

Quantum error correction for beginners

This content has been downloaded from IOPscience. Please scroll down to see the full text.

2013 Rep. Prog. Phys. 76 076001

(<http://iopscience.iop.org/0034-4885/76/7/076001>)

View [the table of contents for this issue](#), or go to the [journal homepage](#) for more

Download details:

IP Address: 128.104.46.206

This content was downloaded on 12/03/2014 at 19:26

Please note that [terms and conditions apply](#).

Quantum error correction for beginners

Simon J Devitt¹, William J Munro² and Kae Nemoto¹

¹ National Institute of Informatics, 2-1-2 Hitotsubashi Chiyoda-ku Tokyo, 101-8340, Japan

² NTT Basic Research Laboratories, NTT Corporation 3-1 Morinosato-Wakamiya, Atsugi Kanagawa, 243-0198, Japan

E-mail: devitt@nii.ac.jp

Received 14 November 2011, in final form 27 December 2012

Published 20 June 2013

Online at stacks.iop.org/RoPP/76/076001

Abstract

Quantum error correction (QEC) and fault-tolerant quantum computation represent one of the most vital theoretical aspects of quantum information processing. It was well known from the early developments of this exciting field that the fragility of coherent quantum systems would be a catastrophic obstacle to the development of large-scale quantum computers. The introduction of quantum error correction in 1995 showed that active techniques could be employed to mitigate this fatal problem. However, quantum error correction and fault-tolerant computation is now a much larger field and many new codes, techniques, and methodologies have been developed to implement error correction for large-scale quantum algorithms. In response, we have attempted to summarize the basic aspects of quantum error correction and fault-tolerance, not as a detailed guide, but rather as a basic introduction. The development in this area has been so pronounced that many in the field of quantum information, specifically researchers who are new to quantum information or people focused on the many other important issues in quantum computation, have found it difficult to keep up with the general formalisms and methodologies employed in this area. Rather than introducing these concepts from a rigorous mathematical and computer science framework, we instead examine error correction and fault-tolerance largely through detailed examples, which are more relevant to experimentalists today and in the near future.

(Some figures may appear in colour only in the online journal)

Contents

1. Introduction	2	8. Quantum error correction with stabilizer codes	11
2. Preliminaries	3	8.1. State preparation	12
2.1. Basic structure of the qubit	3	8.2. Error correction	13
2.2. Some general requirements of quantum error correction	3	9. Digitization of quantum errors	14
3. Quantum errors: cause and effect	4	9.1. Systematic gate errors	14
3.1. Coherent quantum errors: gates which are incorrectly applied	4	9.2. Environmental decoherence	15
3.2. Environmental decoherence	4	9.3. More general mappings	16
3.3. Simple models of loss, leakage, measurement and initialization	4	10. Fault-tolerant quantum error correction and the threshold theorem	16
4. The 3-qubit code: a good starting point for quantum error correction	7	10.1. Error propagation	17
5. The 9-qubit code: the first full quantum code	9	10.2. Concatenation	17
6. Quantum error detection	10	10.3. Fault-tolerance	18
7. Stabiliser formalism	11	10.4. Threshold theorem	18
		11. Fault-tolerant operations on encoded data	19
		11.1. Single qubit operations	20
		11.2. Two-qubit gate	20

12. Fault-tolerant circuit design for logical state preparation	21	14.1. Subsystem codes: Bacon–Shor codes	25
13. Loss protection	22	14.2. Topological codes	27
14. Some modern developments in quantum error correction	25	15. Conclusions and future outlook	30
		Acknowledgments	31
		References	31

1. Introduction

The micro-computer revolution of the late 20th Century has arguably had a greater impact on the world than any other technological revolution in history. The advent of transistors, integrated circuits, and the modern microprocessor has spawned literally hundreds of devices, from pocket calculators to the iPod, all now integrated through an extensive worldwide communications system. However, as we enter the 21st Century, ever increasing computational power is driving us quickly to the realm where quantum physics dominates. The component size of individual transistors on modern microprocessors is becoming so small that quantum effects will soon begin to dominate. Unfortunately, quantum mechanical behaviour will tend to result in unpredictable and unwanted operation in classical microprocessor designs. We therefore have two choices: keep trying to suppress quantum effects in classically fabricated electronics, or move to the field of quantum information processing (QIP) where we exploit them. This leads to a paradigm shift in the way we view and process information and has commanded considerable interest from physicists, engineers, computer scientists and mathematicians. The counter-intuitive and strange rules of quantum physics offer enormous possibilities for information processing and the development of a large-scale quantum computer is the holy grail for many researchers worldwide.

While the advent of Shor’s algorithm [Sho97] certainly spawned great interest in QIP, demonstrating that quantum algorithms could be far more efficient than those used in classical computing, there was a great deal of debate surrounding the practicality of building a large scale, controllable, quantum system. It was well known even before the introduction of quantum information that coherent quantum states were extremely fragile and many believed that to maintain large, multi-qubit, coherent quantum states for a long enough time to complete *any* quantum algorithm was unrealistic [Unr95]. Additionally, classical error correction techniques are intrinsically based on a digital framework, unlike quantum computing where the *readout* of qubits is digital but actual manipulations are analogue. Therefore, how can we adapt the vast knowledge gained from classical coding theory to the quantum regime?

Starting in 1995, several papers appeared, in rapid succession, proposing codes which were appropriate to perform error correction on quantum data [Sho95, Ste96a, CS96, LMPZ96, BSW96]. This was a key theoretical development needed to convince the general community that quantum computation was indeed a possibility. Since this initial introduction, the progress in this field has been extensive.

Initial research on error correction focused heavily on developing quantum codes [Ste96c, Ste96b, Got96, PVK97,

KL97, Kni96], introducing a more rigorous theoretical framework for the structure, properties and operation of QEC [BSW96, KL00, CRSS98, Got97, CG97, KLV00]. Additionally, the introduction of concepts such as fault-tolerant quantum computation [Sho96, DS96, Got98, KLZ98, Got00] led directly to the threshold theorem for concatenated QEC [KLZ96, ABO97]. In more recent years, QEC protocols have been developed for various systems, such as continuous variables [LS98, Bra98, vL08, ATK+09], ion-traps and other systems containing motional degrees of freedom [ST00], adiabatic computation [JFS06] and globally controlled quantum computers [BBK03]. Work also continues on not only developing better (and in some ways, more technologically useful) protocols such as subsystem codes [KLP05, KLPL06, Bac06] and topological codes [Kit97, DKLP02, BMD06, BMD07b, BMD07a, RHG07, FSG09]. Advanced techniques to analyse fault-tolerant thresholds [AGP08, PR12, BMD08, BAO+12] have also been developed to implement error correction in a fault-tolerant manner with concatenated codes [Ste97a, Ste02, DA07, Kni05] and topological codes [BMD07b, KBAMD10, Bom11].

Along with QEC, other methods of protecting quantum information were also developed. These other techniques would technically be placed in a separate category of error suppression rather than error correction. The distinction between error suppression and error correction is also known as passive and active QEC, the latter terminology being much more prevalent in the theoretical community. The most well known techniques of error suppression are protocols such as decoherence free subspaces (DFS) [PSE96, DG97, LCW98, DG98b, ZR97b, ZR97a, DG98a, LW03]. DFS protocols are passive, encoding information in a subspace which is invariant under the dynamics that generate errors. Active methods for error suppression include applying repeated rotations to qubits such that these rotations, combined with the natural dynamics that induce errors, partially cancel. These methods were initially referred to as Bang-Bang control [VL98, VT99, Zan99] and developed primarily for NMR systems. In recent years these techniques have been further generalized and optimized for use in essentially all QIP systems [VKL99, FLP04, VK03, VK05, KL05, Uhr07]. These more advanced techniques are now routinely categorized as dynamical decoupling protocols. As with QEC, this field is vast, incorporating well established ideas from quantum control to create specially designed control sequences to suppress errors before more resource intensive encoding is applied.

This review deals exclusively with the concepts of QEC and fault-tolerant quantum computation. Many papers have reviewed error correction and fault-tolerance [Got97, NC00, Got02, KLA+02, Ste01, Got09], however to cater for a large

audience, we attempt to describe QEC and fault-tolerance in a more basic manner, largely through examples. Instead of providing a more rigorous review of error correction, we instead focus on more practical issues involved when working with these ideas. For those who have recently begun investigating QIP or those who are focused on other important theoretical and/or experimental aspects related to quantum computing, searching through this enormous collection of work is daunting especially if a basic working knowledge of QEC is all that is required. This review of the basic aspects of QEC and fault-tolerance is designed to allow those with little knowledge of the field to quickly become accustomed to the various techniques and tricks that are commonly used. The examples in this review are selected due to their relative simplicity. For individuals seriously interested in examining new ideas for QEC and fault-tolerant computation we strongly advise that the references cited throughout are consulted.

We begin the discussion in section 2 where we describe some preliminary concepts on the required properties of any quantum error correcting protocol. In section 3 we review some basic noise models in the context of how they influence quantum algorithms. Section 4 introduces quantum error correction through the traditional example of the 3-qubit code, illustrating the circuits used for encoding and correction and why the principle of redundant encoding suppresses the failure of encoded qubits. Sections 5 and 6 introduce the first examples of complete quantum codes, Shor's 9-qubit code and a 4-qubit error detection code. Sections 7 and 8 introduce the stabilizer formalism, demonstrating how QEC circuits are synthesized once the structure of the code is known. In section 9 we briefly return to noise models and relate the abstract analysis of QEC, where errors are assumed to be discrete and probabilistic, to some of the physical mechanisms which can cause errors. Sections 10 and 11 introduce the concept of fault-tolerant error correction, the threshold theorem and how logical gate operations can be applied directly to encoded data. We then move onto circuit synthesis in section 12, presenting a fault-tolerant circuit design for state preparation using the 7-qubit Steane code as a representative example. Finally, in section 14, we review specific codes for qubit loss and examine two more modern techniques for error correction. We briefly examine quantum subsystem codes [KLP05, KLPL06, Bac06] and topological surface codes [DKLP02, FSG09] due to both their theoretical elegance and their increasing relevance in quantum architecture designs [IFI⁺02, DFS⁺09, SJ09, MLFY10].

2. Preliminaries

Before discussing quantum errors and how they can be corrected, we first introduce the basics of qubits and quantum gates. We assume a basic working knowledge with quantum information [EJ96, NC00] and this brief discussion is used simply to define our notation for the remainder of this review.

2.1. Basic structure of the qubit

The fundamental unit of quantum information is the qubit. Unlike the classical bit, the qubit can exist in coherent

superpositions of its two states, denoted as $|0\rangle$ and $|1\rangle$. These basis states can be, for example, photonic polarization, atomic spin states, electronic states of an ion or charge states of superconducting systems. An arbitrary state of an individual qubit, $|\phi\rangle$, can be expressed as

$$|\phi\rangle = \alpha|0\rangle + \beta|1\rangle, \quad (1)$$

where $|0\rangle$ and $|1\rangle$ are the two orthonormal basis states of the qubit and $|\alpha|^2 + |\beta|^2 = 1$. Quantum gate operations are represented by unitary operations acting on the Hilbert space of a collection of qubits. Unlike classical information processing, conservation of probability for quantum states requires that all operations be reversible and hence unitary.

When describing a quantum gate on an individual qubit, any dynamical operation, G , is a member of the unitary group $U(2)$, which consists of all 2×2 matrices where $G^\dagger = G^{-1}$. Up to a global (and unphysical) phase factor, any single qubit operation can be expressed as a linear combination of the generators of $SU(2)$ as,

$$G = c_I \sigma_I + c_x \sigma_x + c_y \sigma_y + c_z \sigma_z, \quad (2)$$

where

$$\sigma_x = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}, \quad \sigma_y = \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix}, \quad \sigma_z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}, \quad (3)$$

are the Pauli matrices, σ_I is the 2×2 identity matrix, c_I is a real number, and (c_x, c_y, c_z) are complex numbers satisfying,

$$\begin{aligned} |c_I|^2 + |c_x|^2 + |c_y|^2 + |c_z|^2 &= 1, \\ \Re(c_I c_x^*) + \Im(c_y c_z^*) &= 0, \\ \Re(c_I c_y^*) + \Im(c_x c_z^*) &= 0, \\ \Re(c_I c_z^*) + \Im(c_x c_y^*) &= 0. \end{aligned} \quad (4)$$

Here \Re and \Im are the real and imaginary components, respectively.

2.2. Some general requirements of quantum error correction

Although the field of QEC is largely based on classical coding theory, there are several issues that need to be considered when transferring classical error correction techniques to the quantum regime.

First, coding based on data-copying, which is extensively used in classical error correction, cannot be used due to the no-cloning theorem of quantum mechanics [WZ82]. This result implies that there exists no transformation resulting in the following mapping,

$$U(|\phi\rangle \otimes |\psi\rangle) = |\phi\rangle \otimes |\phi\rangle \quad \forall |\phi\rangle, \quad (5)$$

i.e. it is impossible to perfectly copy an unknown quantum state. This means that quantum data cannot be protected from errors by simply making multiple copies. Secondly, direct measurement cannot be used to effectively protect against errors, since this will act to destroy any quantum superposition that is being used for computation. Error correction protocols

must therefore be employed, which can detect and correct errors without determining *any* information regarding the qubit state. Unlike classical information, qubits are susceptible to both traditional bit errors $|0\rangle \leftrightarrow |1\rangle$, and also phase errors $|0\rangle \leftrightarrow |0\rangle$, $|1\rangle \leftrightarrow -|1\rangle$. Hence any error correction procedure needs to be able to simultaneously correct for both. Finally, errors in quantum information are intrinsically continuous (i.e. qubits do not experience full bit or phase flips but rather an angular shift of the qubit state by any angle). This issue will be discussed in more depth in section 9.

At its most basic level, QEC utilizes the idea of redundant encoding. This is where the total size of the Hilbert space is expanded beyond what is needed to store a single qubit of information. This way, errors on individual qubits are mapped to a large set of mutually orthogonal subspaces, the size of which is determined by the number of qubits utilized in the code. Finally, the error correction protocol cannot allow us to gain information regarding the coefficients, α and β of the encoded state, as doing so would collapse the system.

3. Quantum errors: cause and effect

Before we begin discussing the details of QEC, we first examine some of the common sources of errors in QIP and contextualize what they imply for computation. We will consider several important sources of errors and how they influence two trivial, single qubit quantum algorithms.

The first algorithm is a computation consisting of a single qubit, initialized in the $|0\rangle$ state undergoing N identity operations. If this algorithm is performed perfectly the final state is,

$$|\psi\rangle_{\text{final}} = \prod_i^N I_i |0\rangle = |0\rangle, \quad (6)$$

where $I \equiv \sigma_I$ is the identity gate. Measurement of the qubit in the $|0\rangle$, $|1\rangle$ basis will consequently yield the result 0 with a probability of unity.

The second illustration is an algorithm of three gates,

$$\begin{aligned} |\psi\rangle_{\text{final}} &= H I H |0\rangle = H I \frac{1}{\sqrt{2}} (|0\rangle + |1\rangle) \\ &= H \frac{1}{\sqrt{2}} (|0\rangle + |1\rangle) \\ &= |0\rangle. \end{aligned} \quad (7)$$

This algorithm will *ideally* yield a result $|0\rangle$ with probability of 1 when measured in the $|0\rangle$, $|1\rangle$ basis. This algorithm implements two $H \equiv$ Hadamard operations separated by a wait stage, represented by the Identity gate.

We examine, independently, several common sources of error from the effect they have on these two quantum algorithms. Hopefully, this introductory section will show that while quantum errors are complicated physical effects, in QIP the relevant measure is the theoretical success probability of a given quantum algorithm.

Errors that exist on any quantum system are highly dependent on the specific physical mechanisms that govern the system. The situations that follow are only basic illustrative

examples which can often arise in discussions related to errors and QEC. It is important to stress that the types of errors and the way in which they are analysed need to be considered in the context of the physical system under consideration.

3.1. Coherent quantum errors: gates which are incorrectly applied

The first possible source of error is coherent, systematic control errors. This type of error is typically associated with incorrect knowledge of the system dynamics. Namely, otherwise perfect control assuming a system governed by a Hamiltonian H' when the actual system dynamics is governed by H . As qubit systems must be characterized prior to being used [PCZ97, CSG+05], finite experimental resolution with these processes will lead to coherent control errors. As these errors are coherent inaccuracies applied to the qubit, they can be mitigated using coherent techniques such as composite pulse sequences [LF79, Lev86, BHC04, Jon09]. Other control errors (e.g. arising from fluctuations in control fields) are not coherent processes and are dealt with in a similar way to environmental coupling described further on.

As this source of error is coherent and systematic it causes one to apply the same, undesired gate operation and can be analysed without moving to the density matrix formalism. With respect to the first of our algorithms, we are able to model this in several different ways. To keep things simple, we assume that incorrect characterization of the control dynamics leads to a gate which is not σ_I , but instead introduces a small rotation around the X -axis of the Bloch sphere. This results in the state,

$$|\psi\rangle_{\text{final}} = \prod_i^N e^{i\epsilon\sigma_x} |0\rangle = \cos(N\epsilon)|0\rangle + i \sin(N\epsilon)|1\rangle. \quad (8)$$

We now measure the system in the $|0\rangle$, $|1\rangle$ basis. Due to these errors, the probability of measuring the system in the $|0\rangle$ or $|1\rangle$ state is,

$$\begin{aligned} P(|0\rangle) &= \cos^2(N\epsilon) \approx 1 - (N\epsilon)^2, \\ P(|1\rangle) &= \sin^2(N\epsilon) \approx (N\epsilon)^2. \end{aligned} \quad (9)$$

Hence, the probability of error in this trivial quantum algorithm is given by $p_{\text{error}} \approx (N\epsilon)^2$, which will be small given that $N\epsilon \ll 1$. The systematic error in this system is quadratic in both the small systematic over rotation and the total number of applied identity operations. This is expected as this rotational error always acts in the same direction every time it is applied.

3.2. Environmental decoherence

Environmental decoherence is another important source of errors in quantum systems. We will take a simple decoherence model and examine how our second algorithm behaves. Later in section 9 we will illustrate a more complicated decoherence model that arises from standard mechanisms.

Consider a very simple environment, which is another two level quantum system. This environment has two basis states, $|e_0\rangle$ and $|e_1\rangle$, that satisfy the completeness relations,

$$\langle e_i | e_j \rangle = \delta_{ij}, \quad |e_0\rangle\langle e_0| + |e_1\rangle\langle e_1| = I. \quad (10)$$

We will also assume that the environment couples to the qubit in a specific way. When the qubit is in the $|1\rangle$ state, the coupling flips the environmental state, while if the qubit is in the $|0\rangle$ state nothing happens to the environment. Finally, this model assumes the system/environment interaction only occurs during the wait stage of the algorithm (while the Identity gate is being applied). As with the first algorithm we should measure the state $|0\rangle$ with probability one. The reason for utilizing the second of our algorithms in this example is that this specific decoherence model acts to reduce coherence between the $|0\rangle$ and $|1\rangle$ states. Therefore, we require a coherent superposition to observe any effect from the environmental coupling. We assume the environment is initialized in the state, $|E\rangle = |e_0\rangle$, and then coupled to the system,

$$HIH|0\rangle|E\rangle = \frac{1}{2}(|0\rangle + |1\rangle)|e_0\rangle + \frac{1}{2}(|0\rangle - |1\rangle)|e_1\rangle. \quad (11)$$

As we are considering environmental decoherence, pure states will be transformed into classical mixtures, hence we now move into the density matrix representation for the state $HIH|0\rangle|E\rangle$,

$$\begin{aligned} \rho_f = & \frac{1}{4}(|0\rangle\langle 0| + |0\rangle\langle 1| + |1\rangle\langle 0| + |1\rangle\langle 1|)|e_0\rangle\langle e_0| \\ & + \frac{1}{4}(|0\rangle\langle 0| - |0\rangle\langle 1| - |1\rangle\langle 0| + |1\rangle\langle 1|)|e_1\rangle\langle e_1| \\ & + \frac{1}{4}(|0\rangle\langle 0| - |0\rangle\langle 1| + |1\rangle\langle 0| - |1\rangle\langle 1|)|e_0\rangle\langle e_1| \\ & + \frac{1}{4}(|0\rangle\langle 0| + |0\rangle\langle 1| - |1\rangle\langle 0| - |1\rangle\langle 1|)|e_1\rangle\langle e_0|. \end{aligned} \quad (12)$$

As we have no access to the environmental degrees of freedom, we trace over this part of the system, giving,

$$\begin{aligned} \text{Tr}_E(\rho_f) = & \frac{1}{4}(|0\rangle\langle 0| + |0\rangle\langle 1| + |1\rangle\langle 0| + |1\rangle\langle 1|) \\ & + \frac{1}{4}(|0\rangle\langle 0| - |0\rangle\langle 1| - |1\rangle\langle 0| + |1\rangle\langle 1|) \\ = & \frac{1}{2}(|0\rangle\langle 0| + |1\rangle\langle 1|). \end{aligned} \quad (13)$$

Measurement of the system will consequently return $|0\rangle$ 50% of the time and $|1\rangle$ 50% of the time. This final state is a complete mixture of the two qubit states and is consequently a classical system. The coupling to the environment removed all the coherences between the $|0\rangle$ and $|1\rangle$ states and the second Hadamard transform, intended to rotate $(|0\rangle + |1\rangle)/\sqrt{2} \rightarrow |0\rangle$, has no effect on the qubit state.

As we assumed that the system/environment coupling during the wait stage causes the environmental degree of freedom to ‘flip’ (when the qubit is in the $|1\rangle$ state) this decoherence model implicitly incorporates a temporal effect. The temporal interval of the identity gate in the above algorithm is long enough to enact the controlled-flip operation. If we assumed a controlled rotation that was not a full flip on the environment, the final mixture would not be 50/50. Instead there would be a residual coherence between the qubit states and an increased probability of our algorithm returning a $|0\rangle$. Section 9 revisits the decoherence model and illustrates how time-dependence is explicitly incorporated.

3.3. Simple models of loss, leakage, measurement and initialization

Other sources of error such as qubit initialization, measurement errors, qubit loss and qubit leakage can be modelled in

a very similar manner. As with coherent control errors and environmental decoherence, the specifics of other error channels can be modelled in a coherent or incoherent manner, depending on the physical mechanisms. The examples shown below are commonplace within QEC analysis and adequately apply to a large number of systems.

Measurement errors can be modelled in the same way as environmental decoherence, acting incoherently on the system. Measurement errors can be described in two slightly different ways. The first is to use the following positive operator value measures (POVMs),

$$\begin{aligned} F_0 &= (1 - p_M)|0\rangle\langle 0| + p_M|1\rangle\langle 1|, \\ F_1 &= (1 - p_M)|1\rangle\langle 1| + p_M|0\rangle\langle 0|, \end{aligned} \quad (14)$$

where p_M is the probability of measurement error. These are not measurement projectors as $F_0^2 \neq F_0$ and $F_1^2 \neq F_1$. The second method is to apply the following mapping to the qubit,

$$\rho \rightarrow \rho' = (1 - p_M)\rho + p_M X \rho X \quad (15)$$

followed by a perfect measurement in the $(|0\rangle, |1\rangle)$ basis. These two models give exactly the same probabilities for each outcome. Defining $A_0 = |0\rangle\langle 0|$ and $A_1 = |1\rangle\langle 1|$ as the measurement projectors onto the $(|0\rangle, |1\rangle)$ basis we find

$$\begin{aligned} \text{Tr}(F_0\rho) &= (1 - p_M)\text{Tr}(A_0\rho) + p_M\text{Tr}(A_1\rho), \\ \text{Tr}(F_1\rho) &= (1 - p_M)\text{Tr}(A_1\rho) + p_M\text{Tr}(A_0\rho), \end{aligned} \quad (16)$$

$$\begin{aligned} \text{Tr}(A_0\rho') &= (1 - p_M)\text{Tr}(A_0\rho) + p_M\text{Tr}(X A_0 X \rho) \\ &= (1 - p_M)\text{Tr}(A_0\rho) + p_M\text{Tr}(A_1\rho), \\ \text{Tr}(A_1\rho') &= (1 - p_M)\text{Tr}(A_1\rho) + p_M\text{Tr}(X A_1 X \rho) \\ &= (1 - p_M)\text{Tr}(A_1\rho) + p_M\text{Tr}(A_0\rho). \end{aligned} \quad (17)$$

Hence, either method will result in the same probabilities for a given value of p_M . The difference between these two models is the state the measured qubit is projected to.

When using the POVMs in equation (14), the collapsed state of the qubit is given by

$$\rho \rightarrow \frac{M_i \rho M_i^\dagger}{\text{Tr}(F_i \rho)} \quad i = 0, 1, \quad (18)$$

where

$$\begin{aligned} M_0 &= \sqrt{1 - p_M}|0\rangle\langle 0| + \sqrt{p_M}|1\rangle\langle 1|, \\ M_1 &= \sqrt{1 - p_M}|1\rangle\langle 1| + \sqrt{p_M}|0\rangle\langle 0|. \end{aligned} \quad (19)$$

Therefore the resulting state, after measurement, will *not* be initialized in a known state. If the second model is used, then the system is initialized to either $|0\rangle$ or $|1\rangle$ depending on which projector is used. To see this explicitly, consider the state,

$$\frac{1}{\sqrt{2}}(|0\rangle + |1\rangle). \quad (20)$$

Using both models, the probability of measuring $|0\rangle$ is $1/2$ (as the above state is invariant under a bit-flip error). If the POVM model is used (using the F_0 and M_0 operators), the state after measurement is,

$$\sqrt{(1 - p_M)}|0\rangle + \sqrt{p_M}|1\rangle. \quad (21)$$

The collapsed state, after measurement, is a superposition of $|0\rangle$ and $|1\rangle$, with amplitudes related to p_m . If we use the second model (using the projector A_0), the state after measurement is simply $|0\rangle$. As measurement is generally followed by either discarding the qubit or reinitializing it in a known state, both models can generally be used interchangeably with no adverse consequences.

