Final Report - Quantum Error Correcting Codes

John Donahue[1], Thomas Patecky[1], and Benjamin Spriggs[1]

[1]Portland State University

June 15, 2017

Abstract

Quantum computation, in order to be used reliably and in confidence, needs a reliable way to correct for errors (otherwise you might be able to accurately simulate an unreliable quantum computer by flipping a bunch of coins and accepting whatever aggregate binary string you get out as an answer) [1]. An interferometer is only as good as its ability to decohere accidental interference. Knill and Leflamme developed a general theory of error correction, building off of Shor's observation that basic assumptions can be made about a generality of error states, and that correction of an error does not necessarily entail that a qubit's state be entirely known [19]. Their paper described necessary properties for a coding and operators that allow for a quantum computation to both be checked for errors, and be able to fully utilize the power of quantum interactions [15].

Some examples of error-correcting and error-detecting codes are ones developed by Shor [7], another by Bacon and Shor [4], and surface code developed later by Fowler, Mariantoni, Martinis, and Cleland [13]. Stabilizer codes and their derivatives are most often used to determine if a state has changed to something it's not supposed to be via limiting the degrees of freedom of individual qubits, but allowing for enough degrees of a freedom in a group of error-resistant qubits to constitute a workable Hilbert space [14].

1 Goals

For our paper and presentation we intend to present a survey of error correction in quantum circuits. A natural first step of this goal is understanding the subject material to some degree, which is a non-negligible task in any scientific field with the appellation of "quantum". Once explanations of the error correcting circuits can be achieved beyond an incoherent melange of profanity and the words "thingie" and "doohickie", we can proceed.

Our survey will consist of background information on the problem of error correction for classical information, followed by the problems introduced by use of quantum information. Then we'll describe the listed algorithms for correcting qubit errors in rough order of complexity/discovery, starting with the basic 9-qubit Shor code and finishing with surface code. Along with explaining the algorithms, we'll discuss some of the interesting conclusions from the papers and their implications for actual physical implementation of a robust quantum circuit.

2 Classical Error Correction

To understand the nature of noise for quantum information, it is helpful to first review how noise works in a classical information setting.

A formal mathematical definition of noise was first developed by mathematician Claude Shannon in 1948 [18]. In simple terms when transmitting some amount of information in bits over a channel, noise can be described as each transmitted bit having some probability p of being altered. Since a bit only possesses two possible states, it's rather obvious what the resulting output of an altered bit would be: the bit is flipped.

Recovering the original information content of a message that has been affected by noise is accomplished via Error Correcting Codes, which function by adding some amount of additional redundant bits computed from a function on the original message in order to detect and hopefully pinpoint bit flip errors. Assuming a correctly designed algorithm, these redundant bits can reduce the probability of transmitting an erroneous message to an arbitrary degree.

For any given string of bits $\{0,1\}^n$, the total probability of the message having no errors with a bit flip error probability of p for any given individual bit is $(1-p)^n$. When p is extremely small, such as is the case in memory stored to disk, it suffices to simply checksum the data to detect the presence of error and then retransmit in that event. However as soon as p grows beyond negligibility, the total probability of error exponentiates and redundancy becomes necessary in order to preserve the integrity of the data.

3 Quantum Errors

3.1 Arbitrary Quantum Errors

When extending the concept of signal noise to quantum information, several factors exist that complicate our model beyond the simple bit flip error and correction via copying redundant information. Perhaps most significantly, the fact that a qubit is defined by a pair of complex numbers means that any potential error on a qubit has not merely two potential states as in the classical case, but an uncountably infinite number of potential error states. This arbitrary error can be modeled as a unitary transformation U applied to the quantum state during transmission.

The implication of this is that any quantum error-correcting circuit necessarily needs to be able to precisely diagnose the transformation applied by signal noise and then apply another reversing transformation to recover the original information.

