An Incomplete Introduction to Quantum Error Correction

Matthias Reumann

Technische Universität München, Munich, Germany

Quantum Error Correction is an essential building block towards Fault-Tolerant Quantum Computing. This paper introduces the theoretical minimum required for the former and briefly introduces the latter. We do so in a beginner-friendly and graphically-supplemented fashion.

1 Introduction

Quantum computing is a hot topic. Feynman's dreams of a quantum computer in 1982 [10] are today's promises of companies such as IBM, Google, and Microsoft [1, 2, 16]. The ultimate goal is to run algorithms on these hardware devices. Similar to World War 2, applications in cryptography demonstrate the usefulness of quantum computing technology [22]. Namely, Shor's prime factoring algorithm and its consequences on current cryptographic systems [24]. Over time, a wide range of algorithms with speed-ups over their classical counterparts emerged. Another critical field of research lies between the advancements in quantum hardware and the invention of quantum algorithms: quantum error correction. Its task is to protect delicate qubits from the adverse effects of errors. Error sources include qubit-environment interactions, so-called *decoherence*, and faulty gates.

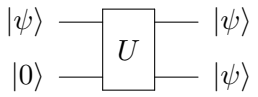
In this paper, we review the fundamental concepts of quantum error correction in an intuitive and easily digestible way. For the content we primarily rely on already existing, but math-heavy, introductions such as Ref. [14] and textbooks such as Ref. [21]. Nevertheless, we always reference source materials as well as contemporary research results for a deeper dive into the literature. A main achievement of this paper is the supplementation of existing resources with graphical intuition.

Matthias Reumann: matthias.reumann@tum.de

2 Challenges

Even though quantum error correction often borrows ideas from its classical counterpart, quantum error correction faces the following unique challenges.

No-Cloning Theorem The no-cloning theorem states that no unitary operation U exists that copies an unknown quantum state $|\psi\rangle$. Consequently, quantum analogs of classical repetition codes, where one copies individual bits and takes a majority vote, are infeasible.



Continous Errors A qubit can be in any superposition of the basis states $|0\rangle$ and $|1\rangle$.

$$\alpha\,|0\rangle + \beta\,|1\rangle \tag{1}$$

where α and β are complex numbers that fulfill the constraint $|\alpha|^2 + |\beta|^2 = 1$. Thus, a quantum computer has a continuum of states. Ordinary classical computers are binary - a bit is either 0 or 1, even after a bit-flip error. In comparison, it seems like quantum error correction requires an infinite amount of precision to detect all possible errors.

Let's look at a concrete and practically relevant example. Imagine that a quantum computer implements the following rotation operation.

$$\begin{pmatrix} 1 & 0 \\ 0 & e^{i\theta} \end{pmatrix} \tag{2}$$

However, due to some miscalibration, it ends up rotating not by θ but $\theta + \delta$. Figure 1 depicts this scenario on the *bloch sphere* for a $\frac{\pi}{2}$ rotation. Throughout the computation, we apply this faulty operation many times. How small δ may be, the accumulation of errors ultimately leads to inaccurate results. A reasonable idea therefore is to detect and correct this error. The difficulty lies

in the fact that δ can be any real number, leaving us with the problem stated in the previous paragraph.

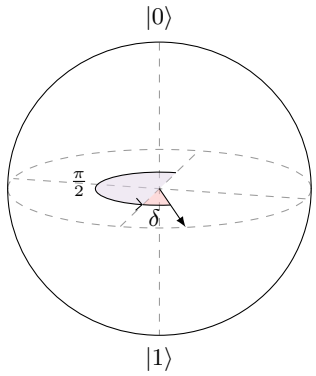


Figure 1: Over-rotation by δ for a desired $\frac{\pi}{2}$ rotation on the bloch sphere.

Measurements destroy quantum information Measurements in quantum mechanics destroy the superposition of a quantum state. Contrary to the classical approach, we can not recover the original superposition after observing the state.

Given these three challenges, quantum error correction seems like a daunting, if not impossible, task. Fortunately, as we will illustrate in the succeeding sections, each challenge can be overcome.

3 Physical & Logical Qubits

By the rules of quantum mechanics, we can't utilize redundancy in the form of repetition codes to protect qubits. However, there are also phenomenons with no classical equivalent such as entanglement. In 1995, Shor was one of the first to take advantage of this quantum effect for quantum error correction.

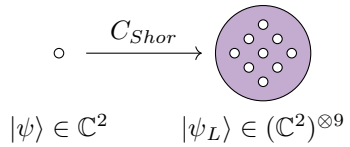


Figure 2: Encoding of one physical qubit $|\psi\rangle$ into one logical qubit $|\psi_L\rangle$. The logical qubit entangles multiple physical qubits.

The Shor code stores the information of a single *logical* qubit in the entanglement between nine *physical* ones [23]. Consequently, no specific (possibly faulty) qubit stores all the state's information. Figure 2 depicts this procedure schematically. The underlying idea is borrowed from classical codes: Use many to protect the few. But entangle qubits instead of cloning.

