BABIS: Exploring the Microarchitectures of Programmable AI Accelerators for Silent Data Corruptions

Odysseas Chatzopoulos
University of Athens
Greece
od.chatzopoulos@di.uoa.gr

Maria Trakosa
University of Athens
Greece
mariatrak@di.uoa.gr

Dimitris Gizopoulos
University of Athens
Greece
dgizop@di.uoa.gr

Abstract

Silent data corruptions (SDCs) due to defective silicon affects data-intensive computations in all computing units: CPUs, GPUs, AI accelerators (AIAs). Accurate measurement of the scale of the problem is necessary to guide hardware or software mitigation strategies. The design space of systolic-array based AIAs is very broad and every piece of the system design matters for the rate and severity of SDCs in these data-parallel architectures. We describe our complete modeling framework for the exploration of the design space of programmable AI accelerators; it flexibly models all necessary circuit, microarchitecture, architecture, and software layers of abstraction that determine how often and in which way silicon defects in AIAs generate SDCs in AI workloads. Such a framework can diligently identify the reliability hotspots of the accelerator design and thus guide informed design decisions for mitigation.

Keywords

AI accelerators, design space exploration, microarchitectural modeling, performance, reliability, simulation, silent data corruption

1 Introduction

Artificial Intelligence (AI) drives innovation across fields like healthcare, autonomous systems, natural language processing, and scientific research, but the growing computational demands, especially in deep learning, have outpaced conventional computing architectures. Specialized hardware accelerators have emerged to optimize AI workloads. While traditionally performance-focused, reliability—particularly Silent Data Corruptions (SDCs) [6]—has become equally critical for large-scale deployments. Hyperscalers (Meta, Google, Alibaba) report widespread SDCs in CPUs and AI accelerators, with defect-induced errors arising from manufacturing, aging, variability, or environmental factors. CPU studies report SDC rates of around *one core per thousand*, similarly observed in accelerators. Although extensive research has examined SDC estimation and detection for CPUs, AI accelerators' unique data-flow architectures and workloads demand tailored solutions. Early-stage accelerator modeling is essential to evaluate performance and reliability quickly and accurately. Microarchitectural simulators effectively balance accuracy and throughput, providing detailed insights into compute units, memory hierarchies, and dataflows. In contrast, software-based fault injection lacks hardware realism, and RTL/gate-level simulations are accurate but prohibitively slow.

We introduce an open-source framework built on *gem5*, supporting comprehensive full-stack modeling and reliability assessment of programmable systolic-array-based AI accelerators. Our approach integrates statistical fault injection targeting SRAMs, registers, and functional units (covering *>90% accelerator area*), along with efficient gate-level fault modeling embedded directly into gem5. Key contributions include: **(i)** full-stack AI accelerator modeling with cycle-level accuracy, **(ii)** comprehensive fault injection across multiple fault types, **(iii)** extensive configurability supporting diverse systolic array sizes, dataflows, datatypes, and memory hierarchies, **(v)** direct evaluation of performance-reliability tradeoffs, **(vi)** ability to gather microarchitectural insights guiding accelerator optimization, and **(vii)** observation of ML model behavior under various fault conditions. Figure 1 compares our modeled system components with state-of-the-art tools [4, 5, 7, 9–12].

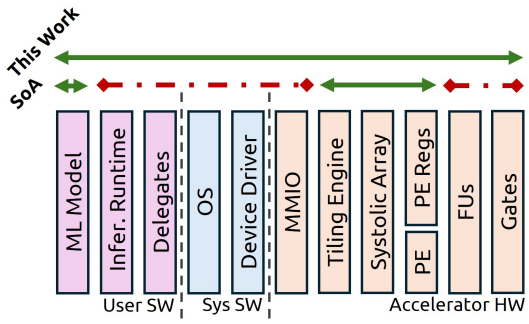

Figure 1: Our modeled compute stack vs. the State-of-the-Art. Many important pieces are missing (red) from existing tools with our work filling all the gaps.

2 Background

2.1 Defects, Faults, and Errors

Hardware defects in AI accelerators originate from manufacturing issues, aging, or environmental conditions, manifesting as faults. These faults can be classified into three types: (i) *transient*, temporary disruptions such as particle strikes; (ii) *intermittent*, sporadic faults from marginal defects or thermal conditions; and (iii) *permanent*, resulting from irreversible damage. Faults may produce errors like *Silent Data Corruptions (SDCs)*, which degrade neural network accuracy without detection, or *Detected Unrecoverable Errors (DUEs)*, which trigger immediate disruptions but allow diagnosis.

