2025 Design, Automation & Test in Europe Conference (DATE 2025)

Assessing Soft Error Reliability in Vectorized Kernels: Vulnerability and Performance Trade-offs on Arm and RISC-V ISAs

Geancarlo Abich*
* INESC-ID, Portugal – *abich@ieee.org*

***Abstract*—The demand for advanced processing capabilities is paramount in the ever-evolving landscape of radiation-resilient computing exploration. With the standardization of vector extensions on Arm and Risc-V ISAs, leading technology companies are adopting high-performance processors to exploit vector capabilities. This work promotes uniform random fault injection techniques to assess the increased vulnerability within the adoption of vector extensions from RISC-V RVV and Arm SVE. The obtained results show the soft error criticality correlation to registers' cross-section and the vectorized benchmarks, emphasizing the necessity of performance and reliability balance in emerging devices with vector capabilities.**

***Index Terms*—Soft Errors, Reliability, Cross-section, Vectorization.**

I. Introduction

RISC-based processors adopt simplicity in instructions and energy efficiency, making them ideal for flagship products that balance performance and power constraints. Arm and RISC-V-based architectures rapidly advance from general-purpose devices to High-Performance Computers (HPC) [1], [2]. Such advancements support mission-critical applications, including High-Performance Spaceflight Computers (HPSC) and autonomous rovers, leveraging data-level parallelism through packed Single Instruction Multiple Data (SIMD) and Vecto Extensions (VE) (e.g., Arm SVE and RISC-V RVV) [3], [1]. However, vectorization introduces vulnerabilities to soft errors caused by Single Event Effects (SEE).

Soft errors pose significant reliability challenges for HPC platforms, particularly in exposed environments [4]. At early design stages, Fault Injections (FI) frameworks based on Virtual Platforms (VP) simulators ease the analysis of soft error reliability from Instruction Set Architectures (ISA) adopted in HPC platforms. Recent research has intensified on soft error reliability in RISC-based HPC platforms, exploring SIMD and multicore optimizations [5], [6], and recently VEs [7]. This paper addresses these challenges by proposing a scalable fault injection methodology for VEs, assessing register file resilience in Arm Scalable Vector Extension (SVE) and RISC-V RISC-V Vector Extensions (RVV) running General Matrix Multiply (GEMM) for different precisions.

II. Methodology: Fault Injection, Consistency and Assessment

For the experimental process of exposing computing devices to SEEs meticulous planning and allocation of resources has to be performed. For this reason, this work extends Soft error Fault Injection Analysis (SOFIA) [5], an open-source FI solution on top of versatile VP simulators [8]. Such a tool simulates bit upsets (e.g., bit-flips) by injecting faults into specified registers or memory locations while running software stacks. This work proposes the scalable vector fault mask, which extends the *random register file* FI technique to support a uniform distribution of bit-flips across all vulnerable bits considering General-Purpose Registers (GPR)s, Floating-Point Unit (FPU), and VE registers. The proposed extension aims to allow the registers' cross-section reliability evaluation, considering the execution of workloads that combine different precisions and optimizations while exploiting vector extensions. Moreover, the extension is suitable for any processor model with current SVE or RVV of the Arm and RISC-V ISAs.

Evaluating a system's reliability requires crafting an accurate, comprehensive, and realistic procedure. To assure the statistical significance of results from SOFIA this work uses the equations an metrics defined in [9], comprising Architecture Vulnerability Factor (AVF) for Mean Work To Failure (MWTF) and the vulnerability window (*VWindow*) through the register's cross-section (i.e., faults in vulnerable registers') during workload and system execution. For instance, requiring a confidence level of 99% and a 1% error margin, with a sample exceeding 1 million, a minimum of 16587 FIs are necessary to instill results consistency. Such a number is equivalent to $\approx$ 66$\times$ the observed Failure-in-Time (FIT) for 10^9 hours of neutrons exposition at sea level, which means neutron-characterized flip-flop susceptibility FIT_{NYC} of 248 errors, considering a particle flux of $2.5\times10^{10} neutrons/cm2/s$ [10].