Equations 3-4 define the Shor code and its logical qubits mathematically. Notice that the qubit triplet in the brackets is the Greenberger-Horne-Zeilinger (GHZ) state, a maximally entangled state.

$$|0_L\rangle = \left(\frac{|000\rangle + |111\rangle}{\sqrt{2}}\right)^{\otimes 3} \tag{3}$$

$$|1_L\rangle = \left(\frac{|000\rangle - |111\rangle}{\sqrt{2}}\right)^{\otimes 3} \tag{4}$$

$$\mathcal{Q}_{shor} = \{|0_L\rangle, |1_L\rangle\} \tag{5}$$

It should be no surprise that the encoding circuit for the Shor code stacks three GHZ circuits. We refer the reader to Ref. [20], which implements the Shor code on a trapped-ion device, to validate this fact.

4 Pauli matrices & the Pauli group

Equations 6-9 define the famous *Pauli matrices*. In the quantum circuit model, the Pauli matrices are referred to as *Pauli gates*.

$$I = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \quad \text{“identity”} \tag{6}$$

$$X = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \quad \text{“bit flip”} \tag{7}$$

$$Y = iXZ = \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix} \quad \text{“bit \& phase flip”} \tag{8}$$

$$Z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix} \quad \text{“phase flip”} \tag{9}$$

A useful fact that will be of significance later is that X, Y, and Z anti-commute. That is, $XZ = -ZX$, written as $\{X, Z\} = 0$, and similarly for any other pair.

The Pauli matrices generate the *Pauli group* $\mathcal{P}_n$. Mathematically, it is the n-fold tensor product of the four Pauli matrices with factors ± 1, and $\pm i$. The factors are necessary to build a valid group. Any two elements of the Pauli group either commute or anti-commute, where one defines commutation as $AB = BA$ and denotes it as $[A, B] = 0$.

These matrices are important because they span the space of all 2×2 matrices. Likewise, the Pauli group spans the space of $2^n \times 2^n$ matrices. Thus, any error E can be expanded as a linear combination of these matrices.

$$E = aI + bX + cY + dZ \tag{10}$$

Now, assume a single-qubit error occurs on the first qubit of many. The resulting state $E_1 |\psi\rangle$ is also a linear combination of the terms $I_1 |\psi\rangle$, $X_1 |\psi\rangle$, $Y_1 |\psi\rangle$, and $Z_1 |\psi\rangle$. Then, after a measurement, the state would collapse to one of these terms. If we knew with which term we ended up, we could correct the error by applying the respective Pauli gate. In the next section, we introduce a formalism that underlies most of today's quantum codes and enables such a detection mechanism.

5 The Stabilizer Formalism

The stabilizer formalism is an important tool for building modern quantum codes. Its fundamental idea resembles parity check matrices from classical coding theory. This section explores this resemblance and introduces stabilizer codes formally. Stabilizers have been pioneered by Daniel Gottesman in his Ph.D. thesis [12].

5.1 Hamming Codes

Hamming codes use parity-check matrices to calculate *syndrome vectors*.

$$s = Hx \pmod 2 \tag{11}$$

If a syndrome vector s equals zero, no error occurred. Otherwise, s indicates which bit has been flipped. Equation 12 shows the parity-check matrix H for the $[7, 4, 3]$ code.

$$H = \begin{pmatrix} r_1 \\ r_2 \\ r_3 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 \end{pmatrix} \tag{12}$$

Figure 3 illustrates the dot-product between each row of H and $x = 0111010$. Note that each dot-product $r_{i \in \{1,2,3\}} x \pmod 2$ yields one bit of the syndrome vector and splits the codeword into a violet (•) and white (○) group. The resulting syndrome vector is $s = (1\ 1\ 0)$, or intuitively (• • ○). Consequently, s tells us the error is in the third bit. The corrected codeword is $\hat{x} = 0101010$.

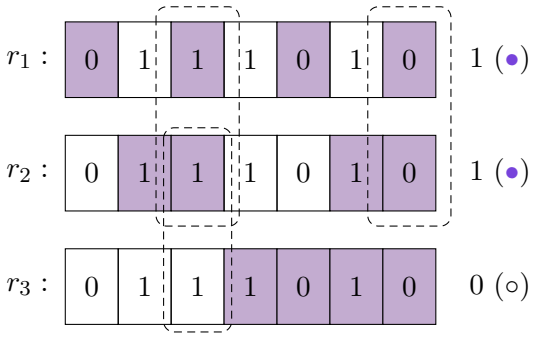


Figure 3: Application of the parity check matrix $H_{[7,4,3]}$ on 0111010 and the resulting syndrome vector (1 1 0). The overlap of the dashed rectangles illustrates the error detection procedure.

5.2 Stabilizer Codes

A matrix $M \in \mathcal{P}_n$ stabilizes the state $|\psi\rangle$ if $M |\psi\rangle = +1 |\psi\rangle$. In other words, $|\psi\rangle$ is an eigenvector of M with eigenvalue $+1$. The states stabilized by a set S of commuting matrices M_i define the codewords of a quantum code $\mathcal{Q}$, where $-I \notin S$.

$$\mathcal{Q} = \{|\psi\rangle : M_i |\psi\rangle = |\psi\rangle, \forall M_i \in S\} \tag{13}$$

Remember that in Section 4 we learned that any two elements of the Pauli group either commute or anti-commute. Furthermore, we can assume that $E \in \mathcal{P}_n$. Hence, either Equation 14 or Equation 15 holds.

$$M_i(E |\psi\rangle) = EM_i |\psi\rangle = +1(E |\psi\rangle) \tag{14}$$

$$M_i(E |\psi\rangle) = -EM_i |\psi\rangle = -1(E |\psi\rangle) \tag{15}$$

Using this fact, we define the syndrome $s = (s_1, s_2, \ldots, s_{|S|})$ associated with an error E is as follows.

$$s_i = \begin{cases} 0 & \text{if } [E, M_i] = 0 \\ 1 & \text{if } \{E, M_i\} = 0 \end{cases} \tag{16}$$

This tells us that we can detect any error E if we measure a set of carefully chosen stabilizers M_i.

