Compilation Process for Multi-Controlled Gates in Qiskit

Shin Nishio[1], Takahiko Satoh[1, 2] and Rodney Van Meter[1, 2]
[1] Keio University Shonan Fujisawa Campuss
[2] Keio University Quantum Computing Center

available at

Tofoli gates and its generalization, controlled gates, serve as the core of many quantum algorithms, but **consume a great deal of quantum resources**. In this study, we added `BinaryControlledGate` class to Qiskit, an SDK for quantum computation, and designed and implemented a **compiler (transpiler pass)** for those class based on the common target rule proposed by Arabzadeh et al. **[1].** This work has not only made it easier to implement a more generalized quantum gate, but also made it feasible at a lower cost. As a result of compiling the quantum walk circuit based on the proposed method, the quantum cost (the number of required CX gates) was reduced from 345 to 181. The source code for the deliverables will be available on the GitHub repository this spring.

1. Background

The controlled gate is operations on an element of a Hilbert space controlled by another element of Hilbert space. The simplest controlled gate is CNOT gate, and there are generalized version of the CNOT gate or gates that control other gates(e.g. Fredkin gate). In this study, we describe the controlled gate based on **Feynman's notation[2]** and its extension shown in Fig.1.

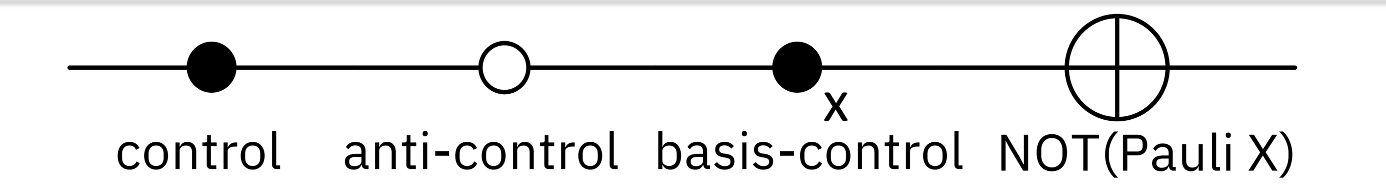

Fig. 1 The notation for the controlled gates

Qiskit is an SDK for Python-based quantum computing developed by IBM Research and others[3]. Qiskit treats quantum circuits as Python classes, which are translated and optimized into lower-level operations by compilers. As shown in Fig. 2, the compiler is composed by software called a **transpiler pass**.

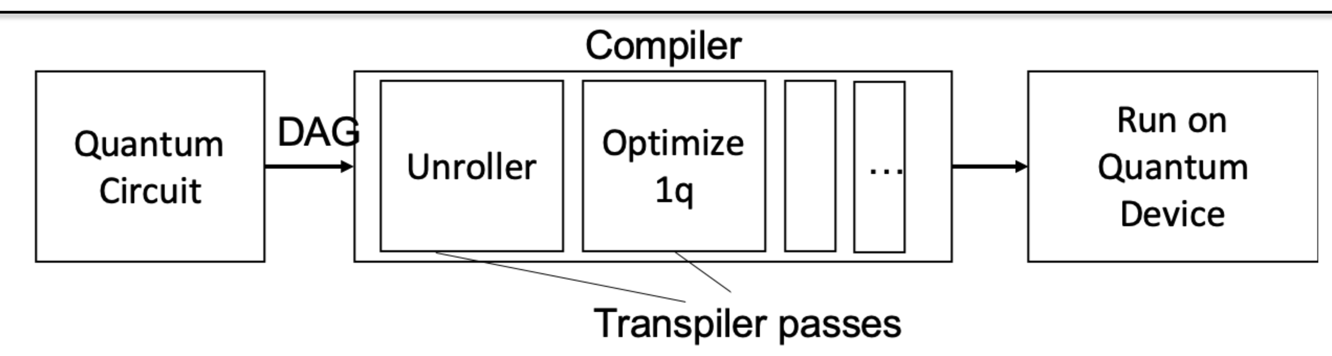

Fig. 2 Compiler for Qiskit

2. Our Method

Adding functions and classes

The `add_control()` function added in Qiksit ver 0.14 can internally call a gate such as mct (Toffoli gate with an increased number of control qubits) and create a Controlled Gate. In this research, we implemented `add_binary_control()` function and `add_basis_control()` as an extension of this function shown in Fig. 3.
These functions call 11 additional gates added by this study. The called gate is treated as `BinaryControlledGate` and `BasisControlldGate` class added in this study.

