

# Citadel: Efficiently Protecting Stacked Memory From Large Granularity Failures

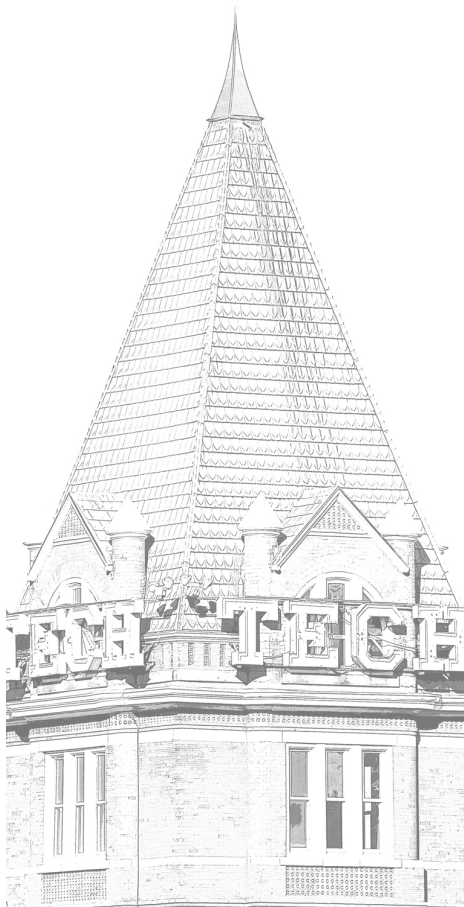
*Dec 15<sup>th</sup> 2014*

*MICRO-47 Cambridge UK*

*Prashant Nair - Georgia Tech*

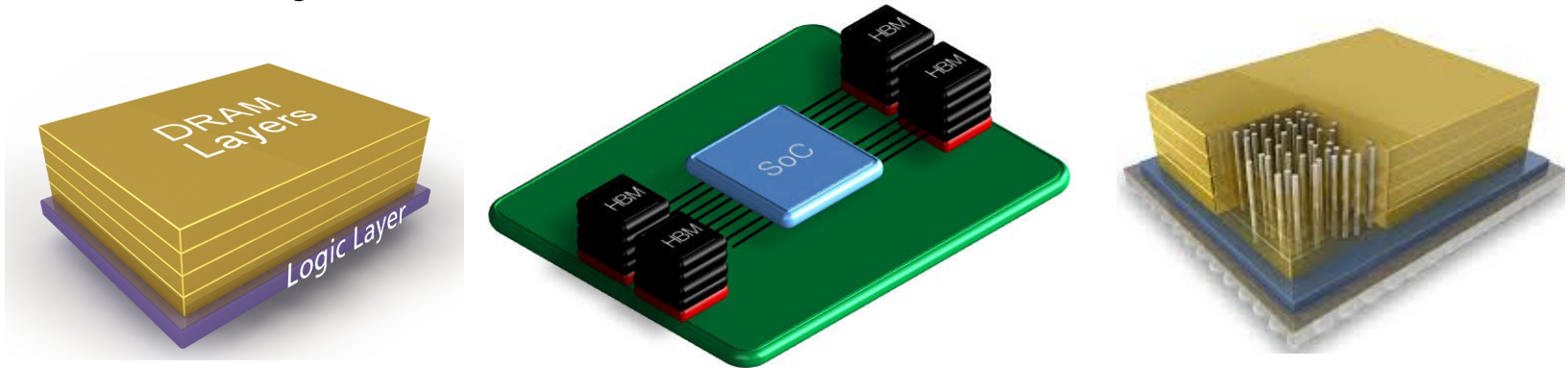
*David Roberts - AMD Research*

*Moinuddin Qureshi - Georgia Tech*



# INTRODUCTION TO 3D DRAM

- DRAM systems face a bandwidth wall

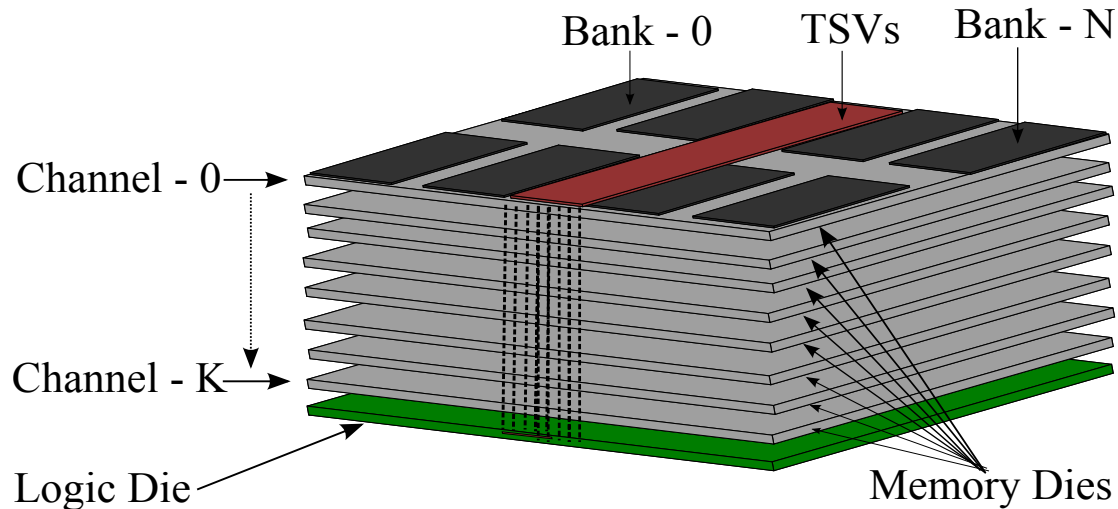


- Stack DRAM Dies over each other → 3D DRAM
- Use Through Silicon Vias (TSV) to connect Dies
- Higher density of TSV → Higher Bandwidth

