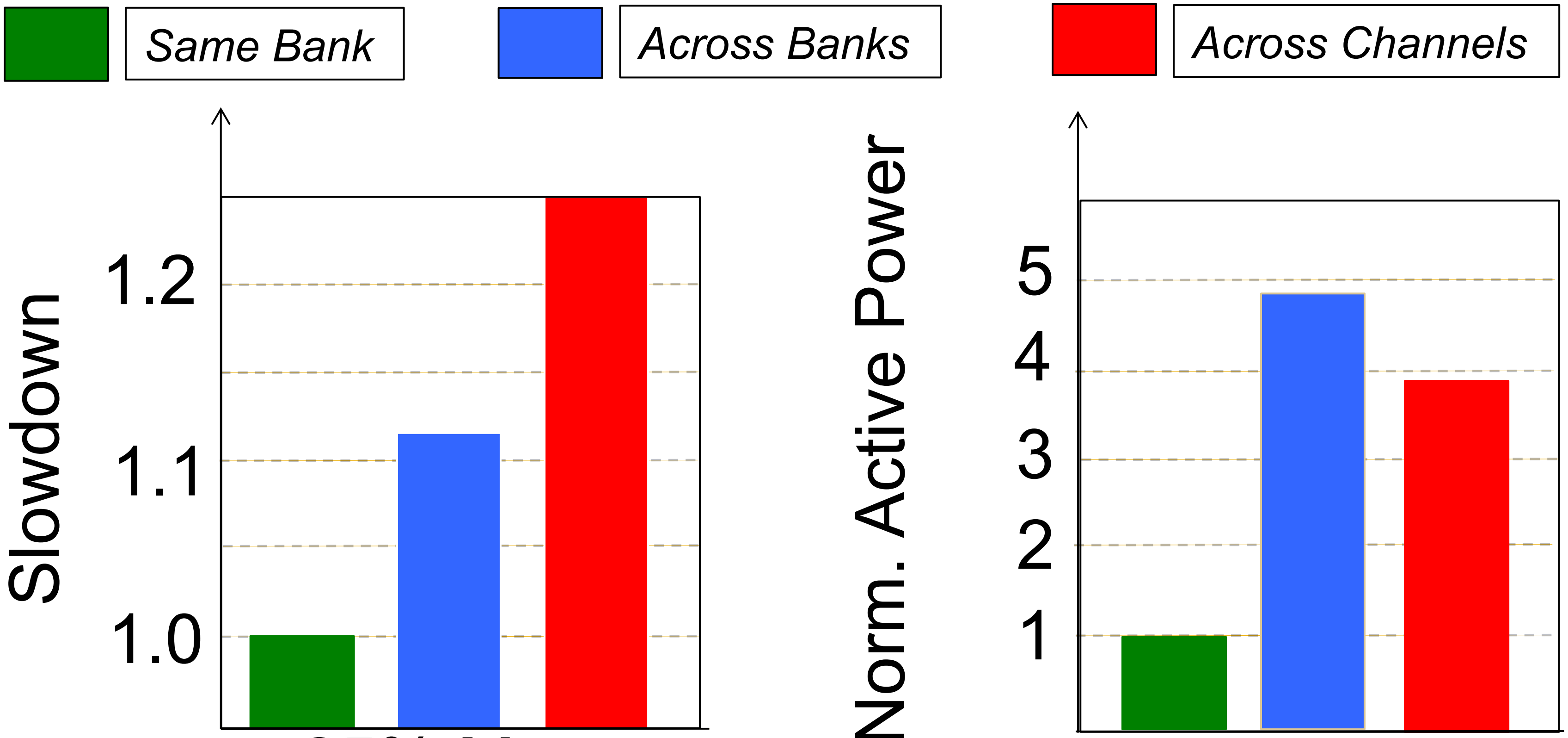
25% More Execution Time

COST OF STRIPING IN 3D DRAM



25% More Execution Time

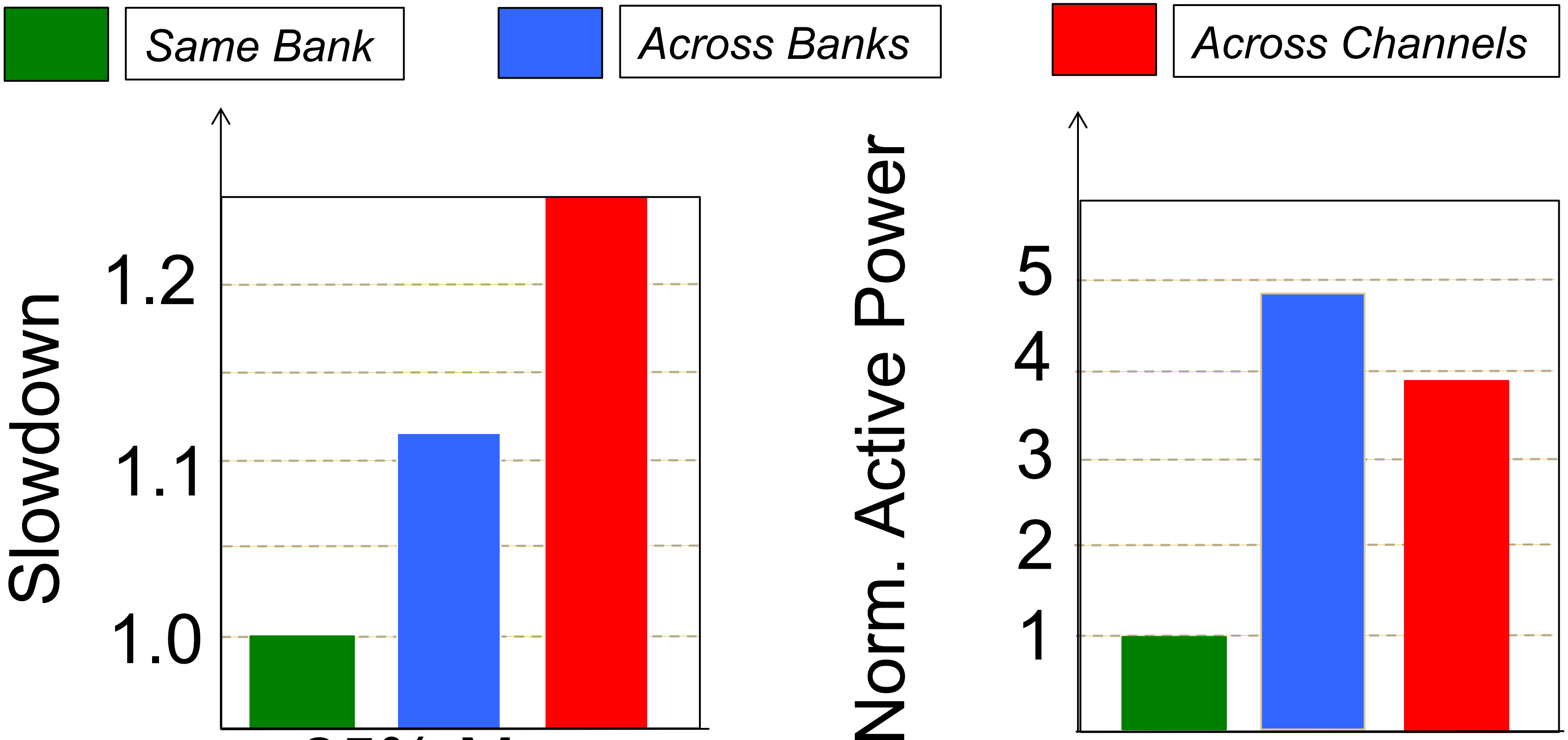
COST OF STRIPING IN 3D DRAM



25% More Execution Time

4.8X More Activation Power

COST OF STRIPING IN 3D DRAM



25% More Execution Time

4.8X More Activation Power

Striping data across banks/channels in 3D is costly

GOAL

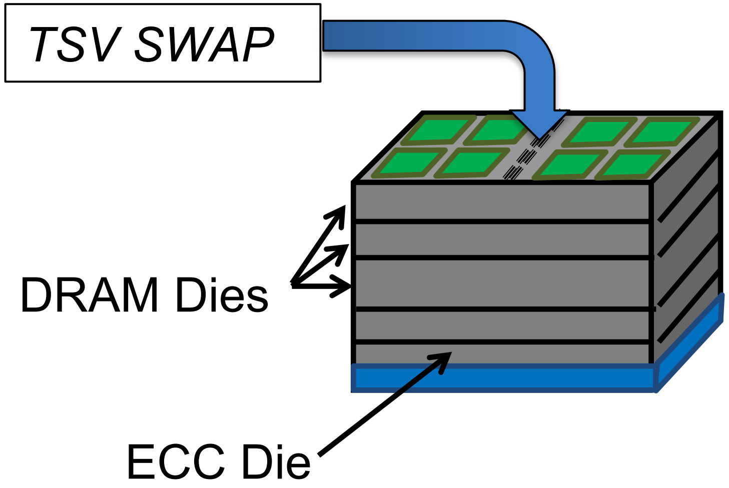
Develop Efficient Solutions to Mitigate TSV and other Large Granularity Faults in Stacked Memory without striping data

OUTLINE

- Introduction and Background
- Citadel ⇦
- Scheme - 1 : TSV-SWAP
- Scheme - 2 : Three Dimensional Parity (3DP)
- Scheme - 3 : Dynamic Dual Grain Sparing (DDS)
- Summary

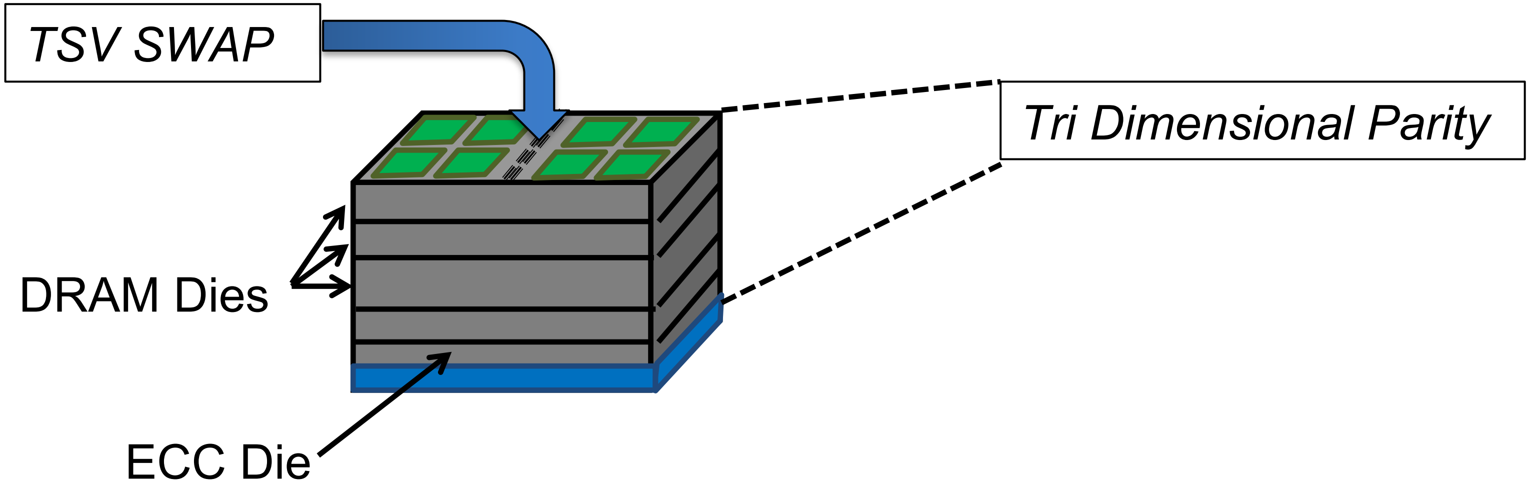
CITADEL: AN OVERVIEW

- Runtime TSV Sparing (TSV-SWAP)



CITADEL: AN OVERVIEW

- Runtime TSV Sparing (TSV-SWAP)
- RAID-5 across 3 dimensions (Tri dimensional parity)

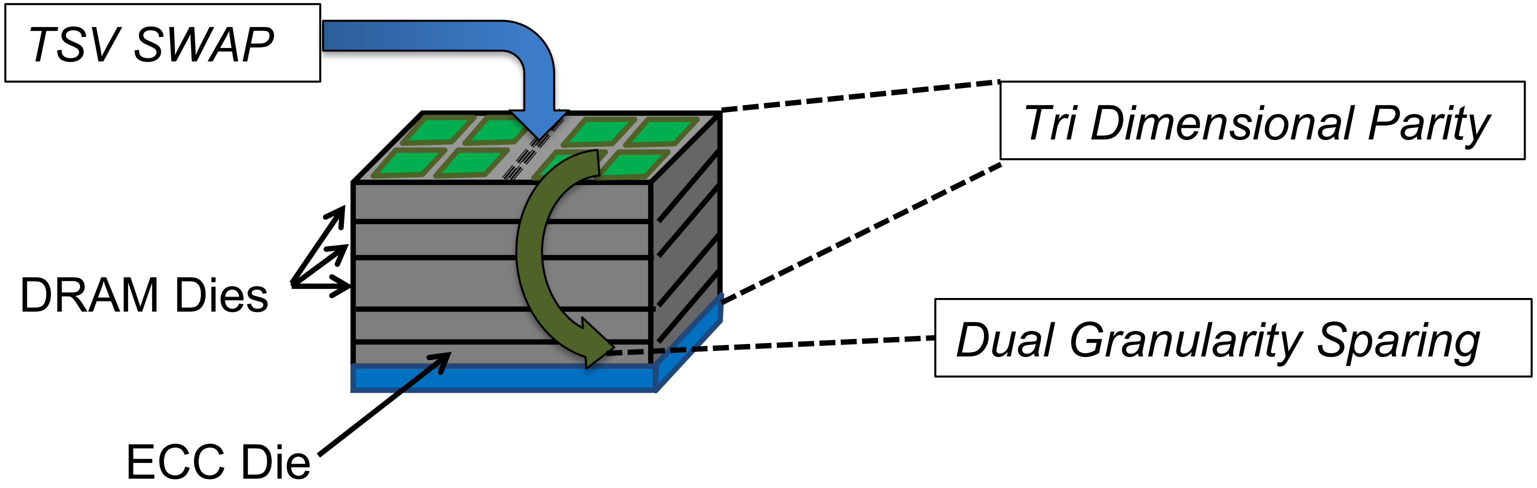


CITADEL: AN OVERVIEW

- Runtime TSV Sparing (TSV-SWAP)

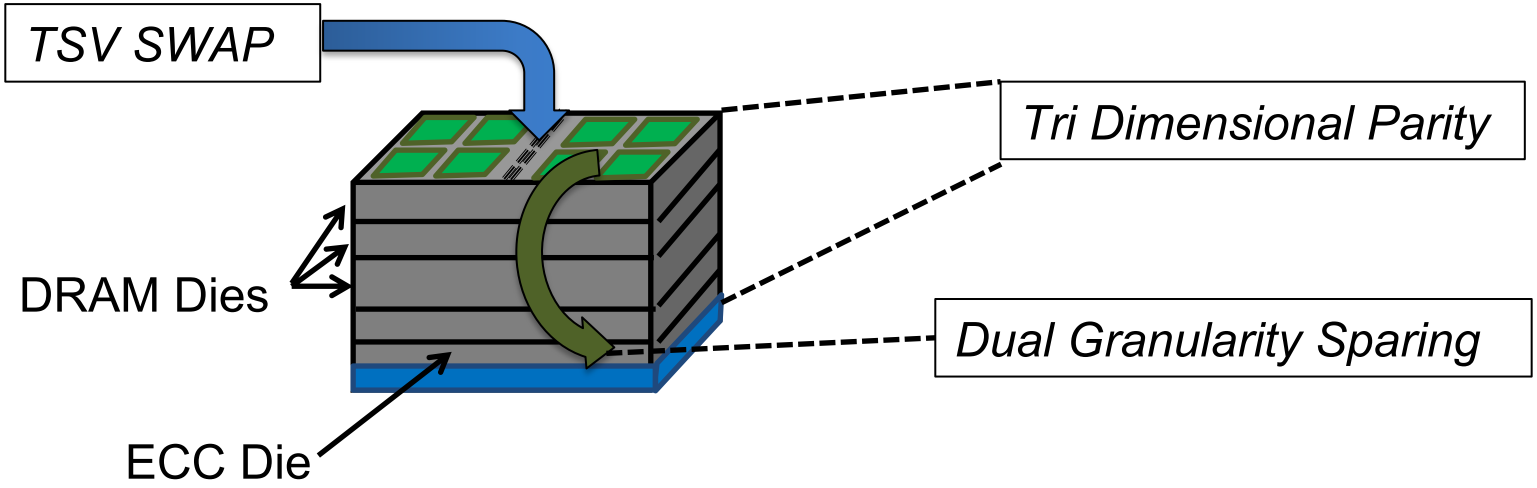
•RAID-5 across 3 dimensions (Tri dimensional parity)

•Spare Faults Regions (Dual Granularity Sparing)



CITADEL: AN OVERVIEW

- Runtime TSV Sparing (TSV-SWAP)
- RAID-5 across 3 dimensions (Tri dimensional parity)
- Spare Faults Regions (Dual Granularity Sparing)



Enable robust stacked memory at very low overheads

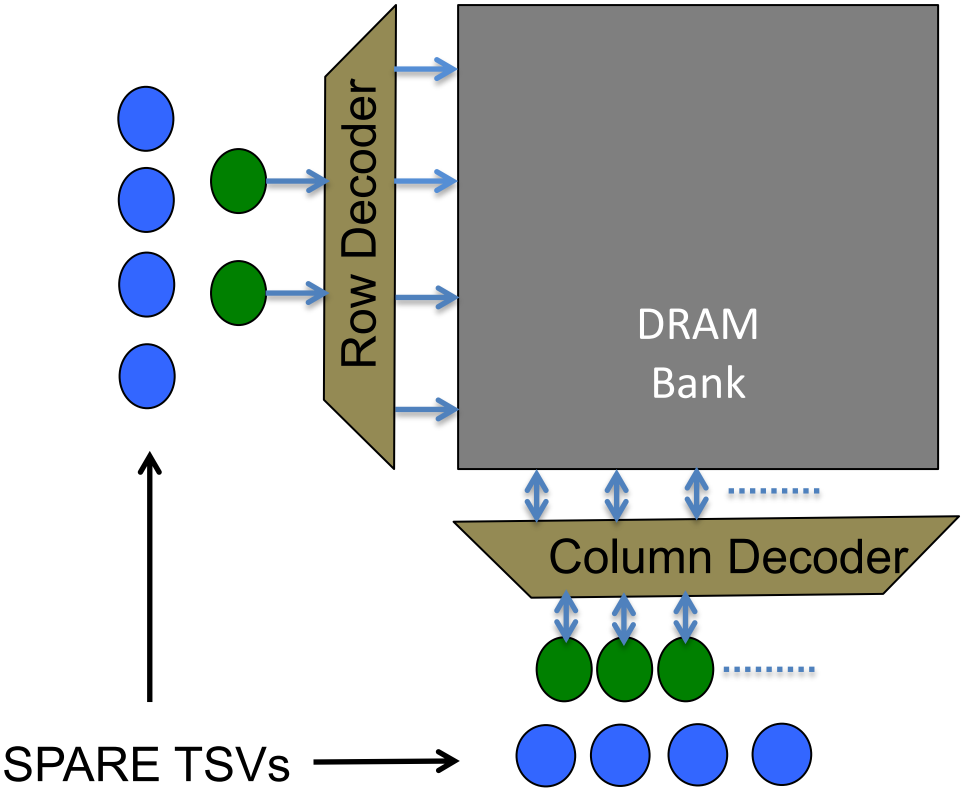
OUTLINE

- Introduction and Background
- Citadel
- Scheme - 1 : TSV-SWAP ⇦
- Scheme - 2 : Three Dimensional Parity (3DP)
- Scheme - 3 : Dynamic Dual Grain Sparing (DDS)
- Summary

