

# Circuit-Aware Analysis of Arithmetic Error Detection Codes

Cheng Chiu 

Computer Science & Engineering  
University of Michigan  
Ann Arbor, MI, USA  
ccchiu@umich.edu

Keyon Mazandarani 

Computer Science & Engineering  
University of Michigan  
Ann Arbor, MI, USA  
keyonm@umich.edu

Nathan Bleier 

Computer Science & Engineering  
University of Michigan  
Ann Arbor, MI, USA  
nbleier@umich.edu

**Abstract**—In modern CMOS VLSI circuits, arithmetic datapaths are increasingly vulnerable to radiation-induced soft errors and circuit-aging, leading to application crashes or silent data corruption. However, existing studies lack a gate-level framework for evaluating the efficacy of arithmetic error-detection codes in the presence of logic and timing masking. We focus on arithmetic codes—such as AN codes and Redundant Residue Number System codes—that natively support arithmetic operations, unlike conventional linear block codes. We prioritize detection over correction given the low cost of re-execution and the rarity of radiation strikes relative to compute throughput. Because 48-bit values and pointers typically suffice in contemporary systems, we adopt a 48+16-bit (data+redundancy) organization. We introduce a statistical framework that models radiation strikes by injecting LET-based pulse-widths into netlists and emulates circuit-aging through delay scaling. The toolchain enables designers to screen arithmetic code schemes before tape-out and tailor resilience for deployments ranging from sea-level cosmic-ray exposure to the high-radiation conditions such as particle accelerators, spacecraft, and defense electronics. By exposing how code parameters and circuit topology interact at the gate level, the framework fosters tightly coupled code-hardware co-design.

## I. INTRODUCTION

In modern VLSI systems—ranging from datacenters to radiation-intensive aerospace electronics—radiation and aging induce low-level faults that evade detection, causing silent data corruption (SDC) [1]–[11]. Extensive prior work evaluates these phenomena via fault injection targeting specific abstraction layers [12]–[18]. While memory is protected by error-correcting codes (ECCs) [19]–[24], arithmetic datapaths remain vulnerable despite dominating silicon area in modern processors (e.g., wide-SIMD and systolic-array accelerators); for instance, in Google’s TPU v1, the arithmetic datapath accounts for nearly two-thirds of the die area [25]. However, these datapaths often lack comparable protection, leaving systems vulnerable to corruption of both bulk data and critical control metadata (e.g., pointers). Notably, in a Tiny DNN classifier, corrupting pointer-like metadata causes failure rates comparable to data corruption despite occupying only 1.8% of program memory. Moreover, adversaries can intentionally accelerate wear-out via “malicious aging” attacks [26].

We present a gate-level analysis framework (Fig. 1) that validates arithmetic codes (ACs) as a low-overhead alternative to modular redundancy [27]. Because ACs natively support arithmetic and fit cleanly at the ISA/ $\mu$ Arch boundary, a 48+16-

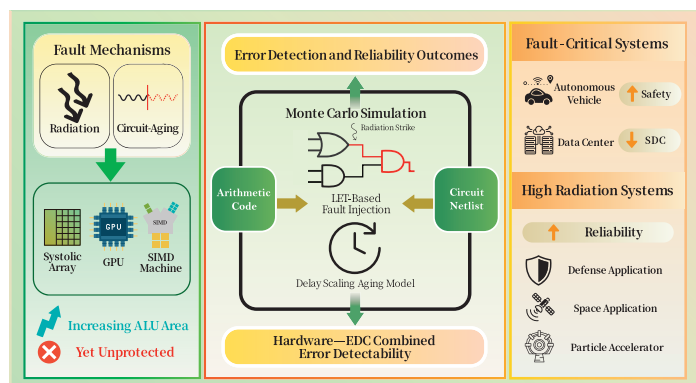


Fig. 1. Summary of contributions. (Left) Reliability concerns in growing arithmetic datapaths. (Middle) Our proposed statistical framework. (Right) Targeting safety-critical and high-radiation systems.

bit (data+redundancy) organization is practical on modern 64-bit ISAs where values and pointers typically fit within 48 bits. The Monte Carlo fault modeling framework provides independent evaluation modes for radiation-induced Single-Event Transient (SET) pulse-widths and aging-induced delay scaling. We apply this analysis to standard datapath topologies, enabling rigorous pre-tapeout error-detection code (EDC)–hardware co-design.

