Greater Quantum Efficiency by Breaking Abstractions

Research Statement

Yunong Shi
yunong@uchicago.edu

Quantum Computing (QC) is at the cusp of a computing revolution. However, building the first realistic quantum computer remains a great challenge that requires the collaborative efforts of device physics, engineering, architectural design and software integration. My research goal is to explore the vast design space for highly optimized and reliable quantum computing stacks. Current quantum toolchains follow a vertically layered design similar to its classical counterpart: algorithms - programming languages - instruction sets - technology mapping - error correction/mitigation - physical implementation. This layered design philosophy hides details in lower level from each layer, which allows software to be simpler and less complex. However, this approach misses opportunities for optimization. My current works develop new methods to improve the computation efficiency and reliability in different phases of quantum computation, mainly by exposing lower level details to higher abstraction levels [6]. These methods focus on problems in areas from high level software development and verification [8, 10, 3] to compilation [7, 2] to physical implementation of qubits [5].

Current work

Breaking the ISA abstraction by pulse-level compilation One central task of QC systems is quantum compilation, which involves decomposing a quantum function into instructions (quantum gates). It was established in the gate-based circuit model more than 20 years ago that such decomposition is possible for a universal gate set of 1- and 2-qubit gates; however, no decomposition algorithms in the last 20 years can saturate or even approach the theoretical lower bound in terms of gate count and circuit latency, except in some special cases. In addition, there is a mismatch between the gate set that circuit decomposition/construction algorithms are targeting and the operations directly supported by the hardware — translating between them adds even more overhead. While improving computing efficiency is always valuable, improving quantum computing efficiency is do-or-die: computation has to finish before qubit decoherence or the results will be worthless. Thus, improving the compilation process is one of the most, if not the most, crucial goals in near-term QC system design. However, restricted by the ISA abstraction, current efforts to optimize gate-based compilation in academia and in industry achieve limited improvement.

My research approach to quantum compilation optimization is to break the ISA abstraction and directly compile sub-circuits down to control pulses [7, 2]. In contrast to the current gate-based compilation, the framework developed in [7] aggregates 1- and 2-qubit gates into larger operations. The compiler then finds commutative operations that allow for much more efficient schedules of control pulses (In fact, it is the first quantum compiler that utilizes the commutation relations of quantum mechanics in gate scheduling). It next uses GRAdient-Pulse-Engineering (GRAPE) method in Quantum Optimal Control (QOC) on the aggregates to produce a set of control pulses optimized for the the under-lying physical architecture. Leveraging QOC theory, this compilation method achieves 2-10X improvement in circuit latency for near-term quantum algorithms. In the follow-up work [2], it further considered the optimization of compilation time for near-term variational algorithms and achieves 100X speedup in compilation latency. In addition to the speedup, this line of work is the first realization that there is a huge design space for quantum compilers beyond the gate-based compilation and offers an architectural choice that supports more complex QC applications (through lower latency, and thus much lower error rates). This compilation framework is a core component of an NSF Expedition in Computing project [1] and is considered to "boost the speed of quantum computers" by media [4].

Breaking the qubit abstraction by Bosonic qubit encodings In current QC stacks, the details of qubit implementations are often stripped away and only the (superpositions of) 0 and 1 information are retained. However, comparing to classical bits, quantum bits are far more fragile and short-lived. Thus, it's appealing to extract the rich error information provided by the physical qubit implementations and perform active error correction/mitigation. Bosonic codes like the Gottesman-Kitaev-Preskill (GKP) qubit encoding provide such architectures that enable upper level software to monitor the correctness of qubit states without destroying them. Based on Continuous-Variable (CV) QC system, these Bosonic codes can be realized in superconducting cavities, offering a promising alternative to current superconducting qubits, which suffers from leakage errors, short coherence time and erroneous gate operations. In addition to its error correcting capability, the GKP encoding is advantageous in eliminating leakage errors, providing longer qubit lifetime (thanks to the long cavity coherence time) and supporting robust gate operations for universal computation. The main difficulty of information processing with the GKP code is to prepare the highly non-classical code state. In [5], we gave the definition of fault-tolerance for the preparation of GKP codes and designed a protocol that fault-tolerantly prepares GKP states. We also gave detailed analysis for realistic errors including nonlinear dispersive shift and Kerr nonlinearity in the protocol. This work for the first time defined fault tolerance for a qubit-cavity system and is one of the first attempt to bridge the fault-tolerance community and the CV code community. This work has received a lot of attention and, although only a few months old, has 58 scites already on the peer-evaluation website *Scirate*.