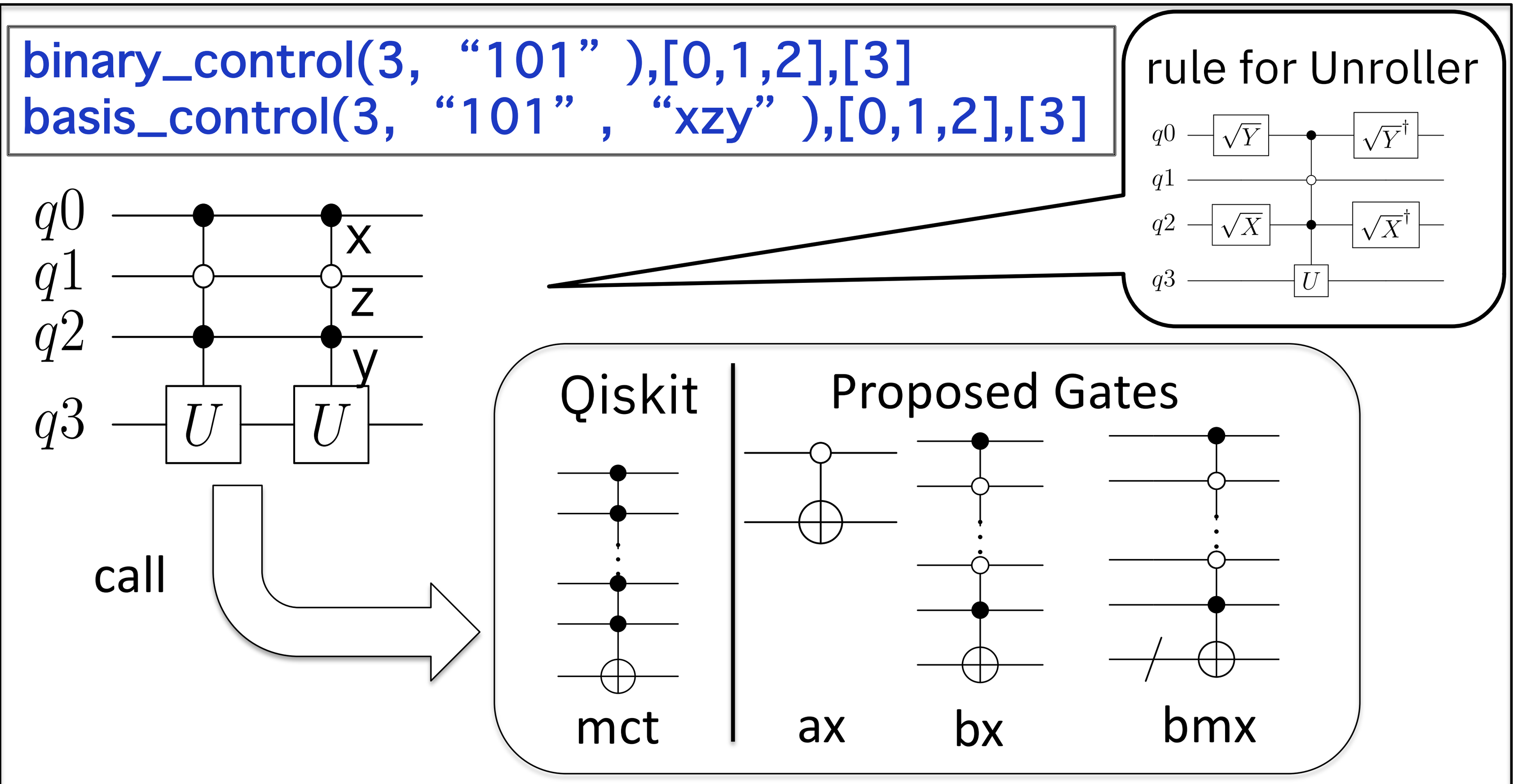

Fig. 3 Proposed Method for Controlled Gate

Transpiler path for controlled gate

We made the transpiler pass that simplifies the Controlled gate executed in the following steps based on the Common Target Rule proposed by Arabzadeh et al.**[1]**
The Common Target Rule is a rule as shown in Fig. 4 and can simplify adjacent Controlled Gates with the same target qubit.

1. Extract `q_condition` of `bx`(binary-controlled x) gate and `cnx`(n-qubit toffoli) gate with the same target qubit and is adjacent.
2. XOR the extracted binary and simplify it with the **Quine McCluskey method [4].**
3. Replace the gate in step 1 with step 2.