## II. ARITHMETIC CODES

Traditional ECC fails in arithmetic datapaths as its finite field  $GF(2^n)$  linearity is incompatible with integer modular arithmetic  $\mathbb{Z}/2^n$ . ACs resolve this mismatch by maintaining linearity under arithmetic operations. They are commonly categorized as: (i) **AN codes** [28], [29], which encode via multiplicative scaling ( $x \mapsto Ax$ ), but require additional treatment for multilinear operations. (ii) **Redundant Residue Number System (RRNS) codes** [30]–[32] represent values as residue vectors ( $x \mapsto (x \bmod m_1, \dots, x \bmod m_n)$ ), enabling parallel arithmetic and error checking with choices such as  $2^n$  moduli that directly support standard  $n$ -bit operations. The fault tolerance of ACs is determined by the arithmetic weight rather than the Hamming weight [29], [33].

### III. METHODOLOGY

*a) Hardware & EDC Configuration:* We evaluate three 48+16-bit (data+EDC) adder topologies—ripple-carry adder (RCA) ( $\Theta(n)$ ), carry-select adder (CSA) ( $\Theta(\sqrt{n})$ ), and Kogge-Stone adder (KSA) ( $\Theta(\log n)$ )—and a synthesizer-optimized multiplier, synthesized with NanGate 45 nm library [34]. AN and unprotected RCA, CSA, and KSA operate at  $T_{\text{clk,best}} = \{3.25, 1.17, 0.75\}$  ns, respectively; with RRNS protection, the corresponding adders operate at  $\{2.61, 1.00, 0.78\}$  ns, and the RRNS multiplier at 5.74 ns. We emphasize error detection over correction given the low cost of re-execution. We compare these baselines against two EDC schemes: AN codes ( $A = 64709$ ) and RRNS codes (Moduli  $\{3, 67, 101, 2^{48}\}$ ).

*b) Injection Framework:* To emulate SETs manifested as transient voltage pulses in combinational logic [1], [2], [35], we developed an open-sourced SystemVerilog injection framework that sweeps pulse-width scaling factors ( $f_{PW}$ ) and initial bit-flips ( $n$ ). Here,  $f_{PW}$  scales a nominal linear energy transfer (LET)-sampled pulse-width to model faster adders, and  $n$  denotes the number of simultaneous upset sites to model multi-node strike events. For each configuration, we execute  $10^6$  Monte Carlo trials using a two-pass simulation flow: Pass 1 injects SET pulses with annotated gate delays, while Pass 2 re-simulates the same injections under zero-delay conditions to extract logic masking effects.<sup>1</sup>

*c) Fault Mechanisms:* For radiation-induced SETs, we assume uniform strikes with pulse-widths  $t_{PW}$  derived from LET values drawn from a truncated normal distribution ( $\mu = 35.0, \sigma = 15.0$  MeV cm<sup>2</sup>/mg) [36]–[38]. For circuit-aging, we emulate delay degradation by scaling back-annotated Standard Delay Format (SDF) delays by  $f_{\text{aging}} \in \{1.3\times, 1.6\times\}$ .

### IV. RESULTS

**Error Rates and Masking Trends:** Fig. 2 shows that wider pulse-widths increase error rates by reducing timing masking. Similarly, increasing initial bit-flips ( $n$ ) raises error rates and reduces logic masking by activating more erroneous paths. While faster adders are generally more susceptible to SET at their optimal clock speeds ( $T_{\text{clk,best}}$ ), fixed-frequency analysis (Table I) reveals that RCA is the most vulnerable compared to CSA and KSA. This suggests that the higher susceptibility of faster adders stems primarily from shorter clock periods (reduced timing masking) rather than architectural properties

TABLE I  
SOFT ERROR RATE (SER) (%) OF ADDERS AT  $T_{\text{clk}} = 3.25$  NS (NO EDC).

