Quantum Error Correction

James Sayre M.D.

Department of Physics, University of Washington

Author Note

James Sayre

Department of Physics

University of Washington

Abstract

Quantum Computing holds enormous promise. Theoretically these computers should be able to scale exponentially in speed and storage. This spectacular capability comes at a price. Qubits are the basic storage unit. The information they store degrades over time due to decoherence and quantum noise. This degradation must be controlled if not eliminated. An entire field of inquiry is devoted to this is the function of quantum error correction. This paper starts with a discussion of the theoretical underpinnings of quantum error correction. Various methods are then described in detail.

Keywords: Quantum Computing, Error Correction, Threshold Theorem, Shor's Algorithm, Steane Code, Shor's Code.

Quantum Error Correction

Quantum Computing holds enormous promise. A useful quantum computer will have the capability to perform searches, factorizations and quantum mechanical simulations far beyond the capability of classical computers [1] [2]. For instance, such a computer running Shor's algorithm will be able to factorize a large number in polynomial time [3]. This is a tremendous speed advantage over a classical computer that requires exponential time to complete the same task. In theory a computation that requires millions of years on a classical computer could take hours on a quantum computer.

The basic building block of a quantum computer is a qubit. Analogous to a classical bit, the qubit combines with other qubits to store numbers. Unlike classical computers, that storage grows exponentially with the number of qubits. This spectacular capability comes at a price. The qubits are unstable. The information they store degrades over time. This degradation must be controlled if not eliminated. Formally this degradation is error at the qubit level. Errors occur both during storage and operations. The greatest source of these errors is interaction with the environment outside the quantum computer. Internal errors also occur. Practically, these errors are large enough that they will ruin all but the simplest of calculations. Improving the design of qubits can reduce the error rates. Modern qubits are still too error prone. Quantum error correction is the combination of additional qubits and code to identify then correct those errors. A functional quantum computer with currently available qubits will require error correction. Enormous effort is directed at quantum error correction [4].

Bits and Qubits

"I can only become invisible when no one is watching" -The Invisible Boy, Mystery Men

A bit is the basic unit of information storage in a classical computer. Each bit may exist in only one of two states, typically labelled 0 or 1. Information is stored in arrays of bits. The bits may be copied from one location to many others. They can undergo operations. The operations might involve a single bit or many. The operations on one set of bits can be conditional on the state of other sets of bits. Enormous effort over 70 years has resulted in machines with trillions of bits operating at gigahertz speeds with stunningly low error rates.

The qubit is the basic unit of information for a Quantum computer. In all implementations the qubit exists in two states. The crucial difference is that the qubit can exist in both states at the same time. This is the property of superposition. Each two-state qubit can exist in a complex linear combination of the two states, $\alpha|0> + \beta|1>$. The α and β factors are complex amplitudes for which $|\alpha|^2 + |\beta|^2 = 1$. Additionally, there is a non-observable phase factor. Hence the two complex amplitudes translate into two degrees of freedom for the superposition.

Qubits can be entangled. Two qubits can exist in four states. Three qubits can exist in eight states. In general n qubits can exist in 2^n states. Each state can have a unique complex amplitude c_n subject to the limitation that $1 = \sum_n c_n$. Hence, an array of qubits can store $2^n - 1$ independent numbers.

Operations on individual qubits can have effects over many states. Qubit operations are natively array processes. This feature allows single operations to have enormously powerful

effects on the arrays. Classical computers must operate on one number at a time. Quantum computers have array processing built it. Hence, Quantum computers have a natural speed advantage over their classical counterparts. In some cases, this speed advantage could be many orders of magnitude.

Measurement

Useful information must be recovered from a quantum computer. Otherwise they are a pointless exercise. Recovering that information is a central concern. Qubit information is derived by measuring the qubit. Measurement is a crucial consideration in Quantum Mechanics. Understanding measurement with all its ramifications has been the most controversial of subjects in physics since the 1930's. Agreement about the nature of measurement still eludes the physics community. The diversity of opinion about this topic is legendary. Nonetheless, the empirical findings in measurement are undisputed.

Measurement of a quantum state is dependent on the basis used for the measurement. That measurement will only return a value parallel to that basis. The probability coefficients will determine the relative probabilities of outcomes for repeated experiments. Suppose many measurements on identically prepared systems are performed. It is possible to obtain an estimate of the probability coefficients from those measurements. Increasing the number of measurements will increase the accuracy of the estimates. There is no way to directly interrogate the probability coefficients.

This is the central conundrum of quantum computing. Arrays of qubits hold vast potential for storage of data. They have a spectacular processing speed. Yet, they have a depressingly limited capability for direct output of any results. The algorithms must work around this

fundamental limitation. Fortunately, a few classes of problems have been solved theoretically. These classes are sufficiently interesting that great effort at creating a functioning quantum computer is ongoing.

No Cloning Theorem

Another limitation on qubits is the inability to copy an arbitrary qubit state to another qubit. Classical computers face no such limitation. They use the capability to copy data from one bit to another location billions of times per second. Qubit states can be swapped between two qubits. But a qubit state cannot be copied onto a second qubit. Here is the proof.

Suppose there is a machine that can copy one qubit onto another. Let Qubit A be the qubit to be copied. It is in an unknown pure quantum state $|\Psi>$. Qubit B is in a pure state $|e>$. Usually, but not necessarily, this will be the ground state. The combined state of the system is

$$|\psi> \otimes |e> .$$

The machine now applies a unitary operator U to the combined state such that

$$|\psi> \otimes |e> \rightarrow U(|\psi> \otimes |e>) = |\psi> \otimes |\psi> .$$

Since the machine can copy an arbitrary state, this equation will hold for a different state of qubit A as well. Let the second state be $|\varphi>$. The following two relations will hold

$$U(|\psi> \otimes |e>) = |\psi> \otimes |\psi>$$

$$U(|\varphi> \otimes |e>) = |\varphi> \otimes |\varphi> .$$

Taking the inner product of both relations

$$<\varphi e|U^{\dagger}U(|\psi e>) =<\varphi\varphi|\psi\psi>$$

$$<\varphi|\psi> =<\varphi|\psi>^2$$

This equation can only be true if

$$< \varphi|\psi > \ = 1 \textit{ or } 0$$

Hence, a cloning machine might only be able to clone a single pre-defined state

$$< \varphi|\psi > \ = 1 \therefore |\psi > \ = |\varphi >$$

or two orthogonal states.

$$< \varphi|\psi > \ = 0 \therefore |\psi > \textit{ is orthogonal to } |\varphi >$$

The cloning machine will be unable to clone an arbitrary qubit.

