UNIVERSITY of CALIFORNIA
Santa Barbara

Benchmarking the Superconducting Josephson Phase Qubit: The Violation of Bell's Inequality

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

Doctor of Philosophy

in

Physics

by

Markus Ansmann

Committee in charge:

Professor John M. Martinis, Chair
Professor Andrew N. Cleland
Professor Andreas Ludwig

June 2009

The dissertation of Markus Ansmann is approved:

Professor Andrew N. Cleland

Professor Andreas Ludwig

Professor John M. Martinis, Chair

June 2009

Benchmarking the

Superconducting Josephson Phase Qubit:

The Violation of Bell's Inequality

by

Markus Ansmann

To my parents,

Klaus and Ulrike Ansmann,

who made it possible for me to study abroad

and who raised me to be persistent and crazy enough

to complete the journey that led to this degree.

Acknowledgements

This work would not have been possible without the intellectual and emotional support of the many amazing people that I had the honor of sharing the past several years of my life with.

First, I would like to thank my fiancée, Doctor Ekaterina Chernobai, for her never-ending patience and support during these crazy years. Ready for more?

I thank Professor John Martinis for the opportunity to be a member of his group and for being my research advisor. Working for John and having the chance to set up a new lab with him has been immensely helpful for me to grow as a scientist, a team-player, and a person. I owe a lot of insights about project management, leadership, and, of course, physics to our interactions.

I am very grateful to Assistant Professor Robert McDermott, Doctor Matthias Steffen, and Doctor Ken Cooper for helping me with my start into the field of experimental condensed matter physics. Their coaching gave me the solid foundation on which I was able to build my thesis work while their great sense of humor helped me survive the beating with a smile.

Without Doctor Haohua Wang this experiment would not have been possible since his incredible skill and dedication in the cleanroom resulted in the qubit devices that were the key to demonstrating the violation of the Bell inequality.

I would like to specifically acknowledge Matthew Neeley, whose enthusiasm and support for LabRAD were instrumental in getting it to the level of quality at which it runs today. Matthew opened my eyes to the world of programming beyond Delphi and helped me significantly improve and diversify my skills.

Erik Lucero played an important role in this work as well by always being ready to fix another blown fuse on the control electronics. His positive attitude and his love for Chai helped me keep my sanity and stay in touch with reality.

I also thank Radek Bialczak, Aaron O'Connell, Doctor Max Hofheinz, Daniel Sank, Jim Wenner, and Doctor Martin Weides. Their dedication to the team and sense of humor helped create a fun, yet extremely productive work environment.

I would like to express my appreciation to the staff of the UCSB Physics Department, the California NanoSystems Institute, the UCSB Machine Shop, and the UCSB Cleanroom. The infrastructure they provided was extremely helpful in efficiently meeting the challenges posed by the different steps of the project.

I also thank Professor Andrew Cleland and Professor Andreas Ludwig for agreeing to be on my thesis committee and for putting up with my last-minute delivery of this work. Please accept my apologies.

Finally, I would like to express my appreciation to my family and friends without whom I would have surely lost my mind over the course of the last years.

Curriculum Vitæ

Markus Ansmann

Education

2009	Doctor of Philosophy, Physics, University of California, Santa Barbara (expected)
2007	Master of Arts, Economics, University of California, Santa Barbara
2007	Certificate, Graduate Program in Management Practice, University of California, Santa Barbara
2003	Bachelor of Science, Physics, University of California, Santa Barbara

Publications

Ansmann, M., Wang, H., Bialczak, R. C., Hofheinz, M., Lucero, E., Neeley, M., O'Connell, A. D., Sank, D., Weides, M., Wenner, J., Cleland, A. N., Martinis, J. M., "Violation of Bell's inequality in Josephson phase qubits", Submitted to Nature (2009)

Martinis, J. M., Ansmann, M., and Aumentado, J., "Energy Decay in Josephson Qubits from Non-equilibrium Quasiparticles", Submitted to Physical Review Letters (2009)

Hofheinz, M., Wang, H., Ansmann, M., Bialczak, R. C., Lucero, E., Neeley, M., O'Connell, A. D., Sank, D., Wenner, J., Martinis, J. M., and Cleland, A. N., "Synthesizing arbitrary quantum states in a superconducting resonator", Nature (2009), 459:546-549

Wang, H., Hofheinz, M., Ansmann, M., Bialczak, R. C., Lucero, E., Neeley, M., O'Connell, A. D., Sank, D., Wenner, J., Cleland, A. N., and Martinis, J. M., "Measurement of the decay of Fock states in a superconducting quantum circuit", Physical Review Letters (2008), 101:240401

Katz, N., Neeley, M., Ansmann, M., Bialczak, R. C., Hofheinz, M., Lucero, E., O'Connell, A. D., Wang, H., Cleland, A. N., Martinis, J. M., and Korotkov, A. N., "Reversal of the Weak Measurement of a Quantum State in a Superconducting Phase Qubit", Physical Review Letters (2008), 101:200401

Levy, A. R., Leonardi, R., Ansmann, M., Bersanelli, M., Childers, J., Cole, T. D., D'Arcangelo, O., Davis, G. V., Lubin, P. M., Marvil, J., Meinhold, P. R., Miller, G., O'Neill, H., Stavola, F., Stebor, N. C., Timbie, P. T., Van der Heide, M., Villa, F., Villela, T., Williams, B. D., Wuensche, C. A., "The White Mountain Polarimeter Telescope and an Upper Limit on Cosmic Microwave Background Polarization", Astrophysical Journal Supplement Series (2008), 177:419-430

Hofheinz, M., Weig, E. M., Ansmann, M., Bialczak, R. C., Lucero, E., Neeley, M., O'Connell, A. D., Wang, H., Martinis J. M., and Cleland, A. N., "Generation of Fock states in a superconducting quantum circuit", Nature (2008), 454:310-314

Neeley, M., Ansmann, M., Bialczak, R. C., Hofheinz, M., Katz, N., Lucero, E., O'Connell, A. D., Wang, H., Cleland, A. N., and Martinis J. M., "Process tomography of quantum memory in a Josephson-phase qubit coupled to a two-level state", Nature Physics (2008), 4:523-526

Lucero, E., Hofheinz, M., Ansmann, M., Bialczak, R. C., Katz, N., Neeley, M., OConnell, A.D., Wang, H., Cleland, A. N., and Martinis, J. M., "High-fidelity gates in a Josephson qubit", Physical Review Letters (2008), 100:247001

Neeley, M., Ansmann, M., Bialczak, R. C., Hofheinz, M., Katz, N., Lucero, E., OConnell, A. D., Wang, H., Cleland, A. N., and Martinis, J. M., "Transformed Dissipation in Superconducting Quantum Circuits", Physical Review B (2008), 77:180508

O'Connell, A. D., Ansmann, M., Bialczak, R. C., Hofheinz, M., Katz, N., Lucero, E., McKenney, C., Neeley, M., Wang, H., Weig, E. M., Cleland, A. N., and Martinis, J. M., "Microwave Dielectric Loss at Single Photon Energies and milliKelvin Temperatures", Applied Physics Letters (2008), 92:112903

Bialczak, R. C., McDermott, R., Ansmann, M., Hofheinz, M., Katz, N., Lucero, E., Neeley, M., O'Connell, A. D., Wang, H., Cleland, A. N., and Martinis, J. M., "1/f Flux Noise in Josephson Phase Qubits", Physical Review Letters (2007), 99:187006

Lisenfeld, J., Lukashenko, A., Ansmann, M., Martinis, J. M., Ustinov, A. V., “Temperature dependence of coherent oscillations in Josephson phase qubits”, Physical Review Letters (2007), 99:170504

Steffen, M., Ansmann, M., Bialczak, R. C., Katz, N., Lucero, E., McDermott, R., Neeley, M., Weig, E. M., Cleland, A. N., and Martinis, J. M., “Measurement of the Entanglement of Two Superconducting Qubits via State Tomography”, Science (2006), 313:1423-1425

Steffen, M., Ansmann, M., McDermott, R., Katz, N., Bialczak, R. C., Lucero, E., Neeley, M., Weig, E. M., Cleland, A. N., and Martinis, J. M., “State tomography of capacitively shunted phase qubits with high fidelity”, Physical Review Letters (2006), 97:050502

Katz, N., Ansmann, M., Bialczak, R. C., Lucero, E., McDermott, R., Neeley, M., Steffen, M., Weig, E. M., Cleland, A. N., Martinis, J. M., and Korotkov, A. N., “Coherent state evolution in a superconducting qubit from partial-collapse measurement”, Science (2006), 312: 1498-1500

Marvil, J., Ansmann, M., Childers, J., Cole, T., Davis, G. V., Hadjiyska, E., Halevi, D., Heimberg, G., Kangas, M., Levy, A., Leonardi, R., Lubin, P., Meinhold, P., O’Neill, H., Parendo, S., Quetin, E., Stebor, N., Villela, T., Williams, B., Wuensche, C. A., and Yamaguchi, K., “An Astronomical Site Survey at the Barcroft Facility of the White Mountain Research Station”, New Astronomy (2006), 11:218-225

Martinis, J. M., Cooper, K. B., McDermott, R., Steffen, M., Ansmann, M., Osborn, K., Cicak, K., Oh, S., Pappas, D. P., Simmonds, R.W., and Yu, C. C., “Decoherence in Josephson Qubits from Dielectric Loss”, Physical Review Letters (2005), 95:210503

Kangas, M. M., Ansmann, M., Copsey, K., Horgan, B., Leonardi, R., Lubin, P., Villela, T., “A 100-GHz high-gain tilted corrugated nonbonded platelet antenna”, IEEE Antennas and Wireless Propagation Letters (2005), 4:304-307

Kangas, M. M., Ansmann, M., Horgan, B., Lemaster, N., Leonardi, R., Levy, A., Lubin, P., Marvil, J., McCreary, P., Villela, T., “A 31 pixel flared 100-GHz high-gain scalar corrugated nonbonded platelet antenna array”, IEEE Antennas and Wireless Propagation Letters (2005), 4:245-248

Honors and Awards

2003	Arnold Nordsieck Award, Physics Department, University of California, Santa Barbara, California
2003	Research Honors, Physics Department, University of California, Santa Barbara, California
2003	Dean’s Honors, University of California, Santa Barbara, California
1999 - 2001	Dean’s List, Stevens Institute of Technology, Hoboken, NJ

Abstract

Benchmarking the

Superconducting Josephson Phase Qubit:

The Violation of Bell's Inequality

by

Markus Ansmann

The concepts of entanglement and superpositions introduced by quantum mechanics promise to allow for the design of a new computing architecture, called a quantum computer, that can exponentially outperform any possible classical computer. Such a performance improvement would make presently intractable computational problems efficiently solvable. Such problems include optimizations, like the traveling salesman problem, factorization, and quantum simulations, e.g. for medical research.

This thesis discusses one approach to implementing a quantum computer that is based on Superconducting Josephson Phase Qubits. An experiment is presented that shows a violation of Bell's inequality using these qubits (quantum bits), i.e. it demonstrates that a pair of these qubits can be placed into a state that shows

a stronger correlation than possible for a classical pair of bits. This experiment meets a major mile-stone for the field of superconducting qubits as it provides strong evidence that the architecture will indeed be able to outperform classical systems.

Furthermore, this experiment is the first demonstration of a violation of Bell's inequality in a solid state system, and the first demonstration in a macroscopic quantum system. It therefore adds valuable supporting evidence that the new ideas proposed by quantum mechanics are indeed valid across different quantum systems and cannot be explained by a deterministic alternative theory.

Contents

List of Figures

List of Tables

Chapter 1

Quantum Computation

1.1 Motivation

1.1.1 The Information Society

During the 20th century, information-intensive activities have grown from a few percent to two-thirds of all US labor [Molitor, 1982]. This was made possible by the advent of the personal computer and the increasingly wide-spread availability of information storage and sharing systems like magnetic hard disks and the Internet. The exponential improvements in performance and the drop in price of information gathering and creating devices has lead to an information deluge that is projected to result in a doubling of the world's information base every 11 hours by 2010 [Coles et al., 2006]. To handle this flood of information efficiently it will need to be met with a continuing dramatic increase in information processing power.

1.1.2 Moore's Law

For the past 40 years, the performance of computing devices has doubled roughly every 18 months. This trend is commonly referred to as "Moore's Law", inspired by an article written in 1965 by Intel co-founder Gordon Moore [Moore, 1965]. Despite the fact that Moore's Law is expected to hold for at least another decade, it is important to prepare for the longer term future when transistors on silicon will have been pushed to their limits.

1.1.3 Church-Turing Thesis

The Church-Turing Thesis makes this even more pressing. Alan Turing in 1938 described a universal logic machine [Turing, 1938] that can efficiently, i.e. in polynomial time, simulate any other computing system. This logic machine is simple enough to allow any classical computer to simulate it efficiently as well. Therefore, any classical computer can simulate any other computing system efficiently. This implies that problems that are intractable, i.e. not solvable in polynomial time, on current computers will remain intractable on all future computers. Examples for such problems include factoring the product of two n-bit primes, the travelling salesman problem, the hidden subgroup problem, etc.

1.1.4 Deutsch-Josza Algorithm

In 1992, David Deutsch and Richard Jozsa proposed a fictitious problem and an algorithm for solving it that indicates that a computer which uses quantum states for computation might not be subject to this limitation imposed by the Church-Turing Thesis [Deutsch and Jozsa, 1992]. The problem consists of determining the nature of an unknown function by queries to an oracle that evaluates the function for a given input. The function acts on an n-bit number and is known to be either constant, i.e. returns 0 (or 1) for all possible inputs, or balanced, i.e. returns 0 for exactly half of all possible inputs and 1 for all others. A classical computer, in the worst case, would have to evaluate the function $2^{n-1} + 1$ times, while a quantum computer would need to evaluate the function only once to determine with certainty whether it is balanced or constant. This is done by evaluating the function once for an input state that consists of a superposition of all possible 2^n states (achievable by the application of a Hadamard gate (see Section 1.3.4) to all input qubits). If the output shows amplitudes for both 0 and 1 the function is balanced, otherwise it is constant.

This algorithm implies that a quantum computer might be infinitely faster than any possible classical computer for certain classes of problems that are considered intractable for classical computers.

1.1.5 Shor's Algorithm

Perhaps the most practical example of this is the problem of factoring the product N of two large primes. The best currently known classical algorithm, the general number field sieve [Lenstra and Lenstra, 1993], requires sub-exponential time $O(e^{(\log N)^{1/3}(\log\log N)^{2/3}})$, while a quantum computing algorithm proposed by Peter Shor in 1997 requires polynomial time $O((\log N)^3)$ [Shor, 1997].

The algorithm is based on reducing the factoring problem to the problem of finding the period r of the function $f(x) = a^x \bmod N$, where a is a random number less than N. If a and r fulfill certain requirements, one of the factors of N is the $\gcd(a^{r/2} \pm 1, N)$, otherwise the procedure is repeated for a new random value for a. This approach makes use of the quantum computer's power by employing it for the period finding step. Since the quantum computer can evaluate $f(x)$ for many values of x simultaneously, it can very quickly generate the period r by fourier transforming interference patterns of different evaluations of $f(x)$.

Many of the modern data encryption schemes, e.g. RSA, are based on the intractability of the factoring problem. Therefore there is great interest by intelligence agencies to develop a system that can implement Shor's algorithm to gauge if present encrypted data will remain secure in the future.

1.1.6 Quantum Annealing

Another problem of great interest is that of function optimization. The biggest challenge in optimization is finding global rather than local optima. A quantum computer is speculated to be very efficient at achieving this by allowing the "solution" to tunnel out of local minima that it might be stuck in [Apolloni et al., 1989]. A possible implementation would consist of encoding the function to be optimized into an energy potential landscape. For this, the quantum system is initially prepared in its state of lowest energy. If the potential of interest is turned on slowly enough, the system will adiabatically remain in its ground-state. After the potential is fully turned on, this ground-state will encode the solution to the optimization problem.

1.1.7 Quantum Simulation

In analogy to the Church-Turing Thesis, it has been shown that every physical quantum system can be simulated efficiently on a general quantum computer [Deutsch, 1985]. Such capabilities would be extremely helpful in fields like chemistry and medicine, where they would allow a more detailed understanding of complex molecules and their interactions.

1.2 The Power of Quantum Computers

The power of quantum computers roots in two revolutionary concepts that quantum mechanics introduced: Superpositions and Entanglement.

1.2.1 Superpositions

According to quantum mechanics any system is described by a set of discrete states in which it can exist: the system's "eigenstates". It is possible for the system to exist in a superposition of these states, i.e. to be in multiple states at the same time. For example, a quantum bit can not only be in the 0 **or** 1 state, but it can be in both, 0 **and** 1, at the same time. To describe the full state of a quantum system, each eigenstate is given a complex amplitude that describes its weight, called the "probability amplitude". A measurement of the system will then force it to "choose" between one of its eigenstates (in the basis of the measurement). The probability for each eigenstate to be chosen is given by the square of its probability amplitude. After the measurement, the system's state "collapses" to the chosen eigenstate, i.e. the chosen eigenstate's probability amplitude becomes 1, while all others go to 0. Since measurements always yield an answer, the square of the probability amplitudes for all states needs to sum to 1, i.e. one of the states has to be chosen.

In terms of quantum bits, this means that a quantum computer can not only use the two states of the bit (0 and 1) to do binary calculations, but can instead use the complex probability amplitude of the 1 state, which provides two analog values (the relative phase and amplitude of the 0 and 1 state) for calculations. This concept is discussed further in Chapter 3.3.1.

1.2.2 Entanglement

Superpositions of single qubits alone do not provide a significant advantage, though, since they can be efficiently simulated by a classical computer by using a collection of classical bits to store the probability amplitudes. The true "magic" happens when several quantum systems are allowed to interact. According to quantum mechanics, the state of a collection of interacting quantum systems can no longer be described as a collection of the states of the individual systems, but instead needs to be described in terms of a new set of states that consists of all possible combinations of the individual states.

For example, for three quantum bits, the combined system is not described in terms of the three qubits' individual states that each are 0 or 1, but instead by the eight new states 000, 001, 010, 011, 100, 101, 110, and 111. In general, a system consisting of n interacting quantum bits needs to be described in terms of 2^n quantum states. A quantum computer can then be in a superposition of these 2^n

states, effectively giving it $2^{n+1} - 2$ analog numbers for calculations in the form of the complex probability amplitudes of each of the 2^n states. Therefore, the number of "registers" that a quantum computer has available for calculations scales exponentially with its number of quantum bits, while for a classical computer this scaling is linear.

If a quantum computer then performs an operation on one of the bits, this one operation will affect the probability amplitudes of all 2^n states. For example, if a three bit quantum computer performs a "NOT" operation on its first bit, it effectively performs four swap operations that exchange the probability amplitudes of the 000 and 100, the 001 and 101, the 010 and 110, as well as the 011 and 111 states. This makes it possible to design algorithms that allow a quantum computer to process data in a massively parallel fashion, with the number of possible simultaneous operations scaling exponentially with the number of its quantum bits.

This makes a 65-bit quantum computer twice as powerful as a 64-bit quantum computer, while a 65-bit classical computer is only about 1.5% more powerful than a 64-bit classical computer.

1.2.3 Implications – The EPR Paradox

Clearly, this exponential scaling of a quantum computer's power with its number of bits does not make immediate intuitive sense. But thinking about the implications of these two ideas – superpositions and entanglement – leads to far more seemingly paradoxical scenarios. For example, if two quantum bits are put into a superposition of the states 10 and 01, i.e. in a perfectly anti-correlated state, a measurement of both qubits will always yield an opposite result for the two bits, i.e. either a 0 and a 1 or a 1 and a 0. Quantum mechanics states that this anti-correlation remains even if the two bits are separated and brought to opposite ends of the universe. The apparent paradox arises from the fact that, even though quantum mechanics tells us that both qubits will yield the opposite measurement result with certainty, it states that it is impossible to predict which of the qubits will yield a 0 and which one will yield a 1 since this choice is made only at the time of measurement. Thus, as soon as one of the bits "decides" on its measurement outcome, it needs to instantaneously inform the other bit about its decision no matter how far away it is. As of today, though, this communication between the bits is presumed impossible as it would have to happen at speeds faster than the speed of light, which is currently believed to be the maximum speed at which anything, even information, can travel. This paradox was first published by Einstein, Podolsky, and Rosen [Einstein et al., 1935] and is commonly referred to as the “spooky action at a distance”, a term coined by Einstein.

As explained further in Chapter 10, this gedanken-experiment was eventually formulated into mathematical inequalities, called “Bell’s Inequalities” [Bell, 1964, Clauser and Horne, 1974, Clauser et al., 1969], that could experimentally test whether the apparently paradoxical predictions made by quantum mechanics could be resolved by the introduction of a deterministic alternative theory, or whether the ideas had to be accepted as true. These inequalities are based on the definition of measures of correlation that can be shown to be limited to a certain value in classical systems that do not require entanglement and superpositions to explain their state. Quantum mechanical systems, though, with “access” to these entanglements and superpositions, can show correlation values that exceed these classical limits and are thus said to “violate Bell’s inequality”.

This makes the demonstration of a violation of Bell’s inequality in a system of proposed quantum bits a strong proof that these quantum bits indeed show behavior that goes beyond what is explainable by classical deterministic theories. Such a demonstration could therefore be seen as very strong evidence that the proposed architecture does promise the capability to eventually lead to the exponential performance increases suggested by quantum mechanics.

This has made the violation of Bell’s inequality a major milestone for any proposed architecture of quantum bits [Clarke and Wilhelm, 2008]. This thesis shall meet this milestone for the first time for the Superconducting Josephson Phase Qubit architecture. Furthermore, this experiment is, to our knowledge, the first demonstration of a violation of Bell's inequality in a solid state system, and the first demonstration using a macroscopic quantum state.

1.3 Requirements – The DiVincenzo Criteria

To establish a guideline along which to quickly evaluate proposed approaches for the implementation of a working quantum computer, David DiVincenzo compiled a list of five criteria [DiVincenzo, 2000] that any candidate system must fulfill to be considered feasible. According to the list, any feasible approach must provide:

- A scalable physical system with well-defined qubits
- Initialization to a simple fiducial state
- Sufficiently long coherence times
- A universal set of quantum gates
- High quantum efficiency, qubit-specific measurements

1.3.1 Scalable Physical System with Well-Defined Qubits

A classical computer can approximate the solution to any problem to arbitrary precision using only binary operations. Similarly, it has been shown that a quantum computer can perform any arbitrary computation using only two-state quantum bits, called “Qubits”, as the operating unit [Barenco et al., 1995]. It is not necessary to implement “Qudits” with more than two levels as they can be efficiently simulated by a sufficiently large collection of qubits. The required number of qubits can grow quickly, though, making it necessary for the proposed architecture to be scalable.

1.3.2 Initializable to a Simple Fiducial State

The architecture needs to provide a reliable way to initialize its set of qubits into an arbitrary, but known, starting state. This state can be as trivial as each qubit being set to “0”.

1.3.3 Sufficiently Long Coherence Times

One of the major problems of many current approaches is the short timescale over which the information stored inside the qubits dissipates into the environment. This has prompted the inclusion of a statement about the qubits’ "coherence times", i.e. the time-scales of information retention, into the list of requirements. Once coherence times reach a system-dependent threshold, it becomes possible to employ error correction techniques that allow for information to be stored indefinitely [Shor, 1995]. Any feasible system must eventually be able to reach this threshold.

1.3.4 Universal Set of Quantum Gates

To be considered a generally useful quantum computer, the architecture needs to support a basic universal set of operations that suffices as a basis set for constructing any arbitrary computation [Barenco et al., 1995]. For a classical computer, the corresponding set commonly consists of just the $NAND$ gate. Neither in classical nor in quantum computation this set of operations is unique.

For quantum systems, the common example-set used by theoretical physicists contains the Clifford Gates H and $cNOT$ augmented by the $P_{\pi/8}$ gate. The H, or Hadamard, gate translates the $|\,0\,\rangle$ state of the qubit into the $|\,0\,\rangle + |\,1\,\rangle$ state and the $|\,1\,\rangle$ state into the $|\,0\,\rangle - |\,1\,\rangle$ state. The $cNOT$, or controlled NOT, gate performs a NOT operation on one of two qubits if and only if the other qubit is in the $|\,1\,\rangle$ state and corresponds to the classical XOR gate. The $P_{\pi/8}$, or $\pi/8$ phase, gate changes the phase of the qubit's $|\,1\,\rangle$ state relative to the $|\,0\,\rangle$ state by $\pi/8$. The actual set of gates provided by any given implementation may deviate from this, but must still be universal. To prove universality, it is sufficient to show how the actual set can be used to implement the gates mentioned above.

Even though being able to directly implement operations involving more than two qubits might lead to a performance increase, they are not required since it has been shown that any n-qubit operation can be implemented with the above mentioned universal set, even though it only includes the two-qubit $cNOT$ gate.

1.3.5 High Quantum Efficiency, Qubit-Specific Measurements

To extract the result of the computation in a way that is compatible with the universality requirement, it is necessary to be able to measure the state of each qubit independently. This measurement needs to be sufficiently accurate, since it is a common part of error correcting schemes that are needed to overcome short qubit coherence times.

Chapter 2

Superconducting Josephson Qubits

2.1 Motivation

Many factors contribute to the decision of which approach to follow in an effort to build a quantum computer. While some approaches, like NMR [Vandersypen et al., 2001] and Ion Traps [Benhelm et al., 2008], yield the immediate satisfaction of naturally good single qubit performance, scalability presents a major hurdle to these systems due to the difficulty of arranging a multitude of these qubits into a layout that allows them to be coupled and controlled in a useful way. Superconducting Josephson qubits fall at the almost opposite end of the spectrum.

While the underlying technology promises extremely good scalability, single qubit performance still is a major challenge to all groups in the field.

The decision to pursue the Josephson qubit approach was based on several factors that promised to naturally address some of the DiVincenzo criteria.

2.1.1 Long Coherence Time

The qubit is formed by quantum states in a superconductor, a material owing its name to the fact that it exhibits no resistance to electrical currents. Early experiments with superconducting magnets have lead to estimated lifetimes of established electrical currents inside such magnets in the hundred thousand year range [Gallop, 1990]. Since energy dissipation is one of the major sources of qubit decoherence, the apparent absence of resistance as a loss mechanism in superconducting circuits lead to the hope that these systems could support qubit states coherently for a very long time.

2.1.2 Scalability

The quantum states in Josephson qubits consist of currents and voltages inside electrical circuits built from mostly standard circuit elements like wires, inductors, capacitors, transformers, etc. As such, the majority of the circuit's behavior can be readily analyzed using standard circuit theory. This greatly simplifies the design of the glue-circuitry needed to connect multiple qubits to their control electronics and to each other.

Furthermore, the circuit, once designed, is fabricated in much the same way as a conventional integrated circuit. These two factors should allow for very straightforward scaling, once the single qubit circuit element is understood. The scalability of integrated circuit technology has been proven excessively over the last decades by the incredible increases in complexity of standard computer processors. As of today, there are no obvious indicators that call the applicability of this scalability to quantum circuits into question.

2.1.3 Initialization, Control and Measurement

Over the past decades, the arsenal of integrated circuits that can provide exquisite voltage and current control and measurement has grown immensely and is becoming continuously more affordable. Specifically the microwave electronics industry has grown rapidly thanks to the high demand for wireless devices of all sorts. The frequencies, voltage and current levels, and control accuracies required for the purposes of building superconducting quantum bits match the industry standards extremely well. Thus, commercial control and readout electronics is very available, giving a lot of flexibility to the design of operation and readout schemes.

The specifics of implementing the initialization, measurement, and universal control required by the DiVincenzo criteria, of course, need to be solved with the quantum nature of the circuits in mind and can thus not be assumed as a priori obvious. Nevertheless this flexibility is making it easy to find feasible solutions.

2.2 Idea

2.2.1 Quantum Mechanics in Electrical Circuits

To turn an electrical circuit into a qubit, one needs to find a regime in which it exhibits quantum behavior. The simplest general system that exhibits quantum behavior is the harmonic oscillator, provided it is driven at extremely low energies. Thus, it is natural to start the circuit design with the electrical analog of the harmonic oscillator, the inductor-capacitor oscillator (or LC oscillator for short). In this circuit, a capacitor and an inductor are connected in series inside a loop configuration as shown in Figure 2.1a. If a charge is present on the capacitor, it tries to force electrons around the loop to remove the charge, while the inductor in the circuit tries to maintain the current flowing through it at a constant level.

This circuit can be readily analyzed using Kirchhoff's current law. All current flowing around the loop is flowing through the inductor and the capacitor, which

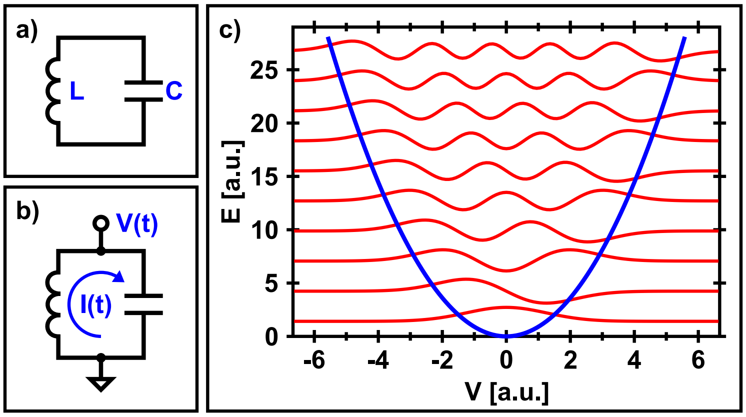

Figure 2.1: Inductor-Capacitor Oscillator – a) Electrical circuit: An inductor with inductance L in parallel with a capacitor with capacitance C. b) Relevant current and voltage: The voltage $V(t)$ across the capacitor drives a current $I(t)$ around the oscillator loop through the inductor. c) Oscillator potential with eigenstates: The potential energy (blue) is a parabolic function of the voltage $V(t)$ this leads to the familiar quantum eigenstates (red) of the simple harmonic oscillator.

gives the following equality if the voltage across the capacitor is called $V(t)$:

$$I(t) = -\frac{1}{L}\int V(t)\,dt = C\,\frac{d}{dt}V(t) \tag{2.1}$$

Taking the derivative gives:

$$C\,\frac{d^2}{dt^2}V(t) = -\frac{1}{L}\,V(t) \tag{2.2}$$

This relation corresponds to the equation of motion of a pendulum:

$$F = ma = m\,\frac{d^2}{dt^2}x(t) = -k\,x(t) \tag{2.3}$$

with $m = C$, $x = V$, and $k = \frac{1}{L}$.

Equation 2.2 can be rewritten as:

$$C\,\frac{d^2}{dt^2}V(t) + \frac{\partial}{\partial V}\left(\frac{1}{2L}\,V^2\right) = 0 \tag{2.4}$$

Here, the "force" acting on the oscillator has been expressed as the partial derivative of the system's potential energy as a parabolic function of the "position" V. Solving the time independent Schrödinger equation (see Section 3.1) yields the familiar quantum eigenstates of the potential:

$$\langle\, x \,|\, \psi_n \,\rangle = \frac{1}{\sqrt{2^n\, n!}}\,\sqrt[4]{\frac{C\,\omega}{\pi\,\hbar}}\; e^{-\frac{C\,\omega}{2\,\hbar}\,x^2} H_n\left(\sqrt{\frac{C\,\omega}{\hbar}}\;x\right) \text{ for } n = 0, 1, 2, \ldots \tag{2.5}$$

Here, H_n are the Hermite polynomials and the angular frequency ω is defined as $1/\sqrt{LC}$. The energy levels corresponding to the eigenstates are:

$$E_n = \hbar\,\omega\,\left(n + \frac{1}{2}\right) \tag{2.6}$$

It is possible to operate the LC oscillator in a regime that allows the observation of quantum mechanical behavior, provided three conditions are met:

- The wiring metal needs to be chosen such that it can be operated below its superconducting transition temperature. Otherwise the resistance in the leads will cause energy decay rates that are much too high to observe meaningful quantum effects. In the analogy, the damping term in the pendulum has to be small enough to allow for quantum excitations to exist for a meaningful time.

- The circuit needs to be cooled to a temperature that does not impact the quantum state of the oscillator via thermal excitations. This is the case when the thermal energy available to the system is less than the energy difference between the oscillator's quantum levels:

$$E_{thermal} = k\,T \ll \hbar\,\omega \tag{2.7}$$

- The control of the circuit needs to be sophisticated enough to create and detect quantum mechanical excitations corresponding to single photons at a time.

The last of these three conditions makes it hard to use LC oscillators directly as qubits. The problem is that the only way to achieve single-photon control over the oscillator is by coupling it to another quantum system [Hofheinz et al., 2009]. This is caused by the fact that all consecutive quantum energy levels E_n in the oscillator are separated by the same energy difference $\Delta E = \hbar\,\omega$. This allows all transitions between neighboring energy levels in the oscillator to absorb photons of only energy $\hbar\,\omega$, making it impossible to selectively address only one of the transitions with a classical excitation (see Section 3.3). Instead, the oscillator will be driven into a coherent state that is nearly indistinguishable from a classical state and is not useful for quantum computation. Thus, to build a quantum bit where the ground and first excited states can be addressed separately from all other states using a classical excitation, one needs to change the spacing of the energy levels, i.e. introduce a non-linearity into the potential. In general it is very straightforward to make circuits behave in a non-linear fashion. In fact, almost all circuits will show some kind of non-linear behavior if they are driven hard enough. The problem here is to build a circuit that does so with only a single photon of energy. Luckily, nature provides a way to achieve just that: The Josephson tunnel junction [Josephson, 1962].

2.2.2 Josephson Tunnel Junctions

A Josephson tunnel junction is formed every time a superconducting lead is interrupted by a thin barrier through which electrons can tunnel. The tunneling provides a weak link between the two superconducting regions and allows the wave functions of the superconducting order parameter on the two sides to interfere. This leads to very interesting electrical characteristics of the junction that are captured by the "Josephson Relations", named after Brian Josephson who received the Nobel Prize in 1973 for predicting them:

$$V(t) = \frac{\Phi_0}{2\pi}\frac{d}{dt}\delta(t) \tag{2.8}$$

$$I(t) = I_c \sin\delta(t) \tag{2.9}$$

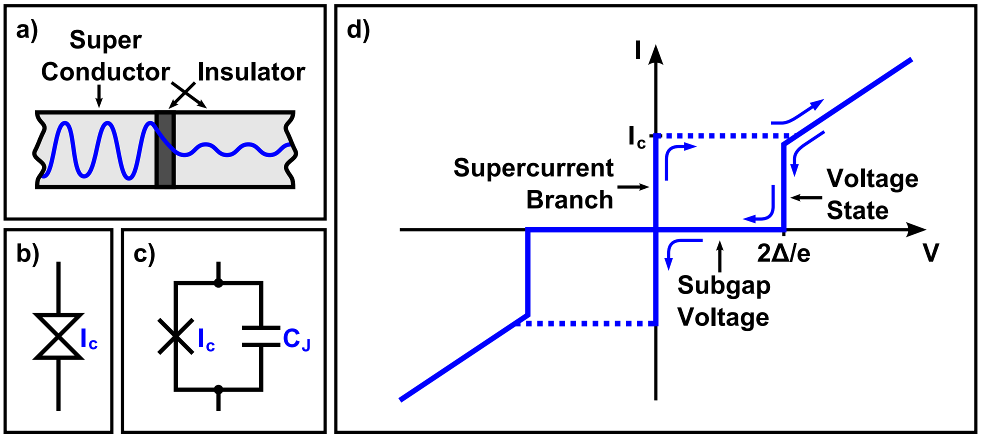

Figure 2.2: Josephson Tunnel Junction – a) Physical structure: Two superconducting regions separated by an insulating barrier. b) Circuit symbol: Possibly depicting point-contact used in early junction fabrication. c) Effective circuit element: The physical structure forms a parallel plate capacitor shunting the tunnel junction. d) Current-voltage response: The junction's response to an oscillating bias is hysteretic and highly non-linear.

Here, the voltage $V(t)$, the current $I(t)$, and the superconducting phase difference $\delta(t)$ across the junction are classical variables. I_c is the "critical current" of the junction, which depends on the barrier thickness, and $\Phi_0 = \frac{h}{2e}$ is the flux quantum.

Figure 2.2d shows the electrical response of a Josephson junction on an I/V plot. The plot shows three distinct features:

- The vertical line along the I-axis is commonly called the "zero-voltage state" or the "supercurrent branch". In this regime, the value of $\delta(t)$ is constant $\left(-\frac{\pi}{2} < \delta(t) < \frac{\pi}{2}\right)$, which implies $V(t) = 0$. It can be seen from Equation 2.9 that the maximum current that can be conducted in this way is $|I(t)| \leq I_c$.

- The horizontal line along the V-axis, called the “sub-gap voltage”, corresponds to a continuously increasing value of $\delta(t)$ as given by Equation 2.8. This leads to a rapidly oscillating current which averages to zero.

- When the voltage across the junction is large enough to create one quasi-particle-excitation in each superconducting lead, the junction will be able to conduct electricity via quasi-particle tunneling. This process lets the junction behave like a simple resistor, leading to the diagonal lines extending to the edges of the plot. The voltage that is required to create the two excitations is equal to twice the superconducting gap Δ, a material property that depends on the superconductor in use. For aluminum the gap is around $190\,\mu$V.

If the junction is driven with a slowly oscillating current bias, its response first follows the supercurrent branch along the I-axis to I_c. Then, the junction “switches” to the “voltage state” and follows the diagonal trace to the edges of the plot. As the bias current drops, the junction responds hysteretically and traces out the remainder of the curve including the subgap voltage. Once the bias current reaches zero, the junction returns to the supercurrent branch.

The cleanliness of the trace yields information about the quality of the fabrication. Specifically, the flatness of the subgap voltage, i.e. the horizontal part of the trace along the V-axis, gives clues about the quality of the insulating barrier of the junction.

The geometry of the junction causes it to also behave like a classical parallel plate capacitor, which can be understood as simply shunting the junction in parallel as shown in Figure 2.2c. The junction's capacitance C_J depends only on its geometrical parameters according to the usual formula:

$$C_J = \frac{\varepsilon\, A}{d} \tag{2.10}$$

Here, ε is the dielectric constant of the insulator, A is the junction's area, and d the insulator's thickness. Since the junction's capacitance scales linearly with the barrier thickness while its critical current scales exponentially, it is possible to control C_J and I_c independently of each other.

2.2.3 Circuit Potential

As the junction's capacitance behaves exactly like a conventional capacitor, the most straightforward way to integrate the junction into the standard LC-oscillator circuit is by putting it in the place of the oscillator's capacitor and designing it such that $C_J = C$. The resulting circuit can be analyzed in a variety of different ways. The easiest approach is to simply use the Josephson relations and Kirchhoff's current law. For this, the current $I(t)$ flowing through the oscillator's inductor

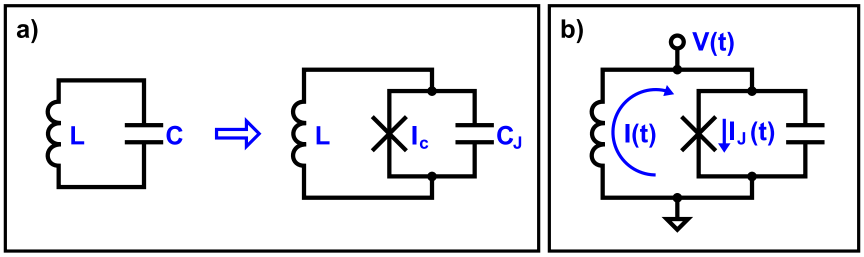

Figure 2.3: Modified Inductor-Capacitor Oscillator – a) Circuit modification: The oscillator's capacitor is replaced by a josephson tunnel junction with its intrinsic capacitance C_J designed to match the original capacitance C. b) Relevant currents and voltages: The voltage $V(t)$ across the capacitor drives the current $I_J(t)$ through the junction and $I(t)$ around the oscillator loop through the inductor.

is set equal to the current flowing through the junction and its capacitor given a voltage $V(t)$ across the elements:

$$I(t) = -\frac{1}{L}\int V(t)\,dt = I_J(t) + C\,\frac{d}{dt}V(t) \tag{2.11}$$

Plugging in the Josephson relations gives:

$$I(t) = -\frac{1}{L}\int \frac{\Phi_0}{2\pi}\,\frac{d}{dt}\delta(t)\,dt = I_c \sin\delta(t) + C\,\frac{d}{dt}\frac{\Phi_0}{2\pi}\,\frac{d}{dt}\delta(t) \tag{2.12}$$

This can be rewritten as:

$$C\left(\frac{\Phi_0}{2\pi}\right)^2 \frac{d^2}{dt^2}\delta + \frac{\partial}{\partial\delta}\left(-\frac{\Phi_0}{2\pi}\,I_c\cos\delta + \frac{1}{2L}\left(\frac{\Phi_0}{2\pi}\right)^2\delta^2 - \frac{\Phi_0}{2\pi}\,I_{dc}\,\delta\right) = 0 \tag{2.13}$$

Here, δ is a function of t, and I_{dc} is the constant of integration which captures an initial flux bias applied to the inductor.

This equation can be interpreted as an equation of motion of a particle of mass $m = C\left(\frac{\Phi_0}{2\pi}\right)^2$ at position δ in the potential $V(\delta) = -\frac{\Phi_0}{2\pi}\,I_c\cos\delta + \frac{1}{2L}\left(\frac{\Phi_0}{2\pi}\right)^2\delta^2 -$

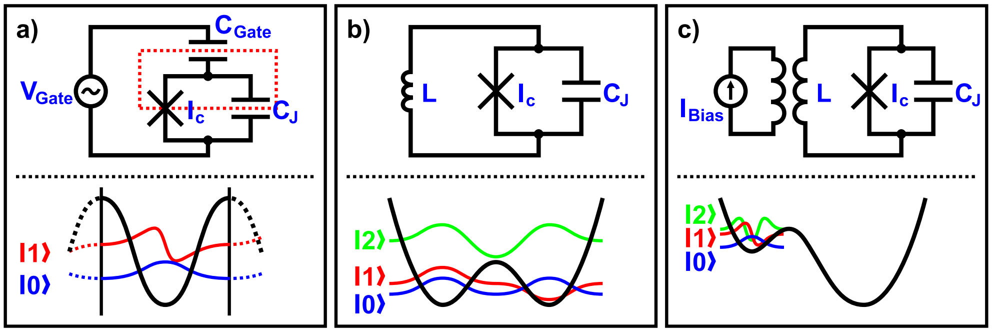

Figure 2.4: Josephson Qubits: Slight differences in the circuit can lead to very different qubits. – a) Charge qubit: By removing the inductor and replacing it with a bias capacitor, the charge qubit circuit creates an island (dashed red box) onto which cooper pairs can tunnel. This leads to a periodic potential that supports the qubit states. b) Flux qubit: A small inductance leads to a parabolic potential with a small bump created by the Josephson junction. This creates two minima that together support the two qubit states. c) Phase qubit: A larger inductance allows the Josephson junction to influence the potential more significantly. By biasing the circuit, the potential can be tilted to create a shallow minimum along one of the sidewalls of the parabolic part of the potential. This minimum holds several unevenly spaced energy levels, the lowest two of which form the qubit states.

$\frac{\Phi_0}{2\pi} I_{dc} \delta$. This potential consists of three parts: a parabola created by the inductor, a cosine oscillation created by the Josephson junction and a tilt that can be applied by flux-biasing the inductor.

2.2.4 Circuit Parameters

Depending on the choice of the circuit parameters – the inductance, capacitance, and critical current – the circuit can be turned into one of three different types of quantum bits as indicated in Figure 2.4.

If the inductor is removed from the circuit and replaced by an open, i.e. $L = \infty$, the $\frac{1}{L}\,\delta^2$ term drops out of the potential equation. If no current bias is applied either, i.e. $I_{dc} = 0$, the only term left is the cosine. One can now understand the circuit diagram as forming an island (as indicated in Figure 2.4a) onto and from which electrons can tunnel via the Josephson junction. This island can be biased via a gate capacitor to influence the equilibrium number of electrons on the island. The number of excess electrons, the charge, on the island is then well quantified and can be used as the qubit state. This design is therefore called the "Charge Qubit" [Nakamura et al., 1999].

On the other extreme, one can choose a small inductance, leading to a potential that is very close to the parabolic potential of the harmonic oscillator. If the parameters are chosen such that the overlaying cosine simply adds a small bump to the bottom of this parabola, one can create an energy landscape with two minima that together only support two energy levels in a symmetric or antisymmetric configuration as shown in Figure 2.4b. The next higher levels span across the bump leading to a large energy difference between the desired transition and all undesired ones. The two lowest states correspond to two distinct currents flowing in the loop creating two well defined flux states in the inductor. Thus, these qubits are called "Flux Qubits" [Orlando et al., 1999].

In between these two extremes, one can set up a potential landscape where the cosine forms a local minimum along one of the sidewalls of the parabola as shown in Figure 2.4c rather than two equally sized ones in the center. This will lead to quantum states that correspond to a very well quantified phase difference across the tunnel junction. This type of qubit is therefore called "Phase Qubit" [Martinis et al., 2002]. Historically the phase qubit has trailed behind the flux and charge qubits in terms of the best reported energy relaxation times. But it makes up for its lower T_1 with much higher visibility single-shot readout, ease of coupling, and frequency tuneability. This makes it currently the best candidate for implementing an experiment to violate Bell's inequality in superconducting circuits.

2.3 Superconducting Qubit Operation

2.3.1 Initialization

Due to the relatively short energy relaxation times, all three types of qubits automatically initialize into their ground state once cooled to their operating temperature.

Since the states in the phase qubit are localized in any one of possibly many stable minima in the potential, the qubit must be biased with a proper sequence to "guide" the relaxation into the desired minimum. This can be achieved either with a constant bias at which the potential only has one minimum or with an oscillating bias that destabilizes all minima except for the desired one (see Chapter 8.2).

2.3.2 Single Qubit Gates

Single qubit operations on all types of superconducting qubits are performed by applying DC or RF pulses to the circuit via its bias line.

In the case of the charge qubit, this bias takes the form of a charge bias applied across the gate capacitor.

For the flux and phase qubit, the bias consists of a flux applied to the qubit's inductive loop.

As explained in Section 3.3, the DC or RF biases can be used to create rotations around the Z-axis or around a vector in the X/Y-plane of the Bloch sphere.

2.3.3 Multi Qubit Coupling

The methods used to couple qubits of the three types vary greatly. This is primarily due to the difference in the electrical impedance of the different qubit circuits.

The phase qubit has the lowest impedance ($\sim 30\,\Omega$) and can thus be coupled with simple capacitive or inductive circuits. The most trivial coupling element consists of just a capacitor wired between two qubits as shown in Figure 2.5a

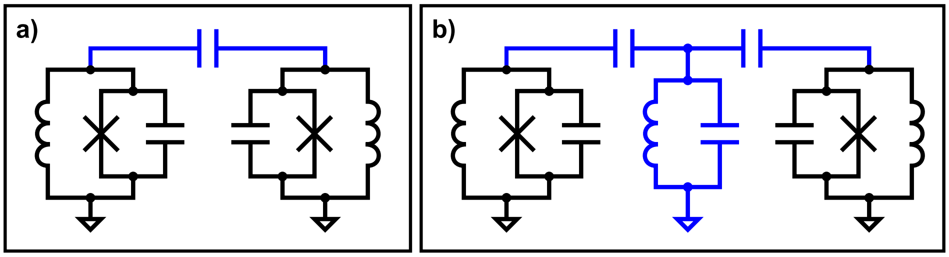

Figure 2.5: Example Qubit Coupling Circuits – a) Capacitive coupling: Phase qubits can be easily coupled via a capacitor that provides fixed-strength always-on coupling. b) Resonator coupling: Phase and charge qubits can be coupled via a resonant bus. For the charge qubit the resonator enables long-distance coupling despite the qubit's high impedance, while for the phase qubit the resonator can be used to provide a band-pass filter for the coupling.

[McDermott et al., 2005]. The low impedance allows the coupling wiring to be quite long and thus gives great flexibility in the design of coupling geometries for many qubits.

The flux qubit is commonly coupled by placing two qubits right next to each other [Majer et al., 2005]. This allows their flux degree of freedom to interact via the resulting mutual inductance between them. Long-distance coupling between flux qubits has not yet been demonstrated, but a one-dimensional chain of nearest-neighbor coupled qubits is sufficient to achieve universal quantum computation. The required fidelities to successfully implement error correction for such a geometry are much more stringent, though.

The high-impedance charge qubit is the least flexible when it comes to coupling. It cannot be coupled with simple wires as the capacitive impedance to ground of a wire of any usable length would be much smaller than that of the qubit, effectively shorting the coupling. The current proposal by the involved groups consists of placing the qubits inside the cavity of a coplanar waveguide resonator [Majer et al., 2007]. Coupling only via resonant excitations in the resonator then allows for long-distance communication between several qubits. The lack of frequency tuneability of the charge qubit makes this method of coupling fairly complex.

2.3.4 Readout

There are four primary types of readout schemes that differ in two binary properties.

The readout can be either single-shot or non-single-shot. Single-shot readout projects every qubit involved in the experiment into the $|\,0\,\rangle$ or $|\,1\,\rangle$ state and returns one specific final output state like $|\,0110001\,\rangle$ for each experimental run. Non-single-shot readout schemes return a measurement that is an analog function of the possible output states. A simple example could consist of a measurement of the total energy stored in all qubits, which is proportional to the number of qubits in the $|\,1\,\rangle$ state. Even though single-shot readout is not required for quantum computation, it is highly desirable as it yields results that don't require extensive post-processing or calibration to extract the actual state probabilities.

Readout schemes can further be categorized by whether they destroy the quantum state of the measured qubit necessitating re-initialization, or whether they project the qubit into a state that can then be used in further operations. The latter are called “Quantum Non-Demolition” (QND) readout schemes. QND readout might be desirable in the future for implementing error correction algorithms that rely on classical feedback.

The charge qubit is commonly read out by the effect of the qubit state onto a coplanar resonator's microwave transmission behavior [Wallraff et al., 2005]. A $|\,1\,\rangle$ state will cause a phase shift in the transmitted signal that differs from the one caused by a $|\,0\,\rangle$ state by a measurable amount. Since this phase shift will be a similar function of the states of all qubits connected to the same resonator, this readout scheme is not single-shot. But as the readout does not destroy the qubit state, it is a QND measurement.

