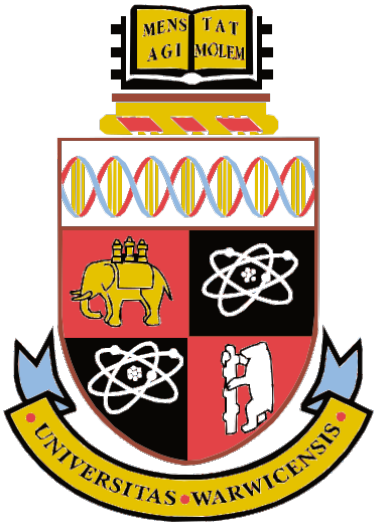

Quantum Error Correcting Codes

by

Rowan Moorsom

MA4K8 Scholarly Report

Submitted to The University of Warwick

Mathematics Institute

April, 2022

Contents

1 Introduction

> When it come to quantum physics, I prefer to do my own research rather than trust the experts
>
> Welcome to Night Vale

Since thez were first proposed, quantum computers have been touted as a more powerful alternative to classical computers, with uses such as cryptography, running logisitcal algorithms more quickly, and modelling physical processes that rely on many variables. However, physical implementations of quantum computers are susceptible to noise and errors from many sources, from the inherent uncertainty of quantum mechanics to unavoidable interactions with the environment, which can affect logic gates, transmission and stored data This means that methods of protecting against or correcting mistakes are essential for a system that is indeed more powerful than a classical computer to be usable. In this report I will be discussing various aspects of error-correcting in quantum computing, beginning overviews of quantum mechanics, the basic functioning of quantum computers, and classical error correction. I will then discuss how error correction is carried out in quantum computing systems, and how codes for this purpose are constructed. Finally, I will give an explanation of one practical application of quantum error correcting codes - fault tolerant quantum computing.

2 A Background in Quantum Mechanics

2.1 A Brief Historical Overview

At the start of the twentieth century, experimental results began to show that a more complex understanding of particles, light and energy was necessary: for example, the double slit experiment which showed that electrons behaved in a wave-like fashion, as well as like particles. This idea of wave-particle duality is one of the key aspects of quantum mechanics: any particle can be described by a wavefunction $\psi(x)$ that describes the *probability* of observing the particle at a given point rather than giving an exact description of its state before measurement (more precisely, $|\psi(x)|^2$ acts as the probability density function). This wavefunction is referred to as the state of the quantum system (the collection of particles or other physical objects being considered). The other key aspect of quantum mechanics is that the energy of a particle does not lie on a continuous spectrum but is rather restricted to a discrete set of values.

2.2 Notation and Mathematical Descriptions

An important notation convention that will be used thorughout this report is Dirac's Bra-Ket notation. This is a method of denoting vectors over a Hilbert space (a complete space

with an inner product [1]); for quantum mechanical purposes $\mathbf{H} = \mathrm{L}^2(\mathbb{C})$ is usually used. A vector ψ in $\mathbf{H}$ is denoted by the *ket* $|\psi\rangle$. The complex conjugate of the vector is denoted by the *bra* $\langle\psi|$ [2], allowing the inner product of the space to be written as $\langle\psi|\phi\rangle$. For discussion of qubits and related systems I will be working in a finite dimensional space. Therefore the relevant inner product is

$$\langle\psi|\phi\rangle = \sum_{i=1}^{i=n} \bar{\psi_i}\phi_i$$

Physical quantities of the system are represented by operators on $\mathbf{H}$, such as position and momentum

$$\text{Position: } X\psi(x) = x\psi(x)$$
$$\text{Momentum: } P\psi(x) = -i\hbar\frac{d\psi}{dx} \tag{1}$$

An operator A can be used to form a linear function with a vector: $\langle\phi| A$ which sends a vector ψ in the space to $\langle\phi|A\psi\rangle$, also written as $\langle\phi|A|\psi\rangle$. The adjoint of the operator is denoted using $A^\dagger$ (instead of the usual A^* from mathematics). 2 vectors in $\mathbf{H}$ can also be used to form a function $|\psi\rangle\langle\phi|$:

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

2.2.1 Density Operators

We can alternatively present the state of a particle using a density operator - a positive-definite hermitian operator with a trace of 1. Rather than a vector, this requires us to view the state as a linear functional ω on the space of operators. Then for every state, there is a density operator ρ so that

$$\omega(A) = \mathrm{Tr}\rho A$$

According to the Riesz representation theorem, this density operator is unique for each state [2]. Additionally, it can be represented as a sum of projectors (operators such that $P^2 = P$) and eigenvalues of the state if the eigenvectors form an orthonormal basis:

$$\rho = \sum_{1}^{n} \lambda_i P_{\varphi_i}$$

where P_{φ_i} is the operator projecting onto the subspace spanned by the eigenvector φ_i defined by $P_{\varphi_i}\alpha = \langle\varphi, \alpha\rangle\varphi$[2]

2.2.2 Operator-Sum Representation

By using density operators, the effect of an operation can also be described using operator-sum representation. Suppose the quantum system and the environment have an inital combined quantum state[1] described by the product of their density operators, $\rho \otimes \epsilon_0$, which is transformed by a unitary operator U. From the final state $U(\rho \otimes \epsilon_0)U^\dagger$ we can find the reduced final density operator

$$\begin{aligned}\rho_f &= \mathrm{Tr}_E[U(\rho \otimes \epsilon_0)U^\dagger] \\ &= \xi(\rho)\end{aligned}$$

ξ is referred to as a superoperator, and its effect can be described as the sum of *operation elements* E_k, also referred to as Kraus operators:

$$\rho_f = \sum_k E_k \rho E_k^\dagger$$

2.3 Measurement

To measure a quantum system, an operator from a collection $\{M_m\}$ is applied to the system. In particular, if a measurement is projective, it can be described by an observable - a hermitian operator which is the sum of the projections onto its possible eigenspaces corresponding to the potential eigenvalues. One important property of this form of measurement is *repeatability*: if the system is measured with the resulting eigenvalue λ, repeating the measurement will give λ again.

2.4 Linear Superposition

In classical mechanics, an object can only be in one state at the same - but quantum superposition means that it is possible for an object to occupy many states at once. This is the idea that allows quantum computers to be so much more powerful, as they can calculate on many different possible states at once, running algorithms on different values in parallel rather than one after the other.

In particular, given possible states $|\psi\rangle$ and $|\varphi\rangle$, the system could take any linear combination of these $a|\psi\rangle + b|\varphi\rangle$ where $a^2 + b^2 = 1$ (of course this can also be applied to more than 2 states). The *amplitudes* a and b are representative of the probability that the system will be in the state $|\psi\rangle$ or $|\varphi\rangle$ when measured. Importantly, this superposition is not just uncertainty over the state - the system is seen as really being in all these states until measurement. Experimentally, this effect can be seen by performing a double slit experiment with single electrons. Without observation, this will create an interference pattern which can only be explained as the possible waveforms of the electron interfering

[1] I will discuss entanglement of a system with its environment in later chapters

with each other.[3]

2.5 The postulates of quantum mechanics

As opposed to the axioms that are fundamental to mathematics, these postulates (sometimes referred to as axioms) are more a collection of guiding principles and standard results for working with quantum systems. [2]

1 Any quantum mechanical system can be described as a unit vector in a Hilbert space.[3]

2 Every physical quantity corresponds to a linear hermitian operator on the Hilbert space.

3 The only outcome of a measurement is one of the eigenvalues of the corresponding operator. If the eigenvectors of an operator A $|\psi_i\rangle$ form an orthonormal basis, we can additionally describe the state

$$|\psi\rangle = \sum_i c_i |\psi_i\rangle \tag{2}$$

with the probability of measuring the eigenvalue a_i being $|c_i|^2$.
The expectation value of the measurement related to A is $\langle\psi|A\psi\rangle$.

4 The time evolution of a system can be described by a unitary operator, which we can derive using Schrödinger's Equation

$$i\hbar\frac{d}{dt}\psi = \hat{H}\psi$$

$\hat{H}$ is the hamiltonian operator which describes the energy as a function of the particle's position and momentum.[6] and depends on the exact system that's being considered.

5 The Hilbert space of a composite quantum system is the tensor product of the component systems, and the overall state can be described as the product of the state of the component systems.

2.6 Quantum Numbers

Typically, a particle, such as the hydrogen atom, is described by using various quantum numbers: the principal quantum number n, the orbital angular-momentum number l and magnetic quantum number m_l. [3]For example, the principal quantum number is a description of the energy level of the atom.[4] When discussing qubits in future chapters,

[2]The number of possible postulates also varies between sources: Hall's Quantum Theory for Mathematicians[4] gives 4, whereas Gaitan's Quantum Error Correction[5] lists 10 so I have presented an amalgamation of those outlined in various sources.

[3]A system in a non-pure state can be described by a non-unit vector; it is sufficient to consider the pure states for our purposes

[4]in hydrogen these are $E_n = -\frac{13.60\text{eV}}{n^2}$

however, the most important number is the *spin*. Experiments such as that by Stern and Gerlach, where an electron beam was passed through a non-uniform magnetic field, showed the beam being split in a way that couldn't be explained by the existing angular momentum and magnetic quantum numbers. We can understand this by considering the electron to be a particle rotating on an axis as well as orbiting a centre.[5] Measuring the spin along the z-axis gives 2 possible values: $S_z = +\frac{1}{2}\hbar$ or $S_z = -\frac{1}{2}\hbar$.These are known as spin-up and spin-down and are commonly used as 0 and 1 in quantum computing.

3 Quantum Computers

3.1 Entanglement and Quantum Tunnelling

2 further aspects of quantum mechanics are important for setting up a quantum computing system: entanglement, and quantum tunnelling.

Entanglement Mathematically, an entangled system is described in the following way:

Consider a quantum system S composed of two subsystems A and B, with Hilbert spaces $\mathbf{H}_\mathrm{A}$ and $\mathbf{H}_\mathrm{B}$. Taking orthonormal bases $\{\ |\ a_i\rangle\ |\ i \in [1...n]\ \}$ and $\{\ |\ b_j\rangle\ |\ j \in [1...m]\ \}$, we can form an orthonormal basis for S by taking all possible tensor products $|a_i\rangle \otimes |b_j\rangle$ of basis vectors of$\mathbf{H}_\mathrm{A}$ and $\mathbf{H}_\mathrm{B}$. Then

$$|\psi\rangle = \sum_{i,j} c_{i,j}(|a_i\rangle \otimes |b_j\rangle)$$

is an entangled state if it cannot be written as the direct product of states in $\mathbf{H}_\mathrm{A}$ and $\mathbf{H}_\mathrm{B}$.