2.2 Reliability Metrics for AI Accelerators

Reliability is commonly assessed using the Architectural Vulnerability Factor (AVF), quantifying the likelihood a fault will cause a visible error based on microarchitecture, software, and inputs. Given the data-flow nature of AI accelerators, injected faults primarily manifest as SDCs. We focus on four distinct SDC metrics: (i) **Top1-Class SDC**, faults changing the predicted top class; (ii) **Top1-Confidence SDC**, altering confidence scores of the top prediction; (iii) **Top5-Class SDC**, faults affecting any of the top five predicted classes; and (iv) **Top5-Confidence SDC**, faults altering

COGARCH @ ISCA 2025, June 22, 2025, Tokyo, Japan

confidence scores among the top five predictions. Our framework flexibly supports these metrics, capturing SDCs across all modeled accelerator components.

3 Methodology

For neural network modeling and training, we use TensorFlow, with inference executed through TensorFlow Lite. We implement a custom TensorFlow Lite delegate to seamlessly offload major operations—fully connected, convolutional, and batch matrix multiplication layers—onto the modeled AI accelerator, significantly improving simulation efficiency and accuracy.

Our framework is built upon gem5, a widely used cycle-level microarchitectural simulator. Our modeled SoC architecture (Figure 2) employs memory-mapped I/O (MMIO) supported through a heavily modified version of the gem5-accel framework [13], incorporating a custom DMA engine with virtual addressing capabilities. Custom drivers abstract hardware details and facilitate software integration.

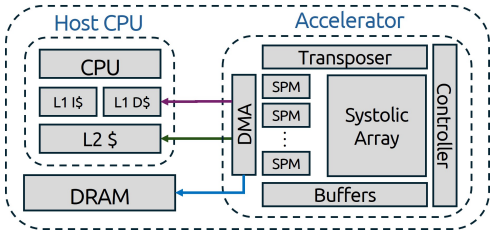

Figure 2: Architecture of the modeled System-on-Chip. Arrows show the different locations in the memory hierarchy where the accelerator can be connected.

Our accelerator (Figure 2) is centered around a configurable systolic array supporting multiple dataflows (Weight-Stationary, Input-Stationary, Output-Stationary [8]) and arbitrary datatypes. It includes buffers, scratchpad memories, a transposer, a DMA engine, and a control module responsible for dataflow-specific operations. Further details on systolic array dataflows, tiling strategies, and datatype support are described extensively in prior work [8]. Reliability is assessed through statistical fault injection (SFI) targeting SRAMs, registers, and functional units, using methodologies described in prior CPU and GPU reliability studies (e.g [3]). Gate-level faults in functional units are efficiently modeled within gem5, following techniques detailed in [1, 2].

4 Results

Due to space constraints, this section focuses exclusively on the impact of permanent faults on accelerator reliability. We present the Top-1 and Top-5 class P_{SDC} for three accelerator components: Processing Element (PE) registers, Weights Scratchpad Memory (SPM), and PE Multiply-Accumulate Functional Units (FUs), across several neural network models. Figure 3 shows P_{SDC} of these components for different accelerator sizes (N) and dataflows.

Figure 4 provides a breakdown of P_{SDC} for each individual workload analyzed. Our evaluation covers four datasets (CIFAR-10, CIFAR-100, MNIST, and Fashion-MNIST) and six distinct neural network architectures. However, it is important to note that the results presented here represent only a small subset of the extensive design-space exploration possibilities enabled by our framework, highlighting its capability to examine diverse accelerator configurations comprehensively.

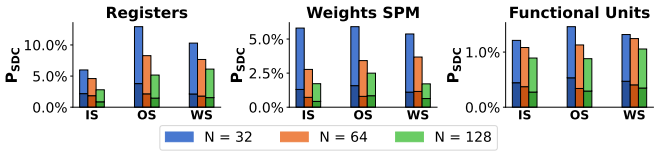

Figure 3: Top1-Class (dark) and Top5-Class (light) P_{SDC} of PE Registers, Weights SPM and FUs for different values of N and dataflow, averaged across workloads.

