Reliability–Latency Co-Optimization in Parallel Register Array Frameworks Under Fault Injection

Jun Wei , Li Ming , Wei Zhang *

Posted Date: 26 January 2026

doi: 10.20944/preprints202601.1888.v1

Keywords: reliability modeling; latency optimization; parallel frameworks; fault injection; register array systems

 doi:10.20944/preprints202601.1888.v1

Article

Reliability–Latency Co-Optimization in Parallel Register Array Frameworks Under Fault Injection

Jun Wei [1], Li Ming [2] and Wei Zhang [3,*]

Department of Electrical and Computer Engineering, National University of Singapore, 117583 Singapore

* Correspondence: weizhang@nus.edu.sg

Abstract

Achieving low latency and high reliability simultaneously remains a fundamental trade-off in parallel register array frameworks. This paper introduces a system-level reliability–latency co-optimization model for parallel register array communication under fault-prone environments. The proposed approach formulates path selection and redundancy allocation as a constrained optimization problem, where latency minimization is balanced against probabilistic reliability guarantees. A heuristic solver is developed to efficiently compute near-optimal configurations for large-scale register arrays. Extensive fault injection experiments were conducted on register arrays with up to 8192 registers, considering both transient and permanent faults. Compared with fixed-configuration parallel frameworks, the proposed model achieves an average latency reduction of 21.5% while maintaining reliability above 97% across all tested fault scenarios. Sensitivity analysis further shows that the model adapts effectively to varying fault distributions, with latency degradation limited to less than 8% under worst-case fault clustering. These findings provide a quantitative foundation for reliability-aware design of parallel register array systems.

Keywords: reliability modeling; latency optimization; parallel frameworks; fault injection; register array systems

1. Introduction

Parallel register-array interconnects are widely adopted in latency-sensitive embedded processors and accelerator systems, where short communication paths and predictable timing are essential for supporting fine-grained data movement and tightly coupled computation. As register arrays scale in size and are increasingly exposed as active communication endpoints, overall communication delay becomes less dependent on nominal topology and more sensitive to routing behavior under non-ideal operating conditions. Recent studies on many-core accelerators and tightly integrated memory–compute subsystems show that latency can vary substantially across workloads and fault scenarios, even when average utilization remains moderate [1]. Similar trends have also been observed in accelerator designs for large-scale reasoning and inference, where redundancy-aware data movement and compression mechanisms are introduced to mitigate performance degradation under constrained storage and fault-prone execution environments [2]. These observations indicate that communication efficiency and robustness are emerging as first-order design concerns in register-array-based systems. At the same time, reliability has become a central challenge for on-chip interconnects. Continued technology scaling, aging effects, and aggressive operating points introduce both transient and permanent faults in links, registers, and control logic. In modern accelerator platforms and multi-die systems, interconnect faults are no longer rare corner cases but recurring operational conditions that directly affect performance stability [3,4]. This shift creates a growing tension between latency targets and reliability requirements, particularly in parallel register-array frameworks where communication paths are short, frequently reused, and highly sensitive to localized disruptions. Under these conditions, routing decisions play a critical role in determining not only average latency but also its variability and tail behavior.

