Efficient Qubit Routing for a Globally Connected Trapped Ion Quantum Computer

*Mark Webber, Steven Herbert, Sebastian Weidt, and Winfried K. Hensinger**

The cost of enabling connectivity in noisy intermediate-scale quantum (NISQ) devices is an important factor in determining computational power. A qubit routing algorithm is created, which enables efficient global connectivity in a previously proposed trapped ion quantum computing architecture. The routing algorithm is characterized by comparison against both a strict lower bound, and a positional swap based routing algorithm. An error model is proposed, which can be used to estimate the achievable circuit depth and quantum volume of the device as a function of experimental parameters. A new metric based on quantum volume, but with native two-qubit gates, is used to assess the cost of connectivity relative to the upper bound of free, all to all connectivity. The metric is also used to assess a square-grid superconducting device. These two architectures are compared and it is found that for the shuttling parameters used, the trapped ion design has a substantially lower cost associated with connectivity.

1. Introduction

Quantum computers are expected to solve classically intractable problems, such as accurately simulating the dynamics of large molecules,[1,2] which would greatly impact both material science and the pharmaceutical industry. In the finance industry, even minor advantages can lead to significant returns.[3] Phase estimation[4] (for quantum chemistry) and Shor's algorithm[5] (for breaking RSA encryption) are two algorithms which promise an exponential speed up,[6,7] but they both require a fault tolerant device for useful applications. Error correction techniques, such as the surface code,[8–10] which facilitate a fault tolerant device have a very large physical to logical qubit overhead, requiring physical qubit numbers in the range of 10^5–10^8.

In recent years, there has been growing interest and algorithmic development for noisy intermediate-scale quantum (NISQ) computers[11] which do not require fault tolerance. The realization of "quantum supremacy"[12] represents a major milestone for such systems. Hybrid quantum algorithms, such as the variational quantum eigensolver,[13,14] may provide an exponential speed up as compared to the classical counterparts. Assessing the capability of NISQ devices to run quantum algorithms is quite distinct from that of full-scale error-corrected quantum computers. For NISQ devices, this typically involves quantifying the achievable circuit depth of the device which represents the number of sequential gate operations that can be executed within the available coherence time. The analysis can be done in reverse to instead tailor near term algorithms to a specific device.

Superconducting circuits[15] and trapped ion devices[16] are two of the leading quantum computing platforms. In particular, one architecture which offers a scalable approach to trapped ion quantum computing is based on a large connected ion trap array.[17,18] This provides a solution to scale to very large qubit numbers, which will be a requirement to run many important algorithms. A key component of this architecture is the shuttling of individual ions to enable connectivity.[19–21]

To utilize this shuttling-based trapped ion architecture, it is necessary to have a routing algorithm which can move large numbers of ions across the square grid array in parallel, and in an efficient manner. In this manuscript, we provide an ion routing algorithm which can enable arbitrary global connectivity, and we quantify its efficiency relative to a lower bound. When considering the achievable circuit depth of a NISQ device, one must include factors such as connectivity, gate fidelity, and the coherence time of the qubit. We provide an error model which can be used to estimate the achievable circuit depth of this quantum computing design as a function of experimental parameters.

Quantum computing architectures vary greatly, from the underlying system which represents the qubit, the available quantum gate set, and to the means by which qubit connectivity is enabled. For superconducting architectures, qubits are stationary and connectivity is enabled through sequences of

M. Webber, Dr. S. Weidt, Prof. W. K. Hensinger
Sussex Centre for Quantum Technologies
Department of Physics and Astronomy
University of Sussex
Brighton BN1 9QH, UK
E-mail: W.K.Hensinger@sussex.ac.uk

Dr. S. Weidt, Prof. W. K. Hensinger
Universal Quantum Ltd
Brighton BN1 6SB UK

Dr. S. Herbert
Department of Computer Science
University of Oxford
OX1 3QD, UK

The ORCID identification number(s) for the author(s) of this article can be found under https://doi.org/10.1002/qute.202000027

DOI: 10.1002/qute.202000027

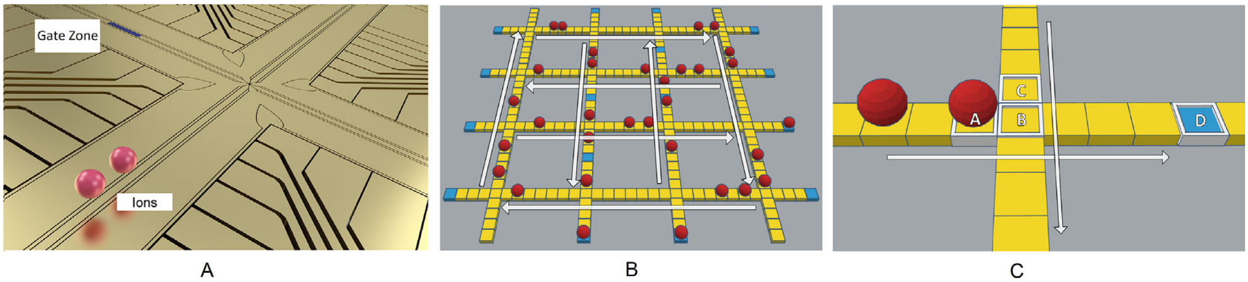

Figure 1. A) A depiction of a single X junction which is repeated to form a grid on which the ions are restricted to, with zones dedicated to specific tasks. B) A 3D representation of a quantum computing device using our proposed routing algorithm, where the yellow grid represents the X-Junctions, which the ions (red spheres) are restricted to, and the blue squares represent gate zones. The digitization of the simulation can be seen with a resolution of seven positions between adjacent X-Junctions. Arrows represent the lane priority of the routing algorithm. C) A close up of an X-Junction from (B). The routing logic used to decongest X-Junction centers involves occasionally ignoring the lane priority. Ions assigned to interior gate zones (blue square labeled D) have the closest X-Junction center (labeled B) as their destination (one space off the center because it is an area of lower trap stability (labeled A and C)). The ion in square A has been assigned to the local gate zone and it will travel back and forth between positions A and C directly, by ignoring the lane priority, to decongest for ions still travelling to their gate zone.

swap gates via nearest neighbor interactions, which will incur a high gate overhead for globally connected algorithms. For square grids with nearest neighbor connectivity, the best known method for globally connected algorithms on N qubit scales with an overhead of $\Theta(N^{0.5})$,[22] although it is only logarithmic if non-planar architectures are considered[23,24] and optimisation of this swapping procedure is necessary to maximize performance.[25,26] The characteristics of the desired algorithm will dictate the degree to which a device with inherent all-to-all connectivity outperforms a device which has a cost associated with enabling connectivity.[27] The way in which connectivity is enabled varies greatly even within the trapped ion architectures. Architectures with stationary ions confined to a linear string benefit from global connectivity and multi-qubit gates[28,29]; however, as the number of ions co-existing in a single trap increases, it becomes progressively challenging to maintain key device specifications, such as gate fidelity. Furthermore, as the ion number N increases, gate times increase as $\sqrt{N}$, and the increasing requirement on the number of motional modes will eventually lead to frequency crowding.[30] Shuttling and swap operations may instead be used to enable connectivity by positioning multiple ions into the same region of space, where local gates may be performed. This introduces different challenges, but all required register reconfiguration operations have been demonstrated and several groups are further improving on aspects such as speed and reliability. There are two main approaches to enable connectivity between trapped ion modules, one involves the use of photonic interconnects,[31,32] while the other, as described in the architecture analyzed in this manuscript, utilizes electric fields to connect adjacent modules. The connected modules form a continuous 2D plane, resulting in connection speeds between modules orders of magnitude higher as compared to photonic interconnects.

