An Automated Tool for Mapping Program Variables to high fidelity on qubit on NISQ processor

Shin Nishio *, Takahiko Satoh†, and Rodney D. Van Meter ‡

Existing quantum computer processors have topological limitations on the execution of CNOT gates and unequal errors for each qubit. Programmers need to generate programs that are highly fidelity and based on topology-adapted qubit placement, but this is difficult. Therefore, the quantum program is optimized with software called a quantum compiler. When trying to obtain an optimum solution when performing quantum compilation, the compiler's computational complexity becomes very large, heuristic search is performed by setting initial correspondence of logical qubit and physical qubit and conducting beam search is needed. At that time, since Fidelity's predicted value is used as an evaluation function, we perform Randomized Benchmarking in this research.

Ⅰ. The Circuit Design Limitation Caused by Processor Topology

The number of qubits in the processor of a quantum computer is increasing day by day [1]. However, it is **difficult to generate the entangled state in any physically non-adjacent qubits** on the processor. In fact, all processors created by IBM are limited in the use of CNOT gates. A CNOT has a control qubit and a target qubit, but in the IBM quantum processors, the choice of qubit for each role is constrained.

This graph of limitation of use of the CNOT gate is referred to as **the processor topology**. For example**, Fig. 1** and **Fig. 2** show the topologies of IBM QX2 and QX4[2]. We have created tools for performing similar mapping [3] in prior work [4]. That work dealt with non-neighboring qubits, but not gate polarity. As shown in **Fig. 3**, by combining the CNOT gate and the Hadamard gate, it is possible to perform the same operation as the CNOT gate in the opposite direction, but care should be taken since the fidelity will be lowered.

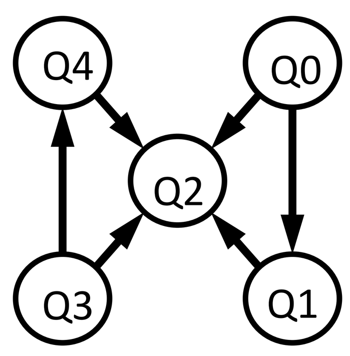


FIG. 1.
The Topology of IBMqx2

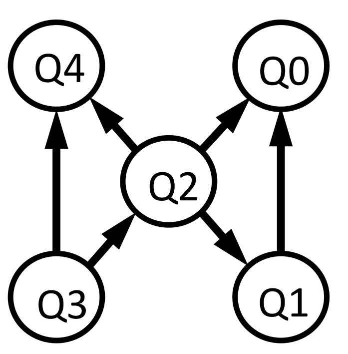


FIG. 2.
The Topology of IBMqx4

IBMQX2 and IBMQX4 are processors with the same number of qubits.
The direction of the arrows indicates the control (tail) and target (head) of CNOT gates.

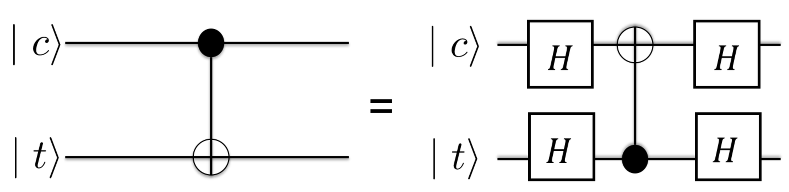


FIG. 3.
Implement a CNOT gate in the opposite direction

In order to assign variables used in the program to the qubit on the hardware to obtain high fidelity, it is necessary to refer to the topology of the processor and the error rate of each qubit. Also, allocation of variable should be automated in order to deal with large-scale and large amount of code and **increase the reusability** of various programs on different hardware. Like a compiler on a classical computer, we need an automated tool to efficiently use resources. From the similarity, what assigns qubits on the quantum circuit is called a **quantum compiler**.

