An Overview of Topics in Quantum Error Correction

Thomas Catalan

June 2022

1 Introduction

2 Classical Computing

The standard unit of information in classical computing is the bit. A bit can take the binary value 0 or 1, and is represented by a two state physical system, like a switch or a 2-level current. Bits can be combined indefinitely into patterns like 0001100 and 01011000 to encode an arbitrary amount of information, and as such, form the basis of letters, numbers, and strings in modern computing.
We also have classical gates, which are ways of manipulating bits to perform computations. For example, the "exclusive-or" gate, denoted by the symbol $\oplus$, performs the following operations:

$$0 \oplus 0 = 0$$

$$1 \oplus 0 = 1$$

$$1 \oplus 1 = 1$$

However, classical computing is not perfect. Due to external noise and interference, along with imprecise hardware, our bits can suffer from computational errors. For example, consider a situation in which we wish to send a bit through a channel, to our friend on the other side. However, this channel is not ideal. For every bit we send through the channel, the output is "flipped" with probability p, and is unchanged with probability 1-p. What this means is that if we send a bit, say, 0, through this channel, then our friend receives the bit 1 with probability p, and 0 with probability 1-p. This channel essentially renders all the information we send useless, because the person on the other side will have no way of detecting whether an error occured or not, and thus will never know the true value of the bit we sent. To combat this problem, we introduce the following *encoding* scheme.
If, instead of sending bits as we have done previously, we send three copies for each bit we want to send, then the following mapping arises:

$$0 \mapsto 000$$

$$1 \mapsto 111$$

which known as a repetition code. Now, with the repetition code, if a bit-flip happens on the second bit, we have the following transformation:

$$000 \mapsto 010$$

and our friend receives the bits 010. Now, our friend can simply look at what value the majority of the bits agree with, in this case being 0, and take that value as the intended one. This mapping is

$$010 \mapsto 0$$

You may note that if two or three bit-flips occur, the value our friend takes will be incorrect, but since $p(1-p)^2 < p^2(1-p)$ for $p < 1$ and the probability of a bit-flip is strictly less than one, it is much less likely for us to have two or more bit flips than just one. As such, we can be confident knowing that the value our friend takes will be more likely to be the correct answer.
As we mentioned before, this is a very simple error-correcting procedure, but it lays the groundwork for more complicated schemes that we will discuss later on. Let us now introduce the language of Dirac Notation.

3 Dirac Notation

To understand quantum computing, one first must be familiar with Dirac Notation. Dirac Notation provides a useful way to describe quantum states and operations, and was invented by Paul Dirac in 1939. The basis of Dirac Notation is the bra and the ket, which represent one dimensional row and column vectors, respectively. Looking past their funky names, a ket $|x\rangle$ belonging to a Hilbert Space $\mathcal{H}$ can be described as

$$|x\rangle = \sum_i v_i \, |\phi_i\rangle$$

where the set of kets $\{|\phi_i\rangle\}$ represents an orthogonal basis for $\mathcal{H}$, and the coefficients v_i are the "amplitudes" of each basis. It's important to note that the character inside the $|\rangle$ is simply a label for the ket, and is not indicative of the values the ket stores. This will be important to remember later when representing quantum basis states. Kets can also be represented in the so called "matrix-formulation," as follows:

$$|x\rangle = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_m \end{bmatrix} \tag{1}$$

which may provide an easier way of understanding kets to those unfamiliar with Dirac notation.

The counterpart to the ket is the bra. For every ket $|x\rangle$ there exists a corresponding bra $\langle x|$ where

$$|x\rangle^\dagger = \langle x| \, .$$

The operator $\dagger$ is known as the "Hermitian Conjugate," which for the ket $|x\rangle$ from Equation 2 has the following effect in vector notation:

$$|x\rangle^\dagger = [x_1^* \; x_2^* \; \dots x_m^*]$$

where $*$ is the complex conjugate operator.
The inner product of a bra and a ket, written $\langle x|y\rangle$, where $\langle x|$ and $|y\rangle$ belong to the space $\mathcal{H}$, is defined as the mapping $\mathcal{H} \mapsto \mathbb{C}$, where $\mathbb{C}$ is the set of complex numbers. The inner product is similar to the dot product in linear algebra, and can also be written in the equation:

$$\langle x|y\rangle = \sum_i x_i^* y_i$$

where x_i and y_i are the i'th elements of $\langle x|$ and $|y\rangle$, respectively.

The Tensor Product, also known as the Kronecker Product, is a way of combining spaces, operators, or vectors. Suppose we have two Hilbert Spaces, $\mathcal{H}_1$ and $\mathcal{H}_2$, with dimension n and m, respectively. The tensor product $\mathcal{H}_1 \otimes \mathcal{H}_2$ of these spaces will be a new, larger Hilbert space with dimension $n \times m$. The same rules of dimensional multiplication apply to the tensor products of vectors and operators. In matrix representation, the tensor product of two vectors can be seen as follows:

$$\begin{pmatrix} x_1 \\ x_1 \end{pmatrix} \otimes \begin{pmatrix} y_1 \\ y_2 \end{pmatrix} = \begin{pmatrix} x_1 y_1 \\ x_1 y_2 \\ x_2 y_1 \\ x_2 y_2 \end{pmatrix} \tag{2}$$

When writing the tensor product of two kets $|\psi\rangle$ and $|\phi\rangle$, for example, the $\otimes$ is often omitted, leaving just $|\psi\rangle\,|\phi\rangle$, and most often it will be written simply as $|\psi\phi\rangle$.
Now, with a grasp on the fundamentals of Dirac Notation, we can discuss the basics of quantum computing.

4 The Basics of Quantum Computing

In Section 1, we saw how classical states could be represented as either 1's or 0's. The reason that quantum computing is so powerful is that states can be represented as not just 1's and 0's, but as *linear combinations* of 1's and 0's. In fact, a general quantum state appears in the form $|\psi\rangle = \alpha\,|0\rangle + \beta\,|1\rangle$, where α and β are known as the *amplitudes* of the 1 and 0 states in $|\psi\rangle$, respectively, and satisfy the equation

$$||\alpha||^2 + ||\beta||^2 = 1 \tag{3}$$

The states $|0\rangle$ and $|1\rangle$, known as the *computational basis states,* can be represented in vector form as:

$$|0\rangle = \begin{pmatrix} 1 \\ 0 \end{pmatrix} |1\rangle = \begin{pmatrix} 0 \\ 1 \end{pmatrix}$$

Thus, the state from Equation 3 can be represented as:

$$\begin{pmatrix} \alpha \\ \beta \end{pmatrix}$$

These states are encoded in *qubits,* which are the quantum equivalent of bits and serve the same purpose of storing state information. These ideas lead quite nicely into the first postulate of quantum computing.

Postulate 4.1. (State Space Postulate) The state of a quantum system is described by a unit vector in a Hilbert Space $\mathcal{H}$.

For our purposes, a Hilbert space can be regarded just as a special type of complex vector space. The notion that a quantum state can exist almost in a "limbo" between the states 1 and 0 is known as *superposition*, and is one of the defining characteristics of quantum computing. We expect that it may take a while to become fully acquainted with this idea, and this is fine. Quantum computing itself exists in a sort of limbo between understanding and intuiting the core ideals, and being able to handle the raw math.
Now, this notation suffices to describe static quantum states, but the quantum states we will look at will almost always be changing. And before we look at how quantum states change, we first need

to look at operators.
An operator U is a mathematical object that has the following effect on a state vector $|\psi\rangle$:

$$U|\psi\rangle = |\psi\rangle'$$

In simple terms, an operator transforms a qubit from one state to another. Single-qubit operators are really just 2×2 matrices, although they are rarely written as such. An important type of operator for quantum computing is the family of *unitary* operators.