The quantum computing architecture investigated in this manuscript consists of an ion trap array on a microchip, giving rise to a 2D grid to which all ions are restricted. The ions (where each ion represents a physical qubit) do not have to be stationary and are instead able to traverse the grid via shuttling operations. Entangling operations are performed by bringing the two (or more) desired ions to the same region of space (a gate zone). The smallest repeated unit of the architecture is the X-Junction (see **Figure 1**A). Logic gates may be performed by applying static voltages to a microchip in the presence of globally applied microwave fields and a local magnetic field gradient.[33] An alternative approach instead makes use of pairs of laser beams to execute quantum gates,[34] but this may be more challenging to implement for large numbers of qubits. This electronic microwave-based architecture has a clear path toward scaling to large qubit numbers,[18] and constitutes a practical blueprint for a quantum computer capable of solving some of the hardest problems, such as breaking RSA encryption. Furthermore, arbitrary two-qubit connectivity can be enabled in near-term devices relying only on ion shuttling operations (which can have a state fidelity comparable to stationary trapped ions[35]), without sequences of swap gates, as may be required in other architectures. To run an algorithm on a quantum computer based on this design, one first needs a routing algorithm which efficiently enables arbitrary connectivity between the ions in the square grid device, which is the main challenge addressed in this manuscript. The relevance of this manuscript is independent of the specific choice of gate operation, ion species, and transition. Finding the optimum instruction set for each individual ion in real time is intractable and so we have solved the problem in a heuristic manner. The solution is motivated by one-way traffic flow with additional rule sets to deal with junction centers more efficiently. We quantify the efficiency of our approach relative to an unattainable lower bound and investigate its flexibility with regards to device shape and ion density. We use these results, in combination with an error model we propose, to investigate the achievable depth and quantum volume for this design as a function of experimental parameters. We have made the error model publicly available.[36] Quantum volume (QV) is a conceived metric for quantum computational power designed to enable sincere comparison between architectures,[37] and we will discuss it in more detail in Section 3.2.

We have developed a simulation tool for the previously proposed architectural design of Lekitsch et al.[18] The simulation tool was used to develop and assess routing algorithms. The remainder of this manuscript is organized as follows: In Section 2, we start by specifying the architecture and the connectivity problem to be solved, and then go on to explain the simulation tool and the developed routing algorithm. In Section 3.1, we quantify the efficiency and versatility of our routing algorithm and in Section 3.2, we present results on the achievable depth and quantum volume as a function of experimental parameters.

2. Problem Specification and Routing Algorithm

In the design being investigated here, ions (each encoding a single qubit) are restricted to a square grid (see Figure 1B) which consists of an array of repeated X-Junctions (see Figure 1A), each containing a single gate zone. Ions must first be shuttled (physically moved) into the gate zones for gates to be performed. The X-Junctions have a defined spacial resolution, which arises from the fixed number of electrodes on each arm but ions may be moved continuously. The gate zones enable both single and two-qubit gates. To perform a quantum algorithm on this device, it must first be decomposed into the native gate set, which can be optimized.[38] A decomposed quantum algorithm is defined by multiple rounds of gates; ideally, all the required gates of an individual round will be applied in parallel; however, the qubit number, gate density of the algorithm, and the number of gate zones will dictate the gate round overhead. In this architecture, each gate round is further broken into two parts, a routing sequence, where ions are shuttled into gate zones, which is then followed by the application of gates. We use the terminology “shuttling” to refer to the act of moving ions in the device, and “routing” to refer to the higher order logic of the shuttling. In this design, gates cannot be applied concurrently with shuttling. When the required number of gates in an individual round exceeds the number of available gate zones, it is necessary to have multiple rounds of shuttling and gates; for example, a gate round overhead of 2 would imply the need for: shuttle, apply gates, shuttle, apply gates. The shuttling round, which enables the connectivity, is the focus of this manuscript. When designing the routing algorithm, we optimized for the total time taken to enable global connectivity.

To enable arbitrary connectivity for this quantum computing architecture, we have created a simulation tool[36] which represents the devices as a square grid consisting of iterated X-Junctions (see Figure 1B). The simulation is digitized to a variable resolution, where each position may contain 0, 1, or 2 ions. The ions are distributed evenly across the grid near the center of each X-Junction and a quantum circuit (list of required two-qubit gates, i.e., ions that must be connected) is inputted. Ions which are assigned to the same gate zone are able to combine as a pair. Naive routing algorithms would not converge on a solution, as ions with opposite traveling directions meet and cause permanent blockages. Positional swaps between ions have been demonstrated experimentally[39] and their usage would simplify the required routing algorithm. Here, we present a solution that does not use swaps, and in Section 3, we compare the effectiveness of routing both with and without swap operations. When bench-marking the device, a randomly generated, globally connected circuit was used. In order to assign ion (qubit) pairs to gate zones, we employ a greedy approach, assigning each pair to the nearest available gate zone (i.e., minimum combined distance of travel for the two ions), and addressing the pairs in an arbitrary order. This greedy approach is sufficient for a proof of principle using this prototype ion-routing algorithm; however, we note that it may not yield the optimum gate-zone designations overall. To this end, a more sophisticated optimisation may be considered in future work, but we note that such combinatorial optimizations are generally hard problems themselves.

At each time step in the simulation, each ion is evaluated and moved sequentially according to the routing algorithm, which involves assessing its location, local environment, and destination. The routing algorithm we have developed assigns alternating direction priorities to each lane of the square grid. The top-most horizontal lane is a right-only lane, the lane below it is left-only, and so on, and this also applies to vertical lanes (see Figure 1B). We ensure that the outer perimeter of the device is a clockwise loop regardless of the number of lanes, so that all gate zones can be reached, which means that odd size devices, for example, one which consists of 3 x 3 X-Junctions, will not have fully alternating lane directions and instead will have, right, left, left, and up, down, down. We define a square grid device formed from M x M X-Junctions to be of device size M. We preferentially position gate zones on the exterior of the device where possible (on the outer arms of the perimeter X-junctions). Exterior gate zones are more favorable for routing as waiting ions do not block the movement of other ions. For square devices, the number of interior gate zones scale with device size as $(M-2)^2$ and the exterior gate zones scale as $4M-4$, which results in a cross over point at device size 7 (98 qubits at 2 per X-Junction).

The centers of X-Junctions are decision points, where an ion will follow the lane priority toward its destination. Ions can enter the outer arms into the exterior gate zones only when it allows them to reach their assigned destination. Ions which are not destined to a gate zone during a given shuttling round have their destination set to their current location, and therefore only move to decongest. During development of the routing algorithm, a major bottleneck identified was congestion at interior gate zones. Devices larger than 2 x 2 have interior gate zones, and the ions waiting there can cause permanent blockages or unnecessary movement depending on how they are handled. To remedy this problem, an additional feature was included, in which ions assigned to interior gate zones wait at the closest available X-Junction center, where they are able to decongest efficiently by temporarily ignoring the lane priority (see Figure 1C). The movements available to each ion are dependent on multiple assigned parameters. The following binary questions determine these parameters: Does this ion have a destination for this round of gates? Is this a single ion or pair? Is this a waiting ion assigned to interior X-Junctions? The valid moves are then determined by using these parameters in combination with the location and local environment of the ion. At any particular time step, an ion may have multiple valid moves available to it; hence, there is a hierarchical list as follows from first priority to last: combine as a valid pair, a lane priority ignoring move as a waiting ion, a lane priority following move. Ions with no valid moves available will wait until the next time step.

3. Results

In Section 3.1 we assess the efficiency and versatility of our routing algorithm. In Section 3.2 we present an error model

