Not All Faults Are Equal: Transient-Fault Sensitivity Characterization of an Open-Source RISC-V Vector Cluster

University of Bologna

Maoyuan Cai maoyuan.cai@unibo.it

Amirhossein Kiamarzi amirhossein.kiamarz2@unibo.it

Davide Rossi davide.rossi@unibo.it

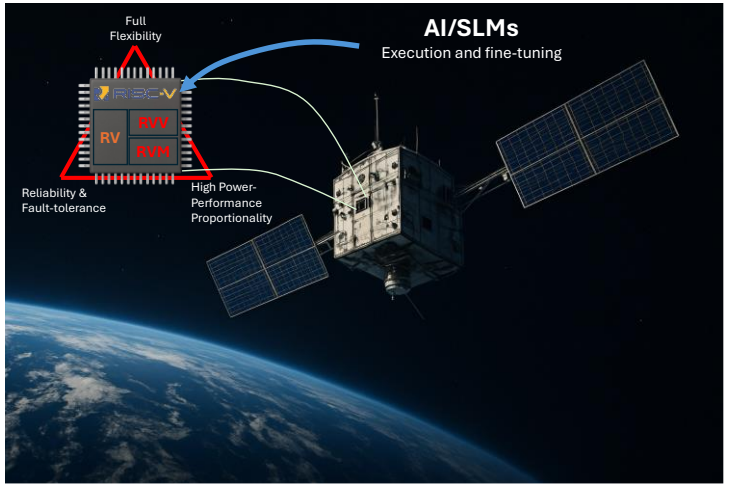
Angelo Garofalo angelo.garofalo@unibo.it

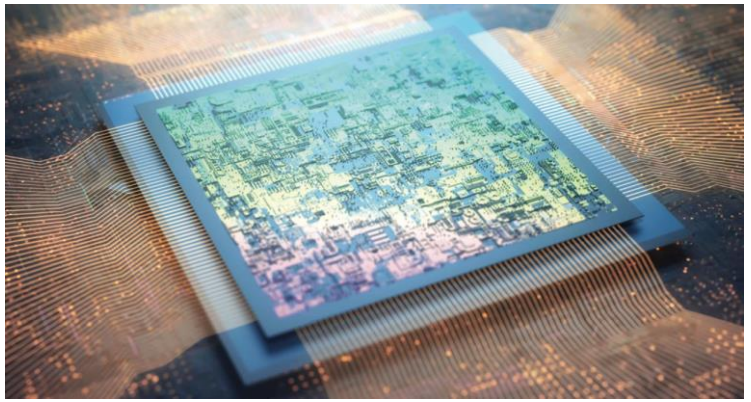
PULP Platform
Open Source Hardware, the way it should be!

@pulp_platform
pulp-platform.org
youtube.com/pulp_platform

Introduction

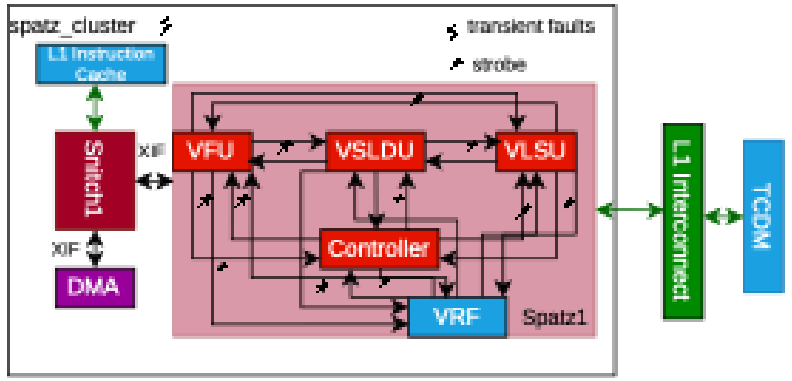
- **Space systems require on-board AI/DSP processing**
  - heterogeneous DSP-/AI-accelerated systems.
  - **vector processors** stand out as a promising paradigm
- **Fault resilience requirements**
  - radiation-induced transient faults (SETs/SEUs).
  - Potential consequences: system crash, Silent Data Corruption, AI accuracy degradation, etc…
- **Uniform hardening is too costly**
  - Conventional protection, e.g. rad-had technology, TMR/DMR, etc, cost too much area for AI accelerators





Introduction – design sensitivity study

- **Research questions:**
  - Which transient faults manifest as errors?
  - Where do they manifest?
  - What consequences do they cause?



