

# Citadel: Efficiently Protecting Stacked Memory

## From Large Granularity Failures

Prashant J. Nair<sup>†</sup>, David A. Roberts<sup>\*</sup> and Moinuddin K. Qureshi<sup>‡</sup>

Georgia Institute of Technology<sup>†‡</sup>, AMD Research - Advanced Micro Devices, Inc.<sup>\*</sup>

Email: pnair6@gatech.edu<sup>†</sup>, david.roberts@amd.com<sup>\*</sup>, moin@ece.gatech.edu<sup>‡</sup>

**Abstract**—Stacked memory modules are likely to be tightly integrated with the processor. It is vital that these modules operate reliably, where failure can require replacement of the entire socket. To make matters worse, stacked memory designs are susceptible to new failure modes (for example, due to faulty through-silicon vias, or TSVs) that can cause large portions of memory, such as a bank, to become faulty. To avoid data loss from large-granularity failures, the memory system may use symbol-based codes that stripe the data for a cache line across several banks (or channels). Unfortunately, such data-stripping reduces memory-level parallelism, causing significant slowdown and higher memory power consumption.

This paper proposes *Citadel*, a robust memory architecture that allows the memory system to store each cache line entirely within one bank, allowing high performance, low power and efficient protection from large-granularity failures. Citadel consists of three components; *TSV-Swap*, which can tolerate both faulty data-TSVs and faulty address-TSVs; *Three Dimensional Parity (3DP)*, which can tolerate column failures, row failures, and bank failures; and *Dynamic Dual-Granularity Sparing (DDS)*, which can mitigate permanent faults by dynamically replacing faulty memory regions with spares, either at a row granularity or at a bank granularity. Our evaluations with real-world DRAM failure data show that Citadel performs within 1% of, and uses only an additional 4% power versus a memory system optimized for performance and power, yet provides reliability that is 7x-700x higher than symbol-based ECC.



## 1 INTRODUCTION

3D-stacked DRAM technology can help with the challenges of power consumption, bandwidth demands and reduced footprint. One of the key enablers of stacked memory is the *through-silicon via* (TSV), which makes it possible to cost-effectively stack multiple memory dies on top of each other [1]. In such a system, memories that develop permanent faults (in TSVs and other components) must continue to work, avoiding expensive replacement of multiple chips. These factors motivate the adoption of a *fail-in-place* philosophy for designing stacked memories [2].

Recent work on DRAM reliability [3], [4] showed that multi-bit DRAM faults are common. To make matters worse, TSV failures introduce errors that can span several dies, causing column failures or bank failures. Thus, stacked memory systems will be even more susceptible to large-granularity failures due to newer fault models, such as those from TSVs. To optimize performance and power for stacked memory, we would like to retain the data for a cache line within a single bank. However, a bank failure would then cause loss of data for the whole cache line. Alternatively, one can adopt a philosophy similar to ChipKill [5] for tolerating large-granularity failures for stacked DRAM. In such a design, the data for a cache line would be striped across several banks (or channels). Unfortunately, such a data mapping would require the memory system to activate several banks to service a single request.

As shown in Figure 1, ideally we want a system that has the performance and power efficiency of storing the entire cache line in one bank (NoStripe), and yet maintains robustness to large granularity faults (Stripe). To that end, this paper proposes *Citadel*, a robust architecture that allows the memory system to retain the cache line within one bank (delivering high performance and low power) and yet efficiently protects the stacked memory from large-granularity failures.

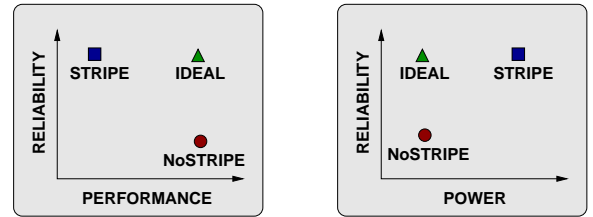


Fig. 1: Striping enhances reliability but sacrifices performance and power efficiency.

Like ECC DIMMs, which use one additional chip per 8 chips, Citadel relies on having one extra die along with eight data dies. ECC-DIMM provides 64 bits of ECC every 512 bits. Similarly, Citadel uses 64 bits of metadata (in the ECC space) for each 512-bit cache line. Citadel employs three components.

The first component, *TSV-Swap*, swaps faulty TSVs with good TSVs based on their architectural vulnerability. We found that while a data TSV typically affects only one bit in a data line (albeit across many lines), a failure of one of the address TSVs can make half of the memory unreachable. Our design uses the resources to repair up to eight TSVs to tolerate faults in data, address and command TSVs.

The second component, *three dimensional parity (3DP)*, mitigates internal DRAM die faults by maintaining parity in three spatial dimensions. The CRC-32 information associated with each line is used to detect errors in the line.

The third component, *Dynamic Dual-Grained Sparing (DDS)*, avoids the frequent correction of permanent faults. When a fault is detected, data is restored using 3DP and the row associated with the data line is remapped to a spare location in memory.

Our evaluations show that Citadel can provide reliability that is 100x-1000x higher than the baseline system while still retaining power and performance similar to a system that maps the entire cache line to the same bank.

