DEPARTMENT OF ELECTRICAL ENGINEERING
INDIAN INSTITUTE OF TECHNOLOGY MADRAS
CHENNAI – 600036

Cluster State Quantum Computing with Superconducting Circuits

A Thesis

Submitted by

PARTH S SHAH

For the award of the degree

Of

MASTER OF TECHNOLOGY IN QUANTUM SCIENCE AND TECHNOLOGY

June 2022

DEPARTMENT OF ELECTRICAL ENGINEERING
INDIAN INSTITUTE OF TECHNOLOGY MADRAS
CHENNAI – 600036

Cluster State Quantum Computing with Superconducting Circuits

A Thesis

Submitted by

PARTH S SHAH

For the award of the degree

Of

MASTER OF TECHNOLOGY IN QUANTUM SCIENCE AND TECHNOLOGY

June 2022

Everything is interesting if you go into it deeply enough

– Dr. Richard P. Feynman

To my family, friends, and everyone who has shaped me into what I am today

THESIS CERTIFICATE

This is to undertake that the Thesis titled **CLUSTER STATE QUANTUM COMPUTING WITH SUPERCONDUCTING CIRCUITS**, submitted by me to the Indian Institute of Technology Madras, for the award of **Master of Technology in Quantum Science and Technology**, is a bona fide record of the research work done by me under the supervision of **Dr. Peter McMahon**. The contents of this Thesis, in full or in parts, have not been submitted to any other Institute or University for the award of any degree or diploma.

In order to effectively convey the idea presented in this Thesis, figures 2.2, 2.4, 2.5 and 2.6 were used from the works of Wan *et al.* (2021), with the permission of the first author.

Chennai 600036 **Parth S Shah**

Date: July 2022

Dr. Peter McMahon
Research advisor
Assisstan Professor
Department of Applied and Engineering Physics
Cornell University

Dr. Anil Prabhakar
Research co-advisor
Professor
Department of Electrical Engineering
IIT Madras

ACKNOWLEDGEMENTS

All of the work that this thesis describes is the result of the support and guidance provided to me by many others. I would like to thank my guide Prof. Peter McMahon for his advice on various matters, both technical and otherwise, that he gave me during the course of this project. Of course, I am also grateful to him for providing me the opportunity to work at his lab and all the encouragement he provided me with during my time at the McMahon Lab.

I would like to thank my co-guide for the project Prof. Anil Prabhakar for all his feedback and assistance on the project. My previous research experience under his guidance was when I first learned the many aspects of doing scientific research. It was him, along with Prof. Prabha Mandayam, who set up this interdisciplinary dual degree on Quantum Science and Technologies. It provided us students, who were interested in quantum technologies, with the opportunities to explore their interests.

I am extremely grateful to all the members of the McMahon Lab, especially to my mentor, Alen Senanian. Every discussion that I had with him helped shed new perspectives on the issue at hand, and it was only through these, often lengthy, meetings that I could solve the challenges faced during the project. I would also like to thank my lab and project mates: Jeremey Kline, Vladimir Kremenetski, Sridhar Prabhu, Tatsuhiro Onodera, Benjamin Malia, and William Banner (MIT), for all the insightful discussions.

Doing research does not involve only technical difficulties, and I would like to thank my friends Gautham, Dhruv, Mansi, Vighnesh, Vaishnavi, Niharika, Akash, Abhiram and Sathvik for all the moral support they provided me with, despite being miles away. Last but not the least, I would like to thank my brother and my parents for their extremely loving, caring, and supportive nature.

ABSTRACT

KEYWORDS Waveguide QED, Multimode cQED Fault Tolerant, Measurement Based Quantum Computing, Cluster State Quantum Computing, Superconducting Circuits

One of the biggest challenges towards building a large-scale quantum computer is tackling the component overhead involved in protecting qubits from errors. In this work, we address this issue by working in a measurement based quantum computing paradigm, or more specifically, by generating a 3D cluster state of bosonic modes. We present two hardware designs to achieve this. The first uses propagating photonic modes for the data qubits, while using transmons as auxiliary qubits to mediate the interactions between the photons. This approach, based on waveguide quantum electrodynamics, allows us to increase the number of qubits with minimal changes to the design: only the effective length of the waveguide would have to change. The second design is based on multimodal circuit quantum electrodynamics by coupling a superconducting qubit to a multimodal resonator. This method presents a way to demonstrate generation of smaller cluster states with currently achievable parameters for the hardware. We hope that these designs may pave the way for the first experimental demonstrations of fault-tolerant cluster state computing on the superconducting platform.

CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

Quantum computing has emerged as a solution to tackle problems which are beyond the reach of conventional, classical methods of computation. Such problems include quantum chemistry, combinatorial optimization tasks, cryptography, and even novel machine learning tasks. Nevertheless, as with any nascent technology, there are a number of challenges to overcome before quantum computers become practically useful. Among these, a key challenge today is increasing the number of qubits that we can use on a device. A major hurdle in doing so is the overhead associated with each qubit that we add such as the wiring and calibration equipment for its measurement and control. This work focuses on tackling this particular issue and aims to reduce such overhead as we develop systems with more qubits.

1.1 MOTIVATION

But why do we want to increase the number of qubits? There are estimates (Greene, 2018) that 100 logical qubits could easily outperform all the current supercomputers in the world combined. While there are devices now that host over a 100 physical qubits such as IBM's Eagle processor, there is a distinction to be made between physical qubits and logical qubits. Each physical qubit is inherently affected by noise which makes it extremely difficult to be used for practical scenarios as is. It is to protect them from noise and make a logical qubit that the concept of quantum error correction was developed.

Quantum error correction has formalised the hardware requirements of a device based on the logical representation of any given quantum algorithm. Using the tools of quantum error correction, one can arrive at thresholds for the physical error rates and estimate the number of qubits that would be required to represent a certain number of logical qubits.

The main idea behind quantum error correction is to encode the logical qubits in multiple physical qubits so as to redundantly store the data. If we were to encode the data in a single qubit, any local noise would adversely affect it. But by storing the data across multiple qubits, only non-local noise can actually corrupt the data. This redundant data storage comes at the additional overhead of extra qubits, which is why we need to scale up.

If one can perform operations on qubits while dealing with the adverse affects of noise, then we deem such qubits as "fault-tolerant." Of course, there are technicalities to this, such as the the tolerance to the noise being only up to a certain threshold, but the point here is that the error is lesser than what the operation would have if we simply used a single physical qubit rather than a logical qubit.

1.2 PARADIGM CHOICE

Reaching such fault-tolerant regimes of operation is the motivation to develop systems with more qubits. In fact, it is estimated that even for a single logical qubit to be fault-tolerant, we would need about 400 physical qubits (Fowler *et al.*, 2012). While there are numerous ways of approaching this challenge of scaling up to fault-tolerance, the focus of this thesis is to reduce the additional component overhead that was described earlier. To this end, we make use of the measurement-based quantum computing paradigm, also known as the cluster state quantum computing model.

But simply working in this formalism would not really do us any good. As we explain in Chapter 2, even with cluster state quantum computing one would need atleast the same order of number of qubits as in a surface code. What enables us to reduce the overhead is our time-multiplexed approach, which is what the measurement-based quantum computing model is especially suited towards. Since the qubits are to be measured out sequentially, we can time-multiplex the measurement of the qubits such that we require just one measurement chain. In our design what we will see is a train of pulses each representing a single data qubit, while we entangle these pulses together using a constant number of stationary, auxiliary qubits.

1.3 CHOICE OF HARDWARE

Having chosen a particular paradigm, we now turn to our choice of hardware. There are currently a number of competing hardware platforms which are viable candidates for future quantum technologies. Of these, superconducting circuits have gained tremendous traction in both academic and industrial research. Possibly the foremost reason is its compatibility with existing CMOS fabrication technologies. This has allowed for great design exploration and flexibility in parameter space. This in turn has enabled the research community to reach regimes of light-matter interactions that are hard to reach otherwise, and has also enabled fast gate speeds . Another important aspect of these circuits is how simulations and analysis of such devices can also be done using a lot of existing softwares. Broadly speaking, these circuits are simply microwave circuits working in an extremely low-loss setting. This low loss regime is also the reason that the coherence times of qubits on these devices are good enough to actually implement small quantum circuits. Recently these values are being pushed to the limits and has even reached the 1ms mark(Somoroff *et al.*, 2021).

All these features of superconducting circuits have made it a promising candidate for large scale quantum computation. Industry giants such as Google and IBM have demonstrated the tremendous capabilities of such devices in the circuit model of quantum computation. Recently, there has also been a growing interest in its applications for the measurement based model by generating large entangled structures with microwave photons (eg: Besse *et al.* (2020)).

A subtle point about the latter remark is how such research is beginning to use hybrid architectures: using microwave modes, travelling or stationary, as data qubits while using stationary auxiliary qubits, which are superconducting qubits, to enact gates between the data qubits. This allows one to use the larger coherence times of photonic modes, while enabling high-speed gates using superconducting qubits. This is, in fact, what we make use of as well. Our data is stored in photonic modes which are travelling modes in the case of the first design and resonant modes in the case of the second. We use transmons to enact gates between the different modes in both the designs.

1.4 ORGANISATION OF THESIS

The remainder of this thesis is organized as follows:

- Chapter 2 goes over the basics of cluster state quantum computing. It also describes the particular cluster state we are targeting, and then delves into a particular algorithm for generating it in a fault-tolerant manner.

- Chapter 3 introduces the basic elements of superconducting circuits specifically for our designs. This includes the transmon, and also how it couples to a resonator. Here, we also introduce parametric flux modulation, a key component to both of our designs.

- Chapter 4 looks at the first design that we propose, which makes use of concepts developed in waveguide Quantum Electrodynamics (QED). We first introduce this framework, then describe our design and present the simulations and calculations done to verify the feasibility of this design.

- Chapter 5 describes our second design, which draws on concepts from multimodal circuit QED. This design seems to be more realisable in the short term, and may very well be the first proof-of-concept, experimental demonstration of 3D cluster state generation with superconducting circuits.

- Chapter 6 concludes the thesis with implications of this work, comparison of the two approaches, and a brief discussion on future directions.

CHAPTER 2

CLUSTER STATE QUANTUM COMPUTING

The circuit model is probably the most well-known model for quantum computing, especially popularized by industry giants like Google, IBM etc. This is not without reason though: the circuit-like depiction is very reminiscent of conventional electrical circuits and other visual tools such as the Bloch sphere further aid in understanding the impact of a gate on qubits. Furthermore, there are many analogues that can be drawn between quantum gates and classical gates, even though they work quite differently.

That being said, it is certainly not the only paradigm for quantum computing. Another equivalent model, as proven by Raussendorf and Briegel (2001), is the cluster state quantum computing model, also known as measurement based quantum computing or the "one-way quantum computer," for reasons that will soon be apparent. They are equivalent in the sense that one can perform any quantum algorithm within either framework, and hence both are universal. Despite this, there are differences in the way the actual computation is carried out in each.

2.1 BASICS OF CLUSTER STATE QUANTUM COMPUTING

In cluster state quantum computing, the starting resource is a specific entangled state of qubits which is called a cluster state or a graph state. For any given undirected graph $G = (E, V)$ (E being the set of edges and V the set of vertices) we can define a cluster state as:

$$|\psi_G\rangle := \prod_{(i,j)\in E} Z_{i,j} \otimes_{i'} |+\rangle_{i'} \tag{2.1}$$

In the above equation, $Z_{i,j}$ represents a controlled Z-gate between qubits i and j, with a qubit being placed on every vertex of the graph. These states can be described in the

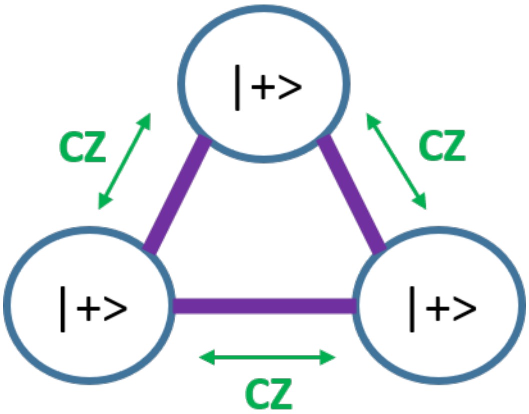


Figure 2.1: An example of generating a cluster state from an underlying graph. In this case, we consider a graph in the shape of a triangle. To generate the corresponding cluster state, consider each vertex to be a qubit initialised in the $|+\rangle$ state. Then apply Controlled-Z gates between each pair of qubits that are connected by an edge. The resulting state is a cluster state.

stabilizer formalism too, as shown by Raussendorf *et al.* (2003).The set of stabilisers for the graph G defined above is given by $\{S_i : i \in V\}$, with

$$S_i := X_i \prod_{j:(i,j)\in E} Z_j \tag{2.2}$$

Given that one can define a cluster state based on any graph, it is fairly obvious that there an infinite number of cluster states that one may consider to use. Based on the graph, cluster states can also be classified into different types. For example, cluster states can be classified based on their dimensionality, and as it turns out, different dimensionalities have different representative power in terms of what they can computer. A 1D cluster state, i.e. a linear chain of qubits, is enough to perform the equivalent of any single qubit operation in the gate circuit model. A 2D cluster state based on a grid on the other hand is a universal quantum computational resource. Moving another dimension higher to a 3D lattice, Raussendorf and Briegel (2001) show that using the body-centred cubic (bcc, see fig.2.2, 2.4 for unit cell) lattice as the graph state provides topological fault tolerance along with universality. In fact, this is the exact state we are interested in and aim to

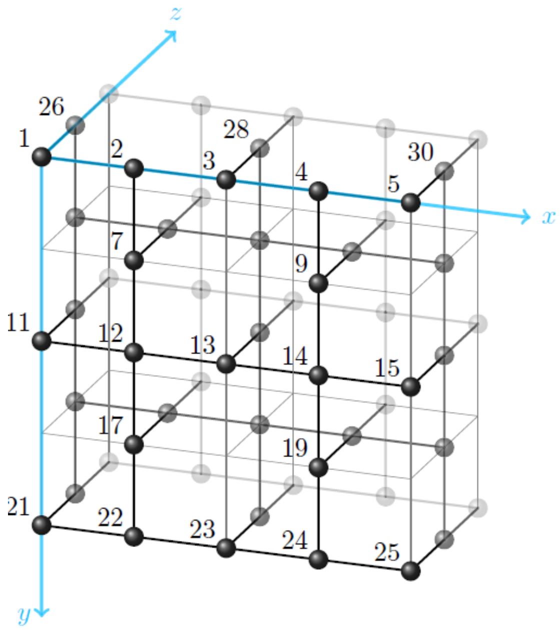


Figure 2.2: 3D cluster state for fault-tolerant quantum computation. The indices actually indicate the order in which the qubits are generated in our scheme, as well as in (Wan *et al.*, 2021)

generate using our proposed architectures.

