A Brief Introduction to Quantum Error Correction

Samuel James Bader
MIT Department of Physics, (4-304) 77 Massachusetts Ave., Cambridge, MA 02139
(Dated: May 4, 2012)

This paper introduces quantum noise, surveys its appearance in many implementations of quantum computing systems, discusses error correction codes for managing it (a three-bit classical code, a three-qubit quantum code, and the nine-qubit Shor code), and closes with a glimpse of fault-tolerant computing and the threshold theorem.

I. INTRODUCTION

Advances in quantum information science over the last decades make ambitious promises ranging from vastly more efficient algorithms to solve intractable problems (such as large integer factorization), cryptographic systems guaranteed secure by physics itself, and a deeper understanding of quantum mechanics with the experimental ability to precisely manipulate the quantum world.

The advantages of quantum computation arise from the quantum mechanical properties of the *qubits* (analogous to classical bits) upon which such computing models are based, namely superposition and entanglement.

For instance, a classical bit can take one of two discrete values (conventionally labelled as "0" and "1") and a string of n classical bits can take on one of 2^n values. Meanwhile, a qubit may take on any superposition in some two-dimensional basis, $\{|0\rangle, |1\rangle\}$, and a string of n qubits may take on any superposition in a 2^n-dimensional basis of product states. Thus, any classical state of n bits may be specified by n 0's and 1's, but a state of n qubits must be specified by 2^n coefficients [12]. So the information stored in the state of n bits grows linearly with n, and the informations stored in the state vector of the n qubits grows exponentially with n. *To double the memory of a quantum computer, add one more qubit.*

Furthermore, linear operations on quantum systems act separately on each element of a superposition, which enables the parallel execution of massive calculations. Clever quantum algorithms (such as the famous Shor's algorithm) exploit these techniques in imaginative ways to solve problems which cannot be managed efficiently in classical models.

However, one difficulty such systems face is the fragility of qubits–the small systems in which the quantum mechanical regime applies may be quite sensitive to noise, decoherence, and perturbation from the environment, which engineering and experimental brilliance can only limit so much. This paper will address the appearance of noise both abstractly and in experimental realizations, and the computational methods for coping with its disturbing influence.

II. QUANTUM NOISE

Quantum errors–unwanted random changes in the qubit state–are caused by both imperfections of the system performing the computation and the undesired interaction of the qubits with the environment. So that the reader may have examples in mind, we shall examine several concrete realizations of quantum computing and their associated errors before discussing correction schemes.

A. Optical Quantum Computing

Optical computers, using photons for their qubits, likely provide the most noise-free implementation[1], since photons propagating in free space or optical fibers do not couple strongly to each other or the environment. The qubit can be stored in the polarization state, the possible paths of a beam-splitted photon, or even with time-binning (where an interferometer with uneven arms separates the pulse into two components and recombines them such that one component is delayed in propagation.)

Subjecting different components of the qubit to different indices of refraction is a simple way to adjust the relative phases (eg. for a polarization qubit, one could use an anisotropic material, which has different indices for different polarization axes, to provide this arrangement.) By optical methods such as these, one can achieve any arbitrary single-qubit unitary transformation.

The difficulty is that multiple-qubit operations require the passage of one photon to affect the other. Since photons do not interact directly, such an effect must usually be mediated by matter. Non-linear materials, cavity QED, or even clever (non-deterministic) linear techniques [1] allow for this. Still, imperfections in these components do allow for error (eg, the polarization of a photon may rotate along an optical cable, and whenever the photon passes through materials, there is the risk of absorption.)

B. Ion traps

One could also use the electronic states of atoms as qubits. Generally, electromagnetic traps are configured to hold individual ions or strings of ions in place.

Lasers can then address transitions between the various atomic and motional states of the individual ions, and the coupling of the motional modes of the ions (via their Coulomb interaction) allows one to entangle qubits and perform multi-qubit operations.

For example, the Quanta group at MIT [2] demonstrated a controlled-NOT gate (a multi-qubit operation which will be important to us in upcoming sections) using the transitions of $^{88}Sr^{+}$ ions suspended by two-dimensional surface electrodes. In order to keep the ions well-localized in the trap and use the low motional states, the system is kept in vacuum and cooled to 4 Kelvin.