Go 3D to Scale Bandwidth Wall

# FAILURES IN 3D DRAM

- 3D DRAM → Communicate using TSVs

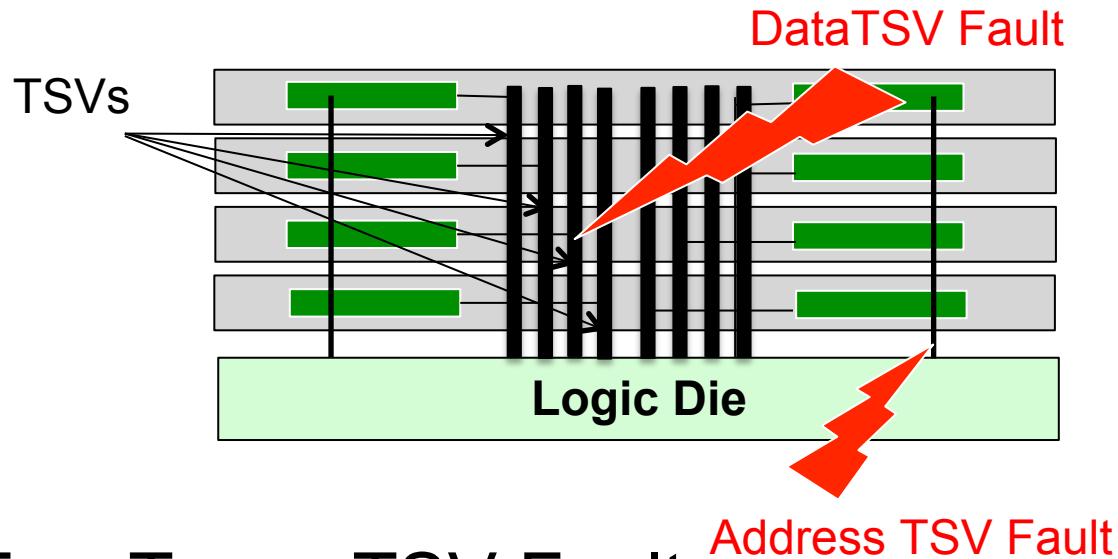


- A New Failure Mode: TSV Failures
- TSV Failures → Large Granularity Failures

TSVs Present New Kind of Large Granularity Failures

# A NEW FAILURE MODE FROM TSVs

TSVs conduit for Address and Data

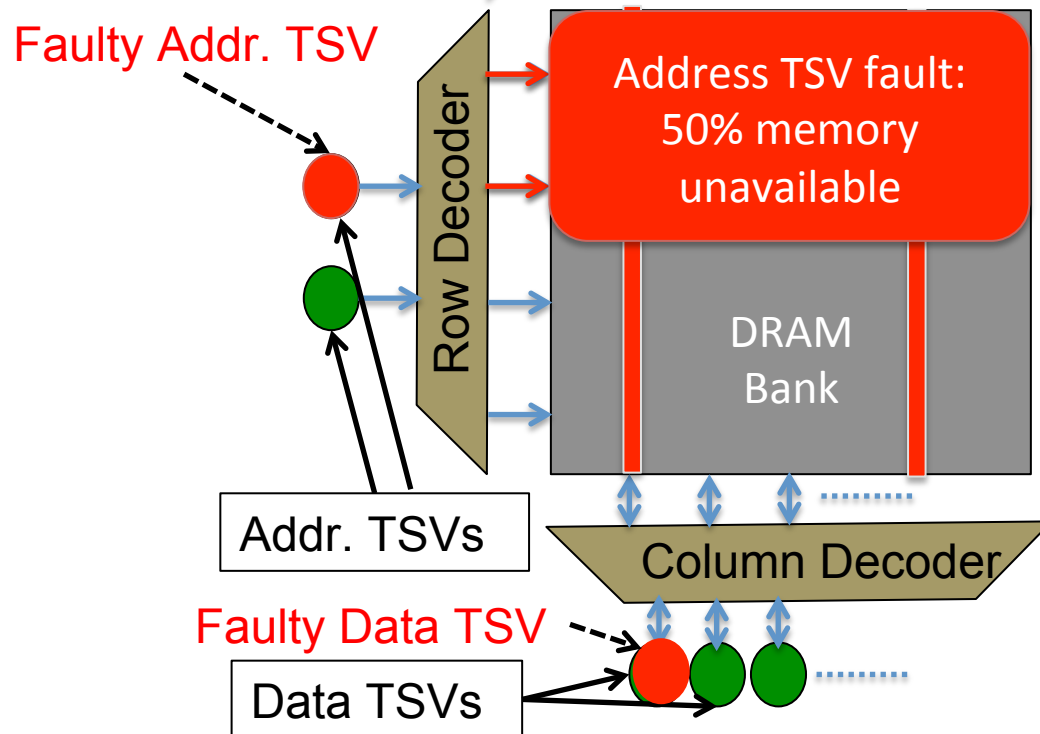


- Mainly Two Types TSV Faults
  - Data (Incorrect Data fetched from DRAM Die)
  - Address (Incorrect address presented to DRAM Die)

TSV Faults cause unavailability of Data and Addresses

# EFFECT OF TSV FAULTS

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

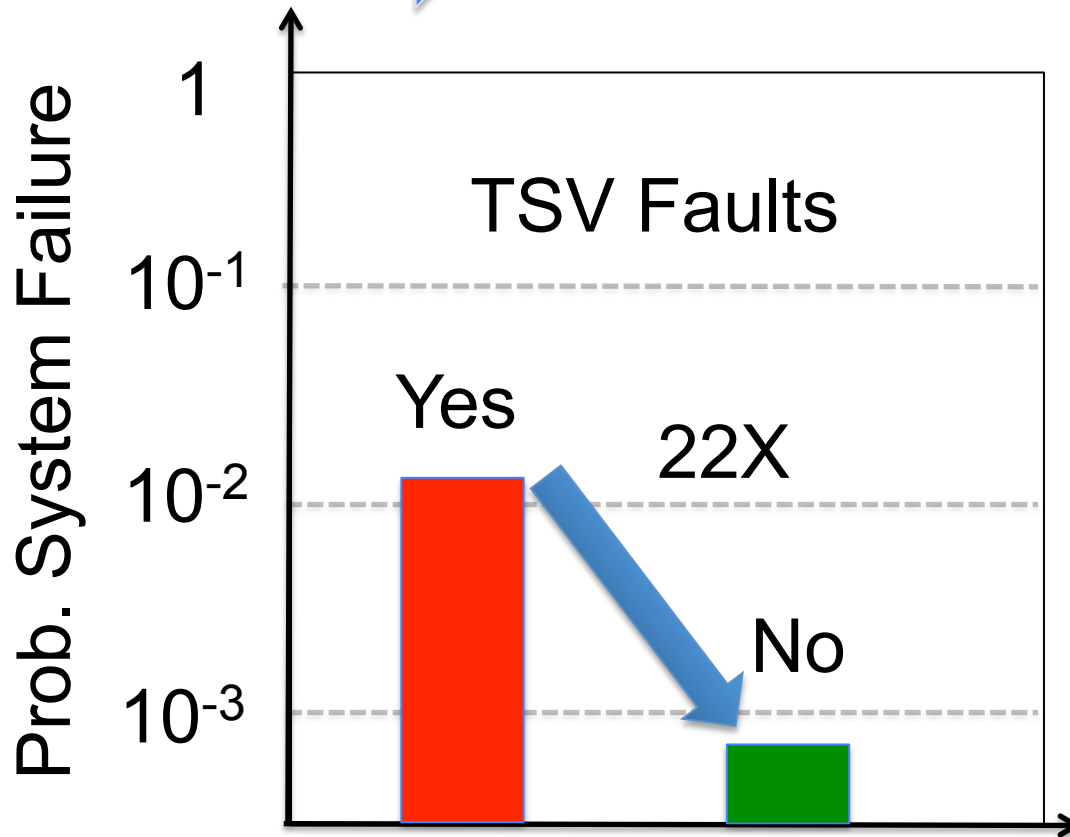


TSVs can cause failures at multiple granularities

# IMPACT OF TSV FAULTS

System: 8GB Stacked Memory (HBM)

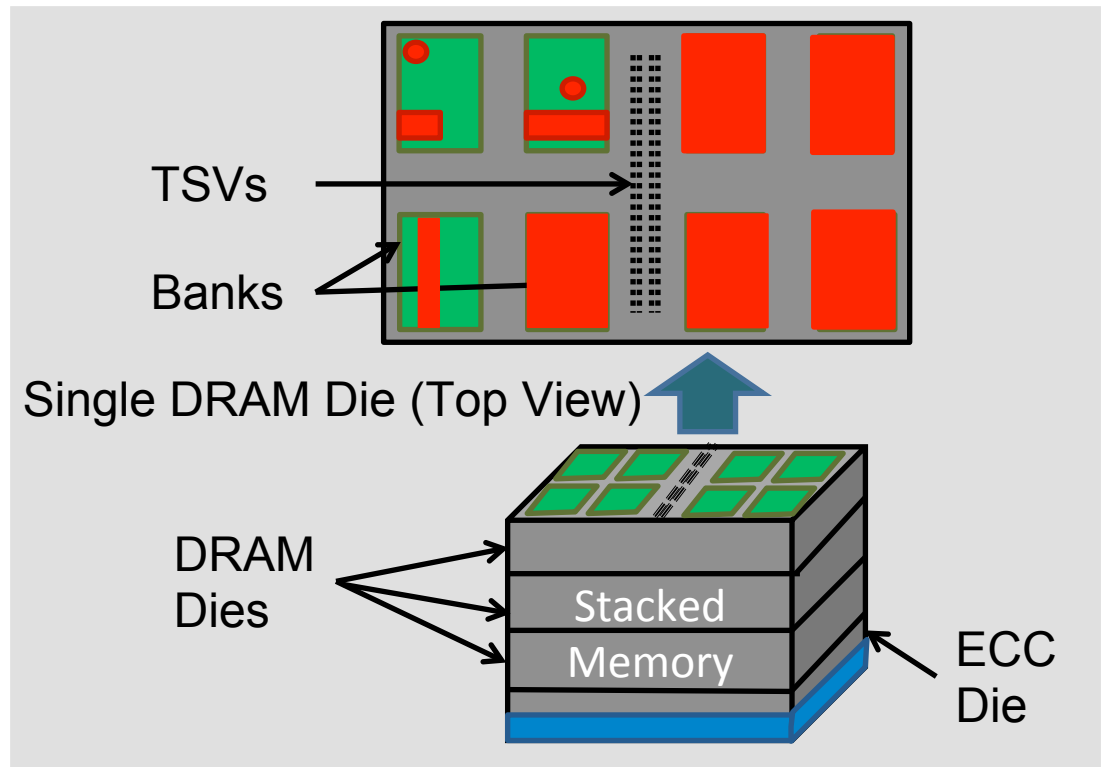
Prob. System Failure  $\rightarrow$  Prob(Uncorrectable Error)