If two or more particles are entangled, the outcomes of measuring an observable on them is correlated: for example, a system could be set up so that two particles will always have the same spin - measuring one as spin-up immediately means that the other is now spin-up too, no matter how far away it is.

Quantum Tunnelling In classical physics, it is impossible for an object to go somewhere it doesn't have enough energy to travel to: for example, giving a ball a small amount of kinetic energy won't be enough for it to travel over a tower block and land on the other side. The tower block acts as a *potential barrier*. But the wave-function description of a particle in quantum physics means that, counterintuitively, a particle could travel through a potential barrier despite its energy being too low. This is due to the fact that the particle's wavefunction can describe a possibility of being beyond the potential barrier. Tunnelling is especially of use in various ion-based qubit modalities (see appendix A)

[5]This demonstrates how important it is to view the electron as *simultaneously* having wave and particle properties

3.2 Qubits and the Bloch Vector

A qubit is any two-level quantum system over a two-dimensional quantum space. It is also possible to have higher dimensional systems, called qutrits for 3 dimensions and qudits for higher.

The space is equipped with a fixed computational basis $|0\rangle, |1\rangle$ which is also represented in vector terms as $\begin{bmatrix}1\\0\end{bmatrix}$ and $\begin{bmatrix}0\\1\end{bmatrix}$ respectively. As seen in the previous chapter, the existence of this basis allows us to write the wavefunction of the qubit as $|\psi\rangle = a|0\rangle + b|1\rangle$, with a and b being normalised and acting as the probability of the qubit being in either state. Usually, $|0\rangle$ corresponds to the "base" level of the 2-level systems being used, such as spin-down or the ground state of an ion.

A convenient visualisation of the qubit is the *Bloch Sphere*.This places all possible pure states of the qubit on the surface of a unit sphere, with $|0\rangle$ at the north pole and $|1\rangle$ opposite at the south pole. Any of its superposition states can then be described in terms of the angle of elevation from the z-axis θ and the angle of its projection to the xy plane with the x axis [7](see 1).

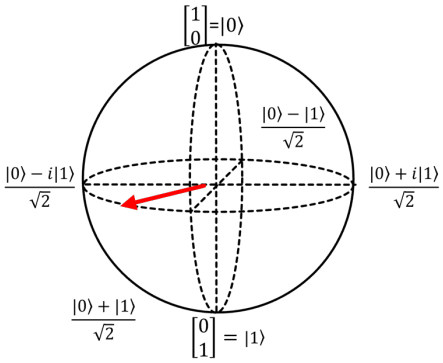

Figure 1: The Bloch vector

3.2.1 Physical construction of qubits

There are several different possibilities for physically constructing the qubits for quantum computing, such as various methods using superconducting materials, utilising trapped ions, and photonics[8],[7]. All these implementations are susceptible to various sources of noise, especially from environmental interference, but their usability in computing is also affected by how scalable they are, which affects how well they can be combined into more powerful devices. More detail about some of these types is included in appendix A.

3.3 Quantum Gates

As in classical computing, quantum programs are broken down to a series of logic gates. In the quantum case, these gates can be described as unitary operators on the state vectors, and represented by matrices. The Pauli Matrices, for example, function as gates. The Hadamard, Phase, CNOT and $\pi/8$ gates are particularly important as they form a universal set of gates which can be used to approxiamte any quantum circuit to arbitrary accuracy. The importance of this will become clearer in the final chapter, where I discuss encoding logic gates for fault tolerance - rather than having to find a way to encode every potential gate, it is only necessary to do this for the universal set.

Gate name	Matrix	Effect
Pauli X gate	$X = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$	The effect of this gate is to change the qubit between states ($\|0\rangle \mapsto \|1\rangle$ and vice versa) - so it is also known as the NOT gate.
Pauli Y gate	$Y = \begin{bmatrix} 0 & -i \\ i & 0 \end{bmatrix}$	This rotates the qubit π radians along the y-axis.
Pauli Z gate	$Z = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}$	This rotates the qubit π radians along the z-axis.
Hadamard gate	$H = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$	This gate is equivalent to rotating the z-axis to the x-axis, and vice versa.
$\pi/8$ gate	$T = \begin{bmatrix} 1 & 0 \\ 0 & e^{i\pi/4} \end{bmatrix}$	This changes the phase of the $\|1\rangle$ state by $e^{i\pi/4}$, rotating the vector around the z axis of the bloch vector.
Phase gate	$P = \begin{bmatrix} 1 & 0 \\ 0 & i \end{bmatrix}$	This is equivalent to T^2, so the effect is to rotate the Bloch vector by $\pi/2$ around the z axis.
Controlled NOT	$CNOT = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix}$	This performs the NOT operation on the target only if the control qubit is in state $\|1\rangle$ Other gates can also be performed in a controlled way, only acting on the target if the control qubit is in the right state.

4 Classical Error Correcting Codes

In classical communication, the primary purpose of error-correcting codes is to prevent corruption of messages due to noise in the communication channel. Often, a system will be subject to noise which may alter the message being sent by altering some of the bits passing through: and clearly if the binary message isn't encoded in any way, any change

of this type could drastically alter what is received at the other end. Therefore, the basic message is converted into a series of codewords which should have the property that, even if 1 or 2 bits in the word are flipped, it can still be interpreted correctly by the receiver. Typically, the codewords are treated as vectors in a linear subspace of $\mathbb{F}_2^n$, where $\mathbb{F}_2$ is the field of 2 elements - for more detail see **??**, where entropy and Shannon's theorem are also discussed.

4.1 Decoding

Once a message has been recieved, there are several possible ways of decoding it, such as the nearest neighbour method mentioned above; I am considering a binary communication channel that has a probability p of switching a bit, independent of any other bit in the message.
$\mathcal{C}$ can be used to partition $\mathbb{F}_2^n$ into cosets $\mathbf{x} + \mathcal{C}$, with a *coset leader* of minimum Hamming weight - the number of 1s in the vector - being chosen to represent each coset. By additionally defining the error syndrome of a message, this allows a different decoding method to be implemented.
The error syndrome of a recieved message $\mathbf{y}$ is $\mathbf{s} = \mathrm{H}\mathbf{y}$. $\mathcal{C}$ is exactly the kernel of the action of H by definition, and so by the First Isomorphism Theorem there is a one-to-one correspondence between the cosets of $\mathcal{C}$ and the possible error syndromes, meaning that the cosets of $\mathcal{C}$can be used to find the most likely decoding for a message.

1 For each syndrome $\mathbf{s}$, pair with the coset leader $\mathbf{e_s}$ of the corresponding coset (as it will be the syndrome of this vector)

2 For $\mathbf{y}$, calculate $H\mathbf{y}$.

3 Decode $\mathbf{y}$ **as** $\mathbf{y} - \mathbf{e_s}$

The error $\mathbf{e}$ is assumed to be $\mathbf{e_s}$ because the error is in the same coset, and it is most probable that it is an element of least weight:

$$\mathbf{e} = \mathbf{x} + \mathbf{y} \therefore \mathbf{e} \in \mathbf{y} + \mathcal{C}$$

This method has the advantage that only 2^{n-k} entries need to be tabulated, making it more memory efficient.

4.2 Hamming Codes

The *Hamming codes* H_r are the $[2^r - 1 =: n, n - r, 3](r \geq 2)$ codes whose parity check matrices have the binary numbers from 1 to $2^r - 1$ as columns. Because of this, $(1, 1, 1, 0 \ldots 0)$ will always be a codeword, so these codes are no more than 3-separated (in fact, d is always exactly 3). H can be rearranged using elementary row operations to the canonical form

$[P|I_{n-k}]$, so that the generating matrix can be represented $\left[\frac{I_k}{P}\right]$:

$$HG = [P|I_{n-k}]\left[\frac{I_k}{P}\right] = P + P = 0$$

Strictly speaking, P in the generating matrix is -P; however over $\mathbb{F}_2$, these are the same.This construction - the standard form, where H is rearranged, can be used for any parity check matrix for a linear code.
As an example of the Hamming code, r = 3: this is a $[7, 4, 3]$ code.[5]

$$\begin{aligned} H &= \begin{bmatrix} 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 \end{bmatrix} \\ &= \begin{bmatrix} 0 & 1 & 1 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 & 0 & 0 & 1 \end{bmatrix} \\ G &= \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 \end{bmatrix}^T \end{aligned}$$

Having rearranged the parity check matrix, it can still be used in the original form with the binary numbers in order, even with code vectors generated from G in its standard form. Suppose our message is $[0110]^T$, which is encoded as $[0110011]^T$, and in transmission the 2nd bit gets flipped. Then the error $H[0010011]^T = [010]^T$, representing 2 in binary. This is a feature of the Hamming codes: if the parity check matrix used is this original form, then the error syndrome will indicate which bit the error occured in.

5 Quantum Error Correcting Codes

Although the aim for quantum error correcting codes is essentially the same as for classical linear codes, there are important diffrences that need to be considered in the quantum case. Firstly, even a basic error model needs to account not only for bit-flip errors, but also for phase-flips ($|1\rangle \to -|1\rangle$). Furthermore, redundancy can't be added just by cloning a bit-string, as shown by the No-Cloning Theorem, and additional care is also necessary in construction and measurement in order to preserve the linear superposition that quantum computing relies on.

5.1 Errors and Noise

5.1.1 Distance Measurements

There are a couple of possible choices for "distance" between quantum states. The following notions of trace distance and fidelity are in fact linked so either can be used, depending on context.

Trace Distance

$$D(\varphi, \psi) = \frac{1}{2} \operatorname{Tr} |\varphi - \psi|$$

In the case of the Bloch sphere, this is simply half the Euclidean distance for a sphere in 3-space. Moreover, the distance can't be increased by applying trace-preserving operations, and so 2 density operators close in this quantum trace distance will have measurement probabilities which are close in the classical trace distance. It is also invariant under unitary operations.