Computer Error

Consider a classical bit storing information. This might be the voltage across a transistor in a thumb drive. The classical bit can exist only in one of two states. The states are labelled 0 or 1. Over time the state might change incorrectly. A 0 might spontaneously change to a 1 or vice versa. This might be due to stray magnetic fields, cosmic rays or thermal noise. The probability of change of the state may be defined as $p < 1$. The probability that the state remains stable over a period of time is $1 - p$.

The classical computer operates on stored information. Operations are performed with the application of gates on bits. Gates can operate on single bits or groups of bits. Each of these gates are also subject to errors. The gates are separated in time and space. Any errors on a gate are usually assumed to be uncorrelated with any other gate errors. The errors are independent. An error on a gate will not affect the error rate on a subsequent gate. Uncorrelated errors in a process are known as Markovian errors [5]. The mathematics of Markovian processes are well worked out.

Fidelity of storage of the bits has improved over the decades. The error rates have dropped over time. A 2009 study shows that server ECC-DIMM memory suffered a recoverable error rate of the order of $p = 10^{-23}$ [6] . That translates to a single bit error in each 4GB DIMM on average much less than once per year.

Error correction is an important feature of classical computers. Early digital machines suffered error rates far greater than current machines. Storing, writing, reading and bitwise operations all could generate bit errors. Naturally the goal is to reach a zero-error condition for the final output of any computation. Even the best engineering will still result in a non-zero error rate. Additional methods must be applied.

The basic principle for error correction methods is to encode the data in such a way that errors may be corrected on the fly. Encoding usually means storing the data redundantly. That way if a part of the data is corrupted, then the original data is recovered when the data is decoded. Data is encoding and decoding is applied before and after each noisy step. The errors are identified and corrected at each step. In this way errors are isolated. They are corrected immediately. The errors do not propagate along the algorithm.

The archetypical technique is for a single bit of information can be encoded into a three-bit code-word. Each of the bits of the code-word are identical. This is a repetition code. If there is an error in one of the bits, then the code-word can be corrected. The majority of the bits is assumed to be the correct value. The code-word is then corrected to that value. Let the error rate be $0 < p < 1$. The possibility of two or three errors is

$$p_f = 3p^2 (1 - p) + p^3$$

If $p_f < p$ then the error correction method is effective at reducing errors. This occurs whenever $p < \frac{1}{2}$. The repetition scheme is strongly improved by reducing p. Adding additional bits to the codeword increases the fidelity of the correction. For instance, a seven-bit code-word could suffer three errors and still perform the correction properly.

Quantum Error Correction

Qubits are subject to errors. The error rate for modern qubits is far higher than for DIMM's. The p-values for modern physical qubits are of the order of 10^{-4} [4] .These errors obey fundamental rules of quantum mechanics. A change in state of a qubit implies an interaction with some other quantum system nearby. That other system might be internal to the quantum computer or external. A qubit in perfect isolation might theoretically be able to remain stable over a long time. Such isolation is not physically possible. Heroic measures are applied to reduce the coupling. For instance, quantum computers are cooled to the mili-Kelvin range in order to reduce coupling. They are situated in sound damped electrically isolated rooms. The internal structure of the computer uses exotic materials and architecture to isolate the qubits from nearby controlling hardware. Despite these efforts, some coupling outside the qubits will always be present.

Errors are unwanted shifts in the probability amplitudes. The amplitudes will still sum to one. The qubit errors therefore represent unwanted rotations on the Bloch sphere. They may be treated mathematically as unitary operations. The errors may be small shifts or large. They can occur during gate operations, storage, initialization and readout. In practice these errors are large enough that without active error correction a useful quantum computer is inconceivable.

In a classical analog system, errors occur constantly with varying amplitudes. Some of these errors might be so small that they cannot be measured. In that instance such errors are effectively zero. For this analysis such small errors are still assumed to exist. The analog errors are typically well handled with standard probability distributions, such as a gaussian.

Qubits errors are unlike analog system errors in that quantum interactions carry a probability of occurrence. The interaction might cause a shift in the probability amplitudes. Or there may be no effect at all. This is fundamentally different from an analog situation where the errors have a probability of occurrence of 1. In the quantum computer, the errors will have a binary probability of occurrence superimposed on the variability of magnitude.

Hard Limits to Error Correction

Active error correction requires that the following hard limits be addressed.

- **Errors are continuous**. The complex probability amplitudes are continuously variable. This confers enormous capability on arrays of qubits. However, the errors will be continuous as well. The repetition codes used in digital computers will function but are greatly diminished in effect. Errors can be ameliorated not expunged. Errors will propagate and sum over the course of the computation.
- **No Cloning**. It is not possible to copy the arbitrary state of one qubit onto another qubit. The repetition and more complicated error correction codes in classical computers all require this operation. Quantum computers may not use this technique.

- **Measurement**. Classical error correction relies on evaluating the state of a code-word then applying a correction. This is the equivalent of measuring the quantum state. As seen above, measurement collapses the state to one of the basis vectors. The information in the qubit of interest is lost. Quantum error correction must avoid direct measurement of the code-words.

At first blush these limitations seem to be the death knell for quantum computing. The original impetus for this author was to explore how these limitations can be overcome. It turns out that one of these limitations lead to methods that render quantum computing theoretically possible. It is a mathematical/theoretical tour de force. The hard limit of measurement collapse is turned into a virtue. It is this virtue that allows the possibility of reducing errors to an arbitrarily low level.

