

Implementing quantum computing on superconducting qubits

Tan, Paul Yuanzheng

2022

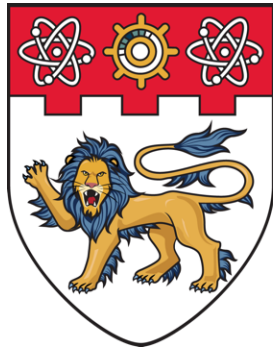
Tan, P. Y. (2022). Implementing quantum computing on superconducting qubits. Master's thesis, Nanyang Technological University, Singapore. <https://hdl.handle.net/10356/160507>

<https://hdl.handle.net/10356/160507>

<https://doi.org/10.32657/10356/160507>

This work is licensed under a Creative Commons Attribution-NonCommercial 4.0 International License (CC BY-NC 4.0).

Downloaded on 09 Apr 2024 11:07:03 SGT



**NANYANG
TECHNOLOGICAL
UNIVERSITY**

SINGAPORE

**IMPLEMENTING QUANTUM COMPUTING ON
SUPERCONDUCTING QUBITS**

TAN YUANZHENG PAUL

SCHOOL OF PHYSICAL AND MATHEMATICAL SCIENCES

2022

IMPLEMENTING QUANTUM COMPUTING ON SUPERCONDUCTING QUBITS

TAN YUANZHENG PAUL

SCHOOL OF PHYSICAL AND MATHEMATICAL SCIENCES

A thesis submitted to the Nanyang Technological
University in partial fulfilment of the requirement for the
degree of Master of Science

2022

Statement of Originality

I hereby certify that the work embodied in this thesis is the result of original research done by me except where otherwise stated in this thesis. The thesis work has not been submitted for a degree or professional qualification to any other university or institution. I declare that this thesis is written by myself and is free of plagiarism and of sufficient grammatical clarity to be examined. I confirm that the investigations were conducted in accord with the ethics policies and integrity standards of Nanyang Technological University and that the research data are presented honestly and without prejudice.

NTU NTU NTU NTU NTU NTU NTU NTU
NTU NTU NTU NTU NTU NTU NTU NTU
NTU NTU NTU NTU NTU NTU NTU NTU
NTU NTU NTU NTU NTU NTU NTU NTU



..... 11/1/2022. . . .
Date

...Tan Yuanzheng Paul.

Supervisor Declaration Statement

I have reviewed the content and presentation style of this thesis and declare it of sufficient grammatical clarity to be examined. To the best of my knowledge, the thesis is free of plagiarism and the research and writing are those of the candidate's except as acknowledged in the Author Attribution Statement. I confirm that the investigations were conducted in accord with the ethics policies and integrity standards of Nanyang Technological University and that the research data are presented honestly and without prejudice

A handwritten signature in black ink, appearing to read 'R. Helmut Dumke', is written over a background of faint, repeating 'NTU' text.

....11/1/2022.....
Date

..Rainer Helmut Dumke

Authorship Attribution Statement

This thesis contains material from a paper published in the following peer-reviewed journal in which I am listed as an author.

Figure 24 is taken from Zaw, L. H. et al. Ghost factors in Gauss-sum factorization with transmon qubits. In: *Phys. Rev. A* 104 (6 2021), p. 062606. doi: 10.1103/PhysRevA.104.062606.

NTU N
NTU N
NTU N
NTU N



NTU NT
NTU NT
NTU NT
NTU NT

...11/1/2022.....
Date

... Tan Yuanzheng Paul.

Abstract

Quantum computing utilizes quantum mechanics to perform computations with superconducting qubits being the more mature technology as of writing in realizing a quantum computer. In this thesis, we describe an implementation of a Quantum Processor Unit (QPU), which is used to execute instructions in the form of microwave pulses on a superconducting qubit device through the use of signal generators and to perform qubit readout. We further extend the QPU as a platform to perform qubit characterization tasks such as spectroscopy and decoherence measurements in order to determine and optimize the working parameters to perform quantum gate operations. We also demonstrate the use of the QPU in performing qubit experiments such as Gauss Sum Factorization in determining the factors of an integer and the Bell Inequality test in examining the entanglement strength between a pair of qubits.

We bridge the divide of quantum computation and qubit hardware in the compilation of quantum circuits into microwave pulse sequences for the QPU to execute. The process of compilation, as well as hardware limitations and pre-compilation optimization procedures are discussed. Finally, we show an example of executing a variational quantum algorithm and break down this example in all layers from the quantum circuit provided by the user to the pulse sequence that will be executed on the QPU.

Acknowledgements

While I was still unsure about my future career, Professor Rainer gave me a tour of his laboratories and proposed a Final Year Project as part of my undergraduate study. He introduced me to this field and pretty much gave me the keys to the experiment. For giving me his trust, I am grateful and hope to work with him in the future.

Long Nyugen was the first person I worked with in the lab and still probably the person I work with the most. He taught me the ropes and let me get on with it. His intuition in seeing trends in data is without peer and has led to many interesting results. Thanks for the company and the board games.

Alessandro Landro built the foundations for the experiment setup and showed me how everything worked. The lessons I learnt from your work I still apply today. Thank you and I hope you are enjoying your time in Finland. Congratulations on your marriage.

Rangga Budoyo does the qubit fabrication in the lab and I frequently consult with him on the insides of the fridge. His experience has contributed a lot in helping me understanding the inner workings of the qubits.

Dr Yap has been a great advisor through my two and a half years in the lab. I have always looked towards his experience in electronics and NMR experiments. While he has not been able to come to the lab with the COVID restrictions, I feel his virtual advice and presence in my work.

Christoph Hufnagel is a good friend, senior and jokester. While we work on different projects, we collaborate occasionally and I gain from his experience. I consult him from time to time when I get stuck and learn from his guidance. Congratulations on your newborn child Julian!

Chong Hian has been a good friend through my journey and we keep in contact from time to time. Thanks for your chats on variational algorithms and the needs of the information theory community.

I learnt a lot on experimenter attitude and philosophy from Koon Siang and Fong En, my seniors working on different projects. Congratulations on your graduations!

Working with Dr Stefano Carrazza and his Qibo team has opened my eyes on the aspects of quantum simulation. Our collaboration has been very insightful in the driving force behind my work. Thank you.

Kun Hee is my partner in crime in designing the qubit control structure while I have worked with Kai Sheng on a qubit control project. Both of them have been great insights on qubit information and experiments.

Dr Farshad Foroughi contributed to the first few qubit samples I worked with and taught me on qubit readout. Thank you.

Dr Koh and Lin Htoo have been great partners on the Gauss sums experiments. Thanks to you two both on your contribution towards the publication and I hope to collaborate more in the future.

Finally, to my friends and family. Your support through thick and thin has guided me on this journey. The person I am today and the values that embody

me are the results of the experiences we have gone through together. This thesis I dedicate to you as a waymark on our endless journey.

Contents

1	Introduction	11
1.1	Qubits	12
1.1.1	Quantum Logic Gates	13
1.1.2	Superposition and Entanglement	14
1.2	Quantum Computation	14
1.2.1	Grover's Algorithm	15
1.2.2	3SAT	16
2	Superconducting Qubits	17
2.1	Superconducting LC Circuits	18
2.2	Transmon Qubit	20
2.2.1	Tunable Qubit	20
2.3	Qubit Driving	21
2.4	Qubit Readout	22
2.5	Qubit Decoherence	23
3	Quantum Processing Unit (QPU)	25
3.1	Infrastructure	25
3.1.1	Signal Generation	25
3.1.2	Signal Readout	29
3.1.3	Pulse Description	29
3.1.4	Execution Workflow	30
3.2	Qubit Calibration	33
3.2.1	Resonator Spectroscopy	33
3.2.2	Qubit Characterization	34
3.2.3	Rabi Spectroscopy	36
3.2.4	Ramsey Spectroscopy	39
3.2.5	Two Qubit Interaction	41
3.3	Qubit Measurements	45
3.3.1	Quantum State Tomography	45
3.3.2	Decoherence	46
3.3.3	Gauss Sum Factorization	51
3.3.4	Bell Inequality	53
4	Software Stack	57
4.1	Quantum Software Development Kit (QSDK)	57
4.2	Circuit Compilation and Execution	58
4.2.1	Pulse Timing Considerations	58
4.3	Cloud-based Quantum Computing	59
4.4	Circuit Pre-compilation	61
4.4.1	Qubit Connectivity	61
4.4.2	Gate Decomposition	62
4.5	Variational Quantum Computing	64

5 Conclusion	66
Appendices	72
A Path finding algorithm for swapping qubits	72

1 Introduction

In the 17th century, Newton published the Principia, which became the foundation for describing the motion of physical bodies. This is also known as classical mechanics, where a physical body can be defined by its position in space and momentum, the product of its mass and velocity. Today, classical mechanics play an integral role in society being the cornerstone behind many fields. For example, dynamics and motion drive architecture and space exploration. Fluid dynamics are key in oil drilling and healthcare.

In the early 20th century, there were phenomena that could not be fully explained by classical mechanics, such as the duality of wave and matter and black body radiation. The theory of quantum mechanics was developed in this time period to reconcile these differences. It proposed that the parameters of a system are not continuous but discrete, the system experiences both wave and particle characteristics and that there is a limit to the accuracy of measuring these parameters.

Quantum chemistry is a field that uses quantum mechanics to understand the behaviour of atoms and molecules. However, as the number of particles increase, so too does the complexity of calculations that are needed to model the system. The number of variables also scales exponentially with the size of the system. To combat this, Feynman proposed the development of quantum computers to model other quantum systems and aid in this task.

This has led to two tracks of development, in realizing a physical system for quantum computation and the development of algorithms to harness the power of quantum computation. In the former, different physical systems such as ion traps, nuclear magnetic resonance (NMR), electron spins, semiconducting qubits and superconducting qubits have been used to implement a quantum computer. In the latter, Shor’s algorithm [1] is an example of quantum computation where integer factorization can be performed with higher efficiency than classical algorithms, as it has polynomial time complexity compared to classical algorithms with exponential time complexity.

As of writing, software packages such as Qiskit and Cirq [2–4] are readily available to the general public and provide a high level abstraction for quantum computing. These packages can be used to simulate or execute user-designed programs on the physical hardware provided by Google and IBM. However, the interface between user and experimental systems is often opaque. The processes used to transpile the user’s code into an experimental sequence for quantum hardware are usually not available to the user. These processes are required in the implementation of a quantum computer. Hence, this thesis focuses on my work in implementing the transpilation processes for quantum computing on superconducting qubits as well as characterization and control methods required for superconducting qubit operation.

In the first chapter, I cover an introduction to the quantum mechanics behind qubits, the basic blocks of quantum computation, and provide an example of how quantum computation is performed. The second chapter introduces the underlying theory behind superconducting qubits. The third chapter describes

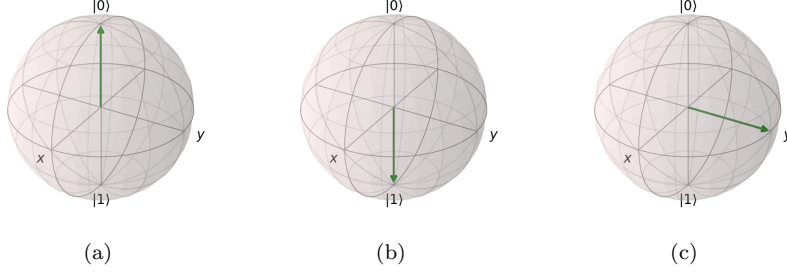


Figure 1: Bloch sphere representation of a qubit. By convention, qubits are measured along the Z -axis. The positive Z -axis is taken to be $|0\rangle$ and the negative as $|1\rangle$. (a) Qubit in the $|0\rangle$ or $|+Z\rangle$ state. (b) Qubit in the $|1\rangle$ or $|-Z\rangle$ state. (c) Qubit in the $|+Y\rangle$ state.

my work in designing the software abstraction and implementation of the QPU, a system for calibrating and performing qubit experiments. Finally, I discuss my current efforts in extending the QPU for use in quantum computation and the future challenges in scaling up the QPU.

1.1 Qubits

Qubits or quantum bits are the building blocks of a quantum computer. Much like how a classical bit is expressed as a binary state, 0 (false) or 1 (true), a qubit is defined to be approximated as a two level quantum system and can be expressed as $|\Psi\rangle = \alpha|0\rangle + \beta|1\rangle$, where $\alpha, \beta \in \mathbb{C}$, $|\alpha|^2 + |\beta|^2 = 1$ and $|0\rangle$ and $|1\rangle$ are the associated states. These complex numbers are known as probability amplitudes. When the qubit is measured, the system collapses to one of its states with a probability given by the square of the state's probability amplitude.

$$|\psi\rangle = e^{i\gamma} \left(\cos \frac{\theta}{2} |0\rangle + e^{i\phi} \sin \frac{\theta}{2} |1\rangle \right) \quad (1)$$

We can also represent the qubit in a Bloch sphere, shown in Figure 1 and described by Equation 1, where θ describes the angle along the longitudinal axis between $|0\rangle$ and $|1\rangle$, ϕ describes the relative phase and γ describes the global phase. They are restricted by $0 \leq \theta \leq \pi$ and $0 \leq \phi \leq 2\pi$. When the qubit is measured in the Z -axis, ϕ is ignored and cannot be observed. However, when the qubit is measured in the X or Y -axes, ϕ has an effect on the measurement result. This relation between measuring ϕ and θ simultaneously alludes to the uncertainty principle and plays a larger role when we discuss the orthogonality of states in the following sections.

Gate	Matrix
X	$\begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$
Y	$\begin{bmatrix} 0 & -i \\ i & 0 \end{bmatrix}$
Z	$\begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}$
CNOT	$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix}$

Table 1: Examples of quantum logic gates and their unitary matrix representations

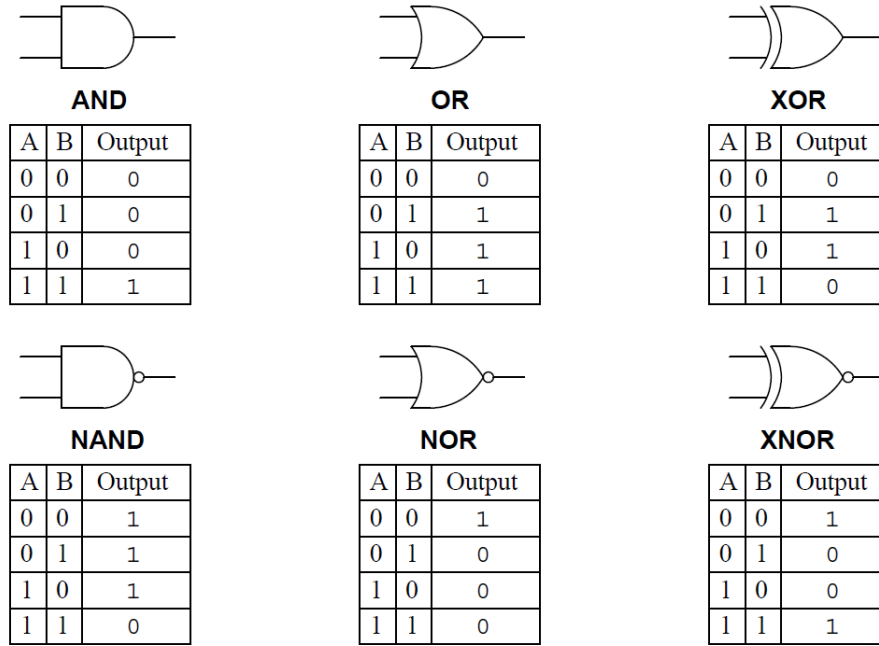


Figure 2: Classical logic gates¹

1.1.1 Quantum Logic Gates

Quantum logic gates on qubits are the analogue of classical logic gates on classical bits. These gates are the means by which the quantum states of the qubits are manipulated. They can be represented as unitary matrices. Any n -qubit computation can be broken down into single qubit and two qubit gates [5, 6],

which we will use later.

It is possible to simulate classical logic using quantum logic gates. The classical NOT gate is equivalent to the quantum X gate and the classical NAND gate is equivalent to the quantum CNOT gate. With these gates we can simulate other classical gates and thus classical logic circuits [7].

1.1.2 Superposition and Entanglement

As the number of qubits in the system increases, the number of possible states also increases. The quantum system can be in a superposition of these states. For example, in a two qubit system, the quantum system can be expressed as $|\Psi\rangle = \alpha_1 |00\rangle + \alpha_2 |01\rangle + \alpha_3 |10\rangle + \alpha_4 |11\rangle$. Like the single qubit case, $\alpha_i \in \mathbb{C}$ and $\sum_i |\alpha_i|^2 = 1$.

$$\begin{aligned} |\Psi\rangle &= |\Psi_1\rangle \otimes |\Psi_2\rangle = (\alpha_1 |0\rangle + \beta_1 |1\rangle) \otimes (\alpha_2 |0\rangle + \beta_2 |1\rangle) \\ &= \alpha_1 \alpha_2 |00\rangle + \alpha_1 \beta_2 |01\rangle + \beta_1 \alpha_2 |10\rangle + \beta_1 \beta_2 |11\rangle \end{aligned} \quad (2)$$

A two qubit system can also be described from the tensor product of individual qubit states in equation 2. However, taking into consideration a Bell state $|\Psi_B\rangle = 1/\sqrt{2}(|00\rangle + |11\rangle)$, the lack of $|01\rangle$ and $|10\rangle$ implies that either of $\alpha_1, \beta_2 = 0$ and either of $\beta_1, \alpha_2 = 0$. This leads to a contradiction as the presence of $|00\rangle$ and $|11\rangle$ implies that all of $\alpha_1, \beta_1, \alpha_2, \beta_2 \neq 0$. In this case, the system cannot be factored into a tensor product of individual qubits. This phenomena is known as entanglement.

In quantum information, entanglement is useful as a resource. Taking the example of the Bell state again, measurement collapses the system to either $|00\rangle$ or $|11\rangle$. If one qubit out of the pair is present, it is possible to discern the state of the other qubit from measurement. However, if one were to measure the system in the X-axis instead ($|+\rangle, |-\rangle$), where $|0\rangle = (|+\rangle + |-\rangle)/\sqrt{2}$ and $|1\rangle = (|+\rangle - |-\rangle)/\sqrt{2}$, one would get an entangled pair

$$|\Psi_B\rangle = \frac{1}{\sqrt{2}}(|++\rangle + |--\rangle) \quad (3)$$

If the two qubits are spread to different parties and they measure their individual qubit in the same axis, there is a correlation between their measurement results. However, if they measure in different axes, for example in Z and X, one may get $|0\rangle$ and the other $(|+\rangle + |-\rangle)/\sqrt{2}$, which cannot be discerned on a single measurement. This indistinguishable property in the orthogonality of states is a cornerstone of quantum information and cryptography [7].

