

ABSTRACT

Title of dissertation: Building and Programming
a Universal Ion Trap Quantum Computer

Caroline Margaret Figgatt
Doctor of Philosophy, 2018

Dissertation directed by: Professor Christopher Monroe
Department of Physics

Quantum computing represents an exciting frontier in the realm of information processing; it is a promising technology that may provide future advances in a wide range of fields, from quantum chemistry to optimization problems. This thesis discusses experimental results for several quantum algorithms performed on a programmable quantum computer consisting of a linear chain of five or seven trapped $^{171}\text{Yb}^+$ atomic clock ions with long coherence times and high gate fidelities. We execute modular one- and two-qubit computation gates through Raman transitions driven by a beat note between counter-propagating beams from a pulsed laser. The system's individual addressing capability provides arbitrary single-qubit rotations as well as all possible two-qubit entangling gates, which are implemented using a pulse-segmentation scheme. The quantum computer can be programmed from a high-level interface to execute arbitrary quantum circuits, and comes with a toolbox of many important composite gates and quantum subroutines.

We present experimental results for a complete three-qubit Grover quantum search algorithm, a hallmark application of a quantum computer with a well-known

speedup over classical searches of an unsorted database, and report better-than-classical performance. The algorithm is performed for all 8 possible single-result oracles and all 28 possible two-result oracles. All quantum solutions are shown to outperform their classical counterparts.

Performing parallel operations will be a powerful capability as deeper circuits on larger, more complex quantum computers present new challenges. Here, we perform a pair of 2-qubit gates simultaneously in a single chain of trapped ions. We employ a pre-calculated pulse shaping scheme that modulates the phase and amplitude of the Raman transitions to drive programmable high-fidelity 2-qubit entangling gates in parallel by coupling to the collective modes of motion of the ion chain. Ensuring the operation yields only spin-spin interactions between the desired pairs, with neither residual spin-motion entanglement nor crosstalk spin-spin entanglement, is a nonlinear constraint problem, and pulse solutions are found using optimization techniques. As an application, we demonstrate the quantum full adder using a depth-4 circuit requiring the use of parallel 2-qubit operations.

BUILDING AND PROGRAMMING
A UNIVERSAL ION TRAP QUANTUM COMPUTER

by

Caroline Margaret Figgatt

Dissertation submitted to the Faculty of the Graduate School of the
University of Maryland, College Park in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
2018

Advisory Committee:
Professor Chris Monroe, Chair/Advisor
Professor Gretchen Campbell
Professor Andrew Childs
Professor Trey Porto
Professor James Williams

© Copyright by
Caroline Figgatt
2018

Dedication

to

Alan Purring, Admiral Grace Murray Hoppurr, Mr. Grayson,
and most of all, Fermi

Acknowledgments

I am incredibly grateful to have received so much support from so many wonderful folks over my years in grad school. I can hardly do them all justice, and hope my humble thanks is enough.

First and foremost, my advisor, Chris Monroe, provided me a wonderful opportunity to work in his lab at UMD. I came to grad school primarily interested in making quantum computers a reality, and my work with Chris has been doing precisely that. (I still sometimes come in to the lab, realize that this is in fact my life, and can hardly believe I'm lucky enough to be doing exactly what I want to be doing.) Chris has provided me with useful guidance throughout grad school and pushed me to do my best work, and I am grateful for his support.

I would also like to thank my thesis committee for taking the time to evaluate my candidacy: Gretchen Campbell, Jimmy Williams, Andrew Childs, and Trey Porto. Gretchen and Jimmy in particular have provided mentorship and support during my time at Maryland.

I have been fortunate to work some incredibly kind, patient, and brilliant colleagues throughout my time at Maryland. Norbert Linke, my postdoc for the last several years, has provided me with invaluable mentorship. He has been an incredibly patient teacher through endless bouts of questions, encouraged me to take on new challenges, stayed cheerful and pushed me to persevere when morale was low, and always made time to help me. I couldn't ask for better and I am deeply grateful. Kevin Landsman has been a fellow grad student in the lab with me for the

past few years, and he has been an exemplary colleague. He has made outstanding contributions to the lab, and has been a conscientious team player. I am grateful for his competence and enthusiasm during the good times, and his level-headedness and forbearance during the bad. Daiwei Zhu joined our project as a grad student a few months ago, but is already making excellent progress and has been a pleasure to work with. My immediate predecessor, Shantanu Debnath, taught me everything I know about optics, and always had a few moments to cheerfully answer a younger grad student's questions. Ken Wright has been a friend as well as a great colleague. By bringing the Igor control program with him from Georgia Tech, he saved me from having to work with Labview throughout my graduate career, an act of kindness I deeply appreciate. Andrew Manning and Taeyoung Choi helped introduce me to the Gates project when I joined as a first year grad student, and I appreciate the time they took to help me learn my way around. I've also appreciated advice and insight from a number of older students and postdocs during my time in the Monroe lab, and am grateful for my time working with all the members of the group I've intersected over the years, including Kristi Beck, Jiehang Zhang, Marty Lichtman, Marko Cetina, Guido Pagano, Michael Goldman, David Wong-Campos, Clay Crocker, Harvey Kaplan, Ksenia Sosnova, Antonis Kyprianidis, Allison Carter, Patrick Becker, Wen Lin Tan, Kate Collins, Laird Egan, Sophia Scarano, Eric Birkelbaw, Micah Hernandez, Nate Dudley, Steven Moses, Kai Hudek, Paul Hess, Hanna Ruth, Jason Amini, Jonathan Mizrahi, Volkan Inlek, Kale Johnson, Jake Smith, Aaron Lee, Geoffrey Ji, David Hucul, Brian Neyenhuis, Phil Richerme, Grahame Vittorini, Crystal Senko, Chenglin Cao, Susan Clark, and Dave Hayes. I am

also grateful for the mentorship and advice I've received from Shelby Kimmel, Kristi Beck, Emily Edwards, Qudsia Quraishi, Steve Olmschenk, Kathy-Anne Soderberg, Dave Moehring, John Berberian, and so many other folks.

Working on a world-class experiment with flexible capabilities has afforded me incredible opportunities to collaborate with a large number of wonderful and brilliant theorists. Aaron Ostrander has been a fantastic collaborator on the parallel gates experiment as gate optimizer extraordinaire, and it has been a pleasure to work with him. Dmitri Maslov has been an important collaborator on several experiments, most particularly the Grover experiment, and I have greatly appreciated his inexhaustible knowledge of quantum computing applications and his help in getting our experiment more exposure in computer science circles. Shengtao Wang was a great collaborator on the phonon hopping project. He was also kind enough to share with me a paper draft containing a detailed derivation of the XX entangling gate fidelity that proved incredibly helpful while I was deriving parallel XX gate outcomes, and for that I will forever be grateful. I enjoyed working with James Leung for the frequency modulated gates project, and appreciated his taking time to answer some questions about theory problems while I was working on the parallel gates project. I thank Zhexuan Gong for taking time on several occasions to answer questions about XX gates and pulse shaping, and enjoyed working with him for the 2014 pulse shaping paper. The quantum scrambling project involved some fascinating physics, and I thank Norman Yao for his energy and incisiveness, Beni Yoshida for his thoughtful insights and endless patience with explanations, and Tommy Schuster for his hard work on simulations; working with them has been a

wonderful learning experience. Curtis Volin provided very helpful insights when I was designing the individual control optical system. I have appreciated kind advice and support from Jungsang Kim and Ken Brown, and appreciated Ken's keen insights during collaborations. I have also been grateful to work with a number of other talented theorists, including Sonika Johri, Alireza Seif, Neal Solmeyer, Radhakrishnan Balu, Luming Duan, Martin Roetteler, Mauricio Gutierrez, Yunseong Nam, Marcello Benedetti, and Alejandro Perdomo-Ortiz.

My undergraduate career at MIT provided an important foundation for my success in grad school, and to that end I'd like to thank those who made that possible. Martin Zwierlein, my undergrad research advisor, has a boundless and infectious enthusiasm for all things physics that makes him a wonderful teacher and a patient mentor. I'm incredibly grateful to him for his mentorship in the lab and his support of me through some difficult times, as well as to his group members for teaching me how to be a useful person in a lab. Nergis Mavalvala, Deepto Chakrabarty, and Sean Robinson also provided incredible support and understanding through various hardships, and I am grateful to them for the role all of them played in my eventual success in grad school.

I am also incredibly grateful for the support of my friends and family throughout this process. My parents, Mr. Grayson, my brother Thomas and sister-in-law Liz, and the rest of my family have been very supportive. Gina Quan, Megan Marshall, and Lora Price have been incredibly supportive friends and gym buddies during my years here, and I couldn't ask for a kinder, more sarcastic group with which to share silly lifting videos and commiserate about the day to day frustra-

tions of being a physics grad student. Zach Skigen, Alan Purring, and Admiral Grace Murray Hoppurr have provided wonderful and fluffy companionship through long days of writing. Alan in particular has been most helpful with data analysis, and Grace has a knack for oversight. My roommates over the past few years have been incredibly supportive and understanding: Richard Friend, Escher, Kearci Carson, Ella, Tiffany Calvert, and in loving memory of Nola. Dawit Girma has been incredibly supportive as my powerlifting coach, and has been endlessly patient and understanding with providing coaching around an often inconsistent schedule. To all my other friends, of whom there are far too many to list here: thank you for your hugs and understanding while I've gone through this process. And of course, my own little Fermi has been a constant companion through this entire process, and has been there for me during good times and bad.

I have been privileged to serve as first treasurer and then president of Women in Physics at UMD during my time here, and I'm incredibly grateful for the support my fellow women in physics have given me and each other over these years. I particularly appreciate my fellow officers' support in stepping in while I focused on my thesis this semester. I'm also grateful to the department for being supportive of WIP. I'm proud of the work the group has done during my leadership tenure and I look forward to seeing where the upcoming leadership takes it.

Table of Contents

Dedication	ii
Acknowledgments	iii
List of Tables	xi
List of Figures	xii
List of Abbreviations	xiv
1 Introduction	1
2 Quantum Computer Hardware	7
2.1 The $^{171}\text{Yb}^+$ Qubit	7
2.1.1 Coherence Times	8
2.1.2 Cooling	9
2.1.3 Qubit Initialization	11
2.1.4 Qubit Readout	12
2.2 The Ion Trap	16
2.3 Coherent Individual Addressing System	21
2.3.1 Raman Beam Geometry	22
2.3.2 32-Channel AOM Characteristics	25
2.3.3 32-Channel AOM Control	26
2.3.4 Optics for Individual Addressing	30
2.4 Individual Addressing System Upgrades	38
2.5 Expansion to 7-Ion System	40
3 Quantum Control	43
3.1 Single-Qubit Rotations R	43
3.1.1 Rabi Flopping on 5 or 7 ions	45
3.1.2 Z Rotations	47
3.2 Two-Qubit Entangling XX Gates	48

3.2.1	Pulse-Shaping Scheme	49
3.3	Universality of the Gate Set $\{R, XX\}$	57
4	Control System	59
4.1	System Control Overview	61
4.2	The Igor Control Program	65
4.3	Compiler	71
4.4	Outlook	75
5	Gate Library	79
5.1	Single-Qubit Gates	79
5.2	Multi-Qubit Circuit Construction	80
5.2.1	Optimization Strategy Adjustments with Experimental Up-grades	82
5.3	Two-Qubit Composite Gates	83
5.4	Toffoli Gates	84
5.4.1	Toffoli-3 Characterization	88
5.4.2	Toffoli-4 Gate	92
5.5	The <i>SWAP</i> Gate	93
5.6	Controlled- <i>SWAP</i> Gate	93
5.7	Using <i>Z</i> -Rotations to Obviate χ Sign in <i>XX</i> Gates	96
5.8	The Square Root of CNOT	97
5.9	Quantum Scrambling Library	99
5.9.1	Deterministic Teleportation Protocol	101
5.9.2	Bell States and Bell Measurements	102
5.9.3	Scrambling Unitary U_S	104
5.9.4	Scrambling Unitary U_{CZ}	106
5.9.5	Classical Scrambler	108
5.9.6	Circuit Optimization	110
6	Complete 3-Qubit Grover Search	112
6.1	The Grover Search Algorithm	112
6.2	Oracles	116
6.3	Circuit Implementation	118
6.4	Data	124
6.5	Additional Iterations	127
6.6	Outlook	129
7	Parallel 2-Qubit Operations	130
7.1	Theory	131
7.1.1	Fidelity of Parallel <i>XX</i> Operations	137
7.2	Implementation	148
7.3	Experimental Results	150
7.3.1	Calculating Experimental Fidelities of 2-Qubit Entangling Gates	162

7.3.2	Fidelity of Parallel 2-Qubit Entangling Gates with Different Degrees of Entanglement	164
7.3.3	Independence of Parallel Gate Calibration	166
7.4	Simultaneous <i>CNOT</i> Gates	168
7.5	The Quantum Full Adder	169
7.6	Toward a Single-Operation GHZ State	173
7.7	Outlook	178
8	Outlook	181
8.1	Experimental Error Sources	181
8.2	Scalability of Ion Traps	182
	Bibliography	185

List of Tables

2.1	Final focal lengths for lenses chosen for the focusing lens system. . . .	35
6.1	Single-Solution Oracles	120
6.2	Two-Solution Oracles	124
7.1	Comparison of optical power for parallel and single XX gates. . . .	160

List of Figures

2.1	Energy levels of interest for $^{171}\text{Yb}^+$.	8
2.2	Qubit initialization.	11
2.3	Qubit readout.	13
2.4	Blade trap pictures.	17
2.5	Vacuum chamber and optical access.	19
2.6	Raman transitions on the experiment.	23
2.7	Counterpropagating beam geometry.	24
2.8	Multi-channel AOM.	27
2.9	Efficiency of the 32-channel AOM.	28
2.10	RF control of 32-channel AOM.	29
2.11	DOE output.	31
2.12	Individual addressing optics solution.	33
2.13	Individual addressing optical setup on table.	37
2.14	Focused beams for individual addressing.	37
3.1	Rotations on the Bloch sphere.	44
3.2	Rabi flops on 5 ions.	45
3.3	Rabi flops on 7 ions in a 9 ion chain.	46
3.4	Pulse-shaping scheme for XX gate.	56
4.1	The computational stack.	61
4.2	Experimental control.	62
4.3	Igor control system.	66
4.4	Example experimental sequence.	69
5.1	Single-qubit composite gates.	80
5.2	Two-qubit composite gates.	83
5.3	Controlled-controlled unitary.	85
5.4	Three-qubit composite gates.	86
5.5	Toffoli-3 gate data.	87
5.6	Toffoli-3 characterization.	89
5.7	Toffoli-4 gate circuit.	90

5.8	Toffoli-4 gate data.	91
5.9	Controlled- <i>SWAP</i> gate implementation	94
5.10	<i>CSWAP</i> gate data.	95
5.11	<i>XX</i> gate χ -obviation circuit.	97
5.12	Component circuits for <i>CNOT</i> and its square root gates	98
5.13	General circuit to probe scrambling behavior.	100
5.14	Grover protocol for quantum scrambling measurement.	102
5.15	Bell measurement circuit.	104
5.16	Circuits for the scrambling unitary U_S	106
5.17	Circuits for the scrambling unitary U_{CZ}	107
5.18	Circuits for the classically scrambling unitary U_{CS}	109
5.19	Unitary equivalence on EPR pair components.	110
5.20	Optimization of <i>XX</i> gates in a scrambling experiment.	110
5.21	Optimization of <i>XX</i> gates in a scrambling experiment.	111
6.1	The Grover search algorithm.	113
6.2	General circuits for a Grover search algorithm.	117
6.3	Grover search algorithm circuits.	119
6.4	Single-solution algorithm.	125
6.5	Two-solution algorithm.	126
7.1	Pulse shape for (1,4), (2,5).	151
7.2	Pulse shape for (1,2), (3,4).	152
7.3	Pulse shape for (1,5), (2,4).	153
7.4	Pulse shape for (1,4), (2,3).	154
7.5	Pulse shape for (1,3), (2,5).	155
7.6	Pulse shape for (1,2), (4,5).	156
7.7	Parity curve for parallel 2-qubit gates on (1,4), (2,5).	157
7.8	Parity curve for parallel 2-qubit gates on (1,2), (3,4).	157
7.9	Parity curve for parallel 2-qubit gates on (1,5), (2,4).	158
7.10	Parity curve for parallel 2-qubit gates on (1,4), (2,3).	158
7.11	Parity curve for parallel 2-qubit gates on (1,3), (2,5).	159
7.12	Parity curve for parallel 2-qubit gates on (1,2), (4,5).	159
7.13	Parity curve for parallel 2-qubit gates on (1,5) $XX(\frac{\pi}{4})$, (2,4) $XX(\frac{\pi}{8})$	165
7.14	Independence of parallel gate calibration.	167
7.15	Independence of parallel gate calibration, continued.	168
7.16	Simultaneous <i>CNOT</i> gates.	169
7.17	Full adder circuits.	170
7.18	Full adder data.	171

List of Abbreviations

ASP	Algorithm Success Probability
AOM	Acousto-Optic Modulator
AWG	Arbitrary Waveform Generator
<i>CNOT</i>	Controlled-NOT gate
DAC	Digital to Analog Converter
DDS	Direct Digital Synthesizer
DOE	Diffraction-Optic Element
ELU	Elementary Logic Unit
EOM	Electro-Optic Modulator
FPGA	Field Programmable Gate Array
GUI	Graphical User Interface
<i>H</i>	Hadamard gate
NA	Numerical Aperture
OTOC	Out of Time Order Correlator
PMT	Photo-Multiplier Tube
QCQP	Quadratically Constrained Quadratic Program
<i>R</i>	Rotation gate
RF	Radio Frequency
SK1	1st order Solovay-Kitaev composite pulse
SPAM	State Preparation And Measurement
SSO	Squared Statistical Overlap
TTL	Transistor-Transistor Logic
UHV	Ultra-High Vacuum
VVA	Voltage Variable Attenuator
<i>XX</i>	Two-qubit $\sigma^x \sigma^x$ entangling gate
Yb	Ytterbium

Chapter 1: Introduction

The potential applications for quantum computers have been of great interest since Manin and Feynman first proposed them in the early 1980s [1, 2]. By taking advantage of properties unique to quantum systems like multi-state superposition and multi-qubit entanglement, a large-scale quantum computer promises advances in humanity's ability to tackle complex chemistry problems that could revolutionize the energy sector, solve optimization problems with uses across almost every industry, accelerate breakthroughs in biology and medicine, and - depending on your point of view - either wreak havoc on or revolutionize the security and data industries.

To simulate a two-level quantum mechanical system on a classical computer (like the one you're probably reading this thesis on), a system with N particles requires keeping track of 2^N amplitudes, an exponential growth in problem size that prevents simulating large quantum systems; even the world's biggest supercomputers have not been able to simulate all amplitudes in a quantum system larger than 49 particles [3]. Future advances in computing will likely push past that, but most predictions anticipate only getting to 60-70 particles. Simulating 300 quantum particles would require a classical computer the size of the known universe, which is not practical. However, a quantum computer can simulate an N -particle quantum

system with much less than exponential growth, simply because it is a quantum system as well; in principle, the problem size can be $O(N)$ particles, with perhaps some additional resources required for error correction. This power of “quantum parallelism” [4] can be used to directly simulate quantum systems of interest, such as for quantum chemistry applications, but can also be harnessed to solve other kinds of problems, such as optimization problems. One particularly interesting potential application in the field of quantum chemistry is to better understand biological nitrogen fixation [5], a process that is central to the creation of fertilizer. Our best fertilizer creation methods require a substantial proportion of the human energy budget, but bacteria can create the same chemicals far more efficiently. A quantum computer could help us discover these efficient methods for producing fertilizer, making it easier to feed humanity while reducing our energy usage, our carbon footprint, and our effects on global climate change. The exact scope of quantum computing technology’s effect on humanity remains tantalizingly unclear, but even pessimistic outlooks envision significant benefits from large quantum computers. Consequently, increasing numbers of government and industry players are investing in this technology - from U.S. government agencies like IARPA, the NSF, the Departments of Defense and Energy, and the NSA, to companies like Google, Intel, Microsoft, and startups like Rigetti and IonQ.

Of course, all of this being of any use at all hinges on an important question: how do we build a quantum computer big enough to do something interesting? In this thesis, I leave aside some important abstract issues that arise from this central question (how do we define “big”? What is “interesting” in this context? What

constitutes a quantum computer, exactly?) to focus on the pragmatic and practical aspects of the question: “building” and “doing”.

Ion trap quantum computers [6, 7] and superconducting quantum computers [8, 9] represent the two proposed quantum computing platforms that so far have made the most progress toward a practical and usable technology, and have optimistic outlooks for scaling up. Recent advances in both of these platforms portend optimism that a large-scale quantum computer will eventually be built, though challenges certainly remain. Additionally, it is not yet clear which platform will prove itself to be the best candidate for a burgeoning technology, or whether the most successful quantum computers will be some other hardware [10] that is currently still being explored (such as neutral atoms [11, 12], quantum dots [13, 14], nitrogen vacancy centers in diamond [15], or large photonic systems [16, 17]), a hybrid of two or more quantum systems [18], or even one that has yet to be proposed at all.

In this thesis, I will discuss an ion trap quantum computing machine we have built [19, 20] and describe some results obtained with it. To demonstrate that our machine is indeed a quantum computer, I use the DiVincenzo criteria [21] as a practical measure for a highly programmable and usable device that is similar enough to modern digital computers to be a familiar and user-friendly concept, but still takes advantage of all available resources of quantum computing. The criteria, and our adherence to them, are summarized below.

- Qubits: We use well-characterized $^{171}\text{Yb}^+$ ions as qubits (see Section 2.1) in a scalable trapped-ion quantum computing architecture [7].

- Qubit initialization: $^{171}\text{Yb}^+$ qubits can be initialized to a known state with high fidelity (see Section 2.1.3).
- Long coherence times: These qubits have typical coherence times longer than 1 s, which is many orders of magnitude greater than gate times of 20-200 μs and hence allows for long coherent gate sequences (see Section 2.1.1).
- A universal gate set: The native gate set available, single-qubit R rotations and two-qubit entangling XX gates, constitute a universal gate set (see Section 3.3).
- Qubit readout: State-dependent fluorescence of $^{171}\text{Yb}^+$ qubits allows for high-fidelity qubit measurement (see Section 2.1.4).

Two additional optional DiVincenzo criteria exist that address a system’s capabilities for quantum communication; these are not addressed here, as the system is not designed to communicate with other quantum systems and hence does not meet these criteria at present. Other efforts exist to implement robust quantum communication and networking [22] and provide long-distance quantum communication [23] using ion traps and other hardware.

To be most useful, a quantum computer should be easily programmable and flexible to be able to tackle many kinds of problems, as a classical computer can. To this end, we have built an ion trap quantum computer with controls organized into separate levels of abstraction, called a computing stack. Each level of the stack represents a different level of control in the system, and is designed to be a black box to the others; this architecture allows users to implement quantum algorithms

without needing to know much, if anything, about what qubits the system uses or the physics and engineering details behind the hardware. This framework is very similar to classical computers; I don't need to know anything about NAND gates or silicon microprocessors to read the news on the Internet or run data analysis calculations for this thesis. Most previous quantum computing experiments have been built to perform a single algorithm, with the hardware designed and tailored for that purpose, and without flexibility to perform many other experiments; the stack framework of this machine represents a step forward from a systems perspective, as well as the previously-undemonstrated results we have generated on it.

The first half of this thesis, Chapters 2-5, will describe the functioning of the ion trap quantum computer by going through the stack, from the low-level hardware up to the high-level user interface. The second half, Chapters 6-7, will describe several new results generated from the system over the course of this thesis work, and can additionally be found in [24, 25]. Additional work performed on this machine over the course of this thesis work but not discussed here in detail can be found in [19, 26-32].

In Chapter 2, I present the hardware level of the stack: trapped $^{171}\text{Yb}^+$ ions as qubits, with various lasers of initialization, readout, and coherent control.

In Chapter 3, I discuss the quantum control level of the stack, consisting of pulse shaping schemes that effect high-quality quantum gates.

In Chapter 4, I explain the compiler level of the stack, which consists of a computer control program that compiles quantum gate commands from the user into the low-level operations necessary to execute the gate sequence on the experiment.

In Chapter 5, I enumerate the library of quantum gates we have assembled for use in the compiler to execute various algorithms.

In Chapter 6, I report results for a complete 3-qubit Grover search algorithm.

In Chapter 7, I introduce a promising new tool for quantum control: parallel 2-qubit operations on pairs of ions in the same ion chain.

In Chapter 8, I conclude with a positive outlook on ion traps as quantum computers, and discuss future challenges and suggest areas to improve.

Chapter 2: Quantum Computer Hardware

In this chapter, I will present an overview of the hardware of the ion trap quantum computer used to perform the work in this thesis. The overview will be brief, as much of this material has already been presented in [20], which concerns the same system. I will also include some upgrades that have since been made to the system, including to the individual addressing system allowing for better individual ion controls crucial to work presented in Chapter 7, and expanding from 5 to 7 ions for use in the work presented in [31] and future work.

2.1 The $^{171}\text{Yb}^+$ Qubit

In this system, we have chosen as our qubit the first-order magnetic-field-insensitive pair of clock states in the hyperfine-split $^2S_{1/2}$ ground level manifold of $^{171}\text{Yb}^+$, with $|0\rangle \equiv |F = 0; m_F = 0\rangle$ and $|1\rangle \equiv |F = 1; m_F = 0\rangle$ (see Figure 2.1) [33]. The two qubit states have a 12.642821 GHz frequency difference, which can be addressed with microwaves or Raman transitions via the 2P levels.

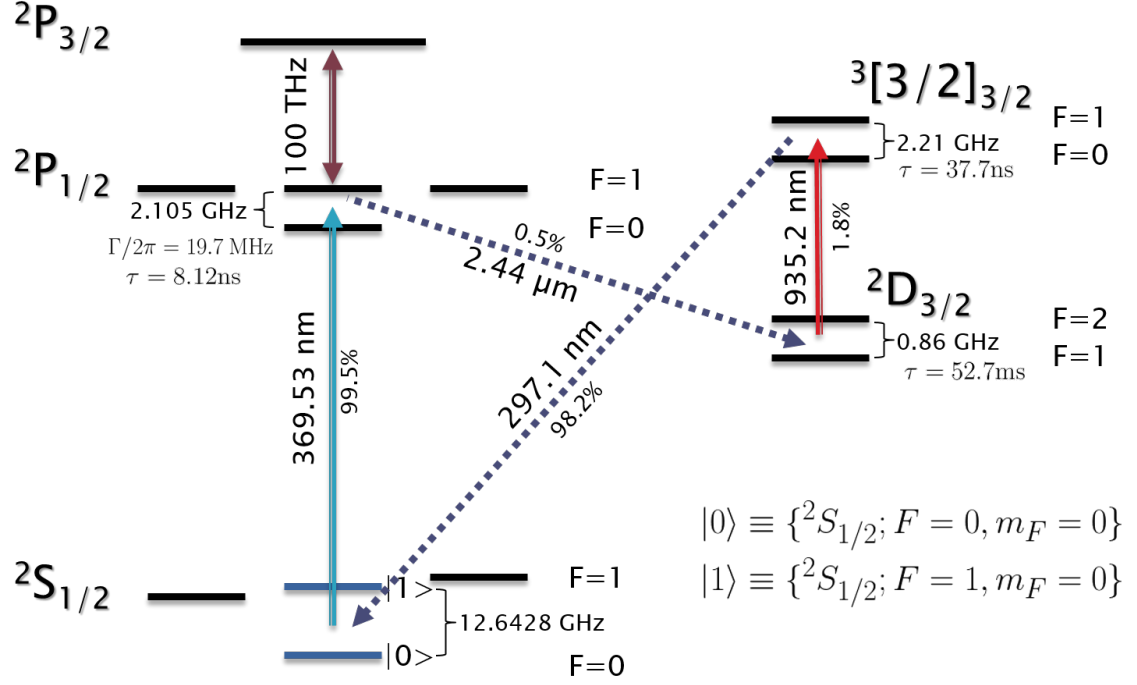


Figure 2.1: Energy levels of interest for the $^{171}\text{Yb}^+$ qubit. The qubit is defined as $|0\rangle \equiv |F=0; m_F=0\rangle$ and $|1\rangle \equiv |F=1; m_F=0\rangle$.

2.1.1 Coherence Times

The $|F=0; m_F=0\rangle$ and $|F=1; m_F=0\rangle$ states in $^{171}\text{Yb}^+$ ions are so-called “clock” qubits because they are insensitive to magnetic fields to first order. On our experiment, our measured coherence time is $1.5(5) \text{ s}$, which is more than 3 orders of magnitude greater than typical single-qubit gate times of $20 \mu\text{s}$ or two-qubit gate times of $200 \mu\text{s}$; for short algorithms with at most a few tens of two-qubit gates, this coherence time is therefore sufficient. On this experiment, performance is primarily limited not by qubit coherence, but by pointing noise on the Raman control beams driving coherent transitions; this is further discussed in Chapter 8.

2.1.2 Cooling

The qubits are cooled to near their motional ground state in a two-step process. First, the ion is cooled using Doppler cooling [34,35]. Light beams detuned from the $\{^2S_{1/2}, F = 1\}$ to $\{^2P_{1/2}, F = 0\}$ and $\{^2S_{1/2}, F = 0\}$ to $\{^2P_{1/2}, F = 1\}$ transitions is applied to the ions, with both π and σ polarizations added to address all of the Zeeman states in the ground state manifold. The two transitions are applied by supplying 369 nm light resonant with the $\{^2S_{1/2}, F = 1\}$ to $\{^2P_{1/2}, F = 0\}$ transition, and adding a 14.7 GHz sideband to access the $\{^2S_{1/2}, F = 0\}$ to $\{^2P_{1/2}, F = 1\}$ transition using the second-order sideband of an EOM at 7 GHz. The 369 nm light is generated by doubling the output of a 739 nm laser that is locked to an iodine vapor cell [36]. The output of the doubler is 300 MHz detuned from the transition, so frequency shifts and TTL control are provided by one IntraAction AOM at 290 MHz to give the optimal Doppler cooling frequency for a $\Gamma/2\pi = 19.7$ MHz linewidth.

A light beam red-detuned from resonance by 300 MHz (the “protection beam”) acts on the ions during trapping only, and two more beams red-detuned by 10 MHz with an AOM cool the ions further during setup before experimental procedures. Red-detuned photons are absorbed by hot ions moving toward the beam source, and then scattered in a random direction, reducing their net momentum (and consequently, their kinetic energy) in the direction of the cooling beams. The three beams are therefore at angles to each other to ensure cooling along all axes; the main 10 MHz beam travels at a very slight angle from the ion chain axis in the horizontal plane, the 300 MHz protection beam is applied in the horizontal plane at

an angle near perpendicular to the chain axis, and the low-power 10 MHz “oblique” beam is applied with a vertical component at an oblique angle to the other two (see Figure 2.5). On our experiment, we define the z axis as the horizontal direction along the trap axis, the x axis as the horizontal axis perpendicular to the trap axis, and the y axis as the vertical direction, perpendicular to the trap axis and the optical table. Consequently, the beams have components along the x , y , and z directions, and while we don’t reach the Doppler limit, the ions are cooled enough in all directions ($\bar{n} = 10$ to 15) that resolved sideband cooling can finish cooling the ions to near their motional ground states.

Second, resolved sideband cooling [37] is necessary to further cool the ions to the Lamb-Dicke regime, which is necessary to execute high-fidelity gates. The Lamb-Dicke parameter η can be described as a measure of the coupling strength between the ion’s internal or spin state (so called because we are only using the two qubit states) and its motional state. In the Lamb-Dicke regime, this parameter is small enough that transitions changing the motional quantum number by more than 1 phonon are strongly suppressed. Specifically,

$$\eta^2(2n + 1) \ll 1 \tag{2.1}$$

where n is the motional quantum number of the ion. Resolved sideband cooling is a process that couples to the motional modes of the ion to progressively remove phonons while cycling between the $|0\rangle$ and $|1\rangle$ spin states [38]. This cools the ions below the Doppler cooling limit, and on our experiment ends with a typical $\bar{n} = 0.1$

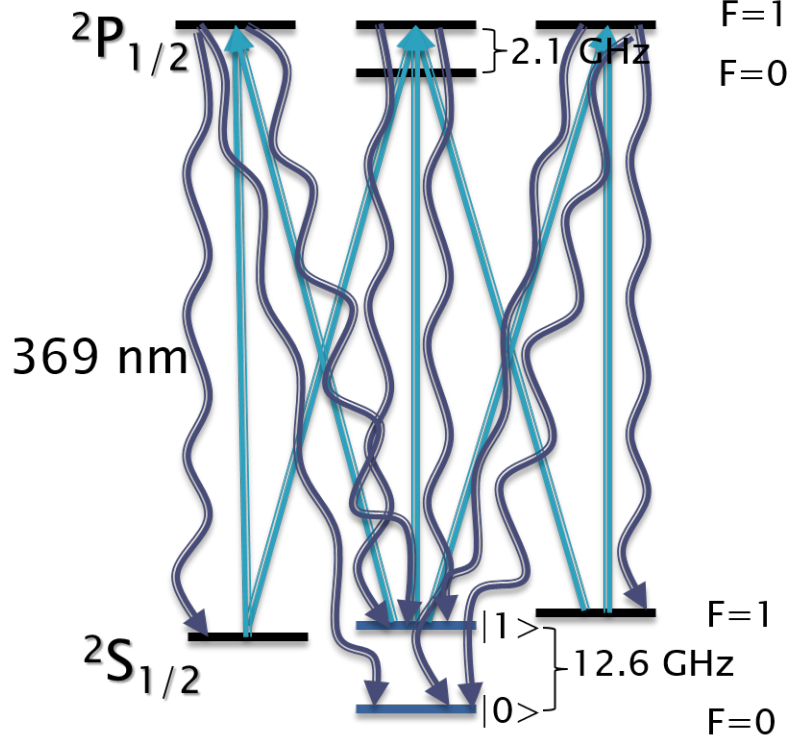


Figure 2.2: Qubits are initialized to the $|0\rangle$ state via optical pumping. The straight blue arrows indicate photon absorption, and the squiggly purple arrows indicate photon emission.

quanta.

2.1.3 Qubit Initialization

The qubits are initialized to their $|0\rangle$ state via optical pumping [33]. The ions are illuminated with light resonant with the transition between the $\{^2S_{1/2}, F = 1\}$ and $\{^2P_{1/2}, F = 1\}$ manifolds, as shown in Figure 2.2. An AOM provides a frequency shift of 300 MHz to make the 369 nm beam on-resonant with the $\{^2S_{1/2}, F = 1\}$ and $\{^2P_{1/2}, F = 0\}$ transition, and an EOM adds 2.1 GHz sidebands to then reach the $\{^2P_{1/2}, F = 1\}$ manifold to clear out the $\{^2S_{1/2}, F = 1, m_F = 0\}$ state. The beam contains both π and σ polarization components to ensure successful pumping for ions

in any of the Zeeman states in the $\{^2S_{1/2}, F = 1\}$ manifold. (π alone is sufficient, but the beam path is shared with the detection beam discussed in Section 2.1.4, which requires both components.) Once pumped to the $\{^2P_{1/2}, F = 1\}$ manifold, the ions then decay either back to the $\{^2S_{1/2}, F = 1\}$ manifold, where they will be pumped again, or to the $\{^2S_{1/2}, F = 0\}$ target state. Once there, ions in the $\{^2S_{1/2}, F = 0\}$ state stay there, as the applied beam is now 14.7 GHz off-resonant from a transition back to the $\{^2P_{1/2}, F = 1\}$ manifold, so running the pump beam for a short amount of time ($5 \mu\text{s}$) is sufficient to initialize all qubits to the $|0\rangle$ state with high fidelity.

2.1.4 Qubit Readout

At the end of a quantum procedure, the qubits are measured and their states read out using state-dependent fluorescence, as shown in Figure 2.3(a). Light resonant with the transition from $\{^2S_{1/2}, F = 1\}$ to $\{^2P_{1/2}, F = 0\}$ is applied to the ions, using a 300 MHz AOM to shift a beam of 369 nm light to resonance. Qubits in the $|1\rangle$ state are then excited to the $\{^2P_{1/2}, F = 0\}$ state, which has a lifetime of 8.12 ns, and then decays back to the $\{^2S_{1/2}, F = 1\}$ manifold, emitting a photon in the process and hence termed “bright.” Polarization elements in π and σ are added to ensure continued cycling from all of the Zeeman levels in the $\{^2S_{1/2}, F = 1\}$ manifold, allowing many photons to be scattered over the course of the detection period. If the qubit is instead in the $|0\rangle$ state, the applied light is 2.1 GHz away from the nearest dipole-allowed transition to the $\{^2P_{1/2}, F = 1\}$ manifold, and so

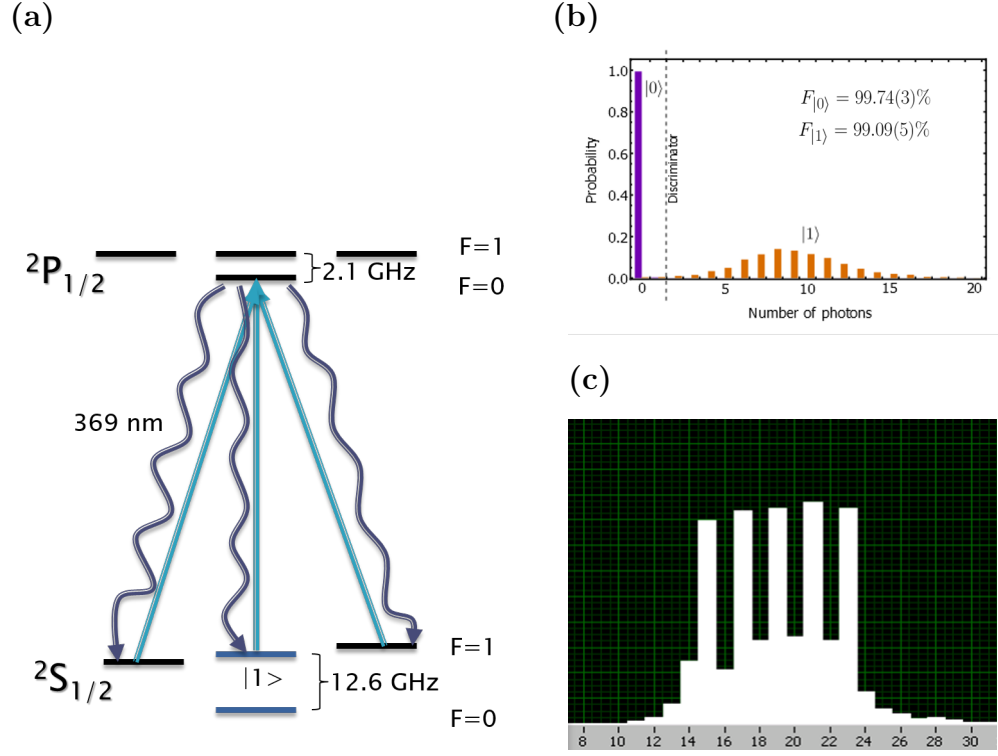


Figure 2.3: (a) Qubits are measured using state-dependent fluorescence. Ions in the $|1\rangle$ state fluoresce in response to an applied detection beam; ions in the $|0\rangle$ state do not. The straight blue arrows indicate photon absorption, and the squiggly purple arrows indicate photon emission. (b) A typical detection histogram showing photon counts for bright (orange) and dark (purple) ions. The measurement fidelity for a qubit in $|0\rangle$ is 99.74(3)% and for a qubit in $|1\rangle$ is 99.09(5)%. From [20]. (c) Individual ions are detected on separate channels of a linear array of 32 PMTs. This shows photon counts for 5 bright ions; the X-axis is the PMT channel number.

the qubit scatters no photons and remains dark. The detection process is run for $150\mu\text{s}$ to ensure sufficient photons are collected to discriminate between bright and dark qubits. The entire experiment is repeated several hundred or thousand times to build up statistics, and photons are collected with an imaging objective with a numerical aperture (NA) of 0.37. Figure 2.3(b) shows a typical histogram showing the number of photon counts for bright and dark states. The distributions are well-distinguished, and we can discriminate between $|0\rangle$ and $|1\rangle$ by determining that any qubit with 0 or 1 counts is dark, and any qubit with 2 or more counts is bright.

Measurement errors arise when a bright qubit is measured as dark or a dark qubit is measured as bright during the measurement process. The former can occur during the measurement due to off-resonant coupling between the $F = 1$ states of $^2S_{1/2}$ and $^2P_{1/2}$, as the $\{^2P_{1/2}, F = 1\}$ manifold is only 2.1 GHz away from the $\{^2P_{1/2}, F = 0\}$ state. Once pumped to the $\{^2P_{1/2}, F = 1\}$ manifold, the ion can decay to $\{^2S_{1/2}, F = 0\}$, where it ceases scattering photons. Off-resonant coupling between $|0\rangle$ and the $\{^2P_{1/2}, F = 1\}$ manifold is also possible, which would scatter a photon upon decay, but with a detuning of 14.7 GHz, this is much less likely. This results in the not-quite-Poissonian histograms in Figure 2.3(b); in particular, the off-resonant coupling of $|1\rangle$ to $\{^2P_{1/2}, F = 1\}$ is the largest effect, resulting in visible histogram points at 0 and 1 counts for an ion prepared in the bright state. Background scatter from the trap blades can also result in photon counts for dark ions, but has a small effect. Consequently, the measurement fidelity for a single ion in $|0\rangle$ is slightly higher than for $|1\rangle$: 99.74(3)% and 99.09(5)%, respectively.

To distinguish between the fluorescence of different ions, the ions are imaged

onto separate channels of a 32-channel linear PMT array¹. Figure 2.3(c) shows typical data for $N = 5$ bright ions on the PMT array, where the X-axis is the channel number. The ions are imaged onto every other channel to reduce electronic crosstalk errors - neighboring channels have about 4% crosstalk error, and next-nearest neighbors about 1%. The crosstalk is also asymmetric, as channels to the right see more spillover from channels to their left than the other way around. Dark counts are 1-2 per channel per second and therefore negligible. When data is collected, each channel is analyzed separately, using the discriminator mentioned above to determine which ions are in $|0\rangle$ or $|1\rangle$ after each experiment. After the experiment is repeated many times, the data can then be assembled into a vector of output populations, which gives the population in each of the N^2 N -qubit states; for 5 ions, the state vector gives the populations in each of $\{00000, 00001, 00010, 00011, \dots\}$.

Bright-to-dark pumping and crosstalk are the two main sources of error for our multi-qubit readout process. We account for this and state preparation errors by measuring our SPAM errors and then renormalizing our data appropriately. To measure our SPAM errors, we trap a single ion, then use the DC voltage controls to move that ion to each of the locations in the trap where an ion would be for an N -ion chain. Two measurements are taken at each location: one where the ion is initialized to the dark state, and another where the ion is initialized and then coherently rotated to the bright state using a microwave pulse resonant with the 12.6 GHz qubit frequency. The measurements are repeated 80,000 times to build robust statistics. The measured data is then assembled into a $N^2 \times N^2$ matrix of

¹Hamamatsu H7260-200

each of the possible states. Once data is taken for an algorithm of interest, this SPAM matrix can then be inverted and multiplied with the corresponding $N^2 \times N^2$ state matrix for the data to remove SPAM errors [20, 39]. Additional methods of measuring SPAM errors on our experiment are presented in [40].

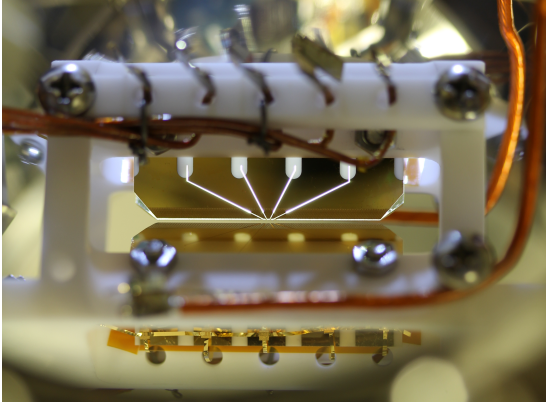
2.2 The Ion Trap

The ions are held in an ion trap in a chamber held at room temperature [41]. Our physical qubits are well-isolated from the environment in a stainless steel chamber that has been evacuated to ultra-high vacuum (UHV) with pressure about 10^{-11} Torr, meaning a given ion will only experience a collision with a background gas particle once every hour or so on average. This is therefore negligible on the order of the coherence time, and primarily limits the lifetime of the ion in the trap. Consequently, we can hold chains of 5 or 7 ions for about an hour at a time.

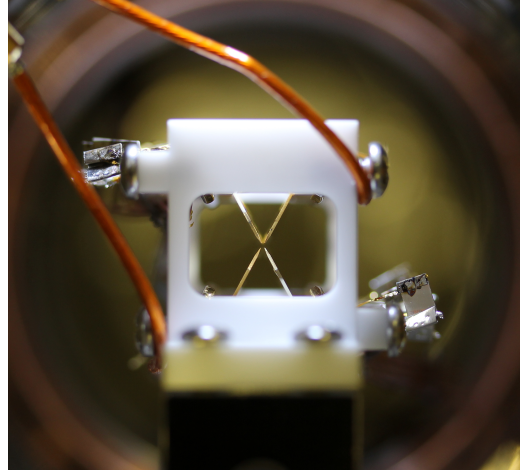
The trap used here is a blade trap, a variant of a linear Paul trap [42]. Four gold-coated electrode substrates shaped like razor blades provide the confining potential (see Figure 2.4 for several views of the blade trap). Two blades, diagonally across from each other, provide an oscillating pseudopotential that confines the ions along the radial directions of the trap. The RF signal supplying the pseudopotential is generated by a DDS and amplified to ~ 300 V using a stabilized helical resonator [43, 44] at a driving frequency of 23.8167 MHz.

The other two blades are divided into 5 electrodes each that then have static voltages applied. These DC electrodes provide an axial confinement of 296 kHz

(a)



(b)



(c)

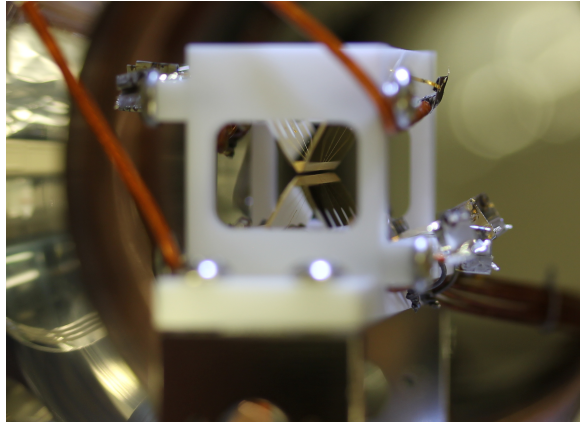


Figure 2.4: Several views of the ion trap blades. (a) A view along the blade, which can be seen here to be divided into 5 electrodes for enhanced ion position control on the DC blades. (b) A view of the 4 blades, looking directly down the trap axis. (c) Another view of the 4 blades, shown at an angle from the trap axis. Image credit S. Debnath.

for 5 ions and also allow for finer control of the spacing between the ions. Axial confinement requires, at minimum, 3 electrodes on each blade, where the center electrode on each blade is set to a negative voltage (lower potential for the positively-charged ion), and the outer electrodes are set to a positive voltage (higher potential for the ion). This creates a quadratic potential well confining the ion. Multiple ions in such a trap will not be equally spaced; the inner ions will be pushed closer together than the outer ions, and the spacing disparities increase with increasing ion number [45]. Since our individual addressing beams are equally spaced, equal spacing of ions would be ideal. Very nearly equal spacing can be achieved by adding at least 2 more electrodes to each blade, which allow for the addition of a quartic term to the trap potential; appropriately selected trap voltages will create a potential with a “flat” area in the center that will hold ions at approximately equal distances from each other. In practice, however, simulations of our blade trap showed that such a potential would require kilovolts on the outer electrodes. Equal spacing for large numbers of ions will likely require a microfabricated surface electrode trap with many DC electrodes to control the positioning of many ions. However, for 5 ions in our blade trap, the spacing differential is small enough that the individual beams are still fairly well aligned on the ions; the Rabi frequencies seen by the ions are comparable (see Section 3.1.1.)

Below the trap, two small stainless steel tubes, or ovens, contain small amounts of solid ytterbium metal that has been isotopically purified to be 95% ^{171}Yb (natural abundance 14%). Niobium wires wrapped around the ovens heat up when current is applied, allowing a small amount of heated ytterbium to sublime into a vapor

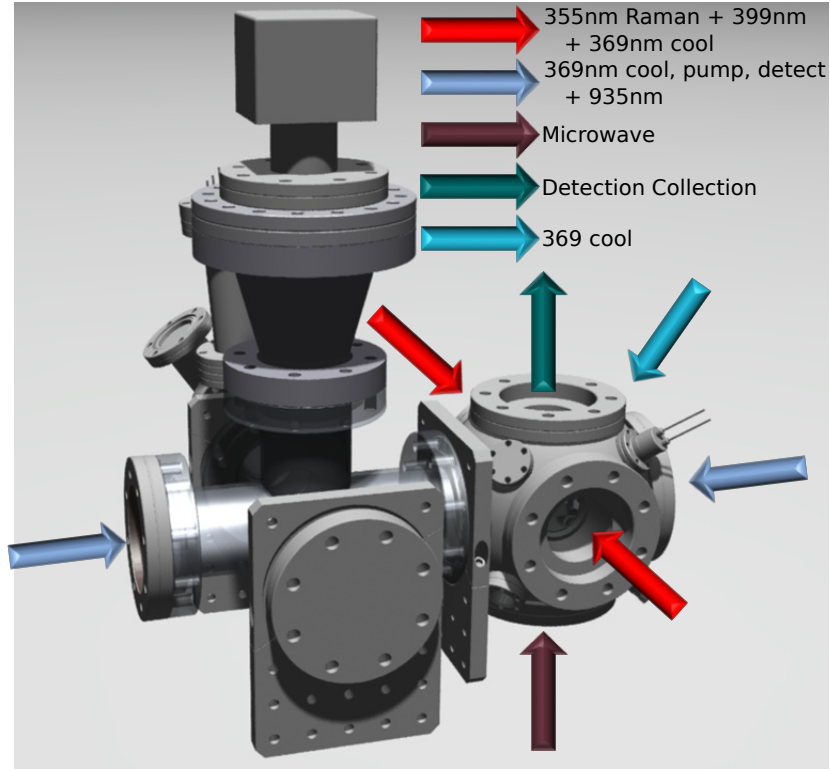


Figure 2.5: A visualization of the vacuum chamber containing the ion trap, and the optical access for the various necessary control components. The Raman control beams (red) and one of the cooling beams are applied radially, orthogonal to the ion chain, while another cooling beam, the detection and pump beams, and the loading beam (blue) are applied along the chain axis. A third cooling beam is applied through a small window at angles to both of the other cooling beams (turquoise.) Microwaves are applied through a horn from below (purple.) Finally, detection light is collected from above the trap (green.) Chamber CAD modified from S. Debnath [19].

of neutral ytterbium atoms, some of which will spread into the trapping area. A laser at 399nm, tuned to the $^1S_0 \rightarrow ^1P_1$ atomic transition in neutral ^{171}Yb , is directed through the trap area, so some atoms will absorb a photon and be excited to the 1P_1 state. An additional photon at ≤ 369 nm is required to achieve the first photoionization energy and strip off one electron to create $^{171}\text{Yb}^+$ [46]; the various 369nm laser beams already passing through the trap that will next be used for cooling purposes serve to complete this ionization process. Once ionized, the atom is now subject to the trap potential, and can be cooled to the motional ground state by the cooling beams. The ions then form a linear Coulomb crystal along the weakly-confined axial direction in the trap. On this experiment, once the ovens heat up (a 3 minute process), 5-9 ions can typically be trapped within a minute or two.