Initial Flips $n$	RCA	CSA	KSA
1	1.1866	0.8556	0.8827
8	5.5884	4.6220	5.0419

**Undetected Errors:** Undetected errors in RRNS were observed exclusively at  $n = 8$  (single-digit counts per million trials), whereas AN codes yielded none. This highlights a trade-off: AN codes provide stronger error detectability, while RRNS

<sup>1</sup>In multi-bit-flip scenarios, we classify a fault as logic-masked only if inverting *all* selected wires produces no error.

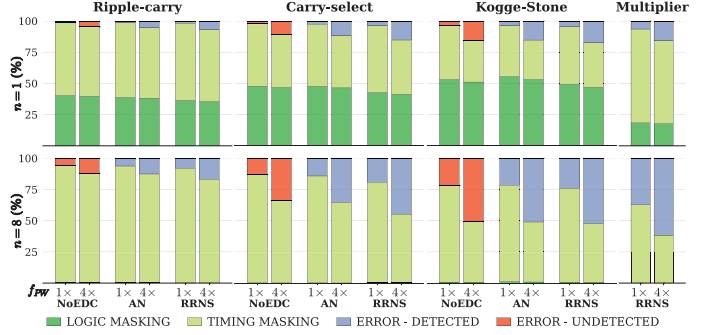


Fig. 2. Error-masking breakdown of a SET causing  $n \in \{1, 8\}$  initial bit-flips in adders and multiplier at their  $T_{\text{clk,best}}$ , grouped by EDC and pulse-width factors  $f_{PW} \in \{1\times, 4\times\}$ .

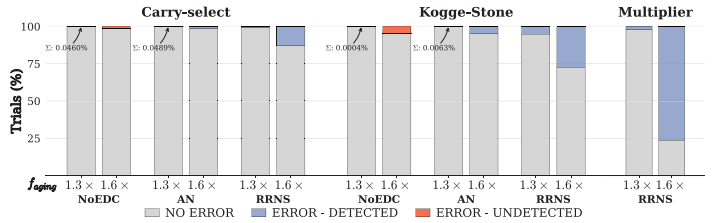


Fig. 3. Circuit-aging results for adders and multiplier at their  $T_{\text{clk,best}}$ , grouped by EDC and delay scaling factors  $f_{\text{aging}} \in \{1.30\times, 1.60\times\}$ . All RCA aging experiments show no errors under these settings, so they are omitted.  $\Sigma$  in the annotations denotes the total error rate (detected + undetected).

supports native multilinear operations and parallelization at the cost of slightly lower detection coverage.

**Circuit-Aging:** In Fig. 3, RRNS-based adders generally exhibit higher aging-induced error rates than their AN counterparts. Note that the RCA showed no errors for  $f_{\text{aging}} \in \{1.3\times, 1.6\times\}$  due to the low probability of inputs exercising its full critical path. In contrast, the multiplier showed significant sensitivity to aging-induced degradation compared to adders.

### V. CONCLUSION AND ARCHITECTURAL IMPLICATIONS

Modern workloads rely heavily on arithmetic datapaths, yet these regions remain vulnerable to radiation- and aging-induced errors compared to protected memory hierarchies. We demonstrated that arithmetic codes (AN, RRNS) offer a practical, low-overhead mitigation without requiring full replication. We propose a 48+16-bit organization that prioritizes error detection over correction, integrating cleanly into existing register files. By performing computations in the encoded domain and decoding only at memory boundaries, the architecture protects data and control metadata while preserving cache and TLB spatial locality. This approach extends to fixed- and floating-point units, potentially allowing for reduced voltage guardbands.

To quantify these benefits, our gate-level statistical fault injection framework provides early visibility into SER and masking, enabling the tight code-hardware co-design essential for hardening ASICs in high-radiation and safety-critical domains.