Let's return to the Shor code. Namely, its stabilizers as defined in Table 1. For brevity, we have omitted the Kronecker product between the Pauli matrices. The *weight* of a stabilizer is the amount of non-trivially acting Pauli matrices, e.g. $wt(M_7) = 6$. Furthermore, using Equation 13 we can alternatively define the logical states of the Shor code in terms of these stabilizers. That is, $\{|\psi\rangle : M_i |\psi\rangle = |\psi\rangle\} = \mathcal{Q}_{shor}$.

	1	2	3	4	5	6	7	8	9
$M_1 =$	Z	Z	I	I	I	I	I	I	I
$M_2 =$	I	Z	Z	I	I	I	I	I	I
$M_3 =$	I	I	I	Z	Z	I	I	I	I
$M_4 =$	I	I	I	I	Z	Z	I	I	I
$M_5 =$	I	I	I	I	I	I	Z	Z	I
$M_6 =$	I	I	I	I	I	I	I	Z	Z
$M_7 =$	X	X	X	X	X	X	I	I	I
$M_8 =$	I	I	I	X	X	X	X	X	X

Table 1: The eight stabilizers for the Shor code. Kronecker products between Pauli matrices omitted for brevity.

Each measurement of a stabilizer yields one syndrome bit. Combining the information of all measurements tells us not only if an error happened but also which one and where. Let us walk through a simple example to see this procedure in action. Let $|x\rangle$ be the $|0_L\rangle$ codeword, where we bit-flip the third qubit:

$$\left(\frac{|001\rangle + |110\rangle}{\sqrt{2}}\right) \otimes \left(\frac{|000\rangle + |111\rangle}{\sqrt{2}}\right)^{\otimes 2} \tag{17}$$

The measurement of all stabilizers M_1 to M_8 yields syndrome vector

$$s = (0\ 1\ 0\ 0\ 0\ 0\ 0\ 0) \tag{18}$$

or alternatively color-coded ($\circ\ \bullet\ \circ\ \circ\ \circ\ \circ\ \circ\ \circ$). A few observations follow. First of all, because it is not the zero vector some error occurred. Secondly, M_2 is the only stabilizer acting non-trivially on $|x\rangle$ which results in a syndrome bit of 1. Lastly, as with classical parity checks combining the information of all syndrome bits tells us that an X error occurred on the third qubit. Thus, applying a X gate on the third qubit corrects the error. Figure 4 depicts this procedure graphically. A comparison to Figure 3 might be illuminating. We leave it up to the reader to validate this procedure for Y and Z errors.

5.3 Degenerate Codes

A quantum code is *degenerate* if two distinct errors E and F have the same syndrome vector. Otherwise, we call it *non-degenerate*.

A stabilizer code is degenerate if $E^\dagger F \in S$. Intuitively, this means that E and F produce the same error on the same codeword.

$$\begin{aligned} |err\rangle &= F\,|\psi_L\rangle = E\,|\psi_L\rangle \\ &= E^\dagger F\,|\psi_L\rangle = E^\dagger E\,|\psi_L\rangle \\ &= (E^\dagger F)\,|\psi_L\rangle = |\psi_L\rangle \end{aligned} \tag{19}$$

For example, it is easy to see that a Z_1, Z_2, and Z_3 error leads to the same erroneous state for the Shor code. Moreover, Z_1Z_2, Z_2Z_3, and Z_1Z_3 are indeed stabilizers. Hence, the Shor code is degenerate. Interestingly, applying Z_1 (or Z_2 or Z_3) corrects each of these errors. This leads to the conclusion that degenerate codes can correct more errors than they can uniquely identify. Remarkably, this property is unknown to classical codes.

5.4 Syndrome Measurements

Each measurement of a stabilizer M_i tells us one bit of the error syndrome. But how exactly can we measure stabilizers non-destructively?

The idea is the following. First, encode the error syndrome on additional qubits, called *ancilla qubits* without disturbing the actual *data qubits*. Then, measure the ancilla qubits and evaluate the error syndrome.

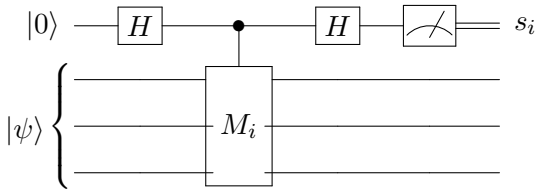


Figure 5: The phase-kickback circuit.

