

# Quantum Error Correcting Codes

Srabonti Ganguly

April 14, 2004

## **Abstract**

We study quantum error correction codes that are important in the eventual realization of full-scale quantum computation. A main focus in our study is a class of quantum error correcting codes called Calderbank-Shor-Steane (CSS) codes. This class of codes is built using classical error-correcting codes but is combined in a way that leads to robust quantum codes. We consider certain specific examples of these codes, namely, the Hamming codes, its dual, and other codes with better error-correcting capabilities.

# Contents

|          |   |           |
|----------|---|-----------|
| <b>1</b> | <b>Introduction</b>                     | <b>3</b>  |
| <b>2</b> | <b>Quantum Computation</b>              | <b>5</b>  |
| 2.1      | Mathematical Background . . . . .       | 7         |
| 2.2      | Axioms of quantum computation . . . . . | 9         |
| 2.3      | Teleportation . . . . .                 | 12        |
| <b>3</b> | <b>Quantum Error Correction</b>         | <b>14</b> |
| 3.1      | Quantum errors . . . . .                | 15        |
| 3.2      | Shor's Quantum Code . . . . .           | 17        |
| 3.3      | Classical Linear Codes . . . . .        | 18        |
| 3.4      | CSS Quantum Codes . . . . .             | 19        |
| <b>4</b> | <b>Explicit Codes</b>                   | <b>22</b> |
| 4.1      | Hamming Codes . . . . .                 | 22        |
| 4.2      | Simplex Codes . . . . .                 | 23        |
| 4.3      | Steane's Quantum Code . . . . .         | 23        |
| 4.4      | Golay codes . . . . .                   | 24        |
| <b>5</b> | <b>Conclusion</b>                       | <b>27</b> |

# 1 Introduction

Quantum computation is the study of computation using the principles of quantum mechanics, where quantum information is the relevant data on which computation is performed. The field of quantum computation is relatively new. In the early 1980s, Feynman [4] conjectured that a computer modeled on the principles of quantum physics, rather than classical physics, could simulate quantum mechanical systems. However, research in this field had not progressed until the last decade or so.

In 1994, Shor [1] described a theoretical quantum algorithm which could efficiently find the prime factors of any integer. The successful demonstration of this algorithm has serious implications on the present state of cryptography. The popular RSA cryptographic system, whose security is based on the hardness of factoring large integers, is widely used in current financial transactions and highly sensitive communications. This cryptographic system relies on the fact that factoring extremely large numbers are "impossible" using present day computers. Fortunately (or unfortunately), Shor's algorithm is only a theoretical result, i.e., there are no known implementations of quantum computers. Nevertheless, his algorithm had sparked significant interest in quantum computation and speculations abound as to whether quantum computers can be built.

When it comes to the physical realization of a quantum computer, physicists encountered severe fundamental practical problems. The preliminary models of quantum computers built so far have shown extreme sensitivity to external influences. Unlike classical computers, the errors affecting a quantum computer form a continuum, and so the accumulation of errors would most certainly destroy the coherence of the entire system. Therefore, an efficient yet non-classical error correcting is required in order for successful quantum computation. Such type of error correction was considered impossible, since classical error correcting is done mostly by duplication or repetition of data. This is impossible in the quantum world due to the famous No-cloning theorem.

But in 1995, Shor [2] gave a scheme to correct errors in a quantum system. This is a major breakthrough in the area of quantum error-correcting codes and gave hope that realistic quantum computation was possible. After Shor's error-correcting code, many other codes were subsequently proposed. These also circumvented the No-cloning theorem. In 1996, Steane [7], and independently Calderbank and Shor [3], came up with a beautiful, general, compact class of quantum codes called the CSS codes.

We outline the organization of this thesis. The first part provides some

background on quantum computation and what makes it different from classical computation. Then, we review necessary mathematical background for quantum computation, followed by a brief example of a quantum circuit for a known quantum operation called teleportation. The main section will then describe some of the quantum error models, explain the No-cloning theorem, and discuss Shor's [2] original 9-qubit quantum code. We also give a short primer on classical linear codes which form the foundation for the CSS codes.

## 2 Quantum Computation

What exactly is quantum computation? Classical computers at the most basic levels use bits 0 and 1 as data and use the NAND, NOR and other gates to perform computations on this data. NOT and AND are the universal gates in the classical world and any manipulation of the bits can be done through these operations. We consider an example involving the AND  $\wedge$  operator on Boolean values. Classically, we may view the operation *AND* that performs the Boolean AND of two bits  $a$  and  $b$  as an action on 2-dimensional Boolean vectors that returns a single Boolean value:

$$\begin{bmatrix} a \\ b \end{bmatrix} \implies a \wedge b$$

Here the bits  $a, b$  encode the classical bits having values of 0 or 1. But we could also look represent the AND action as a mapping from 2-dimensional Boolean vectors into 2-dimensional Boolean vectors in the following way

$$\begin{bmatrix} a \\ b \end{bmatrix} \iff \begin{bmatrix} a \wedge b \\ b \end{bmatrix}$$

While in the first case information is lost and there is no way of finding out the states of  $a$  and  $b$  from just  $a \wedge b$ , in the second case, by forcing  $b$  as an output one can easily find out the state of  $a$ . In other words the second representation is reversible while the first is not.

To anticipate the notation that we will use later, suppose that we denote  $|\psi\rangle = \begin{bmatrix} a \\ b \end{bmatrix}$  as the 2-dimensional column vector of input values. Then, the *reversible* AND operation can be viewed as a matrix that maps  $|\psi\rangle$  to another column vector, namely  $|\phi\rangle = \begin{bmatrix} a \wedge b \\ b \end{bmatrix}$ .

Quantum computation is similar to this scenario. In quantum computation, *qubits* or quantum bits are said to represent data as is done by  $a$  and  $b$ . These bits could physically represent subatomic particles. Suppose our 1-qubit system is represented as  $|\Psi(0)\rangle$  at an initial time  $t = 0$ . Then the state of the quantum system evolves over time  $t$  in the following manner:

$$|\psi(t)\rangle = U_t |\psi(0)\rangle,$$

where  $U_t$  is a matrix mapping that obeys certain mathematical properties. Qubits can be represented mathematically as 2-dimensional complex-valued vectors  $|\psi\rangle = \begin{bmatrix} a \\ b \end{bmatrix}$ , where  $a, b \in \mathbb{C}$  are complex numbers that satisfy  $|a|^2 +$

$|b|^2 = 1$ . The matrix mapping  $U_t$  is modeled mathematically as a unitary matrix, i.e., a matrix  $U$  that obeys  $U^{-1} = U^\dagger$ , where  $U^\dagger$  is the Hermitian (conjugate transpose) of the matrix  $U$ .