- **Related work**

Prior work	Examples	Limitations	This work
Software-only end-to-end studies	ENFOR-SA [10]	miss hardware-level propagation effects	RTL observability at cluster and module level
Python-based evaluation + RTL FI	Udit Kumar Agarwal et al. [1] Sifi-ai [4]	fall short modeling single-event transients; mostly target stateless GEMM accelerators	SET + SEU campaigns on instruction-based vector cluster
Z01X-based analysis	Abhishek Tyagi et al. [11]	do not account for system crashes; remain limited to systolic accelerators	FS vs FD classification plus numerical SDC severity

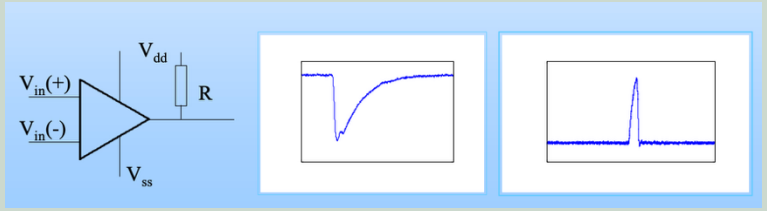
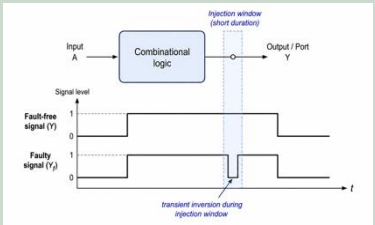
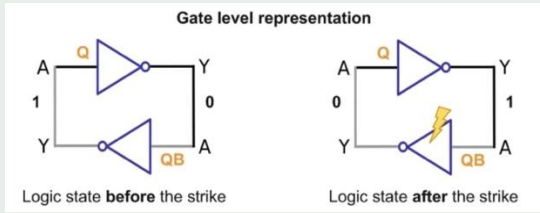
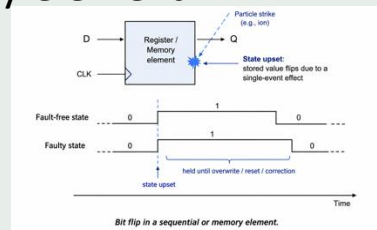
[10] Tonetto, Rafael Billig, et al. "ENFOR-SA: End-to-end Cross-layer Transient Fault Injector for Efficient and Accurate DNN Reliability Assessment on Systolic Arrays." *arXiv preprint arXiv:2602.00909* (2026).
[1] Udit Kumar Agarwal et al. 2023. Towards reliability assessment of systolic arrays against stuck-at faults. In 2023 53rd Annual IEEE/IFIP International Conference on Dependable Systems and Networks-Supplemental Volume (DSN-S). IEEE, 230–236.
[4] Julian Hoefer et al. 2023. Sifi-ai: A fast and flexible rtl fault simulation framework tailored for ai models and accelerators. In Proceedings of the Great Lakes Symposium on VLSI 2023. 287–292.
[12] Tong Xinck et al. 2025. Mitigating multiple single-event upsets during deep neural network inference using fault-aware training. Journal of Instrumentation 20, 02 (2025), C02044.
[11] Abhishek Tyagi et al. 2024. Characterizing Soft-Error Resiliency in Arm's Ethos-U55 Embedded Machine Learning Accelerator. In 2024 IEEE International Symposium on Performance Analysis of Systems and Software (ISPASS). IEEE, 97–108.

Contribution

- **C1: Full-cluster RTL fault-injection methodology**
  - Fault injection is performed on the whole Spatz cluster
  - under both SET and SEU models
  - observation points to distinguish faulty system crashes from faulty data corruption.
- **C2: SDC-oriented precision and bit-field analysis**
  - Quantifies how FP32, FP16, BP16, and FP8 behave under operand-level corruptions
  - Separate the impact of sign, exponent, and mantissa faults.
- **C3: Module-level sensitivity characterization and design guidance**
  - Identify the most sensitive microarchitectural components
  - Derive selective protection strategies for low-overhead reliability improvement.