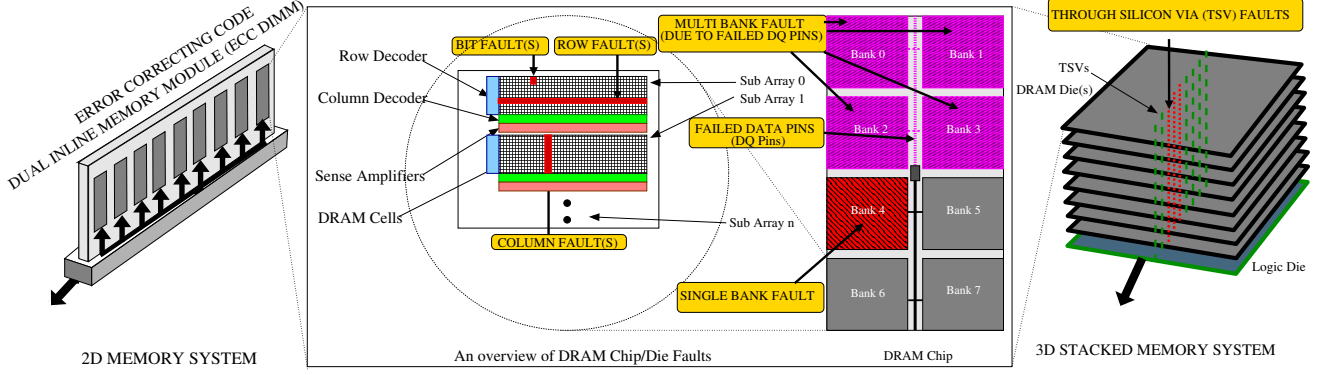


Fig. 2: Granularity of faults that occur in a DRAM Chip/Die. Faults can be at granularities of bit, column, row, bank(s) and TSVs for stacked memory systems. Common wiring faults within a chip cause multiple banks to fail.

## 2 BACKGROUND AND MOTIVATION

Failures can occur in any memory system (2D or 3D) at different granularities, as shown in Figure 2 [3], [4], [6].

### 2.1 Memory faults for traditional systems

The faults in individual chips are largely independent of each other. This paper follows the definitions for chip faults from [3]. Banks share wiring such as data, address and command buses [7], [8]. Single and multi-bank/rank faults are likely due to data, address and command buses failing<sup>1</sup>.

### 2.2 Transposing faults onto 3D-stacked memories

Layout of an individual die in 3D stacked memory systems is very similar to that of a chip in conventional 2D memory systems [9]–[12]. This paper extrapolates failure rates for all fault types except complete bank and complete rank from 2D memory system onto stacked memory systems. The key difference is the introduction of TSVs for connecting data and address lines [1]. Consequently, complete bank and rank faults in a 3D stack are now influenced by TSV faults.

### 2.3 3D-stacked memory: organization

High Bandwidth Memory (HBM) [13], Hybrid Memory Cube (HMC) [10], [14] and Octopus from Tezzaron [15] use a 3D-stacked memory architecture. These standards differ in their data organization and also share TSVs differently. However, these stacked memory systems fundamentally have the same layout. In this paper we present evaluations for Citadel using a hypothetical stack architecture with an HBM-like interface (referred to as HBM for brevity)<sup>2</sup>. Figure 3 shows internal stack organizations of HBM. Each channel may be fully contained in each DRAM die in the stack<sup>3</sup>.

**ECC layout:** The stacked memory consists of ‘n’ data dies and ‘m’ ECC dies (depending on value of n and ECC scheme). Similar to ECC-DIMMs, every parallel data request (512b) fetches its ECC bits (64b) through ECC lanes [13]. We use an 8-die stack with one ECC die for ECC or metadata information having the same overhead as ECC-DIMMs (12.5%).

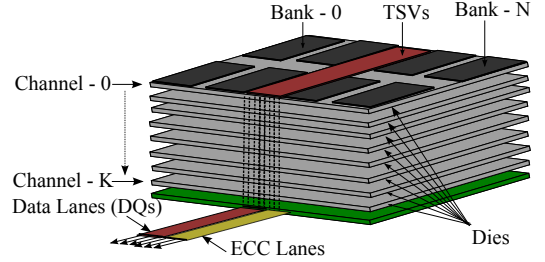


Fig. 3: Our design has one channel per die and all banks in this channel are in the same die. Similar to ECC-DIMM, it also has separate data and ECC I/O lanes.

### 2.4 Data striping in 3D memory systems

Unlike 2D DIMMs that stripe cache lines across chips, stacked memories can place the cache line in three ways.

- **Same Bank:** Within a single bank in a single channel.
- **Across Banks:** Within a single die (channel) and striped across banks.
- **Across Channels:** Within multiple dies (channels) and striped across one bank in each channel.

### 2.5 Impact of data striping

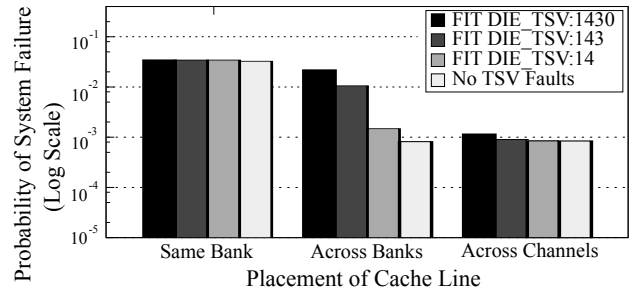


Fig. 4: Impact of data striping on reliability using strong 8-bit symbol based code (similar to Chipkill). Striping data across banks or channels gives higher reliability

To protect stacked DRAM from bank failures or channel failures, we can stripe data across banks or channels. In such a case, each bank/channel would be responsible for only a portion of the data for the cache line, and a correction mechanism (possibly ECC scheme) can be used to fix the sub-line-granularity fault. This organization requires the activation of multiple banks/channels to satisfy each memory request, thereby reducing memory-level parallelism. Figure 4 compares the reliability for three data mapping schemes for strong 8-bit symbol based ECC (similar to ChipKill) for different TSV FIT

1. [7], [8] show that shared bank circuits handle Data and Address links, there may be a few auxiliary circuits which can be protected with redundancy

2. In our additional evaluations, we found the resilience of Citadel to be equally high for HMC and Tezzaron type designs.