Definition 4.1. (Unitary Operator) An operator U is unitary if $U^\dagger = U^{-1}$, where U^{-1} is the inverse of U.

Now, with this definition, we can see how operators play a role in quantum computing, given the following postulate:

Postulate 4.2. (Evolution Postulate) The evolution of a closed quantum system is described by a unitary operator.

This postulate implies that for any transformation of a quantum system from a state $|\psi_1\rangle$ to the state $|\psi_2\rangle$, there exists a unitary operator U such that:

$$U|\psi_1\rangle = |\psi_2\rangle .$$

which just means that we will be using operators to describe how quantum states change.
We say that two operators A and B commute if they satisfy the following relation:

$$A(B|\psi\rangle) = B(A|\psi\rangle) \forall |\psi\rangle$$

and they anti-commute if:

$$AB|\psi\rangle = -BA|\psi\rangle$$

An important set of operators is the Pauli group $\mathcal{P}$, which contains the following operators:

$$I = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \quad X = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$$

$$Z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix} Y = \begin{pmatrix} 0 & -i \\ -i & 0 \end{pmatrix}$$

These operators act on the basis states $|0\rangle$ and $|1\rangle$ as follows:

$$X|0\rangle = |1\rangle , X|1\rangle = |0\rangle$$

$$Z|0\rangle = |0\rangle , Z|1\rangle = -|1\rangle$$

One of the reasons that the Pauli operators are so special is that they span the space of 2x2 matrices. That is, any 2x2 operator U can be written as:

$$U = c_1 I + c_2 X + c_3 Z + c_4 Z,$$

a result that will be important for error correction later on.
Another important gate is known as the Hadamard gate, H. It has the matrix representation

$$\frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}$$

and acts on the basis states as follows:

$$H|0\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$$

$$H|1\rangle = \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle)$$

We can see that the Hadamard Gate puts the basis states $|1\rangle$ and $|0\rangle$ into superpositions of themselves, which are known as $|+\rangle$ and $|-\rangle$, respectively.
So far, we have only looked at single qubit states. Let us now see how the states of larger systems can be represented.

Postulate 4.3. (Composition of Systems Postulate) When two quantum systems in states $|\psi_1\rangle$ and $|\psi_2\rangle$ are combined, their combined state can be represented as

$$|\psi_1\rangle \otimes |\psi_2\rangle$$

All that this postulate means is that the state of a multi-qubit system can be written as the tensor product of the states of the qubits.
However, it is important to note that the state of a 2-qubit system cannot *always* be written as a tensor product of each state. When this is true, the composite state is called *entangled.* For example, the state

$$|\psi\rangle = \frac{|00\rangle + |11\rangle}{\sqrt{2}}$$

is entangled, because there is no way to factor out either of the states of the two-qubits.
We can also have operators acting on multi-qubit systems in the following way: Suppose we have a two-qubit system $|\psi\rangle|\phi\rangle$ and we want to apply the X gate to the first qubit and the Z gate to the second qubit. This is equivalent to applying the operator

$$X \otimes Z$$

to the entire system. In general, if we wish to transform a two-qubit system $|\psi\rangle|\phi\rangle$ with the operators A and B, the transformation is described by:

$$(A \otimes B)(|\psi\rangle \otimes |\phi\rangle) = A|\psi\rangle \otimes B|\phi\rangle .$$

An example of a two-qubit gate is the CNOT gate, which is essentially a quantum equivalent of the classical $\oplus$ gate from earlier. The CNOT gate acts on basis state pairs in the following way:

$$CNOT|00\rangle = |00\rangle , CNOT|01\rangle = |01\rangle , CNOT|10\rangle = |11\rangle , CNOT|11\rangle = |10\rangle$$

where we assume that the first qubit is the control qubit, and the second qubit is the target qubit. The CNOT gate flips the value of the target qubit if the control qubit is in state $|1\rangle$, and leaves the target qubit unchanged if the control qubit is in state $|0\rangle$.
Now, let us introduce the concept of measurement in quantum computing. Measurement is much more important in quantum computing than classical computing, as it has the ability to actually affect quantum states, which we will see. Let us know define measurement with the following postulate:

Postulate 4.4. (Measurement Postulate) Given an orthonormal basis $B = \{|\varphi_i\rangle\}$ and a state $|\psi\rangle$ belonging to the same state space $\mathcal{H}$, we can perform a measurement on $|\psi\rangle$ with respect to the basis B such that measuring

$$|\psi\rangle = \sum_i \alpha_i |\varphi_i\rangle$$

outputs the label i with probability $|\alpha_i|^2$ and leaves the system in state $|\varphi_i\rangle$. For example, in the case of the Hadamard-induced state $|+\rangle$ from earlier, we can measure this state and expect to receive the value 0 with probability $(\frac{1}{\sqrt{2}})^2 = \frac{1}{2}$ and the value 1 with probability also $\frac{1}{2}$. After our measurement, the state will collapse into either $|0\rangle$ or $|1\rangle$, depending on what we measured.
From this, we can see that measuring a quantum system actually has an effect on the system itself. The state of the system before the measurement is different from the state after measurement, which will be a very important part of quantum computing.

5 Mixed States and the Density Matrix

Until this point we have only discussed pure states. Pure states work well when describing error-free quantum computations, but often fail to accurately represent more complicated quantum systems. For this, we now introduce the concept of mixed states. A mixed state, also called an "ensemble" of pure states, is represented by the notation below:

$$\{(|\psi_1\rangle, p_1), (|\psi_2\rangle, p_2), \dots (|\psi_m\rangle, p_m)\} \tag{4}$$

where p_i is the probability of the system being in state $|\psi_i\rangle$. As such, the density matrix can be seen as a combination of classical and quantum probability, with the classical aspect stemming from the p_i probabilities and the quantum from the probabilities behind superposition. It is important to note that a mixed state like the one above does not represent a superposition of pure states, but a classical probability distribution of them. The motivation behind using this notation might be unclear initially, but it will become clearer as we discuss quantum error correction.

Another tool that will become useful later on is the density matrix. A density matrix ρ for a general mixed state is defined as the sum

$$\rho = \sum_i p_i |\psi_i\rangle \langle\psi_i|$$

For example, the density matrix of the ensemble $\{(|0\rangle, \frac{1}{2}), (|1\rangle, \frac{1}{2})\}$ is

$$\frac{1}{2}|0\rangle\langle 0| + \frac{1}{2}|1\rangle\langle 1|$$

We also find that this density matrix is equivalent to that of the pure state $\frac{1}{\sqrt{2}}|0\rangle + \frac{1}{\sqrt{2}}|1\rangle$ which is

$$\left(\frac{1}{\sqrt{2}}|0\rangle + \frac{1}{\sqrt{2}}|1\rangle\right)\left(\frac{1}{\sqrt{2}}\langle 0| + \frac{1}{\sqrt{2}}\langle 1|\right) = \frac{1}{2}|0\rangle\langle 0| + \frac{1}{2}|1\rangle\langle 1|$$

As such, it's important to note that density matrices are not unique, and that multiple states can have the same density matrix.