Qubit loss is modelled in a slightly different manner. When a qubit is lost, it is effectively removed from the system. This channel can be modelled by simply tracing out the lost qubit, $\text{Tr}_i(\rho)$, where i is the index of the lost qubit. This reduces the dimensionality of the qubit space by a factor of two.

The loss of the physical object means that it cannot be directly measured or coupled to any other ancillary system. Even though this model of qubit loss (with regards to the information stored on the qubit) is equivalent to incoherent processes such as environmental coupling, correcting this type of error requires additional machinery on top of standard QEC protocols. Standard correction protocols can protect against the loss of information on a qubit, provided that the physical object still exists in the computer. Therefore, an initial non-demolition detection must be employed (which determines if the qubit is actually present without performing a projective measurement on the computational state) before standard correction can correct the error. Provided that these non-demolition protocols are employed, loss actually becomes a preferred channel in error correction. If a loss event is detected and the qubit is replaced, this heralds the qubit experiencing the error. This additional information can be used within error correction protocols to increase performance.

Initialization of the qubit can be modelled either using a coherent systematic or incoherent process. If an incoherent model is employed, initialization can be modelled in essentially the same way as imperfect measurement. If we have a probability p_I of initialization error (where p_I encapsulates the internal mechanisms that introduce errors) and we initialize the qubit in the $|0\rangle$ state, the initial state of the system is given by the mixture,

$$\rho_i = (1 - p_I)|0\rangle\langle 0| + p_I|1\rangle\langle 1|. \quad (22)$$

In contrast, we could consider an initialization model which is achieved via a coherent unitary operation where the target is the desired initial state. In this case, the initial state is pure, but contains a non-zero amplitude of the undesired target, for example,

$$|\psi\rangle = \alpha|0\rangle + \beta|1\rangle, \quad (23)$$

where $|\alpha|^2 + |\beta|^2 = 1$ and $|\beta|^2 \ll 1$. The interpretation of these two types of initialization models is identical to the coherent and incoherent models presented. Again, the effect of these types of errors relates to the probabilities of measuring the system in an erred state.

One final type of error that we can briefly mention is qubit leakage. Qubit leakage manifests due to the fact that most systems utilized for qubits do not consist of just two levels. For example, figure 1 (from [Ste97b]) illustrates the energy level structure for a $^{43}\text{Ca}^+$ ion utilized for ion trap

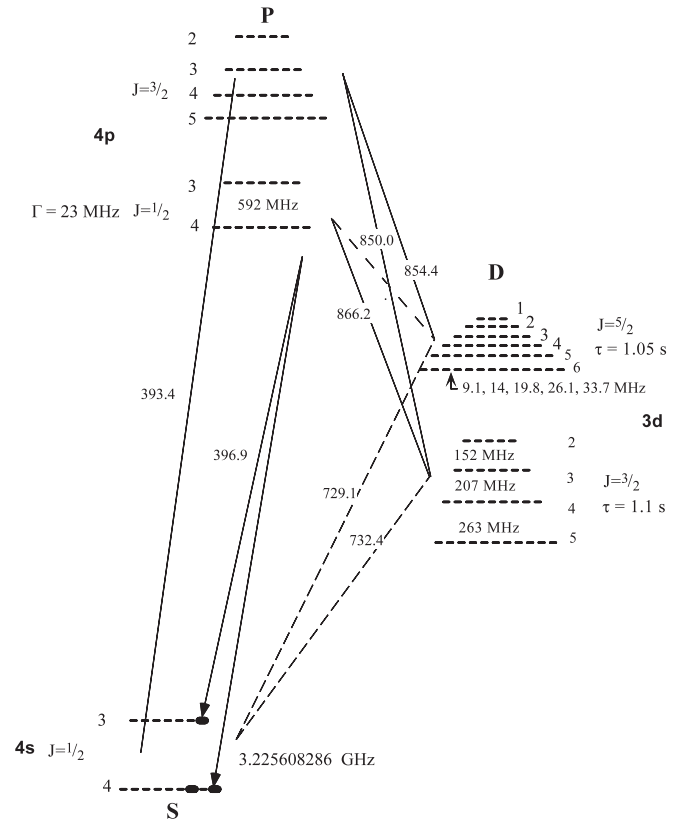


Figure 1. (From [Ste97b]) Energy level structure for the $^{43}\text{Ca}^+$ investigated by the Oxford ion-trapping group. The structure of this ion is clearly not a 2-level quantum system. Hence leakage into non-qubit states is an important factor to consider.

quantum computing. The qubits in this system are defined with two electronic states, however the system itself contains many more levels (including some which are used for qubit readout and initialization). As with systematic errors, leakage can occur when improper control is applied to such a system. In the case of ion-traps, qubit transitions are performed by focusing finely tuned lasers resonant on the relevant transitions. If the laser frequency fluctuates or additional levels are not sufficiently detuned from the qubit resonance, the following transformation could occur,

$$U|0\rangle = \alpha|0\rangle + \beta|1\rangle + \gamma|2\rangle, \quad (24)$$

where the state $|2\rangle$ is a third level, which is now populated due to improper control. The actual effect of this type of error can manifest in several different ways. As quantum circuits and algorithms are fundamentally designed assuming the computational array is of a 2-level system, the transformation shown in equation (24) (which in this case is operating over a 3-level space) will naturally induce unwanted dynamics. Another important implication of applying non-qubit operations is how these levels interact with the environment and hence how decoherence affects the system. For example, in the above case, the unwanted level, $|2\rangle$, may be extremely short lived leading to an emission of a photon and the system relaxing back to the ground state. For these reasons, leakage is one of the most problematic error channels to correct using QEC. In general, leakage

induced errors need to be corrected in a similar manner to loss, which can also be thought of as a leakage process into a vacuum state. Non-demolition techniques are first required to determine if the system is still confined to the qubit subspace [Pre98, GBP97, VWW05] after which standard correction protocols can be utilized. As with qubit loss, if these additional protocols are employed, leakage errors are heralded and the performance of the resulting error correction will increase.

Another well known method is to use a complicated pulse sequence which acts to re-focus an improperly confined operation back to the qubit subspace [WBL02, BLWZ05]. This can be advantageous as it does not require additional qubit resources.

Leakage effects arise from multiple sources, and the best method for correction is often heavily dependent on the error mechanisms. One example is when internal fluctuations in the system change otherwise perfect qubit dynamics. Unfortunately this leakage source generally cannot be predicted and/or characterized and so dedicated leakage protection will need to be employed. A second source is imprecise fabrication of an otherwise perfect qubit system. This source of leakage can, in principal, be engineered away, thereby eliminating the need for specialized leakage correction. In the context of mass manufacturing of qubit systems, leakage would ideally be quantified immediately after the fabrication of a device [DSO+07] to eliminate these systematic leakage errors. If a particular system was found to be improperly confined to the qubit subspace, it would simply be discarded. Employing characterization at this stage could potentially eliminate the need to implement a large amount of leakage protection in the computer, shortening gate times and ultimately reducing error rates in the computer.

In this section we have presented a very basic set of examples to explain some of the ideas of quantum errors and how they affect the success of a quantum algorithm. Section 9 will return to a more realistic set of error models. For those interested in a more complete treatment of quantum errors we encourage readers to refer to [Got97, NC00, Got02, KLA+02, Ste01, Got09].

4. The 3-qubit code: a good starting point for quantum error correction

The 3-qubit bit-flip code is traditionally used as a basic introduction to the concept of quantum error correction. It should be emphasized that the 3-qubit code *does not* represent a full quantum code. This is due to the fact that the code cannot simultaneously correct for both bit and phase flips (section 9). This code is a repetition code extended by Shor [Sho95] to construct the 9-qubit quantum code, demonstrating that QEC was possible.

The 3-qubit code encodes a single logical qubit into three physical qubits with the property that it can correct for a single, σ_x , bit-flip error. The two logical basis states $|0\rangle_L$ and $|1\rangle_L$ are defined as

$$|0\rangle_L = |000\rangle, \quad |1\rangle_L = |111\rangle, \quad (25)$$

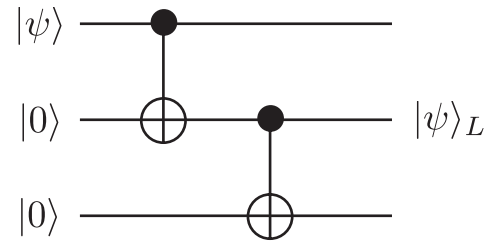


Figure 2. Quantum circuit to prepare the $|0\rangle_L$ state for the 3-qubit code where an arbitrary single qubit state, $|\psi\rangle$ is coupled to two freshly initialized ancilla qubits via CNOT gates to prepare $|\psi\rangle_L$.

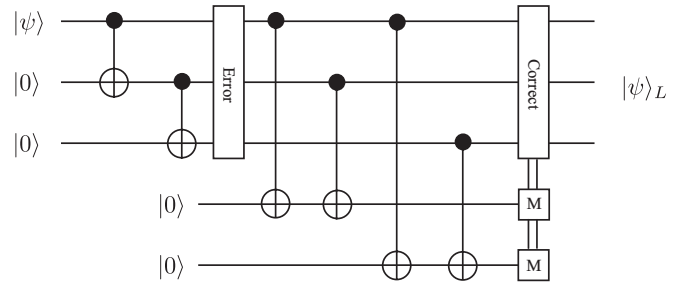


Figure 3. Circuit required to encode and correct for a single σ_x -error. We assume that after encoding a single bit-flip occurs on one of the three qubits (or no error occurs). Two initialized ancilla are then coupled to the data block which only checks the parity between qubits. These ancilla are then measured, with the measurement result indicating where (or if) an error has occurred, without directly measuring any of the data qubits. Using this *syndrome* information, the error can be corrected with a classically controlled σ_x gate.

such that an arbitrary single qubit state $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$ is mapped to,

$$\begin{aligned} \alpha|0\rangle + \beta|1\rangle &\rightarrow \alpha|0\rangle_L + \beta|1\rangle_L \\ &= \alpha|000\rangle + \beta|111\rangle \\ &= |\psi\rangle_L. \end{aligned} \quad (26)$$

Figure 2 illustrates the quantum circuit required to encode a single logical qubit via the initialization of two ancilla qubits and two CNOT gates. The reason that this code is able to correct for a single bit-flip error is the binary distance between the two codeword states. Notice that three individual bit flips are required to take $|0\rangle_L \leftrightarrow |1\rangle_L$, hence if we assume $|\psi\rangle = |0\rangle_L$, a single bit flip on any qubit leaves the final state closer to $|0\rangle_L$ than $|1\rangle_L$. The distance between two codeword states, d , defines the number of errors that can be corrected, t , as $t = \lfloor (d - 1)/2 \rfloor$. In this case, $d = 3$, hence $t = 1$.

How are we able to correct errors using this code without directly measuring or obtaining information about the logical state? Two additional ancilla qubits are introduced, which are used to extract *syndrome* information (information regarding possible errors) from the data block without discriminating the exact state of any qubit. The encoding and correction circuit is illustrated in figure 3. Correction proceeds by introducing two ancilla qubits and performing a sequence of CNOT gates, which checks the parity of the three qubit data block. For the sake of simplicity we assume that all gate operations are perfect and the only place where the qubits are susceptible to error is

Table 1. Final state of the five qubit system prior to the syndrome measurement for no error or a single X error on one of the qubits. The last two qubits represent the state of the ancilla. Note that each possible error will result in a unique measurement result (syndrome) of the ancilla qubits. This allows for a σ_x correction gate to be applied to the data block which is classically controlled from the syndrome result.

Error location	Final state, $ \text{data}\rangle \text{ancilla}\rangle$
No error	$\alpha 000\rangle 00\rangle + \beta 111\rangle 00\rangle$
Qubit 1	$\alpha 100\rangle 11\rangle + \beta 011\rangle 11\rangle$
Qubit 2	$\alpha 010\rangle 10\rangle + \beta 101\rangle 10\rangle$
Qubit 3	$\alpha 001\rangle 01\rangle + \beta 110\rangle 01\rangle$

Table 2. Ancilla measurements for a single σ_x error with the 3-qubit code. Each of the four possible results correspond to either no error or a bit flip on one of the three qubits.

Ancilla measurement	Collapsed state	Consequence
00	$\alpha 000\rangle + \beta 111\rangle$	No error
01	$\alpha 001\rangle + \beta 110\rangle$	σ_x on qubit 3
10	$\alpha 010\rangle + \beta 101\rangle$	σ_x on qubit 2
11	$\alpha 100\rangle + \beta 011\rangle$	σ_x on qubit 1

Table 3. Ambiguity of syndrome results when multiple errors occur.

Error location	Final state, $ \text{data}\rangle \text{ancilla}\rangle$	Assumed error
Qubit 1 and 2	$\alpha 110\rangle 01\rangle + \beta 001\rangle 01\rangle$	σ_x on qubit 3
Qubit 2 and 3	$\alpha 011\rangle 11\rangle + \beta 100\rangle 11\rangle$	σ_x on qubit 1
Qubit 1 and 3	$\alpha 101\rangle 10\rangle + \beta 010\rangle 10\rangle$	σ_x on qubit 2
Qubit 1, 2 and 3	$\alpha 111\rangle 00\rangle + \beta 000\rangle 00\rangle$	No error

the region between encoding and correction (as illustrated in figure 3). We will return to this issue in section 10 when we discuss fault-tolerance. We also assume that at most a single, complete bit-flip error occurs on one of the three data qubits. Table 1 summarizes the state of the whole system, for each possible error, just prior to measurement.

For each possible situation, either no error or a single bit-flip error, the ancilla qubits are flipped to a unique state based on the parity of the data block. These qubits are then measured to obtain the classical *syndrome* result. The result of the measurement will then dictate if a correction gate needs to be applied. This is illustrated in table 2.

Provided that only a single error has occurred, the data block is restored. At no point during correction do we gain any information regarding the coefficients α and β , hence superpositions of the computational state will remain intact during correction.

This code will only work if a maximum of one error occurs. The 3-qubit code is a $d = 3$ code, hence if $t > 1$, then the resulting state becomes ‘closer’ to the wrong logical state. This can be seen in table 3 when tracking multiple errors through the correction circuit. In each case, we assume that the total number of errors is $> d/2$, therefore our correction takes us to the wrong logical state. In every case where $t > 1$ our mis-correction induces a logical bit flip, causing the code to fail.

To be absolutely clear on how QEC acts to restore the system and protect against errors let us now consider a different

and slightly more physically realistic example. We will assume that the errors acting on the qubits are coherent rotations of the form $U = \exp(i\epsilon\sigma_x)$ on each qubit, with $\epsilon \ll 1$. We choose coherent rotations so that we can remain in the state vector representation. This is not a necessary requirement, however more general incoherent mappings would require us to move to density matrices.

We assume that each qubit experiences the same error, hence the error operator acting on the state is

$$\begin{aligned}
 |\psi\rangle_E &= E|\psi\rangle_L, \\
 E &= U^{\otimes 3} = (\cos(\epsilon)\sigma_I + i\sin(\epsilon)\sigma_x)^{\otimes 3} \\
 &= c_0\sigma_I\sigma_I\sigma_I \\
 &\quad + c_1(\sigma_x\sigma_I\sigma_I + \sigma_I\sigma_x\sigma_I + \sigma_I\sigma_I\sigma_x) \\
 &\quad + c_2(\sigma_x\sigma_x\sigma_I + \sigma_I\sigma_x\sigma_x + \sigma_x\sigma_I\sigma_x) \\
 &\quad + c_3\sigma_x\sigma_x\sigma_x,
 \end{aligned} \tag{27}$$

where

$$\begin{aligned}
 c_0 &= \cos^3(\epsilon), \\
 c_1 &= i\cos^2(\epsilon)\sin(\epsilon), \\
 c_2 &= -\cos(\epsilon)\sin^2(\epsilon), \\
 c_3 &= -i\sin^3(\epsilon).
 \end{aligned} \tag{28}$$

Now let us examine the transformation that occurs when we run the error correction circuit in figure 3. This is represented via the unitary transformation, U_{QEC} , over *both* the data and ancilla qubits,

$$\begin{aligned}
 U_{\text{QEC}}(E|\psi\rangle_L|00\rangle) &= c_0|\psi\rangle_L|00\rangle \\
 &\quad + c_1\sigma_x\sigma_I\sigma_I|\psi\rangle_L|11\rangle \\
 &\quad + c_1\sigma_I\sigma_x\sigma_I|\psi\rangle_L|10\rangle \\
 &\quad + c_1\sigma_I\sigma_I\sigma_x|\psi\rangle_L|01\rangle \\
 &\quad + c_2\sigma_x\sigma_x\sigma_I|\psi\rangle_L|10\rangle \\
 &\quad + c_2\sigma_I\sigma_x\sigma_x|\psi\rangle_L|11\rangle \\
 &\quad + c_2\sigma_x\sigma_I\sigma_x|\psi\rangle_L|01\rangle \\
 &\quad + c_3\sigma_x\sigma_x\sigma_x|\psi\rangle_L|00\rangle.
 \end{aligned} \tag{29}$$

Once again, the ancilla block is measured and the appropriate correction operator is applied, yielding the results (up to renormalization) shown in table 4. In each case, after correction (based on the syndrome result), we are left with approximately the same state, a superposition of a ‘clean state’ with the logically flipped state, $\sigma_x\sigma_x\sigma_x|\psi\rangle$. Let us now examine in detail the amplitudes now associated with each term in these states. If we consider the unitary U acting on a single, unencoded qubit, the rotation takes the state $|\psi\rangle$ to,

$$U|\psi\rangle = \cos(\epsilon)|\psi\rangle + i\sin(\epsilon)\sigma_x|\psi\rangle. \tag{30}$$

Consequently, the fidelity of the single qubit state is,

$$F_{\text{unencoded}} = |\langle\psi|U|\psi\rangle|^2 = \cos^2\epsilon \approx 1 - \epsilon^2. \tag{31}$$

Table 4. Resulting logical state after a round of error correction with the 3-qubit code.

Ancilla measurement	Collapsed state (with correction)
00	$c_0 \psi\rangle_L + c_3\sigma_x\sigma_x\sigma_x \psi\rangle_L$
01	$c_1 \psi\rangle_L + c_2\sigma_x\sigma_x\sigma_x \psi\rangle_L$
10	$c_1 \psi\rangle_L + c_2\sigma_x\sigma_x\sigma_x \psi\rangle_L$
11	$c_1 \psi\rangle_L + c_2\sigma_x\sigma_x\sigma_x \psi\rangle_L$

In contrast, the worst case fidelity³ of the encoded qubit state after a cycle of error correction is,

$$\begin{aligned}
 F_{\text{no error detected}} &= \frac{|c_0|^2}{|c_0|^2 + |c_3|^2} \\
 &= \frac{\cos^6(\epsilon)}{\cos^6(\epsilon) + \sin^6(\epsilon)} \\
 &\approx 1 - \epsilon^6,
 \end{aligned} \tag{32}$$

with probability $1 - 3\epsilon^2 + O(\epsilon^4)$ and

$$\begin{aligned}
 F_{\text{error detected}} &= \frac{|c_1|^2}{|c_1|^2 + |c_2|^2} \\
 &= \frac{\cos^4(\epsilon) \sin^2(\epsilon)}{\cos^4(\epsilon) \sin^2(\epsilon) + \sin^4(\epsilon) \cos^2(\epsilon)} \\
 &\approx 1 - \epsilon^2,
 \end{aligned} \tag{33}$$

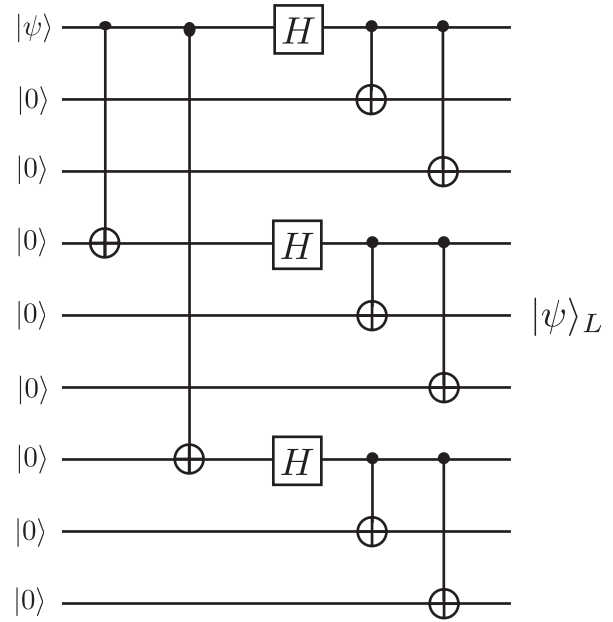
with probability $3\epsilon^2 + O(\epsilon^4)$. This is the general idea of how QEC suppresses errors at the logical level. During a round of error correction, if no error is detected, the error on the resulting state is suppressed from $O(\epsilon^2)$ to $O(\epsilon^6)$. If a single error is detected, the fidelity of the resulting state remains the same. As the 3-qubit code is a single error correcting code, if one error has already been corrected then the failure rate of the logical qubit is conditional on experiencing one further error (which will be proportional to ϵ^2). As $\epsilon \ll 1$ the majority of correction cycles will detect no error and the fidelity of the resulting encoded state is higher than when unencoded.

It should be stressed that no error correction scheme will, in general, fully restore a corrupted state to the original logical state. The resulting state will generally contain a superposition or a mixture of a clean state and a logically erred state depending on whether the error process is coherent or incoherent. The point is that the fidelity of the corrupted states, at the logical level, is greater than the corresponding fidelity for unencoded qubits. Consequently the probability of measuring the correct result at the end of a specific algorithm increases when the system is encoded. The example shown here is somewhat unphysical, i.e. it assumes perfect gate operations, errors that only consist of σ_x -rotations at a specific point in the circuit. In the coming sections we will introduce the concepts necessary when relaxing these assumptions.

5. The 9-qubit code: the first full quantum code

The 9-qubit error correcting code was first developed by Shor [Sho95] in 1995 and is based largely on the 3-qubit repetition

³ Worst case fidelity assumes that the state $|\psi\rangle_L$ is orthogonal to the state $\sigma_x\sigma_x\sigma_x|\psi\rangle_L$.

**Figure 4.** Circuit required to encode a single qubit with Shor's 9-qubit code.

code. The Shor code is a degenerate⁴ single error correcting code, able to correct a logical qubit from one bit-flip, one phase-flip or one of each, on any of the nine physical qubits. This code is therefore sufficient to correct for an arbitrary single qubit error (section 9). The two basis states for the code are

$$\begin{aligned}
 |0\rangle_L &= \frac{1}{\sqrt{8}}(|000\rangle + |111\rangle)(|000\rangle + |111\rangle)(|000\rangle + |111\rangle), \\
 |1\rangle_L &= \frac{1}{\sqrt{8}}(|000\rangle - |111\rangle)(|000\rangle - |111\rangle)(|000\rangle - |111\rangle)
 \end{aligned} \tag{34}$$

and the circuit to perform the encoding is shown in figure 4. Correction for X errors, for each block of three qubits encoded to $(|000\rangle \pm |111\rangle)/\sqrt{2}$ is identical to the 3-qubit code shown earlier. By performing the correction circuit shown in figure 3 for each block of three qubits, single $\sigma_x \equiv X$ errors can be detected and corrected. Phase errors ($\sigma_z \equiv Z$) are corrected by examining the sign differences between the three blocks. The circuit shown in figure 5 achieves this. The first set of six CNOT gates compares the sign of blocks one and two and the second set of CNOT gates compares the sign for blocks two and three. Note that a phase flip on *any* one qubit in a block of three has the same effect, this is why the 9-qubit code is referred to as a degenerate code. In other error correcting codes, such as the 5- or 7-qubit codes [Ste96a, LMPZ96], there is a one-to-one mapping between correctable errors and unique states. In degenerate codes such as the 9-qubit code, the mapping is not unique. Hence provided we know in which block the error occurs it does not matter which qubit we apply the correction operator to.

As the 9-qubit code can correct for a single X error in any one block of three and a single phase error on any of the nine qubits, this code is a full quantum error correcting code. Even if a bit *and* phase error occurs on the same qubit, the

⁴ Degenerate quantum codes are ones where different types of errors have the same effect on the codestates.

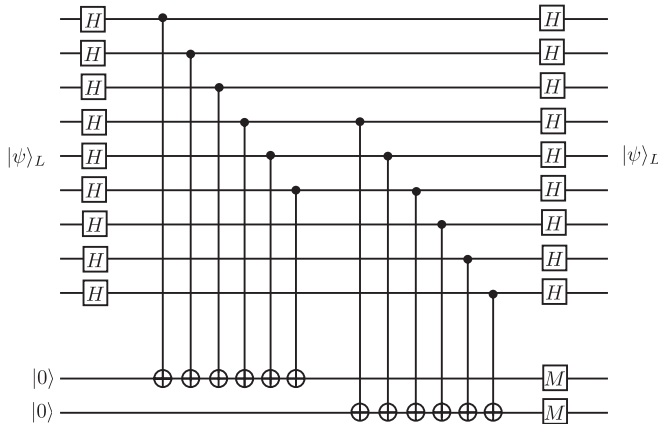


Figure 5. Circuit required to perform Z-error correction for the 9-qubit code.