1.2 Quantum Computation

With qubits and quantum logic gates, one can construct a quantum circuit to perform some computation. The Gottesman–Knill theorem shows that any

¹Figure taken from <https://instrumentationtools.com/logic-gates/>

quantum circuit that prepares and measures qubits in the computational basis and only uses gates from the Clifford group (X, Y, Z, H, CNOT) can be efficiently simulated on a classical computer [8]. Hence, we desire useful quantum algorithms that are difficult to emulate classically and have an advantage over classical algorithms. Here, we provide an example of such a quantum algorithm and the methodology of how to apply it to a useful problem.

1.2.1 Grover's Algorithm

As a quantum system can be expressed in the superposition of states, a gate operation on the system affects all these states at once. An algorithm or computation can be broken down into logic gates and this means that we can perform various operations in parallel with the superposed states.

An example to illustrate this is Grover's algorithm [9]. It operates as a search or inversion algorithm where given a function f and an output y , determine x such that $f(x) = y$. We first prepare the system in a superposition of all possible states Ψ . Next, we apply the oracle unitary based on f and y on the system. This oracle inverts the phase if the input state is x and does nothing otherwise. Then, we apply the Grover diffusion operator to the system. This operator diminishes the probability amplitude of s' , the superposition of states excluding state x , while amplifying the probability amplitude of x and inverts its phase back to normal. By repeating the oracle and diffusion steps enough times, the system will eventually reach state x . Grover's algorithm is given by

Algorithm 1 GROVERSEARCH(f, y, m)

```

Construct an oracle unitary  $U_w = I - 2|x\rangle\langle x|$ 
Prepare the system  $\Psi$  as a superposition of all states  $s$ 
for  $i$  in 1 to  $m$  do
    Apply  $U_f$  on  $\Psi$ 
    Apply the diffusion operator  $2|s\rangle\langle s| - I$  on  $\Psi$ 
end for
return Measure( $\Psi$ )

```

where m is the number of diffusion repetition steps and has upper limit $\lceil (\pi/4)N \rceil$ [9], N is the number of qubits and x is determined from the frequency of the output states of Ψ . The closer m is to the upper limit, the closer the initial state s will eventually evolve to x . The loop requires $O(\sqrt{N})$ iterations, where N is the size of the domain of f . The relaxation or diffusion operator uses 3 qubit Toffoli or CCNOT gates [10], which is not featured in the Clifford group and hence not subject to the Gottesman–Knill theorem. If f is some classical program, it can be compiled into a classical logic circuit and hence simulated on qubits.

Grover's algorithm is useful as a search algorithm. For example, in cryptography, a hash function takes in an input of variable size and gives an output of fixed size. This type of function is used in hiding or storing passwords or

secret keys and it is typically difficult to invert. Hence, Grover's algorithm can be used to efficiently find various inputs that give the same output, otherwise known as hash collision. One can describe it as a brute force approach with the superposition of all possible inputs.

1.2.2 3SAT

The Boolean SATisfiability problem (SAT) is a constraint satisfaction problem. Given a number of true/false (boolean) variables or literals and constraints or clauses, one has to find an assignment of literals such that the clause equates to true.

$$C = x_1 \wedge x_2 \wedge \overline{x_3}. \quad (4)$$

In equation 4, the clause C is described by the three variables x_1, x_2, x_3 and the AND/OR operators \wedge, \vee . Specifically, we require x_1, x_2 and the negation of x_3 to all be true for C to be true. Hence, the assignment to satisfy C is x_1, x_2 as true and x_3 as false.

$$\varphi = (x_1 \vee x_2) \wedge (\overline{x_1} \vee x_3). \quad (5)$$

A formula in conjunctive normal form (CNF) φ is a propositional formula in which all clauses have to be true and literals in clauses are joined together by only OR operators.

Of interest is the 3SAT problem where every clause has exactly three literals. By the Cook-Levin theorem, the 3SAT problem is NP-complete [11]. Nondeterministic Polynomial time (NP) problems are unlikely to be efficiently solvable but have solutions that can be verified in polynomial time $O(n^k)$, where n is the length of the input and k is a positive constant. NP-completeness implies that the problem is in NP and has a polynomial time reduction to another problem in NP. While 3SAT does not sound like a useful problem to solve, one can reduce an equally or less difficult useful problem to 3SAT to solve, such as Max Cut, Knapsack and Partition as shown by Karp [12]. Hence, if one has a 3SAT solver, one can use it to solve other difficult problems in NP.

$$\varphi = (x_1 \vee x_2 \vee x_3) \wedge (\overline{x_1} \vee x_3 \vee x_4). \quad (6)$$

Grover's algorithm can be used to solve the 3SAT problem in $O(1.414^n)$ time [13], where n is the length of the input. Given the CNF in equation 6, one constructs the following quantum circuit.

For an assignment to be true, $|\varphi C_1 C_2\rangle$ is required to be $|100\rangle$. From simulation the valid states for $|\varphi C_1 C_2 x_4 x_3 x_2 x_1\rangle$ are $|1000010\rangle, |1000100\rangle, |1000101\rangle, |1000110\rangle, |1000111\rangle, |1001001\rangle, |1001010\rangle, |1001011\rangle, |1001100\rangle, |1001101\rangle, |1001110\rangle, |1001111\rangle$. These states do indeed fulfill φ .

While it may not be a significant advantage over classical 3SAT solvers, this example shows the methodology of running a quantum algorithm.

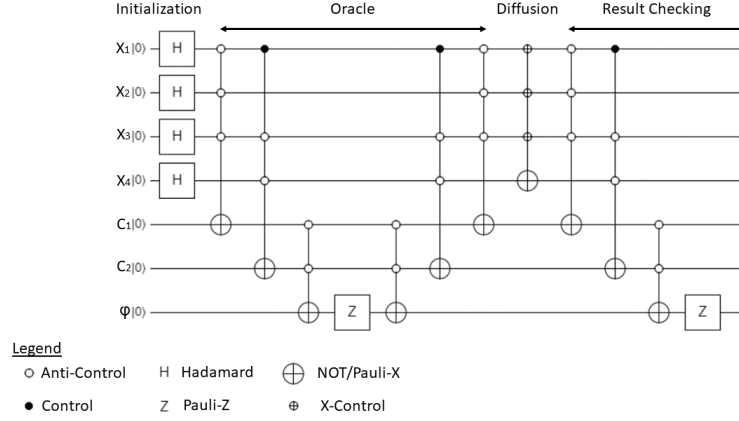


Figure 3: The literals x_i are prepared in a superposition of states s . Then, the clause are checked. If all x_1, x_2, x_3 are false, C_1 is flipped. If x_1 is true but both x_2, x_3 are false, C_2 is flipped. If C is false, then the clause is true. Next, φ is flipped if both C_1, C_2 are false and Z gate is used to flip the phase. Then, the clause checking is applied in reverse order followed by the Grover diffusion operator. Finally, the clause checking is applied at the end.

2 Superconducting Qubits

From Grover's algorithm, a sizable amount of qubits is required to span the input space. The need to repeat the Grover relaxation step multiple times also implies the need for qubits with long lifetime and high fidelity control gates for the length of the algorithm. From DiVincenzo's criteria [14], the following traits in realizing a quantum computation system are desired.

- A scalable physical system with well characterized qubits
- The ability to initialize the state of the qubits to a simple fiducial state
- Long relevant decoherence times
- A “universal” set of quantum gates
- A qubit-specific measurement capability

To implement a qubit, we require a quantum mechanical system with two energy levels. There are existing implementations in ion traps, nuclear magnetic resonance (NMR), electron spins, semiconducting qubits and superconducting qubits. As of writing, superconducting qubits are the more mature technology with 50+ qubit implementations available by Google [15], IBM [16] and China [17].

2.1 Superconducting LC Circuits

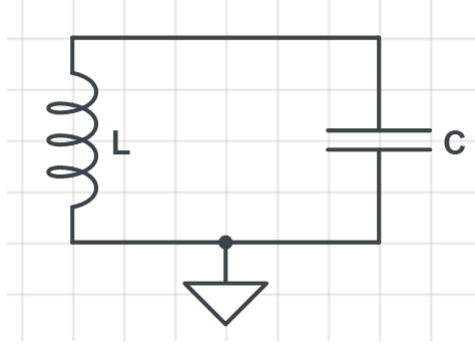


Figure 4: A simple LC circuit. Electrical energy in the circuit is stored in the electric field through the capacitor and magnetic flux in the inductor. The system forms a closed loop where energy oscillates between the capacitor and the inductor.

Superconductors have no dissipation and are thus ideal for implementing reversible quantum gate operations. The superconducting inductor-capacitor (LC) electrical circuit serves as the basis of the superconducting qubit. The energy T in the inductor is defined by

$$T = L\dot{Q}^2, \quad (7)$$

where L is the inductance and Q is the amount of charge. The energy U stored in the capacitor is given by

$$U = \frac{Q^2}{2C}, \quad (8)$$

where C is the capacitance. We can think of T as the kinetic energy of the system and U as the potential energy. The Lagrangian is defined as

$$\begin{aligned} \mathcal{L}(\dot{Q}, Q, t) &= T - U \\ &= \frac{L\dot{Q}^2}{2} - \frac{Q^2}{2C}. \end{aligned} \quad (9)$$

The Euler-Lagrange equation follows

$$\begin{aligned} \frac{\partial \mathcal{L}}{\partial Q} - \frac{d}{dt} \frac{\partial \mathcal{L}}{\partial \dot{Q}} &= 0, \\ -\frac{Q}{C} - L\ddot{Q} &= 0. \end{aligned} \quad (10)$$

For a LC circuit, the resonant frequency is $\omega = \frac{1}{\sqrt{LC}}$ and thus

$$\begin{aligned}\omega^2 Q + \ddot{Q} &= 0, \\ \ddot{Q} &= -\omega^2 Q,\end{aligned}\tag{11}$$

which is consistent with Kirchhoff's laws and shows simple harmonic oscillation. If Q is the 'position' variable of the oscillation, then the conjugate 'momentum' is obtained from

$$\begin{aligned}\frac{\partial \mathcal{L}}{\partial \dot{Q}} &= L\dot{Q} \\ &= \Phi,\end{aligned}\tag{12}$$

where Φ is the amount of flux through the inductor. Hence, the Hamiltonian of the system is defined as

$$\begin{aligned}\mathcal{H} &= \frac{L\dot{Q}^2}{2} + \frac{Q^2}{2C}, \\ &= \frac{\Phi^2}{2L} + \frac{Q^2}{2C}.\end{aligned}\tag{13}$$

Since Q and Φ are conjugate variables, they can be promoted to quantum operators \hat{Q} and $\hat{\Phi}$. This gives the Hamiltonian

$$\hat{\mathcal{H}} = \frac{\hat{\Phi}^2}{2L} + \frac{\hat{Q}^2}{2C},\tag{14}$$

and commutation relation

$$[\hat{\Phi}, \hat{Q}] = i\hbar.\tag{15}$$

The reduced flux is defined as $\phi \equiv 2\pi\Phi/\Phi_0$, where $\Phi_0 = h/(2e)$ is the magnetic flux quantum, and reduced charge as $n = Q/2e$ [18]. This establishes a more common representation of the system as

$$\hat{\mathcal{H}} = 4E_C n^2 + \frac{1}{2}E_L \phi^2,\tag{16}$$

where $E_C = e^2/(2C)$ is the charging energy of an elementary charge and $E_L = (\Phi_0/2\pi)^2/L$ is the inductive energy of an elementary charge. This Hamiltonian follows that of a quantum harmonic oscillator (QHO), and so we can rewrite it in raising and lowering operators a^\dagger and a

$$\hat{\mathcal{H}} = \hbar\omega_r \left(a^\dagger a + \frac{1}{2} \right)\tag{17}$$

where $\omega_r = \sqrt{8E_LE_C}/\hbar = 1/\sqrt{LC}$ is the resonant frequency of the system. Hence, this describes a quantum mechanical system with adjustable transition energy through modifying the capacitance and inductance of the LC circuit.

2.2 Transmon Qubit

The issue of using a QHO as a quantum computer is that the energy levels are equidistant and we do not have a dedicated two level system. Hence, in order to realize qubits, we need to introduce some anharmonicity to the system. This is done by replacing the inductor with a Josephson junction (JJ) [19], governed by the following Josephson relations

$$I = I_c \sin \phi, \quad (18)$$

$$V = \frac{\hbar}{2e} \frac{d\phi}{dt}, \quad (19)$$

where I_c is the critical current of the junction. The Hamiltonian is modified as follows

$$\hat{\mathcal{H}} = 4E_C n^2 - E_J \cos \phi, \quad (20)$$

where $E_C = e^2/(2[C_s + C_J])$, C_s being the shunt capacitance, C_J is the self-capacitance of the junction, and $E_J = I_c \Phi_0/2\pi$ is the Josephson energy. Hence, the cosine term in equation 20 gives us some nonlinearity. The dynamics of the system is then determined by the ratio E_J/E_C . In the regime $E_J < E_C$, the system is sensitive to charge noise and hence the other regime where $E_J \gg E_C$ is desired, which is the current approach to superconducting qubits.

To achieve this regime, E_C is made as small as possible by using a large capacitor to increase C_s . This approach gives rise to the transmission line shunted plasma oscillation qubit or transmon qubit [20].

The transmon has anharmonicity $\alpha = -E_C$ typically in the range of 100 to 300 MHz and the transition frequency between the ground state and first excited state given by [18]

$$\omega_q = (\sqrt{8E_J E_C} - E_C)/\hbar, \quad (21)$$

Typical values of ω_q for transmon qubits range from 2 GHz to 10 GHz, which means that the higher order transitions are well distanced from ω_q . α also limits the duration of qubit control pulses and hence higher anharmonicity is generally desired for a qubit.

2.2.1 Tunable Qubit

Another approach is to replace the JJ with a loop of two identical junctions, forming a direct current superconducting quantum interference device (dc-SQUID). The Hamiltonian of the system becomes

$$\hat{\mathcal{H}} = 4E_C n^2 - 2E_J |\cos \varphi_e| \cos \phi. \quad (22)$$

where $\varphi_e = \pi \Phi_0/\Phi_{ext}$ and Φ_{ext} is an externally applied magnetic flux. With this, the frequency of the qubit can be adjusted by varying the magnetic flux. Hence, interactions between qubits can be allowed to happen by tuning them

to the same frequency for them to exchange energy and disallowed by tuning them far apart.

2.3 Qubit Driving

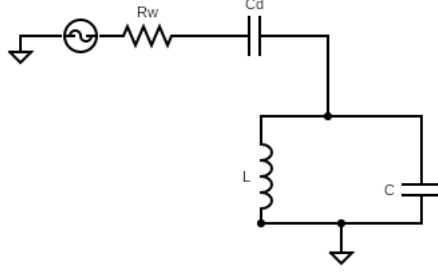


Figure 5: A microwave drive line capacitively coupled to a LC circuit. R_w is the resistance of the line and C_d is a capacitor between the qubit and the drive line.

To drive the qubit, the transmon is coupled to a microwave drive line in the manner of Figure 5. Under the rotating wave approximation, the driving Hamiltonian is as follows [18, 21].

$$\tilde{\mathcal{H}}_d = \Omega V_0 s(t) [(-\cos \phi \cos(\delta\omega t) + \sin \phi \sin(\delta\omega t))\sigma_x + (\cos \phi \sin(\delta\omega t) - \sin \phi \cos(\delta\omega t))\sigma_y], \quad (23)$$

where $\Omega = C_d/(C_s + C_J + C_d)Q_{zpf}$, $s(t)$ is the pulse envelope function, $Q_{zpf} = \sqrt{\hbar/2Z}$ is the zero-point charge fluctuations and $Z = \sqrt{L/C}$ is the impedance of the circuit to the ground, V_0 , ϕ and ω_d are the voltage, phase and frequency of the voltage source, $\delta\omega = \omega_q - \omega_d$ and σ are the Pauli matrices [18].

At the resonant frequency $\omega_d = \omega_q$, this becomes

$$\tilde{\mathcal{H}}_d = \Omega V_0 s(t) (-\cos \phi \sigma_x + \sin \phi \sigma_y). \quad (24)$$

To drive the qubit about the X-axis, ϕ is set to 0

$$\tilde{\mathcal{H}}_d = -\Omega V_0 s(t) \sigma_x, \quad (25)$$

and $\phi = \pi/2$ for the Y-axis.

$$\tilde{\mathcal{H}}_d = \Omega V_0 s(t) \sigma_y. \quad (26)$$

The Rabi frequency Ω_R is the frequency of driving the qubit about the Bloch sphere and is given by the following equation.

$$\Omega_R = \frac{C_d V_0 Q_{zpf}}{C_s + C_J + C_d}. \quad (27)$$

The parameters V_0, s, t, ϕ are externally controlled while the other parameters are designed at qubit fabrication. Hence, one can determine the parameters needed to drive the qubit transition between the ground state and the excited state from the Rabi period. This pulse is known as the π -pulse.

2.4 Qubit Readout

Current realizations of qubit readout involve capacitive coupling of the qubit to a resonator such as a microwave cavity or a coplanar waveguide. The Jaynes-Cummings Hamiltonian describes the interaction of the qubit and the resonator [18]

$$\mathcal{H}_{JC} = \hbar\omega_r \left(a^\dagger a + \frac{1}{2} \right) + \hbar\frac{\omega_q}{2}\sigma_z + \hbar g(\sigma_+ a + \sigma_- a^\dagger), \quad (28)$$

where ω_r is the frequency of the resonator, a^\dagger, a are the creation and annihilation operators for photons in the resonator, ω_q is the frequency of the qubit and σ_+, σ_- represent the operators for exciting and de-exciting the qubit and g is the coupling strength between the resonator and the qubit.

At the dispersive limit where the resonator and qubit frequencies are far apart and compared to the resonator linewidth κ , $\Delta = |\omega_q - \omega_r| \gg g, \kappa$, in this regime, the bare frequencies to 'repel' each other. We can approximate in first order the dispersive Hamiltonian to be

$$\mathcal{H}_{disp} = (\omega_r + \chi\sigma_z) \left(a^\dagger a + \frac{1}{2} \right) + \frac{\tilde{\omega}_q}{2}\sigma_z, \quad (29)$$

where $\chi = g^2/\Delta$ and $\tilde{\omega}_q = \omega_q + \chi$. This means that resonant frequency of the resonator will be shifted by 2χ when the qubit is excited, which allows us to discern the state of the qubit by probing the resonator. The further away the frequency is shifted, the easier it is to discern the qubit state as seen in Figure 6. Hence, higher values of χ are desired to improve the readout fidelity.

