Global Journal of Computer Science and Technology: A Hardware & computation
Volume 25 Issue 1 Version 1.0 Year 2025
Type: Double Blind Peer Reviewed International Research Journal
Publisher: Global Journals
Online ISSN: 0975-4172 & Print ISSN: 0975-4350

Silent Data Errors in GPUs: Challenges and Mitigation in Modern Silicon

By Sameeksha Gupta

Abstract- Silent data errors in graphics processing units (SDEs) represent a critical challenge for modern computational systems that rely on these accelerators in high-performance computing, artificial intelligence, and data center operations. These errors propagate through calculations without triggering detection mechanisms, potentially compromising results in critical applications from autonomous vehicles to medical diagnosis. Quantitative analysis reveals disturbing error rates: 8.15×10^-3 FIT per device at sea level (one error per 14,000 device-hours), with error rates increasing 17-32% when running at full computational capacity in data centers. The physical causes of SDEs include cosmic radiation (causing 61.7% of faults to propagate undetected in streaming multiprocessors), manufacturing variations (contributing to 4.3% of silent computational failures), thermal stress cycles, voltage fluctuations, and aging effects that impact semiconductor reliability.

Keywords: silent data errors, GPU reliability, cosmic radiation sensitivity, architectural vulnerability, workload resilience, error mitigation strategies.

GJCST-A Classification: LCC Code: QA76.9.C65

Strictly as per the compliance and regulations of:

Silent Data Errors in GPUs: Challenges and Mitigation in Modern Silicon

Sameeksha Gupta

Figure

Abstract- Silent data errors in graphics processing units (SDEs) represent a critical challenge for modern computational systems that rely on these accelerators in high-performance computing, artificial intelligence, and data center operations. These errors propagate through calculations without triggering detection mechanisms, potentially compromising results in critical applications from autonomous vehicles to medical diagnosis. Quantitative analysis reveals disturbing error rates: 8.15×10^-3 FIT per device at sea level (one error per 14,000 device-hours), with error rates increasing 17-32% when running at full computational capacity in data centers. The physical causes of SDEs include cosmic radiation (causing 61.7% of faults to propagate undetected in streaming multiprocessors), manufacturing variations (contributing to 4.3% of silent computational failures), thermal stress cycles, voltage fluctuations, and aging effects that impact semiconductor reliability. Architectural vulnerability varies significantly: register files exhibit 36% silent data corruption rates versus 23% for shared memory and 11% for global memory, while instruction vulnerability ranges from 6.1% for integer operations to 42.7% for atomic operations. Workload characteristics dramatically affect error sensitivity, with machine learning inference showing up to 19.3% accuracy reduction from moderate error rates in transformer models versus 8.6% in convolutional networks. Mitigation strategies span hardware (ECC reducing corruption by 78.5%), firmware, and software domains, with recent selective redundancy techniques achieving 91% error coverage with only 32% performance overhead. Cross-layer resilience approaches demonstrated in recent research can reduce critical data integrity errors by up to 93.4% compared to default protection methods. Understanding these complex interactions and implementing targeted protection systems is essential for developing resilient GPU computing platforms that maintain both performance at scale and reliability.

Keywords: silent data errors, GPU reliability, cosmic radiation sensitivity, architectural vulnerability, workload resilience, error mitigation strategies.

I. Introduction

Graphics Processing Units (GPUs) have increased in various fields ranging from niche rendering hardware to core computational accelerators, such as high-performance computing (HPC), Artificial Intelligence, and Data-Scalable Operations. This role has made GPUs key infrastructure elements for applications from weather forecasting and molecular simulations to deep learning model training. As per Maleki et al., a comprehensive investigation of performance and reliability in modern GPUs reveals that for current technology nodes at sea level, silent data corruption (SDC) rates can reach alarming levels of 0.51 FIT/Mbit (Failures In Time per million bits) with overall observable error rates of 0.89 FIT/Mbit [1]. This equates to a disturbing rate of undetectable errors in high-scale deployments, where the total memory footprint of as

Author: Independent Researcher, USA
e-mail: sameekshasamgupta@gmail.com

many as petabytes is possible. But the unrelenting quest for performance gains through higher transistor densities and lower operating voltages has brought forth substantial reliability issues, most notably in the guise of silent data errors (SDEs).

