LA-UR 96-2199

Los Alamos National Laboratory is operated by the University of California for the United States Department of Energy under contract W-7405-ENG-36

TITLE: **CONCATENATED CODES FOR FAULT TOLERANT QUANTUM COMPUTING**

AUTHOR(S): E. Knill, R. Laflamme, Wojciech Zurek

SUBMITTED TO: External Distribution - Hard Copy

DISCLAIMER

Los Alamos

Los Alamos National Laboratory
Los Alamos New Mexico 87545

Concatenated Codes for Fault Tolerant Quantum Computing

Emanuel Knill[1]*, Raymond Laflamme[2] †, Wojciech Zurek[2‡]
[1] CIC-3, MS B265, [2] T-6, MS B288
Los Alamos National Laboratory, NM 87545, USA.

May 1995

Abstract

The application of concatenated codes to fault tolerant quantum computing is discussed. We have previously shown that for quantum memories and quantum communication, a state can be transmitted with error ϵ provided each gate has error at most $c\epsilon$. We show how this can be used with Shor's fault tolerant operations to reduce the accuracy requirements when maintaining states not currently participating in the computation. Viewing Shor's fault tolerant operations as a method for reducing the error of operations, we give a concatenated implementation which promises to propagate the reduction hierarchically. This has the potential of reducing the accuracy requirements in long computations.

1 Introduction

Three recent events are promising to make extensive quantum computations as feasible as their classical equivalents. The first is the discovery by Shor [14], Steane [16] and Calderbank et al. [4, 5] of quantum error-correcting codes which can be used to maintain a quantum state for long

*email: knill@lanl.gov
†laflamme@lanl.gov
‡whz@lanl.gov

periods of time or for long distances, assuming that the requisite recovery (or error-correction) operations can be implemented arbitrarily well. The second is the application of concatenated coding techniques by Knill and Laflamme [10] to quantum communication and memories. They demonstrated that a state can be maintained for an arbitrarily long time or distance at error ϵ provided each operation is implemented with error at most $c\epsilon$ for some constant c. The third is a proposal by Shor [15] to make a computation fault tolerant provided operations can be implemented with polylog bounded error.

From a theoretical perspective, a positive solution to the following problem would be very attractive:

Problem 1.1 *Does there exist a constant δ such that for every $\epsilon > 0$, a quantum algorithm using perfect operations (see below) can be converted to an equivalent quantum algorithm with imperfect operations, each with error at most δ, such that the final error at most ϵ? The overhead of the converted algorithm should be polynomially bounded in the complexity of the original algorithm and the accuracy parameter $1/\epsilon$.*

In [15], δ depends polylogarithmically on ϵ and the complexity of the algorithm. For the purposes of this report, a quantum algorithm is a sequence of two qubit quantum operations starting with a classical initial state and ending in a measurement in the classical basis. This does not include algorithms which operate on unknown states, provided (for example) by a quantum oracle. To apply fault tolerant quantum computing methods to such problems, we need to assume that the oracle provides states already encoded in an appropriate error-correcting code.

It is worth analyzing the assumptions under which Problem 1.1 may be provable. If the error in each operation is completely arbitrary, it clearly cannot hold. Fortunately, it is reasonable to assume that operations are local.

Assumption 1.2 *Locality: If a primitive operation acts on the qubits in set X, then any error in the operation is restricted to the qubits in X.*

This assumption applies equally to quantum gates and to measurement operations. Formally, the assumption should be interpreted as follows: Let

U be a primitive one or two qubit operation to be applied to some target qubits of the state of the computer. Thus the action of U on the complete state is of the form $U \otimes I$, where we factor the systems with the target qubits first. If an error happens during the operation, the actual evolution W can be represented by an error-superoperator $\mathcal{E}$ in the form $W = \mathcal{E}(U \otimes I)$. The locality assumption requires that $\mathcal{E}$ factors as $\mathcal{E}' \otimes I$. A discussion of what it means for the error $\mathcal{E}'$ to be small is in [10]. The ensemble $\mathcal{E}'$ may be presented in the form of a mixed superposition of tensor products of the elements of the unitary error basis of Steane [16] consisting of

$$I = \frac{1}{\sqrt{2}}\begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}, \quad B = \frac{1}{\sqrt{2}}\begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix},$$
$$C = \frac{1}{\sqrt{2}}\begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}, \quad D = \frac{1}{\sqrt{2}}\begin{pmatrix} 0 & -1 \\ 1 & 0 \end{pmatrix}.$$ [1]

