BEIT Canada Inc. $\langle b \mid e^{it}$

CodeQraft

An Optimizing Compiler for Quantum Error Correction

Short Description of Project

TLDR: We are building ***CodeQraft****, a first-of-its-kind toolkit for logical quantum computing.*

The quantum computing landscape is well-equipped with a range of gate-level circuit builders, simulators, compilers and transpilers, such as tket and Qiskit, as well as automatic error suppression and mitigation tools such as Q-CTRL's Fire Opal and Mitiq. However, this space significantly lacks tools that aid in discovering and automating the practical implementation of quantum error correction (QEC). To fill this gap we are developing CodeQraft.

Our goals for CodeQraft are twofold:

1. A computational playground for discovering and manipulating quantum error-correcting codes and their associated fault-tolerant gate sets. As well as a home for modern QEC techniques like flag error correction [1], and quantum low-density parity-check codes [2].
2. An optimizing compiler that analyzes a quantum algorithm and outputs an optimized fault-tolerant logical circuit using an appropriate code or set of codes based on user requirements such as hardware connectivity, qubit count, gate thresholds and circuit depth.

This research-focused project emphasizes discovering and implementing the fundamental QEC protocols underpinning CodeQraft. See the longer description for more details.

Ideal Qualifications

- MITACS eligible student in the field of Quantum Computing, Computer Science, Physics, Mathematics, or related field.
- Demonstrated expertise in quantum computing and ideally quantum error correction, including knowledge of fault-tolerant quantum computing.
- Experience in collaborative software development and scientific computing in Python.
- Experience with using quantum hardware, quantum circuit simulations and quantum noise models.
- Excellent written and oral communication skills in English and the ability to work effectively in cross-functional and collaborative teams.
- Prior research work in Quantum Error Correction and Publications in peer-reviewed journals is a plus.

Logistics

- **Location:** In person —Suite H230-1, 101 College St. Toronto, ON M5G 1L7 (MaRS Center)
- **Number of positions:** 2-4
- **Position Length:** 4-8 months
- **Start dates:** $\geq$ September 2024, flexible

Long Description of Project

The proposed project, an "Optimizing Compiler for Quantum Error Correction (QEC)," addresses a critical and unmet need in the rapidly evolving field of quantum computing (QC). Despite significant advancements in QC, there is a lack of a specialized and commercial-ready tool to efficiently compile sophisticated quantum algorithms into robust error-corrected operations that are ready to run on the Quantum Processing Units (QPUs) that are anticipated to soon be abundant for commercial applications from drug discovery to logistics. The gap in comprehensive QEC tooling: The quantum computing landscape is well-equipped with a range of gate-level circuit compilers and transpilers, such as tket and Qiskit, as well as automatic error suppression and mitigation tools such as Q-CTRL's Fire

Opal and Mitq. However, this space significantly lacks tools that automate the intricacies of quantum error correction. Among the sparse solutions available, for instance, the Munich Quantum Toolkit's QECC library and the Qiskit community QECC package, there are notable limitations: these tools cater to a narrow spectrum of QEC codes and fail to encompass various advanced and more efficient methodologies such as flag fault tolerance, fault-tolerant preparation of encoded states and teleporting quantum states between codes. Modern innovative approaches like these are rapidly emerging from the literature, but are mostly lacking robust software implementations which greatly reduces their impact for the broader QC community and stunts further innovation and commercial applications. This shortcoming is critical because implementing different QEC codes has varying levels of complexity, particularly when it comes to performing logical gates on the code space. The crucial function of an optimizing compiler in this context is two-fold: not only to synthesize a fault-tolerant circuit that is tailored to a specific code, but also to analyze the initial non-fault-tolerant circuit and to select the most suitable code, or a combination of codes and projections, that optimizes logical implementations for the predominant gates in the algorithm and constraints imposed by the hardware. As of now, a fully featured QEC solution, capable of meeting the diverse and evolving needs of the QC community, remains absent in the market. This creates a pressing demand for a sophisticated tool that bridges this gap, enhancing both the efficiency and practicality of quantum computing applications. The urgency of this project is amplified by its potential impact on the field of quantum chemistry, particularly in material and chemical discovery. The project aligns with an existing project of ours that is funded by the European Innovation Council (EIC) on resource estimation and efficient fault-tolerant implementations for electronic structure calculations that are critical for applications such as battery design. By providing a means to find more efficient fault-tolerant implementations for quantum chemistry algorithms, this compiler will not only hasten the realization of these algorithms on quantum hardware, but it will also help to offer commercial value for organizations seeking to integrate QEC into their non-fault-tolerant solutions. The absence of such a tool represents a significant bottleneck in the progress of QC applications, making this project both urgent and vital for the advancement of the field.