X correction circuit will detect and correct for bit flips, while the Z correction circuit will detect and correct for phase flips. As mentioned, the X error correction does not have the ability to correct for up to three individual bit flips (provided each bit flip occurs in a different block of three). However, in general, the 9-qubit code is only a single error correcting code as it cannot handle multiple errors if they occur in certain locations.

The 9-qubit code is in fact related to a useful class of error correcting codes known as Bacon–Shor codes [Bac06]. These codes have the property that certain subgroups of error operators do not corrupt the logical space. For example, in the 9-qubit code, specific pairs of phase errors do not corrupt the logical states. Bacon–Shor codes are very nice codes from a computer architectural point of view. Error correction circuits and gates are generally simpler, allowing for circuit structures more amenable to the physical restrictions of a computer architecture [AC07]. Additionally, Bacon–Shor codes correcting a larger number of errors have a similar structure. Therefore, they are able to perform dynamical switching between codes, in a fault-tolerant manner. This allows us to adapt the amount of error correction to better reflect the noise present at a physical level [SEDH08]. We will return and revisit these codes later in section 14.1.

6. Quantum error detection

So far we have focused on the ability not only to detect errors, but also to correct them. Another approach is to not enforce the correction requirement. Post-selected quantum computation, developed by Knill [Kni05], demonstrated that large-scale quantum computing could be achieved with much higher noise rates when error detection is employed instead of more costly correction protocols. The basic idea in post-selected schemes is to encode a large number of ancilla qubits with error detecting circuits. Sets of encoded qubits which pass error detection are selected and further utilized as encoded ancillas for error correction. In general, error detection is faster and requires fewer qubits than performing active error correction. By producing and verifying large numbers of encoded ancillas which are post-selected after verification, error correction can be performed without data qubits waiting

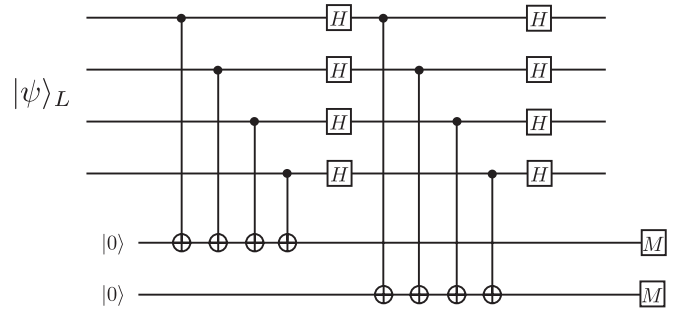


Figure 6. Circuit required to detect errors in the 4-qubit error detection code. If both ancilla measurements return $|0\rangle$, then the code state is error free. If either measurement returns $|1\rangle$, an error has occurred. Unlike the 9-qubit code, the detection of an error does not give sufficient information to correct the state.

Table 5. Qubit and ancilla state, just prior to measurement for the 4-qubit error detection code when a single bit-flip has occurred on at most one of the four qubits.

Error location	Final state, $ \text{data}\rangle \text{ancilla}\rangle$
No error	$ \psi\rangle_L 00\rangle$
Qubit 1	$X_1 \psi\rangle_L 10\rangle$
Qubit 2	$X_2 \psi\rangle_L 10\rangle$
Qubit 3	$X_3 \psi\rangle_L 10\rangle$
Qubit 4	$X_4 \psi\rangle_L 10\rangle$

as long for appropriate ancilla to be prepared, decreasing the number of errors that need to be corrected. One of the downsides to these types of schemes is that although they lead to large tolerable error rates, the resource requirements are much higher.

The simplest error detecting circuit is the 4-qubit code [GBP97]. This encodes two logical qubits on to four physical qubits with the ability to detect a single error on either of the two logical qubits. The four basis states for the code are

$$\begin{aligned}
 |00\rangle &= \frac{1}{\sqrt{2}}(|0000\rangle + |1111\rangle), \\
 |01\rangle &= \frac{1}{\sqrt{2}}(|1100\rangle + |0011\rangle), \\
 |10\rangle &= \frac{1}{\sqrt{2}}(|1010\rangle + |0101\rangle), \\
 |11\rangle &= \frac{1}{\sqrt{2}}(|0110\rangle + |1001\rangle).
 \end{aligned} \tag{35}$$

Figure 6 illustrates the error detection circuit that can be utilized to detect a single bit and/or phase flip on one of these encoded qubits. If a single bit and/or phase flip occurs on one of the four qubits then the ancilla qubits will be measured in the state $|1\rangle$. For example, let us consider the cases when a single bit flip occurs on one of the four qubits. The state of the system, just prior to the measurement of the ancilla, is given in table 5. Regardless of the location of the bit flip, the ancilla system is measured in the state $|10\rangle$. Similarly if one considers a single phase error on any of the four qubits the ancilla measurement will return $|01\rangle$. In both cases no information is obtained regarding *where* the error has occurred, hence it is not possible to correct the state. Instead the circuit must reset and re-run.

7. Stabiliser formalism

So far we have described error correcting codes from the state vector representation of their encoded states. This is a rather inefficient method for describing the codes as the state representations and circuits will differ from code to code. Consequently we would like a representation that has a generalized method for error correction and circuit construction, regardless of the code used. The majority of error correcting codes that are used within the literature are members of a class known as stabilizer codes. Stabilizer codes are very useful to work with. The general formalism applies broadly and there exist general rules to construct preparation circuits, correction circuits and fault-tolerant logical gate operations once the stabilizer structure of the code is known.

The stabilizer formalism was first introduced by Daniel Gottesman [Got97] and uses the Heisenberg representation for quantum mechanics describing quantum states in terms of operators rather than the states. A state $|\psi\rangle$ is defined to be stabilized by some operator, K , if it is a +1 eigenstate of K , i.e.

$$K|\psi\rangle = |\psi\rangle. \quad (36)$$

For example, the single qubit state $|0\rangle$ is stabilized by the operator $K = \sigma_z$, i.e.

$$\sigma_z|0\rangle = |0\rangle. \quad (37)$$

We can define multi-qubit states in the same way, by examining some of the group properties of multi-qubit operators.

Within the group of all possible, single qubit operators, there exists a subgroup, denoted the Pauli group, \mathcal{P} , which contains the following elements,

$$\mathcal{P} = \{\pm\sigma_I, \pm i\sigma_I, \pm\sigma_x, \pm i\sigma_x, \pm\sigma_y, \pm i\sigma_y, \pm\sigma_z, \pm i\sigma_z\}. \quad (38)$$

By utilizing the commutation and anti-commutation rules for the Pauli set, $\{\sigma_i\}_{i=x,y,z}$,

$$[\sigma_i, \sigma_j] = 2i\epsilon_{ijk}\sigma_k, \quad \{\sigma_i, \sigma_j\} = 2\delta_{ij}, \quad (39)$$

where

$$\epsilon_{ijk} = \begin{cases} +1 & \text{for } (i, j, k) \in \{(1, 2, 3), (2, 3, 1), (3, 1, 2)\}, \\ -1 & \text{for } (i, j, k) \in \{(1, 3, 2), (3, 2, 1), (2, 1, 3)\}, \\ 0 & \text{for } i = j, j = k, \text{ or } k = i \end{cases} \quad (40)$$

and

$$\delta_{ij} = \begin{cases} 1 & \text{for } i = j, \\ 0 & \text{for } i \neq j, \end{cases} \quad (41)$$

it is easy to show that \mathcal{P} forms a group under matrix multiplication.

The Pauli group extends over N -qubits by simply taking the N fold tensor product of all elements of \mathcal{P} , denoted as,

$$\mathcal{P}_N = \mathcal{P}^{\otimes N}. \quad (42)$$

An N -qubit stabilizer state, $|\psi\rangle_N$ is then defined by the N generators of an Abelian (all elements commute) subgroup, \mathcal{G} , of the N -qubit Pauli group,

$$\mathcal{G} = \{K^i \mid K^i|\psi\rangle = |\psi\rangle, [K^i, K^j] = 0, \forall (i, j)\} \subset \mathcal{P}_N. \quad (43)$$

The state $|\psi\rangle_N$ can be equivalently defined either through the state vector representation *or* by specifying the generators of the stabilizer group, \mathcal{G} . As each stabilized state, $|\psi\rangle$, satisfies, $K|\psi\rangle = |\psi\rangle$, and each stabilizer is Hermitian, each stabilizer squares to the identity, $K \cdot K = I$.

Many extremely useful multi-qubit states are stabilizer states, including two-qubit Bell states, Greenberger–Horne–Zeilinger (GHZ) states [GHZ89, GHSZ90], Cluster states [BR01, RB01] and codeword states for QEC. As an example, consider a 3-qubit GHZ state,

$$|\text{GHZ}\rangle_3 = \frac{|000\rangle + |111\rangle}{\sqrt{2}}. \quad (44)$$

This state can be expressed via any three linearly independent generators of the $|\text{GHZ}\rangle_3$ stabilizer group,

$$\begin{aligned} K^1 &= \sigma_x \otimes \sigma_x \otimes \sigma_x \equiv XXX, \\ K^2 &= \sigma_z \otimes \sigma_z \otimes \sigma_I \equiv ZZI, \\ K^3 &= \sigma_I \otimes \sigma_z \otimes \sigma_x \equiv IZZ, \end{aligned} \quad (45)$$

where the right-hand side of each equation is the shorthand representation of stabilizers. Similarly, the four orthogonal Bell states,

$$\begin{aligned} |\Phi^\pm\rangle &= \frac{|00\rangle \pm |11\rangle}{\sqrt{2}}, \\ |\Psi^\pm\rangle &= \frac{|01\rangle \pm |10\rangle}{\sqrt{2}}, \end{aligned} \quad (46)$$

are stabilized by the operators, $K^1 = (-1)^a XX$, and $K^2 = (-1)^b ZZ$, where $[a, b] \in \{0, 1\}$. Each of the four Bell states correspond to the four ± 1 eigenstate combinations of these two operators,

$$\begin{aligned} \Phi^+ &\equiv \begin{pmatrix} K^1 = XX \\ K^2 = ZZ \end{pmatrix} & \Phi^- &\equiv \begin{pmatrix} K^1 = -XX \\ K^2 = ZZ \end{pmatrix}, \\ \Psi^+ &\equiv \begin{pmatrix} K^1 = XX \\ K^2 = -ZZ \end{pmatrix} & \Psi^- &\equiv \begin{pmatrix} K^1 = -XX \\ K^2 = -ZZ \end{pmatrix}. \end{aligned} \quad (47)$$

8. Quantum error correction with stabilizer codes

The use of the stabilizer formalism to describe quantum error correction codes is extremely useful since it allows easy synthesis of correction circuits and also allows for quick determination of what logical operations can be applied directly on encoded data. The link between stabilizer codes and stabilizer states comes about by defining a relevant coding *subspace* within the larger Hilbert space of a multi-qubit system.

To illustrate this reduction let us examine a simple two qubit example. A 2-qubit system has a Hilbert space dimension of four, however if we require that this two qubit state

is stabilized by the XX operator, then there are only two orthogonal basis states which satisfy this condition,

$$|0\rangle_L \equiv \frac{1}{\sqrt{2}}(|01\rangle + |10\rangle), \quad |1\rangle_L \equiv \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle), \quad (48)$$

which can be used to define an effective logical qubit. Hence by using stabilizers we can reduce the size of the Hilbert space for a multi-qubit system to an effective single qubit system. In the context of QEC, the stabilizers that are used to define the logical subspace are utilized to detect and correct for errors. A stabilizer code is therefore a subspace defined via stabilizer operators for a multi-qubit system.

Returning to QEC, the example we focus on is the most well known quantum code; the 7-qubit Steane code first proposed in 1996 [Ste96a]. The 7-qubit code is defined as a $[[n, k, d]] = [[7, 1, 3]]$ quantum code, where $n = 7$ physical qubits encode $k = 1$ logical qubit with a distance between basis states $d = 3$, correcting $t = \lfloor (d - 1)/2 \rfloor = 1$ error. As the code contains a single logical qubit, the code must contain two valid code states: $|0\rangle_L$ and $|1\rangle_L$ basis states which are, in state vector notation,

$$\begin{aligned} |0\rangle_L &= \frac{1}{\sqrt{8}}(|0000000\rangle + |1010101\rangle + |0110011\rangle + |1100110\rangle \\ &\quad + |0001111\rangle + |1011010\rangle + |0111100\rangle + |1101001\rangle), \\ |1\rangle_L &= \frac{1}{\sqrt{8}}(|1111111\rangle + |0101010\rangle + |1001100\rangle + |0011001\rangle \\ &\quad + |1110000\rangle + |0100101\rangle + |1000011\rangle + |0010110\rangle). \end{aligned} \quad (49)$$

For a general 7-qubit state, the total dimension of the Hilbert space is 2^7 . For a single logically encoded qubit, we must restrict this to a two-dimensional subspace spanned by the states in equation (49). This reduction can be clearly seen via stabilizers.

The stabilizer set for the 7-qubit code is fully specified by the six operators,

$$\begin{aligned} K^1 &= IIIXXXX, & K^2 &= XIXIXIX, \\ K^3 &= IXXIIXX, & K^4 &= IIIZZZZ, \\ K^5 &= ZIZIZIZ, & K^6 &= IZZIIZZ. \end{aligned} \quad (50)$$

Each of the valid codestates for the 7-qubit code are stabilized by these operators. As the dimensionality of a 7-qubit system is 2^7 and there are six stabilizers, the total dimension of this subspace defined by the stabilizer group is $2^{7-6} = 2$. The stabilizer set now defines an effective two-dimensional subspace, and thus a single encoded qubit⁵.

The final operator fixes the encoded state to one of the two codewords. For the Steane code this operator is $\bar{Z} = ZZZZZZZ = Z^{\otimes 7}$, where $\bar{Z}|0\rangle_L = |0\rangle_L$ and $\bar{Z}|1\rangle_L = -|1\rangle_L$. Notice that these generators for the stabilizer set separate into elements consisting solely of X or Z operators. This defines the code as a Calderbank–Shor–Steane

⁵ In general, for an n qubit system, the total dimensionality of the Hilbert space is 2^n . If a stabilizer set is defined over this system containing k multiplicativity independent generators, then the dimension of the subspace is 2^{n-k} and therefore the stabilizer set defines a subspace containing $n - k$ logical qubits.

Table 6. The eight stabilizers for the 9-qubit Shor code, encoding nine physical qubits into one logical qubit to correct for a single X and/or Z error.

K^1	Z	Z	I	I	I	I	I	I	I
K^2	Z	I	Z	I	I	I	I	I	I
K^3	I	I	I	Z	Z	I	I	I	I
K^4	I	I	I	Z	I	Z	I	I	I
K^5	I	I	I	I	I	I	Z	Z	I
K^6	I	I	I	I	I	I	Z	I	Z
K^7	X	X	X	X	X	X	I	I	I
K^8	X	X	X	I	I	I	X	X	X

Table 7. The four stabilizers for the $[[5,1,3]]$ quantum code, encoding five physical qubits into one logical qubit to correct for a single X , Y or Z error. Unlike the 7- and 9-qubit codes, the $[[5,1,3]]$ code is a non-CSS code, since the stabilizer set does not separate into X and Z sectors. Additionally, this code can only correct a single error whereas the 7- and 9-qubit codes can correct for a maximum of 2 errors (a single X and Z error provided they occur on different qubits).

K^1	X	Z	Z	X	I
K^2	I	X	Z	Z	X
K^3	X	I	X	Z	Z
K^4	Z	X	I	X	Z

(CSS) code. CSS codes are useful since they allow for the straightforward application of several logical gate operations directly to the encoded data (section 11).

Although the 7-qubit code is the most well-known stabilizer code, there are other stabilizer codes which encode multiple logical qubits and correct for more errors [Got97]. Despite these advantages, larger codes generally require more complicated error correction circuits. As the complexity of the error correction circuits increases, adapting such codes to physical computer architectures becomes more difficult. Tables 6 and 7 show the stabilizer structure of two other well-known codes. The 9-qubit code [Sho95] examined earlier, and the 5-qubit code [LMPZ96] which represents the smallest possible quantum code that corrects for a single error.

8.1. State preparation

Using the stabilizer structure for QEC codes, the logical state preparation and error correcting procedure is straightforward. Recall that valid codeword states are defined as simultaneous $+1$ eigenstates of each generator of the stabilizer group. In order to prepare a logical state from an arbitrary input, we need to project qubits into eigenstates of each of these operators.

Consider the circuit shown in figure 7. For an arbitrary input state, $|\psi\rangle_I$, an ancilla which is initialized in the $|0\rangle$ state is used as a control qubit for a unitary and Hermitian operation ($U^\dagger = U$, $U^2 = I$) on $|\psi\rangle_I$. After the second Hadamard gate is performed, the state of the system is,

$$|\psi\rangle_F = \frac{1}{2}(|\psi\rangle_I + U|\psi\rangle_I)|0\rangle + \frac{1}{2}(|\psi\rangle_I - U|\psi\rangle_I)|1\rangle. \quad (51)$$

The ancilla qubit is then measured in the computational basis. If the result is $|0\rangle$, the input state is projected to (neglecting normalization),

$$|\psi\rangle_F = |\psi\rangle_I + U|\psi\rangle_I. \quad (52)$$

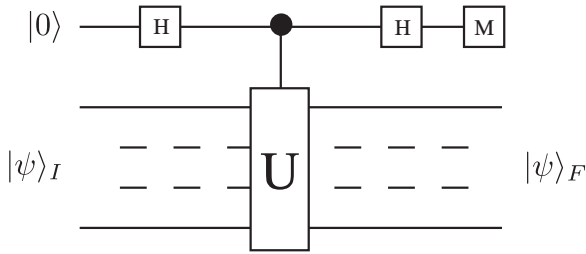


Figure 7. Quantum circuit required to project an arbitrary state, $|\psi\rangle_I$ into a ± 1 eigenstate of the Hermitian operator, $U = U^\dagger$. The measurement result of the ancilla determines which eigenstate $|\psi\rangle_I$ is projected to and since the operator, U is Hermitian and unitary can only have the two eigenvalues ± 1 .

Since U is unitary and Hermitian, $U|\psi\rangle_F = |\psi\rangle_F$, hence $|\psi\rangle_F$ is a $+1$ eigenstate of U . If the ancilla is measured to be $|1\rangle$, then the input is projected to the state,

$$|\psi\rangle_F = |\psi\rangle_I - U|\psi\rangle_I, \quad (53)$$

which is the -1 eigenstate of U . Therefore, provided U is Hermitian, the general circuit of figure 7 will project an arbitrary input state to a ± 1 eigenstate of U ⁶. This procedure is well known and is referred to as either a ‘parity’ or ‘operator’ measurement [NC00].

From this construction it should be clear how QEC state preparation proceeds. Taking the $[[7, 1, 3]]$ code as an example (the following technique is not unique or optimal for preparing and correcting the 7-qubit code [Ste97a, Ste02] but once this idea is understood, readers should have little difficulty understanding other correction techniques), 7-qubits are first initialized in the state $|0\rangle^{\otimes 7}$. The circuit shown in figure 7 is applied three times with $U = K^1, K^2, K^3$, projecting the input state into a simultaneous ± 1 eigenstate of each X generator of the stabilizer group for the $[[7, 1, 3]]$ code. The result of each operator measurement is then used to classically control a single qubit Z gate which is applied to one of the seven qubits at the end of the preparation. This single Z gate converts any -1 projected eigenstates into $+1$ eigenstates. Notice that the final three stabilizers do not need to be measured due to the input state, $|0\rangle^{\otimes 7}$, already being a $+1$ eigenstate of K^4, K^5 and K^6 . Additionally as the state $|0\rangle^{\otimes 7}$ is also a $+1$ eigenstate of K^7 , the initial state will be $|0\rangle_L$ state. Figure 8 illustrates the final circuit, where instead of one ancilla, three are utilized to speed up the state preparation by performing each operator measurement in parallel.

As a quick aside, let us detail exactly how the relevant logical basis states can be derived from the stabilizer structure of the code by utilizing this preparation procedure. We will use the stabilizer set shown in table 7 to calculate the $|0\rangle_L$ state for the 5-qubit code as we have not yet explicitly shown the state vectors for the two logical state. The four stabilizer generators are given by,

$$\begin{aligned} K^1 &= XZZXI, & K^2 &= IXZZX, \\ K^3 &= XIXZZ, & K^4 &= ZXIXZ. \end{aligned} \quad (54)$$

⁶ An operator, U , which is both Hermitian and unitary can only have eigenvalues of ± 1 .

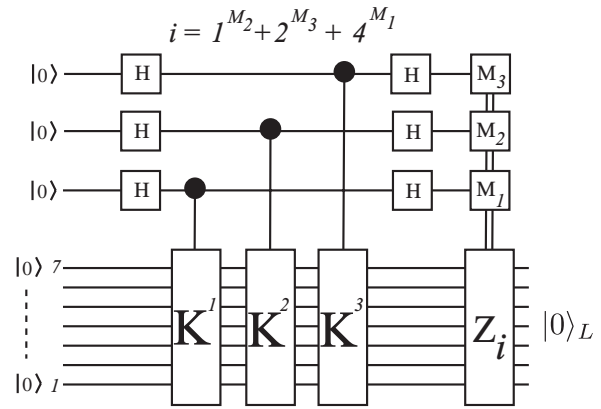


Figure 8. Quantum circuit to prepare the $[[7, 1, 3]]$ logical $|0\rangle$ state. The input state $|0\rangle^{\otimes 7}$ is projected into an eigenstate of each of the X stabilizers shown in equation (50). After each ancilla measurement the classical results are used to apply a single qubit Z gate to qubit $i = 1^{M_2} + 2^{M_3} + 4^{M_1}$ which converts the state from a -1 eigenstates of (K^1, K^2, K^3) to $+1$ eigenstates.

In an identical way to the 7-qubit code, projecting an arbitrary state into a $+1$ eigenstate of these operators defines the two logical basis states, $|0\rangle_L$ and $|1\rangle_L$. The logical operator, $\bar{Z} = ZZZZZ$, then fixes the state to either $|0\rangle_L$ or $|1\rangle_L$. Therefore, calculating $|0\rangle_L$ from some initial unencoded state requires us to project the initial state into a $+1$ eigenstate of these operators. If we take the initial, unencoded state as $|00000\rangle$, then it is already a $+1$ eigenstate of \bar{Z} . Therefore, to find $|0\rangle_L$ we simply calculate

$$|0\rangle_L = \prod_{i=1}^4 (I^{\otimes 5} + K^i) |00000\rangle, \quad (55)$$

up to normalization. Expanding out this product, we find,

$$\begin{aligned} |0\rangle_L &= \frac{1}{4} (|00000\rangle + |01010\rangle + |10100\rangle - |11110\rangle \\ &\quad + |01001\rangle - |00011\rangle - |11101\rangle - |10111\rangle \\ &\quad + |10010\rangle - |11000\rangle - |00110\rangle - |01100\rangle \\ &\quad - |11011\rangle - |10001\rangle - |01111\rangle + |00101\rangle). \end{aligned} \quad (56)$$

Note that the above state vector does not match up with those given in [LMPZ96]⁷. However, these vectors are equivalent up to local rotations on each qubit.

8.2. Error correction

Error correction using stabilizer codes is an extension of the state preparation. Consider an arbitrary single qubit state that has been encoded as

$$\alpha|0\rangle + \beta|1\rangle \rightarrow \alpha|0\rangle_L + \beta|1\rangle_L = |\psi\rangle_L. \quad (57)$$

Now assume that an error occurs on one (or multiple) qubits which is described via the operator E , where E is a combination of X and/or Z errors over the N physical qubits of the logical state (and therefore an element of the N -qubit Pauli group, \mathcal{P}_N). By definition of stabilizer codes, $K^i|\psi\rangle_L = |\psi\rangle_L$,

⁷ The stabilizer formalism was introduced after the 5-qubit code.

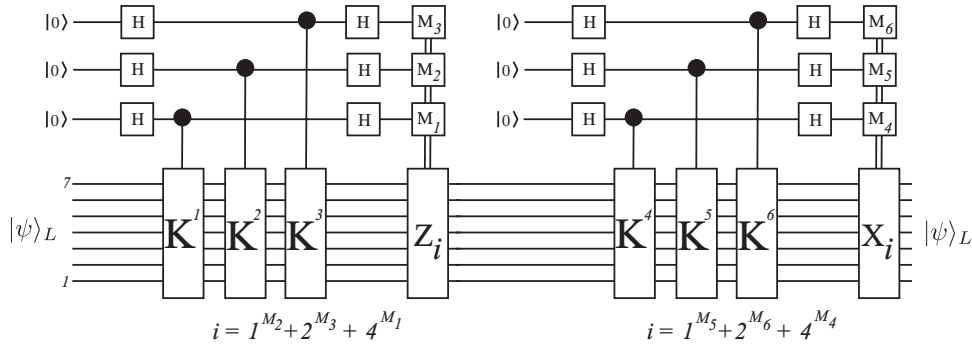


Figure 9. Quantum circuit to correct for a single X and/or Z error using the $[[7, 1, 3]]$ code. Each of the six stabilizers is measured, with the first three detecting and correcting for Z errors, while the last three detect and correct for X errors.