In all these scenarios, after having been given the cluster state as an initial resource, computation is carried out by simply measuring out qubits sequentially in different bases. Measuring in the Z basis simply removes the qubit from the cluster state, while actual "gates" are applied by measuring in the X-Y plane of the Bloch sphere. The exact basis that a qubit is measured in depends on the outcome of the previous measurements. Note that even two qubit gates do not require simultaneous measurements of different qubits; sequential single qubit measurements suffice for universality. This is also the reason it is called a "one-way quantum computer." Once the resource state is created, we can sequentially measure out each qubit, at the end of which the resource state is lost. This actually makes it very suitable for architectures that use time-multiplexed travelling modes as qubits rather than stationary modes, since we no longer waste space on qubits that are not needed.

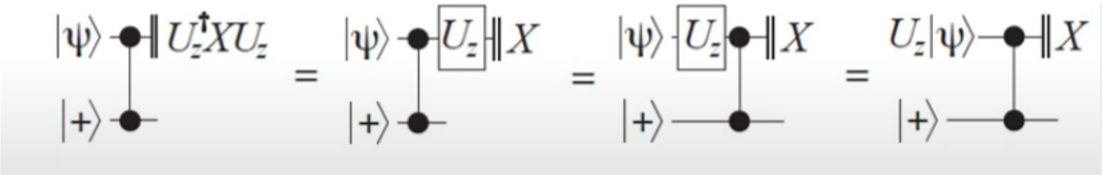


Figure 2.3: Circuit diagram of quantum half-teleportation. It can be seen that measuring in a different basis effectively causes a rotation about the Z-axis of the Bloch sphere. One can trace the evolution of the lower qubit to see that the final state is $HZ^sU_z\left|\psi\right\rangle$. s is the measurement outcome. By applying the circuit twice, i.e. applying the same circuit to the lower qubit again, one can achieve a rotation about the X axis as well. Image from Raussendorf.

There is a nice, somewhat intuitive understanding of how such measurements produce a gate on a qubit. Consider the case of the 1D cluster state. One can imagine a "simulated axis of time" along the length of the chain (from left to right for example). What this means is that as we keep measuring out qubits from left to right, the state that the next qubit ends up in is some rotated version of the initial state. This can be understood by the "half-quantum" teleportation circuit (see Fig. 2.3), which effectively does a rotation about the X axis. Applying the same circuit twice leads to a rotation about the Z axis. With rotations possible about 2 of the axis in the Bloch sphere, one can implement any single qubit gate. Two qubit gate requires a more detailed analysis, but the essential idea is the same (Raussendorf and Briegel (2001) provide more details on the implementation of gates). A key point to note though is that even in the case of two qubit gates we only need sequential single qubit measurements.

This analogy of one axis being that of time actually transfers to higher dimensional cluster states as well. In the 2D case, one axis signifies that of time, and the other is proportional to the total number of logical qubits being used in the circuit. In the 3D case, one can imagine that a slice of the lattice, perpendicular to the direction of time, encodes the logical qubits in some error corrected way, which can be considered a surface code. This is why the 3D cluster state is also considered an exfoliated version of the surface code.

A feature of the cluster state quantum computing model that may be considered disadvantageous is that it seems to require a lot more physical qubits to encode logical qubits than the circuit model. The reason for this apparent flaw is that naively one may expect that we need to store the entire cluster state at all times. For example in the 3D cluster state, we might need to store the entire lattice at all times, including the dimension along simulated time. But this is, in fact, not true and the reason is this analogy of having one axis as that of simulated time. Once all the interactions of a qubit are done with its neighbours in the lattice, which is to say that all the CZ gates on that qubit are done to build the cluster state, we can in principle measure it out immediately. Just before measuring out a qubit in a slice at time i, we can enact the CZ gates between this qubit and its neighbours in the next slice for time $i + 1$. By doing so, we only need to store one slice at a time, which drastically reduces the number of physical qubits required. In the 3D cluster state in fig. 2.2, this boils down to needing approximately the same order of qubits as a surface code.

2.2 BUILDING THE CLUSTER STATE

Although a 3D cluster state is universal and fault tolerant, it is still a challenge to actually build the state itself in a fault tolerant manner. This is what is addressed by Wan *et al.* (2021). We make use of the algorithm they provide, but use a different hardware architecture from what they propose. The scheme outlined in the paper envisions a single quantum emitter interacting with travelling bosonic modes in a waveguide (see fig. 2.5). The bosonic modes are the actual "data" qubits, while the emitter is an "auxiliary" qubit that enables the interactions between the data qubits. While our first architecture seems similar in design to this, there are clear distinctions in how it operates, as we will describe in Chapter 4. Our second design works with resonator modes rather than travelling modes, so that is entirely different from the proposed scheme by Wan *et al.* (2021). Nevertheless, the quantum circuit (in the circuit model representation, shown in fig. 2.6) used in either case remains the same.

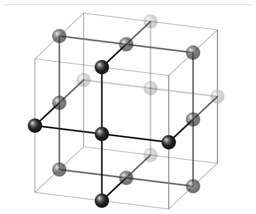


Figure 2.4: BCC lattice elementary cell. The bold lines indicate cluster state edges, while the spheres are qubits.

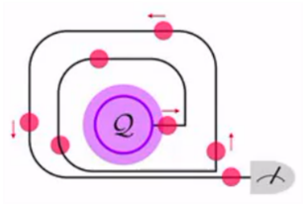


Figure 2.5: Schematic of proposal by Wan *et al.* (2021). A single quantum emitter interacts twice with the photons after it emits them to implement a cluster state.

The paper lays out two possible algorithms to achieve the same final bcc lattice. The first creates a simple cubic lattice and then measures out the unnecessary qubits to create the final bcc lattice. This allows for a very uniform and periodic structure in the algorithm without intermediate measurements. The second algorithm directly creates the final states, but requires intermediate measurements. The authors also show that the second algorithm has a higher tolerance to errors in the creation of the state. Yet, we focus on the first algorithm because it would be easier to implement on a physical design. The circuit for such a scheme would look as shown in fig. 2.6. Note that in this case we actually create a cuboidal lattice, but to generate the lattice in fig. 2.2, we simply measure out, in the Z basis, the qubits that are not required.

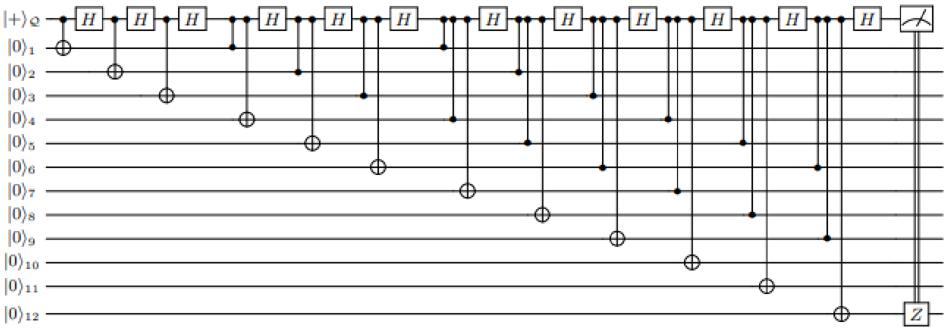


Figure 2.6: Circuit to implement the first algorithm as given by Wan *et al.* (2021). Circuit shows implementation for a 2x3x2 lattice, shown in Fig. 2.7

The main principle behind this circuit behaving fault tolerantly is that it maintains a cluster state at all times. In this way, any errors that may occur doing the creation of the circuit can be shown to be localised errors in the final state, which the underlying surface code can correct.

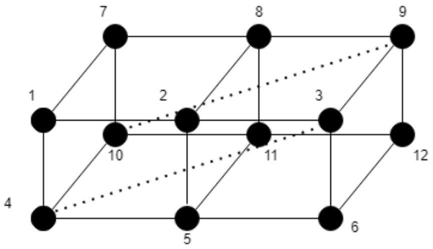


Figure 2.7: Lattice generated by circuit shown in Fig. 2.6. Some additional connections which are unwanted are created due to the nature of the circuit (including a connection between qubits 6 and 7 which is not shown for sake of clarity). These are taken care of by measuring out the qubits on the edges usually. Hence, we usually generate a larger cuboidal lattice than the size of the actual cluster state we want.

The authors of (Wan *et al.*, 2021) also provide some proposals on how to realise the necessary gates. They show that the two qubit Controlled-Not (CNOT) gate can be implemented by simple emission after having initialised the emitter in the right state. The exact pulse scheme depends on whether we choose to implement a single rail or dual rail scheme. The Controlled-Z (CZ) gate is shown to be done with a resonant scattering against the same emitter. This is where our first design distinguishes itself from their proposal. The authors have assumed a chiral waveguide in their approach, but this does not seem to be viable enough with current technologies. Without the chirality, there is in fact no transmission of a photon that is resonant with the excitation of the emitter. This is one challenge that we had to overcome while designing our first physical circuit, which we elaborate upon in the Chapter 4.

CHAPTER 3

SUPERCONDUCTING CIRCUITS

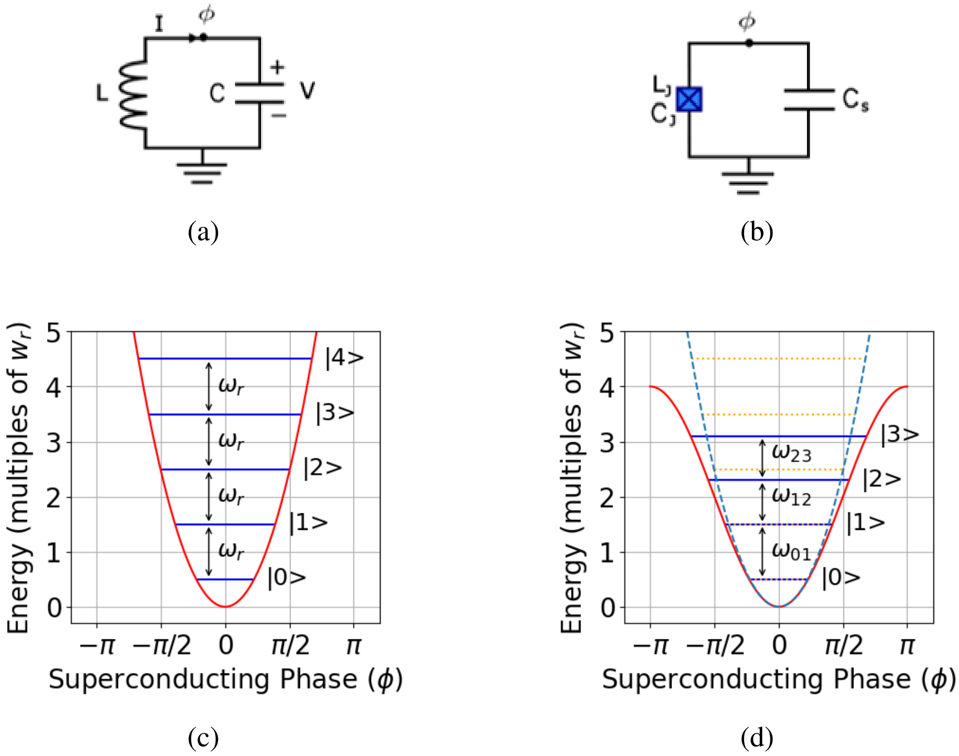
Before getting into the details of our designs, it is worth reviewing the basics of superconducting circuits and circuit Quantum Electrodynamics (cQED). As mentioned in Chapter 1, this particular platform has attracted a lot of attention in the past few years and is a leading candidate for future quantum computers. Since its fabrication process is CMOS compatible, it makes it easy to design fabricate the chips and provides flexibility in design space. It has also been shown that quantum gates can be implemented very fast, mostly due to the high interaction strengths that can be achieved. All this has made it a popular choice for quantum computing hardware.

A superconducting circuit is actually not too different from a conventional microwave circuit. By running the experiment in a superconducting regime, one essentially reduces the loss to a level where we can observe quantum effects. A major component that is more specific to quantum computing circuits is the Josephson junction which is a non-linear element. It allows one to build a variety of components crucial to quantum circuits, especially "artificial atoms." While there are numerous such artificial atoms that are being researched, the most common of these is the transmon.