Both the flux and the phase qubit use a Superconducting Quantum Interference Device (SQuID) to read out the qubit state [Lupascu et al., 2004, Cooper et al., 2004]. Squids are circuits that are highly sensitive to magnetic flux biasing. They can be built by placing one or more Josephson junctions into an inductive loop as shown in Figure 2.6a. The loop will turn an applied flux bias into a current bias through the junction(s). This current bias will add to any additional externally applied current bias. This changes the value of the external current bias that

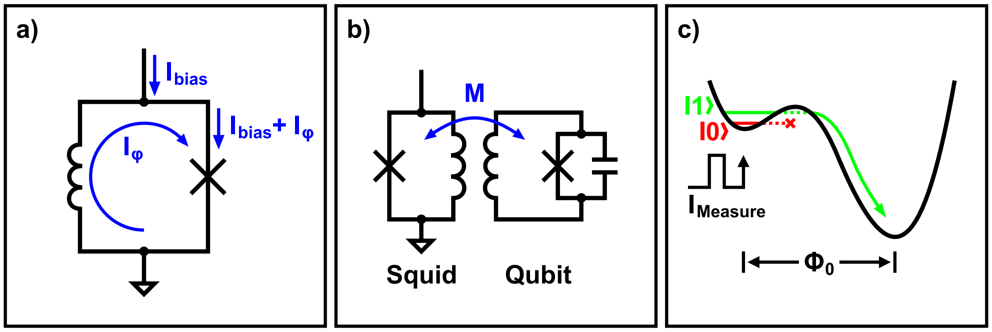

Figure 2.6: Squid Readout Scheme – a) Squid: A squid, to first order, behaves like a junction with a critical bias current I_{bias_c} that can be tuned via an applied flux bias current I_ϕ. b) Coupling: The squid is coupled to the qubit via a mutual inductance M. c) Phase qubit measurement: A measure pulse $I_{Measure}$ temporarily lowers the barrier between the operating minimum and the neighboring minimum to the point where the $|\,1\,\rangle$ state can tunnel, while the $|\,0\,\rangle$ state remains trapped. This results in a flux difference in the qubit loop between the $|\,0\,\rangle$ and $|\,1\,\rangle$ state of about one Φ_0.

causes the junction to exceed its critical current and switch to the voltage state. Squids can detect magnetic flux biases as small as fractions of a flux quantum, making them useful for high-sensitivity applications like MRI or qubit readout.

In the flux qubit, the $|\,1\,\rangle$ state naturally causes a different flux bias in a neighboring squid than the $|\,0\,\rangle$ state. This allows this qubit to be read out directly. If each qubit is provided with its own squid, the readout can be single-shot. The switching of the squid to the voltage state releases a large amount of energy into the circuit and thus randomizes the qubit state making it a non-QND readout scheme.

Since the $|\,0\,\rangle$ and $|\,1\,\rangle$ state of the phase qubit correspond to fairly similar flux states in the qubit's inductor, the state needs to be transcoded into a more easily distinguishable set of flux states before it can be measured. This is achieved by tilting the qubit until the minimum in which the operations have been performed becomes almost unstable as shown in Figure 2.6c. At this point, the higher energy $|\,1\,\rangle$ state tunnels out of the minimum and settles into the neighboring minimum while the $|\,0\,\rangle$ state remains in the operating minimum. This gives a difference in flux in the qubit loop of about one flux quantum between the two states. This can be detected easily with a squid that is inductively coupled to the qubit loop. Just like in the case of the flux qubit, this readout scheme is single-shot, but not QND.

Chapter 3

Understanding Qubits Numerically

3.1 Solving the Schrödinger Equation

The evolution of any quantum system is described by the Schrödinger equation [Schrödinger, 1926], named after Erwin Schrödinger who discovered it in 1926:

$$i\hbar\,\partial_t\,\psi\left(\mathbf{r},t\right)=\hat{H}\,\psi\left(\mathbf{r},t\right) \tag{3.1}$$

Here, $\psi\left(\mathbf{r},t\right)$ is the state of the system expressed by its "wave-function" as a function of position $\mathbf{r}$ and time t. The wave-function is also called the "probability amplitude" as its square gives the probability of finding the system at position $\mathbf{r}$

at time t. $\mathbf{r}$ is not restricted to be a physical position, but contains all measures of interest about the system. $\hat{H}$ is called the "Hamiltonian operator" and calculates the total energy of the system if applied to $\psi\left(\mathbf{r},t\right)$.

3.1.1 Time Dependent versus Time Independent Part

The Schrödinger equation is usually solved by separation of variables using:

$$\psi\left(\mathbf{r},t\right) = \psi^{r}\left(\mathbf{r}\right)\,\psi^{t}\left(t\right) \tag{3.2}$$

Plugging into Equation 3.1:

$$i\hbar\,\partial_t\,\left(\psi^{r}\left(\mathbf{r}\right)\,\psi^{t}\left(t\right)\right) = \hat{H}\,\psi^{r}\left(\mathbf{r}\right)\,\psi^{t}\left(t\right) \tag{3.3}$$

If $\hat{H}$ is independent of time, dividing both sides by $\psi^{r}\left(\mathbf{r}\right)\,\psi^{t}\left(t\right)$ gives:

$$i\hbar\,\frac{\partial_t\,\psi^{t}\left(t\right)}{\psi^{t}\left(t\right)} = \frac{\hat{H}\,\psi^{r}\left(\mathbf{r}\right)}{\psi^{r}\left(\mathbf{r}\right)} \tag{3.4}$$

For this equality to hold for all values of t and $\mathbf{r}$, both sides must equal a constant:

$$i\hbar\,\partial_t\,\psi^{t}\left(t\right) = E\,\psi^{t}\left(t\right) \tag{3.5}$$

$$\hat{H}\,\psi^{r}\left(\mathbf{r}\right) = E\,\psi^{r}\left(\mathbf{r}\right) \tag{3.6}$$

The solution to Equation 3.5 is simple and describes the time evolution of the states of the system:

$$\psi^{t}\left(t\right) = e^{-iEt/\hbar} \tag{3.7}$$

Equation 3.6, also called the “time independent Schrödinger equation”, frequently has many possible solutions for different values of E. These solutions, the eigenvectors of $\hat{H}$, are specific to each system and have physical meaning in that they describe the possible pure quantum states that the system can exist in. They are therefore called the system's “eigenstates”. The eigenvalue E corresponding to an eigenstate gives its energy and the eigenstate with the lowest eigenvalue (i.e. lowest energy) describes the ground-state into which the system will relax if it is cooled sufficiently (provided it does not get trapped in a local energy minimum). Commonly, the states are sorted by ascending energy and labeled with an index starting at 0. The n^{th} eigenstate $\psi_n^r(\mathbf{r})$ has energy E_n and is written as $|\, n\,\rangle$. The full solution to the Schrödinger equation for the n^{th} eigenstate is:

$$\psi\left(\mathbf{r}, t\right) = \psi^r\left(\mathbf{r}\right)\, \psi^t\left(t\right) = e^{-iE_n t/\hbar}\, \psi^r\left(\mathbf{r}\right) = e^{-iE_n t/\hbar}\, |\, n\,\rangle \tag{3.8}$$

Since the eigenstates form a complete basis, any possible real state $\psi(\mathbf{r})$ (or $|\,\psi\,\rangle$ for short) that the system might exist in can be written as a linear superposition of eigenstates, i.e.:

$$|\,\psi\,\rangle = \sum_n a_n\, |\, n\,\rangle \tag{3.9}$$

The coefficients a_n are calculated by projection:

$$a_n = \int \psi\left(\mathbf{r}\right)^* \psi_n^r\left(\mathbf{r}\right)\, d\mathbf{r} = \langle\, \psi\, |\, n\,\rangle \tag{3.10}$$

3.1.2 Effects of a Time Dependent Potential

To find the behavior of a system with a time dependent potential, $\hat{H}$ is broken up into a constant and a time dependent part:

$$\hat{H} = \hat{H}_0 + \hat{V}(t) \tag{3.11}$$

The eigenstates are then found for $\hat{V}(t) = 0$ and used as the basis for the evolving state $\psi(\mathbf{r}, t)$ by allowing the superposition weights a_n to vary with time. The Schrödinger equation gives the evolution of the weights:

$$i\hbar\,\partial_t \left(\sum_m a_m(t)\, e^{-iE_m t/\hbar} \,|\, m \,\rangle \right) = \hat{H} \sum_m a_m(t)\, e^{-iE_m t/\hbar} \,|\, m \,\rangle \tag{3.12}$$

Projecting both sides of the equation onto eigenstate $|\, n \,\rangle$ following the convention established in Equation 3.10 gives:

$$\begin{aligned} \langle\, n \,|\, i\hbar\,\partial_t \left(\sum_m a_m(t)\, e^{-iE_m t/\hbar} \,|\, m \,\rangle \right) = \\ \langle\, n \,| \left(\hat{H}_0 + V(t) \right) \sum_m a_m(t)\, e^{-iE_m t/\hbar} \,|\, m \,\rangle \end{aligned}$$

$$\begin{aligned} i\hbar\,\partial_t \left(\sum_m a_m(t)\, e^{-iE_m t/\hbar} \langle\, n \,|\, m \,\rangle \right) &= \sum_m a_m(t)\, e^{-iE_m t/\hbar} \langle\, n \,|\, \hat{H}_0 + V(t) \,|\, m \,\rangle \\ &= \sum_m a_m(t)\, e^{-iE_m t/\hbar} \left(\langle\, n \,|\, \hat{H}_0 \,|\, m \,\rangle + \langle\, n \,|\, V(t) \,|\, m \,\rangle \right) \\ &= \sum_m a_m(t)\, e^{-iE_m t/\hbar} \left(E_m \langle\, n \,|\, m \,\rangle + \langle\, n \,|\, V(t) \,|\, m \,\rangle \right) \quad (3.13) \end{aligned}$$

As the eigenstates are ortho-normal, $\langle\, n \,|\, m \,\rangle = 1$ if $n = m$, otherwise $\langle\, n \,|\, m \,\rangle = 0$:

$$i\hbar\,\partial_t\,\left(a_n(t)\,e^{-iE_nt/\hbar}\right) =$$

$$\sum_m a_m(t)\,e^{-iE_mt/\hbar}\,E_m\,\langle\, n \,|\, m \,\rangle + \sum_m a_m(t)\,e^{-iE_mt/\hbar}\,\langle\, n \,|\, V(t) \,|\, m \,\rangle$$

$$i\hbar\,e^{-iE_nt/\hbar}\,\partial_t\,a_n(t) + E_n\,a_n(t)\,e^{-iE_nt/\hbar} =$$

$$E_n\,a_n(t)\,e^{-iE_nt/\hbar} + \sum_m a_m(t)\,e^{-iE_mt/\hbar}\,\langle\, n \,|\, V(t) \,|\, m \,\rangle$$

$$i\hbar\,\partial_t\,a_n(t) = \sum_m a_m(t)\,e^{i(E_n-E_m)t/\hbar}\,\langle\, n \,|\, V(t) \,|\, m \,\rangle \tag{3.14}$$

$V(t)$ is often a function of $\mathbf{r}$, in which case:

$$\langle\, n \,|\, V(\mathbf{r},t) \,|\, m \,\rangle = \int \psi^r_n\,(\mathbf{r})^*\,V(\mathbf{r},t)\,\psi^r_m\,(\mathbf{r})\;d\mathbf{r} \tag{3.15}$$

3.2 Finding Eigenstates Numerically

The first step in understanding the qubit is to find its eigenstates. The qubit behaves like a particle moving along a single axis x in a potential $V(x)$. Its total energy is given by the particle's kinetic energy T and its potential energy V, i.e.:

$$\hat{H} = T + V = -\frac{\hbar^2}{2m}\,\frac{d^2}{dx^2} + V(x) \tag{3.16}$$

With this, the time independent Schrödinger equation becomes:

$$E_n\,\psi^r_n(x) = -\frac{\hbar^2}{2m}\,\frac{d^2}{dx^2}\,\psi^r_n(x) + V(x)\,\psi^r_n(x) \tag{3.17}$$

In some cases, it is possible to solve this equation analytically, but for more complicated systems, like the qubit circuits, this equation is often solved numerically. This can be done fairly easily (even though not always quickly) by limiting x to a range of interest and discretizing its allowed values. $V(x)$ and $\psi_n^r(x)$ can then be approximated by vectors whose components are the values of $V(x)$ and $\psi_n^r(x)$ at the allowed positions along x.

This can, for example, be applied to a particle in a simple parabolic potential $V(x) = x^2$. If x is limited to the range from -3 to 3 in steps of 1, $\psi_n^r(x)$ and $V(x)$ become:

$$\psi_{\mathbf{n}}^{\mathbf{r}} = (\psi_n^r(-3), \psi_n^r(-2), \psi_n^r(-1), \psi_n^r(0), \psi_n^r(1), \psi_n^r(2), \psi_n^r(3)) \tag{3.18}$$

$$\mathbf{V} = (V(-3), V(-2), V(-1), V(0), V(1), V(2), V(3)) = (9, 4, 1, 0, 1, 4, 9) \tag{3.19}$$

The derivative operator $\frac{d^2}{dx^2}$ can be approximated numerically as the change in the change of $\psi_n^r(x)$ from one x-value to the next, for example:

$$\begin{aligned} \frac{d^2}{dx^2}\,\psi_n^r(x) &= \frac{d}{dx}\frac{d}{dx}\,\psi_n^r(x) = \frac{d}{dx}\,(\psi_n^r(x+0.5) - \psi_n^r(x-0.5)) \\ &= (\psi_n^r(x+1) - \psi_n^r(x)) - (\psi_n^r(x) - \psi_n^r(x-1)) \\ &= \psi_n^r(x+1) + \psi_n^r(x-1) - 2\psi_n^r(x) \end{aligned} \tag{3.20}$$

This can be expressed as a tri-diagonal matrix operating on the vector $\psi^{\mathbf{r}}_{\mathbf{n}}$, here:

$$\frac{d^2}{dx^2} = \mathbf{D} = \begin{bmatrix} -2 & 1 & & & & & \\ 1 & -2 & 1 & & & & \\ & 1 & -2 & 1 & & & \\ & & 1 & -2 & 1 & & \\ & & & 1 & -2 & 1 & \\ & & & & 1 & -2 & 1 \\ & & & & & 1 & -2 \end{bmatrix} \tag{3.21}$$

Note that if x is discretized in steps of $dx \neq 1$, $\mathbf{D}$ will need to be divided by dx^2. Now we can rewrite the time independent Schrödinger equation as a matrix equation:

$$E_n \, \psi^r_n = \left(-\frac{\hbar^2}{2m} \mathbf{D} + \mathbf{I} \, V \right) \, \psi^r_n \tag{3.22}$$

where $\mathbf{I}$ in this case is the 7×7 identity. To solve this equation, one needs to find the eigenvectors of the matrix $\mathbf{M} = (\mathbf{I} \, V - \frac{\hbar^2}{2m} \mathbf{D})$. This can be done using the “eig” function of the “LAPACK” software routines, e.g. via Matlab (eig) or Python (numpy.linalg.eig). For $m = \frac{\hbar^2}{4}$, $\mathbf{M}$ in our example becomes:

$$\mathbf{M} = (\mathbf{I} \, V - 2\mathbf{D}) = \begin{bmatrix} 13 & -2 & & & & & \\ -2 & 8 & -2 & & & & \\ & -2 & 5 & -2 & & & \\ & & -2 & 4 & -2 & & \\ & & & -2 & 5 & -2 & \\ & & & & -2 & 8 & -2 \\ & & & & & -2 & 13 \end{bmatrix} \tag{3.23}$$

The eigenvalues of this matrix are:

$$1.35, 3.91, 6.04, 8.34, 8.85, 13.75, 13.76$$

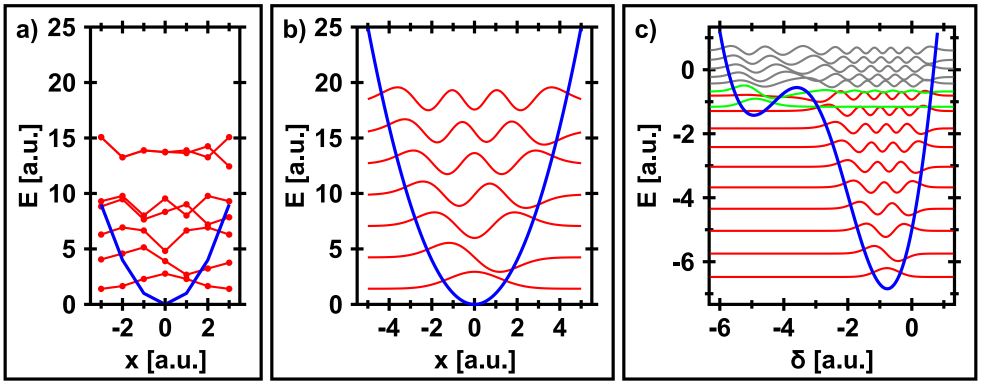

Figure 3.1: Examples of Numerical Simulations: Potential shown with eigenstates offset by their energy. – a) Eigenstates of coarse harmonic oscillator potential: $V(x) = x^2$, $-3.0 \leq x \leq 3.0$, $dx = 1.0$. b) Lowest 7 eigenstates of fine harmonic oscillator potential: $V(x) = x^2$, $-5.0 \leq x \leq 5.0$, $dx = 0.1$ c) Lowest 17 eigenstates of qubit-like potential: $V(\delta) = \delta^2 - 5\cos\delta + 5\delta$, $-8.0 \leq \delta \leq 3.0$, $d\delta = 0.05$. States localized in the shallow (deep) minimum are shown in green (red), while states that span both minima are shown in gray.

Plotting the eigenvectors offset by their corresponding energies gives a plot like Figure 3.1a. For the lower energy states this plot clearly shows the usual oscillating behavior of the wave functions and for the ground state even the exponential decay outside the potential. If the x-step-size is decreased from 1 to 0.1, i.e. the resolution of the approximation is increased by 10×, the first 7 eigenvectors look like Figure 3.1b. Their corresponding energies are:

$$1.58, 4.74, 7.89, 11.06, 14.23, 17.45, 20.80$$

These numbers show the expected equal spacing of the levels fairly well.

The energies of the levels are quite different in the two approximations. Especially for the higher levels, for which the wave functions should extend significantly past the chosen x-range, the approximation becomes fairly poor in the low-resolution case. But as the resolution of the approximation increases and the x-range is expanded, the energy levels and wave-functions get closer and closer to their true values. Unfortunately this also increases the size of the matrix $\mathbf{M}$ and therefore the time to diagonalize it. The latter increases exponentially with the size of $\mathbf{M}$. This makes finding the eigenstates of a quantum system from its potential a computationally hard problem. To ensure accurate conclusions, the approximation should be repeated with several different step sizes and ranges to verify that the energy levels have indeed converged to their true values.

3.2.1 The Eigenstates of the Qubit Potential

Applying this technique to the qubit potential is as straightforward and yields a plot similar to Figure 3.1c. The important pieces of information to take away from this analysis are:

- The number of states localized in the operating minimum as a function of flux bias: Even though the potential might have two or more minima, they might not be deep enough compared to the "mass" of the particle to support the required number (≥ 2) of localized quantum states.

- The energy difference between the ground and first excited state in the operating minimum as a function of flux bias: Dividing this number by $\hbar$ will give the expected (angular) operating frequency of the qubit, i.e. the frequency with which it needs to be driven to perform operations.

- The energy difference between the first and second excited levels in the operating minimum: The frequency corresponding to this transition will need to be significantly different from the operating frequency to allow for operations on the qubit without exciting it into unwanted higher levels.

- The number of states in the right minimum: During measurement, the first excited state in the operating minimum (here: left minimum) will be selectively tunneled into the right minimum. There, it will end up in a level of similar energy to the one it tunneled from, i.e. fairly high up in the minimum. The rate at which the state will decay in the right minimum, and thus the rate with which the measurement "latches" the outcome, is determined by the number of the level that the state tunnels into. Higher states decay faster with a rate of approximately T_1/n, where n is the level number. Fast decay is important to reduce the chance of the state tunneling back to the left before latching.

For the calculation to yield trustable results, a few things need to be kept in mind:

- The x-range over which the potential is approximated needs to be large enough for the wave-functions to "comfortably" go to zero on both sides.

- In a real qubit potential, the right minimum will most likely have hundreds of states at energies below the ground-state of the operating minimum. Since an $n \times n$ matrix will yield only the lowest n eigenstates, $\mathbf{M}$ therefore needs to have several hundred rows and columns.

- The ground-state in the operating minimum will usually not be the level with the lowest overall energy if the other minimum is deeper. In the figure shown, states 9 and 11 (counting from 0) are localized mostly in the left minimum, the states above level 11 span both minima, and all other states are localized in the right minimum. Thus, it is necessary to sort the levels into the correct minimum before subtracting their energies to find transition frequencies.

3.2.2 Eigenstates of Coupled Qubit Systems

It is theoretically possible to extend this method to finding the eigenstates of a system of coupled qubits. For this, the state of the second qubit is added to the Schrödinger equation, making it two-dimensional (a function of δ_1 and δ_2). $\psi_n^r(\delta_1, \delta_2)$ and $V(\delta_1, \delta_2)$ would then be rewritten as vectors following a convention like this:

$$\begin{aligned}\mathbf{V} \;=\; & (\,V(-1,-1), V(-1,0), V(-1,1), \\ & V(\;\;0,-1), V(\;\;0,0), V(\;\;0,1), \\ & V(\;\;1,-1), V(\;\;1,0), V(\;\;1,1)\,)\end{aligned} \tag{3.24}$$

The derivative operators are calculated accordingly and could look like this:

$$\frac{d^2}{d\delta_1^2} = \mathbf{D}_1 = \begin{bmatrix} -2 & & & 1 & & & & & \\ & -2 & & & 1 & & & & \\ & & -2 & & & 1 & & & \\ 1 & & & -2 & & & 1 & & \\ & 1 & & & -2 & & & 1 & \\ & & 1 & & & -2 & & & 1 \\ & & & 1 & & & -2 & & \\ & & & & 1 & & & -2 & \\ & & & & & 1 & & & -2 \end{bmatrix} \tag{3.25}$$

$$\frac{d^2}{d\delta_2^2} = \mathbf{D}_2 = \begin{bmatrix} -2 & 1 & & & & & & & \\ 1 & -2 & 1 & & & & & & \\ & 1 & -2 & & & & & & \\ & & & -2 & 1 & & & & \\ & & & 1 & -2 & 1 & & & \\ & & & & 1 & -2 & & & \\ & & & & & & -2 & 1 & \\ & & & & & & 1 & -2 & 1 \\ & & & & & & & 1 & -2 \end{bmatrix} \tag{3.26}$$

Unfortunately, going to a two-dimensional simulation squares the dimensions of all involved matrices. Since the one-dimensional simulation usually requires matrices of around $1,000 \times 1,000$ elements, the two-dimensional calculation will now have to find the eigenvalues of matrices with $1,000,000 \times 1,000,000$ elements.

This will most likely take forbiddingly long, making it necessary to find a different approach to understanding the dynamics.

3.3 Interaction with the Qubit

The qubit is controlled by varying its potential via changes $\Delta I_{dc}(t)$ in the current bias I_{dc}. As described above, the effect of this can be understood by splitting the resulting $\hat{H}(t)$ into:

$$\hat{H}(t) = \hat{H}_0 + \hat{V}(t) = \hat{H}_0 - \frac{\Phi_0}{2\pi}\,\Delta I_{dc}(t)\,\delta \tag{3.27}$$

This changes the level occupations according to:

$$i\hbar\,\partial_t\,a_n(t) = \sum_m a_m(t)\,e^{i(E_n - E_m)t/\hbar}\,\langle\, n\,|\,V(\delta,t)\,|\,m\,\rangle \tag{3.28}$$

where:

$$\begin{aligned}\langle\, n\,|\,V(\delta,t)\,|\,m\,\rangle &= \int \psi^r_n(\delta)^* V(\delta,t)\,\psi^r_m(\delta)\,d\delta \\ &= -\frac{\Phi_0}{2\pi}\,\Delta I_{dc}(t)\int \delta\,\psi^r_n(\delta)^*\psi^r_m(\delta)\,d\delta \\ &= \Delta I_{dc}(t)\,T_{nm} \end{aligned}\tag{3.29}$$

T_{nm} gives the strength of the coupling between levels n and m. If $T_{nm} = 0$ for certain n and m, no direct transitions can be driven between the levels.

Figure 3.2b shows $|T_{nm}|$ for the harmonic oscillator potential shown with its eigenstates in Figure 3.2a. This figure shows that a perturbation of the form $c(t)\,x$

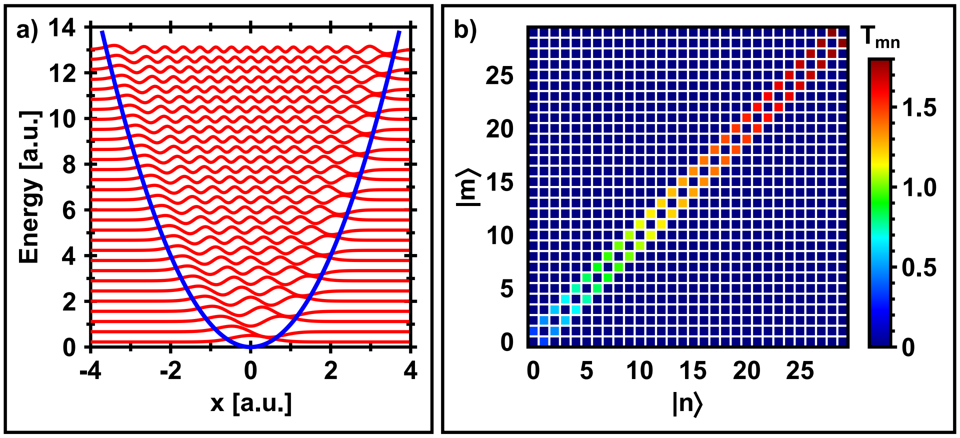

Figure 3.2: Simulation of LC Oscillator – a) Eigenstates: Lowest 30 eigenstates of harmonic oscillator potential: $V(x) = x^2$. b) Transition matrix: Absolute value of transition matrix elements $|T_{nm}|$ showing the expected $\sqrt{n}$ behavior.

can only drive transitions between neighboring levels (only elements on the first off-diagonals are non-zero). The fact that the diagonal elements of T_{nm} are zero implies that it is not possible to change the phase of a level population with this kind of a drive, making Z-rotations impossible (see Section 3.3.1). These results match the analytic solution:

$$T_{nm} = \sqrt{\frac{\hbar}{2\mu\omega}} \begin{pmatrix} & \sqrt{1} & & & \\ \sqrt{1} & & \sqrt{2} & & \\ & \sqrt{2} & & \sqrt{3} & \\ & & \sqrt{3} & & \ddots \\ & & & \ddots & \end{pmatrix} \tag{3.30}$$

Figure 3.3b shows $|T_{nm}|$ for the qubit-like potential shown with its eigenstates in Figure 3.3a. This plot consists of three distinct regions:

- The bottom-left corner corresponds to transitions between states confined

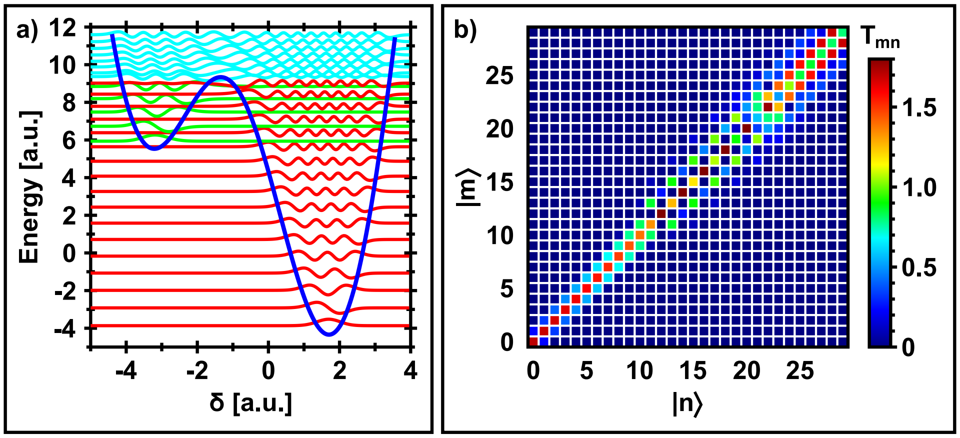

Figure 3.3: Simulation of Mock-Qubit – a) Eigenstates: Lowest 30 eigenstates of example qubit potential: $V(\delta) = \delta^2 - 8\,cos(\delta + 1)$. b) Transition matrix: Absolute value of transition matrix elements $|T_{nm}|$.

to the deep minimum. Just like in the harmonic oscillator case, transitions between non-neighboring states are very hard to achieve ($T_{nm} \approx 0$ for $n < m - 1$ or $n > m + 1$). In contrast to the harmonic oscillator, the diagonal elements here are not zero. This allows for phase changes, i.e. Z-rotations, of the states (see Section 3.3.1).

- The center region corresponds to transitions between states that are alternatingly confined to the shallow or the deep minimum. The fact that the states' confinement alternates perfectly between the minima is a result of the exact potential chosen and does not hold in general. In this region the dominating transitions are also between neighboring states in the same

Table 3.1: Transition Matrix Elements for Qubit-Like Potential

m	$T_{\|\,12\,\rangle,m}$	$T_{\|\,14\,\rangle,m}$	Left Index	Right Index
$\|\,0\,\rangle$	0.000	0.000		0
$\vdots$	$\vdots$	$\vdots$		$\vdots$
$\|\,11\,\rangle$	0.000	0.000		11
$\|\,12\,\rangle$	-3.180	-0.250	0	
$\|\,13\,\rangle$	0.000	0.000		12
$\|\,14\,\rangle$	-0.250	0.311	1	
$\|\,15\,\rangle$	0.000	0.000		13
$\|\,16\,\rangle$	-0.016	-0.359	2	
$\|\,17\,\rangle$	0.000	0.000		14
$\|\,18\,\rangle$	0.000	-0.032	3	
$\|\,19\,\rangle$	0.000	0.000		15
$\|\,20\,\rangle$	0.000	0.002	4	
$\|\,21\,\rangle$	0.000	0.000		16
$\vdots$	$\vdots$	$\vdots$		

minimum, i.e. next-to-nearest neighbors in energy.

- The top-right corner corresponds to transitions between states that span both minima. In the region where the barrier is disappearing (around level 20), the potential looks less and less like a harmonic oscillator. This makes transitions between non-neighboring states progressively easier.

Since the qubit will be formed by the lowest two energy levels in the shallow (left) minimum, the transitions of interest here are the ones between states

$|\,12\,\rangle$ and $|\,14\,\rangle$ (desired transitions) and transitions from states $|\,12\,\rangle$ or $|\,14\,\rangle$ to all other states (undesired transitions). The transition amplitudes T_{nm} are shown in Table 3.1. Luckily, it is already virtually impossible to accidentally drive transitions between the desired states and states in the deep minimum. The main concern therefore is to avoid driving unwanted transitions to states $|\,16\,\rangle$, $|\,18\,\rangle$, or $|\,20\,\rangle$. Specifically, the transition from level $|\,14\,\rangle$ to level $|\,16\,\rangle$ needs to be watched closely. To understand how this can be achieved, let's examine how a drive of the form

$$\Delta I_{dc}(t) = X \cos \omega t + Y \sin \omega t + Z \tag{3.31}$$

affects a specific transition between levels n and m, using:

$$i\hbar\, \partial_t\, a_n(t) = a_n(t)\, e^{i\frac{E_n - E_n}{\hbar}t}\, \Delta I_{dc}(t)\, T_{nn} + a_m(t)\, e^{i\frac{E_n - E_m}{\hbar}t}\, \Delta I_{dc}(t)\, T_{nm} \tag{3.32}$$

$$i\hbar\, \partial_t\, a_m(t) = a_n(t)\, e^{i\frac{E_m - E_n}{\hbar}t}\, \Delta I_{dc}(t)\, T_{mn} + a_m(t)\, e^{i\frac{E_m - E_m}{\hbar}t}\, \Delta I_{dc}(t)\, T_{mm} \tag{3.33}$$

By defining a vector $\mathbf{A}(t) = (a_n(t), a_m(t))$ this can be written in matrix form as:

$$\begin{aligned} i\hbar\, \partial_t\, \mathbf{A}(t) &= \Delta I_{dc}(t) \left[\begin{array}{cc} T_{nn} & e^{-i(E_m - E_n)t/\hbar}\, T_{nm} \\ e^{i(E_m - E_n)t/\hbar}\, T_{mn} & T_{mm} \end{array} \right] \mathbf{A}(t) \\ &= \Delta I_{dc}(t)\, \mathbf{M}\, \mathbf{A}(t) \end{aligned} \tag{3.34}$$

Since $T_{mn} = T_{nm}$, $\mathbf{M}$ can be expressed in terms of the Pauli Matrices:

$$\mathbf{M} = T_{mn}\, \cos \omega_{mn} t\, \sigma_{\mathbf{x}} + T_{mn}\, \sin \omega_{mn} t\, \sigma_{\mathbf{y}} + \frac{T_{nn} - T_{mm}}{2}\, \sigma_{\mathbf{z}} + \frac{T_{mm} + T_{nn}}{2}\, \mathbf{I} \tag{3.35}$$

with $\omega_{mn} = \frac{E_m - E_n}{\hbar}$. Multiplying this with the drive $\Delta I_{dc}(t)$ gives the following evolution:

$$
\begin{aligned}
&i\hbar\,\partial_t\,\mathbf{A}(t) = \\
&\quad\Bigg[T_{mn}\left(\frac{X}{2}\left(\cos\left(\omega_{mn}-\omega\right)t + \cos\left(\omega_{mn}+\omega\right)t\right) + \right. \\
&\qquad\qquad \left. + \frac{Y}{2}\left(\sin\left(\omega_{mn}+\omega\right)t - \sin\left(\omega_{mn}-\omega\right)t\right) + Z\cos\omega_{mn}t\right)\sigma_{\mathbf{x}} + \\
&\quad + T_{mn}\left(\frac{X}{2}\left(\sin\left(\omega_{mn}+\omega\right)t + \sin\left(\omega_{mn}-\omega\right)t\right) + \right. \\
&\qquad\qquad \left. + \frac{Y}{2}\left(\cos\left(\omega_{mn}-\omega\right)t - \cos\left(\omega_{mn}+\omega\right)t\right) + Z\sin\omega_{mn}t\right)\sigma_{\mathbf{y}} + \\
&\quad + \frac{T_{nn} - T_{mm}}{2}\left(X\cos\omega t + Y\sin\omega t + Z\right)\sigma_{\mathbf{z}} \\
&\quad + \frac{T_{mm} + T_{nn}}{2}\left(X\cos\omega t + Y\sin\omega t + Z\right)\mathbf{I}\,\Bigg]\;\mathbf{A}(t)
\end{aligned}
\tag{3.36}
$$

3.3.1 Interactions as Rotations on the Bloch Sphere

To understand the effect of this evolution, it is helpful to introduce a geometrical representation of the system of the two states of interest $|\,n\,\rangle$ and $|\,m\,\rangle$. The amplitudes $a_n(t)$ and $a_m(t)$ are complex numbers and thus described by two real numbers each. As an overall phase of the state is not physically detectable, $a_n(t)$ can be chosen to be real without loss of generality. Thus, the vector $\mathbf{A}(t)$ as defined above is described by three real numbers that can be interpreted as specifying a point in three dimensions. The normalization requirement $|a_n(t)|^2 + |a_m(t)|^2 = 1$ confines this point to the surface of the unit sphere centered at the origin called

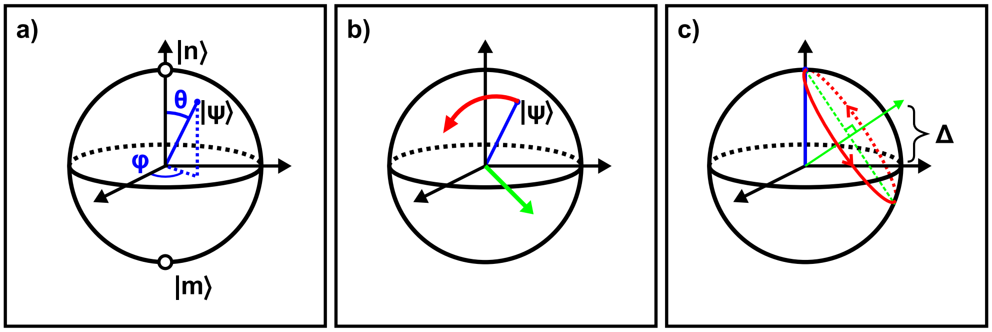

Figure 3.4: Bloch Sphere – a) Bloch sphere: Quantum states in two-level systems can be depicted as vectors on a sphere. b) Rotations: Operations on the system are visualized as rotations of the state vector around an axis defined by the operation. c) Off-resonant rotations: If the qubit is driven off resonance, the rotation vector points out of the X/Y-plane leading to rotations that can no longer cover great circles.

the "Bloch Sphere" after Felix Bloch. The states $|\,n\,\rangle$ and $|\,m\,\rangle$ are placed at the poles of the sphere and any arbitrary superposition of the two is depicted by a vector pointing to the surface of the sphere, called the "Bloch Vector". The spherical coordinates θ and φ that describe the direction of the Bloch Vector are related to the described state via:

$$e^{i\alpha}|\,\psi\,\rangle = \cos\frac{\theta}{2}|\,n\,\rangle + e^{i\varphi}\sin\frac{\theta}{2}|\,m\,\rangle \tag{3.37}$$

In this picture, the qubit interaction described above corresponds to a rotation of the Bloch Vector around an axis pointing in the direction given by the prefactors of $\sigma_{\mathbf{x}}$, $\sigma_{\mathbf{y}}$, and $\sigma_{\mathbf{z}}$. The sum of the squares of the prefactors is related to the rotation angle. Since the **I**-part of the interaction only influences the overall phase factor

α, it can be ignored.

Thus, the interaction can be decomposed into nine different simultaneously occurring rotations being applied to the state:

$$R_1 = \frac{T_{mn}}{2}\left(X\sigma_x + Y\sigma_y\right)\cos\left(\omega_{mn} - \omega\right)t \tag{3.38}$$

$$R_2 = \frac{T_{mn}}{2}\left(-Y\sigma_x + X\sigma_y\right)\sin\left(\omega_{mn} - \omega\right)t \tag{3.39}$$

$$R_3 = \frac{T_{mn}}{2}\left(X\sigma_x - Y\sigma_y\right)\cos\left(\omega_{mn} + \omega\right)t \tag{3.40}$$

$$R_4 = \frac{T_{mn}}{2}\left(Y\sigma_x + X\sigma_y\right)\sin\left(\omega_{mn} + \omega\right)t \tag{3.41}$$

$$R_5 = \frac{T_{nn} - T_{mm}}{2}X\sigma_z\,\cos\omega t \tag{3.42}$$

$$R_6 = \frac{T_{nn} - T_{mm}}{2}Y\sigma_z\,\sin\omega t \tag{3.43}$$

$$R_7 = T_{mn}Z\sigma_x\,\cos\omega_{mn}t \tag{3.44}$$

$$R_8 = T_{mn}Z\sigma_y\,\sin\omega_{mn}t \tag{3.45}$$

$$R_9 = \frac{T_{nn} - T_{mm}}{2}Z\sigma_z \tag{3.46}$$

In eight out of the nine cases, the direction of the rotation oscillates as a function of time. If the frequency of this oscillation is much faster than the rate of rotation, the net rotation of the vector will average to zero and the overall movement of the vector will remain at very small amplitudes. Therefore, if the drive strengths X, Y, and Z are kept at low enough values, R_3 through R_8 can be safely ignored. This step is called the "Rotating Wave Approximation". R_1 and R_2 can also be ignored if ω_{mn} is sufficiently different from ω.

Since R_9 does not change the magnitude of the level populations, but only their relative phases, a transition between levels can therefore only be driven if $\omega \approx \omega_{mn}$. Thanks to the non-linearity in the qubit potential, this makes it possible to operate the qubit solely in the lowest two states (the qubit states) of the shallow minimum by ensuring that the interaction does not drive transitions between one of the qubit states and undesired other states.

In our example case, if the qubit is initialized into state $|\,12\,\rangle$ and $\omega \approx \omega_{|\,12\,\rangle,|\,14\,\rangle}$, the only states that will ever be populated are states $|\,12\,\rangle$ and $|\,14\,\rangle$, which we will from now on call $|\,0\,\rangle$ and $|\,1\,\rangle$ to denote the two logical qubit states. The drive $\Delta I_{dc}(t)$ can then drive transitions between the qubit states via:

$$i\hbar\,\partial_t\,\mathbf{A}(t) = \left(\frac{T_{mn}}{2}\left(X(t)\sigma_x + Y(t)\sigma_y\right) + \frac{T_{nn} - T_{mm}}{2} Z(t)\sigma_z\right)\mathbf{A}(t) \tag{3.47}$$

3.3.2 Operations on a Single Qubit

The evolution of the qubit can be simulated numerically by approximating $\Delta I_{dc}(t)$ with sections $\Delta I_{dc}(t) \rightarrow \Delta I_{dc}(t + \Delta t)$ of temporarily constant drive amplitudes $X_{t\rightarrow t+\Delta t}$, $Y_{t\rightarrow t+\Delta t}$, and $Z_{t\rightarrow t+\Delta t}$. For each section, Equation 3.47 can then be solved exactly:

$$\mathbf{A}(t+\Delta t) = e^{-i(T_{mn}\,X_{t\to t+\Delta t}\,\sigma_x + T_{mn}\,Y_{t\to t+\Delta t}\,\sigma_y + (T_{nn}-T_{mm})\,Z_{t\to t+\Delta t}\,\sigma_z)\Delta t/2\hbar}\mathbf{A}(t) \quad (3.48)$$

Since the transition matrix elements T_{mn} are usually not known exactly and will be calibrated away in the experiment, it is more useful to express the drive strengths X, Y, and Z in terms of the frequency of rotation they cause. This equation then becomes:

$$\mathbf{A}(t+\Delta t) = e^{-i\pi\Delta t\,(X_{t\to t+\Delta t}\,\sigma_x + Y_{t\to t+\Delta t}\,\sigma_y + Z_{t\to t+\Delta t}\,\sigma_z)}\mathbf{A}(t) \quad (3.49)$$

Since the $Z_{t\to t+\Delta t}\,\sigma_z$ term corresponds to a rotation of the state around the Z-axis, it can be used to emulate an off-resonant drive. For example, a drive causing a rotation at a rate of 20 MHz around the Y-axis that is detuned from the qubit by 5 MHz is simulated as:

$$\mathbf{A}(t+\Delta t) = e^{-i\pi\Delta t\,(20\,\mathrm{MHz}\,\sigma_y + 5\,\mathrm{MHz}\,\sigma_z)}\mathbf{A}(t) \quad (3.50)$$

Effectively, this leads to a final rotation around an axis that is tilted out of the X/Y-plane of the Bloch sphere. The vector therefore no longer traces out the great-circle through the $|\,1\,\rangle$-state, but instead rotates faster with less amplitude as shown in Figure 3.4c. This picture also visualizes nicely why a drive that is far off resonance with a given transition can safely be ignored, as it does not significantly move the state away from the pole.

3.3.3 Single Qubit Operations in a Coupled System

To be able to simulate multi-qubit systems, the vector description $\mathbf{A}(t)$ needs to be expanded to include all possible qubit states. For n qubits this vector has 2^n entries to accommodate all binary combinations of the possible measurement outcomes. For two qubits, for example, it would take the form:

$$\mathbf{A}(t) = \big(a_{|\,00\,\rangle}(t), a_{|\,01\,\rangle}(t), a_{|\,10\,\rangle}(t), a_{|\,11\,\rangle}(t)\big) \tag{3.51}$$

Here, $a_{|\,xy\,\rangle}$ is the complex amplitude of state $|\,xy\,\rangle$. Again, the overall phase of the state is arbitrary, allowing us to choose $a_{|\,00\,\rangle}$ to be real. The normalization requirement now applies to the entire state in the form:

$$\big|a_{|\,00\,\rangle}\big|^2 + \big|a_{|\,01\,\rangle}\big|^2 + \big|a_{|\,10\,\rangle}\big|^2 + \big|a_{|\,11\,\rangle}\big|^2 = 1 \tag{3.52}$$

This gives the qubit state six degrees of freedom. In general, an n-qubit state has $2^{n+1} - 2$ degrees of freedom. Since each degree of freedom can be used as a register in a calculation, this leads to an exponential increase in the power of a quantum computer with its number of bits. Unfortunately, this also makes it exponentially harder to simulate. To simulate single qubit operations in an n-qubit system, it is necessary to expand the Pauli matrices σ_x, σ_y, and σ_z to apply to only one qubit in the set. This is done by forming the Kronecker product of these matrices with the identity. For example, an X-rotation on the second of four qubits would be simulated using:

$$\mathbf{A}(t + \Delta t) = e^{-i\pi\Delta t\,(X_2\,\mathbf{I}\otimes\sigma_x\otimes\mathbf{I}\otimes\mathbf{I})}\mathbf{A}(t) \tag{3.53}$$

Simultaneous X, Z, and Y-rotations on three qubits respectively would look like:

$$\mathbf{A}(t + \Delta t) = e^{-i\pi\Delta t\,(X_1\,\sigma_x\otimes\mathbf{I}\otimes\mathbf{I}+Z_2\,\mathbf{I}\otimes\sigma_z\otimes\mathbf{I}+Y_3\,\mathbf{I}\otimes\mathbf{I}\otimes\sigma_y)}\mathbf{A}(t) \tag{3.54}$$

3.3.4 Qubit Coupling

As mentioned in the discussion of the DiVincenzo criteria in Chapter 1.3.4, it is sufficient for universal quantum computation to implement a gate that acts on two qubits at a time. For the phase qubit, a sufficient gate can be easily constructed via a capacitive coupling between the qubits. This gate's natural evolution leads to the so-called i-Swap operation, which swaps the $|\,01\,\rangle$ with the $|\,10\,\rangle$ state, applying a phase-shift in the process. The relevant matrix needed to simulate the interaction is:

$$\mathbf{C} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \tag{3.55}$$

If the coupling strength C_{12} is defined in terms of the resulting swap frequency, the evolution becomes:

$$\mathbf{A}(t + \Delta t) = e^{-i\pi\Delta t\,(C_{12}\,\mathbf{C})}\mathbf{A}(t) \tag{3.56}$$

To couple qubits 2 and 3 out of five, one would use:

$$\mathbf{A}(t+\Delta t) = e^{-i\pi\Delta t\,(C_{23}\;\mathbf{I}\otimes\mathbf{C}\otimes\mathbf{I}\otimes\mathbf{I})}\mathbf{A}(t) \tag{3.57}$$

To combine this operation with a simultaneous X-rotation on qubit 1, one would use:

$$\mathbf{A}(t+\Delta t) = e^{-i\pi\Delta t\,(X_1\,\sigma_x\otimes\mathbf{I}\otimes\mathbf{I}\otimes\mathbf{I}\otimes\mathbf{I}+C_{23}\;\mathbf{I}\otimes\mathbf{C}\otimes\mathbf{I}\otimes\mathbf{I})}\mathbf{A}(t) \tag{3.58}$$

If, instead, the coupling is followed by the X-rotation on qubit 1, the evolution is:

$$\mathbf{A}(t+\Delta t_C+\Delta t_X) = e^{-i\pi\Delta t_X\,(X_1\,\sigma_x\otimes\mathbf{I}\otimes\mathbf{I}\otimes\mathbf{I}\otimes\mathbf{I})}\,e^{-i\pi\Delta t_C\,(C_{23}\;\mathbf{I}\otimes\mathbf{C}\otimes\mathbf{I}\otimes\mathbf{I})}\mathbf{A}(t) \tag{3.59}$$

3.4 Simulating Imperfections

So far, apart from simulating an off-resonant drive via an additional Z-rotation, all operations of the qubits have been assumed ideal. Unfortunately, in reality, this is not the case, and to obtain useful predictions of experimental data, several imperfections need to be taken into account.

3.4.1 Measurement Fidelities

Ideally, the probability of finding the qubit in one of the given states would be calculated from:

$$P_{|\,0\,\rangle}(t) = |a_0(t)|^2 \text{ and } P_{|\,1\,\rangle}(t) = |a_1(t)|^2 = 1 - P_{|\,0\,\rangle}(t) \tag{3.60}$$

This calculation assumes that the states can be identified correctly with 100% certainty. If this is not the case, one can simulate mistakes by defining classical error probabilities $E_{|\,0\,\rangle\rightarrow|\,1\,\rangle}$ and $E_{|\,1\,\rangle\rightarrow|\,0\,\rangle}$ which capture the probability of misidentifying a $|\,0\,\rangle$ as a $|\,1\,\rangle$ or vice versa. The measured probabilities then become:

$$P'_{|\,0\,\rangle}(t) = \left(1 - E_{|\,0\,\rangle\rightarrow|\,1\,\rangle}\right) |a_0(t)|^2 + E_{|\,1\,\rangle\rightarrow|\,0\,\rangle}\, |a_1(t)|^2 \tag{3.61}$$

$$P'_{|\,1\,\rangle}(t) = \left(1 - E_{|\,1\,\rangle\rightarrow|\,0\,\rangle}\right) |a_1(t)|^2 + E_{|\,0\,\rangle\rightarrow|\,1\,\rangle}\, |a_0(t)|^2 = 1 - P'_{|\,0\,\rangle}(t) \tag{3.62}$$

For multi-qubit simulations, this needs to be extended accordingly.

