DOI:10.1145/3511064

Toward Systematic Architectural Design of Near-Term Trapped Ion Quantum Computers

To view the accompanying Technical Perspective, visit doi.acm.org/10.1145/3511063

By Prakash Murali, Dripto M. Debroy, Kenneth R. Brown, and Margaret Martonosi

Abstract

Trapped ions (TIs) are a leading candidate for building Noisy Intermediate-Scale Quantum (NISQ) hardware. TI qubits have fundamental advantages over other technologies, featuring high qubit quality, coherence time, and qubit connectivity. However, current TI systems are small in size and typically use a single trap architecture, which has fundamental scalability limitations. To progress toward the next major milestone of 50–100 qubit TI devices, a modular architecture termed the Quantum Charge Coupled Device (QCCD) has been proposed. In a QCCD-based TI device, small traps are connected through ion shuttling. While the basic hardware components for such devices have been demonstrated, building a 50–100 qubit system is challenging because of a wide range of design possibilities for trap sizing, communication topology, and gate implementations and the need to match diverse application resource requirements.

Toward realizing QCCD-based TI systems with 50–100 qubits, we perform an extensive application-driven architectural study evaluating the key design choices of trap sizing, communication topology, and operation implementation methods. To enable our study, we built a design toolflow, which takes a QCCD architecture's parameters as input, along with a set of applications and realistic hardware performance models. Our toolflow maps the applications onto the target device and simulates their execution to compute metrics such as application run time, reliability, and device noise rates. Using six applications and several hardware design points, we show that trap sizing and communication topology choices can impact application reliability by up to three orders of magnitude. Microarchitectural gate implementation choices influence reliability by another order of magnitude. From these studies, we provide concrete recommendations to tune these choices to achieve highly reliable and performant application executions. With industry and academic efforts underway to build TI devices with 50–100 qubits, our insights have the potential to influence QC hardware in the near future and accelerate the progress toward practical QC systems.

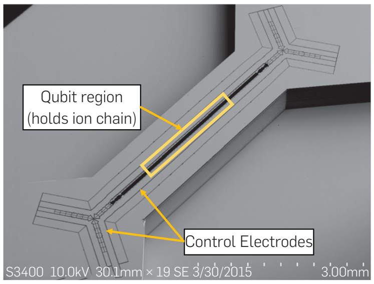
1. INTRODUCTION

Trapped ions (TIs) are one of the leading candidates for building *qubits* (short for quantum bits). Figure 1 shows an example system, where ions are isolated and trapped using an electromagnetic held. To enable computations, the internal atomic states of the ions are used to represent the 0 and 1 basis states for a qubit and laser control pulses are used to implement *gates* (instructions). Industry vendors such as IonQ and Honeywell, along with nearly a hundred academic groups worldwide, are working to build quantum computing (QC) systems using this technology. To date, the largest TI systems have up to 32 qubits (IonQ) and have been used for both demonstrating promising near-term QC applications and, recently, a milestone demonstration of quantum error correction.[5]

To demonstrate quantum advantage over classical computing, QC systems with 50–100 qubits are required. However, most current TI devices have a fundamental architectural scaling bottleneck: they are based on an architecture where all the ions are contained within the same trapping zone. In this single-trap architecture, ion spacing and ion–ion interaction strength reduce as more ions are added to the trap. Hence, with increasing number of qubits, qubit control and gate implementation become increasingly unreliable and time consuming.