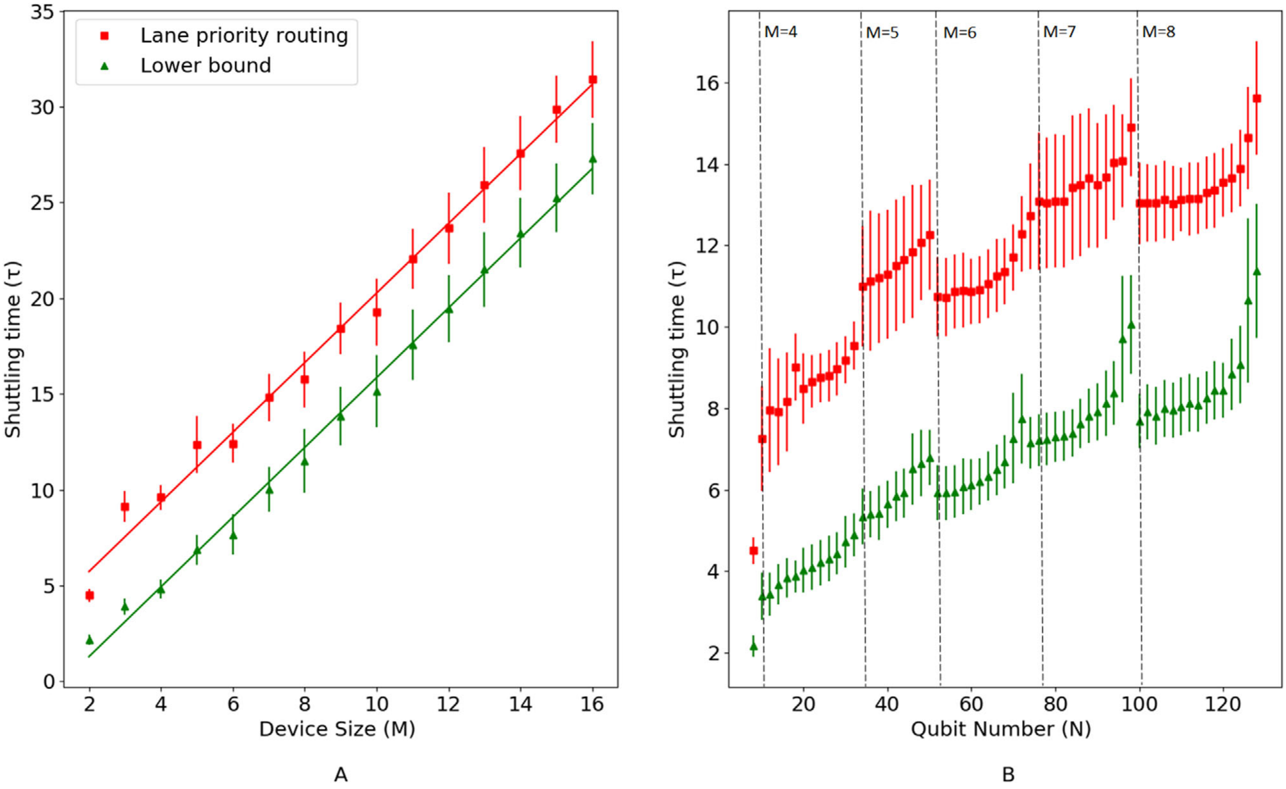

Figure 2. A) Shuttling time, τ (scaled by the resolution of the model so that a time of 1 is equal to the time it takes to shuttle between two adjacent X-Junctions) taken to enable connectivity as a function of device size (defined as a square grid consisting of M x M X-Junctions where M is the device size). There are two qubits initialized per X-Junction (plotted here for a range of 8 to 512 qubits). Red squares: Shuttling time for the routing algorithm. Green triangles: The lower bound (shortest route) shuttling time. The trend lines were generated using linear regression analysis and they both have a gradient of 1.82. Vertical lines represent one standard deviation. The results are the average value over 300 iterations of randomly selected pairings. The iteration number was chosen after investigating the mean and standard deviation convergence rate. B) Shuttling time as a function of qubit number. The device size increases with qubit number when the device can no longer accommodate two qubits per X-Junction. Red squares: Shuttling time for our routing algorithm. Green triangles: The lower bound (shortest route) shuttling time. Vertical lines represent one standard deviation and the dashed lines mark where the device size is increased to accommodate the additional qubits

which utilizes the routing algorithm and includes experimental parameters such as gate fidelities, coherence time, ion loss and shuttling speed. We then use this error model to estimate the achievable depth and quantum volume of quantum computers based on this architecture.

3.1. Assessing the Routing Algorithm

In this section, we characterize the performance and flexibility of our routing algorithm, which we refer to as lane priority routing, within the framework of our abstract simulation tool. In Section 3.2, we will introduce more practical considerations, allowing us to quantify the expected fidelity associated with enabling connectivity. Randomly generated depth 1 circuits on N qubits consisting of $N/2$ two-qubit gates were iterated sufficiently to represent the requirement of global connectivity. After each iteration, we count the total number of time steps which were required (τ), which can be converted into a total time (s) by considering the estimated speed at which one can shuttle between adjacent X-Junctions. It is important to note that increasing the speed of an individual shuttling operation may not always lead to an increase in the final fidelity, as the quality of the shuttling operation may impact on subsequent operations. At each iteration, a lower bound is calculated for that particular set of pairings, which is equal to the minimum number of time steps that will enable connectivity. To calculate the lower bound, it is assumed that qubits (ions) take the shortest path toward their destination and swap with no time penalty (i.e., the time required for an ion to move one discrete step is independent of whether a swap is performed or not). For a particular iteration, the ion with the greatest distance to travel is identified, and the number of spacial steps between its starting location and its destination is equal to the lower bound.

The average shuttling time required to enable the global connectivity can then be compared to the lower bound as shown in **Figure 2**A. These results are for devices with perfect two-qubit gate parallelizability, that is, there are two qubits initialized per X Junction. We conjecture that the total shuttling time would at best scale linearly with device size, M, because randomly selected distances in a square scales linearly with the length of the square. Both our routing procedure and the lower bound scale linearly with the device size and with a gradient of 1.82. There is a constant overhead associated with our routing which becomes

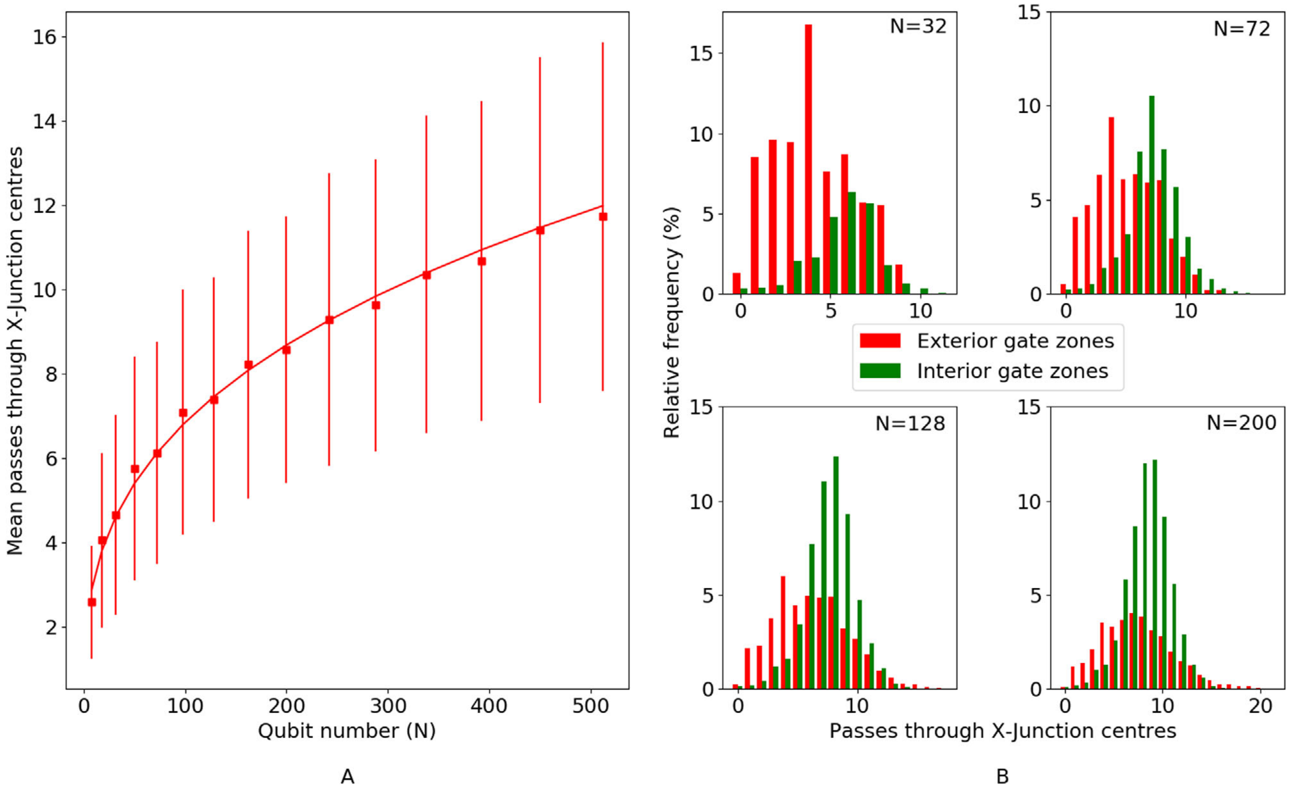

Figure 3. A) The mean number of passes through an X-Junction center per ion as a function of qubit number for square devices with two qubits per X-Junction. Vertical lines represent a single standard deviation. B) The relative frequency distribution of passes through X-Junction centers for four different device sizes, 4 × 4 ($N = 32$), 6 × 6 ($N = 72$), 8 × 8 ($N = 128$), and 10 × 10 ($N = 200$). Red bars: Qubits assigned to exterior gate zones. Green bars: Qubits assigned to interior gate zones. 300 iterations of the globally connected depth-1 algorithm were used to generate a representative sample, and the frequency is scaled accordingly.