The total error amplitude can be estimated by the sum of the moduli of the error operations appearing in the mixed superposition. Note that the "probability" of error is related to the square error amplitude. Although when using the error basis above, the sum of the effective square amplitudes is 1, the errors are usually not orthogonal. In most cases, where worst-case estimates are desired, it is best to just work with amplitudes. Unlike probabilities (to the extent that they make sense), they propagate as expected in the worst case.

In any practical implementation of quantum computing it is very unlikely that the locality assumption is exactly satisfied. However, existing arguments in fault tolerant quantum computation are resistant to non-local errors provided that the amplitudes of non-local error operators decrease exponentially in the number of qubits involved. That this is possible is consistent with current beliefs concerning the locality of quantum mechanical evolution.

An important assumption which is intrinsic to the formalism used to describe quantum computation is the following:

Assumption 1.3 *No leakage: A qubit is truly in a tensor product with the rest of the world. That is, there is no amplitude leakage from the qubit into a direct summand.*

The assumption is not naturally satisfied by known physical systems. Photons have many additional degrees of freedom (including the no photon and multi-photon states). Ions have many levels in addition to the computational basis, some of which may be explicitly exploited for implementing operations. Enforcing the no leakage assumption does not require that the assumption holds literally. One method is to detect leakage and return the leaked amplitude to the support of the qubit whenever possible. Another is to encode the entire relevant system using non-binary quantum codes [2].

An assumption which is not critical to the formal treatment of Problem 1.1, but which may be of great practical significance is the following:

Assumption 1.4 *Parallelizability: Independent quantum gates can be applied in parallel.*

This assumption is significant because to achieve fault tolerance it is necessary to duplicate gates in parallel across independent qubits of encoded states. Parallelism is also introduced in implementing recovery operations using the methods of [15, 7]. If these operations cannot be executed in parallel, it may be much more difficult to achieve fault tolerance against both environmental interaction (which occurs at a constant rate regardless of other factors) and operational errors. In practice, it suffices to be able to execute a set of independent operations sufficiently fast compared to the background error rate.

For the experimentalist we point out that the above assumptions may ultimately drive the search for a good set of devices for implementing quantum computers. Satisfying the assumptions is not an entirely trivial problem. For Cirac and Zoller's ion-trap computer [6], none of them are satisfied. Because each two qubit operation is implemented via a bus qubit (the phonon mode), any two such operations involve a common system and are therefore necessarily dependent, unless the residual error in the phonon mode after the first operation is dissipated (a watchdog effect might be applicable, as proposed by Zurek). This is the problem of "coherent backlash". Since each operation requires access to the phonon mode, parallel implementation of independent

[2] Such codes can be naturally based on Z_n, using a generalization of the "extra special group" based on the Fourier transform and non-commuting pairs of n-dimensional representations of Z_n (unpublished).

qubit operations is unfeasible. However, a lot may already be gained by parallelizing the one-qubit gates, and this is possible in the ion-trap.

One other assumption can be made, if needed:

Assumption 1.5 *Exact classical state preparation and measurement: Preparing a classical state and performing a measurement in the classical basis can be implemented arbitrarily well.*

Currently proposed methods do not need this assumption, only that both preparation and measurement introduce only local errors. In fact, provided that there is no possibility of future interference for states that are measured, one can model inexact measurement by a composition of error operators and exact measurement. The assumption of no future interference is actually a special case of the locality assumption, which requires that unless two systems are intentionally coupled, they evolve independently.