Fidelity

$$F(\varphi, \psi) = \operatorname{Tr} \sqrt{\varphi^{1/2} \psi \varphi^{1/2}}$$

If $|\varphi\rangle$ is a pure state (which is assumed for the scope of this report), then

$$F(|\varphi\rangle, \psi) = \operatorname{Tr} \sqrt{\langle\varphi|\psi|\varphi\rangle |\varphi\rangle \langle\varphi|} = \sqrt{\langle\varphi|\psi|\varphi\rangle}$$

As opposed to the trace distance, the fidelity is non-decreasing, and tends larger as the states become closer together. It is a good descriptor of how much information has been preserved by a quantum channel, by computing $F(|\varphi\rangle, E |\varphi\rangle \langle varphi|)$

Equivalency [6] Suppose $|a\rangle$ and $|b\rangle$ are pure states, and that by the Gram-Schmidt procedure we can find orthonormal $|0\rangle$ and $|1\rangle$ so that $|a\rangle = |0\rangle, |b\rangle = \cos\theta |0\rangle + \sin\theta |1\rangle$, so that the density operators are:

$$|a\rangle = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, |b\rangle = \begin{bmatrix} \cos^2\theta & \cos\theta\sin\theta \\ -\cos\theta\sin\theta & \sin^2\theta \end{bmatrix} D(|a\rangle, |b\rangle) = \frac{1}{2} \operatorname{Tr} \begin{bmatrix} 1-\cos^2\theta & -\cos\theta\sin\theta \\ \cos\theta\sin\theta & -\sin^2\theta \end{bmatrix}$$

$$= |\sin\theta|$$

$$F(|a\rangle, |b\rangle) = \operatorname{Tr} \sqrt{\begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} \cos^2\theta & \cos\theta\sin\theta \\ \cos\theta\sin\theta & \sin^2\theta \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix}}$$

$$= \operatorname{Tr} \sqrt{\begin{bmatrix} \cos^2\theta & 0 \\ 0 & 0 \end{bmatrix}}$$

$$= |\cos\theta|$$

$$D(|a\rangle, |b\rangle) = \sqrt{1 - (F(|a\rangle, |b\rangle))^2}$$

[6] The basic argument is from [9]

5.1.2 Quantum Channels

A quantum channel is typically defined to be a mapping of the inital density operator to the final density operator, as discussed in the previous chapter. The action of the channel can be represented using the Kraus operator in operator sum representation, or by construction of error operators depending on its effect - typically the Kraus operator is simply the error operator scaled by a function of its probability. Physically,one common implementation is done using an optical fiber, where the bit information is transmitted using polarisation - as in the case of photonic qubits, the necessary superposition of states can be transported this way.

5.1.3 Errors and Error Operators

As described previously, the main source of errors during computation is unwanted interaction with the environment, which for example can affect the operation of the qubits or cause the collapse of the wave-function prematurely. Qubits and the environment are able to become entangled; however, some level of environmental interaction is required to enact gates and other operations - for example, by applying a magnetic field - so even if it were possible, total isolation is not desirable.
Working with error operators E, the final state of a code-word is described as $E\left|c\right\rangle$. In this report, I am working with independent and identically distributed errors on a discrete channel, and assuming:
1 Errors occur independently
2 The errors produced by the effects of the Pauli x, y and z matrices are as likely as each other
3 The probability of an error occuring on each qubit is the same
4 The error operators E are unitary and can be represented by a product of the identity matrix and the Pauli matrices acting on the i^{th} qubit
The possible errors can therefore can be described as the *Pauli Group* $\mathcal{G}_n$. This group is generated by four possible operators on each qubit: $\sigma_x^i, \sigma_y^i, \sigma_z^i$ and $I^i = \sigma_0^i$ where the superscript indicates that the matrix is acting on the i^{th} qubit. To make this a group, the products can be multiplied by -1 and $\pm i$, so that an element representing a possible error on n qubits is written

$$e = i^\lambda \otimes \sigma_{j_1}^1 \otimes \ldots \otimes \sigma_{j_n}^n, \lambda \in \{0,1,2,3\}, j_i \in \{0,x,y,z\}$$

Other error models are also used, such as that of the amplitude damping channel. This is easier to describe in terms of its operator-sum representation.

$$E_0 = \begin{bmatrix} 1 & 0 \\ 0 & \sqrt{q-\epsilon^2} \end{bmatrix}, E_1 = \begin{bmatrix} 0 & \epsilon \\ 0 & 0 \end{bmatrix}$$

Then the outcome of applying this channel of the state with density operator $\begin{bmatrix} |a|^2 & ab^* \\ a^*b & |b|^2 \end{bmatrix}$ is:

$$\xi(\rho) = \begin{bmatrix} |a|^2 + \epsilon^2|b|^2 & ab^*\sqrt{1-\epsilon^2} \\ a^*b\sqrt{q-\epsilon^2} & |b|^2(1-\epsilon^2) \end{bmatrix}$$

showing that it has slightly decreased the probability of measuring $|1\rangle$.
The bit-flip and phase-flip channels considered here are modelled as applying the Pauli x and z matrices respectively, with a probability p of affecting a qubit. The operator sum representations are:

$$\text{Pauli X: } E_0 = \sqrt{1-p}I, E_x = \sqrt{p}P_x$$
$$\text{Pauli Z: } E_0 = \sqrt{1-p}I, E_z = \sqrt{p}P_z$$

Combining these effects, along with the Pauli y-matrix, creates the depolaristation channel. This has a probability p of changing the initial state into $\frac{1}{2}I$, which maximises the Von Neumann entropy $S(\rho) = -\operatorname{Tr}(\rho \ln \rho)$.[5] In terms of operator-sum:

$$E_0 = \sqrt{1-\frac{3p}{4}}I, \quad E_1 = \sqrt{\frac{p}{4}}\sigma_x, \quad E_2 = \sqrt{\frac{p}{4}}\sigma_y, \quad E_4 = \sqrt{\frac{p}{4}}\sigma x$$

The fidelity measure defined above is used to calculate how well information is preserved by a channel, by calculating the fidelity between the initial and final states. For example, with the depolarising channel:

$$\begin{aligned} F(|\varphi\rangle, E|\varphi\rangle\langle\varphi|) &= \sqrt{\langle\varphi|(p\frac{I}{2} + (1-p)|\varphi\rangle\langle\varphi|)|\varphi\rangle} \\ &= \sqrt{1-\frac{p}{2}} \end{aligned}$$

5.2 Properties of Quantum Codes

Much like for a classical linear code, quantum data is encoded using a map between spaces. In the case of quantum information, the relevant dimensions for k qubits encoded to n qubits are much higher - let H_2 denote the 2-state space the qubit occupies. Then encoding involves a map ξ from H_n^k to a 2^k dimensional subspace of H_2^n. The Hilbert spaces are given the *computation* or *logical* basis $\{|0\rangle, |1\rangle\}$, with the exact states corresponding to this basis being chosen depending on the particular error set being considered.
There are k basis vectors $\{ \ |\delta_j\rangle \mid j \in [1 \ldots k] \}$ where δ_j can take the values 0 or 1, and these are combined to give the states in H_2^k:

$$\begin{aligned} |\delta\rangle &= |\delta_1 \ldots \delta_k\rangle \\ &= |\delta_1\rangle \otimes \ldots \otimes |\delta_k\rangle \end{aligned}$$

5.2.1 Stabiliser Codes

The code $\mathcal{C}_q$ can be identified with its stabiliser $\mathcal{S} \subset H_2^n$ (the set of elements which fix all vectors $|c\rangle \in \mathcal{C}_q$), which is generated by a set of $n-k$ elements. The eigenvalues of the generators are -1 or +1, and each element of $\mathcal{S}$) can be represented as a unique string in F_2^{n-k}:

$$s = \prod_{i=1}^{i=n-k} g_i^{p_i} \mapsto p_1 p_2 \dots p_{n-k}$$

$\mathcal{C}(l)|l = l_1 l_2 \dots l_n$ is defined as the subspace of H_2^n of eigenvectors for $\{g_i\}$, which all have the eigenvalue $(-1)^{l_i}$ for generator g_i. Evidently, these spaces are all disjoint. By construction, the code vectors $\mathcal{C}_q$ are in $\mathcal{C}(|00\dots0\rangle)$, and in fact the dimensions are equal so the spaces coincide. Another consequence of the use of stabiliser codes is that it allows the possible final states to be described more succinctly. If a qubit is in a state that is in a vector space stabilised by $\mathcal{S}$, then after a unitary gate U is applied the state will be in a vector space stabilised by $U\mathcal{S}U^\dagger$, and this stabiliser is generated by $Ug_iU^\dagger$. Hence rather than the exponential number of potential final states, we can simply use the linearly-sized generating set.

The error operators E map $\mathcal{C}_q$ to another $\mathcal{C}(l)$ - the value of l_i is determined by whether the error operator commutes or anticommutes with the generator g_i. Therefore the error syndrome $E(S)$ of the operator E is defined as the string defining $\mathcal{C}(l)$. The generators can be combined into a parity check matrix H, like those for classical codes, so that $S(E) = HE$.

For a given set of errors $E = \{E_a\}$, the recovery operator of the code $\mathcal{C}_q$ is the quantum operation with operators R_r that finds and corrects the most probable error that has occurred. ($\mathcal{C}_q$, R) is E-error correcting if the following equation - the failure probability for R - is 0 for all E_a

$$\mathcal{E}(\mathcal{C}_q, R, E) = \max_{|c\rangle \in \mathcal{C}_q} \sum_{r,a} |((R_r E_a - \langle c|R_r E_a|c\rangle)\,|c\rangle) \tag{3}$$

This measures the distance from the effect of applying the recovery operator to the actual original code.

Stabilisers also allow measurement without collapsing the state of the system: measuring using a stabilising set places the system into the simultaneuous eigenstate of all the stabilisers, and the stabilisers can be measured without affecting the superposition. [10]

5.2.2 Necessary Conditions

What conditions are necessary and sufficient for a code to be E error correcting? An important consideration for the choice of basis vectors is that the environment should be unable to distinguish them: if this were not the case, then interactions would cause the

superposition to collapse. The basis vectors also need to make sure the images of different states are distuinguishable after being acted on by an error operator. This leads to 2 conditions necessary for designing a code - for all encoded states and all error operators in E:

Orthogonality $\langle i^*|E_a^\dagger E_b|j^*\rangle = 0$

This condition ensures that the different error states can be distinguished, as this is only possible for orthogonal states. (1)

Indistinguishability $\langle i^*|E_a^\dagger E_b|i^*\rangle = \langle j^*|E_a^\dagger E_b|j^*\rangle$

This prevents environmental factors from collapsing the superposition. (2)

In fact, these are also sufficient for a code and its recovery operator R (denoted by $(\mathcal{C}_q, R)$ to be E-error correcting. The proof (from [5]) requires another theorem which gives necessary and sufficient conditions for an error operator E_a to be in the set of correctable errors $E(\mathcal{C}_q, R)$:

Theorem 1. *$E_a \in E(\mathcal{C}_q, R)$ iff over $\mathcal{C}_q, R_r E_a = \gamma_{ra} I \forall R_r \in R$.*

Proof. Of the necessary and sufficient conditions:
$\Rightarrow$: Assume $(\mathcal{C}_q, R)$ is E-error correcting. Using the previous theorem:

$$\begin{aligned}
\langle i^*|E_a^\dagger E_b|j^*\rangle &= \langle i^*|E_a^\dagger I E_b|j^*\rangle \\
&= \sum_r \langle i^*|E_a^\dagger R_r^\dagger R_r E_b|j^*\rangle \\
&= \sum_r \langle i^*|\gamma_{ra}^* \gamma_{rb}|j^*\rangle \\
&= (\sum_r \gamma_{ra}^* \gamma_{rb})\delta_{i^*,j^*} \\
&= \lambda_{a,b}\delta_{i^*,j^*}
\end{aligned}$$