3.1 THE TRANSMON

The transmon is the most popular artificial atom used in the cQED setting. It is simple in design, yet provides enough anharmonicity to allow us to selectively address the lowest levels to implement a qubit. To understand the transmon, it is instructive to first consider the LC oscillator. In the classical case the Hamiltonian of the system is simply given by:

$$H = \frac{1}{2}CV^2 + \frac{1}{2}LI^2 \tag{3.1}$$



(a)

(b)

(c)

(d)

Figure 3.1: Schematics of an LC oscillator (a) vs. a Transmon(b). The QHO is the quantized version of a normal LC oscillator and shows a quadratic potential (c). The transmon has an anharmonicity which can be clearly seen in the uneven level spacing (d). The lowest two levels of the transmon are generally used as the computational $|0\rangle$ and $|1\rangle$. This anharmonicity is introduced by the Josephson Junctiond(depicted by a box with a cross). The orange dotted line represents the approximate quadratic potential of the transmon.

where L and C are the inductance of the inductor and capacitance of the capacitor, respectively. V and I represent the voltage and current in the circuit as shown in figure 3.1. One can then move into the quantum limit and in this case the Hamiltonian turns out to be:

$$\hat{H} = 4E_C\hat{n}^2 + \frac{1}{2}E_L\hat{\phi}^2 \tag{3.2}$$

with $E_C = e^2/(2C)$ and $E_L = (\Phi_0/2\pi)^2/L$ are representative of the capacitive and inductive energies (Φ_0 represents the superconducting flux quantum). $\hat{n}$ is the reduced charge operator (representative of the number of Cooper pairs), and $\hat{\phi}$ is the reduced flux operator (representative of the flux across the inductor). Performing second quantization using the creation ($\hat{a}^\dagger$) and annhilation ($\hat{a}$) operators with $\hat{n} \propto (\hat{a} - \hat{a}^\dagger)$ and $\hat{\phi} \propto (\hat{a} + \hat{a}^\dagger)$:

$$\hat{H} = \hbar\omega_r(\hat{a}^\dagger\hat{a} + \frac{1}{2}) \tag{3.3}$$

where $\omega = 1/\sqrt{LC}$. In this case we arrive at the usual quantum harmonic oscillator with the energy spacing equal between every adjacent levels (see fig.3.1). Since they are equally spaced, it is difficult to selectively address a transition from one state to the next. This is where the josephson junction comes into play. It essentially acts as a non-linear, nearly dissipationless inductor which changes the inductive part of the Hamiltonian:

$$\hat{H} = 4E_C\hat{n}^2 - E_J\cos(\hat{\phi}) \tag{3.4}$$

with $E_C = e^2/(2(C_s + C_J))$ and $E_J = I_c\Phi_0 2\pi$. The josephson junction has a self-capacitance C_J and a critical current I_c (Krantz *et al.*, 2019). This non-linear modification of the Hamiltonian results in a energy spectrum which is no longer equally spaced like the quantum harmonic oscillator (see fig.3.1). Instead, we see an anharmonicity which is negative in the case of the transmon. This allows us to selectively drive transitions between the lowest energy levels, the bottom two of which are used for the qubit in general.

The preceding description is actually a generic description of the charge qubit. The transmon is a charge qubit which is operated in the regime with $E_J >> E_C$. In the case where these two quantities are comparable, as with the Cooper-pair box, the qubit becomes very sensitive to charge noise (fluctuations in the cooper pair occupation) and reduces the coherence times. Tackling charge noise turned out to be harder than doing so for flux noise, and hence the community has moved towards this regime instead.

One can use equation 3.4, expand the cosine with a taylor expansion and then second quantize the Hamiltonian after truncating the expansion. To fourth order the Hamiltonian resembles a Duffing oscillator:

$$\hat{H}/\hbar = \omega_q \hat{a}^\dagger \hat{a} + \frac{\alpha}{2} \hat{a}^\dagger \hat{a}^\dagger \hat{a} \hat{a} \tag{3.5}$$

where ω_q is the qubit frequency given by $(\sqrt{8E_j E_C} - E_C)/\hbar$, and $\alpha = -E_C$ is the difference between the transition frequencies from the ground to first excited level and the first to second excited level. Typical values for these parameters include 3-6GHz for the qubit frequency, and 100-300MHz of anharmonicity. This level of anharmonicity is sufficient to individually address just the lowest two levels, though in our case we also use the second excited level to implement gates.

3.2 SINGLE QUBIT GATES

In most superconducting devices, single qubit gates are applied using the charge drive. We follow this same technique too for the Hadamard gate in 2.6. Applying a charge drive is similar to using a voltage source capacitively coupled to the transmon (fig. 3.2). The Hamiltonian of a transmon with a charge drive can be simplified to (Krantz *et al.*, 2019):

$$\hat{H} = \omega_q \hat{a}^\dagger \hat{a} + \frac{\alpha}{2} \hat{a}^\dagger \hat{a}^\dagger \hat{a} \hat{a} + i\Omega V_d(t)(\hat{a} - \hat{a}^\dagger) \tag{3.6}$$

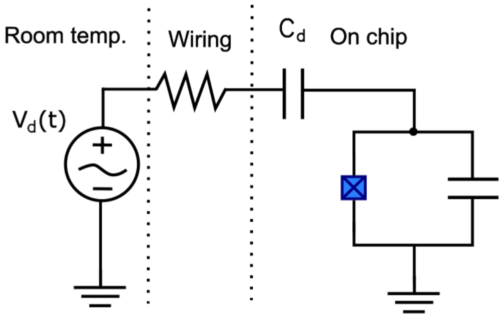


Figure 3.2: Circuit diagram equivalent of a charge drive connected to a transmon. C_d is a capacitor used to couple the charge line to the qubit, on-chip. The drive is a signal generated at room temperature, which then travels through the wiring in a dilution refrigerator to reach the chip.

For the sake of simplicity let us consider just the lowest two levels of the transmon. We can reduce the Hamiltonian in 3.6 to:

$$\hat{H} = -\omega_q \hat{\sigma}_z + \Omega V_d(t)\hat{\sigma}_y \tag{3.7}$$

$V_d(t)$ is the time dependent drive that is applied on the qubit, while Ω is a scaling factor associated with the exact circuit being used. $\hat{\sigma}_z$ and $\hat{\sigma}_y$ are the Pauli Z and Y operators. To clearly see the effect of the drive it helps to move into a rotating frame at the qubit frequency. This takes care of the precession of the qubit about the bloch sphere and leaves us with just the effect of the drive. The unitary to transform into this rotating frame is $\hat{U} = e^{i\omega_q \hat{\sigma}_z t}$. Now we can find the new Hamiltonian in this rotating frame:

$$\begin{aligned} \hat{H}' &= \hat{U}\hat{H}\hat{U}^\dagger + i(\frac{\partial}{\partial t}\hat{U})\hat{U}^\dagger \\ &= \Omega V_d(t)(\cos(\omega_q(t))\hat{\sigma}_y - \sin(\omega_q(t))\hat{\sigma}_x) \end{aligned} \tag{3.8}$$

We can now decompose the drive into its I and and Q quadratures at the qubit frequency ω_q. Let us assume the drive is of the form $s(t)\sin(\omega_d t + \phi)$, with $s(t)$ being an envelope function, ω_d the frequency of the drive and ϕ is a phase. In this case $I = \cos(\phi)$ and $Q = \sin(\phi)$. Equation 3.8 can then be written as:

$$\hat{H}'_d = \frac{1}{2}\Omega s(t)(((-I\cos(\delta\omega t) + Q\sin(\delta\omega t))\hat{\sigma}_x + ((Q\cos(\delta\omega t) - I\sin(\delta\omega t))\hat{\sigma}_y) \tag{3.9}$$

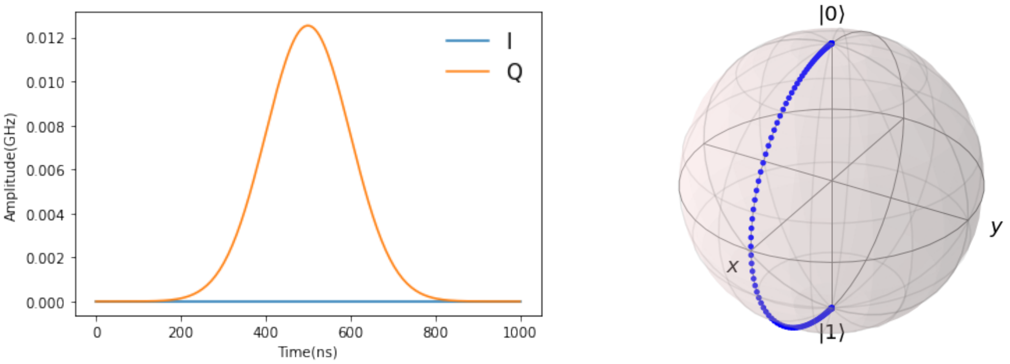
with $\delta\omega = \omega_q - \omega_d$ being the qubit-drive detuning. When the drive is resonant with the qubit we simply get:

$$\hat{H}'_d = -\Omega s(t)(I\hat{\sigma}_x + Q\hat{\sigma}_y) \tag{3.10}$$

Equation 3.10 tells us that the two quadratures of the drive cause rotations about different axes of the Bloch sphere: I causes rotations about the x-axis (R_x gate) while Q works about the y-axis (R_y gate). With access to rotations about two of axes of the Bloch sphere, one can in principle implement any single qubit gate. For example, the angle of rotation about the y-axis for a drive with just an Q component (fig. 3.3) is given by:

$$\Theta(t) = -\Omega \int_0^t s(t')dt' \tag{3.11}$$

In our specific case, we only need to implement the Hadamard gate. To this end, we note that we can decompose a hadamard gate as H = $R_y(\pi/2)$ Z, which is a Z gate followed by a $\pi/2$ rotation about the y-axis. Hence, we only require the Q quadrature. We use the virtual Z gate to implement rotations about the Z axis, which is simply advancing the phase of all the consequent drives by the required angle (Krantz *et al.*, 2019). We could of course decompose the Hadarmard gate into rotations about the X-axis followed by one about Y-axis, but by using the virtual Z gate we eliminate the need to send a second pulse.



(a)

(b)

Figure 3.3: (a) I and Q quadratures of a drive pulse. The Q quadrature is gaussian-shaped, with an area such that the pulse implements a rotation by π about the y-axis.(b) Evolution of a state on the Bloch sphere under the influence of the drive; starts from $|0\rangle$ and rotates about the y-axis to reach $|1\rangle$

3.3 TWO QUBIT GATES USING PARAMETRIC FLUX MODULATION

In the preceding discussion, the qubit frequency was considered to be fixed. But this can in fact be an in-situ tunable parameter. Physically this means using two josephson junctions in parallel and controlling an external flux that flows through the loop between them (Krantz *et al.*, 2019). This tunable nature of the frequency of the transmon actually lets us implement gates between a transmon and a mode coupled to it. Consider a single mode coupled to a tunable frequency transmon (we omit the explicit time-dependence of the operators):

$$\hat{H}/\hbar = \omega_q(t)\hat{a}^\dagger\hat{a} + \frac{\alpha}{2}\hat{a}^\dagger\hat{a}^\dagger\hat{a}\hat{a} + \omega_r\hat{b}^\dagger\hat{b} + g(\hat{b}^\dagger + \hat{b})(\hat{a}^\dagger + \hat{a}) \tag{3.12}$$

$\hat{b}$ is the annhilation operator for the mode at frequency ω_r that is coupled to the transmon, and the bare coupling between the two is g. We use a time dependent qubit frequency by modulating the flux with a mean value of ω_0, and an additional sideband with amplitude ϵ, frequency ω_{sb}, and phase ϕ_{sb}. Hence the qubit frequency looks like:

$$\omega_q(t) = \omega_0 + \epsilon\sin(\omega_{sb}t + \phi_{sb}) \tag{3.13}$$

In such a setting, if $w_{sb} = \omega_0$ and ω_r, i.e. the sideband is at the difference frequency of the mean qubit frequency and the mode, one can drive stimulated rabi oscillations between the two. To see this, we can move into a rotating frame given by (Naik *et al.*, 2017):

$$\hat{U} = \exp[-2\pi i(\omega_0 t - \frac{\epsilon}{2w_{sb}}\cos(\omega_{sb}t))\hat{a}^\dagger\hat{a} - \omega_r\hat{b}^\dagger\hat{b}] \tag{3.14}$$

Hence the Hamiltonian transforms to:

$$\begin{aligned} \hat{H}' &= \hat{U}\hat{H}\hat{U}^\dagger + i(\frac{\partial}{\partial t}\hat{U})\hat{U}^\dagger \\ &= hgJ_0\left(\frac{\epsilon}{2\omega_{sb}}\right)(e^{-i\Delta t\hat{b}^\dagger\hat{a}}) + hg\hat{b}^\dagger\hat{a}(\sum_{m=1}^{\infty}(-1)^m J_m\left(\frac{\epsilon}{2\omega_{sb}}\right)e^{i(m\omega_{sb}-\Delta)t})) + h.c. \end{aligned} \tag{3.15}$$

Δ is the detuning between the mean qubit frequency ω_0 and the mode frequency ω_r. J_n represents the n^{th} order Bessel function. When $\omega_{sb} = \Delta$, one obtains resonant first-order transitions, where the Hamiltonian can be reduced to:

$$\hat{H}' = hgJ_1\left(\frac{\epsilon}{2\omega_{sb}}\right)(\hat{b}^\dagger\hat{a} + \hat{a}^\dagger\hat{b}) \tag{3.16}$$

This is within the rotating wave approximation as well, where we ignore the fast rotating terms in the Hamiltonian (Naik *et al.*, 2017).