Silent data errors pose a specifically pernicious challenge to computational integrity, being ones that afflict systems in silence without activating immediate detection mechanisms within system software or hardware. In contrast to traditional faults that give rise to well-defined, readily recognizable system crashes or diagnostic messages, SDEs travel through calculations undetected, potentially contaminating result accuracy, destabilizing system reliability, and eroding availability in deployed environments. Data center deployments using GPU acceleration for data analysis have shown that error rates grow 17-32% when running at full computational capacity, with an estimated 22% of such errors occurring in the form of silent corruptions that go undetected by traditional monitoring infrastructure, based on SQream's large-scale field testing on 1,500 production nodes [2]. The significance goes beyond simple inconvenience, with potential impact on pivotal decision-making processes in applications like autonomous systems, medical diagnostics, and financial modeling.

This work discusses the multi-faceted character of SDEs in contemporary GPU architectures, pinpointing primary vulnerability factors along architectural sub-components and operational regimes. The discussion covers both physical error-inducing mechanisms and architectural aspects that condition error propagation paths. In addition, this paper analyzes existing mitigation techniques and recommends approaches to improve GPU reliability against SDEs. The comprehension of such intricate interactions is crucial to developing future-generation GPU systems with the ability to provide both reliability and performance at scale.

II. Root Causes of Silent Data Errors in Modern GPU Silicon

Silent data errors in GPUs result from several connected physical effects influencing semiconductor dependability. Cosmic radiation is an important external factor with high-energy neutrons that traverse shielding materials and create electron-hole pairs in semiconductor substrates. Charged particles perturb stored values in memory devices and logic circuits, leading to bit flips that can go undetected. According to Ferreira et al., comprehensive neutron beam testing across multiple generations of GPU architectures has demonstrated that the architectural vulnerability factor (AVF) of register files increases by approximately 25.1% with each process node shrink, with contemporary GPUs exhibiting approximately 8.15×10^-3 FIT per device at sea level altitude—equating to one silent data error in approximately 14,000 device-hours under typical workloads [3]. Their neutron beam testing of 15,840 device-hours experiments proved that streaming multiprocessors (SMs) exhibited especially high susceptibility, where 61.7% of faults injected indeed propagated undetected through computations, as opposed to only 17.2% in traditional CPU pipelines. Susceptibility to such radiation-induced transient faults rises with decreasing feature sizes and degrading critical charge thresholds in future manufacturing nodes. Manufacturing process variations represent another intrinsic source of weakness. Even for advanced fabrication processes, statistical fluctuations in dopant levels, gate oxide thickness, and lithographic alignment result in marginally functional circuit regions. These fluctuations appear as timing violations under some operating conditions and may result in wrong computation outcomes without activating error detection circuits. Research presented at the Workshop on GPU Reliability has demonstrated that process fluctuations in advanced GPU designs can cause threshold voltage (Vt) variations of up to 30mV for individual streaming multiprocessors, resulting in timing differences of 7.5-11.2% on critical paths [4]. Their diligent examination of 27 production GPUs showed that about 4.3% of all silent computational failures were directly attributable to manufacturing differences, with an average seen rate of 1.7×10^-10 errors per operation when running at nominal voltage levels—a number that increases by orders of magnitude to 4.9×10^-8 errors per operation when running at lowered voltage margins to reduce power consumption. The issue is compounded in GPUs, which contain billions of transistors over huge die areas, raising the statistical probability of having susceptible elements.

Thermal cycling also degrades silicon reliability by causing differential expansion coefficients among materials in the GPU package. Electro migration processes are sped up, and the progression of crack formation in interconnect structures is accelerated by repeated thermal cycling. These impacts are especially significant in GPUs because of their high power densities and dynamic workload profiles, which cause extreme temperature gradients within the die. Voltage variations, both long-term droop and short-term noise, are another important mechanism for generating SDEs. Contemporary GPUs run at very tight voltage margins in order to achieve better energy efficiency, narrowing the gap between nominal operation voltage and minimum voltage for correct function. This reduced margin heightens vulnerability to temporary voltage fluctuations that could lead to timing violations in critical paths without invoking protective action.

