Error-Resilient Quantum Compiler Design for Efficient Qubit Mapping, Gate Optimization, and Noise Mitigation in NISQ-Era Devices

Beyza Altan
Bilim Kampüs Anadolu Lisesi

***Abstract*—The rapid evolution of Noisy Intermediate-Scale Quantum (NISQ) devices has intensified the need for robust compilation strategies capable of mitigating hardware-induced errors while maximizing computational efficiency. Quantum compilers play a pivotal role in translating high-level quantum algorithms into hardware-executable instructions, yet they face significant challenges related to qubit connectivity constraints, limited coherence times, and stochastic noise. This paper examines error-resilient quantum compiler design for efficient qubit mapping, gate optimization, and noise mitigation in NISQ-era devices, proposing a framework that integrates topology-aware mapping, adaptive gate synthesis, and probabilistic error suppression techniques. By synthesizing advances in quantum circuit optimization, hardware-aware scheduling, and hybrid classical–quantum error correction strategies, the study evaluates how compiler-level innovations can substantially enhance execution fidelity and resource efficiency. The findings suggest that intelligent compiler architectures are critical enablers of practical quantum advantage during the transitional NISQ phase, bridging theoretical algorithm design with real-world hardware constraints.**

Quantum computing has progressed from theoretical abstraction to experimental realization, with contemporary quantum processors entering what is commonly described as the Noisy Intermediate-Scale Quantum (NISQ) era [5]. These devices, characterized by tens to hundreds of qubits and limited error-correction capabilities, offer unprecedented opportunities for computational exploration while simultaneously presenting formidable engineering and algorithmic challenges. Among the most pressing obstacles are qubit decoherence, gate infidelity, and restricted qubit connectivity, all of which constrain the reliability and scalability of quantum computations [8]. In this context, the design of efficient quantum compilers has emerged as a critical factor in translating algorithmic potential into executable performance.

A quantum compiler functions as the intermediary between abstract quantum algorithms and physical

Digital Object Identifier 10.62802/w6gg1c44
Date of publication 20 02 2026; date of current version 20 02 2026

hardware implementations. Unlike classical compilers, which primarily optimize for execution speed and memory efficiency, quantum compilers must account for probabilistic error propagation, hardware topology limitations, and the temporal fragility of quantum states [7]. These additional constraints necessitate a multidimensional optimization process in which qubit mapping, gate sequencing, and noise mitigation strategies are co-optimized rather than treated as independent tasks.

Qubit mapping represents one of the most significant challenges in quantum compilation. Because qubits in physical devices are not universally connected, logical qubits defined in an algorithm must be strategically assigned to physical qubits in a manner that minimizes the need for costly swap operations [10]. Inefficient mapping can dramatically increase circuit depth, thereby amplifying decoherence effects and reducing overall fidelity. Topology-aware mapping algorithms aim to address this issue by aligning logical qubit interactions with the hardware's native connectivity graph, effectively compressing circuit depth while preserving computational intent [2].

Gate optimization constitutes a complementary dimension of compiler design. Quantum gates, particularly multi-qubit operations, are susceptible to higher error rates and longer execution times [6]. Compiler-level gate synthesis and decomposition techniques seek to reduce gate counts and exploit hardware-native gate sets, thereby minimizing cumulative noise. Advanced approaches incorporate adaptive optimization loops in which classical algorithms iteratively refine quantum circuit structures based on probabilistic error models and performance metrics [4].

Noise mitigation further distinguishes quantum compilation from its classical counterpart. In NISQ devices, full quantum error correction remains impractical due to resource limitations, prompting the development of compiler-assisted error suppression techniques. These include probabilistic error cancellation, zero-noise extrapolation, and dynamic circuit reordering, which collectively aim to enhance execution fidelity without incurring prohibitive overhead [1]. By embedding these strategies directly into the compilation pipeline, compilers evolve from passive translators into active agents of computational resilience.

The convergence of qubit mapping, gate optimization, and noise mitigation underscores the necessity of holistic compiler architectures that integrate hardware awareness with algorithmic adaptability [3]. As quantum hardware continues to evolve, compiler design must likewise adapt to heterogeneous qubit technologies, variable coherence profiles, and emerging quantum instruction sets. Interdisciplinary collaboration among computer scientists, physicists, and electrical engineers is therefore essential to ensure that compiler innovations remain aligned with both theoretical advances and practical device capabilities [9].

This paper investigates error-resilient quantum compiler design as a foundational component of efficient quantum computation in the NISQ era. By examining contemporary strategies for qubit mapping, gate optimization, and noise mitigation, the study aims to clarify how compiler-level intelligence can bridge the gap between abstract quantum algorithms and imperfect hardware realities. Ultimately, the evolution of quantum compilers is positioned not merely as a technical refinement but as a strategic enabler of scalable and reliable quantum computing in the near term.

REFERENCES

1. Chinnarasu, R., Poole, C., Phuttitarn, L., Noori, A., Graham, T. M., Coppersmith, S. N., ... & Saffman, M. (2025). Variational simulation of the Lipkin-Meshkov-Glick model on a neutral atom quantum computer. PRX Quantum, 6(2), 020350.
2. Du, Z., Kan, S., Stein, S., Liang, Z., Li, A., & Mao, Y. (2025). Hardware-aware Compilation for Chip-to-Chip Coupler-Connected Modular Quantum Systems. arXiv preprint arXiv:2505.09036.
3. Khan, J. (2025). Quantum Software Engineering: Developing Algorithms for IBM Quantum Systems. Available at SSRN 5708423.
4. Karuppasamy, K., Puram, V., Johnson, S., & Thomas, J. P. (2025). A comprehensive review of quantum circuit optimization: Current trends and future directions. Quantum Reports, 7(1), 2.
5. Keçeci, M. (2025). Accuracy, Noise, and Scalability in Quantum Computation: Strategies for the NISQ Era and Beyond.

6. Nemirovsky, J., Chuchem, M., & Shapira, Y. (2025). Efficient compilation of quantum circuits using multi-qubit gates. Quantum Science and Technology.
7. Njoku, T. K. (2025). Quantum software engineering: algorithm design, error mitigation, and compiler optimization for faulttolerant quantum computing. Int J Comput Appl Technol Res, 14(4), 30-42.
8. Oukaira, A. (2025). Quantum Hardware Devices (QHDs): Opportunities and Challenges. IEEE Access.
9. Sheelam, G. K. (2025). Advanced Communication Systems and Next-Gen Circuit Design: Intelligent Integration of Electronics, Wireless Infrastructure, and Smart Computing Systems. Deep Science Publishing.
10. Sonavane, A., & Aylani, A. (2025). Exploring the use of quantum algorithms. Quantum Computing and Artificial Intelligence in Logistics and Supply Chain Management.