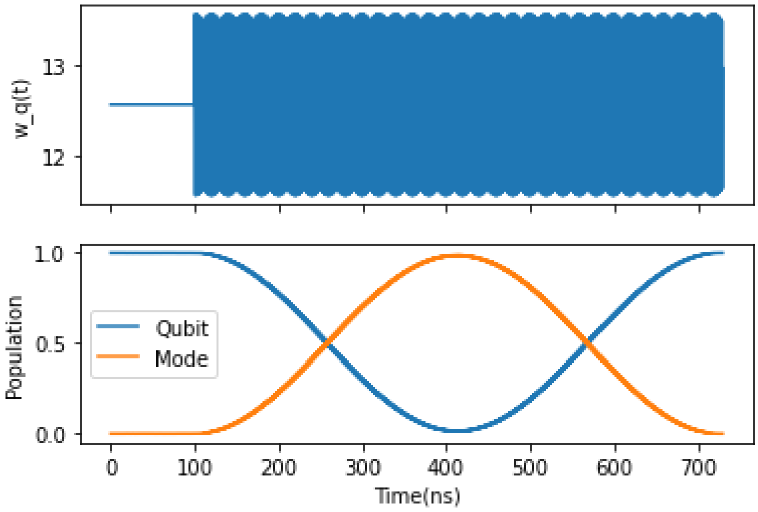


Figure 3.4: Stimualed Rabi oscillations between transmon and mode by sideband flux modulation.

Using this we can implement gates between the mode and the transmon, with an effective coupling rate of $gJ_1\left(\frac{\epsilon}{2\omega_{sb}}\right)$. Using this we can induce stimulated Rabi oscillations between the mode and the transmon as shown in fig.3.4. This analysis can easily be extended to a multimode structure, which is what we use in Chapter 5. We can also use this parametric flux modulation to emit states from a transmon to a waveguide, which is essentially an extension of the described theory to a continuum. By varying the envelope of the sideband in the flux modulation, the pulse shape that is emitted into the waveguide can be controlled. We use this in both of our designs, although we shall not go into the details of this (refer to (Butler, 2022) for more details).

CHAPTER 4

WAVEGUIDE QED BASED DESIGN

4.1 WAVEGUIDE QED

Waveguide QED is a field that has spiked some curiosity fairly recently. In stark contrast to circuit QED approaches of using the bosonic modes of resonators, waveguide QED uses the travelling wave modes of a transmission line or a waveguide as the case may be.

There are many applications of waveguide QED that have been proposed based on the intriguing effects that take place when qubits are coupled to bath of such modes. The demonstration of bright and dark states in a system with 3 transmons coupled to the same transmission line by Kannan *et al.* (2020) clearly demonstrated the interference effects that take place on such a scale. While this experiment largely focused on the states of the transmons themselves, we need to focus on the travelling wave modes instead. There is a lot of work on this particular aspect as well, beginning from the input-output formalism laid out by Gardiner and Collett (1985). More recently, there is the quantum treatment of the input and output pulses as shown by Kiilerich and Mølmer (2019), as well as the temporal mode based treatment by Fischer *et al.* (2018).

What we want to focus on in particular is the scattering of a single photon pulse off of a single emitter. For this purpose we use the master equation as used by Kannan (2018). To start off we consider the transmon as a two-level system coupled to a bath of different modes, which gives us the Hamiltonian:

$$\hat{H} = H_q + H_{wg} + H_{int} \tag{4.1}$$

$$\hat{H}_q = \hbar\omega_q \hat{\sigma}^+ \hat{\sigma}^- \tag{4.2}$$

$$\hat{H}_{wg} = \int d\omega \hbar\omega (\hat{b}_r^\dagger(\omega)\hat{b}_r(\omega) + \hat{b}_l^\dagger(\omega)\hat{b}_l(\omega)) \tag{4.3}$$

$$\hat{H}_{int} = i\hbar g\hat{\sigma}_x \int d\omega\sqrt{\omega}((\hat{b}_r^\dagger(\omega) - \hat{b}_r(\omega) + (\hat{b}_l^\dagger(\omega) - \hat{b}_l(\omega)) \tag{4.4}$$

These equations assume that we are only considering one qubit, at position 0 (the position only matters in the case of multiple qubits). H_q is the Hamiltonian for the qubit with a resonant frequency of ω_q with lowering operator $\hat{\sigma}^-$, H_{wg} that of the waveguide with destruction operator $\hat{b}(\omega)$ at frequency ω , and H_{int} is the interaction Hamiltonian with a coupling strength g, $\hat{\sigma_x}$ being the standard Pauli X operator. We explicitly consider the right and left propagating modes, denoted by the subscript r and l respectively.

To go further, one must make some assumptions, namely the Born-Markov approximation. This gives us the following Lindbladian master equation for the density operator $\hat{\rho}$:

$$\dot{\hat{\rho}} = -\frac{i}{\hbar}[\hat{H}, \hat{\rho}] + \Gamma(\hat{\sigma}^-\hat{\rho}\hat{\sigma}^+ - \frac{1}{2}\{\hat{\sigma}^+\hat{\sigma}^-, \hat{\rho}\}) \tag{4.5}$$

$$\hat{H} = \hbar\omega_q\hat{\sigma}_+\hat{\sigma}^- + \hbar\alpha(t)\hat{\sigma}_x \tag{4.6}$$

Here, Γ defines the decay rate of the qubit into the waveguide, and $\alpha(t)$ is dependent on the drive seen by the qubit. We can assume that the drive is an input from only one side(left) of the transmission line, and hence we assume the form:

$$\alpha(t) = \frac{\Omega}{2}sin(\omega_d(t + \theta)) \tag{4.7}$$

Note that additional dephasing terms can be added to the master equation, but we seek to operate in a regime where this is much smaller than the decay into the waveguide Γ.

To keep track of the travelling wave modes, we turn to input-output theory. We can derive the Heisenberg equations of motions from the above Hamiltonians as (we show just the right propagating fields, but the same is true for the left propagating fields):

$$\dot{\hat{b}}_r(\omega) = -i\hat{b}_r(\omega) + g\sqrt{\omega}\hat{\sigma}_x \tag{4.8}$$

We can then proceed to integrate both sides, and make the assumption that the dynamics of the system are not fast on the time scale that is of interest to us. This lets us arrive at the following equation:

$$\hat{b}_r^{out}(t) = \hat{b}_r^{in}(t) + \sqrt{\frac{\Gamma}{2}}\hat{\sigma}^-(t) \tag{4.9}$$

Here, $\hat{b}_r^{out}(t) = \int \frac{d\omega}{\sqrt{2\pi}}\hat{b}_r(\omega, 0)e^{-i\omega t}$, and $\hat{b}_r^{in}(t) = \int \frac{d\omega}{\sqrt{2\pi}}\hat{b}_r(\omega, t_f)e^{-i\omega t}$, with t_f the final time of interest. Similar equations can be derived for the left propagating modes as well. By taking an ensemble average on both sides, we can relate the state of the qubit to that of that modes in the transmission line.

4.2 REFLECTION OFF TRANSMON

Based on these equations, one can calculate the reflection and transmission coefficients. In the absence of dephasing and non-radiative decay (decay into modes other than the waveguide), an incoming photon that is resonant with the first excitation frequency ω_{ge} of the transmon (ground to excited, $|g\rangle$-$|e\rangle$ transition) will be reflected perfectly. This can be seen in figure 4.1, where we use QuTiP (Johansson *et al.*, 2013) to simulate the master equation for the transmon, and use the input-output relations to determine the reflection and transmission coefficients.

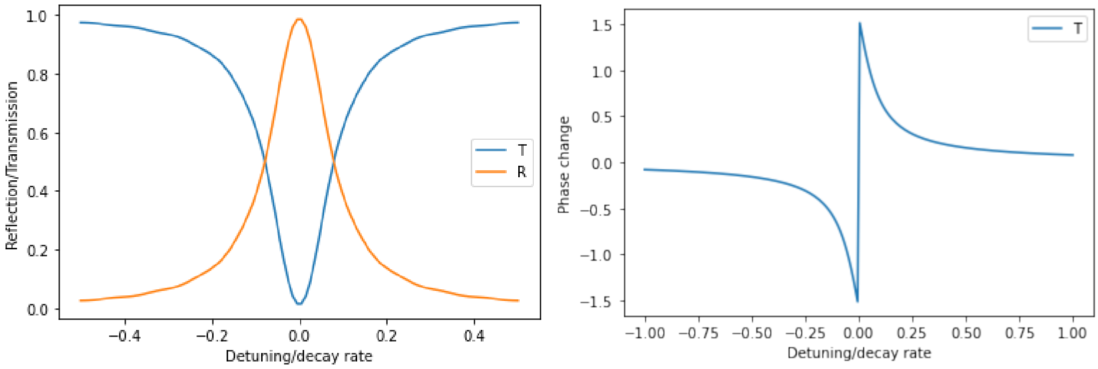


Figure 4.1: Reflection and Transmittivity as a function of the detuning (Δ) of the drive frequency ω_d from the qubit frequency ω_q in the absence of nonradiative decay. As can be seen, the reflection becomes reaches unity as we approach resonance, that is 0 detuning. There is also a phase shift of π in this case (shift shown for transmitted pulse but also exists for reflected pulse)

As we also show, there is a π phase change upon reflection for a resonant photon. This is what we use for implementing the CZ gate. We can observe the same dynamics using the first and second excited level of a transmon, $|e\rangle$ and $|f\rangle$, instead of $|g\rangle$ and $|e\rangle$. In this case too, if the incoming photon is resonant with the $|e\rangle$-$|f\rangle$ transition frequency ω_{ef}, it will be reflected with an additional phase change. Moreoever, since a transmon is an anharmonic oscillator, ω_{ge} and ω_{ef} are well separated. Hence, for a photon at the ω_{ef} frequency the reflection will only occur when the transmon is in the $|e\rangle$ state: when the transmon state is $|g\rangle$ there is no associated resonant transition with the frequency of the incoming photon, i.e. ω_{ef}. Hence, the reflection and the phase shift happen only when the transmon is in state $|e\rangle$ and when there is a photon, which encodes the computational $|11\rangle$ state. In contrast, the state $|00\rangle$, has the transmon in the state $|g\rangle$ and an absence of an incoming photon. The phase change of π causes this state $|11\rangle$ to become $-|11\rangle$, while all the other states have no change. This is essentially the CZ gate, as can be verified by its truth table (shown in table 5.1 in the next chapter).

4.3 EMISSION

For the emission of the pulses into the waveguide we again make use parametric flux modulation. In particular we use a high-pass filter between the waveguide and the emitting transmon. This prevents unwanted decay of the transmon into the waveguide. Additionally, simply letting a transmon decay into the waveguide will produce an exponential pulse shape, which is not ideal for the CZ gate described above. By using parametric flux modulation and shaping the sideband pulse one can emit a gaussian wavepacket instead at a desired frequency (Butler, 2022). The symmetric nature of the wave-packet makes the reflection off the transmon more efficient in practice (Kiilerich and Mølmer, 2019).

As mentioned in Chapter 2, we require that the emission process also be the CNOT gate in our design. This can be done as follows:

1. Excite the transmon from $|e\rangle$ to $|f\rangle$.

2. Emit a photon at ω_p by driving a sideband at $\omega_{ef} - \omega_p$. This will cause the transmon to drop back to $|e\rangle$ while having emitted a photon at the required frequency ω_p. Shape the sideband pulse to create a gaussian wavepacket.

This ensures that only the computational $|1\rangle$ of the transmon, i.e. the physical $|e\rangle$, emits a photon. If the transmon is $|g\rangle$, which is the computational $|0\rangle$, there will be no photon emitted ($|0\rangle$). Additionally, since the excitation in the transmon is preserved (but entangled with the photon) we have created a CNOT gate. Note that this is specific to our design: it is a CNOT gate between the transmon and the waveguide, with the waveguide always starting in the computational $|0\rangle$. At the end of such a sequence, we map a state $(\alpha |0\rangle + \beta |1\rangle) |0\rangle$ to $\alpha |00\rangle + \beta |11\rangle$ (first qubit is transmon, second is photon). We use a similar scheme in our multimode cQED setting, so the reader is encouraged to look at Chapter 5 where we present some results as well.

4.4 THE DESIGN

Using these properties of waveguide QED and the circuit we wish to implement, we arrive at the design shown in 4.2.

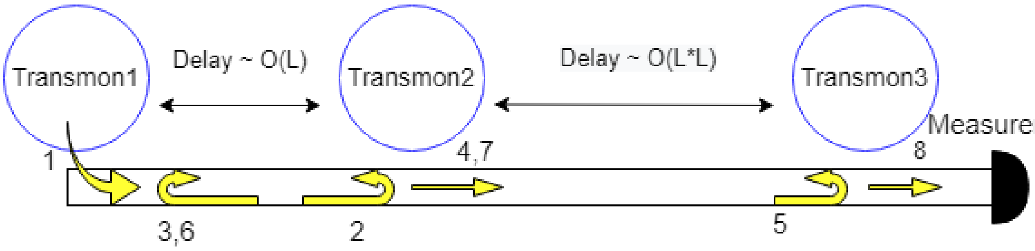


Figure 4.2: Design based on Waveguide QED. L indicates the length of the lattice that we are trying to generate, and is indicative of the total number of logical qubits we are trying to realise. The numbers next to the arrows indicate the sequence of steps a single photon would go through.