In this report we discuss the application of concatenated codes to fault tolerant quantum computing. In Section 2 we summarize the properties of simple concatenated coding without fault tolerant recoveries. It is shown that concatenated coding can be used to simplify the construction of a quantum memory by reducing the accuracy requirements and removing the need for specially designed recovery operations. This works provided that the top level encoding can reliably correct a constant error rate. With independent errors, the latter can be achieved by concatenated codes, even though in general they do not have good minimum distance. In Section 3 we show how encoded operations distribute into a concatenated code; and that if the punctured code construction is used, measurements can be performed reliably in both the standard and the dual basis. This brings out an interesting connection between quantum codes and concatenated codes. We give a simplified implementation of primitive gates provided that the basic code satisfies that its weights are divisible by 8 and the dual has minimum distance 4. In Section 4 we propose viewing the application of fault tolerant recovery operations [15, 7] as an error reduction technique that can be applied recursively into a concatenated code. This promises to reduce the accuracy requirements from those established by Shor [15]. We leave the detailed analysis to another report. Finally, in Section 5 we obtain an explicit relationship between the error per gate and the final error in concatenated

coding without fault tolerant recoveries. The relationship is used to give an order-of-magnitude estimate of the accuracy required for gates in a quantum memory based on the 5-qubit codes of [11, 2, 3].

2 Concatenated codes

We review the concatenated quantum codes from [10]. The basic idea is to hierarchically code each qubit and interlace the procedure with recoveries in such a way that errors during recovery do not propagate as they would using simple repeated recovery operations. The technique recursively applies the coding procedures, where at the lowest level waiting periods (for communication or storage) or encoded operations (for fault tolerant computing) are applied between recovery operations. At the higher level, the recursive procedure is applied to each qubit between recovery operations. Although this allows only very limited application of encoded gates (due to indirect error propagation), this does simplify the quantum communication and memory problem.

To perform concatenated coding, we choose an error-correcting code for a qubit (or other system) of length l (i.e. using l qubits) and a repetition factor r. The repetition factor is taken as large as reasonable subject to constraints to be given below. The length of the code is largely irrelevant, what matters is how much error per qubit can be recovered at a good overall error after recovery.

The lowest level procedure ($\mathrm{CCP}_r(1)$, for Concatenated Coding Procedure of depth 1 with r repetitions) consists of simple iterated recoveries between an encoding and a decoding operation. That is, $\mathrm{CCP}_r(1)$ begins with one qubit, encodes it using the error-correcting code to l qubits, applies a recovery procedure to the code $r-1$ times and finally decodes it back to a single qubit[3]. In between recovery operations, we can either just wait for a certain time interval, transmit each qubit over some distance, or apply a few suitably encoded operations involving other encoded qubits.

The higher level procedures $\mathrm{CCP}_r(h)$ are defined recursively, using a

[3]The repetition factor is r because the final decoding operation is a special form of the recovery operation, so in effect, r recovery operations are used.

procedure like $\mathrm{CCP}_r(1)$, but with the next lower level applied to each qubit between recoveries. That is $\mathrm{CCP}_r(h+1)$ starts with one qubit, encodes it using the code, applies $\mathrm{CCP}_r(h)$ to each of the qubits of the code and recovers the code $r-1$ times, applies $\mathrm{CCP}_r(h)$ to each qubit again and finally decodes the state to one qubit.

The error-correcting properties of $\mathrm{CCP}_r(h)$ are discussed in [9]. In summary, suppose that the following holds: If each qubit of the code is subjected to independent interactions of error amplitude e_d, then the total error after recovery or decoding is at most e_c. Suppose also that $r+1 \leq e_d/e_c$. Then, if the error introduced in each qubit in $\mathrm{CCP}_r(h)$ between recovery operations is independent and bounded by e_d, the final error of the result is at most $(r+1)e_c$. An important property of this technique is that no assumptions are made on the code used. The error propagation assumptions are all strictly worst case–no classical approximation is used which assumes error is perfectly dissipated (see [10]). Also, any sufficiently high fidelity code can be used, not only e-error-correcting codes.