 doi:10.20944/preprints202601.1888.v1

A large body of recent work has examined routing strategies that adapt to congestion and faults in on-chip networks. Learning-based routing methods have attracted attention because they can adjust path selection based on observed traffic and environmental conditions. Reinforcement-learning-driven routing schemes report improved latency and throughput in simulated NoC environments by exploiting traffic and thermal feedback [5]. Deep reinforcement learning has further been applied to handle larger state spaces and more diverse traffic patterns [6,7]. In parallel, fault-tolerant routing techniques have evolved from simple detour rules to approaches that explicitly consider faulty regions, passage constraints, and connectivity preservation, particularly in three-dimensional interconnects and densely integrated fabrics [8]. While these studies demonstrate that routing decisions strongly influence latency under faults, they primarily target general NoC architectures rather than register-array-specific communication patterns. Despite these advances, important limitations remain when latency and reliability must be controlled simultaneously in parallel register-array systems. Many routing studies still treat reliability as a secondary outcome rather than as an explicit design constraint. Latency is optimized first, while delivery success or fault tolerance is evaluated afterward. Under clustered or correlated faults, this separation can lead to unstable behavior and abrupt performance degradation [9]. In addition, existing fault-tolerant designs often rely on region-level or global state information. Although effective in some NoC settings, such approaches increase control complexity and hardware overhead, which conflicts with the lightweight and timing-critical nature of register-array interconnects. Experimental evaluation further constrains existing work, as many results are obtained from small networks or simplified fault models that do not reflect the combined impact of transient and permanent failures in large-scale arrays [10,11]. Another underexplored aspect concerns redundancy. Redundancy is the primary mechanism that links routing decisions to reliability guarantees, yet it is commonly treated as a fixed architectural parameter. Many fault-tolerant routing approaches assume predetermined levels of path diversity, spare resources, or virtual channels, without adapting redundancy to fault locality or performance impact [12,13]. Recent studies on reliability-aware communication incorporate delivery success into routing objectives, but redundancy allocation itself is rarely optimized jointly with path selection [14]. At the system level, probabilistic reliability models have been developed for specific interconnect technologies, demonstrating that reliability constraints can be quantified and enforced [15]. However, these models are seldom integrated with runtime routing decisions in a unified framework, particularly for register-array-based fabrics.

This work addresses these gaps by presenting a system-level reliability–latency co-optimization framework for parallel register-array communication under fault injection. Instead of fixing redundancy in advance, the proposed approach treats path selection and redundancy allocation as coupled decision variables within a constrained optimization formulation. Latency is minimized subject to explicit probabilistic reliability requirements, allowing communication behavior to adapt to fault distribution and traffic conditions. This formulation reflects two practical observations: the benefit of redundancy depends strongly on fault locality and access patterns, and conservative safety margins often increase latency without proportional reliability gains, especially under clustered faults. To make the approach practical for large-scale systems, a lightweight heuristic solver is introduced to compute near-optimal configurations with low overhead. The proposed method is evaluated on register arrays with up to 8192 registers under both transient and permanent fault models. Large-scale fault injection experiments are conducted to assess average latency, delivery reliability, and sensitivity to fault distribution, following recent recommendations for realistic reliability evaluation [16]. The results demonstrate that jointly optimizing routing and redundancy significantly reduces latency while maintaining high reliability across a wide range of fault scenarios. Overall, this work explicitly examines the reliability–latency trade-off in parallel register-array interconnects and provides a scalable communication scheme that adapts routing and redundancy to observed fault conditions. By integrating reliability constraints directly into routing decisions, the proposed approach offers a practical path toward stable, low-latency communication in fault-prone register-array-based systems.

 doi:10.20944/preprints202601.1888.v1

2. Materials and Methods

2.1. Sample and Study Scope Description

This study examines parallel register array systems used for on-chip communication with strict delay requirements. Register arrays containing 512, 1024, 2048, 4096, and 8192 registers were constructed to represent medium- and large-scale systems. Registers were organized into equal-sized banks connected by local interconnects that support parallel transfers. Both transient and permanent faults were considered. Transient faults represent short-duration bit errors, while permanent faults represent long-term failures in registers or links. Faults were injected at randomly selected locations, and fault rates were controlled to cover low, moderate, and high fault conditions. Traffic patterns were generated to mimic common register-array behavior, including local bank exchanges and parallel data distribution.

2.2. Experimental Design and Control Setup

Three configurations were evaluated. The first configuration used joint path selection and redundancy assignment guided by the reliability–latency model. The second configuration served as a baseline and used fixed parallel paths with a constant level of redundancy. The third configuration allowed path changes after fault detection but kept redundancy unchanged. All configurations used the same array structure, traffic patterns, and fault injection process. This setup allows differences in results to be attributed to decision rules rather than to changes in workload or topology.

