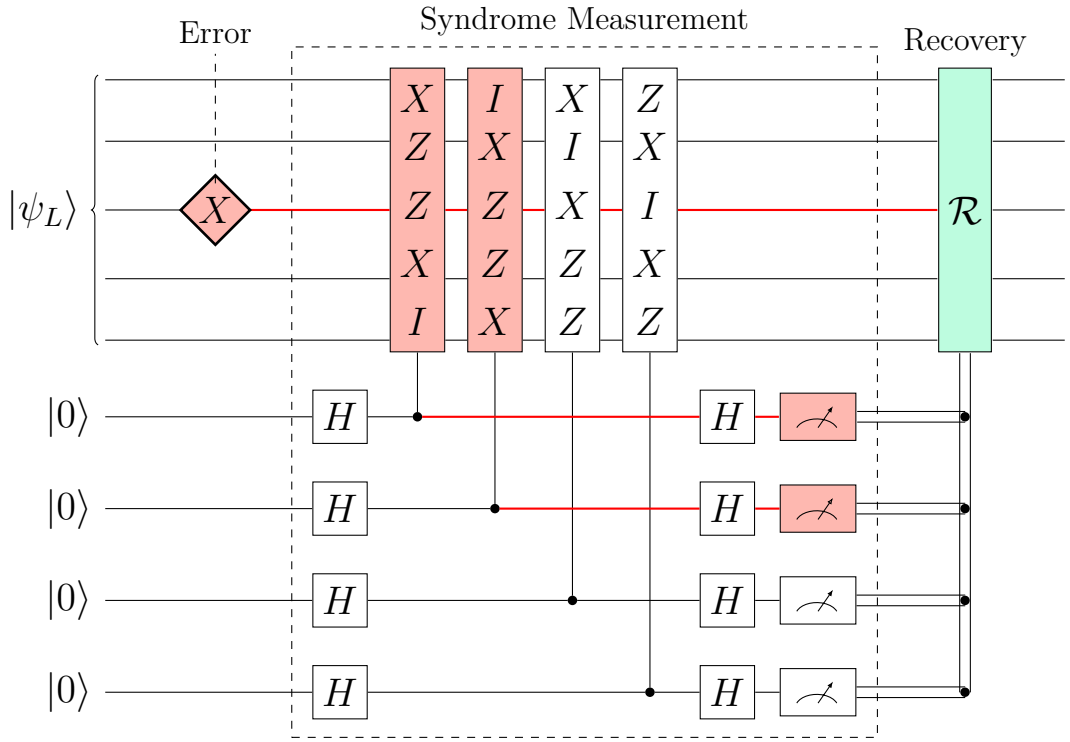
CHALMERS
UNIVERSITY OF TECHNOLOGY



Simulating quantum error correction in a small stabilizer code

Simulating the $[\![5, 1, 3]\!]$ stabilizer code, the smallest possible capable of correcting single qubit errors, under physical device limitations and noise

Master's thesis in Physics

ALEXANDER ANDERSSON AND SEBASTIAN HOLMIN

DEPARTMENT OF PHYSICS
CHALMERS UNIVERSITY OF TECHNOLOGY
Gothenburg, Sweden 2021
www.chalmers.se

Master's thesis 2021

Simulating quantum error correction in a small stabilizer code

Simulating the $[\![5, 1, 3]\!]$ stabilizer code, the smallest possible capable of correcting single qubit errors, under physical device limitations and noise

ALEXANDER ANDERSSON AND SEBASTIAN HOLMIN

CHALMERS
UNIVERSITY OF TECHNOLOGY

Department of physics
Chalmers University of Technology
Gothenburg, Sweden 2021

Simulating quantum error correction in a small stabilizer code
Simulating the $[\![5, 1, 3]\!]$ stabilizer code, the smallest possible capable of correcting single qubit errors, under physical device limitations and noise
ALEXANDER ANDERSSON AND SEBASTIAN HOLMIN

Supervisors: Mats Granath and Anton Frisk Kockum
Examiner: Mats Granath

Master's Thesis 2021
Department of Physics
Chalmers University of Technology
SE-412 96 Gothenburg
Telephone +46 31 772 1000

Cover: Circuit diagram of the quantum program for performing quantum error correction with the $[\![5, 1, 3]\!]$ stabilizer code. The 5-qubit state $|\psi_L\rangle$ redundantly encodes the information of a single qubit. The red, diamond-shaped, X operator represents an error, and the red-colored wires and gates after it represent its propagation through the circuit. The encoded state is entangled with ancillary qubits which are then measured to identify the error. The measurement outcomes are classically processed into a unique recovery operation $\mathcal{R}$ to restore the state to $|\psi_L\rangle$.

Typeset in LaTeX
Printed by Chalmers Reproservice
Gothenburg, Sweden 2021

Simulating quantum error correction in a small stabilizer code
Simulating the $[\![5, 1, 3]\!]$ stabilizer code, the smallest possible capable of correcting single qubit errors, under physical device limitations and noise
ALEXANDER ANDERSSON AND SEBASTIAN HOLMIN
Department of physics
Chalmers University of Technology

Abstract

Quantum error correction is an essential precondition for scaling up the current noisy intermediate-scale quantum (NISQ) computing of today. However, only rudimentary parts of quantum error correction have so far been experimentally demonstrated. Specifically, active correction of errors in distance-3 or above error correcting codes, capable of identifying arbitrary single qubit errors, has so far not been realized. In this work, we simulate repeated rounds of error correction with incoherent noise using the $[\![5, 1, 3]\!]$ error-correcting code, which is the smallest possible fulfilling this requirement, with the aim of determining its viability on a 7-qubit device being designed at the Wallenberg Center for Quantum Technology (WACQT) at Chalmers University of Technology.

We find that the error-correcting code suffers from a large number of incorrectly decoded errors caused by noise during the detection step, which we identify as the primary bottleneck. The lifetime of the logical qubit can be improved by introducing a delay between the cycles of error correction, and by splitting the error measurements over two ancillary qubits. Using the lifetime parameters $T_1 = 40$ µs and $T_2 = 60$ µs, representing the decay of the $|1\rangle$ and $|+\rangle$ states for the constituent single qubits respectively, we reach a logical qubit lifetime of $T_L = 39$ µs, which is narrowly below the break-even point. For cases where $T_2 \gg T_1$ (including $T_1 = 40$ µs and $T_2 = 70$ µs), we find that the lifetime of the logical qubit may exceed the single qubits when comparing the worst case initial conditions. The stronger condition of $T_L > T_1, T_2$ is reached at $T_1 = T_2 = 120$ µs. A more robust error-decoding scheme, perhaps from improved parallelism or fault-tolerant stabilizer circuits, would significantly lower the barrier to the error-correction break-even point and make the code more effective.

Keywords: Quantum computing, QEC, Quantum error correction, Stabilizer code, [[5,1,3]], 5-qubit code, distance-3, Qiskit, Simulation, Quantum mechanics.

Acknowledgements

We would like to extend our deepest gratitude to our supervisors Mats Granath and Anton Frisk Kockum as well as Basudha Srivastava for giving us invaluable input and advice during the entire project. We also thank Chris Warren for providing insight into the experimental constraints and noise present in quantum computers, both in general and for the devices here at Walleberg Centre of Quantum technology (WACQT). Finally we would also like to thank the entirety of WACQT for their interest in our project, giving us the opportunity to both present and discuss our results with them on several occasions.

Alexander Andersson & Sebastian Holmin, Gothenburg, June 2021

Contents

List of Figures

List of Tables

1

Introduction

Quantum computers have the potential to significantly outperform classical computers in many cases such as Shor's algorithm for integer factorization [1] and computational chemistry problems [2][3]. These quantum algorithms would offer exponential speedup to their classical counterpart, presenting new opportunities in their respective fields. However, despite the many theoretical applications, there are currently practical hurdles preventing quantum computers of sufficient size from being realized. One key obstacle is the very large susceptibility to noise due to the fragility of the quantum bits (qubits). Future large-scale quantum computers will employ quantum error correction (QEC) to suppress the influence of noise to allow for scalable controlled computation. Quantum error correcting codes (QECC) employ multiple physical qubits to encode the data of fewer, *logical* qubits. Such a system could be more tolerant to noise, and will usually have both the ability to detect the occurrence of certain classes of errors and even correct them. QECCs are typically very sensitive to the quality of the constituent qubits, as the errors produced during the detection and correction routines must be less frequent than those being corrected to gain any net benefit. As of now, in the current noisy intermediate-scale quantum (NISQ) era where qubits are both noisy and limited in number [4], only the most rudimentary aspects of QEC have been experimentally demonstrated [5][6][7][8][9].

Currently, Wallenberg Center for Quantum Technology (WACQT) at Chalmers University of Technology, is making preparations to build a chip of seven transmon qubits. It is planned to feature high connectivity and high-quality qubits, which introduces the opportunity to experimentally examine more sophisticated QECCs than what has been achieved so far.

In this master thesis project, we will investigate the feasibility of one such QECC, the $[\![5, 1, 3]\!]$ code, in relation to different architectures and qubit fidelities of a seven-qubit chip. The $[\![5, 1, 3]\!]$ stabilizer code is the smallest known QECC with the ability to not only detect the presence of any single-qubit error but to uniquely identify them, such that they can be corrected. Previous experiments [5][6], using codes lacking this feature, have demonstrated logically encoded qubits with higher stability than their constituent qubits through post-selecting out runs where errors were detected.

We propose circuits compatible with the device limitations of the planned transmon architecture, and compare the performance of coupling the qubits in a square lattice to a triangular one, as well as the usage of one or two ancillary qubits. The simulation

is done using the quantum computing development platform Qiskit [10], developed by IBM. We extract the noise parameters required to reach the "break-even" threshold, where the QECC extends the lifetime of the logical qubit above that of a single physical qubit.

1.1 Outline of report

We begin this thesis by briefly explaining the core concepts of quantum computing (QC), followed by an overview of general stabilizer QECCs and how they relate to the $[\![5, 1, 3]\!]$ code. We then proceed with a description of how to model errors using quantum channels, including a summary of the ones we employ for our simulations.

In chapter 3, we give an overview of the physical limitations of present-day quantum computers and how these can be overcome to practically realize the $[\![5, 1, 3]\!]$ code. We also specify the procedures employed for the simulations, and what device parameters are used to mimic realistic conditions.

In chapter 4, we present the compiled circuits and overview their efficiency. We simulate the encoding of logical states, as well as repeated rounds of QEC. We investigate the device parameters needed to reach the QEC break-even point, and whether the triangular chip architecture is better suited to implement the $[\![5, 1, 3]\!]$ code than others with lesser connectivity.

We conclude the thesis with a recap of the core obstacles for experimentally realizing the code, and present prospects for how these could be circumvented to make the scheme viable for experimental demonstration of error correction in future works.

1.2 Limitations

In this work, we have limited ourselves to the toolbox provided by the Qiskit software. This includes assuming perfect two-level qubits that neglect the higher energy levels present in the transmon qubits [11][12] we are emulating. We are thus not able to model certain classes of quantum errors, such as leakage into these higher levels. We also neglect the underlying implementation of the qubit gates, which often utilizes these energy levels temporarily to achieve the desired transitions [13][14]. In addition to this, we will neglect the modeling of so-called coherent noise, which includes systematically faulty operations on the qubits.

For the $[\![5, 1, 3]\!]$ code to be practical, the issue of performing logical operations on the encoded state needs to be investigated. This has been done in previous work, both theoretically [15] and experimentally [9]. As such, we focus on the issue of preserving the information of one qubit through error correction.

A number of additional QEC schemes exist which could be possible to run on other future chips featuring just a few more qubits. In particular larger stabilizer codes, such as $[\![7, 1, 3]\!]$ which has a fault-tolerant version proposed by [16] using 9 qubits in total, $[\![8, 3, 3]\!]$ which can be made fault-tolerant with 10 qubits [17] or the $[\![9, 1, 3]\!]$ which can

be run without fault tolerance also using 10 qubits. A probably more realistic type of QECC is the surface code, which naturally scales with qubit numbers to enhance its error-protecting properties [18]. The surface codes feature a greater parallelization and can measure the presence of multiple errors simultaneously. A common small surface code with the ability to correct any single-qubit error while retaining high performance is the so-called surface-17 [19], which requires 17 qubits in total arranged in a custom tailored layout. While these alternate paradigms offer many advantages over the $[\![5, 1, 3]\!]$ code, we will focus on studying the simplest model.

2

Theory

Quantum computers store and manipulate information embedded into the combined quantum state of qubits to perform calculations. The goal of QECCs is to redundantly encode these qubit states into a larger Hilbert space of more *physical* qubits than necessary in such a way that a particular class of errors, typically single-qubit errors, leaves the logical information unaffected. The space of "valid" quantum states is referred to as the codespace C, and the states outside this set are identified as invalid information which can only be reached when errors occur. A QEC scheme will continuously measure the state in such a way as to, without affecting the logical information, identify if it lies in the codespace.

To explain this in greater detail, we begin with a brief overview of the necessary theory for quantum computing in the circuit model, followed by the error-correcting properties of stabilizer QECCs. For a more thorough treatment of these topics, we refer to [20, Ch. 4] and [20, Sec. 10.5.5][21][22] respectively. In section 2.3, a more realistic (albeit still rather simplistic) representation of quantum noise and how this can be modeled in a QC simulation is presented. We refer to [20, Sec. 8][23][24] for a more in-depth explanation on this.

2.1 Quantum computing

Quantum computers generalize the concept of classical computing to utilize quantum effects for more efficient data storage and processing. The basic building block of a quantum computer is therefore the *qubit*, a quantum analog to the classical two-level system called *bit*. A qubit is simply any sufficiently stable and controllable two-level quantum system, which can assume superpositions of these states given by

$$|\psi\rangle = \alpha\,|0\rangle + \beta\,|1\rangle\,, \quad \alpha, \beta \in \mathbb{C}, \quad |\alpha|^2 + |\beta|^2 = 1.$$

Quantum states are always normalized, $\langle\psi|\psi\rangle = 1$, and hence the time evolution is given by unitary operators

$$|\psi\rangle = U_N \cdot U_{N-1} \cdot \ldots \cdot U_1\,|\psi_0\rangle\,,$$

which are by definition the set of operators that preserves the complex norm. In the context of quantum computing, the unitary time evolution operators are called *gates*. A quantum computer will be capable of performing any unitary evolution in its *gate*

set, which ideally can be combined to form any desired unitary. The vector space of qubits is finite dimensional and we can thus represent the operators as matrices, with the basis given by

$$|0\rangle \equiv \begin{pmatrix} 1 \\ 0 \end{pmatrix}, \quad |1\rangle \equiv \begin{pmatrix} 0 \\ 1 \end{pmatrix}.$$

The most basic quantum gates are the Pauli operators, defined by the matrices

$$X \equiv \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \quad Y \equiv \begin{pmatrix} 0 & -\mathrm{i} \\ \mathrm{i} & 0 \end{pmatrix} \quad Z \equiv \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}. \tag{2.1}$$

If applied to the state of a qubit, they will transform it as

$$\begin{aligned} X|0\rangle &= |1\rangle & X|1\rangle &= |0\rangle, \\ Y|0\rangle &= \mathrm{i}|1\rangle & Y|1\rangle &= -\mathrm{i}|0\rangle, \\ Z|0\rangle &= |0\rangle & Z|1\rangle &= -|1\rangle. \end{aligned}$$

The X gate can thus be applied on a qubit to "flip" its state between $|0\rangle$ and $|1\rangle$, similarly to a classical NOT gate. The Z gate will apply a phase[1] to the $|1\rangle$ state and the Y gate is a combination of Z followed by X with an extra global phase factor of "i".

The most common way to represent quantum gates, however, is using the *circuit model*, where one represents the time evolution in the form of a diagram read from left to right. For example, in this way of expressing a quantum program

$ZYX|0\rangle$ is equivalent to $|0\rangle$ –[X]–[Y]–[Z]–.

The state $|\psi\rangle$ of a quantum system can however not be directly observed. Instead, one is limited to measuring so-called observables, which are represented by Hermitian operators. When an observable is measured, the outcomes correspond to the eigenvalues of the associated matrix, and the quantum state is effectively projected to the corresponding eigenspace. The Pauli operators are both unitary and Hermitian, and so they can be applied as gates on a qubit, but also correspond to measurable properties. The Z operator will then effectively measure whether the qubit is in the $|0\rangle$ or $|1\rangle$ state, corresponding to eigenvalues 1 and -1 respectively. For simplicity, these outcomes are usually relabeled 0 and 1. The Z observable is in practice usually the only form of measurement that a quantum computer can directly perform, but as we will see later, other observables can effectively be measured using clever quantum circuits.

Since measurable states correspond to eigenvectors, the global phase of a quantum system is physically unobservable. Because of this and the requirement of normalization, a general single qubit state can be written as

$$|\psi\rangle = \cos(\theta/2)|0\rangle + \mathrm{e}^{\mathrm{i}\varphi}\sin(\theta/2)|1\rangle, \tag{2.2}$$

[1] Here, a "phase" refers to the phase of the complex amplitude for the $|1\rangle$ state. Specifically multiplying it by a factor $-1 = \mathrm{e}^{\mathrm{i}\pi}$.

using only two parameters. Here we have defined the amplitude for the $|0\rangle$ state as real.

The single qubit state can then be represented as a vector lying on the so-called *Bloch sphere* [20, p. 15], shown in fig. 2.1. Here, the $|0\rangle$ state corresponds to the north pole, while the $|1\rangle$ state corresponds to the south. The states on the x and y axes similarly correspond to the eigenvectors of the X and Y operators.

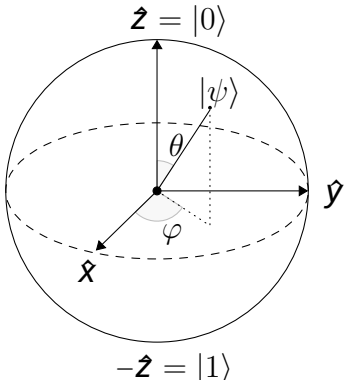


Figure 2.1: *Representation of a single qubit state using the Bloch sphere. Every point on the sphere, defined by its Euler angles, corresponds to one of the possible superpositions of* $|0\rangle$ *and* $|1\rangle$ *using eq.* (2.2).

The computational power of quantum computers, however, comes from the dynamics of multiple qubits. A system of N qubits is described by a superposition of the 2^N possible combinations of their states, written as

$$|\psi\rangle = c_0\,|0...00\rangle + c_1\,|0...01\rangle + c_2\,|0...10\rangle + ... + c_{2^N}\,|1...11\rangle\,, \quad c_n \in \mathbb{C}, \quad \sum_n |c_n|^2 = 1.$$

As an example of how this could allow for more efficient computation, consider the quantum Fourier transform, which is a quantum circuit that performs the discrete Fourier transform on the amplitudes c_n, using $O(n^2)$ gates, compared to the fastest known classical algorithm, FFT, which needs $O(n2^n)$ gates [25][20, Sec. 5.1].