In order to facilitate the calculation, we model that errors in the quantum circuit are composed of the following three. It is shown in Fig. 4.

a). Error accompanying operation of single qubit (**Single-Qubit Gate error**)
b). Error accompanying operation on multiple qubits (**Bi-Qubit gate error**)
c). Error accompanying state preparation and measurement (**SPAM error**)

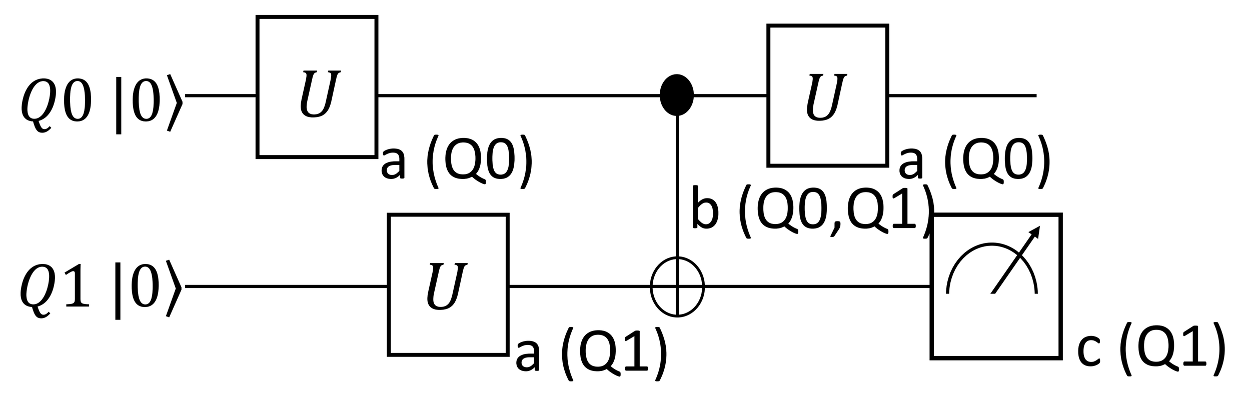


FIG. 4. Single-Qubit Gate error(a), Bi-Qubit Gate error(b), SPAM error(c)

References

[1] J. Q. You and Franco Nori, Superconducting Circuits and Quantum Information, Phys. Today 58 (11), 42 (2005) .
[2] IBM, IBM Q experience Device, https://quantumexperience.ng.bluemix.net/qx/devices (accessed 2018 01 15) (2017)
[3] Kaori Ishizaki, An algorithm for optimizing movement of quantum variables on arbitrary physical qubit structures, Bachelor's thesis, Keio University (2011)
[4] Choi, Byung-Soo and Van Meter, Rodney, On the Effect of Quantum Interaction Distance on Quantum Addition Circuits, J. Emerg. Technol. Comput. Syst, August 2011, 7, 3, 1550-4832, 11-17 (2011)
[5] Toshinari Itoko, Rudy Raymond, and Takashi Imamichi , Minimum Cost Mapping of Quantum Circuits to Quantum Computers (2018)
[6]角田 貴大, 量子ゲート平均忠実度測定方法としての Randomized Benchmarking の有効性と課題, 数理解析研究所講究録第2018巻 2017年 100-113 (2018)

Ⅱ. Computational complexity of Quantum Compiler

When evaluating errors accompanying quantum computation with the model as described above, searching $O(MQ!\,Q^{2M})$ is necessary in order for the quantum compiler to calculate the highest fidelity of circuit with **M** CNOT gates in a processor with **Q** qubit having variations in fidelity[5]. Since the increase in the number of qubits is assumed, the calculation amount is not practical.

In order to perform a heuristic search, there are methods such as.

1. Use the correspondence between logical qubits and physical qubits as **K** inputs
2. Beam search with **B** number of branch (Discard search that predicts fidelity below a certain level using evaluation function)

With these methods, the computational complexity is $O(MKQ^2 + M^2BQ^2)$

Ⅲ. Randomized Benchmarking

For fidelity optimization in compilation, it is possible to perform a heuristic search by using randomized bench- marking(RB)[6]. We are performing RB of IBM Q tokyo(20 qubit processor). The random benchmark evaluates the qubits on the processor assuming that the magnitudes of polarization of the multiple Clifford gates are equal. Since the polarization of the CNOT gate is overwhelmingly large compared with the polarization of the single qubit gate, it is excluded here and is performed only with X,Y,Z,H,S,S† gates.