These states can be trapped for lifetimes on the order of hours, but the referenced paper lists sources of errors such as off-resonant excitations, and minor fluctuations in the frequency and intensity of the laser [2]. Prevention of the heating of the motional states (eg due to surface charge fluctations in the trap) is another important challenge in coherently coupling the qubits [3].

Also widely used are traps with three-dimensional geometries (eg. radio frequency Paul trap), such the one used in demonstration by Rowe et. al [4] of coherent transportation of the qubits, which is one route to implementing an interconnected network of trap computers. This paper also cites imperfect laser pulses in state preparation and detection as their main error sources.

C. NMR Computing

Nuclear Magnetic Resonance computers store the qubits in nuclear spin states, and manipulate them via a rotating magnetic field pulse. Because nuclear magnetic moments are so small, the system must contain many (at least 10^8) nuclei for detectable signals. But fortunately the weakness of nuclear magnetic moments also means they couple weakly, and thus are quite robust [8].

NMR is typically done with molecules, and each atom in such a molecule may have different spin-splittings–even if both atoms are of the same element, the difference can be due to the perturbations from the electronic structure of the molecule. Thus, pulses at different frequencies can address separately the different atoms (the qubits) of a molecule. The readout from measurement of any qubit is an ensemble average over all the molecules. The effective spin-spin coupling of the nuclei mediated by the electrons can also be used to entangle the qubits.

NMR computing is distinct from the other two systems in that it operates on a bulk system which is (initially) in thermal equilibrium (rather than, for instance, a pure state of qubits initialized to $|0\rangle$), and consequently involves many interesting techniques, outside of our scope, for applying typical quantum computations [8].

Decoherence (which is in this case, the many spins representing a qubit falling out of phase with each other) may arise from inhomogeneities in the magnetic field–this effect, however, is actually reversible (via a technique known as "spin echo"). However, spin-spin couplings and the thermalization of the the spins do represent irreversible errors[8].

Having discussed quantum noise in generality and seen its realization in a couple implementations, we are now ready to discuss how to cope with it, almost.

III. A DIGRESSION: CLASSICAL ERROR CORRECTION

Before we dive into correcting quantum errors, it would suit the reader to have some grounding in classical error correction. In this section, we will also introduce circuit logic diagrams, which will generalize to the quantum case, to visually represent our computations.

A. Circuits

Lines will indicate the flow of bits through logic. A bit string, such as "010" enters on the left, each bit getting its own line (order top-to-bottom). The simplest diagram is the identity circuit:

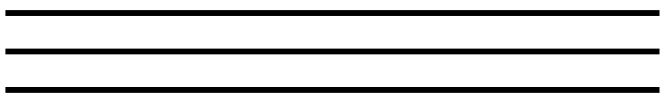

FIG. 1: In a logic diagram, the bits flow from left to right. Here, three bits are acted on by the identity operation.

Let us add a useful component to our inventory. The controlled-not (CNOT) gate, acts on two bits, a "target" and a "control." CNOT flips the target bit if the control bit is 1, and it does nothing at all if the control bit is 0. We will represent it on circuit diagrams as:

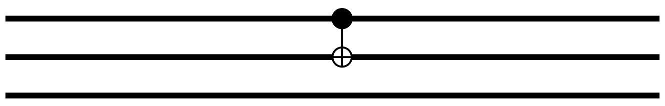

FIG. 2: CNOT. The dark circle marks the control bit, and the + circle marks the target bit. Any other bits (eg the third bit) are unaffected.

The reader should check his understanding by determining that if the three bits begin (on the left) as 110, then they exit on the right as 100.

Those familiar with classical logic gates will recognize this action as a classic XOR (which takes two inputs and outputs one bit indicating whether they are the same), except that the XOR gate then throws away the control bit (because classical logic gates conventionally give a single output).

We will also find the controlled-controlled-not gate (aka Toffoli gate) useful. This gate is the like CNOT but requires *two* control bits to both supply 1's in order to flip a target bit. An example is given below (left of diagram), along with a handy modified version (right of diagram).