The most straightforward way of achieving this is using a *phase kickback* construction. We depict its circuit in Figure 5. The controlled-M_i gate can be constructed from multiple controlled-P_i gates, where P_i is a Pauli gate acting on the ith qubit. Notice that the state after the controlled operation is

$$\frac{|0\rangle\,|\psi\rangle + |1\rangle\,M_i\,|\psi\rangle}{\sqrt{2}}. \tag{20}$$

Without an error, we can rewrite this state by using the fact that M stabilizes $|\psi\rangle$.

$$\frac{|0\rangle\,|\psi\rangle + |1\rangle\,|\psi\rangle}{\sqrt{2}} = |+\rangle\,|\psi\rangle \tag{21}$$

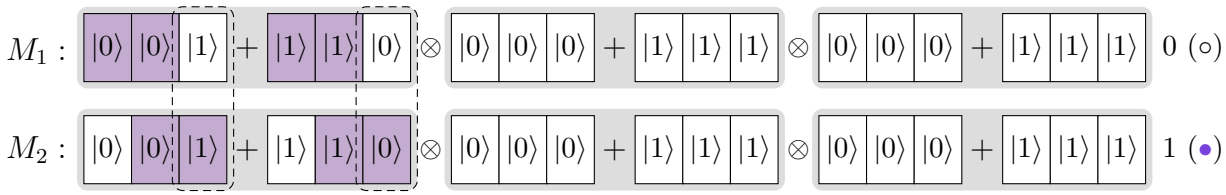


Figure 4: Syndrome measurement of the stabilizers M_1 and M_2 on the $|0_L\rangle$ state for the Shor code where the third qubit has been bit-flipped.

A measurement of the ancilla qubit in the Hadamard basis reveals that the error syndrome bit for this stabilizer is 0. Similarly, for an erroneous state and an appropriate stabilizer, the same measurement gives us a syndrome bit of 1.

$$\frac{|0\rangle (E|\psi\rangle) - |1\rangle (E|\psi\rangle)}{\sqrt{2}} = |-\rangle (E|\psi\rangle) \tag{22}$$

Unfortunately, this procedure isn't fault-tolerant. Any single-qubit error in the controlled-M_i gate can propagate to the ancilla qubit or worse an error on the ancilla results in multiple errors on the data qubits. Needless to say but of course an error on the ancilla might also lead to an incorrect syndrome bit.

It's an open research problem to find a "good" solution for syndrome measurements of stabilizer codes. Error correction schemes proposed by Shor, Steane, and Knill increase the total amount of qubits by the maximum weight of a code's stabilizers. Alternatively, *flag error correction* requires only $d + 1$ ancilla qubits, where d is the distance of a given code [7].

5.5 Logical Gates

At the end of the day, we want to run circuits to perform some kind of computation. However, so far we have only talked about the encoding of logical qubits and how to detect errors on the underlying physical qubits. For computation on logical qubits, we require *logical gates*. A logical gate acts on a logical qubit and takes a valid codeword to another valid codeword. For example, a X_L gate takes the $|0_L\rangle$ state to $|1_L\rangle$ exactly the same way as a X gate takes $|0\rangle$ to $|1\rangle$.

In the stabilizer formalism, a logical gate is a matrix U that commutes with all stabilizers M_i. Define the set of all such U's as $N(S)$ and call it the *normalizer* of S.

$$M_i(U|\psi\rangle) = UM_i|\psi\rangle = U|\psi\rangle \tag{23}$$

Thus, U takes a codeword $|\psi\rangle \in \mathcal{Q}$ to another valid codeword $U|\psi\rangle \in \mathcal{Q}$. Since $U \in S$ takes any codeword to itself, we define the set of logical gates as $N(S)\backslash S$.

Going back to the Shor code, it is easily verifiable that the logical gates in Equations 24-25 act like their physical counterparts.

$$X_L = \bigotimes_{i=1}^{9} Z_i \tag{24}$$

$$Z_L = \bigotimes_{i=1}^{9} X_i \tag{25}$$

Moreover, Equation 23 also tells us that these logical gates are not unique since we can always apply stabilizers M_i to X_L or Z_L.

$$\begin{aligned}(Z_1Z_4Z_7)|\psi\rangle &= (M_2M_4M_6X_L)|\psi\rangle \\ &= X_LM_2M_4M_6|\psi\rangle \\ &= X_L|\psi\rangle\end{aligned} \tag{26}$$

Consequently, $Z_1Z_4Z_7$ is an alternative representation of the X_L gate defined in Equation 24 that uses three instead of nine physical Z gates. This begs the question if there is an "implementation" with less than three physical gates. As it turns out, this isn't possible. For Shor's code, three is the smallest number of physical gates required to take one codeword to another. We call this smallest number the *distance* of a code. This brings us finally to the classification of Shor's code. Namely, it is a $[[9, 1, 3]]$ quantum code, encoding 9 physical qubits into 1 logical qubit with a distance of 3.

6 Universal Quantum Computing

Similarly to classical computers, where any gate can be constructed by the universal logic gates $NAND$ and NOR, we require a *universal gate set* for quantum computers. Any unitary can be approximated in arbitrary precision by circuits constructed from this gate set [21].

6.1 Clifford Gates

The Clifford gates are the unitary matrices H, S, and $CNOT$.

$$H = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} \tag{27}$$

$$S = \begin{pmatrix} 1 & 0 \\ 0 & i \end{pmatrix} \tag{28}$$

$$CNOT = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix} \tag{29}$$

These unitaries generate the Clifford group. A group of unitary matrices that map the group of Pauli matrices to itself under conjugation.

$$\mathcal{C}_n = \{U : U\mathcal{P}_n U^\dagger = \mathcal{P}_n\} \tag{30}$$

This is a special group in many ways. Firstly, the unitaries that generate it, are fundamental building blocks for many quantum algorithms, including the entanglement of two qubits. Moreover, by the definition of stabilizers as elements of the Pauli group, the Clifford group maps stabilizers to other valid stabilizers with potentially different codewords. This fact paves the way to a theorem which we briefly cover in the next section.