However, in practice we do not observe a shift of exactly 2χ in the frequency of the cavity as seen in Figure 6 where $\chi = 3.65$ MHz. This is due to the effect of higher order transitions neglected in Equation 29 since the superconducting qubit is an anharmonic oscillator and not a two level system. Additionally, there is a decrease in transmission at the resonant frequency from the ground state to the excited state. This is because the cavity mode is hybridized with the qubit and the excitation of the qubit causes a decrease in the quality factor of the cavity.

Furthermore, Equation 29 only holds when the photon occupation in the cavity is low. This is due to the nonlinearity of the hybridised cavity mode [22]. At high photon number $n = a^\dagger a \geq n_c \equiv \Delta^2/(4g^2)$, where n_c is the critical photon number, the approximation breaks down. We will discuss how this is characterized in the next chapter.

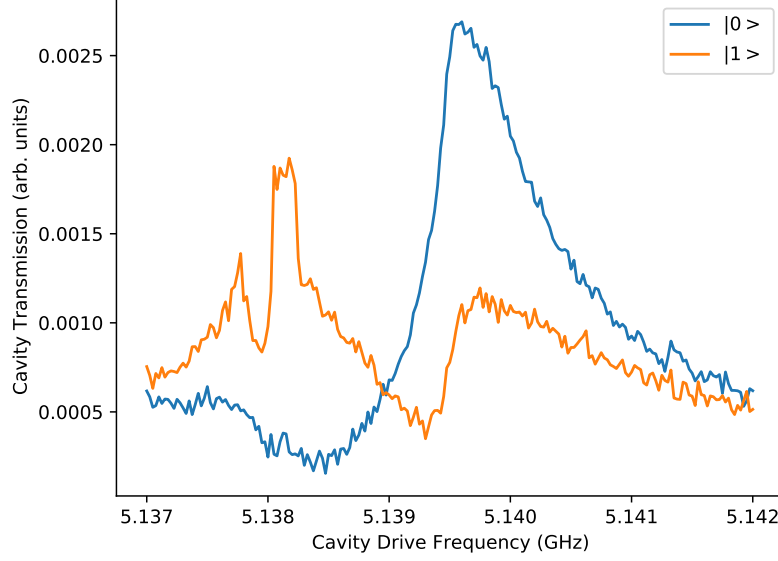


Figure 6: Signal transmission through a 3D microwave cavity versus signal frequency. The cavity is coupled to a transmon qubit. When the qubit is excited from the ground state (blue) to the excited state (orange), the signal transmission at the resonant frequency of 5.1398 GHz decreases, allowing for the determination of the two qubit states.

2.5 Qubit Decoherence

The Bloch-Redfield density matrix of the qubit is given by [18]

$$\rho_{BR} = \begin{bmatrix} 1 + (|\alpha|^2 - 1)e^{-\Gamma_1 t} & \alpha\beta^* e^{i\delta\omega t} e^{-\Gamma_2 t} \\ \alpha^*\beta e^{-i\delta\omega t} e^{-\Gamma_2 t} & |\beta|^2 e^{-\Gamma_1 t} \end{bmatrix} \quad (30)$$

where α, β are the probability amplitudes, $e^{-\Gamma_1 t}$ describes the longitudinal decay $\Gamma_1 = 1/T_1$ and $e^{-\Gamma_2 t}$ describes the transverse decay $\Gamma_2 = 1/T_2$ and $\delta\omega = \omega_q - \omega_d$. The T_1 decay is also known as the fluorescence lifetime where the qubit decays from $|1\rangle$ to $|0\rangle$ while T_2 is also known as the spin-spin relaxation time. These decay times are the limiting factor in the number of gate operations that can be performed.

Typical T_1, T_2 values of superconducting qubits are in the order of microseconds to tens of microseconds. With typical Rabi periods of 40 ns to 100 ns, there is a reasonable number of gate operations before qubit decoherence.

At time $t \gg T_1, T_2$, all terms except the top left fall to zero with the ground state reaching unity. Hence, this provides a method to initialize the qubit in the ground state after measurement. Another method for qubit reset involves

tuning the qubit frequency to a coupled resonator frequency and returning the qubit back to its idle frequency position after a period of time [23]. When the qubit is at the same energy of the resonator, energy exchange occurs and we move the qubit frequency when the qubit energy has been fully transferred to the resonator. The energy is then dissipated from the resonator and the qubit returns to the ground state. An additional method of qubit reset involves fast ring-up and ring-down processes in a readout cavity using microwave pulses [24].

In this chapter, we have shown how a superconducting qubit forms the basis of a qubit with the JJ introducing anharmonicity and how the circuit parameters can be engineered to affect the qubit energies. We have also explained how the qubit is driven and measured. As mentioned at the start of the chapter, the current state of the art superconducting qubit processors have a capacity of 50+ qubits. For fault tolerant quantum computing, we require qubits in the order of 10^3 or higher. Hence, the road ahead lies in scaling up the number of qubits, which is a fabrication and engineering challenge in packaging qubits closer together on a chip while minimizing microwave crosstalk. Additional challenges for an increasing number of qubits includes providing a refrigeration solution at 10 mK and qubit control hardware and logic. In the next chapter, I discuss how I have implemented qubit calibration and control on superconducting qubits in the NTU-CQT lab.

3 Quantum Processing Unit (QPU)

From the previous section, we have covered the theory behind superconducting qubits, which form the the quantum registers of a quantum computer. Here, we describe the QPU, a central platform for executing instructions on the quantum registers. This platform is also used to calibrate the gate operations and readout.

3.1 Infrastructure

The infrastructure needed to realize a superconducting qubit platform is divided into the cryogenic equipment and the room temperature electronics. Figure 7 describes the experimental setup in the NTU-CQT lab, which was initially set up by Alessandro Landro, Ranga Budoyo and Long Nyugen. In this section, I will discuss how I programmed and used the radio frequency (RF) electronics for qubit control and readout. The superconducting qubits used in this section were fabricated by Ranga Budoyo and Long Nyugen.

3.1.1 Signal Generation

As superconducting qubit and readout resonators used are typically in the 2-10GHz regime, we require microwave synthesis in that regime. Most conventional AWGs or Digital-to-Analog Converters (DACs) have sampling rate in the 1-6 GS/s regime and cannot produce the required frequencies directly. Field Programmable Gate Arrays (FPGAs) and Radio Frequency System on a Chip (RFSoc) platforms house DACs for signal generation. Similarly, ADCs or oscilloscopes typically do not have the sampling rate to sample the readout pulse directly.

Mixers are a popular approach to upconvert low frequency signals to high frequency and downconvert high frequency signals to low frequency [25].

Band-pass filters are used to filter out the LO leakage and unnecessary sidebands. The amplitude and phase for maximizing our desired sideband drift over time and thus require periodic calibration [25]. This is performed in our lab by running a Nelder-Mead optimization on the amplitude and phase of each I and Q sideband signal and maximizing the output power of the desired frequency through a spectrum analyzer as seen in Figure 8

Another approach is to use the Nyquist zones of a signal generator [26]. A discrete signal produced has higher frequency aliases in its Nyquist zones and we can use that to generate the high frequency pulses we desire.

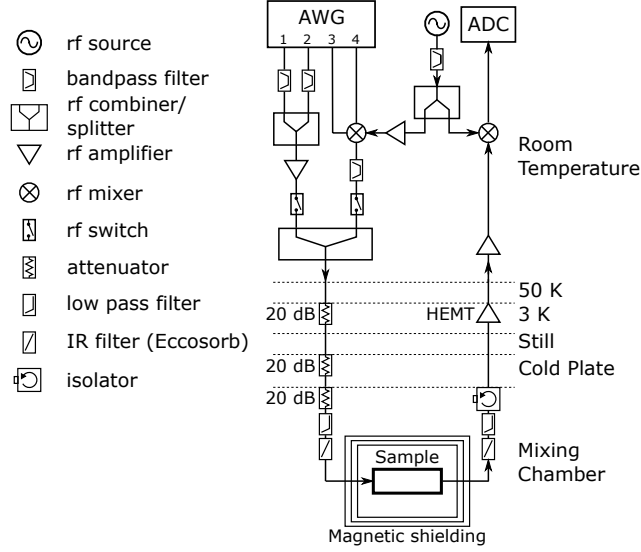


Figure 7: Qubit control and measurement setup in NTU-CQT. At the top-left, a Tektronix AWG5204 Arbitrary Waveform Generator (AWG) is used to synthesize the readout and qubit control pulses. The readout IQ pulses are upconverted by a mixer with a National Instruments QuicSyn Lite serving as a local oscillator (LO), which is marked as the RF source in the figure. The pulses are then sent into a Bluefors XLD dilution refrigerator. The refrigerator is separated into five stages: 50K stage, 3K stage, still, cold plate and mixing chamber, where the qubit sample is mounted. The readout signal exiting the sample is amplified by a High-Electron-Mobility-Transistor (HEMT) amplifier before being downconverted into IQ signals for readout by an AlazarTech ATS9371 Analog-to-Digital Converter (ADC). The same LO is used for readout upconversion and downconversion as a reference to reflect the phase change of the readout signal as it interferes with the readout resonator on the sample.

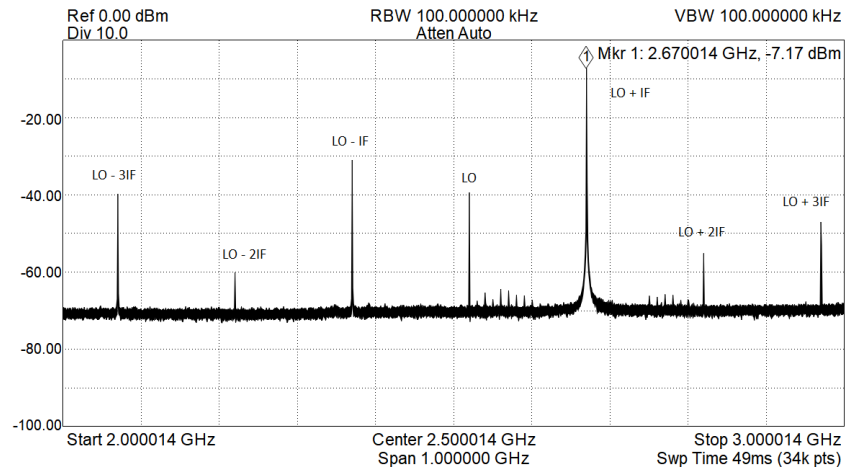
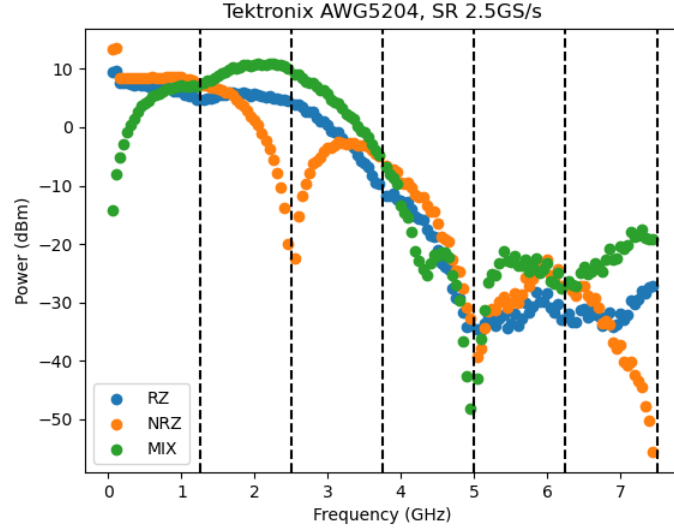
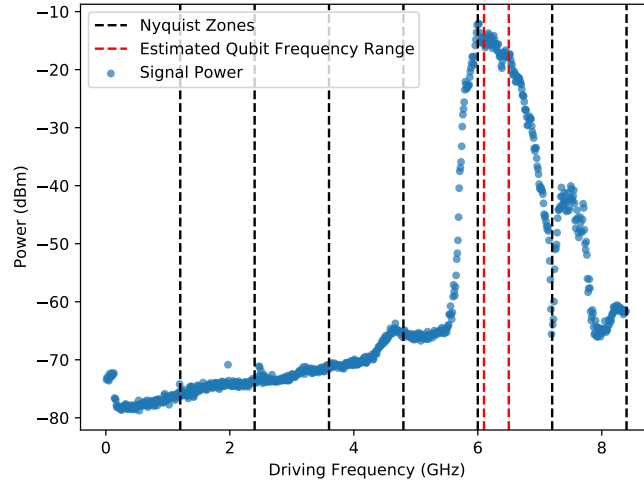


Figure 8: Frequency spectrum of a LO at 2.52 GHz and an IF frequency of 150 MHz. The amplitude and phase of the IQ signals are optimized to give the target upconversion at 2.67 GHz while minimizing the other sidebands and LO leakage.



(a)



(b)

Figure 9: Signal power versus signal frequency for a Tektronix AWG5204. The black dotted lines separate the Nyquist zones. (a) The points in blue represent the Return to Zero (RZ) DAC mode, while the orange represent Non Return to Zero (NRZ) mode and the green represents the mix mode. We first select which frequency range we want to operate at and then select the corresponding DAC mode for maximum power.(b) After we have selected the DAC mode, we use band-pass filters to isolate the intended working range in red.

3.1.2 Signal Readout

After the readout signal passes through the resonator, it is recorded in order to determine the change of the qubit. The high frequency readout signal is downconverted to lower frequency IQ signals through a mixer. These signals are recorded and parsed by the following function.

Algorithm 2 PARSEREADOUTIQ(I, Q, T, f_{IF})

```

Initialize  $I_t = Q_t = 0$ 
for  $i$  in 1 to  $|T|$  do
     $t = T[i]$ 
     $I_t = I_t + I[i] \cos(2\pi f_{IF} t)$ 
     $Q_t = Q_t + Q[i] \cos(2\pi f_{IF} t)$ 
end for
return  $I_t, Q_t$ 

```

The amplitude is calculated by $A = \sqrt{I_t^2 + Q_t^2}$ and phase $\phi = \text{atan2}(Q_t, I_t)$. With either of these two values, we can infer the state of the qubit. From Figure 6, if we send the readout signal at the resonant frequency of the cavity, we expect the resultant amplitude value to be high in state $|0\rangle$ and low in state $|1\rangle$.

3.1.3 Pulse Description

A pulse p has the following attributes

- AMPLITUDE
- DURATION
- FREQUENCY
- START TIME
- PHASE
- ENVELOPE

With regards to equation 22, V_0 corresponds to the amplitude, ω_d corresponds to the driving frequency, ϕ corresponds to the phase and s corresponds to the envelope function. This pulse representation will be used in the next chapter in the quantum circuit transpilation process.

3.1.4 Execution Workflow

An algorithm to use the hardware goes as follows

Algorithm 3 RUNEXPERIMENT(P, N)

```

Initialize an array  $M$  of size  $N$ .
Load pulses  $P$  into the AWG/DAC
for  $i$  in 1 to  $N$  do
    Arm the ADC/oscilloscope
    Play the pulse sequence
    Store the readout signal data in  $M[i]$ 
end for
return  $M$ 

```

where P is the pulse sequence containing the experiment pulses and N is the number of shots or averaging. The repetition rate of the loop depends on how fast we can reset the qubit system to the ground state and also the hardware logic. In our experiments in the NTU-CQT lab we typically use a rest period of $5T_1$ for the qubit to relax to the ground state.

To reduce the overhead, we can process the readout signal asynchronously while waiting for the next pulse sequence to execute and incoming readout signal. Typically, the readout signal duration is much shorter than the full pulse sequence and qubit reset protocol.

Algorithm 4 RUNEXPERIMENTWITHASYNC($P, N, t_{readout}$)

```

Prepare a  $N \times 2$  measurement vector  $M$ 
Generate sampling time array  $T$  based on  $t_{readout}$  and ADC sampling rate
Prepare an array of buffers  $B$  based on  $N$  and  $T$ 
Prepare pointers  $p_B$  for  $B$ ,  $p_b$  for  $B[p_B]$  and  $p_M$  for  $M$ 
Initialize counter  $m = 0$ 
Set ADC to sample for  $|T|$  samples
Set ADC to run ON SIGNAL ASYNC when a signal is received
Let the AWG/DAC repeat pulses  $P$  for  $N$  times
Wait for  $p_M$  to reach the end of  $M$ 
return  $M$ 

```

In this approach, we store the measurement data in a buffer b until it is full. Once it is full, we move on to the next buffer and process b in the free time between readout signals. Then, we extract and process I and Q for each measurement in b . The processed results are stored in M . Once we are done with all measurements in b , we clean it to be used for later measurements. We do this for all shots N . This allows us to avoid storing the large signal data S with size $|T|$ for a total of $2N|T|$ and instead store the final measurement with size $2N$. This also allows us to make use of the free intervals between measurements.