III. Experimental Setup and Results Discussion

Ensuring the consistency of results is essential, thus Table I presents the experimental setup considering all the consistency parameters [9]. GEMM is a fundamental linear algebra routine extensively employed in machine learning and scientific simulations [1]. Shifting GEMM from scalar to vector operations demands data and code refactoring to unlock performance. In this work, the workloads comprise 512x512 GEMMs at six different precisions: 8-bit, 16-bit, and 32-bit integers, and 16-bit, 32-bit, and 64-bit floats (i.e., IEEE 754 standards) using $LMUL > 1$ for vector and unroll optimizations[1].

Table II highlights reliability profiling and register cross-section per byte for various precisions, optimizations, and ISAs. Mixed configurations, commonly used by developers, significantly reduce the vulnerability window through VE, as seen in 8-bit precision, where 512-bit vectors process 64 operands per register. However, this increases the registers'

[1]Loop unrolling reduces loop overhead, i.e., branching and counter updates.

978-3-9826741-0-0/DATE25/© 2025 EDAA

TABLE I: FI Experimental Setup

Vendor Processor IPs	SiFive X280(RV64GCV) Arm Cortex-A75(Armv8.2-A+SVE)
Benchmarks	6 GEMM: 8-bit int, 16-bit int, 32-bit int, 16-bit float, 32-bit float, 64-bit float

OS, Compiler, and opt. flags	Bare metal, LLVM clang 17.0.4 with
	`-O3 -mARCH-vector-bits=512`
	`#pragma loop vectorize`
	`#pragma loop unroll`

Target FI	Register's Cross-section (GP, FP, and Vector Registers)
# FI campaigns	36
Injections per campaign	17k
Total FIs	1.224 Million

TABLE II: Soft error reliability profiling.

		# Ex. Inst. ($\times 10^6$)		V.Window (GB)		R.Cross-sec. ($\times 10^{-5}$)	
Inst.	Precision	RISC-V	Arm	RISC-V	Arm	RISC-V	Arm
scalar	8-bit int	609.5	573.1	1453.13	1400.62	10.73	5.08
mixed	8-bit int	13.9	82.4	33.13	201.36	22.25	86.72
vector	8-bit int	7.9	6.4	18.77	15.60	36.75	54.93
scalar	16-bit int	646.5	573.2	1541.26	1400.82	12.61	5.27
mixed	16-bit int	24.1	80.4	57.51	196.44	23.01	69.44
vector	16-bit int	13.6	10.7	32.53	26.06	59.14	99.72
scalar	32-bit int	651.7	506.3	1553.76	1237.22	14.27	5.36
mixed	32-bit int	34.2	96.3	81.54	235.28	29.56	48.67
vector	32-bit int	27.0	22.4	64.43	54.69	91.00	156.85
scalar	16-bit FP	506.8	573.2	1208.27	1400.82	13.12	5.11
mixed	16-bit FP	18.9	80.4	45.15	196.44	21.19	61.27
vector	16-bit FP	13.4	10.7	31.90	26.06	54.12	52.39
scalar	32-bit FP	517.0	506.3	1232.51	1237.22	13.58	5.49
mixed	32-bit FP	33.9	96.3	80.91	235.28	17.64	41.11
vector	32-bit FP	18.5	31.6	44.09	77.16	112.78	93.22
scalar	64-bit FP	566.5	506.6	1350.56	1238.02	18.49	6.18
mixed	64-bit FP	70.5	128.3	168.06	313.59	21.59	33.86
vector	64-bit FP	60.6	69.7	144.53	170.38	112.73	75.61

cross-section and system vulnerability. Vector configurations generally improve performance, except for RISC-V in double precision, where reduced operands and unroll factors require additional scalar instructions, increasing masking rates but slightly lowering the registers' cross-section.

For Arm SVE, the mixed approach shows smaller performance gains than RISC-V, as reflected in Table II. Double FP vector performance also decreases (84% compared to mixed). Both ISAs reveal an inverse relationship between register cross-section and the vulnerability window, especially at higher precisions where more registers are needed for loop unrolling.