Single qubit states and gates within multi-qubit systems can be represented using tensor products, which are usually omitted for brevity. For example, if three qubits are initialized in the zero state, and the X gate is applied to the first qubit, the Y gate on the second and the identity gate I, equivalent to doing nothing, is applied to the third, then this would be written as

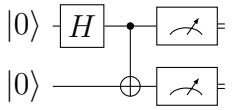
$$XYI\,|000\rangle \equiv (X \otimes Y \otimes I)(|0\rangle \otimes |0\rangle \otimes |0\rangle) = (X\,|0\rangle) \otimes (Y\,|0\rangle) \otimes (I\,|0\rangle) = \mathrm{i}\,|110\rangle\,.$$

More general multi-qubit gates are represented by unitary matrices with dimensions determined by the number of qubits they operate on. An example is the controlled-X gate, simply called the CNOT gate, which acts on a two-qubit state as

$$\mathrm{CNOT} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix}.$$

Note that the basis states are ordered as $|00\rangle\,, |01\rangle\,, |10\rangle$ followed by $|11\rangle$. The gate swaps the amplitudes for the $|10\rangle$ and the $|11\rangle$ state, which can be interpreted as applying an X gate on qubit 1 only if qubit 0 is in the $|1\rangle$ state.

Using the circuit model, we can draw multi-qubit gates as connecting the "wires" of specific qubits. A basic two-qubit circuit may look like this:



This circuit first applies a Hadamard gate, which is defined as

$$H = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} = \frac{X+Z}{\sqrt{2}},$$

and hence it maps the $|0\rangle$ state to the $\frac{|0\rangle+|1\rangle}{\sqrt{2}}$ state[2]. And then proceeds to a CNOT where the first (highest) qubit acts as the control for whether X is applied to the second. The total action of the circuit will thus be

$$\text{CNOT} \cdot (H \otimes I) |00\rangle = \text{CNOT} \cdot \begin{bmatrix} \frac{|0\rangle+|1\rangle}{\sqrt{2}} \otimes |0\rangle \end{bmatrix} = \text{CNOT} \, \frac{|00\rangle+|10\rangle}{\sqrt{2}} = \frac{|00\rangle+|11\rangle}{\sqrt{2}}.$$

We have thus produced what is called a "cat state" (after Schrödinger's cat), which is a superposition of two "opposite" composite states. The subsequent measurements will always produce the outcomes 11 or 00, which is an example of quantum entanglement.

A full list of the gates used in this work and their corresponding matrix representations and circuit symbols can be found in appendix A.

2.2 Stabilizer codes

Stabilizer codes are a natural quantum extension of certain classical codes and even allow for some binary and quaternary codes to be reused. As such, they are inherently suited for dealing with bit and phase flip errors, i.e., erroneous applications of the Pauli operators X, Y, or Z on some qubits. In a quantum computer, an infinite amount of possible single-qubit errors, corresponding to any rotation on the Bloch sphere, could in principle happen, making such a translation seem inappropriate. It turns out, however, that when measuring the presence of one of these errors, one essentially "projects" the continuous error into discrete ones. In general, if a QECC can correct against a set of errors, then it can also protect against any linear combination of these [20, Th. 10.2]. The Pauli matrices, together with the identity I, form a basis for the space of complex 2×2 matrices, and are thus sufficient to represent all unitary single qubit errors.

Stabilizer codes are uniquely determined by their so-called stabilizer $\mathcal{S}$, which is the group of operators that leave the elements of the codespace unaffected. The codepace $C(\mathcal{S})$ is then defined as the eigenspace of $\mathcal{S}$ with eigenvalue $+1$, i.e. $C(\mathcal{S}) = \{\ |\psi\rangle : \forall g \in \mathcal{S}, g\ |\psi\rangle = |\psi\rangle\ \}$. Every QECC has stabilizers, but for the ones referred to as stabilizer codes, valid ones are abelian subgroups of the n-fold Pauli group G_n

[2] This state is the $+1$ eigenvector to the X operator, which lies on the "$\hat{x}$" basis vector of the Bloch sphere in fig. 2.1. Together with the -1 eigenvector $\frac{|0\rangle-|1\rangle}{\sqrt{2}}$ they form an alternative orthonormal basis and are commonly labeled $|+\rangle$ and $|-\rangle$ respectively. The Hadamard is the corresponding basis change matrix.

and cannot contain $-(I^{\ n})$. The n-fold Pauli group is defined as the group generated by tensor products of n Pauli operators. For example,

$$G_1 \equiv \langle X, Y, Z \rangle = \{ \pm I, \pm \mathrm{i} I, \pm X, \pm \mathrm{i} X, \pm Y, \pm \mathrm{i} Y, \pm Z, \pm \mathrm{i} Z \}.$$

One can choose a basis which generates the stabilizer. For a code with n physical qubits encoding k logical ones, there will be $n-k$ elements. That is, $\mathcal{S} = \langle g_1 \ldots, g_{n-k} \rangle$. Errors are identified by measuring the eigenvalues $g_l |\psi\rangle = \beta_l |\psi\rangle$ where $\beta_l = \pm 1$, which gives a corresponding 'syndrome' $\{\beta_l\}_{l=0}^{n-k}$. A trivial syndrome, $\beta_l = 1$ for all l, means that the state is still stabilized by $\mathcal{S}$ and thus has no errors.

Let us look at how the errors relate to the stabilizers. If an error E (or indeed any unitary operation acting on $|\psi\rangle$) commutes with a stabilizer g then it will not affect the eigenvalue since $gE|\psi\rangle = Eg|\psi\rangle = E|\psi\rangle$. This set of operators is called the *centralizer* $\mathcal{Z}(\mathcal{S})$, and errors in it will be undetectable. For the $\mathcal{S}$ defined above, the centralizer coincides with the normalizer $N(\mathcal{S}) = \{E : E\mathcal{S}E^\dagger = \mathcal{S}\}$, which we will henceforth use[3]. If, however, the error lies within $\mathcal{S}$ itself, it will leave the codespace unaffected by definition, hence the following condition for error detection.

Theorem 1. *A set of errors $\{E\} \subset G_n$ is **detectable** for the stabilizer code $C(\mathcal{S})$ if $\forall E, E \notin N(\mathcal{S}) \backslash \mathcal{S}$.*

While theorem 1 describes which errors are detectable, it imposes no constraint that their syndromes are unique. For an error to be correctable one needs to be able to identify a correction procedure for each syndrome which maps the state back to the initial codeword. If the error E is applied, it may flip the eigenvalue of one or more of the stabilizers, $g_l E|\psi\rangle = \beta_l E g_l |\psi\rangle, \forall |\psi\rangle$, or more compactly

$$E^\dagger g_l E = \beta_l g_l, l = 0, ..., n-k. \tag{2.3}$$

Thus, if two errors cause the same syndrome, $E_i^\dagger g_l E_i = E_j^\dagger g_l E_j = \beta_l g_l$ for $l = 0, ..., n$, we have

$$E_i E_j^\dagger g = g E_i E_j^\dagger,$$

hence[4] $E_i^\dagger E_j \in N(\mathcal{S})$. Again, there is a trivial case where $E_i^\dagger E_j \in \mathcal{S}$, where the errors perform the same operations on codewords since

$$E_i^\dagger E_j |\psi\rangle = |\psi\rangle \quad \Longleftrightarrow \quad E_j |\psi\rangle = E_i |\psi\rangle.$$

The full error correction condition [20, Th. 10.8] thus reads:

Theorem 2. *A set of errors $\{E_i\} \subset G_n$ is **correctable** for the stabilizer code $C(\mathcal{S})$ if, $\forall i, j, : E_i^\dagger E_j \notin N(\mathcal{S}) \backslash \mathcal{S}$.*

[3] The definition of the normalizer is weaker and implies that errors that map elements in the stabilizer to *other* elements, as $Eg_iE^\dagger = g_j$, are allowed. However, when $\mathcal{S}$ is a subgroup of G_n and does not contain $-I$, the normalizer and centralizer are the same.

[4] Where we have used the fact that $\beta = 1/\beta$ to flip the order of the Hermitian conjugation.

Analogous to the Hamming distance for classical error correcting/detecting codes, one usually defines the distance n as the minimum number of single-qubit errors needed to go from one valid codeword to another, which corresponds to the smallest amount of (non-trivial) Pauli operators in the tensor product of elements (called *weight*) in $N(\mathcal{S})\backslash\mathcal{S}$. By theorem 1, to *detect* any t single-qubit errors a code needs distance $d \geq t+1$, and by theorem 2 that to *correct* any t single-qubit error it needs $d \geq 2t+1$. One can classify stabilizer codes with the label $[\![n, k, d]\!]$, which implies that the code uses n physical qubits to encode k logical ones, and has distance d.

2.2.1 The $[\![5, 1, 3]\!]$ stabilizer code

The $[\![5, 1, 3]\!]$ is the stabilizer code defined by the generators (where we omit the tensor products)

$$\begin{aligned} g_1 &= XZZXI \\ g_2 &= IXZZX \\ g_3 &= XIXZZ \\ g_4 &= ZXIXZ. \end{aligned} \tag{2.4}$$

It is the smallest code possible with distance 3, which is required to correct any one single-qubit error. That the code has this ability can be directly verified using theorem 2.

Proof. The set of single qubit errors $\{E_i\}$ consists of $\{I, X_1, ..., X_5, Y_1, ..., Y_5, Z_1, ..., Z_5\}$. We have to show that, for any i and j, either $E_i^\dagger E_j$ lies in the stabilizer, or there is an element in the stabilizer that does not commute with it. We first note that $i = j \Rightarrow E_i^\dagger E_j = I$. I stabilizes every vector and so it has to be in $\mathcal{S}$.

We then note that the product of errors acting on the same qubit results in another single-qubit error, i.e., $E_j^\dagger E_i = E_k$ for some k. Two Pauli operators of the same type commute (of course) and two of different types do not. One can thus easily see that E_k would fail to commute with at least one generator since every 'column' in eq. (2.4) contains both an X and a Z and no Pauli operator can commute with both.

For all $E_i^\dagger E_j$ consisting of Pauli operators on two different qubits, we can construct two elements in S such that $E_i^\dagger E_j$ has to not commute with one of them. Simply choose one of the qubits affected by $E_i^\dagger E_j$ and look up the corresponding stabilizers in table 2.1, they will have an identity operator acting on that qubit and different Pauli operators on every other. $E_i^\dagger E_j$ will commute with the 'I' component but must fail to commute with a Pauli operator of one of the stabilizers since, as before, they are different. □

As mentioned above, the codespace is defined as the $+1$ eigenspace of the stabilizer generators in eq. (2.4). By defining the logical Z operator to be $Z_L = ZZZZZ$[5] (similarly $X_L = XXXXX$ and $Y_L = YYYYY$), which commutes with the generators,

[5] This can be put into the alternative form $g_2 Z_L = ZYIIY$, showing that it is a weight-3 operator, which is required for any distance-3 code.

Table 2.1: *Stabilizers in S with an identity operator in the tensor product acting on the qubit corresponding to the column, and di erent Pauli operators acting on every other qubit.*

$g_2 = IXZZX$	$g_3 = XIXZZ$	$g_4 = ZXIXZ$	$g_1g_4 \propto YYZIZ$	$g_1 = XZZXI$
$g_1g_3 \propto IZYYZ$	$g_2g_4 \propto ZIZYY$	$g_1g_2 \propto XYIYX$	$g_2g_3 \propto XXYIY$	$g_3g_4 \propto YXXYI$

we can further define the logical basis states $|0_L\rangle$ and $|1_L\rangle$ as the ± 1 eigenspaces of this operator within the code space[6]. They turn out to be

$$\begin{aligned} |0_L\rangle = \frac{1}{4}[&|00000\rangle + |10010\rangle + |01001\rangle + |10100\rangle \\ &+ |01010\rangle - |11011\rangle - |00110\rangle - |11000\rangle \\ &- |11101\rangle - |00011\rangle - |11110\rangle - |01111\rangle \\ &- |10001\rangle - |01100\rangle - |10111\rangle + |00101\rangle] \end{aligned} \tag{2.5}$$

and

$$\begin{aligned} |1_L\rangle = \frac{1}{4}[&|11111\rangle + |01101\rangle + |10110\rangle + |01011\rangle \\ &+ |10101\rangle - |00100\rangle - |11001\rangle - |00111\rangle \\ &- |00010\rangle - |11100\rangle - |00001\rangle - |10000\rangle \\ &- |01110\rangle - |10011\rangle - |01000\rangle + |11010\rangle]. \end{aligned} \tag{2.6}$$

Note how $Z_L = ZZZZZ$ measures the parity of the states. That is, the eigenvalue is $+1$ for an even amount of ones and -1 for an odd amount. Hence the $|0_L\rangle$ states only have even parity components and $|1_L\rangle$ only has odd parity components.

2.3 Noise and errors in quantum computers

In practice, errors in the form of erroneous unitary single-qubit operations are a fairly unrealistic idealization. Quantum errors can roughly be divided into two groups, coherent and incoherent noise [26][23]. Coherent errors roughly correspond to incorrectly applied quantum operations, or incomplete knowledge of how these function. This typically involves systematic over- or under-rotations. Coherent noise has been shown not to significantly influence the break-even error thresholds of surface codes [27], therefore we focus on modeling the incoherent noise.

Incoherent errors instead describe the perpetual influence of the environment on the quantum state, such as thermal relaxation. To model such interactions, the qubits have to be described as an open system. Closed quantum systems evolve with the reversible operations of unitary operators. Open quantum systems, however, can be affected by a broader range of transformations described by *quantum channels* $\mathcal{E}$.

[6] Note here that Z_L can be seen as having "undefined" behavior on states outside the codespace, and may be considered not actually to be the true "logical" Z operator, which is given by $Z_L^{\text{true}} = |0_L\rangle\langle 0_L| - |1_L\rangle\langle 1_L|$. They are related by $Z_L^{\text{true}} = Z_L I_L$, where $I_L = |0_L\rangle\langle 0_L| + |1_L\rangle\langle 1_L|$ is the projector on the codespace. However, Z_L is more feasible to experimentally measure.

These are completely positive, trace-preserving maps on the space of density matrices ρ.

A quantum operation can be broken down into its operator-sum representation

$$\mathcal{E}(\rho) = \sum_k E_k \rho E_k^\dagger$$

where $\mathcal{E}(\rho)$ is the final state and E_k are called the Kraus operators, which must fulfill $\sum_k E_k^\dagger E_k = 1$.

2.3.1 Amplitude damping

One naturally occurring source of error in qubits is the tendency of a quantum state to undergo thermal relaxation from $|1\rangle$ to $|0\rangle$, since $|1\rangle$ represents a higher energy state in most superconducting qubits[7]. This can be modeled using the quantum channel

$$\mathcal{E}_{\mathbf{AD}}(\rho) = E_0 \rho E_0^\dagger + E_1 \rho E_1^\dagger$$

where E_0, E_1 are the Kraus operators given by

$$E_0 = \begin{pmatrix} 1 & 0 \\ 0 & \sqrt{1-\gamma} \end{pmatrix}, E_1 = \begin{pmatrix} 0 & \sqrt{\gamma} \\ 0 & 0 \end{pmatrix}.$$

Here, γ represents the probability of $|1\rangle \to |0\rangle$ decay. The action of the channel is thus given by

$$\mathcal{E}_{\mathbf{AD}} \begin{pmatrix} \rho_{00} & \rho_{01} \\ \rho_{10} & \rho_{11} \end{pmatrix} = \begin{pmatrix} \rho_{00} + \gamma\rho_{11} & \rho_{01}\sqrt{1-\gamma} \\ \rho_{10}\sqrt{1-\gamma} & (1-\gamma)\rho_{11} \end{pmatrix}.$$

We can see that amplitude damping reduces the components of the density matrix corresponding to superpositions, thus making the state "more classical".

2.3.2 Phase damping

Phase damping is an entirely quantum mechanical form of noise, and describes the loss of information about the relative phase of the eigenstates. Its Kraus operators are given by

$$E_0 = \begin{pmatrix} 1 & 0 \\ 0 & \sqrt{1-\lambda} \end{pmatrix}, \quad E_1 = \begin{pmatrix} 0 & 0 \\ 0 & \sqrt{\lambda} \end{pmatrix}.$$

The action on a single qubit state is thus

$$\mathcal{E}_{\mathbf{PD}} \begin{pmatrix} \rho_{00} & \rho_{01} \\ \rho_{10} & \rho_{11} \end{pmatrix} = \begin{pmatrix} \rho_{00} & \rho_{01}\sqrt{1-\lambda} \\ \rho_{10}\sqrt{1-\lambda} & \rho_{11} \end{pmatrix}.$$

[7] At finite temperatures, there is also a small contribution in the other direction $|0\rangle \to |1\rangle$. However, at the operating temperatures and frequencies of superconducting qubits this rate is exponentially suppressed [23, p. 14] and thus the normal amplitude damping channel forms a good approximation.

2.3.3 Continuous errors

The combined operation of the amplitude and phase damping channels is given by

$$\mathcal{E}_{\mathbf{PD}} \circ \mathcal{E}_{\mathbf{AD}}(\rho) = \begin{bmatrix} 1-(1-\gamma)\,\rho_{11} & \rho_{01}\sqrt{1-\gamma}\sqrt{1-\lambda} \\ \rho_{10}\sqrt{1-\gamma}\sqrt{1-\lambda} & (1-\gamma)\,\rho_{11} \end{bmatrix},$$

where we have used that $\rho_{00} = 1 - \rho_{11}$.

However, this model assumes discrete noise events where the errors occur with a given probability. In reality, the errors are applied continuously over time, which converts the constant factors to exponential decays

$$\begin{aligned} 1-\gamma &= e^{-t/T_1} \\ \sqrt{1-\gamma}\sqrt{1-\lambda} &= e^{-t/T_2}, \end{aligned} \tag{2.7}$$

where t is the timed passed, and T_1 and T_2 are the so-called relaxation and dephasing times, which can be experimentally measured [28]. The continuous amplitude and phase damping channel then acts on a single qubit as

$$\rho \to \mathcal{E}_{\mathbf{D}}(\rho) = \begin{bmatrix} 1-\rho_{11}e^{-t/T_1} & \rho_{01}e^{-t/T_2} \\ \rho_{10}e^{-t/T_2} & \rho_{11}e^{-t/T_1} \end{bmatrix}. \tag{2.8}$$

We can see that T_1 parameterises the rate of the process $|1\rangle \to |0\rangle$, while T_2 parameterises the decay of the off-diagonal components. From eq. (2.7), we can see that with pure amplitude damping, the T_1 and T_2 times are related by $T_2 = 2T_1$. However, when phase damping is introduced, this causes additional decay of the off-diagonal components[8] which instead results in the relation $T_2 \leq 2T_1$. For typical systems, $T_2 > T_1$, and so the excited state decay is the most limiting component of thermal decay.

