Fault-Tolerant Quantum Error Correction with Trapped-Ion Quantum Bits

Dissertation

zur Erlangung des akademischen Grades

„Doktor der Naturwissenschaften“

am Fachbereich Physik, Mathematik und Informatik (FB 08)
der Johannes Gutenberg-Universität Mainz

vorgelegt von

Janine Hilder geb. Nicodemus

geboren in Idar-Oberstein

JOHANNES GUTENBERG UNIVERSITÄT MAINZ

JG|U

Mainz, Februar 2022

1. Gutachter:	Prof. Dr. Ferdinand Schmidt-Kaler
2. Gutachter:	Prof. Dr. Peter van Loock

Datum der mündlichen Prüfung: 01. Dezember 2022

Ich versichere, dass ich die Arbeit selbstständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt sowie Zitate kenntlich gemacht habe.

Ich erkläre hiermit, dass diese Arbeit weder ganz noch teilweise an einer anderen Universität zur Erlangung eines anderen akademischen Grades eingereicht wurde und wird.

Mainz, Februar 2022

Janine Hilder geb. Nicodemus
QUANTUM
Institut für Physik
Staudingerweg 7
Johannes Gutenberg-Universität
D-55128 Mainz
hilder@uni-mainz.de

Abstract

Trapped-ion quantum information processing is among the most promising candidates to realize a scalable quantum computer. A segmented Paul trap can be used to move ions in and out of storage and processing regions via dynamic register reconfiguration operations, enabling effective all-to-all connectivity. The processing region is dedicated to perform laser-driven operations, such as single-qubit rotations and two-qubit entangling gates. To achieve the long-term goal of a large-scale fault-tolerant quantum computer, it is of crucial importance to realize quantum error correction.

This thesis focuses on the experimental realization of a fault-tolerant (FT) weight-4 parity check measurement (PCM) scheme on a trapped-ion quantum processor node. The scheme presented here uses only minimal resource overhead in the form of one additional 'flag' qubit, to detect errors that would proliferate onto the data qubit register as uncorrectable weight-2 errors. This parity check measurement is an important building block in a broad class of resource-efficient flag-based quantum error correction protocols, such as the topological color code. The experimental result presented is one of the first realizations of the FT PCM scheme on a shuttling-based trapped-ion quantum computing architecture. A parity measurement fidelity of 92.3(2)% is achieved, which is increased to 93.2(2)% upon flag-qubit conditioning, exceeding the bare parity fidelity by 4.5 standard errors. Injection of bit- and phase-flip errors shows that the scheme is able to reliably intercept faults. For holistic benchmarking, an entanglement witnessing scheme is used, to verify the generation of six-qubit multipartite entanglement, involving all ions participating in the fault-tolerant parity measurement.

Within the work presented here, improvements of the register reconfiguration operations were carried out, in order to reduce the motional excitation in quantum circuits with a high number of transport, separation/merge and positional ion swap operations, such as the parity measurement scheme.

Taking into account the architectural features of the shuttling-based trapped-ion quantum processor, such as effective all-to-all connectivity and no operational cross-talk, the demonstrated building block of flag-based fault-tolerant quantum error correction lays out a clear path towards scalable fault-tolerant quantum computing.

Zusammenfassung

Die Quanteninformationsverarbeitung mit gefangenen Ionen gehört zu den vielversprechendsten Kandidaten für die Realisierung eines skalierbaren Quantencomputers. Eine segmentierte Paul-Falle kann verwendet werden, um Ionen durch dynamische Register-Rekonfigurationsoperationen in und aus Speicher- und Prozessorregionen zu bewegen, was eine effektive volle Konnektivität ermöglicht. Die Prozessorregion ist für die Durchführung von Laser getriebenen Operationen wie Ein-Qubit Rotationen und Zwei-Qubit-Verschränkungsgattern vorgesehen. Um den Weg für das langfristige Ziel eines skalierbaren, fehlertoleranten Quantencomputers zu ebnen, ist es von entscheidender Bedeutung, Quantenfehlerkorrektur zu realisieren. Diese Arbeit konzentriert sich auf die experimentelle Realisierung eines fehlertoleranten Gewicht-4-Paritätsprüfungs-Messverfahrens auf einem Quantenverarbeitungsknoten mit gefangenen Ionen. Das hier vorgestellte Schema verwendet nur einen minimalen Ressourcen-Overhead in Form eines zusätzlichen "Flag"-Qubits, um Fehler zu erkennen, die sich im Daten-Qubit-Register als unkorrigierbare Gewicht-2-Fehler ausbreiten würden. Diese Paritätsprüfungsmessung ist ein wichtiger Baustein in einer breiten Klasse von ressourceneffizienten flag-basierten Quantenfehlerkorrekturprotokollen, wie z.B. dem topologischen Farbcode. Das präsentierte experimentelle Resultat ist eine der ersten Realisierungen dieses Schemas auf einer rekonfigurations-basierten Quantencomputerarchitektur mit gefangenen Ionen. Die Parität des Daten-Qubit-Registers wird auf das Syndrom-Qubit abgebildet, um die Paritätsinformation auf eine quantenmechanisch nicht-zerstörende Weise zu erfassen. In dem Algorithmus der mit 6 Ionen und 6 Zwei-Qubit Gattern durchgeführt wird, ist es gelungen eine Paritätsmessgenauigkeit von 92,3(2)% zu erreichen. Diese erhöht sich bei der Flag-Qubit-Konditionierung auf 93,2(2)%, eine Verbesserung um 4,5 Standardabweichungen. Um die Effizienz des Algorithmus zu prüfen wurden Bit- und Phasenflip-Fehler induziert und zuverlässig abgefangen. Für ein ganzheitliches Benchmarking wird ein Schema verwendet, um die Erzeugung von Sechs-Qubit-Verschränkung zu verifizieren, an der alle an der fehlertoleranten Paritätsmessung beteiligten Ionen beinhaltet sind.
Im Rahmen dieser Arbeit wurden Verbesserungen der Register-Rekonfigurationsoperationen durchgeführt, um die Bewegungsanregung in Quantenschaltkreisen mit einer hohen Anzahl von Transport-, Separations-/Merge- und Positionstauschoperationen, wie dem Paritätsmessungsschema, zu reduzieren.
Unter Berücksichtigung der architektonischen Merkmale des rekonfigurations-basierten Quantenprozessors mit gefangenen Ionen, wie z.B. effektiver voller Konnektivität und keinem operativen Übersprechen, bereitet der demonstrierte Baustein von flag-basierten fehlertoleranten Quantenfehlerkorrekturverfahren den Weg in Richtung skalierbarer fehlertoleranter Quantencomputer.

Contents

Chapter 1

Introduction

Technology development towards the end of the 20^{th} century was dominated by the advances of classical computers. While first computers filled entire rooms, the invention of the integrated circuit (IC) allowed for incredible performance improvements and reduction of hardware size. Today, impressive computational capabilities are realised in compact devices, such as mobile phones and smart watches, while larger high performance computers include millions of computing cores. Throughout the last decades, the number of transistors on integrated circuits has continuously increased, roughly following Moore's law of doubling every two years [1]. This has been achieved by minimizing the size of single transistors. Nowadays, this becomes more and more of a challenge, since the dimensions are so small that quantum effects have to be considered.

Quantum computing, based on utilizing quantum effects, could outperform classical processors in certain tasks [2–5]. As proposed by Feynman in the 1980s [6], a quantum computer can be used to efficiently simulate quantum systems, which is not possible using a classical computer. Based on the idea of a new type of computer using quantum bits and entangling operations, Deutsch proposed a universal quantum computer [7], which can perform algorithms efficiently where it is not possible on a universal Turing machine [8]. Following the proposal of a universal quantum computer, many suitable algorithms were proposed, such as Shor's algorithm for the efficient factorization of large numbers [9] and Grover's algorithm to search large unstructured databases [10].

Shortly after, single trapped ions carrying the quantum information and manipulated with laser beams were found by Ignacio Cirac and Peter Zoller to serve as a suitable physical system to realize quantum algorithms [11]. The requirements that a quantum system has to fulfill in order to implement a universal quantum computer were discussed by DiVincenzo in [12].

The field of experimental quantum computing quickly emerged and research is focusing on a variety of quantum systems, such as neutral atoms trapped in optical lattices [13], magneto-optical traps [14] or optical cavity systems [15], superconducting qubits based on Josephson junctions [16–18], photonic quantum systems using linear optics [19, 20] or weak cross-Kerr coupling [21]. The trapped-ion quantum systems are among the most promising candidates to realize a universal quantum computer, but also to be used as noisy intermediate-scale quantum (NISQ) devices, advantageous for specific tasks.

Today's trapped-ion quantum computing devices are based on the development of an electrodynamic mass spectrometer by Wolfgang Paul [22]. Very soon after this development, the technology was used to also trap ions [23, 24]. There are two common techniques to trap charged particles, the Penning trap [25, 26] and Paul trap [27]. For this work I will focus on the latter one and its usage in scalable trapped-ion quantum processors. In the experimental setup presented within this thesis, the qubit is encoded in the spin of the outer valence electron of a $^{40}Ca^{+}$ ion (see Sec. 2.1).

Quantum Error Correction

Coherent quantum states are very fragile and it was intensely discussed if it would be possible to maintain a large multi-qubit coherent state for a long enough duration, in order to realize any quantum algorithm. Noisy intermediate-scale quantum computing devices can be used for proof-of-principle demonstrations, but with increasing number of qubits and gates the result of the quantum computing device can not easily be verified anymore. To ensure that a computation gives a reasonable result, one has to trust the computation process itself. This can be achieved by performing error correction throughout the computation. Classical error correction is not applicable to quantum systems, but soon there were multiple proposals of error correction codes for quantum systems [28–30]. Implementing a universal quantum computer to solve problems beyond the capabilities of classical computers requires quantum error correction (QEC) protocols, to mitigate insufficient fidelities of the operational building blocks. In QEC codes, multiple physical qubits, such as atomic ions, are used to encode a single logical qubit. The logical qubit is encoded in entangled states of the physical qubit register. One type of encoding is given by topological codes. The qubits can be arranged on a lattice and the observables can be defined as local stablizers. Detection and correction of errors is possible using only local quantum operations. Errors can be detected by performing quantum non-demolition (QND) parity check measurements (PCM) on the logical qubits [31–34]. The parity information is mapped to an additional auxiliary qubit, which can be measured without destruction of the logical qubit state [35]. Projective measurements on this auxiliary qubit discretize the error if present and thus allow to detect and correct subsequently. The parity information mapping requires entangling gates, which have to be treated as not fault-free operations, thus possibly introducing additional errors during the attempt of error detection and correction. Fault-tolerant (FT) QEC schemes have been developed to address this issue and prevent the unnoticed corruption of the data qubit register during a QEC cycle [36]. The first FT PCM schemes required many additional qubits, as many as the parity check generator with maximum weight [37, 38]. The schemes have been adapted using flag qubits, significantly reducing the overhead of additional qubits and gate operations [39–44, 44–46]. Recently, multiple implementations of fault-tolerant quantum error correction schemes have been shown [47–49].

Scaling Trapped-Ion Quantum Computers

A universal quantum computer, capable of outperforming classical computers for specific tasks, requires scaling of physical qubit systems to much larger sizes than currently realized. There are plenty of interesting applications to perform on noisy intermediate-scale quantum (NISQ) processors, such as quantum chemical simulations, but applications such as Shor's algorithm require thousands or even millions of physical qubits. Considering the fragility of coherent quantum systems, it is of crucial importance to perform fault-tolerant quantum error correction to maintain quantum states throughout the computation and realize a large-scale quantum computer. Error correction codes with multiple encoding layers increase the amount of required physical qubits even further, while being able to tolerate a higher number of errors on the physical qubit register.

An intrinsic advantage of trapped-ion qubits is the non-existence of fabrication variance. Thus, the challenge is not the production of the qubits, but the dedicated control of systems with a large number of qubits, while at the same time protecting the qubits from undesired coupling to the environment. Scaling to a large number of qubits can roughly be divided into two main approaches, increasing the number of physical qubits within one quantum processing unit or interconnecting multiple quantum processing units.

A small number of ions can be confined within one common potential well. This is suitable up to a number of ions ranging from $N = 10$ to 100. Increasing the number of ions in one well does come at the cost of many modes of collective motion, increasing the cooling effort and introducing errors due to heating of motional modes. Keeping a similar trap frequency, the distance between the ions is going to decrease when increasing the number of ions, rendering the individual addressing of single ions within the chain at low crosstalk levels increasingly difficult. A solution to the practical limits of this scheme was proposed by Kielpinski et al. as the quantum charge-coupled device (QCCD) [50–52]. The linear Paul trap can be segmented, to be able to form multiple individual trapping potentials on the same trap chip. In a linear segmented Paul trap, this makes it possible to divide the qubit register in multiple smaller subsets, a technique used in this work. Smaller ion crystals are easier to control and crosstalk during gate operations is reduced to a negligible level due to the large distance of stored ions with respect to the processing region. Entanglement over a larger register size is achieved via subsequent entanglement of small-size crystals. To hold an even higher number of ions and increase flexibility, the linear structure can be extended to 2D. Ions can be transported via a two-dimensional grid structure along linear trap regions connected via junctions. This requires precise control of the trapping potential, especially close to the junctions in order to transport the ions while maintaining low motional excitation. High-fidelity transports via X- [53] and T-junctions [54] have been successfully demonstrated. Further scaling of trapped-ion quantum systems can be realized via movement of ions over long distances between computing modules and trap chips [55, 56] or using ion-photon entanglement [57, 58].

Thesis Structure

This work presents one of the first experimental implementations of a fault-tolerant parity check measurement (FT PCM) on a shuttling-based trapped-ion quantum processing node [59]. The FT PCM represents a key building block for a broad class of flag-based QEC protocols, such as topological color codes. Our hardware setup is based on atomic ions trapped in a micro-structured, segmented Paul trap, allowing to perform qubit register reconfiguration operation. These shuttling operations enable an effective all-to-all connectivity without operational cross-talk, providing important prerequisits for fault-tolerant QEC circuit blocks. Chapter 2 introduces the $^{40}Ca^{+}$ ion as the physical qubit, the Paul trap and the laser-driven operations, followed by the description of the experimental setup in Chapter 3. After a more general introduction to the ion crystal reconfiguration operations in Chapter 4, improvements to the operations on the current hardware setup with corresponding experimental measurement results are presented in Chapter 5. Chapter 6 gives a brief overview of quantum error correction concepts, to prepare for the experimental results on fault-tolerant parity check measurements in the following Chapter 7 and generation of genuine multipartite entanglement in Chapter 8. Finally, Chapter 9 gives an outlook of the next generation hardware setup.

Chapter 2

The $^{40}Ca^{+}$ Spin Qubit

There are several suitable ions to choose from when encoding a qubit utilizing a trapped ion, providing different advantages and challenges. $^{40}Ca^{+}$ has the advantage that all transitions are near the visible range of the spectrum. This allows for the use of commercially available diode lasers, significantly reducing the maintenance effort of the optical setup. Lighter ions such as $^{9}Be^{+}$ or $^{25}Mg^{+}$ feature dipole transitions in the deep UV, which challenges the use of typical fiber optical elements and does not facilitate the detection of the ion fluorescence. Additionally, $^{40}Ca^{+}$ features an intermediate mass compared to heavy elements such as $^{137}Ba^{+}$ and $^{171}Yb^{+}$ also used for trapped-ion quantum information. The comparative low mass of $^{40}Ca^{+}$ is favorable when performing shuttling operations in micro-structured ion traps. Most important for quantum computing is that the ion of choice features long lived levels, which can be used to store quantum information. One option is to employ hyperfine levels, typical examples are $^{9}Be^{+}$ or $^{171}Yb^{+}$, or to use Zeeman sublevels of the ground state in case of elements with nuclear spin I=0. A different scheme to encode the qubit is to use the electronic ground state and an excited metastable state (optical qubit). While the optical qubit encoded in the $S_{1/2}$ and $D_{5/2}$ levels of $^{40}Ca^{+}$ does require a highly stable laser with a line width lower than 200 mHz [60], the Raman transition of the Zeeman qubit does not require this technically demanding stability of the driving laser. Two beams are utilized to drive stimulated Raman transitions. The absolute frequency and optical phase stability of the drive are not of critical importance, only relative stability of the frequency and phase of the two beams with respect to each other is required for gate operations. The rather small qubit frequency of only $2\pi \times$ 10 MHz can be coherently bridged using AOMs. Another advantage of the Zeeman qubit is the non-existence of loss channels, other than actual ion losses in the trap. There is no population transfer to other sublevels during e.g. gate operations. Since $^{40}Ca^{+}$ features a nuclear spin of 0, there is no hyperfine structure. This simplifies the level scheme and therefore also cooling, reset of qubit states and fluorescence detection. A disadvantage of the Zeeman qubit in contrast to the optical qubit [60] is the higher magnetic field sensitivity. The former one is about six times more sensitive in qubit frequency change with respect to magnetic field change compared to the latter one. This can be overcome by protecting the qubit from magnetic field fluctuations in its surrounding using magnetic shielding enclosures and active stabilization. Also, the magnetically sensitive qubit can be of use to measure magnetic fields with a

high accuracy of down to 300 fT and sensitivities down to 12 pT/$\sqrt{\text{Hz}}$ [61]. The second disadvantage is that Zeeman qubits can not be read out directly, but require a technique named *shelving*. The qubit state is spin-selectively transferred to a metastable state, such that even in the absence of direct readout, a very low readout infidelity of $< 0.1\%$ is achieved. The positively charged $^{40}Ca^{+}$ ions are generated by using a two-photon ionization process. An "atomic" oven is heated by typically 2.8 A, generating an effusive beam of neutral ^{40}Ca atoms. These atoms are then ionized using a resonant step at 423 nm and a second non-resonant step at around 374 nm, as shown in Fig. 2.1 **b)**.
The following chapter focuses on qubit encoding, cooling procedures, RF trap and vibrational modes of ions confined within a Paul trap. Furthermore, light-motion coupling is discussed, as well as the stimulated Raman transitions and laser-driven quantum gate operations.

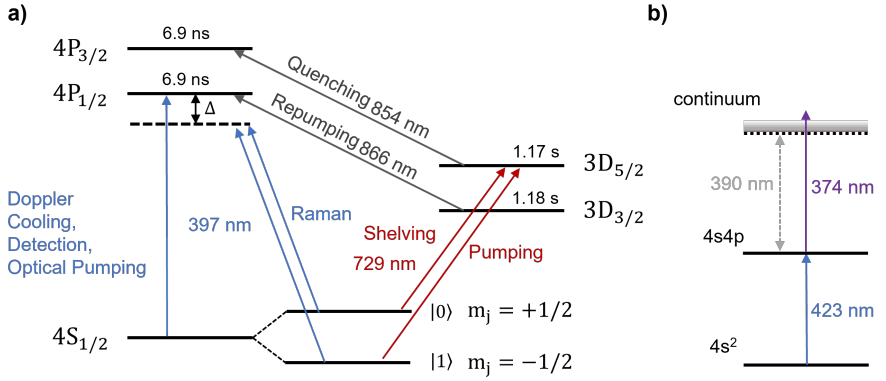

Figure 2.1: Relevant levels and transitions in $^{40}Ca^{+}$. **a)** Relevant atomic transitions of $^{40}Ca^{+}$ with the corresponding laser-driven operations, including the relevant decay times [62–64]. **b)** Scheme of the two-photon ionization of ^{40}Ca atom, to generate the $^{40}Ca^{+}$ ion.

2.1 Qubit Encoding

The qubit is encoded in the $S_{1/2}$ ground state of the outer valence electron of the $^{40}Ca^{+}$ ion. A quantizing magnetic field of about 3.7 G creates a Zeeman splitting of around $2\pi \times$ 10 MHz and therefore provides two levels with infinite T_1 lifetime to encode the qubit, which are $|0\rangle \equiv |S_{1/2}, m_J = +1/2\rangle$ and $|1\rangle \equiv |S_{1/2}, m_J = -1/2\rangle$. There happens to be no spontaneous decay from these qubit levels. This encoding is known as the *Zeeman qubit*. The magnetic field strength is chosen to be high enough to avoid spectral crowding on the $S_{1/2} \leftrightarrow D_{5/2}$ transition used for spin-selective readout, but smaller than the natural linewidth of the $S_{1/2} \leftrightarrow P_{1/2}$ transition, which is favorable for Doppler cooling and detection. Fig 2.1 **b)** shows the relevant atomic transitions used for spin-selective readout, laser cooling and laser-driven quantum gates. The laser

driving the Raman transition features 397 nm, with a detuning to the $P_{1/2}$ transition of typically $\Delta \approx 2\pi \times$ 1 THz. Single-qubit rotations, driven by two co-propagating Raman laser beams, are typically operated at a Rabi frequency of $\Omega \approx$ 800 kHz, corresponding to a time of roughly 4 µs for a π-rotation.

2.2 Doppler Cooling

To perform quantum computing, it is of crucial importance to cool the ions close to the motional ground state. This can be achieved by using multiple steps of cooling, which includes Doppler cooling of the trapped ions, followed by resolved sideband cooling from the Doppler limit to the motional ground state. A detailed discussion of trapped-ion cooling techniques can be found in a large number of publications, for example in [65–68]. Therefore, only a brief summary of the used cooling techniques will be presented here. Doppler cooling of $^{40}Ca^{+}$ is realized using the $S_{1/2} \leftrightarrow P_{1/2}$ transition near 397 nm, referred to as *cycling transition*. A laser near 866 nm is used to continuously pump back the ion from the metastable $D_{3/2}$ state. The light is red detuned about half a linewidth with respect to the atomic resonance, such that it will be resonant in the reference frame of an ion moving towards the laser due to the Doppler effect. The fluorescence rate of an atom in a harmonic oscillator potential and irradiated by a laser beam with detuning $\delta \to \delta - k_x\nu(t)$, taking into account the Doppler effect, is given by

$$R(\nu) = \frac{\Gamma}{2}\frac{S}{1 + S + 4(\delta - k_x\nu(t))^2/\Gamma^2} \tag{2.1}$$

with saturation $S = 2\Omega^2/\Gamma^2$ and Γ being the natural linewidth of the transition to be cooled on. For the red detuned light used in the Doppler cooling process, the absorption rate increases if the ion's velocity has a component antiparallel to the cooling laser's wavevector. The photon absorption will transfer the ion into an excited state, from which it is going to decay and emit a photon again. Since the absorbed photons change the momentum in the laser beam direction and the emission can be handled as nearly isotropic, following a dipole emission pattern in a quantizing magnetic field, a net reduction of the momentum in the laser beam direction is achieved. This process can also be used to cool in the three trap dimensions using only a single laser beam, on the condition that the k-vector of the laser has a non-vanishing projection on all the trap axes. Optimal Doppler cooling, under assumption of an ideal two-level system, is achieved using the parameters $\delta \approx -\Gamma/2$ and typically a low saturation parameter of about $S \approx 0.2..0.4$. While this process is very efficient as a first cooling stage for ions which have just been trapped, the ions can not be cooled down to zero phonons due to the nature of this stochastic process. The achievable temperature for an ideal two-level system is given by the Doppler cooling limit [69]

$$T_D = \frac{\hbar\gamma}{2k_B} \tag{2.2}$$

with Boltzmann constant k_B. For $^{40}Ca^{+}$, taking into account the lifetime of the $P_{1/2}$ state of $\gamma^{-1} = 6.9$ ns, this results in an equilibrium temperature of $T_D \approx 550$ µK.

Based on the usual axial trap frequency of $2\pi \times 1.5$ MHz in our setup, this results in a theoretical minimum mean phonon number of $\bar{n} = 1/\left(e^{\hbar\omega/k_B T} - 1\right) \approx 7$ phonons [70]. This limit, however, does not imply the anisotropy of the emission pattern and neglects the micromotion in a realistic trap, the effect of the non-degenerate Zeeman levels, light polarization, as well as the multi-level system of the ion to be cooled. For $^{40}Ca^{+}$, there is a 6% probability, that the $P_{1/2}$ state decays into the metastable $D_{3/2}$ state instead of the $S_{1/2}$ ground state [64]. An additional laser beam around 866 nm is needed, to repump ions from the metastable state back to $P_{1/2}$, while driving the cycling transition. Typically, ions can be cooled down to around 10 phonons per mode using the Doppler cooling technique in our setup.

2.3 State Preparation and Readout

The initially prepared default state is chosen to be $|0\rangle \equiv |S_{1/2}, m_J = +1/2\rangle$. In order to achieve an initialization infidelity $< 0.1\%$, two stages of optical pumping are executed with light near 397 nm and 729 nm. The laser beam near 397 nm is σ^{+}-polarized and drives the $|S_{1/2}, m_J = -1/2\rangle \leftrightarrow |P_{1/2}, m_J = +1/2\rangle$ transition incoherently. Since the $P_{1/2}$ state decays into both $S_{1/2}$ ground states and the pumping provides spin-selectivity, the population accumulates in $|S_{1/2}, m_J = +1/2\rangle$. Using this process, a preparation fidelity of around 99% is typically reached. To achieve the above mentioned infidelity $< 0.1\%$, a second optical pumping process is needed utilizing light near 729 nm for coherent driving of the dipole-forbidden $|S_{1/2}, m_J = -1/2\rangle \leftrightarrow |D_{5/2}, m_J = +3/2\rangle$ transition. Four π-pulses at a duration of around 10 µs are executed, each followed by a 854 nm "quench" laser pulse for 4 µs, in order to deplete the metastable $D_{5/2}$ state with a lifetime of 1.18 s. This process transfers to the $P_{3/2}$ state, featuring a much shorter natural lifetime of only 6.9 ns. Starting from $|0\rangle$, the required input states are generated using single-qubit rotations as described in Sec. 2.9.1.

Spin-selective readout of the qubit state is realized by a combination of driving the *cycling transition* $S_{1/2} \leftrightarrow P_{1/2}$ and hiding one spin-state in the metastable $D_{5/2}$ state, where the population is not affected upon driving the cycling transition. This process, referred to as *shelving*, is performed by rapid adiabatic passage pulses on the sub-transitions $|S_{1/2}, m_J = +1/2\rangle \leftrightarrow |D_{5/2}, m_J = +1/2\rangle$ and $|S_{1/2}, m_J = +1/2\rangle \leftrightarrow |D_{5/2}, m_J = -3/2\rangle$. These are frequency chirped Gaussian-shaped laser pulses at 729 nm, with variation of the frequency by ±60 kHz and a pulse duration of 200 µs. The pulse power is chosen to maximize the rate of population transfer from $|0\rangle$ to the metastable state, while suppressing parasitic population transfer from $|1\rangle$. The long lifetime of the metastable state, compared to the lifetime of the $P_{1/2}$ state on the cycling transition, enables the option to shelve all participating ions before exposing any ion to the 397 nm detection light. During the 800 µs exposure to 397 nm light, scattered photons are collected using a photomultiplier tube (PMT), allowing for threshold-based discrimination of $|S_{1/2}, m_J = +1/2\rangle$ and $|S_{1/2}, m_J = -1/2\rangle$, and therefore $|0\rangle$ and $|1\rangle$ respectively. The typical acquisition time of 800 µs does reveal a mean number of about 15 photon counts on the PMT for a bright ion. For this data acquisition time, the threshold to distinguish a bright or dark state is typically

set to 3.5 photon counts. Since the $P_{1/2}$ state decays to $D_{3/2}$ with a probability of around 6%, an additional repump beam at 866 nm is needed, in order to avoid population trapping in the metastable $D_{3/2}$ state. After spin-selective state detection of all ions, a second detection cycle is executed including light near 854 nm, depleting the $D_{5/2}$ state. A bright ion is expected in case no ion loss event occured, validating the previous shot.

2.4 Micro-Structured Segmented RF Trap

The ions are trapped in a micro-structured segmented linear RF trap, a variation of a linear Paul trap [23], as schematically shown in Fig. 2.2. The confinement in three dimensions is achieved using a combination of ponderomotive forces created by an alternating radio frequency (RF) field in the y/z plane, as well as a direct current (DC) electrostatic confinement force along the x-axis, which will be defined as trap axis. Multiple, individually controlled DC electrodes are arranged along this trap axis, which allows to form multiple potential minima along x. More details about the dimensions and typical operation parameters can be found in Sec. 3.1.

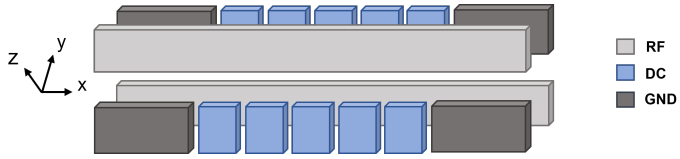

Figure 2.2: Schematic of a linear segmented Paul trap.

A detailed description of the dynamics of single trapped ions and the derivation of the following equations is found in a review [65]. The total potential along a direction $q \in \{x, y, z\}$ is given by

$$\Phi(q,t) = \alpha_q^{(RF)} q^2 U_{RF} \cos(\Omega_{RF} t) + \alpha_q^{(DC)} q^2 U_{DC} \tag{2.3}$$

with potential curvatures $\alpha_q^{(RF)}$ and $\alpha_q^{(DC)}$ in the corresponding direction, assuming a potential generated by a voltage of 1 V at the DC and RF electrode and the angular radio frequency Ω_{RF}. DC and RF potentials have to satisfy the Laplace equation $\Delta V = 0$ individually, leading to

$$\alpha_x^{(RF)} + \alpha_y^{(RF)} + \alpha_z^{(RF)} = 0 \tag{2.4}$$

$$\alpha_x^{(DC)} + \alpha_y^{(DC)} + \alpha_z^{(DC)} = 0 \tag{2.5}$$

and due to the chosen linear geometry of the trap (see Fig. 2.2)

$$\alpha_x^{(DC)} = \alpha_y^{(DC)} = -2\alpha_z^{(DC)} \equiv \alpha^{(DC)} \tag{2.6}$$

and

$$\alpha_y^{(RF)} = -\alpha_z^{(RF)}, \alpha_x^{(RF)} = 0. \tag{2.7}$$

The axial potential (x-axis) does not depend on the RF potential, but only on the DC. Therefore, the confinement along the trap axis can be fully controlled using only the DC segments. This is of importance when carrying out shuttling operations along the trap axis. The motion of a singly positive charged particle of mass m and elementary charge e, can be described by the Mathieu differential equations, along direction q

$$\frac{\mathrm{d}^2 q}{\mathrm{d}\tau^2} + (a_q - 2b_q \cos(2\tau))\, q = 0 \tag{2.8}$$

with

$$\tau = \frac{\Omega_{RF} t}{2} \qquad a_q = \frac{4|e|\alpha_q^{(DC)} U_{DC}}{m\Omega_{RF}^2} \qquad b_q = \frac{2|e|\alpha_q^{(RF)} U_{RF}}{m\Omega_{RF}^2}. \tag{2.9}$$

where U_{DC} and U_{RF} are the corresponding voltages applied to DC and RF electrodes. In the limit of $a_q < b_q^2 \ll 1$, a stable solution for this equations can be found to be

$$q(t) \approx q_0 \cos\left(\frac{\gamma_q \Omega_{RF}}{2} t\right)\left(1 + \frac{b_q}{2}\cos\left(\Omega_{RF} t\right)\right) \tag{2.10}$$

with $\gamma_q = \sqrt{a_q + b_q^2/2}$, describing a *secular harmonic motion* with frequency

$$\omega_q = \frac{\gamma_q \Omega_{RF}}{2}, \tag{2.11}$$

which corresponds to the frequencies of secular motion along axis $q \in \{x, y, z\}$ in Sec. 2.5. This motion is superimposed by the rapidly oscillating, small amplitude *micromotion* at frequency Ω_{RF} [65].
The total axial potential seen by an ion at position x can be obtained using basis functions [71]. These functions $\Phi_i(x)$ represent the potential generated by the electrode pairs i at a unit voltage of 1 V, while all other electrodes are at 0 V. The total axial potential is then given by the multiplication of the basis function with the applied voltage U_i and taking the sum over all electrode pairs i:

$$\Phi(x) = \sum_i U_i \Phi_i(x). \tag{2.12}$$

To create individual trapping wells for single positively charged ions, a negative voltage is applied to the corresponding segments, while 0 V is applied to the surrounding segments. This leads to multiple individual trapping potentials along the x-axis. The axial trap frequency ω_x at a potential minimum position x_{min} with $\Phi'(x_{min}) = 0$ and $\Phi''(x_{min}) > 0$ is proportional to the second derivative of the trap potential at this position and, assuming a harmonic potential, given by

$$\omega_x = \sqrt{\frac{e}{m}\Phi''(x_{min})}. \tag{2.13}$$

2.5 Vibrational Modes of Trapped-Ion Crystals

Due to trap confinement in all three spatial directions and coulomb repulsion of ions, an *ion crystal* is formed along the weakest trap axis, which is chosen to be the x-axis of the system. The stable confinement in the y- and z-axis is provided by ponderomotive forces created by rapidly oscillating electric fields, where ideally the node with zero force aligns with the trap x-axis. Properties of the ion string, which are the ion equilibrium positions and frequencies of the common motional modes, are specified by the potential energy of a set of N singly-charged ions of mass m

$$\begin{aligned} V &= V_{Trap} + V_{Coulomb} \\ &= \frac{1}{2}m\sum_{n=1}^{N}\left(\omega_x^2 x_n^2 + \omega_y^2 y_n^2 + \omega_z^2 z_n^2\right) + \frac{e^2}{8\pi\epsilon_0}\sum_{\substack{n,m=1 \\ n\neq m}}^{N}\frac{1}{\sqrt{(\vec{r}_n - \vec{r}_m)^2}}, \end{aligned} \tag{2.14}$$

where the coordinates of the n-th ion are given by $\vec{r}_n = (x_n, y_n, z_n)^T$, the axial trap frequency ω_x and the transversal trap frequencies $\omega_{y/z}$ [72]. The equilibrium position of the Nth ion is given by the minimum of the total potential energy. Assuming a linear arrangement of the ions along the trap axis, the equilibrium positions are determined by

$$\left[\frac{\partial V}{\partial x_n}\right]_{x_n = x_n^{(0)}} = 0. \tag{2.15}$$

With the length scale l defined to be:

$$l^3 = \frac{Z^2 e^2}{4\pi\epsilon_0 m\omega_x}. \tag{2.16}$$

The dimensionless equilibrium positions $u_n = x_n^{(0)}/l$ can be determined analytically for $N \leq 3$ to be $u_1 = -(1/2)^{2/3}$, $u_2 = (1/2)^{2/3}$ for $N = 2$ and $u_1 = -(5/4)^{1/3}$, $u_2 = 0$, $u_3 = (5/4)^{1/3}$ for $N = 3$ [73]. Ion positions for larger N can be determined numerically. As the oscillations near the quantum mechanical ground state are much smaller compared to the equilibrium distances, a harmonic approximation is valid.
The ions undergo collective vibrational motion know as *secular motion*. The eigenfrequencies of the harmonic oscillator potential, presented in Table 2.1, can be evaluated by expanding Eq. 2.15 to second order in the coordinates. A detailed derivation can be found in [67] and [70]. The frequencies of the secular motion in the presented setup are typically around $2\pi \times \{1.49, 3.88, 4.64\}$ MHz for the in-phase center of mass (com) modes and around $2\pi \times \{2.57, 3.57, 4.37\}$ MHz for the out-of-phase breathing and rocking modes of the $\{x, y, z\}$-axis, respectively.

axis	mode	frequency
x	com	ω_x
x	breathing	$\sqrt{3}\omega_x$
y/z	com	$\omega_{y/z}$
y/z	rocking	$\sqrt{\omega_{y/z}^2 - \omega_x^2}$

Table 2.1: The common motional modes of a two-ion crystal, relevant for the experiments presented in this work. Shown are the in-phase center of mass (com) and the out-of-phase breathing and rocking modes.

2.6 Light-Motion Coupling

Interaction of trapped ions with light are of special importance in many respects, as the ground-state cooling of the trapped ions and laser-driven quantum gates rely on it. Here, we focus on a coherently driven transition between two internal states $|g\rangle$ and $|e\rangle$, separated by the energy $\hbar\omega_0 = \hbar(\omega_e - \omega_g)$. We assume that this transition is spectrally isolated from other transitions and that the natural lifetime of both states is much longer than any period associated with secular motion. This regime is commonly referred to as *resolved sideband regime*. The Hamiltonian describing the internal electronic level structure is then given by [65]

$$\hat{H}_e = \hbar\frac{\omega_0}{2}\hat{\sigma}_z \tag{2.17}$$

with the Pauli z-matrix $\hat{\sigma}_z$. The ion motion in the 1D harmonic oscillator can be described in the Heisenberg picture by the Hamiltonian

$$\hat{H}_m = \hbar\omega_x\left(\hat{a}^\dagger\hat{a} + 1/2\right). \tag{2.18}$$

with the creation and annihilation operators $\hat{a}^\dagger$ and $\hat{a}$. When reducing to a single ion and a single secular mode, the interaction Hamiltonian $\hat{H}_i$ describing the coupling of the internal states and a running wave light field is given by [65]

$$\begin{aligned}\hat{H}_i &= \frac{\hbar\Omega}{2}\left(|g\rangle\langle e| + |e\rangle\langle g|\right) \times \left[e^{i(kx-\omega_l t+\phi)} + e^{-i(kx-\omega_l t+\phi)}\right] \\ &= \frac{\hbar\Omega}{2}\left(\sigma_+ + \sigma_-\right) \times \left[e^{i(kx-\omega_l t+\phi)} + e^{-i(kx-\omega_l t+\phi)}\right]\end{aligned} \tag{2.19}$$

with $|e\rangle\langle g| = \sigma_+$ and $|g\rangle\langle e| = \sigma_-$. This is restricted to one dimension for simplicity, including the assumption that the wave vector k is aligned along the corresponding x-axis, where x is the position of the ion, Ω the resonant Rabi frequency, ω_l the light frequency and ϕ the phase of the light field.
The complete Hamiltonian of the system can then be written as

$$\hat{H} = \hat{H}_m + \hat{H}_e + \hat{H}_i \tag{2.20}$$

We transform into the interaction picture using the free Hamiltonian $\hat{H}_0 = \hat{H}_m + \hat{H}_e$. After application of the *rotating-wave approximation*, i.e. neglecting the rapidly oscillating terms, as well as an expansion up to first order with respect to the *Lamb-Dicke* parameter

$$\eta = k\sqrt{\hbar/(2m\omega_x)}, \tag{2.21}$$

the transformed interaction Hamiltonian in the Lamb-Dicke regime can be written as

$$\hat{H}_i^{(LD)} = \frac{\hbar}{2}\Omega_0\sigma_+ \left[1 + i\eta(\hat{a}e^{-i\omega_x t} + \hat{a}^\dagger e^{i\omega_x t})\right] e^{i(\phi-\delta t)} + h.c. \tag{2.22}$$

with the bare Rabi frequency Ω_0 and $\delta = \omega_l - \omega_0$ being the detuning of the effective light frequency with respect to the atomic transition. The assumption of the Lamb-Dicke regime holds true as long as $\eta\sqrt{\langle(a+a^\dagger)^2\rangle} \ll 1$ [65]. The interaction Hamiltonian in Eq. 2.22 can describe the following important three transitions between the atomic levels, taking into account the quantization of the motion:

- Carrier resonance (car), $\delta = 0$:

$$\hat{H}_{car} = \frac{\hbar}{2}\Omega_0(\sigma_+ e^{i\phi} + \sigma_- e^{-i\phi}) \tag{2.23}$$

  Driving the transition $|n\rangle|g\rangle \leftrightarrow |n\rangle|e\rangle$ not affecting the motional state, with Rabi frequency Ω_0, used for spin flips.

- Red sideband (rsb), $\delta = -\omega_x$:

$$\hat{H}_{rsb} = \frac{\hbar}{2}\Omega_0\eta(\hat{a}\sigma_+ e^{i\phi} + \hat{a}^\dagger\sigma_- e^{-i\phi}) \tag{2.24}$$

  Driving the transition $|n\rangle|g\rangle \leftrightarrow |n-1\rangle|e\rangle$ with Rabi frequency $\Omega_{n,n-1} \approx \Omega_0\eta\sqrt{n}$, used for resolved sideband cooling, spectroscopic measurements and probing of motional excitation. A $\pi/2$-pulse on this transition entangles the motion and the spin as $\frac{1}{\sqrt{2}}(|n\rangle|g\rangle + |n-1\rangle|e\rangle)$.

- Blue sideband (bsb), $\delta = +\omega_x$:

$$\hat{H}_{bsb} = \frac{\hbar}{2}\Omega_0\eta(\hat{a}^\dagger\sigma_+ e^{i\phi} + \hat{a}\sigma_- e^{-i\phi}) \tag{2.25}$$

  Driving the transition $|n\rangle|g\rangle \leftrightarrow |n+1\rangle|e\rangle$ with Rabi frequency $\Omega_{n,n+1} \approx \Omega_0\eta\sqrt{n+1}$. This transition can be used for spectroscopic measurements and probing of motional excitation. A $\pi/2$-pulse on this transition entangles the motion and the spin as $\frac{1}{\sqrt{2}}(|n\rangle|g\rangle + |n+1\rangle|e\rangle)$.