less significant the larger the device is. The scaling for total shuttling time, τ, as a function of qubits, N, where $N = 2 \times M^2$ is $\tau = 1.3\,(3)\,\sqrt{N} + 2\,(5)$, the fit and standard error were calculated using linear regression. An oscillating pattern on the lane priority routing results is noticeable with its relative magnitude decreasing with device size, which results from even-sized devices outperforming odd-sized devices. Odd-sized devices (e.g., a device of 3 × 3 X-Junctions) cannot fully realize the alternating lane priority because we ensure that the outer perimeter lane is always a clockwise path.

The routing algorithm is flexible and works well for a wide range of qubit numbers for a given device size. Figure 2B shows the shuttling dependence on qubit number for qubit densities less than or equal to 2 per X-Junction, that is, with full gate parallelizability. The oscillating pattern resulting from odd and even device sizes is more notable. Shuttling time increases for both the lane priority routing and lower bound as more qubits are added to a device of static size, and peaks at a density of two qubits per X-Junction.

The main criteria we optimized for when creating the routing algorithm was the total time. To calculate the achievable circuit depth at which a device can run, the total error will not just be a function of the total time, but also include factors such as gate fidelity and ion loss. Traversing an X-Junction will have a corresponding ion loss rate which may be higher than the loss associated with linear shuttling. In order to quantify the associated error, we have used our simulation to count the number of times qubits are expected to move through an X-Junction center. The implications of these results for achievable depth will be explored in the following section. In **Figure 3A**, the mean number of passes through an X-Junction, X_{count}, is plotted as a function of qubit number with vertical lines corresponding to a single standard deviation, and the dependence is well described by the following equation, $X_{\text{count}} = 0.4\,(1)\,\sqrt{N} + 2\,(2)$. The distribution of passes is investigated in Figure 3B for four different device sizes, 4, 6, 8, and 10. The qubits are separated into two data sets, according to whether they are assigned to an interior or exterior gate zone. Across all device sizes investigated, the maximum passes did not exceed 4x the stated mean. For the device with 72 qubits investigated in Figure 3B, the probability of an individual ion passing through an X-Junction center ⩾14 times is low, at approximately 0.2%.

It may be desirable to increase the qubit density beyond 2 per X-Junction despite the potential loss of gate parallelizability, as additional X-Junctions are experimentally costly to implement. **Figure 4A** shows the efficiency of the routing protocol for three different qubit densities. The increase in shuttling time is predominantly attributed to the multiple rounds of shuttling (and gates) which are required for the 100% gate density (where gate density is the percentage of qubits involved in gates per time step)

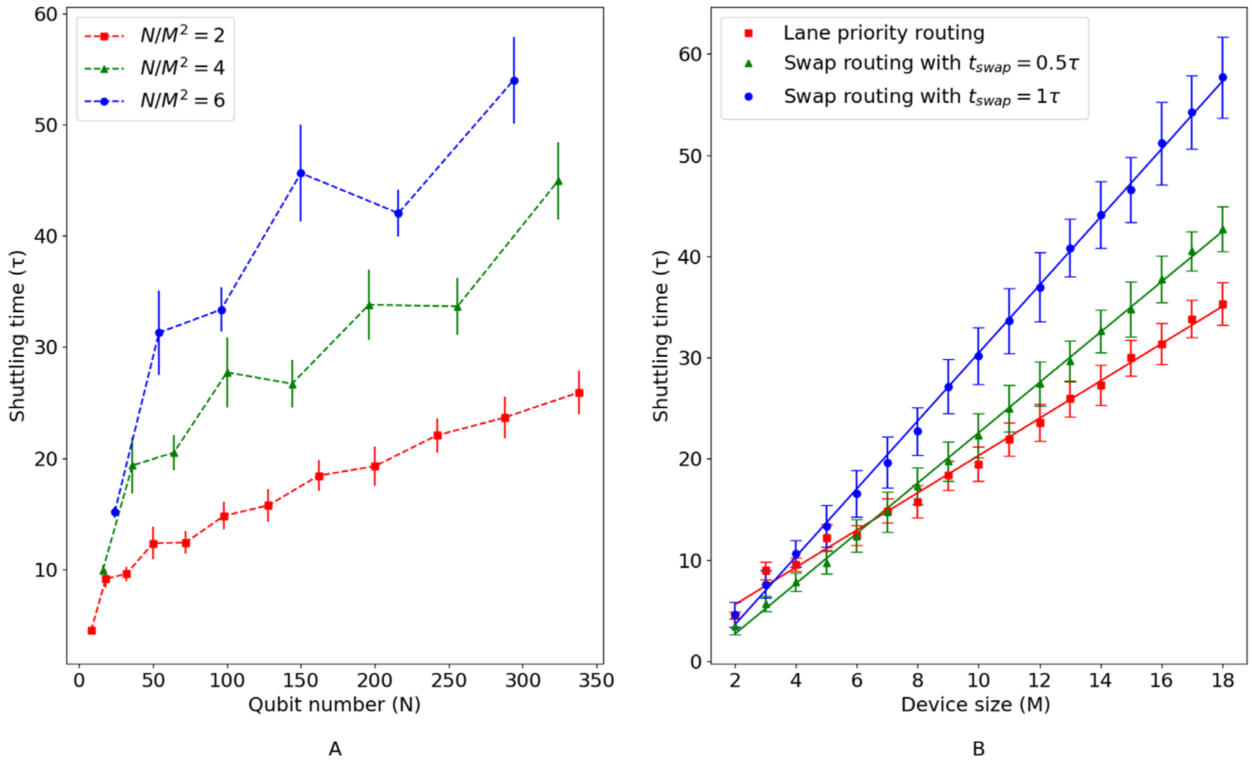

Figure 4. A) Required shuttling time as a function of qubit number for three different qubit densities with red squares: 2 qubits per X-Junction; green triangles: 4 qubits per X-Junction; blue circles: 6 qubits per X-Junction. Vertical lines represent one standard deviation. B) Required shuttling time as a function of device size (defined as a square grid consisting of M by M X-Junctions where M is device size), with 2 ions per X-Junction, comparing swap-based routing to our lane priority routing algorithm. Red squares: Lane priority routing. Green Triangles: Swap-based routing with a swap time penalty equivalent to half the time it takes to shuttle between two adjacent X-Junctions. Blue Circles: Swap-based routing with a swap time penalty equivalent to the time it takes to shuttle between two adjacent X-Junctions. Vertical lines represent one standard deviation and trend lines are fit using linear regression.

algorithm which we are assessing against. With a density of four qubits per X-Junction, a 100% two-qubit gate density algorithm would be completed by two full rounds of shuttling and gate applications. The oscillating pattern attributed to odd and even devices becomes more apparent with increasing qubit density. This analysis only includes the additional time associated with the multiple rounds of shuttling and does not include the gate time. The overall cost of increasing qubit density will depend on the gate density of the desired algorithm.