3.2 Diagnosing Quantum Errors

In attempting to determine what if any transformation has occurred to a qubit via noise, we run into our second issue in quantum computation that complicates error correction. Observation of a qubit collapses its state from a pair of complex numbers down to a single classical bit, essentially eliminating the qubit's computational power in the circuit. Clearly we want to preserve the state of our data, while also analyzing the qubit to see if it has been transformed.

If we had this problem of observation causing state loss in a classical circuit, a simple solution would be to copy the bit in order to preserve the information. Of course the no-cloning theorem precludes this strategy: a quantum state can be teleported but not duplicated. So we are left with the following problem to solve: take a qubit that cannot be copied, potentially apply a random unitary transformation to it, and then without observation detect and reverse that transformation.

3.3 Syndrome Measurement

As it turns out, we are able to work around the limitations of quantum information to still perform error correction. As an alternative to copying qubits and violating the no-cloning theorem, it is possible to spread the information contained in a single qubit by entangling it with ancilla qubits and sending that collection of qubits over the noisy channel. On the receiving end, the challenge then becomes locating which of the entangled bits were transformed and reversing that transformation to rebuild the original qubit.

Our ability to reverse the transformation applied by noise depends on being able to reduce the myriad possible potential errors into a finite set of errors that can be measured and corrected in a quantum circuit. We take advantage of the fact that the identity matrix I and three Pauli operators X, Y, and Z for a basis by which we can define any arbitrary unitary operator U as a linear combination of those basis operators. We further take advantage of a property of syndrome measurement in our correcting circuit, namely that the arbitrary error it itself projected onto this basis space. This reduces a potential error down to either none or one of the Pauli operators and allows our circuit to reverse that Pauli operation and recover the original quantum state, as we will see with the Shor code.

3.4 Non-unitary Quantum Errors

Some other non-systematic, environmental errors are more tricky - in the case of other quantum systems becoming somehow entangled or interacting with the system that we are performing computations with. As well, many physical systems that can be used to model quantum systems have more degrees of freedom than anticipated (ex. Ca^+ ion w/ >2 levels of freedom), and can cause leakage of information, or even a collapse of the state to a vacuum state.

4 Quantum Error Correction

4.1 Existence of Reliable QECC

Calderbank and Shor outlined many definitions useful for recovering quantum states from coherent errors via projection onto a codeword-based Hilbert subspace. [7]

> A code C of length n is a set of binary vectors of length n, called *codewords*. In a *linear code* the codewords are those vectors in a substace of F_2^n (the n-dimensional vector space over the field F_2 on two elements). ... A linear code with length n, dimension k, and minimum distance d is called an $[n, k, d]$ code.

The elements of a quantum error correcting code act on encoded states: $Q : \mathcal{H}_2^k \mapsto \mathcal{H}_2^n$, where the amount of information in a codeword compared to some arbitrary string of Q $= k/r$. The properties of this code depend on the subspace of $\mathcal{H}_2^n$, $Q\mathcal{H}_2^k$. Calderbank and Shor leave this projection for a simpler orthoganal basis on the subspace of $\mathcal{H}_2^n$, where the elements of this basis are the quantum codewords.

Using these definitions, they proved the existence of a code that would correct for a variety of errors in a fault-tolerant way, opening the way for codes like [8], which use smaller overhead error-catching ancilla gates to stabalize a logical qubit.

4.2 Properties of QECC

Considering a QECC in the context of purifying singlets from mixed states [6], the goal of a QECC might look like this:

Assume there is a state $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$, that is encoded by a unitary transformation to a higher dimensional Hilbert space, with basis vectors $|v_0\rangle$ and $|v_1\rangle$. Can we choose these basis vectors such that the original state can be decoded after applying Werner-type errors (un-normalized linear operators to the original state vector, including measurement of random qubits) to the encoded state?