The total number of intervals between recoveries at the lowest level of the $\mathrm{CCP}_r(h)$ is r^h, the total number of parallel recovery operators is $O(r^h)$, and the maximum number of qubits required is l^h, where l is the length of the code. Thus, if the number of time intervals for which the state needs to be maintained is n, then the total overhead in qubits is $O(n^{1+c})$, with $1+c = \log_r l$. An explicit relationship between the error amplitude per operation and the overall error amplitude of the state will be given in Section 5.

Using concatenated codes as suggested above is very helpful for applications where each qubit to be preserved can be treated independently. However, to apply it to a quantum memory used during quantum computations requires more explanation. Preserving a qubit to within bounded error is only useful if that error does not propagate to the full state of the computation. Shor [15] has shown how to accomplish this by using fault tolerant recovery operations. Thus, if the state of the computation is already encoded in a code which can tolerate error ϵ per qubit, then one can store each of the qubits of the encoded state using a concatenated code which meets this requirement without loosing recoverability and fault tolerance of the encoded state. The code of the computation must be able to tolerate a constant error rate per qubit. Luckily, concatenated codes achieve this in

principle, provided the errors are effectively independent. Each concatenation reduces the uncorrectable error of the lower level by a power related to the error-correction capacity of the basic code, if an exact recovery operator is used.

3 Operating on states encoded in concatenated codes

A critical issue in operating directly on encoded states is to ensure that errors introduced during operations are corrected by recovery operators. In current proposals, this is achieved by ensuring that the encoded operations are transversal to the codes in the layout pattern of the qubits. Suppose that we wish to operate on m qubits and that they are encoded using a code of length l. The complete set of qubits can be placed into an array of dimensions $l \times m$, where each column consists of the qubits supporting a coded qubit[4]. If the code can correct any e errors, then the entire array is a code which can correct any type of error spanning at most e qubits in each column. Somewhat loosely speaking, if we can implement encoded operations in such a way that non-local errors never extend for more than e qubits in a column and remain at sufficiently low amplitudes, then with some bounded error, recovery operations applied to each column can restore the intended state.

The dependencies of the encoded operations are determined by the graph of the target qubits of each primitive operation that is applied. That is, connect qubits x and y if one of the operations targets both x and y. The connected components of the resulting graph represent the potential dependencies between qubits, and it is these components that we must attempt to limit in their extent in each column. If the code corrects one error, the constraint requires that each primitive operation targets only qubits within a row (in some permutation of the elements of each column). We therefore refer to this as the *transversality* constraint. It may of course be possible to relax these constraints if suitable codes are used, maybe even non-block

[4]These columns are referred to as "qubytes" by Zurek and Laflamme [17]. They can be considered as the "qublocks" of a quantum block code.

codes[5].

As an aside, let us briefly consider the possibility of weakening the transversality constraint while still ensuring that an encoded operation does not introduce uncorrectable errors at high amplitudes. If each operation is followed immediately by a recovery step, then the errors preceding the application of the encoded operation are essentially restricted to those types which can be introduced by the previous recovery operation. The encoded operation has the effect of spreading preexisting errors and (if there are dependencies) those introduced during the operation. The spreading is strongly restricted by the pattern of operations that are applied, and by the ordering of the operations. For example, consider applying controlled-nots to pairs of qubits corresponding to the edges in a path. If the desired effect requires applying the controlled-nots sequentially along the path, a bit flip error introduced by the first operation can be propagated to an equal amplitude error spread across the whole path. Other errors remain localized. If instead the desired effect can be achieved by applying the controlled-nots in two parallel steps of independent operations, then error propagation is restricted to at most three qubits at a time (for the dominant terms), and these triple errors are strongly constrained in terms of where they can occur. If this is known beforehand, the decoding procedure can be adapted to correctly decode such errors, even if not all triples can be corrected.

The set of *encodable primitive operations* determines how computations are actually implemented. An important issue is how well the desired operations can be approximated by primitive encodable ones. In order for these methods to work, the ideal computation without errors in the primitives must be sufficiently close to the desired one. According to a naive argument, the approximation by the encodable primitive operations of the constructs of a computation must be within at least $1/n$ (pessimistically) and at least $1/\sqrt{n}$ (optimistically), where n is the length of the computation. It is therefore important either to have a rich enough set of encodable primitives, or to be able to efficiently approximate the desired set of computational constructs.