Unitary matrices act like the standard Boolean gates used in classical computing. Unitary gates are reversible while classical gates are not. So one can represent and transform quantum data using mathematical models making it more abstract. As a result quantum computation can be described using a purely mathematical basis.

Another important part of computation is reading out the result or measuring the outcome of the computation. Measurement in quantum computation is more complicated than in its classical counterpart. Classical computing relies on the deterministic process for computation but because of the property of quantum superposition, measurement in the quantum world is a probabilistic event. This is because a quantum bit can be measured only once, and on measuring, the state of the system is disturbed and the system collapses. As a result the quantum bit becomes a classical (probabilistic) bit.

After a brief introduction to linear algebra, the axioms of quantum computation will be discussed. We described how data is represented, what unitary transformations are, and how the probabilistic measurement is done. So quantum computation can be done, at least, in theory but though there are some major problems associated with the actual simulation of these computations. Physicists, mathematicians, and computer scientists are willing to put a lot of effort into this field. The reason is the tremendous promise that computation in the world of quantum mechanics has shown.

The quantum world possesses characteristics which are considered improbable if viewed in a classical light, although physicists do say that the classical world could also be explained using the quantum mechanical principles. One of the strange characteristics already mentioned is that of superposition. A quantum bit in the state of superposition is equivalent in information to 2 classical bits. This is one of the properties which make quantum computation so powerful.

Quantum systems are very prone to noise and can be easily destabilized. This was one of the first obstacles of practical computation and as a result some kind of quantum error-correction is necessary. Though quantum computation has shown a lot of promise, it must be noted that every classical problem may not have a quantum solution, even if it does, this does not mean that such a solution will always be preferred over the classical one. There have been quantum algorithms, like the factoring algorithm, which are superior to any known classical algorithm. The goal of quantum com-

putation is not just to find a solution, but one which is considerably more efficient than a classical one.

## 2.1 Mathematical Background

Before we describe some useful mathematical terminology, we tabulate some notation that will be used throughout. The notation of representing a column vector as  $|\cdot\rangle$  is called the Dirac notation in quantum physics. The following table in Figure 1 shows some of the common notations used. We denote the basis quantum bits as

$$|0\rangle = \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \quad |1\rangle = \begin{bmatrix} 0 \\ 1 \end{bmatrix}.$$

| Dirac Notation                      | Description   |
|-------------------------------------|---|
| $z^*$                               | complex conjugate of $z \in \mathbb{C}$   |
| $A^T$                               | the transpose of matrix $A$   |
| $A^*$                               | complex conjugate of matrix $A$   |
| $A^\dagger$                         | conjugate transpose of $A$ or $(A^T)^*$   |
| $ \psi\rangle$                      | <i>ket</i> or column vector   |
| $\langle\psi $                      | <i>bra</i> or row vector that is the dual of $ \Psi\rangle$ or $( \psi\rangle)^\dagger$ |
| $\langle\phi \psi\rangle$           | the inner product of $ \phi\rangle$ and $ \psi\rangle$                                  |
| $ \phi\rangle \otimes  \psi\rangle$ | the tensor product of $ \phi\rangle$ and $ \psi\rangle$                                 |

Figure 1: Dirac notation in quantum theory and matrix algebra terminology.

**Vector Spaces** These are the basic "objects" of linear algebra. The vector space used most often in quantum theory is  $\mathbb{C}^n$  which is a space of all  $n$ -dimensional complex vectors. The elements of  $\mathbb{C}^n$  are denoted as column vectors, as follows:

$$\begin{bmatrix} z_1 \\ \vdots \\ z_n \end{bmatrix}$$

where  $z_1, \dots, z_n \in \mathbb{C}$  are complex numbers.

**Basis** Every vector space has a set called the *basis*. A main property of the basis set is that every vector in the vector space can be written as a unique linear combination of the vectors in the basis.

For example for a vector space  $\mathbb{C}^2$ , the basis set could be  $|v_1\rangle = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$  and  $|v_2\rangle = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$ . Any vector  $|v\rangle$  in  $\mathbb{C}^2$  can be written as a linear combination of  $|v_1\rangle$  and  $|v_2\rangle$ :

$$|v\rangle = \alpha|v_1\rangle + \beta|v_2\rangle.$$

One can also construct a different basis set using  $|v_1\rangle = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 \\ 1 \end{bmatrix}$  and  $|v_2\rangle = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 \\ -1 \end{bmatrix}$ . The basis set of a vector space is a linearly independent set. This means that if a linear combination of the bases vectors is the zero vector, then all coefficients in the linear combination are all necessarily zeroes. The number of elements of a basis vector is called the *dimension* of the vector space. In general, we deal only with finite dimensional vector spaces.

**Linear operators** A linear operator  $A$  between two vector spaces  $V$  and  $W$  is a mapping from  $V$  to  $W$  that is linear in its inputs. This means that  $A \sum_j |u_j\rangle = \sum_j A|u_j\rangle$ . In our case, linear operators are simply matrix mappings.

**Inner Product** The inner product mapping is a function  $(\cdot, \cdot)$  which takes two input vectors and returns a complex number as output. As we have seen earlier, the inner product of two vectors belonging to the same vector space is denoted with  $\langle v|w\rangle$  or  $(|v\rangle, |w\rangle)$ . If one represents  $|v\rangle$  as a column matrix, the dual can be represented as a row matrix  $\langle v|$ . The inner product satisfies the following three properties:

1. *Linearity*: it is linear in the second argument

$$(|u\rangle, \sum_j |v_j\rangle) = \sum_j (|u\rangle, |v_j\rangle).$$

2. *Conjugate symmetric*:

$$(|v\rangle, |w\rangle) = (|w\rangle, |v\rangle)^*$$

3. *Positivity*:

$$(|v\rangle, |v\rangle) \geq 0$$

and equality is achieved if and only if  $|v\rangle$  is the zero vector.

