UC Santa Barbara
UC Santa Barbara Electronic Theses and Dissertations

Title

Resolving correlated errors in superconducting quantum computers

Permalink

https://escholarship.org/uc/item/17z8n5gx

Author

McEwen, Matthew James

Publication Date

2022

Peer reviewed|Thesis/dissertation

eScholarship.org Powered by the California Digital Library
University of California

University of California
Santa Barbara

Resolving correlated errors in superconducting quantum computers

A dissertation submitted in partial satisfaction
of the requirements for the degree

Doctor of Philosophy
in
Physics

by

Matthew James McEwen

Committee in charge:

Professor Ben Mazin, Chair
Professor Chetan Nayak
Professor Ania Jayich

December 2022

The Dissertation of Matthew James McEwen is approved.

Professor Chetan Nayak

Professor Ania Jayich

Professor Ben Mazin, Committee Chair

October 2022

Resolving correlated errors in superconducting quantum computers

Copyright © 2022

by

Matthew James McEwen

Curriculum Vitæ
Matthew James McEwen

Education

2018 – 2022	Doctor of Philosophy in Physics (Expected), Masters of Arts in Physics, University of California, Santa Barbara.
2012 – 2016	Bachelor of Science in Physics, Bachelor of Computer Engineering (Honours), University of New South Wales, Sydney, Australia

Employment

2018 – 2022	Student Researcher, Google Quantum Hardware, Santa Barbara
2014 – 2017	Research Assistant, Center for Excellence in Quantum Computing and Communication Technology, University of New South Wales, Sydney, Australia
2016 – 2017	Cybersecurity consultant, PwC, Threat and Vulnerability Management, Sydney, Australia

Selected Publications

[1] Matt McEwen et al. “Removing leakage-induced correlated errors in superconducting quantum error correction”. In: *Nature Communications* 12.1 (Dec. 2021), p. 1761. DOI: `10.1038/s41467-021-21982-y`.

[2] Google Quantum AI et al. “Exponential suppression of bit or phase errors with cyclic error correction”. In: *Nature* 595.7867 (July 15, 2021). _eprint: 2102.06132, pp. 383–387. DOI: `https://doi.org/10.1038/s41586-021-03588-y`.

[3] Matt McEwen et al. “Resolving catastrophic error bursts from cosmic rays in large arrays of superconducting qubits”. In: *Nature Physics* 18.1 (Jan. 2022), pp. 107–111. DOI: `10.1038/s41567-021-01432-8`.

[4] Google Quantum AI et al. “Suppressing quantum errors by scaling a surface code logical qubit”. In: (2022). DOI: `10.48550/ARXIV.2207.06431`.

[5] Craig Gidney, Michael Newman, and Matt McEwen. “Benchmarking the Planar Honeycomb Code”. In: *Quantum* 6 (Sept. 21, 2022), p. 813. DOI: `10.22331/q-2022-09-21-813`.

[6] Kevin C. Miao and Matt McEwen. “Complete Leakage Removal in the Surface Code on Superconducting Qubits”. In: *Submitted Nature Physics* (2022).

Abstract

Resolving correlated errors in superconducting quantum computers

by

Matthew James McEwen

Quantum computers can provide new computational abilities, but only if their intrinsic noise can be suppressed. Quantum error correction (QEC) promises to suppress errors exponentially, provided they are sufficiently uncorrelated. Large arrays of superconducting qubits are a leading platform for implementing quantum error correction, but feature several sources of error that are highly correlated; a single physical process that produces the equivalent of many independent errors to be corrected. These correlated errors must be studied and mitigated in order for quantum error correction to succeed. This thesis addresses two correlated error sources in particular; leakage and impacts from high-energy radiation. Leakage of information out of the states selected for computation presents a significant challenge, as leakage populations spread virally through the device and induce a large pattern of errors if they are not suppressed. We develop several new techniques for removing leakage during QEC codes, eventually achieving regular leakage removal from all qubits, successfully curtailing the ability of leakage to spread. High-energy radiation impacting the device present another problematic error source, as the energies deposited are enough to cause significant error over the entire chip at current sizes. We present time-resolved measurements of such impacts and explain the underlying physical processes with a view toward future mitigation of this error source in hardware. This work understanding and suppressing these correlated errors in our hardware has run parallel to and been integrated into the development of the first scaling demonstrations of surface code quantum error correction.

Contents

Curriculum Vitae **iv**

Abstract **v**

List of Figures **ix**

1 Introduction and Background **1**

1.1 Why bother with Quantum Computing? 1
1.2 Quantum Computing Basics 7
1.3 Error Correction 12
1.4 The Surface Code 14
1.5 Error Models 18
1.6 Decoding 22
1.7 Correlated Errors 25

2 Superconducting Quantum Devices **34**

2.1 Superconductivity 35
2.2 Microwave Circuit QED 37
2.3 Junctions and Squids 38
2.4 The Transmon 40
2.5 Device Architecture and System Architecture 44
2.6 Hardware Correlated Errors 46

3 Multilevel Reset **53**

3.1 Abstract 53
3.2 Introduction 54
3.3 Reset Gate Theory and Implementation 55
3.4 Experimental realization of the Multilevel Reset Gate 58
3.5 Integrating Reset into the Bit-flip Code 60
3.6 Injection of Leakage 64
3.7 Leakage-induced Correlations in the Bit-flip Code 65
3.8 Logical performance and Λ_{bit} 66

4 Complete Leakage Removal 76
4.1 Abstract 76
4.2 Introduction 77
4.3 Characterizing the spread of leakage 79
4.4 Suppressing leakage populations during a code circuit 83
4.5 Effect on QEC logical performance 87
4.6 Summary and Outlook 92

5 High-Energy Impacts 99
5.1 Abstract 99
5.2 Introduction 100
5.3 Rapid Repetitive Correlated Sampling 103
5.4 Timing of Events and Independent Background Error 105
5.5 Impact Localization and Evolution 107
5.6 Quasiparticle Signature and Event Magnitude 111
5.7 Discussion 112

6 Walking Surface Codes 119
6.1 Introduction 119
6.2 Detecting Regions 121
6.3 Step Code 129
6.4 Walking Surface Code 130
6.5 Benchmarking 132
6.6 Use Cases 137
6.7 Outlook 140

A Supplementary Information regarding Multilevel Reset 144
A.1 Reset Gate Parameters 144
A.2 Leakage Error and suppression 146
A.3 Leakage accumulation during the bit-flip code 151
A.4 The $\mathbf{p_{ij}}$-matrix 154
A.5 The checkerboard pattern in the $\mathbf{p_{ij}}$-matrix 155
A.6 Statistics and postselection in the bit-flip code 156
A.7 Subsampling for analysis of scaling performance 157

B Supplementary Information regarding Complete Leakage Removal 159
B.1 Effects of leakage on diabatic CZ gate 159
B.2 Leakage Removal Strategy Details 161
B.3 Effect of reset strategies on leakage dynamics 163
B.4 Effect of reset strategies on QEC error detection 165
B.5 Fitting techniques for logical error probability 166
B.6 Surface code simulations well below threshold 167
B.7 Error correlations in the surface code experiment 168

C Supplementary Information regarding High-Energy Impacts 178
C.1 Energy cascade following a high energy impact 178
C.2 Absence of Excitation Errors 185
C.3 Peak-finding and Event Parameter Distributions 187
C.4 Further Events with High Time Resolution 192

List of Figures

1 Introduction and Background

1.1 Stabilizers of the surface code. 15

2 Superconducting Quantum Devices

3 Multilevel Reset

3.1 Removing leakage with reset. 56
3.2 Reset gate benchmarking. 59
3.3 Leakage during the Bit-flip code. 62
3.4 Injection of leakage. 63
3.5 Correlations caused by leakage 67
3.6 Logical code performance. 68

4 Complete Leakage Removal

4.1 Leakage in a structured QEC circuit. 80
4.2 Leakage transport and phase errors in CZ gates. 81
4.3 Leakage population during code execution. 85
4.4 Bit-flip code logical performance and dependence on injected errors. . . 88
4.5 Surface code logical performance and dependence on injected errors. . . 91

5 High-energy Impacts

5.1 Rapid repetitive correlated sampling. 101
5.2 Identifying events and background error. 106
5.3 Localization and spread of error. 108
5.4 Extracting energy decay times during events. 110

6 Walking Surface Codes

6.1 Mid-cycle states in the surface code. 122
6.2 Simple detecting regions. 124
6.3 Detecting regions in the repetition codes. 125
6.4 Overlapping detecting regions in the repetition code. 126
6.5 Circuits and detecting regions in the Step Code. 128

6.6 Circuit and detecting regions for a standard surface code cycle. 131
6.7 How to step the surface code. 133
6.8 Circuit and detecting regions for a stepping surface code cycle. 134
6.9 Four behaviours for a patch possible using the stepping circuit. 135
6.10 Threshold plots for walking codes. 138
6.11 Fan plots for walking codes. 138
6.12 Sliding for simultaneous surface code patch movements. 141

A Supplementary Information regarding Multilevel Reset

A.1 Dependence on swap frequency. 147
A.2 Dependence on adiabatic slope parameter. 148
A.3 Computational vs. Leakage error. 149
A.4 Readout for distinguishing leakage. 150
A.5 Leakage Populations during bit-flip code. 153
A.6 Odd-even periodicity of energy relaxation events. 156

B Supplementary Information regarding Complete Leakage Removal

B.1 Measuring leakage transport. 171
B.2 Measuring leakage phases. 172
B.3 Cross-entropy benchmarking (XEB) of DQLR operation. 173
B.4 Leakage dynamics in a surface code with different removal strategies. . . 174
B.5 Logical performance of bit-flip code in time. 175
B.6 Distance-3 surface code detection probabilities. 176
B.7 Surface code p_{ij} correlations. 177

C Supplementary Information regarding High-Energy Impacts

C.1 Decay vs Excitation Errors. 186
C.2 Peak-finding procedure. 188
C.3 Impact Parameter distributions. 189
C.4 Further highly time-resolved events. 190

Chapter 1

Introduction and Background

Quantum computing as a field has built a remarkable reputation for being hard to understand, certainly aided by the proliferation of different and sometimes misleading ways it is explained. In this chapter, we add another introductory explanation to the pile, more in the hopes that it will resonate with some readers than that it covers the the full technical background[1].

1.1 Why bother with Quantum Computing?

Computation is the backbone of modern society. Each time humanity has increased our ability to do it, momentous changes have followed. Our ability to do computation can be seen as a vast resource that we spend offloading and automating repetitive tasks, communicating with each other, and most generally not having to do the equivalent mechanical thinking ourselves, freeing us to do more interesting thinking.

Since the 1930s, when humanity began collectively grappling with the powers and limits of computation, we have broadly had the same understanding of what it is that

---

[1] For those interested in the full technical treatment, I recommend a Ph.D in Physics.

computers can do for us. These limits are embodied in the *Church-Turing Thesis*, a statement that colloquially boils down to “If something can be computed at all, it can be computed by the kinds of computers we know how to design and build”. In the decades since then, computers have become smaller, faster and more sophisticated at a remarkable pace, hugely improving the practical limits of what we can do with them. This inspired the *Extended Church-Turing Thesis*, colloquially “If something is computable *efficiently* at all, it can be computed *efficiently* by the kinds of computers we know how to design and build.”[2] To a human with a finite lifespan (who is not a computer scientist) the extended thesis is perhaps the more relevant one.

In the 1970s, by which time we collectively had a much better handle on quantum mechanics, Richard Feynman (in)famously pointed out that the act of simulating a quantum mechanical system seemed to be outside the realm of what was efficiently computable, at least with classical computers. Meanwhile, Nature didn’t seem to mind that computers couldn’t keep up, and quantum mechanical systems continued to ‘know what to do next’ regardless. Rather than being a problem, this represents a possibility; if whatever a quantum mechanical system is up to could be harnessed to do computation for us, it would seem to provide access to efficient computational abilities that a classical computer could not have. This possibility violates the Extended Church-Turing thesis; in the colloquial language above “If something can be computed efficiently, it might be doable with the computers we have, but it might need a quantum computer to do it, which we don’t have.”

This gives a flavour of the transformative addition of quantum mechanics to computers: it provides intrinsic new abilities that turn some possible computations from ‘impractical’ to ‘practical’, provided you can build the machine to do it. Such a qualitative

---

[2] Here, ‘computable efficiently’ is being used here as if it means ‘computable in polynomial time’. A physicist might complain that efficiently could be better used to indicate ‘computable over the weekend / my lifetime / the age of the universe’. I invite any such physicist to read [7].

change is embodied in the *Quantum Church-Turing Thesis*, "If something is computable efficiently at all, it can be computed *efficiently by a quantum computer*." The remaining challenge is for us to be able to add "the kind we know how to design and build."

1.1.1 But really, why bother?

The above story is a little abstract, so it is worth reflecting on exactly which problems a quantum computer might allow practical access to that we didn't have before. These are bound up with the development of *quantum algorithms*, procedures for computing something that works on a quantum computer efficiently, with the typical implication that they don' work efficiently on a classical computer. This implication is important; there have already been several examples of a quantum algorithm being invented and later *de-quantized* so that the problem could be solved efficiently on a classical computer [8, 9, 10]. (While this process makes everyone better off, it does not justify building a quantum computer.)

Historically, the most important quantum algorithm that has been developed is *Shor's Algorithm* [11], which enables the efficient computation of the prime factors of an integer. This problem has no known efficient classical algorithm, and we collectively had enough faith that one would not be found that this problem underpins much of modern cryptography and internet security. This was therefore a particularly surprising algorithm to find. The underlying technique in Shor's algorithm, the Quantum Fourier Transform, is quite generally applicable to problems involving analysing repetitive behaviour. This algorithm justified the construction of quantum computers by providing a specific and difficult mathematical problems that we could now compute efficiently. This is exciting if you're a mathematician or computer scientist - if the last few decades have taught us anything, when the computer scientists are excited we should all be excited, or at least wary. However, it is reasonably to ask if there's something more directly exciting that a quantum computer enables us to do.

The next major milestone in quantum algorithms is *Grover's Algorithm* [12], which changes from being much more efficient at a particularly hard problem to being somewhat better at 'basically anything if you don't think about it'. This algorithm concerns black-box search problem; Grover's algorithm applies to any problem where you're capable of recognising the right answer when you find it. However, while it solves this problem faster than any classical algorithm, it does not improve things enough to become 'efficient'. Grover's algorithm illustrates that even with no extra information about the problem to grasp onto, a quantum computer is better than a classical one. However, this also demonstrates that a quantum computer is not magic; coming up with a quantum algorithm to make a previously impractical problem efficiently computable requires looking into the details of the problem itself.

Perhaps the most general quantum algorithm that demonstrates such a speed up is the *HHL algorithm* [13]. This algorithm lets you inspect the answer to the solution to any linear system of equations efficiently[3], whereas classically this cannot be done efficiently. This algorithm also has the remarkable property that it is 'the hardest things a quantum computer can do'[4]. Any other task that a quantum computer can compute efficiently can be efficiently transformed into the task that the HHL algorithm addresses; if you can only pick one quantum algorithm to perform, this is the right one to pick. The ubiquity of linear systems of equations throughout science and engineering gives this algorithm the potential for extremely widespread use. However, while the use case here is perhaps near universal, it'd be nice to have a exciting story of how to impact people's lives by solving hard problems, rather than reducing the computing budget of the average engineering

---

[3] More technically, the algorithm estimates the results of applying scalar measurements to the solution vector to a given set of linear equations, provided they're sparse

[4] For experts, it is BQP-Complete.

firm.

This leaves us to return to the class of problems that Richard Feynman first hinted at, the issue of simulating complex quantum mechanical systems. Such systems are present at the cutting edge of research in many fields. In chemistry, improving our knowledge of the bindings and reactions between molecules at the quantum level helps unlock new capabilities in encouraging or preventing chemical reactions. Closer to biology, enzymes and neural receptors are both quite complex molecular systems, where improved understanding could lead to more effective pharmaceuticals. Closer to physics, improved understanding of the quantum interactions between fields, ions and materials is at the center of the development of improved battery technology, and improvements in energy production from solar to nuclear fusion. Quantum computing will not be a panacea for these research fields any more than access to classical supercomputers was. But just like access to classical supercomputers, they would provide a powerful new tool for simulating and understanding these important problems.

1.1.2 Why haven't we done it already?

This brings us around to the elephant in the room; if theorists first suggested the power of quantum computers in the 1970s, rigorously proved their usefulness in the 1990s, and broadly finished the job of discovering what they could do in principal in 2008, where are the experimentalists?

The obvious answer, especially given the thesis you're reading, is "we're working on it". This section aims to broadly indicate why the problem of building a working quantum computer is exceptionally hard.

In classical computing, we are extremely good at dealing with 'noise'. From the earliest computers, we stored information and processed it with macroscopic systems, where the chance of unexpected natural processes altering the information without our knowing were extremely remote. As we miniaturized and improved computers, we began pushing up against sources of noise. For isntance, DRAM memory must be periodically refreshed to fight against the noisy effects of tiny leakage currents in their constituent transistors. However, this is less representative of the tyranny of noise than our mastery over it; we're capable of building systems with essentially no noise, and choose to build systems with noticeable noise because we understand it and can handle it to squeeze out better performance. In the quantum realm, we do not yet have either option.

Removing noise in a quantum mechanical system entails isolating it from interactions with its environment which affects the system in unknown ways. Such isolation is technically challenging. All quantum computing platforms employ various forms of isolation, but it is broadly believed that isolation alone cannot remove enough noise to enable practical computation. This is particularly true because we must isolate the system from the environment at large, but not from our ability to actually program and control the quantum computer. This is typically understood as a dilemma: the more easily a system is controllable, typically the more noise it is exposed to; the less noise it sees, the more difficult it is to control. Finding creative ways around this trade-off lies at the heart of experimental quantum computing. One approach involves creating physical systems that allow only very specific interactions to affect them, ones we can design and use for control but that nature doesn't conspire to commonly produce, essentially 'building better quantum systems'. Other approaches attempt to measure and understand the influence of the environment and reverse it, protecting the system from noise that still physically occurs, an approach generally called 'error correction'. Both approaches are employed at the same time by most attempts to build a quantum computer, and both approaches have their own internal rich history of difficulties and progress. While it's fair to say we have not yet found the magical combination of these that permits mastery over noise and opens the way to effective quantum computing, the present era represents a time of impressive progress on both fronts.

1.2 Quantum Computing Basics

With motivation and inspiration now covered, we turn to more boring subjects; 'What even is a quantum computer?'. Following more the traditions of computer science than those of physics, we begin by introducing the abstract concept of qubits and build up to more complex systems.

1.2.1 Qubits in Theory

The theory of classical computation is built on the simple abstract concept of the *bit*, a classical object that can be in one of two states, usually labeled **0** and **1**, with the boldface reminding you that these are state labels not numbers. The theory of quantum computation is built on a similarly simple concept, the *qubit*. A qubit is a two-level quantum system, the simplest quantum system that isn't vacuous[5]. Here, a 'level' is a state the system can occupy that will remain stable over time. We traditionally label these two states using the symbols $|0\rangle$ and $|1\rangle$, with the enclosing "kets" there to remind us that these are *quantum mechanical* state labels.

Abstractly, the difference between bits and qubits is the following: Bits may be fully described by which state they are in (either 0 or 1) or under situations of uncertainty

---

[5] Get it?

the probability of finding the bit in each of its two states:

$$\begin{aligned} \text{Bit State} &:= a \times \mathbf{0} + b \times \mathbf{1} \\ \text{where } a \text{ and } b \text{ are real numbers} &\in \mathbb{R} \\ \text{and } a + b &= 1 \end{aligned}$$

Qubits on the other hand enjoy additional freedom; their state can be described similarly, but with the freedom of taking *complex* combinations:

$$\begin{aligned} \text{Qubit State} &:= \alpha \times |0\rangle + \beta \times |1\rangle \\ \text{where } \alpha \text{ and } \beta \text{ are complex numbers} &\in \mathbb{C} \\ \text{and } \left|\alpha\right|^2 + \left|\beta\right|^2 &= 1 \end{aligned}$$

The upshot is that whereas the state of a bit was a human-graspable “it has a chance of being in one state or the other”, a qubit state is a less-intuitive mixing of the states; there are many[6] distinct qubit states that correspond to “half $|0\rangle$ and half $|1\rangle$”, and we have to specify additional information to tell them apart (in this case typically called the *phase angle*).

The new normalization constraint $\left|\alpha\right|^2 + \left|\beta\right|^2 = 1$ encodes the *Born Rule*, a key postulate of quantum mechanics that states that the coefficients in a quantum state (α, β) are not probabilities but *amplitudes*, and their square gives the probability of finding the system in that state when it is measured. We choose now to skip over the rich history of interpretations of quantum mechanics.

---

[6] Typically in a physics context this would read ‘infinite’, but I’m purposefully avoiding a discussion on whether infinite precision exists in reality. If that interests you, consider reading up on the computable numbers and going from there.

The qubit definition given above makes light of the strong connection between quantum mechanics and linear algebra. The qubit states $|0\rangle$ and $|1\rangle$ can be represented as unit vectors:

$$|0\rangle = \begin{pmatrix} 1 \\ 0 \end{pmatrix}, |1\rangle = \begin{pmatrix} 0 \\ 1 \end{pmatrix}$$

This makes more clear that the choice of describing the qubit state as a combination of $|0\rangle$ and $|1\rangle$ was a choice of basis, albeit one foisted upon us by physics. This basis, usually called the 'computational basis', is the one that corresponds directly to the physical energy system when it is in the two states, as we will address in more detail in Chapter 2.

Operations on qubits

Operations taking one qubit state to another generally need the property that they preserve the normalization condition discussed above, requiring them to be *unitary operators*. When represented as matrices, unitary operators (or indeed any complex 2×2 matrix) can be decomposed into the convenient Pauli basis:

$$\begin{aligned} U : |\psi\rangle \to |\psi'\rangle &= \lambda_0\sigma_0 && +\lambda_1\sigma_1 && +\lambda_2\sigma_2 && +\lambda_3\sigma_3 \\ &= \lambda_0 \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} && +\lambda_1 \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} && +\lambda_2 \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix} && +\lambda_3 \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix} \\ &= \lambda_0 I && +\lambda_1 X && +\lambda_2 Y && +\lambda_3 Z \end{aligned}$$

These basic operations X, Y, Z are generally called the 'Pauli gates', where 'gate' is inherited from the traditions of classical computing circuits. In particular, if we can apply the Pauli gates with variable exponents, for example allowing $X^{1/2}$, then we have full control over the state of our qubit.

Combining qubits

We defined the simple qubit in order to build more complex systems, so the nest step is learning to combine them. Two single qubit systems can be combined into a single system by taking a tensor product:

$$\begin{aligned} |A\rangle \otimes |B\rangle &= (\alpha_A |0\rangle_A + \beta_A |1\rangle_A) \otimes (\alpha_B |0\rangle_B + \beta_B |1\rangle_B) \\ &= |AB\rangle = \alpha |00\rangle + \beta |01\rangle + \gamma |01\rangle + \delta |11\rangle \end{aligned}$$

Similarly, operations on the qubits can be combined by taking tensor products; Applying the Pauli X gate on A and doing nothing (I) to B becomes applying $X_A \otimes I_B = XI$ on the two qubit system. This lets us easily define multi-qubit Pauli gates as simple tensor products of the single-qubit Pauli gates.

Notice that this description means that the field of quantum computing implicitly comes down in favour of choosing locality over reality when faced with the consequences of Bell's Theorem. The tensor product has a built in sense of locality; an operation on a single qubit doesn't effect any other qubit, and single qubit operations on two different qubits always commute (the order in which you apply them doesn't matter).

However, not all operations on a pair of qubits can be expressed by tensor products of single qubit operations. Most importantly are the operations capable of *entangling* the two qubits, taking them to an 'entangled state' that cannot be described as a simple product of two single qubit states that obey the normalization constraint. Attempting to describe the state of only one qubit in an entangled state forces you to discard information and consider probabilistic combinations of the ‘pure’ qubit states we have considered so far. These combinations are referred to as ‘mixed states’. The capability of performing entangling operations between qubits augments the sense of locality described above; two qubits must be ‘local’ to each other to perform an entangling operation, and the resulting states may not be completely separable into distinct local parts again afterwards without loss of information.

A more restricted group of operations that will prove important for error correction is the ‘Clifford gates’, which are the gates that normalize the Pauli gates. Specifically, given a pair of gates PC where P is a Pauli gate, C is a Clifford gate if $PC = CP'$, where P' is also a Pauli gate. Colloquially, Clifford gates allow Pauli gates to be ‘moved through them’ and remain Pauli gates, providing us flexibility in the ordering of operations that proves quite useful. However, it is worth noticing that, despite including entangling gates, circuits consisting only of Clifford gates are not universal for quantum computing, and can be simulated efficiently on a classical computer [14]. The power of quantum computing is more subtle than ‘can use entanglement’.

1.2.2 Qubits in Reality (Noise)

In reality, any quantum system we make will not be so isolated from its environment that we can regard it as a being in a pure state. Weak entangling interactions with surrounding material and fields will slightly violate separability and leave our system in a mixed state. However, instead of being forced to keep track of the entanglement with the environment, or track the mixed state in its entirety, we can represent our quantum system being in an intended pure state with a (hopefully small) probability of having an unintended operation applied to it, which we call an *error*.

Cutting-edge error rates per operation in experimental quantum computers sit around

$1 - 0.1\%$. There remains a vast gulf between this and the errors rates required to run the kinds of algorithms discussed in Subsection 1.1.1. It is estimated in [15] that running Shor's Algorithm on a scale relevant to modern cryptography would require around 1500 qubits with error rates per operation around 1×10^{-12}. While possible improvements in the intrinsic error rates of current quantum computers could net us perhaps another order of magnitude, and more exotic improvements to hardware may enable intrinsic error rates as low as 1×10^{-6}, even this would be insufficient in the face of the challenge we are facing.

1.3 Error Correction

Rather than being limited to trying to prevent errors happening, we can also turn to techniques for identifying errors that have occurred and correcting them.

In classical computing, this is usually achieved using *redundancy*, where a bit can be copied many times to produce a larger *logical bit* which is much more robust to noise than any of its constituent bits. While the No-Cloning Theorem [16] prevents us following the same strategy for qubits, it is possible to instead construct multi-qubit entangled states that provides a similar robustness to noise [17].

This is all very well if we desire to communicate a quantum state, where we can assemble an entangled state, subject it once to a noisy communication process and confidently disassemble it at the other end. Performing computation on a quantum state demands more. Specifically, we need not just a prescription of how to assemble many noisy qubits into better logical qubits, but also ways of performing operations between logical qubits that do not compromise their error correction capabilities, which we refer to as *fault-tolerance*.

One particularly helpful way of understanding fault-tolerant code operation is the *stabilizer formalism* [18]. This formalism involves choosing specific error correcting codes which can be described by a set of *stabilizers*, which are multi-qubit Pauli operators which map the state to itself; they *stabilize* the state. In such a code, identifying the presence of errors is easy; construct a operation that measures the stabilizers. In the absence of error, this stabilizer measurement won't affect the state of the system. However, should a qubit have undergone an error process that doesn't commute with one or more of the stabilizers, the value measured for that stabilizer will change. Appropriately overlapping the stabilizers can provides us with an unambiguous way to identify where an error has happened and which error it was. Even if we allow for the fact that the process of measuring the stabilizers will be noisy, we can simply measure them repeatedly, building up greater confidence in the correct measurement values and distinguishing mere measurement noise from true errors on the encoded state.

Further, many effective quantum error can be understood as combination of two classical error correcting codes. The simplest examples are the CSS codes [19, 20], which combine two classical codes by assigning one to identify errors in the Pauli X basis and the other to identify errors in the Pauli Z basis. The Pauli Y is implicitly covered as Y can be considered a combination of X and Z: $Y = -iZX$. There are many prescriptions for how to actually combine classical codes into a quantum code, including hypergraph products [21], balanced products [22] and more. We'll say more about why handling the Pauli X and Z errors is sufficient in Section 1.5.

What's left is to implement these measurements such that they add fewer errors to the system than they notice and correct. This general idea is encoded in the concept of a 'threshold' [23]. In the simplest interpretation, this is the error rate you need beat in order for adding more stabilizers to improve the final logical error rate. Indeed, the promise of quantum error correction is that if you can achieve error rates below threshold, *all* you need to do to achieve truly tiny error rates is add more noisy physical qubits and more stabilizers being measured. This scaling is also extremely favourable; the logical error rate drops exponentially for effective codes operated below threshold, so adding some number more qubits tends to e.g. halve the logical error rate, letting us bridge the chasm to useful logical error rates by simply throwing more of the same noisy qubits at the problem.

The task of experimentally implementing quantum error correction is then to make good on this promise; to make a system that behaves well enough (i.e. below threshold) that using a big enough pile of qubits and an effective QEC code reduces the chance of logical errors to tiny levels and permits running useful algorithms. This overall plan can be summarized as follows:

$$\begin{aligned}\text{Useful Quantum Computing } &= \text{QEC with a threshold}\\ &\quad + \text{ Physical error rates below threshold}\\ &\quad + \text{ Lots and lots of qubits}\end{aligned} \tag{1.1}$$

We now turn to discussing a particularly popular implementation of such a code. However, we will return later to what might go wrong in our assumption that handling only independent Pauli errors is enough.

1.4 The Surface Code

A leading candidate for performing quantum error correction is the *surface code* [24, 25, 26]. This code is traditionally thought of as a CSS stabilizer code consisting of multi-qubit X and Z stabilizers corresponding to ‘plaquettes’ that tile a surface, with special arrangements of plaquettes at the boundaries of the ‘patch’. It can also be approached elegantly as a hypergraph product of simple classical repetition codes [27].

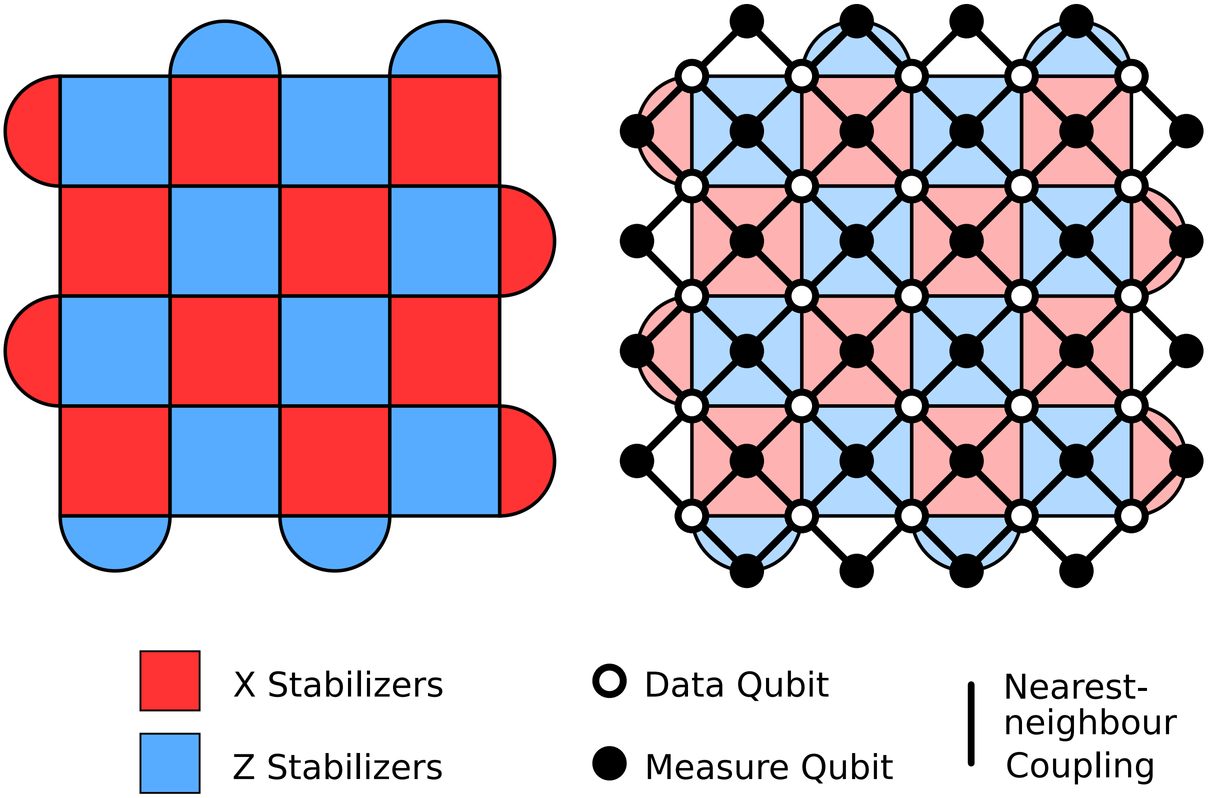


Figure 1.1: **Stabilizers of the surface code.** Left, a schematic showing the stabilizers for a small CSS surface code. Each shape indicated a single stabilizer measuring either X (red) or Z (blue) on each qubit at a vertex of the shape. Right, the physical qubit grid the surface code can be embedded on. The physical qubits (circles) are divided into two groups: data (white) qubits lie at the vertices of the stabilizers, and support the entangled state that encodes quantum information. Measure qubits (black) sit at the center of each stabilizer and are connected to each data qubit involved in that stabilizer, allowing the stabilizer to be measured. The result is a square grid of qubits, with each qubit coupled to their four nearest neighbours.

The surface code can be embedded on a square grid of qubits, with the stabilizers each covering 4 qubits in a local neighbourhood, as shown schematically in Figure 1.1. Additional qubits called *measure qubits* are used at the center of each stabilizer to enable the measurement of the stabilizer operators. Repeatedly measuring these stabilizers allows us to locally detect errors that have occurred.

A surface code patch supports an entangled state on the qubits subject to the stabilizers, so-called *data qubits*, which has the nice property that it retains a global distributed degree of freedom; it can encode one qubit worth of information that is not accessible to any operations smaller than the size of the patch. This is the *logical qubit* embedded in the patch. It takes at least a number of local operations equal to the side length of the patch in order to affect the state of the logical qubit in a way that isn't noticed by the stabilizers. We can purposefully use such large operators to perform logical single qubit Clifford operations. On the other hand, this large operator is also what it necessary for nature to conspire to perform in order to affect our logical state.

The surface code is especially nice from a fault-tolerance perspective because it has a very nice way of implementing logical operations between patches called *lattice surgery* [28, 29, 30]. Unfortunately, lattice surgery provides access to only the Clifford operations, which is not quite enough for universal quantum computing. Fortunately, the remaining necessary operations can be performed fault-tolerantly if we have access to so-called *magic states*. These are special states that are consumed to perform small angle rotations, a necessary part of useful quantum algorithms. It is easily possible to generate noisy magic states using physical operations, but maintaining fault-tolerance requires us to have very clean magic states. Happily, we have the strategy of *magic state distillation* [31, 32, 33, 34] at our disposal, which allows us to convert many noisy copies of a magic state into one or more much cleaner magic states. It is this unusual operation that is expected to dominate the run-time of useful quantum algorithms in these architectures [15].

The surface code is favoured by experimentalists for two major reasons; its logical performance and its connectivity requirements. The performance of the surface code can be evaluated in many ways, but the simplest and most historically important is its threshold, as discussed above. While the numerical value of the threshold is sensitive to details in the error model, the surface code displays a relatively acceptable threshold around 1×10^{-2} for reasonable models. This is notable higher than cutting-edge error rates achieved in superconducting qubit devices, which are typically on the order of

1×10^{-3}, implying that operation of an entire device below threshold is achievable.

Strictly, the connectivity requirement for the surface code is ‘the ability to measure the stabilizers’. In theory, it is possible to do this with any connected graph of qubits and couplings, at worst by moving information around using swap operations. In practice, we want to measure the stabilizers in parallel using the minimum number of operations to stay under the threshold. A natural approach is to use 4 entangling interactions in order to measure weight-4 stabilizers, with each interaction essentially checking one part of the stabilizer before the result can be read out. A square grid of qubits with 4 nearest-neighbour couplings necessary to perform this strategy is now a relatively standard architecture for superconducting qubits [35, 2, 4, 36]. It is worth noting that higher connectivity is more difficult to design. Some architectures purposefully use lower connectivity than the 4-connected square grid just described [37], but this generally requires the use of a different code or requires additional operations, both of which complicate achieving error rates below threshold. On the other hand, better performing codes generally require more or longer range connectivity (or even all-to-all connectivity), which has thus far proved challenging to implement and has not been integrated into a large superconducting device architecture. The surface code’s relatively acceptable hardware demands for its level of performance is what makes it a favoured code for current error correction experiments.

On the other hand, the primary issue with the surface code is its coding rate; essentially the number of physical qubits needed to make a particularly good logical qubit. In the surface code, making the logical qubit linearly better requires making the *area* of the patch bigger, which takes d^2 qubits, where d is the side-length of the patch. Equivalently, the coding rate is $\sqrt{N}$ where N is the total number of qubits, which being worse than linear prevents the surface code from achieving the status of a ‘good’ quantum code. More practically speaking, this leads to estimated numbers of physical qubit per useful logical qubit to end up in the thousands, which creates the demand for millions of physical qubits to run effective and useful computations on thousands of logical qubits. There has been much recent and exciting progress is good quantum codes with much better coding rates, but current proposals generally place difficult demands on the hardware needed to implement them.

The surface code presents an excellent compromise between the realities of hardware and the demands on QEC code performance. Physical error rates compatible with suppressing errors are difficult to reach but achievable, numbers of qubits needed are difficult to produce but achievable, and the final architecture for the entire system is challenging to manufacture, but (hopefully) achievable. Producing better performance than the surface code under similar hardware constraints, or the same performance under fewer constraints, would be a surprising but very welcome development in the field.

In the meantime, we have further detailed our plan (Equation 1.1):

$$\begin{aligned}\text{Useful Quantum Computing } = &\text{ Surface Code QEC}\\ &+ \text{ Lattice Surgery} + \text{Magic State distillation}\\ &+ \text{ Physical error rates below threshold}\\ &+ \text{ Big square grid of qubits}\end{aligned} \tag{1.2}$$

1.5 Error Models

We now turn to investigate further what we mean by ‘physical error rates below threshold’, first addressing ‘errors’ and then ‘physical errors’ and ‘below threshold’.

1.5.1 Pauli Errors

Essentially any evolution of the qubit states that we don't intend can be described as 'error', but its typical to reserve 'an error' to indicate 'a Pauli error': the effect of applying an unintended full-size Pauli gate X, Y, or Z. A probabilistic application of a Pauli error gates provide a convenient representation for a wide class of other error processes; for example the effect of any small unitary (information-conserving) evolution of a qubit state can be expressed to leading order as a small probability of applying a Pauli gate. If a qubit undergoes some small continuous noisy evolution, when we next perform stabilizing measurements we either force it to project back to the stabilized state, or jump to the state with an entire Pauli error applied to it, which the measurement of the stabilizer notices.

Not all errors are exactly representable as Pauli errors; in general the influence of the environment on the quantum system can induce non-unitary dynamics. However, in the context of error correction this distinction can be elided due to the nature of measuring stabilizers: similar to the case with small unitary errors, the stabilizer measurements will project the non-unitary error into either having no effect or producing an entire Pauli error (or possibly Pauli errors on multiple qubits). *Pauli twirling* [38, 39] provides a formal procedure for taking an arbitrary error on a qubit and finding the 'closest' equivalent error supported only on the Pauli gates, which is generally what the non-unitary dynamics may be projected to by the stabilizers and is what the error correction code focuses on actually correcting.

In some sense, the theory of classical error correction rests on the adage "You only have to worry about individual bit-flip errors." If a physical bit undergoes some continuous noisy process, we can easily inspect it to discover whether that process has or has not changed the state of the bit. The equivalent for quantum error correction is then

“You only have to worry about the Pauli errors.” If we can handle the case where we notice any and all Pauli errors on our qubits, then we can handle the full spectrum of possible error processes in these codes.

However, we do not produce codes that can correct all Pauli errors on our qubits: ‘Randomly apply a Pauli error to every qubit at the same time’ is an error we have essentially no chance of recovering from, but represents an extremely unlikely conspiracy on the part of the random noise we imagine our qubits are subject to. When we derive the existence of a threshold for our error correcting codes, we rely on the assumption that errors on different qubits or at different times are *independent*. The likelihood of seeing multiple errors at once is the product of the chance of each error happening alone, and the likelihood of many correlated failures drops off exponentially as the number of failures gets larger, reflecting the lack of a noisy conspiracy in reality.

These two insights (‘Just Handle Pauli Errors’ and ‘Non-Conspiratorial Noise’) collectively form the assumption of *Independent Pauli Errors*, which is a widely regarded bedrock of QEC theory[7].

1.5.2 Pauli Error Models

However, we still have to consider the physical operations that we’re performing and the errors that they are likely to actually produce, as these will determine the real performance of our code.

The simplest starting point is considering ‘measure the stabilizers’ as an the operation itself, one that also has a chance of inducing some undesired Pauli errors on the qubits it is applied to. This is referred to as the ‘code-capacity’ error model because of a correspondence to classical code-capacity. In neglecting the possibility that the value

---

[7] It should not surprise you to learn that this assumption is not entirely accurate, and that is going to have consequences later.

reported by the measurement can be subject to noise and misrepresent the state, this model misses the difficulties of achieving fault-tolerance and usually gives unrealistically high estimates of the code threshold in practice. However, in its simplicity it also displays an extremely nice correspondence to the well-understood spin models of condensed matter physics [40].

The most obvious improvement to the error model is to simply add a chance that the measurement outcome is flipped, or equivalently add a chance of a non-commuting Pauli immediately before the measurement, which is generally called the 'phenomenological noise model' [41]. While this model does fully represent the possible failures, it doesn't relate those failures to anything that is actually happening in reality.

To get our error model close to the errors that physically happen in our device, first specify the operations that are physically happening; we need to decompose the measurement of the stabilizers into a 'circuit', the list of operations that we will perform to assemble and measure the stabilizer information. As previously mentioned, for the surface code this is traditionally (but not necessarily) done by having an extra measure qubit for each stabilizer which has check the stabilizer qubits one by one using an entangling interaction, and can then be read out to learn whether the stabilizer has been changes by an error or not.

Once we have a circuit, we can associate with each operation a rate that it induces a Pauli error as it is applied (and additionally a chance that a measurement outcome is reported incorrectly), which we call a 'circuit noise model'. These models can have rates chosen to accurately reflect the true rate of errors when a device performs each operation, and this gives the most realistic and useful representation of the actual performance of the code.

A circuit noise model can also be further augmented with additional chances of adding in errors to reflect other unintended physical processes unrelated to the operations, such as 'idle errors' associated with simply leaving a qubit alone for a period of time, 'crosstalk errors' associated with a small stray interaction between qubits in the system, and others. We'll discuss the importance of these additions in Section 1.7.

It is the error model as a whole that must be 'below threshold' for a code to suppress errors. In a circuit noise model, different components of the error model might have differently sized influences on the level of error suppression; typically the measurements being reported incorrectly are the least important and the entangling interactions are the most [2].

With this in mind, we can again extend the detail in our plan (Equation 1.2):

$$\begin{aligned}\text{Useful Quantum Computing} &= \text{Surface Code QEC}\\ &\quad+\ \text{Lattice Surgery} + \text{Magic State distillation}\\ &\quad+\ \text{Choice of circuit to measure the stabilizers}\\ &\quad+\ \text{Circuit element error rates below threshold}\\ &\quad+\ \text{Big square grid of qubits}\end{aligned} \tag{1.3}$$

1.6 Decoding

Now we turn to a detail we have thus far brushed over several times: why is it enough to merely notice where the errors have happened, and how exactly do we do that?

When we try to run a computation using the surface code, we will be regularly presented with a set of measurement outcomes corresponding to the stabilizers that were just measured. The first thing we will notice is which stabilizers have *changed* measured values since last time we measured them. This is an indication of an error; possibly applied to a data qubit since we last measured the stabilizer, but possible just this particular measurement being unreliable. This difference in measured values is called a *detection event*.

In the bulk of a surface code patch, all errors produce two detection events. An X error on a data qubit will be noticed by both Z stabilizers that check that qubit. A measured value being misreported will be noticed because it disagrees with a previously correct measurement before it, and a hopefully correct measurement afterwards. Two errors can result in flipping a measurement away from and then back to it's previous value and avoiding producing a detection event, but both errors will still produce another detection event each. At the boundary of the patch it is possible to apply an error that produces only one detection event. This provides another way of seeing how many errors are necessary to change the logical state; we start with an error at one boundary that induces one detection event, add errors that cancel that event and produce another slightly closer to the other boundary, continuing until we reach the opposite side of the patch and can induce another single detection event to cancel the one we have, producing a chain of errors across the patch which is not noticed by the stabilizers. In fact, this change is an operation on our logical state, and if Nature conspires to apply this pattern of errors by chance our state will indeed be modified without our knowledge, and our computation ruined. This also provides the intuition for why the error suppression of the code improves with patch size; it demands a larger and larger conspiracy on the part of nature, which becomes less and less likely if we believe that the errors happen randomly without correlation.

We can summarize this perspective with a construct called the 'error graph'. Each possible detection event in our computation provides a node in the graph, and each possible error provides an edge connecting the detection events that will detect that error. There are various options for how to represent the edges at the boundary, but simplistically we can attach an 'edge to nowhere' to each of the boundary detection event nodes. This picture simplifies out all the details of the experiment itself and keeps only what is necessary to figure out whether the logical state has likely been changed by errors or not. If we mark all the node that feature an actual detection event in this graph, then the task of error correction is simplified to simply pairing these detection events up correctly, either to another event or to the boundary. If we correctly pair up the events that were produced together, then we will effectively have learned any effect that these errors might have had on the logical state we will measure out at the end of the computation. If we pair them up incorrectly, then we might fail to notice that the errors have affected the logical observable, and we'll get the wrong answer at the end of our algorithm. This problem is helpfully well-studied and there are several known algorithms that perform well at this task, including 'Minimum-weight Perfect Matching', 'Union Find', 'Belief matching' and more [4]. Below threshold, where the code works effectively, this problem corresponds to mostly seeing isolated pairs of detection events or single events right next to the boundary, making pairing up easy. Near or above threshold, it becomes difficult to distinguish one choice of pairing from another which has the opposite effect on the logical information, and thus leads to high rates of logical errors.

Rather than being able to solve this problem after the computation is finished, the way magic-state distillation works requires us to solve it as we go, which is called 'real-time decoding'. At the end of a distillation process, we have a chance of having distilled the correct state or of having distilled the state with an additional Clifford applied to it. A logical measurement at the end of the distillation process tells us which of these two cases occurred, and we can easily fix the state if we're in the second case. However, knowing the outcome of that logical measurement requires that we have an up to date idea as to whether it has been flipped by the errors that happened during the process or not. We can keep the not-yet-corrected magic state around for a while before using it, in order to gain a bit more time for our decoding to catch up. Bu this makes clear that we need the decoding to be able to keep pace with the measurements coming out of the quantum computer, or else it won't be able to catch up and our computation will fall apart. For expected superconducting quantum computers, measurements will occur around a million times every second, at which point the decoding process will need to update it's best idea of what events are matches with what before the next measurements come in. When the computer reaches the sizes expected to be necessary for useful computation, each layer of measurements will be providing millions of new measurement outcomes. As such, real-time decoding represents a relatively challenging classical computational task; perhaps the majority of the infrastructure and power dissipation for running a quantum computer in this way will come from the classical compute necessary to keep pace with the error correction.

This too requires an update to our latest plan (Equation 1.3):

$$\begin{aligned}\text{Useful Quantum Computing } = {} & \text{Surface Code QEC}\\ & + \text{ Lattice Surgery} + \text{Magic State distillation}\\ & + \text{ Real-time Decoding}\\ & + \text{ Choice of circuit to measure the stabilizers}\\ & + \text{ Circuit element error rates below threshold}\\ & + \text{ Big square grid of qubits}\end{aligned} \tag{1.4}$$

1.7 Correlated Errors

Armed with the idea of the error graph, we can now return to the validity of the independent Pauli assumption and its implications for the error models we've discussed.

Let's consider a new physical error source; a meteor hits the city containing the quantum computer. Even if we (somewhat inaccurately) Pauli-twirl the result of this error mechanism as producing a random Pauli error on every qubit and thus losing all our qubit's information (rather than their abject destruction), we'll discover that this error is not correctable by the surface code (or indeed any code, for obvious reasons.) The property that underlies this non-correctability is the correlated nature of the error; all the qubits fail together. No part of the code remains error free to check in and correctly report on errors elsewhere - with every qubit affected, there is nowhere for the logical information to hide. Integrating this term into our decoder error model requires more than an edge in the error graph: it is represented as a *hyperedge*, a connection between many nodes simultaneously rather than just between two. A hyperedge for this would simply connect all the nodes on every qubit after the meteor hits.

These examples provides us with the most useful definition of correlated errors in the context of quantum error correction:

Definition (Correlated Error)

An error mechanism capable of causing more than 2 detection events.

In fact, the surface code plan does typically feature some inbuilt correlated errors. The simplest example is the Pauli Y error. This error decomposes as an X and a Z error *together*, producing 4 total detection events. We could choose to represent this as a weight-4 hyper-edge in our error graph, but we generally choose not to. One excellent reason for this is that adding hyperedges to the error graph takes the matching-up-events problem from pretty tractable to completely impractical. Instead, we rely on noticing both decomposed X and Z errors rather than one correlated Y error[8]. In general, it is always possible to decompose the action of a correlated error into a (possibly large) set

---

[8] There are cheap pre- and post-processing tricks we can do to still somewhat exploit our knowledge that X and Z errors are more likely to happen together than we'd otherwise expect [42], but fully exploiting correlations remains computationally expensive.

of uncorrelated errors, which tells us the weight of the correlations:

Definition (Correlated Error Weight)

The number of uncorrelated errors necessary to cause the same detection events as a given correlated error.

In fact, the difficulty of handling a correlated error is essentially determined by the weight, with essentially no major influence from other factors such as the spatial or time-like distances involved [43].

There are important error sources in the vast gulf of weight between the weight-2 Y errors and meteor impacts, and these prove to be extremely important when running QEC experiments. Higher weight correlated errors can also be dealt with by attempting to catch all the decomposed errors, but this usually blows the error budget; either remaining below threshold for realistic devices becomes impossible, or the error suppression when below threshold gets significantly worse. If below threshold performance can still be acheived, This requires significantly more infrastructure to run the device and to achieve real-time decoding, and generally blows the financial budget of the research effort trying to build the quantum computer.

Therefore, for our plan to use error correction to achieve quantum computing to be considered promising, we add one more detail of our plan, the one that is most important for this thesis:

$$\begin{aligned}\text{Useful Quantum Computing } &= \text{Surface Code QEC}\\ &+ \text{ Lattice Surgery} + \text{Magic State distillation}\\ &+ \text{ Real-time Decoding}\\ &+ \text{ Choice of circuit to measure the stabilizers}\\ &+ \text{ Circuit element error rates below threshold}\\ &+ \text{ Big square grid of qubits}\\ &+ \textit{ No significantly correlated errors}\end{aligned} \tag{1.5}$$

References

[2] Google Quantum AI et al. "Exponential suppression of bit or phase errors with cyclic error correction". In: *Nature* 595.7867 (July 15, 2021). _eprint: 2102.06132, pp. 383–387. DOI: `https://doi.org/10.1038/s41586-021-03588-y`.

[4] Google Quantum AI et al. "Suppressing quantum errors by scaling a surface code logical qubit". In: (2022). DOI: `10.48550/ARXIV.2207.06431`.

[7] Scott Aaronson. "NP-complete problems and physical reality". In: *ACM SIGACT News* 36.1 (Mar. 2005), pp. 30–52. URL: `https://www.scottaaronson.com/papers/npcomplete.pdf`.

[8] András Gilyén, Seth Lloyd, and Ewin Tang. "Quantum-inspired low-rank stochastic regression with logarithmic dependence on the dimension". In: (2018). Publisher: arXiv Version Number: 1. DOI: `10.48550/ARXIV.1811.04909`.

[9] Ewin Tang. "A quantum-inspired classical algorithm for recommendation systems". In: *Proceedings of the 51st Annual ACM SIGACT Symposium on Theory*

of Computing. STOC '19: 51st Annual ACM SIGACT Symposium on the Theory of Computing. Phoenix AZ USA: ACM, June 23, 2019, pp. 217–228. DOI: `10.1145/3313276.3316310`.

[10] Ewin Tang. "Quantum Principal Component Analysis Only Achieves an Exponential Speedup Because of Its State Preparation Assumptions". In: *Physical Review Letters* 127.6 (Aug. 4, 2021), p. 060503. DOI: `10.1103/PhysRevLett.127.060503`.

[11] P.W. Shor. "Algorithms for quantum computation: discrete logarithms and factoring". In: *Proceedings 35th Annual Symposium on Foundations of Computer Science*. 35th Annual Symposium on Foundations of Computer Science. Santa Fe, NM, USA: IEEE Comput. Soc. Press, 1994, pp. 124–134. DOI: `10.1109/SFCS.1994.365700`.

[12] Lov K. Grover. "A fast quantum mechanical algorithm for database search". In: *Proceedings of the twenty-eighth annual ACM symposium on Theory of computing - STOC '96*. the twenty-eighth annual ACM symposium. Philadelphia, Pennsylvania, United States: ACM Press, 1996, pp. 212–219. DOI: `10.1145/237814.237866`.

[13] Aram W. Harrow, Avinatan Hassidim, and Seth Lloyd. "Quantum Algorithm for Linear Systems of Equations". In: *Physical Review Letters* 103.15 (Oct. 7, 2009), p. 150502. DOI: `10.1103/PhysRevLett.103.150502`.

[14] Daniel Gottesman. "The Heisenberg Representation of Quantum Computers". In: (1998). Publisher: arXiv Version Number: 1. DOI: `10.48550/ARXIV.QUANT-PH/9807006`.

[15] Craig Gidney and Martin Ekerå. "How to factor 2048 bit RSA integers in 8 hours using 20 million noisy qubits". In: *Quantum* 5 (Apr. 15, 2021). Publisher: Verein zur Förderung des Open Access Publizierens in den Quantenwissenschaften, p. 433. DOI: `10.22331/q-2021-04-15-433`.

[16] W. K. Wootters and W. H. Zurek. “A single quantum cannot be cloned”. In: *Nature* 299.5886 (Oct. 1982), pp. 802–803. DOI: `10.1038/299802a0`.

[17] Peter W. Shor. “Scheme for reducing decoherence in quantum computer memory”. In: *Physical Review A* 52.4 (Oct. 1, 1995), R2493–R2496. DOI: `10.1103/PhysRevA.52.R2493`.