With deeper integration with the upper level error correction, we believe that bosonic qubit architectures like the GKP qubit will serve as strong alternatives to the existing qubit implementations like the transmon qubits.

Reliable quantum software by formal verification The verification of quantum tool chains is a newly developed area and is of great importance for near-term quantum computing. I started and led a collaboration with IBM and Columbia University to design and implement the CertiQ verification framework [8] for the widely-used IBM Qiskit quantum compiler. CertiQ is the first mostly-automated verification tool for a real-world commercial quantum compiler. There are several challenges that makes formal verification of Qiskit difficult. First, checking the equivalence of quantum circuits is generally intractable. To mitigate this problem, CertiQ introduces the calculus of quantum circuit equivalence such that circuit equivalence and the correctness of compiler transformation can be statically and efficiently reasoned about. Based on the calculus, we design, specify, and verify a library of functions that perform primitive circuit transformations that are proved to be semantics preserving. Compilation phases implemented with this library can be easily verified using symbolic reasoning. The second challenge is that compiler implementations in community code submission can be complicated, making automated verification intractable due to state explosion. In CertiQ, we developed a novel way of combining symbolic execution and Design-by-Contract methodology to achieve high level of automation and scalable verification. We verified four compiler phases and seven transpiler pass implementations of Qiskit in four case studies. With these verified CertiQ implementations, we successfully identify three bugs in Qiskit, two of which are unique in quantum software.

We believe CertiQ is only the first step towards formal verification of the quantum software stack and the methods we developed in CertiQ can be extended to other quantum software components and pave the way to a truly reliable quantum tool chain.

Program-level error estimation by tensor networks Currently, no quantum compilers integrate error information from the devices to users, thus users cannot estimate the probability of success for their input circuit. In [9], we provide the first compiler-level error estimation framework to bound the errors in near-term quantum programs.

Reasoning about errors in a quantum circuit is a non-trivial task. First, it is difficult to efficiently simulate quantum computation. Quantifying the error effects usually involves a full simulation of the quantum computation because of quantum entanglement will change all the qubit states collectively. Secondly, characterizing and bounding realistic errors is complex due to the complicated error channels in real device.

In our work, error bounding is made possible by connecting the tensor network formalism to constrained Semi-Definite Programming (SDP) for calculating the diamond distance of quantum errors. We first utilize a class of efficiently computable tensor network called Matrix Product State (MPS) to approximate the circuit. Then we developed a method to transform the obtained MPS to a fine-grained description of the quantum state and use it as the linear constraint for SDP (Fig. 1). This technique gives us a greatly improved error bound comparing to the worst-case analysis of unconstrained diamond norm and the average-case analysis of fidelity. To characterize realistic errors in the quantum circuits, we adapted to a randomized compiling technique based on Pauli Twirling. The key idea is that quantum programs are evaluated by averaging over multiple runs, we can insert randomized gates in each run to change error statistics (without altering the circuit functionality). In this way, error channels that are hard to characterize, i.e., decoherence and cross-talks, are averaged and cast to simple stochastic errors.

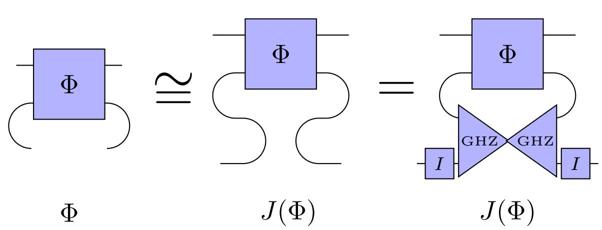

Figure 1: Tensor network representation of the Choi-Jamiolkowski isomorphism between a superoperator and a state. This enables a crucial step in the transformation of the MPS description to the linear constraint required by SDP.