6.2 Gottesman-Knill-Theorem

According to the Gottesman-Knill-Theorem, a probabilistic classical computer can efficiently simulate any circuit of Clifford gates initialized with a stabilizer state and measured in the Pauli basis [13]. Bravyi and Kitaev provide a visual addition to the theorem in [4]. They speculate that the transition from classical computing to quantum computing occurs on the boundary of the octahedron in Figure 6. As briefly mentioned before, many circuits can be constructed by the Clifford gates alone. For example, this includes the circuit for the teleportation algorithm. A full discussion of the theorem's implications does not fit here, hence, we refer the interested reader to Ref. [8]. Nevertheless, this beautifully shows that "quantum supremacy" is nuanced.

6.3 The T Gate

The Clifford gates alone do not form a universal gate set. Therefore, we need an additional non-Clifford gate to achieve this. A common choice for this non-Clifford gate is the T gate. Intuitively, the T gate (or any other non-Clifford) unlocks the rest of the Bloch sphere in Figure 6.

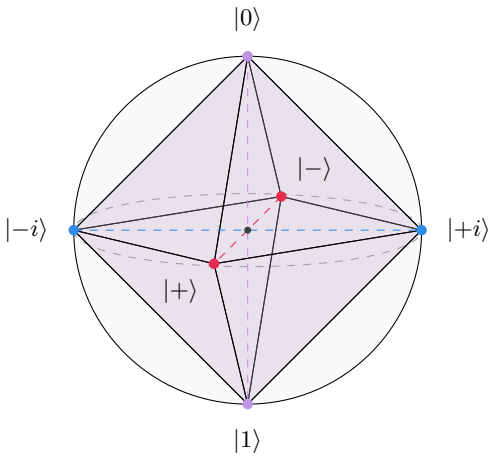


Figure 6: The octahedron spanned by $|0\rangle$ state initialization and application of the Clifford gates. Figure inspired by [3].

$$T = \begin{pmatrix} 1 & 0 \\ 0 & e^{i\pi/4} \end{pmatrix} \tag{31}$$

The T gate is particularly interesting because we can construct Toffoli (CCNOT) gates with it. Its textbook implementation requires seven T gates [21]. However, recent work reduced that number to four [18]. In turn, the importance of the Toffoli gate stems from its usage in quantum algorithms such as Shor's factoring algorithm.

7 The Steane Code

In Section 5 we alluded to the close relationship between classical parity checks and stabilizers. Naturally, this hints at the possibility of constructing quantum codes from classical ones. In this section, we introduce such a code, namely the $[[7, 1, 3]]$ Steane code.

The Steane code belongs to the larger family of CSS codes named after their inventors Calderbank, Shor, and Steane. For the present discussion, a formal introduction to CSS codes is unnecessary and hence we refer the reader to Ref. [21], Ref. [14], or one of the original papers [5].

The code construction stacks two parity check matrices H of the $[7, 4, 3]$ classical hamming code on top of each other. Instead of 1's X and Z matrices are used for phase and bit-flip errors, respectively. Table 2 lists the resulting stabilizers.

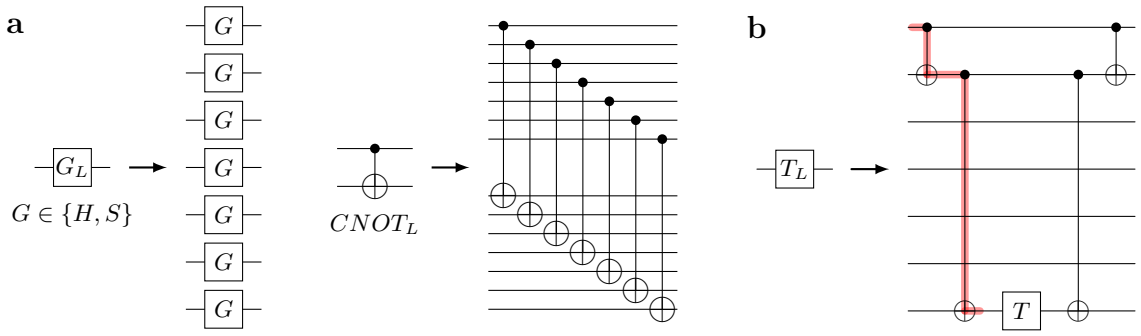


Figure 7: **a)** Transversal implementation of the logical Clifford gates for the Steane code. The logical H_L and S_L gates are trivially transversal because the physical gates act on each qubit individually. For the logical $CNOT_L$ notice that the ith qubit of the first block of seven qubits only interacts with the ith qubit of the second block. **b)** Non-transversal implementation of the T gate for the Steane code. The red line indicates the propagation of a single-qubit error to three different qubits.

Considering Section 5, it should feel somewhat intuitive that such a construction works.

Advantageous for fault tolerance is that the implementation of all logical Clifford gates is *transversal*. That is, the ith physical qubit of a logical qubit only interacts with the ith physical qubit of another logical qubit. This is best seen in the circuits in Figure 7a. This is an important property because we do not want a single error to propagate into multiple ones. Especially, if the error correction code is only able to detect and correct one-qubit errors.

1	2	3	4	5	6	7
Z	I	Z	I	Z	I	Z
I	Z	Z	I	I	Z	Z
I	I	I	Z	Z	Z	Z
X	I	X	I	X	I	X
I	X	X	I	I	X	X
I	I	I	X	X	X	X

Table 2: The stabilizers of the 7-Qubit Steane code.