As shown in Figure 2.1, states in the $^2P_{1/2}$ manifold will decay to the metastable $^2D_{3/2}$ level with a branching ratio of 200:1. To repump the ion, a 935 nm laser beam is applied constantly to the ions. This repumper induces a transition to the $^3[3/2]_{3/2}$ level, which then decays to the ground level. The repumper beam has 3.07 GHz sidebands added by an EOM and has both π and σ polarization components to address all states in the $^2D_{3/2}$ manifold.

All of the optical controls discussed so far, as well as the Raman beams described in the next section, present a challenge in delivering all of these controls to the ions in the trap. Figure 2.5 shows a mockup of the vacuum chamber, which has 8 windows around the trap providing ample optical access for cooling beams (369nm) at 3 different angles, a pump/detection beam (369nm) transported through a shared fiber at different times during the experimental sequence, the loading beam

(399nm), the repumper beam (935nm), counterpropagating 355nm Raman beams applied from opposing sides (see Section 2.3), microwaves (12.6 GHz), and detection light collection.

2.3 Coherent Individual Addressing System

Coherent transitions between the $|0\rangle$ and $|1\rangle$ states of the qubit can be driven using direct microwave transitions at 12.642821 GHz, or via the P -levels using Raman transitions driven by two beams with a beatnote of 12.642821 GHz. Microwaves have two major drawbacks. First, they are not straightforward to focus in free space to individually address an ion in a trap [47]. Second, microwaves provide very little net momentum transfer to couple to the motional modes of the chain, which is what we use to entangle multiple ions; doing so is possible, but again not straightforward, as it requires a surface trap with waveguides [48]. Individual addressing and implementation of two-qubit gates on non-clock-state qubits can alternatively be accomplished with microwaves by generating a very large magnetic field gradient, which induces different Zeeman splittings in the qubits that can then be individually addressed by changing the microwave frequency and used to couple to motional modes [49]. However, the magnetic field gradient required is not scalable to large numbers of ions in the trap, as the gradient required for more than a handful of ions quickly becomes infeasible. On our experiment, we do provide a microwave horn that emits a 12.642821 GHz signal and globally addresses all ions, as a useful tool for troubleshooting and performing diagnostics. Coherent operations during algorithm-

mic processes are provided by individually-addressed Raman beams, as described in this section.

2.3.1 Raman Beam Geometry

Coherent qubit operations are performed by a pair of counterpropagating Raman beams on the ions in our trap (see Figure 2.6(a)). The Raman transition is performed by a mode locked 355nm pulsed laser. The two Raman beams therefore each consist of a frequency comb [50], with components separated by the laser repetition rate, about 118 MHz. To address the 12.6 GHz qubit splitting, the counterpropagating Raman beams must be lined up in frequency space so that there is a 12.6 GHz beat note between each corresponding tooth in the two frequency combs. To account for noise on the laser repetition rate, an AOM beat note lock (similar to that described in [51]) is constructed, in which the repetition rate noise is inverted and added to an AOM on one Raman beam. Consequently, one Raman beam can coherently stimulate a transition from the $|0\rangle$ or $|1\rangle$ state to a virtual state partway between the $^2P_{1/2}$ and $^2P_{3/2}$ manifolds in $^{171}\text{Yb}^+$, then the other beam induces a transition to the other qubit state, as shown in Figure 2.6(b). The commercially-available 355 nm laser wavelength is close to minimum spontaneous emission from the $^2P_{1/2}$ and $^2P_{3/2}$ manifolds, which is 1/3 of the way between the two manifolds since $^2P_{3/2}$ has more states in the manifold and therefore twice the coupling of the $^2P_{1/2}$ manifold. Due to the counterpropagating geometry and the large amount of energy carried by 355 nm photons (844 THz), stimulated Raman transitions impart

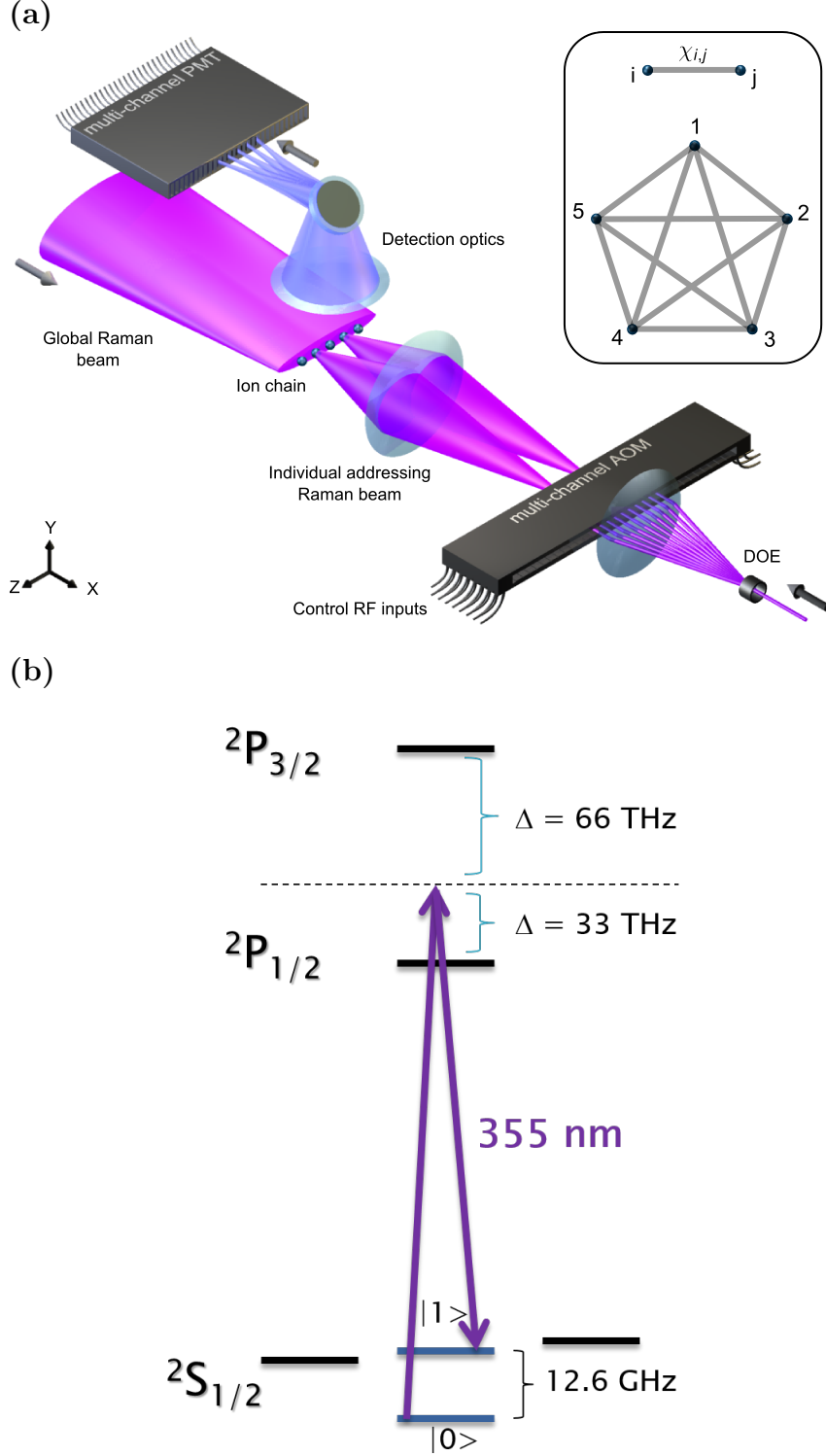


Figure 2.6: (a) An overview of the experimental setup, showing the counterpropagating Raman beams, the ion chain, and the multi-channel PMT detection apparatus. The use of spin-motion couplings to enact 2-qubit entangling gates results in all possible interaction pairs being available for entangling gates. Image from [19]. (b) Raman transitions at 355 nm in $^{171}\text{Yb}^+$.

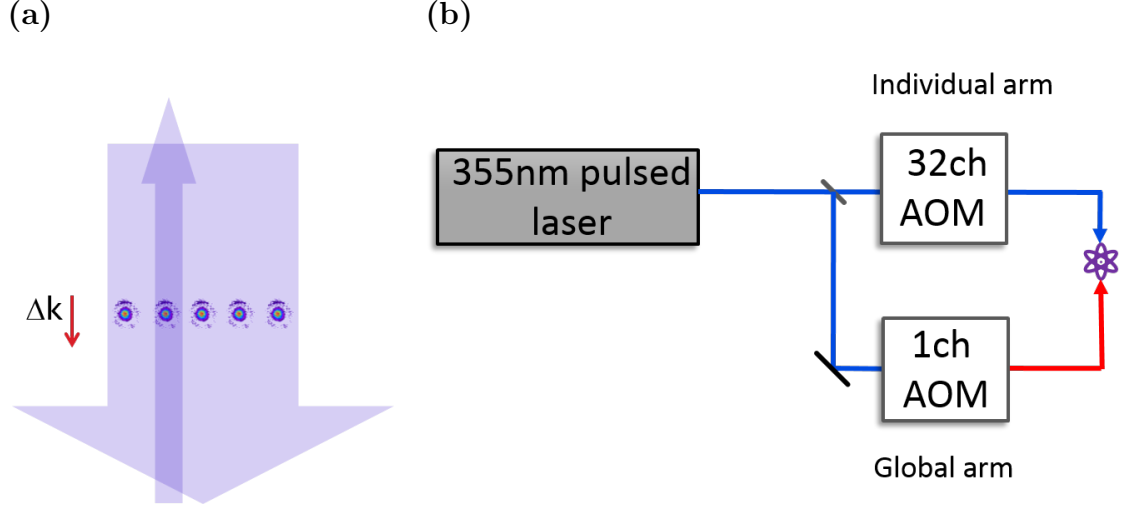


Figure 2.7: (a) Counterpropagating beams overlapping 5 example ions. The wide beam is the global beam that interacts with all ions, while the narrow beam is one individual beam aligned to interact with a single ion. (b) The two arms of the Raman beam setup.

a large momentum kick to the ions (see Figure 2.7(a)); this will allow us to couple strongly to the motional modes to implement 2-qubit gates (see Section 3.2.)

The beam path is split to provide the two arms of the counterpropagating geometry, as shown in Figure 2.7(b). Each arm is then modulated by an AOM, one of which is controlled by the beat note lock, and the other of which is controlled by the output of an AWG that modulates the frequency as needed, such as to address the red and blue sidebands necessary to implement a Mølmer-Sørensen gate [52, 53]; see Section 2.4 for discussion of AWG control of the AOMs, which was changed partway through this thesis work.

The global beam is modulated with an IntraAction AOM at 130 MHz. After the AOM, cylindrical lenses shape the beam to be wide along the trap axis and narrow in the transverse direction, so that light from the beam falls on all of the ions in the chain, while maintaining a narrower focus of $\sim 20 \mu\text{m}$ perpendicular to

the chain. The individual beam arm of the Raman setup is split horizontally into 10 beams; each individually couples to one of 10 neighboring channels of the 32-channel AOM. The beams are deflected vertically, which is orthogonal to the ion chain axis. Since the zeroth-order beams are not needed, a D-mirror is used after the AOM to redirect them to a beam dump. The first-order beams continue to the ions, where the AOM-provided modulation allows Raman transitions to occur. Since only deflected beams reach the ions, shutting off the RF signal to a given channel will shut off the beam that reaches the corresponding ion; hence, control over the RF signal to each channel provides control over which ions perform operations.

2.3.2 32-Channel AOM Characteristics

AOMs are a tool used frequently in atomic physics experiments. Laser light of frequency f and wavelength λ is passed through a transparent crystal, such as fused silica. A piezoelectric transducer, controlled by an applied RF signal of frequency F , transmits sound waves of wavelength Λ into the crystal, which varies the index of refraction in the crystal. In the Bragg regime, the incoming light is then diffracted into symmetric orders m (see Figure 2.8(a)), each of which is also modulated by its order number times the applied RF frequency,

$$f \rightarrow f + mF, \tag{2.2}$$

and displaced by an angle $m\theta$, where

$$\sin \theta = \frac{m\lambda}{2\Lambda} \quad (2.3)$$

Most of the light can be directed into the first diffraction order by adjusting the incoming light's angle of incidence in the plane of the acoustic waves.

The 32-channel AOM varies from the typical single-channel model in that it consists of a single crystal with 32 evenly spaced transducers, as shown in Figure 2.8(b). The transducers are spaced about 450 μm apart, with a channel width of 100 μm . Each transducer is controlled by its own RF input channel; all of the channels are designed for a drive frequency at 200 MHz frequency and 700 mW in power. The channels are arranged horizontally, and activated channels deflect beams vertically. The crystal and transducers are watercooled to accommodate that much RF power while suppressing temperature-based signal fluctuations. We characterized the efficiency of the AOM, and found that all channels had $> 65\%$ efficiency into the first order, and a comfortable 20 MHz range over which the efficiency was constant (for a typical channel, see Figure 2.9.) Crosstalk was also found to be minimal, with $< 1\%$ optical and RF crosstalk.

2.3.3 32-Channel AOM Control

Providing RF power to the AOM channels has some important considerations in order to work with the Raman beam setup, and so the circuit design and component selection must be performed carefully. The circuit design is shown in Figure

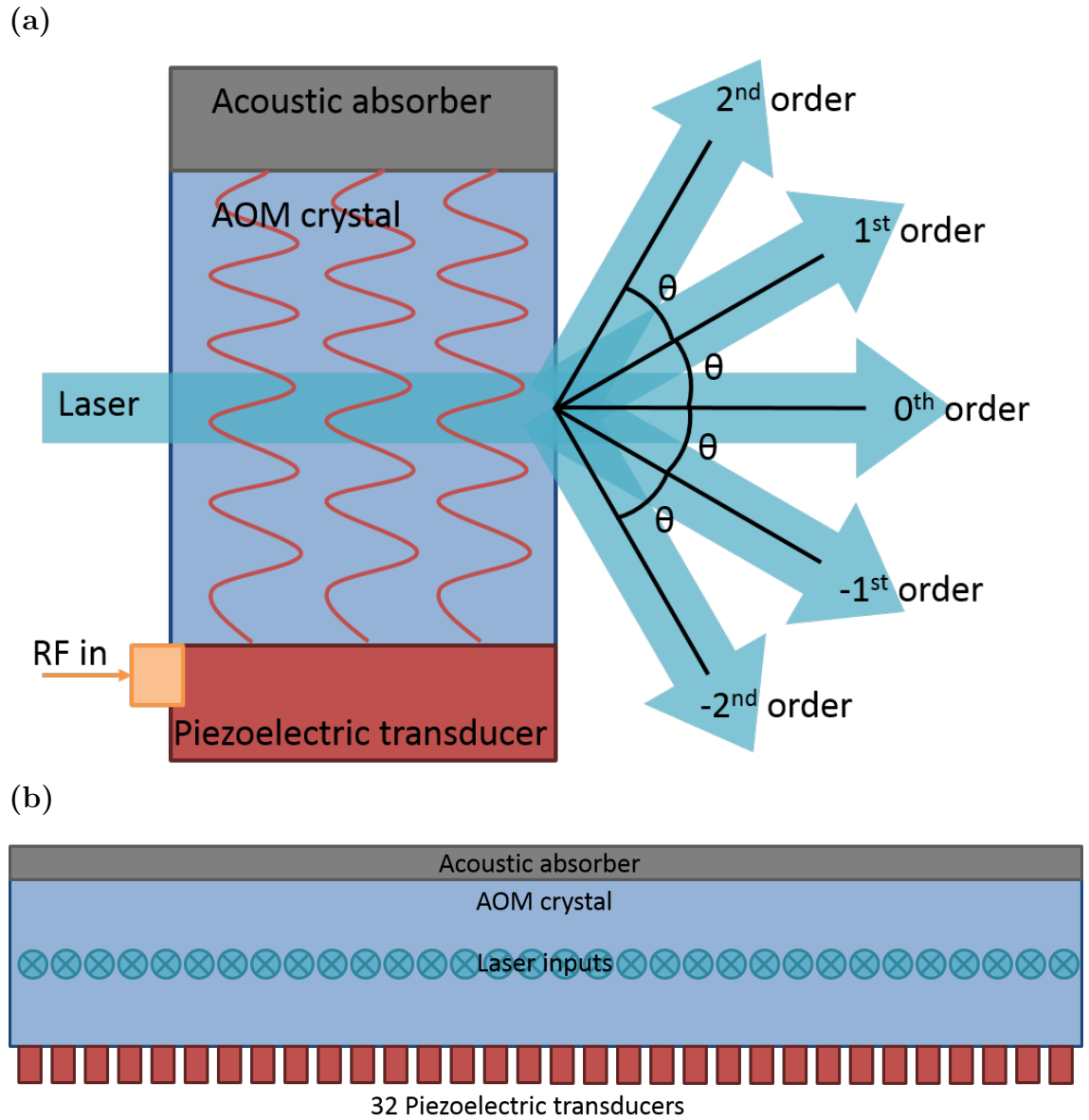


Figure 2.8: (a) Diagram showing how a single channel AOM works, with two positive and negative orders shown. (b) The 32-channel AOM differs from its more common single-channel counterparts in that it has 32 piezoelectric transducers on a single crystal, hence creating 32 channels that can independently modulate an input laser beam.

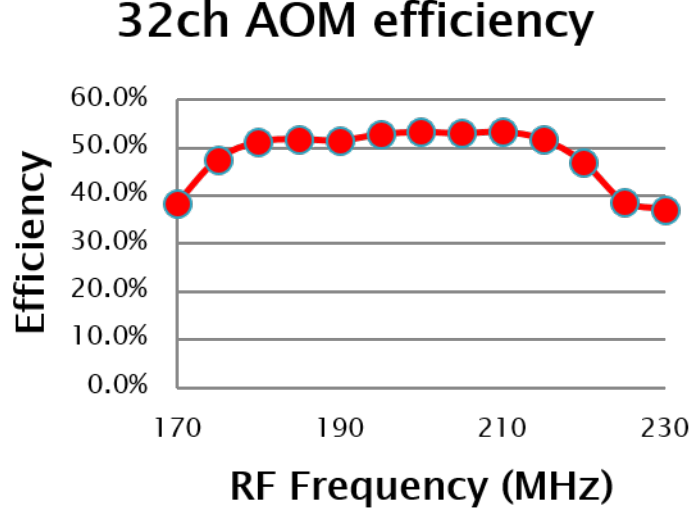


Figure 2.9: Efficiency for channel A15 of the 32-channel AOM. The 20 MHz range over which the efficiency is maximized is sufficient for experimental use, as the RF control frequency signal does not vary more than 10%.

2.10. First, to control 10 beams, we need 10 independent channels. A two-way power splitter (Minicircuit ZSCJ-2-1+) is used, and each output is then sent to a five-channel control box with a five-way splitter (Minicircuits ZFSC-5-1-SB+) for a total of 10 channels.

Second, we must provide a means of control over when a beam is deflected. A TTL switch (Minicircuits ZASWA-2-50DR+) is used on the RF input for each channel, which is in turn operated by the experimental control computer. Since the switch time is at most 20ns, this allows a given ion's Raman beam to be shut off or turned on quickly relative to qubit rotations that take tens of microseconds, or two-qubit gate times of hundreds of microseconds. We also provide a manual switch option.

Third, the 32-channel AOM is rated by the manufacturer to take up to 700mW

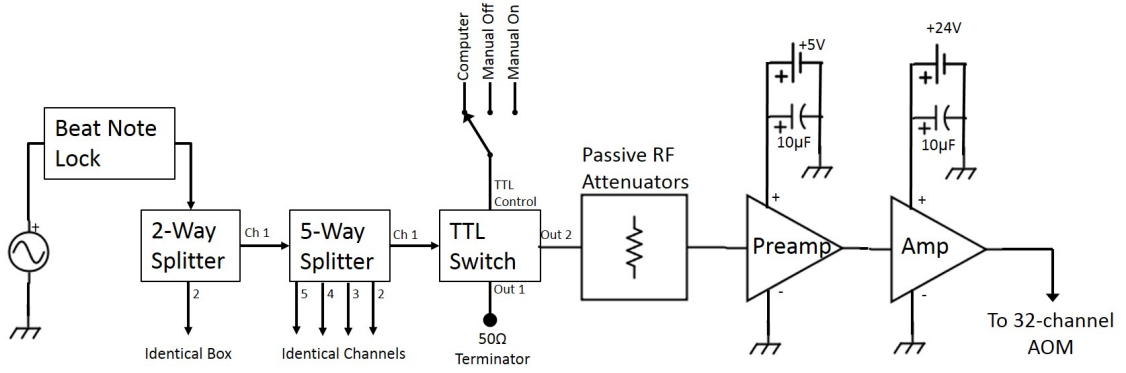


Figure 2.10: Component layout for RF control of 32-channel AOM.

(+28.45dBm) of RF power per channel. Hence, the signal control on each channel should be designed to produce no more than 700mW. To adhere to this limit, the beat note power output was measured, insertion loss from the TTL and splitters were calculated, and then amplifiers were chosen that would produce at least +29dBm (Minicircuits ZX60-P103LN+, gain +18dB; Minicircuits ZHL-2-12, gain +26dB). The channels were then individually calibrated by measuring the power output of each channel with a spectrum analyzer, and then adding passive attenuators before the pre-amplifier until the measured power output was less than +28.45dBm.

To ensure steady Rabi frequencies on each ion, there should be no power fluctuations over time on any given channel. Such fluctuations can be caused by fluctuations in the DC voltage provided to power the channel amplifier, so we used low-ripple Acopian power supplies. For the large amplifier, we used a Gold Box A24H850, which has a 0.25 mV ripple and provides +24VDC; for the pre-amplifier, we used 5EB150, which has a 1 mV ripple and provides +5VDC. For additional protection, bypass capacitors were installed on each power supply, the DC bulkhead

connection on the control box, and the DC power connection to the amplifier. These bypass capacitors of $10\ \mu\text{F}$ eliminate fluctuations by acting as a resistor to ground for any AC components that may be in the signal, while the DC component sees the capacitor as an open circuit and instead continues to the amplifier. It should be noted that small power differences ($< 1\text{dB}$) from one channel to the next are acceptable, as these differences can be measured and accounted for.

Another important consideration is that there cannot be phase variations between channels when individual phase control is not available. This requires that the path length of the RF signal be the same for all channels. Hence, we must be certain to use identical components for every channel, and same-length coaxial cables to connect corresponding components. Additionally, RF components were chosen that had very small phase imbalances.

2.3.4 Optics for Individual Addressing

In order to use the 32-channel AOM for individual optical addressing, an optical system must be designed and constructed to split 355nm light from the pulse laser into 10 beams, focus those beams into the AOM with $450\ \mu\text{m}$ beam spacing between adjacent acoustic modes and $< 50\ \mu\text{m}$ waist on each beam, and then focus those beams onto the ions in the trap, which requires $5\ \mu\text{m}$ beam spacing and $< 2\ \mu\text{m}$ beam waists. The problem is additionally complicated by the practical constraints of a lab: the limited real estate on an optical table limits the allowable beam path length, and due to commercial optics availability and quality, the total



Figure 2.11: An image of the 10 beams from the diffractive optic element, demonstrating even beam spacing, even power distribution, and very low-power higher-order beams.

beam diameter should not exceed 2 inches at any point on the beam path.

The 10-way beam splitting problem is solved with a diffractive optic element (DOE) from Holo-Or, part number MS-244-U-Y-A. It splits one incoming collimated Gaussian beam with waist at least 0.24 mm into 10 evenly spaced, collimated Gaussian beams of equal power and with the same waist as the input beam (see Figure 2.11). It has an observed diffraction efficiency of $> 56\%$ and a diffraction angle of 4.3 mrad.

Since the optical system to be designed has so many constraints, it was important to perform careful analysis of the optical path for each solution attempted. Additionally, both the spacing between the beams and the waist of each beam are important. Consequently, ray transfer matrix analysis was used to simultaneously keep track of beam spacing and beam widths. In the ray picture, we assume that each beam acts like a ray, and then keep track of one beam's displacement x from the optical axis to determine the spacing between beams. The optical system starts with the 10-way diffractive optic beamsplitter (10BS) splitting the incoming beam with a diffraction angle of $\theta = 4.3$ mrad. For simplicity, we will examine the size and displacement of a beam that is deflected $\theta = 4.3$ mrad from the optical axis, which is equivalent to examining the size and spacing between 2 beams that are each deflected $\theta = 2.15$ mrad on either side of the optical axis, which will be the

closest 2 beams for a diffractive optic element producing an even number of beams.

The initial ray vector is

$$r_0 = \begin{pmatrix} x_0 \\ \theta_0 \end{pmatrix} = \begin{pmatrix} 0 \\ 0.0043 \end{pmatrix}. \quad (2.4)$$

In the Gaussian picture, we instead assume each beam is a Gaussian beam, and use the complex beam parameter q to keep track of one beam's width and calculate relevant waists. For a Gaussian beam of wavelength $\lambda = 0.000355$ mm, radius of curvature $R = \infty$ for collimated beams, refractive index $n \approx 1$ for air, and radius w ,

$$\frac{1}{q_0} = \frac{1}{R} + \frac{i\lambda}{\pi n w^2} = \frac{0.000113i}{w^2}. \quad (2.5)$$

Using the ABCD matrix method, we can find the final complex beam parameter for a given system using

$$q_1 = \frac{Aq_0 + B}{Cq_0 + D} \quad (2.6)$$

and extract the beam radius from q_1 . For both the ray and Gaussian pictures, we use

$$\begin{pmatrix} A & B \\ C & D \end{pmatrix} = \begin{pmatrix} 1 & d \\ 0 & 1 \end{pmatrix} \quad (2.7)$$

to represent a distance d through free space, and

$$\begin{pmatrix} A & B \\ C & D \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ -\frac{1}{f} & 1 \end{pmatrix} \quad (2.8)$$

to represent a thin lens with focal length f . A naive solution would involve simply focusing the 10 beams into 10 channels of the AOM, and then using one lens to

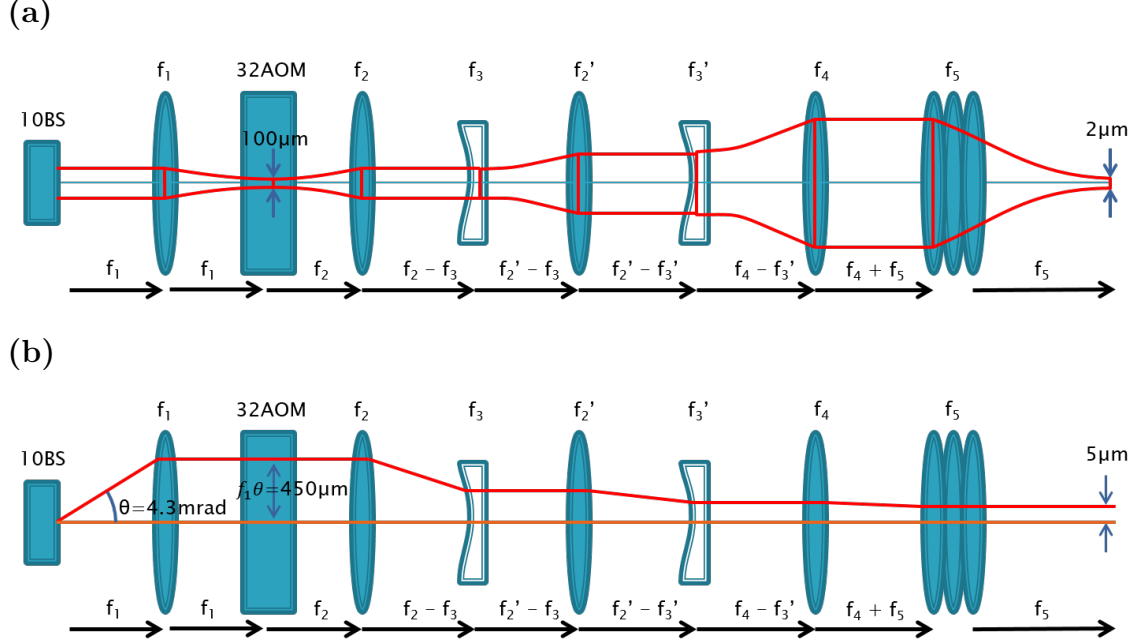


Figure 2.12: The solution for the optics system. The system starts with the 10-way diffractive optic element beamsplitter (10BS) and is then focused into the 32-channel AOM (32AOM) with the correct spacing. After that, the system uses alternating pairs of lenses to bring the separate beams closer together, while expanding each beam's radius to then focus very sharply with the final objective f_5 . The beam radius is expanded between f_3 and f_2' and between f_3' and f_4 , while the individual beams are kept parallel. The individual beams are brought closer together between lenses f_2 and f_3 , f_2' and f_3' , and f_4 and f_5 , where each beam is also kept collimated. For appropriate lens selection, this will achieve our desired waist of $1\mu\text{m}$ and spacing of $5\mu\text{m}$ at the ions. (a) The Gaussian solution shows the width of a single beam passing through the lens system. (b) The ray solution shows the spacing of 2 beams (one red, one orange) passing through the lens system.

image the AOM onto the ions. However, this solution requires that the imaging lens have a focal length of ~ 8 meters, which is entirely impractical. Instead, using the ray transfer analysis, we found a solution that uses two telescope arrangements after the 32-channel AOM to achieve the desired waist and spacing on the ions, while reducing the total length of the system. The solution is shown in Figure 2.12.

The first important step in the optics system is focusing the 10 beams into the AOM onto 10 separate channels. Since the diffractive angle $\theta = 4.3\text{ mrad}$ for the

10-way beamsplitter is small, we can write

$$x_{AOM} = f_1 \theta \quad (2.9)$$

for the spacing of the beams at the 32-channel AOM. So f_1 must be chosen to achieve the correct beam spacing at the AOM channels. After the AOM, the system's magnification can be calculated,

$$m = \frac{f_4 f_2 f'_2}{f_1 f_3 f'_3} = 28 \quad (2.10)$$

where ray transfer matrix analysis shows that $m = 28$ is the ideal magnification to yield a $5 \mu\text{m}$ spacing at the ions given a choice of a 32.7 mm objective (see next paragraph.) Much like in Equation 2.9, the final spacing is

$$x_f = \frac{f_5 \theta}{m} = 5 \mu\text{m}. \quad (2.11)$$

Taking into account constraints from commercially available lenses, the final lenses chosen are listed in Table 2.1. f_1 was chosen experimentally; the 90mm Thorlabs lens allowed us to successfully line up 10 parallel beams into 10 channels. For an input waist of 0.2 mm to the 10-way beamsplitter, this yields a waist of $55 \mu\text{m}$; while this yields a spot slightly larger than the AOM channel, we do not see any spillover since the channels are $450 \mu\text{m}$ apart, and the efficiency is still high enough that the solution is workable. For the telescopes after the 32-channel AOM, the magnification is $m = 74.2$, which is close to the ideal from equation 2.10.

Lens	Focal length at 355 nm	Lens used
f_1	87.0 mm	90 mm Thorlabs plano-convex
f_2	96.6 mm	100 mm Thorlabs plano-convex
f_3	-29.0 mm	-30 mm Thorlabs plano-concave
f_2'	120.8 mm	125 mm Thorlabs plano-convex
f_3'	-57.3 mm	Doublet: -150 mm +meniscus and -100 mm , Thorlabs plano-concave
f_4	483.4 mm	Doublet: 1000 mm and 1000 mm +meniscus, Thorlabs plano-convex
f_5	32.7 mm	Ronar-Smith objective, 0.25 NA, working distance 32.7 mm

Table 2.1: Final focal lengths for lenses chosen for the focusing lens system.

The final lens, f_5 , is a triplet objective made by Ronar-Smith with a 0.25 NA and working distance 32.7 mm that was chosen because it will minimize aberrations on the ion. Since it will be focusing a large-diameter beam to a very small focal point, it is important that f_5 be chosen carefully. The total system length, from 10-way beamsplitter to ions, is 1.59 m. To save more space, it is possible to move the objective f_5 back toward f_4 , thus reducing the total length to slightly over 1 m. This is because f_4 is a very slow lens; the individual beams have a very small deflection angle as they converge after the lens, and so placing the objective closer to f_4 will still yield beams very close to parallel at the ions. The beam spacing will still be correct at the working distance of the objective. This yields the additional advantage that the individual beams can be focused on the ions without having to move the entire optics system; instead, small adjustments in the objective will ensure the individual beams are focused at the ions. While the system will no longer be telecentric, meaning the beam spacing will vary with objective position, the amount of variance is negligible.

The lens system was assembled on the table (see Figure 2.13) and a camera

placed at the objective focus to image the results. The assembly yielded 10 beams separated by $5\text{ }\mu\text{m}$ and with waists of $1\text{ }\mu\text{m}$, as shown in Figure 2.14(a-b). Once the optics setup was achieved, a retroreflector was added to the global beam arm of the Raman beam setup to adjust and match the optical path length of the global beam arm to the individual beam arm. It is important to note that while the individual beam spacing is fixed, the ion spacing can be adjusted using the five DC electrodes on the blade trap; this will allow the beams to be aligned as well as possible to the ions in the trap. Coarse beam alignment was achieved through a rough method of observing direct scatter off the trap blades to align the Raman beams to the middle of the trap. Daily fine-tuning consists of optimizing the alignment by maximizing the observed Rabi frequency.

A higher-resolution 2D profile of 5 focused beams is shown in Figure 2.14(c), which was imaged using the ion response. With one ion in the trap, the individual beams were moved across the ion and the Rabi rate on the ion, which corresponds to the square root of the optical intensity seen by the ion, was measured. The beams were moved by moving the final imaging objective, which is mounted on a translation stage controlled by picomotors along all 3 axes. The picomotors allow the objective to be moved in steps of about 20 nm without opening the protective enclosure around the Raman optics, which reduces air currents that induce beam pointing instability and therefore noise on the experiment's coherent operations. These picomotors are also used to fine-tune the individual beam alignment on the ions, which drifts on a daily timescale.

The final ion positions relative to the center ion were $[-10.74, -5.05, 0, 5.16, 10.85]$

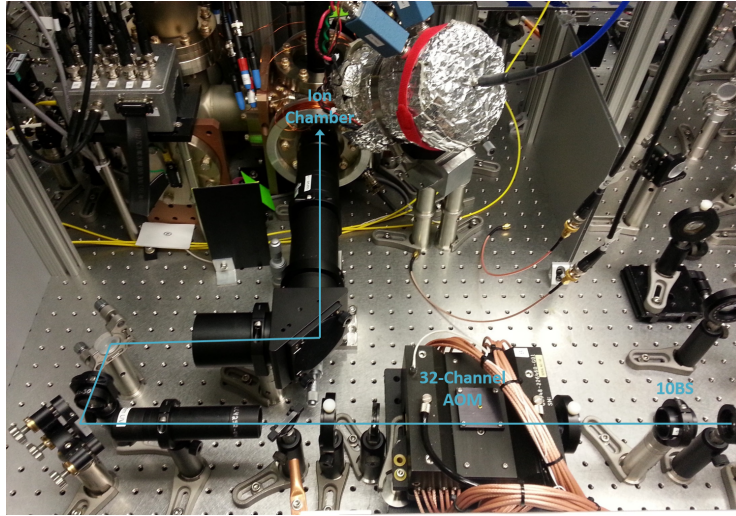


Figure 2.13: Photograph of the 32-channel AOM on the optics table. Beam path is shown in blue, along with the 10-way diffractive optic element (10BS), 32-channel AOM, and the ion chamber. (This is a slightly earlier version of the setup, with some different lenses.)

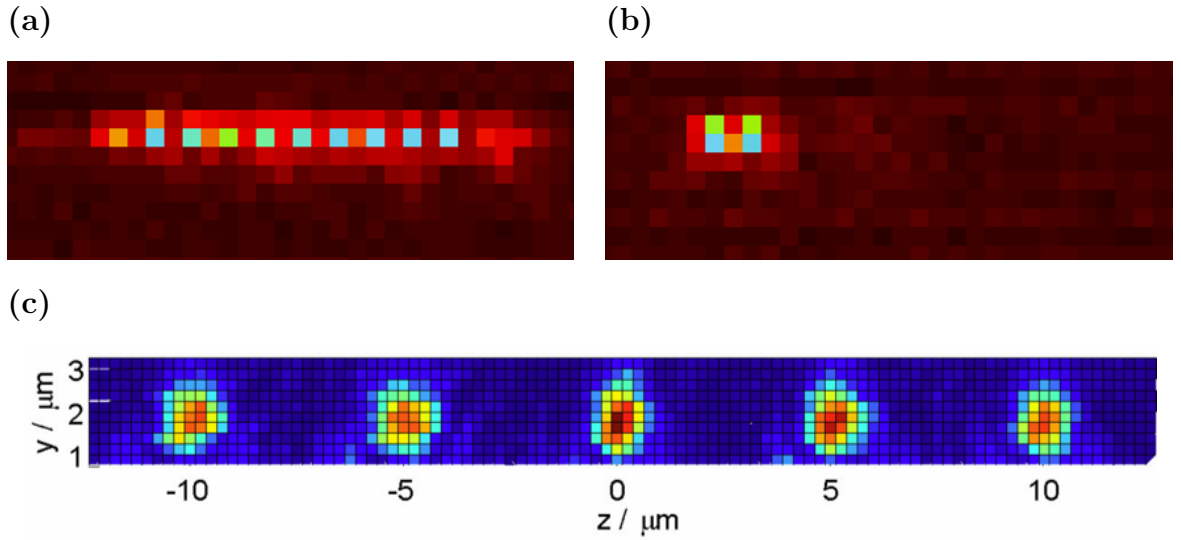


Figure 2.14: Images of focused beams to be applied to ions. 10 beams are available to shine on 10 ions; they can be turned off and on independently to achieve arbitrary individual addressing. (a) Camera image of 10 focused beams with $1\ \mu\text{m}$ waist and $4.5\ \mu\text{m}$ spacing; this image was taken with an earlier lens setup, and was later changed to a $5\ \mu\text{m}$ beam spacing. (b) Camera image of beams for ions 1 and 2 turned on, and the rest off. (c) 2D profile across 5 beams focused to $<1\ \mu\text{m}$ waist and separated by $5\ \mu\text{m}$, imaged by the ions themselves by moving the beams across the ions and measuring the resulting Rabi frequencies experienced by the ions.

μm , yielding spacing differences of $[5.69, 5.05, 5.16, 5.69] \mu\text{m}$, measured with an error of about $0.16 \mu\text{m}$.

2.4 Individual Addressing System Upgrades

Some upgrades to the individual addressing system have been added since the results reported in [19, 20], which will apply to some but not all of the results reported in this thesis. The Grover search algorithm discussed in Chapter 6 was performed before the following upgrades, but several more results, including those in Chapter 7 were made possible by the following upgrades.

For the earlier work on the experiment, a single AWG output provided RF control to the global beam AOM, supplying sidebands and pulse-shaping to control two-qubit entangling gates. The multi-channel AOM controlling the individual beams was controlled by a single RF source from the beatnote lock output; only on/off TTL control was available for individual beams.

A significant upgrade was enabled by the addition of a 4-channel AWG², bringing to 5 the total number of AWG channels available. This allowed for total phase, frequency, and amplitude control for the signal on each ion, not just on/off TTL control. To use these new capabilities, the AWG controls were transferred to the RF signals to the individual beam channels on the multi-channel AOM, and the beatnote lock signal was applied to the AOM on the global beam. The controls for the AWG were then integrated into the Igor control program using the NI-VISA com-

²Model WX1284C-1 1.25 GS/s Four Channel Arbitrary Waveform Generator, PN: 126182, Tabor Electronics Ltd.

munication protocol; a suite of utility functions were added in a separate procedure file that managed communication with the device and ensured uploaded waveforms adhered to the device’s parameters.

The additional, individually addressed controls provided by the AWG enabled several procedures previously impossible to implement, including simultaneous rotations, the simultaneous π pulses on different sidebands required to implement phonon hopping and blockades in [29], and the parallel two-qubit operations demonstrated in Chapter 7. Improved phase control for individual ions also enhanced experimental performance for all subsequent work by better compensating for differences in accumulated phases as operations are performed on different ions. Finally, the arbitrary phase control on each ion allowed for the classical implementation of Z rotations, further discussed in Section 3.1.2.

For further scaling in ion traps, however, continuing to rely on AWG control for individual addressing is not ideal; at a cost of about \$10,000 per channel, this is not cost effective. Efforts such as Sinara [54] have made important progress in this area, designing alternative control signal generators using an FPGA and a fast DAC to create an effective dynamic AWG; this design is estimated to lower costs by an order of magnitude to \sim \$1000 per channel. Using only commercially available non-AWG signal generators to generate control signals is unfortunately not currently possible, due to the need to change frequencies while maintaining phase coherence. Direct control over phase coherence also allows for compensation of effects like Stark shifts, which can reduce fidelities if left unaddressed.

2.5 Expansion to 7-Ion System

One experiment performed over the course of this thesis work required a 7-ion system (specifically [31]), and we anticipate performing more 7-qubit experiments in future. This presented several challenges. First and foremost was arranging for 7 ions positioned in the trap to align well with the individual addressing beams, which have fixed equal spacing due to the optical setup (see Section 2.3.4). As discussed in Section 2.2, perfectly equal spacing of more than 3 ions in this trap is not feasible. While 5 ions are still pretty close to equal spacing, more ions in the trap will have bigger and bigger spacing differentials between the outer and inner ions. Consequently, 7 ions will be less well-positioned relative to the individual addressing beams than for 5 ions. However, this can be mitigated by instead trapping 9 ions, and ignoring the outer 2 ions. The inner 7 ions are then pushed together by the sacrificial outer ions, resulting in a subchain of 7 ions with smaller spacing differentials than would occur with just 7 ions in the trap.

With this strategy, several candidate DC voltage sets based on simulations were tested on 9 ions in the trap to find the best experimental values. Once a voltage set was chosen, we next needed to align the individual beams on the not-quite-equally-spaced center 7 ions. Some experimentation demonstrated that the best alignment strategy was to try to maximize the alignment of all ions, but to particularly prioritize ions 2-6. This yielded reasonably comparable Rabi flops on all ions, further discussed in Section 3.1.1. The selected ion spacing yielded motional sidebands at [2.9424, 2.9604, 2.9774, 3.0044, 3.0149, 3.0264, 3.0384, 3.0534]

MHz detuned from the carrier. The ion positions relative to the center ion were $[-23.24, -16.49, -10.69, -5.21, 0, 5.27, 10.59, 16.49, 23.40] \mu\text{m}$, corresponding to differential spacings of $[6.76, 5.80, 5.48, 5.21, 5.27, 5.32, 5.90, 6.91] \mu\text{m}$, measured with an error of about $0.16 \mu\text{m}$.

As discussed in Section 2.4, we now have 5 AWG channels for full phase, frequency, and amplitude control on each individual addressing beam, but expanding to 7 qubits means we no longer have 1 AWG channel per qubit. We instead had some qubit controllers share AWG channels. For our initial setup, we had ions 1 and 7 share an AWG channel, and ions 2 and 6 share an AWG channel. This consisted of installing a TTL-controlled switch on the two shared AWG outputs that directed the AWG output to one qubit or the other, and adding software to the control program to seamlessly handle the shared controllers by queueing control sequences to the AWG and managing TTL timing accordingly. The tradeoff was that ions 1 and 7, and ions 2 and 6, could not be controlled at the same time, ruling out any 2-qubit gates between those pairs. For purposes of [31], we did not need those gates, so this was sufficient. Future experiments may require these gates, in which case we can set up a slightly more complicated TTL-controlled switching system to supply two different AWG channels to a given pair of ions to perform a 2-qubit entangling gate. Finally, we also wanted to maintain our existing 5-qubit infrastructure for ongoing 5-qubit experiments, so the control program was modified to allow easy switching between controls for 5 and 7 qubits; a few button pushes change the relevant parameters in the control program, 7 wires for the individual addressing RF signals need to be shuffled, and the alignment of the readout objective

is adjusted slightly to better align the ions to the PMT array.

Chapter 3: Quantum Control

In this chapter I discuss our two mechanisms for quantum control: single qubit rotations, or R gates, and two-qubit entangling interactions, or XX gates. These two operations form the gate set native to our hardware, and together they form a universal gate set that can implement any desired operation.

3.1 Single-Qubit Rotations R

In the matrix representation of quantum computation, we define the two qubit states as

$$|0\rangle = \begin{pmatrix} 1 \\ 0 \end{pmatrix} \qquad |1\rangle = \begin{pmatrix} 0 \\ 1 \end{pmatrix}. \quad (3.1)$$

Superposition states can be represented as points on the surface of the Bloch sphere (Figure 3.1), where $|0\rangle$ and $|1\rangle$ define the poles of the Z axis of the sphere. The polar angle θ represents the angular distance from the Z axis, and the azimuthal angle ϕ gives the projection in the XY plane running orthogonal to the Z axis. We define the azimuthal angle as $\phi = 0$ at the X axis; $\phi = \frac{\pi}{2}$ at the Y axis. Consequently, coherent transformations of a qubit can be described as rotations on

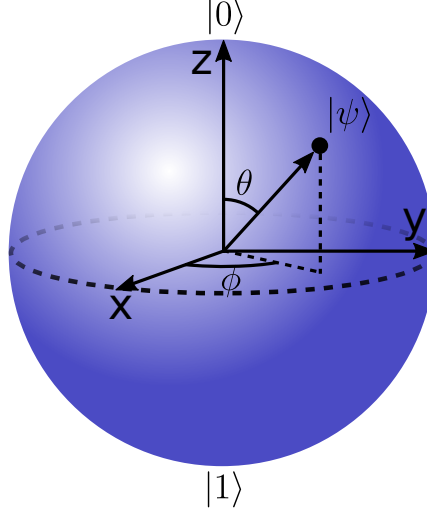


Figure 3.1: Quantum states can be represented as points on the surface of the Bloch sphere with the radial variables (θ, ϕ) . Single-qubit operations can be represented as rotations performed about a selected axis on the Bloch sphere.

the Bloch sphere.

Rotations of an angle θ about an axis described by the azimuthal angle ϕ in the XY plane are given by

$$R(\theta, \phi) = \begin{pmatrix} \cos \frac{\theta}{2} & -ie^{-i\phi} \sin \frac{\theta}{2} \\ -ie^{i\phi} \sin \frac{\theta}{2} & \cos \frac{\theta}{2} \end{pmatrix}, \quad (3.2)$$

with rotations about the X and Y axes given by

$$R_x(\theta) = R(\theta, \phi = 0) = \begin{pmatrix} \cos \frac{\theta}{2} & -i \sin \frac{\theta}{2} \\ -i \sin \frac{\theta}{2} & \cos \frac{\theta}{2} \end{pmatrix} \quad (3.3)$$

and

$$R_y(\theta) = R\left(\theta, \phi = \frac{\pi}{2}\right) = \begin{pmatrix} \cos \frac{\theta}{2} & -\sin \frac{\theta}{2} \\ \sin \frac{\theta}{2} & \cos \frac{\theta}{2} \end{pmatrix}. \quad (3.4)$$

Such rotations can be implemented on the experiment by applying a Raman pulse and adjusting the duration and phase of the pulse accordingly. The duration of

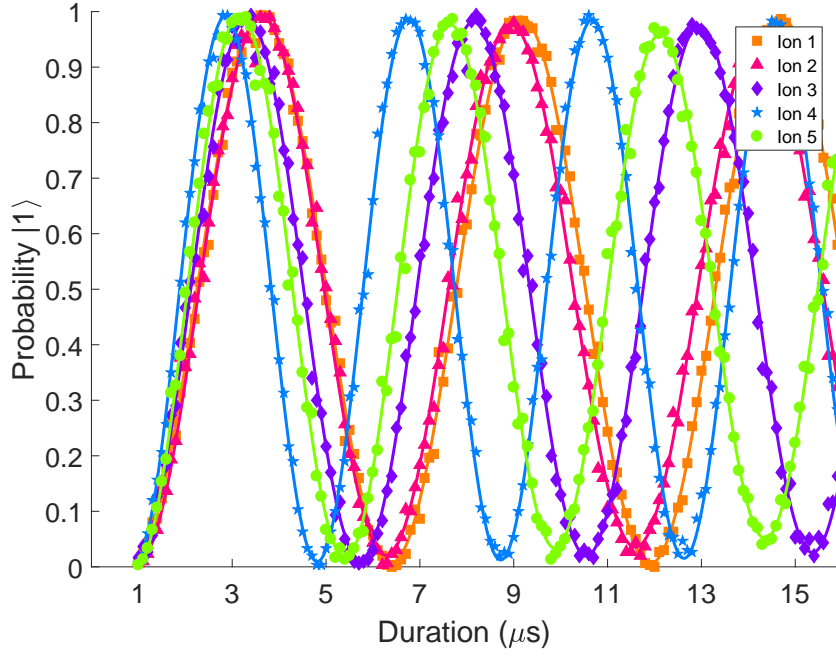


Figure 3.2: Sideband-cooled Rabi flops on 5 ions.

the pulse governs the magnitude of the rotation θ , to which it is directly proportional by way of the Rabi frequency. The duration required to transfer all population from the $|0\rangle$ to the $|1\rangle$ state, which corresponds to a $R(\theta = \pi)$ rotation, is known as the π time. Rotation calibration consists of measuring the π time for each ion, and using it to calculate the pulse duration necessary to implement each rotation: $\tau_\theta = \frac{\theta}{\pi} \cdot \tau_\pi$. The rotation angle ϕ is set by the phase of the control signal, with $\phi = 0$ defined to be the X axis.

3.1.1 Rabi Flopping on 5 or 7 ions

With individual Raman addressing beams in place, we can coherently cycle between the $|0\rangle$ and $|1\rangle$ states of each ion using Rabi flopping. Sideband-cooled Rabi flopping on 5 ions is shown in Figure 3.1.1. Applying Raman pulses for a

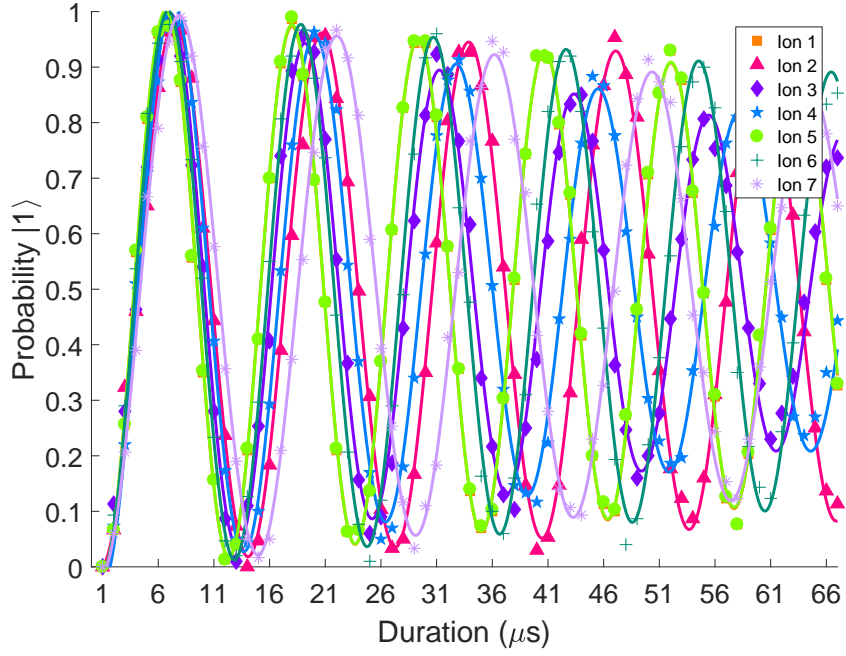


Figure 3.3: Sideband-cooled Rabi flops on 7 ions in a 9 ion chain at low power. Ion 1 is directly behind ion 5 in the plot.

controlled duration with a controlled phase allows us to implement coherent rotation gates. Typical rotation fidelities are 98-99%. Rotation fidelities are limited largely by intensity fluctuations on the Raman beams and imperfect cooling. Intensity fluctuations also inhibit our ability to accurately measure and calibrate the π times for gate implementation.