The first transmon (Transmon 1) will act as the main emitter for our circuit, while the remaining two (Transmon 2 and 3) will act as tunable "mirrors". The emitter will emit photons at its ω_{ef} frequency. We design the remaining mirrors such that their first transition frequency is $\omega_{ge,2,3} = \omega_{ef,1}$ where the index represents the transmon number.

This allows us to use the mirrors without implementing the CZ gate, but reflection off of Transmon 1 will cause a CZ gate as shown earlier in section 4.2 . Flux modulation provided to the other two transmons will be what provides the control over the reflection. From the waveguide QED results, we can clearly see that the off resonant photons do not get reflected. Hence by changing the flux of the transmons to push them off or on resonance with the frequency of the photons ($\omega_{ef,1}$), we will be able to control transmission and reflection of the incoming photons.

The delays may be implemented via long transmission lines, or for a more scalable approach, tunable delay lines (Ferreira *et al.*, 2021). The lengths of the delay line are dependent on the size of the cluster state. Each qubit in the cluster state has neighbours in 3 different dimensions. The emitter emits photons in succession which creates entanglement in one dimension (such as between qubits 1 and 2 in fig.2.7). The first delay line's length must be such that when a photon returns to interact with the emitter for the first time, the emitter must be in the process of emitting the photon that is entangled in the second dimension with the incoming photon(such as between qubits 1 and 4 in fig.2.7). Similarly, the longer delay line implements the entanglement in the third dimension (such as between qubits 1 and 7 in fig.2.7).

We encode the data in the absence ($|0\rangle$) or presence ($|1\rangle$) of a photon at frequency $\omega_{ef,1}$. This is the single rail encoding. The sequence of interactions of the photon are as shown by the numbers in 4.2:

1. Each photon is first emitted by Transmon 1 (**CNOT gate**). Note that photons are emitted one after another, with a Hadamard gate (2) applied to Transmon1 between each emission. The following steps describe the trajectory of any single photon.

2. After being emitted by Transmon 1, the photon travels towards Transmon 2, at which point, we maintain the flux to make it resonant. Hence, the photon gets reflected and travels back to Transmon 1.

3. Once it reaches Transmon 1, it gets reflected, then travels back to Transmon 2 (**first CZ gate**).

4. This time we modulate the flux of Transmon 2 to keep it off resonance, and hence the photon just passes through.

5. Then it impinges on Transmon 3, where we again cause it to reflect

6. The photon then travels back to Transmon 1 (Transmon 2 is kept off resonant at this point).It is again reflected (**second CZ Gate**).

7. The photon passes past Transmon 2 and Transmon 3 since both are kept off resonant at this point.

8. Lastly, we measure out the photon in the required basis.

The implementation of the various gates have been discussed before. By following the evolution of a single photon, one can see that this scheme implements the exact same circuit as shown in 2.6. The advantage of our a scheme is that we only require a few transmons even for larger number of qubits: all that would need to change is the delay length of the transmission lines.

4.5 DUAL-RAIL SCHEME

The above discussion was entirely on a single-rail implementation wherein the absence or presence of a photon denotes the computational $|0\rangle$ or $|1\rangle$, respectively. Just as in communication technologies, it would be much better to use what is known as a "dual-rail" scheme, where the states $|0\rangle$ or $|1\rangle$ are each encoded in different frequencies. This way photon-loss during the travel through the measurement chain can be detected and faulty measurements can be prevented to some extent.

The dual-rail scheme is definitely harder to implement, but we have investigated ways of modifying our design to do it. In particular, we considered ways of making a tunable mirror for two different frequencies. We use parametric flux modulation to achieve this, by driving a sideband at the difference frequency of the second frequency we wish to reflect. Performing simulations of this requires modifications to the input-output relations presented in 4.9 which we show in appendix A. Some preliminary results on this are shown in figure 4.3.

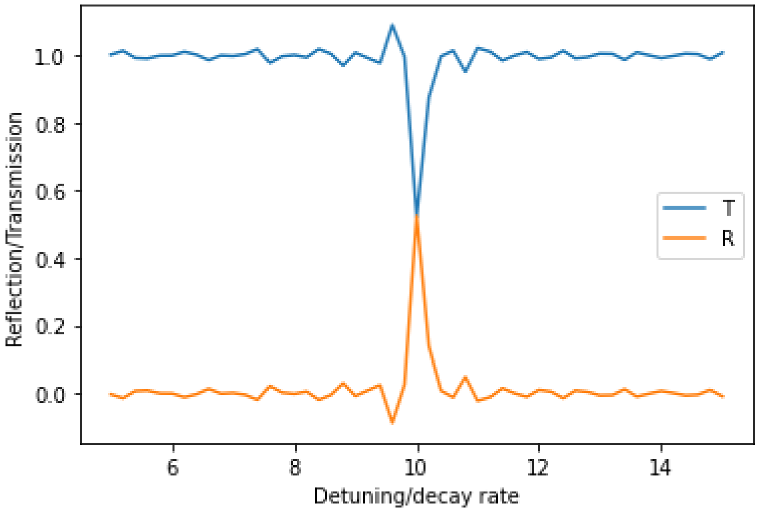


Figure 4.3: Use of a parametric flux drive to cause reflection at a frequency that is detuned from the qubit frequency. In this case, the decay rate of the qubit into the waveguide was 10MHz, while the detuning of the sideband was at 100MHz. As can be seen, there are some fluctuations which arise due to certain approximations made to the integral shown in the appendix A. The reflection is also not perfect which is something we are looking into to correct.

4.6 FEASIBILITY DISCUSSION

So far we've focused on the individual units of our design which are the transmons. What is just as important are the delay lines along which the pulses travel. As we describe below, this is currently the main impediment for the implementation of this design.

In the design above, we mentioned that the larger of the delay lines must have a time delay, or length, proportional to O(L^2) with L being the size of the cluster state we are trying to generate. The exact time delay of the longer delay line is given by $(L^2 - L) * \tau_{p-p}$ where τ_{p-p} is the pulse-to-pulse separation, determined by how fast we can sequentially emit photons from the first transmon. Recent research has demonstrated this value to be in the range of 50-100ns in the optimistic case (Kjaergaard *et al.*, 2020). Now, assume we want just a 3D cluster state of size 3, which means we will have 9 pulses in our circuit at any given time (remember that one dimension of the cluster state is "simulated-time"). Additionally assume that the speed of microwave propagation in superconducting waveguides or transmission lines is 2×10^8m/s. Even for such a small cluster state, the required length of the delay line then turns out to be 120m. This is obviously too large to fit on chip, and is also quite large to fit in a dilution refrigerator. Moving out of the dilution unit would incur additional losses.

One possible way to reduce the length required would be to use "slow-light" metamaterials such as in (Ferreira *et al.*, 2021). But even then, some of the best devices in such a case provide a slow down by a factor of ≈ 400 (Kuzmin *et al.*, 2019). The required length, even for a size 3 cluster state, is still 30cm. This is something we could possibly fit on chip, but it would require great engineering efforts.

Another factor to consider is the loss in the waveguide. As the authors of Wan *et al.* (2021) show, the cluster state can correct for photon-loss, but there is a threshold for this value. The larger the loss in the waveguide, the larger the cluster state needs to be. Wan *et al.* (2021) calculate the required cluster state size for the break-even point (where the logical error rate is lower than the physical error rate) and also the corresponding waveguide loss that the model can handle. This turns out to be 30 qubits (totally requires 900 pulses at a time) with a waveguide loss of 7.4×10^{-4} per time step of their algorithm. They also mention that this loss factor is currently lower than what can be achieved experimentally in the microwave regime.

All these factors made us reconsider our short-term goals and come up with another design. We decided to focus not on reaching a fault-tolerant regime just yet, but instead on the generation of small 3D cluster states as a proof-of-concept. For the delay lines, we started thinking of "catch-and-release" schemes which is where the multimode cQED design was inspired from. We explain this design in more detail in the following chapter.

CHAPTER 5

MULTIMODE CQED DESIGN

While the design in the previous chapter would work, the long lengths of delay lines currently prove to be an impediment. To demonstrate the generation of a 3D cluster state in the short term with a few modes, we resorted to multimode circuit quantum electrodynamics and came up with a second design.

5.1 MULTIMODE CIRCUIT QUANTUM ELECTRODYNAMICS

Multimode Circuit Quantum Electrodynamics is an extension of cQED wherein we couple transmons (or any other superconducting qubit) to multimodal resonators. This allows the transmon to individually access the various standing modes of the resonator, sort of in a "random memory access" fashion (Naik *et al.* (2017)). Each such mode can be considered an independent qubit (in the single excitation manifold of each mode), with some coupling to the transmon. In our case, we drive interactions between the modes and the transmon using parametric flux modulation, described earlier in Chapter 3.

Coupling the transmon to a multimode resonator gives us a Hamiltonian that looks like:

$$\hat{H} = \omega_q(t)\hat{a}^\dagger\hat{a} + \frac{\alpha}{2}\hat{a}^\dagger\hat{a}^\dagger\hat{a}\hat{a} + \sum_{k=1}^{N}\{\omega_k\hat{b}_k^\dagger\hat{b}_k + g_k(\hat{b}_k^\dagger + \hat{b}_k)(\hat{a}^\dagger + \hat{a})\} \tag{5.1}$$

Where k indexes the different modes and the remainder of the notation is the same as in Chapter 3.

Since the modes reside in the same resonator, they are spaced apart in frequency by the FSR of the resonator. This spacing is what allows us to individually access each mode. As the difference frequency between each transmon and the modes are unique (again spaced apart by the FSR), we can parametrically drive the flux of the transmon to access specific modes without interacting with the others. Of course, this requires that the FSR be more than the bandwidth of the flux drive signal, but this can be achieved in practice.

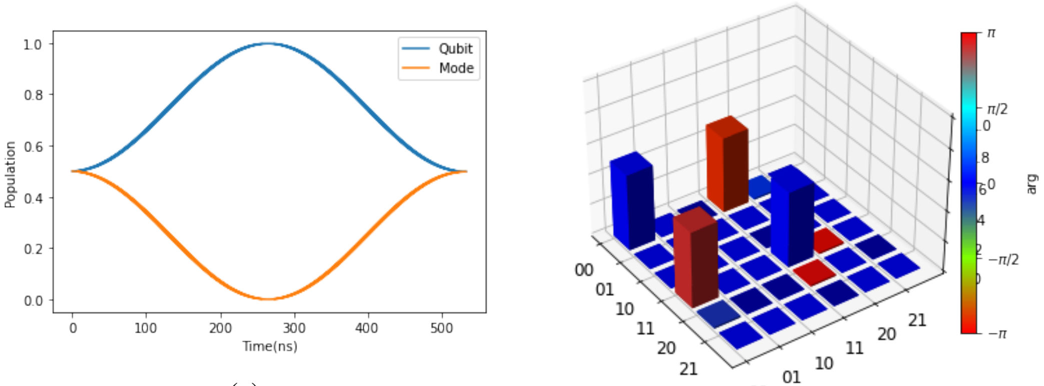
Note that in equation 5.1 the bare couplings g_k may be different for each mode: they usually follow a certain dispersive relation based on the resonator. In our case the values are fairly similar for the modes we consider (verified by simulations of dispersion relations of a particular chip in the lab), so the simulations assume the same bare coupling for all the modes.

Going back to the circuit we wish to implement (fig. 2.6), it can be seen that the single qubit gates are required only on the transmon (which is the emitter), and the two qubit gates required are the CNOT and the CZ. Furthermore, the CNOT is only applied to the modes once at the beginning, when they are in the vacuum state. This simplified things for the first design, and does so here as well.

We implement the single qubit gates on the transmon in the standard way using a charge drive as discussed in Chapter 2. The two qubit gates are done by using the 2nd excited level of the transmon and the exact procedure is described below (Naik *et al.*, 2017).

5.1.1 CZ Gate

To implement the CZ gate between the transmon and a mode we first note that the only state that causes a change is where the both the transmon and the mode are in the first excited state (Table: 5.1). Hence, we only need to selectively map $|e1\rangle$ to $-|e1\rangle$ to implement the CZ gate. We do so by using the flux drive to cause a stimulated Rabi oscillation between the states $|e1\rangle$ and $|f0\rangle$. By driving such an oscillation for one whole period, a phase of π is picked up in the process. This can be understood by considering the evolution between these two states on the Bloch sphere. The phase accumulated is related to the solid angle swept during the process, and since we drive for one whole cycle, this is π.