3. There is provision for multiple channels to share a die [13]. As die (chip) failures are common, conventional ECC schemes will perform even poorer if multiple channels share a die.

rates (other parameters are described in Section 3). System failure is the occurrence of an uncorrectable fault within a seven-year chip lifetime. ‘Across Channels’ offers the best layout giving highest reliability.

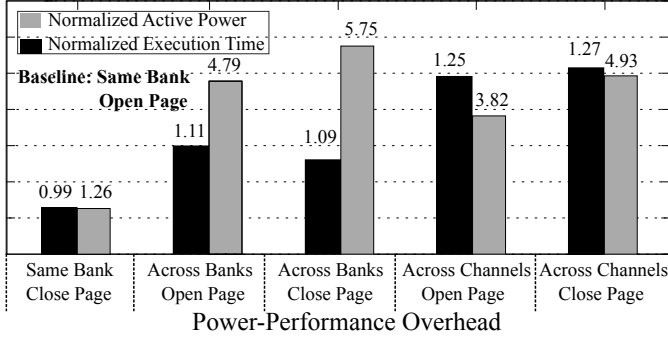


Fig. 5: Impact of data striping on power & performance. Striping data across banks or channels comes at a significant price in performance (10%-25%) and power (3x-5x)

However, the reliability benefits of ‘across banks’ and ‘across channels’ come at a significant price. Figure 5 shows that striping data ‘across banks’ causes a slowdown of approximately 10%, and ‘across channels’ causes a slowdown of approximately 25%. ‘Across banks’ and ‘across channels’ consumes 3-5x more active power than the ‘same bank’ mapping (‘across channels’ takes longer to execute, consuming energy over a longer time, hence the reduction in power).

## 2.6 Goal

To optimize for performance and power, we would like to maintain a ‘same bank’ mapping, yet have a means to tolerate large-granularity failures efficiently. We describe our methodology before describing our solutions.

## 3 EXPERIMENTAL METHODOLOGY

### 3.1 Fault models and simulation

TABLE 1: Stacked memory failure rates (8Gb dies)

DRAM Die Failure Mode	Fault Rate (FIT)	
	Transient	Permanent
Single bit	113.6	148.8
Single word	11.2	2.4
Single column	2.6	10.5
Single row	0.8	32.8
Single bank	6.4	80
TSV(Complete Bank/Channel)		
TSV (Address and Data)	Sweep:14 FIT - 1,430 FIT	

Real-world field data from Sridharan et al. [3] provides the failure rates as failures-in-time (FIT) for 1 Gb DRAM chips. We assume 8 Gb in line with industry projections, thus we accordingly increase the FIT rates from the 1 Gb die studies based on the relative component counts per logical block. Because we do not have field data for device failures due to TSVs, we perform a sensitivity study for TSV device FITs. We assume 0.01 to 1 device failures in 7 years (translating to Device FIT of 14 to 1,430) due to TSV faults. Table 1 shows these parameters. To evaluate reliability of different schemes, we use an industry grade repair simulator called *FaultSim* [16].

## 3.2 Performance and Power Evaluation

The baseline configuration for our in-house simulator is described in Table 2. Virtual-to-physical translation uses a first-touch policy with a 4KB page size.

TABLE 2: Baseline System Configuration

Processors	
Number of cores	8
Processor clock speed	3.2 GHz
Last-level Cache	
L3 (shared)	8MB
Associativity	8-way
Latency	24 cycles
Cache-line size	64Bytes
DRAM 2x8GB 3D Stacks	
Memory bus speed	800MHz DDR
Memory channels	8
Capacity per channel	1GB
Banks per channel	8
Row-buffer size	2KB
Data TSVs	256/Channel
Addr TSVs	24/Channel
$t_{WTR}-t_{CAS}-t_{RCD}-t_{RP}-t_{RAS}$	7-9-9-9-36

For our evaluations, we chose all 29 benchmarks from the SPEC CPU 2006 [17] suite. We also used memory-intensive benchmarks from the PARSEC [18] and BioBench [19] suite in rate mode. We measure active (read, write, refresh and activation) power using the equations from the Micron Memory System Power Note for 8Gb chip [20], [21].

## 4 CITADEL: AN OVERVIEW

We propose *Citadel*, a robust memory architecture that can tolerate both small- and large-granularity faults effectively. Figure 6 shows an overview of Citadel. HBM provisions 64 bits of ECC every 64 data bytes [13]. Citadel provisions each 64B cache line with 64 bits of metadata, similar to these standards. Citadel uses the ECC die to store different types of metadata information, each geared towards tolerating different types of faults. Each 64B (512b) transaction fetches 40b of metadata over ECC lanes. The remaining 24 bits is used for sparing in the metadata die. Citadel consists of three component schemes: *TSV-SWAP*, *Three Dimensional Parity (3DP)* and *dynamic dual-granularity sparing (DDS)*.