Applying an operator U to a mixed state such as the one in Equation 4 results in the state

$$\{(U|\psi_1\rangle, p_1), (U|\psi_2\rangle, p_2), \dots (U|\psi_m\rangle, p_m)\}$$

with density matrix

$$\sum_i p_i U |\psi_i\rangle \langle\psi_i| U^\dagger = U \left(\sum_i p_i |\psi_i\rangle \langle\psi_i| \right) U^\dagger = U\rho U^\dagger$$

One of the benefits of the density matrix formulation is the ability to describe subsystems of a larger quantum system. For example, consider a 2-qubit entangled state $|\psi\rangle_{AB} \in \mathcal{H}_A \otimes \mathcal{H}_B$. Since this state is entangled, it is not possible to factor out and obtain the state vector for the first qubit $|\psi\rangle_A \in \mathcal{H}_A$. To overcome this challenge, we can describe the first qubit as a mixed state. From there, we can calculate the "reduced density operator" ρ^A for $|\psi\rangle_A$ with the operation known as the *partial trace.* This can be done with the following equation:

$$\rho^A \equiv \mathrm{Tr}_B(\rho^{AB})$$

In this equation, Tr_B is the partial trace over B, and is defined by the equation

$$\mathrm{Tr}_B(|a_1\rangle \langle a_2| \otimes |b_1\rangle \langle b_2| \equiv |a_1\rangle \langle a_2| \mathrm{Tr}(|b_1\rangle \langle b_2|) \tag{5}$$

where a_n and b_n are states in $\mathcal{H}^A$ and $\mathcal{H}^B$, respectively. Remembering that

$$\mathrm{Tr}(|b_1\rangle \langle b_2|) = \langle b_2|b_1\rangle ,$$

Equation 5 simplifies to

$$\mathrm{Tr}_B(|a_1\rangle \langle a_2| \otimes |b_1\rangle \langle b_2|) = \langle b_2|b_1\rangle |a_1\rangle \langle a_2| ,$$

which is a result we will use later on to discuss quantum errors.
Now that we have seen how ensembles of quantum states can be represented, we can begin to discuss quantum error correction.

6 Quantum Error Correction

We have now covered an overview of the basics of quantum computing, and will now begin to discuss quantum errors and how we can correct them. But first, let us briefly attempt to motivate the ideas behind quantum error correction.
Quantum computers have the potential to be incredibly powerful. The fundamental features of quantum systems, i. e., entanglement and superposition, give quantum computers an immense edge over their classical counterparts. For example, Grover's algorithm, a quantum search algorithm, works in time complexity $O(N^{\frac{1}{2}})$, which is a quadratic speedup over classical sort algorithms. Shor's algorithm, which can factor an arbitrary number in $O(\log(N)^3)$, provides almost an exponential increase over classical factoring algorithms.
Yet while quantum computing yields great promise, many of these super-fast algorithms have yet to be implemented. This is because of the difficulties presented by building large quantum computers. In fact, the biggest quantum computer at the time this paper is written only contains 127 qubits. This is because quantum computers are extremely sensitive to external interference, and thus require incredibly demanding quantum hardware. It is also very difficult to maintain quantum states, as they collapse in milliseconds without stabilizing procedures. As such, quantum computers are very prone to error, and require heavy error-correcting protocols to function in any way. Thus, the importance of quantum error correction cannot be overstated, as it holds the key to the future of quantum computing. However, quantum error correction is far from easy. In fact, there are three main challenges to quantum error correction that we now present:

1. It is impossible to clone an arbitrary quantum state. This is known as the No-Cloning Theorem, and prevents the implementation of repetition-codes in quantum computing.

   Theorem 6.1 (No-Cloning Theorem)**.** *There is no unitary operator U that performs the following operation on an arbitrary choice of $|\psi\rangle$*

   $$U|\psi\rangle|\phi\rangle \mapsto |\psi\rangle|\psi\rangle, \tag{6}$$

   where $|\psi\rangle$ is some fixed ancilla state.

   We can prove this statement using the fact that unitary operators preserve the inner products of kets. That is, if:

   $$A|x\rangle = |m\rangle$$
   $$A|y\rangle = |n\rangle$$

   for a unitary operator A, then

   $$\langle x|y\rangle = \langle m|n\rangle.$$

   Now, to prove the No-Cloning theorem, we first assume that there exists a unitary operator U_c that can perform the operation given in Equation 6. Then, for the unique, non-orthogonal states $|\psi\rangle$ and $|\psi\rangle$, and the normalized ancilla state $|s\rangle$, we have:

   $$U_c|\psi\rangle|s\rangle = |\psi\rangle|\psi\rangle$$

   $$U_c|\phi\rangle|s\rangle = |\phi\rangle|\phi\rangle$$

   Since U_c is a unitary operator and thus must preserve the inner product, we have

   $$(\langle\psi|\langle s|)(|\phi\rangle|s\rangle) = (\langle\psi|\langle\psi|)(|\phi\rangle|\phi\rangle)$$

   $$\langle\psi|\phi\rangle\langle s|s\rangle = \langle\psi|\phi\rangle\langle\psi|\phi\rangle$$

   Since $|s\rangle$ is normalized and the inner product of a normalized vector with itself is always 1, we have:

   $$\langle\psi|\phi\rangle = (\langle\psi|\phi\rangle)^2$$

   which can only happen when $\langle\psi|\phi\rangle$ is equal to 1 or 0. For $\langle\psi|\phi\rangle$ to be 1 or 0, either $|\psi\rangle$ and $|\psi\rangle$ must be orthogonal, which we said they were not, or they must be the same ket, which we also clarified was not true. Thus, there cannot exist an operator U_c that can clone an arbitrary state, and the No-Cloning Theorem has been proved.

2. Measuring quantum state causes them to collapse and lose the information they hold. This means that it's impossible to check for errors midway through a computation without disturbing the computation itself.

3. Qubits experience continuous errors. Quantum errors come not only in the form of full bit and phase flips, but also in angular shifts of the qubit by a continuous range of values. Qubits can also experience phase errors, which have the following mapping: $|0\rangle \mapsto |0\rangle$ and $|1\rangle \mapsto -|1\rangle$, which adds another challenging aspect to correcting quantum errors.

7 Coherent and Decoherent Errors

In the last section, we discussed the main problems that quantum error correction faces. Now we introduce a few examples of quantum errors and how they can effect a system in unique ways.