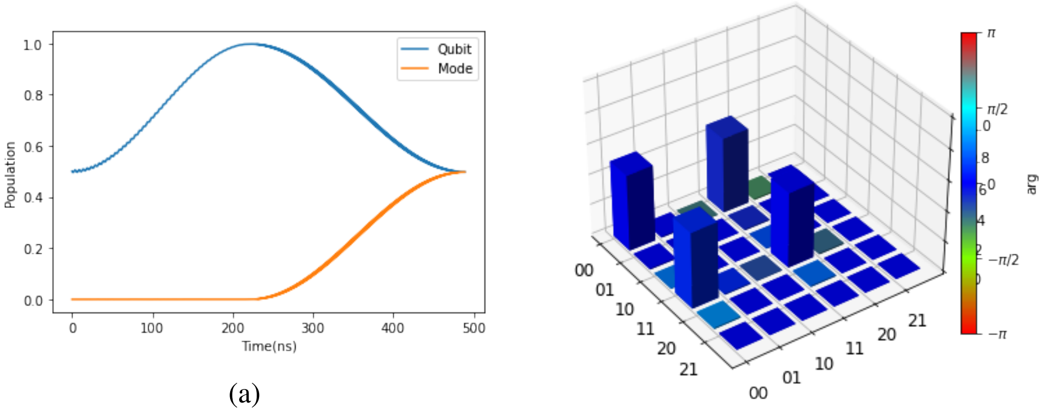
(a)

(b)

Figure 5.1: (a) CZ Gate dynamics, starting from the state $\frac{|g0\rangle+|e1\rangle}{2}$. Transmon states are $|g\rangle, |e\rangle, |f\rangle$ and mode states are $|0\rangle, |1\rangle$. The state $|e1\rangle$ is selectively made to transition to $|f0\rangle$ and back using the parametric flux drive, which is why the populations expected values follow the dynamics shown.(b)Density matrix of final state. Final state reached is $\frac{|g0\rangle-|e1\rangle}{2}$, beginning from $\frac{|g0\rangle+|e1\rangle}{2}$. This clearly shows that the gate being implemented is working as needed (truth table in table 5.1).

5.1.2 CNOT Gate

As mentioned earlier, the CNOT gate only needs to be implemented between the transmon in an arbitrary state, and the mode under consideration in the vacuum state ($|0\rangle$). Hence we only need to take care of a few transitions (Table: 5.2). Taking this into account, we can perform a CNOT gate by simply creating a photon when the transmon is in the first excited state, while maintaining the excitation in the transmon. Hence, we first drive the transmon from the first excited level ($|e\rangle$) to the second ($|f\rangle$) using a charge pulse, and then drive a stimulated Rabi oscillation between the $|f0\rangle$ state and the $|e1\rangle$ state for half a period. In doing so, we must take into account the extra phase accumulated during the oscillation, and drive the flux accordingly.



(a)

(b)

Figure 5.2: (a) CNOT Gate dynamics, starting from the state $\frac{(|g\rangle+|e\rangle)|0\rangle}{2}$. Transmon states are $|g\rangle, |e\rangle, |f\rangle$ and mode states are $|0\rangle, |1\rangle$. We first cause a $|e\rangle$ to $|f\rangle$ transition in the transmon using the charge drive. Then we transition from the $|f0\rangle$ level to the $|e1\rangle$ level.(b) Density matrix of final state. Final state reached is $\frac{|g0\rangle+|e1\rangle}{2}$, beginning from $\frac{(|g\rangle+|e\rangle)|0\rangle}{2}$. This clearly shows that the gate being implemented is working as needed (truth table in table 5.2).

Initial State	Final State
$|00\rangle$	$|00\rangle$
$|01\rangle$	$|01\rangle$
$|10\rangle$	$|10\rangle$
$|11\rangle$	$-|11\rangle$

Table 5.1: Truth table of CZ gate

Initial State	Final State
$|00\rangle$	$|00\rangle$
$|10\rangle$	$|11\rangle$

Table 5.2: Truth table of implemented CNOT gate for relevant states. First qubit is the transmon and second is the resonator mode

5.2 THE DESIGN

Now that we have the basic ingredients ready to implement the circuit, we can put them all together and come up with a design to implement the scheme. The proposed design is shown in figure 5.3. It consists of two transmons, a multimode resonator, and a filter between one of the transmons and the transmission line to the measurement unit.

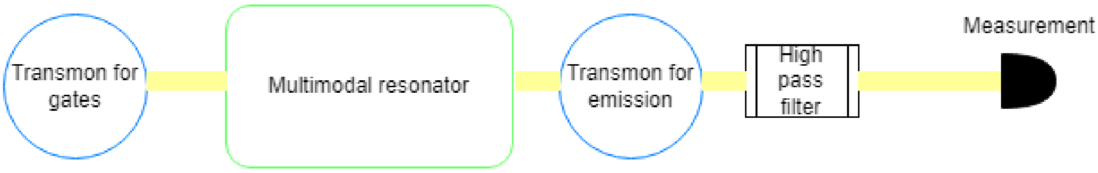


Figure 5.3: Proposed design based on multimode cQED.

The first transmon (the one on the left in figure 5.3) is what is used to enact the gates on the different modes and actually create the cluster state. We follow the same circuit shown in Figure 2.6 by implementing the gates as mentioned above. The Hadamard gate on the transmon is implemented using a charge drive followed by a virtual-z gate as explained in Chapter 2.

The second transmon is primarily for measurement. To send out a mode for measurement we first swap the mode's excitation onto this transmon using parametric flux modulation. Then we send out the mode into the transmission line from the transmon, again using parametric flux modulation. The high-pass filter in between is to prevent the transmon from spontaneously decaying into the waveguide. This also allows us to shape the pulse that we emit as we did in the first design. The cutoff frequency of the filter is well above the transmon's frequency (by about 2GHz) which prevents this. The pulse then goes onto the measurement chain, which generally includes a parametric amplifier before the actual measurement.

One may ask why we use such an elaborate scheme for measurement rather than just a dispersive readout of the transmon. The main reason is the time it takes for such a readout. In the case of one of the chips being designed in our lab, this time period is around 2us. Since the lifetimes of the resonator modes are only around 15us, this greatly limits the number of modes we would be able to measure before the states decohere. Hence, we "convert" these standing modes into traveling modes using the second transmon, which takes much lesser time (0.5us, limited by about twice the gate speeds of transmon-mode gates). Once the transmon has emitted one mode, it can immediately start converting another one. Note that we also make use of the fact that we only need to store one "slice" of the cluster state at a time as described in Chapter 2, so we pretty much only need to generate 2D cluster states in this case.

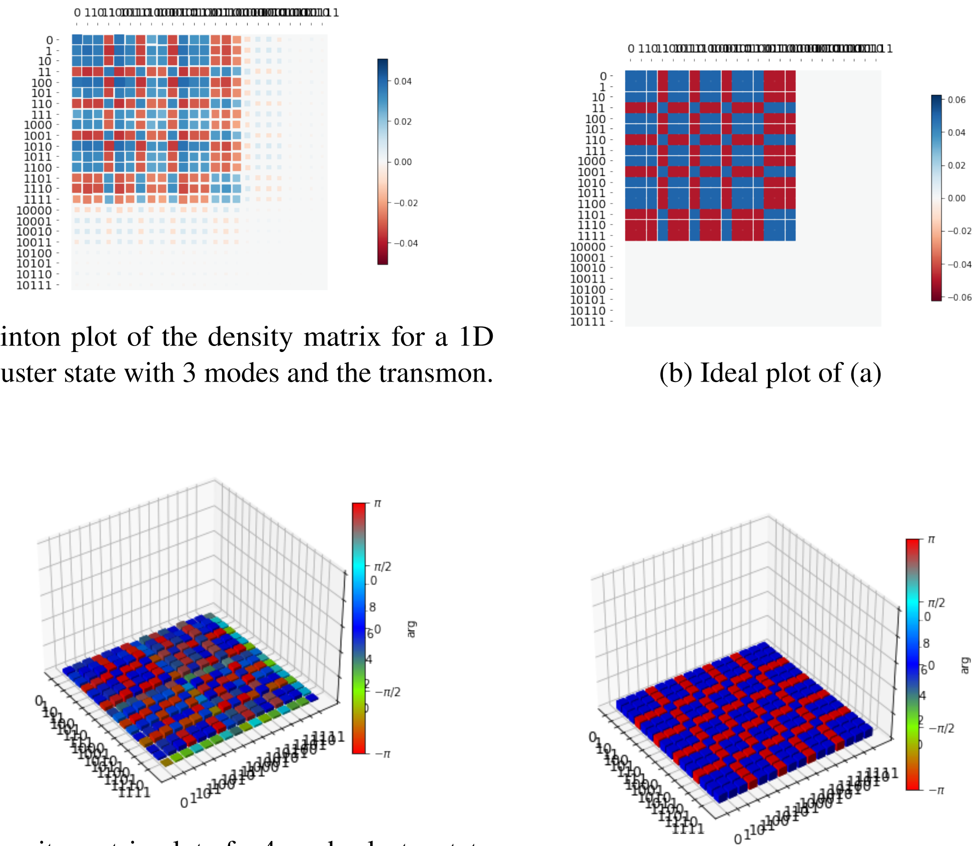
5.3 RESULTS

Using the scheme elaborated above, we perform simulations on a transmon coupled to a multimode resonator using QuTiP (Johansson *et al.*, 2013). For the density matrix plots shown in fig. 5.4, we readout the states of the resonator modes at the end of our circuit. Table 5.3 lists the quantum state fidelities of most of the generated states.

State	Fidelity	State	Fidelity
2 Qubit cluster	0.992	4 Qubit, square lattice	0.88
3 Qubit, triangle	0.946	3 Modes, triangle	0.74
4 Qubits, 1D	0.901	4 Modes, square	0.72

Table 5.3: Fidelities of different generated states, named based on number of qubits/modes and the shape of the underlying graph."x" modes indicates that the graph mentioned is only between "x" resonator modes: the transmon is connected to only one other node in the graph and is not part of the shape mentioned."y" qubits means the transmon is included in the shape mentioned.

As can be seen, for higher number of modes in the cluster state, the fidelity gets worse. One can also notice that there is residual excitation in the $|f\rangle$ level of the transmon in fig. 5.4a. Both of these issues seem to arise from dispersive effects: as the modes get populated the qubit frequency shifts slightly. Due to this, there is a mismatch between the driving frequency and the required excitation frequencies. This difference is small and hence the populations are nearer to the values expected, but there is a phase accumulation over time because of this as well. Since the circuit depth is larger with more qubits, this phase accumulation increases with the number of qubits, and that's what we see.



(a) Hinton plot of the density matrix for a 1D cluster state with 3 modes and the transmon.

(b) Ideal plot of (a)

(c) Density matrix plot of a 4 mode cluster state, with the underlying graph being a square.

(d) Ideal plot of (c)

Figure 5.4: Examples of generated states and their ideal states.(a) and (b) include the second excited level of the transmon. The last 8 rows/columns in figure (a) represent the second excited state of the transmon. To generate (c) and (d), we measure out the transmon from the cluster state. Values used for the simulations are well within range of experimental possibilities: transmon frequency of 2.8GHz, anharmonicity of 200MHz, lowest resonator mode frequency of 6.2GHz, FSR of 170MHz, bare coupling rate of 10MHz.

These simulations assume the same bare coupling rate between the transmon and all the modes. This, along with the phase accumulation issue described above, are what currently hinder us from generating larger lattices. As one starts using higher modes of the multimode resonator, the bare coupling grows weaker and hence the transmon-mode gates take longer. This affects the fidelity and also limits how many modes we could operate on before they decohere.

CHAPTER 6

CONCLUSION AND OUTLOOK

The future of quantum computing seems bright as both, industry and academia, have demonstrated great advancements in the field and are continuing to push the limits. A majority of these effort are indeed towards building larger and larger quantum systems to reach the fault tolerant regime described in Chapter 1. As we mentioned then, it is important for us to think about how best scale up while not making the system too complex to control and measure.

This was the fundamental driving factor for both of our designs; there are numerous ways to implement the scheme mentioned in Wan *et al.* (2021), but we feel that our particular approach to the problem will minimize the overhead of introducing new qubits. The control in either design is limited to 3 or lesser transmons and there is only a single measurement chain. In particular, compare this to some of the largest existing systems such as IBM's Eagle processor (127 qubits) or Google's Sycamore processor (53 qubits) where each qubit requires a separate control line. There are also numerous measurement chains, even though the readout is multiplexed (each line can usually only multiplex around 15 qubits at most). What allows us to get rid of this issue in particular is that we time-multiplex the readout of the different modes instead. This is an approach also followed by Deshpande *et al.* (2022) in their experiments, which has proven to be one of the biggest achievements in the field so far.

Another important aspect of our approach is the hybrid nature of our hardware. We use a superconducting platform, while storing our data in photonic qubits. This allows us to take the best of both worlds an is definitely worth considering while scaling up. Focusing on one kind of hardware will almost always have some limitation: some may have large coherence times, but will also be difficult to manipulate, while others may have fast gate speeds, but lower coherence times. By using hybrid architectures such as ours it may be possible to have both, large coherence times and high speed quantum gates. This will enable us not only to increase the system size, but also to cross the fault-tolerant thresholds. This too is an approach that is being investigated thoroughly in the academic community, such as for Continuous-Variable computing (eg. Grimm *et al.* (2020) and Reuer *et al.* (2021)).

The above points are what made it possible for us to demonstrate, albeit only in simulations for now, the creation of cluster states with our design. The cluster states shown in this thesis are definitely on the smaller side, but the important fact is the dimensionality. There have been demonstrations of 1D cluster states (Besse *et al.*, 2020) but none of 3D yet. In fact, 2D cluster states too have only very recently been demonstrated (conference presentation:Butler (2022) during the creation of this thesis) and is still not a published topic. Hence, our designs could very well be one of the first to demonstrate a 3D cluster state, even if it is just a smaller unit cell.

To the best of our knowledge, the idea of using parametric flux modulation to cause a reflection at a frequency well detuned from the bare qubit frequency has also not been investigated before on superconducting circuits. Such uses of parametric flux modulation may help in other areas of quantum information processing as well, such as in quantum routers. Our hope is that this work will encourage more research in this direction.