The approximations on the Rabi frequencies do only hold in the Lamb-Dicke regime. Beyond this regime, the effective Rabi frequency $\Omega_{n,n\pm N}$ for a transition of a state with n phonons to a state with $n \pm N$ phonons can be expressed by

$$\Omega_{n,n\pm N} = M_{n,n\pm N}\Omega, \tag{2.26}$$

using the matrix elements [74]

$$M_{n,n\pm N} = \langle n \pm N| \, e^{ik\hat{x}} \, |n\rangle = e^{-\eta^2/2}(i\eta)^N \mathcal{L}_n^N(\eta^2) \left(\frac{n!}{(n\pm N)!}\right)^{\pm 1/2} \tag{2.27}$$

with the Laguerre polynomials $\mathcal{L}_n^N$. Off-resonant excitation can be neglected, if η and Ω are sufficiently small and if the detuning is close to a motional sideband. Starting from the ground state, Rabi oscillations will be driven on the two-level system of the ground and excited state. The time-dependent population of the excited state can be obtained to be

$$p_e(t) = \sum_n |c_{e,n(t)}|^2 = \sum_n p_n \sin^2(\Omega_{n,n\pm N} t/2), \tag{2.28}$$

with the phonon probability distribution $p_n^{(th)}$. After Doppler cooling of ions, $p_n^{(th)}$ is expected to follow a Maxwell-Boltzmann distribution

$$p_n^{(th)} = \frac{\bar{n}^n}{(\bar{n}+1)^{n+1}} \text{ with } \bar{n} = \frac{k_B T}{\hbar\omega_\nu}. \tag{2.29}$$

2.7 Stimulated Raman Transitions

Driving the transition between the qubit state $|0\rangle$ and $|1\rangle$ is realized using two-photon stimulated Raman transitions. Two light fields around 397 nm are used. The effective wave vector of this combined light field is given by $\boldsymbol{k} \equiv \boldsymbol{k_1} - \boldsymbol{k_2}$ and the effective light frequency by $\omega_l \equiv \omega_1 - \omega_2 = \omega_0 + \delta$, where δ is the small detuning to the qubit resonance ω_0 (see Fig. 2.3). The effective coupling strength of the states $|0\rangle$ and $|1\rangle$ is given by [65]

$$(\hbar/2)\Omega_{eff} = -\hbar\frac{|\Omega_1\Omega_2|}{\Delta} e^{-i\Delta\phi}, \tag{2.30}$$

with the dipole Matrix elements $\hbar\Omega_1$ and $\hbar\Omega_2$, pertaining to resonant driving of the corresponding transition between $S_{1/2} \leftrightarrow P_{1/2}$ according to Fig. 2.3. $\Delta\phi$ represents the phase difference between the two light fields.
Stimulated Raman transitions have two main advantages. First, only the relative frequency and phase between both beams is of relevance for the stimulated Raman transition, while the absolute frequency and phase of the driving laser is unimportant. A higher detuning to the $P_{1/2}$ level is favorable to reduce scattering, but minor instabilities in the driving laser frequency are not a relevant error source for the quantum gate fidelities. In the current setup, a detuning of around $\Delta \approx 2\pi \times$ 1 THz

is used, as a trade-off between required laser power and low scattering rate, reducing the error source of unintentional scattering on $P_{1/2}$. Under the assumption of equal intensity I of the two light fields at the position of the ion, the Rabi frequency scales with $\Omega \propto I/\Delta$, while the decoherence due to spontaneous emission scales as $R \propto I/\Delta^2$ [67, 75]. Therefore, if intensity and detuning are both increased, the same Rabi frequency can be maintained, while the spontaneous emission rate is reduced by $\propto 1/\Delta$. While one laser beam is used at a static frequency ω_2, the frequency of the second laser beam ω_1 and therefore the detuning δ can be adjusted using acousto-optic modulators (AOMs), driven by phase-locked rf-sources. The frequency adjustment during measurement sequences allows to change the effective light field and to drive the carrier or motional sideband transitions according to Eq. 2.23, 2.24 and 2.25, selectively.

Second, the coupling to the motional modes of the ion crystal can be controlled by choosing the alignment of the beams with respect to each other and the trap axis and therefore varying the effective wavevector $\boldsymbol{k}$. Copropagating beams can be used to tune the Lamb-Dicke factor close to zero $\eta = (|\boldsymbol{k_1}| - |\boldsymbol{k_2}|)x_0 \approx 0$ and thus render the transition motionally insensitive [65].

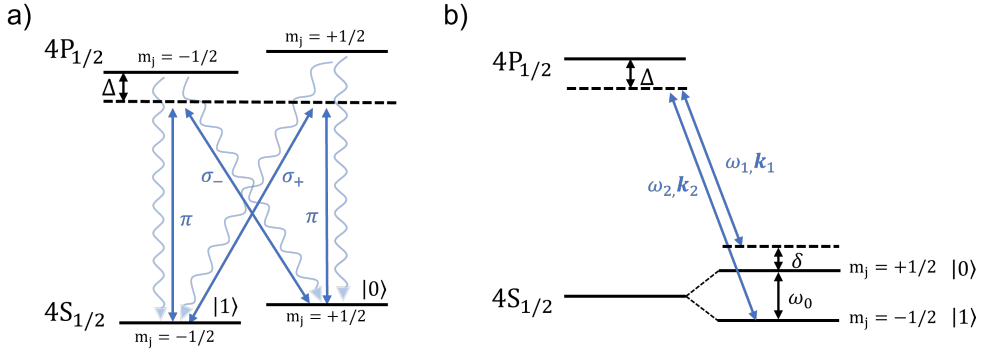

Figure 2.3: **a)** Relevant levels and polarization for the two-photon process of stimulated Raman transition. The wavy arrows show possible paths of decay into both qubit states after scattering on the $P_{1/2}$ level. **b)** Scheme for stimulated Raman transitions between the $S_{1/2}$ ground states of the $^{40}Ca^{+}$, via two laser fields with a detuning of Δ to the $P_{1/2}$ level. The splitting of the Zeemann sublevels ω_0 is typically around $2\pi \times 10$ MHz. A small detuning δ is added to drive the red and blue sideband transitions, according to Sec. 2.8.

2.8 Resolved Sideband Cooling on the Raman Transition

Since the two-qubit gate, presented in Sec. 2.9.2, requires the ion qubits to be cooled close to the motional ground state, another cooling step has to be performed following Doppler cooling. *Resolved sideband cooling* makes use of driving the red sideband (rsb) transition to reduce the phonon number, upon population transfer from ground to excited state [76]. Resolved sideband cooling can be realized on either the $S_{1/2} \leftrightarrow D_{5/2}$ transition (729 nm) or using the stimulated Raman transitions with two beams at 397 nm [67]. The stimulated Raman transitions feature the advantage of different available beam combinations in the setup, offering the possibility to cool all motional modes (see Sec. 3.3 for details on available beam pairs). Additionaly, a higher Lamb-Dicke parameter and the two-photon process do render the cooling faster and more efficient. For $^{40}Ca^{+}$, stimulated Raman transitions between the qubit states $|0\rangle \equiv |S_{1/2}, m_J = +1/2\rangle$ and $|1\rangle \equiv |S_{1/2}, m_J = -1/2\rangle$ are used on the red sidebands of the relevant motional modes, approximately realizing the transition

$$|0\rangle |n\rangle \rightarrow |1\rangle |n-1\rangle , \tag{2.31}$$

and reducing the phonon number of the corresponding motional mode. The procedure starts with a thermal phonon distribution (see Eq. 2.29). Due to this distribution, only imperfect π-pulses can be applied. The pulse time is varied during the cooling procedure, in order to maintain the cooling efficiency, as the phonon number decreases throughout the process. Pulse schemes for efficient cooling can be obtained using optimization algorithms [77]. As $|1\rangle$ does not naturally decay back to $|0\rangle$, one has to reinitialize back to $|0\rangle$ after every rsb pulse, to establish an efficient cooling cycle. The σ_+-polarized beam near 397 nm with a pulse duration of 1 µs is used to reset the ion state according to

$$|1\rangle |n-1\rangle \rightarrow |0\rangle |n-1\rangle . \tag{2.32}$$

This cooling procedure is well established in our setup and has been discussed in more detail in [67, 70, 75].

2.9 Quantum Gate Operations

The $^{40}Ca^{+}$ spin qubit is encoded in the Zeeman sublevels of the $S_{1/2}$ ground state of the valence electron. Typically, a magnetic field of around 3.7 G is used to provide a qubit splitting of about $\omega_0 \approx 2\pi \times 10$ MHz. Laser-driven qubit rotations (Sec. 2.9.1) can be carried out using *stimulated Raman transitions* [76, 78], described in 2.7. Two-qubit gates can be performed using the *light-shift gate* [79], described in Sec. 2.9.2.

2.9.1 Single-Qubit Gate

Single-qubit rotations are realized via the previously described technique of stimulated Raman transition. Two copropagating laser beams are used, where polarization and geometry are chosen, such that the two-photon process can be carried out as pictured in Fig. 2.3. This beam geometry holds the advantage of a vanishing k-vector of the effective drive field and therefore no coupling to the motional modes of the single- or two-ion crystal. While this coupling is of crucial importance for the resolved sideband cooling and the realization of the geometric phase gate, it is representing an error source for the single-qubit gates, since the Rabi-frequency depends on the number of motional quanta. By eliminating the motional coupling and using a Raman detuning of around $2\pi \times 1$ THz, high quality single-qubit gates can be realized with a typical error rate of around 10^{-4}.
The resonant driving of the single-qubit rotation can be described by the unitary operator

$$\hat{R}(\theta, \phi) = \exp\left[-i\frac{\theta}{2}\left(\cos(\phi)\hat{\sigma}_x + \sin(\phi)\hat{\sigma}_y\right)\right] \tag{2.33}$$

with the Pauli matrices in the logical basis $|0\rangle$, $|1\rangle$ to be

$$\hat{\sigma}_x \equiv \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}, \hat{\sigma}_y \equiv \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix}, \hat{\sigma}_z \equiv \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}. \tag{2.34}$$

Rotation angle ϕ can be controlled via the phase of the applied light field, while the pulse area $\theta = \Omega t$ directly depends on the pulse exposure time t and the Rabi frequency Ω. Experimentally, the laser power and therefore the Rabi frequency is stabilized to a constant value, while the exposure time can be varied in order to realize the desired rotation angle. A single-qubit rotation can be either applied to a single qubit, but also to two qubits confined in the same potential well. Details about the laser beam arrangement in our setup can be found in Sec. 3.3.

2.9.2 Two-Qubit Gate

A set of universal quantum gates [80] requires not only single-qubit gates, but also a gate acting on two qubits to create entanglement [81]. Different possible implementations of this entangling gate for trapped-ion qubits have been shown, such as the historically important *Cirac-Zoller gate* [11], the *Mølmer–Sørensen gate* [82] and the *light-shift gate* [79], which works even for small residual phonon numbers. A detailed explanation of these gates, their relation and differences can be found in [60]. Our architecture uses the laser-driven *light-shift gate*, employing the radial com mode. The gate is realized by two orthogonally propagating laser beams, oriented such that the effective k-vector is aligned perpendicular to the trap axis. The beams, arranged in lin-$\perp$-lin polarization geometry, build a moving standing wave with a spin-dependent optical dipole force acting on the two ions (see Fig. 2.4). The effective frequency, which is the frequency difference of the two laser fields $\omega \equiv \omega_1 - \omega_2$, is tuned close

to one of the collective motional modes, which is chosen to be the transverse in-phase mode in z direction around $2\pi \times 4.64$ MHz. A small detuning of $\delta \approx 2\pi \times 20$ kHz leads to transient oscillatory excitation of the mode, returning to the starting point in phase space after the duration $T = 2\pi/\delta \approx 50$ µs. Due to this motion, a phase Φ proportional to the enclosed phase space area will be acquired, which can be tuned via the power of the driving light field. A phase of $\Phi = \pi/2$ pertains to a maximal entanglement. The unitary describing this phase accumulation only on the even parity qubit states, while not affecting the odd qubit states, is

$$ZZ_{ij}(\Phi) = e^{\frac{i}{2}\Phi(Z_i \otimes Z_j + \mathbb{1}\otimes\mathbb{1})}, \tag{2.35}$$

where Z_i and Z_j describe the $\hat{\sigma}_z$ Pauli matrix on the corresponding qubit i and j. The term $\mathbb{1} \otimes \mathbb{1}$ could be neglected due to the fact that it only adds a global phase. The phase Φ is accumulated within two separate loops, with a rephasing pulse $R(\pi, -\pi/2)$ in between (see Fig. B.1). This double loop gate reduces the effect of several error sources, such as uncompensated AC Stark shifts, see also D. Pijn [83]. A detailed discussion of the gate error sources was carried out by C. Ballance [84]. The total gate unitary transforms to

$$G_{ij} \quad = \quad ZZ_{ij}(\pi/4)R(\pi, -\pi/2)ZZ_{ij}(\pi/4). \tag{2.36}$$

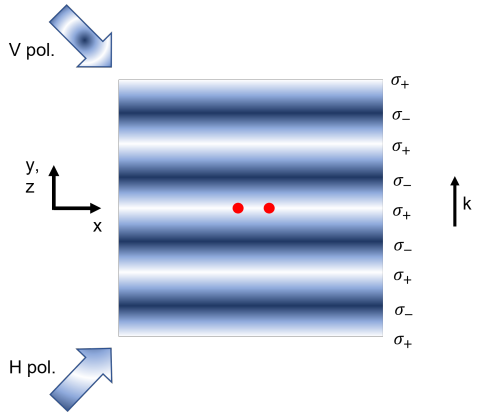

Figure 2.4: Sketch of the light field used to generate two-qubit entanglement. Two laser beams are used, one vertically and one horizontally polarized. The effective light field forms a moving standing wave with a repeating pattern of σ_+, lin., σ_-, lin polarization. The oscillating electric field causes an AC Stark shift, which leads to a spin-dependent optical dipole force on the ions (red dots).

Chapter 3

Experimental Setup

This chapter gives an overview of the setup used for optimizations and experiments presented in this thesis, including register configuration improvements and fault-tolerant parity measurements. The experimental setup was primarily developed and described in detail in previous works [70, 75, 85]. Sec. 3.1 provides technical details of the trap setup, Sec. 3.2 describes the quantizing magnetic field and the shielding from surrounding magnetic field fluctuations. The laser system with the beam alignment is shown in Sec. 3.3. Sec. 3.4 covers the experimental control setup, including detailed description of the hardware and software components, which were significantly expanded during this work.

3.1 Trap Setup

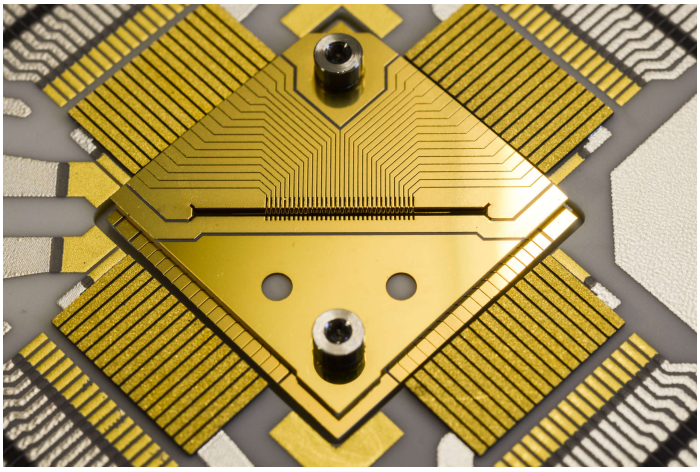

Figure 3.1: Picture of the micro-structured rf-trap used for all experiments presented (Image taken from [70]).

The centerpiece of the apparatus is the micro-structured segmented ion trap located in the vacuum chamber, set up by Henning Kaufmann as part of his PhD thesis [70].

It consists of two metalized device layers, separated by a non-conducting spacer layer. The device layers are made up of laser machined alumina wafers with a thickness of 127 µm and were structured with a precision of a few µm using a femtosecond laser. After an extensive cleaning procedure, the chip layers were gold coated using a vapor deposition technique with a titanium adhesion layer of 50 nm followed by a gold layer of 500 nm, which creates the conducting surface for the electrodes. In order to create the individually addressable electrodes, the gold coating is removed using laser machining. After successful separation of the electrode connections, the thickness of the gold layer was increased to about 8 µm using electroplating. The device layers are mounted inverted w.r.t. the DC and RF electrodes, forming a Paul trap as described in Sec. 2.4. A picture of one side of the assembled trap is shown in Fig. 3.1. The register processing region consists of 32 DC electrode pairs, each able to form a linear Paul trap. All laser operations are performed at the *laser interaction zone* (LIZ), which is chosen to be segment 19. It shall be noted that the segment used as LIZ can in principle be easily changed by laser beam and detection optics alignment to any other segment. The dimensions of the trap feature a slit height between DC and RF segments of a single layer of $h = 400$ µm and a distance d of the two conducting layers of $d = 254$ µm. The segment center to center distance is given by 200 µm, consisting of a segment width of $w = 150$ µm and a gap of 50 µm. After increasing the gold surface thickness in the electroplating procedure, the actual segment parameters are measured to be around $w = 166$ µm and $h = 384$ µm [70]. The dimensions were chosen to provide sufficient optical access, while also resulting in high electric fields for tight confinement, as well as high control over the axial potential due to small enough segment width. This allows for fast shuttling operations with low motional excitation. The trap is operated at room temperature within a vacuum chamber at ultra high vacuum (UHV), providing low background collision rates. The trap is typically operated with DC voltages of around -6 V at segments where ions are stored and 0 V at other segments, resulting in an axial trap frequency of $\omega_x \approx 2\pi \times 1.48$ MHz. The typical RF drive frequency of 33.4 MHz and amplitude of around 400 Vpp results in radial secular trap frequencies of $\omega_y \approx 2\pi \times 3.8$ MHz and $\omega_z \approx 2\pi \times 4.6$ MHz [51].

3.2 Quantizing Magnetic Field and Shielding

The quantizing magnetic field, as well as the shielding of external magnetic fields, are of high importance to operate the magnetic field sensitive qubit. It has been shown that using permanent magnets instead of coils to create the quantizing magnetic field, leads to an increase of the qubit spin-echo coherence time from 45(3) ms to 2.2(1) s [61]. The vacuum chamber is protected from ambient magnetic field fluctuations by a μ-metal enclosure with outer dimensions of 510 mm x 625 mm x 625 mm, consisting of two layers of μ-metal with 2 mm thickness, separated by a 6 mm aluminium layer, designed by Claudia Warschburger and further described in [61]. This μ-metal enclosure shields mostly higher frequency noise, but does also feature a shielding factor of about 25 for magnetic field drifts on the time-scale of minutes. The quantizing magnetic field features a magnetic field strength of around 3.7 G at the ion position, leading to a Zeeman splitting of around $2\pi \times 10.5$ MHz. The magnetic field outside of the shielding

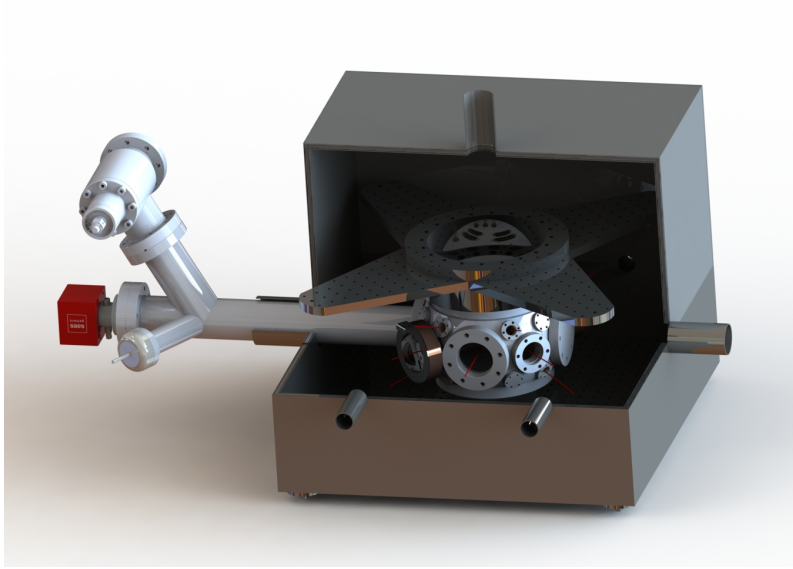

Figure 3.2: Setup of the vacuum chamber and μ-metal shielding. Picture rendered using the existing CAD drawing of the original design.

is actively stabilized using 3D fluxgate sensors and large magnetic field coils, further described by D. Pijn [83]. Ambient magnetic field fluctuations at a timescale of seconds can be compensated for using a varying current applied to the coils [86]. The ions inside the shielding and trap are still sufficiently well protected from the magnetic field noise caused by coils, while the permanent magnets inside the shielding provide a static quantizing magnetic field.

3.3 Laser Systems

The laser setup is used to drive several operations on the $^{40}Ca^{+}$ ions, which are

- Photoionization
- Doppler cooling
- Resolved sideband cooling
- Initialization of the internal state by optical pumping
- Single- and two-qubit gates
- Spin-selective readout

Lasers with their corresponding characteristics are explained in more detail by Thomas Ruster in [75] and Henning Kaufmann in [70]. Detailed sketches of the laser focusing

and alignment optics inside the magnetic shielding enclosure can also be found in [75]. Fig. 3.3 shows the laser setup w.r.t. the trap within the vacuum chamber. While Raman transitions, photoionization, Doppler cooling, sigma pumping, quenching and repumping are provided by beams directed at an angle of 45° to the x-axis of the trap, only the 729 beam used for spin-selective readout is aligned at an angle of 90° to the x-axis. Since the ion crystal is aligned along this x-axis, the 729 nm does not couple to the axial mode of motion. This alignment serves to maintain good spin-selective readout, also in the presence of axial motion arising from shuttling operations.

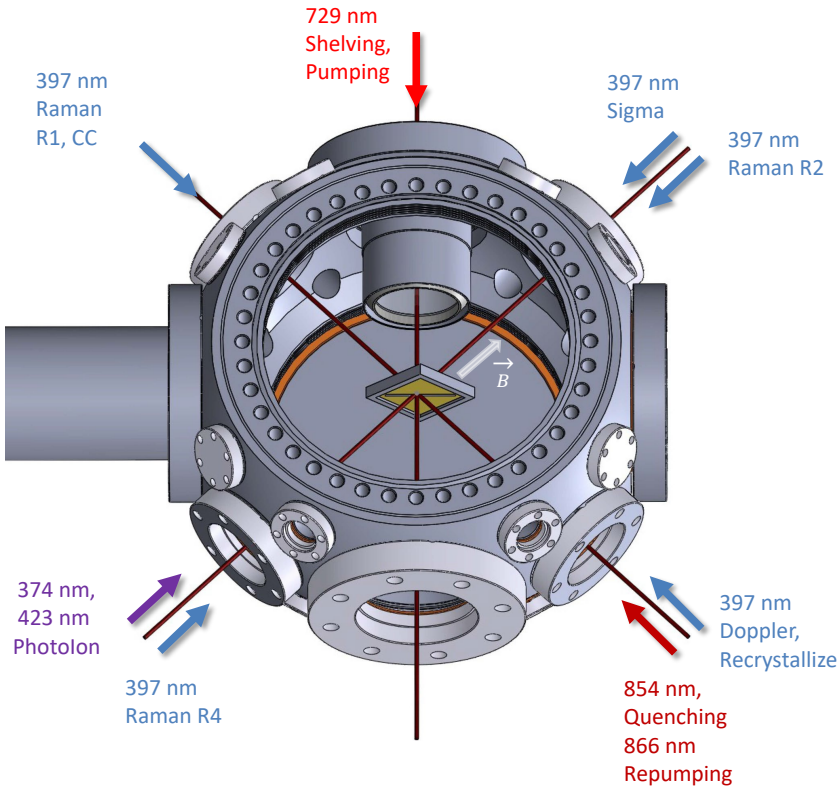

Figure 3.3: Sketch of the laser beam delivery with respect to the ion trap. Based on the original CAD of the vacuum chamber.

Photoionization - 423 nm and 374 nm

An effusive calcium oven provides a thermal beam of neutral ^{40}Ca atoms, which are ionized using a two-photon process. Both required laser beams are coupled into the same single-mode fiber, such that they can be commonly aligned on the trapping region. When ion retrapping is required, both laser beams at 423 nm and 374 nm

are turned on, while photon count rate and EMCCD camera picture are evaluated to determine the number of ions within the LIZ. The ionization beams are shut off as soon as the correct amount of ions is detected. In the case of trapping multiple ion crystals, the potential wells are moved along the trap axis, while the ion crystals are sequentially loaded and cooled until all ions are at their desired locations.

Doppler Cooling and Fluorescence Detection - 397 nm

Doppler cooling driving the cycling transition $S_{1/2} \leftrightarrow P_{1/2}$ is realized using a single beam aligned at 45° w.r.t. the trap x-, y- and z-axis. Therefore, all motional modes can be cooled close to the Doppler cooling limit using only a single beam. Fluorescence detection is realized using the same beam at higher optical power and detuned closer to resonance with an exposure time of 800 µs. When trapping ions, a second beam of the 397 nm laser called *recrystallize* is used, which is detuned -300 MHz from the Doppler cooling wavelength. This beam is especially suitable to cool down hot ions after trapping.

Sigma Pumping - 397 nm

Optical pumping is performed using a σ_+-polarized beam at 397 nm, referred to as sigma beam. It is aligned parallel to the magnetic field, counterpropagating to the magnetic field direction and driving exclusively the $|S_{1/2}, m_j = -1/2\rangle \leftrightarrow |P_{1/2}, m_j = +1/2\rangle$ transition, leading to a net population transfer into the $|S_{1/2}, m_j = +1/2\rangle$.

Repumping - 866 nm

When driving the cycling transition, there is a 6% probability per scattered photon to decay into the metastable $D_{3/2}$ state with a lifetime of around 1.18 s. Therefore, during every exposure of near resonant 397 nm light for Doppler cooling or detection, a repumping beam with a wavelength of 866 nm is turned on, which depletes the $D_{3/2}$ state via the $P_{1/2}$ state.

Initialization and Shelving - 729 nm

A laser beam at 729 nm is used to improve the state initialization via optical pumping, as well as for shelving. Shelving is used to render one spin state inaccessible for the cycling transition and therefore allows for distinguishing of qubit states upon detection, as described in Sec. 2.3.

Quenching - 854 nm

The spin-selective readout is based on transferring the $|S_{1/2}, m_j = +1/2\rangle$ selectively into the metastable $D_{5/2}$ state with a lifetime of around 1.17 s. Since this time is long compared to the typical detection cycling time of 800 µs, this so-called quench process is used to deplete the $D_{5/2}$ state again after detection and maintain a fast single-shot measurement cycle.

Stimulated Raman Transitions - 397 nm

Stimulated Raman transitions rely on the combination of two light fields using the effective wave vector and frequency of the combined field, to drive carrier and motional sideband transitions upon the qubit levels (see Sec. 2.7). We employ multiple combinations of laser beam orientations (see Fig. 3.3), in order to provide different coupling strength to the motional modes:

R1 & CC: The combination of R1 and CC is delivered through a common optical fiber. This maintains a perfect common alignment on the ion and therefore the effective wave vector of these copropagating beams vanishes. While this alignment could be reached using two fibers, the single fiber also serves to keep the relative optical phase of the two beams constant. Raman transitions between the qubit states $|0\rangle$ and $|1\rangle$ can be driven without coupling to any secular mode. This combination is optimal to drive single-qubit rotations as described in Sec. 2.9.1.

R1 & R2: R1 and R2 couple to the axial mode of motion and are therefore utilized for axial sideband cooling as well as probing of the axial motional excitation. This configuration is typically not used within actual quantum circuits, but it is of crucial importance for trap characterization and to improve shuttling operations along this axis.

R1 & R4: R1 and R4 couple to motional modes in both transverse directions. This combination is utilized for sideband cooling on transverse motional modes, as well as for characterization measurements.

R4 & CC: These beams, while directed in the same manner as R1 and R4, feature a lin-$\perp$-lin polarization. Due to the absence of a π-polarization component, the qubit is not driven, but rather spin-dependent optical dipole forces are generated. This combination of R4 & CC is used to create a spin-dependent light force, which enables driving the entangling gate operation (see. Sec. 2.9.2).

3.4 Experimental Control System

The experimental control system comprises of a large number of hardware and software components. Fast control of the trap, modification of laser beam frequencies and amplitudes, data readout and processing is required to operate the trapped-ion quantum processor. The following description gives an overview of the current status of the system, where most of the components frequently undergo revision and improvement cycles. Especially the software control has undergo a major rework in the last years to facilitate maintenance and development in this advanced, rapidly developing system and also to harness new hardware features.

3.4.1 Hardware

The most important hardware components to control the shuttling-based trapped-ion quantum processor are outlined in Fig. 3.4. Composition of the measurement sequences and major control and evaluation is carried out on the control PC, which supplies a variety of devices with data.

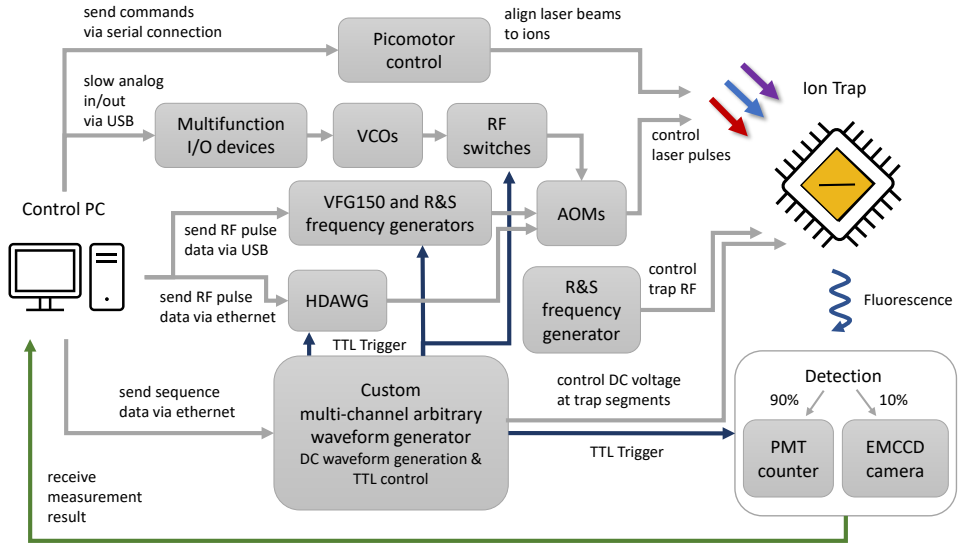

Figure 3.4: Sketch of the hardware setup controlling the shuttling-based trapped-ion quantum processing node.

The devices can be separated into two categories, realtime devices and near-realtime devices. The real-time devices perform time critical operations during the execution of a quantum algorithm. A single execution of the algorithm is referred to as *shot*. The field-programmable gate array (FPGA) inside the custom multi-channel arbitrary waveform generator (mAWG) is controlling the sequence execution. It can output transistor–transistor logic (TTL) signals to trigger devices with a time resolution of 20 ns. The near-realtime devices do not require a change in settings or output signals during the execution of a measurement sequence. All settings are made by the control PC in a comparably slow timescale of a few milliseconds. This communication is carried out via Ethernet or USB interfaces, depending on the hardware.

Realtime devices:

- **Custom multi-channel arbitrary waveform generator (mAWG)** A critical part of the hardware setup is the mAWG based on a SoC including a FPGA for real-time execution of the measurement sequence, in detail described in [51, 85]. It provides 80 channels of DC voltages that are generated via digital analog

converters (DACs) and up to 25 TTL channels synchronized to the time-dependent analog voltage ramps. The TTL channels can be used to trigger RF waveform generators, the photomultiplier tube (PMT) counter and EMCCD camera readout, as well as several (RF) switches and mechanical shutters.

- **VFG150** The acousto-optical modulators (AOMs) of laser beams that need in-sequence frequency and amplitude tuning, are supplied with RF waveforms by a VFG150[1]. The VFG150 belongs to the realtime devices, since it receives TTL trigger pulses from the mAWG. The control PC sends a list of RF pulses to the VFG150 prior to a sequence execution. Amplitude, frequency and phase of the output signal can be programmed in steps of 5 ns. The VFG150 is set to triggered mode, where it is acting like a FIFO queue for the pulses. The timing for the execution of the next pulse is based on the TTL trigger send by the mAWG.

- **HDAWG** The HDAWG arbitrary waveform generator[2] can be used to supply the AOMs of laser beams that need in-sequence frequency and amplitude tuning, similar to the VFG150. In contrast to the only single output of the VFG150, the HDAWG has four individual outputs. Due to the higher number of outputs, it enables the possibility to use shaped pulses on both AOMs involved to control the two laser beams for single- and two-qubit rotations. During the data acquisition for this thesis, the integration of the HDAWG was still under development, which is why it is not used for the measurements presented here.

- **RF switches** A network of RF switches[3] is used to control the path of the RF signal from the RF sources to the target AOM in the laser beam setup. This is controlled using multiple TTLs outputs of the mAWG.

- **PMT Counter & EMCCD Camera** PMT counter[4] and camera hardware[5] collect photon counts during the detection time and communicate measurement results back to the control PC, where a threshold-based bright/dark discrimination is carried out. The data acquisition is timed using a TTL trigger of the mAWG.

Near-Realtime devices:

- **Picomotor control** The laser beams can be precisely aligned using picomotor piezo screws[6] on mirror mounts. The beam pointing can be optimized, to e.g. illuminate two qubits equally, using the Picomotor screw remote controller. This optimization is usually only required after several days or weeks of operation or after larger temperature fluctuations in the lab, causing mechanical drifts of the setup.

[1] Versatile Frequency Generator VFG150
[2] Zurich Instruments HDAWG multi-channel Arbitrary Waveform Generator
[3] Mini-Circuits ZYSW-2-50DR
[4] Photon counting head H10682-210, Hamamatsu Photonics K.K.
[5] EMCCD camera, Andor iXon, Model No. DV860DCS-UVB
[6] Picomotor^{TM} Piezo Linear Actuators

- **Multifunction I/O devices** The Doppler, recrystallize, sigma, repump and quench laser beam amplitude and frequency are tuned using multifunction input/output (I/O) devices[7]. They supply voltages to a setup based on tunable attenuators for the amplitude and voltage controlled oscillators (VCOs)[8] for frequency control of e.g. Doppler and sigma AOMs. Since these laser beam properties are not changed in-sequence and there is no need for special pulse shaping, the communication can be realized via comparably slow USB communication to the I/O devices. The input channels of the I/O devices are also used to read in voltages given by e.g. the Greisinger temperature sensors attached to the permanent magnets.

- **VCOs** Voltage controlled oscillators are currently in use to supply RF-signals to acousto-optical modulators of the 854, 866 and 397 cooling and sigma pumping laser beams. The frequency can be controlled via a voltage between 0 and 5 V. Tunable attenuators are used to control the amplitude of the signal via a voltage. Two control voltages are provided by NI USB multifunction I/O devices to each VCO and attenuator combination.

- **AOMs** Acousto-optic modulators (AOMs)[9] are used to control the amplitude and frequency of laser beams. RF waveforms, which in-turn control all optical AOM outputs are turned on and off by the custom multi-channel arbitrary waveform generator using RF switches.

- **R&S frequency generators** The R&S frequency generators[10] are used to supply a static RF signal. One is used for the trap RF to the helical resonator at fixed 33.48 MHz and -16 dBm. Another R&S signal generator is used at fixed 110 MHz to supply the AOM controlling the second Raman laser beam.

- **Keysight power supplies** Keysight power supplies[11] can be digitally controlled and are used to set the current applied to the Ca atom oven, a reference voltage for the Raman laser power stabilization and current of the active magnetic field stabilization.

- **Laser controller** Lasers are controlled using Toptica laser control modules[12]. Some of them using the SYS DC 110 supply modules and newer lasers using the DLC Pro with remote interface. These can be used to adjust the laser properties and also remotely relock lasers.

[7] NI^{TM} USB 6000, USB 6001, USB 6008
[8] Mini-Circuits ZOS-150
[9] Gooch & Housego, e.g. I-M110-3C10BB-3-GH27, I-M095-3C10BB-3-GH27
[10] R&S SML 01, ROHDE & SCHWARZ GmbH & Co. KG
[11] Keysight E3633A DC power supply
[12] TOPTICA Photonics, DLC Pro and SYS DC 110

3.4.2 Software

Not only the hardware, but also dedicated software modules are needed to control the system. Fig. 3.5 shows the outer basic structure of the software before and after the major rework.[13] The control software is fully custom made and written in C++, currently using VisualStudio 2019 and its corresponding compiler. Before the rework, the software stack included one main application, the Master Control Program (MCP), shared over several experiments, which loads only a single lab and setup specific dynamic link library typically called ScriptDll. This included code to build up the real-time analog and digital sequence to be communicated to the System-on-Chip (SoC) and used on the FPGA, as well as custom interfaces to external hardware, measurement scheduling, data handling and much more. Source code between the labs was shared manually, making maintenance and bug fixing over all existing copies extremely difficult. Therefore, it was preferable to generalize parts of the ScriptDll and outsource it to external libraries where the same version can be shared internally over all experimental setups. The packages are hosted on the gitlab server provided by the University of Mainz[14], which allows dedicated version control, issue tracking and user permission management. Of special importance is also the ability to use this tool to work jointly with multiple people on the rapidly growing software stack of the quantum computing project.

3.4.2.1 Details on Packages and Modules

This section summarizes the currently available software packages and modules, including a short description of the purpose of each module. A more detailed description for each module can be found in the Appendix C.

MCP - Master Control Program

The MCP project is divided into three major packages. It consists of MCPLibs to maintain and handle included third-party libraries, MCPHardware, with its hardware related utility libraries and MCPCore, which includes a graphical user interface, loading of dynamic link libraries during runtime and the actual executable.

MCPLibs (third-party libraries) The MCPLibs package allows easy access to external libraries in the compatible version, without a separate install process by the user.

- **Boost** Peer-reviewed collection of c++ libraries.[15] It includes multiple high quality, well maintained libraries for e.g. file read/write access, regular expressions and multithreading.

[13] The software rework has been realized as a shared project with Alexander Stahl, who focused on the MCP rework, resulting in an experiment specific modularized code in the ScriptDll.
[14] https://gitlab.rlp.net/quantumqiv
[15] https://www.boost.org/

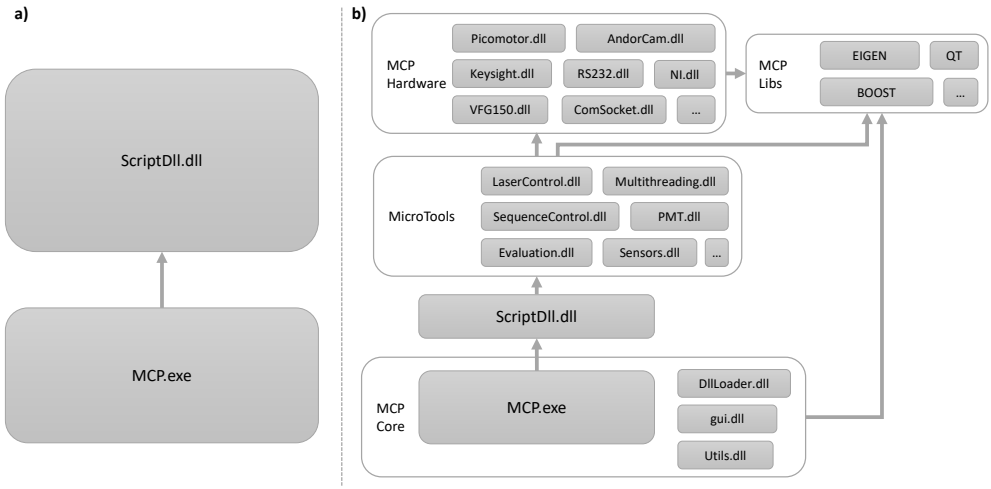

Figure 3.5: Sketch of the software controlling the shuttling-based trapped-ion quantum processor and processing the measurement data. The sketch does not attempt to be a full dependency graph including all libraries, but rather gives an overview of the most important structure. **a)** shows the previous status before the software structure rework, where the main application MCP.exe dynamically loads just a single library typically named ScriptDll.dll, which includes various control and data processing code. **b)** shows the new modularized structure, made possible by the work of Alexander Stahl, who separated MCP into MCPCore, MCPHardware and MCPLibs and enabled the DllLoader to load more than a single dll. This allowed splitting the experimental setup specific ScriptDll into multiple generalized sublibraries now located in MicroTools and MCPHardware, which can easily be maintained and shared by multiple experimental setups.

- **QT** Framework to create the graphical user interface of the master control program.[16]
- **MCPLibs::Qwt** The Qwt - Qt widgets for technical applications - library includes utility classes and gui components based on qt.[17]
- **MCPLibs::NI** NI-DAQmx driver to communicate with National Instruments hardware such as the multifunction I/O devices.
- **MCPLibs::fftw** FFTW, a well established library to compute the discrete fourier transform.[18]
- **MCPLibs::OpenCV2** OpenCV is a real-time computer vision library.[19]
- **MCPLibs::eigen** Eigen is a header-only c++ library for linear algebra.[20]
- **MCPLibs::qpp** Quantum++ is a header-only quantum computing library based on eigen 3 [87].[21]
- **MCPLibs::OPC** OPCClientToolkit to communicate to a server using Open Platform Communication (OPC).[22]
- **MCPLibs::OpenSSL** A toolkit for Transport Layer Security (TLS) and Secure Sockets Layer (SSL) protocol.[23]
- **MCPLibs::JSON** JSON - JavaScript Object Notation - is a human-readable data format.[24]
- **MCPLibs::JSONRPC** JSON-RPC is a remote procedure call (RPC) protocol, which is encoded in JSON.[25]

MCPHardware The MCPHardware package includes multiple interfaces for convenient hardware access. Since most hardware manufacturer provide an API for c++, the modules mostly consist of wrapper classes to provide the functionality needed within our control software structure.

- **MCP::AllowIO** AllowIO library is used within MCP::DAC8814 and MCP::DigitalIO. It is used for communication purposes via PortTalk 2.0.
- **MCP::AndorCam** AndorCam module provides an interface to the Andor EMCCD iXon camera, which is used to capture 2D images of the trapped ions.

[16] https://www.qt.io/
[17] https://qwt.sourceforge.io/
[18] https://www.fftw.org/
[19] https://opencv.org/
[20] https://eigen.tuxfamily.org
[21] https://github.com/softwareQinc/qpp
[22] https://sourceforge.net/projects/opcclient/
[23] https://www.openssl.org/
[24] https://json.nlohmann.me/
[25] https://github.com/badaix/jsonrpcpp

- **MCP::ComSocket** ComSocket module contains classes to create sockets for network communication via IP addresses and port numbers.
- **MCP::DAC8814** DAC8814 library is used to communicate with a 16 bit digital to analog converter, also named DAC8814.
- **MCP::DigitalIO** DigitalIO library uses AllowIO, PortIODriver and IOInterface to use the LPT parallel port interface, also know as printer port.
- **MCP::FPGAV1** FPGAV1 library is a first version of the implementation of the AnalogOut, DigitalOut and Sequence classes to structure measurement sequences which can be send to an FPGA to be executed in real-time.
- **MCP::FPGAV2** FPGAV2 library is the follow up version of FPGAV1.
- **MCP::Gnuplot** Gnuplot library serves to create graphs for data visualization e.g. to present intermediate results to the user during the measurement procedure.
- **MCP::HWgpib** HWgpib library can be used to communicate with hardware via GPIB (General Purpose Interface Bus).
- **MCP::HWpmt** HWpmt library makes use of the MCP::SerialStream to write and read from a microcontroller connected to a photomultiplier tube (PMT).
- **MCP::IGnuPlot** IGnuPlot library includes a very basic C++ interface to gnuplot[26], a tool to visualize data and function in interactive plots.
- **MCP::IOInterface** IOInterface library is used by a variety of other modules, including deprecated ones such as FPGAV1 and DAC8814, but also MCP::NI and MCP::VFG150, which are still currently in use.
- **MCP::Keysight** Keysight library provides a user interface to the Keysight/Agilent E364xA voltage/current supply. Keysight current supplies are used to compensate external magnetic fields by controlling the current of coils.
- **MCP::NI** NI library makes use of the third-party library provided by National Instruments and covered in the MCPLibs::NI module, as well as the IOInterface.
- **MCP::OPC** OPC library, based on the third-party library in MCPLibs::OPC, is the user-friendly interface to communicate with hardware that uses the Open Platform Communication standards.
- **MCP::Picomotor** Picomotor libary provides classes to communicate with the New Focus Picomotor control. Picomotor screws are used to fine adjust the laser beam position inside the magnetic shielding.
- **MCP::PortIODriver** PortIODriver library is used by DAC8814 and DigitalIO.
- **MCP::RohdeSchwarz** RohdeSchwarz library can be used to remotely control the Rohde & Schwarz frequency generators.