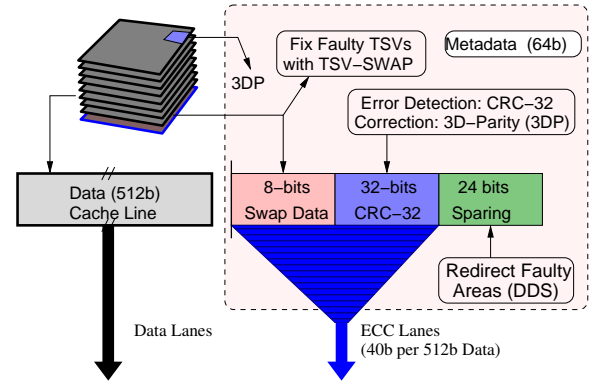


Fig. 6: Overview of Citadel

Citadel differentiates faults in memory elements from faults in TSVs. The TSV-SWAP technique of Citadel can tolerate TSV faults by dynamically identifying the faulty TSVs and decommissioning such TSVs. The data of faulty TSVs is replicated in the metadata (up to 8 bits). TSV-SWAP not only

protects against faulty data TSVs as well as faulty address TSVs, which tend to be even more severe in practice.

Citadel relies on CRC to detect data errors. Once an error is detected, it is corrected using the 3DP scheme, which maintains parity in three dimensions: across banks, across rows within one die, and across rows of different dies. 3DP can not only tolerate small-granularity failures such as bit and word failures as well as large-granularity failures such as row and bank failures. 3DP uses one of the data banks to implement bank-level parity (storage overhead of 1.6%).

Citadel employs data sparing to avoid frequent correction of faulty data. This not only prevents the performance overheads of error correction, but also makes the system more robust because permanent faults tend to accumulate over time. The DDS sparing scheme of Citadel exploits the observation that a bank either has a few small granularity faults (less than 4) or many (more than 1,000) faults; DDS spares at either a row granularity or a bank granularity. DDS uses three out of the eight banks of the metadata die for data sparing.

## 5 TSV-SWAP: MITIGATING TSV FAULTS

3D stacked memory systems use TSVs to connect data, address and command links between the logic die and DRAM dies. Without loss of generality, this section explains the working of TSVs, fault models and our solution.

### 5.1 Background on TSV

The HBM system in this paper consists of 8 channels of 256 data TSVs (DTSV) with 24 address/command TSVs (ATSV). Internally, TSVs transfer the address and command information for the channel to the corresponding die. For a memory request for one cache line, the entire 2KB of data for the row (DRAM page) is addressed and brought into the sense amplifiers. From the 2KB (16Kb) page, 64B (512bits) of data are multiplexed and transferred via the TSVs. The 256 DTSVs will transfer data in two bursts. Since all banks in the same die share the TSVs, TSV faults cause multi-bank failures.

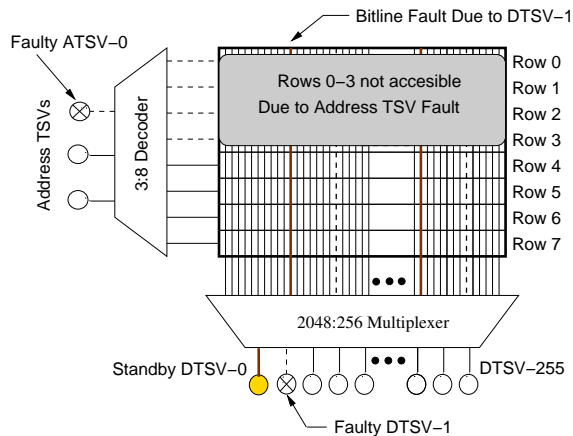


Fig. 7: Faults in data (DTSV) and address TSVs (ATSV). TSV-SWAP creates stand-by TSVs to tolerate TSV faults.

### 5.2 Severity of TSV faults: DTSV vs. ATSV

The vulnerability of the system to TSV faults depends on whether the fault happens in DTSV or ATSV, as shown in Figure 7. Because the burst size is 2, each DTSV fault will cause 2 bits to fail in every cache line. Faults in ATSV are more severe and make half of the memory unreachable.

## 5.3 TSV-SWAP vs using spare TSVs

Ang-Chih Hsieh et. al [22] propose an efficient method of using spare TSVs. TSV-SWAP can mitigate TSV faults at run-time without relying on manufacturer-provided spare TSVs and distinguishes between the severity of faults in address and data TSVs.

## 5.4 Design of TSV-SWAP

### 5.4.1 Creating stand-by TSVs

TSV-SWAP creates stand-by TSVs by duplicating the data of predefined TSV locations into the 8-bit swap data provided by metadata in Citadel (see Figure 6). Our design designates four TSVs as stand-by TSVs from a pool of 256 DTSV (DTSV-0, DTSV-64, DTSV-128, and DTSV-192). As each DTSV bursts two bits of data for each cache line, 8 bits from each cache line are replicated in the metadata (bit[0], bit[64], ..., bit[448]). The four stand-by TSVs are used to repair faulty TSVs.

### 5.4.2 Detecting faulty TSVs

Citadel computes CRC-32 over the concatenation of address and data to detect both fault types. A TSV error will result in an incorrect checksum. To differentiate between TSV faults and data faults, TSV-SWAP employs two additional rows per die that stores a fixed sequence of data. On detecting a CRC mismatch, data from these fixed rows are read and compared against the pre-decided sequence. If there is a data mismatch, the error is likely to be from a TSV fault. The memory system invokes the BIST logic and checks for TSV faults.