DESIGN-TIME TSV SPARING

Designers provision spares TSVs alongside

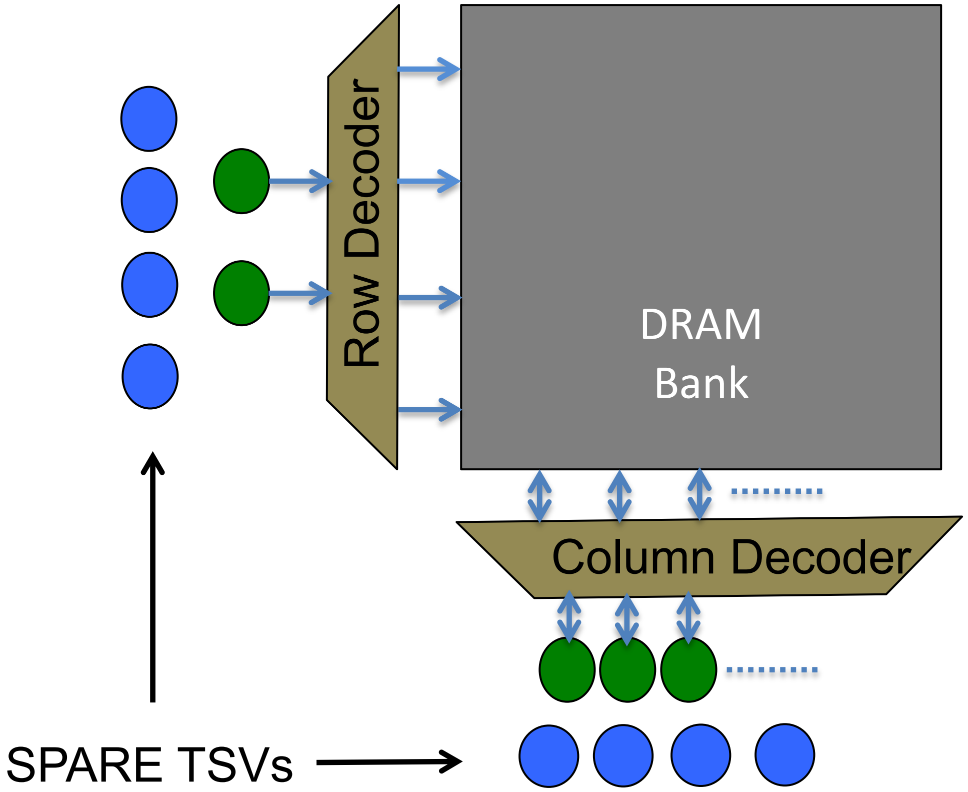
Data TSVs and Address TSVs



DESIGN-TIME TSV SPARING

Designers provision spares TSVs alongside

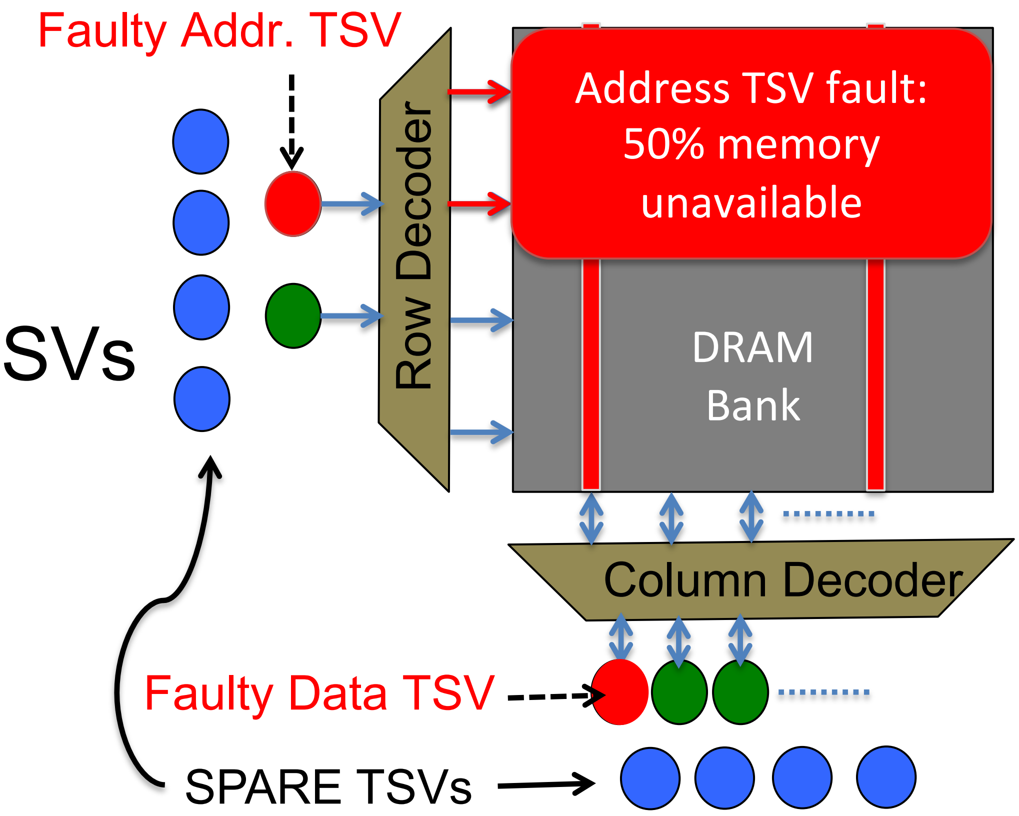
Data TSVs and Address TSVs



Additional Spare TSVs can replace faulty TSVs

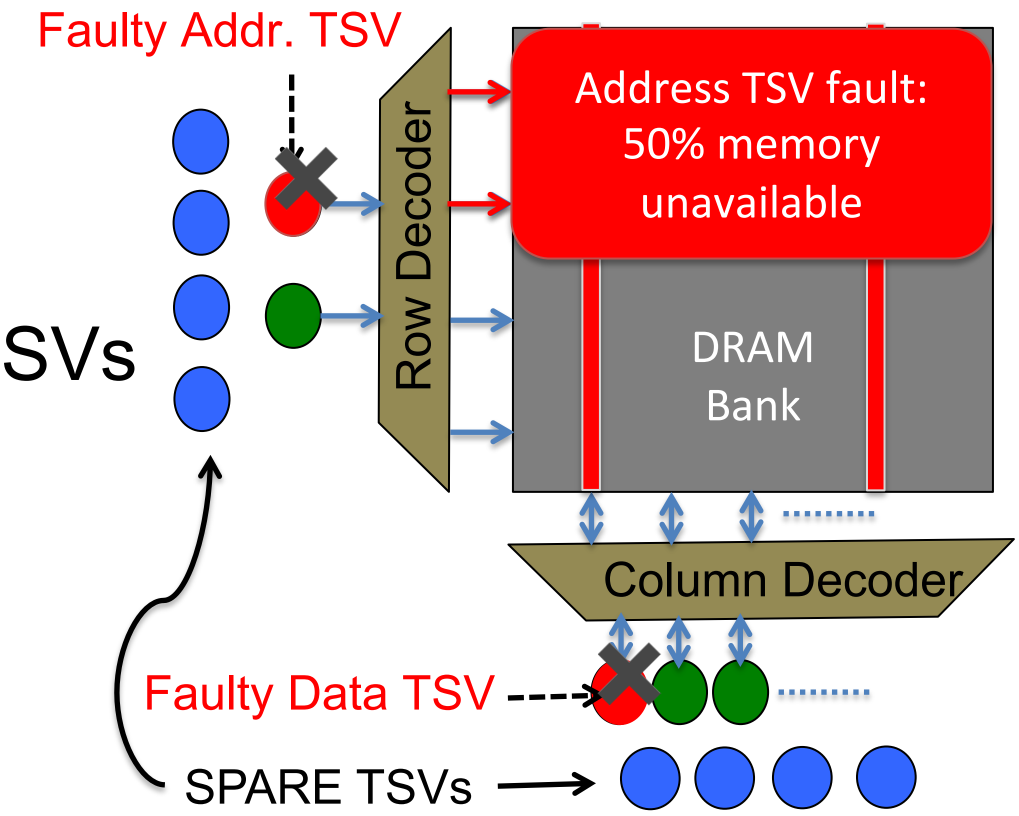
DESIGN-TIME TSV SPARING: OPERATION

- Deactivate Broken TSVs



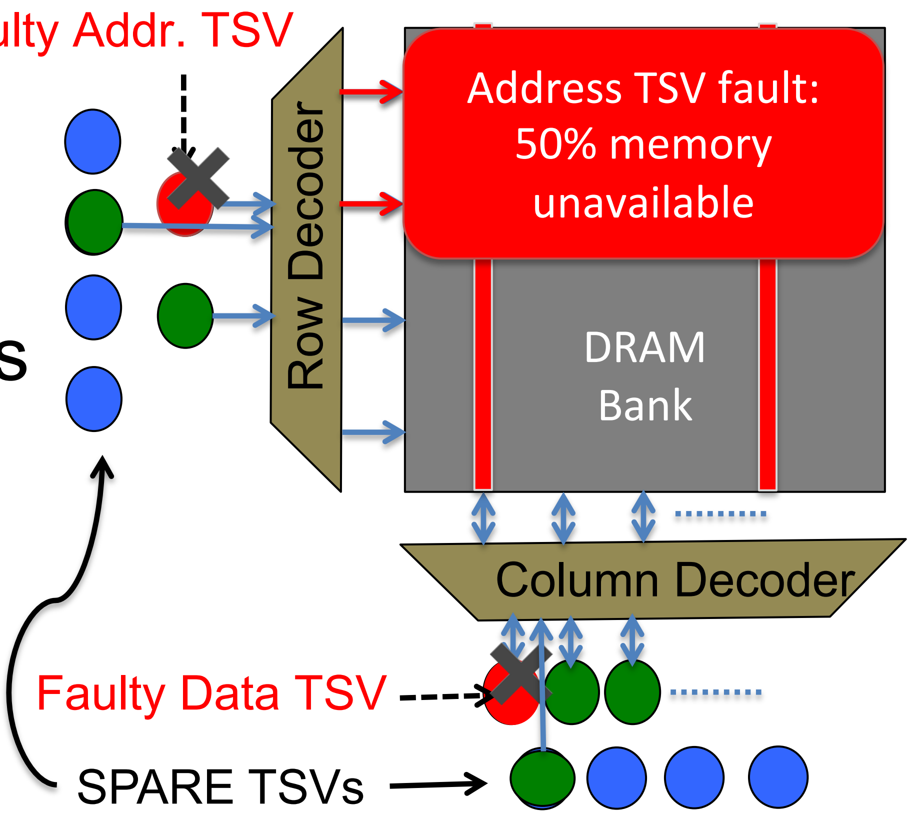
DESIGN-TIME TSV SPARING: OPERATION

- Deactivate Broken TSVs



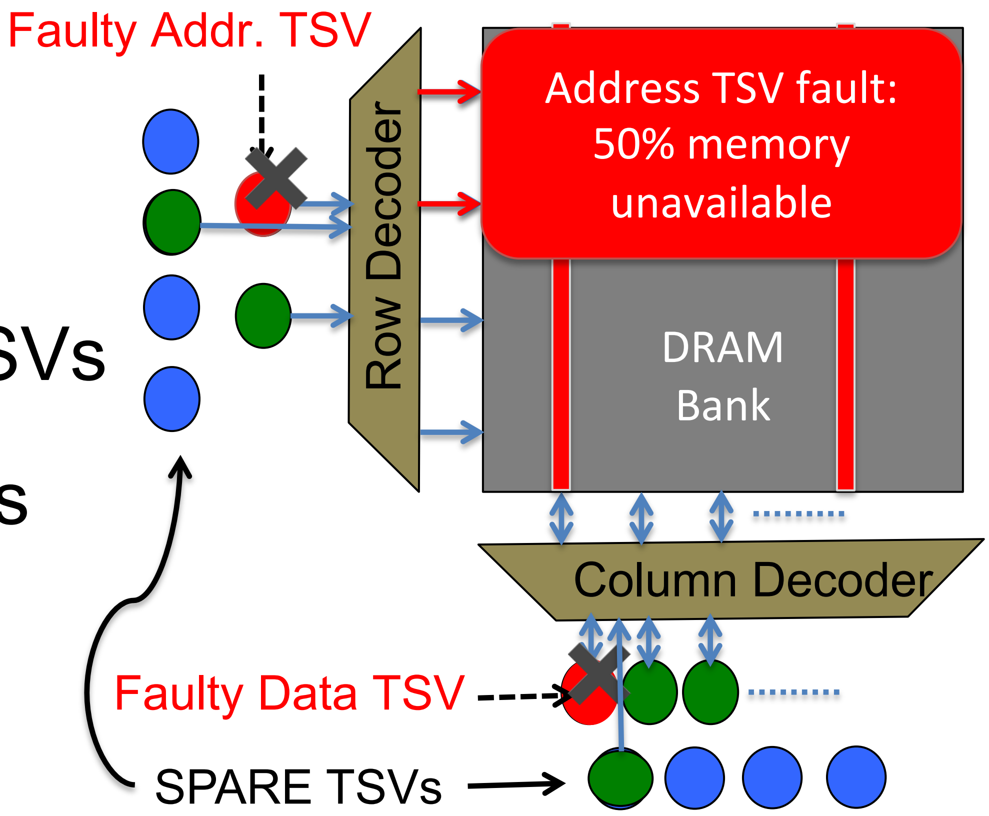
DESIGN-TIME TSV SPARING: OPERATION

- Deactivate Broken TSVs
- Activate SPARE TSVs



DESIGN-TIME TSV SPARING: OPERATION

- Deactivate Broken TSVs
- Activate SPARE TSVs



Deactivation of Faulty TSVs and Activation of Spare TSVs is performed at design time

DESIGN-TIME TSV SPARING: PROBLEMS

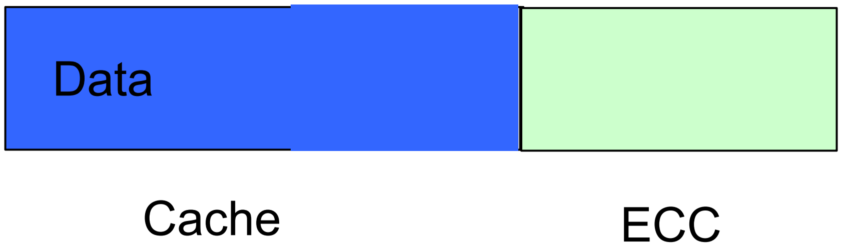
Additional TSVs are required for TSV Sparing

and

What happens if TSVs turn faulty at runtime?

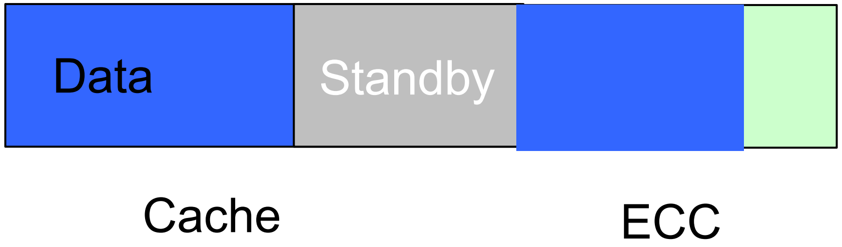
TSV-SWAP: RUNTIME TSV SPARING

STEP-1: CREATE STANDBY TSVs



TSV-SWAP: RUNTIME TSV SPARING

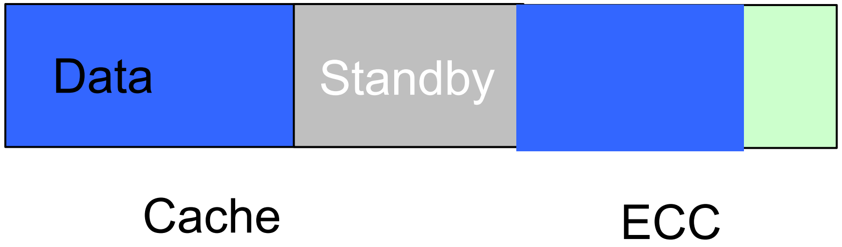
STEP-1: CREATE STANDBY TSVs

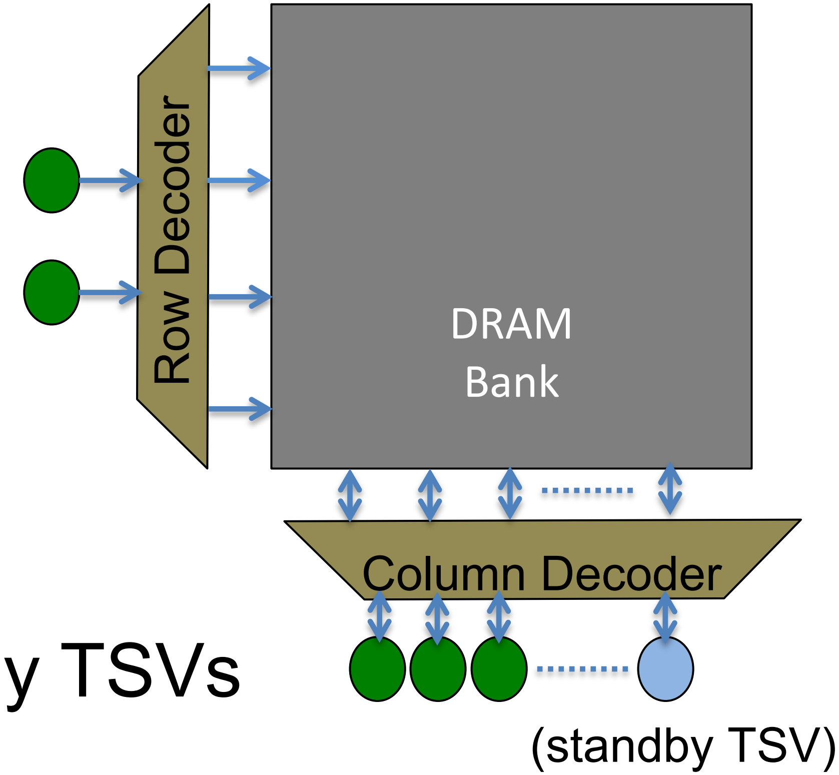


- Few Data TSVs as Standby TSVs
- Replicate Standby Data in ECC

TSV-SWAP: RUNTIME TSV SPARING

STEP-1: CREATE STANDBY TSVs





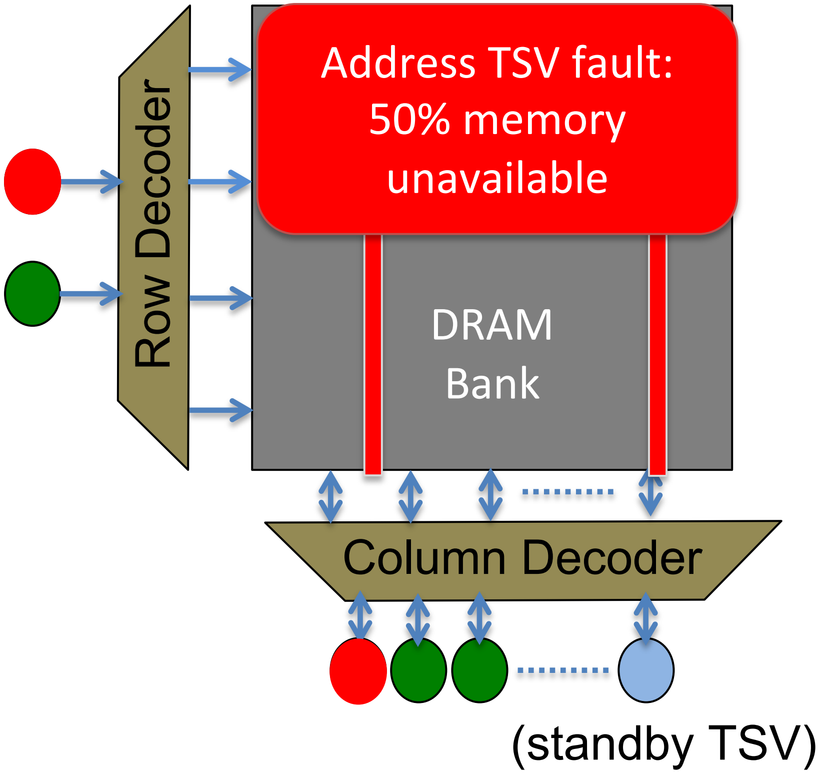
- Few Data TSVs as Standby TSVs
- Replicate Standby Data in ECC