Efficient Techniques to Mitigate TSV Faults

# 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

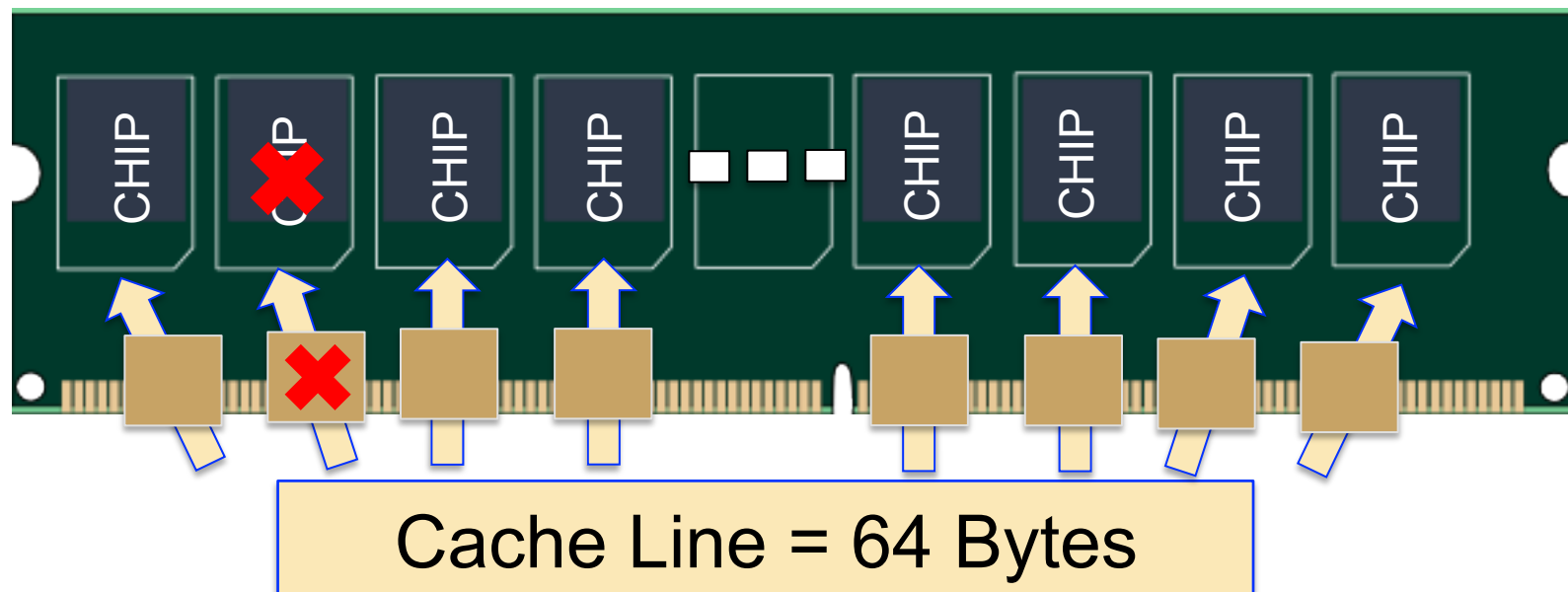
Die Failure Mode	* Permanent Fault Rate (FIT)	
Bit	148.8	✓ <i>SECDED</i>
Word	2.4	} = 125.7
Column	10.5	
Row	32.8	
Bank	80	

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



# CONVENTIONAL SCHEMES

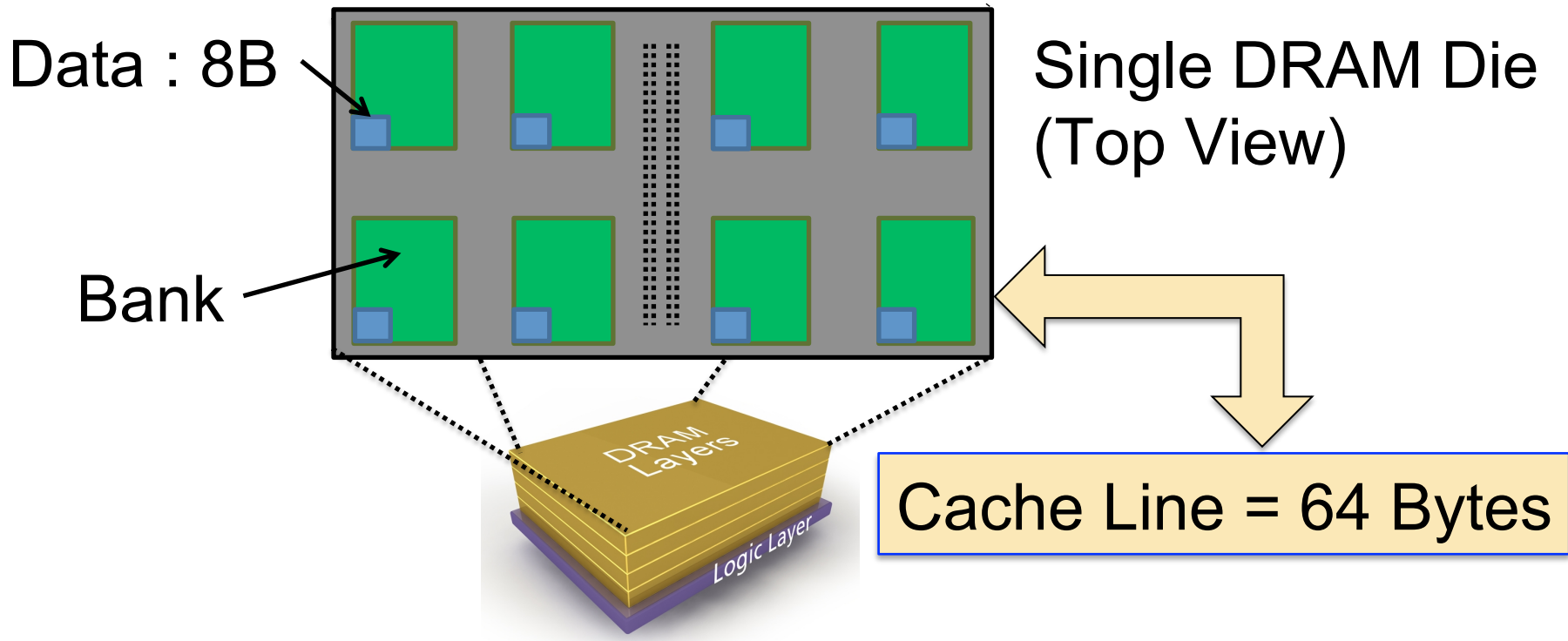
Current Systems Naturally Stripe Data Across Chips



- ChipKill : Mitigate Large Failures (Whole Chip)

ChipKill relies on data striping to tolerate large granularity failures

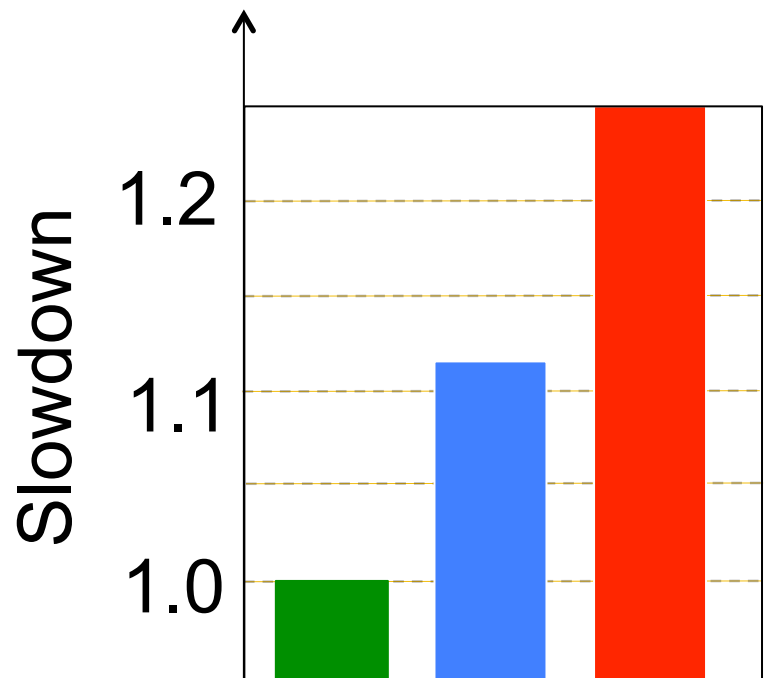
# CHIPKILL IN STACKED MEMORY



- A request activates at least 8 Banks or 8 Channels

At least 8X  activation power, 8X  DRAM parallelism

# COST OF STRIPING IN 3D DRAM



25% More Execution Time

Striping data across banks/channels in 3D is costly

# GOAL

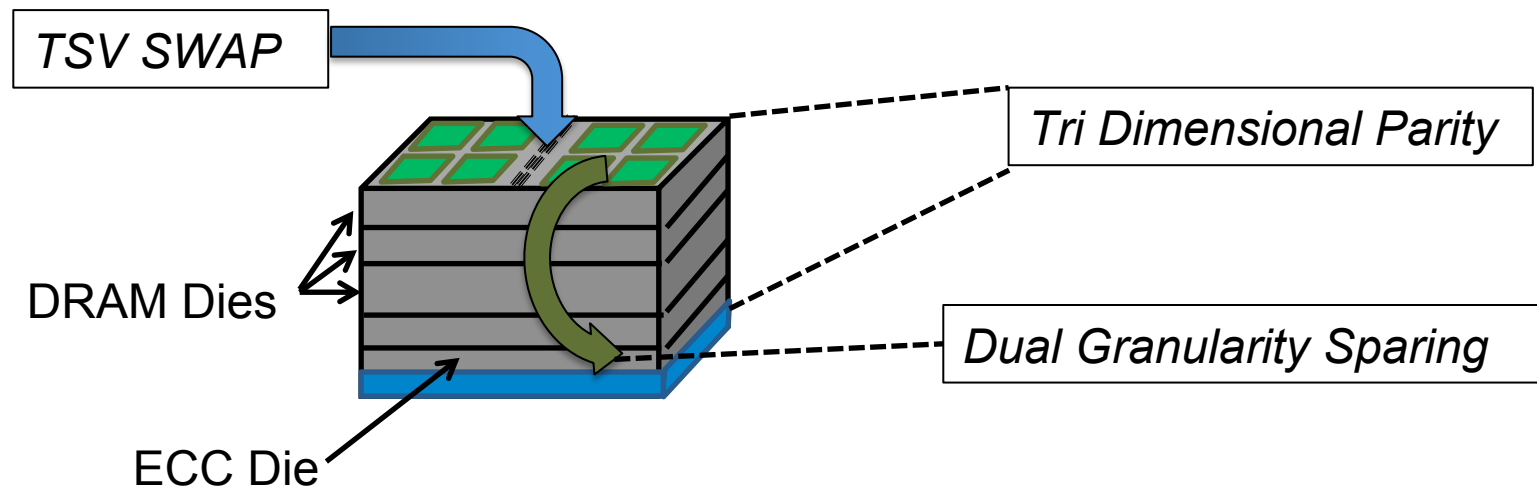
*Develop Efficient Solutions to Mitigate TSV and other Large Granularity Faults in Stacked Memory without striping data*