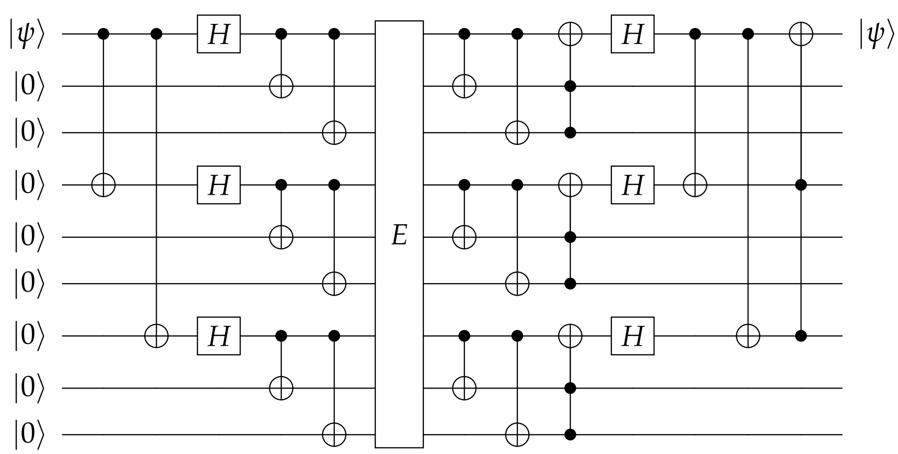
The necessary conditions for these basis vectors are that the environment producing the Werner-type errors can acquire no additional information about the original input state by random measurement:

$$\forall i \ \cdot \langle v_0|R_i^\dagger R_i|v_0\rangle = \langle v_0|R_i^\dagger R_i|v_0\rangle \tag{1}$$

$$\forall i \ \cdot \langle v_1|R_i^\dagger R_i|v_0\rangle = 0 \tag{2}$$

4.3 Shor Code

The Shor code is a simple algorithm developed by Peter Shor in 1995 that is capable of correcting an arbitrary single qubit error [19]. Most qubit errors are either bit flips, sign flips, or both, and represented by the three Pauli operators X, Y, and Z respectively. However any arbitrary single qubit error can be described as a unitary transformation U. This arbitrary unitary operation can be expressed as a superposition of basis operations, in this case a linear combination of the Identity matrix and the three Pauli operators: $U = \alpha I + \beta X + \gamma Y + \delta Z$. Error syndrome measurement has the projective effect of measurement which forces an arbitrary qubit error into a specific Pauli operation, meaning that an error-correcting circuit capable of correcting both sign and phase errors is sufficient to correct any arbitrary single qubit error.



The Shor code works by entangling the qubit with the information we wish to protect with eight other qubits, sending those qubits through the noisy channel, and then performing syndrome measurement to detect and correct any bit flip or phase flip error that might have occurred. The initial qubit $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$ is transformed into the following product state.

$$|\psi'\rangle = \alpha(\frac{1}{2\sqrt{2}}(|000\rangle + |111\rangle) \otimes (|000\rangle + |111\rangle) \otimes (|000\rangle + |111\rangle))+$$
$$\beta(\frac{1}{2\sqrt{2}}(|000\rangle - |111\rangle) \otimes (|000\rangle - |111\rangle) \otimes (|000\rangle - |111\rangle))$$

After passing through the noise transformation E, if none or one of the qubits has had an error applied, the circuit is able to correct that error. The first set of CNOT gates reverses a possible bit flip error by comparing the eigenvalues of pairs of qubits to isolate the location of the bit flip and reverse it. The second set of CNOTs similarly works to isolate and reverse a phase flip within any one of the three-qubit clusters.