We created a new routing algorithm which relies on positional swaps where qubits take the shortest available route (ignoring the previously mentioned lane priority routing) and swap to decongest. We have compared the total shuttling time of the swap routing against the lane priority routing, for two different swap time penalties, shown in Figure 4B. The time penalties were chosen based on early experimental results. H, Kaufmann et al. demonstrated fast ion swapping of 42 µs at a process fidelity of 99.5%.[39] Van Mourik et al. demonstrated positional ion swapping with an associated coherence loss of 0.2(2)%.[40] For ion shuttling speed, Walther et al. demonstrated fast shuttling of cold ions, over a distance of 280 µm in 3.6 µs[41] and P, Kaufmann et al. demonstrated a state fidelity of 99.9994%, for shuttling over a distance of 280 µm in 12.8 µs.[35] We characterized the average number of swaps, n_{swap}, per qubit for each connectivity run and found that for 18 qubits; the average was 1 swap, and for 50 qubits, the average was 1.7 swaps. The dependence was well described by the following equation: $n_{\text{swap}} = 0.23(2)\sqrt{N} + 0.1(2)$, where the fit and standard error were calculated using linear regression. The average number of swaps per ion which was required to enable connectivity was found to be only weakly dependent on the swap time cost penalty; therefore, doubling the time penalty results in minimal change to the number of swaps. For a wide range of device sizes, the lane priority routing outperforms the swap-based routing on total time taken, for the swap time penalties used here. This analysis suggests that for efficient routing in this 2D trap design, it will not be necessary to perform positional swap operations. However, some combination of the lane priority routing and positional swaps may be favorable, depending on the expected costs associated with these operations. In the following section, we will bring in more practical considerations to define the expected fidelity of a globally connected algorithm using our lane priority routing algorithm.

3.2. Achievable Depth and Quantum Volume Estimations

For comparison between near-term quantum computers, one must consider more than just the number of qubits. Quantum volume (QV) is a conceived metric for quantum computational power designed to enable sincere comparison between architectures.[37] QV includes factors such as gate fidelity, qubit number, connectivity, and the available gate set, and is given by

$$\mathrm{QV} = \underset{N}{\operatorname{argmax}} \left[\min \left[N, \frac{1}{N \times \epsilon_{\mathrm{eff}}(N)} \right] \right]^2 \tag{1}$$

for the number of qubits within the device N, and effective error rate ϵ_{eff}, which typically depends on N. QV reflects the limiting factor of the device, which is either the qubit number or the achievable depth D, where $D = 1/(N \times \epsilon_{\mathrm{eff}}(N))$. To compute QV, a randomly generated depth-1 circuit on N qubits with $N/2$ arbitrary (SU(4)) two-qubit gates is used. The achievable depth represents the circuit depth at which the device can run before coherence is lost, specifically, the depth at which at least one qubit error is statistically likely. The achievable depth is a useful metric which can be used separately from QV to estimate the feasibility of running an algorithm on a NISQ device.

The effective error ϵ_{eff} for each depth-1 circuit includes gate error, and errors associated with gate decomposition, connectivity, and parallelizability. The effective error can be used to calculate the achievable depth. Many iterations of the randomly generated circuit should be used to best capture the properties of the device. In this section, we present an error model for the quantum computing design analyzed in this manuscript and present results for a range of experimental parameters that may be achievable. In the following analysis, we assume linear propagation of errors, which represents a worst-case outcome, as it does not account for the possibility of a new error reducing a previous error. We combine the errors associated with connectivity and gates, as opposed to a full simulation of the quantum states and associated noise model. The advantage of this methodology is that we are able to make estimations on effective error (and therefore achievable depth) for a wide range of qubit numbers and device sizes. The effective error ϵ_{eff} for this design and circuit requirement is, $\epsilon_{\mathrm{eff}} = \epsilon_{\mathrm{gate}} + \epsilon_{\mathrm{conn}}$, where ϵ_{gate} is the two-qubit gate error and ϵ_{conn} is the error associated with enabling the required global connectivity. We decompose ϵ_{conn} into two components $\epsilon_{\mathrm{conn}} = \epsilon_{\mathrm{deco}} + \epsilon_{\mathrm{loss}}$, where ϵ_{deco} is the quantum decoherence associated with the total time taken to enable connectivity where $\epsilon_{\mathrm{deco}} = 1 - e^{-t/c}$ for time t and coherence time c. Recent work by Kaufmann et al.[35] demonstrated high state fidelity shuttling (99.9994%), where the coherence time associated with shuttling was extrapolated to be $2.13s$. A coherence time of $50s$ has been demonstrated for stationary ions in the atomic clock states of calcium.[43] In Figure 2A, we quantify the average time required to enable connectivity as a function of device size (qubit number). The stated dimensionless time τ can be converted to a real time by multiplying it with the expected time to shuttle an ion between two adjacent X-Junctions. For ion shuttling speed, a distance of 280 µm has been demonstrated in 3.6 µs[41] and 12.8 µs.[35] There will be an additional time cost associated with performing a single combination and a separation of ions, per iteration of the depth-1 circuit, which has been performed in 80 µs.[10,42] ϵ_{loss} represents the likelihood for an ion to be lost to the vacuum per iteration of the depth-1 circuit. Investigations of ion loss for routing across X-Junction centers[44] found that continuously Doppler cooled ions survive more than 10^5 round trips, whereas uncooled ions survive at least 65 round trips. Ion loss occurs when its motional energy exceeds the trap depth, which can be remedied by increasing the trapping potentials and by cooling techniques. Significant work is carried out in order to allow the application of large trapping voltages in order to increase the effective trapping potential; recently trapping voltages as large as 1000V have been demonstrated.[45] In Figure 3A, we quantify the average number of X-Junction crosses, X_{count}, as a function of device size (qubit number), which can be combined with an ion loss per shuttle rate, X_{loss}, for ϵ_{loss}. This can all be combined into a single equation defining the effective error in this design

$$\epsilon_{\mathrm{eff}} = \epsilon_{\mathrm{gate}} + (1 - e^{-t/c}) + (X_{\mathrm{count}} \times X_{\mathrm{loss}}) \tag{2}$$

This error model can be used to estimate the achievable depth for a wide variety of device sizes and experimental parameters for devices following this design. The gate error will depend on the requirement of the algorithm we are assessing against, which in the case of QV is the arbitrary two-qubit gate. The focus of this manuscript is the cost of enabling connectivity; therefore, we have chosen to utilize the concept behind QV but alter its algorithm requirement to instead be the native two-qubit gate of the architecture being assessed. We will refer to this new metric as $\mathrm{QV}_{\mathrm{native}}$ going forward. The costs associated with arbitrary two-qubit gate decompositions will be discussed later.

We use our error model to quantify $\mathrm{QV}_{\mathrm{native}}$ as a function of two-qubit gate fidelity for this architecture with two different assumptions on experimental shuttling parameters, shown in **Figure 5**. These can be compared to the upper bound of this metric which corresponds to a hypothetical architecture with free, all-to-all connectivity. To demonstrate an example, a device with free all-to-all connectivity and a two-qubit gate fidelity of 99.9% has a $\log_2 \mathrm{QV}_{\mathrm{native}}$ of 31.25. This implies that one could effectively run a globally connected native two-qubit gate algorithm with approximately 30 qubits at depth 30. We investigate up to a two-qubit gate fidelity of 99.99%; this analysis indicates that without error correction techniques, chasing high qubit numbers will be futile even with considerable improvement to the current state of the art two-qubit gate fidelities. The trapped ion plots of Figure 5 have an ion loss rate of 10^{-5}; we found that increasing this rate substantially decreases the $\mathrm{QV}_{\mathrm{native}}$, which seriously emphasises the importance of achieving an ion loss rate of this order. The ion loss rate can be improved by deeper trapping potentials and by techniques such as sympathetic cooling. Ions may also be automatically reloaded from a filled reservoir trap section.