$i \in [1, \dots, N - k]$, for a code encoding k logical qubits. Hence the erred state, $E|\psi\rangle_L$, satisfies,

$$K^i E|\psi\rangle_L = (-1)^m E K^i |\psi\rangle_L = (-1)^m E|\psi\rangle_L, \quad (58)$$

where m is defined as $m = 0$, if $[E, K^i] = 0$ and $m = 1$, if $\{E, K^i\} = 0$ (E and K are Pauli group operators and all Pauli group operators either commute or anti-commute). Therefore, if the error operator commutes with the stabilizer, the state remains a $+1$ eigenstate of K^i , if the error operator anti-commutes with the stabilizer then the logical state is flipped to a -1 eigenstate of K^i .

The general procedure for error correction is identical to state preparation. Each of the code stabilizers is sequentially measured. Since an error free state is already a $+1$ eigenstate of all the stabilizers, errors which anti-commute with any of the stabilizers describing the code will flip the relevant eigenstate and consequently measuring the parity of these stabilizers will return a result of $|1\rangle$. Taking the $[[7, 1, 3]]$ code as an example, if the error operator is $E = X_i$ (where $i = 1, \dots, 7$), representing a bit-flip on any *one* of the 7 physical qubits, then regardless of the location, E will anti-commute with a unique combination of K^4, K^5 and K^6 . Hence the classical results of measuring these three operators will indicate if and where a single X error has occurred. Similarly, if $E = Z_i$, then the error operator will anti-commute with a unique combination of, K^1, K^2 and K^3 . Consequently, the first three stabilizers for the $[[7, 1, 3]]$ code correspond to Z error correction while the second three stabilizers correspond to X error correction. Note that correction for Pauli Y errors is achieved by correcting in the X and Z sector since a Y error on a single qubit is equivalent to both an X and Z error on the same qubit, i.e. $Y = iXZ$.

The circuit shown in figure 9 illustrates the circuit for full error correction with the $[[7, 1, 3]]$ code. As you can see it is simply an extension of the preparation circuit shown in figure 8, where all six stabilizers are measured across the data block. Even though we have specifically used the $[[7, 1, 3]]$ code as an example, this procedure for error correction and state preparation will work for all stabilizer codes (although other correction procedures are possible [Ste97a, Ste02, Kni05]).

9. Digitization of quantum errors

Up until now we have remained fairly abstract regarding the analysis of quantum errors. Specifically, we have examined

QEC from the standpoint of a discrete set of Pauli errors occurring at certain locations within a larger quantum circuit. In this section we examine how this analysis of errors relates to some of the more realistic processes such as environmental decoherence and systematic gate errors.

Digitization of quantum noise is often assumed when people examine the stability of quantum circuit design or attempt to calculate thresholds for concatenated error correction (section 10.4). However, the link between discrete Pauli errors and more general, continuous noise only makes sense when we consider the stabilizer nature of the correction procedure. Recall from section 7 that correction is performed by re-projecting a potentially corrupt data block into $+1$ eigenstates of the code stabilizers. A general continuous mapping from a ‘clean’ codeword state to a corrupt one will not be eigenstates of the code stabilizers. Instead they will be in superpositions of eigenstates. Measuring the parity of each of the stabilizers acts to digitize the quantum noise. We will first introduce how a coherent systematic error, caused by imperfect implementation of quantum gates, is digitized during correction, after which we will briefly discuss environmental decoherence from the standpoint of a Markovian decoherence model.

9.1. Systematic gate errors

We have already shown a primitive example of how systematic gate errors are digitized into a discrete set of Pauli operators in section 3. However, in that case we only considered a very restrictive type of error, namely the coherent operator $U = \exp(i\epsilon X)$. We can easily extend this analysis to cover all types of systematic gate errors. Consider an N qubit unitary operation, U_N , which is valid on encoded data. Assume that U_N is applied inaccurately such that the resultant operation is actually U_N . Given a general encoded state $|\psi\rangle_N$, the final state can be expressed as

$$U_N |\psi\rangle_L = U_E U_N |\psi\rangle_L = \sum_j \alpha_j E_j |\psi_2\rangle_L, \quad (59)$$

where $|\psi_2\rangle_L = U_N |\psi\rangle_L$ is the perfectly applied N qubit gate, (the stabilizer group for $|\psi\rangle_L$ remains invariant under the operation U_N (see section 11)) and U_E is a coherent error operator which is expanded in terms of the N qubit Pauli Group, $E_j \in P_N$. Now append two ancilla blocks, $|A_0\rangle^X$

and $|A_0\rangle^Z$. These are generally initialized to the state that corresponds to no detected errors (but can be other initial states up to a redefinition of the measurement results) and are used for X and Z correction. We then run the syndrome extraction procedure, which we represent by the unitary operator, U_{QEC} . It will be assumed that $|\psi\rangle_L$ is encoded with a QEC code which can correct for a single error (both X and/or Z), and the error operators E_j are a maximum of weight one (i.e. E_j contains at most one non-identity term e.g. $E_1 = X_1 \otimes I^{\otimes(N-1)}$)⁸, hence there is a one-to-one mapping between the error operators, E_j , and the orthogonal basis states of the ancilla blocks,

$$\begin{aligned} & U_{\text{QEC}} U'_N |\psi\rangle_L |A_0\rangle^X |A_0\rangle^Z \\ &= U_{\text{QEC}} \sum_j \alpha_j E_j |\psi\rangle_L |A_0\rangle^X |A_0\rangle^Z \\ &= \sum_j \alpha_j E_j |\psi\rangle_L |A_j\rangle^X |A_j\rangle^Z. \end{aligned} \quad (60)$$

The above assumes a non-degenerate quantum code⁹. The ancilla blocks are then measured, projecting the data blocks into the state $E_j |\psi\rangle_L$ with probability $|\alpha_j|^2$. After measurement the correction E_j^\dagger is applied based on the syndrome result. As the error operation E_j is simply an element of \mathcal{P}_N , correcting for X and Z independently is sufficient to correct for all error operators (as Y errors are corrected when a bit and phase error is detected and corrected on the same qubit).

For well designed gates, very small systematic inaccuracies lead to the expansion coefficient $\alpha_0 \approx 1$, with all other coefficients, $\alpha_{j \neq 0} \ll 1$. Hence during correction there will be a very high probability that no error is detected. This is the digitization effect of QEC. Since codeword states are eigenstates of the stabilizers, re-projecting the state when each stabilizer is measured forces any continuous noise operator to collapse. The strength of the error is then related to the probability that the data block collapses to a specific Pauli error acting on the codestate. The physical mechanisms which are used to construct quantum gates are what determine the form of U_E and hence the form of the error operators, E_j and their respective coefficients, α_j .

9.2. Environmental decoherence

A complete analysis of environmental decoherence in relation to quantum information is a lengthy topic. Instead of a detailed review, we will instead present a simplified example to highlight how QEC relates to environmental effects.

The Lindblad formalism [Gar91, NC00, DWM03] provides an elegant method for analysing the effect of decoherence on open quantum systems. This model does have several

⁸ This assumption is for demonstration purposes. In reality, all qubits will experience errors and hence E_j can be of higher weight (up to a weight N operator on an N qubit system). The ability of the error correction code to correct for higher weight errors depends on how all these E_j map the ancilla states under U_{QEC} .

⁹ A degenerate quantum code is one where multiple unique errors can map to the same state; in the case of equation (60) this would mean two operators E_j and E'_j , under the unitary U_{QEC} map $|A_0\rangle^X$ and $|A_0\rangle^Z$ to the same ancilla state.

assumptions, most notably that the environmental bath couples weakly to the system (Born approximation), the system and environment are initially in some separable state and that each qubit experiences temporally un-correlated noise (Markovian approximation). While these assumptions are utilized for a variety of systems [BHPC03, BM03, BKD04], it is known that they may not hold in some cases [HMCS00, MCM⁺05, APN⁺05, ALKH02]. This is particularly important in superconducting systems, where decoherence can be caused by small numbers of fluctuating charges which induces coloured noise therefore violating the assumption of non-Markovian dynamics. In this case more specific decoherence models need to be considered.

Using this formalism, the evolution of the density matrix can be written as

$$\partial_t \rho = -\frac{i}{\hbar} [H, \rho] + \sum_k \Gamma_k \mathcal{L}_k[\rho], \quad (61)$$

where H is the Hamiltonian, representing coherent, dynamical evolution of the system and $\mathcal{L}_k[\rho] = ([L_k, \rho L_k^\dagger] + [L_k \rho, L_k^\dagger])/2$ represents the incoherent evolution. The operators L_k are known as the Lindblad quantum jump operators and are used to model specific decoherence channels, with each operator associated with some rate $\Gamma_k \geq 0$. This differential equation is known as the quantum Liouville equation or more generally, the density matrix master equation.

To link Markovian decoherence to QEC, consider a special set of decoherence channels which represent a single qubit undergoing dephasing, spontaneous emission and spontaneous absorption. This helps to simplify the calculation. Dephasing of a single qubit is modelled by the Lindblad operator $L_1 = Z$ while spontaneous emission/absorption are modelled by the operators $L_2 = |0\rangle\langle 1|$ and $L_3 = |1\rangle\langle 0|$, respectively. For the sake of simplicity, we assume that absorption/emission occur at the same rate, Γ . Although this is unphysical, it does simplify the calculation significantly for this example. The density matrix evolution is given by,

$$\partial_t \rho = -\frac{i}{\hbar} [H, \rho] + \Gamma_Z (Z \rho Z - \rho) + \frac{\Gamma}{2} (X \rho X + Y \rho Y - 2\rho). \quad (62)$$

If we assumed that the qubit is not undergoing any coherent evolution ($H = 0$), i.e. a memory stage within a quantum algorithm, then equation (62) can be solved by re-expressing the density matrix in the Bloch formalism. Set $\rho(t) = I/2 + x(t)X + y(t)Y + z(t)Z$, then equation (62) with $H = 0$ reduces to $\partial_t S(t) = AS(t)$ with $S(t) = (x(t), y(t), z(t))^T$ and

$$A = \begin{pmatrix} -(\Gamma + 2\Gamma_Z) & 0 & 0 \\ 0 & -(\Gamma + 2\Gamma_Z) & 0 \\ 0 & 0 & -2\Gamma \end{pmatrix}. \quad (63)$$

This differential equation is easy to solve, leading to

$$\begin{aligned} \rho(t) &= [1 - p(t)]\rho(0) + p_x(t)X\rho(0)X \\ &+ p_y(t)Y\rho(0)Y + p_z(t)Z\rho(0)Z, \end{aligned} \quad (64)$$

where,

$$\begin{aligned} p_x(t) &= p_y(t) = \frac{1}{4}(1 - e^{-2\Gamma t}), \\ p_z(t) &= \frac{1}{4}(1 + e^{-2\Gamma t} - 2e^{-(\Gamma+2\Gamma_Z)t}), \\ p(t) &= p_x(t) + p_y(t) + p_z(t). \end{aligned} \quad (65)$$

If this single qubit is part of an encoded data block, then each term represents a single error on the qubit experiencing decoherence. Two blocks of ancilla qubits, initialized to the state corresponding to no detected error, are added to the system. The error correction protocol is then run. Once the ancilla qubits are measured, the state will collapse to no error with probability $1 - p(t)$, or a single X , Y or Z error, with probabilities $p_x(t)$, $p_y(t)$ and $p_z(t)$ respectively. Although the above example is somewhat artificial, the above should allow the reader to perform their own calculations for expected QEC error rates given a more physical complete error model.

We can also see how temporal effects are incorporated into the error correction model. The temporal integration window t of the master equation will influence how probable an error is detected for a fixed rate Γ . The longer between correction cycles, the more probable the qubit experiences an error.

9.3. More general mappings

Both the systematic gate errors and the errors induced by environmental decoherence illustrate the digitization effect of QEC. However, we can generalize digitization to other mappings of the density matrix. In this case consider a more general Krauss map on a multi-qubit density matrix,

$$\rho \rightarrow \sum_k A_k^\dagger \rho A_k, \quad (66)$$

where $\sum_k A_k^\dagger A_k = I$. For the sake of simplicity let us choose a simple mapping where $A_1 = (Z_1 + iZ_2)/\sqrt{2}$ and $A_k = 0$ for $k \neq 1$. This mapping essentially represents dephasing on two qubits. However, this type of mapping (when considered in the context of error correction) represents independent Z errors on either qubit one or two.

The above discussion assumes that we are working with a non-degenerate quantum code. If the code is degenerate, i.e. if the errors Z_1 and Z_2 have the same effect on the codestates then the correction protocol does not simplify this error mapping as described above.

To illustrate, first expand out the density matrix (neglecting normalization),

$$\rho \rightarrow A_1^\dagger \rho A_1 = Z_1 \rho Z_1 + Z_2 \rho Z_2 - i Z_1 \rho Z_2 + i Z_2 \rho Z_1. \quad (67)$$

Note that only the first two terms in this expansion, on their own, represent physical mixtures. However, the last two off-diagonal terms are removed by the process of syndrome extraction and are irrelevant in the context of QEC. To illustrate, we assume that ρ represents a protected qubit, where Z_1 and Z_2 are *physical* errors on qubits comprising the code block. As we are only considering phase errors in this example, we will ignore X correction (but the analysis automatically generalizes if the error mapping contains X terms). A fresh ancilla block, represented by the density matrix ρ_0^z , is coupled to the system and the unitary U_{QEC} is run,

$$\begin{aligned} U_{QEC}^\dagger \rho' \otimes \rho_0^z U_{QEC} = & Z_1 \rho Z_1 \otimes |Z_1\rangle\langle Z_1| \\ & + Z_2 \rho Z_2 \otimes |Z_2\rangle\langle Z_2| \\ & - i Z_1 \rho Z_2 \otimes |Z_1\rangle\langle Z_2| \\ & + i Z_2 \rho Z_1 \otimes |Z_2\rangle\langle Z_1|, \end{aligned} \quad (68)$$

where $|Z_1\rangle$ and $|Z_2\rangle$ represent the two orthogonal syndrome states of the ancilla that are used to detect phase errors on qubits one and two respectively. The important part of the above expression is that when the syndrome qubits are measured, the state collapses to,

$$\begin{aligned} \rho & \rightarrow \frac{\langle Z_1 | \rho | Z_1 \rangle | Z_1 \rangle \langle Z_1 |}{\text{Tr}(\rho | Z_1 \rangle \langle Z_1 |)} \\ \text{or } \rho & \rightarrow \frac{\langle Z_2 | \rho | Z_2 \rangle | Z_2 \rangle \langle Z_2 |}{\text{Tr}(\rho | Z_2 \rangle \langle Z_2 |)}. \end{aligned} \quad (69)$$

The two cross terms in the above expression are never observed. In this mapping the only two possible states that exist after the measurement of the ancilla system are,

$$\begin{aligned} Z_1 \rho Z_1 \otimes |Z_1\rangle\langle Z_1| & \quad \text{with Probability} = \frac{1}{2}, \\ Z_2 \rho Z_2 \otimes |Z_2\rangle\langle Z_2| & \quad \text{with Probability} = \frac{1}{2}. \end{aligned} \quad (70)$$

Therefore, not only are the cross terms eliminated via error correction but the final density matrix again collapses to a single error perturbation of ‘clean’ codeword states with no correlated errors.

Consequently, it is common in standard QEC analysis to assume that discrete X and/or Z errors are applied stochastically to each qubit after each elementary gate operation, measurement, initialization and memory step with some probability p . The QEC protocol itself digitizes the continuous noise, either coherent or incoherent errors, into a discrete set of bit and/or phase flips. The set of discrete errors is determined by the set of errors that are distinguishable by the quantum code used. The magnitude of the continuous error is translated to the probability of detecting a discrete error. In this way error correction can be analysed by assuming perfect gate operations and discrete, probabilistic errors. The probability of these errors occurring can then be independently calculated via analysis of the physical mechanisms which produce errors. While a local, stochastic error model is generally the most common, depending on the physical system under consideration and the type of quantum codes being used a more complicated analysis may be needed [Pre98].

10. Fault-tolerant quantum error correction and the threshold theorem

Section 7 detailed the protocols required to correct for quantum errors, however in the discussion so far, we have implicitly assumed the following,

1. Errors only occur during ‘memory’ regions, i.e. when quantum operations or error correction are not being performed and ancilla qubits are error free.
2. The quantum gates themselves do not induce any systematic errors within the logical data block.

Clearly these are two very unrealistic assumptions. Fault-tolerant QEC is how we address these two issues. As the name suggests, fault-tolerance is a design methodology that allows us to tolerate faults, allowing QEC to remain effective. Combining error correction techniques with fault-tolerance allows us to design correction procedures and logical gate operations such that they can still function when the above assumptions are relaxed.

10.1. Error propagation

Before discussing the nature of fault-tolerant computation, we first examine how errors can propagate in quantum circuits. There are essentially two dominant channels that can cause errors to be copied: the first is quantum gates. For obvious reasons, gates operating on single qubits do not copy errors, however quantum gates which couple qubits can cause errors to propagate.

For example, take the two qubit state $E_j|\psi\rangle$, where $E_j = \{X_1I_2, Y_1I_2, Z_1I_2, I_1X_2, I_1Y_2, I_1Z_2\}$ are each single qubit errors (either on qubit one or two respectively). We now perform a CNOT operation, $U = \text{CNOT}$, where the control is qubit one. This gives,

$$UE_j|\psi\rangle = (UE_jU^\dagger)U|\psi\rangle, \quad (71)$$

using the identity $(U^\dagger U) = I$. The effect of the gate is to transform the error E_j to $E'_j = (UE_jU^\dagger)$. For the six single qubit errors, this gives

$$\begin{aligned} U(X_1I_2)U^\dagger &= X_1X_2, \\ U(Y_1I_2)U^\dagger &= Y_1X_2, \\ U(Z_1I_2)U^\dagger &= Z_1I_2, \\ U(I_1X_2)U^\dagger &= I_1X_2, \\ U(I_1Y_2)U^\dagger &= Z_1Y_2, \\ U(I_1Z_2)U^\dagger &= Z_1Z_2. \end{aligned} \quad (72)$$

Therefore, the CNOT gate copies X errors from the control qubit to the target and copies Z errors from the target to the control. Error propagation rules for any multi-qubit unitary can also be calculated for an arbitrary gate, U , and error, E^{10} .

The above example assumes a perfect 2-qubit gate operation, however the gates themselves can also introduce multiple errors when they fail. In the case of coherent errors, an arbitrary 2-qubit unitary, U , can be written in the form, $U_E U_p$, where U_p is the perfectly applied gate and U_E is some coherent error operator. This error operator can be expressed as a linear combination of the 2-qubit Pauli group,

$$U_E = \sum_{(i,j)=1}^4 d_{i,j} \sigma_i \otimes \sigma_j, \quad [\sigma_1, \sigma_2, \sigma_3, \sigma_4] = [I, X, Y, Z]. \quad (73)$$

Therefore inaccurate application of the gate U_p (including complete failure) can introduce a single X , Y or Z error to each of the two qubits or it can introduce an error to *both* qubits (the details of what kind of errors can be introduced by faulty gates are dependent on the mechanisms causing the failure). Therefore, not only can gates coupling qubits copy pre-existing errors but a single failure event can introduce errors on both qubits involved in the interaction. Correlated errors also occur in the density matrix representation when incoherent noise causes gate inaccuracies. This generalizes for higher order operations, e.g. errors on three qubit gates can introduce errors on all three qubits, etc.

¹⁰ If U is not a member of the Clifford group (operators which map Pauli operators to Pauli operators) then E will map to a linear combination of Pauli errors.

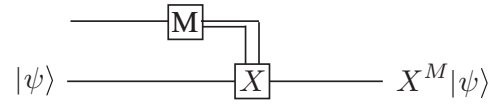


Figure 10. Error propagation due to measurement error. If the measurement result for qubit one is faulty, we inadvertently introduce an error to the second qubit.

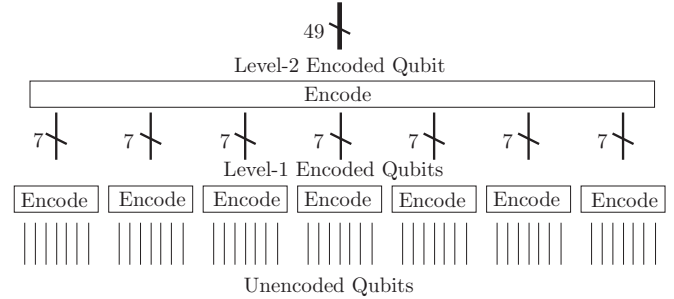


Figure 11. Illustration explaining the concept of concatenation with the $[[7, 1, 3]]$ code. 49 physical qubits are first encoded into seven level-1 logical qubits (seven blocks of seven qubits). These seven logical qubits are then encoded into a level-2 logical qubit. This level-2 qubit can correct up to a maximum of $((3^2 - 1)/2) = 4$ errors on the 49 physical qubits forming the encoded block.

A second process which can cause errors to cascade is the classical correction of quantum data from a measurement result¹¹. Take the circuit shown in figure 10. The measurement result on the top qubit is used to determine if the gate, X , is applied to the second qubit. In this case the gate is applied if the upper qubit is measured in the state $|1\rangle$. If we now assume that this measurement is faulty, then with some probability the ‘classical’ information extracted from the measurement device is wrong. Therefore the correction that is applied to the second qubit should not have been applied. Not only has a bit-flip error now occurred on qubit one (due to the inaccurate measurement result), but we also have accidentally introduced a bit-flip error on the second qubit due to this inaccurate information.

Hence gates which couple qubits and measurement errors (which subsequently control further quantum gates) can copy errors from qubit to qubit.

10.2. Concatenation

Concatenation is where a group of encoded qubits are further encoded (not necessarily with the same error correction code). This forms a second level encoded qubit which figure 11 illustrates for the $[[7, 1, 3]]$ code. Starting with a group of unencoded qubits, we first encode into level-1 logical qubits, each containing a block of seven. Each of these logical qubits are now protected with a distance three quantum code. If we now take seven of these level-1 logical qubits and perform the same encoding operation (using valid encoded gate operations, discussed in the following sections) we form a level-2 logical qubit. This is now a block containing 49 physical qubits and has a code distance of $3^2 = 9$. A level-2 logical qubit can

¹¹ It should be noted that this process is equivalent to directly coupling qubits with quantum gates and then measuring one of the qubits, hence errors will be copied in the same manner as direct coupling.

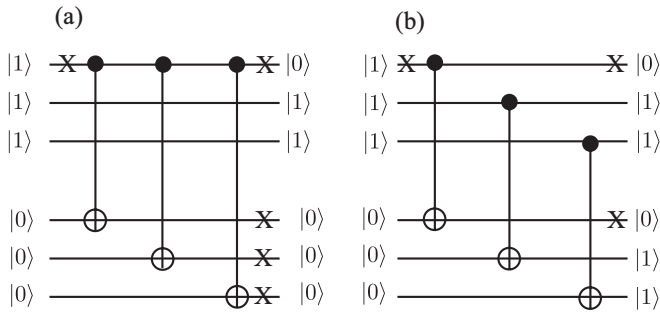


Figure 12. Two circuits to implement the transformation $|111\rangle|000\rangle \rightarrow |111\rangle|111\rangle$. (a) shows a version where a single X error can cascade into four errors while (b) shows an equivalent circuit where the error only propagates to a second qubit.

therefore be faithfully recovered if four or less errors occur on any of the 49 physical qubits. This process can be repeated indefinitely, for g levels of concatenation, the encoded state will consist of 7^g physical qubits and have a code distance of 3^g , allowing for the correction of $(3^g - 1)/2$ individual errors. Hence for a concatenation scheme, both the number of physical qubits required and the number of errors it can correct scale exponentially (with the number of qubits growing faster than the number of correctable errors).

10.3. Fault-tolerance

The concept of fault-tolerance in computation is not a new idea, it was first developed in relation to classical computing [Neu55, G83, Avi87]. However, in recent years the precise manufacturing of digital circuitry has made large-scale error correction and fault-tolerant circuits largely unnecessary.

The basic principle of fault-tolerance is that the circuits used for gate operations and error correction procedures should not cause errors to cascade. As shown in the previous section, gates which couple multiple qubits and measurements which are used to control a quantum operation can cause errors to be copied. How do we design operations to avoid cascading errors in quantum algorithms?

This can be seen more clearly when we look at a simple CNOT operation between two qubits (figure 12). In this circuit we are performing a sequence of three CNOT gates which act to take the state $|111\rangle|000\rangle \rightarrow |111\rangle|111\rangle$. In figure 12(a) we consider a single X error which occurs on the top most qubit prior to the first CNOT. This single error will cascade through each of the three gates such that the X error has now propagated to four qubits. Figure 12(b) shows a slightly modified design that implements the same operation, but the single X error now only propagates to two of the six qubits. If we consider each block of three as a single logical qubit, then the staggered circuit will only induce a total of one error in each logical block, given a single X error occurred somewhere during the circuit. Therefore, one of the standard definitions of fault-tolerance is

fault-tolerant circuit element: A single error will cause at most one error in the output for each logical qubit block.