It is important to note that this circuit only succeeds in the event of zero or one error occurring in the set of nine qubits transmitted. Particularly if two phase flips occur then this circuit will perform a bit flip on the original state, and vice versa should two bit flips occur. Assuming each qubit is independently subjected to the same probability p of an arbitrary error occurring, the total probability of the circuit being in error is bounded by $16p^2$. Thus, the Shor code only increases the reliability of transmission if $p < \frac{1}{16}$. [16]

4.4 Stabilizer Formalism

A formalism to describe codes that satisfy the necessary conditions for higher-dimensional encodings of qubits was developed by Daniel Gottesman in a 1997 paper [14]. In it, he describes a group-theoretic structure, and a subclass of QECC he names "stabalizer codes". The error model of the stabilizer formalism that he uses assumes independent Pauli-gate errors on different qubits, with equal probability of coming across a $\sigma_x, \sigma_y, \sigma_z$ error on a error-ed qubit. This model is deficient in capturing the generality of errors, but Gottesman continues on to describe the necessary properties of a group of operators that makes them a candidate for a stabilizing code.

The group is composed of operators that do not transform valid codewords:

$$M|\psi\rangle = |\psi\rangle$$

But do transform error-ed codewords back to valid ones (an operator M commutes with another Pauli error E):

$$ME|\psi\rangle = EM|\psi\rangle = E|\psi\rangle$$

And by measuring the eigenvalues of each of the operators in the group, an error syndrome can be reproduced, without disturbing the state vector of the original encoded state in the codeword space (the error syndrome was later proved not needed to be fully revealed, and still be able to reproduce a signal on a noisy quantum channel by [20]).

An example of a "perfect" stabilizer code is described by Gottesman1 (every possible error syndrome is used by the single-qubit errors, a property shared by Shor's 9-qubit code).

$$\begin{array}{c|cccccccc}
M_1 & \sigma_X & \sigma_X & \sigma_X & \sigma_X & \sigma_X & \sigma_X & \sigma_X & \sigma_X \\
M_2 & \sigma_Z & \sigma_Z & \sigma_Z & \sigma_Z & \sigma_Z & \sigma_Z & \sigma_Z & \sigma_Z \\
M_3 & I & \sigma_X & I & \sigma_X & \sigma_Y & \sigma_Z & \sigma_Y & \sigma_Z \\
M_4 & I & \sigma_X & \sigma_Z & \sigma_Y & I & \sigma_X & \sigma_Z & \sigma_Y \\
M_5 & I & \sigma_Y & \sigma_X & \sigma_Z & \sigma_X & \sigma_Z & I & \sigma_Y \\
\hline
\overline{X}_1 & \sigma_X & \sigma_X & I & I & I & \sigma_Z & I & \sigma_Z \\
\overline{X}_2 & \sigma_X & I & \sigma_X & \sigma_Z & I & I & \sigma_Z & I \\
\overline{X}_3 & \sigma_X & I & I & \sigma_Z & \sigma_X & \sigma_Z & I & I \\
\overline{Z}_1 & I & \sigma_Z & I & \sigma_Z & I & \sigma_Z & I & \sigma_Z \\
\overline{Z}_2 & I & I & \sigma_Z & \sigma_Z & I & I & \sigma_Z & \sigma_Z \\
\overline{Z}_3 & I & I & I & I & \sigma_Z & \sigma_Z & \sigma_Z & \sigma_Z
\end{array}$$

Table 1: An example stabilizer a eight-qubit code described by Gottesman.

4.5 Compression Formalism

The Pauli-matrix error assumptions that lie at the heart of the Stabilizer formalism lacks the ability to generalize errors and correct errors on arbitrary quantum channels with reliable results. The Compression formalism, which uses a higher-order matrix analysis framework, attempts to present another attempt at capturing the general error on an arbitrary channel [10].

4.5.1 Compression Problem

Numerical ranges of a matrix $T \in \mathbb{C}^{nxn}$ are defined:

$$\Lambda_1 = W(T) = \{\langle T\psi|\psi\rangle : |\psi\rangle \in \mathbb{C}^n,\ |||\psi\rangle|| = 1\}$$

Where Λ_1 represents the $k = 1$ numerical range. The k-rank range is then:

$$\Lambda_k = \{\lambda \in \mathbb{C}^n : PTP = \lambda P \text{ for some rank-}k \text{ projection } P\}$$

The "compression problem" is the question of what those scalars λ_i are, and their associated projections P_i are.