3.4.2 Measurement Crosstalk

In addition, in multi-qubit systems, it is possible that the tunneling of one qubit influences the tunneling of other qubits [McDermott et al., 2005]. Specifically, when one qubit tunnels to the deeper minimum and subsequently decays to the ground-state in that minimum, it will radiate energy as it transitions to lower energy levels. This radiation can couple into the other qubits in the circuit and drive a transition to the excited state, which leads to an undesired tunneling of those qubits as well. This form of crosstalk is a classically probabilistic event and can thus be simulated using the same approach as used for the measurement fidelities, e.g.:

$$P''_{|\,00\,\rangle}(t) = P'_{|\,00\,\rangle}(t) \tag{3.63}$$

$$P''_{|01\rangle}(t) = (1 - X_{21})P'_{|01\rangle}(t) \tag{3.64}$$

$$P''_{|10\rangle}(t) = (1 - X_{12})P'_{|10\rangle}(t) \tag{3.65}$$

$$P''_{|11\rangle}(t) = P'_{|11\rangle}(t) + X_{21}P'_{|01\rangle}(t) + X_{12}P'_{|10\rangle}(t) \tag{3.66}$$

Here, X_{21} (X_{12}) capture the classical probability that a tunneling of qubit 2 (1) causes a tunneling of qubit 1 (2).

3.4.3 Microwave Crosstalk

Another source of errors in an actual qubit experiment is the commonly insufficient electrical isolation between the microwave drives of the different qubits. Due to the high frequency of the driving field and the close proximity of the qubits it is fairly difficult to ensure that no photons leak from one drive line to another qubit. Even though this effect is usually fairly small ($-20\,\mathrm{dB}$) and can be compensated for by sending a correction pulse to all other qubits, it can be easily simulated if needed by applying a simultaneous rotation to the other qubits with a proportionally smaller amplitude and a potential phase-shift.

3.4.4 Decoherence – The Density Matrix Formalism

The final error source that needs to be included in the simulation is the one specifically highlighted in the DiVincenzo criteria: Decoherence.

Decoherence consists of two parts: Energy relaxation and dephasing. These affect the two degrees of freedom of the qubit state in different ways.

Energy relaxation describes the process by which the qubit loses energy and decays back to its $|\,0\,\rangle$-state. This decay primarily affects the θ degree of freedom of the state. It is caused by undesired coupling between the qubit and the environment which provides a path for the qubit to dissipate its energy.

Dephasing captures a loss of information stored in the φ degree of freedom of the state. This is usually caused by magnetic flux noise that effectively applies random Z-rotations to the qubit's state. Other noise sources, like critical current noise in the Josephson junction, can lead to the same effect, but seem to be less important [Bialczak et al., 2007]. The two decoherence mechanisms each have a timescale associated with them. T_1 captures the rate at which the qubit relaxes back into the $|\,0\,\rangle$-state and T_φ captures the rate at which the qubit's phase information is randomized. Since the decay to the $|\,0\,\rangle$-state also causes a loss in phase information, a new timescale T_2 is commonly defined to replace the less physical quantity T_φ:

$$\frac{1}{T_2} = \frac{1}{2T_1} + \frac{1}{T_\varphi} \tag{3.67}$$

Both T_1 and T_φ are the result of classically random processes. Therefore, they do not affect the quantum state in a coherent way. Dephasing, for example, cannot be adequately described by a rotation of the state vector on the Bloch sphere. Instead, it corresponds to this vector shrinking towards the Z-axis. The above described simulation formalism can therefore not be used in its presented form to capture these processes as the state's normalization requirement $|a_n(t)|^2 + |a_m(t)|^2 = 1$ forces the state vector to remain on the surface of the Bloch sphere.

To allow for the extra degree of freedom needed, it is necessary to move to the "density formalism" for describing the qubit state. This formalism adds a degree of freedom to the state by describing it as a probabilistic ensemble of pure states. Each possible pure state $|\, A \,\rangle$ is assigned a weight w_A describing the classical probability with which a randomly drawn member of the ensemble is in the state $|\, A \,\rangle$. The weights w_A fulfill:

$$\sum_A w_A = 1 \tag{3.68}$$

In this formalism, the state is described by a "density matrix" ρ, which takes the form:

$$\rho = \sum_A w_A |\, A \,\rangle\langle\, A \,| = \sum_n w_n |\, n \,\rangle\langle\, n \,| \tag{3.69}$$

To simulate the evolution of the state, one simply needs to evolve both factors of the outer product $|\, A \,\rangle\langle\, A \,|$:

$$\begin{aligned} \rho(t + \Delta t) &= \sum_A w_A |\, A(t + \Delta t) \,\rangle\langle\, A(t + \Delta t) \,| \\ &= \sum_A w_A e^{-i\pi\Delta t(\dots)} \,|\, A(t) \,\rangle\langle\, A(t) \,|\, e^{i\pi\Delta t(\dots)} \\ &= e^{-i\pi\Delta t(\dots)} \,\rho\, e^{i\pi\Delta t(\dots)} \end{aligned} \tag{3.70}$$

Note the difference in signs of the two exponents. For example, an X-rotation applied to the density matrix would be:

$$\rho(t+\Delta t) = e^{-i\pi\Delta t(X\,\sigma_x)}\,\rho\,e^{i\pi\Delta t(X\,\sigma_x)} \tag{3.71}$$

The density matrix of an n-qubit state is a hermitian $2^n \times 2^n$ matrix. Its diagonal elements give the probabilities of the possible measurement outcomes, e.g.:

$$\rho = \begin{bmatrix} P_{|\,00\,\rangle} & a & b & c \\ a^* & P_{|\,01\,\rangle} & d & e \\ b^* & d^* & P_{|\,10\,\rangle} & f \\ c^* & e^* & f^* & P_{|\,11\,\rangle} \end{bmatrix} \tag{3.72}$$

The complex off-diagonal elements capture coherences between the states. Since the probabilities are real and need to sum to unity, i.e. $\mathrm{Tr}(\rho) = 1$, the number of degrees of freedom of the density matrix for an n-qubit system is given by:

$$D(n) = n^2 - 1 \tag{3.73}$$

3.4.5 Decoherence – The Kraus Operators

One way to simulate the decoherence of a qubit state expressed in the density matrix formalism is with the use of the Kraus Operators [Kraus, 1983]:

$$\mathbf{K}_{1a}(\Delta t) = \begin{bmatrix} 1 & 0 \\ 0 & e^{-\Delta t/2T_1} \end{bmatrix} \tag{3.74}$$

$$\mathbf{K}_{1b}(\Delta t) = \begin{bmatrix} 0 & \sqrt{1 - e^{-\Delta t/T_1}} \\ 0 & 0 \end{bmatrix} \tag{3.75}$$

$$\mathbf{K}_{\varphi a}(\Delta t) = \begin{bmatrix} 1 & 0 \\ 0 & e^{-\Delta t/2T_\varphi} \end{bmatrix} \tag{3.76}$$

$$\mathbf{K}_{\varphi b}(\Delta t) = \begin{bmatrix} 0 & 0 \\ 0 & \sqrt{1 - e^{-\Delta t/T_\varphi}} \end{bmatrix} \tag{3.77}$$

$\mathbf{K}_{1a}$ and $\mathbf{K}_{1b}$ capture the energy relaxation process, while $\mathbf{K}_{\varphi a}$ and $\mathbf{K}_{\varphi b}$ capture dephasing. Since both energy relaxation and dephasing are "non-unitary" process, they need to be broken up mathematically into two steps, yielding the following decoherence operations:

$$\rho'(t+\Delta t) = \mathbf{K}_{1a}(\Delta t)\,\rho(t+\Delta t)\mathbf{K}^\dagger_{1a}(\Delta t) + \mathbf{K}_{1b}(\Delta t)\,\rho(t+\Delta t)\mathbf{K}^\dagger_{1b}(\Delta t) \tag{3.78}$$

$$\rho''(t+\Delta t) = \mathbf{K}_{\varphi a}(\Delta t)\,\rho'(t+\Delta t)\mathbf{K}^\dagger_{\varphi a}(\Delta t) + \mathbf{K}_{\varphi b}(\Delta t)\,\rho'(t+\Delta t)\mathbf{K}^\dagger_{\varphi b}(\Delta t) \tag{3.79}$$

Intuitively, $\mathbf{K}_{1b}$ transfers some of the population from the excited state into the ground state while $\mathbf{K}_{1a}$ ensures that the resulting state is still normalized. $\mathbf{K}_{\varphi b}$ reduces the phase information of the state while $\mathbf{K}_{\varphi a}$, again, ensures normalization. For accurate results, the decay operation needs to be interleaved into the stream of rotation and coupling operations fairly frequently. Δt should be chosen to be much smaller than any timescales of rotation or coupling, i.e.:

$$\Delta t \ll \min \sqrt{\frac{1}{X_n^2 + Y_n^2 + Z_n^2}} \text{ and } \Delta t \ll \min \frac{1}{C_{nm}} \tag{3.80}$$

Chapter 4

Designing the Phase Qubit Integrated Circuit

The design of the actual phase qubit device consists of three steps:

- First, the electrical parameters of the circuit need to be chosen based on experimental limitations and the understanding gained from the simulations explained in the previous chapter.

- Second, these electrical parameters need to be implemented by defining the physical geometry of the structures that will make up the different circuit elements.

- Lastly, the right materials and processes need to be chosen or developed to

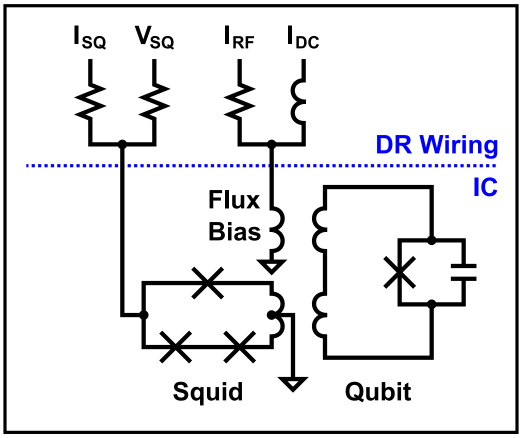

Figure 4.1: Qubit Circuit: The qubit is controlled via a flux bias line that sends RF and DC bias pulses to the qubit. It is read out via a squid.

build the structures in a way that minimizes imperfections.

4.1 Electrical Circuit Design

The simplest electrical circuit that can implement a viable phase qubit consists of three components:

- The qubit itself is made up of an inductive loop containing a Josephson junction with its parallel shunting capacitance.
- The qubit is controlled by a bias line that terminates in a single bias inductor which connects to the qubit via a mutual inductance.
- The qubit is read out via a three-junction squid to which it is coupled by a

mutual inductance between the qubit loop and the squid loop.

In addition, the design needs to include coupling circuitry that allows two or more qubits to interact.

4.1.1 Qubit Circuit Parameters

The exact electrical values of the qubit's inductance, capacitance, and junction critical current are only very loosely specified. It is possible to build viable qubits with a wide array of parameter combinations. But most choices involve a trade-off of different performance parameters or ease of implementation. The simplest approach is to use the eigenstate simulation outlined in the previous chapter to design a qubit circuit that addresses the following concerns:

The easiest requirement to pin down is the frequency at which the qubit will operate, i.e the energy difference of the lowest two states in the operating minimum. To prevent thermal excitations from destroying the qubit state, the operating frequency will need to be chosen to fulfill:

$$E_{thermal} = k\,T \ll \hbar\,\omega \qquad (4.1)$$

Since readily available cryogenic technology can cool devices to around 25 mK, this requires the qubit's operating frequency to lie significantly above 2 GHz. Current wireless standards use frequencies around 2-3 GHz, which makes it easy to obtain control electronics that can handle signals up to 10-15 GHz. Thus, a frequency choice of 5-10 GHz makes good sense.

For maximum flexibility, the signals that control the qubit will need to be generated digitally. Since current high-speed digital electronics operates at frequencies around 1 GHz, the pulses used to control the qubit should be on the order of a few ns long. Given the best-case frequency spread of pulses of that length (Slepian pulses), the non-linearity (i.e. difference in energy spacing between the lowest three eigenstates) of the qubit should be at least 150-200 MHz to allow for quick transitions (high amplitude pulses) that don't drive undesired transitions.

The phase qubit needs at least two stable minima in its potential to allow for the selective tunneling and decay of the $|\,1\,\rangle$ state into a state that is detectably different from the $|\,0\,\rangle$ state. This existence of multiple minima adds a slight challenge to the qubit reset process as a random initial state might decay into any one of the potential's minima. It is possible to design the qubit such that it can be biased to have either one or two stable minima. Such a qubit is the easiest to reset. It is also possible to reset qubits with three or more stable minima at a time by using a back-and-forth-tilting bias that destabilizes all minima except for the desired one. But as the number of minima grows, this probabilistic reset process takes longer and longer. Therefore, potentials with maximally two or three minima are the most useful.

During the measurement, the $|1\rangle$ state is tunneled out of the operating minimum into a neighboring minimum. After the tunneling, the state needs to decay in the neighboring minimum to "latch" the measurement before it has time to tunnel back to the operating minimum. The energy relaxation time scales roughly as $1/n$, where n is the number of the starting level. Therefore it is desirable to have as many states as possible in the neighboring minimum at the time of tunneling. This number should be in the several hundred states and needs to be increased as T_1 is improved.

The qubit will most likely be subject to $1/f$ flux noise, the origin of which is still being investigated [Sendelbach et al., 2008]. This flux noise will be converted to a bias noise by the qubit's inductor leading to dephasing of the qubit's state. A larger inductance will reduce this effect and is therefore preferred.

These considerations taken together can lead to critical current values around $2\,\mu$A, inductances of around 720 pH and capacitances of around 1 pF, but none of these values need to be hit exactly.

4.1.2 Biasing Circuit Parameters

The biasing circuit consists of three parts: The inductive coil in the phase qubit integrated circuit, the filtering and wiring inside the dilution refrigerator, and the electronics that will generate the required biasing sequences. This chapter will focus only on the integrated circuit, while the other two components will be discussed later.

The electrical characteristics of the bias coil are fully described by three parameters: The coil's inductance, its mutual inductance with the qubit loop and its mutual inductance with the squid loop.

Since the squid readout should be influenced as little as possible by the qubit bias, the last value is easy to pick: The mutual inductance between the bias coil and the squid loop should be as close to zero as possible.

The inductance of the bias coil and the mutual inductance between the qubit loop and the bias coil are chosen based on two concerns: On the one hand, the inductances need to be large enough to allow a reasonable current ($< 2\,\mathrm{mA}$) in the bias inductor to cause a sufficiently large tilt of the qubit potential to implement the reset. For this, it is sufficient if the bias coil can apply about two flux quanta (Φ_0) to the qubit loop. On the other hand, the bias line will be subject to electrical noise generated in the elements at higher temperature stages of the dilution refrigerator. Furthermore, the bias line is terminated with the usual $50\,\Omega$ impedance, which allows it to dissipate energy. To reduce the amount of noise coupled into the qubit and energy dissipated by the bias line, the mutual inductance should be kept as small as possible. A ratio of the inductance of the qubit to the mutual inductance to the flux bias line of $\frac{L_{qubit}}{M_{qubit/flux\,bias}} \sim 100$ yields the right impedance transformation to allow for long energy relaxation times T_1 while still allowing for a sufficiently strong bias.

Given these two restrictions, values of 180 pH for the coils inductance and 2 pH for the mutual inductance between the flux coil and the qubit loop seem to work well.

4.1.3 Readout Squid Parameters

It turns out that the design of the readout squid is actually one of the most interesting and complex parts of the circuit design process. This is due to the fact that the squid needs to be strongly coupled to the qubit during the readout process but preferably uncoupled during the qubit operation.

The readout squid consists of an inductive loop containing three Josephson junctions. These junctions are arranged as shown in Figure 4.2a. The loop shares a mutual inductance with the qubit loop. This allows the qubit to apply a state dependent current bias I_Q to the squid loop. In the branch of the squid containing one junction, this current bias adds to any externally applied current bias I_{bias} while in the other branch it subtracts. This changes the external bias needed to exceed the critical current of the squid's junctions at which point the squid switches to the voltage state and thus generates a measurable signal. Just like for a single junction, this voltage signal is large enough to create quasi-particle

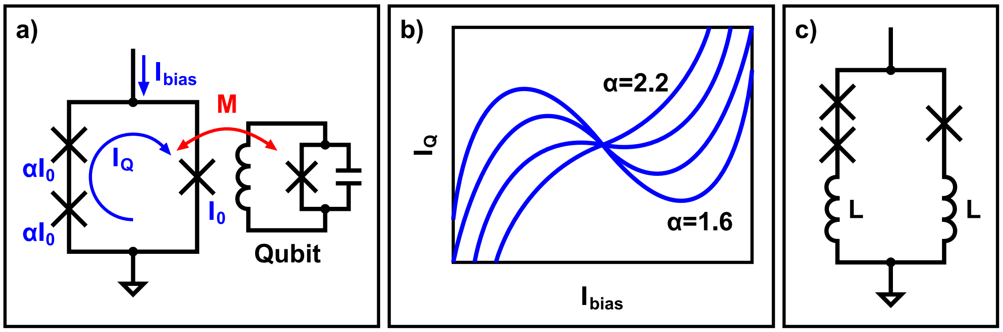

Figure 4.2: Readout Squid – a) Circuit: The squid consists of three junctions, two of which have a critical current that is $\alpha\times$ bigger than the third. The qubit state creates a loop current I_Q inside the squid via the mutual inductance M. circuit is placed inside a cutout in the IC's ground-plane. b) Dissipation coupling: The loop current I_Q creates changes in the bias current I_{bias} via the Josephson Effect in the three junctions. This causes dissipation in the qubit unless the squid is biased to one of the insensitive points where $\frac{dI_Q}{dI_{bias}} = 0$. c) Complete circuit: To create the mutual inductance, the squid loop needs to also contain inductors as indicated.

excitations. Since these are a potential source of decoherence for the qubit, the squid's voltage jump needs to be kept low. This is achieved by externally shunting the squid with a resistor on the order of $30\,\Omega$. This resistor does not need to be placed on the phase qubit integrated circuit chip, but can instead be placed outside the sample mounting box.

The three-junction design was chosen as it allows for the squid to be decoupled from the qubit when it is not needed for the readout [Neeley et al., 2008b]. This can be understood by examining the circuit shown in Figure 4.2a. The squid is externally biased with a current I_{bias}. In addition, the qubit causes a current I_Q to flow in the squid's loop. These currents are related via the phases across the Josephson junctions as given by the Josephson relations:

$$I_{bias} = I_0 \sin\delta + \alpha I_0 \sin\frac{\delta}{2} \tag{4.2}$$

$$I_Q = I_0 \sin\delta - \alpha I_0 \sin\frac{\delta}{2} \tag{4.3}$$

These equations assume that the two junctions αI_0 are identical and therefore show the same phase difference across them because of symmetry. Figure 4.2b, a plot of I_{bias} versus I_Q, shows how a change in the qubit's loop flux translates into a change in the squid bias current. If the squid is biased such that the slope dI_Q/dI_{bias} is non-zero, the qubit can couple to the squid's bias line and drive current through the above mentioned shunting resistor. This opens a path for the qubit to dissipate energy and thus needs to be avoided. As can be seen in Figure 4.2b, for values of α smaller than 2.0 the squid can be biased with a current I^*_{bias} at which the qubit does not couple to the squid's bias line to first order. To ensure the existence of such an insensitive point despite slight variations in the critical currents during fabrication, a value of $\alpha = 1.7$ makes good sense.

Another electrical characteristic of the squid that needs to be chosen is the inductance L of the squid loop as shown in Figure 4.2c, which is used to generate the mutual inductance with the qubit. The important concern here is the ratio of this inductance to the inductance of the squid's Josephson junctions at zero bias. This ratio is given by:

$$\beta = 2\pi \, L \, I_0/\Phi_0 \tag{4.4}$$

This ratio is chosen to balance two counter-acting effects: On the one hand, a larger loop inductance of the squid allows for a larger mutual inductance with the qubit and thus increases the coupling during the readout. On the other hand, a larger inductance will drop more of the phase difference δ across the squid loop, reducing the exposure of the junctions to the bias. A value of β around 0.9 seems to balance these concerns well.

The exact critical current of the squid is not quite as important, so, for fabrication convenience, one of the junctions, e.g. the single one, can be chosen to match the qubit's critical current around $2\,\mu$A. The other two junctions then need to be

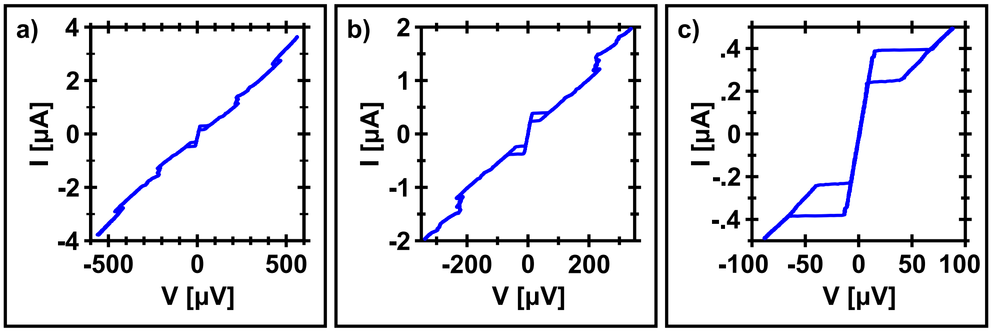

Figure 4.3: Squid I/V Traces – a) Large bias range: Due to the many junctions in the squid, the trace shows a lot of structure even out to high bias values. b) Medium bias range: The shunting resistor prevents the squid from switching all the way to the gap to reduce quasi particle generation. The gap at 2Δ is still visible as a step (around $\pm 200\,\mu$V). c) Small bias range: The squid's switching response is hysteretic.

designed with a slightly larger area to yield the required higher critical current of $3.4\,\mu$A given the same oxide barrier thickness.

The shunting resistor across the squid will change the shape of the squid's current voltage response to resemble Figure 4.3. The trace shows the same supercurrent branch as a single junction, but its switching behavior and subgap voltage are much less pronounced due to the shunt. The step at twice the superconducting gap is still visible in the plot, but its height is significantly reduced.

4.1.4 Coupler Circuit

The phase qubit's low impedance provides a large amount of flexibility when it comes to the choice and design of circuitry used to couple multiple qubits

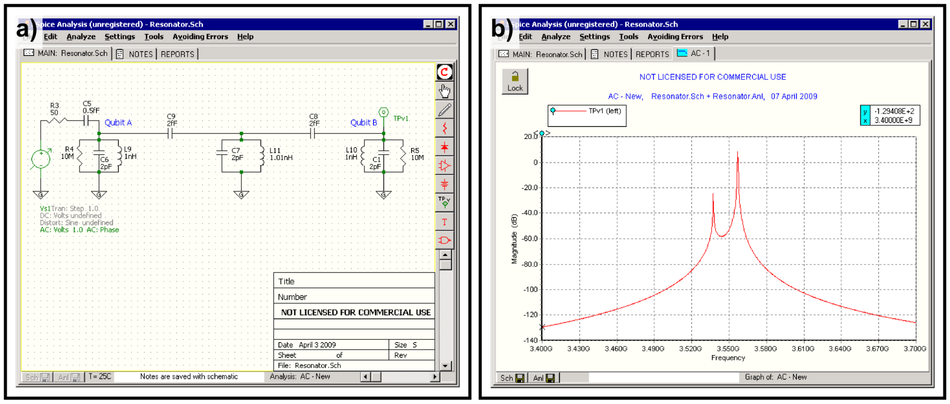

Figure 4.4: Spice Coupler Design – a) Schematic: The qubit is emulated by a damped LC oscillator. One qubit is driven with a microwave source and the response of the second qubit is analyzed. b) Analysis: The resulting response curve shows a splitting that is equal to the coupling strength.

to each other. The simplest way to design a coupling element is via the use of circuit modeling software like SPICE. For this purpose, the qubit can be emulated simply via a parallel RLC oscillator with its electrical values chosen to adjust its resonance frequency to the qubit's operating frequency. One of the qubits can then be driven with an AC voltage and the response of the second qubit to this bias can be measured. Figure 4.4b shows the frequency response of the second qubit to a drive on the first. In this case, the two qubits are coupled through an LC oscillator as shown in Figure 4.4a. This trace shows a response peak that is split by about 20 MHz. This splitting gives a fairly good estimate of the coupling strength that this element will yield in the final qubit circuit.

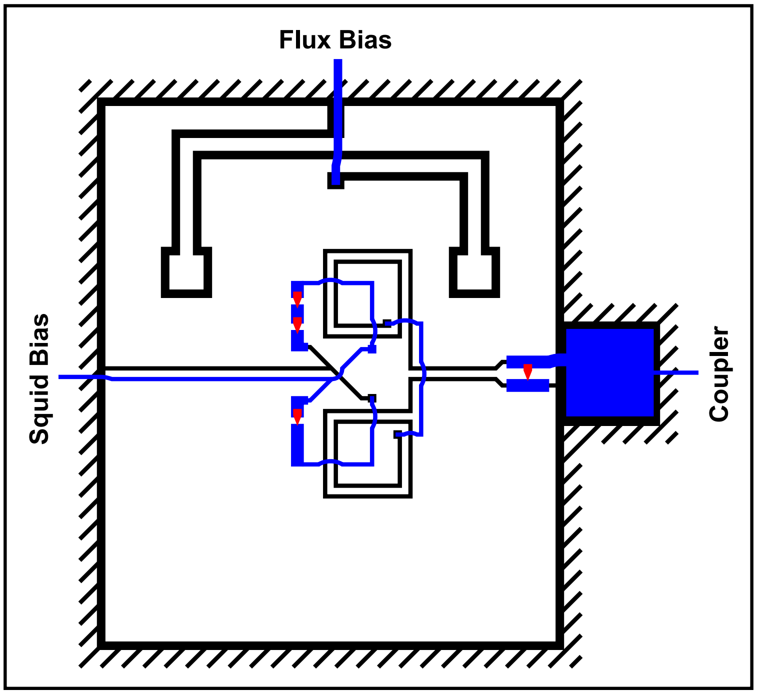

Figure 4.5: Qubit Integrated Circuit: The qubit circuit is placed inside a cutout in the IC's ground-plane. The geometrical arrangement of the qubit, squid, and flux bias coils determine their mutual inductances. The layout has three terminals to connect the squid bias, flux bias, and qubit coupler.

4.2 Geometric Circuit Element Layout

Now that all electrical design values are chosen, the next step is to lay out how the elements will be implemented. For almost all elements, their electrical characteristics are primarily determined by their geometric shape.

4.2.1 Qubit

When designing the inductor of the qubit it is important to reduce the circuit's sensitivity to any potentially fluctuating external background magnetic fields. This can be achieved to first order by arranging the inductor into a symmetric figure-8 configuration. With this, any current induced in one of the loops by a background field is exactly cancelled by the current induced in the other loop. This makes the qubit sensitive only to gradients in magnetic fields and is therefore also called a gradiometer design. The actual shape of the inductor is best designed with modeling software. A very powerful free tool that serves this purpose well is FastHenry. It allows for the specification of traces of given dimensions and will then calculate the resulting inductance of all connected traces and all mutual inductances between different sets of connected traces. Again, the design allows for a lot of flexibility in the choices of exact parameters but there are a few concerns to keep in mind. The width of traces used in the design should be large enough to yield reproducible results during fabrication, but not too large to avoid trapping magnetic flux vortices. A good size here seems $2\,\mu$m. The number of turns in the inductor needs to be balanced between the overall size of the structure and the added capacitance due to the needed crossovers. Two turns here seem to be a good number.

The geometry of the qubit junction is a lot more strictly defined. It needs to be as small as possible since even a single materials defect in the junction couples strongly to the qubit and thus needs to be avoided. On the other hand it cannot be so small as to not yield reliable fabrication results. Also, since the junction's oxide thickness is somewhat irreproducible, it is useful to generate an array of junctions on the wafer with slightly different areas to guarantee that some dies on the wafer will yield the desired critical current. A design with $2\,\mu\mathrm{m}^2$ wedge-shaped junctions, all oriented in the same direction so that they can be pass-shifted together, works well.

The requirement that the junction needs to be as small as possible does not allow for its capacitance to be large enough to reach the needed value. This can be easily remedied with an external shunting capacitor. This capacitor can be implemented with a trivial parallel-plate design. Its geometry is chosen using the relations:

$$C = C_{ext} + C_{junc} \tag{4.5}$$

$$C_{ext} = \frac{\varepsilon\, A}{d} \tag{4.6}$$

When choosing A versus d, the only concerns are the reliability of fabrication and the size of the final structure.

4.2.2 Bias Coil

The design of the bias coil can be laid out using FastHenry as well. Here, it is important to keep in mind that the gradiometer design of the qubit requires the coil to be arranged to create the needed field gradient. This can be achieved easily by placing the coil off to one side of the design. FastHenry will be able to calculate the resulting mutual inductance to find the exact placement needed.

If the bias coil consists of two counter-wound loops, it creates fields that average to zero along the axis of symmetry of the circuit. Since the mutual inductance between the bias coil and the squid is desired to be zero, this provides a natural place for the squid loop.

Since the bias coil needs to carry a relatively large amount of current, it is important to design its traces with a sufficiently large cross-section to ensure that it can carry the required current without losing its superconducting properties.

4.2.3 Squid

To achieve sufficient sensitivity to the qubit state while maintaining immunity to outside biases the squid should also be laid out using a gradiometer design. FastHenry, again, can generate the resulting mutual inductances between the squid loop, the qubit loop, and the bias coil, as well as the inductance of the squid loop. Thus, this tool can be used to optimize the size, shape, and relative position of the squid loop. One additional point of concern here is to keep the capacitive coupling between the squid and the qubit at acceptably low levels.

4.3 Materials and Processes

The last step in designing the phase qubit integrated circuit is to choose the materials and processes to be used in the fabrication.

4.3.1 Superconductor

The first choice here is which material to use as the superconductor. Again, many options are viable, but Aluminum, Niobium, or Rhenium make for good candidates due to their accessibility, high superconducting transition temperatures T_c, and the availability of processes for deposition and etching. Due to the relative simplicity of oxidation, Aluminum has been the superconductor of choice for this experiment.

4.3.2 Junction Dielectrics

The dielectric which forms the tunnel barriers for the Josephson junctions used in the circuits is commonly a function of the used superconductor. Due to the small scales of these barriers, it is fairly hard to use anything but an oxide of the superconductor. For this experiment, the junctions' tunnel barriers will therefore be formed by amorphous aluminum oxide. Unfortunately, the quality of the junction dielectric is one of the crucially important details that will determine the final device performance. This is due to the very small thickness of these junctions as it will causes large oscillating electric fields to form across the dielectric during device operation. These fields can easily couple to individual defect states in the tunnel barrier provided they resonate at the qubit's operating frequency [Neeley et al., 2008a]. Therefore, a crystalline barrier would be a very desirable achievement.

4.3.3 Crossover Dielectric

Another material of great concern is that used for general insulating layers like in the external parallel plate capacitor or the wiring crossovers. Even though the fields formed across these during operation are much smaller, their larger bulk more than makes up for this by providing a greater number of defects. According to our current understanding, it is this bath of defect states that is the predominant cause for qubit energy relaxation [Martinis et al., 2005]. Thus, it is necessary to pay close attention to the development of a material with as low a defect density as possible. Again, crystalline dielectrics would be the preferred solution, but for now, amorphous silicon seems to provide a reasonable and much more obtainable alternative.

4.3.4 Wafer

The last choice is the wafer to be used: Since electric fields created by amorphous dielectrics can polarize a 2D electron gas in silicon, we choose sapphire (crystalline Al_2O_3).

Chapter 5

Phase Qubit Fabrication

The fabrication process of building the qubit samples is very similar to the process of building integrated semiconductor circuits. Its 1 μm-resolution seven-lithography-step process is fairly involved by the standards of common condensed matter physics experiments, but still rather straightforward by the standards of current IC technology.

5.1 Mask Design

The process begins with the design of the qubit layout in a standard CAD program like L-Edit. The design is fractured into the different layers described below and exported to *.GDS files, which can be used to "print" the mask plates

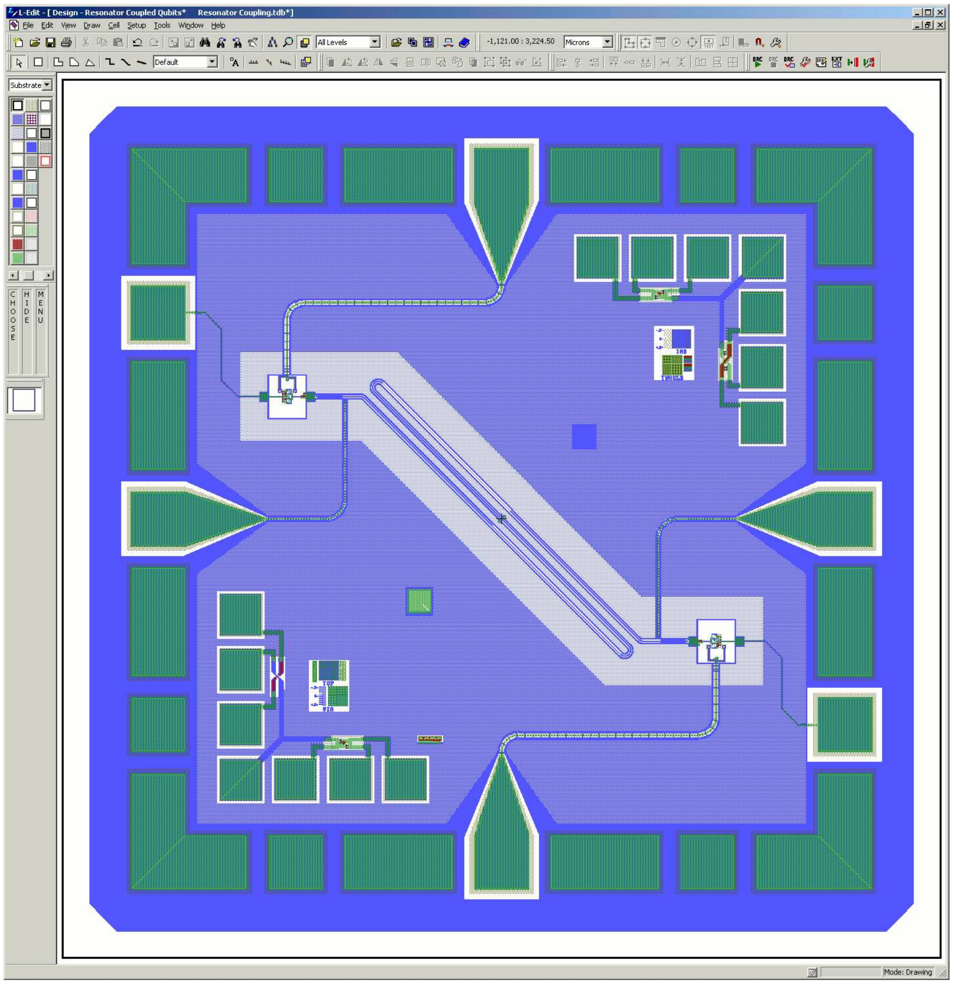
L-Edit - [Design - Resonator Coupled Qubits* Resonator Coupling.tdb*]
File Edit View Draw Cell Setup Tools Window Help
All Levels
-1,121.00 : 3,224.50
Microns
Default
Substrate
CHOOSE
HIDE
MENU
Mode: Drawing

Figure 5.1: L-Edit Mask Layout Tool: The CAD program L-Edit assists the design of integrated circuits by allowing for scriptable layered composition of geometrical elements that represent traces in the final circuit.

that will be used in the later fabrication steps. During processing, the features on the mask will be reduced by a factor of 5 when they are transferred to the wafer. Since the smallest features in our design are on the length scales of 1 μm, the "printing" of the masks is somewhat involved and requires a specialized mask-writer. It uses a laser or shuttered light source to create the pattern rather than a simple ink-jet or thermo-transfer printer, which is sufficient for feature sizes of 10's to 100's of μms.

5.2 Fabrication Overview

The basic qubit design consists of five layers that are defined in seven lithography steps. The bottom layer, the "Base Wiring" layer, contains aluminum traces that define most of the electrical connections in the circuit. This layer is covered by an "Insulator" layer followed by another wiring layer, the "Top Wiring" layer. Vias (holes) in the insulator layer allow for electrical connections between the base and the top wiring layers. Up to here, the qubit design very closely resembles a miniaturized two-layer printed circuit board. The fourth layer, called the "Junction Dielectric" layer, is a very thin oxide layer on top of the top wiring layer. This layer is capped with the "Junction Wiring" layer, another aluminum layer that provides the electrical counter-contact to the junctions. Overall, these five

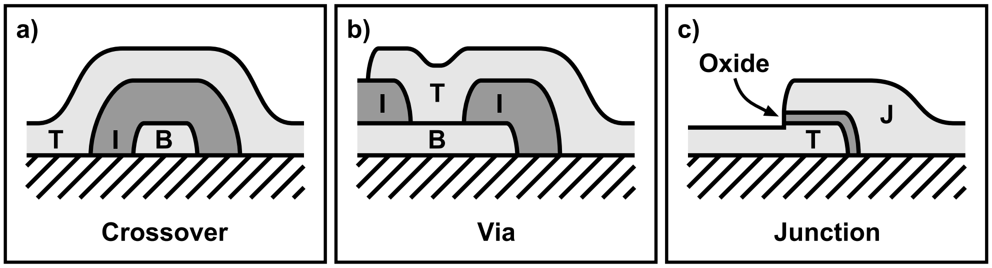

Figure 5.2: Fabrication Building Blocks – a) Crossover: The insulator layer “I” allows for a top wiring trace “T” to cross a base wiring trace “B” without making electrical contact. b) Via: A hole in the insulator layer “I” allows the top wiring “T” to connect to the base wiring “B”. c) Junction: A tunnel junction is formed using a controlled oxide on the top wiring “T” to provide a thin barrier to the junction wiring layer “J”.

layers provide the basic building blocks of the qubit circuit: Traces, crossovers, vias, and junctions (see Figure 5.2).

5.3 Base Wiring Layer

5.3.1 Aluminum Sputter Deposition

The fabrication begins with the definition of the base wiring layer. For this, a sapphire (crystalline Al_2O_3) wafer (thin round disk) is covered with a 150 nm thick layer of aluminum using a method called “Sputtering”. Sputtering is a process in which a disk of pure aluminum (called “Target”) is bombarded with argon ions. The impact causes aluminum atoms to be ejected from the target. If the sapphire wafer is positioned close to the target, some of the ejected aluminum atoms will hit and settle on the wafer to slowly build up an amorphous metal layer on the wafer. The rate at which this layer grows is slow enough ($\sim 10\,\frac{\mathrm{nm}}{\mathrm{min}}$) to allow for its thickness to be controlled to an accuracy of several nm. Sputtering needs to be done in a vacuum system that is initially pumped to $\leq 10^{-7}\,$Torr to remove contaminants. The sputtering process itself is done with an argon pressure of around 1-10 mTorr to find the right balance between a sufficiently high availability of argon ions and a sufficiently long mean free path for the argon ions and aluminum atoms so that they can reach their respective destinations.

5.3.2 Lithography

The blanket aluminum film now needs to be defined into the traces required for the base wiring layer. This is done via an ICP etch (see Section 5.3.3) through a photo-resist mask. The mask is created with a process called photo-lithography. For this, the entire wafer is covered with a thin film of photo-resist (here: SPR-955), a polymer-solution that changes its soluability when exposed to UV light. A few drops of the solution are applied onto the wafer. The wafer is then spun at $4,000\,$rpm for one minute to distribute the solution across the wafer. The speed, spin-time, and viscosity of the photo-resist determine the final thickness of the layer (here $\sim 1\,\mu$m). The covered wafer is then heated slightly to pre-harden the

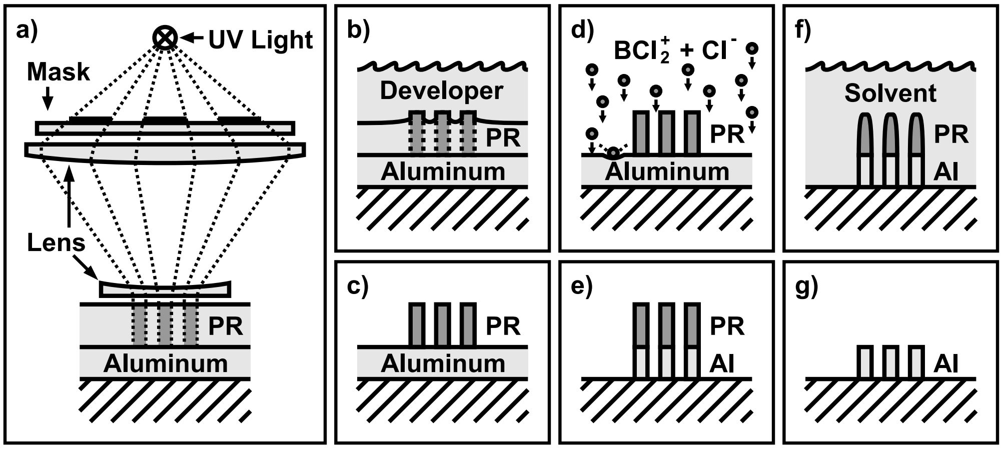

Figure 5.3: Photolithography and Etching – a) Exposure: Irradiation with UV light changes the soluability of the photo-resist (PR). A mask is used to block the UV radiation where desired. Optics reduce the pattern by 5×. b) Development: Developer selectively removes the photo-resist. c) Completed development: The mask pattern has been replicated in the photo-resist. d) Etch: A mixture of ionized gases removes the metal where it is not protected by the photo-resist. e) Completed etch. f) Strip: A solvent removes the remaining photo-resist. g) Completion: The mask pattern has been replicated as traces on the wafer.

photo-resist before it is placed into a machine called a "Stepper". The Stepper exposes an array of 5× reduced copies of the base wiring pattern "printed" on the mask plate created in L-Edit. For this it shines UV light through a system of lenses and the mask onto the wafer for 1.2 s at a time. When all exposures are complete, the wafer is dipped into a developer like MF-701, which removes the more soluable parts of the photo-resist. This leaves a positive image of the mask pattern on the wafer as shown in Figures 5.3a-c.

5.3.3 ICP Etch

Next, the wafer is loaded into a machine called an "ICP Etcher" (ICP = Inductively Coupled Plasma). Inside the etching chamber, the wafer is exposed to ions that will attack and remove the desired material. In this case, the chamber is filled with a low pressure mixture of BCl_3 and Cl_2 gas. An RF-plasma in the chamber dissociates the gas into BCl_2^+ and Cl^- ions. These are accelerated towards the wafer by an electrostatic bias. There, they contact the aluminum film wherever it is not protected by photo-resist. The Cl^- ions will react with the aluminum to form aluminum chloride gas, which is pumped out of the chamber via a turbo pump. Since the etching is done primarily via the chemical reaction between the ions and the substrate and not via physical bombardment, this etching method can be highly selective in the materials that it will remove. For example, the above described chemical mixture will remove the deposited aluminum, but not the aluminum oxide (sapphire) wafer below it.

5.3.4 Photo-Resist Strip

After the wafer comes out of the etching chamber, the remaining photo-resist is removed from it by a dip in acetone or in a chemical called "Stripper" which is specially formulated to effectively remove the given type of photo-resist. Sometimes the removal can be a bit tricky, since the Cl^- exposure during etching can harden the resist. Complete removal is important, though, since left-over resist will impact adhesion of later layers as well as qubit performance. At this point, the aluminum base wiring layer is completed.

5.4 Insulator Layer – Part I

5.4.1 PECVD Deposition

The process continues with the deposition of 250 nm of amorphous silicon to form the insulator layer. This deposition is done with a PECVD (Plasma Enhanced Chemical Vapor Deposition) system, which dissociates concentrated silane gas SiH_4 into its constituent ions using an RF-plasma. The silicon ions settle on the wafer and form an amorphous insulating layer. The deposition rate ($\sim 80\,\frac{\mathrm{nm}}{\mathrm{min}}$) is again low enough to allow for 10 nm resolution in the created film thickness. The insulator deposited in this step helps to form the capacitor in the qubit circuit. During the qubit operation, it will be subjected to electric fields created by the capacitor plates. Since these fields "contain" the qubit state, defects in the insulator that influence them will directly impact qubit performance. Specifically, defects like loose bonds that can form quantum mechanical two-level states contribute significantly to the qubit's energy relaxation time T_1. Thus, the quality of the film deposited in this step is of crucial importance for the final performance of the devices. Since many parameters (deposition temperature, gas pressure, plasma intensity, etc.) contribute to this film quality, a lot of material science and engineering has to be invested into the optimization of this step. The fact that the PECVD system used here is shared between many users from different groups using very different materials further complicates this optimization since the different recipes leave behind a constantly changing chemical environment in the deposition chamber. Extensive cleaning and chamber conditioning before each deposition is therefore required to achieve the needed film qualities. The bulk quality of the film also needs to be balanced with other physical requirements like the internal stress of the film and its adhesion to the base wiring layer as well as the sapphire substrate.

5.4.2 Via Cut

Next, the above described lithography – ICP etch – strip process is used to cut holes (vias) into the insulator layer in the locations where the traces in the base wiring layer need to be contacted by the top wiring layer. The recipe used for the etching is based on SF_6 gas, which removes amorphous silicon, but not aluminum or aluminum oxide.

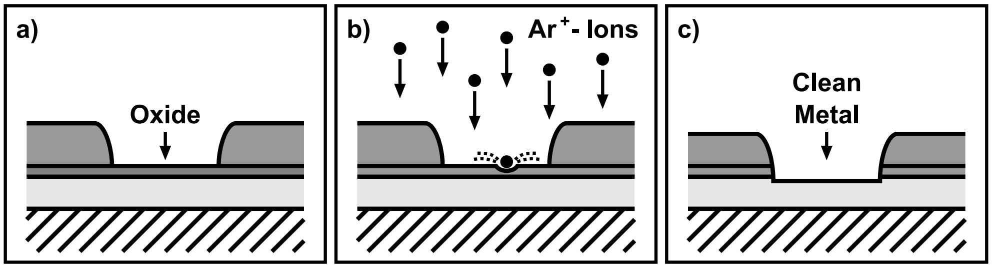

Figure 5.4: Clearing Vias from Native Oxide – a) Native oxide: Aluminum very quickly forms a 2 nm-thick native oxide layer when exposed to air. b) Argon mill: The entire sample is bombarded with argon ions to remove a thin layer of material everywhere. c) Clean via: After a while, the native oxide has been removed, exposing the clean base wiring layer and allowing for good electrical contact to the top wiring layer.

5.5 Top Wiring Layer – Part I

5.5.1 Argon Mill

After the vias have been defined and the sample is cleared from any photoresist, the sample is returned into the aluminum sputter system for the deposition of the top wiring layer. Since aluminum oxidizes fairly quickly when it is exposed to air, an oxide layer will have formed on the surface ($\sim$ 2 nm thick) of the base wiring traces. As this oxide layer is not electrically conductive, it needs to be removed before the top wiring aluminum is deposited to guarantee perfect electrical contact between the two layers. This removal is done in situ with an argon mill step that precedes the aluminum deposition as shown in Figure 5.4.

During this step, argon ions are accelerated with an electric field of around 500 V towards the sample where they impact the surface and remove material through physical bombardment. Since this process is not based on a chemical reaction, the resulting etch cannot distinguish between the different materials on the substrate. Thus, the etch needs to be timed such that it only removes the aluminum's oxide layer, but not the trace beneath. But as the mill rate can be closely controlled with the beam current density, this is not hard to achieve.

5.5.2 Aluminum Deposition

Using the same aluminum sputter technique as described above, one can now deposit 200 nm of aluminum to form the top wiring layer.

5.5.3 Junction Gap Cut

For reasons that will become clear during the junction definition, the first step in defining the top wiring traces is to cut holes only in those regions necessary to prevent a shorting of the qubit junctions (see Figure 5.5). The cutting is performed with exactly the same lithography – ICP etch – strip process that was used to define the base wiring traces.

5.6 Junction Layers

5.6.1 Oxidation / Deposition

The next step is to form the qubit junction. The junction consists of a thin layer of aluminum oxide sandwiched between two aluminum electrodes. The thickness of the oxide needs to be very well controlled, since it will determine the junction's critical current. Thus, the uncontrolled native oxide layer that formed on the top wiring layer during exposure to air needs to be removed with the argon mill step described in Section 5.5.1. After, a controlled amount of oxygen is bled into the chamber of the sputter system to oxidize the exposed clean aluminum to the desired depth. Immediately after, the entire wafer is covered with another 150 nm of sputtered aluminum to form the junction's counter-electrode.

5.6.2 Junction Definition via Argon-Chlorine Etch

Since the critical current of the junctions not only depends on the thickness of their oxide, but also on their area, a range of critical currents can be created across the wafer by offsetting the position of the features slightly for the different rows on the wafer during the photo-lithography. This time, the etching of the junctions is not done via the usual BCl_3/Cl_2 etch, since the oxide layer on the top wiring is too thin to allow for a selective ICP etch to stop on it. The usual etch

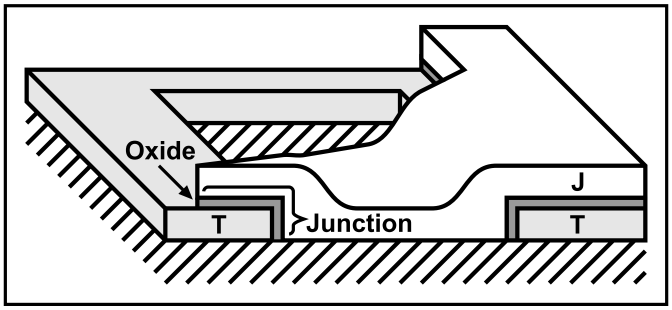

Figure 5.5: 3D View of Junction: The qubit and squid junctions are formed by cutting a hole into the top wiring layer “T”, forming a controlled oxide on it, and covering it with the junction wiring “J”. This process forms two junctions at a time: A small junction (left) under the wedge-shaped tip of the junction wiring and a large junction (right) on the opposite side of the hole. The large junction is big enough to essentially behave like a short.

would remove not only the desired aluminum from the junction layer, but also all exposed aluminum from the top wiring layer followed by all parts of the base layer that aren't covered by the insulator. To prevent this, an argon-chlorine plasma is used to perform a slow mill (Ar mill) that carries away the aluminum (reacts with Cl) to prevent shorting of the junctions through re-deposited material. This recipe has a much lower ($< 10\,\frac{\mathrm{nm}}{\mathrm{min}}$ versus $\sim 1\,\frac{\mu\mathrm{m}}{\mathrm{min}}$) and more controllable etch rate. To ensure that the junction counter-electrodes are entirely disconnected from the top wiring layer, the etch is timed such that it cuts a bit into the top wiring layer to guarantee that the aluminum from the junction layer is fully removed. This is the reason why the top wiring layer was deposited 50 nm thicker than the base wiring and only small holes were cut into it in the previous step as it allows the top wiring layer to protect all lower layers from the milling. The mill is followed by the usual photo-resist strip.