Background

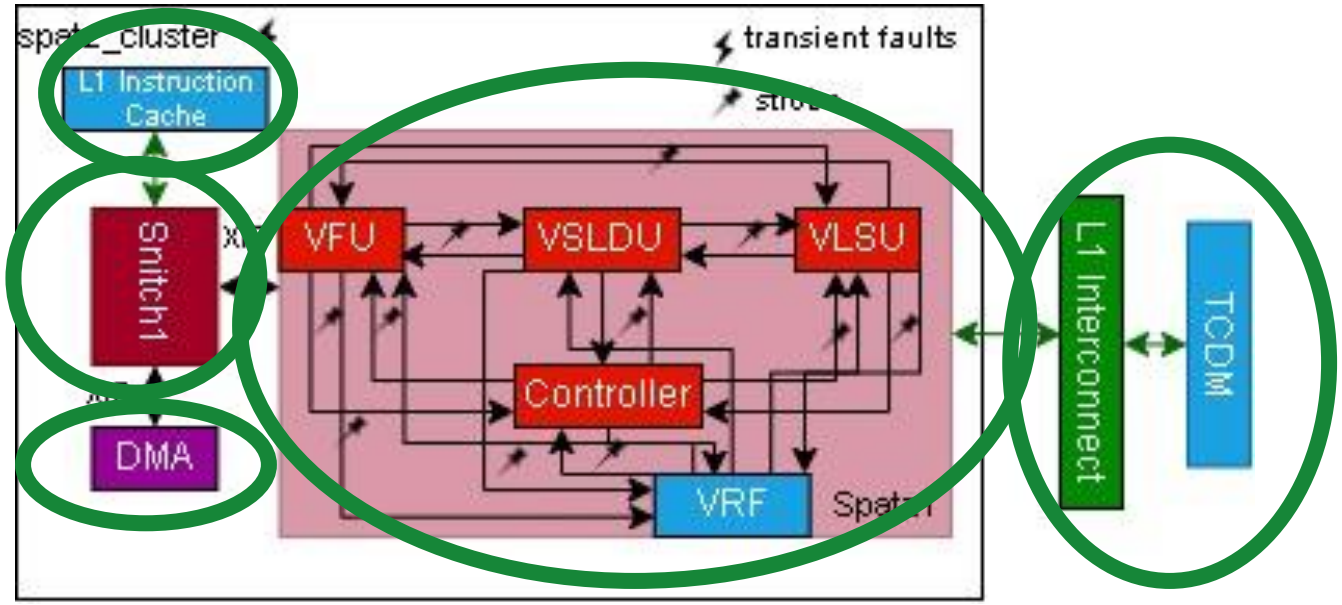
Transient fault models

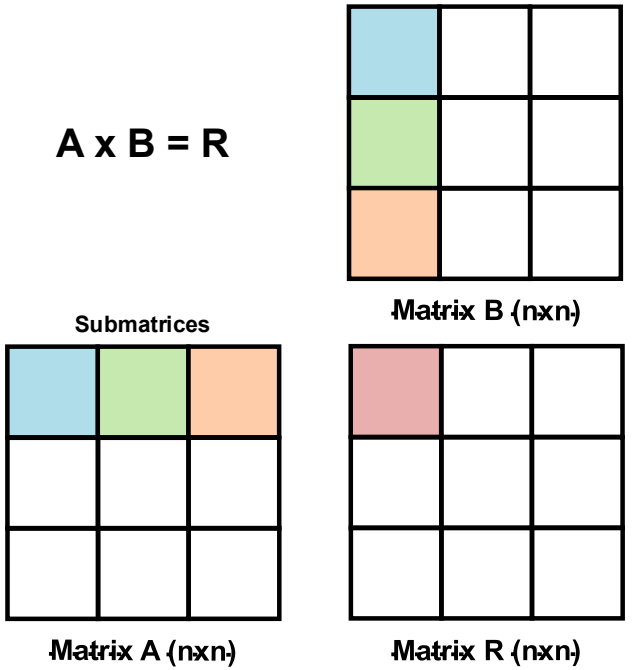
Fault Models	Physical Definition	RTL Abstraction
Single Event Transient (SET)	Momentary voltage excursion at a circuit node induced by particle-generated charge [1] 	Temporary inversion of a **combinational** signal or port 
Single Event Upset (SEU)	Observable storage-state upset caused by localized energy deposition from a single particle [2] 	Temporary bit flip in a **sequential** or memory element 