Table 1: Physical Phenomena Contributing to Silent Data Errors in GPUs [3, 4]

Physical Mechanism	Error Manifestation	Vulnerability Trends	Key Affected Components
Cosmic Radiation	Electron-hole pair generation	Increases with node shrinking	Register files, SRAM cells
Manufacturing Variations	Timing violations	Higher impact in large dies	Critical timing paths, marginal circuits
Thermal Stress	Electromigration acceleration	Exacerbated by workload variation	Interconnect structures, package interfaces
Voltage Fluctuations	Timing margin violations	Worsens with efficiency optimizations	Critical paths, dynamic voltage domains
Aging Effects (NBTI, HCI, TDDB)	Progressive parameter shift	Cumulative degradation over time	Transistor characteristics, noise margins

III. Error Manifestation Across GPU Architectural Units

The heterogeneity of GPU architectures generates varied avenues through which silent data errors emerge and spread. In memory subcutaneous, SRAM-based units such as register files, cache, and shared memories are especially susceptible to single phenomena upset. Such devices usually run at low voltage levels to save electricity, reduce their noise immunity, and increase sensitivity to transient disturbances. According to Sullivan et al., detailed characterization of GPU vulnerability through extensive fault injection campaigns has revealed that memory devices exhibit significantly different error propagation patterns, with approximately 36% of fault injections in register files manifesting as silent data corruptions compared to 23% for shared memory and 11% for global memory [5]. Their work proved register file faults are especially challenging because they presented an average 4,372-cycle latency before detection, with errors having the potential to propagate through several computational phases. Furthermore, 47% of register file faults in scientific simulations left the system in functional mode but with silent computational errors without crashing the system, exacerbating a reliability gap. While bigger memory organizations such as HBM and GDDR6 generally include error-correcting codes (ECC), internal SRAM buffers, and smaller caches in most GPU implementations are not protected or apply only parity-based detection without correction.

Execution units have unique vulnerability profiles depending on their computation properties. Floating-point units have intricate arithmetic circuits with long computational pipelines that prolong temporal vulnerability windows. Integer units, though overall more tolerant, are still vulnerable to errors in timing-critical paths. Tensor cores, optimized for matrix operations within AI applications, mix high computation density with low-precision formats, forming intricate error-propagation channels that can amplify initially subtle perturbations among matrix elements. Research by Mei et al. using advanced fault-injection methodologies demonstrated that instruction-level susceptibility varies dramatically, with architectural vulnerability factors (AVF) of 6.1% for basic integer instructions, 29.4% for complex floating-point operations, and peaks of 42.7% for atomic instructions that interact with memory subsystems [6]. Their fine-grain analysis of 16 GPGPU programs exposed that single-precision floating-point multiply-accumulate instructions had an average of 2.13 single-bit errors propagating into an average of 9.47 output elements, resulting in a significant error magnification effect. Most problematic was the fact that in 88,467 instruction-level fault injections, nearly 18.3% of computational unit errors led to results that looked valid but actually were erroneous, highlighting the difficulty of silent data corruptions.

Data movement infrastructure, such as on-chip networks, memory controllers, and PCIe interfaces, adds new error vectors. These elements need to preserve signal integrity over different distances and across multiple clock domains, providing opportunities for transmission errors that go undetected. This problem is compounded by sophisticated power management features that dynamically manage clock frequencies and voltage levels, setting up potentially transient conditions that enable silent failures during domain crossing or state transitions. Control logic that manages thread scheduling, workload allocation, and synchronization is a very sensitive point of vulnerability. Failures impacting these structures have the potential to create multiplicative failures by sending computation to the wrong execution elements, misallocating memory access behaviors, or polluting synchronization primitives.