[18] Daniel Gottesman. “Stabilizer Codes and Quantum Error Correction”. Publisher: arXiv Version Number: 1. PhD thesis. 1997. URL: `https://arxiv.org/abs/quant-ph/9705052`.

[19] A. R. Calderbank and Peter W. Shor. “Good quantum error-correcting codes exist”. In: *Physical Review A* 54.2 (Aug. 1, 1996), pp. 1098–1105. DOI: `10.1103/PhysRevA.54.1098`.

[20] Andrew Steane. “Multiple-particle interference and quantum error correction”. In: *Proceedings of the Royal Society of London. Series A: Mathematical, Physical and Engineering Sciences* 452.1954 (Nov. 8, 1996), pp. 2551–2577. DOI: `10.1098/rspa.1996.0136`.

[21] Jean-Pierre Tillich and Gilles Zemor. “Quantum LDPC Codes With Positive Rate and Minimum Distance Proportional to the Square Root of the Blocklength”. In: *IEEE Transactions on Information Theory* 60.2 (Feb. 2014), pp. 1193–1202. DOI: `10.1109/TIT.2013.2292061`.

[22] Nikolas P. Breuckmann and Jens N. Eberhardt. “Balanced Product Quantum Codes”. In: (2020). Publisher: arXiv Version Number: 3. DOI: `10.48550/ARXIV.2012.09271`.

[23] Dorit Aharonov and Michael Ben-Or. "Fault Tolerant Quantum Computation with Constant Error". In: (1996). Publisher: arXiv Version Number: 2. DOI: `10.48550/ARXIV.QUANT-PH/9611025`.

[24] A. Yu Kitaev. "Fault-tolerant quantum computation by anyons". In: *Annals of Physics* 303.1 (1997), pp. 2–30. DOI: `10.1016/S0003-4916(02)00018-0`.

[25] S. B. Bravyi and A. Yu. Kitaev. "Quantum codes on a lattice with boundary". In: (1998). Publisher: arXiv Version Number: 1. DOI: `10.48550/ARXIV.QUANT-PH/9811052`.

[26] Austin G. Fowler et al. "Surface codes: Towards practical large-scale quantum computation". In: *Physical Review A* 86.3 (Sept. 18, 2012). Publisher: American Physical Society, p. 032324. DOI: `10.1103/physreva.86.032324`.

[27] Anirudh Krishna and David Poulin. "Fault-Tolerant Gates on Hypergraph Product Codes". In: *Physical Review X* 11.1 (Feb. 4, 2021), p. 011023. DOI: `10.1103/PhysRevX.11.011023`.

[28] Clare Horsman et al. "Surface code quantum computing by lattice surgery". In: *New Journal of Physics* 14.12 (Dec. 7, 2012), p. 123011. DOI: `10.1088/1367-2630/14/12/123011`.

[29] Benjamin J. Brown et al. "Poking Holes and Cutting Corners to Achieve Clifford Gates with the Surface Code". In: *Physical Review X* 7.2 (May 24, 2017), p. 021029. DOI: `10.1103/PhysRevX.7.021029`.

[30] Daniel Litinski and Felix von Oppen. "Lattice Surgery with a Twist: Simplifying Clifford Gates of Surface Codes". In: *Quantum* 2 (May 4, 2018), p. 62. DOI: `10.22331/q-2018-05-04-62`.

[31] E. Knill. “Fault-Tolerant Postselected Quantum Computation: Schemes”. In: (2004). Publisher: arXiv Version Number: 1. DOI: `10.48550/ARXIV.QUANT-PH/0402171`.

[32] Sergey Bravyi and Alexei Kitaev. “Universal quantum computation with ideal Clifford gates and noisy ancillas”. In: *Physical Review A* 71.2 (Feb. 22, 2005), p. 022316. DOI: `10.1103/PhysRevA.71.022316`.

[33] Sergey Bravyi and Jeongwan Haah. “Magic-state distillation with low overhead”. In: *Physical Review A* 86.5 (Nov. 27, 2012), p. 052329. DOI: `10.1103/PhysRevA.86.052329`.

[34] Earl T. Campbell, Barbara M. Terhal, and Christophe Vuillot. “Roads towards fault-tolerant universal quantum computation”. In: *Nature* 549.7671 (Sept. 14, 2017), pp. 172–179. DOI: `10.1038/nature23460`.

[35] Frank Arute et al. “Quantum supremacy using a programmable superconducting processor”. In: *Nature* 574.7779 (Oct. 24, 2019). Publisher: Nature Publishing Group, pp. 505–510. DOI: `10.1038/s41586-019-1666-5`.

[36] Sebastian Krinner et al. “Realizing repeated quantum error correction in a distance-three surface code”. In: *Nature* 605.7911 (May 26, 2022). Publisher: Springer Science and Business Media LLC, pp. 669–674. DOI: `10.1038/s41586-022-04566-8`.

[37] Neereja Sundaresan et al. “Matching and maximum likelihood decoding of a multi-round subsystem quantum error correction experiment”. In: (2022). DOI: `10.48550/ARXIV.2203.07205`.

[38] Christoph Dankert et al. “Exact and approximate unitary 2-designs and their application to fidelity estimation”. In: *Physical Review A* 80.1 (July 6, 2009), p. 012304. DOI: `10.1103/PhysRevA.80.012304`.

[39] Steven T. Flammia and Joel J. Wallman. “Efficient Estimation of Pauli Channels”. In: *ACM Transactions on Quantum Computing* 1.1 (Dec. 9, 2020), pp. 1–32. DOI: `10.1145/3408039`.

[40] Eric Dennis et al. “Topological quantum memory”. In: *Journal of Mathematical Physics* 43.9 (Sept. 2002), pp. 4452–4505. DOI: `10.1063/1.1499754`.

[41] Andrew J. Landahl, Jonas T. Anderson, and Patrick R. Rice. “Fault-tolerant quantum computing with color codes”. In: (2011). Publisher: arXiv Version Number: 1. DOI: `10.48550/ARXIV.1108.5738`.

[42] Austin G. Fowler. “Optimal complexity correction of correlated errors in the surface code”. In: (2013). Publisher: arXiv Version Number: 1. DOI: `10.48550/ARXIV.1310.0863`.

[43] Austin G. Fowler and John M. Martinis. “Quantifying the effects of local many-qubit errors and nonlocal two-qubit errors on the surface code”. In: *Physical Review A* 89.3 (Mar. 12, 2014). Publisher: American Physical Society, p. 032316. DOI: `10.1103/PhysRevA.89.032316`.

Chapter 2

Superconducting Quantum Devices

Having now discussed quantum computing in the abstract for some time, we turn to what a quantum computer looks like in real life[1]

As mentioned in Subsection 1.1.2, designing a good quantum system comes down to isolating it from its environment. Quantum mechanics in general has been long associated with tiny systems like single atoms or single electrons because these systems are much easier to isolate from their environments, and it is easier to understand their behaviour of once they are isolated and quantum mechanics comes into play. Many proposals for quantum computers exploit exactly such systems, aiming to isolate atomic or subatomic particles in reasonable numbers and then engineer them to interact with each other. This approach benefits from using building blocks that are both well studied and guaranteed to be identical, but suffers from the difficulties of designing and fabricating trapping and control structures on the atomic scale.

However, there is another great non-atomic-scale tradition of quantum mechanics, which is the study of quantum effects in mesoscopic and even macroscopic scales. Typically concerning systems of trillions of atoms at the low end, this academic tradition is

---

[1] Thus confirming that this thesis is indeed in *experimental* physics.

often referred to (without irony) as “Many-body physics”. Quantum behaviour in large ensembles of particles produces many striking physical effects on scales observable by humans, such as the behaviour of superconductors, superfluids and the unusual electrical characteristics produced by the quantum Hall effect.

Using a macroscopic quantum phenomenon to pursue quantum computing presents some immediate benefits. At larger scales, the engineering of the system becomes much easier. First, the quantum system itself is easier to design and fabricate, allowing the designer to choose and adjust the sizes, shapes, energy-scales and couplings in the device with freedom, as opposed to being forced to use the atoms and couplings that nature provides at the atomic scale. Second, the surrounding engineering of controls structures and wiring, ‘talking to the outside world’ becomes much easier, as it lands much closer to the regime perfected in classical computing devices, neatly sidestepping the complexity of design and fabrication with individual atoms. This freedom in design is both a blessing and a curse; variation in and poor execution of designs both frustrate progress. But mostly it is a blessing. This chapter will present components of our current designs that are the culmination of decades of trying new structures, solving old problems and developing new approaches to trade-offs.

2.1 Superconductivity

Superconductivity permits electrical currents to flow through a material without loss. More specifically, this means that the electrical current has decoupled from its environment, and no longer loses energy to, or indeed even interacts with, the lattice of atoms through which it is moving. This decoupling from the environment is the result of the current (or more accurately, all the charge carriers that make up the current) being able to behave as a single coherent quantum object, rather than as trillions of independent charge carriers.

Superconductivity is a particularly excellent macroscopic quantum phenomenon for the purposes of quantum computing. First, it's an electrical phenomenon, allowing it to benefit from our knowledge of and integrate well with all our classical electronics and computing infrastructure. Similarly, it's solid state, being amenable to the already-popular modality of making chips and attaching electrical leads to them. It's relatively easy to make the phenomenon occur - taking most pure metals and simply cooling them will suffice. Finally, it's relatively well understood and well studied for a macroscopic quantum phenomenon [44]. A simple but effective model for superconductivity involved the pairing up of charge carrying electrons to form *Cooper pairs*, which are amenable to *condensing* into the aforementioned collective quantum object called a *condensate*. The lossless supercurrent in a superconductor is the current of moving Cooper pairs. This pairing up occurs because of a weak, indirect, attractive interaction between electrons, which is usually drowned out by thermal motion and interactions with defects in the metallic lattice. Once the metallic lattice is cooled below a critical temperature, pairing takes over and the resulting Cooper pairs can flow unimpeded.

As with all interesting physics, this is not the whole story. Excitations with enough energy can still break pairs out of the condensate, producing fragments called *quasiparticles*; these are almost but not quite regular electrons that are persisting in the superconducting state[2]. Quasiparticles, unlike their paired cousins, are very happy to interact with the lattice and do not move without loss. At any finite temperature, there is always a small population of quasiparticles produced by random fluctuations[3]. Being a

---

[2] One particularly illuminating question that can hep tease out the difference between electrons and quasiparticles is to as "What is the electrical charge of a superconducting quasiparticle?". The answer is surprisingly not "one electron charge".

[3] In fact, is it an ongoing mystery in the field that there are generally more quasiparticles than you'd expect just from the temperature, the so called anthermal quasiparticle population. Many explanations for their existence, from the difficulty of providing a sufficiently cold environment for the chip [45], to physics extending the thermalization time of quasiparticles [46], to energetic cosmic radiation [47]

primary cause of loss, some effort is typically put into ensuring background quasiparticles are heavily suppressed, especially by ensuring that the superconductor is cooled to well below the critical temperature.

2.2 Microwave Circuit QED

This leads us to discuss what noise processes are likely to affect a superconducting circuit, and how we might avoid them so as to make a qubit. We already know that we would like to make sure our system is well below the superconducting critical temperature, but also that we're likely to want some way to inject energy into our qubit without breaking superconductivity.

Dilution refrigerators are an essentially off-the-shelf way of reaching temperatures as cold as 10-30 mK. In terms of frequency of thermal excitations this works out to be on the order of 1 GHz - if we'd like to make a system that isn't sensitive to the remaining thermal noise in the system, we'd do well to make sure it isn't sensitive at frequencies around 1 GHz or lower. The superconductivity in our device will also need this property; most sensible choices of metals we will only lose their superconductivity when exposed to signals or noise on the order of 10-100 GHz. If we'd like to be able to excite and control our qubit, then we should also ensure that the qubit only requires control signals lower than this, so as to not accidentally break superconductivity and induce loss.

This regime (10-100 GHz) conveniently aligns with the range of *microwave circuits*, a well studied field in electrical engineering [48]. Microwave circuits have long been useful in communications, forming the backbone of technologies like mobile cell phone service, Wi-Fi, and satellite communication. This not only provides a great deal of knowledge in the design and operation of these kinds of circuits, but also provides a robust ecosystem

---

have been proposed and studied as the source of this population, but none have provided a completely satisfying conclusion thus far.

of cheap and effective classical electronics we can use for interfacing with and controlling our qubits. As such, much of the history of superconducting qubits begins with simple microwave circuits made from superconducting materials and manipulated by microwave signals.

2.3 Junctions and Squids

With these choices under our belt, we are presented with one remaining challenge, which superconductivity will also solve for us. When pushed into the quantum regime, microwave circuits become quantized and display many[4] individual allowed energy levels, rather than a continuous spectrum of possible energies. So far so good. We ideally wish make circuits where we can control which level we are in precisely, and so choose two levels to remain in and operate the circuit as a qubit, ignoring other levels entirely.

Unfortunately, the simplest microwave circuits are linear resonators[5]. A microwave resonator can be easily constructed by appropriately combining just a capacitor and an inductor, but even more simply by most arrangements of a metal strip near a ground-plane [48]. However, resonators display levels which are evenly and linearly spaced apart. What this means is that we cannot move between specifically chosen levels; the energy difference between each level and its neighbours are all the same, and so when we try to change the level by adding that amount of energy, we can cause many transitions and will rapidly occupy many different levels, preventing us staying only in two we'd selected to make up our qubit. To separate out levels we require a source of non-linearity - uneven spacing, so that when we try to transition between two specific levels, that transition can be uniquely addressed.

---

[4] Again, read 'many' as 'infinite' levels if you prefer, but I hope you don't.

[5] It's not unfortunate the resonators are simple, but it is unfortunate that they are resonators, at least for us.

Fortuitously, this is exactly the behaviour provided by when the path of a supercurrent is interrupted by a very thin non-conducting barrier, a *Josephson Junction*. Again, skipping over the interesting history of the Josephson effect, these junctions provide a very simple way to produce an non-linear inductive circuit element. If we use these junctions in our microwave circuits instead of simple indicators, then we can introduce nonlinearity, break up the uneven spacing, and produce a system we can operate like a qubit.

The Josephson junction lies at the heart of superconducting qubit designs. They are simultaneously easy and difficult to fabricate. Making a thin non-conducting barrier is not extremely difficult - most metals will produce a a thin non-conductive oxide in the presence of oxygen, which we can then cover with further metal to make an effective junction. On the other hand, they are difficult to make because the Josephson effect is extremely (exponentially) sensitive in the thickness of the barrier. Making exactly the same junction twice can be quite difficult, because the chemistry involved in oxidising the metal is not reliable on the atomic scale[6].

One way of addressing this fabrication challenge is to use pairs of junctions wired in parallel, a superconducting quantum interference device or *squid*[7]. A squid behaves like a single junction that can have its inductance tuned by an applied magnetic field. When used in a microwave circuit, using a squid generally allows the frequency of the circuit to be lowered by applying such a magnetic flux, leading to such circuits being called 'frequency-tunable' or 'flux-tunable'. This provides benefits above simply working around fabrication variance, as we'll discuss in Section 2.6.

---

[6] In fact, this is a primary reason that the gold-standard superconducting junctions are made of aluminum - due to some interesting physics in its superconductivity, Aluminum is only very sensitive to barrier thickness, rather than extremely sensitive.

[7] I've chosen throughout to render 'squid' as a common word rather than an acronym in an effort to promote it into the venerable crowd of 'laser' and 'radar' and so push the English language forward.

2.4 The Transmon

Rather than address the long history of innovation in superconducting Josephson qubits, we'll now skip to the current leading candidate, the Transmon qubit [49]. This qubit presents an excellent set of compromises in design, avoiding many possible pitfalls and producing a qubit that is relatively easy to design and fabricate, displays excellent coherence characteristics, is amenable to adding local couplings and has rapidly become the gold-standard qubit in the field.

The primary benefit of the transmon design, especially when compared to the designs it was derived from, is its charge insensitivity. Physically, the transmon's quantized states are subtly different oscillating currents passing through the junction, called a 'plasma oscillation' or a 'plasmon'. The locations of the charges involved in these oscillations average out over even short time, and as such the transmon is affected only very weakly by external electric fields, which are a very common source of noise. The two lowest energy oscillations are used as the computational states $|0\rangle$ and $|1\rangle$, and oscillations with higher energies are generally labeled $|2\rangle$, $|3\rangle$, and so on.

Despite the benefits of superconductivity, the energy states of a transmon are not totally decoupled from their environment. Quasiparticles provide one mechanism for this, but others include small amounts of loss at the surface of the superconductors, and in the non-superconducting dielectric substrate we fabricate the superconducting circuit on top of. While careful design can keep these interactions to a minimum, these channels tend to eventually steal energy from the qubit, forcing it to *decay* down the ladder of states toward the lowest energy $|0\rangle$ state. The timescale on which this happens, the *energy coherence time* or *T1 time* is a primary metric for how good a transmon, or indeed any qubit is, and sets the timescale inside which you'd like to be able to do error correction and notice a decay process if it occurs. For transmons, this time is typically in the range of 10-100 μs, which is happily slower than a typical operation on a transmon, which usually take around 10 ns, and slower than relatively standard microwave electronics and the classical computers that control them.

The price to be paid for all these benefits is a weak nonlinearity. The charge insensitivity is achieved by using a large capacitor, which relatively speaking washes out the nonlinear effect of the junction. The energy levels are only just not-evenly-spaced, meaning that when operating the transmon we must be delicate if we want to ensure we are not exciting the aforementioned higher levels. Even with great care, these higher levels are often excited anyways, a general process called *leakage* and giving the higher non-computational states the name *leakage states.* While there are also decay processes that tend to eventually remove energy from leaked qubit and return them to the computational states, this process typically takes a long time. We'll discuss the implications of this in Section 2.6.

2.4.1 Couplings and couplers

As was alluded to in Subsection 1.4, coupling qubits together presents an additional and important challenge. Transmons are particularly amenable to being coupled; because they feature large capacitance, that capacitance can easily be broken up into a self-capacitance and relatively strong capacitance to neighbouring qubits with plenty to space. This process of direct coupling has some significant disadvantages though; if the qubits have similar energies, they will always be strongly coupled and will talk to each-other uncontrollably. If they are set at very different energies to prevent this, then they will be hard to couple when we want to couple them. We require the addition of either a method of producing coupling that bridges the energy gap, such as parametric or cross-resonant driving [50], or to use frequency-tunable qubits so we can bring them together when we want coupling and part when we do not [51]. Both of these strategies have significant disadvantages, requiring either strong microwave drives or needing to move qubit through large frequency excursions.

Another option is to add additional structure in the form of a *coupler*, a controllable quantum system that mediates the interaction between two qubits. With a controllable coupler, we're not dependant on applying control signals to the qubits to produce couplings as well as single qubit control, breaking up the design problem. Couplers have produced the lowest error entangling operations in superconducting qubits so far, and provide a large amount of freedom in what interactions can be achieved [52, 53, 54]. Their cost is in the additional complexity they bring to the overall device, requiring additional structures to be designed and fabricated and to be operated correctly[8].

2.4.2 Dispersive Readout

If there is another primary downside to the transmon qubit (other than its non-linearity), it's that it is too good at being a qubit. Throughout our operation of an error corrected device, we rely on being able to quickly readout the state of the qubit to measure a stabilizer with high accuracy. Designing circuit elements to allow this operation is relatively difficult [55]. The transmon's high performance as a qubit arises from it not providing obvious couplings to its outside environment. Where previous superconducting qubit designs were read out using a charge sensor or a magnetic flux sensor, neither of these strategies works well on a transmon. Instead, we're forced to read the transmon out *dispersively*, in a way that relies only on it's non-linearity, which as mentioned now is not that strong to begin with.

When designing a readout method, the important performance criteria are:

---

[8] Notice however that this cost plays into the strengths of the approach of using highly flexible and design-able quantum systems in general.

1. that the readout process does not disturb the qubit when you're not applying it,
2. that you successfully measure the qubit state as it was at the start of the process,
3. that the qubit is in a known state at the end of the process.

Not disturbing the qubit when you're not measuring it and measuring it fast enough that you can actually learn something before it decays turn out to be at odds with each other [56]; generically, if you have a stronger coupling so you can readout more easily, you have a stronger coupling that affects the qubit when you're not reading it out. In dispersive readout of transmons particularly, it's possible to improve on this trade-off by adding further elements; first another linear resonator coupled to the qubit which we inspect in lieu of inspecting the qubit itself, and then by adding a further resonator (usually called a 'Purcell filter') to filter our inspection signal at frequencies the qubit is sensitive to[9]

The requirement that the qubit be in a known state at the end of the process is somewhat relaxed compared to how it is typically phrased. Often, we also demand that this known state is 'the state the measurement told you it was in', which is a seemingly natural thing to ask for and is called a *quantum non-demolition* or QND measurement. These measurements have the nice property that they can be easily repeated or extended to gain more information about the state of the qubit if the readout is noisy. Dispersive readout is a QND readout.

In some readout processes however, the state produced at the end is $|0\rangle$, regardless of what was measurement was reported, which is called a *demolition measurement*. These measurements have the nice property that you don't have to do anything else before reusing that qubit, which turns out to be more natural for error correction. A demolition

---

[9] Again, notice that our solutions here play to the strengths of our inherent ability to use design flexibility and complexity to get the outcomes we want.

measurement can be constructed from a QND measurement followed by a demolition process. Such processes are conveniently provided by the same readout circuitry; if we're able to readout 'too hard' then we have a convenient way of destroying the qubit state when we want to. More sophisticated methods will be discussed at length in Chapter 3.

2.5 Device Architecture and System Architecture

With these basic elements in hard, what remains to to put them all together into a single device, each of which we can individually control and can measure, and some arrangement of which we can couple together, and ideally all of which are highly coherent, well behaved and no have any additional couplings we didn't intend. The work of putting the pieces together into a coherent device architecture is perhaps the central challenges in producing a working quantum computer. All of the work in this thesis uses and concerns the use of the Google Sycamore architecture [35], which is a robust combination of a square grid of frequency-tunable transmons, couplers between nearest-neighbouring qubits, multiplexed dispersive readout shared across multiple qubits, and other design elements that all together has proven to be highly performant.

Device architecture represents some of the most difficult trade-offs, as it influences many very different parts of the plan at once; a change might make readout better at the cost of stray couplings that are long range on the chip, which might or might not be highly problematic for our hypothetical error correction strategy. In practice, one of the more difficult tasks is arranging the control structures. Each qubit requires a signal line to control it, and each coupler requires a line to control it, and the readout structures require coupling to the outside world as well. Each of these lines generally does not want to couple to any other, requiring separation and shielding. To handle this, the Sycamore design consists of two chips *bump-bonded* together [57], where one carries wiring over the top of the other, where the qubits are situated. Future devices with additional complexity will likely push us towards even more complex arrangements, with multiple chips [58] and multiple wiring layers [59].

The device architecture sits within the larger surrounding architecture of the entire system that supports its operation. Once a device is in hand, it still must be appropriately packaged and shielded from stray electromagnetic fields, and then installed in a dilution refrigerator to cool it to its operating temperature. The refrigerator also needs to provide access to the chip for the hundreds of control lines, reaching all the way up to room temperature without allowing so much heat down that the chip doesn't cool. These wires are attached to the extensive classical control electronics which are installed around the fridge, which must work in concert to operate all the elements of the device at once, accounting for distortions introduced by the wiring inside the fridge and the packaging of the chip itself. Eventually, these electronics will also be responsible for the real-time decoding of error correction measurements, informing later actions in the circuit as it progresses.

One especially important abstract part of the system architecture are the decisions about how exactly to operate the device, generally called *calibration*. The process of calibrating a device from scratch works like a long sequence of experiments, building up form learning basic physical parameters of the device, to approximately good operating points for further measurements, to highly accurate optimization of many device operations in parallel [60]. Even once the device is performing, maintaining that performance in the face of noise and drift is an ongoing challenge [61]. Eventually, we will need to perform these re-calibrations during the continuous operation of error correction for running a quantum algorithm [62].

2.6 Hardware Correlated Errors

With this high-level understanding of the device that will implement our plan for useful quantum computing (Equation 1.5), we should discuss the error sources in our device that don't fit nicely into the paradigm introduces in Section 1.5 and can possibly cause us some problems. In general, these are sources of high-weight correlated errors, which is we tried to handle with regular error correction would induce a large amount of Pauli-equivalent errors to detect and most likely prevent the suppressing of errors in the manner we were promised by the abstract theory of error correction. Another valid description are that these are the errors that it is up to the experimentalists to fix in order to make good on the promise that errors are relatively uncorrelated; these are the conspiracies we have to break up. We'll discuss a few different sources of correlated errors here, but this list is by no means exhaustive. In vague order of the scale of their correlations and how easy they are to mitigate, we'll discuss:

1. Crosstalk
2. Dead qubits
3. TLSs
4. Leakage
5. Impact error bursts

The prototypical example of a hardware correlated error is *crosstalk*, where either in general or during a particular operation, qubits that are not intended to be coupled are interacting with each other. This typically happens over a relatively local neighborhood and the strength of the additional interactions drops off with distance, providing a limit on how many errors it is likely to induce at once. However, careful avoidance of these kinds of errors drives a lot of design decisions at the architecture level.

A less typical source of 'correlated error' are qubits or couplers that are 'dead' - either because they were fabricated incorrectly, have become damaged, or have had their parameters varied in a way that makes them unusable. Attempting to perform the error correction circuit as if these qubits were 'living' is not conducive to good performance - it is highly likely to induce an effective error on that qubit at every time-slice, producing a string of errors that extends through time and a very large number of errors overall. Obviously some care is put into preventing dead elements, but as devices grow perfect yield becomes more and more unrealistic. Instead, this correlated error can be handled on the level of the code itself; sub-system codes [63, 64] permit changes to the surface code to drop out a qubit from the middle of the grid with only a constant cost to the distance and a minor increase in decoding complexity, disconnecting the correlated error source from the code.

An especially pernicious source of correlated errors are the general phenomenon of *two-level systems* or TLSs. Rather than referring to our transmon qubits, which do not have two levels[10], TLS is a general term for referring to any number of microscopic systems that couple to the qubit excitation at a specific frequency[11]. TLSs have a nasty habit of moving in frequency either diffusively or by jumping back and forth over a wide range, and it is generally unsafe to operate a qubit in close proximity of a TLS in frequency. If a TLS wanders near the operating point of a qubit, it will generally induce problematic errors at a high rate until it wanders away or the qubit is moved. Designing qubits with fewer and better behaved TLSs remains a significant and important open

---

[10] It is not lost on our field that TLSs can be strictly more qubit-like than our qubits. Indeed, it is also not impossible to find a TLS with a longer coherence time than your qubit.

[11] Not all things that couple at a single frequency are two-level systems, but the term is often used as a general descriptor of the problem than a strictly correct descriptor, like saying 'Bigfoot ate my sheep' when you really mean 'an unknown cryptid'

challenge. However, it is possible to work around TLSs using frequency-tunable qubits combined with adaptive calibration strategies that move the qubits when a TLS comes close by [61].

The correlated error source that is most important for the following thesis is leakage. As discussed in Section 2.4, the weak nonlinearity of the transmon makes it especially susceptible to having the qubit leak out of the computational states and into higher energy states. When using transmons, all operations from single qubit control to entangling operations to measurement induce small amounts of leakage, even when great care is taken to avoid this. Once leaked, a qubit will remain so for quite some time, on the order of the energy coherence time, and typically 10s of quantum error correction measurement cycles. While in a leaked state, the qubit won't be able to participate in the QEC circuit correctly, disturbing checks that touch it and seeming to be constantly producing Pauli errors. Worse, as our qubits become better and coherence times get longer, leakage states remain populated for longer before they decay, increasing the number of errors they induce. Even this picture underestimates the scale of the problem, as it has been discovered that leakage can spread virally through entangling operations as discussed in Chapter 4. One leaked qubit can cause another qubit to leak as well, producing an expanding bubble of high error rates and inducing a terrific number of decomposed Pauli errors before it finally decays away. Leakage can be considered a uniquely challenging and problematic source of correlated errors in weakly nonlinear qubits, and requires the demonstration of highly effective mitigation if transmons are to prove an effective foundation for quantum error correction in general. In Chapter 3 and Chapter 4, we'll discuss the effects of leakage on quantum error correcting codes in more detail, and present methods of mitigating it.

Finally, while we're relatively well justified in ignoring the meteor-impact-error, meteors are not the only dangerous cosmic phenomenon that can impact our chip. High-energy radiation in ubiquitous at sea level, both terrestrial in origin in the form of gamma rays and cosmic in origin in the form of tertiary muons descending from the heavens. While present at a scale too low to bother humans, or indeed be noticeable by most classical electronics, our extremely well isolated quantum systems never the less prove a sensitive target. Such radiation is almost impossible to shield against on the timescales we expect to want to run a quantum algorithm, and their effect on the chip is catastrophic: the amount of energy deposited by such a particle as it skims past is more than enough to break superconductivity across the entire device. If it is not impeded by specific choices in materials and device design, this will induce an error burst on all qubits that essentially totally scrambles the quantum information everywhere. This is essentially a direct implementation of the 'un-correctable error' discussed about previously. In Chapter 5, we'll discuss the measurement and characterisation of just such error bursts in our processor, which must be mitigated if QEC is to be successful in such a platform.

While these last two sources of error rate the focus of this thesis, the other three mentioned are the subject of continuous and ongoing work, as are other sources like correlated noise in room temperature electronics. These all factor into our work to improve on our current error budget and achieve performance well below threshold, alongside efforts reducing regular independent error rates, increasing coherence, and better optimising trade-offs in device and system architecture. Work on mitigating these correlated errors represents a vital building block in building a quantum computer, but only one of many important building blocks. Indeed, these error sources are difficult to study especially because they require so much to already be built before they become apparent and can be studied effectively. Working on these problems would not be possible without the impressive progress made by the team at Google and by the field of superconducting quantum computing in general in making devices on a scale where these errors become important. This consistent progress in the performance of superconducting quantum computers, and especially how it is ploughing ahead in the face of ever more complex problems, is what gives me great hope for the future of the field and the advent of useful quantum computing in these systems.

References

[35] Frank Arute et al. "Quantum supremacy using a programmable superconducting processor". In: *Nature* 574.7779 (Oct. 24, 2019). Publisher: Nature Publishing Group, pp. 505–510. DOI: `10.1038/s41586-019-1666-5`.

[44] Charles Kittel. *Introduction to Solid State Physics.* 8th ed. Wiley, 2004.

[45] K. Serniak et al. "Hot Nonequilibrium Quasiparticles in Transmon Qubits". In: *Physical Review Letters* 121.15 (Oct. 10, 2018). Publisher: American Physical Society, p. 157701. DOI: `10.1103/PhysRevLett.121.157701`.

[46] John M. Martinis, M. Ansmann, and J. Aumentado. "Energy Decay in Superconducting Josephson-Junction Qubits from Nonequilibrium Quasiparticle Excitations". In: *Physical Review Letters* 103.9 (Aug. 26, 2009). Publisher: American Physical Society, p. 097002. DOI: `10.1103/PhysRevLett.103.097002`.

[47] Antti P. Vepsäläinen et al. "Impact of ionizing radiation on superconducting qubit coherence". In: *Nature* 584.7822 (Aug. 27, 2020). Publisher: Springer Science and Business Media LLC, pp. 551–556. DOI: `10.1038/s41586-020-2619-8`.

[48] David M. Pozar. *Microwave engineering.* 4th ed. OCLC: ocn714728044. Hoboken, NJ: Wiley, 2012. 732 pp.

[49] Jens Koch et al. "Charge-insensitive qubit design derived from the Cooper pair box". In: *Physical Review A* 76.4 (Oct. 12, 2007). Publisher: American Physical Society, p. 042319. DOI: `10.1103/PhysRevA.76.042319`.

[50] Chad Rigetti and Michel Devoret. "Fully microwave-tunable universal gates in superconducting qubits with linear couplings and fixed transition frequencies". In: *Physical Review B* 81.13 (Apr. 5, 2010), p. 134507. DOI: `10.1103/PhysRevB.81.134507`.

[51] J. Kelly et al. "State preservation by repetitive error detection in a superconducting quantum circuit". In: *Nature* 519.7541 (Mar. 5, 2015). Publisher: Nature Publishing Group, pp. 66–69. DOI: `10.1038/nature14270`.

[52] Charles Neill. "A path towards quantum supremacy with superconducting qubits". PhD thesis. University of California, Santa Barbara, 2017.

[53] Fei Yan et al. "Tunable Coupling Scheme for Implementing High-Fidelity Two-Qubit Gates". In: *Physical Review Applied* 10.5 (Nov. 28, 2018). Publisher: American Physical Society, p. 054062. DOI: `10.1103/PhysRevApplied.10.054062`.

[54] B. Foxen et al. "Demonstrating a Continuous Set of Two-qubit Gates for Near-term Quantum Algorithms". In: *Physical Review Letters* 125.12 (Sept. 15, 2020). _eprint: 2001.08343, p. 120504. DOI: `https://doi.org/10.1103/PhysRevLett.125.120504`.

[55] Daniel Sank. "Fast, accurate state measurement in superconducting qubits". PhD thesis. University of California, Santa Barbara, 2014. URL: `https://www.alexandria.ucsb.edu/lib/ark:/48907/f3w0942t`.

[56] Eyob A. Sete, John M. Martinis, and Alexander N. Korotkov. "Quantum theory of a bandpass Purcell filter for qubit readout". In: *Physical Review A* 92.1 (July 21, 2015). Publisher: American Physical Society, p. 012325. DOI: `10.1103/PhysRevA.92.012325`.

[57] B Foxen et al. “Qubit compatible superconducting interconnects”. In: *Quantum Science and Technology* 3.1 (Jan. 1, 2018). _eprint: 1708.04270, p. 014005. DOI: `https://doi.org/10.1088/2058-9565/aa94fc`.

[58] Alysson Gold et al. “Entanglement across separate silicon dies in a modular superconducting qubit device”. In: *npj Quantum Information* 7.1 (Dec. 2021), p. 142. DOI: `10.1038/s41534-021-00484-1`.

[59] Sergey Bravyi et al. “The Future of Quantum Computing with Superconducting Qubits”. In: (2022). Publisher: arXiv Version Number: 1. DOI: `10.48550/ARXIV.2209.06841`.

[60] Julian Kelly et al. “Physical qubit calibration on a directed acyclic graph”. In: (2018). Publisher: arXiv Version Number: 1. DOI: `10.48550/ARXIV.1803.03226`.

[61] Paul V. Klimov et al. “The Snake Optimizer for Learning Quantum Processor Control Parameters”. In: (2020). _eprint: 2006.04594. DOI: `https://doi.org/10.48550/arXiv.2006.04594`.

[62] J. Kelly et al. “Scalable *in situ* qubit calibration during repetitive error detection”. In: *Physical Review A* 94.3 (Sept. 26, 2016), p. 032321. DOI: `10.1103/PhysRevA.94.032321`.

[63] H. Bombin. “Topological subsystem codes”. In: *Physical Review A* 81.3 (Mar. 3, 2010), p. 032301. DOI: `10.1103/PhysRevA.81.032301`.

[64] Sergey Bravyi et al. “Subsystem surface codes with three-qubit check operators”. In: (2012). Publisher: arXiv Version Number: 2. DOI: `10.48550/ARXIV.1207.1443`.

Chapter 3

Multilevel Reset

This chapter reproduces the work published as *Removing leakage-induced correlated errors in superconducting quantum error correction* [1]. The supplementary information for this publication is reproduced in Appendix A

3.1 Abstract

Quantum computing can become scalable through error correction, but logical error rates only decrease with system size when physical errors are sufficiently uncorrelated. During computation, unused high energy levels of the qubits can become excited, creating leakage states that are long-lived and mobile. Particularly for superconducting transmon qubits, this leakage opens a path to errors that are correlated in space and time. Here, we report a reset protocol that returns a qubit to the ground state from all relevant higher level states. We test its performance with the bit-flip stabilizer code, a simplified version of the surface code for quantum error correction. We investigate the accumulation and dynamics of leakage during error correction. Using this protocol, we find lower rates of logical errors and an improved scaling and stability of error suppression with increasing qubit number. This demonstration provides a key step on the path towards scalable quantum computing.

Quantum computing can become scalable through error correction, but logical error rates only decrease with system size when physical errors are sufficiently uncorrelated. During computation, unused high energy levels of the qubits can become excited, creating leakage states that are long-lived and mobile. Particularly for superconducting transmon qubits, this leakage opens a path to errors that are correlated in space and time. Here, we report a reset protocol that returns a qubit to the ground state from all relevant higher level states. We test its performance with the bit-flip stabilizer code, a simplified version of the surface code for quantum error correction. We investigate the accumulation and dynamics of leakage during error correction. Using this protocol, we find lower rates of logical errors and an improved scaling and stability of error suppression with increasing qubit number. This demonstration provides a key step on the path towards scalable quantum computing.

3.2 Introduction

Quantum error correction stabilizes logical states by operating on arrays of physical qubits in superpositions of their computational basis states [25, 26, 65]. Superconducting transmon qubits are an appealing platform for the implementation of quantum error correction [49, 51, 67, 68, 69, 70, 71, 72, 73, 66]. However, the fundamental operations, such as single-qubit gates [74, 75], entangling gates [76, 53, 77, 78, 79], and measurement [80] are known to populate non-computational levels, creating a demand for a reset protocol [81, 82, 83, 84, 85, 86] that can remove leakage population from these higher states without adversely impacting performance in a large scale system. Directly quantifying leakage during normal operation presents another challenge, as optimizing measurement for detecting multiple levels is hard to combine with high speed and fidelity. This calls for analysis methods that use the errors detected during the stabilizer code's operation to find and visualize undesired correlated errors.

Here we introduce a multi-level reset gate using an adiabatic swap operation between the qubit and the readout resonator combined with a fast return. It requires only 250 ns to produce the ground state with a fidelity over 99%, with gate error accurately predicted by an intuitive semi-classical model. This fidelity is achieved simultaneously on all of the first three excited states for a single parameter choice. The gate is straightforward to calibrate and robust to drift due to the adiabaticity. Further, it uses only existing hardware as needed for normal operation and readout, and does not involve strong microwave drives that might induce crosstalk, making it attractive for large scale systems.

We benchmark the reset gate using the bit-flip error correction code [51] and measure growth and removal of leakage in-situ. By purposefully injecting leakage, we also quantify the gate's impact on errors detected in the code. Finally, we introduce a technique for computing the probabilities of error pairs, which allows identifying the distinctive patterns of correlations introduced by leakage. We find applying reset reduces the magnitude of correlations. We use these pair probabilities to inform the identification and correction of errors, improving the code's performance and stability over time.

3.3 Reset Gate Theory and Implementation

The multi-level reset gate consists of the three distinct stages dubbed "swap", "hold", and "return" (Fig. 3.1a). First, we swap all qubit excitations to the resonator by adiabatically sweeping the qubit frequency to ~ 1 GHz below the resonator frequency. We then hold the qubit below the resonator while excitations decay to the environment. Finally, we return the qubit diabatically to its initial frequency.

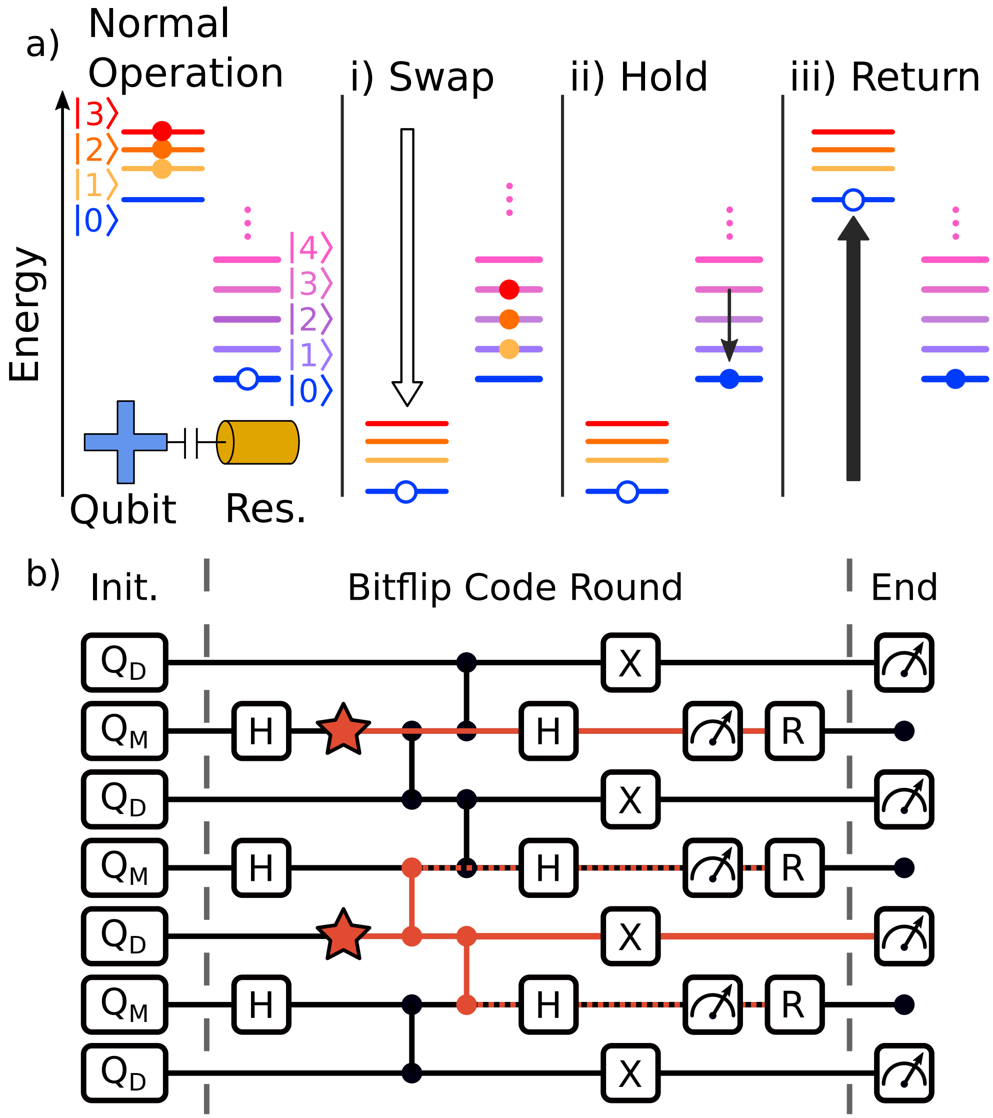


Figure 3.1: **Removing leakage with reset.** (a) Schematic of the multi-level reset protocol. The qubit starts with population in its first three excited states (closed circles), with the readout resonator in the ground state (open circle). (i) The qubit is swept adiabatically past the resonator to swap excitations. (ii) Resonator occupation decays to the environment while the qubit holds. (iii) After the resonator is sufficiently depleted, the qubit returns diabatically to its operating frequency. The total duration of the reset protocol is about 250 ns. (b) Circuit for the bit-flip stabilizer code including reset (R). Measure qubits (Q_M) cyclically apply parity measurements to neighbouring data qubits (Q_D) using Hadamard (H) and CZ gates. We add X gates to data qubits to depolarize energy relaxation error. When introducing reset, leakage errors (stars) may be removed from both measure and data qubits, either directly or via transport through the CZ gates (red lines).

Pulse engineering of the "swap" stage is critical to achieving efficient population transfer. We adopt a fast quasi-adiabatic approach [87], where the qubit frequency changes rapidly when far detuned from the resonator level crossing but changes slowly when near the level crossing. Since the frequency changes more slowly near the level crossing than a linear ramp, the probability of a diabatic error ${P_D}^{(s)}$ can be upper bounded by a Landau-Zener transition. This gives ${P_D}^{(s)} \ll \exp\left(-(2\pi g)^2 t_{\mathrm{swap}}/\Delta f\right) \sim 10^{-3}$, where $t_{\mathrm{swap}} = 30$ ns, $\Delta f = 2.5$ GHz is the total qubit frequency change and $g \approx 120$ MHz is the qubit-resonator coupling [35].

The "hold" stage of the protocol is primarily described by resonator photon decay. This decay follows $\exp(-\kappa t_{\mathrm{hold}}) \sim 10^{-3}$, with $t_{\mathrm{hold}} \sim 300$ ns and $\kappa \sim 1/(45\mathrm{ns})$ the resonator decay rate. The qubit's excitation number remains mostly unchanged during the hold below the resonator as Purcell decay [56] through the resonator is small. For swap durations below 30 ns the adiabaticity of the swap transition breaks down, and the system enters the "hold" stage in a superposition of the two adiabatic eigenstates. As a result, the probability undergoes coherent Rabi oscillations, which causes an incomplete reset and manifests itself as fringes.

If a single photon remains in the qubit-resonator system, the "return" stage of the protocol can be well described by a Landau-Zener transition. Achieving diabaticity is limited by the finite bandwidth of the control system. We can estimate an effective detuning velocity $\nu_r = \frac{1}{h}\frac{d}{dt}(E_{01} - E_{10}) = \Delta f/t_r$ using the typical ramp timescale $t_{\mathrm{r}} = 2$ ns. The probability of the desired diabatic transition is then $P_D^{(r)} = \exp[-(2\pi g)^2/\nu_r] \approx 0.6$. This description can be further extended to the multi-photon case using the Landau-Zener chain model [88].

Combining the semi-classical descriptions of each stage, we can identify two error channels in the reset of a single excitation. The first channel corresponds to the photon adiabatically swapping into the resonator, but then surviving over the hold time and adiabatically transitioning back to the qubit during the return. This is the dominant error channel, with probability $(1 - {P_D}^{(s)})e^{-\kappa t_{\text{hold}}}(1 - P_D^{(r)}) \sim 5 \cdot 10^{-4}$. The second channel corresponds to a failed initial swap of the qubit photon, followed by a diabatic transition during the return. The probability of this error is small, approximately ${P_D}^{(s)} P_D^{(r)} \ll 10^{-4}$. The reset dynamics of the $|2\rangle$ and $|3\rangle$ states is similar, with multiple adiabatic transitions moving 2 and 3 photons to the resonator respectively, after which they undergo rapid decay.

3.4 Experimental realization of the Multilevel Reset Gate

We experimentally test our reset gate on a Sycamore processor [35], consisting of an array of flux-tunable superconducting transmon qubits [49, 89] with tunable couplers [35, 54, 53, 52]. Each qubit is coupled to a readout resonator with strength $g \approx 120$ MHz, and having a frequency ~ 1.5 GHz below the qubit. Resonators are coupled to the outside environment through a Purcell filter [90].

The reset gate is implemented using flux-tuning pulses to steer the qubit's frequency to interact with the resonator, see Fig. 3.2a. The selected qubit has an idle frequency of 6.09 GHz and a nonlinearity of -200 MHz. The qubit starts at its idle frequency, moves past the resonator at 4.67 GHz, and is held 1 GHz below it, followed by a fast return to the idle frequency. We define the reset error as the likelihood of producing any state other than the ground state. The dependence of reset error on swap duration is shown in Fig. 3.2b for the cases when the qubit is initialized to $|1\rangle$, $|2\rangle$, and $|3\rangle$. We find that the reset error for all of the initialized states decreases until it reaches a readout visibility floor at about 30 ns swap duration. This floor of $\sim 0.2\%$ was also measured

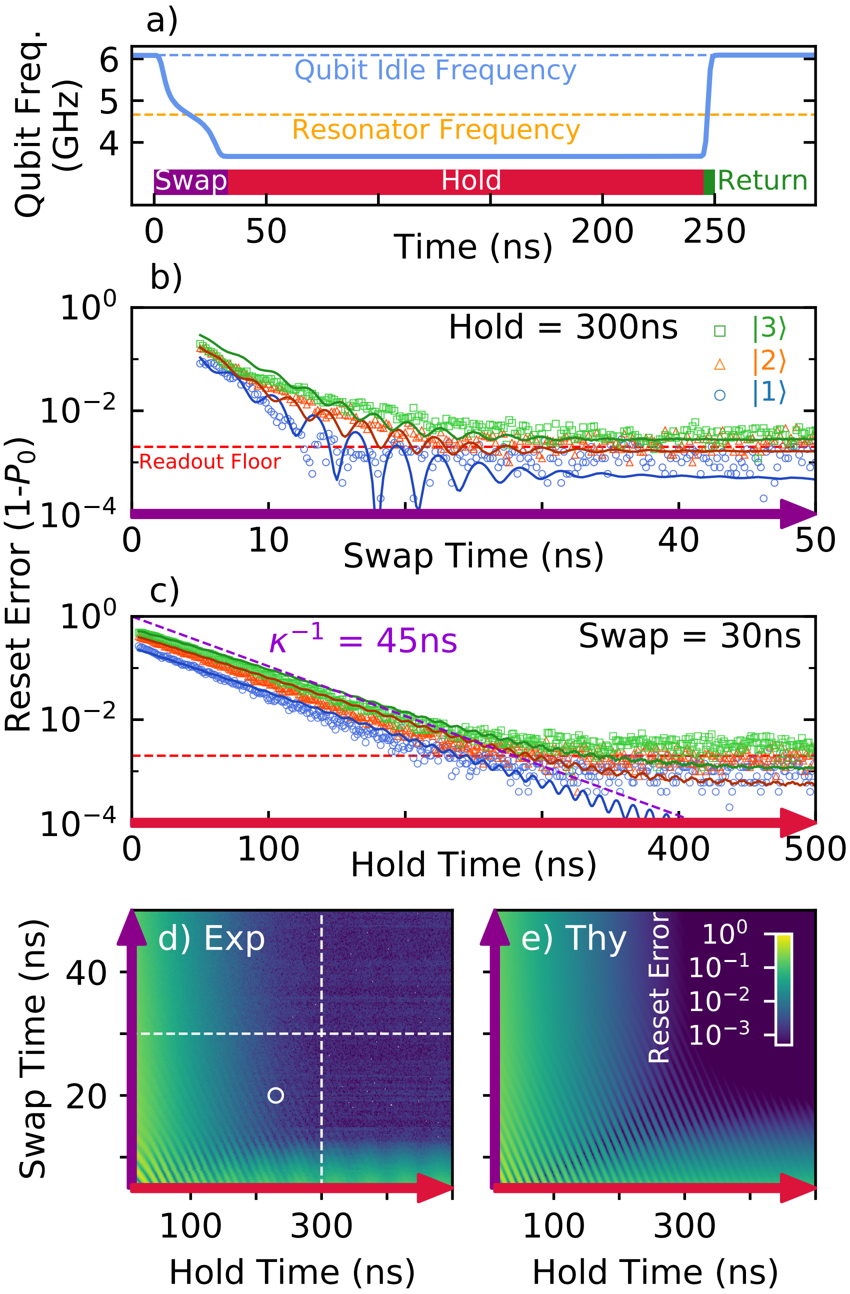


Figure 3.2: **Reset gate benchmarking.** (a) The qubit frequency trajectory for implementing reset consists of three stages. We plot the ground state infidelity when resetting the first three excited states of the qubit versus swap (b) and vs hold times (c). We include experimental data (points) and theory prediction (solid lines). Reset error versus swap and hold for experiment (d) and theory (e) show a wide range of optimal parameters. Dashed white lines indicate linecuts for (b) and (c). White circle indicates the point of operation.

independently as the ground state measurement error after heralding; postselecting on a prior measurement of $|0\rangle$. This indicates that the floor is intrinsic to the measurement, not to the reset gate itself. We notice oscillations in the data which arise from an incomplete swap and are reproduced by the theoretical model results. In Fig. 3.2c, we keep the swap duration fixed at 30 ns and vary the hold duration. We find that the reset error decreases exponentially until it reaches the readout visibility floor, with a decay that is compatible with $1/\kappa = 45$ns. We show the landscape of the reset error for the qubit initialized in $|1\rangle$, experimentally in Fig. 3.2d, and the model results in Fig. 3.2e. For a wide choice of parameters above a minimum swap and hold duration, the ground state can be achieved with high fidelity: Experimentally we are limited by readout and theoretically the deviation from the ground state is below 10^{-3}. We also note that the majority of error is favorably in the computational basis, which stabilizer codes can naturally identify and correct. The landscape involving other parameters can be found in the Supplementary Note 1.

The data and model results in Fig. 3.2 show that one can reset a qubit within 250 ns to the ground state with an error of around 10^{-3}. Moreover, the insensitivity to parameter choice, stemming from the adiabaticity of the gate, highlights the protocol's robustness to drift and noise. This makes it amenable for use in large-scale systems. Finally, the demonstrated ability to simultaneously remove occupation from the $|1\rangle$, $|2\rangle$, and $|3\rangle$ states for a single choice of parameters makes this protocol a prime candidate for mitigating leakage in quantum error correction.

3.5 Integrating Reset into the Bit-flip Code

We now benchmark this protocol in the bit-flip stabilizer code [51], a precursor to the surface code. Here, a fast cycle of Hadamard, entangling, and measurement gates is repeated (Fig. 3.1b) to extract parity measurements to stabilize the logical state. We note the addition of X gates on the data qubits to depolarize energy relaxation error. Since the reset protocol is designed to unconditionally prepare the ground state, and thus remove all quantum data, we apply it only on the measure qubits immediately after readout. In the absence of reset, we apply no feedback to the measure qubit state but account for this during syndrome decoding as in [51].

We implement a 21 qubit chain on a Sycamore processor (inset of Fig. 3.3). The qubits chosen had an average T_1 near 14 μs, with their experimental parameters chosen by optimization [61]. We start by directly measuring the growth of leakage to $|2\rangle$ by running the code for a number of rounds and terminating with a measurement that can resolve $|2\rangle$ on all qubits. Each round is 955 ns long when we include reset. We note that the leakage population is subject to a different readout floor than seen in Fig. 3.2, as further detailed in the Supplementary Note 2. We average over 40 random initial states for the data qubits, and find that the population of $|2\rangle$ grows and saturates. In the absence of reset, the measure qubits build up a larger $|2\rangle$ state population than the data qubits.

We fit a simple rate equation model and calculate the leakage ($\gamma_\uparrow$) and decay ($\gamma_\downarrow$) rates for the $|2\rangle$ state population [75]. Applying reset to the measure qubits breaks the established pattern of growth and requires a different fitting procedure, detailed in Supplementary Note 3. We find a forty-fold increase in $\gamma_\downarrow$ on average for measure qubits with the addition of reset. We also find a 2.4x increase in $\gamma_\downarrow$ on average for data qubits, indicating transport of leakage population from data to measure qubits. We understand this effect as arising naturally in our CZ gate [54], which requires a condition that also places $|21\rangle$ and $|03\rangle$ on resonance, where the $|2\rangle$ is on the lower frequency qubit. Where a data qubit is below the measure qubit in frequency, transport of $|2\rangle$ from the data qubit to $|3\rangle$ in the measure qubit can occur, where it is subsequently removed by reset.