[1] https://ecss.nl/?item=2&glossary_id=101
[2] https://ecss.nl/?item=1&glossary_id=1628

Background

Spatz cluster [7] and Scope of analysis



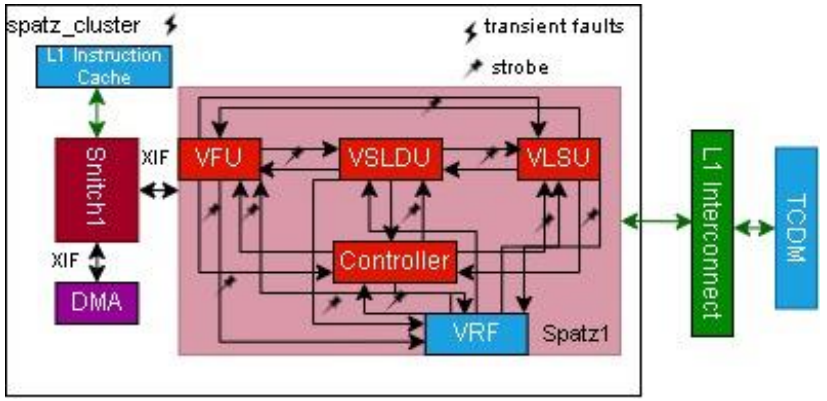


$$R = A \times B + A \times B + A \times B$$

[1] https://ecss.nl/item/?glossary_id=101
[2] http://sciencedirect.com/topics/earth-and-planetary-sciences/single-event-upset
[7] Matteo Perotti et al. 2025. Spatz: Clustering compact RISC-V-based vector units to maximize computing efficiency. IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems (2025).
[6] Stefan Mach et al. 2020. FPnew: An open-source multiformat floating-point unit architecture for energy-proportional transprecision computing. IEEE Transactions on Very Large Scale Integration (VLSI) Systems 29, 4 (2020), 774–787.

Methodology - module-level fault injection campaign

- **FI campaign objective:**
  - Classify how injected transient faults manifest at module boundaries during kernel execution.



1. Select workload

FP32/FP16/BP16/FP8 MatMul or Widening MatMul

→

2. Inject transient fault

SET on combinational ports
SEU on sequential flops

→

3. Compare GM vs FM

VC Z01X concurrent fault simulation

→

4. Classify outcome

Masked, FS, or FD based on strobes

Outcome definitions used by the strobe comparison

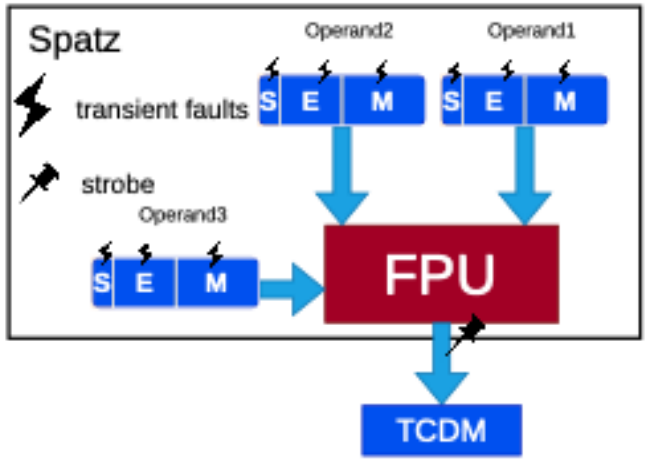
Faulty System Crash (**FS**): handshake/control mismatch that can deadlock execution.
Faulty Data (**FD**): mismatch in functional data values such as operands, intermediate values, or output data.

Scale: 100,000 fault injections across six workload configurations.