Fault-tolerant QC by flag qubit scheme In fault-tolerant QC, universal quantum computing can be achieved by perfect Clifford operations and high fidelity magic states prepared by distillation protocols. However, magic distillation protocols also require perfect Clifford operations due to their non fault-tolerant nature, which requires layers of encodings and make the cost astronomical. In work in preparation, I collaborate with Quantum Error Correction (QEC) experts and adapt an efficient error detecting scheme utilizes flag qubits. Flag qubits are ancillary qubits that equipped with a pair of CNOT gates that can locate errors in between by returning a non-trivial error syndrome upon measurement (Fig. 2). By fault-tolerantly preparing magic states in a color code using the flag qubit scheme and then expand to a higher distance color code for distillation (Fig. 3), we show that the cost of fault-tolerant QC can be lowered by orders of magnitude.

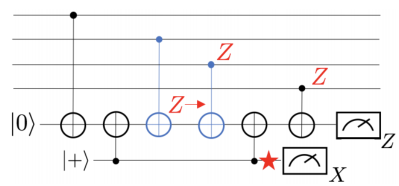

Figure 2: A flag qubit circuit that detects any weight 2 errors. The last qubit is the flag qubit.

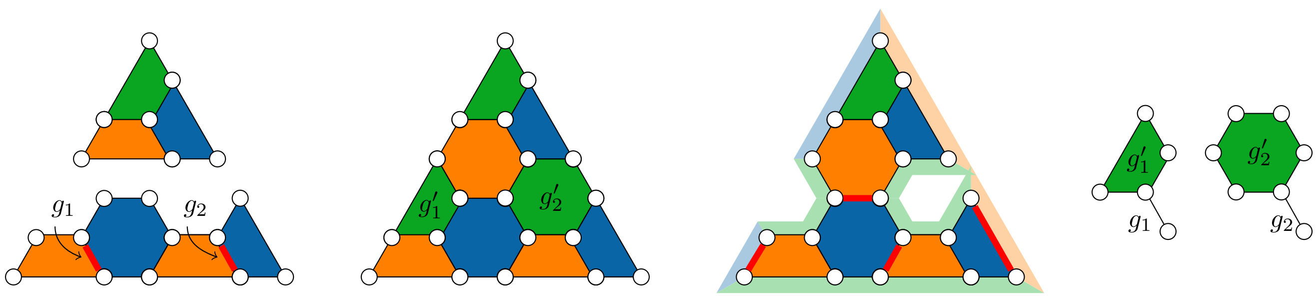

Figure 3: Fault-tolerant expansion of distance 3 color code (Steane code) to distance 5 color code using gauge fixing.

Future Research Agenda

My long term research goal is to bridge quantum software and quantum hardware and build a highly efficient and reliable vertical integration of QC stack. In the near future, I first plan to achieve this goal by further exploring my current directions,

Pulse-level compilation To maximize the potential of the current pulse-level compilation framework, I will further investigate into the following directions:

- I plan to work with physicists to perform experiments based on the pulse-level compilation and materialize its advantage. I also plan to work with computer architects to explore micro-controller designs that provide hardware-efficient gradient-based optimizations and fast feedback control, which could leverage current technologies in accelerators for Deep Learning. This attempt might lead to unexpected new designs of classical control system for QC platforms.

- I plan to develop new pulse optimizations that are algorithm-aware. GRAPE-based pulse generation in my current work performs optimizations in a high-dimensional parameter space and can become ineffective due to the high dimensionality. With the information from the specific quantum applications being optimized, the number of parameters can be largely reduced. The simplest example is the "partial pulse engineering" technique. In a quantum chemistry application, if we know an aggregated instruction is a Jordan-Wigner string with only the rotation angle of one qubit is varied, we can fix the pulse control fields of all Hamiltonian terms but the one that controls the rotation angle of that qubit. In this case, the optimization of this instruction becomes a 1-dimensional problem and doesn't scale with the instruction size.