Bit-Flip Correction

Three qubit bit-flip code is analogous to the classical code described above. The original impetus was to develop a technique similar to the one that worked in classical computers. The great limitation is the No Cloning theorem. Classical correction copies the original bit onto other bits. It is not possible to directly copy a quantum state onto a separate qubit. It *is* possible to create an entangled state in three qubits. If an error occurs on one of the qubits, it is then possible to perform a correction to recover the original state on the initial qubit. The recovery is possible by judicious measurement of two of the qubits. Error correction is applied based on the results of those measurements. The three-bit quantum code is shown below.

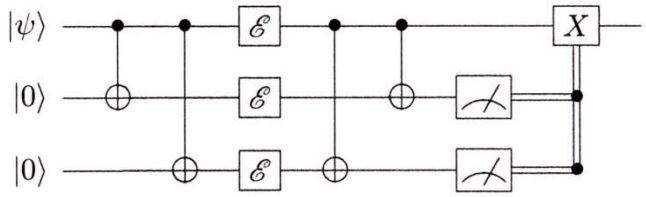

Figure 1 Bit flip code [7]

The code is analogous to the classical three-bit correction code. The first two CNOT gates encode the state to $\alpha|000> + \beta|111>$. The state is "copied" onto the basis. This doesn't violate the no cloning theorem as the state is not copied to an independent qubit. The three gates labelled E at the center of the code represent potential bit-flip errors. The application of the flanking CNOT gates creates an error syndrome. The following measurements evaluate the syndrome. The application of the CNOT corrects the wave-function back to the original state. This correction code will work for any single bit-flip error. In this derivation the error is a bit flip or Pauli X gate. The errors occur with frequency p. The gates are inactive with frequency $1-p$. If there are no errors, then the code should evolve this way

$$CX_{12}CX_{13}\,(\alpha\,|000> \;+\beta|\,111>) = \alpha\,|000> \;+\beta|\,100> \;= (\alpha\,|0> \;+\beta|\,1>)\,|00>$$

The original state on qubit one is recovered. The measurement operation recover 0 and no correction is applied. Suppose an error occurs on the first qubit. The code evolves this way

$$CX_{12}CX_{13}\,(\alpha\,|100> \;+\beta|\,011>) = \alpha\,|111> \;+\beta|\,011> \;= (\alpha\,|1> \;+\beta|\,0>)\,|11>$$

The measurement gates both become active and the pauli X gate is applied to qubit one recovering the initial state, $X_1\ (\alpha\,|1> \;+\beta|\,0>)\,|11> \;=\; (\alpha\,|0> \;+\beta|\,1>)\,|11>$. Errors on the other two lines also recover the initial state to qubit one.

The code should fail if there are two or three errors. That occurs with frequency $3\,(1-p)\,p^2 + p^3$. The single qubit should fail with frequency p. The relation

$$p > \; 3\,(1-p)\,p^2 + p^3$$

holds if $p < \frac{1}{2}$. Hence the correction scheme improves the error rate if the individual qubit error rate is less than $\frac{1}{2}$.

Fidelity

Some initial states suffer greater changes from bit flip errors than other initial states. For instance, the state $\frac{|0>+|1>}{\sqrt{2}}$ is unaffected by a bit flip. Hence an error here would make no difference to a calculation. But a bit flip for the state |0> would result in state as far as possible from the correct value. It is important to develop a measure for an average degradation to the information based on the single qubit as compared with error corrected logical qubits.

Hamming distance is a classical metric of how far apart two states are. It is defined as the number of bits that are different between two bit arrays. Fidelity, F, is one metric of quantum distance. The fidelity measurement allows an estimate of the degradation of a circuit by errors.

Fidelity is defined as $F(\sigma,\rho) = \sqrt{<\sigma^{\frac{1}{2}}|\rho|\sigma^{\frac{1}{2}}>}$, where σ is the density matrix of initial state and ρ is the density matrix of the final state. If ρ and σ are identical, then the distance is zero and $F = 1$. Error correction should decrease distance between the initial and final states. Correction should therefore increase fidelity.

The qubit initial state will be $|\psi> = a|0> + b|1>$. For bit-flip errors, the final density matrix will be

$$\rho = (1-p)|\psi><\psi| + pX|\psi><\psi|X.$$

So

$$F = \sqrt{(1-p) + p<\psi|X|\psi><\psi|X|\psi>}.$$

The fidelity is lowest when $<\psi|X|\psi> = 0$. That occurs when $|\psi> = |0> \; or \; |1>$.

$$F_{min1} = \sqrt{(1-p)}.$$

For three qubits the initial state is $|\psi> = a|0_L> + b|1_L>$. The final density matrix is

$$\rho = ((1-p)^3 + 3p(1-p)^2)|\psi><\psi| + (3p^2(1-p) + p^3)X|\psi><\psi|X$$

So

$$F = \sqrt{((1-p)^3 + 3p(1-p)^2) + (3p^2(1-p) + p^3)<\psi|X|\psi><\psi|X|\psi>}$$

Again, The fidelity is lowest when $<\psi|X|\psi> = 0$.

$$F_{min3} = \sqrt{((1-p)^3 + 3p(1-p)^2)}$$

$$F_{min1} < F_{min3} \;\; if \;\; p < \tfrac{1}{2}$$

Regardless of the initial state, the degradation of the state will be less with the three-qubit error correction code than without it. That is, if there are two or three errors, the overall fidelity will still be improved with the error correction code.

Phase Flip

The bit flip error operator is the Pauli X. Qubits can suffer Pauli Y and Z errors as well. The Pauli Z operator is a phase flip. For this error, the 3-qubit error code above is inadequate. The X operator will have a profound effect on the pure state $|0>$ but no effect on the Bell state $\frac{|0>+|1>}{\sqrt{2}}$. Conversely the Z operator will affect the Bell state but leave the pure state unaffected. A different strategy is required for correcting phase flips.

We do have a correction code that works well where the pure states are reversed by the X operator. An ingenious method is to transform the initial state into a basis for which the Z operator has the same effect. By applying the Hadamard operation to the input and output of the error correction, the basis of the initial state is changed to a basis where our error correction code is now effective.

