The importance of randomness to quantum error correction algorithms and quantum error correction algorithms applicable to IBM Quantum

Daniel Tseng   
*Department of Information and Computer Engineering  
Chung Yuan Christian University*Taoyuan City, Taiwan(R.O.C)  
 daniel29348679@gmail.comTing Hsuan Chiang  
*Department of Information and Computer Engineering  
Chung Yuan Christian University*Taoyuan City, Taiwan(R.O.C)  
[sherrychiang01@gmail.comTsung-Wei](mailto:sherrychiang01@gmail.comTsung-Wei) Huang  
Undergraduate Program in Intelligent Computing and Big Data  
and quantum informagion center  
*Chung Yuan Christian University*Taoyuan City, Taiwan(R.O.C)  
tsungwei@cycu.edu.tw

*Abstract*—As classic computer technology advances, it has encountered a bottleneck. Under the von Neumann architecture, data storage technology progressed slowly, and it was difficult to significantly improve the computing power of classic computers. In comparison, although quantum computers are still relatively immature, their time complexity for specific problems far exceeds that of classical computers.

In a classical computer, a bit can only be in the state 0 or 1. However, quantum computers are not subject to this limitation, making quantum computers much faster than classical computers on certain computing tasks. However, until now there is not a perfect quantum computer system, and the qubits are always with error. As a result, error correction becomes extremely challenging, requiring fundamentally different approaches than traditional methods.

In our study, we use simulator to implement Shor’s error correction code (SRCC) and develop Shor’s error correction code with random method (SRCC-R) on bit flip errors problem. We consider the IBM quantum computer system with the error rate 0.007 and 0.002. In our result shows that using SRCC the flip errors can correct less than 25 controlled-not gates with 0.97 correct rate. In our method, SRCC-R the flip errors can correct larger than 500 controlled-not gates with 0.95 correct rate.

Keywords—grouping error correction in circle, Shuffle Randomly

# Shor's Error Correction Code

Shor's Error Correction Code is a type of quantum error correction code, which was proposed by American physicist Peter Shor in 1995[1]. The basic principle is to distribute the data of one qubit on multiple qubits.

## Repetition Code

Repeating codes are used to extend the data of one qubit to three qubits.

For status as below:



The encoded status of it is:

2

## Phase Code

Each physical qubit is then encoded into three qubits. The final encoded state is 9 qubits, and every three qubits correspond to the same state, representing an initial logical qubit.

Shor error correction code is one of the basic technologies in the field of quantum computing. It distributes the information of one qubit to multiple quantum qubits and uses redundancy to detect and correct common errors in the quantum computing process. Based on this algorithm, the conception of this article was formed. However, it should be mention that in 1996, a general code in 7 qubits code was discovered by Steane [2]. Also the general theory of quantum error correction was shown on papers of Calderbank and Shor [3] and Steane [4].

# Reasons for Error Correction

## Common Quantum Errors

Quantum errors comes from the hardware, including prepare initial state[5], coherence time[6], gate operation error[7], measure errors[8] and so on. There are many kinds of quantum errors. The following are common quantum errors.

* *Bit Flip Errors* : This error occurs when a quantum bit (qubit) flips from 0 to 1, or from 1 to 0, due to external interference or other reasons during the calculation process.
* *Phase Flip Errors* : This error involves the phase information of the qubit, where the qubit may be affected by external factors during the calculation process, causing its phase to be incorrect.
* *Gate Operation Errors* : Quantum computers use gate operations to perform calculations. These operations may be interfered by the external environment, or there may be some imperfect performance in the actual implementation, resulting in gate operation errors.
* *Qubit Crosstalk Errors* : Interactions between different qubits may lead to errors in information transfer between them, which may occur between adjacent qubits.
* *Phase Correction Errors* : These errors occur when trying to correct phase errors and can lead to further calculation errors.
* *Connectivity Errors* : Since the connection relationship between qubits may be limited, connection errors may occur during the calculation process, affecting the correct execution of the calculation.
* *Timing Errors :* These errors can result from inaccurate timing during calculations, resulting in incorrect synchronization of gate operations and qubits.
* *Measurement Errors* : At the end of a quantum calculation, the measurement operation may be disturbed, leading to errors in the calculation results.

## The Arrangement of the Quantum Computer Qubit Grid

The quantum error correction algorithm must not only consider the accuracy after error correction, but the two-qubit gate used for error correction must also conform to the arrangement of the quantum computer qubit grid.

* IBM Heron r1 processor(shown as Fig.1)

CZ-gate is mainly used by this quantum processor. The median error rate is 0.004101. Since quantum errors are not only bit-flip errors, but also phase errors, this error rate of 0.004101 will be too large to get the correct answer after more than a dozen of times of operation. Therefore, quantum error correction algorithms are very important for the successful operation of quantum computers.

# Implementation and Analysis

## Three-Bit Repetition Code

Repeating a qubit operation three times should result in three identical results. However, errors may occur during the operation, causing the three qubits to be different. The three-bit repetition code is to uniformly correct these three qubits to the majority answer, which means putting these three qubits into the full adder and taking the carry bit.

## Grouping Error Correction in Circle(shown as Fig.2)

For consider the structure of IBM quantum computer system, we consider the Shor’s error correction code in circle structure. Perform a qubit operation to be corrected six times, and group these results cyclically for error correction. Assign 012,123,234...and so on as a group to form a three-bit repetition code and perform error correction.