5.7 Top Wiring Layer – Part II

Next, the traces of the top wiring layer are defined with the usual lithography – ICP etch – strip method. During this step, it needs to be ensured that the traces in neither the junction layer nor the base wiring layer are exposed to the etch. The latter should already be covered by the insulator layer or traces in the top wiring layer, while the former need to be explicitly protected by photo-resist. Since during all ICP etch steps the sample is bombarded with ions, electrical charge accumulates across the wafer. To ensure that this charge does not arc through and destroy the junctions, the top wiring traces need to contain shorting straps that provide a current path around the junctions.

5.8 Insulator Layer – Part II

Since even the most optimized amorphous silicon is not defect-free enough to not impact qubit performance, it is desirable to remove it wherever possible. This is done in the next step using the same lithography – ICP etch – strip step that was used to cut the vias into the insulator layer before.

5.9 Top Wiring Layer – Part III

Since the insulator removal is the last etch step in the process, the above mentioned shorting straps can now be removed. Due to the described charging effect, they cannot be removed with the usual ICP etch. Instead, the lithography is followed by a wet etch in which the sample is submerged into Transene-A, a chemical that etches aluminum without creating a charge. The etch is followed by the usual photo-resist strip.

5.10 Dicing

The main advantage of clean-room fabrication is that the stepper used during photo-lithography can very quickly shoot almost 100 copies of our 0.25” × 0.25” qubit design onto the 3” diameter wafer for each step. In all other fabrication steps these 100 devices are processed in parallel at no extra effort. The only thing left to do at the end is to cut the wafer into the desired dies. This is done on a dicing saw which consists of a $200\,\mu$m-thick blade spinning at $30,000\,$rpm that is forced through the wafer. Since sapphire is the second hardest material known to man, the blades cut with embedded particles made out of diamond, the hardest material known to man. Before this, the wafer is protected with a fresh coat of photo-resist. Once this coat is removed, the fabrication is complete.

Chapter 6

Device Testing Equipment

The testing and characterization of the qubit devices employs a host of different tools which will be examined in this chapter. During this discussion, it is important to keep in mind that, even though some of the described tools are fairly sophisticated, almost all of them are based on well-developed technology and are readily available for off-the-shelf purchase (although sometimes at significant cost). This is one of the reasons that make superconducting integrated circuits an interesting candidate for quantum computation.

6.1 Physical Quality Control during Fabrication

Already during the fabrication, the qubits need to be subjected to constant quality control. Thus, the first section of this chapter focuses on the tools used in the clean-room during fabrication

6.1.1 Optical Microscopy

The angular resolution of the unaided eye is about 0.02-0.03 ° and it can focus on objects as close as 15-30 cm (depending on age). This corresponds to a minimum feature size that the eye can resolve of about 50-150 μm. Since the features in the qubit design get as small as 1 μm, it is often not possible to tell with the naked eye whether a step in the fabrication yielded the desired result. But with an optical microscope that can magnify the qubit's features by up to $1,000\times$, it is fairly easy to tell whether the development of the photo-resist or an etch step left residue behind, or whether the features in the different layers are correctly aligned. It is even sometimes possible to judge the quality of the aluminum films or insulator films by looking for proxy features like pitting or other large-scale surface defects. This makes a good optical microscope an indispensible tool that is usually used after every single deposition, lithography, etch, or strip step.

6.1.2 Scanning Electron Microscopy

Optical microscopes quickly reach their limitations if the feature sizes drop into the nm range. Things like step coverage or the quality of the insulating barrier or even surface roughness at more detailed levels cannot be resolved with an optical microscope. The reason for this is the finite wavelength of light. This finite wavelength gives photons a “size” of several hundred nanometers, making it impossible to resolve features smaller than this size without distortion to the image due to interference effect. To circumvent this limit, a different type of microscope can be used that is known as an SEM (Scanning Electron Microscope). As the name suggests, this microscope does not use optical photons to probe the surface, but a beam of electrons that is reflected off the sample. The small size of the electrons allows for magnifications up to $250,000\times$, which is sufficient to, for example, look at the profile of the 2 nm oxide barrier inside the qubit junction. The use of electrons requires the sample to be conductive so that it can reflect the electrons efficiently. Commonly, non-conductive samples are therefore first covered with a thin layer of gold. The images generated by the SEM can naturally not preserve the color of the sample.

6.1.3 Atomic Force Microscopy

To resolve even finer details than the SEM can deliver, one can employ the help of an AFM (Atomic Force Microscope). It works by scanning a needle with a tip the size of a single atom across the surface of the sample. The tip's deflection due to the forces between it and the surface is measured by reflecting a laser beam off the cantilever supporting the tip. With this technique, it is possible to resolve the surface structure of a film at the atomic level ($\sim 0.1\,\mathrm{nm}$). The process of scanning a needle across a surface is a very delicate and therefore slow operation, making the AFM much less convenient to use.

6.1.4 Dektak

Back at the opposite end of the resolution spectrum, the Dektak functions as the AFM's easier, less sophisticated cousin. It is also based on dragging a tip across the sample surface, but the readout mechanism is much cruder. The Dektak is very convenient to use as a tool that can quickly determine the step profile of layered films with a resolution of around a few nm. The Dektak is therefore very useful for investigating deposition and etch rates by giving a good estimate of the resulting film thickness or trench depth.

6.2 Electrical Screening after Fabrication

6.2.1 4-Wire Measurements

As described in Chapter 5.6.2, the junction features on the qubit wafer can be exposed with pass-shifts such that they provide a gradient in the junction areas across the wafer. This allows for post-selection of the correct junction resistances which can compensate for variations in the junction oxidation process. For this, the qubit design contains test junctions on each die that are geometrically identical to the qubit junctions but provide two contact pads on either side. Since the junction resistance is fairly small and the contact between probes touching the wafer has a comparable or even larger resistance, it is necessary to employ a technique called a 4-wire measurement to measure the junction resistance independently of the lead resistance. For this, the junction is biased by a current through one pair of pads as shown in Figure 6.1. The voltage developed across the junction is then measured via the other two pads. Since the current can be determined reliably despite additional series resistance and since the lead resistance is negligible compared to the internal resistance of the volt meter, this measurement gives an accurate reading even under varying lead resistance. With this technique, those dies can be chosen from the qubit wafer that are the most likely to yield qubit junction resistances in the right range.

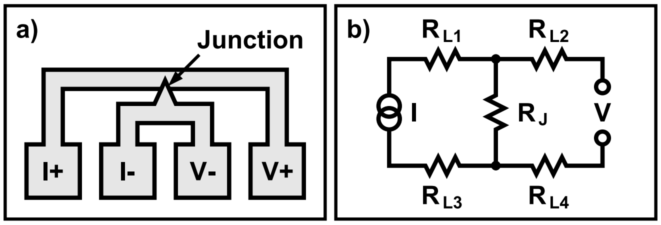

Figure 6.1: 4-Wire Measurement – a) On-chip trace layout: Each side of the junction to be measured is connected to two pads, one of which is used for a current bias and the other for a voltage measurement. b) Electrical diagram: A current source drives a known current I through the series resistor of interest R_J. This causes a voltage $V = I\,R_J$ to develop across the resistor, which can be measured at the indicated terminals. If the volt-meter is assumed to have infinite internal resistance, the wiring resistances R_{L1} through R_{L4} do not influence the result.

6.2.2 Adiabatic Demagnetization Refrigerator

Unfortunately, the other circuit elements, like the inductor, capacitor, and squid, can only be tested cold, i.e. at temperatures where the aluminum traces become superconducting ($< 1\,\mathrm{K}$). Since the final cool-down in the dilution refrigerator (see Section 6.3.1) is rather time consuming and expensive, it makes sense to screen the candidate dies further in an "Adiabatic Demagnetization Refrigerator" (ADR). This refrigerator can cool samples to around 100 mK and stays cold for a few hours at a time. The cooling happens in two stages: The first stage, a closed cycle pulse tube cooler, works very much like a conventional refrigerator, except that it circulates helium gas rather than tetrafluoroethane (the environmentally conscious replacement for Freon). This cooler brings the refrigerator down to temperatures around 4 K. The final temperature is reached using the cooling power of the randomization process of magnetic spins. Once the refrigerator reaches 4 K, a superconducting magnet is energized to generate a strong field ($\sim$ 4 T) through a gadolinium-gallium garnet (GGG) crystal and a ferric ammonium alum (FAA) salt pill. The magnetic spins inside the GGG and FAA align with the applied field, releasing heat that is absorbed by the closed cycle cooler. After the system equilibrates, the stages are thermally disconnected and the magnetic field is slowly relaxed back to zero. This causes the magnetic spins to randomize and absorb heat to facilitate the increase in entropy. This effect cools the GGG crystal to about 1 K and the FAA to around 100 mK until the randomization is complete. These temperatures are cold enough to perform initial electrical tests on the qubits as described in Chapter 8 (Squid I/Vs and Squid Steps). But due to the short duration of the cold period, the amount of electrical and vibrational noise of the refrigerator, and the limited wiring possibilities due to the low cooling power, more involved qubit experiments are not possible.

6.3 Quantum Measurements at 25 mK

6.3.1 Dilution Refrigerator

Once a sample has been deemed likely to perform, it can be prepared for the real cool-down in the "Dilution Refrigerator" (DR). This refrigerator reaches 25 mK in four stages. It consists of a vacuum that houses one tank filled with liquid nitrogen (LN_2) and one with liquid helium (LHe). Just like water at standard pressure boils at 100 °C no matter how much energy is put into it, LN_2 boils at 77 K and LHe boils at 4 K. This keeps the two reservoirs at these temperatures as long as they are kept full of liquid. (This need for keeping the reservoirs filled is one of the reasons why the DR is much more costly to operate than the ADR.) Attached to the LHe reservoir is a thin tube that slowly feeds LHe into a small volume called the "Pot". This pot is pumped on by a vacuum pump to lower the boiling point of the LHe further to 1.5 K. The remaining cooling to 25 mK is achieved with a closed system that cycles a mixture of He-3 and He-4. Its cooling power results from the entropy increase when He-3 mixes with He-4. A continuous mixing is achieved with a two-chamber design consisting of a destillation chamber ("Still") that selectively removes He-3 by evaporation and a mixing chamber where the He-3 is allowed to mix back in with the He-4. The entropy increase in the mixing chamber cools the chamber down to around 25 mK. Since this process happens continuously, the DR can theoretically stay cold forever. The rather large cooling power of the mixing process ($\sim$ 20-50 μW) allows for a large sample stage ($\sim$ 150 in^2) with several hundred electrical connections. The DR we use in our lab is a custom design, but it is also possible to buy pre-built DRs from companies like Oxford and Janis.

6.3.2 Sample Mount

Even though it might seem trivial at first thought, the exact design of the sample mount used for connecting the sample inside the DR is actually quite crucial. This is due to the fact that the states of the qubit correspond to electromagnetic oscillations inside the circuit at GHz frequencies. Therefore, any box-modes that the sample holder might have that resonate in this frequency range will couple to the qubit and degrade its performance. Also, several microwave drive lines converge in a 0.25" square die that need to be electrically isolated in a way that each qubit can be addressed individually with minimal electrical crosstalk from its neighbors. Another concern is the reaction of the box to internal and external magnetic fields. A printed-circuit-board (PCB) design with a centered hole in which the qubit chip is placed, for example, is not a useful design since the current loop formed by the PCB interacts with the flux biases applied to the qubit and leads to flux settling times in the many 10's of microseconds. An optimal design seems to be a box machined out of solid aluminum with coaxial feeds for the microwave lines. This allows the entire box to become superconducting and shield the sample from external magnetic fields. The box needs to support the chip with minimal contact above a cavity to not form a ground-plane underneath the chip that capacitively disperses microwaves across the chip. Nevertheless, the chip needs to be very well grounded to the box, which is achieved by hundreds of closely spaced wire bonds (see Section 6.3.3) that reduce the overall inductance in the grounding. The microwave lines should maintain their $50\,\Omega$ impedance throughout the box to minimize pulse distortion caused by reflections. This makes the sample mount one of the few parts that we have not been able to just buy off the shelf. But after the box was designed in a CAD program, it was very straightforward to have it machined to sufficient accuracy by the university's machine shop.

6.3.3 Wire Bonding

To be able to wire the qubit into the DR, the sub-millimeter traces of the qubit chip need to be connected to macroscopic coaxial cables that can be managed by hand. This can be done with a device called a “Wire Bonder”. Wire bonding works by feeding a thin ($\sim 1\,\mathrm{mil}$) aluminum wire through a tip such that the tip can push the wire down onto the surface that it is meant to adhere to. While applying pressure to the wire, the tip is vibrated rapidly with ultrasonic waves.

This melts the aluminum wire and bonds it to the surface. This allows the system to run wires between pads without the use of solder. This technique makes it possible to make contact to pads that are only a few hundred μm wide.

6.3.4 Dilution Refrigerator Wiring

The other non-off-the-shelf component needed is the wiring inside the DR (see Figure 6.2). It consists of several different parts that have to be chosen/built to manage heat loads, noise, and cryogenic properties.

Starting at the outside of the qubit box, the squid is connected immediately to a $\sim 30\,\Omega$ shunting resistor. This resistor is needed to limit the voltage generated in the squid when it switches to the voltage state as explained in Chapter 4.1.3. If this resistor is omitted, the squid switching generates a large amount of quasiparticle excitations in the qubit circuit which reduce qubit performance. The next component along the squid line is a copper powder filter (Cu). This device consists of a wire wound into a spiral that is sitting in a cavity filled with copper powder and epoxy for thermal contact. Electrically, it functions as a very quiet low-pass filter that absorbs most noise coming down the squid line. The $\sim 30\,\Omega$ resistor and the copper powder filter are both mounted at the 25 mK stage of the DR to prevent them from creating noise due to their temperature. Along the squid line, at the 4 K stage of the DR is a resistor network that splits the line into a bias and

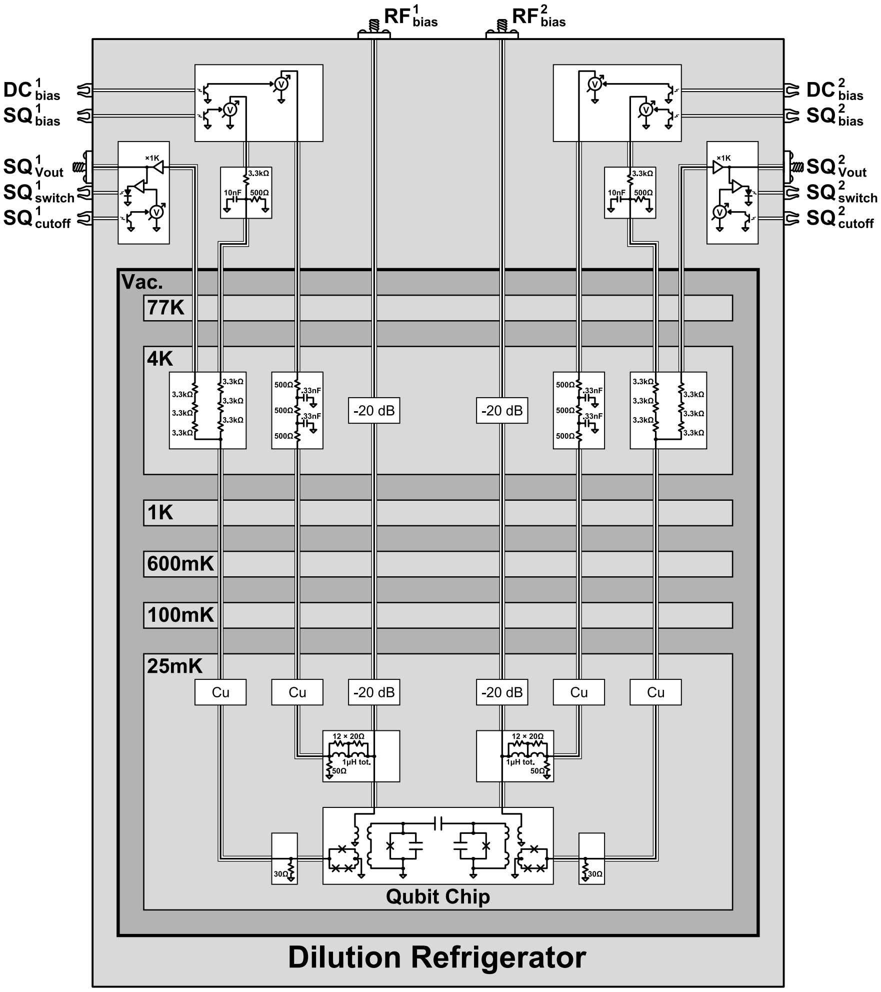
$\mathbf{RF}^1_{\mathbf{bias}}$
$\mathbf{RF}^2_{\mathbf{bias}}$
$\mathbf{DC}^1_{\mathbf{bias}}$
$\mathbf{SQ}^1_{\mathbf{bias}}$
$\mathbf{SQ}^1_{\mathbf{Vout}}$
$\mathbf{SQ}^1_{\mathbf{switch}}$
$\mathbf{SQ}^1_{\mathbf{cutoff}}$
$\mathbf{DC}^2_{\mathbf{bias}}$
$\mathbf{SQ}^2_{\mathbf{bias}}$
$\mathbf{SQ}^2_{\mathbf{Vout}}$
$\mathbf{SQ}^2_{\mathbf{switch}}$
$\mathbf{SQ}^2_{\mathbf{cutoff}}$
V
×1K
3.3kΩ
10nF
500Ω
Vac.
77K
4K
3.3kΩ
500Ω
.33nF
-20 dB
-20 dB
1K
600mK
100mK
25mK
Cu
Cu
-20 dB
-20 dB
Cu
Cu
12 × 20Ω
1μH tot.
50Ω
30Ω
Qubit Chip
Dilution Refrigerator

Figure 6.2: Dilution Refrigerator Wiring.

a readout line. These two lines are identical up to room temperature where the readout line goes into the PreAmp card (see Section 6.3.6) and the bias line goes through a 10:1 divider/low-pass filter to a FastBias card (see Section 6.3.5).

The flux bias line of the qubit is split right outside the sample box into an RF and a DC part by a "Bias-T". The RF part simply passes straight through the bias-T into a 20 dB attenuator, while the DC part passes through a shunted inductor coil to a copper powder filter. For noise reasons, these components are again all located at 25 mK. Up the bias lines at 4 K, the RF part passes through another 20 dB attenuator and then straight to room temperature. The total attenuation of 40 dB is split between the two temperature stages to balance the heat load on the refrigerator with the noise generated. The DC line passes through an RC low pass filter network at 4 K and terminates at room temperature at another FastBias card.

6.3.5 FastBias Card

All DC (< 100 kHz) biasing is done via custom designed digitally controlled voltage sources we call the "FastBias" cards. They consist of low-noise 16-bit digital-to-analog converters (DACs) that are controlled by an FPGA which receives binary commands serially through a fiber-optic cable. The FPGA is programmed to turn off the card's clock source whenever it is not receiving or processing data. This protects the voltage output from the large amount of white noise that digital electronics always create during switching. The fiber-optic link was chosen to allow for electrical isolation of the DR from the control electronics. For the same reason, the FastBias and PreAmp cards are powered by a battery box rather than a power supply that plugs into a wall outlet.

6.3.6 PreAmp Card

The PreAmp Card is used to detect the switching of the readout squid by monitoring it through the readout line. After optional low- and high-pass filtering, it amplifies the incoming signal by $1,000\times$ and then compares it to a digitally controlled cutoff voltage. The result of this comparison is made available via a binary fiber-optic line. Just like the FastBias card, the DAC that sets the cutoff voltage is controlled by an FPGA that turns off the card's clock whenever it is not needed.

6.3.7 Bias Box

Both the FastBias and the PreAmp card are designed to plug into a rack-mount bias box. This box supplies the cards with power from batteries and provides two analog and two digital buses that the cards can use to make signals available for monitoring. This is useful for the early stages of the qubit bring procedure described in Chapter 8 (specifically the Squid I/V measurement).

6.3.8 GHz DACs

The fiber-optical signals are sent to the FastBias and received from the PreAmp cards by another custom electronic device, the GHz DAC. It is a high-speed FPGA-based signal generator that uses two 14-bit DACs running at 1 GHz to synthesize waveforms with frequencies up to 500 MHz. These waveforms are used to control the RF portion of the qubit bias. This is done by compositing one waveform that is directly synthesized by a DAC and another that is the result of using the DACs to modulate a carrier microwave signal with an I/Q mixer. The first half of the waveform generates Z-rotations and the measure pulse. The second half of the waveform is used to generate X/Y-rotations. The GHz DAC boards are controlled by a computer via a 100 MBit Ethernet connection and can synchronize to each other via a 10 MHz reference clock and a daisy-chain trigger.

The GHz DACs as well as the bias cards could easily be replaced by off-the-shelf electronics, but the cost of the commercial equivalents would increase the price of adding one more qubit channel by a factor of more than 20. This, and the desire for exact control over the electronics' performance, prompted us to design these modules ourselves.

6.3.9 Anritsu Microwave Source

The microwave signal that is modulated by the GHz DACs is generated by a CW microwave generator manufactured by Anritsu. The device is phase-locked to the GHz DAC board with a 10 MHz reference clock and is controlled by a computer via GPIB.

6.3.10 Microwave Components

Several off-the-shelf microwave components are used in the setup to shape the pulses. These consist of attenuators, amplifiers, filters, and I/Q mixers.

Attenuators are used to weaken the signal strength without distorting its envelope. They are frequently used in microwave setups since any component or connector that is not matched exactly to 50 Ω reflects part of the microwaves passing through it. These reflections need to be damped from the signal to not distort the final output. An attenuator in line with the offending component can help reduce this problem since the desired pulse passes through it only once, while reflections pass through it three or more times and thus get attenuated more.

Amplifiers perform the exact opposite function of attenuators: They increase a signal's amplitude. For cost reasons, we have developed our own microwave amplifier circuits, but off-the-shelf components perform just as well.

Filters selectively attenuate certain frequency components of a microwave signal. Low-pass filters, for example, remove frequencies above a certain cutoff. Components like I/Q mixers frequently generate higher harmonics (e.g. they turn a 5 GHz signal into mostly 5 GHz with some 10 GHz, 15 GHz, etc.). These undesired harmonics can be filtered out easily with off-the-shelf low-pass filters.

I/Q mixers split an incoming microwave signal into two halves (I and Q) of equal amplitude. One of the halves (Q) is then phase-shifted by $90\,^\circ$. Each half is passed through a mixer, which multiplies its amplitude by another input signal. Finally, the two halves are summed together to create the output. I/Q mixers can be used to generate signals of arbitrary phase and amplitude. This can be seen from simple trigonometrics. The input signal, $\sin\omega t$, is split into a $\sin\omega t$ part (I) and a $\cos\omega t$ part (Q). These are then multiplied by an I and Q input signal and summed, yielding: $I\sin\omega t + Q\cos\omega t$. This can be rewritten as $A\sin\left(\omega t + \delta\right)$ where $A^2 = I^2 + Q^2$ and $\tan\delta = \frac{I}{Q}$.

Chapter 7

Control Software – LabRAD

7.1 Motivation

The electronics that control the qubits are extremely complex and need to adapt very quickly for a wide variety of continuously changing experiments. This inadvertently leads to a highly involved software effort to provide a meaningful user interface to the experiment. The usual approach in physics is to write one monolithic program to provide the required control. This frequently leads to a code base that becomes unmanageable very quickly, especially if multiple people are working on the same program. The general lack of formal training and programming experience in physics labs commonly leads to "quick hacks" rather than a well-structured, maintainable code base. Due to the complexity of this experiment, the scalability requirements, and the structure of the group, this approach is unacceptable here. I therefore decided that we needed to develop a custom software platform that allows us to efficiently attack all aspects of the project.

7.2 Requirements

7.2.1 Scalability

One of the requirements for this software platform is dictated directly by the DiVincenzo criteria. Just like the qubit design needs to be scalable, so do the control electronics and the software setup. To make a software platform scalable, two main issues need to be addressed.

First, the software needs to abstract all resources in such a way that adding or replacing control devices or qubits does not require a rewrite of the entire code base. In the long run it will be unacceptable to have to develop an entirely new control program for every additional qubit that gets added to the setup. The basic software needs to look the same, independent of whether it drives one, two, three, or many qubits. Furthermore, it needs to be possible to implement calibration routines at different levels that seamlessly correct for imperfections in the underlying hardware. Higher level functions need to be able to talk to a virtual device that can be treated as ideal. I will call this requirement the Abstraction

Requirement.

Second, as the control effort grows, it is quite possible that a single computer will not be able to provide or control enough resources for the entire setup. Even though computers are rapidly becoming more powerful, their processing power, memory capacity, and hard-disk space will always be finite. On top of that, there are resources that have actually become **less** plentiful in modern computers than they used to be. One example is the number of PCI slots that a modern computer offers. This can become an issue, since a quantum computer consisting of hundreds of qubits might require a large number of different control devices that interface over a GPIB bus, for example. Each GPIB bus can only support about 30 devices, though. Thus, to support 100 microwave generators, the computer needs to provide room for three or more GPIB interface cards. It is therefore very likely that any large-scale quantum computing effort will eventually be controlled by several different computers that perform partially complementary and partially identical tasks. The software should be able to transparently support this. I will call this requirement the Load-Sharing Requirement.

7.2.2 Maintainability

As the size of a software project grows, it becomes harder and harder to keep it maintainable. If such a software project is undertaken by a group of self-taught people without any formal training in best practices, it becomes virtually impossible to keep the code base clean. This is aggravated by the fact that physicists often see software development as a necessary evil that stands in the way of them focusing on the actual experiment. This leads to an overwhelming preference for selfish quick fix solutions without any thought about potential implications for the rest of the project. If, for example, a Matlab function needs to be able to perform a new task that requires more information, a quick addition of new parameters to the function definition is often preferred over the creation of a new copy of this function. If this function had been used by other members of the group, this modification will most likely break their code, leading to unnecessary debugging headaches. Since it will not be feasible to train everyone in the required best practices, the underlying software platform should instead try to support the "physicist coding style" in such a way that the resulting damage to the code base can be minimized or at least locally contained. To achieve this, there needs to be a very well defined way to break the monolithic project into small modules that can be independently developed, tested, and maintained. I will call this requirement the Modularity Requirement.

7.2.3 Efficiency

To promote acceptance of this software platform by the group it needs to fulfill another very important requirement: The platform needs to make the development process noticeably easier right from the start. If the benefits of learning and using the framework are not immediately obvious, or, worse yet, if coding within the framework is perceived to be a hassle, the common disinterest in the future maintainability of the code base will cause the platform to be rejected. Therefore, the platform needs to be well integrated with the programming language(s) that would otherwise be used for the software effort. It needs to seem lightweight and natural. I will call this requirement the Integration Requirement.

Since it is usually not a priori clear exactly which experiments will need to be done in the future, the software platform also has to support quick turn-around development of new functionality. To achieve this, the platform needs to allow for features to be added and extended in a seamless, backwards-compatible way. It also needs to provide straightforward access to all functionality available. I will call this requirement the Flexibility Requirement.

7.2.4 Performance

Last, but certainly not least, the platform has to be able to deliver enough performance to not limit the data rate beyond the physical limits imposed by the experimental setup itself. Since the rate at which data can be collected directly impacts the group's turn-around time, any slowdown due to the software will translate directly into an undesirable overall slowdown of the project's rate of progress. I will call this requirement the Performance Requirement.

7.3 Approach

To address these issues, Matthew Neeley and I developed a platform called LabRAD. RAD is a programming acronym that stands for Rapid Application Development and usually refers to visual programming environments like Delphi. LabRAD is based on a set of core ideas that help attack the above listed requirements.

7.3.1 Modularity

The most important concept for the development of LabRAD was the Modularity Requirement. Being able to break the project up into small pieces not only allows for more rapid progress thanks to parallel development of independent modules, but it also provides some valuable clues for attacking the other requirements.

Most modern programming languages already have the ability to modularize source-code, but many of them lack important features to make the provided facilities sufficient in a physicist driven project. The biggest problem is that most languages do not have a way to manage and index the available modules. In most cases the modules are kept somewhere in the file system and are accessed based on their location or filename. In a physics lab, this usually leads to one code repository per group member. Everyone works in their own directories on their own code using their own conventions. If code is to be shared, usually a copy is made from one user to another, effectively branching the development of that module as both users will now make edits to their copies of the code. Duplication of effort is extremely common due to the difficulties involved in reusing code. This is further aggravated by the general lack of documentation.

To alleviate these problems, LabRAD takes modularization from the source-code level to the "executable" level. Every module in LabRAD is developed as a completely independent program (or script) and the interaction with external functionality happens at runtime. The different modules communicate via a well defined protocol through a central dispatching agent, the LabRAD Manager. When connecting, each module has to identify itself as either a Server Module or a Client Module. Client Modules only use the functionality provided by the LabRAD system, while Server Modules also provide new functionality that other Server and Client Modules can use. This distinction allows the LabRAD Manager to maintain an index of all available Server Modules to quickly give an overview of the available functionality. Each Server Module then registers so-called "Settings" with the Manager that provide the interface to the Server's functionality. Both the Servers and their Settings have a human-readable name for easy identification as well as a numeric ID that allows for reduced traffic and quicker Request routing. When registering a Setting with the Manager, the Server also has to provide a help text and specifications about which types of data this Setting requires and returns. This information is made available by the Manager to provide a one-stop source for basic documentation of all features of the system.

Since Settings are addressed by name and can register multiple acceptable data types it is easily possible to extend the collection of Settings or the functionality of individual Settings provided by a Server without breaking backwards compatibility. Most modern programming languages can achieve this same result through things like overloading of functions or polymorphic SubVIs, but this ability is usually seen as a fringe feature and does not get much attention. Therefore, casual programmers like physicists usually either don't know about it or don't feel comfortable using it.

Another benefit of this hard separation of Modules are the “social” implications. Since Modules are accessed through the abstract interface provided by LabRAD, it is possible to use them without ever having to look at their source-code. Thus, Modules can and will be effectively treated as black boxes, which has two main advantages.

First, since it is never necessary to look at a Module’s source-code, the perception of the Module as a magical whole remains, quenching the desire for another user to ever mess with it. Just like most people would never consider editing a program like Open Office despite the fact that the source code is available. People either find ways around missing features or request their implementation from the original developers, rather than trying to add new features themselves, which always bears the risk of breaking existing code.

Secondly, the hard separation makes it easy to declare a specific person responsible for the maintenance of the Module. This way, any code changes are made by the person that has the best overview of potential implications for both reliability and performance.

Last, but not least, the modular nature of LabRAD addresses the Abstraction Requirement very effectively. A LabRAD system can be layered just like the HALs (Hardware Abstraction Layers) used in operating systems. For example, there can be one Module each to provide access to the lab’s different GPIB buses. A single second Module talks to all these and uses “*IDN?”-queries to create a central list of all available GPIB devices by manufacturer and model number. A third group of modules provides access to the higher-level functionality of a specific type of device by translating a “Set Frequency to 6.5 GHz” request to an “Output OF6500MHz to device 18 on GPIB bus X” request, and so forth. This translation can even include calibration steps that transparently correct for the underlying hardware’s imperfections, allowing the developer of higher level functionality to assume the hardware to be ideal.

This approach makes it possible to build “driver stacks” inside the LabRAD system that mimic the ones used in operating systems like Windows. These driver Modules, if written correctly, immediately handle as many devices of the same type as are available, making the system trivially scalable, just like it takes no extra effort on the software side to add more than one USB flash drive to a computer.

It is even possible to replace an instrument with an entirely different model that provides equivalent functionality. All one has to do is to make sure that the interface presented by the managing Server Module provides the same Settings. Just like most Windows applications don’t need to worry about the exact type of video card that the system uses.

A hard separation of Modules is also helpful for the development of complex systems. It makes it possible to attack the problem in small chunks (and from different angles simultaneously, if desired) making the overall effort a lot more manageable.

Due to the dynamic nature of the experimental setups typically found in physics labs and for debuggability, LabRAD employs a flat Module hierarchy, meaning that all Modules sit at the same level, giving each Module access to all other Modules. This allows Client Modules to tap into the system at any point in an abstraction stack to gain as much direct access to the hardware as needed. This way, every user can choose to bypass any Modules that are too restrictive or unstable. It also makes the system a lot easier to debug as every layer in the abstraction stack can be tested individually. This approach is different from many operating system models where only special facilities (like DirectX) allow for lower level module access.

7.3.2 Network Distribution

The next decision to make was how to interface the Modules with each other. There are several different options, the two most relevant of which are using an IPC (Inter Process Communication) mechanism or using a standard TCP/IP network connection, like the ones used to connect a local browser to web services. IPC allows different applications on one computer to exchange data, while a network connection allows applications on the same (through loopback) or on remote computers to talk to each other. Usually IPC delivers quite a bit better performance, but a TCP network connection has two major advantages that far outweigh the performance impact in most cases.

First, a TCP connection can transparently support different Modules running on different computers that are attached to the Internet from anywhere in the world if needed. This allows LabRAD to be trivially distributed to inherently address the Load-Sharing Requirement. It also provides other conveniences like fast remote access.

Second, every major programming language has native support for TCP/IP connections available. This makes it possible to provide an interface to the LabRAD system from languages as diverse as Python, Delphi, LabVIEW, Java, Matlab, etc.

The performance impact from using TCP over IPC is reduced significantly by the immediate availability of fast networking infrastructure up to 10 GBit/s. This performance is likely to increase even further in the future, making the choice very acceptable. In fact, for the current requirements in our lab, a dedicated 100 MBit connection is more than enough to not cause a noticeable performance impact, so the dedicated 1 GBit LAN that we are currently using will be able to deliver sufficient performance for quite some time. Furthermore, network latencies and bandwidth limitations can often be hidden entirely by things like pipelining.

7.3.3 Cross-Language and Cross-Platform

Since every programming language has its unique strengths but certainly also its weaknesses, it is a major advantage to be able to write different Modules in different languages. For example, Modules that require the best possible performance should be written in a compiled language like Delphi or C++. Modules that rely on a comprehensive user interface can easily be designed in LabVIEW. And Modules that script complex processes can be quickly implemented in Python. The data-structures that LabRAD supports were specifically designed to make implementation in different languages as natural as possible and thus it is possible to provide very seamless interfaces to LabRAD for almost all programming languages. In this way, LabRAD addresses the Flexibility and the Integration Requirement in one shot.

Furthermore, it is possible to design LabRAD Modules for different operating systems that can all communicate via the OS agnostic network protocol. This allows for further flexibility in the choice of resources used for the project. For example, employing computers running a free operating system like Linux can slightly reduce the overall cost of the project.

7.3.4 Performance

Addressing the Performance Requirement is a more pervasive task that spans not only the general design criteria, but also every single line of implementing code. It involves decisions like the exact choice of programming language for each module, the threading structure for multi-threaded Modules (specifically the LabRAD Manager), the binary layout of the protocol itself that influences parseability and the amount of traffic, etc. It also dictates in part the way in which the project needs to be broken up into Modules to allow for efficient implementation of concepts like pipelining and parallel processing.

7.3.5 Open-Source

Due to its very general design, only a few Modules in our LabRAD Setup are specific to the experiment at hand. A large fraction of the system can be useful in many other labs that are facing the same issues, like code base maintainability, remote access, etc. We therefore decided to share the project with the world and publish it as open source software under the GPL license on SourceForge.net. This not only gives us access to the software maintenance tools that SourceForge provides, but also gives other developers from around the world a chance to join the project and share Modules with us which they developed but may become useful to us in the future. Beyond that, it forces us to do a good job documenting the project which will be useful for future members joining the group.

7.4 Components

The LabRAD platform consists of three fundamental components on which every experiment-specific implementation is built: The LabRAD Protocol defines the exact binary layout of the network packets that are exchanged between the different components. The LabRAD Manager routes packets between different Modules and provides some basic system features. The LabRAD APIs provide access to the LabRAD system from different programming languages.

7.4.1 LabRAD Protocol

The LabRAD Protocol is designed for both speed and flexibility. Data is encapsulated in Packets that consist of routing and context information followed by an arbitrary number of Records that each wrap up one Request or Response for a Server's Setting. The ability to send multiple Requests in one Packet can reduce network traffic significantly and can be used for other purposes like defining atomic operations, etc.

LabRAD uses binary encoding of the data to reduce overhead. The Protocol is designed such that the LabRAD Manager can auto-detect whether a Client or Server Module uses a little endian or a big endian data format (MSB versus LSB first). This significantly simplifies Module implementations in different languages and across different platforms.

All variable length components of the payload are prefixed by a length specification that allows the decoding algorithm to predict the amount of memory needed to hold the parsed data. This greatly reduces the number of costly memory reallocation operations during the parsing. Furthermore, the Packet structure is such that the data stream can be parsed in three independent steps, breaking it first into Packets, then into Records, and finally into data. This again facilitates efficient implementation and memory management by allowing for things like lazy parsing.

The data types that LabRAD uses are modeled closely after the ones supported by LabVIEW since it is the most restrictive language we use in the lab. In fact, the native LabVIEW “Flatten to String” and “Unflatten from String” functions are sufficient to implement the LabRAD protocol. Just like in LabVIEW, the type of the binary data is described by a Type Tag, except the LabRAD Type Tags are strings that are designed to be human readable. LabRAD supports the basic data types listed in Table 7.1 and the composite data types listed in Table 7.2. Type Tags can be annotated with unit information or comments using the conventions

Table 7.1: Basic LabRAD Types

Type Tag	Name	Description	Example Data
b	Boolean	Flag	True
i	Integer	Signed whole number	$-1,500,000,000$
w	Word	Unsigned whole number	$3,750,000,000$
s	String	Text	“Hello World”
v	Value	Real number	-1.637×10^{23}
c	Complex	Complex number	$3.2 + 1.6i$
t	Timestamp	Time and date	1/15/2006 12:53pm
?	Any	Placeholder for any type	
_	Empty	Unspecified array type	

Table 7.2: Composite LabRAD Types

Type Tag	Name	Description	Example Type Tag and Data
*? or *n?	Array	n-D list of data	*s: “Tom”, “Tim”, “Jim”
$(\ldots)$	Cluster	Collection of data	(sw): “Tim”, 27
E or E?	Error	Error message	E: 15, “Can’t divide by 0!”

Table 7.3: LabRAD Type Annotations

Type Tag	Name	Description	Example Type Tag
[...]	Units	Units of a Value or Complex	v[GHz]
{...}	Comment	Type tag annotation	s{Name}
:...	End	Marks end of type tag	b: Turn on(T)/off(F)

Table 7.4: LabRAD Data Flattening Rules

Tag	Length[Bytes]	Description
b	1	False: 0x00, True: anything else
i	4	Signed 32bit Value
w	4	Unsigned 32bit Value
s	$4 + \text{len}(s)$	Flattened i giving length followed by raw data
v	8	64-bit double precision value
c	$8 + 8$	Flattened (vv) for real and complex parts
t	$8 + 8$	Signed 64-bit Integer giving seconds since 1/1/1904 12:00am UTC followed by signed 64-bit Integer giving fractions of seconds
_	0	Nothing
(...)	$0 + \text{len}(\ldots)$	Flattened cluster elements concatenated in order
*?	$4 + \text{len}(?)$	Flattened i giving number of entries followed by flattened elements concatenated in order
*n?	$4 \times n + \text{len}(?)$	n Flattened i's giving number of entries along each dimension, followed by flattened elements concatenated in order
E	$4 + 4 + \text{len}(txt)$	Flattened (is)
E?	$\text{len}(E) + \text{len}(?)$	Flattened (is?)

Table 7.5: LabRAD Packet Structure ((ww)iws)

Field	Type Tag	Description
Context	(ww)	Context in which the Packet is to be interpreted
Request	i	Packet's Request ID: > 0: Request $= 0$: Message < 0: Reply
Src/Tgt	w	Incoming packet: Source ID Outgoing packet: Target ID
Records	s	Flattened Records concatenated in order

Table 7.6: LabRAD Record Structure (wss)

Field	Type Tag	Description
Setting	w	Setting ID that the data is meant for / came from
Type Tag	s	Type Tag of data contained in this record
Data	s	Flattened data

listed in Table 7.3.

For transmission over the network, the data to be sent is encapsulated into Records and Packets. The Packet and Record format can be described using LabRAD Type Tags and is described in Table 7.5 and Table 7.6, respectively. The Packets are flattened to binary using conventions that are compatible with LabVIEW as explained in Table 7.4.

As an example, a Packet for Target 1 (the Manager) in Context (0, 8) with Request ID 5 containing one Record for Setting 3 with String Data “Test Server” would be flattened to either (Big Endian):

```
00 00 00 00 00 00 00 08 00 00 00 05 00 00 00 01 00 00 00 1C 00 00 00 03
00 00 00 01 73 00 00 00 0F 00 00 00 0B 54 65 73 74 20 53 65 72 76 65 72
```

or (Little Endian):

```
00 00 00 00 08 00 00 00 05 00 00 00 01 00 00 00 1C 00 00 00 03 00 00 00
01 00 00 00 73 0F 00 00 00 0B 00 00 00 54 65 73 74 20 53 65 72 76 65 72
```

Notice that “Test Server” is flattened by prepending the length (00 00 00 0B) to form the data content, which is in turn prepended by a length again (00 00 00 0F) when it is inserted into the Record.

7.4.2 LabRAD Manager

The LabRAD Manager acts as the central routing agent for all LabRAD traffic. This makes it a potential bottleneck for the entire system and thus extra care had to be employed when designing it to ensure maximal performance. The LabRAD Manager was written in Delphi, a version of Object Pascal that provides RAD (Rapid Application Development) capabilities through visual designers that assist in the design of the user interface. Delphi has not received much attention in the US market, but in Europe it is highly valued as a very efficient language with a compiler that produces code that rivals any modern C++ compiler's in execution performance. It provides the usual high-performance features like code optimization, data alignment, inlined Assembly, etc.

Apart from optimized memory management, one major point of concern was a well-designed thread layout for the application. A first approach that dedicated one thread to each network connection had to be abandoned due to the high cost of Windows context switching (switching between threads). Making the application dual-threaded instead yielded an order of magnitude performance improvement. The first thread, the network thread, is dedicated to routing all traffic and handling Requests to the Manager and the Registry Server Module. It uses direct access to the event-based, non-blocking methods provided by the Windows Sockets 2

Architecture to manage network connections with maximum performance. The second thread, the GUI thread, handles the user interface. It keeps the list of active connections current and makes sure the application stays responsive to user interaction. These two threads interact “rarely” and in a way that minimizes impact on the network thread.

The LabRAD Manager provides several features, the most important of which is Packet routing between Server and Client Modules. In the process of routing, the Manager ensures that the data contained in a Request Packet matches the types registered by the Server as acceptable. During this check, the Manager also converts the units of any Values or Complex Values that are sent to a Setting that specifies units in the accepted types. For this, the LabRAD Manager parses composite unit strings into their base components and respective fractional exponents. For example “m/s^2” will get parsed into “m” with exponent 1 and “s” with exponent -2. This is done for the source unit string and the target unit string. To find the conversion factor the Manager divides the source units by the target units by subtracting the exponents of all base components in the target units from the exponents in the source units. All base components that end up with exponent 0 are then dropped. Now, for the first time, the Manager tries to identify the base components and to convert them to SI units. This allows the Manager to handle unsupported units as long as they drop out during the division process. The SI conversion factors then get raised to their corresponding exponents and multiplied together to yield the final conversion factor. The resulting quantity has to be unit-less for the conversion to succeed.

The LabRAD Manager implements a minimal set of security features. They consist of an IP white-list that only allows connections to the Manager originating from certain computers as well as an MD5-hash based challenge-response authentication mechanism. Currently there are no facilities for restricting access to individual components of the LabRAD system. Once a Module is authenticated, it can call any Setting on any Server with any of the accepted data types. Thus, it is strongly recommended to limit access to the system to only trusted agents employing additional techniques like secure tunnels through firewalls. In our case, the LabRAD system runs on a dedicated LAN with no direct connection to the Internet. A Linux computer provides outside access to this network via authenticated SSH tunneling. It is possible, though, to implement access control on a per-Module basis.

The LabRAD Manager is written such that any errors or exceptions that happen during the processing of incoming or outgoing data get handled gracefully either by an error message sent to the offending party, or, for more severe errors, by the termination of only the offending network connection and an entry in the Manager's error log. This should allow for maximum uptime of the LabRAD

Manager. In fact, in our lab, the only time we ever have to restart the Manager is if we add new features and in the process of debugging these. We frequently run the Manager without problems for many months at a time.

7.4.3 LabRAD APIs

The LabRAD APIs (Application Programming Interfaces) provide access to the LabRAD system from different programming languages and different platforms. Currently these APIs include full support for Delphi and Python as well as basic support for LabVIEW, Java, and Matlab (via Java). If needed, any other language and platform that provides support for TCP/IP network connections can be added.

Since the LabRAD API provides the interface to LabRAD for the Module developers, it is important that they are written with usability in mind to effectively address the Integration Requirement. Due to the differences in data types, this can mean very different things for different programming languages.

The feature that probably requires the most thought is the choice of interface for handling the different LabRAD data types. This choice is highly dependent on the type of programming language that is to be supported. The difficulty primarily lies in LabRAD's support for cascadable heterogeneous composite data structures (Clusters), which lead to infinitely many valid data types. In dynamically typed languages like Python it should be possible to represent LabRAD data seamlessly using its native counterpart (if available). In statically typed languages like Delphi and Java on the other hand, an interface to a special data object needs to be designed that makes the access to the contained data as painless as possible while still providing good performance. Dynamic data types like Variants can sometimes be used here to make implementation more natural. Thanks to Variants, for example, it is possible in LabVIEW to simply pass any supported native data structure directly into the Packet assembly functions. Unfortunately, converting received Packets back into LabVIEW data is much less elegant as it requires explicit type casting.

Since the LabRAD Protocol specifies target Servers and Settings by ID rather than name, all APIs should implement seamless, preferably cached, lookup of the required IDs at runtime. This not only makes the code more readable and thus easier to troubleshoot, but also facilitates backwards compatible changes to Servers. To make caching easier, the LabRAD Manager preserves a Server's ID across reconnects and provides mechanisms to announce the connection and disconnection of Server Modules. In addition, IDs can be looked up via the "Lookup" Setting (ID 3) of the LabRAD Manager (ID 1).

The LabRAD APIs should also seamlessly handle Request IDs. To allow for pipelining, it is possible for a Client to send multiple Requests to a Server before waiting for their completion. The Client then needs to be able to sort through the Replies to re-associate them with their corresponding Requests. This is done by tagging every outgoing Request with a unique, positive Request ID. When a Server handles the Request, it tags the Reply with the negative of the Request ID (Request 5 yields Reply -5). It should always be possible for the API to take care of the entire management of these Request IDs. Therefore, if an API is written well, the Module developer should not ever have to come in contact with Request IDs. For the development of Server Modules, the API can even hide the entire Packet structure of incoming Requests by always passing the Requests to the Server on a Record by Record basis. This way, the Server developer does not need to be concerned at all about whether, for example, two Request arrived as two Packets with one Record each or as one Packet with two Records. By being in sole charge of the construction of the Reply Packet, the API can then enforce the correct Packet structure to ensure compliance with the LabRAD protocol.

The LabRAD APIs should also assist Server development as much as possible with the management of Contexts. Contexts are intended to provide a method for Server Modules to maintain data that persists across Requests. For example, a Client might select a device it would like to use. To reduce redundant traffic, the Server should be able to remember this selection as well as any relevant device settings for future Requests. For this, Contexts provide an index into a memory structure kept in the Server that holds the collection of all relevant configuration data.

As far as the LabRAD Protocol is concerned, a Context is nothing more than a Cluster of two Words (ww). When forwarding Packets, the LabRAD Manager makes the following alteration to a Packet's Context before sending it on: For incoming Packets where the first Context Word is 0, the LabRAD Manager sets it to the ID of the source Module. For outgoing Packets where the first Context Word is equal to the ID of the target Module, the Manager sets it to 0. Clients will usually use 0 as the first Context Word. Servers will then receive Requests "in a Context" where the first Word is equal to the ID of the Client that sent the Request. If a Server needs to make secondary Requests to fulfill the original Request, it can choose whether to make these in its own Context (first Word equals 0) or in the Context of the original Request. The latter can make sense if this Server simply assists Clients in the communication with an underlying Server. The Client, for example, selects the device with the underlying Server and does some initial setup and then calls the assistant Server to complete a more involved part of the device setup.

The second Word of a Context can be chosen by the Client at will and allows this Client to have multiple simultaneous Contexts open with any given Server. Apart from being generally useful, this feature is indispensable for the implementation of pipelining and certain types of parallel processing. The LabRAD Manager keeps track of all Contexts that all Servers have ever seen and, when a Client disconnects, it sends a Context Expiration Message to the relevant Servers to allow them to free up the memory that stores the data associated with the Context. Clients can also ask the Manager to send Context Expirations at any other time to make sure a Context can be safely reused without having to worry about a prior state. To allow for this assumption, all Context aware Servers (i.e. the ones that preserve state across Requests using Contexts) need to honor these Context Expirations.

To assist with the development of Server Modules, a LabRAD API should provide at least basic Context management. In statically typed languages, this can be as simplistic as maintaining a single untyped pointer for each Context that the developer needs to manage by providing functions to create and free the required data-structures. In dynamically typed languages like Python this can be as comprehensive as automatically providing a dedicated dictionary (or hash-table) for each Context that can contain all relevant data.

Contexts are also used to control the flow of Request handling. A Server must handle all Requests within a Context in the exact order in which they were received. To limit potential problems, this order should be enforced directly by the API. Requests in different Contexts can be interleaved to handle them in parallel.