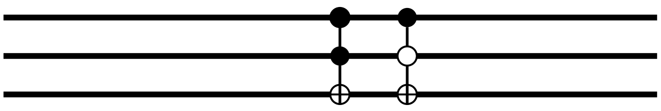

FIG. 3: Controlled-controlled-not. Flips the target bit conditioned on two control bits supplying correct values. For convenience, we use filled circles to require 1's, and hollow circles to require 0's. This syntactic sugar just saves us from having to include a bunch of NOTs in the circuit.

B. Errors

In the simplest and most illuminating model [9], errors will be treated as a blackbox operation which, with some probability p, flips any given bit (independently). Further, as both abstraction and as a first-attempt simplification, we will assume that we can implement the error correction codes with perfect, reliable logic gates. Thus we localize the possible errors outside of our error correction mechanism (we can imagine, perhaps, that we are attempting noisy transmission between two reliable circuits).

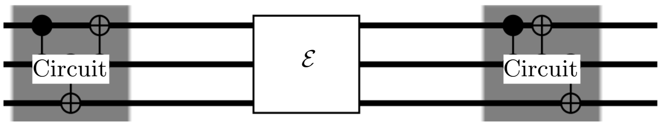

FIG. 4: Errors are represented by a single operation which, for each bit independently, flips it with probability p.

Suppose we are trying to protect a single bit. The most straightforward approach would be to replicate the bit, and send three copies of the bit through the circuit. That way, whenever we find that one bit has been flipped, we can take a majority vote of the three bits to correct the error.

Of course our result will be incorrect if the noise flips two bits, but this code is certainly an improvement. In the original set-up of trying to send one bit through a noisy element, we had a probability p of error. In this error correction code, we have a probability

$$P(\text{flip more than one bit}) = 3p^2(1-p) + p^3 = 2p^3 + 3p^2$$

of error. We have eliminated the error to first-order, and our result is an improvement so long as $p < \frac{1}{2}$. Obviously, adding more bits would create an even more robust code.

Let us try to implement this code (using our logic circuits) to correct *single-bit flip* errors. In general, there are three parts. (1) Encoding: store the bits with some form of redundancy. (2) Syndrome Measurement: determine if there was an error. (3) Recovery: correct the error.

1. Encoding

This step can be done with two CNOTs. Assuming the top bit is the one we are trying to send, and we have two other bits (hereby referred to as ancilla bits) *initialized to 0*, the following will do the trick.

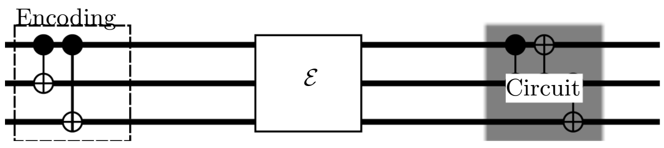

FIG. 5: Assuming the ancilla are initialized to zero, the encoding stage above will set all three bits equal to one another.

2. Syndrome Measurement

We will measure the encoded bits and store our results in two additional ancilla initialized to 0's. Suppose we measure the parity [13] of the 1-2 pair and the parity of the 1-3 pair (numbered from the top). If all the parities are even, then we know there has been no error (remember that we are assuming at most *single* bit-flips). If exactly one of those parities is odd, then we know that bit #2 or #3 respectively is in error. If both results are odd, we know that #1 is in error.

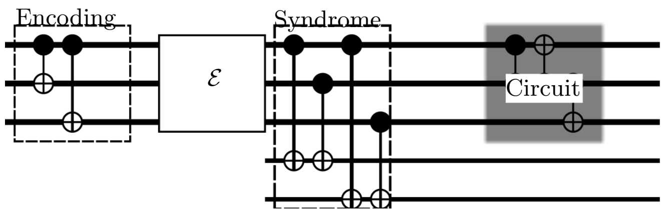

FIG. 6: The first additional ancilla will now store the parity of the 1-2 pair, and the second will store the parity of the 1-3 pair.

3. Recovery

The method for recovery uses the measured parities to signal a trio of controlled-controlled not gates to correct the erred bit.

After these three stages, we expect, assuming the noise did not flip more than one bit, to have a corrected state of encoded bits. This is the Classical Three-Bit Code [9].