Table 2: Error Vulnerability across GPU Architectural Components [5, 6]

Architectural Component	Vulnerability Profile	Error Propagation Characteristics	Protection Status
Register Files	High (SRAM-based, reduced voltage)	Extended propagation before detection	Limited/Parity only
Shared Memory	Moderate vulnerability	Medium propagation scope	Partial ECC in newer designs
Global/HBM Memory	Lower relative vulnerability	Limited propagation scope	Typically ECC protected
Floating-Point Units	High (complex arithmetic)	Error amplification in dependent ops	Minimal protection
Integer Units	Moderate vulnerability	Limited error propagation	Limited protection
Tensor Cores	High (dense operations)	Significant error amplification	Implementation-dependent
Control Logic	Critical vulnerability	Multiplicative error effects	Limited redundancy
Data Movement Infrastructure	Moderate with hotspots	Cross-domain propagation	Protocol-level detection

IV. Workload Characteristics and Error Sensitivity

Computational workloads have different degrees of resistance to silent data errors, thus having a multifaceted relationship between application properties and error sensitivity. Scientific computing applications based on iterative solvers can show intrinsic error attenuation in some instances, since numerical algorithms converge toward reliable solutions even with transient perturbations. But these same applications usually have pivotal computations where even small mistakes can cause disastrous divergence or invalidate results altogether. As demonstrated in quantum computing research by Zhao et al., detailed error analysis of GPU-accelerated scientific codes reveals significant variation in error manifestation rates, with numerical simulation codes exhibiting Silent Data Corruption (SDC) in 37.5% of observed events, while signal processing applications showed only 18.2% SDC rates under identical testing conditions [7]. Their in-depth experiment with 2,304 hours of neutron beam testing showed matrix multiplication kernels to be far more sensitive to single-bit flipping (43.7% of faults injected resulting in incorrect outputs) than FFT implementations (21.3%). Of particular note was that they found around 27.8% of radiation-induced errors in iterative solvers spreading through subsequent computational steps undetected, despite the presence of checking routines to detect numerical irregularities. This heterogeneity poses immense difficulties for broad error protection measures and emphasizes the necessity for application-dependent resilience strategies.

Machine learning applications exhibit a very subtle error sensitivity profile. Training stages typically exhibit resistance to the occasional numerical inaccuracies because optimization algorithms are stochastic and training datasets involve inherent noise. This native resilience has led to an investigation into deliberatively lowered precision computations that sacrifice numerical accuracy for speed and energy efficiency. Inference workloads, however, tend to need more accurate computations, especially in safety-critical domains where misleading predictions could have deleterious effects. Experiments by Sharma and Sharma illustrate through extensive fault injection campaigns across various neural network architectures that bit error rates of as low as 10^-6 in tensor cores can lead to a classification accuracy loss of 12.7% for inference workloads, while training operations would have decent convergence even for error rates of 10^-4 [8]. Their thorough analysis of 12,800 error injection cases across five typical DNN models disclosed that transformer models exhibited exceptionally strong sensitivity, with a mean accuracy reduction of 19.3% at moderate error rates versus 8.6% for convolutional networks. Most alarming was the discovery of their work that 31.2% of silent errors in safety-critical vision models yielded high-confidence misclassifications of critical objects such as pedestrians and traffic signals. This extreme contrast highlights the necessity of error containment strategies specific to deployment context as opposed to protection mechanisms in general.

Graphics rendering pipelines exhibit aspects of both deterministic and probabilistic computation. Some rendering algorithms, especially those utilizing stochastic sampling methods such as path tracing, exhibit inherent tolerance to the rare occurrence of errors. On the other hand, geometry processing phases demand accurate computation to preserve visual correctness, with any errors tending to show up as observable artifacts or structural distortions in rasterized scenes. Data-dependent sensitivity of error makes things even tougher. Some patterns of data or sequential operations tend to activate weaknesses in certain circuit components that are normally latent.

Table 3: Workload Characteristics and Error Resilience [7, 8]

Application Domain	Error Resilience	Critical Vulnerability Points	Error Amplification Risk
Scientific Computing: Iterative Solvers	Moderate natural attenuation	Convergence-critical operations	Low to Moderate
Scientific Computing: Matrix Multiplication	Low inherent resilience	Core arithmetic operations	High
Scientific Computing: FFT	Moderate resilience	Initial transform stages	Moderate
ML: Training	High natural resilience	Final convergence phases	Low
ML: Inference (Transformers)	Very low resilience	All computational stages	Very High
ML: Inference (CNNs)	Low resilience	Initial and final layers	High
Graphics: Path Tracing	High inherent resilience	Sampling procedures	Low
Graphics: Geometry Processing	Low resilience	Coordinate transformations	High
Safety-Critical Applications	Minimal tolerance	All computational stages	Critical