Unfortunately, as it turns out, there doesn't exist a transversal logical T gate for the Steane code. We depict its non-transversal implementation in Figure 7b. Even worse, the Eastin-Knill-Theorem states that no gate set exists that is both universal and transversal for a quantum code detecting one-qubit errors [9]. Thus, we can not simply find another non-Clifford gate and implement this gate transversally. Consequently, if we want to keep using the Steane code, we need to find alternative techniques to implement the T gate fault tolerantly for universal quantum computing. The next two sections explore two of these techniques, namely, code-switching and magic state distillation.

8 Code Switching

For the Steane code the implementation of the logical T gate is not transversal. However, notice that the circuit in Figure 7b only requires $CNOT$ and T gates. What if we had a second quantum code C_2 that permits a transversal implementation of these gates? Then, if we use seven logical qubits of C_2 instead of seven physical qubits, the resulting circuit is transversal as well. This idea, proposed by Jochym-O'Connor and Laflamme, uses the $[[15, 1, 3]]$ Reed-Muller quantum code as C_2 [17]. The technique of producing larger codes from two smaller ones is called *concatenation*. We depict it visually in Figure 8a.

Previously, single-qubit errors propagate between physical qubits potentially leading to a logical error. Now, single-qubit errors propagate between logical qubits causing a detectable and correctable single-qubit error on each of the logical qubits of the Reed-Muller quantum code (Figure 8b). Consequently, the construction above yields a transversal, i.e. fault-tolerant, implementation of the T_L gate.

Reversely, we implement the transversal Clifford gates of the Steane code with possibly non-transversal gates of the Reed-Muller quantum code. As a consequence of non-transversality, a single-qubit error can propagate to multiple qubits on a given Reed-Muller code block and therefore can cause a logical fault. However, from the perspective of the Steane code, only a single-

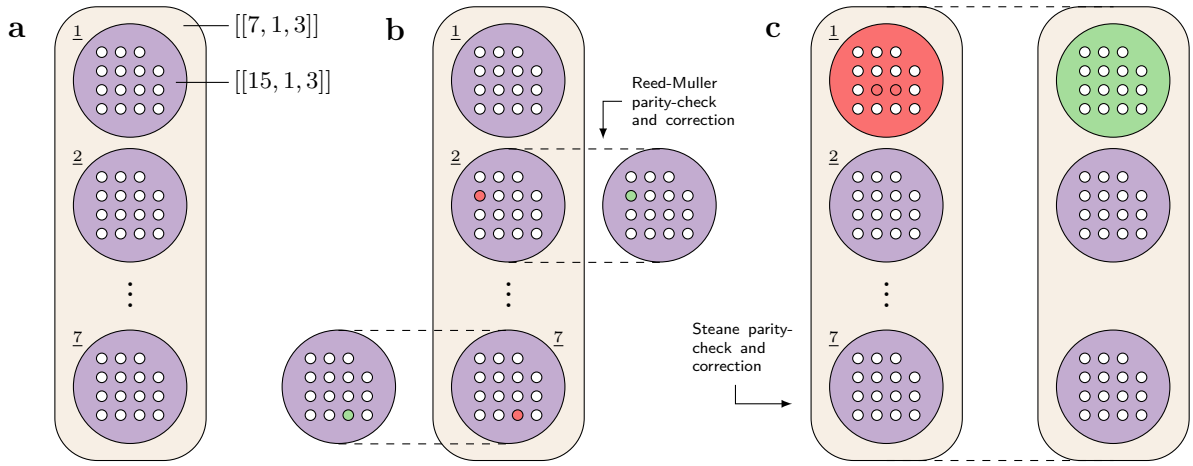


Figure 8: **a)** Concatenation of the $[[7,1,3]]$ Steane code and $[[15,1,3]]$ Reed-Muller code. Each underlying qubit of the Steane code is a Reed-Muller logical qubit using 15 physical qubits. **b)** Possible erroneous state after the application of the T_L for the concatenated code. Due to transversality, single-qubit errors propagate to different logical Reed-Muller qubits. Error correction on these logical Reed-Muller qubits corrects the overall state. **c)** Possible erroneous state after the application of a logical Clifford gate for the concatenated code. Due to the non-transversality of the underlying Reed-Muller logical gate, single-qubit errors propagate to multiple ones resulting in a logical Reed-Muller fault. Error correction on the Steane code can detect and correct one of such errors. Figure inspired by [6].

qubit error occurs and is therefore detectable and correctable (Figure 8c).

The final property we require is that the implementation of the six stabilizers of the Steane code is globally transversal. Otherwise, errors occurring during error correction can propagate and destroy further logical computation. Fortunately, the Reed-Muller code allows for transversal X_L and Z_L gates. We conclude that universal quantum computing is achievable using clever concatenation schemes. The obvious downside is that one logical qubit requires $7 \cdot 15$ physical ones for the correction of a single-qubit error.

9 Magic State Distillation

Another approach to achieving universal quantum computing was proposed by Bravyi and Kitaev in 2005 [4]. Their idea is to shift the problem from the implementation of a transversal non-Clifford gate (for our current discussion the T gate) to the purification, also *distillation*, of *magic states*.