[26] http://www.gnuplot.info/

- **MCP::RS232** RS232 library is the base for all serial communication in all current modules.
- **MCP::RSSML01_NI / MCP::RSSML01_QIV** The RSSML01_NI library is dedicated to the specific model SML01 signal generator by Rohde & Schwarz.
- **MCP::SerialPort** SerialPort library includes a small wrapper class around the boost::asio::serial_port.
- **MCP::SerialStream** SerialStream library includes a boost::asio based iostream compatible serial port class.
- **MCP::TaborWW1072** TaborWW1072 library includes the WW1072 class to remotely control the corresponding dual-channel arbitrary waveform generator by Tabor Electronics.
- **MCP::ThorlabsMotor** ThorlabsMotor library is used to control motors by Thorlabs, as in motorized rotation mounts.
- **MCP::VFG150** VFG150 library provides the software control for the versatile frequency generator VFG150.

MCPCore The MCPCore package summarizes the actual executable for the Master Control Program, the libraries to control loading the lab specific ScriptDll and multiple dependencies, as well as the creation of the graphical user interface (GUI).

- **MCP::gui** Gui library of the MCPCore package includes all necessary classes to create the graphical user interface of the control program.
- **MCP::DllLoader** DllLoader is responsible of loading the ScriptDll and its dependencies.
- **MCP::Utils** Utils library summarizes several minor tool functions used in MCP in the utils namespace.
- **MCP::Headers** Headers library of MCP serves as a header-only include of the MCP features for the libraries to be later loaded by the MCP.exe.
- **MCP::MCP** MCP libary serves as an include for libraries that do need more than just header-only support, but also MCP::gui with its data structures for e.g. data visualization.
- **MCP.exe** MCP.exe is the main executable for the master control program.

TCPCom

The TCPCom package provides a boost::asio::ip::tcp based server, client and a client based data logger. It serves to establish communication between different processes and computers.

- **TCPCom::TCPClient** TCPClient library includes the TCPCom::Client class.
- **TCPCom::TCPServer** TCPServer library contains the exported TCPCom::Server class and the Session class.
- **TCPCom::TCPDataLog** TCPDataLog library includes the TCPDataLogger, combining the Client class with features to automatically write the received data into a file.

MicroTools

The MicroTools package is a collection of generalized, non lab-specific tools. It contains libraries for measurement scheduling, data acquisition, evaluation and maintenance.

- **MicroTools::ActiveMagFieldComp** Active magnetic field compensation control via C++.
- **MicroTools::DataLogging** DataLogging library summarizes classes to be used for data logging and presentation.
- **MicroTools::Evaluation** Data evaluation in parallel to the data acquisition is one key piece of the control system. The evaluation library includes multiple classes for data processing and saving of the processed data.
- **MicroTools::LaserControl** LaserControl library is a summary of useful tools when it comes to controlling the laser beams. It includes the WavemeterLock class, which can be used for a P regulation based on a wavemeter reading.
- **MicroTools::LockGuard** LockGuard class makes use of an AnalogInNIChannel to be found in the MCP::NI module. It is able to read in a voltage level correlated to laser lock stability from the multifunction I/O devices by NI connected to the control PC.
- **MicroTools::Multithreading** This library combines different tools related to multithreading.
- **MicroTools::Optimization** Optimization library includes an Optimizer and Algorithm class and serves as a framework to optimize parameter sets based on measurement results.
- **MicroTools::PMT** Photomultiplier class of the PMT module is used to read out the data of the PMT Counter device, presented in Sec. 3.4.1.
- **MicroTools::PhaseEstimation** PhaseEstimation module offers a brute force bayesian estimator to acquire the phase and contrast out of measurement data.
- **MicroTools::ProbDistributions** This module contains classes for one-dimensional and two-dimensional discrete probability distributions.
- **MicroTools::Sensors** Sensors module includes interfaces for multiple sensors used in the lab environment.

- **MicroTools::SequenceControl** Module SequenceControl is the most important part of the software, regarding the scheduling of the measurement and recalibration sequences.
- **MicroTools::Teams** Teams module can be used to automatically send messages to Microsoft Teams channels.
- **MicroTools::Yocto** Yocto module contains the API to control the yocto weather station, measuring temperature, pressure and humidity in the lab.

MicroWavemeterScriptDll

MicroWavemeterScriptDll is specified to communicate with the wavemeter[27] and laser lock related tasks, such as control of supply voltages for fine tuning of a cavity and display of cavity transmission signals.

MicroScriptDll

MicroScriptDll package includes all hardware and setup-specific experimental code. Prior to the modularization, it was the only library to be loaded by MCP.exe. After separation into modules, the generalized packages MicroTools and MCPHardware can be easily shared between different setups and experimental libraries.

- **HDAWG** HDAWG module is used to communicate with the corresponding arbitrary waveform generator.
- **FPGAHardwareDefinition** This module uses the Bertha::FPGAV4 and Bertha::FPGAV3V4 -Adapter for the definition of the analog and digital channels of the mAWG.
- **MicroScriptDll** MicroScriptDll is the experiment-specific dynamic link library of a quantum computing experiment with micro-structured segmented ion trap. It also serves as an example for other group internal setups. Previous to the work of this thesis, it included all source code about handling the hardware and controlling sequencing and evaluation, which can now be found in improved and extended versions in the MicroTools and MCPHardware packages. This allows proper version control, separate development and unit testing of the comparably small libraries. It also allows to share the same libraries easily between multiple setups and labs. Therefore, MicroScriptDll is mostly reduced to very specific hardware and experiment related code.

[27] WSU, HighFinesse Laser and Electronics Systems GmbH

3.4.2.2 Contribution

During the work on this thesis the following contributions to the software stack of the quantum computing project were made:

- Implementation of a thread-safe double-ended queue to be used in e.g. the evaluation module.
- Establishing multithreading for measurement and evaluation using thread-safe queue.
- Implementation of the LockGuard module to monitor lock status of lasers.
- Implementation of the TCPCom module for multi-client tcp communication, including client and server.
- Implementation of ActiveMagFieldComp in C++, a library to control the active magnetic field stabilization via the MCP.
- Generalization of existing code in ScriptDll, e.g. replacing hardcoded properties with variables to prepare the code to be shared with other setups.
- Modularization of existing code in ScriptDll, e.g. restructuring classes and files to be separable and split into individual modules with proper include hierarchy.
- Creation of MicroTools package, which is a collection of generalized modules based on the code in ScriptDll.
- Transfer of generalized modules from ScriptDll to either MicroTools or MCPHardware.
- Implementation of Tracker interface in MicroTools::SequenceControl, which is used to schedule parameter tracking and recalibration measurements interleaved with running measurements.
- Integration of new features and measurements into ScriptDll, e.g. generalized ion loading, measurements with six ions, such as fault-tolerant parity check measurement and multiple calibration measurements.
- Fixing of bugs, cleaning up source code and adding documentation.

Chapter 4

Ion Shuttling Operations in Segmented RF Traps

Quantum information processing requires to perform quantum operations on individual qubits, which are part of a larger register. There are different approaches how this can be realized. One is to store single ions in one linear crystal and use tightly focused laser beams to individually address them [88]. This renders to become very difficult for large size registers. A different approach is fully shuttling-based, where only small ion crystals are formed and moved in and out of processing regions and laser beams are applied globally at an ion crystal in a laser interaction zone [85]. These approaches can be combined to take advantage of the high control of the shuttling-based architecture and reduce the shuttling overhead due to individual addressing in small-size registers. The next generation of our setup, currently under construction, is using this combination and is going to feature individual addressing for a ten-ion crystal in the laser interaction zone. The quantum information processing platform employed for this work is based on $^{40}Ca^{+}$ trapped-ion qubits, confined within a segmented radiofrequency trap. It is following the fully shuttling-based approach, using multiple ion crystals of only up to two ions and applying laser beams globally on an ion crystal within the laser interaction zone. This chapter gives an introduction on ion crystal reconfiguration operations, described in Sec. 4.1, including basic transport (Sec. 4.2), separation and merge (Sec. 4.3), and swap (Sec. 4.4). Finally, the shuttling-induced qubit phases are discussed, which arise from moving and storing ions in an inhomogeneous magnetic field along the linear trap axis (Sec. 4.5).

4.1 Qubit Register Reconfiguration

The idea of a trapped-ion Quantum Charge Coupled Decive (QCCD), was proposed by Kielpinski et al. [50], and is based on micro-structured electrode arrays, which allow for establishing multiple trapping zones. The stored qubits can be moved within such segmented trap by appropriate variation of the electrode voltages, in close analogy to moving charges in an actual CCD chip used for imaging. This idea is followed by several groups [51, 52, 55, 89, 90]. We employ an architecture very similar to the proposed QCCD, where the laser beams used for realizing all required qubit operations are statically directed to a fixed trap segment, the laser interaction zone (LIZ). For any

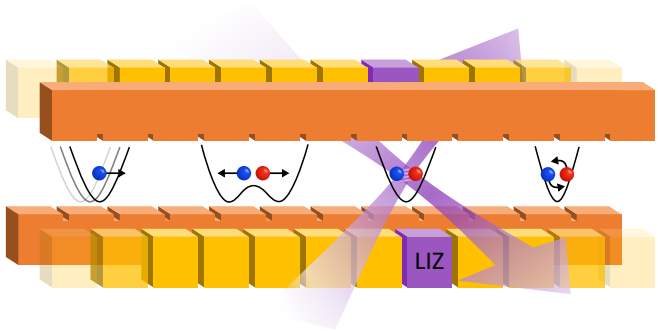

Figure 4.1: Linear segmented Paul trap architecture and reconfiguration operations, from left to right: transport, separation and merge, laser-driven quantum operations in laser interaction zone (LIZ) and ion swap (Image taken from [51] Fig. 1).

qubit operation, the subset of involved qubits has to be moved to the LIZ, while the other qubits remain stored at other segments, equivalent to storage or memory zones. Presently, the number of ion qubits commonly stored at one site, within a common potential, is limited to two. In order to move single or multiple qubits to the LIZ to carry out qubit operations, three fundamental register reconfiguration operations are needed (see Fig. 4.1):

- Ions have to be moved in and out of the LIZ and storage regions, which is realized by segment-wise **transport** of the ions, via shifting the minimum of the corresponding potential well from the initial to a neighboring destination segment.

- **Separation and merge** operations are used to deterministically split up multiple ions confined within one trapping potential into two separate ones or to merge them in a single harmonic potential well.

- Since the qubits are arranged within a 1D register, with nearest-neighbor coupling limited to the LIZ, reordering of the ions is essential for achieving an effective all-to-all connectivity. **Ion swapping** is realized via time-dependent voltage ramps on the neighboring diagonal segment pairs to rotate the ion crystal out of the RF node and exchange the ion positions. This operation can also be used as a full SWAP operation, without the need of any laser-driven gates.

Despite being optimized for low motional excitation, qubit register reconfiguration operations cause a finite amount of excitation, mostly along the axial direction. This can be tolerated, due to the fact that the gate operation does not couple to axial modes, but is only sensitive to the transversal motion of the two-ion crystal. Table 4.1 shows a summary of the currently implemented shuttling operations, as well as short- and long-term goals.

Implemented	Single- & two-ion transport over one segment
	Multi-segment transport
	Parallel transport
	Two-ion separation and merge
	Two- & three-ion SWAP
	Automated loading of multiple ion crystals
Short-term Goals	Separation and merge outside LIZ
	Variable multi-ion separation and merge
	Parallel general shuttling operations
	Shuttling operations with different ion masses
Long-term Goals	Transport in 2D RF trap architecture
	In-sequence cooling via cooling ions

Table 4.1: Summary of the qubit reconfiguration operations, which are currently implemented in our setup, which are the short-term goals to be addressed next and the long-term goals.

4.2 Transport

The movement along the x-axis of the trap is realized by a controlled movement of the potential well confining the ion crystal in this direction, which is accomplished by varying the voltages applied to the DC electrodes. Usually, the ions are stored at the segment centers. During a transport operation to a neighboring segment, the ions follow the potential minimum across the gap in between the segments and come to rest at the center of the neighboring destination segment. Concatenation of this single-segment transport allows to transport over larger distances. A typical voltage ramp to perform a single-segment transport is shown in Fig. 4.2, as well as the position of a single ion, following the shifting potential minimum over the transport time and the axial trap frequency at the ion position.
Transport operations on the µs-timescale have been experimentally demonstrated with low residual motional excitation down to the sub-phonon range [91, 92]. Although the shuttling operation does lead to oscillatory excitation, it can be cancelled out nearly completely by choosing an appropriate voltage kick at the end of the transport or by appropriately timing successive transport operations. Even though this technique is showing very low motional excitation after the transport of a single ion, it is not viable to use these within larger qubit registers due to the calibration overhead for the timing and influence on neighboring stored ions for the voltage kicks. Therefore, it is favorable to design the single-segment transport ramp in a way that it minimizes the motional excitation on the axial mode, such that the residual excitation on the axial mode and the timing overhead are low enough to be tolerated, rather than using the

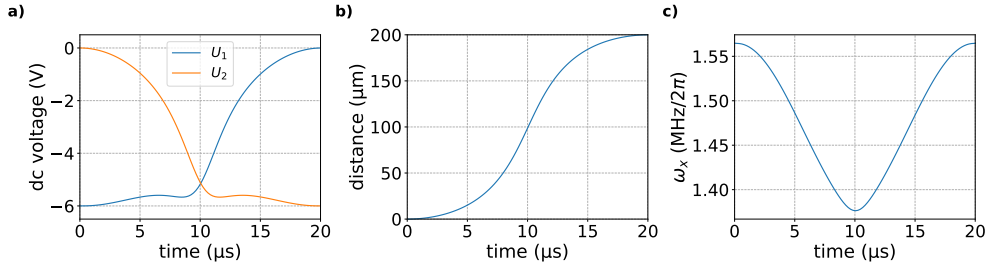

Figure 4.2: Transport to a neighboring segment. **a)** Time-dependent voltage ramps $U_1(t)$ and $U_2(t)$ on the initial and target segment, respectively. **b)** Time-dependent ion distance w.r.t. the initial position. The final position is located at the center of the target segment at a distance of 200 µm. **c)** Time-dependent variation of the axial frequency ω_x during the transport operation.

diabatic methods discussed above.

4.3 Separation/Merge

The process of fast ion crystal separation or merging with low motional excitation in the segmented micro-structured RF trap has been theoretically discussed in detail in [93] and experimentally demonstrated in [94]. Therefore, only a short summary is given here. We consider two ions commonly confined within one electrostatic potential well. For inter-ion distances which are small as compared to the segment spacing, the potential along the x-axis can be approximated using a Taylor expansion around the center of the storage segment [93]

$$\Phi(x,t) \approx \beta(t)x^4 + \alpha(t)x^2 + \gamma(t)x, \tag{4.1}$$

with the time-dependent coefficients $\alpha(t)$, $\beta(t)$ and $\gamma(t)$ being given by the trap geometry, as well as the time-dependent voltages applied to the *outer* (LIZ±2) segments $U_O(t)$, the *split* (LIZ±1) segments $U_S(t)$ and the *center* (LIZ) segment $U_C(t)$:

$$\alpha(t) = U_C(t)\alpha_C + U_S(t)\alpha_S + U_O(t)\alpha_O + \alpha' \tag{4.2}$$
$$\beta(t) = U_C(t)\beta_C + U_S(t)\beta_S + U_O(t)\beta_O + \beta' \tag{4.3}$$
$$\gamma(t) = \Delta U_S(t)\gamma_S + \Delta U_O(t)\gamma_O + \gamma' \tag{4.4}$$

The time-dependent voltages for separating a two-ion crytal to the neighboring segments are shown in Fig. 4.3, as well as the ion positions and axial trap frequency during the separation process. The merging process can be treated analogously, just as a time-inverse separation. The α_i, β_i and γ_i coefficients with $i \in \{C, S, O\}$ can be determined by the second, fourth and first derivative of the respective segment's unit

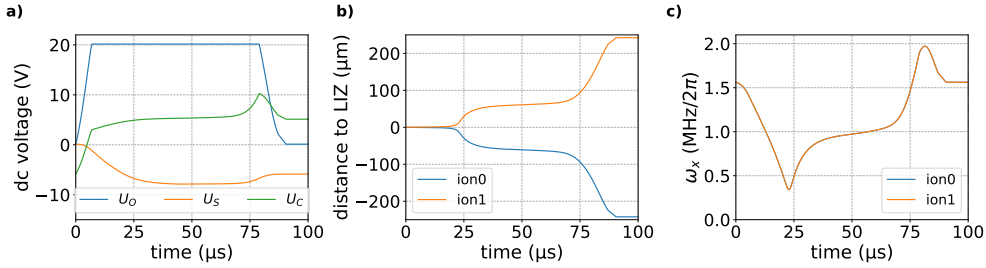

Figure 4.3: Separation of a two-ion crystal into individually confined ions. **a)** Time-dependent voltage ramps on the participating electrodes. **b)** Time-dependent position of ions during the separation operation. Final ion distance is larger than two times the segment distance of 200 µm, due to the potential barrier still present at the intermediate segment U_c. This barrier is reduced to 0 V after further transport of one of the ions. **c)** Time-dependent axial trap frequency ω_x for both ions. The confinement is significantly reduced at the critical point of the separation process.

potential, taken at the center of the C segment [93]. α', β' and γ' are spurious offset coefficients, to take into account imperfections such as stray charges and residual ponderomotive forces along the trap axis, which can be caused by manufacturing imperfections or small misalignment of the trap layers [94]. $\Delta U_S(t)$ and $\Delta U_O(t)$ are differential voltages between the corresponding electrode pairs. During the separation process, the potential well is initially dominated by the harmonic confinement with $\alpha \gg 0$ and has its minimum at the *center* segment C, which is transformed into a double-well potential with two minima at the *split* segments S. The axial confinement reaches its minimum at the *critical point* (CP), where $\alpha = 0$. To reach low motional excitation, the change rate of α should be low around the CP [94]. Due to the sensitivity of the process, possibly leading to axial mode phonon numbers on the order of 10^6 when miscalibrated, it is necessary to perform not only simulations but also calibrations of the geometry parameters for the actual trap hardware.

4.4 Ion Swap

A SWAP gate can be realized via a physical swap of the positions of two commonly confined ions, which means via controlled rotation of the ion crystal. The general swapping procedure is shown as a schematic in Fig. 4.4. The ion crystal, which is aligned along the trap x-axis, has to be rotated within the plane spanned by the axial x direction and the transverse direction with the weaker confinement. This can be achieved using a diagonal DC quadrupole potential breaking the symmetry, leading to a slight tilt of the ion crystal's alignment axis. The confinement voltage applied to the center segment is gradually changed to provide stronger axial confinement, eventually the confinement along the axial direction will exceed the confinement along the weak transverse direction and the two ions will controllably flip to vertical alignment.

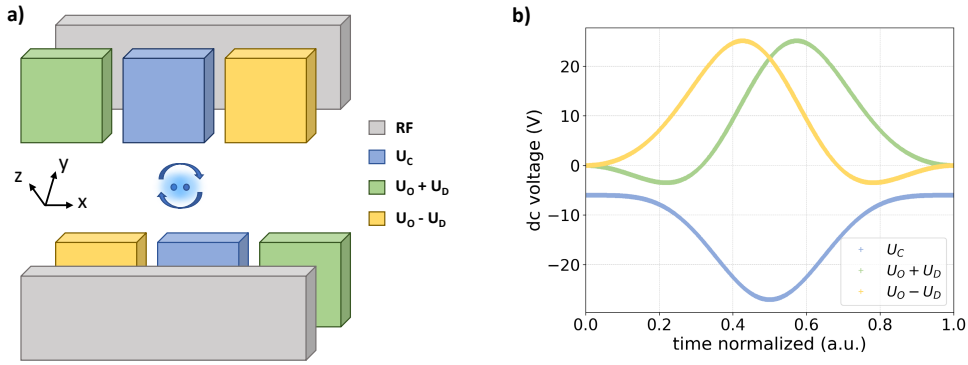

Figure 4.4: Swap operation. **a)** Shows relevant trap electrodes to realize the positional ion swap operation. **b)** Time-dependent voltage ramps on the electrodes shown in a).

At this point, the tilt quadrupole's polarity is reversed, the axial confinement is weakened again, and the tilt quadrupole is switched off. While fast ion swapping providing low motional excitation on all six secular modes of a two-ion crystal has already been demonstrated on our platform [95], it has been significantly improved within the work of this thesis. The transfer of motional excitation from the axial to the transversal modes could be reduced, by keeping the trap frequencies more constant during the swapping process. This maintains a low motional excitation on the transversal two-qubit gate mode, also under the conditions given by a complex reconfiguration sequence, leading to non-negligible excitation on the axial motional modes. Details on the modifications to the swapping voltage ramps and a comparison of the variation of the trap frequencies during the swap process is discussed in Sec. 5.4.

4.5 Shuttling-Induced Qubit Phases

The qubit frequency ω_0 is determined by the quantizing magnetic field B_0 at the ion position. This field is generated by permanent magnets outside the vacuum chamber bearing the ion trap. Since this field is not perfectly homogeneous along the trap axis, the varying field causes a dynamic variation of the qubit frequency upon transporting and storing the ions at different positions within the segmented trap. The qubit frequency ω_0 depending on B_0 is given by

$$\omega_0 = g_J \Delta m_J \frac{\mu_B}{\hbar} B_0, \tag{4.5}$$

with the Bohr magneton μ_B, Landé factor $g_J = 2$ and the difference of the magnetic quantum numbers $\Delta m_J = 1$ for the $^{40}Ca^+$ spin qubit. Along the x-axis of the trap,

the variation of the quantizing magnetic field can be described by adding a $\delta B(x)$ relative to B_0 in the laser interaction zone

$$B(x) = B_0 + \delta B(x). \tag{4.6}$$

The qubit frequency deviation $\delta\omega(x)$ relative to ω_0 in the laser interaction zone can be defined accordingly

$$\omega(x) = \omega_0 + \delta\omega(x). \tag{4.7}$$

If a qubit in a superposition state is moved along the x-axis with a time-dependent translation $x(t)$, it undergoes a rotation $\hat{R}_z(\phi(t))$ with the time-dependent phase

$$\phi(t) = \int_0^t \delta\omega\left(x(t')\right) dt'. \tag{4.8}$$

For calibration purposes, the accumulated phase and contrast loss of each individual qubit can be measured via Ramsey schemes. First the qubits are initialized to $|-\rangle$ by means of a $\pi/2$-rotation on the default initial state $|0\rangle$:

$$|-\rangle = \hat{M}_x\,|0\rangle = \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle), \tag{4.9}$$

with

$$\begin{aligned} \hat{M}_x &= \hat{R}\left(\frac{\pi}{2}, -\frac{\pi}{2}\right), \\ \hat{M}_y &= \hat{R}\left(\frac{\pi}{2}, 0\right). \end{aligned} \tag{4.10}$$

Then, the shuttling sequence is executed, with the exact timing as used in the actual measurement sequence. Due to the inhomogeneity of the magnetic field along the trap axis, the phase ϕ_c is accumulated:

$$|\Psi\rangle = \hat{R}_z(\phi_c)\,|-\rangle = e^{-i\hat{\sigma}_z\frac{\phi_c}{2}}\,|-\rangle = \frac{1}{\sqrt{2}}\left(e^{-i\frac{\phi_c}{2}}\,|0\rangle - e^{+i\frac{\phi_c}{2}}\,|1\rangle\right) \tag{4.11}$$

A dephasing channel is taken into account by adding a contrast parameter $0 \leq c \leq 1$. The density matrix in the logical basis can be written as:

$$\rho = \frac{1}{2}\begin{pmatrix} 1 & -ce^{-i\phi_c} \\ -ce^{+i\phi_c} & 1 \end{pmatrix}. \tag{4.12}$$

The independently measured probabilities $p_b^{(X,Y)}$ to end up in the state $|1\rangle$ upon readout operation $\hat{M}_{x,y}$ can be used to estimate the accumulated phase and contrast loss:

$$\begin{aligned} p_b^{(X)} &= \langle 1|\hat{M}_x^\dagger\hat{\rho}\hat{M}_x|1\rangle = \frac{1}{2}(1 + c\cos\phi_c), \\ p_b^{(Y)} &= \langle 1|\hat{M}_y^\dagger\hat{\rho}\hat{M}_y|1\rangle = \frac{1}{2}(1 + c\sin\phi_c). \end{aligned} \tag{4.13}$$

If the magnetic field gradient remains constant, these phases can be calibrated once for a given sequence of reconfiguration operations and be corrected for in the local qubit rotations adding a correction phase ϕ_c

$$\hat{R}(\theta, \phi) \rightarrow \hat{R}(\theta, \phi - \phi_c). \tag{4.14}$$

The correction phase needs to be calibrated for each individual single-qubit rotation.

Chapter 5

Advanced Register Reconfiguration Operations

The essential register reconfiguration operations in a shuttling-based trapped-ion quantum processor are transport of ions, separation/merge and positional swap operation (see Chapter 4). While it has previously been shown that all of these operations can be executed causing only low excitation on all motional modes of a two-ion crystal [91, 94, 95], further improvements were required when used within more complex sequences, such as the fault-tolerant (FT) parity check measurement (PCM) with about 100 reconfiguration operations. The combination of a high number of transport and separation/merge operations leading to significant motional excitation on the axial modes in combination with the not yet optimal swap operations, resulted in a crucial bottleneck to successfully realize the FT PCM circuit. Improvements presented in this chapter not only made the experimental results presented in Chapter 7 possible, but enable other protocols with a comparable amount of shuttling operations. Sec. 5.1 shows the implementation of automated loading of multiple ion crystals, Sec. 5.2 describes observed deviations of the trapping potential at the LIZ from neighboring potential wells and Sec. 5.3 describes the improvement of segment-wise transport. Finally, Sec. 5.4 shows how the energy transfer between axial and radial modes of motion during the swap operation can be mitigated.

5.1 Automated Loading of Multiple Ion Crystals

In the current experimental setup, only up to two ions are commonly confined within one potential well. If a sequence requires more ions, they are trapped in separate potential wells. A voltage of -2.4 V at the LIZ features a DC potential minimum at the LIZ segment center position, where the ionized atoms can be trapped. The trapping is automatically monitored via PMT count rates and optionally also via image recognition using the EMCCD camera picture [96]. Fig. 5.1 **a)** shows an EMCCD camera picture of a two-ion crystal. Ions are typically separated by about 4 µm. Upon successful trapping, photoionization lasers are blocked and the trapping potential is stiffened by ramping DC voltages applied to the corresponding electrodes to -6 V. The three- and four-qubit circuits operated on our system prior to this work included specialized loading sequences. In this work, a generalized loading sequence

called *shuffle load* was implemented, to automatically load multiple ion crystals. The number of ion crystals to be loaded is only restricted by the number of segments available to form individual potential wells on the hardware executed on. The loading process moves potential minima along the trap, such that all wells are exposed to cooling and ionization beams in a sequential manner, until all ions are at their desired location. Fig. 5.1 **b)** shows the shuttling sequence for a single loading cycle of 3×2 ions. In order to prevent unintentional loading in neighboring potential wells, three segments are kept empty neighboring the LIZ, while the distance between the stored ions only requires two empty segments in between, to allow for simultaneous transport operations of neighboring wells.

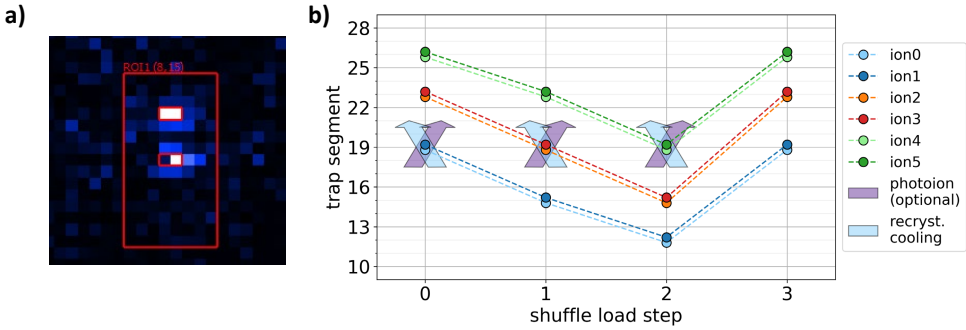

Figure 5.1: Ion crystal loading. **a)** EMCCD camera picture of two commonly confined ions, typically separated by around 4 µm due to Coulomb repulsion. **b)** Shuttling schedule for one cycle of the shuffle load sequence for 3×2 ions. A loading distance of three empty segments to LIZ is maintained, while the stored ion crystals only require a pitch of two empty segments in between. Every time a crystal stops at the LIZ, it is laser cooled using the far-detuned recrystallize cooling beam.

5.2 Separation with Surrounding Potential Wells

A detailed investigation of the separation in the presence of surrounding potential wells was performed, since separation is a process with high sensitivity to small offset forces, especially around the critical point where the single-well potential is transformed into a double-well potential. As discussed in Sec. 4.3, precise control over the separation process requires precise control over the parameter α quantifying the harmonic confinement along the trap axis. This is enabled by spectroscopic calibration of the harmonic coefficients α_i, for $i = \{C, S, O\}$ denoting the involved control electrodes. Fig. 5.2 shows measurements of the axial trap frequency ω_x vs. the voltage applied at the corresponding electrode, which are used to calibrate the respective α_i coefficients.
Throughout complex shuttling sequences, additional storage wells within the trap

significantly alter the harmonic confinement at the LIZ, which can lead to excessive motional excitation. Establishing a fixed configuration of empty or filled potential wells is therefore required to maintain symmetry around the LIZ throughout a separation or merge operation. These positions are LIZ±6, LIZ±9 and LIZ±12. While the axial secular frequencies seem not to vary a lot on the scale of the large plots, a deviation of 1 to 3 kHz is visible when zoomed in. Adding the additional potential wells leads to a systematically lower axial frequency. The deviation is small enough, such that the separation and merge operation is still functional within the larger sequence, but to perform optimizations of ion separations, it is beneficial to use the α parameters of the calibration including the actual potential situation, as present in the sequence. A linear fit to the squared axial secular frequency vs. control voltage yields the corresponding α_i parameter with $i \in \{C, S, O\}$. This helps maintain moderate motional excitation on the axial modes, especially for repeated executions of this operation. Fig. A.5 and A.3 in the Appendix show the reduction of axial motional excitation when using the calibration including the potential situation with additional wells at positions LIZ±6, LIZ±9 and LIZ±12, compared to the standard calibration with 0 V at all segments other than the LIZ. The shift of ω_x also causes a shift in the radial rocking modes (see Sec. 2.5). A frequency deviation of up to 3 kHz can be observed when comparing the frequency of the radial rocking mode with and without the additional potential wells (see Appendix Fig. A.6).

Another critical parameter is the splitting tilt γ', which is required due to imperfections of the trap electrodes or stray fields. A tilt voltage can be added to compensate an asymmetry in the time-dependent voltage ramps of the separation operation. This tilt voltage is chosen in order to cancel out the effective potential tilt and deterministically separate a two-ion crystal into two individually confined ions. Calibration of this parameter is carried out by scanning the tilt voltage, while testing the success of the separation process. One observes a *success window* with respect to the tilt compensation voltage, within which the crystal is properly separated. Outside this window, both ions end up either in the left or right well of the resulting double well potential. Typically, it is favorable to choose the center of this voltage window with a width of around 19 mV as the tilt voltage. This will provide the highest symmetry and minimizes the combined excitation on both ions after the separation [75]. However, it has been observed that the procedure of simply choosing the center of the window as optimal tilt voltage is not sufficient for larger shuttling sequences with multiple separation/merge operations. A combined 6 ion dark state readout (initializing in the dark state $|0\rangle^{\otimes 6}$) performed after shuttling sequences has been used as a benchmarking scenario to minimize the excitation and therefore the SPAM error after the full sequence. Fig. 5.3 shows the separation success window overlayed with the probability to read out all six ions correctly as 'dark', versus the tilt voltage. Reproducibly, the optimal position is located at an offset of about -4 mV from the right border of the splitting tilt voltage success window, which is far off the center. Even though this tilt voltage success window can drift by several tens of mV within a day, the substructure can be reproduced and is moving along with the success window. The drift of the success window is most likely caused by exposing the dielectrics between the trap electrodes to UV laser light, causing static charges, e.g. during retrapping.

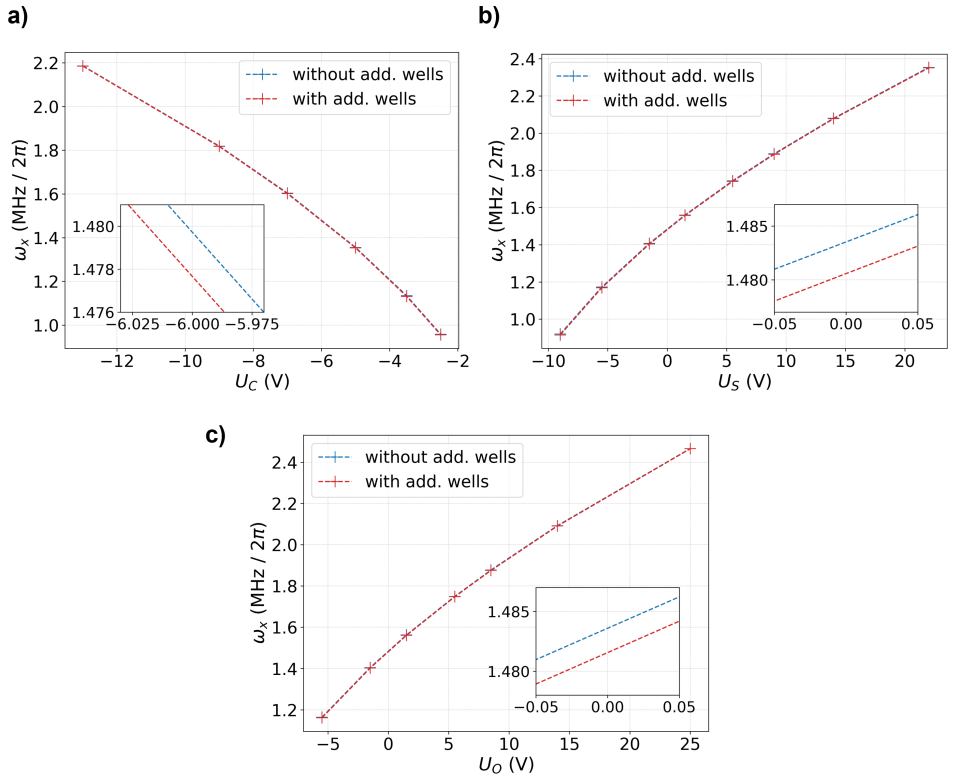

Figure 5.2: Calibration of the harmonic parameters α_i for the separation and merge operation. The axial trap frequency is measured using a single ion varying **a)** the voltage U_C at the center segment of the separation which is here the LIZ, **b)** the voltage U_S at the split segments which are placed at LIZ±1 and **c)** the voltage U_O at the outer segments at LIZ±2 while keeping the LIZ at constant -6 V. The additional potential wells are placed at the positions LIZ±6, LIZ±9 and LIZ±12.

A systematic shift of the window can be observed when varying the duty cycle of retrapping and thus the time of the trap being exposed to the photoionization beams near 423 nm and 374 nm. To address this issue of a time-dependent drift of the optimal voltage tilt, an automated software module for splitting tilt re-calibration was established, taking into account a custom offset from the success window center.

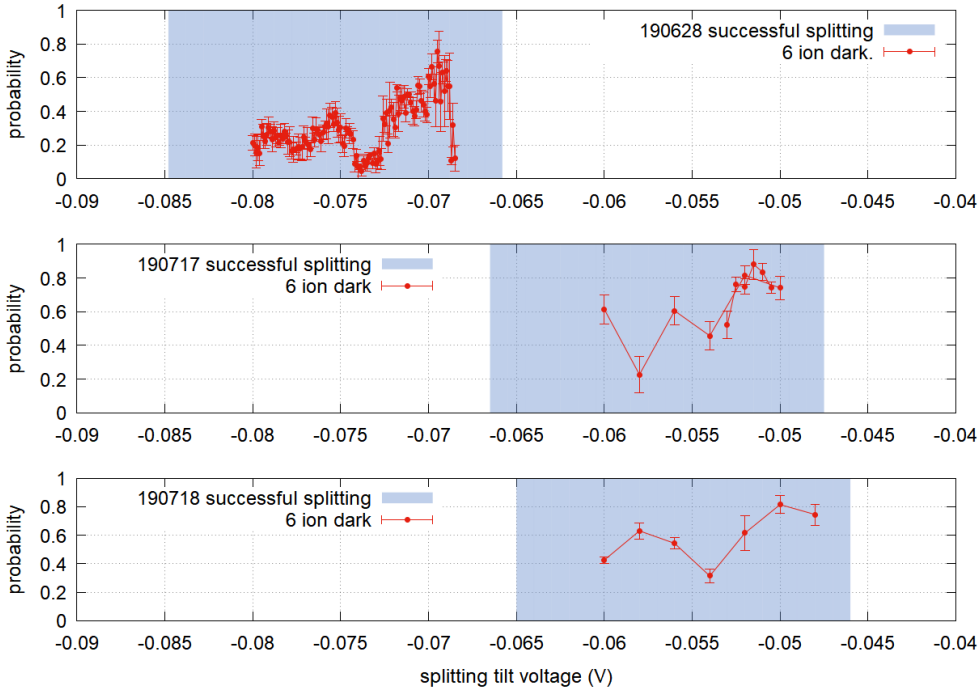

Figure 5.3: Splitting tilt voltage success window (blue) at different measurement days. The red line shows the probability to measure all six ions in the initialized state $|0\rangle$, at the end of the ion reconfiguration operations of the FT PCM scheme (shuttling scheme in Sec. 7.1, without execution of the gates). A high dark state readout indicates successful shelving and thus a low SPAM error.

5.3 Optimization of Segment-Wise Transport

Transporting ions along the trap axis can lead to excitation of ion motion in the confining harmonic potential. A transport in the adiabatic regime, where nearly no motional excitation is created, can be carried out if the transport duration is chosen to be slow enough that the ions transient displacement is smaller than the size of its ground-state wavefunction [97]. This requires rather long transport duration, which renders it to be impractical when a lot of reconfiguration operations are to be performed within the coherence time of the qubits. For shorter transport durations, the force on the ions when moving the potential causes oscillatory motion. The state of the excited ion can be described by a *coherent state*

$$|\alpha\rangle = \hat{D}(\alpha)\,|0\rangle \tag{5.1}$$

with the *displacement operator*

$$\hat{D}(\alpha) = e^{\alpha\hat{a}^\dagger - \alpha^*\hat{a}} \tag{5.2}$$

acting on the vacuum state $|0\rangle$.
The phonon distribution follows a Poissonian distribution

$$p_n^{(coh)}(\alpha) = |\,\langle n|\alpha\rangle\,|^2 = \frac{|\alpha|^{2n}}{n!}e^{-|\alpha|^2} \tag{5.3}$$

with a mean phonon number of

$$\bar{n}_{coh} = |\alpha|^2. \tag{5.4}$$

The final excitation after a transport is defined by the Fourier component of the velocity of the moving potential well at the secular trap frequency. When transporting along the trap axis x, the potential can be written as $\frac{1}{2}m\omega_x^2(x - x_0(t))^2$, where $x_0(t)$ is the time-dependent position of the potential minimum and m the mass of the ion. Assuming that initially, the ion is in the motional ground state, centered in the harmonic potential at $x = x_0(0)$, i.e. in state $|\Psi(0)\rangle \equiv |0\rangle$, the coherent state throughout the transport is given by $|\Psi(t)\rangle = |\alpha(t)\rangle$, with the amplitude $\alpha(t)$ given by [92, 98]

$$\alpha(t) \equiv \sqrt{\frac{m\omega_x}{2\hbar}}\left[x_0(t) - e^{-i\omega_x t}\int_0^t \dot{x}_0(t')e^{i\omega_x t'}\,dt'\right]. \tag{5.5}$$

In combination with the possible thermal excitation, the actual phonon probability distribution is defined by the convolution of the coherent distribution $p_n^{(coh)}$ (see Eq. 5.3) and the thermal distribution $p_n^{(th)}$ (see Eq. 2.29). Coherent excitation created by the displacement operator $\hat{D}(\alpha)$ can be cancelled out by application of another displacement operator with the same amplitude, but its phase shifted by 180°, expressed as $\hat{D}(-\alpha)$. It has been shown experimentally that this can be utilized to counteract the excitation and realize transport at the timescale a few secular cycles at sub-phonon residual excitation [91, 92]. There are two different approaches to

neutralize the excitation caused by the transport. Either one has to time the start of the next transport with respect to the phase of the ion motion in order to counteract the oscillatory motion, or to apply a short voltage kick at the end of the transport, which requires precise calibration of its timing and amplitude. However, such diabatic techniques are impractical for larger qubit registers and complex shuttling sequences. Both methods require a prohibitive amount of calibration measurements when dealing with larger registers. In addition, the voltage kick method has the disadvantage of impacting not only on the ions which excitation is supposed to be counter-kicked, but also on other ions stored within the trap. Parameters chosen to optimally reduce the excitation on the target ion crystal, could even increase the oscillatory excitation on neighboring stored ions. Therefore, for the use of larger registers, it is currently favorable to perform the transports not at the high speed of 3.86 µs/segment as in [91], but at a reduced acceleration and a speed of around 20 µs/segment. This transport duration is at the crossover from the diabatic to the adiabatic regime and represents a trade-off between duration and final excitation of the axial motion along the trap axis. An axial motional excitation of a few phonons can be tolerated, since neither the quantum gate operations, nor the shelving for the spin-selective readout are sensitive to the axial mode. A very high excitation on the other hand would lead to anharmonic coupling of the radial modes and can be transferred to radial modes via the swap operation (see Sec. 5.4).

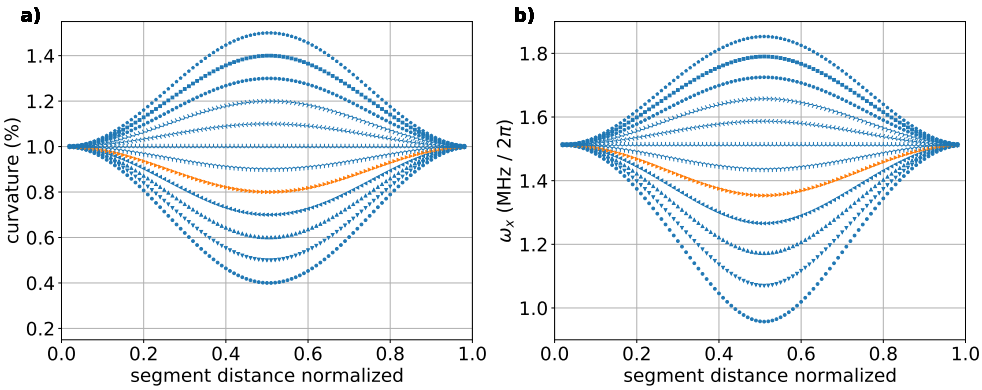

Figure 5.4: Curvature and axial trap frequency during transport operation. **a)** Set of curvature c vs. normalized transport position x for target curvatures at half the transport in between the two electrodes ranging from 40% to 150% in steps of 10%. Highlighted in orange 80%, which is experimentally found to cause the lowest excitation. **b)** trap frequencies during the transport of all ramps of a), calculated based on the re-scaled voltage ramps starting and ending at $-6\ V$.