Consider a qubit initialized in state $|0\rangle$ to which we wish to apply the identity operation N times. If the system is error-free, then the outcome of this computation will be the state

$$|\psi\rangle' = \prod_{i}^{N} I_i |0\rangle = |0\rangle$$

Upon measuring this output in the computational basis, we will receive the state $|0\rangle$ every time. As such, the probability of error is given by $p_{error} = 0$. However, if in our quantum computer we have designed the identity gate such that instead of leaving a state unchanged, it introduces a slight rotation of the input, our qubit evolves to the state

$$|\psi\rangle' = \prod_{i}^{N} e^{i\epsilon X} |\psi\rangle = cos(N\epsilon) |0\rangle + i \sin(N\epsilon) |1\rangle$$

after the computation. When we measure this state in the computational basis, we will receive the following states with probabilities:

$$P(|0\rangle) = \cos(n\epsilon)^2 \approx 1 - (N\epsilon)^2$$

$$P(|1\rangle) = \sin(n\epsilon)^2 \approx (N\epsilon)^2$$

As such, the probability of an error appearing in this computation is $(N\epsilon)^2$. This is an example of a coherent error. Coherent errors often arise from incorrect knowledge of how a system works, and can be compounded through repeated applications of a faulty gate. In quantum computing, there also exist decoherent errors, which are responsible for the collapse of superpositions. Decoherent errors often result from unwanted interactions with a system's environment. Consider again a qubit initialized to the state $|0\rangle$. Suppose we wish to apply the series of gates HIH to this qubit. In an error-free system, this computation would result in the following:

$$\begin{aligned} HIH |0\rangle &= H \frac{1}{\sqrt{2}} (|0\rangle + |1\rangle) \\ &= |0\rangle \end{aligned} \tag{7}$$

where the identity gate I represents a wait-stage in the computation. Now, consider the simple environment $|E\rangle$ with the following properties: First, the environment is a 2-level quantum system, just like our qubit in state $|1\rangle$. As such, this environment has two basis states, $|E_0\rangle$ and $|E_1\rangle$. Second, when $|E\rangle$ interacts with a qubit in state $|1\rangle$, a process known as "coupling," the state of the environment is flipped, while if the qubit is in state $|0\rangle$, nothing changes. Lastly, assume that this "flipping" interaction only happens in the wait stage(I gate) of a computation. Now, let us re-examine the computation from Equation 7, but with the qubit coupled to the environment in

state $|E_0\rangle$. The computation proceeds as follows:

$$\begin{aligned} HIH\,|0\rangle\,|E\rangle &= HI(\frac{1}{\sqrt{2}}(|0\rangle + |1\rangle))\,|E_0\rangle \\ &= H * I(\frac{1}{\sqrt{2}}(|0\rangle\,|E_0\rangle + |1\rangle\,|E_0\rangle)) \\ &= H(\frac{1}{\sqrt{2}}(|0\rangle\,|E_0\rangle + |1\rangle\,|E_1\rangle) \\ &= \frac{1}{2}(|0\rangle + |1\rangle)\,|E_0\rangle + \frac{1}{2}(|0\rangle - |1\rangle)\,|E_1\rangle \end{aligned} \tag{8}$$

We can describe this state with density matrix:

$$\rho = (\frac{1}{2}(|0\rangle + |1\rangle)\,|E_0\rangle + \frac{1}{2}(|0\rangle - |1\rangle)\,|E_1\rangle)(\frac{1}{2}(\langle 0| + \langle 1|)\,\langle E_0| + \frac{1}{2}(\langle 0| - \langle 1|)\,\langle E_1|)$$

Expanding out, we have:

$$\begin{aligned} \rho = &\frac{1}{4}(|0\rangle\,\langle 0| + |0\rangle\,\langle 1| + |1\rangle\,\langle 0| + |1\rangle\,\langle 1|)\,|E_0\rangle\,\langle E_0| \\ &+ \frac{1}{4}(|0\rangle\,\langle 0| - |0\rangle\,\langle 1| - |1\rangle\,\langle 0| + |1\rangle\,\langle 1|)\,|E_1\rangle\,\langle E_1| \\ &+ \frac{1}{4}(|0\rangle\,\langle 0| - |0\rangle\,\langle 1| + |1\rangle\,\langle 0| - |1\rangle\,\langle 1|)\,|E_0\rangle\,\langle E_1| \\ &+ \frac{1}{4}(|0\rangle\,\langle 0| + |0\rangle\,\langle 1| - |1\rangle\,\langle 0| - |1\rangle\,\langle 1|)\,|E_1\rangle\,\langle E_0| \end{aligned} \tag{9}$$

We can trace over the environment to obtain the reduced density operator for our qubit:

$$\begin{aligned} \rho_{re} = \mathrm{Tr}_E(\rho) = &\langle E_0|E_0\rangle\,\mathrm{Tr}(\frac{1}{4}(|0\rangle\,\langle 0| + |0\rangle\,\langle 1| + |1\rangle\,\langle 0| + |1\rangle\,\langle 1|)) \\ &+ \langle E_1|E_1\rangle\,\mathrm{Tr}(\frac{1}{4}(|0\rangle\,\langle 0| - |0\rangle\,\langle 1| - |1\rangle\,\langle 0| + |1\rangle\,\langle 1|)) \\ &+ \langle E_1|E_0\rangle\,\mathrm{Tr}(\frac{1}{4}(|0\rangle\,\langle 0| - |0\rangle\,\langle 1| + |1\rangle\,\langle 0| - |1\rangle\,\langle 1|)) \\ &+ \langle E_0|E_1\rangle\,\mathrm{Tr}(\frac{1}{4}(|0\rangle\,\langle 0| + |0\rangle\,\langle 1| - |1\rangle\,\langle 0| - |1\rangle\,\langle 1|)) \end{aligned} \tag{10}$$

Remembering that $|E_0\rangle$ and $|E_1\rangle$ are basis states and must satisfy the following:

$$\langle E_i|E_j\rangle = \delta_{ij},$$

Equation 10 simplifies to:

$$\begin{aligned} \rho_{re} &= \frac{1}{4}(|0\rangle\,\langle 0| + |0\rangle\,\langle 1| + |1\rangle\,\langle 0| + |1\rangle\,\langle 1| \\ &+ \frac{1}{4}(|0\rangle\,\langle 0| - |0\rangle\,\langle 1| - |1\rangle\,\langle 0| + |1\rangle\,\langle 1| \\ &= \frac{1}{2}(|0\rangle\,\langle 0| + |1\rangle\,\langle 1|) \end{aligned} \tag{11}$$

Thus, the qubit will be measured in the $|0\rangle$ state 50% of the time, and the $|1\rangle$ state 50% of the time. This final state is a complete classical mixture of the basis states, rather than a quantum

superposition. We also see that the second Hadamard gate, which we thought would transform the state $\frac{1}{\sqrt{2}}(|0\rangle + |1\rangle) \mapsto |0\rangle$, had no effect in the computation. Decoherent errors like these cause quantum states to lose superposition, or the "coherence" between the $|0\rangle$ and $|1\rangle$ states, and become classical systems. This is not good, as it will interfere with many of the quantum algorithms that use superposition to their advantage.

8 The Error Model

In the last section, we saw two examples of quantum errors. Let us now formulate a general way of describing quantum errors. Suppose that a qubit is in state $|\psi\rangle$, and its environment is in state $|E\rangle$. When a general error operator U_{err} acts on this system, it results in the state $U_{err}\,|\psi\rangle\,|E\rangle$, described by the density matrix

$$\rho = U_{err}\,|\psi\rangle\,|E\rangle\,\langle E|\,\langle\psi|\,U_{err}^{\dagger}$$

The state of the environment isn't important to us, so we trace it out to obtain:

$$\mathrm{Tr}(\rho) = \mathrm{Tr}(U_{err}\,|\psi\rangle\,|E\rangle\,\langle E|\,\langle\psi|\,U_{err}^{\dagger}) = \sum_i \mathcal{E}_i\,|\psi\rangle\,\langle\psi|\,\mathcal{E}_i^{\dagger}$$

where the coefficients $\mathcal{E}_i$ are error operators acting on the system. An intuitive way to understand this equation is by thinking of the terms $\mathcal{E}_i\,|\psi\rangle\,\langle\psi|\,\mathcal{E}_i^{\dagger}$ as the different outcomes of possible errors acting on $|\psi\rangle$. When these terms are summed together, they represent a classical distribution of the errors that can occur on $|\psi\rangle$.