Equation 32 defines one particular magic state.

$$|A^{\pi/4}\rangle = \frac{1}{\sqrt{2}}(|0\rangle + e^{i\frac{\pi}{4}}|1\rangle) \tag{32}$$

We now illustrate how to apply the following gate by "consuming" the $|A^{\pi/4}\rangle$ state.

$$\begin{pmatrix} 1 & 0 \\ 0 & e^{i\pi/4} \end{pmatrix} \tag{33}$$

The careful reader might have noticed that this is the T gate. Let $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$ be the state on which we want to apply the T gate. Then, the initial state $|\psi\rangle \otimes |A^{\pi/4}\rangle$ is

$$\alpha|00\rangle + \alpha e^{i\frac{\pi}{4}}|01\rangle + \beta|10\rangle + \beta e^{i\frac{\pi}{4}}|11\rangle \tag{34}$$

Now, apply a $CNOT$ gate where the control is the first qubit, i.e. the $|\psi\rangle$ state. The resulting state is

$$\alpha|00\rangle + \alpha e^{i\frac{\pi}{4}}|01\rangle + \beta|11\rangle + \beta e^{i\frac{\pi}{4}}|10\rangle \tag{35}$$

We can simplify this term by re-grouping the first and second qubits and multiplying the second term with a global phase.

$$\begin{aligned}(\alpha|0\rangle + \beta e^{i\frac{\pi}{4}}|1\rangle) \otimes |0\rangle \\ +e^{i\frac{\pi}{4}}(\alpha|0\rangle + \beta e^{-i\frac{\pi}{4}}|1\rangle) \otimes |1\rangle\end{aligned} \tag{36}$$

Thus by measuring the second qubit, we end up with either the desired state $T|\psi\rangle$ or have to apply a corrective $\frac{\pi}{2}$ rotation, i.e. a S gate. Figure 9 depicts the complete procedure as a circuit.

We are left to address the question of how to create such magic states. First, note that $|A^{\pi/4}\rangle = TH|0\rangle$. Hence, if we protect the state $|\psi\rangle$ by encoding it in the Steane code the problem of the

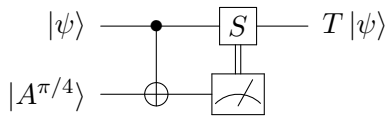


Figure 9: Circuit consuming a magic state $|A^{\pi/4}\rangle$ to apply a T gate on $|\psi\rangle$.

non-transversal logical T gate arises again. It is worth mentioning that the circuit in Figure 9 consists only of Clifford gates which we can perform fault-tolerantly thanks to the Steane code. As in the previous section, the transversal logical T gate of the $[[15, 1, 3]]$ Reed-Muller code is useful. Here, we will use it to distill a single logical $|A^{\pi/4}\rangle$ in the Steane code.

1. Encode 15 logical qubits of the Steane code in the Reed-Muller code as $|+_L\rangle$
2. Apply the transversal T gate of the Reed-Muller code
3. Perform error detection (and correction)
4. Decode the Reed-Muller code to yield a single $|A_L^{\pi/4}\rangle$ in the Steane code

As code-switching, this approach requires 105 physical qubits to distill one logical magic state for the application of one logical T gate. An in-depth analysis of the time-resource-cost of magic state distillation is beyond the scope of this paper. Thus, we refer the reader to Ref. [19].

Lastly, it is worth mentioning that recent work demonstrated the encoding of magic states using a four-qubit code on an actual quantum chip [15]. This result shows that quantum hardware slowly but surely catches up with the theory.

10 Summary

In this paper, we began by introducing the challenges that quantum error correction faces. We then showed how each challenge can be overcome and reviewed the theoretical foundations and key ideas of quantum error correction along the way. After that, we reviewed some of the fundamentals of fault-tolerant quantum computing. We especially emphasized on the non-transversal T gate of the Steane code and possible solutions for that problem - namely code-switching and magic state distillation.

This is by no means a complete introduction to quantum error correction and fault-tolerant quantum computation (and was never meant to be). Nevertheless, after finishing this paper the reader should have a basic and intuitive understanding of its concepts.

A possible route for the curious reader is to study the referenced source material, which should now be relatively easier, or continue with topics such as topological quantum computing. For an introduction of the latter see Ref. [11].

References

[1] Azure Quantum | Azure Quantum Roadmap. URL https://quantum.microsoft.com/en-us/our-story/quantum-roadmap.

[2] Google Quantum AI. Our quantum computing journey, May 2024. URL https://dreamcoat-guggenheim.uc.r.appspot.com/learn/map/.

[3] Utkarsh Azad. Efficient Simulation of Clifford Circuits. *PennyLane Demos*, April 2024. URL https://pennylane.ai/qml/demos/tutorial_clifford_circuit_simulations/. Publisher: Xanadu.

[4] Sergei Bravyi and Alexei Kitaev. Universal Quantum Computation with ideal Clifford gates and noisy ancillas. *Physical Review A*, 71(2):022316, February 2005. ISSN 1050-2947, 1094-1622. DOI: 10.1103/PhysRevA.71.022316. URL http://arxiv.org/abs/quant-ph/0403025. arXiv:quant-ph/0403025.

[5] A. R. Calderbank and Peter W. Shor. Good Quantum Error-Correcting Codes Exist. *Physical Review A*, 54(2):1098–1105, August 1996. ISSN 1050-2947, 1094-1622. DOI: 10.1103/PhysRevA.54.1098. URL http://arxiv.org/abs/quant-ph/9512032. arXiv:quant-ph/9512032.

