Quantum Error Stabilizers and their Impact on Quantum Error Correction

by

Saaketh Rayaprolu

CSCI 599

A paper submitted to Prof. Ming-Deh Huang in the Department of Computer Science in fulfillment of the final paper for the class CSCI 599: Quantum Computing and Quantum Cryptography

University of Southern California

December 13, 2021

Abstract

Stabilizers are a unique approach to quantum error correction. Not only do they constitute a simpler way to represent quantum states, but they also demonstrate the error tolerance of a state. Introduced and accelerated by Peter Shor and Daniel Gottesman, the conversation around stabilizer algebra has evolved from simply grouping various error correcting codes to understanding the structure of these stabilizers and utilizing them to better understand the structure of quantum errors that can and can't be mitigated via stabilizer codes. This paper will serve to explain the fundamentals of quantum stabilizers and their benefits within quantum error correction, after which we will discuss important properties of stabilizer algebra, explore the effects of circuits on stabilizers, and finally explore ongoing research areas involving stabilizers, such as error correcting subspace codes, subsystem codes, and the Hamming bound.

Introduction

In quantum information theory, stabilizers are often used as a class of error codes that can protect against decoherence, or environmental noise. Stabilizer codes are unique in that they approach quantum states through the lens of the operators applied to a state rather than the actual state. A unique set of operators one can follow is the Pauli set, a group of Hermitian and unitary 2×2 matrices which can represent any logic gate possible on a qubit. The Pauli matrices are as follows: $\sigma_X = \left(\begin{smallmatrix} 0 & 1 \\ 1 & 0 \end{smallmatrix}\right)$, $\sigma_Y = \left(\begin{smallmatrix} 0 & i \\ -i & 0 \end{smallmatrix}\right)$, and $\sigma_Z = \left(\begin{smallmatrix} 1 & 0 \\ 0 & -1 \end{smallmatrix}\right)$. These matrices have properties that make them easy to operate by; one such property is that all three of these matrices anticommute with the others. Given a global phase of ± 1 or $\pm i$, any two of these matrices can multiply to create the third, which is a property that will be crucial when dealing with stabilizers. Furthermore, the Hadamard matrix, defined as $\frac{1}{\sqrt{2}}$ $\left(\begin{smallmatrix} 1 & 1 \\ 1 & -1 \end{smallmatrix}\right)$, is a gateway between the X and Z Pauli matrices, such that $HXH = Z$ and $HZH = X$. Finally, given the basic quantum states $|0\rangle = \left(\begin{smallmatrix} 1 \\ 0 \end{smallmatrix}\right)$ and $|1\rangle = \left(\begin{smallmatrix} 0 \\ 1 \end{smallmatrix}\right)$, we can see that $Z|0\rangle = |0\rangle$ and $Z|1\rangle$ = -$|1\rangle$. Similarly, with $|+\rangle = \frac{1}{\sqrt{2}}\left(\begin{smallmatrix} 1 \\ 1 \end{smallmatrix}\right)$ and $|-\rangle = \frac{1}{\sqrt{2}}\left(\begin{smallmatrix} 1 \\ -1 \end{smallmatrix}\right)$, we can see that $X|+\rangle = |+\rangle$ and $X|-\rangle$ = -$|-\rangle$. Due to this property, we say that the Pauli Z matrix stabilizes $|0\rangle$ and $|1\rangle$, and X stabilizes $|+\rangle$ and $|-\rangle$.

Stabilizers and Generators

As we saw above, each base single-qubit state can be stabilized with a Pauli matrix; this property extends to multi-qubit states as well. For example, given the state $|0+\rangle$, we see that the first qubit is stabilized by Z and the second by X. It is also important to note that all states are stabilized by the identity matrix I, as by definition, it will not alter a state. Therefore, since each qubit has two stabilizers, we have a total of 2^n stabilizers for n qubits [1], or in this case, 4 stabilizers, which are *II, ZI, IX,* and *ZX*. By taking the nontrivial stabilizers of each qubit, we get what is called a generator of the stabilizers for that state. Since the nontrivial stabilizer for $|0\rangle$ is Z and for $|+\rangle$ is X, the generator would be $\langle ZI,\ IX\rangle$, the products of which would generate every stabilizer for the state $|0+\rangle$. More examples of quantum states and their stabilizer generators are shown in Table 1. Note that not all generators simply consist of stabilizers with all I except for one term; especially in the case of entangled states, generators will contain more complex stabilizer states.