Using eq. (2.8), the evolution of a pure single qubit $|\psi\rangle = \alpha\,|0\rangle + \beta\,|1\rangle$ is given by

$$\mathcal{E}_{\mathbf{D}}(|\psi\rangle\langle\psi|) = \begin{bmatrix} 1-(|\alpha|^2-1)e^{-t/T_1} & \alpha\beta\; e^{-t/T_2} \\ \alpha\;\beta e^{-t/T_2} & |\beta|^2 e^{-t/T_1} \end{bmatrix}.$$

After a time t, the probability of remaining in the same state is given by the state fidelity

$$\langle\psi|\,\mathcal{E}_{\mathbf{D}}(|\psi\rangle\langle\psi|)\,|\psi\rangle = |\alpha|^2 + \mathrm{e}^{-\frac{t}{T_1}}\,|\beta|^4 - \mathrm{e}^{-\frac{t}{T_1}}\,|\alpha|^2|\beta|^2 + 2\,\mathrm{e}^{-\frac{t}{T_2}}\,|\alpha|^2|\beta|^2.$$

For the special cases of a qubit initiated in the $|1\rangle$, $|+\rangle$ and $|0\rangle$ states the expression reduces to

$$\begin{aligned} \langle 1|\,\mathcal{E}_{\mathbf{D}}(|1\rangle\langle 1|)\,|1\rangle &= \mathrm{e}^{-\frac{t}{T_1}} \\ \langle +|\,\mathcal{E}_{\mathbf{D}}(|+\rangle\langle +|)\,|+\rangle &= \frac{1}{2} + \frac{1}{2}\mathrm{e}^{-\frac{t}{T_2}} \\ \langle 0|\,\mathcal{E}_{\mathbf{D}}(|0\rangle\langle 0|)\,|0\rangle &= 1. \end{aligned} \tag{2.9}$$

In the first two cases, the probabilities reduce to exponential decays with lifetimes given by T_1 and T_2 respectively. For the $|0\rangle$ state, the fidelity never decreases, as we expect from our interpretation of thermal decay.

[8] The specific contribution of phase damping to T_2 can be isolated by $\frac{1}{T_2} = \frac{1}{2T_1} + \Gamma_\phi$, where Γ_ϕ is called the *pure dephasing* rate.

3

Methods

All simulations of the $[\![5, 1, 3]\!]$ code were done using Qiskit [10] (version 0.25.0), an open-source framework for creating and simulating quantum programs founded by IBM Research. Qiskit provides the ability to perform density matrix simulations with noisy quantum channels defined by Kraus operators, including channels for amplitude and phase damping described in section 2.3. We have also utilized Qiskit's transpilation functionality, along with manual adjustments, to adapt the circuits to the device limitations.

In this chapter, we start by illustrating the circuits that realize the $[\![5, 1, 3]\!]$ code, and proceed to outline the specific device limitations and how the circuits were adapted to them. We then detail how the simulation and associated data processing were performed. We have extended the base Qiskit framework for applying noise, transpiling circuits and analyzing results, which will be explained in the rest of this chapter.

3.1 The $[\![5, 1, 3]\!]$ circuit

The procedure for realizing repeated error correction using the $[\![5, 1, 3]\!]$ code, as outlined in multiple places [17][29], goes as follows:

1. Encode the desired single qubit state
2. Measure the four stabilizers
3. Decode the error syndrome using classical processing
4. Perform corrections to reverse any detected error
5. Repeat from 2.

To perform error detection by post-selection instead of error correction, replace step 4 by "discarding the run when errors are detected". When performing actual computation using error correction, one would also perform logical operations on the encoded state, in between step 4 and 5.

The circuit to encode an arbitrary state is depicted in fig. 3.1. It maps the single qubit state $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$ into $|\psi_L\rangle = \alpha|0_L\rangle + \beta|1_L\rangle$, with $|0_L\rangle$ and $|1_L\rangle$ as defined in eq. (2.5) and eq. (2.6). This encoding circuit was adapted from [15] and is, to the best

of our knowledge, the shortest known encoding circuit capable of encoding an arbitrary state in the $[\![5, 1, 3]\!]$ code to date. Shorter circuits capable of producing specific logical states have been shown [30].

To measure the stabilizer, the logical states are entangled with an ancilla qubit, which is then projectively measured[1]. The general scheme for measuring an observable M with eigenvalues ± 1 on the multi-qubit state $|\psi\rangle$, is shown in fig. 3.2. Applying this scheme to the stabilizer generators in eq. (2.4) produces the full stabilizer measurement shown in fig. 3.3. The ancilla is reset after each measurement and reused. Once the full syndrome is measured, the appropriate correction is applied to $|\psi_L\rangle$, following a simple lookup table detailed in appendix B. As the stabilizer generators commute, measuring them consecutively will project the state to one of their simultaneous eigenvectors. Thus, any continuous error can be "discretized". After error correction, the state will effectively be projected to the codespace.

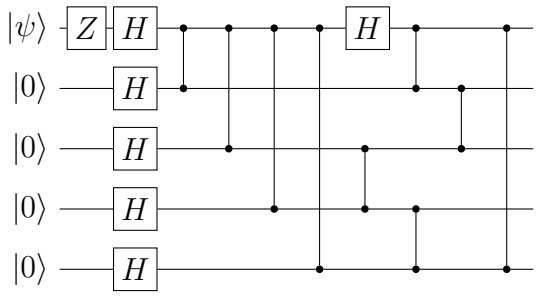


Figure 3.1: *Encoding circuit for the* $[\![5, 1, 3]\!]$ *code. The single qubit state* $|\psi\rangle$ *in the first qubit will be encoded into all five qubits to form the logical state.*

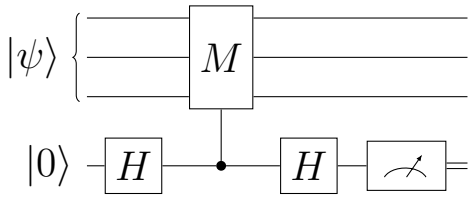


Figure 3.2: *Projective measurement of the observable* M *with eigenvalues* ± 1 *utilizing an ancilla qubit. Here* $|\psi\rangle$ *may represent a multi-qubit state.*

In principle, the ancilla qubit does not need to be reset after measurement. The effect of the circuit in fig. 3.2 is more specifically to "flip" the ancilla qubits state from the initial value (which would be $|0\rangle$ or $|1\rangle$ depending on the previous measurement outcome) when the measurement of observable M results in -1. However, if the last stabilizer measurement results in -1, this will effectively "flip" all syndromes retrieved in the next cycle. When neglecting the reset step, the alternate correction operations shown in appendix B should be used instead. Because of the very limited functionality for conditioning gates on measurement outcomes in Qiskit, this turns out to be unfeasible to implement. As the ancilla is more stable in its $|0\rangle$ state, not resetting the ancilla may in principle slightly influence the correctness of the error measurements. However, as we demonstrate in appendix D.1, the impact is negligible and it can be disregarded in experimental setups.

[1] The need for this comes from the fact that, as mentioned earlier, quantum computers lack the ability to directly measure arbitrary observables. Commonly, they are only capable of measuring the Pauli Z operator, i.e. determining the $|0\rangle$ or $|1\rangle$ states of the individual qubits. Another limitation of quantum computers is that direct measurements destroy the quantum state.

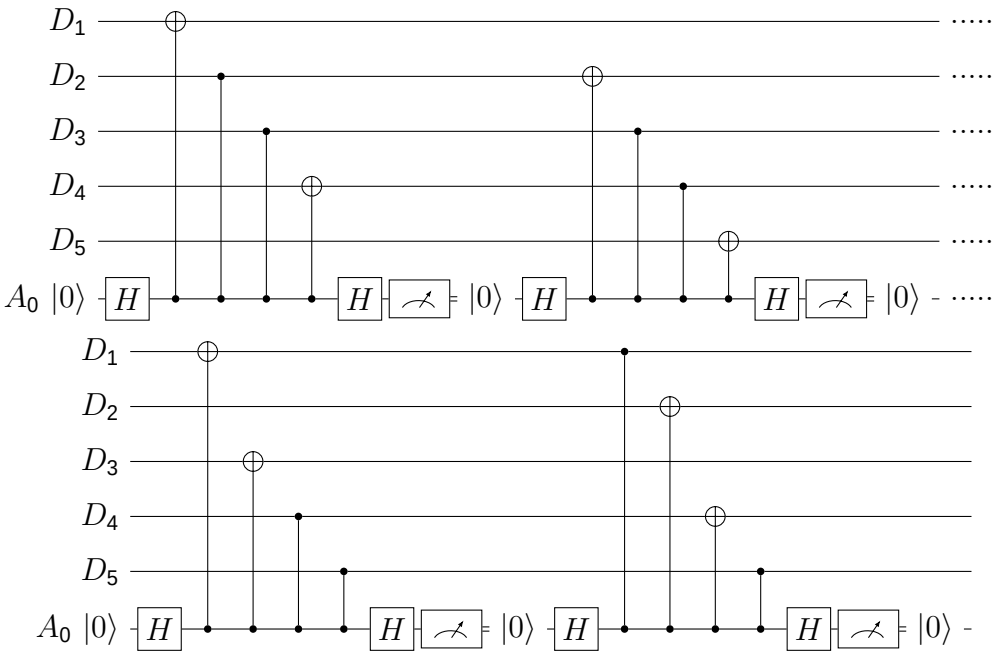


Figure 3.3: *Full stabilizer cycle of the* $[\![5, 1, 3]\!]$ *code with full connectivity. Qubits* D_{1-5} *denotes the data qubits encoding our logical state* $|\psi_L\rangle$ *and* A_0 *the ancilla qubit.*

3.2 Device limitations

Real quantum hardware is limited in what operations can be performed, something which has to be carefully considered when constructing quantum programs on current small scale devices. For example, only certain qubits have the physical connection required to allow two-qubit gates between them to be performed. In this project, we have adapted the $[\![5, 1, 3]\!]$ circuits for the two 7-qubit device connectives shown in fig. 3.4, where the left is a common square architecture utilized in previous QEC experiments [6][5], and the middle is the triangular lattice investigated by WACQT. From hereon we instead refer to the triangular grid as *hexagonal* layout, corresponding to the shape formed by its constituent qubits. We will also compare the devices to the ideal case of full connectivity, shown to the right.

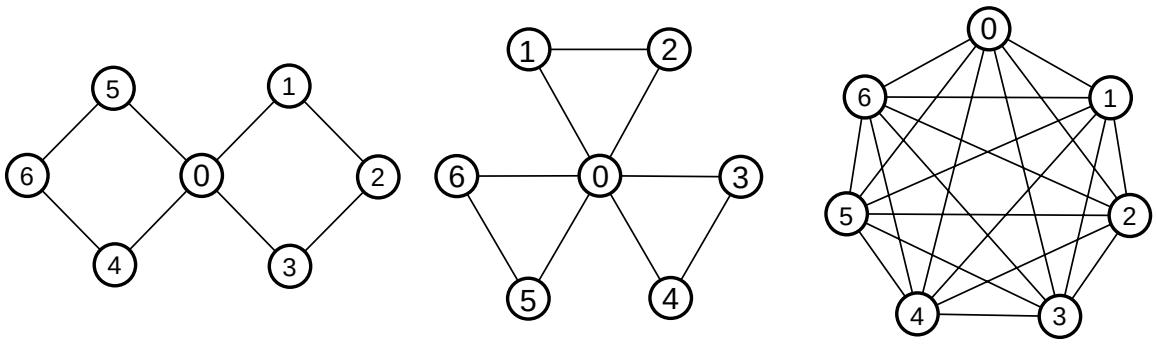


Figure 3.4: *Connectivity of explored devices. The vertices represent qubits and the edges which pairs of qubits two-qubit gates can operate on.*

In addition to limited connectivity, only a subset of gates is typically implemented in

hardware. In this work, we consider the gate set[2]

$$G = \{CZ, \text{iSWAP}, X, Y, Z(\theta), \sqrt{X}, \sqrt{Y}\}$$

which is the set experimentally demonstrated at WACQT [31] excluding iSWAP. It is therefore relevant to explore the option of limiting circuits to CZ as the only two-qubit gate, that is $G \setminus \{\text{iSWAP}\}$. However, using both CZ and iSWAP in the same circuit has been experimentally shown in similar hardware [32]. Therefore we still utilize the full gate set G unless otherwise specified. For further details regarding specific gates, the reader is referred to appendix A.

3.2.1 Transpilation of a quantum program

The process of modifying (quantum) programs for compatibility with device limitations without altering their behavior is called transpilation[3]. To adapt the necessary connectivity, the assignment of theoretical circuit qubits to the physical qubits is first considered. Ideally, all two-qubit gates can be performed within the limited connectivity with an adequate placement. If not, SWAP gates, defined as

$$\text{SWAP} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix},$$

must be added. As the name suggests, the SWAP gate is a two-qubit gate which effectively swaps the quantum states of the two qubits involved, allowing the qubit states to be "moved" into a position where the desired two-qubit gate can be performed. For this purpose, the Qiskit transpiler was used as an initial step, followed by further manual optimization. The built-in transpiler implements the *SABRE* [33] algorithm for assigning qubits and introducing SWAPs, which we found consistently performed the best compared to other available algorithms.

Thereafter, the gates are decomposed into those of the required set based on preset equivalences. While several such decompositions are already defined within the `qiskit.circuit.EquivalenceLibrary`, it offers no native translation of SWAP gates into iSWAP gates, the most favorable option for our gate set. As such, we added a decomposition of the SWAP gate into an iSWAP, CZ and two single qubit Z-rotations, which is shown in Figure 3.5.

Finally, the Qiskit transpiler attempts to optimize the circuit based on a performance metric. Optimization level 1 (from 0-3) was used which combines adjacent gates when possible. Higher levels offer further gate cancellation by stochastic reordering of commuting gates, but were not used due to incompatibility with the iSWAP gate. As the optimization and qubit layout is stochastic, the transpilation was repeated multiple times, and the circuit of the shortest duration was chosen. Further optimization was performed manually through inspection.

[2] The $Z(\theta)$ gate is also commonly labeled $U_1(\theta)$ or $R_Z(\theta)$

[3] Some authors simply prefer the term "compilation", we chose to adopt the language used in the Qiskit API.

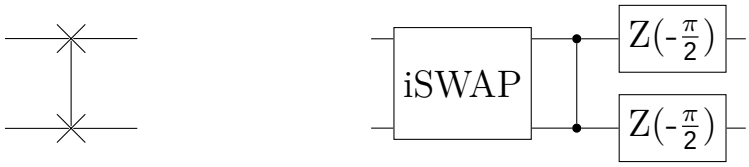


Figure 3.5: *Equivalent gates used in transpilation. The left circuit represents the SWAP gate the right its breakdown into the available gate set.*

Table 3.1: *Our choice for default operation, T_1 and T_2 times used across simulations, unless otherwise stated. Note that the Z-gate is excepted from single qubit gates, as it is instant.*

Parameter	Standard times [ns]
Single qubit gate	20
Two-qubit gate	100
Measurement	300
Feedback	350
T_1	40000
T_2	60000

We have also investigated the use of the transpilation tool t|ket⟩ [34], a quantum development platform produced by Cambridge Quantum Computing Ltd, which includes state-of-the-art quantum transpilation. We found, however, that it was heavily optimized for specific NISQ devices and was unable to efficiently decompose the SWAP gates into an equivalent iSWAP dependent circuit such as in fig. 3.5. This could potentially be achieved through the customization available in their API, but as the circuits we consider in this project are fairly small it was considered outside the scope of this project in favor of hand made optimizations.

3.3 Noise model

In this work, we limit ourselves to incoherent noise based on the continuous amplitude and phase damping channels, as defined in eq. (2.8). Qiskit implements this error channel directly in the `qiskit.providers.aer.noise.thermal_relaxation_error` module. This error channel is entirely time dependent. Realistic values for the necessary parameters vary greatly between experimental setups, ranging between 20 µs to 200 µs for $T_{1,2}$ [35][5][6][7][9][36][37][38]. The same goes for gate times provided by several sources [5][9][36][13]. Therefore, we choose to run simulations with $T_1 = 40$ µs, $T_2 = 60$ µs along with gate times close to experimental values, detailed in table 3.1 as base values unless otherwise specified. Operation times are split into single-qubit gate, two-qubit gate, measurement and feedback delay. Feedback delay refers to the time from measurement of the error syndrome to the application of a correction gate, corresponding to step 3 and 4 of the list in section 3.1. The exception to this is the $Z(\theta)$ gate, which can be implemented with zero duration by adjusting the underlying pulses of adjacent gates [39].

Qiskit does not natively support applying noise to qubits during idle periods, it is

limited to noise applied in conjunction with gates. We have extended this by implementing code for analyzing the time dependence of circuits given specific gate times. We calculate the duration qubits need to stay idle in between gates and directly insert error channels in the circuit proportional to the length of these periods.

3.4 Extraction of logical qubit fidelities

To quantify the efficiency of QEC, we will investigate how well the state of an encoded logical qubit is preserved compared to the state of a single qubit. As we saw in eq. (2.9) using quantum state fidelity, the probability of a single qubit remaining in its initial state over time depends greatly on the initial condition. For simplicity, we will compare to the worst-case scenario, which for a single qubit is an exponential decay with time constant being the smallest of T_1 and T_2.

During the simulations, we have access to the full density matrix of the seven-qubit state at all times. We can remove the states of the ancilla qubits through the partial trace operation. The probability of measuring the encoded state ρ as equal to its initial pure state $|\psi_L\rangle$ is similarly given by the state fidelity

$$F(|\psi_L\rangle, \rho) = \langle\psi_L| \rho |\psi_L\rangle . \tag{3.1}$$

If ρ also describes a pure state $\rho = |\psi_\rho\rangle\langle\psi_\rho|$, this reduces to the familiar $|\langle\psi_L|\psi_\rho\rangle|^2$.

Using the default $T_1 = 40$ µs, $T_2 = 60$ µs, we simulate the decay of the (non-error corrected) encoded logical qubit, shown in the left side of fig. 3.6. We can see that it does not show any dependence on its initial conditions. The decays of the fidelity for each cardinal state are equal until they all start to approach thee $|00000\rangle$ state. This has a 1/16 overlap with the $|0_L\rangle$ state and 0 overlap with $|1_L\rangle$ as can be seen in eq. (2.5) and eq. (2.6).

One can further project ρ onto the codespace using

$$\rho_{L,ij} = \langle j|\rho|i\rangle / P_L, \quad |i\rangle, |j\rangle \in \{ \ |0_L\rangle, \ |1_L\rangle \} , \tag{3.2}$$

where $P_L = \sum_i \langle i|\rho|i\rangle$ is the probability of ρ residing in the codespace [6]. The "logical" fidelity to the single qubit state $|\psi\rangle$ is then simply given by

$$F_L(|\psi\rangle, \rho) = \langle\psi| \rho_L |\psi\rangle ,$$

which corresponds to the probability of matching $|\psi_L\rangle$ conditioned on ρ lying in the codespace. Or, equivalently, the probability of matching the initial state after an ideal stabilization to the codespace. This is related to eq. (3.1) by $F = F_L P_L$. In the right side of fig. 3.6 we can see the logical fidelity of the cardinal logical states to their initial conditions over time. The encoded state shows inherent protection against logical errors from the amplitude and phase damping channels. Only after roughly 30 µs, when they all approach $|00000\rangle$, does the underlying logical information get corrupted.