If such a function exists, then the vector space is called an inner product space or a *Hilbert* space.

**Tensor Product** Tensor product is a way of creating larger vector spaces from smaller ones. If there are two vector spaces  $V$  and  $W$  of dimensions  $m$  and  $n$ , then  $V \otimes W$  is of dimension  $m \times n$ . One can denote tensor products using matrices. If  $A$  is a matrix of dimension  $m \times n$  and  $B$  is a matrix of dimension  $p \times q$ , then:

$$A \otimes B = \begin{bmatrix} A_{11}B & A_{12}B & \dots & \dots & A_{1n}B \\ A_{21}B & A_{22}B & \dots & \dots & A_{2n}B \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ A_{m1}B & A_{m2}B & \dots & \dots & A_{mn}B \end{bmatrix}$$

is a matrix of dimension  $mp \times nq$ .

**Spectra** An eigenvalue  $v$  with eigenvector  $|v\rangle$  for a linear operator  $A$  satisfies the eigenvalue-eigenvector relation

$$A|v\rangle = v|v\rangle.$$

## 2.2 Axioms of quantum computation

In defining the notion of *computation*, we need to define the basic unit of *information* or data, the allowable *operations* or transformations on the data, and how to *measure* or collect the outcome of the computation. In the classical case, most information is represented as *bits* or Boolean-valued data, the operations are the standard Boolean operators, such as AND, OR, and NOT, and the outcome is simply collected from the output of the final Boolean operation in the computation. In the following, we describe these three ingredients, namely, information, transformation, and measurement, that are relevant for quantum computation.

**Qubits** Quantum bits (qubits) are the representation of data in the quantum world. They can be modeled mathematically on an abstract level. A single qubit can be a  $|0\rangle$  or a  $|1\rangle$ , or a linear combination of both. A qubit is represented as  $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$ , where  $\alpha$  and  $\beta$  are complex numbers that satisfy  $|\alpha|^2 + |\beta|^2 = 1$ . This representation means that, on measuring  $|\psi\rangle$  using the basis  $\{|0\rangle, |1\rangle\}$ , there is a chance of getting  $|0\rangle$  with probability  $|\alpha|^2$  and a chance of getting  $|1\rangle$  with a probability  $|\beta|^2$ .

**Unitary transformations** The physically allowable operations on quantum states are linear operators that are unitary. These operators are represented by unitary matrices. Recall that a unitary matrix  $U$  satisfies  $U^\dagger = U^{-1}$ . Some of the important unitary matrices for quantum computation are mentioned below. The unitary matrices are multiplied with the input vector to give the output vector of the computation. A sequence of computation will be represented as a circuit that contains a number of wires acted upon by a set of unitary gates.

Classically lines represent wires, in the quantum world these lines mean any channel for communication, be it light or particles. Therefore, unitary functions act like quantum gates, however unlike their classical counterparts, like *AND*, *NAND*, and others, unitary matrices are reversible.

**Examples of unitary transformations**

1. *Hadamard* gate:

$$H = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$$

Example:

$$|0\rangle \xrightarrow{H} \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle) \text{ also represented as } |+\rangle$$

$$|1\rangle \xrightarrow{H} \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle) \text{ also represented as } |-\rangle$$

or more generally,

$$|x\rangle \xrightarrow{H} \frac{1}{\sqrt{2}}[|0\rangle + (-1)^x|1\rangle]$$

2. Pauli *X* gate:

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

This matrix is also called the *bit flip* matrix.

Example:

$$\alpha|0\rangle + \beta|1\rangle \xrightarrow{X} \alpha|1\rangle + \beta|0\rangle$$

3. Pauli *Z* gate:

$$Z = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}$$

This matrix is also called the *phase flip* matrix.

Example:

$$\alpha|0\rangle + \beta|1\rangle \xrightarrow{Z} \alpha|0\rangle - \beta|1\rangle$$

4. Identity  $I$  gate (No Operation):

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

There is no change in the qubit after the action of  $I$ .

5. CNOT or *Controlled-Not* gate:

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

Example:

If the Control bit  $|0\rangle$  and Target bit is  $|0\rangle$ , after going through the CNOT gate, the Target bit remains unchanged, however if the Control bit is  $|1\rangle$  then after going through the CNOT gate Target bit flips to  $|1\rangle$ .

**Measurement** Measurement is represented using the concepts of projection of vectors and probability. Measurement of a one-qubit system will be shown below.

For a one-qubit system, the only two possibilities are  $|0\rangle$  and  $|1\rangle$ . The measurement set is formed by taking the projections of  $|0\rangle$  and  $|1\rangle$ . As a result the measurement set for a one-qubit system is  $M_0 = |0\rangle\langle 0|$  and  $M_1 = |1\rangle\langle 1|$ , where

$$|0\rangle\langle 0| = \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix}, \quad |1\rangle\langle 1| = \begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix}.$$

Then, performing a measurement with the set  $M = \{M_0, M_1\}$  on the qubit  $|\psi\rangle$  yields the state

$$M(|\psi\rangle) \implies \begin{cases} \frac{M_0|\psi\rangle}{\sqrt{\langle\psi|M_0^t M_0|\psi\rangle}} & \text{with probability } \langle\psi|M_0^t M_0|\psi\rangle \\ \frac{M_1|\psi\rangle}{\sqrt{\langle\psi|M_1^t M_1|\psi\rangle}} & \text{with probability } \langle\psi|M_1^t M_1|\psi\rangle \end{cases}$$

### 2.3 Teleportation

In this section we discuss an example of a quantum computation for creating what is known as the *Bell* state  $\beta_{00}$ . This is a state that was conceived by Einstein, Podolsky and Rosen [5] to disprove quantum mechanics.

This state possesses the unique property of entanglement. Entanglement occurs when two qubits form a state in which, they are inextricably linked with each other. If by some way the two qubits are separated by vast amount of space, they still have a common bond between them. Such states can be prepared in laboratories. The qubits, say qubit  $A$  and qubit  $B$  have been separated by a large distance. The observation which was found to be amazing, was that whenever qubits  $A$  and  $B$  were measured simultaneously, qubit  $A$  could predict the position of qubit  $B$  with great accuracy. If qubit  $A$  measured 1, qubit  $B$  would also measure 1. The fact that despite such great distance separating them, the qubits seemed to produce identical results was inexplicable and when this phenomenon was first noted led to a lot of puzzlement. Although till today it is not known what causes the property of entanglement it has been exploited in the field of quantum computation.