Rabi flopping on 7 ions is shown in Figure 3.1.1. after sideband cooling, the third Rabi flop peak was still above 90% bright for all ions, which is comparable to sideband-cooled Rabi flopping with 5 ions. The Rabi flops decay faster for ions that are less well aligned to the individual beams, which is primarily due to beam pointing fluctuations on the individual addressing beams; while this is still manageable for 7 ions, this problem gets worse as more ions are added to the chain and

the non-equally-spaced ions become less and less well matched to the equally-spaced individual beams.

3.1.2 Z Rotations

Qubit rotations about the Z axis on the Bloch sphere, given by

$$R_z(\theta) = \begin{pmatrix} e^{-i\frac{\theta}{2}} & 0 \\ 0 & e^{i\frac{\theta}{2}} \end{pmatrix}, \quad (3.5)$$

can be implemented on the ions in one of 2 ways. The first way is through a composite series of rotations about axes in the XY plane, as discussed in Section 5.1. The second way is by adjusting the phase of individual controllers relative to the ion. With this method, instead of performing a rotation on the qubit and moving it on the Bloch sphere, the controller is rotating the Bloch sphere around the qubit. This effectively rotates the XY plane around the Z axis and redefines the locations of the X and Y axes, and is equivalent to rotating the ion around the Z axis of the Bloch sphere. Subsequent operations are then implemented relative to the rotated axes.

The upgrades discussed in Section 2.4 permitted individual phase offsets to be applied differentially to individual ions. Because they can be implemented purely classically by simply shifting the phase of the frequency source, the error introduced by Z rotations is the same as the error of the frequency source itself. Consequently, Z rotation contributions to the error in fidelity during an algorithm are negligible compared to the error introduced by an $R(\theta, \phi)$ of the same θ and can be disregarded.

This has implications for composite sequence optimization further discussed in Section 5.2.1.

3.2 Two-Qubit Entangling XX Gates

We implement two-qubit entangling gates using a Mølmer-Sørensen scheme [41, 52, 55, 56] that creates an XX spin-spin interaction between two ions using the normal motional modes of the chain as an information bus. In this experiment, we couple to the transverse phonon modes [57, 58] to drive the interaction.

The two-qubit XX entangling gate is

$$XX(\chi) = \begin{pmatrix} \cos(\chi) & 0 & 0 & -i \sin(\chi) \\ 0 & \cos(\chi) & -i \sin(\chi) & 0 \\ 0 & -i \sin(\chi) & \cos(\chi) & 0 \\ -i \sin(\chi) & 0 & 0 & \cos(\chi) \end{pmatrix}, \quad (3.6)$$

where the parameter χ can be varied continuously, $0 < \chi \leq \frac{\pi}{4}$, by adjusting the overall power applied to the gate. The gate is maximally entangling for $\chi = \pm \frac{\pi}{4}$, where

$$XX\left(\chi = \frac{\pi}{4}\right) = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 0 & 0 & -i \\ 0 & 1 & -i & 0 \\ 0 & -i & 1 & 0 \\ -i & 0 & 0 & 1 \end{pmatrix}, \quad XX\left(\chi = -\frac{\pi}{4}\right) = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 0 & 0 & i \\ 0 & 1 & i & 0 \\ 0 & i & 1 & 0 \\ i & 0 & 0 & 1 \end{pmatrix}. \quad (3.7)$$

When applied to two ions initialized to $|00\rangle$, this yields a maximally entangled state:

$$\begin{aligned} XX\left(\chi = \frac{\pi}{4}\right)|00\rangle &= \frac{1}{\sqrt{2}}(|00\rangle - i|11\rangle) \\ XX\left(\chi = \frac{\pi}{4}\right)|00\rangle &= \frac{1}{\sqrt{2}}(|00\rangle + i|11\rangle). \end{aligned} \tag{3.8}$$

The sign of χ for a given gate depends on the interactions between the two ions in question and the modes of motion primarily excited by the gate to produce the spin-spin entanglement. The sign of the gate parameter χ is measured empirically for each XX gate pairing. Accommodation for the difference in χ signs is handled by the compiler at the gate library level (see Chapter 5). For ease of discussion, I will only reference the $\chi = \frac{\pi}{4}$ case for the remainder of this chapter.

3.2.1 Pulse-Shaping Scheme

We implement entangling XX gates using a pulse-shaping scheme designed to ensure that the motional modes are fully detangled from the qubits at the end of the gate, leaving only spin-spin entanglement [53, 57, 59]. Two ions in a chain of N ions are illuminated with red and blue sidebands that are detuned near the normal transverse modes of motion and couple the modes to the qubit spins. The pulse-shaping scheme is also robust against detuning errors, and the number of pulses needed to implement gate solutions grow linearly with the number of ions in the chain. The detuning μ and gate time τ are independent parameters that can be chosen before calculating a gate pulse shape, and a unit fidelity gate is possible at any detuning [53].

Because we couple to the normal motional modes of the chain to induce XX gates, all ions participate in the motional modes, and so couplings are available between all possible ion pairs. This fully connected architecture has significant advantages over less-connected architectures, which require additional gates to transfer quantum information between unconnected qubits [27].

On this experiment, we have also implemented entangling XX gates using a pulse-shaping scheme that modulates the frequency of the gate driver, rather than the amplitude [28].

In order to entangle ions i and j in a chain of N ions with N motional modes ω_k using red and blue sidebands with detuning μ , we have the following 2-qubit unitary:

$$\begin{aligned} U_{XX}(\tau) &= \exp \left(i \sum_i \phi_i(\tau) \sigma_i^x + i \sum_{i < j} \chi_{ij}(\tau) \sigma_i^x \sigma_j^x \right) \\ &= \exp \left(i \left[\phi_i(\tau) \sigma_i^x + \phi_j(\tau) \sigma_j^x + \chi_{ij}(\tau) \sigma_i^x \sigma_j^x \right] \right) \end{aligned} \quad (3.9)$$

where τ is the gate time, the spin-motion interaction $\phi_i(\tau)$ is

$$\phi_i(\tau) = \alpha_{i,k}(\tau) \hat{a}_k^\dagger - \alpha_{i,k}^*(\tau) \hat{a}_k, \quad (3.10)$$

\hat{a}_k^\dagger and \hat{a}_k are the raising and lowering operators for the motional phonons, the spin-motion parameter $\alpha_{i,k}(\tau)$ is

$$\alpha_{i,k}(\tau) = \int_0^\tau \eta_{i,k} \Omega_i(t) \sin(\mu t) e^{i\omega_k t} dt, \quad (3.11)$$

$\eta_{i,k}$ is the spin-motion coupling or Lamb-Dicke parameter, $\Omega_i(t)$ is the Rabi frequency of the optical field applied to ion i , and the spin-spin interaction term $\chi_{ij}(\tau)$ is

$$\chi_{ij}(\tau) = 2 \int_0^\tau dt' \int_0^{t'} dt \sum_k \eta_{i,k} \eta_{j,k} \Omega_i(t) \Omega_j(t) \sin(\mu t) \sin(\mu t') \sin(\omega_k(t' - t)). \quad (3.12)$$

At the end of the gate, the spin-motion terms must go to zero, ensuring that all mode trajectories in phase space return to the origin. So, we require all $2N$ spin-motion parameters (2 ions, N modes) $\alpha_{\{i,j\},k}(\tau) = 0$. Entangling the ion pair (i, j) requires that the spin-spin coupling term $\chi_{ij} = \chi_{ij}^{\text{ideal}}$, where $0 < \chi_{ij}^{\text{ideal}} \leq \frac{\pi}{4}$. χ_{ij}^{ideal} is typically $\frac{\pi}{4}$ for a maximally entangling XX gate but can be set to smaller values to implement partially-entangling gates. This then yields a set of $2N + 1$ parameters to control for when calculating pulse sequences to implement an XX gate:

$$\begin{aligned} \alpha_{\{i,j\},k}(\tau) &= 0 \\ \chi_{ij}(\tau) &= \chi_{ij}^{\text{ideal}}, 0 < \chi_{ij}^{\text{ideal}} \leq \frac{\pi}{4}. \end{aligned} \quad (3.13)$$

To provide full control during the gate and fulfill the constraints in Equation 3.13, we divide up the gate amplitude $\Omega_i(t)$ into S segments of equal duration τ/S , and vary the amplitude in each segment Ω_s as an independent variable. The gate

amplitude on a the ions then becomes a piecewise-constant function,

$$\Omega_i(t) = \begin{cases} \Omega_1 & 0 \leq t < \tau/S \\ \Omega_2 & \tau/S \leq t < 2\tau/S \\ \vdots & \vdots \\ \Omega_s & (s-1)\tau/S \leq t < s\tau/S \\ \vdots & \vdots \\ \Omega_S & (S-1)\tau/S \leq t < \tau. \end{cases} \quad (3.14)$$

Consequently, Equations 3.11 and 3.12 can be re-written as

$$\begin{aligned} \alpha_{i,k}(\tau) &= \sum_{s=1}^S \Omega_s \left[\eta_{i,k} \int_{(s-1)\tau/S}^{s\tau/S} \sin(\mu t) e^{i\omega_k t} dt \right] \\ &= \sum_{s=1}^S \Omega_s C_{k,s}^i \end{aligned} \quad (3.15)$$

and

$$\begin{aligned} \chi_{ij}(\tau) &= \sum_{s=1}^S \sum_{s'=1}^S \Omega_s \Omega_{s'} \int_{(s-1)\tau/S}^{s\tau/S} dt' \int_{(s'-1)\tau/S}^{s'\tau/S} dt \sum_k \eta_{i,k} \eta_{j,k} \sin(\mu t) \sin(\mu t') \sin(\omega_k(t' - t)) \\ &= \sum_{s=1}^S \sum_{s'=1}^S \Omega_s \Omega_{s'} D_{s,s'}, \end{aligned} \quad (3.16)$$

where the terms

$$C_{k,s}^i = \eta_{i,k} \int_{(s-1)\tau/S}^{s\tau/S} \sin(\mu t) e^{i\omega_k t} dt \quad (3.17)$$

and

$$D_{s,s'}^{ij} = \int_{(s-1)\tau/S}^{s\tau/S} dt' \int_{(s'-1)\tau/S}^{s'\tau/S} dt \sum_k \eta_{i,k} \eta_{j,k} \sin(\mu t) \sin(\mu t') \sin(\omega_k(t' - t)) \quad (3.18)$$

are pre-calculated constants that are functions only of the motional mode frequencies ω_k , the detuning μ , and the segment number s , and so can be arranged into $S \times N$ and $S \times S$ matrices for each ion, respectively. Note that the time ordering of the double integral in Equation 3.18 requires that $t < t'$, so the time-segmented scheme requires $s \leq s'$. In the case $s = s'$, we must force the $t < t'$ constraint, yielding

$$D_{s=s'}^{ij} = \int_{(s-1)\tau/S}^{s\tau/S} dt' \int_{(s'-1)\tau/S}^{t'} dt \sum_k \eta_{i,k} \eta_{j,k} \sin(\mu t) \sin(\mu t') \sin(\omega_k(t' - t)). \quad (3.19)$$

If we now arrange the segment amplitudes Ω_s into the vector $\mathbf{\Omega}_{ij}$, we can now write our constraint equations from Equation 3.13 as

$$\begin{pmatrix} \mathbf{C}^i \\ \mathbf{C}^j \end{pmatrix} \mathbf{\Omega}_{ij} = 0$$

$$\mathbf{\Omega}_{ij}^T \mathbf{D}^{ij} \mathbf{\Omega}_{ij} = \chi_{ij}^{\text{ideal}} \quad (3.20)$$

where $\{\mathbf{C}^i, \mathbf{C}^j\}$ are the $S \times N$ spin-motion interaction matrices for each segment on each ion and \mathbf{D}^{ij} is the $S \times S$ spin-spin interaction matrix for each segment. With the constraints in Equation 3.20, we can calculate a pulse shape on a given pair of ions that will create an XX entangling interaction between them.

An XX gate implemented with such a scheme has a theoretical maximum unit

fidelity. From [59], the gate fidelity as a function of the spin-motion closure terms $\alpha_{\{i,j\},k}(\tau)$ for a fully entangling gate with $\chi_{ij}^{\text{ideal}} = \frac{\pi}{4}$ is

$$F_{XX} \left(\chi = \frac{\pi}{4} \right) = \frac{1}{8} [2 + 2(\Gamma_i + \Gamma_j) + \Gamma_+ + \Gamma_-], \quad (3.21)$$

where

$$\begin{aligned} \Gamma_{\pm} &= \exp \left(-\frac{1}{2} \sum_k \beta_k |\alpha_{i,k} \pm \alpha_{j,k}|^2 \right) \\ \Gamma_{i(j)} &= \exp \left(-\frac{1}{2} \sum_k \beta_k |\alpha_{i(j),k}|^2 \right) \end{aligned} \quad (3.22)$$

and the inverse mode temperature β_k is

$$\beta_k = \coth \left(\frac{\hbar \omega_k}{k_B T} \right) = \coth \left[\frac{1}{2} \ln \left(1 + \frac{1}{\bar{n}_k} \right) \right], \quad (3.23)$$

where \bar{n}_k is the average phonon number in the k^{th} mode. Errors on the gate arise if the motional modes are not fully closed in phase space, in which case the number and temperature of any phonons in these modes determine the extent of experimental errors.

Since we are also interested in partially-entangling gates, we re-calculate this fidelity for a general value of χ_{ij} , and obtain

$$F_{XX}(\chi_{ij}) = \frac{1}{8} [2 + 2(\Gamma_i + \Gamma_j) \cos(2\Delta\chi) + \Gamma_+ + \Gamma_-] \quad (3.24)$$

where $\Delta\chi = \chi_{ij}^{\text{ideal}} - \chi_{ij}$. Plugging in the ideal-case parameters, where $\alpha_{\{i,j\},k}(\tau) = 0$

and $\chi_{ij} = \chi_{ij}^{\text{ideal}}$, we indeed get $F_{XX} = 1$.

An XX gate between a given pair of ions is implemented by finding a pulse shape solution that fulfills the constraints in Equation 3.13. While in principle $2N + 1$ segments are needed for a given gate to control for the $2N + 1$ constraints, in practice gates can be constructed with fewer segments. The modes closest to the detuning μ have the biggest participation in the spin-motion coupling, but those far from the detuning participate very little and don't move far from the origin in phase space. Consequently, fewer segments can be used, and the solution optimized to close phase space for the high-participation modes where phase space closure matters most.

Several solutions at different detunings are tested on the experiment to ensure a high-quality gate is found. While all solutions in principle have high fidelity, anharmonicities in the experiment mean that not all solutions will perform well on our experiment. The process must be repeated for each ion pairing one wishes to use. While this process of pre-calculating and testing solutions incurs some overhead, once a good solution has been found, no further tweaks to the solution are required. The only day-to-day calibration needed is to measure the overall scaling factor to the gate necessary to correctly implement a fully-entangling gate, which is a function of the Rabi frequency and hence subject to drifts in the beam intensity seen by the ions.

An example pulse shape with 9 segments is shown in Figure 3.4, including plots of the trajectory of each mode in phase space over the course of the gate, and a parity scan from which fidelity can be calculated. The pulse-shaping scheme was

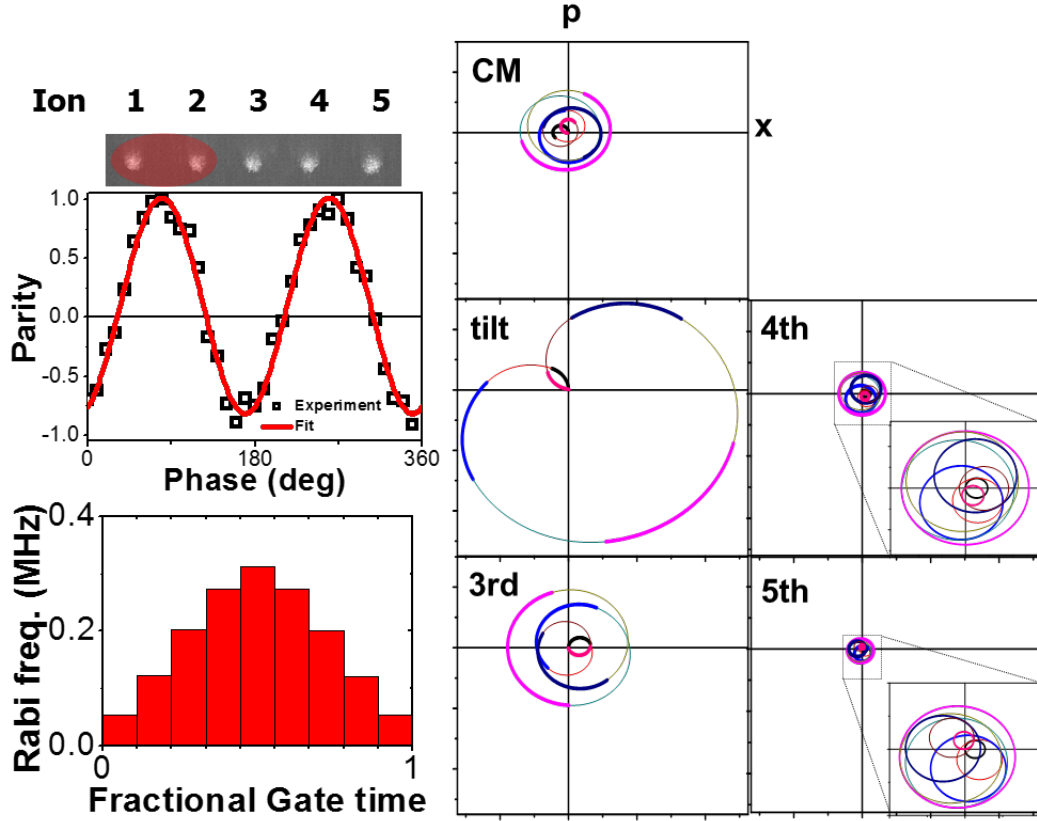


Figure 3.4: We use a pulse-shaping scheme to implement robust, high-fidelity XX gates. Here, we show a pulse shape used to implement a gate on ions 1 and 2 in a previous hardware setup; the spin-motion trajectories in phase space successfully close at the end of the gate. Parity scan data shows the gate fidelity here is 95(2)% [53]. Improvements in the hardware and analysis allowing for SPAM correction have since increased typical XX gate fidelities to 98-99%.

implemented on two ions in a five-ion chain in a previous experimental setup with a different trap [53]. The hardware improvements gained using our blade trap setup, improved individual addressing, and SPAM correction analysis have since raised typical 2-qubit gate fidelities to 98-99%. Fidelities for XX gates can be measured by scanning the phase of a global $\frac{\pi}{2}$ rotation applied after the XX gate and calculating the parity, as described in Section 7.3.1. Solutions were calculated with gate time $\tau = 230\mu s$. The gate time can be adjusted on the experiment a little to maximize gate fidelities, yielding experimental values of 210-260 μs . We measure motional sidebands at $\omega_k/(2\pi) = [2.946, 2.978, 3.005, 3.027, 3.045]$ MHz detuned from the carrier.

I will revisit this pulse shaping method to implement a scheme for parallel XX gates in Chapter 7.

3.3 Universality of the Gate Set $\{R, XX\}$

With quantum control consisting of single-qubit rotation R gates and two-qubit entangling XX gates, we now want to be able to construct more interesting unitaries. In fact, to be a fully programmable quantum computer, we require that we have access to a universal gate set that can implement any possible unitary, just as classical computers have access to a universal gate set that can implement any possible classical computation. Using the quantum circuit model of computation [60], 2-level unitaries have been shown to be universal, meaning that any unitary can be built from 1- and 2-qubit gates [61]; however, deterministic ways of building

such unitaries do not necessarily yield the most efficient gate sequence for doing so. Several gate sets have been shown to be universal, including the set of all 1-qubit rotations and a $CNOT$ gate [62, 63], as well as the more commonly cited universal gate set of $\{CNOT, H, T\}$ [64, 65].

Any single-qubit rotation can be constructed using a sequence requiring two orthogonal Pauli rotations [65]. Since we have access to all rotations $R(\theta, \phi)$ in the XY plane of the Bloch sphere, we can construct any single-qubit rotation R . Additionally, as will be shown in Figure 5.2(a), a $CNOT$ gate can be constructed from an XX gate and several single-qubit rotations R . Consequently, our available gate set of $\{R, XX\}$ is universal, and we can use it to implement any quantum algorithm on this programmable quantum computer.

A toolbox of useful quantum gates have been developed in the literature, and their unitaries decomposed into already-known quantum gates [65, 66]. In the next chapter, I will discuss the gate library of composite gates that has been developed and implemented on our machine so far.

Chapter 4: Control System

Chapters [2-5](#) presented the hardware, quantum control, and gate library layers of a full-stack ion trap quantum computer. The highest level of the stack is the implementation of quantum algorithms, and the user interface where this level of control happens. While our control system software is hardly a production-ready product, it nevertheless uses principles of abstraction and compartmentalization to control the quantum computer, and make it easy for a knowledgeable user to implement a quantum algorithm without having to worry about lower levels of control. Here, I present in some detail the workings of the control software, built in Igor, and show how all the parts of the system are stitched together by the control software using a systems-level perspective.

In this chapter, I will discuss the compiler layer of the stack, as well as the experimental control program it is embedded in. This will provide a systems-level perspective on the experimental setup, as the control program manages every step in the experimental process. The control program is written with Wavemetrics IgorPro, a data processing tool with some instrumentation interfacing and GUI construction capabilities. Our control program uses Igor in ways it wasn't really designed for, and as such we've had to add some creative workarounds to push the

boundaries of the program’s capabilities. Other projects, such as ARTIQ [67] and the pyIonControl project through Sandia National Labs [68], have produced open-source ion trap control programs in more versatile and widely-used languages like Python that may prove more suitable with sufficient development.

In broad strokes, the Igor program provides the GUI the user interacts with, the compiler that interprets algorithm sequences, and ongoing control of certain low-level experimental processes necessary to trap and hold a small chain of ions. When the user initiates an experiment, the Igor program compiles the user’s inputs, assembles the complete experimental procedure, and hands it off to the FPGA for real-time execution. The experiment requires a number of steps to be performed by disparate experimental elements quickly and with precise timing, so an FPGA, which can initiate and terminate TTL signals to other instrumentation on a 20ns clock cycle, is ideal for such a role. When the user writes an algorithm for the experiment to execute, they are writing a melody. The compiler then provides the harmonies for the melody, the control program arranges the score for a full orchestra, and when the user starts the experiment running, Igor hands the score off to the FPGA, which then conducts the symphony orchestra that is our ion trap quantum computer, ensuring each instrument - lasers, RF sources, electronic controllers, and so on - comes in at the right time to create a harmonious piece of music.

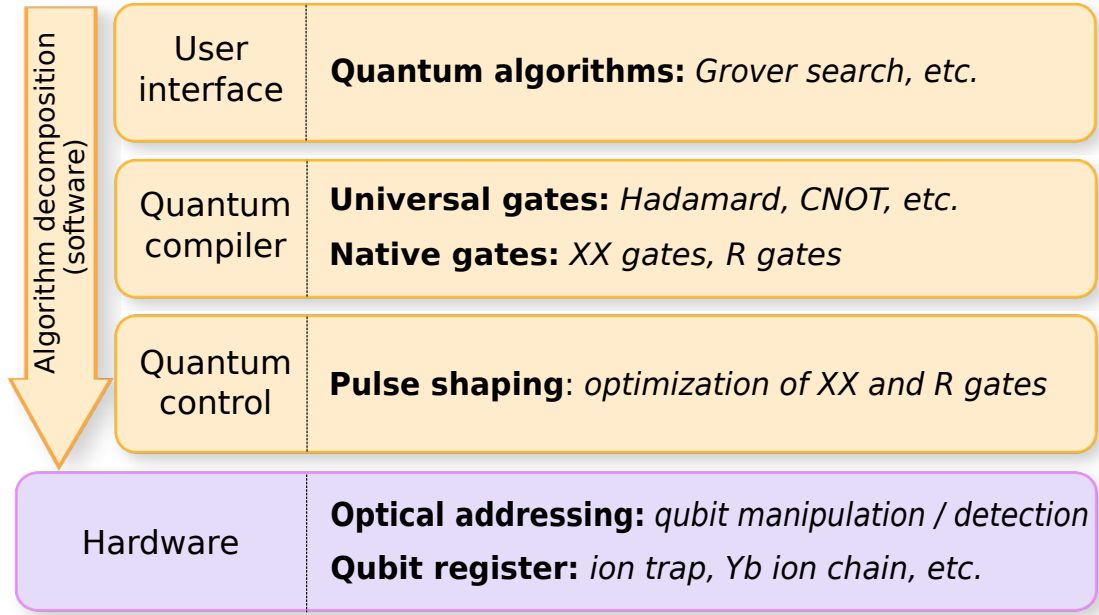


Figure 4.1: A diagram of the computational stack, from the low-level hardware up to algorithm implementation through a user interface. Image modified from [19].

4.1 System Control Overview

Figure 4.2 provides a systems-level view of the experimental control structure. The Igor control program, run on a commercial PC, provides a user interface that accepts user instructions and translates them into an experimental sequence. This sequence and related control information is then passed to peripherals that execute the instructions on the physical system. During an experimental run, an FPGA is used to initiate operations in real time. The 20 ns clock cycle allows for fast and consistently precise timing of experimental operations, which happen on microsecond timescales; a typical consumer computer is too slow and inconsistent for our purposes. Consequently, the Igor program pre-sets any instrumentation that uses static values before an experimental sequence begins. During experiments, the

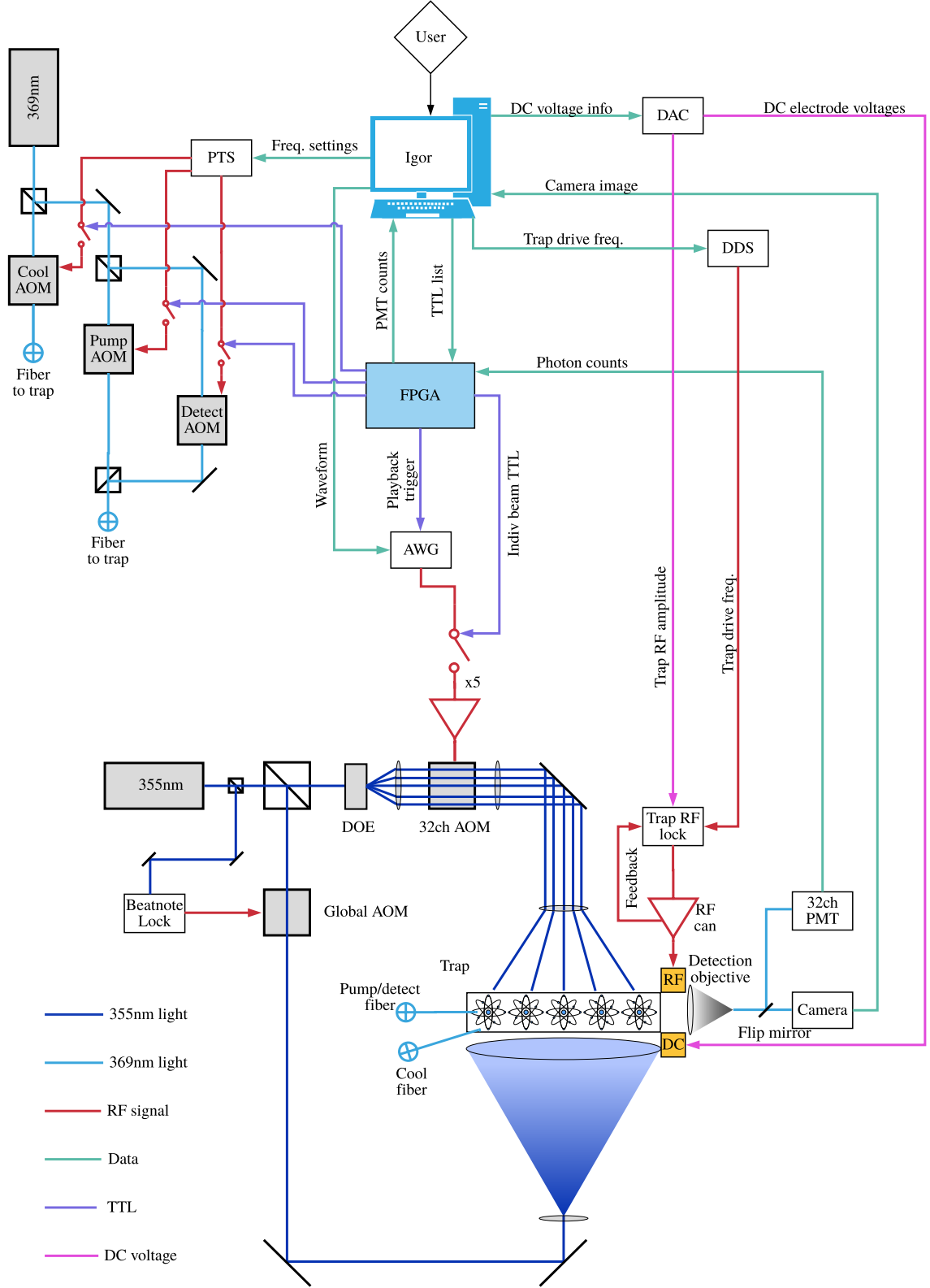


Figure 4.2: An overview of the control setup for all major pieces of the experiment. A note that all beams after AOMs are deflected first orders; the dumped 0th order and the angle of deflection are omitted here for simplicity of viewing.

FPGA sends TTL pulses at pre-specified times to trigger the instrumentation in the correct order for the correct length of time.

The Igor program sets the trap voltages using a DAC (digital to analog converter.) When the user changes the DC electrode voltages, the new voltage values are sent to the DAC, which then outputs those voltages to the corresponding trap electrodes. The DAC also sets the amplitude of the RF drive, by outputting a voltage that controls a VVA (voltage variable attenuator) on the RF drive signal. The trap drive signal is provided by a DDS frequency source, which is set through the Igor program. The drive signal is amplified by a helical resonator [44], colloquially referred to as an RF can, and its output is actively stabilized with a lock [43]. Real-time shuttling can be implemented by triggering the DAC with the FPGA.

The Doppler cooling, optical pumping, and detection operations are all implemented similarly. The Igor control program statically sets the frequency and amplitude of a PTS¹ frequency synthesizer for each operation. Each PTS channel then outputs an RF signal at the specified frequency and amplitude that, when applied to the corresponding AOM, deflects the incoming 369 nm beam. The deflected beams are aligned along the rest of the optical pathway to the ions and the zero orders are not, so applying the driving RF to the AOM turns the beam “on” from the points of view of the ions. A TTL-controlled switch on the RF signal serves as an on-off switch for each beam. During an experiment, the FPGA triggers each operation by sending a TTL pulse to the switch, thereby applying the beam to the ions.

¹Programmed Test Sources, Inc., model PTS 3200

The AWG provides the RF control signal used to implement coherent operations. Before an experiment starts, the Igor program calculates the AWG waveform to be used in the experiment, which requires specifying the voltage the AWG should output at every point in time. The program calculates the voltage output with 14-bit resolution for every 1 ns timestep. The waveform is then uploaded to the AWG, and repeated for all channels. During run time, the FPGA sends a single TTL trigger to the AWG, at which point it plays back the entire waveform stored in each channel. The FPGA also sends TTL pulses to “turn on” individual beams, timed to correspond with the AWG waveform needing to be applied to that channel. As with the 369 nm AOM controls, this works by having the TTL from the FPGA control a switch on the RF drive applied to each AOM channel in the 32-channel AOM. These switches and TTLs proved particularly useful for controlling 7 ions with only 5 AWG channels.

Data is collected by a multi-channel PMT, which counts photons collected through an objective lens with an NA of 0.37. The detection light from each ion is imaged onto a different PMT channel. After each experimental shot, the photon counts for each channel are sent to the FPGA, which collects it for the specified number of experimental repetitions per data point, then sends the compiled photon counts back to the Igor control program for analysis. A flip mirror controls whether the detection light is directed onto the PMT channels, or instead to an ICCD camera, which is used for 2D imaging of ions while loading. The camera is controlled entirely by the Igor program; a GUI provides a user interface for setting the gain, exposure, and other parameters, and the camera data is sent back to the control program for

display.

4.2 The Igor Control Program

The Igor control program provides a user interface consisting of GUIs that control nearly all aspects of the experiment. The front end allows the user to perform several activities through GUIs. The user can create new experimental sequences, consisting of a series of experimental operations. They can load such a sequence and adjust the parameters of experimental operations in the sequence. They can change static controls on certain peripherals, such as the ion trap voltage controls and the frequency and amplitude settings for the RF signals controlling the cool, pump, and detect AOMs. Finally, the user can initiate an experimental run and select from various options for data display. Two modes of experimental run are available: scanning and alignment sweeper. In a scan, the user designates one or more experimental operation parameters to scan over, and provides a start value, stop value, and scan increment. The program collects and displays a data point at each scan value and then stops. In alignment sweeper mode, the experiment is performed with the same values for each data point, and repeats until the user stops it. This mode is particularly useful for continually running and updating experimental outcomes while aligning or adjusting an apparatus on the experiment, hence the name.

The back end of the Igor control program consists of several Igor procedure (.ipf) files that handle each modular aspect of the control system. An overview

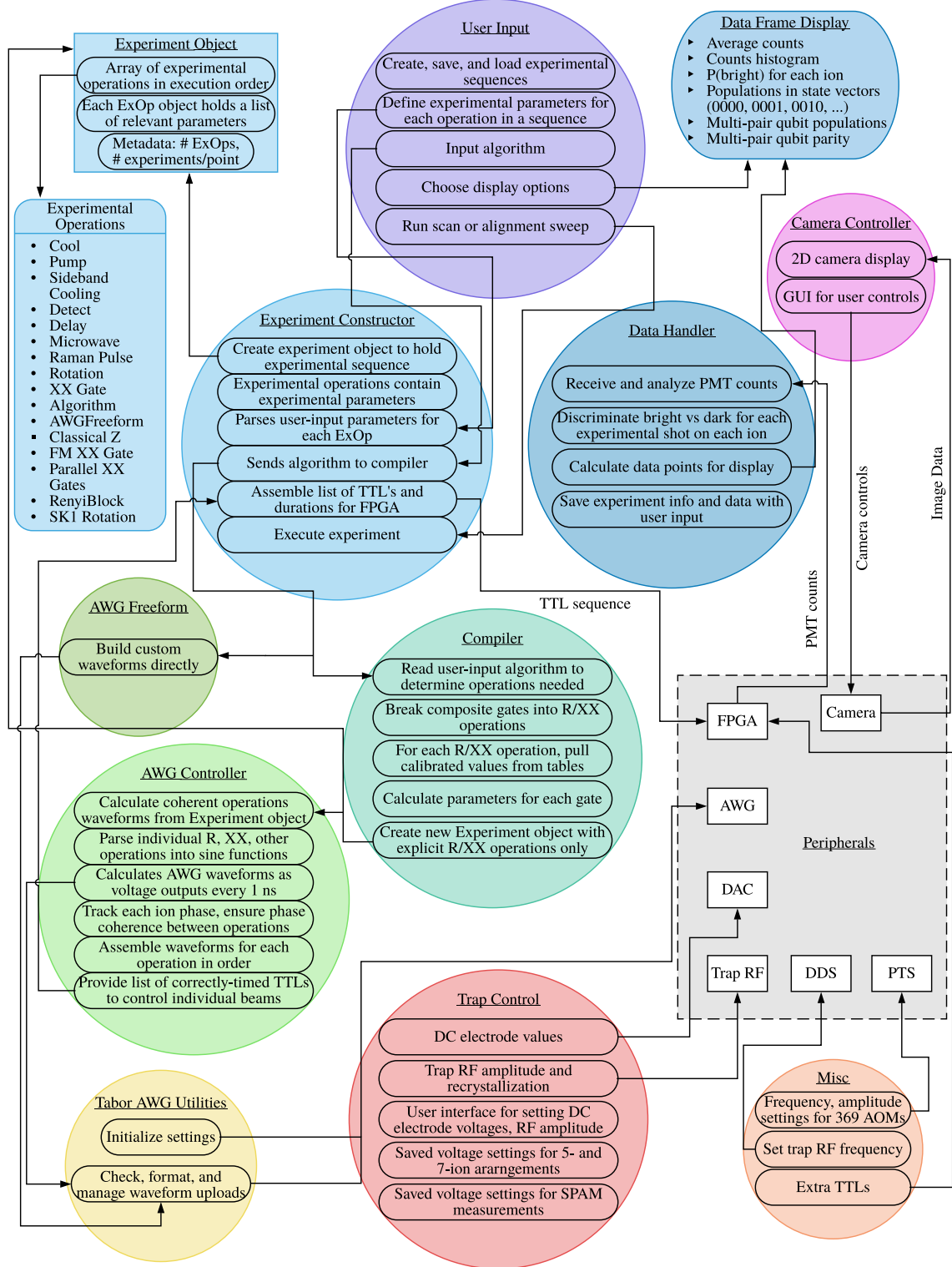


Figure 4.3: The Igor control system.

is shown in Figure 4.3. The experiment constructor file handles experimental parameters and execution. The data handler handles data analysis and display. The compiler consists of the algorithm constructor (interpreter) and the AWG controller (see Section 4.3). The pulse GUI provides the user interface for creating and executing experimental procedures. The initialization module creates GUIs for the user to control various peripheral settings, including the PTS settings controlling the cool, pump, and detect AOM drive signals. The trap voltage control provides a GUI interface for modifying the trap potential with the DC and RF electrodes, as well as some utility functions for loading saved voltage sets and recrystallizing the ion chain. Additional modules control the camera peripheral and the DDS frequency source.

The control program manages experimental procedures through the use of an experiment object. The experiment object stores an array of experimental operation objects in the order in which they should be executed. An example such sequence is shown in Figure 4.4. The experiment also has two metadata variables related to the experiment: the number of experimental operations to be executed, and the number of experimental shots to be performed for each data point. Each experimental operation object consists of an operation definition and a list of parameters relevant to that particular definition. For example, the Cool experimental operation has a parameter list that consists of the duration of the cooling to be applied. A Rotation experimental operation has a list of 3 parameters: θ , ϕ , and the ion channel(s) specifying the ions to be rotated. The experimental operation also stores variables used for scanning the experimental operation parameters in a scanning experiment.

A full list of experimental operations is given in Figure 4.3.

The experiment constructor module creates an experiment object when the user initiates an experiment with the operation sequence and parameters specified by the user. The program parses through each experimental operation in the experiment to construct a list of TTL names (pointing to one or more specific TTLs that correspond to specific operations) and a corresponding list of TTL durations (corresponding to how long that operation should be run.) The list of TTL's and durations is handed to the FPGA for execution during the experiment. If there is an algorithm chapter, it is handed to the compiler. The experiment constructor module also manages the experimental run, tracks and increments scan variables, and listens for a user command to stop the experiment execution.

One limitation to this experiment object architecture is that Igor limits arrays of objects to 100 items. Consequently, we cannot execute experiments with more than 100 experimental operations. For most experiments performed over the course of this thesis work, gate sequences were short enough that this was not a problem. However, the measurement of Renyi entropy performed in [30] required multiple repetitions of a circuit block of 20 operations, and for longer sequences, the 100 operation limit was exceeded. To overcome this obstacle, the control was moved down the stack. Instead of sending the compiler a set of 20 R and XX gates per repetition, a new experimental operation was created at the control level. The Renyi Block experimental operation executes an entire circuit, consisting of several R and XX gates, as a single experimental operation. To do so, the AWG Controller calculates a very long waveform, comprised of multiple sections of what would nor-

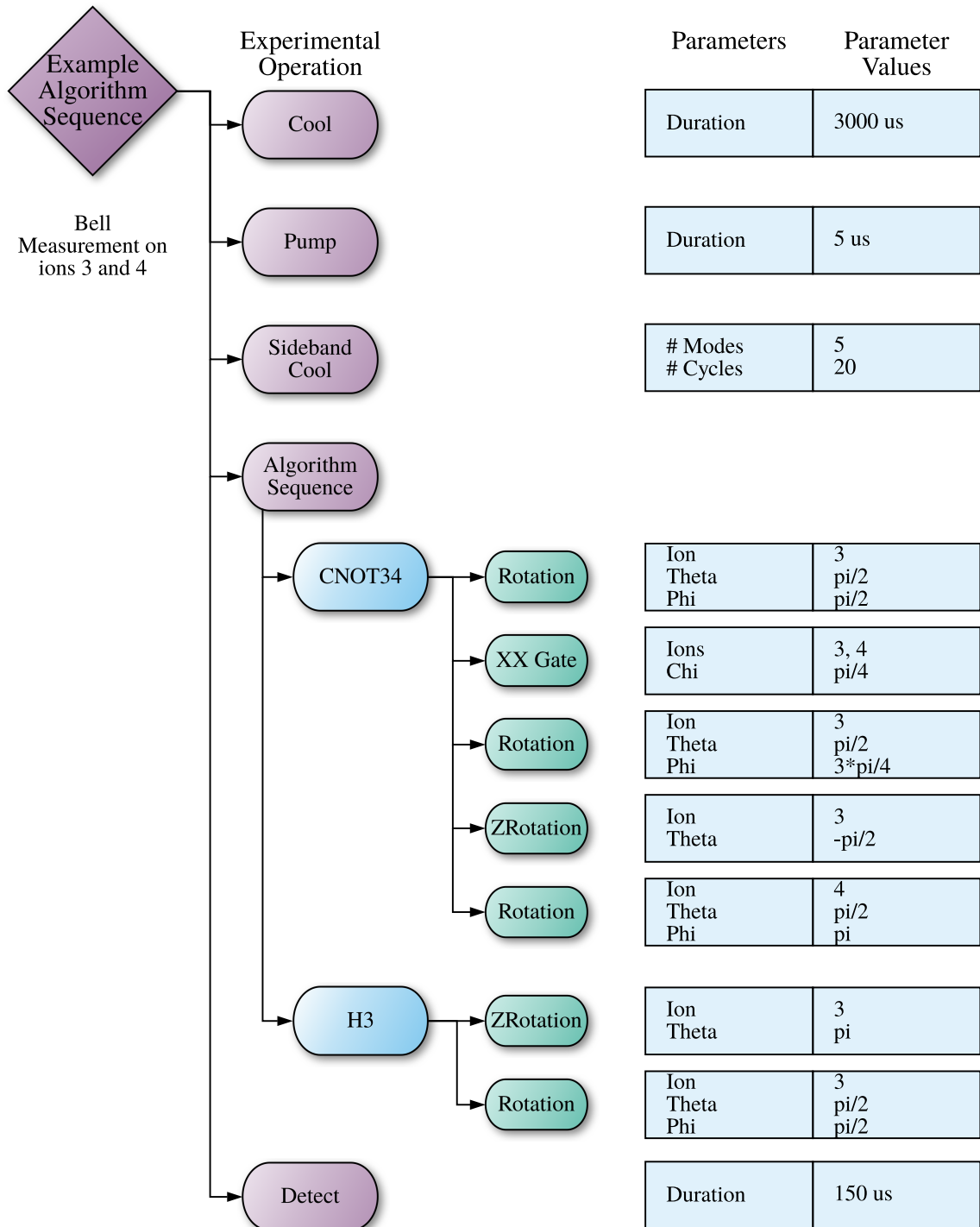


Figure 4.4: An example sequence to be performed on the experiment. This will implement a Bell measurement on qubits 3 and 4.

mally be standalone R and XX gate waveforms. Thus, many repetitions of the gate sequence became possible, allowing data collection to proceed.

The data handler receives photon counts from the PMT via the FPGA and analyzes the contents for display in the data frame. It uses a discriminator on each PMT channel to determine which ions in the chain are bright and dark for each experimental shot. This data is then aggregated to calculate bright and dark populations on each ion. Before running the experiment, the user selects what analyses will be displayed in the 4 panels of the data frame from several options. Per-ion analysis options include the average photon counts seen for each ion and probability each ion is bright. A histogram of the photon counts seen for each ion is also displayed as a monitor; deviations from the usual histogram distribution indicate something is wrong in the ion with the ion control. For multi-qubit analysis, the data is aggregated into a state vector for each data point that contains the population in each of the possible multi-qubit states. For example, the user can specify that they want to see populations in the 4-qubit state vector for ions 1, 3, 4, and 5. The data frame will then display a plot in the data frame showing the populations in $(0000, 0001, 0010, \dots)$ for each data point. Another option analyzes the populations in 2-qubit state vectors for each sub-pair in a set of 3 or more ions. For example, if the user selects ions 1, 2, 4, and 5 for this analysis, then the populations in $(00, 01, 10, 11)$ will be displayed for each of the 6 ion pairs $(1,2)$, $(1,4)$, $(1,5)$, $(2,4)$, $(2,5)$, and $(4,5)$. This option is particularly useful for looking at the result of parallel 2-qubit operations. Finally, parity can be calculated and displayed for a pair of ions, or for multiple pairs of ions within a set of 3 or more. After an

experiment is ended, the data handler can also save data from the experiment into a series of related text files for the user, including the raw data, analyzed data files, and the experiment parameters used.

The AWG Freeform experimental operation can be used in lieu of an algorithm chapter. It consists of an editable procedure file that calculates and assembles AWG waves directly from user input. The file is provided with pre-written instructions and templates to ease programming new sequences; the user simply fills in the blanks with desired parameters. This setup allows for more flexibility and can be used to implement schemes outside the standard R/XX gate set; for example, it was used to implement phonon hopping and blockades [29]. This module may prove useful for implementing other kinds of quantum simulation experiments that would particularly benefit from our individual addressing capabilities not available on many other experiments. The module could benefit from improvements such as a user-friendly GUI control interface.

The Igor control program for ion experiments was originally developed at Georgia Tech. The program has since been significantly expanded, with many new functionalities added to create a program capable of controlling a programmable ion trap quantum computer.

4.3 Compiler

When a user provides an algorithm for the quantum computer to execute, the compiler breaks it down from a high-level gate specification into a quantum machine

language - a series of pulses that are then executed on the qubit. The high-level gate specification is provided by the user as a string of abbreviated control statements separated by colons; for example, “CN24” indicates that a *CNOT* gate should be performed with control ion 2 and target ion 4. Generic native *R* and *XX* gates can also be included. Each control statement requires the inclusion of certain gate parameters, depending on the specific operation or composite gate being called.

The interpreter of the compiler reads each control statement in turn, and then creates one or more new experimental operations that are added to the experiment object corresponding to the control statement. In our “CN24” example, the interpreter identifies that a *CNOT* gate is needed on the ions in question. It then adds to the experiment object the series of *R* and *XX* experimental operations that will execute this composite gate (see Section 5.3), specifying the required parameters for each *R/XX* gate. For rotation *R* gates, the interpreter specifies the rotation angle θ , the rotation axis ϕ , and the ion(s) to be rotated. For classical *Z* rotations, the interpreter specifies the rotation angle θ and the ion(s) to be rotated. For two-qubit entangling *XX* gates, the interpreter provides the ion pair to be entangled, and the absolute value of the entanglement parameter χ . For parallel 2-qubit *XX* gates, the interpreter specifies two pairs of ions to be entangled, and two χ values. For composite gates containing *XX* gates, the interpreter also accesses the sign of each *XX* gate’s χ parameter in the *XX* gate parameter lookup table, which may inform the parameters of *R* gates in the composite operation to ensure fully modular operations. How this is handled for specific gates is further discussed in Chapter 5.

With the algorithm section of the experiment object consisting entirely of *R*

and XX native gates, the experiment is handed off to the AWG controller, which compiles the waveforms needed to implement the gates. For each gate, the AWG controller calculates a waveform for every AWG control channel; channels not used are simply set to 0 V for the duration. For channels in use, the AWG controller accesses calibration data from lookup tables to correctly calculate the needed waveform using the parameters specified in the experimental operation. For R gates, the program looks up the π time(s) on the ion(s) in question to calculate the necessary pulse duration to execute a rotation angle θ , and then calculates a sine wave of the specified duration with phase set by the rotation angle ϕ and the phase tracked by the phase counter. For XX gates, the program calls information about the pulse shape specific to the ion pair in question from the lookup table, including the gate time, sideband detuning, and the normalized pulse shape. It also accesses the calibrated gate power scaling factor, which scales the overall power of the gate pulse shape to ensure the correct value of χ is imparted to the ions. The AWG controller then calculates a bichromatic signal using these parameters, changing the amplitude over the course of the gate in accordance with the pulse shape. Two sine waves are calculated, red and blue of the carrier by the specified gate detuning, and then summed, with the phase set by the phase counter. Parallel two-qubit XX gate parameters are similar to those of standard XX gates, but with two ion pairs, pulse shapes, calibrated gate powers, and χ values. The AWG controller calculates 2 separate bichromatic waveforms, one applied to each ion pair.

A phase counting scheme serves to track the individual phase of each ion as it evolves over the course of the experiment. During an operation, an ion experiences

a small Stark shift due to the applied laser field, accumulating phase at a slightly different rate than in the dark. This phase accumulation must be carefully tracked for phase-coherent operations. For each operation performed, the program calculates how much phase each ion accumulates and adds it to a table of running phase counters, which are then used to set the phase of the signal for the next operation. Classical Z rotations are performed by adding a phase of θ to this counter for the ion(s) undergoing the rotation.

The calculated waveforms consist of a sequence of values corresponding to plotting a point of the waveform function every 1 ns. The sine waves are initially calculated for a normalized maximum amplitude of 1. The waveforms for each channel are then formatted to voltage values by the AWG utility functions; the Chase and Tabor AWGs have different formattings, so the initial waves are calculated in a normalized manner first and then formatted separately for ease of modularity.

A set of utility functions written to control the 4-channel Tabor AWG using the NI-VISA communication protocol initialize the instrument's settings. It also provides functionality to check that a calculated waveform is within uploadable parameters, format it for the AWG, and manage the upload to each channel over a USB connection. The Chase AWG is directly connected to the control computer through a PCI connection, and requires a much more limited set of utility functions to serve a similar purpose. Once uploaded to the AWG channels, a TTL signal from the FPGA triggers all channels to start outputting their waveforms at the same time. The waveforms act as RF controls for the individual channels of the 32-channel AOM, which sends deflected beams onto their corresponding ions for

coherent control.