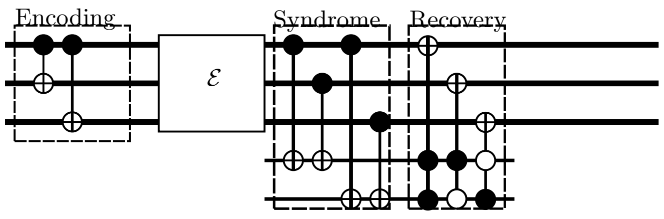

FIG. 7: Use the parities to correct the encoded bits. After doing so, we dispose of the additional ancilla.

IV. QUANTUM ERROR CORRECTION

A. The possible errors

In classical computation, the only error which could happen to a bit during some interaction is that its discrete value is unintentionally flipped.

In the quantum case, the coefficients on the $|0\rangle$ and $|1\rangle$ states can change in both magnitude ("bit flip") and relative phase ("phase flip"). And both of these potentially disturbed parameters are continuous values. It seems that our codes for error correction will thus have to monitor two effects of continuous errors! How can we generalize our classical circuit to handle this?

B. Quantum circuit diagrams

We have already discussed several implementations of quantum computing, but now we shall abstract away such detail. Broadly speaking, an operation can be done on a set of qubits by exposing them to some Hamiltonian, which then leads to a unitary time evolution. Each "circuit element" then can be described by a unitary matrix acting on the state vector of the qubits. Now let us describe the action of the quantum logic gates, so that we can try to translate our error correction code.

The vital task of flipping a single qubit ($|0\rangle \to |1\rangle$, $|1\rangle \to |0\rangle$) will be handled by the Pauli X matrix:

$$X = |1\rangle\langle 0| + |0\rangle\langle 1|$$

With this notation, the CNOT gate (taking the first qubit as the control), applies the identity I or the Pauli X to the second qubit, conditioned on the first qubit.

$$CNOT = |0\rangle_1 \langle 0|_1 \otimes I_2 + |1\rangle_1 \langle 1|_1 \otimes X_2$$

Similarly, the Toffoli gate could be written

$$TOFFOLI = \{1 - |11\rangle_{12} \langle 11|_{12}\} \otimes I_3 + |11\rangle_{12} \langle 11|_{12} \otimes X_3$$

For continuity, we shall continue to use the same notation in our diagrams for the quantum CNOT and quantum Toffoli gates as their classical analogues, so that the full Classical Three-Bit Code diagram shown at the end of the previous section is a valid quantum circuit. The question still remains, however, is it a valid quantum error code?

C. Should quantum error correction be possible?

Since we have determined, in our discussion of the classical case, that redundancy can be useful for conveying information, it seems prudent to repeat once more the stages of error correction: encoding, syndrome measurement, recovery. Let us address some concerns with the quantum implementation of these.

1. Encoding

For this stage, we just store the qubits with repetition so that so that, if one bit is flipped, we can recover the information. Simple enough. *However*, the famous no-cloning theorem of quantum computation forbids processes which duplicate arbitrary quantum states. The proof is so simple, we include it below:

Proof. Suppose, by manner of contradiction, that some unitary operation copies any arbitrary unknown state $|\psi\rangle$ onto another qubit, which, without loss of generality, begins in state $|0\rangle$.

$$|\psi\rangle |0\rangle \to |\psi\rangle |\psi\rangle$$

Let us try to copy two arbitrary states $|\phi_1\rangle$ and $|\phi_2\rangle$. Since unitary operators unitary operators preserve inner products, we can equate the inner products of these two system states before and after copying:

$$\begin{aligned} \langle\phi_1| \langle 0| |\phi_2\rangle |0\rangle &= \langle\phi_1| \langle\phi_1| |\phi_2\rangle |\phi_2\rangle \\ \langle\phi_1|\phi_2\rangle &= \langle\phi_1|\phi_2\rangle^2 \end{aligned}$$

But we quickly arrive at a statement which is certainly not true for arbitrary $|\phi_1\rangle$ and $|\phi_2\rangle$. So such a process, which duplicates arbitrary states cannot exist. □

This does not bode well for our hopes to use a repetition encoding.

2. Syndrome Measurement

For this stage, we simply measure whether an error has occurred. Unfortunately, quantum measurement collapses superpositions, destroys the information, and cannot exactly determine the state of the qubits. And furthermore, the possible errors are continuous, as discussed earlier.