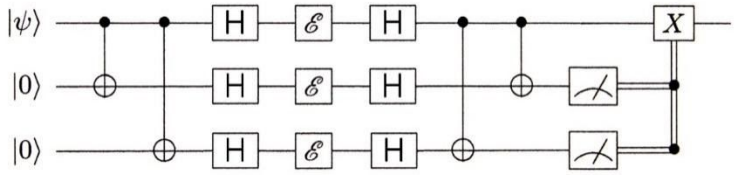

Figure 2 Phase flip code [7]

In this code the errors also occur with probability p. The error is the Pauli Z operator. The initial state is encoded exactly as before in the bit flip code. Then the state is rotated to the $|+>, |->$ basis. The errors are allowed to occur with the Z operator. The state is rotated back to the $|0>, |1>$ basis and correction is applied. The mathematics are identical for outcomes and fidelity.

Shor Code

The natural next step to the flip codes is to correct for the Y error. The Y operator is equivalent to the operator iXZ. The factor i adds only a non-observable phase to the qubit and may be safely ignored. Hence correcting for both X and Z would automatically correct for Y. A reasonable approach would be to nest the X and Z corrections. This is the impetus for the Shor Code. It uses 9 qubits in sets of three.

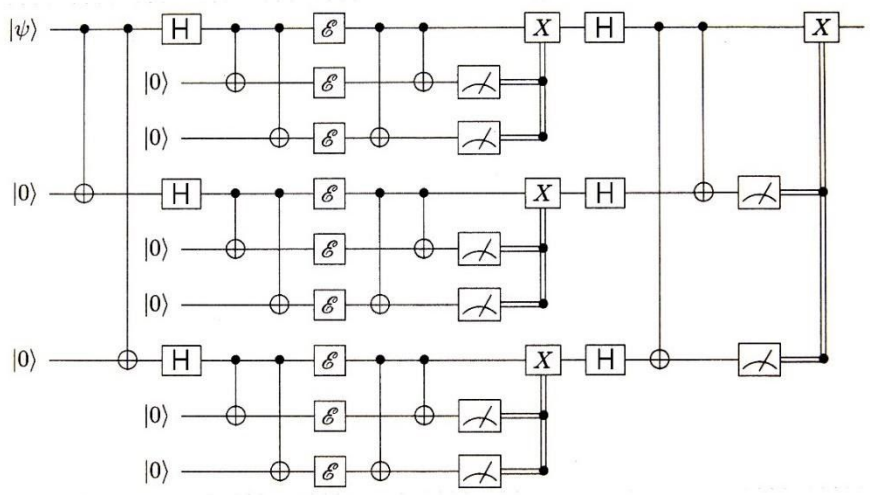

Figure 3 Shor code correcting for bit and phase flip errors [7]

Suppose an error occurs on the first qubit. The error is a combination of X and Z rotations. The inner code of the three qubits between the upper Hadamard operations will correct the Z error. The X error won't be corrected. The qubit with the X error now passes out of the inner code via the Hadamard. This changes the qubits back to the $|0>, |1>$ basis. Now the outer code composed of three qubits will correct the residual X error.

Discretization

The act of measurement leads to a surprising result. Suppose that error is no longer just the presence or absence of a bit or phase flip. Rather, suppose that the error E_i is a linear combination of operators

$$E_i = e_{i0}I + e_{i1}X_1 + e_{i2}Z_1 + e_{i3}X_1Z_1.$$

The quantum state $E_i|\psi>$ is a superposition of the terms $|\psi>, X_1|\psi>, Z_1|\psi> and X_1Z_1|\psi.$ Measurement collapses the state to one of these terms. Our error correction code corrects for any of these four states. Our code was initially designed for full bit and phase flip errors. It also works for any linear combination of errors. Even tiny rotations are corrected by the same code! The errors may be tiny or large along a continuous scale. The action of measurement causes the errors to be discrete. The errors are then amenable to correction with Pauli X and Z gates. This is an amazing result. Any single error of arbitrary magnitude can be corrected using these codes. The way is open for exact calculations using noisy systems.

Operator Measurement

The Shor code seems to contradict the measurement restriction. The code successfully determines whether an error has occurred by measuring qubits. The method employed in Shor is a variation of a basic circuit principle. There is a way to measure the eigenvalue of an operator acting on a qubit without changing the qubit's state. We are not learning the probability amplitudes. We are learning what the effect of a particular operator would be. A circuit to measure an operator with eigenvalues of ± 1 is

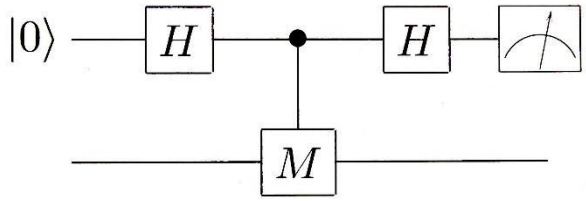

Figure 4 Operator measurement [7]

The qubit being measured is an ancillary qubit. It is initialized to the $|0>$ state. This circuit leaves the qubit of interest unchanged for any Pauli operation.

Quantum Hamming Bound

The Shor code corrects for any linear combination of errors produced by the Pauli X, Y or Z operators. The Y operator is equivalent to the XZ error in the equation above. It is useful to know the minimum number of qubits necessary to correct for an arbitrary number of errors. Assume that we use non-degenerate code-words. Suppose we use this code to encode k qubits with n qubits and can correct errors on t or fewer qubits. There are $(n\,j\,)$ combinations for j errors on n qubits. There are three possible errors that might occur independently. Therefore, the total number of errors that might occur on t or fewer of n qubits is $\sum_{j=0}^{t}(n\,j\,)3^{j}$. The words must be encoded onto an orthogonal non-degenerate subspace. This must be a subspace of the 2^{n} space of the n-qubits. The following equation relates the subspace to the 2^{n} qubit space

$$\sum_{j=0}^{t}(n\,j\,)3^{j}2^{k}\leq 2^{n}.$$

For a single qubit encoded with n qubits, tolerating a single error, the bound is $(1+3n)\,2\leq 2^{n}$. The bound is satisfied if $n\geq 5$. For instance, the Shor code requires 9 qubits to

correct for errors on a single qubit. The overhead is nine to one. Far better if the overhead can be reduced. The Hamming bound suggests that the overhead can be reduced to five to one.