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



Enable robust stacked memory at very low overheads

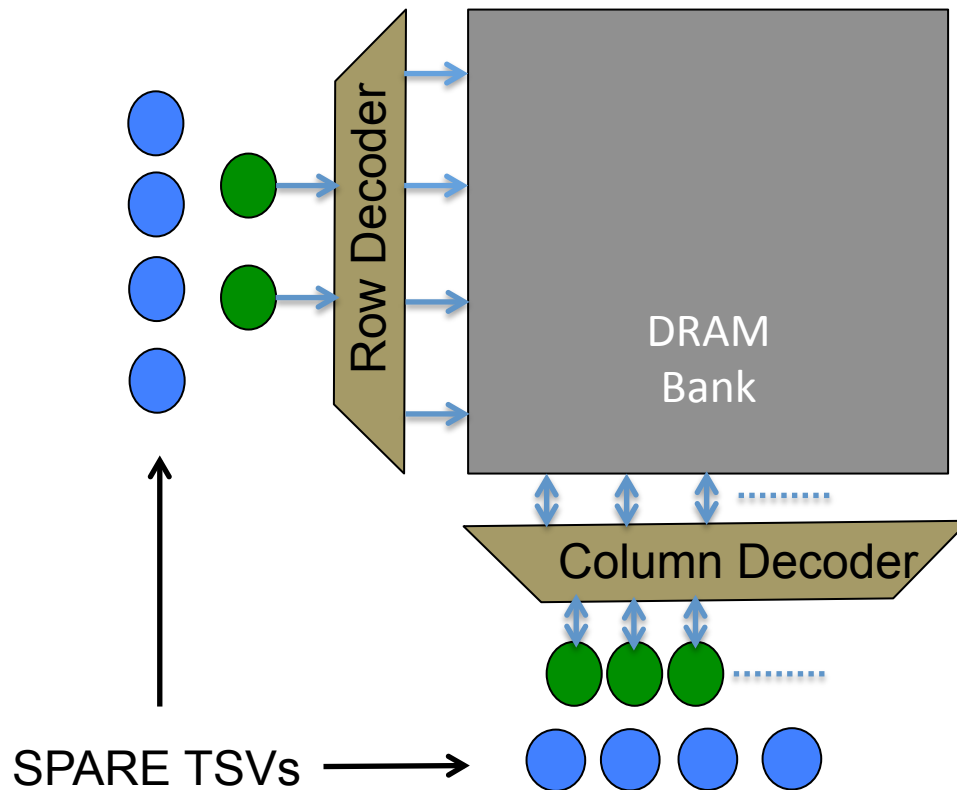
# OUTLINE

- 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

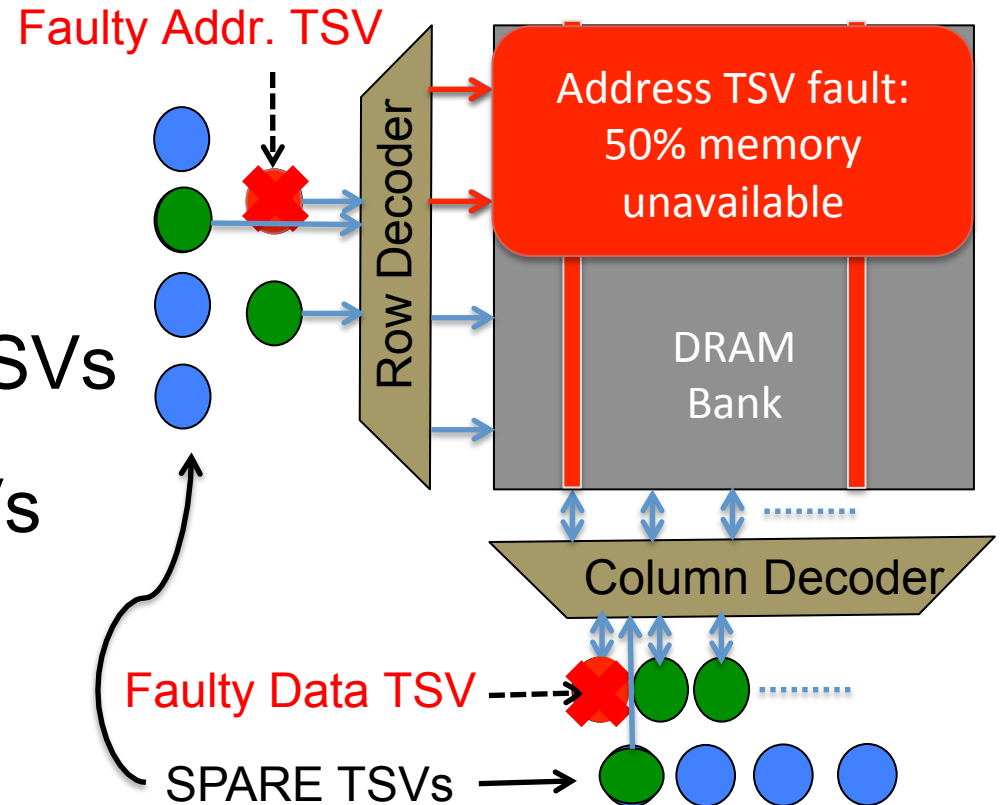


Additional Spare TSVs can replace faulty TSVs



# 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

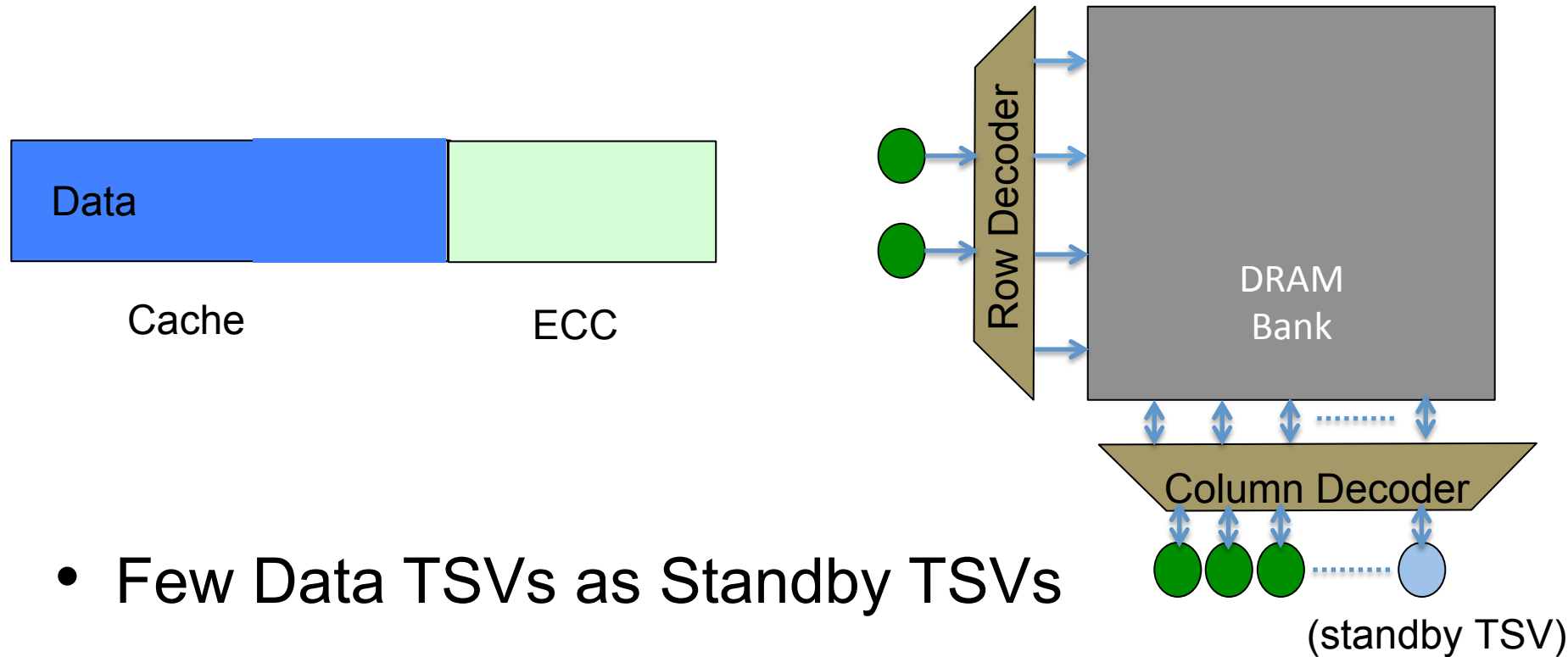
# DESIGN-TIME TSV SPARING: PROBLEMS

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



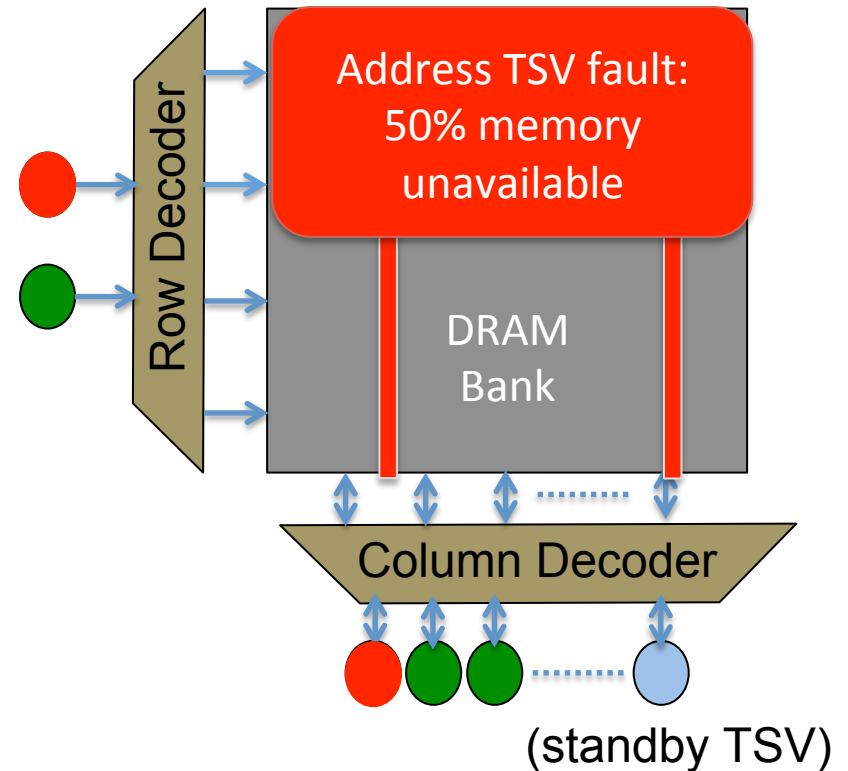
- Few Data TSVs as Standby TSVs
- Replicate Standby Data in ECC