We use a simpler set of primitive operations than Shor [15]. This imposes

[5]Using coding techniques other than non-block codes might be substantially more efficient, but may result in more difficult to implement encoded operations.

an additional constraint on the codes but avoids introducing complicated states to implement the Toffoli gates.

$$
\begin{aligned}
A &= \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}, \\
B &= \begin{pmatrix} 1 & 0 \\ 0 & i \end{pmatrix}, \\
C &= \begin{pmatrix} 1 & 0 \\ 0 & -i \end{pmatrix}, \\
D &= \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & i \end{pmatrix}, \\
E &= \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & -i \end{pmatrix}, \\
N &= \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix}.
\end{aligned}
$$

The operations D and E are controlled phase shifts by i and $-i$, respectively, using the standard lexicographic labeling of the classical states of two qubits. N is a controlled-not. According to Shor [15], it suffices to use A and D to obtain a sufficiently dense set of operations for quantum computation, provided a Toffoli gate can be simulated. We added the others so that the set of operations used in the encoding is identical to that being encoded. This simplifies the analysis of the implementation of the operations on the concatenated codes. To see that our set of operations is sufficiently dense we observe that D^2 is a controlled sign flip, $A^\dagger D^2 A$ (with A acting on the second qubit) is a controlled-not, the two-controlled sign flip can be implemented using two controlled-nots, two E's and one D using the well-known trick of Barenco et al. [1], and the Toffoli gate can be obtained from the two-controlled sign flip the same way as the controlled-not can be obtained from the controlled sign flip and A. The claim then follows from Shor's statement

in [15].

We do not know of any codes which easily permit transversal encoding of the complete set. However, as we will see, B, C, D, E and N as well as state preparation of $|+\rangle = |0\rangle + |1\rangle$ and measurement in the $|+\rangle, |-\rangle = |0\rangle - |1\rangle$ basis can be accomplished [6]. This suffices for computing A by using a simple version of the trick used to implement Toffoli gates in [15]. Suppose that a qubit is in state $|\psi\rangle$, and we wish to obtain $A|\psi\rangle$. We first introduce a second qubit in state $|0\rangle + |1\rangle$. Applying a controlled-not controlled by the second qubit yields the state $|\psi\rangle|0\rangle + \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} |\psi\rangle|1\rangle$. Apply B to the second qubit and change basis to obtain

$$|\psi\rangle(|+\rangle+|-\rangle)+\begin{pmatrix} 0 & i \\ i & 0 \end{pmatrix} |\psi\rangle(|+\rangle-|-\rangle) = \begin{pmatrix} 1 & i \\ i & 1 \end{pmatrix} |\psi\rangle|+\rangle+\begin{pmatrix} 1 & -i \\ -i & 1 \end{pmatrix} |\psi\rangle|-\rangle .$$

Measuring in the $|+\rangle, |-\rangle$ basis thus yields either $X = \begin{pmatrix} 1 & i \\ i & 1 \end{pmatrix}$ or $Y = \begin{pmatrix} 1 & -i \\ -i & 1 \end{pmatrix}$, and we know which one. We have $-i \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} X = Y$, and $BYB = A$.

Codes which permit transversal implementation of the desired set of operations exist. Let C be a code with weights divisible by 8 and whose dual D has minimum distance at least 4. The Reed-Muller codes RM(r, m) have this property for $m \geq 3r+1$ and $r \geq 1$ (with dual having minimum distance 4) and $r \geq 2$ (with dual having minimum distance 8) [13]. For example, RM$(1, 4)$, is a code of length 16 which works. Let C' be the punctured code, with C'_0 the even subcode. If D' is the punctured dual code, then $C'^{\perp}_0 = D'$ and $C'^{\perp}$ is the subcode D'_0 of D' derived from the vectors with a 0 in the deleted bit. We encode $|0\rangle$ and $|1\rangle$ as suggested in [16, 15]:

$$|0\rangle = \sum_{c \in C'_0} |c\rangle \, , |1\rangle = \sum_{c \in C'_1} |c\rangle \, ,$$

where C'_1 denotes the odd subset of C'. Note that if we apply the full

[6] We routinely leave out normalization factors.