The fourth line comes about from the assumption that $|i^*\rangle$ and $|j^*\rangle$ are orthogonal by construction, and hence we see that their images are also orthogonal, proving that condition (1) holds. Moreover, applying this result to $\langle i^*|E_a^\dagger E_b|i^*\rangle$ and $\langle i^*|E_a^\dagger E_b|j^*\rangle$, we see that the final value only depends on E_a and E_b. Thus both these brakets will give the answer λ_{ab} and condition (2) also holds.
$\Leftarrow$: Assume that we have a code so that conditions (1) and (2) hold. Define $\mathcal{V}_{i^*}$ to be the subspace which is spanned by $\{E_a|i^*\rangle\}$, with an orthonormal basis $\{v_k^{i^*}\}$. Then for $i \neq j$, (1) tells us that $\mathcal{V}_{i^*}$ and $\mathcal{V}_{j^*}$ are orthogonal. If necessary, the orthogonal complement of $\bigoplus_{i^*} \mathcal{V}_{i^*}$ is denoted by $\mathcal{O}$ with an orthonormal basis $|o_m\rangle$. Then $\{|v_k^{i^*}\rangle, |o_m\rangle\}$ is a basis for H_2^n.
Define the recovery operator for $\mathcal{C}_q$ as $R = \{\hat{\mathcal{O}}, R_1, \ldots R_k, \ldots\}$ where $\hat{\mathcal{O}}$ is the projector onto $\mathcal{O}$, and $R_k = \sum_{i^*} |i^*\rangle \langle v_k^{i^*}|$. This is trace preserving, as the states considered are an orthonormal basis. To show that this is the correct recovery operator, we define U_i

mapping $\mathcal{V}_{0^*}$ to $\mathcal{V}_{i^*}$. Consider $|v_\alpha\rangle \in \mathcal{V}_{0^*}$, which U_i maps to $|w_\alpha\rangle$:

$$|v_\alpha\rangle = (\sum_a \alpha_a E_a)\,|0^*\rangle \mapsto (\sum_a \alpha_a E_a)\,|i^*\rangle = |w_\alpha\rangle$$

This is a one-to-one correspondence, sends $E_a\,|0^*\rangle$ to $E_a\,|i^*\rangle$, sends $|v_r^{0^*}\rangle$ to $|v_r^{i^*}\rangle$, and is unitary. The second property is immediately clear, and the third is a simple consequence of the one-to-one and unitary nature.

Suppose $|v_\alpha\rangle$ and $|v_\beta\rangle$ are both mapped onto $|w\rangle$, so that $|w\rangle = \sum_a \alpha_a E_a\,|i^*\rangle = \sum_a \beta_a E_a\,|i^*\rangle$. So we must have $\alpha_a = \beta_a$ for each E_a, and so the two kets were the same. For any $|w\rangle = \sum_a \alpha_a E_a\,|i^*\rangle \in \mathcal{V}_{i^*}$, it is immediately clear that it must be the image of $|v_\alpha\rangle = \sum_a \alpha_a E_a\,|0^*\rangle$ in $\mathcal{V}_{0^*}$.

Let $|w\rangle$ and $|z\rangle$ be the images of $|v_\alpha\rangle = \sum_a \alpha_a E_a\,|0^*\rangle$ and $|v_\beta\rangle = (\sum_b \beta_b E_b)\,|0^*\rangle$ respectively.

$$\begin{aligned}
\langle v_\alpha|U_i^\dagger U_i|v_\beta\rangle &= \langle w|z\rangle \\
&= \sum_{a,b} \alpha_a^*\beta_b\,|i^*|E_b^\dagger E_a|i^*\rangle \\
&= \sum_{a,b} \alpha_a^*\beta_b\,|0^*|E_b^\dagger E_a|0^*\rangle \ \text{(using indistinguishability)} \\
&= \langle v_\alpha|v_\beta\rangle
\end{aligned}$$

It remains to see that $R_r E_a = \gamma_{ar} I$ for each $E_a \in E$ and $R_r \in R$. Now evaluating $E_a\,|c\rangle$:

$$\begin{aligned}
E_a\,|c\rangle &= \sum_{i^*} c_i E_a\,|i^*\rangle \\
&= \sum_{i^*} c_i U_i E_a\,|0^*\rangle \\
&= \sum_{i^*} C_i U_i [\sum_{r'} \beta_{ar'}^{0^*}\,|v_{r'}^{0^*}\rangle \\
&= \sum_{i^*,r'} c_i \beta_{ar'}^{0^*}\,|v_{r'}^{i^*}\rangle
\end{aligned}$$

$$\begin{aligned}
R_r E_a\,|c\rangle &= \sum_{i^*,r'} c_i \beta_{ar'}^{0^*} R_r\,|v_{r'}^{i^*}\rangle \\
&= \sum_{i^*,r'} c_i \beta_{ar'}^{0^*} \sum_{j^*} |j^*\rangle\,\langle v_r^{j^*}|v_{r'}^{i^*}\rangle \ \text{(uisng the definition of } R_r) \\
&= \sum_{i^*} c_{i^*} \beta_{ar}^{0^*}\,|i^*\rangle \\
&= \beta_{ar}^{0^*}\,|c\rangle
\end{aligned}$$

So $R_r E_a$ is indeed equal to some constant scaling of the identity operator. Finally, for $\hat{\mathcal{O}}$, as this projects onto the orthogonal complement of the direct sum of $\mathcal{V}_i$

$$\hat{\mathcal{O}} E_a \left|i^*\right\rangle = 0I$$

and therefore fulfil the conditions of the previous theorem, proving that this code is indeed E-error correcting. □

To summarise, in order to choose an effective basis for the code Hilbert space, the vectors should have the following properties: they need to be orthogonal to each other, at least one should anti-commute with each error operator being considered, and $\langle i^*|E_a^\dagger E_b|i^*\rangle = \langle j^*|E_a^\dagger E_b|j^*\rangle$ for all basis vectors i, j and errors E_a, E_b.
If a code corrects E, then it can also correct linear combinations of elements in E: $F_b = \sum_{a,b} m_{b,a} E_a$. Formally ([9]):

Theorem 2. *Suppose $\mathcal{C}_q, R$ is E-error correcting, and F is an operation with operation elements that are linear combinations of elements of E. Then R can also correct the effects of F.*

Proof. The elements E_a satisfy conditions 1 and 2. Applying R to F_b:

$$\begin{aligned} R_r F_b \left|c\right\rangle &= \sum_a m_{b,a} R_r E_a \left|c\right\rangle \\ &= \sum_a m_{b,a} \beta_{ar}^{0^*} \left|c\right\rangle \\ &= m_{b,r} \beta_{br}^{0^*} \left|c\right\rangle \end{aligned}$$

Therefore the condition of theorem (1) holds and F_b is correctable. □

An immediate consequence of this theorem is that rather than checking if the conditions hold for every potential error being considered, it is enough to find a spanning set for E and focus on developing a code that corrects these specific errors.

5.2.3 Using Entanglement to Correct Errors

One common way of obtaining and using the error syndrome for a particular calculation is by entangling the computer with ancilla qubits. As discussed previously, it is likely that a quantum computer will become entangled with its environment, and this further entanglement counteracts that. By copying a state to an ancilla qubit and performing a measurement there, the superposition of the original qubit is also preserved.

Intial Entanglement Initially the environment E is in state $|e\rangle$ and the computer Q in

state $|\omega\rangle$. The error operators are E_i:

$$|e\rangle |\omega\rangle \rightarrow \sum_i |e_i\rangle \{E_i |\omega\rangle\}$$

Introduction of Ancilla Qubits The ancilla qubits are coupled to Q using a unitary operator C. They are initally in the state $|a_0\rangle$:

$$C[|a_0\rangle \{E_i |\omega\rangle\}] = |i\rangle \{E_i |\omega\rangle\}$$

The $|s\rangle$ are an orthonormal set so can be used to form a basis.

Measurement Finally, the ancilla qubits are measured in the $|i\rangle$ basis to return the error syndrome S. In this basis we obtain a non-entangled pure state

$$|\rho_{pm}\rangle = |e_S\rangle |S\rangle \{E_S |\omega\rangle\}$$

Identifying E_S and applying its inverse to this state gives the final state

$$|e_S\rangle |S\rangle |\omega\rangle$$

In this state the correct operation to reverse errors and obtain the correct intial state can be applied.

5.3 Correcting Bit- and Phase-Flip Errors

On their own, each of these errors is relatively simple to provide redundancy against: if by construction, $|0\rangle$ and $|1\rangle$ are described by orthogonal vectors, then they can be encoded simply by repetition - although this is not the most efficient method. As a demonstration, consider the bit-flip channel (based on teh construction of the phase-flip channel in [5]). An error on the first qubit is described:

$$E_1 = \sigma_x^1 \otimes I^2 \otimes I^3$$

and similarly for the other possibilities. In this instance, the probability of a one-qubit error ($p(1-p)^2$) is much greater than that of a two ($p^2(1-p)$ or more qubit error as we assume p is small; therefore the error syndromes will be assumed to indicate the one-qubit error. As σ_x and σ_x anti-commute, it makes sense to use σ_z to construct the 2 generators.

$$g_1 = \sigma_z^1 \sigma_z^2$$
$$g_2 = \sigma_z^2 \sigma_z^3$$

Now the error syndrome for each expected error can be calculated: if the bit is flipped in qubit 1, then g_1 anti-commutes with this operator, but g_2 commutes, making the syn-

drome 10.

Error	Syndrome
$I^1I^2I^3$	00
$\sigma_x^1I^2I^3$	10
$I^1\sigma_x^2I^3$	11
$I^1I^2\sigma_x^3$	01

Error	Syndrome
$\sigma_x^1\sigma_x^2\sigma_x^3$	00
$\sigma_x^1\sigma_x^2I^3$	10
$\sigma_x^1I^2\sigma_x^3$	11
$I^1\sigma_x^2\sigma_x^3$	01

Figure 2: The error syndromes for the bit-flip code

The first table in 2 shows the more probable error relating to each syndrome, and therefore the one defined as the expected syndrome for that error. The Pauli matrices have the advantage of being self-inverse: the recovery operator for each error simply corresponds to applying the same matrix again.
The stabiliser group $\mathcal{S}$ is $\{I, \sigma_z^1\sigma_z^2, \sigma_z^2\sigma_z^3, \sigma_z^1\sigma_z^3\}$. Using the essential stabilising property of this group,a particular implementation of the computational basis states can be found. The most logical choice is the eigenstates of σ_z corresponding to its two eigenvalues, $+1$ and -1:

$$|0\rangle \equiv |\sigma_z = +1\rangle \ ; \ |1\rangle \equiv |\sigma_z = -1\rangle$$

Then the encoding is done by repetition:

$$|0^*\rangle = |000\rangle, |1^*\rangle = |111\rangle$$

so that the state $a|0\rangle + b|1\rangle$ becomes $a|000\rangle + b|111\rangle$. The parity check matrix will be

$$\begin{bmatrix} 1 & 1 & 0 \\ 0 & 1 & 1 \end{bmatrix}$$

which is analagous to measuring the eigenvalue output of the generating operators, considering our computational basis. The computed error syndromes correspond to those in the table, and so the most probable error can be corrected.