Let there be a two qubit system  $|00\rangle$ . We hit the first qubit with the standard Hadamard matrix. The output of this together with the second qubit in the system become the input to a CNOT gate. The result is that an EPR or Bell state is formed.

$$\beta_{00} = \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle)$$

Now suppose the state  $\beta_{00}$  has been formed and the two qubits of this state have been separated. One qubit is kept with the sender, who wishes to teleport a different qubit, the other is kept with the receiver who will receive the mentioned qubit.

Let the qubit, which needs to be teleported be  $|\Psi\rangle$ . The following is done to it and the bell state qubits.

In this circuit,  $M_1$  is the measurement taken after  $|\Psi\rangle$  is hit by the Hadamard gate and  $M_2$  is the measurement taken after the sender's bell qubit is hit by the CNOT gate.  $|\Psi\rangle$  is the control qubit and the sender's bell qubit is the target qubit.  $M_1$  and  $M_2$  are the classical bits obtained. In the process both the qubits possessed by the sender gets destroyed. The classical bits from  $M_1$  and  $M_2$  are then sent to the receiver by a classical channel. If  $M_2$  is a 1, the receiver flips his Bell qubit with the Pauli  $X$  gate. If  $M_1$  is 1, the receiver flips his Bell qubit with the Pauli  $Z$  gate. The resulting qubit becomes , the original qubit that the sender wanted to teleport.

Here is how it happens. Initially the system is at

$$|\Psi_0\rangle = |\Psi\rangle\beta_{00} = \frac{1}{\sqrt{2}}[\alpha|0\rangle(|00\rangle + |11\rangle) + \beta|1\rangle(|00\rangle + |11\rangle)]$$

We assume that the first two qubits on the left are the sender's qubits and the third qubit is the receiver's qubit. After the CNOT gate, the system changes to:

$$|\Psi_1\rangle = \frac{1}{\sqrt{2}}[\alpha|0\rangle(|0\rangle + |1\rangle)(|00\rangle + |11\rangle) + \beta|1\rangle(|0\rangle - |1\rangle)(|00\rangle + |11\rangle)]$$

After rearranging the equation we obtain

$$\frac{1}{2}[|00\rangle(\alpha|0\rangle + \beta|1\rangle) + |01\rangle(\alpha|1\rangle + \beta|0\rangle) + |10\rangle(\alpha|0\rangle - \beta|1\rangle) + |11\rangle(\alpha|1\rangle - \beta|0\rangle)]$$

So one can easily see that if the sender measures a 00, the original state of is received by the receiver. If a 01 is measured, the receiver performs a bit flip operation on the qubit he owns, if a 10 is measured then a phase flip operation is performed and if a 11 is measured, then the receiver performs both the bit flip and phase flip operation is performed. In each case the original state of is obtained which is amazing, because this is not a copy of, but the actual qubit itself which has been transferred from the sender to the receiver.

### 3 Quantum Error Correction

In classical error-correction, the goal is to encode a set of  $k$ -bit sequences (called messages) into a set of  $n$ -bit sequences (called codewords), where  $n > k$ , such that if at most  $t$  positions are corrupted in any codeword, it is possible to recover the original  $k$ -bit message. So, the goal is to create two mappings  $G$  and  $H$ , where  $G : \mathbb{Z}_2^k \rightarrow \mathbb{Z}_2^n$  is the encoding operation and  $H : \mathbb{Z}_2^n \rightarrow \mathbb{Z}_2^k$  is the decoding operation.

The intuition behind error-correction is to have  $G$  map the space  $\mathbb{Z}_2^k$  into  $2^k$  codewords in  $\mathbb{Z}_2^n$  that are separated by a distance of at least  $2t + 1$  from each other. The distance notion used here is the Hamming distance between two binary sequences, i.e., the number of bit positions that are different in the two sequences. If codewords are of distance  $2t + 1$  from each other, any changes of at most  $t$  bits, will still allow a perfect recovery to the original message.

In the case of quantum error-correction, there are some major difficulties which we list below.

- *The No-cloning theorem:*

Most classical error-correction relies on the idea of duplicating parts or most of the components in a message. For example, consider the *repetition* code that maps 0 to 000 and 1 to 111, where decoding simply amounts to taking a majority vote. In the quantum model, the No-cloning theorem, whose proof we describe below, states that it is impossible to duplicate or copy arbitrary quantum states. This seems to present an impossible obstacle to quantum error-correction. In fact, this is the main reason that discouraged progress in this area, until Shor proved that this intuition is wrong [2].

- *A continuum of errors:*

In the classical case, an error that affects a bit can only do one of two things: do nothing or negate the bit. In the quantum model, a quantum error on a single qubit can be an arbitrary unitary operator. Since there are infinitely many unitary operators, it seems that a quantum error-correction scheme must be able to defend against an infinite amount of possible errors.

- *Measurement:*

To correct errors, it seems essential to observe or measure the data so that one can reverse its effects. In the quantum world, a measurement

of a quantum system destroys it irreversibly. Also it is not possible to measure the system multiple times.

We discuss each of these obstacles in the following.

**No-cloning theorem** As mentioned earlier, the No-cloning theorem prevents cloning of arbitrary quantum bits and is the main reason why simple classical error correcting techniques can not be applied on quantum system, though this may mean that the theorem is a hindrance to quantum computation, it actually aids in secure quantum cryptography, because it is impossible to copy information being transmitted. Shor [2] and Steane [7] devised quantum error correcting codes which circumvent this theorem.