It should be stressed that the idea of fault-tolerance is a discrete definition: either a certain quantum operation is fault-tolerant or it is not. What is defined to be fault-tolerant can change depending on the error correction code used. For example, for a single error correcting code, the above definition is the only one available, since any more than one error in a logical qubit will result in the error correction procedure failing. However, if the quantum code employed is able to correct multiple errors, then the definition of fault-tolerance can be relaxed, i.e. if the code can correct three errors then circuits may be designed such that a single failure results in at most three errors in the output (which is then correctable). In general, for a code correcting $t = \lfloor (d-1)/2 \rfloor$ errors, fault-tolerance requires that $\leq t$ errors during an operation does not result in $>t$ errors in the output for each logical qubit.

10.4. Threshold theorem

The threshold theorem is truly a remarkable result in quantum information and is a consequence of fault-tolerant circuit design and the ability to perform dynamical error correction. Rather than present a detailed derivation of the theorem for a variety of noise models, we will instead take a very simple case where we utilize a quantum code that can only correct for a single error, using a model that assumes uncorrelated errors on individual qubits. For more rigorous derivations of the theorem see [ABO97, Got97, Ali07].

Consider a quantum computer where each physical qubit experiences either an X and/or Z error independently with probability p , per gate operation. Furthermore, it is assumed that logical gate operations and error correction circuits are designed according to the rules of fault-tolerance and that a cycle of error correction is performed after each elementary *logical* gate operation using a code that corrects for a single error ($[[n, k, 3]]$ code). If an error occurs during a logical gate operation, then fault-tolerance ensures this error will only propagate to at most one error in each block, after which a cycle of error correction will remove the error. Hence if the failure probability of unencoded qubits per time step is p , then a single level of error correction will ensure that the logical step fails only when two (or more) errors occur. Hence the failure rate of each logical operation, to leading order, is now $p_L^1 = cp^2$, where p_L^1 is the failure rate (per logical gate operation) of a level-1 logical qubit and c is the upper bound for the number of possible 2-error combinations which can occur at a physical level within the sequence of operations consisting of a correction cycle, the logical gate operation and a second correction cycle [Ali07]. We now concatenate, encoding the computer further, such that a level-2 logical qubit is formed. We assume this is done using the same $[[n, k, 3]]$ quantum code level-1 encoded qubit. It is assumed that all error correcting procedures and gate operations at level-2 are self-similar to the level-1 operations (i.e. the circuit structures for the level-2 encoding are identical to the level-1 encoding). Therefore, if the level-1 failure rate per logical time step is p_L^1 , then by the same argument, the failure rate of a level-2 operation is given by, $p_L^2 = c(p_L^1)^2 = c^3 p^4$. This iterative procedure is then

Table 8. Various threshold estimates, associated geometric constraints and architectural considerations when performing the calculation. We have ordered the table with respect to the level of architectural detail that has been developed compatible with the threshold result. Note that this is not an exhaustive list.

Relevant work	Code	Threshold	Geometric constraints	Architectural context
[ABO08]	[[7, 1, 3]]	$O(10^{-6})$	A.I. ^a	None
[Got97]	[[7, 1, 3]]	$O(10^{-4} - 10^{-6})$	A.I.	None
[KLZ96]	[[7, 1, 3]]	$O(10^{-6})$	A.I.	None
[PR12]	[[23, 1, 7]]	$O(10^{-3})$	A.I.	None
[Kni05]	4- and 6-qubit detection	$O(10^{-2})$	A.I.	None
[BAO ⁺ 12]	Toric code	18%	2D NN ^b array, P.B.C. ^c	None, theoretical upper bound ^d
[MCT ⁺ 04]	[[7, 1, 3]] ^e	$O(10^{-4})$	A.I. with movement penalty	Specific to ion-traps
[SFH08]	[[7, 1, 3]]	$O(10^{-6})$	Bilinear NN array	Kane P : Si system [Kan98]
[SBF ⁺ 06]	[[7, 1, 3]]	$O(10^{-7})$	Variable width NN array	General NN systems
[SDT07]	[[7, 1, 3]]	$O(10^{-5})$	2D NN arrays	General NN systems
[FTY ⁺ 07]	[[7, 1, 3]]	$O(10^{-6})$	Bilinear NN array	Superconducting qubits
[BKSO05]	[[7, 1, 3]]	$O(10^{-9})$	A.I. with movement penalty	Ion-traps
[RHG07]	Topological cluster	$O(10^{-2} - 10^{-3})$	3D NN array	Photonic qubits [DFS ⁺ 09]
[WFSH10]	Surface codes	$O(10^{-2} - 10^{-3})$	2D NN array	Quantum dots, diamond [JMF ⁺ 12, YJG ⁺ 12]

^a Arbitrary interactions.

^b Nearest neighbour.

^c Periodic boundary conditions.

^d This result is calculated assuming perfect quantum gates and is known as the code capacity.

^e Only a preliminary estimate.

repeated up to level- g , such that the logical failure rate, per time step, of a level- g encoded qubit is given by

$$p_L^g = \frac{(cp)^{2g}}{c}. \quad (74)$$

Equation (74) implies that for a finite *physical* error rate, p , per qubit, per time step, the failure rate of a level- g encoded qubit can be made arbitrarily small if $cp < 1$. This inequality defines the threshold. The physical error rate experienced by each qubit per time step must be $p_{th} < 1/c$ to ensure that multiple levels of concatenated error correction reduce the failure rate of logical components.

Hence, provided sufficient resources are available, an arbitrarily large quantum circuit can be successfully implemented to arbitrary accuracy, once the physical error rate is below threshold. The calculation of thresholds is therefore an extremely important aspect to quantum architecture design. Initial estimates at the threshold, which gave $p_{th} \approx 10^{-4} - 10^{-6}$ [Kit97, ABO97, Got97], did not sufficiently model the physical architectures. Recent results [SFH08, SDT07, SBF⁺06, MCT⁺04, BKSO05] have been estimated for more realistic quantum processor architectures, showing significant differences in threshold when architectural considerations are taken into account. Many of these estimates are summarized in table 8 with a brief description of the constraints considered by the model. We order the entries by the level of detail presented in each of the papers with regards to physical implementation of fault-tolerant error correction protocols on realistic quantum architectures.

From the above table you can see the differences in threshold that occur once more specific architectural considerations are taken into account. This variation is heavily dependent on what error models are assumed in the analysis, if qubit transport can be done directly (such as in ion traps where the qubits are physically moved throughout the computational

array) or via the application of SWAP gates (required in condensed matter systems such as P : Si, quantum dots or NV defects in diamond) and what is the structure of the QEC code that is ultimately used. In terms of concatenated QEC codes, the majority of analysis was done using the [[7, 1, 3]] Steane code and only the results of Knill [Kni05] show a threshold of the order of 1%. However, the results of Knill have not yet been adapted to a physical architecture and it remains unclear if the constraints of a 1D, 2D or 3D nearest neighbour architecture will reduce the threshold to that of other concatenated codes and/or substantially increase qubit resources to achieve a high threshold.

Currently, the most detailed architectural designs are based on topological cluster codes [RHG07] and surface codes [Kit97, DKLP02, FSG09]. The general framework for this type of error correction is discussed later in section 14.2. These codes are promising for large-scale computers as they exhibit significantly higher thresholds to concatenation and they have an intrinsic structure that is compatible with 2D or 3D nearest neighbour architectures.

11. Fault-tolerant operations on encoded data

Sections 7, 8 and 10 showed how fault-tolerant QEC allows for any quantum algorithm to be run to arbitrary accuracy. However, the results of the threshold theorem assume that logical operations can be performed directly on the encoded data without the need for continual decoding and re-encoding. Using stabilizer codes, a large class of operations can be performed on logical data in an inherently fault-tolerant way.

If a given logical state, $|\psi\rangle_L$, is stabilized by K , and the logical operation U is applied, the new state, $U|\psi\rangle_L$ is stabilized by UKU^\dagger , i.e.

$$UKU^\dagger U|\psi\rangle_L = UK|\psi\rangle_L = U|\psi\rangle_L. \quad (75)$$

In order for the codeword states to remain valid, the stabilizer set for the code, $\{G_i\} \in \mathcal{G}$, must remain fixed through every operation. Hence for U to be a valid operation on the data, $UG_iU^\dagger = G_j$, $G_j \in \mathcal{G} \forall i$. As a shorthand notation, we express this relationship as $UGU^\dagger = \mathcal{G}$.

11.1. Single qubit operations

For an elementary operation A , the equivalent logical operator \bar{A} will now be formed via a sequence of elementary operations on the physical qubits comprising the code block. The logical \bar{X} and \bar{Z} operations on a single encoded qubit are the first examples of valid codeword operations. Taking the $[[7, 1, 3]]$ code as an example, \bar{X} and \bar{Z} are given by,

$$\begin{aligned}\bar{X} &= XXXXXX \equiv X^{\otimes 7}, \\ \bar{Z} &= ZZZZZZ \equiv Z^{\otimes 7}.\end{aligned}\quad (76)$$

Since the single qubit Pauli operators satisfy $XZX = -Z$ and $ZXZ = -X$ then, $\bar{X}K^i\bar{X} = K^i$ and $\bar{Z}K^i\bar{Z} = K^i$ for each of the $[[7, 1, 3]]$ stabilizers given in equation (50). The fact that each stabilizer has a weight of four guarantees that UKU^\dagger picks up an even number of -1 factors. Since the stabilizers remain fixed, the operations are valid. However, what transformations do equation (76) actually perform on encoded data?

For a single qubit, a bit-flip operation X takes $|0\rangle \leftrightarrow |1\rangle$. Recall that for a single qubit $Z|0\rangle = |0\rangle$ and $Z|1\rangle = -|1\rangle$, hence for \bar{X} to actually induce a logical bit-flip it must take, $|0\rangle_L \leftrightarrow |1\rangle_L$. For the $[[7, 1, 3]]$ code, the final operator which fixes the logical state is $K^7 = Z^{\otimes 7}$, where $K^7|0\rangle_L = |0\rangle_L$ and $K^7|1\rangle_L = -|1\rangle_L$. As $\bar{X}K^7\bar{X} = -K^7$, any state stabilized by K^7 becomes stabilized by $-K^7$ (and vice-versa) after the operation of \bar{X} . Therefore, \bar{X} represents a logical bit flip. The same argument can be used for \bar{Z} by considering the stabilizer properties of the states $|\pm\rangle = (|0\rangle \pm |1\rangle)/\sqrt{2}$. Hence, the logical bit- and phase-flip gates can be applied directly to logical data by simply using seven single qubit X or Z gates (figure 13).

Two other useful gates which can be applied in this manner are the Hadamard rotation and phase gates,

$$H = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}, \quad P = \begin{pmatrix} 1 & 0 \\ 0 & i \end{pmatrix}. \quad (77)$$

These gates are useful since when combined with the two-qubit CNOT gate, they can generate a subgroup of all multi-qubit gates known as the Clifford group (gates which map Pauli group operators back to the Pauli group). Again, using the stabilizers of the $[[7, 1, 3]]$ code and the fact that for single qubits,

$$\begin{aligned}HXH &= Z, & HZH &= X, \\ PX P^\dagger &= iXZ, & PZ P^\dagger &= Z,\end{aligned}\quad (78)$$

a seven qubit bit-wise Hadamard gate will switch X with Z and therefore will simply flip $\{K^1, K^2, K^3\}$ with $\{K^4, K^5, K^6\}$, and is a valid operation. The bit-wise application of the P gate will leave any Z stabilizer invariant, but takes $X \rightarrow iXZ$. This

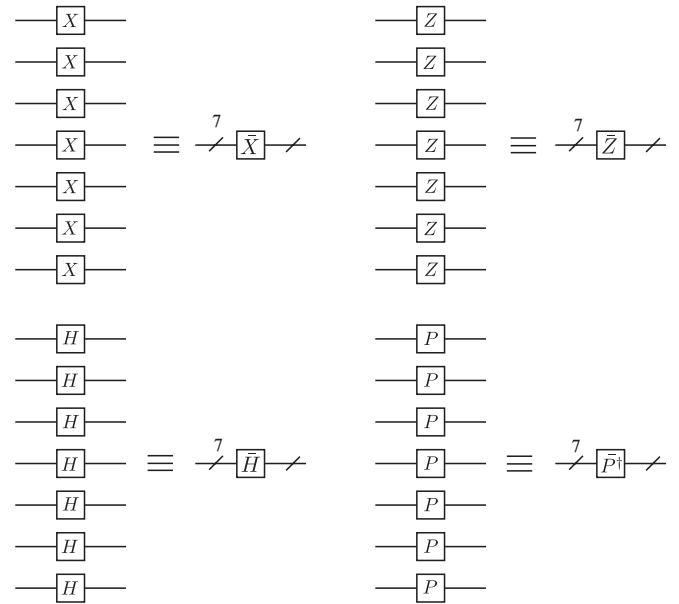


Figure 13. Bit-wise application of single qubit gates in the $[[7, 1, 3]]$ code. Logical X , Z , H and P gates can trivially be applied by using seven single qubit gates, fault-tolerantly. Note that the application of seven P gates results in the logical \bar{P}^\dagger being applied and vice versa.

is still valid, since provided there are a multiple of four non-identity operators for each generator of the stabilizer group, the factors of i will cancel. Hence seven bit-wise P gates is valid for the $[[7, 1, 3]]$ code.

What do these logical operators do to the logical state? For a single qubit, the Hadamard gate flips any Z stabilized state to an X stabilized state, i.e. $|0, 1\rangle \leftrightarrow |+, -\rangle$. Looking at the transformation of K^7 , using $\bar{H} = H^{\otimes 7}$ we have, $\bar{H}K^7\bar{H} = X^{\otimes 7}$. Therefore, the bit-wise Hadamard gate, \bar{H} , is the logical Hadamard operation. The single qubit P gate leaves a Z stabilized state invariant, while an X eigenstate becomes stabilized by iXZ . Hence, denoting $\bar{P}^\dagger = P^{\otimes 7}$, $\bar{P}^\dagger(X^{\otimes 7})\bar{P}^\dagger = -i(XZ)^{\otimes 7}$ and the bit-wise gate, \bar{P}^\dagger , is the logical P^\dagger gate. Similarly, bit-wise gate, $\bar{P} = P^{\dagger \otimes 7}$, enacts a logical P gate (figure 13). Each of these fault-tolerant operations on a logically encoded block are commonly referred to as transversal operations. A transversal operation can be defined as a logical operator which is formed by applying the individual physical operators to each qubit in the code block (or between two equivalent physical qubits in the case of entangling operations such as the CNOT gate shown below). The physical operations forming a transversal logical operator need not be the same, for example in figure 13, the \bar{P}^\dagger gate is formed via individual P operations and vice versa. Hence, valid transversal gates may possibly be constructed from a set of physical operations which are unique to each qubit.

11.2. Two-qubit gate

A two-qubit logical CNOT operation can also be applied in the same way, as a transversal bit-wise operation of individual CNOT gates between corresponding physical qubits. For unencoded qubits, a CNOT operation performs the following

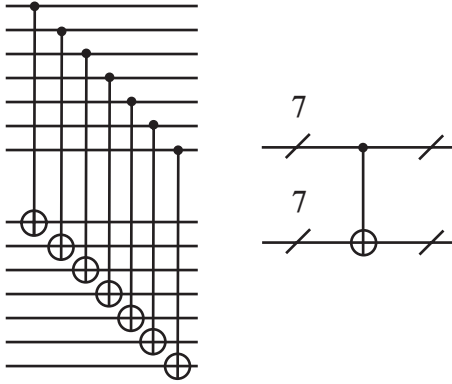


Figure 14. Bit-wise application of a CNOT gate between two logical qubits. Since each CNOT only couples corresponding qubits in each block, this operation is inherently fault-tolerant.

mapping on the two qubit stabilizer set,

$$\begin{aligned}
 X \otimes I &\rightarrow X \otimes X, \\
 I \otimes Z &\rightarrow Z \otimes Z, \\
 Z \otimes I &\rightarrow Z \otimes I, \\
 I \otimes X &\rightarrow I \otimes X,
 \end{aligned} \tag{79}$$

where the first operator corresponds to the control qubit and the second operator corresponds to the target. Now consider the bit-wise application of seven CNOT gates between logically encoded blocks of data (figure 14). First the stabilizer set must remain invariant,

$$\text{CNOT} \mathcal{G} \text{CNOT} = \{\text{CNOT}(K^i \otimes K^j)\text{CNOT}\} = \mathcal{G}, \tag{80}$$

i.e. each element in \mathcal{G} must be mapped to another element of \mathcal{G} . Table 9 details the transformation for all the stabilizer generators under seven bit-wise CNOT gates, demonstrating that this operation is valid on the $[[7, 1, 3]]$ code. The transformations in equation (79) are trivially extended to the logical space, showing that seven bit-wise CNOT gates invoke a logical CNOT operation.

$$\begin{aligned}
 \bar{X} \otimes I &\rightarrow \bar{X} \otimes \bar{X}, \\
 I \otimes \bar{Z} &\rightarrow \bar{Z} \otimes \bar{Z}, \\
 \bar{Z} \otimes I &\rightarrow \bar{Z} \otimes I, \\
 I \otimes \bar{X} &\rightarrow I \otimes \bar{X}.
 \end{aligned} \tag{81}$$

The issue of fault-tolerance with these logical operations should be clear. The $\bar{X}, \bar{Z}, \bar{H}, \bar{P}^\dagger$ and \bar{P} gates are trivially fault-tolerant since the logical operation is performed through seven bit-wise single qubit gates. The logical CNOT is also fault-tolerant since each two-qubit gate only operates between counterpart qubits in each logical block. Hence if any gate is inaccurate then, at most, a single error will be introduced in each block.

In contrast to the $[[7,1,3]]$ code, let us also take a quick look at the $[[5,1,3]]$ code. Unlike the 7-qubit code, the full set of Clifford gates cannot be implemented in the same transversal manner. To see this clearly we can examine how

the generators of the stabilizer group for the code transforms under a transversal Hadamard operation,

$$\begin{aligned}
 K^1 &= X Z Z X I && \xrightarrow{H^{\otimes 5}} && Z X X Z I \\
 K^2 &= I X Z Z X && && I Z X X Z \\
 K^3 &= X I X Z Z && && Z I Z X X \\
 K^4 &= Z X I X Z && && X Z I Z X.
 \end{aligned} \tag{82}$$

The stabilizer group is not preserved under this transformation and therefore is not a valid logical operation for the $[[5,1,3]]$ code. One thing to briefly note is that there are methods for performing logical Hadamard and phase gates on the $[[5,1,3]]$ code [Got97]. However, they essentially involve performing a valid, transversal, three qubit gate and then measuring out two of the logical ancilla. Although it has been demonstrated that a valid set of transversal operations for universal computing is incompatible with a quantum code that corrects an arbitrary single error [EK09], it has not been shown that a non-CSS code cannot have a transversal set of operations generating the Clifford group. This example is used to illustrate the transformation for an invalid logical operation on an encoded state.

While these gates can be conveniently implemented on error protected data, they do not represent a universal set for quantum computation. In fact it has been shown that by using the stabilizer formalism, these operations can be efficiently simulated on a classical device [Got98, AG04]. In order to achieve universality one of the following gates is generally added to the available set,

$$T = \begin{pmatrix} 1 & 0 \\ 0 & e^{i\pi/4} \end{pmatrix}, \tag{83}$$

or the Toffoli gate [Tof81]. Applying them in a similar transversal way, for the majority of stabilizer codes, transform the stabilizers out the group and is consequently not a valid operation. Circuits implementing these two gates in a fault-tolerant manner have been developed [NC00, GC99, SI05, SFH08], for various codes (most often the $[[7, 1, 3]]$ code) but at this stage the circuits are complicated and resource intensive. This has practical implications for encoded operations. If universality is achieved by adding the T gate to the list, arbitrary single qubit rotations require long gate sequences (utilizing the Solovay-Kitaev theorem [Kit97, DN06]) to approximate arbitrary logical qubit rotations and these sequences often require many T gates [Fow05]. Finding more efficient methods to achieve universality on encoded data is therefore still an active area of research.

12. Fault-tolerant circuit design for logical state preparation

Section 10 introduced the basic rules for fault-tolerant circuit design and how these rules lead to the threshold theorem for concatenated error correction. However, what does a full fault-tolerant quantum circuit look like? Here, we introduce a full fault-tolerant circuit to prepare the $[[7, 1, 3]]$ logical $|0\rangle$ state. As the $[[7, 1, 3]]$ code is a single error correcting code, we use the one-to-one definition of fault-tolerance and therefore

Table 9. Transformations of the $[[7, 1, 3]]$ stabilizer generators under the gate operation $U = \text{CNOT}^{\otimes 7}$, where $\mathcal{G} \rightarrow U^\dagger \mathcal{G} U$. Note that the transformation does not take any stabilizer outside the group generated by $K^i \otimes K^j$ ($i, j \in [1, \dots, 6]$), therefore $U = \text{CNOT}^{\otimes 7}$ represents a valid operation on the codespace.

$K^i \otimes K^j$	K^1	K^2	K^3	K^4	K^5	K^6
K^1	$K^1 \otimes I$	$K^1 \otimes K^1 K^2$	$K^1 \otimes K^1 K^3$	$K^1 K^4 \otimes K^1 K^4$	$K^1 K^5 \otimes K^1 K^5$	$K^1 K^6 \otimes K^1 K^6$
K^2	$K^2 \otimes K^1 K^2$	$K^2 \otimes I$	$K^2 \otimes K^2 K^3$	$K^2 K^4 \otimes K^2 K^4$	$K^2 K^5 \otimes K^2 K^5$	$K^2 K^6 \otimes K^2 K^6$
K^3	$K^3 \otimes K^3 K^1$	$K^3 \otimes K^3 K^2$	$K^3 \otimes I$	$K^3 K^4 \otimes K^3 K^4$	$K^3 K^5 \otimes K^3 K^5$	$K^3 K^6 \otimes K^3 K^6$
K^4	$K^4 \otimes K^1$	$K^4 \otimes K^2$	$K^4 \otimes K^3$	$I \otimes K^4$	$K^4 K^5 \otimes K^5$	$K^4 K^6 \otimes K^6$
K^5	$K^5 \otimes K^1$	$K^5 \otimes K^2$	$K^5 \otimes K^3$	$K^5 K^4 \otimes K^4$	$I \otimes K^5$	$K^5 K^6 \otimes K^6$
K^6	$K^6 \otimes K^1$	$K^6 \otimes K^2$	$K^6 \otimes K^3$	$K^6 K^4 \otimes K^4$	$K^6 K^5 \otimes K^5$	$I \otimes K^6$

only need to consider the propagation of a single error during the preparation (any more than one error during correction represents a higher order effect and is ignored).

As described in section 7, logical state preparation can be done by initializing an appropriate number of physical qubits and measuring each of the X stabilizers that describe the code. Therefore, a circuit which allows the measurement of a Hermitian operator in a fault-tolerant manner needs to be constructed. The general structure of the circuit used was first developed by Shor [Sho96], however it should be noted that several more recent methods for fault-tolerant state preparation and correction now exist [Ste97a, Ste02, DA07, Kni05].

The circuits shown in figures 15(a) and (b), which measure the stabilizer $K^1 = IIIXXX$ are not fault-tolerant, since a single ancilla is used to control each of the four CNOT gates. As CNOT gates can copy X errors (figure 12), a single X error on this ancilla can be copied to multiple qubits in the data block. Instead, four ancilla qubits are used which are prepared in the state $|\mathcal{A}\rangle = (|0000\rangle + |1111\rangle)/\sqrt{2}$. This can be done by initializing four qubits in the $|0\rangle$ state and applying a Hadamard, then a sequence of CNOT gates (figure 15(c)). Each of these four ancilla are used to control a separate CNOT gate, after which the ancilla state is decoded and measured. By ensuring that each CNOT is controlled via a separate ancilla, any X error will only propagate to a single qubit in the data block. However, during the preparation of the ancilla state there is the possibility that a single X error can propagate to multiple ancilla, which are then fed forward into the data block. In order to combat this, the ancilla block needs to be verified against possible X errors. Tracking through all the possible locations where a single X error can occur during ancilla preparation leads to the following unique states.

$$\begin{aligned}
|\mathcal{A}\rangle_1 &= \frac{1}{\sqrt{2}}(|0000\rangle + |1111\rangle), \\
|\mathcal{A}\rangle_2 &= \frac{1}{\sqrt{2}}(|0100\rangle + |1011\rangle), \\
|\mathcal{A}\rangle_3 &= \frac{1}{\sqrt{2}}(|0010\rangle + |1101\rangle), \\
|\mathcal{A}\rangle_4 &= \frac{1}{\sqrt{2}}(|0001\rangle + |1110\rangle), \\
|\mathcal{A}\rangle_5 &= \frac{1}{\sqrt{2}}(|0111\rangle + |1000\rangle), \\
|\mathcal{A}\rangle_6 &= \frac{1}{\sqrt{2}}(|0011\rangle + |1100\rangle).
\end{aligned} \tag{84}$$

From these possibilities, the first four states correspond to no error or at most one bit flip and the last two states correspond to multiple errors caused by errors copied by the CNOT operations.