Research components of this project will focus on three major areas.

- Fault Tolerant State Preparation

Most if not all quantum algorithms start by preparing qubits into a desired state which is typically a combination of $|0\rangle$, $|1\rangle$, $|+\rangle$ and $|-\rangle$ over all input qubits. Efficient methods for preparing these states on the logical level depend heavily on the structure of the quantum error-correcting code used and typical implementations are not fault-tolerant by design. Due to the lack of fault tolerance, simulations must be run to algebraically discover the potentially harmful heigh weight errors, then minimal syndrome detection circuits need to be designed fault tolerantly (often using modern methods like flag fault tolerance). Investigating and optimizing this process for a large library of quantum codes and implementing generalizable insights into robust software is the objective of this subproject.

- Fault-tolerant logical gates

Quantum error-correcting codes have an underlying structure that permits some set of logical level gates to be implemented using fault-tolerant combinations of physical gates + potentially a set of syndrome measurements and corrections. For well-studied codes such as the surface code [3] and Stean's code [4], such procedures are well understood, however, for other codes, there is a significant lack of best practices for implementing fault-tolerant logical gates–for example, even a code as simple as the $[\![8, 3, 2]\!]$ colour code [5] presents a significant challenge to implement a fault-tolerant logical Hadamard gate. This subproject would focus on cataloging implementations of fault-tolerant gates for a collection of quantum codes with the added focus on discovering new methods such as new code families with non-trivial transversal gates [6].

- Syndrome Readout and Error Correction

There are various proposals for performing syndrome readout and correction such as Shor [7], Steane [4], and Knill [8], which each come with a different set of constraints and resource requirements–for instance, Stean error correction [4] can only work with CSS codes. Newer protocols such as measurement-free QEC [9] and flag error correction [1] promise significant advantages in certain settings. The goal of cataloging error correction schemes is to identify and compare resource requirements across our library of codes to determine what scheme is the most suitable for different codes in different use cases. As with the subprojects above the goal will be to develop modular qiskit implementations of these schemes while searching for new innovations and generalizable insights.

Bibliography

[1] R. Chao and B. W. Reichardt, “Flag Fault-Tolerant Error Correction for any Stabilizer Code,” *PRX Quantum*, vol. 1, no. 1, Sep. 2020, doi: 10.1103/prxquantum.1.010302.

[2] S. Bravyi, A. W. Cross, J. M. Gambetta, D. Maslov, P. Rall, and T. J. Yoder, “High-threshold and low-overhead fault-tolerant quantum memory.” 2024.

[3] A. G. Fowler, M. Mariantoni, J. M. Martinis, and A. N. Cleland, “Surface codes: Towards practical large-scale quantum computation,” *Phys. Rev. A*, vol. 86, no. 3, p. 32324–32325, Sep. 2012, doi: 10.1103/PhysRevA.86.032324.

[4] A. M. Steane, “Error Correcting Codes in Quantum Theory,” *Phys. Rev. Lett.*, vol. 77, no. 5, pp. 793–797, Jul. 1996, doi: 10.1103/PhysRevLett.77.793.

[5] A. Kubica, B. Yoshida, and F. Pastawski, “Unfolding the color code,” *New Journal of Physics*, vol. 17, no. 8, p. 83026–83027, Aug. 2015, doi: 10.1088/1367-2630/17/8/083026.

[6] B. Eastin and E. Knill, “Restrictions on Transversal Encoded Quantum Gate Sets,” *Phys. Rev. Lett.*, vol. 102, no. 11, p. 110502–110503, Mar. 2009, doi: 10.1103/PhysRevLett.102.110502.

[7] P. W. Shor, “Fault-tolerant quantum computation.” 1997.

[8] S. Bravyi and A. Kitaev, “Universal quantum computation with ideal Clifford gates and noisy ancillas,” *Phys. Rev. A*, vol. 71, no. 2, p. 22316–22317, Feb. 2005, doi: 10.1103/PhysRevA.71.022316.

[9] M. A. Perlin, V. N. Premakumar, J. Wang, M. Saffman, and R. Joynt, “Fault-tolerant measurement-free quantum error correction with multiqubit gates,” *Phys. Rev. A*, vol. 108, no. 6, p. 62426–62427, Dec. 2023, doi: 10.1103/PhysRevA.108.062426.