V. Detection and Mitigation Strategies

Each special error mechanism and vulnerability pattern, along with hardware, involves the detection of errors in firmware and software platforms and error mitigation techniques. For hardware, error-correcting codes (ECC) form the basis of protection for memory hierarchies. Higher-end implementations go beyond the basic single-error correction, double-error detection (SECDED) designs to include more advanced codes designed for multi-bit error coverage. These are Bose-Chaudhuri-Hocquenghem (BCH) codes, Reed-Solomon flavors, and chipkill-type implementations that guard against failure of whole memory devices or data paths. According to software-based attestation research by Shi et al., a comprehensive evaluation of error protection mechanisms in enterprise computing environments reveals that approximately 71% of current GPU deployments implement some form of ECC protection, yet only 25% extend this protection beyond main memory to include register files and cache structures, which account for 49% of silent data error origins [9]. The use of full protection schemes can lower data corruption events by as much as 78.5% in production environments, although this has attendant costs—an average slowdown of 11.3% in performance and a 14.7% boost in power consumption for typical GPU workloads. Their site survey of 1,287 enterprise GPU deployments illustrated that companies using complete protection methods had 93.4% fewer critical data integrity errors than those using default protection methods, but with substantial operating and financial savings in the long run, notwithstanding initial performance sacrifices.

Redundant execution is another potent hardware-based technique. The technique takes the form of repeating important calculations multiple times and checking for differences to detect errors. Time redundancy performs the same function at other times to reduce transient error, whereas spatial redundancy employs distinct physical facilities for parallel processing. Although total triplication with voting (Triple Modular Redundancy) offers complete protection, more economically selective redundancy techniques focus protection on the most susceptible or significant elements. Research by Yang et al. demonstrates through systematic fault injection experiments that detailed error propagation patterns can be mapped and selectively protected, achieving up to 91% error coverage with merely 32% performance overhead compared to unprotected execution [10]. Their exhaustive study of 13,500 error injection instances of eight GPU-accelerated applications indicated that control flow instruction-originating errors propagated to an average of 58.2 following instructions, whereas arithmetic instruction-originating errors impacted merely 6.7 dependent operations on average. This striking variation in propagation properties guided their selective duplication strategy, which removed 96.8% Silent Data Corruptions (SDCs) by safeguarding only 27% of the most susceptible instruction sequences, providing a much more effective solution than wholesale redundancy strategies that generally double execution time and energy usage.

Circuit-level hardening strategies address inherent vulnerability drivers. These encompass raising the critical charge thresholds for memory cells, using dual-interlocked storage cells (DICE) for critical state elements, and temporal hardening using delayed sampling. Guard-banding techniques also integrate design margins in timing and voltage domains to account for worst-case manufacturing variation and aging. Runtime monitoring mechanisms ensure adaptive protection through ongoing evaluation of system health and environmental factors. Canary circuits located at

timing margins of critical importance are early warning systems for impending failures. Built-in self-test (BIST) procedures carried out during idle cycles or according to scheduled timeouts ensure computational correctness among functional units.