Data TSVs reused as Standby TSVs

TSV-SWAP: RUNTIME TSV SPARING

STEP-2: DETECTING FAULTY TSVs

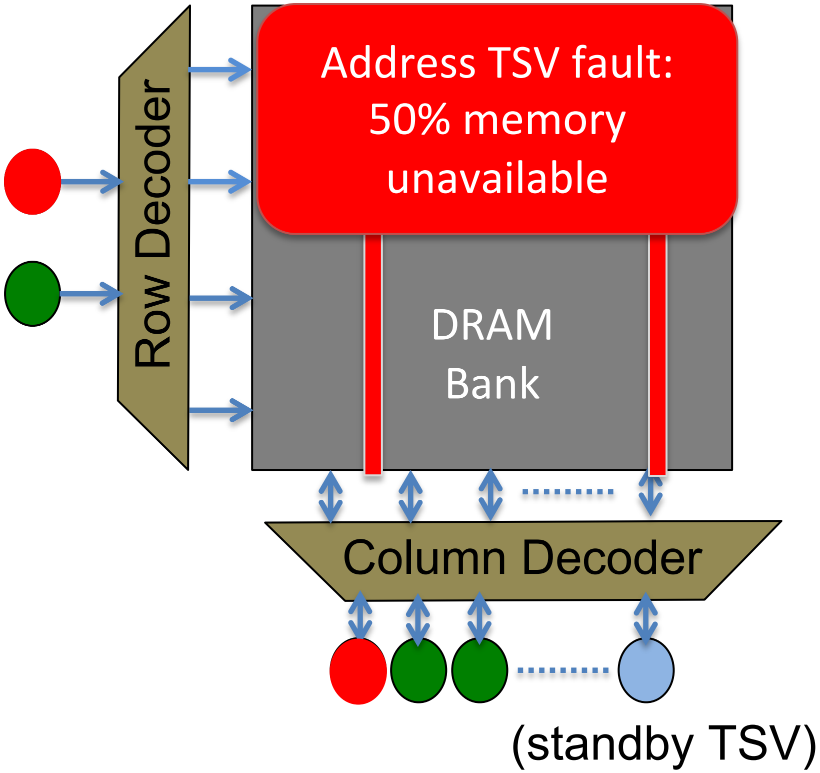
- CRC-32 ➡ address + data



TSV-SWAP: RUNTIME TSV SPARING

STEP-2: DETECTING FAULTY TSVs

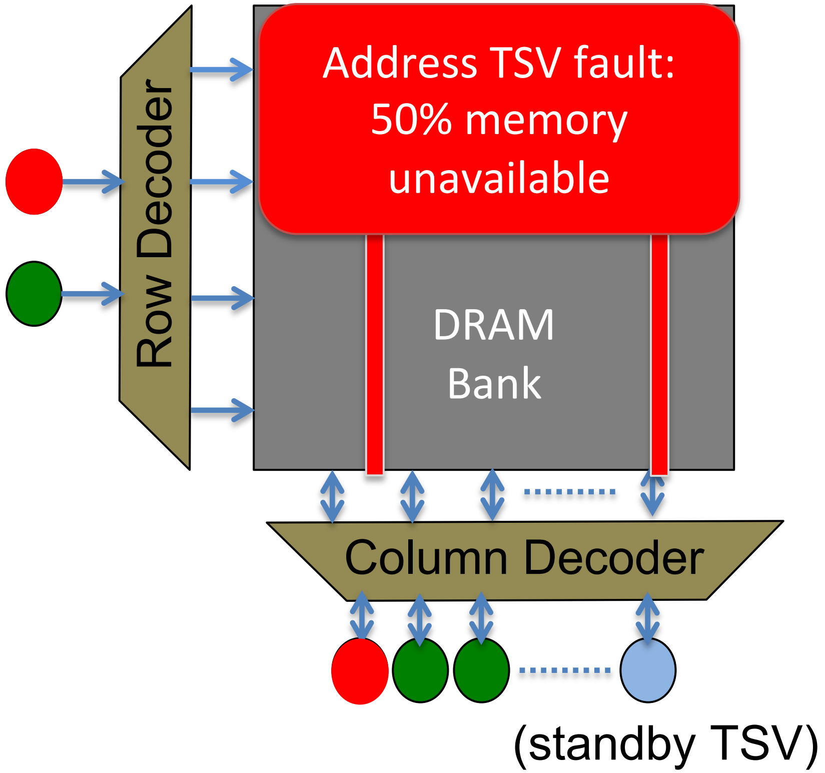
- CRC-32 ➡ address + data
- BIST diagnoses faulty TSVs



TSV-SWAP: RUNTIME TSV SPARING

STEP-2: DETECTING FAULTY TSVs

- CRC-32 ➡ address + data
- BIST diagnoses faulty TSVs

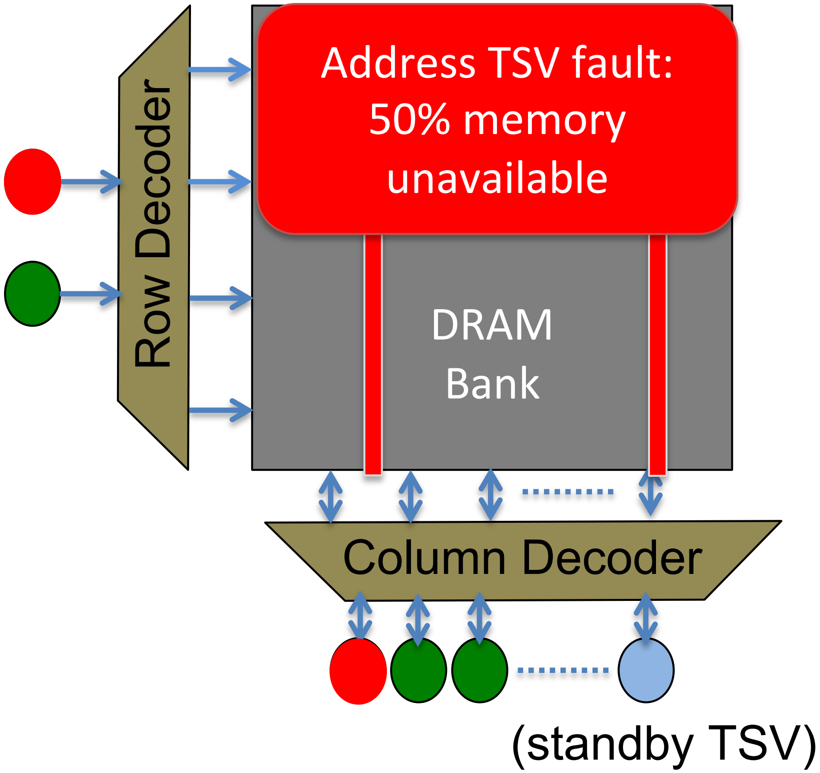


Data vs Address TSV Faults Using CRC-32+BIST

TSV-SWAP: RUNTIME TSV SPARING

STEP-3: REDIRECTING FAULTY TSVs

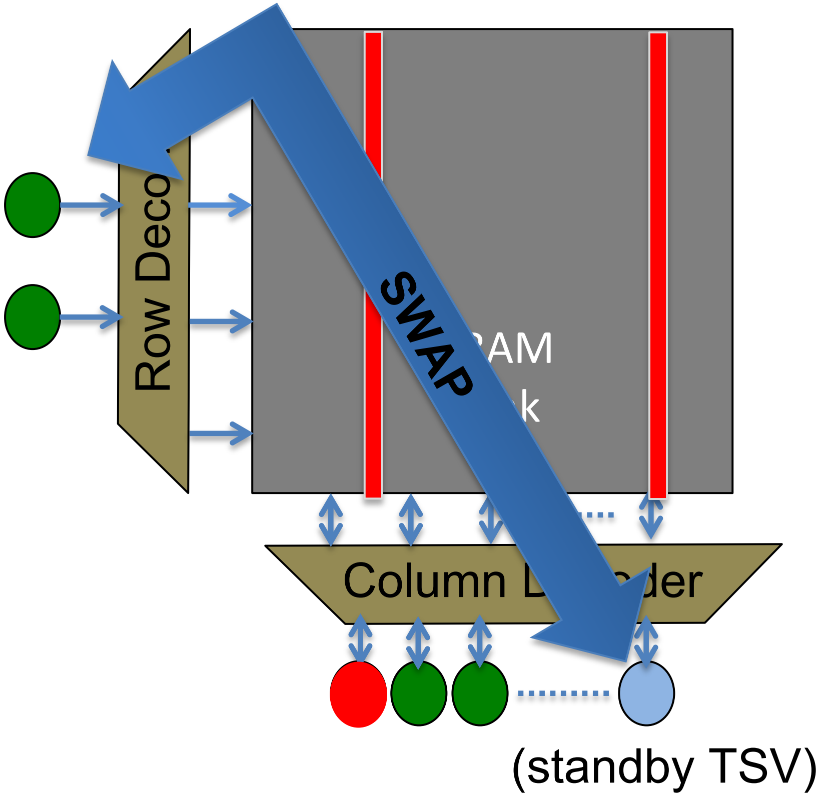
Swap Faulty TSVs with Standby TSVs at runtime



TSV-SWAP: RUNTIME TSV SPARING

STEP-3: REDIRECTING FAULTY TSVs

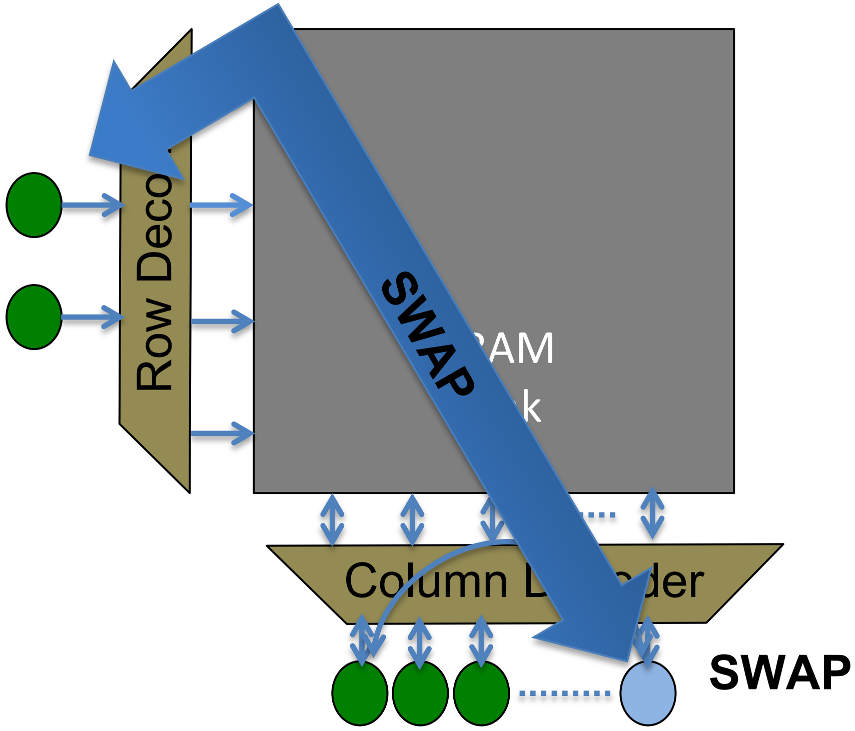
Swap Faulty TSVs with Standby TSVs at runtime



TSV-SWAP: RUNTIME TSV SPARING

STEP-3: REDIRECTING FAULTY TSVs

Swap Faulty TSVs with Standby TSVs at runtime

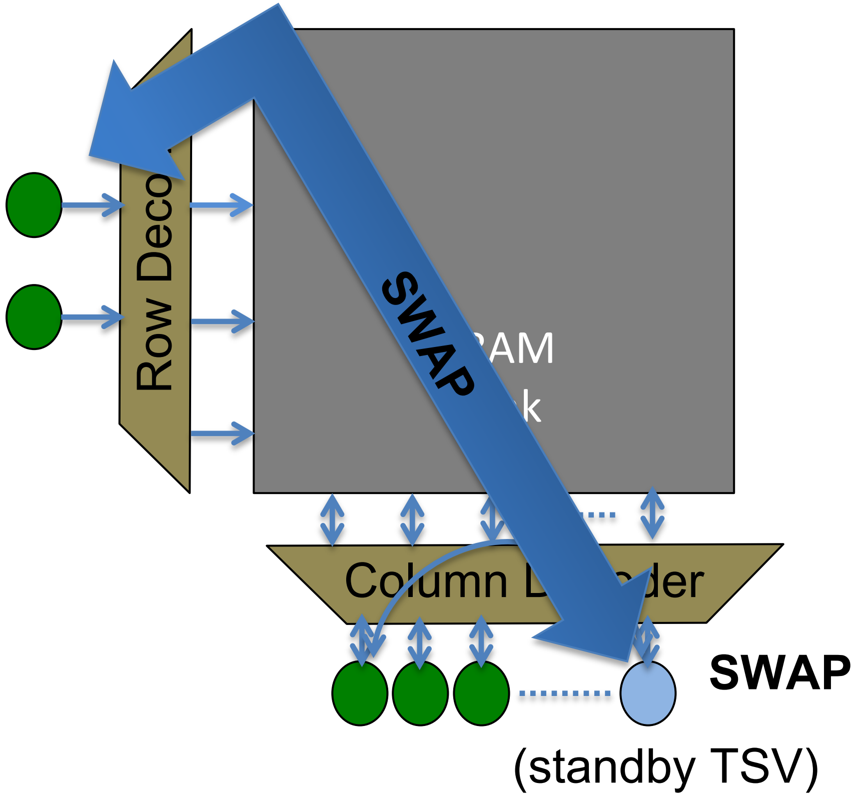


(standby TSV)

TSV-SWAP: RUNTIME TSV SPARING

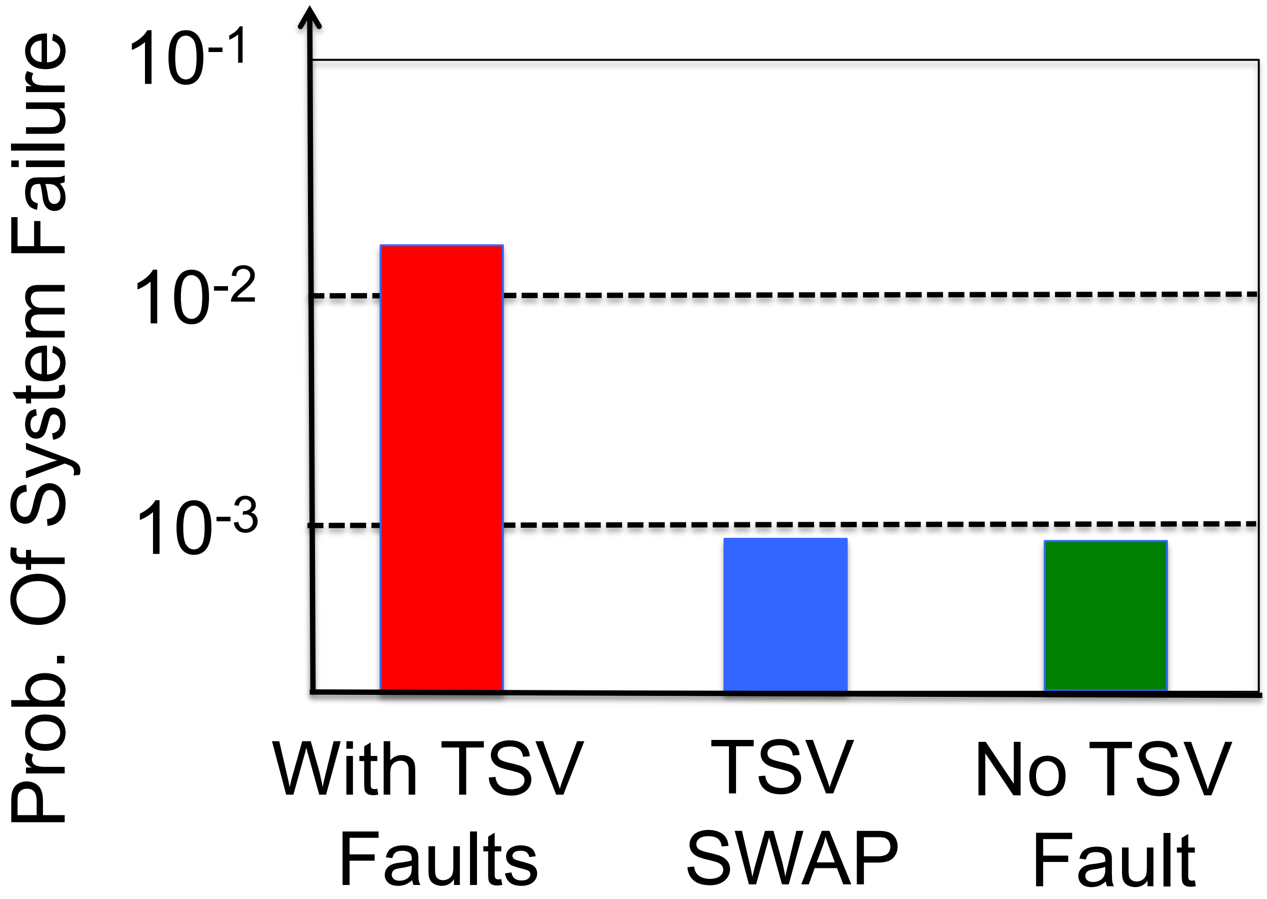
STEP-3: REDIRECTING FAULTY TSVs

Swap Faulty TSVs with Standby TSVs at runtime



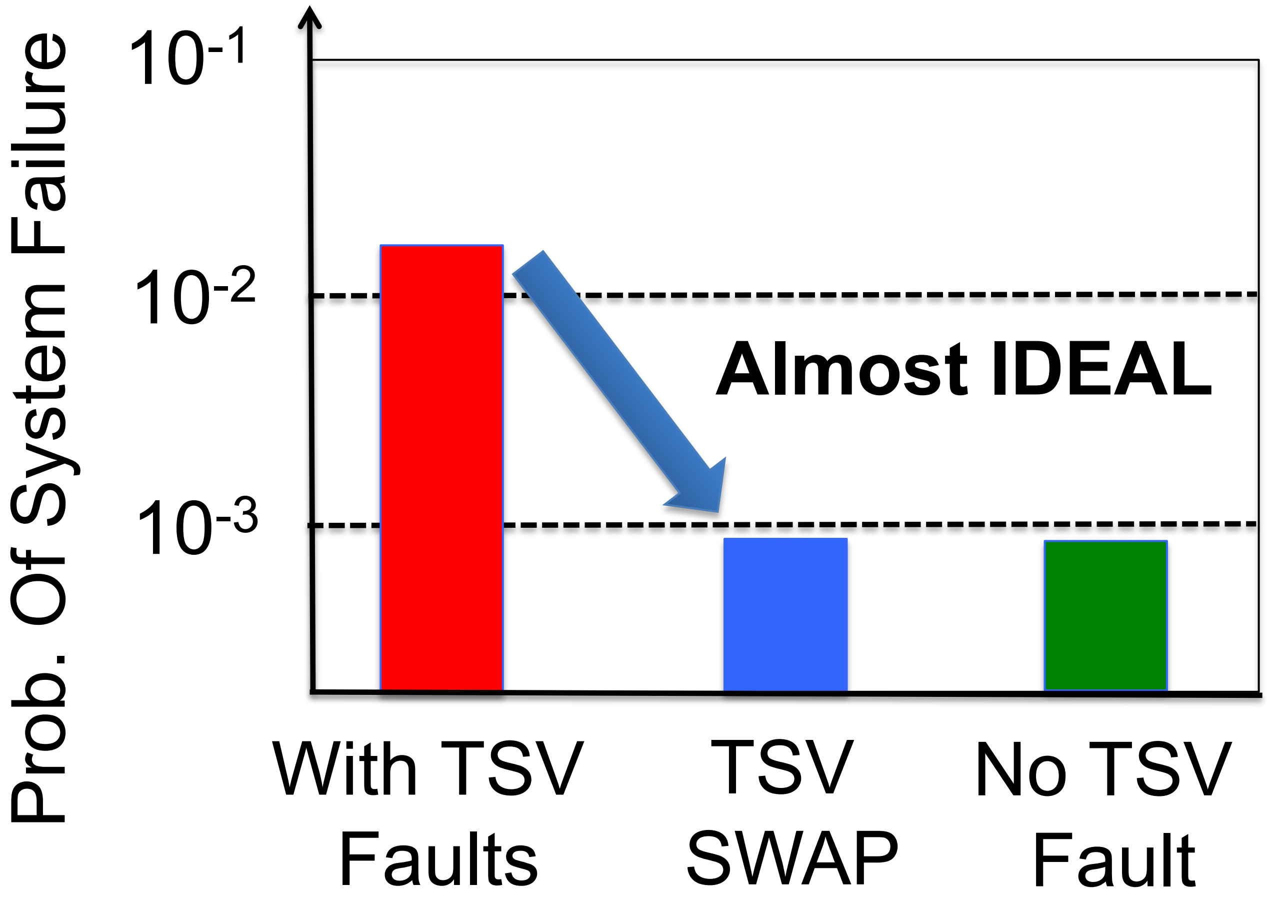
TSV-SWAP is a runtime technique that does not rely on additional spare TSVs