3. Recovery

Perhaps we could still recover the encoded qubits, except that we probably destroyed all our information when applying that syndrome measurement (on those encodings which cannot exist in the first place).

D. The Magic

Despite all of the above objections, we will find that the circuit below can protect an arbitrary qubit from bit-flips (again assuming no more than one qubit gets flipped by the noise). [9]

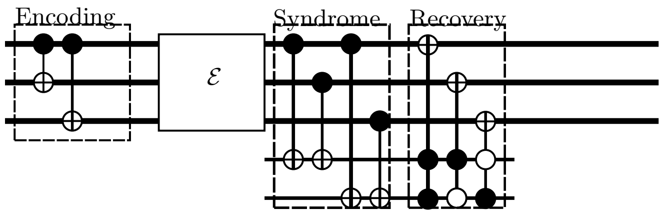

FIG. 8: A three-qubit bit-flip code. As in the classical case, ancilla are initialized to $|0\rangle$.

The astute reader may recognize this as the exact same diagram from the classical code, the one which we just argued should be impossible to implement quantum mechanically. To reconcile this, we shall certainly have to reinterpret our method. What is each stage actually accomplishing?

1. Encoding

If the first qubit begins in the state $\alpha|0\rangle + \beta|1\rangle$, and the ancilla begin in $|0\rangle$, then, using the above discussion of the quantum CNOT gate, the encoding stage will take

$$\{\alpha|0\rangle + \beta|1\rangle\} \otimes |0\rangle \otimes |0\rangle = \alpha|000\rangle + \beta|100\rangle$$
$$\downarrow$$
$$\alpha|000\rangle + \beta|111\rangle$$

Notably, this does *not* violate the no-cloning theorem, because we have *not* made three copies of a qubit: we have created an entangled state of three qubits which contains the information previous held in a single bit. Here is the difference between the entanglement and the copying, spelled out in mathematically:

$$\alpha|000\rangle + \beta|111\rangle$$
$$\neq$$
$$\{\alpha|0\rangle + \beta|1\rangle\} \otimes \{\alpha|0\rangle + \beta|1\rangle\} \otimes \{\alpha|0\rangle + \beta|1\rangle\}$$

Because the three are entangled, we cannot describe each bit separately (they are not *pure states*). In other words, it is *not* the case that each bit independently contains the information, for instance, if we were to measure one of the bits, we would collapse *all three.*

2. Syndrome Measurement

The "measurement" entangles two more qubits into the system. As in the classical case, the first new addition reflects the 1-2 parity, and the second addition gives reflects the 1-3 parity. As noted, the error is continuous, so the noise could "partially flip" a bit. For example,

$$|000\rangle \to \alpha_2|000\rangle + \beta_2|100\rangle$$

But, since our measurement is quantum mechanical as well, that does not pose a problem. Our five-qubit system (including the new ancillae) will simply take on the syndrome superposition corresponding to the error superposition:

$$\{\alpha_2|000\rangle + \beta_2|100\rangle\} \otimes |00\rangle \to \alpha_2|00000\rangle + \beta_2|10011\rangle$$

Observe that the additional qubits in which we display the value of the measurement do not actually determine the value of the encoded qubit we are trying to protect. In fact, they only mark, for each pure state of the system, the information of which error has occurred (the syndrome). The importance of this will soon be clear.

3. Recovery

Now we apply various Toffoli gates, as in the classical circuit, to correct the encoded qubits for each possible error. Continuing the example from the syndrome section

$$\alpha_2|00000\rangle + \beta_2|10011\rangle$$
$$\downarrow$$
$$\alpha_2|00000\rangle + \beta_2|00011\rangle$$

We have recovered the original encoded qubits, and, as we can see, the encoded bits are not actually entangled with the ancilla anymore:

$$\alpha_2|00000\rangle + \beta_2|00011\rangle = |000\rangle \otimes \{\alpha_2|00\rangle + \beta_2|11\rangle\}$$

So if we "dispose" of these ancilla now, (ie the computer resets them both back to $|0\rangle$ by some environmental interaction, for later usage), then they will not collapse/measure the qubit in any way.

Voilà! We have corrected any arbitrary bit-flipping error on a single quantum bit [9].