4.5.2 Properties of Higher Rank Numerical Ranges and Application

Various properties of higher rank numeric ranges, also studied by the authors [9], were used to develop codes that work on an arbitrary bipartite qubit system.

Some of the properties:

- The rank-k numerical range of a n dimensional Hermitian operator invariant under complex conjugation coincides with the set $[a_k, a_{n-k+1}]$ ($k \geq 1$ and is fixed).
- For the same Hermitian operator σ, the rank-k numerical range has the following property:

$$\Lambda_k(\sigma) \subseteq \bigcap_{\Gamma} \mathrm{co}\Gamma$$

And a third property used later to determine the compression values λ_i for the higher rank numeric range:

> Let U be a unitary on $\mathcal{H} = \mathbb{C}^4$. Then we have the following characterizations of the numerical ranges for U:
>
> (i) $\Lambda_1(U) = W(U)$ is the subset of the unit disk in $\mathbb{C}$ given by the convex hull of the eigenvalues for U.
>
> (ii) $\Lambda_2(U)$ is non-empty and given as follows. Let $z_k = e^{i\theta_k}$, with $\theta_k \in [0, 2\pi)$ for $k = 1, 2, 3, 4$, be the eigenvalues for U, ordered so that $0 \leq \theta_1 \leq \theta_2 \leq \theta_3 \leq \theta_4 < 2\pi$.
>
> (a) If the spectrum of U is non-degenerate, so $\theta_k \neq \theta_j\ \forall k \neq j$, then $\Lambda_2(U) = \{\lambda\}$ where λ is the intersection point in $\mathbb{C}$ of the line ℓ_{13} through z_1 and z_3, with the line ℓ_{24} through z_2 and z_4.
>
> (b) If U has three distinct eigenvalues, say $\theta_j = \theta_k$ for some pair $j \neq k$ but $\theta_j \neq \theta_l$ otherwise, then $\Lambda_2(U) = \{z_j\}$.
>
> (c) If U has two distinct eigenvalues, each of multiplicity two, say z and w, then $\Lambda_2(U)$ consists of the line segment $L = [z, w]$ joining z and w.

(d) If U has two distinct eigenvalues, one $\lambda = z$ with multiplicity three and the other with multiplicity one, then $\Lambda_2(U) = \{z\}$.

(e) If U has a single eigenvalue $\lambda = z$, then U is the scalar operator $U = z\mathbb{1}_4$ and $\Lambda_2(U) = \{z\}$.

(iii) $\Lambda_3(U)$ is non-empty if and only if $\Lambda_3(U) = \{\lambda_0\}$ is a singleton set and λ_0 is an eigenvalue for U of geometric multiplicity at least three;

$$\dim(\ker(U - \lambda_0\mathbb{1})) \geq 3. \tag{3}$$

(iv) $\Lambda_4(U)$ is non-empty if and only if U is a scalar multiple of the identity operator.

And more generally, to find all correctable codes of a bi-unitary channel $E = \{V, W\}$ [9]:

1. Compute the set of compression-values λ_i of $\Lambda_2(V^\dagger W)$ using the above properties.

2. Obtain the family of projections P that satisfy $PV^\dagger WP = \lambda P$ for each compression-value λ_i.

The subspaces corresponding to P are precisely the correctable codes for $E = \{V, W\}$.

The authors note that, in order for codes to be developed out of the formalism, explicit projections have to be identified.