Hadamard transform H to the state we obtain

$$H|+\rangle = \sum_{d \in D_0'} |d\rangle, \quad H|-\rangle = \sum_{d \in D_1'} |d\rangle,$$

where D_1' is the complement of D_0' in D'. H is of course obtained by applying A to each qubit. Thus a measurement in the $|+\rangle, |-\rangle$ basis is obtained by Hadamard transforming each qubit and measuring in the classical basis. This is equivalent to measuring each qubit in its $|+\rangle, |-\rangle$ basis. A measurement is automatically fault resistant, provided that the inferred state is deduced by using a classical syndrome decoding method which matches the one used to correct the quantum errors in both bases.

Because C's weights are divisible by 8, C''s weights are either 0 or 7mod(8), depending on whether the word is in C_0'. This immediately allows encoding B by applying C to each qubit and C by applying B to each qubit. The not operation can also be effected by applying it to each qubit. If D (or E or N) is applied individually to each corresponding pair of qubits in two encoded states, then the operation E (or D or N, respectively) is applied to the encoded state. For N this follows from the fact that C' is closed under sums. To see that it works for D and E, we show that if $x, y \in C'$, then the intersection of x and y is either 0mod(4) or 3mod(4), where the latter holds only if both are in C_1'. Let $|z|$ be the weight of a word z. Let k be the overlap between x and y. If $x, y \in C_0'$, then $|x + y| = |x| + |y| - 2k = 0\text{mod}(8)$. Since $|x| = 0\text{mod}(8)$ and $|y| = 0\text{mod}(8)$, $k = 0\text{mod}(4)$. If $x \in C_0'$ and $y \in C_1'$, then $x + y \in C_1'$, so $|x + y| = |x| + |y| - 2k = 7\text{mod}(8)$ and $|y| = 7\text{mod}(8)$ imply that $k = 0\text{mod}(4)$. If $x, y \in C_1'$, then $x + y \in C_0'$. We have $|x + y| = |x| + |y| - 2k = 6\text{mod}(8) - 2k = 0\text{mod}(8)$, so $k = 3\text{mod}(4)$.

To apply the encoded operations to the concatenated codes, each encoded operation can be re-encoded at each level of the hierarchy. In effect, the operations are applied transversally to the leaf qubits of the code, with the desired effect. It is well worth considering the construction of the concatenated code directly in terms of the theory of linear codes. As an example, we describe the punctured code construction for one concatenation. Let C_2 consist of codewords obtained by first selecting a codeword of C, then replacing each bit other than the first one with either a codeword of C_0' or C_1', depending on whether the bit is 0 or 1. As can be seen, the construction

used for quantum codes naturally appears in this context. The concatenated quantum code is obtained by puncturing C_2 at the bit that was not replaced. Note that the dual of C_2 is obtained by an identical construction: Take a codeword of D, then replace all but the bit to be punctured by a codeword of D_0' or D_1', depending on the value of the bit. As a consequence both C and D can be decoded very easily, simply by computing syndromes hierarchically.

4 Fault tolerant recoveries as error reduction

Consider the application of fault tolerant recovery operations as proposed in [15] to a specific code, which we require to correct at least one error in both bases. First we deal with the state generation problem, in our case that of generating $|0\rangle + |1\rangle = |+\rangle$. By applying the fault tolerant recovery method with one additional syndrome in the dual basis to any initial state, the final state can be forced into $|+\rangle$ with low residual error (depending on operational accuracy). This is so because $|+\rangle$ is the only state which is supported in C' in the classical basis and in D_0' in the dual basis.