E. Phase error

However, there we still haven't corrected any errors in the relative phase between the $|0\rangle$ and $|1\rangle$ state. None of the circuit elements we've dealt with care about the phase, so we'll have to introduce something new.

Another common gate in quantum computation is the Hadamard gate.

$$\begin{aligned} |0\rangle &\to \tfrac{1}{\sqrt{2}}(|0\rangle + |1\rangle) \equiv |+\rangle \\ |1\rangle &\to \tfrac{1}{\sqrt{2}}(|0\rangle - |1\rangle) \equiv |-\rangle \end{aligned}$$

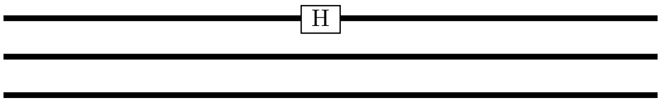

FIG. 9: The Hadamard gate.

We notice that if the noise changes the relative phase by π between $|0\rangle$ and $|1\rangle$, than it converts $|+\rangle$ to $|-\rangle$. So encoding in the $\{|+\rangle, |-\rangle\}$ basis turns the phase noise into a $\{|+\rangle, |-\rangle\}$-bit-flip noise.

So to protect from phase noise instead of bit-flip noise, we simply switch our encoding scheme by applying the Hadamard gate to each qubit:

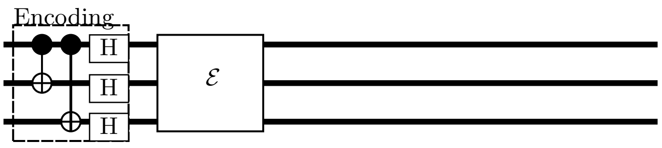

FIG. 10: An encoding to protect from phase errors.

Explicitly, the encoding is then

$$\{\alpha |0\rangle + \beta |1\rangle\} \otimes |0\rangle \otimes |0\rangle = \alpha |000\rangle + \beta |100\rangle$$
$$\downarrow$$
$$\alpha |{+}{+}{+}\rangle + \beta |{-}{-}{-}\rangle$$

And we can simply reuse our previous Syndrome Measurement and Recovery codes if we conjugate them by the Hadamard operator into this basis [9].

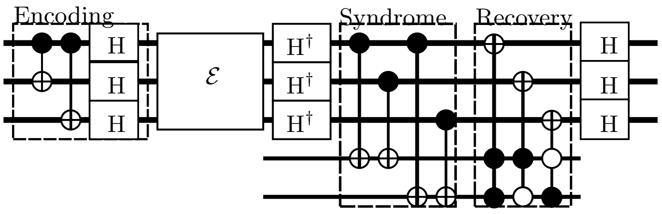

FIG. 11: A three-qubit phase-flip code.

F. Now we're cooking

We can *concatenate* these two encodings to form the Nine-qubit Shor Code. This code can protect a single qubit from any single bit-flip and/or single phase flip type error. We first do a three-qubit phase encoding, and then encode each of those three qubits with a three-qubit bit-flip encoding. Sparing the arithmetic, the end result is

$$|0\rangle \to \frac{1}{2\sqrt{2}}(|000\rangle + |111\rangle)(|000\rangle + |111\rangle)(|000\rangle + |111\rangle)$$
$$|1\rangle \to \frac{1}{2\sqrt{2}}(|000\rangle - |111\rangle)(|000\rangle - |111\rangle)(|000\rangle - |111\rangle)$$

A moment's consideration will show any single bit flip and/or single phase flip on the encoding of $|0\rangle$ will create some other state orthogonal to it, but clearly distinguishable from any state which such errors could make from the encoding of $|1\rangle$ (and vice versa). Thus, a the syndrome measurement can distinguish which is the original state of any erred state. The error can be corrected [9].

Using a classical code, superposition, and concatentation, we have generated a nine-qubit quantum error correction scheme. In fact, with a rumination in group theory, one can even come up with codes requiring fewer qubits, the lower bound for this level of protection being five qubits [9].

V. EXPERIMENTAL REALIZATION

Quantum error correction has been realized in a variety of implementations, (eg a three-qubit code on trapped ions [5], and on superconducting circuits [6], and the remarked five-qubit code in NMR [7]).