5 Surface Code

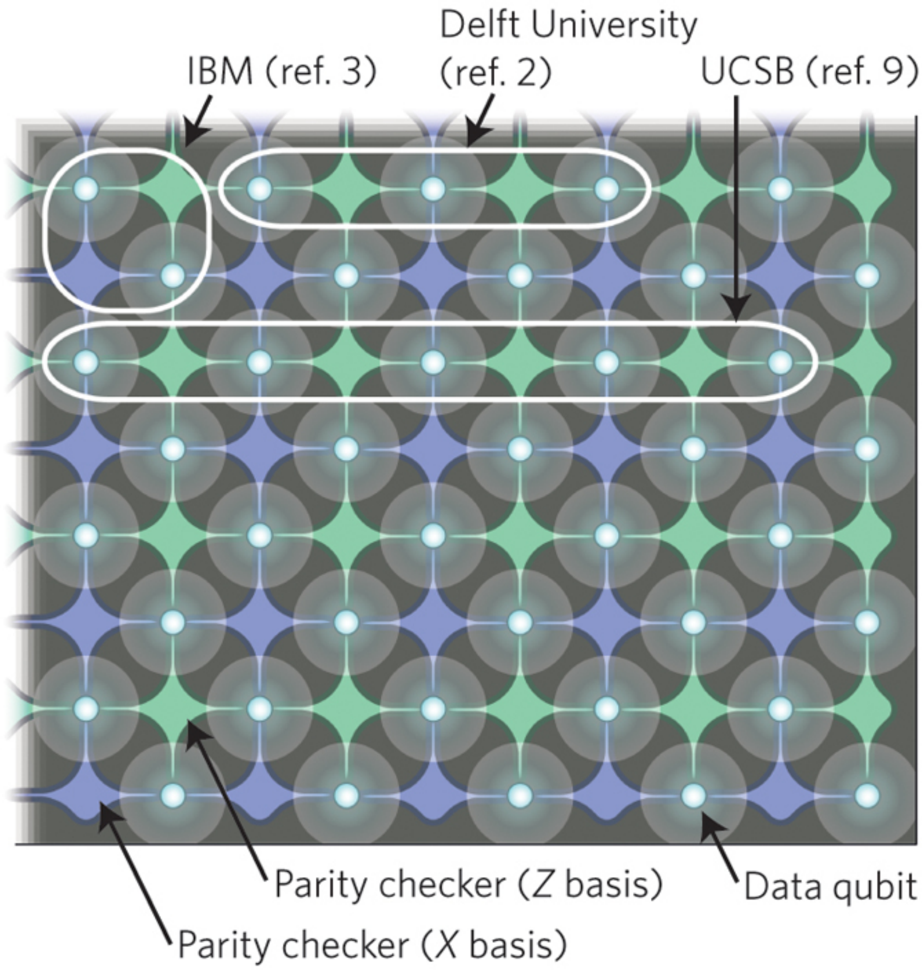
5.1 Toric Code

"Surface code" is the name of a specific variety of quantum error correction. The idea traces its origins back to the concept of Toric codes, invented by Alexei Kitaev. As their name implies, Toric codes involve a torus, at least in one interpretation. They can be more easily understood as a square lattice, where the vertices and faces of the lattice each correspond to their own stabilizer codes, $A_v = \Pi_{i \in v}\sigma_i^x$ and $B_p = \Pi_{i \in p}\sigma_i^z$. These stabilizer codes do not affect the qubits in the base state, for example, $A_v|\psi\rangle = |\psi\rangle$ in the defined stabilizer space for both codes. Therefore, if the above isn't true, then we know that an error has been introduced. Additionally, because