This is especially useful if secondary Requests are used, since there is no reason for the Server to idle while waiting for the completion of these secondary Requests. All Servers that take part in pipelined Requests have to support multi-tasking at least while waiting for secondary Requests, otherwise they will implicitly serialize the Requests in the pipeline, which renders it ineffective. The API should provide multi-tasking capability as seamlessly as possible. Again, this can mean very different things for different programming languages. In event-based languages like Delphi, an event can be provided that signals the arrival of a Request. The event handler can then choose to complete the Request in one shot or to return a flag that tells the API that the Request is still incomplete. The API then needs to put all further Requests in this Context on hold, but can already announce the arrival of Requests in different Contexts. Once a Request finally completes, the Server calls an API function that handles the Reply and reactivates the Context for processing. Another approach, in languages that support it, is the use of co-routines or generators that allow a Request handler function to pause in the middle of its execution and wait for a specified event to cause its continuation. For performance reasons, it is NOT advised to handle each Request in a separate thread, since the overhead of thread creation is usually immense. If multiple threads are unavoidable, a thread-pool should be used to which the Requests are dispatched.

Since the API guarantees that all Requests in one Context are correctly serialized, the Request handler function of a Server Module can always assume exclusive access to the associated Context data. This makes it extremely straightforward to write Servers that behave as if they were multi-threaded without having to worry about the headaches associated with memory access race conditions.

7.5 Our Setup

Taking a closer look at the LabRAD setup used for this experiment will help shed more light on the some of the concepts explained above. The setup consists of several layers of hardware abstraction and makes use of pipelining, parallel processing, and load-sharing to achieve essentially experiment-limited performance.

The full LabRAD setup handles almost everything in the lab including the monitoring and control of the dilution and adiabatic demagnetization refrigerators. But for the purposes here, a look at only the parts directly relevant to the qubit operation shall suffice since the employed concepts are mostly the same.

7.5.1 Overview

As detailed in the previous chapters, the two qubits each use two "DC" lines for the flux and squid bias, one high-speed line for measure pulses and Z-rotations,

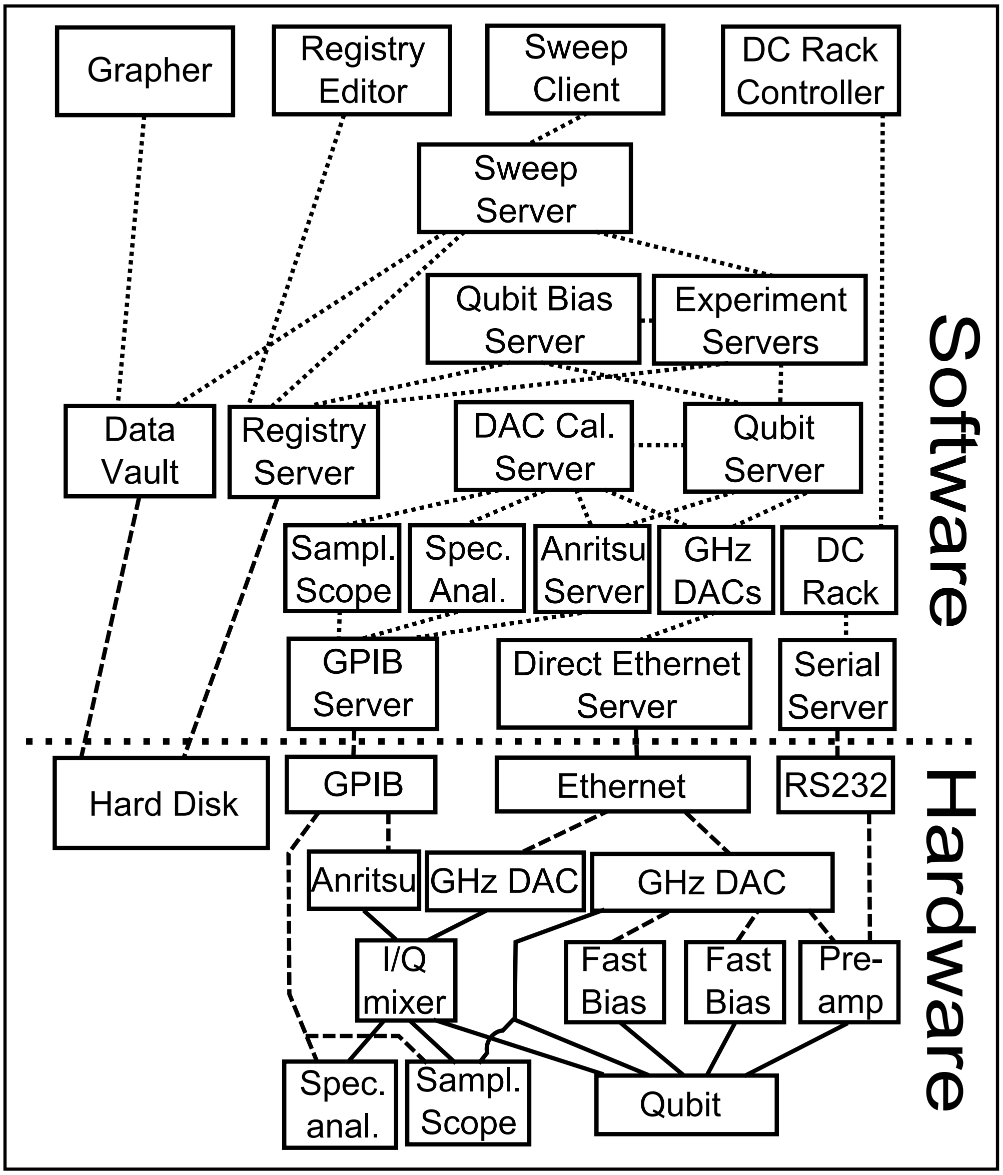

Figure 7.1: Control Layout: The control software layout mimics the hardware layout as much as possible.

one microwave line for X/Y-rotations, and one readout line to detect the switching of the squid. These lines get driven by several GHz DAC boards and one microwave source. The GHz DAC boards connect to a computer via ethernet and the microwave source uses a GPIB connection. Furthermore, the squid readout pre-amplifier gets configured via a serial link.

These three hardware interfaces – GPIB, ethernet, and RS-232 – are exposed to the LabRAD system via one dedicated Server Module each – the GPIB Server, the Direct Ethernet Server, and the Serial Server. Above these sits a set of device Servers that each implement the communication protocol of one device type to provide higher level functions like “Set the output of the microwave source to 7 GHz at 2.7dBm”.

The next level in the stack – the Qubit and DAC Calibration Servers – abstract the exact hardware configuration by allowing the user to, for example, load a sequence of voltages into the “Squid Bias” channel on “Qubit 2” (rather than “FO 1” on “GHz DAC 17”). This data is then corrected for imperfections in the analog electronics chain and automatically sent to the right GHz DAC board. Up to this level the software does not impose any limits on the hardware control beyond preventing erroneous configurations. Until here, starting from the hardware/software interface, the first layers in the software abstraction stack closely mimic the first layers in the hardware stack.

The next layers correspond to the "applications" (i.e. the experiments) that are to be run on the quantum computer. For now, at the base of this stack is a collection of Experiment Servers that provide functions to take one data point of a specific experiment (Rabi, T1, Bell Violation, etc.). These Servers are called by the "Sweep Server" which varies one or more of the experimental parameters to generate n-D datasets. At the highest level in the stack sit several Client Modules that allow the user to edit parameters, run sweeps, and view the resulting data sets. Let's go over the different Modules in detail starting from the Client end.

7.5.2 DC Rack Controller

Starting with the right-most branch of the diagram in Figure 7.1, the DC Rack Controller consists of a LabVIEW VI that provides an interface to the user to set up the DC bias rack, which houses the FastBias and PreAmp cards. Apart from configuring diagnostic outputs, the most important feature is the ability to control the cutoff voltage that the PreAmp uses to detect the squid's switch to the voltage state. The only function of the DC Rack Controller is to translate user interactions with switches and dials on the front panel to calls to the DC Rack Server that performs the actual updates.

7.5.3 DC Rack Server

The DC Rack Server implements the binary protocol used by the different bias cards to provide access to all features offered by the cards. It allows other Modules in the LabRAD system to select active cards and change their settings, e.g. initialize the reference DAC on a PreAmp-Card, control the behavior of the front-panel LEDs, set the monitor channels, etc. One instance of this Server manages all DC bias racks that are accessible from the LabRAD system via Serial Servers that provide the actual hardware link.

7.5.4 Serial Server

The Serial Server provides direct access to all COM-ports of the computer that it is running on. It can list the available ports, open a port by name, select connection options like the baud-rate, byte-size, and parity, and control the state of the RTS and DTR lines. One copy of this Server is run on every computer that needs to share access to its ports. To give each of these Server Modules a unique name, "Serial Server" is prefixed with the name of the computer that it is running on.

7.5.5 Grapher

At the far left of the connection diagram (Figure 7.1), the Grapher Client Module gives the user a way to view datasets as they are taken or to browse existing datasets. It offers 1D and 2D plotting capability as well as zooming and basic curve fitting (exponentials and parabolas). The Grapher also provides access to meta-data that is stored with a dataset as well as a mechanism for annotating and discussing datasets that resembles an instant messaging chat session. Multiple copies of the Grapher can be run by different users to allow for live discussion of incoming data among collaborators located anywhere in the world. The Grapher accesses the data and meta-information of different dataset via calls to the "Data Vault" Server Module.

Being able to provide powerful data plotting and analysis tools as a separate module rather than as a part of each data taking script significantly simplifies the development of these scripts and thus improves experimental turn-around times. Separating the plotting from the data acquisition and processing also allows for higher data rates as the usually CPU intensive plotting does not take resources away from the data taking. In fact, even over very slow network connections, the same data rates can be achieved in remote data taking sessions as from inside the lab. This is made possible by being able to run the bandwidth intense parts of the data taking on the lab servers and only having the very low bandwidth control panel running on the remote machine. The Grapher will then display points as quickly as it can retrieve them from the Data Vault, potentially reducing the refresh rate of the plots, but usually not impacting the overall time it takes for an experiment to complete.

7.5.6 Data Vault

The Data Vault Server Module acts as the central data storage location for all datasets taken with the LabRAD system. Datasets are stored in a directory structure and accessed by name, much like a conventional computer file system. It provides the usual created/last-accessed/last-modified time stamps as well as other named meta-data that can be stored alongside the data. The data itself consists of a 2D table of LabRAD "Values" in which each row corresponds to one data point. The columns contain first the values along the independent axes followed by the values along the dependent axes. The Data Vault also associates a message log with each dataset to provide the ability to annotate the data later using, for example, the Grapher's chat session. To allow for the Grapher to automatically open datasets and to keep directory listings updated, the Data Vault can send notification messages to other Modules to announce the availability of new data, etc. The data itself is stored simply as flat files in folders on the local hard drive.

Having a central authority manage the data files has several advantages. For one, it prevents access conflicts between different programs trying to open the same data files at the same time. It also makes change notifications much more straightforward than it would be for independent file access. Last but not least, it makes the data available to anyone who is connected to the LabRAD system, even if this connection comes from a remote computer behind a firewall, a scenario that makes it somewhat tricky to make the actual hard drive containing the data files available on the remote machine.

7.5.7 Registry Editor / Server

Just like the Data Vault provides a central location for data storage, the Registry Server provides a central location to store any relevant configuration data. The configuration data is organized as named keys in a directory structure. The keys can hold data of any possible LabRAD Data Type. When reading a key, a Module can specify the desired type of the returned key data. In addition to type-checking, this allows for on-the-fly unit conversion of any stored (Complex) Values.

To guarantee availability and best possible performance, the Registry is integrated directly into the LabRAD Manager. Combined with in-memory caching of keys and the ability to set Context-local override values this allows the Registry to significantly simplify the development of complex experimental setups. This is demonstrated by the interaction between the Sweep Server and the Experiment Servers as detailed in Section 7.5.8.

Just like the Data Vault, the Registry Server stores its data as files in directories on the local hard drive.

The Registry Editor Client Module provides a convenient interface to the end user for changing, adding, deleting, and copying keys in the Registry.

7.5.8 Sweep Client / Server

The Sweep Server uses the functionality provided by the Registry to greatly simplify the implementation of new experiments. It allows the user to specify keys in the Registry whose values get swept in an n-D pattern. For each step in the sweep, the Sweep Server calls a user-selectable "Run"-Setting on another Server. This Setting's responsibility is to run the experiment once to take one data point. All configuration data needed to run the experiment needs to be read from the Registry. Using value-overrides in the Registry, the Sweep Server can then use the "Run"-Setting to take data for different experimental conditions automatically without knowing what the actual values in the Registry do. The Sweep Server expects the "Run"-Setting to return an array of dependent values, which it prepends with the current sweep position (the independent values) and forwards to the Data Vault.

The Sweep Server will automatically call the “Run”-Setting multiple times in different Contexts in parallel. This will cause the execution of the data taking to be pipelined, as actions in different Contexts are automatically executed in parallel whenever possible. In fact, this shields the developer of the “Run”-Setting almost entirely from having to worry about pipelining. All that has to be guaranteed is that the code passes control back to the API whenever it is waiting for other Servers to handle Requests. Since this should be done in any case, though, it does not really add any new requirements. Furthermore, the APIs, if written correctly, will make it very easy to write code in this way.

The Sweep Client provides an interface to the Sweep Server that allows the user to quickly define sweeps and execute them. Before running a sweep, it initializes the dataset with the Data Vault and sets up the Registry Server and Qubit Server. The Qubit Server makes the Sweep Client specific to our experimental setup while the Sweep Server is completely general and can be used in other LabRAD Setups as well. The Sweep Client allows the user to store sweep definitions in the Registry and run them seamlessly on different qubit configurations. It also provides a progress-bar with an estimate of the remaining time until the sweep completes. This can be extremely useful for preparing long sweeps that need to finish within a certain time.

7.5.9 Optimizer Client / Server

Along with the Sweep Client / Server that performs n-dimensional sweeps, we have also developed an Optimizer Client and Server that uses the exact same execution model to perform n-dimensional optimization of a desired experimental quantity using several different optimization algorithms: Nelder Mead Simplex, Particle Swarm, or SPSA. The calling convention of the "Run"-Setting that the Optimizer uses is identical to that of the Sweeper, such that any setting that can be used with the Sweeper can be used without modification by the Optimizer. The only requirement that the Optimizer adds is that the settings return the value (either by itself or as part of an array) that is to be optimized. The optimization can either be a minimization, maximization, or an attempt to get it as close as possible to a given number. The Optimizer runs until it is manually stopped and sends data to the Data Vault that shows its progress.

7.5.10 Experiment Servers

Below the Sweep/Optimizer Server sits a collection of Servers that provide different "Run"-Settings for the different types of experiments needed. Apart from hardware changes, these Servers should be the only part of the LabRAD system that needs to be changed in order to implement new experiments. Since all that these Servers need to do is to provide a Setting that takes a single data point with the parameters provided by the Registry, these Servers are usually extremely simple and concise. This allows for very high turn-around for new experiments. This is further assisted by the fact that these Servers do not even need to know what parameters in the Registry are being swept for a given data run. This commonly allows the Settings to be used for a variety of different types of experiments. As an example, consider a Setting that runs a sequence that consists of only a single microwave pulse on each qubit followed by a delayed measurement. Such a Setting can be used to measure Rabi oscillations as a function of pulse power and pulse length, T_1 decays as a function of the delay of the measurement pulse, multi-qubit swap operations, and even step-edges and s-curves by setting the pulse amplitude to 0 (these experiments are explained in Chapter 8).

When running an experiment, the Sweep Client selects an experimental setup with the Qubit Server that defines the list of qubits that are to be used in the experiment as well as the hardware channels that are needed to control them. The Qubit Server provides a Setting ("Experiment Involved Qubits") that allows the Experiment Servers to request the list of qubits that are part of the current setup. This way, the Experiment Servers can automatically loop over all involved qubits and run the desired sequence on all qubits with the different sequence parameters read from different sub-directories in the Registry. The "Run"-Settings can therefore be written in a way that is independent of the number of qubits that are being used, providing for maximum scalability of the experiment.

7.5.11 Qubit Bias Server

To assist the Experiment Servers in preparing and running an experiment, the Qubit Bias Server provides Settings to set up common parts of the experimental sequences with the Qubit Server. Specifically, it provides a Setting that uses parameters stored in the Registry to generate the flux bias sequence necessary to reset each of the involved qubits and set them to their respective operating biases. Another Setting sets the flux biases to the respective readout bias and executes the necessary squid ramps in the correct order to read out all qubits. Using these two Settings, the only thing that a typical Experiment Server has to do is to configure the high-speed microwave, Z-, and measure pulses and to analyze the resulting timing data in the appropriate way.

7.5.12 Qubit Server

Both the Experiment Servers and the Qubit Bias Server perform their functions via calls to the Qubit Server. This Server maintains a list of all qubits with their respective hardware hook-ups. For example, it knows which GHz DAC boards are involved in the experiment and which qubits their different control lines are connected to. This allows users to simply say “Output a top-hat pulse of this amplitude and length on the ‘Measurement’ line of ‘Qubit 4’ ” rather than having to generate the required memory content and sending it to “GHz DAC Board 17”.

The way in which the Qubit Server provides access to its features allows it to automatically keep the sequences that are running on the different GHz DACs aligned in time to ensure correct execution. It also makes sure that a microwave source that is shared between different qubits is not configured twice with conflicting settings.

But the most powerful feature that it provides is to abstract away the signal deconvolution that needs to be applied to all analog channels to correct for different imperfections in the electronics chain. Modules that talk to the Qubit Server can simply assume that the hardware is ideal and a Gaussian pulse that is to be output on a specific IQ Mixer Channel is automatically corrected to yield the most optimal final output. The Qubit Server achieves this functionality via calls to the “DAC Calibration Server”.

To run the actual sequence, the Qubit Server sends the required setup packets to the “GHz DAC Server” to tell it to upload the correct data onto the right boards and to run them with the right number of repetitions. These calls also contain a collection of other initialization packets that are to be sent to other Servers right before the execution of the sequence. This allows the Qubit Server to forward the burden of setting up the microwave generator and other involved hardware to the GHz DAC Server. As described in Section 7.5.14, this is necessary to allow for the execution of the sequences to be pipelined in the most efficient way.

7.5.13 DAC Calibration Server

The DAC Calibration Server provides functions to modify signal data that is to be output via the GHz DAC boards in order to correct for imperfections in the output electronics chain. For this, the DAC Calibration Server first takes a set of calibration data that measures, for example, the response of the output electronics to a delta-function signal. This data is stored in the Data Vault to later allow for a signal to be corrected by taking the desired signal's Fourier transform, dividing the result by the Fourier transform of the delta-function response, and inverting the Fourier transform to recover a corrected version of the data. The corrected data is returned to (in this case) the Qubit Server so that it can be uploaded to the respective GHz DAC boards.

To correct for as many different electronic deficiencies as possible, the DAC Calibration Server needs to take several different calibration datasets. These include traces returned by a sampling scope as well as measurements with a spectrum analyzer. The DAC Calibration Server takes this data automatically via calls to the “Sampling Scope Server”, the “Spectrum Analyzer Server”, the “Anritsu Server”, as well as the “GHz DAC Server”.

7.5.14 GHz DAC Server

The GHz DAC Server plays one of the most crucial roles in the data taking process. Due to the fact that most of the data taking action is controlled by the GHz DAC boards, it handles pretty much all of the resource scheduling required to run multiple experiments at the same time on the same hardware. The GHz DAC Server can receive requests for data runs in multiple different Contexts at the same time and actively serializes them on the hardware in the most efficient way possible. It does so by allowing a Context to already prepare its run by uploading the required data onto the GHz DAC boards while the previous run is still executing. It then halts the execution of that Context until the previous run completes. This completion is detected via the arrival of all expected timing data at the Direct Ethernet Server. Before this data is actually read from the Direct Ethernet Server, though, the setup of the next run is finalized by sending out the required hardware configuration packages (see Section 7.5.12) and the run is started. This approach minimizes the amount and size of LabRAD Requests that happen between data runs as much as possible to achieve the best possible performance. Since all other processing of data happens in parallel with this, one can achieve very-close-to hardware limited performance by making sure that every single step in the sequence generation and data processing takes less time than the execution of the sequence. If a step in the sequence generation takes longer than the execution, the GHz DAC Server will not have a new context waiting immediately when the previous run is completed, which will lead to downtime. If a step in the data processing takes longer, the data will be taken at maximum speed, but it won't be available for viewing and thus decision making, at the maximum rate.

Since the GHz DAC Server controls all scheduling of data runs on the hardware, it is not only possible to run a single sweep in multiple contexts to allow the different steps of the data taking process to be pipelined, but also to run two completely independent experiments at the same time. This allows two or more users to take data simultaneously with reduced data rate due to the interleaving of the actual runs.

7.5.15 Direct Ethernet Server

The Direct Ethernet Server uses the Winpcap packet capture library to read raw ethernet packets (IEEE 802.3) directly from the network adapters present in the computer it is running on. It provides Settings to select an adapter, send packets, set packet filters (by MAC address, length, and content), wait for the reception of packets, and return the content of received packets. Since it handles the traffic between LabRAD and the GHz DAC boards, its performance has a great impact on the overall throughput achievable with the setup. Thus, this Server was written in Delphi and great care was employed in the design of the threading structure and memory management of the application. For debugging purposes, the Direct Ethernet Server provides a user interface that shows statistics on packet traffic and gives the user a way to monitor the raw byte traffic on the wire. The latter, due to the large volume of data, can severely impact performance, though, and is thus disabled by default.

Like the Serial Server, one copy of this Server is run on every computer that needs to share access to its network adapters. To give each of these Server Modules a unique name, "Direct Ethernet" is prefixed with the name of the computer that it is running on.

7.5.16 Anritsu, Sampling Scope, and Spectrum Analyzer Servers

The Anritsu, Sampling Scope, and Spectrum Analyzer Servers implement the GPIB protocol to each control a certain type of instrument. By talking to all GPIB Servers present in the lab, they compile a list of all devices of the respective type to provide a one-stop location for their control. Essentially, all that these Servers do is generate GPIB command strings and parse GPIB response strings. These strings are sent and read via calls to the respective GPIB Servers. As an example, the Anritsu Server provides a Setting that allows another LabRAD Module to set the output frequency of the microwave generator to a given value. Thanks to the automatic unit conversion in the LabRAD Manager, this Setting can be called with a "Value" with any units compatible with MHz. The Manager will convert the value to MHz (e.g. 6.8 GHz to 6800 MHz) and the Anritsu server will create the correct GPIB command (OF6800MH) and pass it on to the right GPIB Server.

7.5.17 GPIB Server

The GPIB Server, much like the Serial Server, provides direct access to all GPIB bus controllers connected to the computer that it is running on. It can provide a list of connected devices, select an active device, and communicate with the device using timeouts. One copy of this Server is run on every computer that needs to share access to its GPIB buses. To give each of these Server Modules a unique name, "GPIB Server" is prefixed with the name of the computer that it is running on.

7.5.18 IPython

In addition to using the user interface Client Modules described above, the entire LabRAD Setup can also be controlled through a Python command shell, like IPython, and other Python scripts. This gives a user of the system ultimate flexibility to mix and match the access to any part in the system at any level to achieve whichever effect he or she desires. This way, the system can be extensively tested and debugged and it is ready for any form of future experiments that might be devised.

Chapter 8

Single Qubit Bring-Up and Characterization

The limited connectivity to the qubit necessitates the development of a rather involved calibration and characterization procedure. Many relevant quantities cannot be measured directly, but must instead be inferred from proxy-experiments. In fact, the only directly measurable parameter of the qubit is the voltage response to a current bias of the readout squid. Thus, the bring-up procedure begins with an examination of this response.

8.1 Squid I/V Response

As explained in Chapters 2.3.4 and 4.1.3, SQuIDs (Superconducting Quantum Interference Devices) can be used as highly sensitive magnetometers. To first order, a squid can be understood to behave like a single Josephson function (see Chapter 2.2.2) whose critical current I_c depends on the magnetic flux bias that is applied to the squid's inductive loop. The mutual inductance between the qubit and the squid loop then makes it possible to measure the qubit's magnetic flux state by probing the squid's critical current, i.e. it causes the critical current I_c to depend monotonically on the qubit state's position along the δ-axis.

To verify that the squid is operating correctly and to calibrate the readout procedure, it is useful to measure the squid's voltage response to an applied sinusoidal current bias (its I/V curve) as shown in Figure 8.1. The current bias in this case is done by placing a $10\,\mathrm{k}\Omega$ resistor in series with the squid and voltage biasing the two elements. Since the squid's resistance is much lower than $10\,\mathrm{k}\Omega$, this results in a current bias $I_{bias} \approx V_{bias} \,/\, 10\,\mathrm{k}\Omega$. Since the qubit circuit at this point is still unbiased, the qubit will have settled into a random magnetic flux state which corresponds to an unknown (and potentially fluctuating) flux bias applied to the squid. The resulting X-Y trace should look similar to the one shown in Figure 8.1b. The trace should be symmetric except for an offset on the critical

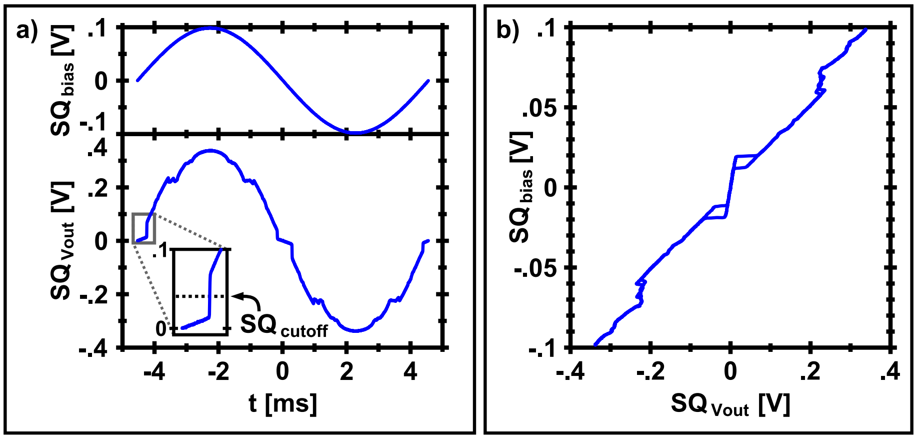

Figure 8.1: Squid I/V: Axis scales are given in terms of the values applied and measured outside the DR. These relate to the values at the squid as follows: $I \approx SQ_{bias}/10\,\mathrm{k}\Omega$ and $SQ_{Vout} \approx 1,000 \times V$. – a) X-T and Y-T plots: The squid is biased with an oscillating drive (top) to which it responds hysteretically (bottom). b) X-Y plot: The I/V response shows the hysteretic switching to the voltage state close to the center and the entry of the squid's normal conductance at 2Δ.

current due to the random flux bias. At sufficiently large biases the 2Δ-rise should be visible as explained in Chapter 4.1.3.

Just like a single junction, the squid can conduct a small amount of current without generating a significant voltage. Once the current bias exceeds this flux bias dependent critical current $I_c(\phi)$, the squid switches to the voltage state and begins generating a voltage. The critical current can therefore be measured by slowly increasing the bias and recording the point at which the squid's voltage jumps up. Since for the qubit readout only the point of the jump is relevant, but not the exact behavior of the response voltage, the jump can be encoded into a digital signal via the use of a comparator that indicates whether the squid's response voltage is smaller or larger than half of the jump voltage. This cutoff voltage SQ_{cutoff} is programmed into the PreAmp card (see Chapter 6.3.6) which uses it to generate a fiber-optic digital signal that indicates whether the squid is currently in the voltage state or in the supercurrent state.

To simplify the required control sequence, the Fast Bias cards (see Chapter 6.3.5) that are used to bias the squids provide a switchable resistor-capacitor filter that can set the ramp rate of the bias output. The GHz DAC (see Chapter 6.3.8) then simply needs to set the bias voltage to the desired final value and start a timer that waits for the PreAmp card to indicate the switching of the squid. The measured time to the switch t_{Switch} is then a monotonic measure of the critical current of the squid and thus of the δ-coordinate of the qubit's state. To shorten the sequence, this voltage ramp can be restricted to the range in which the critical current may fall to yield a bias sequence like the one shown as SQ_{bias} in Figure 8.2a.

8.2 Squid Steps

To understand how to properly bias the qubit for reset and operation, the squid can now be used to map out the position of the minima in the qubit's potential

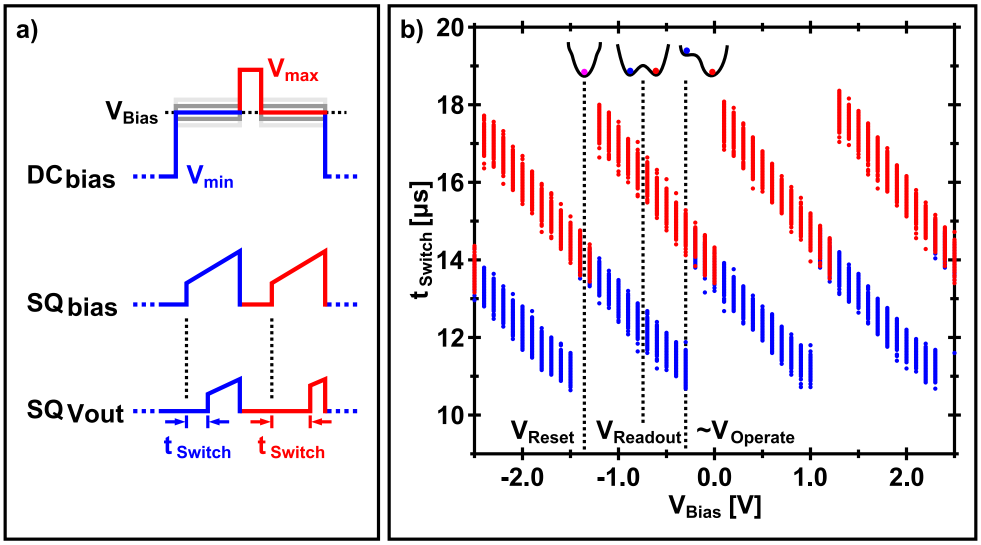

Figure 8.2: Squid Steps – a) Bias sequence: The qubit is biased to the minimum (maximum) voltage before being biased to V_{Bias}. The time along the squid ramp when the squid switches to the voltage state is plotted as the blue (red) dots and gives a measure of the position along the δ-axis of the qubit's potential minima. b) Data: As the qubit's potential is tilted, minima appear and disappear. The qubit is reset into a known minimum by biasing it to a point where only one minimum exists (V_{Reset}). It is operated in a shallow minimum ($V_{Operate}$) and it is biased with a maximal barrier between the left and the right minimum during the readout squid ramp to protect the state from accidental tunneling ($V_{Readout}$).

energy landscape (see Chapter 2.2.3) as a function of the applied flux bias DC_{bias}. For this, the qubit is held at a given bias point for a time much longer than T_1 so that the qubit state has time to decay into the lowest energy state of one of the local minima of the potential. The δ-coordinate of the minimum then corresponds to a certain magnetic flux in the qubit loop and thus to a certain critical current of the readout squid. This critical current is then probed via the described squid bias current ramp and the switching time t_{Switch} is recorded. This procedure is repeated many times each for various bias voltages V_{Bias} to generate a scatter plot of the observed qubit states as a function of qubit bias as shown in Figure 8.2b, which we call “Squid Steps”.

Since the choice of the local minimum into which the qubit state settles depends hysteretically on the biasing history, it is useful to take this data in two steps. First, the qubit is biased at the maximum negative bias V_{min} before bringing it to the bias point of interest V_{Bias}. Since the bias does not change instantaneously from V_{min} to V_{Bias}, this leads to the qubit state preferentially settling into the left-most local minimum at V_{Bias}. To map out the right-most local minima as well, the qubit is then biased at the maximum positive bias V_{max} before bringing it to V_{Bias}. This sequence is shown in Figure 8.2a.

The resulting scatter plot shows the location of the qubit potential's minima as a function of the qubit bias V_{Bias}. The qubit examined here shows bias regions where the potential has two stable local minima, e.g. for $-1.2\,\mathrm{V} \leq V_{Bias} \leq -0.3\,\mathrm{V}$, and regions where the potential only has one stable minimum, e.g. for $-0.2\,\mathrm{V} \leq V_{Bias} \leq 0.0\,\mathrm{V}$.

As the bias is increased, i.e. the potential is tilted more and more, the plot shows local minima appearing on the right (bigger δ, longer t_{Switch}), e.g. at $V_{Bias} \approx -1.2\,\mathrm{V}$, growing to their maximum depth, e.g. at $V_{Bias} \approx -0.1\,\mathrm{V}$, and becoming shallower again until they disappear on the left (smaller δ, shorter t_{Switch}), e.g. at $V_{Bias} \approx 1.0\,\mathrm{V}$. We call all points in the scatter plot that correspond to the same local minimum a "branch" and the maximum number of simultaneous local minima the "overlaps". The qubit here shows three full branches (and two partial ones) and has two overlaps.

For a qubit to be usable, it must have at least two overlaps so it can be measured (to hold the $|\,0\,\rangle$-state and the $|\,1\,\rangle$-state after tunneling) and it must show at least one full branch so it can be reset. Figure 8.3 shows a collection of non-optimal squid steps resulting from bad squid bias ramps, which should be correctable by readjusting the ramp parameters. But qubits that show no bias dependence of the scatter plot at all or have no regions of two or more overlaps usually must be pronounced dead.

The squid steps dataset (Figure 8.2b) now tells us how to reset and read out the qubit. The first step in desigining the operation sequence consists of

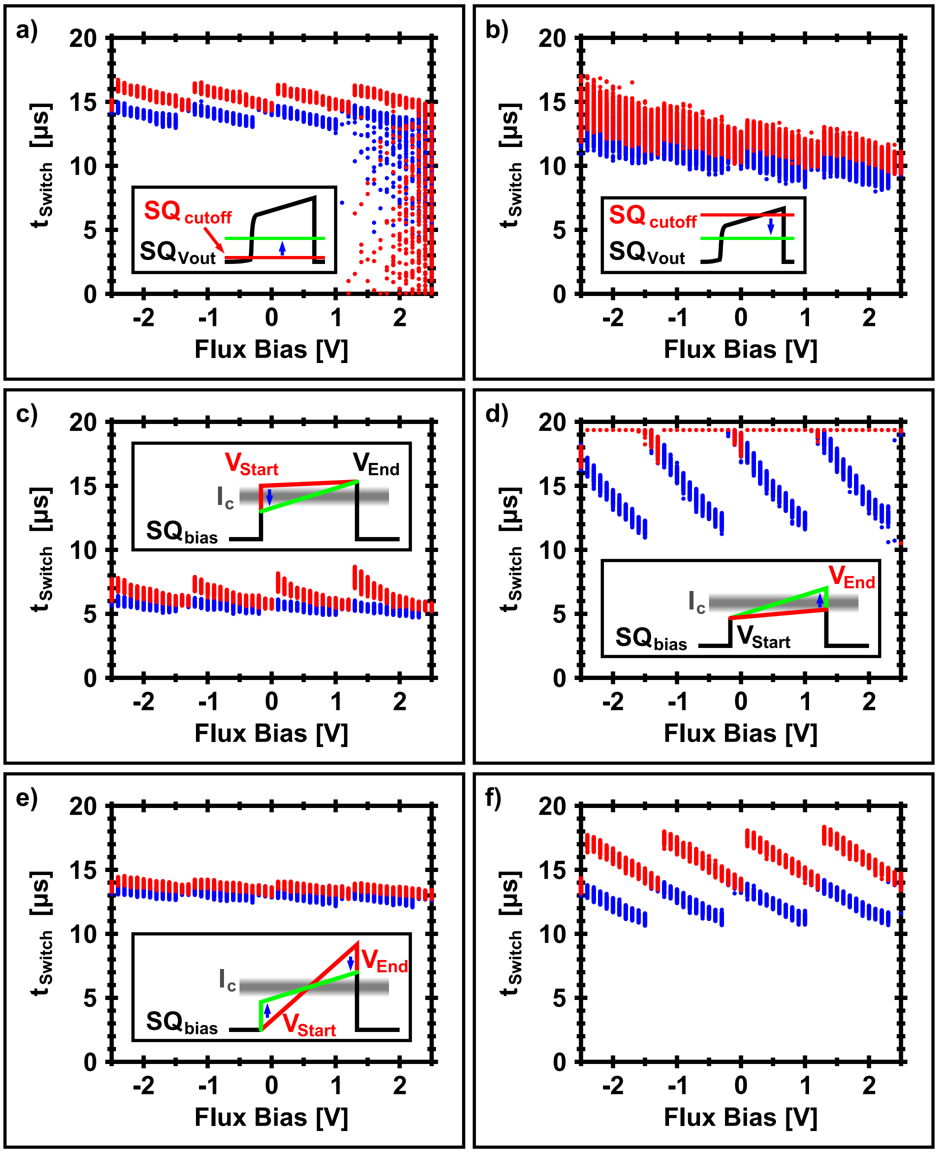

Figure 8.3: Squid Steps Failure Modes: Insets indicate problem (red) and correction (green). – a) Low SQ_{cutoff}: The voltage cutoff detecting the switching is set too low leading to false positives. b) High SQ_{cutoff}: The voltage cutoff is too high to detect the switch; instead it is crossed when the squid moves along the resistive part of the voltage branch. c) High V_{Start}: The bias crosses I_c during the ramp to V_{Start}. d) Low V_{End}: The bias barely/never crosses I_c. e) Large ramp: The bias is ramped too quickly through I_c. f) Functional squid: Perfect at last.

picking the "operating branch", i.e. the potential minimum in which the qubit state will reside during operations. Here, for example, we can choose the branch that extends from $V_{Bias} \approx -2.4\,\mathrm{V}$ to $V_{Bias} \approx -0.3\,\mathrm{V}$. The qubit is reset into this branch by biasing it to a point V_{Reset} at which the operating branch is the only minimum of the potential, e.g. $V_{Reset} = -1.35\,\mathrm{V}$. If the qubit is held at this bias for a time $t_{Reset} \gg T_1$, its state will decay into the operating branch with certainty.

To maximize the non-linearity during operation, i.e. the difference in energy spacing of the lowest levels in the operating minimum, the qubit is biased close to the "end" of the operating branch, i.e. to a point where the operating minimum becomes very shallow, for example around $V_{Operate} \approx -0.3\,\mathrm{V}$. This is necessary so that the lowest two energy levels in the minimum can be addressed exclusively as described in Chapter 3.3.

To measure the qubit, the excited state ($|\,1\,\rangle$) will be selectively tunneled out of the operating branch as described in Section 8.5 into the neighboring branch – here, the branch that extends from $V_{Bias} \approx -1.2\,\mathrm{V}$ to $V_{Bias} \approx -1.0\,\mathrm{V}$. The qubit is then biased to the point $V_{Readout}$ where the depth of the two potential minima is maximized, e.g. $V_{Readout} = -0.75\,\mathrm{V}$. This maximizes the barrier height between the two minima to reduce the chance of the qubit state tunneling between the minima by accident.

At this point, the squid ramp can be executed and the switching time t_{Switch} will indicate whether the qubit state remained in the operating branch during the tunneling measurement (here: $t_{Switch} \approx 12.4\,\mu$s) or whether it tunneled to the neighboring branch (here: $t_{Switch} \approx 16.2\,\mu$s). This completes the DC part of the biasing sequence as shown in Figure 8.4a.

If the qubit's ciritical current is too high and the qubit potential always has many stable minima, i.e. three or more overlaps, the qubit can not be reset into the operating branch as described above. In that case, the qubit is reset by alternatingly biasing it to the points where the operating branch is the left-most or right-most stable branch. This will dynamically destabilize all branches other than the operating branch and leads to a probability for finding the qubit in the target branch that increases with the number of reset cycles. Usually 3 to 5 cycles are sufficient to bring this probability to above 99.99%.

8.3 Step Edge

To maximize the non-linearity and thus the speed at which operations can be performed on the qubit without driving unwanted transitions, it is desirable to bias the qubit such that the operating minimum is as shallow as possible. We call the point at which the qubit ground-state ($|\,0\,\rangle$) in the operating minimum

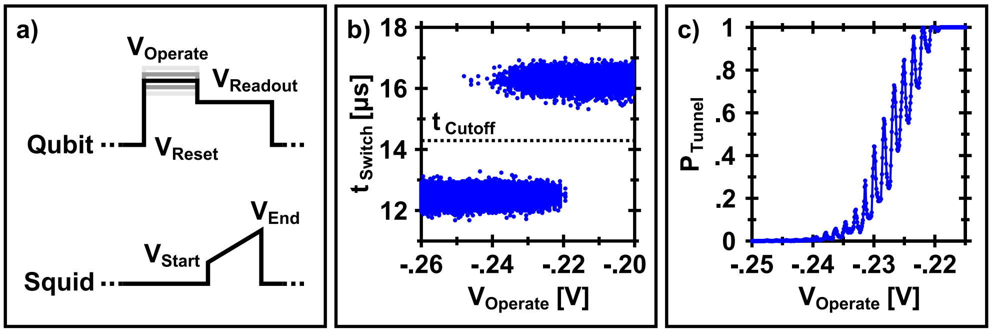

Figure 8.4: Step Edge – a) Bias sequence: The qubit is reset into the operating minimum and moved to the operating bias $V_{Operate}$ for a few μs to allow for any tunneling. After, the qubit is biased at the readout point and the squid is ramped to measure I_c. This sequence is repeated many times for each value of $V_{Operate}$. b) Switching data: As $V_{Operate}$ is increased, the qubit state tunnels from the operating minimum (lower branch) to the neighboring minimum (higher branch). c) Probability: If the switching times are sorted using a cutoff time between the switching distributions, the probability for the state to tunnel can be extracted as a function of $V_{Operate}$.

starts to tunnel to the neighboring branch the “Step Edge”. To determine the closest possible bias point to the step edge, it is useful to repeatedly run the qubit through its biasing cycle while varying its operating bias $V_{Operate}$ as indicated in Figure 8.4a. A scatter plot of the resulting switching times t_{Switch} as a function of the operating bias $V_{Operate}$ might then look like Figure 8.4b.

At this point it is useful to move to a representation of the data that describes the qubit state not in terms of a collection of switching times, but instead as the probability P_{Tunnel} of finding the qubit in the neighboring branch versus in the operating branch. This probability is obtained by defining a cutoff time t_{Cutoff} that separates the switching times into two groups, each corresponding to one branch. In the example here, all switching times $t_{Switch} < 14.2\,\mu$s correspond to the qubit state being in the operating branch during the squid ramp, while all switching times $t_{Switch} > 14.2\,\mu$s correspond to qubit states that have tunneled out of the operating branch into the neighboring branch, i.e. $t_{Cutoff} = 14.2\,\mu$s. The probability P_{Tunnel} is then defined as:

$$P_{Tunnel} = \frac{\#\text{ of runs with } t_{Switch} > 14.2\,\mu\text{s}}{\text{total }\#\text{ of runs}} \tag{8.1}$$

From this equation it is clear that, to obtain an accurate measurement of P_{Tunnel}, the experiment of interest needs to be repeated many times to collect enough statistical samples for t_{Switch}.

Plotting P_{Tunnel} versus the qubit's operating bias $V_{Operate}$ now gives a much more useful dataset like Figure 8.4c. In this plot we can easily see that the probability of the qubit state tunneling out of the operating minimum rapidly increases to unity once a certain bias threshold (here: $V^*_{Operate} \approx -0.24\,\mathrm{V}$) is crossed. Therefore it is necessary to keep the qubit bias safely below this threshold during qubit operation to prevent the states of interest from tunneling out of the operating branch before the actual measurement.

It is interesting to note that often this transition does not occur smoothly but instead shows many spikes. These are caused by the fact that the energy levels in the neighboring minimum are also quantized and therefore provide a discretized set of target states. If one of these target states happens to line up (in energy) with the qubit state in the operating minimum, the probability for the qubit state to tunnel is enhanced, leading to the observed resonance peak structure.

8.4 RF bias

This point marks a logical break in the bring-up sequence in the sense that the control sequence up to here contains only "slow" bias changes, i.e. on the order of microseconds. The following calibrations, on the other hand, are concerned with the high-frequency operations of the qubit which happen on time scales on the

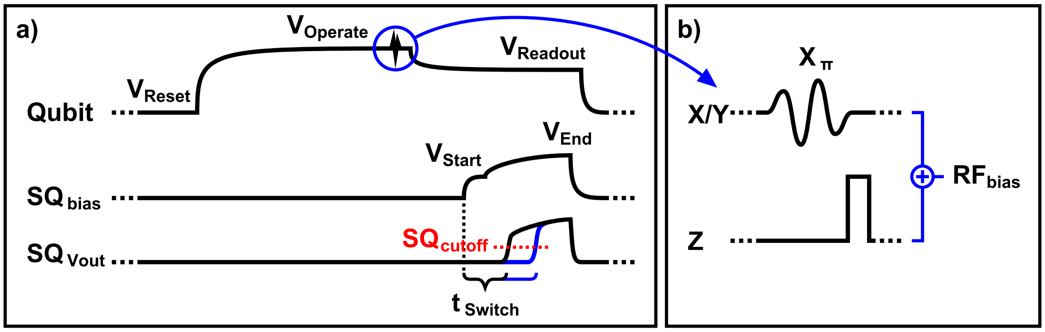

Figure 8.5: General Bias Sequence – a) DC bias sequence: The qubit is initialized and read out with slowly varying pulses with lengths on the order of 10 μs. b) RF bias: At the very end of the operating bias time, microwave pulses (X/Y) and fast bias pulses (Z) are used to perform the actual quantum operations. The entire RF bias sequence is (currently) only a few hundred nanoseconds long.

order of nanoseconds. In fact, the "action", i.e. the interesting qubit operations like rotations, coupling, etc., all happens in a tiny window at the very end of the time when the qubit is biased at its operating bias (see Figure 8.5).

Thus, to prevent unnecessary repetition of the invariant part of the bias sequence, the figures in the following chapters will, unless otherwise noted, focus on solely the RF_{bias} component of the control. The DC_{bias} and SQ_{bias}, as well as the SQ_{Vout} behavior should be understood to resemble the sequence shown in Figure 8.5. To clarify the sequence traces, the RF_{bias} components are split into X/Y operations corresponding to I/Q modulated microwave pulses and Z operations corresponding to fast flux bias pulses.

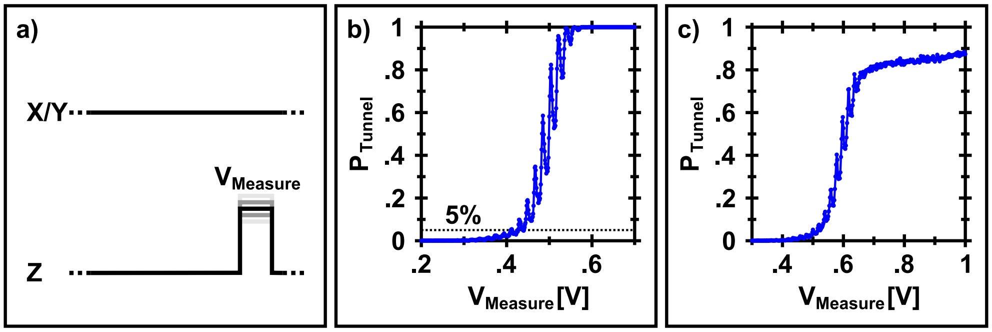

Figure 8.6: S-Curve – a) Bias sequence: The qubit is pulsed with a progressively larger measure pulse. b) S-curve: As $V_{Measure}$ is increased, the potential is tilted more and more until the qubit ground-state tunnels to the neighboring minimum. c) Short measure pulse: If the measure pulse is too short, the potential tilts back before the state has time to decay, leading to reduced visibility due to a chance for the state to retrap in the operating minimum [Zhang et al., 2006].

8.5 S-Curve

The first calibration of the RF_{bias} sequence will be the shape of the pulse that selectively tunnels the population of the excited states from the operating minimum on the left to the neighboring minimum on the right, but not the population of the ground-state. We will call this pulse the “measure pulse”. The simplest pulse that will achieve this is a rectangular (top-hat) pulse of calibrated length and amplitude. This pulse temporarily tilts the qubit potential to lower the barrier between the operating minimum and the neighboring minimum. The exact barrier height during the pulse will then determine the tunneling behavior of the different qubit states.

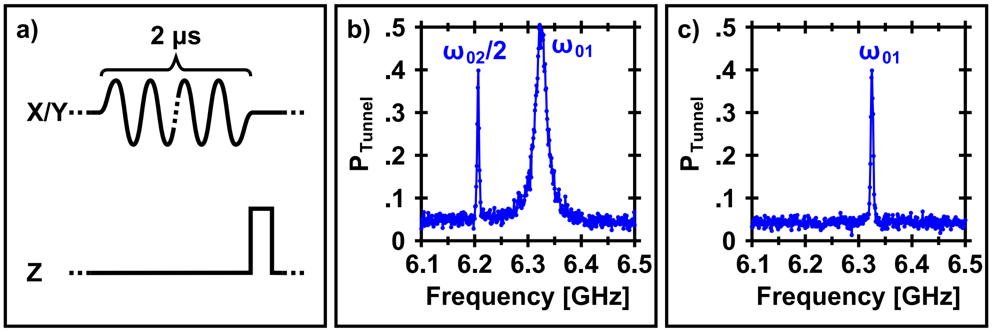

Figure 8.7: Spectroscopy – a) Bias sequence: A long microwave pulse of varying frequency is followed by a measure pulse. b) High power: If the microwave pulse is strong enough, it can drive higher transitions, like for example $|\,0\,\rangle \leftrightarrow |\,2\,\rangle$. c) Low power: At lower power (here: 20 dB less than b) all peaks but the $|\,0\,\rangle \leftrightarrow |\,1\,\rangle$ transition disappear.

For now, the length of the measure pulse is not too relevant and can be chosen around 20 ns. To calibrate the amplitude, one can measure the tunneling probability of the ground-state as a function of the measure pulse amplitude to obtain a plot like Figure 8.6b. For now, the point at which P_{tunnel} rises to about 5% (based on experience) can be used as an initial guess for the right measure pulse amplitude. A more exact value will be determined in a later experiment (Section 8.8). The data here shows similar resonant tunneling behavior at the step edge as was seen in Figure 8.4c.