Data TSVs reused as Standby TSVs

# TSV-SWAP: RUNTIME TSV SPARING

## STEP-2: DETECTING FAULTY TSVs

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

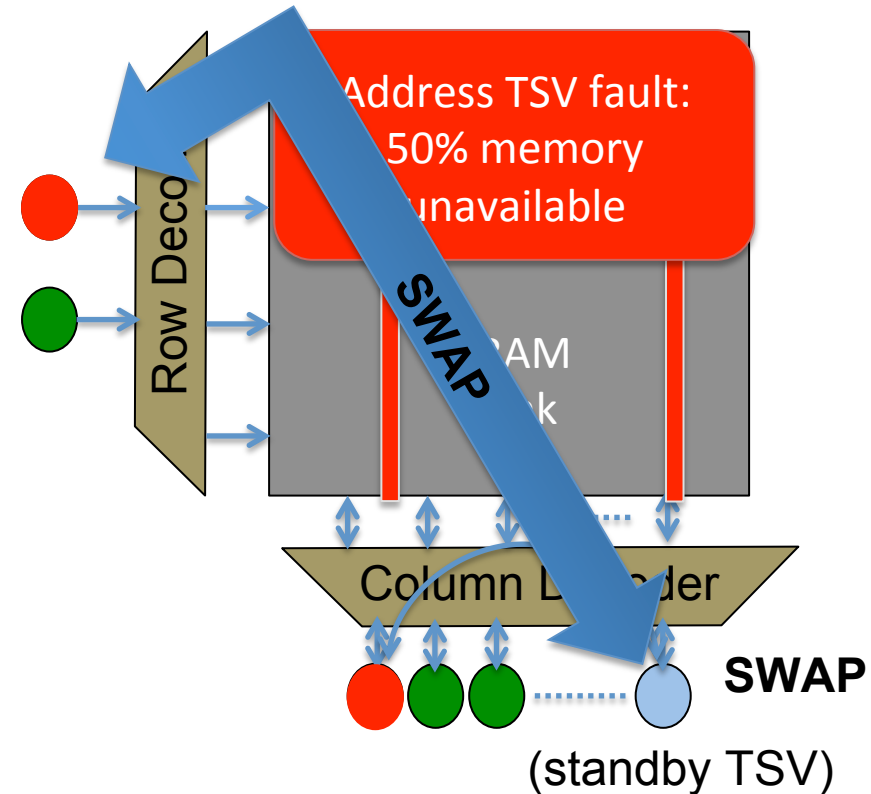


Data vs Address TSV Faults Using CRC-32+BIST

# TSV-SWAP: RUNTIME TSV SPARING

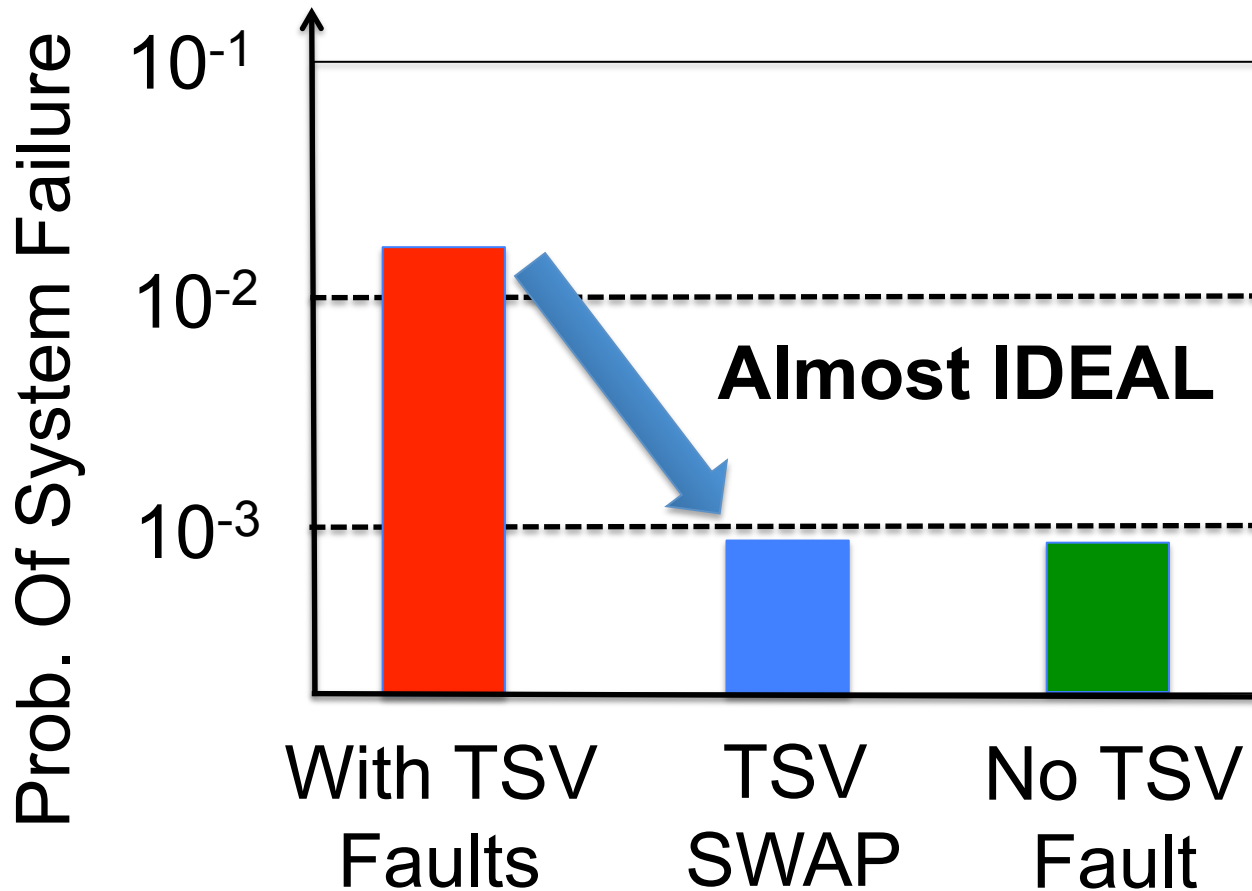
## STEP-3: REDIRECTING FAULTY TSVs

Swap Faulty TSVs with Standby TSVs at runtime



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

# EFFECTIVENESS OF TSV-SWAP



Rate: One  
TSV Fault  
Every 7 years

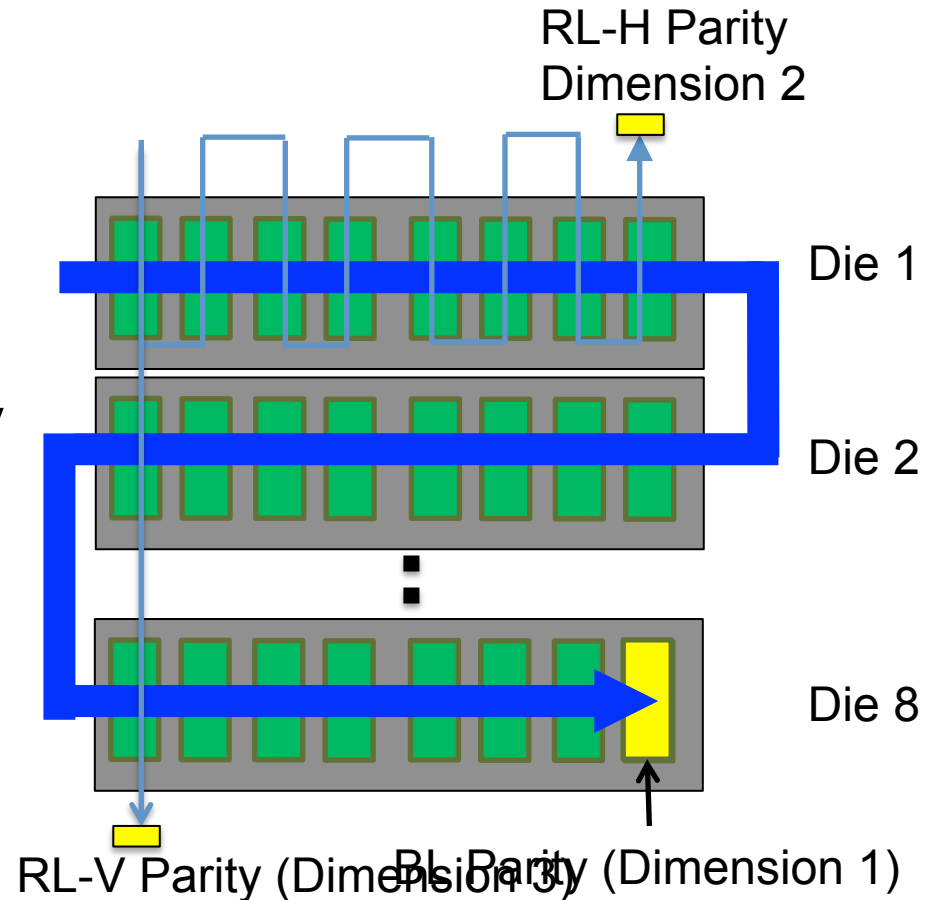
TSV-SWAP is Effective at Tolerating TSV Faults