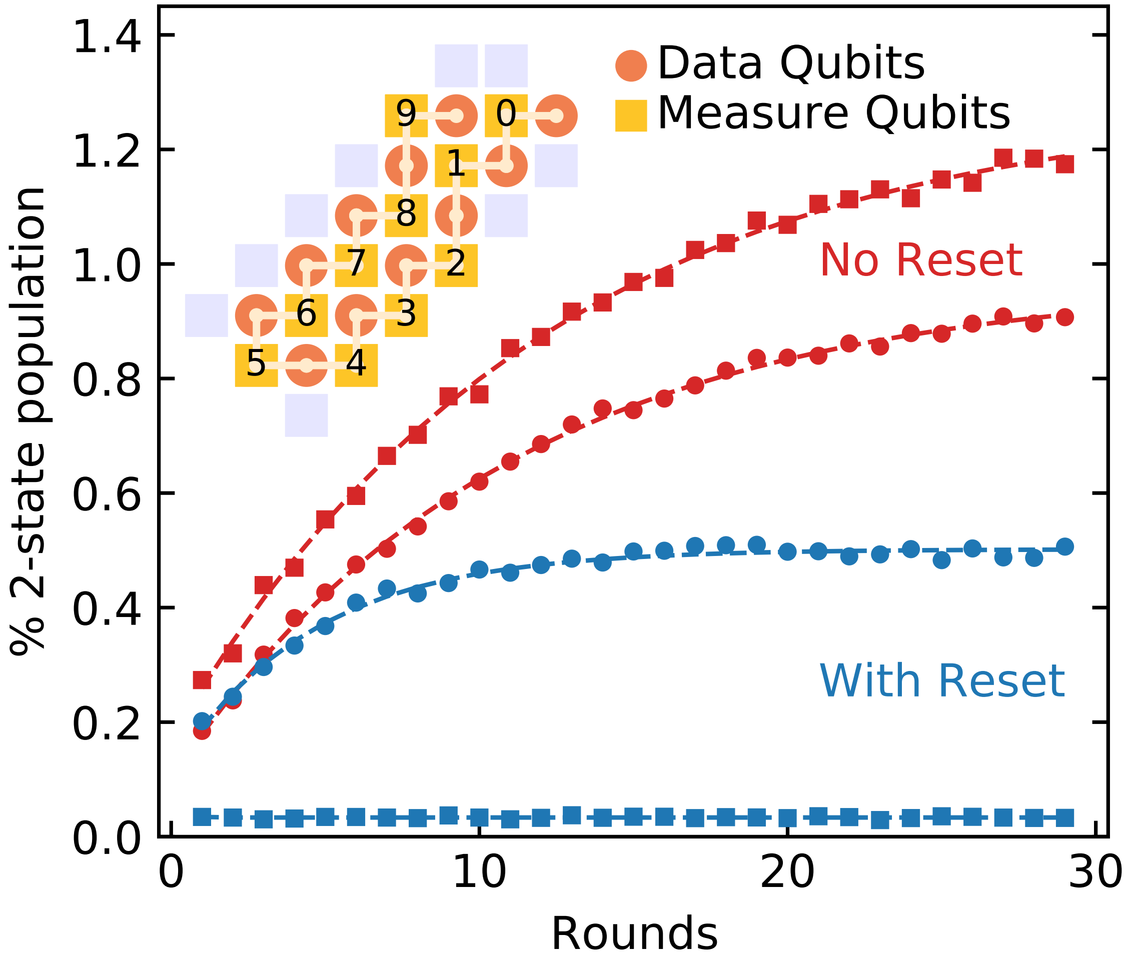


Figure 3.3: **Leakage during the Bit-flip code.** The growth in $|2\rangle$ population vs. stabilizer code length. The circuit is run for a number of rounds and terminated with a readout sensitive to $|2\rangle$ population. The experimental data is averaged over measure or data qubits and fitted to an exponential (dashed lines) to extract rates. Further data is included in Supplementary Note 3. The inset shows the 21 qubit chain as implemented on the Sycamore device.

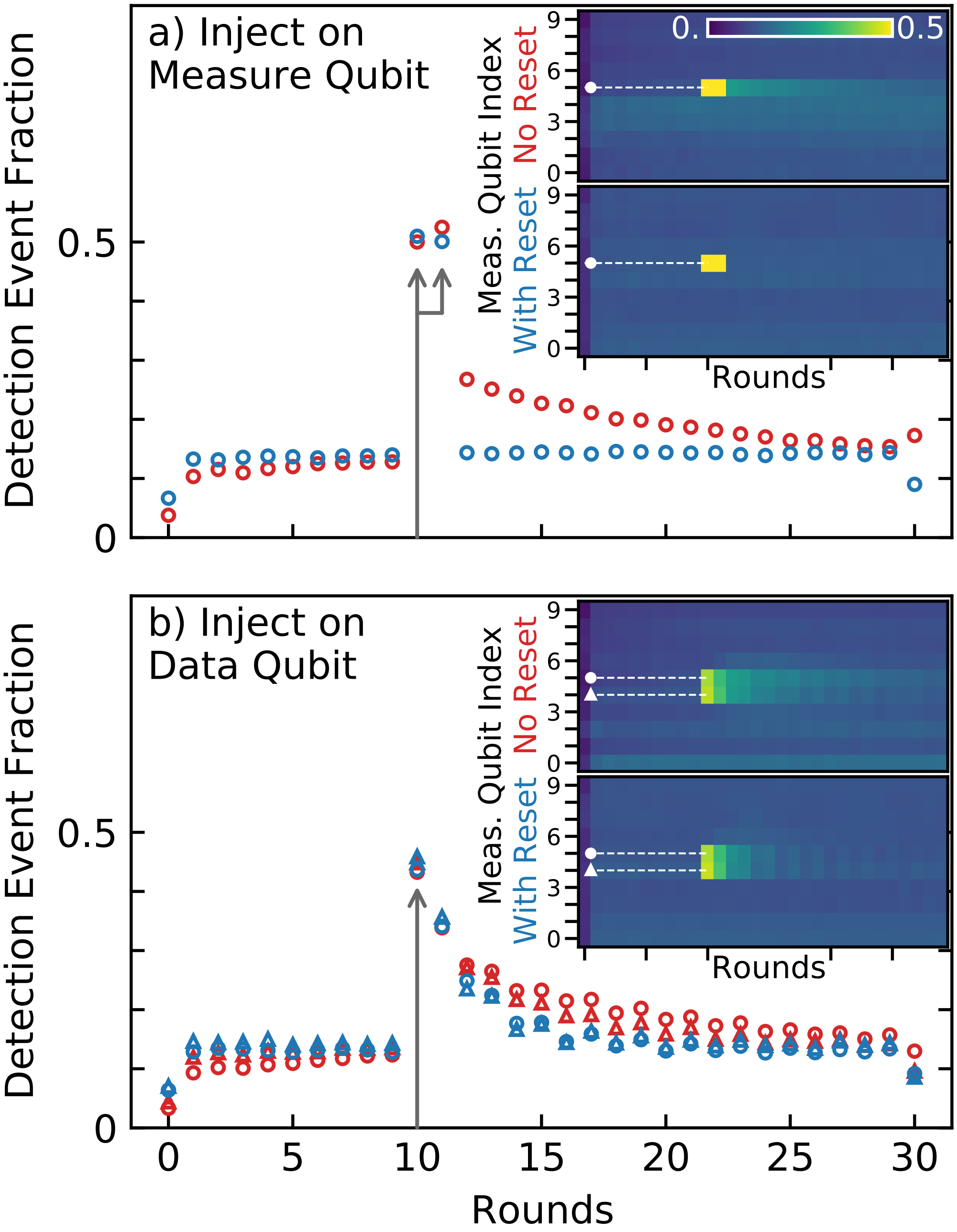


Figure 3.4: **Injection of leakage.** Detection event fraction when a full $|1\rangle \to |2\rangle$ rotation is inserted in round 10 after the first Hadamards (a) on measure qubit 5 and (b) on the data qubit between measure qubits 4 (circles) and 5 (triangles). Insets show the event fraction across all measure qubits, indicating the traces plotted in the main figure (dashed lines). See Fig. 3.3 inset for qubit locations.

3.6 Injection of Leakage

To visualize the pattern of errors that leakage produces, we now inject $|2\rangle$ into the stabilizer code at specific locations. We insert a complete rotation between $|1\rangle$ and $|2\rangle$ on a single qubit immediately after the first Hadamard gates in round 10 of a 30 round experiment. As the data qubits are in either $|0\rangle$ or $|1\rangle$, and measure qubits are in an equal superposition of $|0\rangle$ and $|1\rangle$ after the Hadamard, the amount of injected $|2\rangle$ is the same for both measure and data qubits on average. Fig. 4 shows the fraction of error detection events, which represents the portion of runs where a given stabilizer measurement reports an unexpected result, indicating an error occurred [51]. Injected leakage produces two distinct effects; a pair of detection events at injection, and a tail of correlated detection events over the lifetime of the leakage state. As with discrete bit-flip errors, the initial pairs of detection events appear sequentially in time for injection on measure qubits, while for data qubits both adjacent measure qubits report error (gray arrows).

The detection event fractions for all qubits are shown in the insets and cross-sections are shown in the main figure. We note that the value of the detection event fraction deviates for the first round due to initialization, and for the last round as data qubit measurements are involved [51]. As can be seen in Fig. 4a, the insertion of leakage in measure qubit 5 (see inset of Fig. 3 for its location) creates two adjacent peaks at a detection event fraction of 0.5, as the injection produces a random readout result in round 10. This is followed by a clear tail of anomalously high levels of detection events that slowly decays over many rounds, indicating errors that are correlated in time. When applying reset, the errors on all measure qubits are more uniform, and the increase in detection events for the first nine rounds becomes flattened. Importantly, the slow decay in errors is no longer visible as the detection event fraction drops to the baseline immediately after the initial pair of detection events. We also insert leakage in the data qubit between measure qubits 4 and 5, see Fig. 4b. We again notice an increase of detection events that slowly decays, now on both neighbouring measure qubits. The error decreases more rapidly with reset, corroborating our prior observation that higher level states can migrate to measure qubits. In addition, we notice a small increase in detection events around the leakage injection in qubits 3 and 6 in the case of no reset, further indicating that higher level states can move between qubits. We notice for both cases a small odd-even oscillation in the data, which we understand as arising from the fact that the injected $|1\rangle$ to $|2\rangle$ rotation does not affect the data qubit when it is in state $|0\rangle$. Since the X gates on data qubits swap $|0\rangle$ and $|1\rangle$ in each round, we see a higher likelihood of bit error from energy relaxation in odd rounds after the injection.

The data in Fig. 4 show that the reset protocol can remove large populations of leakage in measure qubits and helps to decrease leakage in data qubits, thereby strongly suppressing time-correlated tails of detection events. This result also raises the question how higher level state occupations that naturally arise during the stabilizer codes lead to correlated errors.

3.7 Leakage-induced Correlations in the Bit-flip Code

To further quantify this, we analyze the correlations between detection events that arise during normal code operation using the error graph [51], see Fig. 3.5a. We model detection events as arising from independent random processes that flip pairs of measurements [2]. The probability p_{ij} of the process that flips measurements i and j can be obtained from the observed correlations between detection events,

$$p_{ij} = \frac{1}{2} - \frac{1}{2}\sqrt{1 - \frac{4\left(\langle x_i x_j\rangle - \langle x_i\rangle\langle x_j\rangle\right)}{1 - 2\langle x_i\rangle - 2\langle x_j\rangle + 4\langle x_i x_j\rangle}}, \tag{3.1}$$

where $x_i = 1$ if there is a detection event at a given measurement i and $x_i = 0$ otherwise. Here $\langle x \rangle$ denotes averaging x over many experimental realizations.

In Fig. 3.5, we visualize p_{ij} and show autocorrelations for measure qubit 6 and cross correlations between measure qubits 5 and 6. The standard error correction model assumes that detection events occur only in local pairs. For detection events occurring on the same measure qubit, we expect only correlations between adjacent rounds, corresponding to elements adjacent to the main diagonal ($p_{i,i\pm1}$) of Fig. 3.5b,c. For detection events occurring on neighboring measure qubits, we expect only correlations between the qubits in the same round, or adjacent rounds due to the staggered placement of CZ gates. This corresponds to non-zero elements only on and immediately below the main diagonal of Fig. 3.5d,e. In contrast, without reset, we find that significant unexpected correlations appear (left panels), covering distances of over 10 rounds. With reset, these long-range correlations are mostly removed (right panels). This reveals an underlying checkerboard pattern which arises similarly to the aforementioned odd-even oscillations (see Supplementary Note 5).

3.8 Logical performance and Λ_{bit}

Having shown the reset protocol removes leakage and suppresses long-distance correlations, we now look at logical error rates. We run the stabilizer code to a given number of rounds, and feed the detection events into a minimum weight perfect matching algorithm [91] that identifies and keeps track of errors to return the corrected logical state.

We perform the experiment on 21 qubits, and use subsampling to evaluate performance for smaller subsets of the code (See Supplementary Notes 6 and 7 for details). We use the p_{ij} elements to set the weights for the matching algorithm. We convert the probability of a logical error (P_L) at a given number of rounds k to a logical error

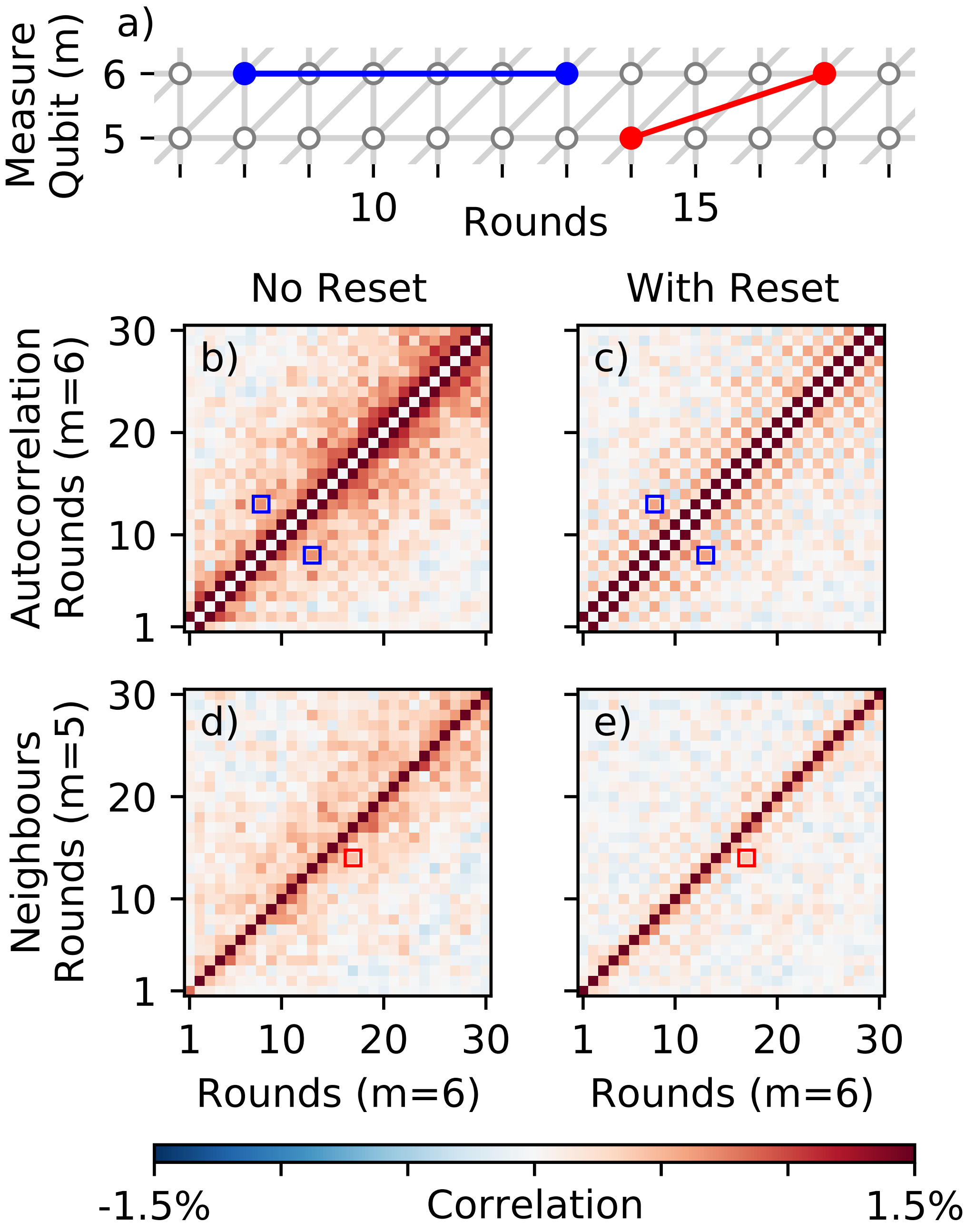


Figure 3.5: **Correlations caused by leakage.** p_{ij} matrices show the strength of non-local correlations in the detected errors. These undesired correlations are significantly reduced with the addition of reset. (a) The error graph for the bit-flip code, highlighting examples of non-local correlations on both space and time, indicating their corresponding p_{ij} elements below (boxes). (b,c) Time-correlations on measure qubit 6, with and without reset. (d,e) Cross-correlations between measure qubits 5 and 6, with and without reset.

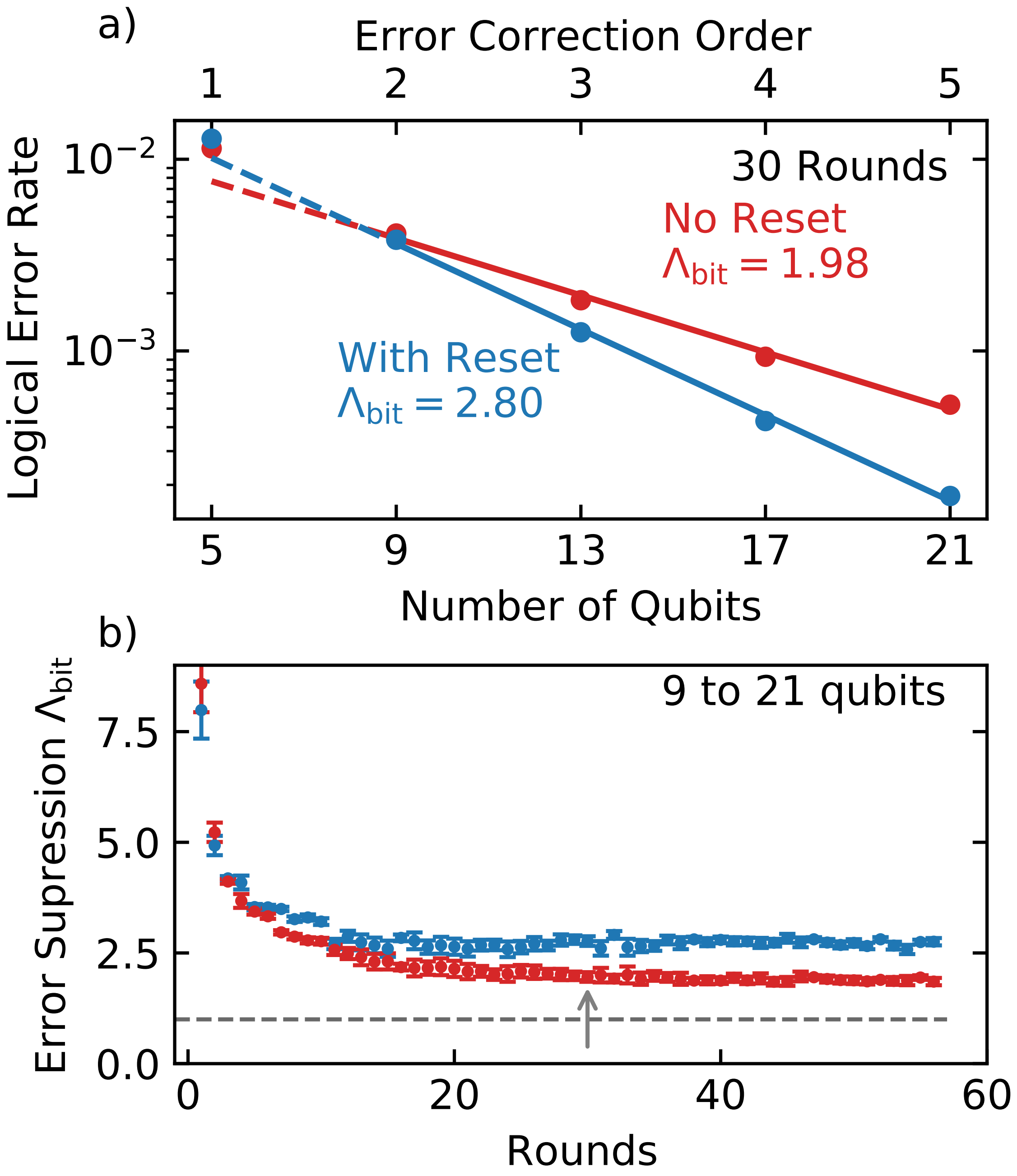


Figure 3.6: **Logical code performance.** (a) The logical error rate for 30 rounds vs system size. The error suppression factor Λ_{bit} is fitted to the data from nine qubits up. (b) Λ_{bit} versus code depth, showing that with reset logical error suppression is improved consistently. The error bars indicate the 1 standard deviation error in the fit of error rate versus number of qubits. The threshold for the bit-flip code (unity) is shown as a dashed line. The arrow indicates the data in (a).

rate $\epsilon = [1 - (1 - 2P_L)^{(1/k)}]/2$ [92] for the number of rounds k, shown at 30 rounds in Fig. 3.6a. Here, the logical error rate is plotted from 5 to 21 qubits, corresponding to an error correction order of $n = 1$ to 5, meaning at least $n + 1$ errors must occur to cause a logical error. The error rate of the bit-flip code in the absence of correlations should be exponentially suppressed with $\epsilon \propto 1/\Lambda_{\text{bit}}^{n+1}$.

We find that the logical error rate decreases with number of qubits, with an exponential dependence from 9 qubits up. The data at 5 qubits shows degraded logical performance, which we attribute to relatively narrow code width impacting the performance of syndrome decoding. [43]. As including these points would reduce the quality of the fit and artificially increase the reported value of Λ_{bit}, we exclude them.

We plot Λ_{bit} versus rounds in Fig. 3.6b. A constant logical error rate should produce a Λ_{bit} that is independent of round number. In practice, effects including the buildup of leakage, the thermalization of data qubits, and short time boundary effects will produce a higher apparent Λ_{bit} prior to saturation. Without reset, we observe Λ_{bit} decaying over 30 rounds toward a saturation value of 1.98. With reset, Λ_{bit} stabilizes faster, within 10 rounds, to a higher value of 2.80. Notably, error suppression is enhanced despite the time added to the cycle by reset, where data qubits are exposed to additional decoherence. This highlights the importance of removing the time-correlated errors induced by leakage, as seen in Fig 3.5.

We observe the logical performance stabilizing to values of $\Lambda_{\text{bit}} > 1$, and that the addition of reset improves both the long-time performance and rate with which the code approaches this value. Moreover, we see deviations from ideal behaviour where experiments are small in number of qubits or rounds. This highlights that error suppression is a property that asymptotically emerges with space and time.

In summary, we introduce a reset protocol that uses existing hardware to remove higher level states and test it using the bit-flip stabilizer code. We show that reset mitigates leakage-induced long-time correlated errors and significantly improves logical error suppression. While optimizing gates and readout to have minimal leakage is a necessary strategy, the correlated nature of the error that leakage induces makes reset protocols critical for practical quantum error correction.

References

[1] Matt McEwen et al. “Removing leakage-induced correlated errors in superconducting quantum error correction”. In: *Nature Communications* 12.1 (Dec. 2021), p. 1761. DOI: `10.1038/s41467-021-21982-y`.

[2] Google Quantum AI et al. “Exponential suppression of bit or phase errors with cyclic error correction”. In: *Nature* 595.7867 (July 15, 2021). _eprint: 2102.06132, pp. 383–387. DOI: `https://doi.org/10.1038/s41586-021-03588-y`.

[25] S. B. Bravyi and A. Yu. Kitaev. “Quantum codes on a lattice with boundary”. In: (1998). Publisher: arXiv Version Number: 1. DOI: `10.48550/ARXIV.QUANT-PH/9811052`.

[26] Austin G. Fowler et al. “Surface codes: Towards practical large-scale quantum computation”. In: *Physical Review A* 86.3 (Sept. 18, 2012). Publisher: American Physical Society, p. 032324. DOI: `10.1103/physreva.86.032324`.

[35] Frank Arute et al. “Quantum supremacy using a programmable superconducting processor”. In: *Nature* 574.7779 (Oct. 24, 2019). Publisher: Nature Publishing Group, pp. 505–510. DOI: `10.1038/s41586-019-1666-5`.

[43] Austin G. Fowler and John M. Martinis. “Quantifying the effects of local many-qubit errors and nonlocal two-qubit errors on the surface code”. In: *Physical Re-*

view A 89.3 (Mar. 12, 2014). Publisher: American Physical Society, p. 032316. DOI: `10.1103/PhysRevA.89.032316`.

[49] Jens Koch et al. "Charge-insensitive qubit design derived from the Cooper pair box". In: *Physical Review A* 76.4 (Oct. 12, 2007). Publisher: American Physical Society, p. 042319. DOI: `10.1103/PhysRevA.76.042319`.

[51] J. Kelly et al. "State preservation by repetitive error detection in a superconducting quantum circuit". In: *Nature* 519.7541 (Mar. 5, 2015). Publisher: Nature Publishing Group, pp. 66–69. DOI: `10.1038/nature14270`.

[52] Charles Neill. "A path towards quantum supremacy with superconducting qubits". PhD thesis. University of California, Santa Barbara, 2017.

[53] Fei Yan et al. "Tunable Coupling Scheme for Implementing High-Fidelity Two-Qubit Gates". In: *Physical Review Applied* 10.5 (Nov. 28, 2018). Publisher: American Physical Society, p. 054062. DOI: `10.1103/PhysRevApplied.10.054062`.

[54] B. Foxen et al. "Demonstrating a Continuous Set of Two-qubit Gates for Near-term Quantum Algorithms". In: *Physical Review Letters* 125.12 (Sept. 15, 2020). _eprint: 2001.08343, p. 120504. DOI: `https://doi.org/10.1103/PhysRevLett.125.120504`.

[56] Eyob A. Sete, John M. Martinis, and Alexander N. Korotkov. "Quantum theory of a bandpass Purcell filter for qubit readout". In: *Physical Review A* 92.1 (July 21, 2015). Publisher: American Physical Society, p. 012325. DOI: `10.1103/PhysRevA.92.012325`.

[61] Paul V. Klimov et al. "The Snake Optimizer for Learning Quantum Processor Control Parameters". In: (2020). _eprint: 2006.04594. DOI: `https://doi.org/10.48550/arXiv.2006.04594`.

[65] Barbara M. Terhal. "Quantum error correction for quantum memories". In: *Reviews of Modern Physics* 87.2 (Apr. 7, 2015). Publisher: American Physical Society, pp. 307–346. DOI: `10.1103/RevModPhys.87.307`.

[66] Christian Kraglund Andersen et al. "Repeated quantum error detection in a surface code". In: *Nature Physics* 16.8 (Aug. 2020). Publisher: Springer Science and Business Media LLC, pp. 875–880. DOI: `10.1038/s41567-020-0920-y`.

[67] Jerry M. Chow et al. "Implementing a strand of a scalable fault-tolerant quantum computing fabric". In: *Nature Communications* 5.1 (Sept. 2014). Publisher: Nature Publishing Group, p. 4015. DOI: `https://doi.org/10.1038/ncomms5015`.

[68] A. D. Córcoles et al. "Detecting arbitrary quantum errors via stabilizer measurements on a sublattice of the surface code". In: *Nat. Commun.* 6 (2014). Publisher: Nature Publishing Group, p. 6979. DOI: `https://doi.org/10.48550/arXiv.1410.6419`.

[69] Maika Takita et al. "Demonstration of Weight-Four Parity Measurements in the Surface Code Architecture". In: *Physical Review Letters* 117.21 (Nov. 18, 2016). Publisher: American Physical Society, p. 210505. DOI: `10.1103/PhysRevLett.117.210505`.

[70] D. Nigg et al. "Quantum computations on a topologically encoded qubit". In: *Science* 345.6194 (July 18, 2014). Publisher: American Association for the Advancement of Science, pp. 302–305. DOI: `10.1126/science.1253742`.

[71] D. Ristè et al. "Detecting bit-flip errors in a logical qubit using stabilizer measurements". In: *Nature Communications* 6.1 (Nov. 2015). Publisher: Nature Publishing Group, p. 6983. DOI: `10.1038/ncomms7983`.

[72] M. D. Reed et al. "Realization of three-qubit quantum error correction with superconducting circuits". In: *Nature* 482.7385 (Feb. 2012). Publisher: Springer Science and Business Media LLC, pp. 382–385. DOI: `10.1038/nature10786`.

[73] Christian Kraglund Andersen et al. "Entanglement stabilization using ancilla-based parity detection and real-time feedback in superconducting circuits". In: *npj Quantum Information* 5.1 (Dec. 2019). Publisher: Springer Science and Business Media LLC, p. 69. DOI: `10.1038/s41534-019-0185-4`.

[74] F. Motzoi et al. "Simple Pulses for Elimination of Leakage in Weakly Nonlinear Qubits". In: *Physical Review Letters* 103.11 (Sept. 8, 2009). Publisher: American Physical Society, p. 110501. DOI: `10.1103/PhysRevLett.103.110501`.

[75] Zijun Chen et al. "Measuring and Suppressing Quantum State Leakage in a Superconducting Qubit". In: *Physical Review Letters* 116.2 (Jan. 13, 2016). Publisher: American Physical Society, p. 020501. DOI: `10.1103/PhysRevLett.116.020501`.

[76] R. Barends et al. "Superconducting quantum circuits at the surface code threshold for fault tolerance". In: *Nature* 508.7497 (Apr. 2014). Publisher: Nature Publishing Group, pp. 500–503. DOI: `https://doi.org/10.1038/nature13171`.

[77] M. A. Rol et al. "Fast, High-Fidelity Conditional-Phase Gate Exploiting Leakage Interference in Weakly Anharmonic Superconducting Qubits". In: *Physical Review Letters* 123.12 (Sept. 18, 2019). Publisher: American Physical Society, p. 120502. DOI: `10.1103/PhysRevLett.123.120502`.

[78] V. Negîrneac et al. "High-Fidelity Controlled- Z Gate with Maximal Intermediate Leakage Operating at the Speed Limit in a Superconducting Quantum Processor". In: *Physical Review Letters* 126.22 (June 4, 2021). _eprint: 2008.07411, p. 220502. DOI: `https://doi.org/10.1103/PhysRevLett.126.220502`.

[79] Sumeru Hazra et al. “Engineering cross resonance interaction in multi-modal quantum circuits”. In: *Applied Physics Letters* 116.15 (Apr. 13, 2020). Publisher: AIP Publishing, p. 152601. DOI: `10.1063/1.5143440`.

[80] Daniel Sank et al. “Measurement-Induced State Transitions in a Superconducting Qubit: Beyond the Rotating Wave Approximation”. In: *Physical Review Letters* 117.19 (Nov. 4, 2016). Publisher: American Physical Society, p. 190503. DOI: `10.1103/physrevlett.117.190503`.

[81] M. D. Reed et al. “Fast reset and suppressing spontaneous emission of a superconducting qubit”. In: *Applied Physics Letters* 96.20 (May 17, 2010). Publisher: AIP Publishing, p. 203110. DOI: `10.1063/1.3435463`.

[82] K. Geerlings et al. “Demonstrating a Driven Reset Protocol for a Superconducting Qubit”. In: *Physical Review Letters* 110.12 (Mar. 20, 2013). Publisher: American Physical Society, p. 120501. DOI: `10.1103/PhysRevLett.110.120501`.

[83] Martin Suchara, Andrew W. Cross, and Jay M. Gambetta. “Leakage Suppression in the Toric Code”. In: *Quantum Inf. Comput.* 15.11 (2014), pp. 997–1016. DOI: `https://doi.org/10.48550/arXiv.1410.8562Focustolearnmore`.

[84] P. Magnard et al. “Fast and Unconditional All-Microwave Reset of a Superconducting Qubit”. In: *Physical Review Letters* 121.6 (Aug. 7, 2018). Publisher: American Physical Society, p. 060502. DOI: `10.1103/PhysRevLett.121.060502`.

[85] C. C. Bultink et al. “Protecting quantum entanglement from leakage and qubit errors via repetitive parity measurements”. In: *Science Advances* 6.12 (Mar. 20, 2020). Publisher: American Association for the Advancement of Science (AAAS), eaay3050. DOI: `10.1126/sciadv.aay3050`.

[86] Boris Mihailov Varbanov et al. “Leakage detection for a transmon-based surface code”. In: *npj Quantum Information* 6.1 (Dec. 2020). Publisher: Springer Science and Business Media LLC, p. 102. DOI: `10.1038/s41534-020-00330-w`.

[87] John M. Martinis and Michael R. Geller. “Fast adiabatic qubit gates using only \sigma z control”. In: *Physical Review A* 90.2 (Aug. 8, 2014), p. 022307. DOI: `10.1103/PhysRevA.90.022307`.

[88] N. A. Sinitsyn, J. Lin, and V. Y. Chernyak. “Constraints on scattering amplitudes in multistate Landau-Zener theory”. In: *Phys. Rev. A* 95.1 (2016). Publisher: American Physical Society, p. 012140. DOI: `https://doi.org/10.48550/arXiv.1609.06285`.

[89] Morten Kjaergaard et al. “Superconducting Qubits: Current State of Play”. In: *Annual Review of Condensed Matter Physics* 11.1 (Mar. 10, 2020), pp. 369–395. DOI: `10.1146/annurev-conmatphys-031119-050605`.

[90] Evan Jeffrey et al. “Fast Accurate State Measurement with Superconducting Qubits”. In: *Physical Review Letters* 112.19 (May 15, 2014). Publisher: American Physical Society, p. 190504. DOI: `10.1103/PhysRevLett.112.190504`.

[91] Austin G. Fowler, Adam C. Whiteside, and Lloyd C. L. Hollenberg. “Towards practical classical processing for the surface code: Timing analysis”. In: *Physical Review A* 86.4 (Oct. 12, 2012). Publisher: American Physical Society, p. 042313. DOI: `10.1103/PhysRevA.86.042313`.

[92] T. E. O’Brien, B. Tarasinski, and L. DiCarlo. “Density-matrix simulation of small surface codes under current and projected experimental noise”. In: *npj Quantum Information* 3.1 (Dec. 2017). Publisher: Nature Publishing Group, p. 39. DOI: `https://doi.org/10.1038/s41534-017-0039-x`.

Chapter 4

Complete Leakage Removal

This chapter reproduces the work submitted for publication as *Complete Leakage Removal in the Surface Code on Superconducting Qubits* [6], in close collaboration with Kevin Miao. The supplementary information for this publication is reproduced in Appendix B

4.1 Abstract

Leakage of quantum information out of computational states into higher energy states represents a major challenge in the pursuit of quantum error correction (QEC). In a QEC circuit, leakage builds over time and spreads through multi-qubit interactions. This leads to correlated errors that degrade the exponential suppression of logical error with scale, challenging the applicability of QEC as a path towards fault-tolerant quantum computation. Here, we demonstrate the execution of a distance-3 surface code and distance-21 bit-flip code on a Sycamore quantum processor where leakage is removed from all qubits in each cycle. This shortens the lifetime of leakage and curtails its ability to spread and induce correlated errors. We report a ten-fold reduction in steady-state leakage population on the data qubits encoding the logical state and an average leakage

population of less than 1×10^{-3} throughout the entire device. The leakage removal process itself efficiently restores leakage population back to the computational states. Adding it to a code circuit prevents leakage inducing correlated error across cycles, restoring a fundamental assumption of surface code QEC. With this demonstration that leakage can be contained, we resolve a key challenge for practical QEC at scale.

4.2 Introduction

Quantum error correction (QEC) promises to exponentially suppress common errors in quantum computing devices, bridging the gap between achievable physical error rates and the low logical rates required for useful quantum algorithms [25, 26, 65]. The surface code is a leading candidate for experimental implementations of QEC, where a repetitive stabilizer circuit protects a logical qubit state.

Superconducting transmon qubits [49, 89] represent a leading platform for implementing surface code QEC, with recent demonstrations of architectures compatible with QEC and capable of scaling [72, 67, 68, 71, 51, 69, 73, 66, 2, 37, 36, 4]. However, transmon qubits feature small separation from the transition between *computational* states to the nearby transitions to non-computational states, so-called *leakage states*. These transitions are therefore difficult to avoid, with leakage states being erroneously populated by single-qubit gates [74, 75], entangling gates [76, 52, 53, 77, 78], and measurement [80, 93].

Leakage is particularly dangerous in the context of QEC [94, 43, 86, 85]. A key underlying assumption of QEC is that the errors to be suppressed are sufficiently uncorrelated in both space and time. Contrary to this requirement, a qubit in a leakage state can induce errors on multiple neighbouring qubits, even causing them to leak as well. The correlated spread of errors through the device represents a major problem for QEC experiments. Identifying and post-selecting out leakage events has permitted cutting-edge experiments on the surface code [36, 37], and partial leakage removal has been integrated into surface code circuits [2, 4]. However, all these experiments displayed a characteristic rise in the number of detected errors as the code progressed, indicative of accumulating leakage population in the device. A demonstration of leakage removal from all qubits in a surface code circuit has not yet been reported. Further, stabilizing the leakage populations such that error rates do not grow over time is a requirement for scalable QEC, and this remains an important open challenge.

Here, we study and remove the effects of leakage in a surface code circuit on an array of transmon qubits. First, we detail the dynamics of leakage in the QEC circuit and the spread of error through space and time. We quantify the effect of leaked qubits undergoing multi-qubit interactions, which is the primary vehicle for spatial propagation of leakage. Second, we demonstrate the effective removal of leakage from all qubits involved in the surface code circuit. We show residual leakage populations averaged over all qubits are suppressed to below 0.1%, and do not grow as the code is extended in time. Finally, we show that removing leakage improves logical performance. Using a distance-21 bit-flip code with leakage removal, injected leakage impacts logical performance equivalently to injected Pauli errors. This confirms that leakage removal is effective in suppressing the correlated nature of leakage-induced errors. Then, using a distance-3 surface code, we show that leakage removal both decreases the rate of logical errors and prevents the code performance from declining over time, proving that QEC can be stable over long time periods. We extrapolate this behaviour to lower error rates, finding that injected leakage now effects logical error rates in the same fashion as uncorrelated Pauli errors. In summary, leakage removal resolves an important obstacle to growing QEC to algorithmically relevant scales.

4.3 Characterizing the spread of leakage

Leakage states are particularly problematic in structured QEC circuits because they are long-lived and spread through the device, inducing correlated errors in both time and space. The surface code circuit shown in Figure 4.1b shows a single cycle, which consists of a number of moments. Four such moments correspond to CZ gates used to measure the surface code stabilizer. When a qubit in the circuit leaks, subsequent gates involving that qubit produce additional errors.

Figure 4.1c illustrates the dynamics of leakage in a distance-3 surface code circuit. At the round labeled 0, we inject a full $|1\rangle \rightarrow |2\rangle$ rotation on the central data qubit, producing an expected near-50% $|2\rangle$ population. It takes many surface code cycles before this injected leakage population decays sufficiently, with an exponential decay constant around 4.4 cycles. However, this decay is somewhat faster than the expected decay from T_1 of $|2\rangle$ alone. The insets show that the leakage population does not stay on the injected qubit, but is also transported to neighbouring qubits as the circuit progresses. At the small code distance being considered, this transport is enough to affect every qubit involved in the circuit.

Without any attempt to remove it, a single leakage event persists for many rounds and spreads a significant distance through the device, affecting many measurements and inducing error detections equivalent to many individual Pauli errors [94]. This high Pauli-equivalent-weight of leakage events makes them especially problematic for QEC.

The precise dynamics of leakage depends primarily on the details of the entangling gate. Here, we focus on the diabatic CZ gate used in the Sycamore architecture [54, 2, 4]. This gate involves biasing qubits to satisfy the resonance conditions indicated in Figure 4.2a, and tuning the interaction strength to achieve a rotation of 2π in $|11\rangle \leftrightarrow |20\rangle$. We maintain the convention that the higher energy qubit state is listed first in two-qubit

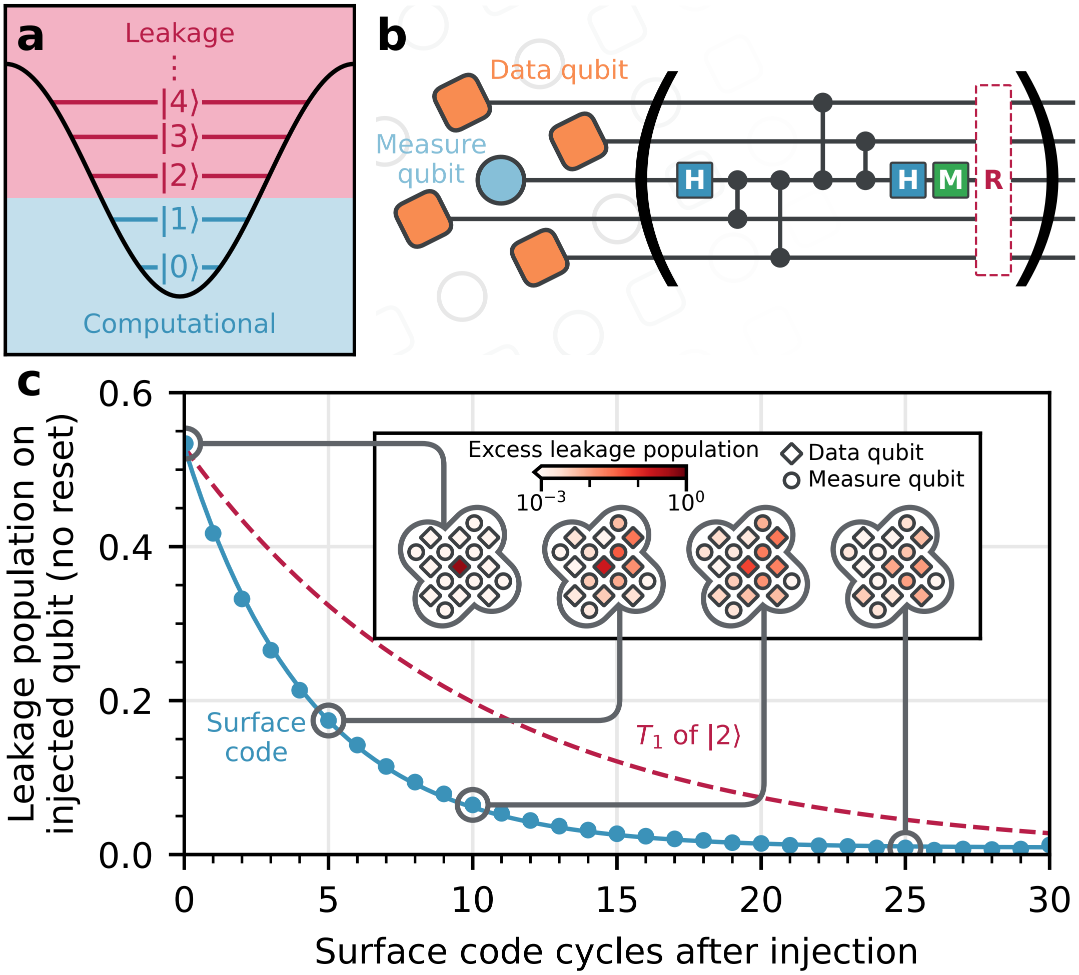


Figure 4.1: **Leakage in a structured QEC circuit.** a) The potential of a transmon qubit, illustrating the computational energy levels $|0\rangle$ and $|1\rangle$ and the leakage levels $|2\rangle$ and higher. b) The circuit for surface code QEC, showing a square grid of measure qubits (light blue circle) and data qubits (orange squares). The cycle consists of four layers of entangling gates, along with intervening single qubit rotations, followed by the measurement (**M**) and reset (**R**). The reset operation here is shown across all qubits; it may be implemented as single qubit operations on the measure qubit, or include entangling operations with various neighbouring data qubits. c) The time decay (main) and spatial spread (inset) leakage in a distance-3 surface code following the injecting of $|2\rangle$ on the central data qubit. Each cycle takes 1 μs. The expected decay from T_1 on the leaked qubit alone is indicated (dashed red), showing that the leakage leaves the qubit faster than decay alone would predict. The leakage populations on neighboring qubits rise above levels measured without injection on the central data qubit, indicating leakage from the injection qubit has spread to neighbouring qubits. Excess leakage population is defined as the subtraction of leakage population in the absence of injection from the leakage population in the presence of injection.

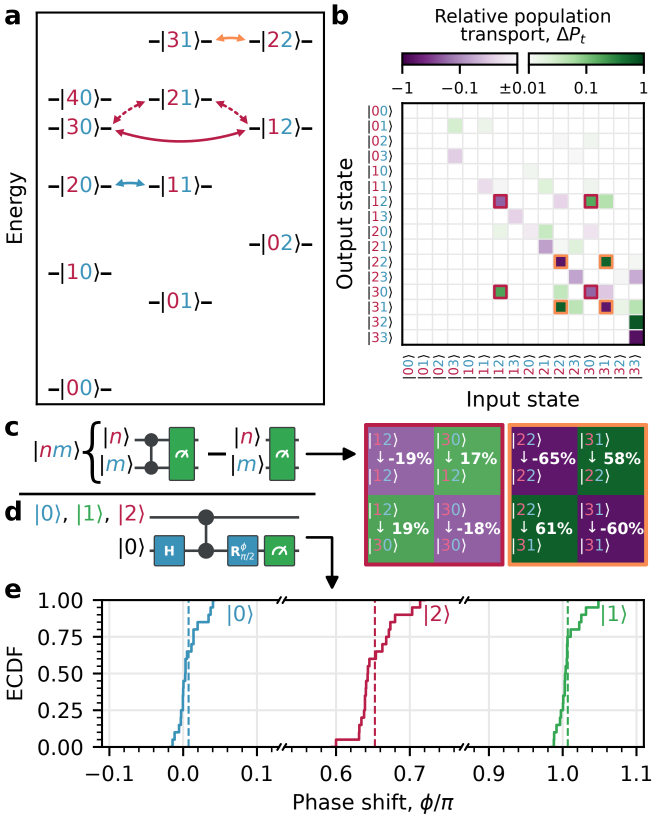


Figure 4.2: **Leakage transport and phase errors in CZ gates.** a) The Jaynes-Cummings ladder for a pair of qubits during a diabatic CZ gate, where the qubits are detuned by their common non-linearity η. We give combined qubit states with the higher frequency qubit state first. In addition to the intended resonance ($|20\rangle \leftrightarrow |11\rangle$), higher levels also satisfy a resonance condition, either directly ($|31\rangle \leftrightarrow |22\rangle$) or mediated by a 2-photon process ($|30\rangle \leftrightarrow |12\rangle$). b) The net change in state populations for the diabatic CZ gate, including the first two leakage levels. The rotation in $|20\rangle \leftrightarrow |11\rangle$ has been calibrated to 2π. Highlighted are the off-diagonal elements due to the couplings between higher levels, which are additionally shown below in detail. c) The two circuits used to measure the relative transport population shown in (b). We subtract the baseline experiment without a CZ gate from the experiment with a CZ gate. d) The circuit for the modified Ramsey experiment shown in (e) with an interleaved CZ to a neighbouring qubit at a higher frequency, followed by tomography on the lower frequency qubit. e) The measured phase shift ϕ during the Ramsey circuit on the primary qubit with the neighbouring qubit being prepared in $|0\rangle$, $|1\rangle$, or $|2\rangle$, shown in an ECDF over 20 measured pairs, with the mean value indicated by the dashed line. The CZ should produce an phase shift of $\phi = 0$ for an input $|0\rangle$, and a shift of $\phi = \pi$ for an input $|1\rangle$. When $|2\rangle$ is prepared on the neighbouring qubit, a spurious phase shift near $\phi \approx 0.65\pi$ radians is produced, equivalent to a significant phase error on the CZ gate.

kets. This resonance condition also aligns other resonances which are relevant for the leakage states. In particular, the $|30\rangle \leftrightarrow |12\rangle$ resonance enables a two-photon process which allows $|2\rangle$ on the lower-energy qubit to move to $|3\rangle$ on the higher-energy qubit. Similarly, the $|31\rangle \leftrightarrow |22\rangle$ resonance enables $|3\rangle$ on the higher-energy qubit to cause the lower-energy qubit to leak to $|2\rangle$, while the higher-energy qubit remains leaked in $|2\rangle$. These so-called *leakage transport* processes are what allow leakage to spread even in a single QEC cycle.

The amount of leakage transport a gate produces is not normally calibrated, and so depend on the chosen gate length and effective coupling between levels. Figure 4.2b shows how a normally calibrated CZ gate affects populations. Data for each individual experiment and further characterisation of the readout can be found in Section 1 of the Supplementary Information. In this device, we find around 20% of the population of $|30\rangle$ is transported to $|12\rangle$ and vice versa. The transport population is around 60% for $|31\rangle \leftrightarrow |22\rangle$. We can also see the first indications of expected higher resonances such as $|42\rangle \leftrightarrow |33\rangle$, discussed in more detail in Supplementary Information Section 1.

Even in the absence of leakage transport, we find that leakage induces additional errors in the CZ gate. When the higher energy qubit is in $|2\rangle$ and the lower energy qubit is in the computational basis, leakage transport is not possible, but a significant phase error is imparted on the non-leaked qubit. When a CZ gate is applied as in Figure 4.2d with the higher energy qubit in $|0\rangle$, we expect to see no relative phase on the lower energy qubit. With the high energy qubit prepared in $|1\rangle$, we expect to see a phase of π, indicating a well-calibrated CZ gate. Figure 4.2d shows the relative phase for 20 pairs of qubits. When computational states are prepared we see a tight grouping around the expected relative phase. , However, when a leakage state is prepared on the higher energy qubit, we see a relative phase near 0.65π rad. This represents a significant computational error on the non-leaked qubit, and is a significant source of errors to be detected and corrected as leakage spreads.

These results illuminate the danger of leakage; a single leakage event on any qubit will expose many CZ gates to a leaked input state before it finally decays, each of which has a significant probability to introduce new computational errors, move the leakage to another qubit, or induce additional leakage on previously non-leaked qubits. These effects are damaging enough that the must be included in simulations for them to match experimental performance [4]. Accordingly, we are motivated to remove leakage in the code circuit so as to suppress these effects.

4.4 Suppressing leakage populations during a code circuit

Having better understood the dangers of leakage in QEC circuits, we turn to removing it. An unconditional reset gate can remove both leakage and computational states and can be applied to the measure qubits at the end of each cycle [81, 82, 84, 1, 95]. However, our study of leakage transport motivates the need to remove it from the data qubits as well, where an unconditional reset is inappropriate. Leaving the computational state intact requires a more delicate *leakage removal* operation.

Three broad strategies for leakage removal have been proposed: swap-type strategies [94, 96], where the roles of measure and data qubits are exchanged at a regular interval by the use of additional operations; feedback-type [86, 85] where the leakage is identified classically from measurement patterns and feedback is applied to return the qubit to the computational subspace; and direct strategies [97] where an operation is used to remove leakage from a qubit without disturbing the computational states. In light of our findings on leakage transport, swap-type strategies become more difficult to justify; only half the qubits are reset in each round, and so leakage may still move between qubits and thereby spread through time. Similarly, the conditional nature of feedback-type approaches prevents them from fully solving the leakage problem – leakage states cause several errors before they are noticed and corrected. Hence, we pursue a direct removal approach.

In the following section, we present and compare three leakage removal techniques. First, *No Reset* forgoes any operations at the end of the cycle, representing the best case for a simple Pauli error model, but the worse case for leakage. Second, *MLR* applies multi-level reset (MLR) gates [1] on measure qubits immediately after measurement at the end of every cycle. This adds additional Pauli error to the cycle, primarily in the form of additional data qubit idling while the gate is performed, but has been previously shown to remove leakage population and improve code performance compared with the baseline *No Reset* case [1]. Finally, in *DQLR* we perform a multi-level reset on the measure qubits followed by data qubit leakage removal (DQLR), consisting of a two-qubit interaction to transport leakage from data to measure qubits and a second fast reset gate on measure qubits. Additional details on the DQLR process and constituent operations are included in Supplementary Information Section 2.