8.6 Spectroscopy

Using this preliminarily calibrated measure pulse, it is possible to investigate the level structure in the operating minimum in more detail. Specifically, the resonance frequency of the transition from the ground to the first excited energy level is of particular interest as it is needed to perform logic operations on the qubit. This resonance frequency can be found by irradiating the qubit with long ($\gg T_1$) microwave pulses of varying frequency and then measuring the occupation of the excited levels in the qubit operating minimum using the measure pulse from above. Depending on the power of the drive, the data should resemble Figure 8.7b or 8.7c. If multiple peaks are visible in the data, the power can be reduced until only one peak remains, which should correspond to the qubit's transition frequency from the ground-state to the first excited state. The width of the peak gives a first hint at the relevant quality measures of the qubit, like the dephasing time T_φ. But since there are more sensitive measures for these, the only calibration value that needs to be taken away from this dataset is the resonance frequency ω_{01}.

8.7 Rabi Oscillation

The next step is to investigate the temporal response of the qubit to a microwave pulse like the one used above. For this, the qubit is driven with a micr-

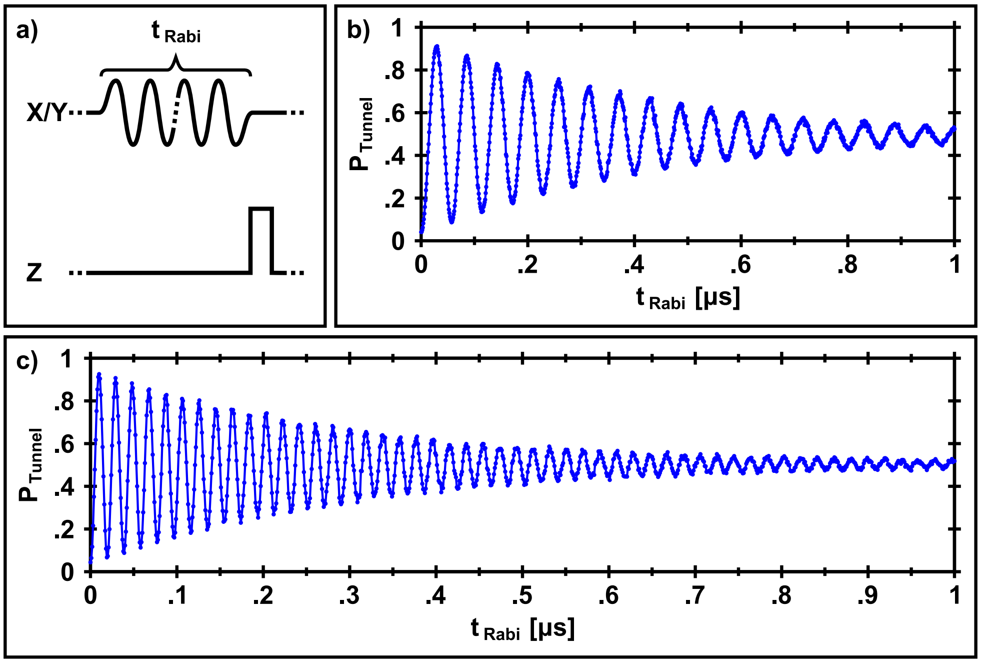

Figure 8.8: Rabi Oscillation – a) Bias sequence: A microwave pulse of varying length is followed by a measure pulse. b) Low power: The microwave pulse drives transitions between the qubit's $|\,0\,\rangle$ and $|\,1\,\rangle$ states. c) High power: A 3× stronger microwave pulse results in a 3× higher oscillation frequency.

wave pulse at the qubit resonance frequency of length t_{Rabi} followed by a measure pulse as indicated in Figure 8.8a. The probability of the qubit state tunneling out of the operating minimum is recorded as a function of the microwave pulse length to yield data as shown in Figures 8.8b and 8.8c. The data shows a decaying sinusoidal oscillation, which is caused by the qubit being excited into the first excited state and then returned to the ground-state via stimulated emission. This effect is called a Rabi oscillation and is the quantum two-level equivalent of a driven harmonic oscillator. The decay of the oscillation is caused by the qubit state relaxing back into the ground-state randomly on the timescales of the energy relaxation time T_1. The effect of this is different from the effect of damping on a driven harmonic oscillator. In the oscillator case, damping will cause the maximum oscillation amplitude to be limited, while in the case at hand, the relaxation causes a complete decay of the oscillation. This is due to the fact that the probability measured in this experiment is obtained from an ensemble measurement of many repetitions of the same experiment. In any single experiment, the occupation probability of the first excited state will oscillate forever at full amplitude since the period of the oscillation is much shorter than the energy relaxation time T_1. Every so often, though, while the qubit is in the excited part of the oscillation, the qubit will relax back into the ground-state and the oscillation will start over. This leads to a slow dephasing of the individual oscillations of the qubits in the ensemble with respect to each other and causes the population average to settle at 50% for long pulses.

In addition to this, the qubit's potential is subject to flux noise, i.e. a constant "jiggling". This leads to noise on the spacing of the energy levels, which causes the qubit resonance frequency to vary slightly. This is equivalent to random Z-rotations, i.e. phase shifts, being continuously applied to the qubit state. This effect is called dephasing and its timescale is measured by the dephasing time T_φ. Overall, the decay envelope of the Rabi oscillation follows the following formula:

$$A \propto e^{-\frac{3t}{4T_1} - \frac{t}{2T_\varphi}} \tag{8.2}$$

Since there are more direct measurements available to determine T_1 and T_φ separately and more accurately, there is only one value that needs to be taken away from this measurement. This value is the length of the microwave pulse that maximizes the probability of the qubit to be in the excited state. This pulse is called a "π-Pulse" [Lucero et al., 2008] and corresponds to a classical "NOT" gate. Since the frequency of the Rabi oscillation increases with higher amplitude of the microwave pulse, another way to calibrate the π-pulse is to choose a pulse length and to adjust the pulse amplitude to maximize the excited state probability. A sweep of the pulse amplitude results in a sinusoidal oscillation as well and is usually called a "Power-Rabi".

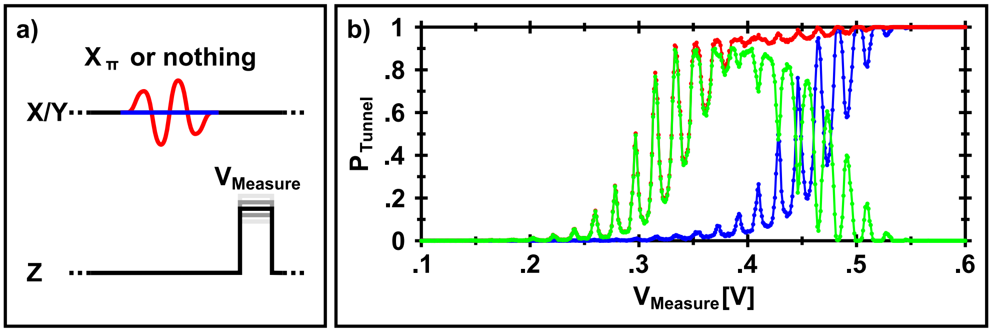

Figure 8.9: Visibility – a) Bias sequence: The qubit is either left in the $|\,0\,\rangle$ state (blue trace) or excited into the $|\,1\,\rangle$ state via a π-pulse (red trace) before it is measured with a pulse of increasing amplitude. b) S-curves: The blue (red) trace shows the tunneling behavior of the $|\,0\,\rangle$ ($|\,1\,\rangle$) state. The green trace shows the difference and gives the fidelity with which a measure pulse of given height can distinguish between the $|\,0\,\rangle$ and $|\,1\,\rangle$ states.

8.8 Visibility

Being able to prepare the qubit in either the ground-state (with just the initialization) or the excited state (with the π-pulse) allows us to fully optimize the measure pulse. This is done via a measurement of the "visibility", i.e. our ability to distinguish the excited state from the ground-state. It can be determined by taking the difference of the tunneling probability when the qubit is in the excited state and when the qubit is in the ground-state. This visibility can now be measured as a function of the measure pulse amplitude, length, and shape. Figure 8.9 shows the result of such a measurement as a function of measure pulse amplitude. The sequence that yields the curve for the ground-state is identical to the one

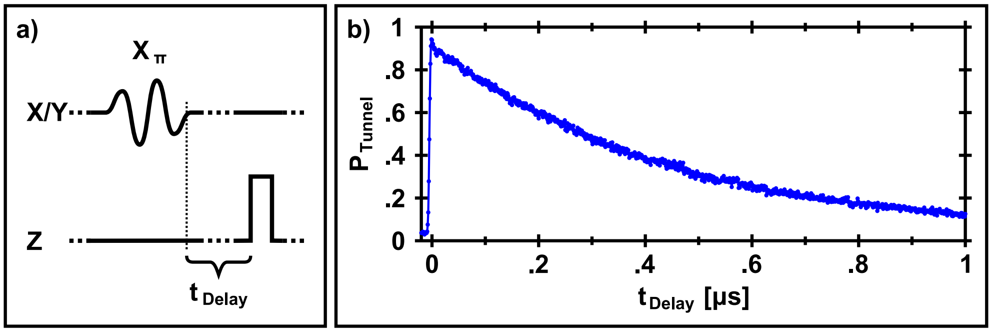

Figure 8.10: T_1 – a) Bias sequence: The qubit is excited into the $|\,1\,\rangle$ state with a π-pulse and measured after an increasing delay t_{Delay}. b) T_1: After the π-pulse, the qubit decays exponentially from the $|\,1\,\rangle$ state back to the $|\,0\,\rangle$ state.

from Section 8.5. The 5% point used in the initial calibration of the measurement pulse was chosen since, from experience (and from the math describing the tunneling of the two states) the maximum visibility usually lies around the point where the ground-state tunnels about 5% of the time. All parameters describing the measure pulse can now be adjusted to maximize the visibility.

8.9 T_1

Now that we have a calibrated way to prepare the qubit in the excited state and to measure its state as accurately as possible, we can start determining the qubit's intrinsic quality measures. The easiest quantity to measure is the qubit's energy relaxation time T_1. To determine it, all one needs to do is prepare the qubit in the excited state with a π-pulse and then measure its excited state population as a function of the delay between the π-pulse and the measure pulse. The data will look like Figure 8.10. Fitting the decaying part of the trace to the function $P(t) = P_{offs} + Viz * e^{-t/T_1}$, gives the quantity of interest: T_1. Since both the measurement visibility (Viz) and the $\sim 5\%$ offset (P_{offs}) are free parameters in this fit, the measurement process does not affect the obtained value of T_1. Even an imperfect π-pulse would only affect the value if part of the state is excited into the second excited state. Thus, this measurement yields a very robust number.

8.10 Ramsey

Unfortunately, measuring the dephasing time T_2 is less straightforward. This is due to the fact that the phase of the qubit's state only has meaning relative to an external clock source like the state of another quantum system. The phase of the qubit's state can also be measured by interfering it with the microwave drive. This is done by using a pulse of half the area of the π-pulse, i.e. a $\frac{\pi}{2}$-pulse, to excite the qubit into the equator of the Bloch sphere. There, the qubit is allowed to dephase for a time t_{Delay} and finally hit with another $\frac{\pi}{2}$-pulse to complete the rotation into the excited state before it is measured. As a function of t_{Delay}, the occupation probability of the excited state looks like Figure 8.11b. The problem with this measurement is that the decay envelope only gives a correct measure

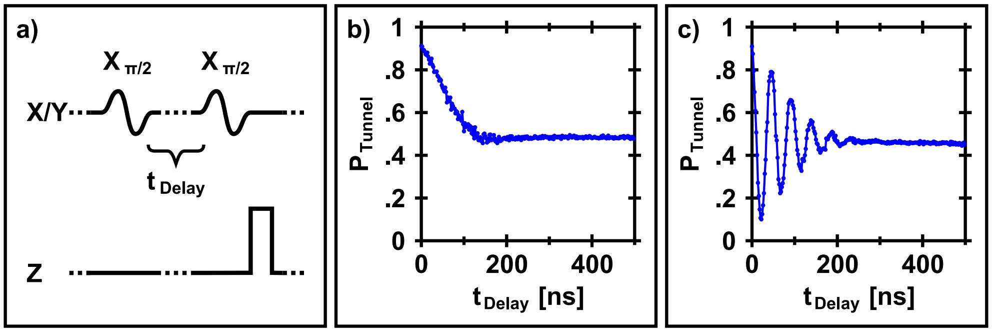

Figure 8.11: Ramsey – a) Bias sequence: A $\frac{\pi}{2}$-pulse excites the qubit into the equator of the Bloch sphere where it is allowed to dephase for a time t_{Delay}. A second $\frac{\pi}{2}$-pulse continues the rotation of the qubit state (ideally) to the $|\,1\,\rangle$ state. b) On resonance: As the qubit dephases its chance of making it to the $|\,1\,\rangle$ state decreases. c) Off resonance: If the qubit and the pulses are not at the same frequency, the qubit state spins around the equator of the Bloch sphere during the delay. Depending on the angle at which the second pulse "catches" the state, it either continues its rotation towards the $|\,1\,\rangle$ state or reverses it back to the $|\,0\,\rangle$ state. The oscillation frequency of the data should match the detuning, here 20MHz.

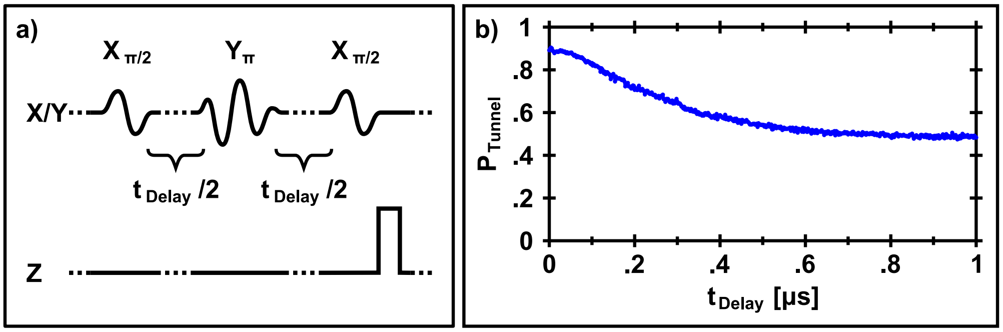

Figure 8.12: Spin Echo – a) Bias sequence: A π-pulse around the Y-axis that is placed in the middle of the Ramsey delay reverses the qubit state's precession angle causing it to undo the precession during the second half of the delay. This compensates for slow bias drifts and off-resonant pulses. b) Spin echo: The qubit's phase information decays following a gaussian shape.

of T_2 if the frequency of the microwave pulse is exactly on resonance with the transition frequency of the qubit. If the microwave drive is detuned, the qubit will not only dephase, but also precess in the equator of the Bloch sphere, leading to a trace like Figure 8.11c. The frequency of the oscillation is equal to the detuning of the microwaves from the qubit. Thus, it is not necessarily clear, how much of the "decay" in Figure 8.11b. is due to T_2 and how much is due to the beginning of a slow oscillation caused by a very slight detuning of the microwaves.

8.11 Spin-Echo

The detuning effect on the measurement of the Ramsey trace can be eliminated by a trick called "Spin Echoing". It works by inserting a π-pulse with a 90°

phase shift into the middle of the delay time. This pulse effectively inverts the accumulated phase of the qubit state in the equator, such that any precession in the first half of the delay time undoes itself in the second half of the delay time. This removes the effect of a potential microwave detuning onto the data and yields a better estimate of T_2 if the data is fit to the function $P(t) = P_{offs} + Viz * e^{-t/T_2}$. Even if the qubit's resonance frequency was perfectly stable, the trace measured in this experiment would still decay to around 50% due to the fact that T_1 decays the qubit state from the equator back into the ground-state. Since the ground-state does not have any phase associated with it, this decay also erases phase information. T_2 is defined as the timescale on which the qubit loses its phase information and thus, this T_1 effect does not need to be removed from the fit, but instead does contribute to the real value of T_2. This leads to the fact that T_2 can never be larger than $2T_1$. There also exists a quantity that measures the loss of phase information due to only the instability in the qubit's resonance frequency. This quantity is called T_φ and can be calculated from:

$$\frac{1}{T_2} = \frac{1}{2\,T_1} + \frac{1}{T_\varphi}. \tag{8.3}$$

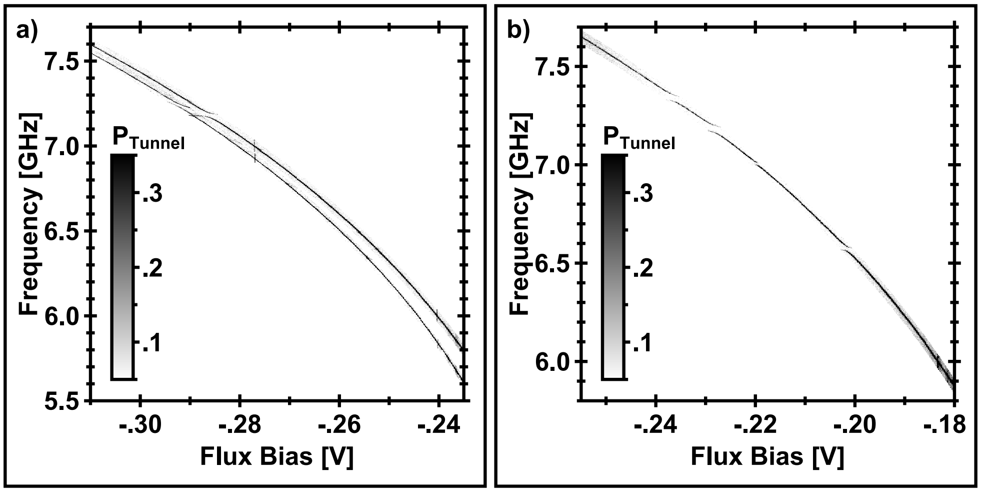

Figure 8.13: Fine Spectroscopy – a) High power: The $|\,0\,\rangle \leftrightarrow |\,1\,\rangle$ and the $|\,0\,\rangle \leftrightarrow |\,2\,\rangle$ transition are visible. This qubit couples to very few two-level states. b) Low power: Only the $|\,0\,\rangle \leftrightarrow |\,1\,\rangle$ transition is visible. This qubit couples to more two-level states.

8.12 2D-Spectroscopy

The final scan that is part of our standard qubit bringup sequence is called "2D-Spectroscopy". It is probably the single most useful calibration scan we do due to the large amount of information it provides. The scan is simply the 2D extension of the above mentioned spectroscopy scan as a function of operating bias. The data is usually drawn as a 2D color plot and should resemble Figure 8.13a or b depending on whether the microwave pulse power is high enough to excite the two-photon $|\,0\,\rangle \rightarrow |\,2\,\rangle$ transition. The dependence of the response frequency ω_{01} on the operating bias follows a quadratic equation almost exactly. The two-photon excitation behaves similarly. The fits of ω_{01} and ω_{02} versus the flux bias ϕ can be used to extract the actual values of the qubit's inductance, capacitance, and critical current.

The 2D-Spectroscopy scan also shows avoided level crossings that correspond to the qubit interacting with a two-level defect in the tunnel barrier which responds at a certain frequency [Cooper et al., 2004, Neeley et al., 2008a]. Since we believe that these defects are the primary source of energy relaxation in our qubit [Simmonds et al., 2004, Martinis et al., 2005], one should choose an operating bias that places the qubit far (in frequency) from one of these defects.

Chapter 9

Coupled Qubit Bringup

The exact bringup sequence for running multiple qubits can vary greatly depending on the coupling scheme. Nevertheless, there are several concerns that apply to all coupling schemes. These will be examined first.

9.1 Controlling Multiple Qubits

9.1.1 Control Synchronization

Most multi-qubit sequences require the relative timing of operations on different qubits to be tightly controlled. To achieve this, the control hardware needs to provide a way to synchronize the control channels on the different qubits with respect to each other. For the purposes of scaling to a large number of qubits, it is desirable to modularize the control hardware (and software) such that the different channels are independently deployable copies of each other. Each channel then needs to provide a way to synchronize its signal stream to all other channels. This synchronization consists of two components: Not only do the signal streams need to start at the same time $t = 0$, they also need to progress at the same rate.

The definition of a common starting time $t = 0$ is usually achieved by a trigger pulse for which the different channels wait before playing back their signal sequence. For this, the trigger pulse needs to be distributed to all channels. This can be done either by splitting the output of a single trigger source and distributing it to all channels or by a so-called "daisy chain" in which each channel forwards the trigger signal to the next. Usually, each channel needs to provide facilities to shift its specific definition of $t = 0$ by an offset Δt that compensates for any delays in the arrival of the trigger pulse or any delays in the delivery of the channel's output signal to the respective qubit. The calibration of this delay is best done with experiments run on the qubits. The type of experiment needed depends on the involved channels and the type of coupling element used and will be discussed below.

Actively synchronizing the rate at which the different channels play back their signals is necessary since affordable clock sources do not natively provide the desired accuracy. Specifically, the inter-channel phase jitter in the clock signals is of significant concern for reliable qubit operation. This is due to the fact that for rotations around axes in the X/Y-plane of the Bloch sphere, the angle of the rotation axis is determined by the phase of the signal.

The common approach is to synchronizing the clocks with the use of a 10 MHz reference signal to which each channel locks a VCO (Voltage Controlled Oscillator) circuit to generate the clock frequency it needs for its operation. To achieve best possible phase locking between the microwave control signals on different channels, we currently use a single carrier signal that is split and distributed to the different channels. But this approach is only feasible for a small number of channels. Beyond that, the carrier signal generators will need to be chosen and synchronized with phase jitter in mind.

9.1.2 Flux Bias Crosstalk

Unless the qubits' integrated circuit is laid out perfectly, the magnetic field created by the flux bias coil of one qubit will also be seen by all other qubits, although to a much lesser degree. This leads to changes in the actual bias seen by the qubits and thus to changes of the operating parameters (reset bias, resonance frequency, etc.) found by calibrating the qubits independently from each other. Therefore, it is necessary to repeat the single qubit bringup experiments that lead to the parameters while putting all qubits through their motions as if the final sequence was run. This allows for the DC component of the biasing crosstalk to effectively be calibrated away. This will work “out of the box” if all qubits share the same timing for the biasing pulses. The repeated bringup will then simply yield slightly adjusted bias values that correct for crosstalk automatically. If the qubits do not share the same pulse timing, corrective bias pulses might need to be added into the sequence manually.

The flux bias crosstalk also concerns the RF component of the bias, i.e. the microwave signals. The effect of this can also be removed by simply driving every qubit with a pulse that exactly cancels the leaked signal. For this cancellation to work correctly, not only the amplitude of the canceling pulse needs to be adjusted, but also its phase. But since the RF components of the bias crosstalk are usually very small in our samples ($\lesssim -20\,\mathrm{dB}$), they can often simply be ignored.

9.1.3 Readout Squid Crosstalk

Just like the flux bias line of one qubit can potentially couple to all qubits on the chip, so can the different qubits’ squids. Even though the coupling strength of this crosstalk is usually much smaller, it still constitutes a major problem as the squids create a large voltage step when they switch to the voltage state. If, at that time, another squid is close to its critical current, this step will usually trigger it to switch as well. This will lead to a reliable measurement of I_c only

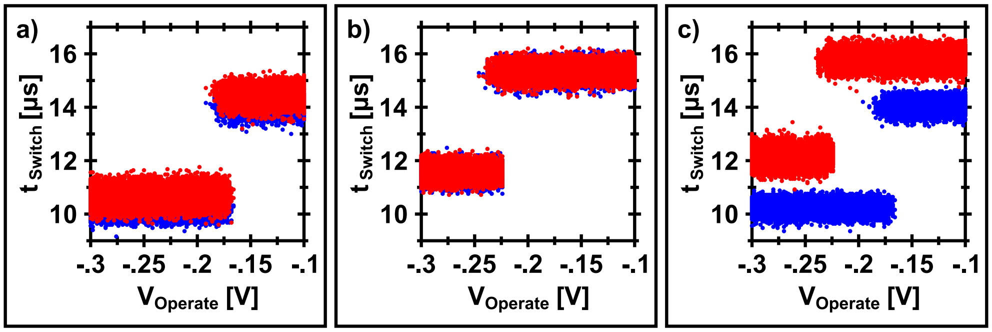

Figure 9.1: Effect of Squid Crosstalk on Step Edge – a) "Blue" switches "Red": A switch in the "blue" qubit's squid causes the squid of the "red" qubit to switch, moving the "red" step edge on top of the "blue". b) "Red" switches "Blue": A switch in the "red" qubit's squid causes the squid of the "blue" qubit to switch, moving the "blue" step edge on top of the "red". c) No crosstalk: Both step edges occur at their expected bias points with 0% tunneling before and 100% tunneling after.

for the squid that switches first. To counteract this effect, it is necessary to ramp the squids one after the other through their switching region while keeping the respective other squids biased far away from their critical current.

For this to work reliably, it is necessary to bias all qubits such that the potential barrier between their operating minimum and the neighboring minimum (minima) is as large as possible to keep the measurements locked in until it is their turn to be read out. Depending on the maximum possible barrier heights and the strength of the squid crosstalk, not all ramp-orders will be guaranteed to work. It is therefore necessary to run simple test experiments (e.g. step edges as shown in Figure 9.1) to determine the right ramping order, if one can be found. If it is not possible to read out all qubits correctly, the squids and coupling strengths need to be redesigned to reduce the crosstalk.

9.2 Always-On Capacitive Coupling

Qubits that are coupled with a simple capacitor suffer the most from the above described crosstalk effects. In fact, if the coupling is too strong ($\gtrsim 30\,\mathrm{MHz}$), it might not be possible at all to find an order in which to ramp the squids such that their switching does not randomize the measured state of all qubits.

9.2.1 Measurement Crosstalk and Timing

One of the biggest issues for a capacitively coupled qubit system is the problem of measurement crosstalk [Kofman et al., 2007, McDermott et al., 2005]. Measurement crosstalk is the process by which the tunneling of one qubit causes a tunneling of other qubits even if they were in the $|\,0\,\rangle$-state. This happens because the tunneling process leaves the tunneled qubit in a highly excited state in the neighboring minimum. As the qubit decays to the ground state of this minimum, it radiates photons of progressively decreasing frequency into the circuit. These photons can couple via the coupling capacitor to the other qubits and excite them into higher states in the operating minimum. If this excitation

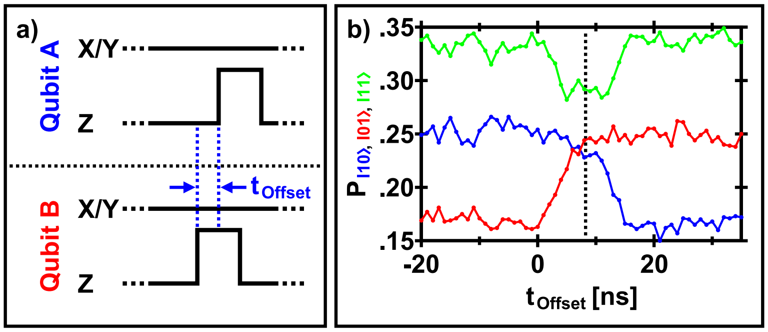

Figure 9.2: Measure Pulse Timing – a) Sequence: Both qubits are measured with a pulse that tunnels the $|\,0\,\rangle$ state 50% of the time. The measure pulse on the second qubit is shifted in time by t_{Offset} relative to the first qubit. b) Crosstalk: When the measure pulses reach the qubits at the same time (dashed line), the measurement crosstalk is minimized leading to a dip in $P_{|\,11\,\rangle}$.

happens before the measurement of these qubits is over, these states will tunnel as well. For a two-qubit system, this leads to a $|\,01\,\rangle$-state or a $|\,10\,\rangle$-state to be misidentified as a $|\,11\,\rangle$-state. The $|\,00\,\rangle$-state is not affected by this problem.

This crosstalk does offer an opportunity, though, in that it provides a way to synchronize the timing of the measurement channels between the different qubits. The method is based on the fact that the crosstalk can only affect qubits until the point when their measurement is complete. This temporal asymmetry can be used to adjust the relative timing of the different channels. For this, an S-Curve experiment is used to independently calibrate a measurement pulse for each qubit that yields a $|\,0\,\rangle$-state tunneling probability of about 50%. As this tunneling probability is a purely classical probability, in a system without measurement crosstalk, a simultaneous application of such measure pulses would yield all possible outcomes with equal probability. For a two-qubit system, this would mean that the readout would yield the states $|\,00\,\rangle$, $|\,01\,\rangle$, $|\,10\,\rangle$, and $|\,11\,\rangle$ all with 25% probability. In a situation with crosstalk where qubit 1 (2) was measured significantly before qubit 2 (1) the probability of measuring $|\,10\,\rangle$ ($|\,01\,\rangle$) would be reduced and $|\,11\,\rangle$ would be increased. Thus, if the probabilities are measured as a function of the delay t_{Offset} between the measure pulses, one obtains a plot like Figure 9.2.

The point on this plot where the probability of $|\,11\,\rangle$ is minimized will then correspond to the point where the qubits are measured at the same time. The difference at this point between 25% and the measured probability of $|\,11\,\rangle$ captures the amount of residual measurement crosstalk. It is possible to minimize this residual crosstalk by carefully shaping the measure pulses in a way that maximizes each qubit's visibility while minimizing crosstalk. A saw-tooth shape seems to work well for this.

9.2.2 Spectroscopy

In the same way that the frequency of the microwave bias needs to be matched to the qubits resonance frequency in order to drive transitions, to get several qubits to couple via a simple capacitor, they need to be biased such that their resonance frequencies are the same. To achieve this, the most useful scan is the simple

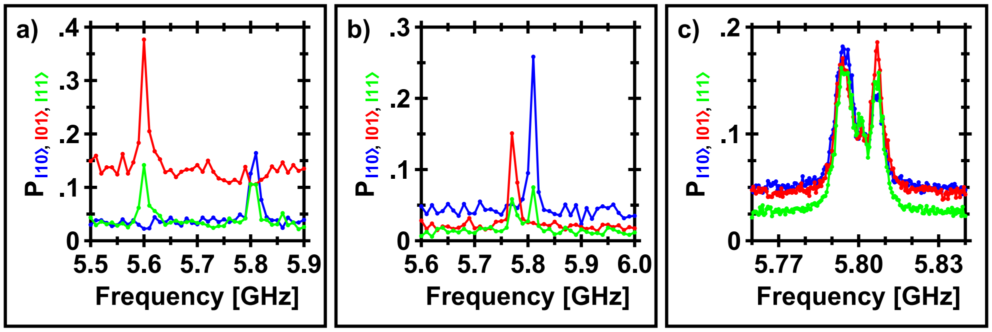

Figure 9.3: Spectroscopy of Coupled Qubits – a) Far off: Initially the flux bias crosstalk will probably place the qubits off resonance. b) Closer: As the flux bias is corrected, the resonance peaks move closer. c) Splitting: When the qubits are on resonance, the coupling causes them to split.

one-dimensional spectroscopy scan introduced above. If it is run on all qubits simultaneously, it will show each qubit's resonance peak in the presence of all other qubits' biases. If the operating biases of all qubits are then adjusted to set the resonance frequency of all qubits to the same value, the qubits will begin to couple. This can sometimes be seen in a splitting of the response peak in the spectroscopy sweep, as shown in Figure 9.3c.

9.2.3 Swaps

The next step is to examine the time dynamics from the qubits' interaction. This can be achieved with a sweep that is essentially a multi-qubit T_1 experiment, except that only one of the involved qubits is prepared in the $|1\rangle$-state with a π-pulse, while all other qubits remain in the ground state. In addition to the

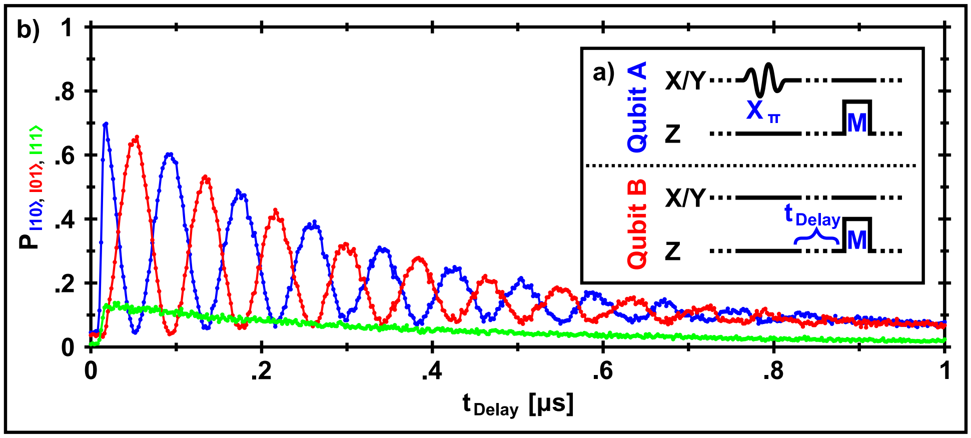

Figure 9.4: Capacitive Coupling Swaps – a) Sequence: The first qubit is excited into the $|1\rangle$ state with a π-pulse (X_π). The qubits are allowed to interact for a time t_{Delay}. After, both qubits are measured (M). b) Swaps: The excitation swaps between the qubits, leading to a final state that oscillates between $|10\rangle$ and $|01\rangle$. Measurement crosstalk and the measurement error in identifying the $|0\rangle$ state causes a relatively large probability for measuring $|11\rangle$. As a function of time, T_1 decays the signal and relaxes the qubits into the $|00\rangle$ state.

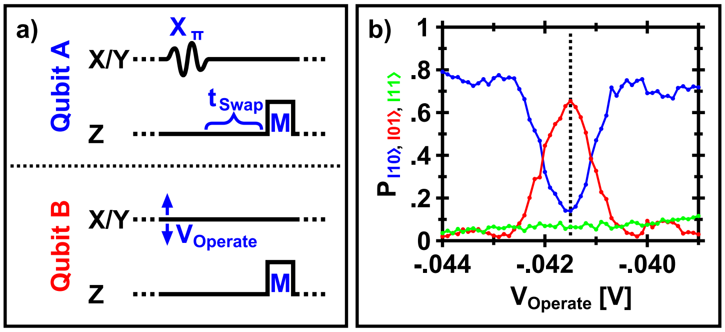

Figure 9.5: Capacitive Coupling Resonance Calibration – a) Sequence: The first qubit is excited into the $|1\rangle$ state with a π-pulse (X_π). The qubits are allowed to interact for a time t_{Swap} that should result in a swap when the qubits are on resonance. After, both qubits are measured (M). The flux bias $V_{Operate}$ of the second qubit is varied to find the optimal bias point. b) Data: At the point where the qubits are exactly on resonance (dashed line), the swapping is maximized.

usual T_1 decay, the energy of the excited qubit then gets transferred back and forth among the qubits via the coupling capacitor [McDermott et al., 2005]. For two qubits, this leads to a simple swapping of the excitation between the qubits as shown in Figure 9.4, much like the oscillatory energy transfer in a system of coupled pendulums.

9.2.4 Resonance Calibration

For small coupling capacitors (< 10 MHz), it is extremely important to make sure that the qubits are biased exactly on resonance in order to achieve the maximum swap amplitude. The quickest way to calibrate the biases needed is by repeating the Swaps measurement with t_{Delay} set to achieve one full swap (~ 50 ns

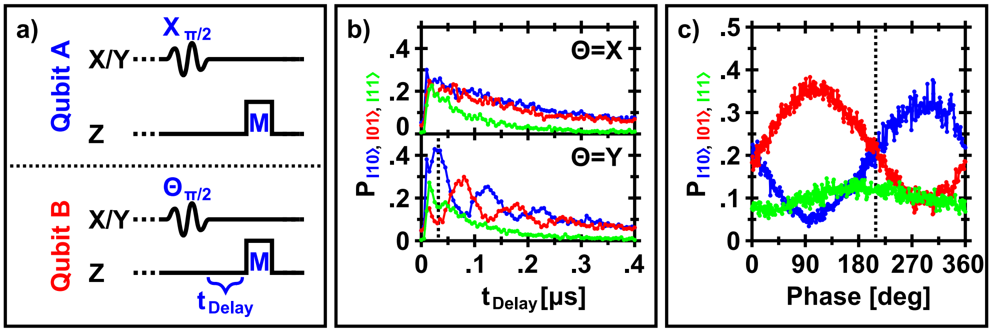

Figure 9.6: Capacitive Coupling Phase Calibration – a) Sequence: Both qubits are hit with a $\frac{\pi}{2}$-pulse ($X_{\pi/2}$, $\Theta_{\pi/2}$) to generate the state $|\,00\,\rangle + |\,10\,\rangle + e^{i\,\theta}|\,01\,\rangle + e^{i\,\theta}|\,11\,\rangle$. The qubits are allowed to interact for a time t_{Delay} before they are measured (M). b) Time trace: An X-pulse on both qubits results in an eigenstate of the coupling that simply decays, while a pulse of different phase on the two qubits results in a state that will undergo a swap operation. This swap is maximized at t_{Swap} (dashed line). c) Phase trace: If t_{Delay} is set to t_{Swap}, the phase θ of the pulse on the second qubit can be swept to find the phase offset between the qubits. The point where the curves cross (dashed line) gives this offset. Which of the crossings is the right one depends on the definition of the coordinate system used in the Bloch sphere.

in Figure 9.4) for slightly different operating bias values for the qubit that was initialized in the $|\,0\,\rangle$-state (here: Qubit B). The data will look like Figure 9.5b, showing a maximized swap at the bias that places the qubits exactly on resonance. If t_{Delay} was not picked exactly right, the maximal swap in this dataset might be less than the maximum achievable swap, but it should still give the right bias calibration.

9.2.5 Phase Calibration

The trickiest calibration is that of the relative phase of the microwave signals as seen by the qubits. Due to the high frequency of the drive ($\sim 6\,\mathrm{GHz}$), even the slightest difference in the electrical length of the wiring can lead to significant phase shifts. These need to be calibrated by interfering phase-sensitive states created in the qubits via the coupling capacitor. The easiest way to do this is to simultaneously drive each qubit with a $\frac{\pi}{2}$-pulse into the state $|\,0\,\rangle + e^{i\alpha}|\,1\,\rangle$ and observe the qubits' interaction (Figure 9.6b). If all qubits are driven with the same phase, the resulting bell state $(|\,01\,\rangle + |\,10\,\rangle)$ will be an eigenstate of the coupling Hamiltonian and thus show only the expected T_1 decay. If the qubits are driven with different phases, the resulting states (e.g. $|\,01\,\rangle + i|\,10\,\rangle$) are not eigenstates and thus will show oscillations similar to the ones seen in the Swaps experiment.

The problem with this calibration is that driving the qubits at the exact opposite phase will also lead to an eigenstate $(|\,01\,\rangle - |\,10\,\rangle)$ that does not show swaps. This ambiguity reflects the freedom in choosing the exact representation of states on the Bloch sphere in terms of whether a left-handed or right-handed coordinate system is used or (equivalently) whether then $|\,0\,\rangle$-state is placed at the South or North pole of the sphere. The choice of which swapping minimum corresponds to a phase difference of 0° rather than 180° will then need to be made by predicting the direction of the swaps (does $|\,01\,\rangle$ or $|\,10\,\rangle$ go up first?) for a drive with a 90° phase shift.

9.2.6 "Controllable Coupling" via Bias Changes

Since the qubits will only couple to each other if they are biased to have the same resonance frequency, it is possible to turn the coupling on or off to a certain degree by changing the qubits' biases over the course of their RF_{bias} control sequence. The coupling strength hereby depends on the on-resonance coupling strength g and the detuning Δ via:

$$C = \frac{g}{\sqrt{g^2 + \Delta^2}} \tag{9.1}$$

To calibrate the amplitude of the bias pulse that sweeps the qubits onto resonance and thus turns on the coupling, the same method can be used as described above under Resonance Calibration. The important thing to realize is that these bias pulses also perform Z-rotations on the qubit that can be quite large (many full revolutions). These need to be understood and taken into account when determining the phase of any following microwave pulses. For this to work, it is important to ensure a repeatable bias pulse shape as run-to-run differences will introduce phase errors.

9.3 Resonator Based Coupling

The measurement crosstalk described in Section 9.2.1 can be extremely detrimental to the quality of the final data for a coupled qubit experiment. Especially for an experiment that attempts to violate Bell's inequality, measurement crosstalk is unacceptable as it actively introduces correlations into the dataset, the very thing that the experiment is trying to quantify [Kofman and Korotkov, 2008b]. Therefore, it is necessary to develop coupling schemes that prevent measurement crosstalk as much as possible. One such scheme is based on placing a resonator between the qubits as shown in Figure 2.5b. This resonator will effectively act as a band-pass filter for the coupling between the qubits. Thus, during the decay of the tunneled state, only photons at the frequency of the resonator can couple to the other qubit. Since this frequency will not be on resonance with that other qubit, there will be no unwanted excitations, effectively eliminating the problem of measurement crosstalk.

For the purpose of understanding the following sections, and even for the implementation of the Bell experiment described in Chapter 11, the resonator can be understood simply as a third qubit that is capacitively coupled to each of the two real qubits. This works here, since there is always only one photon present in the circuit during coupling operations, which prevents the resonator from ever

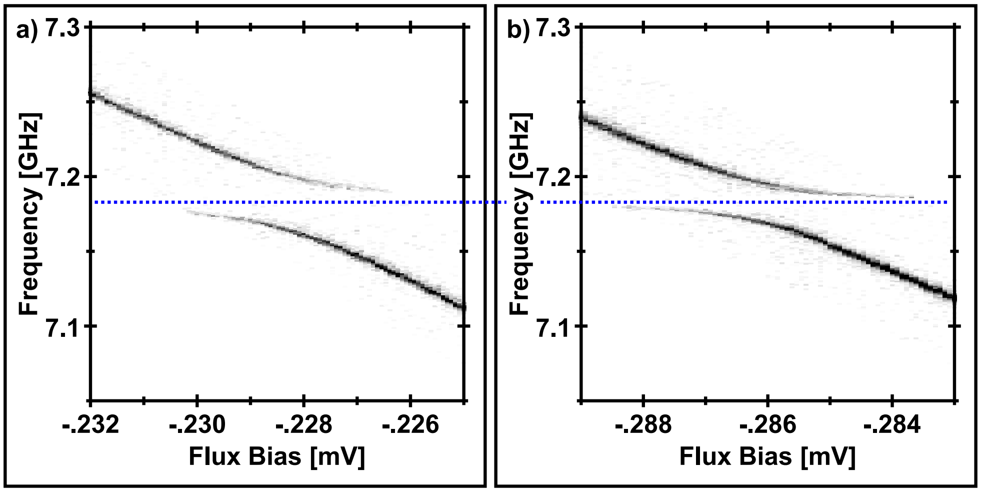

Figure 9.7: Fine Spectroscopy of Resonator Coupling – a) First qubit: The qubit's spectroscopy shows a splitting at around 7.18 GHz due to the coupling to the resonator. The width of the splitting matches the 40 MHz coupling strength. b) Second qubit: The splitting is located at the same frequency, but is smaller due to the 27 MHz coupling strength.

being excited into the $|\,2\,\rangle$ state. In general, though, since the energy levels in the resonator are equally spaced, this equivalence does not hold, as the resonator will be able to store more than one photon by populating its higher excited levels. For example, while a $|\,11\,\rangle$ state of two coupled qubits does not evolve, a $|\,11\,\rangle$ state of a qubit-resonator system will result in the system swapping photons to transition to the $|\,02\,\rangle$ state and back [Hofheinz et al., 2008].

9.3.1 2D-Spectroscopy

To couple a set of qubits via a resonator, one needs to find a way to bias these to the same frequency as the resonator. Due to variations in the fabrication, the resonance frequency of the resonator might not be known exactly and thus needs to be measured. Since, in the simplest design, the resonator cannot be read out directly, this is best done with the 2D-Spectroscopy experiment described in Chapter 8.12. In the dataset, the resonator will show up as a splitting, just like the two-level states. The size of the splitting will depend on the coupling strength between the qubit and the resonator. Since this coupling strength is often comparable to the coupling strength between the qubit and a two-level state, it might not be immediately obvious which splitting corresponds to the resonator. In most cases, though, due to the random frequencies of the two-level states, looking at the 2D-Spectroscopy of both qubits and finding a splitting at the same frequency resolves this ambiguity.

9.3.2 Swapping a Photon into the Resonator

Since the resonator can accept many photons from an on-resonant microwave drive, it is not possible to perform single qubit operations while the qubits are on resonance with the resonator. Thus, the qubits need to be kept off resonance and

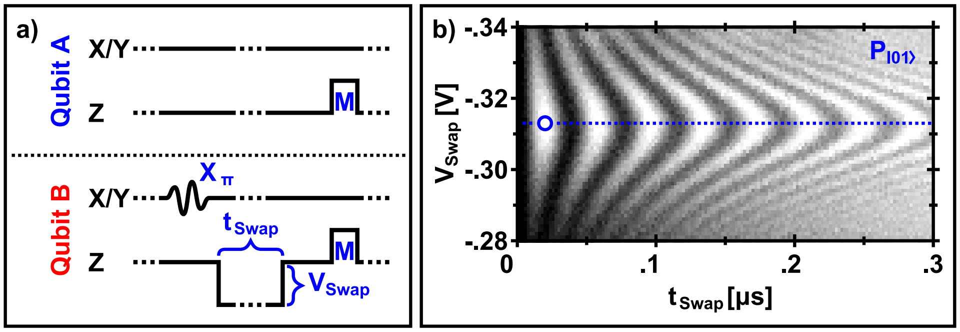

Figure 9.8: Swapping Photon into Resonator – a) Sequence: The second qubit is excited into the $|\,1\,\rangle$ state with a π-pulse (X_π). The qubit is then hit with a bias pulse of height V_{Swap} that sweeps its resonance frequency closer to the resonator for a time t_{Swap}. Both qubits are measured (M). b) Swaps: The excitation in the qubit swaps between the qubit and the resonator. If the bias pulse places the qubit exactly on resonance with the resonator (dashed blue line), the swaps are slowest and have the largest amplitude. At exactly the right time (blue circle), the interaction causes the excitation to swap to and remain in the resonator.

swept on resonance only when needed. The easiest way to calibrate this sweep is by preparing one qubit in the $|\,1\,\rangle$-state via a π-pulse and then adjusting the length and amplitude of the bias pulse to transfer as much of the excitation as possible out of the qubit and, in this case, into the resonator. A 2D-sweep of $P|\,01\,\rangle$ versus bias pulse amplitude versus bias pulse length could look like Figure 9.8b. The minimum in this dataset describes the optimal swap pulse.

9.3.3 Retrieving the Photon from the Resonator

This can now either be repeated for all qubits, or the other qubits' bias pulses can be calibrated to retrieve as much of the excitation as possible out of the

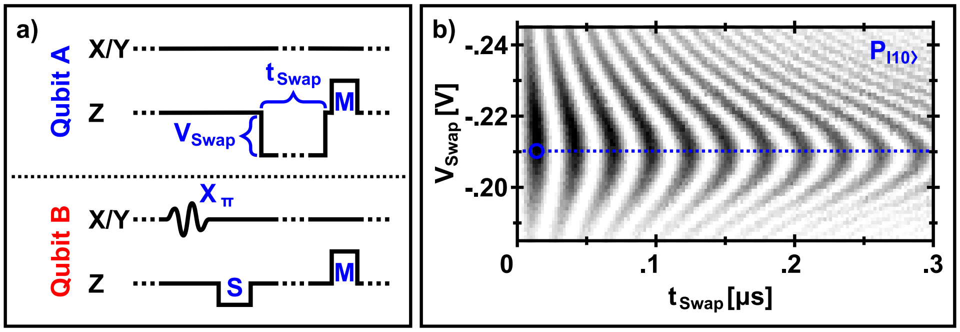

Figure 9.9: Swapping Photon out of Resonator – a) Sequence: The second qubit is excited into the $|1\rangle$ state with a π-pulse (X_π). The qubit is the brought on resonance with the resonator for the time needed to swap the excitation into the resonator (S). The first qubit is then hit with a bias pulse of height V_{Swap} that sweeps its resonance frequency closer to the resonator for a time t_{Swap}. Both qubits are measured (M). b) Swaps: The excitation in the resonator swaps between the first qubit and the resonator. If the bias pulse places the qubit exactly on resonance with the resonator (dashed blue line), the swaps are slowest and have the largest amplitude. At exactly the right time (blue circle), the interaction causes the excitation to swap to and remain in the first qubit.

resonator. The latter calibration is preferred as it simultaneously ensures that the excitation indeed was swapped into the resonator rather than a two-level state. For this, a second bias pulse is added to the sequence to sweep one of the other qubits onto resonance with the resonator. The amplitude and length of this second pulse can then be swept to maximize the excitation of the respective qubit. The data looks like Figure 9.9b.