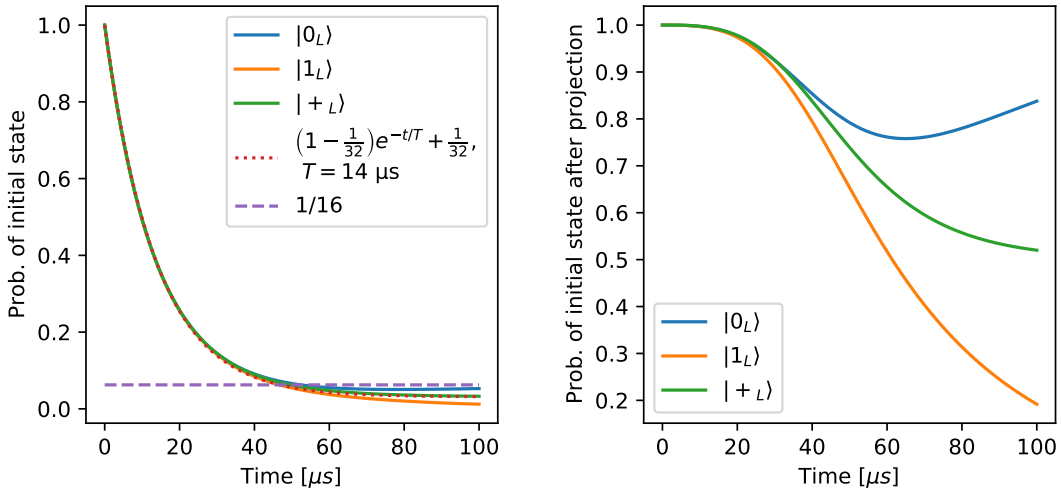


Figure 3.6: *Probability of matching initial state (fidelity) of logically encoded cardinal states without error correction over time. The left plot shows just the state fidelity. The right plot shows the fidelity where the extracted density matrix has been projected to the codespace (in post-processing outside of the simulation). Note how the encoded state decays far quicker than its constituent qubits, but the logical information is still preserved for a longer time.*

4

Results

There are a plethora of different key elements that determine the performance of the $[\![5, 1, 3]\!]$ code, for which the quantum state fidelity of the encoded state to its target is our primary metric. We aim to investigate how different factors influence the fidelity and whether a break-even is reachable, where the logical and single-qubit lifetimes are equal. The chapter begins by first showcasing the transpiled circuits, followed by encoding fidelities. Afterwards, we turn to active error-correction and noise parameters are varied to search for a possible break-even threshold. Finally, we investigate how the code's performance is impacted by different layouts and strategies.

4.1 Circuit transpilations

Following the procedures in section 3.2.1, encoding and stabilizer cycles were transpiled both together and separately. The resulting encoding circuits are shown in fig. 4.1 for both connectivities along with circuit times and depth in table 4.1, yielding a depth of. The most notable difference is the number of swaps necessary; two and three for the hexagonal and square layout respectively. In both cases, qubits are assigned to put the initial state in the central qubit (denoted 0 in fig. 3.4 and fig. 4.1), and repositioning the ancilla into the center as the final step after encoding.

While it remains unproven that these are the shortest possible circuits under our constraints, it is most likely the optimal transpilation of the initial scheme. The same unitary circuit can be composed in several different ways, but the transpiler only seeks to optimize and adapt the specific gates involved. For shorter circuits with identical device limitations, one might need to design an encoding from scratch with the specific gates and connectivity in mind.

As for the stabilizer cycles, there is not as much optimization to be done. CNOT gates in the projective measurements can be decomposed into CZ without adding circuit time[1], and for the hexagonal layout, no iSWAP is necessary at all. Only the square layout benefits from optimization, as it requires four iSWAPs for each cycle. These can, however, be neatly run in parallel with the actual stabilizer measurements, resulting in only 120 ns of additional time (Compared to the 800 ns of time to perform four swaps), as seen in table 4.1. Decomposing SWAPs into only CZ gates instead,

[1] While the decomposition of CNOT into CZ adds two Hadamard gates on the target qubit, these do not increase circuit time as these are performed in parallel with existing operations.

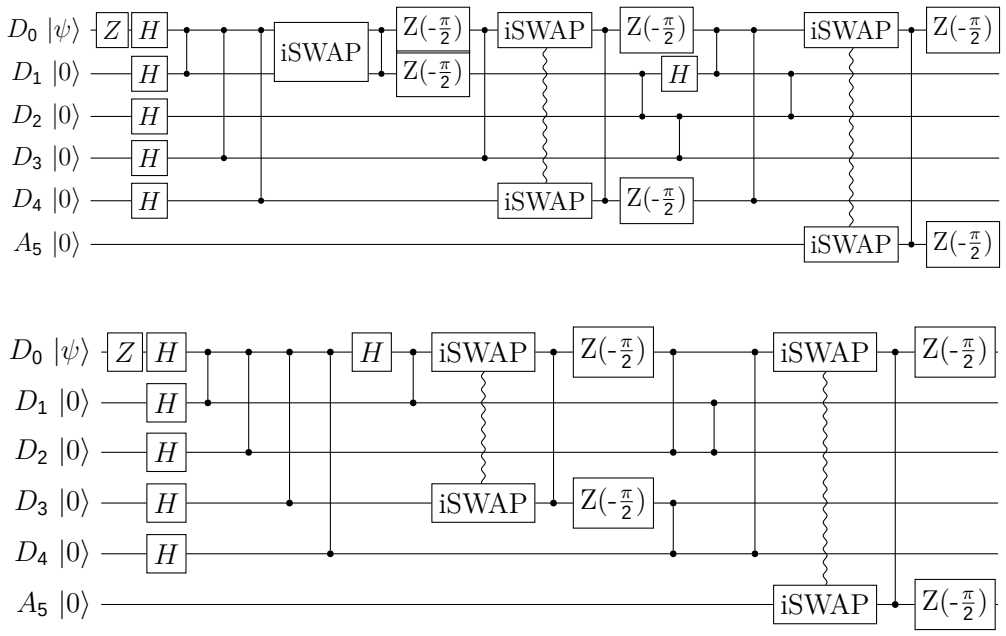


Figure 4.1: *Encoding circuits following the square (top) and hexagonal (bottom) layout. D represents data qubits set to encode the logical state and A the ancilla qubit, with numbers representing their position in fig. 3.4. Note that qubit 6 represents the ancilla during encoding, but is swapped into the central 0 position at the end. While H is not within our gate set, its decomposition is equivalent to a single qubit gate, omitted in the figure for clarity. For full details, refer to appendix A.*

does not see such benefits, resulting in a total of 400 ns extra time per stabilizer cycle, compared to the hexagonal counterpart. The full circuits are detailed in appendix C.

Table 4.1: *Circuit times and depth for the encoding and a stabilizer cycles using the default gate times in table 3.1, along with the general time formulas decomposed into sums of constituent parts. In these formulas, t_1 denote single-qubit gate time, t_2 the two-qubit gate time, t_m measurement time and t_f feedback time. Note that circuit times may di er from these formulas in some rare cases as, for example, two-qubit gates and measurements are run in parallel. Circuit depth is calculated as the number of steps in the circuit, with a step corresponding to any set of gates which can be run in parallel. Here we omit the Z gate as it can be made instant [39]. The full circuits for all these layouts are found in appendix C.*

Device layout	**Encoding**			**Stabilizer cycle**		
	Time [ns]	Time formula	Depth	Time [ns]	Time formula	Depth
Full Connectivity	640	$2t_1 + 6t_2$	8	3310	$8t_1 + 16t_2 + 4t_m + t_f$	28
Hexagonal	1140	$2t_1 + 11t_2$	13	3310	$8t_1 + 16t_2 + 4t_m + t_f$	28
Hex., no iSWAP	1460	$8t_1 + 13t_2$	21	3310	$8t_1 + 16t_2 + 4t_m + t_f$	28
Square	1220	$t_1 + 12t_2$	13	3430	$9t_1 + 17t_2 + 4t_m + t_f$	30
Square, no iSWAP	1700	$10t_1 + 15t_2$	25	3710	$13t_1 + 19t_2 + 4t_m + t_f$	44

4.2 Encoding

With all circuits defined, the first step of utilizing QEC is to encode the logical state. Using our default device parameters, the probability of matching the target 5-qubit

encoded state was determined using the state fidelity of eq. (3.1) and is shown in table 4.2. The fidelities range from 0.884 for the square layout without iSWAP gates to 0.970 for full connectivity. The fidelity is directly dependent on the duration of the encoding circuit, and thus follows the circuit times in table 4.1. We see the best fidelity when encoding the $|-_L\rangle$ state for each architecture.

Table 4.2: *Success probabilities (state fidelity F and logical state fidelity F_L) of encoding four cardinal states for the different transpiled circuits in table 4.1 using T_1 = 40 µs, T_2 = 60 µs and the gate times in table 3.1.*

Device layout	**Encoding success prob.** F				**Projected encoding suc. prob.** F_L			
	$\|0_L\rangle$	$\|1_L\rangle$	$\|+_L\rangle$	$\|-_L\rangle$	$\|0_L\rangle$	$\|1_L\rangle$	$\|+_L\rangle$	$\|-_L\rangle$
Full Connectivity	0.959	0.959	0.959	0.970	0.994	0.994	0.999	1.0
Hexagonal	0.919	0.919	0.925	0.935	0.987	0.987	0.999	1.0
Hex. no iSWAP	0.901	0.901	0.906	0.923	0.986	0.986	0.999	1.0
Square	0.911	0.911	0.916	0.929	0.988	0.988	0.999	1.0
Square, no iSWAP	0.890	0.890	0.884	0.919	0.987	0.987	0.999	1.0

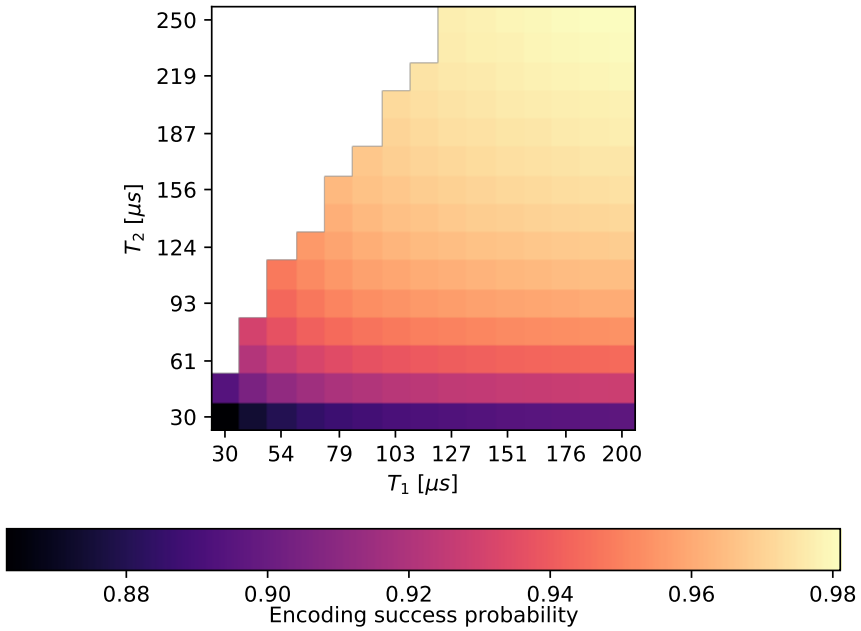


Figure 4.2: *Success probabilities of encoding the $|1_L\rangle$ state (fidelity of the states F) with the hexagonal layout for a range of T_1 and T_2 times, using the gate times in table 3.1. Empty tiles represent the invalid $T_2 > 2\,T_1$ region.*

We also projected the density matrix onto the logical codespace, using eq. (3.2), before measuring the fidelity. This *logical* fidelity F_L corresponds to the probability of correctly encoding the desired state conditioned on measuring ρ to be in the codespace, which happens with a probability P_L. This is related to the physical fidelity F through $F = F_L P_L$. We can see significantly improved fidelities after projection, indicating that the logical information itself is not scrambled to a high degree. When performing error correction, the state will ideally be projected to the codespace, increasing F to, at most, the level of F_L.

We also investigate the impact of a range of T_1 and T_2 times for the $|1_L\rangle$ state using the hexagonal layout, which is shown in fig. 4.2. Values of $T_1 = 80$ µs and $T_2 = 95$ µs are sufficient to reach a fidelity of $F = 0.95$, while almost $T_1 = 200$ µs and $T_2 = 250$ µs is needed to reach $F = 0.98$.

Note that our noise model is invariant to the scaling of the T_1, T_2 and gate time parameters in proportion to one another. Moreover, the impact of single and two-qubit gate times can almost entirely be described by total duration of the encoding circuit, thus one can determine the encoding success probability for other gate times using

$$P(T_1, T_2, t_1, t_2) = P_{\text{def}}\left(T_1 \frac{1140 \text{ ns}}{2\,t_1 + 11\,t_2}, T_2 \frac{1140 \text{ ns}}{2\,t_1 + 11\,t_2}\right),$$

where $P_{\text{def}}(T_1, T_2)$ is the success probability using the default gate times in table 3.1, which is read from fig. 4.2, and t_1 and t_2 are alternate single and two-qubit gate times in ns, respectively.

4.3 Active error correction

Having explored the transpilation of circuits and the encoding of the logical states, it is time to evaluate the correction of errors. The decay of the state fidelity F and the logical state fidelity F_L of a repeatedly error-corrected logical qubit is shown in fig. 4.3. Ideally, the encoded state is continuously projected to the codespace from the error correction, and should therefore approach the mixed state $\rho = (|0_L\rangle\langle 0_L| + |1_L\rangle\langle 1_L|)/2$ after complete loss of information at $t \to \infty$, which corresponds to a fidelity of 0.5. We find, however, that the overlap with the codespace, given by $P_L = F/F_L$, takes on the *constant* value of 0.81 throughout the entire simulation, except at the beginning when the qubit is perfectly initialized in the pure $|0_L\rangle$ state. Excluding the initial data point[2], the logical fidelity, nonetheless, with good approximation decays exponentially as

$$F_L(t) = 0.5\, e^{-t/T_L} + 0.5,$$

where t is the time passed and T_L is the logical lifetime[3], and thus approaches $F_L = 0.5$ for $t \to \infty$. The physical fidelity therefore also decays with the same lifetime T_L µs as

$$F(t) = F_L(t)P_L = 0.5\, P_L\, (e^{-t/T_L} + 1).$$

Hence, the parameters T_L and P_L are sufficient to specify the fidelity of the encoded state over time. We perform a curve fit using `scipy.optimize` and find $T_L = 26$ µs, which is shorter than that of the single qubit, indicating that we are below the break-even threshold.

The probability of the error-corrected qubit retaining its initial state surpasses the single qubit after roughly 20 µs. However, this is a consequence of the asymmetry

[2] In the formulas, as opposed to the plot, we redefine $t = 0$ as being after the first error correction cycle for clarity.

[3] Other works have used the exponential decay of other metrics to define the logical lifetime, we elaborate on this and make comparisons in appendix D.2.

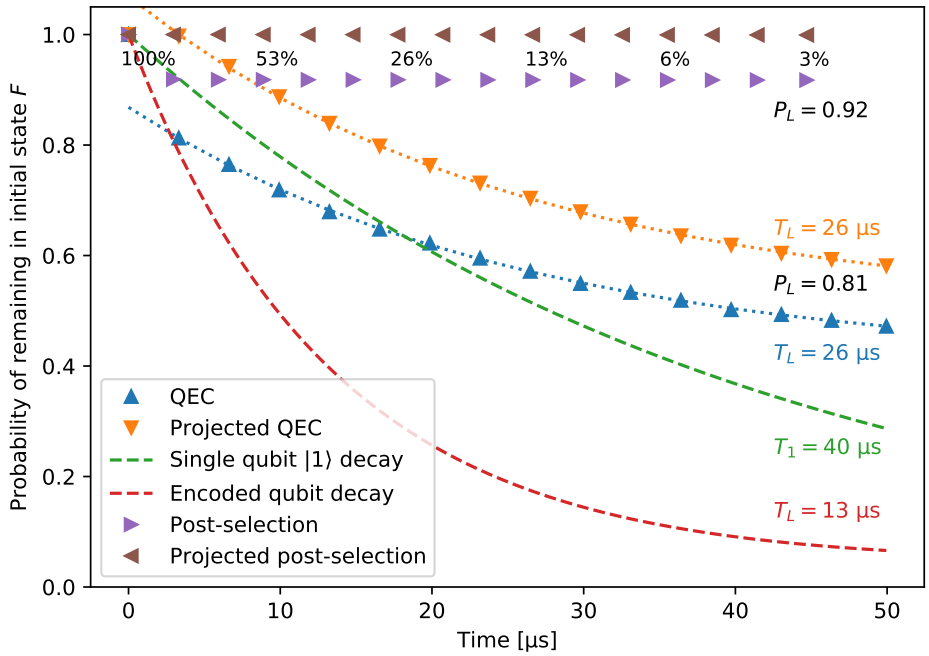


Figure 4.3: *Repeated quantum error correction and detection. We initialize the simulator in the $|0_L\rangle$ state ($|1\rangle$ for single qubit) perfectly for every simulation, and measure the fidelity F to that state over time, which corresponds to the probability of remaining in the initial state. The quantum error-corrected qubit (blue* N*) drops steeply after the first stabilizer cycle, and afterwards decays exponentially (blue dotted line). Its lifetime is improved over the uncorrected logically encoded qubit (red dashed line), but lower than that of the uncorrected single qubit (green dashed line). The logical fidelity F_L, corresponding to the fidelity after projecting the extracted density matrix to the codespace, is also shown (orange* H*). It shows no initial drop and actually remains roughly at 1 after the first correction step, but then decays exponentially with a lifetime of $T_L = 26$ µs. The fraction $P_L = F/F_L$, corresponding to the probability of the logical qubit residing in the codespace, remains constant after the first correction step at $P_L = 0.81$. Post-selection is also shown (purple* I*), which also exhibits a dip after the first cycle followed by an almost constant fidelity. The logical fidelity of the post-selection (brown* J*) shows no dip, and in fact stays above 0.999 during the entire simulation. The P_L probability for the post-selection is thus also constant at $P_L = 0.92$. The percentage of states not post-selected away (for both projected and non-projected) is also shown as numbers and decays exponentially with lifetimes* 14 µs*.*

of the amplitude damping channel; the single qubit decays towards the $|0\rangle$ state as opposed to an equal mixed state of $|0\rangle$ and $|1\rangle$. The symmetric decay of the logical qubit may be considered an advantage, but does not reflect the distinguishability of the $|0_L\rangle$ and $|1_L\rangle$ states. The error-corrected qubit does show an improved lifetime over the non-error-corrected logically encoded qubit, demonstrating that the error-correction itself improves the stability of the logical qubit.

Figure 4.3 also shows the fidelity and logical fidelity of the encoded qubit using post-selection of errors from repeated *error-detection*. The cycles for these circuits are shorter as the recovery step is omitted. Here we see $P_L = 0.92$, also constant for the entire simulation. The projected logical qubit shows perfect retention of logical

information, indicating that the noise is always detected and post-selected before errors can congregate to undetectable ones. This is consistent with fig. 3.6, where the physical fidelity F decays quickly but the logical F_L stays constant for a while. The detection of errors is delayed for long enough that small errors, which only affect F, occur but not for F_L to decay. The percentage of runs not terminated because of detected errors is printed in black at intervals, and decays exponentially with a lifetime of 14 µs, which is roughly the same as the decay rate of the non-error-corrected logical qubit.