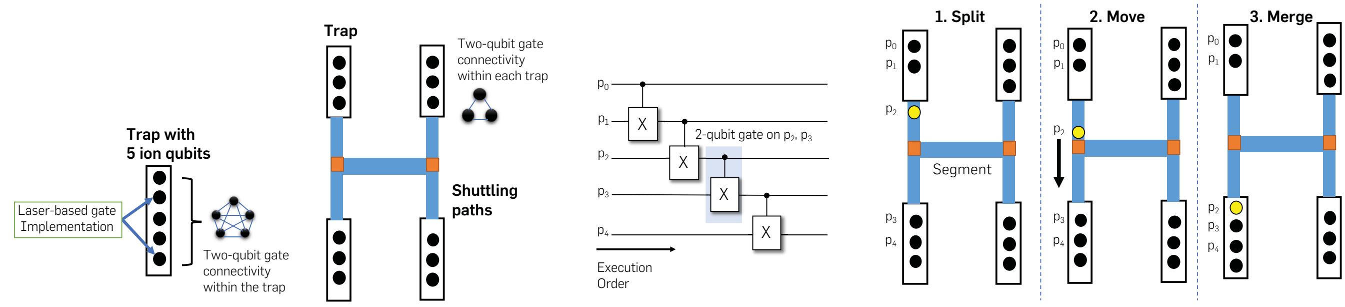
To circumvent this bottleneck, a modular architecture called Quantum Charge Coupled Device (QCCD) was proposed nearly two decades ago.[11] Figure 2b shows an example. QCCD systems eschew long ion chains in favor of multiple traps, each housing a smaller ion chain. Similar to single-trap architectures, gates can be performed on one or more ions that are co-located within the same trap. To enable gates across traps, QCCD uses *ion shuttling*. That is,

Figure 1. Scanning electron micrograph of the HOA-2 trap designed and fabricated at Sandia National Laboratories. Figure adapted with permission from Maunz.[15] A single trap houses all the ions. Control electrodes are used to load, remove, and move ions. This architecture does not scale beyond 50–100 qubits because of gate implementation challenges in long ion chains.



The original version of this paper is entitled "Architecting Noisy Intermediate-Scale Trapped Ion Quantum Computers" and was published in *Proceedings of the ACM/IEEE 47th Annual International Symposium on Computer Architecture*, 2020.

Figure 2. (a) A 5-qubit TI system with a single trap. Each black circle represents a qubit. Two-qubit gates are performed by pulsing the desired pair of qubits with lasers, allowing a single trap to support full connectivity among the qubits. (b) A modular Quantum Charge Coupled Device (QCCD) with 4 traps. Each trap initially has 3 ions and a maximum capacity of 4 ions. The traps are interconnected through shuttling paths to move ions from one trap to another. The orange squares represent junctions where shuttling paths meet. (c) An example program intermediate representation (IR). For clarity, we show only two-qubit gates. Real program IR also includes single-qubit gates and qubit measurement operations. To execute the IR on the device in (a), each ion in the device can be used to represent one qubit from the IR, and gates can be executed using the laser controller. (d) To execute the IR on the device in (b), p_0, p_1, and p_2 are mapped onto one trap, and p_3 and p_4 are mapped onto another. The first two gates are executed within the top left trap. For the gate on p_2 and p_3, the qubits need to be co-located within the same trap, so p_2 is shuttled to the trap containing p_3 and the gate is performed inside the bottom left trap.



(a) 5-qubit system with a single trap

(b) Modular QCCD system

(c) Example program IR

(d) Shuttling operation on p_2

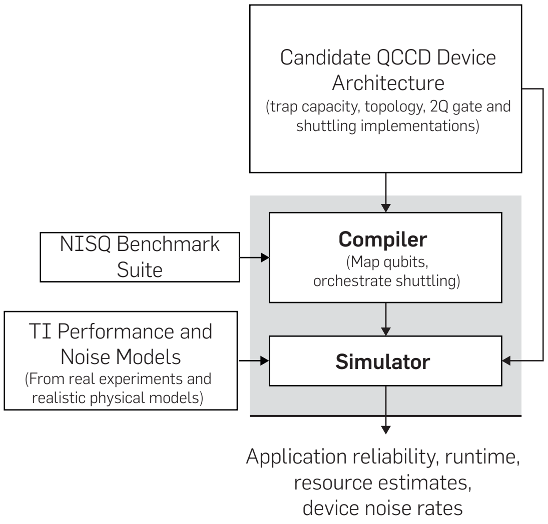
prior to a gate that involves ions from different traps, one of the ions is physically moved from one trap to the other. Figure 2c and 2d show an example shuttling operation. While several other scaling proposals exist in theory, all basic components required for QCCD systems have been developed and refined over the last decade, and several groups are working on prototyping systems.[7, 10, 20] Recently, Honeywell demonstrated the first generation of 10-qubit QCCD systems, which are capable of running algorithms.[20]

To scale QCCD systems to the next major milestone of 50–100 qubits, hardware designers have to navigate a variety of conflicting design choices regarding the number of ion qubits per trap, communication topology, and gate and shuttling implementation methods. Although individual experiments have been performed to understand some design choices, current hardware is largely designed from physics considerations alone, without considering the capabilities of the rest of the software stack, architecture, or application characteristics. Our work is the first effort toward systematically exploring these design options, using proven design approaches from classical computer architecture. To co-design the next generation of mid-sized TI systems with application requirements, we develop the design toolflow shown in Figure 3. Using this toolflow, we perform an extensive application-driven design analysis and propose recommendations for future hardware designs.