Since we've presented two different designs, there are a few advantages and disadvantages of both. Table 6.1 lists the features of both the designs. There are also various points to improve upon in both the designs that we list as future work:

- For the waveguide QED based design, design slow-light metamaterials which could help reduce the length of the delay lines.
- For design based on multimode cQED, the size of the cluster state that we can generate can be increased by taking into account dispersive shifts in all the gates.

Waveguide QED Design	Multimode cQED Design
Extensible architecture: only the length of delay line changes	Flexible architecture: can also generate states other than cluster states very easily
Gate times for each mode are the same	Calibration of gates is simple since we have control over the phase using the different drives
Propagating bosonic modes have low dephasing errors which may make it easier to reach the fault-tolerant regime	Can be implemented-on chip with existing technologies

Table 6.1: Comparison between the two proposed designs

- For either design, the pulse shape emitted into the waveguide must be controlled through parametric flux modulation. The pulse shape determines the measurement efficiency, as well as the reflection efficiency in the waveguide QED design. The optimal pulse shape depends largely on the parameters of the system, such as the coupling rates and frequencies. Looking into this and coming up with an optimal sideband pulse for the parametric flux modulation is definitely worth researching.

- Dual-rail schemes for either design requires further consideration. This would help in detecting photon-loss that may have occured during the generation or in the measurement chain.

Of course, we are also looking into a possible experimental demonstration of our schemes. In particular, the multimode approach is one that we may be able to perform on some of the chips that the lab is already designing. The parameters of this chip are expected to be able to create atleast small 3D cluster states, as verified by simulations.

In conclusion, we hope that this work will motivate further research into hybrid architectures for designing quantum systems that scale better to help develop the novel and exciting field of quantum computing.

APPENDIX A

INPUT OUTPUT THEORY EQUATIONS

In this section we briefly derive the input-output theory relations shown in Chapter 4. We use the same notation as used in the main text. We start with the hamiltonians for the waveguide, the two-level system and their interaction (Kannan, 2018):

$$\hat{H} = H_q + H_{wg} + H_{int} \tag{A.1}$$

$$\hat{H}_q = \hbar\omega_q \hat{\sigma}^+ \hat{\sigma}^- \tag{A.2}$$

$$\hat{H}_{wg} = \int d\omega \hbar\omega (\hat{b}_r^\dagger(\omega)\hat{b}_r(\omega) + \hat{b}_l^\dagger(\omega)\hat{b}_l(\omega)) \tag{A.3}$$

$$\hat{H}_{int} = i\hbar g \hat{\sigma}^x \int d\omega\sqrt{\omega}((\hat{b}_r^\dagger(\omega) - \hat{b}_r(\omega) + (\hat{b}_l^\dagger(\omega) - \hat{b}_l(\omega)) \tag{A.4}$$

Using the Hamiltonians above, we can derive the Heisenberg equations of motion for the waveguide operators:

$$\dot{\hat{b}}_r(\omega) = \frac{i}{h}[\hat{H}, \hat{b}_r(\omega)] = -i\hat{b}_r(\omega) + g\sqrt{\omega}\hat{\sigma}_x \tag{A.5}$$

We can integrate the above expression from time $t = 0$ to t,as well as from $t = t$ to t_f to yield:

$$\hat{b}_r(\omega, t) = \hat{b}_r(\omega, 0)e^{-i\omega t} + g\sqrt{\omega}\int_0^t e^{-i\omega(t-\tau)\hat{\sigma}_x(\tau)d\tau} \tag{A.6}$$

$$\hat{b}_r(\omega, t) = \hat{b}_r(\omega, 0)e^{-i\omega t} - g\sqrt{\omega}\int_t^{t_f} e^{-i\omega(t-\tau)\hat{\sigma}_x(\tau)d\tau} \tag{A.7}$$

In writing our master equation in the Lindbladian form in 4.6, we have already made an implicit assumption that the system dynamics are not too fast. Quantitatively, since most of our dynamics occur in the GHz range (superconducting qubits are usually of this range), we are restricted to dynamics where t» 0.1ns. We can now use the above two equations and arrive at:

$$\hat{b}_r^{out}(t) = \hat{b}_r^{in}(t) + \int_0^\infty \frac{d\omega}{\sqrt{2\pi}} g\sqrt{\omega} \int_0^{t_f} e^{-i\omega(t-\tau)\hat{\sigma}_x(\tau)d\tau} \tag{A.8}$$

Where $\hat{b}_r^{out}(t)$ and $\hat{b}_r^{in}(t)$ are as defined in the main text. Intuitively, they are the fourier transforms of the frequency modes in the waveguide before and after interaction with the qubit.

What remains is the integral in 4.9. To this end we can write the Pauli X operator at time τ in terms of its value at time t:

$$\hat{\sigma}_x(\tau) =^{i\hat{H}(\tau-t)/\hbar} \hat{\sigma}_x(t)^{-i\hat{H}(\tau-t)/\hbar} \tag{A.9}$$

One can further ignore the contributions of the interaction hamiltonian in $\hat{H}$ since its contribution would be lesser than that of the bare qubit hamiltonian. This means that the Pauli X operator will evolve just due to the precession of the qubit about the Bloch sphere at the qubit frequency, i.e. $\hat{\sigma}_x(\tau) \approx \hat{\sigma}(t)e^{-i\omega_q(\tau-t)}$. Additionally, in the long time limit (after sufficient interaction), $\omega_q(\tau - t) \to \infty$. Using these approximations we arrive at the input output relation shown in the main text:

$$\hat{b}_r^{out}(t) = \hat{b}_r^{in}(t) + \sqrt{\frac{\Gamma}{2}}\hat{\sigma}^-(t) \tag{A.10}$$

A.1 INPUT OUTPUT THEORY IN PRESENCE OF PARAMETRIC FLUX MODULATION

In the above derivation, we've assumed that the qubit frequency remains fixed, and more importantly that the bare qubit hamiltonian is not time dependent. In the presence of a parametric flux drive this is not the case and one needs to include additional terms in the final input output relations. Most importantly, the assumption of $\hat{\sigma}_x(\tau) \approx \hat{\sigma}_x(t)e^{-i\omega_q(\tau-t)}$ breaks down.

If we assume the qubit frequency to be modulated as $\omega_q(t) = \omega_0 + \epsilon\sin(\omega_{sb}t)$, we can approximate the time dependence of the qubit Pauli X operator as:

$$\hat{\sigma}_x(\tau) \approx \hat{\sigma}(t) e^{-i\omega_q(\tau-t)} e^{-i\frac{\epsilon}{\omega_{sb}}(\cos(\omega_{sb}(t))-\cos(\omega_{sb}\tau))} \tag{A.11}$$

This can then be substituted in A.8 to arrive at the final relations. The integral is quite difficult, and hence analytic methods must be used in this case instead. For the results shown in the main text, we use approximate methods to show the reflection at the sideband frequency. Of course, as can be seen there, the approximations used are quite crude and evaluating this integral better is part of future work.

APPENDIX B

CHIP DESIGN AND EXPERIMENTAL CALIBRATION

B.1 EM SIMULATIONS

Apart from all the simulations done in QuTiP for the two designs described in the main content, we have also worked on electromagnetic simulations in ANSYS Q3D and Sonnet EM. These were done for parts of a chip that is being designed at the lab.

The Sonnet EM simulations were done to optimize the parameters the Purcell filter for the readout of this chip. In particular, we had a target frequency of the Purcell filter and a target coupling to the transmission line. These parameters can be estimated from the S11 response of the filter at the input. A screenshot of the design of the filter in Sonnet EM is shown below in fig.B.1. To achieve these values, we had to change the lengths of the resonator being used, as well as change the parameters (finger length, number of fingers etc.) of the interdigitated capacitor used to couple to the transmission line.

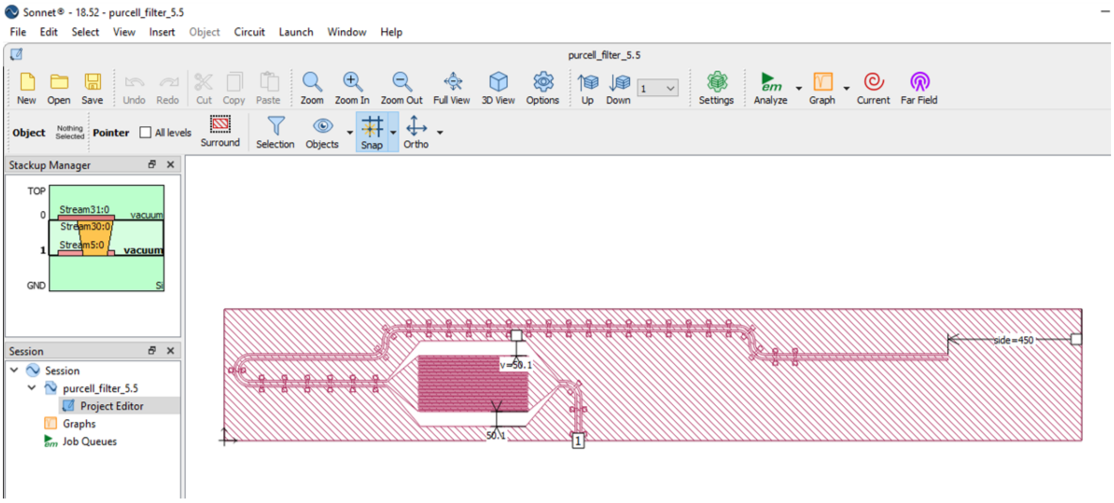

Sonnet® - 18.52 - purcell_filter_5.5
File Edit Select View Insert Object Circuit Launch Window Help
purcell_filter_5.5
New Open Save Undo Redo Cut Copy Paste Zoom Zoom In Zoom Out Full View 3D View Options Up Down Settings Analyze Graph Current Far Field
Object Nothing Selected Pointer All levels Surround Selection Objects Snap Ortho
Stackup Manager
TOP
Stream31:0
vacuum
0
Stream30:0
Stream5:0
vacuum
1
GND
Si
Session
Session
purcell_filter_5.5
Project Editor
Graphs
Job Queues
side=450
v=50.1
50.1
1

Figure B.1: Purcell filter simulation in Sonnet.

We also performed simulations to extract the coupling capacitances of the transmon to the readout resonator, its self-capacitance, as well as its coupling to a multimode resonator. These were done using ANSYS Q3D, where we simulated just a section of the chip that contribute to these values the most (see fig.B.2). By adjusting the lengths of various parts we were able to reach the desired capacitances.

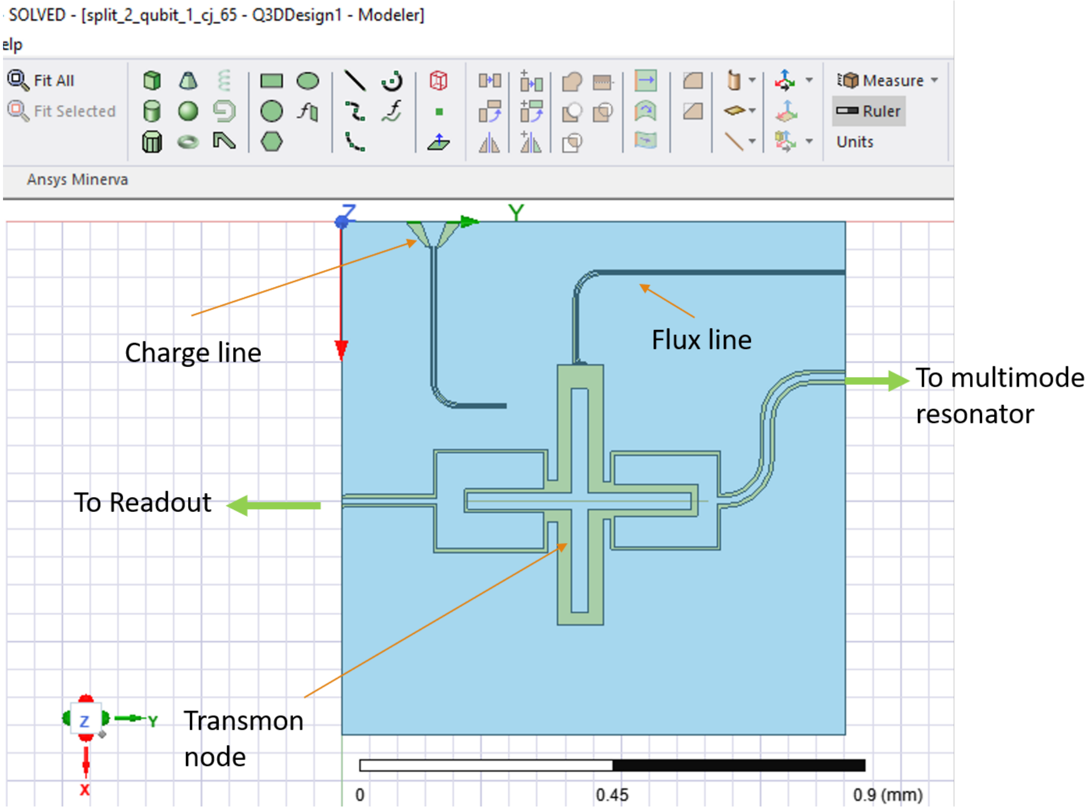
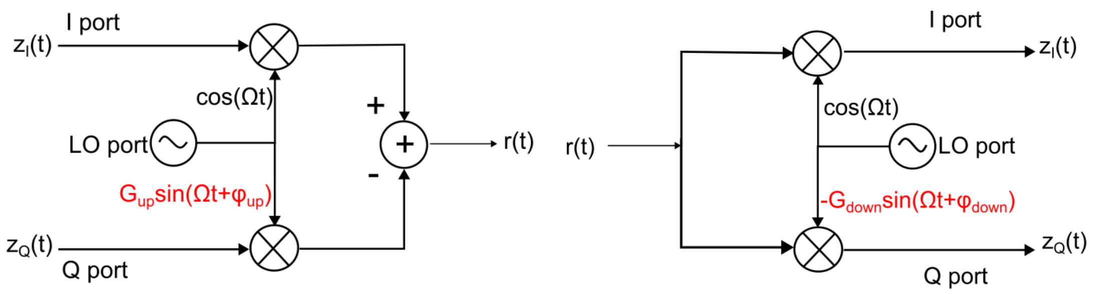


Figure B.2: Simulated design in ANSYS Q3D. Different sections are labelled in the figure. The majority of the blue region surrounding the transmon node is the ground plane.