Methodology - SDC severity metric

- **To study how Silent Data Corruption (SDC) depends on**
  - The data precision (FP32/FP16/BP16/FP8)
  - The floating-point value composition (sign/exponent/mantissa)
- **Experiment setup**
  - 1,000 targeted fault-injection trials per field: S/E/M.
  - In each trial, flip exactly one bit in the input operands to the FPU
    - at a randomly selected bit-level location within the selected componen
    - and at a randomly selected clock cycle within the kernel execution window
  - if FS -> drop FS;

    else -> to observe FD;



Methodology - SDC severity metric

- **SDC severity metrics**
  - Step1: Compare GM and FM output bit patterns at result write-out. If any output element differs → SDC.
  - Step2: Decode bit patterns

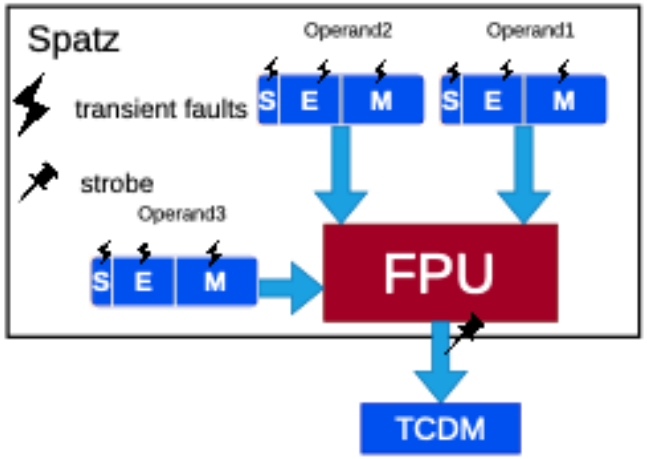
$$y^{GM} = D_p(b^{GM}), \qquad y^{FM} = D_p(b^{FM})$$

  - Step3: Define corrupted outputs

$$\mathcal{I} = \{i \mid b_i^{FM} \neq b_i^{GM}\}, \qquad K = |\mathcal{I}|$$

- Step4: Quantify numerical severity – root mean square error (RMSE)

$$RMSE = \sqrt{\frac{1}{K}\sum_{i\in\mathcal{I}}\left(y_i^{FM} - y_i^{GM}\right)^2}$$



Results: faulty data corruption dominates

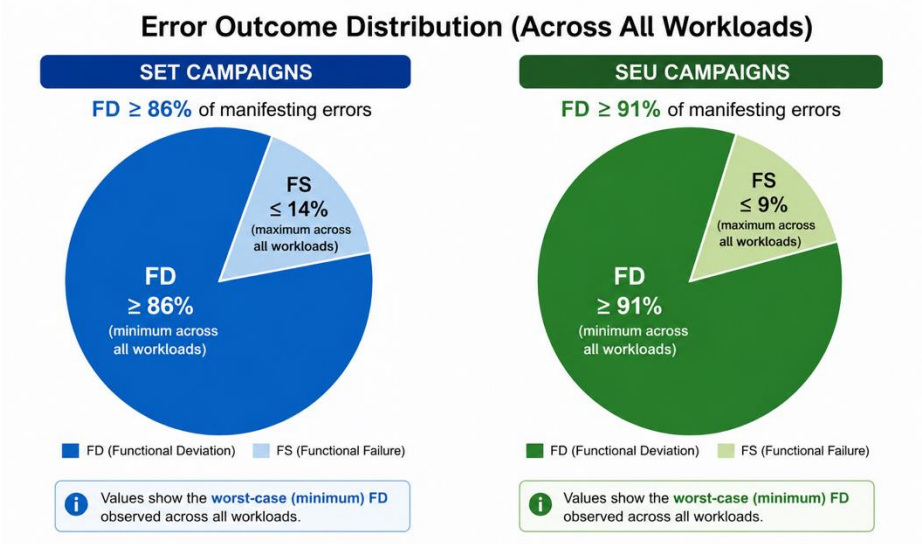
Across workloads, manifesting faults are much more likely to corrupt data than to crash the system.

SET CAMPAIGNS

FD ≥ 86% of manifesting errors

SEU CAMPAIGNS

FD ≥ 91% of manifesting errors



Interpretation

A fault that escapes masking is more likely to propagate through the datapath than to trigger an immediately visible failure.

SDC severity must be measured!