4.4 Outlook

The control program described here has proved very capable at executing small quantum algorithms with short control sequences on a few qubits. However, scaling up will present challenges. The most immediate need is to automate calibration functions. Currently, experimental operators spend a significant amount of time calibrating the π times for R gates and the scaling power needed for XX gates by hand as the Rabi frequency of the individual controllers drifts on a scale of hours. Automating these procedures will speed the process significantly, as well as reducing the burden on experimental operators. Ongoing efforts include implementing automated calibration, as well as automated loading of 5 or 9 ions. This will be necessary for more ambitious projects, such as applying machine learning techniques to build quantum circuits using a method that will require many repetitions of a feedback loop that provides experimental data to a machine learning module [69].

Implementing longer gate sequences will require several structural changes to the control systems. As discussed in Section 4.2, the experiment object in Igor is limited in the number of experimental operations it can perform as it is currently designed. While workarounds can be implemented on an ad hoc basis², a structural redesign allowing for unlimited experimental operations will be needed. A better solution, however, will be to switch to a broader, more robust control software scheme altogether, such as those provided with ARTIQ [67] or Sandia’s

²This is sufficient for an experiment in an academic lab, right up until it isn’t.

pyIonControl software [68], which use a Python, a more well-known and versatile programming language. They also have the benefit of being an open-source project with contributions from other users who can help build new functionality and fix bugs. Other platforms may also prove to be good candidates to build such a control system.³ On the hardware side, using the AWG for control waveforms is not scalable not only in terms of cost, but also in terms of time. Pre-calculating and uploading waveforms to the AWG is the slowest part of the experimental sequence; a sequence with 10-20 XX gates may take tens of seconds to calculate and upload. Hardware upgrades such as those provided in the Sinara project [54], which calculates and outputs waveforms on the fly, will obviate this problem.

Another challenge will be managing the growth in the number of two-qubit gate connections. For a fully-connected system with N qubits, the number of two-qubit gates is

$$\binom{N}{2} = \frac{N!}{2!(N-2)!} = \frac{N^2 - N}{2} \sim O(N^2), \quad (4.1)$$

yielding a quadratic growth rate in the number of connections. While this is still polynomial (and not exponential), and larger ion traps will likely be missing some connections, it still presents a challenge. Our current strategy of using lookup tables to supply gate parameters may prove slow in the long term; more nimble database architectures may be needed. Additionally, this presents significant growth in the overhead required to set up all available gate connections by pre-calculating pulse shapes for each ion pairing and then selecting one that performs well in practice.

³Just do yourself a favor and don't use Labview.

While this is indeed overhead, and does not slow down operational execution once gates are set up, the growing overhead will present costs for those needing to iterate through many machines for development purposes. Finding methods to automatize gate solution selection may be helpful here. Additionally, several of our two-qubit gates use the same pulse shape and/or detuning; characterizing why and how this happens may facilitate finding experimentally optimal solutions.

Although we already have quite a few composite operations in our gate library (see Chapter 5), further expanding the library will be important for providing pre-optimized subroutines for larger and larger algorithms. Also useful will be a user-friendly interface for defining new additions to the gate library; this currently must be done by writing new code in an Igor procedure file. Improvements to the overall user-friendliness and robustness of a quantum computing control program will be important for allowing more users to operate the machine without having to know either how to program changes to the control software or how to avoid the little quirks and bugs always present in ad hoc software with no quality control process. Integrating a universal language for gate entry that functions across multiple platforms - just as programming languages like Python and Java can be used on many hardwares and operating systems in the classical world - will be crucial for collaboration as quantum computers scale up. Finally, integrating software that can automatically optimize new user-entered gate sequences before implementing it on the hardware will be incredibly useful. Efforts such as Microsoft's LIQUi| > [70] and Rigetti's Forest [71] have started to address these needs.

While the control program described here certainly has limitations, and more

streamlined approaches will be needed as the size of quantum computers grows, this can certainly serve as a demonstration that, moving forward, quantum computer control software will require modularity, flexible construction, and well-thought-out program design.

Chapter 5: Gate Library

Now that we have a programmable quantum computer with high-quality qubits (Chapter 2) and coherent single- and two-qubit operations yielding a universal gate set (Chapter 3), the next step is to build a gate library. These gates are used in Chapters 6-7 and [30, 31] to construct complex, multi-qubit algorithms, and the modular nature of these gates will be crucial to flexible ion-qubit mappings and allow great ease of implementation. In this chapter, I present detailed circuit diagrams for all of the operations used in the course of the research for this thesis, shown in terms of the $R(\theta, \phi)$ and $XX(\chi)$ gates directly implemented by the experiment.

5.1 Single-Qubit Gates

From Chapter 3, the single-qubit rotation is defined as

$$R(\theta, \phi) = \begin{pmatrix} \cos \frac{\theta}{2} & -ie^{-i\phi} \sin \frac{\theta}{2} \\ -ie^{i\phi} \sin \frac{\theta}{2} & \cos \frac{\theta}{2} \end{pmatrix}. \quad (5.1)$$

Rotations about the X -axis ($R_x(\theta)$) are achieved by setting $\phi = 0$, and rotations about the Y -axis ($R_y(\theta)$) are achieved by setting $\phi = \frac{\pi}{2}$. Rotations about the Z axis ($R_z(\theta)$) can be constructed from three rotations about axes in the XY plane,

(a)

$$|q\rangle \text{---} \boxed{R_z(\theta)} \text{---} = \text{---} \boxed{R_y(\frac{\pi}{2})} \boxed{R_x(\theta)} \boxed{R_y(-\frac{\pi}{2})} \text{---}$$

(b)

$$|q\rangle \text{---} \boxed{H} \text{---} = \text{---} \boxed{R_y(\frac{\pi}{2})} \boxed{R_x(\pi)} \text{---} = \text{---} \boxed{R_z(\pi)} \boxed{R_y(\frac{\pi}{2})} \text{---}$$

Figure 5.1: (a) $R_z(\theta)$ gate implementation using $R_x(\theta)$ and $R_y(\theta)$ gates. (b) Hadamard (H) gate implementations using $R_x(\theta)$, $R_y(\theta)$, and $R_z(\theta)$ gates.

as demonstrated in Figure 5.1(a).

A Hadamard (H) gate,

$$H = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}, \quad (5.2)$$

can be performed with 2 rotations, as shown in Figure 5.1(b), with a few implementation options. The fidelities of these composite rotations gates are comparable to the fidelities of individual rotations, typically 98-99%.

5.2 Multi-Qubit Circuit Construction

From Chapter 3, the two-qubit XX entangling gate is

$$XX(\chi) = \begin{pmatrix} \cos(\chi) & 0 & 0 & -i \sin(\chi) \\ 0 & \cos(\chi) & -i \sin(\chi) & 0 \\ 0 & -i \sin(\chi) & \cos(\chi) & 0 \\ -i \sin(\chi) & 0 & 0 & \cos(\chi) \end{pmatrix}. \quad (5.3)$$

The parameter χ can be varied continuously, $0 < \chi \leq \frac{\pi}{4}$, by adjusting the overall power applied to the gate, but the gates used here require only $\chi = \pm \frac{\pi}{4}$ or $\chi =$

$\pm \frac{\pi}{8}$. The ability to vary χ continuously has proved useful in implementing circuits for other applications not discussed in depth here, including the quantum Fourier transform [19] and a Bayesian quantum game [32]. The gate is maximally entangling for $\chi = \pm \frac{\pi}{4}$.

Two-qubit XX gates are combined with rotation R gates to construct composite gates. The parameter χ can be positive or negative, depending on what ion pair is chosen and the particulars of the pulse segmentation solution chosen for the ion pair in question; the sign of χ ($\text{sgn}(\chi)$) is determined experimentally for each ion pair. Consequently, some composite gate circuits include rotations with parameters that depend on $\text{sgn}(\chi)$; all composite gates are constructed to account for both signs of χ for each XX gate used, so that composite gates are always fully modular.

Composite gates are constructed by starting with known circuits, converting constituent parts into R and XX gates using lower-level constructions, and then optimizing the circuit. First, the number of XX gates is minimized (as in the Toffoli-3 gate, described in Section 5.4). Second, the single-qubit gates are optimized by minimizing the sum of all rotation angles θ , as this minimizes the total time for the experiment.

Several composite circuits used in algorithms performed in this thesis are presented in this chapter. Other quantum algorithms may be implemented on this system in a similar fashion. First, decompose the algorithms subroutines into high-level circuits. Second, optimize those circuits to minimize the number of two-qubit interactions required. Third, decompose the high-level circuits into physical-level R and XX gates. Finally, perform further optimizations to first minimize the num-

ber of two-qubit XX gates required, and then to minimize the total rotation time (the sum of all rotation angles θ) across all R gates. However, the optimization of quantum circuits is in the QMA-Hard complexity class, which is the quantum analog of the classical NP-Hard complexity class. We therefore anticipate that future improvements in algorithm design, circuit synthesis, and circuit optimization techniques may result in more efficient circuit implementations, facilitating increased experimental performance.

5.2.1 Optimization Strategy Adjustments with Experimental Upgrades

This chapter features composite gates developed for use before and after the improvements described in Section 2.4 were implemented, and consequently are optimized slightly differently. In particular, the introduction of individual ion phase control allowing for the purely classical implementation of Z rotations (see Section 3.1.2) leads to different optimization strategies for minimizing rotations. Pre-upgrade, any Z rotations had to be implemented via a combination of several rotations in the XY plane. Consequently, composite gates were optimized by minimizing the total rotation angle θ for all rotations $R(\theta, \phi)$ in the XY plane; Z rotations were to be avoided where possible, and compiled into X and Y rotations for further optimization when not. Post-upgrade, however, Z rotations became effectively error-free to perform relative to $R(\theta, \phi)$ rotations. Therefore, when optimizing a given composite gate, the total rotation angle θ over all $R(\theta, \phi)$ rotations can be further minimized by

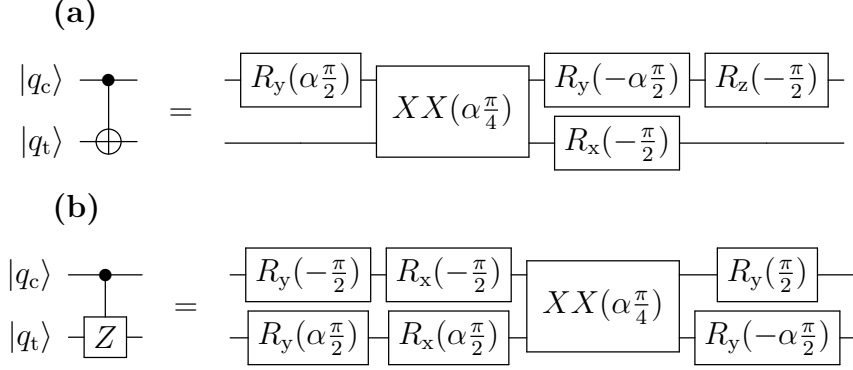


Figure 5.2: χ_{ct} is the parameter for the XX gate between the two qubits. Let $\alpha = \text{sgn}(\chi_{\text{ct}})$. (a) $CNOT$ gate implementation using $XX(\chi)$, $R_x(\theta)$, $R_y(\theta)$, and $R_z(\theta)$ gates. (b) Controlled- Z gate implementation using $XX(\chi)$, $R_x(\theta)$, and $R_y(\theta)$ gates.

employing Z rotations instead wherever possible. In particular, the following Pauli rotation identities were very useful for converting between X , Y , and Z rotations to minimize rotations other than Z :

$$R_z(\theta) = R_y\left(\frac{\pi}{2}\right) \cdot R_x(\theta) \cdot R_y\left(-\frac{\pi}{2}\right) = R_x\left(-\frac{\pi}{2}\right) \cdot R_y(\theta) \cdot R_x\left(\frac{\pi}{2}\right) \quad (5.4)$$

$$R_x(\theta) = R_z\left(\frac{\pi}{2}\right) \cdot R_y(\theta) \cdot R_z\left(-\frac{\pi}{2}\right) = R_y\left(-\frac{\pi}{2}\right) \cdot R_z(\theta) \cdot R_y\left(\frac{\pi}{2}\right) \quad (5.5)$$

$$R_y(\theta) = R_x\left(\frac{\pi}{2}\right) \cdot R_z(\theta) \cdot R_x\left(-\frac{\pi}{2}\right) = R_z\left(-\frac{\pi}{2}\right) \cdot R_x(\theta) \cdot R_z\left(\frac{\pi}{2}\right) \quad (5.6)$$

It is also useful to note that X rotations commute with XX gates, and can therefore be moved around more to minimize rotations. In this chapter, I will identify which composite gates were optimized under which set of circumstances.

5.3 Two-Qubit Composite Gates

The controlled- NOT , or $CNOT$, gate, is a two-qubit interaction used as the building block for many quantum algorithms. A control qubit $|q_c\rangle$ acts on a target

qubit $|q_t\rangle$ by flipping the target qubit's state if the control qubit is in the $|1\rangle$ state; otherwise, it does nothing. This has the unitary evolution operator

$$U_{CNOT} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix}. \quad (5.7)$$

Another useful two-qubit interaction is the controlled- Z , or $C(Z)$, gate, which flips the phase of the two-qubit state if the two qubits are in the $|11\rangle$ state:

$$U_{C(Z)} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & -1 \end{pmatrix}. \quad (5.8)$$

Circuits for the $CNOT$ and $C(Z)$ gates are shown in Figures 5.2(a-b) respectively. They each require one $XX\left(\frac{\pi}{4}\right)$ gate and several rotations. Both gates were first demonstrated on this system in [19], with typical fidelities of 96-98% [19, 20]. The upgrades described in Section 2.4 have resulted in typical $CNOT$ gate fidelities increasing to 98-99%. In this work, the $CNOT$ gate will be used in Chapters 6 and 7, while the controlled- Z gate will only be needed for Chapter 6.

5.4 Toffoli Gates

Next, we scale up to a three-qubit composite gate by implementing a controlled-controlled-NOT ($C^2(NOT)$), or Toffoli-3, gate. The Toffoli gate [72] requires two control qubits (q_1 and q_2) and one target qubit (q_t). Like the $CNOT$ gate, it flips

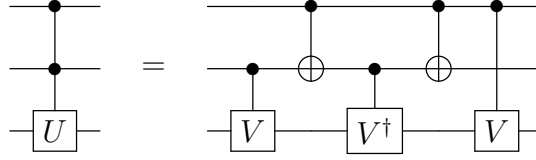


Figure 5.3: Any doubly-controlled unitary U can be constructed out of 5 two-qubit interactions using $CNOT$ gates and controlled- V operations, where $V^2 = U$ [63].

the target qubit state if and only if both control qubits are in the $|1\rangle$ state:

$$U_{T3} = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{pmatrix}. \quad (5.9)$$

Our Toffoli-3 gate is constructed from 5 two-qubit gates (three $XX\left(\frac{\pi}{8}\right)$ and two $XX\left(\frac{\pi}{4}\right)$ gates) in a manner similar to the Toffoli gate demonstrated in [73]. As shown in Figure 5.3, any doubly-controlled unitary $C^2(U)$ operation can be performed with 5 two-qubit interactions (two $CNOT$ s, two $C(V)$ s, and one $C(V^\dagger)$) if a controlled- V operation is available such that $V^2 = U$ [63]. Since

$$\left[XX\left(\frac{\pi}{8}\right)\right]^2 = XX\left(\frac{\pi}{4}\right), \quad (5.10)$$

we can add single-qubit rotations to construct a Toffoli-3 gate with minimal use of two-qubit gates, as shown in Figure 5.4(a). This compares favorably to the 6 two-qubit gates that would be necessary if only $CNOT$ (or equivalently, $XX\left(\frac{\pi}{4}\right)$) gates were available. These constructions also provide for the implementation of

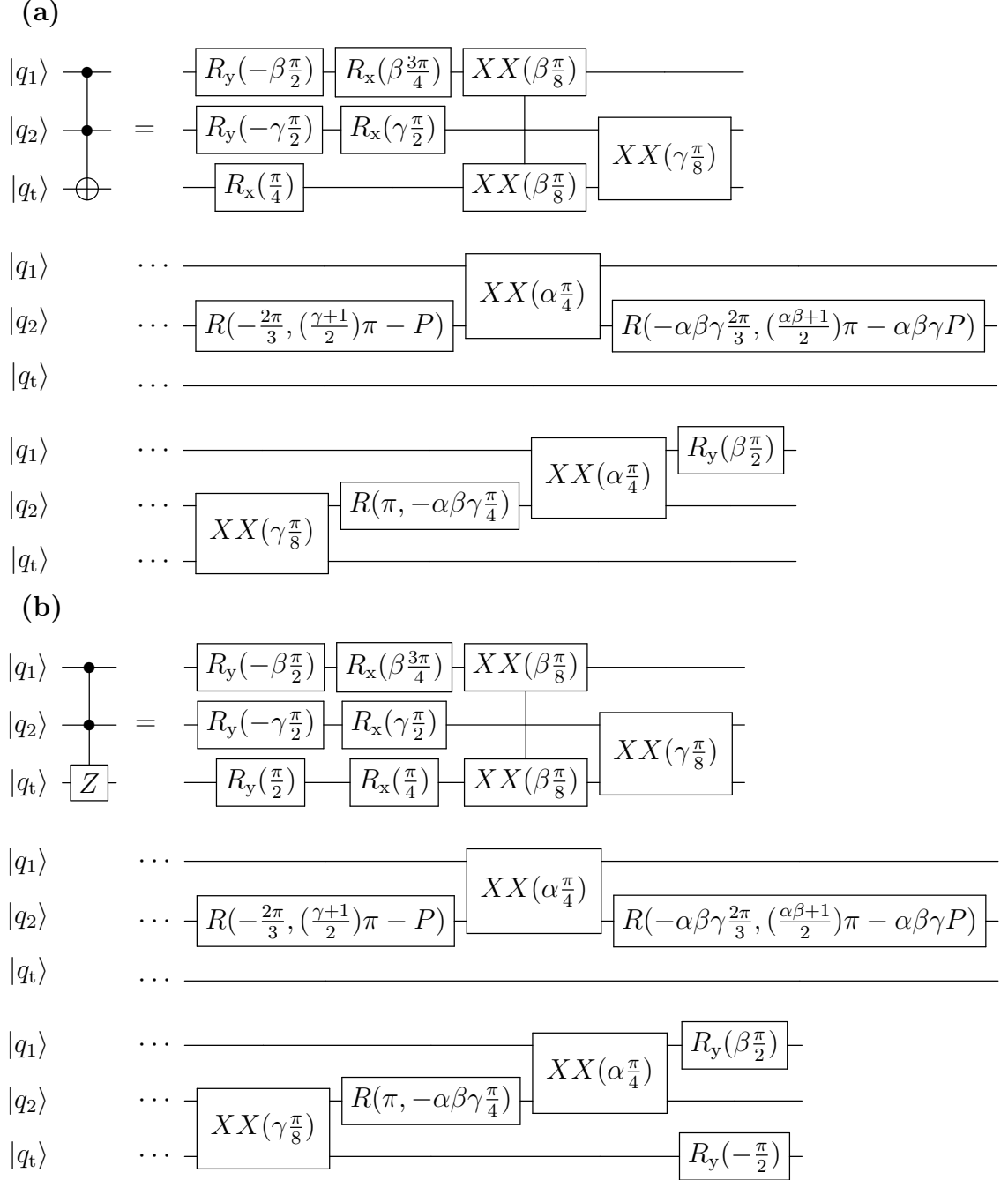


Figure 5.4: Three-qubit composite gates using $XX(\chi)$, $R_x(\theta)$, $R_y(\theta)$, and $R(\theta, \phi)$ gates. Let $\alpha = \text{sgn}(\chi_{12})$, $\beta = \text{sgn}(\chi_{1t})$, $\gamma = \text{sgn}(\chi_{2t})$, and $P = \arcsin\left(\sqrt{\frac{2}{3}}\right)$. (a) Toffoli-3 gate implementation. (b) Controlled-controlled-Z (CCZ) gate implementation.

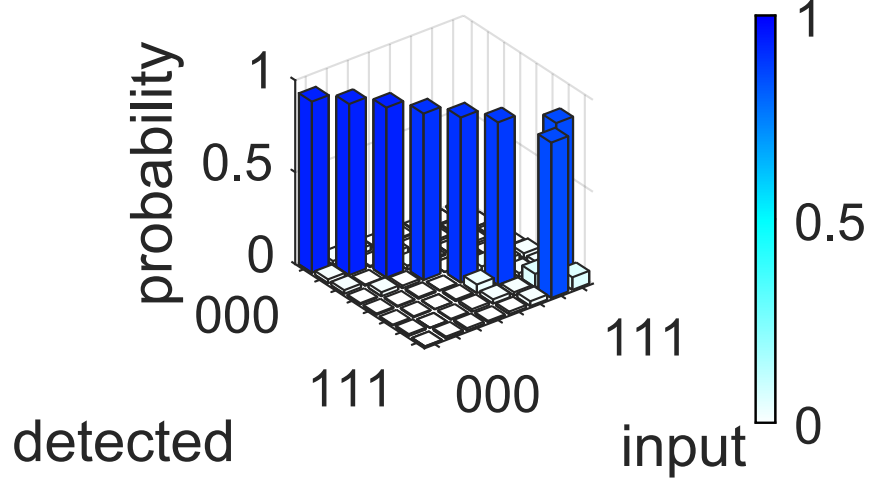


Figure 5.5: Measured truth table for a Toffoli-3 gate. The average process fidelity is 89.6(2)%, corrected for a 1.5% average state preparation and measurement (SPAM) error.

$C^2(Z)$ gates, which can be constructed by adding a few single-qubit rotations to a Toffoli-3 gate (see Figure 5.4(b)) and has the unitary evolution operator

$$U_{C^2(Z)} = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 \end{pmatrix}. \quad (5.11)$$

For all circuits, the single-qubit rotations are further optimized to minimize total rotation time [74], optimized under pre-upgrade constraints.

Here, we show results for a Toffoli-3 gate, with a process fidelity of 89.6(2)% (see Figure 5.5). Toffoli-3 gates have been previously performed in NMR systems [75] and ion traps [76], including this system [24,27]. We employed a limited tomography procedure to verify that the Toffoli-3 gate performed had no spurious phases on the

outputs (see Section 5.4.1). This gate will be used for the Grover search algorithm in Chapter 6.

5.4.1 Toffoli-3 Characterization

We employed a limited tomography procedure to characterize the outputs of the Toffoli-3 gate performed. A global rotation into the X basis was applied to all 3 ions before and after the Toffoli-3 gate for each input (see Figure 5.6(a)): $R_y(\frac{\pi}{2})$ for the even inputs (000, 010, 100, 110) and $R_y(-\frac{\pi}{2})$ for the odd inputs (001, 011, 101, 111). An ideal Toffoli-3 gate will result in an anti-diagonal input-output matrix in the Z basis when this procedure is applied. The experimental results of this verification procedure are shown in Figure 5.6(b) with an average success probability of 82.1(2)%, indicating the Toffoli-3 is faithful for arbitrary input states.

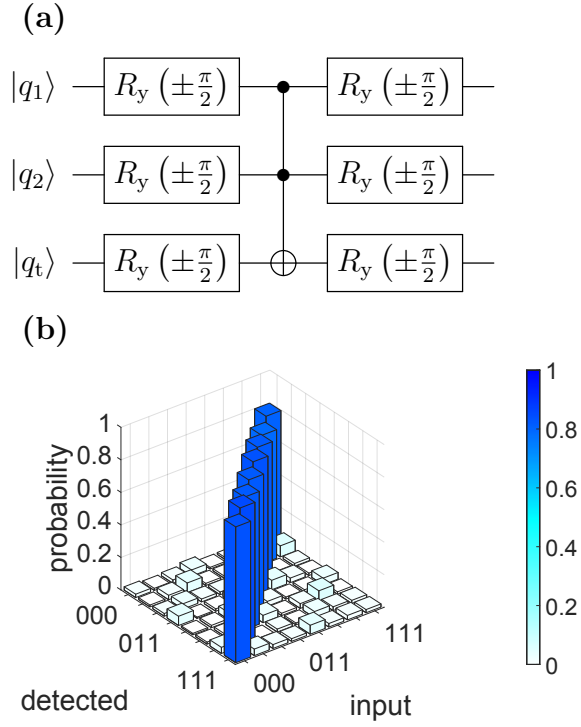
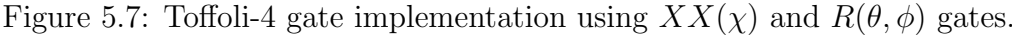


Figure 5.6: (a) Circuit for implementing the Toffoli-3 limited tomography procedure. The global rotations are positive for even input states and negative for odd input states. (b) Limited tomography check performed on the Toffoli-3 gate to verify phases. The average success probability is 82.1(2)%, corrected for a 2.4% average SPAM error.



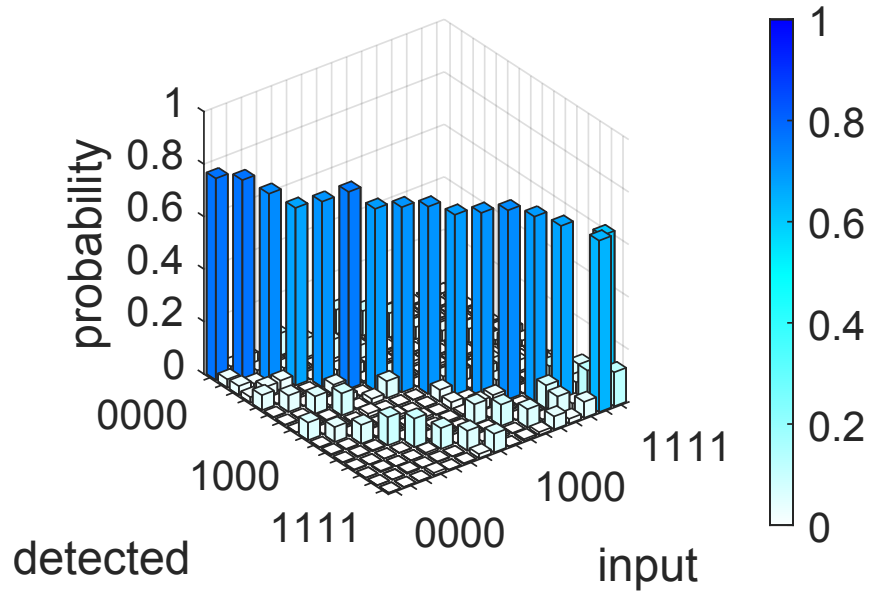


Figure 5.8: Measured truth table for a Toffoli-4 gate performed with 3 controls, 1 target, and 1 ancilla qubit. The average process fidelity is 70.5(3)%, corrected for a 1.9% average SPAM error.

5.4.2 Toffoli-4 Gate

We use a related strategy to construct a Toffoli-4 gate as was done for the Toffoli-3 gate, where the Toffoli-4 gate is a triply-controlled *NOT* gate,

$$U_{T4} = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}. \quad (5.12)$$

Using the methods described in [77], we construct a circuit between 3 control qubits (q_1 , q_2 , and q_3), one target qubit (q_t), and an additional ancilla qubit (q_a), shown in Figure 5.7. In the circuit diagram, $\alpha_1 = \text{sgn}(\chi_{1a})$, $\alpha_2 = \text{sgn}(\chi_{2a})$, $\alpha_3 = \text{sgn}(\chi_{3a})$, $\alpha_t = \text{sgn}(\chi_{ta})$, $\beta = \text{sgn}(\chi_{3t})$, $P = \arcsin\left(\sqrt{\frac{2}{3}}\right)$, and $Q = \frac{1}{8}(4 - 3\alpha_2\alpha_t)$. By again using both $XX\left(\frac{\pi}{4}\right)$ and $XX\left(\frac{\pi}{8}\right)$ gates, we are able to save one two-qubit gate relative to a construction limited to *CNOT* gates [77]. When executing the gate on the system, we report an average process fidelity of 70.5(3)% (see Figure 5.8) [24].

5.5 The *SWAP* Gate

A *SWAP* gate exchanges the quantum states of two qubits, like so:

$$U_{SWAP} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}. \quad (5.13)$$

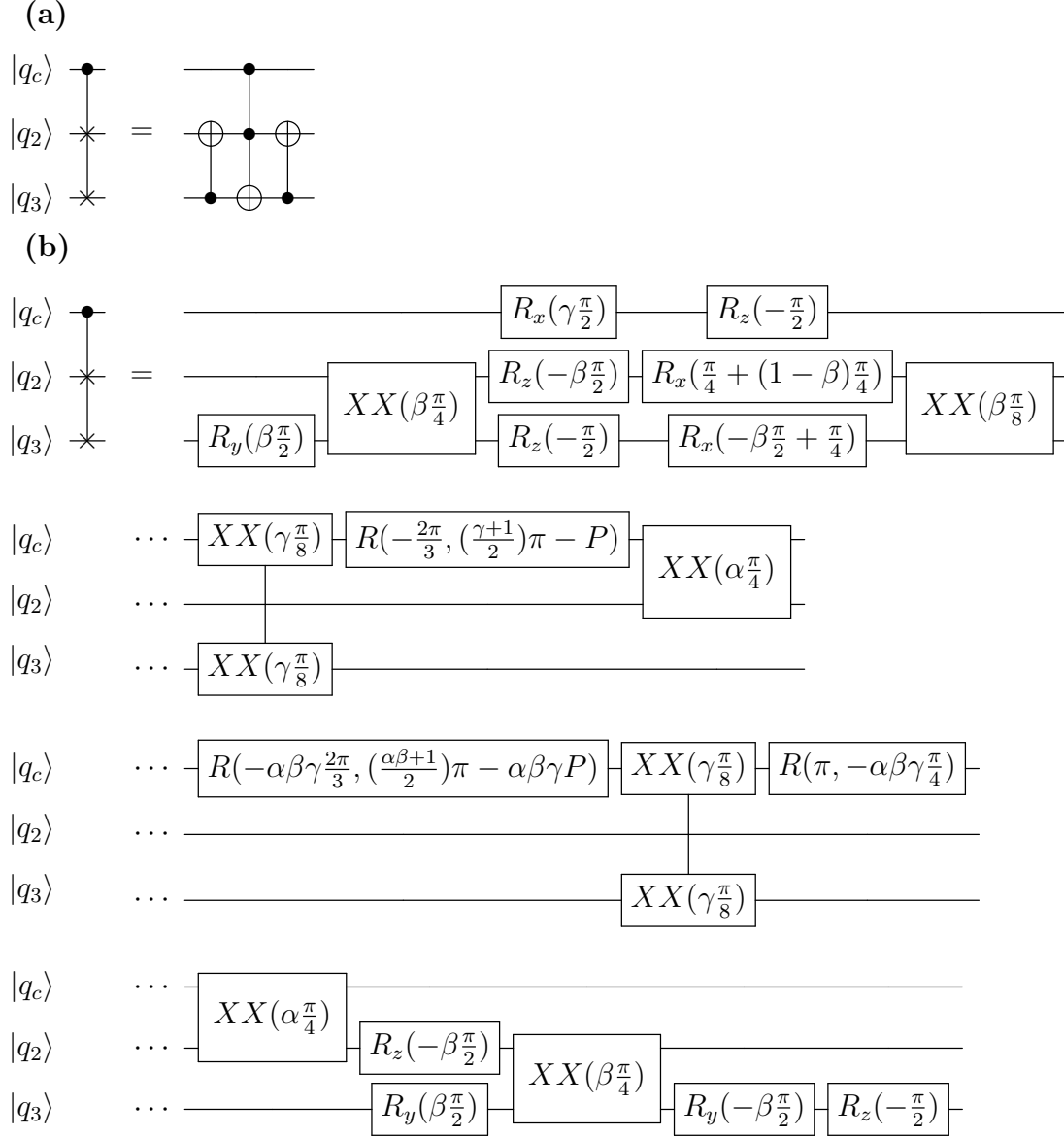
If one assumes that qubits are fixed wires, one could naively implement a *SWAP* gate as follows [65]:



$$= \text{classical swap} \quad (5.14)$$

However, if one has a fully-connected system - for example, an ion-trap quantum computer where the normal modes of motion yield multi-qubit entangling gates with all possible pairwise interactions available - then one can simply swap the qubit assignments, a purely classical operation that saves us 3 two-qubit interactions. This will come in handy in Section 5.9. A comparison to another architecture, the IBM Quantum Experience, shows that our system benefits from the absence of *SWAP*-gate overhead due to its complete connectivity graph [27].

5.6 Controlled-*SWAP* Gate



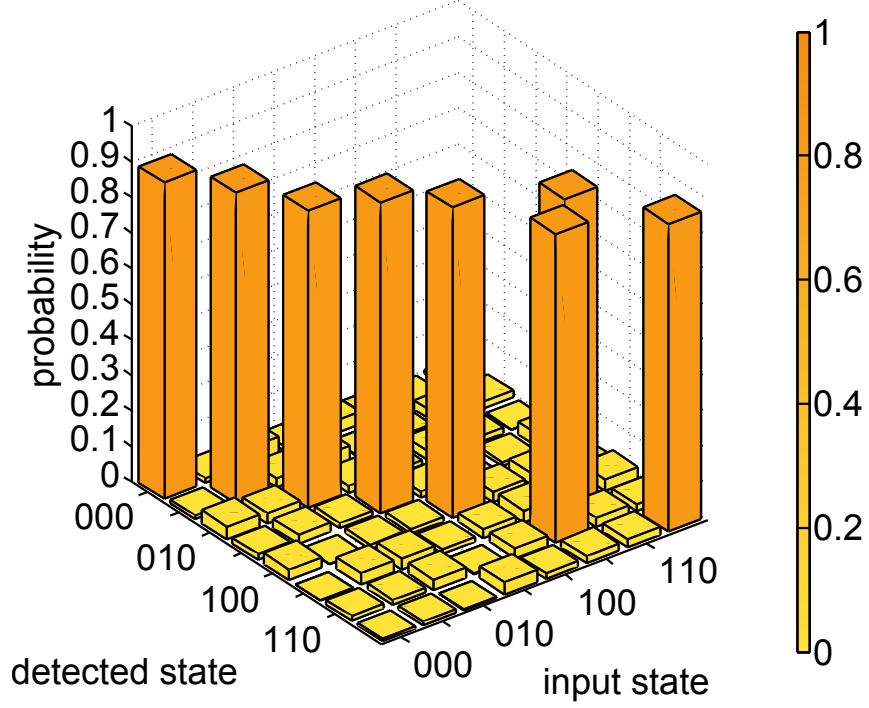


Figure 5.10: Measured truth table for a Controlled-*SWAP* gate. The average process fidelity is 86.8(3)%, corrected for a 1% average state preparation and measurement (SPAM) error.

The controlled-*SWAP* (*CSWAP*, or Fredkin) gate [78] is a three-qubit interaction that operates by swapping the last two qubits $|q_2\rangle$ and $|q_3\rangle$ if the control qubit $|q_c\rangle$ is in the state $|1\rangle$, and has the unitary evolution operator

$$U_{CSWAP} = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}. \quad (5.15)$$

It has been previously demonstrated on NMR [79] and photonic systems [80, 81]; this represents its first demonstration on a trapped ion system.

The *CSWAP* can be implemented using two *CNOT* gates and a Toffoli gate, as shown in Figure 5.10(a). The final composite circuit, shown in Figure 5.10(b), was constructed by assembling the pre-optimized *CNOT* and Toffoli circuits in Figures 5.2(a) and 5.4(a) respectively, and then further optimizing the combined rotations. The Toffoli and *CNOT* gates used here were optimized for use on the machine pre-upgrade, but the *CSWAP* was optimized for use post-upgrade; consequently, it is possible further rotation savings could be achieved were a more optimal Toffoli gate determined first. Future uses of the *CSWAP* may benefit from a more thorough optimization procedure taking advantage of the upgraded experimental capabilities.

The *CSWAP* gate was used as a readout module to measure the Renyi-2 entropy in a two-site Fermi-Hubbard model system [30]. Data for all bitwise inputs and outputs is shown in Figure 5.10, with an average process fidelity of 86.8(3)%. For use in the Renyi-2 entropy measurement scheme, only the outcome of the control qubit is needed; its process fidelity is 94.0(2)%.

5.7 Using Z -Rotations to Obviate χ Sign in XX Gates

With Z rotations implemented classically (see Section 5.2.1), an opportunity now arises to abstract away the considerations made for the effects of the sign of χ on the construction of a composite gate. Since $XX\left(\frac{\pi}{4}\right) \neq XX\left(-\frac{\pi}{4}\right)$, a mechanism is needed to ensure correct gate sequences. Previously, a parameter for the sign of the χ for each XX gate had to be introduced, and affected the properties of other rotations in the circuit in order to ensure the gate was fully modular and could be

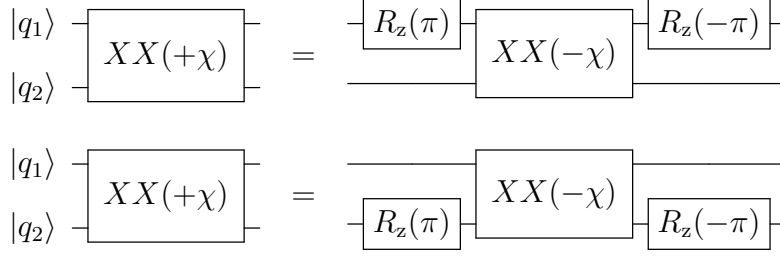


Figure 5.11: Adding an appropriate Z rotation on one qubit on either side of an $XX(\chi)$ gate flips the sign of its χ parameter, obviating the need to consider all possible combinations of χ signs when optimizing a modular, programmable gate. When implementing a gate on a given set of ions, simply add Z rotations on either side of any XX gates on ion pairs with experimentally-verified negative χ signs.

used on any qubit selection, regardless of the χ -signs of the XX gates in use. This made circuit optimization considerably more complicated, and led to some circuits (such as the Toffoli-3 in Figure 5.4(a)) that were not necessarily optimal for all possible χ sign combinations.

Conveniently, it turns out that adding a classical Z rotation on the same qubit before and after an $XX(\chi)$ gate flips the sign of its χ parameter, as shown in Figure 5.11. By adding these classically-implemented rotations whenever an XX gate with a negative χ parameter is used, all XX gates are rendered identical without an error penalty, making composite gate optimization much simpler (I optimized all of the following gate sequences by hand) and ensuring that all composite gates are always optimal.

5.8 The Square Root of CNOT

In Section 7.5, to construct the full adder circuit from [82] we will need to construct a Controlled- V ($C(V)$) gate, where $C(V)$ is the square root of a $CNOT$ gate, or $C(V)^2 = CNOT$, and consists of a controlled- $X(\theta)$ operation. As discussed

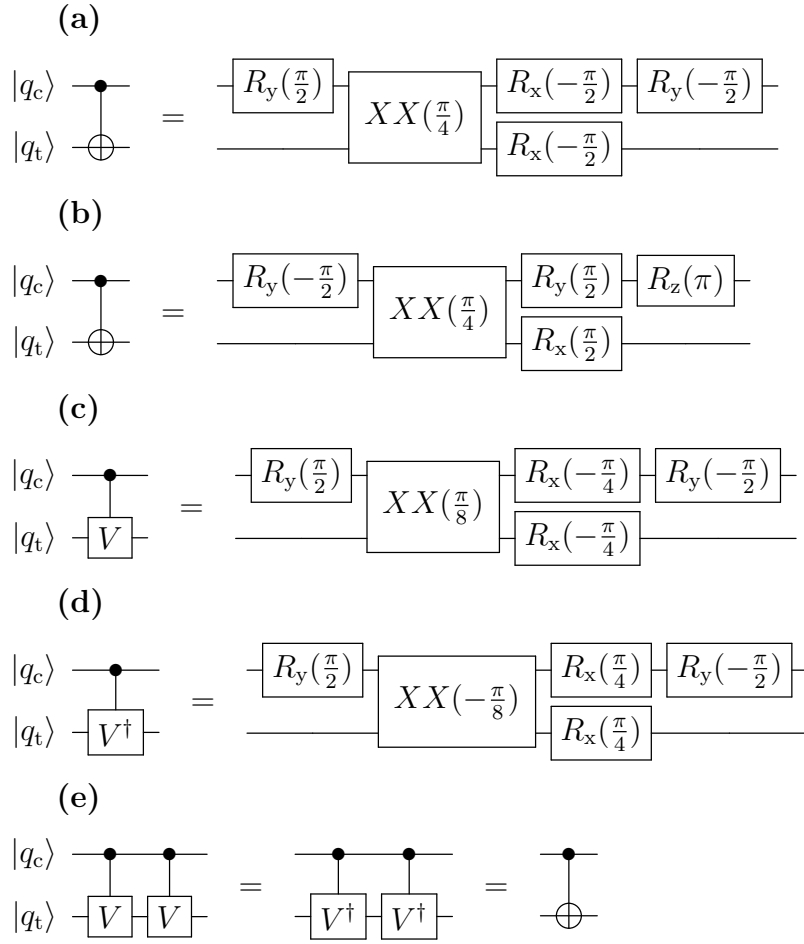


Figure 5.12: Component circuits for $CNOT$ and its square root gates using $XX(\chi)$, $R_x(\theta)$, $R_y(\theta)$, and $R_z(\theta)$ gates.. (a) and (b) Alternate $CNOT$ gate implementations. (c) Controlled- V ($C(V)$) gate implementation. (d) Controlled- V^\dagger ($C(V^\dagger)$) gate implementation. (e) $C(V)$ and $C(V^\dagger)$ gates are equivalent to the square root of a $CNOT$ gate.

in Section 5.4, we can adjust the χ parameter of the $XX(\chi)$ gate to achieve such a relationship, shown in Equation 5.10. Figure 5.12(a) shows a variation on the $CNOT$ gate from Figure 5.2(a) that allows for the creation of a \sqrt{CNOT} by adjusting the χ of the XX gate, and concurrently adjusting the length of the following X rotations on each ion; the resulting circuit for a $C(V)$ gate is shown in Figure 5.12(c). Additionally, $C(V^\dagger)$ is needed, shown in Figure 5.12(d); from Section 5.7, we can easily implement the $XX(-\frac{\pi}{8})$ gate by adding Z rotations on either side of the XX gate. By squaring the $C(V)$ or $C(V^\dagger)$ gates, a $CNOT$ gate is performed ($C(V^\dagger)$ (Figure 5.12(e)).

The unitary for the $C(V) = \sqrt{CNOT}$ gate is

$$U_V = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & \frac{1}{2}(1-i) & \frac{1}{2}(1+i) \\ 0 & 0 & \frac{1}{2}(1+i) & \frac{1}{2}(1-i) \end{pmatrix}. \quad (5.16)$$

5.9 Quantum Scrambling Library

Quantum scrambling is a phenomenon of interest that describes how information spreads through a system. It has direct implications for out of time order correlators (OTOCs) and has implications for the study of quantum information in black holes [83]. We have implemented scrambling unitaries on our experiment and probed their properties in several ways [31], as proposed in [84]. Here, I discuss the library of quantum circuits used for implementing and probing scrambling unitaries.

The general circuit diagram to test whether a unitary U is a quantum scram-

bling unitary is shown in Figure 5.13. In the case of a scrambling unitary, the input state $|\psi\rangle$ will teleport to the output state $|\varphi\rangle$ with fidelity $F = |\langle\psi|\varphi\rangle|^2 = 1$, conditioned on the outcome of the Bell measurement. The Bell measurement itself will yield the correct Bell state $|00\rangle + |11\rangle$ with post-selection probability P . Three possible pairs of qubits can be used for Bell measurements: (3,4), (2,5), and (1,6). If successful teleportation with the correct post-selection probability P for all possible input states and all possible choices of Bell pair, quantum scrambling is verified. Using the teleportation fidelity F and the post-selection probability P , we can probe the relationship between scrambling, OTOCs, and decoherence [84].

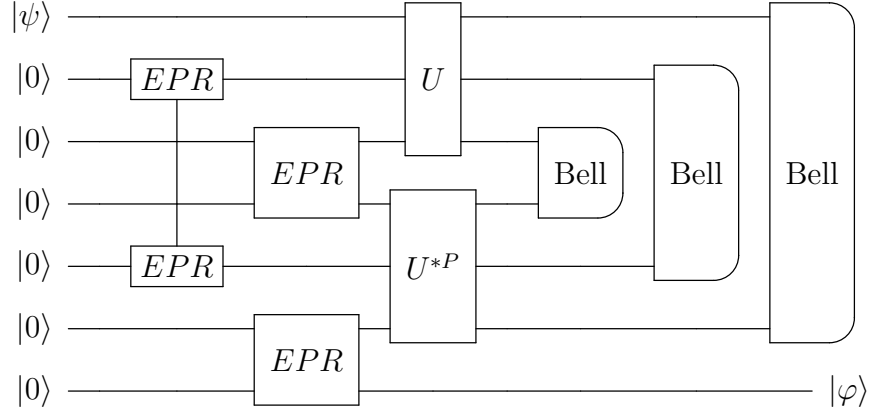


Figure 5.13: General figure for testing the scrambling properties of a unitary U .

The unitary U^{*P} is a permutation of the unitary U^* that permutes the first

and third input, such that $U^{*P} = PU^*P$, where

$$P = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}. \quad (5.17)$$

This can also be modeled by simply swapping the inputs of qubits 4 and 6 into the unitary U^* , and swapping them back afterwards.

5.9.1 Deterministic Teleportation Protocol

An alternative protocol that teleports $|\psi\rangle$ deterministically, eliminating the need to perform post-selection, can be performed using a variant of the Grover search protocol [85] and implemented as shown in Figure 5.14(a). The Grover operations, G_D and G_A , have the form

$$G = I - 2|EPR\rangle\langle EPR| = \begin{pmatrix} 0 & 0 & 0 & -1 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ -1 & 0 & 0 & 0 \end{pmatrix}, \quad (5.18)$$

which is a SWAP operation and a few rotations. The circuit is shown in Figure 5.14(b). We can implement the SWAP by simply re-assigning the qubits to the other ion; no quantum operation needed.

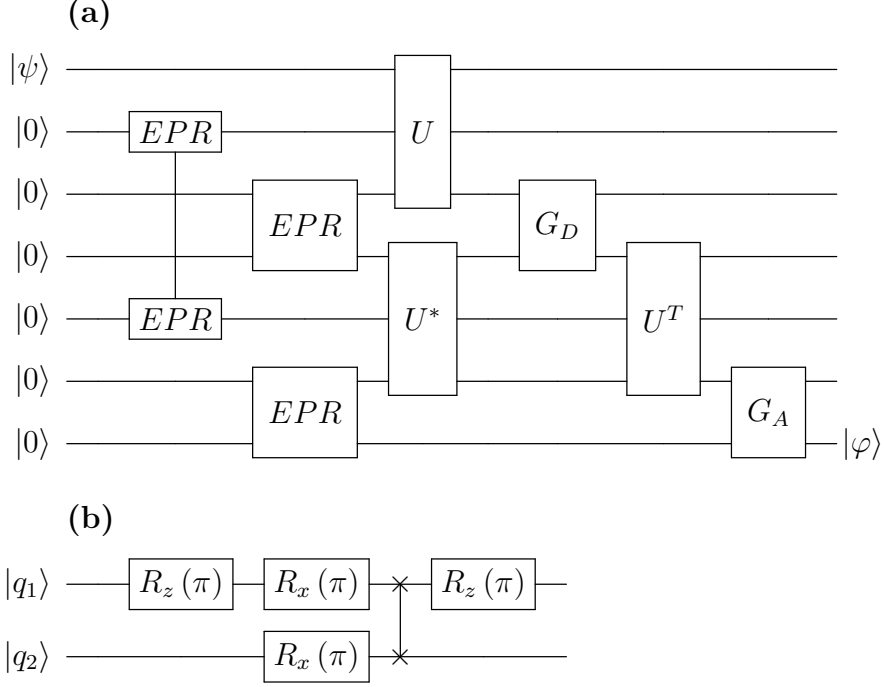


Figure 5.14: (a) General figure for testing the scrambling properties of a unitary U , using a Grover protocol requiring G . (b) Circuit for Grover protocol G .

5.9.2 Bell States and Bell Measurements

For the quantum scrambling experiments presented in [31], we will need to create and measure Bell pairs. Below are the schemes used for dealing with Bell pairs, optimized under the post-upgrades capabilities.

The four Bell states can be created using the following recipe (or related variations) as a method of constructing the EPR pairs required by the scramble-tester

circuit in Figure 5.13:

$$|\Phi^+\rangle = \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle) = R_{Z,1}\left(\alpha\frac{\pi}{2}\right) XX\left(\alpha\frac{\pi}{4}\right)|00\rangle \quad (5.19)$$

$$|\Phi^-\rangle = \frac{1}{\sqrt{2}}(|00\rangle - |11\rangle) = R_{Z,1}\left(-\alpha\frac{\pi}{2}\right) XX\left(\alpha\frac{\pi}{4}\right)|00\rangle \quad (5.20)$$

$$|\Psi^+\rangle = \frac{1}{\sqrt{2}}(|01\rangle + |10\rangle) = R_{Z,1}\left(\alpha\frac{\pi}{2}\right) R_{X,2}(\pi) XX\left(\alpha\frac{\pi}{4}\right)|00\rangle \quad (5.21)$$

$$|\Psi^-\rangle = \frac{1}{\sqrt{2}}(|01\rangle - |10\rangle) = R_{Z,1}\left(-\alpha\frac{\pi}{2}\right) R_{X,2}(\pi) XX\left(\alpha\frac{\pi}{4}\right)|00\rangle \quad (5.22)$$

Our native $XX(\chi)$ gates create the entangling state $\frac{1}{\sqrt{2}}(|00\rangle - i|11\rangle)$. As such, $XX(\frac{\pi}{4})$ gates can be used by themselves to create EPR pairs. Since the resulting output in Figure 5.13 simply adds a global phase to the teleported state $|\varphi\rangle$, the bare $XX(\frac{\pi}{4})$ gates can be used without modification in the circuit. Note that proper Bell measurement as described next is still necessary - we can't use bare XX gates there.

Bell measurements can be performed using the circuit shown in Figure 5.15(a) [65]. This takes as input a state $|q_1q_2\rangle$ in the Bell basis, so

$$|q_1q_2\rangle = a|\Phi^+\rangle + b|\Psi^+\rangle + c|\Phi^-\rangle + d|\Psi^-\rangle, \quad (5.23)$$

where $a^2 + b^2 + c^2 + d^2 = 1$, and outputs the result

$$|q_1q_2\rangle = a|00\rangle + b|01\rangle + c|10\rangle + d|11\rangle. \quad (5.24)$$

This circuit can be optimized using our experimentally available gates $R(\theta, \phi)$ and

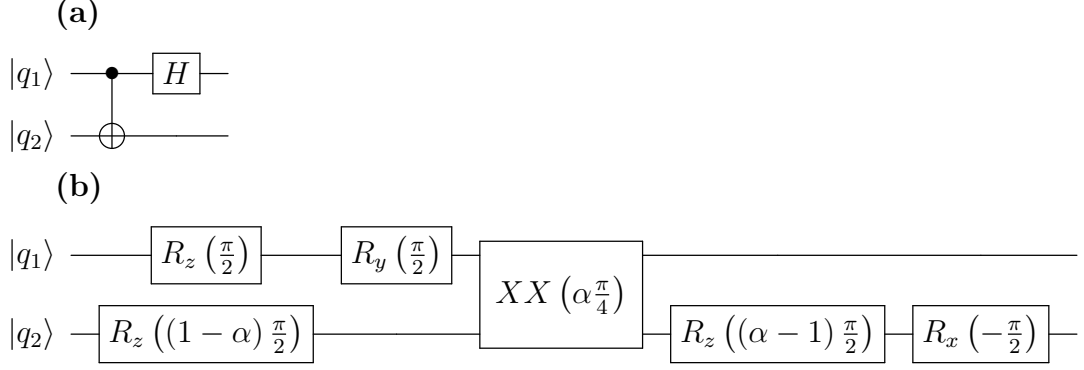


Figure 5.15: (a) Simple Bell measurement circuit implemented with a $CNOT$ gate and a Hadamard gate. (b) Optimized Bell measurement gate sequence using R and XX gates.