### 5.4.3 Redirecting faulty TSVs

TSV-SWAP provisions both the DTSV and ATSV with a redirection circuit that can replace a faulty TSV with one of the stand-by TSVs. The redirection circuit is simply a multiplexer and a register. On detecting a TSV fault, the BIST circuitry enables the TSV redirection circuit for the faulty TSV by configuring it to use one of the stand-by TSVs instead of the faulty DTSV or ATSV. TSV-SWAP requires a control logic that activates a stand-by TSV for a pool of address and data TSVs in case of a fault.<sup>4</sup>

## 5.5 Results for TSV-SWAP

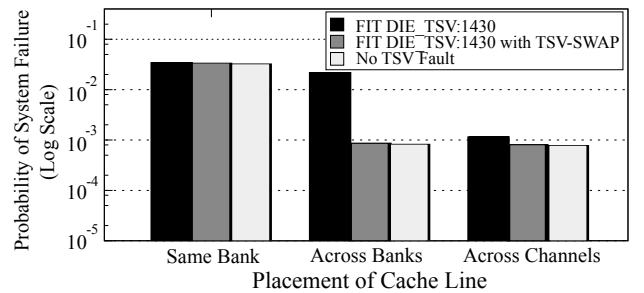


Fig. 8: Impact of TSV-SWAP under TSV failure conditions.

Figure 8 shows that TSV-SWAP is effective at mitigating TSV failures, even at a rate as high as 1,430. Subsequent sections assume a system that employs TSV-SWAP.

4. On detecting a TSV fault, the particular set register for the swap set in the die is loaded with sequence that enables TSV-SWAP. The register is a FIFO queue and loaded serially using at least two control TSVs (for redundancy). The chances of both the control TSVs failing is negligible



## 6 THREE-DIMENSIONAL PARITY (3DP)

The second component of Citadel targets error detection and error correction of data values. Citadel provisions each line with a 32-bit cyclic redundancy code (CRC-32), which is highly effective at detecting data errors [23], [24]<sup>5</sup>. In our multi-dimensional parity scheme, even if one dimension encounters two faults, these faults are highly unlikely to fall into the same block in the other two dimensions.

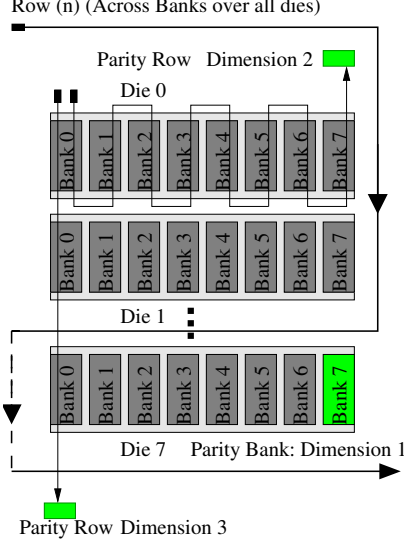


Fig. 9: Dimension 1 stripes parity across one row in every bank for all dies and generates a row in the parity bank. Dimension 2 stripes parity across all rows in every bank within a die to generate a parity row. Dimension 3 stripes parity across all rows in single bank across dies to generate a parity row.

### 6.1 Design of 3DP

Figure 9 shows the design of Dimension 1. It computes the parity for a row in every bank across dies as specified in equation (1). This requires dedicating one bank as a parity bank for the entire stack (1.6% overhead, for our 8 channel system, with 8 banks for each channel).

$$\text{ParityBank}[\text{row}_n] = \text{Die}_0.\text{Bank}_0[\text{row}_n] \oplus \text{Die}_0.\text{Bank}_1[\text{row}_n] \oplus \dots \oplus \text{Die}_7.\text{Bank}_6[\text{row}_n] \quad (1)$$

Figure 9 shows the design of Dimensions 2 and 3. In Dimension 2, parity is taken across all rows in all banks within a die. Equation (2) shows the computation *Parity Row* in Dimension 2 for Die 0. Because there are 9 dies (including the metadata die), the storage overhead is 9× the size of a DRAM row.

$$\text{ParityRowDim2}_{\text{Die0}} = [\text{Bank}_0[\text{row}_0] \oplus \text{Bank}_0[\text{row}_1] \oplus \dots \oplus \text{Bank}_7[\text{row}_n]]_{\text{Die0}} \quad (2)$$

Dimension 3 computes parity across dies for all rows in a single bank. Equation (3) shows the computation for *Parity Row* in Dimension 3 for Bank 0. Because there are 8 banks per die, the storage overhead of is 8×size of DRAM row. Dimension 1 is designed to tolerate bank failures, Dimensions 2 and 3 prevent independent row, word and bit failures. 3DP corrects multiple errors that occur at the same time.

$$\text{ParityRowDim3}_{\text{Bank0}} = [\text{Die}_0[\text{row}_0] \oplus \text{Die}_0[\text{row}_1] \oplus \dots \oplus \text{Die}_7[\text{row}_n]]_{\text{Bank0}} \quad (3)$$

5. The probability of overlapping CRC-32 checksum is  $\frac{1}{2^{32}} \approx 10^{-10}$ , the probability that failed element has same CRC-32 is extremely low ( $\ll 10^{-14}$ )

## 6.2 Reducing overheads for parity update

We avoid the performance overheads of updating the parity for dimensions 2 and 3 by keeping the parity information on-chip. We propose caching the parity for dimension 1 parity. The average hit rate of parity caching is 85%, showing that parity caching is quite effective. Because old data must be compared with the new modified data to compute new parity, 3DP also requires a read-before-write (RBW) operation. A write transaction following this RBW only encounters a small penalty of the write-to-read turnaround delay.

### 6.3 Results for 3DP

We compare the resilience, performance, and power of the 3DP scheme to a theoretical scheme that employs an 8-bit symbol-based coding with data striping.