5.4 Caldebank-Shor-Steane Codes

Caldebank-Shor-Steane (CSS) codes are constructed using classical linear codes, producing codes which can correct both bit- and phase-flip errors. The construction uses 2 codes C_1 and C_2 with $C_2 \subset C_1$, of parameters $[n, k_1, d_1]$ and $[n, k_2, d_2]$. It requires C_1 and $C_2^\perp$ to be able to correct t errors. For example, C_1 is the Hamming [7,4,3] code and C_2 is its dual code - this construction is the Steane code.
The construction is based on the cosets created by C_2 in C_1. WIth each coset a codeword is identified:

$$|c_i\rangle = \frac{1}{\sqrt{2^{k_2}}} \sum_{x \in C_2} |c_i + x\rangle \tag{4}$$

Naturally for $|0^*\rangle$ the coset elements are the codewords for C_2. It is then transmitted through a channel that causes both bit- and phase-flip errors, resulting in the final state

$$\frac{1}{\sqrt{2^{k_2}}} \sum_{x \in C_2} (-1)^{(c+x)e_p} |c + x + e_b\rangle \tag{5}$$

For the Steane code:

$$|0^*\rangle = \frac{1}{\sqrt{8}}[|0000000\rangle + |1010101\rangle + |0110011\rangle + |1100110\rangle + |0001111\rangle + |1011010\rangle + |0111100\rangle + |1101001\rangle]$$

$$|1^*\rangle = \frac{1}{\sqrt{8}}[|1111111\rangle + |0101010\rangle + |1001100\rangle + |0011001\rangle + |1110000\rangle + |0100101\rangle + |1000011\rangle + |0010110\rangle]$$

If e_b is 1, this will change $|c + x\rangle$ from 0 to 1 or vice versa, and if e_p and (c+x) are 1, this causes a phase flip. Introducing ancilla qubits to extract error syndromes, detecting and correcting errors is carried out:

Bit-Flip Because all the states $|c + x\rangle$ are in C_1, apply the parity check matrix H_1: $|0\rangle$ represents the appended ancilla qubit, which are typically initally prepared in this state.

$$|c + x + e_b\rangle |zero\rangle \to |c + x + e_b\rangle |H_1(c + x + e_b)\rangle = |c + x + e_b\rangle |H_1 e_b\rangle$$

The ancilla is measured to find $H_1 e_b$, then discarded to return to the original corrupted state. As the classical error syndromes are known for C_1, e_b can therefore be found easily, and applying a NOT gate to each qubit where the error occurs results in

$$\frac{1}{\sqrt{2^{k_2}}} \sum_{x \in C_2} (-1)^{(c+x)e_p} |c+\rangle$$

Phase-Flip The Hadamard gate is applied to each qubit, which swaps x and z and therefore converts the phase difference to a bit difference.

$$\frac{1}{2^{k_2} 2^n} \sum_{v \in \mathbb{F}_2^n} \sum_{x \in C_2} (-1)^{(c+x)\cdot(e_p+v)} |v\rangle$$

Using $w = v + e_p$ this can be rewritten

$$\frac{1}{\sqrt{2^{k_2+n}}} \sum_{v \in \mathbb{F}_2^n} \sum_{x \in C_2} (-1)^{(c+x)\cdot w} |w + e_p\rangle$$

If w is in the dual of C_2, then $w \cdot x = 0$ and $\sum_{x \in C_2} (-1)^{xw} = |C_2|$, and if not then $\sum_{x \in C_2} (-1)^{xw} = 0$. So we obtain

$$\frac{1}{\sqrt{2^n/2^{k_2}}} \sum_{w \in C_2^\perp} (-1)^{c \cdot w} |w + e_p\rangle$$

Now this can be corrected like a bit-flip error, by applying ancilla qubits and using the parity check matrix for $C_2^\perp$ - this is why the condition $C_2^\perp$ is t-error correcting was required. The resulting state is

$$\frac{1}{\sqrt{2^{n-k_2}}} \sum_{w \in C_2^\perp} (-1)^{c \cdot w} |w\rangle$$

The action of the Hadamard gate can then be reversed, obtaining the orignal state. So this code is $[n, k_1 - k_2]$ and can correct t errors.

In the example of the Steane code, the final code is [7,1] and 1-error correcting.

$$\text{Parity Check for step 1: } \begin{bmatrix} 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 1 & 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 & 1 & 0 & 1 \end{bmatrix}$$

$$\text{Parity Check for step 2: } \begin{bmatrix} 0 & 1 & 1 & 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 & 1 \end{bmatrix}$$

As this maps 1 qubit to 7, there are 6 generators for the code subspace, which need to anticommute with all the errors considered between them: the potential errors are σ_z^i and σ_x^i, and additionally the error syndrome for each part should correspond to the number of the qubit it occurs in (like the classical Hamming code)

Z error in:	1	2	3	4	5	6	7
g_1	0	0	0	1	1	1	1
g_2	0	1	1	0	0	1	1
g_3	1	0	1	0	1	0	1

X error in:	1	2	3	4	5	6	7
g_4	0	0	0	1	1	1	1
g_5	0	1	1	0	0	1	1
g_6	1	0	1	0	1	0	1

g_1 can be taken to g_3 as a product of σ_x acting on the qubits with a 1 in the correct row in the table, and g_4 to g_6 the same with σ_z, for example $g_1 = \sigma_x^4 \sigma_x^5 \sigma_x^6 \sigma_x^7$.

5.4.1 A Five-Qubit Code

This is a [5,1,3] code with has 4 generators for the stabiliser, which are cyclic permutations of each other.

$$
\begin{aligned}
g_1 &= \sigma_x^1 \sigma_z^2 \sigma_z^3 \sigma_x^4 I \\
g_2 &= I \sigma_x^2 \sigma_z^3 \sigma_z^4 \sigma_x^5 \\
g_3 &= \sigma_x^1 I \sigma_x^3 \sigma_z^4 \sigma_z^5 \\
g_4 &= \sigma_z^1 \sigma_x^2 I \sigma_x^4 \sigma_z^5
\end{aligned}
$$

So $\mathcal{S}$ is all the possible products of these generators. The basis is taken to be

$$
\begin{aligned}
|0^*\rangle &= \sum_{s \in \mathcal{S}} s \, |00000\rangle \\
|1^*\rangle &= (\sigma_x^1 \sigma_x^2 \ldots \sigma_x^5) \times |0^*\rangle
\end{aligned}
$$

5.5 Surface Codes

Another family of quantum stabiliser codes are the surface codes, which are based on an array of physical qubits. The qubits are divdided into data qubits and measurement qubits, alternating across the array, and are connected to their four nearest neighbours. This means that every measurement qubit is coupled to four data qubits, and every data qubit will be acted on by four measurement qubits (apart from the qubits at the edge of the array, which become important in encoding logic gates later). The measurement qubits are then divided down further, into X and Z measurements corresponding to the Pauli matrices, which act on the data qubits by forcing them to take one or the other of the X or Z eigenstates. The type of measurement alters between rows which results in each data qubit being coupled to 2 X and 2 Z measurement qubits. (see 3)

The X and Z measurements are equivalent to $\sigma_x^1 \sigma_x^2 \sigma_x^3 \sigma_x^4$ or $\sigma_z^1 \sigma_z^2 \sigma_z^3 \sigma_z^4$, labelled according to the data qubits measured. This means that they correspond to measuring the stabilisers formed by four Pauli X or Z matrices. Each qubit in the system is in a state referred to as the *quiescent* state $|\psi\rangle$, which is an eigenstate for both the X and Z measurements simultaneously - something which is possible because the blocks of four operators commute with each other. This also prevents the act of measuring collapsing the quantum state.

When running, the array performs a continuous loop of measurement cycles, which maintains the quiescent state across the whole array. This allows errors to be detected and localised, as they cause the quiescent state to change on single sets of qubits. For example, a Z error on a data qubit would cause the eigenstate measured by the X measurement qubits adjacent to it to change, but not affect any other measurements, which allows the location of the error to be recorded. Similarly, an X error will affect the Z measurements performed by the adjacent qubits, and a Y error will affect all the surrounding measure-

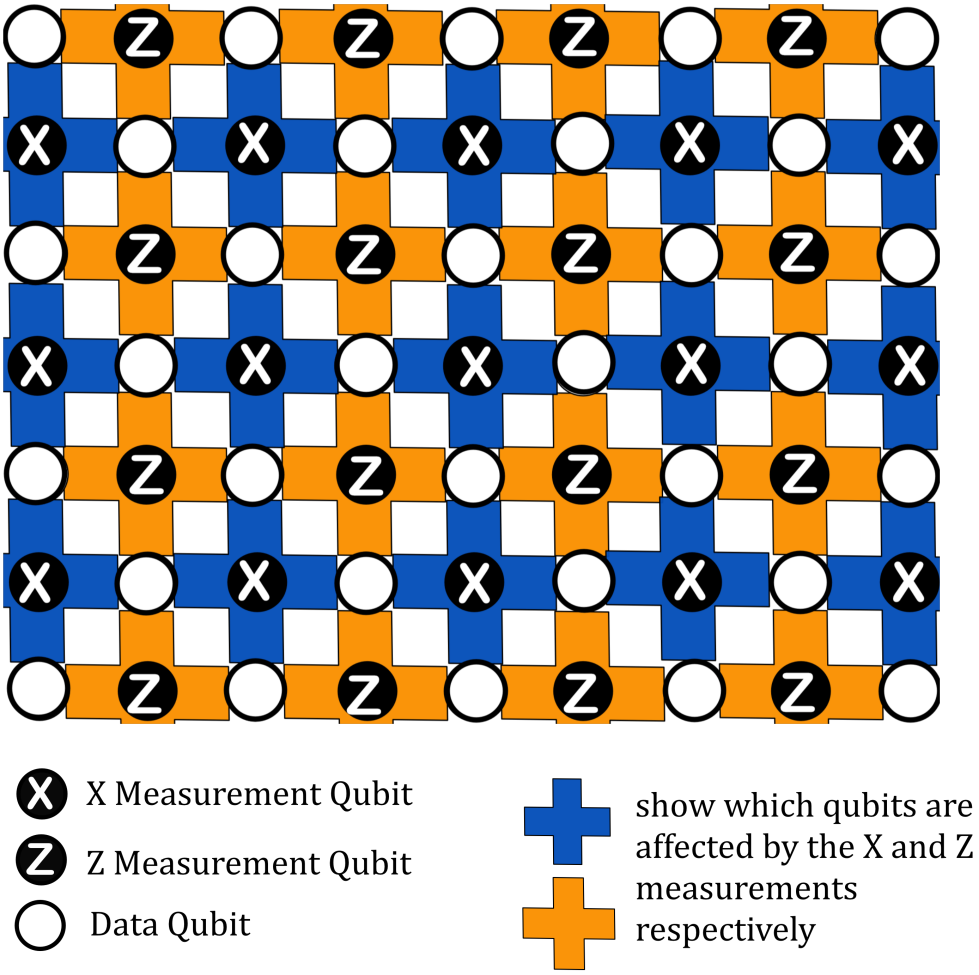

Figure 3: Qubit Array for Surface Code (original diagram, based on [10])

ments.