Linear Codes

The quantum codes above used classical digital codes as inspiration. Continuing that theme leads to important results. Classical linear codes encode digital words into longer codewords. The increased length is effectively redundancy. Judicious design of the encoding process yields robust error correction with shorter codewords.

Classical binary code encodes k bits of information into n bit codewords. n will be larger than k. Codewords form a space. The codewords are linear if any of the codewords added together result in codeword in the same space. The addition is modulo 2. The codewords can then form a vector space with dimension n. The field F_2^n is defined as n bit words with 2 possible values for each of the bits, typically 0 and 1. Each codeword will be a vector, v_i. In a closed space $v_1 + v_2 = v_f$ where v_f is also in the closed space, called C. The codewords can be derived from a complete basis, $\{w_i\}_{i=1}^{k}$ or

$$v = \sum_{i=1}^{k} a_i w_i$$

With $v \in C,\ where\ a_i = 0\ or\ 1$. Let the factor a_i be the ith bit of the word to encode, then the codeword v will be a linear combination of the basis vectors w_i. The basis vectors are conveniently assembled into an $n \times k$ generator matrix G. The codeword is generated by taking the modulo 2 matrix multiplication of the word to be encoded with the generator.

A seven-bit Hamming generator is

$$G = (1\ 1\ 1\ 0\ 0\ 0\ 0\ 1\ 0\ 0\ 1\ 1\ 0\ 0\ 0\ 1\ 0\ 1\ 0\ 1\ 0\ 1\ 1\ 0\ 1\ 0\ 0\ 1).$$

In this case the words are 4 bits in length. The codeword is then given by $v = mod(a \times G)$. It is possible to create another matrix H for which each row i gives

$$Mod\ 2\ (H_i \cdot v) = 0$$

Where 0 is a column vector of length k with values all zero.

H is of dimension $(n-k)\ by\ k$. It is known as a parity matrix. Each of the codewords multiplied by H will give a 0 result. If there is a single bit error, then the result will be 1. Many different parity matrices can be created given a generator matrix. A requirement is that each row be orthogonal to each column of the generator. An additional requirement is that any d-1 columns be linearly independent.

Applying a parity allows the detection of errors. The codewords were selected such that any single bit error does not map to any other codeword. Hence, single errors in the codeword can be corrected directly back to the original codeword. In this instance, one Hamming parity matrix is

$$(1\ 0\ 1\ 0\ 1\ 0\ 1\ 0\ 1\ 1\ 0\ 0\ 1\ 1\ 0\ 0\ 0\ 1\ 1\ 1\ 1)$$

The parity matrix has the feature that the vector resulting from the multiplication $H \cdot v$ corresponds to a column of the Hamming matrix. In this case for instance, a result of $(0, 1, 1)$ indicates the error occurred on the 6th bit. The correction is easily applied. Let w be word that has a single bit error. The word will one of the codewords plus an error word with a single bit set, e. $H \cdot w = H \cdot (v + e) = H \cdot v + H \cdot e = 0 + H \cdot e$. $H \cdot e$ is the error syndrome. The single bit set picks the column elements of the parity matrix and uniquely identifies the erroneous bit.

Any linear code may be classed as an $[n, k, d]$ code where k bits are encoded into k bits. The distance d is the number of errors that the code can *detect*. In the parity matrix no two columns are linearly dependent. However, there are three columns that are linearly dependent. Hence, the distance $d = 3$. The number of correctable errors is $t = (d - 1)/2$. The Hamming code above encodes 4 bits into 7 bits with one correctable error allowed. It is a [7,4,3] code. The generator matrix G for a code C is a $k \times n$ matrix. The parity matrix H is a $(k - n) \times n$ matrix.

CSS Codes

The classical codes correct for bit flips. Quantum codes must also correct for phase flips. The Shor code above was the first discovered in a class of QECC known as CSS codes. These were based on the classical linear codes altered to correct for phase flips in addition to bit flip. As shown above the application of a Hadamard rotation allows the bit flip operations to instead correct phase flips. The basic method is to use the Hadamard operation to encode the qubit ensemble so that the bit flip correction also corrects for phase flips. In the Shor code this was done by nesting Hadamard encoded ensembles. More efficient methods deploy the Hadamard operation in a more cunning manner.

It is possible to reverse the roles of the generator and parity matrices. Consider a code $C^{\perp}$ where the roles of G and H are reversed. $C^{\perp}$ is known as the dual of C. $G^{\perp} = H \text{ and } H^{\perp} = G$. $C^{\perp}$ will be a $[n, n - k, t']$ code. The matrices G and H are orthogonal. $HG^{T} = 0$. Exchanging the roles of the H and G matrices does not alter the orthogonality. Hence the $C^{\perp}$ code generator and parity matrices are also orthogonal. The $C^{\perp}$ codewords are orthogonal to the C codewords. Combining the two codes leads to a quantum code that corrects

for both bit and phase flip errors. Suppose there is an $[n, k_1, d_1]$ code C with a subcode $C_1\ [n, k_2, d_2]$ and dual code $C_1^{\perp} = C_2\ [n, n-k_2, d_3]$. It is then possible to create a bit and phase flip correcting code $C_q\ [n, k_1 - k_2, d_{min}]$ where d_{min} is the minimum of d_1 and d_3. These are known as CSS codes. The derivation is in reference [7] page.

The central notion is that a portion of the encoded qubits are not measured directly. The encoded qubits are entangled with a set of ancilla qubits. This entanglement is complex and specific. The ancilla qubits are then measured. The appropriate error correction is then applied. This leads to the Steane Code [8]. It is the best known of the CSS codes. It is a $[7, 1, 3]$ code. One qubit is encoded into 7. The code can correct one error. The logical qubit word is constructed from the Steane generators. $|0_L> = |0 + C_2>$. For the example above, the parity matrix of C_1 is the generator matrix of C_2. Taking all linear combination of that matrix, the encoding is:

$$|0_L> = \tfrac{1}{\sqrt{8}}[|0000000> + |0001111> + |0110011> + |0111100> + |1010101> + |1011010> + |1100110>$$

$$|1_L> = \tfrac{1}{\sqrt{8}}[|1111111> + |111000> + |1001100> + |1000011> + |0101010> + |0100101> + |0011001> +$$

A bit flip or phase flip on either of these code words can be detected and corrected with appropriate application of gates.