In fact, only the last state, $|\mathcal{A}\rangle_6$, needs to be identified since two bit-flip errors exist on the ancilla state. To verify this ancilla state, a fifth ancilla is added, initialized and used to perform a parity check on the ancilla block. This fifth ancilla is then measured. If the result is $|0\rangle$, the ancilla block can be coupled to the data. If the ancilla result is $|1\rangle$, then either a single error has occurred on the verification qubit or the ancilla state has been prepared in either the $|\mathcal{A}\rangle_5$ or $|\mathcal{A}\rangle_6$ state. In either case, the entire ancilla block is reinitialized and prepared again. This is continued until the verification qubit is measured to be $|0\rangle$ (figure 16). The re-preparation of the ancilla block protects against multiple X errors, which can propagate forward through the CNOT gates. Z errors propagate in the other direction. Any Z error which occurs in the ancilla block will propagate straight through to the final measurement. This results in the measurement not corresponding to the actual errors which have occurred and can result in mis-correction once all stabilizers have been measured. To protect against this, each stabilizer is measured 2–3 times and a majority vote of the measurement results taken. As any additional error represents a second order process, if the first or second measurement has been corrupted by a Z error, then the third measurement will only contain additional errors if a higher order error process has occurred. Therefore, we are free to ignore this possibility and assume that the third measurement is error free. The full circuit for $[[7, 1, 3]]$ state preparation is shown in figure 17, where each stabilizer is measured 2–3 times. The total circuit requires a minimum of 12 qubits (7-data qubits and a 5-qubit ancilla block).

As you can see, the circuit constructions for full fault-tolerant state preparation (and error correction) are not simple circuits. However, they are easy to design in generic ways when employing stabilizer coding.

13. Loss protection

So far we have focused the discussion on correction techniques which assume that error processes maintain a qubit structure to the Hilbert space. As we noted in section 3.3, the loss of physical qubits within the computer violates this assumption and in general requires additional correction machinery beyond what we have already discussed. This section examines a basic correction techniques for qubit loss. Specifically, we detail one such scheme which was developed with single photon based architectures in mind.

Protecting against qubit loss requires a different approach than other general forms of quantum errors

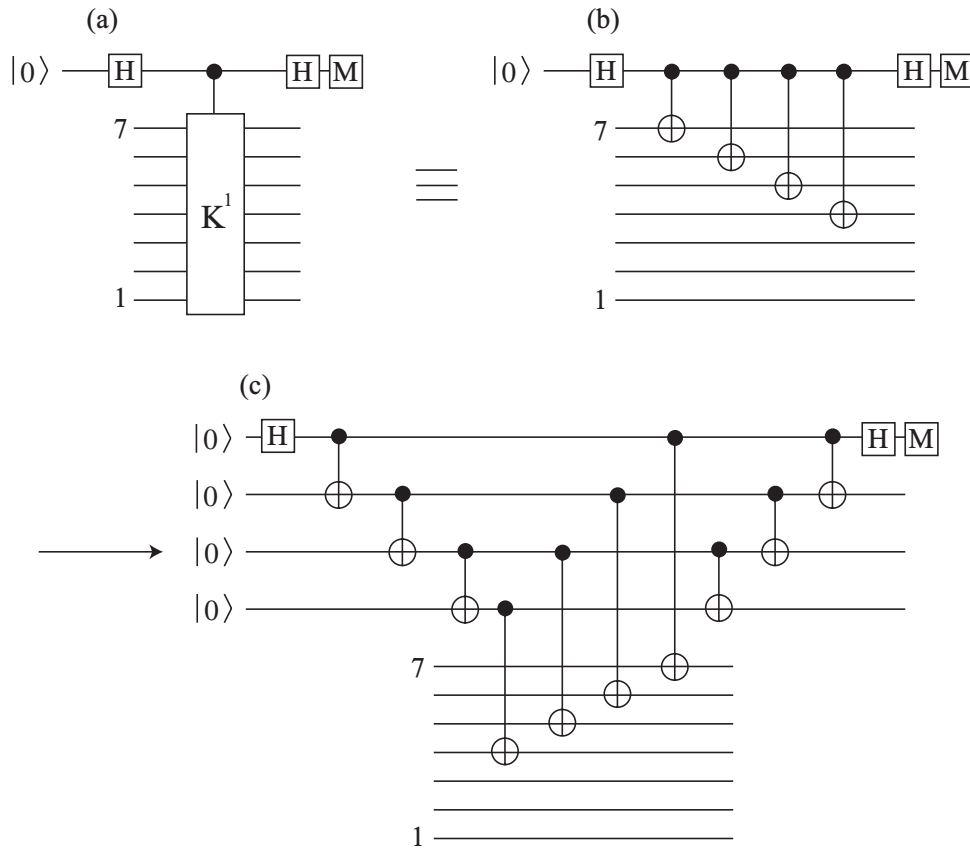


Figure 15. Three circuits which measure the stabilizer K^1 . (a) represents a generic operator measurement where a multi-qubit controlled gate is available. (b) decomposes this into single- and two-qubit gates, but in a non-fault-tolerant manner. (c) introduces four ancilla such that each CNOT is controlled via a separate qubit. This ensures fault-tolerance.

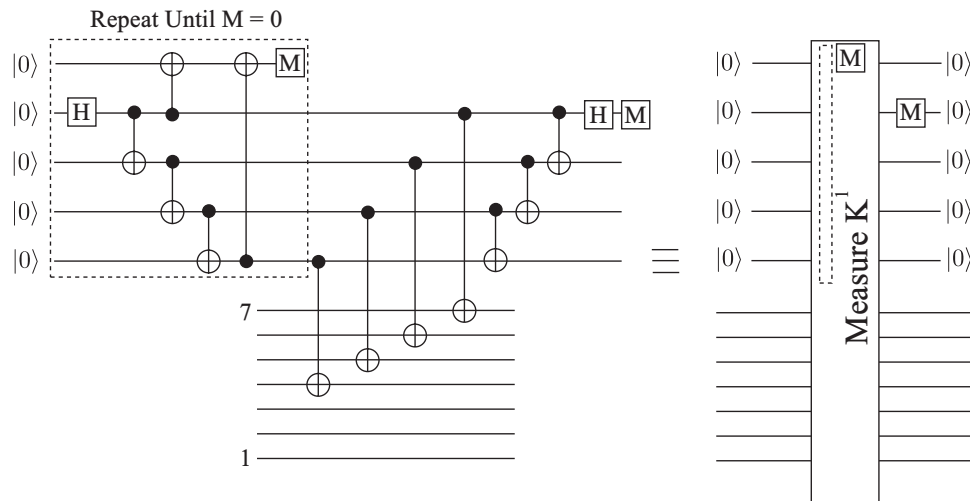


Figure 16. Circuit required to measure the stabilizer K^1 , fault-tolerantly. A four qubit GHZ state is used as ancilla with the state requiring verification against multiple X errors. After the state has passed verification it is coupled to the data block and a syndrome is extracted.

such as environmental decoherence or systematic control imperfections. The cumbersome aspect related to correcting qubit loss is detecting the presence of a qubit at the physical level. The specific machinery that is required for loss detection is dependent on the underlying physical architecture, but the basic principal is that the presence or absence of the physical qubit must be determined without discriminating the actual quantum state.

Certain systems allow for loss detection in a more convenient way than others. Electronic spin qubits, for example, can employ single electron transistors (SETs) to detect the presence or absence of the charge without performing measurement on the spin degree of freedom [DS00, CGJH05, AJW⁺01]. Optics in contrast requires more complicated non-demolition measurement [MW84, IHY85, POW⁺04, MNBS05]. This is due to the fact that typical

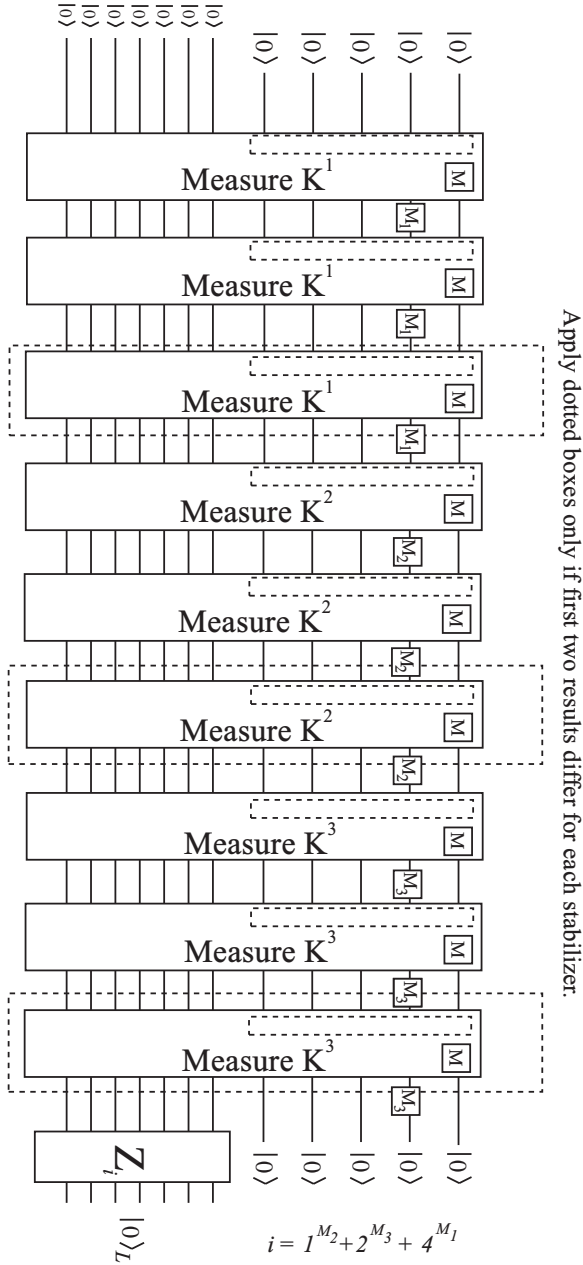


Figure 17. Circuit required to prepare the $[[7, 1, 3]]$ logical $|0\rangle$ state fault-tolerantly. Each of the X stabilizers are sequentially measured using the circuit in figure 16. To maintain fault-tolerance, each stabilizer is measured 2–3 times with a majority vote taken.

photonic measurement is performed via photo-detectors which have the disadvantage of physically destroying the photon.

Once the detection of the presence of the physical qubit has been performed, a freshly initialized qubit can be injected to replace the lost qubit. Once this has been completed, the standard error correcting procedure can correct for the error. A freshly initialized qubit state, $|0\rangle$ can be represented as projective collapse of a general qubit state, $|\psi\rangle \neq |1\rangle$, as

$$|0\rangle \propto |\psi\rangle + Z|\psi\rangle. \quad (85)$$

If we consider this qubit as part of an encoded block, then the above corresponds to a 50% error probability of experiencing

a phase error on this qubit. Therefore, a loss event that is corrected by non-demolition detection and standard QEC, 50% of the time, will result in a detection event in the QEC cycle and, in the absence of other errors, correction. The probability of loss needs to be at a comparable rate to standard errors as the correction cycle after a loss detection event will, with high probability, detect an error.

If a loss event is detected and the qubit replaced, the error detection code shown in section 6 becomes a single error correction code. This is due to the fact that via the identification of the loss event, these errors subsequently have known locations. Consequently error detection is sufficient to perform full correction, in contrast to the error channels we have already considered where in general the location is unknown.

A second method for loss correction is related to systems that have high loss rates compared to systematic and environmental errors, for example in optical systems. Due to the high mobility of single photons and their relative immunity to environmental interactions, loss is a major error channel that generally dominates over other error sources. The use of error detection and correction codes for photon loss is undesirable due to the need for non-demolition detection of the lost qubit. While techniques for measuring the presence or absence of a photon without direct detection have been developed and implemented [POW+04], they require multiple ancilla photons and controlled interactions. Ultimately it may be more desirable to redesign the loss correction code such that it can be employed directly with photo-detection rather than more complicated non-demolition techniques [YNM06].

One such scheme was developed by Ralph, Hayes and Gilchrist in 2005 [RHG05]. This scheme was a more efficient extension of an original parity encoding method developed by Knill, Laflamme and Milburn [KLM01]. The general parity encoding for a logical qubit is an N photon GHZ state in the conjugate basis, i.e.

$$\begin{aligned} |0\rangle_L^N &= \frac{1}{\sqrt{2}}(|+\rangle^{\otimes N} + |-\rangle^{\otimes N}), \\ |1\rangle_L^N &= \frac{1}{\sqrt{2}}(|+\rangle^{\otimes N} - |-\rangle^{\otimes N}), \end{aligned} \quad (86)$$

where $|\pm\rangle = (|0\rangle \pm |1\rangle)/\sqrt{2}$. The motivation with this type of encoding is that measuring any qubit in the $|0, 1\rangle$ basis simply removes it from the state, reducing the resulting state by one,

$$\begin{aligned} P_{0,N} |0\rangle_L^N &= (I_N + Z_N) |0\rangle_L^N \\ &= \frac{1}{\sqrt{2}}(|+\rangle^{N-1} + |-\rangle^{N-1}) |0\rangle_N = |0\rangle_L^{N-1} |0\rangle_N, \\ P_{1,N} |0\rangle_L^N &= (I_N - Z_N) |0\rangle_L^N \\ &= \frac{1}{\sqrt{2}}(|+\rangle^{N-1} - |-\rangle^{N-1}) |1\rangle_N = |1\rangle_L^{N-1} |1\rangle_N, \end{aligned} \quad (87)$$

where $P_{(0,1),N}$ are the projectors corresponding to measurement in the $|0, 1\rangle$ basis on the N -th qubit (up to normalization). The effect for the $|1\rangle_L$ state is similar. Measuring the N -th qubit in the $|0\rangle$ state simply removes it from the encoded state, reducing the logical zero state by one, while measuring the N -th qubit as $|1\rangle$ enacts a logical bit flip at the same time as reducing the size of the logical state.

Instead of introducing the full scheme developed in [RHG05], we give the general idea of how such encoding allows for loss detection without non-demolition measurements. Photon loss in this model is assumed equivalent to measuring the photon in the $|0\rangle, |1\rangle$ basis, but not knowing the answer (section 3.3). Our ignorance of the measurement result could lead to a logical bit flip error on the encoded state, therefore we require the ability to protect against logical bit-flip errors on the above states. As already shown, the 3-qubit code allows us to achieve such correction. Therefore the final step in this scheme is encoding the above states into a redundancy code (a generalized version of the 3-qubit code), where an arbitrary logical state, $|\psi\rangle_L$ is now given by,

$$|\psi\rangle_L = \alpha|0\rangle_1^N |0\rangle_2^N \dots |0\rangle_q^N + \beta|1\rangle_1^N |1\rangle_2^N \dots |1\rangle_q^N \quad (88)$$

where $|0\rangle^N, |1\rangle^N$ are the parity encoded states shown in equation (87) and the fully encoded state is q -blocks of these parity states.

This form of encoding protects against the loss of qubits by first encoding the system into a code structure that allows for the removal of qubits without destroying the computational state and then protecting against logical errors that are induced by loss events. In effect, it maps errors un-correctable by standard QEC to error channels that are correctable, in this case qubit loss \rightarrow qubit bit flip. Even though this code can be used to convert loss events into bit-flips, the size of the code will steadily decrease as qubits are lost.

This general technique is common with pathological error channels. If a specific type of error violates the standard ‘qubit’ assumption of QEC, additional correction techniques are always required to map this type of error to a correctable form, consequently additional physical resources are usually needed.

14. Some modern developments in quantum error correction

Up until this stage we have restricted our discussions on error correction to the most basic principals and codes. The ideas and methodologies we have detailed represent some of the introductory techniques that were developed when error correction was first proposed. For readers who are only looking for a basic introduction to the field, you can quite easily skip the remainder of this paper as we will now examine some of the more modern protocols that are utilized when considering the construction of a large-scale quantum computer.

Providing a fair and encompassing review of the more modern and advanced error correction techniques that have been developed is far outside our goal for this review. However, we would be remiss if we did not briefly examine some of the more advanced error correction techniques that have been proposed for large-scale QIP. For the remainder of this discussion we choose two closely related error correction techniques, subsystem coding and topological coding which have been receiving significant attention in the fields of architecture design and large-scale QIP. While some

readers may disagree, we review these two modern error correction protocols because they are currently two of the most useful correction techniques when discussing the physical construction of a quantum computer.

We again attempt to keep the discussion of these techniques simple and provide specific examples when possible. However, it should be stressed that these error correcting protocols are far more complicated than the basic codes shown earlier. Topological error correction alone has, since its introduction, essentially become its own research topic within the broader error correction field. Hence we encourage the reader who is interested to refer to the cited articles below for more rigorous and detailed treatment of these techniques.

14.1. Subsystem codes: Bacon–Shor codes

Quantum subsystem codes [KLP05, KLPL06, Bac06] are one of the newer and highly flexible techniques to implement quantum error correction. The traditional stabilizer codes that we have reviewed are more formally identified as subspace codes, where information is encoded in a relevant coding subspace of a larger multi-qubit system. In contrast, subsystem coding identifies multiple subspaces of the multi-qubit system as equivalent. Specifically, multiple states are identified with the logical $|0\rangle_L$ and $|1\rangle_L$ states. Rather than review the subsystem codes in general, we will focus on a subset of these codes known as Bacon–Shor codes [Bac06].

The primary benefit to utilizing Bacon–Shor [BS] codes is the general nature of their construction. The description of arbitrarily large error correcting codes is conceptually straightforward, error correction circuits are much simpler to construct [AC07], and the generality of their construction introduces the ability to perform dynamical code switching in a fault-tolerant manner [SEDH08]. This final property gives BS coding significant flexibility as the strength of error correction within a quantum computer can be changed fault-tolerantly during operation of the computer.

As with the other codes presented in this review, BS codes are stabilizer codes but now defined over a square lattice. The lattice dimensions represent the X and Z error correction properties and the size of the lattice in either of these two dimensions dictates the total number of errors the code can correct. In general, a $\mathcal{C}(n_1, n_2)$ BS code is defined over a $n_1 \times n_2$ square lattice which encodes one logical qubit into $n_1 n_2$ physical qubits with the ability to correct at least $\lfloor (n_1 - 1)/2 \rfloor$ Z errors and at least $\lfloor (n_2 - 1)/2 \rfloor$ X errors. Again, keeping with the spirit of this review, we focus on a specific example, the $\mathcal{C}(3,3)$ BS code. This code, encoding one logical qubit with nine physical qubits, can correct for one X and one Z error. In order to define the code structure we begin with a 3×3 lattice of qubits, where qubits are identified with the vertices of the lattice. Note that this 2D structure represents the structure of the code, it does not imply that a physical array of qubits *must* be arranged into a 2D lattice. Figure 18 illustrate three sets of stabilizer operators which are defined over the lattice. The first group, illustrated in figure 18(a) is the stabilizer group, \mathcal{S} , which is generated by the operators,

$$\mathcal{S} = \langle X_{i,*} X_{i+1,*}; Z_{*,j} Z_{*,j+1} \mid i \in \mathbb{Z}_2; j \in \mathbb{Z}_2 \rangle, \quad (89)$$

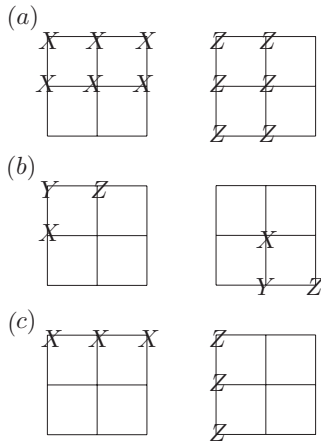


Figure 18. Stabilizer structure for the $\mathcal{C}(3,3)$ code. (a) gives two of the four stabilizers from the group \mathcal{S} . (b) illustrates one of the four encoded sets of Pauli operators from each subsystem defined with the Gauge group, \mathcal{T} . (c) gives the two logical operators from the group \mathcal{L} which enact valid operations on the encoded qubit.

where we retain the notation utilized in [AC07, SEDH08]. $U_{i,*}$ and $U_{*,j}$ represent an operator, U , acting on all qubits in a given row, i , or column, j , respectively, and $\mathbb{Z}_2 = \{1, 2\}$. The second relevant subsystem is known as the gauge group (figure 18(b)), \mathcal{T} , and is described via the non-Abelian group generated by the pairwise operators

$$\mathcal{T} = \langle X_{i,j}X_{i+1,j}; Z_{j,i}Z_{j,i+1} \mid i \in \mathbb{Z}_2; j \in \mathbb{Z}_3 \rangle. \quad (90)$$

The third relevant subsystem is the logical space (figure 18(c)), \mathcal{L} , which can be defined through the logical Pauli operators

$$\mathcal{L} = \langle Z_{*,1}; X_{1,*} \rangle. \quad (91)$$

The stabilizer group \mathcal{S} , defines all relevant code states, i.e. every valid logical space is a +1 eigenvalue of this set. For the $\mathcal{C}(3,3)$ code, there are a total of nine physical qubits and a total of four independent stabilizers in \mathcal{S} , hence there are five degrees of freedom left in the system which can house 2^5 logical states which are simultaneous eigenstates of \mathcal{S} . This is where the gauge group, \mathcal{T} , becomes relevant. As the gauge group is non-Abelian, there is no valid code state which is a simultaneous eigenstate of all operators in \mathcal{T} . However, if you examine closely there are a total of four encoded sets of Pauli operations within \mathcal{T} . Figure 18(b) illustrates two such sets. As all elements of \mathcal{T} commute with all elements of \mathcal{S} we can identify each of these four sets of valid ‘logical’ qubits to be equivalent, i.e. we define $\{|0\rangle_L, |1\rangle_L\}$ pairs which are eigenstates of \mathcal{S} and one Abelian subgroup of \mathcal{T} and then ignore exactly what gauge group we are in¹². Therefore, each of these gauge states represent a subsystem of the code, with each subsystem logically equivalent.

The final group we consider is the logical group \mathcal{L} . This is the set of two Pauli operators which enact a logical X or Z gate on the encoded qubit *regardless* of the gauge choice and

¹² Provided we are in eigenstates of \mathcal{S} we can, in principal, be in superpositions of eigenstates of the operators in \mathcal{T} . However, the error correction procedure for the BS code projects us back to eigenstates of the gauge operators.

consequently represent true logical operations to our encoded space.

In a more formal sense, the definition of these three group structures allows us to decompose the Hilbert space of the system. If we let \mathcal{H} denote the Hilbert space of the physical system, \mathcal{S} forms an Abelian group and hence can act as a stabilizer set defining subspaces of \mathcal{H} . If we describe each of these subspaces by the binary vector, \vec{e} , formed from the eigenvalues of the stabilizers, \mathcal{S} , then each subspace splits into a tensor product structure

$$\mathcal{H} = \bigoplus_{\vec{e}} \mathcal{H}_{\mathcal{T}} \otimes \mathcal{H}_{\mathcal{L}}, \quad (92)$$

where elements of \mathcal{T} act only on the subsystem $\mathcal{H}_{\mathcal{T}}$ and the operators \mathcal{L} act only on the subsystem $\mathcal{H}_{\mathcal{L}}$. In the context of storing qubit information, a logical qubit is encoded into the two dimensional subsystem $\mathcal{H}_{\mathcal{L}}$. As the system is already stabilized by operators in \mathcal{S} and the operators in \mathcal{T} act only on the space $\mathcal{H}_{\mathcal{T}}$, qubit information is only altered when operators in the group \mathcal{L} act on the system.

This formal definition of how BS coding works may be more complicated than the standard stabilizer codes shown earlier, but this slightly more complicated coding structure has significant benefits when we consider how error correction is performed.

In general, to perform error correction, each of the stabilizers of the codespace must be checked to determine which eigenvalue changes have occurred due to errors. The stabilizer group, \mathcal{S} , consists of qubit operators that scale with the size of the code. In general, for a $n_1 \times n_2$ lattice, the X stabilizers are $2n_1$ dimensional and the Z stabilizers are $2n_2$ dimensional. If techniques such as Shor’s method (section 12) were used, we would need to prepare a large ancilla state to perform fault-tolerant correction, and this is clearly undesirable. This problem can be mitigated due to the gauge structure of these codes [AC07].

Each of the stabilizers in \mathcal{S} is simply the product of certain elements from \mathcal{T} , for example,

$$\begin{aligned} X_{1,1}X_{1,2}X_{1,3}X_{2,1}X_{2,2}X_{2,3} &\in \mathcal{S} \\ &= (X_{1,1}X_{2,1}) \cdot (X_{1,2}X_{2,2}) \cdot (X_{1,3}X_{2,3}) \in \mathcal{T}. \end{aligned} \quad (93)$$

Therefore if we check the eigenvalues of the three, 2-qubit operators from \mathcal{T} we are able to calculate what the eigenvalue is for the six-dimensional stabilizer. This decomposition of the stabilizer set for the code can only occur since the decomposition is in terms of operators from \mathcal{T} which, when measured, has no effect on the logical information encoded within the system. In fact, when error correction is performed, the gauge state of the system will almost always change based on the order in which the eigenvalues of the gauge operators are checked.