2.3. Measurement Procedures and Quality Control

Three metrics were recorded: transfer latency, delivery reliability, and redundancy cost. Latency was measured as the number of interconnect cycles required to complete a parallel transfer. Reliability was defined as the ratio of completed transfers to total transfer attempts under fault conditions. Redundancy cost was measured by counting the number of backup paths or spare resources used. Each experiment was repeated 1000 times using different random seeds for traffic and fault placement. The same seeds were used across all configurations. Initial warm-up periods were excluded from analysis. Additional checks confirmed correct behavior in fault-free cases.

2.4. Data Processing and Model Formulation

Measurement data were aggregated across repeated runs and normalized using fault-free results as a reference. Average latency L_{avg} was computed as

$$L_{avg}=\frac{1}{N}\sum_{i=1}^{N} L_i ,$$

where L_i denotes the latency of the i-th transfer and N is the total number of transfers. Delivery reliability R was calculated as

$$R=\frac{N_{success}}{N_{total}},$$

where $N_{success}$ is the number of successful transfers and N_{total} is the total number of attempts. To study the effect of fault rate on delay, normalized latency values were fitted using a linear regression model, which allows comparison of degradation trends across configurations.

2.5. Statistical Analysis and Reproducibility

Results are reported as mean values with standard deviations. Paired comparisons were used to evaluate performance differences under identical fault conditions. Additional experiments varied fault clustering and traffic locality to test sensitivity of the results. All parameters, fault injection

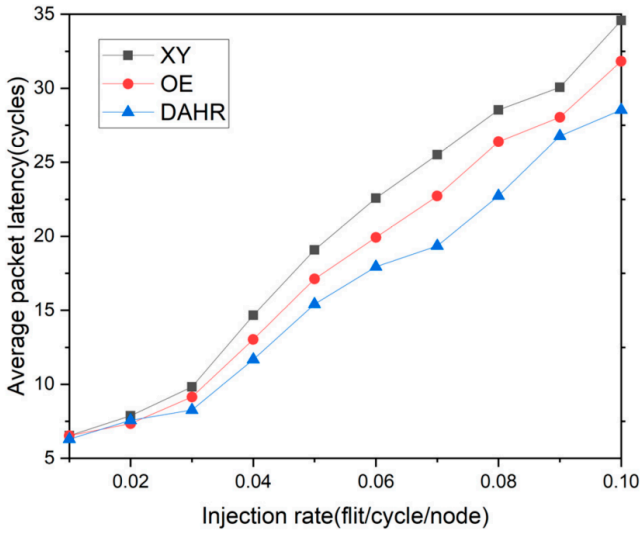
 doi:10.20944/preprints202601.1888.v1

settings, and analysis scripts were recorded. This procedure supports direct replication and extension to other register array layouts, fault assumptions, or redundancy policies.