$XX(\chi)$ to the Bell measurement circuit shown in Figure 5.15(b).

5.9.3 Scrambling Unitary U_S

The scrambling unitary U_S used for much of the data discussed in [31],

$$U_S = \frac{1}{2} \begin{pmatrix} -1 & 0 & 0 & -1 & 0 & -1 & -1 & 0 \\ 0 & 1 & -1 & 0 & -1 & 0 & 0 & 1 \\ 0 & -1 & 1 & 0 & -1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 & 0 & -1 & -1 & 0 \\ 0 & -1 & -1 & 0 & 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & -1 & 0 & 1 & -1 & 0 \\ 1 & 0 & 0 & -1 & 0 & -1 & 1 & 0 \\ 0 & -1 & -1 & 0 & -1 & 0 & 0 & -1 \end{pmatrix}, \quad (5.25)$$

is verified to be a scrambling unitary by showing

$$\begin{aligned}
U^\dagger(X \otimes I \otimes I)U &= X \otimes Z \otimes Z \\
U^\dagger(I \otimes X \otimes I)U &= Z \otimes X \otimes Z \\
U^\dagger(I \otimes I \otimes X)U &= Z \otimes Z \otimes X \\
U^\dagger(Y \otimes I \otimes I)U &= Y \otimes X \otimes X \\
U^\dagger(I \otimes Y \otimes I)U &= X \otimes Y \otimes X \\
U^\dagger(I \otimes I \otimes Y)U &= X \otimes X \otimes Y \\
U^\dagger(Z \otimes I \otimes I)U &= Z \otimes Y \otimes Y \\
U^\dagger(I \otimes Z \otimes I)U &= Y \otimes Z \otimes Y \\
U^\dagger(I \otimes I \otimes Z)U &= Y \otimes Y \otimes Z
\end{aligned} \tag{5.26}$$

where $\{X, Y, Z, I\}$ are the Pauli operators. The circuit to implement this unitary consists of 3 XX gates followed by 3 YY gates; the latter can be implemented by adding rotations to an XX gate to move the interaction to the YY basis. Because this unitary is fully real, $U_S = U_S^*$. The gate sequence to produce this unitary is shown in Figure 5.16(a). A circuit with a continuously-adjustable parameter θ that scans how much scrambling the unitary performs is shown in Figure 5.16(b); for $\theta = 0$, the unitary is the identity I , and for $\theta = \frac{\pi}{2}$, the unitary is the maximally-scrambling unitary U_S . We note that the 3 XX gates in each group of consecutive XX gates can be done in any order.

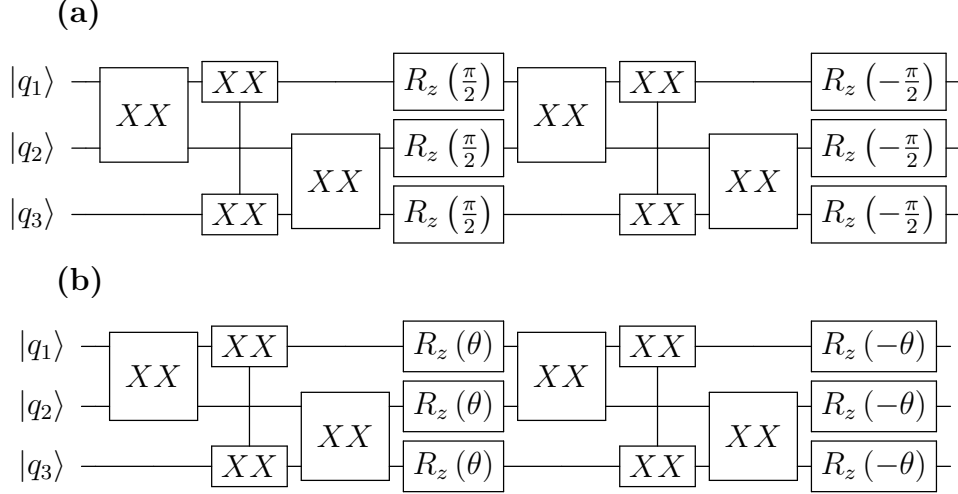


Figure 5.16: (a) Circuit to produce the unitary U_S . (b) Circuit for $U(\theta)$, with an adjustable parameter θ that goes from the identity $U(\theta = 0) = I$ to the maximally-scrambling $U(\theta = \frac{\pi}{2}) = U_S$, allowing for a scan of the amount of scrambling performed by the unitary. XX indicates a $XX(\frac{\pi}{4})$ gate.

5.9.4 Scrambling Unitary U_{CZ}

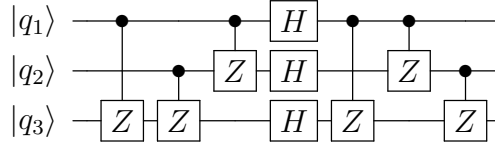
An additional scrambling unitary, U_{CZ} , is also probed in [31] for use with the Grover protocol, where

$$U_{CZ} = \frac{1}{2\sqrt{2}} \begin{pmatrix} 1 & 1 & 1 & -1 & 1 & -1 & -1 & -1 \\ 1 & -1 & 1 & 1 & 1 & 1 & -1 & 1 \\ 1 & 1 & -1 & 1 & 1 & -1 & 1 & 1 \\ -1 & 1 & 1 & 1 & -1 & -1 & -1 & 1 \\ 1 & 1 & 1 & -1 & -1 & 1 & 1 & 1 \\ -1 & 1 & -1 & -1 & 1 & 1 & -1 & 1 \\ -1 & -1 & 1 & -1 & 1 & -1 & 1 & 1 \\ -1 & 1 & 1 & 1 & 1 & 1 & 1 & -1 \end{pmatrix}. \quad (5.27)$$

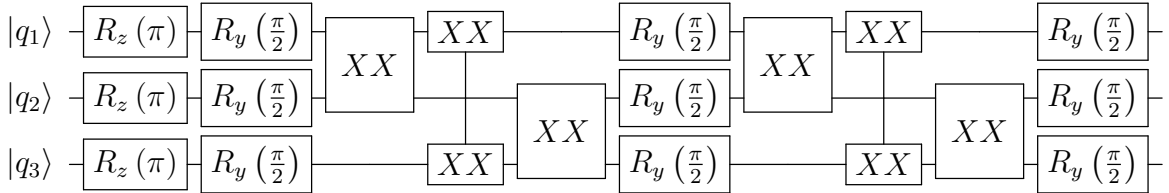
The gate sequence yielding this unitary is shown in Figure 5.17(a), and is based on an arrangement of 6 Control- Z gates. An optimized gate sequence is shown in Figure 5.17(b). Additionally, a gate sequence with the same number of

operations as the unitary but instead performing the identity is shown in Figure 5.17(c), allowing for direct comparisons and error characterization. As with U_S , we note that the 3 XX gates in each group of consecutive XX gates can be done in any order. Additionally, the rotations before the first XX gate and after the last XX gate can all have their signs flipped without changing the unitary performed. This comes in handy when optimizing the overall circuit with the Grover protocol, and allows for some XX gate cancellations (see Section 5.9.6.) Finally, we note that $U_{CZ} = U_{CZ}^* = U_{CZ}^T$.

(a) U_{CZ}



(b) U_{CZ} , optimized



(c) Identity I

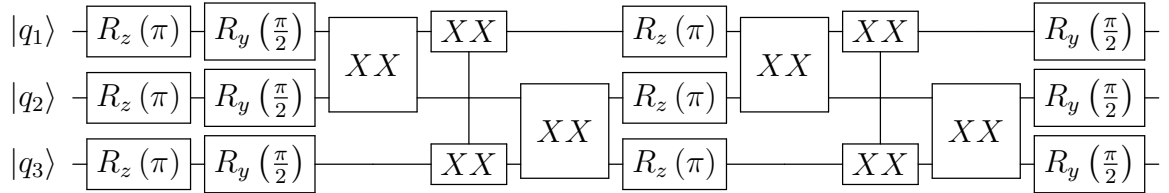


Figure 5.17: (a) Basic circuit to produce the scrambling unitary U_{CZ} . (b) Optimized circuit to produce the scrambling unitary U_{CZ} . (c) Identity circuit requiring the same number of gates as U_{CZ} , allowing for error characterization and direct comparisons between maximally- and minimally-scrambling unitaries. XX indicates a $XX \left(\frac{\pi}{4}\right)$ gate.

5.9.5 Classical Scrambler

Some unitaries will scramble quantum information non-maximally. This means they can transform 1-body information into 3-body information in some bases, but not others. Maximally-scrambling unitaries transform 1-body information into 3-body information in all bases. One such unitary is the “classical” scrambling unitary U_{CS} , which scrambles information in the X and Y bases, but not information in Z .

Here,

$$U_{CS} = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 \end{pmatrix} \quad (5.28)$$

and we have

$$\begin{aligned} U_{CS}^\dagger(X \otimes I \otimes I)U_{CS} &= X \otimes Z \otimes Z \\ U_{CS}^\dagger(I \otimes X \otimes I)U_{CS} &= Z \otimes X \otimes Z \\ U_{CS}^\dagger(I \otimes I \otimes X)U_{CS} &= Z \otimes Z \otimes X \\ U_{CS}^\dagger(Y \otimes I \otimes I)U_{CS} &= Y \otimes Z \otimes Z \\ U_{CS}^\dagger(I \otimes Y \otimes I)U_{CS} &= Z \otimes Y \otimes Z \\ U_{CS}^\dagger(I \otimes I \otimes Y)U_{CS} &= Z \otimes Z \otimes Y. \end{aligned} \quad (5.29)$$

However, $U_{CS}^\dagger(Z \otimes I \otimes I)U_{CS}$, $U_{CS}^\dagger(I \otimes Z \otimes I)U_{CS}$, and $U_{CS}^\dagger(I \otimes I \otimes Z)U_{CS}$ are not equal to any combination of three Pauli matrices, and therefore no scrambling occurs in that basis. The gate sequence yielding this unitary (controlled- Z gates connecting each pair of qubits) is shown in Figure 5.18(a), with its optimized version in Figure 5.18(b).

Because this unitary only scrambles classical information, the circuit in Figure 5.13 will only teleport classical information; in other words, information about populations will be teleported from the input to the output state, but not information about the phase on each state. For an input state $|\psi\rangle = a|0\rangle + b|1\rangle$ on qubit 1, a measurement on the output qubit φ will yield a^2 population in $|0\rangle$ and b^2 population in $|1\rangle$ conditioned on the Bell pair $|00\rangle + |11\rangle$. Additionally, the probability P of getting the Bell pair $|00\rangle + |11\rangle$ will be 50% rather than 25%.

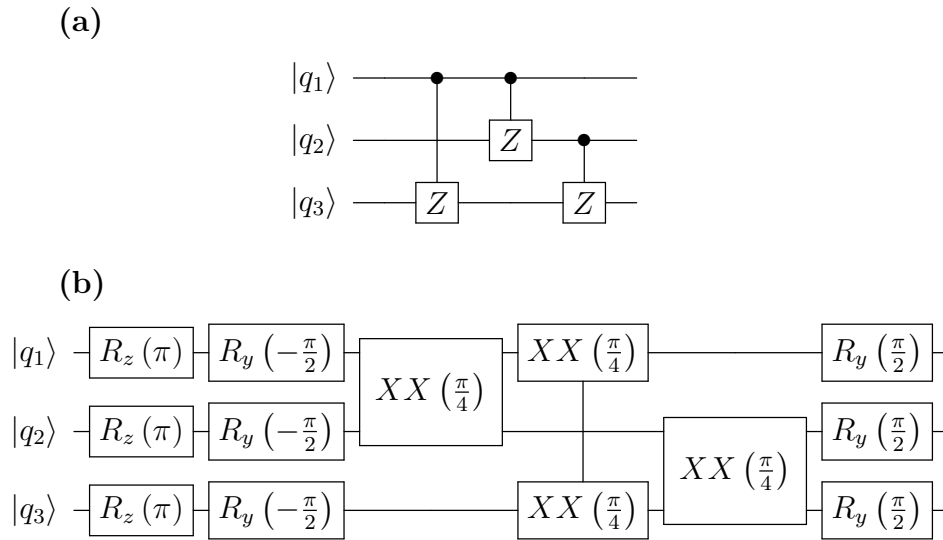


Figure 5.18: (a) Basic circuit to produce the unitary U_{CS} , which scrambles classical but not phase information. (b) Optimized circuit to produce the unitary U_{CS} .

5.9.6 Circuit Optimization

We can eliminate several XX gates by cancelling XX gates repeated between repeated scrambling unitaries on the same qubits. Additionally, we can use properties of EPR pairs that allow us to “move” gates from one half of an EPR pair to the other half, subject to a transformation. Specifically, any unitary U applied on one half of an EPR pair is the same as that unitary’s transpose U^T applied to the other half (see Figure 5.19). As an example of how this improves circuit performance, Figures 5.20 and 5.21 show the gates eliminated with these techniques for U_{CZ} implemented in the circuit from Figure 5.13.

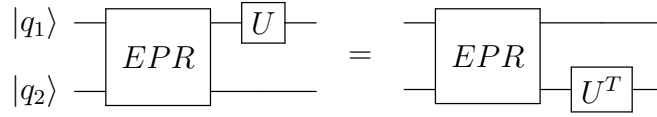


Figure 5.19: A unitary U performed on one half of an EPR pair is the same as its transpose U^T performed on the other half of the EPR pair.

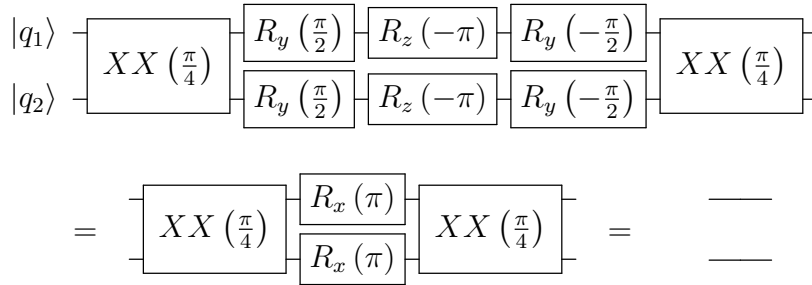


Figure 5.20: Equivalence used to eliminate XX gates in neighboring, repeated U_{CZ} unitaries.

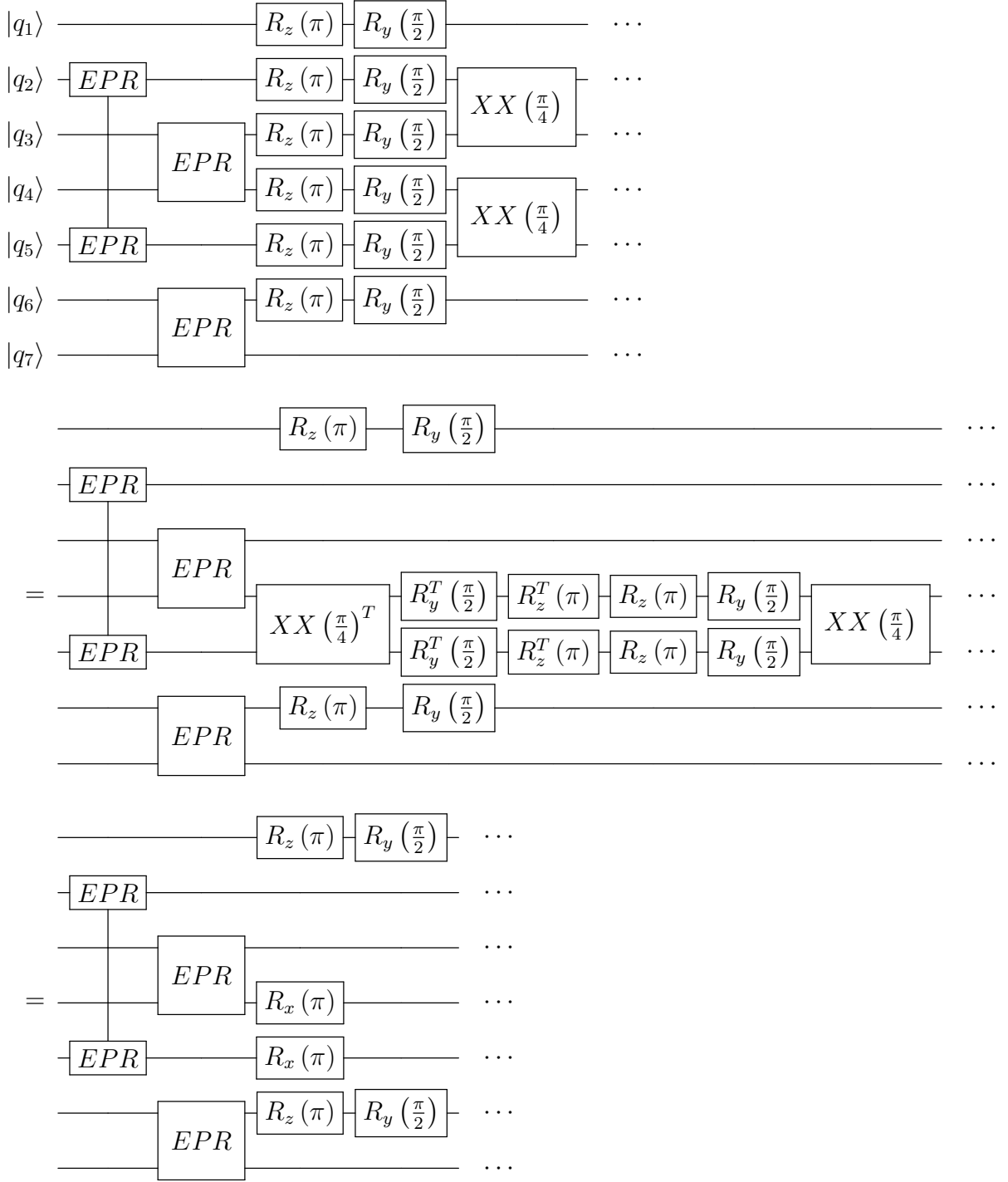


Figure 5.21: Equivalence used to eliminate XX gates from stacked U_{CZ} unitaries using the EPR property in Figure 5.19.

Chapter 6: Complete 3-Qubit Grover Search

The Grover quantum search algorithm is a hallmark application of a quantum computer with a well-known speedup over classical searches of an unsorted database. Here, we report results for a complete three-qubit Grover search algorithm using the scalable quantum computing technology of trapped atomic ions, with better-than-classical performance. Two methods of state marking are used for the oracles: a phase-flip method employed by other experimental demonstrations, and a previously-undemonstrated Boolean method requiring an ancilla qubit that is directly equivalent to the state-marking scheme required to perform a classical search. The data presented here is also presented in [24].

6.1 The Grover Search Algorithm

Searching large databases is an important problem with broad applications. The Grover search algorithm [86,87] provides a powerful method for quantum computers to perform searches with a quadratic speedup in the number of required database queries over classical computers. It is an optimal search algorithm for a quantum computer [88], and has further applications as a subroutine for other

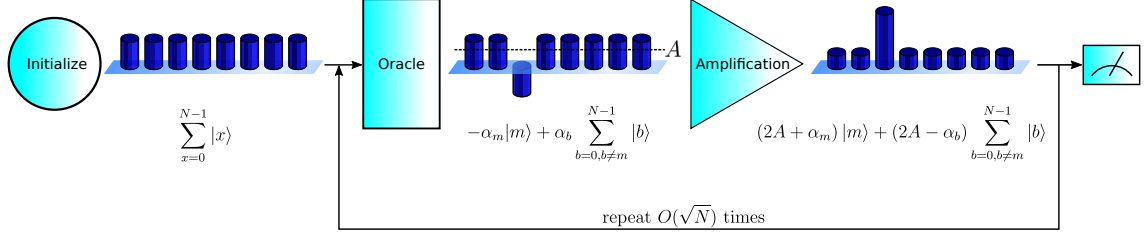


Figure 6.1: Evolution of relative amplitudes for each state during a Grover search algorithm. The initialization stage creates an equal superposition of all possible input states, so the amplitude $\alpha_x = 1$ for all basis states $|x\rangle$. The oracle stage marks the desired state, so the amplitude α_m of the marked state $|m\rangle$ becomes negative while the amplitudes α_b of the unmarked states $|b\rangle, b \neq m$ remain unchanged. The amplification stage performs a reflection about the mean vector $\sum_{x=0}^{N-1} |x\rangle$, which has amplitude $A = \frac{1}{N} \sum_{x=0}^{N-1} \alpha_x = \frac{1}{N} (-\alpha_m + (N-1)\alpha_b)$, to amplify the marked state. An appropriate number of repetitions of the oracle and amplification stages will maximize the amplitude of the correct answer. All qubit states are normalized by the factor $\frac{1}{\sqrt{N}}$. The algorithm can also be generalized to mark and amplify the amplitude of t desired states.

quantum algorithms [89,90]. Searches with two qubits have been demonstrated on a variety of platforms [91–97] and proposed for others [98], but larger search spaces have only been demonstrated on a non-scalable NMR system [73].

The Grover search algorithm has 4 stages: initialization, oracle, amplification, and measurement, as shown in Figure 6.1. The initialization stage creates an equal superposition of all states, $\frac{1}{\sqrt{N}} \sum_{x=0}^{N-1} |x\rangle$ for all basis states $|x\rangle$. The oracle stage marks the solution(s) $|m\rangle$ by flipping the sign of that state’s amplitude, yielding $\frac{1}{\sqrt{N}} \left(-\alpha_m \sum_m |m\rangle + \sum_{b \neq m} \alpha_b |b\rangle \right)$ for marked state amplitude(s) α_m and non-marked state amplitudes α_b . Inverting an amplitude α about an average amplitude A can be written as $-\alpha + 2A$ or $A + (A - \alpha)$. Here, the amplification stage performs a reflection about the mean vector $\sum_{x=0}^{N-1} |x\rangle$, thus increasing the

amplitude of the marked state:

$$|\Psi_{amp}\rangle = \frac{1}{\sqrt{N}} \left[(2A + \alpha_m) \sum_m |m\rangle + (2A - \alpha_b) \sum_{b \neq m} |b\rangle \right], \quad (6.1)$$

where A is the amplitude of the mean vector. Finally, the algorithm output is measured. For a search database of size N with t possible solutions, the single-shot probability of measuring the correct answer is maximized to near-unity by repeating the oracle and amplification stages $O(\sqrt{N/t})$ times before measurement. The probability of failing to measure a correct answer is guaranteed to be less than $\frac{t}{N}$ if the optimal number of iterations, $j = \lfloor \frac{\pi}{4\theta} \rfloor$, is used, where $\sin^2(\theta) = \frac{t}{N}$, $0 < \theta \leq \frac{\pi}{2}$; for $t \ll N$, the probability of failure is negligible [86,87]. By comparison, a classical search algorithm will get the correct answer after an average of $N/2$ queries of the oracle, and in the worst case may require up to N queries to find the correct answer. For large databases, this quadratic speedup represents a significant advantage for quantum computers.

Here, we implement the Grover search algorithm on $n = 3$ qubits, which corresponds to a search database of size $N = 2^n = 8$. All searches are performed with a single iteration ($j = 1$). After a single iteration, the initial amplitudes $\alpha_x = 1$ on all basis states $|x\rangle$, so the mean vector has amplitude $A = \frac{1}{N} \sum_{x=0}^{N-1} \alpha_x = \frac{1}{N} (-\alpha_m \cdot t + (N - t) \alpha_b) = \frac{N-2t}{N}$. Plugging in to Equation 6.1 above and rewriting

the inversion about the mean vector as $A + (A - \alpha)$, we have

$$\begin{aligned}
|\Psi_{amp}(j=1)\rangle &= \frac{1}{\sqrt{N}} \left[(A + (A + \alpha_m)) \sum_m |m\rangle + (A + (A - \alpha_b)) \sum_{b \neq m} |b\rangle \right] \\
&= \frac{1}{\sqrt{N}} \left[\left(\frac{N-2t}{N} + \left(\frac{N-2t}{N} + 1 \right) \right) \sum_m |m\rangle \right. \\
&\quad \left. + \left(\frac{N-2t}{N} + \left(\frac{N-2t}{N} - 1 \right) \right) \sum_{b \neq m} |b\rangle \right] \\
&= \frac{1}{\sqrt{N}} \left[\left(\frac{N-2t}{N} + \frac{2(N-t)}{N} \right) \sum_m |m\rangle + \left(\frac{N-2t}{N} - \frac{2t}{N} \right) \sum_{b \neq m} |b\rangle \right].
\end{aligned} \tag{6.2}$$

Since there are t possible correct solutions $|m\rangle$, the probability of measuring one correct solution after one iteration is

$$P_m = t \cdot \left(\left[\frac{N-2t}{N} + \frac{2(N-t)}{N} \right] \frac{1}{\sqrt{N}} \right)^2. \tag{6.3}$$

In contrast, the optimal classical search strategy consists of a single query (equivalent to a single Grover iteration) followed by a random guess in the event the query failed. Therefore, the total probability of finding a correct solution is $P(\text{success}) = P(\text{query correct}) + P(\text{query incorrect}) * P(\text{guess correct}|\text{query incorrect})$, where $P(\text{query correct}) = \frac{t}{N}$, $P(\text{query incorrect}) = \frac{N-t}{N}$, and $P(\text{guess correct}|\text{query incorrect}) = \frac{t}{N-1}$. In general, for t solutions in the classical case,

$$P(\text{success}) = \frac{t}{N} + \frac{N-t}{N} \cdot \frac{t}{N-1}. \tag{6.4}$$

Consequently, for a single-solution algorithm ($t = 1$), the algorithmic probability of measuring the correct state after one iteration is $t \cdot \left(\left[\frac{N-2t}{N} + \frac{2(N-t)}{N} \right] \frac{1}{\sqrt{N}} \right)^2 = \left(\frac{5}{4\sqrt{2}} \right)^2 = 78.125\%$ [87], compared to $\frac{t}{N} + \frac{N-t}{N} \cdot \frac{t}{N-1} = \frac{1}{8} + \frac{7}{8} \cdot \frac{1}{7} = 25\%$ for the optimal classical search strategy.

In the two-solution case ($t = 2$), where two states are marked as correct answers during the oracle stage and both states' amplitudes are amplified in the algorithm's amplification stage, the probability of measuring one of the two correct answers is $t \cdot \left(\left[\frac{N-2t}{N} + \frac{2(N-t)}{N} \right] \frac{1}{\sqrt{N}} \right)^2 = 2 \cdot \left(\frac{16}{8\sqrt{8}} \right)^2 = 2 \cdot \left(\frac{1}{\sqrt{2}} \right)^2 = 100\%$ for the quantum case, as compared to $\frac{2}{8} + \frac{6}{8} \cdot \frac{2}{7} = \frac{13}{28} \approx 46.4\%$ for the classical case.

The algorithm is performed with both a phase oracle, which has been previously demonstrated on other experimental systems, and a Boolean oracle, first reported here, which requires more resources but is directly comparable to a classical search. All quantum solutions are shown to outperform their classical counterparts.

6.2 Oracles

We examine two alternative methods of encoding the marked state within the oracle. While both methods are mathematically equivalent [65], only one is directly comparable to a classical search. The previously-undemonstrated Boolean method requires the use of an ancilla qubit initialized to $|1\rangle$, as shown in Figure 6.2(a). The oracle is determined by constructing a circuit out of *NOT* and $C^k(\text{NOT})$ ($k \leq n$) gates such that, were the oracle circuit to be implemented classically, the ancilla bit would flip if and only if the input to the circuit is one of the marked states. By using

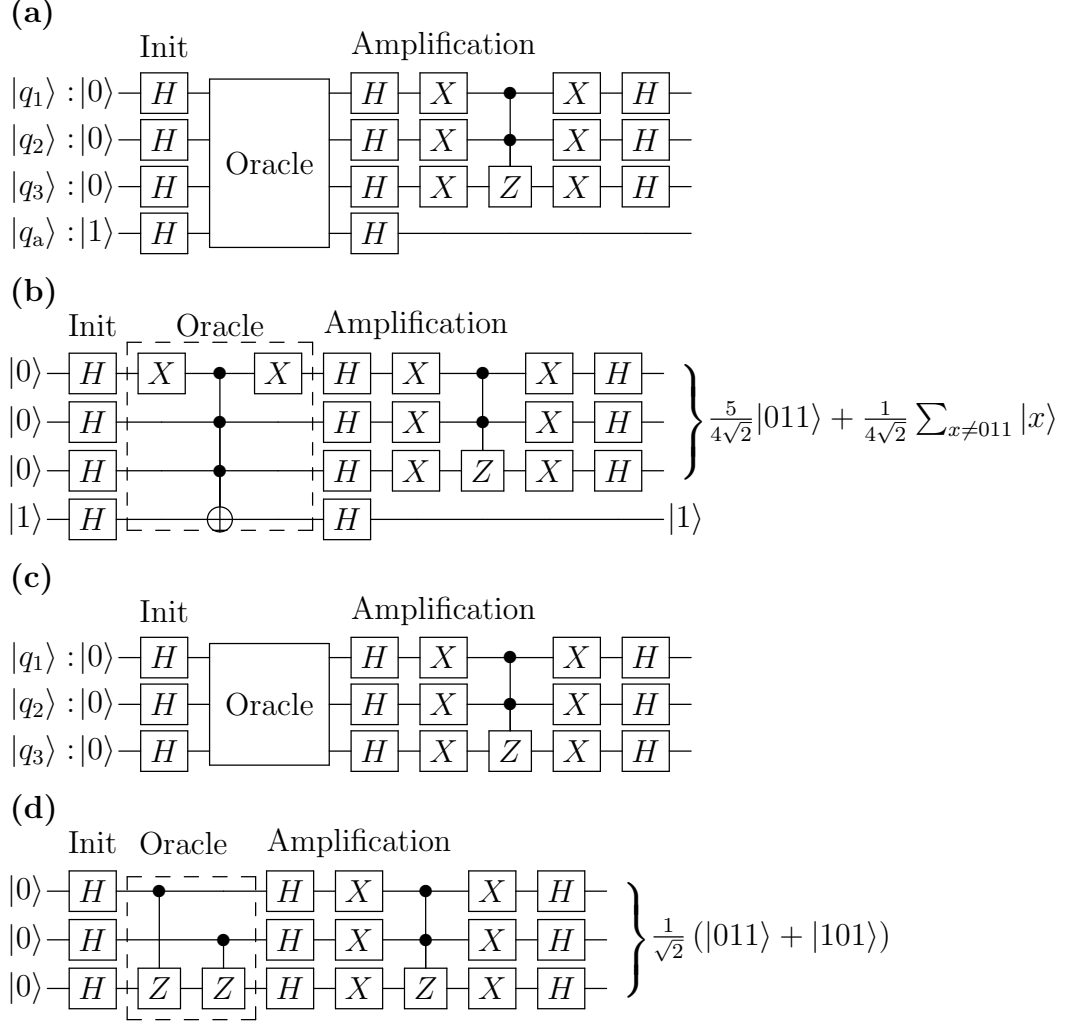


Figure 6.2: (a) General circuit diagram for a Grover search algorithm using a Boolean oracle, depicted using standard quantum circuit diagram notation [65]. The last qubit q_a is the ancilla qubit. (b) Example single-solution Boolean oracle marking the $|011\rangle$ state. (c) General circuit diagram for a Grover search algorithm using a phase oracle. (d) Example two-solution phase oracle marking the $|011\rangle$ and $|101\rangle$ states.

classically available gates, this oracle formulation is directly equivalent to the classical search algorithm, and therefore can most convincingly demonstrate the quantum algorithm's superiority. On a quantum computer, because the initialization sets up an equal superposition of all possible input states, the $C^m(NOT)$ gate targeted on the ancilla provides a phase kickback that flips the phase of the marked state(s) in

the data qubits. An example oracle is shown in Figure 6.2(b) to illustrate this. The phase method of oracle implementation does not require the ancilla qubit. Instead, the oracle is implemented with a circuit consisting of Z and $C^k(Z)$ ($k \leq n-1$) gates that directly flip the phase(s) of the state(s) to be marked (see Figures 6.2(c-d)).

6.3 Circuit Implementation

The Grover search algorithm is implemented using circuits that are equivalent to those shown in Figures 6.1(b,d), but with the initialization and amplification stages optimized to minimize gate times, as shown in Figures 6.3(a-b). The circuits shown are for use with Boolean oracles; in the phase oracle case, the ancilla qubit q_a is simply omitted. To preserve the modularity of the algorithm, the initialization stage and amplification stage were each optimized without regard to the contents of the oracle, so each possible oracle can simply be inserted into the algorithm without making any changes to the other stages.

Oracles for the Grover search algorithm were constructed using a combination of reversible and classical logic synthesis techniques. For Boolean oracles, reversible logic synthesis was employed to find a set of $X, C^N(NOT)$ gates that marked the desired state(s) for each oracle. For phase oracles, EXOR polynomial synthesis was used to find a set of $Z, C^N(Z)$ gates that marked the desired state(s) for each oracle. For example, for Boolean oracles, the selection was limited to the classically available X (or NOT) and $C^N(NOT)$ gates, and a reversible circuit was constructed such that the output bit (corresponding to the ancilla qubit in the quantum oracle) would

Quantum circuit diagram for the 4-qubit algorithm. The circuit involves four qubits, labeled $|q_1\rangle$, $|q_2\rangle$, $|q_3\rangle$, and $|q_a\rangle$, each initialized to $|0\rangle$.

- Qubits $|q_1\rangle$, $|q_2\rangle$, and $|q_3\rangle$ each pass through a rotation gate $R_x(\pi)$ followed by a rotation gate $R_y(-\frac{\pi}{2})$.
- Qubit $|q_a\rangle$ passes through a rotation gate $R_y(-\frac{\pi}{2})$.
- All four qubits then enter a multi-controlled NOT gate, represented by a large rectangle with four control lines (one for each qubit) and a target line.

Figure 1: Quantum circuit for the generalized quantum state transfer protocol. The circuit consists of two main parts. The top part involves qubits $|q_1\rangle$, $|q_2\rangle$, $|q_3\rangle$, and $|q_a\rangle$. $|q_1\rangle$ has a $R_y((1-\beta)\frac{\pi}{2})$ gate, followed by a controlled- $R_x(\beta\frac{3\pi}{4})$ gate with $|q_2\rangle$ as control and $|q_1\rangle$ as target, and then a controlled- $XX(\beta\frac{\pi}{8})$ gate with $|q_2\rangle$ as control and $|q_1\rangle$ as target. $|q_2\rangle$ has a $R_y((1-\gamma)\frac{\pi}{2})$ gate, followed by a controlled- $R_x(\gamma\frac{\pi}{2})$ gate with $|q_1\rangle$ as control and $|q_2\rangle$ as target, and then a controlled- $XX(\gamma\frac{\pi}{8})$ gate with $|q_1\rangle$ as control and $|q_2\rangle$ as target. $|q_3\rangle$ has a $R_y(\pi)$ gate, followed by a controlled- $R_x(\frac{\pi}{4})$ gate with $|q_2\rangle$ as control and $|q_3\rangle$ as target, and then a controlled- $XX(\beta\frac{\pi}{8})$ gate with $|q_2\rangle$ as control and $|q_3\rangle$ as target. $|q_a\rangle$ has a $R_y(\frac{\pi}{2})$ gate, followed by a controlled- $R_x(\pi)$ gate with $|q_2\rangle$ as control and $|q_a\rangle$ as target. The bottom part of the circuit involves multiple qubits. The first qubit has a controlled- $XX(\alpha\frac{\pi}{4})$ gate with the second qubit as control and the first qubit as target. The second qubit has a controlled- $R(-\alpha\beta\gamma\frac{2\pi}{3}, (\frac{\alpha\beta+1}{2})\pi - \alpha\beta\gamma P)$ gate with the first qubit as control and the second qubit as target, followed by a controlled- $XX(\gamma\frac{\pi}{8})$ gate with the first qubit as control and the second qubit as target, and then a controlled- $R(\pi, -\alpha\beta\gamma\frac{\pi}{4})$ gate with the first qubit as control and the second qubit as target. The third qubit has a controlled- $XX(\alpha\frac{\pi}{4})$ gate with the second qubit as control and the third qubit as target, followed by a controlled- $R_y((\beta-1)\frac{\pi}{2})$ gate with the second qubit as control and the third qubit as target, and then a controlled- $R_y(-\frac{\pi}{2})$ gate with the second qubit as control and the third qubit as target. The fourth qubit has a controlled- $R_y(\pi)$ gate with the second qubit as control and the fourth qubit as target.

119

be flipped if and only if a marked state was used as the input to the circuit. While there are many possible circuit constructions for each oracle, the oracle chosen for implementation was one that first minimized the number of two-qubit interactions required, and then minimized the number of single-qubit interactions needed. The synthesis techniques used are scalable and can be applied to oracles of any size. The oracles used here were implemented as per the circuit diagrams shown in Table 6.1 for single-solution oracles and Table 6.2 for two-solution oracles. The algorithm is executed for all 8 possible single-result oracles and all 28 possible two-result oracles.

Mark	Boolean Oracle	Phase Oracle	Mark	Boolean Oracle	Phase Oracle
000			100		
001			101		
010			110		
011			111		

Table 6.1: Table of all oracles used for the single-solution Grover

search algorithm.

Marked	Boolean Oracle	Phase Oracle
000, 001		
000, 010		
000, 011		
000, 100		
000, 101		
000, 110		
000, 111		
001, 010		

Table 6.2 – *Continued on next page*

Table 6.2 – Continued from previous page

Marked	Boolean Oracle	Phase Oracle
001, 011		
001, 100		
001, 101		
001, 110		
001, 111		
010, 011		
010, 100		
010, 101		

Table 6.2 – Continued on next page

Table 6.2 – Continued from previous page

Marked	Boolean Oracle	Phase Oracle
010, 110		
010, 111		
011, 100		
011, 101		
011, 110		
011, 111		
100, 101		
100, 110		

Table 6.2 – Continued on next page

Table 6.2 – *Continued from previous page*

Marked	Boolean Oracle	Phase Oracle
100, 111		
101, 110		
101, 111		
110, 111		

Table 6.2: Table of all oracles used for the two-solution Grover search algorithm.

Detailed information and individual fidelities for constituent composite gates can be found in Chapter 5.

6.4 Data

Figures 6.4 and 6.5 show the results, respectively, of single- and two-solution Grover search algorithms, each using both the Boolean and phase marking methods. All possible oracles are tested to demonstrate a complete Grover search (see Tables 6.1 and 6.2). Two figures of merit are provided with the data for each oracle. The

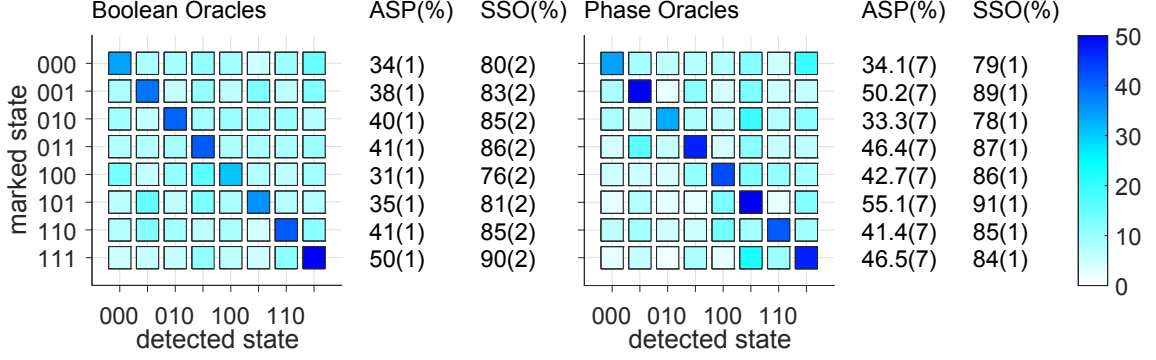


Figure 6.4: Results from a single iteration of a single-solution Grover search algorithm performed on a 3-qubit database. Data for the Boolean oracle formulation is shown on the left, and data for the phase oracle formulation is shown on the right. The plots show the probability of detecting each output state. All values shown are percents, with a theoretical ASP of 78.1% and theoretical SSO of 100%. Data is corrected for average SPAM errors of 1%.

algorithm success probability (ASP) is the probability of measuring the marked state as the experimental outcome. For the two-solution algorithm, the ASP is calculated by summing the probabilities of measuring each of the two marked states. The squared statistical overlap (SSO) measures the statistical overlap between the measured and expected populations for all states: $SSO = \left(\sum_{j=0}^N \sqrt{e_j m_j} \right)^2$, where e_j is the expected population and m_j is the measured population for each state j [99]. Additionally, all of the data shown in this paper is corrected to account for state preparation and measurement (SPAM) errors (see figure captions for values), similar to the method proposed in [39] while also accounting for multi-ion crosstalk [19]. All uncertainties given are statistical uncertainties based on the number of experiments performed.

The single-iteration, single-solution Grover search algorithm shown in Figure 6.4 has a theoretical ASP of 78.1%, as discussed above. The SSO takes into account

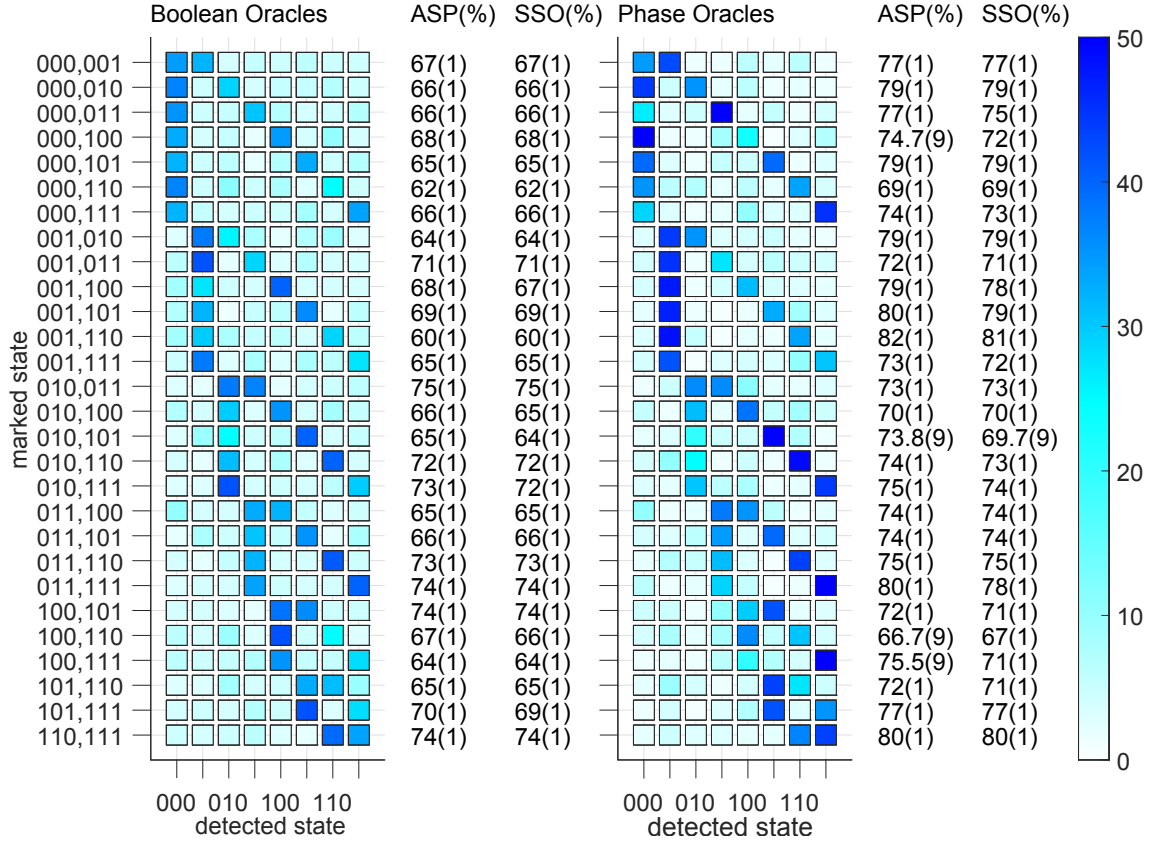


Figure 6.5: Results from the execution of a two-solution Grover search algorithm performed on a 3-qubit database. Data for the Boolean oracle formulation is shown on the left, and data for the phase oracle formulation is shown on the right. The plots show the probability of detecting each output state. All values shown are percents. The ASP is the sum of the probabilities of detecting each of the two marked states. Data is corrected for average SPAM errors of 1%.

that the 7 unmarked states then have equal expected probabilities totaling 21.9% of being measured. For all Boolean oracles, the average ASP is 38.9(4)% and the average SSO is 83.2(7)%, while phase oracles have an average ASP of 43.7(2)% and an average SSO of 84.9(4)%; the reduced use of resources in the phase oracles (10 $XX(\chi)$ gates and 3 qubits for phase oracles compared to 16 $XX(\chi)$ gates and 5 qubits for Boolean oracles) results in better performance, as expected. These results compare favorably with the classical ASP of 25%.

The two-solution Grover search algorithm shown in Figure 6.5 has a theoretical ASP of 100%, as discussed above. For all Boolean oracles, the average ASP is 67.9(2)% and the average SSO is 67.6(2)%, while phase oracles have an average ASP of 75.3(2)% and an average SSO of 74.4(2)%; the reduced use of resources in the phase oracles (6-8 $XX(\chi)$ gates and 3 qubits for phase oracles compared to 10-14 $XX(\chi)$ gates and 4 qubits for Boolean oracles) results in better performance, as expected. For all oracles in both cases, the two states with the highest measurement probability are also the two marked states. These results compare favorably with the classical ASP of 46.4%.

6.5 Additional Iterations

Performing an additional iteration on the single-solution Grover search algorithms was inhibited by circuit-depth limitations in the experimental control program, which will be fixed for future work. Here, we estimate the impact of an additional iteration on algorithm performance. While a single iteration of the single-

solution Grover search algorithm has a maximum ASP of 78.125%, performing two iterations raises the maximum ASP to 94.5312%. Applying the error estimation models used in [27], the likely performances of two Grover search algorithm cases were examined: the Boolean 000 oracle and the phase 111 oracle, which correspond to the worst- and best-case oracles by gate count. The random error estimation model assumes random error propagation for each operation of the form $(1 - \epsilon_g)^{\sqrt{N}}$, and the systematic error estimation model assumes coherent over- or under-rotations for each operation and has the form $(1 - \epsilon_g)^N$, where N is the number of operations and ϵ_g is the error per operation. Based on the analysis in [27] on this same system, we expect the actual results to fall somewhere between these two models. For a single iteration of the phase 111 oracle, we estimate an SSO of 86% and an ASP of 41% using the random error model, or an SSO of 61% and an ASP of 16% using the systematic error model; as in the analysis in [27], we compare this to the measured SSO of 84(1)% and ASP of 46.5(7)% and see that the experiment performs slightly worse than the random error model, and better than the systematic error model. Extending the analysis to two iterations of the phase 111 oracle, we estimate an SSO of 81% and ASP of 60% using the random error model, or an SSO of 40% and an ASP of 19% using the systematic error model.

Similarly, for a single iteration of the Boolean 000 oracle, we estimate an SSO of 83% and an ASP of 37% using the random error model, or an SSO of 45% and an ASP of 6% using the systematic error model; as before, we compare this to the measured SSO of 80(2)% and ASP of 34(1)% and see that the experiment performs slightly worse than the random error model. Extending the analysis to two

iterations of the phase 111 oracle, we estimate an SSO of 77% and ASP of 55% using the random error model, or an SSO of 22% and an ASP of 6% using the systematic error model. We expect the experiment would perform somewhere between these two models, although we do not know how well these error models hold for very deep circuits; it is not clear whether the experiment would have outperformed the best classical strategy with two iterations, which has a success probability of 37.5%.

6.6 Outlook

We note that this implementation of the Grover search algorithm scales linearly in the two-qubit gate count and ancilla count for increasing search database size as a function of the number of qubits n , and for a constant number of solutions t . For a database of size $N = 2^n$ stored on n qubits, the amplification stage requires one Toffoli- n gate, and the t -solution oracle stage requires at worst t Toffoli- n (for a phase oracle) or Toffoli- $(n + 1)$ (for a Boolean oracle) gates; optimal oracles for particular sets of marked states may require even fewer two-qubit gates. The method used in Section 5.4.2 to construct the Toffoli-4 circuit scales to Toffoli- n gates as $6n - 13$ in the two-qubit gate count and as $\lceil \frac{n-3}{2} \rceil$ in the ancilla count [77]. This paves the way for more extensive use of the Grover search algorithm in solving larger problems on quantum computers, including using the circuit as a subroutine for other quantum algorithms.

Chapter 7: Parallel 2-Qubit Operations

In any computer, quantum or classical, parallel operations are highly desirable. Parallel operations save considerable time over performing operations in series. Current trends in the (classical) computer industry include a focus on developing multi-core processors and developing software with multiple parallel threads. Quantum computers have a long way to go before such a scale will be possible, but the ability to perform parallel operations is nevertheless crucial to our ultimate ability to scale up.

In ion-trap quantum computers, two-qubit interactions are mediated by the normal modes of motion in the ion chain. However, as the chain grows in size, so do the number of modes of motion, and spectral crowding makes sideband resolution more difficult. Two-qubit interactions can be implemented in less time by using more optical power in the Raman beams, but this has the consequence of reducing sideband resolution, which degrades gate fidelity. Consequently, gate speed is limited by sideband resolution, a limitation that gets worse as the processor size grows. Parallel two-qubit operations are a tool to speed up computation that avoids this problem. Parallel gate operations also reduce the overall gate depth of a given process, permitting more operations before decoherence and error accumulation obviate

any meaningful outputs. Several quantum computing subroutines and composite gates have been shown to benefit directly from parallel entangling gates, including the quantum Fourier transform [100, 101], multiply controlled Toffoli gates, and stabilizer circuits [101]. Quantum algorithms that will similarly benefit from parallel entangling gates include Shor’s integer factoring [102], solving the discrete logarithm problem over the elliptic curve group [103], simulating Hamiltonian dynamics using the Suzuki-Trotter formula [104], and quantum chemistry [105].

Here, we present experimental results for a pair of two-qubit gates performed simultaneously in a single chain of trapped ions. We employ a pulse shaping scheme that modulates the phase and amplitude of the Raman laser to drive programmable high-fidelity 2-qubit XX gates in parallel by coupling to the collective modes of motion of the ion chain. Ensuring the interaction produced yields only spin-spin interactions between the desired pairs with neither residual spin-motion entanglement nor crosstalk spin-spin entanglement between non-desired ion pairs is a nonlinear constraint problem, and optimal pulse shapes are found using optimization techniques. As an application of these parallel operations, we demonstrate the quantum full adder using a depth-4 circuit requiring the use of parallel 2-qubit operations as well as single- and two-qubit operations previously demonstrated on this system.