6 Application: Fault Tolerance

Quantum error correcting codes are not just useful for transmitting data through channels, but can be applied as computations are being done: this protects the data if there is a faulty logic gate in the process, or a storage error. The same codes can be used for all of these purposes. The goal of fault tolerance in designing a computing system is to have a small error probability, which can be achieved as long as the proability of an error in the storage qubits and quantum gates is less than a certain threshold: this is the Accuracy Threshold Theorem, which I will explain in more detail later in this chapter.
As previously discussed, whereas in a classical computer redundancy can be provided by cloning data, this isn't always possible in quantum computing. It is also necessary

to consider that applying measurements at an aribitrary point in a computation would cause the system to collapse into a single state, and that applying an error correcting procedure could actually introduce further errors into the system. Finally, errors caused by faulty gates can propagate because of the use of controlled gates: applying these based on incorrect results will create new errors. For example[9], suppose the initial state of a qubit is $|\varphi\rangle = |0\rangle \otimes [a\,|1\rangle + b\,|0\rangle]$, and the state of the first qubit is changed with probability p to $|1\rangle$. Applying a CNOT gate to the incorrect state $|\varphi_e\rangle$:

$$\begin{aligned} U_{CNOT}\,|\varphi_e\rangle &= |1\rangle \otimes [a\,|0\rangle + b\,|1\rangle] \\ &= \sigma_x^1\sigma_x^2(|0\rangle \otimes [a\,|1\rangle + b\,|0\rangle] \\ &= \sigma_x^1\sigma_x^2 U_{CNOT}\,|\varphi\rangle \end{aligned}$$

Hence this controlled gate has not only maintained the bit-flip error on the first qubit but added a new one on the second qubit. This actually demonstrates a potential probelm with implementing an error-correcting encoding: controlled gates are often used to achieve this, and so can propagate errors in the inital data.
As previously described, a map ξ is used to encode a state into the code space $\mathcal{C}_Q$. This map can also be used on operators: if P is an operator in $\mathcal{G}_k$ acting on the inital state, then $\xi P \xi^\dagger$ is an operator for the code space. More robustly: suppose U is a unitary gate, applied to $|\varphi\rangle$, and $P(U)$ is a sequence of gates. This is an encoding if

$$\forall \text{ superpositions } |\varphi\rangle\,, P(U)\,|\varphi^*\rangle = |U\,|\varphi\rangle\rangle$$

[5]

6.1 Basics of Fault Tolerance

Being encoded alone does not provide fault-tolerance, however: the example of the CNOT gate above is true for applying a coded gate to a coded message as well. A system where an error-correcting code is used to encode data into blocks of qubits is said to be fault tolerant if an error in implementing a single gate or a single storage error causes at most one error in each code block - so evidently if 1 error is converted into 2, the condition is not satisifed. The aim is for this to achieved by using encoded gate and measurement operations on encoded qubit states, which reduces error probabilities, while still computing efficiently. Further reduction of the potential error rate is achieved by concatenating codes - the initial qubits are encoded, and then the qubits this coded state are stored on are also encoded, and so on. For example, Knill [11] describes a C_4/C_6 architecture, where the initial qubits are encoded, forming pairs, using C_4. The qubit pairs are then grouped and encoded using C_6.
In fact, using concatenated codes, it is possible to achieve arbitrary accuracy:

Theorem 3 (Threshold Theorem[12]). *Let $\epsilon > 0$, C a computation. Then there is a*

threshold P_a such that for a quantum circuit Q, it is possible to find a circuit Q' such that if the error rate (from gates and storage errors) is $P_e < P_a$, the function computed by Q' is ϵ-close to that computed by Q

There are several sources for the storage and gate errors that are mentioned in the theorem. Due to entaglement with the environment, it is possible for a storage register of qubits to change the data it is storing. This can be modelled as applying various operators to the data: the identity operator with probaility $1 - p_{stor}$, and a differing one with probability p_{stor}. The qubits may also be prepared in the inital state incorrectly - this is often a highly probabilsitic process, as it is not possible to measure the qubits to check whether they are correct. Gate errors may also occur due to environmental factors, or the physical equipment need to implement them - such as an incorrect magnetic field.
It is important to note that there is always a chance of more than one error occurring, which would overwhelm a system designed to prevent single errors from propagating. This chance increases as the number of qubits used in a system grows, so as quantum computers approach the size needed for practical uses, the methods of fault tolerance will need to be adjusted to the sitaution. In particular, it may be necessary to use codes with a greater error-correcting capability than 1.

6.2 The C_4/C_6 Architecture and Concatenated Codes

6.2.1 Concatenation

Intuitively, combining layers of encoding on qubits will affect the error probability polynomially. It can also increase the distance between codewords.
Following [5]: take an $[n_1, k_1, d_1]$ code C_1, and an $[n_2, k_2, d_2]$ code C_2 - I am assuming they are quantum stabiliser codes with generating sets $G_1 = \{g_i^1 | i = 1 \dots n_1 - k_1\}$ and $G_2 = \{g_j^2 | j = 1 \dots n_2 - k_2\}$. One possible procedure for this maps $k_1 k_2$ qubits into $n_1 n_2$ qubits, which form n_1 blocks of n_2 qubits, and can be extended to create more layers, by taking more qubits initally so that the second layer encoding can be divided into blocks to be encoded by a third code, and so on.

1 Take k_2 sets of k_1 qubits, which C_1 encodes into sets of n_1 qubits to which are associated copies of G_1.

2 For $i = 1 \dots n_1$, the i^{th} qubit is taken from each encoded set, to create a block $b(i)$ of size k_2

3 Each $b(i)$ is encoded into a block of n_2 qubits, giving us $n = n_1 n_2$ total qubits used. In order to implement this as a stabiliser code, generators are constructed: a copy of G_2 (denoted $G_{2,i}$) is associated to each block. Then, the image of the copies of G_1 under C_2 are added to the generator set: $C_2(G_1) = \{\xi_2 g_i^1 \xi_2^\dagger | i = 1 \dots n_1 - k_1\}$. As expected, the total number of generators is $n_1 n_2 - k_1 k_2$, meaning that $k_1 k_2$ qubits are encoded.

If k_2 is a factor of n_1 (for example, if C_2 is an $[n_2, 1, d_2]$ code), the encoding can be performed more efficiently, by dividing the encoded qubits from stage 1 into blocks of length k_2. This results in an $[\frac{n_1 n_2}{k_2}, k_1, d]$ code. This results in a code of the same rate as the general case, but requires fewer qubits to be used.

6.2.2 C_4/C_6

C_4[11] is the [4,2] stabiliser code with generators $g_x = \sigma_x^1\sigma_x^2\sigma_x^3\sigma_x^4$ and $g_z = \sigma_z^1\sigma_z^2\sigma_z^3\sigma_z^4$. Unlike the example of the Steane code I gave before, this encodes a *pair* of qubits into 4, so that each of the 2 qubits have their own Pauli X and Z operators:

$$X_1 = \sigma_x^1\sigma_x^2; Z_1 = \sigma_z^1\sigma_z^3 \tag{6}$$

$$X_2 = \sigma_x^2\sigma_x^4; Z_2 = \sigma_z^3\sigma_z^4 \tag{7}$$

C_6 is a [6,2] code and has generators $(\sigma_x^1\ I\ I\sigma_x^4\sigma_x^5\sigma_x^6), (\sigma_x^1\sigma_x^2\sigma_x^3\ I\ I\sigma_x^6),\ (\sigma_z^1\ I\ I\sigma_z^4\sigma_z^5\sigma_z^6)$ and $(\sigma_z^1\sigma_z^2\sigma_z^3\ I\ I\sigma_z^6)$. The encoded X and Z operators are

$$X_1 = \sigma_x^2\sigma_x^3; Z_1 = \sigma_z^3\sigma_z^4\sigma_z^6 \tag{8}$$

$$X_2 = \sigma_x^1\sigma_x^3\sigma_x^4; Z_2 = \sigma_z^4\sigma_z^5 \tag{9}$$

To achieve a 2-level fault-tolerant encoding, start with 12 qubits. With C_4, each set of 4 physical qubits is used to encode a pair of "logical" qubits; therefore we have 3 of these pairs. In turn, these pairs can be used to encode a pair of qubits using C_6.[11] At each level, ancilla qubits are also applied, to calculate error syndromes (see figure 4)

In order to perform error correction, the syndromes are checked for each level in turn,

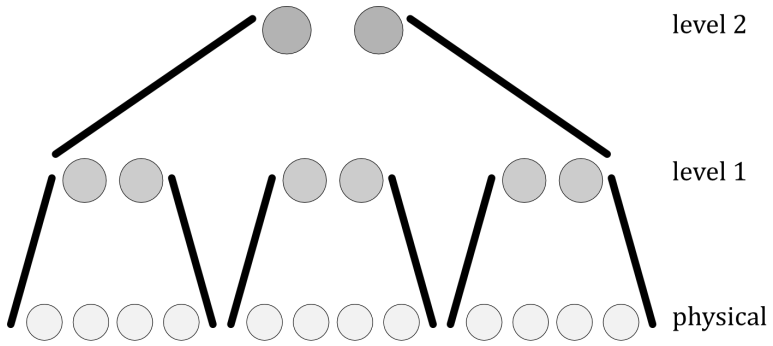

Figure 4: Using 12 physical qubits to create a 2-level encoded logical qubit pair

in order to pinpoint the location of an error. First, the C_4 syndromes for physical qubits encoding pairs are checked, and if an error is found, this pair is marked. Next, on level 2 the C_6 syndrome for each set of three pairs is checked - as C_6 is 1-error correcting, if exactly 1 pair has an error, it is corrected. If there are 2 or 3 errors, the pairs are simply marked as having an error. This procedure is continued if there are more layers of

encoding, until each layer has been checked. [11]
It is possible for the process of extracting the error syndrome to have errors as well, which necessitates a fault-tolerant method. One method involves remeasuring the syndrome:

Syndrome = 0 This is accepted as correct, and no recovery operations are applied

Syndrome $\neq$ 0 The ancilla block used is discarded and a new one is prepared, and the syndrome is remeasured. The new result is assumed to be correct.