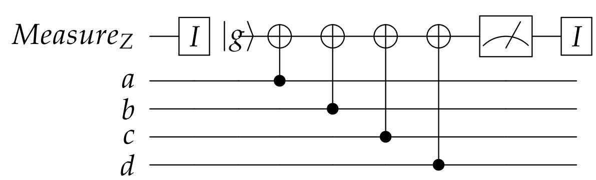
5.2 Surface Code

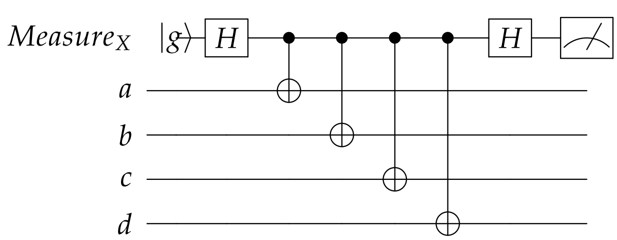
Surface code is notable in that it's very tolerant of local errors, relative to other methods of quantum error correction. However, this comes at the cost of using a great deal of physical qubits just to represent a single logical qubit. In fact, it requires a minimum of thirteen physical qubits to represent a logical one, and if one wants a logical qubit that's reasonably error tolerant, then one should be ready to dedicate hundreds or thousands of physical qubits to representing it. The ultimate number depends on the error rate, though. For example, if the error rate is improved by a factor of 10, then the number of physical qubits needed is reduced by an order of magnitude.

The actual code itself involves two kinds of qubits differentiated by how we use them; data qubits that store computational quantum states, and measurement qubits that affect and stabilize the data qubits. Functionally, qubit errors come in two flavors, bit-flip errors and phase-flip errors. Therefore, there are two kinds of measurement qubits as well, measure-X qubits that help with bit-flip errors, and measure-Z qubits that do the same for phase-flips. Essentially, when a data qubit develops an error, trying to measure it comes with the issue of destroying the information it contains. However, measure-X and measure-Z qubits fix this.



Above is a 2-dimensional representation of a surface code system. [5] Note how, here, every logical qubit is flanked by two measure-X and two measure-Z qubits, here called parity checkers on the X or Z basis. While actual systems can be far more complex, this diagram is useful for picturing what's going on while it runs.





Essentially, the system goes through surface code cycles where the measurement qubits run through CNOT gates in a very specific fashion (see the diagrams above) with the logical qubits around them and are measured, arriving at a total combined state $|\psi\rangle$. Remarkably, in subsequent cycles, due to the nature of stabilizer codes, if none of the logical qubits suffer an error, the state will remain $|\psi\rangle$. However, if a particular qubit q undergoes an error, such as a phase flip (and is not automatically fixed by a surface code cycle, which is statistically likely), it will change the measurement outcomes of the two measure-X qubits flanking q, causing a difference in $|\psi\rangle$ while allowing us to also know what logical qubit is responsible, so we can manually phase-flip it to correct the error without even needing to directly measure it.

The above circuits also bear several quirks that deserve mention. The stablizer code properties of surface code are quite fragile, breaking without proper care and attention. For example, note how the circuit involving a measure-z qubit has two identity gates. This is because, in order for the stabliter property to hold, both measure-x and measure-z qubits must act in lockstep with each other, running at the exact same time. However, because the circuit for measure-x qubits runs it through two Hadamard gates, it would be out of sync with the circuit for measure-z qubits without additional identity gates (note that, while this is sufficient for a theoretical understanding, there my still be a physical issue here if the physical implementation of an identity gate does not take as long as the implementation of the Hadamard gate). Additionally, the qubits a, b, c, and d listed in each circuit are processed in a zig-zag pattern. a is above the measurement qubit, b is to it's left, c to it's right, and d is below it. Again, this is specifically because following a different pattern would break the stablizer property.

One might also wonder, since the code notes any difference from the base state as an error, how we can affect the contents of the data cubits without the surface code undoing our actions. This can be achieved by changing the base state. We are also able to utilize these measure qubits to implement regular operations on the logical qubits without affecting the stabelizer code properties that allow the system to find and correct errors. However, due to the complexity of these operations, their summery will not be included here. Suffice to say, much like other QECC's, implementation of logical gates work on the code as a whole, measure qubits included. [13]

In all, even considering how it's complexity would require extra considerations for physical implementation and the large number of physical qubits required if the error rate it high, surface code still remains one of the most promising QECC's for an actual machine today. The gains of nearly automatic stabilization, even in the face of common errors, are too great to ignore.

6 Physical Implementations of QECC

It can become easy to get lost in all the theory and advanced mathematics involved in QECC, so much so that one can lose track of the goal. Physical realizations of QECC in quantum circuits have been made and documented, with decently improving results.