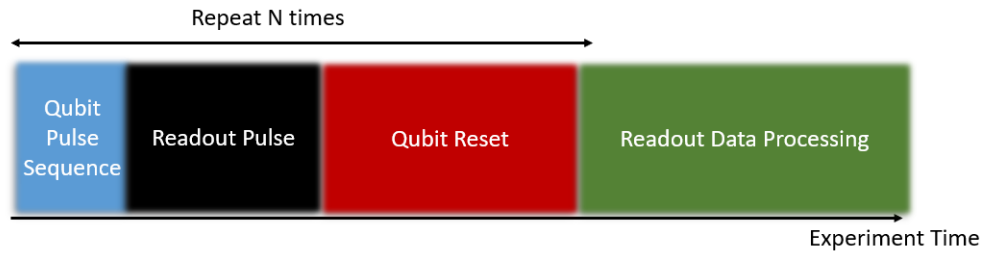
Algorithm 5 ONSIGNALASYNC(S)

Extract I and Q from S
Select buffer $B[p_B]$ as b
Store I and Q in b starting from p_b
Move p_b to the end of the new data in b
 $m = m + 1$
if b is full or $m = N$ **then**
 Move p_B to the next available buffer
 Reset p_b to the start of the new buffer
 Remove b from B
 Run asynchronously ONBUFFERFULLASYNC(b)
end if

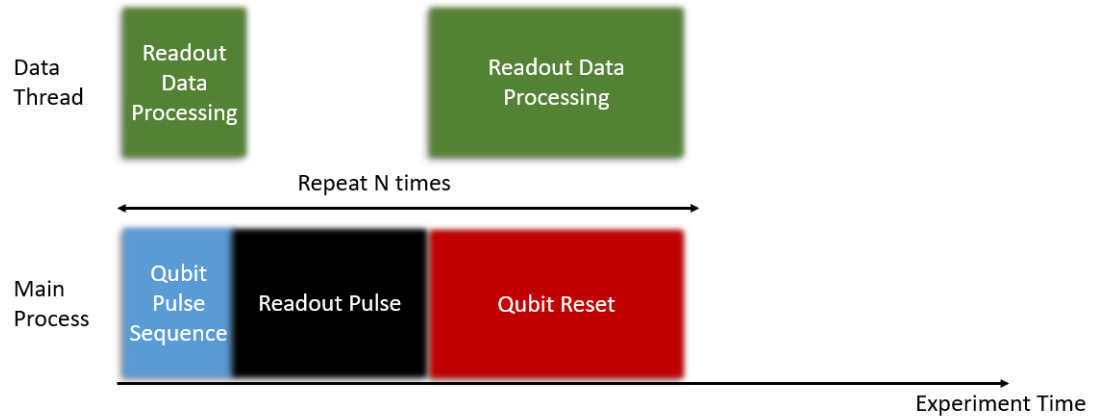
Algorithm 6 ONBUFFERFULLASYNC(b)

Let $k = 0$
while $k < |b|$ and $b[k]$ is not null **do**
 Extract I and Q from $b[k : k + 2|T|]$
 Store PARSEREADOUTIQ(I, Q, T, f_{IF}) in $M[p_M]$
 $k = k + 2|T|$
 Move p_M to the next entry in M
end while
Clean b and add it back to B

This approach in algorithms 5 and 6 can be implemented on programmable PCI ADCs, FPGAs and RFSocS with sufficient memory but not oscilloscopes as they do not support programmable asynchronous logic. Since PARSEREADOUTIQ has time complexity scaling linearly to the number of samples, one can further extend this procedure on single measurements to translate the I_t and Q_t values into qubit states for further memory reduction, if the downtime period is sufficiently long enough. This step is necessary for quantum error correction in identifying qubit states during the pulse sequence with the latency being dependent on processor speed and computation complexity.



(a)



(b)

Figure 10: (a) Synchronous experiment procedure and (b) asynchronous experiment handling. In (a), the readout data is only parsed at the end of N shots. In (b), the readout data is being processed in the waiting time between readout pulses until all N shots are completed.

3.2 Qubit Calibration

After describing how the QPU is used to carry out a qubit experiment, we move on to the experiment procedures of using the QPU to characterize and calibrate the qubit system.

3.2.1 Resonator Spectroscopy

First, the readout resonator coupled to the qubit is characterized. As discussed in Section 2.4, the resonant frequency of the resonator depends on the photon occupation and the state of the qubit. We want to determine the dispersive frequency of the cavity and the dispersive shift and the power where we enter the dispersive regime $n < n_c$. This is typically done with a Vector Network Analyzer (VNA).

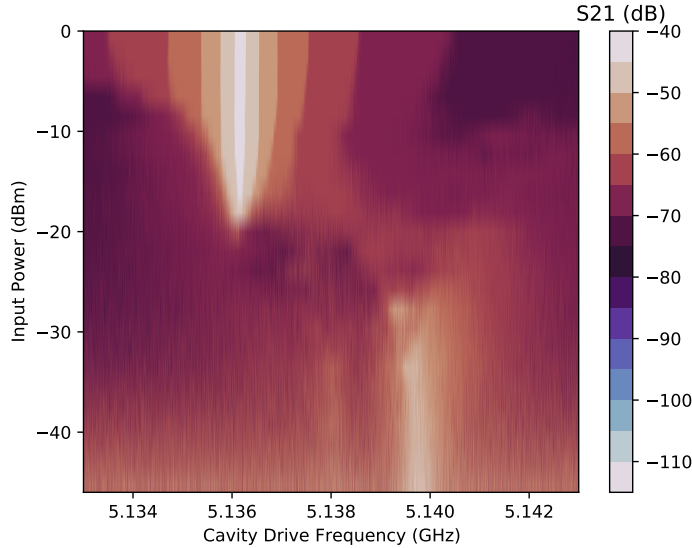


Figure 11: Transmission (S21) for input signal power versus signal frequency for a microwave cavity coupled to a transmon qubit. At higher powers of > -20 dBm, we find the cavity resonant frequency at 5.1361 GHz. At lower powers of < -30 dBm, the resonant frequency is dispersively shifted to 5.1398 GHz.

In Figure 11, we sweep across a range of driving powers for a microwave cavity coupled to a transmon qubit and notice that we enter the dispersive regime when the input power is roughly less than -30 dBm. For qubit readout, we use signal driving power in this regime and position the frequency of the readout signal at the dispersive resonant frequency. As seen from Figure 6, this frequency provides good contrast between the two states of the qubit. Algorithm 7 describes the methodology of how this is performed in calibration.

Algorithm 7 READOUTSPECTROSCOPY(f_{center}, f_{span})

Set the VNA to scan about f_{center} with span f_{span}
Scan the resonator at high power and fit the peak
Initialize the high power cavity resonant frequency ω_{rbare} as the center frequency of the peak fit
Scan the resonator at low power and fit the peak
Initialize the dispersively shifted cavity resonant frequency ω_r as the center frequency of the peak fit
 $\chi = |\omega_r - \omega_{rbare}|$
return ω_r, χ

3.2.2 Qubit Characterization

Next, we want to determine the frequency of the qubit. We drive the qubit for qubit for an extended period of time ($t \gg T_2$) where the qubit decoheres to the mixed state and sweep the driving frequency to find large transmission changes which will correspond to a transition.

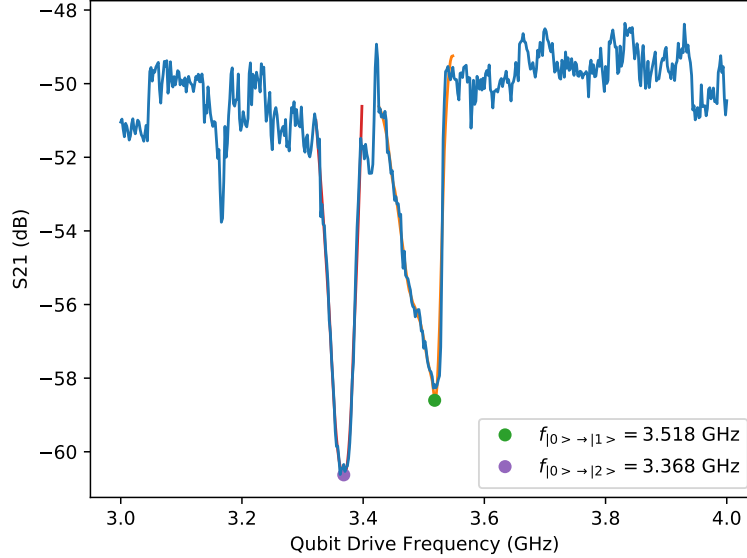


Figure 12: Transmission of a microwave cavity at its dispersive resonant frequency versus qubit drive frequency performed with a VNA. We observe the first excited state transition at 3.518 GHz. We also observe the two photon process $|0\rangle$ to $|2\rangle$ at 3.368 GHz.

If enough power is used, we may observe the two photon transition from $|0\rangle$

to $|2\rangle$ as seen in Figure 12.

$$\begin{aligned}\omega_{02} &\approx \omega_{01} + \omega_{12} \\ &\approx 2\omega_{01} - \alpha,\end{aligned}\tag{31}$$

where α is the anharmonicity. Then the two photon frequency would be approximately

$$\omega'_{02} \approx \omega_{01} - \alpha/2,\tag{32}$$

Hence, this allows us to identify the $|0\rangle$ to $|1\rangle$ transition in Figure 12 as 3.518 GHz since it will have the higher frequency. As discussed in the previous chapter, the state of the qubit causes a shift in the resonant frequency, which results in the decrease of transmission in the readout signal. Here, we can observe the higher energy transition pulling the cavity frequency further away, which leads to further drop in transmission. Finally, the experiment procedure with the QPU is described in Algorithm 8.

Algorithm 8 QUBITSPECTROSCOPY($f_{start}, f_{end}, f_{step}, p_{readout}, N$)

```

Initialize result array  $R$ 
Initialize pulse sequence  $P$  and add  $p_{readout}$  to  $P$ 
for  $f_Q$  in  $f_{start}$  to  $f_{end}$  with step  $f_{step}$  do
    Drive the qubit at frequency  $f_Q$ 
     $M = \text{RUNEXPERIMENT}(P, N)$ 
    Average and convert  $M$  into amplitude  $A$ 
    Append  $R$  with  $A$ 
end for
Choose the peak based on the anharmonicity
Fit the peak and obtain qubit frequency  $\omega_Q$ 
return  $\omega_Q$ 

```

We perform QUBITSPECTROSCOPY with the QPU instead of the VNA in Figure 13. There is a shift in transition frequencies between the two setups as the the QPU uses short readout pulses while the S21 measurement from the VNA continuously drives the cavity and causes an AC Stark shift in the transition frequencies. Now that we have identified the qubit frequency at 3.538 GHz, we can use it to characterize the π -pulse.

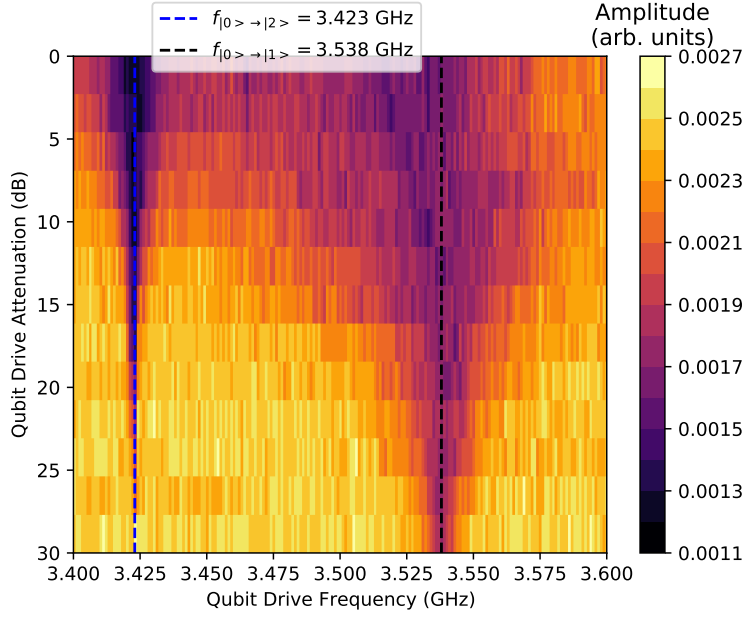


Figure 13: QUBITSPECTROSCOPY performed with the QPU while varying the driving attenuation of the qubit. We observe the first excited state transition at 3.538 GHz and the two photon process $|0\rangle$ to $|2\rangle$ at 3.423 GHz. As the driving power decreases, the linewidth of each transition decreases. This phenomenon is known as power broadening.

3.2.3 Rabi Spectroscopy



Figure 14: Pulse sequence of a Rabi experiment. We send the qubit drive pulse and the readout pulse and vary the duration of the qubit drive pulse.

After we have determined the approximate qubit frequency from the above spectroscopy methods, we can adjust the following parameters: readout drive

power, qubit drive power, readout drive frequency, qubit drive frequency in the Rabi sequence, assuming we keep the readout duration constant. From each sequence, we obtain the Rabi frequency and the separation between the ground and excited state. We desire the set of parameters that give us the greatest separation between the two states of the qubit for readout fidelity. This can be done with a classical optimization approach [27].

Algorithm 9 RABISPECTROSCOPYSWEEP TIME($\omega_Q, A_Q, T_{end}, T_{step}, p_{readout}, N$)

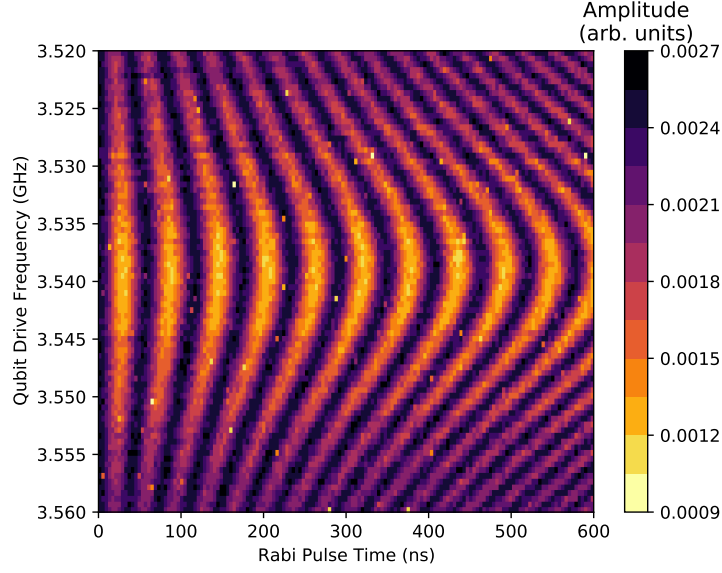
```

Initialize result array  $R$ 
Generate array  $T$  from 0 to  $T_{end}$  evenly spaced by step  $T_{step}$ 
for  $t$  in  $T$  do
    Initialize pulse sequence  $P$ 
    Prepare driving pulse  $p$  with amplitude  $A_Q$ , frequency  $\omega_Q$  and duration  $t$ .
    Set readout pulse  $p_{readout}$  to play at time  $t$ 
    Add  $p$  and  $p_{readout}$  to  $P$ 
     $M = \text{RUNEXPERIMENT}(P, N)$ 
    Average and convert  $M$  into amplitude  $A$ 
    Append  $R$  with  $A$ 
end for
 $\Omega_{Rabi}, A_R = \text{FITRABI}(T, R)$ 
return  $\Omega_R, A_R$ 

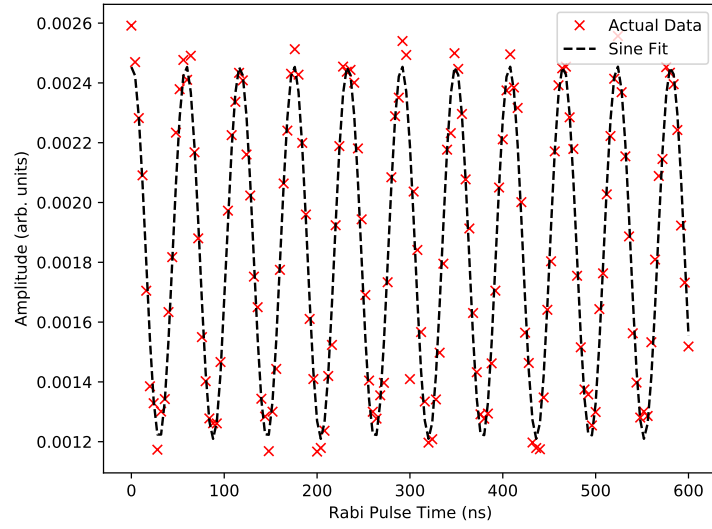
```

In RABISPECTROSCOPYSWEEP TIME, we sweep pulse duration from 0 to T_{end} and record the amplitude for each duration step. We then fit the amplitude versus pulse duration sweep curve to a sine wave and extract the fit amplitude and frequency. The state separation is given by A_{Rabi} and Rabi frequency given by Ω_{Rabi} .

In Figure 15, we perform the Rabi sequence on the qubit used in Figures 6, 11 and 12 using a guessed frequency of 3.538 GHz and 20 dB of attenuation. This gives a Rabi frequency of 17.2 MHz and π -pulse duration of 29 ns in Figure 15b. The chevron pattern in Figure 15a is obtained from repeating the Rabi sequence across different frequencies. However, it is difficult to estimate the qubit resonant frequency from the broad linewidth of the chevron pattern.



(a)



(b)

Figure 15: (a) Rabi chevron pattern with qubit driving frequency versus pulse duration. As the driving frequency tends towards the resonant frequency, the Rabi oscillation frequency decreases. (b) Rabi oscillation at the resonant frequency. The oscillation is fitted to a sine wave and the π -pulse duration is determined from the period of the oscillation.

3.2.4 Ramsey Spectroscopy

Ramsey spectroscopy is a technique to determine the actual qubit frequency [28]. Typically, the linewidth of the qubit from earlier spectroscopy is too broad to determine the exact qubit frequency. Hence, this method is used to find the detuning Δ between the driving frequency ω_d and qubit frequency ω_q . Two $\pi/2$ pulses are sent to the qubit separated by a flight time τ before the system is measured.

Algorithm 10 RAMSEYSPECTROSCOPY($p_{\pi/2}, p_{readout}, \tau_{end}, \tau_{step}, N$)

```

Initialize result array  $R$ 
Generate array  $T$  from 0 to  $\tau_{end}$  evenly spaced by step  $\tau_{step}$ 
for  $\tau$  in  $T$  do
    Initialize pulse sequence  $P$ 
    Initialize  $p_1 = p_2 = p_{\pi/2}$ 
    Set pulse  $p_2$  to play after  $p_1$  and  $\tau$ 
    Set pulse  $p_{readout}$  to play after  $p_2$ 
    Add  $p_1, p_2, p_{readout}$  to  $P$ 
     $M = \text{RUNEXPERIMENT}(P, N)$ 
    Average and convert  $M$  into amplitude  $A$ 
    Append  $R$  with  $A$ 
end for
 $\Delta = \text{FITRAMSEY}(T, R)$ 
return  $\Delta$ 

```

From the Rabi frequency, we get the π -pulse duration as $t_\pi = 1/\omega_{Rabi}$. Then, we generate the $\pi/2$ -pulse with amplitude A_Q , frequency f_Q and duration $t_\pi/2$. An approximate value of half the duration of the π -pulse is used for the $\pi/2$ -pulse. Then, we use this pulse and sweep the flight time τ to obtain Δ through fitting with a sine curve. We correct for the resonant frequency with f_Q and Δ .

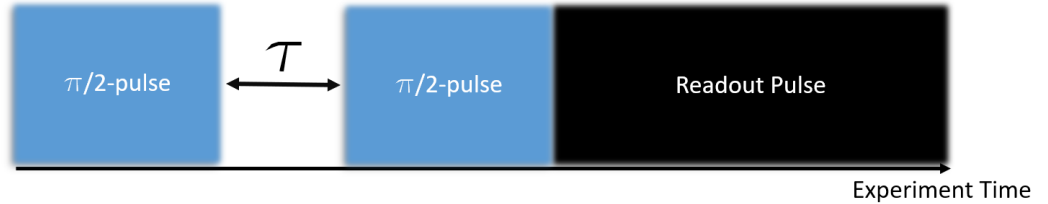


Figure 16: Pulse sequence of a Ramsey experiment. We send two $\pi/2$ -pulses separated by a delay τ and readout pulse to the system and vary τ .