# 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
- 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

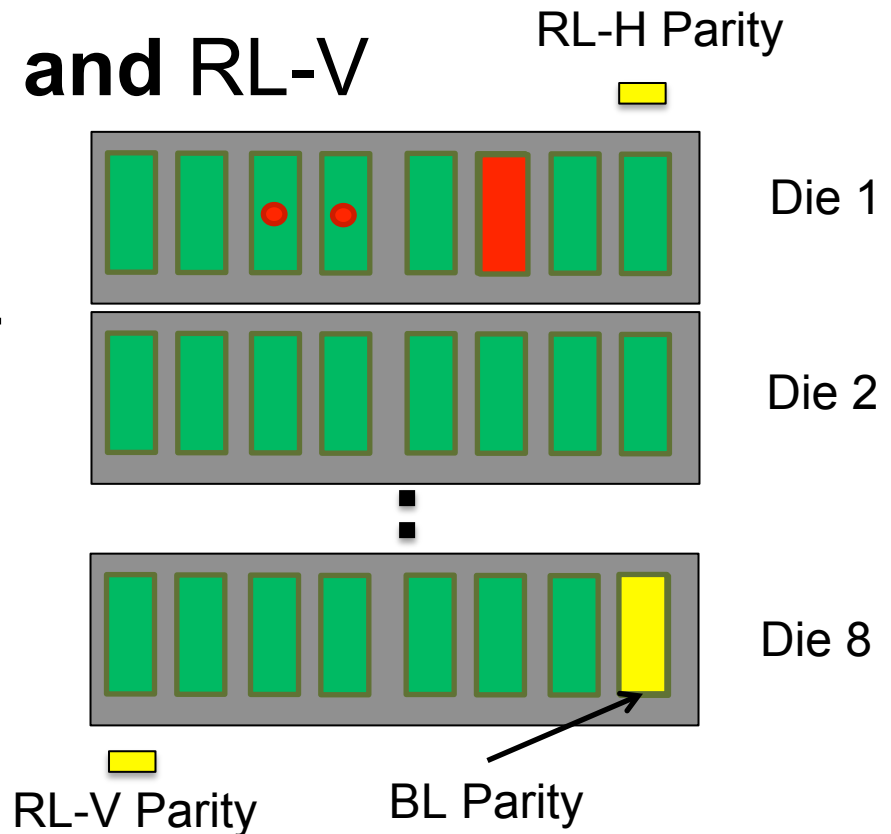


# 3DP: DATA CORRECTION

If Fault → Compute Parity and Correct

- 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

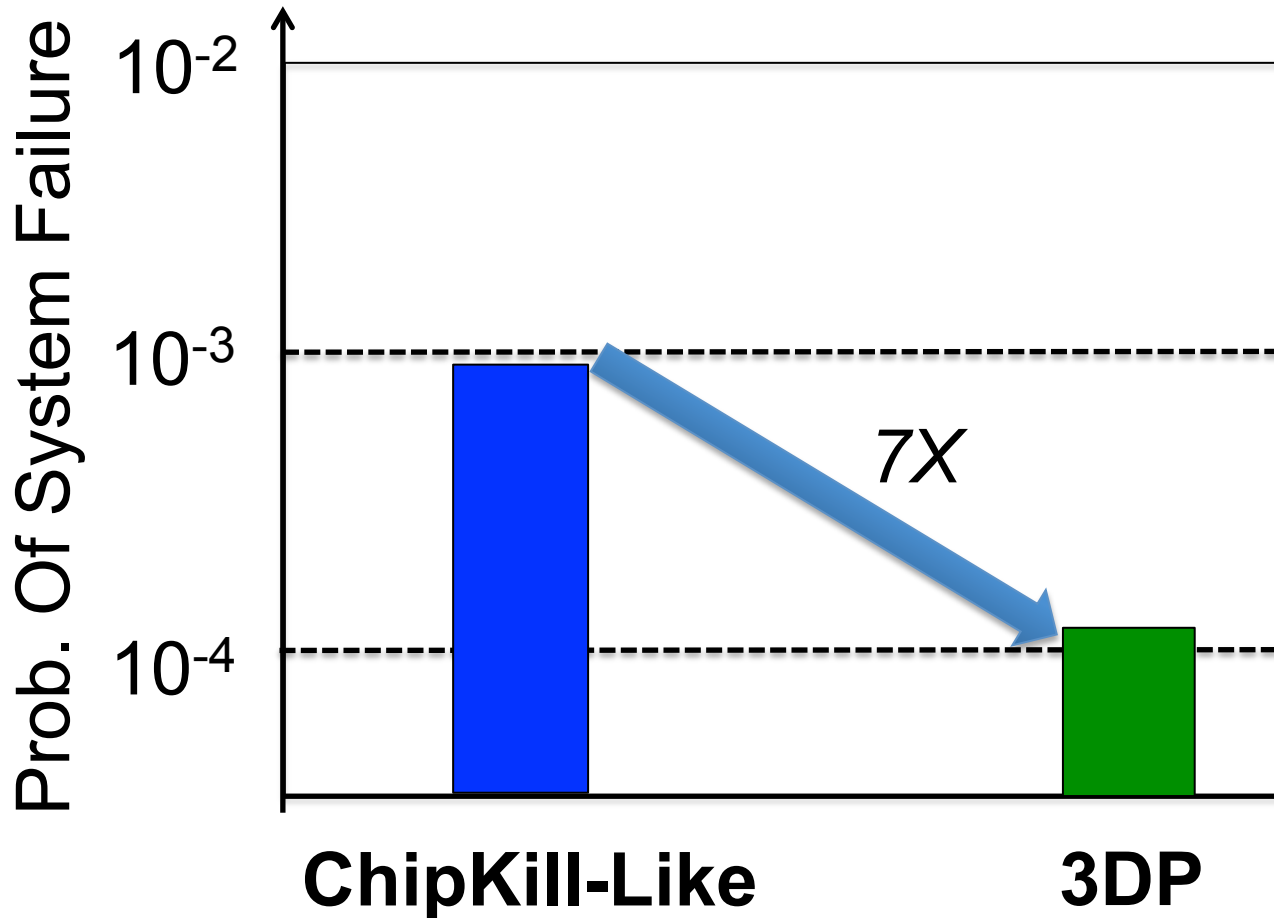


# OVERHEADS IN UPDATING PARITY

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



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

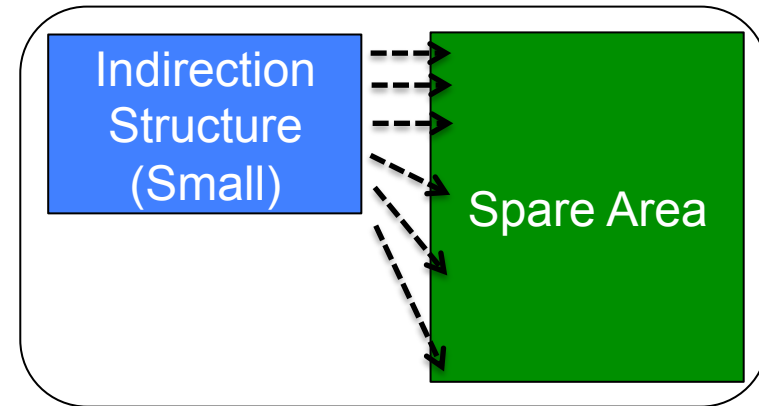
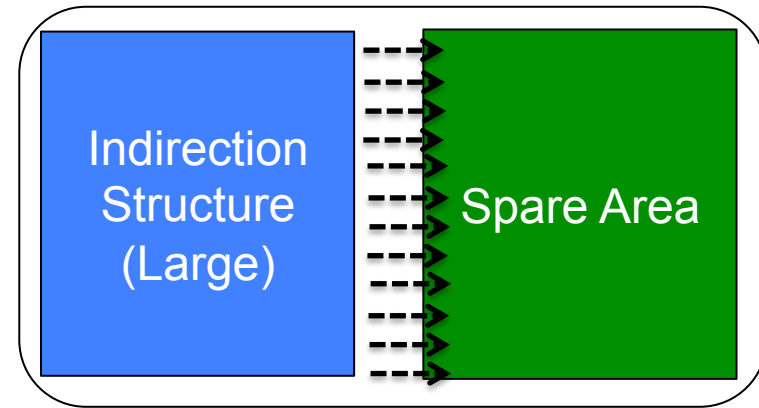