We also quantify this metric for a model of a superconducting architecture, which is a square grid with only nearest neighbor interactions. In superconducting square grid systems, connectivity is enabled by sequences of swap operations, and the best known method has an overhead of $\Theta(N^{0.5})$[22] for the random complete graph (global connectivity). IBM provide an equation to estimate the depth overhead, of the form $(a\sqrt{N} + b)$, for a square grid but it includes their gate decomposition costs of arbitrary two-qubit gates.[37] Cowtan et al. developed a compiler to map quantum circuits to devices with restricted qubit connectivity and provides results on the depth overhead for nearest neighbor square devices.[25] Using the publicly available software, CQC's $t\,|ket\rangle$ and its recently improved connectivity compiler, the depth

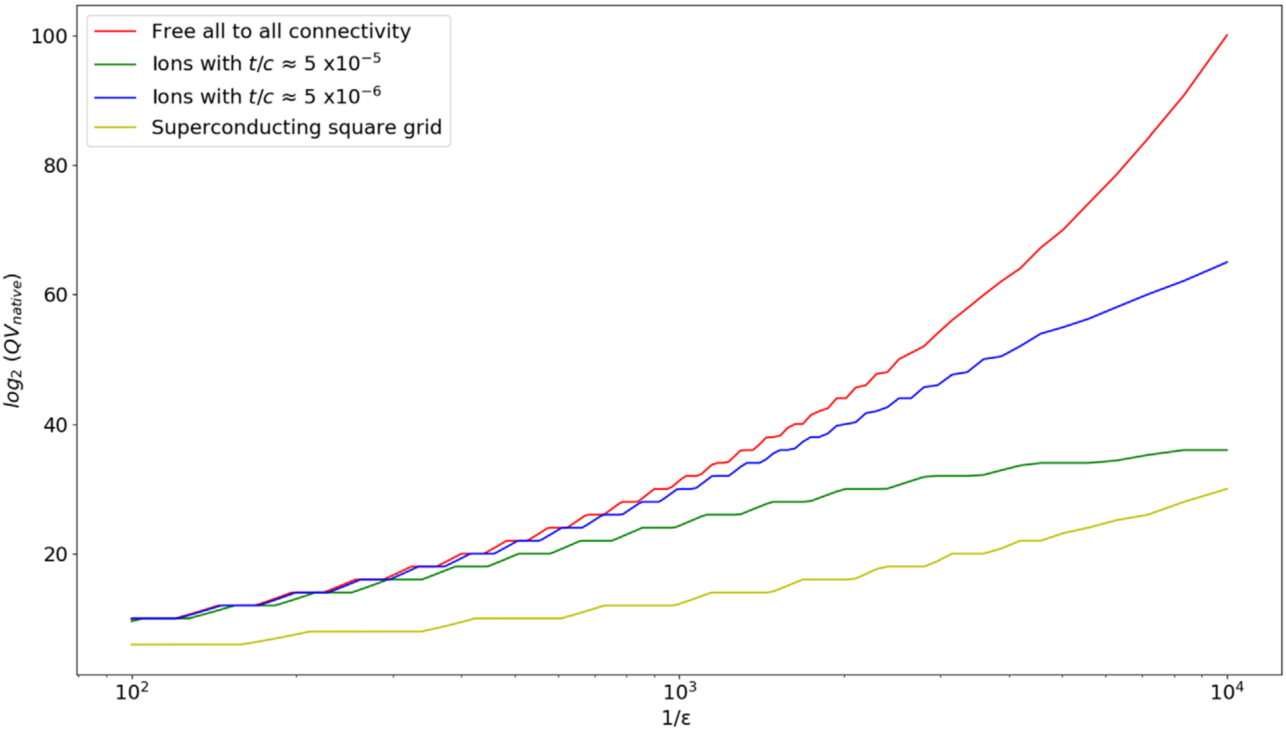

Figure 5. Log$_2$ of the quantum volume with a native two-qubit gate requirement as a function of inverse gate error, $1/\epsilon$, for different architectures. Here, the number of qubits utilized to achieve a given value of $\mathrm{QV}_{\mathrm{native}}$ is equal to $\log_2 \mathrm{QV}_{\mathrm{native}}$ rounded up to the nearest even integer. Red: An architecture with all-to-all connectivity where $\mathrm{QV}_{\mathrm{native}}$ is solely defined by the native two-qubit gate fidelity using Equation (1) and represents the upper bound. Blue: The trapped ion architecture investigated in this manuscript using our proposed error model and the the routing results of the previous section. The coherence time and the time taken to shuttle between adjacent X-Junctions are extrapolated from work by Kaufmann et al.[35] We assume a distance between adjacent X-Junctions of 2500 μm,[18] which implies a shuttling time, t, of 114 μs, and we use the demonstrated state fidelity of shuttling[35] to infer a coherence time, c, of 2.13 s, and so $t/c \approx 5 \times 10^{-5}$. We assume an ion loss rate of 10^{-5} per X-Junction pass. We assume each iteration of the depth-1 circuit requires one combination and one separation operation, each of which have a duration of 80 μs,[10,42] and we assume the state fidelity of the operation can be inferred from the coherence time. Green: All the assumptions are identical to the above except for the coherence time which has been increased by a factor 10.[43] Yellow: A model of a square grid superconducting architecture where connectivity is enabled through sequences of nearest neighbour swap operations which require 3 native two-qubit gates (the CNOT). The depth overhead was found to scale as a function of qubit number N as $2.77\sqrt{N} - 4.53$ using the publicly available quantum compiling software, CQC's $t\,|ket\rangle$; improvements to the connectivity compiler would reduce this overhead.

overhead was found to scale with qubit number N as, $2.77\sqrt{N} - 4.53$. This overhead corresponds to a depth-1, 100% gate density, native two-qubit gate (CNOT) algorithm with $10N$ iterations. A swap gate is implemented with three CNOTs and no advantageous initial qubit mapping was utilized.

The native two-qubit gate of this trapped ion design is the Mølmer-Sørensen,[46] and although it does not directly depend on the motional state, it is affected by the heating rate and experimental offsets whereby it is favorable to begin in a low motional state. Therefore, to reach the high two-qubit gate fidelities used in Figure 5, it will be necessary to use cooling techniques. Techniques such as Doppler and sideband cooling are only suitable for the beginning of a quantum algorithm as they do not preserve quantum information. Sympathetic cooling is a way of actively cooling throughout a quantum algorithm, whereby the qubit is sympathetically cooled via a different laser cooled ion species. It is likely to be a critical technique for the use of trapped ion devices, particularly in the fault tolerant regime. Shuttling-based designs may benefit from multi-species shuttling. The relative difference between the upper bound of free all-to-all connectivity and the plots for trapped ions increases with the two-qubit fidelity due to the independent cost associated with shuttling. We find a notable difference in the $\mathrm{QV}_{\mathrm{native}}$ between the superconducting plot and the all-to-all, particularly at higher two-qubit gate fidelities. Superconducting square grids have a slower growth rate with two-qubit gate fidelity because the associated depth overhead of swaps increases with the number of qubits (the size of the grid). In this model, the trapped ion design outperforms the superconducting square grid for this set of experimental shuttling parameters. The number of shuttling operations, τ, required to enable connectivity in the trapped ion design analyzed here, scales as $\tau = 1.3\,(3)\,\sqrt{N} + 2\,(5)$, which is comparable to the depth overhead for swapping on the superconducting square grid. Extrapolating from the high state fidelity shuttling of Kaufmann et al.,[35] it implies a fidelity per shuttling operation (2500 μm) of 99.995%, which is significantly higher than the two-qubit gate fidelities achieved so far by superconducting systems. In order to facilitate further work with our error model by others, we have made it open access.[36] To experimentally implement the work presented here, a key challenge is to build and operate such a trap as shown in Figure 1B. A trap needs to be fabricated for which a number of approaches are being perused.[47]

The QV metric requires application of arbitrary, randomly generated SU(4) two-qubit gates, as opposed to the native two-qubit gate investigated above. The purpose of this requirement is to capture the power of the architecture's native gate set. There is an upper bound circuit which can express any arbitrary U(4) using 3 CNOTs and 15 elementary single-qubit gates,[48] with a native gate set consisting of $R_x(\theta), R_z(\theta)$, and the CNOT. We have translated this upper bound circuit into the native gate set of the architecture analyzed here, which is $R_x(\theta), R_y(\theta)$, and the Mølmer–Sørensen (MS) two-qubit gate[46] (see Appendix). The gate count of the new upper bound circuit is 3 MS gates and 18 elementary single-qubit gates. We reduced the initial single-qubit gate count from 29 to 18 by utilizing basic commutation relations and the degrees of freedom which are available.[38] The upper bound circuit represents a worst case and optimal circuits can be found for particular SU(4)s using analytical techniques[49] but most exact decompositions of arbitrary two-qubit gates will require the three native two-qubit gates of the upper bound. A new technique demonstrated by IBM can considerably improve the fidelity of decomposing these gates[37]; Cross et al. instead start with an allowable error on the decomposition, which allows one to identify cases which require less than the upper bound of three two-qubit gates. This can result in a considerable improvement to the final fidelity, particularly when working with lower two-qubit gate fidelities. The quantum volume with native qubit gates we have used in this section is a clear tool of comparison for the cost of connectivity in these two architectures. To extend this comparison to architectures with drastically different gate sets, such as those in some trapped ion designs which enable multiple (> 2) qubit gates, the original QV metric is more suitable. Once more research characterizing quantum volume for various quantum computing designs becomes available, a more detailed comparison would be warranted.