#### 6.3.1 Resilience

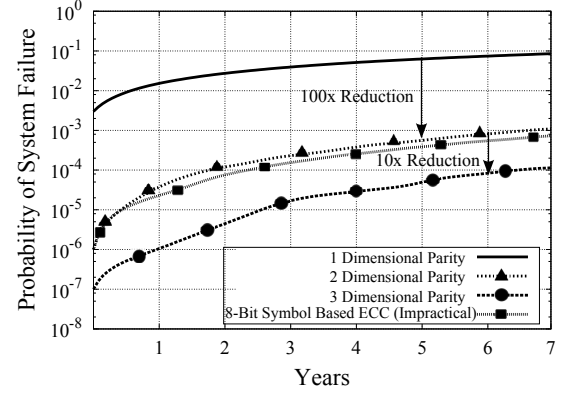


Fig. 10: 3DP has 7x more resilience than an 8-bit symbol-based ECC code for tolerating large-granularity failures in stacked memory. 3DP has 10x more resilience than 2DP

Figure 10 shows that 3DP achieves 7x stronger resilience than an 8-bit symbol-based ECC because it can handle multiple concurrent faults with one bank failure.

#### 6.3.2 Performance

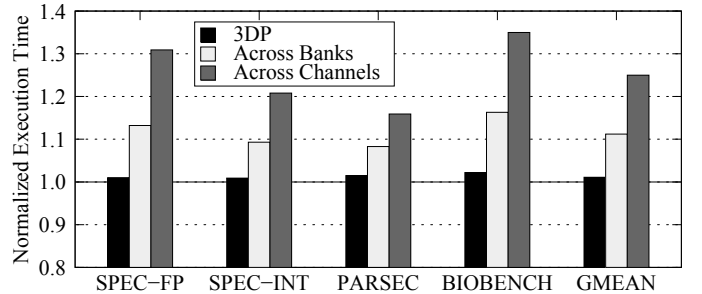


Fig. 11: Normalized execution time: 3DP has negligible slowdown, whereas data striping causes 10-25% slow-down.

Figure 11 shows that 3DP scheme with caching has performance within 1% of the baseline, 3DP without caching degrades performance by 4.5%. However, alternative schemes degrade performance 10% to 25%, on average. 3DP with parity caching increases memory traffic by only 8%.

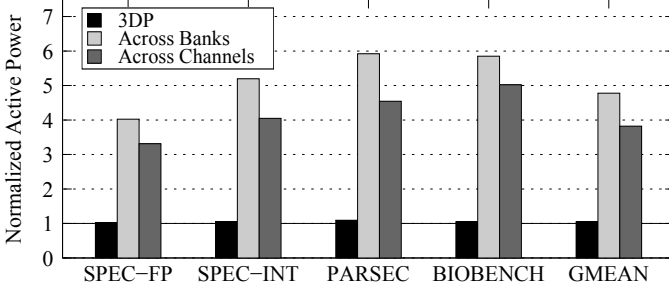


Fig. 12: Active power consumption: 3DP has negligible power overheads, whereas data striping has 3-5x greater overhead.

### 6.3.3 Power

Figure 12 shows that 3DP increases power by 4% compared to the baseline system. Comparatively, the AB and AC data organizations increase power by 3-5x because of higher bank/channel activations and row conflicts.

## 7 DYNAMIC DUAL-GRANULARITY SPARING

The third component of Citadel replaces faulty data blocks. The correction employed by 3DP fixes the data error by recomputing the data based on parity and can be a time-consuming process if its done everytime(recomputing parity and isolating the fault in each dimension). Citadel avoids this by using dynamic sparing, whereby a data item once corrected is redirected to an alternate location. Dynamic Dual-granularity Sparing (DDS) spares small and large granularity faults separately.

### 7.1 Observation: Failures tend to be bimodal

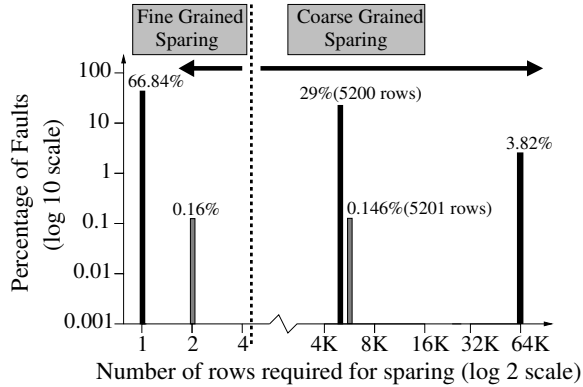


Fig. 13: A permanent fault affects either very few (less than 4) or very many (> 1000) rows. This motivates a dual-grained sparing architecture at row and bank granularity.

Only for the analysis in this section, we will classify all faults that are smaller than or equal to a row fault as causing a row failure. Figure 13 shows the distribution of the number of rows that are used by a faulty bank, on average. The smaller-granularity faults do not occur in many multiples. In fact, in all our simulations, no more than two rows per bank were affected by a small-granularity fault within a scrubbing interval. However, there are two peaks; one at 5,200 rows (size of a sub-array) and another at 65K rows (size of a bank). This motivates our design to use two granularities of sparing, either a row or a bank.

### 7.2 Design of dynamic dual-granularity sparing

DDS has two components; the spare area and the redirection table. Because we employ two granularities of sparing we have two redirection tables; one at row granularity and the other one at a bank granularity.

**Spare area and redirection:** Three banks in the metadata die are partitioned into coarse-granularity sparing banks (*spare bank-0* and *spare bank-1*) and a fine granularity bank (*spare bank-2*) that provides space for row-based sparing. The banks and rows are identified using redirection tables.

### 7.3 Results

Figure 14 shows that DDS when applied with 3DP delivers a 700x improvement in resilience compared to the baseline strong 8-bit symbol-based ECC code.