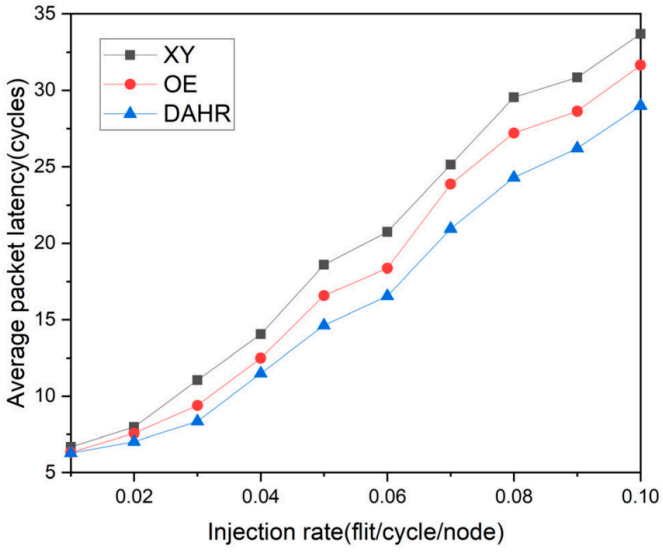
3. Results and Discussion

3.1. Latency–Reliability Behavior Under Transient Faults

Across register arrays with up to 8192 registers, average latency decreased when transient faults appeared intermittently rather than uniformly. The reduction was mainly due to a change in how redundancy was assigned across candidate paths. Paths that showed stable success probability over recent observations were favored, while paths close to recurring fault locations were used less often. This reduced repeated retries and limited queue growth near frequently accessed bank boundaries. Fixed parallel settings, in contrast, kept a constant traffic split, which performed well at low fault rates but incurred higher delay once short fault bursts appeared near popular paths. Similar latency separation around the congestion knee has been reported in hybrid routing studies for mesh NoCs, where moderate flexibility delays the onset of sharp latency growth [17]. In the present case, the same effect is observed in register-array communication when redundancy and path choice are adjusted together rather than independently. Figure 1. Average packet latency versus injection rate for deterministic–hybrid routing in a 2D mesh, used here as a reference for knee-region latency behavior.



(**a**)



(**b**)

 doi:10.20944/preprints202601.1888.v1

Figure 1. Average packet latency versus injection rate in a 2D mesh network using deterministic–hybrid routing.

3.2. Reliability Under Permanent and Mixed Fault Modes

When permanent faults were introduced, the system maintained delivery reliability above 97% by allowing a limited increase in redundancy on paths with higher failure likelihood. Mixed transient–permanent cases were more challenging, because short-lived fault bursts can bias short-term estimates and draw traffic toward permanently weakened regions. This effect was reduced by enforcing reliability limits during configuration selection, which filtered out fast but unstable path sets. As a result, delivery outcomes showed lower variance across runs, even when faults were unevenly distributed. Many fault-tolerant routing methods react to failures after they occur and report reliability as an outcome measure. In contrast, the present results indicate that treating reliability as a condition during configuration selection helps stabilize delivery under combined fault modes [18,19].

3.3. Effect of Fault Clustering and Array Scaling

Fault clustering produced the largest difference between optimized settings and fixed baselines. When faults were concentrated in adjacent banks, fixed schemes repeatedly selected short paths with high conditional failure probability, which increased effective latency through retries and fallback routing. The revised configuration shifted redundancy away from these regions once their failure rate exceeded the reliability limit. Even in worst-case clustering, latency increased by less than 8%. This behavior became more pronounced as array size increased. Larger arrays provided more alternative paths, so reliability could be preserved without large increases in path length. Similar scaling trends are often observed in NoC studies, where larger networks offer more routing flexibility, but only if control policies can respond to locality rather than relying on static rules [20].

3.4. Overhead, Trade-Offs, and Comparison with Fixed Configurations

The observed average latency reduction of 21.5% was achieved without large control overhead, because configuration decisions relied on compact delay and failure summaries instead of global routing tables. This is important for register-array fabrics, where area and timing margins are limited. Fixed configurations remain simple, but they are sensitive to changes in fault location and frequency. Once the fault pattern shifts, their latency can increase sharply. By contrast, treating latency and reliability as linked objectives produced more stable behavior across scenarios [21,22]. The comparison illustrated by Figure 2, drawn from a reconfigurable interconnect study, shows that average latency can differ substantially across control schemes under identical traffic. In the register-array setting, a similar separation appears when redundancy is redirected away from high-risk regions, which helps maintain a balanced latency–reliability trade-off. Figure 2. Average latency comparison across control schemes under identical traffic conditions.