7.1 Theory

We perform parallel gates by modulating the Rabi frequency of the individual Raman beams. This is accomplished in a similar manner to the 2-qubit gates

already implemented on this experiment [19, 53] (see Section 3.2), which implement entangling XX gates using a Mølmer-Sørensen scheme [41, 52, 55, 56], where red and blue sidebands are applied to entangle the ion spins of illuminated ions with the radial normal motional modes of the ions in the trap. Using the motional modes as an information bus, the spin states of separate ions become entangled, and the pulse shape is engineered so that at the end of the gate, the motional modes are entirely disentangled from the spins, leaving only spin-spin entanglement [53, 57, 59]. The pulse shape is controlled by dividing the gate time into several equal-length segments, and adjusting the amplitude of the gate modulation in each segment. Alternative possible schemes include varying the detuning μ [106] or the beatnote phase [107] to engineer high-fidelity gates; the former scheme has been demonstrated to generate 2-qubit entangling XX gates on this system [28].

In order to perform parallel entangling operations involving 2 pairs of qubits (i, j) and (m, n) in a chain of N ions with N motional modes ω_k using red and blue sidebands with detuning μ , we have a 4-qubit unitary, as follows:

$$\begin{aligned}
U_{||}(\tau) &= \exp \left(i \sum_i \phi_i(\tau) \sigma_i^x + i \sum_{i < j} \chi_{ij}(\tau) \sigma_i^x \sigma_j^x \right) \\
&= \exp \left(i \left[\phi_i(\tau) \sigma_i^x + \phi_j(\tau) \sigma_j^x + \phi_m(\tau) \sigma_m^x + \phi_n(\tau) \sigma_n^x \right. \right. \\
&\quad \left. \left. + \chi_{ij}(\tau) \sigma_i^x \sigma_j^x + \chi_{mn}(\tau) \sigma_m^x \sigma_n^x + \chi_{im}(\tau) \sigma_i^x \sigma_m^x \right. \right. \\
&\quad \left. \left. + \chi_{in}(\tau) \sigma_i^x \sigma_n^x + \chi_{jm}(\tau) \sigma_j^x \sigma_m^x + \chi_{jn}(\tau) \sigma_j^x \sigma_n^x \right] \right) \quad (7.1)
\end{aligned}$$

where τ is the gate time, the spin-motion interaction $\phi_i(\tau)$ is

$$\phi_i(\tau) = \alpha_{i,k}(\tau)\hat{a}_k^\dagger - \alpha_{i,k}^*(\tau)\hat{a}_k, \quad (7.2)$$

\hat{a}_k^\dagger and \hat{a}_k are the raising and lowering operators for the motional phonons, the spin-motion parameter $\alpha_{i,k}(\tau)$ is

$$\alpha_{i,k}(\tau) = \int_0^\tau \eta_{i,k} \Omega_i(t) \sin(\mu t) e^{i\omega_k t} dt, \quad (7.3)$$

$\eta_{i,k}$ is the spin-motion coupling or Lamb-Dicke parameter, $\Omega_i(t)$ is the Rabi frequency of the optical field applied to ion i , and the spin-spin interaction term $\chi_{ij}(\tau)$ is

$$\chi_{ij}(\tau) = 2 \int_0^\tau dt' \int_0^{t'} dt \sum_k \eta_{i,k} \eta_{j,k} \Omega_i(t) \Omega_j(t) \sin(\mu t) \sin(\mu t') \sin(\omega_k(t' - t)). \quad (7.4)$$

At the end of the gate, the spin-motion terms must go to zero, ensuring that all mode trajectories in phase space return to the origin. So, we require all $4N$ spin-motion parameters (4 ions, N modes) $\alpha_{\{i,j,m,n\},k}(\tau) = 0$. Since we wish to entangle only the ions pairs (i, j) and (m, n) , we do not wish to entangle the “crosstalk” pairs (i, m) , (i, n) , and so on. Consequently, for the entangling pairs, we require $\chi_{ij} = \chi_{ij}^{\text{ideal}}$ and $\chi_{mn} = \chi_{mn}^{\text{ideal}}$, where $0 < \chi^{\text{ideal}} \leq \frac{\pi}{4}$. χ^{ideal} is typically $\frac{\pi}{4}$ for a maximally entangling XX gate but can be set to smaller values to implement partially-entangling gates. For the crosstalk pairs, we require $\chi_{im} = \chi_{in} = \chi_{jm} = \chi_{jn} = 0$ to ensure no unwanted entanglement at the end of the gate. This then yields a set of $4N + 6$ parameters to control for when optimizing pulse sequences to implement parallel

XX gates:

$$\begin{aligned}
\alpha_{\{i,j,m,n\},k}(\tau) &= 0 \\
\chi_{ij}(\tau) &= \chi_{ij}^{\text{ideal}} \\
\chi_{mn}(\tau) &= \chi_{mn}^{\text{ideal}} \\
\chi_{im}(\tau) = \chi_{in}(\tau) = \chi_{jm}(\tau) = \chi_{jn}(\tau) &= 0.
\end{aligned} \tag{7.5}$$

To provide optimal control during the gate and fulfill the constraints in Equation 7.5, we divide up the gate amplitude $\Omega_i(t)$ into S segments of equal duration τ/S , and vary the amplitude in each segment Ω_s as an independent variable. In order to implement independent XX gates, we implement independent signals on the two ion pairs we want to entangle; ions (i, j) see one pulse shape, while ions (m, n) see another. Separate signals on the two ion pairs are necessary to provide sufficient control to simultaneously entangle the desired ion pairs but not the crosstalk pairings. Without two different signals, there is no way to provide cancellation to crosstalk entanglement. On the experiment, this ion-specific signal shaping is provided by the upgrade to a multi-channel AWG, as discussed in Section 2.4. The

gate amplitude on a given ion then becomes a piecewise-constant function,

$$\Omega_i(t) = \begin{cases} \Omega_1 & 0 \leq t < \tau/S \\ \Omega_2 & \tau/S \leq t < 2\tau/S \\ \vdots & \vdots \\ \Omega_s & (s-1)\tau/S \leq t < s\tau/S \\ \vdots & \vdots \\ \Omega_S & (S-1)\tau/S \leq t < \tau. \end{cases} \quad (7.6)$$

Consequently, Equations 7.3 and 7.4 can be re-written as

$$\begin{aligned} \alpha_{i,k}(\tau) &= \sum_{s=1}^S \Omega_s \left[\eta_{i,k} \int_{(s-1)\tau/S}^{s\tau/S} \sin(\mu t) e^{i\omega_k t} dt \right] \\ &= \sum_{s=1}^S \Omega_s C_{k,s}^i \end{aligned} \quad (7.7)$$

and

$$\begin{aligned} \chi_{ij}(\tau) &= \sum_{s=1}^S \sum_{s'=1}^S \Omega_s \Omega_{s'} \int_{(s-1)\tau/S}^{s\tau/S} dt' \int_{(s'-1)\tau/S}^{s'\tau/S} dt \sum_k \eta_{i,k} \eta_{j,k} \sin(\mu t) \sin(\mu t') \sin(\omega_k(t' - t)) \\ &= \sum_{s=1}^S \sum_{s'=1}^S \Omega_s \Omega_{s'} D_{s,s'}, \end{aligned} \quad (7.8)$$

where the terms

$$C_{k,s}^i = \eta_{i,k} \int_{(s-1)\tau/S}^{s\tau/S} \sin(\mu t) e^{i\omega_k t} dt \quad (7.9)$$

and

$$D_{s,s'}^{ij} = \int_{(s-1)\tau/S}^{s\tau/S} dt' \int_{(s'-1)\tau/S}^{s'\tau/S} dt \sum_k \eta_{i,k} \eta_{j,k} \sin(\mu t) \sin(\mu t') \sin(\omega_k(t' - t)) \quad (7.10)$$

are pre-calculated constants that are functions only of the motional mode frequencies ω_k , the detuning μ , and the segment number s , and so can be arranged into $S \times N$ and $S \times S$ matrices for each ion and ion-ion pair, respectively. Note that the time ordering of the double integral in Equation 7.10 requires that $t < t'$, so the time-segmented scheme requires $s \leq s'$. In the case $s = s'$, we must force the $t < t'$ constraint, yielding

$$D_{s=s'}^{ij} = \int_{(s-1)\tau/S}^{s\tau/S} dt' \int_{(s'-1)\tau/S}^{t'} dt \sum_k \eta_{i,k} \eta_{j,k} \sin(\mu t) \sin(\mu t') \sin(\omega_k(t' - t)). \quad (7.11)$$

If we now arrange the segment amplitudes Ω_s into two vectors $\mathbf{\Omega}_{ij}$ and $\mathbf{\Omega}_{mn}$, one for each entangling pair (i, j) and (m, n) , we can now write our constraint equations from Equation 7.5 as

$$\begin{aligned} & \begin{pmatrix} \mathbf{C}^i \\ \mathbf{C}^j \\ \mathbf{C}^m \\ \mathbf{C}^n \end{pmatrix} \begin{pmatrix} \mathbf{\Omega}_{ij} \\ \mathbf{\Omega}_{mn} \end{pmatrix} = 0 \\ & \begin{pmatrix} \mathbf{\Omega}_{ij}^T & \mathbf{\Omega}_{mn}^T \end{pmatrix} \begin{pmatrix} \mathbf{D}^{ij} & 0 \\ 0 & 0 \end{pmatrix} \begin{pmatrix} \mathbf{\Omega}_{ij} \\ \mathbf{\Omega}_{mn} \end{pmatrix} = \chi_{ij}^{\text{ideal}} \\ & \begin{pmatrix} \mathbf{\Omega}_{ij}^T & \mathbf{\Omega}_{mn}^T \end{pmatrix} \begin{pmatrix} 0 & 0 \\ 0 & \mathbf{D}^{mn} \end{pmatrix} \begin{pmatrix} \mathbf{\Omega}_{ij} \\ \mathbf{\Omega}_{mn} \end{pmatrix} = \chi_{mn}^{\text{ideal}} \\ & \begin{pmatrix} \mathbf{\Omega}_{ij}^T & \mathbf{\Omega}_{mn}^T \end{pmatrix} \begin{pmatrix} 0 & \mathbf{D}^{\text{cross}} \\ 0 & 0 \end{pmatrix} \begin{pmatrix} \mathbf{\Omega}_{ij} \\ \mathbf{\Omega}_{mn} \end{pmatrix} = 0, \end{aligned} \quad (7.12)$$

where $\{\mathbf{C}^{\mathbf{i}}, \mathbf{C}^{\mathbf{j}}, \mathbf{C}^{\mathbf{m}}, \mathbf{C}^{\mathbf{n}}\}$ are the $S \times N$ spin-motion interaction matrices for each segment on each ion, $\{\mathbf{D}^{\mathbf{ij}}, \mathbf{D}^{\mathbf{mn}}\}$ are the two $S \times S$ spin-spin interaction matrices for each segment on the entangling ion pairs, and $\mathbf{D}^{\text{cross}} = \{\mathbf{D}^{\mathbf{im}}, \mathbf{D}^{\mathbf{in}}, \mathbf{D}^{\mathbf{jm}}, \mathbf{D}^{\mathbf{jn}}\}$ are the four $S \times S$ spin-spin interaction matrices for each segment on the crosstalk ion pairs. While the C -constraints are linear, the 6 D constraints on the spin-spin interaction terms are not. With multiple quadratic constraints and no evident guarantees the constraint matrices are positive or negative semidefinite, this is now a non-convex quadratically constrained quadratic program (QCQP). In the general case, this kind of problem is NP-hard. As a result, analytical approaches are intractable, and so we use optimization techniques to find solutions that fit the constraints in Equation 7.12 as well as possible.

7.1.1 Fidelity of Parallel XX Operations

Next, we calculate the fidelity of simultaneous XX gate operations as a function of the control parameters in Equation 7.5. The fidelity is given by

$$F_{||} = \langle \psi_{\text{init}} | U_{\text{ideal}}^\dagger \rho_r U_{\text{ideal}} | \psi_{\text{init}} \rangle, \quad (7.13)$$

where ρ_r is the density matrix for the experimental operation traced over the motion,

$$\rho_r = \text{Tr}_m \left[U_{\text{expt}} | \psi_{\text{init}} \rangle \langle \psi_{\text{init}} | U_{\text{expt}}^\dagger \right]. \quad (7.14)$$

For convenience of calculations, this derivation will be performed in the X basis.

Consequently, we use the initial state

[illegible]

and calculations of the unitary will be performed with σ_z instead of σ_x . The final fidelity is independent of the basis in which it is calculated. In matrix form, the

general parallel 2-qubit gate unitary from Equation 7.1 is

$$\begin{aligned}
 U_{\text{expt}} = & \begin{pmatrix} e^{i\Phi_{0000}} & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & e^{i\Phi_{0001}} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & e^{i\Phi_{0010}} & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & e^{i\Phi_{0011}} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & e^{i\Phi_{0100}} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & e^{i\Phi_{0101}} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & e^{i\Phi_{0110}} & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & e^{i\Phi_{0111}} \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix} \dots \\
 & \dots \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ e^{i\Phi_{1000}} & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & e^{i\Phi_{1001}} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & e^{i\Phi_{1010}} & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & e^{i\Phi_{1011}} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & e^{i\Phi_{1100}} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & e^{i\Phi_{1101}} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & e^{i\Phi_{1110}} & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & e^{i\Phi_{1111}} \end{pmatrix},
 \end{aligned}
 \tag{7.16}$$

where

$$\begin{aligned}
\Phi_{0000} &= \phi_i + \phi_j + \phi_m + \phi_n + \chi_{ij} + \chi_{im} + \chi_{in} + \chi_{jm} + \chi_{jn} + \chi_{mn} \\
\Phi_{0001} &= \phi_i + \phi_j + \phi_m - \phi_n + \chi_{ij} + \chi_{im} - \chi_{in} + \chi_{jm} - \chi_{jn} - \chi_{mn} \\
\Phi_{0010} &= \phi_i + \phi_j - \phi_m + \phi_n + \chi_{ij} - \chi_{im} + \chi_{in} - \chi_{jm} + \chi_{jn} - \chi_{mn} \\
\Phi_{0011} &= \phi_i + \phi_j - \phi_m - \phi_n + \chi_{ij} - \chi_{im} - \chi_{in} - \chi_{jm} - \chi_{jn} + \chi_{mn} \\
\Phi_{0100} &= \phi_i - \phi_j + \phi_m + \phi_n - \chi_{ij} + \chi_{im} + \chi_{in} - \chi_{jm} - \chi_{jn} + \chi_{mn} \\
\Phi_{0101} &= \phi_i - \phi_j + \phi_m - \phi_n - \chi_{ij} + \chi_{im} - \chi_{in} - \chi_{jm} + \chi_{jn} - \chi_{mn} \\
\Phi_{0110} &= \phi_i - \phi_j - \phi_m + \phi_n - \chi_{ij} - \chi_{im} + \chi_{in} + \chi_{jm} - \chi_{jn} - \chi_{mn} \\
\Phi_{0111} &= \phi_i - \phi_j - \phi_m - \phi_n - \chi_{ij} - \chi_{im} - \chi_{in} + \chi_{jm} + \chi_{jn} + \chi_{mn} \\
\Phi_{1000} &= -\phi_i + \phi_j + \phi_m + \phi_n - \chi_{ij} - \chi_{im} - \chi_{in} + \chi_{jm} + \chi_{jn} + \chi_{mn} \\
\Phi_{1001} &= -\phi_i + \phi_j + \phi_m - \phi_n - \chi_{ij} - \chi_{im} + \chi_{in} + \chi_{jm} - \chi_{jn} - \chi_{mn} \\
\Phi_{1010} &= -\phi_i + \phi_j - \phi_m + \phi_n - \chi_{ij} + \chi_{im} - \chi_{in} - \chi_{jm} + \chi_{jn} - \chi_{mn} \\
\Phi_{1011} &= -\phi_i + \phi_j - \phi_m - \phi_n - \chi_{ij} + \chi_{im} + \chi_{in} - \chi_{jm} - \chi_{jn} + \chi_{mn} \\
\Phi_{1100} &= -\phi_i - \phi_j + \phi_m + \phi_n + \chi_{ij} - \chi_{im} - \chi_{in} - \chi_{jm} - \chi_{jn} + \chi_{mn} \\
\Phi_{1101} &= -\phi_i - \phi_j + \phi_m - \phi_n + \chi_{ij} - \chi_{im} + \chi_{in} - \chi_{jm} + \chi_{jn} - \chi_{mn} \\
\Phi_{1110} &= -\phi_i - \phi_j - \phi_m + \phi_n + \chi_{ij} + \chi_{im} - \chi_{in} + \chi_{jm} - \chi_{jn} - \chi_{mn} \\
\Phi_{1111} &= -\phi_i - \phi_j - \phi_m - \phi_n + \chi_{ij} + \chi_{im} + \chi_{in} + \chi_{jm} + \chi_{jn} + \chi_{mn}. \tag{7.17}
\end{aligned}$$

In an ideally-executed pair of parallel XX gates, with all parameters set as in

Equation 7.5, we get

$$\begin{aligned}
U_{\text{ideal}} &= U(\alpha_{\{i,j,m,n\},k} = 0, \chi_{ij} = \chi_{ij}^{\text{ideal}}, \chi_{mn} = \chi_{mn}^{\text{ideal}}, \chi_{im} = \chi_{in} = \chi_{jm} = \chi_{jn} = 0) \\
&= \begin{pmatrix} e^{i\chi_{++}} & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & e^{i\chi_{+-}} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & e^{i\chi_{-+}} & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & e^{i\chi_{++}} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & e^{i\chi_{-+}} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & e^{i\chi_{--}} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & e^{i\chi_{--}} & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & e^{i\chi_{-+}} \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix} \dots \\
&\dots \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ e^{i\chi_{-+}} & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & e^{i\chi_{--}} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & e^{i\chi_{--}} & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & e^{i\chi_{-+}} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & e^{i\chi_{++}} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & e^{i\chi_{+-}} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & e^{i\chi_{+-}} & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & e^{i\chi_{++}} \end{pmatrix}, \quad (7.18)
\end{aligned}$$

where

$$\begin{aligned}
\chi_{++} &= \chi_{ij}^{\text{ideal}} + \chi_{mn}^{\text{ideal}} \\
\chi_{+-} &= \chi_{ij}^{\text{ideal}} - \chi_{mn}^{\text{ideal}} \\
\chi_{-+} &= -\chi_{ij}^{\text{ideal}} + \chi_{mn}^{\text{ideal}} \\
\chi_{--} &= -\chi_{ij}^{\text{ideal}} - \chi_{mn}^{\text{ideal}}.
\end{aligned} \tag{7.19}$$

Next, we calculate the density matrix ρ_r , given by Equation 7.14. As an intermediate step, we calculate the matrix A to be traced over. The first 3 columns of this 16×16 matrix are shown here:

$$\begin{aligned}
A &= U_{\text{expt}} |\psi_{\text{init}}\rangle \langle \psi_{\text{init}}| U_{\text{expt}}^\dagger \\
&= \frac{1}{16} \begin{pmatrix}
1 & e^{2i(\phi_n + \chi_{in} + \chi_{jn} + \chi_{mn})} & e^{2i(\phi_m + \chi_{im} + \chi_{jm} + \chi_{mn})} \\
e^{-2i(\phi_n + \chi_{in} + \chi_{jn} + \chi_{mn})} & 1 & e^{2i(\phi_m - \phi_n + \chi_{im} - \chi_{in} + \chi_{jm} - \chi_{jn})} \\
e^{-2i(\phi_m + \chi_{im} + \chi_{jm} + \chi_{mn})} & e^{-2i(\phi_m - \phi_n + \chi_{im} - \chi_{in} + \chi_{jm} - \chi_{jn})} & 1 \\
e^{-2i(\phi_m + \phi_n + \chi_{im} + \chi_{in} + \chi_{jm} + \chi_{jn})} & e^{-2i(\phi_m + \chi_{im} + \chi_{jm} - \chi_{mn})} & e^{-2i(\phi_n + \chi_{in} + \chi_{jn} - \chi_{mn})} \\
e^{-2i(\phi_j + \chi_{ij} + \chi_{jm} + \chi_{jn})} & e^{-2i(\phi_j - \phi_n + \chi_{ij} - \chi_{in} + \chi_{jm} - \chi_{mn})} & e^{-2i(\phi_j - \phi_m + \chi_{ij} - \chi_{im} + \chi_{jn} - \chi_{mn})} \\
e^{-2i(\phi_j + \phi_n + \chi_{ij} + \chi_{in} + \chi_{jm} + \chi_{mn})} & e^{-2i(\phi_j + \chi_{ij} + \chi_{jm} - \chi_{jn})} & e^{-2i(\phi_j - \phi_m + \phi_n + \chi_{ij} - \chi_{im} + \chi_{in})} \\
e^{-2i(\phi_j + \phi_m + \chi_{ij} + \chi_{im} + \chi_{jn} + \chi_{mn})} & e^{-2i(\phi_j + \phi_m - \phi_n + \chi_{ij} + \chi_{im} - \chi_{in})} & e^{-2i(\phi_j + \chi_{ij} - \chi_{jm} + \chi_{jn})} \\
e^{-2i(\phi_j + \phi_m + \phi_n + \chi_{ij} + \chi_{im} + \chi_{in})} & e^{-2i(\phi_j + \phi_m + \chi_{ij} + \chi_{im} - \chi_{jn} - \chi_{mn})} & e^{-2i(\phi_j + \phi_n + \chi_{ij} + \chi_{in} - \chi_{jm} - \chi_{mn})} \\
e^{-2i(\phi_i + \chi_{ij} + \chi_{im} + \chi_{in})} & e^{-2i(\phi_i - \phi_n + \chi_{ij} + \chi_{im} - \chi_{jn} - \chi_{mn})} & e^{-2i(\phi_i - \phi_m + \chi_{ij} + \chi_{in} - \chi_{jm} - \chi_{mn})} \\
e^{-2i(\phi_i + \phi_n + \chi_{ij} + \chi_{im} + \chi_{jn} + \chi_{mn})} & e^{-2i(\phi_i + \chi_{ij} + \chi_{im} - \chi_{in})} & e^{-2i(\phi_i - \phi_m + \phi_n + \chi_{ij} - \chi_{jm} + \chi_{jn})} \\
e^{-2i(\phi_i + \phi_m + \chi_{ij} + \chi_{in} + \chi_{jm} + \chi_{mn})} & e^{-2i(\phi_i + \phi_m - \phi_n + \chi_{ij} + \chi_{jm} - \chi_{jn})} & e^{-2i(\phi_i + \chi_{ij} - \chi_{im} + \chi_{in})} \\
e^{-2i(\phi_i + \phi_m + \phi_n + \chi_{ij} + \chi_{jm} + \chi_{jn})} & e^{-2i(\phi_i + \phi_m + \chi_{ij} - \chi_{in} + \chi_{jm} - \chi_{mn})} & e^{-2i(\phi_i + \phi_n + \chi_{ij} - \chi_{im} + \chi_{jn} - \chi_{mn})} \\
e^{-2i(\phi_i + \phi_j + \chi_{im} + \chi_{in} + \chi_{jm} + \chi_{jn})} & e^{-2i(\phi_i + \phi_j - \phi_n + \chi_{im} + \chi_{jm} - \chi_{mn})} & e^{-2i(\phi_i + \phi_j - \phi_m + \chi_{in} + \chi_{jn} - \chi_{mn})} \\
e^{-2i(\phi_i + \phi_j + \phi_n + \chi_{im} + \chi_{jm} + \chi_{mn})} & e^{-2i(\phi_i + \phi_j + \chi_{im} - \chi_{in} + \chi_{jm} - \chi_{jn})} & e^{-2i(\phi_i + \phi_j - \phi_m + \phi_n)} \\
e^{-2i(\phi_i + \phi_j + \phi_m + \chi_{in} + \chi_{jn} + \chi_{mn})} & e^{-2i(\phi_i + \phi_j + \phi_m - \phi_n)} & e^{-2i(\phi_i + \phi_j - \chi_{im} + \chi_{in} - \chi_{jm} + \chi_{jn})} \\
e^{-2i(\phi_i + \phi_j + \phi_m + \phi_n)} & e^{-2i(\phi_i + \phi_j + \phi_m - \chi_{in} - \chi_{jn} - \chi_{mn})} & e^{-2i(\phi_i + \phi_j + \phi_n - \chi_{im} - \chi_{jm} - \chi_{mn})}
\end{pmatrix} \\
&\dots \quad \text{etc.} \Big). \tag{7.20}
\end{aligned}$$

Now we find the trace over the motion for this matrix A to find the density matrix.

Since the spin-spin *chi* terms have no motional components, they are unaffected by the trace operation. The scalars on the diagonal are similarly unaffected. Hence, we only have to worry about the spin-motion $\phi_{\{i,j,m,n\}}$ terms. We further note that the displacement operator D is

$$\begin{aligned} D(\alpha_i^k(\tau)) &= e^{(\alpha_i^k(\tau)\hat{a}_k^\dagger - \alpha_i^{k*}(\tau)\hat{a}_k)} \\ &= e^{(|\alpha_i^k(\tau)|^2/2)} e^{(-\alpha_i^{k*}(\tau)\hat{a}_k)} e^{(\alpha_i^k(\tau)\hat{a}_k^\dagger)} \end{aligned} \quad (7.21)$$

for a given parameter $\alpha_{\{i,j,m,n\}}^k$. Consequently, as an example, tracing over the term in row 4, column 1 of Equation 7.20 can be written as

$$\begin{aligned} \text{Tr}_m [e^{-2i(\phi_m + \phi_n + \chi_{im} + \chi_{in} + \chi_{jm} + \chi_{jn})}] &= e^{-2i(\chi_{im} + \chi_{in} + \chi_{jm} + \chi_{jn})} \text{Tr}_m [e^{-2i(\phi_m + \phi_n)}] \\ &= e^{-2i(\chi_{im} + \chi_{in} + \chi_{jm} + \chi_{jn})} \prod_k \text{Tr}_m [D(2\alpha_m^k + 2\alpha_n^k)] . \end{aligned} \quad (7.22)$$

In Fock space, a thermal state for the k^{th} motional mode in a linear ion chain can be written as

$$\rho_{\text{therm}} = (1 - e^{\hbar\omega_k/k_B T}) \sum_{n=0}^{\infty} e^{-n\hbar\omega_k/k_B T} |n\rangle \langle n| , \quad (7.23)$$

where n is the phonon number. For a generic displacement operator $D(\alpha)$ using Equation 7.21 and using the properties of the raising and lowering operators \hat{a}^\dagger and

\hat{a} , we can write

$$\begin{aligned}
\text{Tr}_m [D(\alpha)] &= \text{Tr} [D(\alpha)\rho_{\text{therm}}] \\
&= (1 - e^{\hbar\omega/k_B T}) e^{|\alpha|^2/2} \sum_{n=0}^{\infty} e^{-n\hbar\omega/k_B T} \text{Tr} \left[e^{-\alpha^* \hat{a}} e^{\alpha \hat{a}^\dagger} |n\rangle \langle n| \right] \\
&= (1 - e^{\hbar\omega/k_B T}) e^{|\alpha|^2/2} \sum_{n=0}^{\infty} e^{-n\hbar\omega/k_B T} \text{Tr} \left[\sum_{p=0}^{\infty} \frac{1}{p!} (-\alpha^* \hat{a})^p \sum_{q=0}^{\infty} \frac{1}{q!} (\alpha \hat{a}^\dagger)^q |n\rangle \langle n| \right] \\
&= (1 - e^{\hbar\omega/k_B T}) e^{|\alpha|^2/2} \sum_{n=0}^{\infty} e^{-n\hbar\omega/k_B T} \text{Tr} \left[\sum_{p=0}^{\infty} \frac{(-1)^p |\alpha|^{2p}}{(p!)^2} \hat{a}^p (\hat{a}^\dagger)^p |n\rangle \langle n| \right] \\
&= (1 - e^{\hbar\omega/k_B T}) e^{|\alpha|^2/2} \sum_{n=0}^{\infty} e^{-n\hbar\omega/k_B T} \times \\
&\quad \dots \sum_{p=0}^{\infty} \frac{(-1)^p |\alpha|^{2p}}{(p!)^2} (n+1)(n+2) \times \dots \times (n+m) \\
&= e^{-\frac{|\alpha|^2}{2} \coth(\hbar\omega/2k_B T)}
\end{aligned} \tag{7.24}$$

where we note that the trace is only nonzero for $q = p$. So continuing our example

from Equation 7.22, we can now write

$$\begin{aligned}
& \text{Tr}_m \left[e^{-2i(\phi_m + \phi_n + \chi_{im} + \chi_{in} + \chi_{jm} + \chi_{jn})} \right] \\
&= e^{-2i(\chi_{im} + \chi_{in} + \chi_{jm} + \chi_{jn})} \prod_k \text{Tr}_m \left[D(2\alpha_m^k + 2\alpha_n^k) \right] \\
&= e^{-2i(\chi_{im} + \chi_{in} + \chi_{jm} + \chi_{jn})} \prod_k e^{\left(-\frac{|2\alpha_m^k + 2\alpha_n^k|^2}{2} \coth(\hbar\omega_k/2k_B T) \right)} \\
&= e^{-2i(\chi_{im} + \chi_{in} + \chi_{jm} + \chi_{jn})} e^{\left(-\frac{1}{2} \sum_k |2\alpha_m^k + 2\alpha_n^k|^2 \coth(\hbar\omega_k/2k_B T) \right)} \\
&= e^{-2i(\chi_{im} + \chi_{in} + \chi_{jm} + \chi_{jn})} e^{\left(-\frac{1}{2} \sum_k \beta_k |2\alpha_m^k + 2\alpha_n^k|^2 \right)} \\
&= e^{-2i(\chi_{im} + \chi_{in} + \chi_{jm} + \chi_{jn})} \Gamma_{00++}. \tag{7.25}
\end{aligned}$$

Here, to better abbreviate the representation, we define a set of parameters,

$$\Gamma_{A_i A_j A_m A_n} = \exp \left(-\frac{1}{2} \sum_k \beta_k |2(A_i \alpha_{i,k} + A_j \alpha_{j,k} + A_m \alpha_{m,k} + A_n \alpha_{n,k})|^2 \right), \tag{7.26}$$

where the parameters $\{A_i, A_j, A_m, A_n\}$ can be $\{0, \pm 1\}$ and are indicated as $\{0 \rightarrow 0, +1 \rightarrow +, -1 \rightarrow -\}$. The inverse mode temperature β_k is

$$\beta_k = \coth \left(\frac{\hbar\omega_k}{k_B T} \right) = \coth \left[\frac{1}{2} \ln \left(1 + \frac{1}{\bar{n}_k} \right) \right], \tag{7.27}$$

where \bar{n}_k is the average phonon number in the k^{th} mode. By applying the trace operation to all elements of Equation 7.20, we get the final density matrix.

Plugging in all values to the fidelity equation from Equation 7.13 and solving,

we derive the fidelity,

$$\begin{aligned}
F_{\parallel} \left(\alpha_{\{i,j,m,n\},k}, \chi_{ij}, \chi_{ij}^{\text{ideal}}, \chi_{mn}, \chi_{mn}^{\text{ideal}}, \chi_{im}, \chi_{in}, \chi_{jm}, \chi_{jn} \right) = \\
\frac{1}{128} (8 + \Gamma_{+---} + \Gamma_{+--+} + \Gamma_{+-++} + \Gamma_{-+++} + \Gamma_{++--} + \Gamma_{++-+} + \Gamma_{++++} + \Gamma_{++++} \\
+ 2(\Gamma_{0+--} + \Gamma_{+000}) \cos[2(\Delta\chi_{ij} - \chi_{im} - \chi_{in})] \\
+ 2(\Gamma_{0++-} + \Gamma_{+000}) \cos[2(\Delta\chi_{ij} + \chi_{im} - \chi_{in})] \\
+ 2(\Gamma_{0+-+} + \Gamma_{+000}) \cos[2(\Delta\chi_{ij} - \chi_{im} + \chi_{in})] \\
+ 2(\Gamma_{0+++} + \Gamma_{+000}) \cos[2(\Delta\chi_{ij} + \chi_{im} + \chi_{in})] \\
+ 2(\Gamma_{0+00} + \Gamma_{+0--}) \cos[2(\Delta\chi_{ij} - \chi_{jm} - \chi_{jn})] \\
+ 2(\Gamma_{00++} + \Gamma_{+-00}) \cos[2(\chi_{im} + \chi_{in} - \chi_{jm} - \chi_{jn})] \\
+ 2(\Gamma_{0+00} + \Gamma_{+0+-}) \cos[2(\Delta\chi_{ij} + \chi_{jm} - \chi_{jn})] \\
+ 2(\Gamma_{00+-} + \Gamma_{++00}) \cos[2(\chi_{im} - \chi_{in} + \chi_{jm} - \chi_{jn})] \\
+ 2(\Gamma_{0+00} + \Gamma_{+0-+}) \cos[2(\Delta\chi_{ij} - \chi_{jm} + \chi_{jn})] \\
+ 2(\Gamma_{00+-} + \Gamma_{+-00}) \cos[2(\chi_{im} - \chi_{in} - \chi_{jm} + \chi_{jn})] \\
+ 2(\Gamma_{0+00} + \Gamma_{+0++}) \cos[2(\Delta\chi_{ij} + \chi_{jm} + \chi_{jn})] \\
+ 2(\Gamma_{00++} + \Gamma_{++00}) \cos[2(\chi_{im} + \chi_{in} + \chi_{jm} + \chi_{jn})] \\
+ 2(\Gamma_{00+0} + \Gamma_{+-0+}) \cos[2(\chi_{im} - \chi_{jm} + \Delta\chi_{mn})] \\
+ 2(\Gamma_{0+0-} + \Gamma_{+0-0}) \cos[2(\Delta\chi_{ij} - \chi_{in} - \chi_{jm} + \Delta\chi_{mn})] \\
+ 2(\Gamma_{00+0} + \Gamma_{++0+}) \cos[2(\chi_{im} + \chi_{jm} + \Delta\chi_{mn})] \\
+ 2(\Gamma_{0+0+} + \Gamma_{+0+0}) \cos[2(\Delta\chi_{ij} + \chi_{in} + \chi_{jm} + \Delta\chi_{mn})] \\
+ 2(\Gamma_{0+-0} + \Gamma_{+00-}) \cos[2(\Delta\chi_{ij} - \chi_{im} - \chi_{jn} + \Delta\chi_{mn})]
\end{aligned}$$

$$\begin{aligned}
& + 2 (\Gamma_{000+} + \Gamma_{+-+0}) \cos [2 (\chi_{in} - \chi_{jn} + \Delta\chi_{mn})] \\
& + 2 (\Gamma_{0++0} + \Gamma_{+00+}) \cos [2 (\Delta\chi_{ij} + \chi_{im} + \chi_{jn} + \Delta\chi_{mn})] \\
& + 2 (\Gamma_{000+} + \Gamma_{+++0}) \cos [2 (\chi_{in} + \chi_{jn} + \Delta\chi_{mn})] \\
& + 2 (\Gamma_{00+0} + \Gamma_{+-0-}) \cos [2 (\chi_{im} - \chi_{jm} - \Delta\chi_{mn})] \\
& + 2 (\Gamma_{0+0+} + \Gamma_{+0-0}) \cos [2 (\Delta\chi_{ij} + \chi_{in} - \chi_{jm} - \Delta\chi_{mn})] \\
& + 2 (\Gamma_{00+0} + \Gamma_{++0-}) \cos [2 (\chi_{im} + \chi_{jm} - \Delta\chi_{mn})] \\
& + 2 (\Gamma_{0+0-} + \Gamma_{+0+0}) \cos [2 (\Delta\chi_{ij} - \chi_{in} + \chi_{jm} - \Delta\chi_{mn})] \\
& + 2 (\Gamma_{0++0} + \Gamma_{+00-}) \cos [2 (\Delta\chi_{ij} + \chi_{im} - \chi_{jn} - \Delta\chi_{mn})] \\
& + 2 (\Gamma_{000+} + \Gamma_{+--0}) \cos [2 (\chi_{in} - \chi_{jn} - \Delta\chi_{mn})] \\
& + 2 (\Gamma_{0+-0} + \Gamma_{+00+}) \cos [2 (\Delta\chi_{ij} - \chi_{im} + \chi_{jn} - \Delta\chi_{mn})] \\
& + 2 (\Gamma_{000+} + \Gamma_{++-0}) \cos [2 (\chi_{in} + \chi_{jn} - \Delta\chi_{mn})] .
\end{aligned} \tag{7.28}$$

Here,

$$\begin{aligned}
\Delta\chi_{ij} &= \chi_{ij} - \chi_{ij}^{\text{ideal}} \\
\Delta\chi_{mn} &= \chi_{mn} - \chi_{mn}^{\text{ideal}} .
\end{aligned} \tag{7.29}$$

Plugging in the ideal-case parameters, where $\alpha_{\{i,j,m,n\},k}(\tau) = 0$, $\chi_{im} = \chi_{in} = \chi_{jm} = \chi_{jn} = 0$, $\chi_{ij} = \chi_{ij}^{\text{ideal}}$, and $\chi_{mn} = \chi_{mn}^{\text{ideal}}$, we indeed get $F_{||} = 1$.

The Mathematica notebook for this derivation is available upon request; it includes the full matrix for Equation 7.20 and the final density matrix ρ_r .

See also Appendix D.3-D.5 of [108]), which provides a very nice treatment of

the fidelity calculation for a 2-ion XX gate.

7.2 Implementation

To find a sequence of pulses that would give a Rabi frequency modulation optimally satisfying the above constraints, an optimization scheme was implemented. The built-in MATLAB unconstrained multivariable optimization function “fminunc” was used, where the objective function included the above constraints on α and χ parameters, as well as a term to minimize power. (The fidelity for a given solution was not calculated until after the solution was optimized, as calculating the fidelity with Equation 7.28 slowed the solution-finding process considerably; simply minimizing the parameters in question was sufficient.) Sequences were calculated for a gate time of $\tau_{gate} = 250 \mu s$, which is comparable to the standard 2-qubit XX gates already used on the experiment (see Section 3.2), and for a range of detunings μ . This generated a selection of solutions, which were tested on the experimental setup; the solution generating the highest-quality gate using the least amount of power was chosen. The quick screening method used to determine comparative gate quality consisted of running the gate and looking at the populations in the states $|00\rangle$, $|01\rangle$, $|10\rangle$, and $|11\rangle$ for each of the 6 pairs available from the 4 ions in use. The entangled pairs should show populations at 50% in each of $|00\rangle$ and $|11\rangle$ with no population in $|01\rangle$ or $|10\rangle$, whereas the crosstalk pairs should have 25% population in each of $|00\rangle$, $|01\rangle$, $|10\rangle$, and $|11\rangle$. The amount of odd parity ($|01\rangle$ and $|10\rangle$) population for the two entangled pairs provided a strong heuristic for gate quality; on

the experiment, high quality gates will show odd parity populations of 0-2% on the entangled pairs (before correcting for SPAM errors.)

Experimental gates were found for 6 ion pair combinations: (1,4) and (2,5); (1,2) and (3,4); (1,5) and (2,4); (1,4) and (2,3); (1,3) and (2,5); (1,2) and (4,5). For each set of parallel 2-qubit gates, Figures 7.1-7.6 show the pulse sequence applied to each entangled pair to construct the gate, as well as the trajectories in phase space of each mode-pair interaction. Within each figure, the phase space plots are plotted on axes of the same size, so relative engagement of each mode is shown. The phase space trajectories start out at the blue open circle and follow the path to end at the green star. Most phase space trajectories do end up back at the origin (indicating the corresponding $\alpha = 0$), but since these solutions were generated with a numerical optimization procedure, not all of them are perfect. The 5 transverse motional modes in this 5 ion chain have sideband frequencies $\nu_x = \{3.045, 3.027, 3.005, 2.978, 2.946\}$ MHz, and the plot captions include information on the detuning value used for that solution; the phase space trajectories for each set show that the mode interactions closest to the selected detuning exhibit the greatest activity, and contribute the most to the final spin-spin entanglement by enclosing more of phase space.

Negative-amplitude pulses are implemented by inverting the phase of the control signal. This capability was crucial, as changing the control signal phase allows the entangling pairs to continue accumulating entanglement while cancelling out accumulated entanglement with cross-talk pairs. To that end, the initial guess used for the gates in the optimization protocol was that one pair would have two positive “humps” (since regular XX gate pulse shapes frequently featured symmetric

increasing and then decreasing segment amplitudes), and the other pair would have one positive “hump” and one negative “hump,” the idea being the two gates would perform half of their entanglement process each in their first humps, and then perform the second half of the entanglement during the second humps while the relative change in phase on pair 2 cancelled out the accumulated crosstalk entanglement. The pulse shapes in Figure 7.1 provide a good example of this; most of the other pulse solutions feature similar patterns with some kind of symmetry, increasing and decreasing segment amplitudes, and phase flips on one pair to cancel out crosstalk entanglement.

7.3 Experimental Results

Here, we present experimental results from implementing parallel 2-qubit entangling gates on several ion pair selections. Fidelities are calculated by performing the parallel gates followed by an analysis pulse, then using the calculated parity to determine the fidelity (see Section 7.3.1.) The analysis pulses are rotations using the SK1 composite pulse for increased robustness against errors in the rotation angle [109, 110]. For the four ions involved in each gates, the parity analysis was performed for all 6 possible pairs within the set, allowing for analysis of the 2 entangled ion pairs as well as the 4 crosstalk pairs. Parity curves are shown in Figures 7.7 - 7.12. Entangling gate fidelities were typically between 96-99%, with crosstalk of a few percent. An exception is the (1,2), (4,5) gate, for which the (4,5) gate has a fidelity of 91% (Figure 7.12); looking at its phase space closure diagram in Figure

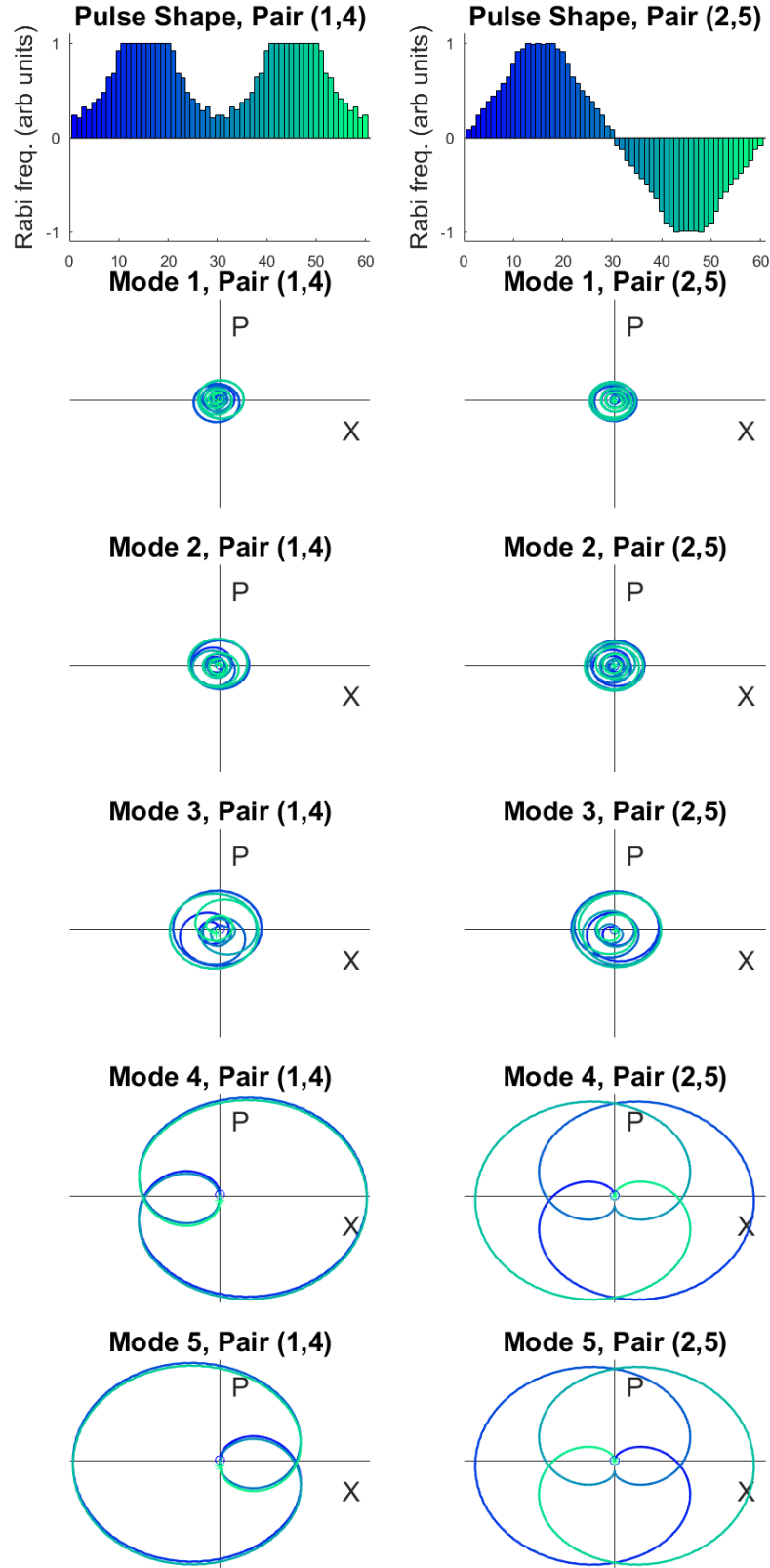


Figure 7.1: Pulse shapes and phase space trajectories for parallel XX gates on ions (1,4) and (2,5), with detuning $\mu = 2.962$ MHz and theoretical fidelity 99.63%.

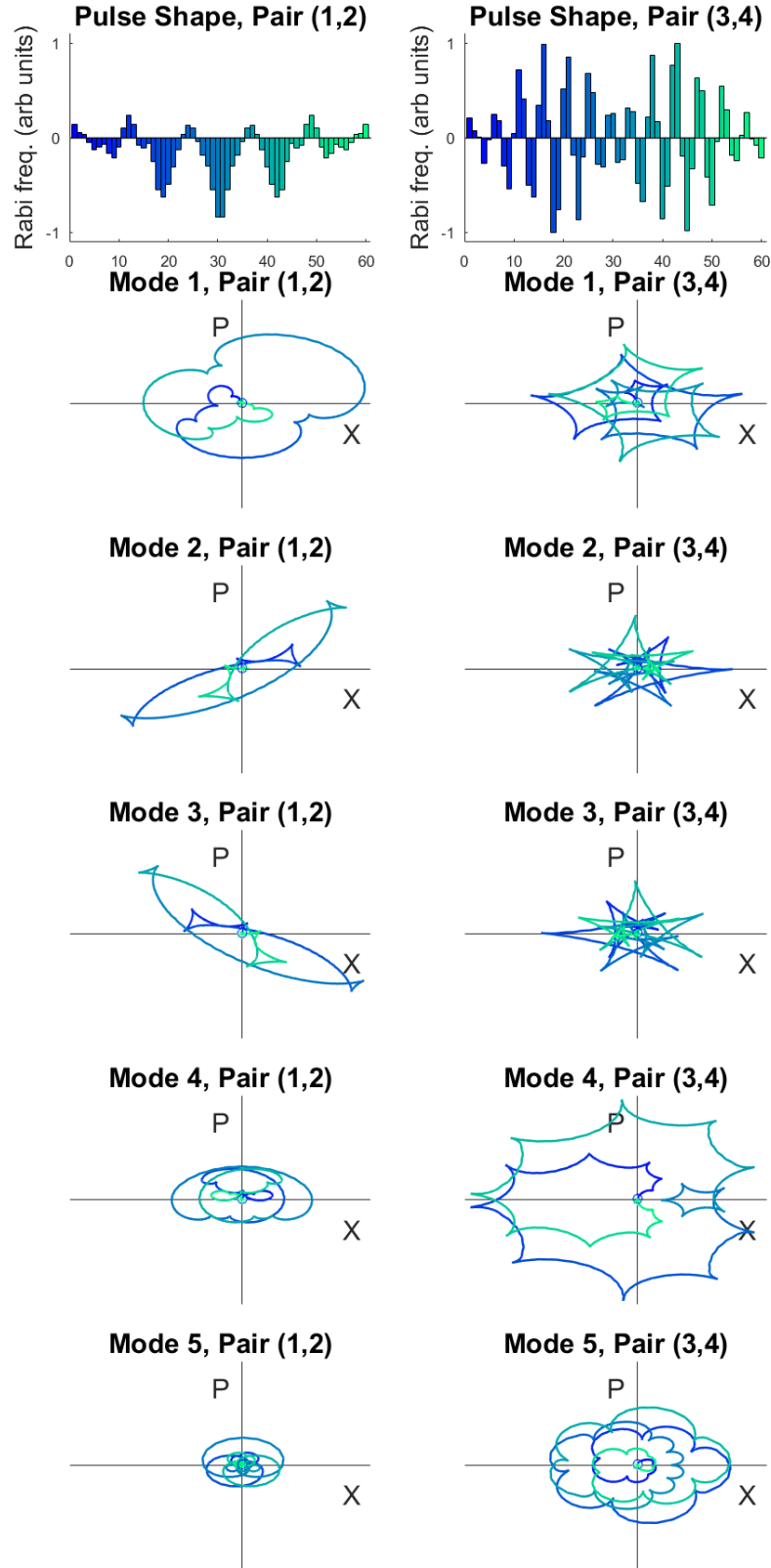


Figure 7.2: Pulse shapes and phase space trajectories for parallel XX gates on ions (1,2) and (3,4), with detuning $\mu = 3.016$ MHz and theoretical fidelity 99.97%.