This exploitation of the gauge properties of subsystem coding is extremely beneficial for the design of fault-tolerant correction circuits. As the stabilizer operators can now be decomposed into multiple two-dimensional operators, fault-tolerant circuits for error correction do not require any encoded ancilla states. Furthermore, even if we decide to scale the

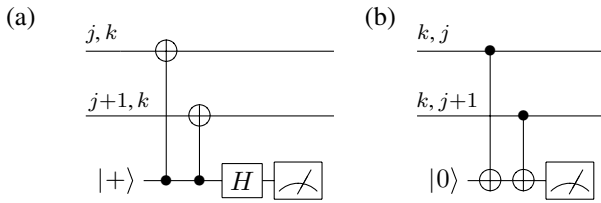


Figure 19. (From [AC07]) Circuits for measuring the gauge operators and hence performing error correction for subsystem codes. (a) measures, fault-tolerantly, the operator $X_{j,k}X_{j+1,k}$ with only one ancilla. (b) measures $Z_{k,j}Z_{k,j+1}$. The results of these two qubit parity checks can be used to calculate the parity of the higher dimensional stabilizer operators of the code.

code-space to correct more errors (increasing the lattice size representing the code), we do not require measuring operators with higher dimensionality. Figure 19 taken from [AC07] illustrates the fault-tolerant circuit constructions for Bacon–Shor subsystem codes. As each ancilla qubit is only coupled to two data qubits, no further circuit constructions are required to ensure fault-tolerance¹³. The measurement results from these two-dimensional parity checks are then combined to calculate the parity of the higher dimensional stabilizer of the subsystem code.

A second benefit to utilizing subsystem codes is the ability to construct fault-tolerant circuits to perform dynamical code switching. Dynamical code switching is a technique that could be utilized when noise sources are highly biased. In many systems the physical mechanisms which lead to errors are not symmetric in X and Z . For example, simple Markovian dephasing introduces only Z errors into the system and in all circumstances is the more dominant error channel.

One technique is to use concatenation in a clever way. The QEC codes we have largely examined in this review are symmetric in the X and Z sector. The 5-, 7- and 9-qubit codes correct for *one* X error and *one* Z error. Aliferis and Preskill considered a concatenated code, where the lower level code is a simple n qubit repetition code (section 4) that only corrected for Z errors [AP08]. By utilizing this lower level code, they symmetrize the Z and X noise (the size of the n qubit code is determined by the level of asymmetry in the physical noise). The subsequent levels are standard QEC codes, but it now operates at the next encoded level which will operate with symmetrical rates for logical X and Z errors. By utilising the simpler n -qubit repetition code at the lower level, qubit resources are reduced, compared to using a symmetric code at every level. This work was extended and adapted to a realistic model in superconducting systems [ABD⁺09].

Asymmetric coding may also be required if the noise acting on a quantum computer changes over the course of its operation. In this case it may be required to switch from a symmetric code to an asymmetric code, dynamically. Naively, this can be done by simply decoding all qubits and then re-encoding them with the new code. However, this procedure would not be fault-tolerant, as any error that occurs

¹³ It should be noted that for higher distance BS codes, specifically for $d \geq 5$, multiple measurements for each syndrome are required to ensure fault-tolerance.

to the system when it is decoded will propagate through to the new encoded state. A second technique would be to encode the system with an asymmetric code at the next level of concatenation and then decode the lower level. However, when utilizing the BS codes, we can perform this code switching in a different way. As noted before, the $\mathcal{C}(n_1, n_2)$ code corrects for $\lfloor (n_1 - 1)/2 \rfloor$ X errors and $\lfloor (n_1 - 1)/2 \rfloor$ Z errors. The ability to convert between two codes, $\mathcal{C}(n_1, n_2)$ and $\mathcal{C}(n'_1, n'_2)$, in a fault-tolerant manner would allow for us to dynamically change the strength (or asymmetry) of the error correction whenever noise rates fluctuate in the computer. It was shown in [SEDH08] how such switching can be achieved, allowing for the fault-tolerant conversion between arbitrary sized QEC codes.

14.2. Topological codes

A conceptually similar (but technically distinct) coding technique to the Bacon–Shor subsystem codes is the idea of topological error correction, first introduced with the Toric code of Kitaev in 1997 [Kit97]. Topological coding is similar to subsystem codes in that the code structure is defined on a lattice (which, in general, can be of dimension ≥ 2) and the scaling of the code to correct more errors is conceptually straightforward. However, in topological coding schemes the protection afforded to logical information relies on the unlikely application of error chains which define non-trivial topological paths over the code surface.

Topological error correction is a complicated area of QEC and fault-tolerance and any attempt to fairly summarize the field is not possible within this review. In brief, there are two ways of approaching these schemes. The first is simply to treat topological codes as a class of stabilizer codes over a qubit system. This approach is more amenable to current information technologies and is being adapted to methods in cluster state computing [RHG07, FG09], optics [DFS⁺09, DMN11], ion-traps [SJ09] and superconducting systems [IFI⁺02]. The second approach is to construct a physical Hamiltonian model based on the structure of the topological code or choose systems which appear to exhibit topological order. This leads to the more complicated field on anyonic quantum computation [Kit97]. For example, the coding structure of the Toric code [Kit97] can be treated as a physical Hamiltonian system, with the ground state corresponding to the logical states of the code (in the case of the Torus it is a 4-fold degenerate ground state corresponding to two encoded qubits). Excitations from the ground state of this Hamiltonian correspond to the creation of anyonic quasi-particles and are energetically unfavourable (since their physical Hamiltonian symmetries reflect the coding structure imposed).

This and other models of anyonic computation utilise quasi-particles that exhibit fractional quantum statistics (they acquire fractional phase shifts when their positions are exchanged twice with other anyons, in contrast to bosons or fermions which always acquire ± 1 phase shifts). The unique properties of anyonic systems therefore allow for natural, robust, error protection. However, the major issue with this

model is that it relies on quasi-particle excitations that do not, in general, arise naturally. Although certain physical systems have been shown to exhibit anyonic properties, most notably in the fractional quantum Hall effect [NSS⁺08]. However, it is a daunting task to both manufacture a reliable anyonic system and to reliably design and construct a large-scale computing system. Note: Recently it has been shown that anyonic models based on the 2D codes illustrated in this section do not exhibit self-correcting properties [KC08, BT09]. Higher dimensional topological coding models, currently a minimum of 4D, are actually required to implement self-correcting quantum memories [AHHH10].

As there are several extremely good discussions of both anyonic [NSS⁺08] and non-anyonic topological computing [DKLP02, FSG09, FG09], we will not review any of the anyonic methods for topological computing and simply provide a brief example of one topological coding scheme, namely the surface code [BK01, DKLP02, FSG09]. The surface code for QEC is a desirable error correction model for several reasons. As it is defined over a two-dimensional lattice of qubits it can be implemented on architectures that only allow for the coupling of nearest neighbour qubits (rather than the arbitrary long distance coupling of qubits in separate regions of the computer). The surface code also exhibits one of the highest fault-tolerant thresholds of any QEC scheme; recent simulations estimate a threshold approaching 1% [RHG07, WFSH10]. Finally, the surface code has been analysed with respect to loss errors, showing a high tolerance when such errors are heralded [SBD09]. This subfield of error correction has been heavily researched in recent years. Methods in statistical physics are now routinely utilized to help calculate fault-tolerant thresholds [BMD08, KBMD09, BAO⁺12, ABKMD12], more advanced coding models and methods for computing with these models are under investigation [BMD07b, BMD07a, KBAMD10, Bom11, Fow12]. This section will present a basic introduction, hopefully readers will feel more comfortable studying these advanced techniques afterwards.

The surface code, as with subsystem codes, is a stabilizer code defined over a two-dimensional qubit lattice, as figure 20 illustrates. We identify each edge of the 2D lattice with a physical qubit. The stabilizer set consists of two types of operators. The first is the set of $Z^{\otimes 4}$ operators which circle every lattice face (or plaquette). The second is the set of $X^{\otimes 4}$ operators which encircle every vertex of the lattice. The stabilizer set is consequently generated by the operators,

$$A_p = \bigotimes_{j \in b(p)} Z_j, \quad B_v = \bigotimes_{j \in s(v)} X_j, \quad (94)$$

where $b(p)$ is the four qubits surrounding a plaquette and $s(v)$ is the four qubits surrounding each vertex in the lattice and identity operators on the other qubits are implied. First note that all of these operators commute as any plaquette and vertex stabilizer will share either zero or two qubits. If the lattice is not periodic in either dimension, this stabilizer set completely specifies one unique state, i.e. for a $N \times N$ lattice there are $2N^2$ qubits and $2N^2$ stabilizer generators. Hence this stabilizer set defines a unique multi-qubit entangled state

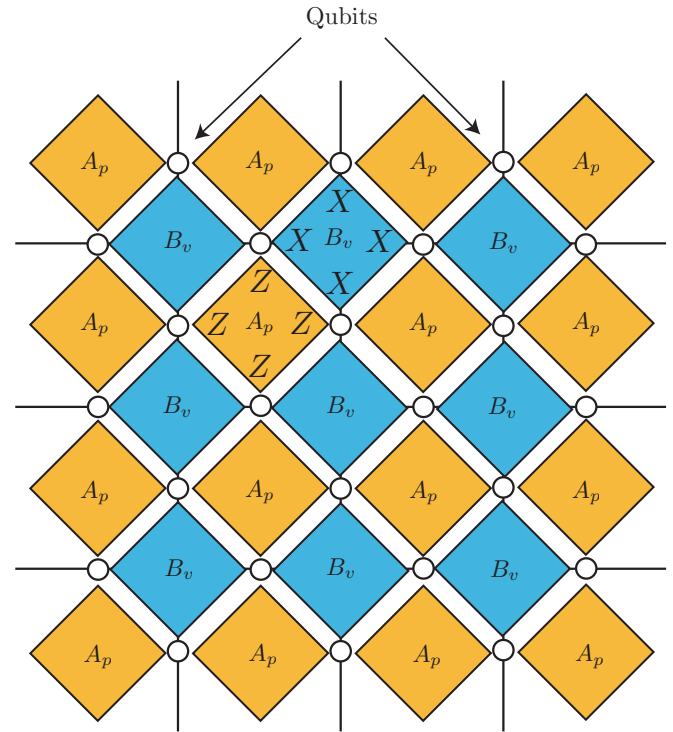


Figure 20. General structure of the surface code. The edges of the lattice correspond to physical qubits. The four qubits surrounding each face (or plaquette) are +1 eigenstates of the operators A_p while the four qubits surrounding each vertex are +1 eigenstates of the operators B_v . If all eigenstate conditions are met, a unique multi-qubit state is defined as a ‘clean’ surface.

which is generally referred to as a ‘clean’ surface. Detailing exactly how this surface can be utilized to perform robust quantum computation is far outside the scope of this review and there are several papers to which such a discussion can be referred [RH07, RHG07, FSG09, FG09]. Instead, we can quite adequately show how robust error correction is possible by simply examining how a ‘clean’ surface can be maintained in the presence of errors. The X and Z stabilizer sets, A_p and B_v , define two equivalent 2D lattices which are interlaced, as figure 21 illustrates. If the total 2D lattice is shifted along the diagonal by half a cell then the operators B_v are now arranged around a plaquette and the operators A_p are arranged around a lattice vertex. Since protection against X errors is achieved by detecting eigenvalue flips of Z stabilizers and vice versa, these two interlaced lattices correspond to error correction against X and Z errors respectively. Therefore we can quite happily restrict our discussion to one possible error channel, for example correcting X errors (since the correction for Z errors proceeds identically when considering the stabilizers B_v instead of A_p).

Figure 22(a) illustrates the effect a single X error has on a pair of adjacent plaquettes. Since X and Z anti-commute, a single bit-flip error on one qubit in the surface will flip the eigenvalue of the $Z^{\otimes 4}$ stabilizers on the two plaquettes adjacent to the respective qubit. As single qubit errors act to flip the eigenvalue of adjacent plaquette stabilizers we examine how chains of errors affect the surface. Figures 22(b) and (c) examine two longer chains of errors. As you can see, if multiple errors occur, only the eigenvalues of the stabilizers

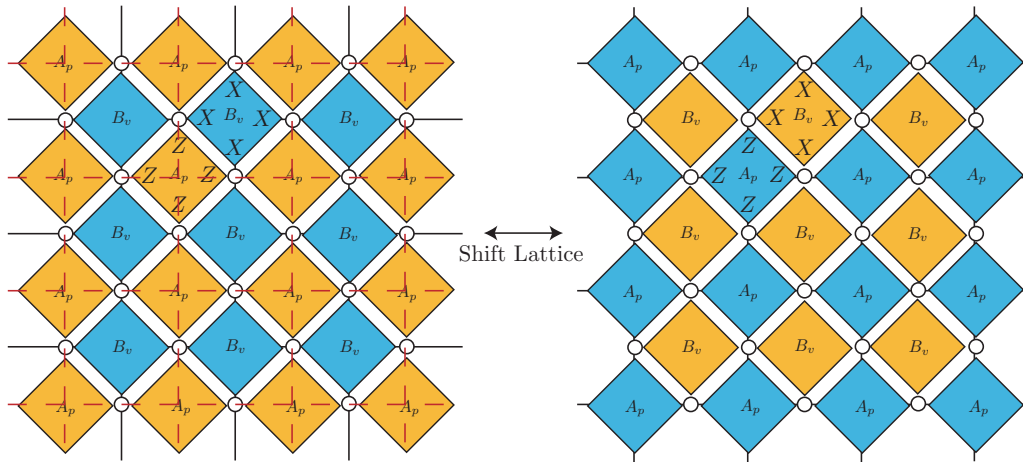


Figure 21. The surface code imbeds two self similar lattices that are interlaced, generally referred to as the primal and dual lattice. (a) illustrates one lattice where plaquettes are defined with the stabilizers A_p . (b) illustrates the dual structure where plaquettes are now defined by the stabilizer set B_v . The two lattice structures are interlaced and are related by shifting along the diagonal by half a lattice cell. Each of these equivalent lattices are independently responsible for X and Z error correction.

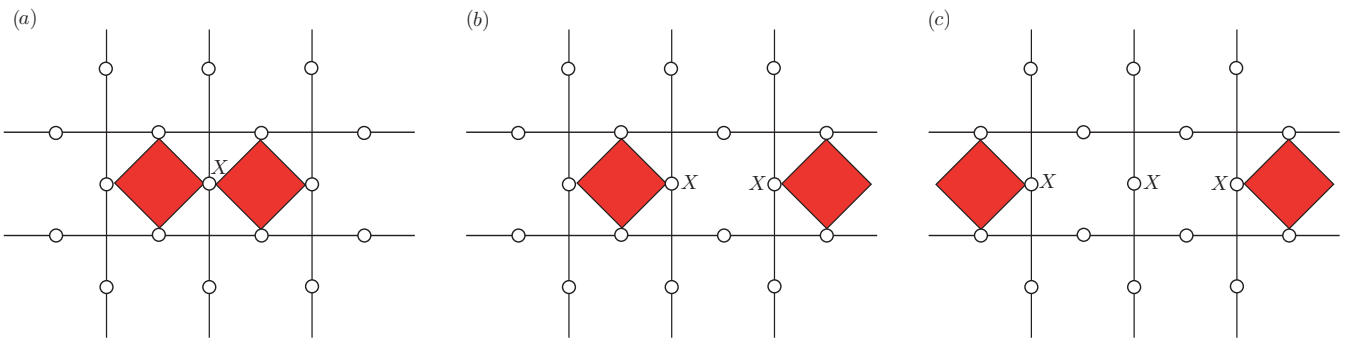


Figure 22. Examples of error chains and their effect on the eigenvalues for each plaquette stabilizer. (a): a single X error causes the parity of two adjacent cells to flip. (b) and (c): longer chains of errors only cause the end cells to flip eigenvalue as each intermediate cell will have two X errors and hence the eigenvalue for the stabilizer will flip twice.

associated with the ends of the error chains flip. Each plaquette along the chain will always have two X errors occurring on different boundaries and consequently the eigenvalue of the $Z^{\otimes 4}$ stabilizer around these plaquettes will flip twice.

If we now consider an additional ancilla qubit which sits in the centre of each plaquette and can couple to the four surrounding qubits, we can check the parity by running the simple circuit shown in figure 23. If we assume that we initially prepare a perfect ‘clean’ surface we then, at some later time, check the parity of every plaquette over the surface. If X errors have occurred on a certain subset of qubits, the parity associated with the endpoints of error chains will have flipped. We now take this two-dimensional classical data tree of eigenvalue flips and pair them up into the most likely set of error chains. Since it is assumed that the probability of error on any individual qubit is low, the most likely set of errors which reflects the eigenvalue changes observed is the minimum weight set (i.e. connect up all plaquettes where eigenvalues have changed into pairs such that the total length of all connections is minimized). This classical data processing is quite common in computer science and minimum weight matching algorithms such as the Blossom package [CR99, Kol09] have a running time polynomial in the total number of data points in the classical set. Once this

minimal matching is achieved, we can identify the likely error chains corresponding to the end points and correction can be applied accordingly.

The failure of this code is therefore dictated by error chains that cannot be detected through changes in plaquette eigenvalues. If you examine figure 24, we consider an error chain that connects one edge of the surface lattice to another. In this case every plaquette has two associated qubits that have experienced a bit flip and no eigenvalues in the surface have changed. Since we have assumed that we are only wishing to maintain a ‘clean’ surface, these error chains have no effect, but when one considers the case of storing information in the lattice, these types of error chains correspond to logical errors on the qubit [BK98, FSG09]. Hence undetectable errors are chains which connect boundaries of the surface to other boundaries (in the case of information processing, qubits are artificial boundaries within the larger lattice surface). It should be stressed that this is a simplified description of the full protocol, but it does encapsulate the basic idea. The important thing to realize is that the failure rate of the error correction procedure is suppressed, exponentially with the size of the lattice. If we consider an error model where each qubit experiences a bit flip, independently, with probability p , then an error chain of one occurs with probability p , error

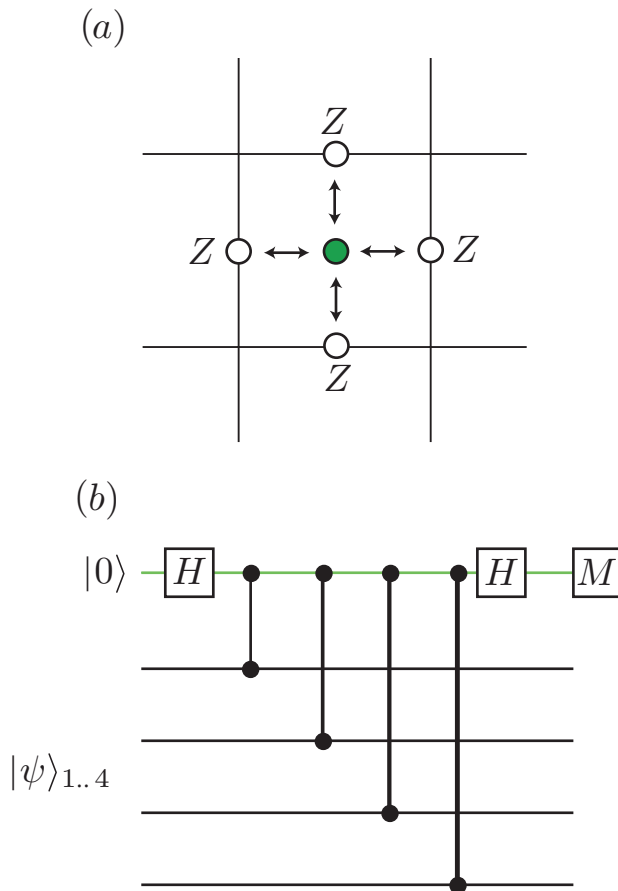


Figure 23. (a) Lattice structure to check the parity of a surface plaquette. An additional ancilla qubit is coupled to the four neighbouring qubits that comprise each plaquette. (b) Quantum circuit to check the parity of the $Z^{\otimes 4}$ stabilizer for each surface plaquette.

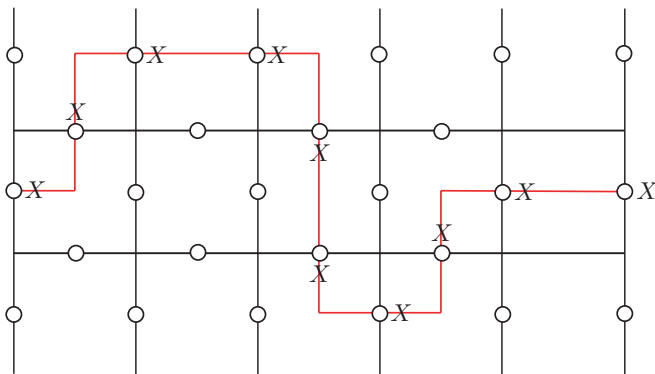


Figure 24. Example of a chain of errors which do not cause any eigenvalue changes in the surface. If errors connect boundaries to other boundaries, the error correction protocol will not detect them. In the case of a ‘clean’ surface, these error chains are invariants of the surface code. When computation is considered, qubit information are artificial boundaries within the surface. Hence if error chains connect these information qubits to other boundaries, logical errors occur.

chains of weight two occur with probability $O(p^2)$, chains of three $O(p^3)$, etc. If we have an $N \times N$ lattice and we extend the surface by *one* plaquette in each dimension, then the probability of having an error chain connecting two boundaries

will drop by a factor of p (one extra qubit has to experience an error)¹⁴. Extending an $N \times N$ lattice by one plaquette in each dimension requires $O(N)$ extra qubits, hence this type of error correcting code suppresses the probability of having undetectable errors exponentially with a qubit resource cost which grows linearly.

As we showed in section 10, standard concatenated coding techniques allow for an error rate suppression which scales with the concatenation level as a double exponential while the resource increase scales exponentially. For the surface code, the error rate suppression scales exponentially while the resource increase scales linearly. While these scaling relations might be mathematically equivalent, the surface code offers much more flexibility at the architectural level. Being able to increase the error protection in the computer with only a linear change in the number of physical qubits is far more beneficial than using an exponential increase in resources when utilizing concatenated correction. Specifically, consider the case where an error protected computer is operating at a logical error rate which is just above what is required for an algorithm. If concatenated error correction is employed, then adding another layer of correction will not only increase the number of qubits by an exponential amount, but it will also drop the effective logical error rate far below what is actually required. In contrast, if surface codes are employed, we increase the qubit resources by a linear factor and drop the logical error rate sufficiently for successful application of the algorithm.

We now leave the discussion regarding topological correction models. We emphasize again that this was a *very* broad overview of the general concept of topological codes. There are many details and subtleties that we have deliberately left out of this discussion and we urge the reader, if interested, to refer to the referenced articles for a more thorough treatment of this topic.

15. Conclusions and future outlook

This review has hopefully provided a basic introduction to some of the most important theoretical aspects of QEC and fault-tolerant quantum computation. The ultimate goal of this discussion was not to provide a rigorous theoretical framework for QEC and fault-tolerance, but instead attempted to illustrate most of the important rules, results and techniques that have evolved out of this field. Hopefully this introduction will serve as a starting point for individuals introducing themselves to this topic. For those wishing to continue research into the QEC field, we highly recommend consulting the papers referenced throughout this review and especially references [Got97, NC00, Got02, KLA⁺02, Ste01, Got09] which provide a more mathematically rigorous review of QEC and fault-tolerance.

We have not only covered the basic aspects of QEC through specific examples, but also briefly discussed how physical errors influence quantum computation and how these processes are interpreted within the context of QEC. One of

¹⁴ The reduction is not exactly p as there is a combinatorial factor that accounts for all the extra error chains that are now possible.

the more important aspects of this review is the discussion related to the stabilizer formalism, circuit synthesis and fault-tolerant circuit construction. Stabilizers are arguably the most useful theoretical formalism in QEC as once it is sufficiently understood, most of the important properties of error correcting codes can be investigated and understood largely by inspection.

The study of QEC and fault-tolerance is still an active area of QIP research. Although the library of quantum codes and error correction techniques is vast there is still a significant disconnect between the abstract framework of quantum coding and the more physically realistic implementation of error correction for large-scale quantum information processing.

There are several future possibilities for the direction of quantum information processing. Even with the development of many of these advanced techniques, the physical construction and accuracy of current qubit fabrication is still insufficient to obtain any benefit from QEC. Many in the field now acknowledge that the future development of quantum computation will most likely split into two broad categories. The first is arguably the more physically realistic, namely small qubit applications in quantum simulation. Beyond these smaller qubit applications, we move to truly large-scale quantum computation, i.e. implementing large algorithms such as Shor on qubit arrays well beyond 1000 physical qubits. This would undoubtedly require active techniques in error correction. Future work needs to focus on adapting the many codes and fault-tolerant techniques to the architectural level. As we noted in section 10.4, the implementation of QEC at the design level largely influences the fault-tolerant threshold exhibited by the code itself. Being able to efficiently incorporate both the actual quantum code and the error correction procedures at the physical level is extremely important when developing an experimentally viable, large-scale quantum computer.

There are many differing opinions within the quantum computing community as to the future prospects for quantum information processing. Many remain pessimistic regarding the development of a million qubit device and instead look towards quantum simulation in the absence of active error correction as the realistic goal of quantum information. However, in the past few years, the theoretical advances in error correction and the fantastic speed in the experimental development of few qubit devices continues to offer hope for the near-term construction of a large-scale computer, incorporating many of the ideas presented within this review. While we could never foresee the possible successes or failures in quantum information science, we remain hopeful that a large-scale quantum computer is still a goal worth pursuing.

Acknowledgments

The authors wish to thank A M Stephens, R Van Meter, A G Fowler, L C L Hollenberg and A D Greentree for helpful comments and acknowledge the support of MEXT, JST and FIRST projects.