**Theorem 1** *There is no unitary  $U$  such that for any qubit  $|\psi\rangle$*

$$U(|\psi\rangle \otimes |0\rangle) = |\psi\rangle \otimes |\psi\rangle.$$

**Proof.** We prove this by contradiction. So, assume there is  $U$  such that  $U(|\psi\rangle \otimes |0\rangle) = |\psi\rangle \otimes |\psi\rangle$ , for any qubit  $|\psi\rangle$ . Let  $|\Psi_1\rangle$  and  $|\Psi_2\rangle$  be two arbitrary qubits. By the cloning property of  $U$ ,

$$U(|\Psi_1\rangle \otimes |0\rangle) = |\Psi_1\rangle \otimes |\Psi_1\rangle, \quad U(|\Psi_2\rangle \otimes |0\rangle) = |\Psi_2\rangle \otimes |\Psi_2\rangle.$$

Thus  $U((|\Psi_1\rangle + |\Psi_2\rangle) \otimes |0\rangle) = (|\Psi_1\rangle + |\Psi_2\rangle)^{\otimes 2}$ . But, by linearity,

$$U((|\Psi_1\rangle + |\Psi_2\rangle) \otimes |0\rangle) = |\Psi_1\rangle \otimes |\Psi_1\rangle + |\Psi_2\rangle \otimes |\Psi_2\rangle,$$

which is not equal to  $(|\Psi_1\rangle + |\Psi_2\rangle)^{\otimes 2}$ . ■

### 3.1 Quantum errors

The general idea behind error correcting of any kind is to encode the data to be stored or transmitted in order to protect it. When the data needs to be retrieved, the data is decoded back to its original state. The data is decoded in such a way that errors are detected in the data and then corrected.

There can be infinitely many ways in which an error can be inflicted on a qubit. A fortunate fact is that any error operator  $E$  can be written as a linear combination of the basic operators of identity  $I$ , Pauli  $X$ , Pauli  $Z$ , and  $XZ$ :

$$E = \alpha I + \beta X + \gamma Z + \delta XZ,$$

for some constants  $\alpha, \beta, \gamma, \delta$ . This decomposition allows the arbitrary quantum error  $E$  to be discretized into only four possible error operators. The  $X$  Pauli matrix is responsible for the bit flip of a qubit and the  $Z$  Pauli matrix is responsible for the phase error of a qubit as shown previously. When both these errors act together, they can be of the form  $XZ$  or  $ZX$ .

**Bit flip errors** Suppose the system is initially  $|\Psi\rangle = a|0\rangle + b|1\rangle$ , then we can encode  $|0\rangle$  to  $|000\rangle$  and  $|1\rangle$  to  $|111\rangle$  using CNOT gates and ancillary qubits in the following way.

Once the qubit is encoded and transmitted, it needs to be decoded correctly. We assume that an error strikes, it will strike just one of the three qubits. Right now the system is in the following state:

$$|\Psi\rangle = a|000\rangle + b|111\rangle$$

The error can then strike on any of the three positions; the projections of the errors can be represented in a set and is used as the measurement set to measure the system. The following are the projections or error/measurement syndrome.

When no error has struck:

$$P_0 = |000\rangle\langle 000| + |111\rangle\langle 111|$$

When error strikes the first qubit:

$$P_1 = |100\rangle\langle 100| + |011\rangle\langle 011|$$

When error strikes the second qubit:

$$P_2 = |010\rangle\langle 010| + |101\rangle\langle 101|$$

When error strikes the third qubit:

$$P_3 = |001\rangle\langle 001| + |110\rangle\langle 110|$$

Suppose the system is struck with an error on the first qubit, then

$$|\Psi'\rangle = a|100\rangle + b|011\rangle$$

On applying the above projections on the system, the following happens.  $P_0|\Psi'\rangle = 0$ , because the projection is orthogonal to the vectors  $|100\rangle$  and  $|011\rangle$ .

The same follows for  $P_2$  and  $P_3$ . But  $P_1|\Psi\rangle$  yields  $|\Psi\rangle$ , so we do not need to change the system and we know at which position the error has occurred. Now all one needs to do is to apply the bit flip operator  $X$  and get back the original state  $|\Psi\rangle$ .

There is another way at looking at it. Suppose we perform the  $Z$  operation on the first two qubits, represented as  $Z_1Z_2$  which equals

$$(|00\rangle\langle 00| + |11\rangle\langle 11|) \otimes I - (|01\rangle\langle 01| + |10\rangle\langle 10|) \otimes I$$

it can be thought of as comparing the first two qubits and resulting in a +1 if the values are same and -1 if not. One can then perform a measurement  $Z_2Z_3$  on the second and third qubits and compare the values in the same way. If  $Z_1Z_2$  gave a -1 and  $Z_2Z_3$  gave a +1, then obviously the first bit has been flipped.

**Phase flip errors** The phase flip error has no classical equivalent, however the basis states of  $|0\rangle$  and  $|1\rangle$  can be changed to  $|+\rangle$  and  $|-\rangle$  defined in Section 3.2 under Unitary Transformations. So we encode the system  $|\Psi\rangle$  exactly as we did earlier using CNOT gates and ancillary variables, then we hit each qubit with the Hadamard gate. As a result,  $|\Psi\rangle = a|+++ \rangle + b|--- \rangle$ .

One can easily see the benefit of doing this. On applying the  $Z$  gate on any of the qubits,  $|+\rangle$  changes to  $|-\rangle$  and vice versa, as a result acting like the bit flip error. So, we handle the phase flip error just like the bit flip errors. Thus, the measurement/error syndrome is analogous. The difference being that  $|+\rangle$  replaces  $|0\rangle$  and  $|-\rangle$  replaces  $|1\rangle$ .

### 3.2 Shor's Quantum Code

The first example of a quantum error correction code was given by Shor [2]. Shor's construction shows that measurement can be used judiciously in performing quantum error-correction tasks. The qubit  $|0\rangle$  is encoded to  $|+++ \rangle$  and the qubit  $|1\rangle$  is encoded to  $|--- \rangle$ , using the Hadamard gate as in the case for handling phase errors.

Next we encode  $|+\rangle$  to  $\frac{1}{\sqrt{2}}(|000\rangle + |111\rangle)$  and  $|-\rangle$  to  $\frac{1}{\sqrt{2}}(|000\rangle - |111\rangle)$ , as a result we get a 9-qubit code.

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

$$|1\rangle \longrightarrow \frac{1}{2\sqrt{2}}(|000\rangle - |111\rangle)(|000\rangle - |111\rangle)(|000\rangle - |111\rangle)$$

This code is able to protect against a single phase or bit flip error. In the case of a bit flip error, we perform the measurement  $Z_1Z_2$  to compare the first two qubits and then  $Z_2Z_3$  to compare the last two qubits. This can be done again till all the nine qubits are compared.