9.3.4 Timing Calibration

Due to the absence of measurement crosstalk, it is now no longer as straightforward to calibrate the timing of the bias channels for the different qubits. Instead, the coupling of the excitation through the resonator needs to be used. If the second pulse on the receiving qubit is placed progressively earlier, it will eventually happen before the excitation is fully swapped into the resonator. At this point, the measured final amplitude will decrease as more and more of the excitation remains in the resonator after the swap is over. Since the swaps take a finite time, though, this reduction in the final amplitude happens gradually and might show features resulting from complicated dynamics while both qubits are on resonance with the resonator. Therefore, this method only gives a rough calibration of the timing delay. If the final sequence only requires a single coupling operation in one direction, though, it does not matter too much if the excitation remains in the

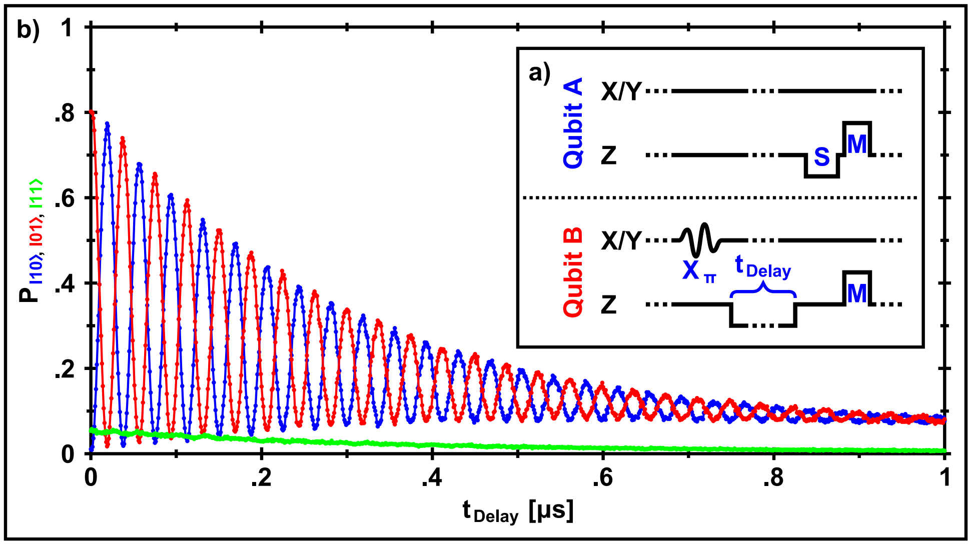

Figure 9.10: Resonator Swaps – a) Sequence: The second qubit is excited into the $|\,1\,\rangle$ state with a π-pulse (X_π). The qubit is brought on resonance with the resonator for a time t_{Delay} allowing it to swap the state back and forth between the qubit and the resonator. After, the first qubit is brought on resonance for the time needed for one full swap (S) to transfer the state from the resonator into the first qubit. Finally both qubits are measured (M). b) Swaps: The final coupled qubit state oscillates between the $|\,10\,\rangle$ and $|\,01\,\rangle$ state. The measurement error in identifying the $|\,0\,\rangle$ state causes a non-zero probability for measuring $|\,11\,\rangle$. As a function of time, T_1 decays the signal and relaxes the qubits into the $|\,00\,\rangle$ state.

resonator for a while before it is retrieved, especially since the T_1 of our current coplanar resonators are much better (few μs) than the T_1's of the qubits.

9.3.5 Swaps

At this point it is possible to reproduce the time resolved swapping experiment described above. In this case, though, the oscillation will be either between the

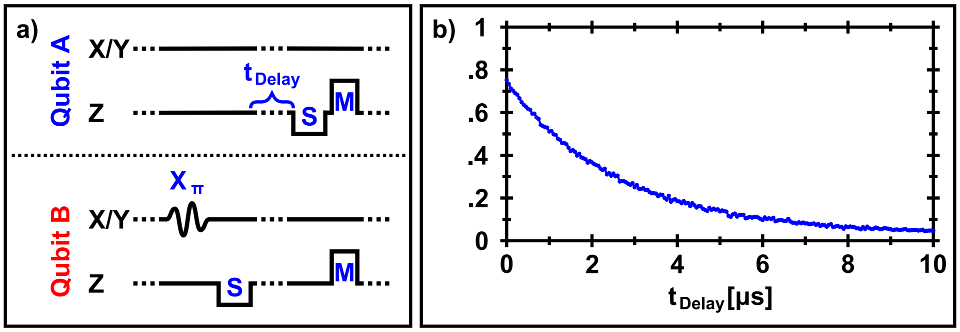

Figure 9.11: Resonator T_1 – a) Sequence: The second qubit is excited into the $|\,1\,\rangle$ state with a π-pulse (X_π). The qubit is brought on resonance with the resonator for the time needed for one full swap (S) to transfer the excitation into the resonator. There, the excitation is allowed to decay for a time t_{Delay}. After, the first qubit is brought on resonance for the time needed for one full swap (S) to transfer the resonator state into the first qubit. Finally both qubits are measured (M). b) T_1: The excitation in the resonator decays exponentially.

source qubit and the resonator or between the resonator and the target qubit. The former will reproduce the dataset from above more faithfully as the latter will lead to some of the excitation sometimes remaining in the resonator at the end. Varying the length of the first swap pulse in the last used sequence will yield the familiar plot shown in Figure 9.10.

9.3.6 Resonator T_1 and T_2

To fully characterize the coupling, it is desirable in this case to also measure the resonator's T_1 and T_2. The T_1 can be easily measured by delaying the retrieval of the excitation from the resonator as shown in Figure 9.11. To measure the T_2, a Ramsey-type experiment needs to be conducted where a $\frac{\pi}{2}$-pulse prepares the source qubit in a state in the X/Y-plane of the Bloch sphere. This state is then swapped into the resonator. After a variable delay t_{Ramsey}, the state is retrieved with another swap operation and the result is interfered with another $\frac{\pi}{2}$-pulse. The same concerns apply to the resulting data of this experiment as described for the single qubit case.

Chapter 10

Hidden Variable Theories versus Quantum Mechanics

10.1 Introduction

10.1.1 Is Quantum Mechanics Incomplete?

Due to its radically new way of seeing the world, quantum mechanics was initially met with great skepticism from many renowned physicists, including Albert Einstein. Specifically the idea that the world was inherently non-deterministic and random did not sit well with Einstein and many of his colleagues. This dislike culminated in Einstein's famous statement: "I, at any rate, am convinced that

He does not throw dice" [Einstein et al., 1971], which is commonly paraphrased as "God does not play dice with the universe". The fact that quantum mechanics does not provide a way to predict outcomes of all possible measurements with certainty lead to the suspicion that quantum mechanics had to be incomplete [Einstein et al., 1935], i.e. the wave-function representation of a system's state does not contain all relevant information about the system. To complete the theory, a way must be found to capture the missing information in extra variables, often called "hidden variables" as they could not be measured. A deterministic alternate theory to quantum mechanics would therefore be called a "Hidden Variable Theory" (HVT).

10.1.2 Is Quantum Mechanics Wrong?

In 1964, John S. Bell investigated the theoretical implications of a possible local HVT and showed that quantum mechanics could not be derived from such a theory to arbitrary accuracy [Bell, 1964]. With this, a hidden variable theory could no longer be a compatible extension to quantum mechanics, but would instead refute quantum mechanics altogether.

10.1.3 Settling the Question Experimentally

J.F. Clauser, M.A. Horne, A. Shimony, and R.A. Holt later formulated one example of an incompatibility between a hidden variable theory and quantum mechanics into an experiment that could test whether quantum mechanics was indeed incomplete [Clauser et al., 1969]. In the proposed experiment a source is used that produces pairs of particles (e.g. photons or ions) in a perfectly anti-correlated state (e.g. opposite polarization or spin). For the purposes of illustration, the state of the entangled pair (particle A and B) can be taken to be the Bell singlet state $\frac{|\,01\,\rangle - |\,10\,\rangle}{\sqrt{2}}$. These particles are then physically separated by a large enough distance to disallow any classical transfer of information between them throughout the remainder of the experiment, i.e. $d_{AB} \gg c\,t_{expt}$. At those remote locations the particles are then measured along random axes (e.g. projected onto the X, Y, or Z-axis of the Bloch sphere). If the singlet state is re-expressed in the basis of the measurement axes, it will still show perfect anti-correlation if the axes are equal, e.g.:

$$\begin{aligned}
\frac{|\,X^-X^+\,\rangle - |\,X^+X^-\,\rangle}{\sqrt{2}} &= \frac{\frac{|\,0\,\rangle - |\,1\,\rangle}{\sqrt{2}} \otimes \frac{|\,0\,\rangle + |\,1\,\rangle}{\sqrt{2}} - \frac{|\,0\,\rangle + |\,1\,\rangle}{\sqrt{2}} \otimes \frac{|\,0\,\rangle - |\,1\,\rangle}{\sqrt{2}}}{\sqrt{2}} \\
&= \frac{\frac{|\,00\,\rangle + |\,01\,\rangle - |\,10\,\rangle - |\,11\,\rangle}{2} - \frac{|\,00\,\rangle - |\,01\,\rangle + |\,10\,\rangle - |\,11\,\rangle}{2}}{\sqrt{2}} \\
&= \frac{|\,01\,\rangle - |\,10\,\rangle}{\sqrt{2}} \qquad (10.1)
\end{aligned}$$

Thus, every time both particles happen to be measured along the same axis (independent of which axis it is), they will yield an opposite outcome with certainty. But since measurements along orthogonal axes do not commute, quantum mechanics not only forbids a simultaneous prediction of the outcome of all possible measurements, but states that this information is not present in the state of the two particles before the measurement. Instead, a measurement of particle A instantaneously collapses the wave-function (changes the state) of particle B despite the fact that they are causally disconnected by their distance. Einstein called this non-local effect of entanglement the "spooky action at a distance".

A possible local hidden variable theory would instead state that the particles agree on all possible measurement outcomes before their separation. This agreement would be contained in the state of the particles in extra unmeasured degrees of freedom, the hidden variables. If the measurements of particles A and B are limited to two possible choices of axes each, a and a' as well as b and b', and the outcomes are encoded in binary (1 or 0), this agreement implies that the particles have to choose at the time of separation to belong to one of the 16 possible populations shown in Table 10.1. Next, one defines a correlation measurement E_{xy} which takes the value 1 if the outcome of a measurement of particle A along axis x and particle B along axis y yields the same result for both particles and a value of -1 for opposite results. For experimental implementation, the expectation value

Table 10.1: Possible Populations for Locally Deterministic Particle Pairs

Pop.	a	a'	b	b'	E_{ab}	$E_{a'b}$	$E_{ab'}$	$E_{a'b'}$	S
n_0	0	0	0	0	1	1	1	1	2
n_1	0	0	0	1	1	1	-1	-1	2
n_2	0	0	1	0	-1	-1	1	1	-2
n_3	0	0	1	1	-1	-1	-1	-1	-2
n_4	0	1	0	0	1	-1	1	-1	-2
n_5	0	1	0	1	1	-1	-1	1	2
n_6	0	1	1	0	-1	1	1	-1	-2
n_7	0	1	1	1	-1	1	-1	1	2
n_8	1	0	0	0	-1	1	-1	1	2
n_9	1	0	0	1	-1	1	1	-1	-2
n_{10}	1	0	1	0	1	-1	-1	1	2
n_{11}	1	0	1	1	1	-1	1	-1	-2
n_{12}	1	1	0	0	-1	-1	-1	-1	-2
n_{13}	1	1	0	1	-1	-1	1	1	-2
n_{14}	1	1	1	0	1	1	-1	-1	2
n_{15}	1	1	1	1	1	1	1	1	2

of E_{xy} is measured by expressing it in terms of the measured state probabilities:

$$E_{xy} = P_{|\,00\,\rangle}(x,y) - P_{|\,01\,\rangle}(x,y) - P_{|\,10\,\rangle}(x,y) + P_{|\,11\,\rangle}(x,y) \tag{10.2}$$

The correlation values E_{xy} are then combined into a measure S using:

$$S = E_{ab} + E_{a'b} - E_{ab'} + E_{a'b'} \tag{10.3}$$

The value of S for each of the possible pair-populations n_i is either $+2$ for $i = 0, 1, 5, 7, 8, 10, 14, 15$ or -2 for $i = 2, 3, 4, 6, 9, 11, 12, 13$. Thus, a measurement of the expectation value of S over an ensemble of many particles drawn from these populations will result in a measured value of S that is a weighted average of -2 and $+2$ with the weights given by the respective fractions n_i:

$$\begin{aligned} S &= 2\,(n_0 + n_1 + n_5 + n_7 + n_8 + n_{10} + n_{14} + n_{15}) \\ &\quad -2\,(n_2 + n_3 + n_4 + n_6 + n_9 + n_{11} + n_{12} + n_{13}) \end{aligned} \tag{10.4}$$

Since n_i captures the probability of the particle pair belonging to population i, it needs to fulfill:

$$0 \leq n_i \leq 1 \tag{10.5}$$

$$\sum_i n_i = 1 \tag{10.6}$$

This gives the following restriction for the measured value of S:

$$|S| \leq 2 \tag{10.7}$$

This restriction is called the Bell inequality. The beauty of this inequality is that its derivation does not assume anything about the used measurement axes a, a', b and b' or even about the fractions n_i, i.e. about the distribution of pairs that the source produces. This allows for a lot of freedom in the implementation and makes the inequality very robust against imperfections.

It turns out that there are choices for the possible measurement axes a, a', b and b' for which quantum mechanics predicts a value that violates the Bell inequality for certain pair states. To give an example, the source can be taken to prepare particle pairs in the Singlet state $|\,\psi\,\rangle = \frac{|\,01\,\rangle - |\,10\,\rangle}{\sqrt{2}}$ and all measurements can be confined to axes to the X/Z-plane. The measurement axes a, a', b, and b' can then be specified by the angles they form with the Z-axis, α, α', β, and β'. Quantum mechanics predicts the probability $P_{|\,1\,\rangle}$ of a positive outcome of a measurement around an axis x of a single particle in state $|\,\psi\,\rangle$ in the Z-basis $|\,0\,\rangle$ and $|\,1\,\rangle$ as:

$$P_{|\,1\,\rangle,\psi}(\theta) = |\langle\,\theta\,|\,\psi\,\rangle|^2 = \left|\left(\cos\frac{\theta}{2}\langle\,0\,| + \sin\frac{\theta}{2}\langle\,1\,|\right)|\,\psi\,\rangle\right|^2 \tag{10.8}$$

The probability $P_{|\,0\,\rangle}$ of a negative outcome can be seen as the probability of a positive outcome if the measurement had been in the opposite direction, i.e.:

$$P_{|\,0\,\rangle,\psi}(\theta) = P_{|\,1\,\rangle,\psi}(\theta + 180^\circ) \tag{10.9}$$

For a measurement of two particles in state $|\,\psi\,\rangle$, this is extended to:

$$\begin{aligned}
P_{|\,11\,\rangle,\psi}(x,y) &= |\langle\, x,y\,|\,\psi\,\rangle|^2 \\
&= \left|\left(\left(\cos\frac{x}{2}\langle\, 0\,| + \sin\frac{x}{2}\langle\, 1\,|\right) \otimes \left(\cos\frac{y}{2}\langle\, 0\,| + \sin\frac{y}{2}\langle\, 1\,|\right)\right)|\,\psi\,\rangle\right|^2 \\
&= \left|\left(\cos\frac{x}{2}\cos\frac{y}{2}\langle\, 00\,| + \sin\frac{x}{2}\cos\frac{y}{2}\langle\, 10\,|\right.\right. \\
&\qquad \left.\left. + \cos\frac{x}{2}\sin\frac{y}{2}\langle\, 01\,| + \sin\frac{x}{2}\sin\frac{y}{2}\langle\, 11\,|\right)|\,\psi\,\rangle\right|^2 \qquad (10.10)
\end{aligned}$$

Applied to the Singlet state $\frac{|\,01\,\rangle - |\,10\,\rangle}{\sqrt{2}}$, this gives:

$$\begin{aligned}
P_{|\,11\,\rangle,\psi}(x,y) &= \left|\left(\cos\frac{x}{2}\cos\frac{y}{2}\langle\, 00\,| + \sin\frac{x}{2}\cos\frac{y}{2}\langle\, 10\,|\right.\right. \\
&\qquad \left.\left. + \cos\frac{x}{2}\sin\frac{y}{2}\langle\, 01\,| + \sin\frac{x}{2}\sin\frac{y}{2}\langle\, 11\,|\right)\frac{|\,01\,\rangle - |\,10\,\rangle}{\sqrt{2}}\right|^2 \\
&= \frac{1}{2}\left(\sin\frac{x}{2}\cos\frac{y}{2} - \cos\frac{x}{2}\sin\frac{y}{2}\right)^2 \qquad (10.11) \\
P_{|\,00\,\rangle,\psi}(x,y) &= P_{|\,11\,\rangle,\psi}(x+180^\circ, y+180^\circ) \\
&= \frac{1}{2}\left(-\cos\frac{x}{2}\sin\frac{y}{2} + \sin\frac{x}{2}\cos\frac{y}{2}\right)^2 \qquad (10.12) \\
P_{|\,01\,\rangle,\psi}(x,y) &= P_{|\,11\,\rangle,\psi}(x+180^\circ, y) \\
&= \frac{1}{2}\left(\cos\frac{x}{2}\cos\frac{y}{2} + \sin\frac{x}{2}\sin\frac{y}{2}\right)^2 \qquad (10.13) \\
P_{|\,10\,\rangle,\psi}(x,y) &= P_{|\,11\,\rangle,\psi}(x, y+180^\circ) \\
&= \frac{1}{2}\left(-\sin\frac{x}{2}\sin\frac{y}{2} - \cos\frac{x}{2}\cos\frac{y}{2}\right)^2 \qquad (10.14)
\end{aligned}$$

This leads to the correlation values:

$$\begin{aligned}\langle\, E_{xy}\,\rangle &= P_{|\,00\,\rangle,\psi}(x,y) - P_{|\,01\,\rangle,\psi}(x,y) - P_{|\,10\,\rangle,\psi}(x,y) + P_{|\,11\,\rangle,\psi}(x,y)\\ &= 2\cos^2\frac{x}{2} + 2\cos^2\frac{y}{2} - 4\cos^2\frac{x}{2}\cos^2\frac{y}{2}\\ &\qquad -4\sin\frac{x}{2}\cos\frac{x}{2}\sin\frac{y}{2}\cos\frac{y}{2} - 1 \qquad (10.15)\end{aligned}$$

If the angles are chosen for example as $\alpha = -135°$, $\alpha' = +135°$, $\beta = 0°$, and $\beta' = -90°$, quantum mechanics therefore predicts the following expectation value of S:

$$\langle\, S\,\rangle = \langle\, E_{\alpha\beta}\,\rangle + \langle\, E_{\alpha'\beta}\,\rangle - \langle\, E_{\alpha\beta'}\,\rangle + \langle\, E_{\alpha'\beta'}\,\rangle = 2\sqrt{2} \approx 2.8284 \qquad (10.16)$$

This value clearly violates the Bell inequality. Thus, quantum mechanics predicts that an experimental measurement of S can potentially yield a value > 2 and if it does, the idea of a possible local hidden variable theory has to be rejected.

10.2 Experimental Results

Many experiments have meanwhile implemented versions of this test and obtained values for S that violated the Bell inequality by many standard deviations. Despite some potential points of criticism, this, together with quantum mechanic's other predictive successes, has lead to a broad acceptance of the theory.

10.2.1 Photons

The first and most notable experiments to violate the Bell inequality were based on measurements of entangled pairs of polarized photons. The sources used to create the pairs produced a random stream of photon pairs and the measurement was based on coincidence detection of photons received by a detector behind a polarization filter. This setup necessitated the development of modified versions of the Bell inequality, like the CH74 [Clauser and Horne, 1974] inequality. This is due to the fact that the unpredictability of the photon source effectively makes it impossible to ever identify a $|\,00\,\rangle$ measurement and the inefficiencies of the detectors make $|\,01\,\rangle$ or $|\,10\,\rangle$ measurements highly unreliable.

Using the CH74 inequality, Aspect et al. showed a violation of the inequality by 9 standard deviations in 1981 [Aspect et al., 1981]. Meanwhile, the experimental setups have become so optimized that more recent experiments are trying to obtain a value of S as close as possible to the quantum mechanical limit of $2\sqrt{2}$. For example, in 2005 J.B. Altepeter reported a value of $S = 2.7252 \pm 0.000585$, which corresponds to a violation by 1239 standard deviations [Altepeter et al., 2005].

Unfortunately, due to the non-ideal detector efficiencies, these types of experiments are vulnerable to criticism. They suffer from a flaw called the "Detection Loophole" [Pearle, 1970], which is based on the fact that the photons besides "choosing" between polarizations also have the option to not be detected at all. If two photon pairs are emitted by the source in close succession and in each pair one photon "decides" to remain undetected, the remaining two photons can incorrectly be attributed as belonging to the same pair.

10.2.2 Ions

To close the detection loophole, M.A. Rowe et al. re-implemented the experiment using entangled pairs of $^9Be^+$ ions [Rowe et al., 2001]. Since these ions could be sourced predictably, one pair at a time, this allowed the group to use a complete set of measurements of all four possible outcomes. The group obtained a value for S of $S = 2.25 \pm 0.03$, also disproving the existence of local hidden variable theories in favor of quantum mechanics.

This experiment was still susceptible to criticism, since the ions remained in relatively close proximity during the entire time of the experiment. Thus, one could postulate an interaction between the ions at the time of measurement that leads to an apparent higher correlation. This flaw is called the "Locality Loophole".

10.2.3 Ion and Photon

In the attempt to close both loopholes, D.L. Moehring published an experiment in 2004 that used an atom and a photon as the entangled pair [Moehring et al., 2004]. The high detection efficiency of the measurement of the atom's state combined with the theoretical possibility to remove the photon far from the atom could eventually allow this approach to implement a loophole-free Bell inequality test. But due to limitations of the experiment, the group was not able to close the locality loophole in this version of the experiment. The group reported an S-value of $S = 2.218 \pm 0.028$.

10.3 The Bell Inequality versus Phase Qubits?

Given the fact that the overwhelming evidence [Weihs et al., 1998, Roos et al., 2004] has effectively settled the question of whether a local hidden variable theory should replace quantum mechanics, one might ask why another implementation of a test of Bell inequality in superconducting qubits is desirable, especially since such an experiment would most likely also be susceptible to the locality loophole. The argument in favor of the experiment is two-fold:

- On the one hand, it would be the first implementation of the test using a macroscopic quantum state. All experiments to date have relied on microscopic quantum systems like atoms, ions, photons, etc. The quantum state of a superconductor can extend over many hundreds micrometers and involves a collection of around a billion electrons.

- On the other hand, a successful implementation of this experiment will provide strong evidence that superconducting qubits can indeed show non-classical behavior and are thus a viable candidate for the implementation of a quantum computer [Clarke and Wilhelm, 2008]. In addition, the operations required for a successful implementation cover almost all of the DiVincenzo criteria (except scalability) and the experiment places extremely high demands on the fidelities of these operations. Thus, the S-value obtained can be used as a very powerful single-number benchmark for the overall quality of the qubit pair.

Chapter 11

Implementing the Bell Test

Due to the inherent freedom in the circuit design, a Bell inequality test with superconducting phase qubits can be implemented in several different ways. Here, two approaches are investigated that differ primarily in the coupling scheme used to prepare the entangled state of a pair of qubits. In one case, the coupling is achieved with an always-on coupling capacitor, while in the other a coplanar resonator is used to act as a band-pass filter for the coupling to eliminate measurement crosstalk. Measurement crosstalk is a major concern for testing the Bell inequality as it actively introduces correlations into the results, an effect that needs to be avoided as it constitutes a major loophole [Kofman and Korotkov, 2008b].

11.1 State Preparation

Since the ability for quantum states to violate the Bell inequality is rooted in their entanglement, the first step to any implementation is the generation of a highly entangled state between two qubits, i.e. building the source of particle pairs.

11.1.1 Initialization in the $|\,10\,\rangle$ state

The sequence begins by initializing the qubit pair into the $|\,10\,\rangle$-state. For this, both qubits are first allowed to reset into the ground state $|\,00\,\rangle$ via energy decay. This is followed by a π-pulse on one of the qubits to create the state $|\,10\,\rangle$. The procedure for calibrating the π-pulse is outlined in Chapter 8.7.

11.1.2 Entangling the Qubits – Capacitive Coupling

Next, the qubit pair needs to be entangled using a coupling operation. The procedure for this varies depending on the coupling scheme. For always-on capacitive coupling, the qubits are brought on resonance with each other as described in Chapters 9.2.2 and 9.2.4 for enough time to allow for half of a swap operation. This leads to the application of the $\sqrt{i - Swap}$ gate which (ideally) leaves the qubits in the $|\,10\,\rangle - i|\,01\,\rangle$ state. Even though this state is not quite the Bell singlet state due to the extra factor of i, it nevertheless shows the same degree of entanglement. Since the initial state of the qubit pair prepared by the source is not a condition used in the derivation of the Bell inequality, it is not of fundamental importance that the state created by the "particle source" yields qubit pairs in the Bell singlet state. In fact, depending on the specifics of the experiment's imperfections, a different initial state might yield a higher S-value.

Simulations and experiments show that imperfections in the state preparation caused by sweeping the qubits on or off resonance are more detrimental to the value of S than imperfections caused by an impaired π-pulse due to the qubits being coupled during the pulse. Thus, it turns out to be beneficial for the outcome and the simplicity of the experiment to begin the sequence with both qubits immediately on resonance and applying the π-pulse while the coupling is on. This works due to the fact that the π-pulse is shorter than the time-scales of the coupling operation. The final entangling sequence for the qubit pair source using always-on capacitive coupling is shown in Figure 11.1a.

11.1.3 Entangling the Qubits – Resonator Coupling

The pair preparation using a resonator based coupling element requires a sequence that is slightly different from the simple capacitive coupling. This is due to the fact that the resonator can itself store excitations which impede the entangling

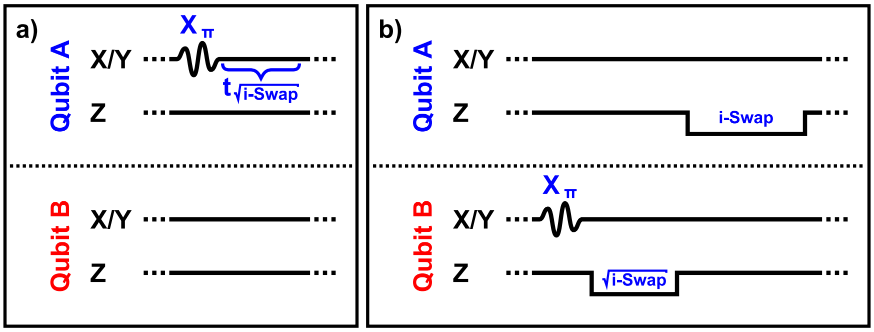

Figure 11.1: Bell State Preparation – a) Capacitive coupling: The qubits are biased on resonance. One of the qubits is excited into the $|\,1\,\rangle$ state with a π-pulse. The qubits are allowed to interact for a time $t_{\sqrt{i-Swap}}$ resulting in the state $|\,10\,\rangle - i|\,01\,\rangle$. b) Resonator coupling: One of the qubits is excited into the $|\,1\,\rangle$ state with a π-pulse and brought on resonance with the resonator for a time $t_{\sqrt{i-Swap}}$ to entangle the qubit with the resonator. The other qubit is brought on resonance with the resonator for a time t_{i-Swap} to transfer the entanglement to that qubit. This leaves the qubit system in the state $|\,10\,\rangle + e^{i\,\alpha}|\,01\,\rangle$, with the unknown phase α caused by the Z-rotations inherent in the bias changes needed to bring the qubits on and off resonance with the resonator.

process of the qubits in two ways:

- If the circuit is driven with microwaves at a frequency that matches that of the resonator, the resonator will begin accepting photons as described in Chapter 3.3. This leads to complicated entangled states between the qubits and the resonator that make it very hard to create a clean entanglement only between the two qubits. Thus, the qubits need to be prepared in the $|\,10\,\rangle$ state while they are biased at a frequency far away from the resonator.
- If both qubits are placed on resonance with the resonator at the same time, the interaction between the three systems also leads to very complicated dynamics that will prevent clean qubit-only entanglement. Therefore, the qubits need to interact with the resonator one after the other to create the state.

Overall, the entangling sequence consists of bringing the excited qubit on resonance with the resonator for a time that yields half a swap operation and then bringing the second qubit on resonance for a full swap time, as indicated in Figure 11.1b. The half-swap creates the entangled state $|\,10\,\rangle - i|\,01\,\rangle$ between the first qubit and the resonator, while the full-swap transfers the resonator's share of the entanglement into the other qubit, leaving the resonator in the ground state and the two qubits in the state $|\,01\,\rangle + e^{i\alpha}|\,10\,\rangle$. The phase factor $e^{i\alpha}$ results from

Table 11.1: Entangled State Density Matrix – Raw

ρ_{raw}	$\|00\rangle$	$\|01\rangle$	$\|10\rangle$	$\|11\rangle$
$\|00\rangle$	0.151	$-0.005+0.034i$	$0.041+0.034i$	$-0.003+0.005i$
$\|01\rangle$	$-0.005-0.034i$	0.369	$-0.380-0.000i$	$-0.030-0.012i$
$\|10\rangle$	$0.041-0.034i$	$-0.380+0.000i$	0.428	$-0.004-0.054i$
$\|11\rangle$	$-0.003-0.005i$	$-0.030+0.012i$	$-0.004+0.054i$	0.051

Table 11.2: Entangled State Density Matrix – Corrected for Visibilities

ρ_{corr}	$\|00\rangle$	$\|01\rangle$	$\|10\rangle$	$\|11\rangle$
$\|00\rangle$	0.135	$-0.006+0.040i$	$0.046+0.037i$	$-0.003+0.006i$
$\|01\rangle$	$-0.006-0.040i$	0.387	$-0.431+0.000i$	$-0.034-0.015i$
$\|10\rangle$	$0.046-0.037i$	$-0.431-0.000i$	0.449	$-0.004-0.061i$
$\|11\rangle$	$-0.003-0.006i$	$-0.034+0.015i$	$-0.004+0.061i$	0.029

the fact that the pulses that sweep the first qubit off and the second qubit on resonance also cause Z-rotations.

11.1.4 Verifying Entanglement – State Tomography

At this point, for either coupling scheme, the qubit pair should be prepared in a highly entangled state similar to the Bell singlet. To verify the quality of the created state, one can employ a technique called "Quantum State Tomography"

[Steffen et al., 2006] (the details of which are beyond the scope of this thesis) to measure not just the fractional populations of the possible states, but the entire density matrix. The result is shown in Table 11.1 for the entangled pair created using the resonator coupling scheme.

Comparing this density matrix to the ideal Bell singlet state $|\,10\,\rangle - |\,01\,\rangle$ yields a trace-fidelity of $\mathrm{F}(\rho_{raw}) = \sqrt{\mathrm{Tr}(\langle\, Singlet\,|\rho_{raw}|\, Singlet\,\rangle)} = 88.3\%$. Calculating the "Entanglement of Formation" [Hill and Wootters, 1997] (EoF) of the state yields a value of $\mathrm{EoF}(\rho_{raw}) = 0.378$. The EoF gives a monotonic measure of the entanglement shown by the coupled quantum state. It ranges from 0 for classical states to 1 for maximally entangled states. Intuitively, it gives the inverse of the number of identical copies of the state needed that would allow a purification protocol to combine the copies into a maximally entangled state.

To get a better understanding of the true state of the coupled pair, the data can be corrected for measurement visibilities to obtain the result shown in Table 11.2. This state shows a trace-fidelity of $\mathrm{F}(\rho_{corr}) = \sqrt{\mathrm{Tr}(\langle\, Singlet\,|\rho_{corr}|\, Singlet\,\rangle)} = 92.1\%$ and an entanglement of formation of $\mathrm{EoF}(\rho_{corr}) = 0.449$.

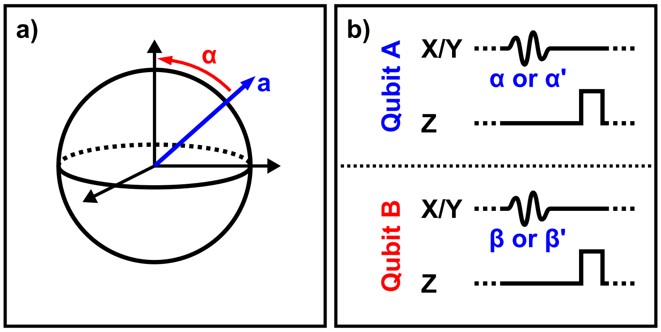

Figure 11.2: Bell Measurements – a) Rotation: Since the tunneling measurement always measures the qubits along the Z-axis, a measurement along a different axis needs to be emulated by rotating that axis onto the Z-axis first. b) Measurement sequence: The entangled qubits are rotated and then measured.

11.2 Correlation Measurements

The entangling sequence can now be repeated many times over to predictably yield an ensemble of entangled qubit pairs one at a time. These pairs then need to be put through the different measurements to obtain values for the correlation measures E_{xy} and S as described above. The measurements need to be executed in two steps, though, since the tunneling measurement scheme used to read out the qubits always measures the Z-component of the qubit state and therefore does not directly allow for the needed measurements around arbitrary axes.

11.2.1 Bell Rotations

The first step in the measurement consists of applying a rotation operation to each qubit that rotates the qubits' states along the Bloch sphere in such a manner that the axis along which the measurement is to be done is moved onto the Z-axis. As described before, this is achieved with a microwave pulse of the right phase and amplitude as indicated in Figure 11.2b. For different experimental runs, i.e. different pairs from the ensemble, the rotation is adjusted to correspond to the axes a, a', b, and b' as defined in the previous chapter.

11.2.2 Tunneling Measurement

In the second step, the usual tunneling measurement is used to measure the state along the Z-axis. Each experimental run will then yield one of the four states $|\,00\,\rangle$, $|\,01\,\rangle$, $|\,10\,\rangle$ or $|\,11\,\rangle$. Their occurrences are counted for the different rotations a, a', b, and b' and their relative rates for each axes combination are interpreted as the probabilities $P_{|\,00\,\rangle}(a, b)$, $P_{|\,00\,\rangle}(a', b)$, $P_{|\,00\,\rangle}(a, b')$, $P_{|\,00\,\rangle}(a', b')$, $P_{|\,01\,\rangle}(a, b)$, etc. These probabilities are combined into the correlation measures E_{xy} according to Equation 10.2, which in turn give the value for S (Equation 10.3).

11.2.3 Statistical Analysis

For the final value of S to carry meaning, it needs to be supplemented with an estimate of its standard deviation σ_S. To come up with a meaningfull estimate of this standard deviation turns out to be somewhat involved. A simple-minded approach is to break up an experiment consisting of N runs into n sections of $\frac{N}{n}$ runs each. For each section i, an S-value S_i can be calculated and the resulting sample set can be statistically analyzed to obtain an estimator for the sample mean $\overline{S}$, the sample standard deviation σ_{S_i}, and the resulting standard error of the sample mean $\sigma_{\overline{S}}$:

$$\overline{S} = \frac{1}{n}\sum_{i=1}^{n} S_i \tag{11.1}$$

$$\sigma_{S_i} = \sqrt{\frac{1}{n-1}\sum_{i=1}^{n}\left(S_i - \overline{S}\right)^2} \tag{11.2}$$

$$\sigma_{\overline{S}} = \frac{\sigma_{S_i}}{\sqrt{n}} \tag{11.3}$$

The problem with this analysis is that it assumes the errors of S_i to be uncorrelated, an assumption that does not hold in the presence of systematic errors in the experiment like drift and $1/f$ noise. This leads to different estimates of $\sigma_{\overline{S}}$ for different choices for the number of sections n. In reality, the overall standard error in S as a function of sampling-time is the result of two competing effects:

- S is calculated from a linear combination of 16 probabilities ($P_{|00\rangle}(ab)$, ...,

$P_{|11\rangle}(a'b'))$ estimated by sampling four multinomial distributions. As the number of samples increases, the statistical sampling noise goes down and the estimates of the probabilities becomes better following $\sigma_p = \sqrt{\frac{p(1-p)}{n}}$.

- Since the experiment is subject to $1/f$ noise, the drift in the "true" value of S over the course of the experiment becomes worse and worse as the sampling time is increased.

For small sample sizes, the standard error on S is therefore limited by statistical sampling noise, while for large sample sizes, the error will be limited by drift of S. Thus, there will exist a certain sample size n^* for which the two effects become equal and the error switches from being dominated by sampling noise to being dominated by experimental drifts. For sample sizes smaller than n^*, the above described simple-minded analysis approach is valid, while for sample sizes larger than n^*, this approach can severely underestimate the standard error σ_S.

To avoid having to model the statistical effects of $1/f$ noise on S, the data in this thesis will be presented as a collection of several measurements of S with individual estimates of σ_S, each of which is based on data taken over short enough periods to guarantee sample sizes smaller than n^*.

11.3 Calibration

The most difficult part of implementing any qubit experiment is that of calibrating the different control signals. The beauty of the Bell inequality used in this experiment is that its derivation is independent of almost all of these calibrations. If a hidden variable theory existed, the inequality would have to hold for any prepared state of the qubit pair as long as the preparation is entirely independent of the choice of measurement axes. It would also have to hold for any choice of measurement axes a, a', b, and b'. It would even have to hold in the presence of energy relaxation, dephasing, and reduced measurement visibility as long as these effects are uncorrelated with the choice of measurement axes and do not introduce artificial correlations into the qubit pair's state [Kofman and Korotkov, 2008a].

11.3.1 Global Optimization

This implies that it is not necessary to calibrate the relevant experimental parameters in detail independently, but instead gives the freedom to include most of them in a global optimization that maximizes (minimizes) the measured value of S. The parameters that can be included in this optimization are given in Tables 11.3, 11.4, and 11.5.

For the optimization, the experiment is simply viewed as an oracle that evalu-

Table 11.3: Sequence Parameters – Capacitive State Preparation

Name	Description	Comments
Φ_A	Qubit A flux bias	Places qubit far from TLS's
Φ_B	Qubit B flux bias	Places qubits on resonance
t_π	Length of π-pulse	Much less than $t_{\sqrt{i-Swap}}$
A_π	Amplitude of π-pulse	Sufficiently small to avoid $\mid 2 \rangle$-state errors
f_π	Frequency of π-pulse	Corrects for AC stark shift $\rightarrow$ function of A_π
ϕ_π	Phase of π-pulse	Shouldn't matter
$t_{\sqrt{i-Swap}}$	Swap time	Can be absorbed into rotation pulse delay

ates S as a function of all free parameters. Since it is not possible to measure the derivative of S, the optimization is limited to "Direct Search" methods that do not rely on the knowledge of such derivatives. The algorithms to be used need to further be evaluated with respect to their tolerance for noisy data. Three algorithms seem to be good candidates for this optimization: Manual search, Nelder-Mead Simplex Optimization [Nelder and Mead, 1965], and Particle Swarm Optimization [Eberhart and Kennedy, 1995].

11.3.2 Manual Search

Manual search, as the name suggests, involves the experimenter probing the oracle manually in order to gain an understanding of the behavior of S to eventu-

Table 11.4: Sequence Parameters – Resonator State Preparation

Name	Description	Comments
Φ_A	Qubit A flux bias	Places qubit far from TLS's and resonator
Φ_B	Qubit B flux bias	Places qubit far from TLS's, resonator, and other qubit
t_π	Length of π-pulse	Short
A_π	Amplitude of π-pulse	Sufficiently small to avoid $\|\,2\,\rangle$-state errors
f_π	Frequency of π-pulse	Corrects for AC stark shift $\rightarrow$ function of A_π
ϕ_π	Phase of π-pulse	Shouldn't matter
$dt_{\sqrt{i-Swap}}$	Entangling pulse delay	Ensures π-pulse is done
$t_{\sqrt{i-Swap}}$	Entangling time	Maximizes entaglement
$A_{\sqrt{i-Swap}}$	Entangling pulse amplitude	Brings qubit and resonator on resonance
$O_{\sqrt{i-Swap}}$	Entangling pulse overshoot	Adds delta function to beginning and end of pulse to compensate for qubit slowly moving onto resonance
dt_{i-Swap}	Swap pulse delay	Ensures entangling is done
t_{i-Swap}	Swap time	Adjusted for maximal state transfer
A_{i-Swap}	Swap pulse amplitude	Brings qubit and resonator on resonance
O_{i-Swap}	Swap pulse overshoot	Compensates for qubit slowly moving onto resonance

Table 11.5: Sequence Parameters – Correlation Measurement

Name	Description	Comments
dt_A	Qubit A rotation delay	Ensures state prep. is done
t_a	Length of a-rotation pulse	As short as possible
A_a	Amplitude of a-rotation pulse	Determines angle α
f_a	Frequency of a-rotation pulse	Corrects AC stark shift → fn of A_a
ϕ_a	Phase of a-rotation pulse	Shouldn't matter
$t_{a'}$	Length of a'-rotation pulse	As short as possible
$A_{a'}$	Amplitude of a'-rotation pulse	Determines angle α'
$f_{a'}$	Frequency of a'-rotation pulse	Corrects AC stark shift → fn of $A_{a'}$
$\phi_{a'}$	Phase of a'-rotation pulse	Should equal ϕ_a
dt_{MP_A}	Qubit A meas. pulse delay	Ensures rotation pulse is done
A_{MP_A}	Qubit A meas. pulse ampl.	Maximizes visibility
t_{MP_A}	Qubit A meas. pulse length	Maximizes visibility (cap. coupl.: Minimizes crosstalk)
dt_B	Qubit B rotation delay	Ensures state prep. is done
t_b	Length of b-rotation pulse	As short as possible
A_b	Amplitude of b-rotation pulse	Determines angle β
f_b	Frequency of b-rotation pulse	Corrects AC stark shift → fn of A_b
ϕ_b	Phase of b-rotation pulse	Should result in same phase as ϕ_a (corrects for cable lengths)
$t_{b'}$	Length of b'-rotation pulse	As short as possible
$A_{b'}$	Amplitude of b'-rotation pulse	Determines angle β'
$f_{b'}$	Frequency of b'-rotation pulse	Corrects AC stark shift → fn of $A_{b'}$
$\phi_{b'}$	Phase of b'-rotation pulse	Should equal ϕ_b
dt_{MP_B}	Qubit B meas. pulse delay	Ensures rotation pulse is done (cap. coupl.: Minimizes crosstalk)
A_{MP_B}	Qubit B meas. pulse ampl.	Maximizes visibility
t_{MP_B}	Qubit B meas. pulse length	Maximizes visibility (cap. coupl.: Minimizes crosstalk)

ally determine an optimal value for each parameter. The easiest way to implement this approach is by varying one parameter at a time while holding all other parameters fixed to find a local optimum along the resulting line through parameter space. This process can be iteratively repeated for all free parameters.

The main benefit of this method of optimization is that it provides a rough intuitive understanding of the behavior of S which can be used to expose flaws in the experimental setup or potential loopholes. This method of optimization, if implemented correctly, is also fairly immune to wasting time in parts of the parameter space that are known to have no hope of yielding a good outcome.

There also are three shortcomings to this method:

- For one, this method becomes very hard to implement if different parameters influence the value of S in a correlated way. For example, a change in the bias pulses used to sweep the qubits on or off resonance will cause a change in the Z-rotation associated with the pulse, which will in turn affect the optimal value for the phase of the Bell rotation pulses.

- The duty cycle with which this method will be able to query the oracle is fairly low as each completed run is followed by a period of analysis, parameter adjustment, and preparation for the next run. Not only does this reduce the rate at which information is learned about S, but the irregular breaks in the data taking can lead to thermal drifts between runs that cause changes in the optimal parameter values.

- The method is extremely labor intensive.

11.3.3 Nelder-Mead Simplex Algorithm

A very common Direct Search optimization algorithm is the Nelder-Mead Simplex algorithm [Nelder and Mead, 1965]. It is based on the definition of a simplex, i.e. a geometrical object in the n-dimensional parameter space consisting of $n+1$ vertices. Each vertex is assigned a fitness value which is given by the S-value reported by the oracle for the parameters that specify the position of the vertex. The vertex with the lowest fitness (worst S-value) is then mirrored around the "center of gravity" (average position) of all other vertices. Its fitness is evaluated at the new position and compared to the fitness of all other vertices. If the modified vertex has a better fitness than all other vertices, it is moved even further in the same direction. If the vertex remained the worst in the set, it is moved closer towards the center of gravity of the remaining vertices. If it still remains the worst, the entire simplex is shrunk towards the current best vertex.

This algorithm performs extremely well for functions with only a few small local minima. Most common optimization routines included in software packages are based on this method, including Matlab's "fminsearch" function. The algorithm also handles noise reasonably well.

The major drawbacks of the simplex method are that its performance is extremely dependent on the choice of the initial simplex, which is not always easy to do right. It also gets stuck fairly easily in local minima. Furthermore, the algorithm interleaves function evaluations with decision steps that are based on the results of these function evaluations. This makes it impossible to pipeline the evaluations except for the rare cases when all vertices need to be reevaluated. This leads to a low and somewhat random experimental duty cycle and thus potential thermal drifts.

11.3.4 Particle Swarm Optimization

Relatively recently, an attempt to model social decision making behavior led to the invention of a new class of Direct Search algorithms called "Particle Swarm Optimization" [Eberhart and Kennedy, 1995]. These algorithms are based on the random placement of "particles" throughout the parameter space. Each particle probes the fitness of the function at its position and keeps track of the position of the best fitness it has ever observed. The particles' positions are updated sequentially by simulating their motion through parameter space under forces that accelerate them randomly towards the current globally known best position and back towards the position of their personal best fitness. The particles are assigned a uniform inertial mass to favor exploration, and the space is given a viscosity to eventually damp the motion and cause convergence.

Particle Swarm Optimization is considered a class of algorithms, as there are many possible ways in which the algorithm can be implemented. Choices include whether the particles' knowledge of the global best position is limited to information about only a few neighboring particles, whether particles get added to or removed from the swarm dynamically, whether other points of attraction or repulsion are kept, and the exact values of the particles' mass and the space's viscosity. This flexibility makes it possible to customize the algorithm to yield optimal behavior for the circumstances of the given problem.

Particle Swarm Optimization has several major advantages. It is very robust against noisy data and getting stuck in local minima (if the number of particles, their mass and the vicosity are chosen right). It can be modified to allow for pipelining with 100% function evaluation duty cycle. The modification requires a slight relaxation of the definition of global best fitness in that it must exclude the new values for points that are currently being evaluated. This restriction does not noticeably reduce the performance of the algorithm, though. Last, but not least, it is extremely easy to implement and has very little overhead.

One disadvantage is the number of function evaluations that the algorithm requires to converge. Since the algorithm has much less built-in “smart” decision making, it needs to draw its information from more data. But due to its pipelineability and thus higher duty cycle, it does not run for much longer than the Nelder-Mead Simplex algorithm before it converges.

11.4 Experimental Results

Even though the main result of the experiment consists of only the obtained S-value together with its standard deviation, this number needs to be supplemented with several datasets that test the experiment for flaws. The most useful of this additional information are the parameters of the sequence found by the optimization, since angles that match the theoretically expected values are a strong indication that the implementation can be trusted. To claim a reliable violation, the dataset should further address all known mechanism for the introduction of artificial correlations during the time of measurement, e.g. microwave and measurement crosstalk. Additional checks that show the variation of S as a function of certain parameters can be of assistance in debugging the experiment, but are of lesser importance for proving its correctness due to the robustness of the inequality. Finally, to explain the obtained S-value, the performance parameters of the involved qubits (and resonator), like T_1 and T_2, can be used in a simulation

Table 11.6: Qubit Sample Parameters – Capacitive Coupling Implementation

Parameter	Value
Qubit A:	
T_1	340 ns
Visibility	85.8%
Qubit B:	
T_1	480 ns
Visibility	85.3%
Coupling:	
Swap Frequency	11.4 MHz
Measurement Crosstalk	$\sim 10\%$

to create an error budget for the experiment.

11.4.1 Capacitive Coupling

For the capacitively coupled sample, the optimization yielded an S-value of 1.816 (see Table 11.7) for the sequence parameters shown in Table 11.8. As expected from theory, the two measurements on each qubit are roughly perpendicular ($\alpha - \alpha' \approx 73^\circ - (-16^\circ) = 89^\circ$ and $\beta - \beta' \approx 168^\circ - 82^\circ = 86^\circ$). The phases of the rotation pulses, i.e. the planes in which the qubits are measured, seem less correct at first glance. But it is hard to come to a reliable conclusion since $\phi_{a'}$ and ϕ_b

Table 11.7: Bell Violation Results – Capacitive Coupling Implementation

Parameter	ab	$a'b$	ab'	$a'b'$
$P_{\|00\rangle}$	0.262	0.290	0.111	0.325
$P_{\|01\rangle}$	0.101	0.072	0.297	0.128
$P_{\|10\rangle}$	0.140	0.184	0.356	0.119
$P_{\|11\rangle}$	0.497	0.454	0.237	0.427
E	0.518	0.489	-0.305	0.505
S	1.816			

have little influence on the state as their associated rotations are close to 0° and 180° respectively ($-16°$ and $168°$). The two remaining phases are not expected to have a predictable relationship since they include the phase difference caused by the different electrical delays in the two bias lines. The coupling time $t_{\sqrt{i-Swap}}$ of 6.0 ns can be understood by realizing that the qubits are on resonance during the π-pulse and during the Bell rotations. If one assumes that this contributes roughly half of the pulse length to the coupling time, this yields a total coupling time of $\frac{20.0\,\mathrm{ns}}{2} + 6.0\,\mathrm{ns} + \frac{16.0\,\mathrm{ns}}{2} = 24.0\,\mathrm{ns}$. Given the coupling strength of 11.4 MHz (see Table 11.6), one would expect $t_{\sqrt{i-Swap}} = \frac{1}{4\times 11.4\,\mathrm{MHz}} = 21.9\,\mathrm{ns}$, a reasonably good match.