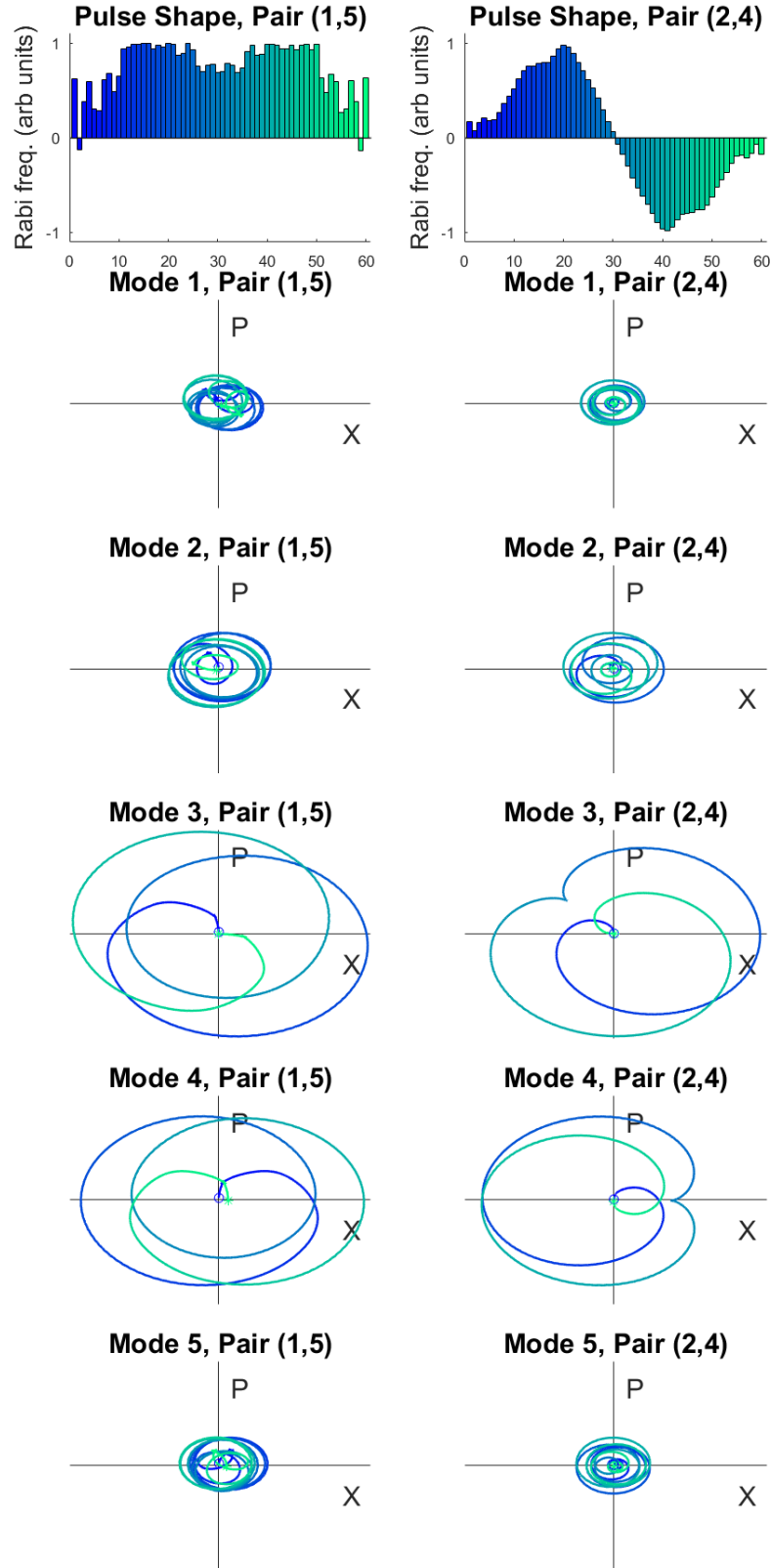


Figure 7.3: Pulse shapes and phase space trajectories for parallel XX gates on ions (1,5) and (2,4), with detuning $\mu = 2.992$ MHz and theoretical fidelity 99.75%.

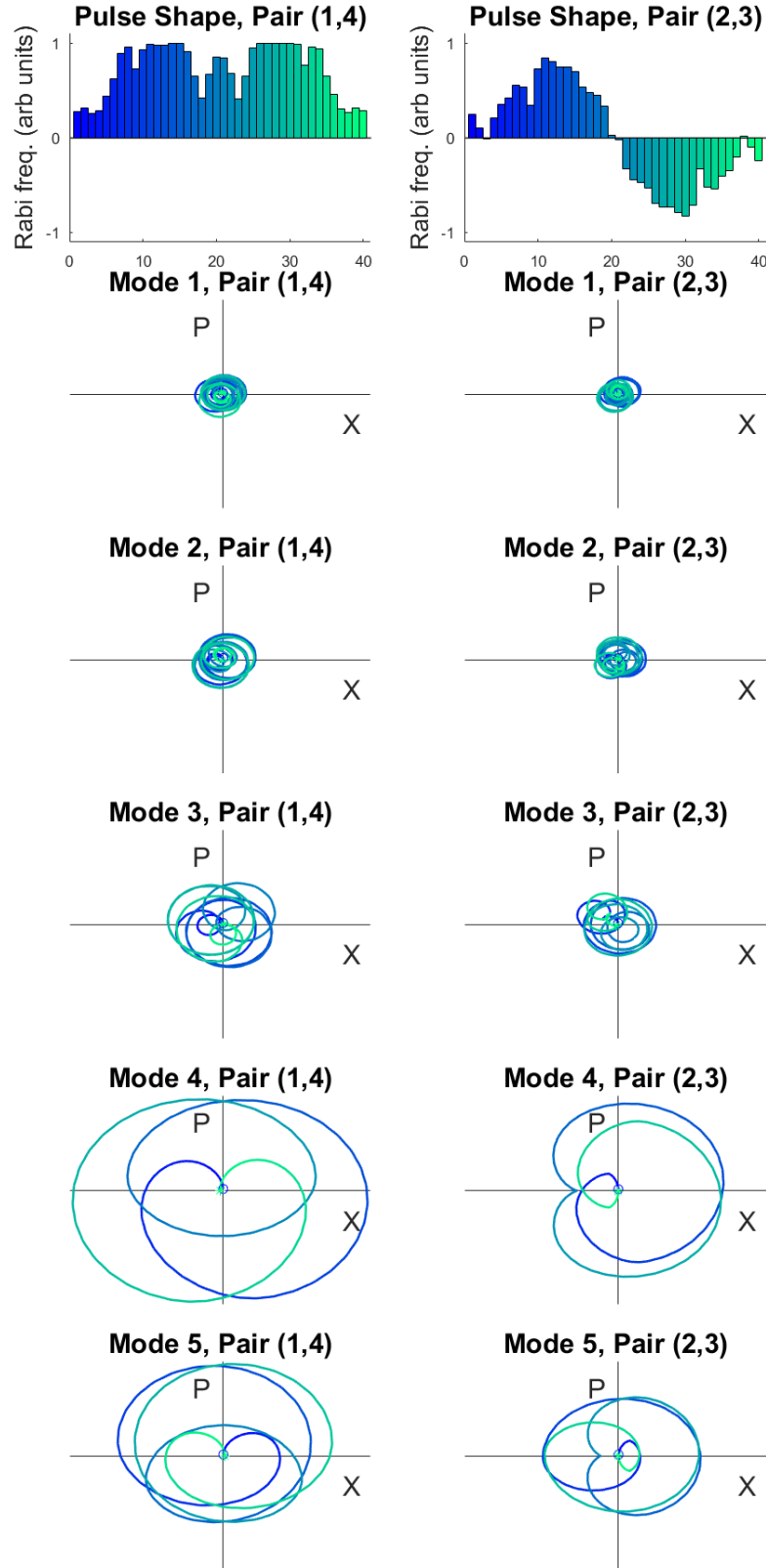


Figure 7.4: Pulse shapes and phase space trajectories for parallel XX gates on ions (1,4) and (2,3), with detuning $\mu = 2.964$ MHz and theoretical fidelity 99.85%.

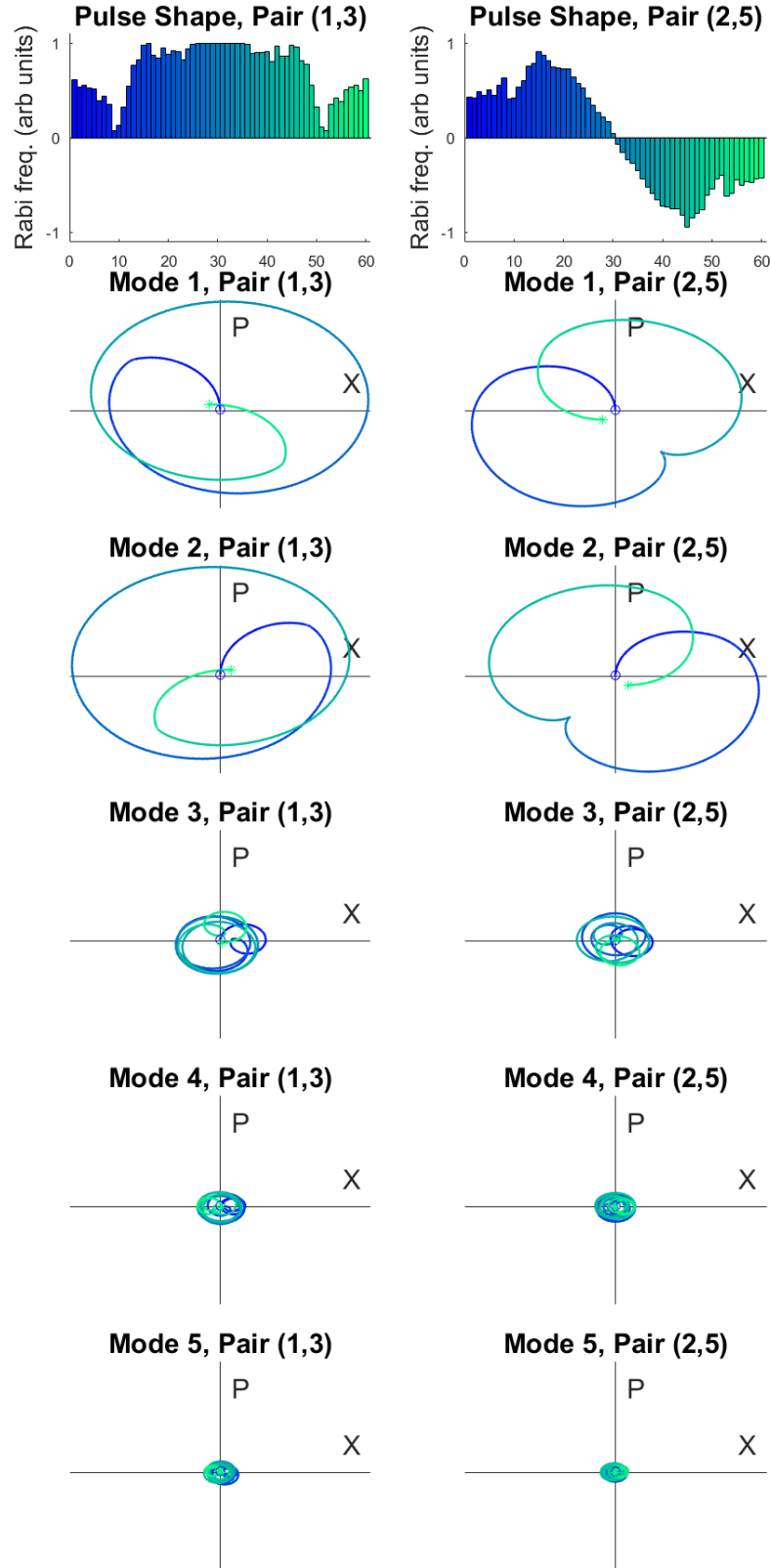


Figure 7.5: Pulse shapes and phase space trajectories for parallel XX gates on ions (1,3) and (2,5), with detuning $\mu = 3.036$ MHz and theoretical fidelity 98.97%.

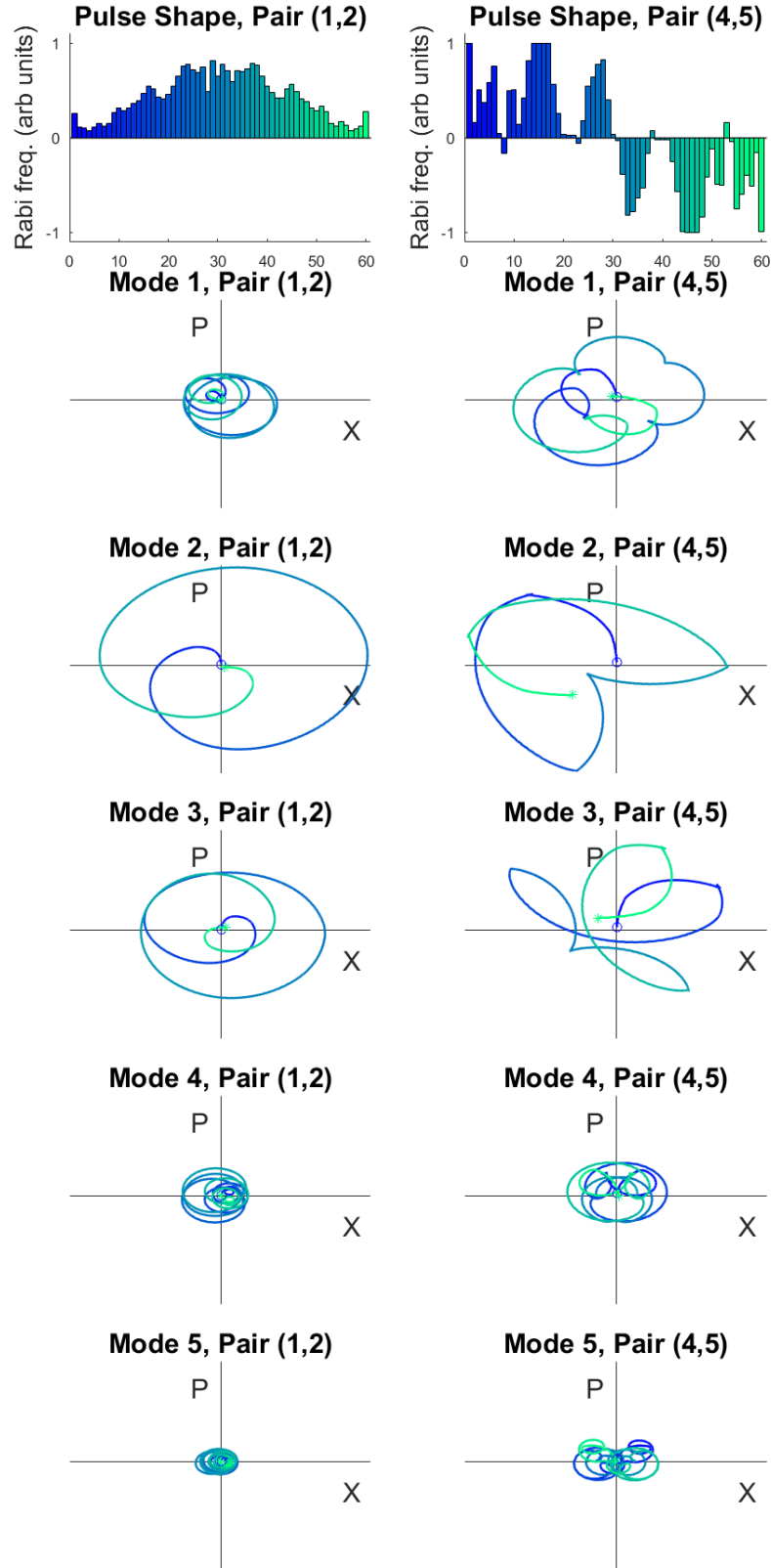


Figure 7.6: Pulse shapes and phase space trajectories for parallel XX gates on ions (1,2) and (4,5), with detuning $\mu = 3.018$ MHz and theoretical fidelity 95.91%.

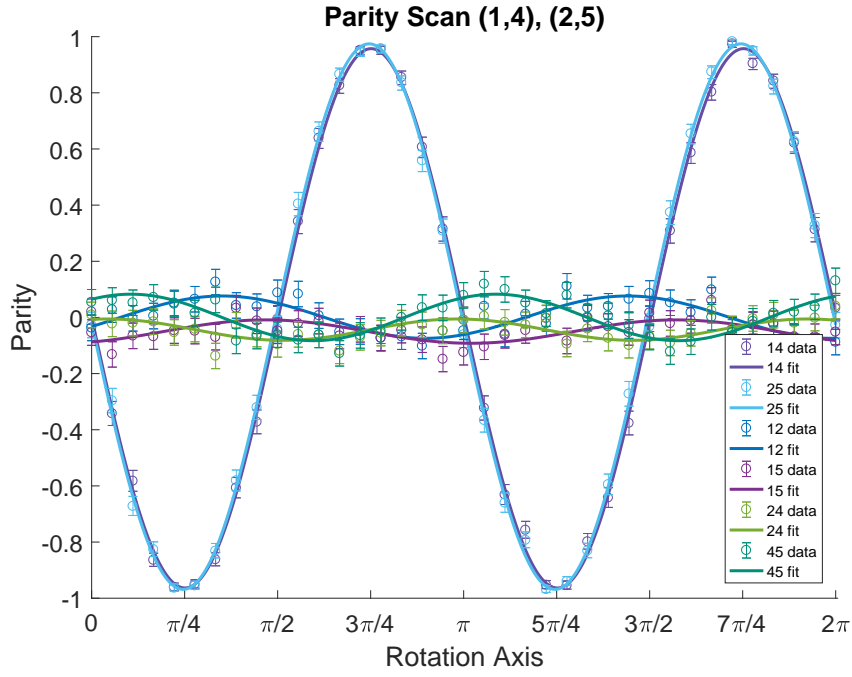


Figure 7.7: Parity curve for parallel XX gates on ions (1,4) and (2,5), yielding fidelities of 96.5(4)% and 97.8(3)% on the respective entangled pairs, with an average crosstalk error of 3.6(3)% and corrected for 3% SPAM errors.

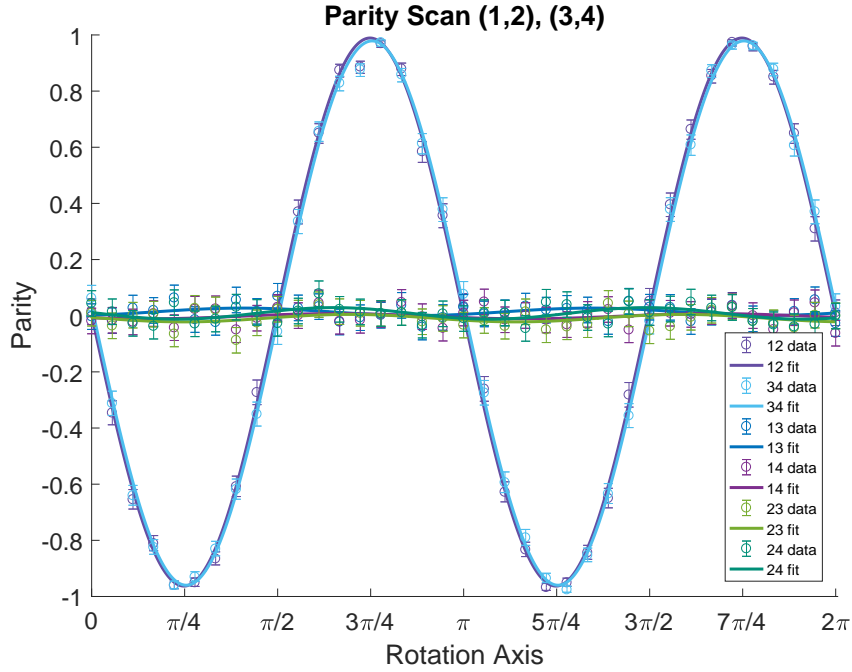


Figure 7.8: Parity curve for parallel XX gates on ions (1,2) and (3,4), yielding fidelities of 98.4(3)% and 97.7(3)% on the respective entangled pairs, with an average crosstalk error of 0.6(3)% and corrected for 3% SPAM errors.

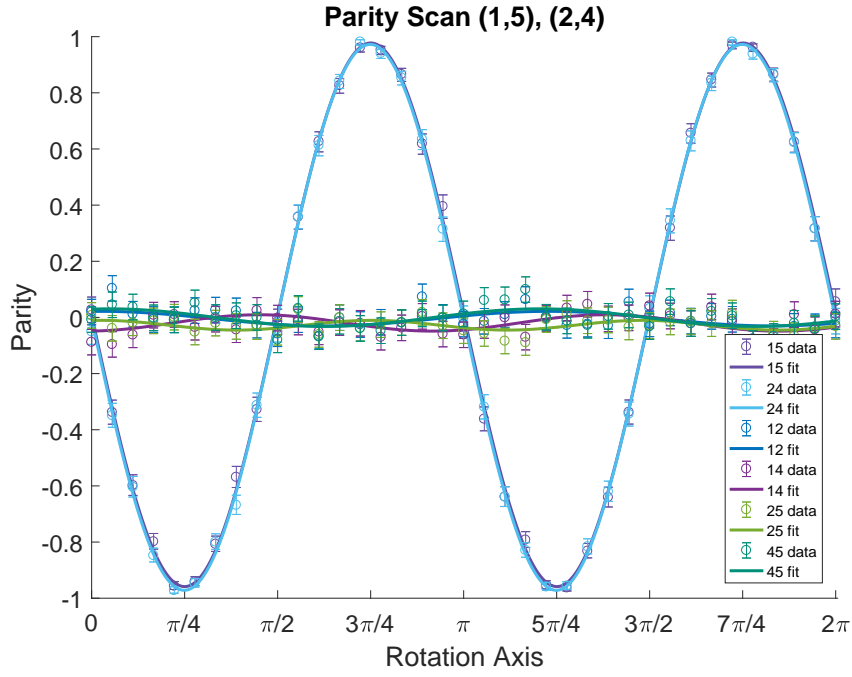


Figure 7.9: Parity curve for parallel XX gates on ions (1,5) and (2,4), yielding fidelities of 96.8(3)% and 98.1(2)% on the respective entangled pairs, with an average crosstalk error of 1.7(3)% and corrected for 2% SPAM errors.

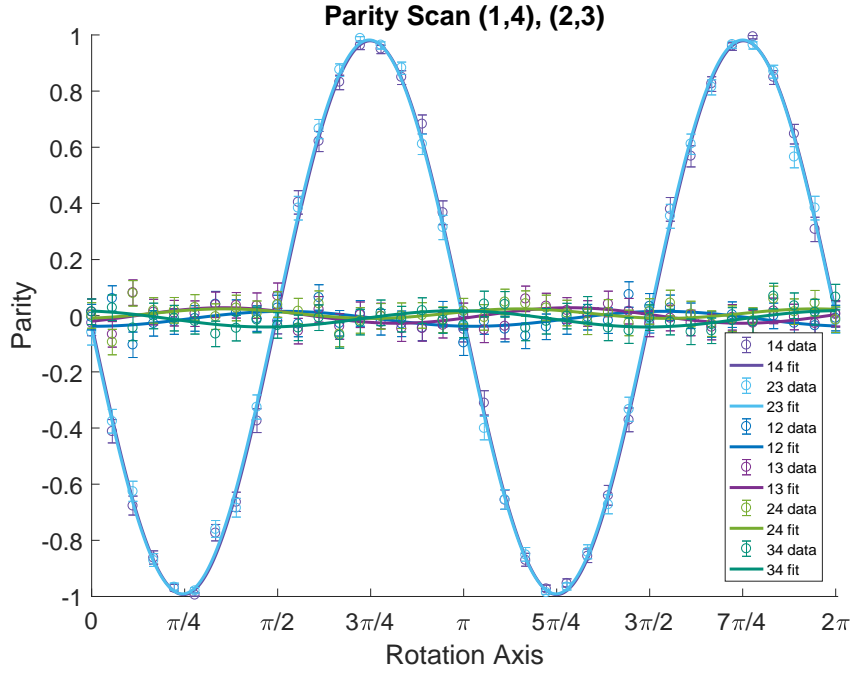


Figure 7.10: Parity curve for parallel XX gates on ions (1,4) and (2,3), yielding fidelities of 98.8(3)% and 99.0(3)% on the respective entangled pairs, with an average crosstalk error of 1.4(3)% and corrected for <1% SPAM errors.

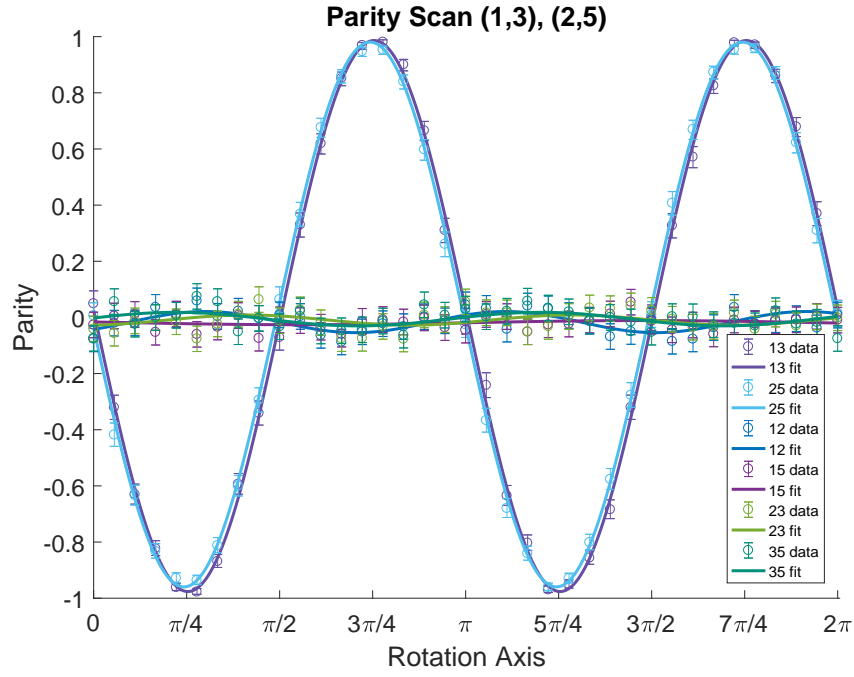


Figure 7.11: Parity curve for parallel XX gates on ions (1,3) and (2,5), yielding fidelities of 98.3(3)% and 97.5(2)% on the respective entangled pairs, with an average crosstalk error of 0.8(4)% and corrected for 3% SPAM errors.

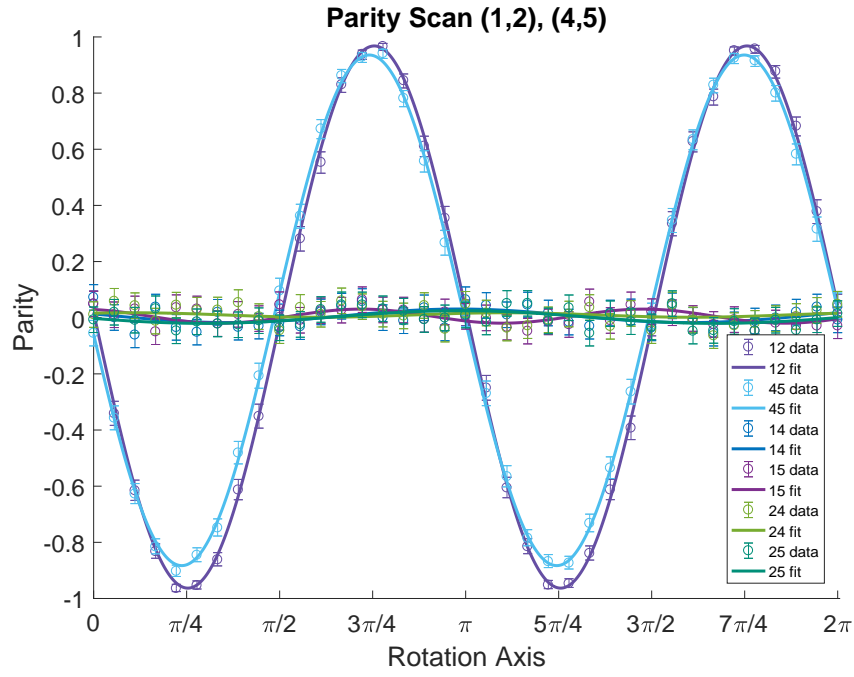


Figure 7.12: Parity curve for parallel XX gates on ions (1,2) and (4,5), yielding fidelities of 97.2(3)% and 91.9(3)% on the respective entangled pairs, with an average crosstalk error of 0.9(3)% and corrected for 2% SPAM errors.

Parallel Gate Pairs	$R_{ }$, Pair 1	$R_{ }$, Pair 2
(1,4) and (2,5)	4.3	1.8
(1,2) and (3,4)	7.9	5.0
(1,5) and (2,4)	2.1	1.6
(1,4) and (2,3)	4.3	3.8
(1,3) and (2,5)	0.9	1.5
(1,2) and (4,5)	2.2	2.2

Table 7.1: For each pair of parallel XX gates implemented, we compare the power required to perform each component XX with its corresponding stand-alone 2-qubit XX gate by calculating the power ratio $R_{||}$.

7.6, however, it is clear that it did not close very well, the likely source of the low fidelity.

Error bars and fidelity errors are statistical errors. Data has been corrected for state preparation and measurement (SPAM) errors, as described in [19, 26] and Section 2.1.4. Crosstalk errors were found by fitting the crosstalk pair parity scan to a sine curve as if it were a normal parity flop, calculating its fidelity as an entangling gate, and subtracting out the 25% base fidelity that represents a complete statistical mixture; any fidelity above that represents a correlation or small amount of entanglement that is considered an error here. All crosstalk fidelities for all pairs was well below 50% (in fact, most were quite close to 25%), indicating that no crosstalk pairs had verifiable entanglement.

While the gate time $\tau_{gate} = 250 \mu s$ for running 2 XX gates in parallel is comparable to that of a single XX gate (and consequently, half the time it would take to execute two XX gates in series), the parallel gates scheme requires somewhat more optical power. The Rabi frequency Ω is proportional to the square root of each beam intensity I , $\Omega \propto \sqrt{I_0 I_1}$, where I_0 and I_1 are the beam intensities for the

individual and global beams, respectively. We can therefore calculate the ratio $R_{||}$ of the power for a gate executed in parallel to the power required for a single XX gate on the same ions as $R_{||} = \frac{P_{||}}{P_{XX}} = \frac{I_{||}}{I_{XX}} = \left(\frac{\Omega_{||}}{\Omega_{XX}}\right)^2$. Since intensity is power per unit area, and the beam sizes do not vary, this cancels out. Power ratios for each gate are shown in Table 7.1. While some gates required rather more power (for example, we had some trouble finding a solution for (1,2), (3,4) that was both high-quality and low power), most gates performed in parallel require about two to four times as much power as their singly-performed counterparts. However, a full accounting of power requirements on this experiment must also take into account power wasted by unused beams, and the total time required to perform equivalent operations. Since the individual addressing system has all individual beams on at all times and are dumped after the AOM when not in use (see Section 2.3), any ion not illuminated corresponds to an individual beam wasting power. Running 2 XX gates in parallel takes $\tau_{gate} = 250 \mu s$ and uses beams each with power P to illuminate 4 ions, but performing those same 2 gates in series using stand-alone XX gates requires time $2\tau_{gate}$ and uses 4 beams each with power $P/4$ to $P/2$ to illuminate 2 ions, wasting 2 beams. This yields a choice of using twice the power (or more) in half the time versus half the power in twice the time, coming out close to equal cost depending on the gate in question; these parallel gates are then very useful when faster calculation is a higher priority than minimizing laser power.

7.3.1 Calculating Experimental Fidelities of 2-Qubit Entangling Gates

The fidelity of a two-qubit $XX(\chi)$ entangling gate can be measured by scanning the phase ϕ of a global $\frac{\pi}{2}$ rotation applied after performing the XX gate and calculating the parity at each point of the scan [58, 97, 111]. We start with a global rotation on 2 qubits,

$$\begin{aligned} R_G\left(\frac{\pi}{2}, \phi\right) &= R_1\left(\frac{\pi}{2}, \phi\right) \otimes R_2\left(\frac{\pi}{2}, \phi\right) \\ &= \frac{1}{2} \begin{pmatrix} 1 & -ie^{-i\phi} & -ie^{-i\phi} & -e^{-2i\phi} \\ -ie^{i\phi} & 1 & -1 & -ie^{-i\phi} \\ -ie^{i\phi} & -1 & 1 & -ie^{-i\phi} \\ -e^{2i\phi} & -ie^{i\phi} & -ie^{i\phi} & 1 \end{pmatrix}, \end{aligned} \quad (7.30)$$

and a general 2-qubit density matrix ρ_g that represents the density matrix produced after experimentally performing an XX gate,

$$\rho_g = \begin{pmatrix} \rho_{00} & \rho_{01} & \rho_{02} & \rho_{03} \\ \rho_{01}^* & \rho_{11} & \rho_{12} & \rho_{13} \\ \rho_{02}^* & \rho_{12}^* & \rho_{22} & \rho_{23} \\ \rho_{03}^* & \rho_{13}^* & \rho_{23}^* & \rho_{33} \end{pmatrix}, \quad (7.31)$$

where $\rho_{00} = |00\rangle\langle 00|$, $\rho_{01} = |00\rangle\langle 01|$, \dots , $\rho_{23} = |10\rangle\langle 11|$, $\rho_{33} = |11\rangle\langle 11|$. After performing the analysis pulse $R_G\left(\frac{\pi}{2}, \phi\right)$, the new density matrix $\rho_a = R_G\left(\frac{\pi}{2}, \phi\right) \cdot$

$\rho_g \cdot R_G^\dagger(\frac{\pi}{2}, \phi)$ is used to calculate the parity:

$$\begin{aligned}
\Pi(\rho_a, \phi) &= (\rho_a^{00} + \rho_a^{33}) - (\rho_a^{11} + \rho_a^{22}) \\
&= 2A_{12} \cos \phi_{12} - 2A_{03} \cos(2\phi - \phi_{03}) \\
&= 2A_{12} \cos \phi_{12} - A_{\Pi} \cos(2\phi - \phi_{03}), \tag{7.32}
\end{aligned}$$

where parity is defined as the sum of the even parity populations minus the sum of the odd parity populations and the coherences (off-diagonal density matrix elements) from ρ_g are re-written in the form $\rho_{xy} = A_{xy}e^{-i\phi_{xy}}$. Let us also define the parity amplitude $A_{\Pi} \equiv 2A_{03}$.

Now we calculate the fidelity of an $XX(\chi)$ gate. Using the $XX(\chi)$ gate unitary (see Equation 5.3), we construct the ideal density matrix after an $XX(\chi)$ gate,

$$\begin{aligned}
\rho_{\text{ideal}} &= XX(\chi) \cdot |00\rangle\langle 00| \cdot XX(\chi)^\dagger \\
&= \begin{pmatrix} \cos^2(\chi) & 0 & 0 & i \cos(\chi) \sin(\chi) \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ -i \cos(\chi) \sin(\chi) & 0 & 0 & \sin^2(\chi) \end{pmatrix}. \tag{7.33}
\end{aligned}$$

The fidelity of the general fidelity matrix ρ_g with respect to the ideal fidelity matrix ρ_{ideal} is given by

$$F(\chi) = \text{Tr} \left[\rho_{\text{ideal}}(\chi) \cdot \rho_g \cdot \rho_{\text{ideal}}^\dagger(\chi) \right]. \tag{7.34}$$

Plugging in equations 7.31 and 7.33, using $A_{\Pi} \equiv 2A_{03}$, and simplifying yields

$$F(\chi) = \rho_{00} \cos^2(\chi) + \rho_{33} \sin^2(\chi) + A_{\Pi} \cos(\chi) \sin(\chi) \quad (7.35)$$

as the fidelity of an $XX(\chi)$ gate. Specifically for maximally entangling gates, we plug in $\chi = \frac{\pi}{4}$ and get

$$F\left(\chi = \frac{\pi}{4}\right) = \frac{1}{2}(\rho_{00} + \rho_{33}) + \frac{1}{2}A_{\Pi}. \quad (7.36)$$

While ρ_{00} and ρ_{33} are simply the populations in $|00\rangle$ and $|11\rangle$ respectively after an XX gate, we still need the A_{Π} term. We can extract this from a parity scan using Equation 7.32. Given a perfect $XX(\chi)$ gate where $A_{12} = 0$, $\phi_{03} = -\frac{\pi}{2}$, and $A_{\Pi} = 2A_{03} = 2\cos(\chi)\sin(\chi)$ (from Equation 7.33), scanning the analysis phase ϕ from 0 to 2π and measuring the parity at each point will yield a sine curve of amplitude $2\cos(\chi)\sin(\chi)$ with 2 periods in the range from 0 to 2π . (For a fully entangling $XX(\chi = \frac{\pi}{4})$ gate, the sine curve should have amplitude 1.) Consequently, by fitting a sine curve to this measured parity curve, we can estimate the parity amplitude A_{Π} and use it in Equation 7.35 to calculate the gate fidelity.

7.3.2 Fidelity of Parallel 2-Qubit Entangling Gates with Different Degrees of Entanglement

Since the XX gates in this parallelization scheme have independent calibration (see Section 7.3.3), the χ parameters of the two XX gates are independent. The

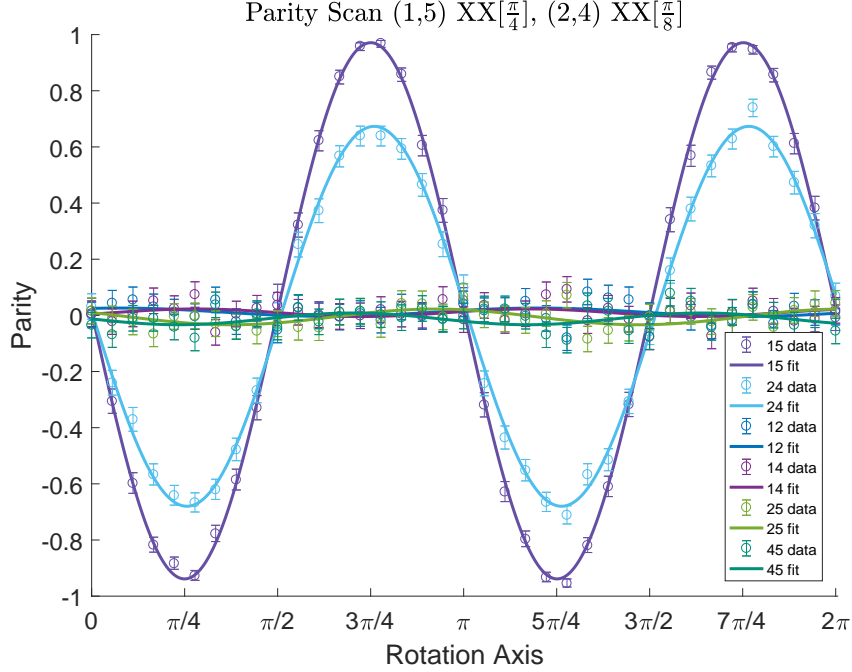


Figure 7.13: Parity curve for parallel $XX(\chi)$ gates on ions (1,5) and (2,4), where an $XX(\frac{\pi}{4})$ gate is performed on ions (1,5), and an $XX(\frac{\pi}{8})$ gate is performed on ions (2,4). This yields fidelities of 96.4(3)% and 99.4(3)% on the respective entangled pairs, with an average crosstalk error of 2.2(3)% and corrected for 1% SPAM errors.

continuously-variable parameter χ is directly related to the amount of entanglement generated between the two qubits, given by

$$XX(\chi) |00\rangle = \frac{1}{\sqrt{2}} (\cos(\chi) |00\rangle - i \sin(\chi) |11\rangle), \quad (7.37)$$

and can be adjusted on the experiment by scaling the power of the overall gate. Consequently, we can simultaneously implement two XX gates with different degrees of entanglement, which may prove useful for some applications. For example, the full adder implementation in Section 7.5 will require simultaneously performing an $XX(\frac{\pi}{4})$ gate on one pair of qubits, and an $XX(\frac{\pi}{8})$ gate on another pair of qubits. To demonstrate this capability, Figure 7.13 shows parity scan data for a

simultaneous $XX\left(\frac{\pi}{4}\right)$ gate on ions (1,5) and an $XX\left(\frac{\pi}{8}\right)$ gate on ions (2,4). The data is analyzed as in Section 7.3, but while we use Equation 7.36 (setting $\chi = \frac{\pi}{4}$) to calculate the fidelity for the (1,5) gate, we use Equation 7.35 and set $\chi = \frac{\pi}{8}$ for the (2,4) gate. The respective gate fidelities are therefore 96.4(3)% and 99.4(3)%, with an average crosstalk error of 2.2(3)% and corrected for 1% SPAM errors.

7.3.3 Independence of Parallel Gate Calibration

Parallel gates can be calibrated independently from one another by adjusting a scaling factor that controls the overall power on the gate without modifying the pulse shape. Furthermore, adjusting a scaling factor that controls the power on a single ion only affects the gate in which it participates by modifying the total amount of entanglement, without any apparent ill effects on the gate quality. This was confirmed experimentally using parallel operations on ions (1,2) and (3,4) by scanning over the scaling factors associated with ions 1 and 2. Figure 7.14 shows several such scans over the scaling factors for ions 1 and 2 while keeping the (3,4) gate “on”, with the scaling factor for those two ions set near to a fully-entangling gate; Figures 7.14(a,c) show scans with just the scaling factor for ion 1 while holding the scale factor for ion 2 constant, and Figures 7.14(b,d) show scans over the scaling factor for ions 1 and 2 together. Figure 7.15 shows scans over the scaling factors for ions 1 and 2 while keeping the interaction on (3,4) “off”; the scaling factor for this gate is set to 0, so the ions see no light and therefore perform no interaction during the gate. Figure 7.15(a) scans the scale factor just on ion 2 while holding the scale

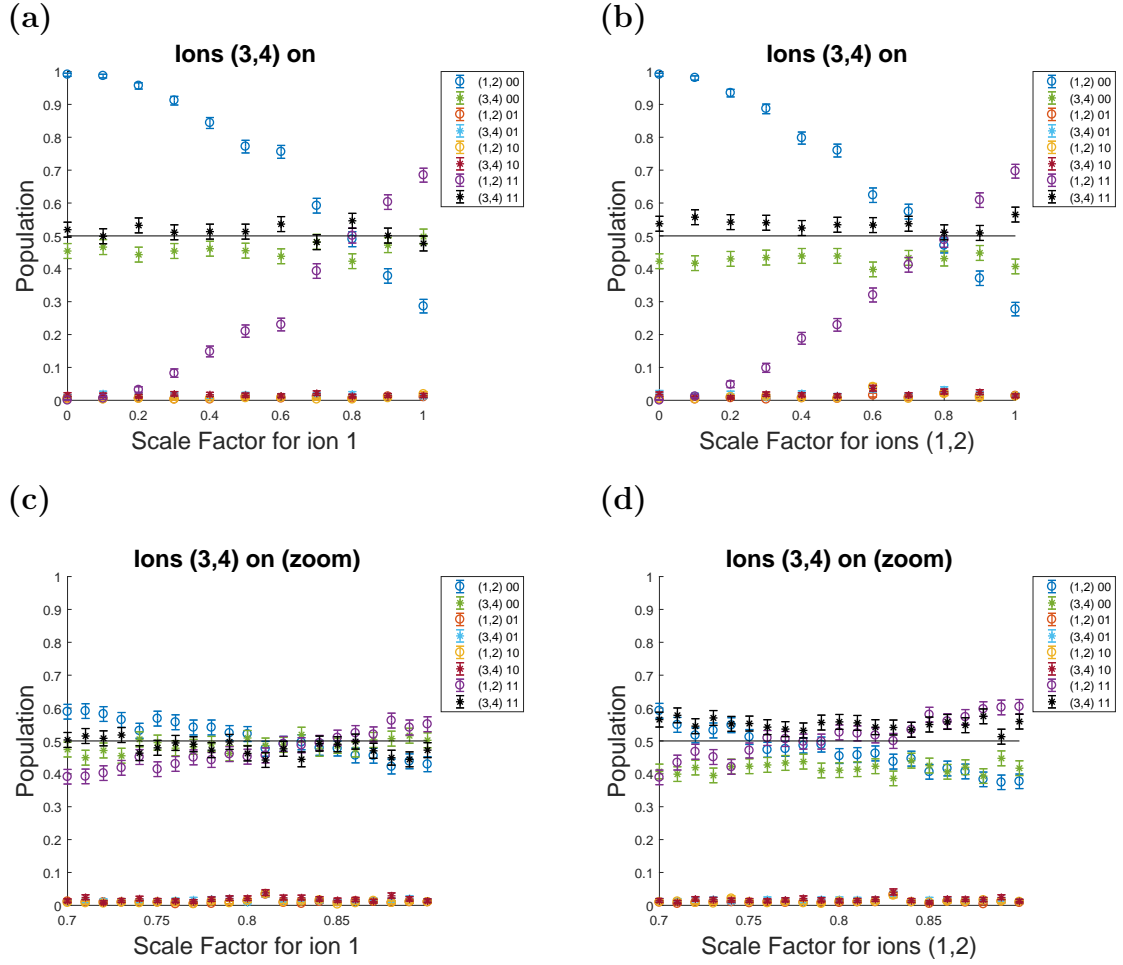


Figure 7.14: Parallel gates can be calibrated independently. Here, we perform an entangling gate on ions (3,4) while simultaneously performing a gate on ions (1,2) and scanning a scale factor, which determines the overall amplitude envelope on the control signal applied to perform the gate on ions (1,2). This indicates we can independently set the amount of entanglement created by each gate when performed in parallel. (a) Scanning the scale factor on ion 1, with ions (3,4) performing an entangling gate. (b) Scanning the scale factor on ions (1,2), with ions (3,4) performing an entangling gate. (c) Scanning the scale factor on ion 1, with ions (3,4) performing an entangling gate, with higher resolution zoomed in close to full entanglement for ions (1,2). (d) Scanning the scale factor on ions (1,2), with ions (3,4) performing an entangling gate, with higher resolution zoomed in close to full entanglement for ions (1,2).

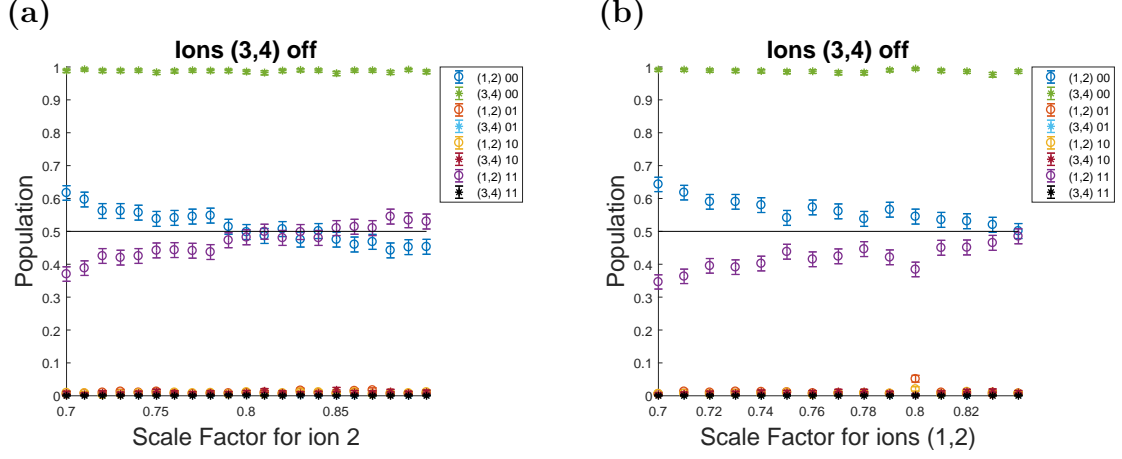


Figure 7.15: Parallel gates can be calibrated independently. (a) Scanning the scale factor on ion 2, with no light on ions (3,4). (b) Scanning the scale factor on ions (1,2), with no light on ions (3,4).

on ion 1 constant, and Figure 7.15(b) scans the overall scaling factor for ions 1 and 2 together. For all of these scans, as the scaling factors are increased, the population in $|11\rangle$ for ions 1 and 2 increases (and the population in $|00\rangle$ decreases correspondingly), while the $|00\rangle$ and $|11\rangle$ populations for the (3,4) gate remain unchanged.

7.4 Simultaneous *CNOT* Gates

As an example application of a parallel operation, we performed a pair of *CNOT* gates in parallel on two pairs of ions. The *CNOT* gate sequence (compiled version with *R* and *XX* gates shown in Figure 5.2(a)) was performed simultaneously on the pair (1,4), with ion 1 acting as the control and ion 4 acting as the target, and on the pair (2,3), with ion 2 acting as the control and ion 3 acting as the target. Each constituent operation in the composite gate was performed in parallel, with each rotation performed at the same time as the corresponding rotation on the other pair, and the two *XX* gates performed using parallel *XX* gates on ion pairs (1,4)

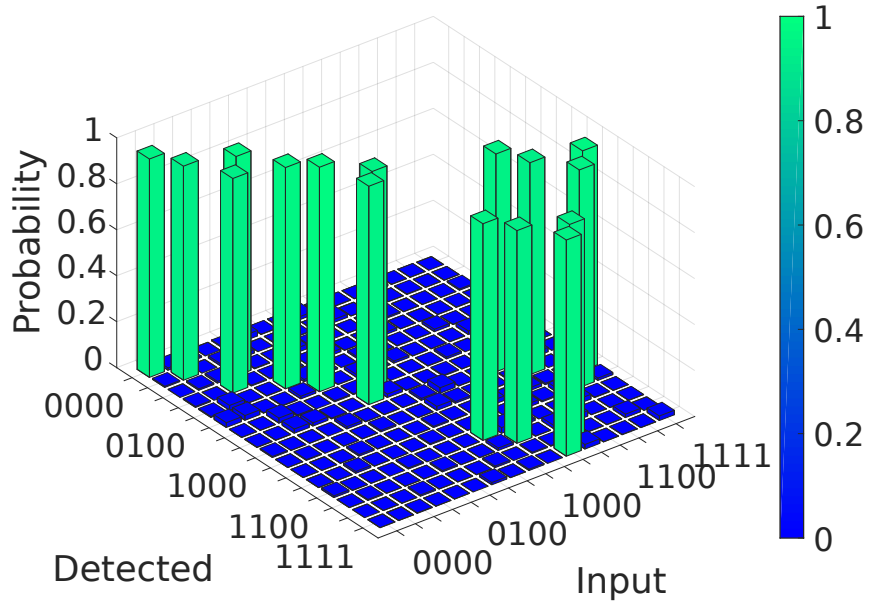


Figure 7.16: Data for simultaneous $CNOT$ gates on ions (1,4) and (2,3), with an average process fidelity of 94.5(2)% and corrected for average SPAM errors of 5%.

and (2, 3) (see Figure 7.10 for fidelity data.)

The simultaneous $CNOT$ gates were performed for each of the 16 possible bitwise inputs, and population data for the 16 possible bitwise outputs is shown in Figure 7.16 with an average process fidelity of 94.5(2)% and corrected for average SPAM errors of 5%.