Phase errors are handled analogous to the phase flip error modeled described earlier. If a qubit is phase flipped, it simply changes from  $|000\rangle + |111\rangle$  to  $|000\rangle - |111\rangle$ . Phase error is detected by comparing the signs of two blocks of qubits at a time.

### 3.3 Classical Linear Codes

In this section we briefly describe the basic theory of linear codes. We focus on *binary* codes where the alphabet used for encoding is  $\{0, 1\}$ . But first we state some terminology that will be used in relation to the vector space  $\mathbb{Z}_2^n$ . The vector space  $\mathbb{Z}_2^n$  represents the space of all  $n$ -bit sequences along with the vector addition operation and the scalar multiplication over the trivial Boolean domain  $\mathbb{Z}_2$ .

| Notation    | Description   |
|-------------|---|
| $x_j$       | the $j$ -th bit of $x \in \mathbb{Z}_2^n$   |
| $ x $       | <i>weight</i> of $x$ or the number of 1s in $x \in \mathbb{Z}_2^n$                  |
| $x + y$     | the <i>bitwise Exclusive OR</i> of $x, y \in \mathbb{Z}_2^n$                        |
| $x \cdot y$ | the Boolean inner product of $x, y \in \mathbb{Z}_2^n$ or $\sum_j x_j y_j \pmod{2}$ |
| $d(x, y)$   | the <i>distance</i> between $x$ and $y$ or $ x + y $                                |

Figure 2: Terminology for linear codes over  $\mathbb{Z}_2^n$ .

In a classical binary  $(n, k)$  linear code  $C$ , the set of  $k$ -bit vectors into  $n$ -bit vectors, where  $n > k$ .  $C$  is said to have distance  $d$  if the distance between each  $n$ -bit vectors of  $C$  is at least  $d$ . Sometimes we will use the notation  $(n, k, d)$  to denote a  $(n, k)$  linear code with distance  $d$ . The  $k$  bits are represented in vector form as a matrix. The matrix used to encode this vector is called the Generator matrix,  $G$  with dimension  $n \times (n - k)$ . The matrix to detect errors in the encoded message is called the parity check matrix  $H$ .

Essentially, the code  $C$  is transferred from a vector space of  $k$  bits and embedded into a vector space of  $n$  bits in such a way that the distance between each bit in the new vector space is  $d$ . The number of errors that such a code can detect is then  $d/2$ .

The  $k$  bits are treated as a vector and represented by a column vector

$X = [x_1 \ \dots \ x_k]^T$ . The generator matrix  $G$  then encodes this matrix and the result is  $GX = V$ . The encoded message is then transmitted and the original data map and the transmitted data map are compared. To check which bit was hit by the error, the parity check matrix  $H$  is multiplied by  $V$ . If there was no error on transmission, the resulting matrix will be a 0, else it will be binary equivalent of the index of the bit in  $V$  where there was an error.

The *distance* of a linear code  $C$  is defined as the minimum distance between all pairs of codewords in  $C$ , i.e.,

$$d(C) = \min\{d(u, v) \mid u, v \in C\}.$$

### 3.4 CSS Quantum Codes

To understand how the CSS code is used, we require the notion of the dual of a linear code.

**Definition 2** *The dual  $C^\perp$  of a linear code  $C$  with generator matrix  $G$  and parity check matrix  $H$  is a linear code with parity check matrix  $H^\perp = G^T$  and generator matrix  $G^\perp = H^T$ . The codewords of  $C^\perp$  consists of vectors  $u$  such that  $u \cdot v = 0$  for all  $v \in C$ .*

*If  $C \subset C^\perp$  then the code  $C$  is called weakly self-dual or self-orthogonal. If  $C = C^\perp$  then  $C$  is called strictly self-dual.*

Next, we describe an important coding lemma associated with codes and their duals is the following.

**Lemma 3** *Let  $C$  be a linear code and  $C^\perp$  be its dual. If  $x \in C^\perp$ ,  $\sum_{y \in C} (-1)^{x \cdot y} = |C|$ . If  $x \notin C^\perp$ ,  $\sum_{y \in C} (-1)^{x \cdot y} = 0$ .*

**Proof.** When  $x \in C^\perp$ ,  $x \cdot y = 0$ , hence  $\sum_{y \in C} (-1)^{x \cdot y} = \sum_{y \in C} (-1)^0 = |C|$ . When  $x \notin C^\perp$ ,  $x \cdot y = 0$  or  $x \cdot y = 1$ , let there be a  $z$  in  $C$ , such that  $x \cdot z = 1$ , then there exists a  $z''$  such that  $z'' = z + z$ , as a result,  $x \cdot z'' = 0$ . Thus the number of 1 obtained from  $(-1)^{x \cdot z''}$  is cancelled by the number of  $-1$  obtained from  $(-1)^{x \cdot z}$ . Similarly those terms where  $x \cdot y = 0$ , when  $y \in C$  there exists a  $y''$ , such that  $y'' = y + z$  as a result  $x \cdot y'' = 1$ . Thus the number of 1 obtained in the sum from  $x \cdot y = 0$  cancels the number of  $-1$  obtained in the sum when  $x \cdot y'' = 1$ .

■

**Definition 4** *(Calderbank-Shor-Steane (CSS) quantum codes)*

*Let  $C_1$  and  $C_2$  be two binary linear codes that satisfy the following conditions:*

1.  $C_1$  is a  $(n, k_1)$  code and  $C_2$  is a  $(n, k_2)$  code.
2.  $C_2 \subset C_1$ .
3. both  $C_1$  and  $C_2^\perp$  correct  $t$  bit errors.

Then, the Calderbank-Shor-Steane (CSS) quantum code construction creates a quantum code  $CSS(n, k_1 - k_2)$  using the quantum states

$$|x + C_2\rangle = \frac{1}{\sqrt{|C_2|}} \sum_{y \in C_2} |x + y\rangle,$$

for each  $x \in C_1$ . This quantum code can correct up to  $t$  qubit errors.

Here,  $k_1 - k_2$  denotes the number of qubits that can be protected,  $n$  denotes the number of qubits that will be needed to encode  $k_1 - k_2$  qubits, and  $t$  is the number of bit flips or phase flips or both that can be handled for  $n$  encoded qubits. The goal in creating an effective CSS construction using  $C_1$  and  $C_2$  is to achieve large  $k_1 - k_2$  and  $t$ .