Early experimental error correction used error correcting procedures realized in trichloroethylene that implemented the three-qubit code in a NMR setting [11] [12]. In 2011, improving on the methods and techniques used earlier to improve the fidelity, researchers were able to increase fidelity by 10% (> 95 % at 0 seconds after computations) [21]. In 2012, a three-qubit system implementing the CNOT gate was able to, with 85 ± 1 percent fidelity to the expected classical results, and 75 ± 1 percent fidelity to the expected quantum results [17].

At some point, quantum computing will be reliable enough, via error-correcting codes and other methods, to produce results with high fidelity. High enough fidelity to even be used commercially, as some companies are expecting [3] [2] as either a per-customer based solution, or a centralized quantum mainframe solution.

References

[1] Astrology.

[2] Quantum Computing - IBM Q - US.

[3] Qci - quantum circuits computer, 2016. http://quantumcircuits.com/.

[4] Dave Bacon. Operator quantum error-correcting subsystems for self-correcting quantum memories. *Phys. Rev. A*, 73:012340, Jan 2006.

[5] Simon Benjamin and Julian Kelly. Superconducting qubits: Solving a wonderful problem. *Nature materials*, 14(6):561–563, 2015.

[6] Charles H. Bennett, David P. DiVincenzo, John A. Smolin, and William K. Wootters. Mixed state entanglement and quantum error correction. 1996.

[7] A. R. Calderbank and Peter W. Shor. Good quantum error-correcting codes exist. *Phys. Rev. A*, 54:1098–1105, Aug 1996.

[8] Rui Chao and Ben W Reichardt. Fault-tolerant quantum computation with few qubits. *arXiv preprint arXiv:1705.05365*, 2017.

[9] Man-Duen Choi, David W. Kribs, and Karol Zyczkowski. Higher-rank numerical ranges and compression problems. 2005.

[10] Man-Duen Choi, David W. Kribs, and Karol Zyczkowski. Quantum error correcting codes from the compression formalism. 2005.

[11] D. G. Cory, A. F. Fahmy, and T. F. Havel. Ensemble quantum computing by NMR spectroscopy. *Proceedings of the National Academy of Sciences*, 94(5):1634–1639, mar 1997.

[12] D. G. Cory, M. D. Price, W. Maas, E. Knill, R. Laflamme, W. H. Zurek, T. F. Havel, and S. S. Somaroo. Experimental quantum error correction. *Phys. Rev. Lett.*, 81:2152–2155, Sep 1998.

[13] Austin G. Fowler, Matteo Mariantoni, John M. Martinis, and Andrew N. Cleland. Surface codes: Towards practical large-scale quantum computation. *Phys. Rev. A*, 86:032324, Sep 2012.

[14] Daniel Gottesman. Stabilizer codes and quantum error correction. *arXiv preprint quant-ph/9705052*, 1997.

[15] Emanuel Knill and Raymond Laflamme. Theory of quantum error-correcting codes. *Phys. Rev. A*, 55:900–911, Feb 1997.

[16] John Preskill. Quantum error correction, 1999. http://www.theory.caltech.edu/people/preskill/ph229/notes/chap7.pdf.

[17] M. D. Reed, L. DiCarlo, S. E. Nigg, L. Sun, L. Frunzio, S. M. Girvin, and R. J. Schoelkopf. Realization of three-qubit quantum error correction with superconducting circuits. *Nature*, 482(7385):382–385, Feb 2012.

[18] Claude E Shannon. W. weaver the mathematical theory of communication. *Urbana: University of Illinois Press*, 29, 1949.

[19] Peter W. Shor. Scheme for reducing decoherence in quantum computer memory. *Phys. Rev. A*, 52:R2493–R2496, Oct 1995.

[20] Peter W. Shor and John A. Smolin. Quantum error-correcting codes need not completely reveal the error syndrome, 1996.

[21] Jingfu Zhang, Dorian Gangloff, Osama Moussa, and Raymond Laflamme. Experimental quantum error correction with high fidelity. 2011.