# WHY SPARE FAULTY DATA?

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

Sparing Mitigates Performance Overheads and  
Enhances Reliability

# TRACKING STRUCTURES IN SPARING

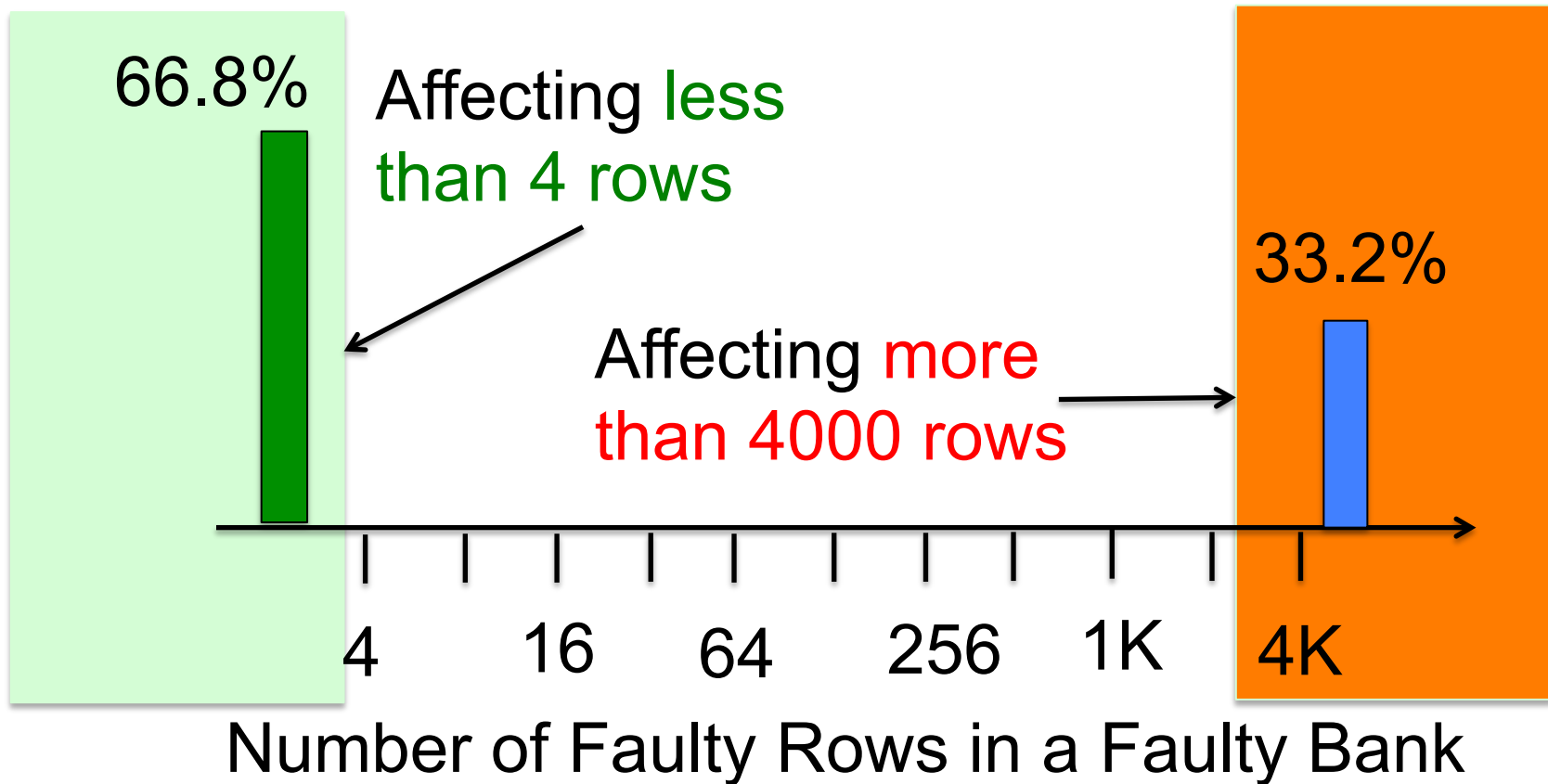
- Row Level Tracking
  - Large Indirection Structure
  - Sparing Area Used Efficiently
- Bank Level Tracking
  - Small Indirection Structure
  - Sparing Area Used Inefficiently



Ideally We Need Small Indirection Structures  
Which Use Spare Area Efficiently

# BIMODAL FAILURES

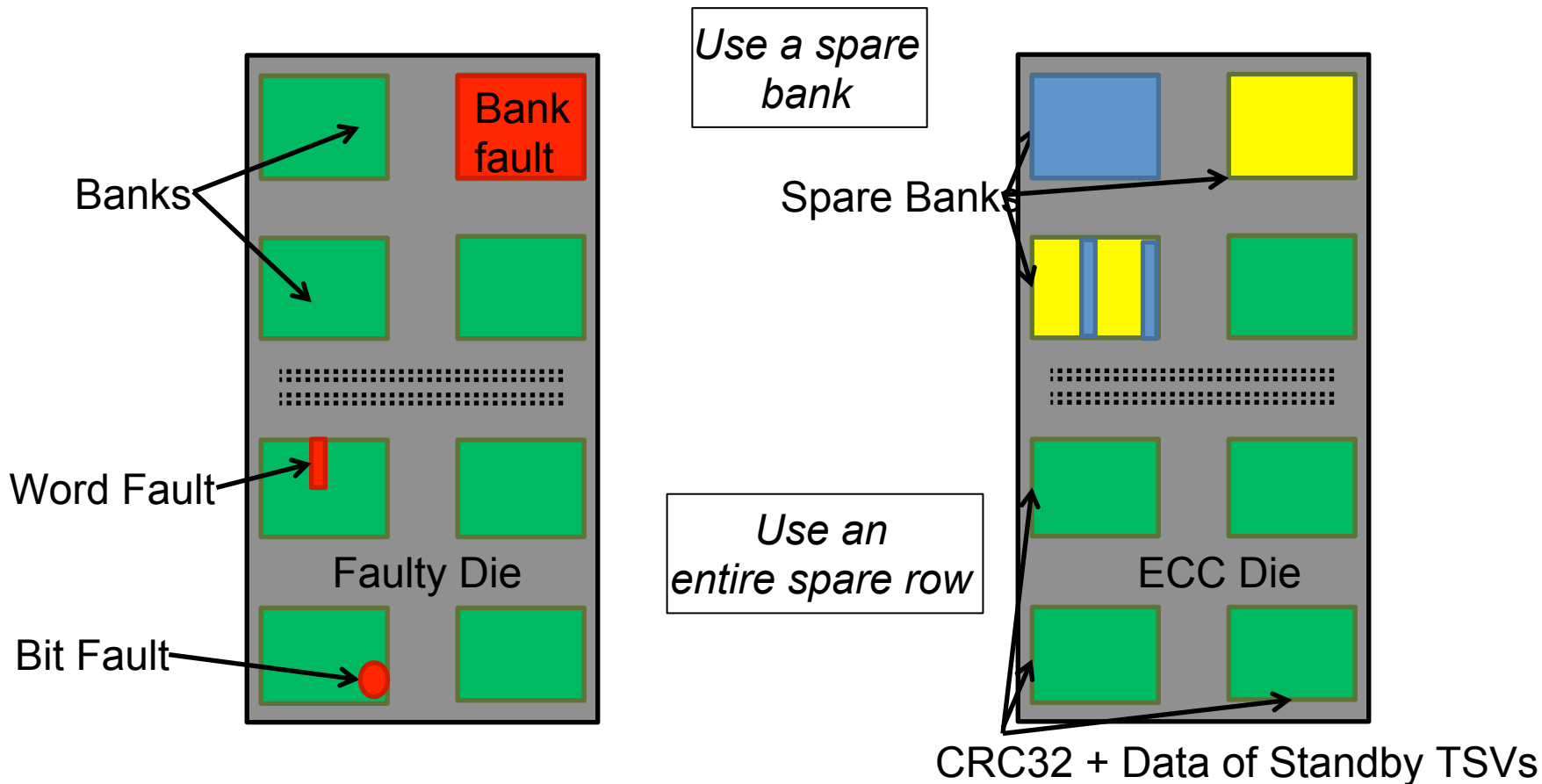
- **Observation** : Either  $< 4$  or  $> 4000$  row failures