The high fidelity of the post-selected states has been demonstrated experimentally [6][5], but with a more pronounced (albeit still very slight) slope than present in our simulations. This is likely caused by our idealized error model, which lacks modeling of significant error modes, such as leakage, that may cause high weight errors that stabilizer codes are not adept at preventing.

The performance of the error correction is significantly hampered by faulty error decoding caused by errors occurring in between the measurements of the stabilizer generators. Specifically, if an error that would trigger a -1 measurement outcome on one of the stabilizer generators occurs after that generator has been measured, but before the end of the cycle, it will be identified as a different error or be missed entirely. This may cause the recovery scheme to apply a faulty correction procedure, which can result in a weight-2 error. If the error is missed, it may still be caught in the next cycle as long as another error does not occur before detection. This flaw is not very surprising as the circuit contains no operations other than repeated stabilization, so the probability of errors occurring during this process is more likely than outside it. In a future application of quantum error correction, logical calculations may take up a significant portion of the computing time and errors during these could still be detected.

The impact of faulty error decoding is demonstrated in fig. 4.4, which compares the basic QEC to a version with perfect error decoding. Here, we let the system idle for the duration of a stabilizer cycle, and then perform instant error-free detection and correction. In this way, the same amount of noise is applied, but never during the detection step. We see that the "dip" disappears. The logical lifetime is also significantly improved, above that of the single qubit, reaching the breaking-even point of the error-correction.

Figure 4.4 also shows post-selection of errors. The perfect decoding version almost flawlessly isolates error-free runs, resulting in maximum fidelity. The fact that the perfect decoding post-selection outperforms the perfect decoding error-correction is probably a result of weight-2 errors being detectable, as there is no need to uniquely identify every error. Normal post-selection is for this reason also not as hampered by incorrect error decoding. While the active error-correction suffers from faulty corrective operations, post-selected runs will only contain errors when the error syndrome incorrectly indicates no errors whatsoever. This can be seen from the reduces "dip" after the first cycle, compared to the active error-correction.

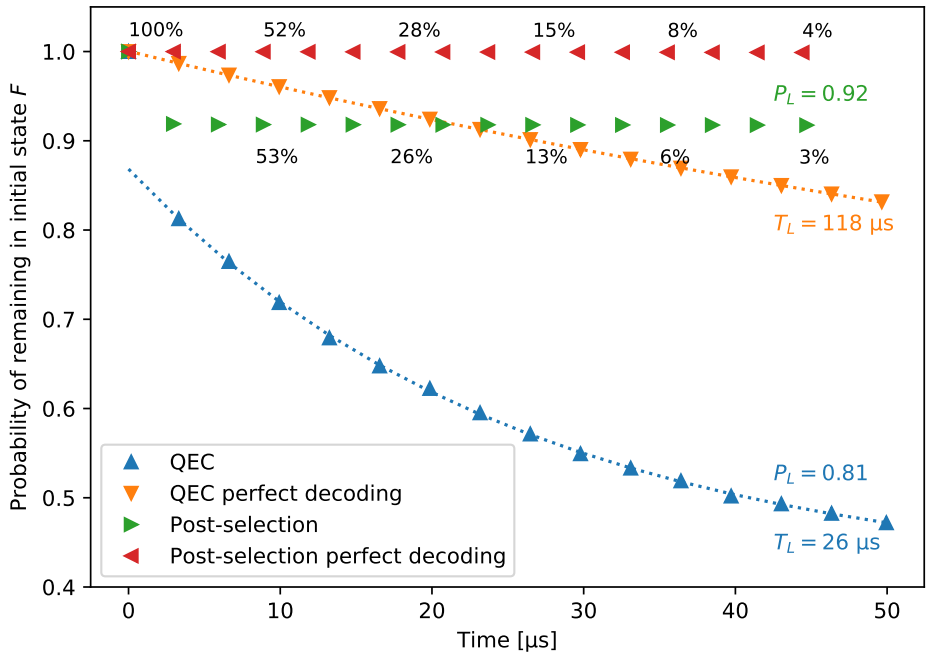


Figure 4.4: *Influence of decoding errors on error correction and detection. The same normal repeatedly error corrected and post-selected runs as in fig. 4.3 are shown as blue* N *and green* I*. A simulation with perfect decoding, where the state is left idle for the duration of a cycle followed by instant noiseless error correction (orange* H*), shows a significantly improved lifetime and no initial dip, surpassing the single qubit by a large margin. A similar perfect decoding version of post-selection also shows no dip (red* J*), and in fact flawlessly selects error free runs. We can see that post-selection is less hampered by incorrect decoding, as the errors do not need to be uniquely identified to post-selected the run. The fraction of states not post-selected away is also shown and decays exponentially with lifetimes* 14 µs *for both the normal post-selection and the perfect decoding post-selection.*

4.3.1 Dependence on device parameters

We now move on to investigate how the QEC depends on the device characteristics, and whether the error-correction break-even point can be reached with higher quality components. In terms of qubit decoherence and our noise model, the parameters to consider are operation times, T_1 and T_2. Here we further split the operation times into the four different categories used across simulations; single-qubit gates, two-qubit gates, measurements and feedback time. We limit ourselves to the hexagonal layout in this subsection.

Initially, we compare the impact of different operation times. This was done by varying the time of specific operations one at a time, while keeping the others at their standard duration specified in table 3.1, two-qubit gates, measurement and feedback times separately, as well as scaling all gates proportionally[4]. We have also included a fifth category; a time delay between cycles. This cycle delay is added directly *after* the correction procedure (and our snapshot of the density matrix). During this time

[4]Scaling single qubit gates separately was omitted due to their short gate time.

qubits are idle, but it can be interpreted as the possibility of performing logical operations on the encoded state. Results are given in fig. 4.5 with both logical lifetime and probability of remaining in code space as a function of the total resulting stabilizer cycle duration.

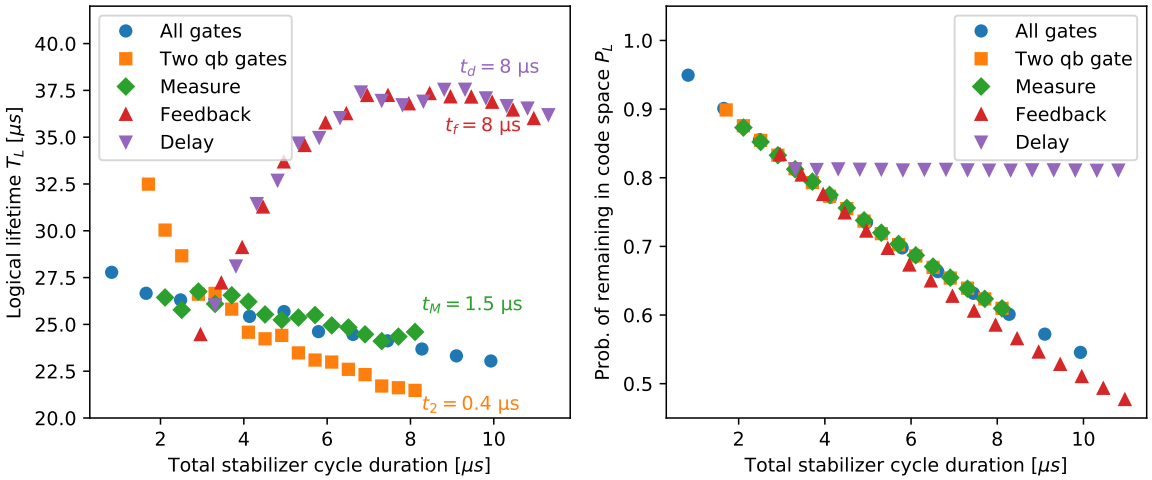


Figure 4.5: *The logical lifetime (left) and probability of remaining in code space (right) of the error-corrected qubit using a range of operation times, plotted against the resulting total duration of the stabilizer cycle. The blue circles describes the scaling of all operations in proportion to one another. The remaining colors/symbols describe varying the duration of one operation type, from t = 0 ns to the value specified at the rightmost data point, while keeping the remaining operations at their standard durations. The point of convergence for all plots around the cycle time t_{cyc} = 3 µs corresponds to standard gate times used. Durations are evenly spaced, but their specific values can also be obtained using table 4.1 for the hexagonal layout.*

There are several noteworthy results from fig. 4.5. When scaling all operation times proportionally, there is very little improvement in the logical lifetime with T_L ranging from 28 µs for $t_{cyc} = 1$ µs to 24 µs at $t_{cyc} = 10$ µs. While this suggests that cycle time has minimal impact, the probability of remaining in the code space P_L is almost entirely dependent on the total cycle time, and decreases roughly in proportion to it.

Despite this, an improvement of only the two-qubit gate times increases the lifetime far beyond their impact on the cycle durations. T_L increases almost exponentially, peaking at $T_L = 32.5$ µs with instant CZ[5] and other operations unchanged. While instant gates are impossible, a 40 ns CZ gate has been shown experimentally [13] (compared to standard time of 100 ns used here). This is likely a result of two-qubit gates having a particularly large impact on the accuracy of the decoding. Thus, they are a possible point of focus when realizing the $[\![5, 1, 3]\!]$ code.

Feedback time and idle delay after correction, however, act in the opposite way; longer durations increase the logical lifetime T_L. This effect peaks at cycle durations of 8 µs for both idle delay and feedback, which corresponds to an idle-time of 5 µs. A simulation of error correction at these peak values compared to the standard ones is

[5] While all two-qubit gates are sped up, only CZ is used in the stabilizer cycle.

shown in fig. 4.6. We can see that any improvement of the fidelity from the increased lifetime when using a longer feedback delay is negated by the decreased P_L. However, the insertion of an idle-time in between the stabilizer cycles does show a real improvement in the preservation of the logical qubit, increasing the lifetime to $T_L = 37$ µs, narrowly missing the break-even point.

The increased lifetime from inserting a delay in between cycles can likely be explained by the inherent protection of the underlying logical state, as shown in fig. 3.6. The logical fidelity F_L shows the upper limit of how much physical fidelity can be recovered by error correction, which is almost constant for 20 µs, far longer than the 3.3 µs duration of one cycle. Performing correction cycles too often is thus not only unnecessary, but may cause more harm than good by letting errors propagate throughout the circuit. We therefore conclude that, for our default operation times, T_1 and T_2, optimal performance is given when idling for ~ 5 µs after each error-correction cycle. In theory this idle time could instead be used for computations on the logical qubit.

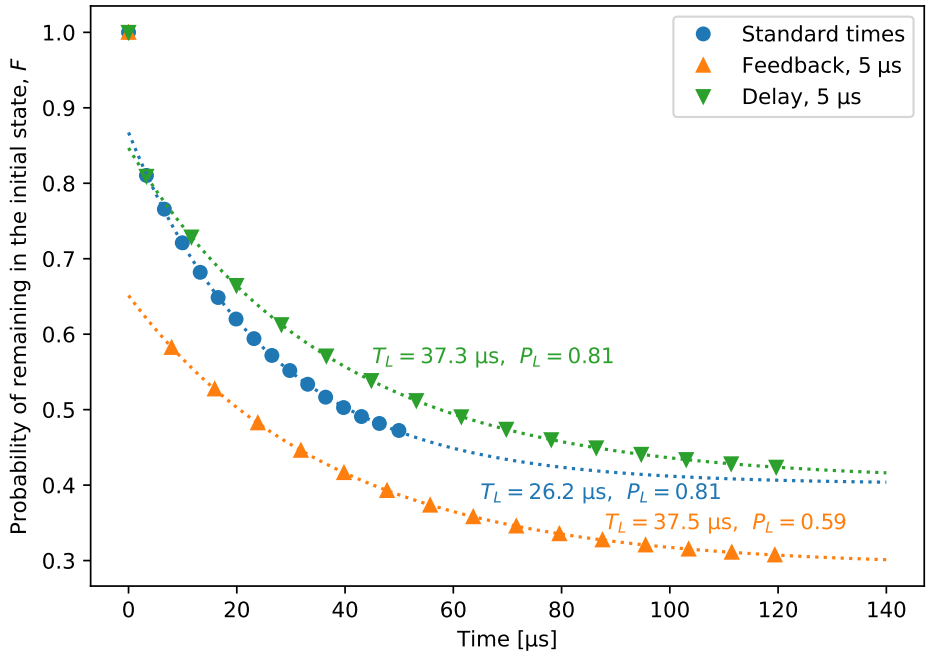


Figure 4.6: *Demonstration of the improvement from including an idle-time in between stabilizer cycles. Repeated quantum error correction using the standard operation times (which includes no delay) is shown as blue ●, along with error correction where the feedback time is increased to 5* µs *(orange* N*) and where 5* µs *of idle-time is inserted between cycles (green* H*). While the two methods of adding delay yields roughly equal lifetimes, a longer feedback time drastically lowers the probability P_L of remaining in the code space, thereby resulting in a net loss of fidelity. However, an idle-time before initializing the next round of error correction shows the same increase in lifetime, with no decline in P_L, resulting in a real gain of logical state preservation. This shows a large improvement but still narrowly missed the error-correction break-even threshold.*

We now move on to investigate the impact of the T_1 and T_2 parameters. We perform simulations for a range of values including the idle delay, with and without perfect

decoding, which is shown in fig. 4.7. As our noise model is invariant to scaling T_1, T_2 and all operation times proportionally we should see that the optimal idle-time is dependent on T_1 and T_2. For simplicity, we have chosen to assume that it scales as $t_d(T_1, T_2) = 5\,\mu s \cdot (T_1 + T_2)/(100\,\mu s)$, where "100 µs" represents $T_1 + T_2$ for the default noise parameters.

For the normal simulation, we find that both T_L and P_L increase steadily with T_1 and T_2. For biased choices of parameters, in particular when $T_2 \gg T_1$, we find that the logical lifetime may exceed one of the decay parameters. This technically qualifies as a breaking of the threshold compared to the worst case for the single qubit, which is the definition we use for this thesis. Using the minimum lifetime of a single qubit can be considered to give an unjustified advantage to the logical encoding. Other works have used the mean fidelity over the cardinal states of the Bloch sphere for the single qubit [40]. This will result in a decay that is not quite exponential and so we have chosen the minimum to more easily determine the break-even point. For a more in-depth discussion on the definition of the logical lifetime, see appendix D.2. A stronger version of the condition for breaking even, where T_L surpasses both T_1 and T_2, is reached when both parameters are at 120 µs (where we find $T_L = 122$ µs) or above.

The break-even threshold is, however, reached for every simulation when using perfect error decoding, corroborating the result that this is the dominant bottleneck for the $[\![5, 1, 3]\!]$ code. We also found that $P_L = 1$ for every choice of parameters in the perfect decoding simulations, and is thus omitted from the figure.

4.3.2 Comparing connectivity graphs

Finally, we compare the performance of the $[\![5, 1, 3]\!]$ code on the connectivity graphs presented in fig. 3.4. Much like the previous section, this was done while initializing the simulator perfectly in the $|0_L\rangle$ state, as the quality of the encoding has already been compared in section 4.2. Simulations were carried out for both the square and hexagonal architecture. The circuit for full connectivity is equal to that of the hexagonal as all data qubits have direct connection to the ancilla. These connectivities only differ when encoding, which is not considered here. Two different setups are compared for the hexagonal chip; a standard and a *pipelined* approach. The pipeline utilizes the seventh qubit by swapping with the ancilla after the stabilizer but before ancilla readout (circuit is found in appendix C). This allows for a time save of 300 ns per cycle with our default times, as the subsequent stabilizers can be performed while measuring the previous one. Such a setup was not considered for the square grid as the seventh ancilla is two nodes away from the central qubit and would require two swaps instead of one, offering no speedup.

Fidelities and lifetimes from these simulations are presented in fig. 4.8 both with and without a 5 µs delay between cycles as discussed in section 4.3.1. We see a slight difference in logical lifetimes; 26.2, 21.9 and 26.1 µs without delay for the hexagonal, square and pipeline configuration respectively. With idle time, lifetime is increased to 37.6, 33.7 and 38.8 µs. As expected, the square grid falls short in both cases with a significantly lower lifetime. The pipeline approach however, improves drastically with

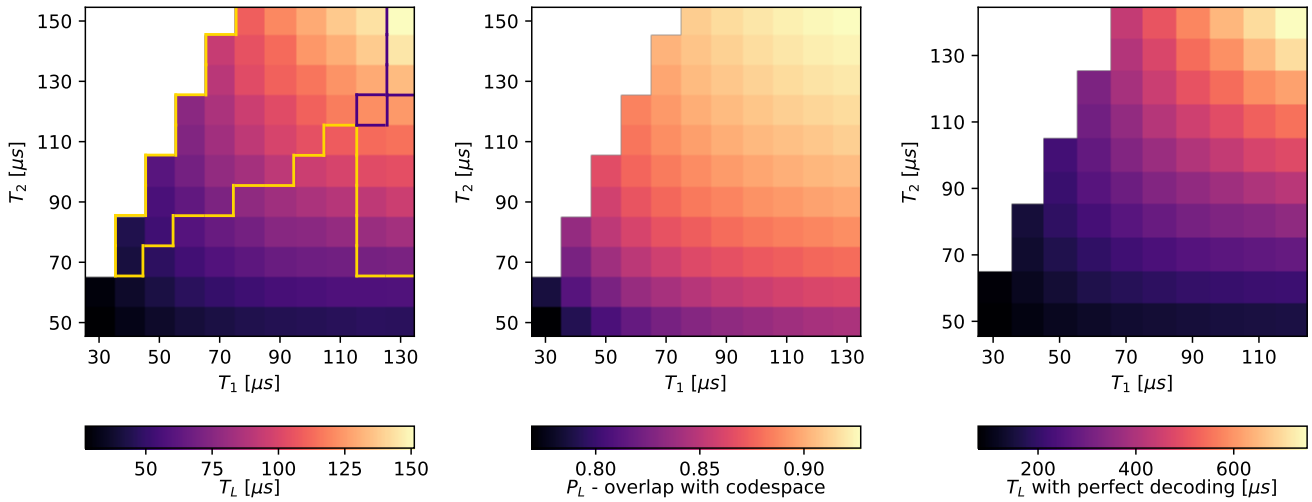


Figure 4.7: *Logical lifetime T_L and overlap with logical codespace P_L for different values of T_1 and T_2 using standard gate times. The left and middle picture corresponds to regular QEC. The orange border marks parameter values where T_L surpasses T_1 or T_2, marking the break-even threshold when comparing the worst-case decays to that of the single qubits. Note how the threshold is reached for a large span of values where the single qubits are heavily T_1 limited. The dark purple border similarly marks where T_L surpasses both T_1 and T_2. The right plot shows perfect-decoding simulations with an idle-time matching the full stabilizer cycle time, followed by instant (noiseless) measurements. This version broke the threshold and maintained $P_L = 1$ for every choice of parameters and thus these where omitted from the figure. The blank area in top left corners of each plot corresponds to the invalid parameter space $T_2 > 2T_1$.*

added delay, slightly outperforming the regular approach giving the best results of all configurations.