In addition, by changing the number of Clifford gates to be executed and executing it multiple times, it is possible to obtain the value of the gate error as the attenuation factor. At this time, it is an advantage over the process tomography that the SPAM error and the gate error can be evaluated separately. The procedure of the RB performed this time is as follows. It is shown in Fig. 5.

1. Prepare |0⟩ state
2. Select Clifford gates

Randomly select and arrange n gates from X,Y,Z,H,S,S†. For example, (X−Z−S†−H)-(*)

3. Reverse the state

Create a gate row in which the reverse operation of the gate selected in Step 2 is arranged from behind. For (*), (H −S −Z −X)

4. Measurement

Measurement the state and get qubit's state. If get |0⟩ , calculation can be regarded as success. If get |1⟩ , result can be regarded as error.

In this research, we are going to do RB for all qubits of IBMQ20Tokyo which processor has 20 qubits. We are going to perform 1000 shots for 20 codes of 0, 10, 20, 30, 40, 50, 60, 70, 80, 90, 100 clifford gates. We need to run the quantum computer for 20 × 1000 × 20 × 11 = 4400000 times in total.

RB has been executed only for the 0th to the 6th qubits. Since the required number of times of execution is large, the seventh to nineteenth bit RB has not been completed yet. Also, it is necessary to evaluate the CNOT gates. For the 0 to 100 gate circuit, the results could be linearly approximated. The coefficient of determination is 0.94 in the 0 qubit of IBMQ20Tokyo. The Fig. 6. shows that the game error is 0.110 and the SPAM Error is 2.71. Fig.9 is the table summarizing the results.

Ⅳ. Conclusion

For RB, we need to end RB at single-qubit gate and CNOT gate at all qubits. Also, it is necessary to evaluate whether the number of trials is appropriate.

Also, it is necessary to simulate how the error distribution for each qubit influences the compiler's fidelity.

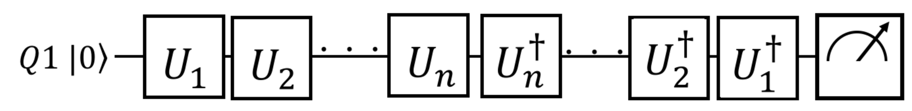
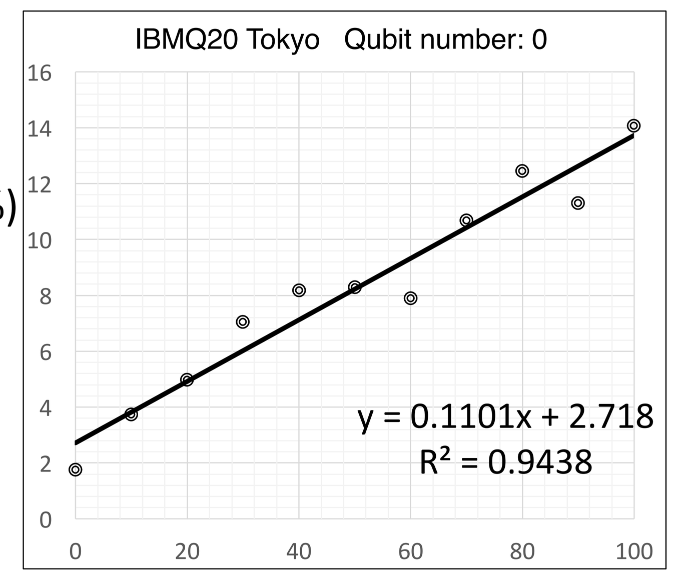


FIG. 5. Randomized Benchmarking

	0	1	2	3	4	5	6
G	0.1101	0.1138	0.0934	0.1436	0.1397	0.11	0.0523
$SPAM$	2.718	2.783	4.128	2.529	1.378	2.124	2.780
R^2	0.9438	0.5924	0.8272	0.9852	0.9904	0.9184	0.8857

FIG. 7. The result

Error rate(%)



Number of single-qubit clifford gates

FIG. 6. The result of RB(Q0)

物性研短期研究会 量子情報・物性の新潮流 , 2018年7月31日～8月3日

*慶應義塾大学 バンミーター研究室
†慶應量子コンピューティングセンター
‡ 慶應義塾大学環境情報学部

http://aqua.sfc.wide.ad.jp