Detailed investigations of the voltage ramps used for the transport were performed, in order to reduce the excitation per transport without the need of actively counter-acting it. In particular, the axial confinement in between segments, at half the transport

duration, was varied to experimentally identify the optimum. Fig. 5.5 shows the transport ramps, based on a simulation of the electric field generated by the DC electrodes as described in Sec. 2.4. The curvature ramps $c(x)$ and the corresponding trap frequency ramps $\omega_x(x)$ are shown in Fig. 5.4. The trap frequency ω_x directly relates to the second derivative of the axial electrostatic potential, i.e. its curvature, via

$$\omega_x = \sqrt{\frac{e}{m}\Phi''}, \tag{5.6}$$

with mass m of the ion and electron charge e. The set of position-dependent curvatures $\Phi''(x) \equiv c(x)$ is generated based on a cos-shaped ramp with a varying target curvature at half the transport duration, i.e. at the position between the segments, given by $c(x) = 1 - 0.5(1 - c_t)(1 - cos(x))$, with the target percentage c_t at half the transport varied between 40% and 150% in steps of 10%, where 100% corresponds to the reference curvature at the start of the transport. Based on a given curvature ramp $c(x)$. Voltages required to generate this curvature can be calculated by solving the equation system [1]

$$\begin{aligned} 0 \equiv \Phi'(U_i, U_{i+1}, x) &= U_i\Phi_i'(x) + U_{i+1}\Phi_{i+1}'(x), \\ c(x) \equiv \Phi''(U_i, U_{i+1}, x) &= U_i\Phi_i''(x) + U_{i+1}\Phi_{i+1}''(x), \end{aligned}$$

with x being the position of the potential minimum and voltages U_i and U_{i+1} applied to the initial and destination segments of the transport, respectively. The potential $\Phi_i(x)$ is determined by the trap simulation, assuming 1 V applied to the electrode pair comprising segment i. Voltage ramps are re-scaled to the usual storage voltage of -6 V as a start and end condition (see Fig. 5.5) and used for the calculation of the simulation-based estimation of the axial trap frequency ω_x during transport (see Fig. 5.4 **b)**).

In order to characterize the impact of the transiently altered axial confinement, we perform measurements of the final motional energy for different transport voltage ramps. A single ground-state cooled ion is transported from the laser interaction zone to a neighboring segment. Since there is only a single LIZ, a back-transport is required for the measurement. The ion is moved back after a waiting time of around 30 µs and a variable additional delay time. The aim is to find transport parameters leading to reduced motional excitation, independently of the exact timing. For each voltage ramp, we measure the residual motional energy versus the additional delay time between the single-trip transports and the overall duration of a single-trip transport. Measurement of Rabi flops on the carrier, blue and red sideband of the axial mode can be used to identify the phonon number as described in [51]. The necessary amount of measurements to investigate transport curves and timing parameters required a reduced measurement scheme instead of a full and precise quantification. Only the excitation probability of the red sideband is measured, at a pulse area given the π-time of ground-state cooled blue-sideband Rabi flops. Close to ground state, this bright

[1] using the NSolve functionality of Wolfram Mathematica

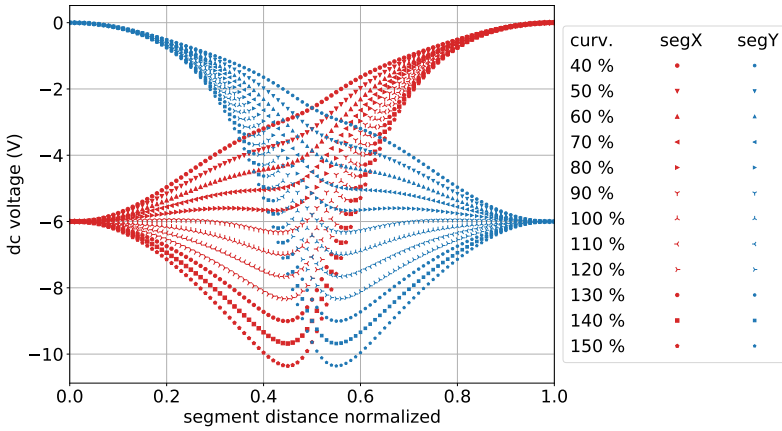

Figure 5.5: Set of transport voltage ramps from segment X to segment Y. The curvature in the center between the segments is varied from 40% to 150% in steps of 10%.

probability is expected to be close to zero, while it increases for increasing phonon numbers. The dynamic range of this measurement is limited to small excitations $\bar{n} \lesssim 5$. The measurement results are shown in Fig. 5.6, with the optimum to be the transport ramping the curvature between segments down to 80% of its initial value.

Even though the trap segments are uniform, i.e. all segments ideally feature the same geometry, and the LIZ is located sufficiently far away from the trap edges, there appears to be a high asymmetry when transporting from segment 19 (LIZ) to segment 18 as compared to segment 20. This asymmetry is possibly caused either by the segment itself or by the DC voltage supply lines. Detailed measurements of the properties of the supply channels could not reveal any issue regarding the voltage channels of the arbitrary waveform generator, the filter boards or the trap itself. A full characterization remains very difficult, there are no means of measuring the actual time-dependent voltage levels at the trap electrodes. A different behavior of the two trap segments could be confirmed more directly using voltage kicks to controllably excite a ground-state cooled ion stored at the LIZ. Using the same voltage kick either on segment 18 or segment 20 leads to significantly different excitation values, which indicates a spurious asymmetry of the trap geometry, the DC control lines or both. Corresponding measurements of Rabi flops with varying voltage kick parameters on different segments are shown in the Appendix, Fig. A.7. Application of static voltages at slow timescales was found to cause a roughly equivalent position shift of a single ion on the EMCCD camera. These investigations point towards different filtering characteristics on the two compared channels, causing a different dynamic behaviour. Based on these findings, ongoing improvements of the analog channels, as well as a new filter design by Alexander Stahl and Maximilian Orth [86] are possible ways to resolve this issue in the near future and to further improve our fast transport capabilities. The investigation of transport operations revealed a problematic asymmetric behaviour of

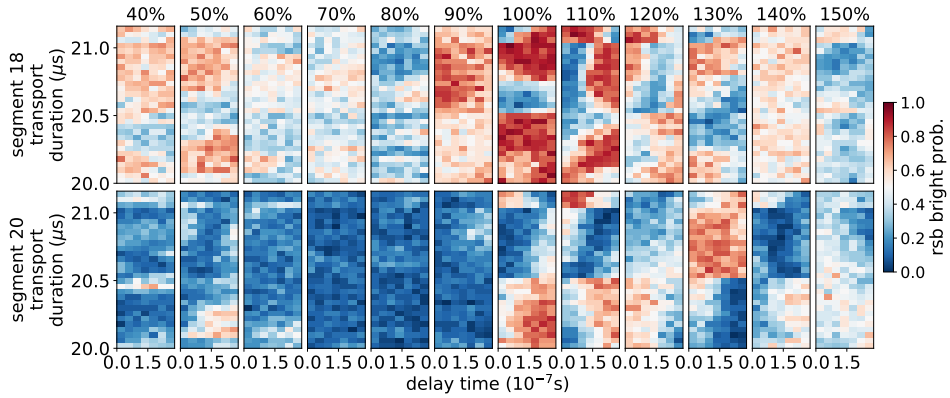

Figure 5.6: Probing axial excitation after transport, varying duration and delay time. Shown is the bright probability after red side band (rsb) pulse at cooled blue side band (bsb) π-time after a transport from segment 19 (LIZ) to either segment 18 or segment 20, indicating the motional excitation after transport. Scanned are the duration of transport and variable delay time, which is waiting time on the remote segment before transport back (plus constant offset). The scans are performed for the set of curves shown in Fig. 5.5.

the trap segments. Until the release of new trap or filter hardware, segment-wise transport ramping down the curvature to 80% in between the segments is used to reduce excitation.

5.4 Positional Ion Swap with Low Secular Frequency Variation

Positional ion swap operations are used to rearrange the order of ions along the linear trap. This allows the combination of any two ions within a common potential well and is of crucial importance to realize effective all-to-all connectivity (see Sec. 4.4). It was previously shown that our time-dependent voltage ramps used to rotate a two-ion crystal only lead to minimal excitation on all motional modes [95]. When building a larger reconfiguration sequence out of the standardized building blocks for transport, separation/merge and swap, a significant reduction of the entangling gate fidelity was observed. Investigations of the syndrome qubit contrast after performing entangling gates in the FT PCM scheme, show a significant dependence on the execution of SWAP operations in the circuit (more details are shown in the corresponding FT PCM Sec. 7.4). Performing the swap operation in between 100 reconfiguration operations leads to an increased motional excitation on the radial modes of motion. Since the entangling gates are performed on radial modes, excitation on one of the transverse modes reduces the entangling gate fidelity [67, 84] and needs to be avoided.

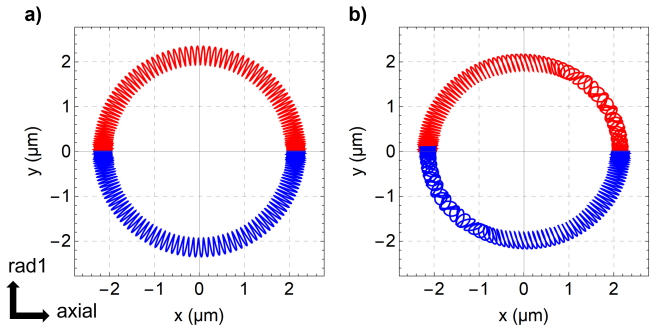

Figure 5.7: Classical simulation of motion of the first (red) and second (blue) ion during ion swap operation, assuming an initial out-of-phase motion along the trap x-axis (excitation of the axial stretch mode). The y-axis corresponds to the transverse axis with lower confinement, referred to as rad1. **a)** Swap operation where the frequencies of motional modes are kept constant. The initial and final excitation do not significantly differ. **b)** Swap operation with a degeneracy of the axial stretch and rad1 com mode frequency at vertical ion crystal alignment. The classical simulation reveals a significant difference in the initial and final motion of the ions. The motion in axial direction is transferred to radial motion along the y-axis throughout the swap process.[2]

[2]Simulations of the ion motion were performed by Ulrich Poschinger using Wolfram Mathematica and a recent boundary element method based simulation of the trap potential. The simulation of trap potentials was carried out by Maximilian Orth using COMSOL [86].

A classical simulation of the ion motion reveals the possibility of energy transfer from the axial mode of motion into the radial mode. Fig. 5.7 shows the simulation in case of a transient degeneracy of the axial stretch mode and lower-frequency radial center-of-mass mode.
Frequency spectra showing the variation of motional sidebands during the swap operation are shown in Fig. 5.8. The modes frequencies exhibit significant variations. Mode crossings do occur, in particular for the axial center-of-mass and lower-frequency radial rocking modes, as well as the axial stretch and lower-frequency radial center-of-mass modes. A detailed probing of the excitation, using Rabi flops on the red and blue sidebands of the modes, confirms an increased excitation of the axial stretch and the lower-frequency radial center-of-mass mode after a swap operation in the presence of injected excitation on the axial center-of-mass mode prior to the swap operation (see Appendix Fig. A.8). The aim is to reduce this variation of the frequencies and thus to avoid degeneracy and energy transfer between the modes.

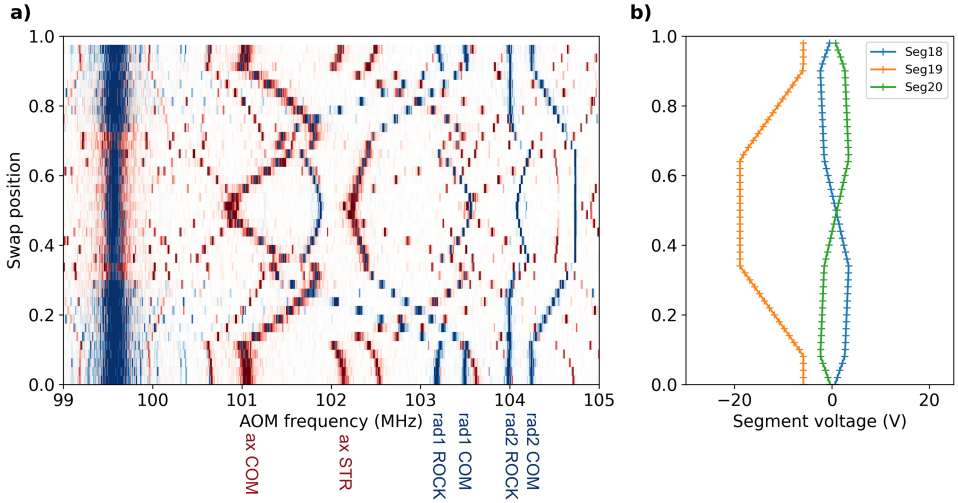

Figure 5.8: Typical swap operation prior to the optimization of the time-dependent voltage. **a)** 2D frequency spectrum monitoring frequencies of the motional sidebands during the swap operation. Two laser beam configurations are used, R1 & R2 (red) and R1 & R4 (blue), in order to investigate the axial and radial modes of motion throughout the whole process (see Sec. 3.3). The frequencies do highly vary during the operation and there are multiple crossings of axial com and rad1 rock, as well as axial stretch and rad1 com. **b)** Voltage ramps applied to the top electrode of segment 18, 19 (LIZ) and 20 vs. the relative swap position. For the bottom electrodes, voltages at segment 18 and 20 are swapped, as described in Sec. 4.4.

Based on boundary element method (BEM) simulations of the potential in the trap, time-dependent voltage ramps are calculated in order to reduce secular frequency

variations during the position exchange. Imperfections of the trap, which are not included in the simulation, are partially compensated for by including a scaling parameter s in the voltage ramps. Time-dependent voltages for each of the six involved control electrodes can now be calculated using the three predefined ramps $U_O(t), U_D(t)$ and $U_C(t)$ shown in Fig. 5.9 to be:

$$\begin{aligned} U_{LIZ,top\&bottom}(t) &= U_C(t), \\ U_{LIZ-1,top}(t) &= sU_O(t) + U_D(t), \\ U_{LIZ+1,top}(t) &= sU_O(t) - U_D(t), \\ U_{LIZ-1,bottom}(t) &= sU_O(t) - U_D(t), \\ U_{LIZ+1,bottom}(t) &= sU_O(t) + U_D(t), \end{aligned} \tag{5.7}$$

where U_C is the center voltage, U_O the offset on all neighboring electrodes and U_D the diagonal voltage on the neighboring electrodes, which mediates to the position exchange.

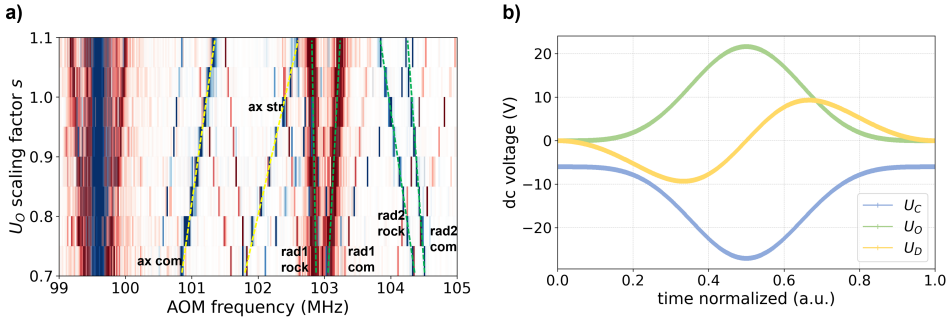

Figure 5.9: Scalable time-dependent voltage ramps of the swap operation. **a)** Frequency spectra at a relative swap position of 0.5, i.e. vertical alignment of the two-ion crystal. The scaling parameter s of the offset voltage U_O is varied, to identify the optimal offset for reduced frequency variation during the swap. **b)** Time-dependent voltage ramps for the center electrode U_C, as well as offset U_O and diagonal U_D voltage ramps, which are used to supply the electrodes according to Eq. 5.7.

A scaling parameter of $s = 0.8$ leads to similar separation of the frequencies at half the swap, i.e. at vertical ion crystal alignment, as compared to the usual trapping configuration with crystal alignment along the x-axis. Detailed measurements of secular frequencies during the swap operation using the improved voltage ramps and scaling parameter of 0.8 shows that degeneracies of first order motional sidebands are avoided (see Fig. 5.10). Probing of motional sideband Rabi flops after deliberately introducing axial excitation confirms a significant improvement. The measurement data shows no significant excitation transfer after swap operations (see Appendix

Fig. A.9). This improvement allows for the use of swap operations within sequences showing significant axial motional excitation. The axial excitation can be tolerated, since it is no longer transferred into transversal motional excitation.

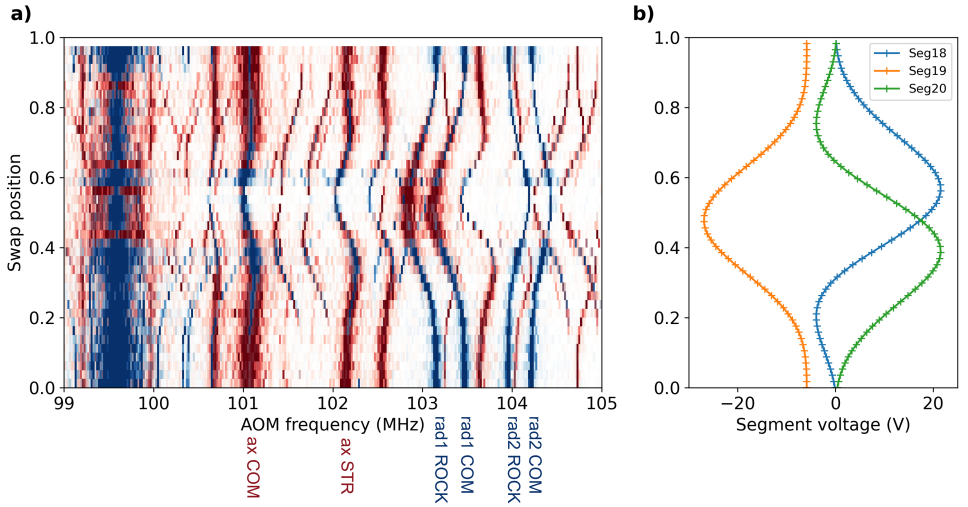

Figure 5.10: Improved swap operation with reduced sideband frequency variation. **a)** 2D frequency spectrum monitoring the frequencies of motional sidebands during the swap operation. Two laser beam configurations are used, R1 & R2 (red) and R1 & R4 (blue), in order to investigate the axial and radial modes of motion throughout the entire process (see Sec. 3.3). Traces of the first sidebands do not cross anymore. **b)** Voltage ramps applied to the top electrodes of segment 18, 19 (LIZ) and 20 vs. the relative swap position.

Chapter 6

Quantum Error Correction

Fault-tolerant parity check measurements, experimentally realized within this work, serve as an important building block for multiple error correction codes, such as the topological [[7,1,3]] code. This chapter introduces the concept of logical qubits based on encoding quantum information in multiple physical qubits and quantum error correction (QEC) codes. The stabilizer formalism can be used to describe QEC codes efficiently. General examples in this chapter, such as the 3-qubit code examples, can be found in more detail in introductory QEC literature such as [35, 99, 100].

6.1 Logical Qubit Encoding

Quantum information is using the qubit as a fundamental unit. In contrast to the classical bit, a qubit does feature not only binary states 0 and 1, but coherent superpositions of the two orthonormal basis states $|0\rangle$ and $|1\rangle$ in the form of

$$|\psi\rangle = \alpha\,|0\rangle + \beta\,|1\rangle\,, \tag{6.1}$$

where the prefactors fulfill the condition $|\alpha|^2 + |\beta|^2 = 1$. Basis states can be expressed in the vector representation as

$$|0\rangle \equiv \begin{pmatrix} 1 \\ 0 \end{pmatrix}, |1\rangle \equiv \begin{pmatrix} 0 \\ 1 \end{pmatrix}. \tag{6.2}$$

For quantum gate operations the matrix needs to be unitary, which means $\hat{U}^\dagger\hat{U} = \hat{U}\hat{U}^\dagger = I$, where $\hat{U}^\dagger$ is the adjoint of $\hat{U}$. Any single-qubit rotation $\hat{U}$, can be expressed (up to a global phase) as a linear combination of the Pauli matrices

$$\hat{\sigma}_x \equiv \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}, \hat{\sigma}_y \equiv \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix}, \hat{\sigma}_z \equiv \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix} \tag{6.3}$$

and the 2×2 identity matrix $\mathbb{1}_{2\times2}$

$$\hat{U} = c_I \mathbb{1}_{2\times2} + c_x\hat{\sigma}_x + c_y\hat{\sigma}_y + c_z\hat{\sigma}_z, \tag{6.4}$$

where c_I is a real number, while $c_{x/y/z}$ are complex numbers.
Classical error correction is often based on redundancy: Multiple copies of data is stored, computations or transmissions performed and the results are compared to verify integrity. Such techniques are not applicable to quantum error correction due to the no-cloning theorem [101]. Another limitation, not present in classical computing, is that no direct measurement on the data qubits can be performed during the correction procedure. Upon measurement, superposition states collapse to the chosen measurement basis, such that the stored information is altered due to this inevitable measurement back-effect. Therefore, a different measurement technique is required to detect errors, without actually measuring the states of the data qubits themselves. An example of error syndrome readout is the mapping of the data qubit register parity information onto an auxiliary qubit. This auxiliary qubit does not reveal the data qubit state, but only holds the information of even or odd data qubit register parity. A third important difference to classical bits is that not only bit flip errors of the form $|0\rangle \leftrightarrow |1\rangle$ can occur, but also phase errors of the form $|0\rangle \leftrightarrow |0\rangle$, $|1\rangle \leftrightarrow -|1\rangle$. Additionally, these errors on qubits are continuous, the qubit undergoes for example only a tiny phase shift, instead of a full spin or phase flip error. Upon measurement of the auxiliary qubit holding the error syndrome information, the data qubits are projected into an erroneous or non-erroneous state. The error is digitized and thus becomes correctable.
Error sources can be imperfect gate operations, depolarization and decoherence induced by coupling to the environment, but also state initialization and measurement errors, as well as qubit loss and qubit leakage. How to handle and suppress these, as well as their relevance as a significant error source, is depending strongly on the chosen physical implementation of the qubit. A detailed review on error sources for trapped-ion qubits in radiofrequency traps is given by Ballance [84].
Although copying of a quantum state is not possible, other forms of data redundancy can be established. Multiple physical qubits are used to encode a so-called logical qubit. A commonly used example to introduce this concept is the 3-qubit bit-flip code, that can be extended to a full QEC scheme with the 9-qubit Shore code [28]. A single-qubit in state $|\psi\rangle$ is coupled with two other qubits via CNOT gates. Three physical qubits serve to encode a single logical qubit with the two basis states

$$\begin{aligned} |0\rangle \rightarrow |0\rangle_L &\equiv |000\rangle\,, \\ |1\rangle \rightarrow |1\rangle_L &\equiv |111\rangle\,. \end{aligned} \tag{6.5}$$

A single physical qubit's state $|\psi\rangle$ is then mapped to the logical qubit state $|\psi\rangle_L$ according to

$$|\psi\rangle = \alpha\,|0\rangle + \beta\,|1\rangle \rightarrow |\psi\rangle_L = \alpha\,|0\rangle_L + \beta\,|1\rangle_L\,. \tag{6.6}$$

The two so-called codeword states, $|0\rangle_L$ and $|1\rangle_L$ feature a code distance of $d = 3$, since three bit flips would be needed to transfer from one to the other state. Therefore, the number of errors which can be corrected for is given by $t = \lfloor (d-1)/2 \rfloor$ to $t = 1$. To correct not only bit flip, but also phase flip errors, more advanced quantum error

correction codes can be used, which encode the quantum information in a higher number of physical qubits. The following Sec. 6.2 provides more details on how error detection and correction is implemented and used.

6.2 Introduction to Quantum Error Correction Codes

The basic principle of quantum error correction codes can be introduced utilizing the 3-qubit code for logical qubit encoding introduced in the previous Sec. 6.1, described in more detail in [99]. We assume that Alice and Bob want to transfer qubits through a noisy channel, as illustrated in Fig. 6.1. First, Alice does encode the state $|\phi\rangle$ using two additional physical qubits, preparing the logical qubit $|\phi\rangle_L$ according to the mapping $\alpha|0\rangle + \beta|1\rangle \rightarrow \alpha|000\rangle + \beta|111\rangle$. This encoding is realized by preparing two qubits in the state $|0\rangle$ and applying CNOT gates between the first and second, as well as between the first and third qubit, with the first one being the control qubit. Then, this encoded qubit state is transmitted to Bob via a noisy channel. Due to the simplicity of this introductory example, there are some stringent assumptions about the noise. The noise acts on the physical qubits in an uncorrelated manner. For each qubit, the noise does either apply a Pauli-X operator on the qubit, hence a spin flip error, or leaves the qubit unchanged. There is a higher probability $(1-p)$ to leave the qubit unchanged, than to perform the spin flip error with probability $p < 1/2$.
As the qubits that Bob receives are affected by noise in the channel, Bob has to perform error correction. He introduces two auxiliary qubits, prepared in $|0\rangle$, in order to gain information about the error. This requires four more CNOT operations, two between the first and second qubits received from Alice and Bob's first auxiliary and two between the first and third qubit from Alice and Bob's second auxiliary qubit. These operations are shown in the *correct* block of Fig. 6.1. Table 6.1 shows the possible states of the received qubits, the corresponding auxiliary qubit measurement outcome after the CNOT operations and the probability to receive this state, depending on the p that a σ_x error occurred on a qubit due to the noise.
Bob does only measure the auxiliary qubits, not the qubits received by Alice. These two auxiliary qubits reveal two classical bits of information after the measurement, which is the so-called *error syndrome*. Bob can now act on the data qubits, based on the information he got from the syndrome measurement. It shall be noted that there are different states, which lead to the same auxiliary outcome, e.g. measuring $|01\rangle$ could belong to the state $\alpha|001\rangle + \beta|110\rangle$, but also to $\alpha|110\rangle + \beta|001\rangle$. Under the condition $p < 1/2$, the first one has a higher probability than the latter one. This is why Bob is going to assume that a single error on the 3. qubit occurred, instead of two errors on the 1. and 2. qubit. The correction operations of Bob, depending on the auxiliary measurement outcome, are shown in Table 6.2. Finally, Bob decodes the logical qubit by applying two CNOT operations between the first and second, and first and third data qubit. The state after this decoding operations is given either by the original state $\alpha|0\rangle + \beta|1\rangle$ or this state after an additional σ_x operation $\alpha|1\rangle + \beta|0\rangle$. The probability for this error to occur on more than two qubits is given by the sum of

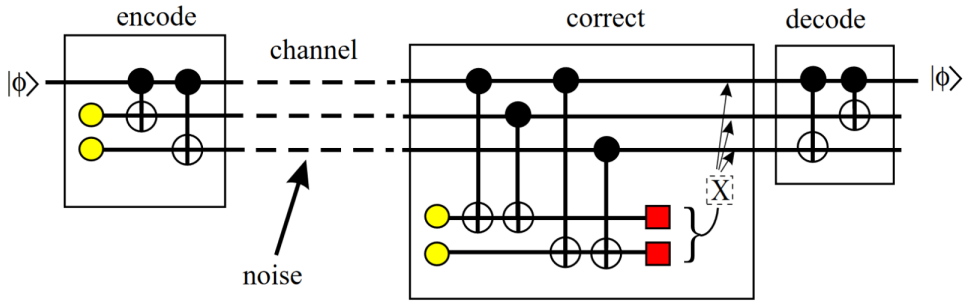

Figure 6.1: Example illustrating the encoding, correction and decoding of the 3-qubit code (Illustration taken from [99], Fig. 1). Alice wants to transmit the qubit state $|\phi\rangle$ to Bob through a noisy channel. In the encoding block, this single-qubit state is encoded in three qubits, using two more qubits initialized in $|0\rangle$, represented by yellow circles. The encoding is realized via two CNOT operations. Bob receives this three qubits and wants to recover the joint state using a correction block before decoding. The correction block uses two more qubits, initialized in $|0\rangle$ (yellow circles), and four CNOT operations. Red squares indicate the readout of these two auxiliary qubits. Depending on this measurement outcome, a σ_x operation is optionally applied to one of the data qubits. After the correction block, Bob uses two CNOT operations for decoding.

State	Auxiliary qubits	Probability	Error
$\alpha\,\|000\rangle + \beta\,\|111\rangle$	$\|00\rangle$	$(1-p)^3$	no err.
$\alpha\,\|100\rangle + \beta\,\|011\rangle$	$\|11\rangle$	$p(1-p)^2$	err. on 1. qubit
$\alpha\,\|010\rangle + \beta\,\|101\rangle$	$\|10\rangle$	$p(1-p)^2$	err. on 2. qubit
$\alpha\,\|001\rangle + \beta\,\|110\rangle$	$\|01\rangle$	$p(1-p)^2$	err. on 3. qubit
$\alpha\,\|110\rangle + \beta\,\|001\rangle$	$\|01\rangle$	$p^2(1-p)$	err. on 1. & 2. qubit
$\alpha\,\|101\rangle + \beta\,\|010\rangle$	$\|10\rangle$	$p^2(1-p)$	err. on 1. & 3. qubit
$\alpha\,\|011\rangle + \beta\,\|100\rangle$	$\|11\rangle$	$p^2(1-p)$	err. on 2. & 3. qubit
$\alpha\,\|111\rangle + \beta\,\|000\rangle$	$\|00\rangle$	p^3	err. on 1., 2. & 3. qubit

Table 6.1: Possible states of the three qubits sent by Alice and received by Bob. The auxiliary qubit measurement results with the corresponding probability and occurred errors are shown.

Auxiliary qubit	Bobs Action
$\|00\rangle$	None
$\|01\rangle$	σ_x on 3. qubit
$\|10\rangle$	σ_x on 2. qubit
$\|11\rangle$	σ_x on 1. qubit

Table 6.2: Auxiliary qubit measurement and the corresponding error correction operation performed by Bob. The correction operation is chosen according to the initial state with highest probability.

the corresponding probabilities in Table 6.1 $3p^2(1-p)+p^3 = 3p^2-2p^3$. This is smaller than p, for small error rates $p < 1/2$. Applying the encoding and correction to a 3-qubit logical state, Bob reduces the probability of failure on the estimated state from $\mathcal{O}(p)$ to $\mathcal{O}(p^2)$. The same principle can be transferred to quantum error correction codes with more qubits and larger error correction capabilities.
As shown by this 3-qubit spin-flip correction example, no error correction code is going to fully recover the original state from a compromised state. However, quantum error correction renders the scaling of register corruption probability with respect to the intrinsic error rates more favorable, keeping the correct quantum information alive. Concatenated error correction requires more qubits, but can increasingly well preserve the encoded information [102].

6.3 Stabilizer Formalism

For the description of the 3-qubit code example, the state vector representation of the encoded states is used. This turns out to be rather inefficient for larger codes. The stabilizer formalism, first introduced by Daniel Gottesman in [103] and also presented by Calderbank et al. [104], makes use of the Heisenberg representation for quantum mechanics. A state is described by the use of operators instead of the vector representation. An operator is defined to *stabilize* the state $|\psi\rangle$, if $|\psi\rangle$ is a $+1$ eigenstate of the operator g, such that

$$g\,|\psi\rangle = |\psi\rangle\,. \tag{6.7}$$

A very simple example is the operator σ_z to be a stabilizer for the single qubit state $|0\rangle$, since $\sigma_z\,|0\rangle = +1\,|0\rangle$.
A subgroup of all single qubit operations is the Pauli group

$$\mathcal{P} = \{\pm\sigma_I, \pm i\sigma_I, \pm\sigma_x, \pm i\sigma_x, \pm\sigma_y, \pm i\sigma_y, \pm\sigma_z, \pm i\sigma_z\}. \tag{6.8}$$

$\mathcal{P}$ can be extended from a single qubit to N-qubits by taking the N-fold tensor product [35], such that

$$\mathcal{P}_N = \mathcal{P}^{\otimes N}. \tag{6.9}$$

An N-qubit state $|\psi\rangle_N$ can then be defined by a subgroup of the N-qubit Pauli group, the set $\mathcal{S}$ of N generators g_i, which are stabilizing this state, where

$$\mathcal{S} = \{\ g_i \mid g_i |\psi\rangle = |\psi\rangle\,,\ [g_i, g_j] = 0,\ \forall\ (i,j)\} \subset \mathcal{P}_N. \tag{6.10}$$

By definition, $\mathcal{S}$ is an Abelian subgroup, as all elements mutually commute. Each stabilizer of the group is Hermitian and squares to identity. The product of two stabilizer generators is also a stabilizer, since

$$g_i g_j |\psi\rangle = g_i(+1)|\psi\rangle = (+1)|\psi\rangle\,. \tag{6.11}$$

As an example one can consider the three qubit bit flip code with the basis states $|0\rangle_L = |000\rangle$ and $|1\rangle_L = |111\rangle$ as given in Eq. 6.5 [2]. The stabilizer generators in this example are given by

$$g_1 = \sigma_z \otimes \sigma_z \otimes \sigma_I \equiv Z_1 Z_2, \tag{6.12}$$

$$g_2 = \sigma_I \otimes \sigma_z \otimes \sigma_z \equiv Z_2 Z_3, \tag{6.13}$$

forming the set

$$\mathcal{S} = \{Z_1 Z_2, Z_2 Z_3\}. \tag{6.14}$$

The expressions at the right hand side are standard notation employed in the field of quantum error correction, indices correspond to the physical qubit number. The error set $\{I, X_1, X_2, X_3\}$ forms a correctable set of errors considering this stabilizers. The measurement of $Z_1 Z_2$ can be interpreted as a comparison of the values of the first and second qubit, while $Z_2 Z_3$ compares the second and third qubit. Same values do lead to a result of +1, while different values result in -1. An X_1 error would transform the stabilizers to $\{-Z_1 Z_2, Z_2 Z_3\}$, thus resulting in an error syndrome measurement of -1 of $Z_1 Z_2$ and +1 on $Z_2 Z_3$. When both syndrome measurements result in -1, it is assumed that an error on the second qubit occured, since this has higher probability than a combined error on the first and third. In the error-free case, both syndrome measurements result in +1. Table 6.3 shows all syndrome measurement result combinations, the corresponding assumed error type and the action. For correction, the inverse operation of the assumed error is applied. An important point is that the error syndrome measurements do only compare the value of the qubits with respect to each other, but do not give any information about amplitudes α and β of the encoded quantum state $|\psi\rangle$.

$g_1 = Z_1Z_2$	$g_2 = Z_2Z_3$	Error type	Action
+1	+1	None	None
+1	-1	X_3	σ_x on 3. qubit
-1	+1	X_1	σ_x on 1. qubit
-1	-1	X_2	σ_x on 2. qubit

Table 6.3: Syndrome measurement results and the corresponding error type and action to correct for the error.

6.4 The Topological [[7,1,3]] Code

Many quantum error correction codes can be classified as $[[n, k, d]]$ stabilizer codes [102], where n is the number of physical qubits, k the number of logical qubits and d the code distance between the basis states. The stabilizer formalism can be used to describe quantum error correction codes, by defining a subspace of the N-qubit Hilbert space. Defining the stabilizers effectively reduces the Hilbert space of the multi-qubit system of physical qubits to a single logical qubit. The Hilbert space of an N-qubit system has a dimensionality of 2^N. If a set of k stabilizers is defined over the system, the dimension of the Hilbert space is reduced to 2^{N-k}, thus defining a subspace of $N-k$ logical qubits. The following example by [35] illustrates this. Let's picture a system of two qubits. The Hilbert space features a dimension of four. If the assumption is added that the qubit state is stabilized by X_1X_2, the orthogonal basis states of the system can be described as

$$|0\rangle_L = \frac{1}{\sqrt{2}}(|01\rangle + |10\rangle), \tag{6.15}$$

$$|1\rangle_L = \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle). \tag{6.16}$$

This reduces the Hilbert space to two dimensions, defining the logical qubit on a subset of the original multi-qubit Hilbert space.
In this work I present the experimental realization of a weight-4 fault-tolerant (FT) parity check measurement (PCM) scheme using a shuttling-based trapped-ion quantum processor (see Sec. 7.5). This scheme is a key building block of the smallest distance-3 topological color code, being equivalent to the $[[n = 7, k = 1, d = 3]]$ Steane code [29, 105]. The FT PCM scheme is not only an important block of this specific code, it does also represent a building block for FT circuit constructions in 2D topological quantum error correction (QEC) codes of larger size [40, 44, 106].
The $[[7, 1, 3]]$ code uses $n = 7$ physical qubits, to encode $k = 1$ logical qubits with a code distance of $d = 3$. Therefore, it is capable of correcting $t = \lfloor (d-1)/2 \rfloor = 1$ arbitrary errors on one of the physical qubits. The 7-qubit color code, illustrated in

Fig. 6.2, can be defined by the six stabilizers operating on three plaquettes, each comprised of four physical qubits [107]:

$$\begin{aligned} S_z^{(1)} &= Z_1 Z_2 Z_3 Z_4, \quad S_x^{(1)} = X_1 X_2 X_3 X_4, \\ S_z^{(2)} &= Z_1 Z_2 Z_6 Z_7, \quad S_x^{(2)} = X_1 X_2 X_6 X_7, \\ S_z^{(3)} &= Z_1 Z_4 Z_5 Z_6, \quad S_x^{(3)} = X_1 X_4 X_5 X_6, \end{aligned} \tag{6.17}$$

where the indices correspond to the data qubit number accordingly. Analogous to the three qubit bit flip code example in Sec. 6.3, the combined information about the error syndrome readout of the stabilizers do allow to identify on which qubit an error occured with highest probability. This is possible due to the overlap of the plaquettes on certain qubits. An error syndrome measurement result of -1 on only the green plaquette of Fig. 6.2 points to an error on data qubit d_3, assuming that only a single error occured, which has a higher probability than a combination of two errors. An error on d_1, d_2 or d_4 would also have influenced the error syndrome readout of the other plaquettes.

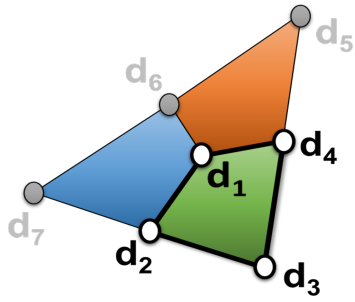

Figure 6.2: Sketch of the [[7,1,3]] color code, showing all seven data qubits and how they are arranged in three plaquettes of different color. The FT PCM scheme operates on one arbitrarily chosen plaquette of data qubit d_1 to d_4, highlighted here in green.

The code can be classified as a Calderbank-Shor-Steane (CSS) code, since the generators of the set of stabilizers do only consist of X and Z operators [29, 30].

6.5 Fault-Tolerant Quantum Error Correction

The parity check measurement (PCM) allows to detect errors on the encoded qubit. The information is mapped to an auxiliary qubit by performing entangling gates between the data qubits and the auxiliary qubit. Projective measurements on this auxiliary qubit reveal possible errors and allow to correct for them subsequently. However, these entangling gates exhibit finite error rates. Therefore, the schemes to detect and correct for errors can inject errors by themselves. Fault-tolerant quantum

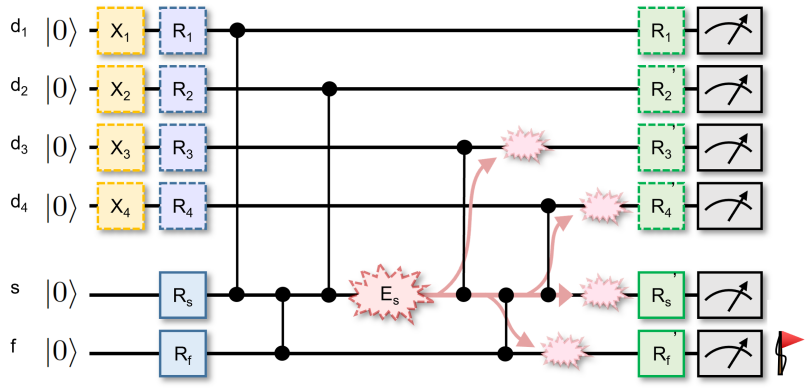

Figure 6.3: Quantum circuit of the fault-tolerant parity check measurement (FT PCM) on a four data qubit plaquette. Single-qubit rotations are performed for initialization (yellow, blue) and analysis (green) purposes, prior to the projective readout (grey). The data qubits are entangled with the syndrome via ZZ-gates. Two additional gates between the syndrome and flag qubit serve to raise the flag in case of dangerous weight-2 errors on the syndrome.

error correction schemes address this issue [36], by detecting *hook errors*, which are errors on the syndrome that would proliferate onto the data register, as shown in Fig. 6.5. Additional auxiliary qubits are used, ensuring that errors injected throughout the PCM can be detected and possibly handled by the scheme. The circuit in Fig. 6.3 is adapted to the native gate set of trapped-ion quantum computers using the translation shown in Fig. 6.4 [108]. While in the standard gate set usually the CNOT operation is used as a two-qubit entangling operation, the ZZ-gate is implemented on our trapped-ion system to create entanglement.

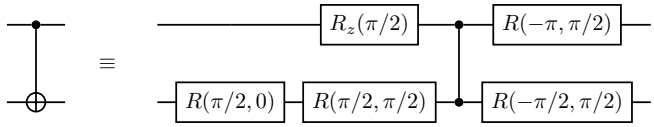

Figure 6.4: Correspondence between CNOT and ZZ entangling gate used in the trapped-ion system, as used to translate the FT PCM circuit to the native trapped-ion gates [108].

The first suggestions for FT PCM schemes required as many auxiliary qubits as the parity check generator with maximum weight [37, 38], which results in a large resource overhead to ensure fault-tolerance. More recently developed FT PCM schemes are based on so-called flag qubits and reduce the overhead in terms of additional qubits as well as gate operations [39–44, 44–46]. It has been shown that only two auxiliary qubits

can ensure fault-tolerant operation of the PCM, which leads to the guarantee that any single error per QEC cycle can be corrected [40]. Several QEC protocols and building blocks have been demonstrated to date on multiple platforms, including trapped ions [109–115], superconducting circuits [116–119], nuclear magnetic resonance [120, 121] and nitrogen-vacancy centers [47, 122, 123]. Elements of fault-tolerant QEC, such as FT state preparation and detection, have been realized on platforms such as superconducting circuits [34, 116, 124] and trapped ions [125]. The preparation and operation of an encoded qubit on a distance-3 Bacon-Shor code, based on nine data qubits and four auxiliary qubits, in a fault-tolerant manner, was recently implemented [126]. Very recently, a fault-tolerant encoding of a single logical qubit using the $[[7, 1, 3]]$ color code was presented on a trapped-ion quantum processor [48].
In this thesis, a flag-based weight-4 FT PCM scheme is realized on a shuttling-based trapped-ion quantum processor [59]. This scheme reduces the overhead for the fault-tolerant syndrome readout to only two auxiliary qubits. One *syndrome* qubit, to detect the parity of the data qubits, as well as the *flag* qubit, used to detect faults which would proliferate onto two out of the four data qubits of the plaquette. Fig. 6.3 shows the FT PCM circuit to perform the parity check measurement of a four-qubit plaquette using one syndrome and one flag auxiliary qubit.