Note that if  $x, x'' \in C_2$ , then  $|x + C_2\rangle = |x'' + C_2\rangle$ .

Let  $x - x'' = z \in C_2$ . Then

$$|x + C_2\rangle = \frac{1}{\sqrt{|C_2|}} \sum_{y \in C_2} |x'' + z + y\rangle,$$

but  $z \in C_2$ , so  $|x + C_2\rangle = |x'' + C_2\rangle$ .

Assuming that both a bit flip error and a phase error occur, the one-qubit system  $|0\rangle$  is encoded in the following manner.

$$|x + C_2\rangle|0\rangle = \frac{1}{\sqrt{|C_2|}} \sum_{y \in C_2} |x + y\rangle|0\rangle.$$

When the error  $e_1$  (bit flip) and  $e_2$  (phase error) strike, then one could represent the system in the following way:

$$\frac{1}{\sqrt{|C_2|}} \sum_{y \in C_2} (-1)^{(x+y)e_2} |x + y + e_1\rangle|0\rangle.$$

Let us first handle error  $e_1$ , we strike the system with the parity check matrix of  $C_1$ ,  $H_1$  and measure the resulting qubit. Since  $x + y \in C$ ,  $|H(x + y + e_1)\rangle = 0 + |He_1\rangle$  implies  $|x + C_2\rangle = \frac{1}{\sqrt{|C_2|}} \sum_{y \in C_2} (-1)^{(x+y)e_2} |x + y + e_1\rangle|He_1\rangle$ .

Because of the parity check matrix, applied on each qubit, one knows the error position for each of them and one can simply apply the Pauli  $X$  matrix and eliminate  $e_1$ . After  $e_1$  is eliminated, we get

$$\frac{1}{\sqrt{C_2}} \sum_{y \in C_2} (-1)^{(x+y)e_2} |x+y\rangle.$$

Now we need to handle the phase error  $e_2$ . For this we need to apply the Hadamard gate on all the qubits. Now,

$$|x\rangle \xrightarrow{H} \frac{|0\rangle + (-1)^x |1\rangle}{\sqrt{2}},$$

so on tensoring  $n$  times, we get

$$|x\rangle \xrightarrow{H^{\otimes n}} \frac{1}{\sqrt{2^n}} \sum_z (-1)^{x \cdot z} |z\rangle$$

where  $z$  is being summed over all of  $\{0, 1\}^n$ . So the qubits after being hit by the Hadamard matrix can be represented as

$$\frac{1}{\sqrt{C_2}} \sum_z \sum_{y \in C_2} (-1)^{(x+y)(e_2+z)} |z\rangle$$

Now, suppose  $z'' = z + e_2$ . Then the expression can be rewritten as

$$\frac{1}{\sqrt{C_2}} \sum_{z''} \sum_{y \in C_2} (-1)^{(x+y)(z'')} |z'' + e_2\rangle.$$

Now we can separate the powers  $x$  and  $y$ . So we use Lemma 3 to obtain

$$\frac{1}{\sqrt{2^n/C_2}} \sum_{z'' \in C_2^\perp} (-1)^{x \cdot z''} |z'' + e_2\rangle.$$

Here  $e_2$  can easily be eliminated like  $e_1$ , using the parity check matrix of  $C_2^\perp$ . So we get

$$\frac{1}{\sqrt{2^n/C_2}} \sum_{z'' \in C_2^\perp} (-1)^{x \cdot z''} |z''\rangle.$$

Now we know that Hadamard gates are unitary operators which are reversible, so on reapplying the Hadamard matrix, we should get back the original pure state of the qubits, i.e.,

$$\frac{1}{\sqrt{C_2}} \sum_{y \in C_2} |x+y\rangle.$$

## 4 Explicit Codes

In this section, we discuss several classical linear codes that are relevant for creating quantum codes. First, we describe the class of Hamming codes and its dual (which is the class of Simplex codes). Then, we describe Steane's construction of a quantum code that uses the Hamming code  $(7, 4, 3)$  and its dual. Finally, we briefly discuss a class of linear codes with better error-correcting capabilities, namely, the Golay codes.

### 4.1 Hamming Codes

The Hamming code  $H_k$  is a  $(2^k - 1, 2^k - 1 - k)$  linear code. The parity check matrix of  $H_k$  is a  $k \times (2^k - 1)$  matrix whose columns contain all non-zero  $k$ -bit sequences.

In the following, we describe as an example the Hamming code  $H_3$  that is a  $(7, 4, 3)$  linear code. Here the parity check matrix for  $H_3$  is:

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

As one can see, the first, second, and fourth columns of the matrix are linearly independent, so this code can detect up to  $\lfloor \frac{3}{2} \rfloor = 1$  error. Note that we require  $HG = 0$ .

After matrix manipulation of  $H$ ,  $H$  can be represented as two matrices  $A$  and  $I$ .

$$H = [A_{(n-k) \times k} | I_{(n-k) \times (n-k)}].$$

To make  $HG = 0$ ,

$$G = \begin{bmatrix} I_{k \times k} \\ -A \end{bmatrix}.$$

Also note that since we are dealing with just 0 and 1, the negative sign does not matter.

As an example suppose

$$X = [0 \ 1 \ 1 \ 1]^T.$$

After encoding with  $G$ ,

$$V = [0 \ 1 \ 1 \ 1 \ 1 \ 0 \ 1]^T$$

Now one can easily check that  $HV = 0$ .

Suppose there is an error in the 4-th bit, then this error can also be represented as a matrix,

$$E = [0 \ 0 \ 0 \ 1 \ 0 \ 0 \ 0]^T$$

Then the data struck with this error will be  $V'' = V + E$ . So  $H(V + E) = HV + HE$ .

$$HE = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}$$

Note that 100 is the binary representation of 4, so it indicates that the error has struck on the 4-th bit.

**Theorem 5** For all  $k \geq 2$ , the Hamming code  $H_k$  has distance 3.

**Proof.** Recall that the columns of the parity check matrix for  $H_k$  consist of all non-zero  $k$ -bit sequences. It suffices to show that any non-zero codeword must have weight at least 3. Let  $x$  be a non-zero codeword of  $H_k$ . The weight of  $x$  is not 1 since the all-zero column is excluded and it is not be 2 since no two columns are identical. Thus, the weight of  $x$  must be at least 3. It is easy to find a codeword with weight exactly 3. Thus,  $H_k$  has a distance of 3. ■

## 4.2 Simplex Codes

The Simplex codes are the dual of the Hamming codes. They have the same properties as any other dual of classical linear codes as well as some specific properties. The Simplex codes are equidistant codes, that is the distance between each codeword is the same. The other more important property is the property of self-orthogonality. A code  $C$  is termed self-orthogonal when  $C \subseteq C^\perp$ . All Simplex codes are self-orthogonal for  $n$ -ary codeword domain with  $n \geq 3$ .