Stabilizer Codes

The Z_1Z_2 and X_1X_2 act as identities for the $|+> = \tfrac{1}{\sqrt{2}}(|00> + |11>)$ state.

$$Z_1Z_2|+> = X_1X_2|+> = |+>$$

These operators are defined as stabilizers for the $|+>$ state. The $|+>$ state is unique in that it is the only state that is stabilized by both these operators.

The states for the Steane code above are difficult to work with. They are complex. Carrying calculations on quantum codes is cumbersome. The stabilizer formalism allows a much more compact notation for working with qubits in error correction.

The stabilizer formalism relies on Group theory. The Pauli group is all Pauli operators and the identity matrix, each with factors of $\pm 1 \; and \pm i$ for a total of sixteen operators. Pauli operators for n qubits will be the tensor multiple of each of the Pauli matrices in qubit order. Let this Pauli group be G. Define a subgroup of G, S. Let V_s be the set of qubit states stabilized by S. That is, any member of V_s is unchanged by the application of any operator in S.

The group $S \equiv \{I,\ Z_1Z_2,\ Z_2Z_3,\ Z_1Z_3\}$ for the case where there are 3 qubits is considered as an example. The I operator stabilizes any state of three qubits. The states $|000>,\ |001>,\ |110> \; and \; |111>$ are stabilized by Z_1Z_2. The states $|000>,\ |011>,\ |100> \; and \; |111>$ are stabilized by Z_2Z_3. The states $|000>,\ |010>,\ |101> \; and \; |111>$ are stabilized by Z_1Z_3. Hence, $|000> and \; |111>$ are the only states stabilized by all four operators in the group S. All other states will be altered by the application of one or more of the operators. Using a simple 3-qubit repetition code would allow the group S to act as an error check. Measuring the operators in group S for the codewords $|000> and \; |111>$ return eigenvalues of 1. All other words return eigenvalue -1. By carefully selecting the measured operators the flipped bit can be identified and then corrected in the following step.

Stabilizers allow an efficient description of the code. In the example above, the stabilizer $Z_1 Z_3$ is the same as $(Z_1 Z_2)(Z_2 Z_3)$. I is equivalent to $(Z_1 Z_2)^2$. The description of the code is then reduced to $S = < Z_1 Z_2, Z_2 Z_3 >$. Here the two elements are the generators of the group S. All elements of the group can be created by multiples of the generators. In effect this means that any operation in the group can be performed with combinations of the generators. With the Pauli matrices, any group can be shown to have at most O(log N) generators where N is the number of elements in the group.

For the Steane code above, the stabilizer generators are given by:

Name	Operator
g_1	$I\ I\ I\ X\ X\ X\ X$
g_2	$I\ X\ X\ I\ I\ X\ X$
g_3	$X\ I\ X\ I\ X\ I\ X$
g_4	$I\ I\ I\ Z\ Z\ Z\ Z$
g_5	$I\ Z\ Z\ I\ I\ Z\ Z$
g_6	$Z\ I\ Z\ I\ Z\ I\ Z$

In the operator measurement above there was only a single qubit undergoing a CNOT. However, the technique works for ensembles of qubits. Any members of an ensemble of qubits can be included in the operation to be measured. There is no restriction to operate on all qubits or just one qubit of the ensemble. The result will be a sum mod 2 of the operated qubits.

The Steane generators can be measured over the ensemble of seven qubits. If the ensemble is identical to the codewords then the result will be eigenvalue +1. If there is a single error, then the result will be -1. By measuring each operator, it is possible to characterize

whether the error was a phase flip, bit flip or both. The measurements also give an error syndrome. The error correction path is easily calculated from that syndrome.

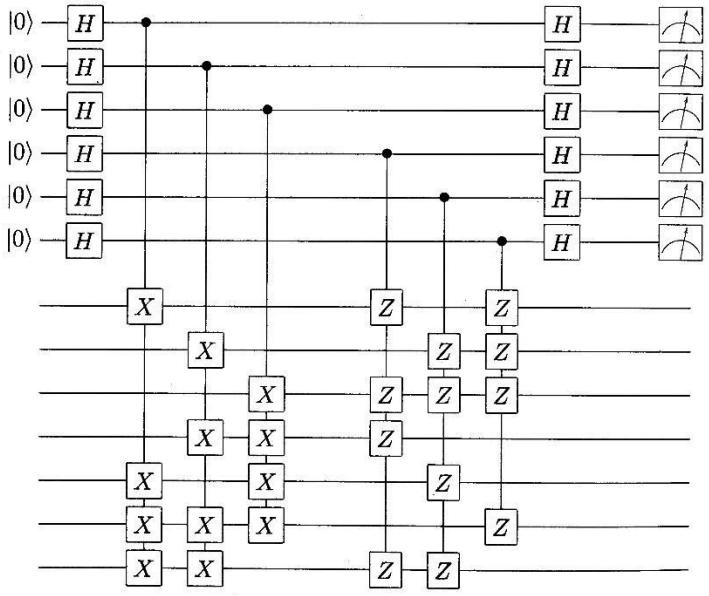

Figure 5 Steane error recovery code [7]

A simple analysis for the seven-qubit ensemble shows that the frequency of a two or more errors on the ensemble is $1-(1-p)^7-7p(1-p)^6$. As p becomes small this expression reduces to $21p^2$ A single qubit will have an error rate of p. Error correction with 6 additional qubits will only be of benefit if $21p^2 + O(p^3) < p \text{ or } p < 0.058$. For the Steane code there is a threshold error rate of approximately 5.8% using this simplistic analysis. That means the error rate is unchanged overall with the addition of 6 qubits. If layered error correction is to be useful, the error rate must fall faster than the number of qubits used to achieve the reduction. The naïve analysis requires that the error rate fall by a factor of 7 over the single qubit rate just to make the exercise worthwhile. A better error threshold would be 0.007. Here the reduction in error rate exactly matches the number of qubits in the seven-bit Steane code.