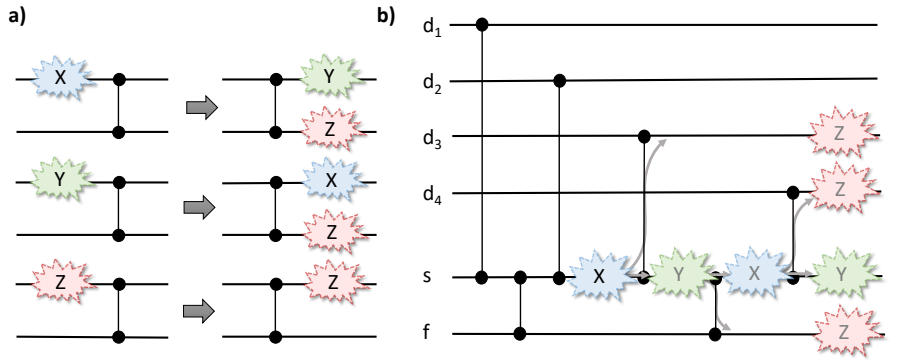

Figure 6.5: Error propagation through ZZ-gates. **a)** Propagation of X-, Y- and Z-type errors. **b)** Propagation of an X-type error on the syndrome between gates d_2s and d_3s. The error is propagating onto the data qubits d_2 and d_3, as well as the flag qubit.

Chapter 7

Flag-Based Fault-Tolerant Parity Readout

Flag-based fault-tolerant parity check measurements (FT PCM) are of high importance as a quantum error correction building block (see Sec. 6.4 and 6.5). The following chapter presents one of the first experimental realizations of a flag-based fault-tolerant parity check measurement on a shuttling-based trapped-ion quantum processing node. Sec. 7.1 shows the general quantum circuit and shuttling sequences for the time-dependent reconfiguration operations, Sec. 7.2 presents timings of sequence blocks such as cooling, state preparation, execution of the parity mapping and readout. Sec. 7.4 discusses entangling gates and syndrome contrast throughout the FT PCM sequence. Sec. 7.5 shows the experimental results of parity check measurements and the improvements due to post-selection on the flag qubit state. The measurement results concerning error catch rates of injected coherent error rotations are shown in Sec. 7.6.

7.1 Quantum Circuit and Shuttling Sequence

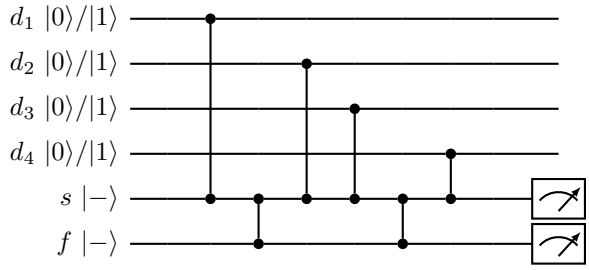

Figure 7.1: Flag-based fault-tolerant parity check measurement quantum circuit.

The flag-based fault-tolerant parity readout circuit for $g_z = Z_1Z_2Z_3Z_4$, shown in Fig. 7.1, consists of four data qubits and two auxiliary qubits. Upon entanglement with the data qubits, the syndrome acquires a phase depending on the data qubit state.

The four entangling gates lead to a total phase of 0 or π on the syndrome, depending on even or odd parity of the four data qubits. This phase difference can be detected on the syndrome and makes it possible to gain insight on the combined data qubit state in a quantum non-demolition way. The flag qubit serves to catch errors within the PCM at positions, where it would lead to a weight-2 error, propagating onto two data qubits. Realization of this circuit on our shuttling-based hardware is presented in the figures 7.2, 7.3 and 7.4, showing the preparation of the ion states, the actual FT PCM gate sequence and the spin-selective readout. This circuit is implemented based on a register of 3×2 ions and the register reconfiguration operations discussed in Sec. 4. Small subsets of up to two ions feature a high degree of control and a low number of motional modes compared to all ions in a single crystal. Cross-talk of gate operations on stored ions is highly reduced due to the large distance in the range of several hundreds of µm up to several mm in contrast to laser beam diameters around 30 µm. Translation of the circuit into a shuttling sequence is performed with a focus on minimizing the total number of reconfiguration operations. I had to develop and optimize the shuttling sequence for the FT PCM circuit manually, while within the time of writing this work a quantum circuit compiler was established by Christian Melzer, Janis Wagner and Fabian Kreppel. It allows to automatically translate a quantum circuit into a circuit using the native gate set of our hardware setup. This circuit is then compiled for our linear segmented trap architecture, resulting in an optimized sequence of shuttling and gate operations. The 3×2 ions are trapped in positions LIZ (segment 19), LIZ+4 (segment 23) and LIZ+7 (segment 26). After successful loading, ions are sequentially distributed over the segments 19, 22 and 25. Starting in LIZ+4 ions are moved sequentially to the LIZ for state preparation, portrayed in Fig. 7.2 or optional high detuned recrystallize cooling. While this optional first cooling stage was helpful to cool highly excited ions before the shuttling operations were improved, it is no longer necessary after the improvements described in Chapter 5 were made, reducing the sequence time significantly. Ions are sequentially Doppler cooled in the order $\{d_3, d_4\}$, $\{s, f\}$ and $\{d_2, d_1\}$, driving the cycling transition around 397 nm for a time of 2 ms. These Doppler cooled ions then undergo resolved sideband cooling in reverse crystal order, driving stimulated Raman transitions on the red sidebands (rsb) of the transverse motional modes. Only the transverse modes are ground state cooled, since these are involved in driving the entangling gate. Each rsb cooling pulse is performed to be an approximate π-pulse reducing the phonon number by one, while transferring from $|0\rangle$ to $|1\rangle$. To maintain an efficient cooling cycle, repumping to $|0\rangle$ is necessary before the next rsb pulse. This is achieved via optical pumping using the σ_+-polarized sigma laser beam near 397 nm for a pulse duration of 1 µs. The cooling sequence covers all 2nd and 1st red sidebands of the transverse modes, in-phase and out-of phase motion of the two ions. Cooling starts on the 2nd rsb and continues with the corresponding 1st rsb, in multiple iterations. After cooling all ion crystals, a shortened cooling sequence is executed only on the in-phase modes, which acquire a small amount of excitation due to anomalous heating in the waiting time[1]. Using this stack of cooling stages, a low motional excitation close to

[1] A summary and review of electric-field noise measurements in ion traps and a discussion of noise sources is given by Blatt [127].

the ground state is achieved for all ions within the register. The cooling is followed by an individual state initialization scheme further described in Sec. 7.3 and finalizing Fig. 7.2. Based on the prepared register, shuttling sequences dedicated to the actual mapping of parity information to the syndrome qubit is carried out, as shown in Fig. 7.3. First syndrome and flag qubits are transferred to their appropriate initial state for the measurement by means of a single-qubit rotation. The qubit register is sequentially rearranged in order to bring qubits required for their intended two-qubit gates into a common potential well and move them to the laser interaction zone. These operations make use of the previously described essential shuttling building blocks such as transport, separation/merge and positional ion swap. The sequence ends with the application of the readout mapping pulses to the syndrome and flag qubit individually. The following sequence part for multi-qubit spin-selective detection and verification shown in Fig. 7.4 is further described in the following Sec. 7.3.

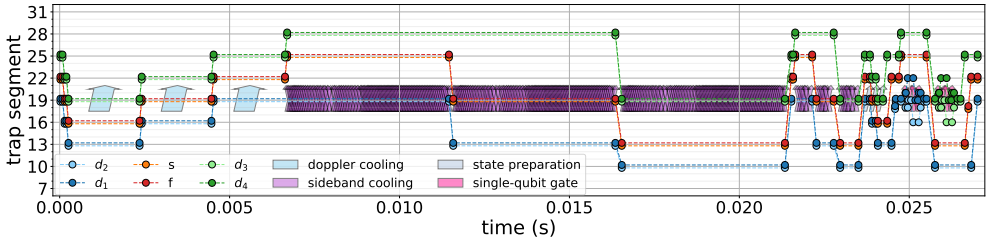

Figure 7.2: Shuttling sequence for state preparation. First, the three two-ion crystals are sequentially Doppler cooled, followed by sequential resolved sideband cooling. Afterwards, all three two-ion crystals are initialized to the default spin state $|0\rangle$. The sequence part ends with individual single-qubit rotations on all data qubits, in order to initialize the computational basis state.

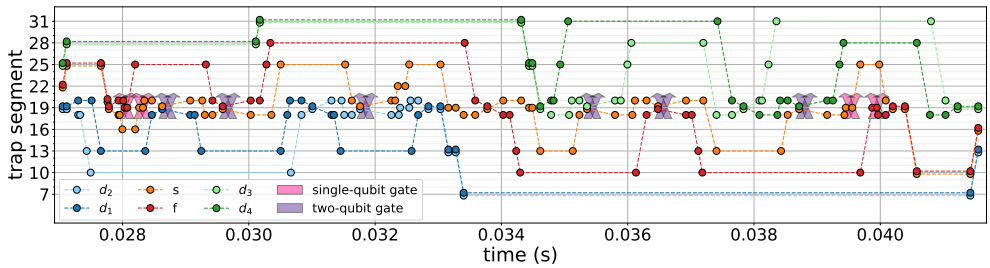

Figure 7.3: Shuttling sequence for flag-based fault-tolerant parity readout. First, syndrome and flag are initialized to e.g. $|-\rangle$, followed by the reconfiguration operations and two-qubit entangling gates to map the parity of data qubits onto the syndrome qubit in a fault-tolerant manner. Finally, single-qubit analysis rotations on the syndrome and flag are carried out to readout the state of the auxiliary qubits in different basis.

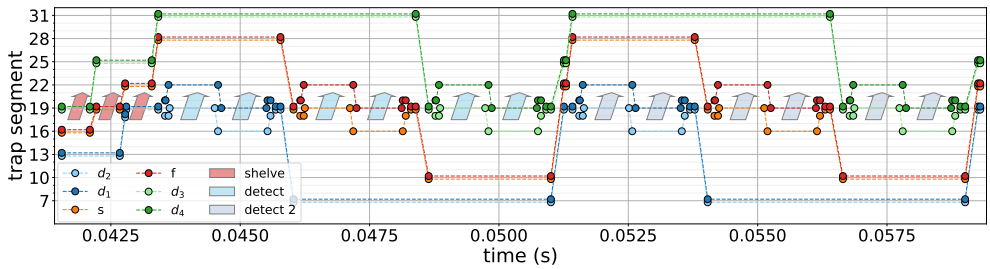

Figure 7.4: Shuttling sequence for spin-selective multi-qubit detection and verification. All three two-ion crystals are sequentially shelved, prior to any detection light in the trap. Then the detection is carried out on each individual qubit, using light near 397 nm and 866 nm. The second detection on all qubits also includes the quenching laser near 854 nm, which ensures that bright ions are detected, independent of their prior qubit state. The second detection is utilized to identify ion losses and discard invalid shots.

7.2 Sequence Timing

A detailed time budget of the combined shuttling sequence consisting of state preparation, FT PCM register reconfiguration including all six entangling gates and spin-selective multi-qubit detection and verification part can be found in Fig. 7.5. It shows clearly, that a lot of time is dedicated to the register preparation and qubit register reconfiguration operations such as transport, separation/merge and swap. With a duration of 28 ms, preparation of the qubit register consumes nearly half of the total time, mostly due to time intense sequential resolved sideband cooling on all transverse modes. In the sequence part dedicated to map the data qubit register parity to the syndrome, very little time is spend on the laser-driven single- and two-qubit gates, only around 1 ms, while about 12.6 ms are needed for register reconfiguration operations. Especially here, but also all over the sequence, the total amount of time needed could be significantly reduced by reducing the shuttling overhead. More details about ongoing improvements in this direction are described in Chapter 9.

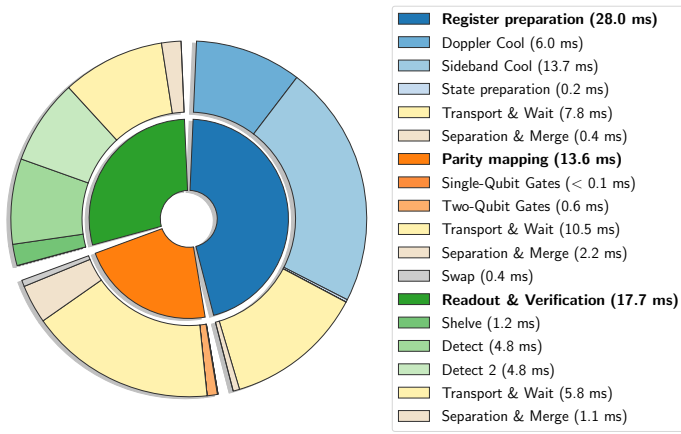

Figure 7.5: Timing of the flag-based fault-tolerant readout measurement sequence.

7.3 State Init and Readout

The state preparation shown in Fig. 7.2 initializes all ions to state $|0\rangle$ using optical pumping in two stages. The optical pumping is performed on each two-ion crystal, consisting of exposure to a σ_+-polarized beam near 397 nm for 1 µs to deplete the state $|1\rangle$ and a second frequency selective pumping state using four cycles of optical pumping on the dipole-forbidden $|S_{1/2}, m_J = -1/2\rangle \leftrightarrow |D_{5/2}, m_J = +1/2\rangle$ transition near 729 nm. The pulses have a duration of 10 µs and power is chosen to realize a π-pulse accordingly. Each of the four 729 nm π-pulses is followed by a 4 µs quenching pulse near 854 nm, depleting the $D_{5/2}$ state with its comparably long lifetime of about 1.17 s, to maintain the pumping cycle. This optical pumping scheme typically initializes the qubits to state $|0\rangle$ with an infidelity of less than 0.1%. This scheme is followed by individual qubit initialization, using single-qubit rotations on the previously prepared state $|0\rangle$. To prepare different logical input states of the data qubits, optional rotations of the form $R(\pi, -\pi/2)$ are performed on the data qubits, while rotations $R(\pi/2, -\pi/2)$ and $R(3\pi/2, -\pi/2)$ are used to initialize to states $|-\rangle$ and $|+\rangle$, respectively. A successful verification of the spin-flip functionality of each individual ion and proper assignment of the detected qubit states is shown in Fig. 7.6.

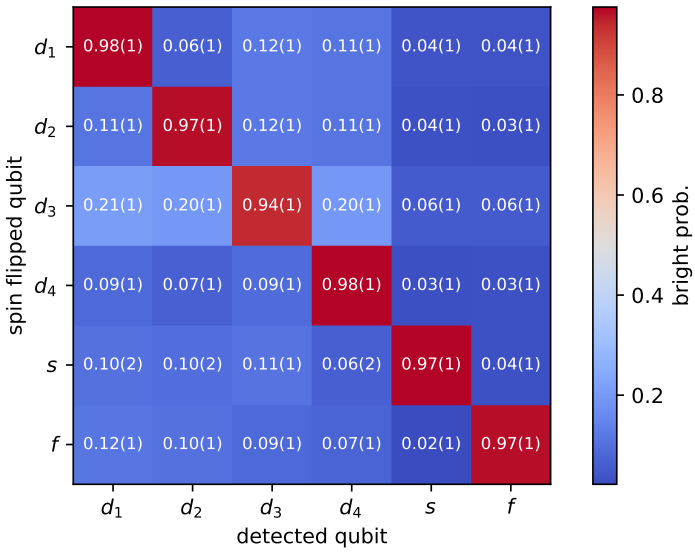

Figure 7.6: Matrix showing initialization and spin flip of each involved ion. The correlation between the detected and spin-flipped qubit indicates a correct execution of the shuttling operations. By improving the reconfiguration operations, SPAM errors could be further reduced, as shown in the final state preparation and detection fidelities in Table 7.1.

Spin-selective readout is performed by selectively transferring qubits to the metastable

state, causing them to be inaccessible for the cycling transition used to detect a bright ion. This transfer, also referred to as shelving, is performed using rapid adiabatic passage pulses on the transitions $|S_{1/2}, m_J = +1/2\rangle \leftrightarrow |D_{5/2}, m_J = +1/2\rangle$ and $|S_{1/2}, m_J = +1/2\rangle \leftrightarrow |D_{5/2}, m_J = -3/2\rangle$. Chirped Gaussian-shaped laser pulses are used, using a variation of the optical frequency by ±60 kHz around the resonance and a duration of 200 µs. The pulse power is chosen in order to maximize shelving of $|0\rangle$ and minimize unintentional cross-shelving of $|1\rangle$. Electron shelving is carried out pairwise on all ions prior to any detection, in the order $\{d_3, d_4\}$, $\{s, f\}$ and $\{d_2, d_1\}$. After shelving, ion pairs are moved into the LIZ again in reverse order. The pairs are separated and ions are moved individually into the LIZ. Scattered photons are collected via a PMT while applying 397 nm detection light at about onefold saturation power and 866 nm repumping light for 800 µs. This allows for threshold based discrimination of the qubit states $|0\rangle$ and $|1\rangle$. To verify the validity of this detection, especially distinguishing of a dark state detection from ion loss events, a second round of this detection shuttling scheme is carried out, but including light near 854 nm, depleting the metastable state and leading to bright detection independent of prior qubit state. Only sequence runs with a valid so-called postselection measurement on all participating ions is taken into account for data acquisition. Currently, a valid measurement cycle ratio of around 83% is obtained. The multi-qubit detection sequence is finished by moving the ion pairs back to their trapping positions, preparing to start either the next shot or a retrapping sequence in case of ion losses.

State preparation and measurement (SPAM) errors are of crucial importance when it comes to QEC schemes, since these do rely on the single-shot readout fidelity of auxiliary qubits in order to condition optional feedback operations. A high fidelity state preparation featuring an infidelity $< 0.1\%$ is typically achieved via the two-stage optical pumping and does therefore not contribute to the SPAM error significantly in the case of larger shuttling sequences. Without shuttling, a combined preparation and readout infidelity of $9(4) \times 10^{-4}$ for state $|0\rangle$ and $5(3) \times 10^{-4}$ for state $|1\rangle$ is achieved. The actual bottleneck within sequences, including around 100 register reconfiguration steps, can be identified as the electron shelving required to perform state-selective readout of the spin qubit. The laser at 729 nm is aligned perpendicular to the trap x-axis, along which transports are carried out. Therefore, it should not couple to the axial modes of motion, which acquire a significant amount of excitation at the end of the FT PCM scheme. The dependence of the readout fidelity on the axial excitation does lead to the assumption that the beam is not perfectly aligned. A presumed residual coupling to axial modes leads to a considerably high degradation in the readout fidelity. Table 7.1 shows the combined state preparation and measurement fidelities for the prepared states $|0\rangle$ and $|1\rangle$ within the two measurement schemes used in the following.

For the FT PCM shuttling sequence, data qubits are initialized in their final state prior to the actual FT PCM shuttling block, in order to not interfere with the actual QEC building block. In the GME sequence, it was chosen to initialize all ions to $|0\rangle$ in the state preparation sequence part and then do the last single-qubit rotation and final readout rotation as close to the two-qubit gates as possible. This was done to keep the coherent sequence part for each ion short. After this early readout pulse, the

FT PCM Sequence	$'bright' \equiv \|1\rangle$	$'dark' \equiv \|0\rangle$
s	98.8(2)%	98.6(3)%
f	98.8(2)%	98.1(3)%
GME Sequence	$'bright' \equiv \|1\rangle$	$'dark' \equiv \|0\rangle$
d_1	99.7(1)%	98.2(4)%
d_2	99.6(1)%	98.7(3)%
d_3	99.2(2)%	99.0(3)%
d_4	98.6(2)%	98.6(3)%
s	99.6(1)%	97.1(5)%
f	99.6(1)%	98.4(3)%

Table 7.1: State preparation and detection fidelity including the FT PCM reconfiguration operation sequence. For the data pertaining to the GME, data qubit init, readout and shelving pulses are performed on the corresponding ions directly before and after the two-qubit gates.

shelving sequence is applied immediately. Despite there being several tens of ms in between the first shelving and detection, infidelity due to the lifetime of around 1.17 s of the metastable state outperformed the fidelity loss of shelving the data qubits at the very end of the register reconfiguration sequence. A huge drift of readout fidelity vs. time on the data qubits has been observed, especially ions d_1 and d_2, spending most time stored in the segments $\ll 19$ (LIZ). This could be caused by a change of the motional heating rate on this segment over the course of several hours of data acquisition. Using early shelving on the data qubits, readout infidelities are stable ranging from 0.3(1)% to 2.9(5)%, depending on the qubit and state.

7.4 Gates and Coherence

Quantum error correction requires repeated execution of syndrome extraction and correction. To allow for many repetitions, it is of significant importance to perform high fidelity gates and maintain coherence throughout all executed sub-circuits. Previous to this thesis, entanglement of up to four qubits was realized in our shuttling-based trapped-ion setup [70]. These qubits were all nearest neighbors within the linear ordering, therefore four-qubit entanglement did not require swap operations. The FT PCM scheme now includes six qubits and six entangling gates $d_1s, sf, d_2s, d_3s, sf, d_4s$, where reordering of the linear configuration of the ions is necessary between each of the two-qubit gates performed in order to commonly confine the specific data or flag and syndrome qubit. Gates are performed as described in Sec. 2.9.2, using the transverse in-phase collective vibrational mode of the commonly confined ions at a frequency of around $2\pi \times 4.64$ MHz and providing in-between rephasing pulses.
Upon first test executions of the sequence, a huge contrast loss was observed when enabling the entangling gates. Further investigation suggested that the swap operation was the cause of loss of contrast. Operation of light-shift gates in the presence of non-negligible excitation of either the gate mode or any other radial spectator mode represents a significant error source, as investigated in detail by C. Ballance [128]. A measurement of the syndrome contrast at the gate execution positions with disabled entangling gates shows a much higher contrast and thus supports this assumption. Residual light leakage of the recrystallize cooling beam also represents a source of decoherence, but a full blocking of this beam during the sequence execution does only give a minor improvement on the syndrome contrast, as can be seen in Fig. 7.7.

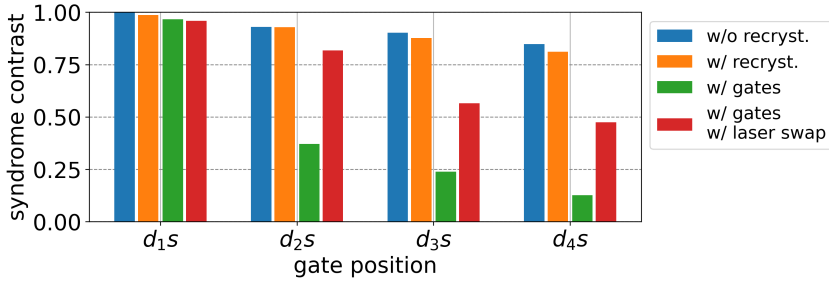

Figure 7.7: Contrast after gates prior to the improvement of the positional ion swap. To investigate the source of contrast loss, multiple measurement configurations are carried out. The syndrome contrast at the gate positions is compared without entangling gates (blue, orange) and with gates including positional ion swap (green) as well as a laser-driven swap operation (red).

Even though a laser-driven SWAP operation is not suitable for the FT PCM scheme, since it would contradict the fault-tolerancy requirement, it can be used to temporarily circumvent the execution of potentially sub-optimal positional ion swaps to swap qubit states. A laser-driven SWAP operation circuit and an operational test on

the computational basis states can be found in Appendix Fig. B.2 and B.3. The laser-driven SWAP itself creates a huge overhead in gate operations, three additional entangling gates were used to replace each positional ion swap operation. In total, nine additional ZZ-gates are needed for the d_4s contrast measurement with gates and laser-driven SWAP operation. Nevertheless, this scheme considerably outperformed the syndrome contrast after entangling gates compared with positional ion swaps. Overall, these findings pointed towards undesired energy transfer from the axial to radial mode of motion during the positional ion swap operation. Axial excitation is caused by the transport and separation/merge operations within the FT PCM sequence prior to the swap operations. More measurements probing the motional state of a two-ion crystal after swap operation supported the assumption that the positional ion swap transferred axial to radial excitation (see Fig. A.8). The radial motional excitation contributes to a large error when performing the light-shift gate. Therefore the positional ion swap operation desperately required improvements to realize the FT PCM scheme and any comparable or even larger shuttling-based quantum circuits. Fortunately, the swap operation has been improved significantly by optimizing time-dependent voltage ramps to keep axial and radial trap frequencies more constant and well separated throughout the swap operation, described in more detail in Sec. 5.4. Using the improved positional ion swap an effectively error-free swap operation, required for fault-tolerancy, was established and contrast of the syndrome did not significantly decrease when executing the circuit and entangling gate operations, as shown in Fig. 7.8.

Residual contrast loss on the observed parity fidelity of 93.2(2)% is consistent with the contrast loss on the syndrome without any two-qubit gate, which is 91(4)%. This contrast loss and therefore the dominant error source can be attributed to qubit dephasing due to ambient magnetic field fluctuations. A future hardware development is going to address this issue by using an improved magnetic field shielding.

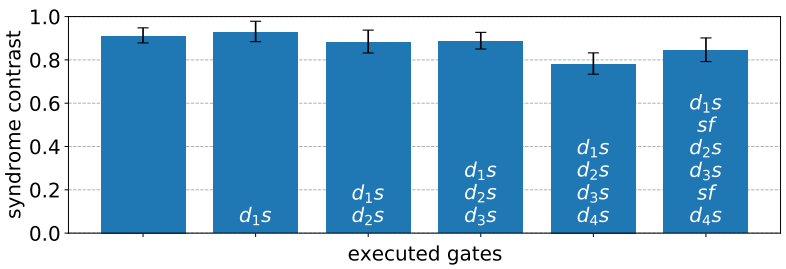

Figure 7.8: Readout of the syndrome contrast at the position of the last gate d_4s, versus the amount of executed gates. The contrast estimation is based on 160 shots per Pauli X, Y operator.

7.5 Fault-Tolerant Parity Check Measurement

The circuit for flag-based fault-tolerant weight-4 parity check, $g_z = Z_1Z_2Z_3Z_4$, shown in Fig. 7.9, is including all applied gates, starting from the default initialization $|0\rangle$ on all qubits. The parity check g_x is an equivalent, since it only requires local rotations of the X-basis on data qubits prior to the syndrome readout. The circuit starts with an optional single-qubit rotation on each data qubit of the form $R(\pi, -\pi/2)$ (yellow boxes), enabling the sequential preparation of all 16 computational input states using the same circuit. Syndrome and flag qubit are independently initialized in state $|-\rangle = \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle)$ by means of a rotation $R(\pi, -\pi/2)$. All following two-qubit gates follow equation 2.36, featuring the rephasing π-pulse at half the gates phase pickup, maintaining coherence on the participating qubits. This rephasing especially improves coherence on the syndrome qubit, since it undergoes six rephasing pulses in total. Data qubits are read out in the Z-basis, $M^{(Z)}$, and therefore do no require any single-qubit rotations prior to the shelving and detection, while syndrome and flag are read out in the rotated basis $M^{(X)}$, which is realized by an analysis rotation $R(\pi/2, -\pi/2)$ before the detection. An additional phase $\phi_{s/f}$ included in this final rotation serves the compensation of shuttling induced phase as described in Sec. 4.5. This phase is calibrated by execution of the very same sequence, only removing the entangling phase gates, but not the rephasing pulses. The qubit used to calibrate the phase is initialized in $|-\rangle$ and readout in X- and Y-basis in order to reconstruct the contrast and phase according to equation 4.13. Experimental measurement results of the syndrome $M_s^{(X)}$ and $M_f^{(X)}$ are presented in Fig. 7.10. The probability $p(M_s^{(X)} = -1)$ of finding the syndrome in the 'bright' state $|1\rangle$ is shown vs. the prepared computational state of the data qubit register. For the flag qubit the probability $p(M_f^{(X)} = -1)$ of detecting state $|1\rangle$ is shown in red, averaged over all data register input states.

Probability p_{+1} of measuring 'bright' $M_s^{(X)} = -1$ for a data qubit register prepared to the input parity $P_{in} = +1$ can be defined as

$$p_{+1} = p(M_s = -1 \mid P_{in} = +1) \tag{7.1}$$

and similarly the probability p_{-1} of measuring 'dark' $M_s^{(X)} = +1$ for a data qubit register state featuring $P_{in} = -1$

$$\begin{aligned} p_{-1} &= p(M_s = +1 \mid P_{in} = -1), \\ &= 1 - p(M_s = -1 \mid P_{in} = -1). \end{aligned} \tag{7.2}$$

Each logical input state requires a new shot, since it requires preparation of the data qubit register accordingly. $N = 960$ independent shots were performed for every logical input state, leading to a total number of shots of $N_{tot} = 16 \times 960 = 15360$. Therefore, the combined probabilities for even and odd parity are based on half of this logical input states $N_p = 8 \times 960$. Errors are calculated using

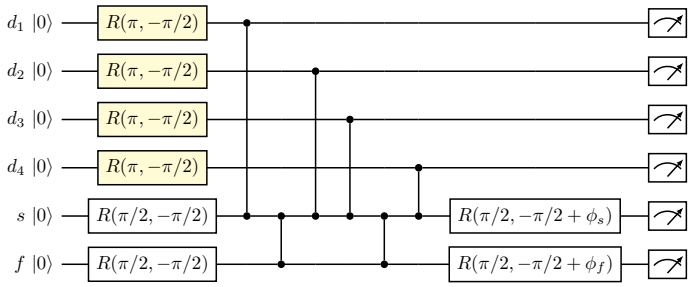

Figure 7.9: Quantum circuit of the FT PCM, featuring default state preparation to $|0\rangle$, followed by $\pi/2$-pulses on syndrome and flag to initialize to state $|-\rangle$. This circuit is based on Fig. 7.1, but showing all single-qubit rotations in detail. The data qubits are optionally spin-flipped by application of a π-pulse. Readout on syndrome and flag is carried out by another $\pi/2$-pulse. Mandatory gate boxes are presented in white, while optional ones are yellow. The phases ϕ_s and ϕ_f correct for shuttling-induced phase offsets.

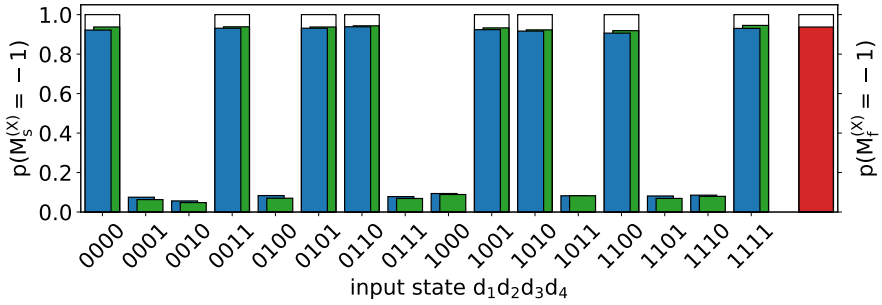

Figure 7.10: Parity syndrome readout with flag-selection. The probability of measuring the syndrome in $M_s^{(X)} = -1$ state is shown for each computational basis state of the data qubit register. Once for all valid shots (blue) and for the shots post-selected on the flag qubit measurement result (green), versus the ideal rate (white). 960 shots per input state are measured, leading to an average shot noise of about 7×10^{-3}. Probability to measure the flag in the non-error indicating state $M_f^{(X)} = -1$ is shown separate and averaged over all computational basis states (red).

$$\begin{aligned} \Delta p_{+1} &= \sqrt{p_{+1}(1-p_{+1})/(N_p-1)} \\ \Delta p_{-1} &= \sqrt{p_{-1}(1-p_{-1})/(N_p-1)}. \end{aligned} \tag{7.3}$$

The parity measurement fidelity (PMF) shall be defined as the probability of measuring the correct syndrome state $M_s^{(X)}$ with respect to the initially prepared state parity P_{in} of the data register

$$\mathcal{P} = \frac{1}{2}(p_{+1} + p_{-1}). \tag{7.4}$$

Since p_{+1} and p_{-1} are based on independent measurement shots, the error of the parity measurement fidelity can be calculated using Gaussian error propagation according to

$$\Delta\mathcal{P} = \sqrt{\left(\frac{1}{2}\Delta p_{-1}\right)^2 + \left(\frac{1}{2}\Delta p_{+1}\right)^2}. \tag{7.5}$$

The measurement shown in Fig. 7.10 results in a parity fidelity of $P = 92.3(2)\%$. The flag qubit is detected to be $M_f^{(X)} = -1$ in $93.7(2)\%$ of the shots, showing a low rate of weight-2 errors. The data depicted in green bars is based on the subset of the 960 shots each, conditioned on the single-shot flag result detected as $M_f^{(X)} = -1$, which corresponds to no error. The flag-selected parity fidelity is found to be $P = 93.2(2)\%$, which exceeds the bare parity fidelity by 4.5 standard errors. This indicates that the FT PCM scheme succeeds in catching native errors occurring throughout the FT PCM sequence.

7.6 Error Injection and Flag Verification

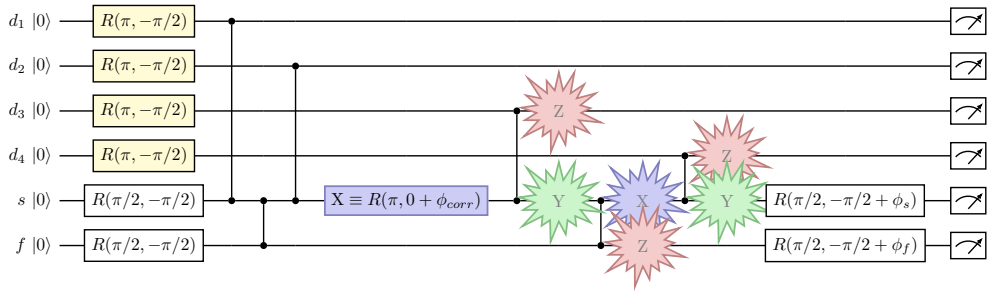

Figure 7.11: Quantum circuit of the FT PCM scheme as in Sec. 7.5, including an injected X-type error (blue rectangle) between the gates $d_2 s$ and $d_3 s$. Star-shaped boxes indicate the propagation of this error throughout the circuit.

To verify the capability of the flag qubit to reliably detect critical weight-2 errors occuring during the FT PCM scheme, Pauli-X and Pauli-Y errors are injected on the syndrome qubit in between gates $d_2 s$ and $d_3 s$. These errors can be integrated, with equivalency up to a global phase, by application of a single-qubit rotation on the syndrome qubit of the form $R_s(\pi, 0)$ and $R_s(\pi, \pi/2)$ for X_s and Y_s, respectively. Fig. 7.11 shows the modified circuit including the injected X-error and the error propagation according to Fig. 6.5. A Pauli-X error on one qubit prior to the ZZ-gate is going to propagate onto both qubits as a Y-type error on the qubit previously faulty and as a Z-type error on the second qubit. Similarly a Pauli-Y error will be transformed into an X-type error on the same qubit, while the second also gains a Z-type error. Only the Z-type error is going to commute with the ZZ-gate and does not affect the second qubit. The measurement scheme is very close to the one described in Sec. 7.5, with optional π-pulses on the data qubits to initialize all 16 computational input states and the preparation of syndrome and flag in $|-\rangle$ and readout in $M^{(X)}$ including the compensation of the shuttling induced phase. It shall be noted that here the phase calibration has to take into account additional time for the single-qubit rotation representing the error on the syndrome. The measurement result of $N = 140$ shots per input state is shown in Fig. 7.12. The injected Y_s in Fig. 7.12 **a)** represents the occurrence of a combined bit and phase flip error and thus manifests in not only a Z-errors on d_3, d_4 and f, but also an X-type error on the final syndrome parity readout. While the final X_s error escapes the detection on the syndrome parity itself, it nevertheless proliferates onto data qubits of the following two-qubit gates and leads to a weight-2 error on the encoded state. Even in the presence of the X-type error, syndrome parity readout still shows a high agreement with the input parity $P = 88.3(7)\%$. Importantly, this error can be caught upon flag detection $M_f^{(X)} = +1$ with an error catch rate of 90.6(6)%. The injected X_s error in Fig. 7.12 **b)** does lead to a final Y_s error on the syndrome, inverting measurement results compared to the

expectation without error and leading to $P = 14.7(7)\%$. In the injected X_s error case, the flag qubit also reliably detects the error with a catch rate of 89.8(6)%, showing that the flag serves its use of detecting weight-2 Z-errors on the data qubit register and therefore is able to preserve the fault-tolerance of this parity readout scheme.

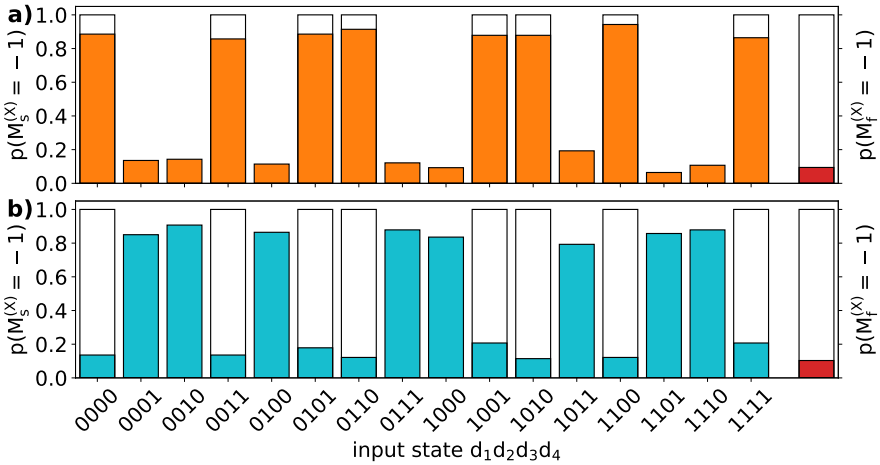

Figure 7.12: Fault-tolerant parity check measurement including an injected error. The probability to measure the syndrome qubit in the $M_s^{(X)} = -1$ state is shown versus the computational basis state of the input data qubit register. 140 shots per input state are acquired, leading to an average shot-noise error of about 2.5×10^{-2}. **a)** portrays an injected Y-type error on the syndrome, which does not affect the syndrome readout (orange), while **b)** pertains to an injected X-type error on the syndrome, inverting the syndrome measurement (light blue) compared to the expectation (white). In both error cases, the flag qubit is detected mostly in state $M_f^{(X)} = +1$, indicating the error.

Chapter 8

Generation of Genuine Multipartite Entanglement

Successful generation of entanglement between all six qubits participating in the FT PCM scheme can be seen as a benchmarking tool to verify that our entangling gate fidelity is sufficiently high. The measurement of complete state tomography is scaling detrimentally with n and does therefore not qualify as a practical tool to verify the entanglement. The witness operators instead can be used to verify the n-qubit genuine multipartite entanglement (GME) in an efficient manner [108]. This chapter gives a brief introduction on the technique of "entanglement witnessing" in Sec. 8.1, followed by the experimental realization of entanglement and its witnessing. Using these building blocks multiple GMEs are presented starting with four-qubit GME in Sec. 8.2, five-qubit GME in Sec. 8.3 and finally six-qubit GME in 8.4.

8.1 Entanglement Witnessing

Entanglement witnessing can be used as a holistic benchmarking technique for the FT PCM measurement. Despite not being a sufficient condition for the FT PCM, it is a necessity that the entangling gate fidelity is high enough, such that the generation of genuine multipartite entanglement can be witnessed. Usually, an entanglement witness for the state $|\Psi^{(n)}\rangle$ can be constructed as

$$\tilde{W} = l_n \mathbb{1} - |\Psi^{(n)}\rangle \langle\Psi^{(n)}|, \tag{8.1}$$

where l_n is the smallest constant to fulfill $Tr(\rho W) \geq 0$ for all separable states and $Tr(\rho W) \leq 0$ for some entangled states [129, 130]. Measurement of the witness requires a decomposition to locally measurable operators, where the number of local measurements can increase exponentially with the number of qubits n. A reduction of the measurement effort can be reached by using a modified witness. It has been proven that the witness constructed to be

$$W_{|\psi^{(n)}\rangle} = (n-1)\mathbb{1} - \sum_{i=1}^{n} g_i, \tag{8.2}$$

does detect genuine entanglement of n qubits for states close to an n-qubit GHZ state [129, 130]. This type of witness has the advantage, that it does require only up to n different measurement settings, in order to acquire the values of the n stabilizer generators g_i. The ideal output state $|\Psi\rangle_{out}^{(n)}$ can be defined as eigenstates of the stabilizer generators g_i with eigenvalue +1, such that

$$g_i \, |\Psi\rangle_{out}^{(n)} = + \, |\Psi\rangle_{out}^{(n)} . \tag{8.3}$$

For convenience, we use a normalized version of Eq. 8.2 of the form

$$W_n \quad = l_n \mathbb{1} - \tfrac{1}{n} \textstyle\sum_{i=1}^{n} g_i. \tag{8.4}$$

For the ideal state $|\Psi^{(n)}\rangle$, the second term does add up to 1. The condition $\langle W_n \rangle < 0$ is satisfied in case that $\frac{1}{n} \sum_{i=1}^{n} g_i$ exceeds the critical value $l_n = (n-1)/n$. Therefore, the target values for the verification of four-, five- and six-qubit entanglement are $l_4 = 3/4$, $l_5 = 4/5$ and $l_6 = 5/6$, respectively.
The experimental verification of the six-qubit GME, with participation of all qubits, is build up sequentially. First, the four-qubit GME only involving the data qubits is verified, second the five-qubit GME is verified, including the syndrome qubit, followed by integration of the flag, in order to verify the full six-qubit GME. The generator expectation values $\langle g_i \rangle$ are determined by measuring the qubits in the corresponding basis. This is realized by application of appropriate analysis pulses before the $M^{(Z)}$ readout. The expectation value is derived from the probabilities of the detected computational basis state in the Z-basis measurement $M_i^{(Z)}$ of the ith qubit, based on

$$Z_i = P_0 - P_1, \tag{8.5}$$

where $P_0 = \frac{1}{2}(\mathbb{1} + \sigma_z)$ and $P_1 = \frac{1}{2}(\mathbb{1} - \sigma_z)$ are the projectors onto the states $|0\rangle$ and $|1\rangle$, respectively. By default, our detection corresponds to $M_i^{(Z)}$, while logical results in $M_i^{(X)}$ ($M_i^{(Y)}$) can be achieved by performing an analysis rotation $R(\pi/2, -\pi/2)$ ($R(\pi/2, 0)$) on the respective ith qubit before detection. As an example for a weight-2 stabilizer, the expectation value of X_1X_2 can be derived through

$$\langle X_1 X_2 \rangle \quad = \quad p_{00} - p_{01} - p_{10} + p_{11}, \tag{8.6}$$

where p_{ij} is the probability to detect the corresponding computational basis state combination of ion i and j, after $M^{(X)}$ readout on both qubits. This can be generalized to the probability p_{+1} for detecting a state with parity equal to +1, subtracted by the probability p_{-1} for detecting a state with parity -1:

$$\begin{aligned} \langle g_i \rangle \quad &= \quad p_{+1} - p_{-1} \\ &= \quad 2p_{+1} - 1. \end{aligned} \tag{8.7}$$

Since the two probabilities add up to 1, it is possible to perform the substitution $p_{-1} = 1 - p_{+1}$. The basis state measurement configurations for different stabilizer generators do overlap in a way that allows to perform less than n different readout basis state configurations. The stabilizer X_1X_2 e.g. requires the 1st and 2nd qubit to be read out in X-base, while X_2X_3 requires the 2nd and 3rd qubit in the same basis. Therefore, a single run using the measurement configuration $X_1X_2X_3$ reveals the information necessary for both, the calculation of $\langle X_1X_2\rangle$ and $\langle X_2X_3\rangle$. This has the advantage of less measurement configurations and possible parallel data acquisition for multiple expectation values, at the cost of non-independent errors of the $\langle g_i\rangle$. For the following experimental realizations of four-, five- and six-qubit GME, it was possible to take enough measurement shots to evenly split up the data sets into subsets. All stabilizer generators are calculated from independent data sets, which were measured interleaved. Therefore, the error can be calculated using Gaussian error propagation

$$\Delta\langle g_i\rangle \quad = \quad 2\Delta p_{+1}, \tag{8.8}$$

with the binomial shot noise error

$$\Delta p_{+1} = \sqrt{p_{+1}(1-p_{+1})/(N-1)}, \tag{8.9}$$

where N is the number of shots.