Spare Faulty Regions At Two Granularities

# DYNAMIC DUAL GRAIN SPAIRING

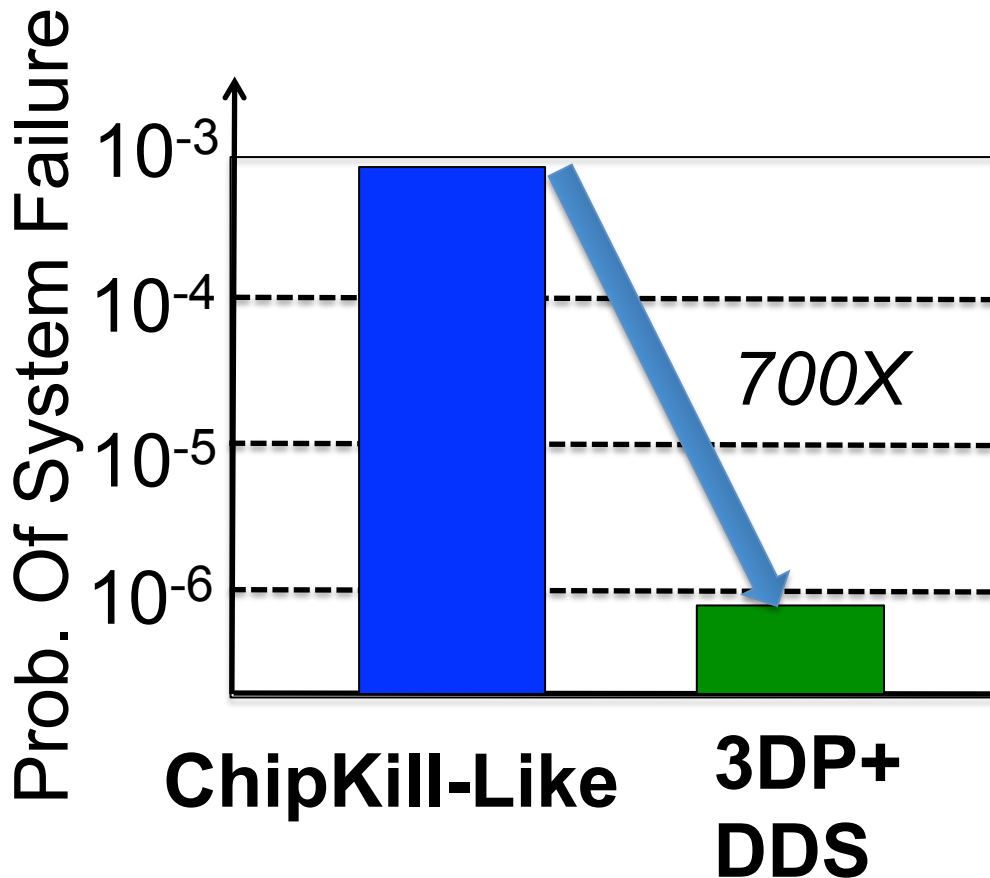
- Provision Spare Area for Two Granularities



Dual Grain Sparing Efficiently Uses Spare Area



# 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

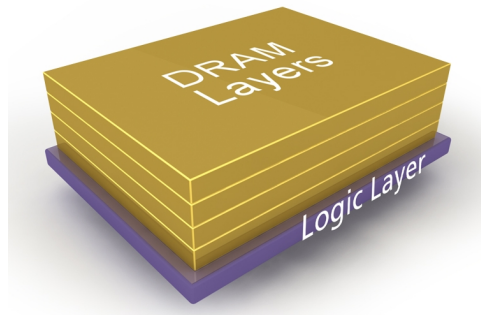
Citadel provides **700X** more resilience, consuming only 4% additional power and 1% additional execution time

# 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
- 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

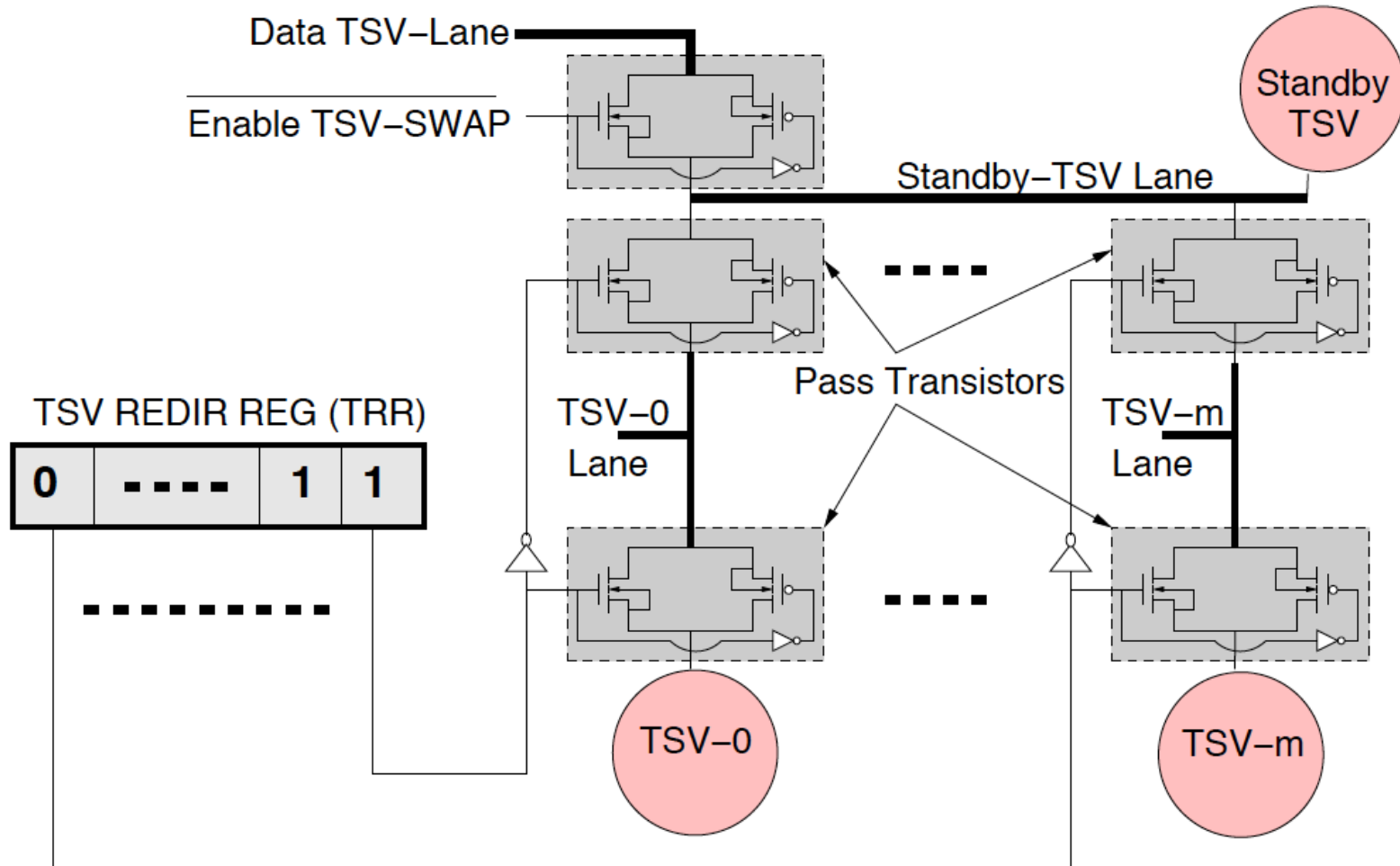
Recent papers<sup>\*+</sup> shows that

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

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

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

# TSV-SWAP REPAIR CIRCUIT



(Connect Standby TSV, Enable TSV-SWAP=1)

# PARITY CACHE: HIT RATE