EFFECTIVENESS OF TSV-SWAP



Rate: One TSV Fault Every 7 years

EFFECTIVENESS OF TSV-SWAP



Rate: One TSV Fault Every 7 years

TSV-SWAP is Effective at Tolerating TSV Faults

OUTLINE

- Introduction and Background
- Citadel
- Scheme - 1 : TSV-SWAP
- Scheme - 2 : Three Dimensional Parity (3DP)
- Scheme - 3 : Dynamic Dual Grain Sparing (DDS)
- Summary

TRI DIMENSIONAL PARITY (3DP)

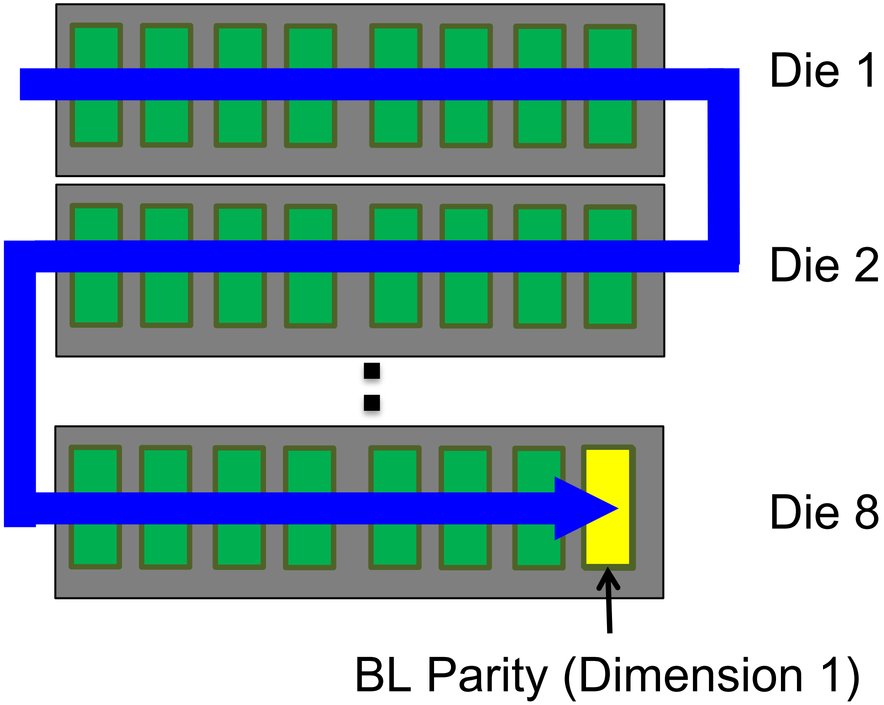
- Use RAID-5 like scheme over three dimensions

TRI DIMENSIONAL PARITY (3DP)

- Use RAID-5 like scheme over three dimensions
- Detect using CRC-32

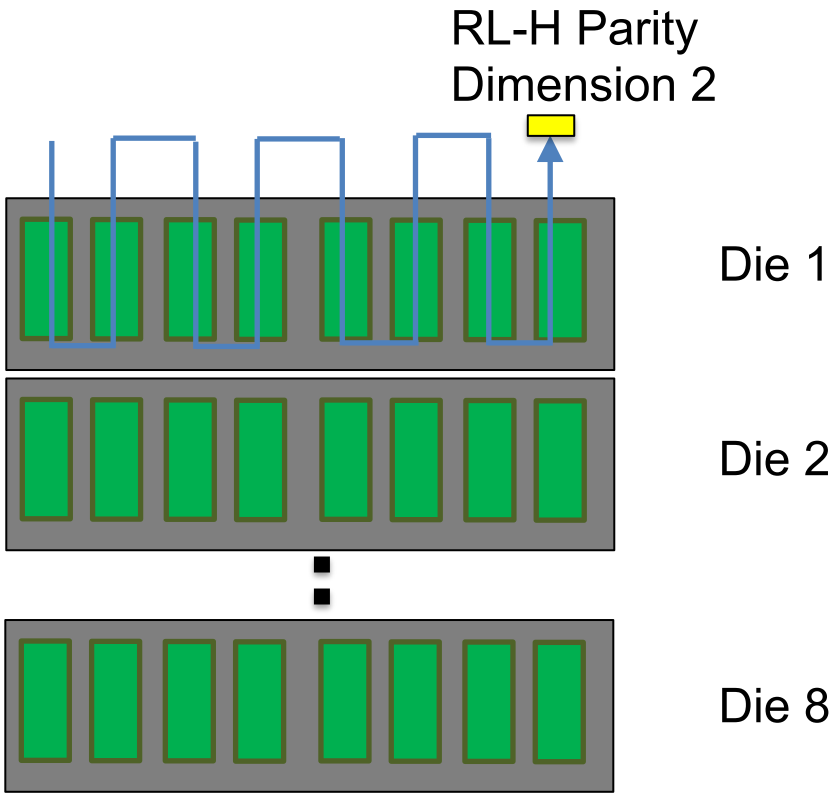
TRI DIMENSIONAL PARITY (3DP)

- Use RAID-5 like scheme over three dimensions
- Detect using CRC-32
- Correct using Parity
  - – Bank Level (BL) Parity



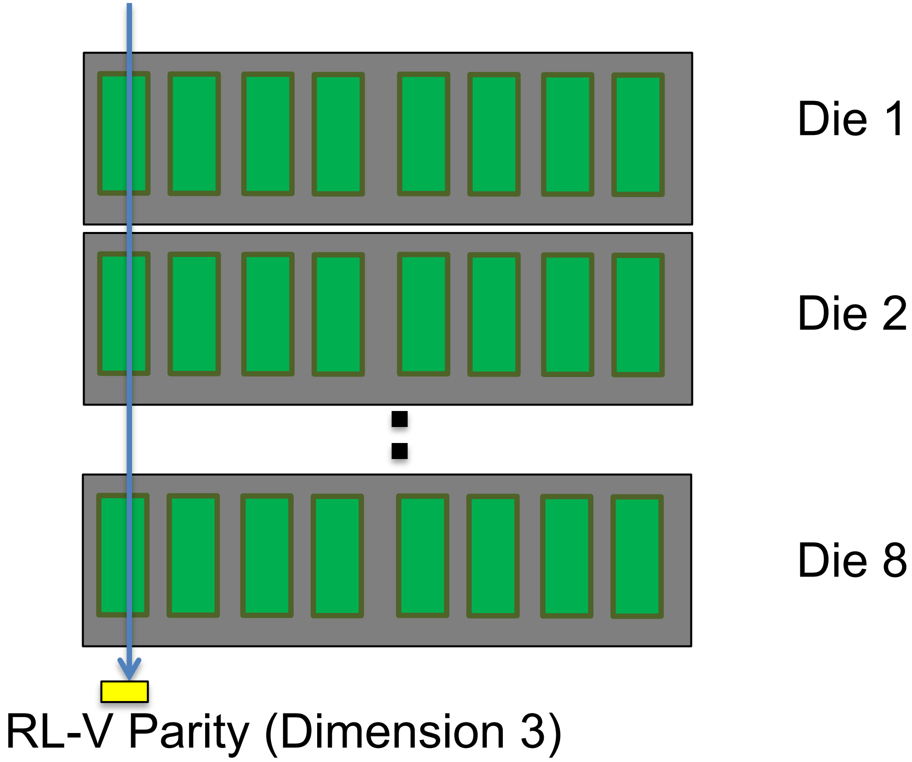
TRI DIMENSIONAL PARITY (3DP)

- Use RAID-5 like scheme over three dimensions
- Detect using CRC-32
- Correct using Parity
  - – Bank Level (BL) Parity
  - – Row Level (RL-H) Parity per die



TRI DIMENSIONAL PARITY (3DP)

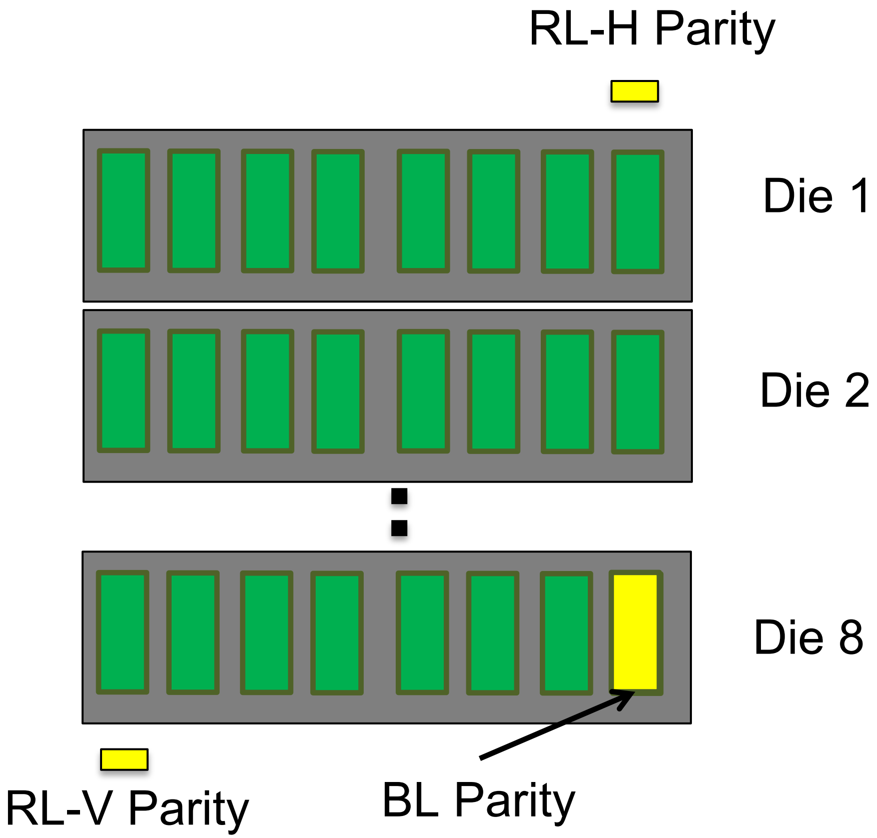
- Use RAID-5 like scheme over three dimensions
- Detect using CRC-32
- Correct using Parity
  - Bank Level (BL) Parity
  - Row Level (RL-H) Parity per die
  - Row Level (RL-V) Parity across dies



Three Dimensions Help In Multi-Fault Handling

3DP: DATA CORRECTION

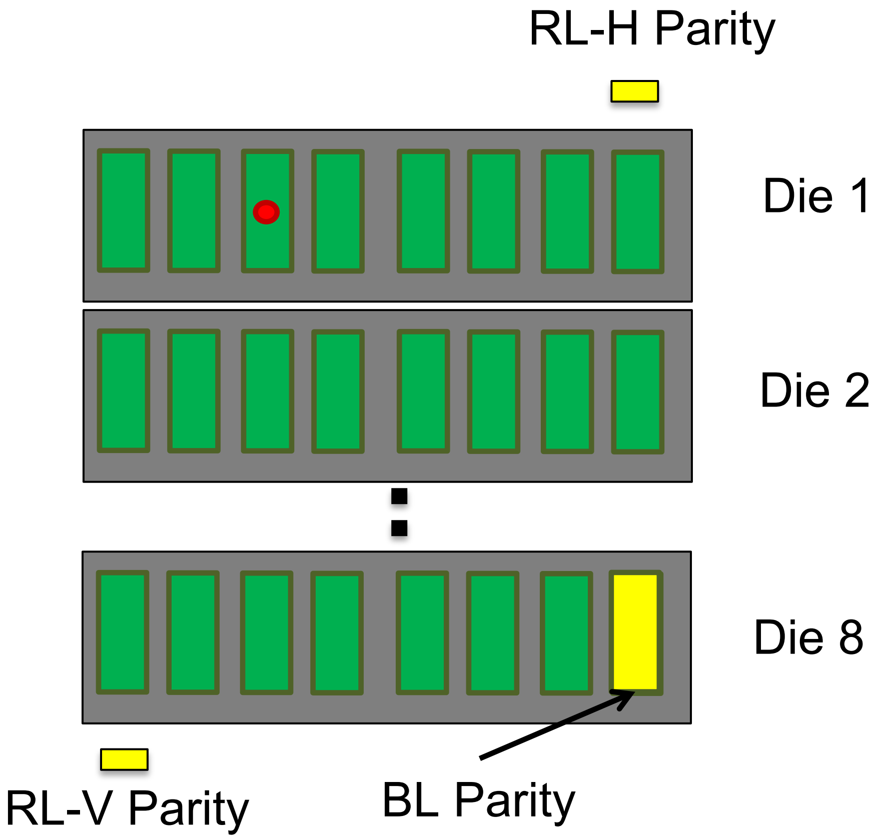
If Fault → Compute Parity and Correct



3DP: DATA CORRECTION

If Fault → Compute Parity and Correct

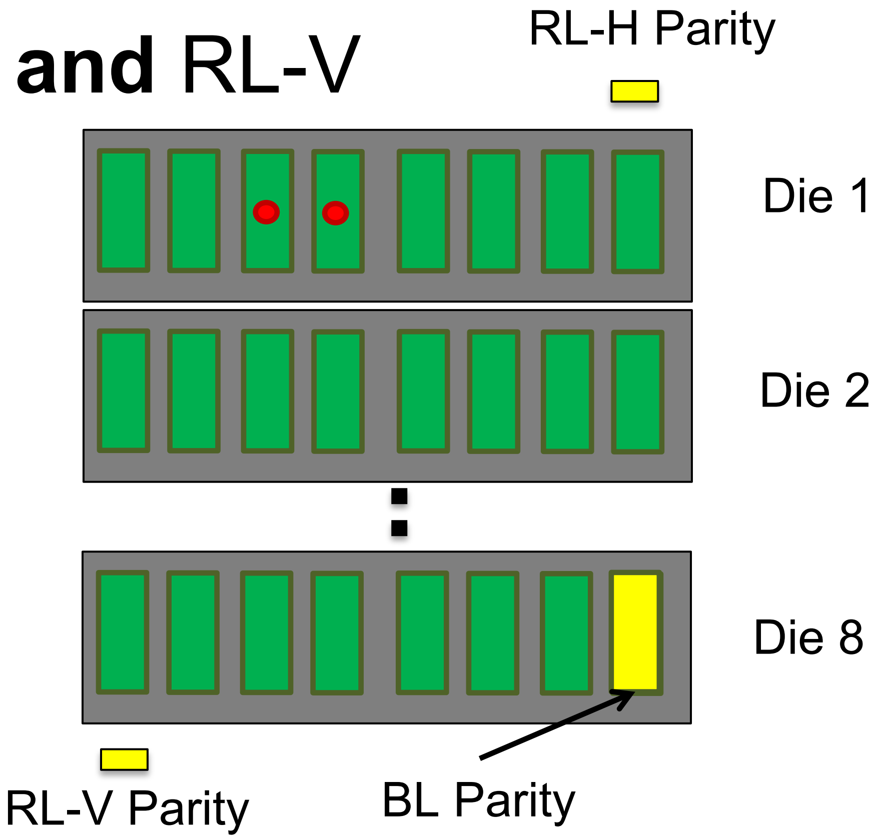
- 1-Small Fault → RL-H **or** RL-V



3DP: DATA CORRECTION

If Fault ➡ Compute Parity and Correct

- 1-Small Fault ➡ RL-H **or** RL-V
- 2-Small Faults ➡ RL-H **and** RL-V

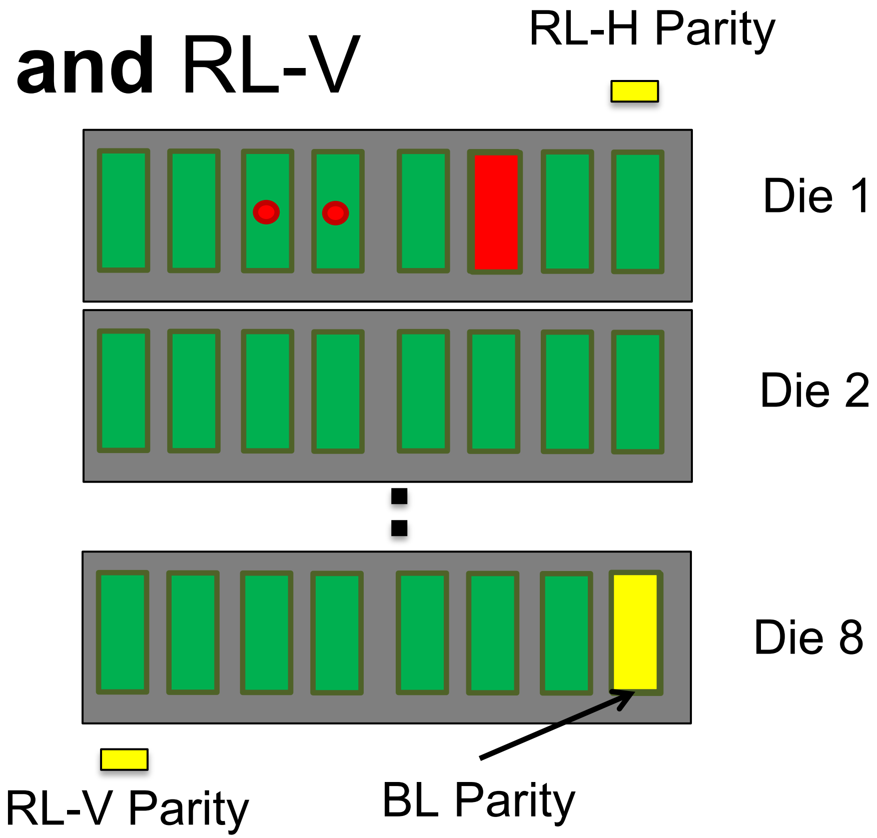


3DP: DATA CORRECTION

If Fault → Compute Parity and Correct

- 1-Small Fault → RL-H **or** RL-V
- 2-Small Faults → RL-H **and** RL-V
- 2 Small + 1 Large Fault

  RL-H **and** RL-V **and** BL

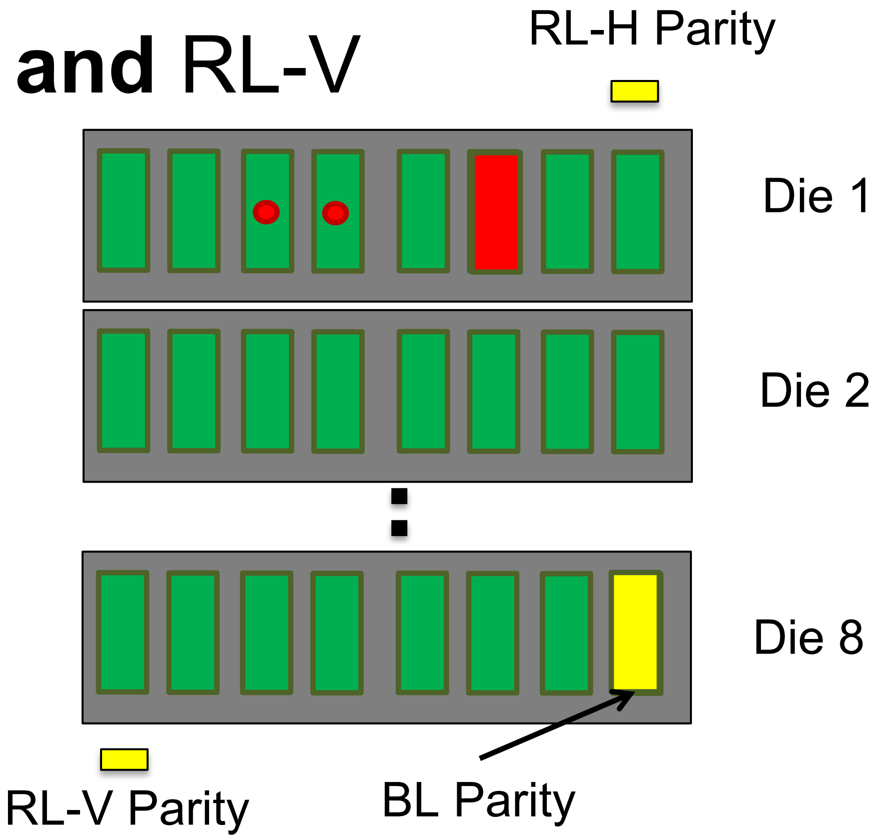


3DP: DATA CORRECTION

If Fault → Compute Parity and Correct

- 1-Small Fault → RL-H **or** RL-V
- 2-Small Faults → RL-H **and** RL-V
- 2 Small + 1 Large Fault