[6] Earl T. Campbell, Barbara M. Terhal, and Christophe Vuillot. Roads towards fault-tolerant universal quantum computation. *Nature*, 549(7671):172–179, September 2017. ISSN 0028-0836, 1476-4687. DOI: 10.1038/nature23460. URL https://www.nature.com/articles/nature23460.

[7] Rui Chao and Ben W. Reichardt. Flag fault-tolerant error correction for any stabilizer code. *PRX Quantum*, 1(1), September 2020. ISSN 2691-3399. DOI: 10.1103/prxquantum.1.010302. URL http://dx.doi.org/10.1103/PRXQuantum.1.010302.

[8] Michael E. Cuffaro. On the Significance of the Gottesman-Knill Theorem. *The British Journal for the Philosophy of Science*, 68 (1):91–121, March 2017. ISSN 0007-0882, 1464-3537. DOI: 10.1093/bjps/axv016. URL http://arxiv.org/abs/1310.0938. arXiv:1310.0938 [physics, physics:quant-ph].

[9] Bryan Eastin and Emanuel Knill. Restrictions on Transversal Encoded Quantum Gate Sets. *Physical Review Letters*, 102(11):110502, March 2009. ISSN 0031-9007, 1079-7114. DOI: 10.1103/PhysRevLett.102.110502. URL http://arxiv.org/abs/0811.4262. arXiv:0811.4262 [quant-ph].

[10] Richard P Feynman. Simulating physics with computers. *International journal of theoretical physics*, 21(6/7):467–488, 1982.

[11] Keisuke Fujii. Quantum Computation with Topological Codes: from qubit to topological fault-tolerance, April 2015. URL http://arxiv.org/abs/1504.01444. arXiv:1504.01444 [quant-ph].

[12] Daniel Gottesman. Stabilizer codes and quantum error correction, 1997.

[13] Daniel Gottesman. The Heisenberg Representation of Quantum Computers, July 1998. URL http://arxiv.org/abs/quant-ph/9807006. arXiv:quant-ph/9807006.

[14] Daniel Gottesman. An Introduction to Quantum Error Correction and Fault-Tolerant Quantum Computation, April 2009. URL http://arxiv.org/abs/0904.2557. arXiv:0904.2557 [quant-ph].

[15] Riddhi S. Gupta, Neereja Sundaresan, Thomas Alexander, Christopher J. Wood, Seth T. Merkel, Michael B. Healy, Marius Hillenbrand, Tomas Jochym-O'Connor, James R. Wootton, Theodore J. Yoder, Andrew W. Cross, Maika Takita, and Benjamin J. Brown. Encoding a magic state with beyond break-even fidelity. *Nature*, 625 (7994):259–263, January 2024. ISSN 0028-0836, 1476-4687. DOI: 10.1038/s41586-023-06846-3. URL https://www.nature.com/articles/s41586-023-06846-3.

[16] IBM. IBM Quantum Roadmap, April 2024. URL https://www.ibm.com/roadmaps/quantum/www.ibm.com/roadmaps/quantum.

[17] Tomas Jochym-O'Connor and Raymond Laflamme. Using concatenated quantum codes for universal fault-tolerant quantum gates. *Physical Review Letters*, 112(1):010505, January 2014. ISSN 0031-9007, 1079-7114. DOI: 10.1103/PhysRevLett.112.010505. URL http://arxiv.org/abs/1309.3310. arXiv:1309.3310 [quant-ph].

[18] Cody Jones. Novel constructions for the fault-tolerant Toffoli gate. *Physical Review A*, 87(2):022328, February 2013. ISSN 1050-2947, 1094-1622. DOI: 10.1103/PhysRevA.87.022328. URL http://arxiv.org/abs/1212.5069. arXiv:1212.5069 [quant-ph].

[19] Daniel Litinski. Magic State Distillation: Not as Costly as You Think. *Quantum*, 3:205, December 2019. ISSN 2521-327X. DOI: 10.22331/q-2019-12-02-205. URL http://arxiv.org/abs/1905.06903. arXiv:1905.06903 [quant-ph].

[20] Nhung H. Nguyen, Muyuan Li, Alaina M. Green, Cinthia Huerta Alderete, Yingyue Zhu, Daiwei Zhu, Kenneth R. Brown, and Norbert M. Linke. Demonstration of Shor encoding on a trapped-ion quantum computer. *Physical Review Applied*, 16(2):024057, August 2021. ISSN 2331-7019. DOI: 10.1103/PhysRevApplied.16.024057. URL http://arxiv.org/abs/2104.01205. arXiv:2104.01205 [quant-ph].

[21] Michael A. Nielsen and Isaac L. Chuang. *Quantum Computation and Quantum Information*. Cambridge University Press, 2000.

[22] John Preskill. Quantum computing 40 years later, February 2023. URL http://arxiv.org/abs/2106.10522. arXiv:2106.10522 [quant-ph].

[23] Peter W. Shor. Scheme for reducing decoherence in quantum computer memory. *Physical Review A*, 52(4):R2493–R2496, October 1995. ISSN 1050-2947, 1094-1622. DOI: 10.1103/Phys-

RevA.52.R2493. URL https://link.aps.org/doi/10.1103/PhysRevA.52.R2493.

[24] Peter W. Shor. Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. *SIAM Journal on Computing*, 26 (5):1484–1509, October 1997. ISSN 1095-7111. DOI: 10.1137/s0097539795293172. URL http://dx.doi.org/10.1137/S0097539795293172.