## REFERENCES

- [1] P. Dodd and L. Massengill, "Basic mechanisms and modeling of single-event upset in digital microelectronics," *IEEE Transactions on Nuclear Science*, vol. 50, no. 3, pp. 583–602, 2003.
- [2] R. Baumann, "Soft errors in advanced computer systems," *IEEE Design & Test of Computers*, vol. 22, no. 3, pp. 258–266, 2005.
- [3] S. Mukherjee, J. Emer, and S. Reinhardt, "The soft error problem: an architectural perspective," in *11th International Symposium on High-Performance Computer Architecture*, 2005, pp. 243–247.
- [4] P. Shivakumar, M. Kistler, S. Keckler, D. Burger, and L. Alvisi, "Modeling the effect of technology trends on the soft error rate of combinational logic," in *Proceedings International Conference on Dependable Systems and Networks*, 2002, pp. 389–398.
- [5] M. Riera, R. Canal, J. Abella, and A. Gonzalez, "A detailed methodology to compute soft error rates in advanced technologies," in *2016 Design, Automation & Test in Europe Conference & Exhibition (DATE)*, 2016, pp. 217–222.
- [6] D. K. Schroder and J. A. Babcock, "Negative bias temperature instability: Road to cross in deep submicron silicon semiconductor manufacturing," *Journal of Applied Physics*, vol. 94, no. 1, pp. 1–18, 07 2003. [Online]. Available: <https://doi.org/10.1063/1.1567461>
- [7] C. Hu *et al.*, "Hot-electron-induced mosfet degradation - model, monitor, and improvement," *IEEE Journal of Solid-State Circuits*, vol. 20, no. 1, pp. 295–305, 1985.
- [8] J. Black, "Electromigration—a brief survey and some recent results," *IEEE Transactions on Electron Devices*, vol. 16, no. 4, pp. 338–347, 1969.
- [9] H. D. Dixit *et al.*, "Silent data corruptions at scale," 2021. [Online]. Available: <https://arxiv.org/abs/2102.11245>
- [10] O. Chatzopoulos *et al.*, "Veritas – demystifying silent data corruptions: march-level modeling and fleet data of modern x86 cpus," in *2025 IEEE International Symposium on High Performance Computer Architecture (HPCA)*, 2025, pp. 1–14.
- [11] S. Kamat, M. Zheng, B. Fang, and D. Dai, "Be aware of metadata corruption in parallel file system: It can be silent and catastrophic," in *2025 IEEE International Parallel and Distributed Processing Symposium (IPDPS)*, 2025, pp. 1–13.
- [12] S. Nema *et al.*, "Eris: Fault injection and tracking framework for reliability analysis of open-source hardware," in *2022 IEEE International Symposium on Performance Analysis of Systems and Software (ISPASS)*, 2022, pp. 210–220.
- [13] P. Omland *et al.*, "Hpc hardware design reliability benchmarking with hdfit," *IEEE Transactions on Parallel and Distributed Systems*, vol. 34, no. 3, pp. 995–1006, 2023.
- [14] O. Chatzopoulos, G. Papadimitriou, V. Karakostas, and D. Gizopoulos, "Gem5-marvel: Microarchitecture-level resilience analysis of heterogeneous soc architectures," in *2024 IEEE International Symposium on High-Performance Computer Architecture (HPCA)*, 2024, pp. 543–559.
- [15] S. K. S. Hari, T. Tsai, M. Stephenson, S. W. Keckler, and J. Emer, "Sassifi: An architecture-level fault injection tool for gpu application resilience evaluation," in *2017 IEEE International Symposium on Performance Analysis of Systems and Software (ISPASS)*, 2017, pp. 249–258.
- [16] Q. Lu, M. Farahani, J. Wei, A. Thomas, and K. Pattabiraman, "Llfi: An intermediate code-level fault injection tool for hardware faults," in *2015 IEEE International Conference on Software Quality, Reliability and Security*, 2015, pp. 11–16.
- [17] A. Mahmoud *et al.*, "Pytorchfi: A runtime perturbation tool for dnns," in *2020 50th Annual IEEE/IFIP International Conference on Dependable Systems and Networks Workshops (DSN-W)*, 2020, pp. 25–31.
- [18] Z. Chen *et al.*, "Tensorfi: A flexible fault injection framework for tensorflow applications," in *2020 IEEE 31st International Symposium on Software Reliability Engineering (ISSRE)*, 2020, pp. 426–435.