CSS states, a special class of states determined by Calderbank, Shor, and Steane, contain the unique property of having generators containing stabilizers that are only composed of either X's and I's or Z's and I's [2]. The discovery of CSS states was important for quantum error correction (QEC) and fault-tolerant quantum computation (FTQC) as it determined a way to create codes such that machines only had to worry about one kind of error matrix in each stabilizer.

Phase Kickback

A powerful tool at the dispense of stabilizers is that they can be used as a substitute for quantum states in circuits. Rather than tracking operators throughout the course of a circuit, we can track the evolution of the stabilizers by which the quantum states identify. However, to do so, we must understand the effect of the CNOT gate, which is crucial for entanglement, on stabilizers. Similarly to the Hadamard gate, the CNOT must be applied from both sides of a stabilizer. This has interesting effects when applied to certain X or Z states. For example,

$$(\text{CNOT})(X\otimes I)(\text{CNOT}) = \begin{pmatrix} 1&0&0&0\\0&1&0&0\\0&0&0&1\\0&0&1&0 \end{pmatrix} \begin{pmatrix} 0&0&1&0\\0&0&0&1\\1&0&0&0\\0&1&0&0 \end{pmatrix} \begin{pmatrix} 1&0&0&0\\0&1&0&0\\0&0&0&1\\0&0&1&0 \end{pmatrix} = \begin{pmatrix} 0&0&0&1\\0&0&1&0\\0&1&0&0\\1&0&0&0 \end{pmatrix} = (X\otimes X).$$

Since CNOT is its own inverse, $X\otimes X$ maps to $X\otimes I$; however, $I\otimes X$ and $I\otimes I$ still map exactly to themselves. In other words, an X in the first term forces a multiplier of X in the second term. Similarly,

$$(\text{CNOT})(I\otimes Z)(\text{CNOT}) = \begin{pmatrix} 1&0&0&0\\0&1&0&0\\0&0&0&1\\0&0&1&0 \end{pmatrix} \begin{pmatrix} 1&0&0&0\\0&-1&0&0\\0&0&1&0\\0&0&0&-1 \end{pmatrix} \begin{pmatrix} 1&0&0&0\\0&1&0&0\\0&0&0&1\\0&0&1&0 \end{pmatrix} = \begin{pmatrix} 1&0&0&0\\0&-1&0&0\\0&0&-1&0\\0&0&0&1 \end{pmatrix} = (Z\otimes Z),$$

and consequently, $Z\otimes Z$ maps to $I\otimes Z$, but $I\otimes I$ and $Z\otimes I$ still map to themselves, meaning an Z in the second term forces a multiplier of Z in the first term. This property, phase kickback, is rather useful for quickly evaluating the evolution of stabilizers in a circuit. For example, given a CNOT over the stabilizer state $Y\otimes Z$, we can translate that to $iXZ\otimes Z$, which, after phase kickback, would become $iXZZ\otimes XZ$ $= X\otimes Y$.

Circuit Applications

As previously mentioned, stabilizers are crucial in the understanding of circuits, as they can substitute for the states they represent. Given any valid set of stabilizers, it is possible to design a circuit using the base states ($|0\rangle, |1\rangle, |+\rangle, |-\rangle$) and a sequence of CNOT and Hadamard gates (and possibly other unitary operations that may be present in the circuit). For example, given the circuit from Figure 1, due to the starting states being $|+\rangle, |+\rangle$, and $|0\rangle$, we start with the stabilizer generator set $\langle XII,\ IXI,\ IIZ \rangle$ and apply the given gates as follows:

$$\begin{matrix} XII & & XIX & & XIX & & XIX & & XIZ \\ IXI & \rightarrow CX_{1,3} \mapsto & IXI & \rightarrow CX_{2,1} \mapsto & XXI & \rightarrow SWAP_{1,3} \mapsto & IXX & \rightarrow H_3 \mapsto & IXZ \\ IIZ & & ZIZ & & ZZZ & & ZZZ & & ZZX \end{matrix}$$

Thus, the final generator set of the stabilizers for the final state after this circuit is $\langle XIZ,\ IXZ,\ ZZX \rangle$. Similarly, given any circuit, the stabilizers can be run through each transform to produce the final state. As a result, this property can be further expanded to more complex algorithms involving unitary transforms, which is an extremely useful property in QEC and FTQC.