7.5 The Quantum Full Adder

The ability to add numbers is fundamental to classical computers, and indeed was one of the first motivations for computing machines; arguably the very first computing machine, the abacus, was a tool for adding and subtracting numbers. In modern computing, a full adder is a basic circuit that can be cascaded to add many-bit numbers, and can be found in processors as components of arithmetic logic units

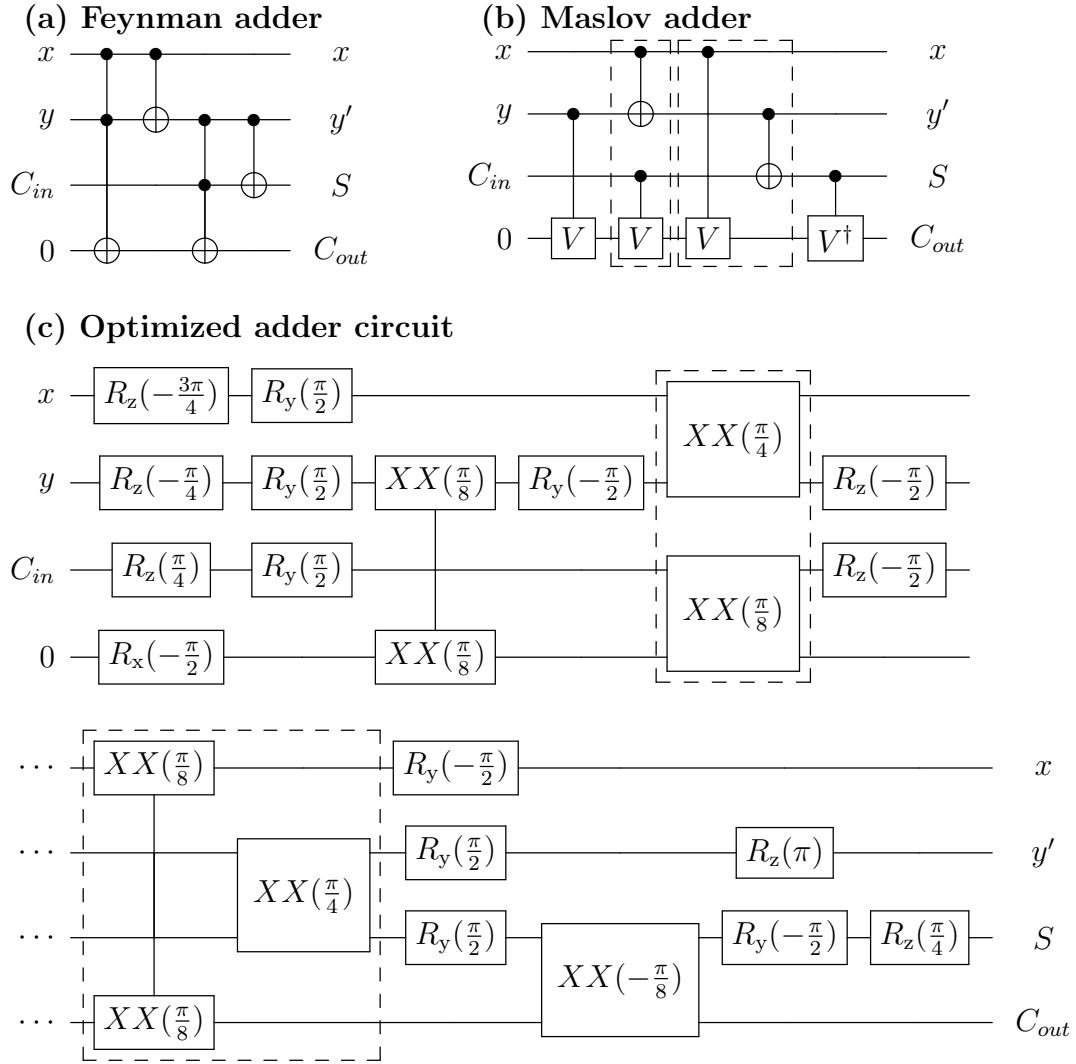


Figure 7.17: (a) Feynman quantum full adder [112]. (b) Maslov adder with 2-qubit gate depth 4 [82]. (c) Application-optimized full adder implementation using $XX(\chi)$, $R_x(\theta)$, and $R_y(\theta)$ gates. The two parallel 2-qubit operations are outlined in dashed boxes for both (b) and (c).

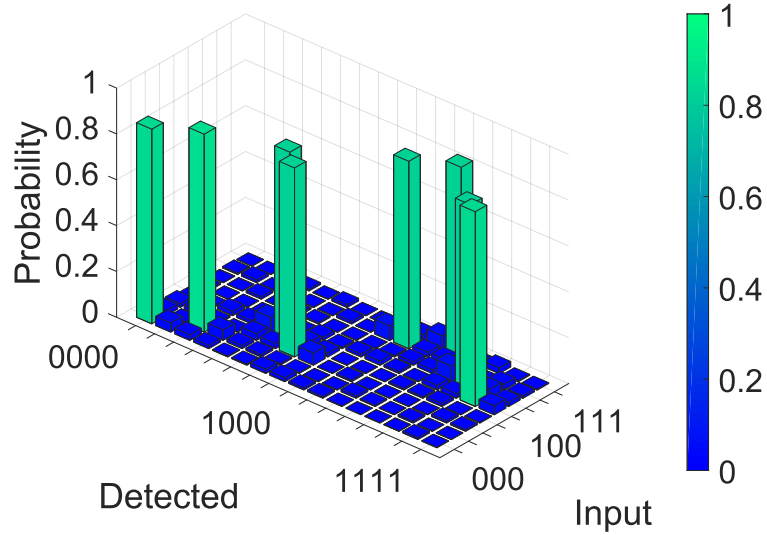


Figure 7.18: Data for full adder using simultaneous 2-qubit gates on ions (1,2,4,5), with an average process fidelity of 83.3(3)% and corrected for average SPAM errors of 3%.

(ALU's) and performing low-level operations like computing register addresses.

A cascadable full adder is one that takes 3 inputs - two bits x and y you wish to add, plus a carry bit C_{in} stemming from a previous addition - and has 2 outputs that communicate the 2-bit value of the sum of the two inputs. To build a quantum adder, one must design a circuit that accomplishes these goals while being reversible. Feynman first designed such a circuit using *CNOT* and Toffoli gates [112], shown in Figure 7.17(a).

However, with 5 2-qubit interactions per Toffoli meaning that this full adder would require 12 *XX* gates to implement on an ion trap quantum computer, this circuit is not efficient. A more efficient circuit requiring at most 6 2-qubit interactions was shown in [82], one which has the further advantage of being reduced to a gate depth of 4 if simultaneous two-qubit operations are available, as shown by the dashed outlines in Figure 7.17(b). Like the Feynman circuit, the quantum

full adder requires 4 qubits, 3 for the inputs x , y , and the carry bit C_{in} , and the fourth a qubit initialized to $|0\rangle$. The four outputs consist of the first input, x , simply carrying through; y' , which carries $x \oplus y$ (an additional $CNOT$ can be added to extract y if desired); and the sum S and output carry C_{out} , which together comprise the 2-bit result of summing x , y , and C_{in} , where C_{out} is the most significant bit and hence becomes the carry bit to the next adder in the cascade, and S is the least significant bit. We can also write sum as $S = x \oplus y \oplus C_{in}$ and the output carry as $C_{out} = (x \cdot y) \oplus (C_{in} \cdot (x \oplus y))$.

The optimized full adder circuit to be implemented on the experiment, shown in Figure 7.17(c), is constructed by combining the $CNOT$, $C(V)$, and $C(V^\dagger)$ gates from Figure 5.12 and further optimizing the rotations per the method described in Section 5.2.1. The two parallel 2-qubit operations are outlined in dashed boxes for Figures 7.17(b-c).

The full adder was implemented using 2 different parallel XX gate configurations, as well as the rotations and additional XX gates shown in Figure 7.17(c). The inputs x , y , C_{in} , and 0 were mapped to the qubits (1, 2, 4, 5) respectively. Consequently, the first parallel gates were implemented on the pairs (1,2) $XX(\frac{\pi}{4})$ and (4,5) $XX(\frac{\pi}{8})$ (see Figure 7.12 for (1,2) $XX(\frac{\pi}{4})$, (4,5) $XX(\frac{\pi}{4})$ fidelity data), and the second parallel gate was implemented on the pairs (1,5) $XX(\frac{\pi}{4})$ and (2,4) $XX(\frac{\pi}{8})$ (see Figure 7.9 for (1,5) $XX(\frac{\pi}{4})$, (2,4) $XX(\frac{\pi}{4})$ fidelity data and Figure 7.13 for (1,5) $XX(\frac{\pi}{4})$, (2,4) $XX(\frac{\pi}{8})$ fidelity data.) Figure 7.18 shows the resulting data from implementing this algorithm, with all 8 possible bitwise inputs on the 3 input qubits, and displaying the populations in all 16 possible bitwise outputs on

the 4 qubits used. The data yielded an average process fidelity of 83.3(3)%, and was corrected for average SPAM errors of 3%.

7.6 Toward a Single-Operation GHZ State

This control scheme for parallel 2-qubit entangling gates in ions also suggests a method for performing multi-qubit entanglement in a single operation. Of particular interest is the creation of GHZ states [113], which are a class of non-biseparable maximally-entangled multi-qubit states. To calculate parallel XX gates, as discussed in Section 7.1, we set the spin-spin interaction terms $\chi = \frac{\pi}{4}$ for the two desired entangling interactions, and to 0 for the remaining 4 crosstalk interactions.

However, setting all 6 spin-spin interaction terms to $\chi = \frac{\pi}{4}$ yields the unitary

$$\begin{aligned}
U_{\text{GHZ}}^{\text{ideal}} &= U \left(\alpha_{\{i,j,m,n\},k} = 0, \chi_{ij} = \chi_{im} = \chi_{in} = \chi_{jm} = \chi_{jn} = \chi_{mn} = \frac{\pi}{4} \right) \\
&= \frac{1}{\sqrt{2}} \begin{pmatrix}
1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -i \\
0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -i & 0 \\
0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -i & 0 & 0 \\
0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -i & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & -i & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & -i & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & -i & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & -i & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & -i & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & -i & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & -i & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & -i & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\
0 & 0 & -i & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\
0 & -i & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\
-i & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1
\end{pmatrix}
\end{aligned} \tag{7.38}$$

and adding one Z rotation produces a 4-qubit GHZ state:

$$R_z^1 \left(\frac{\pi}{2} \right) \cdot U_{\text{GHZ}}^{\text{ideal}} |0000\rangle = \frac{1}{\sqrt{2}} (|0000\rangle + |1111\rangle). \tag{7.39}$$

Following a similar derivation as in Section 7.1, we therefore calculate the objective fidelity function to be

$$\begin{aligned}
F_{\text{GHZ}} \left(\alpha_{\{i,j,m,n\},k}, \chi_{ij}^{\text{ideal}}, \chi_{im}^{\text{ideal}}, \chi_{in}^{\text{ideal}}, \chi_{jm}^{\text{ideal}}, \chi_{jn}^{\text{ideal}}, \chi_{mn}^{\text{ideal}}, \chi_{ij}, \chi_{im}, \chi_{in}, \chi_{jm}, \chi_{jn}, \chi_{mn} \right) = \\
\frac{1}{128} (8 + \Gamma_{+---} + \Gamma_{+--+} + \Gamma_{+-+-} + \Gamma_{-+++} + \Gamma_{++--} + \Gamma_{++-+} + \Gamma_{++++} + \Gamma_{++++})
\end{aligned}$$

$$\begin{aligned}
& + 2 (\Gamma_{0+++} + \Gamma_{+000}) \cos [2 (\Delta\chi_{ij} + \Delta\chi_{im} + \Delta\chi_{in})] \\
& + 2 (\Gamma_{0+--} + \Gamma_{+000}) \cos [2 (\Delta\chi_{ij} - \Delta\chi_{im} + \Delta\chi_{in})] \\
& + 2 (\Gamma_{0++-} + \Gamma_{+000}) \cos [2 (\Delta\chi_{ij} + \Delta\chi_{im} - \Delta\chi_{in})] \\
& + 2 (\Gamma_{0+--} + \Gamma_{+000}) \cos [2 (\Delta\chi_{ij} - \Delta\chi_{im} - \Delta\chi_{in})] \\
& + 2 (\Gamma_{0+00} + \Gamma_{+0++}) \cos [2 (\Delta\chi_{ij} + \Delta\chi_{jm} + \Delta\chi_{jn})] \\
& + 2 (\Gamma_{00++} + \Gamma_{++00}) \cos [2 (\Delta\chi_{im} + \Delta\chi_{in} + \Delta\chi_{jm} + \Delta\chi_{jn})] \\
& + 2 (\Gamma_{0+00} + \Gamma_{+0-+}) \cos [2 (\Delta\chi_{ij} - \Delta\chi_{jm} + \Delta\chi_{jn})] \\
& + 2 (\Gamma_{00+-} + \Gamma_{+-00}) \cos [2 (\Delta\chi_{im} - \Delta\chi_{in} - \Delta\chi_{jm} + \Delta\chi_{jn})] \\
& + 2 (\Gamma_{0+00} + \Gamma_{+0+-}) \cos [2 (\Delta\chi_{ij} + \Delta\chi_{jm} - \Delta\chi_{jn})] \\
& + 2 (\Gamma_{00+-} + \Gamma_{++00}) \cos [2 (\Delta\chi_{im} - \Delta\chi_{in} + \Delta\chi_{jm} - \Delta\chi_{jn})] \\
& + 2 (\Gamma_{0+00} + \Gamma_{+0--}) \cos [2 (\Delta\chi_{ij} - \Delta\chi_{jm} - \Delta\chi_{jn})] \\
& + 2 (\Gamma_{00++} + \Gamma_{+-00}) \cos [2 (\Delta\chi_{im} + \Delta\chi_{in} - \Delta\chi_{jm} - \Delta\chi_{jn})] \\
& + 2 (\Gamma_{00+0} + \Gamma_{++0+}) \cos [2 (\Delta\chi_{im} + \Delta\chi_{jm} + \Delta\chi_{mn})] \\
& + 2 (\Gamma_{0+0+} + \Gamma_{+0+0}) \cos [2 (\Delta\chi_{ij} + \Delta\chi_{in} + \Delta\chi_{jm} + \Delta\chi_{mn})] \\
& + 2 (\Gamma_{00+0} + \Gamma_{+-0+}) \cos [2 (\Delta\chi_{im} - \Delta\chi_{jm} + \Delta\chi_{mn})] \\
& + 2 (\Gamma_{0+0-} + \Gamma_{+0-0}) \cos [2 (\Delta\chi_{ij} - \Delta\chi_{in} - \Delta\chi_{jm} + \Delta\chi_{mn})] \\
& + 2 (\Gamma_{0++0} + \Gamma_{+00+}) \cos [2 (\Delta\chi_{ij} + \Delta\chi_{im} + \Delta\chi_{jn} + \Delta\chi_{mn})] \\
& + 2 (\Gamma_{000+} + \Gamma_{+++0}) \cos [2 (\Delta\chi_{in} + \Delta\chi_{jn} + \Delta\chi_{mn})] \\
& + 2 (\Gamma_{0+-0} + \Gamma_{+00-}) \cos [2 (\Delta\chi_{ij} - \Delta\chi_{im} - \Delta\chi_{jn} + \Delta\chi_{mn})] \\
& + 2 (\Gamma_{000+} + \Gamma_{+-+0}) \cos [2 (\Delta\chi_{in} - \Delta\chi_{jn} + \Delta\chi_{mn})]
\end{aligned}$$

$$\begin{aligned}
& + 2 (\Gamma_{00+0} + \Gamma_{++0-}) \cos [2 (\Delta\chi_{im} + \Delta\chi_{jm} - \Delta\chi_{mn})] \\
& + 2 (\Gamma_{0+0-} + \Gamma_{+0+0}) \cos [2 (\Delta\chi_{ij} - \Delta\chi_{in} + \Delta\chi_{jm} - \Delta\chi_{mn})] \\
& + 2 (\Gamma_{00+0} + \Gamma_{+-0-}) \cos [2 (\Delta\chi_{im} - \Delta\chi_{jm} - \Delta\chi_{mn})] \\
& + 2 (\Gamma_{0+0+} + \Gamma_{+0-0}) \cos [2 (\Delta\chi_{ij} + \Delta\chi_{in} - \Delta\chi_{jm} - \Delta\chi_{mn})] \\
& + 2 (\Gamma_{0+-0} + \Gamma_{+00+}) \cos [2 (\Delta\chi_{ij} - \Delta\chi_{im} + \Delta\chi_{jn} - \Delta\chi_{mn})] \\
& + 2 (\Gamma_{000+} + \Gamma_{++-0}) \cos [2 (\Delta\chi_{in} + \Delta\chi_{jn} - \Delta\chi_{mn})] \\
& + 2 (\Gamma_{0++0} + \Gamma_{+00-}) \cos [2 (\Delta\chi_{ij} + \Delta\chi_{im} - \Delta\chi_{jn} - \Delta\chi_{mn})] \\
& + 2 (\Gamma_{000+} + 2\Gamma_{+--0}) \cos [2 (\Delta\chi_{in} - \Delta\chi_{jn} - \Delta\chi_{mn})], \tag{7.40}
\end{aligned}$$

where

$$\begin{aligned}
\Delta\chi_{ij} &= \chi_{ij} - \chi_{ij}^{\text{ideal}} \\
\Delta\chi_{mn} &= \chi_{mn} - \chi_{mn}^{\text{ideal}} \\
\Delta\chi_{im} &= \chi_{im} - \chi_{im}^{\text{ideal}} \\
\Delta\chi_{in} &= \chi_{in} - \chi_{in}^{\text{ideal}} \\
\Delta\chi_{jm} &= \chi_{jm} - \chi_{jm}^{\text{ideal}} \\
\Delta\chi_{jn} &= \chi_{jn} - \chi_{jn}^{\text{ideal}}. \tag{7.41}
\end{aligned}$$

This indicates we may be able to use the same optimization approach to produce pulse shapes that will create GHZ states when applied to the ions. Unlike with parallel gates, however, it may be necessary to allow independent pulse shapes on all 4 ions, rather than solving for pairwise solutions; this will provide more free pa-

rameters. Additional challenges will include finding effective calibration techniques when implementing such gates on the experiment, since there will be 6 interactions that will all need to be at the same strength, but only 4 control signals. Our current approach of calibrating a 2-qubit gate by adjusting the overall power for the pulse shape applied by the control signal may no longer work; new techniques with more degrees of freedom may be needed, such as independently adjusting the power for different segments of the pulse shape on each ion.

The benefits of implementing GHZ states with fewer gates would be significant, as it would substantially reduce the circuit depth of several important algorithms. With only 2-qubit gates available, building a GHZ state of size N requires $O(N)$ 2-qubit gates. With parallel 2-qubit gates available, the gate depth required to build a GHZ state is reduced to $O(\log(N))$; this is accomplished with a binary tree algorithm by dividing all qubits into pairs and entangling those pairs in parallel, then entangling pairs of these pairs, and so on until all are entangled. A single-operation GHZ state would drop this circuit depth to unity. Single-operation GHZ state construction will greatly enhance the efficiency of several algorithms; for example, arbitrary stabilizer circuits require $O(\frac{N^2}{\log(N)})$ $CNOT$ gates [114], but could be implemented in $O(N)$ gates with single-operation GHZ state circuitry [115]. Single-operation GHZ state creation will also benefit applications such as quantum secret sharing [116], Toffoli- N gates, the quantum Fourier transform, and quantum Fourier adder circuits [115].

7.7 Outlook

The scaling outlook on simultaneous gates is polynomial or better in the number of constraints to consider when calculating optimal solutions. As discussed in Section 7.1, two parallel XX gates in a chain of N ions requires $4N+6 \sim O(N)$ constraints, so the problem growth is linear in N . Entangling more pairs at once grows quadratically: entangling M pairs involves the interactions of $2M$ ions, yielding the number of spin-spin interactions we must control to be

$$\binom{2M}{2} = \frac{(2M)!}{2!(2M-2)!} = 2M^2 - M \sim O(M^2) \quad (7.42)$$

and the number of spin-motion interactions to be the number of ions times the number of modes, $2MN$. Scaling both the number of entangled pairs M and the number of ions N in the chain therefore gives a total problem growth rate of

$$2MN + 2M^2 - M = M(2N - 1) + 2M^2 \sim O(M^2 + MN). \quad (7.43)$$

The work presented here represents a successful first attempt at implementing parallel 2-qubit entangling operations in a chain of trapped ions, and further exploration will likely produce improved results. In particular, future research could investigate better optimization techniques to produce solutions that have higher fidelities and require lower power on the experiment. Since the solutions are found via optimization rather than analytically, the theoretical fidelities for these gates are

less than 1; most of the gates presented here have theoretical fidelities $<99.9\%$, so optimization approaches increasing the theoretical fidelities to $\geq 99.99\%$ will help. An additional problem is that testing the possible solutions generated by the optimization techniques is a time-consuming process; better systematics to pinpoint the most promising candidates may help here. Techniques that further suppress entanglement crosstalk between undesired pairs may also enhance experimental performance. However, these problems are all ones of overhead. Once a high-quality gate solution is implemented on the experiment, no further calculations are needed; only a single calibration is required to determine the overall power on each XX gate as the Rabi frequencies seen by the ions drift.

Several areas of exploration may help with increasing solution fidelity, both theoretically and experimentally. One is measuring and using the experimental Lamb-Dicke parameter $\eta_{i,k}$ for each spin-mode coupling, rather than calculating what they should be based on the ion spacings and measured motional modes. The trap in our experiment has some anharmonicities that cause asymmetries in the motional mode eigenvectors, so the participation of each ion in each mode is a little different from what one would theoretically expect. Calculating gate solutions with experimentally measured parameters may result in solutions that perform better on the experiment. Another parameter of interest is the gate time τ ; while we settled on $250\ \mu\text{s}$ as one comparable to our existing XX gate times, adjusting that parameter - theoretically and on the experiment - may prove beneficial. Easing constraints on the power needed may also allow for higher-fidelity solutions to be calculated, although increasing power on the experiment can exacerbate errors due

to Stark shifts; an accounting of tradeoffs between power usage and gate fidelity will optimize operational success given experimental conditions.

While the gate solution for $N = 5$ ions in principle has $4N + 6 = 26$ degrees of freedom that need to be solved (per Section 7.1), since solutions were found via optimization rather than analytically, we provided the optimization program additional degrees of freedom to increase the probability of finding good solutions; most of the gate solutions shown here use 60 segments per pair, although the (1,4), (2,3) optimization was performed with only 40 segments to no apparent ill effects. Further explorations of this technique may benefit from determining the optimal number of segments to use. A final line of future inquiry could include investigating whether the constraint matrices in Equation 7.12 are in fact positive or negative semidefinite, or can be modified to have such properties, as convex QCQP's are readily solved using semidefinite programming techniques and could allow for higher-fidelity solutions.

Chapter 8: Outlook

Trapped ion systems are a promising candidate for constructing a large-scale quantum computer. Here, we have reported results for several algorithms on a fully programmable machine with modular gates, and exhibiting a stack architecture in the control system that will be imperative for effective control of larger machines. With high initialization and detection efficiencies, well-understood coherent controls, and nature-provided, stable qubit frequencies providing long coherence times, scaling up to larger systems has a positive outlook. While much work certainly remains to be done, the biggest challenges can largely be tackled with engineering solutions.

8.1 Experimental Error Sources

The measured process fidelities for results discussed in this thesis are reduced from the theoretical ideal fidelities primarily through technical imperfections in the experimental system. The predominant source of error is beam pointing instability on the individual Raman beams, causing laser intensity fluctuations at the ions. This results in small random coherent errors, as the effective Rabi frequency seen by each ion drifts randomly during the course of an experiment. Causes for this instability

include air currents through the free-space Raman optical setup and phase noise on the laser itself. Future work on the experiment will include installing a piezo mirror controlled by a quadrant photodiode to actively stabilize the power and position of the individual addressing beams.

Another source of error is crosstalk between individual ion controls. While the individual Raman beams are very tightly focused, aberrations cause some spillover, so a given ion will see small amounts of light from its neighbors' control beams. Additionally, there is also crosstalk between neighboring channels at the 32-channel AOM. Overall, the ions typically experience $\sim 2\%$ nearest-neighbor crosstalk from their coherent optical controls, and $\leq 1\%$ from the next nearest neighbors [20]. Other error sources include inhomogeneous Stark shifts across the ion chain that could not be perfectly compensated. These control problems can largely be solved through improved engineering of key components.

8.2 Scalability of Ion Traps

Continuing challenges will include building apparatuses with more ions per trap that can execute long sequences of high-fidelity quantum gates. More qubits will be needed to implement algorithms larger than can be simulated classically. While error correction schemes provide protection against experimental errors, baseline gate fidelities will need to be higher than we have demonstrated here to achieve fault tolerant computation.

Schemes to implement entangling gates in chains of a few hundred ions have

been proposed [117], although full connectivity will not be possible at such scaling; co-designing algorithms with available connection architecture will mitigate the costs associated with reduced connectivity. Experimentally, improvements in multi-zone, microfabricated ion trap technology provide control over tens of ions in a single trap, and ongoing efforts in chip trap technology at Sandia National Labs, MIT Lincoln Labs, and elsewhere will likely scale to hundreds of ions in future. Vacuum chambers cooled to low temperatures using cryostats can operate at significantly lower pressure than room-temperature ion traps, allowing for longer chains to be held in a trap for longer periods of time before collisions with background gas disrupt the crystal.

Fidelity gains can be achieved through improvements in the quantum control hardware. Other experiments have shown up to four 9's of fidelity for single-qubit gates and up to three 9's of fidelity for two-qubit gates in ions [118, 119]. However, these were performed with only one or two ions in the trap; achieving these kinds of fidelities for larger chains will present a significant engineering challenge. As discussed in Section 8.1, a major source of experimental errors on our experiment is noise on the coherent addressing beams. Optical intensity stabilization will be imperative to achieving long, high-fidelity gate sequences, and can likely be achieved with integrated photonics, minimal use of free-space optics, active stabilization of the source power, and other engineering improvements. Coherence times for ions can be extended using techniques like dynamic decoupling [120] and additional magnetic shielding.

Further scalability may be achieved by connecting multiple ion trap modules

through ion shuttling [121] or photonic interconnects [6, 7]. In the latter scheme, each qubit register would constitute an elementary logic unit (ELU), and consist of a vacuum chamber containing an ion trap holding a single chain of ions, much like the one discussed in this work. Many such ELU modules could be connected remotely through coherent photon exchange over fibers. The qubits within a single register would perform computations together, much like the bits of an ELU in a classical computer. Quantum information that needs to be used in another register would be imparted from an atomic qubit to a photon, and the photon sent to another chamber to implement remote entanglement. The ion-photon information transfer could occur via a different ion species that emits photons at frequencies more suitable for fiber communication [122]. Optical switches would permit connections between all qubit registers. The modularity of this design would enable scalability, allowing for far bigger quantum computers than can be achieved with a single ion trap.

Bibliography

- [1] Yu I. Manin. In *Vychislimoe i nevychislimoe [Computable and Noncomputable]*, pages 13–15. Sov. Radio., 1980.
- [2] Richard P. Feynman. Simulating physics with computers. *International Journal of Theoretical Physics*, 21(6):467–488, 1982.
- [3] Edwin Pednault, John A. Gunnels, Giacomo Nannicini, Lior Horesh, Thomas Magerlein, Edgar Solomonik, and Robert Wisnieff. Breaking the 49-qubit barrier in the simulation of quantum circuits. *arXiv:1710.05867*, 2017.
- [4] D. Deutsch. Quantum theory, the Church-Turing principle and the universal quantum computer. *Proceedings of the Royal Society of London A*, 400(1818):97–117, 1985.
- [5] Markus Reiher, Nathan Wiebe, Krysta M. Svore, Dave Wecker, and Matthias Troyer. Elucidating reaction mechanisms on quantum computers. *Proceedings of the National Academy of Sciences*, 201619152, 2017.
- [6] C. Monroe and J. Kim. Scaling the ion trap quantum processor. *Science*, 339(6124):1164–1169, 2013.
- [7] C. Monroe, R. Raussendorf, A. Ruthven, K. R. Brown, P. Maunz, L.-M. Duan, and J. Kim. Large-scale modular quantum-computer architecture with atomic memory and photonic interconnects. *Physical Review A*, 89(2):022317, 2014.
- [8] M. H. Devoret and R. J. Schoelkopf. Superconducting circuits for quantum information: An outlook. *Science*, 339(6124):1169–1174, 2013.
- [9] Jay M. Gambetta, Jerry M. Chow, and Matthias Steffen. Building logical qubits in a superconducting quantum computing system. *npj Quantum Information*, 3(1):2, 2017.
- [10] T. D. Ladd, F. Jelezko, R. Laflamme, Y. Nakamura, C. Monroe, and J. L. O’Brien. Quantum computers. *Nature*, 464(7285):45–53, 2010.

- [11] P. S. Jessen, I. H. Deutsch, and R. Stock. Quantum information processing with trapped neutral atoms. *Quantum Information Processing*, 3(1):91–103, 2004.
- [12] M. Saffman. Quantum computing with atomic qubits and Rydberg interactions: progress and challenges. *Journal of Physics B: Atomic, Molecular and Optical Physics*, 49(20):202001, 2016.
- [13] Floris A. Zwanenburg, Andrew S. Dzurak, Andrea Morello, Michelle Y. Simmons, Lloyd C. L. Hollenberg, Gerhard Klimeck, Sven Rogge, Susan N. Coppersmith, and Mark A. Eriksson. Silicon quantum electronics. *Reviews of Modern Physics*, 85(3):961–1019, 2013.
- [14] Charles D. Hill, Eldad Peretz, Samuel J. Hile, Matthew G. House, Martin Fuechsle, Sven Rogge, Michelle Y. Simmons, and Lloyd C. L. Hollenberg. A surface code quantum computer in silicon. *Science Advances*, 1(9):e1500707, 2015.
- [15] Lilian Childress and Ronald Hanson. Diamond NV centers for quantum computing and quantum networks. *MRS Bulletin*, 38(2):134–138, 2013.
- [16] E. Knill, R. Laflamme, and G. J. Milburn. A scheme for efficient quantum computation with linear optics. *Nature*, 409(6816):46–52, 2001.
- [17] Pieter Kok, W. J. Munro, Kae Nemoto, T. C. Ralph, Jonathan P. Dowling, and G. J. Milburn. Linear optical quantum computing with photonic qubits. *Reviews of Modern Physics*, 79(1):135–174, 2007.
- [18] Ze-Liang Xiang, Sahel Ashhab, J. Q. You, and Franco Nori. Hybrid quantum circuits: Superconducting circuits interacting with other quantum systems. *Reviews of Modern Physics*, 85(2):623–653, 2013.
- [19] S. Debnath, N. M. Linke, C. Figgatt, K. A. Landsman, K. Wright, and C. Monroe. Demonstration of a small programmable quantum computer with atomic qubits. *Nature*, 536(7614):63–66, 2016.
- [20] Shantanu Debnath. *A Programmable Five Qubit Quantum Computer Using Trapped Atomic Ions*. PhD thesis, University of Maryland, 2016.
- [21] David P. DiVincenzo. The physical implementation of quantum computation. *Fortschritte der Physik*, 48(9):771–783, 2000.
- [22] D. Hucul, I. V. Inlek, G. Vittorini, C. Crocker, S. Debnath, S. M. Clark, and C. Monroe. Modular entanglement of atomic qubits using photons and phonons. *Nature Physics*, 11(1):37–42, 2015.
- [23] L.-M. Duan, M. D. Lukin, J. I. Cirac, and P. Zoller. Long-distance quantum communication with atomic ensembles and linear optics. *Nature*, 414(6862):413–418, 2001.

- [24] C. Figgatt, D. Maslov, K. A. Landsman, N. M. Linke, S. Debnath, and C. Monroe. Complete 3-qubit grover search on a programmable quantum computer. *Nature Communications*, 8(1):1918, 2017.
- [25] Caroline Figgatt, Aaron Ostrander, Norbert M. Linke, Kevin A. Landsman, Daiwei Zhu, Dmitri Maslov, and Christopher Monroe. Parallel 2-qubit operations on a programmable ion trap quantum computer. *forthcoming*, 2018.
- [26] Norbert M. Linke, Mauricio Gutierrez, Kevin A. Landsman, Caroline Figgatt, Shantanu Debnath, Kenneth R. Brown, and Christopher Monroe. Fault-tolerant quantum error detection. *Science Advances*, 3(10):e1701074, 2017.
- [27] Norbert M. Linke, Dmitri Maslov, Martin Roetteler, Shantanu Debnath, Caroline Figgatt, Kevin A. Landsman, Kenneth Wright, and Christopher Monroe. Experimental comparison of two quantum computing architectures. *Proceedings of the National Academy of Sciences*, 114:3305–3310, 2017.
- [28] Pak Hong Leung, Kevin A. Landsman, Caroline Figgatt, Norbert M. Linke, Christopher Monroe, and Kenneth R. Brown. Robust 2-qubit gates in a linear ion crystal using a frequency-modulated driving force. *Physical Review Letters*, 120(2):020501, 2018.
- [29] S. Debnath, N.M. Linke, S.-T. Wang, C. Figgatt, K.A. Landsman, L.-M. Duan, and C. Monroe. Observation of hopping and blockade of bosons in a trapped ion spin chain. *Physical Review Letters*, 120(7):073001, 2018.
- [30] Norbert M. Linke, Sonika Johri, Caroline Figgatt, Kevin A. Landsman, Anne Y. Matsuura, and Christopher Monroe. Measuring the Renyi entropy of a two-site Fermi-Hubbard model on a trapped ion quantum computer. *arXiv:1712.08581*, 2017.
- [31] Kevin A. Landsman, Caroline Figgatt, Thomas Schuster, Norbert M. Linke, Beni Yoshida, Norman Y. Yao, and Christopher Monroe. Scalable measurements of quantum scrambling and out-of-time-order correlations with trapped ions. *forthcoming*, 2018.
- [32] Neal Solmeyer, Norbert M. Linke, Caroline Figgatt, Kevin A. Landsman, Radhakrishnan Balu, George Siopsis, and Christopher Monroe. Demonstration of a Bayesian quantum game on an ion trap quantum computer. *arXiv:1802.08116*, 2018.
- [33] S. Olmschenk, K. C. Younge, D. L. Moehring, D. N. Matsukevich, P. Maunz, and C. Monroe. Manipulation and detection of a trapped Yb^+ hyperfine qubit. *Physical Review A*, 76(5):052314, 2007.
- [34] Wayne M. Itano and D. J. Wineland. Laser cooling of ions stored in harmonic and penning traps. *Physical Review A*, 25(1):35–54, 1982.

- [35] D. J. Wineland, C. Monroe, W. M. Itano, D. Leibfried, B. E. King, and D. M. Meekhof. Experimental issues in coherent quantum-state manipulation of trapped atomic ions. *Journal of Research of the National Institute of Standards and Technology*, 103(3):259–328, 1998.
- [36] Andrew Chew. *Doppler-Free Spectroscopy of Iodine at 739nm*. Senior honors thesis, University of Michigan, 2008.
- [37] Simon Webster. *Raman Sideband Cooling and Coherent Manipulation of Trapped Ions*. DPhil thesis, Oxford University, 2005.
- [38] F. Diedrich, J. C. Bergquist, Wayne M. Itano, and D. J. Wineland. Laser cooling to the zero-point energy of motion. *Physical Review Letters*, 62(4):403–406, 1989.
- [39] C. Shen and L.-M. Duan. Correcting detection errors in quantum state engineering through data processing. *New Journal of Physics*, 14(5):053053, 2012.
- [40] Alireza Seif, Kevin A. Landsman, Norbert M. Linke, Caroline Figgatt, Christopher Monroe, and Mohammad Hafezi. Machine learning assisted readout of trapped-ion qubits. *forthcoming*, 2018.
- [41] G. J. Milburn, S. Schneider, and D.F.V. James. Ion trap quantum computing with warm ions. *Fortschritte der Physik*, 48(9-11):801–810, 2000.
- [42] Wolfgang Paul. Electromagnetic traps for charged and neutral particles. *Reviews of Modern Physics*, 62(3):531–540, 1990.
- [43] K. G. Johnson, J. D. Wong-Campos, A. Restelli, K. A. Landsman, B. Neyenhuis, J. Mizrahi, and C. Monroe. Active stabilization of ion trap radiofrequency potentials. *Review of Scientific Instruments*, 87(5):053110, 2016.
- [44] W. W. Macalpine and R. O. Schildknecht. Coaxial resonators with helical inner conductor. *Proceedings of the IRE*, 47(12):2099–2105, 1959.
- [45] D. F. V. James. Quantum dynamics of cold trapped ions with application to quantum computation. *Applied Physics B*, 66(2):181–190, 1998.
- [46] Chr. Balzer, A. Braun, T. Hannemann, Chr. Paape, M. Ettler, W. Neuhauser, and Chr. Wunderlich. Electrodynamically trapped Yb^+ ions for quantum information processing. *Physical Review A*, 73(4):041407, 2006.
- [47] D. P. L. Aude Craik, N. M. Linke, M. A. Sepiol, T. P. Harty, J. F. Goodwin, C. J. Ballance, D. N. Stacey, A. M. Steane, D. M. Lucas, and D. T. C. Allcock. High-fidelity spatial and polarization addressing of $^{43}\text{Ca}^+$ qubits using near-field microwave control. *Physical Review A*, 95(2):022337, 2017.
- [48] T.P. Harty, M.A. Sepiol, D.T.C. Allcock, C.J. Ballance, J.E. Tarlton, and D.M. Lucas. High-fidelity trapped-ion quantum logic using near-field microwaves. *Physical Review Letters*, 117(14):140501, 2016.

- [49] M. Johanning, A. Braun, N. Timoney, V. Elman, W. Neuhauser, and Chr. Wunderlich. Individual addressing of trapped ions and coupling of motional and spin states using rf radiation. *Physical Review Letters*, 102(7):073004, 2009.
- [50] D. Hayes, D. N. Matsukevich, P. Maunz, D. Hucul, Q. Quraishi, S. Olmschenk, W. Campbell, J. Mizrahi, C. Senko, and C. Monroe. Entanglement of Atomic Qubits Using an Optical Frequency Comb. *Physical Review Letters*, 104(14):140501, 2010.
- [51] J. Mizrahi. *Ultrafast Control of Spin and Motion in Trapped Ions*. PhD thesis, University of Maryland, 2013.
- [52] Klaus Mølmer and Anders Sørensen. Multiparticle entanglement of hot trapped ions. *Physical Review Letters*, 82(9):1835–1838, 1999.
- [53] T. Choi, S. Debnath, T. A. Manning, C. Figgatt, Z.-X. Gong, L.-M. Duan, and C. Monroe. Optimal quantum control of multimode couplings between trapped ion qubits for scalable entanglement. *Physical Review Letters*, 112(19):190502, 2014.
- [54] M-Labs. sinara: Hardware designs for the ARTIQ DRTIO ecosystem (Sayma, Metlino, Kasli), 2018. <https://github.com/m-labs/sinara>.
- [55] J. I. Cirac and P. Zoller. Quantum Computations with Cold Trapped Ions. *Physical Review Letters*, 74(20):4091–4094, 1995.
- [56] E. Solano, R. L. de Matos Filho, and N. Zagury. Deterministic bell states and measurement of the motional state of two trapped ions. *Physical Review A*, 59(4):R2539–R2543, 1999.
- [57] S.-L. Zhu, C. Monroe, and L.-M. Duan. Trapped ion quantum computation with transverse phonon modes. *Physical Review Letters*, 97(5):050505, 2006.
- [58] K. Kim, M.-S. Chang, R. Islam, S. Korenblit, L.-M. Duan, and C. Monroe. Entanglement and tunable spin-spin couplings between trapped ions using multiple transverse modes. *Physical Review Letters*, 103(12):120502, 2009.
- [59] S.-L. Zhu, C. Monroe, and L.-M. Duan. Arbitrary-speed quantum gates within large ion crystals through minimum control of laser beams. *Europhysics Letters*, 73(4):485, 2006.
- [60] D. Deutsch. Quantum computational networks. *Proceedings of the Royal Society of London A*, 425(1868):73–90, 1989.
- [61] Michael Reck, Anton Zeilinger, Herbert J. Bernstein, and Philip Bertani. Experimental realization of any discrete unitary operator. *Physical Review Letters*, 73(1):58–61, 1994.

- [62] David P. DiVincenzo. Two-bit gates are universal for quantum computation. *Physical Review A*, 51(2):1015–1022, 1995.
- [63] 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. *Physical Review A*, 52(5):3457–3467, 1995.
- [64] P. O. Boykin, T. Mor, M. Pulver, V. Roychowdhury, and F. Vatan. On universal and fault-tolerant quantum computing: a novel basis and a new constructive proof of universality for shor’s basis. In *40th Annual Symposium on Foundations of Computer Science (Cat. No.99CB37039)*, pages 486–494, 1999.
- [65] Michael A. Nielsen and Isaac L. Chuang. *Quantum Computation and Quantum Information: 10th Anniversary Edition*. Cambridge University Press, New York, NY, USA, 10th edition, 2011.
- [66] David P. DiVincenzo. Quantum gates and circuits. *Proceedings of the Royal Society of London A: Mathematical, Physical and Engineering Sciences*, 454(1969):261–276, 1998.
- [67] M-Labs. artiq: Next-generation control system for quantum information experiments, 2018. <https://github.com/m-labs/artiq>.
- [68] pyIonControl. IonControl, 2017. <https://github.com/pyIonControl/IonControl>.
- [69] Marcello Benedetti, Delfina Garcia-Pintos, Yunseong Nam, and Alejandro Perdomo-Ortiz. A generative modeling approach for benchmarking and training shallow quantum circuits. *arXiv:1801.07686*, 2018.
- [70] Microsoft. LIQUi>: The language integrated quantum operations simulator. <http://stationq.github.io/Liquid/>.
- [71] Rigetti Computing. Forest. <https://www.rigetti.com/forest>.
- [72] Tommaso Toffoli. Reversible computing. In *Automata, Languages and Programming*, Lecture Notes in Computer Science, pages 632–644. Springer, Berlin, Heidelberg, 1980.
- [73] Lieven M. K. Vandersypen, Matthias Steffen, Mark H. Sherwood, Costantino S. Yannoni, Gregory Breyta, and Isaac L. Chuang. Implementation of a three-quantum-bit search algorithm. *Applied Physics Letters*, 76(5):646–648, 2000.
- [74] Dmitri Maslov. Basic circuit compilation techniques for an ion-trap quantum machine. *New Journal of Physics*, 19(2):023035, 2017.

- [75] David G. Cory, Mark D. Price, and Timothy F. Havel. Nuclear magnetic resonance spectroscopy: An experimentally accessible paradigm for quantum computing. *Physica D*, 120(1):82–101, 1998.
- [76] T. Monz, K. Kim, W. Hänsel, M. Riebe, A. S. Villar, P. Schindler, M. Chwalla, M. Hennrich, and R. Blatt. Realization of the quantum Toffoli gate with trapped ions. *Physical Review Letters*, 102(4):040501, 2009.
- [77] Dmitri Maslov. Advantages of using relative-phase Toffoli gates with an application to multiple control Toffoli optimization. *Physical Review A*, 93(2):022311, 2016.
- [78] Edward Fredkin and Tommaso Toffoli. Conservative logic. *International Journal of Theoretical Physics*, 21(3):219–253, 1982.
- [79] Jiangfeng Du, Ping Zou, Xinhua Peng, Daniel K. L. Oi, L. C. Kwek, C. H. Oh, and Artur Ekert. Experimental quantum multimeter and one-qubit fingerprinting. *Physical Review A*, 74(4):042319, 2006.
- [80] Raj B. Patel, Joseph Ho, Franck Ferreyrol, Timothy C. Ralph, and Geoff J. Pryde. A quantum Fredkin gate. *Science Advances*, 2(3):e1501531, 2016.
- [81] Takafumi Ono, Ryo Okamoto, Masato Tanida, Holger F. Hofmann, and Shigeki Takeuchi. Implementation of a quantum controlled-SWAP gate with photonic circuits. *Scientific Reports*, 7:45353, 2017.
- [82] D. Maslov, G. W. Dueck, D. M. Miller, and C. Negrevergne. Quantum circuit simplification and level compaction. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 27(3):436–444, 2008.
- [83] Patrick Hayden and John Preskill. Black holes as mirrors: quantum information in random subsystems. *Journal of High Energy Physics*, 2007(9):120–120, 2007.
- [84] Beni Yoshida and Norman Y. Yao. Disentangling scrambling and decoherence via quantum teleportation. *forthcoming*, 2018.
- [85] Beni Yoshida and Alexei Kitaev. Efficient decoding for the Hayden-Preskill protocol. *arXiv:1710.03363*, 2017.
- [86] Lov K. Grover. Quantum Mechanics Helps in Searching for a Needle in a Haystack. *Physical Review Letters*, 79(2):325–328, 1997.
- [87] Michel Boyer, Gilles Brassard, Peter Høyer, and Alain Tapp. Tight bounds on quantum searching. *Fortschritte der Physik*, 46(4):493–505, 1998.
- [88] C. Bennett, E. Bernstein, G. Brassard, and U. Vazirani. Strengths and weaknesses of quantum computing. *SIAM Journal on Computing*, 26(5):1510–1523, 1997.

- [89] F. Magniez, M. Santha, and M. Szegedy. Quantum algorithms for the triangle problem. *SIAM Journal on Computing*, 37(2):413–424, 2007.
- [90] C. Dürr, M. Heiligman, P. Høyer, and M. Mhalla. Quantum query complexity of some graph problems. *SIAM Journal on Computing*, 35(6):1310–1328, 2006.
- [91] Isaac L. Chuang, Neil Gershenfeld, and Mark Kubinec. Experimental Implementation of Fast Quantum Searching. *Physical Review Letters*, 80(15):3408–3411, 1998.
- [92] N. Bhattacharya, H. B. van Linden van den Heuvell, and R. J. C. Spreeuw. Implementation of Quantum Search Algorithm using Classical Fourier Optics. *Physical Review Letters*, 88(13):137901, 2002.
- [93] K.-A. Brickman, P. C. Haljan, P. J. Lee, M. Acton, L. Deslauriers, and C. Monroe. Implementation of Grover’s quantum search algorithm in a scalable system. *Physical Review A*, 72(5):050306(R), 2005.
- [94] P. Walther, K. J. Resch, T. Rudolph, E. Schenck, H. Weinfurter, V. Vedral, M. Aspelmeyer, and A. Zeilinger. Experimental one-way quantum computing. *Nature*, 434(7030):169–176, 2005.
- [95] L. DiCarlo, J. M. Chow, J. M. Gambetta, Lev S. Bishop, B. R. Johnson, D. I. Schuster, J. Majer, A. Blais, L. Frunzio, S. M. Girvin, and R. J. Schoelkopf. Demonstration of two-qubit algorithms with a superconducting quantum processor. *Nature*, 460(7252):240–244, July 2009.
- [96] Stefanie Barz, Elham Kashefi, Anne Broadbent, Joseph F. Fitzsimons, Anton Zeilinger, and Philip Walther. Experimental Demonstration of Blind Quantum Computing. *Science*, 335(6066):303–308, 2012.
- [97] T.A. Manning. *Quantum information processing with trapped ion chains*. PhD thesis, University of Maryland, 2014.
- [98] Klaus Mølmer, Larry Isenhower, and Mark Saffman. Efficient Grover search with Rydberg blockade. *Journal of Physics B: Atomic, Molecular and Optical Physics*, 44(18):184016, 2011.
- [99] J. Chiaverini, J. Britton, D. Leibfried, E. Knill, M. D. Barrett, R. B. Blakestad, W. M. Itano, J. D. Jost, C. Langer, R. Ozeri, T. Schaetz, and D. J. Wineland. Implementation of the Semiclassical Quantum Fourier Transform in a Scalable System. *Science*, 308(5724):997–1000, 2005.
- [100] R. Cleve and J. Watrous. Fast parallel circuits for the quantum fourier transform. In *Proceedings 41st Annual Symposium on Foundations of Computer Science*, pages 526–536, 2000.

- [101] D. Maslov. Linear depth stabilizer and quantum fourier transformation circuits with no auxiliary qubits in finite neighbor quantum architectures. *Physical Review A*, 76(5):052310, 2007.
- [102] Austin G. Fowler, Simon J. Devitt, and Lloyd C. L. Hollenberg. Implementation of Shor’s algorithm on a linear nearest neighbour qubit array. *Quantum Information and Computation*, 4:237–251, 2004.
- [103] Donny Cheung, Dmitri Maslov, Jimson Mathew, and Dhiraj K. Pradhan. On the design and optimization of a quantum polynomial-time attack on elliptic curve cryptography. In *Theory of Quantum Computation, Communication, and Cryptography*, Lecture Notes in Computer Science, pages 96–104. Springer, Berlin, Heidelberg, 2008.
- [104] Tiago Januario and Sebastian Urrutia. An edge coloring heuristic based on vizing’s theorem. In *Proceedings of the Brazilian Symposium on Operations Research*, pages 3994–4002, 2012.
- [105] Ian D. Kivlichan, Jarrod McClean, Nathan Wiebe, Craig Gidney, Aln Aspuru-Guzik, Garnet Kin-Lic Chan, and Ryan Babbush. Quantum simulation of electronic structure with linear depth and connectivity. *Physical Review Letters*, 120(11):110501, 2018.
- [106] S. Korenblit, D. Kafri, W. C. Campbell, R. Islam, E. E. Edwards, Z.-X. Gong, G.-D. Lin, L.-M. Duan, J. Kim, K. Kim, and C. Monroe. Quantum simulation of spin models on an arbitrary lattice with trapped ions. *New Journal of Physics*, 14(9):095024, 2012.
- [107] Todd J. Green and Michael J. Biercuk. Phase-modulated decoupling and error suppression in qubit-oscillator systems. *Physical Review Letters*, 114(12):120502, 2015.
- [108] Shengtao Wang. *Applications of Atomic Systems in Quantum Simulation, Quantum Computation and Topological Phases of Matter*. PhD thesis, University of Michigan, 2017.
- [109] Chingiz Kabytayev, Todd J. Green, Kaveh Khodjasteh, Michael J. Biercuk, Lorenza Viola, and Kenneth R. Brown. Robustness of composite pulses to time-dependent control noise. *Physical Review A*, 90(1):012316, 2014.
- [110] Kenneth R. Brown, Aram W. Harrow, and Isaac L. Chuang. Arbitrarily accurate composite pulse sequences. *Physical Review A*, 70(5):052318, 2004.
- [111] C. A. Sackett, D. Kielpinski, B. E. King, C. Langer, V. Meyer, C. J. Myatt, M. Rowe, Q. A. Turchette, W. M. Itano, D. J. Wineland, and C. Monroe. Experimental entanglement of four particles. *Nature*, 404(6775):256–259, 2000.
- [112] Richard P. Feynman. Quantum mechanical computers. *Optics News*, 11(2):11–20, 1985.

- [113] Daniel M. Greenberger, Michael A. Horne, and Anton Zeilinger. Going beyond bells theorem. In *Bells Theorem, Quantum Theory and Conceptions of the Universe*, Fundamental Theories of Physics, pages 69–72. Springer, Dordrecht, 1989.
- [114] Ketan N. Patel, Igor L. Markov, and John P. Hayes. Optimal synthesis of linear reversible circuits. *Quantum Information and Computation*, 8(3):0282–0294, 2008.
- [115] Dmitri Maslov and Yunseong Nam. Use of global interactions in efficient quantum circuit constructions. *New Journal of Physics*, 20(3):033018, 2018.
- [116] Mark Hillery, Vladimr Buek, and Andr Berthiaume. Quantum secret sharing. *Physical Review A*, 59(3):1829–1834, 1999.
- [117] G.-D. Lin, S.-L. Zhu, R. Islam, K. Kim, M.-S. Chang, S. Korenblit, C. Monroe, and L.-M. Duan. Large-scale quantum computation in an anharmonic linear ion trap. *Europhysics Letters*, 86(6):60004, 2009.
- [118] C.J. Ballance, T.P. Harty, N.M. Linke, M.A. Sepiol, and D.M. Lucas. High-Fidelity Quantum Logic Gates Using Trapped-Ion Hyperfine Qubits. *Physical Review Letters*, 117(6):060504, 2016.
- [119] J.P. Gaebler, T.R. Tan, Y. Lin, Y. Wan, R. Bowler, A.C. Keith, S. Glancy, K. Coakley, E. Knill, D. Leibfried, and D.J. Wineland. High-Fidelity Universal Gate Set for $^9\text{Be}^+$ Ion Qubits. *Physical Review Letters*, 117(6):060505, 2016.
- [120] Ye Wang, Mark Um, Junhua Zhang, Shuoming An, Ming Lyu, Jing-Ning Zhang, L.-M. Duan, Dahyun Yum, and Kihwan Kim. Single-qubit quantum memory exceeding ten-minute coherence time. *Nature Photonics*, 11(10):646–650, 2017.
- [121] D. Kielpinski, C. Monroe, and D. J. Wineland. Architecture for a large-scale ion-trap quantum computer. *Nature*, 417(6890):709–711, 2002.
- [122] I.V. Inlek, C. Crocker, M. Lichtman, K. Sosnova, and C. Monroe. Multi-species trapped-ion node for quantum networking. *Physical Review Letters*, 118(25):250502, 2017.