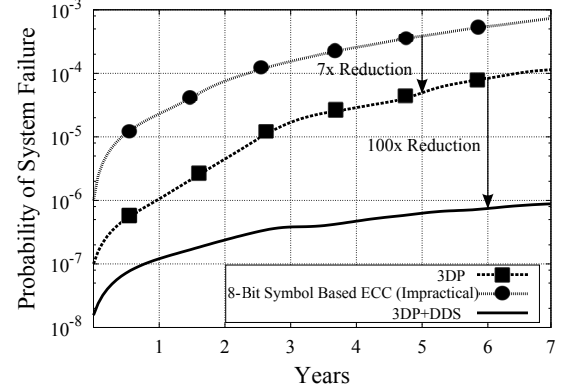


Fig. 14: Resilience: 3DP+DDS provides 700x more resilience than symbol-based codes that rely on data striping

### 7.4 Overall storage overhead of Citadel

Citadel relies on having an extra die for storing metadata for the eight data dies (12.5% overhead). Dimension-1 parity has 1.6% overhead (one bank out of 64 banks). Remaining parity bits have a 35KB overhead. Citadel provides 700x better reliability while requiring a storage overhead (14%) similar to ECC DIMM (12.5%).

## 8 RELATED WORK

Memory reliability for emerging memory technologies and existing DRAM systems has become an important topic. We describe the schemes that are most relevant to our proposal.

Citadel employs TSV-SWAP to mitigate faulty TSVs. Faulty TSVs can be avoided at manufacturing time using spare TSVs. Several techniques have been proposed for “swapping in” such redundant TSVs to replace faulty TSVs in a 3D die stack [25]. To the best of our knowledge, this paper is the first to address run-time mitigation of TSVs and without relying on manufacturer-provided spare TSVs.

The work that is most closely related to our work is on reliably architecting stacked DRAM as caches [24]. It uses CRC-32 to detect errors in caches. However, correction is performed simply by disabling clean lines and replicating dirty lines. While such correction can be useful for caches, disabling random locations of lines is an impractical option for main memory. Furthermore, replicating all the data for

main memory leads to a capacity loss of 50% and doubles the memory activity. Our work provides low-cost and effective fault tolerance for using stacked DRAM as main memory.

Yoon et al. [26] proposed *Virtual and Flexible ECC*. Rather than using uniform error correction across the entire memory space, it allows the user to specify stronger levels of ECC for high-priority applications and weaker levels of ECC for low-priority applications. Citadel uses multi-dimensional parity rather than multi-tiered ECC. Citadel is more area-efficient and does not require any support from the OS.

Efficient memory repair for bit-level faults has been proposed for both SRAM [27] [28] and DRAM [29]. However, such techniques are effective only for random bit errors, and become ineffective at tolerating large-granularity faults. Erasure Codes can identify faulty chips to be disabled [30]–[32]. However, they can operate only at one granularity. Unlike erasure codes, DDS enables flexible granularity sparing.

Citadel uses parity for error correction, as do other schemes such as RAID [33]. However, parity based RAID schemes need to store detection codes along with parity blocks for correction and incur a high area overhead (upto 25%). BCH codes can be used to provide protection for multiple-bit errors (e.g. 6 or more bits) [34] [35]. Unfortunately, strong BCH codes cannot handle large-granularity faults without significant overheads.

## 9 CONCLUSION

3D die stacking introduces new multi-bit failure modes, exacerbating the large-granularity faults identified by recent DRAM field studies. This paper proposes *Citadel* to tolerate such large-granularity faults efficiently. To this effect, this paper makes the following contributions:

- 1) TSV-SWAP, which mitigates TSV faults at run-time, without relying on manufacturer-provided spare TSVs. It remains effective even at high TSV failure rates.
- 2) Three-dimensional parity (3DP), which can correct a wide variety of faults, including bit, word, row and bank faults.
- 3) Dynamic dual-granularity sparing (DDS), which can spare faulty data blocks either at a row granularity or at a bank granularity to avoid the accumulation of permanent faults and frequent episodes of error correction. DDS with 3DP gives 700x more resilience than a 8-bit symbol based code.