- [19] I. S. Reed and G. Solomon, "Polynomial codes over certain finite fields," *Journal of the Society for Industrial and Applied Mathematics*, vol. 8, no. 2, pp. 300–304, 1960. [Online]. Available: <https://doi.org/10.1137/0108018>
- [20] D. A. Patterson, G. Gibson, and R. H. Katz, "A case for redundant arrays of inexpensive disks (raid)," in *Proceedings of the 1988 ACM SIGMOD International Conference on Management of Data*, ser. SIGMOD '88. New York, NY, USA: Association for Computing Machinery, 1988, p. 109–116. [Online]. Available: <https://doi.org/10.1145/50202.50214>
- [21] R. W. Hamming, "Error detecting and error correcting codes," *The Bell System Technical Journal*, vol. 29, no. 2, pp. 147–160, 1950.
- [22] M. Y. Hsiao, "A class of optimal minimum odd-weight-column sec-ded codes," *IBM Journal of Research and Development*, vol. 14, no. 4, pp. 395–401, 1970.
- [23] T. J. Dell, "A white paper on the benefits of chipkill-correct ecc for pc server main memory," *IBM Microelectronics division*, vol. 11, no. 1-23, pp. 5–7, 1997.
- [24] J. Kim, M. Sullivan, and M. Erez, "Bamboo ecc: Strong, safe, and flexible codes for reliable computer memory," in *2015 IEEE 21st International Symposium on High Performance Computer Architecture (HPCA)*, 2015, pp. 101–112.
- [25] N. P. Jouppi *et al.*, "In-datacenter performance analysis of a tensor processing unit," in *2017 ACM/IEEE 44th Annual International Symposium on Computer Architecture (ISCA)*. Los Alamitos, CA, USA: IEEE Computer Society, Jun. 2017, pp. 1–12. [Online]. Available: <https://doi.ieeecomputersociety.org/10.1145/3079856.3080246>
- [26] N. Karimi, A. K. Kanuparthi, X. Wang, O. Sinanoglu, and R. Karri, "Magic: Malicious aging in circuits/cores," *ACM Trans. Archit. Code Optim.*, vol. 12, no. 1, Apr. 2015. [Online]. Available: <https://doi.org/10.1145/2724718>
- [27] C. Bolchini, A. Miele, and M. D. Santambrogio, "Tmr and partial dynamic reconfiguration to mitigate seu faults in fpgas," in *22nd IEEE International Symposium on Defect and Fault-Tolerance in VLSI Systems (DFT 2007)*, 2007, pp. 87–95.
- [28] A. Avizienis, "Arithmetic error codes: Cost and effectiveness studies for application in digital system design," *IEEE Transactions on Computers*, vol. C-20, no. 11, pp. 1322–1331, 1971.
- [29] W. W. Peterson and E. J. Weldon, *Error-correcting codes*. MIT press, 1972.
- [30] H. L. Garner, "The residue number system," *IRE Transactions on Electronic Computers*, vol. EC-8, no. 2, pp. 140–147, 1959.
- [31] J. James and A. Pe, "Error correction based on redundant residue number system," in *2015 IEEE International Conference on Electronics, Computing and Communication Technologies (CONECCT)*, 2015, pp. 1–5.
- [32] Y. Zhang, "An FPGA implementation of redundant residue number system for low-cost fast speed fault-tolerant computations," Master of Engineering thesis, Nanyang Technological University, Singapore, 2018.
- [33] T. R. Rao, *Error coding for arithmetic processors*. Elsevier, 1974.
- [34] NanGate, Inc., "Nangate 45nm open cell library," [http://www.nangate.com/?page\\_id=2325](http://www.nangate.com/?page_id=2325), 2008.
- [35] E. Normand, "Single event upset at ground level," *IEEE Transactions on Nuclear Science*, vol. 43, no. 6, pp. 2742–2750, 1996.
- [36] H. Jiang *et al.*, "Impact of particle let on combinational logic single-event effects for advanced technologies," in *2016 16th European Conference on Radiation and Its Effects on Components and Systems (RADECS)*, 2016, pp. 1–4.
- [37] S. Azimi, C. De Sio, and L. Sterpone, "Analysis of radiation-induced transient errors on 7 nm finfet technology," *Microelectronics Reliability*, vol. 126, p. 114319, 2021, proceedings of ESREF 2021, 32nd European Symposium on Reliability of Electron Devices, Failure Physics and Analysis. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0026271421002857>
- [38] D. M. Fleetwood, "Radiation effects in a post-moore world," *IEEE Transactions on Nuclear Science*, vol. 68, no. 5, pp. 509–545, 2021.