Now let's examine the case of the classical bit-flip channel from before. By replacing the bits with qubits and the bit-flips with applications of the X gate, we can obtain a quantum equivalent of the bit-flip channel. If we wish to transmit a state $|\psi\rangle$ through the channel, it will be received in state $X|\psi\rangle$ with probability p and remain in the state $|\psi\rangle$ with probability 1-p. Thus, we can describe the state of the qubit after transmission with the density matrix

$$\rho = (1-p)\,|\psi\rangle\,\langle\psi| + pX\,|\psi\rangle\,\langle\psi|\,X$$

with corresponding error operators

$$\mathcal{E}_0 = \sqrt{1-p}\,\mathbb{I}$$
$$\mathcal{E}_1 = \sqrt{p}\,X$$

Now that we have discussed a general formalism of quantum error models, we can begin to find ways of correcting these errors.

9 Encoding and Recovery Operators

One of, if not the most important step to quantum error correction is the process of encoding. Encoding is a way of protecting quantum information so that errors can be detected and fixed later on in the computation. An encoding operator U_{enc} acts on a register of qubits $|\psi\rangle\,|000...0\rangle$ as follows:

$$U_{enc}\,|\psi\rangle\,|000...0\rangle = |\psi_{enc}\rangle$$

thereby taking an initial quantum state and a set of ancilla qubits and producing the encoded state $|\psi_{enc}\rangle$. This operation can be thought of as embedding the state $|\psi\rangle$ into a higher dimensional Hilbert space, or equivalently as spreading the knowledge contained on a single qubit onto many

highly entangled ancilla qubits. A central task of quantum error correction is finding the best U_{enc} for a desired error model, which we will see more of later in the section.

Suppose we have a state $|\psi\rangle$ that we encode and subject to general error U_{err}. The outcome of this operation will be a state with density matrix

$$\sum_i \mathcal{E}_i |\psi_{enc}\rangle \langle\psi_{enc}| \mathcal{E}_i^\dagger$$

In general, we will not always be able to retrieve the original state of the qubit by simply decoding it with $U_{enc}^\dagger$(the inverse of U_{enc}). That is, in most cases,

$$\mathrm{Tr}_{anc}\left[U_{enc}^\dagger \left(\sum_i \mathcal{E}_i |\psi_{enc}\rangle \langle\psi_{enc}| \mathcal{E}_i^\dagger\right) U_{enc})\right] \neq |\psi\rangle \langle\psi|$$

To correct the errors on the encoded qubit $|\psi_{enc}\rangle$, we will need the help of *recovery operators.* An operator $\mathcal{R}$ is said to *correct* an error U_{err} if it satisfies the following:

$$\mathrm{Tr}_{anc}\left[\sum_j \mathcal{R}_j \left(U_{enc}^\dagger \left(\sum_i \mathcal{E}_i |\psi_{enc}\rangle \langle\psi_{enc}| \mathcal{E}_i^\dagger\right) U_{enc}\right) \mathcal{R}_j^\dagger\right] = |\psi\rangle \langle\psi| \tag{12}$$

That is, it reverses the effect U_{err} has on the affected qubit.

Applying an encoding operator U_{enc} to the basis states $|0\rangle$ and $|1\rangle$ produces the "codewords" $|0_{enc}\rangle$ and $|1_{enc}\rangle$. For a code to be helpful, a recovery operator $\mathcal{R}$ must exist that satisfies Equation 12 for the codewords $|0_{enc}\rangle$ and $|1_{enc}\rangle$. For such an $\mathcal{R}$ to exist, the error operators must satisfy the equation

$$\langle\phi_h| \mathcal{E}_i^\dagger \mathcal{E}_j |\phi_k\rangle = c_{ij}\delta_{hk} \tag{13}$$

where ϕ_h and ϕ_k belong to the set of codewords $\mathcal{C}$. Equation 13 dictates when errors can be corrected, and when they cannot. It implies that the codewords must remain orthogonal after undergoing error, and that the error must scale them by the same amount. Together, these conditions ensure that the relative coefficients of qubits are not disturbed, and that the codeword a corrupted state resulted from can be determined easily. Another important aspect of Equation 13 is that if it is satisfied by a set of correctable errors $\mathcal{E}_i$, then it is also satisfied by any linear combination of errors from $\mathcal{E}_i$. This means that linear combinations of correctable errors are also correctable! And as we mentioned in Section 4, any unitary operator can be written as a combination of the Pauli operators I, X, Y, and Z. So, if we can create an error correcting code that corrects on I, X, Y, and Z errors, we will be able to correct *any* arbitrary single qubit error. This result makes quantum error correction as a whole much less daunting, as it means that it suffices to only look at a finite set of errors, even when trying to correct continuous errors.

10 The 3 Bit Code

Now we are ready to look at our first quantum error correcting code, the 3 Bit Code. The 3 Bit Code was first proposed in 1985 by Asher Peres, an Israeli physicist who worked at the Israel Institute of Technology. This code can correct single bit-flip errors, and, while not the most powerful code, provides a robust introduction to the broad field of quantum error correcting codes.

To begin, recall the quantum bit-flip channel from Section 8. It is important to remember that the effect of the channel was to leave the transmitted state $|\psi\rangle$ unchanged with probability 1-p, and

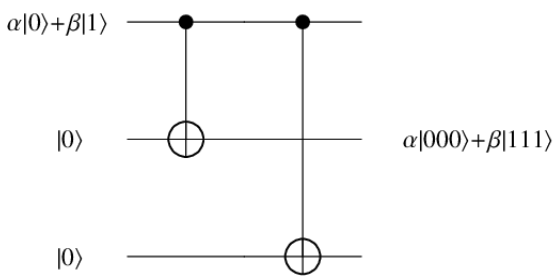

Figure 1: A circuit for encoding a state with the 3-qubit code from researchgate.net, Todd A. Brun, *Quantum Error Correction.*

to transform it to the state $X|\psi\rangle$ with probability p. If we try to send an arbitrary state through the channel with no encoding process, there is no way to tell if an error occurred without measuring the qubit and thus collapsing the information it contains. Let us now encode the state $|\psi\rangle$ before sending it through the channel. The encoding operation acts on the basis states $|0\rangle$ and $|1\rangle$ to produce the states

$$|0_L\rangle = |000\rangle$$
$$|1_L\rangle = |111\rangle$$

Due to the linearity of quantum physics, this encoding operator acts on some arbitrary superposition $|\psi\rangle = \alpha\,|0\rangle + \beta\,|1\rangle$to produce the state $|\psi_{enc}\rangle = \alpha\,|000\rangle + \beta\,|111\rangle$. This can be accomplished through the use of two CNOT gates, as shown in Figure 1.