Results: module-level sensitivity

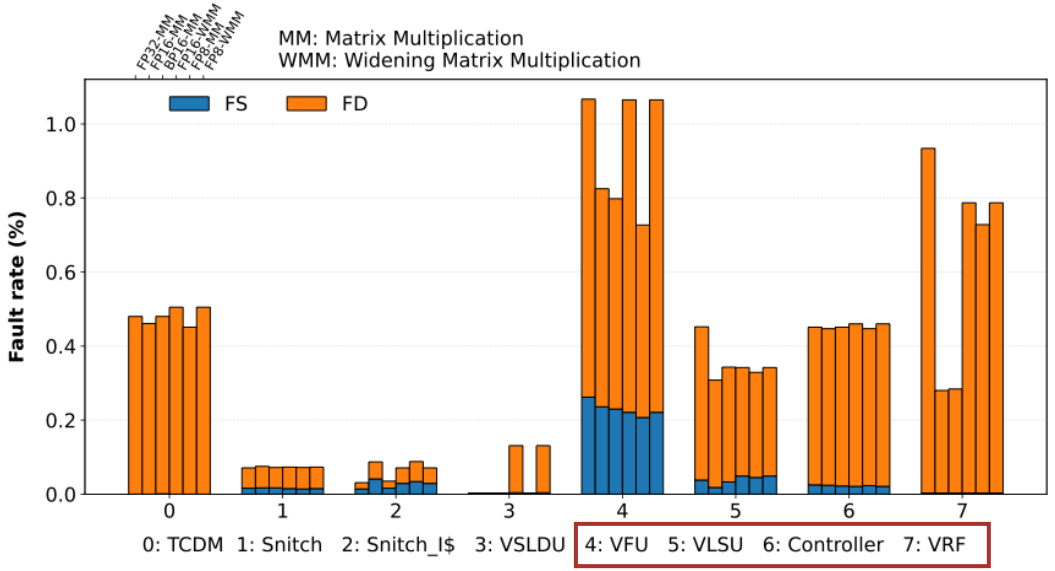


Fig. 3: module sensitivity to SETs

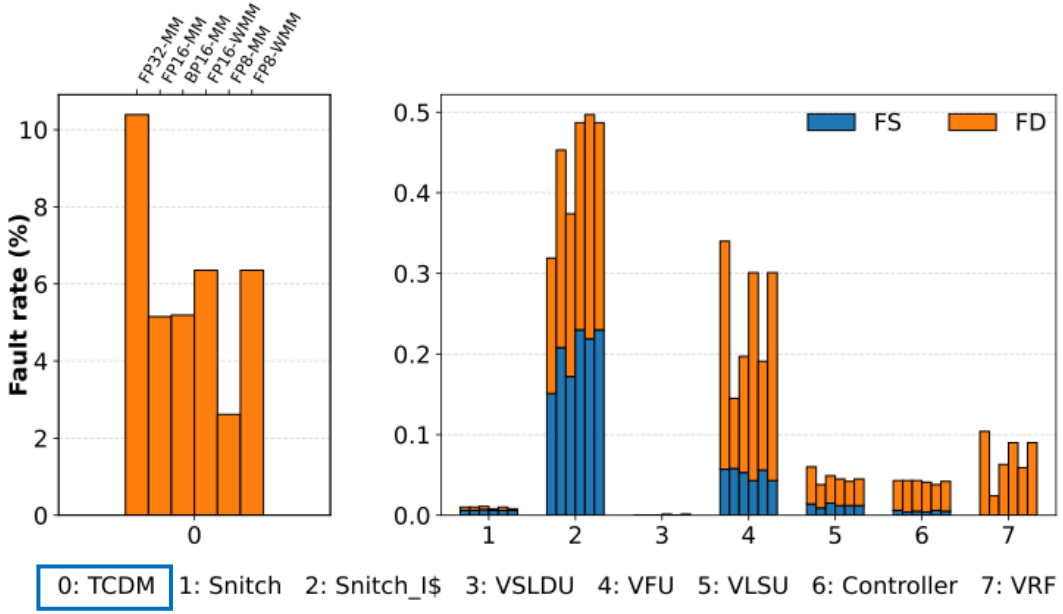


Fig. 4: module sensitivity to SEUs

SET hotspot **Vector execution path:** ≥75%

TCDM: ≥75% **SEU hotspot**

Different protection strategies for SET and SEU

Results: SDC severity by precision and bit-field

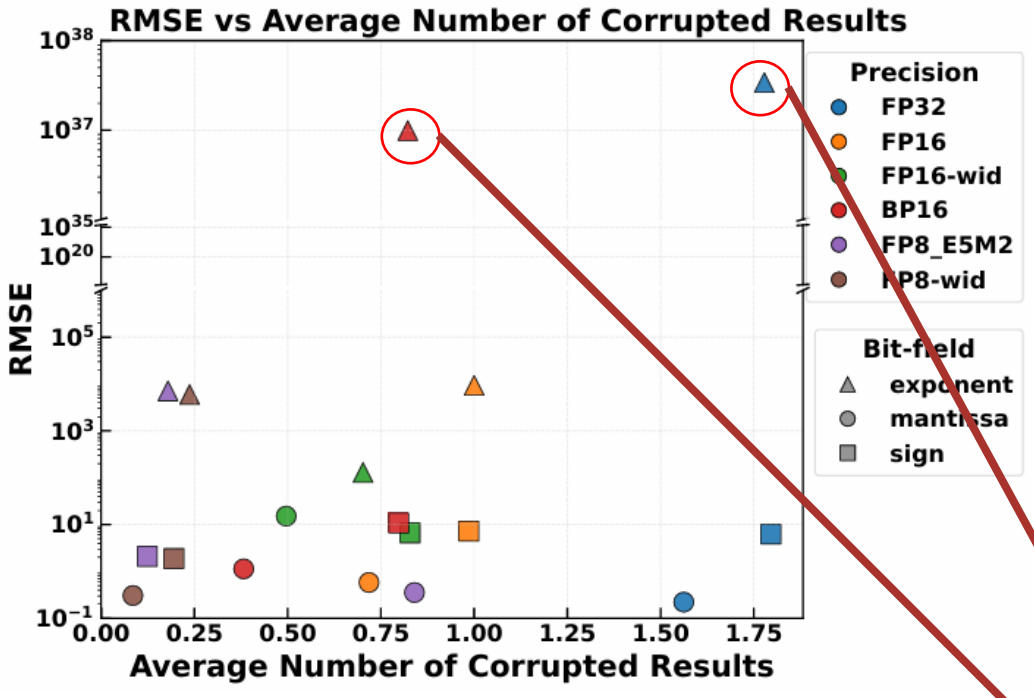


Fig. 5: RMSE versus average number of corrupted outputs for SDC runs

Main observations

1) **FP8** shows the lowest output impact: fewer corrupted outputs and lower RMSE.

2) **FP16** Widening MatMul reduces both corruption spread and RMSE compared with FP16 MatMul.

3) **Widening** is less effective for FP8 because accumulation is 16-bit rather than 32-bit.

4) **Exponent**-targeted faults generate the most severe SDC events, especially FP32 and BP16 outliers.

Protection priority focus checking on exponent-related high-impact cases rather than all bits equally.

- Selectively protecting exponent-targeted faults in FP32 and BP16 is especially attractive

Conclusion

This study supports selective protection, because vulnerability is concentrated in specific error models, modules, and numerical fields.

1	**FD dominates**	Among manifesting errors, FD accounts for ≥86% in SET and ≥91% in SEU campaigns.