  RL-H **and** RL-V **and** BL



Multiple Multi-granularity Faults Are Corrected At Runtime

OVERHEADS IN UPDATING PARITY

- RL-H and RL-V Parity just 32 KB → stored in SRAM
- BL Parity is 128 MB → stored in DRAM

OVERHEADS IN UPDATING PARITY

- RL-H and RL-V Parity just 32 KB → stored in SRAM
- BL Parity is 128 MB → stored in DRAM
- Updating BL Parity has performance overhead

OVERHEADS IN UPDATING PARITY

- RL-H and RL-V Parity just 32 KB → stored in SRAM
- BL Parity is 128 MB → stored in DRAM
- Updating BL Parity has performance overhead
- Employ Demand Caching of BL Parity in LLC

OVERHEADS IN UPDATING PARITY

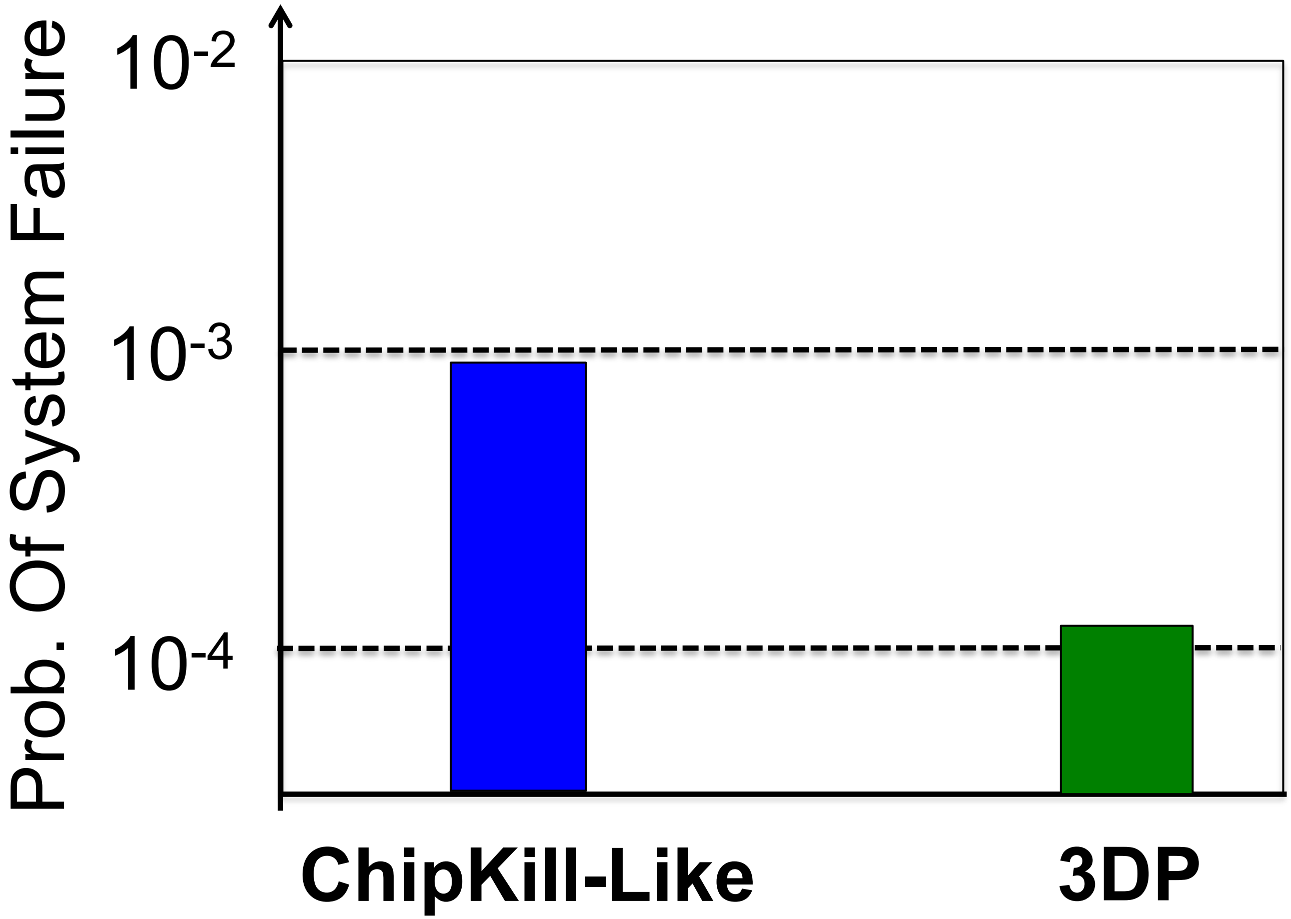
- RL-H and RL-V Parity just 32 KB → stored in SRAM
- BL Parity is 128 MB → stored in DRAM
- Updating BL Parity has performance overhead
- Employ Demand Caching of BL Parity in LLC
- Mitigate overheads of updating BL Parity

OVERHEADS IN UPDATING PARITY

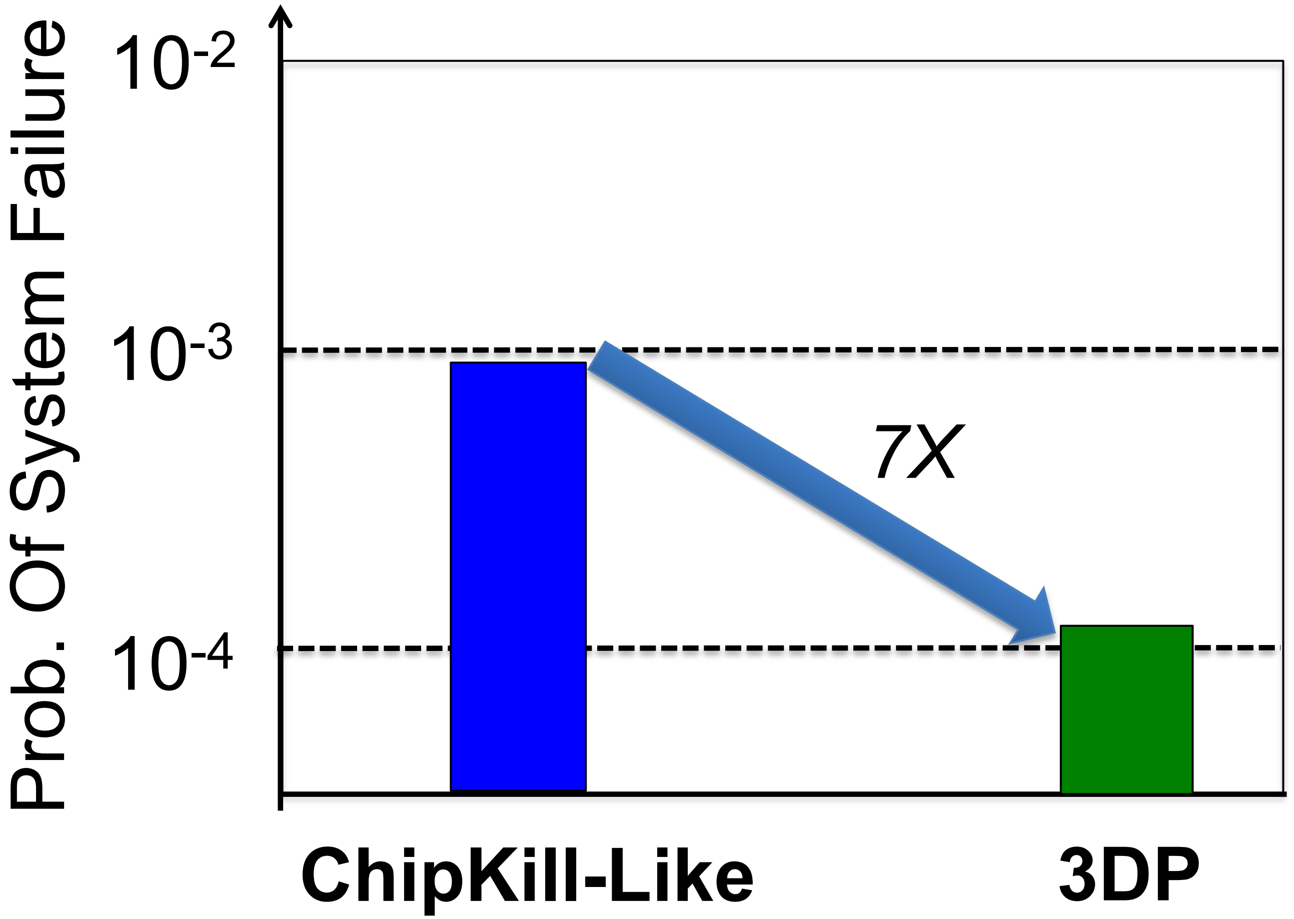
- RL-H and RL-V Parity just 32 KB → stored in SRAM
- BL Parity is 128 MB → stored in DRAM
- Updating BL Parity has performance overhead
- Employ Demand Caching of BL Parity in LLC
- Mitigate overheads of updating BL Parity

Demand Caching of BL Parity Has 85% Hit Rate And Mitigates Performance Overheads

EFFECTIVENESS OF 3DP



EFFECTIVENESS OF 3DP



3DP is 7X Stronger Than A ChipKill-Like Scheme

OUTLINE

- Introduction and Background
- Citadel
- Scheme - 1 : TSV-SWAP
- Scheme - 2 : Three Dimensional Parity (3DP)
- Scheme - 3 : Dynamic Dual Grain Sparing (DDS) ⇦
- Summary

WHY SPARE FAULTY DATA?

- Correcting Large Faults Has Performance Overhead

WHY SPARE FAULTY DATA?

- Correcting Large Faults Has Performance Overhead
- To prevent accumulation of faults

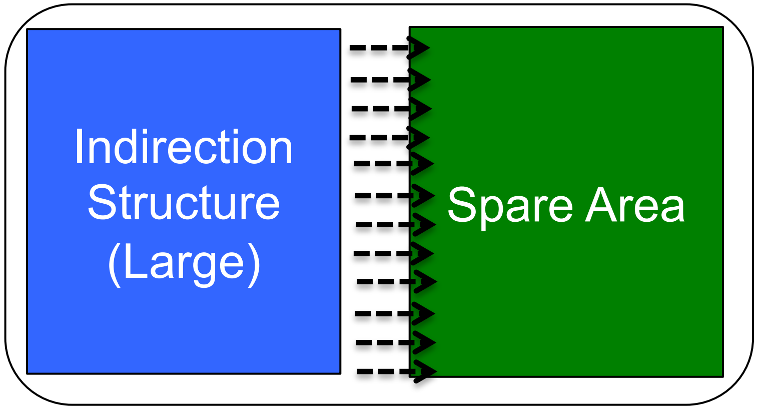
WHY SPARE FAULTY DATA?

- Correcting Large Faults Has Performance Overhead
- To prevent accumulation of faults

Sparing Mitigates Performance Overheads and Enhances Reliability

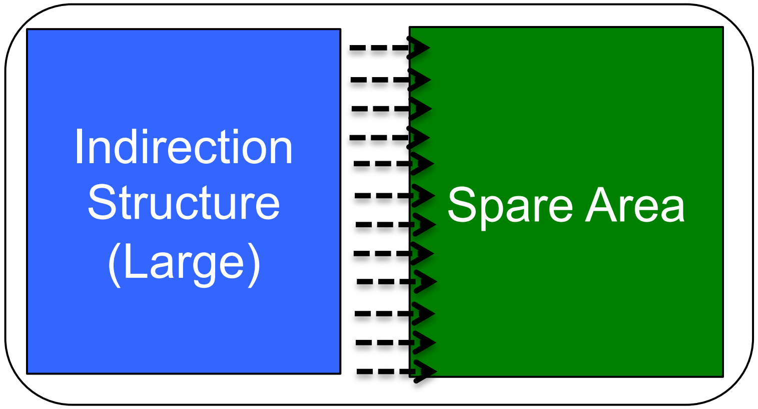
TRACKING STRUCTURES IN SPARING

- Row Level Tracking
  - Large Indirection Structure
  - Sparing Area Used Efficiently

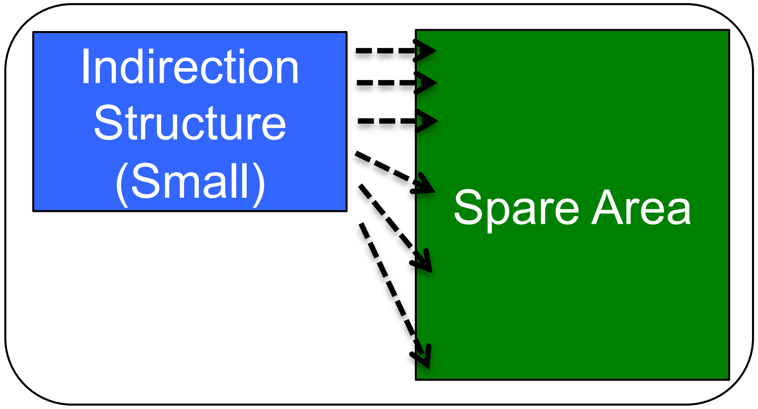


TRACKING STRUCTURES IN SPARING

- Row Level Tracking
  - Large Indirection Structure
  - Sparing Area Used Efficiently

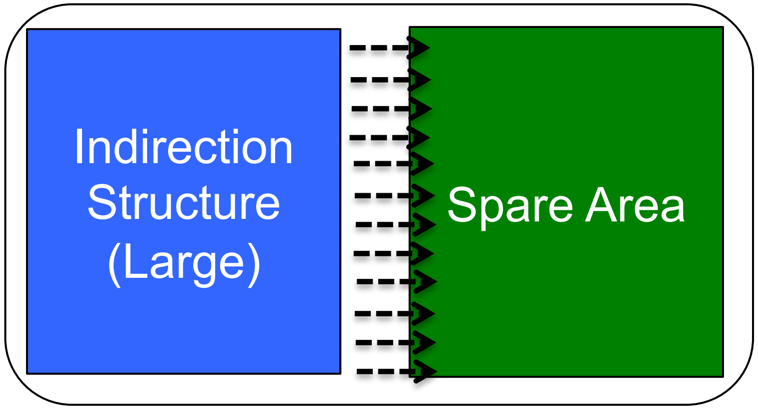


- Bank Level Tracking
  - Small Indirection Structure
  - Sparing Area Used Inefficiently

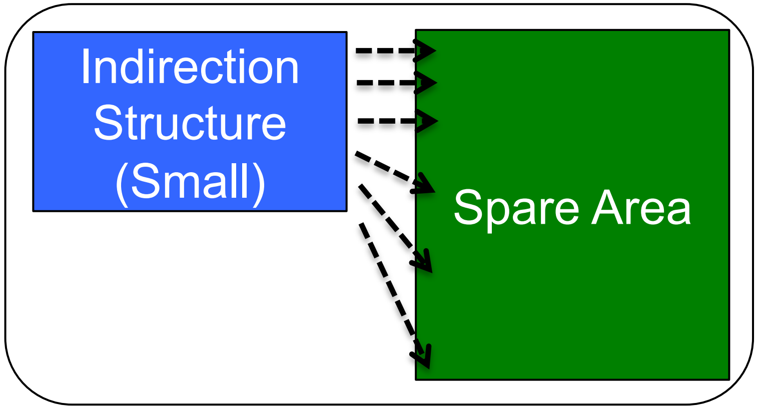


TRACKING STRUCTURES IN SPARING

- Row Level Tracking
  - Large Indirection Structure
  - Sparing Area Used Efficiently



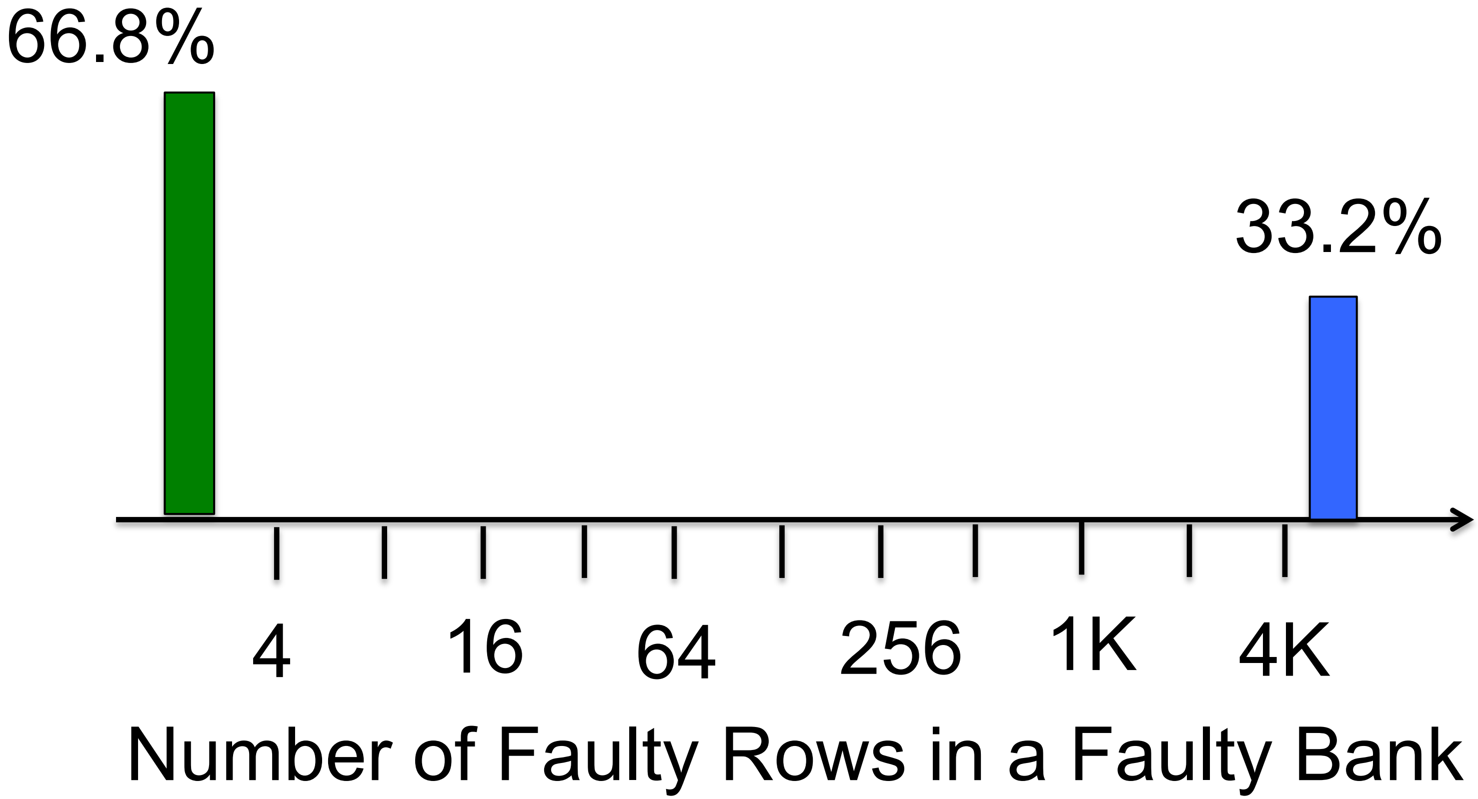
- Bank Level Tracking
  - Small Indirection Structure
  - Sparing Area Used Inefficiently