A screenshot of a computer

Description automatically generated

1. The current error rate map of IBM Heron r1 processor[5]

A blue and orange lines on a black background

Description automatically generated

1. grouping error correction in circle

But the results are not good, and sometimes the accuracy is even lower than before(shown as Fig.3). It is caused by consecutively errors. Assuming we're aim to correct all qubits to 0. An infinite loop will occur when more than one 1 (which are wrong) are encountered consecutively, and no matter how many times the error corrections we executed, those errors cannot be corrected to 0. That is, if there are continuous errors, there will be no way to correct them. Two qubits getting wrong consecutively is very easy to happen, especially cx-gate (or other dual qubit gates). Hence, we assume that to break the infinite loop, the sampling sequence must be disrupted. For cheking the assumtion we design the c and d subsection.

A graph of a number of people

Description automatically generated with medium confidence

1. Results of 6-Bit Grouping Error Correction in Circle

## Sampling according to Index: [-2,0,1]

First, we need to find out the best result in our simulator. After experiments, we found that sampling according to Index: [-2,0,1] (as shown in Fig.2) has the best error correction effect. The results will be shown in Fig.4.

A graph of a number of numbers

Description automatically generated with medium confidence

1. Results of Sampling according to Index: [-2,0,1]

## Write Back After all bits are finished

Although we ask the best result and write back to all qubits, the error still appeared. It is easy to check that more qubits are changed, the more errors will occur, like as :

100001 -> 110001 -> 111001…

The reason is that if the error correction result of the first group of three-bit repetition codes is still wrong, after writing back the 0-qubit, it will be sampled by the second group of three-bit repetition codes for error correction. This will obviously lead to a wrong step and a wrong result.

A better approach is to temporarily store the error correction results of each group of three-bit repetition codes into another group of qubits corresponding to the 6-qubit index, and then sample the next group of three-bit repetition codes. Original uncorrected errors to avoid each error correction affecting each other. After all error corrections are completed, the results are written back to the original q0. The results of *“write back after all bits are finished”* is shown as Fig.5.

A graph showing the correct number of gates

Description automatically generated with medium confidence

1. The results of “write back after all bits are finished”

As the results, it can be found that when the error rate becomes slightly larger, the error correction rate decreases significantly.

## Shuffle Randomly

According above thinking, if we can shuffle randomly chose index qubit, the error correction will be more efficient. Considering the actual arrangement of qubits, although this method of random shuffling does not exist in reality, we can observe whether "random shuffling" can help greatly improve the error correction capability, and it is found that it does.

Take the qubit arrangement of IBM Brisbane's quantum computer as an example to demonstrate the steps of random shuffling:

### The qubits in the orange box are the data to be corrected, and the qubits in the red box are the additional qubit space to store the error correction results. In the figure, the numbers 40, 41, and 42 in the orange box are adjacent qubits. First, these three qubits are taken for error correction (put them into the adder to get the carry bit), and the results are stored in the qubit numbered 60. In the same way, take numbers 44, 45, and 46 for error correction and store the results in qubit number 64(as Fig.6).

### After the error correction results have been stored in numbers 53 and 54, the qubit positions of the data to be corrected are exchanged in pairs. The exchange method is shown in the blue frame line in Figure 2. After completion, the qubit positions are shown in the orange line in Figure 3. When the two swap positions, a disruption is completed. After scrambling, new qubits numbered 40, 41, and 42 are obtained. These three qubits are used for error correction and the results are stored in qubit numbered 60. By analogy, after each disruption, the qubits numbered 40, 41, and 42 are taken for error correction, and the results are sequentially filled into the storage space below until the space is filled, and all error correction results are obtained. (as Fig.7)

A diagram of a data system

Description automatically generated with medium confidence

1. Step1

A diagram of numbers and a number

Description automatically generated with medium confidence A diagram of numbers and a number

Description automatically generated with medium confidence

1. Step2

Fig.8 show the results of our work.

A graph of a number of numbers

Description automatically generated with medium confidence

1. Results

# Conclusion

In conclusion, quantum error correction is a complex issue. Although all the methods we proposed could not maintain a consistent accuracy rate under an error rate of 0.007, if we optimistically use the latest quantum computer error rate from IBM (0.002) for experimentation, our method can indeed maintain the accuracy rate at a certain level (0.95). In this process, we also applied a significant error rate at a specific time to see if our error correction method could flexibly restore the accuracy rate to its original level. The results were excellent; our error correction method can maintain a certain accuracy rate on IBM's quantum computer. At the same time, we also understand the importance of random permutations for all error correction algorithms. When developing error correction algorithms in the future, attention should be paid to whether the method will cause random permutations.

##### Acknowledgment

We extend our gratitude to … for providing…

##### References

1. Shor P. W., Phys. Rev. A, 52 (1995) R2493-R2496
2. Steane A. M., Phys. Rev. Lett., 77 (1996) 793-767.
3. Calderbank A. R. and Shor P. W., Phys. Rev. A, 54 (1996) 1098-1105
4. Steane A. M., Proc. Roy. Soc. Lond. A, 452 (1996) 2551-2577.
5. I. H. Kim, Noise-resilient preparation of quantum many-body ground states, arXiv:1703.00032 (2017).
6. Aaron Somoroff, Quentin Ficheux, Raymond A. Mencia, Haonan Xiong, Roman Kuzmin, and Vladimir E. Manucharyan, Phys. Rev. Lett. 130, 267001
7. D. Willsch, M. Nocon, F. Jin, H. De Raedt, and K. Michielsen, Phys. Rev. A 96. 062306
8. Lena Funcke, Tobias Hartung, Karl Jansen, Stefan Kühn, Paolo Stornati, and Xiaoyang Wang, Phys. Rev. A 105, 062404
9. <https://www.ibm.com/quantum>