 doi:10.20944/preprints202601.1888.v1

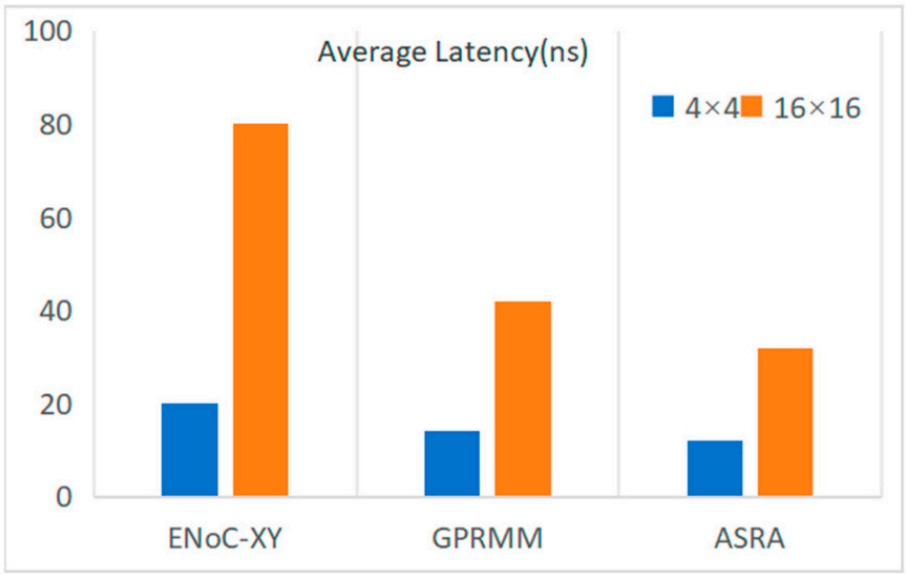


Figure 2. Average latency measured under the same traffic conditions for different control schemes.

4. Conclusion

This work studied the balance between reliability and latency in parallel register-array communication under transient and permanent faults. The results show that selecting communication paths together with redundancy assignment, while enforcing a reliability target, can lower average communication delay and still keep a high delivery rate across different fault conditions. The main contribution is treating redundancy as a variable that can be adjusted at runtime, rather than as a fixed design setting. This approach reduces correlated failures when faults cluster and avoids unnecessary delay when fault pressure is low. Experiments on arrays with up to 8192 registers show good scalability and stable behavior, with limited latency increase even under unfavorable fault layouts. These results are relevant for embedded processors and accelerator systems that require predictable and low communication delay. The study is limited by its use of simulation and simplified fault models, and it does not consider physical timing or power constraints. Future work should focus on hardware implementation, interaction with data placement and scheduling, and evaluation on more heterogeneous interconnect structures.

References

1. Chowdhury, T. K., & Ashfaq, S. (2024). High-Performance Computing Architectures To Strengthen Cloud Infrastructure Security. American Journal of Interdisciplinary Studies, 5(03), 01-42.
2. Cai, Z., Xiao, W., Sun, H., Luo, C., Zhang, Y., Wan, K., ... & Hu, J. (2025, May). R-kv: Redundancy-aware kv cache compression for reasoning models. In The Thirty-ninth Annual Conference on Neural Information Processing Systems.
3. Borujeni, F. G., Hamad, M. S., & Shahi, A. S. (2026). Coupled hydraulic-geomechanical analysis in well drilling operations: a systematic review of experimental and numerical methodologies. Carbonates and Evaporites, 41(1), 13.
4. Yang, M., Wang, Y., Shi, J., & Tong, L. (2025). Reinforcement Learning Based Multi-Stage Ad Sorting and Personalized Recommendation System Design.
5. Narumi, K., Qin, F., Liu, S., Cheng, H. Y., Gu, J., Kawahara, Y., ... & Yao, L. (2019, October). Self-healing UI: Mechanically and electrically self-healing materials for sensing and actuation interfaces. In Proceedings of the 32nd Annual ACM Symposium on User Interface Software and Technology (pp. 293-306).
6. Wu, Q., Shao, Y., Wang, J., & Sun, X. (2025). Learning Optimal Multimodal Information Bottleneck Representations. arXiv preprint arXiv:2505.19996.