Fault Tolerant Operations

Up to this point the error corrections have been directed purely at storing information. It is a remarkable result that analog quantum information errors can be corrected by discretization. Storing data is important but it is not enough. At readout an ensemble of qubits carries the same information as an identical number of classical bits. So just storing data in a qubit ensemble would be a fool's errand. The fantastic difficulty of working with qubits compared with their classical counterparts means a purely storage quantum array is untenable.

The benefits of quantum computing enter when operations are performed on the data. Specific algorithms can be spectacularly faster operating with qubits than with classical bits. Here is where the concept of fault tolerant computation comes in. Each of the algorithms requires application of numerous gates to the qubits. The qubits must be initialized to specific states. Those states then undergo numerous operations. The states are finally read out to provide an answer. Each of those operations can create errors with probabilities specific to those operations.

Higher level algorithms such as the Grover search and Shor factorization codes are designed to operate on registers of single qubits. Error correction is not built in. Adding error correction complicates the algorithms. At first glance the qubits could be encoded, error corrected and then decoded prior to each operation. This cycle recurs for the entirety of the algorithm. But each decode-encode cycle adds to the probability of uncorrected errors entering the calculation. Far better to operate on the encoded qubits and apply error correction after each operation. That way an entire encode-decode cycle is avoided between each step. Only the error correction is necessary.

Applying gates to individual qubits is built into the hardware. Applying gates to encoded data is more complicated. It requires application of gates to specific qubits in specific order to achieve the same effect. These are known as fault-tolerant procedures. A useful quantum computer may apply a universal set of gates to any of the qubits. A universal set is defined as a set of gates that can be combined to perform any possible operation on the qubits. For instance, one universal set of gates is composed of three gates. The Hadamard, phase, $\frac{\pi}{8}$ and CNOT applied in various combinations can perform any arbitrary qubit operation.

The composition of fault tolerant operations is uniquely dependent on the encoding procedure. The stabilizer generators will determine the exact procedures for application of gates within a specific fault tolerant operation. The procedure for Hadamard application is straightforward. The H operation is applied to each qubit in the logical qubit.

Concatenation

Error correction is applied to each of the qubits. This is at the expense of a greatly increased number of qubits. Suppose the Steane [7,1,3] code is used for a logical qubit. 7 qubits are required for the error correction. This reduces the error rate. But, it may not reduce the error rate to an acceptable level. Each of the 7 qubits may then be encoded with a second layer of error correction. That way each of the first layer qubits will have a lowered error rate and the logical qubit will experience an even lower error rate. There are now 49 qubits in the logical array. A third layer might be applied to reduce the error still further. This comes at the expense of 343 total qubits. The layering of qubit arrays is concatenation. The error rate decreases. But the number of qubits expands exponentially. This is potentially damning for useful computation. If

the error rate does not fall faster than the number of qubits, then an arbitrary accuracy might not be obtainable.

Assume a four-layer code. Suppose that there is a failure rate for the single gate on a physical qubit is p. The layer above will yield an error rate of cp^2 where the factor c is the number of points in the circuit where two errors will cause an error to propagate through the logical qubit. The factor c is determined by the characteristics of the gate and qubits used for the calculation. The next layer up will have an error rate of $c(cp^2)^2 = c^3p^4$. The highest layer is the logical qubit. It will have an error rate of $c(c^3p^4)^2 = c^7p^8$. In general, the error rate will be $c^{2^k-1}p^{2^k}$.

Threshold Theorem

The Threshold Theorem was first presented in 1997 [9]. The theorem states that there exists an error rate small enough that with sufficient error correction, any quantum computation may be performed to a desired accuracy. Any desired level of accuracy is obtainable with a polynomial number of qubits if the error rate p is lower than a certain threshold.

The theorem makes several assumptions. Most important is that the noise is Markovian. That is each individual error is uncorrelated with any other. The errors are entirely random. Gates will have no systematic errors. The delay lines also have purely uncorrelated errors. No uniform drift is allowed. The error correction is applied by creating logical qubits from groups of physical qubits. A single qubit is entangled with several qubits. These additional qubits are measured as a syndrome. The measurement determines what corrective action need be taken. The logical qubit is corrected back to the proper state.

The derivation in the paper is complex. A simplification is to consider the error rate of concatenated code given above. Assume a desired error for the final calculation of ε and a measure of the size of the quantum circuit n. The number of gates will be a polynomial function in n, $p(n)$. The factor c is the number of points in the circuit where two errors will cause an error to propagate through to the logical qubit. For concatenation to get to the desired error rate, the following relation must hold

$$\frac{(cp)^{2^k}}{c} \leq \frac{\varepsilon}{p(n)}$$

The exponent on the right will blow up if $p > 1/c$. A threshold value is defined. $p_{th} \equiv 1/c$. The number of gates n is bounded on the order of

$$O(poly(\log log\, p(n)/\epsilon)) .$$

This means that any needed level of accuracy of results can be obtained with a poly-logarithmically limited number of gates. If the error rate is below a certain value, then tractable number of gates will accomplish the desired calculation.

Threshold example

The factor c is of crucial importance to the threshold. It is the number of places where an error will cause failure of the circuit. Take a simple fault tolerant circuit. For our purposes a CNOT gate between two logical qubits. The fault tolerant circuit is

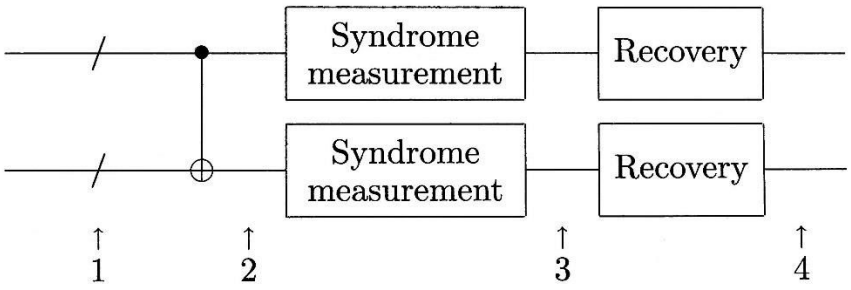

Figure 6 Fault tolerant error correction circuit for CNOT [7]

The important consideration is propagation of errors through the circuit. Errors may arise anywhere in the circuit. It is the action of syndrome measurement and recovery to ensure that none of the errors survives the operation. Making reasonable assumptions about where the errors occur and their interactions it is possible to derive an estimate of $c \cong 10^4$ for the Steane code. Hence, $p_{th} = 10^{-4}$.