Figure 1 shows the results from the FI campaigns considering the fault classification along with the MWTF for both ISAs. The results highlight a notable increase in detected faults (tolerable, critical, and crash) across all workloads due to increased vulnerable bits with ISAs featuring VEs. While up to 95% of faults are tolerable, scalar versions still present critical errors, especially in double precision FP operations (up to 5.9%). The precision notably influences fault occurrence. Integer-based vector approaches presented an occurrence of 78% critical faults in higher precisions, attributed to the constraint imposed by the vector length (512 bits) and precision (32 bits). This limitation increases the probability of faults between operations and their propagation to outputs. However, the vectorization positively impacts performance, increasing MWTF and enhancing overall reliability even with higher criticality. In RISC-V ISA, the MWTF improvements from vector approaches reduce from $25.4\times$ in 8-bit int to $5.8\times$ in 32-bit int, indicating the impact of the precision in the soft error reliability. When comparing ISAs, Armv8.2-A+SVE presented more crashes in mixed approaches characterized by predicate states. The compiler's use of double precision to store lower precision data results in faults outside the specified range becoming exceptions (i.e., crashes), a behavior similarly observed and tolerated in RISC-V RVV. However, this effect is mitigated in vector versions within the Arm ISA.

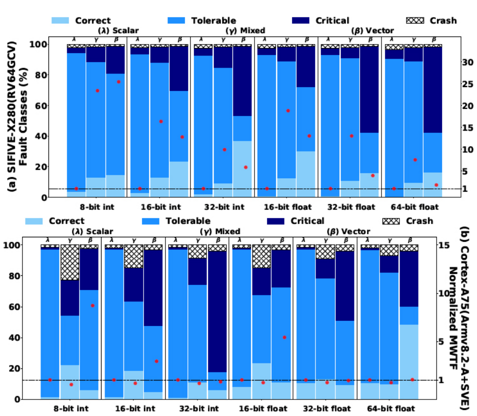

Fig. 1: Fault classification results.

IV. Conclusions

Vector and mixed configurations enhance performance by reducing execution time but increase register cross-section and critical fault rate (up to $37\times$ with higher precision), undermining MWTF gains and worsening workload vulnerability. Key observations include: *(i)* Vector GEMM exhibits frequent critical faults, mainly at higher precisions; *(ii)* As precision increases, the average critical fault rate rises significantly, negatively impacting MWTF benefits from vectorization; *(iii)* RISC-V RVV generally shows lower vulnerability than Arm SVE, particularly under mixed configurations, which exacerbate fault occurrences due to increased register usage. The results indicate that while VEs improve performance and reduce the vulnerability window, they increase fault occurrences due to larger chip area exposure. Additionally, the flexibility of scalable vector extensions can reduce performance when combining registers (e.g., LMUL) to extend vector length [1].

References

[1] N. Adit and A. Sampson, "Performance left on the table: An evaluation of compiler autovectorization for RISC-V," *IEEE Micro*, vol. 42, no. 5, pp. 41–48, 2022.

[2] E. J. Marinissen *et al.*, "Silent data corruption: Test or reliability problem?" in *2024 IEEE ETS*, 2024, pp. 1–7.

[3] N. Stephens, S. Biles, M. Boettcher *et al.*, "The arm scalable vector extension," *IEEE Micro*, vol. 37, no. 2, pp. 26–39, 2017.

[4] ISO, "Road vehicles – Functional safety," 2011. [Online]. Available: https://www.iso.org/standard/68383.html

[5] J. Gava, V. Bandeira, F. Rosa, R. Garibotti, R. Reis, and L. Ost, "SOFIA: An automated framework for early soft error assessment, identification, and mitigation," *Journal of Systems Architecture*, vol. 131, 2022.

[6] G. Abich, J. Gava, R. Garibotti, R. Reis, and L. Ost, "Applying Lightweight Soft Error Mitigation Techniques to Embedded Mixed Precision Deep Neural Networks," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 68, pp. 4772–4782, 2021.

[7] C. Imianosky, D. A. Santos, D. R. Melo, F. VieE, and L. Dilillo, "Implementation and reliability evaluation of a RISC-V vector extension unit," in *IEEE DFT Symposium*, 2023, pp. 1–6.

[8] Synopsys, "ImperasDV - Virtual Platform Development and Simulation," 2025. [Online]. Available: https://www.synopsys.com/

[9] G. Abich *et al.*, "Evaluation of the soft error assessment consistency of a JIT-based virtual platform simulator," *IET Computers & Digital Techniques*, vol. 15, no. 2, pp. 125–142, 2021.

[10] JEDEC Open Standards, "Measurement and reporting of alpha particle and terrestrial cosmic ray-induced soft errors in semiconductor devices," JEDEC Solid State Technology Association, Tech. Rep. JESD89B, 2021.