In general, consider an encoded state, with some preexisting errors. The encoded state has two properties important to fault tolerant computing. The first is the fidelity of the recoverable state. Define the *recoverable state* to be the ensemble obtained in the code if a perfect recovery operator is applied. The recoverable state is in a sense the true state that is encoded, and may be in an entanglement with other systems. The recoverability of a state compared to the intended state or entanglement is the fidelity between the two. The *loss* is the associated error amplitude. Unfortunately, recoverability does not behave well under errors or encoded operations due to local error propagation. An additional property is required: fault tolerance. The fault tolerance of a state depends on the distribution of amplitudes in the various subspaces associated with different types of errors. Ideally, these amplitudes are concentrated in low error subspaces. In that case, introduction of additional errors does not immediately cause the syndrome decoding to fail.

For a fixed error-correcting code, the syndrome computations in fault tolerant recovery can be hardwired, for example by computing the whole

set three times, taking the first two or second two outcomes, if they agree, or discarding the state altogether otherwise. If the amplitude of one set of syndrome computations is p, then the amplitude of failure is at most $O(p^2)$. Let us follow the procedure of taking an incoming encoded qubit (or two), applying a requested operation in the encoding, then applying the recovery operation. This involves a finite set of operations in all circumstances. Here is the intuitive argument: If the incoming state has independent errors in the qubits or pairs of qubits of amplitude $O(p)$, then its loss is $O(p^2)$, due to the error-correcting properties of the code. Applying the operations introduces independent error of $O(p)$, the recovery operation fails with amplitude $O(p^2)$ and otherwise fixes the previous errors and introduces new independent error of $O(p)$ total amplitude. Thus the encoded operation and fault tolerant recovery seems to act with accuracy $O(p^2)$ on the encoded state. Note that the error of $O(p^2)$ introduced into the encoded qubit by operations may not be fault tolerantly encoded, but is "committed" by the next recovery operation. We will give a detailed analysis of the actual behavior in a future report. For now, we propose the following: Use the encoded operations as a more accurate version of the unencoded ones. Thus the entire procedure can be applied, instead of with qubits, with encoded qubits, implementing recovery operations and encoded operations hierarchically, just as with concatenated codes. This has the potential of amplifying operational accuracy explicitly and reducing the overall accuracy requirement. The reader can infer what the implications of the intuitive argument above would be.

5 Analysis of concatenated codes for quantum memories and channels

To see exactly what is needed to implement a quantum memory or channel, we give explicit worst-case relationships between the error parameters, making no assumption on fault-tolerant computations for recovery operators, but assuming an e-error-correcting code.

The parameters of the method are determined by the basic code used and the depth of the concatenation. The following lists all the relevant parameters:

l: The length of the code.

e_d: The maximum error intended to be corrected in a qubit between recoveries.

e_c: The total error after recovery (or decoding) of an encoded qubit with each supporting qubit subjected to an error of at most e_d.

e_{op}: The maximum error per operation.

h: The depth of the encoding hierarchy.

r: The repetition factor in the hierarchy. This must satisfy $r+1 \leq e_d/e_c$ for the final error to be bounded by e_d and the error to be preserved across recursive implementation of the code.

n: The number of intervals between recoveries and decodings at the leaves.

s: The length of the concatenated code, $s = l^h$.

We first determine the size of h to ensure n intervals between recoveries. The total number of intervals is $r^h \geq n$, so $h = \log_r(n)$. Thus $l^h = n^{\log_r(l)}$. Thus the overhead is polynomial in n. The exponent can be made small by choosing a large r, however that requires a more accurate implementation of the primitive operations.

Let ϵ be the desired final error introduced by the concatenated coding procedure, i.e. $\epsilon < e_d$. Given a fixed code, it is clear that e_c can be bounded in terms of a constant multiple of e_{op} and the unrecoverable error after correction of at most e_d error per qubit. The multiple is dependent on the number of primitive operations in the recovery, which is bounded by l^2, but can be less by a constant, depending on efficient implementation of the syndrome computations. If the error types are well understood, bounds can be obtained as discussed in [10]. A simple bound is obtained by directly adding the uncorrected error amplitudes. For $e_d < (e+2)/(2(l-e-1))$,

$$e_c \leq 2\binom{l}{e+1} e_d^{e+1} + l^2 e_{\mathrm{op}} .$$

Set e_c equal to the right hand side of the above equation, and use $re_c \leq e_d$ to get