Threshold Considerations

"The more they overtake the plumbing, the easier it is to stop up the drain."

-Commander Montgomery Scott, Chief Engineer, Federation Starship Enterprise.

The threshold theorem is the lynch pin of Quantum Computing. The theorem gives assurance that any quantum computation can be achieved with a non-exponential number of gates. Without this assurance, quantum computation is at risk. The number of gates could become exponentially large. This is not tenable. A large calculation might require a physically impossible number of qubits.

It is one thing to show that the number of qubits is polynomial bounded for any computation. It is a different thing when the calculation shows how many qubits will be required for a real-world machine. The estimates for the number of qubits range in the 10^5 *to* 10^7 . This number of classical bits is easily in reach with modern production techniques. Qubits on the other hand require close proximity in order to interact. It is hard to imagine a geometry that allows for interaction between any pair of 1 million qubits. The algorithms may partially reduce the need for every pair of qubits to be close. That restriction might drop by a factor of 1000. That still means 1,000 qubits must all be in close proximity. No geometry satisfies that requirement.

An assumption of the threshold theorem is parallel processing. Gate operations must be carried out concurrently. If not, some qubits would undergo delays. Additional error correction steps are required for stabilizing the qubits waiting for the next gate. Those steps require additional gates. The number of gates is then exponential. The threshold theorem no longer applies.

Each layer of error correction relies on measuring ancilla qubits. The ability to rapidly and accurately initialize many ancilla qubits is crucial. The most efficient and accurate methods require several gates to prepare the ancilla qubits. They are usually prepared to the |0> state. The state is assured by measurement. But, if the measurement basis is not perfectly aligned, then systematic errors are introduced. The notion of ensuring that the measurement basis of 10^5 qubits are all within the 10^{-4} threshold is a staggering challenge.

Even if measurement is performed perfectly, there is still the problem of control. The outcome of the measurements determines the next step in the process. The application of specific gates on specific qubits depends on the outcome of the measurement. Use the Steane code as an

example. The measurements on six ancilla qubits will determine which of two gates on one of seven data qubits will be activated. That decision and control process is handled externally by a classical computer. That means each logical block sends six bits out and receives two of fourteen signals back. Depending on the hardware the communication overhead might be reduced with multiplexing. At best 6 bits out and 4 bits back must occur for each clock cycle in the Quantum computer. And this must occur for each logical qubit at each error correction level. The data rates for such operation seem staggering.

A Final Classical Consideration

Quantum computing has been shown to have a theoretical advantage over classical computing for many computational problems. That presumed advantage was shaken recently. A quantum recommendation algorithm had been felt to be superior to any classical algorithm. And yet a classical algorithm was found that outperforms the quantum version. [10, 11] The same researcher was able to perform that same feat with a second algorithm for inversion of low rank matrices. [12] The notion of Quantum supremacy is based on the current understanding of classical algorithms. Future classical algorithms might be found to nullify that supremacy.

Conclusion

Quantum computers hold tremendous promise. They face enormous challenges. The greatest of these is error. Errors occur at every level in any current realization of a quantum computer. If the error rate can be reduced below a threshold typically quoted as 10^{-4} then error correction procedures will reduce the error of the final answer to an acceptable level. Other

challenges of architecture, uniformity, speed and control must also be addressed. They may even be more stringent than the error threshold.

The ability to perform fantastically difficult computations out of reach of classical supercomputers is a siren call. Researchers around the world will continue to attack the remarkably difficult challenges on the road to a useful quantum computer. It is unclear if the effort will pay off. The reward is so great that many are willing to pay enormous cost in time and money for the quest.

References

[1] R. Raz and A. Tal, "Oracle Separation of BQP and PH," ECCC, 2018.

[2] Editorial, "Computer Games: Classical and Quantum Machines are Battling for Computational Superiority," *Nature,* p. 302, 27 Dec 2018.

[3] P. W. Shor, "Algorithms for quantum computation: discrete logarithms and factoring," in *Proceedings 35th Annual Symposium on Foundations of Computer Science*, 1994.

[4] R. Takagi, T. J. Yoder and I. L. Chuang, "Error Rates and Resource Overheads of Encoded Three-Qubit gates," *Phys. Rev. A.,* pp. 042302-1,13, 4 Oct 2017.

[5] A. A. Markov, Theory of Algorithms, Moscow: Moscow Academy of Sciences, 1954.

[6] B. Schroeder, E. Pinhiero and W. Wold-Dietrich, "DRAM Errors in the Wild: A Large-Scale Field Study," in *Sigmetrics/Performance '09*, Seattle, 2009.

[7] M. A. Nielsen and I. L. Chuang, Quantum Computation and Quantum Information, Cambridge: Cambridge University Press, 2010.

[8] A. M. Steane, "Error Correcting Code in Quantum Theory," *Phys. Rev. Lett.,* pp. 793-7, 29 July 1996.

[9] M. B.-O. Dorit Abrahamov, "Fault-tolerant quantum computation with constant error," in *Proceedings of the twenty-ninth annual ACM symposium on theory of computing*, 1997.

[10] E. Tang, "A quantum-inspired classical algorithm for recommendation systems," ECCC, 2019.

[11] I. Kerenidis and A. Prakash, "Quantum Recommendation Systems," in *Leibniz International Proceedins in Informatics*, Leibniz, 2017.

[12] A. Gilyen, S. Lloyd and E. Tang, "Quantum-inspired low-rank stochastic regression with logarithmic dependence on the dimension," in *Preprint*, 2018.