If we assume that the bit-flip channel acts on at most one qubit at a time, we can send the encoded state $|\psi_{enc}\rangle$ through the channel to receive one of four distinct outcomes, summarized by the table below:

Flipped Qubit	Final State
None	$\alpha\,\|000\rangle + \beta\,\|111\rangle$
Qubit 1	$\alpha\,\|100\rangle + \beta\,\|011\rangle$
Qubit 2	$\alpha\,\|010\rangle + \beta\,\|101\rangle$
Qubit 3	$\alpha\,\|001\rangle + \beta\,\|110\rangle$

At this point, we can see that the task of correcting a single bit-flip reduces to finding where the error occurred and applying an X gate to the affected qubit. We can accomplish this through the use of *syndrome measurements*. A syndrome measurement is a way of extracting information about an error without directly measuring the effected qubits. In the case of the 3 Bit Code, we can introduce two ancilla bits which will store the computed parities of the three-qubit block. The parity measurements tell whether each qubit agrees with one another by comparing the arrangement of 1's, and can be computed with two sets of two CNOT gates, as seen in diagram . For example, The parity of the state when no error occurs is calculated by adding the first 0 in the qubit with the second 0(modulo 2), and the first 0 with the third. In both cases, the result is 0, so the ancilla bits are left in the joint state $|00\rangle$. When a bit-flip error occurs on the second qubit and the parities are computed, the 0 in the first qubit is added to the 1 in the second qubit to produce 1, and the 0 in the first qubit is added to the 0 in the third qubit to produce the value 0. In this case, the the ancilla register is left in state $|10\rangle$ These parity measurements give us information about where the bit-flip occurred, and the full list of outcomes, along with the corrective protocol needed, is displayed in the table below.

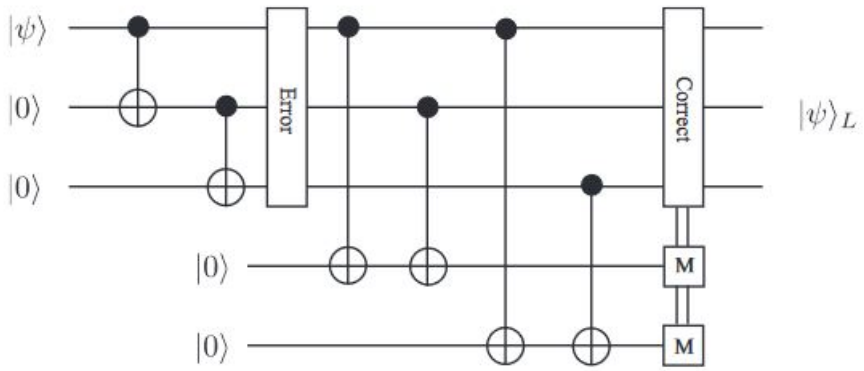

Figure 2: A circuit for computing the parities of the 3-qubit code and applying a classically controlled error recovery gate from arXiv.org, Simon J Devitt, *Quantum Error Correction for Beginners.*

Final State	Ancilla Values	Recovery Operation
$\alpha\|000\rangle + \beta\|111\rangle$	00	None
$\alpha\|100\rangle + \beta\|011\rangle$	11	X on Qubit 1
$\alpha\|010\rangle + \beta\|101\rangle$	10	X on Qubit 2
$\alpha\|001\rangle + \beta\|110\rangle$	01	X on Qubit 3

Thus, we can use these parity measurements to determine where an error occurred, and apply the X gate to correct it. This is known as a classical controlled recovery operation, and the diagram for the whole circuit is shown below.

An important part of quantum error correction to consider is the existence of phase errors. While phase errors do not have a classical counterpart, they are still relatively easy to correct on qubits. We now turn to the phase-flip channel, which is much like the bit-flip channel, but with applications of the Z gate instead of the X gate. As such, a qubit in state $|\psi\rangle$ that is subjected to the phase-flip channel remains in state $|\psi\rangle$ with probability 1-p, and is transformed to the state $Z|\psi\rangle$ with probability p. This state of the qubit after being transmitted is represented by the density matrix

$$\rho = (1-p)\,|\psi\rangle\langle\psi| + pZ\,|\psi\rangle\langle\psi|\,Z$$

This density matrix looks very similar to the one for the bit-flip channel. As such, we can adapt the encoding scheme for the bit-flip channel as follows. First, consider the Hadamard Basis states

$$|+\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$$

$$|-\rangle = \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle)$$

The effect of the Z gate on the $|+\rangle$ state is to transform it into $|-\rangle$, and vice versa. We can see that the Z gate has the same effect on the Hadamard basis states as the X gate does on the computational basis states, namely, flipping each state to the other. Thus, for the phase-flip channel we encode $|0\rangle$ as $|+++\rangle$ and $|1\rangle$ as $|---\rangle$. A general superposition for this scheme is encoded as:

$$\alpha\,|0\rangle + \beta\,|1\rangle \mapsto \alpha\,|+++\rangle + \beta\,|---\rangle$$

Now, the operators for error detection and recovery are performed exactly as they were for the bit-flip channel, but with respect to the Hadamard basis instead of the computational basis. This involves changing every CNOT gate to a controlled-Z gate, and implementing a three qubit Hadamard gate after the encoding for the bit-flip code. It will also be useful to consider the idea of the "phase parity" of a product of the $|+\rangle$ and $|1\rangle$ states. Phase parity can be defined as whether the number of $|-\rangle$'s in the product is even or odd.

11 Stabilizer Formalism

So far, we have been describing codes through the state vector representation of qubits. This method tends to become very inefficient as codes grow in complexity, as circuits and state representations differ with each code. As such, we desire a concise formalism for error correction and code construction that doesn't depend on the particular code we are using. We will now introduce the idea of stabilizer states and codes.
An operator P is said to *stabilize* a state $|\psi\rangle$ if the state $|\psi\rangle$ is an eigenstate of P with eigenvalue +1. This definition can be seen mathematically through the equation:

$$P|\psi\rangle = |\psi\rangle$$

In other words, applying the stabiliser P to $|\psi\rangle$ leave the state unchanged. For example, the Pauli operator Z stabilises the state $|0\rangle$, as

$$Z|0\rangle = |0\rangle$$

Now, let us revisit the Pauli group $\mathcal{P}$, which is defined as the set of single-qubit operators

$$\mathcal{P} = \{X, Y, Z, I\}$$

The Pauli group establishes a basis of 2-dimensional operators. $\mathcal{P}$ also forms a group under multiplication. The Pauli group can be extended to N-qubits by considering the N-fold tensor of the Pauli group, denoted

$$\mathcal{P}_N = \mathcal{P}^{\otimes N} = \underbrace{\mathcal{P} \otimes \mathcal{P} \otimes \ldots \mathcal{P}}_{N\ times}$$

As a reminder, the tensor product of two operators A and B is defined as the mapping

$$(A \otimes B)(|\psi\rangle\,|\phi\rangle) \mapsto A|\psi\rangle \otimes B|\phi\rangle$$

for the states $|\psi\rangle$ and $|\psi\rangle$.

Definition 11.1 (Stabilizer State)**.** An N-qubit state $|\psi\rangle_N$ is known as a stabilizer state if there exists a subgroup S of the N-manifold Pauli group $\mathcal{P}^N$ such that

$$A|\psi\rangle = |\psi\rangle\ \forall\ A \in S$$

Alternatively, a stabilizer state $|\psi\rangle$ can be specified by the set of "generators" $\mathcal{G}$ where

$$\mathcal{G} = \{K_i : K_i|\psi\rangle = |\psi\rangle, \forall(i,j)\} \tag{14}$$

Another important property of the generators of a stabilizer state is that every operator commutes with each other. In other words:

$$[K_i, K_j] = 0 :\ \forall\,(i,j)$$

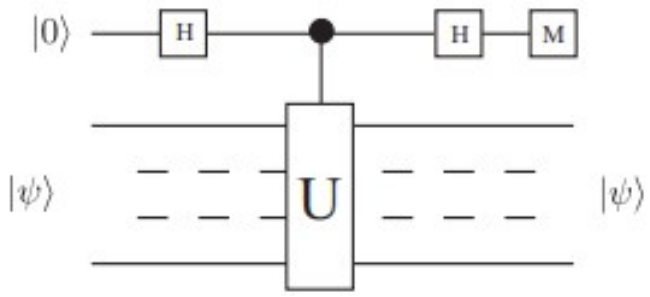

Figure 3: A circuit for projecting a state $|\psi\rangle$ into a +1 eigenstate of operator U from arXiv.org, Simon J Devitt, *Quantum Error Correction for Beginners.*.