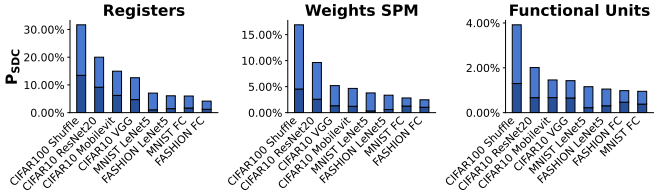

Figure 4: Top1-Class (dark) and Top5-Class (light) P_{SDC} for each benchmark due to permanent faults in PE registers, Weights SPM and FUs aggregated for all values of N, dataflow.

References

[1] O. Chatzopoulos, N. Karystinos, G. Papadimitriou, D. Gizopoulos, H. D. Dixit, and S. Sankar. 2025. Veritas – Demystifying Silent Data Corruptions: μArch-Level Modeling and Fleet Data of Modern x86 CPUs. In *2025 IEEE International Symposium on High-Performance Computer Architecture (HPCA)*.

[2] O. Chatzopoulos, G. Papadimitriou, D. Gizopoulos, H. D. Dixit, and S. Sankar. 2025. From Gates to SDCs: Understanding Fault Propagation Through the Compute Stack. In *2025 Design, Automation and Test in Europe Conference (DATE)*.

[3] O. Chatzopoulos, G. Papadimitriou, V. Karakostas, and D. Gizopoulos. 2024. Gem5-MARVEL: Microarchitecture-Level Resilience Analysis of Heterogeneous SoC Architectures. In *2024 IEEE International Symposium on High-Performance Computer Architecture (HPCA)*. IEEE Computer Society, Los Alamitos, CA, USA, 543–559. https://doi.org/10.1109/HPCA57654.2024.00047

[4] Zitao Chen, Guanpeng Li, Karthik Pattabiraman, and Nathan DeBardeleben. 2019. Binfi: An efficient fault injector for safety-critical machine learning systems. In *Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis*. 1–23.

[5] Zitao Chen, Niranjhana Narayanan, Bo Fang, Guanpeng Li, Karthik Pattabiraman, and Nathan DeBardeleben. 2020. Tensorfi: A flexible fault injection framework for tensorflow applications. In *2020 IEEE 31st International Symposium on Software Reliability Engineering (ISSRE)*. IEEE, 426–435.

[6] Dimitris Gizopoulos. 2024. SDCs: A B C. https://www.sigarch.org/sdcs-a-b-c/

[7] Yi He, Prasanna Balaprakash, and Yanjing Li. 2020. Fidelity: Efficient resilience analysis framework for deep learning accelerators. In *2020 53rd Annual IEEE/ACM International Symposium on Microarchitecture (MICRO)*. IEEE, 270–281.

[8] S. Kung. 1985. VLSI Array processors. *IEEE ASSP Magazine* 2, 3 (1985), 4–22. https://doi.org/10.1109/MASSP.1985.1163741

[9] Abdulrahman Mahmoud, Neeraj Aggarwal, Alex Nobbe, Jose Rodrigo Sanchez Vicarte, Sarita V Adve, Christopher W Fletcher, Iuri Frosio, and Siva Kumar Sastry Hari. 2020. Pytorchfi: A runtime perturbation tool for dnns. In *2020 50th Annual IEEE/IFIP International Conference on Dependable Systems and Networks Workshops (DSN-W)*. IEEE, 25–31.

[10] Salvatore Pappalardo, Annachiara Ruospo, Ian O'Connor, Bastien Deveautour, Ernesto Sanchez, and Alberto Bosio. 2023. A Fault Injection Framework for AI Hardware Accelerators. In *2023 IEEE 24th Latin American Test Symposium (LATS)*. 1–6. https://doi.org/10.1109/LATS58125.2023.10154505

[11] Jingweijia Tan, Qixiang Wang, Kaige Yan, Xiaohui Wei, and Xin Fu. 2023. Saca-FI: A microarchitecture-level fault injection framework for reliability analysis of systolic array based CNN accelerator. *Future Generation Computer Systems* 147 (2023), 251–264.

[12] Abhishek Tyagi, Yiming Gan, Shaoshan Liu, Bo Yu, Paul Whatmough, and Yuhao Zhu. 2024. Thales: Formulating and Estimating Architectural Vulnerability Factors for DNN Accelerators. arXiv:2212.02649 [cs.AR] https://arxiv.org/abs/2212.02649

[13] João Vieira, Nuno Roma, Gabriel Falcao, and Pedro Tomás. 2023. gem5-accel: A Pre-RTL Simulation Toolchain for Accelerator Architecture Validation. *IEEE Computer Architecture Letters* (2023).