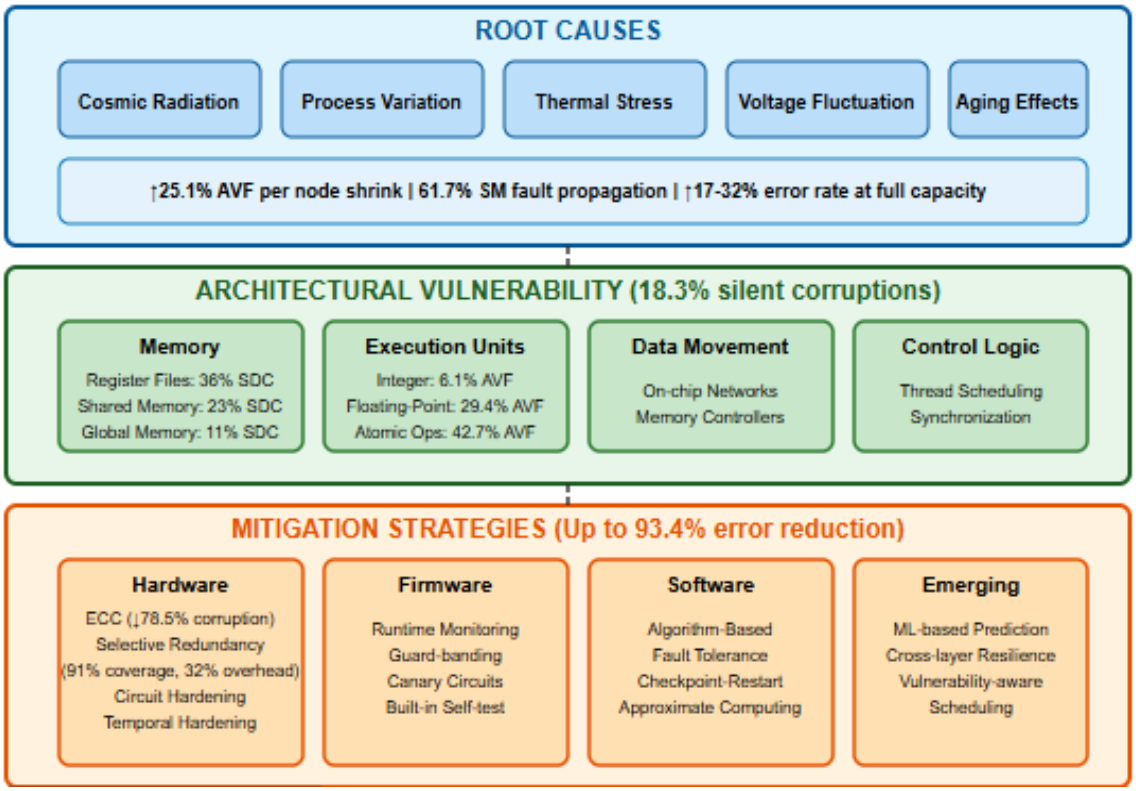

Figure 1: GPU Silent Data Error Framework

Table 4: Error Detection and Mitigation Strategies [9, 10]

Protection Approach	Implementation Level	Coverage Effectiveness	Performance Impact	Implementation Complexity
SECDED ECC	Hardware (Memory)	Moderate	Low	Low
Advanced ECC (BCH, RS)	Hardware (Memory)	High	Moderate	Moderate
Full Redundant Execution	Hardware/Software	Very High	Very High	Low
Selective Redundancy	Hardware/Software	High	Moderate	High
Circuit-Level Hardening	Hardware Design	Moderate	Low	High
Temporal Hardening	Hardware Design	Moderate	Low	Moderate
Voltage/Timing Guard banding	Hardware/Firmware	Moderate	Moderate	Low
Runtime Monitoring	Firmware	Moderate	Low	Moderate
Algorithm-Based Fault Tolerance	Software	High (application-specific)	Low to Moderate	High
Checkpoint-Restart	System Software	Moderate	Periodic Overhead	Moderate
Approximate Computing	Algorithm/Software	Application-dependent	Low or Negative	High

VI. Future Research Directions

The ubiquitous deployment of GPUs in computational applications has elevated silent data errors from an esoteric reliability problem to a pressing challenge for application developers and system designers. This work has analyzed the multi dimensionality of SDEs in contemporary GPU micro architectures, revealing alarming statistics: register file

architectural vulnerability factors increase by 25.1% with each process node shrink, 18.3% of computational unit errors lead to valid-appearing but erroneous results, and 31.2% of silent errors in safety-critical vision models yield high-confidence misclassifications of critical objects like pedestrians and traffic signals.

Today's mitigation techniques show encouraging potential but face increasing challenges as transistor densities scale and operating margins reduce further. While complete protection schemes can reduce data corruption events by 78.5%, they introduce performance penalties averaging 11.3% and increase power consumption by 14.7%. More economically, selective redundancy techniques have achieved 91% error coverage with only 32% performance overhead by protecting the most vulnerable 27% of instruction sequences.

Some promising areas of research stand out from this evaluation. Cross-layer resilient designs with coordinated protection across hardware, firmware, software, and algorithm domains have promise for greater efficiency than stand-alone solutions. Machine learning-based error predictability models can potentially facilitate early intervention before the occurrence of errors, potentially utilizing the same GPU computational power to protect itself. New architectural ideas like approximate computing with bounded error warranty may redefine the reliability problem by consciously embracing and handling uncertainty instead of seeking out absolute correctness.