8.2 Four-Qubit GME

Witnessing of four-qubit genuine multipartite entanglement is carried out using the FT PCM circuit scheme, using only data and syndrome auxiliary qubit. All qubits are initialized to the state $|+\rangle$ by means of an initial local rotation $R(3\pi/2, -\pi/2)$ applied to $|0\rangle$. The entangling gates G_{ij}, further described in Sec. 2.9.2, feature a rephasing pulse $R(\pi, -\pi/2)$ to maintain coherence throughout the circuit. The flag qubit does not participate in the four-qubit entanglement circuit. Therefore, the sf entangling gates are switched off during the four-qubit GME circuit, while rephasing pulses on the syndrome at the gate positions are retained. The full circuit is shown in Fig. 8.1. Analysis pulses are chosen to match the readout basis configuration of the stabilizer generator. For X_i, the final rotation on qubit i is based on a rotation $R(\pi/2, -\pi/2)$ with an additional phase $-\pi/2$ to incorporate $Z(\pi/2)$ on the data qubits before the analysis pulse and an offset $\phi_c^{(i)}$ to correct for phases induced by ion positioning in an inhomogeneous magnetic field (see Sec. 4.5). To realize the readout of Z_i, the yellow marked rotation pulse on the ith ion is paused and the usual detection is performed without final projection pulse. A waiting time equal to the analysis pulse time is added in case of $M^{(Z)}$, to maintain the sequence timing and ion positioning induced phases. Data qubits are either measured in $M^{(X)}$ or $M^{(Z)}$, depending on the stabilizer generator. The syndrome is always measured in $M^{(X)}$, since the expectation values are conditioned on the $M_s^{(X)}$ result.

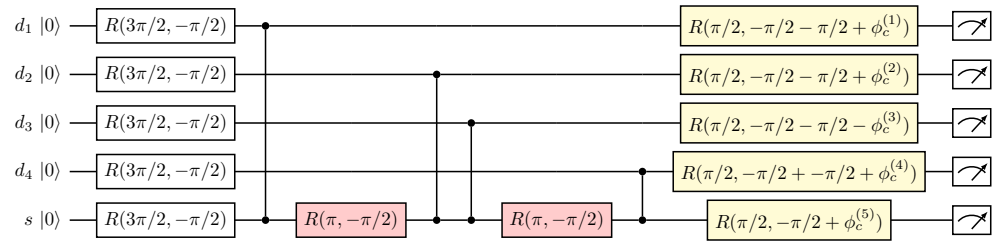

Figure 8.1: Quantum circuit for a four-qubit GME witnessing using the FT PCM circuit scheme. The flag qubit is not shown, since it does not participate in the entanglement. The first rotation (white) initializes all qubits to the state $|+\rangle$. At the sf gate positions, entangling gate pulses are removed, while rephasing pulses are still executed (red). Data qubits feature a $Z(\pi/2)$ rotation after the entangling gate, which is integrated in the analysis rotation (yellow). The optional analysis rotation (yellow) is added or neglected depending on $M^{(X)}$ or $M^{(Z)}$ readout.

The expected GME state is given by

$$|\psi\rangle^{(4)} = \frac{1}{\sqrt{2}} \left(|\psi\rangle^{(4)}_{out} |-\rangle_s + |\psi^{\perp}\rangle^{(4)}_{out} |+\rangle_s \right), \tag{8.10}$$

with the sub-state for $M_s^{(X)} = +1$ to be

$$|\psi\rangle^{(4)}_{out} = \frac{1}{\sqrt{2}} \left(|--++\rangle + |++--\rangle \right) \tag{8.11}$$

and the respective sub-state for $M_s^{(X)} = -1$

$$|\psi^{\perp}\rangle^{(4)}_{out} = \frac{1}{\sqrt{2}} \left(|--++\rangle - |++--\rangle \right). \tag{8.12}$$

The stabilizer generator set for the four-qubit GME can be defined as

$$\mathcal{S}_4 = \{g_1 = X_1X_2, g_2 = -X_2X_3, g_3 = X_3X_4, g_4 = \pm Z_1Z_2Z_3Z_4\}, \tag{8.13}$$

where only the parity of $Z_1Z_2Z_3Z_4$ depends on the readout result of the syndrome. The four data qubits are projected into the ± 1 eigenspace of the parity check operator upon the syndrome $M_s^{(X)}$ readout. In the case of a given input state, the data qubits are projected into one of the two GHZ-like states given in Eq. 8.11 and 8.12. Details of the measurement result for each $\langle g_i \rangle$ is shown in Fig. 8.2 and 8.3. In order to calculate the expectation values of stablizer generators, it is required to analyze probabilities of the involved data qubit computational basis states. The quantum circuit of Fig. 8.1 is repeated multiple times in two configurations. Data for the analysis rotation configurations $X_1X_2X_3X_4X_s$ and $Z_1Z_2Z_3Z_4X_s$ is acquired in an interleaved manner.

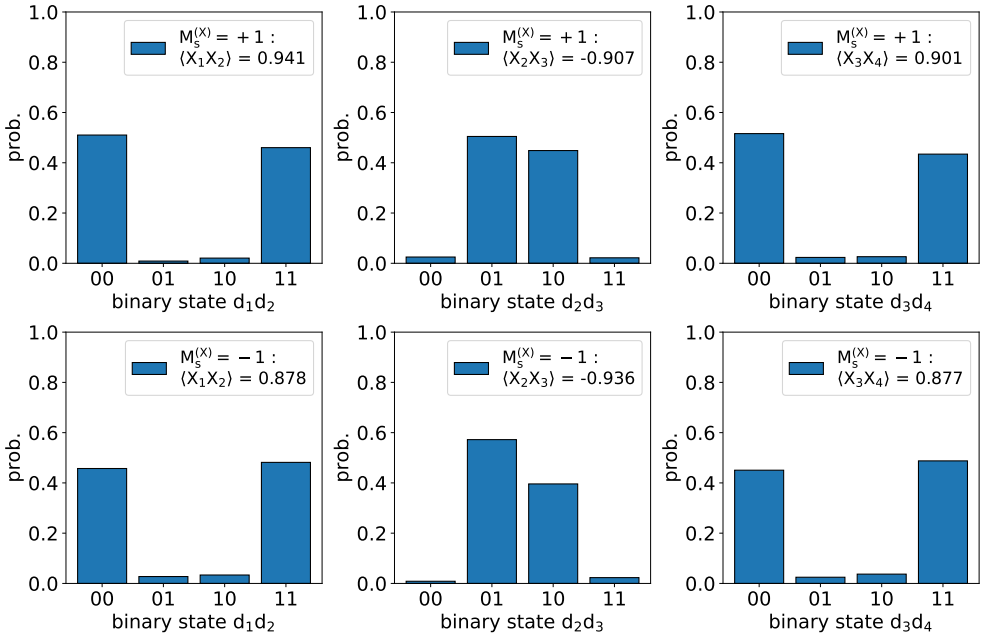

Figure 8.2: Measurement result of the X_iX_j stabilizer generator expectation values. The upper row shows probabilities of the corresponding data qubit binary states conditioned on a $M_s^{(X)} = +1$, the lower row conditioned on a $M_s^{(X)} = -1$ syndrome measurement result. The results confirm the expectation of no dependence of the X_iX_j parity conditioned on jointly measured syndrome parity.

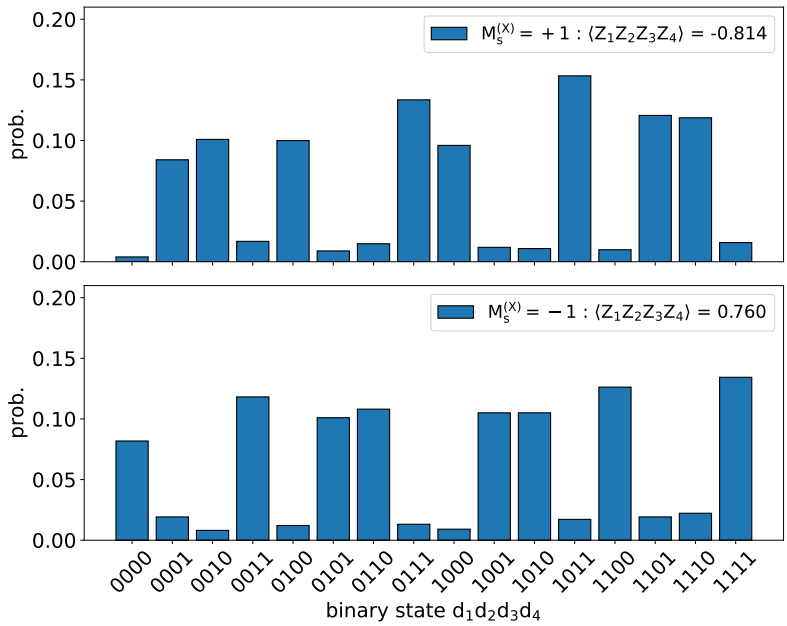

Figure 8.3: Measurement result of the $Z_1Z_2Z_3Z_4$ stabilizer generator expectation value. The data subsets conditioned on the $M_s^{(X)}$ readout clearly reveal the dependency of the $Z_1Z_2Z_3Z_4$ parity on the syndrome state. 2001 shots were taken, which results in roughly 1000 shots per conditioned Z-type stabilizer generator.

The first configuration reveals the required information to calculate X_1X_2, X_3X_4 and X_2X_3, while the second one is required for $Z_1Z_2Z_3Z_4$. Fig. 8.2 and 8.3 show the probability of the corresponding binary state results, conditioned on the syndrome readout parity $M_s^{(X)} = +1$ and $M_s^{(X)} = -1$. The analysis is based on $N = 2001$ shots for each readout rotation configuration. In order to have uncorrelated data sets, the total shots of the $X_1X_2X_3X_4X_s$ configuration are split in an alternating sequence into three sets of $N_X = 667$ single shots for each X-type stabilizer. Data separation based on the syndrome parity does not necessarily lead to evenly split subsets for $M_s^{(X)} = +1$ and $M_s^{(X)} = -1$. These subsets include roughly half of N_X shots. All 2001 shots of the second configuration can be used for the calculation of Z-type stabilizer, only divided into two data sets conditioned on the syndrome result. The dependence of stabilizer generator parity on the $M_s^{(X)}$ readout is consistent with the expected behaviour. X_1X_2, X_3X_4 (X_2X_3) do show even (odd) parity, independent of the final syndrome state. The parity of $Z_1Z_2Z_3Z_4$ clearly depends on the syndrome readout. Measured expectation values of the four stabilizer generators, calculated from the binary state probabilities, are presented in Fig. 8.4. Following the description of the normalized witness operator W_n of Eq. 8.4 using $n = 4$, the expectation value of the entanglement witness can be defined as

$$\langle W_4 \rangle = l_4 \mathbb{1} - \frac{1}{4} \sum_{i=1}^{4} \langle g_i \rangle \tag{8.14}$$

with $l_4 = 3/4$.

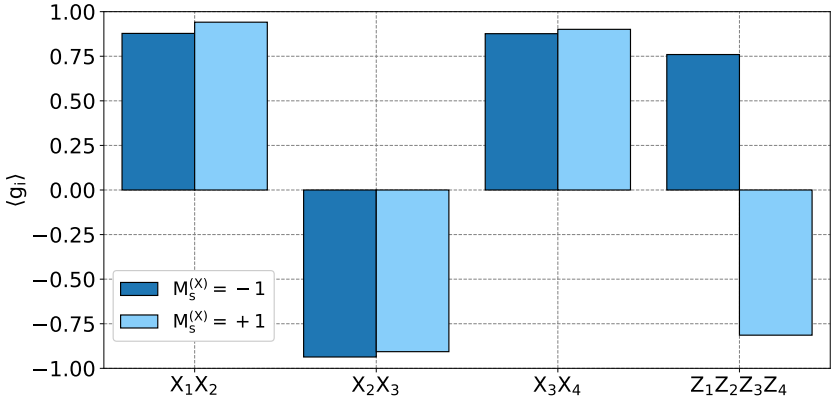

Figure 8.4: Expectation values for all stabilizer generators in Eq. 8.2, conditioned on the $M_s^{(X)}$ syndrome parity readout. The X-type stabilizers are based on about 330 shots each, while the Z-type stabilizer is based on about 1000 shots. This results in an average shot-noise error of about 2×10^{-2}.

The $\langle g_i \rangle$ are determined out of independent sets of data, which allows for the use of Gaussian error propagation without correlation. Therefore, the error is calculated using

$$\Delta \langle W_4 \rangle \quad = \quad \frac{1}{4}\sqrt{\sum_{i=1}^{4} \left(\Delta \langle g_i \rangle\right)^2}. \tag{8.15}$$

The entanglement witness expectation values obtained are

$$\begin{aligned} M_s^{(X)} = +1 \quad &: \quad \langle W_4 \rangle = -0.14(1), \\ M_s^{(X)} = -1 \quad &: \quad \langle W_4 \rangle = -0.11(1), \end{aligned} \tag{8.16}$$

passing zero by more than 10 standard errors each and therefore certifying conditional four-qubit GME.

8.3 Five-Qubit GME

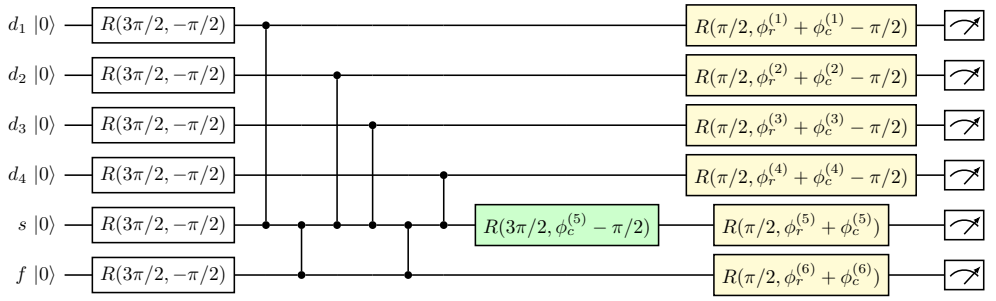

Figure 8.5: Quantum circuit of the five-qubit GME witnessing based on the FT PCM circuit scheme. All qubits, per default in state $|0\rangle$ upon sequence start, are initialized to $|+\rangle$ by means of a single-qubit rotation (white). Then entanglement gates of type G_{ij} (see Eq. 2.36) are performed between syndrome, flag and data qubits. The five-qubit entanglement is created upon insertion of an additional rotation $R(3\pi/2, -\pi/2)$ (green). Measurement in the desired basis is realized using an analysis rotation pulse (yellow) prior to the $M^{(Z)}$ detection. The calibrated phase $\phi_c^{(i)}$ serves to correct for the ion positioning induced phase on the ith qubit.

The circuit for witnessing of five-qubit genuine multipartite entanglement, shown in Fig. 8.5, builds up on the four-qubit GME circuit shown in Fig. 8.1. As in the previous circuit, all ions are intialized to $|+\rangle$ by means of a $R(3\pi/2, -\pi/2)$ rotation on the default state $|0\rangle$. The flag does also participate in this updated circuit, which is why the rephasing pulses on the syndrome are replaced by the actual sf entangling gates. After all the entangling gates, there is an additional rotation pulse of the type $R_s(3\pi/2, -\pi/2)$ applied to the syndrome, in order to create the five-qubit entangled

state. All phases $\phi_c^{(i)}$ are added to compensate the phase induced by the ion positioning in a non-vanishing magnetic field gradient along the trap axis, as described in Sec. 4.5. The generated output state of this circuit is given by

$$|\psi\rangle_{out}^{(5)} = \frac{1}{\sqrt{2}}\left(|--++-\rangle + |++--+\rangle\right). \tag{8.17}$$

In the case of five-qubit GHZ-like state including the syndrome, there is no postselection on the measurement result of s. The corresponding stabilizer generator set can be defined as

$$\begin{aligned}\mathcal{S}_5 = \langle g_1 = X_1X_2, g_2 = -X_2X_3, g_3 = X_3X_4,\\ g_4 = -X_4X_s, g_5 = Z_1Z_2Z_3Z_4Z_s\rangle.\end{aligned} \tag{8.18}$$

Following the definition of the normalized witness W_n in Eq. 8.4 and using $n = 5$, entanglement witness for the five-qubit GME is given by

$$\langle W_5\rangle = l_5 - \frac{1}{5}\sum_{i=1}^{5}\langle g_i\rangle, \tag{8.19}$$

with the entanglement bound $l_5 = 5/6$ and g_i being generators of $\mathcal{S}_5$, which project onto a five-qubit GHZ-type state. The error can be calculated using Gaussian error propagation without correlation, since independent data sets were used to obtain the expectation value for each generator. It shall be noted that the error of the generator expectation value $\Delta\langle g_i\rangle$ is the shot-noise error

$$\Delta\langle W_5\rangle \quad = \quad \frac{1}{5}\sqrt{\sum_i \left(\Delta\langle g_i\rangle\right)^2}. \tag{8.20}$$

Detailed results of the binary state probabilities for each generator out of $\mathcal{S}_5$ are shown in Fig. 8.6 and Fig. 8.7. All measurement results show a significant dependence of the probabilities on the parity of the binary state and do coincide with the expected signum of the generator according to the stabilizer generator set. A summary of the expectation values $\langle g_i\rangle$ for the whole set is presented in Fig. 8.8. In total, 2000 single measurement shots were acquired each, for two analysis pulse configurations $X_1X_2X_3X_4X_5X_sX_f$ and $Z_1Z_2Z_3Z_4Z_5Z_sX_f$. To obtain the expectation values of the X-type stabilizers, the data set is evenly divided in an interleaved manner into 500 shots each. This results in a shot-noise error of about 1×10^{-2}. All 2000 shots were used for the Z-type stabilizer. Based on Eq. 8.19, the measurement results in an entanglement witness of

$$\langle W_5\rangle = -0.076(6), \tag{8.21}$$

exceeding the bound by about 12 standard errors, thus verifying the witnessing of a five-qubit GHZ-like state.

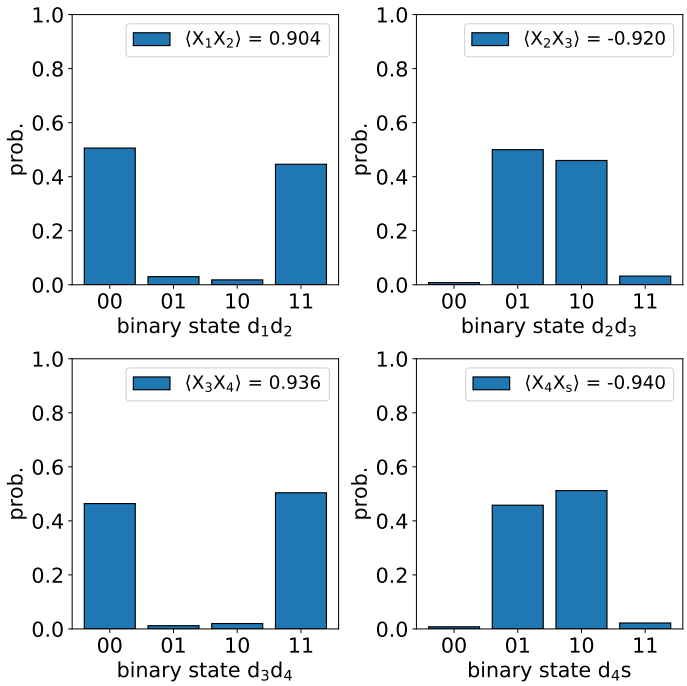

Figure 8.6: Measurement result of the X-type stabilizer generator expectation values. The results show a high selectivity of either parity +1 or -1 states, for all X_iX_j generators. The measured parity coincides with the expected sign of the g_i in the set of stabilizer generators in Eq. 8.18. 500 shots were acquired for each g_i, leading to a shot noise error of about 1×10^{-2}.

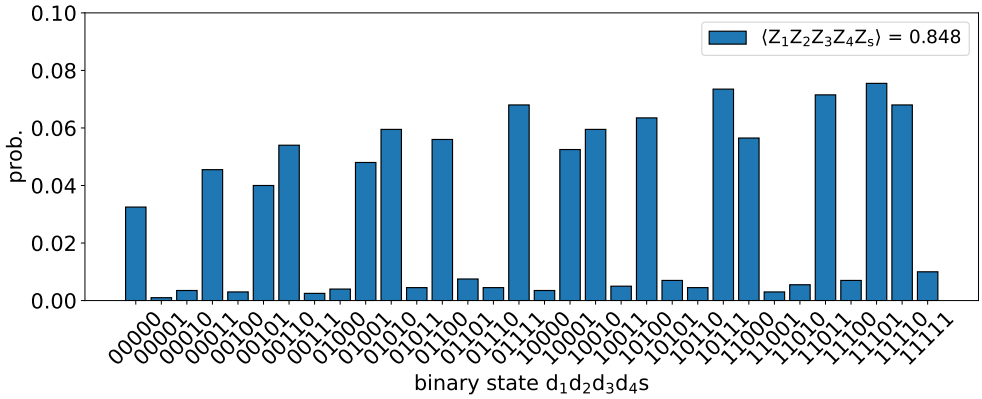

Figure 8.7: Measurement result of the $Z_1Z_2Z_3Z_4Z_s$ stabilizer generator expectation value. As expected, the probabilities to measure a $d_1d_2d_3d_4d_s$ binary state with even parity is significantly higher than the states with odd parity. The data is based on 2000 shots.

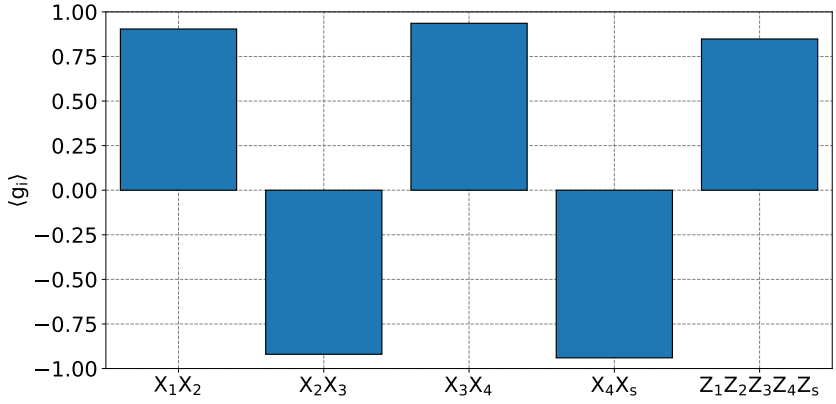

Figure 8.8: Expectation values for all stabilizer generators of the set $\mathcal{S}_5$. Results for X-type stabilizers are based on 500 shots each, Z-type stabilizer on 2000, leading to a shot noise error of about 1×10^{-2}.

8.4 Six-Qubit GME

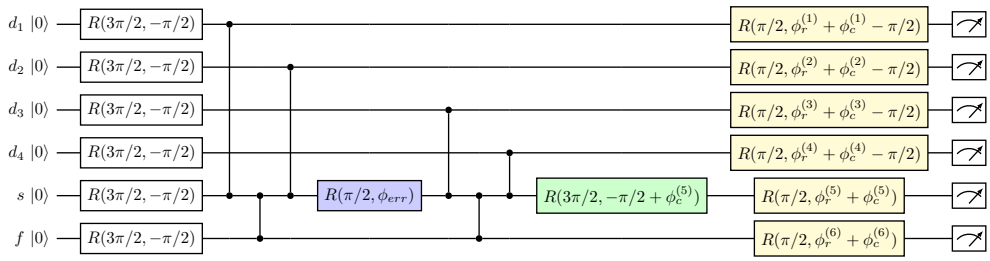

Figure 8.9: Quantum circuit for the generation of six-qubit multipartite entanglement. The circuit builds upon the five-qubit GME in Fig. 8.5. An Additional coherent rotation is added (blue) in between the entangling gates d_2s and d_3s.

The witnessing technique is used to show that the implemented circuit is capable of generating six-qubit genuine multipartite entanglement. The quantum circuit shown in Fig. 8.9 is based on the circuit used for five-qubit GME depicted in Fig. 8.5. It includes all six qubits of the FT PCM scheme, as well as all the single- and two-qubit gates of the five-qubit GME witnessing. Additionally, a rotation of the form $R_s(\pi/2, \phi_{err})$ is added on the syndrome between the gates d_2s and d_3s. This pulse can in the quantum error correction context be seen as a coherent rotation error, where ϕ_{err} is calibrated in order to correct for the ion positioning induced phase of the syndrome, to ensure a coherent X-type error. This error propagates through the subsequent entangling gates and finally results in a six-qubit superposition state. For the $\pi/2$ coherent rotation, an equal-weighted coherent superposition state is created of the form

$$|\psi\rangle_{out}^{(6)} = \frac{1}{\sqrt{2}}\left(|\psi\rangle_{out}^{(5)}|-\rangle_f + i\,|\psi^{\perp}\rangle_{out}^{(5)}|+\rangle_f\right), \tag{8.22}$$

with the two five-qubit states:

$$\begin{aligned} |\psi\rangle_{out}^{(5)} &= \frac{1}{\sqrt{2}}\left(|--++-\rangle + |++--+\rangle\right), \\ |\psi^{\perp}\rangle_{out}^{(5)} &= \frac{1}{\sqrt{2}}\left(|----+\rangle + |++++-\rangle\right). \end{aligned} \tag{8.23}$$

The first one corresponds to the error propagating onto two data qubits and the flag-qubit capturing the error, indicated by the $|-\rangle_f$. The second state corresponds to the fault-free component, indicated by $|+\rangle_f$, respectively. The stabilizer generator set of the GME state $|\psi\rangle_{out}^{(6)}$ is given by

$$\begin{aligned}\mathcal{S}_6 \quad = \quad & \{g_1 = -X_3X_s, g_2 = -X_4X_s, \\ & g_3 = -X_1X_sX_f, g_4 = -X_2X_sX_f, \\ & g_5 = Z_1Z_2Y_f, g_6 = Z_1Z_2Z_3Z_4Z_s\}.\end{aligned} \tag{8.24}$$

Expectation values $\langle g_i \rangle$ are acquired using two measurement configurations, $X_1X_2X_3X_4X_sX_f$ and $Z_1Z_2Z_3Z_4Z_sY_f$, with 2000 shots each. The acquired data is split upon the X- and Z-type stabilizers in an interleaved manner, in order to prevent possible calibration drifts over time to affect the expectation values unevenly. Using this data set distribution, the four X-type stabilizers $-X_3X_s$, $-X_4X_s$, $-X_1X_sX_f$ and $-X_2X_sX_f$ are based on 500 shots each. The Z-type stabilizers $Z_1Z_2Y_f$ and $Z_1Z_2Z_3Z_4Z_s$ are based on 1000 shots.

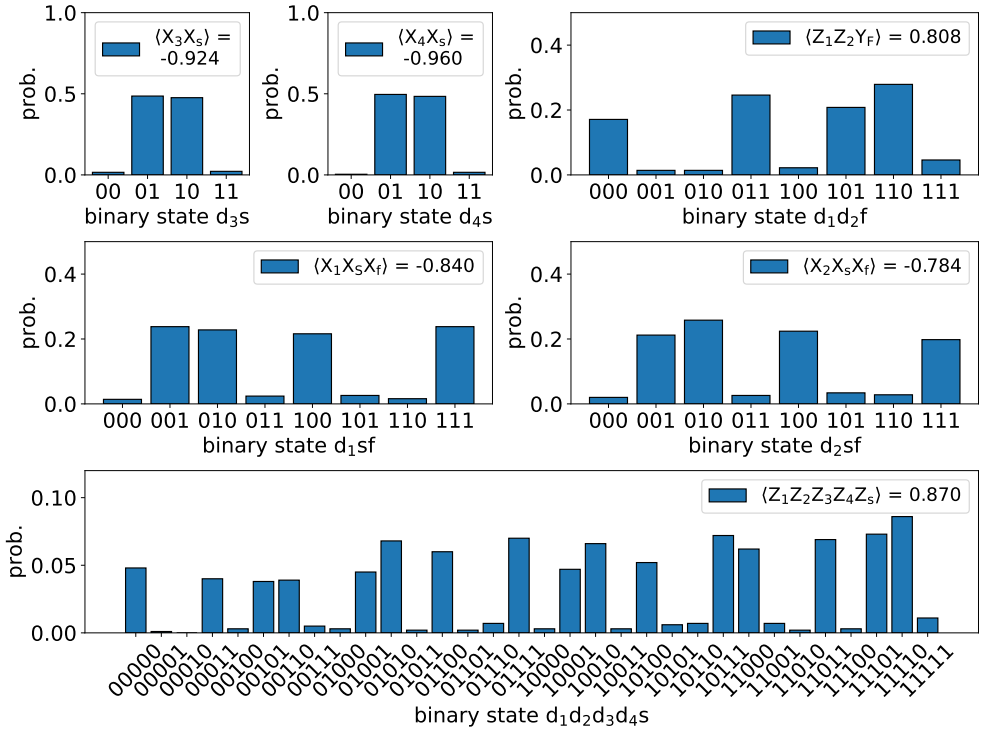

Figure 8.10: Measurement results of all stabilizer generators in the set of Eq. 8.24. Probabilities of the measured binary states are clearly depending on the parity of the state. All expectation values do coincide with the expected sign, corresponding to the expected output state in Eq. 8.22. X-type stabilizers are evaluated from 500 shots each, while the Z-type stabilizers are based on 1000 shots. This leads to an average shot-noise error of about 2×10^{-2}.

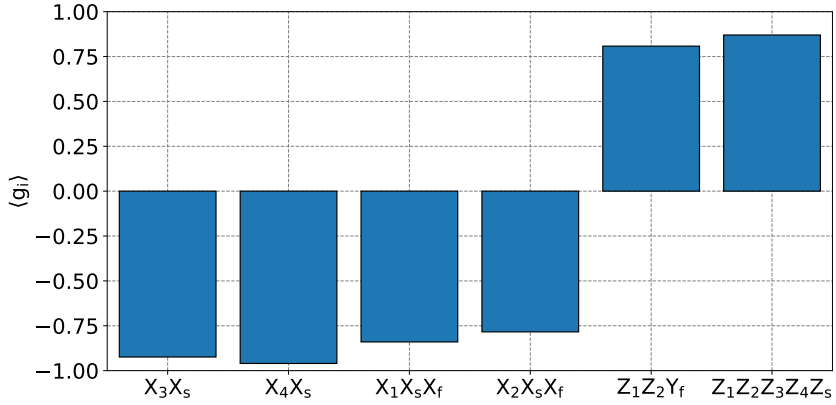

Figure 8.11: Expectation values for all stabilizer generators of the set $\mathcal{S}_6$. All values show a high agreement with the expected result. The data was acquired using two measurement configurations, $X_1X_2X_3X_4X_sX_f$ and $Z_1Z_2Z_3Z_4Z_sY_f$, 2000 shots each. The four X-type stabilizers are based on 500 shots each, the two Z-type stabilizers on 1000 shots.

The expectation value of the witnessing operator can be calculated using the definition of the normalized witness W_n in Eq. 8.4 and using $n = 6$ as

$$\langle W_6 \rangle = l_6 - \frac{1}{6} \sum_{i=1}^{6} \langle g_i \rangle \,, \tag{8.25}$$

with the entanglement bound $l_6 = 5/6$ and stabilizer generators g_i out of the set $\mathcal{S}_6$. Since the $\langle g_i \rangle$ are from independent data sets, Gaussian error propagation without correlation can be applied. The witness error is based on the shot-noise error of each stabilizer generator expectation value $\langle g_i \rangle$:

$$\Delta \langle W_6 \rangle \quad = \quad \frac{1}{6} \sqrt{\sum_i \left(\Delta \langle g_i \rangle \right)^2}. \tag{8.26}$$

The measured expectation values in Fig. 8.11 can be combined to the expectation value of the 6 QB GME witness according to Eq. 8.25, resulting in

$$\langle W_6 \rangle = -0.031(8). \tag{8.27}$$

This expectation value below zero shows that the entanglement bound is exceeded by 3.8 standard errors. Therefore, the capability of the flag-based PCM circuit to create a six-qubit GHZ-like state is verified, verifying the entangling gate fidelity reached in the experimental setup. Compared to the four-qubit GME expectation value, exceeding its entanglement bound of $l_4 = 3/4$ by more than 10 standard errors, the expectation value for six-qubit GME does exceed its bound of $l_6 = 5/6$ only by 3.8 standard errors.

For higher number n of qubits, it gets more challenging to exceed the bound, due to the definition of the bound to be $l_n = (n-1)/n$ and additionally the integration of stabilizer generators of higher weight in the set of stabilizer generators $\mathcal{S}_n$.

Chapter 9

Outlook

Experimental results of the fault-tolerant parity check measurement confirm the ability to perform this circuit on our shuttling-based trapped-ion architecture. Currently, the fidelity is not yet sufficiently high to demonstrate above-threshold quantum error correction. Therefore, it is necessary to investigate the error sources and improve the operational fidelity further. The significant SPAM error when performing an increased number of shuttling operations can be addressed by minimizing the coupling to the axial mode of motion, e.g. by fine tuning the angle of the incoming 729 laser beam relative to the trap axis. The contrast loss of the syndrome qubit, where the parity information is mapped to, can be assigned to ambient magnetic field fluctuations and is consistent with the observed parity measurement fidelity. The dominant error source of qubit dephasing due to magnetic field fluctuations can be addressed by improving the ambient magnetic field shielding. Currently, a next generation shuttling-based quantum processor is under construction, including an improved three-layer μ-metal shielding. First preliminary measurements predict a roughly $1000\times$ higher shielding factor, with respect to the current setup. Currently, the shuttling operations cover a significant part of the sequence execution time. One aim is to speed up the shuttling operations by means of faster ramping of the DC electrodes while keeping motional excitation low. Further ongoing improvements on the analog output stage of the arbitrary waveform generator and DC voltage ramp filtering, in combination with simulation based numerical optimization of the individual shuttling voltage ramps, are expected to result in much faster shuttling operations. Readout and verification consume about one third of the total sequence time, evenly distributed over shuttling of the ion crystals and actual detection and verification laser pulses. One option for readout improvement is the use of a second ion species, where the qubit state can be mapped on to for detection.

Based on the presented FT PCM building block, errors upon a data qubit register encoding a logical qubit can be detected. To apply a correction rotation to the data qubit register, depending on the single-shot error syndrome measurement result, it is required to perform in-sequence readout and conditional execution of subsequences. Currently, this feature is under development and will soon be integrated in the custom multi-channel arbitrary waveform generator of the quantum computing control setup. Once integrated, this enables the ability to attempt full quantum error correction schemes and correction of a logical qubit.

The general idea of the shuttling-based approach is to keep the ion crystals small enough to benefit from the enhanced control. It can be seen as a 'divide and conquer' approach, performing operations on subsets of the qubit register. But a fully shuttling-based approach with only up to two commonly confined ions is not easily scalable in a linear trap configuration. When increasing the number of ions, a large amount of reconfiguration operations would be required. Therefore, the next generation of the hardware setup is including an addressing unit replacing the laser interaction zone described in this work, allowing to individually address up to ten ions confined in a common potential well. Increasing the qubit register size on which operations are performed on, allows to reduce the shuttling overhead, while keeping the flexibility of reconfiguration of the larger subsets. These larger qubit registers can then be separated into smaller subsets and recombined, very similar to the separation and recombination of two-ion crystals presented.
For more than a thousand qubits, the standard QCCD approach might be challenging due to the increased complexity of junctions, laser access and extensive hardware resource for the individually supplied DC electrodes of the segmented trap chip. Therefore, it is of interest to interconnect multiple individual quantum processing units, such that quantum information can be coherently transported. Shared entanglement between registers on different trap chips can be realized by either moving an ion over long distances to the next trap module [56] or via photonic interconnections. Coupling a trapped ion to an optical cavity could be used to transfer the quantum information between ions (static qubits) and photons (flying qubits) with high interaction rates. An approach with reduced technical complexity is to combine photons emitted by ions and collected using high NA optics in the different registers on a 50/50 beamsplitter. The interference of the photons can be used to herald ion-ion entanglement with high fidelity [131]. While there are still technical challenges to address, proof-of-concept for both techniques have already been realized [56, 58], underlining the scalability of a shuttling-based trapped-ion quantum computer.

Appendix A

Additional Measurement Results

A.1 Probing Motional Sidebands after Separation Operation

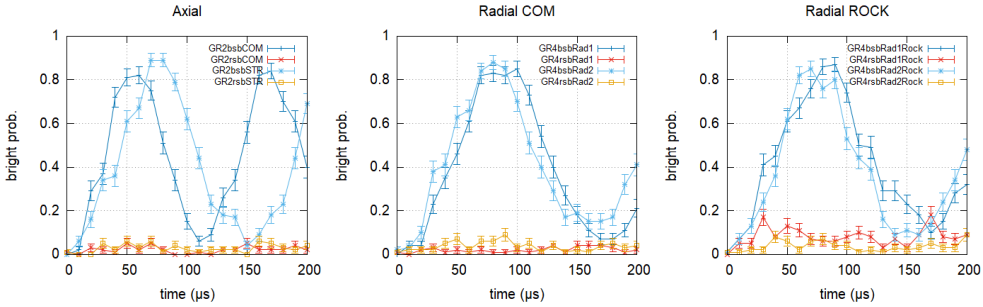

Figure A.1: Reference of rsb and bsb pulse duration scans after resolved sideband cooling.

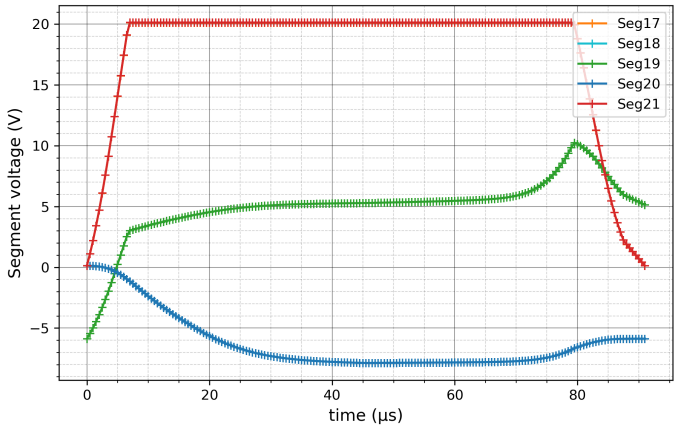

Figure A.2: Time-dependent voltage ramp of the separation operation with typical separation duration of 90 µs.

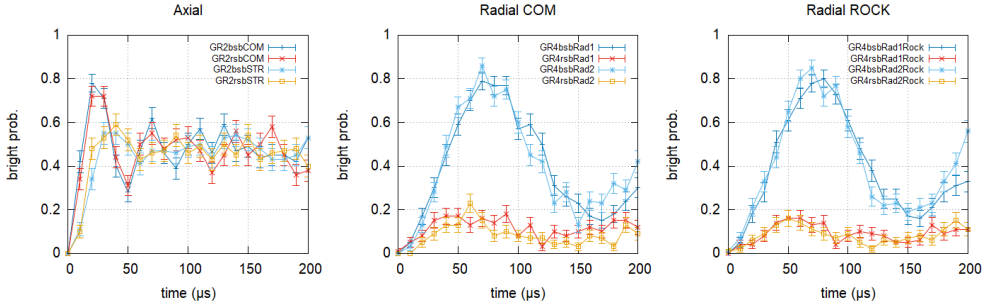

Figure A.3: Probing of rsb and bsb after one separation and merge operation using the voltage ramp of Fig. A.2.

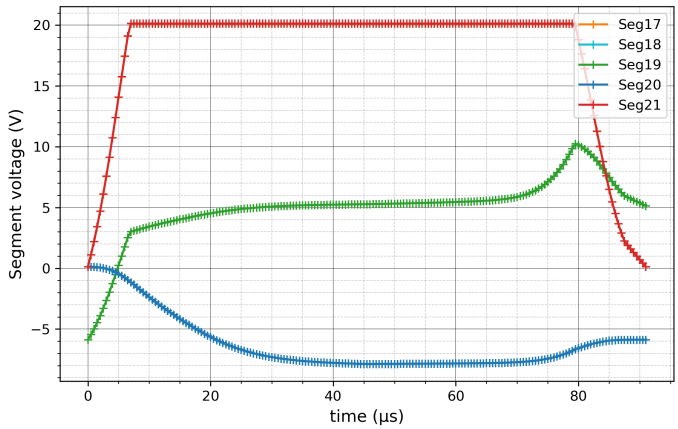

Figure A.4: Time-dependent voltage ramp of the separation operation with typical separation duration of 90 µs. The α calibration was performed with the additional surrounding potential wells for the typical six ion configuration, as described in Sec. 5.2.