To compare the leakage dynamics for the three cases, we implement a distance-3 surface code on a Sycamore processor. We measure the evolution of leakage population as the surface code progresses by truncating the code in time and performing a measurement that can resolve $|2\rangle$ on all qubits [1]. In Figure 4.3a, we perform this truncation at the end of each surface code cycle. Using *No Reset*, we observe a gradual rise in leakage populations over all qubits, reaching nearly 5% average leakage population for data qubits and nearly 3% for measure qubits over 30 cycles. We note that even after 30 cycles, leakage populations have not stabilized and continue to grow. Using *MLR* reduces average measure qubit leakage populations to about 3×10^{-4}, but average data qubit populations

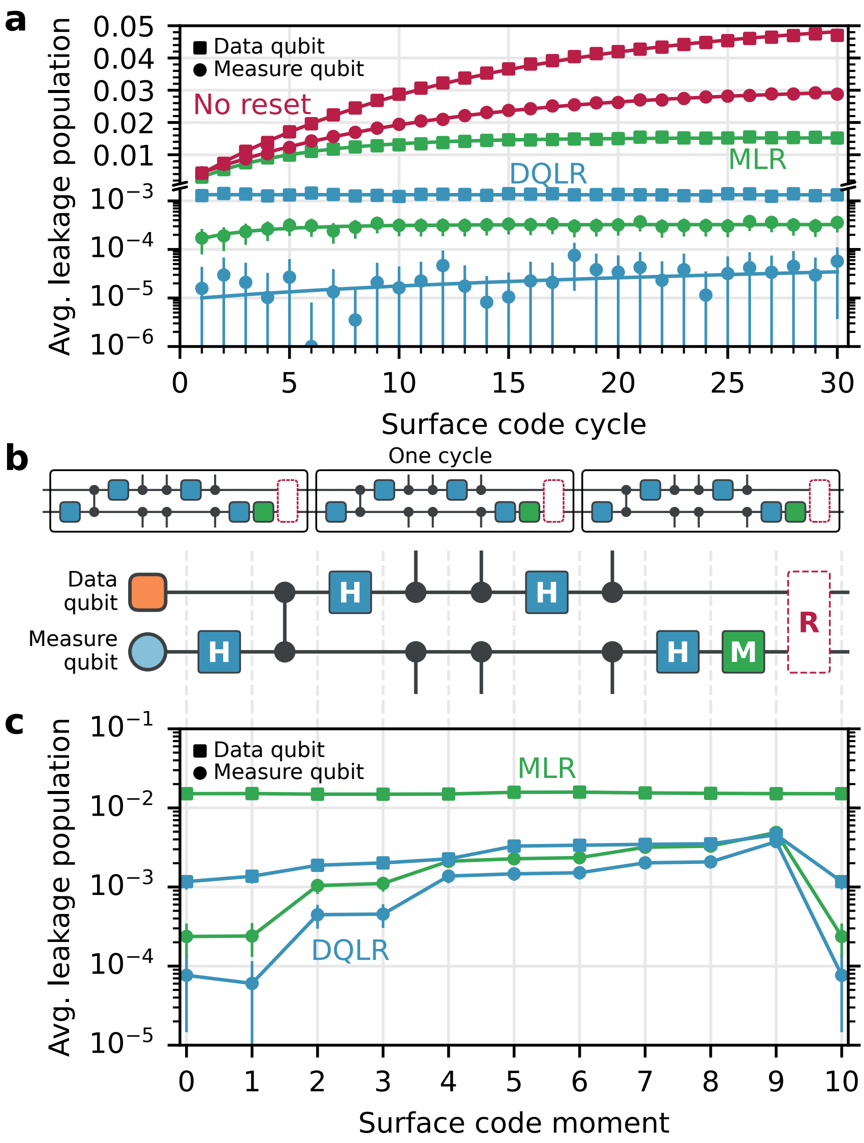


Figure 4.3: **Leakage population during code execution.** a) Average leakage populations for data qubits (diamonds) and measure qubits (circles) measured at the end of each surface code cycle with No Reset (red), MLR (green), and DQLR (blue). b) The surface code circuit shown for a neighbouring pair of measure and data qubits, additionally showing each moment in the cycle. c) Leakage populations after each moment in the cycle for MLR (green) and DQLR (blue) leakage removal strategies, averaged over data qubits (diamonds) and measure qubits (circles) and over cycles 25–30.

still rise to over 1.5%. Using *DQLR* suppresses average leakage populations to around 10^{-3} for data qubits and less than 10^{-4} for measure qubits. Most importantly, *DQLR* maintains these levels throughout the full 30 cycles.

We can use the same technique to study the dynamics of leakage within a surface code cycle, by truncating the circuit midway through a cycle. Figure 4.3c shows the leakage population measured after each gate in the cycle, averaged over cycles 25–30 where the leakage populations have stabilized. We neglect the *No Reset* case here, as leakage populations do not stabilize. With *MLR*, the average leakage population on the data qubits saturates to a stable value around 1.5%, consistent with Figure 4.3a. However, the average measure qubit leakage population starts each cycle at a very low value near 2×10^{-4}, grows over the course of the cycle as operations produce leakage, and is then reduced back to its initial low value by the reset procedure. This lets us estimate that the operations produce around 0.4% leakage in each cycle. With *DQLR*, we see that leakage populations for both measure and data qubits grow over the course of the cycle, and are removed by the reset procedure. The data qubits start each cycle with around 0.1% leakage population, increasing to around 0.5% immediately following measurement, before it is removed. The measure qubits attain even lower leakage populations compared to *MLR*.

These results demonstrate that our DQLR procedure successfully suppresses steady state leakage populations to previously unachievable levels and stabilizes those levels over the course of a long code circuit. The removal strategy also contains the leakage dynamics to a single cycle. These single-cycle dynamics, especially the ability for leakage to spread and induce hook errors, should be the subject of further study.

4.5 Effect on QEC logical performance

Having achieved the low leakage populations in both data qubits and measure qubits with our DQLR procedure, we turn to evaluating logical performance. We consider two codes providing complementary information: a distance-21 bit-flip code and a distance-3 surface code. Our physical qubit error rates place the surface code close to threshold, whereas the bit-flip code is well below threshold [2, 4]. The vastly lower logical error rates for the bit-flip code give us finer resolution on the effect of leakage within the code. In contrast, the surface code is a more challenging circuit for calibration and operation, and is sensitive to both bit-flip and phase-flip errors, providing an environment where more potentially adverse effects from reset can be detected and measured.

Figure 4.4a shows the logical error probability of a distance-21 bit-flip code carried out to 60 cycles while introducing both leakage and Pauli errors. We inject leakage population P_L into all qubits by applying a $|1\rangle \leftrightarrow |2\rangle$ rotation on each qubit immediately after the first Hadamard gate layer (Figure 4.4b, left), where the rotation angle θ_L is

$$\theta_L = 2\sin^{-1}\left(\sqrt{2P_L}\right).$$

The injection gate itself takes the same time as our single qubit rotations (25 ns). We compare P_L to injected Pauli error "population" P_P, which is produced by X and Z rotations on the data and measure qubits respectively, taking advantage of the classical nature of the bit-flip code. The Pauli error rotation angle θ_P is

$$\theta_P = 2\sin^{-1}\left(\sqrt{P_P}\right),$$

where the missing factor of 2 relative to the definition of leakage population accounts for Pauli rotations always affecting the qubit state in the computational basis, whereas

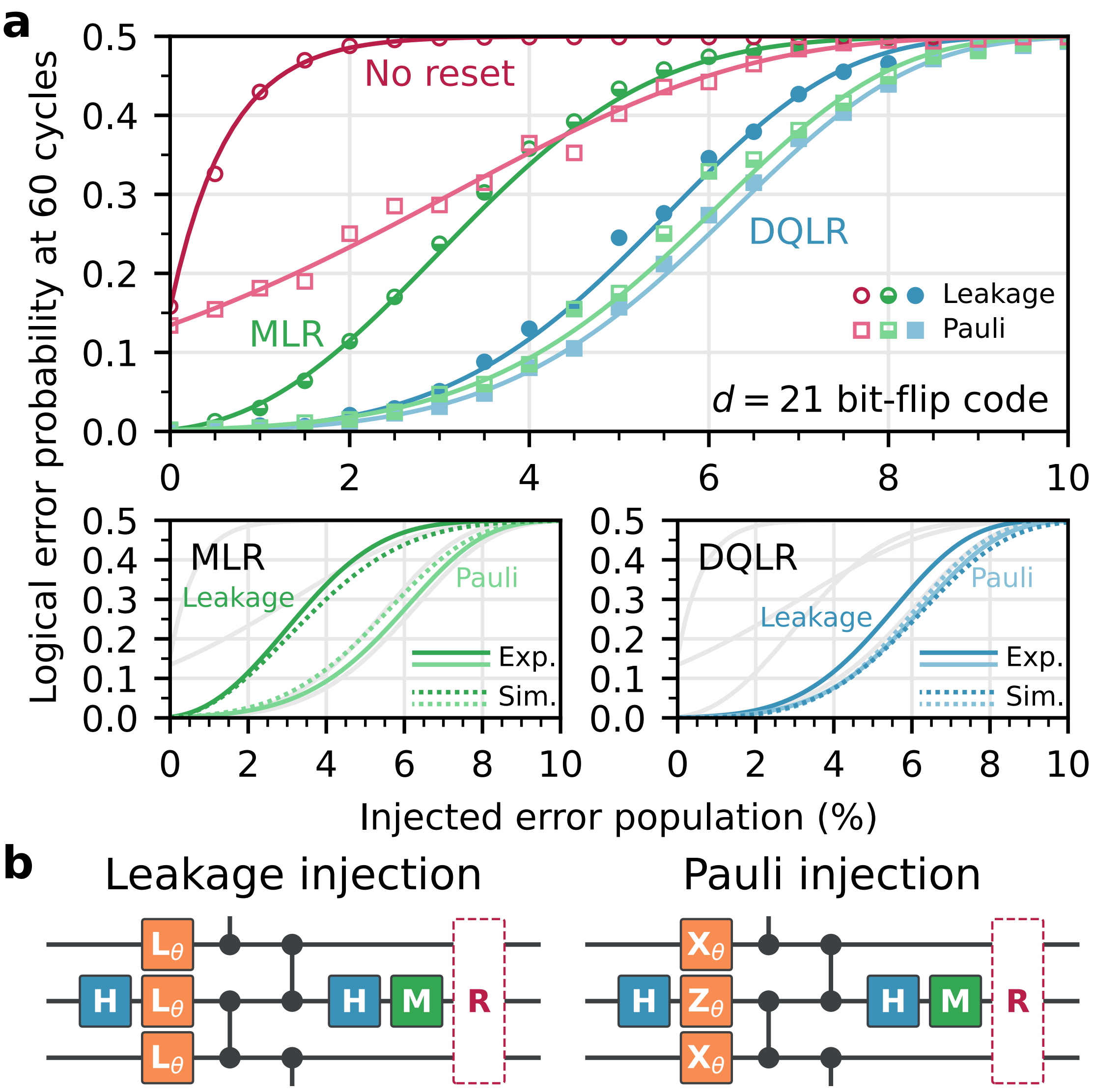


Figure 4.4: **Bit-flip code logical performance and dependence on injected errors.** a) Logical error probability for a distance-21 bit-flip code run for 60 cycles, showing both injected leakage (dark circles) and injected Pauli errors (light squares), with offset power law fits. (Below) Highlights of fits to experiment (solid) and numerical simulations (dashed) for the MLR and DQLR cases. For MLR, we see that injecting leakage produces significantly more logical error than injecting the equivalent amount of Pauli errors. With DQLR, we see that the two types of error injection produce essentially similar amounts of logical error. b) Circuits for the bit-flip code, showing the injection locations for both leakage and Pauli errors.

leakage injection only applies to qubit population in $|1\rangle$. We fit the experimental data to an offset power law as a guide, as detailed in Section XXX of the Supplementary Information.

With *No Reset*, even small amounts of injected leakage population <1% cause the logical error probability to rise above 40%. This is in contrast with correctable Pauli errors, which can be introduced to around 5% population before similar logical error probabilities are encountered. With *MLR*, the logical error probability is drastically lowered without injection, consistent with prior measurements in bit-flip codes [1]. Still, the logical error probability rises much more rapidly when injecting leakage compared to injecting Pauli errors. We attribute this to unmitigated leakage accumulation on the data qubits, which leads to high Pauli-equivalent-weight errors and ultimately logical errors. When we prevent this leakage buildup with *DQLR*, we observe a much smaller difference between the code's response to injected leakage compared to injected Pauli errors. This is strong evidence that the DQLR operation has successfully reduced the number of Pauli-equivalent errors a leakage event induces to close to 1. In this situation, leakage has around the same influence on logical performance as a Pauli error of the same magnitude, and has been prevented from effectively spreading and inducing correlated errors.

We also note the good agreement between data and numerical simulation for injected leakage and Pauli errors, quantifying our understanding of the effects of leakage in the code with both *MLR* and *DQLR* strategies. In both cases, we note that we slightly underestimate the logical error induced by injected leakage, illustrating the difficulties of fully capturing the effect of correlated errors even with *DQLR* preventing substantial spread across cycles, and emphasising the importance of future work on leakage dynamics inside a single cycle. Nonetheless, the close correspondence of the Pauli simulation to the injected leakage experimental data for *DQLR* helps justify future Pauli simulations as useful estimates of final code performance when leakage is removed each cycle.

Figure 4.5a shows the average detection probabilities corresponding to the weight-4 stabilizers in the distance-3 surface code. Detection probabilities are the fraction of the total number of experiments where an error was detected on a given stabilizer. With *No Reset*, the buildup of leakage population produces more errors as the code progresses, creating a rising pattern of detection probability. With *MLR*, a large portion of this rise is mitigated, but the detection probability still rises by 2.5% over the course of the first 15 cycles. With *DQLR* the detection probability immediately stabilizes to around 18% and remains static through the code. We attribute this to the recurrent removal of leakage on all qubits preventing growth in leakage populations and resulting correlated errors over time. This resolves a key concern in state of the art QEC work [36, 37, 4] where detection probabilities were found to rise even with partial leakage removal or post-selection. These results confirm the relationship between rising detection probability and rising leakage populations and demonstrate the resolution of this effect.

In Figure 4.5b, we evaluate the three leakage removal strategies by measuring the logical error probability of a distance-3 surface code after 15 cycles. At 0% injected leakage the circuit corresponds to the standard code circuit with an additional idling where the injection is otherwise inserted. Over the range of injected leakage population values, *No Reset* exhibits the worst logical performance, followed by *MLR*, with *DQLR* having the lowest logical error probability. This confirms that the additional errors caused by the DQLR operation are more than offset by the removal of leakage and suppression of correlated errors. Further, *No Reset* and *MLR* degrade in logical performance faster with more injected leakage when compared to *DQLR*.

In order to study surface code performance in a regime further below threshold, we turn to numerical simulations of distance-5 and distance-7 surface codes. In Figure 4.5c, we investigate the exponential error suppression factor between a distance-5 and distance-

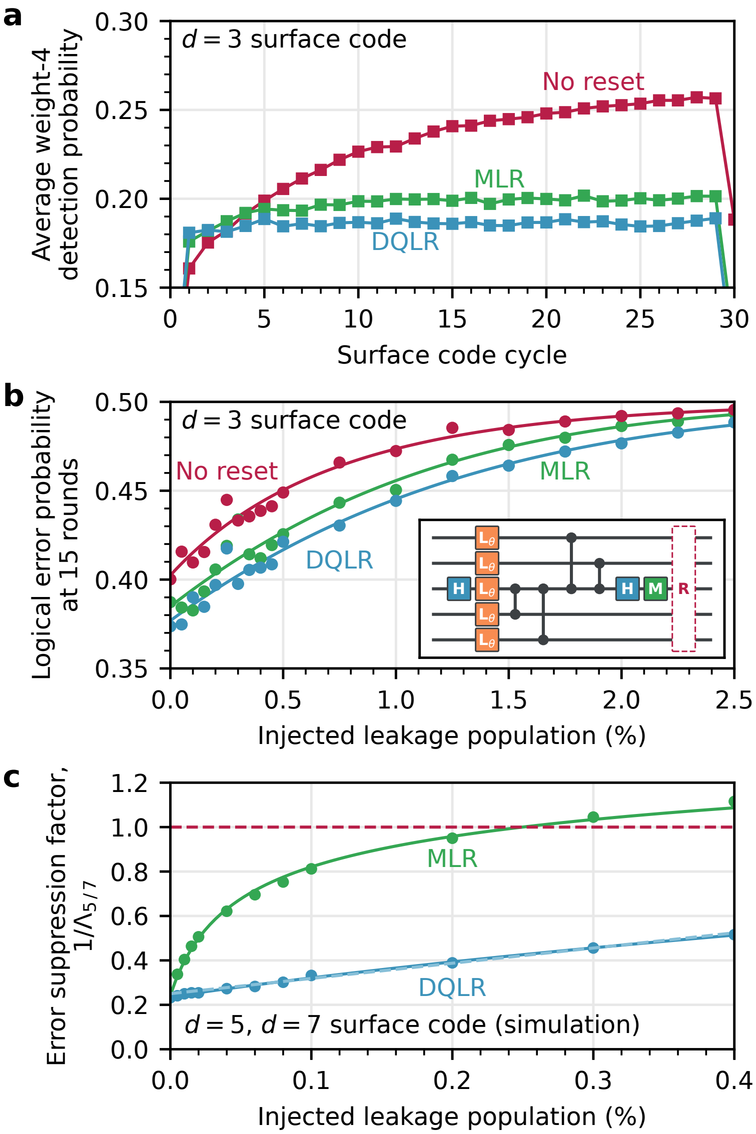


Figure 4.5: **Surface code logical performance and dependence on injected errors.** a) The detection probability averaged for the weight-4 stabilizers in a distance-3 surface code. Without reset, the detection probability grows as the code progresses. With MLR, the detection probability rises more slowly and saturates by around 10 cycles. With DQLR, the detection probability is flat with respect to cycles. b) The probability of logical error for a distance-3 surface code run for 15 cycles. The inset shows that the circuit has an included layer where leakage is injected by performing a $|1\rangle \leftrightarrow |2\rangle$ rotation. c) Dependence of projected exponential error suppression factor between a distance-5 and distance-7 surface code $1/\Lambda_{5/7}$ under injected leakage population. With MLR, small injected leakage populations cause exponential error suppression performance to degrade. With DQLR, exponential error suppression is maintained for a larger range of injected leakage population.

7 surface code, $1/\Lambda_{5/7}$, for a hypothetical device with lower component errors than what is currently realizable (see Supplementary Information for details). In particular, we set intrinsic leakage rates to zero and study only injected leakage. With no leakage in the system, the error suppression factor is expected to be around 0.25, independent of leakage removal strategy. However, when injecting up to 4×10^{-3} leakage populations per round (comparable to intrinsic leakage rates in current devices), $1/\Lambda_{5/7}$ rapidly rises for *MLR*, indicating a breakdown in exponential error suppression. In contrast, with *DQLR*, leakage degrades $1/\Lambda_{5/7}$ much more slowly and with a near-linear dependence on injected leakage population, characteristic of an uncorrelated error source[2, 4]. With this ability to maintain effective error suppression in the presence of leakage, *DQLR* successfully mitigates the dangers of correlated leakage-induced errors.

4.6 Summary and Outlook

We have demonstrated the effective removal of leakage from all qubits involved in a surface code QEC circuit. Moreover, we have shown that when leakage is removed, correlated leakage-induced errors are suppressed, while the logical performance of the code improves outright and stabilizes in time. We confirm the conjecture that growth in logical errors is attributable to leakage, and we do not uncover other major sources of logical error that grow as the code continues in time.

We demonstrate a leakage removal strategy that is compatible with scaling QEC in both number of qubits and number of cycles. With these findings, we positively resolve the longstanding concern that qubits with weak nonlinearity cannot successfully implement QEC at long times due to correlated leakage-induced errors. As such, we confirm that large arrays of transmon qubits are a viable and promising architecture for QEC at scale.

References

[1] Matt McEwen et al. “Removing leakage-induced correlated errors in superconducting quantum error correction”. In: *Nature Communications* 12.1 (Dec. 2021), p. 1761. DOI: `10.1038/s41467-021-21982-y`.

[2] Google Quantum AI et al. “Exponential suppression of bit or phase errors with cyclic error correction”. In: *Nature* 595.7867 (July 15, 2021). _eprint: 2102.06132, pp. 383–387. DOI: `https://doi.org/10.1038/s41586-021-03588-y`.

[4] Google Quantum AI et al. “Suppressing quantum errors by scaling a surface code logical qubit”. In: (2022). DOI: `10.48550/ARXIV.2207.06431`.

[6] Kevin C. Miao and Matt McEwen. “Complete Leakage Removal in the Surface Code on Superconducting Qubits”. In: *Submitted Nature Physics* (2022).

[25] S. B. Bravyi and A. Yu. Kitaev. “Quantum codes on a lattice with boundary”. In: (1998). Publisher: arXiv Version Number: 1. DOI: `10.48550/ARXIV.QUANT-PH/9811052`.

[26] Austin G. Fowler et al. “Surface codes: Towards practical large-scale quantum computation”. In: *Physical Review A* 86.3 (Sept. 18, 2012). Publisher: American Physical Society, p. 032324. DOI: `10.1103/physreva.86.032324`.

[36] Sebastian Krinner et al. “Realizing repeated quantum error correction in a distance-three surface code”. In: *Nature* 605.7911 (May 26, 2022). Publisher: Springer Science and Business Media LLC, pp. 669–674. DOI: `10.1038/s41586-022-04566-8`.

[37] Neereja Sundaresan et al. “Matching and maximum likelihood decoding of a multi-round subsystem quantum error correction experiment”. In: (2022). DOI: `10.48550/ARXIV.2203.07205`.

[43] Austin G. Fowler and John M. Martinis. “Quantifying the effects of local many-qubit errors and nonlocal two-qubit errors on the surface code”. In: *Physical Review A* 89.3 (Mar. 12, 2014). Publisher: American Physical Society, p. 032316. DOI: `10.1103/PhysRevA.89.032316`.

[49] Jens Koch et al. “Charge-insensitive qubit design derived from the Cooper pair box”. In: *Physical Review A* 76.4 (Oct. 12, 2007). Publisher: American Physical Society, p. 042319. DOI: `10.1103/PhysRevA.76.042319`.

[51] J. Kelly et al. “State preservation by repetitive error detection in a superconducting quantum circuit”. In: *Nature* 519.7541 (Mar. 5, 2015). Publisher: Nature Publishing Group, pp. 66–69. DOI: `10.1038/nature14270`.

[52] Charles Neill. “A path towards quantum supremacy with superconducting qubits”. PhD thesis. University of California, Santa Barbara, 2017.

[53] Fei Yan et al. “Tunable Coupling Scheme for Implementing High-Fidelity Two-Qubit Gates”. In: *Physical Review Applied* 10.5 (Nov. 28, 2018). Publisher: American Physical Society, p. 054062. DOI: `10.1103/PhysRevApplied.10.054062`.

[54] B. Foxen et al. “Demonstrating a Continuous Set of Two-qubit Gates for Near-term Quantum Algorithms”. In: *Physical Review Letters* 125.12 (Sept. 15, 2020). _eprint: 2001.08343, p. 120504. DOI: `https://doi.org/10.1103/PhysRevLett.125.120504`.

[65] Barbara M. Terhal. “Quantum error correction for quantum memories”. In: *Reviews of Modern Physics* 87.2 (Apr. 7, 2015). Publisher: American Physical Society, pp. 307–346. DOI: `10.1103/RevModPhys.87.307`.

[66] Christian Kraglund Andersen et al. “Repeated quantum error detection in a surface code”. In: *Nature Physics* 16.8 (Aug. 2020). Publisher: Springer Science and Business Media LLC, pp. 875–880. DOI: `10.1038/s41567-020-0920-y`.

[67] Jerry M. Chow et al. “Implementing a strand of a scalable fault-tolerant quantum computing fabric”. In: *Nature Communications* 5.1 (Sept. 2014). Publisher: Nature Publishing Group, p. 4015. DOI: `https://doi.org/10.1038/ncomms5015`.

[68] A. D. Córcoles et al. “Detecting arbitrary quantum errors via stabilizer measurements on a sublattice of the surface code”. In: *Nat. Commun.* 6 (2014). Publisher: Nature Publishing Group, p. 6979. DOI: `https://doi.org/10.48550/arXiv.1410.6419`.

[69] Maika Takita et al. “Demonstration of Weight-Four Parity Measurements in the Surface Code Architecture”. In: *Physical Review Letters* 117.21 (Nov. 18, 2016). Publisher: American Physical Society, p. 210505. DOI: `10.1103/PhysRevLett.117.210505`.

[71] D. Ristè et al. “Detecting bit-flip errors in a logical qubit using stabilizer measurements”. In: *Nature Communications* 6.1 (Nov. 2015). Publisher: Nature Publishing Group, p. 6983. DOI: `10.1038/ncomms7983`.

[72] M. D. Reed et al. “Realization of three-qubit quantum error correction with superconducting circuits”. In: *Nature* 482.7385 (Feb. 2012). Publisher: Springer Science and Business Media LLC, pp. 382–385. DOI: `10.1038/nature10786`.

[73] Christian Kraglund Andersen et al. “Entanglement stabilization using ancilla-based parity detection and real-time feedback in superconducting circuits”. In: *npj Quantum Information* 5.1 (Dec. 2019). Publisher: Springer Science and Business Media LLC, p. 69. DOI: `10.1038/s41534-019-0185-4`.

[74] F. Motzoi et al. “Simple Pulses for Elimination of Leakage in Weakly Nonlinear Qubits”. In: *Physical Review Letters* 103.11 (Sept. 8, 2009). Publisher: American Physical Society, p. 110501. DOI: `10.1103/PhysRevLett.103.110501`.

[75] Zijun Chen et al. “Measuring and Suppressing Quantum State Leakage in a Superconducting Qubit”. In: *Physical Review Letters* 116.2 (Jan. 13, 2016). Publisher: American Physical Society, p. 020501. DOI: `10.1103/PhysRevLett.116.020501`.

[76] R. Barends et al. “Superconducting quantum circuits at the surface code threshold for fault tolerance”. In: *Nature* 508.7497 (Apr. 2014). Publisher: Nature Publishing Group, pp. 500–503. DOI: `https://doi.org/10.1038/nature13171`.

[77] M. A. Rol et al. “Fast, High-Fidelity Conditional-Phase Gate Exploiting Leakage Interference in Weakly Anharmonic Superconducting Qubits”. In: *Physical Review Letters* 123.12 (Sept. 18, 2019). Publisher: American Physical Society, p. 120502. DOI: `10.1103/PhysRevLett.123.120502`.

[78] V. Negîrneac et al. “High-Fidelity Controlled- Z Gate with Maximal Intermediate Leakage Operating at the Speed Limit in a Superconducting Quantum Processor”. In: *Physical Review Letters* 126.22 (June 4, 2021). _eprint: 2008.07411, p. 220502. DOI: `https://doi.org/10.1103/PhysRevLett.126.220502`.

[80] Daniel Sank et al. “Measurement-Induced State Transitions in a Superconducting Qubit: Beyond the Rotating Wave Approximation”. In: *Physical Review Letters* 117.19 (Nov. 4, 2016). Publisher: American Physical Society, p. 190503. DOI: `10.1103/physrevlett.117.190503`.

[81] M. D. Reed et al. “Fast reset and suppressing spontaneous emission of a superconducting qubit”. In: *Applied Physics Letters* 96.20 (May 17, 2010). Publisher: AIP Publishing, p. 203110. DOI: `10.1063/1.3435463`.

[82] K. Geerlings et al. “Demonstrating a Driven Reset Protocol for a Superconducting Qubit”. In: *Physical Review Letters* 110.12 (Mar. 20, 2013). Publisher: American Physical Society, p. 120501. DOI: `10.1103/PhysRevLett.110.120501`.

[84] P. Magnard et al. “Fast and Unconditional All-Microwave Reset of a Superconducting Qubit”. In: *Physical Review Letters* 121.6 (Aug. 7, 2018). Publisher: American Physical Society, p. 060502. DOI: `10.1103/PhysRevLett.121.060502`.

[85] C. C. Bultink et al. “Protecting quantum entanglement from leakage and qubit errors via repetitive parity measurements”. In: *Science Advances* 6.12 (Mar. 20, 2020). Publisher: American Association for the Advancement of Science (AAAS), eaay3050. DOI: `10.1126/sciadv.aay3050`.

[86] Boris Mihailov Varbanov et al. “Leakage detection for a transmon-based surface code”. In: *npj Quantum Information* 6.1 (Dec. 2020). Publisher: Springer Science and Business Media LLC, p. 102. DOI: `10.1038/s41534-020-00330-w`.

[89] Morten Kjaergaard et al. “Superconducting Qubits: Current State of Play”. In: *Annual Review of Condensed Matter Physics* 11.1 (Mar. 10, 2020), pp. 369–395. DOI: `10.1146/annurev-conmatphys-031119-050605`.

[93] Ross Shillito et al. “Dynamics of Transmon Ionization”. In: (2022). Publisher: arXiv Version Number: 1. DOI: `10.48550/ARXIV.2203.11235`.

[94] Austin G. Fowler. “Coping with qubit leakage in topological codes”. In: *Physical Review A* 88.4 (Oct. 8, 2013). Publisher: American Physical Society, p. 042308. DOI: `10.1103/PhysRevA.88.042308`.

[95] Yu Zhou et al. “Rapid and unconditional parametric reset protocol for tunable superconducting qubits”. In: *Nature Communications* 12.1 (Dec. 2021). Publisher:

Springer Science and Business Media LLC, p. 5924. DOI: `10.1038/s41467-021-26205-y`.

[96] Natalie C. Brown and Kenneth R. Brown. "Leakage mitigation for quantum error correction using a mixed qubit scheme". In: *Physical Review A* 100.3 (Sept. 18, 2019). Publisher: American Physical Society, p. 032325. DOI: `10.1103/PhysRevA.100.032325`.

[97] F. Battistel, B.M. Varbanov, and B.M. Terhal. "Hardware-Efficient Leakage-Reduction Scheme for Quantum Error Correction with Superconducting Transmon Qubits". In: *PRX Quantum* 2.3 (July 26, 2021), p. 030314. DOI: `10.1103/PRXQuantum.2.030314`.

Chapter 5

High-Energy Impacts

This chapter reproduces the work published as *Resolving catastrophic error bursts from cosmic rays in large arrays of superconducting qubits* [3]. The supplementary information for this publication is reproduced in Appendix C.

5.1 Abstract

Scalable quantum computing can become a reality with error correction, provided coherent qubits can be constructed in large arrays [26, 65]. The key premise is that physical errors can remain both small and sufficiently uncorrelated as devices scale, so that logical error rates can be exponentially suppressed. However, energetic impacts from cosmic rays and latent radioactivity violate both of these assumptions. An impinging particle ionizes the substrate, radiating high energy phonons that induce a burst of quasiparticles, destroying qubit coherence throughout the device. High-energy radiation has been identified as a source of error in pilot superconducting quantum devices [98, 47, 99], but lacking a measurement technique able to resolve a single event in detail, the effect on large-scale algorithms and error correction in particular remains an open

question. Elucidating the physics involved requires operating large numbers of qubits at the same rapid timescales as in error correction, exposing the event's evolution in time and spread in space. Here, we directly observe highly correlated error bursts produced by high-energy rays impacting a large-scale quantum processor. We introduce a rapid space and time-multiplexed measurement method and identify large bursts of quasiparticles that simultaneously and severely limit the energy coherence of all qubits, causing chip-wide failure. We track the events from their initial localised impact to high error rates across the chip. Our results provide direct insights into the scale and dynamics of these damaging error bursts in large-scale devices, and highlight the necessity of mitigation to enable quantum computing to scale.

5.2 Introduction

Quantum states are inherently fragile. Superconducting qubits can achieve significant coherence only when cooled to milliKelvin temperatures and protected from their environment through extensive engineering. Maintaining stable coherence becomes an especially important task in quantum error correction (QEC), which is only possible if all qubits display low levels of physical error throughout lengthy algorithms. A correlated error event is particularly problematic, as affecting many qubits simultaneously is a recipe for logical faults [43].

Energetic radiation is likely to produce such errors, as devices are bathed in a constant flux of high-energy rays [100]. Cosmic rays constantly impact the upper atmosphere, producing high-energy muons which can strike the chip directly or scatter in surrounding material to produce secondary rays. Additionally, gamma rays are commonly emitted from trace radioactive impurities, both directly and from Compton scattering of beta emissions.

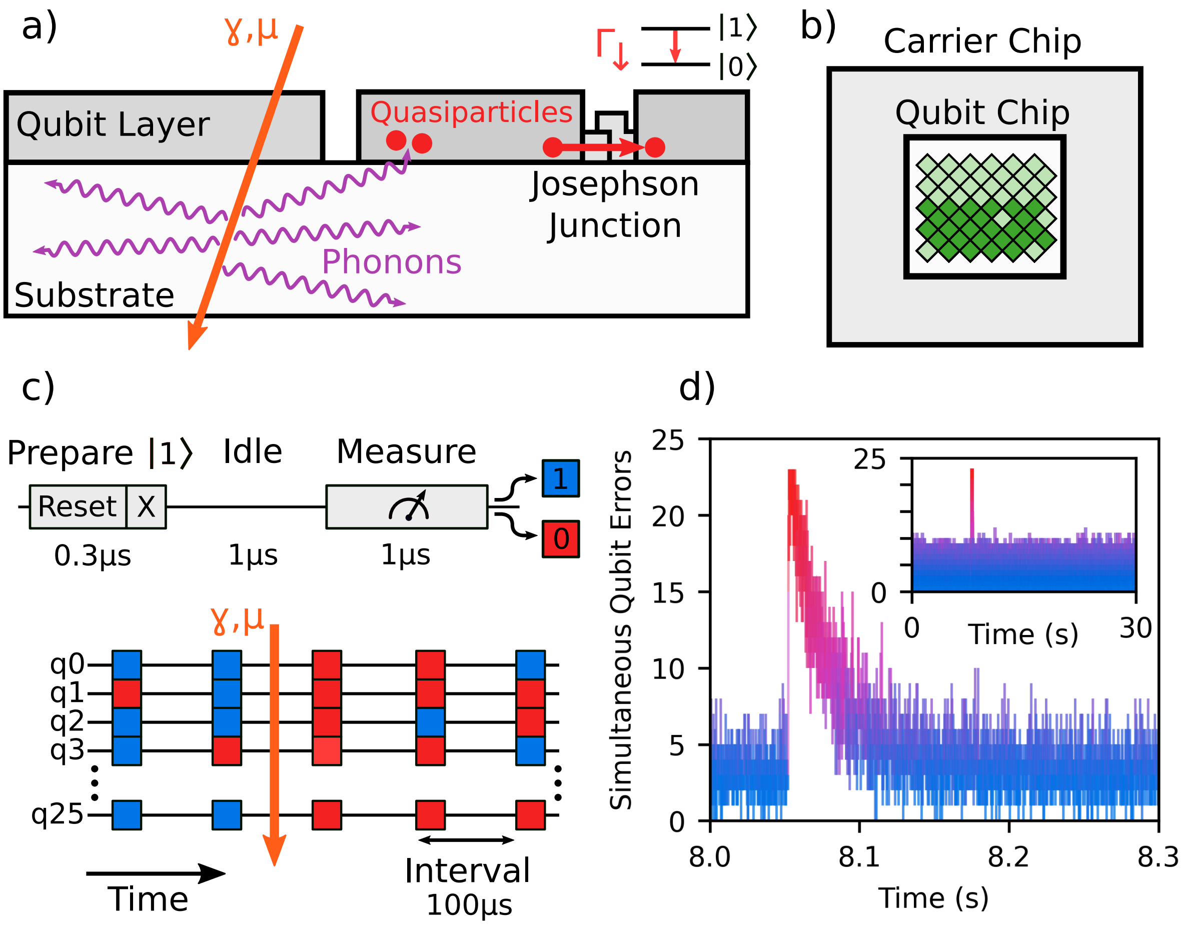


Figure 5.1: **Rapid repetitive correlated sampling.** (a) High energy radiation impinging on the device deposits energy, which spreads in the form of high energy phonons. In superconducting structures, this energy creates quasiparticles, which cause qubit energy decay as they tunnel across the Josephson junction. (b) We use a 26 qubit subset (dark green) of a Google Sycamore processor. The qubit chip is attached to a larger carrier chip using indium bumpbonds. (c) The Rapid Repetitive Correlated Sampling (RReCS) experiment consists of repeated cycles of preparation, idling and measurement. The idling time of 1 μs sets the sensitivity to decay errors. The interval between the start of each cycle is 100 μs. (d) A timeslice of a 30 second long dataset, showing a correlated error event. The number of simultaneous qubit decay errors jumps from baseline ~ 4 up to ~ 24, effectively saturating the chip. The number of errors returns to baseline with an exponential time constant of ~ 25 ms. We do not find any correlated error events when preparing the qubits in $|0\rangle$.

As illustrated in Fig. 5.1, incoming rays interact with matter along their path, ionizing the substrate and producing energetic phonons with long lifetimes. As they spread through the chip, these phonons can break paired electrons in superconducting structures and produce large densities of excess quasiparticles. A single quasiparticle may cause state decay as it tunnels across the junctions at the heart of superconducting qubits [46, 101, 102, 103, 104, 45]. These rays can deposit energies in the 100 keV to 1 MeV range [105, 99], dwarfing the energy scale necessary to induce a quasiparticle ($\sim$ 150 μeV) that can cause a decay event. This energy is transported over long distances by pair-breaking phonons traveling ballistically through the insulating substrate, and also diffusion in metallic structures as energy is freely exchanged between quasiparticles and phonons, discussed in detail in the Supplementary Information. The avalanche of quasiparticles produced by such an impact that spread throughout the chip and cause a severe suppression of coherence [106].

High-energy radiation has been identified as a source of quasiparticles in transmon qubits [47], and single events have been reported in small numbers of specialized devices such as resonators [98] as well as qubits that are charge-sensitive [99]. However, observing these discrete events in detail presents a challenge in metrology. To understand the effect on QEC, one would need to observe how the events unfold in a device directly to characterize nature of the induced correlations. This includes measuring the fast dynamics of the initial impact, the spread of errors through the qubit grid, and the eventual recovery to equilibrium. Therefore, a large array of qubits operated at rapid cycle times is required to illuminate the individual events and diagnose their impact on practical error correction.

Here, we directly measure the occurrence of high-energy events in a large-scale working device in the form of a Google Sycamore processor and provide insights into the microscopic dynamics of these events. We show that high-energy events produce discrete bursts of errors that affect an entire qubit patch on the processor, effectively lasting for thousands of error correction cycles. Using fine time-resolved measurements, we show that events are initially localized but spread over the chip, providing strong evidence for a high-energy impact. Finally, we introduce a method to monitor the energy coherence time T_1 during an event and find it to be severely suppressed across all qubits, a clear signature of quasiparticle poisoning throughout the chip.

5.3 Rapid Repetitive Correlated Sampling

In order to measure these events in detail, we must rapidly identify correlated errors in large qubit arrays. We use a subset of a Google Sycamore Processor [35], as indicated in Fig. 5.1b. The qubit chip consists of an array of flux-tunable superconducting transmon qubits [49, 89] with tunable couplers [52, 53, 54]. Qubit operating frequencies are chosen algorithmically [61] between 6 and 7 GHz, with resulting T_1 at operating frequency around 15 μs. Each qubit features a readout resonator to allow dispersive readout. We turn off the coupling between neighbouring pairs of qubits. We operated only a subset of the device and chose $N_Q = 26$ qubits which could be operated in parallel with high fidelity. Each qubit lies around 1 mm from its nearest neighbours on a qubit chip measuring 10 mm x 10 mm, which is attached to a larger carrier chip measuring 20 mm x 24 mm using indium bumpbonds [57].

We introduce a method that rapidly and simultaneously measures qubit states to identify correlated errors, which we call Rapid Repetitive Correlated Sampling (RReCS). As indicated in Fig. 5.1c, all qubits are prepared in the $|1\rangle$ state, allowed to idle for a short sampling time (1 μs), and then simultaneously measured. This cycle is repeated at rapid regular intervals (100 μs) for extended periods of time, with any measurements where the qubit state has decayed to $|0\rangle$ recorded as an error. Finite T_1 and readout fidelities will produce errors that are independent between qubits, creating a low background error rate. With this technique, the quantum processor becomes a time-resolved detector for events that affect large numbers of qubits.

A time slice from a RReCS experiment is shown in Fig. 5.1d. It features a distinct peak where the total number of errors jumps from a baseline of $\sim$4 simultaneous errors up to $\sim$24 errors. This event has effectively saturated the qubit patch, with all qubits experiencing a high probability of reporting an error, indicating total failure of the coherence on the chip. The peak features an exponential decay back to the baseline error rate with a time constant around 25 ms, which is much larger than the typical QEC round time of 1 μs [1, 2]. The presence of such a long time period of elevated error rates would be unacceptable for any attempt at logical state preservation using QEC.

One signature of quasiparticle poisoning is an asymmetry between decay and excitation errors. Quasiparticles rapidly scatter and cool to energies near the superconducting gap Δ, where they become unable to excite the qubit state from $|0\rangle \rightarrow |1\rangle$, which requires energy $\Delta + E_{01}$, where E_{01} is the energy difference between the $|0\rangle$ and $|1\rangle$ states. However, quasiparticles maintain the ability to absorb the qubit energy and cause a decay error $|1\rangle \rightarrow |0\rangle$. This asymmetry is distinct from photon-assisted tunneling which produces nearly symmetric errors [107]. As a test, we run the RReCS experiments for excitation errors, initialising $|0\rangle$ states and recording excitation to $|1\rangle$ states as an error. We do not find any correlated error peaks, indicating that events are produced by a highly asymmetric decay error mechanism, which is compatible with quasiparticle poisoning across the chip. Further detail on these experiments is included in the Supplementary Information.

5.4 Timing of Events and Independent Background Error

To understand the arrival rate and uniformity of impact events, we now deploy RReCS experiments for long time periods to gather large numbers of events. We acquire 100 back-to-back datasets of 60 seconds each, and apply a matched filter to isolate events over the background independent error rate. Details on this filtering are included in the Supplementary Information. Four sequential datasets are shown in Fig. 5.2, selected to include one dataset without any events present. In Fig. 5.2a, the raw time series data illustrates the background error rate, but the filtered data displays low noise and clearly identifies events even at scales lower than the background noise level. Fig. 5.2b shows corresponding histograms over the number of simultaneous errors, where the black lines indicate the expected background distribution of independent errors.

We include a simple independent error model, where we assume perfect initialization, followed by population decay with an independently measured T_1 time over the 1 μs sampling time, and finally account for separately measured finite readout fidelities. In the absence of events, we note a strong correspondence of the background error distribution to this simple model, as illustrated in Fig. 5.2b (II). In the presence of events, we note a distinct excess of high numbers of simultaneous errors, well above what is reasonable for uncorrelated error sources. This indicates that the baseline performance of the experiment is well understood and that the peaks represent anomalous correlated error events.

Using our matched filter, we extract 415 events from these datasets, which we then fit individually to extract a peak height and exponential decay timescale. Details on this analysis and the distributions of extracted parameters are included in the Supplementary Information. We find that the decay timescale is tightly grouped in the 25 to 30 ms range,

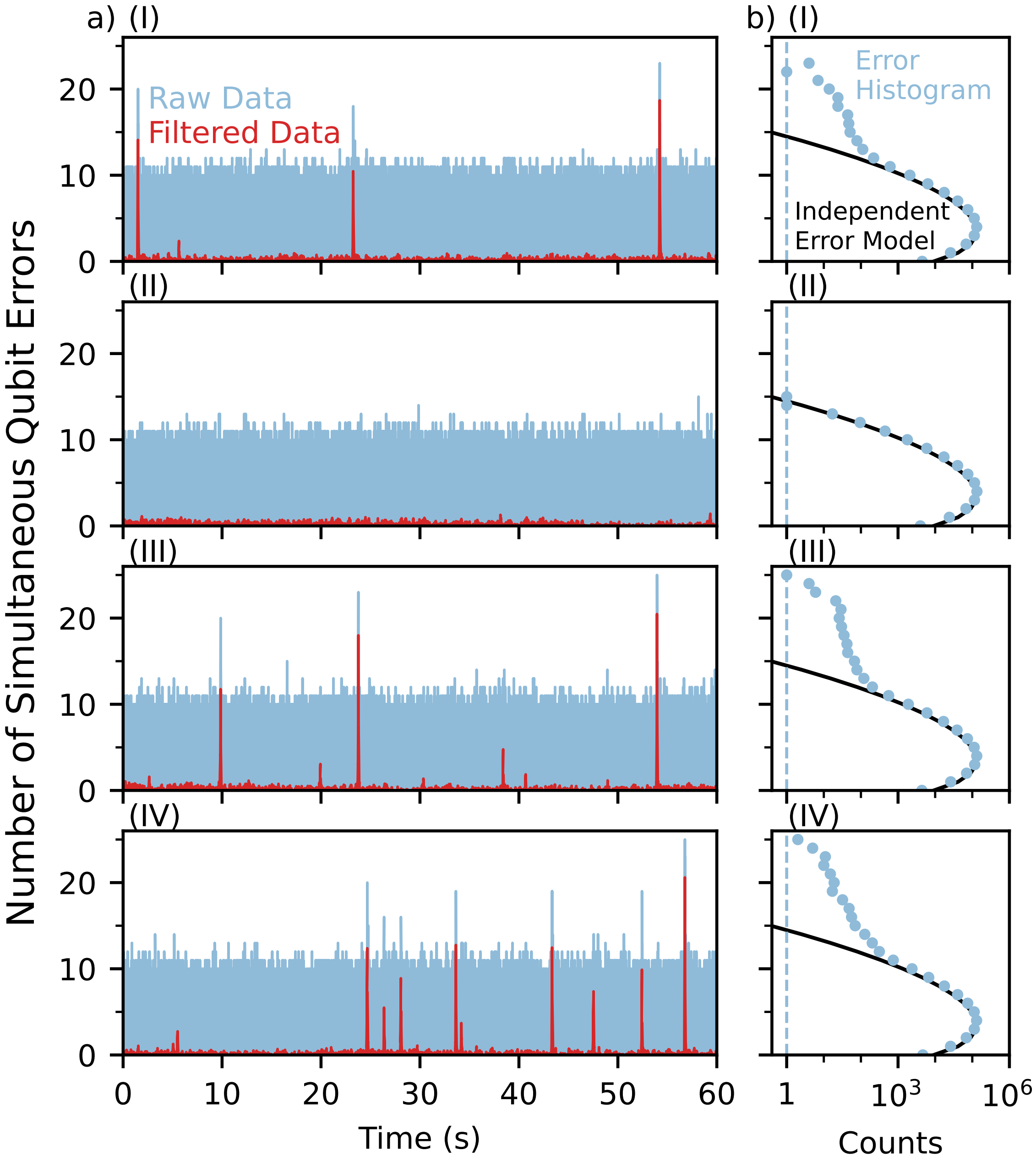


Figure 5.2: **Identifying events and background error.** (I-IV) Sequential datasets selected from a series of 100 datasets of 60 seconds each, with a time between data points of 100 μs. (a) Raw (blue) and filtered (red) timeseries. Matched filtering allows for identifying events otherwise obscured by background noise. Events occur independently every 10 seconds on average. (b) Histograms of the number of errors, along with a model of independent errors arising from qubit T_1 and readout. In the absence of events the data closely correspond to this model, whereas when events occur high error counts appear.

and that peak heights range from the minimum identifiable by our analysis up to the full number of qubits used. We also extracted 326 time periods between events occurring in the same dataset and find a strong correspondence to an exponential distribution with an average event rate $\lambda = 1/(10\text{ s})$. This indicates that the occurrence of the events is independent over time, occurring on average every 10 s without significant bunching or anti-bunching. This timescale is long compared to the typical qubit coherence times, and therefore will have a limited influence on typical qubit T_1 measurements [47]. However, this timescale is quite short compared to the run time of error corrected algorithms, which is projected to be several hours [15], so any attempt to preserve a logical state for computation is very likely to see such an event.

5.5 Impact Localization and Evolution

We now turn to experiments with higher time resolutions in order to observe the evolution of individual events as they progress. Our use of a recently developed reset protocol [1] was key in allowing us to achieve 3 μs intervals between measurements and thereby acquire resolution inside the rising edge of the event. In Fig. 5.3a, we show the raw time trace focusing on the start of an event, with Fig. 5.3b showing the longer tail of the event. We find three distinct timescales; an immediate jump in error from baseline at ∼4 errors to ∼10 errors in only ∼10 μs, a slower saturation up to a maximum of ∼15 over the following ∼1 ms, and a typical ∼25 ms exponential decay back towards baseline. Fig. 5.3c shows heatmaps of the errors over the device averaged over a 300 μs window: (1) shows the baseline performance starting 400 μs prior to the impact, displaying homogeneous low error rates. (2) includes the immediate jump to elevated error rates, displaying a localized hot spot with radius ≈ 2mm where the highest error rates are concentrated. (3) shows the end of the saturation regime at 1.5 ms following the impact, where the

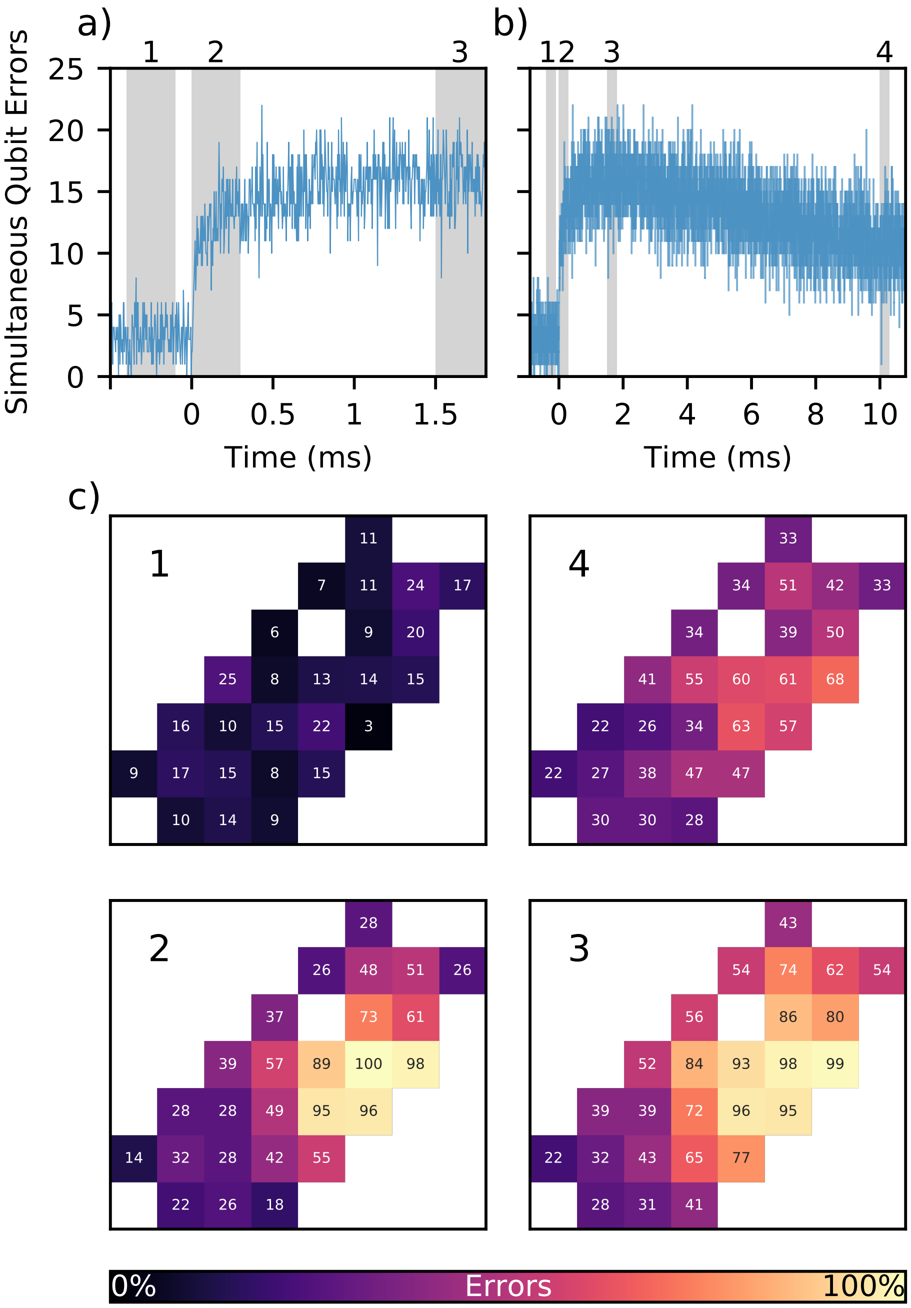


Figure 5.3: **Localization and spread of error.** (a-b) Timeslices of the same event taken from a dataset with a sampling time of 1 μs and an interval of 3 μs between datapoints. The number of errors jumps up from a baseline of $\sim$ 4 to $\sim$10 errors in around 10 μs, then rises to $\sim$ 15 errors in around 1 ms, before returning to the baseline following an exponential decay with a time constant of $\sim$ 25 ms. (c) Heatmaps of the qubit patch, showing the error rate in percent averaged over 300 μs slices located (1) before the event, (2) at the initial impact, (3) after the rise to the peak value, (4) during the recovery of equilibrium. High error rates are initially localized to a small number of qubits, but spread through the device over the course of the event.

hot spot has grown in size and all of the qubit patch sees noticeably elevated error rates. Finally, (4) shows the performance 10 ms after the event during the exponential tail. The initial impact site is still visible but less distinct from the surrounding area. Error rates throughout the chip are still noticeably elevated above baseline levels. Two further events at this level of time resolution are included in the Supplementary Information for comparison, each featuring the same timescales and showing clear localization around different points on the chip.

This is direct evidence of a localised impact on the device which induces high error rates as it spreads over the chip. It further identifies the relevant timescales for the dynamics associated with the initial impact and spread, providing the first insights into the device physics underlying these processes. The spread of the initial hot spot arises from the interplay between phonons traveling through the substrate before being absorbed, and the rate of quasiparticles recombining and emitting further phonons in the qubit layer. From modeling the propagation of phonons and recombination of quasiparticles in the device, we estimate a timescale of $\sim 180\ \mu$sat the start of this process, in reasonable agreement with the timescale of the rise in error. We include details on these interactions and estimates in the Supplementary Information. As the event continues, the energy will be distributed over a larger volume, slowing the rate of recombination, and explaining the qualitative shape of the rising behaviour seen here. These timescales will be also influenced by additional structures and materials on the chip, including the indium bumpbonds and second substrate in a flip chip device. As devices continue to grow in size and complexity, and choices of materials become more diverse, one could expect to see more non-trivial dynamics in response to high-energy stimuli.

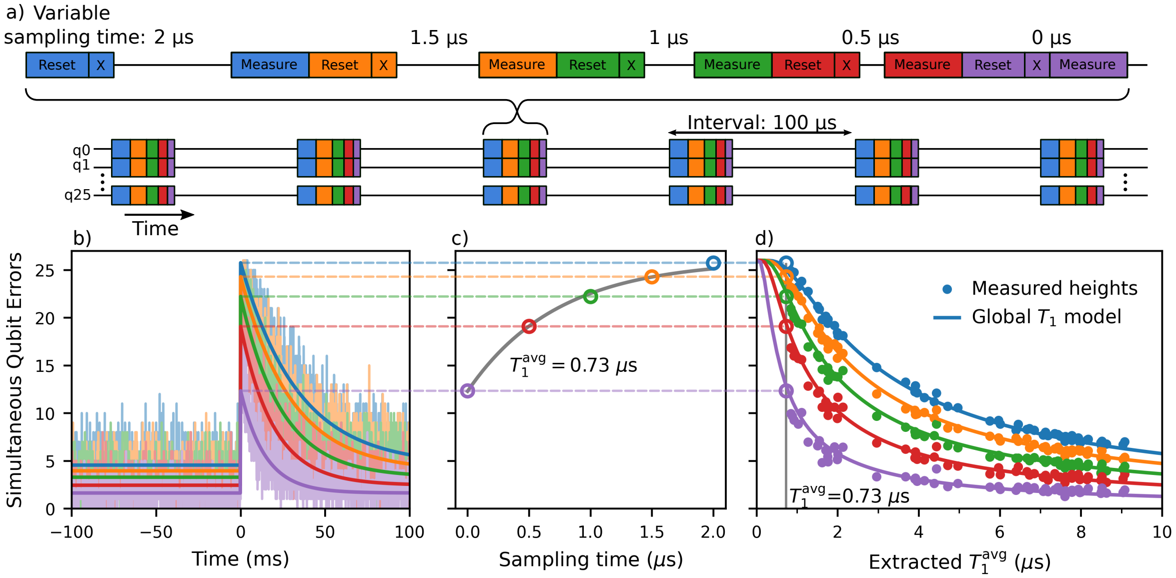


Figure 5.4: **Extracting energy decay times during events.** (a) A time-multiplexed rapid repetitive correlated sampling (T-RReCS) experiment, where each cycle consists of five measurements of varying sampling time. (b) A single event shown in five concurrent time-series with different sampling times, with truncated exponential fits to extract peak heights. (c) Peak heights h vs. sampling time t_{sampling}, showing that the peak heights scale exponentially with sampling time, as expected for chip-wide quasiparticle-limited T_1 times. T_1^{avg} represents the average T_1 over all qubits at the peak of the event. (d) Peak heights h plotted vs. extracted T_1^{avg} for 37 events, where each vertical set of 5 points are the peak heights extracted for one event. The data correspond well to a global T_1 model with no free parameters calculating the expected number of errors from $N_Q = 26$ qubits with a homogeneous T_1 equal to the average T_1^{avg}. This shows the events can be summarized by a chip-wide drop in T_1 that can be inferred directly from the peak height and sampling time. Grey line indicates the decay time extracted in (c).