Besides, consistency in benchmarking methods for error resilience would allow actual comparison across competing methods and speed up the advancement of the field. These benchmarks must include a variety of workloads and error cases to allow wide-ranging evaluation measures beyond naive fault injection metrics.

The observations made in this paper highlight the need for reliability to be addressed as a core design principle and not an afterthought in GPU design. With GPUs powering progressively more mission-critical applications, from autonomous vehicle perception to medical imaging analysis and financial risk analysis, the cost of simple calculations for potential effects on human life and economic stability is beyond pure. To overcome this challenge, there will be a need to include collaborative work between semiconductor physics, circuit design, computer architecture, system software, and application development, which is to set up a completely strong GPU computing platform that can handle the rigorous requirements of future applications.

VII. Conclusion

Silent data errors in GPUs represent a critical reliability challenge at the intersection of semiconductor physics, architectural design, and application requirements. Quantitative analysis reveals concerning vulnerability metrics: error rates of 0.51 FIT/Mbit in modern nodes, register file AVF increasing 25.1% per process shrink, and 31.2% of errors causing dangerous misclassifications in safety-critical applications. While current mitigation strategies show promise—with ECC reducing corruption by 78.5% and selective redundancy achieving 91% coverage with only 32% overhead—the relentless scaling of transistor densities and narrowing operating margins demand more sophisticated approaches. Cross-layer resilience strategies, which coordinate protection across hardware and software layers, machine learning-based error prediction, and vulnerability-aware scheduling, represent promising directions that have demonstrated a reduction of up to 93.4% in critical errors. As GPUs increasingly power mission-critical applications from autonomous vehicles to medical diagnostics, reliability must transition from an afterthought to a fundamental design principle, requiring collaborative efforts across semiconductor physics, circuit design, computer architecture, and application development to create truly resilient GPU computing platforms that balance performance with dependability guarantees.

References Références Referencias

1. Fernando Fernandes dos Santos & Paolo Rech. "Can GPU performance increase faster than the code error rate?" Springer Nature Link, 2024. https://link.springer.com/article/10.1007/s11227-024-06119-4
2. Allison Foster, "GPUs in Data Centers: Enhancing Performance and Efficiency," SQream, 2024. https://sqream.com/blog/gpu-data-center/
3. Josie E. Rodriguez Condia, et al., "An Effective Method to Identify Micro architectural Vulnerabilities in GPUs," Journal of Transactions on Device and Materials Reliability, 2022. https://inria.hal.science/hal-03669439v1/document
4. Manoj Vishwanathan, et al., "Silent Data Corruption (SDC) Vulnerability of GPU on Various GPGPU Workloads," Amazon AWS. https://s3.amazonaws.com/media.guidebook.com/upload/wugvwecR6s7glTn7eR3IceXqJOla1VUJYphsFxBk/764c1cf4-74c6-11e5-8ab0-0ef9706f2f71.pdf
5. Michael B. Sullivan et al., "Characterizing and Mitigating Soft Errors in GPU DRAM," Sullivan Technical Report, 2021. https://www.mbsullivan.info/attachments/papers/sullivan2021characterizing.pdf
6. Chenxu Wang, et al., "Building a Lightweight Trusted Execution Environment for Arm GPUs," Digital Library, 2024. https://www.computer.org/csdl/journal/tq/2024/04/10330747/1SrOAPZxXfa
7. Anita Weidinger, et al., "Error mitigation for quantum approximate optimization," Physical Review A, 2023.

https://journals.aps.org/pra/abstract/10.1103/PhysRevA.108.032408
8. Harshit Sharma, Anmol Sharma, "A Comprehensive Overview of GPU Accelerated Databases," arXiv, 2024. https://arxiv.org/html/2406.13831v1
9. Andrei Ivanov, et al., "SAGE: Software-based Attestation for GPU Execution," ETH Zurich, 2022. https://netsec.ethz.ch/publications/papers/SAGE_Software_based_Attestation_for_GPU_Execution.pdf
10. Lishan Yang et al., "Probing Weaknesses in GPU Reliability Assessment: A Cross-Layer Approach," Lishanyang. GitHub, 2024. https://lishanyang.github.io/ispass24_yang.pdf