4. Conclusion

The quantum computing architecture analyzed in this manuscript has a clear path toward scaling to large qubit numbers. Arbitrary connectivity between qubits can be enabled in this design on near-term devices, relying only on shuttling across a square grid, but prior to this work, there were no proposed routing algorithms. We have created a routing algorithm which efficiently enables connectivity in this design. A simulation tool was created which allowed us to characterize the routing algorithm and compare it against a strict lower bound to which it scales with an equal gradient. The routing algorithm compares favorably against positional-swap-based routing for the experimental values used. We propose an error model which can be combined with the results from the simulation tool to estimate the circuit depth of a device as a function of experimental parameters. We use a metric, $\mathrm{QV}_{\mathrm{native}}$, based on quantum volume which instead has native two-qubit gates, to focus on and assess the cost of connectivity in this trapped ion design. The ion loss per shuttling operation was found to be an important parameter of the model and needed to be low, at 10^{-4}–10^{-5}, to reach appreciable circuit depths and it can be improved experimentally with larger trapping potentials. It is necessary to maintain a sufficiently low motional state energy of the ions to reach high two-qubit gate fidelities, which highlights the importance of developing techniques such as sympathetic cooling, and therefore multi-species shuttling. We use $\mathrm{QV}_{\mathrm{native}}$ to assess a model of a square grid superconducting device, and find that for the shuttling parameters used, this trapped ion design has a substantially lower cost associated with connectivity. The simulation tool and this analysis can be used to inform the development of devices following this design, by metering experimental priorities, and by solidifying the requirements on shuttling. This work has implications for error correction schemes, especially those which rely on non-nearest neighbor interactions.

Appendix

A1. Decomposing Arbitrary Two-Qubit Gates

An upper bound circuit for expressing arbitrary two-qubit gates in terms of $R_x(\theta), R_z(\theta)$, and the CNOT, was found by Vatan et al.[48]

An arbitrary single-qubit gate U_1, can be expressed in the form

$$U_1 = e^{i\alpha} R_{\hat{n}}(\beta) R_{\hat{m}}(\gamma) R_{\hat{n}}(\delta) \tag{A1}$$

for appropriate choices of $\alpha, \beta, \gamma, \sigma$, where $\hat{n}$, and $\hat{m}$ are non-parallel real unit vectors in three dimensions.[6] We have converted the circuit of **Figure A1** into the native gate set of the architecture investigated here, which is, $R_x(\theta), R_y(\theta)$ and the Mølmer-Sørensen gate $U_{\mathrm{MS}}(\chi)$[46] which has the form

$$U_{\mathrm{MS}}(\chi) = \begin{pmatrix} \cos(\chi) & 0 & 0 & -i\sin(\chi) \\ 0 & \cos(\chi) & -i\sin(\chi) & 0 \\ 0 & -i\sin(\chi) & \cos(\chi) & 0 \\ -i\sin(\chi) & 0 & 0 & \cos(\chi) \end{pmatrix} \tag{A2}$$

where χ can be set between $-\pi/4$ and $\pi/4$. The new converted circuit is shown in **Figure A2**, and has a gate count of three MS gates and 18 single-qubit gates. The single-qubit gate count was reduced by combining superfluous sequences of single-qubit gates, utilizing commutation relations, and the available degrees of freedom. The MS gate commutes with any $R_x(\theta)$. When decomposing the CNOT and $R_z(\theta)$ gate, there is an available degree

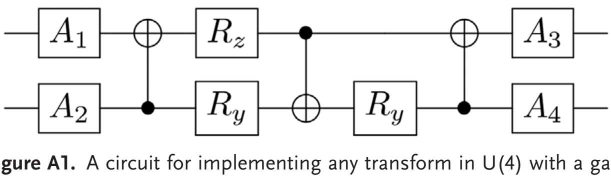

Figure A1. A circuit for implementing any transform in U(4) with a gate set consisting of $R_x(\theta)$, $R_z(\theta)$, and the CNOT, where the gate A_i here represents an arbitrary single-qubit transform, for a total gate count of 15 elementary single-qubit gates and 3 CNOTs.

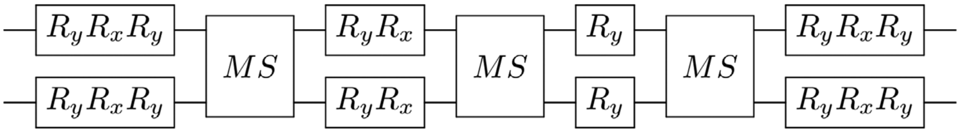

Figure A2. A circuit for implementing any transform in U(4) with a gate set consisting of $R_x(\theta)$, $R_y(\theta)$, and the Mølmer–Sørensen gate,[48] for a total gate count of 18 elementary single-qubit gates and 3 MS gates.

of freedom, where one may choose the direction of rotation on certain R_y gates, which can then be used to eliminate some R_y gates from the circuit.[38]

Acknowledgements

This work was supported by the U.K. Engineering and Physical Sciences Research Council via the EPSRC Hub in Quantum Computing and Simulation (EP/T001062/1), the U.K. Quantum Technology hub for Networked Quantum Information Technologies (No. EP/M013243/1), the European Commission's Horizon-2020 Flagship on Quantum Technologies Project No. 820314 (MicroQC), the U.S. Army Research Office under Contract No. W911NF-14-2-0106, and the University of Sussex. Thanks are given to David Bretaud for helpful conversations regarding gate decomposition, to Zak Romaszko for help with 3D modeling, to Mitchell Peaks for proofreading, and finally to Silas Dilkes for help with utilizing $t\,|ket\rangle$ to calculate the superconducting square grid swap cost.

Conflict of Interest

The authors declare no conflict of interest.

Keywords

ion trapping, noisy intermediate-scale quantum, quantum computing, quantum information processing, quantum volume

Received: February 28, 2020
Revised: May 29, 2020
Published online:

[1] B. P. Lanyon, J. D. Whitfield, G. G. Gillett, M. E. Goggin, M. P. Almeida, I. Kassal, J. D. Biamonte, M. Mohseni, B. J. Powell, M. Barbieri, A. Aspuru-Guzik, A. G. White, *Nat. Chem.* **2010**, *2*, 106.

[2] M. Reiher, N. Wiebe, K. M. Svore, D. Wecker, M. Troyer, *Proc. Natl. Acad. Sci. USA* **2017**, *114*, 7555.

[3] R. Orús, S. Mugel, E. Lizaso, *Rev. Phys.* **2019**, *4*, 100028.

[4] A. Y. Kitaev, *arXiv:9511026*, **1995**.

[5] P. W. Shor, *SIAM J. Comput.* **1997**, *26*, 1484.

[6] M. A. Nielsen, I. L. Chuang, *Quantum Computation and Quantum Information*, Cambridge University Press, Cambridge **2000**.

[7] A. Montanaro, *npj Quantum Inf.* **2016**, *2*, 15023.

[8] A. G. Fowler, A. M. Stephens, P. Groszkowski, *Phys. Rev. A: At., Mol., Opt. Phys.* **2009**, *80*, 052312.

[9] A. G. Fowler, M. Mariantoni, J. M. Martinis, A. N. Cleland, *Phys. Rev. A: At., Mol., Opt. Phys.* **2012**, *86*, 3.

[10] A. Bermudez, X. Xu, R. Nigmatullin, J. O'Gorman, V. Negnevitsky, P. Schindler, T. Monz, U. G. Poschinger, C. Hempel, J. Home, F. Schmidt-Kaler, M. Biercuk, R. Blatt, S. Benjamin, M. Müller, *Phys. Rev. X* **2017**, *7*, 4.