Since this experiment did not yield the desired S-value greater than 2.0, it is

Table 11.8: Optimization Results – Capacitive Coupling Implementation

Parameter	Value	Comments
Φ_A	$-174.4\,\mathrm{mV}$	
Φ_B	$384.1\,\mathrm{mV}$	
t_π	$20.0\,\mathrm{ns}$	Overall length of Slepian pulse ($\sim 10.0\,\mathrm{ns}$ FWHM)
A_π	0.635	Corresponds to $\sim 180°$
$f_\pi, f_a, f_{a'}, f_b, f_{b'}$	$5.477\,\mathrm{GHz}$	
ϕ_π	$-11°$	
$t_{\sqrt{i-Swap}}$	$6.0\,\mathrm{ns}$	
dt_A, dt_B	$0.0\,\mathrm{ns}$	
$t_a, t_{a'}, t_b, t_{b'}$	$16.0\,\mathrm{ns}$	Overall length of Slepian pulse ($\sim 8.0\,\mathrm{ns}$ FWHM)
A_a	0.320	Corresponds to $\sim 73°$
ϕ_a	$-112°$	
$A_{a'}$	-0.070	Corresponds to $\sim -16°$
$\phi_{a'}$	$-107°$	
dt_{MP_A}, dt_{MP_B}	$3.0\,\mathrm{ns}$	Includes cable delay compensation
A_{MP_A}	0.336	
t_{MP_A}, t_{MP_B}	$3.0\,\mathrm{ns} + 40.0\,\mathrm{ns}$	$3.0\,\mathrm{ns}$ flattop followed by $40.0\,\mathrm{ns}$ ramp
A_b	0.740	Corresponds to $\sim 168°$
ϕ_b	$-129°$	
$A_{b'}$	0.360	Corresponds to $\sim 82°$
$\phi_{b'}$	$136°$	
A_{MP_B}	0.500	

Table 11.9: Error Budget – Capacitive Coupling Implementation

Error	Contribution	New S
Theoretical		2.828
Ideal simulation	0.000	2.828
T_1 decay	0.133	2.695
Visibility	0.512	2.183
Finite pulses during always-on coupling	0.116	2.067
Measurement crosstalk	0.240	1.827
Experimental result (T_2, calibrations, ...)		1.816

useful primarily as a tool for understanding the error mechanisms that led to the lower number. With the help of the numerical simulations described in Chapter 3.3 it is possible to create an error budget that identifies the contributions of different imperfections to the reduction of the S-value. The results of this analysis are shown in Table 11.9. Even though T_1 decay is commonly cited as the biggest challenge faced by the field, for this experiment it has a relatively small effect since the overall sequence is fairly short. The biggest reduction results from the problems associated with the measurement, specifically the low visbilities and high crosstalk. Measurement crosstalk affects the experiment in an additional way since it actively introduces correlations into the result. It can be shown that this leads to a modification of the inequality that raises the bound on the value of S that can be achieved with a locally realistic hidden variable theory [Kofman et al., 2007]:

$$-2 + 4\,\min\{p_c^a,\, p_c^b\} \leq S \leq 2 + 2\left|p_c^a - p_c^b\right| \tag{11.4}$$

Here, p_c^a (p_c^b) is the classical probability that a tunneling of qubit A (B) causes a tunneling of qubit B (A), i.e. the probability that the state $|\,10\,\rangle$ ($|\,01\,\rangle$) is measured as $|\,11\,\rangle$ for reasons other than the non-ideal measurement fidelity of qubit B (A).

This effect on the inequality itself makes measurement crosstalk very undesirable as it significantly complicates the justification of a claimed violation. To address this issue, a different coupling scheme can be used that allows the qubits to be decoupled during the measurement. Since controllable coupling was not available at the time of the implementation of this experiment, a temporary solution can be found by coupling the qubits through a coplanar resonator.

11.4.2 Resonator Coupling

Resonant buses are frequently used in other qubit designs (like the charge qubit) to enable coupling despite the qubits' high impedance [Majer et al., 2007]. Here, the purpose of the resonator instead is to act as a band-pass filter for the coupling. Since the qubits are far off resonance from the pass-band during

Table 11.10: Bell Violation Results Total – Resonator Coupling

Parameter	ab	$a'b$	ab'	$a'b'$
$P_{\|00\rangle}$	0.4162	0.3978	0.1046	0.3612
$P_{\|01\rangle}$	0.1575	0.1759	0.3700	0.1136
$P_{\|10\rangle}$	0.0852	0.0731	0.3904	0.1185
$P_{\|11\rangle}$	0.3412	0.3531	0.1350	0.4066
E	0.5147	0.5019	-0.5208	0.5358
S	2.0732			

measurement, this shields them very effectively from unwanted excitations. Measurement crosstalk as explained in Section 3.4.2 thus is small. In fact, for the same coupling strength, this coupling method yields about two orders of magnitude less measurement crosstalk. In addition, this allows for the qubits to be strongly decoupled during the π and Bell rotation pulses. Not only does this address one additional error mechanism in the error budget above (the errors caused by always-on coupling), it also makes the experiment conceptually much cleaner as the qubits can be assumed to be much more causally disconnected during the measurement as required by the derivation of the inequality.

As shown in Table 11.10, this experiment yielded an S-value of 2.0732 for the sequence parameters shown in Table 11.11 suggesting a successful violation of the

Table 11.11: Optimization Results – Resonator Coupling Implementation

Parameter	Value	Comments
Φ_A	$-213.0\,\mathrm{mV}$	
Φ_B	$-304.95\,\mathrm{mV}$	
t_π	$8.25\,\mathrm{ns}$	FWHM of Gaussian pulse
A_π	0.293	
f_π	$6.659\,\mathrm{GHz}$	
ϕ_π	$0°$	
$dt_{\sqrt{i-Swap}}$	$-0.6\,\mathrm{ns}$	
$t_{\sqrt{i-Swap}}$	$9.92\,\mathrm{ns}$	
$A_{\sqrt{i-Swap}}$	-0.281	
$O_{\sqrt{i-Swap}}$	0.0	
dt_{i-Swap}	$0.0\,\mathrm{ns}$	
t_{i-Swap}	$12.43\,\mathrm{ns}$	
A_{i-Swap}	-0.218	
O_{i-Swap}	-0.3	
$dt_A, dt_{MP_A}, dt_{MP_B}$	$-1.0\,\mathrm{ns}$	
t_a, $t_{a'}$	$6.5\,\mathrm{ns}$	FWHM of Gaussian pulse
A_a	0.379	Corresponds to $\sim 149°$
f_a, $f_{a'}$	$6.750\,\mathrm{GHz}$	
ϕ_a	$185°$	
$A_{a'}$	0.442	Corresponds to $\sim 156°$
$\phi_{a'}$	$13°$	
A_{MP_A}	0.533	
t_{MP_A}, t_{MP_B}	$10.0\,\mathrm{ns} + 70.0\,\mathrm{ns}$	10 ns flattop + 70 ns ramp
dt_B	$-14.0\,\mathrm{ns}$	
t_b, $t_{b'}$	$5.75\,\mathrm{ns}$	FWHM of Gaussian pulse
A_b	0.003	Corresponds to $\sim 1°$
f_b, $f_{b'}$	$6.651\,\mathrm{GHz}$	
ϕ_b	$143°$	
$A_{b'}$	0.257	Corresponds to $\sim 92°$
$\phi_{b'}$	$166°$	
A_{MP_B}	0.497	

Bell inequality. To make this claim, the number needs to be supplemented with an estimate of its standard error as described in Section 11.2.3. Since the experiment was run over roughly 8 hours to collect around 34.1 million statistical samples for each of the involved probabilities, the standard error is most likely dominated by $1/f$ noise and drifts in the experiment. Thus, one would need to model these error mechanisms in order to obtain a meaningful estimate for the standard error. This can be avoided by breaking the dataset into sections for which the standard error is known to be dominated by statistical sampling noise.

11.5 Analysis and Verification

11.5.1 Standard Error of the S-Value

To find the right section size, the dataset is tentatively split into n sections. Each section is then divided into two halves that each yield an S-value $S_{i,1}$ and $S_{i,2}$. The difference between these can be used as a drift-sensitive estimate of the internal variance for each section:

$$v_i = (S_{i,2} - S_{i,1})^2 \tag{11.5}$$

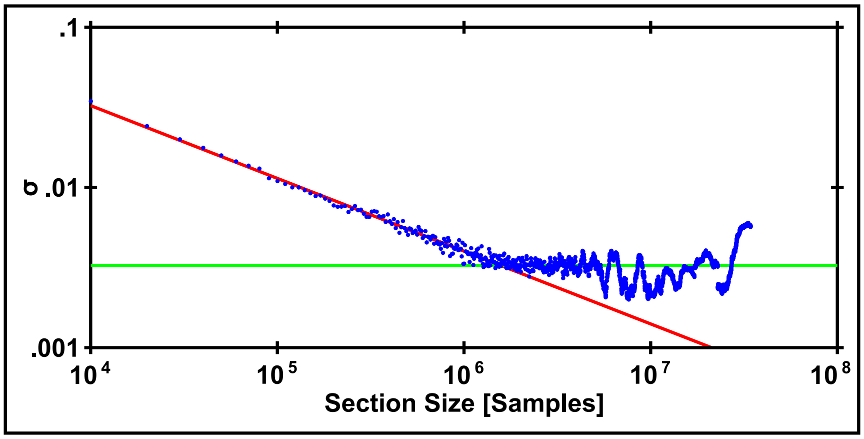

Figure 11.3: Standard Error Analysis: As the sample size increases, the standard error of the estimated mean shifts from being dominated by statistical sampling noise (red line) to being dominated by $1/f$ drift in the experiments (green line). The point where the two lines cross gives the maximum sample size that can be statistically analyzed in a meaningful way without modeling the $1/f$ noise.

The variances v_i are then averaged over the entire dataset to give an estimate of the standard error:

$$\sigma_n = \sqrt{\frac{1}{n}\sum_{i=1}^{n} v_i} \tag{11.6}$$

The resulting σ_n's as a function of the section size $\frac{N}{n}$ are shown in Figure 11.3. As the section size is increased, the statistical noise in the estimate of S_i goes down, leading to the expected $\frac{1}{\sqrt{N/n}}$ decrease in σ_n. Eventually, the sections become large enough for drifts to dominate their internal variance (as defined by Equation 11.5) and the resulting σ_n's level off. If straight lines are fitted through each of these two regions, their point of intersection gives an estimate of the maximum acceptable sample size, in this case 1.55 million samples, or about

Table 11.12: Bell Violation Results By Section – Resonator Coupling

Section	S	Violation
1	2.06952 ± 0.00138	50.3σ
2	2.07227 ± 0.00134	53.9σ
3	2.07003 ± 0.00139	50.3σ
4	2.07249 ± 0.00137	53.1σ
5	2.06769 ± 0.00141	48.0σ
6	2.06661 ± 0.00143	46.7σ
7	2.07200 ± 0.00144	50.1σ
8	2.07419 ± 0.00142	52.4σ
9	2.06758 ± 0.00144	46.8σ
10	2.06933 ± 0.00146	47.6σ
11	2.07177 ± 0.00138	52.1σ
12	2.07251 ± 0.00130	55.8σ
13	2.07573 ± 0.00144	52.6σ
14	2.07739 ± 0.00137	56.3σ
15	2.07413 ± 0.00142	52.1σ
16	2.07482 ± 0.00146	51.4σ
17	2.07679 ± 0.00138	55.7σ
18	2.07646 ± 0.00140	54.6σ
19	2.08055 ± 0.00135	59.5σ
20	2.07769 ± 0.00137	56.9σ
21	2.07438 ± 0.00138	54.0σ
22	2.07197 ± 0.00145	49.7σ
All	2.07320 ± 0.00030	244.0σ

20 minutes worth of data. Within each section, the standard error can now be assumed to be dominated by statistical sampling noise, i.e. the errors on S are independent of each other. The standard analysis technique described in Section 11.2.3 then yields 22 S-values with their standard errors as shown in Table 11.12. With this, we can now claim a violation of the Bell inequality by 59.5σ with an S-value of 2.08055 ± 0.00135 based on section 19.

Each section's S_i and σ_i imply a probability (erfc $\frac{S_i - 2.0}{\sigma_i} \approx 10^{-1200}$) for the actual S-value of that section to be less than 2.0. The probability for the actual S-value of the combined dataset to be less than 2.0 is then the product of the probabilities for the individual sections:

$$P_{S<2.0} = \prod_{i=1}^{22} P_{S_i<2.0} = \prod_{i=1}^{22} \text{erfc}\, \frac{S_i - 2.0}{\sigma_i} = 1.27 \times 10^{-26253} \tag{11.7}$$

This probability can be used to calculate the corresponding standard error for the average S-value for the combined dataset using:

$$P_{S<2.0} = \text{erfc}\, \frac{S - 2.0}{\sigma} = \text{erfc}\, \frac{0.07320}{\sigma} = 1.27 \times 10^{-26253} \;\rightarrow\; \sigma = 0.00030 \tag{11.8}$$

With this, the entire dataset shows a violation of 244.0σ with an S-value of 2.07320 ± 0.00030. It is interesting to note that the standard error estimated in this way is very close to the value estimated if the entire dataset is treated as one single section.

11.5.2 Dependence of S on Sequence Parameters

Since, for example, an experiment without measurement pulses yields an S-value of 2.0 as all states are read as $|\,00\,\rangle$, it is possible for the optimization process to get stuck in a local maximum that shows a maximized classical correlation but little or no quantum entanglement. Any artificially introduced correlations during the measurement might then yield a value of $S > 2.0$ and thus a false claim of a violation.

A simple check of the behavior of S versus one of the sequence parameters is useful to quickly expose such major problems in the experiment. Figure 11.4a plots the dependence of S on the phase of the Bell rotation pulses (the plane in which the qubit is measured) on the second qubit. The data shows the sinusoidal response predicted by quantum mechanics, and thus provides strong evidence that the experiment is implemented in the expected way.

Due to the short coherence times of the involved qubits, it was necessary to minize the overall sequence length as much as possible. This included moving the Bell rotations and measurement on the second qubit forward to place them right after the pulse that entangles the qubit with the resonator. Even though the short distance between the qubits relative to the time-scales of measurement makes it impossible to close the locality loophole in this experimental setup anyway,

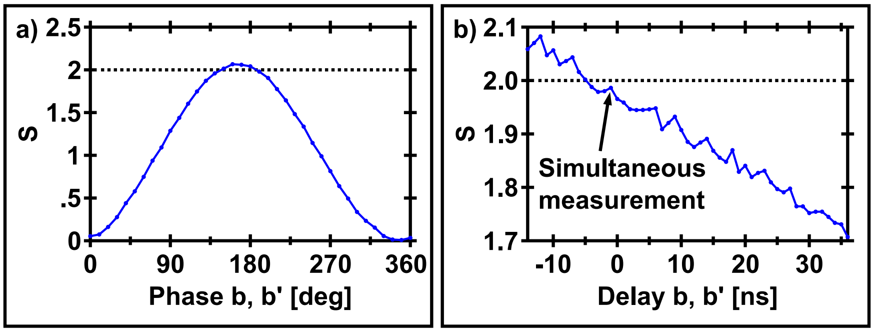

Figure 11.4: Behavior of S: Examining the behavior of S versus sequence parameters can verify a trustworthy implementation – a) S versus phase b, b': As the phase of the rotation pulses on the second qubit is varied, S shows the expected sinusoidal response. b) S versus measurement delay: The fact that S shows only the expected T_1 and T_2 decay, but no other features indicating a dependence on the relative timing of the measurements, is strong evidence that the qubits are truly decoupled during measurement.

the early measurement of one qubit might still prompt criticism as it opens the locality loophole even wider by giving the qubits even more time to interact via the unknown processes postulated by the loophole. To counter this criticism, it can be shown that the relative timing of the measurement does not influence the resulting S-value in a way that suggests the existence of an additional interaction allowed by the time difference. Figure 11.4b shows the dependence of S on the delay of the Bell rotations and measurement on the second qubit. As expected from the additional qubit decoherence, S decreases as the measurement is delayed. But the plot shows no features that correlate with the relative timing of the measurements on the two qubits. Specifically, there is no reduction in S around $t = -1\,\mathrm{ns}$, the point where the measurement happens simultaneously. This dataset can therefore be seen as strong evidence against a problem introduced by the early measurement.

11.5.3 Microwave and Measurement Crosstalk

The most important step in the verification of the claim is the analysis of known mechanisms that introduce artificial correlations into the result during the measurement. The experimental setup at hand is susceptible to two of these: Microwave and measurment crosstalk.

Microwave crosstalk results from insufficient electrical isolation of the two qubits that allows a microwave drive applied to one qubit to be seen by the other. Earlier investigations have shown this microwave crosstalk to be suppressed by about 20 dB, i.e. the second qubit sees about 1% of the drive applied to the first qubit. Since, in this experiment, the qubits are placed off-resonance from each other by at least 100 MHz, even these leaked microwaves are not able to have any effect on the "wrong" qubit. Therefore, microwave crosstalk does not constitute a problem for this experiment.

Measurement crosstalk, on the other hand, is still present despite the band-pass filtering provided by the resonator coupling. Figure 11.5 shows the result of experiments that quantify this crosstalk. In the experiment, a Rabi oscillation is driven on one of the qubits to cause it to be alternatingly measured as $|\,1\,\rangle$ or $|\,0\,\rangle$.

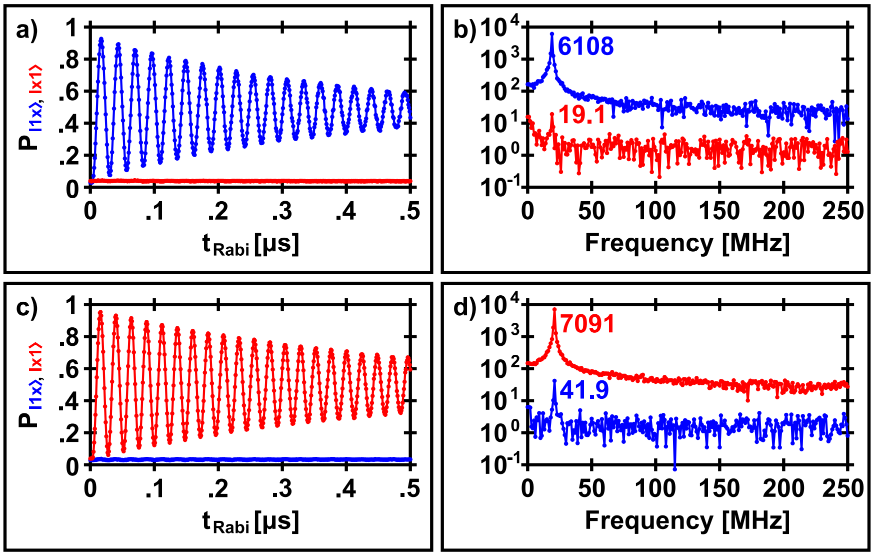

Figure 11.5: Quantifying Measurement Crosstalk: Measurement crosstalk can be quantified by driving a Rabi oscillation on one qubit and observing the other qubit's response. Fourier transforming the data allows the isolation of the relevant features. – a) Rabi oscillation on the first qubit: The measured state of the second qubit only shows a very weak dependence on whether the first qubit is in the $|\,1\,\rangle$ or $|\,0\,\rangle$ state. b) Fourier transform of a: The ratio of the responses of the two qubits at the same frequency as the Rabi oscillation on the first gives a number for the measurement crosstalk, here: $\frac{19.1}{6108} = 0.31\%$. c) Rabi oscillation on second qubit. d) Fourier transform of c: The data shows $\frac{41.9}{7091} = 0.59\%$ measurement crosstalk.

The second qubit is not driven and should thus remain in the $|\,0\,\rangle$ state.

Measurement crosstalk causes the second qubit to be sometimes measured as $|\,1\,\rangle$ conditionally on the first qubit being measured as $|\,1\,\rangle$. Therefore, measurement crosstalk should introduce a small oscillation of $P_{|\,x1\,\rangle}$ at the same phase and frequency of the oscillation of $P_{|\,1x\,\rangle}$ caused by the Rabi drive. Fourier transforming the measurements of both qubits exposes this oscillation as a peak at the same frequency as the peak on the driven qubit. The amplitude ratio of the two peaks then gives a number for the strength of the measurement crosstalk. Here, the crosstalk is less than 1% in either direction.

As mentioned above, measurement crosstalk causes a correction to the limits on S that are achievable by a hidden variable theory. According to Equation 11.4 the crosstalk measured here yields the following new limit on S:

$$S \leq 2 + 2\left|p_c^a - p_c^b\right| = 2 + 2\left|0.0059 - 0.0031\right| = 2.0056 \tag{11.9}$$

This new limit lowers the violations quoted in Table 11.12 by about 10%. For example, the S-value and standard error of section 19 imply a violation in the presence of measurement crosstalk of 55.5σ rather than the quoted 59.5σ. This correction is sufficiently small to not challenge the underlying claim of a violation.

11.5.4 Numerical Simulation

To understand whether the obtained S-value makes sense, it is useful to simulate the experiment numerically using the techniques described in Chapter 3. For accurate results, the simulation must include the resonator used for the coupling. In general, a resonator needs to be simulated as a harmonic oscillator along with its higher excited states. In this experiment, though, there is always only one photon available during the coupling between a qubit and the resonator, and therefore the higher levels in the resonator will never be excited. This allows us to treat the resonator simply as a third qubit, yielding a combined system with the eight possible states $|\,000\,\rangle$, $|\,001\,\rangle$, $|\,010\,\rangle$, ..., $|\,111\,\rangle$, descibed by an 8×8 density matrix.

The system is initialized in the $|\,000\,\rangle$ state, which corresponds to a density matrix with a single 1 in the top left corner and all other elements equal to 0. According to the sequence used, the second qubit is then excited into the $|\,1\,\rangle$ state (overall: $|\,001\,\rangle$) with a π-pulse:

$$\mathbf{A}_\pi = e^{-i\pi\left(\frac{180^\circ}{360^\circ}(\mathbf{I}\otimes\mathbf{I}\otimes\sigma_x)\right)}\,|\,000\,\rangle = |\,001\,\rangle \tag{11.10}$$

Next, the second qubit is coupled with the resonator for enough time to cause half of a swap operation:

$$\mathbf{A}_{\mathbf{entangled}} = e^{-i\pi\left(\frac{90^\circ}{360^\circ}(\mathbf{I}\otimes\mathbf{C})\right)}\,\mathbf{A}_\pi = \frac{|\,001\,\rangle + e^{i\phi}|\,010\,\rangle}{\sqrt{2}} \tag{11.11}$$

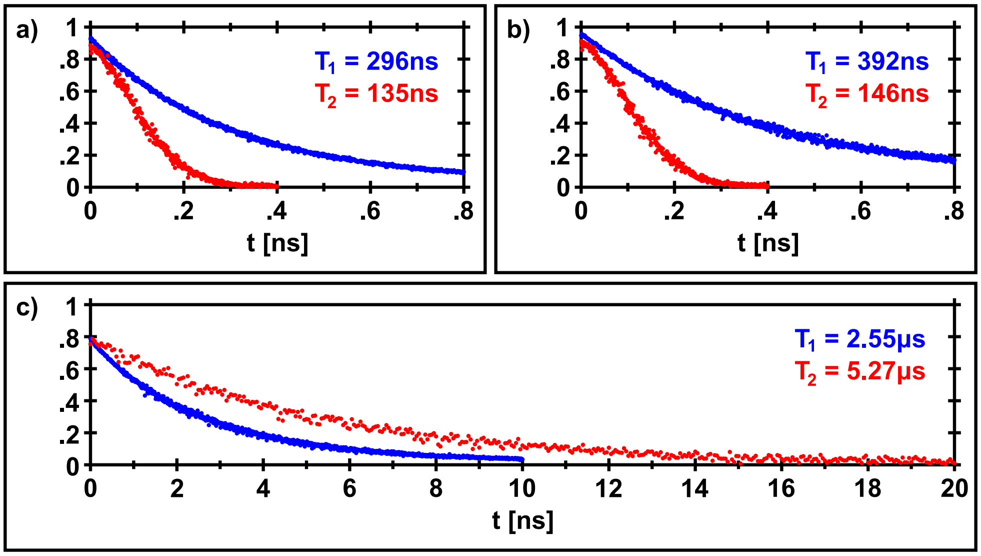

Figure 11.6: Resonator Coupled Sample Specifications: Measurement of T_1 (blue) and T_2 (red) of the different components in the resonator coupled sample. – a) Qubit A. b) Qubit B. c) Resonator.

The entanglement is then swapped from the resonator to the first qubit:

$$\mathbf{A_{swap}} = e^{-i\pi\left(\frac{180^\circ}{360^\circ}(\mathbf{C}\otimes\mathbf{I})\right)}\,\mathbf{A_{entangled}} = \frac{|\,001\,\rangle + e^{i\phi}|\,100\,\rangle}{\sqrt{2}} \tag{11.12}$$

Finally, the qubits are subjected to the required Bell rotations:

$$\mathbf{A_{ab}} = e^{-i\pi\left(-\frac{135^\circ}{360^\circ}(\sigma_x\otimes\mathbf{I}\otimes\mathbf{I})+\frac{0^\circ}{360^\circ}(\mathbf{I}\otimes\mathbf{I}\otimes\sigma_x)\right)}\,\mathbf{A_{swap}} \tag{11.13}$$

$$\mathbf{A_{a'b}} = e^{-i\pi\left(\frac{135^\circ}{360^\circ}(\sigma_x\otimes\mathbf{I}\otimes\mathbf{I})+\frac{0^\circ}{360^\circ}(\mathbf{I}\otimes\mathbf{I}\otimes\sigma_x)\right)}\,\mathbf{A_{swap}} \tag{11.14}$$

$$\mathbf{A_{ab'}} = e^{-i\pi\left(-\frac{135^\circ}{360^\circ}(\sigma_x\otimes\mathbf{I}\otimes\mathbf{I})-\frac{90^\circ}{360^\circ}(\mathbf{I}\otimes\mathbf{I}\otimes\sigma_x)\right)}\,\mathbf{A_{swap}} \tag{11.15}$$

$$\mathbf{A_{a'b'}} = e^{-i\pi\left(\frac{135^\circ}{360^\circ}(\sigma_x\otimes\mathbf{I}\otimes\mathbf{I})-\frac{90^\circ}{360^\circ}(\mathbf{I}\otimes\mathbf{I}\otimes\sigma_x)\right)}\,\mathbf{A_{swap}} \tag{11.16}$$

Table 11.13: Qubit Sample Parameters – Resonator Coupling Implementation

Parameter	Value	Source
Qubit A:		
T_1	296 ns	Figure 11.6a
T_2	135 ns	Figure 11.6a
T_φ	175 ns	T_1 and T_2
$F_{\lvert 0 \rangle}$	97.04%	Figure 11.7a
$F_{\lvert 1 \rangle}$	96.32%	Figure 11.7a
Qubit B:		
T_1	392 ns	Figure 11.6b
T_2	146 ns	Figure 11.6b
T_φ	179 ns	T_1 and T_2
$F_{\lvert 0 \rangle}$	96.18%	Figure 11.7b
$F_{\lvert 1 \rangle}$	98.42%	Figure 11.7b
Resonator:		
T_1	$2,552$ ns	Figure 11.6c
T_2	$5,266$ ns	Figure 11.6c
T_φ	∞	T_1 and T_2
Coupling:		
Qubit A ↔ resonator	36.2 MHz	Figure 9.8b
Qubit B ↔ resonator	26.1 MHz	Figure 9.9b
Measurement Crosstalk:		
Qubit A → qubit B	0.31%	Figure 11.5b
Qubit B → qubit A	0.59%	Figure 11.5d

From the resulting states, the 16 probabilities $P_{ab}(|\,00\,\rangle)$, $P_{ab}(|\,01\,\rangle)$, … are extracted and combined to yield S. If the simulation is run as described, i.e. without including imperfections, it yields the expected value of $S = 2.828$. Using the Kraus operators K_{1a} and K_{1b} as described in Section 3.4.5, energy decay can be added to the simulation using the values for T_1 from Table 11.13 and the pulse lengths from Table 11.11. This lowers the resulting S-value to $S = 2.443$. If dephasing is added as well, using $K_{\varphi a}$ and $K_{\varphi b}$, the value further decreases to $S = 2.247$. The final error mechanism to include are the errors caused by non-ideal measurement fidelities, which yields $S = 1.984$.

Since this simulation corresponds to an experiment where the Bell rotations and measurement happen simultaneously on both qubits, this S-Value needs to be compared to the one shown in Figure 11.4b at $t = -1\,\mathrm{ns}$: $S = 1.986$. The agreement is remarkably good.

If the simulation is modified to move the Bell rotations and measurement on the second qubit forward as done in the actual experiment, it yields $S = 2.064$. The corresponding error budget is shown in Table 11.14. The observation that the experimental result is even slightly better than the prediction by simulation is most likely rooted in that fact that the theoretical rotation angles ($\alpha = -135°$, $\alpha' = 135°$, $\beta = 0°$, and $\beta' = -90°$) used in the simulation are not actually optimal in the presence of decoherence and imperfect measurement. Also, the

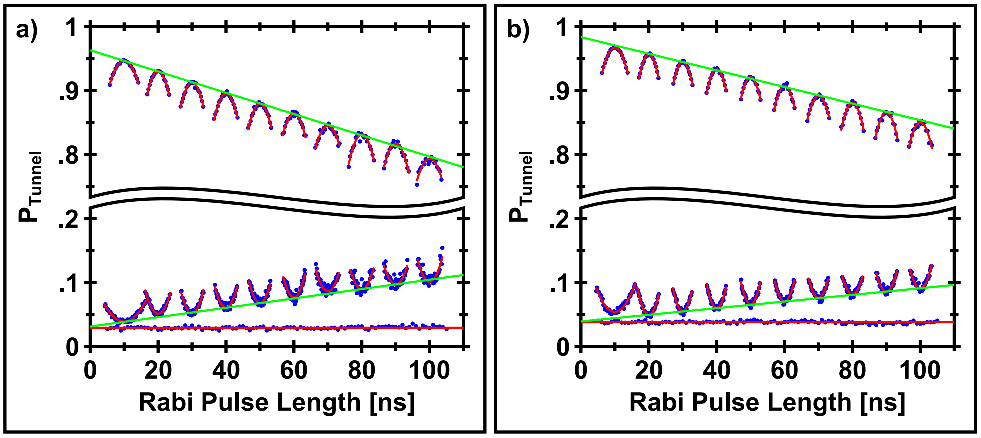

Figure 11.7: Visibility Analysis – Resonator Coupled Sample: Composite of several datasets. Blue dots represent data, red lines are fits through the data, and green lines are fits through the extrema of the red lines. The upper parabolas correspond to Rabi oscillations driven with pulses at fixed length and increasing amplitude (Power Rabis) around the point where they yield a π-pulse. The bottom parabolas are Power Rabis driven around the point where they yield a 2π pulse. The horizontal dataset at the bottom corresponds to no drive on the qubit. The green fits through the parabolas' extrema (optimal π or 2π pulses) give the measurement visibility when extrapolated to $t = 0$, i.e. to an optimal, instantaneous pulse. The horizontal line checks the method by providing a direct measurement of the $|\,0\,\rangle$ state visibility. Since the measurements obtained agree to high precision, the method can be trusted to extract a $|\,1\,\rangle$ state fidelity for which no direct measurement is available. – a) Qubit A: $F_{|\,0\,\rangle,\,Rabis} = 96.86\%$, $F_{|\,0\,\rangle,\,Direct} = 97.04\%$, $F_{|\,1\,\rangle} = 96.32\%$. b) Qubit B: $F_{|\,0\,\rangle,\,Rabis} = 96.06\%$, $F_{|\,0\,\rangle,\,Direct} = 96.18\%$, $F_{|\,1\,\rangle} = 98.42\%$.

Table 11.14: Error Budget – Resonator Coupling Implementation

Error	Contribution	New S
Theoretical		2.828
Ideal simulation	0.000	2.828
Energy decay (T_1)	0.328	2.500
Dephasing (T_φ)	0.163	2.337
Measurement Fidelities	0.273	2.064
Experimental result		2.073

simulation models dephasing via an exponential decay, while in the experiment the decay follows a Gaussian profile, which, at short timescales, causes smaller errors. Nevertheless, the agreement is remarkable and is a strong indication that the experiment and all supporting datasets are reasonable.

11.5.5 Measurement Correction

Since the non-ideal measurement fidelities are caused by classically probabilistic events that occur independently on the two qubits, it is possible to correct the measurement mathematically to extract an estimate of the S-value that would have been achieved with perfect measurement. This corrected S-value then provides an idea of how well the quantum operations were performed on the qubit pair, independent of the measurement. As explained in Section 3.4.1, non-ideal measurement fidelities can be understood simply as a probability to misidentify the qubit state. This can be written in the form of a matrix equation that expresses the measured probabilities $\mathbf{P_M} = (P_M(|\,0\,\rangle), P_M(|\,1\,\rangle))$ as a function of the actual probabilities $\mathbf{P_A} = (P_A(|\,0\,\rangle), P_A(|\,1\,\rangle))$ and the fidelities $F_{|\,0\,\rangle}$ and $F_{|\,1\,\rangle}$:

$$\mathbf{P_M} = \begin{bmatrix} F_{|\,0\,\rangle} & 1 - F_{|\,1\,\rangle} \\ 1 - F_{|\,0\,\rangle} & F_{|\,1\,\rangle} \end{bmatrix} \mathbf{P_A} \tag{11.17}$$

This equation can be inverted to obtain the actual probabilities:

$$\mathbf{P_A} = \begin{bmatrix} F_{|\,0\,\rangle} & 1 - F_{|\,1\,\rangle} \\ 1 - F_{|\,0\,\rangle} & F_{|\,1\,\rangle} \end{bmatrix}^{-1} \mathbf{P_M} \tag{11.18}$$

For a coupled qubit system, this can be expanded to estimate the actual probabilities $\mathbf{P_A^c} = (P_A^c(|\,00\,\rangle), P_A^c(|\,01\,\rangle), P_A^c(|\,10\,\rangle), P_A^c(|\,11\,\rangle))$ from the measured probabilities $\mathbf{P_M^c} = (P_M^c(|\,00\,\rangle), P_M^c(|\,01\,\rangle), P_M^c(|\,10\,\rangle), P_M^c(|\,11\,\rangle))$ given the fidelities on the two qubits:

$$\mathbf{P_A^c} = \begin{bmatrix} F^A_{|\,0\,\rangle} & 1 - F^A_{|\,1\,\rangle} \\ 1 - F^A_{|\,0\,\rangle} & F^A_{|\,1\,\rangle} \end{bmatrix}^{-1} \otimes \begin{bmatrix} F^B_{|\,0\,\rangle} & 1 - F^B_{|\,1\,\rangle} \\ 1 - F^B_{|\,0\,\rangle} & F^B_{|\,1\,\rangle} \end{bmatrix}^{-1} \mathbf{P_M^c} \tag{11.19}$$

Applying this correction to the measurements shown in Table 11.10 using the measurement fidelities from Table 11.13 yields an estimated S-value of $S = 2.3552$ (Table 11.15) in good agreement with the simulated value of $S = 2.337$.

Table 11.15: Bell Violation Results Corrected – Resonator Coupling

Parameter	ab	$a'b$	ab'	$a'b'$
$P_{\|00\rangle}$	0.4406	0.4213	0.0900	0.3813
$P_{\|01\rangle}$	0.1343	0.1539	0.3790	0.0880
$P_{\|10\rangle}$	0.0726	0.0599	0.4166	0.1092
$P_{\|11\rangle}$	0.3525	0.3649	0.1145	0.4215
E	0.5862	0.5724	-0.5911	0.6055
S	2.3552			

Chapter 12

Conclusion

12.1 Claim of Violation of Bell's Inequality

Based on the experimental results and verifications detailed above, we claim the observation of a violation of the CHSH Bell inequality by over 200σ in our resonator-coupled superconducting Josephson phase qubit circuit. To our knowledge, this is the first violation observed in a macroscopic solid state system.

If our dataset is limited to a size where statistical sampling noise (rather than experimental drift) dominates the errors in the obtained S-value (1.55 million samples over 20 minutes), we find $S = 2.08055 \pm 0.00135$ (in the best case), corresponding to a violation of the bound $S \leq 2.0$ by 59.5σ. If the entire dataset (34.1 million samples over 8 hours) is used, we find $S = 2.07320 \pm 0.00030$, corresponding to a violation by 244.0σ. In the latter case, the standard error on S is estimated using a combined-probability argument.

The analysis of measurement crosstalk in the experiment shows asymmetric crosstalk magnitudes of 0.31% from qubit A to qubit B and 0.59% from qubit B to qubit A. This leads to a correction in the positive bound on S achievable by a hidden variable theory to $S \leq 2.0056$ instead of $S \leq 2.0$, reducing the above mentioned violations from 59.5σ to 55.5σ and from 244.0σ to 225.3σ

Correcting for non-ideal fidelities during the tunneling measurement, we estimate that the entangled pair of qubits before the tunneling measurement shows an S-value of $S = 2.337$.

All obtained results can be explained to very good agreement with quantum simulations including energy decay (T_1), dephasing (T_φ), and non-ideal measurement fidelities as the only imperfections.

12.2 S-Value as Qubit Pair Benchmark

Given the sensitivity of the measured S-value to every single control and performance parameter of a coupled qubit pair, we suggest that it could be used as a powerful single-number performance benchmark usable as the basis of direct comparison of different qubit architectures. For this, the value of S itself provides information about the quality of the operations performed on the qubit pair, while its standard error gives information about the stability of the implementing architecture.

Furthermore, being able to demonstrate a violation of a Bell inequality is a strong indication that the implementing system can support quantum entanglement that goes beyond classically achievable correlations. Since entanglement is the basis for the exponential performance scaling of quantum computers, this is an important demonstration to verify the validity of the proposed architecture.

12.3 Josephson Phase Qubit Performance

The successful implementation of this experiment required exquisite control over our coupled qubit pair, with gate fidelities in the upper 90% range. Simulations suggest that the reduction in the measured S-value as compared to the theoretical optimum of $S_{max} = 2.828$ is the result of primarily single qubit performance characteristics. The capacitive resonator coupling scheme used in the experiment does not seem to introduce an additional source of imperfections. Thus, we believe that this experiment demonstrates that our expectations about the good scalability of our system to many qubits are warranted. Efforts to improve qubit performance should therefore be targetted at the single-qubit level to reduce energy decay and dephasing and to increase measurement fidelities.

12.4 Future Direction

By improving single qubit performance, it should be possible to significantly increase the measured value of S to eventually bring it close to the theoretically achievable maximum of $S_{max} = 2.828$. In the meantime, the qubits' performance is sufficient to allow for the implementation of many other interesting experiments, like the generation of GHZ or Werner states, etc.

Since scalability does not seem to be a problem for the system, it makes sense to explore experiments with more than two qubits in parallel with efforts to improve single qubit performance. Care must be taken in the design of qubit coupling circuits to minimize measurement crosstalk as this error mechanism significantly complicates the analysis and therefore implementation of demanding experiments.

Bibliography

J. Altepeter, E. Jeffrey, and P. Kwiat. Phase-compensated ultra-bright source of entangled photons. *Opt. Express*, 13(22):8951–8959, October 2005.

B. Apolloni, C. Caravalho, and D. De Falco. Quantum stochastic optimization. *Stochastic Processes and their Applications*, 33:233–244, 1989.

Alain Aspect, Philippe Grangier, and Gérard Roger. Experimental tests of realistic local theories via bell's theorem. *Phys. Rev. Lett.*, 47(7):460–463, Aug 1981.

Adriano Barenco, Charles H. Bennett, Richard Cleve, David P. DiVincenzo, Norman Margolus, Peter Shor, Tycho Sleator, John A. Smolin, and Harald Weinfurter. Elementary gates for quantum computation. *Phys. Rev. A*, 52(5): 3457–3467, Nov 1995.

J.S. Bell. On the einstein podolsky rosen paradox. *Physics*, 1:195–200, 1964.

Jan Benhelm, Gerhard Kirchmair, Christian F. Roos, and Rainer Blatt. Towards fault-tolerant quantum computing with trapped ions. *Nat Phys*, 4(6):463–466, June 2008. ISSN 1745-2473.

Radoslaw C. Bialczak, R. McDermott, M. Ansmann, M. Hofheinz, N. Katz, Erik Lucero, Matthew Neeley, A. D. O'Connell, H. Wang, A. N. Cleland, and John M. Martinis. 1/f flux noise in josephson phase qubits. *Physical Review Letters*, 99(18):187006, 2007.

John Clarke and Frank K. Wilhelm. Superconducting quantum bits. *Nature*, 453 (7198):1031–1042, June 2008. ISSN 0028-0836.

John F. Clauser and Michael A. Horne. Experimental consequences of objective local theories. *Phys. Rev. D*, 10(2):526–535, Jul 1974.

John F. Clauser, Michael A. Horne, Abner Shimony, and Richard A. Holt. Proposed experiment to test local hidden-variable theories. *Phys. Rev. Lett.*, 23 (15):880–884, Oct 1969.

P. Coles, T. Cox, C. Mackey, and S. Richardson. The toxic terabyte - how data-dumping threatens business efficiency. *IBM Global Technical Services*, July 2006.

K. B. Cooper, Matthias Steffen, R. McDermott, R. W. Simmonds, Seongshik Oh, D. A. Hite, D. P. Pappas, and John M. Martinis. Observation of quantum oscillations between a josephson phase qubit and a microscopic resonator using fast readout. *Phys. Rev. Lett.*, 93(18):180401, Oct 2004.

D. Deutsch. Quantum theory, the church-turing principle and the universal quantum computer. In *Proceedings of the Royal Society of London*, volume A, pages 97–117, 1985.

D. Deutsch and R. Jozsa. Rapid solutions of problems by quantum computation. In *Proceedings of the Royal Society of London*, volume 439 of *A*, page 553, 1992.

David P. DiVincenzo. The physical implementation of quantum computation. arXiv:quant-ph/0002077v3, 2000.

R. C. Eberhart and J. Kennedy. A new optimizer using particle swarm theory. In *Sixth International Symposium on Micromachine and Human Science, Nagoya, Japan*, pages 39–43, 1995.

A. Einstein, B. Podolsky, and N. Rosen. Can quantum-mechanical description of physical reality be considered complete? *Phys. Rev.*, 47(10):777–780, May 1935.

Albert Einstein, Max Born, and Hedwig Born. *The Born-Einstein Letters: Friendship, Politics and Physics in Uncertain Times*, chapter Letter to Max Born on 4 December 1926. London, Macmillan, 1971.

John C. Gallop. *SQUIDS, the Josephson Effects and Superconducting Electronics.* CRC Press, 1990.

Scott Hill and William K. Wootters. Entanglement of a pair of quantum bits. *Phys. Rev. Lett.*, 78(26):5022–5025, Jun 1997.

Max Hofheinz, E. M. Weig, M. Ansmann, Radoslaw C. Bialczak, Erik Lucero, M. Neeley, A. D. O/'Connell, H. Wang, John M. Martinis, and A. N. Cleland. Generation of fock states in a superconducting quantum circuit. *Nature*, 454 (7202):310–314, July 2008. ISSN 0028-0836.

Max Hofheinz, H. Wang, M. Ansmann, Radoslaw C. Bialczak, Erik Lucero, M. Neeley, A. D. O'Connell, D. Sank, J. Wenner, John M. Martinis, and A. N. Cleland. Synthesizing arbitrary quantum states in a superconducting resonator. *Nature*, 459(7246):546–549, May 2009. ISSN 0028-0836.

B. D. Josephson. Possible new effects in superconductive tunnelling. *Physics Letters*, 1(7):251 – 253, 1962. ISSN 0031-9163.

A. G. Kofman and A. N. Korotkov. Bell-inequality violation versus entanglement in the presence of local decoherence. *Physical Review A (Atomic, Molecular, and Optical Physics)*, 77(5):052329, 2008a.

Abraham G. Kofman and Alexander N. Korotkov. Analysis of bell inequality violation in superconducting phase qubits. *Physical Review B (Condensed Matter and Materials Physics)*, 77(10):104502, 2008b.

Abraham G. Kofman, Qin Zhang, John M. Martinis, and Alexander N. Korotkov. Theoretical analysis of measurement crosstalk for coupled josephson phase qubits. *Physical Review B (Condensed Matter and Materials Physics)*, 75(1):014524, 2007.

Karl Kraus. *States, effects, and operations : fundamental notions of quantum theory : lectures in mathematical physics at the University of Texas at Austin.* Springer-Verlag, 1983.

A. K. Lenstra and H. W. Lenstra, Jr. *The development of the number field sieve*, volume 1554. Springer Berlin / Heidelberg, 1993.

Erik Lucero, M. Hofheinz, M. Ansmann, Radoslaw C. Bialczak, N. Katz, Matthew Neeley, A. D. O'Connell, H. Wang, A. N. Cleland, and John M. Martinis. High-fidelity gates in a single josephson qubit. *Physical Review Letters*, 100(24): 247001, 2008.

A. Lupascu, C. J. M. Verwijs, R. N. Schouten, C. J. P. M. Harmans, and J. E. Mooij. Nondestructive readout for a superconducting flux qubit. *Phys. Rev. Lett.*, 93(17):177006, Oct 2004.

J. Majer, J. M. Chow, J. M. Gambetta, Jens Koch, B. R. Johnson, J. A. Schreier, L. Frunzio, D. I. Schuster, A. A. Houck, A. Wallraff, A. Blais, M. H. Devoret, S. M. Girvin, and R. J. Schoelkopf. Coupling superconducting qubits via a cavity bus. *Nature*, 449(7161):443–447, September 2007. ISSN 0028-0836.

J. B. Majer, F. G. Paauw, A. C. J. ter Haar, C. J. P. M. Harmans, and J. E. Mooij. Spectroscopy on two coupled superconducting flux qubits. *Physical Review Letters*, 94(9):090501, 2005.

John M. Martinis, S. Nam, J. Aumentado, and C. Urbina. Rabi oscillations in a large josephson-junction qubit. *Phys. Rev. Lett.*, 89(11):117901, Aug 2002.

John M. Martinis, K. B. Cooper, R. McDermott, Matthias Steffen, Markus Ansmann, K. D. Osborn, K. Cicak, Seongshik Oh, D. P. Pappas, R. W. Simmonds, and Clare C. Yu. Decoherence in josephson qubits from dielectric loss. *Phys. Rev. Lett.*, 95(21):210503, Nov 2005.

R. McDermott, R. W. Simmonds, Matthias Steffen, K. B. Cooper, K. Cicak, K. D. Osborn, Seongshik Oh, D. P. Pappas, and John M. Martinis. Simultaneous state measurement of coupled josephson phase qubits. *Science*, 307(5713):1299–1302, 2005.

D. L. Moehring, M. J. Madsen, B. B. Blinov, and C. Monroe. Experimental bell inequality violation with an atom and a photon. *Phys. Rev. Lett.*, 93(9):090410, Aug 2004.

Graham T.T. Molitor. *Communications tomorrow: the coming of the information society: selections from The futurist*, chapter The Information Society: The Path to Post-Industrial Growth. Bethesda, Md.: World Future Society, 1982.

Gordon E. Moore. Cramming more components onto integrated circuits. *Electronics*, 38(8), April 19 1965.

Y. Nakamura, Yu. A. Pashkin, and J. S. Tsai. Coherent control of macroscopic quantum states in a single-cooper-pair box. *Nature*, 398(6730):786–788, April 1999. ISSN 0028-0836.

Matthew Neeley, M. Ansmann, Radoslaw C. Bialczak, M. Hofheinz, N. Katz, Erik Lucero, A. O/'Connell, H. Wang, A. N. Cleland, and John M. Martinis. Process tomography of quantum memory in a josephson-phase qubit coupled to a two-level state. *Nat Phys*, 4(7):523–526, July 2008a. ISSN 1745-2473.

Matthew Neeley, M. Ansmann, Radoslaw C. Bialczak, M. Hofheinz, N. Katz, Erik Lucero, A. O'Connell, H. Wang, A. N. Cleland, and John M. Martinis. Transformed dissipation in superconducting quantum circuits. *Physical Review B (Condensed Matter and Materials Physics)*, 77(18):180508, 2008b.

J. A. Nelder and R. Mead. A simplex method for function minimization. *The Computer Journal*, 7(4):308–313, January 1965.

T. P. Orlando, J. E. Mooij, Lin Tian, Caspar H. van der Wal, L. S. Levitov, Seth Lloyd, and J. J. Mazo. Superconducting persistent-current qubit. *Phys. Rev. B*, 60(22):15398–15413, Dec 1999.

Philip M. Pearle. Hidden-variable example based upon data rejection. *Phys. Rev. D*, 2(8):1418–1425, Oct 1970.

C. F. Roos, G. P. T. Lancaster, M. Riebe, H. Häffner, W. Hänsel, S. Gulde, C. Becher, J. Eschner, F. Schmidt-Kaler, and R. Blatt. Bell states of atoms with ultralong lifetimes and their tomographic state analysis. *Phys. Rev. Lett.*, 92(22):220402, Jun 2004.

M. A. Rowe, D. Kielpinski, V. Meyer, C. A. Sackett, W. M. Itano, C. Monroe, and D. J. Wineland. Experimental violation of a bell's inequality with efficient detection. *Nature*, 409(6822):791–794, February 2001. ISSN 0028-0836.

E. Schrödinger. An undulatory theory of the mechanics of atoms and molecules. *Phys. Rev.*, 28(6):1049–1070, Dec 1926.

S. Sendelbach, D. Hover, A. Kittel, M. Mück, John M. Martinis, and R. McDermott. Magnetism in squids at millikelvin temperatures. *Physical Review Letters*, 100(22):227006, 2008.

Peter W. Shor. Scheme for reducing decoherence in quantum computer memory. *Phys. Rev. A*, 52(4):R2493–R2496, Oct 1995.

Peter W. Shor. Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. *SIAM Journal on Computing*, 26:1484–1509, 1997.

R. W. Simmonds, K. M. Lang, D. A. Hite, S. Nam, D. P. Pappas, and John M. Martinis. Decoherence in josephson phase qubits from junction resonators. *Phys. Rev. Lett.*, 93(7):077003, Aug 2004.

Matthias Steffen, M. Ansmann, Radoslaw C. Bialczak, N. Katz, Erik Lucero, R. McDermott, Matthew Neeley, E. M. Weig, A. N. Cleland, and John M. Martinis. Measurement of the entanglement of two superconducting qubits via state tomography. *Science*, 313(5792):1423–1425, 2006.

A. M. Turing. On computable numbers, with an application to the entscheidungsproblem. a correction. *Proc. London Math. Soc.*, s2-43(6):544–546, 1938.

Lieven M. K. Vandersypen, Matthias Steffen, Gregory Breyta, Costantino S. Yannoni, Mark H. Sherwood, and Isaac L. Chuang. Experimental realization of shor's quantum factoring algorithm using nuclear magnetic resonance. *Nature*, 414(6866):883–887, December 2001. ISSN 0028-0836.

A. Wallraff, D. I. Schuster, A. Blais, L. Frunzio, J. Majer, M. H. Devoret, S. M. Girvin, and R. J. Schoelkopf. Approaching unit visibility for control of a superconducting qubit with dispersive readout. *Physical Review Letters*, 95(6): 060501, 2005.

Gregor Weihs, Thomas Jennewein, Christoph Simon, Harald Weinfurter, and Anton Zeilinger. Violation of bell's inequality under strict einstein locality conditions. *Phys. Rev. Lett.*, 81(23):5039–5043, Dec 1998.

Qin Zhang, Abraham G. Kofman, John M. Martinis, and Alexander N. Korotkov. Analysis of measurement errors for a superconducting phase qubit. *Physical Review B (Condensed Matter and Materials Physics)*, 74(21):214518, 2006.