5.6 Quasiparticle Signature and Event Magnitude

One key signature of a quasiparticle poisoning event is a suppressed qubit T_1, producing an exponential relationship between error rate and sampling time. For a chip-wide event, we should also see that the absolute number of errors is predicted by a limited T_1 across all qubits. Here, we use the speed of our approach to perform Time-multiplexed RReCS (T-RReCS), rapidly cycling through sequential measurements. Each measurement probes for a varying sampling time, between 2 μs and 0 μs as shown in Fig. 5.4a. We separate this data into 5 concurrent time-series, one for each sampling time.

We show a time slice from one such experiment in Fig. 5.4b. To extract an effective T_1 during the event, we first perform an exponential fit to find the height of the peak for each sampling time. While this fit neglects the rising behaviour shown previously, it is dominated by the long 25 ms tail of the event and produces an accurate estimate of the number of errors at the peak of the event. Fig. 5.4c shows the fitted heights plotted versus sampling time. We fit the heights to a two-parameter exponential model $h = a[1 - \exp(-t_{\text{sampling}}/T_1^{\text{avg}})]$, where h is the height of the fitted peak, and t_{sampling} is the sampling time. The timescale T_1^{avg} extracted from the relative peak heights is analogous to an average T_1 over the device at the peak of the event. To confirm the validity of this interpretation, we then compare this extracted T_1^{avg} to a simple global T_1 model for predicting the absolute peak heights, as shown in Fig. 5.4d. Each vertical group of points represents 5 heights extracted from a single event. The absolute peak heights correspond well to a zero-parameter model; $h = N_Q(1 - \exp(-t'_{\text{sampling}}/T_1^{\text{avg}}))$, corresponding to the expected number of qubit errors for $N_Q = 26$ qubits with identical T_1 equal to T_1^{avg}. Here, $t'_{\text{sampling}} = t_{\text{sampling}} + 500$ ns accounts for the finite readout time, as any decay during the first half of measurement integration of 1 μs will be recorded as an error.

This shows that events can be summarized by a global drop in effective T_1 during an event. The presence of exponential scaling provides direct evidence of decay errors induced by excess quasiparticles in the qubits, rather than by coherent control or readout failure. Further, the magnitude of the T_1 drop can be inferred directly from the peak height. While there is localization on short timescales, the events generically affect all qubits enough to correspond well to a simple homogeneous model. This allows us to infer the density of quasiparticles induced across the device at the peak of an event, which we find to be compatible with our theoretical modeling. Details are included in the supplementary. The largest detected events experience excess quasiparticle densities of $x_{qp} \approx 3 \times 10^{-5}$, producing a chip-wide reduction in T_1 to less than 1 μs.

5.7 Discussion

We have shown direct measurements of widespread, long-lived and highly detrimental correlated events in a practical quantum processor, presenting an existential challenge to quantum error correction. Our findings provide a natural explanation for the observations of short periods of significantly elevated error found in recent experiments with stabilizer codes [1, 2]. Our findings are also relevant for other solid-state quantum systems, such as spin qubits that would be sensitive to the induced charges [108], and Majorana fermion devices that would be sensitive to induced quasiparticle population [109, 110].

While the incidence rate can be reduced by shielding [47, 98], the ability of even a single event to cause widespread failure and the need to perform calculations on the order of hours [15] means that reducing the event rate alone is unlikely to be sufficient for practical computation. Rather, mitigation efforts need to be introduced on the chip itself [106].

The efficacy of quasiparticle and phonon trapping in qubits is well understood [111, 104, 112] and has improved device performance through a reduction of the steady-state background as well as injected quasiparticle populations. Yet, given the large energy scales, the fast dynamics involved, and the interplay arising from continuous conversions between phonons and quasiparticles, it is an open question whether trapping approaches are sufficient.

Recent work in astronomical detectors with comparably large devices [105] suggests a promising pathway toward successful on-chip mitigation. Highly-correlated error events have been significantly reduced through the introduction of membrane structures to reduce phonon propagation, and a tantalum backplane for downconverting phonons. Integrating such techniques into a large-scale system without reducing performance, especially as current systems are close to the error correction threshold, presents a vital challenge.

Crucially, adopting such approaches successfully will rely on an understanding of the detailed physics of the events themselves, particularly as devices continue to grow in scale and complexity. We hope that our work spurs research into the physics of correlated events and accelerates the development of mitigation efforts that help enable quantum error correction at scale.

References

[1] Matt McEwen et al. "Removing leakage-induced correlated errors in superconducting quantum error correction". In: *Nature Communications* 12.1 (Dec. 2021), p. 1761. DOI: `10.1038/s41467-021-21982-y`.

[2] Google Quantum AI et al. "Exponential suppression of bit or phase errors with cyclic error correction". In: *Nature* 595.7867 (July 15, 2021). _eprint: 2102.06132, pp. 383–387. DOI: `https://doi.org/10.1038/s41586-021-03588-y`.

[3] Matt McEwen et al. “Resolving catastrophic error bursts from cosmic rays in large arrays of superconducting qubits”. In: *Nature Physics* 18.1 (Jan. 2022), pp. 107–111. DOI: `10.1038/s41567-021-01432-8`.

[15] Craig Gidney and Martin Ekerå. “How to factor 2048 bit RSA integers in 8 hours using 20 million noisy qubits”. In: *Quantum* 5 (Apr. 15, 2021). Publisher: Verein zur Förderung des Open Access Publizierens in den Quantenwissenschaften, p. 433. DOI: `10.22331/q-2021-04-15-433`.

[26] Austin G. Fowler et al. “Surface codes: Towards practical large-scale quantum computation”. In: *Physical Review A* 86.3 (Sept. 18, 2012). Publisher: American Physical Society, p. 032324. DOI: `10.1103/physreva.86.032324`.

[35] Frank Arute et al. “Quantum supremacy using a programmable superconducting processor”. In: *Nature* 574.7779 (Oct. 24, 2019). Publisher: Nature Publishing Group, pp. 505–510. DOI: `10.1038/s41586-019-1666-5`.

[43] Austin G. Fowler and John M. Martinis. “Quantifying the effects of local many-qubit errors and nonlocal two-qubit errors on the surface code”. In: *Physical Review A* 89.3 (Mar. 12, 2014). Publisher: American Physical Society, p. 032316. DOI: `10.1103/PhysRevA.89.032316`.

[45] K. Serniak et al. “Hot Nonequilibrium Quasiparticles in Transmon Qubits”. In: *Physical Review Letters* 121.15 (Oct. 10, 2018). Publisher: American Physical Society, p. 157701. DOI: `10.1103/PhysRevLett.121.157701`.

[46] John M. Martinis, M. Ansmann, and J. Aumentado. “Energy Decay in Superconducting Josephson-Junction Qubits from Nonequilibrium Quasiparticle Excitations”. In: *Physical Review Letters* 103.9 (Aug. 26, 2009). Publisher: American Physical Society, p. 097002. DOI: `10.1103/PhysRevLett.103.097002`.

[47] Antti P. Vepsäläinen et al. “Impact of ionizing radiation on superconducting qubit coherence”. In: *Nature* 584.7822 (Aug. 27, 2020). Publisher: Springer Science and Business Media LLC, pp. 551–556. DOI: `10.1038/s41586-020-2619-8`.

[49] Jens Koch et al. “Charge-insensitive qubit design derived from the Cooper pair box”. In: *Physical Review A* 76.4 (Oct. 12, 2007). Publisher: American Physical Society, p. 042319. DOI: `10.1103/PhysRevA.76.042319`.

[52] Charles Neill. “A path towards quantum supremacy with superconducting qubits”. PhD thesis. University of California, Santa Barbara, 2017.

[53] Fei Yan et al. “Tunable Coupling Scheme for Implementing High-Fidelity Two-Qubit Gates”. In: *Physical Review Applied* 10.5 (Nov. 28, 2018). Publisher: American Physical Society, p. 054062. DOI: `10.1103/PhysRevApplied.10.054062`.

[54] B. Foxen et al. “Demonstrating a Continuous Set of Two-qubit Gates for Near-term Quantum Algorithms”. In: *Physical Review Letters* 125.12 (Sept. 15, 2020). _eprint: 2001.08343, p. 120504. DOI: `https://doi.org/10.1103/PhysRevLett.125.120504`.

[57] B Foxen et al. “Qubit compatible superconducting interconnects”. In: *Quantum Science and Technology* 3.1 (Jan. 1, 2018). _eprint: 1708.04270, p. 014005. DOI: `https://doi.org/10.1088/2058-9565/aa94fc`.

[61] Paul V. Klimov et al. “The Snake Optimizer for Learning Quantum Processor Control Parameters”. In: (2020). _eprint: 2006.04594. DOI: `https://doi.org/10.48550/arXiv.2006.04594`.

[65] Barbara M. Terhal. “Quantum error correction for quantum memories”. In: *Reviews of Modern Physics* 87.2 (Apr. 7, 2015). Publisher: American Physical Society, pp. 307–346. DOI: `10.1103/RevModPhys.87.307`.

[89] Morten Kjaergaard et al. "Superconducting Qubits: Current State of Play". In: *Annual Review of Condensed Matter Physics* 11.1 (Mar. 10, 2020), pp. 369–395. DOI: `10.1146/annurev-conmatphys-031119-050605`.

[98] L. Cardani et al. "Reducing the impact of radioactivity on quantum circuits in a deep-underground facility". In: *Nature Communications* 12.1 (Dec. 2021). _eprint: 2005.02286, p. 2733. DOI: `https://doi.org/10.1038/s41467-021-23032-z`.

[99] C. D. Wilen et al. "Correlated charge noise and relaxation errors in superconducting qubits". In: *Nature* 594.7863 (June 17, 2021), pp. 369–373. DOI: `10.1038/s41586-021-03557-5`.

[100] Particle Data Group et al. "Review of Particle Physics". In: *Progress of Theoretical and Experimental Physics* 2020.8 (Aug. 14, 2020), p. 083C01. DOI: `10.1093/ptep/ptaa104`.

[101] M. Lenander et al. "Measurement of energy decay in superconducting qubits from nonequilibrium quasiparticles". In: *Physical Review B* 84.2 (July 1, 2011). Publisher: American Physical Society, p. 024501. DOI: `10.1103/PhysRevB.84.024501`.

[102] Antonio D. Córcoles et al. "Protecting superconducting qubits from radiation". In: *Applied Physics Letters* 99.18 (Oct. 31, 2011). Publisher: AIP Publishing, p. 181906. DOI: `10.1063/1.3658630`.

[103] G. Catelani et al. "Quasiparticle Relaxation of Superconducting Qubits in the Presence of Flux". In: *Physical Review Letters* 106.7 (Feb. 16, 2011). Publisher: American Physical Society, p. 077002. DOI: `10.1103/PhysRevLett.106.077002`.

[104] C. Wang et al. "Measurement and control of quasiparticle dynamics in a superconducting qubit". In: *Nature Communications* 5.1 (Dec. 2014), p. 5836. DOI: `https://doi.org/10.1038/ncomms6836`.

[105] K. Karatsu et al. “Mitigation of cosmic ray effect on microwave kinetic inductance detector arrays”. In: *Applied Physics Letters* 114.3 (Jan. 21, 2019). Publisher: AIP Publishing, p. 032601. DOI: `10.1063/1.5052419`.

[106] John M. Martinis. “Saving superconducting quantum processors from decay and correlated errors generated by gamma and cosmic rays”. In: *npj Quantum Information* 7.1 (Dec. 2021), p. 90. DOI: `10.1038/s41534-021-00431-0`.

[107] M. Houzet et al. “Photon-Assisted Charge-Parity Jumps in a Superconducting Qubit”. In: *Physical Review Letters* 123.10 (Sept. 6, 2019). Publisher: American Physical Society, p. 107704. DOI: `10.1103/PhysRevLett.123.107704`.

[108] Floris A. Zwanenburg et al. “Silicon quantum electronics”. In: *Reviews of Modern Physics* 85.3 (July 10, 2013). Publisher: American Physical Society (APS), pp. 961–1019. DOI: `10.1103/revmodphys.85.961`.

[109] Diego Rainis and Daniel Loss. “Majorana qubit decoherence by quasiparticle poisoning”. In: *Physical Review B* 85.17 (May 30, 2012). Publisher: American Physical Society, p. 174533. DOI: `10.1103/PhysRevB.85.174533`.

[110] Torsten Karzig, William S. Cole, and Dmitry I. Pikulin. “Quasiparticle Poisoning of Majorana Qubits”. In: *Physical Review Letters* 126.5 (Feb. 4, 2021). Publisher: American Physical Society, p. 057702. DOI: `10.1103/PhysRevLett.126.057702`.

[111] I. Nsanzineza and B. L. T. Plourde. “Trapping a Single Vortex and Reducing Quasiparticles in a Superconducting Resonator”. In: *Physical Review Letters* 113.11 (Sept. 12, 2014). Publisher: American Physical Society, p. 117002. DOI: `10.1103/PhysRevLett.113.117002`.

[112] Fabio Henriques et al. "Phonon traps reduce the quasiparticle density in superconducting circuits". In: *Applied Physics Letters* 115.21 (Nov. 18, 2019). Publisher: AIP Publishing, p. 212601. DOI: `10.1063/1.5124967`.

Chapter 6

Walking Surface Codes

This as yet unpublished work was done in close collaboration with Craig Gidney. While not directly on the subject of correlated errors, it provides a useful background in a new approach to error correcting circuits which was inspired by the desire to have a cheaper way of removing leakage from the code circuit.

6.1 Introduction

The traditional approach to fault-tolerance in stabilizer codes is to repeatedly measure the code stabilizers, continuously projecting back into the logical subspace [25, 26, 65]. However, performing these measurements experimentally requires a decomposition of the stabilizer measurements into a circuit capable of being run on quantum hardware. The details of the circuit decomposition can have a significant impact on how the code performs in reality. A common strategy for decomposing a stabilizer code into a circuit is to choose a fixed set of physical qubits as the *data qubits*, add a fixed set of ancillary *measure qubits*, and to implement entangling operations to assemble the stabilizer information on the measure qubit to be read out directly. These operations can then be easily repeated to achieve fault-tolerance. This strategy suits architectures with fixed arrays of qubits with local connections capable of performing 2-qubit entangling gates, such as large superconducting qubit arrays [4, 36, 37]. Other hardware architectures with different elementary operations can similarly be targeted by different circuit decomposition strategies [113, 114, 115].

The surface code remains a leading candidate for experimental implementations, as it is naturally instantiated on a practical square lattice and displays relatively high performance under circuit noise. There are two common ways of decomposing the surface code into a circuit for measure qubits and entangling 2-qubit gates: First, the two kinds of stabilizers can be measured sequentially, as in [36]; or second, the two kinds of stabilizers can be measured simultaneously, as in [4]. The relative performance of these two strategies is dominated by the details of the noise model. In each case, the stabilizers are measured the same number of times and with the same relative frequency. The code itself is logically "the surface code" in both cases, and is decoded identically by matching.

The constructions in this work build on the latter approach, where all stabilizers are measured simultaneously. One interesting feature of this circuit is that after two layers of entangling gates, the qubits are collectively in a larger surface code state which we call the "mid-cycle state", as shown in Figure 6.1. It is an unrotated surface code state; there are twice the number of qubits involved for the same code distance, and the instantaneous stabilizers are apparently at 45 degrees to the typical rotated code stabilizers. The final two layers of entangling gates return the code to its "end-cycle" configuration and unentangle the measure qubits, permitting them to be measured and telling the experimenter the outcome of the stabilizer measurement.

Further freedom in how the stabilizer code is decomposed into a circuit has so far remained relatively unexplored, primarily because most changes make the code performance worse. Changing the ordering of the entangling gates can change the direction that hook errors propagate, producing low-weight errors that can halve the final code distance. In the simultaneous measurement implementation, arbitrarily changing the gate ordering can have one partially measured stabilizer disturb another, destroying the code structure and affecting logical performance quite noticeably. When the circuit is modified for another purpose such as leakage reduction [94, 96], additional noisy operations are added to the circuit, creating a trade-off between the additional errors induced by the operations and those they remove or suppress. Finding circuit decomposition that preserve the logical code structure and don't introduce additional error mechanisms requires care and attention.

In this work, we introduce a new family of circuit decompositions for the surface code. These constructions permit new freedom in the embedding of the code on a grid of physical qubits, including moving the boundaries of a surface code patch without introducing additional gate layers, as indicated in Figure 6.1b. Further, these decompositions leave the logical structure of the code intact, avoiding the introduction of any new exotic or hook errors that negatively impact the code distance. They are compatible standard techniques for decoding and error analysis, which we demonstrate by Clifford benchmarking various examples of these constructions using the same decoder throughout.

Using these techniques, we express new primitive behaviours for lattice surgery patches, and present possible uses of these primitives: at the physical level for leakage reduction without additional gate overhead; and at the logical level for cheaper operations on densely packed logical qubits.

6.2 Detecting Regions

In stabilizer circuits, the occurrence of errors are identified by "detectors". In general, a detector is a set of measurements in a circuit which have a deterministic product in the absence of noise. A detector producing a outcome other than that expected product a “detection event”, which provides information about the occurrence of unexpected errors.

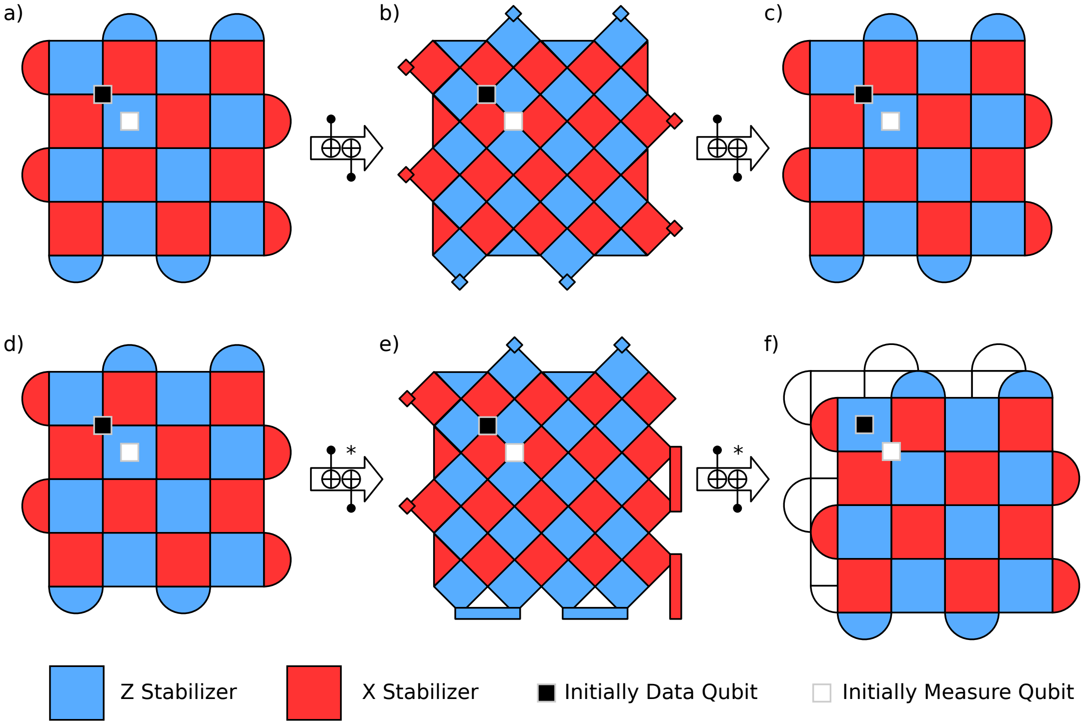


Figure 6.1: **Mid-cycle states in the surface code.** A schematic indicating the state of qubits in a surface code where all stabilizers are to be measured simultaneously using measure qubits. a) A rotated surface code state, indicated by a combination of weight-4 (square) and weight two (semi-circlular) stabilizer plaquettes, with included qubits being those at the corners of the respective shapes. Measure qubits at the center of each stabilizer plaquette allow the stabilizer to be measured. Blue plaquettes indicate Z stabilizers, and red indicate X stabilizers. An initial data and measure qubit are indicated by the black and white squares with grey borders. b) The mid-cycle state after the first two layers of entangling gates. This is an unrotated surface code state, with twice the number of qubits as the state in a due to the inclusion of the measure qubits, but the same distance. c) The final state after the last two entangling gates, identical to the initial state in (a). d) The same initial state shown in (a). e) An unusual mid-cycle state, produced by different choices of Reset and CNOT gates (asterix). f) An unusual final state, produced by different choices of the CNOT gates and measurements. It is equivalent to the initial state in d), but shifted diagonally by one step on the physical qubit grid, which also has the effect of exchanging the roles of measure and data qubits; The black square which was originally a data qubit is now in the center of a plaquette, and the white square that was originally a measure qubit is now at the corner of plaquettes.

For each detector, there is a small closed region of the stabilizer circuit where inserting an error can flip the detector result, which we call the “detecting region”. Figure 6.2 shows some simple detecting regions. Reset gates produce states that are stabilized by Z, and as such can terminate a Z-type section of a detecting region. Measurements in the Z basis terminate Z-type sections of detecting regions. Other gates propagate the detecting region following their stabilizer tableau.

A detecting regions is sensitive to any non-commuting error on sections of the circuit that they include, which is to say that any error applied to the circuit that doesn’t commute with a detecting region will flip the value of the associated detector. It is worth explicitly recognising that the detecting region is a collections of sections of the circuit, rather than a boundary enclosing a part of the circuit. The “boundaries” of the detecting region are the reset and measurements that it terminates on.

Figure 6.3 shows some more complex detectors for the bit-flip and phase-flip repetition codes with reset. In these codes, the detectors in the bulk are time-neighbouring pairs of measurements on the same measure qubit. These regions consist of a small closed and relatively local part of the circuit, visually indicating how much of the circuit each detector is responsible for.

When a circuit with annotated detector measurement sets is provided, the detecting region can be discovered quickly, either by brute force search, or by propagating Z-type sections of the detecting region out of involved measurements backwards through the circuit to form a closed shape. When the detecting regions are provided, the detectors are given implicitly; the set of measurements that a detecting region terminates on. Note that regions can transit through measurements without terminating on them, which does not include the measurement in the detector. Finally, if only the gates of the circuit are provided, the act of finding the appropriate detectors and so defining the code structure can be aided by trying to find detecting regions: starting with Z-type sections terminating

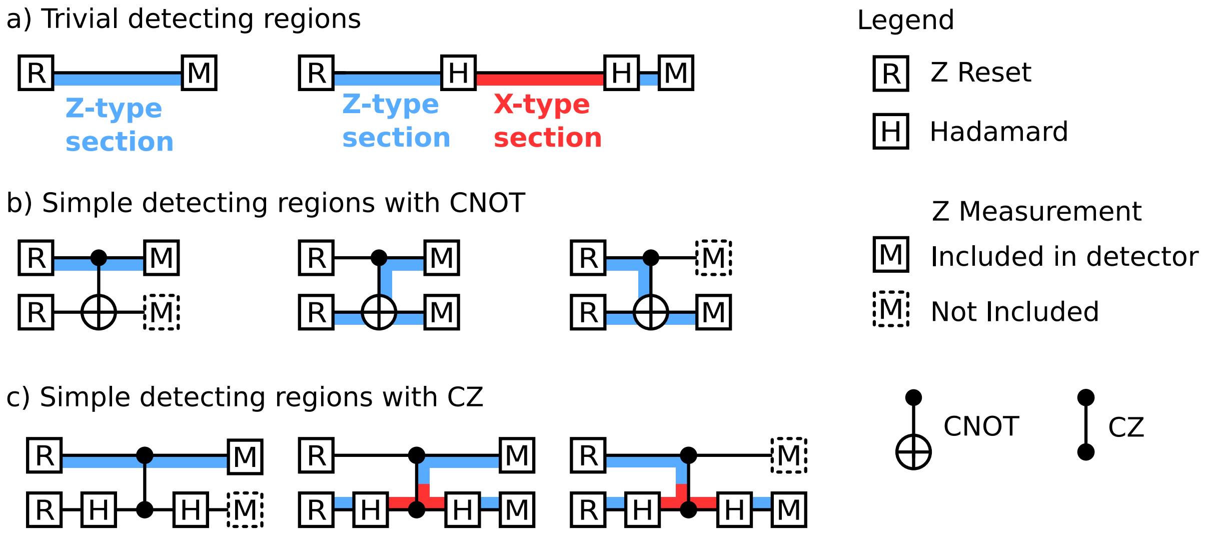


Figure 6.2: **Simple detecting regions.** a) Detecting regions for 1-qubit circuits. Here, the detectors are just the single measurement in the circuit, which should have a deterministic outcome in the absence of noise. Z Reset gates produce states that are stabilized by Z, indicated by a Z-type section of the detecting region. Z Measurements terminate Z-type sections of the detecting region. Hadamard gates change the type of the detecting region as it moves through the circuit, producing X-type sections of the detecting region. b) Three copies of the a simple circuit with a CNOT gate, indicating the three possible detectors and their detecting regions. Note CNOT gates do not change the type of the detecting region. Any two of these detectors form a generating set for the full set of detectors. c) The same circuit and detectors compiled into CZ and Hadamard gates. Note that CZ gates also produce X-type sections of the detecting region.

on measurements and reset gates and propagating them around the circuit in search of small closed shapes can be helpful in finding for good detectors, although the confirmation that a detector structure is good must be left to explicit benchmarking.

The procedure of propagating detecting regions is even more easily applied when using the ZX-calculus, where the stabilizer rules may be followed at each node rather than for each gate, and rewriting rules can simplify circuits that were complicated by their compilation to hardware gates. Again, this is shown in Figure 6.3 for various repetition codes.

In the repetition and surface codes, the detecting regions exist for two neighbouring

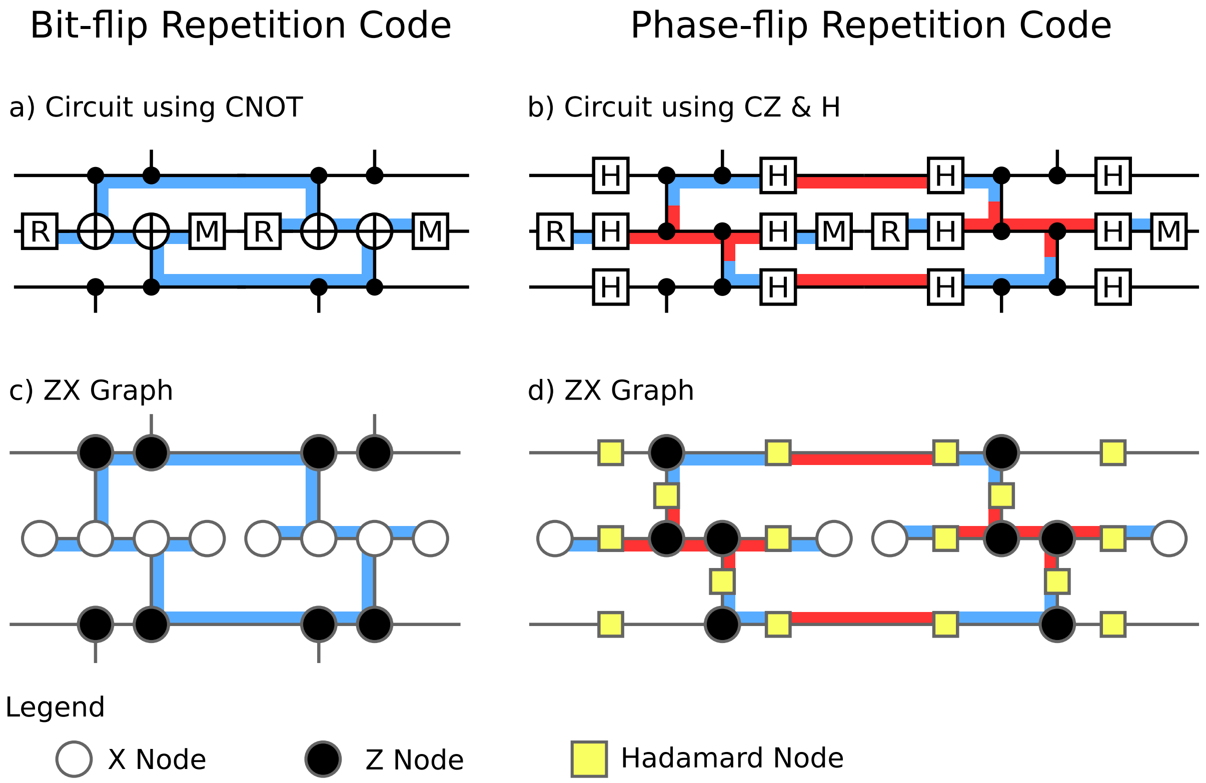
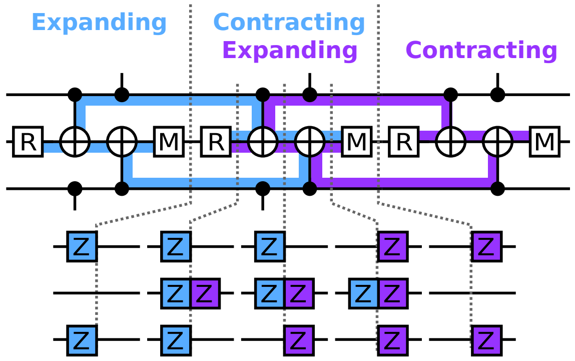


Figure 6.3: **Detecting regions in the repetition codes.** a) A part of a circuit for a bit-flip repetition code, using CNOT and Reset gates. The two time-neighbouring measurements shown form a detector. The associated detecting region is highlighted (blue), and is Z-type throughout. Specifically, the region covers neighbouring data qubits in Z-type during the measurement, reflecting the bit-flip code's ZZ stabilizers. b) A part of a phase-flip repetition code, using CZ and Reset gates. Similarly, the two time-neighbouring measurements shown form a detector. The associated detecting region is highlighted, containing circuit locations of both Z-type (blue) and X-type (red). The "phase-flip" nature of this circuit reflects that the data qubits are covered by X-type regions during measurement, equivalent to defining the code stabilizers as neighbouring XX rather than the typical ZZ. c), d) the ZX-calculus graphs equivalent to the circuits in a and b respectively. CNOTs and CZs are equivalent to two and three nodes respectively, and Z-type resets and measurements are equivalent to single X nodes with only one outgoing edge.

Bit-flip Repetition Code

a) Detecting regions during a cycle



b) All detecting regions for a distance-5 code run for 4 cycles

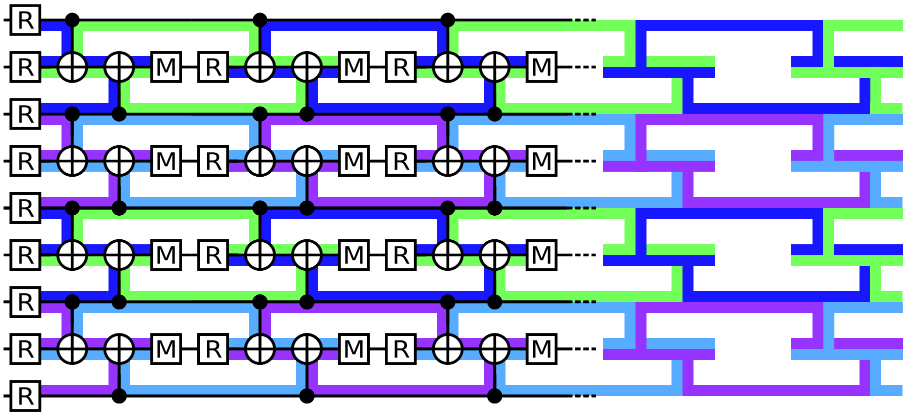


Figure 6.4: **Overlapping detecting regions in the repetition code.** All sections of all regions shown here are Z-type. a) Two neighbouring detecting regions of the bit-flip repetition code (blue, purple), shown over three cycles. Each detecting region covers two cycles: In the first cycle, it emerges from a reset on the measure qubit and expands to cover the code stabilizer. In the second cycle, it contracts from the code stabilizer down to terminate on a measurement. During the middle cycle, where the two shown regions co-exist, time-slices of the regions are shown as their corresponding the terms in the instantaneous stabilizer group. b) All detecting regions for a distance-5 bit-flip repetition code. The detection regions overlap such that all locations in the circuit are covered by two detecting regions, except the boundaries which are covered by 1. Note the smaller detecting regions at the start of the code that emerge from resets on both measure and data qubits. After three cycles, we omit the circuit elements to emphasise that the detecting regions alone fully specify the behaviour of the code.

cycles, and cover the code stabilizer at the time-slice between those cycles. Figure 6.4a illustrates this for the repetition code, where the time-slice between measure and reset gate layers shows the existing region covering the ZZ stabilizer. During a cycle, two detecting regions coexist; one emerging from that cycle's reset gate and expanding to cover the stabilizer, and one contracting from the stabilizer to terminate on that cycle's measurement gate. At the middle of the cycle, we can see the regions collectively produce the same pattern of stabilizers as the typical code state (ZZ on neighbouring qubits), but involving all measure and data qubits, rather than only the data qubits.

Detecting regions overlap to form the code structure. Where two detecting regions overlap, an error that fail to commute with both regions will be flip both regions, corresponding to an edge in the error hypergraph or equivalently a term in the detector error model. In this sense, the detecting regions complementary to the error hypergraph. Figure 6.4b shows the full set of detecting regions for a short repetition code, revealing the pairwise overlap structure; all locations in the bulk of the circuit are covered by two regions, meaning any X error in the bulk will touch two regions and flip two detectors. Boundaries in the repetition code are covered by one detecting region. This structure allows the repetition code to be decoded by matching.

Detecting regions are useful primitive, as they single-handedly define relevant code concepts. They trivially define detectors as the measurements they terminate on. The overlaps between regions define the error hypergraph. The shape and types involved in the regions define the circuit operations from their stabilizer tableaus. Development of new code circuits, and especially understanding behaviour at the boundaries, can be aided by looking at the detecting regions alone, and how different detecting regions can fit together to cover spacetime.

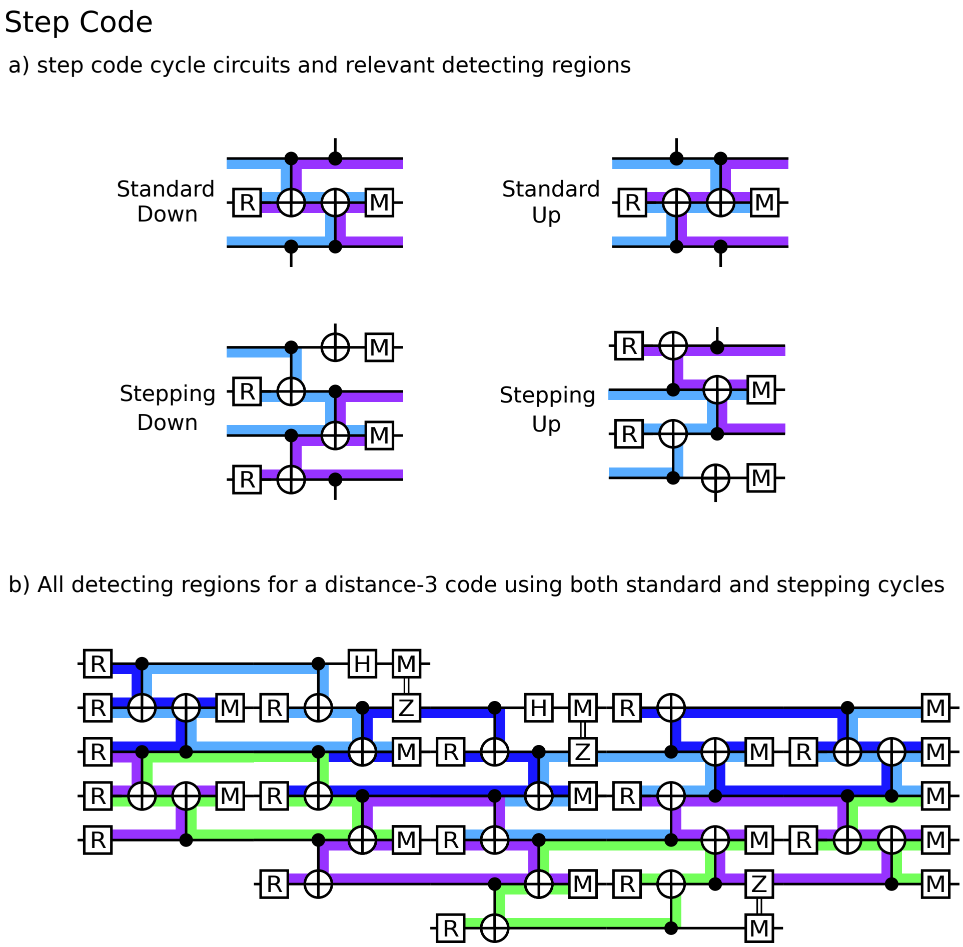

Step Code
a) step code cycle circuits and relevant detecting regions
Standard Down
Standard Up
Stepping Down
Stepping Up
b) All detecting regions for a distance-3 code using both standard and stepping cycles
R
M
H
Z

Figure 6.5: **Circuits and detecting regions in the Step Code.** Logically equivalent to the bit-flip repetition code, the step code provides a choice of cycles and a greater variety of detecting region shapes. All sections of all detecting regions shown are Z-type. a) The four possible cycles in the step code. The Standard cycles are equivalent typical circuits for the bit-flip code. The Stepping cycles measure the same stabilizers as the standard cycle, but the qubits that are measured are not the same qubits that were reset at the beginning of the cycle. This effectively exchanges the roles of data and measure qubits. b) All detecting regions for a distance-3 step code. In order, the cycles are [Standard Down, Step Down, Step Down, Step Up, Standard Down]. The combination of these cycles produces a variety of shapes of detecting regions, but their size and overlapping structure is the same as for the standard repetition code. Notice the inclusion of effective X-basis measurement and Z classical feedback at boundaries being stepped away from; these correct the distance-1 X observable, and are unnecessary when considering the repetition code as a classical code. By combining various cycles, the repetition code state can step in either direction by one physical qubit per cycle.

6.3 Step Code

Now we have the concept of detecting regions in hand, we can consider alternative shapes of detecting regions. Figure 6.5 introduces the cycles of the “Step Code”, a modified repetition code where the identities of measure and data qubits can change.

From a circuit perspective, the step code cycles amount to unusual choices in the directions for the CNOT gates and which qubits to reset and measure. From a detecting region perspective, it amounts to alternative choices for the shape of the region, altering where it terminates. All the cycles given preserve the overall structure of the repetition code: ZZ stabilizers on next neighbouring data qubits are measured in each cycle, and detectors consisting of two measurements cover the bulk of the code circuit. All the “step code” has added to the repetition code is the possibility of unusual circuit embedding on the physical qubits.

One way of understanding the availability of the circuit freedom used in the Step Code is recognising that all cycles have the same structure in their mid-cycle state. Here, all qubits, regardless of previous measure or data roles, are involved in instantaneous ZZ stabilizers with their two physical neighbours. This state “doesn’t remember where it came from” in terms of physical qubits. All that is necessary to preserve the code structure is that whatever half-cycle happens next finishes expanding detecting regions that are expanding, and finishes contracting regions that are contracting — it doesn’t matter to where the expansion and contraction happens. This is a key concept that will apply in the case of the surface code as well.

6.4 Walking Surface Code

Having seen a construction for the repetition code that permits moving the state and exchanging the roles of measure and data qubits without gate overhead, we're encouraged to ask if there is an equivalent construction in the surface code. It is not obvious that this is the case; the 1-dimensional boundaries and classical nature of the repetition code make finding appropriate boundary detecting regions relatively simple.

We begin by considering the standard surface code and its detecting regions. Figure 6.6 shows the bulk surface code circuit, along with time-slices of the detecting region for a distance-7 patch at each point through the circuit.

Prior to the reset gates, the contracting detecting regions correspond to the usual stabilizer plaquettes. After the reset gates, new qubits have been added to various detecting regions: Expanding regions emerge from the reset measure qubits. Contracting regions have the reset measure qubit added to them, now including the qubits at each of 5 vertices, indicated by flag-like pentagonal shapes.

The mid-cycle state after 2 layers of entangling gates is the unrotated surface code state. Notice the highlighted Z-type regions are separated vertically, as necessary to avoid hook errors propagating in a problematic direction. The typical 4-fold rotational symmetry is broken to 2-fold by considering which detecting regions are expanding and which are contracting. Only half of the detecting regions correspond to measurements that will be performed before we return to the mid-cycle state.

Prior to the measurement layer, we have returned to a state similar to the state following the reset layer, but with expanding and contracting detecting regions exchanged. Following measurement, the contracting regions have been terminated, and the expanding regions have become contracting regions. Particularly, the highlighted expanding region has replaced the highlighted contracting region in the same location in the patch.

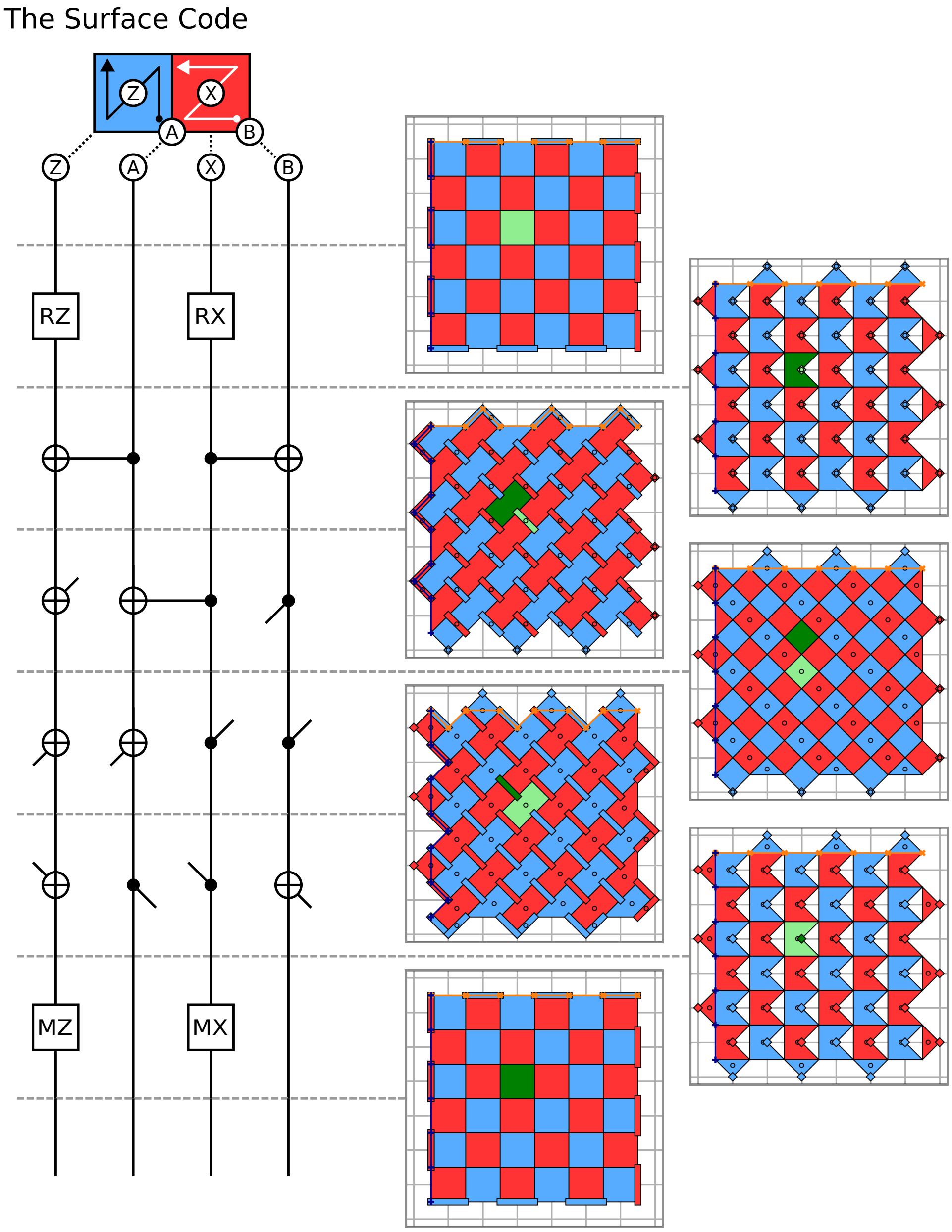


Figure 6.6: **Circuit and detecting regions for a standard surface code cycle.** Left, the circuit for 4 qubits in the bulk of the surface code. Qubits labeled Z and X are measure qubits, and qubits labeled A and B are data qubits. The arrows annotating the plaquettes indicate the gate ordering for that measure qubit. Right, all detecting regions at the indicated circuit time-slice for a distance-7 surface code patch. Gridlines in grey indicate the position of measure qubits. Regions are annotated by a small circle if they are expanding, and are otherwise contracting regions. The logical observable are indicated by orange (X) and bark blue (Z) lines. Two Z-type regions have been highlighted, one contracting (light green) and one expanding (dark green).

The key insight we used to generate the stepping circuit construction was observing that the mid-cycle state had symmetry such that we could choose to contract regions toward different qubits they expanded from. This is shown schematically in Figure 6.7. This change is similar to the modification in circuit to make the classical step code, where we change the final layer of CNOTs. Here we changing the direction of the gate layer on the physical grid as well exchanging target and control. That change alone is sufficient to generate the new detecting region shapes in the bulk, so what remains is to specify the behaviour at the boundaries.

Finding appropriate detecting regions on the boundary proved to be the major challenge. There are a variety of detecting regions (or equivalently, choices of resets, gates and measurements) at the boundaries that permit the code to be constructed and to suppress errors, but many choices permit unfortunate error propagation similar to hook errors, which reduces the code distance to half the patch length. Figure 6.8 schematically shows the bulk circuit and a set of detecting regions at the boundary that do not add unfortunate paths and preserves the graph-like code distance, even when steps are taken in every cycle in any direction. Which gates are included at the boundary is implied by the difference in the time-slices of detecting regions shown.

6.5 Benchmarking

We now turn to numerical benchmark to evidence the claim that this construction is logically equivalent to the surface code, doesn't introduce unfortunate errors and enjoys similar logical performance.

We used the *Stim* software package, and in particular the *Sinter* interface that was recently released. We took up to 100,000,000 shots for each case, but terminated early if we found 1000 logical errors, judging this to be sufficient for good visibility of performance

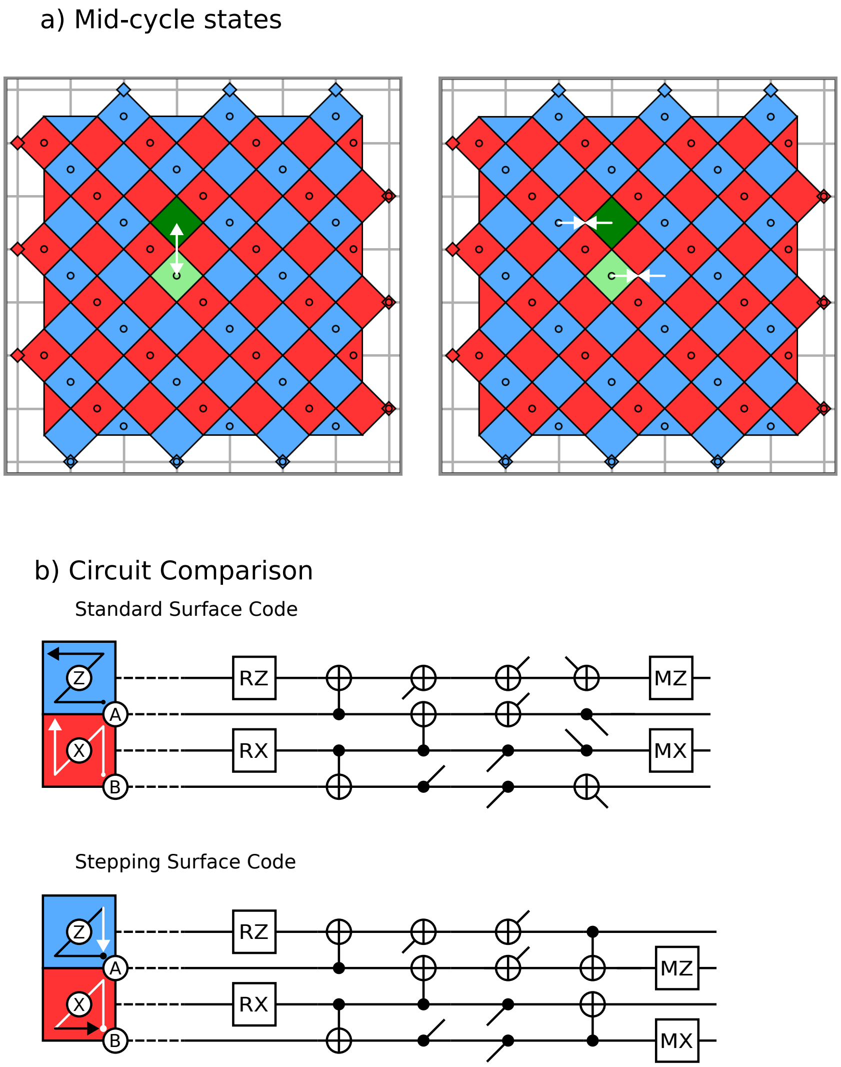


Figure 6.7: **How to step the surface code.** a) Two copies of the mid cycle state for the surface code. On the left white arrows indicate how the highlighted pair of detecting regions (light and dark green) that included the center measure qubit were moved. In the standard code, these will come back together centered on the measure qubit they share. On the right, white arrows show a different choice of regions to move together. These pairs will then be centered on the qubit they share, which was originally a data qubit. b) Part of the bulk circuits for the standard and stepping surface code. The stepping circuit implements the concept above of moving different detecting regions together. The final layer of CNOTs and the layer of measurement has been changed. This has the effect of exchanging the roles of the measure and data qubits.

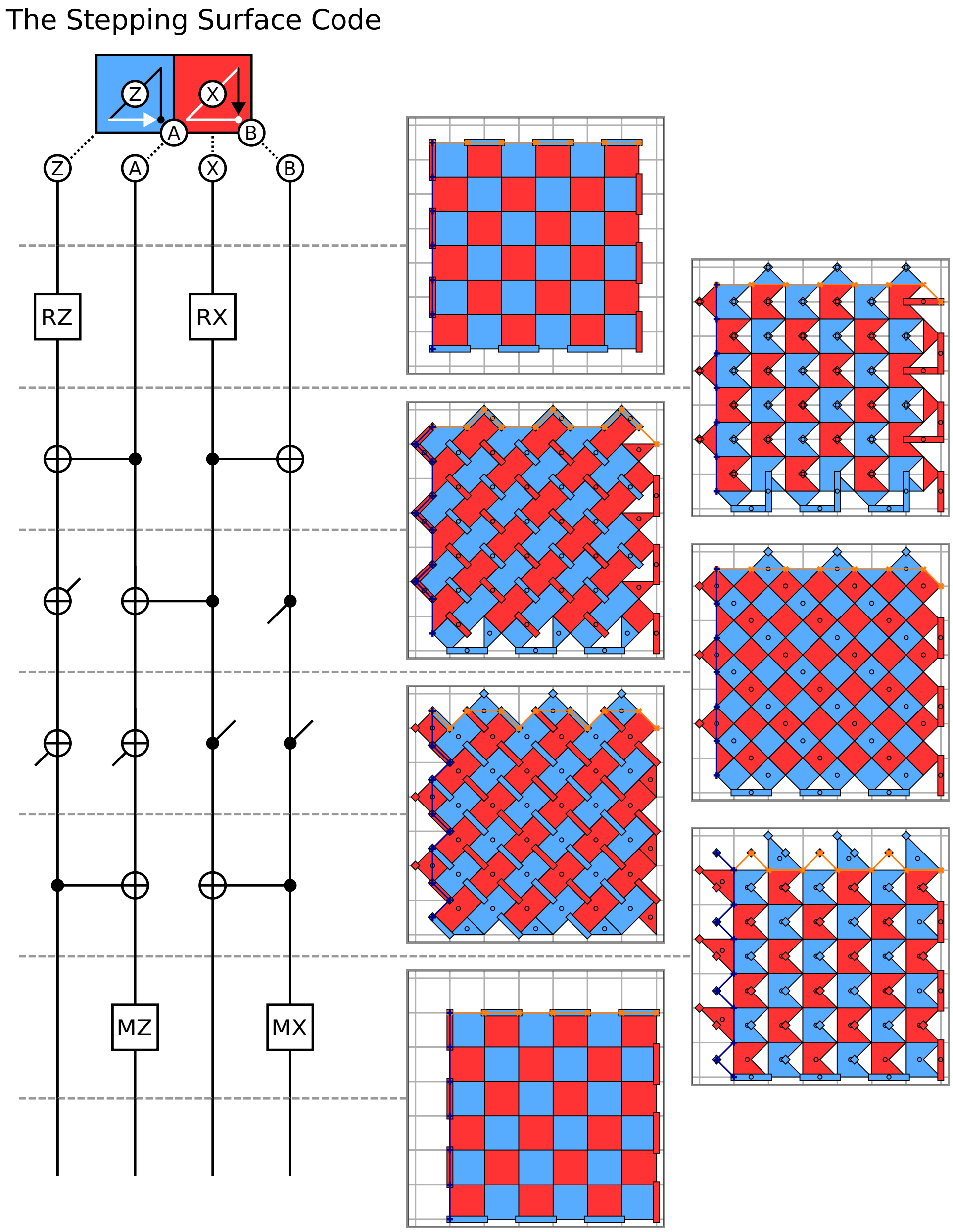


Figure 6.8: **Circuit and detecting regions for a stepping surface code cycle.** Similar to Figure 6.6. Left, the circuit for 4 qubits in the bulk of the patch for a stepping circuit cycle. Right, all detecting regions shown at each time slice. Notice especially at the boundaries, where unusual patterns of qubits are included at the reset layer, and are measured at the measure layer. After the measure layer, the detecting regions correspond to the standard code state shifted diagonally on the grid of physical qubits, having exchanges the roles of data and measure qubits.