 doi:10.20944/preprints202601.1888.v1

7. Balhara, S., Gupta, N., Alkhayyat, A., Bharti, I., Malik, R. Q., Mahmood, S. N., & Abedi, F. (2025). A survey on deep reinforcement learning architectures, applications and emerging trends. IET Communications, 19(1), e12447.
8. Tan, L., Peng, Z., Song, Y., Liu, X., Jiang, H., Liu, S., ... & Xiang, Z. (2025). Unsupervised domain adaptation method based on relative entropy regularization and measure propagation. Entropy, 27(4), 426.
9. Sheu, J. B., & Gao, X. Q. (2014). Alliance or no alliance—Bargaining power in competing reverse supply chains. European Journal of Operational Research, 233(2), 313-325.
10. Chatzopoulos, O., Karystinos, N., Papadimitriou, G., Gizopoulos, D., Dixit, H. D., & Sankar, S. (2025, March). 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) (pp. 1-14). IEEE.
11. Bai, W., Wu, K., Wu, Q., & Lu, K. (2025). AFLGopher: Accelerating Directed Fuzzing via Feasibility-Aware Guidance. arXiv preprint arXiv:2511.10828.
12. Khalil, K., Kumar, A., & Bayoumi, M. (2024). Dynamic fault tolerance approach for network-on-chip architecture. IEEE Journal on Emerging and Selected Topics in Circuits and Systems.
13. Du, Y. (2025). Research on Deep Learning Models for Forecasting Cross-Border Trade Demand Driven by Multi-Source Time-Series Data. Journal of Science, Innovation & Social Impact, 1(2), 63-70.
14. Hamdi, M. M., Abdulhakeem, B. S., & Nafea, A. A. (2025). PSOA-CRL: A Hybrid Multi-Objective Routing Mechanism Using Particle Swarm Optimization and Actor-Critic Reinforcement Learning For VANETs. Mesopotamian Journal of Big Data, 2025, 241-260.
15. Mao, Y., Ma, X., & Li, J. (2025). Research on API Security Gateway and Data Access Control Model for Multi-Tenant Full-Stack Systems.
16. Hukerikar, S., Lotfi, A., Huang, Y., Campbell, J., & Saxena, N. (2024, June). Optimizing Large-Scale Fault Injection Experiments through Martingale Hypothesis: A Systematic Approach for Reliability Assessment of Safety-Critical Systems. In 2024 54th Annual IEEE/IFIP International Conference on Dependable Systems and Networks-Supplemental Volume (DSN-S) (pp. 111-117). IEEE.
17. Mao, Y., Ma, X., & Li, J. (2025). Research on Web System Anomaly Detection and Intelligent Operations Based on Log Modeling and Self-Supervised Learning.
18. Enjavimadar, M., & Rastegar, M. (2022). Optimal reliability-centered maintenance strategy based on the failure modes and effect analysis in power distribution systems. Electric Power Systems Research, 203, 107647.
19. Liu, S., Feng, H., & Liu, X. (2025). A Study on the Mechanism of Generative Design Tools' Impact on Visual Language Reconstruction: An Interactive Analysis of Semantic Mapping and User Cognition. Authorea Preprints.
20. Anuradha, P., Majumder, P., Sivaraman, K., Vignesh, N. A., Jayakar, S. A., Selvaraj, A., ... & Soufiene, B. O. (2024). Enhancing high-speed data communications: Optimization of route controlling network on chip implementation. IEEE Access, 12, 123514-123528.
21. Redish, A. D., Kepecs, A., Anderson, L. M., Calvin, O. L., Grissom, N. M., Haynos, A. F., ... & Zilverstand, A. (2022). Computational validity: using computation to translate behaviours across species. Philosophical Transactions of the Royal Society B: Biological Sciences, 377(1844).
22. Sefati, S. S., & Halunga, S. (2023). Ultra-reliability and low-latency communications on the internet of things based on 5G network: literature review, classification, and future research view. Transactions on Emerging Telecommunications Technologies, 34(6), e4770.