It will be revealing to examine some results from that superconducting realization (Figure 12).

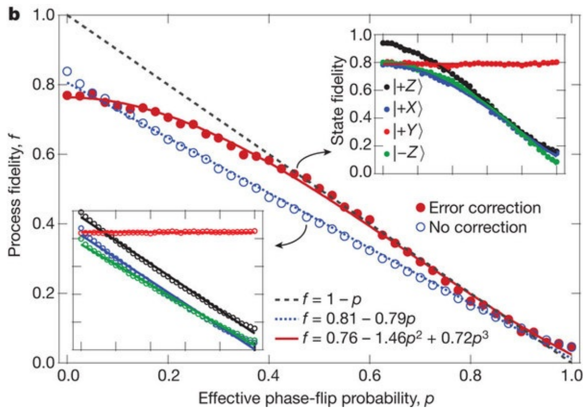

FIG. 12: Experimental performance of a quantum error correction code versus degree of induced noise. The red (filled) circles represent error corrected results, the blue (unfilled) are uncorrected results, but subjected to a similar series of operations, and the black line is the simulated case of an uncorrected qubit, with all other parts of the circuit and experiment idealized [6]. Fidelity is a measure of how well the state of the qubit is preserved. Fitted expression for the curves are given. Also, ignore the subplots.

There are two observations to take away from this plot. First, the quantum gates themselves are not perfect; we can see that, even when the induced error is zero, there is some loss of coherence. In fact, for small probabilities of error (leftmost region of plot), the mere addition of the gates to implement this correction scheme does more damage than good.

However, we do see the hallmark of our correction scheme in the fitted expressions (in the bottom of the plot). Although the corrected results are not an impressive improvement, they contain no terms of first-order in

p, [14] This corresponds to eliminating the single-qubit errors as we predicted (see the discussion at the end of the classical case).

VI. THE BIG PICTURE

Quantum error correction is obviously vital to the practical implementation of even the smallest quantum computers. But to be effective, error correction fits into a larger scheme of fault-tolerant computing, because, as we noticed in the previous section, attaching error correction codes onto an algorithm may make matters worse since this adds more fallible gates to circuit.

The rough idea of fault-tolerant computing is to carry out all operations on encoded bits, and carefully design gates and codes such that single correctable errors do not propagate to become uncorrectable errors among multiple qubits [10]. Then applying error-correction in-between fault-tolerant subsystems should prevent the propagation of errors throughout the computation.

Concatenation, which we used to derive the nine-bit code, and fault-tolerant operations constitute a dream-team. There is in fact, a divine theorem which, given any quantum circuit and given quantum gates which operate below a certain threshold level of error, guarantees one can construct an equivalent circuit with any arbitrarily small error rate using a polynomial-scaling number of gates [9]. The magnificence of this theorem is that it implies that quantum noise is not a fundamental barrier to the future of quantum computation. So let us rest easy, knowing that our qubits are safe.

[1] Jeremy L. O'Brien, Science **318**, 5856, (2007)
[2] Wang et al. Phys. Rev. A **81**, 062332 (2010)
[3] Labaziewicz et al. Phys. Rev. Lett. 100, 013001
[4] Quantum Information and Computation, Vol 2 (2002)
[5] Chiaverini et al. Nature **432**, 602-605 (2004)
[6] Reed et al. Nature **482**, 382385 (2012)
[7] Knill et al. Phys. Rev. Lett. **86**, 58115814 (2001)
[8] Chuang and Nielson, *Quantum Computation and Quantum Information*, (Cambridge University Press, Cambridge, 2000), Ch. 10.
[9] Kaye, Laflamme, and Mosca, *An Introduction to Quantum Computing* (Oxford University Press, New York, 2010), Ch. 10.
[10] Preskill arXiv:quant-ph/9712048v1
[11] Feynman, *The Feynman Lectures on Physics* (Addison-Wesley, Boston, 1964), Vol 3.
[12] Actually there is slightly less information, because of the constraint of normalization and arbitrary global phase, but those constraints do not scale with n.
[13] Parity is defined as even (0) if two bits agree, odd (1) if two bits disagree.
[14] This paper mentions that, if a linear term is allowed in this fit, its coefficient would be about .03, which is minuscule compared to the other dependencies.