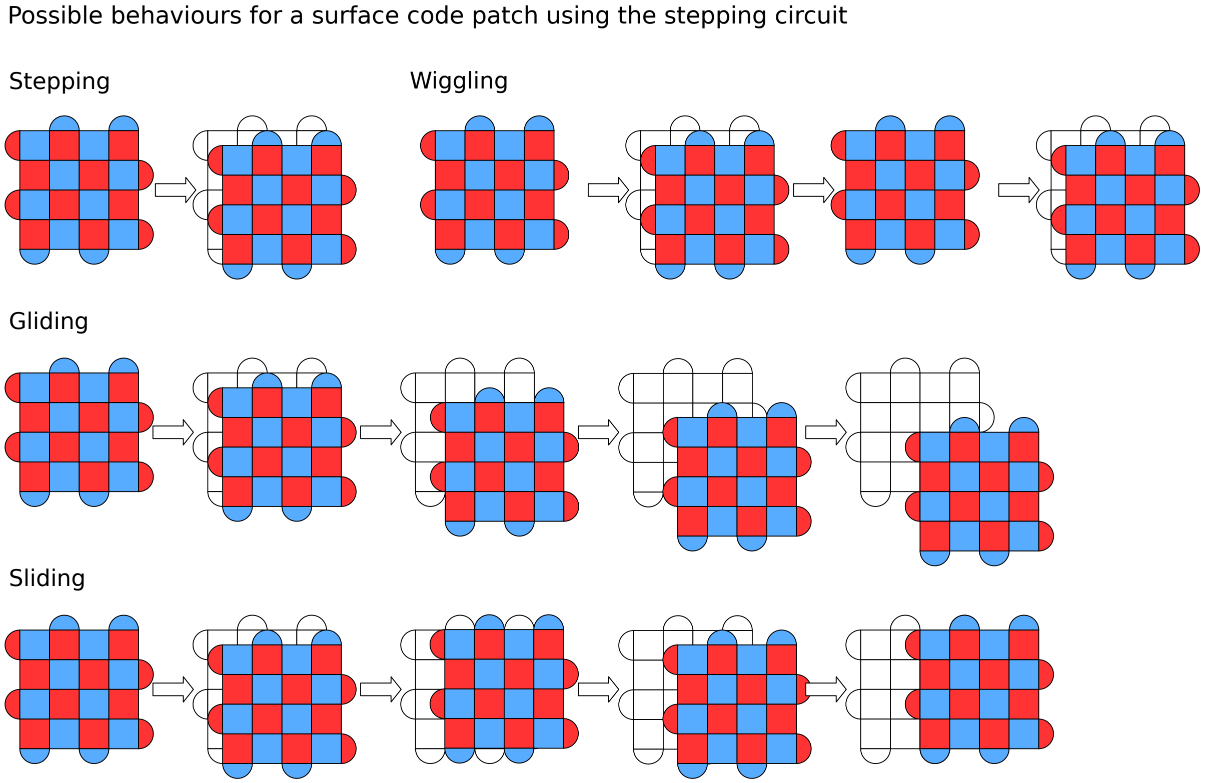

Possible behaviours for a surface code patch using the stepping circuit
Stepping
Wiggling
Gliding
Sliding

Figure 6.9: **Four behaviours for a patch possible using the stepping circuit.** "Stepping" refers to a patch taking a single step diagonally on the physical qubit grid during one cycle. "Wiggling" refers to taking a step in one direction, and taking a step in the opposite direction in subsequent cycles. This exchanged the roles of measure and data qubits each cycle while taking the minimum of extra qubit space. "Gliding" refers to continuing to take steps in the same direction in each cycle, leading to the patch moving continuously. "Sliding" refers to taking orthogonal steps in each cycle such that the patch progresses laterally on the physical qubit grid.

in cases with high logical error rates. We performed the benchmarking on a 96 core machine over the course of around three days. We used an internal correlated minimum-weight perfect matching decoder for all shots.

We used a circuit error model with a single parameter p as follows:

- Immediately after each Reset, flip the state of the qubit with probability p. (Apply an X following an RZ, and a Z following an RX)

- Immediately after each CNOT gate, apply uniform 2Q depolarizing noise with prob-

ability p.
(Apply the identity with probability $1-p$, and otherwise apply one of the non-trivial two-qubit Paulis uniformly at random.)

- Immediately before each Measurement, flip the state of the qubit with probability p. (Apply an X before an MZ, and a Z before an MX)

We constructed circuits for four behaviours: Standard, where the patch does not move; Wiggling, where the patch moves back and forth on the spot; Gliding, where the patch continuously moves in the same direction; and Sliding, where it moves in orthogonal directions such that it moves laterally every two cycles. We also investigated other behaviours, such as Twirling (stepping in an orthogonal direction in each subsequent cycle such that we return to the starting position after 4 cycles) and random walks (choosing to move the patch in any of the four diagonal directions or not to step with uniform probability); we found these had the same performance as Gliding and Sliding during initial explorations, but did not benchmark them rigorously.

In the following figures, we include a highlight over values that are any more than 1000 times less likely than the maximum likelihood hypothesis , essentially giving a visual indication of the confidence of each point given the number of samples. Figure 6.10 shows the threshold plots for the four benchmarked behaviours. The threshold for all four cases is just greater than 1% for the considered error model. Figure 6.11 shows the same data rearranged into a "fan" plot, with code distance on the x-axis. Here, horizontal lines correspond to threshold, and lines with negative slope are below threshold where errors are exponentially suppressed with code distance. Ignoring smaller code distances where finite-size effects are not negligible, the slope of this line is the error suppression factor Λ. These plots indicate that the performance loss from using the stepping circuit is minimal, although more extensive analysis is necessary to quantify this in terms of the error suppression or affect on the footprint of good logical qubits.

In summary, we have found that the logical performance of the code is not changed significantly by using stepping cycles. While we did use circuit noise, we used a relatively simple error model. The benchmarking could be improved by compiling the circuit to target a specific hardware gate set and using an error model that reflects realistic error rates for that hardware, as was done in [5].

6.6 Use Cases

Given the results of our benchmarking have demonstrated that we can employ this modified cirucit with minimal cost, we turn to analysing the advantages of this code circuit. Here' we discuss three major advantages, two which improve the situation on the physical qubit level, and one which provides a new tool at the logical qubit level.

6.6.1 Leakage reduction by wiggling

Leakage in a generic effect in qubits with additional energy levels, where physical operations can sometimes move the qubit out of the computational levels and into a "leakage state". In particular, arrays of weakly anharmonic transmons are easily excited into leakage states, and this is a limiting factor in recent cutting edge demonstrations of surface code error correction [4, 36]. Immediately after measurement, a measure qubit is not holding important information, and at this point operations can be applied to unconditionally move the qubit into the ground state, even if it has leaked [1]. For data qubits, using reset to remove leakage is not possible because the qubit is part of the code state at all times. This necessitates more complex operations to remove population from leakage states without disturbing the computational states. Another approach is to exchange the roles of measure and data qubits, typically by adding additional operations

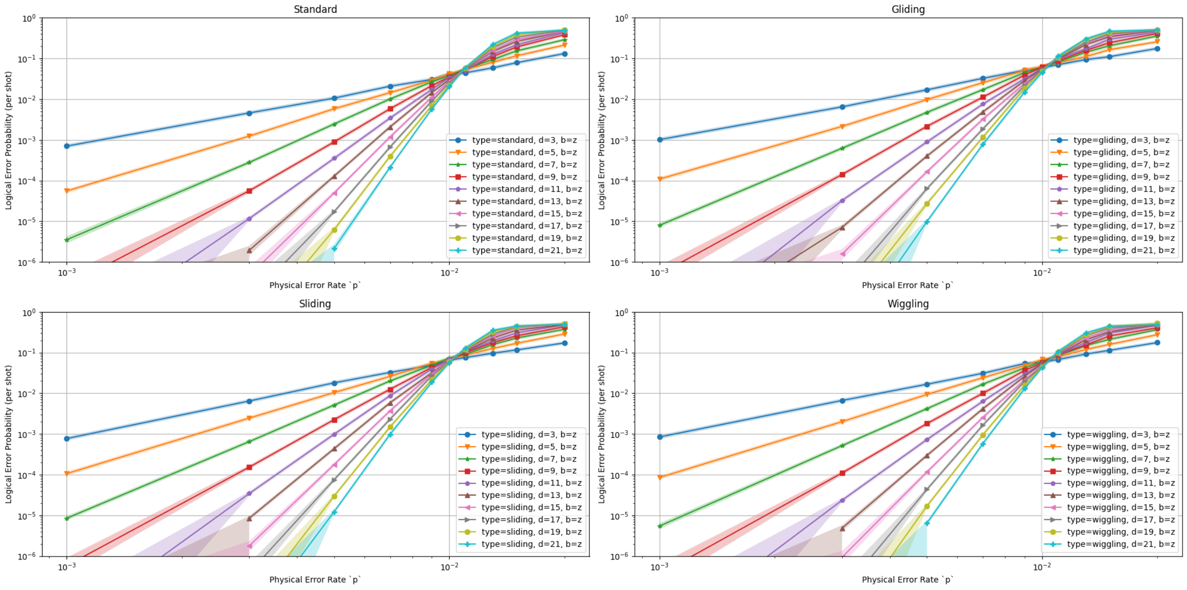


Figure 6.10: **Threshold plots for walking codes.**

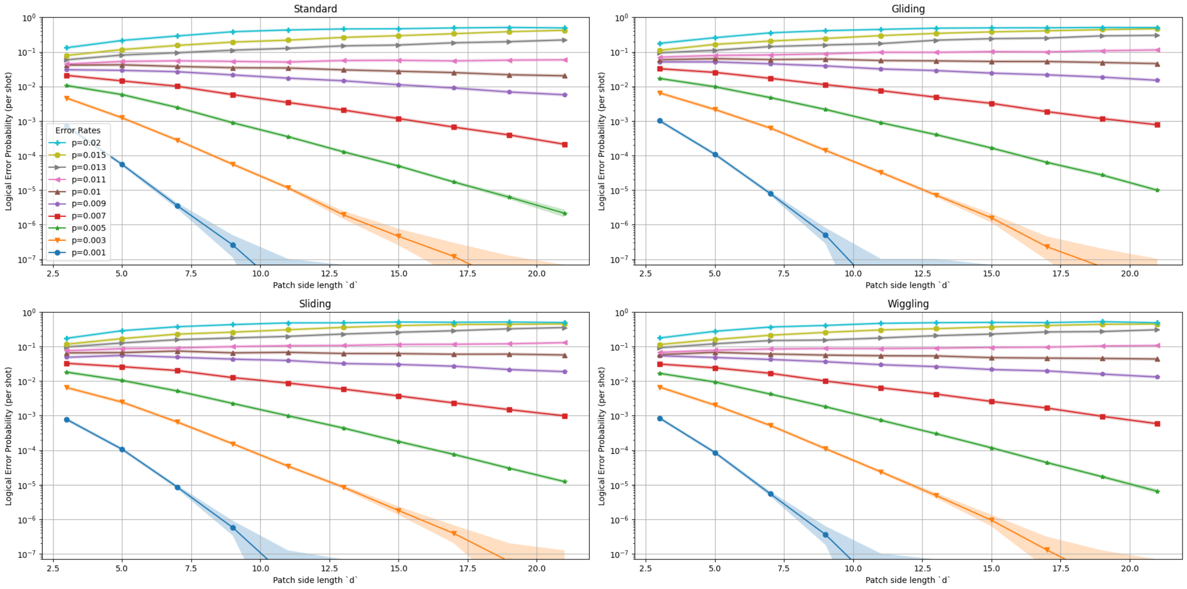


Figure 6.11: **Fan plots for walking codes.**

of the cycle [94, 96]. These operations are generally noisy and negatively impact logical performance. The stepping circuit provides an alternative, where the wiggling behaviour can exchange the roles of measure and data qubits without adding any additional gate layers and without affecting logical performance, and with only a small overhead in total number of qubits. This provides a compelling alternative to currently favoured strategies of handling leakage errors in error correction circuits.

6.6.2 Improving the cost of register shifting

At the level of logical algorithms using lattice surgery, one relatively primitive operation is to 'shift' a densely packed register of patches, each containing a single logical qubit. This operation is natural in places where for instance a stack or queue of qubits is desired. In lattice surgery, qubits are typically moved by expanding and contracting them; that is making the patch larger on the device such that it covers its intended destination, repeatedly measuring to become sufficiently confident that the expansion was performed without error and so ensure fault-tolerance, and then measuring out the qubits not in the destination to complete the movement. For a square patch with side length and therefore code distance d, that same number of cycles are performed between expanding and contracting. Because all of the qubits between origin and destination are used during this operation, the total space-time volume for moving a single qubit by S patch side-lengths $(S + 1)d^3$. To move multiple qubits using this construction, we are forced to move them one at a time into already emptied space, so that we can add the necessary qubits into each patch to move it, as illustrated in Figure 6.12. This makes shifting an entire register of N qubits a relatively expensive operation taking $N(N+S)d^3$ space-time volume. This is particularly costly if the register is large because the volume is quadratic in N.

Using the sliding circuit construction, we can reduce the cost of this primitive operation by moving all the patches in the register simultaneously. For a single qubit, it takes $2d$ cycles of the sliding circuit to move the patch over by one patch side-length d. If we perform this circuit on all patches in the register, it takes a total space-time volume of $2NSd^3$ to achieve a shift by S patch side-lengths. This volume is lower for shifts distances $S < N$ the number of patches in the register. This permits shifting at a cost that grows only linearly with register size, making the use of very large registers as stacks or queues much more appealing on a logical level.

6.7 Outlook

This work reveals significant new freedom in the decomposition of the standard surface code into circuits for physical hardware. Approaching the code from the perspective of detecting regions and the overlapping structure proved important in finding these constructions, and represents an interesting new avenue for understanding fault-tolerant circuits in general. In particular, the freedom to approach the instantaneous stabilizer group of the surface code at locations other than the measurement layer also proved insightful, with the story of combining stabilizers shown in Figure 6.7a illustrating the strengths of this approach. Further work is needed to fully exploit the benefits of this approach.

For this construction itself, benchmarking with realistic noise, and particularly in the presence of leakage, represents important future work. Quantifying the improvement of wiggling over the standard behaviour in the presence of leakage should provide important information to guide predicting future logical performance and efforts to reduce leakage directly.

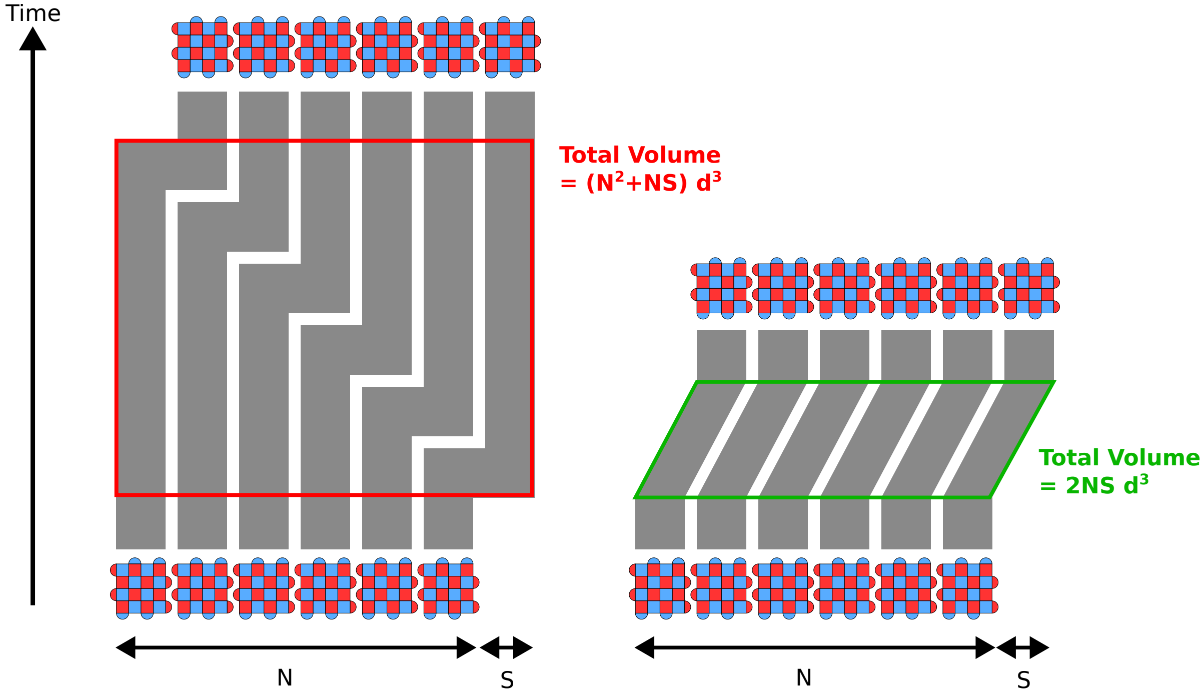


Figure 6.12: **Sliding for simultaneous surface code patch movements.** Considering a line of densely packed surface code patches encoding logical qubits, we show two methods of achieving a 'shift' operation. Each square patch is side-length d, there are N patches in the register, and the operation moves the patches over by S patch side-lengths. On the left, a standard construction in lattice surgery moves each patch one-by-one by expanding toward its final position, taking d cycles in time to maintain fault tolerance, and then contracting it to its final position. Overall, the volume of this construction is $N(N+S)d^3 = (N^2+NS)d^3$ On the right, we use the sliding circuit to have all patches walk laterally at the same time. It takes $2d$ cycles for the patch to move over by 1 patch side length. The construction takes a total volume of $2NSd^3$. Whenever the distance to shift $S < N$ the size of the register, sliding takes a lower total volume.

References

[1] Matt McEwen et al. "Removing leakage-induced correlated errors in superconducting quantum error correction". In: *Nature Communications* 12.1 (Dec. 2021), p. 1761. DOI: `10.1038/s41467-021-21982-y`.

[4] Google Quantum AI et al. "Suppressing quantum errors by scaling a surface code logical qubit". In: (2022). DOI: `10.48550/ARXIV.2207.06431`.

[5] Craig Gidney, Michael Newman, and Matt McEwen. “Benchmarking the Planar Honeycomb Code”. In: *Quantum* 6 (Sept. 21, 2022), p. 813. DOI: `10.22331/q-2022-09-21-813`.

[25] S. B. Bravyi and A. Yu. Kitaev. “Quantum codes on a lattice with boundary”. In: (1998). Publisher: arXiv Version Number: 1. DOI: `10.48550/ARXIV.QUANT-PH/9811052`.

[26] Austin G. Fowler et al. “Surface codes: Towards practical large-scale quantum computation”. In: *Physical Review A* 86.3 (Sept. 18, 2012). Publisher: American Physical Society, p. 032324. DOI: `10.1103/physreva.86.032324`.

[36] Sebastian Krinner et al. “Realizing repeated quantum error correction in a distance-three surface code”. In: *Nature* 605.7911 (May 26, 2022). Publisher: Springer Science and Business Media LLC, pp. 669–674. DOI: `10.1038/s41586-022-04566-8`.

[37] Neereja Sundaresan et al. “Matching and maximum likelihood decoding of a multi-round subsystem quantum error correction experiment”. In: (2022). DOI: `10.48550/ARXIV.2203.07205`.

[65] Barbara M. Terhal. “Quantum error correction for quantum memories”. In: *Reviews of Modern Physics* 87.2 (Apr. 7, 2015). Publisher: American Physical Society, pp. 307–346. DOI: `10.1103/RevModPhys.87.307`.

[94] Austin G. Fowler. “Coping with qubit leakage in topological codes”. In: *Physical Review A* 88.4 (Oct. 8, 2013). Publisher: American Physical Society, p. 042308. DOI: `10.1103/PhysRevA.88.042308`.

[96] Natalie C. Brown and Kenneth R. Brown. “Leakage mitigation for quantum error correction using a mixed qubit scheme”. In: *Physical Review A* 100.3 (Sept. 18,

2019). Publisher: American Physical Society, p. 032325. DOI: `10.1103/PhysRevA.100.032325`.

[113] Matthew B. Hastings and Jeongwan Haah. “Dynamically Generated Logical Qubits”. In: *Quantum* 5 (Oct. 2021). Publisher: Verein zur Forderung des Open Access Publizierens in den Quantenwissenschaften, p. 564. DOI: `10.22331/q-2021-10-19-564`.

[114] Rui Chao et al. “Optimization of the surface code design for Majorana-based qubits”. In: *Quantum* 4 (Oct. 2020). Publisher: Verein zur Forderung des Open Access Publizierens in den Quantenwissenschaften, p. 352. DOI: `10.22331/q-2020-10-28-352`.

[115] C. Ryan-Anderson et al. “Implementing Fault-tolerant Entangling Gates on the Five-qubit Code and the Color Code”. In: (2022). Publisher: arXiv Version Number: 1. DOI: `10.48550/ARXIV.2208.01863`.

Appendix A

Supplementary Information regarding Multilevel Reset

A.1 Reset Gate Parameters

The multi-level reset gate has five main parameters that determine its shape, shown in Fig. 2a in the main text: the swap, hold, and return durations, and two parameters that determine the swap trajectory. We calibrate the swap and hold durations as described in the main text, and we use a minimum return duration imposed by filtering of 2 ns so as to maximize $P_D^{(r)}$.

The adiabatic swap we use follows the quasi-adiabatic approach of Ref. [87] with a modification explained below. In this approach the pulse shape $f_q(t)$ is designed in terms of the control angle θ on the Bloch sphere of states $|01\rangle$ and $|10\rangle$, $\tan(\theta) = 2g/(f_q - f_r)$, $0 < \theta < \pi$, where $f_q(t)$ is the qubit frequency, f_r is the resonator frequency, and g is the qubit-resonator coupling. However, the reset gate has to operate not only for the initial state $|1\rangle$ of the qubit, but also for the states $|2\rangle$ and $|3\rangle$, which have stronger couplings. Moreover, there are three relevant resonance conditions for these cases: $f_r = f_q$, $f_r =$

$f_q+\eta$, and $f_r = f_q+2\eta$, where $\eta \simeq -200\,\mathrm{MHz}$ is the qubit nonlinearity. We therefore use a phenomenological approach and design the pulse $f_q(t)$ as in Ref. [87], but for somewhat different coupling and resonator frequency. We replace g and f_r with free parameters μ and $f_{\rm swap}$ respectively, and optimize experimental performance of the reset gate over these parameters.

For clarity, we now describe the process outlined in Ref. [87] in more detail. The pulse shape $f_q(t)$ is parametrized as $d\tilde{\theta}/dx = (\tilde{\theta}_{\rm fin} - \tilde{\theta}_{\rm in})\sum_{n=1}^{3}\lambda_n[1-\cos(2\pi n x)]$, where $\sum_{n=1}^{3}\lambda_n = 1$, $\tan(\tilde{\theta}) = 2\mu/(f_q - f_{\rm swap})$, and $0 < \tilde{\theta} < \pi$. Here, the initial and final values of $\tilde{\theta}$ are defined as $\tan(\tilde{\theta}_{\rm in}) = 2\mu/(f_{\rm idle} - f_{\rm swap})$ and $\tan(\tilde{\theta}_{\rm fin}) = 2\mu/(f_{\rm hold} - f_{\rm swap})$. We also define a dimensionless natural time x, $0 \leq x \leq 1$, for which the Rabi frequency is constant. This dimensionless time is related to the physical time t as $t = t_{\rm swap}\int_0^x \sin\tilde{\theta}(x')\,dx'/[\int_0^1 \sin\tilde{\theta}(x')\,dx']$. We first calculate $\tilde{\theta}(x)$ analytically, then calculate $t(x)$ numerically, and then use numerical interpolation to find $\tilde{\theta}(x(t))$. Finally, the qubit trajectory is obtained as $f_q(t) = f_{\rm swap} + 2\mu\cot[\tilde{\theta}(t)]$. We use $\lambda_1 = 1.15$, $\lambda_2 = -0.2$, and $\lambda_3 = 0.05$, similar to the values used in Ref. [76]. We set $f_{\rm hold}$ to be 1 GHz below the readout resonator frequency f_r to minimize the hybridization of levels.

Figure A.1 shows the error of the reset gate as a function of the parameter $f_{\rm swap}$ and the swap duration $t_{\rm swap}$ for the qubit initial states (a) $|1\rangle$, (b) $|2\rangle$, and (c) $|3\rangle$. For the initial state $|1\rangle$ and small values of swap duration, we see that performance is optimized near $f_{\rm swap} = f_r$, whereas at higher values of $t_{\rm swap}$ the dependence is obscured by the readout floor. As expected, for the initial states $|2\rangle$ and $|3\rangle$, the optimal value of $f_{\rm swap}$ is higher than f_r. Nevertheless, we set $f_{\rm swap} = f_r$; this gives an acceptable performance for all initial states at sufficiently long $t_{\rm swap}$.

The parameter μ affects the slope of the pulse shape $f_q(t)$ at $f_{\rm swap}$, and therefore the adiabaticity. A larger value of μ (compared with g) increases the slope near $f_{\rm swap}$ but decreases the slope at the sides of the pulse, thus broadening the frequency range around

f_{swap} over which the slope is approximately constant. This results in a larger diabatic error for the initial state $|1\rangle$ (for which $\mu = g$ would be optimum), but decreases the error for the initial states $|2\rangle$ and $|3\rangle$, for which the first resonance occurs at $f_q + \eta = f_r$ and $f_q + 2\eta = f_r$. Figure A.2 shows the error landscape as a function of μ and t_{swap} for the three initial states. As expected, for the initial states $|2\rangle$ and $|3\rangle$, having values of μ larger than $g = 120\,\text{MHz}$ is preferred. As a compromise, we choose $\mu = 150\,\text{MHz}$. This value not only relaxes the frequency selectivity for the crossing point, but also decreases sensitivity to noise or drift in the frequency bias.

A.2 Leakage Error and suppression

We can distinguish between two kinds of error produced by the reset gate. We define the 'computational error' as the probability that the qubit is in the $|1\rangle$ state after reset, and 'leakage error' as the probability that it remains in a higher state ($|2\rangle$ and $|3\rangle$). In the context of error correction, computational error is preferred as the code naturally identifies and corrects for errors within the computational basis.

In Fig. A.3, we show reset performance separated into computational and leakage error. For short swap and hold lengths, we can see that the ratio of leakage to computational error depends on the initial state, with higher states producing more leakage error as expected. We also see a higher rate of reduction for leakage error than for computational error with hold time, reflecting the higher rate of energy relaxation from higher states. At swap and hold lengths long enough to reach the readout floor. We see that the leakage error is around 10x lower than the computational error for all states, indicating that the dominant error type is computational error.

As in Fig. 3 in the main article, we can distinguish these two error types using a readout optimized for detecting higher states. A representative example of such a readout

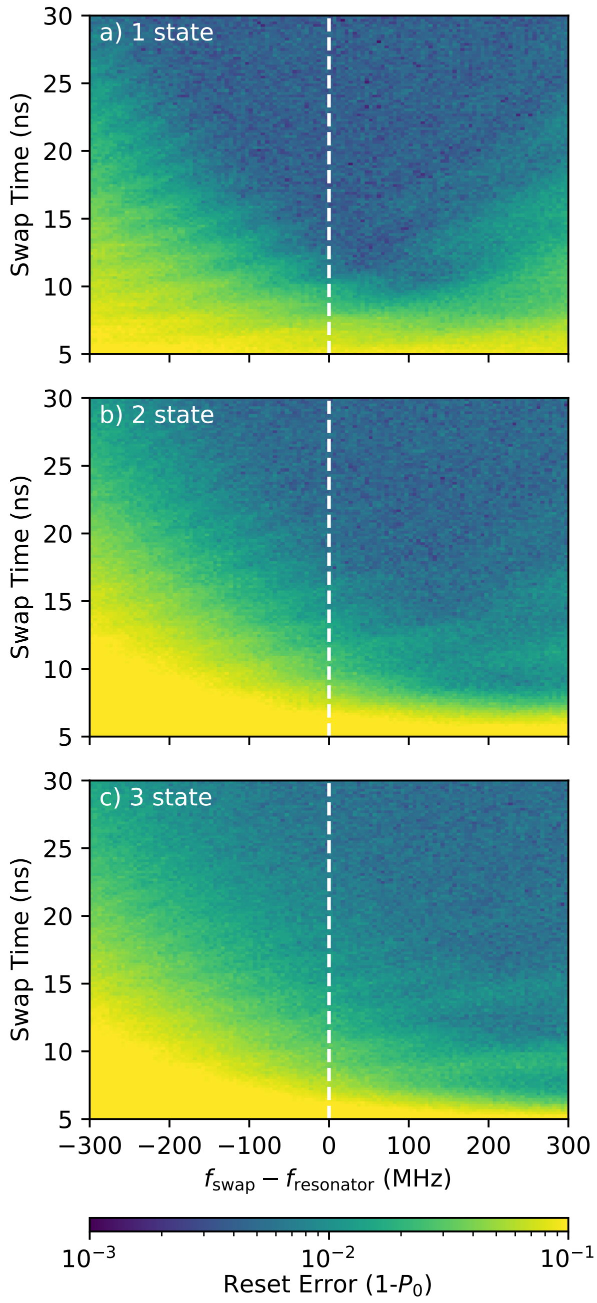


Figure A.1: **Dependence on swap frequency.** Reset performance when applied to a qubit initialized in $|1\rangle$ (a), $|2\rangle$ (b) and $|3\rangle$ (c) versus $f_{\text{swap}} - f_{\text{resonator}}$. For short swap lengths and on input $|1\rangle$, performance is approximately optimal for $f_{\text{swap}} = f_{\text{resonator}}$ (dashed line). Reset of higher states also involve transitions when the qubit is above the readout resonator due to the negative nonlinearity, producing distinct landscapes. At long swap lengths, this dependence is obscured by the readout floor, but degraded performance on $|2\rangle$ and $|3\rangle$ states is visible for $f_{\text{swap}} < f_{\text{resonator}}$.

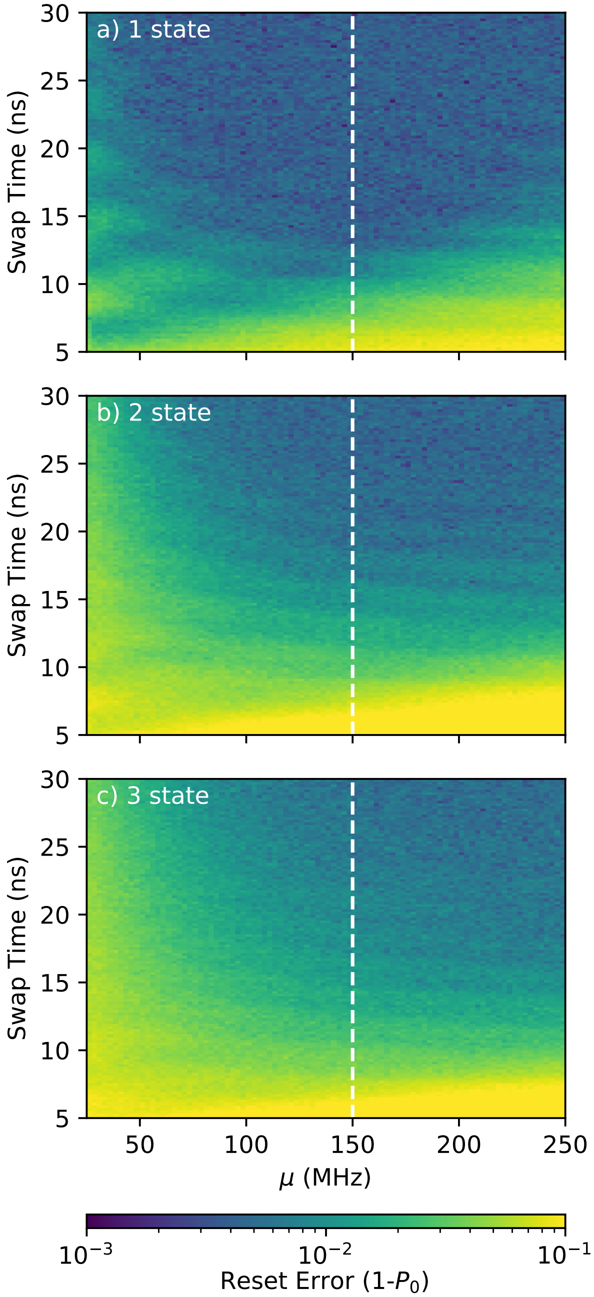


Figure A.2: **Dependence on adiabatic slope parameter.** Reset performance when applied to qubits in states $|1\rangle$ (a), $|2\rangle$ (b) and $|3\rangle$ (c) versus the adiabatic slope parameter μ. For short swap lengths, performance by different input states show different profiles depending on the number of transitions involved. We choose a value of μ=150 MHz as a compromise between these three cases (dashed lines). At long swap lengths, this dependence is mostly obscured by the readout floor, but degraded performance on $|2\rangle$ and $|3\rangle$ states is visible at smaller values of μ.

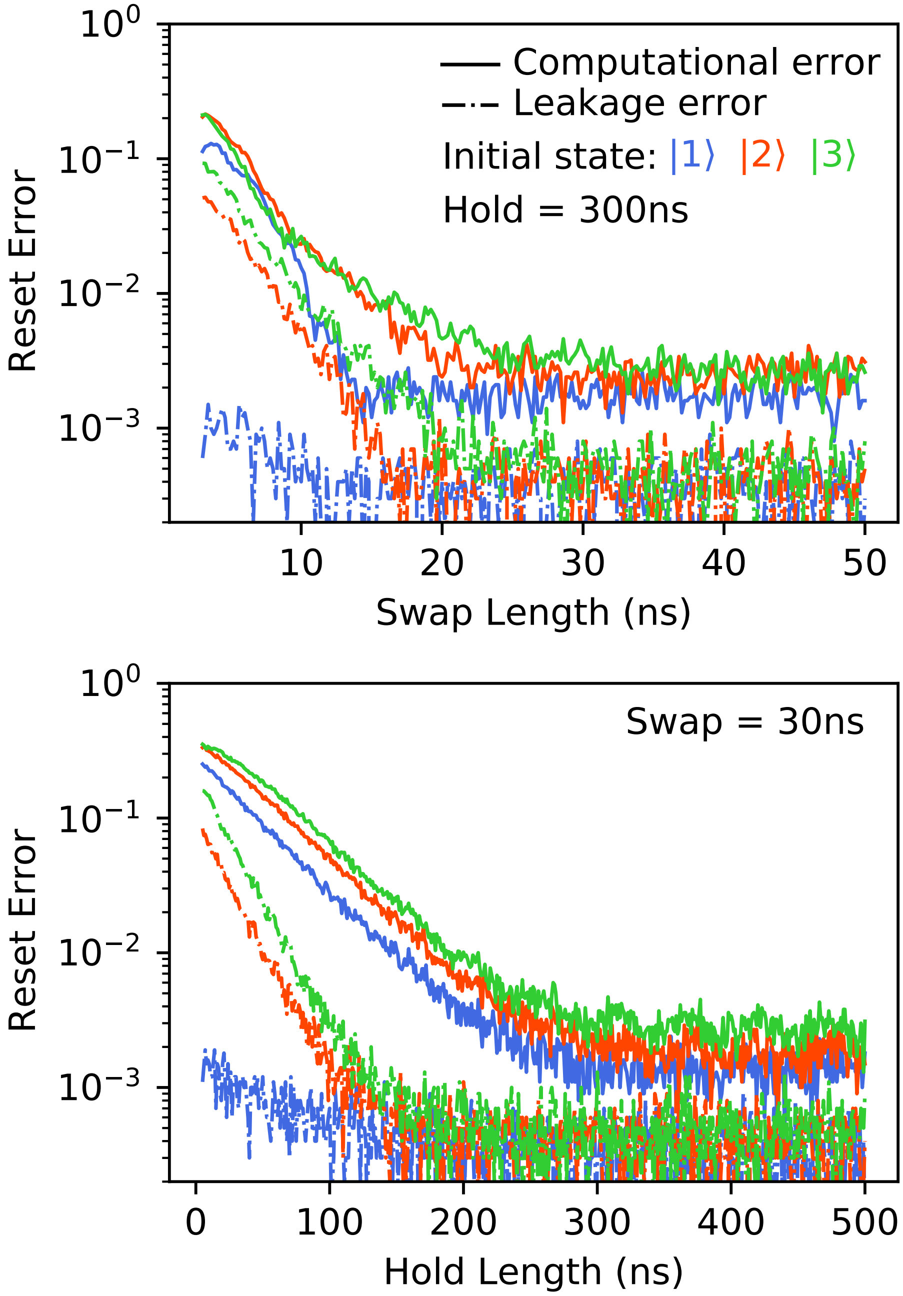


Figure A.3: **Computational vs. Leakage error.** Reset error separated into computational error (P_1, solid lines) and leakage error ($P_2 + P_3$, dash-dotted lines) for the reset gate applied to the first three excited states. We see that computational error accounts for the majority of error in all cases.

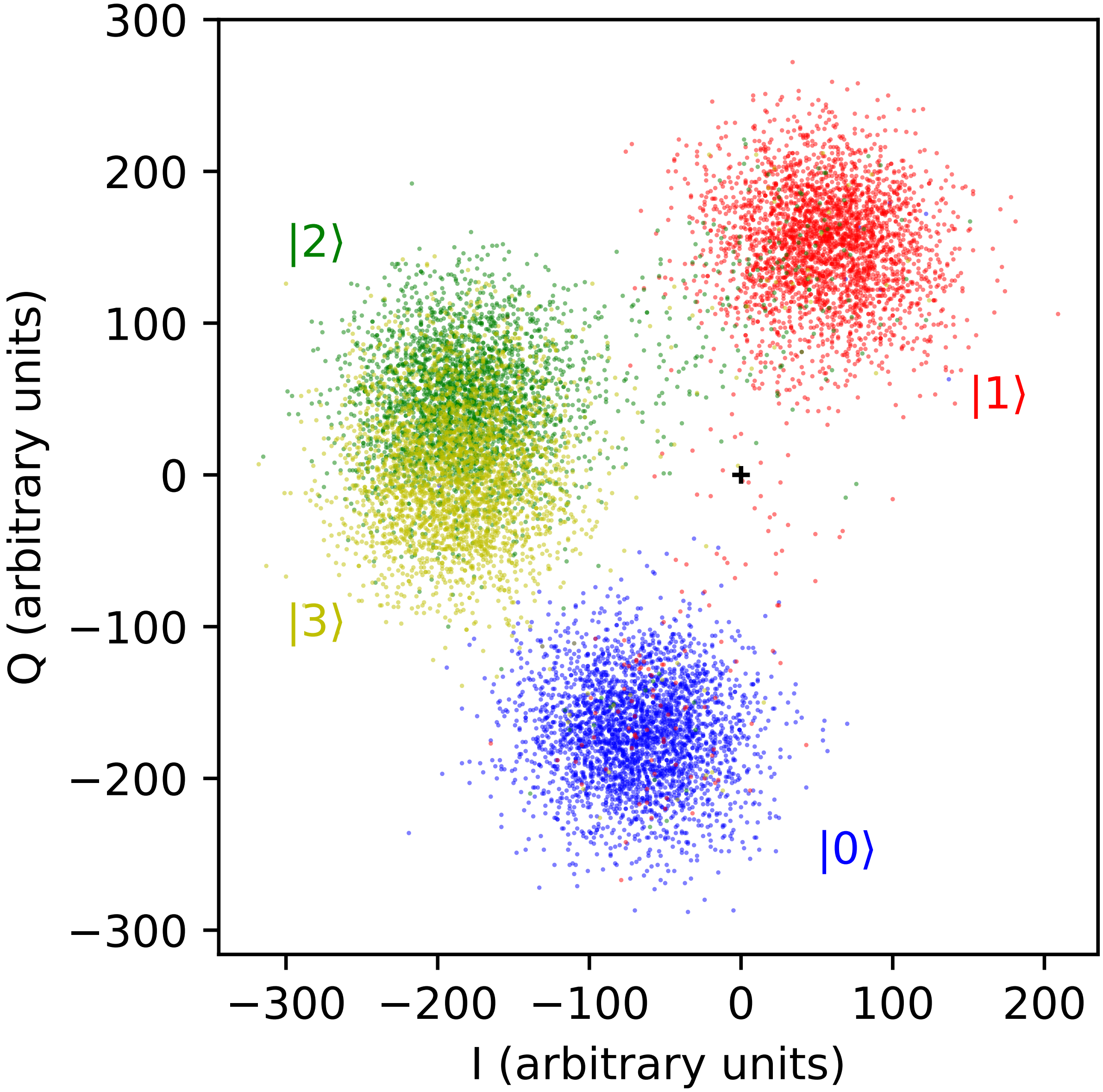


Figure A.4: **Readout for distinguishing leakage.** Demodulated in-phase (I) and quadrature (Q) components measured using a readout optimized for distinguishing computational states and leakage states. The origin is marked by a black plus. The qubit is repeatedly prepared in the $|0\rangle$, $|1\rangle$, $|2\rangle$ or $|3\rangle$ states and the raw readout signal is recorded. This illustrates our ability to distinguish the two computational states from each other and from higher level states with high fidelity.

result is shown in Fig. A.4. The qubit is repeatedly prepared in $|0\rangle$, $|1\rangle$, $|2\rangle$ or $|3\rangle$, and the complex readout signal is measured and demodulated. Each shot is plotted as a single point, colored by the prepared state, allowing us to evaluate the readout fidelity for various states and to calibrate the discrimination of states. This readout was optimized to distinguish the two computational states from leakage states with high fidelity, but does not attempt to distinguish the leakage states $|2\rangle$ and $|3\rangle$ from each other.

For such a readout, we find the readout floor in Fig. 2 in the main article by heralding; performing two sequential measurements on the qubit, postselecting on $|0\rangle$ on the first measurement and calculating the fidelity of measuring $|0\rangle$ on the second measurement. We find that the total infidelity is around $\sim 0.2\%$ as shown in Fig. 2. We can further break this down into $\sim 0.18\%$ computational and $\sim 0.02\%$ leakage infidelity respectively, which are compatible with the values found at saturation in Fig A.3. This also illuminates our ability to measure values for 2-state population significantly below the readout visibility floor; in Fig. 3 and Fig. A.3, we show measured values of leakage population reaching down to the floor for leakage error at $\sim 0.02\%$, below the $\sim 0.2\%$ visibility floor shown in Fig. 3.

A.3 Leakage accumulation during the bit-flip code

In Fig. 3 of the main text, we measure the growth of leakage population during the bit-flip code using a readout similar to that shown in Fig. A.4. In Fig. A.5, we show the leakage population for each qubit over the length of the code, as well as the average that was included in Fig. 3. We note that there is significant inter-qubit variation, which we attribute to the optimization procedures we employ [61]. When applying reset, we see that all measure qubits display leakage populations around the readout floor for leakage error indicated in Fig. A.3 for all code lengths. We note that the reset protocol is capable of strongly suppressing even anomalously high rates of leakage on the measure qubits. We find the same qualitative behaviour over all qubits: The leakage populations exponentially approach saturation values of similar order over the course of the code, hence we focus our analysis on the average.

We fit the average leakage population to an exponential to extract parameters for a rate equation [75].

$$P_{|2\rangle}(k) = p_\infty \left(1 - e^{-\Gamma k}\right) + p_0 e^{-\Gamma k} \tag{A.1}$$

$$\Gamma = \gamma_\uparrow + \gamma_\downarrow \qquad p_\infty = \frac{\gamma_\uparrow}{\Gamma} \tag{A.2}$$

The rates are displayed in Table A.1, showing an increase in effective leakage decay rate when reset is applied. As seen in Fig. 3 and in Fig. A.5, applying reset to the measure qubits breaks the established behaviour for growth of leakage. In order to estimate $\gamma_\downarrow$, we therefore assume a value of $\gamma_\uparrow$ equal to the case of no reset, and a value of p_∞ given by the average error for measure qubits across all rounds.

Table A.1: Effective leakage growth and decay rate per stabilizer round, using Eq. A.1. For the case of measure qubits with reset, $\gamma_\downarrow$ was estimated from the value of p_∞ when assuming $\gamma_\uparrow$ equal to the case of no reset (asterisk).

		$\gamma_\uparrow$	$\gamma_\downarrow$	p_∞
No Reset	Data	0.09%	9.1%	0.97%
	Measure	0.11%	8.1%	1.30%
With Reset	Data	0.11%	22.1%	0.50%
	Measure	0.11%*	328%*	0.03%*

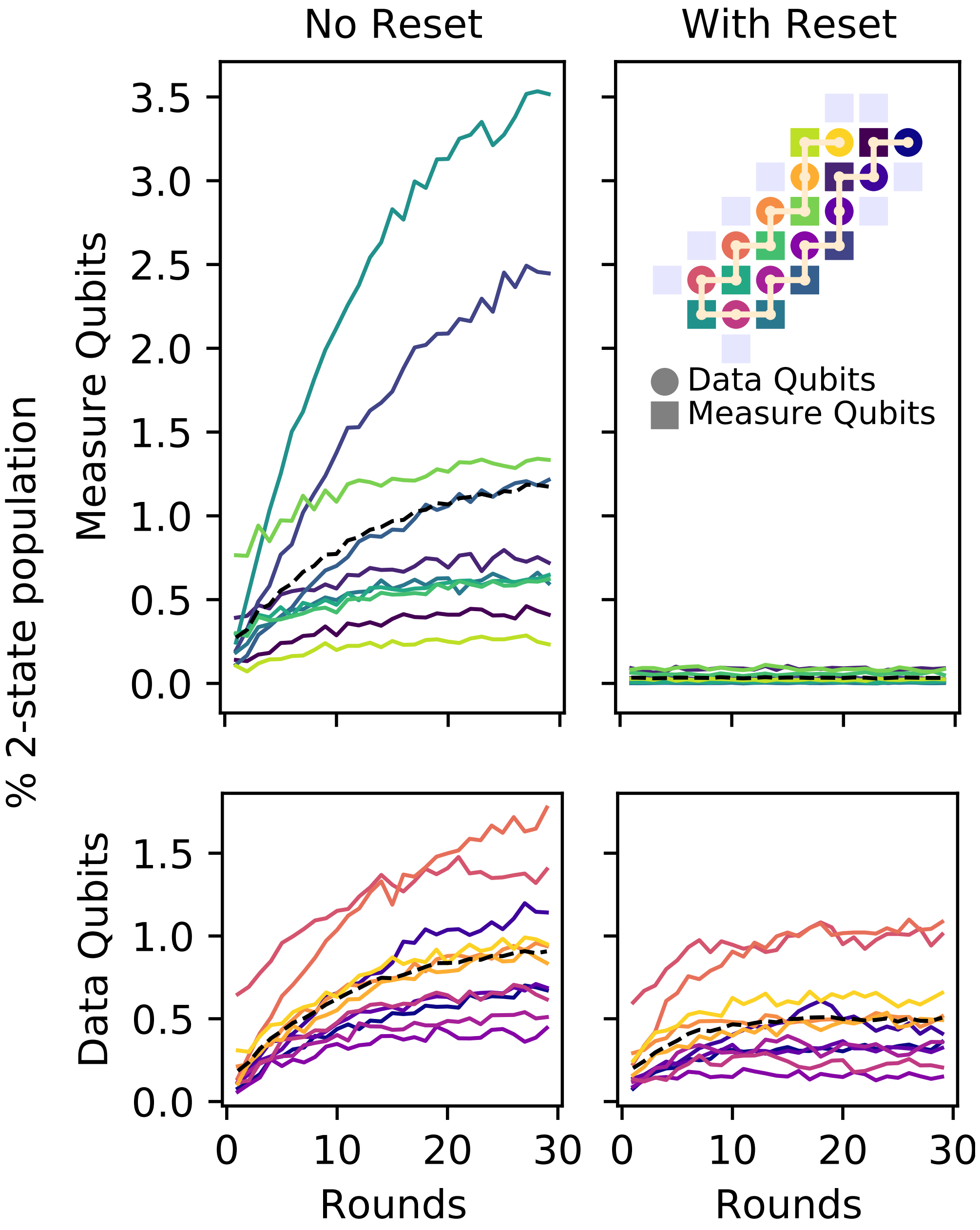


Figure A.5: **Leakage Populations during bit-flip code.** The growth in $|2\rangle$ population vs. stabilizer code length for each qubit. As in Fig. 3, the circuit is run for a number of rounds and terminated with a readout sensitive to $|2\rangle$ population. The average is included as a dashed black line. We note significant inter-qubit variation produced by calibrations, but the same qualitative behaviour across all qubits. The inset indicates the location of each qubit on the Sycamore device.

A.4 The $\mathrm{p_{ij}}$-matrix

We first describe the model of detection events that is used to obtain Eq. 1 of the main text. We consider an error graph where each node is associated with a measure qubit and a round of the bit-flip code [51], and the edges are between *all* pairs of nodes. The state of a node corresponds to whether an error is detected and takes values $x_i = 0$ for no detected error or $x_i = 1$ if an error was detected. The edges represent possible errors, whose occurrence flips the states of its two nodes, $x_i \to 1 - x_i$ and $x_j \to 1 - x_j$. We assume that, in each realization of the bit-flip code, an error on each edge ij occurs independently and according to a fixed probability p_{ij}, where i and j indicate the nodes connected by the error edge. The detection event at a node i will be registered ($x_i = 1$) in a realization if an odd number of errors on edges connected to the node i have occurred in that realization.

Due to the statistical independence of the errors, the statistics of nodes i and j detecting errors can be understood considering three independent processes: (1) the occurrence of an error on edge ij that flips both x_i and x_j; (2) the occurrence of an odd number of errors on edges ik with $k \neq j$, flipping x_i but not x_j; (3) the occurrence of an odd number of errors on edges kj with $k \neq i$, flipping x_j but not x_i. We can then express the averages $\langle x_i \rangle$, $\langle x_j \rangle$, and $\langle x_i x_j \rangle$ (where $\langle \cdot \rangle$ indicates averaging over realizations) in terms of the probabilities of these three processes: p_{ij}, p_i and p_j, respectively. After solving this nonlinear algebraic system of three equations, we obtain the error edge probability p_{ij} that is given in Eq. 1 of the main text.

A.5 The checkerboard pattern in the $\mathrm{p_{ij}}$-matrix

There is a clear checkerboard pattern visible in Fig. 5c in the main article, in which the values of the correlation matrix p_{ij} for measure qubit 6 are larger for correlations spanning an odd number of rounds. For edges spanning an even number of rounds, the correlations are smaller and can even be negative. A similar but less pronounced checkerboard pattern can be seen in Fig. 5e for the cross-correlation between measure qubits 5 and 6. In fact, (b) and (c) display similar patterns, but these are visually masked by the presence of significant leakage-induced correlations. Both edges shown in Fig. 5a span odd numbers of rounds and show values larger than neighbouring values in the p_{ij} matrix.

This checkerboard pattern is caused by correlations between energy relaxations on the same data qubit. The mechanism of the correlation is illustrated in Fig. A.6. An energy relaxation event on a data qubit, $|1\rangle \rightarrow |0\rangle$, produces a pair of detection events on the neighboring measure qubits (red circles in Fig. A.6). Subsequent X-gates applied to the data qubit each round (see Fig. 1b in the main article) alternate the qubit state: $|0\rangle \rightarrow |1\rangle \rightarrow |0\rangle \rightarrow |1\rangle \rightarrow \ldots$. As a result, the qubit can relax again 1, 3, 5, ... rounds later, while relaxation after 2, 4, 6, ... rounds is unlikely. This creates a positive correlation between the errors separated by an odd number of rounds and negative correlation for separation by an even number of rounds, producing the checkerboard pattern. The correlations gradually decay with increasing separation. The checkerboard pattern is more pronounced in Fig. 5c as a single measure qubit is affected by both neighbouring data qubits, while in Fig. 5e the checkerboard pattern is caused by only the one data qubit between the measure qubits 5 and 6.

A.6 Statistics and postselection in the bit-flip code

When benchmarking performance in the bit-flip code, we average over a large number of realizations, including over randomly chosen initial states for the data qubits. For the leakage populations shown in Fig. 3 of the main text, we chose 20 random initial bitstrings for the data qubits, and repeated the experiment 5000 times for each bitstring. For data shown in Figs. 4, 5 and 6, we chose 40 random bitstrings and repeated the experiment 1000 times for each bitstring for 40 000 total realizations. However, the probabilities of logical errors are smaller at low numbers of rounds, requiring additional averaging to reduce statistical error. For runs with 10 or fewer rounds, we therefore chose 100 random initial bitstrings and took 10 000 repetitions at each bitstring, for 1 000 000 total realizations.

Over these large numbers of runs, we see a small number of short events where the detection fractions are significantly elevated compared to the average. We are investigating these effects [In preparation, Google AI Quantum and Collaborators][In preparation, McEwen et al.]. These events are not representative of the normal functioning of the

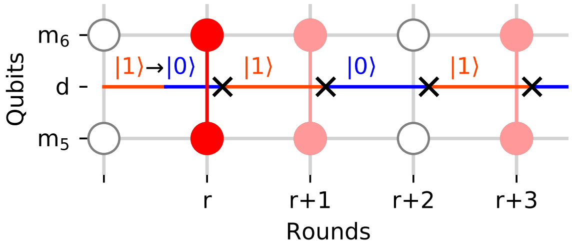


Figure A.6: **Odd-even periodicity of energy relaxation events.** The state of a data qubit d is flipped each round by an X gate (black). An energy relaxation event $|1\rangle \rightarrow |0\rangle$ in d in round r produces detection events (red circles) in the neighboring measure qubits m_5 and m_6. As the next energy relaxation event can occur only when d is in $|1\rangle$, future energy relaxation errors will be preferentially separated by an odd number of rounds from the initial event (pink circles), producing an alternating pattern of correlations.

device and so we postselect them out using the following procedure. We calculate the logical error for each time-ordered realization and then calculate a moving average of the logical error over 30 realisations. The this average is typically below 3%, but during events the moving average can reach 50%. We choose a threshold of 25% to identify the start and end of an event. We remove 500 realizations before the start and 500 realizations after the end of each event, typically removing around 1200 realizations in total. This procedure removes approximately 0.8% of the data.