This same effect also works the other way around. Given any valid stabilizer set, it is possible to create a circuit to take any product of $|0\rangle$ states to the stabilizer set. It is essentially a reverse of the process described above, as the CNOT and Hadamard gates, as we discussed above, are self-inverse. For example, given we start with the set of states [*XZZ,ZXZ,ZZX,IYY,XXX*], we first check for linear independence of the states. Since *IYY* is a product of *ZZX* and *ZXZ* and *XXX* is a product of the first three stabilizers, we can discard these stabilizers. Now that we have the generator set we hope to derive, we can do the following:

$$\begin{matrix} XZZ & & XIZ & & XIX & \\ ZXZ & \rightarrow CX_{2,3} \mapsto & -IXZ & \rightarrow H_3 \mapsto & -IXX & \rightarrow CNOT_{1,3} \mapsto \\ ZZX & & ZZX & & ZZZ & \end{matrix}$$

$$\begin{array}{lll} XII & XII & |+\rangle \\ -IXX \rightarrow CNOT_{2,3} \mapsto & -IXI \Rightarrow & |-\rangle \\ IZZ & IIZ & |0\rangle \end{array}$$

With these steps, we arrive at a series of base states that we can easily create from $|0\rangle$ state; Figure 2 shows the circuit that was determined above. Note that the gates applied will be in reverse order on the circuit, as it shows the evolution of the base state to the given generator set.

Further Research: Quantum Error Correcting Codes

One of the most important applications of stabilizer algebra is towards error correcting codes. In regular quantum systems, it is very hard to control miniscule, random errors that may affect the circuit from the bath, and as a result, it is imperative to cleverly encode the qubits into "logical" qubits to minimize the types of errors we encounter within the system. This is where stabilizers come into play, as the very essence of stabilizers is the ability to protect a given state against certain errors. Error correcting codes have been researched thoroughly, and while plenty of research continues to be done, some codes stand out as the current leaders in error correction and detection. These codes are often referred to as [[n,k,d]] codes, with n representing the number of qubits required to build the code, k representing the number of logical qubits generated by the code, and d representing the maximum distance of the errors guaranteed to be protected against by the code. It is important to note that for any [[n,k,d]] code, there will be n-k stabilizers in the generator, and consequently 2^{n-k} total stabilizers for the state, as explained above.

One such example is the 5-qubit "perfect" code, which encodes 5 qubits into one logical qubit with the stabilizer generator set $\langle XZZXI, IXZZX, XIXZZ, ZXIXZ \rangle$. Having stabilizers to refer to this [[5,1,3]] code is extremely handy, as states as elementary as $|\overline{0}\rangle$ and $|\overline{1}\rangle$ are each a superposition of 16 5-qubit states, and as such, very painstaking to write out or operate upon[1]. It is also interesting to note that the [[5,1,3]] code is a cyclic code, as each stabilizer can be thought of as a one-digit offset from the previous stabilizer. Finally, this code is known as a "non-degenerate" code, since it is possible to uniquely identify every error within the distance of the code, whereas degenerate codes are not capable of distinction before correction.

One example of a degenerate code is the [[9,1,3]] Shor code, which has a generator set of $\langle Z_1Z_2,\ Z_2Z_3,\ Z_4Z_5,\ Z_5Z_6,\ Z_7Z_8,\ Z_8Z_9,\ X_1X_2X_3X_4X_5X_6,\ X_4X_5X_6X_7X_8X_9\rangle$[1] (Figure 3 shows a simpler visualization of this code using a grid structure for the qubits). Currently, very crucial research surrounding a concept called the Surface code, a [[25,1,5]] code, is being conducted by Google, using a lattice structure to align information qubits with X and Z operators in a symmetrical and computationally efficient manner[3], an example of which is shown in Figure 4.