If the syndrome returned is 0, but a single error is present, then because the protocol does nothing to the block after measurement this error does not propagate. If the syndrome is non-zero, it is most likely that exactly one error has occured. Another error occurring while the second syndrome extraction occurs is assumed to be incredibly unlikely - $\mathcal{O}(p^2)$ - and so we can assume that the measured value is correct. Therefore, applying the recovery operator for the extracted syndrome will not cause a further error. This makes the protocol fault tolerant (as p is typically taken to be very small).[5]

6.3 Implementing Fault-Tolerant Operations

In order to do fault-tolerant computing, we need to develop fault-tolerant *universal logic gates*: Hadamard, phase, CNOT and $\pi/8$. If an encoded gate can be implemented transversally - either only using single qubit gates on a block, or only causing the i^{th} qubits in separate blocks to interact - then it is also fault tolerant. In the second case, as the different qubits being affected by the gate are in different blocks, any errors caused by the gate will also be applied to at most one qubit in each of the different blocks, fulfilling the conditions of fault tolerance.
Transversality can easily be introduced into several of the gates by applying them bitwise - applying to each qubit of the code block in turn. This extends to multi-qubit gates like the CNOT: for each $1 \leq i \leq n$, a separate CNOT gate is applied to the i^{th} qubit of the target block, according to the i^{th} qubit of the control block.

6.3.1 Gates for the [5,1] code

As discussed in the previous chapter, the encoded Pauli operators for this code are $X^* = \sigma_x^1\sigma_x^2\sigma_x^3\sigma_x^4\sigma_x^5$ and $Z^* = \sigma_z^1\sigma_z^2\sigma_z^3\sigma_z^4\sigma_z^5$. Additionally, $Y^* = iX^*Z^* = i(\sigma_x^1\sigma_z^1)(\sigma_x^2\sigma_z^2)\dots(\sigma_x^5\sigma_z^5) = (\sigma_y^1)(-i\sigma_y^2)\dots(-i\sigma_y^5) = \sigma_y^1\sigma_y^2\sigma_y^3\sigma_y^4\sigma_y^5$.
These are similar to those for the Steane [7,1] code found in [9] and [13]. Most of the gates can easily be encoded by bitwise application of the necessary gate to each of the unencoded qubits: hence the Hadamard gate, which interchanges the X and Z operators, can be encoded $H^* = H^1H^2H^3H^4H^5$, and the phase gate S, which takes the X operator to the Y operator, can be encoded $S = S^1S^2S^3S^4S^5$ [7]

[7] in fact, this is even easier than the Steane code, for which bitwise application of the unencoded phase gate takes X^* to $-Y^*$. This requires a Z gate to be applied bitwise as well to remove the negative

The final gate to consider is the $\pi/8$ gate, $T = \begin{bmatrix} 1 & 0 \\ 0 & e^{i\pi/4} \end{bmatrix}$, which is not implemented bitwise, but instead using the fault-tolerant gates already constructed. It is easier to construct this by considering the effect on an arbitrary state[9]:

$$T(a\,|0\rangle + b\,|1\rangle) = a\,|0\rangle + be^{i\pi/4}$$

The implementation begins by preparing a bundle of 5 ancilla qubits in the state $\frac{1}{\sqrt{2}}(|0\rangle + e^{i\pi/4}\,|1\rangle)$, applying unencoded Hadamard and $\pi/8$ gates to them in turn. Then a set of CNOT gates with the ancillas as the source and the qubits we want to apply the gate to as the target are implemented. This gives the a state where the amplitudes of the superposition are again a superposition of $|0\rangle$ and $|1\rangle$ states, allowing an intermediate measurement to be carried out:

$$(a\,|0\rangle + be^{i\pi/4}\,|1\rangle)\,|0\rangle + (a\,|1\rangle + be^{i\pi/4}\,|0\rangle)\,|1\rangle$$

Finally the target qubit is measured. Clearly if the measurement is 0, this gives the required state (here, the measurement only collapses the state of the second qubit to 0, leaving the superposition $(a\,|0\rangle + be^{i\pi/4}\,|1\rangle)$ intact). Otherwise, the combination SX 0 is applied to $a\,|1\rangle + be^{i\pi/4}\,|0\rangle$, again giving the correct effect. As long as the measurement and ancilla preparation are enacted fault-tolerantly, this gate will also be fault tolerant, but a failure in the measurement process could cause multiple errors as SX is applied in a controlled way.

All the gates apart from $\pi/8$ above are transversal and thus fault-tolerant; as the $\pi/8$ gate is constructed from fault tolerant procedures, errors will also not propagate. A single error occurring before the CNOT gate will lead to one error in each of the qubit blocks. As only one of these blocks is measured, this fault-tolerant measurement procedure will also not cause this error to propagate due to the controlled-SX. If the error occurs after the CNOT gate, the fault tolerant nature of each of the remaining gates again prevents it from propagating. [9]

An interesting difference between the gates I have described above and potential gates for the C_4/C_6 architecture is with the Hadamard gate. Considering the C_4 code, while the CNOT gate can be implemented transversally in the same way, a gate interchanging $\sigma_x^1\sigma_x^2$ and $\sigma_z^1\sigma_z^3$ would seem to need a slightly more complicated approach. When considering potential gates for this code, my initial idea was the encoded gate HXZ; however using permutations allows an easier transversal implementation: relabelling the qubits without any physical movement, and then applying bitwise Hadamard gates, is also effective and has one fewer gate to fail.[11]

6.3.2 Measurement

As shown by the construction of the $\pi/8$ gate above, it is important to also develop a fault tolerant measurement scheme in order to properly run a fault tolerant computation. As well as gate implementation, it is also important for retrieving error syndromes in the process of computing. The encoded operators need to not only not propagate errors, but be able to give the correct reading with a very high probability $(1-O(p^2))$, so that controlled gates and other operations dependent on measurement are less likely to fail. Suppose an observable M needs to be measured. The procedure uses ancilla qubits, prepared in a 'cat' state[8] $|0\ldots0\rangle + |1\ldots1\rangle$. A verification procedure makes sure that they are all in the correct state.

Next, controlled measurement operators are applied, depending on the state of the ancilla qubits; the measurement gate is applied to the data qubit if the paired ancilla qubit is in $|1\ldots1\rangle$, and not if the ancilla qubit is in $|0\ldots0\rangle$. Finally, CNOT and Hadamard gates are used to decode the ancilla qubits, which will return a value of 0 or 1 depending on the eigenvalue of the encoded data. Using the cat state means that a single error during verification or the controlled M-gate causes at most one error in the data, and in the final stage the CNOT and H gates only act on the ancilla, so that any error at this stage won't affect the encoded data; so this is a fault tolerant procedure. If the data were measured just by bitwise implementation of a controlled M-gate, there would be potential for error propagation: there would be a single ancilla qubit, so a failure in this qubit would cause many errors on the data block. [9]

6.4 Operating with the Surface Code

Carrying out logical operations on a surface code array makes use of the discrepancy between the number of data and the number of measurement qubits, which allows some degrees of freedom for the data qubits. The way in which the array is set up may mean, for example, that the left and right boundaries have X-measurement and data qubits, and the top and bottom Z-measurement and data. In this case, the boundary data qubits are only measured by 3 or 2 measurement qubits.

2 X-measurement operations above and below a data qubit commute with the Z stabiliser to its right, as they both change the eigenstate by -1. This indicates that the way to implement an X operation is a chain of X operations across the array, from a data qubit on the left boundary to one on the right. In the same way, 2 Z operations commute with the X stabiliser, so the logical Z operation is implemented by a chain of Z operations from top to bottom. The measurement outcomes for the new quiescent states will be the same, indicating no error has occurred, but the state itself will be different. Using these operations allows the array to be used as a single logical qubit: in order to increase the number of logical qubits it represents, it is necessary to create more degrees of freedom.

[8] This refers to a state being a superposition of opposing coherent states - like Schrödinger's cat being both alive and dead

It is common for this to be done by creating "defects", which is done by switching off some of the measurement qubits, resulting in a Z-cut or X-cut qubit depending on the disabled qubit. This increases the degrees of freedom and therefore allows more logical operators to be defined, creating more qubits. For example, suppose an X-cut - called a rough defect - has been created. Then an X operator is defined by applying σ_x to each of the data qubits surrounding the disabled X measurement (paired across the Z measurement qubits around the "hole"), and a Z operator is defined as a chain of σ_z operations from the nearest edge of the array to one of the same data qubits surrounding the hole. A similar Z defect is referred to as a smooth defect.
Creating further defects, or switching off more stabilisers per defect, creates more possibilities for logical operators, such as between the boundaries of different X-cuts. Implementing logic gates takes a toplogical approach, by moving the position of the holes around the array.[10] Compared to other coding protocols, the main advantage of surface codes are their increased accuracy threshold - it is possible for this to be about 1%, compared to around 1×10^{-5}% for the Steane code. This, however, comes with the requirement of a very large number of qubits - from 10^3 to 10^4 physical are required for fault-tolerance for *each* logical qubit.[14]

6.5 Current Research in Fault Tolerance

Current areas of interest in making computers fault tolerance include increasing efficiency, and working to make sure that fault-tolerant computers are scalable to the size needed for practical quantum computers. For example, in [15], the authors outline an proposal for a computer using photonic qubits, assering that such a computer is not only scalable but practical: photonic qubits are a popular phsyical modality as they are able to function at room temperature.
Meanwhile, in [14] Gottesmann discusses various potential avenues of research for the field, such as the use of low-density parity check codes, which are a specialised subset of stabiliser codes. These codes have generators which act on a constant (for large numbers of physical qubits) number of qubits, and each qubit is only involved in a constant number of generators; research is needed to try and develop parallel gate implementations for fault-tolerance, in order to make these a sensible choice compared to other families of codes such as surface codes. Another area of research he outlines is adaptation of codes to specific hardware - on qubit modalities with continuous rather than discrete degrees of freedom, like harmonic oscillators, bosonic codes are a possibility. These take advantage of the infinite Hilbert space that the hardware has. Finally, fault-tolerance protocols can also be developed which are time-based as well as space-based, such as flag codes. In this protocol, a phase error in a CNOT gate, for example, will also cause the phase of an ancilla qubit to change. Then by finding out when the error occurred, it is possible to tell whether this error will have affected one qubit in the target block, or many. In particular, the use of a single ancilla means less additional hardware is required.