Similar to results in section 4.3, this could be explained by faulty error decoding. Despite only a small increase of 120 ns cycle time between hexagonal and square layouts, the latter shows worse results than scaling of gate times in fig. 4.5 suggests. This added time however, is by the fourth stabilizer. As such, there is a larger time frame for errors to arise at the end. If these errors would yield syndromes with -1 eigenvalues on any of the first three stabilizers, they are incorrectly identified and a erroneous correction step may cause an uncorrectable weight two error. Another explanation is that the introduction of four iSWAPs into the cycle gives additional paths for errors to propagate; ultimately decreasing the state fidelity.

The increased lifetime with pipeline shows promise as the best performer, although its strongly dependent on the underlying gate times. Time saving comes from the difference in operation times between a measurement and double two qubit gates[6] i.e. the time it takes to measure or completely swap the ancilla. Therefore speeding up measurement time while keeping two-qubit gates constant lessens the time gain and (most likely) the lifetime increase. The opposite also applies; speeding up CZ and

[6] The difference is multiplied by 3 as it can be done for the first three stabilizers. The fourth must wait for feedback time regardless and offers no such benefit.

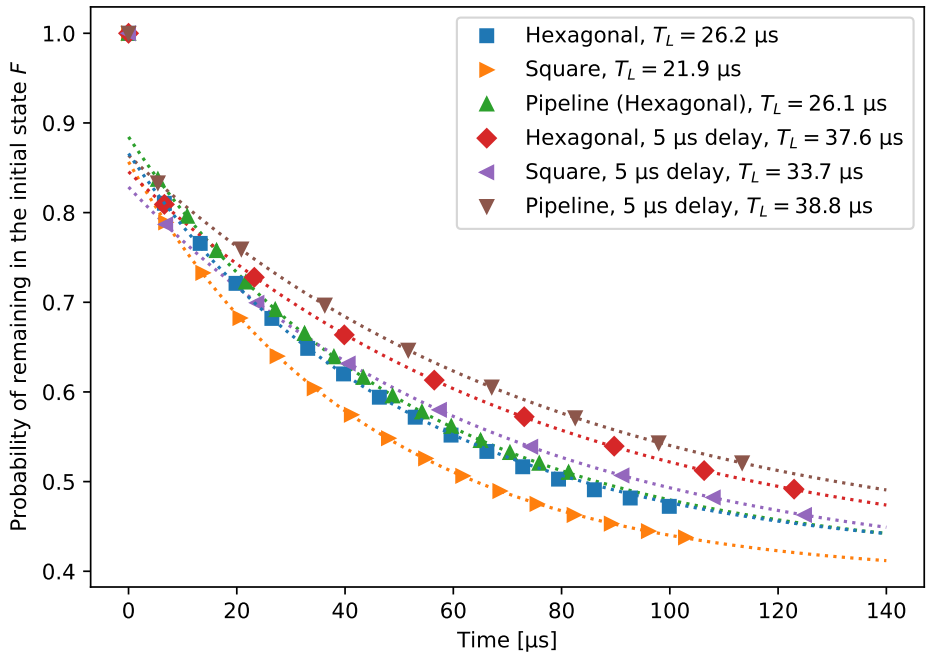


Figure 4.8: *Repeated quantum error correction on $|0_L\rangle$ for di erent setups: Standard procedure on a hexagonal layout (blue), square layout (orange I) and pipelined approach for hexagonal layout (green* N*). The same layouts were also tested with a 5* µs *delay between cycles plotted as red , purple* J *and brown* H *for the hexagonal, square, and pipeline approach respectively. Shapes represent data points and dotted lines fitted exponential decays. All simulations were carried out using noise parameters detailed in table 3.1 and the best setup falls just short of the break-even point of T_L = 40* µs*, corresponding to the worst-case decay of a single qubit.*

iSWAP should enhance the improvement instead.

Despite this, we still fall just short of the break-even of 40 µs. While we have presented several areas of possible improvement, it is critical to remember that this break-even point is for the worst-case single qubit. Throughout simulations we used $T_1 = 40$ µs and $T_2 = 60$ µs yet fail to reach $T_L = \min(T_1, T_2)$. To improve upon this, focus should be at minimizing decoding errors. Both errors propagating due to faulty extraction during stabilizer measurements and possibly the classical processing of the resulting bit string.

5

Conclusions

To summarize, we have investigated the feasibility of the $[\![5, 1, 3]\!]$ error-correcting code in terms of the coherent noise limit for a variety of setups and parameters. Circuits were transpiled to match a scalable square and hexagonal grid with gates native to WACQT devices. Encoding fidelity was determined for these layouts, and the logical lifetime for repeated stabilization and correction was estimated. We showed that the best performance was given when idling between each error-correction cycle for 5 µs, which resulted in a lifetime of $T_L = 38.8$ µs compared to the single qubit limited by $T_1 = 40$ µs with the considered parameters. Decreasing gate times, specifically two qubit gates, showed the most promise in increasing logical lifetime. Aside from that, the threshold could be reached by increasing T_2 while keeping T_1 constant. Since we defined a threshold as the worst case of a single qubit (in this case, T_1), improving T_2 leaves the threshold unaffected while benefiting the encoded state.

We pinpointed the cause of most errors by testing the limits of perfect syndrome extraction and perfect error correction, realized through error-free measurements and post-selection respectively. This gave the unexpected results that our scheme is not limited by qubit decoherence but the correct identification of such errors. The mentioned idle time thus reduced the frequency of error detection and correction, decreasing the number of decoding errors. Another possible solution to decoding is to parallelize stabilizers. However, such approach requires four ancillas, and the $[\![5, 1, 3]\!]$ code offers no obvious way to do this due to its need of high connectivity.

However, we do not ultimately rule out the usefulness of experimental implementations of the $[\![5, 1, 3]\!]$ code despite frequent decoding errors, as some crucial aspects of QEC could still be demonstrated. Repeated QEC showed significantly higher fidelity than decay of the logical state $|\psi_L\rangle$. As such, it can still successfully correct errors in the majority of cases, even if this does not show an improvement over the constituent qubits. The $[\![5, 1, 3]\!]$ code also remains the smallest distance-3 code and can thus act as a stepping stone in the experimental realization of QEC. It acts as a natural extension of the $[\![4, 1, 2]\!]$ experiments [6][5] without requiring more qubits, the clear difference being the classical feedback in correction steps. The delay time of this step has little negative impact on the performance, so there is no need to implement *fast* classical feedback to realize the $[\![5, 1, 3]\!]$ code.

5.1 Outlooks

Our results also open up for further exploration of implementations of the $[\![5, 1, 3]\!]$ code. The poor performance mostly arises from incorrectly identified error syndromes, which is a result of the basic decoding using only a simple lookup table to match syndromes. It may be possible to more accurately estimate the probability of different errors given specific syndromes, in particular when also taking into consideration syndromes from previous cycles. An improved decoder of these bit strings would provide performance in between the basic and perfectly decoded simulation presented in fig. 4.4.

Modifications to the $[\![5, 1, 3]\!]$ code that introduce redundancy in the detection of error, such as including a fifth stabilizer generator $g_1\, g_2\, g_3\, g_4 = ZZXIX$, could prove more robust against decoding errors. Other small stabilizer codes with a higher degree of parallelism may also exist, which would be inherently more resistant to this phenomenon. Potentially, a combination of QEC and post-selection would be beneficial, correcting errors with a high probability of matching the given syndrome while post-selecting others. While such procedures would require more classical processing we have shown that an increased feedback time is not a major concern to lifetime.

An entirely alternate method of investigating error correcting codes could be possible utilizing post-processing. In such an implementation, errors would only be *tracked* and not corrected, and their effects would be reversed in processing of the experimental data. In appendix B.1 we show that, as for active error correction, every error can be uniquely identified even when the state is not actively stabilized provided that no more than one error occurs for every stabilizer cycle. This is true even if the final state contains errors of weight-2 or above. In appendix D.1, we demonstrate the equivalence to active error correction. Post-processing has the benefit of posing no restrictions on the possible decoding procedure, and could thus possess the capability of reaching the error threshold within our noise model. Although it may improve the logical lifetime, not actively correcting errors does have the downside of being unable to correctly perform two qubit gates between logical states.

However, to assess the implementation of a more sophisticated decoder a much more realistic model of noise needs to be considered. We have only modeled the coherent limit of noise using amplitude- and phase damping. Except for the time passed during gates, these operations function perfectly, unlike physical implementations. Multiple sources of noise, including flux noise, SPAM errors, ZZ crosstalk and in particular leakage have been shown to be particularly impactful [5]. Correlated errors may arise from the large number of CZ gates and leakage could give unidentifiable syndrome measurements. Both might result in weight-2 or greater errors decreasing the logical lifetime. Therefore such noise needs to be properly investigated, possibly by simulating the circuit at Hamiltonian level instead; simulating the physical system rather than the circuit it represents.

Bibliography

[1] P. W. Shor, "Polynomial-Time Algorithms for Prime Factorization and Discrete Logarithms on a Quantum Computer," *SIAM Journal on Computing*, vol. 26, no. 5, pp. 1484–1509, 10 1997. doi: 10.1137/s0097539795293172. [Online]. Available: http://dx.doi.org/10.1137/S0097539795293172

[2] A. Aspuru-Guzik, A. D. Dutoi, P. J. Love, and M. Head-Gordon, "Simulated Quantum Computation of Molecular Energies," *Science*, vol. 309, no. 5741, p. 1704, 9 2005. doi: 10.1126/science.1113479. [Online]. Available: http://science.sciencemag.org/content/309/5741/1704.abstract

[3] D. S. Abrams and S. Lloyd, "Simulation of Many-Body Fermi Systems on a Universal Quantum Computer," *Physical Review Letters*, vol. 79, no. 13, pp. 2586–2589, 9 1997. doi: 10.1103/PhysRevLett.79.2586. [Online]. Available: https://link.aps.org/doi/10.1103/PhysRevLett.79.2586

[4] J. Preskill, "Quantum Computing in the NISQ era and beyond," *Quantum*, vol. 2, pp. 79–79, 1 2018. doi: 10.22331/q-2018-08-06-79. [Online]. Available: https://arxiv.org/abs/1801.00862

[5] J. F. Marques, B. M. Varbanov, M. S. Moreira, H. Ali, N. Muthusubramanian, C. Zachariadis, F. Battistel, M. Beekman, N. Haider, W. Vlothuizen, A. Bruno, B. M. Terhal, and L. Dicarlo, "Logical-qubit operations in an error-detecting surface code," QuTech, Delft University of Technology, Delft, Tech. Rep., 2 2021. [Online]. Available: https://arxiv.org/abs/2102.13071

[6] C. K. Andersen, A. Remm, S. Lazar, S. Krinner, N. Lacroix, G. J. Norris, M. Gabureac, C. Eichler, and A. Wallraff, "Repeated quantum error detection in a surface code," *Nature Physics*, vol. 16, no. 8, pp. 875–880, 8 2020. doi: 10.1038/s41567-020-0920-y

[7] C. K. Andersen, A. Remm, S. Balasiu, S. Krinner, J. Heinsoo, J.-C. Besse, M. Gabureac, A. Wallraff, and C. Eichler, "Entanglement Stabilization using Parity Detection and Real-Time Feedback in Superconducting Circuits," *npj Quantum Information*, vol. 5, no. 1, 2 2019. doi: 10.1038/s41534-019-0185-4. [Online]. Available: https://arxiv.org/abs/1902.06946

[8] Z. Chen, K. J. Satzinger, J. Atalaya, A. N. Korotkov, A. Dunsworth, D. Sank, C. Quintana, M. McEwen, R. Barends, P. V. Klimov, S. Hong, C. Jones,

A. Petukhov, D. Kafri, S. Demura, B. Burkett, C. Gidney, A. G. Fowler, H. Putterman, I. Aleiner, F. Arute, K. Arya, R. Babbush, J. C. Bardin, A. Bengtsson, A. Bourassa, M. Broughton, B. B. Buckley, D. A. Buell, N. Bushnell, B. Chiaro, R. Collins, W. Courtney, A. R. Derk, D. Eppens, C. Erickson, E. Farhi, B. Foxen, M. Giustina, J. A. Gross, M. P. Harrigan, S. D. Harrington, J. Hilton, A. Ho, T. Huang, W. J. Huggins, L. B. Ioffe, S. V. Isakov, E. Jeffrey, Z. Jiang, K. Kechedzhi, S. Kim, F. Kostritsa, D. Landhuis, P. Laptev, E. Lucero, O. Martin, J. R. McClean, T. McCourt, X. Mi, K. C. Miao, M. Mohseni, W. Mruczkiewicz, J. Mutus, O. Naaman, M. Neeley, C. Neill, M. Newman, M. Y. Niu, T. E. O'Brien, A. Opremcak, E. Ostby, B. Pató, N. Redd, P. Roushan, N. C. Rubin, V. Shvarts, D. Strain, M. Szalay, M. D. Trevithick, B. Villalonga, T. White, Z. J. Yao, P. Yeh, A. Zalcman, H. Neven, S. Boixo, V. Smelyanskiy, Y. Chen, A. Megrant, and J. Kelly, "Exponential suppression of bit or phase flip errors with repetitive error correction," 2 2021. [Online]. Available: http://arxiv.org/abs/2102.06132

[9] M. Gong, X. Yuan, S. Wang, Y. Wu, Y. Zhao, C. Zha, S. Li, Z. Zhang, Q. Zhao, Y. Liu, F. Liang, J. Lin, Y. Xu, H. Deng, H. Rong, H. Lu, S. C. Benjamin, C.-Z. Peng, X. Ma, Y.-A. Chen, X. Zhu, and J.-W. Pan, "Experimental exploration of five-qubit quantum error correcting code with superconducting qubits," *National Science Review*, 7 2019. doi: 10.1093/nsr/nwab011. [Online]. Available: http://arxiv.org/abs/1907.04507

[10] J. Gambetta, M. Treinish, P. Nation, P. Kassebaum, D. M. Rodríguez, S. de la Puente González, qiskit-bot, S. Hu, K. Krsulich, L. Zdanski, J. Yu, D. McKay, J. Gomez, L. Capelluto, Travis-S-IBM, J. Gacon, lerongil, R. I. Rahman, S. Wood, J. Schwarm, L. Bello, M. GEORGE, M. Marques, O. C. Hamido, RohitMidha23, S. Dague, S. Garion, tigerjack, Y. Kobayashi, and abbycross, "Qiskit/qiskit: Qiskit 0.23.6," 2 2021. [Online]. Available: https://doi.org/10.5281/zenodo.4549740

[11] J. Koch, T. M. Yu, J. Gambetta, A. A. Houck, D. I. Schuster, J. Majer, A. Blais, M. H. Devoret, S. M. Girvin, and R. J. Schoelkopf, "Charge insensitive qubit design derived from the Cooper pair box," *Physical Review A*, vol. 76, no. 4, 2 2007. doi: 10.1103/PhysRevA.76.042319. [Online]. Available: http://arxiv.org/abs/cond-mat/0703002http://dx.doi.org/10.1103/PhysRevA.76.042319

[12] J. A. Schreier, A. A. Houck, J. Koch, D. I. Schuster, B. R. Johnson, J. M. Chow, J. M. Gambetta, J. Majer, L. Frunzio, M. H. Devoret, S. M. Girvin, and R. J. Schoelkopf, "Suppressing Charge Noise Decoherence in Superconducting Charge Qubits," *Physical Review B*, vol. 77, no. 18, 12 2007. doi: 10.1103/PhysRevB.77.180502. [Online]. Available: http://arxiv.org/abs/0712.3581

[13] M. Rol, F. Battistel, F. Malinowski, C. Bultink, B. Tarasinski, R. Vollmer, N. Haider, N. Muthusubramanian, A. Bruno, B. Terhal, and L. DiCarlo, "Fast, High-Fidelity Conditional-Phase Gate Exploiting Leakage Interference in Weakly Anharmonic Superconducting Qubits," *Physical Review Letters*, vol. 123, no. 12,

p. 120502, 9 2019. doi: 10.1103/PhysRevLett.123.120502. [Online]. Available: https://link.aps.org/doi/10.1103/PhysRevLett.123.120502

[14] S. S. Hong, A. T. Papageorge, P. Sivarajah, G. Crossman, N. Didier, A. M. Polloreno, E. A. Sete, S. W. Turkowski, M. P. da Silva, and B. R. Johnson, "Demonstration of a parametrically activated entangling gate protected from flux noise," *Physical Review A*, vol. 101, no. 1, p. 12302, 1 2020. doi: 10.1103/PhysRevA.101.012302. [Online]. Available: https://link.aps.org/doi/10.1103/PhysRevA.101.012302

[15] A. De and L. P. Pryadko, "Universal set of Dynamically Protected Gates for Bipartite Qubit Networks II: Soft Pulse Implementation of the [[5,1,3]] Quantum Error Correcting Code," *Physical Review A*, vol. 93, no. 4, 9 2016. doi: 10.1103/PhysRevA.93.042333. [Online]. Available: http://arxiv.org/abs/1509.01239

[16] T. J. Yoder and I. H. Kim, "The surface code with a twist," *Quantum*, vol. 1, p. 2, 4 2017. doi: 10.22331/q-2017-04-25-2. [Online]. Available: http://dx.doi.org/10.22331/q-2017-04-25-2

[17] R. Chao and B. W. Reichardt, "Quantum Error Correction with Only Two Extra Qubits," *Physical Review Letters*, vol. 121, no. 5, 8 2018. doi: 10.1103/physrevlett.121.050502. [Online]. Available: http://dx.doi.org/10.1103/PhysRevLett.121.050502

[18] S. B. Bravyi and A. Y. Kitaev, "Quantum codes on a lattice with boundary," 11 1998. [Online]. Available: http://arxiv.org/abs/quant-ph/9811052

[19] Y. Tomita and K. M. Svore, "Low-distance Surface Codes under Realistic Quantum Noise," *Physical Review A*, vol. 90, no. 6, 4 2014. doi: 10.1103/PhysRevA.90.062320. [Online]. Available: http://arxiv.org/abs/1404.3747

[20] M. A. Nielsen and I. L. Chuang, *Quantum Computation and Quantum Information*, 10th ed. Cambridge: Cambridge University Press, 2010. ISBN 9780511976667

[21] D. Gottesman, "An Introduction to Quantum Error Correction and Fault-Tolerant Quantum Computation," in *Quantum information science and its contributions to mathematics, Proceedings of Symposia in Applied Mathematics*, 4 2009, pp. 13–58. [Online]. Available: https://arxiv.org/abs/0904.2557

[22] J. Preskil, "Chapter 7. Quantum Error Correction," California, 1999. [Online]. Available: http://theory.caltech.edu/~preskill/ph229/notes/chap7.pdf

[23] P. Krantz, M. Kjaergaard, F. Yan, T. P. Orlando, S. Gustavsson, and W. D. Oliver, "A Quantum Engineer's Guide to Superconducting Qubits," 4 2019. doi: 10.1063/1.5089550. [Online]. Available: http://arxiv.org/abs/1904.06560

[24] J. Preskill, "Chapter 3. Foundations II: Measurement and Evolution," California,

10 2018. [Online]. Available: http://theory.caltech.edu/~preskill/ph219/chap3_15.pdf