A.7 Subsampling for analysis of scaling performance

A central assumption in quantum error correction is that the logical performance of a code should scale exponentially with number of qubits. To analyse this scaling, instead of running multiple experiments at different numbers of qubits, we use subsampling to extract performance at lower orders from from experiments consisting of larger number of qubits, as in Ref. [51].

The 21-qubit code is 5th order fault tolerant, meaning it can correct up to 5 simultaneous X errors. This code has three possible 17-qubit subsets which are each 4th order fault tolerant. For each of these 17-qubit subsets, we can discard the appropriate data from a run of the 21-qubit code and infer the performance at 4th-order fault tolerance. Averaging over all subsets at each lower order gives more accurate estimation of the scaling performance than running separate experiments at each lower order. It naturally provides a large number of instances at low order, avoids introducing variation due to calibration drift over time and is significantly less experimentally taxing. This technique is discussed in greater detail in Section IV of the Supplementary Materials in Ref. [51].

References

[51] J. Kelly et al. “State preservation by repetitive error detection in a superconducting quantum circuit”. In: *Nature* 519.7541 (Mar. 5, 2015). Publisher: Nature Publishing Group, pp. 66–69. DOI: `10.1038/nature14270`.

[61] Paul V. Klimov et al. “The Snake Optimizer for Learning Quantum Processor Control Parameters”. In: (2020). _eprint: 2006.04594. DOI: `https://doi.org/10.48550/arXiv.2006.04594`.

[75] Zijun Chen et al. “Measuring and Suppressing Quantum State Leakage in a Superconducting Qubit”. In: *Physical Review Letters* 116.2 (Jan. 13, 2016). Publisher: American Physical Society, p. 020501. DOI: `10.1103/PhysRevLett.116.020501`.

[76] R. Barends et al. “Superconducting quantum circuits at the surface code threshold for fault tolerance”. In: *Nature* 508.7497 (Apr. 2014). Publisher: Nature Publishing Group, pp. 500–503. DOI: `https://doi.org/10.1038/nature13171`.

[87] John M. Martinis and Michael R. Geller. “Fast adiabatic qubit gates using only \sigma z control”. In: *Physical Review A* 90.2 (Aug. 8, 2014), p. 022307. DOI: `10.1103/PhysRevA.90.022307`.

Appendix B

Supplementary Information regarding Complete Leakage Removal

B.1 Effects of leakage on diabatic CZ gate

During the diabatic CZ gate, additional levels are placed on resonance and contribute to the leakage transport phenomenon depicted in the main text Figure 2.

The resonance $|12\rangle \leftrightarrow |30\rangle$ allows a 2-photon transition mediated by $|21\rangle$, which is detuned by around the nonlinearity η. If g is the induced coupling between $|11\rangle$ and $|20\rangle$, then the effective couplings are:

$$
\begin{aligned}
g_{|12\rangle\leftrightarrow|21\rangle} &= 2g \\
g_{|30\rangle\leftrightarrow|21\rangle} &= \sqrt{3}g \\
g_{\text{eff}} = g_{|12\rangle\leftrightarrow|30\rangle} &= -g_{|12\rangle\leftrightarrow|21\rangle} \times g_{|30\rangle\leftrightarrow|21\rangle}/\eta
\end{aligned}
$$

Assuming a nonlinearity $\eta/2\pi = 200$ MHz and a simplistic step shape in the induced coupling g reaching $g/2\pi = 20$ MHz for $t = 10$ ns, we estimate $g_{\text{eff}}/2\pi \approx 7$ MHz and a transport probability $p = \sin^2(g_{\text{eff}}t) \approx 18\%$, which matches quite well to the measured transport. An improved model would attempt to account for the true pulse shape, variations in the qubit nonlinearity and variations from perfect resonance in $|11\rangle$ and $|20\rangle$.

To measure the leakage transport in the diabatic CZ gate, we calibrate a readout pulse capable of distinguishing all of the four lowest qubit energy levels, as shown in Figure B.1a. When we encounter $|4\rangle$ during measurement using this readout pulse, Although not shown in these readout clouds, when we measure $|4\rangle$ on either in both cases population in $|4\rangle$ is labeled as $|3\rangle$. The baseline experiment consists of preparing a given two-qubit state using microwave drives and then performing simultaneous readout of the qubits. The "Baseline" matrix of Figure B.1b shows the results of this experiment, illustrating that performing readout simultaneously does not impact the high distinguishability between all multi-qubit states. We can also see that the majority of the error in this simultaneous readout is due to T_1 decay during the readout process. These decay channels reduce the populations on the main diagonal by a few percent, and become more prominent for higher levels. The "With CZ gate" matrix of Figure B.1b shows the same experiment with a CZ gate inserted between state preparation and measurement. We then see new off-diagonal processes corresponding to the leakage transport. We subtract the "Baseline" matrix from the "With CZ gate" matrix to produce the matrix shown in Figure 2b of the main text.

To measure the spurious phase that higher leaked qubits impact through the CZ gate, we perform the experiment detailed in Figure B.2. The higher energy qubit in the pair is prepared in each of $|0\rangle$, $|1\rangle$, and $|2\rangle$, while a Ramsey experiment is performed on the lower frequency qubit with an interleaved CZ gate. We vary the tomography phase ϕ_T of the second pulse relative to the first in the Ramsey experiment to record data as shown in Figure B.2 for each input state. We fit a sinusoid and extract the phase offset. We performed this experiment on 20 pairs of qubits, and show the cumulative histogram of the extracted phases in the main text Figure 2e.

B.2 Leakage Removal Strategy Details

We studied three leakage removal strategies; *No reset*, *MLR* and *DQLR*. We now describe them in greater detail below.

For *No reset*, we add no additional operations at the end of each cycle. Because this prepares the qubit for the next cycle in whatever state was measured rather than deterministically in $|0\rangle$, this also requires the redefinition of the detectors in the surface and repetition code circuits: rather than comparing time-neighbouring measurements, we compare time-next-neighbouring measurements on the same measure qubit to detect errors. While this redefinition will have only an insignificant impact on code performance, especially when compared to the studied effects of leakage, and so we neglect it from our analysis.

For *MLR*, we add the multi-level reset operation introduced in [1] on the measure qubits at the end of each cycle. Additional pulse shaping on the diabatic return and calibration improvements allows us to reduce gate times to 160 ns without impacting performance.

For *DQLR*, we perform a 'LeakageISWAP' between pairs of measure and data qubits, followed by fast reset of the measure qubits. The 'LeakageISWAP' gate is similar to the diabatic CZ used in the surface code cycle, but rather than the $|11\rangle \leftrightarrow |20\rangle$ rotation angle being calibrated to 2π, it is calibrated to π. This executes an ISWAP gate in the $|11\rangle - |20\rangle$ subspace. We use the same gate time as our calibrated CZs, around 28 ns. When the measure qubit is lower in frequency than the data qubit and has been deterministically prepared in $|0\rangle$ by the MLR operation, the effect of the LeakageISWAP gate is to transfer any population in $|2\rangle$ on the data qubit to $|1\rangle$ on both data and measure qubits. When leakage is removed, the data qubit is left in $|1\rangle$, essentially converting leakage into Pauli error. The $|1\rangle$ on the measure qubit must then be removed, but this can be done with a reset. Compared to the earlier multi-level reset gate, this reset can be optimised to target only the $|1\rangle$ state, allowing us to neglect dynamics associated with the nonlinearity and shorten the gate to 25 ns. We note that this DQLR procedure relies on the high fidelity of the MLR operation; any reset error leaving $|1\rangle$ on the measure qubits will be converted into leakage on the data qubit by the LeakageISWAP. Our results show that this error path is sufficiently low probability so as not to increase the leakage population on the data qubits.

Ideally, DQLR should not induce additional errors on the data qubit. However, the non-zero time to execute the DQLR procedure introduces incoherent errors caused by relaxation, as well as coherent errors from miscalibration. We evaluate the impact of the DQLR procedure on the data qubit state using cross-entropy benchmarking (XEB). The inset of Figure B.3 shows the experimental circuit used to evaluate XEB error. The upper and lower qubits mimic the role of data and measure qubits, respectively. The section of the circuit within the parentheses is repeated a variable number of times, and the final state of the upper qubit is measured. The cross-entropy of the measured and expected distributions of states is calculated as a function of repetitions, and then the XEB error per repetition is extracted. In a given repetition, a random unitary $\mathbf{U}$ is executed on the upper qubit, followed by a reset operation $\mathbf{R}$. In the case of DQLR, $\mathbf{R}$ is substituted with the DQLR procedure. We compare this to the Idle case, where $\mathbf{R}$ is replaced with waiting for the duration of the DQLR procedure. We carry this measurement out over the 9 pairs of data and measure qubits corresponding to the pairings used in the distance-3 surface code experiment. By comparing the resulting distribution of XEB error per cycle for DQLR and Idle, we conclude that DQLR does not induce significantly more errors than idling for the equivalent duration. Furthermore, this mean error rate of $<0.25\%$ per cycle is low enough that the operation is suitable to be added to a sensitive QEC circuit such as the surface code. We note that leakage is still a relevant consideration in this XEB experiment, and is captured by XEB error per cycle as an incoherent error. Thus, it is possible that the leakage removal properties of DQLR result in underreported XEB error per cycle when compared to Idle, which allows for leakage to accumulate over the course of the XEB circuit.

B.3 Effect of reset strategies on leakage dynamics

As we have demonstrated in the main text, leakage transport can move leakage population from qubit to qubit in a structured circuit such as the surface code. Once a data qubit is leaked, leakage removal techniques must be employed or the leakage population may remain for many QEC cycles and cause additional leakage and leakage-induced error through leakage transport. In Figure B.4, we evaluate the dynamics of leakage population in a surface code under the three leakage removal techniques discussed in this work. We fully inject leakage at the beginning of the first cycle by performing a $|1\rangle \rightarrow |2\rangle$ rotation on the central data qubit to obtain near-50% $|2\rangle$ population. We measure excess leakage population, which is defined as the leakage population without injection subtracted from the leakage population with injection. This allows us to separate the contribution of intrinsic heating to leakage population dynamics from the injected leakage population.

For *No reset*, leakage transport allows for leakage to move freely throughout the qubits surrounding the central data qubit, eventually resulting in measurable excess leakage population in nearly all 17 qubits involved in the distance-3 surface code. The lack of leakage removal procedures on either the measure qubits or data qubits leaves T_1 energy relaxation of $|2\rangle$ as the primary mechanism for leakage population dissipation. As we showed in Figure 1c of the main text, T_1 of $|2\rangle$ can be longer than 10 surface code cycles, and this number is expected to increase as qubit coherence improves and surface code cycle durations shorten. Hence, relying on energy relaxation is not a viable strategy for leakage removal in QEC. This is partially addressed by *MLR* by applying a multi-level reset gate on all measure qubits at the end of each cycle. This operation effectively removes leakage by swapping the leaked qubit's excitations into its readout resonator and allowing for the resonator's excitations to decay. In Figure B.4, the effect of this operation appears as reduced excess leakage populations on all measure qubits at the end of every cycle. However, leakage transport within a QEC cycle can still move leakage population beyond nearest-neighbor qubits. This is readily observed in the excess leakage population of the data qubit two sites away from the central data qubit, which exhibits increasing population even though the measure qubit between the two data qubits has its leakage population removed by the MLR operation at the end of every cycle.

DQLR suppresses this effect where leakage can hop between data qubits by directly removing a large fraction of all the data qubits' leakage populations at the end of each cycle. In particular, the central data qubit has its excess leakage population reduced to <1% after the first cycle. Similarly, other data qubits that previously leaked due to leakage transport have their excess leakage populations reduced close to the measurement floor. The shortened lifetime of leakage on all qubits is clearly seen for *DQLR* after two QEC cycles, where excess leakage population over all qubits is <0.1%.

B.4 Effect of reset strategies on QEC error detection

In Figure B.5, we compare the time dynamics of the bit-flip code under the three different reset strategies presented in the main text. We execute a distance-21 bit-flip code over 60 cycles, as described by the circuit in Figure 4d of the main text. For MLR and DQLR, we inject both 0% and 1% leakage population in each cycle, whereas for No reset we do not inject leakage. In the first few cycles of code execution, differences in the logical performance of the bit-flip code between the various leakage removal strategies and injection populations are difficult to distinguish – we attribute this to time-boundary effects where physical errors have not sufficiently accumulated to manifest as a logical error, and statistical limitations where the logical error probability is much smaller than what is resolvable by the number of trials. However, the logical performance for the three leakage removal strategies begin to diverge after about 10 cycles. The accumulation of leakage on measure and data qubits causes a rapid rise of logical error probability for No reset, where it exceeds 1% error by 25 cycles. This is in contrast to MLR and DQLR without leakage injection, which continue to have <0.3% logical error probability through 60 cycles; DQLR has the best performance with about 0.1% logical error probability at 60 cycles.

Turning to the cases where we inject 1% leakage population per cycle for MLR and DQLR, we observe a notable distinction in logical error probability scaling over cycle number. With MLR, the logical error probability rises to about 1% by 30 cycles, with a similar qualitative behavior to No reset without leakage injection. However, when we use DQLR, the code sustains logical error probabilities <0.5% over 60 cycles. This is a signature that QEC scaling in time can be more easily achieved when using DQLR.

In Figure B.6, we present the average weight-2 stabilizer detection probabilities in addition to the average weight-4 stabilizer values already shown in Figure 4a of the main text. Additionally, we show the detection probabilities associated with the individual stabilizers. We can draw parallel conclusions for the weight-2 stabilizer behavior as we did for weight-4 stabilizers in the main text. For No reset, the average weight-2 stabilizer detection fractions rise and do not stabilize over the course of 30 surface code cycles, and even exhibit damped oscillations at early cycles. When using MLR, the detection probability stability improves significantly and the average probability only rises by about 1% over 10 cycles before flattening. However, the best performance and stability is achieved with DQLR, where the average weight-2 stabilizer detection probability remains at 11% over the course of the entire 30-cycle experiment.

B.5 Fitting techniques for logical error probability

We employ phenomenological models to fit experimental and simulated data. The models are implemented on the logical error per cycle ε of the QEC code. In experimental and simulated data, we measure logical error probability p_L after n QEC cycles. The conversion between ε and p_L after n cycles is given by

$$\begin{aligned}\varepsilon &= \frac{1-(1-2p_L)^{\frac{1}{n}}}{2},\\ p_L &= \frac{1-(1-2\varepsilon)^n}{2}.\end{aligned}$$

With respect to injected error population P, ε can be modeled as a power law with an offset,

$$\varepsilon(P) = a(P+P_0)^b,$$

where a, b, P_0 are phenomenological free parameters. We do not ascribe a physical meaning to P_0 even though it may appear to be "intrinsic" error at $P = 0$. In order to model $1/\Lambda_{5/7}$ as a function of injected leakage population P_L in Figure 5c of the main text, we take the ratio of the distance-5 and distance-7 logical error per cycle,

$$\Lambda_{5/7}\left(P_L\right) = \frac{\varepsilon_7\left(P_L\right)}{\varepsilon_5\left(P_L\right)}.$$

For p_L with respect to cycle n of a bit-flip code operated well below threshold and in the presence of leakage dynamics (Figure B.5), we use a Gompertz model to describe ε,

$$\varepsilon\left(n\right) = a \exp\left(-b \exp\left(-cn\right)\right),$$

where a, b, and c are phenomenological free parameters. A Gompertz model can partly capture the transient dynamics of ε at small n, where time-boundary effects and still-increasing leakage populations make ε highly dependent on n. At large n, ε tends to a constant as leakage populations stabilize.

B.6 Surface code simulations well below threshold

To gain insight into the future importance of leakage removal for scaling quantum error correction, we performed numerical simulations of distance-5 and distance-7 surface codes and evaluated their logical performance subject to different levels of leakage. We performed these simulations using a Kraus operator simulation detailed in [4]. We include operators that accurately reflect leakage transport, leakage phase errors, and the leakage removal parameters for both the *MLR* and *DQLR* strategies, but do not include any sources of leakage in the baseline model. The error model for the baseline simulations is detailed in Table B.1, which reflects values well below the threshold for the surface code, where the ratio of the logical error rate between distance-5 and distance-7 surface codes $\Lambda_{5/7}$ is around 4. We repeat these simulations with varying amounts of leakage error added in for the two strategies with leakage removal, which is presented in the main text Figure 5c.

B.7 Error correlations in the surface code experiment

One technique to evaluate error correlations in the surface code is to employ p_{ij} correlation matrices [92, 1, 2, 4].

By analyzing the data presented in Figure B.6 with p_{ij} correlation matrices, we can elucidate the presence of error correlations in space and time over the course of the QEC experiment.

In Figure B.7a, we present averaged autocorrelated p_{ij} matrices for both X- and Z-basis stabilizers over the 30 cycle experiment under the three leakage removal strategies investigated in this work. Z-basis stabilizers do not report detection probabilities at the time-boundary cycles 0 and 30 and thus do not have p_{ij} elements associated with those cycles. An autocorrelated p_{ij} matrix for a given stabilizer reports the correlation of detection events between cycles i and j. Independent Pauli errors present themselves as detection events on consecutive cycles $|i - j| = 1$. In such an ideal setting with only fully independent errors, we expect non-local correlated errors to vanish, *i.e.* $p_{ij} = 0$ where $|i - j| > 1$. Leakage and other time-non-local error sources will break this assumption and force those p_{ij} elements to be non-zero.

For *No reset*, non-local correlations immediately appear at cycle 1 and increase in intensity over the course of the 30-cycle experiment. The high (>1%) values for non-local p_{ij} matrix elements at large correlation distances $|i - j|$ indicates that non-local correlations are significant contributors to logical performance degradation, and that QEC cannot be scaled in time under these conditions.

For *MLR*, non-local correlations are visibly reduced when compared to *No reset*. Still, an impactful degree of non-local correlation remains, primarily stemming from data qubit leakage and resultant leakage-induced correlated errors, such as CZ phase errors highlighted in Figure 2e of the main text.

For *DQLR*, nearly all non-local correlations are heavily suppressed in the experiment. This is seen by very small correlation magnitudes <0.2% at correlation distances greater than 1. Qualitatively, the effective suppression of time-non-local correlations suggests we are closer to fulfilling the QEC requirement of uncorrelated errors. Furthermore, this is evidence that our DQLR procedure does not introduce additional unwanted correlations within the experiment.

We isolate and average the autocorrelated p_{ij} matrix elements for cycles $i = 29$ and j in 19–27, inclusive, in Figure B.7b. This offers a quantitative profile to the degree of non-local correlations present in the surface code as a function of correlation distance $i - j$. Again, under ideal circumstances with fully independent errors, the correlation magnitude $|p_{ij}|$ should be 0 at all correlation distances greater than 1. *DQLR* most closely approximates this condition, where $|p_{ij}|$ exceeds 0.1% only for distance-2 correlation and otherwise is below 0.1% for all distances up through 10. The 1 SD error bars for *DQLR* suggest that we cannot resolve variations in $|p_{ij}| < 0.1\%$ for these non-local correlations. However, for *MLR*, $|p_{ij}|$ is about an order of magnitude higher at 1% for distance-2 correlation, and slowly decays with distance, remaining above 0.1% even at distance-10. Finally, *No reset* never has $|p_{ij}| < 1\%$ for any of the correlation distances considered here.

These observed correlation magnitudes suggest that if scalable QEC requires near-independent errors, complete leakage removal on all qubits must be carried out in some form. The *DQLR* strategy presented in this work provides one pathway to reaching that requirement. As a corollary, we show that *No reset* and *MLR* cannot support this requirement under existing leakage rates and transport mechanisms.

References

[1] Matt McEwen et al. “Removing leakage-induced correlated errors in superconducting quantum error correction”. In: *Nature Communications* 12.1 (Dec. 2021), p. 1761. DOI: `10.1038/s41467-021-21982-y`.

[2] Google Quantum AI et al. “Exponential suppression of bit or phase errors with cyclic error correction”. In: *Nature* 595.7867 (July 15, 2021). _eprint: 2102.06132, pp. 383–387. DOI: `https://doi.org/10.1038/s41586-021-03588-y`.

[4] Google Quantum AI et al. “Suppressing quantum errors by scaling a surface code logical qubit”. In: (2022). DOI: `10.48550/ARXIV.2207.06431`.

[92] T. E. O’Brien, B. Tarasinski, and L. DiCarlo. “Density-matrix simulation of small surface codes under current and projected experimental noise”. In: *npj Quantum Information* 3.1 (Dec. 2017). Publisher: Nature Publishing Group, p. 39. DOI: `https://doi.org/10.1038/s41534-017-0039-x`.

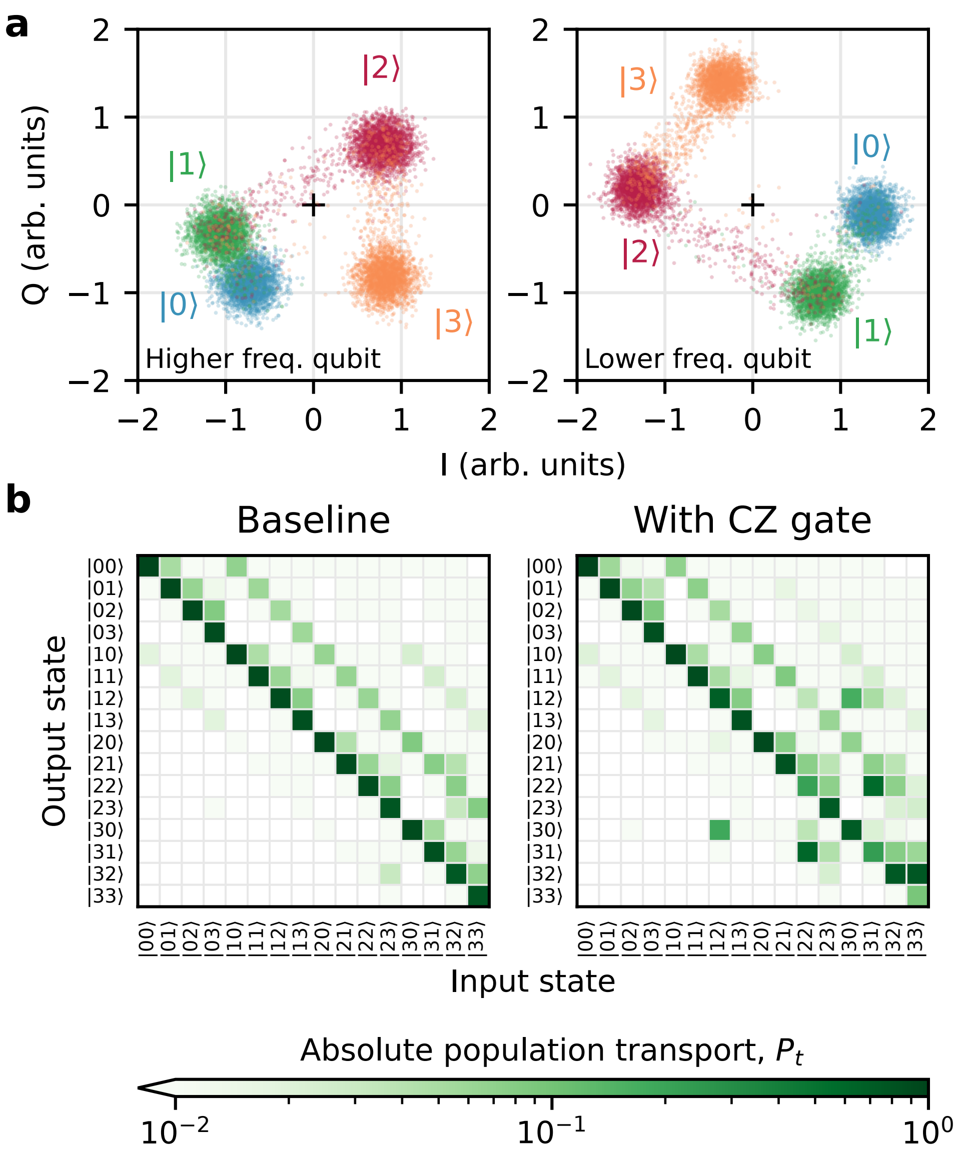


Figure B.1: **Measuring leakage transport.** a) Readout clouds for the pair of qubits used in the leakage transport experiment, showing good distinguishability between all of the lowest four qubit energy states. b) Raw measured population transport matrices for the two transport experiments. In both matrices, we can see the effect of T_1 decay during measurement, which is enhanced for the higher levels. Subtracting "Baseline" from "With CZ gate" produces the matrix shown in Figure 2b of the main text.

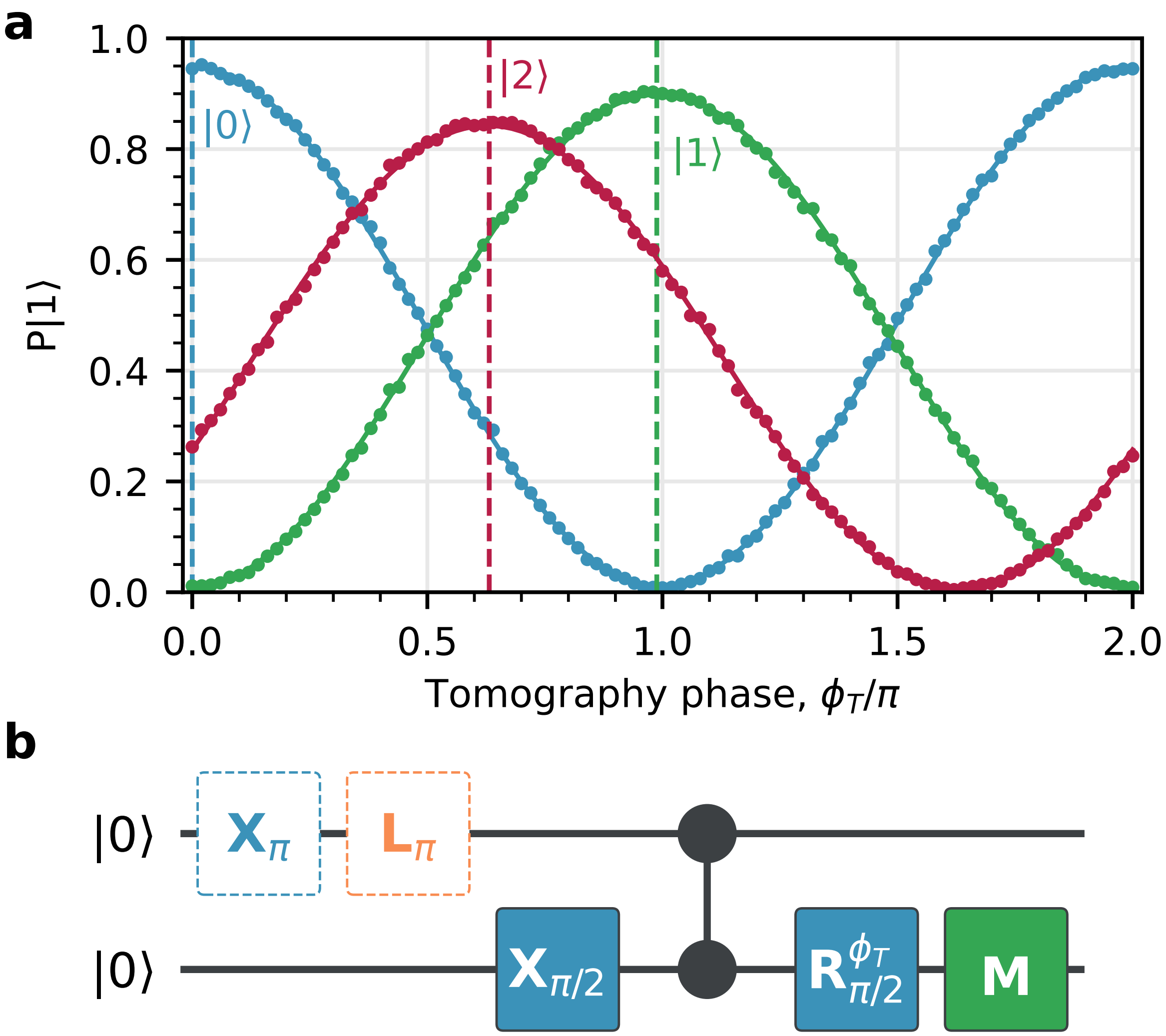


Figure B.2: **Measuring leakage phases.** a) Raw data for the leakage phase experiment on a single pair of qubits. Each of $|0\rangle$, $|1\rangle$, and $|2\rangle$ is prepared on the higher energy qubit, and then a Ramsey experiment with an interleaved CZ gate is performed on the lower frequency qubit. The solid lines are sinusoidal fits and the dashed lines indicate the extracted phase shifts ϕ shown in the main text. b) The hardware circuit executed in the above experiment in the $|2\rangle$ case. The two initial rotations (**X** and **L**) on the higher energy qubit (top) are removed when preparing $|0\rangle$, and the second rotation **L** is removed when preparing $|1\rangle$.

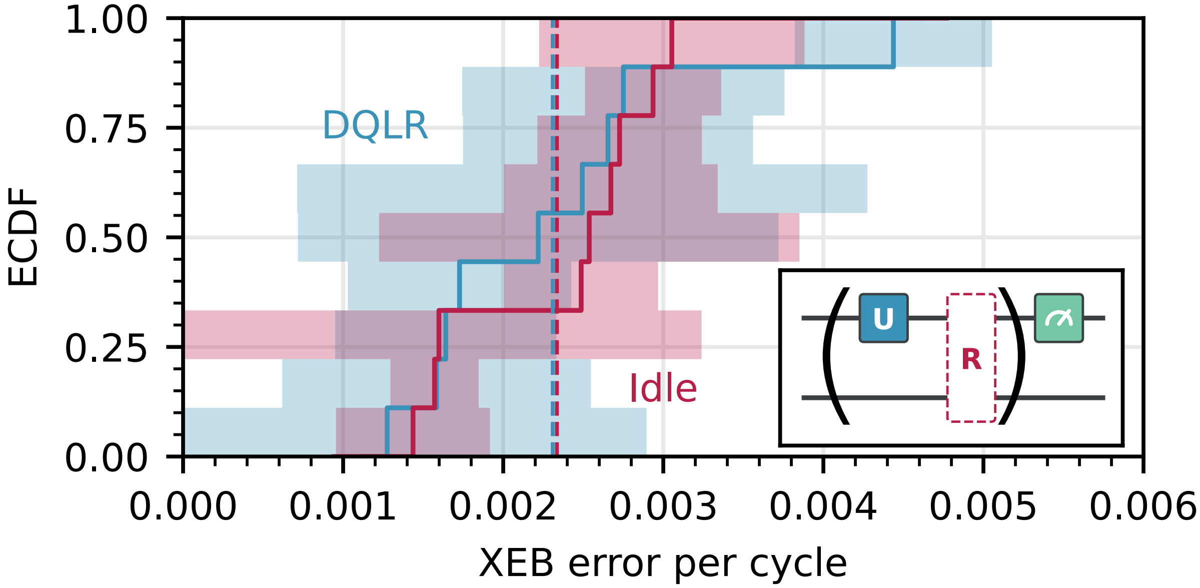


Figure B.3: **Cross-entropy benchmarking (XEB) of DQLR operation.** Inferred XEB error per cycle for 9 data qubits when DQLR is applied on the data qubit, compared to XEB error per cycle when the data qubit idles for the equivalent time (53 ns). Shaded region indicates 1 SD error of inferred XEB error per cycle. Vertical dashed lines indicate the average XEB error per cycle over all data qubits. (Inset) XEB circuit where random single-qubit unitary rotations (**U**) are repeatedly applied to the target qubit, interleaved with a reset operation (**R**), which is either DQLR or 53 ns of idle. The final state of the target qubit is measured. The cross-entropy between the measured and expected distribution of states is calculated and the resulting XEB error per cycle is inferred.

Table B.1: **Hypothetical device error model capable of achieving $1/\Lambda_{5/7} = 1/4$.**

Parameter	Value
Single qubit gate Pauli error	2×10^{-4}
CZ gate Pauli error	1×10^{-3}
Readout and reset error	1×10^{-2}
Idling Pauli error from decay from dynamical decoupling	3.25×10^{-3} 1×10^{-3}
Qubit T_1	75 μs
Qubit T_2	75 μs
Single qubit gate time	15 ns
CZ gate time	25 ns
Readout + Reset time	300 ns

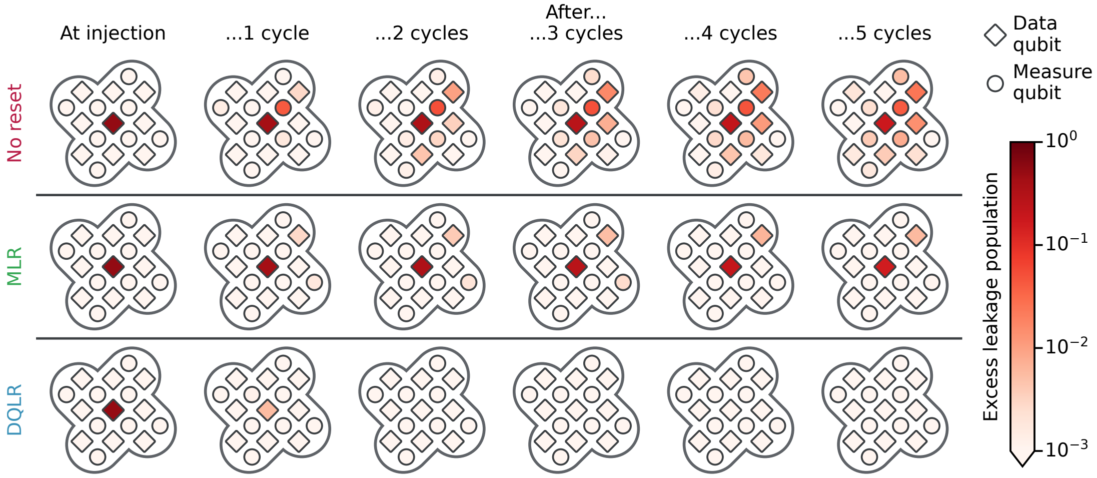


Figure B.4: **Leakage dynamics in a surface code with different removal strategies.** Comparison of excess leakage population dynamics over 5 cycles for all qubits in a distance-3 surface code after a full $|1\rangle \rightarrow |2\rangle$ leakage injection during the first cycle. With *No reset*, leakage transport mechanisms lead to increasing leakage population over nearly all qubits involved in the code. With the introduction of *MLR*, a sink for leakage population is present on measure qubits, mitigating the spread of leakage from leakage transport effects. The central leakage-injected data qubit still remains significantly leaked, even after 5 cycles. Using *DQLR*, leakage populations on all qubits in the code are brought to about 0.1% or lower within 2 cycles.

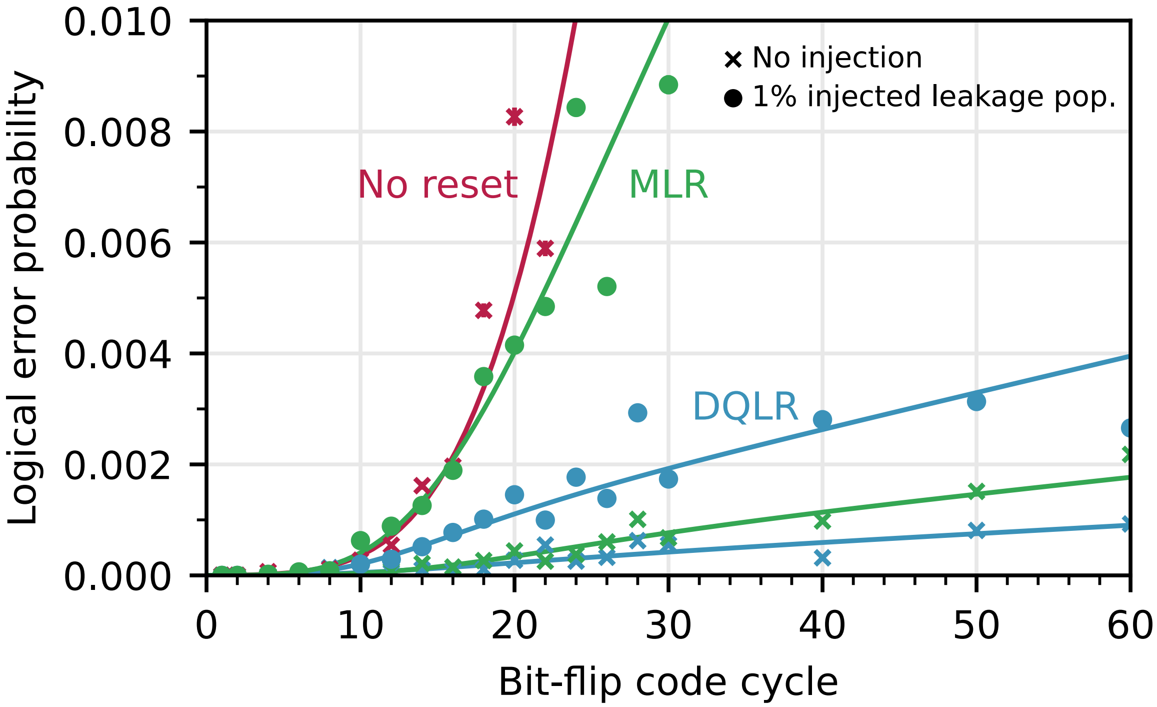


Figure B.5: **Logical performance of bit-flip code in time.** Logical error probability of the distance-21 bit-flip code over 60 cycles for the three reset strategies *No reset* (red), *MLR* (green), and *DQLR* (blue). For *MLR* and *DQLR*, we also execute the code while injecting 1% leakage population per cycle. At small (<5) numbers of cycles, boundary effects cause all three reset strategies to perform similarly. As the code progresses through more and more cycles, however, logical performance for the three strategies diverge as leakage populations are handled differently.

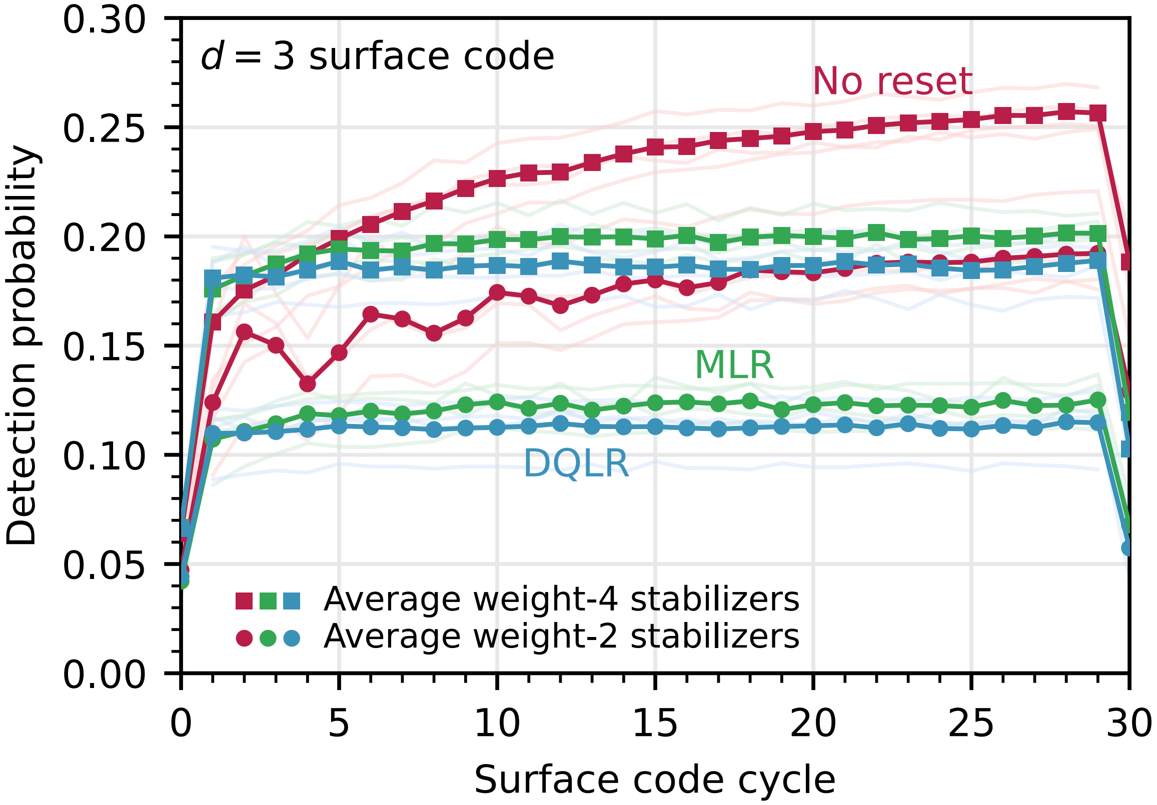


Figure B.6: **Distance-3 surface code detection probabilities.** Average detection probabilities over 30 surface code cycles for weight-2 (circles) and weight-4 (squares) stabilizers for three leakage removal strategies *No reset* (red), *MLR* (green), and *DQLR* (blue). Individual stabilizer detection probabilities are shown as lighter lines.

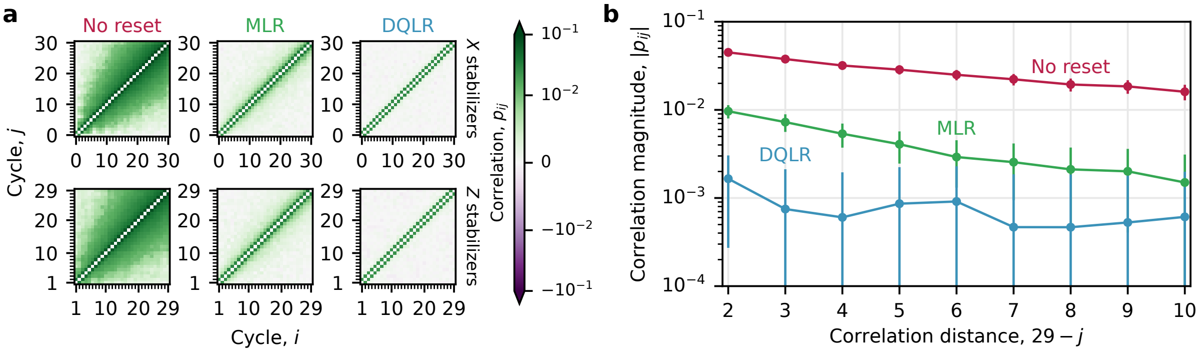


Figure B.7: **Surface code p_{ij} correlations.** a) Average p_{ij} matrices of X and Z stabilizers under different leakage removal strategies. Correlations along the upper and lower diagonals (p_{ij} where $|i-j|=1$) represent independent timelike errors, whereas non-local correlations (p_{ij} where $|i-j|>1$) can be primarily attributed to leakage-induced correlated errors. For *No reset*, non-local correlations intensify as the code is executed in time, suggesting increasing leakage-induced correlated errors from growing leakage population. With *MLR*, non-local correlations are reduced but remain, which are manifestations of data qubit leakage-induced correlated errors. For *DQLR*, complete leakage removal over all qubits results in suppression of non-local correlations to about 0.1% or lower. b) Correlation magnitude $|p_{ij}|$ averaged over all stabilizers for cycles $i=29$ and $j\in[19..27]$ under different leakage removal strategies. Average non-local correlation magnitudes do not exceed 0.2% for *DQLR*, whereas *No reset* and *MLR* have exponential decays in correlation magnitude with respect to correlation distance.

Appendix C

Supplementary Information regarding High-Energy Impacts

C.1 Energy cascade following a high energy impact

A high-energy impact produces an avalanche of excitations as the energy spreads through the system. Impacts are severe, depositing energy at much larger scales than those characteristic of our qubits and chip materials. As this energy quickly cascades from the initially small number of high energy particles to lower energies, it produces an explosion of excitations in the device.

This cascade progresses through several distinct stages as the event evolves [106]. The primary interactions in this avalanche are the initial deposition of energy by the incident radiation and the recombination of resulting charges, the propagation and down-conversion of phonons, and finally the creation and eventual recombination of quasiparticles in the qubits. We will discuss these in turn.

The impact event starts with the high energy particle crossing the chip and depositing a large amount of energy. The two particles of most concern are gamma rays from radioactivity in materials surrounding the chip, and muons produced by cosmic ray showers. Gamma rays are more common, depositing energy in the substrate via Compton scattering. Muons are more rare but deposit higher energies, leaving an ionization track along their path through the device. For a smaller sample with area 40 mm^2 gamma ray and cosmic ray events where reported with a rate of $\sim 1/(100 \text{ s})$ and $\sim 1/(500 \text{ s})$, and deposited energies around ~ 100 keV and ~ 1 MeV respectively [99]. For our carrier chip of 20 mm x 26 mm, the equivalent event rates are $\sim 1/(7.6 \text{ s})$ and $\sim 1/(38 \text{ s})$ respectively.

These impacts create electron-hole pairs in the insulating substrate of the device. The typical energy for an electron-hole pair is $E_{\text{e-h}} = 3.75\text{eV}$ [116], so the number of charges created by an event is $\sim 10^5$. The majority of electron-hole pairs immediately recombine and emit photons but $\sim 20\%$ of them avoid immediate recombination and propagate in the substrate [99]. These remaining free charges create further high energy phonons as they slow down to the speed of sound. In Si, the mobility of electrons is higher than that of holes, so typical events generate a net current of electrons resulting in the $1/f^2$ charge noise observed in charge sensitive transmons [117]. Slow charges eventually get trapped by ions and other defects in the substrate, with a characteristic length of $\lambda_{trap} \sim 300$ μm in Si [118]. Charges that arrive at superconducting materials are trapped at much shorter distances. They quickly emit phonons as they decrease in energy down to the superconducting gap and may get trapped in local variations of the gap. Such trapped charges act as effective two-level systems, and can be characterized by a non-uniform density of states [119].

Phonons produced by the event are primarily responsible for the spread of energy through the chip. High-energy phonons downconvert to lower energies as they propagate through the substrate, through the phonon-phonon interaction arising from the anharmonicity of the substrate. Initially, the high-energy phonons quickly downconvert to acoustic phonons with energies below the Debye energy ($\hbar\omega_D = 56$ meV for Si). This phonon decay rate however slows down at lower energies with

$$\gamma_{\text{insulator}}(E) \approx g\frac{E}{\hbar}\left(\frac{E}{\hbar\omega_D}\right)^4, \tag{C.1}$$

where $g \approx 0.01$ is the phonon anharmonicity for a typical insulator [120]. During the early stages of the impact (~ 100 μs), phonons propagating through the substrate can decay down to energies near ~ 5 meV, still significantly higher than the superconducting gap of aluminum $2\Delta \approx 0.36$ meV.

The absorption of phonons by metallic structures is the primary mechanism for further down conversion. In normal metals, acoustic phonons produce electron-hole pairs. Each particle in the pair emits low energy phonons, resulting in an efficient energy downconversion process with decay rate $\gamma_{\text{metal}}(E) \propto E$. In superconductors, the phonon absorption is similar to that in normal metals above the superconducting gap $E \gtrsim 2\Delta$ but ceases below it. The resulting quasiparticles similarly cool rapidly to near the superconducting gap by emitting phonons.

The spatial extent of the initial hotspot is also determined by the absorption of phonons by superconductors on the chip. The chip features an aluminum groundplane and qubit layer 100 nm thick on a silicon substrate around 500 μm thick, and also features significant volumes of indium in the form of bump bonds. These bump bonds are around 5 μm tall and cover $\sim 15\%$ of the surface area of the qubit chip. Indium features a superconducting gap $\Delta_{\text{In}} \approx 0.52$ meV, nearly three times higher than aluminum.

The rate at which a phonon traveling in a superconductor will break pairs and produce quasiparticles is given by [106, 121]

$$\tau_b^{-1} \approx \frac{1}{\tau_0^{ph}}; \qquad E_{ph} \sim 2\Delta \tag{C.2}$$

$$\tau_b^{-1} \approx \frac{1}{\pi\tau_0^{ph}} \frac{E_{ph}}{\Delta}; \qquad E_{ph} \gg 2\Delta \tag{C.3}$$

where τ_0^{ph} is the characteristic phonon lifetime, which is proportional to $1/\Delta$ and for aluminum is $\tau_0^{ph} \approx 0.24\,\mathrm{ns}$ [121]. First, we discuss the aluminium layer alone. Considering phonons downconverted by the substrate to $E_{ph} = 5$ meV, we find $\tau_b \approx 27$ ps. The absorption length is then $l_{ph} = c\tau_b \approx 170$ nm, where $c = 6.4$ km/s is the speed of sound in aluminum. This distance is comparable to the aluminum thickness, so most of the high energy phonons radiating from the initial impact will be absorbed locally by the aluminum. The initial quaiparticles will cool rapidly toward the superconducting gap, radiating further phonons with lower energies. These phonons will have energies ranging from the initial energy of the quasiparticle to near the superconducting gap. At a moderately lower energy of $E_{ph} = 1$ meV close to the superconducting gap of indium, we find a longer pairbreaking time $\tau_b \approx 130$ ps and absorption length $l_{ph} \approx 870$ nm. At the aluminum gap $E_{ph} = 2\Delta \approx 0.4$ meV, we find $\tau_b \approx 240$ ps and $l_{ph} \approx 1500$ nm. The reduced absorption at these lower energies would result in significant spread of the initial hotspot when considering the aluminum layer alone.

However, the large volumes of indium will also absorb phonons with energies above the indium superconducting gap. Even at the lowest relevant phonon energy $E_{ph} = 2\Delta_{\mathrm{In}} \approx 1$ meV, the phonon lifetime in indium is only $\tau_0^{ph} \approx 170$ ps. Given the speed of sound is $c \approx 1.2$ km/s, the phonon absorption length near the gap is $l = c\tau_0^{ph} \approx 200$ nm. Given the thickness of the bump bonds, any phonons above $E_{ph} = \Delta_{\mathrm{In}}$ impinging on the indium bumps will be absorbed. This additional absorptivity will contribute to tighter localization of the initial hotspot and will slow the effective spread of phonons until they are below $2\Delta_{\mathrm{In}}$. However, as the aluminum will downconvert phonons below the indium superconducting gap, we expect the rate of spread to increase as the event progresses.

The spread of error through the chip is also influenced by the rate of recombination in the superconductors. The recombination rate of quasiparticles that have cooled to near the gap is proportional to the quasiparticle density [121],

$$\tau_r^{-1} = \frac{2\pi\Delta\alpha^2(2\Delta)F(2\Delta)}{Z\hbar}x_{qp} \tag{C.4}$$

with $\alpha^2(2\Delta)F(2\Delta)$ the Eliashberg function at the gap energy, Z a renormalization parameter, and $x_{qp} = n_{qp}/n_{cp}$ the normalized density of quasiparticles. The above can be approximated as $\tau_r^{-1} \approx x_{qp} \cdot 21.8/\tau_0$ [106], with $\tau_0 = 440$ ns the characteristic quasiparticle time for aluminum [121]. Again, we will first consider the aluminum layer only. For a median impact delivering 100 keV of energy to the substrate [99], where 20% of the energy remains in free charges, the energy absorbed by the initial hotspot in the aluminum layer will be around $E_{hs} = 80$ keV. We take area of the hotspot to be $A = 10$ mm^2, similar to that seen in segment 1 of Fig. 4 of the main paper. The aluminum thickness $d = 100$ nm. For aluminum, $\Delta \approx 0.18$ meV and $n_{cp} \approx 4 \times 10^6$ μm^3. We calculate the density of quasiparticles $x_{qp} = E_{hs}/Ad\Delta n_{cp} \approx 1.1 \times 10^{-4}$. The recombination time is then $\tau_r \approx$ 180 μs, which agrees well with the experimentally observed time for the expansion of the initial hotspot. As the energy spreads through the device, the resulting quasiparticle densities will be lower and this recombination time will grow.

We now consider the recombinaiton in the indium. Indium features a very high diffusivity of quasiparticles, meaning any induced quasiparticle density will distribute itself throughout the bumpbond volume quickly. The bumpbonds are fabricated on top of the aluminum groundplane, separated by a thin titanium nitride (TiN) diffusion barrier. The superconducting gap of TiN films depends sensitively on the thickness and presence of contaminants, and there is evidence in literature for the possibility of a smeared gap that might allow conduction [122].

In the case that the TiN film has a lower superconducting gap than the indium, quasiparticles will rapidly flow from the indium into the aluminum, where they will cool to the aluminum gap. Unable to flow back into the indium, they will recombine in the aluminum at around the rate found previously. If the TiN film has a higher gap and features no subgap conduction, then we can similarly estimate the recombination time for the quasiparticles trapped in the indium. The initially absorbed phonons will distribute energy between the aluminum and indium by their exposed surface area and rate of absorbtion. Taking the same hotspot area of 10 mm^2, and considering an energy absorbed in the indium to be $E_{\mathrm{In}} = 50$ keV, the recombination timescales are similar. In indium, $n_{cp} \approx 13 \times 10^6$ μm^3, so we find a small quasiparticle density $x_{qp} \approx 1 \times 10^{-6}$ due to the large volume. The short characteristic phonon timescale $\tau_0^{ph} = 0.799$ ns results in a recombination time $\tau_r \approx 36$ μs. However, only recombinations taking place near the surface of the indium will be able to radiate out. Accounting for ratio of the phonon absorption length $l \approx 200$ nm and the height of the bumpbonds $z \approx 5$ μm, we calculate the effective relaxation time of the indium bumps as $\tau_r^{\mathrm{eff}} = \tau_r z/l \approx 0.9$ ms. This timescale is similar to the spread of energy from recombination in the groundplane, so recombination in indium is also compatible with the observed spread of the hotspot. In the case where a lower proportion of energy is trapped in the indium, the recombination time scale grows larger than the contribution from aluminum and will contribute to the leading dynamics of the spread. As such, the indium may play a role in the initial absorption of energy. However, the presence of the aluminum will soon downconvert phonons to below the indium gap, effectively freezing the indium out of the dynamics at longer timescales.

The presence of quasiparticles near the junctions in the qubits are the cause of the observed errors. The relaxation rate of the qubits is proportional to the density of quasiparticles [104]:

$$\frac{\Gamma_1}{\omega_q} = \sqrt{\frac{2\Delta}{\pi^2 \hbar \omega_q}} x_{qp} \tag{C.5}$$

In Fig. 4 of the main paper we extracted an effectively global T_1 time around 1 μs at the peak of large events, which is compatible with estimates based on similar device structures [106]. As a consistency check, we can extract the total energy necessary in the qubit layer to induce this T_1 time. Using a typical qubit frequency $\omega_q = 2\pi \times 6$ GHz, we calculate a quasiparticle density of $x_{qp} \approx 2.1 \times 10^{-5}$ at the peak of the event, once the error has spread over a large area. We can convert this to an absolute density of $n_{qp} = 87\ \mu\text{m}^{-3}$. Producing this density evenly over the full aluminum qubit layer on the 10 mm x 10 mm qubit chip would require an absorbed energy in the aluminum of $E = n_{qp} V \Delta \approx 160$ keV, which corresponds well to the expected energy depositions from large events.