Ideally We Need Small Indirection Structures Which Use Spare Area Efficiently

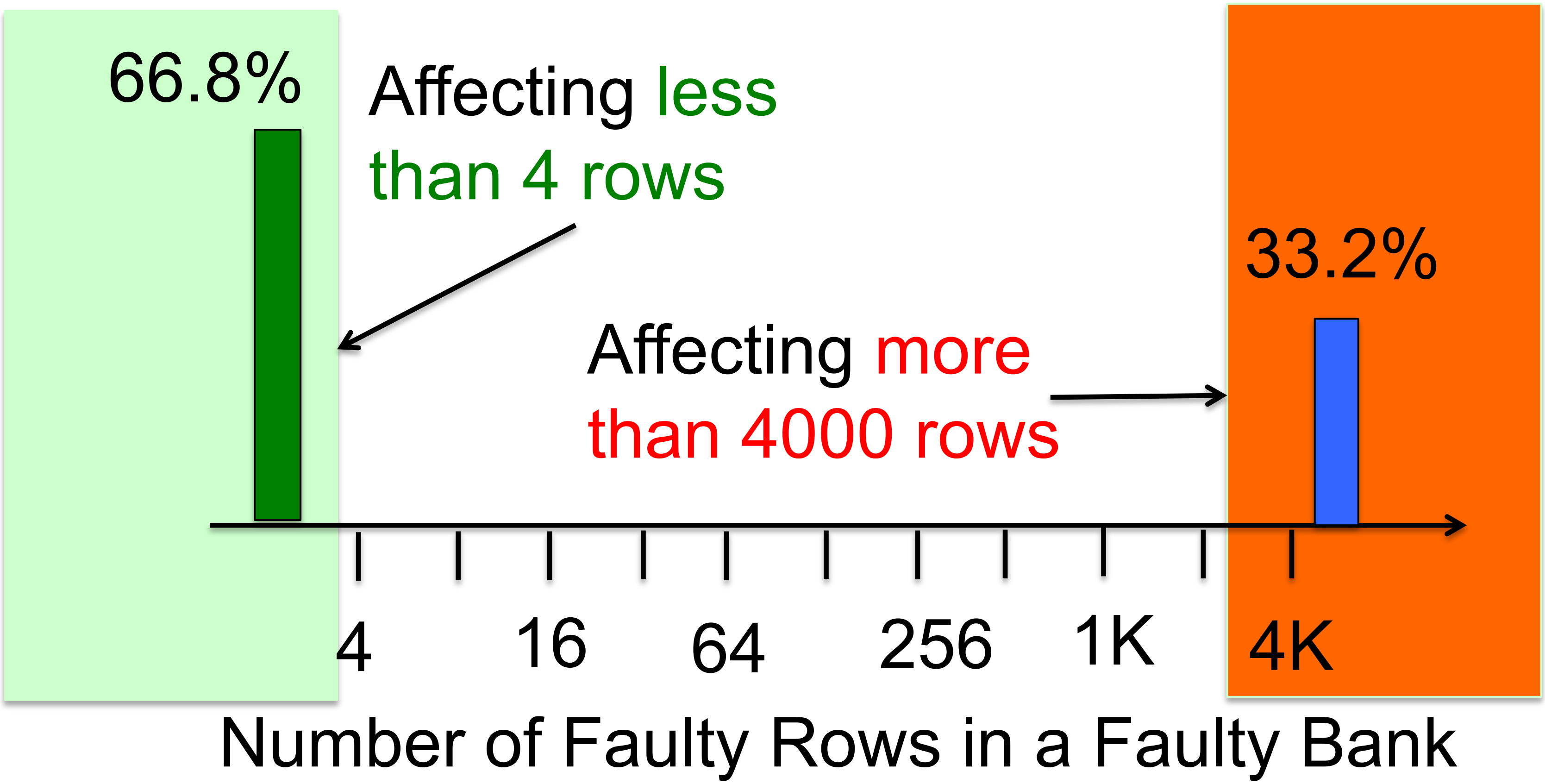
BIMODAL FAILURES

- **Observation** : Either < 4 or > 4000 row failures



BIMODAL FAILURES

- **Observation** : Either < 4 or > 4000 row failures

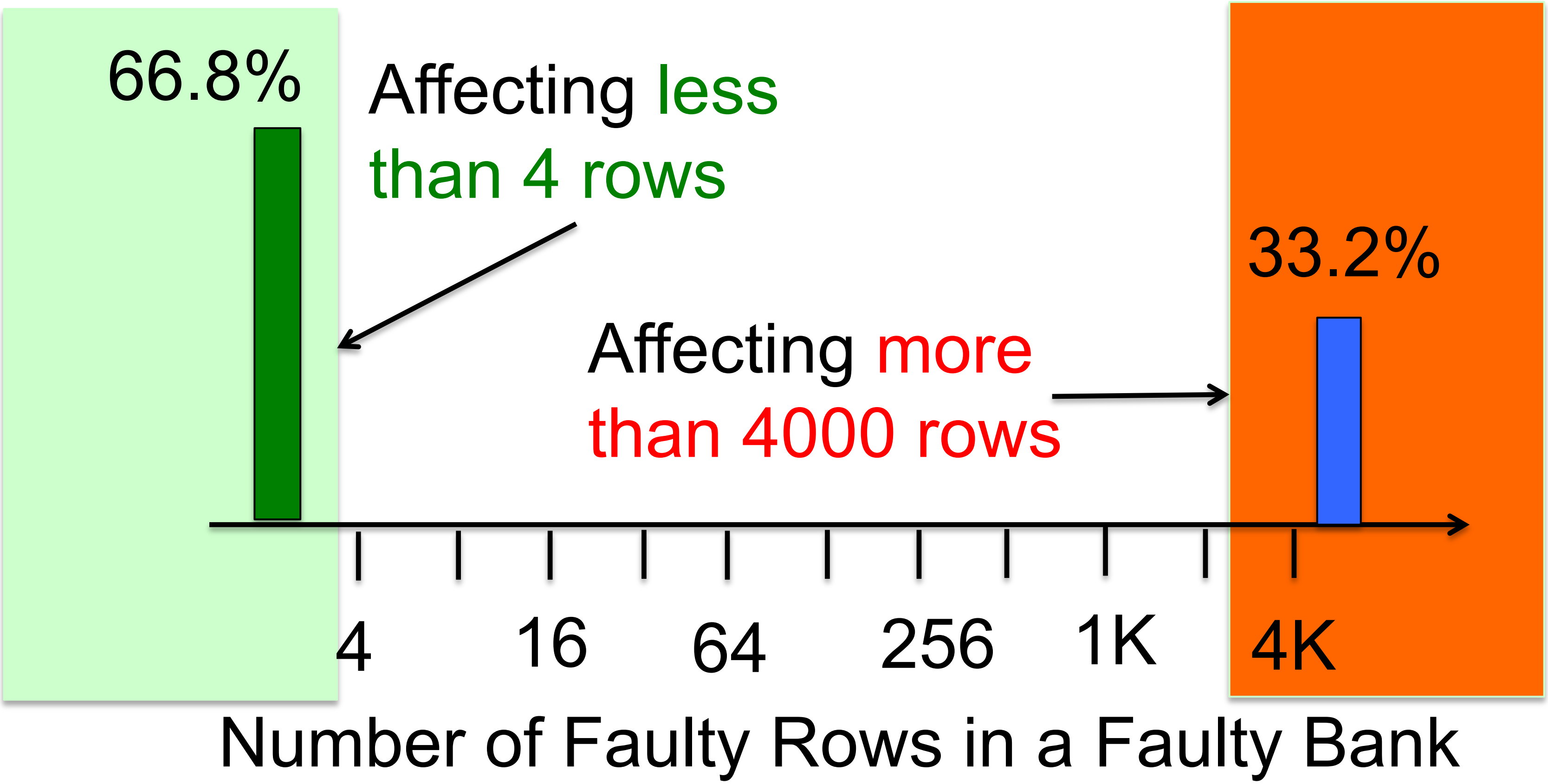


Affecting less than 4 rows

Affecting more than 4000 rows

BIMODAL FAILURES

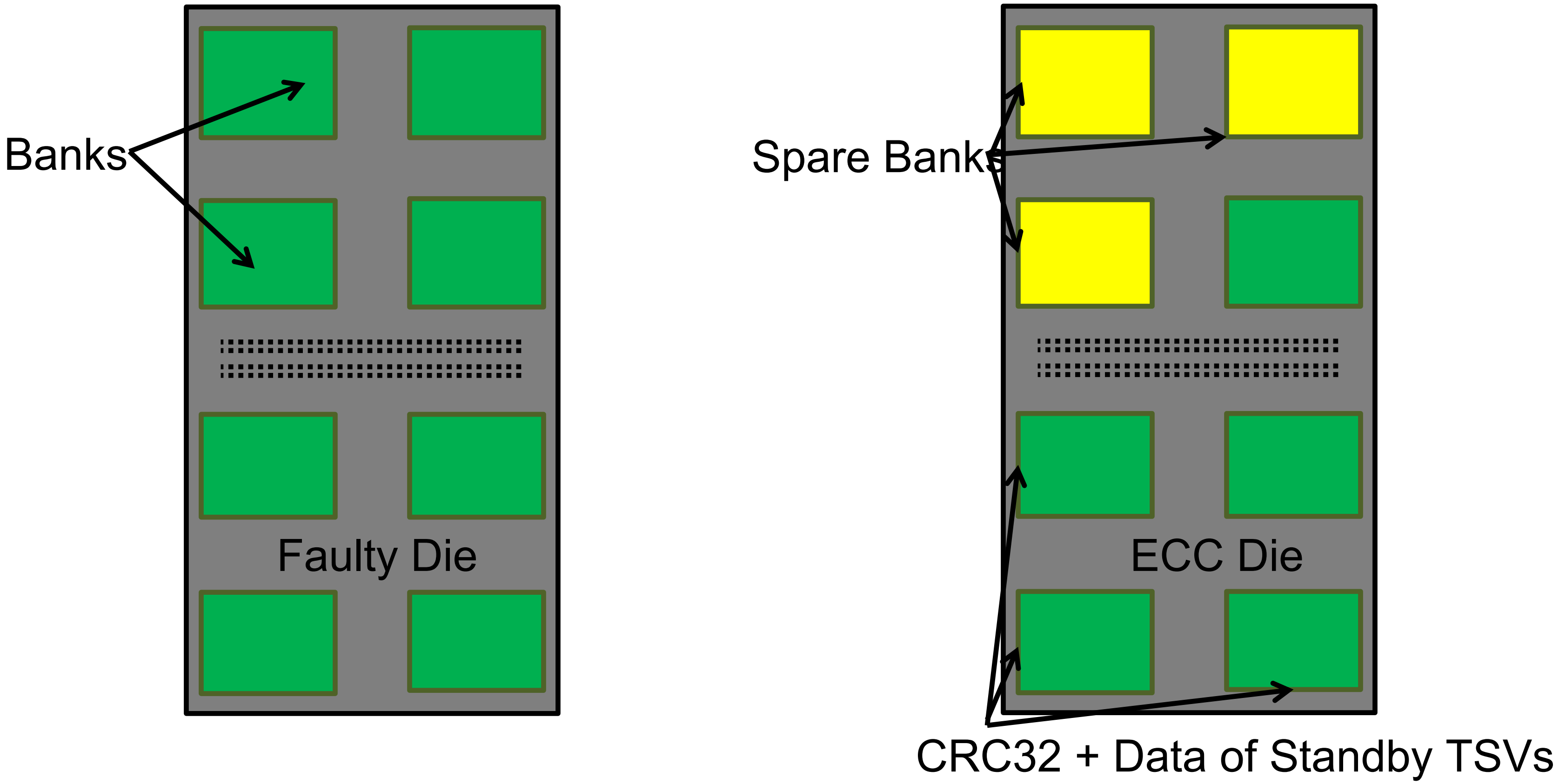
- **Observation** : Either < 4 or > 4000 row failures



Spare Faulty Regions At Two Granularities

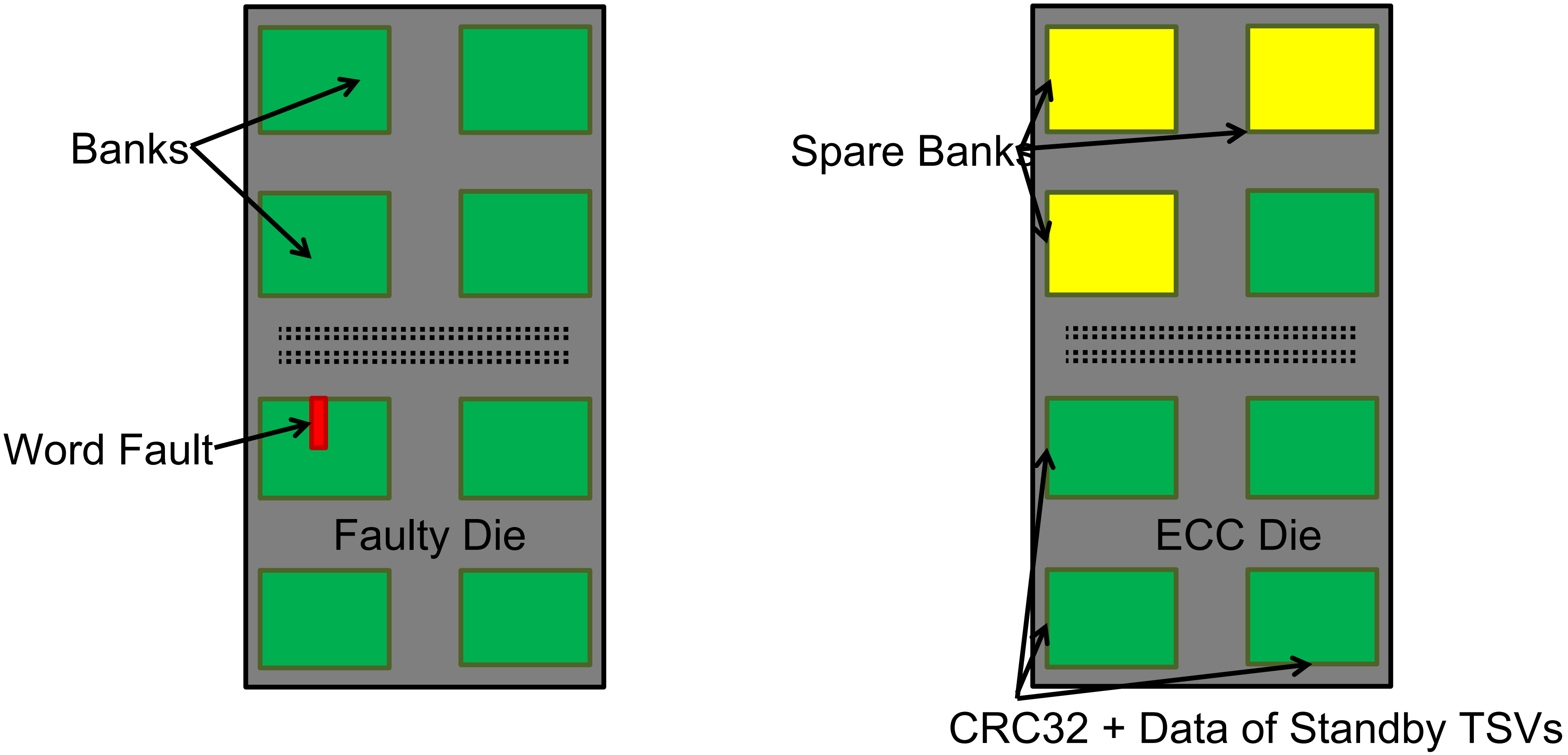
DYNAMIC DUAL GRAIN SPAIRING

- Provision Spare Area for Two Granularities



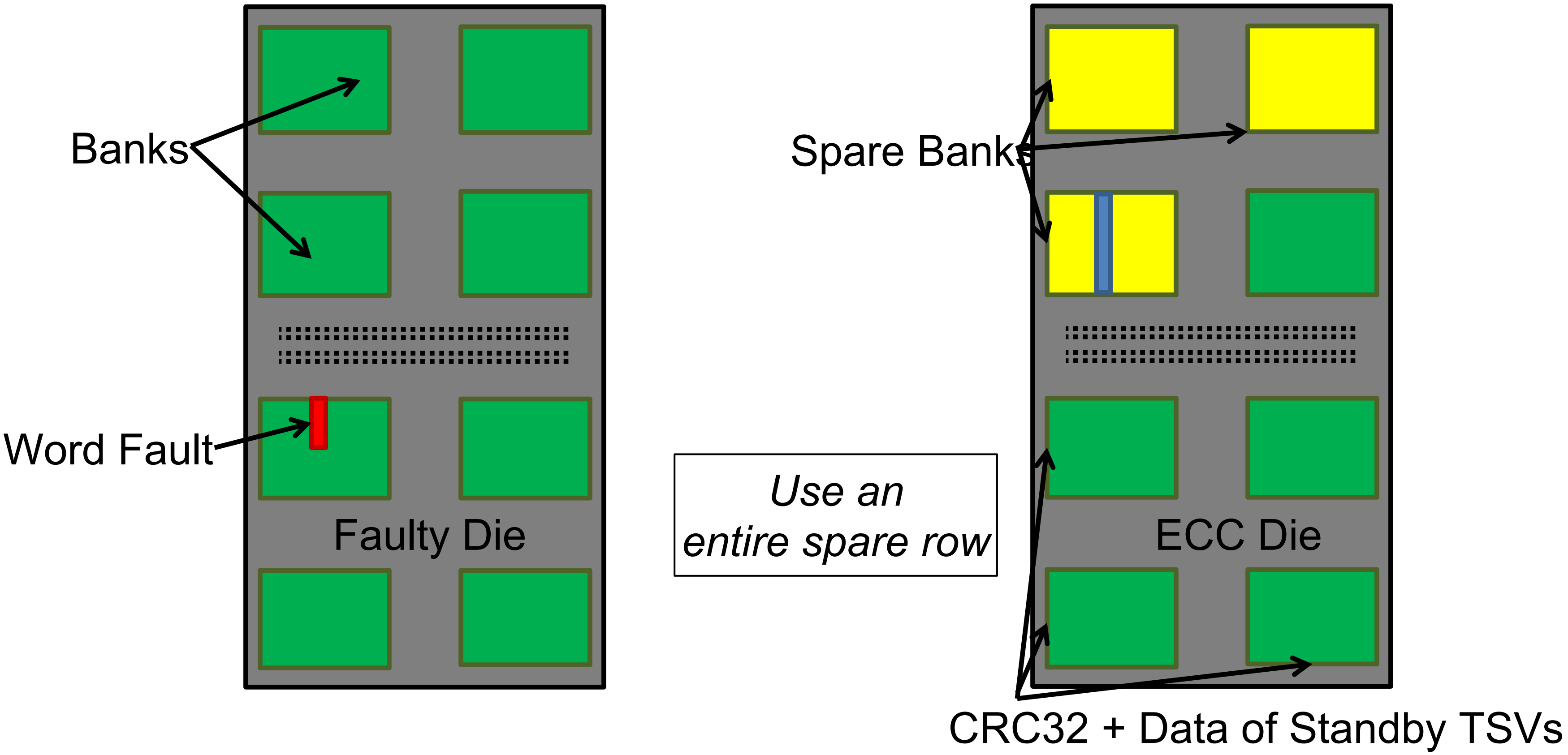
DYNAMIC DUAL GRAIN SPAIRING

- Provision Spare Area for Two Granularities



DYNAMIC DUAL GRAIN SPAIRING

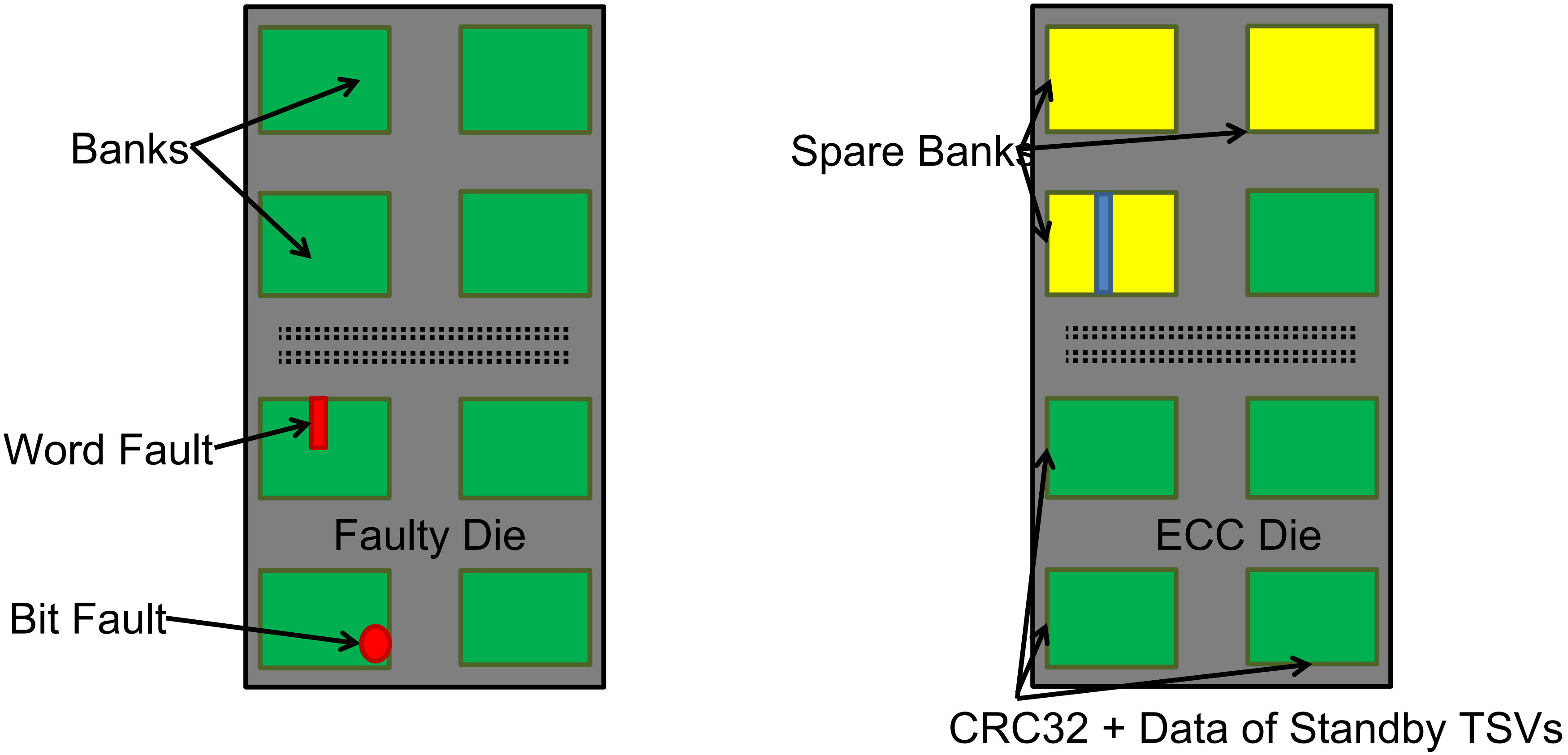
- Provision Spare Area for Two Granularities