- I plan to look into methods to achieve pulse-level parallelism for efficient scheduling. With my current work being the first compilation scheme to adapt commutation relations for gate-level parallelism, it's worthwhile to look into a more fine-grained pulse-level parallelism. For example, by utilizing the Baker-Campbell-Hausdorff formula in quantum mechanics to cancel the extra unitaries generated by altering the time order, two non-commuting aggregated instructions can be parallelized with high-precision approximation.

Bosonic qubit encodings For the direction of Bosonic qubit encodings, I plan to work with QC theorists to develop connections between kernel methods in Quantum Machine Learning (QML) and Bosonic qubit encodings. The infinite-dimensional Hilbert space in the Continuous-Variable (CV) QC system provides an ideal environment for kernel methods in QML classification tasks. Specifically, current Bosonic qubit encodings give intuition about how information can be efficiently encoded in such systems and what the associate kernel functions might look like. In a recent paper, it has been shown that squeezed states in CV systems can be used to encode classical information in a (classically-simulable) Gaussian kernel. With the GKP encoding and the phase estimation protocol adapted in my current work, it's possible to design more sophisticated data processing and preparation schemes that efficiently encode classical data into the feature space that breaks the I/O bottle-neck of current QML methods.

Reliable quantum software Building reliable quantum software will continue to be a central topic of my research in the future. I will focus in the following directions at different stages,

- As a long term goal, I will continue to collaborate with experts in formal verification to build a formally verified QC system. My current verification work of CertiQ focused on the compilation stage, I expect to extend the developed methods in CertiQ to high-level quantum algorithm libraries, hardware description language of the classical control system and eventually the pulse-level implementation of the quantum computation. A similar approach for classical computing is already outlined in the current work of verified OS kernels, file systems and virtual machines. The difficulty in the quantum case will be the contradiction between heavy optimization across abstraction layers and the requirement of modular verification for maintaining complexity. This will require novel design of certified abstraction layers in the verification framework.
- I will work with Programming Language (PL) experts to design more robust quantum software utilizing classical PL tools including type theory. I will use my past experience working on Qiskit and ScaffCC to guide the design of new types and control flows for quantum data.
- I also plan to use mechanized proof assistants like Coq to model Quantum Error Correction (QEC) theory for proving the correctness and detecting bugs of complicated QEC protocols. For example, some implementations of Color code decoding algorithms involve more than 20,000 lines of C code. To prove the correctness of these decoding methods will require enormous man×hour and only be possible by designing and applying highly-automated verification tactics in proof assistants. On a higher level, a formal verification framework can provide the QEC community with a standardized tool to compare, validate and improve QEC protocols, serving a role similar to SMT-LIB for SMT solvers with stronger verification support.

In addition to my current research topics, I will also start my research in the following new directions,

Algorithm-specific fault-tolerant protocols I will work with Quantum Error Correction (QEC) experts and quantum chemistry experts to explore methods that combines quantum chemistry algorithms with customized QEC scheme. Efficient fault-tolerant implementations tailored to specific quantum algorithms has been conjectured before and some recent work in quantum chemistry suggests that the veracity of the conjecture. These methods, including the recently proposed Generalized Superfast Encoding (GSE) and the Majorana Loop Stabilizer Code (MLSC), looked at Fermion-to-qubit mappings equipped with customized error correction/mitigation schemes. In GSE and MLSC, the overhead of mapping Fermionic operators onto qubit operators stays constant with the qubit number (as opposed to linear scaling in the usual Jordan-Wigner encoding or logarithmic scaling in the Bravy-Kitaev encoding). On the other hand, qubit operators in these mappings are logical operators of a distance 3 error correction code so that we can correct all weight 1 qubit errors in the algorithm. In a unpublished manuscript, I made the first attempt to generalize these methods to fault-tolerant protocols. In the future, I will utilize classical coding theory to scale up the error correcting ability of such error correction schemes and design more noise-aware quantum algorithm implementations.

Noise-adaptive compilation Near-term quantum devices have errors from elementary operations like 1- and 2-qubit gates, but also emergent error modes like cross-talk and non-Markovian errors. However, traditional error characterization methods consider only the errors in the fundamental operations and compose them, thus not able to account for the emergent errors for noisy devices. Recently, it has been shown that noise-tailoring techniques like randomized compiling could transform complicated noise channels including cross-talk, State Preparation And Measurement (SPAM) errors into simple stochastic Pauli errors, which could potentially enable subsequent noise-adaptive compilation optimizations. I plan to design compilation schemes based on these noise-tailoring methods that can minimize the noise in quantum circuits.