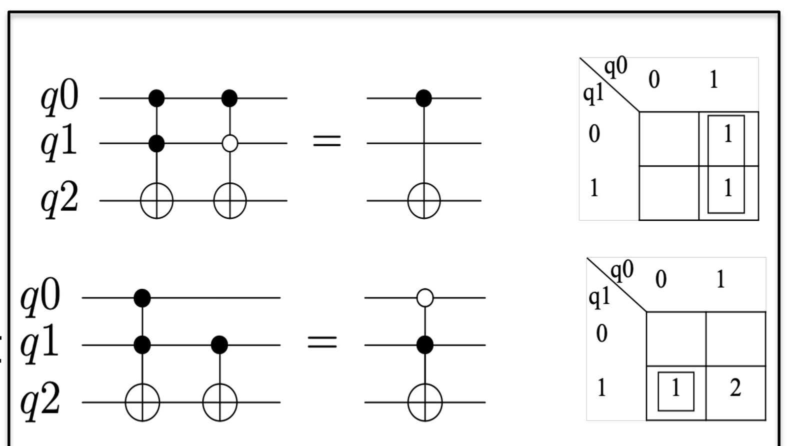

Fig. 4 the Common Target Rule

3. Evaluation

We simplified the quantum circuit of the quantum walk, which is a quantum algorithm that makes extensive use of controlled gates. As a result of applying the circuit to a transpiler, the quantum cost was reduced from 345 to 181. Quantum cost is the number of CX gates required when disassembling a circuit, was proposed by Barenco et al. [5] and values shown in Maslov et al.[6].
Since the evaluation is insufficient, we plan to compile Reversible Logic Synthesis circuits[6] and QASM benchmarks in the future.

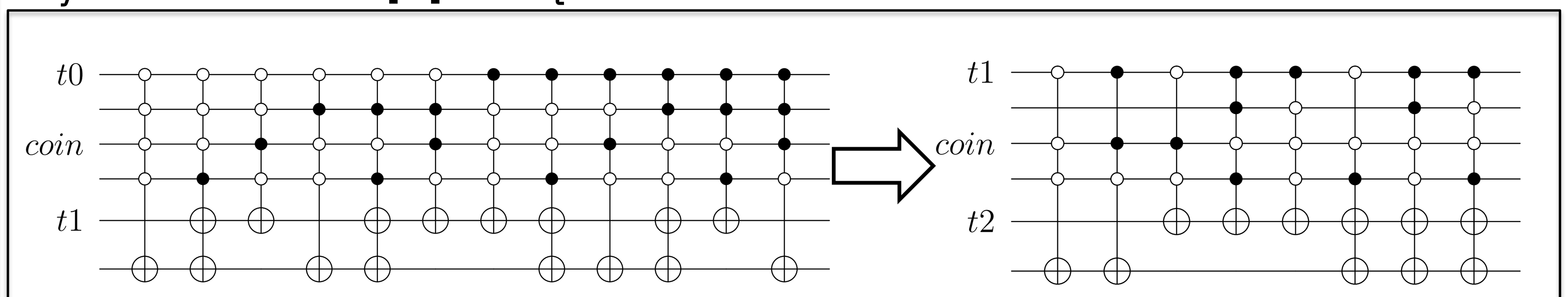

Fig. 5 The result of compiling the quantum walk circuit shown on the left based on the proposed method. Quantum cost reduced from 345 to 181

4. Conclusion and Future Work

This study not only **facilitated the implementation of a complex controlled gate**, but also reduced its **cost**.
The heuristic method is needed to deal with larger inputs, with the running time of the Quine–McCluskey method **grows exponentially** with the number of variables. The evaluation when used in **combination with other transpiler** passes is Future work.

References

[1] Mona Arabzadeh, Mehdi Saeedi, Morteza Saheb Zamani, **Rule-Based Optimization of Reversible Circuits**, arXiv:1004.1755[quant-ph] (2010)
[2] R. P. Feynman, **Quantum Mechanical Computers**, Optics News 11(2), 11 (1985)
[3] Héctor Abraham et al., **Qiskit: An Open-source Framework for Quantum Computing**(2019)
[4] Edward J McCluskey. **Minimization of boolean functions.** The Bell System Technical Journal , Vol. 35, No. 6, pp. 1417–1444, 1956.
[5] Adriano Barenco, Charles Bennett, Richard Cleve, David P. DiVincenzo, Norman Margolus, Peter Shor, Tycho Sleator, John A. Smolin, and Harald Weinfurter, **Elementary gates for quantum computation**, PhysRevA.52.3457(1995)
[6] Dmitri Maslov, **Reversible Logic Synthesis Benchmarks**(2005), https://webhome.cs.uvic.ca/~dmaslov/ accessed Jan. 2020

International Workshop for Young Researchers on the Future of Quantum Science and Technology, February 3rd - 6th, 2020