By using a frequency detuning Δ in the Ramsey sequence, we obtain an oscillation with frequency close to Δ . This can be seen in Figure 17, where we

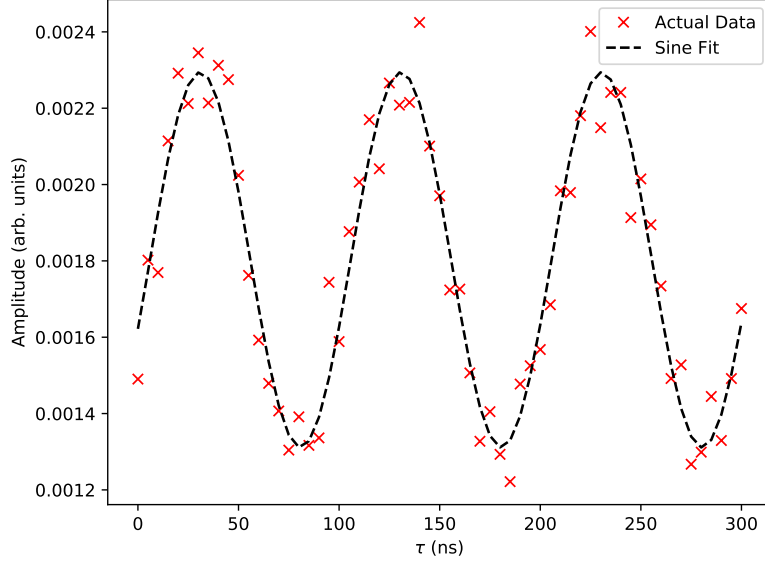


Figure 17: Ramsey oscillation with a detuning of 10 MHz. The qubit driving frequency was estimated at 3.538 GHz and the Ramsey oscillation has a frequency of 10.02 MHz. Hence, the new qubit frequency is set as 3.53798 GHz.

use a detuning of 10 MHz and obtained a Ramsey frequency of 10.02 MHz. The difference between the frequency and the detuning gives us a better estimate of the qubit frequency at 3.53798 GHz. Afterwards, we run the Rabi procedure again to get the π -pulse at the resonant frequency. Hence, a fresh calibration procedure proceeds as follows

Algorithm 11 FRESHCALIBRATION(R_{guess}, Q_{guess}, N)

$\omega_r, \chi = \text{READOUTSPECTROSCOPY}(R_{guess}, N)$
Initialize the readout pulse $p_{readout}$ with default settings and frequency ω_r
 $\omega_Q = \text{QUBITSPECTROSCOPY}(Q_{guess}, p_{readout}, N)$
 $\omega_R, A_R = \text{RABISPECTROSCOPYSWEEPTIME}(\omega_Q, Rabi_{default}, p_{readout}, N)$
Initialize the π -pulse with duration $1/2\omega_R$ and frequency ω_Q
Generate the $\pi/2$ -pulse $p_{\pi/2}$
 $\Delta = \text{RAMSEYSPECTROSCOPY}(p_{\pi/2}, p_{readout}, Ramsey_{default}, N)$
Determine the qubit frequency ω'_Q from ω_Q and Δ
Optimize $p_{readout}, A_Q$ with ω'_Q using $\text{RABISPECTROSCOPYSWEEPTIME}$
Obtain the new π -pulse p_π from the optimization
return $p_{readout}, p_\pi$

With repeated calibration, we can simply repeat the last optimization as we already have knowledge of the qubit frequency, readout pulse and driving

amplitude.

3.2.5 Two Qubit Interaction

With superconducting qubits, one can engineer interactions between coupled qubits either with magnetic flux [29] or with microwave pulses [30, 31]. With magnetic flux, an interaction is formed by having qubits of similar energy level interact and exchange energy. This action forms a SWAP operation. On Google’s Sycamore processor, this is done by with fixed-frequency bus resonator between qubits and biasing flux tunable transmon qubits to the bus resonator frequency [15]. IBM does the opposite with fixed frequency transmon qubits and flux tunable bus resonators [32].

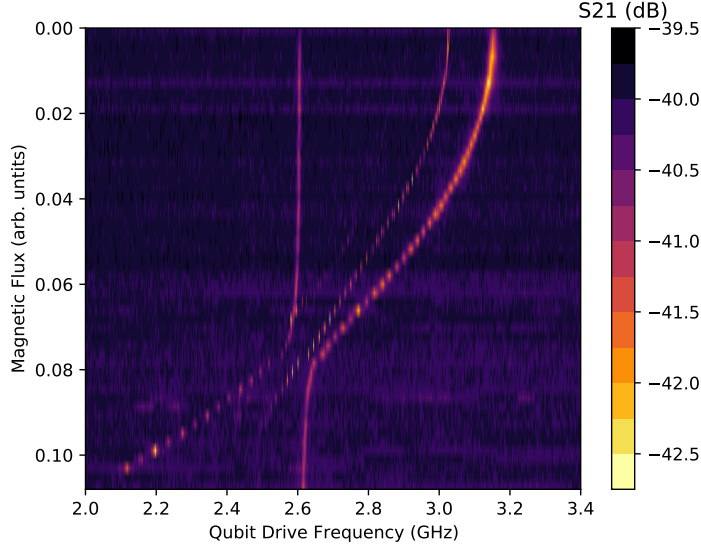


Figure 18: Flux versus qubit frequency in a system of a fixed frequency qubit and flux tunable qubit coupled to a microwave cavity. The fixed qubit has resonant frequency 2.605 GHz while the maximum frequency of the tunable qubit is at 3.145 GHz. As the flux increases, the frequency of the tunable qubit decreases to near that of the fixed qubit, where we observe an avoided level crossing.

An example of the microwave approach is the **S**peeding up **W**ave forms by **I**nducing **P**hases to **H**armful **T**ransitions (SWIPHT) gate, which functions as a reversed Controlled-NOT (CNOT) gate [30, 31].

Premaratne [31] defines the pulse time for the SWIPHT gate as

$$\tau_g = \frac{5.87}{2|\chi_{qq}|}, \quad (33)$$

Input		Output	
Control	Target	Control	Target
0	0	0	1
0	1	0	0
1	0	1	0
1	1	1	1

Table 2: Truth table of the SWIPHT gate. This gate functions as a reversed CNOT as it flips the target qubit if the state of the control qubit is $|0\rangle$ instead of $|1\rangle$ and does nothing otherwise.

$$\chi_{qq} = \frac{g_{qq}^2}{\Delta_q}, \quad (34)$$

where χ_{qq} is a coupling term between qubits, g_{qq} is the two qubit coupling strength and Δ_q is the frequency difference between the two qubits. and the pulse shaping as

$$\Omega = \frac{\ddot{\gamma}}{\sqrt{\chi_{qq}^2 - \dot{\gamma}^2}} - 2\sqrt{\chi_{qq}^2 - \dot{\gamma}^2} \cot 2\gamma, \quad (35)$$

where

$$\gamma(t) = 138.9 \left(\frac{t}{\tau_g} \right)^4 \left(1 - \frac{t}{\tau_g} \right)^4 + \frac{\pi}{4}. \quad (36)$$

We then calibrate the amplitude of the SWIPHT gate pulse. From Figure 20, we choose an amplitude of 0.55 V at which state inversion happens for $|00\rangle$ and $|01\rangle$. Unfortunately, due to the high qubit-qubit coupling of 66 MHz, there was leakage acting on the state $|01\rangle$, which caused the amplitude to shift. Later, we will introduce and use Quantum State Tomography to characterize a Bell state created with this gate.

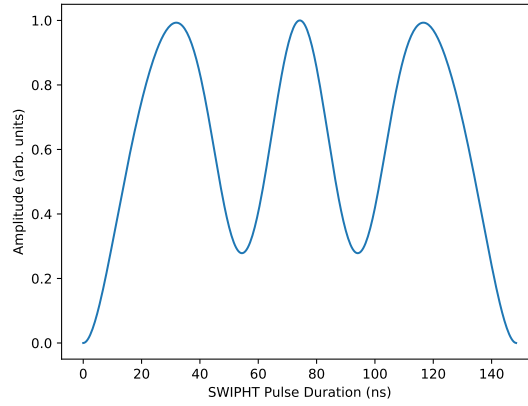


Figure 19: SWIPHT Pulse Envelope for a two qubit system with frequencies 3.0636 GHz and 3.2840 GHz coupled to a microwave cavity. This pulse envelope aims to drive the intended transition in Table 2 while inducing a phase shift in the undesired transitions. With a coupling strength of 66 MHz, the SWIPHT gate duration is 148.5 ns.

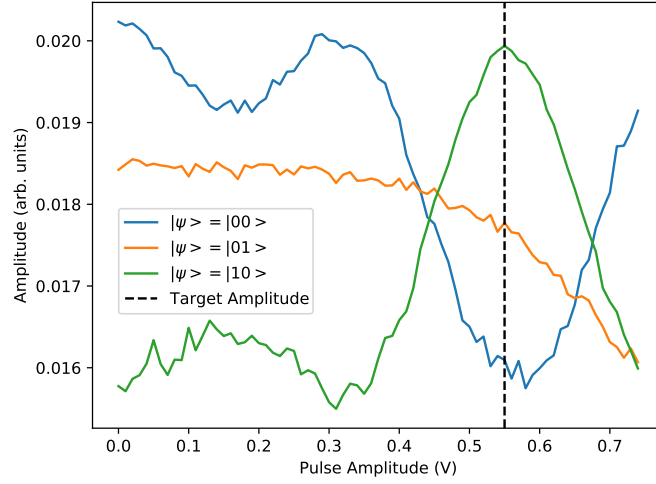


Figure 20: Transmission through the cavity versus amplitude of the SWIPHT gate with duration 46 ns applied at different states of the system in Figure 19. The first qubit is the target qubit while the second is the control. The blue data corresponds to the system initialized at $|00\rangle$, orange corresponds to the system at $|01\rangle$ and green corresponds to the system at $|10\rangle$. At low pulse power, the system remains in their initial states. We aim to select the pulse amplitude where $|00\rangle$ inverts with $|10\rangle$ while $|01\rangle$ remains the same.

3.3 Qubit Measurements

3.3.1 Quantum State Tomography

Quantum state tomography (QST) can be used to reconstruct the density matrix of a system [33, 34]. Given an arbitrary density matrix ρ for a single qubit, where

$$\rho = \begin{pmatrix} p_1 & p_2 \\ p_3 & p_4 \end{pmatrix},$$

and an arbitrary measurement operator M_1 , where

$$M_1 = \begin{pmatrix} m_{11} & m_{13} \\ m_{12} & m_{14} \end{pmatrix},$$

one obtains the expectation value $P_1 = \text{Tr}(M_1\rho) = p_1m_{11} + p_2m_{12} + p_3m_{13} + p_4m_{14}$. By constructing three other operators, one has

$$A = \begin{pmatrix} m_{11} & m_{12} & m_{13} & m_{14} \\ m_{21} & m_{22} & m_{23} & m_{24} \\ m_{31} & m_{32} & m_{33} & m_{34} \\ m_{41} & m_{42} & m_{43} & m_{44} \end{pmatrix}, X = \begin{pmatrix} p_1 \\ p_2 \\ p_3 \\ p_4 \end{pmatrix} \text{ and } B = \begin{pmatrix} P_1 \\ P_2 \\ P_3 \\ P_4 \end{pmatrix},$$

to construct the linear equation $AX = B$. Since M_i is known and P_i can be determined, one can obtain ρ by solving the set of linear equations. The probability of each state is obtained from the diagonal terms of the density matrix.

For two qubits, the set G of 16 measurement operators M_i is listed in Figure 21. Before applying the readout pulse, the corresponding prerotation is applied. Maximum Likelihood Estimation (MLE) can be used to find the best fit for ρ [35]. By applying further tensor products, one can extend state tomography to a n -qubit system.

For systems with low readout signal to noise ratio (SNR), resolving the state of the system may be difficult for a single shot. Hence, the readout signals are averaged in this scenario to improve the SNR. However, when the result is averaged, any correlation between qubits cannot be recovered. QST can be used in this scenario to recover the correlation from the reconstructed density matrix. However, the number of prerotations requires scales by $O(4^n)$, where n is the number of qubits, which increases the time required to determine the qubit state.

The QST analysis in the NTU-CQT lab was implemented by Long Nyugen while I implemented the prerotation procedure. In Figure 22, we performed QST on a two-qubit system used in Figure 20 initialized in the Bell state $1/\sqrt{2}(|10\rangle + i|01\rangle)$ and obtained a state fidelity of 85.9%. This state was prepared using a $-X/2$ gate on the second qubit to form $1/\sqrt{2}(|00\rangle + i|01\rangle)$, followed by the SWIPT gate using the second qubit as control and first as target. This flips the $|00\rangle$ state to $|10\rangle$ while leaving the $|01\rangle$ unperturbed. Since we have demonstrated the use of QST in identifying entangled states, we will use it in a later part to explore entanglement correlation.

	Prerotation	Measurement Operator, G
M_{01}	$I \otimes I$	$II + ZI + IZ + ZZ$
M_{02}	$R_x^\pi \otimes I$	$II - ZI + IZ - ZZ$
M_{03}	$I \otimes R_x^\pi$	$II + ZI - IZ - ZZ$
M_{04}	$R_x^{\pi/2} \otimes I$	$II + YI + IZ + YZ$
M_{05}	$R_x^{\pi/2} \otimes R_x^{\pi/2}$	$II + YI + IY + YY$
M_{06}	$R_x^{\pi/2} \otimes R_y^{\pi/2}$	$II + YI - IX - YX$
M_{07}	$R_x^{\pi/2} \otimes R_x^\pi$	$II + YI - IZ - YZ$
M_{08}	$R_y^{\pi/2} \otimes I$	$II - XI + IZ - XZ$
M_{09}	$R_y^{\pi/2} \otimes R_x^{\pi/2}$	$II - XI + IY - XY$
M_{10}	$R_y^{\pi/2} \otimes R_y^{\pi/2}$	$II - XI - IX + XX$
M_{11}	$R_y^{\pi/2} \otimes R_x^\pi$	$II - XI - IZ + XZ$
M_{12}	$I \otimes R_x^{\pi/2}$	$II + ZI + IY + ZY$
M_{13}	$R_x^\pi \otimes R_x^{\pi/2}$	$II - ZI + IY + ZY$
M_{14}	$I \otimes R_y^{\pi/2}$	$II + ZI - IX + ZX$
M_{15}	$R_x^\pi \otimes R_y^{\pi/2}$	$II - ZI - IX + ZX$
M_{16}	$R_x^\pi \otimes R_x^\pi$	$II - ZI - IZ + ZZ$

Figure 21: A set of measurement operators and prerotations that can be used in two qubit state tomography

3.3.2 Decoherence

As the relaxation times impact the number of operations, it is essential to determine them in order to understand the impact on the quantum circuit. The T_1 time is measured by applying a π -pulse to the qubit, waiting for time τ before measurement.

Algorithm 12 T1CHARACTERIZATION($p_\pi, p_{readout}, \tau_{end}, \tau_{step}, N$)

```

Initialize result array  $R$ 
Generate array  $T$  from 0 to  $\tau_{end}$  evenly spaced by step  $\tau_{step}$ 
for  $\tau$  in  $T$  do
    Initialize pulse sequence  $P$ 
    Set pulse  $p_{readout}$  to play after  $p_\pi$  and  $\tau$ 
    Add  $p$  and  $p_{readout}$  to  $P$ 
     $M = \text{RUNEXPERIMENT}(P, N)$ 
    Average and convert  $M$  into amplitude  $A$ 
    Append  $R$  with  $A$ 
end for
 $T_1 = \text{FIT}T1(T, R)$ 
return  $T_1$ 

```

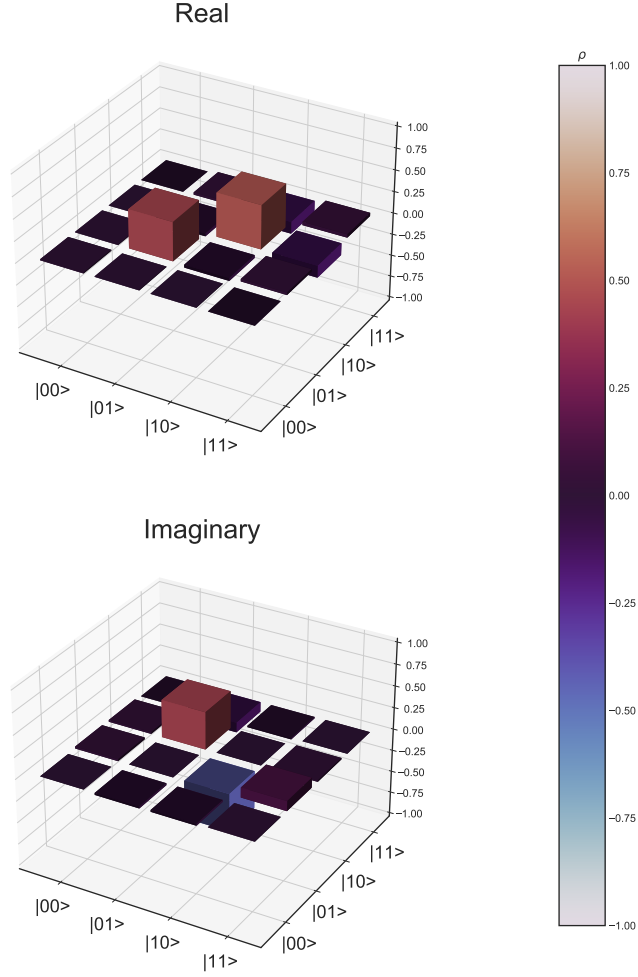


Figure 22: QST of the Bell state $1/\sqrt{2}(|10\rangle + i|01\rangle)$. A $-X/2$ gate with duration 56 ns followed by the SWIPHT gate was used to initialize the state before the sequence of 16 prerotations in Figure 21 was applied. The top plot represents the real component of the density matrix ρ while the bottom represents the imaginary component. The state fidelity is 85.9%.



Figure 23: Pulse sequence of a T_1 experiment. A π -pulse and readout pulse is sent to the system and the delay τ is varied between the two pulses.