This property is necessary for the stabilizers to be measured simultaneously, which we will see later on. An example stabilizer state is the three qubit state

$$|\psi\rangle = \frac{|000\rangle + |111\rangle}{\sqrt{2}}$$

which is also known as the GreenbergerHorne-Zeilinger (GHZ) state. This state can be defined by the following generators:

$$K_1 = X \otimes X \otimes X,$$
$$K_2 = Z \otimes Z \otimes I,$$
$$K_3 = I \otimes Z \otimes Z$$

as they all leave the GHZ state unchanged.

The use of stabilizer states will become evident in the next section as we discuss stabilizer codes and how they can be used for correcting quantum errors.

12 Error Correction with Stabilizer Codes

To begin correcting quantum errors with stabilizer codes, we first need to know how to prepare an arbitrary state. This preparation involves projecting the state into a +1 eigenstate of all the generators, which we need to do so that we can identify and correct errors later on in the procedure.

The procedure for projecting an arbitrary state into a +1 eigenstate of a desired operator U is shown in Figure 3.

As we can see from the diagram, our state $|\psi\rangle$ that we want to prepare is first initialized, along with an ancilla qubit in state $|0\rangle$. Then, a Hadamard Gate is applied to our ancilla qubit, which is then used for a controlled-U operation on $|\psi\rangle$. At this point in the circuit, the state of our system is

$$|\psi_{next}\rangle = \frac{1}{\sqrt{2}}(|0\rangle\,|\psi\rangle + |1\rangle\,U\,|\psi\rangle)$$

Then, we apply another Hadamard gate to the ancilla qubit. After this gate, the state of the entire system is

$$|\psi_{enc}\rangle = \frac{1}{2}(|\psi\rangle + U\,|\psi\rangle)\,|0\rangle + \frac{1}{2}(|\psi\rangle - U\,|\psi\rangle)\,|1\rangle$$

Now, we measure the ancilla qubit in the computational basis. If the qubit is in state $|0\rangle$, the output of our circuit becomes

$$|\psi_{enc}\rangle = |\psi\rangle + U|\psi\rangle$$

We can show that this is a +1 eigenstate of U by calculating U$|\psi_{enc}\rangle$:

$$U|\psi_{enc}\rangle = U(|\psi\rangle + U|\psi\rangle) = U|\psi\rangle + UU|\psi\rangle . \tag{15}$$

Since U is unitary and Hermitian, we have that UU = 1 and Equation 15 simplifies to

$$U|\psi_{enc}\rangle = U|\psi\rangle + |\psi\rangle = |\psi_{enc}\rangle$$

Thus, our encoded state is a +1 eigenstate of U. If the result of measuring our ancilla qubit is $|1\rangle$, then our system becomes the state

$$|\psi_{enc}\rangle = |\psi\rangle - U|\psi\rangle ,$$

a -1 eigenstate of U which we can show in the same way as before. As such, we have shown that the circuit projects an arbitrary state into a ± 1 eigenstate of U, which we can turn into all +1 states with classically controlled Pauli gates. Now, consider an error E acting on an arbitrary encoded state $|\psi_{enc}\rangle$. If E is a combination of Z and/or X errors(and thus an element of the Pauli group) then the erred state $E|\psi_{enc}\rangle$ satisfies the following:

$$K_i E|\psi_{enc}\rangle = (-1)^m E K_i|\psi_{enc}\rangle = (-1)^m E|\psi_{enc}\rangle \tag{16}$$

where K_i is an element of the stabilizer group and m is a variable such that m=0 if E and K_i commute and m=1 if E and K_i anti-commute. We know that those are the only two options for E and K_i because they are both members of the Pauli group and all Pauli operators either commute or anti-commute. The implications from Equation 16 are that if E and K_i commute, then the state remains in a +1 eigenstate of K_i, but if E and K_i anti-commute, then the state is flipped to a -1 eigenstate of K_i. Furthermore, E will commute and anti-commute with a unique set of stabilizers. To see this property in action, let us now examine the 7-qubit code.

The 7-qubit code was discovered in 1996 by Andrew Steane, and corrects for any single qubit error. The generators of the 7-qubit code are:

$$\begin{aligned} &K_1 = IIIXXXX,\ K_2 = XIXIXIX, \\ &K_3 = IXXIIXX, \quad K_4 = IIIZZZZ \\ &K_5 = ZIZIZIZ, \quad K_6 = IZZIIZZ. \end{aligned} \tag{17}$$

If we encode a state with the 7-qubit code and an X error occurs on the first qubit, then the error will commute with the generator K_5, and anti-commute with K_4 and K_6. Similarly, an X error on the second qubit will commute with K_6 and anti-commute with K_4 and K_5, and an X error on the third qubit will commute with K_4 and anti-commute with K_5 and K_6. For X errors on qubits 4-7 there also exists a unique set of commuting and anti-commuting stabilizers, and if no error occurs then the state will commute with all generators(remember that any valid codestate is a +1 eigenstate of all stabilizers.) If a Z error were to occur on any of the qubits, then a unique combination of stabilizers K_1, K_2 and K_3 would be able to identify the position of the error. Also, remember that a Y error corresponds to a Z and an X error on the same qubit, which could be computed sequentially. As such, after determining where and what type of error has occurred on our state, we can implement a classically controlled recovery operation to fix it. It is important to note that the 7-qubit code is

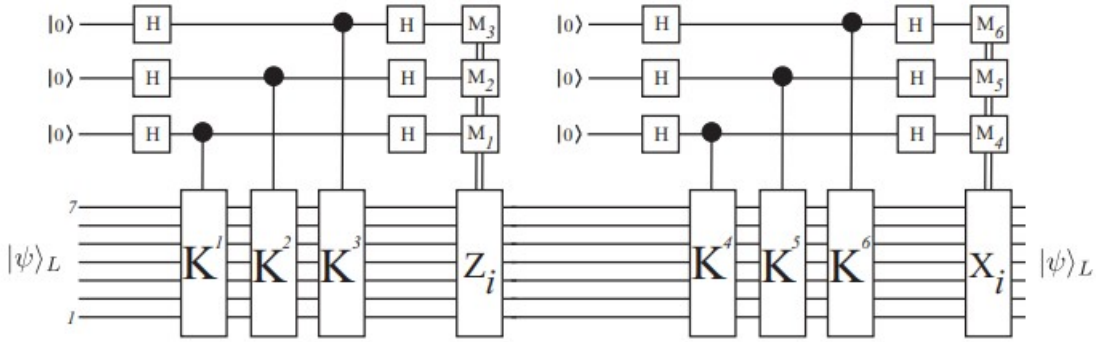

Figure 4: An implementation of error correction with the 7-qubit stabilizer code. The first block corrects for Z errors, while the second block corrects for X errors. From arXiv.org, Simon J Devitt, *Quantum Error Correction for Beginners.*

not the most compact code to protect against arbitrary single qubit errors. There exists a 5-qubit code defined by the following set of stabilizers:

$$X \otimes Z \otimes Z \otimes X \otimes I$$

$$I \otimes X \otimes Z \otimes Z \otimes X$$

$$X \otimes I \otimes X \otimes Z \otimes Z$$

$$Z \otimes X \otimes I \otimes X \otimes Z$$

However, no code that uses less than 5 qubits can protect against a general single qubit error.