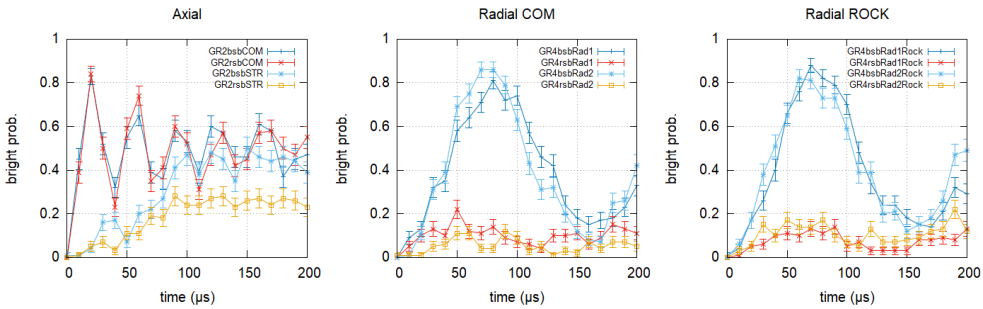

Figure A.5: Probing of rsb and bsb after one separation and merge operation using the voltage ramp of Fig. A.4. The axial motional excitation, especially of the stretch mode, is reduced compared to Fig. A.3.

A.2 Frequency Spectra of Radial Modes with Surrounding Potential Wells

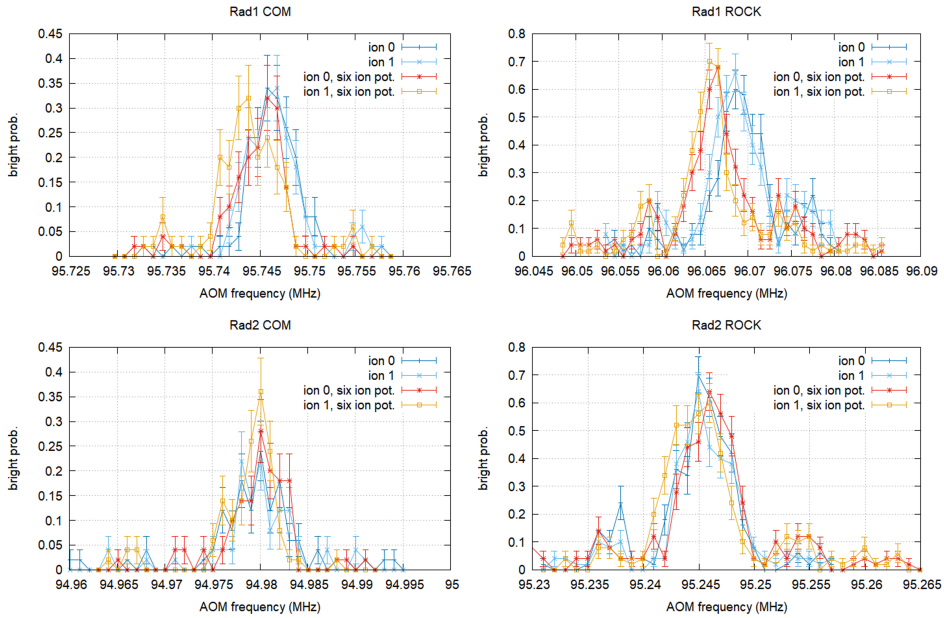

Figure A.6: Frequency spectra of the radial modes of motion of a two-ion crystal. Each spectrum is measured once without surrounding potential wells and once with the typical potential well configuration of a six ion sequence, referred to in Sec. 5.2.

A.3 Probing Motional Excitation after Voltage Kicks

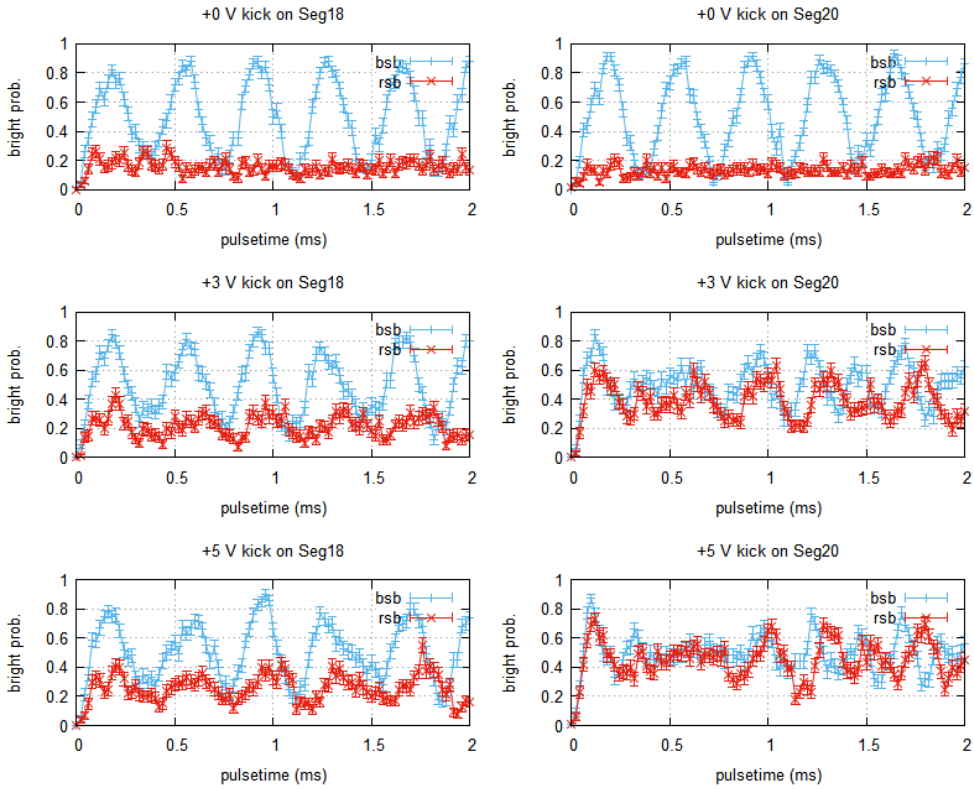

Figure A.7: Rabi flops on the axial com mode after a voltage kick on segment 18 or segment 20. The ion is located at segment 19 (LIZ) during the kick on a neighboring segment. The voltage is only applied during a single analog sample update, which is 380 ns. The fast voltage kick causes motional excitation on the axial com mode of the ion. Even though segment 18 and 20 are arranged symmetric around the LIZ, the same voltage kick results in a significantly different excitation.

A.4 Probing Motional Excitation after Swap Operation

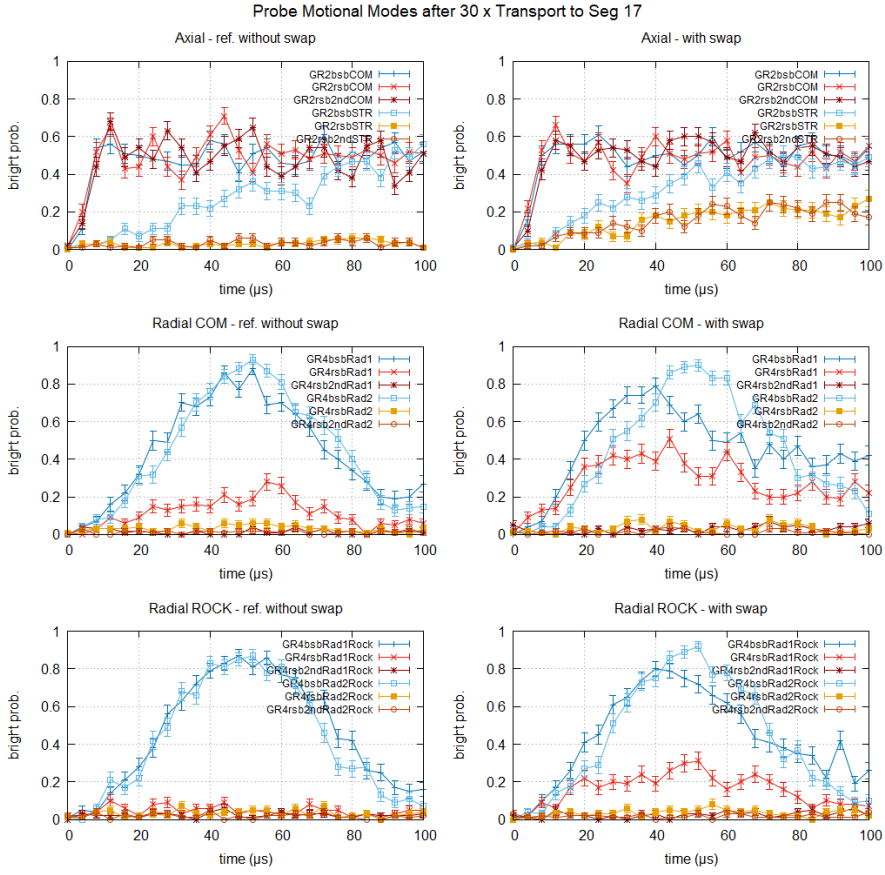

Figure A.8: Comparison of the motional excitation by measuring Rabi flops of the motional sidebands of a two-ion crystal after 10 transport operations with and without subsequent positional ion swap. Here, the swap prior to the improvement is used. The transport operations do cause motional excitation mostly on the axial com mode of the ions. After the swap, there is a significant increase visible on the red sidebands of the axial stretch and rad1 rock mode.

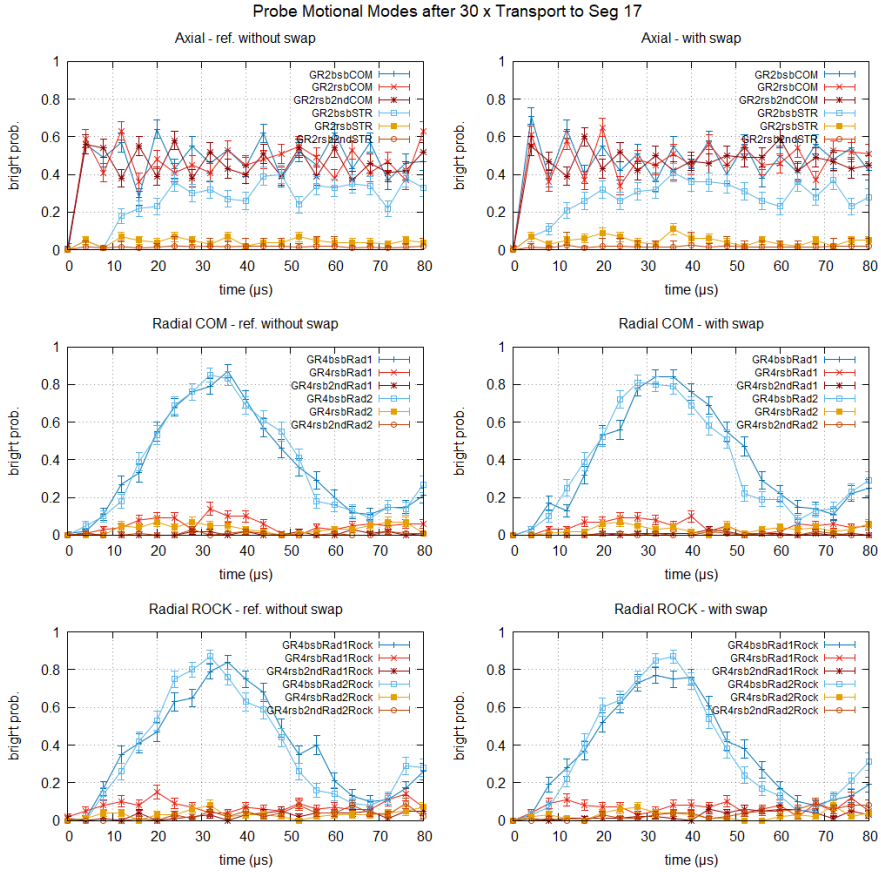

Figure A.9: Comparison of the motional excitation by measuring Rabi flops of the motional sidebands of a two-ion crystal after 10 transport operations with and without subsequent positional ion swap. The improved swap operation with low variation of the trap frequencies is used.

Appendix B

Additional Figures

B.1 ZZ-Gate with Rephasing Pulse

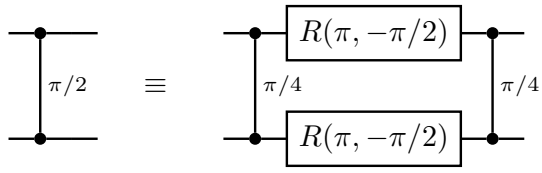

Figure B.1: ZZ-gate including a rephasing pulse. The total desired phase accumulation is split in two ZZ-type gates, interleaved by a spin-flip rotation on both qubits, leading to suppression of spin-asymmetric error sources [84].

B.2 Laser-Driven SWAP Operation

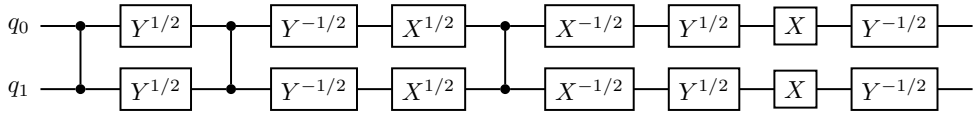

Figure B.2: Circuit of the laser-driven SWAP operation (up to a global phase). The circuit including three entangling gates and several single-qubit rotations is temporary used instead of the positional ion swap, in order to compare the performance of the FT PCM reconfiguration operation scheme with both swaps. The comparison revealed an issue with the positional ion swap operation. After improving the positional ion swap, the laser-driven swap operation is typically not in use.

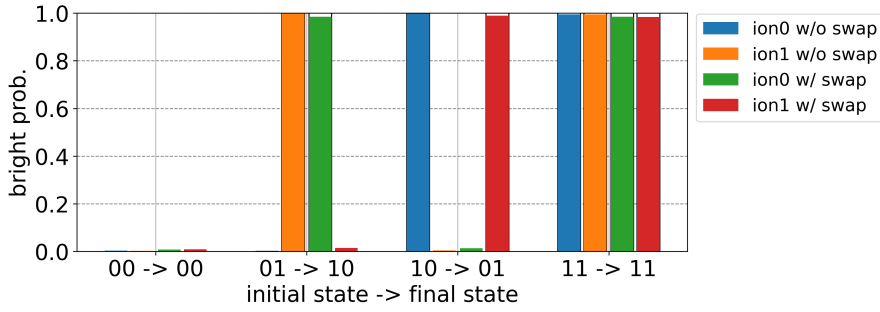

Figure B.3: Test of the laser-driven swap operation with all two-qubit computational basis states as input. The comparison of the 'bright' probability shows a high fidelity of the laser-driven swap operation.

Appendix C

Software Details

MCP - Master Control Program

The MCP project is divided into three major packages. It consists of MCPLibs to maintain and handle the include of third-party libraries, the MCPHardware, with its hardware related utility libraries and the MCPCore, which includes the graphical user interface, the loading of the dynamic link libraries during runtime and the actual executable.

MCPLibs (third-party libraries) The MCPLibs package allows easy access to external libraries in the compatible version, without a separate install process by the user. Upon first execution of the Install.ps1 script of MCP, the corresponding precompiled binaries are loaded from the ConanPackages git repository set up for the quantum computing software. This serves reliability and availability, since it is independent of the access to the actual original sources and version updates are under full control of the quantum computing software maintainers. The external libraries currently used are:

- **Boost** Peer-reviewed collection of c++ libraries.[1] It includes multiple high quality, well maintained libraries for e.g. file read/write access, regular expressions and multithreading. Boost libraries are used within a large number of the custom libraries. The version currently used is boost 1.71.0.
- **QT** Framework to create the graphical user interface of the master control program.[2] The widget toolkit is used to create the gui layout such as windows, tabs and buttons. The version in use is qt 5.9.8.
- **MCPLibs::Qwt** The Qwt - Qt widgets for technical applications - library includes utility classes and gui components based on qt.[3] Qwt is used to create dynamic 2D plots as the EMCCD camera picture or the PMT count vs. time directly on the MCP gui. The version in use is qwt 6.1.4.
- **MCPLibs::NI** NI-DAQmx driver to communicate with National Instruments hardware such as the multifunction I/O devices. Requires the previous

[1] https://www.boost.org/
[2] https://www.qt.io/
[3] https://qwt.sourceforge.io/

installation of the driver.[4][5] The module MCP::NI uses this driver to provide a convenient user interface. The current driver version is NI-DAQmx 21.0.

- **MCPLibs::fftw** FFTW, standing for "Fastest Fourier Transform in the West", is a well established library to compute the discrete fourier transform.[6] It is used in every of our c++ modules that needs fft computation. The version currently used is fftw 3.3.8.

- **MCPLibs::OpenCV2** OpenCV is a real-time computer vision library.[7] It is used to process real-time image data, such as the camera pictures of cavity transmission. These beams are watched in order to capture laser unlocking events and be able to automatically react, e.g. by pausing the measurement sequence or trying to relock. The version in use is opencv 4.1.1.

- **MCPLibs::eigen** Eigen is a header-only c++ library for linear algebra.[8] It provides generic classes to perform fast and reliable calculations with matrices and vectors of any size. The version in use is eigen 3.3.5.

- **MCPLibs::qpp** Quantum++ is a header-only quantum computing library based on eigen 3 [87].[9] It implements commonly used quantum gates and operations and is currently used within the Randomized Benchmarking code of the MicroScriptDll in version qpp 2.5.

- **MCPLibs::OPC** OPCClientToolkit to communicate to a server using Open Platform Communication (OPC).[10] OPC is a standard for reliable and secure data exchange in the industrial telecommunication. This module is currently not used by MicroTools or MicroScriptDll, but provides backwards compatibility for other labs. The used version is OPCClientToolkit v0.3.

- **MCPLibs::OpenSSL** A toolkit for the Transport Layer Security (TLS) and the Secure Sockets Layer (SSL) protocol.[11] it also includes cryptography functionality and can be used where TLS/SSL is required in the communication. The currently used version is openssl 1.1.1g.

- **MCPLibs::JSON** JSON - JavaScript Object Notation - is a human-readable data format. It is following the JavaScript object syntax. It consists of data objects as attribute-value pairs and arrays, which can be hierachichally stacked in a tree structure. One advantage is not only the human-readability but also the language-independence of the format and the availability of third-party libraries for multiple programming languages as python and c++. This allows to easily use this format for data exchange between the different parts of the software

[4] https://www.ni.com/de-de/support/downloads/drivers/download.ni-daqmx.html
[5] https://www.ni.com/de-de/support/downloads/drivers/download.ni-488-2.html
[6] https://www.fftw.org/
[7] https://opencv.org/
[8] https://eigen.tuxfamily.org
[9] https://github.com/softwareQinc/qpp
[10] https://sourceforge.net/projects/opcclient/
[11] https://www.openssl.org/

stack. To process data in JSON format on the c++ side, the MCP uses the nlohmann_json library in version 3.9.1.[12]

- **MCPLibs::JSONRPC** JSON-RPC is a remote procedure call (RPC) protocol, which is encoded in JSON. It can be used to send a method execution request to a server. Within the JSON format, the name of the method to be exectued and parameters can be passed to the server, as well as an id to receive the corresponding responce. JSON-RPC is e.g. used to communicate with the next generation of the custom arbitrary waveform generator. Included in MCPLibs is a lightweight c++ JSON-RPC 2.0 library named jsonrpc++ version 1.3.2.[13]

MCPHardware The MCPHardware package includes multiple interfaces for convenient hardware access. Since most hardware manufacturer provide an API for c++, the Modules mostly consist of wrapper classes to provide the functionality needed within our control software structure. Then for each device of a certain type, an instance of this class can be created. The device settings can be either controlled directly via the GUI or within the control software. The MCPHardware package does also include deprecated libraries, which are kept for backwards compatibility reasons, since a device using it might still exist in some setups. This backwards compatibility is of importance to easily update the software in several labs to the new modularized system, without the need of hardware adaptions.

- **MCP::AllowIO** The AllowIO library is used within MCP::DAC8814 and MCP::DigitalIO. It is used for communication purposes via PortTalk 2.0. This module is deprecated and does not receive further development.

- **MCP::AndorCam** The AndorCam module does provide an interface to the Andor EMCCD iXon camera, which is used to capture 2D images of the trapped ions. The Camera class serves as a wrapper for the functions of the API by the manufacturer. It does e.g. load the correct driver, initializes the camera and sets proper default parameter upon a Camera constructor call. The class CCameraHelper connects a Camera object with the PlotArray objects of MCP::gui and provides some high level camera functionality specific to our experiment such as a method GetNumIons. The connected object of type Data2DPlotArray can be placed on the MCP GUI and show a live plot of the camera image. There is the option to place region of interest (ROI) boxes on the 2D plot and show the corresponding countrate in these boxes within plots of the type Data1DPlotArray.

- **MCP::ComSocket** The ComSocket module contains classes to create sockets for network communication via IP addresses and port numbers. It includes a general socket interface, as well as two implementation for tcp communication using WinSock2 and Boost. The FPGAV3 and FPGAV4 library use this Windows socket implementation to establish a data exchange with the mAWG.

[12] https://json.nlohmann.me/
[13] https://github.com/badaix/jsonrpcpp

- **MCP::DAC8814** The DAC8814 library is used to communicate with a 16 bit digital to analog converter, also named DAC8814. This module is deprecated and does not receive further development.

- **MCP::DigitalIO** The DigitalIO library uses AllowIO, PortIODriver and IOInterface to use the LPT parallel port interface, also know as printer port. Since most PCs do not include this 25 D-Sub interface anymore, the module is deprecated and only serves the purpose of backwards compatibility to old devices. It is not in use in the current quantum computing setup.

- **MCP::FPGAV1** The FPGAV1 library is the first version of the implementation of the AnalogOut, DigitalOut and Sequence classes to structure measurement sequences which can be send to an FPGA to be executed in real-time. It uses the IOInterface library and includes the IGnuPlot library for data visualization. This software version belongs to a deprecated hardware version of the mAWG.

- **MCP::FPGAV2** The FPGAV2 library is the follow up version of FPGAV1. It provides more convenient methods for easier sequence handling by the user. This software version does also belong to deprecated hardware of the mAWG. The version currently in use is the new library Bertha::FPGAV4 including the backwards compatibility for the FPGAV3 hardware.

- **MCP::Gnuplot** The Gnuplot library serves to create graphs for data visualization e.g. to present intermediate results to the user during the measurement procedure. This plots are interactive, gnuplot input events can be registered to pass the x/y plot coordinates of the cursor and call a user-defined function using this information. This allows to e.g. set optimum values manually by clicking into a plot.

- **MCP::HWgpib** The HWgpib library can be used to communicate with hardware via GPIB (General Purpose Interface Bus). It uses the MCP::SerialStream to write and read from a device. This module is deprecated and does not receive further development.

- **MCP::HWpmt** The HWpmt library does as well make use of the MCP::SerialStream to write and read from a microcontroller connected to a photomultiplier tube (PMT). It has been replaced by the MicroTools::PMT module, which uses the more recent MCP::RS232 library. The MCP::HWpmt module is deprecated and does not receive further development.

- **MCP::IGnuPlot** The IGnuPlot library includes a very basic c++ interface to gnuplot[14], a tool to visualize dara and function in interactive plots. This plotting library is integrated in MCP, since it is used by the deprecated modules FPGAV1 and FPGAV2. For other libraries, it is recommended to use the newer MCP::Gnuplot library instead.

[14] http://www.gnuplot.info/

- **MCP::IOInterface** The IOInterface library is used by a variety of other modules, such as the deprecated ones FPGAV1 and DAC8814, but also MCP::NI and MCP::VFG150, which are still currently in use. It includes multiple useful classes for analog and digital in- and outputs, as well as classes for sequences, counting and exceptions related to hardware connections.

- **MCP::Keysight** The Keysight library provides a user interface to the Keysight/Agilent E364xA voltage/current supply. The communication is established using MCP::RS232. The methods in the KeysightE364xA class do build up the command strings according to the manual and send the data to the device at the corresponding USB virtual COM port given in the constructor.

- **MCP::NI** The NI library makes use of the third-party library provided by National Instruments and covered in the MCPLibs::NI module, as well as the IOInterface. It is in use to communicate with several types of multifunction I/O devices, e.g. the NI USB 6000/6001.

- **MCP::OPC** The OPC library, based on the third-party library in MCPLibs::OPC, is the user-friendly interface to communicate with hardware that uses the Open Platform Communication standards. It is not used in the quantum computing setup, but kept in the package to ensure backwards compatibility. The module is deprecated and does not receive further development.

- **MCP::Picomotor** The Picomotor libary does provide classes to communicate with the New Focus Picomotor control. The control PC is connected via an ethernet cable and the connection is established using Boost tcp sockets. The classes PicomotorDriver8753 and PicomotorEthernetController8752 are used to control the connection, while the Picomotor class defines the user interface. It includes easy-to-use commands such as GoForward, GoBackward and Stop to control the piezo screws which are e.g. attached to mirror mounts to control laser beam positions.

- **MCP::PortIODriver** The PortIODriver library is used by DAC8814 and DigitalIO. The module is obsolete, so are the two modules depending on it. It does not receive any further updates and only serves backwards compatibility purposes.

- **MCP::RohdeSchwarz** The RohdeSchwarz library can be used to remotely control the Rohde & Schwarz frequency generators. It includes two different options for the communication, which is either serial communication via a USB virtual COM port using RohdeSchwarzRS232 based on the RS232 module, or alternatively an ethernet connection and a boost::asio::ip::tcp socket using RohdeSchwarzTCP.

- **MCP::RS232** The RS232 library is the base for serial communication in all current modules. Therefore, it is included by several libraries in MCPHardware, as well as in MicroTools. The serial communication is realized using boost::asio::serial_port. More hardware specialized classes can inherit from the basic RS232Device and provit from the default implementations of Connect, Read, Write and CloseConnection. The serial communication parameters such

as baudrate, parity, stopbit and port number are passed to the RS232Device within the constructor call.

- **MCP::RSSML01_NI** The RSSML01_NI library is dedicated to the specific model SML01 signal generator by Rohde&Schwarz. The NI version of this library makes use of the gpib hardware interface of the MCP::NI module. This module is deprecated, the usage of the RohdeSchwarz module is recommended instead.

- **MCP::RSSML01_QIV** The RSSML01_QIV library is as well dedicated to the specific model SML01 signal generator by Rohde&Schwarz. The QIV version of the library does not require the NI module and drivers, but instead uses the HWgpib and IOInterface module. This module is deprecated, the usage of the RohdeSchwarz module is recommended instead.

- **MCP::SerialPort** The SerialPort library includes a small wrapper class around the boost::asio::serial_port. This module is deprecated, only for backwards compatibility with existing experimental source code. The usage of the more powerful RS232 library for serial communication is recommended.

- **MCP::SerialStream** The SerialStream library includes a boost::asio based iostream compatible serial port class. It is used by the HWgpib and HWpmt modules. This module is kept for supporting these still existing but deprecated modules. To establish serial connection in new libraries, it is highly recommended to use the RS232 module, which includes more features and testing.

- **MCP::TaborWW1072** The TaborWW1072 library includes the WW1072 class to remotely control the corresponding dual-channel arbitrary waveform generator by Tabor Electronics. This device and its software is currently not integrated in the quantum computing setup. The library requires the NI module and the NI drivers accordingly. The remote interface allows to set the waveform properties and the execution of triggered RF pulses.

- **MCP::ThorlabsMotor** The ThorlabsMotor library is used to control motors by Thorlabs, as in motorized rotation mounts. The class uses a WidgetWrapper on the MCP GUI, in order to control the movement of the attached motor.

- **MCP::VFG150** The VFG150 library provides the software control for the versatile frequency generator VFG150. It provides classes for pulse shapes to be executed, e.g. CVFGSqPulse for an RF pulse with a square envelope of the amplitude, as well as CVFGSmoothSqPulse, where the amplitude of the signal increases/decreases more smoothly within a certain rise/fall time. These pulses can be added to a pulse list, which is transferred to the device using the transferSequence method. The pulse types can be extended by creating new pulse classes inheriting from CVFGPulse. The implementation of the custom AddToPulseList method can be used to define how the amplitude and frequency are varied during the pulse. The interface does also allow to set a static frequency and amplitude output, instead of externally triggered pulses.

MCPCore The MCPCore package summarizes the actual executable for the Master Control Program, the libraries to control loading the lab specific ScriptDll and multiple dependencies, as well as the creation of the graphical user interface (GUI).

- **MCP::gui** The gui library of the MCPCore package includes all necessary classes to create the graphical user interface of the control program. It is qt based using the qwt widgets for e.g. buttons, input fields and simple 1D and 2D plots integrated in the gui main window. The gui offers multiple user interactions, such as adding, moving and editing controls during runtime of the application. The gui library depends on the DllLoader and Exports.

- **MCP::DllLoader** The DllLoader is responsible of loading the ScriptDll and its dependencies. When loading a dll, a lookup table of the registered symbols is created using a boost::spirit::qi based parser.

- **MCP::Utils** The Utils library summarizes several minor tool functions used in MCP in the utils namespace. It includes for example a funtion to read data from a file and also multiple functions for data manipulation, such as swapping data entries, round to nearest integer or linear interpolation between two double values.

- **MCP::Exports** The Exports library includes classes and macros regarding the export of objects and functions with their gui accessibility and update behavior. A library can create an objects of class Module, such that other libraries can register this one as a dependency. This enables the DllLoader to load not only the export table of the actually loaded experimental dll, but also the exported objects and functions of the dependencies. The macros MODULE and MODULE_DEPENDS_ON should be used to specify the dependency and the forwarding of the export.

- **MCP::Headers** The Headers library of MCP serves as a header-only include of the MCP features for the libraries to be later loaded by the MCP.exe. This covers the export functionality.

- **MCP::MCP** The MCP libary serves as a include for libraries that do need more than just header-only support, but also MCP::gui with its data structures for e.g. data visualization.

- **MCP.exe** The MCP.exe is the main executable for the master control program. It depends on gui and Utils, therefore indirectly also on the other modules of MCPCore. The executable does start up an application with a main window and a control menu in order to load a dll and add controls. The graphical user interface is stored within gui.log files, which can be modified, stored and reloaded.

Bertha

The package Bertha includes V3 and V4 of the FPGA module, to control our custom arbitrary waveform generator (mAWG). It also includes a more general module to

handle requests and a module providing an interface to an oscilloscope, which is very useful to remotely debug the output of the mAWG.

- **Bertha::JSON_RPC** The JSON_RPC library implements an interface for using the remote procedure call protocol encoded in JSON. The usual use case for the RPC protocol is a client that wants to trigger the execution of a certain method on a remote server. The JSON encoded message of the client can also include params and an id for the request. The server replies with a JSON encoded message holding a result, error and the request id. This library is using MCPLibs::JSONRPC and implements classes to handle the requests, properties and notification.

- **Bertha::MG07000B** The MG07000B library contains a JSON_RPC based interface to the Agilent MGO7000B Series Oscilloscopes. It allows to select the input channels and set parameters accordingly. The measured data can be acquired as a vector of doubles for further processing in the software. There is also the option to save a PNG picture of the oscilloscope screen.

- **Bertha::FPGAV3** The FPGAV3 library includes the c++ interface to communicate with the FPGA build into the mAWG. It contains the class DigitalData128Bit to describe the state of the 128 output pins of the FPGA in a bitwise representation. The class FPGAControl is handling the data transfer from the control PC to the FPGA and controls digital and analog sequences. It also stores the last known state of the FPGA, which corresponds to the last send data. The SequenceFPGA class combines digital and analog updates to a sequence object, such that the user access is more convenient. It includes several maps to handle the synchronized timing of all sequence parts, analog as well as digital.

- **Bertha::FPGAV4** The FPGAV4 library is an improved version of the FPGAV3, developed by Alexander Stahl. It adapts to the upcoming hardware features of the next generation mAWG. The FPGA class represents a single mAWG hardware device, on which Sequences and BranchedSequences can be created and executed. The module covers improved and extended versions of the classes to handle analog and digital outputs, as well as in-sequence measurement feedback in branched sequences.

- **Bertha::FPGAV2V4_Adapter** The FPGAV2V4_Adapter serves as connection between the FPGAV2 and the FPGAV4 library. The interface of the new FPGAV4 library can already be adapted in the description of measurement sequences, while backwards compatibility to the corresponding FPGAV2 hardware is maintained.

- **Bertha::FPGAV3V4_Adapter** The FPGAV3V4_Adapter serves as connection between the FPGAV3 and the FPGAV4 interface. The ScriptDll in the lab could already be updated to use the FPGAV4 interface, while the new hardware is not yet integrated. The adapter serves as a bridge to forward the calls to the previously used FPGAV3. This has the advantage that upon completion of the

development, the hardware replacement can be realized quickly by deactivating the adapter and switching to FPGAV4 library only.

TCPCom

The TCPCom package provides a boost::asio::ip::tcp based server, client and a client based data logger. It can easily be included to establish communication between different processes, either both in c++, or using the c interface of the library to use it via another language such as python. The client and server can both also communicate with other sockets using the transmission control protocol (tcp). Currently it is in use to communicate between the control PC and several raspberry pi, as well as communication between one MCP process running on the main control PC and another MCP process on a wavemeter control PC.

- **TCPCom::TCPClient** The TCPClient library includes the TCPCom::Client class. The constructor requires the server IP and port as input parameters. It is possible to either use the method sendRequest, sending a string message to the server while waiting for a reply, or use the sendMsg method, without necessary reply by the server. The exported user interface is intentionally small, in order to be easily used and to reduce possible failure.
- **TCPCom::TCPServer** The TCPServer library contains the exported TCPCom::Server class and the Session class. The server is started upon construction of the object, using a custom port number and callback function for the server reply. The user can hand over a custom function, which just has to satisfy the input and return type to be a std::string. The server can handle mutliple client requests. A new session is started whenever a client connects and the request is handled according to the reply function. This session handling is realized using asynchronous calls and the server can keep listening on the desired port for new clients. The server is stopped within the destructor call. A simple usage example to be able to start and stop the server during runtime using smart pointers can be found in DefaultServer.cpp. This example also serves as an easy to use default server with C interface. This global functions to start and stop the server can be handy if a process only requires a single server.
- **TCPCom::TCPDataLog** The TCPDataLog library includes the TCPDataLogger, combining the Client class with features to automatically write the received data into a file. This class is especially useful to log the data on a different device than the one acquiring the data initially, e.g. a raspberry pi with limited hardware resources. The user can specify the logging directory and the file header, the file will be split up and named according to the data acquisition time. It is also possible to set the log interval manually in millisecond precision. The received server messages can optionally also be written to std::out for debugging purposes.

MicroTools

The MicroTools package is a collection of generalized, non lab-specific tools. It contains libraries for measurement scheduling, data acquisition and evaluation and

maintenance.

- **MicroTools::ActiveMagFieldComp** Active magnetic field compensation control via C++. Uses magnetic field measurements in x,y,z-axis, received via TCPDataLogger from the raspberry pi reading out the fluxgate sensors. For each axis, an independent PID regulation uses this feedback to control the current output of a Keysight power supply, which is connected to the corresponding flat-ribbon coil. This was the first version of the active magnetic field compensation. By now a python based advanced regulation was developed and is running on the rasbperry pi connected to the sensors. The MicroTools::ActiveMagFieldComp can be used if a regulation via the control pc is required, while the python regulation on the rasbperry pi has the advantage of independence of the MCP process. The magnetic field values are only received by MCP for monitoring purposes.

- **MicroTools::DataLogging** The DataLogging library summarizes classes to be used for data logging and presentation. Currently it includes a general Data1DPlotter and the TrackedTCPDataLogger. The Data1DPlotter combines a user-defined function to get the next value. This gives full flexibility for e.g. calling another function or requesting data from a remote tcp server. This data is then inserted in a Data1DPlotArray to be presented to the user on the qt-based gui. The general Data1DPlotter serves as a convenient wrapper class combining this with start and stop functionality. The TrackedTCPDataLogger is an advanced version of the TCPCom::TCPDataLogger and requests data from a TCPCom::TCPServer at a given IP address and port and saves it into a given logging directory. It includes basic logging functionality like user-defined loginterval, start, stop but also custom plotting functions.

- **MicroTools::Evaluation** Data evaluation in parallel to the data acquisition is one of the key pieces of the control system. The evaluation library includes multiple classes for data processing and saving of the processed data. It also serves to create live plots during the measurement using the MCP::Gnuplot library. Photon counts or emccd camera pixel information can be processed using the classes DetectionPMT and DetectionEMCCD. Objects of the type StateMeasurement include a vector of Detections. A StateMeasurement represents a quantum state with multiple single-qubit detections. Analyzing the detections grouped into StateMeasurements is of interest to determine binary probabilites of a quantum register, i.e. the probability to find a two-qubit register in one of the states $|00\rangle$, $|01\rangle$, $|10\rangle$ and $|11\rangle$. It is possible to group single-qubit detections into multiple StateMeasurements. The shot data is based on a vector of StateMeasurement. The MeasurementResult class groups multiple shots. These MeasurementResults can be handled by a huge variety of data processors in the Evaluation Module, to plot and save the raw photon count data, but also histograms, state probabilites per ion and per binary state, as well as data analysis to automatically pertain calibration data. The base classes can be used to inherit from and develop even more specialized custom data processors.

- **MicroTools::LaserControl** The LaserControl library is a summary of useful tools when it comes to controlling the laser beams. It includes the WavemeterLock class, which can be used for a P regulation based on a wavemeter reading. The regulation output is fed back into the laser system via a control voltage on an AnalogOutNI. This voltage can be used to adjust the Scan Control voltage of a laser system and enables automized finetuning on the wavelength, accordingly. The TuneLaser and TuneLaserVFG classes are used to

- **MicroTools::LockGuard** The LockGuard class makes use of an AnalogInNIChannel to be found in the MCP::NI module. It is able to read in a voltage level correlated to the laser lock stability from the multifunction I/O devices by NI connected to the control PC. This can e.g. be a signal of a photodiode capturing a cavity transmission. The LockGuard provides a custom threshold for the lock signal and a logging functionality, to acquire and save the lock status data in a custom defined log interval. The LockGuard implements getIsLocked(), returning the current lock status as a bool. This function can be used within ScriptDll to automatically check the lock status for each laser represented by a LockGuard instance between measurements. The control software can then react in case of an unlock event. Currently, the sequence is paused and waits for the user to relock the laser and unpause to continue the measurement.

- **MicroTools::Multithreading** This library combines different tools related to multithreading. It includes the generic ThreadSafeQueue class, which provides thread safe operations on a double ended queue. This queue is used in MicroTools::SequenceControl, to safely hand over measurement data from the measurement thread to the evaluation thread. The class StateLoopThread is a class managing a thread supporting the states RUN, PAUSE, HOLD and STOP as well as the flags PAUSE, HOLD, STOP and LOOP. To use it, one can inherit from the StateLoopThread and implement the on... functions, i.e. onInit(), onLoop() and onPause(), to define what should happen in that state. This concept is used in the MicroTools::SequenceControl library for our MainSequence::MainSequenceThread to control the state and corresponding actions of a measurement sequence.

- **MicroTools::Optimization** The optimization library was recently developed by Andreas Conta as a bachelor thesis project []. It includes an Optimizer and Algorithm class and serves as a framework to optimize parameter sets based on measurement results. It is successfully in use to optimize the sequence of resolved sideband cooling pulses and can be used in a lot of applications as RAP pulse or qubit-gate parameter optimization.

- **MicroTools::PMT** The Photomultiplier class of the PMT module is used to read out the data of the PMT Counter device, presented in Sec. 3.4.1. It can be used to read a single photon count, but also features a continuous read mode as well as a triggered sequence mode. The continuous mode is used to acquire the current photon counts within a predefined time bin and display it in a live plot on the lab gui. To get a live feedback of the current photon count rate is

of special importance for maintenance and debugging purposes. The triggered sequence mode is used to acquire the measurement detection data. The PMT counts within a given duration are accumulated and stored. The data acquisition is triggered by a TTL output of the FPGA based custom arbitrary waveform generator. Multiple data bins within a single sequence are possible to be saved and can be retrieved after the sequence execution. MicroTools::PMT is using the MCP::RS232 to establish a serial communication with the microcontroller of the PMT counter hardware.

- **MicroTools::PhaseEstimation** The PhaseEstimation module offers a brute force bayesian estimator to acquire the phase and contrast out of measurement data. It is based on a Ramsey scheme measurement with a single ion, reading out X and Y basis. When initializing to $|0\rangle \equiv' dark'$, the probabilities $p_b^{(X/Y)}$ to find the ion in the $|1\rangle \equiv' bright'$ state, after an X/Y base readout pulse, depends on the accumulated phase Φ and the contrast c:

$$\begin{aligned} p_b^{(X)} &= \langle 1|\hat{M}_x^\dagger \hat{\rho} \hat{M}_x|1\rangle = \frac{1}{2}(1 + c\cos\Phi) \\ p_b^{(Y)} &= \langle 1|\hat{M}_y^\dagger \hat{\rho} \hat{M}_y|1\rangle = \frac{1}{2}(1 + c\sin\Phi) \\ &. \end{aligned} \tag{C.1}$$

The run(N_X,n_X,N_Y,n_Y) method of the CPhaseEstimator in this module is used to find the most likely Φ and c values for a given set of jointly measured n_X number of 'bright' shots out of N_X total shots and n_Y number of 'bright' shots out of N_Y total shots. It is e.g. used within the frequency resonance trackers in ScriptDll, since the knowledge of the phase Φ and the waiting time t used in the Ramsey scheme directly relates to the frequency deviation from resonance Δf via $\Phi = \Delta f \times t$. The deviation of the qubit frequency can be obtained and automatically corrected using the MagneticFieldTracker.

- **MicroTools::PotentialMinFinder** This modul can be used to simulate the equilibrium position of ions, based on a simulation of the trap, providing potential coefficients and the voltages applied to the segments. Rational functions are used to describe the potentials of each trap segment. It was especially designed for the splitting experiments of a two ion crystal. The set of splitting parameters results in voltage vs. time ramps for several segments. This ramps can then be verified to lead to successful ion separation, at least based on the simulation assumption.

- **MicroTools::ProbDistributions** This module contains classes for one-dimensional and two-dimensional discrete probability distributions. It stores the data of the distribution and provides the functionality to e.g. normalize the distribution and calculate the mean and standard deviation.

- **MicroTools::Sensors** The sensors module includes interfaces for multiple sensors used in the lab environment. The class MagSensorHMC5883 can be used to communicate with an Arduino/Magnetometer HMC5883L via a RS232 Serial port. These magnetometers do have a lower resolution compared to the flux-gate sensors used in the active magnetic field stabilization. They can be used to reliably measure in the µT regime, while the flux-gate sensors can resolve magnetic field changes on the nT level. But in contrast to the flux-gate sensors with external power supply and digital-to-analog converter, the HMC5883L does only require the sensor, an arduino and the USB connection to a PC. The TemperatureSensor class can read out the thermometer GMH3750[15]. The thermometers are in use to measure the temperatur of the permanent magnets. This parameter should be monitored, due to the temperature dependence of the magnetic field strength. The Sensors module does also include a Meteo class to control and read out a Yocto weather station[16]. The sensors measure the temperature, humidity and pressure, thus acquiring interesting information about the experimental environment. The data of all sensors is permanently logged in the lab. This allows for later data analysis and investigation of possible correlations with e.g. laser stability and measurement parameter drifts.