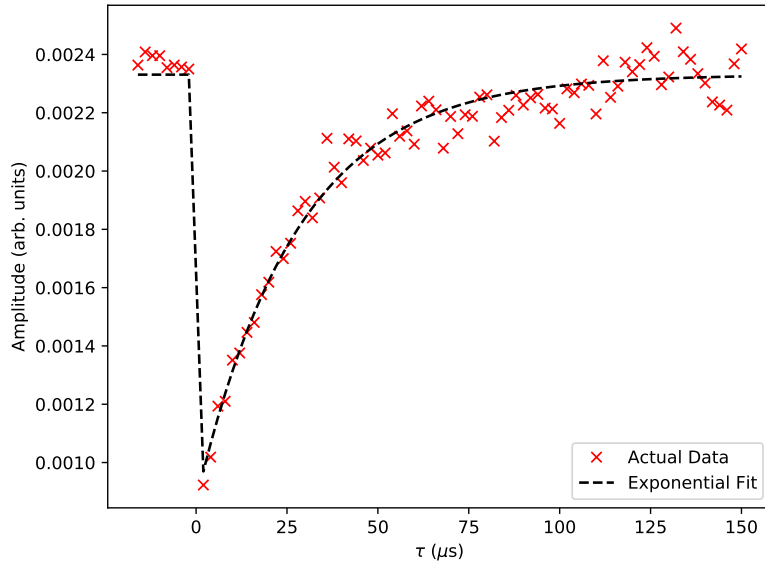


Figure 24: Transmission versus τ for T_1 of a transmon qubit. The data before 0μ s is the transmission at the $|0\rangle$ state for reference. As τ increases, the qubit decays from the excited state to the ground state. The data is then fit with an exponential decay to determine a T_1 time of 27.4μ s.

Ramsey spectroscopy can also be used to find the transverse relaxation of the qubit. The relaxation time in this case is commonly referred to as the T_2^* time. We use a small detuning in MHz in order to create a Ramsey oscillation with periods in the μs regime. We fit this decay to the following equation

$$A = B + C \sin(2\pi\Delta\tau + \phi) \exp\left(-\frac{\tau}{T_2^*}\right) \quad (37)$$

where A is the readout signal amplitude, B is the oscillation offset, C is the oscillation amplitude, Δ is the detuning and ϕ is a phase offset.

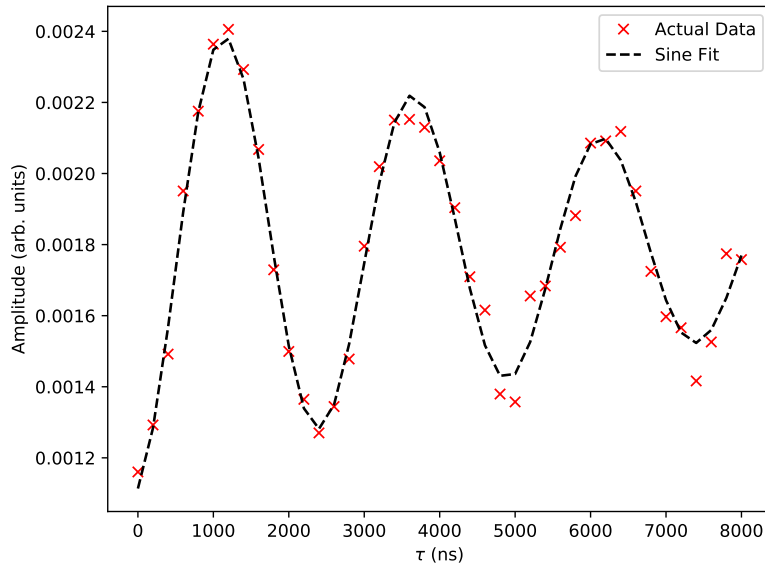


Figure 25: Ramsey spectroscopy with a small detuning of 0.5 MHz and long τ in μs . In this case, the oscillation is fit to a sine wave with an exponentially decaying envelope to obtain a T_2^* time of 7.73 μs .

As seen from Figure 26, the system T_2 value can drift over time [36]. With current superconducting qubits, T_2 is often the limiting factor in the maximizing the gate depth. This may be due to internal factors such as quasiparticle tunnelling [37] or external environmental factors [38]. To reduce this detrimental effects, some remedies include increasing the E_J/E_C ratio in qubit design and fabrication to reduce tunnelling rates, quantum error correction schemes [39] and using RF filters to filter out infrared radiation [40].

For the single junction transmon qubit discussed so far, the T_1 time of 27.4 demonstrated is up to par with conventional qubits reaching 10 to 30 μs while the T_2 time is lacking at 7.73 μs compared to an average of 20 μs [41]. The coherence times of these qubits can vary greatly in fabrication, as seen from the variance

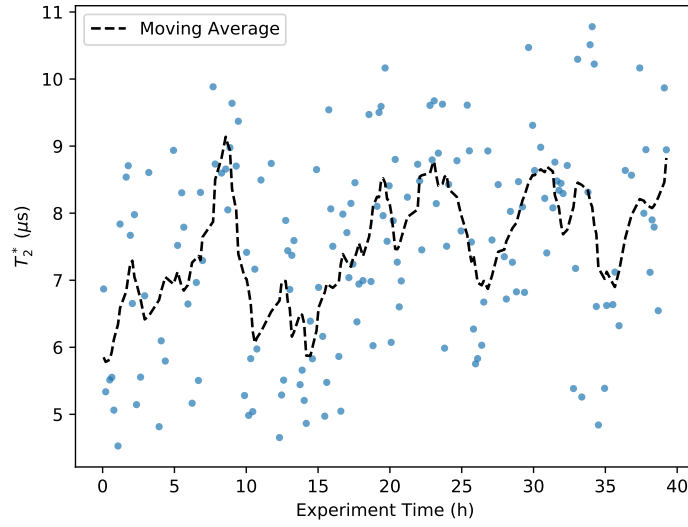


Figure 26: T_2^* time obtained over a period of 40 hours through Ramsey spectroscopy. The mean is $7.49 \mu\text{s}$ and the standard deviation is $1.47 \mu\text{s}$. From the moving average with a 25 sample window, we can observe the variation of T_2^* over time.

in IBM's single junction transmon samples [41] with some T_1 times exceeding $100 \mu\text{s}$. Nonetheless, in relation to the short π -pulse length of 29 ns , we are able to perform single qubit algorithms such as the Gauss Sum Factorization, which will be discussed in the following part.

3.3.3 Gauss Sum Factorization

A quadratic Gauss sum has the form [42],

$$G(a, p) = \sum_{n=0}^{p-1} e^{2\pi i a n^2 / p} \quad (38)$$

where a is an integer and p is a prime number. It is possible to use a qubit to compute this sum, with the phase ϕ in Equation 1. Cold atoms [43] and superconducting qubits [44] have used Gauss sums to factorize numbers, through the truncated reciprocal Gauss sum

$$G(N, l, M) = \frac{1}{M+1} \sum_{m=0}^M e^{2\pi i m^2 \frac{N}{l}}, \quad (39)$$

where N is the number that we want to factorize, l is the trial factor and M is the number of pulses. Instead of summing to p , it is truncated at a smaller number m . The reciprocal sum comes from flipping the fraction a/p . We can see that if l is not a factor of N , the fraction N/l will not be a positive integer.

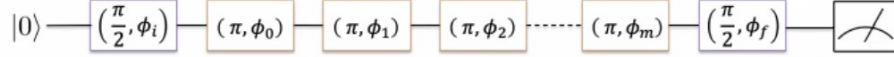


Figure 27: Pulse sequence for a Gauss summand. $\phi_i = \phi_f = -\pi/2$. $\phi_m = (-1)^m \pi(2m-1)N/l$. We send a $\pi/2$ pulse with phase $-\pi/2$ to rotate the qubit to the mixed state. Then, we send a series of π -pulses with phase $(-1)^m \pi(2m-1)N/l$ where m is the pulse number. Finally, we send a $\pi/2$ pulse phased shifted by $-\pi/2$ before measurement.

From the pulse sequence, if l does not divide N , the system will experience some arbitrary rotation during the series of π -pulses. However, if l is indeed a factor of N , the qubit will stay in the superposition state during the sequence and the final $\pi/2$ pulse will rotate the qubit from the superposition state to the excited state.

As mentioned before, we truncate the sum at a smaller number m instead of $p-1$. Figure 28 shows that using more pulses causes the system to be more vulnerable to decoherence. However, using too few pulses causes the effect of ghost factors, non-factors that appear close to actual factors, to be more predominant. Hence, a trade-off is needed in the number of pulses m . Techniques to treat the effect of ghost factors are discussed in [44]. Finally, the full algorithm for integer factorization is as follows

Algorithm 13 GAUSSSUMMAND($N, l, m, p_\pi, p_{\pi/2}, p_{readout}, S$)

```

Initialize pulse sequence  $P$ 
Initialize  $p_1 = p_2 = p_{\pi/2}$ 
Set  $p_1$  to play at the start with phase  $-\pi/2$ 
Set pulse  $p_\pi$  to play after  $p_1$ 
for  $i$  in 0 to  $m - 1$  do
    Initialize  $p = p_\pi$  with phase  $(-1)^i(2i - 1)\frac{N}{l}\pi$ 
    Set  $p$  to play after the previous pulse
    Add  $p$  to  $P$ 
end for
Set  $p_2$  to play after the previous pulse with phase  $-\pi/2$ 
Add  $p_1, p_2, p_\pi, p_{readout}$  to  $P$ 
 $M = \text{RUNEXPERIMENT}(P, S)$ 
return  $M$ 

```

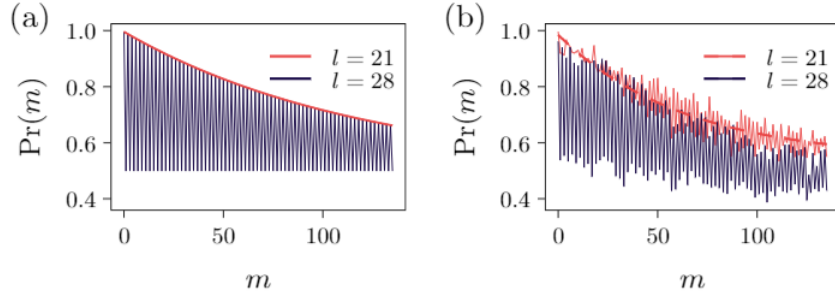


Figure 28: Excited state probability versus the number of pulses m for $N = 263193$ and trial factors $l = 21$ (red) and $l = 28$ (blue)². 21 is an actual factor while 28 is a non-factor. (a) represents a simulation and (b) represents the experiment results carried out on a transmon qubit. The actual factor stays at the excited state while the non-factor moves randomly. As the number of pulses increases, so too does the experiment time and hence the impact of decoherence on the measurement results. $l = 28$ shows the behaviour of a ghost factor [44].

Algorithm 14 GAUSSSUMFACTORIZATION($N, l, M, p_\pi, p_{\pi/2}, p_{readout}, S$)

```

Prepare measurement counter  $M_G$ 
for  $m$  in 0 to  $M$  do
     $R = \text{GAUSSSUMMAND}(N, l, m, p_\pi, p_{\pi/2}, p_{readout}, S)$ 
    Append  $M_G$  with  $R$ 
end for
Get the excited state probability  $p$  from  $M_G$ 
return  $p$ 

```

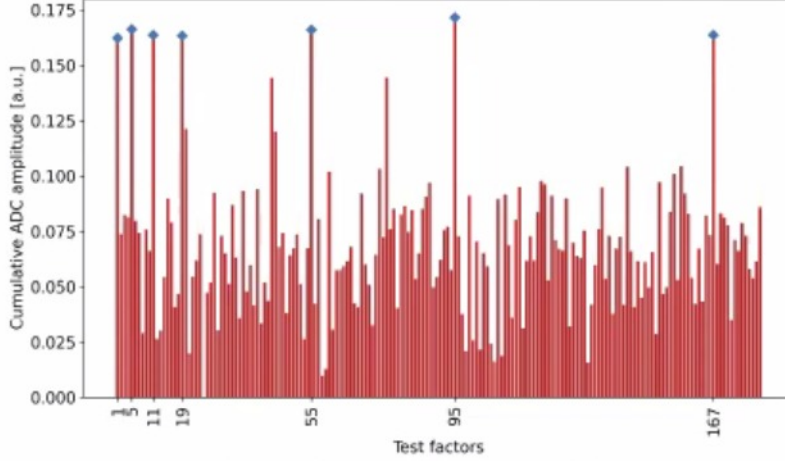


Figure 29: Excited state probability versus trial factors for the Gauss sum factorization of $N = 47293565$, with $M = 15$. The actual factors are $l = 1, 5, 11, 19, 167$ with composites 55, 95 have higher probability than the other non-factors.

While this algorithm currently does not have practical applications due to needing to determine the phase N/l , it serves as a good test of single qubit control and measurement over the number of pulses needed M and the discernability σ between factors and non-factors.

3.3.4 Bell Inequality

In 1935, Einstein, Podolsky and Rosen published the EPR paradox which suggests that the results from measuring a quantum state are pre-determined by local hidden variables [45]. 29 years later, Bell's theorem showed that for systems with multiple axes of measurement, it is possible to violate the Bell's inequality and show that local hidden variable theory is incompatible with quantum mechanics [46].

The Clauser-Horne-Shimony-Holt (CHSH) inequality is a generalization of the Bell inequality [47]. Given two qubits A and B , the entanglement correlation E is given by

$$\begin{aligned} E(x, y) &= P_{\text{same}}(x, y) - P_{\text{diff}}(x, y) \\ &= P_{00}(x, y) - P_{01}(x, y) - P_{10}(x, y) + P_{11}(x, y), \end{aligned} \quad (40)$$

where x is an axis of which qubit A is measured along and y is likewise for qubit B. The Bell signal is defined as

$$S = E(a, b) - E(a', b) + E(a, b') + E(a', b'), \quad (41)$$

²Figure taken from [44]

where $a, a' \in x$ and $b, b' \in y$. Local variable theory predicts a limit of $|S|$ is 2 while the quantum limit is $2\sqrt{2}$. Hence, if a value of $|S| > 2$ is obtained, for the set $\{a, a', b, b'\}$, this system violates the inequality.

The CHSH inequality is used to determine how entangled a system is based on how close the Bell signal S is to the Tsirelson's bound $2\sqrt{2}$. [48].

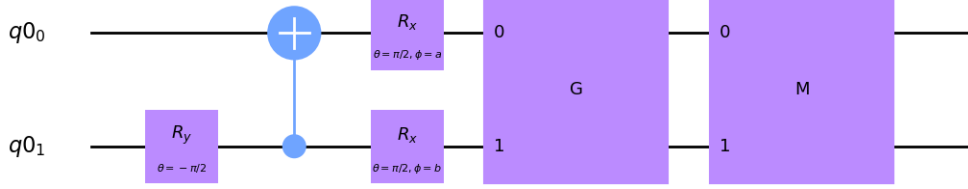
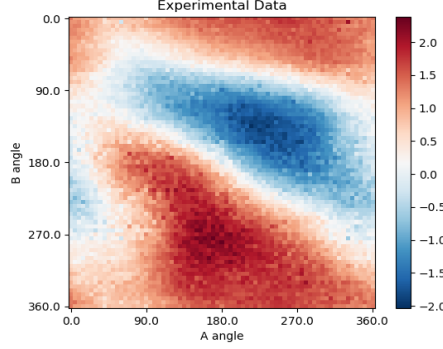


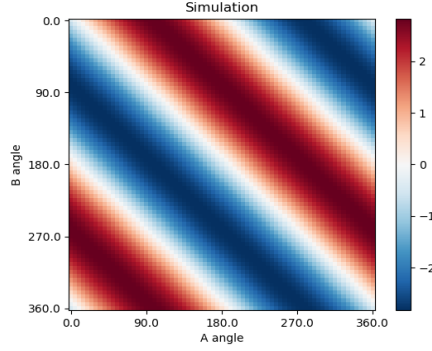
Figure 30: Quantum circuit for violating the Bell inequality. The superposition state $|00\rangle - |01\rangle$ is generated by applying a $-Y/2$ rotation on the second qubit. Next, the entangled state $|10\rangle - |01\rangle$ is obtained by applying the SWIPHT gate as a CNOT gate with the second qubit as the control and the first as the target. The first qubit is rotated by $\theta = \pi/2, \phi = a$ and the second by $\theta = \pi/2, \phi = b$ to change the measurement axes. A prerotation from Figure 21 is then applied to the system before measurement. This sequence is repeated for each of the sixteen measurement operators.

We apply the circuit in Figure 30 to a system of two flux tunable transmon qubits with frequency 3.0636 GHz and 3.2840 GHz respectively capacitively coupled to a 3D microwave cavity with resonant frequency 4.51721 GHz. The SWIPHT gate is used at the entangling CNOT gate between the two qubits to create a Bell state with fidelity 74.12%.

We sweep a and b from 0° to 360° with steps of 5° using 10000 shots per measurement and select $a' = a + 90$ and $b' = b + 90$. From state tomography, we can obtain the values of $P_{00} - P_{01} - P_{10} + P_{11}$ for each set of a, b to determine E .



(a) Experiment Results



(b) Simulation

Figure 31: (a) Bell signal S from the experimental data for angles a, b from 0° to 360° with an angle difference of 90° . The red area represents the maxima of S while the blue area represents the minima. (b) Simulation of the experiment. The results agree with the simulation in the alternating pattern of maxima and minima and there are two regions where $|S| > 2$, which violates the Bell inequality. However, there is an offset in b and difference in the gradient from the actual results compared to the simulation. The angle offset is due to a phase shift of about 240° in the control qubit by the SWIPHT gate, which has also been reported by Premaratne [31]. The gradient shift could be due to the permanent capacitive coupling and low energy separation between the two qubits.

The experiment results suggest a S maximum at $a = 170^\circ$, $b = 275^\circ$ of 2.388 and minimum at $a = 215^\circ$, $b = 135^\circ$ of -2.035 in Figure 31. Other superconducting qubit setups have reported Bell signal values of 2.629 ± 0.028 and 2.07326 ± 0.0003 [49, 50]. Hence, we have demonstrated the capability of the QPU to control two qubits at the same time.

To conclude, we have shown how the RF components function in the QPU to drive pulses for qubit control and readout and the analysis of the readout pulses for qubit state determination. We have also shown how a qubit is characterized by determining the readout frequency of the readout resonator, the frequency of the qubit, the pulse length and amplitude needed for qubit control and the decoherence times. Finally, we have demonstrated the use of the QPU in carrying out a single qubit experiment in Gauss Sum Factorization and a two qubit experiment in violating the Bell Inequality.

4 Software Stack

In this section, I describe my implementation of the interface between a quantum computing user and the quantum computer and the circuit to pulse transpilation processes.

At the top layer controlling the QPU to execute instructions on the qubits is the software stack. With this stack, a conceptual quantum circuit is compiled into a sequence of microwave pulses to be executed on a qubit device. This stack also streamlines the running of quantum algorithms.

4.1 Quantum Software Development Kit (QSDK)

A QSDK is a software suite for a quantum programmer to design the control sequence for a quantum computer. Commonly used QSDKs are Qiskit [2] and Cirq [3]. These allow users to design quantum circuits where qubits are controlled with quantum logic gates. Formally, we can describe a quantum circuit as an ordered pair $QC = (Q, G)$ where Q is the set of qubits and G is the list of gates. Typically, quantum circuits are defined with the number of qubits at initialization and have the following operations.