## 4.3 Steane's Quantum Code

The Steane quantum code uses the  $(7, 4, 3)$  Hamming Code and its dual as  $C_1$  and  $C_2$ , respectively, in the CSS construction. Since  $C_1$  is  $(7, 4)$  and  $C_2$  is  $(7, 3)$ ,  $CSS(C_1, C_2)$  is  $(7, 1)$  which corrects up to 1 error.

The codewords for this code are:

$$\begin{aligned}
 |0\rangle &= (|0000000\rangle + |1010101\rangle + |0110011\rangle + |1100110\rangle + \\
 &\quad |0001111\rangle + |1011010\rangle + |0111100\rangle + |1101001\rangle) \\
 |1\rangle &= (|1111111\rangle + |0101010\rangle + |1001100\rangle + |0011001\rangle + \\
 &\quad |1110000\rangle + |0100101\rangle + |1000011\rangle + |0010110\rangle)
 \end{aligned}$$

However, the Hamming Codes always generate a CSS code which can only correct one error. In the next section we shall look at a linear code which can potentially correct larger amount of errors.

The Steane quantum code is a beautiful construction that exploits the weakly self-dual property of the Hamming code  $H_3$ . It is possible that there are other classes of codes that can be used in the Steane construction.

#### 4.4 Golay codes

We consider the class of Golay codes in exploring potential construction of quantum codes with larger error-correcting capabilities, i.e.,  $T > 1$ . The Golay  $G_{24}$  [8] is constructed from the set of codewords formed by the  $(7, 4, 3)$  Hamming Code and a variation of the codewords say  $K$  which is obtained by simply reversing the order of the bits. To both these sets of codewords,  $H$  and  $K$  a parity check bit is added to make the number of 1's even. So, the resulting codewords,  $H^+$  and  $K^+$  are as follows.

| $H^+$             | $K^+$             |
|-------------------|-------------------|
| 00000000 11111111 | 00000000 11111111 |
| 11010001 00101110 | 00010111 11101000 |
| 01101001 10010110 | 00101101 11010010 |
| 00110101 11001010 | 01011001 10100110 |
| 00011011 11100100 | 10110001 01001110 |
| 10001101 01110010 | 01100011 10011100 |
| 01000111 10111000 | 11000101 00111010 |
| 10100011 01011100 | 10001011 01110100 |

The result is two  $(8, 4)$  codes with a minimum distance of 4. The codewords for the  $G_{24}$  are of the form  $a + x$ ,  $b + x$ ,  $a + b + x$ , where  $a, b \in H^+$  and  $x \in K^+$ . Therefore an example of a Golay codeword would be

$$100010001100111100011110$$

Next we describe the generator matrix for the  $G_{24}$  code. For this we require the Generator matrices for  $H^+$  and  $K^+$ . The generator matrix for  $H^+$  is

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

which is obtained by adding a 1 in front of the row of generator matrix of the  $(7, 4, 3)$  Hamming code, if the number of 1's in that row are even and 0 if they are odd.

The generator matrix for  $K^+$  is obtained by reversing the order of the first seven rows for the above matrix.

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

The generator matrix for  $G_{24}$  can be obtained by using the encoding formula,  $a + x$ ,  $b + x$ ,  $a + b + x$ , we assume that out of the three variables, exactly

one is non-zero and we get a  $24 \times 12$  matrix.

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

The Golay codes could be used in the CSS construction in a similar manner to the Steane construction. However, Golay codes are strictly self-dual. Thus, Golay codes are not immediately applicable in a Steane-like CSS construction. But this does not rule out a combination of Golay codes with other linear codes.

## 5 Conclusion

The goal of this thesis was to discover quantum codes with different characteristics. Recall that Steane's  $(7, 1)$  quantum code can encode  $K = 1$  qubit using  $N = 7$  qubits and is robust against  $T = 1$  qubit errors. So, the three relevant parameters are the number of qubits  $N$  required to encode a certain number of  $K$  qubits while being robust against  $T$  qubit errors.

We planned to generalize the Steane quantum code to other values of  $N$ ,  $K$ , and  $T$ . For example, we could use the Hamming code  $H_k$  and its dual  $H_k^\perp$ . It turns out that in the ternary case, the critical condition  $H_k^\perp \subseteq H_k$  is satisfied. But this requires an encoding scheme that works on qutrits (instead of qubits). Although this could *not* be used for binary codewords, however ternary codewords could potentially be used in the CSS construction. For this, more investigation is required as to whether the error-correction capabilities of the CSS codes are restricted to the binary domain. If ternary or higher domain codewords are acceptable, then a family of CSS codes can emerge from the Hamming Codes. We leave this question for future work.

## References

- [1] P.W. Shor. Algorithms for quantum computation: discrete logarithms and factoring. *Proc. 35th Annual Symposium on Foundations of Computer Science*, pp. 124-134, 1994.
- [2] P.W. Shor. Scheme for reducing decoherence in quantum computer memory. *Phys. Rev. A*, **52**:2493, 1995.
- [3] A.R. Calderbank and P.W. Shor. Good quantum error-correcting codes exist. *Phys. Rev. A*, **54**:1098, 1996.
- [4] R. P. Feynman. Simulating physics with computers. *Int. J. Theor. Phys.*, **21**:467, 1982.
- [5] A. Einstein, B. Podolsky, and N. Rosen. Can quantum-mechanical description of physical reality be considered complete? *Phys. Rev.*, **47**:777-780, 1935.
- [6] M. Nielsen and I. Chuang. *Quantum Computation and Quantum Information*. Cambridge University Press, 2000.
- [7] A. Steane. Error correcting codes in quantum theory. *Phys. Rev. Lett.*, **77**:793, 1996.
- [8] O. Pretzel. *Error-Correcting Codes and Finite Fields*. Oxford University Press, 1992