We have now shown an example of how stabilizer codes can be used to correct arbitrary quantum errors. The importance of the commuting and anti-commuting relations of operators cannot be overstated, as it is key to detecting and fixing errors of any type. As we have now addressed the procedures for correcting single qubit gate-induced errors, We will now move on to the idea of "fault tolerant" quantum computing.

13 Quantum Fault Tolerance

So far, we have looked mostly at error correcting schemes for resting quantum states(i. e., no gates are being applied). Let us now look at how error works in a general *circuit*. For a circuit with S gates, where the probability of each gate introducing an error is given by p, the probability of at least one error occurring in the circuit is at most Sp, and the expected number of errors in the output is Sp. If an error happens in the first part of a circuit, this error can *propagate* through the circuit, leading to unrecoverable errors by the end of the circuit. By unrecoverable errors, we mean errors that don't satisfy the restraints given by Equation 13. Now, suppose that we want to design a quantum error-correcting procedure to protect our states from these propagated, unrecoverable errors. We might try encoding the qubits before the circuit, and then, once we need to apply a gate, we could decode the qubits, apply the gate, and then re-encode the qubits. However, this sort of procedure only protects qubits from errors *in between* gates, not from errors that the gates themselves can introduce. As such, we can ditch this type of error-correction protocol and instead

design gates that can act directly on encoded states. In fact, we want to design a series of gates *fault-tolerantly*, meaning that error is not allowed to propagate, and if an error happens on one gate, it doesn't completely ruin the rest of the computation.

Definition 13.1. Suppose we have a quantum gate that introduces an error in the output with probability p. A fault-tolerant implementation of this gate introduces an unrecoverable error with probability cp^2 for some constant c.

It is important to note that our fault-tolerant upper bound cp^2 is only an improvement over the non fault-tolerant gate provided that $p < \frac{1}{c}$. This condition is known as the *threshold condition* and $\frac{1}{c}$ is called the *threshold error probability.*
In summary, to achieve our goal of fault-tolerant quantum computing, we first choose a suitable quantum code, and then design implementations for:

1. *fault tolerant universal set of gates.* We would like to design gates that can act directly on encoded states, where the probability of an unrecoverable error in the output is less than cp^2 for a constant c.

2. *fault tolerant measurements.* We would also like to design measurement procedures that introduce errors with a probability less than cp^2.

For a circuit with S gates, we can devise a fault-tolerant implementation of this circuit using an encoding procedure and fault tolerant gates and measurements such that the probability of an error in the output of the circuit is at most Scp^2 for a constant c. This is an improvement over Sp as long as $p < \frac{1}{c}$, which we have seen earlier.
We will now look at how codes can be *concatenated* to achieve even further reduced error probabilities.

14 Code Concatenation and the Threshold Theorem

Using the techniques from the previous section, we can design a circuit of size S(the number of gates) whose overall probability of error is less than or equal to cSp^2, where c is a constant. In this section, we will show how codes can be combined for even further improvement. We will use this idea to obtain a bound on the number of gates required to implement a circuit with error probability error less than a desired threshold.

The idea of concatenating quantum codes is quite simple. In *first-level encoding*, which includes all the codes we have discussed so far, we encode each qubit with a code of our choice. Then, for *second-level encoding*, we simply encode each qubit in the codeword with the same code as before. Thus, we see that after two levels of encoding with an n-qubit code, every qubit in the initial register is ultimately encoded by n^2 qubits. This process of encoding multiple times is known as *concatenation,* and it works well as long as the error model at each level of encoding is of the same form.

Suppose we have a circuit that is prone to error with probability p. We have shown that we can encode this circuit fault-tolerantly such that the error is reduced to cp^2. Now, if we simply encode this circuit again with the same code, the error is further reduced to $c(cp^2)^2 = c^3p^4$. After k levels of encoding, the probability of error is reduced to $\frac{(cp)^{2^k}}{c}$, which is an improvement as long as $p < \frac{1}{c}$. The number of gates required to implement each fault-tolerant gate(after k levels of concatenation) is d^k for some constant d.

If we can obtain a bound on d^k, we can show that the error rate decreases faster than the size of the circuit grows, which would effectively justify the use of quantum error correction. Suppose we wish to create a fault-tolerant implementation of a quantum circuit with S gates, and we want the total error probability for the circuit to be less than ϵ. For us to do this successfully, each gate must have an error probability under $\frac{\epsilon}{S}$. As such, we must concatenate our codes k times such that

$$\frac{(cp)^{2^k}}{c} \leq \frac{\epsilon}{S}$$

There exists such a k as long as p is below the threshold $\frac{1}{c}$. We now can multiply by c, take the logarithm of both sides and rearrange to obtain

$$2^k \leq \frac{\log(\frac{S}{c\epsilon})}{\log(\frac{1}{cp})}$$

Substituting $2 = d^{1/\log_2 d}$, we get:

$$\begin{aligned} d^k &\leq \left(\frac{\log(\frac{S}{c\epsilon})}{\log(\frac{1}{cp})}\right)^{\log_2 d} \\ &\in O\left(\log^m\left(\frac{S}{\epsilon}\right)\right), \end{aligned} \tag{18}$$

where m is a constant greater than or equal to 1. Therefore, a fault-tolerant circuit with S gates concatenated to k levels is bounded by the number of gates

$$Sd^k = O\left(S\left(\log^m\left(\frac{S}{\epsilon}\right)\right)\right)$$

As such, we have the following *threshold theorem* for quantum computing.

Theorem 14.1 (Quantum Threshold Theorem)**.** *A quantum circuit of S gates can be built with a probability of error below ϵ with*

$$O\left(S\left(\log^m\left(\frac{S}{\epsilon}\right)\right)\right)$$

gates on hardware that each introduce error with probability p less than a constant threshold, and given reasonable information about the noise in the hardware.

This theorem essentially tells us that if we can build quantum hardware such that the error per gate is below a fixed value, we will be able to perform arbitrarily long quantum computations using a polynomial amount of resources. In other words, the theorem says that noise and imprecision of physical devices will not prevent the creation of a quantum computer. It shows that the quantum model of computing is robust, and backs the notion that quantum computers will be stronger than their classical counterparts. Although the type of hardware that the Threshold Theorem requires is quite demanding at this time in quantum history, the Threshold Theorem gives us confidence that a working, full-sized quantum computer *can* be built. As better error-correcting codes are found and scientists continue to improve quantum hardware, a future where quantum computers can solve the world's greatest problems draws near.

References

[DMN13] Simon J Devitt, William J Munro, and Kae Nemoto. Quantum error correction for beginners. *Reports on Progress in Physics*, 76(7):076001, jun 2013.

[KLM06] Phillip Kaye, Raymond Laflamme, and Michele Mosca. *An introduction to quantum computing*. OUP Oxford, 2006.

[KLM06] [DMN13]