- **MicroTools::SequenceControl** The module SequenceControl is the most important part of the software, regarding the scheduling of the measurement and recalibration sequences. The MainSequenceScheduler class starts its own thread, within it handles a queue of MainSequences and the corresponding data evaluation. It ensures that only one sequence at a time is executed on the hardware and that multiple sequences wait in incoming order in a queue and are executed and stored accordingly. Sequences can be paused, resumed, stopped, saved and removed via the scheduler. The MainSequenceSchedulerGUI class provides an easy control via the graphical user interface of the MCP. MainSequence is the main class for handling an experimental sequence. The scheduler creates a new directory for each MainSequence, to store the log, data and plot files. The class offers multiple methods to either enable one of the standard scan functions to iterate a variable scan parameter, as e.g. SetLinearScan. In case the desired behavior is not covered by a default scan, the method SetScanFunction can be used to pass a custom function to the MainSequence. Multiple SetPlot[..] methods offer the possibility to turn on and off live plots for each measurement individually, overwriting the MainSequence default values given on the GUI. The actual real-time measurement sequence is described using an object of type PatternSequence. The experimental operations can be described in a function handed over to the MainSequence via a SetPattern call. To describe the measurement, the ScriptDll offers experiment-specific higher level functions for convenience, such as AddDopplerCool or AddDetection. The module covers a large number of scan functions to describe different behavior of how to iterate the ScannedParameter, such as linear, center and log scan, as well as the so-called PseudoScan and NumberedPseudoScan. They both

[15] Greisinger GMH3750, Pt100 High precision thermometer
[16] Yoctopuce, Yocto-Meteo-V1/V2

repeat a single pattern without changing parameters and use the elapsed time since start of the MainSequence execution and the iteration number as x-axis value, respectively. In order to realize automated recalibration measurements, interleaved with the actual running measurement, the default scan functions do call the execute method of all registered Tracker objects after every measurement data point. The Tracker interface, also defined in the SequenceControl module, requires at least an init, execute and clean method to be implemented by a class inheriting from Tracker. Init is called within scan functions before the data acquisition for this MainSequence starts. Execute is called in between the data points and finally, upon completion of the data acquisition, clean is called, since some Trackers might require to e.g. properly close a connection before ending. Inside the execute, each Tracker can decide on given parameters, such as the time passed since the last calibration, if it is necessary to remeasure and correct a certain parameter. A tracker example is the MagneticFieldTracker, it measures the qubit resonance frequency and corrects for drifts, e.g. caused by a drifting magnetic field. A MainSequence features two separate threads, one of the class MainSequenceThread, which is an extended version of a StateLoopThread of the MicroTools::Multithreading module, and a std::thread for the data evaluation. After the measurement of a new data point with a given number of shots, all shots of this measurement are pre-evaluated individually to investigate the data validity. Additional detections marked with the PostSelection flag are used, where the expected detection result is 'bright', independent of the qubit state. After acquisition of the required number of valid shots, the data packet is pushed into an object of type ThreadSafeQueue out of the MicroTools::Multithreading library. This queue can be safely accessed by the thread measuring the data, as well as the thread analyzing the data. Per default, the measurement thread goes on measuring, not waiting for the further evaluation to finish. This parallelization is especially interesting to speed up time intense measurement and evaluation. It is required to wait for the evaluation in cases, where the parameters of the next measurement point depend on the last result. The method SetDoParallelEval is used to disable the parallel evaluation. Then it is ensured that the data is fully analyzed and can be safely accesssed after the MeasurePoint call of the MainSequence returns with true. In order to do the evaluation of the raw data after each measurement point, the MainSequence contains a collection of default ResultProcessors of the MicroTools::Evaluation module. More ResultProcessors can be created and attached to the MainSequence via the AddResultProcessor method.

- **MicroTools::Teams** The Teams module can be used to automatically send messages to Microsoft Teams channels. Microsoft Teams is widely used for communication and for this reason the perfect channel also for the lab to communicate with the users. Messages can be automatically send in e.g. the case of parameters exceeding their desired range at a level were it requires maintenance. In the MicroTeam these information is currently shared via one Monitoring channel. It requires setting up an incoming Webhook in the Microsoft Teams configuration. Then an instance of the Teams::Webhook class can be used to directly send

messages to this channel.

- **MicroTools::Yocto** The Yocto module contains the API for the hardware of the yocto weather station. It is used by the MicroTools::Sensors module. It combines the interface provided by the manufacturer Yoctopuce with a custom class YAPI-Hub to manage registering and unregistering the device properly. This eases the usage inside the Sensors module and reduces failures.

SITCONS

The Segmented Ion Trap CONtrol System (SITCONS) is a software framework that allows for an automated generation of suitable time-dependent voltage ramps to perform ion crystal reconfiguration operations as further described in Chapter 4. The module is still under development. Currently, the hardware of the mAWG and filter boards in between mAWG and trap electrodes do deform the voltage ramps at a significant level, which renders it difficult to apply optimal filtercompensated time-dependent voltages to the electrodes. The hardware restrictions and software limitations were addressed recently [86], which paths the way of using optimized voltages adapted to very specific trapping situations with multiple potential wells at the same time.

MicroWavemeterScriptDll

MicroWavemeterScriptDll is specified to communicate with the wavemeter[17] and laser lock related tasks, such as control of supply voltages for fine tuning of a cavity and display of cavity transmission signals.

MicroScriptDll

The MicroScriptDll package includes all hardware and setup-specific experimental code. Prior to the modularization, it was the only library to be loaded by MCP.exe. After the separation into modules, the generalized packages MicroTools and MCPHardware can be easily shared between different setups and experimental libraries.

- **HDAWG** The HDAWG module is used to communicate with the corresponding arbitrary waveform generator[18]. It can be used to create shaped RF pulses, which can supply the AOMs for the laser pulse control. The HDAWG class in this module wrappes the large amount of necessary API command executions into a reduced, task-oriented interface. The Pulse class represents an RF pulse with its properties, such as amplitude, frequency, risetime, phase and the pulse type. These easy to create Pulse objects, can then be added to the HDAWG pulse list by using the addPulse method. The method executeSequence uploads the list of sequences to the HDAWG device. The pulses are going to be executed according to external trigger signals provided by TTLs of our custom mAWG.

[17] WSU, HighFinesse Laser and Electronics Systems GmbH
[18] Zurich Instruments HDAWG multi-channel Arbitrary Waveform Generator

This module is still under construction and being tested. Upon completion of the interface, it is going to be moved to the MicroTools library collection, to be shared with other labs.

- **FPGAHardwareDefinition** This module uses the Bertha::FPGAV4 and Bertha::FPGAV3V4 -Adapter for the definition of the analog and digital channels of the mAWG. The library in use is already the updated version FPGAV4, while the FPGAV3V4-Adapter provides the backwards compatibility to the established previous hardware version, currently integrated in the lab. This allows a smooth transition between the current and the next generation hardware. The FPGAHardwareDefinitions module contains global FPGA::AnalogOut and FPGA::DigitalOut definitions, connecting them with the corresponding DAC card and channel of the mAWG. The chosen names contain information about the card and channel, as in C0Ch1 for card 0 channel 1. It is recommended to use FPGAV4::VirtualAnalogOut references in the ScriptDll with task-specific naming. An example are the analog channels C0Ch0 and C0Ch8, which can be combined to a VirtualAnalogOut reference named Seg1, since they supply the two electrodes of the first trap segment. The same holds for the DigitalOut values. It is recommended to store them in names reflecting their purpose, as in PmtTTL to control the TTL trigger for the photomultiplier tube.

- **MicroScriptDll** The MicroScriptDll is the experiment-specific dynamic link library of the quantum computing experiment with the microstructured segmented ion trap. It serves as example also for other group internal setups. Previous to the work of this thesis, it included all source code about handling the hardware and controlling the sequencing and evaluation, which can now be found in improved and extended versions in the MicroTools and MCPHardware packages. This allows proper version control, separate development and unit testing of the comparably small libraries. It also allows to share the same libraries easily between multiple setups and labs. Therefore, the MicroScriptDll is mostly reduced to very specific hardware and experiment related code. Config files allow to define path variables, also for the individual usage of the dll on e.g. a development PC. For this purpose of developing and testing without the actual hardware of the lab connected, one can add a user_config.h file and define NO_HARDWARE, as well as changes to the default path variables in it. The file will automatically be included in the project if present in the config directory and is not tracked by the git repository, such that the changes do only affect the local version of the library. The HardwareDefinitions contain global variables and functions related to the hardware setup, making use of the multiple classes in the libraries of MCPHardware and MicroTools. The header and cpp file include the declaration and definition of the objects controlling the hardware. Upon creation of the object, the necessary setup-specific connection parameters, such as IP addresses or virtual com port numbers are passed to the constructor calls. These objects are usually created in the global scope of the MicroScriptDll and are also exported by the library, to be used on the graphical

user interface of MCP. The HardwareDefinitions also includes the connection of the AnalogOut and DigitalOut references from the FPGAHardwareDefinition module to more task-oriented names and structures of outputs. An example is a vector of references to the segments, such that the DC trapping voltages at a certain position can be accessed more easily using e.g. segment[i] for the ith electrode pair. A detailed knowledge of the segment - channel connection by the lab user is then not required. The source code in the CustomEvaluation category of the MicroScriptDll includes multiple ResultProcessors which are e.g. hardware specific or still under construction. If a custom evaluation processor might be useful also for other labs or setups, it is recommended to generalize its settings and move it to the shared MicroTools::Evaluation module. An additional part of the MicroScriptDll is a custom SequenceControl category extending the control included in the generalized MicroTools::SequenceControl module. It includes the SeqPatternToolkit and the SeqPatternTransportToolkit, each containing higher level methods to fill a PatternSequence with hardware commands. The lab user writing an experiment does not necessarily need to know the details about TTL and voltage ramp outputs, but just uses commands like AddIonTranslate with the position parameters for transporting ions. The implementation of these functions is very setup-specific, since the exact TTLs, channels and timings are involved, but the actual interface is quite general to trapped-ion microstructured setups. Therefore it is interesting to also generalize this to a PatternToolkit module e.g. in MicroTools, such that the methods and their interface are ensured. The setup-specific ScriptDll could then only fill the implementation with the hardware specific actions to fulfill the task such as ion transport or detection. This would allow to reuse a lot of standard calibration experiment descriptions on multiple systems without code redundancy. The Experiments category includes the definition of a large number of method to build up experimental sequences and add them to the scheduler queue for execution. Due to the EXPORT keyword, the methods are available to be placed on the MCP GUI. The input parameters can be varied on the graphical user interface and the method is executed upon click on a button. One more important feature can be currently found in the ScriptDll.cpp file. The MCP, loading the ScriptDll dynamic link library during runtime, features three functions which are called automatically on certain events: ScriptDll_Startup, ScriptDll_AutocallsExexuted and ScriptDll_Cleanup. The first one is called after the MCP loaded the ScriptDll, just after the constructors of global objects have been called. The second one is called after execution of the autocall functions of the GUI elements. Since all variables on the GUI are set accordingly after the autocall execution, this function can be used for tasks with auto-start such as starting a temperature logging thread. The last function is called when the MCP is going to close and cleans up, but before the destructors of the global objects are called. This allows e.g. for cleanly closing threads.

List of Figures

List of Tables

Bibliography

[1] G. Moore. *Cramming More Components Onto Integrated Circuits.* Electronics **38** (1965)

[2] M. A. Nielsen and I. L. Chuang. *Quantum Computation and Quantum Information.* Cambridge University Press (2000)

[3] A. Montanaro. *Quantum algorithms: an overview*. npj Quantum Information **2, 1**, 15023 (2016)

[4] F. Arute, K. Arya, R. Babbush, et al. *Quantum supremacy using a programmable superconducting processor*. Nature **574, 7779**, 505–510 (2019)

[5] Y. Alexeev, D. Bacon, K. R. Brown, R. Calderbank, L. D. Carr, F. T. Chong, B. DeMarco, D. Englund, E. Farhi, et al. *Quantum Computer Systems for Scientific Discovery*. PRX Quantum **2**, 017001 (2021)

[6] R. P. Feynman. *Simulating physics with computers*. International Journal of Theoretical Physics **21, 6**, 467–488 (1982)

[7] D. Deutsch and R. Penrose. *Quantum theory, the Church-Turing principle and the universal quantum computer*. Proceedings of the Royal Society of London. A. Mathematical and Physical Sciences **400, 1818**, 97–117 (1985)

[8] A. M. Turing. *On Computable Numbers, with an Application to the Entscheidungsproblem*. Proceedings of the London Mathematical Society **s2-42, 1**, 230–265 (1937)

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

[10] L. K. Grover. *A Fast Quantum Mechanical Algorithm for Database Search*. *Proceedings of the Twenty-Eighth Annual ACM Symposium on Theory of Computing*, STOC '96, 212–219. Association for Computing Machinery, New York, NY, USA (1996)

[11] J. I. Cirac and P. Zoller. *Quantum Computations with Cold Trapped Ions*. Phys. Rev. Lett. **74**, 4091–4094 (1995)

[12] D. P. DiVincenzo. *The Physical Implementation of Quantum Computation*. Fortschritte der Physik **48, 9-11**, 771–783 (2000)

[13] I. Bloch. *Quantum coherence and entanglement with ultracold atoms in optical lattices*. Nature **453, 7198**, 1016–1022 (2008)

[14] C. C. Nshii, M. Vangeleyn, J. P. Cotter, P. F. Griffin, E. A. Hinds, C. N. Ironside, P. See, A. G. Sinclair, E. Riis, and A. S. Arnold. *A surface-patterned chip as a strong source of ultracold atoms for quantum technologies*. Nature Nanotechnology **8, 5**, 321–324 (2013)

[15] S. Ritter, C. Nölleke, C. Hahn, A. Reiserer, A. Neuzner, M. Uphoff, M. Mücke, E. Figueroa, J. Bochmann, and G. Rempe. *An elementary quantum network of single atoms in optical cavities*. Nature **484, 7393**, 195–200 (2012)

[16] Y. Nakamura, Y. A. Pashkin, and J. S. Tsai. *Coherent control of macroscopic quantum states in a single-Cooper-pair box*. Nature **398, 6730**, 786–788 (1999)

[17] I. Chiorescu, Y. Nakamura, C. Harmans, and J. Mooij. *Coherent Quantum Dynamics of a Superconducting Flux Qubit*. Science (New York, N.Y.) **299**, 1869–71 (2003)

[18] J. Clarke and F. K. Wilhelm. *Superconducting quantum bits*. Nature **453, 7198**, 1031–1042 (2008)

[19] E. Knill, R. Laflamme, and G. J. Milburn. *A scheme for efficient quantum computation with linear optics*. Nature **409, 6816**, 46–52 (2001)

[20] H. J. Kimble. *The quantum internet*. Nature **453, 7198**, 1023–1030 (2008)

[21] W. J. Munro, K. Nemoto, T. P. Spiller, S. D. Barrett, P. Kok, and R. G. Beausoleil. *Efficient optical quantum information processing*. Journal of Optics B: Quantum and Semiclassical Optics **7, 7**, S135–S140 (2005)

[22] W. Paul and H. Steinwedel. *Notizen: Ein neues Massenspektrometer ohne Magnetfeld*. Zeitschrift für Naturforschung A **8, 7**, 448–450 (1953)

[23] W. Paul. *Electromagnetic traps for charged and neutral particles*. Rev. Mod. Phys. **62**, 531–540 (1990)

[24] E. Fischer. *Die dreidimensionale Stabilisierung von Ladungsträgern in einem Vierpolfeld*. Zeitschrift fur Physik **156, 1**, 1–26 (1959)

[25] H. G. Dehmelt. *Spin Resonance of Free Electrons Polarized by Exchange Collisions*. Phys. Rev. **109**, 381–385 (1958)

[26] H. Dehmelt. *Radiofrequency Spectroscopy of Stored Ions I: Storage**Part II: Spectroscopy is now scheduled to appear in Volume V of this series.* volume 3 of *Advances in Atomic and Molecular Physics*, 53–72. Academic Press (1968)

[27] W. Paul, O. Osberghaus, and E. Fischer. *Ein Ionenkafig*. VS Verlag fur Sozialwissenschaften (1958)

[28] P. W. Shor. *Scheme for reducing decoherence in quantum computer memory*. Phys. Rev. A **52**, R2493–R2496 (1995)

[29] A. M. Steane. *Error Correcting Codes in Quantum Theory*. Phys. Rev. Lett. **77**, 793–797 (1996)

[30] A. R. Calderbank and P. W. Shor. *Good quantum error-correcting codes exist*. Phys. Rev. A **54**, 1098–1105 (1996)

[31] D. B. Hume, T. Rosenband, and D. J. Wineland. *High-Fidelity Adaptive Qubit Detection through Repetitive Quantum Nondemolition Measurements*. Phys. Rev. Lett. **99**, 120502 (2007)

[32] A. Lupaşcu, S. Saito, T. Picot, P. C. de Groot, C. J. P. M. Harmans, and J. E. Mooij. *Quantum non-demolition measurement of a superconducting two-level system*. Nature Physics **3, 2**, 119–123 (2007)

[33] J. T. Barreiro, M. Müller, P. Schindler, D. Nigg, T. Monz, M. Chwalla, M. Hennrich, C. F. Roos, P. Zoller, and R. Blatt. *An open-system quantum simulator with trapped ions*. Nature **470, 7335**, 486–491 (2011)

[34] A. D. Córcoles, E. Magesan, S. J. Srinivasan, A. W. Cross, M. Steffen, J. M. Gambetta, and J. M. Chow. *Demonstration of a quantum error detection code using a square lattice of four superconducting qubits*. Nature Communications **6, 1**, 6979 (2015)

[35] S. J. Devitt, W. J. Munro, and K. Nemoto. *Quantum error correction for beginners*. Reports on Progress in Physics **76, 7**, 076001 (2013)

[36] J. Preskill. *Fault-Tolerant Quantum Computation*, 213 (1996)

[37] D. P. DiVincenzo and P. W. Shor. *Fault-Tolerant Error Correction with Efficient Quantum Codes*. Phys. Rev. Lett. **77**, 3260–3263 (1996)

[38] D. P. DiVincenzo and P. Aliferis. *Effective Fault-Tolerant Quantum Computation with Slow Measurements*. Phys. Rev. Lett. **98**, 020501 (2007)

[39] R. Chao and B. W. Reichardt. *Quantum Error Correction with Only Two Extra Qubits*. Phys. Rev. Lett. **121**, 050502 (2018)

[40] C. Chamberland and M. E. Beverland. *Flag fault-tolerant error correction with arbitrary distance codes*. Quantum **2**, 53 (2018)

[41] C. Chamberland and A. W. Cross. *Fault-tolerant magic state preparation with flag qubits*. Quantum **3**, 143 (2019)

[42] A. Bermudez, X. Xu, M. Gutiérrez, S. C. Benjamin, and M. Müller. *Fault-tolerant protection of near-term trapped-ion topological qubits under realistic noise sources*. Phys. Rev. A **100**, 062307 (2019)

[43] B. W. Reichardt. *Fault-tolerant quantum error correction for Steane's seven-qubit color code with few or no extra qubits*. Quantum Science and Technology **6, 1**, 015007 (2020)

[44] T. Tansuwannont, C. Chamberland, and D. Leung. *Flag fault-tolerant error correction, measurement, and quantum computation for cyclic Calderbank-Shor-Steane codes*. Phys. Rev. A **101**, 012342 (2020)

[45] C. Chamberland, A. Kubica, T. J. Yoder, and G. Zhu. *Triangular color codes on trivalent graphs with flag qubits*. New Journal of Physics **22, 2**, 023019 (2020)

[46] R. Chao and B. W. Reichardt. *Flag Fault-Tolerant Error Correction for any Stabilizer Code*. PRX Quantum **1**, 010302 (2020)

[47] M. H. Abobeih, Y. Wang, J. Randall, S. J. H. Loenen, C. E. Bradley, M. Markham, D. J. Twitchen, B. M. Terhal, and T. H. Taminiau. *Fault-tolerant operation of a logical qubit in a diamond quantum processor*, `arXiv:2108.01646` (2021)

[48] C. Ryan-Anderson, J. G. Bohnet, K. Lee, D. Gresh, A. Hankin, J. P. Gaebler, D. Francois, A. Chernoguzov, D. Lucchetti, et al. *Realization of Real-Time Fault-Tolerant Quantum Error Correction*. Phys. Rev. X **11**, 041058 (2021)

[49] L. Egan, D. M. Debroy, C. Noel, A. Risinger, D. Zhu, D. Biswas, M. Newman, M. Li, K. R. Brown, et al. *Fault-tolerant control of an error-corrected qubit*. Nature **598, 7880**, 281–286 (2021)

[50] D. Kielpinski, C. Monroe, and D. J. Wineland. *Architecture for a large-scale ion-trap quantum computer*. Nature **417**, 709 (2002)

[51] V. Kaushal, B. Lekitsch, A. Stahl, J. Hilder, D. Pijn, C. Schmiegelow, A. Bermudez, M. Müller, F. Schmidt-Kaler, and U. Poschinger. *Shuttling-based trapped-ion quantum information processing*. AVS Quantum Science **2, 1**, 014101 (2020)

[52] J. M. Pino, J. M. Dreiling, C. Figgatt, J. P. Gaebler, S. A. Moses, M. S. Allman, C. H. Baldwin, M. Foss-Feig, D. Hayes, et al. *Demonstration of the trapped-ion quantum CCD computer architecture*. Nature **592, 7853**, 209–213 (2021)

[53] R. B. Blakestad, C. Ospelkaus, A. P. VanDevender, J. M. Amini, J. Britton, D. Leibfried, and D. J. Wineland. *High-Fidelity Transport of Trapped-Ion Qubits through an* **X***-Junction Trap Array*. Phys. Rev. Lett. **102**, 153002 (2009)

[54] W. K. Hensinger, S. Olmschenk, D. Stick, D. Hucul, M. Yeo, M. Acton, L. Deslauriers, C. Monroe, and J. Rabchuk. *T-junction ion trap array for two-dimensional ion shuttling, storage, and manipulation*. Applied Physics Letters **88, 3**, 034101 (2006)

[55] B. Lekitsch, S. Weidt, A. G. Fowler, K. Mølmer, S. J. Devitt, C. Wunderlich, and W. K. Hensinger. *Blueprint for a microwave trapped ion quantum computer*. Science Advances **3, 2** (2017)

[56] F. Stopp, H. Lehec, and F. Schmidt-Kaler. *A Deterministic Single Ion Fountain*, `arXiv:2108.06948` (2021)

[57] C. Monroe and J. Kim. *Scaling the Ion Trap Quantum Processor*. Science **339, 6124**, 1164–1169 (2013)

[58] L. Stephenson, D. Nadlinger, B. Nichol, S. An, P. Drmota, T. Ballance, K. Thirumalai, J. Goodwin, D. Lucas, and C. Ballance. *High-Rate, High-Fidelity Entanglement of Qubits Across an Elementary Quantum Network*. Physical Review Letters **124, 11** (2020)

[59] J. Hilder, D. Pijn, O. Onishchenko, A. Stahl, M. Orth, B. Lekitsch, A. Rodriguez-Blanco, M. Müller, F. Schmidt-Kaler, and U. Poschinger. *Fault-tolerant parity readout on a shuttling-based trapped-ion quantum computer*, `arXiv:2107.06368` (2021)

[60] H. Häffner, C. Roos, and R. Blatt. *Quantum computing with trapped ions*. Physics Reports **469, 4**, 155–203 (2008)

[61] T. Ruster, C. T. Schmiegelow, H. Kaufmann, C. Warschburger, F. Schmidt-Kaler, and U. G. Poschinger. *A long-lived Zeeman trapped-ion qubit*. Applied Physics B **122, 10** (2016)

[62] J. Jin and D. A. Church. *Precision lifetimes for the* Ca^+ *4p* ^2P *levels: Experiment challenges theory at the 1% level*. Phys. Rev. Lett. **70**, 3213–3216 (1993)

[63] A. Kreuter, C. Becher, G. Lancaster, A. Mundt, C. Russo, H. Haeffner, C. Roos, W. Hänsel, F. Schmidt-Kaler, et al. *Experimental and theoretical study of the 3d 2D level lifetimes of 40Ca+*. Physical Review A **71**, 32504– (2005)

[64] M. Hettrich, T. Ruster, H. Kaufmann, C. Roos, C. Schmiegelow, F. Schmidt-Kaler, and U. Poschinger. *Measurement of Dipole Matrix Elements with a Single Trapped Ion*. Physical Review Letters **115, 14** (2015)

[65] D. Leibfried, R. Blatt, C. Monroe, and D. Wineland. *Quantum dynamics of single trapped ions*. Rev. Mod. Phys. **75**, 281–324 (2003)

[66] U. G. Poschinger, G. Huber, F. Ziesel, M. Deiß, M. Hettrich, S. A. Schulz, K. Singer, G. n, M. Drewsen, et al. *Coherent manipulation of a 40 Ca + spin qubit in a micro ion trap*. Journal of Physics B: Atomic, Molecular and Optical Physics **42, 15**, 154013 (2009)

[67] U. G. Poschinger. *Quantum Optics Experiments in a Microstructured Ion Trap*. Ph.d. thesis, Universität Ulm (2010)

[68] G. Poulsen. *Sideband Cooling of Atomic and Molecular Ions*. Ph.d. thesis, University of Aarhus (2011)

[69] D. J. Wineland, W. M. Itano, J. C. Bergquist, and R. G. Hulet. *Laser-cooling limits and single-ion spectroscopy*. Phys. Rev. A **36**, 2220–2232 (1987)

[70] H. Kaufmann. *A Scalable Quantum Processor*. Ph.d. thesis, Johannes Gutenberg-Universität Mainz (2017)

[71] D. Hucul, M. Yeo, S. Olmschenk, C. Monroe, W. Hensinger, and J. Rabchuk. *On the Transport of Atomic Ions in Linear and Multidimensional Ion Trap Arrays*. Quantum Information and Computation **8**, 501–578 (2008)

[72] A. Steane. *The ion trap quantum information processor*. Applied Physics B: Lasers and Optics **64, 6**, 623–643 (1997)

[73] D. James. *Quantum dynamics of cold trapped ions with application to quantum computation*. Applied Physics B: Lasers and Optics **66, 2**, 181–190 (1998)

[74] D. J. Wineland and W. M. Itano. *Laser cooling of atoms*. Phys. Rev. A **20**, 1521–1540 (1979)

[75] T. Ruster. *Entanglement-based magnetometry with separated trapped ions*. Ph.d. thesis, Johannes Gutenberg-Universität Mainz (2017)

[76] C. Monroe, D. M. Meekhof, B. E. King, S. R. Jefferts, W. M. Itano, D. J. Wineland, and P. Gould. *Resolved-Sideband Raman Cooling of a Bound Atom to the 3D Zero-Point Energy*. Phys. Rev. Lett. **75**, 4011–4014 (1995)

[77] A. Conta. *Software Framework for Automated Optimization of Operational Blocks of a Trapped-Ion-Based Quantum Computer*. Bachelor thesis, Johannes Gutenberg-Universität Mainz (2021)

[78] 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)

[79] D. Leibfried, B. DeMarco, V. Meyer, D. Lucas, M. Barrett, J. Britton, W. M. Itano, B. Jelenkovic, C. Langer, et al. *Experimental demonstration of a robust, high-fidelity geometric two ion-qubit phase gate*. Nature **422, 6930**, 412–415 (2003)

[80] D. Deutsch. *Quantum Computational Networks*. Proceedings of the Royal Society of London Series A **425, 1868**, 73–90 (1989)

[81] A. Barenco, C. H. Bennett, R. Cleve, D. P. DiVincenzo, N. Margolus, P. Shor, T. Sleator, J. A. Smolin, and H. Weinfurter. *Elementary gates for quantum computation*. Phys. Rev. A **52**, 3457–3467 (1995)

[82] A. Sørensen and K. Mølmer. *Quantum Computation with Ions in Thermal Motion*. Phys. Rev. Lett. **82**, 1971–1974 (1999)

[83] D. Pijn. *Thermodynamics in Trapped Ion Quantum Processors*. Ph.d. thesis, Johannes Gutenberg-Universität Mainz (2022)

[84] C. J. Ballance. *High-Fidelity Quantum Logic in Ca^+*. Ph.d. thesis, University of Oxford (2014)

[85] V. Kaushal. *A shuttling-based trapped-ion quantum information processing node*. Ph.d. thesis, Johannes Gutenberg-Universität Mainz (2019)

[86] M. H. Orth. *Advanced Ion Shuttling Operations for Scalable Quantum Computing*. Master thesis, Johannes Gutenberg-Universität Mainz (2021)

[87] V. Gheorghiu. *Quantum++: A modern C++ quantum computing library*. PLOS ONE **13, 12**, e0208073 (2018)

[88] H. C. Nägerl, D. Leibfried, H. Rohde, G. Thalhammer, J. Eschner, F. Schmidt-Kaler, and R. Blatt. *Laser addressing of individual ions in a linear ion trap*. Phys. Rev. A **60**, 145–148 (1999)

[89] P. Murali, D. M. Debroy, K. R. Brown, and M. Martonosi. *Architecting Noisy Intermediate-Scale Trapped Ion Quantum Computers*. *Proceedings of the ACM/IEEE 47th Annual International Symposium on Computer Architecture*, ISCA '20, 529–542. IEEE Press (2020)

[90] C. D. Herold, S. D. Fallek, J. T. Merrill, A. M. Meier, K. R. Brown, C. E. Volin, and J. M. Amini. *Universal control of ion qubits in a scalable microfabricated planar trap*. New Journal of Physics **18, 2**, 023048 (2016)

[91] A. Walther, F. Ziesel, T. Ruster, S. T. Dawkins, K. Ott, M. Hettrich, K. Singer, F. Schmidt-Kaler, and U. G. Poschinger. *Controlling fast transport of cold trapped ions*. Phys. Rev. Lett. **109**, 080501 (2012)

[92] R. Bowler, J. Gaebler, Y. Lin, T. R. Tan, D. Hanneke, J. D. Jost, J. P. Home, D. Leibfried, and D. J. Wineland. *Coherent Diabatic Ion Transport and Separation in a Multizone Trap Array*. Phys. Rev. Lett. **109**, 080502 (2012)

[93] H. Kaufmann, T. Ruster, C. T. Schmiegelow, F. Schmidt-Kaler, and U. G. Poschinger. *Dynamics and control of fast ion crystal splitting in segmented Paul traps*. New Journal of Physics **16, 7**, 073012 (2014)

[94] T. Ruster, C. Warschburger, H. Kaufmann, C. T. Schmiegelow, A. Walther, M. Hettrich, A. Pfister, V. Kaushal, F. Schmidt-Kaler, and U. G. Poschinger. *Experimental realization of fast ion separation in segmented Paul traps*. Phys. Rev. A **90**, 033410 (2014)

[95] H. Kaufmann, T. Ruster, C. T. Schmiegelow, M. A. Luda, V. Kaushal, J. Schulz, D. von Lindenfels, F. Schmidt-Kaler, and U. G. Poschinger. *Fast ion swapping for quantum-information processing*. Phys. Rev. A **95**, 052319 (2017)

[96] D. Wessel. *Techniques for Conditional Quantum Logic and Improved Detection for Trapped-Ion Qubits*. Master thesis, Goethe-Universität Frankfurt (2020)

[97] M. Rowej, A. Ben-Kish, B. Demarco, D. Leibfried, V. Meyer, J. Beall, J. Britton, J. Hughes, W. Itano, et al. *Transport of Quantum States and Separation of Ions in a Dual RF Ion Trap*. Quantum Information & Computation **2**, 257–271 (2002)

[98] H.-K. Lau and D. F. V. James. *Decoherence and dephasing errors caused by the dc Stark effect in rapid ion transport*. Phys. Rev. A **83**, 062330 (2011)

[99] A. Steane. *A Tutorial on Quantum Error Correction*. Proceedings of the International School of Physics "Enrico Fermi" **162** (2006)

[100] J. Roffe. *Quantum error correction: an introductory guide*. Contemporary Physics **60, 3**, 226–245 (2019)

[101] W. K. Wootters and W. H. Zurek. *A single quantum cannot be cloned*. Nature **299, 5886**, 802–803 (1982)

[102] D. Gottesman. *Stabilizer Codes and Quantum Error Correction*. Ph.d. thesis, arXiv:9705052 (1997)

[103] D. Gottesman. *Class of quantum error-correcting codes saturating the quantum Hamming bound*. Phys. Rev. A **54**, 1862–1868 (1996)

[104] A. R. Calderbank, E. M. Rains, P. W. Shor, and N. J. A. Sloane. *Quantum Error Correction and Orthogonal Geometry*. Phys. Rev. Lett. **78**, 405–408 (1997)

[105] H. Bombin and M. A. Martin-Delgado. *Topological Quantum Distillation*. Phys. Rev. Lett. **97**, 180501 (2006)

[106] L. Lao and C. G. Almudever. *Fault-tolerant quantum error correction on near-term quantum processors using flag and bridge qubits*. Phys. Rev. A **101**, 032333 (2020)

[107] A. Bermudez, X. Xu, R. Nigmatullin, J. O'Gorman, V. Negnevitsky, P. Schindler, T. Monz, U. G. Poschinger, C. Hempel, et al. *Assessing the Progress of Trapped-Ion Processors Towards Fault-Tolerant Quantum Computation*. Phys. Rev. X **7**, 041061 (2017)

[108] A. Rodriguez-Blanco, A. Bermudez, M. Müller, and F. Shahandeh. *Efficient and Robust Certification of Genuine Multipartite Entanglement in Noisy Quantum Error Correction Circuits*. PRX Quantum **2**, 020304 (2021)

[109] D. Nigg, M. Müller, E. A. Martinez, P. Schindler, M. Hennrich, T. Monz, M. A. Martin-Delgado, and R. Blatt. *Quantum computations on a topologically encoded qubit*. Science **345, 6194**, 302–305 (2014)

[110] M. Müller, A. Rivas, E. A. Martínez, D. Nigg, P. Schindler, T. Monz, R. Blatt, and M. A. Martin-Delgado. *Iterative Phase Optimization of Elementary Quantum Error Correcting Codes*. Phys. Rev. X **6**, 031030 (2016)

[111] P. Schindler, J. T. Barreiro, T. Monz, V. Nebendahl, D. Nigg, M. Chwalla, M. Hennrich, and R. Blatt. *Experimental Repetitive Quantum Error Correction*. Science **332, 6033**, 1059 (2011)

[112] D. Kielpinski, A. Ben-Kish, J. Britton, V. Meyer, M. Rowe, W. Itano, D. Wineland, C. Sackett, and C. Monroe. *Recent results in trapped-ion quantum computing at NIST*. Quantum Info. Comput. **1**, 113 (2001)

[113] J. Chiaverini, D. Leibfried, T. Schaetz, M. D. Barrett, R. B. Blakestad, J. Britton, W. M. Itano, J. D. Jost, E. Knill, et al. *Realization of quantum error correction*. Nature **432**, 602 (2004)

[114] V. Negnevitsky, M. Marinelli, K. K. Mehta, H.-Y. Lo, C. Flühmann, and J. P. Home. *Repeated multi-qubit readout and feedback with a mixed-species trapped-ion register*. Nature **563, 7732**, 527–531 (2018)

[115] R. Stricker, D. Vodola, A. Erhard, L. Postler, M. Meth, M. Ringbauer, P. Schindler, T. Monz, M. Müller, and R. Blatt. *Experimental deterministic correction of qubit loss*. Nature **585**, 207 (2020)

[116] J. Kelly, R. Barends, A. G. Fowler, A. Megrant, E. Jeffrey, T. C. White, D. Sank, J. Y. Mutus, B. Campbell, et al. *State preservation by repetitive error detection in a superconducting quantum circuit*. Nature **519, 7541**, 66–69 (2015)

[117] N. Ofek, A. Petrenko, R. Heeres, P. Reinhold, Z. Leghtas, B. Vlastakis, Y. Liu, L. Frunzio, S. Girvin, et al. *Extending the lifetime of a quantum bit with error correction in superconducting circuits*. Nature **536**, 441 (2016)

[118] C. K. Andersen, A. Remm, S. Lazar, S. Krinner, N. Lacroix, G. J. Norris, M. Gabureac, C. Eichler, and A. Wallraff. *Repeated quantum error detection in a surface code*. Nature Physics **16, 8**, 875–880 (2020)

[119] Z. Chen, K. J. Satzinger, J. Atalaya, et al. *Exponential suppression of bit or phase flip errors with repetitive error correction*, `arXiv:2102.06132` (2021)

[120] J. Zhang, R. Laflamme, and D. Suter. *Experimental Implementation of Encoded Logical Qubit Operations in a Perfect Quantum Error Correcting Code*. Phys. Rev. Lett. **109**, 100503 (2012)

[121] E. Knill, R. Laflamme, R. Martinez, and C. Negrevergne. *Benchmarking Quantum Computers: The Five-Qubit Error Correction Code*. Phys. Rev. Lett. **86, 18**, 5811 (2001)

[122] G. Waldherr, Y. Wang, S. Zaiser, M. Jamali, T. Schulte-Herbrüggen, H. Abe, T. Ohshima, J. Isoya, J. F. Du, et al. *Quantum error correction in a solid-state hybrid spin register*. Nature **506, 7487**, 204 (2014)

[123] T. Unden, P. Balasubramanian, D. Louzon, Y. Vinkler, M. B. Plenio, M. Markham, D. Twitchen, A. Stacey, I. Lovchinsky, et al. *Quantum Metrology Enhanced by Repetitive Quantum Error Correction*. Physical Review Letters **116**, 230502 (2016)

[124] M. Takita, A. W. Cross, A. D. Córcoles, J. M. Chow, and J. M. Gambetta. *Experimental Demonstration of Fault-Tolerant State Preparation with Superconducting Qubits*. Phys. Rev. Lett. **119**, 180501 (2017)

[125] N. M. Linke, M. Gutierrez, K. A. Landsman, C. Figgatt, S. Debnath, K. R. Brown, and C. Monroe. *Fault-tolerant quantum error detection*. Science Advances **3, 10** (2017)

[126] L. Egan, D. M. Debroy, C. Noel, A. Risinger, D. Zhu, D. Biswas, M. Newman, M. Li, K. R. Brown, et al. *Fault-Tolerant Operation of a Quantum Error-Correction Code*, `arXiv:2009.11482` (2021)

[127] M. Brownnutt, M. Kumph, P. Rabl, and R. Blatt. *Ion-trap measurements of electric-field noise near surfaces*. Rev. Mod. Phys. **87**, 1419–1482 (2015)

[128] C. J. Ballance, V. M. Schäfer, J. P. Home, D. J. Szwer, S. C. Webster, D. T. C. Allcock, N. M. Linke, T. P. Harty, D. P. L. Aude Craik, et al. *Hybrid quantum logic and a test of Bell's inequality using two different atomic isotopes*. Nature **528, 7582**, 384–386 (2015)

[129] G. Tóth and O. Gühne. *Entanglement detection in the stabilizer formalism*. Phys. Rev. A **72**, 022340 (2005)

[130] G. Tóth and O. Gühne. *Detecting Genuine Multipartite Entanglement with Two Local Measurements*. Phys. Rev. Lett. **94**, 060501 (2005)

[131] L. Stephenson, D. Nadlinger, B. Nichol, S. An, P. Drmota, T. Ballance, K. Thirumalai, J. Goodwin, D. Lucas, and C. Ballance. *High-Rate, High-Fidelity Entanglement of Qubits Across an Elementary Quantum Network*. Physical Review Letters **124, 11** (2020)

Publications and Conference Contributions

Publications

- **Fault-tolerant parity readout on a shuttling-based trapped-ion quantum computer**
  J.Hilder, D. Pijn, O. Onishchenko, A. Stahl, M. Orth, B. Lekitsch, A. Rodriguez-Blanco, M. Müller, F. Schmidt-Kaler, U. Poschinger,
  arXiv:2107.06368 (Phys. Rev. X (2022) in print)

- **Detecting heat leaks with trapped ion qubits**
  D. Pijn, O. Onishchenko, J. Hilder, U. Poschinger, F. Schmidt-Kaler, R. Uzdin,
  arXiv:2110.03277 (Phys. Rev. Lett. (2022) accepted)

- **Shuttling-Based Trapped-Ion Quantum Information Processing**
  V. Kaushal, B. Lekitsch, A. Stahl, J. Hilder, D. Pijn, C. Schmiegelow, A. Bermudez, M. Müller, F. Schmidt-Kaler, U. Poschinger,
  AVS Quantum Sci. 2, 014101 (2020)

Conference Contributions

- **Automatisierte Positionskontrolle von Ionen in einer segmentierten Paulfalle**
  J. Nicodemus, T. Ruster, V. Kaushal, D. Pijn, B. Lekitsch, U. Poschinger und F. Schmidt-Kaler,
  Talk at the DPG-Frühjahrstagung 2018, Friedrich-Alexander-Universität Erlangen-Nürnberg, Germany

- **A shuttling-based trapped-ion quantum information processing node**
  J. Hilder, V. Kaushal, D. Pijn, A. Stahl, T. Ruster, B. Lekitsch, U. Poschinger and F. Schmidt-Kaler,
  Poster at the European Conference on Trapped Ions (ECTI) 2018, Weizmann Institute of Science, Israel

- **Ein Quantenprozessor mit Ionenkristallen**
  J. Hilder, D. Pijn, V. Kaushal, A. Stahl, B. Lekitsch, U. Poschinger und F. Schmidt-Kaler,
  Talk at the DPG-Frühjahrstagung 2019, Universität Rostock, Germany

- **A shuttling-based trapped-ion quantum information processing node**
  J. Hilder, D. Pijn, A. Stahl, O. Onishchenko, B. Lekitsch, U. Poschinger and F. Schmidt-Kaler,
  Poster at the 737. WE-Heraeus-Seminar on 'Advances in Scalable Hardware Platforms for Quantum Computing', Physikzentrum Bad Honnef, Germany

- **Fault-tolerant parity readout on a trapped-ion quantum information processing node**
  J. Hilder, D. Pijn, A. Stahl, O. Onishchenko, B. Lekitsch, U. Poschinger and F. Schmidt-Kaler,
  Poster at the Munich Conference on Quantum Science and Technology (MCQST) 2021, Munich Center for Quantum Science and Technology (online), Germany

- **Fault-tolerant parity readout on a trapped-ion quantum information processing node**
  J. Hilder, D. Pijn, O. Onishchenko, A. Stahl, M. Orth, B. Lekitsch, A. Rodriguez-Blanco, M. Müller, F. Schmidt-Kaler, U. G. Poschinger,
  Talk at the European Conference on Trapped Ions (ECTI) 2021, hybrid satellite meeting at University of Mainz, Germany

- **Fault-tolerant parity readout on a trapped-ion quantum information processing node**
  J. Hilder, D. Pijn, O. Onishchenko, A. Stahl, M. Orth, B. Lekitsch, A. Rodriguez-Blanco, M. Müller, F. Schmidt-Kaler, U. G. Poschinger,
  Talk at the European Quantum Technologies Conference (EQTC) 2021, Irish Centre for High-End Computing and the National University of Ireland (online), Ireland