[25] D. Coppersmith, "An approximate Fourier transform useful in quantum factoring," *arXiv preprint quant-ph/0201067*, 1 2002. [Online]. Available: http://arxiv.org/abs/quant-ph/0201067

[26] S. J. Devitt, W. J. Munro, and K. Nemoto, "Quantum error correction for beginners," *Reports on Progress in Physics*, vol. 76, no. 7, 2013. doi: 10.1088/0034-4885/76/7/076001

[27] S. Bravyi, M. Englbrecht, R. König, and N. Peard, "Correcting coherent errors with surface codes," *npj Quantum Information*, vol. 4, no. 1, 12 2018. doi: 10.1038/s41534-018-0106-y

[28] J. Ghosh, A. G. Fowler, and M. R. Geller, "Surface code with decoherence: An analysis of three superconducting architectures," *Physical Review A*, vol. 86, no. 6, 10 2012. doi: 10.1103/PhysRevA.86.062318. [Online]. Available: http://arxiv.org/abs/1210.5799

[29] J. Niwa, K. Matsumoto, and H. Imai, "Simulating the Effects of Quantum Error-correction Schemes," *arXiv preprint quant-ph/0211071*, 12 2018. [Online]. Available: https://arxiv.org/abs/quant-ph/0211071

[30] A. Lanka, Y. Wang, and P. Prabhu, "Testing of flag-based fault-tolerance on IBM quantum devices," *arXiv preprint arXiv:2011.03224*, 11 2020. [Online]. Available: http://arxiv.org/abs/2011.03224

[31] A. Bengtsson, P. Vikstål, C. Warren, M. Svensson, X. Gu, A. F. Kockum, P. Krantz, C. Križan, D. Shiri, I.-M. Svensson, G. Tancredi, G. Johansson, P. Delsing, G. Ferrini, and J. Bylander, "Improved Success Probability with Greater Circuit Depth for the Quantum Approximate Optimization Algorithm," *Physical Review Applied*, vol. 14, no. 3, p. 34010, 9 2020. doi: 10.1103/PhysRevApplied.14.034010. [Online]. Available: https://link.aps.org/doi/10.1103/PhysRevApplied.14.034010

[32] M. Ganzhorn, G. Salis, D. J. Egger, A. Fuhrer, M. Mergenthaler, C. Müller, P. Müller, S. Paredes, M. Pechal, M. Werninghaus, and S. Filipp, "Benchmarking the noise sensitivity of different parametric two-qubit gates in a single superconducting quantum computing platform," *Physical Review Research*, vol. 2, no. 3, p. 33447, 9 2020. doi: 10.1103/PhysRevResearch.2.033447. [Online]. Available: https://link.aps.org/doi/10.1103/PhysRevResearch.2.033447

[33] G. Li, Y. Ding, and Y. Xie, "Tackling the Qubit Mapping Problem for NISQ-Era Quantum Devices," in *Proceedings of the Twenty-Fourth International Conference on Architectural Support for Programming Languages and Operating Systems*, 9 2019, pp. 1001–1014. [Online]. Available: https://arxiv.org/abs/1809.02573

[34] S. Sivarajah, S. Dilkes, A. Cowtan, W. Simmons, A. Edgington, and R. Duncan,

"t|ket⟩: a retargetable compiler for NISQ devices," *Quantum Science and Technology*, vol. 6, no. 1, p. 014003, 11 2020. doi: 10.1088/2058-9565/ab8e92. [Online]. Available: https://iopscience.iop.org/article/10.1088/2058-9565/ab8e92

[35] B. Yang, R. Raymond, H. Imai, H. Chang, and H. Hiraishi, "Testing Scalable Bell Inequalities for Quantum Graph States on IBM Quantum Devices," *arXiv preprint arXiv:2101.10307*, 1 2021. [Online]. Available: https://arxiv.org/abs/2101.10307

[36] C. C. Bultink, T. E. O'Brien, R. Vollmer, N. Muthusubramanian, M. W. Beekman, M. A. Rol, X. Fu, B. Tarasinski, V. Ostroukh, B. Varbanov, A. Bruno, and L. DiCarlo, "Protecting quantum entanglement from leakage and qubit errors via repetitive parity measurements," *Science Advances*, vol. 6, no. 12, p. eaay3050, 3 2020. doi: 10.1126/sciadv.aay3050. [Online]. Available: http://advances.sciencemag.org/content/6/12/eaay3050.abstract

[37] J. J. Burnett, A. Bengtsson, M. Scigliuzzo, D. Niepce, M. Kudra, P. Delsing, and J. Bylander, "Decoherence benchmarking of superconducting qubits," *npj Quantum Information*, vol. 5, no. 1, p. 54, 2019. doi: 10.1038/s41534-019-0168-5. [Online]. Available: https://doi.org/10.1038/s41534-019-0168-5

[38] A. Osman, J. Simon, A. Bengtsson, S. Kosen, P. Krantz, D. P. Lozano, M. Scigliuzzo, P. Delsing, J. Bylander, and A. Fadavi Roudsari, "Simplified Josephson-junction fabrication process for reproducibly high-performance superconducting qubits," *Applied Physics Letters*, vol. 118, no. 6, p. 064002, 2 2021. doi: 10.1063/5.0037093. [Online]. Available: https://doi.org/10.1063/5.0037093

[39] D. C. McKay, C. J. Wood, S. Sheldon, J. M. Chow, and J. M. Gambetta, "Efficient Z-gates for quantum computing," *Physical Review A*, vol. 96, no. 2, p. 22330, 8 2017. doi: 10.1103/PhysRevA.96.022330. [Online]. Available: https://link.aps.org/doi/10.1103/PhysRevA.96.022330

[40] T. E. O'Brien, B. Tarasinski, and L. DiCarlo, "Density-matrix simulation of small surface codes under current and projected experimental noise," 3 2017.

[41] N. Ofek, A. Petrenko, R. Heeres, P. Reinhold, Z. Leghtas, B. Vlastakis, Y. Liu, L. Frunzio, S. M. Girvin, L. Jiang, M. Mirrahimi, M. H. Devoret, and R. J. Schoelkopf, "Extending the lifetime of a quantum bit with error correction in superconducting circuits," *Nature*, vol. 536, no. 7617, 8 2016. doi: 10.1038/nature18949

A

Gate operations

This Appendix contains full descriptions of all quantum gates used in the report, following the notations of Nielsen & Chuang [20]. The most basic ones are the Pauli operators, already defined in eq. (2.1) as

$$X \equiv \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \equiv -\boxed{X}-$$

$$Y \equiv \begin{bmatrix} 0 & -\mathrm{i} \\ \mathrm{i} & 0 \end{bmatrix} \equiv -\boxed{Y}-$$

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

which all rotate the qubit state around their respective axis of the Bloch sphere by an angle π. Similarly the $\sqrt{\mathbf{X}}$ and $\sqrt{\mathbf{Y}}$ gates rotate by $\pi/2$ instead

$$\sqrt{X} \equiv \frac{1}{2}\begin{bmatrix} 1+\mathrm{i} & 1-\mathrm{i} \\ 1-\mathrm{i} & 1+\mathrm{i} \end{bmatrix} \equiv -\boxed{\sqrt{X}}-$$

$$\sqrt{Y} \equiv \frac{1}{2}\begin{bmatrix} 1+\mathrm{i} & -1-\mathrm{i} \\ 1+\mathrm{i} & 1+\mathrm{i} \end{bmatrix} \equiv -\boxed{\sqrt{Y}}-$$

with, of course, $\sqrt{X}^2 = X$ and $\sqrt{Y}^2 = Y$. Additionally the Z can be generalized to an arbitary rotation around the Z-axis

$$Z(\theta) \equiv \begin{bmatrix} 1 & 0 \\ 0 & e^{\mathrm{i}\theta} \end{bmatrix} \equiv -\boxed{Z(\theta)}- \equiv -\boxed{R_Z(\theta)}- \equiv -\boxed{U_1(\theta)}-$$

also referred to as $U_1(\theta)$, $R_Z(\theta)$. The final gate used is the Hadamard gate

$$H \equiv \frac{X+Z}{\sqrt{2}} \equiv \frac{1}{\sqrt{2}}\begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \equiv -\boxed{H}-$$

which notably moves $|0\rangle$ ($|1\rangle$) to $|+\rangle$ ($|-\rangle$).

Along these single qubit gates, we also use a number of two-qubit gates. The most common ones are the controlled-X (controlled NOT) and controlled-Z gate

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

$$CZ \equiv \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & -1 \end{bmatrix} \equiv$$ 

applying the X or Z gate on the target qubit (bottom) if the control qubit (top) is in $|1\rangle$. Note that these are not completely equivalent to the classical CNOT counterpart, as even the control qubits state can change by it.

The final gates we consider are the SWAP and iSWAP. Their most notable feature is effectively swapping the quantum states of the involved qubits. Its represented as

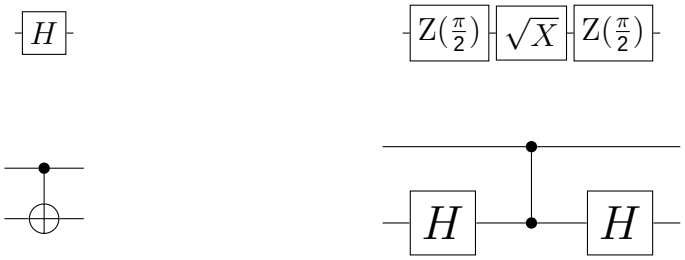
$$\text{SWAP} \equiv \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \equiv$$

$$\text{iSWAP} \equiv \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & i & 0 \\ 0 & i & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \equiv$$ 

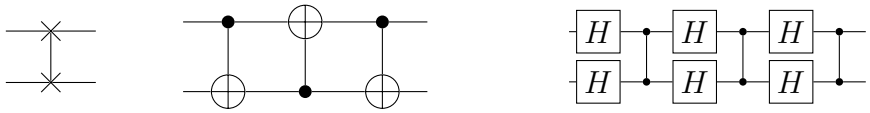
which can be interpreted as replacing $|10\rangle$ with $|01\rangle$ and vice versa, leaving $|00\rangle$ and $|11\rangle$ unaffected. The main difference between the gates is that iSWAP adds a phase i when states are swapped.

A.1 Gate equivalences

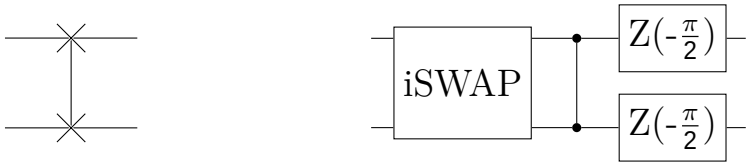
This section shows all gate decompositions utilized in transpilation of our circuits. These compositions are all equivalent, as in they give the same unitary matrix. As such the decompositions works both ways if necessary. The following ones were already included in Qiskit's equivalence library by default.



For replacing SWAP gates, there was no default equivalence well suited for our gate set. Instead the decomposition present used a two step process of turning SWAP into CX and then further into CZ and Hadamard gates.



While this allows circuits to run with CZ as the only two-qubit gate, a more efficient decomposition was implemented.



B

Error syndromes

This Appendix contains a list of correction procedures corresponding to specific outcomes when measuring the stabilizer generators for the $[\![5, 1, 3]\!]$ code. We will follow the convention of labeling the measurement outcomes with a binary number, where "0" corresponds to the eigenvalue $+1$ and "1" corresponds to the eigenvalue -1. We measure the generators in the order shown in eq. (2.4), and produce a 4-digit binary number from right to left. For the case of not resetting the ancilla between measurements, an error causes the state of the ancilla qubit to "flip". E.g., the syndrome "1100" without resets is becomes "0100" with resets. This carries over to the following cycle of measurements and so there are two different cases; whether the ancilla ended the previous cycle in the $|0\rangle$ or $|1\rangle$ state. If the ancilla starts in the $|1\rangle$ state at the start of cycle (instead of $|0\rangle$), the entire syndrome is flipped.

Error syndrome	**Corrective operation**		
	Reset	No reset, start in $\|0\rangle$	No reset, start in $\|1\rangle$
0000	$IIIII$	$IIIII$	$IXIII$
0001	$IXIII$	$IIXII$	$IIIIZ$
0010	$IIIIZ$	$IIIXI$	$IIYII$
0011	$IIXII$	$ZIIII$	$IIZII$
0100	$IIZII$	$IIIIX$	$YIIII$
0101	$ZIIII$	$IIIYI$	$IIIIY$
0110	$IIIXI$	$IZIII$	$IYIII$
0111	$IIYII$	$IIIZI$	$XIIII$
1000	$XIIII$	$XIIII$	$IIIZI$
1001	$IIIZI$	$IYIII$	$IZIII$
1010	$IZIII$	$IIIIY$	$IIIYI$
1011	$IYIII$	$YIIII$	$IIIIX$
1100	$IIIIX$	$IIZII$	$ZIIII$
1101	$YIIII$	$IIYII$	$IIIXI$
1110	$IIIIY$	$IIIIZ$	$IIXII$
1111	$IIIYI$	$IXIII$	$IIIII$

B.1 Post-processing

It is possible to identify multiple single qubit errors given that they occur no more than once per stabilizer cycle. This can be done as any generator can be dynamically redefined with a -1 phase to switch its eigenvalues so that the post-error state lies within a new codespace. This will update the above table to identify deviations from the new logical subspace.

Given that one error E_i with syndrome $\boldsymbol{\beta_i} = \{\beta_l\}_{l=0}^n$ has occurred. What is the syndrome measured if another error E_j with $\boldsymbol{\beta_j} = \{\beta_l\}_{l=0}^n$ occurs? By applying eq. (2.3) twice we get

$$g_l E_i E_j = \beta_l E_i g_l E_j = \beta_l \beta_l E_i E_j g_l,$$

hence, we simply get the element wise multiple of $\boldsymbol{\beta}_i$ and $\boldsymbol{\beta}_j$, or if they are stored as strings of bits, the bit-wise XOR.

A demonstration of the equivalence to active error correction can be found in appendix D.1.

C

Circuits

Here we show all different circuits used throughout simulations and results presented in the report. Circuits already shown as part of the main text are still included here for full clarity. In general, D denotes data qubits encoding the logical state $|\psi_L\rangle$ and A the ancilla used for measurements. Indexes represent the vertices the position of qubits on the layouts in fig. 3.4. Note that figures here still use the Hadamard gate, despite it not being native to our gate set. This is due to visual clarity. Its decomposition in appendix A still has identical gate time due to the virtual Z-gate[39].

C.1 Encoding circuits

This section details the circuits used for encoding and are in the following order: Full connectivity, hexagonal layout (with and without iSWAP) and square layout (with and without iSWAP). While the ancilla generally is in node 6 in these circuits, the final step of encoding swaps it into the central node (denoted 0). This is not needed in the full connectivity case, as all qubits can interact with each other.

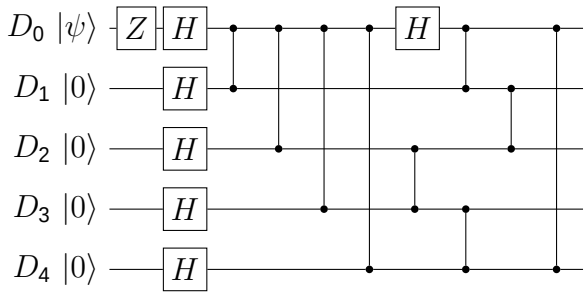


Figure C.1: *Encoding circuit with full connectivity*

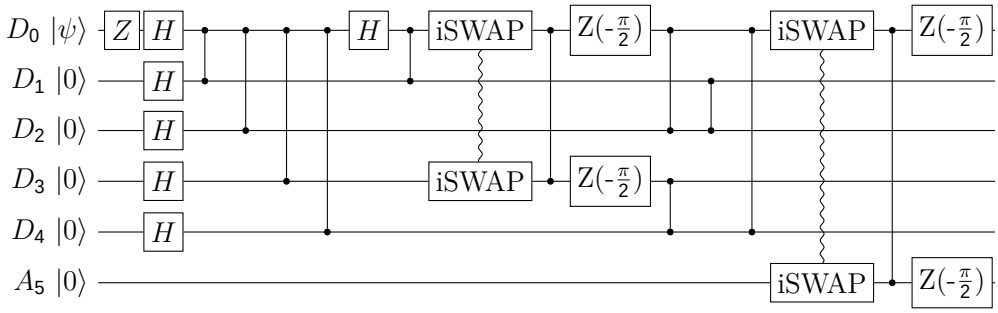


Figure C.2: *Encoding, hexagonal connectivity*

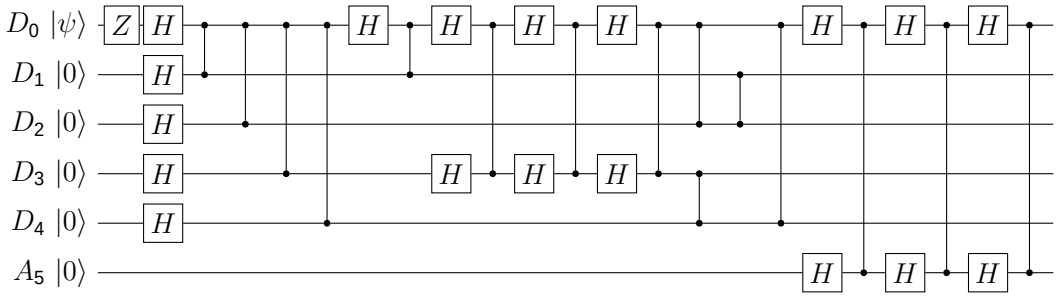


Figure C.3: *Encoding without iSWAP, hexagonal connectivity*

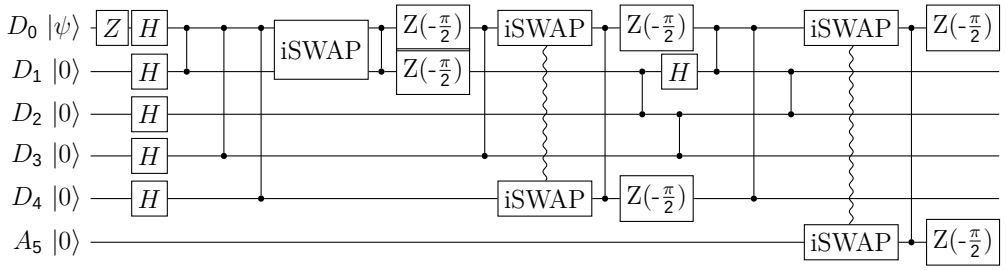


Figure C.4: *Encoding, square connectivity.*

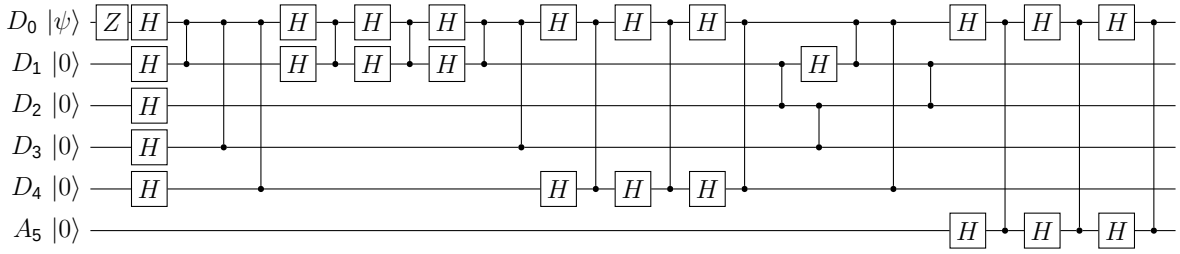


Figure C.5: *Encoding without iSWAP, square connectivity.*

C.2 Stabilizer circuits

Following the previous section, we list all different circuits used for stabilizer measurements. They are listed in the following order: Standard, transpiled hexagonal, pipelined and transpiled square grid.