References

- [ABD⁺09] Aliferis P, Brito F, DiVincenzo D P, Preskill J, Steffen M and Terhal B M 2009 Fault-tolerant computing with biased-noise superconducting qubits *New J. Phys.* **11** 013061
- [ABKMD12] Andrist R S, Bombin H, Katzgraber H G and Martin-Delgado M A 2012 Optimal error correction in topological subsystem codes *Phys. Rev. A* **85** 050302(R)
- [ABO97] Aharonov D and Ben-Or M 1997 Fault-tolerant quantum computation with constant error *Proc. 29th Annual ACM Symp. on Theory of Computing* (New York, NY) p 46
- [ABO08] Aharonov D and Ben-Or M 2008 Fault-tolerant quantum computation with constant error rate *SIAM J. Comput.* **38** 1207–82
- [AC07] Aliferis P and Cross A W 2007 Subsystem fault tolerance with the Bacon–Shor code *Phys. Rev. Lett.* **98** 220502
- [AG04] Aaronson S and Gottesman D 2004 Improved simulation of stabilizer circuits *Phys. Rev. A* **70** 052328
- [AGP08] Aliferis P, Gottesman D and Preskill J 2008 Accuracy threshold for post selected quantum computation *Quantum Inform. Comput.* **8** 181–244
- [AHHH10] Alicki R, Horodecki M, Horodecki P and Horodecki R 2010 On thermal stability of topological qubit in Kitaev’s 4D model *Open Syst. Inform. Dyn.* **17** 1
- [AJW⁺01] Aassime A, Johansson G, Wendin G, Schoelkopf R J and Delsing P 2001 Radio-frequency single-electron transistor as readout device for qubits: charge sensitivity and backaction *Phys. Rev. Lett.* **86** 3376
- [Ali07] Aliferis P 2007 Level reduction and the quantum threshold theorem *PhD Thesis* Caltech
- [ALKH02] Ahn D, Lee J, Kim M S and Hwang S W 2002 Self-consistent non-Markovian theory of a quantum-state evolution for quantum-information-processing *Phys. Rev. A* **66** 012302
- [AP08] Aliferis P and Preskill J 2008 Fault-tolerant quantum computation against biased noise *Phys. Rev. A* **78** 052331
- [APN⁺05] Astafiev O, Pashkin Yu A, Nakamura Y, Yamamoto T and Tsai J S 2005 Quantum noise in the Josephson charge qubit *Phys. Rev. Lett.* **93** 267007
- [ATK⁺09] Aoki T, Takahashi G, Kajiyama T, Yoshikawa J, Braunstein S L, van Loock P and Furusawa A 2009 Quantum error correction beyond qubits *Nature Phys.* **5** 541
- [Avi87] Avizienis A 1987 *The Evolution of Fault-Tolerant Computing* (New York: Springer)
- [Bac06] Bacon D 2006 Operator quantum error-correcting subsystems for self-correcting quantum memories *Phys. Rev. A* **73** 012340
- [BAO⁺12] Bombin H, Andrist R S, Ohzeki M, Katzgraber H G and Martin-Delgado M A 2012 Strong resilience of topological codes to depolarization *Phys. Rev. X* **2** 021004
- [BBK03] Briar A, Benjamin S C and Kay A 2003 Quantum error correction in globally controlled arrays (arXiv:quant-ph/0308113)
- [BHC04] Brown K R, Harrow A W and Chuang I L 2004 Arbitrarily accurate composite pulse sequences *Phys. Rev. A* **70** 052318
- [BHPC03] Boulant N, Havel T F, Pravia M A and Cory D G 2003 Robust method for estimating the Lindblad

- operators of a dissipative quantum process from measurements of the density operator at multiple time points *Phys. Rev. A* **67** 042322
- [BK98] Bravyi S and Kitaev A 1998 Quantum codes on a lattice with boundary (arXiv:quant-ph/9811052)
- [BK01] Bravyi S B and Kitaev A Y 2001 Quantum codes on a lattice with boundary *Quantum Comput. Comput.* **2** 43
- [BKD04] Burkard G, Koch R H and DiVincenzo D P 2004 Multilevel quantum description of decoherence in superconducting qubits *Phys. Rev. B* **69** 064503
- [BKSO05] Balensiefer S, Kregor-Stickles L and Oskin M 2005 An evaluation framework and instruction set architecture for ion-trap based quantum micro-architectures *SIGARCH Comput. Archit. News* **33** 186
- [BLWZ05] Byrd M S, Lidar D A, Wu L-A and Zanardi P 2005 Universal leakage elimination *Phys. Rev. A* **71** 052301
- [BM03] Barrett S D and Milburn G J 2003 Measuring the decoherence rate in a semiconductor charge qubit *Phys. Rev. B* **68** 155307
- [BMD06] Bombin H and Martin-Delgado M A 2006 Topological quantum distillation *Phys. Rev. Lett.* **97** 180501
- [BMD07a] Bombin H and Martin-Delgado M A 2007 Optimal resources for topological 2D stabilizer codes: comparative study *Phys. Rev. A* **76** 012305
- [BMD07b] Bombin H and Martin-Delgado M A 2007 Topological computation without braiding *Phys. Rev. Lett.* **98** 160502
- [BMD08] Bombin H and Martin-Delgado M A 2008 Statistical mechanical models and topological color codes *Phys. Rev. A* **77** 042322
- [Bom11] Bombin H 2011 Clifford gates by code deformation *New. J. Phys.* **13** 043005
- [BR01] Briegel H-J and Raussendorf R 2001 Persistent entanglement in arrays of interacting particles *Phys. Rev. Lett.* **86** 910
- [Bra98] Braunstein S L 1998 Error correction for continuous quantum variables *Phys. Rev. Lett.* **80** 4084
- [BSW96] Bennett C H, DiVincenzo D P, Smolin J A and Wootters W K 1996 Mixed state entanglement and quantum error correction *Phys. Rev. A* **54** 3824
- [BT09] Bravyi S and Terhal B M 2009 A no-go theorem for a two-dimensional self-correcting quantum memory based on stabilizer codes *New. J. Phys.* **11** 043029
- [CG97] Cleve R and Gottesman D 1997 Efficient computations of encodings for quantum error correction *Phys. Rev. A* **56** 76
- [CGJH05] Conrad V I, Greentree A D, Jamieson D N and Hollenberg L C L 2005 Analysis and geometric optimization of single electron transistors for read-out in solid-state quantum computing *J. Comput. Theor. Nanosci.* **2** 214
- [CR99] Cook W and Rohe A 1999 Computing minimum-weight perfect matchings *INFORMS J. Comput.* **11** 138
- [CRSS98] Calderbank A R, Rains E M, Shor P W and Sloane N J A 1998 Quantum error correction via codes over GF(4) *IEEE Trans. Inform. Theory* **44** 1369
- [CS96] Calderbank A R and Shor P W 1996 Good quantum error-correcting codes exist *Phys. Rev. A* **54** 1098
- [CSG+05] Cole J H, Schirmer S G, Greentree A D, Wellard C J, Oi D K L and Hollenberg L C L 2005 Identifying an experimental two-state Hamiltonian to arbitrary accuracy *Phys. Rev. A* **71** 062312
- [DA07] DiVincenzo D P and Aliferis P 2007 Effective fault-tolerant quantum computation with slow measurement *Phys. Rev. Lett.* **98** 020501
- [DFS+09] Devitt S J, Fowler A G, Stephens A M, Greentree A D, Hollenberg L C L, Munro W J and Nemoto K 2009 Architectural design for a topological cluster state quantum computer *New. J. Phys.* **11** 083032
- [DG97] Duan L-M and Guo G-C 1997 Preserving coherence in quantum computation by pairing quantum bits *Phys. Rev. Lett.* **79** 1953
- [DG98a] Duan L-M and Guo G-C 1998 Prevention of dissipation with two particles *Phys. Rev. A* **57** 2399
- [DG98b] Duan L-M and Guo G-C 1998 Reducing decoherence in quantum computer-memory with all quantum bits coupling to the same environment *Phys. Rev. A* **57** 737
- [DKLP02] Dennis E, Kitaev A, Landahl A and Preskill J 2002 Topological quantum memory *J. Math. Phys.* **43** 4452
- [DMN11] Devitt S J, Munro W J and Nemoto K 2011 High performance quantum computing *Prog. Inform.* **8** 1–7
- [DN06] Dawson C M and Nielsen M A 2006 The Solovay-Kitaev algorithm *Quantum Inform. Comput.* **6** 81
- [DS96] DiVincenzo D P and Shor P W 1996 Fault-tolerant error correction with efficient quantum codes *Phys. Rev. Lett.* **77** 3260
- [DS00] Devoret M H and Schoelkopf R J 2000 Amplifying quantum signals with the single-electron transistor *Nature* **406** 1039
- [DSO+07] Devitt S J, Schirmer S G, Oi D K L, Cole J H and Hollenberg L C L 2007 Subspace confinement: how good is your qubit? *New. J. Phys.* **9** 384
- [DWM03] Daffer S, Wódkiewicz K and McIver J K 2003 Quantum Markov channels for qubits *Phys. Rev. A* **67** 062312
- [EJ96] Ekert A and Jozsa R 1996 Quantum computation and Shor's factoring algorithm *Rev. Mod. Phys.* **68** 733
- [EK09] Eastin B and Knill E 2009 Restrictions on transversal encoded quantum gate sets *Phys. Rev. Lett.* **102** 110502
- [FG09] Fowler A G and KGoyal A G 2009 Topological cluster state quantum computing *Quantum Inform. Comput.* **9** 721
- [FLP04] Facchi P, Lidar D A and Pascazio S 2004 Unification of dynamical decoupling and the quantum Zeno effect *Phys. Rev. A* **69** 032314
- [Fow05] Fowler A G 2005 Towards large scale quantum computation *PhD Thesis* Melbourne (arXiv:quant-ph/0506126)
- [Fow12] Fowler A G 2012 Low overhead surface code Logical H *Quantum Inform. Comput.* **12** 970
- [FSG09] Fowler A G, Stephens A M and Groszkowski P 2009 High threshold universal quantum computation on the surface code *Phys. Rev. A* **80** 052312
- [FTY+07] Fowler A G, Thompson W F, Yan Z, Stephens A M, Plourde B L T and Wilhelm F K 2007 Long-range coupling and scalable architecture for superconducting flux qubits *Phys. Rev. B* **76** 174507
- [G83] Gács P 1983 Reliable computation with cellular automata *Proc. ACM Symp. Theory of Computing (Orlando, FL)* vol 32 pp 17–78
- [Gar91] Gardiner C W 1991 *Quantum Noise (Springer Series in Synergetics vol 56)* (Berlin: Springer)
- [GBP97] Grassl M, Beth Th and Pellizzari T 1997 Codes for the quantum erasure channel *Phys. Rev. A* **56** 33

- [GC99] Gottesman D and Chuang I L 1999 Demonstrating the viability of universal quantum computation using teleportation and single qubit operations *Nature* **402** 390
- [GHSZ90] Greenberger D M, Horne M A, Shimony A and Zeilinger A 1990 Bell's theorem without inequalities *Am. J. Phys.* **58** 1131
- [GHZ89] Greenberger D M, Horne M A and Zeilinger A 1989 *Bell's theorem, Quantum, and Conceptions of the Universe* (Dordrecht: Kluwer)
- [Got96] Gottesman D 1996 A class of quantum error-correcting codes saturating the quantum hamming bound *Phys. Rev. A* **54** 1862
- [Got97] Gottesman D 1997 Stabilizer codes and quantum error correction *PhD Thesis* Caltech (arXiv:quant-ph/9705052)
- [Got98] Gottesman D 1998 A theory of fault-tolerant quantum computation *Phys. Rev. A* **57** 127
- [Got00] Gottesman D 2000 Fault-tolerant quantum computation with local gates *J. Mod. Opt.* **47** 333
- [Got02] Gottesman D 2002 An introduction to quantum error correction *Quantum Computation: A Grand Mathematical Challenge for the Twenty-First Century and the Millennium* ed S J Lomonaco Jr pp 221–35 (Providence, RI: American Mathematical Society) (arXiv:quant-ph/0004072)
- [Got09] Gottesman D 2009 An introduction to quantum error correction and fault-tolerant quantum computation (arXiv:0904.2557)
- [HMCS00] Hope J J, Moy G M, Collett M J and Savage C M 2000 Steady-state quantum statistics of a non-Markovian atom laser *Phys. Rev. A* **61** 023603
- [IFI+02] Ioffe L B, Feigel'man M V, Ioselevich A, Ivanov D, Troyer M and Blatter G 2002 Topologically protected quantum bits from Josephson junction arrays *Nature* **415** 503
- [IHY85] Imoto N, Haus H A and Yamamoto Y 1985 Quantum nondemolition measurement of the photon number via the optical Kerr effect *Phys. Rev. A* **32** 2287
- [JFS06] Jordan S P, Farhi E and Shor P W 2006 Error correcting codes for adiabatic quantum computation *Phys. Rev. A* **74** 052322
- [JMF+12] Cody Jones N, Van Meter R, Fowler A G, McMahon P L, Kim J, Ladd T D and Yamamoto Y 2012 A layered architecture for quantum computing using quantum dots *Phys. Rev. X* **2** 031007
- [Jon09] Jones J A 2009 Composite pulses in NMR quantum computation *J. Indian Inst. Sci.* **89** 303
- [Kan98] Kane B E 1998 A silicon-based nuclear spin quantum computer *Nature* **393** 133
- [KBAMD10] Katzgraber H G, Bombin H, Andrist R S and Martin-Delgado M A 2010 Topological color codes on Union Jack lattices: a stable implementation of the whole Clifford group *Phys. Rev. A* **81** 012319
- [KBMD09] Katzgraber H G, Bombin H and Martin-Delgado M A 2009 Error threshold for color codes and random 3-body Ising models *Phys. Rev. Lett.* **103** 090501
- [KC08] Kay A and Colbeck R 2008 Quantum self-correcting stabilizer codes (arxiv:0810.3557)
- [Kit97] Kitaev A Y 1997 Quantum computations: algorithms and error correction *Russ. Math. Surv.* **52** 1191
- [KL97] Knill E and Laflamme R 1997 Theory of quantum error-correcting codes *Phys. Lett. A* **55** 900
- [KL00] Knill E and Laflamme R 2000 A theory of quantum error-correcting codes *Phys. Rev. Lett.* **84** 2525
- [KL05] Khodjasteh K and Lidar D A 2005 Fault-tolerant quantum dynamical decoupling *Phys. Rev. Lett.* **95** 180501
- [KLA+02] Knill E, Laflamme R, Ashikhmin A, Barnum H, Viola L and Zurek W H 2002 Introduction to quantum error correction *Los Alamos Sci.* **27** 188
- [KLM01] Knill E, Laflamme R and Milburn G J 2001 A scheme for efficient quantum computation with linear optics *Nature* **409** 46
- [KLP05] Kribs D, Laflamme R and Poulin D 2005 Unified and generalized approach to quantum error correction *Phys. Rev. Lett.* **94** 180501
- [KLPL06] Kribs D, Laflamme R, Poulin D and Lesosky M 2006 Operator quantum error correction *Quantum Inform. Comput.* **6** 383
- [KLV00] Knill E, Laflamme R and Viola L 2000 Theory of quantum error correction for general noise *Phys. Rev. Lett.* **84** 2525
- [KLZ96] Knill E, Laflamme R and Zurek W H 1996 Accuracy threshold for quantum computation (arXiv:quant-ph/9610011)
- [KLZ98] Knill E, Laflamme R and Zurek W H 1998 Resilient quantum computation *Science* **279** 342
- [Kni96] Knill E 1996 Non-binary unitary error bases and quantum codes *LANL report LAUR-96-2717* (arXiv:quant-ph/9608048)
- [Kni05] Knill E 2005 Quantum computing with realistically noisy devices *Nature* **434** 39
- [Kol09] Kolmogorov V 2009 Blossom V: A new implementation of a minimum cost perfect matching algorithm *Math. Prog. Comput.* **1** 43
- [LCW98] Lidar D A, Chuang I L and Whaley K B 1998 Decoherence-free subspaces for quantum computation *Phys. Rev. Lett.* **81** 2594
- [Lev86] Levitt M H 1986 Composite pulses *Prog. NMR Spectros.* **18** 61
- [LF79] Levitt M H and Freeman R 1979 NMR population inversion using a composite pulse *J. Magn. Reson.* **33** 473
- [LMPZ96] Laflamme R, Miquel C, Paz J P and Zurek W H 1996 Perfect quantum error correcting code *Phys. Rev. Lett.* **77** 198
- [LS98] Lloyd S and Slotine J E 1998 Analog quantum error correction *Phys. Rev. Lett.* **80** 4088
- [LW03] Lidar D A and Whaley K B 2003 Decoherence-free subspaces and subsystems *Irreversible Quantum Dynamics (Springer Lecture Notes in Physics vol 622)* ed F Benatti and R Floreanini (Berlin: Springer) pp 83–120 (arXiv:quant-ph/0301032)
- [MCM+05] Martinis J M *et al* 2005 Decoherence in Josephson qubits for dielectric loss *Phys. Rev. Lett.* **95** 210503
- [MCT+04] Metodiev T, Cross A, Thaker D, Brown K, Copsey D, Chong F T and Chuang I L 2004 Preliminary results on simulating a scalable fault-tolerant ion trap system for quantum computation *3rd Workshop on Non-Silicon Computing (NSC-3) (Urbana, IL)* www.csif.cs.ucdavis.edu/~metodiev/papers/NSC3-setso.pdf
- [MLFY10] Van Meter R, Ladd T D, Fowler A G and Yamamoto Y 2010 Distributed quantum computation architecture using semiconductor nanophotonics *Int. J. Quantum Inform.* **8** 295
- [MNBS05] Munro W J, Nemoto K, Beausoleil R G and Spiller T P 2005 High-efficiency quantum-nondemolition single-photon-number-resolving detector *Phys. Rev. A* **71** 033819
- [MW84] Milburn G J and Walls D F 1984 State reduction in quantum-counting quantum nondemolition measurements *Phys. Rev. A* **30** 56
- [NC00] Nielsen M A and Chuang I L 2000 *Quantum Computation and Information* 2nd edn (Cambridge: Cambridge University Press)

- [Neu55] Von Neumann J 1955 Probabilistic logics and the synthesis of reliable organisms from unreliable components *Automata Stud.* **34** 43
- [NSS+08] Nayak C, Simon S H, Stern A, Freedman M and Das Sarma S 2008 Non-Abelian anyons and topological quantum computation. *Rev. Mod. Phys.* **80** 1083
- [PCZ97] Poyatos J F, Cirac J I and Zoller P 1997 Complete characterization of a quantum process: the two-bit quantum gate *Phys. Rev. Lett.* **78** 390
- [POW+04] Pryde G J, O'Brien J L, White A G, Bartlett S D and Ralph T C 2004 Measuring a photonic qubit without destroying it *Phys. Rev. Lett.* **92** 190402
- [PR12] Paetznick A and Reichardt B W 2012 Fault-tolerant ancilla preparation and noise threshold lower bound for the 23-qubit Golay code *Quantum Inform. Comput.* **12** 1034
- [Pre98] Preskill J 1998 Introduction to quantum computation *Introduction to Quantum Computation* ed H-K Lo *et al* (Singapore: World Scientific) pp 213–70 (arXiv:quant-ph/9712048)
- [PSE96] Massimo Palma G, Suominen K-A and Ekert A K 1996 Quantum computers and dissipation *Proc. R. Soc. Lond. A* **452** 567–84
- [PVK97] Plenio M B, Vedral V and Knight P L 1997 Quantum error correction in the presence of spontaneous emission *Phys. Rev. A* **55** 67
- [RB01] Raussendorf R and Briegel H-J 2001 A one way quantum computer *Phys. Rev. Lett.* **86** 5188
- [RH07] Raussendorf R and Harrington J 2007 Fault-tolerant quantum computation with high threshold in two dimensions *Phys. Rev. Lett.* **98** 190504
- [RHG05] Ralph T C, Hayes A J F and Gilchrist A 2005 Loss tolerant optical qubits *Phys. Rev. Lett.* **95** 100501
- [RHG07] Raussendorf R, Harrington J and Goyal K 2007 Topological fault-tolerance in cluster state quantum computation *New J. Phys.* **9** 199
- [SBD09] Stace T M, Barrett S D and Doherty A C 2009 Thresholds for topological codes in the presence of loss *Phys. Rev. Lett.* **102** 200501
- [SBF+06] Skzopek T, Boykin P O, Fan H, Roychowdhury V P, Yablonoitch E, Simms G, Gyure M and Fong B 2006 Threshold error penalty for fault-tolerant computation with nearest neighbour communication *IEEE Trans. Nanotechnol.* **5** 42
- [SDT07] Svore K M, DiVincenzo D P and Terhal B M 2007 Noise threshold for a fault-tolerant two-dimensional lattice architecture *Quantum Inform. Comput.* **7** 297
- [SEDH08] Stephens A, Evans Z W E, Devitt S J and Hollenberg L C L 2008 Subsystem code conversion *Phys. Rev. A* **77** 062335
- [SFH08] Stephens A, Fowler A G and Hollenberg L C L 2008 Universal fault-tolerant computation on bilinear nearest neighbor arrays *Quantum Inform. Comput.* **8** 330
- [Sho95] Shor P W 1995 Scheme for reducing decoherence in quantum computer memory *Phys. Rev. A* **52** R2493
- [Sho96] Shor P W 1996 Fault-tolerant quantum computation *Proc. 37th IEEE Symp. on Foundations of Computer Science (Burlington, VT)* pp 56–65
- [Sho97] Shor P W 1997 Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer *SIAM J. Sci. Statist. Comput.* **26** 1484
- [SI05] Steane A M and Ibinson B 2005 Fault-tolerant logical gate networks for Calderbank–Shor–Steane codes *Phys. Rev. A* **72** 052335
- [SJ09] Stock R and James D F V 2009 A scalable, high-speed measurement based quantum computer using trapped ions *Phys. Rev. Lett.* **102** 170501
- [ST00] Steinbach J and Twamley J 2000 Motional quantum error correction *J. Mod. Opt.* **47** 453
- [Ste96a] Steane A M 1996 Error correcting codes in quantum theory *Phys. Rev. Lett.* **77** 793
- [Ste96b] Steane A M 1996 Multiple particle interference and quantum error correction *Proc. R. Soc. Lond. A* **452** 2551
- [Ste96c] Steane A M 1996 Simple quantum error-correcting codes *Phys. Rev. A* **54** 4741
- [Ste97a] Steane A M 1997 Active stabilization, quantum computation and quantum state synthesis *Phys. Rev. Lett.* **78** 2252
- [Ste97b] Steane A M 1997 The ion trap quantum information processor *Appl. Phys. B* **64** 623
- [Ste01] Steane A M 2001 Quantum computing and error correction *Decoherence and its Implications in Quantum Computation and Information Transfer* ed Gonis and Turchi (Amsterdam: IOS Press) pp 284–98 (arXiv:quant-ph/0304016)
- [Ste02] Steane A M 2002 Fast fault-tolerant filtering of quantum codewords (arXiv:quant-ph/0202036)
- [Tof81] Toffoli T 1981 Bicontinuous extension of reversible combinatorial functions *Math. Syst. Theory* **14** 13–23
- [Uhr07] Uhrig G S 2007 Keeping a quantum bit alive by optimized π -pulse sequences *Phys. Rev. Lett.* **98** 100504
- [Unr95] Unruh W G 1995 Maintaining coherence in quantum computers *Phys. Rev. A* **51** 992
- [VK03] Viola L and Knill E 2003 Robust dynamical decoupling of quantum systems with bounded controls *Phys. Rev. Lett.* **90** 037901
- [VK05] Viola L and Knill E 2005 Random decoupling schemes for quantum dynamical control and error suppression *Phys. Rev. Lett.* **94** 060502
- [VKL99] Viola L, Knill E and Lloyd S 1999 Dynamical decoupling of open quantum systems *Phys. Rev. Lett.* **82** 2417
- [VL98] Viola L and Lloyd S 1998 Dynamical suppression of decoherence in two-state quantum systems *Phys. Rev. A* **58** 2733
- [vL08] van Loock P 2008 A note on quantum error correction with continuous variables (arXiv:0811.3616)
- [VT99] Vitali D and Tombesi P 1999 Using parity kicks for decoherence control *Phys. Rev. A* **59** 4178
- [VWW05] Vala J, Whaley K B and Weiss D S 2005 Quantum error correction of a qubit loss in an addressable atomic system *Phys. Rev. A* **72** 052318
- [WBL02] Wu L-A, Byrd M S and Lidar D A 2002 Efficient universal leakage elimination for physical and encoded qubits *Phys. Rev. Lett.* **89** 127901
- [WFSH10] Wang D S, Fowler A G, Stephens A M and Hollenberg L C L 2010 Threshold error rates for the toric and surface codes *Quantum Inform. Comput.* **10** 456
- [WZ82] Wootters W K and Zurek W H 1982 A single quantum cannot be cloned *Nature* **299** 802
- [YJG+12] Yao N Y, Jiang L, Gorshkov A V, Maurer P C, Giedke G, Cirac J I and Lukin M D 2012 Scalable

- architecture for a room temperature solid-state quantum information processor *Nature Commun.* **3** 800
- [YNM06] Yamaguchi F, Nemoto K and Munro W J 2006 Quantum error correction via robust probe modes *Phys. Rev. A* **73** 060302
- [Zan99] Zanardi P 1999 Symmetrizing evolutions *Phys. Lett. A* **258** 77
- [ZR97a] Zanardi P and Rasetti M 1997 Error avoiding quantum codes *Mod. Phys. Lett. B* **11** 1085
- [ZR97b] Zanardi P and Rasetti M 1997 Noisless quantum codes *Phys. Rev. Lett.* **79** 3306