## REFERENCES

- [1] U. Kang et al., “8gb 3d ddr3 dram using through-silicon-via technology,” in *ISSCC 2009*, pp. 130–131, 131a, 2009.
- [2] M. Dubash, “Not hot swap but ‘fail in place’,” in *TechWorld*, 2004.
- [3] V. Sridharan and D. Liberty, “A study of dram failures in the field,” in *High Performance Computing, Networking, Storage and Analysis (SC)*, 2012 *International Conference for*, pp. 1–11, 2012.
- [4] V. Sridharan, J. Stearley, N. DeBardeleben, S. Blanchard, and S. Gurumurthi, “Feng shui of supercomputer memory: Positional effects in dram and sram faults,” *SC ’13*, pp. 22:1–22:11, 2013.
- [5] T. J. Dell, “A white paper on the benefits of chipkillcorrect ecc for pc server main memory,” Tech. Rep. 11/19/97, IBM, 1997.
- [6] B. Schroeder and G. Gibson, “A large-scale study of failures in high-performance computing systems,” *Dependable and Secure Computing, IEEE Transactions on*, vol. 7, no. 4, pp. 337–350, 2010.
- [7] J.-H. Yoo et al., “A 32-bank 1 gb self-strobing synchronous dram with 1 gbyte/s bandwidth,” *Solid-State Circuits, IEEE Journal of*, vol. 31, no. 11, pp. 1635–1644, 1996.
- [8] S. Shiratake et al., “A pseudo multi-bank dram with categorized access sequence,” in *VLSI Circuits, 1999. Digest of Technical Papers. 1999 Symposium on*, pp. 127–130, 1999.
- [9] J.-S. Kim et al., “A 1.2v 12.8gb/s 2gb mobile wide-i/o dram with 4x128 i/os using tsv-based stacking,” in *ISSCC 2011*, pp. 496–498, 2011.
- [10] J. T. Pawlowski, “Hybrid memory cube (hmc),” in *HOT-CHIPS*, 2011.
- [11] T. Hollis, “Modeling and simulation challenges in 3d memories,” in *DesignCon*, 2012.
- [12] J. Bolaria, “Micron reinvents dram memory,” in *Microprocessor Report (MPR)*, 2011.
- [13] J. Standard, “High bandwidth memory (hbm) dram,” in *JESD235*, 2013.
- [14] H. M. C. Consortium, “Hybrid memory cube specification 1.0,” 2013.
- [15] Tezzaron Semiconductor, *Octopus 8-Port DRAM for Die-Stack Applications: TSC100801/2/4*, 2010.
- [16] D. Roberts and P. Nair, “Faultsim: A fast, configurable memory-resilience simulator,” in *The Memory Forum: In conjunction with ISCA-41*.
- [17] “Spec cpu2006 benchmark suite,” in *Standard Performance Evaluation Corporation*.
- [18] C. Bienia, “Benchmarking modern multiprocessors,” in *Ph.D. Thesis, Princeton University*, 2011.
- [19] K. Albayraktaroglu, A. Jaleel, null Xue Wu, M. Franklin, B. Jacob, null Chau-Wen Tseng, and D. Yeung, “Biobench: A benchmark suite of bioinformatics applications,” vol. 0, pp. 2–9, 2005.
- [20] Micron, *Calculating Memory System Power for DDR3*, 2007.
- [21] Micron, *MT41J512M4:8Gb QuadDie DDR3 SDRAM Rev. A 03/11*, 2010.
- [22] A.-C. Hsieh, T. Hwang, M.-T. Chang, M.-H. Tsai, C.-M. Tseng, and H.-C. Li, “Tsv redundancy: Architecture and design issues in 3d ic,” in *Design, Automation Test in Europe Conference Exhibition (DATE)*, 2010, pp. 166–171, March 2010.
- [23] W. Peterson and D. Brown, “Cyclic codes for error detection,” *Proceedings of the IRE*, vol. 49, no. 1, pp. 228–235, 1961.
- [24] J. Sim, G. H. Loh, V. Sridharan, and M. O’Connor, “Resilient die-stacked dram caches,” in *Proceedings of the 40th Annual International Symposium on Computer Architecture, ISCA ’13*, (New York, NY, USA), pp. 416–427, ACM, 2013.
- [25] L. Jiang, Q. Xu, and B. Eklow, “On effective tsv repair for 3d-stacked ics,” in *Design, Automation Test in Europe Conference Exhibition (DATE)*, pp. 793–798, 2012.
- [26] D. H. Yoon and M. Erez, “Virtualized and flexible ecc for main memory,” *ASPLOS ’10*, (New York, NY, USA), pp. 397–408, ACM, 2010.
- [27] D. Roberts, N. S. Kim, and T. Mudge, “On-chip cache device scaling limits and effective fault repair techniques in future nanoscale technology,” *DSD ’07*, (Washington, DC, USA), pp. 570–578, IEEE Computer Society, 2007.
- [28] C. Wilkerson, H. Gao, A. R. Alameldeen, Z. Chishti, M. Khellah, and S.-L. Lu, “Trading off cache capacity for reliability to enable low voltage operation,” *ISCA ’08*, (Washington, DC, USA), pp. 203–214, IEEE Computer Society, 2008.
- [29] P. J. Nair, D.-H. Kim, and M. K. Qureshi in *Proceedings of the 40th Annual International Symposium on Computer Architecture, ISCA ’13*, (New York, NY, USA), pp. 72–83, ACM, 2013.
- [30] J. Nerl, K. Pomaranski, G. Gostin, A. Walton, and D. Soper, “System and method for controlling application of an error correction code (ecc) algorithm in a memory subsystem.”
- [31] D. H. Yoon, J. Chang, N. Muralimanohar, and P. Ranganathan, “Boom: Enabling mobile memory based low-power server dimms,” in *Proceedings of the 39th Annual International Symposium on Computer Architecture, ISCA ’12*, (Washington, DC, USA), pp. 25–36, IEEE Computer Society, 2012.
- [32] J. Nerl, K. Pomaranski, G. Gostin, A. Walton, and D. Soper, “System and method for applying error correction code (ecc) erasure mode and clearing recorded information from a page deallocation table.”
- [33] A. Thomasian and J. Menon, “Raid5 performance with distributed sparing,” *Parallel and Distributed Systems, IEEE Transactions on*, vol. 8, no. 6, pp. 640–657, 1997.
- [34] S. Li, K. Chen, M.-Y. Hsieh, N. Muralimanohar, C. Kersey, J. Brockman, A. Rodrigues, and N. Jouppi, “System implications of memory reliability in exascale computing,” in *High Performance Computing, Networking, Storage and Analysis (SC)*, 2011 *International Conference for*, pp. 1–12, 2011.
- [35] C. Wilkerson, A. R. Alameldeen, Z. Chishti, W. Wu, D. Somasekhar, and S.-L. Lu, “Reducing cache power with low-cost, multi-bit error-correcting codes,” *ISCA ’10*, 2010.