[11] J. Preskill, *Quantum* **2018**, *2*, 79.

[12] F. Arute, K. Arya, R. Babbush, D. Bacon, J. C. Bardin, R. Barends, R. Biswas, S. Boixo, F. G. Brandao, D. A. Buell, B. Burkett, Y. Chen, Z. Chen, B. Chiaro, R. Collins, W. Courtney, A. Dunsworth, E. Farhi, B. Foxen, A. Fowler, C. Gidney, M. Giustina, R. Graff, K. Guerin, S. Habegger, M. P. Harrigan, M. J. Hartmann, A. Ho, M. Hoffmann, T. Huang, et al., *Nature* **2019**, *574*, 505.

[13] J. R. McClean, J. Romero, R. Babbush, A. Aspuru-Guzik, *New J. Phys.* **2016**, *18*, 2.

[14] N. Moll, P. Barkoutsos, L. S. Bishop, J. M. Chow, A. Cross, D. J. Egger, S. Filipp, A. Fuhrer, J. M. Gambetta, M. Ganzhorn, A. Kandala, A. Mezzacapo, P. Müller, W. Riess, G. Salis, J. Smolin, I. Tavernelli, K. Temme, *Quantum Sci. Technol.* **2018**, *3*, 3.

[15] G. Wendin, *Rep. Prog. Phys.* **2017**, *80*, 10.

[16] C. D. Bruzewicz, J. Chiaverini, R. McConnell, J. M. Sage, *Appl. Phys. Rev.* **2019**, *6*, 021314.

[17] D. Kielpinski, C. Monroe, D. J. Wineland, *Nature* **2002**, *417*, 709.

[18] B. Lekitsch, S. Weidt, A. G. Fowler, K. Mølmer, S. J. Devitt, C. Wunderlich, W. K. Hensinger, *Sci. Adv.* **2017**, *3*, 2.

[19] H. Kaufmann, T. Ruster, C. T. Schmiegelow, M. A. Luda, V. Kaushal, J. Schulz, D. Von Lindenfels, F. Schmidt-Kaler, U. G. Poschinger, *Phys. Rev. Lett.* **2017**, *119*, 150503.

[20] V. Kaushal, B. Lekitsch, A. Stahl, J. Hilder, D. Pijn, C. Schmiegelow, A. Bermudez, M. Müller, F. Schmidt-Kaler, U. Poschinger, *AVS Quantum Sci.* **2020**, *2*, 014101.

[21] D. Hanneke, J. P. Home, J. D. Jost, J. M. Amini, D. Leibfried, D. J. Wineland, *Nat. Phys.* **2010**, *6*, 13.

[22] D. Cheung, D. Maslov, S. Severini, presented at The Eleventh Workshop on Quantum Information Processing (QIP 2008), New Delhi, India, December **2007**.

[23] S. Herbert, *arXiv:1805.12570*, **2018**.

[24] S. Brierley, *arXiv:1507.04263*, **2015**.

[25] A. Cowtan, S. Dilkes, R. Duncan, A. Krajenbrink, W. Simmons, S. Sivarajah, in *14th Conference on the Theory of Quantum Computation, Communication and Cryptography (TQC 2019)*, Maryland, USA, June 2019 (Eds: W. van Dam, L. Mancinska), Leibniz International Proceedings in Informatics (LIPIcs), Vol. 135, Schloss Dagstuhl–LZI GmbH, **2019**.

[26] S. Herbert, A. Sengupta, *arXiv:1812.11619*, **2018**.

[27] N. M. Linke, D. Maslov, M. Roetteler, S. Debnath, C. Figgatt, K. A. Landsman, K. Wright, C. Monroe, *Proc. Natl. Acad. Sci. USA* **2017**, *114*, 3305.

[28] S. Debnath, N. M. Linke, C. Figgatt, K. A. Landsman, K. Wright, C. Monroe, *Nature* **2016**, *536*, 63.

[29] P. Schindler, D. Nigg, T. Monz, J. T. Barreiro, E. Martinez, S. X. Wang, S. Quint, M. F. Brandl, V. Nebendahl, C. F. Roos, M. Chwalla, M. Hennrich, R. Blatt, *New J. Phys.* **2013**, *15*, 123012.

[30] K. R. Brown, J. Kim, C. Monroe, *npj Quantum Inf.* **2016**, *2*, 16034.

[31] C. Monroe, R. Raussendorf, A. Ruthven, K. R. Brown, P. Maunz, L. M. Duan, J. Kim, *Phys. Rev. A: At., Mol., Opt. Phys.* **2014**, *89*, 2.

[32] R. Nigmatullin, C. J. Ballance, N. de Beaudrap, S. C. Benjamin, *New J. Phys.* **2016**, *18*, 103028.

[33] S. Weidt, J. Randall, S. C. Webster, K. Lake, A. E. Webb, I. Cohen, T. Navickas, B. Lekitsch, A. Retzker, W. K. Hensinger, *Phys. Rev. Lett.* **2016**, *117*, 22.

[34] C. D. Herold, S. D. Fallek, J. T. Merrill, A. M. Meier, K. R. Brown, C. E. Volin, J. M. Amini, *New J. Phys.* **2016**, *18*, 2.

[35] P. Kaufmann, T. F. Gloger, D. Kaufmann, M. Johanning, C. Wunderlich, *Phys. Rev. Lett.* **2018**, *120*, 1.

[36] M. Webber, Depth model, https://github.com/mawebber1/Calculataing-achievable-depth-for-Trapped-Ions (accessed: May 2020).

[37] A. W. Cross, L. S. Bishop, S. Sheldon, P. D. Nation, J. M. Gambetta, *Phys. Rev. A* **2019**, *100*, 3.

[38] D. Maslov, *New J. Phys.* **2017**, *19*, 2.

[39] H. Kaufmann, T. Ruster, C. T. Schmiegelow, M. A. Luda, V. Kaushal, J. Schulz, D. Von Lindenfels, F. Schmidt-Kaler, U. G. Poschinger, *Phys. Rev. A* **2017**, *95*, 052319.

[40] M. W. van Mourik, E. A. Martinez, L. Gerster, P. Hrmo, T. Monz, P. Schindler, R. Blatt, *arXiv:2001.02440*, **2020**.

[41] A. Walther, F. Ziesel, T. Ruster, S. T. Dawkins, K. Ott, M. Hettrich, K. Singer, F. Schmidt-Kaler, U. Poschinger, *Phys. Rev. Lett.* **2012**, *109*, 080501.

[42] T. Ruster, C. Warschburger, H. Kaufmann, C. T. Schmiegelow, A. Walther, M. Hettrich, A. Pfister, V. Kaushal, F. Schmidt-Kaler, U. G. Poschinger, *Phys. Rev. A: At., Mol., Opt. Phys.* **2014**, *90*, 3.

[43] T. P. Harty, D. T. Allcock, C. J. Ballance, L. Guidoni, H. A. Janacek, N. M. Linke, D. N. Stacey, D. M. Lucas, *Phys. Rev. Lett.* **2014**, *113*, 22.

[44] K. Wright, J. M. Amini, D. L. Faircloth, C. Volin, S. Charles Doret, H. Hayden, C. S. Pai, D. W. Landgren, D. Denison, T. Killian, R. E. Slusher, A. W. Harter, *New J. Phys.* **2013**, *15*, 033004.

[45] R. C. Sterling, H. Rattanasonti, S. Weidt, K. Lake, P. Srinivasan, S. C. Webster, M. Kraft, W. K. Hensinger, *Nat. Commun.* **2014**, *5*, 1.

[46] K. Mølmer, A. Sørensen, *Phys. Rev. Lett.* **1999**, *82*, 1835.

[47] Z. D. Romaszko, S. Hong, M. Siegele, R. K. Puddy, F. R. Lebrun-Gallagher, S. Weidt, W. K. Hensinger, *arXiv:1908.00267*, **2019**.

[48] F. Vatan, C. Williams, *Phys. Rev. A: At., Mol., Opt. Phys.* **2004**, *69*, 032315.

[49] M. Blaauboer, R. L. De Visser, *J. Phys. A: Math. Theor.* **2008**, *41*, 39.