Use an entire spare row

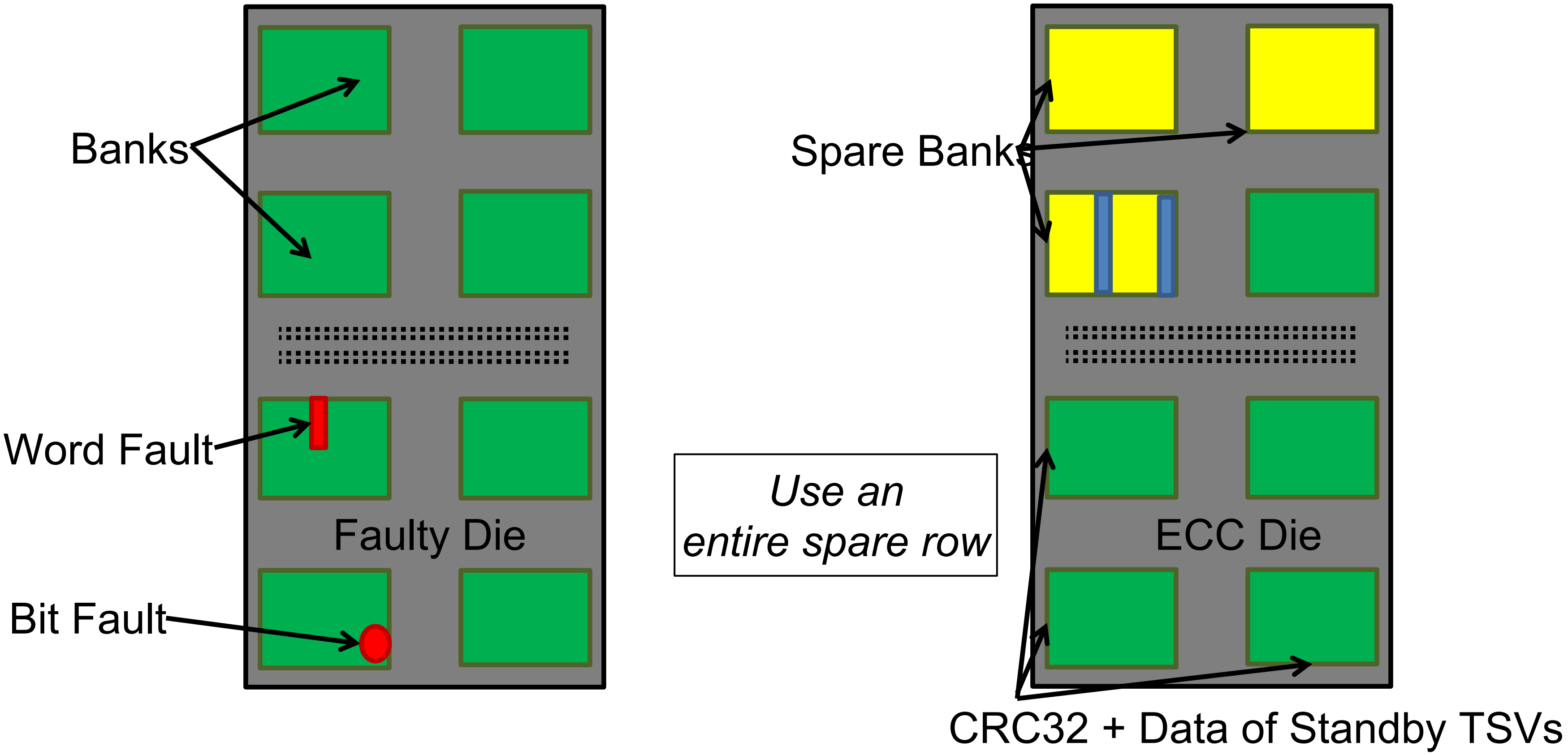
DYNAMIC DUAL GRAIN SPAIRING

- Provision Spare Area for Two Granularities



DYNAMIC DUAL GRAIN SPAIRING

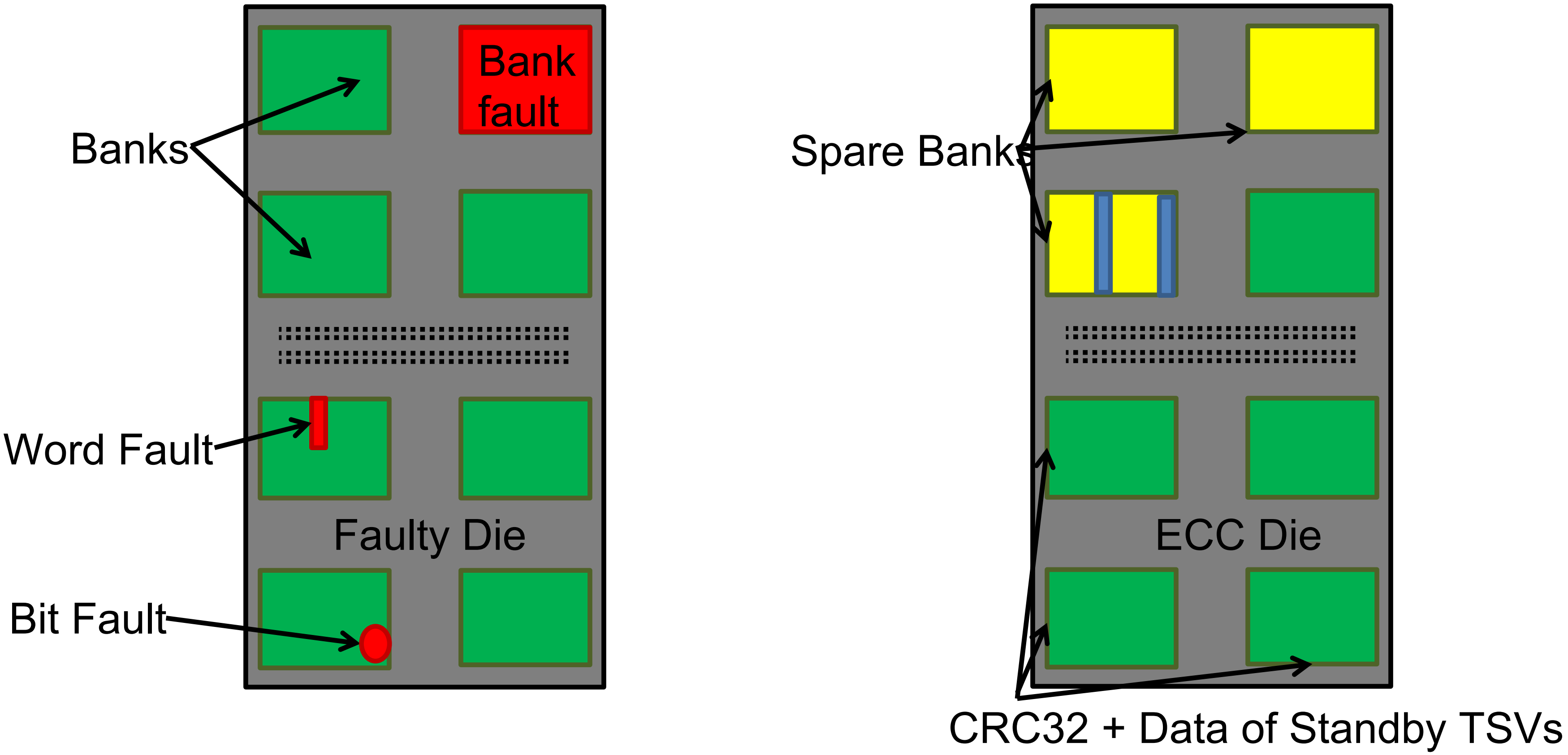
- Provision Spare Area for Two Granularities