The Bacon-Shor code, derived from the Shor code, sparked research in the field of subsystem codes[6]. While previous codes limited the states of qubits to a subspace, subsystem codes separate the Hilbert space $\mathscr{H}$ of the original code to a tensor product of two spaces, $\mathscr{H}_A \otimes \mathscr{H}_B$. Errors on the subsystem $\mathscr{H}_A$ are corrected, but errors on $\mathscr{H}_B$, the gauge qubits, are discarded as they are in a trivial subsystem. While this sacrifices qubits, the benefit of the trade-off is that fewer stabilizers are needed in order to define the $\overline{X}$ and $\overline{Z}$ (and consequently $\overline{Y}$) gates (shown in Figure 5), making it far more computationally efficient. Further research on subsystem codes has been greatly pioneered by Aly, Klappenecker[4], and Sarvepalli[5].

There are many open optimization problems revolving around subspace and subsystem codes, such as maximizing the ratio (n:k) and distance of codes, and surpassing the quantum Hamming bound (QHB) with a degenerate code. The QHB is an upper limit bound on the maximum distance possisble on a code given the n and k values, and while it has been proven for non-degenerate codes, no conclusive result has been obtained for degenerate and subsystem codes quite yet[5]. Finding results that beat the QHB would be crucial, as it would allow codes to achieve higher distances than previously thought possible, which in turn would be a crucial advancement in QEC and QFTC. In short, the versatility of stabilizer algebra could hold the key to the future of the entire field of quantum computing.

Tables and Images

Table 1:

$\|000\rangle$	$\langle ZII,IZI,IIZ\rangle$, also noted as $\langle Z_1, Z_2, Z_3\rangle$
$\frac{\|00\rangle+\|11\rangle}{2}$	$\langle XX,ZZ\rangle$
$\|0+10-\rangle$	$\langle Z_1, X_2, -Z_3, Z_4, -X_5\rangle$
$\frac{\|01\rangle-\|10\rangle}{2}$	$\langle$-XX,-$ZZ\rangle$

Figure 1:

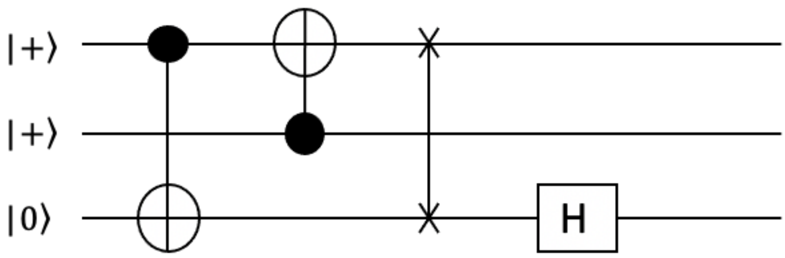


Figure 2:

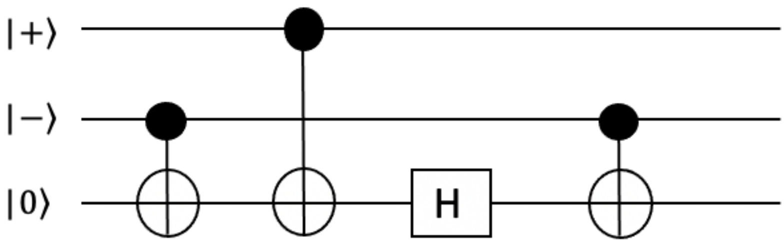


Figure 3:

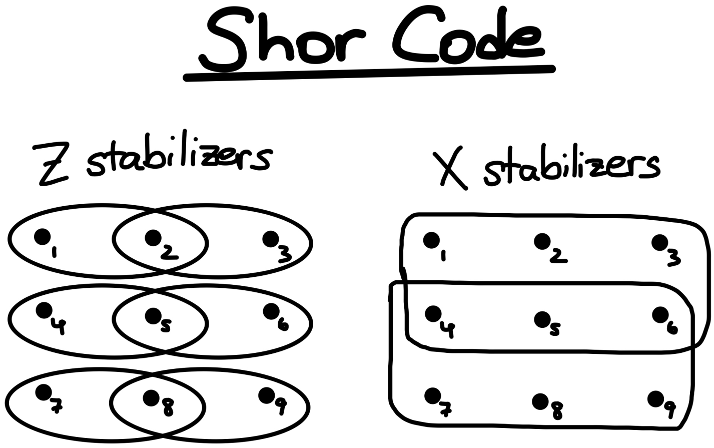
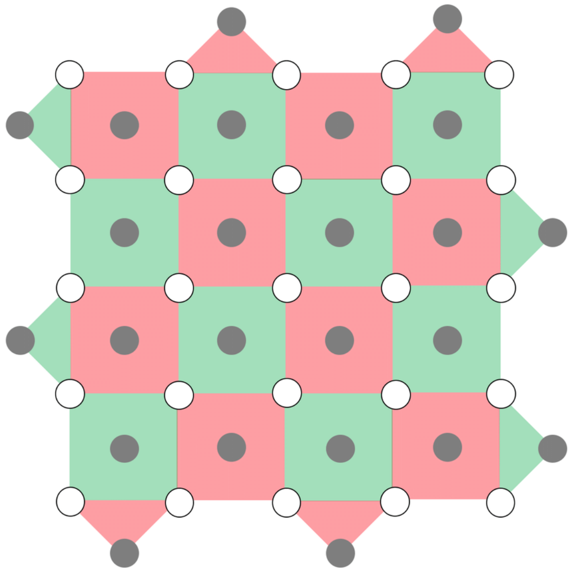
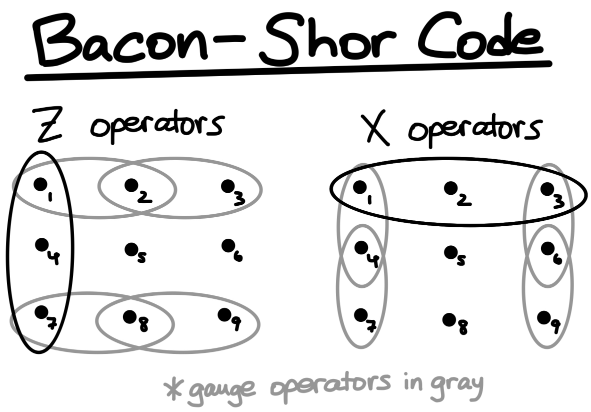


Figure 4:



The white circles are qubits, the dots in red are X, and the dots in green are Z.

Figure 5:



Bibliography

[1] Gottesman, Daniel Eric. "Stabilizer Codes and Quantum Error Correction." Thesis / Dissertation ETD, ArXiv, 1997, pp. 72–101, https://arxiv.org/pdf/quant-ph/9705052.pdf. Accessed 2 Dec. 2021.

[2] Lai, Ching-Yi, et al. "Fault-Tolerant Preparation of Stabilizer States for Quantum Calderbank-Shor-Steane Codes by Classical Error-Correcting Codes." Physical Review A, vol. 95, no. 3, 2017, https://doi.org/10.1103/physreva.95.032339.

[3] Fowler, Austin G., et al. "Surface Codes: Towards Practical Large-Scale Quantum Computation." Physical Review A, vol. 86, no. 3, 2012, https://doi.org/10.1103/physreva.86.032324.

[4] Aly, Salah A., and Andreas Klappenecker. "Constructions of Subsystem Codes over Finite Fields." International Journal of Quantum Information, vol. 07, no. 05, 10 Nov. 2008, pp. 891–912., https://doi.org/10.1142/s021974990900564x.

[5] Klappenecker, Andreas, and Pradeep Kiran Sarvepalli. "On Subsystem Codes Beating the Quantum Hamming or Singleton Bound." Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences, vol. 463, no. 2087, 2007, pp. 2887–2905., https://doi.org/10.1098/rspa.2007.0028.

[6] Bacon, Dave, and Andrea Casaccino. Quantum Error Correcting Subsystem Codes From Two Classical Linear Codes, ArXiv, 17 Oct. 2006, https://arxiv.org/pdf/quant-ph/0610088v2.pdf.