B.2 CALIBRATION OF IQ MIXER FOR MEASUREMENTS

To perform measurements on the devices that will soon be fabricated, we have been working towards setting up the necessary equipment, including the dilution refrigerator and the IQ mixers. In this appendix, we outline the approach we have followed towards minimising the image sideband that arises due to imbalances in our IQ mixers. We largely follow the work done by Jolin *et al.* (2020) and model the imbalance in a similar fashion.



(a) Up-conversion (b) Down-conversion

Figure B.3: Imbalance modelling

As can be seen in figure B.3, the imbalance is modelled as a phase and amplitude imbalance in the I and Q arms of the mixer. We also need to calibrate for the LO leakage (not shown in the model) which arises due to diode imperfections of the mixer circuit. This results in the up-converted spectrum with an image sideband(include an image of this), as well as a component at the LO frequency. The LO leakage can be fixed by simply sweeping through the DC offsets of the I and Q inputs. Hence we shall focus on correcting the imbalance. The result of such an imbalance gives the following I and Q components to the up converted signal:

$$I = (z_I - z_Q G_{up} \sin(\phi_{up})) \cos(\Omega t) \tag{B.1}$$

$$Q = z_Q G_{up} \sin(\Omega t) \cos(\phi_{up}) \tag{B.2}$$

It is straightforward to see that simple modulations on the I and Q components independently is not enough to correct for this imbalance. Hence the pre-distortion applied before the up converter is of the form shown in figure B.4. Hence we need to calibrate for these two parameters α and β. Ideally, we would want $\alpha = G_{up} \cos(\phi_{up})$ and $\beta = G_{up} \sin(\phi_{up})$, which perfectly cancels out the imbalance.

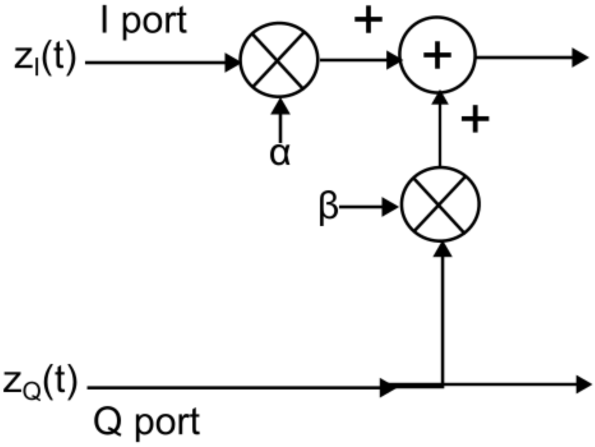


Figure B.4: Pre-distortion applied before going into mixer to counter the imbalance

Since the down-converter also has an imbalance, we need to account for that too. It would indeed be best if both the up-converter and downconverter could be calibrated independently, in-situ. Jolin *et al.* (2020) do this by applying a Kalman filter for the down-converter to filter out noise, then calibrating the up-converter based on estimates for the Image Leakage Ratio(ILR). To decouple the down-converter's imbalance from that of the up-conversion stage, the Kalman filter is applied to the output of the down-converter when there is no input. Since the input is now uncorrelated, we can derive quantities that help correct the imbalance in the down-converter. Using these estimates, we then provide an input to the up-converter as well, and proceed to reduce the ILR.

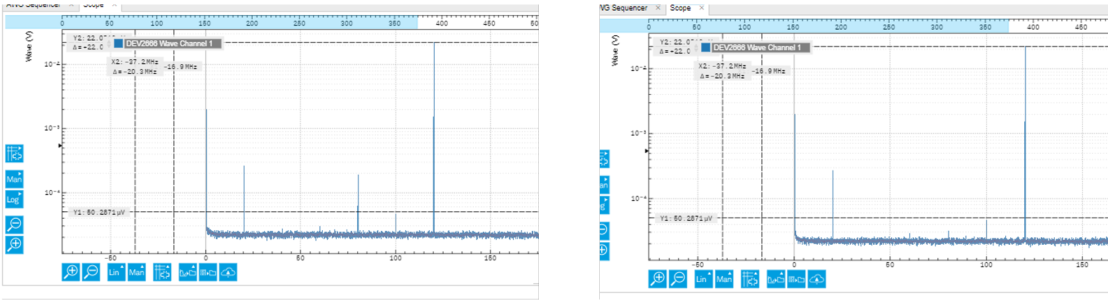


Figure B.5: Signal at 20MHz on a carrier frequency which was ≈ 5 G Hz, and we demodulate to 100 MHz. Left: uncalibrated received signals. We want to eliminate the lower sideband at 80MHz. Right: After calibration of pre-distortion. Lower sideband is suppressed.

REFERENCES

1. **Besse, J.-C.**, **K. Reuer**, **M. C. Collodo**, **A. Wulff**, **L. Wernli**, **A. Copetudo**, **D. Malz**, **P. Magnard**, **A. Akin**, **M. Gabureac**, **G. J. Norris**, **J. I. Cirac**, **A. Wallraff**, and **C. Eichler** (2020). Realizing a deterministic source of multipartite-entangled photonic qubits. *Nature Communications*, **11**(1). URL `https://doi.org/10.1038%2Fs41467-020-18635-x`.

2. **Butler, G. K. O. P., V Ferreira**, Deterministic generation of multidimensional microwave photonic cluster states with a single quantum emitter. 2022.

3. **Deshpande, A.**, **A. Mehta**, **T. Vincent**, **N. Quesada**, **M. Hinsche**, **M. Ioannou**, **L. Madsen**, **J. Lavoie**, **H. Qi**, **J. Eisert**, **D. Hangleiter**, **B. Fefferman**, and **I. Dhand** (2022). Quantum computational advantage via high-dimensional gaussian boson sampling. *Science Advances*, **8**(1). URL `https://doi.org/10.1126%2Fsciadv.abi7894`.

4. **Ferreira, V. S.**, **J. Banker**, **A. Sipahigil**, **M. H. Matheny**, **A. J. Keller**, **E. Kim**, **M. Mirhosseini**, and **O. Painter** (2021). Collapse and revival of an artificial atom coupled to a structured photonic reservoir. *Phys. Rev. X*, **11**, 041043. URL `https://link.aps.org/doi/10.1103/PhysRevX.11.041043`.

5. **Fischer, K. A.**, **R. Trivedi**, **V. Ramasesh**, **I. Siddiqi**, and **J. Vučković** (2018). Scattering into one-dimensional waveguides from a coherently-driven quantum-optical system. *Quantum*, **2**, 69. ISSN 2521-327X. URL `http://dx.doi.org/10.22331/q-2018-05-28-69`.

6. **Fowler, A. G.**, **M. Mariantoni**, **J. M. Martinis**, and **A. N. Cleland** (2012). Surface codes: Towards practical large-scale quantum computation. *Phys. Rev. A*, **86**, 032324. URL `https://link.aps.org/doi/10.1103/PhysRevA.86.032324`.

7. **Gardiner, C. W.** and **M. J. Collett** (1985). Input and output in damped quantum systems: Quantum stochastic differential equations and the master equation. *Phys. Rev. A*, **31**, 3761–3774. URL `https://link.aps.org/doi/10.1103/PhysRevA.31.3761`.

8. **Greene, T.** (2018). Here's why 100 qubit quantum computers could change everything. URL `https://thenextweb.com/news/heres-why-100-qubit-quantum-computers-could-change-everything`.

9. **Grimm, A.**, **N. E. Frattini**, **S. Puri**, **S. O. Mundhada**, **S. Touzard**, **M. Mirrahimi**, **S. M. Girvin**, **S. Shankar**, and **M. H. Devoret** (2020). Stabilization and operation of a kerr-cat qubit. *Nature*, **584**(7820), 205–209. URL `https://doi.org/10.1038%2Fs41586-020-2587-z`.

10. **Johansson, J.**, **P. Nation**, and **F. Nori** (2013). Qutip 2: A python framework for the dynamics of open quantum systems. *Computer Physics Communications*, **184**(4),

1234–1240. ISSN 0010-4655. URL https://www.sciencedirect.com/science/article/pii/S0010465512003955.

11. **Jolin, S. W.**, **R. Borgani**, **M. O. Tholén**, **D. Forchheimer**, and **D. B. Haviland** (2020). Calibration of mixer amplitude and phase imbalance in superconducting circuits. *Review of Scientific Instruments*, **91**(12), 124707. ISSN 1089-7623. URL http://dx.doi.org/10.1063/5.0025836.

12. **Kannan, B.** (2018). Waveguide quantum electrodynamics with superconducting qubits. URL https://dspace.mit.edu/handle/1721.1/120400.

13. **Kannan, B.**, **M. J. Ruckriegel**, **D. L. Campbell**, **A. F. Kockum**, **J. Braumüller**, **D. K. Kim**, **M. Kjaergaard**, **P. Krantz**, **A. Melville**, **B. M. Niedzielski**, **A. Vepsäläinen**, **R. Winik**, **J. L. Yoder**, **F. Nori**, **T. P. Orlando**, **S. Gustavsson**, and **W. D. Oliver** (2020). Waveguide quantum electrodynamics with superconducting artificial giant atoms. *Nature*, **583**(7818), 775–779. URL https://doi.org/10.1038%2Fs41586-020-2529-9.

14. **Kiilerich, A. H.** and **K. Mølmer** (2019). Input-output theory with quantum pulses. *Physical Review Letters*, **123**(12). ISSN 1079-7114. URL http://dx.doi.org/10.1103/PhysRevLett.123.123604.

15. **Kjaergaard, M.**, **M. E. Schwartz**, **J. Braumüller**, **P. Krantz**, **J. I.-J. Wang**, **S. Gustavsson**, and **W. D. Oliver** (2020). Superconducting qubits: Current state of play. *Annual Review of Condensed Matter Physics*, **11**(1), 369–395. URL https://doi.org/10.1146/annurev-conmatphys-031119-050605.

16. **Krantz, P.**, **M. Kjaergaard**, **F. Yan**, **T. P. Orlando**, **S. Gustavsson**, and **W. D. Oliver** (2019). A quantum engineer's guide to superconducting qubits. *Applied Physics Reviews*, **6**(2), 021318. URL https://doi.org/10.1063%2F1.5089550.

17. **Kuzmin, R.**, **R. Mencia**, **N. Grabon**, **N. Mehta**, **Y.-H. Lin**, and **V. E. Manucharyan** (2019). Quantum electrodynamics of a superconductor–insulator phase transition. *Nature Physics*, **15**(9), 930–934. URL https://doi.org/10.1038%2Fs41567-019-0553-1.

18. **Naik, R. K.**, **N. Leung**, **S. Chakram**, **P. Groszkowski**, **Y. Lu**, **N. Earnest**, **D. C. McKay**, **J. Koch**, and **D. I. Schuster** (2017). Random access quantum information processors using multimode circuit quantum electrodynamics. *Nature Communications*, **8**(1). URL https://doi.org/10.1038%2Fs41467-017-02046-6.

19. **Raussendorf, R.** (). Introduction to measurement based quantum computation. URL https://phas.ubc.ca/~raussen/Supplements/VideoLectures.html.

20. **Raussendorf, R.** and **H. J. Briegel** (2001). A one-way quantum computer. *Phys. Rev. Lett.*, **86**, 5188–5191. URL https://link.aps.org/doi/10.1103/PhysRevLett.86.5188.

21. **Raussendorf, R.**, **D. E. Browne**, and **H. J. Briegel** (2003). Measurement-based quantum computation on cluster states. *Physical Review A*, **68**(2). ISSN 1094-1622. URL `http://dx.doi.org/10.1103/PhysRevA.68.022312`.

22. **Reuer, K.**, **J.-C. Besse**, **L. Wernli**, **P. Magnard**, **P. Kurpiers**, **G. J. Norris**, **A. Wallraff**, and **C. Eichler** (2021). Realization of a universal quantum gate set for itinerant microwave photons. URL `https://arxiv.org/abs/2106.03481`.

23. **Somoroff, A.**, **Q. Ficheux**, **R. A. Mencia**, **H. Xiong**, **R. V. Kuzmin**, and **V. E. Manucharyan** (2021). Millisecond coherence in a superconducting qubit. URL `https://arxiv.org/abs/2103.08578`.

24. **Wan, K.**, **S. Choi**, **I. H. Kim**, **N. Shutty**, and **P. Hayden** (2021). Fault-tolerant qubit from a constant number of components. *PRX Quantum*, **2**(4). URL `https://doi.org/10.1103%2Fprxquantum.2.040345`.