Use an entire spare row

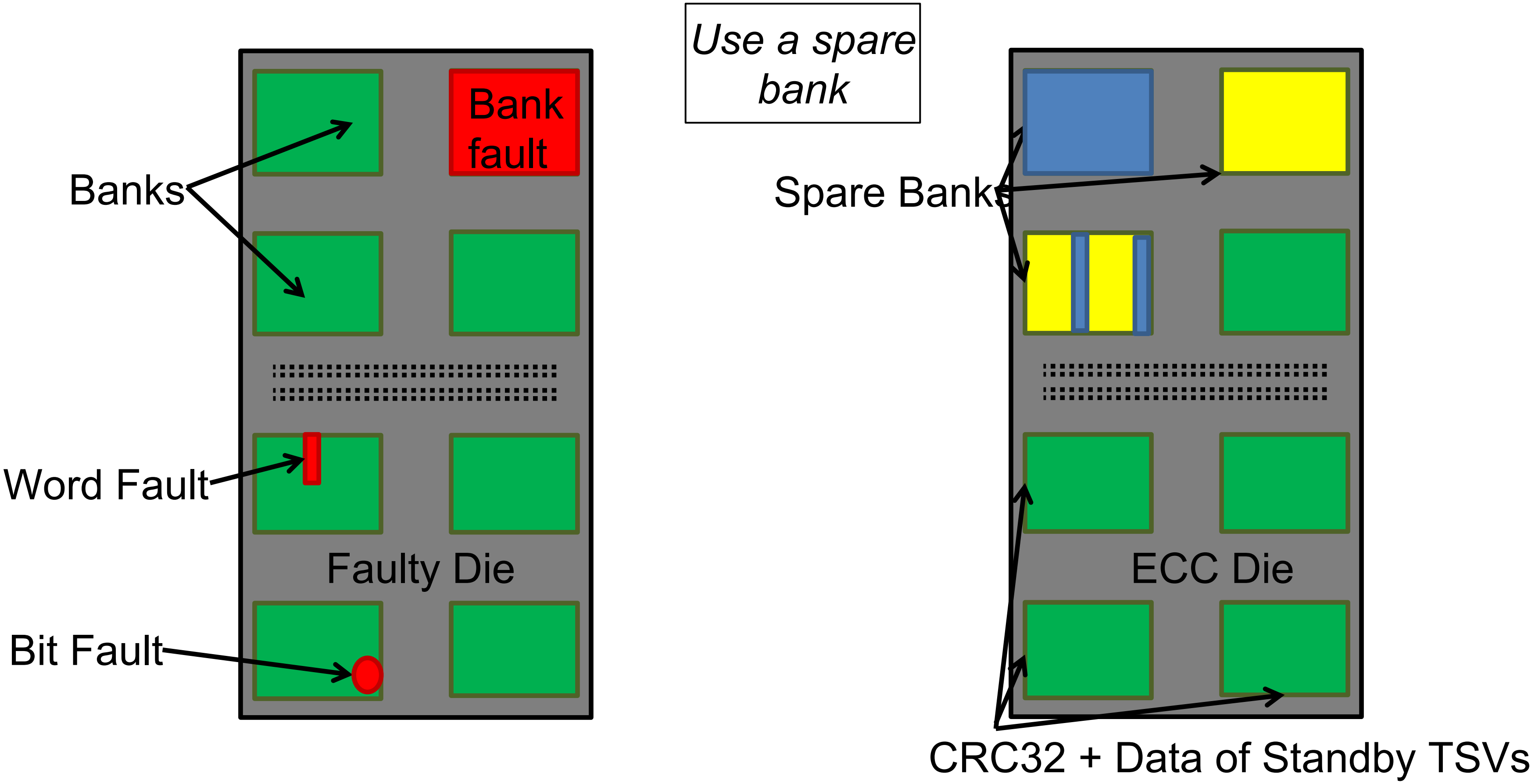
DYNAMIC DUAL GRAIN SPAIRING

- Provision Spare Area for Two Granularities



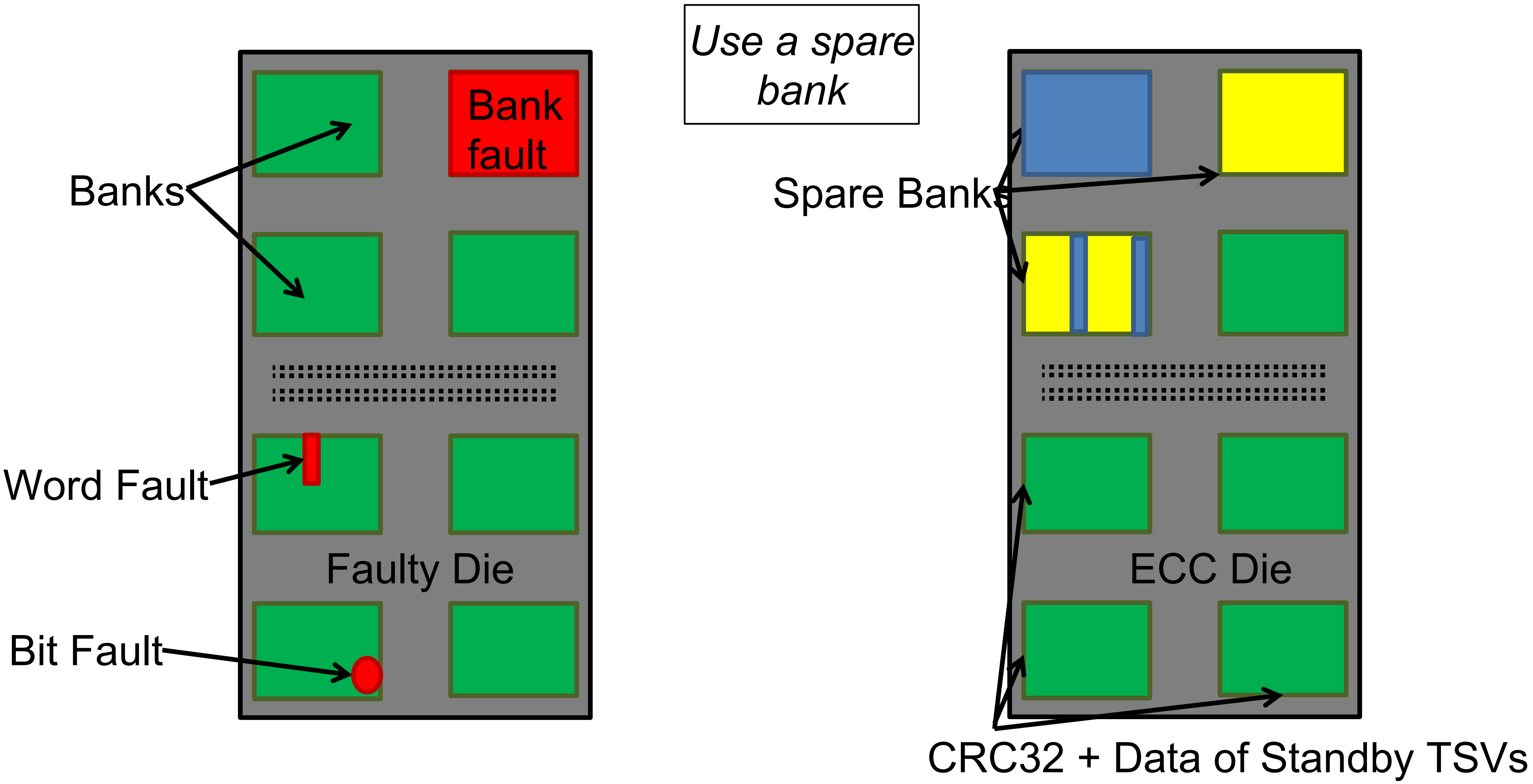
DYNAMIC DUAL GRAIN SPAIRING

- Provision Spare Area for Two Granularities



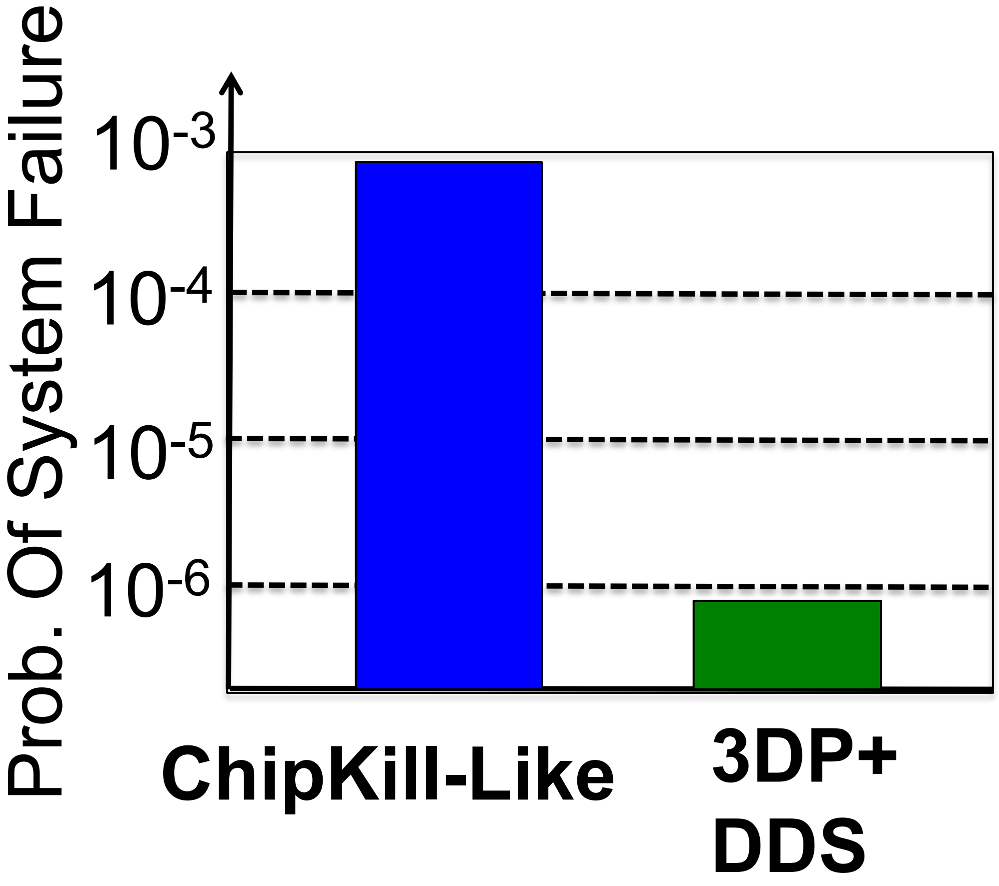
DYNAMIC DUAL GRAIN SPAIRING

- Provision Spare Area for Two Granularities

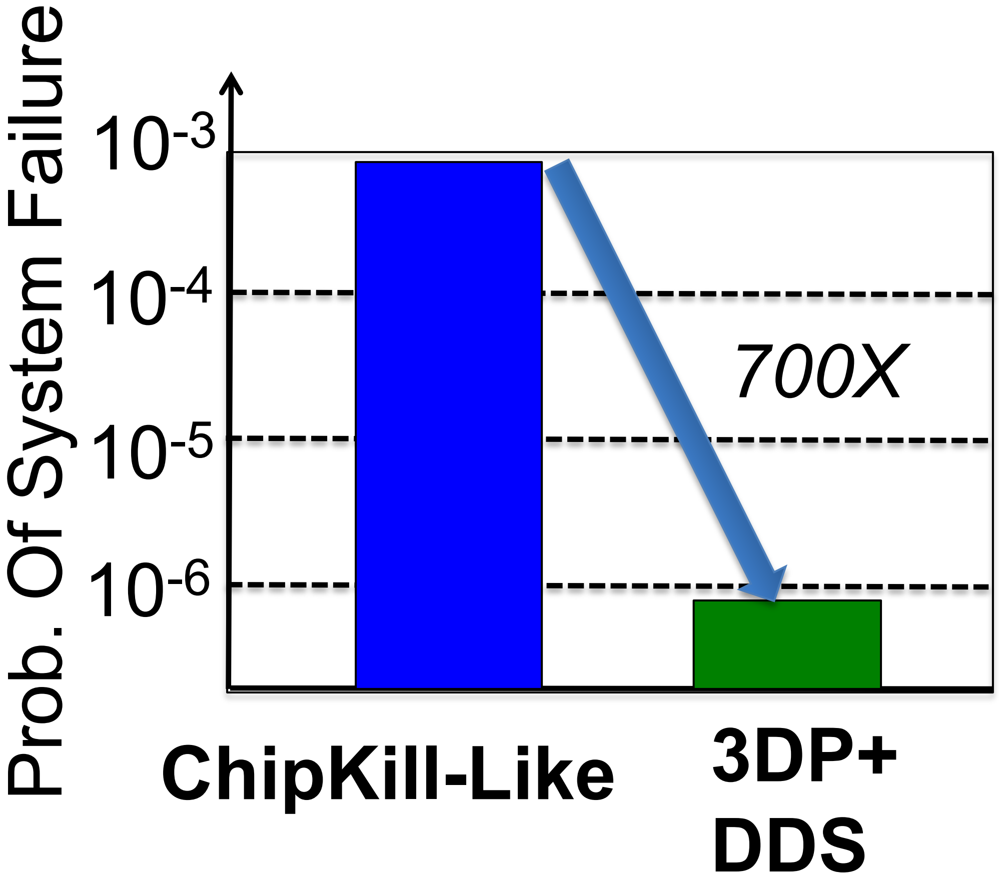


Dual Grain Sparing Efficiently Uses Spare Area

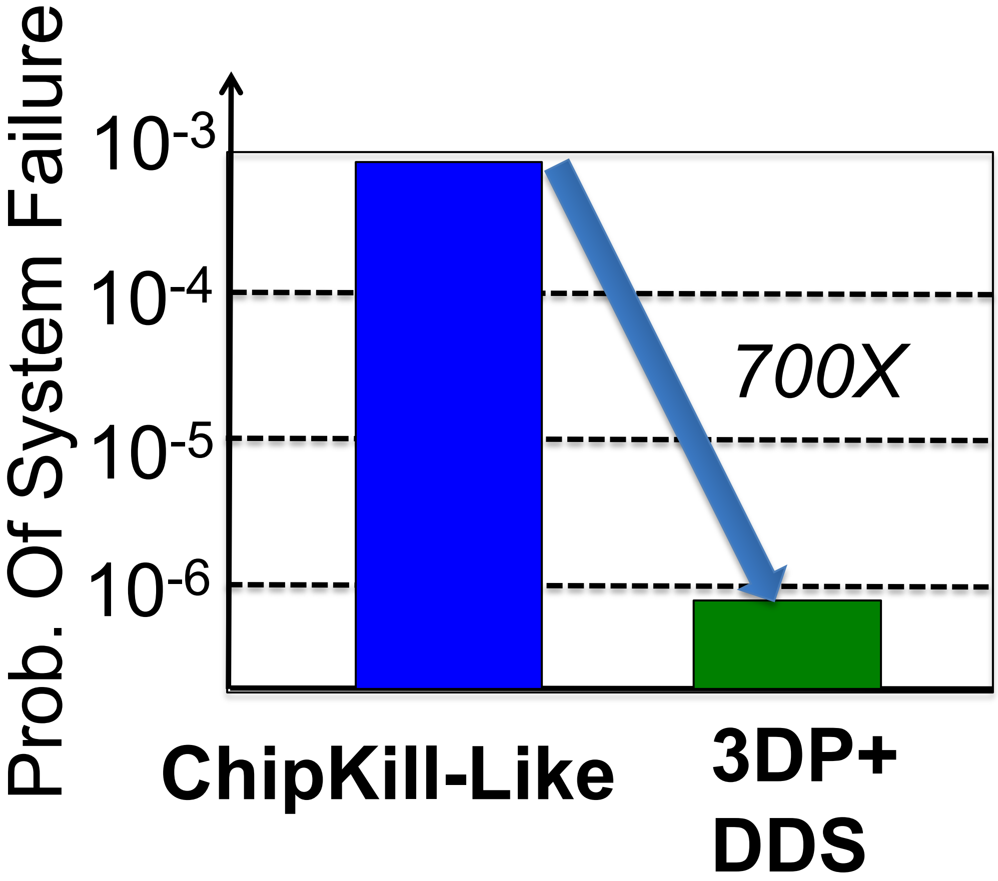
CITADEL: RESULTS



CITADEL: RESULTS



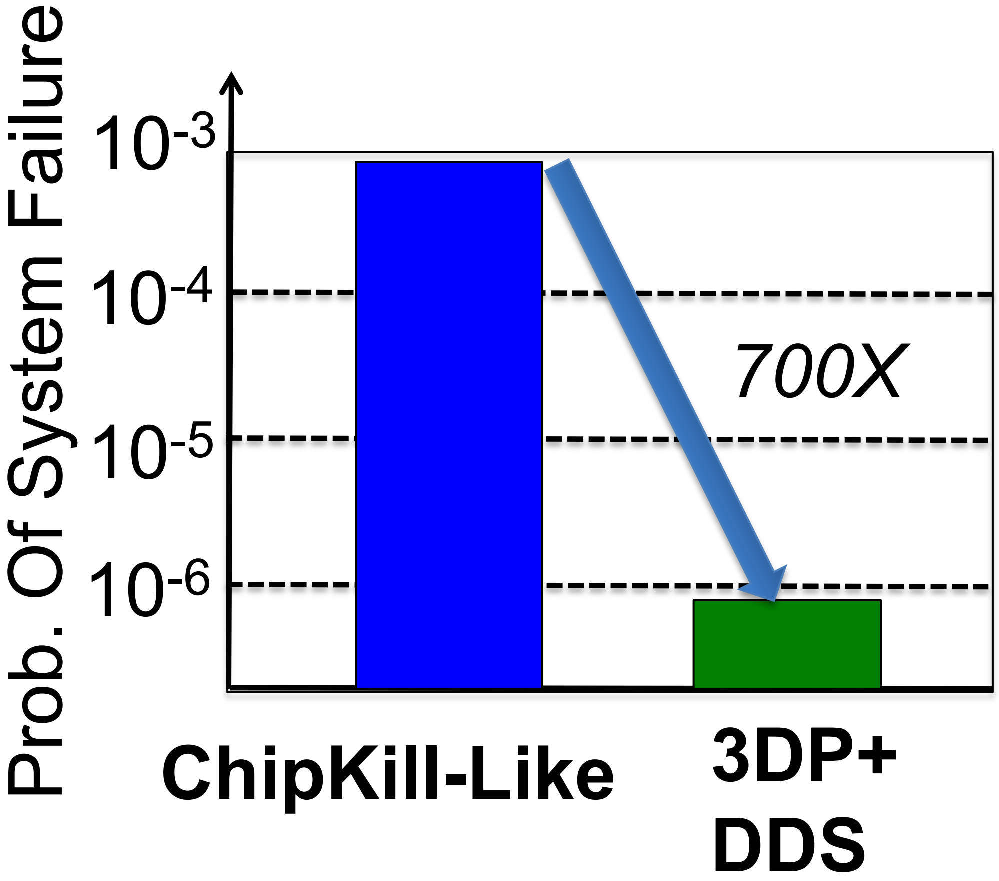
CITADEL: RESULTS



System: 8GB HBM @ DDR3-1600
Baseline: No Protection + Same Bank

Scheme	Slowdown	Active Power
ChipKill	1.25	3.8X
Citadel	1.01	1.04X

CITADEL: RESULTS



System: 8GB HBM @ DDR3-1600
Baseline: No Protection + Same Bank

Scheme	Slowdown	Active Power
ChipKill	1.25	3.8X
Citadel	1.01	1.04X

Citadel provides **700X** more resilience, consuming only 4% additional power and 1% additional execution time

OUTLINE

- Introduction and Background
- Citadel
- Scheme - 1 : TSV-SWAP
- Scheme - 2 : Three Dimensional Parity (3DP)
- Scheme - 3 : Dynamic Dual Grain Sparing (DDS)
- Summary ⇦

SUMMARY

- 3D stacking can enable high bandwidth DRAM
- Newer failure modes like TSV failures
- Striping data to protect against faults is costly

SUMMARY

- 3D stacking can enable high bandwidth DRAM
- Newer failure modes like TSV failures
- Striping data to protect against faults is costly
- Citadel enables robust and efficient 3D DRAM by:
  - TSV-SWAP runtime TSV SPARING

SUMMARY

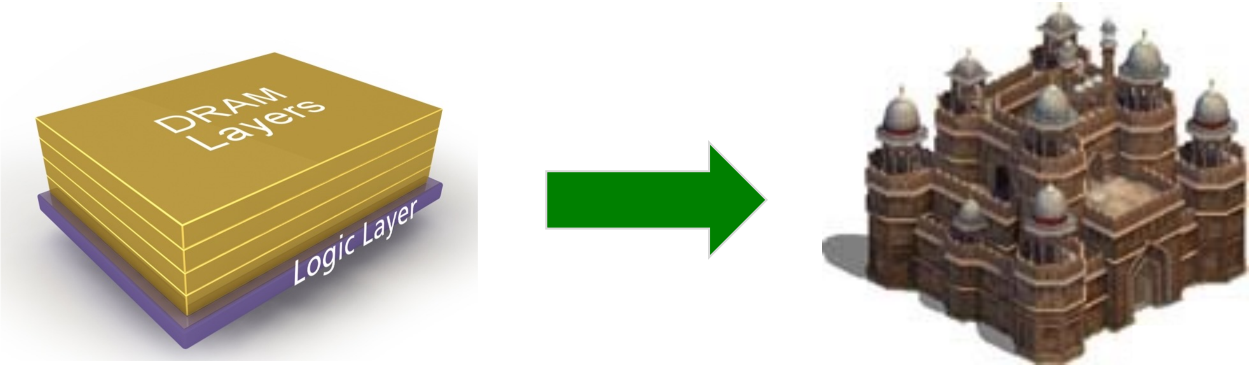
- 3D stacking can enable high bandwidth DRAM
- Newer failure modes like TSV failures
- Striping data to protect against faults is costly
- Citadel enables robust and efficient 3D DRAM by:
  - TSV-SWAP runtime TSV SPARING
  - Handling multiple-faults using 3DP

SUMMARY

- 3D stacking can enable high bandwidth DRAM
- Newer failure modes like TSV failures
- Striping data to protect against faults is costly
- Citadel enables robust and efficient 3D DRAM by:
  - TSV-SWAP runtime TSV SPARING
  - Handling multiple-faults using 3DP
  - Isolating faults using DDS

SUMMARY

- 3D stacking can enable high bandwidth DRAM
- Newer failure modes like TSV failures
- Striping data to protect against faults is costly
- Citadel enables robust and efficient 3D DRAM by:
  - TSV-SWAP runtime TSV SPARING
  - Handling multiple-faults using 3DP
  - Isolating faults using DDS
- Citadel provides all benefits of stacking at 700X higher resilience without the need for striping data



Thank You

Questions?

BACKUP SLIDES

CAUSES OF TSV FAULTS

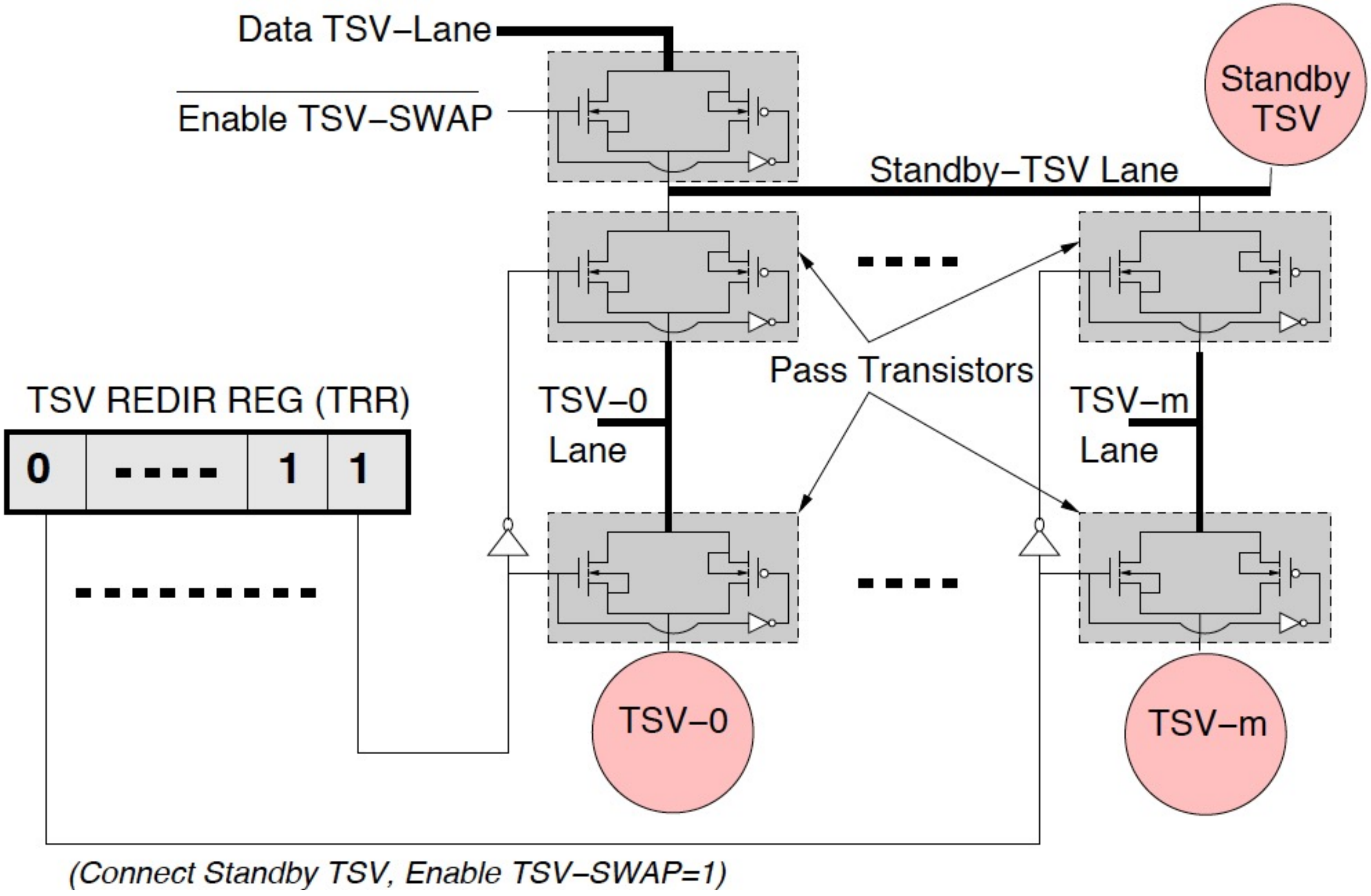
Recent papers[*+] shows that

1. TSVs prone to EM-induced voiding effects[*+]
2. Interfacial cracks → thermal-mechanical stress[*+]
3. EM-induced voids increase TSV resistance, causing path delay faults and TSV open defects[*+]
4. Micro-Bump faults[+]

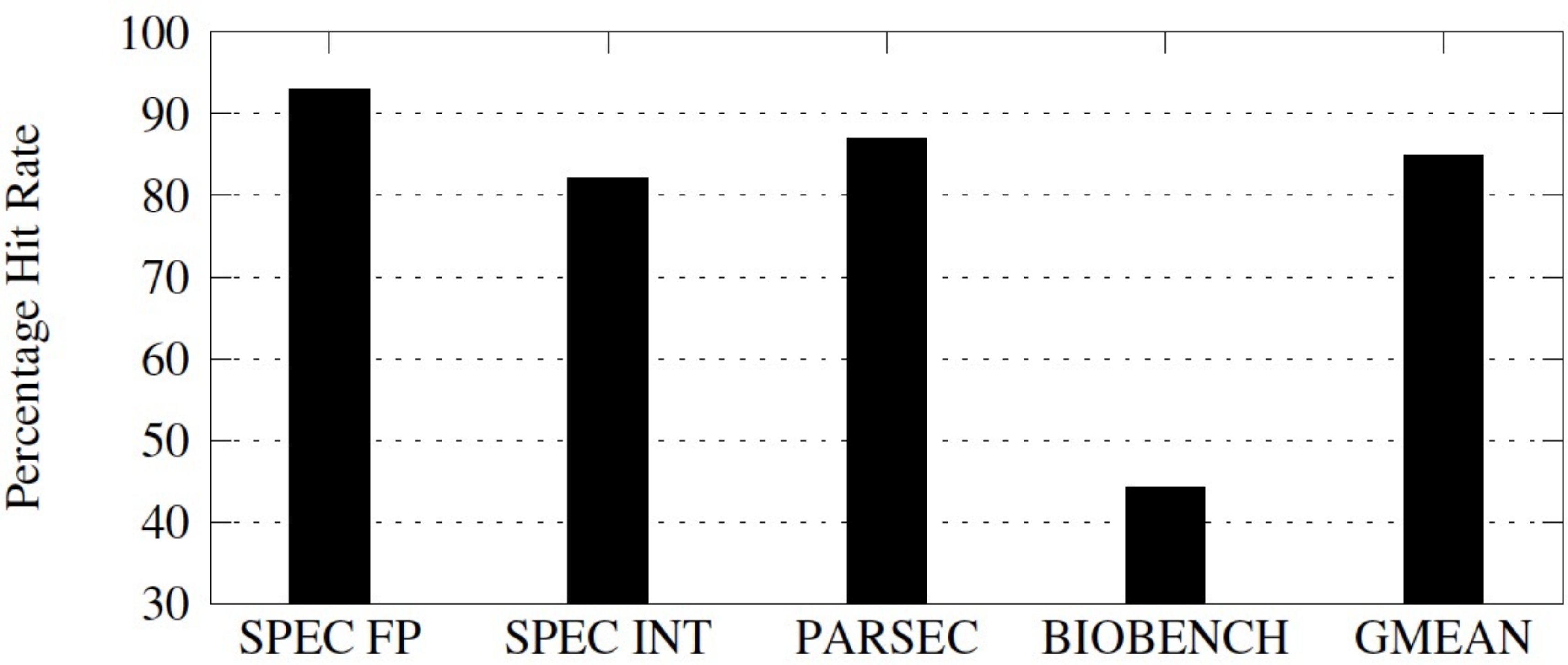
[*] Li Jiang et. al. [DAC 2013]

[+] Krishnendu C. et. al. [IRPS 2012]

TSV-SWAP REPAIR CIRCUIT



PARITY CACHE: HIT RATE



Benchmarks