2	**SET and SEU differ**	SET hotspots are in vector execution; SEU hotspots are dominated by TCDM.
3	**Severity is format-dependent**	FP8 has the lowest output impact; FP16 widening improves robustness.
4	**Exponent bits matter most**	Exponent corruptions produce the largest RMSE outliers, especially in FP32 and BP16.

Design guideline: **combine low-cost redundancy for handshake/control paths with ECC/parity/scrubbing for storage and targeted checking for high-impact exponent cases.**

Future work

Broader kernels	Extend beyond MatMul/Widening MatMul to additional AI and DSP kernels.
Application-level impact	Connect SDC severity to downstream AI accuracy and mission-level tolerance.
Protection evaluation	Quantify area, timing, and reliability trade-offs for ECC/parity, scrubbing, duplication, and exponent-focused checking.

PULP Platform

Open Source Hardware, the way it should be!

Maoyuan Cai maoyuan.cai@unibo.it

Amirhossein Kiamarzi amirhossein.kiamarz2@unibo.it

Davide Rossi davide.rossi@unibo.it

Angelo Garofalo angelo.garofalo@unibo.it

Institut für Integrierte Systeme – ETH Zürich
Gloriastrasse 35
Zürich, Switzerland

DEI – Universitá di Bologna
Viale del Risorgimento 2
Bologna, Italy

ETH zürich ALMA MATER STUDIORUM UNIVERSITÀ DI BOLOGNA

@pulp_platform

pulp-platform.org

youtube.com/pulp_platform