$$e_d \geq 2r\binom{l}{e+1}e_d^{e+1} + rl^2 e_{\rm op}\,.$$

Set $a = 2r\binom{l}{e+1}$, $b = rl^2$ and write $e_d = \alpha b$. The inequality requires that $ab^e\alpha^{e+1} + e_{\rm op} \leq \alpha$. Assume $\epsilon < .5$ (which is reasonable). If $e_{\rm op} \leq \epsilon/(2a^{1/e}b)$, then $\alpha = e_{\rm op}$ is a solution. Thus it suffices to have

$$e_{\rm op} \leq \epsilon/(2a^{1/e}b) = \frac{\epsilon}{(2r)^{1+1/e}l^2\binom{l}{e+1}^{1/e}}\,.$$

For a 5 qubit one-error correcting code and $r = l$ (giving a linear qubit overhead) this gives $e_{\rm op} < 410^{-5}\epsilon$. This is of course too small for our current technical abilities. However, the estimate above does not take into account fault tolerant recovery methods and uses the most pessimistic estimates of error propagation. Taking account of this should improve the overall behavior of the simple concatenated coding scheme by at least a square root.

6 Acknowledgments

Special thanks to Ben Schumacher and Richard Hughes. We have greatly benefited from interaction with the Quantum Computer group at Los Alamos National Laboratory. We also thank Alexei Ashikhmin for his assistance with classical error correcting codes. This work was partially performed under the auspices of the U.S. Department of Energy under Contract No. W-7405-ENG-36.

References

[1] A. Barenco, C. H. Bennett, R. Cleve, pD. P. DiVincenzo, N. Margolus, P. Shor, T. Sleator, J. Smolin, H. Weinfurter, Elementary Gates for Quantum Computation, submitted to *Phys. Rev. A* (1995)

[2] C. H. Bennett, D. P. DiVincenzo, J. A. Smolin, W. K. Wootters, Mixed state entanglement and quantum error-correcting codes, quant-ph/9604024 (1996)

[3] S. L. Braunstein, Perfect quantum error correction coding in 26 laser pulses, quant-ph/9604036 (1996)

[4] A. R. Calderbank, P. W. Shor, Good quantum error-correcting codes exist, quant-ph/9512032 (1995).

[5] A. R. Calderbank, E. M. Rains, P. W. Shor, N. J. A. Sloane, Quantum error correction and orthogonal geometry, quant-ph/9605005 (1996).

[6] J. Cirac, P. Zoller, Quantum computations with cold trapped ions, *Phys. Rev. Lett.*, 74:4091 (1995)

[7] D. P. DiVincenzo and P. W. Shor, Fault-tolerant error correction with efficient quantum codes, quant-ph/9605031 (1996)

[8] R.J. Hughes, D.F.V. James, E. Knill, R. Laflamme and A.G. Petschek, Decoherence bounds on quantum computation with trapped ions, Los Alamos National Laboratory Report LAUR-96-1266 (1996)

[9] E. Knill, R. Laflamme, A Theory of Quantum Error-correcting Codes, Los Alamos National Laboratory Report LAUR-96-1300 (1996).

[10] E. Knill, R. Laflamme, Iterated error correction, preprint available from the authors (1996)

[11] R. Laflamme, C. Miquel, J.-P. Paz, W. Zurek, Perfect quantum error correction code, quant-ph/9602019

[12] J.H. van Lint, Introduction to Coding Theory, Springer-Verlag (1991)

[13] F. J. MacWilliams, N. J. A. Sloane, The Theory of Error Correcting Codes, North-Holland Publishing Company (1977)

[14] P. Shor, Scheme for reducing decoherence in quantum computer memory, preprint, (1995)

[15] P. W. Shor, Fault-tolerant quantum computation, quant-ph/9605011 (1996)

[16] A. Steane, Multiple particle interference and quantum error correction, *Prof. Royal Soc. London A*, in press; quant-ph/9601029 (1996)

[17] W. Zurek, R. Laflamme, Quantum logical operations on encoded qubits, quant-ph/9605013 (1996)