Finally, the energy must eventually escape the chip through thermalization with the surrounding cryostat. The chip is floating in the package, suspended by aluminum wirebonds. The observed ~ 25 ms timescale for the return of the device to equilibrium is in reasonable agreement with a simple model of thermalization that gives ~ 8 ms [106]. Given the shorter timescale for this relaxation found in Ref. [99], we expect that increased thermalization will serve to reduce the overall duration of the event.

These interactions can help us to understand the error mechanism observed in this work, but we neglect some of the intricacies involved in the energy cascade in a practical device. Impacts depositing different initial energies will influence the details of the induced densities and timescales of recombination. Impacts in different locations across the device will also influence the exposure of the qubit patch to the initial hotspot and to the spreading of errors. Finally, the interplay between other structures and materials on the chip will also introduce additional complexity, including the distribution of energy between the indium and aluminum, the behaviour of titanium nitride diffusion barrier and the presence of a second substrate as carrier in flip-chip devices. A quantitative understanding of the energy cascade in such large devices remains an open question, and will be important for guiding efforts to mitigate events by introducing structures that affect the energy cascade.

C.2 Absence of Excitation Errors

A key characteristic of quasiparticle poisoning in our qubits is the predicted asymmetry between excitation and decay errors. To measure this asymmetry, we also perform RReCS experiments looking for excitation errors by preparing qubits in $|0\rangle$ and recording measurements of $|1\rangle$ as errors. Fig. C.1a shows timetraces for both a regular RReCS experiment initializing $|1\rangle$ and for excitation-sensitive RReCS experiment initializing $|0\rangle$, including both the raw timeseries and matched filtered timeseries. We note that initializing $|0\rangle$ produces much lower background error rates, as T_1 error is suppressed. We find a lack of correlated events, especially clear in the low noise matched filtered timeseries. Fig. C.1 shows histograms for 5 such datasets of each type. We see that the histograms when initializing $|0\rangle$ correspond closely to the independent error model, and do not feature the elevated counts of high-error-number points characteristic of the events found when initializing $|1\rangle$. Over more than 35 of such datasets, we found no evidence of any correlated events in excitation.

This represents further evidence that the error mechanism responsible for the correlated error events is high densities of quasiparticles. Quasiparticles in the superconductor rapidly cool to energies near the superconducting gap Δ. When tunneling across the

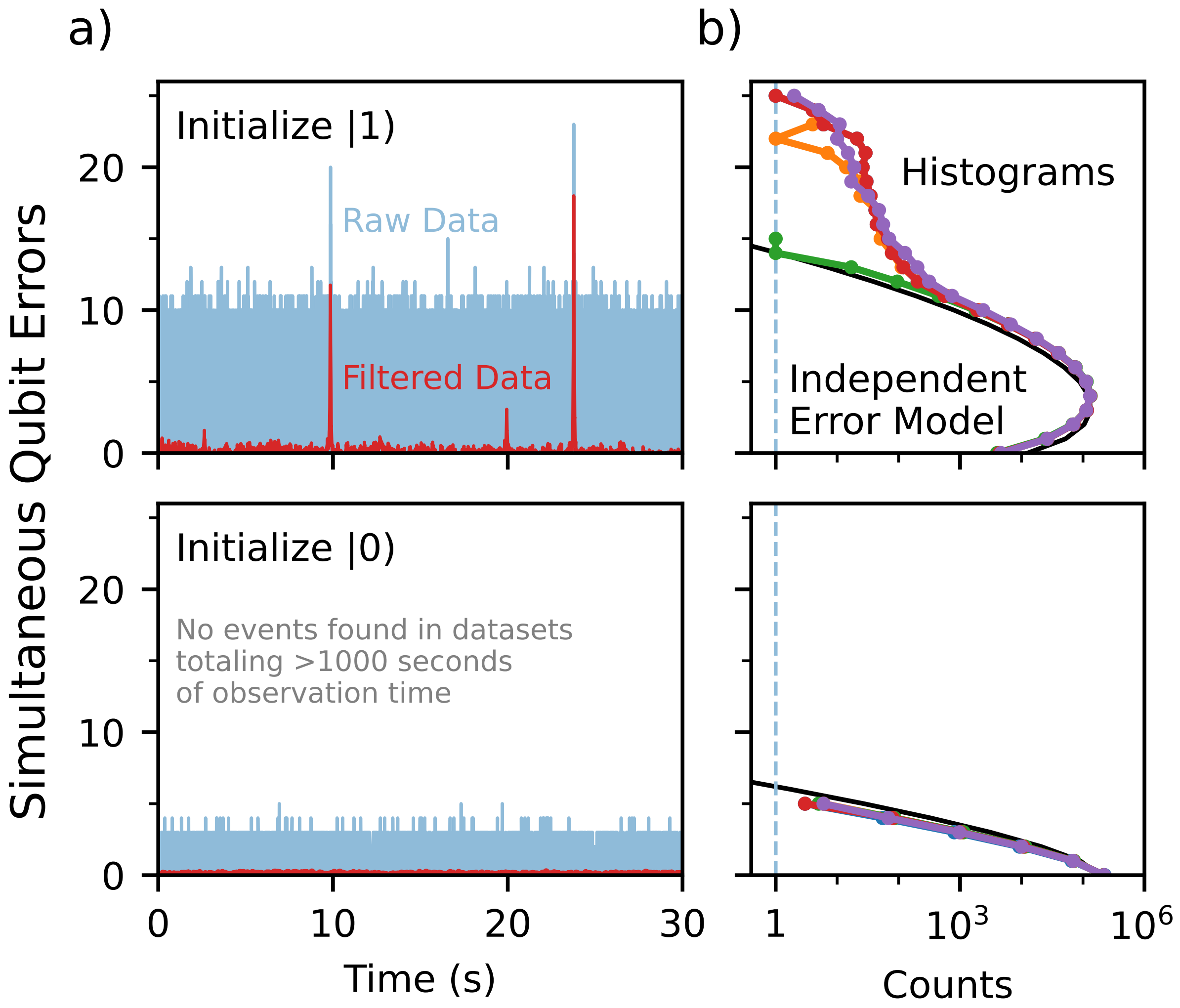


Figure C.1: **Decay vs Excitation Errors.** (a) Timetraces for RReCS experiments which initialise $|1\rangle$ and $|0\rangle$ respectively. Baseline error rates are much lower when initialising $|0\rangle$. (b) Histograms for 5 datasets (different colors) initialising $|1\rangle$ and $|0\rangle$ respectively. No datasets initializing $|0\rangle$ display any high-number counts that indicate a correlated error event.

Josephson junction, these quasiparticles are able to absorb the qubit energy E_{10}, producing a decay error. Once cooled to near the gap, quasiparticles are not capable of exciting the qubit state, as this requires energies of at least $\Delta + E_{10}$. As such, excitation errors are not expected to be present in chip-wide quasiparticle poisoning, as they would be in the case of an induced chip-wide readout or coherent control failure. The strong asymmetry in the rates of correlated decay and excitation errors is therefore strong evidence that the errors in events are due to excess quasiparticles.

C.3 Peak-finding and Event Parameter Distributions

Given the large volume of data that long RReCS experiments produce, it is important to be able to identify events in a reliable and automated fashion. Fig. C.2a shows time slice of raw data including an event, illustrating the sharp jump and exponential decay that is characteristic of all events. Because the event shape is reliable, we can use a matched filter to optimally reduce noise and locate peaks. Throughout, we will use a template function:

$$\text{Template(t)} = \begin{cases} a\exp(-(t-t_0)/\tau_{\text{decay}}) + c & ; t \geq t_0 \\ c & ; t < t_0 \end{cases}$$

We remove the DC component of the time series, and then apply a matched filter created from the template with the values $\tau_{\text{decay}} = 20$ ms, $a = 1$, $c = 0$, $t_0 = 0$. Fig. C.3b shows the resulting filtered function, illustrating the low noise level and symmetric peak produced by the matched filter. Our selection of $\tau_{\text{decay}} = 20$ ms will influence the symmetry and scale of the resulting peak, but not the location of the maximum. We then apply a threshold of 2 errors to the filtered time series and return local maximum for

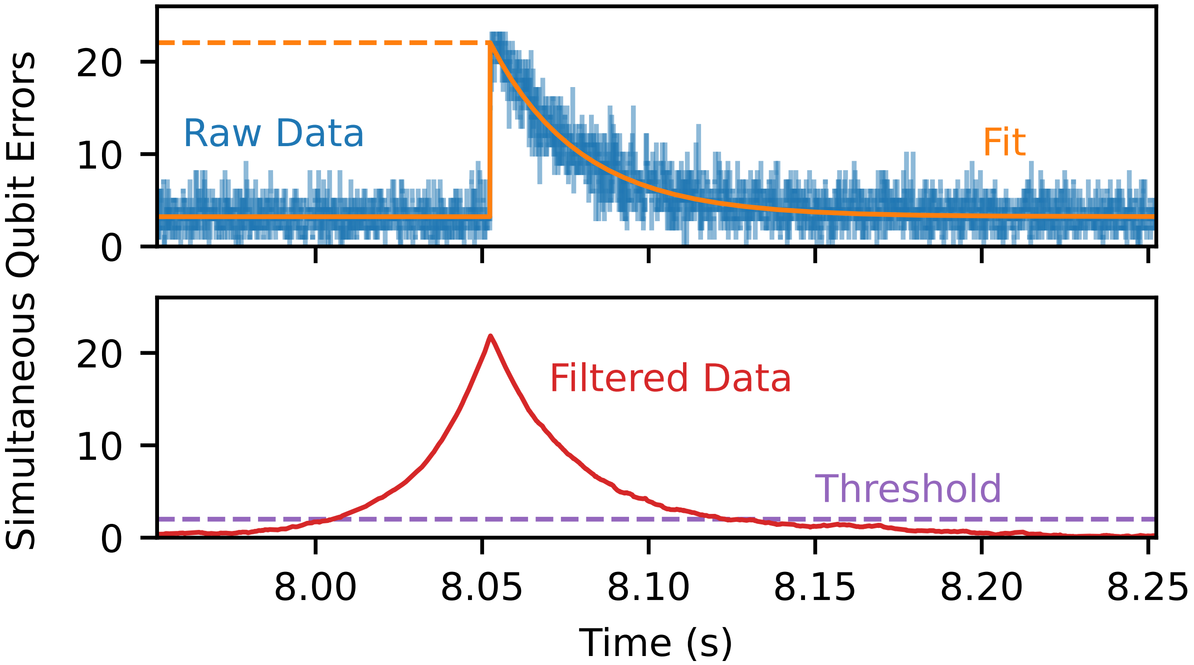


Figure C.2: **Peak-finding procedure.** (a) A time slice including a single event, showing the raw data (blue) and the final least squares fit used to determine peak parameters (orange). (b) The same time slice after application of a matched filter (red), including the threshold at 2 errors above mean (purple) used to locate peaks for fitting.

that our experimental values are compatible with

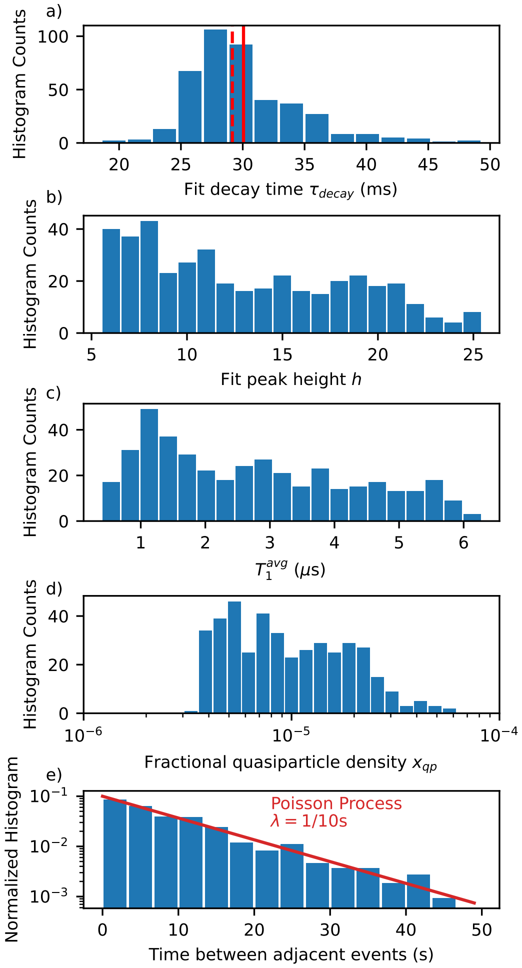
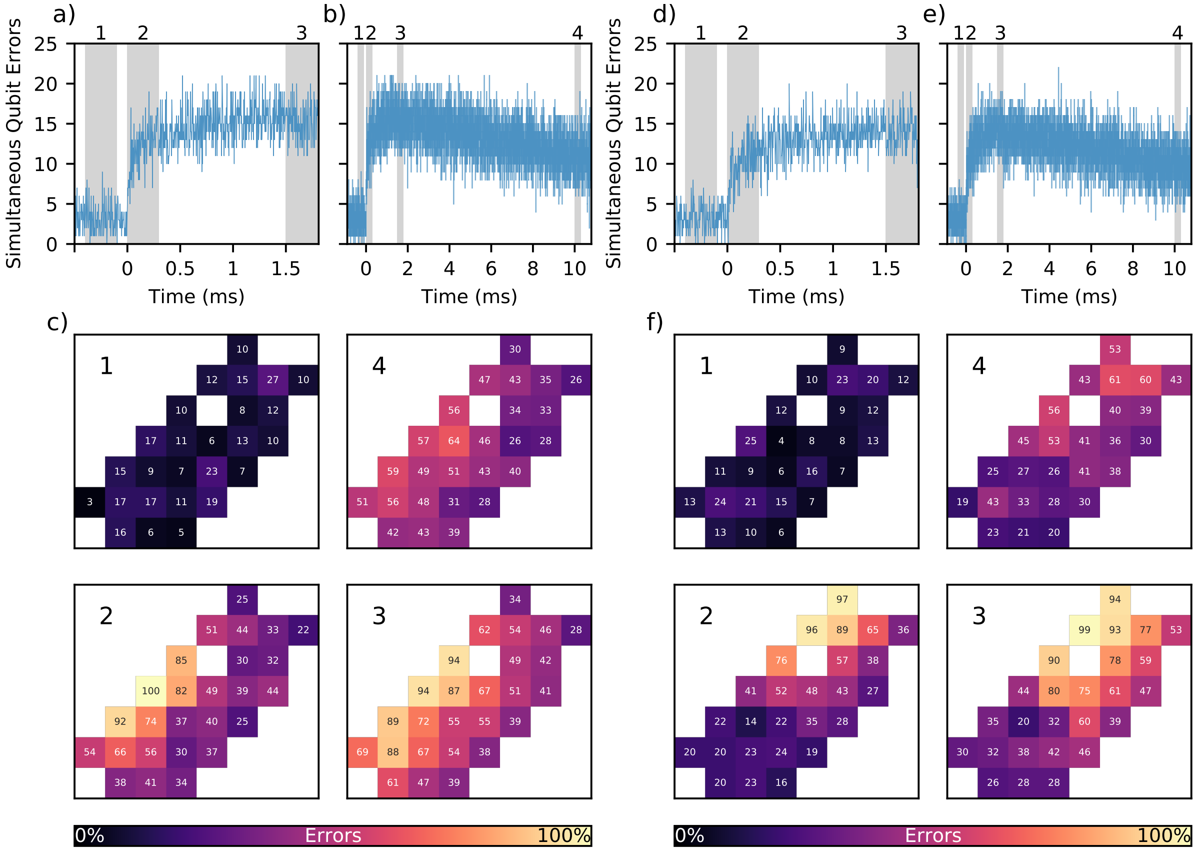


Figure C.3: **Impact Parameter distributions.** Data from 415 events, extracted from 100 datasets of 60 s as discussed in the main paper and of which a subset is shown in Fig. 2. (a) The distribution of exponential decay times τ_{decay} fit to events, including the mean (solid) and median (dashed). (b) The distribution of fit peak heights. The minimum peak height possible to identify with this analysis method is around 6, and the maximum is the full number of qubits at 26. (c) The distribution of average T_1 values, computed from the event heights using the global T_1 model given in the main paper: $h = N_Q[1 - \exp(-t'_{\text{sampling}}/T_1^{\text{avg}})]$. (d) The distribution of fractional quasiparticle density induced, assuming equal distribution over the total volume of Al on the chip. Typically acceptable backgrounds are $x_{qp} \approx 10^{-8}$. (e) The distribution of time periods between neighbouring events, showing a strong correspondence to the distribution for a Poisson process with $\lambda = 1/10s$ (red).



Event B Event C

Figure C.4: **Further highly time-resolved events.** (a,b,d,e) Time slices from two highly time-resolved RReCS experiments with an interval of 3 μs between datapoints. Both events show the same timescales over their evolution despite differing in height. (c,f) Heat maps of the qubit patch, averaged over 300μs slices located (1) before the event, (2) at the initial impact, (3) after the rise to the peak value, (4) during the exponential tail of the event. Both events show initial high-error hot-spots in different locations, but later display elevated error levels throughout the chip.

each section over the threshold to identify peaks. This technique is capable of detecting peaks of heights greater than 2 errors over the background error value around 4 errors, with the smallest identified peaks having absolute heights of 6 errors. This technique could be easily refined to capture events with heights even closer to baseline due to the reliability of the shape of the event even at very small scales.

To extract parameters for each event, we return to the raw data to make a direct fit. We perform a least squares fit using the template given above to the time-series around the located peak. Fig. C.2a shows the resulting fit over the raw time series, from which we extract the peak time t_0, the decay constant τ_{decay} and the peak height $h = a + c$.

We perform this fitting procedure for each of the peaks found in our data. Fig. C.3a shows the distribution of fitted decay constants τ_{decay} for the 415 peaks found in the 100 datasets presented in the main paper, showing a tight distribution around 25-30 ms. Fig. C.3b shows the distribution of heights $h = a + c$ for the same peaks. We see events from the minimum size identifiable of 6 errors at peak, up to the full size of our qubit patch at $N_Q = 26$ simultaneous errors. Smaller events are found more frequently, qualitatively matching the distribution of deposited energies for impinging particles [99]. Using the global T1 model described in the main paper, we can convert these peak heights to an effective chip-wide T_1^{avg}. The resulting distribution is shown in Fig. C.3c. Finally, we can use Eqn.C.4 to calculate the necessary quasiparticle density to produce the observed T_1^{avg}, assuming a uniform distribution of quasiparticles over the full volume of the aluminum on chip. This distribution is shown in Fig. C.3d. We can see that the resulting densities are several orders of magnitude above the typically acceptable background $x_{qp} \approx 10^{-8}$.

We also investigate the distribution of events over time. One clear indicator of a Poisson distribution is an exponential probability density function in the time between neighbouring events. From the same 100 datasets as above, we find 326 time periods between adjacent events in the same dataset. Fig. C.3e shows the normalised distribution of these times, along with the probability density function for a Poisson process with $\lambda = 1/10\text{s}$. The strong correspondence indicates that events occur independently, as one might expect for events produced by radioactive decays or isolated cosmic ray impact events. We also note that the event rate found is sensitive to the lower cutoff for size of events located, as less sensitive methods will locate fewer events.

C.4 Further Events with High Time Resolution

In order to study the evolution of events in greater detail, we performed RReCS experiments with a 3 μs interval between datapoints, and lasting 3 s total. Along with the event shown in Fig. 3 of the main text (Event A), two further events at this time resolution are shown in Fig. C.4 (Events B and C).

All three events display the same overall timescales, showing an initial jump in a few tens of μs, a saturation to peak value in around 1 ms and an exponential decay back to baseline with a timescale of $\sim$ 25 ms. However, each event shows a different location for the initial area of high error rates; Event A features an initial impact centered on the right side of the qubit patch, Event B features an initial impact on the bottom left side of the qubit patch, and Event C sees an initial impact on the top left of the patch. All three immediate impacts primarily affect a small patch of qubits, producing a jump from $\sim$ 4 simultaneous errors at baseline to $\sim$ 10 errors. Both events then spread through the chip, rising to a peak value around 1.5 ms after the initial impact. Finally, both events show the same exponential decay to baseline.

This provides further evidence that the initial localization is caused by a single point of impact and in a location that is independent between different events.

References

[99] C. D. Wilen et al. “Correlated charge noise and relaxation errors in superconducting qubits”. In: *Nature* 594.7863 (June 17, 2021), pp. 369–373. DOI: `10.1038/s41586-021-03557-5`.

[104] C. Wang et al. “Measurement and control of quasiparticle dynamics in a superconducting qubit”. In: *Nature Communications* 5.1 (Dec. 2014), p. 5836. DOI: `https://doi.org/10.1038/ncomms6836`.

[106] John M. Martinis. “Saving superconducting quantum processors from decay and correlated errors generated by gamma and cosmic rays”. In: *npj Quantum Information* 7.1 (Dec. 2021), p. 90. DOI: `10.1038/s41534-021-00431-0`.

[116] K. Ramanathan and N. Kurinsky. “Ionization yield in silicon for eV-scale electron-recoil processes”. In: *Physical Review D* 102.6 (Sept. 28, 2020). _eprint: 2004.10709, p. 063026. DOI: `https://doi.org/10.1103/PhysRevD.102.063026`.

[117] B. G. Christensen et al. “Anomalous charge noise in superconducting qubits”. In: *Physical Review B* 100.14 (Oct. 24, 2019). Publisher: American Physical Society, p. 140503. DOI: `10.1103/PhysRevB.100.140503`.

[118] R. A. Moffatt et al. “Spatial imaging of charge transport in silicon at low temperature”. In: *Applied Physics Letters* 114.3 (Jan. 21, 2019), p. 032104. DOI: `10.1063/1.5049691`.

[119] S. E. de Graaf et al. “Two-level systems in superconducting quantum devices due to trapped quasiparticles”. In: *Science Advances* 6.51 (Dec. 18, 2020). Publisher: American Association for the Advancement of Science, eabc5055. DOI: `10.1126/sciadv.abc5055`.

[120] V. L. Gurevich. *Kinetics of phonon systems*. Moscow: Nauka, 1980.

[121] S. B. Kaplan et al. “Quasiparticle and phonon lifetimes in superconductors”. In: *Physical Review B* 14.11 (Dec. 1, 1976). Publisher: American Physical Society, pp. 4854–4873. DOI: `10.1103/PhysRevB.14.4854`.

[122] P. C. J. J. Coumou et al. "Electrodynamic response and local tunneling spectroscopy of strongly disordered superconducting TiN films". In: *Physical Review B* 88.18 (Nov. 7, 2013). Publisher: American Physical Society, p. 180505. DOI: `10.1103/PhysRevB.88.180505`.

Full Bibliography

[1] Matt McEwen et al. “Removing leakage-induced correlated errors in superconducting quantum error correction”. In: *Nature Communications* 12.1 (Dec. 2021), p. 1761. DOI: `10.1038/s41467-021-21982-y`.

[2] Google Quantum AI et al. “Exponential suppression of bit or phase errors with cyclic error correction”. In: *Nature* 595.7867 (July 15, 2021). _eprint: 2102.06132, pp. 383–387. DOI: `https://doi.org/10.1038/s41586-021-03588-y`.

[3] Matt McEwen et al. “Resolving catastrophic error bursts from cosmic rays in large arrays of superconducting qubits”. In: *Nature Physics* 18.1 (Jan. 2022), pp. 107–111. DOI: `10.1038/s41567-021-01432-8`.

[4] Google Quantum AI et al. “Suppressing quantum errors by scaling a surface code logical qubit”. In: (2022). DOI: `10.48550/ARXIV.2207.06431`.

[5] Craig Gidney, Michael Newman, and Matt McEwen. “Benchmarking the Planar Honeycomb Code”. In: *Quantum* 6 (Sept. 21, 2022), p. 813. DOI: `10.22331/q-2022-09-21-813`.

[6] Kevin C. Miao and Matt McEwen. “Complete Leakage Removal in the Surface Code on Superconducting Qubits”. In: *Submitted Nature Physics* (2022).

[7] Scott Aaronson. “NP-complete problems and physical reality”. In: *ACM SIGACT News* 36.1 (Mar. 2005), pp. 30–52. URL: `https://www.scottaaronson.com/papers/npcomplete.pdf`.

[8] András Gilyén, Seth Lloyd, and Ewin Tang. “Quantum-inspired low-rank stochastic regression with logarithmic dependence on the dimension”. In: (2018). Publisher: arXiv Version Number: 1. DOI: `10.48550/ARXIV.1811.04909`.

[9] Ewin Tang. “A quantum-inspired classical algorithm for recommendation systems”. In: *Proceedings of the 51st Annual ACM SIGACT Symposium on Theory of Computing*. STOC ’19: 51st Annual ACM SIGACT Symposium on the Theory of Computing. Phoenix AZ USA: ACM, June 23, 2019, pp. 217–228. DOI: `10.1145/3313276.3316310`.

[10] Ewin Tang. “Quantum Principal Component Analysis Only Achieves an Exponential Speedup Because of Its State Preparation Assumptions”. In: *Physical Review Letters* 127.6 (Aug. 4, 2021), p. 060503. DOI: `10.1103/PhysRevLett.127.060503`.

[11] P.W. Shor. “Algorithms for quantum computation: discrete logarithms and factoring”. In: *Proceedings 35th Annual Symposium on Foundations of Computer Science*. 35th Annual Symposium on Foundations of Computer Science. Santa Fe, NM, USA: IEEE Comput. Soc. Press, 1994, pp. 124–134. DOI: `10.1109/SFCS.1994.365700`.

[12] Lov K. Grover. “A fast quantum mechanical algorithm for database search”. In: *Proceedings of the twenty-eighth annual ACM symposium on Theory of computing - STOC ’96*. the twenty-eighth annual ACM symposium. Philadelphia, Pennsylvania, United States: ACM Press, 1996, pp. 212–219. DOI: `10.1145/237814.237866`.

[13] Aram W. Harrow, Avinatan Hassidim, and Seth Lloyd. “Quantum Algorithm for Linear Systems of Equations”. In: *Physical Review Letters* 103.15 (Oct. 7, 2009), p. 150502. DOI: `10.1103/PhysRevLett.103.150502`.

[14] Daniel Gottesman. “The Heisenberg Representation of Quantum Computers”. In: (1998). Publisher: arXiv Version Number: 1. DOI: `10.48550/ARXIV.QUANT-PH/9807006`.

[15] Craig Gidney and Martin Ekerå. “How to factor 2048 bit RSA integers in 8 hours using 20 million noisy qubits”. In: *Quantum* 5 (Apr. 15, 2021). Publisher: Verein zur Förderung des Open Access Publizierens in den Quantenwissenschaften, p. 433. DOI: `10.22331/q-2021-04-15-433`.

[16] W. K. Wootters and W. H. Zurek. “A single quantum cannot be cloned”. In: *Nature* 299.5886 (Oct. 1982), pp. 802–803. DOI: `10.1038/299802a0`.

[17] Peter W. Shor. “Scheme for reducing decoherence in quantum computer memory”. In: *Physical Review A* 52.4 (Oct. 1, 1995), R2493–R2496. DOI: `10.1103/PhysRevA.52.R2493`.

[18] Daniel Gottesman. “Stabilizer Codes and Quantum Error Correction”. Publisher: arXiv Version Number: 1. PhD thesis. 1997. URL: `https://arxiv.org/abs/quant-ph/9705052`.

[19] A. R. Calderbank and Peter W. Shor. “Good quantum error-correcting codes exist”. In: *Physical Review A* 54.2 (Aug. 1, 1996), pp. 1098–1105. DOI: `10.1103/PhysRevA.54.1098`.

[20] Andrew Steane. “Multiple-particle interference and quantum error correction”. In: *Proceedings of the Royal Society of London. Series A: Mathematical, Physical and Engineering Sciences* 452.1954 (Nov. 8, 1996), pp. 2551–2577. DOI: `10.1098/rspa.1996.0136`.

[21] Jean-Pierre Tillich and Gilles Zemor. “Quantum LDPC Codes With Positive Rate and Minimum Distance Proportional to the Square Root of the Blocklength”. In: *IEEE Transactions on Information Theory* 60.2 (Feb. 2014), pp. 1193–1202. DOI: `10.1109/TIT.2013.2292061`.

[22] Nikolas P. Breuckmann and Jens N. Eberhardt. "Balanced Product Quantum Codes". In: (2020). Publisher: arXiv Version Number: 3. DOI: `10.48550/ARXIV.2012.09271`.

[23] Dorit Aharonov and Michael Ben-Or. "Fault Tolerant Quantum Computation with Constant Error". In: (1996). Publisher: arXiv Version Number: 2. DOI: `10.48550/ARXIV.QUANT-PH/9611025`.

[24] A. Yu Kitaev. "Fault-tolerant quantum computation by anyons". In: *Annals of Physics* 303.1 (1997), pp. 2–30. DOI: `10.1016/S0003-4916(02)00018-0`.

[25] S. B. Bravyi and A. Yu. Kitaev. "Quantum codes on a lattice with boundary". In: (1998). Publisher: arXiv Version Number: 1. DOI: `10.48550/ARXIV.QUANT-PH/9811052`.

[26] Austin G. Fowler et al. "Surface codes: Towards practical large-scale quantum computation". In: *Physical Review A* 86.3 (Sept. 18, 2012). Publisher: American Physical Society, p. 032324. DOI: `10.1103/physreva.86.032324`.

[27] Anirudh Krishna and David Poulin. "Fault-Tolerant Gates on Hypergraph Product Codes". In: *Physical Review X* 11.1 (Feb. 4, 2021), p. 011023. DOI: `10.1103/PhysRevX.11.011023`.

[28] Clare Horsman et al. "Surface code quantum computing by lattice surgery". In: *New Journal of Physics* 14.12 (Dec. 7, 2012), p. 123011. DOI: `10.1088/1367-2630/14/12/123011`.

[29] Benjamin J. Brown et al. "Poking Holes and Cutting Corners to Achieve Clifford Gates with the Surface Code". In: *Physical Review X* 7.2 (May 24, 2017), p. 021029. DOI: `10.1103/PhysRevX.7.021029`.

[30] Daniel Litinski and Felix von Oppen. "Lattice Surgery with a Twist: Simplifying Clifford Gates of Surface Codes". In: *Quantum* 2 (May 4, 2018), p. 62. DOI: `10.22331/q-2018-05-04-62`.

[31] E. Knill. "Fault-Tolerant Postselected Quantum Computation: Schemes". In: (2004). Publisher: arXiv Version Number: 1. DOI: `10.48550/ARXIV.QUANT-PH/0402171`.

[32] Sergey Bravyi and Alexei Kitaev. "Universal quantum computation with ideal Clifford gates and noisy ancillas". In: *Physical Review A* 71.2 (Feb. 22, 2005), p. 022316. DOI: `10.1103/PhysRevA.71.022316`.

[33] Sergey Bravyi and Jeongwan Haah. "Magic-state distillation with low overhead". In: *Physical Review A* 86.5 (Nov. 27, 2012), p. 052329. DOI: `10.1103/PhysRevA.86.052329`.

[34] Earl T. Campbell, Barbara M. Terhal, and Christophe Vuillot. "Roads towards fault-tolerant universal quantum computation". In: *Nature* 549.7671 (Sept. 14, 2017), pp. 172–179. DOI: `10.1038/nature23460`.

[35] Frank Arute et al. “Quantum supremacy using a programmable superconducting processor”. In: *Nature* 574.7779 (Oct. 24, 2019). Publisher: Nature Publishing Group, pp. 505–510. DOI: `10.1038/s41586-019-1666-5`.

[36] Sebastian Krinner et al. “Realizing repeated quantum error correction in a distance-three surface code”. In: *Nature* 605.7911 (May 26, 2022). Publisher: Springer Science and Business Media LLC, pp. 669–674. DOI: `10.1038/s41586-022-04566-8`.

[37] Neereja Sundaresan et al. “Matching and maximum likelihood decoding of a multi-round subsystem quantum error correction experiment”. In: (2022). DOI: `10.48550/ARXIV.2203.07205`.

[38] Christoph Dankert et al. “Exact and approximate unitary 2-designs and their application to fidelity estimation”. In: *Physical Review A* 80.1 (July 6, 2009), p. 012304. DOI: `10.1103/PhysRevA.80.012304`.

[39] Steven T. Flammia and Joel J. Wallman. “Efficient Estimation of Pauli Channels”. In: *ACM Transactions on Quantum Computing* 1.1 (Dec. 9, 2020), pp. 1–32. DOI: `10.1145/3408039`.

[40] Eric Dennis et al. “Topological quantum memory”. In: *Journal of Mathematical Physics* 43.9 (Sept. 2002), pp. 4452–4505. DOI: `10.1063/1.1499754`.

[41] Andrew J. Landahl, Jonas T. Anderson, and Patrick R. Rice. “Fault-tolerant quantum computing with color codes”. In: (2011). Publisher: arXiv Version Number: 1. DOI: `10.48550/ARXIV.1108.5738`.

[42] Austin G. Fowler. “Optimal complexity correction of correlated errors in the surface code”. In: (2013). Publisher: arXiv Version Number: 1. DOI: `10.48550/ARXIV.1310.0863`.

[43] Austin G. Fowler and John M. Martinis. “Quantifying the effects of local many-qubit errors and nonlocal two-qubit errors on the surface code”. In: *Physical Review A* 89.3 (Mar. 12, 2014). Publisher: American Physical Society, p. 032316. DOI: `10.1103/PhysRevA.89.032316`.

[44] Charles Kittel. *Introduction to Solid State Physics*. 8th ed. Wiley, 2004.

[45] K. Serniak et al. “Hot Nonequilibrium Quasiparticles in Transmon Qubits”. In: *Physical Review Letters* 121.15 (Oct. 10, 2018). Publisher: American Physical Society, p. 157701. DOI: `10.1103/PhysRevLett.121.157701`.

[46] John M. Martinis, M. Ansmann, and J. Aumentado. “Energy Decay in Superconducting Josephson-Junction Qubits from Nonequilibrium Quasiparticle Excitations”. In: *Physical Review Letters* 103.9 (Aug. 26, 2009). Publisher: American Physical Society, p. 097002. DOI: `10.1103/PhysRevLett.103.097002`.

[47] Antti P. Vepsäläinen et al. “Impact of ionizing radiation on superconducting qubit coherence”. In: *Nature* 584.7822 (Aug. 27, 2020). Publisher: Springer Science and Business Media LLC, pp. 551–556. DOI: `10.1038/s41586-020-2619-8`.

[48] David M. Pozar. *Microwave engineering*. 4th ed. OCLC: ocn714728044. Hoboken, NJ: Wiley, 2012. 732 pp.

[49] Jens Koch et al. "Charge-insensitive qubit design derived from the Cooper pair box". In: *Physical Review A* 76.4 (Oct. 12, 2007). Publisher: American Physical Society, p. 042319. DOI: `10.1103/PhysRevA.76.042319`.

[50] Chad Rigetti and Michel Devoret. "Fully microwave-tunable universal gates in superconducting qubits with linear couplings and fixed transition frequencies". In: *Physical Review B* 81.13 (Apr. 5, 2010), p. 134507. DOI: `10.1103/PhysRevB.81.134507`.

[51] J. Kelly et al. "State preservation by repetitive error detection in a superconducting quantum circuit". In: *Nature* 519.7541 (Mar. 5, 2015). Publisher: Nature Publishing Group, pp. 66–69. DOI: `10.1038/nature14270`.

[52] Charles Neill. "A path towards quantum supremacy with superconducting qubits". PhD thesis. University of California, Santa Barbara, 2017.

[53] Fei Yan et al. "Tunable Coupling Scheme for Implementing High-Fidelity Two-Qubit Gates". In: *Physical Review Applied* 10.5 (Nov. 28, 2018). Publisher: American Physical Society, p. 054062. DOI: `10.1103/PhysRevApplied.10.054062`.

[54] B. Foxen et al. "Demonstrating a Continuous Set of Two-qubit Gates for Near-term Quantum Algorithms". In: *Physical Review Letters* 125.12 (Sept. 15, 2020). _eprint: 2001.08343, p. 120504. DOI: `https://doi.org/10.1103/PhysRevLett.125.120504`.

[55] Daniel Sank. "Fast, accurate state measurement in superconducting qubits". PhD thesis. University of California, Santa Barbara, 2014. URL: `https://www.alexandria.ucsb.edu/lib/ark:/48907/f3w0942t`.

[56] Eyob A. Sete, John M. Martinis, and Alexander N. Korotkov. "Quantum theory of a bandpass Purcell filter for qubit readout". In: *Physical Review A* 92.1 (July 21, 2015). Publisher: American Physical Society, p. 012325. DOI: `10.1103/PhysRevA.92.012325`.

[57] B Foxen et al. "Qubit compatible superconducting interconnects". In: *Quantum Science and Technology* 3.1 (Jan. 1, 2018). _eprint: 1708.04270, p. 014005. DOI: `https://doi.org/10.1088/2058-9565/aa94fc`.

[58] Alysson Gold et al. "Entanglement across separate silicon dies in a modular superconducting qubit device". In: *npj Quantum Information* 7.1 (Dec. 2021), p. 142. DOI: `10.1038/s41534-021-00484-1`.

[59] Sergey Bravyi et al. "The Future of Quantum Computing with Superconducting Qubits". In: (2022). Publisher: arXiv Version Number: 1. DOI: `10.48550/ARXIV.2209.06841`.

[60] Julian Kelly et al. "Physical qubit calibration on a directed acyclic graph". In: (2018). Publisher: arXiv Version Number: 1. DOI: `10.48550/ARXIV.1803.03226`.

[61] Paul V. Klimov et al. “The Snake Optimizer for Learning Quantum Processor Control Parameters”. In: (2020). _eprint: 2006.04594. DOI: `https://doi.org/10.48550/arXiv.2006.04594`.

[62] J. Kelly et al. “Scalable *in situ* qubit calibration during repetitive error detection”. In: *Physical Review A* 94.3 (Sept. 26, 2016), p. 032321. DOI: `10.1103/PhysRevA.94.032321`.

[63] H. Bombin. “Topological subsystem codes”. In: *Physical Review A* 81.3 (Mar. 3, 2010), p. 032301. DOI: `10.1103/PhysRevA.81.032301`.

[64] Sergey Bravyi et al. “Subsystem surface codes with three-qubit check operators”. In: (2012). Publisher: arXiv Version Number: 2. DOI: `10.48550/ARXIV.1207.1443`.

[65] Barbara M. Terhal. “Quantum error correction for quantum memories”. In: *Reviews of Modern Physics* 87.2 (Apr. 7, 2015). Publisher: American Physical Society, pp. 307–346. DOI: `10.1103/RevModPhys.87.307`.

[66] Christian Kraglund Andersen et al. “Repeated quantum error detection in a surface code”. In: *Nature Physics* 16.8 (Aug. 2020). Publisher: Springer Science and Business Media LLC, pp. 875–880. DOI: `10.1038/s41567-020-0920-y`.

[67] Jerry M. Chow et al. “Implementing a strand of a scalable fault-tolerant quantum computing fabric”. In: *Nature Communications* 5.1 (Sept. 2014). Publisher: Nature Publishing Group, p. 4015. DOI: `https://doi.org/10.1038/ncomms5015`.

[68] A. D. Córcoles et al. “Detecting arbitrary quantum errors via stabilizer measurements on a sublattice of the surface code”. In: *Nat. Commun.* 6 (2014). Publisher: Nature Publishing Group, p. 6979. DOI: `https://doi.org/10.48550/arXiv.1410.6419`.

[69] Maika Takita et al. “Demonstration of Weight-Four Parity Measurements in the Surface Code Architecture”. In: *Physical Review Letters* 117.21 (Nov. 18, 2016). Publisher: American Physical Society, p. 210505. DOI: `10.1103/PhysRevLett.117.210505`.

[70] D. Nigg et al. “Quantum computations on a topologically encoded qubit”. In: *Science* 345.6194 (July 18, 2014). Publisher: American Association for the Advancement of Science, pp. 302–305. DOI: `10.1126/science.1253742`.

[71] D. Ristè et al. “Detecting bit-flip errors in a logical qubit using stabilizer measurements”. In: *Nature Communications* 6.1 (Nov. 2015). Publisher: Nature Publishing Group, p. 6983. DOI: `10.1038/ncomms7983`.

[72] M. D. Reed et al. “Realization of three-qubit quantum error correction with superconducting circuits”. In: *Nature* 482.7385 (Feb. 2012). Publisher: Springer Science and Business Media LLC, pp. 382–385. DOI: `10.1038/nature10786`.

[73] Christian Kraglund Andersen et al. “Entanglement stabilization using ancilla-based parity detection and real-time feedback in superconducting circuits”. In: *npj Quantum Information* 5.1 (Dec. 2019). Publisher: Springer Science and Business Media LLC, p. 69. DOI: `10.1038/s41534-019-0185-4`.

[74] F. Motzoi et al. “Simple Pulses for Elimination of Leakage in Weakly Nonlinear Qubits”. In: *Physical Review Letters* 103.11 (Sept. 8, 2009). Publisher: American Physical Society, p. 110501. DOI: `10.1103/PhysRevLett.103.110501`.

[75] Zijun Chen et al. “Measuring and Suppressing Quantum State Leakage in a Superconducting Qubit”. In: *Physical Review Letters* 116.2 (Jan. 13, 2016). Publisher: American Physical Society, p. 020501. DOI: `10.1103/PhysRevLett.116.020501`.

[76] R. Barends et al. “Superconducting quantum circuits at the surface code threshold for fault tolerance”. In: *Nature* 508.7497 (Apr. 2014). Publisher: Nature Publishing Group, pp. 500–503. DOI: `https://doi.org/10.1038/nature13171`.

[77] M. A. Rol et al. “Fast, High-Fidelity Conditional-Phase Gate Exploiting Leakage Interference in Weakly Anharmonic Superconducting Qubits”. In: *Physical Review Letters* 123.12 (Sept. 18, 2019). Publisher: American Physical Society, p. 120502. DOI: `10.1103/PhysRevLett.123.120502`.

[78] V. Negîrneac et al. “High-Fidelity Controlled- Z Gate with Maximal Intermediate Leakage Operating at the Speed Limit in a Superconducting Quantum Processor”. In: *Physical Review Letters* 126.22 (June 4, 2021). _eprint: 2008.07411, p. 220502. DOI: `https://doi.org/10.1103/PhysRevLett.126.220502`.

[79] Sumeru Hazra et al. “Engineering cross resonance interaction in multi-modal quantum circuits”. In: *Applied Physics Letters* 116.15 (Apr. 13, 2020). Publisher: AIP Publishing, p. 152601. DOI: `10.1063/1.5143440`.

[80] Daniel Sank et al. “Measurement-Induced State Transitions in a Superconducting Qubit: Beyond the Rotating Wave Approximation”. In: *Physical Review Letters* 117.19 (Nov. 4, 2016). Publisher: American Physical Society, p. 190503. DOI: `10.1103/physrevlett.117.190503`.

[81] M. D. Reed et al. “Fast reset and suppressing spontaneous emission of a superconducting qubit”. In: *Applied Physics Letters* 96.20 (May 17, 2010). Publisher: AIP Publishing, p. 203110. DOI: `10.1063/1.3435463`.

[82] K. Geerlings et al. “Demonstrating a Driven Reset Protocol for a Superconducting Qubit”. In: *Physical Review Letters* 110.12 (Mar. 20, 2013). Publisher: American Physical Society, p. 120501. DOI: `10.1103/PhysRevLett.110.120501`.

[83] Martin Suchara, Andrew W. Cross, and Jay M. Gambetta. “Leakage Suppression in the Toric Code”. In: *Quantum Inf. Comput.* 15.11 (2014), pp. 997–1016. DOI: `https://doi.org/10.48550/arXiv.1410.8562Focustolearnmore`.

[84] P. Magnard et al. “Fast and Unconditional All-Microwave Reset of a Superconducting Qubit”. In: *Physical Review Letters* 121.6 (Aug. 7, 2018). Publisher: American Physical Society, p. 060502. DOI: `10.1103/PhysRevLett.121.060502`.

[85] C. C. Bultink et al. “Protecting quantum entanglement from leakage and qubit errors via repetitive parity measurements”. In: *Science Advances* 6.12 (Mar. 20, 2020). Publisher: American Association for the Advancement of Science (AAAS), eaay3050. DOI: `10.1126/sciadv.aay3050`.

[86] Boris Mihailov Varbanov et al. “Leakage detection for a transmon-based surface code”. In: *npj Quantum Information* 6.1 (Dec. 2020). Publisher: Springer Science and Business Media LLC, p. 102. DOI: `10.1038/s41534-020-00330-w`.

[87] John M. Martinis and Michael R. Geller. “Fast adiabatic qubit gates using only \sigma z control”. In: *Physical Review A* 90.2 (Aug. 8, 2014), p. 022307. DOI: `10.1103/PhysRevA.90.022307`.

[88] N. A. Sinitsyn, J. Lin, and V. Y. Chernyak. “Constraints on scattering amplitudes in multistate Landau-Zener theory”. In: *Phys. Rev. A* 95.1 (2016). Publisher: American Physical Society, p. 012140. DOI: `https://doi.org/10.48550/arXiv.1609.06285`.

[89] Morten Kjaergaard et al. “Superconducting Qubits: Current State of Play”. In: *Annual Review of Condensed Matter Physics* 11.1 (Mar. 10, 2020), pp. 369–395. DOI: `10.1146/annurev-conmatphys-031119-050605`.

[90] Evan Jeffrey et al. “Fast Accurate State Measurement with Superconducting Qubits”. In: *Physical Review Letters* 112.19 (May 15, 2014). Publisher: American Physical Society, p. 190504. DOI: `10.1103/PhysRevLett.112.190504`.

[91] Austin G. Fowler, Adam C. Whiteside, and Lloyd C. L. Hollenberg. “Towards practical classical processing for the surface code: Timing analysis”. In: *Physical Review A* 86.4 (Oct. 12, 2012). Publisher: American Physical Society, p. 042313. DOI: `10.1103/PhysRevA.86.042313`.

[92] T. E. O’Brien, B. Tarasinski, and L. DiCarlo. “Density-matrix simulation of small surface codes under current and projected experimental noise”. In: *npj Quantum Information* 3.1 (Dec. 2017). Publisher: Nature Publishing Group, p. 39. DOI: `https://doi.org/10.1038/s41534-017-0039-x`.

[93] Ross Shillito et al. “Dynamics of Transmon Ionization”. In: (2022). Publisher: arXiv Version Number: 1. DOI: `10.48550/ARXIV.2203.11235`.

[94] Austin G. Fowler. “Coping with qubit leakage in topological codes”. In: *Physical Review A* 88.4 (Oct. 8, 2013). Publisher: American Physical Society, p. 042308. DOI: `10.1103/PhysRevA.88.042308`.

[95] Yu Zhou et al. "Rapid and unconditional parametric reset protocol for tunable superconducting qubits". In: *Nature Communications* 12.1 (Dec. 2021). Publisher: Springer Science and Business Media LLC, p. 5924. DOI: `10.1038/s41467-021-26205-y`.

[96] Natalie C. Brown and Kenneth R. Brown. "Leakage mitigation for quantum error correction using a mixed qubit scheme". In: *Physical Review A* 100.3 (Sept. 18, 2019). Publisher: American Physical Society, p. 032325. DOI: `10.1103/PhysRevA.100.032325`.

[97] F. Battistel, B.M. Varbanov, and B.M. Terhal. "Hardware-Efficient Leakage-Reduction Scheme for Quantum Error Correction with Superconducting Transmon Qubits". In: *PRX Quantum* 2.3 (July 26, 2021), p. 030314. DOI: `10.1103/PRXQuantum.2.030314`.

[98] L. Cardani et al. "Reducing the impact of radioactivity on quantum circuits in a deep-underground facility". In: *Nature Communications* 12.1 (Dec. 2021). _eprint: 2005.02286, p. 2733. DOI: `https://doi.org/10.1038/s41467-021-23032-z`.

[99] C. D. Wilen et al. "Correlated charge noise and relaxation errors in superconducting qubits". In: *Nature* 594.7863 (June 17, 2021), pp. 369–373. DOI: `10.1038/s41586-021-03557-5`.

[100] Particle Data Group et al. "Review of Particle Physics". In: *Progress of Theoretical and Experimental Physics* 2020.8 (Aug. 14, 2020), p. 083C01. DOI: `10.1093/ptep/ptaa104`.

[101] M. Lenander et al. "Measurement of energy decay in superconducting qubits from nonequilibrium quasiparticles". In: *Physical Review B* 84.2 (July 1, 2011). Publisher: American Physical Society, p. 024501. DOI: `10.1103/PhysRevB.84.024501`.

[102] Antonio D. Córcoles et al. "Protecting superconducting qubits from radiation". In: *Applied Physics Letters* 99.18 (Oct. 31, 2011). Publisher: AIP Publishing, p. 181906. DOI: `10.1063/1.3658630`.

[103] G. Catelani et al. "Quasiparticle Relaxation of Superconducting Qubits in the Presence of Flux". In: *Physical Review Letters* 106.7 (Feb. 16, 2011). Publisher: American Physical Society, p. 077002. DOI: `10.1103/PhysRevLett.106.077002`.

[104] C. Wang et al. "Measurement and control of quasiparticle dynamics in a superconducting qubit". In: *Nature Communications* 5.1 (Dec. 2014), p. 5836. DOI: `https://doi.org/10.1038/ncomms6836`.

[105] K. Karatsu et al. "Mitigation of cosmic ray effect on microwave kinetic inductance detector arrays". In: *Applied Physics Letters* 114.3 (Jan. 21, 2019). Publisher: AIP Publishing, p. 032601. DOI: `10.1063/1.5052419`.

[106] John M. Martinis. "Saving superconducting quantum processors from decay and correlated errors generated by gamma and cosmic rays". In: *npj Quantum Information* 7.1 (Dec. 2021), p. 90. DOI: `10.1038/s41534-021-00431-0`.

[107] M. Houzet et al. “Photon-Assisted Charge-Parity Jumps in a Superconducting Qubit”. In: *Physical Review Letters* 123.10 (Sept. 6, 2019). Publisher: American Physical Society, p. 107704. DOI: `10.1103/PhysRevLett.123.107704`.

[108] Floris A. Zwanenburg et al. “Silicon quantum electronics”. In: *Reviews of Modern Physics* 85.3 (July 10, 2013). Publisher: American Physical Society (APS), pp. 961–1019. DOI: `10.1103/revmodphys.85.961`.

[109] Diego Rainis and Daniel Loss. “Majorana qubit decoherence by quasiparticle poisoning”. In: *Physical Review B* 85.17 (May 30, 2012). Publisher: American Physical Society, p. 174533. DOI: `10.1103/PhysRevB.85.174533`.

[110] Torsten Karzig, William S. Cole, and Dmitry I. Pikulin. “Quasiparticle Poisoning of Majorana Qubits”. In: *Physical Review Letters* 126.5 (Feb. 4, 2021). Publisher: American Physical Society, p. 057702. DOI: `10.1103/PhysRevLett.126.057702`.

[111] I. Nsanzineza and B. L. T. Plourde. “Trapping a Single Vortex and Reducing Quasiparticles in a Superconducting Resonator”. In: *Physical Review Letters* 113.11 (Sept. 12, 2014). Publisher: American Physical Society, p. 117002. DOI: `10.1103/PhysRevLett.113.117002`.

[112] Fabio Henriques et al. “Phonon traps reduce the quasiparticle density in superconducting circuits”. In: *Applied Physics Letters* 115.21 (Nov. 18, 2019). Publisher: AIP Publishing, p. 212601. DOI: `10.1063/1.5124967`.

[113] Matthew B. Hastings and Jeongwan Haah. “Dynamically Generated Logical Qubits”. In: *Quantum* 5 (Oct. 2021). Publisher: Verein zur Forderung des Open Access Publizierens in den Quantenwissenschaften, p. 564. DOI: `10.22331/q-2021-10-19-564`.

[114] Rui Chao et al. “Optimization of the surface code design for Majorana-based qubits”. In: *Quantum* 4 (Oct. 2020). Publisher: Verein zur Forderung des Open Access Publizierens in den Quantenwissenschaften, p. 352. DOI: `10.22331/q-2020-10-28-352`.

[115] C. Ryan-Anderson et al. “Implementing Fault-tolerant Entangling Gates on the Five-qubit Code and the Color Code”. In: (2022). Publisher: arXiv Version Number: 1. DOI: `10.48550/ARXIV.2208.01863`.

[116] K. Ramanathan and N. Kurinsky. “Ionization yield in silicon for eV-scale electron-recoil processes”. In: *Physical Review D* 102.6 (Sept. 28, 2020). _eprint: 2004.10709, p. 063026. DOI: `https://doi.org/10.1103/PhysRevD.102.063026`.

[117] B. G. Christensen et al. “Anomalous charge noise in superconducting qubits”. In: *Physical Review B* 100.14 (Oct. 24, 2019). Publisher: American Physical Society, p. 140503. DOI: `10.1103/PhysRevB.100.140503`.

[118] R. A. Moffatt et al. “Spatial imaging of charge transport in silicon at low temperature”. In: *Applied Physics Letters* 114.3 (Jan. 21, 2019), p. 032104. DOI: `10.1063/1.5049691`.

[119] S. E. de Graaf et al. “Two-level systems in superconducting quantum devices due to trapped quasiparticles”. In: *Science Advances* 6.51 (Dec. 18, 2020). Publisher: American Association for the Advancement of Science, eabc5055. DOI: `10.1126/sciadv.abc5055`.

[120] V. L. Gurevich. *Kinetics of phonon systems.* Moscow: Nauka, 1980.

[121] S. B. Kaplan et al. “Quasiparticle and phonon lifetimes in superconductors”. In: *Physical Review B* 14.11 (Dec. 1, 1976). Publisher: American Physical Society, pp. 4854–4873. DOI: `10.1103/PhysRevB.14.4854`.

[122] P. C. J. J. Coumou et al. “Electrodynamic response and local tunneling spectroscopy of strongly disordered superconducting TiN films”. In: *Physical Review B* 88.18 (Nov. 7, 2013). Publisher: American Physical Society, p. 180505. DOI: `10.1103/PhysRevB.88.180505`.