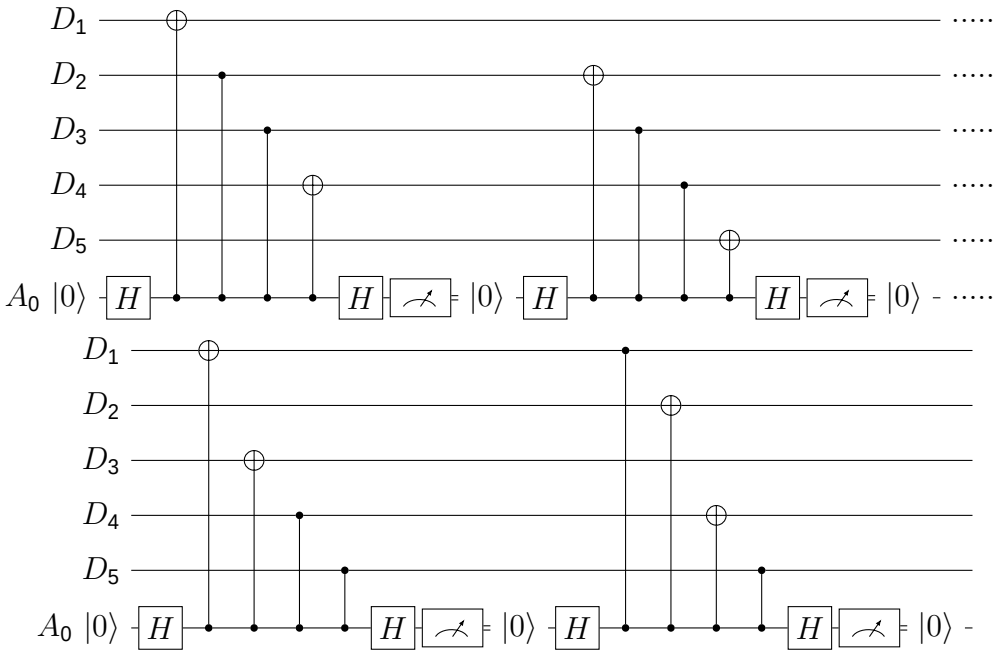


Figure C.6: *Full stabilizer cycle before any transpilation.*

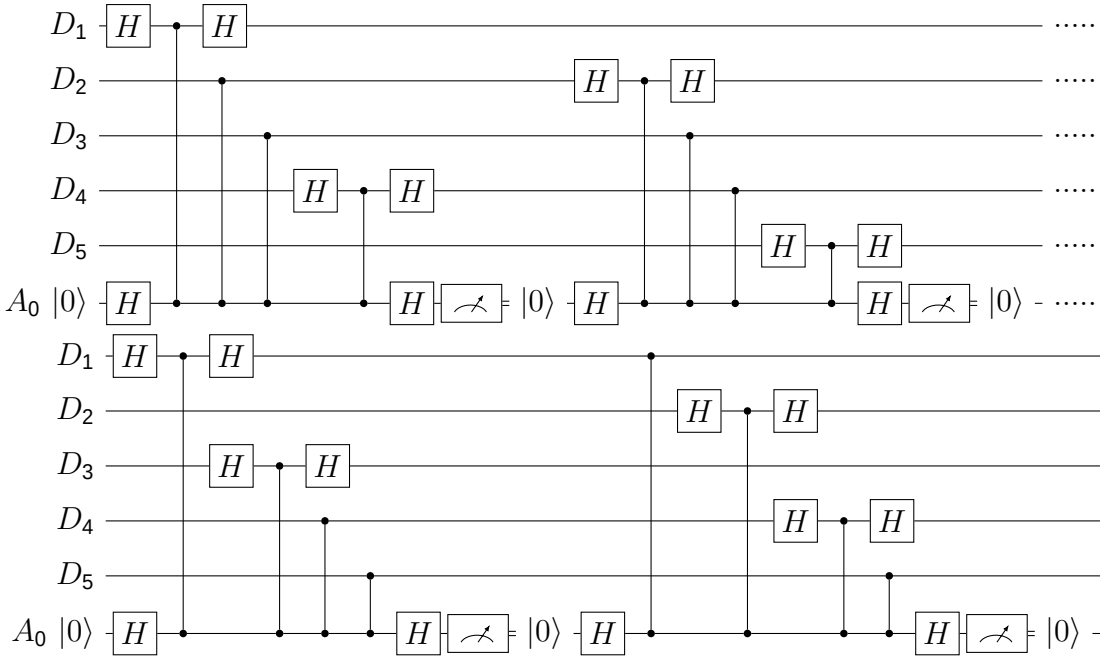


Figure C.7: *Transpiled stabilizer cycle, full or hexagonal connectivity.*

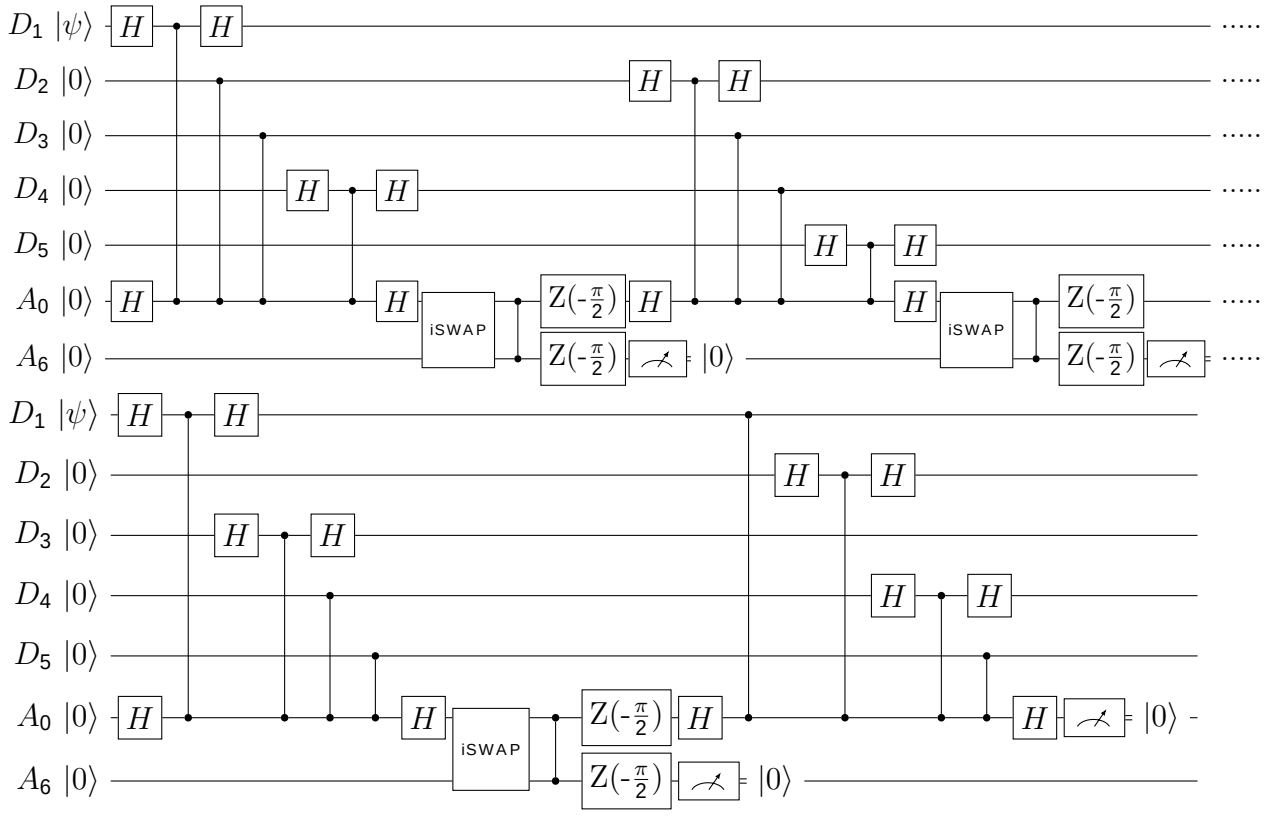


Figure C.8: *Pipelined stabilizer cycle following hexagonal connectivity. No swapping is needed for the final stabilizer as it must wait for classical feedback regardless.*

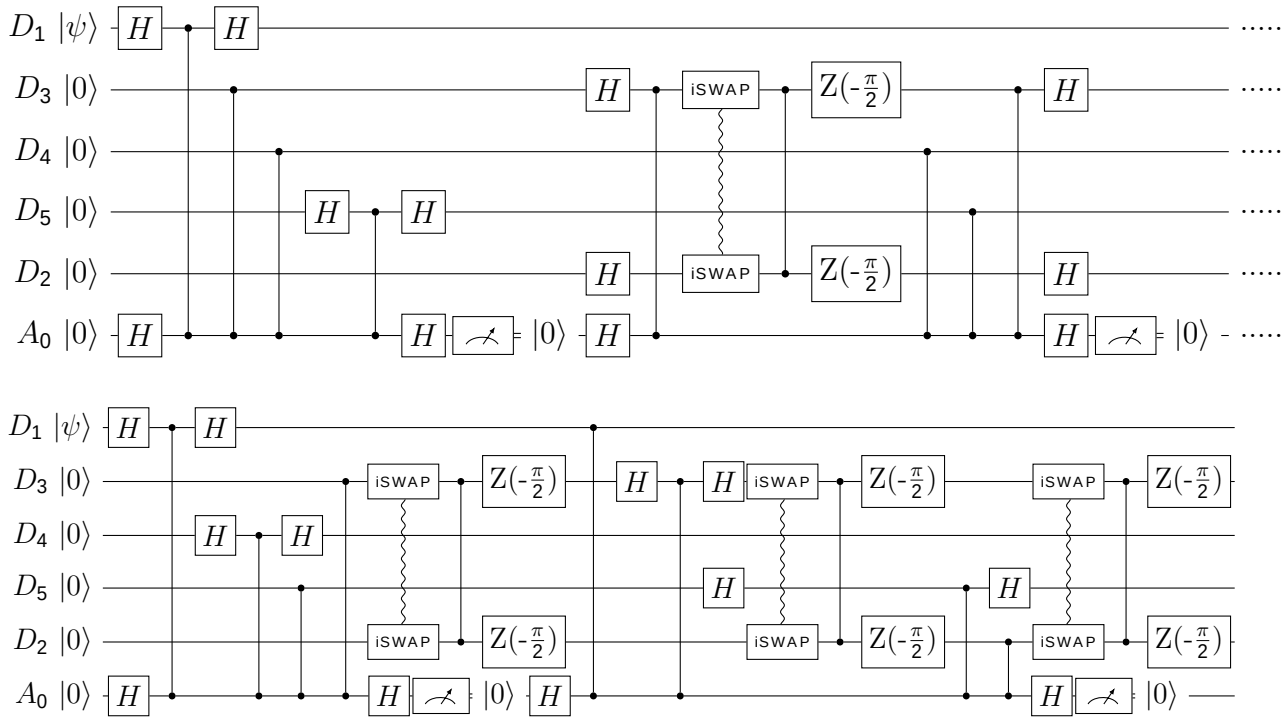


Figure C.9: *Transpiled stabilizer cycle, square connectivity. Note the permutations of data qubits to match the necessary connectivity.*

D

Supplemental data

This appendix contains further details regarding ancilla resetting, lifetime of different logical states and full data from figures used in the main text. While not necessary for the conclusions reached, they support some assumptions made throughout simulations.

D.1 Ancilla strategy and post-processing

Due to limitations in Qiskit, there are difficulties in implementing the QEC procedure without resetting the ancilla between stabilizer measurements. As such, we have limited ourselves to perfectly resetting after each measurement for our main results. Here we will show that this has no significant impact on the results, using *post-processing*. Instead of performing real-time error correction, the errors are instead tracked across stabilizer cycles and a full correction for all errors detected is applied at the end. The details of this procedure is given in appendix B.1, and its results are shown in fig. D.1 comparing real-time QEC and post-processing both with and without resets. Across all runs the standard gate times and parameters in section 3.3 were used giving a logical lifetime of 26 ± 1, 26 ± 1 and 27 ± 1 µs respectively. By this, we see that resetting does not significantly alter our results in the main report, and can be omitted. It also verifies the equivalence of post-processing to active error correction.

D.2 Alternate definitions of logical lifetime

Other works have used the exponential decay of varying metrics to determine the logical lifetime. As the full density matrix is challenging to determine in experimental setups[1], an alternative sometimes used is to measure the expectation value of the logical operators $Z_L = ZZZZZ$ or $X_L = XXXXX$ [6][5]. These can be directly calculated by measuring the qubits individually (after applying a Hadamard in the case of X_L).

[1] In physical experiments one can never measure two non-commuting observables without one measurement influencing the other. The full density matrix can still be retrieved through a process called state tomography, in which you measure a complete set of observables multiple times on a set of identically prepared systems, from which you can reconstruct the density matrix describing the ensemble of their states. This process is usually destructive however, and thus it can only be performed at the end of the experiment. This is still is a common way of validating the fidelity of QEC in experimental setups [6][5][41].

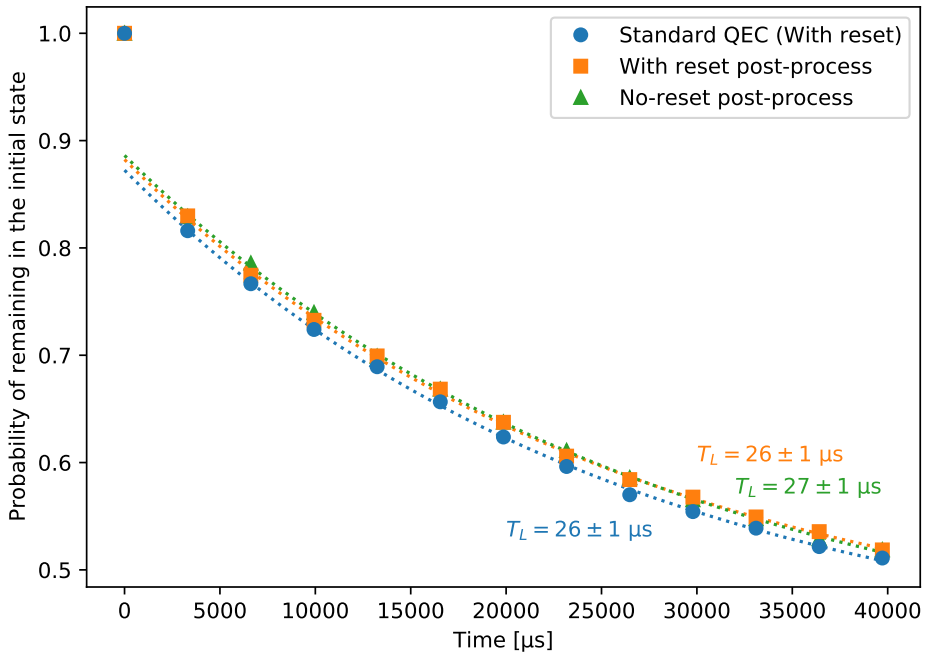


Figure D.1: *Comparison of state fidelity across stabilizer cycles using either real-time QEC (blue circles), post-processing with ancilla reset after each measurement (orange squares) and post-processing without ancilla resets (green triangles). All simulations were performed using standard gate times definer in section 3.3 along with T_1 = 40 µs and T_2 = 60 µs. Dotted lines represent exponential fits, with logical lifetime T_L defined as the decay.*

In fig. D.2, a comparison of using state fidelity and $\langle Z_L \rangle$ is made. The resulting logical lifetimes are identical, this is true also for the expectation value of the X_L observable for the $|+_L\rangle$ and $|-_L\rangle$ states although that is not shown here. Looking at the difference in expectation values for the $|0_L\rangle$ and $|1_L\rangle$, we can directly observe how distinguishable these codewords become over time. We see that the distinguishability roughly corresponds to the fidelity of the $|0_L\rangle$ state over time, which corroborate the use of state fidelity.

An approach more general than looking at the fidelity of the quantum state after correction is to model the QEC procedure as a quantum channel using quantum process tomography [20, Sec. 8.4.2], and compare its fidelity to the identity channel. This definition is entirely independent of initial conditions and has been favored in other works [41]. We determined it to be unnecessarily costly to implement in our simulations as we anticipate that it will give a very similar result to our simpler formula.

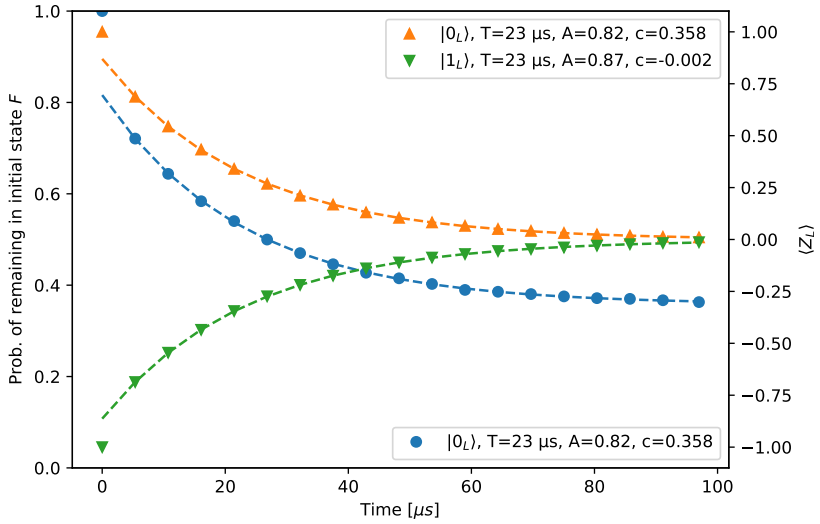


Figure D.2: *Comparison of state fidelity and expectation value of the Z_L operator as metrics for logical qubit lifetime. We perform repeated rounds of error correction and fit the curve $(A-c)e^{-t/T}+c$ to the di erent metrics for state preservation. Blue • shows the fidelity of the $|0_L\rangle$ state to its initial condition over time on the left y-axis, The orange* N *and green* H *show the expectation value of the Z_L observable for the $|0_L\rangle$ and $|1_L\rangle$ states over time respectively on the right y-axis. We find the same logical lifetime.*

DEPARTMENT OF PHYSICS
CHALMERS UNIVERSITY OF TECHNOLOGY
Gothenburg, Sweden
www.chalmers.se

CHALMERS
UNIVERSITY OF TECHNOLOGY