- **ADD**(g): Adds a quantum logic gate g to G .
- **SET-PARAMETER**(p): Sets the parameter of parameterized gates $g_{p_1}, g_{p_2}, \dots, g_{p_k}$ in G based on the tuple $p = (p_1, p_2, \dots, p_k)$ for some $k \in \mathbb{N}$.
- **MEASURE**(Q_M): Set the measure gate M at the end of G acting on Q_M qubits where $Q_M \subseteq Q$.

For the ADD operation, the input qubits for g are defined in the initialization of g in some QSDKs. In others, the input qubits are also arguments for this operation.

Algorithm 15 SET-PARAMETER(p)

```

Let  $k = 1$ 
for  $g$  in  $G$  do
  if  $g$  is a parametric gate then
    Set the parameter of  $g$  as  $p[k]$ 
     $k = k + 1$ 
  end if
end for

```

Parametric quantum logic gates such as rotation gates like RX, RY and RZ, the rotation operators about the Bloch sphere, have adjustable parameters, such as the the axis rotation angle. They are used in variational quantum computing [51] where the system is prepared in some ansatz and classical optimization techniques are used to find the optimum parameters for the ansatz. This approach is used in various areas such as quantum chemistry [52] in simulating the

Hamiltonian of various molecules. SET-PARAMETER is used by the optimizer to adjust the ansatz.

Another way this is implemented is that in the ADD operation, if g to be added is a parametric gate, it is also added to a subset G_p of parametric gates. SET-PARAMETER then iterates through G_p instead of G ,

Finally, MEASURE sets the measure gate at the end of the gate sequence with information on which qubits are to be measured. Since this is the final step of the quantum computation, certain simulation-based QSDKs omit this operation.

4.2 Circuit Compilation and Execution

To run a quantum circuit on an actual quantum computer, we need to be able to translate it into instructions that can be executed on hardware. From Section 3.2, we have calibration methods to obtain a set of pulses for controlling each qubit in Q . Hence, a basic idea of compiling the gates into pulses is as follows.

Algorithm 16 COMPILECIRCUITTOPULSES(Q, G)

```

Let  $P = \emptyset$ 
for  $g$  in  $G$  do
    Find input qubits  $Q_g$  for gate  $g$ 
    Lookup the pulses  $p$  that represent gate  $g$  from  $Q_g$ 
    Set pulses  $p$  to play at the earliest available opportunity
    Merge  $p$  into  $P$ 
end for
return  $P$ 

```

Algorithm 17 EXECUTECIRCUITONHARDWARE(Q, G, N)

```

Initialize qubit state measurement counter  $M_Q$ 
 $P = \text{COMPILECIRCUITTOPULSES}(Q, G)$ 
 $M = \text{RUNEXPERIMENT}(P, N)$ 
Parse  $M$  into qubit states and record the statistics in  $M_Q$ 
return  $M_Q$ 

```

In COMPILECIRCUITTOPULSES, we have a method to generate a sequence of pulses P from gates G and qubits Q and in EXECUTECIRCUITONHARDWARE, we have a method to fire P over N shots and return the frequency of the result states to the user.

4.2.1 Pulse Timing Considerations

In compiling the circuit to pulses, we have a step *Set pulses p to play at the earliest available opportunity*, which is ambiguous. If we play the gate pulses one after another, the depth of the quantum circuit becomes the total number

of gates, which is the worst case scenario. As each device has limited number of operations in qubit lifetime, we want to arrange the pulses in a way that reduces the circuit depth.

Algorithm 18 COMPILECIRCUITTOPULSESWITHTIMING(Q, G)

```

Let  $P = \emptyset$ 
Let  $T$  be a  $|Q|$ -sized array of all 0s.
for  $g$  in  $G$  do
    Find input qubits  $Q_g$  for gate  $g$ 
    Lookup the pulses  $p$  that represent gate  $g$  from  $Q_g$ 
    if  $g$  is a two-qubit gate then
         $s = \text{MAX}(T[q_{g_1}], T[q_{g_2}])$ 
         $T[q_{g_1}] = s + \text{DURATION}(p)$ 
         $T[q_{g_2}] = T[q_{g_1}]$ 
    else
         $s = T[q_g]$ 
         $T[q_g] = s + \text{DURATION}(p)$ 
    end if
    Set pulses  $p$  to play at starting time  $s$ 
    Merge  $p$  into  $P$ 
end for
return  $P$ 

```

In this modified algorithm, we initialize a counter array T for each qubit to track its availability in time. For each gate $g \in G$, if g is a single-qubit gate, we start the pulse sequence p at $T[q_g]$, where q_g is the target qubit of the gate. Then, we advance $T[q_g]$ by the duration of p . If g is a two-qubit gate, we start p based on which qubit has a larger value in T (the qubit that has experienced longer operations thus far). Afterwards, we set T for both qubits to be the ending of P .

Essentially, we run single qubit operations in parallel until two-qubit gates are required. If we have a two-qubit gate, we start the gate pulse after the latest pulse that involves either qubit and continue. With this, the circuit depth may be reduced from the worst case scenario. This approach works well when the qubits in the system are well isolated and crosstalk is minimal. However, the dynamics of the system may require additional steps or correction pulses in addressing gate errors.

4.3 Cloud-based Quantum Computing

Quantum computers are expensive to set up and so the primary approach to make them accessible to researchers and other users in the industry is to host them for cloud access. This approach is used in supercomputer centers to rent and distribute computing power. From the earlier section, we can see that to run a quantum circuit, we only need to specify the number of qubits to use Q , a

list of quantum logic gates G and the number of shots N . Hence, a cloud-based platform needs to be designed to accept the same set of parameters.

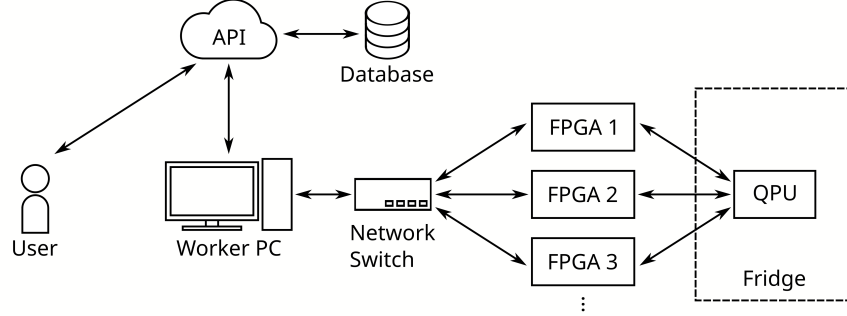


Figure 32: A schematic of cloud based execution of quantum circuits³. The user submits a quantum circuit to the Application Protocol Interface (API) hosted on a cloud server with a desired number of shots. The server will place this job in a database and return the user a job number for tracking. At the QPU, a program checks the database for new jobs and runs them with EXECUTE CIRCUIT ON HARDWARE. After successful execution, the measurement statistics is uploaded to the database for the user to retrieve.

After the user constructs the quantum circuit, it has to be exported to an external service for execution. A common representation for quantum circuits is the Open Quantum Assembly Language (OpenQASM) [53]. QSDKs usually support a conversion of quantum circuits to the OpenQASM format.

```

1 from qiskit import assemble, QuantumCircuit
2
3 circuit = QuantumCircuit()
4 payload = assemble(circuit, shots=1024)

```

Listing 1: Qiskit code to assemble a circuit for execution with 1024 shots

The job payload is then sent to the cloud server. Qiskit natively supports this function via their **Backend** class. On the server, an application needs to handle job requests from users. For development-level deployment, frameworks like Flask [55] and Django [56] can be used to host the API. Given the nature of the job queue, an in-memory database like Redis [57] is recommended for job and result storage and retrieval.

This approach can also be extended to be used in qubit calibration or testing arbitrary waveform sequences for qubit control [54, 58, 59].

³Figure taken from [54]

4.4 Circuit Pre-compilation

In the cloud-based execution of quantum circuits, the user is able to specify an arbitrary amount of qubits, gates to use and number of shots. With the former and latter, restrictions have to be put in place to avoid failure and lengthy operations. The number of gates is also a concern for the room temperature electronics. The duration limit of the sequence of gate pulses is determined by storage capacity of the signal generator and must be regulated.

Another concern is that while the QPU supports a number of logic gates, the user may request an arbitrary gate not defined in the calibration. Hence, we need a mapping from arbitrary logic gates to control pulses.

Hence, before the execution of a quantum circuit, certain modifications have to be made to the circuit for it to be executed on the hardware. In Cirq, the user is able to inspect these modifications or adjust them but in Qiskit, these adjustments are performed by the QPU and hidden from the user.

4.4.1 Qubit Connectivity

The user may request a two-qubit gate between qubits that have no direct connection with each other. To address this, we introduce SWAP gates to bring the state of one of the gate qubits to a connecting qubit and then perform the two-qubit gate between the connecting qubit and the other qubit. Next, we use SWAP gates to bring the state back.

If a gate g is a single qubit gate or is a two-qubit gate between connected qubits, we do not need to make any adjustments. Otherwise, we have to find a path between the two qubits and introduce SWAP gates. We want a path that involves as few SWAP gates as possible and hence less qubits and higher fidelity of SWAP gates. An algorithm to find such a path is described in the appendix.

Algorithm 19 PRECOMPILESWAP(Q, G)

Initialize G' as an empty list

```
for gate  $g$  in  $G$  do
  if  $g$  is not a two-qubit gate then
    Add  $g$  to  $G'$ 
  else
    Let  $q_{g_1}$  and  $q_{g_2}$  be the input qubits of gate  $g$ 
    if  $q_{g_1}$  and  $q_{g_2}$  are connecting qubits then
      Add  $g$  to  $G'$ 
    else
      Select a path  $p$  between  $q_{g_1}$  and  $q_{g_2}$  with maximum SWAP gate
      fidelities.
      for  $i$  in 1 to  $|p| - 2$  do
        Add a SWAP gate between qubits  $p[i]$  and  $p[i + 1]$  to  $G'$ 
      end for
      Add a gate  $g'$  that is identical to  $g$  but with input qubits  $p[|p| - 1]$ 
      and  $q_{g_2}$  to  $G'$ 
      for  $i$  in to  $|p| - 1$  to 2 do
        Add a SWAP gate between qubits  $p[i]$  and  $p[i - 1]$  to  $G'$ 
      end for
    end if
  end if
end for
return  $G'$ 
```

4.4.2 Gate Decomposition

To execute arbitrary gates, we require a decomposition to known gates. Each gate has a unitary matrix representation and the product of unitary matrices is also unitary.

Algorithm 20 DECOMPOSESINGLEQUBITGATESTOUNITARY(G_s)

```
Initialize identity matrix  $R$ 
for gate  $g$  in  $G_s$  do
  Fetch the unitary matrix representation  $U$  of  $g$ 
   $R = U \times R$ 
end for
return  $R$ 
```

Now that we have a unitary matrix representation R of a set of single qubit gates to be applied, we decompose this into single qubit rotations U_3 . This has the form

$$U_3(\theta, \phi, \lambda) = \begin{bmatrix} \cos \frac{\theta}{2} & -e^{i\lambda} \sin \frac{\theta}{2} \\ e^{i\phi} \sin \frac{\theta}{2} & e^{i(\phi+\lambda)} \cos \frac{\theta}{2} \end{bmatrix} \quad (42)$$

where θ, ϕ, λ are Euler angles. From Section 3.2, we have calibrated rotations about the X and Y axes of the qubit and so we use these representations here [2].

$$U_3(\theta, \phi, \lambda) = e^{i\gamma} R_X(\phi) \cdot R_Y(\theta) \cdot R_X(\lambda) \quad (43)$$

where γ is a phase parameter, R_X, R_Y are the rotation operators on the X and Y axes respectively on the Bloch sphere in Figure 1. Decompositions to other rotation bases are also possible. Then, we solve the linear equations

$$\begin{aligned} R_{00} &= \cos \frac{\theta}{2} \\ R_{01} &= -e^{i\lambda} \sin \frac{\theta}{2} \\ R_{10} &= e^{i\phi} \sin \frac{\theta}{2} \\ R_{11} &= e^{i(\phi+\lambda)} \cos \frac{\theta}{2} \end{aligned} \quad (44)$$

for θ, ϕ, λ and to determine γ . In doing so, we reduce a single qubit gate sequence of arbitrary length to 3 gates, which improves the gate depth.

4.5 Variational Quantum Computing

A popular use of current generation quantum computers is variational quantum algorithms. Here, we describe an example of running a simple example through the software stack, namely the Bell test in Section 3.2.8 using the same system using a different circuit. In this example, I have implemented an early prototype of the circuit to pulse transpilation procedure on the Qibo QSDK [4] and ran the algorithm on the NTU-CQT QPU.

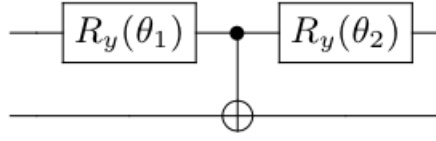


Figure 33: Quantum circuit to violate the Bell inequality [4]. We adjust θ_1 and θ_2 to vary the entanglement and measurement axis respectively. Then, we measure the system in the ZZ , XZ , ZX and XX basis to compute the Bell signal S .

Algorithm 21 CHSHOBJECTIVEFUNC(p)

```

Set parameters  $p$  onto the Bell circuit
Measure the system in the  $ZZ$  basis
Measure the system in the  $XZ$  basis
Measure the system in the  $ZX$  basis
Measure the system in the  $XX$  basis
Compute  $S$  from the measurements
return  $2\sqrt{2} - |S|$ 

```

We copy four circuits from the circuit in Figure 33 and add Hadamard gates to rotate the measurement basis of each qubit accordingly. Then, we use SETPARAMETER to adjust the rotation angles of θ_1 and θ_2 . Next, we execute each circuit. This operation causes the gate to pulse compilation to be executed. Finally, the pulses are played on the signal generator and readout signals recorded. The readout signals are parsed into qubit states and returned to the user. As a value as close as possible to the Tsirelson bound is desired, a classical optimizer is employed to find the parameters p that minimizes the objective function.

The full angle scan took 14 hours while the classical optimizer can take between 1.4 to 4 hours per execution. The three valid values in Table 3 are better than or comparable to the full scan value of 2.388. This is due to the angle step size of the full scan being relatively large and caused the critical positions to be skipped in order to save on measurement time. Hence, while a

Run	CHSH Value	Time (h)
1	-2.74	-
2	-2.44	-
3	-1.81	2.19
4	-1.99	1.41
5	-1.58	3.91
6	-1.75	3.73
7	-2.38	3.56

Table 3: Table of CHSH values from a classical optimizer using CHSHOBJECTIVEFUNC for different runs on the setup in Section 3.2.8 and the time taken for each run. The first two runs did not have their time recorded. Runs 1, 2 and 7 surpass the classical limit of 2. However, the other runs fall short. As the initial parameters are randomly chosen, they may have caused the optimizer to stagnate in the sink regions of Figure 22 below 2.

classical optimizer may be able to achieve better values in a shorter period of time, it can be subject to bad initial parameters and give undesirable results as seen in the bad values.

In this objective function, the circuit has to be executed four times. On a typical cloud based platform, circuit jobs are executed in a queue, which means that the circuits here may not be executed back to back or in a short enough time frame. This leads to concerns about qubit parameters drifting with time [60]. Hence, there is a need to repeat qubit calibration periodically.

Overall, this example highlights how the circuit abstraction layer interacts with the pulse compiler to obtain instructions for the QPU. We convert a conceptual quantum circuit into electromagnetic microwave pulses that manipulate the states of physical superconducting qubits. Then, we are able to readout these qubits and parse the electronic signals into qubit states to get a quantum computation result. Thus, we have a workflow of a quantum computation from user to hardware.

5 Conclusion

With superconducting qubits, it is possible to realize a two level quantum system for quantum computation that fulfills DiVincenzo’s criteria. These qubits are controlled and measured with the QPU, comprising of signal generators and signal digitalizers. We have documented various qubit characterizations algorithms using the QPU such as resonator spectroscopy, qubit spectroscopy, Rabi spectroscopy and Ramsey spectroscopy. These algorithms are then used in concert to fully characterize a superconducting qubit device for basic qubit operations. We have also shown methods to induce two-qubit interactions, through flux biasing of tunable qubits and microwave control in the SWIPHT gate. Furthermore, we explained how the QPU is used to carry out qubit experiments, demonstrating single qubit control through integer factorization with Gauss sums and two qubit control and entanglement with the Bell inequality.

Lastly, we linked the QPU with quantum computation, by detailing the process of which arbitrary quantum gates on a quantum circuit are compiled into microwave pulses for superconducting qubit control. Compiler optimization such as introducing SWAP gates and gate decomposition were also discussed. With this, a quantum circuit provided can be executed on an actual superconducting qubit device and the resulting qubit states measured and returned to the user. We further extend this to a cloud based service. Finally, we demonstrated an example of executing a variational quantum algorithm using the circuit compilation process and a classical optimizer.

We have used the architecture in this paper to control a two qubit device and are looking to further develop it for a ten qubit device. As we increase the number of qubits, we also have to expand the circuit compilation capabilities in Chapter 4 to accomodate n -qubit computation. While the single qubit gate decomposition is sufficient in the current setup, we will need n -qubit gate decomposition to single qubit and two qubit gates. In Chapter 3, we have described qubit control using the pulse architecture which is sufficient for XY-axis control lines, but may require a different type of logic for Z-axis control lines in flux control.

In conclusion, we have established a platform for implementing quantum computing on superconducting qubits. While further refinement has to be done for computations with more qubits, the platform has demonstrated the essential concepts for executing quantum computation and serves as the basis for further iteration and development.

References

- [1] P. W. Shor. “Algorithms for quantum computation: discrete logarithms and factoring”. In: *Proceedings 35th Annual Symposium on Foundations of Computer Science* (1994), pp. 124–134.