Classical control through Single Flux Qubit I will collaborate with experimentalists to explore novel classical control methods for QC systems. One prominent example is the Single Flux Qubit (SFQ) regime. Based on the superconducting technology, SFQ provides pico-second level control pulses and ultra-low power consumption. More importantly, it can co-locate in the dilution fridge with the qubit devices and serve as the classical data processing unit with fast feedback control. It has been demonstrated that SFQ can generate high-quality control pulses for 1- and 2-qubit gates and also be used in classical decoding algorithms for surface code. I plan to investigate into the SFQ logic, optimizations of SFQ generated pulses and related architectural design.

Dissipation-assisted error mitigation We generally think of dissipation as competing with quantum coherence. However, with careful design of the quantum system, dissipation can be engineered and used for improving the stability of the underlying qubit state. Previous work on autonomous qubit stabilization and error correction, including the Very Small Logic Qubit (VSLQ) and stabilization of parametrically coupled qubits, suggest that properly engineered dissipation could largely extend qubit coherence time. Exploring the design space of such systems and their associated error correction/mitigation schemes can provide us alternatives to an efficient and scalable quantum computing stack. I will collaborate with experts in device physics to design architectures based on these technologies.

References

[1] EPiQC, 2017. https://www.epiqc.cs.uchicago.edu.

[2] Pranav Gokhale, Yongshan Ding, Thomas Propson, Christopher Winkler, Nelson Leung, **Yunong Shi**, David I. Schuster, Henry Hoffmann, and Frederic T. Chong. Partial compilation of variational algorithms for noisy intermediate-scale quantum machines. In *Proceedings of the 52Nd Annual IEEE/ACM International Symposium on Microarchitecture*, MICRO '52, pages 266–278, New York, NY, USA, 2019. ACM.

[3] Ali JavadiAbhari, Shruti Patil, Daniel Kudrow, Jeff Heckey, Alexey Lvov, Frederic T. Chong, and Margaret Martonosi. Scaffcc: Scalable compilation and analysis of quantum programs. *Parallel Computing*, 45:2–17, Jun 2015.

[4] PhysNews, 2018. https://phys.org/news/2019-04-boost-quantum.html.

[5] **Yunong Shi**, Christopher Chamberland, and Andrew Cross. Fault-tolerant preparation of approximate GKP states. *New Journal of Physics*, 21(9):093007, sep 2019.

[6] **Yunong Shi**, Pranav Gokhale, Prakash Murali, Jonathan M. Baker, Casey Duckering, Yongshan Ding, Natalie C. Brown, Christopher Chamberland, Ali Javadi Abhari, Andrew W. Cross, David I. Schuster, Kenneth R. Brown, Margaret R. Martonosi, and Frederic T. Chong. Resource-efficient quantum computingby breaking abstractions. *PIEEE Special Issue*, Jan 2020.

[7] **Yunong Shi**, Nelson Leung, Pranav Gokhale, Zane Rossi, David I. Schuster, Henry Hoffmann, and Frederic T. Chong. Optimized compilation of aggregated instructions for realistic quantum computers. In *Proceedings of the Twenty-Fourth International Conference on Architectural Support for Programming Languages and Operating Systems*, ASPLOS '19, pages 1031–1044, New York, NY, USA, 2019. ACM.

[8] **Yunong Shi**, Xupeng Li, Runzhou Tao, Ali Javadi-Abhari, Andrew W. Cross, Frederic T. Chong, and Ronghui Gu. Contract-based verification of a realistic quantum compiler. *arXiv e-prints*, Aug 2019.

[9] **Yunong Shi**, Runzhou Tao, Xupeng Li, Jerry Lin, Frederic T. Chong, and Ronghui Gu. Bounding errors in quantum programs via tensor network (in preparation).

[10] Gadi Aleksandrowicz; ...**Yunong Shi**; ... Christa Zoufa. Qiskit: An open-source framework for quantum computing. 16, 2019.