7 Conclusion

As quantum computing becomes more powerful and the hardware is scaled up further, robust error correction protocols will be essential, in order to overcome the increased potential for noise from outside interactions as well as for enabling communication between different computers. An important factor in this, which I did not have room to discuss, will be maximising efficiency of encoding and fault-tolerance schemes, so that the increased speed compared to classical computers can be properly realised. Furthermore, every stage of a quantum computation requires a fault-tolerance protocol to prevent the propagation of errors, from logic gates and measurement to the preparation of the ancilla qubits used to make the measurement process fault tolerant. Given more space, I would like to have additionally discussed how this state preparation is done, as well as explore current areas of research such as the flag code in more detail. As well as the error model based on the Pauli group, various environmental interactions can cause errors in other forms, such as amplitude damping, which aren't as widely investigated due to their more complex nature. Therefore this would also be an interesting area of further research.

Bibliography

References

[1] C. Clapham and J. Nicholson, *Oxford Concise Dictionary of Mathematics*. Oxford University Press.

[2] D. Ueltschi, "Ma4a7: Quantum Mechanics: Basic Principles and Probabilistic Methods," 2021.

[3] H. D. Young, R. A. Freedman, A. L. Ford, F. W. Sears, and M. W. Zemansky, *Sears and Zemansky's university physics: with modern physics*. No. Book, Whole, Essex, England: Pearson, fourteenth, global ed., 2016.

[4] B. C. Hall, *Quantum Theory for Mathematicians*. Springer, 2013.

[5] F. Gaitan, *Quantum error correction and fault tolerant quantum computing*. No. Book, Whole, Boca Raton, FL: CRC Press, 2008.

[6] "Schrödinger equation - Encyclopedia of Mathematics."

[7] V. Kasirajan, *Fundamentals of Quantum Computing*. Springer, 2020.

[8] "Qubits: A Primer (Part 6 of 8)."

[9] M. A. Nielsen and I. L. Chuang, *Quantum computation and quantum information*. No. Book, Whole, Cambridge: Cambridge University Press, 2000.

[10] A. G. Fowler, M. Mariantoni, J. M. Martinis, and A. N. Cleland, "Surface codes: Towards practical large-scale quantum computation," *Physical Review A*, vol. 86, sep 2012.

[11] E. Knill, "Quantum computing with realistically noisy devices," *Nature*, vol. 434, pp. 39–44, Mar. 2005. Place: London Publisher: Nature Publishing Group.

[12] D. Aharonov and M. Ben-Or, "Fault-Tolerant Quantum Computation With Constant Error Rate," June 1999. arXiv:quant-ph/9906129.

[13] I. Djordjevic, *Quantum information processing and quantum error correction: an engineering approach*. No. Book, Whole, Waltham, MA;Oxford, UK;: Academic Press, 2012.

[14] D. Gottesman, "Opportunities and challenges in fault-tolerant quantum computation," 2022.

[15] J. E. Bourassa, R. N. Alexander, M. Vasmer, A. Patil, I. Tzitrin, T. Matsuura, D. Su, B. Q. Baragiola, S. Guha, G. Dauphinais, K. K. Sabapathy, N. C. Menicucci,

and I. Dhand, "Blueprint for a scalable photonic fault-tolerant quantum computer," *Quantum*, vol. 5, p. 392, feb 2021.

[16] "Superconducting quantum bits," Dec. 2004.

[17] "What are Josephson junctions? How do they work?."

[18] "hyperfine structure | physics | Britannica."

[19] A. M. Steane, "The Ion Trap Quantum Information Processor," *Applied Physics B: Lasers and Optics*, vol. 64, pp. 623–643, June 1997. arXiv:quant-ph/9608011.

[20] W. C. Huffman and V. Pless, *Fundamentals of Error Correcting Codes*. Cambridge University Press, 2003.

A Physical Qubit Modalities

Superconducting Qubits These are one of the most popular and widely-researched types of qubit. Superconductance is a phenomenom observed in many metals and alloys when they are cooled to close to absolute zero. At this temperature, free electrons attract each other rather than repel, and this drops them into a lower energy state, allowing them to flow without the ion scattering that causes electrical resistance in normal conductors - up to a maximum "critical current".[16]

Superconducting qubits utilise Josephson junctions, where a very thin barrier strip of non-superconducting material is placed between two superconducting islands. Electrons tunnel through the barrier, causing the flow of a supercurrent around the junction.[17]

The main sources of noise are various fluctuations in magentic spin, charge, and magnetic vortices. [8]

Trapped Ion Qubits For these qubits, ions are created and held in place using radio frequencies and DC electrodes. Typically, the ions used are Ca^+, and the $|0\rangle$ and $|1\rangle$ states are the two different hyperfine[9] ground states - the $|1\rangle$ state fluoresces but the $|0\rangle$ state does not, making measurements easy - operations are performed using fine-tuned radio frequencies.[19] Alternatively, the ground state and an excited state are used, and a the ion is excited or de-excited between energy levels. In this case, measurement is performed by measuring whether a photon is emitted, indicating a $|1\rangle$ state.

Because the amount of energy used to change the energy levels of the electron is so small, extra thermal energy can interfere with the qubits and produce noise, requiring the system to be only a few mK above absolute zero. Noise is also possible due to instability in the lasers, and generally due to environmental interference.

[9] Essentially, the hyperfine structure of an atom arises from further splitting of energy levels due to spin[18]

Quantum Dots These utilise semiconductors to confine the spin-$\frac{1}{2}$ states of an electron. There are a range of energy levels in the material which the electron cannot occupy - the bandgap. Most electrons are found below this in the valence band, but some occupy the conductance band of higher energy levels. The width of the gap is aritficially manipulated, and a particular voltage is applied so that an electron can only enter the dot from a reservoir when another elctron has tunnelled out, controlling the number of electrons in the dot. When there is a single electron in the dot, the two possible states are spin-up and spin-down. Gates are applied using a microwave field, which can be precisely controlled.

The main error source for this implementation is decoherence, which can be described as the gradual decay of the superposition to one state.[8]

Photonics Photon-based qubits can use various properties of light to form a superposition, such as the polarisation of its electromagnetic field or the path it takes after encountering a beam splitter. They are susceptible to noise from photon loss.[8]

B Basics of classical error correction

B.1 Definitions and Basic Principles

B.1.1 Notation

Binary codes are typically described in terms of their rate, and the minimum distance between codewords. If k bits of information are encoded into n bits, and the minimum distance between vectors is d (the code is *d-separated*), then this is an [n,k,d] code. It is also a t-error correcting code, where $t = \lfloor \frac{d-1}{t} \rfloor$. This is the maximum possible radius of disjoint spheres in $\mathbb{F}_2^n$, meaning that if at most t errors are made in transmission, the received vector will remain in the sphere centred at the original codeword, and therefore will be corrected by nearest-neighbour decoding.

Linear codes are typically defined using a matrix: either consider a basis for $\mathcal{C}$, in which case the $n \times k$ *generator matrix* G, which has k independent vectors as its columns, is constructed; or the linear relations defining $\mathcal{C}$ are used to define the $(n-k) \times n$ parity check matrix H

$$\mathcal{C} = \{\mathbf{x} \in \mathbb{F}_2^{\mathrm{n}} : \mathrm{H}\mathbf{x}^{\mathrm{T}} = 0\} \tag{10}$$

The generator matrix is used as an encoder, with the codewords being $\mathbf{x} = \mathrm{G}\mathbf{m}$, and the two matrices are related: as the rows of G are in $\mathcal{C}$, multiplying them by the parity check matrix will give 0. Therefore, $HG = 0$. An error can also be defined as a vector in $(\mathbb{F}_2)^n$: if the sent vector is $\mathbf{x}$and the recieved $\mathbf{y}$, then the error vector $\mathbf{e}$ is defined as $\mathbf{x} + \mathbf{y} = \mathbf{x}$ - $\mathbf{y}$. This is a vector with entry 1 everywhere the sent and recieved vectors differ, and the sum of its entries gives the Hamming distance - the number of bits differing between vectors - of $\mathbf{x}$ from $\mathbf{y}$. Commonly, nearest-neighbour decoding is used: the sent message is

assumed to be the one whose Hamming distance (the number of bits different) from the recieved message is smallest. The Hamming weight of an individual vector is the sum (in $\mathbb{R}$) of the individual vector entries.[5]

B.2 Shannon's Theorem

Again, we will consider a binary channel with a crossover probability p of flipping a bit, a capacity $C(p)$[10] and a *word error rate* of P_{err}; the word error rate is the probability of the decoder choosing the wrong codeword. Shannon's theorem tells us that with certain conditions there is an efficient error correcting code.

Theorem 4 (Shannon)**.** *Let $\delta > 0$ and $R < C(p)$. Then for n large enough, there exists an [n,k] binary linear code $\mathcal{C}$ with k/n $\geq$ R, such that $P_{err} < \delta$*

By decreasing δ and the distance between R and C(p), we are able to obtain a code that has almost maximum accuracy and rate of transmission. [20]

B.3 Entropy

Entropy is a measure of how indeterminate a variable is: in classical information theory this is described using the Shannon entropy, which calculates the average amount of information in each source symbol: suppose α is a discrete random variable, taking values $x_1, x_2...$ with the probability $P(\alpha = x_k) = p_k$. The amount of information relates to p_k - if this is smaller, the information contained if this symbol occurs is greater

$$I(x_n) = -\log_2(p_n)$$

Then the Shannon entropy of α is:

$$H(\alpha) = \sum_k p_k I(x_k) = \sum_k p_n(-\log_2(p_n))$$

This concept is extended into quantum information with the Von Neumann entropy, which is a measurement of the uncertainty in the system before measurements are made. Describing the state in terms of a density operator $\rho = \sum_i \lambda_i |\phi_i\rangle \langle\phi_i|$, which has a probability p_x of being in the state ρ_x. Its von Neumann entropy is defined as

$$S(\rho) = -\operatorname{Tr}(\rho \log \rho) = -\sum_i \lambda_i \log \lambda_i \tag{11}$$

If all the states considered are pure and orthogonal (with $\rho = \sum_x p_x \rho_x$ then

$$S(\rho) = \sum_i p_i \log p_i \tag{12}$$

[10]The channel capacity is a function of the crossover probability and is related to the Hilbert entropy of the channel