- [2] G. Aleksandrowicz et al. *Qiskit: An Open-source Framework for Quantum Computing*. Version 0.7.2. Jan. 2019. DOI: 10.5281/zenodo.2562111. URL: <https://doi.org/10.5281/zenodo.2562111>.
- [3] Cirq Developers. *Cirq*. Version v0.12.0. See full list of authors on Github: <https://github.com/quantumlib/Cirq/graphs/contributors>. Aug. 2021. DOI: 10.5281/zenodo.5182845. URL: <https://doi.org/10.5281/zenodo.5182845>.
- [4] S. Efthymiou et al. “Qibo: a framework for quantum simulation with hardware acceleration”. In: *Quantum Sci. Technol.* 7.1 (2021), p. 015018. ISSN: 2058-9565. DOI: 10.1088/2058-9565/ac39f5. URL: <http://dx.doi.org/10.1088/2058-9565/ac39f5>.
- [5] A. Barenco et al. “Elementary gates for quantum computation”. In: *Phys. Rev. A: At. Mol. Opt. Phys.* 52.5 (1995), 3457–3467. ISSN: 1094-1622. DOI: 10.1103/physreva.52.3457. URL: <http://dx.doi.org/10.1103/PhysRevA.52.3457>.
- [6] F. Vatan and C. Williams. “Optimal quantum circuits for general two-qubit gates”. In: *Phys. Rev. A: At. Mol. Opt. Phys.* 69.3 (2004). ISSN: 1094-1622. DOI: 10.1103/physreva.69.032315. URL: <http://dx.doi.org/10.1103/PhysRevA.69.032315>.
- [7] Nielsen MA and Chuang IL. *Quantum Computation and Quantum Information*. Cambridge University Press, 2010. DOI: 10.1017/CB09780511976667.
- [8] D. Gottesman. “The Heisenberg Representation of Quantum Computers”. In: 1998.
- [9] L. K. Grover. “A Fast Quantum Mechanical Algorithm for Database Search”. In: *Proceedings of the Twenty-Eighth Annual ACM Symposium on Theory of Computing*. STOC ’96. Philadelphia, Pennsylvania, USA: Association for Computing Machinery, 1996, 212–219. ISBN: 0897917855. DOI: 10.1145/237814.237866. URL: <https://doi.org/10.1145/237814.237866>.
- [10] T. Toffoli. “Reversible Computing”. In: *Automata, Languages and Programming, 7th Colloquium, Noordwijkerhout, The Netherlands, July 14-18, 1980, Proceedings*. Ed. by J. W. de Bakker and Jan van Leeuwen. Vol. 85. Lecture Notes in Computer Science. Springer, 1980, pp. 632–644. DOI: 10.1007/3-540-10003-2_104. URL: https://doi.org/10.1007/3-540-10003-2_104.
- [11] S. A. Cook. “The Complexity of Theorem-Proving Procedures”. In: *Proceedings of the Third Annual ACM Symposium on Theory of Computing*. STOC ’71. Shaker Heights, Ohio, USA: Association for Computing Machinery, 1971, 151–158. ISBN: 9781450374644. DOI: 10.1145/800157.805047. URL: <https://doi.org/10.1145/800157.805047>.

- [12] R. M. Karp. “Reducibility among Combinatorial Problems”. In: *Complexity of Computer Computations: Proceedings of a symposium on the Complexity of Computer Computations, held March 20–22, 1972, at the IBM Thomas J. Watson Research Center, Yorktown Heights, New York, and sponsored by the Office of Naval Research, Mathematics Program, IBM World Trade Corporation, and the IBM Research Mathematical Sciences Department*. Ed. by Raymond E. Miller, James W. Thatcher, and Jean D. Bohlinger. Boston, MA: Springer US, 1972, pp. 85–103. ISBN: 978-1-4684-2001-2. DOI: 10.1007/978-1-4684-2001-2_9. URL: https://doi.org/10.1007/978-1-4684-2001-2_9.
- [13] C. Zalka. “Grover’s quantum searching algorithm is optimal”. In: *Phys. Rev. A: At. Mol. Opt. Phys.* 60.4 (1999), 2746–2751. ISSN: 1094-1622. DOI: 10.1103/physreva.60.2746. URL: <http://dx.doi.org/10.1103/PhysRevA.60.2746>.
- [14] D. P. DiVincenzo. “The Physical Implementation of Quantum Computation”. In: *Fortschr. Phys.* 48.9-11 (2000), 771–783. ISSN: 1521-3978. DOI: 10.1002/1521-3978(200009)48:9/11<771::aid-prop771>3.0.co;2-e. URL: [http://dx.doi.org/10.1002/1521-3978\(200009\)48:9/11<771::AID-PROP771>3.0.CO;2-E](http://dx.doi.org/10.1002/1521-3978(200009)48:9/11<771::AID-PROP771>3.0.CO;2-E).
- [15] F. Arute et al. “Quantum Supremacy using a Programmable Superconducting Processor”. In: *Nat* 574 (2019), 505–510. URL: <https://www.nature.com/articles/s41586-019-1666-5>.
- [16] E. J. Zhang et al. *High-fidelity superconducting quantum processors via laser-annealing of transmon qubits*. 2020. arXiv: 2012.08475 [quant-ph].
- [17] W. Yulin et al. “Strong Quantum Computational Advantage Using a Superconducting Quantum Processor”. In: *Phys. Rev. Lett.* 127.18 (2021). ISSN: 1079-7114. DOI: 10.1103/physrevlett.127.180501. URL: <http://dx.doi.org/10.1103/PhysRevLett.127.180501>.
- [18] P. Krantz et al. “A quantum engineer’s guide to superconducting qubits”. In: *Appl. Phys. Rev.* 6.2 (2019), p. 021318. ISSN: 1931-9401. DOI: 10.1063/1.5089550. URL: <http://dx.doi.org/10.1063/1.5089550>.
- [19] B. D. Josephson. “The discovery of tunnelling supercurrents”. In: *Rev. Mod. Phys.* 46 (2 1974), pp. 251–254. DOI: 10.1103/RevModPhys.46.251. URL: <https://link.aps.org/doi/10.1103/RevModPhys.46.251>.
- [20] M. F. Johannes. “Quantum Nonlinearities in Strong Coupling Circuit Qed”. In: 2011.
- [21] Zijun C. “Metrology of Quantum Control and Measurement in Superconducting Qubits”. In: 2018.
- [22] M. Reed. *Entanglement and Quantum Error Correction with Superconducting Qubits*. 2013. DOI: 10.48550/ARXIV.1311.6759. URL: <https://arxiv.org/abs/1311.6759>.

- [23] Zhou Y. et al. “Rapid and unconditional parametric reset protocol for tunable superconducting qubits”. In: *Nature Communications* 12.1 (2021). DOI: 10.1038/s41467-021-26205-y. URL: <https://doi.org/10.1038/s41467-021-26205-y>.
- [24] D. McClure et al. “Rapid Driven Reset of a Qubit Readout Resonator”. In: *Phys. Rev. Applied* 5 (1 2016), p. 011001. DOI: 10.1103/PhysRevApplied.5.011001. URL: <https://link.aps.org/doi/10.1103/PhysRevApplied.5.011001>.
- [25] S. W. Jolin et al. “Calibration of mixer amplitude and phase imbalance in superconducting circuits”. In: *Rev. Sci. Instrum.* 91.12 (2020), p. 124707. ISSN: 1089-7623. DOI: 10.1063/5.0025836. URL: <http://dx.doi.org/10.1063/5.0025836>.
- [26] W. D. Kalfus et al. “High-Fidelity Control of Superconducting Qubits Using Direct Microwave Synthesis in Higher Nyquist Zones”. In: *IEEE Trans. Quantum Eng.* 1 (2020), 1–12. ISSN: 2689-1808. DOI: 10.1109/tqe.2020.3042895. URL: <http://dx.doi.org/10.1109/TQE.2020.3042895>.
- [27] Y. Xu et al. *Automatic Qubit Characterization and Gate Optimization with QubiC*. 2021. arXiv: 2104.10866 [quant-ph].
- [28] N. F. Ramsey. “A Molecular Beam Resonance Method with Separated Oscillating Fields”. In: *Phys. Rev.* 78 (6 1950), pp. 695–699. DOI: 10.1103/PhysRev.78.695. URL: <https://link.aps.org/doi/10.1103/PhysRev.78.695>.
- [29] J. J. García-Ripoll, A. Ruiz-Chamorro, and E. Torrontegui. “Quantum Control of Frequency-Tunable Transmon Superconducting Qubits”. In: *Phys. Rev. Appl.* 14.4 (2020). ISSN: 2331-7019. DOI: 10.1103/physrevapplied.14.044035. URL: <http://dx.doi.org/10.1103/PhysRevApplied.14.044035>.
- [30] S. E. Economou and E. Barnes. “Analytical approach to swift nonleaky entangling gates in superconducting qubits”. In: *Phys. Rev. B* 91 (16 2015), p. 161405. DOI: 10.1103/PhysRevB.91.161405. URL: <https://link.aps.org/doi/10.1103/PhysRevB.91.161405>.
- [31] S. P. Premaratne et al. “Implementation of a generalized controlled-NOT gate between fixed-frequency transmons”. In: *Phys. Rev. A* 99 (1 2019), p. 012317. DOI: 10.1103/PhysRevA.99.012317. URL: <https://link.aps.org/doi/10.1103/PhysRevA.99.012317>.
- [32] P. Jurcevic et al. *Demonstration of quantum volume 64 on a superconducting quantum computing system*. 2020. arXiv: 2008.08571 [quant-ph].
- [33] Y. Liu, L. F. Wei, and F. Nori. “Tomographic measurements on superconducting qubit states”. In: *Phys. Rev. B: Condens. Matter* 72.1 (2005). ISSN: 1550-235X. DOI: 10.1103/physrevb.72.014547. URL: <http://dx.doi.org/10.1103/PhysRevB.72.014547>.

- [34] J. B. Altepeter, D. F. V. James, and P.G. Kwiat. “4 Qubit Quantum State Tomography”. In: *Quantum State Estimation*. Ed. by Matteo Paris and Jaroslav Řeháček. Berlin, Heidelberg: Springer Berlin Heidelberg, 2004, pp. 113–145. ISBN: 978-3-540-44481-7. DOI: 10.1007/978-3-540-44481-7_4. URL: https://doi.org/10.1007/978-3-540-44481-7_4.
- [35] J. Shang, Z. Zhang, and H. K. Ng. “Superfast maximum-likelihood reconstruction for quantum tomography”. In: *Phys. Rev. A: At. Mol. Opt. Phys.* 95.6 (2017). ISSN: 2469-9934. DOI: 10.1103/physreva.95.062336. URL: <http://dx.doi.org/10.1103/PhysRevA.95.062336>.
- [36] J. J. Burnett et al. “Decoherence benchmarking of superconducting qubits”. In: *npj Quantum Inf.* 5.1 (2019). ISSN: 2056-6387. DOI: 10.1038/s41534-019-0168-5. URL: <http://dx.doi.org/10.1038/s41534-019-0168-5>.
- [37] G. Catelani et al. “Decoherence of superconducting qubits caused by quasiparticle tunneling”. In: *Physical Review B* 86.18 (2012). DOI: 10.1103/physrevb.86.184514. URL: <https://doi.org/10.1103/physrevb.86.184514>.
- [38] L. Cardani et al. “Reducing the impact of radioactivity on quantum circuits in a deep-underground facility”. In: *Nature Communications* 12.1 (2021). DOI: 10.1038/s41467-021-23032-z. URL: <https://doi.org/10.1038/s41467-021-23032-z>.
- [39] N. Ofek et al. *Demonstrating Quantum Error Correction that Extends the Lifetime of Quantum Information*. 2016. DOI: 10.48550/ARXIV.1602.04768. URL: <https://arxiv.org/abs/1602.04768>.
- [40] S. Krinner et al. “Engineering cryogenic setups for 100-qubit scale superconducting circuit systems”. In: *EPJ Quantum Technology* 6.1 (2019). DOI: 10.1140/epjqt/s40507-019-0072-0. URL: <https://doi.org/10.1140/epjqt/s40507-019-0072-0>.
- [41] H.J. Mamin et al. “Merged-Element Transmons: Design and Qubit Performance”. In: *Physical Review Applied* 16.2 (2021). DOI: 10.1103/physrevapplied.16.024023. URL: <https://doi.org/10.1103/physrevapplied.16.024023>.
- [42] K. F. Ireland and M. I. Rosen. *A classical introduction to modern number theory*. Springer, 2011.
- [43] M. Gilowski et al. “Gauss Sum Factorization with Cold Atoms”. In: *Phys. Rev. Lett.* 100.3 (2008). ISSN: 1079-7114. DOI: 10.1103/physrevlett.100.030201. URL: <http://dx.doi.org/10.1103/PhysRevLett.100.030201>.
- [44] L. H. Zaw et al. “Ghost factors in Gauss-sum factorization with transmon qubits”. In: *Phys. Rev. A* 104 (6 2021), p. 062606. DOI: 10.1103/PhysRevA.104.062606. URL: <https://link.aps.org/doi/10.1103/PhysRevA.104.062606>.

- [45] A. Einstein, B. Podolsky, and N. Rosen. “Can Quantum-Mechanical Description of Physical Reality Be Considered Complete?” In: *Phys. Rev.* 47 (10 1935), pp. 777–780. DOI: 10.1103/PhysRev.47.777. URL: <https://link.aps.org/doi/10.1103/PhysRev.47.777>.
- [46] J. S. Bell. “On the Einstein Podolsky Rosen paradox”. In: *Phys. Phys. Fiz.* 1 (3 1964), pp. 195–200. DOI: 10.1103/PhysicsPhysiqueFizika.1.195. URL: <https://link.aps.org/doi/10.1103/PhysicsPhysiqueFizika.1.195>.
- [47] J. F. Clauser et al. “Proposed Experiment to Test Local Hidden-Variable Theories”. In: *Phys. Rev. Lett.* 23 (15 1969), pp. 880–884. DOI: 10.1103/PhysRevLett.23.880. URL: <https://link.aps.org/doi/10.1103/PhysRevLett.23.880>.
- [48] B. S. Cirel’son. “Quantum generalizations of Bell’s inequality”. In: *Lett. Math. Phys.* 4 (1980), pp. 93–100.
- [49] Y. P. Zhong et al. “Violating Bell’s inequality with remotely connected superconducting qubits”. In: *Nat. Phy.* 15.8 (2019), 741–744. ISSN: 1745-2481. DOI: 10.1038/s41567-019-0507-7. URL: <http://dx.doi.org/10.1038/s41567-019-0507-7>.
- [50] M. Ansmann et al. “Violation of Bell’s inequality in Josephson phase qubits”. In: *Nat.* 461 (Sept. 2009), pp. 504–6. DOI: 10.1038/nature08363.
- [51] M. Cerezo and others. “Variational quantum algorithms”. In: *Nat. Rev. Phys.* 3.9 (2021), 625–644. ISSN: 2522-5820. DOI: 10.1038/s42254-021-00348-9. URL: <http://dx.doi.org/10.1038/s42254-021-00348-9>.
- [52] J. R. McClean et al. “The theory of variational hybrid quantum-classical algorithms”. In: *New Journal of Physics* 18.2 (2016), p. 023023. ISSN: 1367-2630. DOI: 10.1088/1367-2630/18/2/023023. URL: <http://dx.doi.org/10.1088/1367-2630/18/2/023023>.
- [53] A. W. Cross et al. *Open Quantum Assembly Language*. 2017. arXiv: 1707.03429 [quant-ph].
- [54] K. H. Park et al. *ICARUS-Q: A scalable RFSoc-based control system for superconducting quantum computers*. 2021. arXiv: 2112.02933 [quant-ph].
- [55] Pallets. *Flask*. 2021. URL: <https://flask.palletsprojects.com>.
- [56] Django Software Foundation. *Django*. 2021. URL: <https://www.djangoproject.com>.
- [57] Redis Ltd. *Redis*. 2021. URL: <https://redis.io/>.
- [58] P. Gokhale et al. *Optimized Quantum Compilation for Near-Term Algorithms with OpenPulse*. 2020. arXiv: 2004.11205 [quant-ph].
- [59] D. C. McKay et al. *Qiskit Backend Specifications for OpenQASM and OpenPulse Experiments*. 2018. arXiv: 1809.03452 [quant-ph].

- [60] A. R. R. Carvalho et al. “Error-Robust Quantum Logic Optimization Using a Cloud Quantum Computer Interface”. In: *Phys. Rev. Appl.* 15.6 (2021). ISSN: 2331-7019. DOI: 10.1103/physrevapplied.15.064054. URL: <http://dx.doi.org/10.1103/PhysRevApplied.15.064054>.
- [61] E. W. Dijkstra. “A Note on Two Problems in Connexion with Graphs”. In: *Numer. Math.* 1.1 (1959), 269–271. ISSN: 0029-599X. DOI: 10.1007/BF01386390. URL: <https://doi.org/10.1007/BF01386390>.

Appendices

A Path finding algorithm for swapping qubits

We run a modified version of Dijkstra’s algorithm [61] where Q is the set of all qubits, q_s is the source qubit, q_t is the target qubit and g is the intended two qubit interaction gate. First we initialize the set of reached qubits R , the distance function $dist$ and the array of previous qubits P . On each step of the loop, we first select the qubit with the maximum SWAP fidelity thus far as q . We add this qubit to R and then iterate through the neighbouring qubits of q . If a neighbouring qubit u is not in R , we fetch the fidelity of the gate between q and u as f . Then, we calculate the fidelity of the sequence from q_s to q and then from q to u as $d = dist(q) + \log f$. If this new path has greater fidelity than an existing path, we set the new fidelity of q_s to u as d and set the previous qubit of this path as q . We do this for all qubits until we select q_t . In that case, the loop ends and we iterate through P starting from q_t to find the path of maximum fidelity from q_t to q_s .

The worst case running time of this is $O(n \log n + m)$, where n is the number of qubits and m is the number of connections between qubits.

Algorithm 22 PATHFINDING(Q, q_s, q_t, g)

```
Initialize  $\text{dist}(q) = -\infty$  for  $q$  in  $Q$ 
Set  $\text{dist}(q_s) = 0$ 
Initialize  $P$  as an empty array with length  $|Q|$ 
Initialize  $R = \emptyset$ 
while  $R \neq Q$  do
    Select  $q$  in  $Q - R$  with max  $\text{dist}(q)$ 
    Add  $q$  to  $R$ 
    if  $q = q_t$  then
        Exit the while loop
    end if
    for  $u$  in the neighbouring qubits of  $q$  do
        if  $u$  is not in  $R$  then
            Fetch the SWAP gate fidelity between  $u$  and  $q$  as  $f$ 
            if  $u = q_t$  then
                Set  $f$  to be the fidelity of  $g$  for  $q$  and  $u$  instead
            end if
             $d = \text{dist}(q) + \log f$ 
            if  $d > \text{dist}(u)$  then
                 $\text{dist}(u) = d$ 
                 $P[u] = q$ 
            end if
        end if
    end for
end while
Let  $p = P[q_t]$ 
Initialize  $S = [q_t]$ 
while  $p$  is not undefined do
    Add  $p$  to  $S$ 
     $p = P[p]$ 
end while
Reverse  $S$ 
return  $S$ 
```