Our contributions include the following:

First, while recent works have focused on architecture for superconducting QC systems,[6, 8, 14] there has been less attention on TI systems although the technology is very promising. Our work performs the first architectural studies targeting systems with 50–100 qubits, which are the next major milestone for TI systems. Our simulations emphasize the importance of optimizing the architecture; across the hardware design space, application reliability varies up to five orders of magnitude depending on the choice of trap capacity, connectivity, and gate implementations.

Figure 3. Our framework for evaluating a candidate QCCD-based TI system. Taking a candidate architecture, a set of NISQ applications, and realistic performance models as input, the toolflow computes application metrics like runtime and reliability (fidelity) and device metrics like heating rates.


Candidate QCCD Device Architecture
(trap capacity, topology, 2Q gate and shuttling implementations)
NISQ Benchmark Suite
Compiler
(Map qubits, orchestrate shuttling)
TI Performance and Noise Models
(From real experiments and realistic physical models)
Simulator
Application reliability, runtime, resource estimates, device noise rates

Second, our work provides concrete guidance for device designers as they architect larger systems. We find that having a capacity of 15–25 qubits per trap is ideal across applications and device topologies. This capacity range minimizes the impact of ion heating, laser beam instabilities, and motional energy hot spots across the device while still offering very good application performance. In addition, device topology must be co-designed for the needs of applications to achieve high reliability. For near-term applications such as QAOA, linear device topologies work well and simplify hardware implementation.

Third, our work provides insights on the best microarchitectural choices. We evaluate four entangling gate implementations and two methods for chain reordering and show that the most reliable implementations vary according to application characteristics. That is, the microarchitecture must be reconfigurable according to application requirements.

2. QUANTUM COMPUTING BACKGROUND

2.1. Principles of quantum computing

Qubits. The building block of a QC system is a *qubit* (quantum bit). Qubits have two basis states, $|0\rangle$ and $|1\rangle$. Using superposition, a qubit can be in a complex linear combination of the basis states, represented by $a|0\rangle+|1\rangle$, for $\alpha, \beta \in \mathbb{C}$. This allows an n-qubit system to potentially represent all 2^n basis states simultaneously, unlike a classical n-bit register, which can be in exactly one of the 2^n states.

Gates. To manipulate information, QC systems use *gates* to modify the qubit amplitudes. Gates act on one or more qubits at a time. Similar to universal gates in classical computing, QC systems typically support a set of universal single-qubit and two-qubit gates. QC applications are expressed using these gate sets. To run a program, a sequence of gates is executed on a set of appropriately initialized qubits. The gates transform the qubit amplitudes, evolving the state space toward the desired output. To obtain classical output at the end of the algorithm, a qubit is measured, collapsing its state to either $|0\rangle$ or $|1\rangle$.

2.2. Overview of trapped ion QC systems

Qubit register (ion chain). In a TI quantum computer, information is stored in the internal states of ions, which are trapped within an oscillatory potential. DC electrodes on both ends of the trap provide a barrier along the axis of the trap, and a radio-frequency oscillating electric field fluctuates in the other two directions, causing the ions to be arranged as linear chain with even spacing.

Qubit states. To store the $|0\rangle$ and $|1\rangle$ states required for QC, there are a wide variety of ion internal states, like hyperfine and Zeeman states, that can be chosen each having different strengths and weaknesses. The performance models used in our work assume qubits defined on hyperfine states, which is the standard choice in current devices. However, the insights from our work will also apply to other qubit states.

Gate implementation using lasers. Gates are implemented by exciting ions using lasers. Single qubit gates involve a single laser interacting with the desired ion, while two-qubit gates use multiple lasers, in order to excite the internal states of the ions and also the vibrational motions of the chain. Two-qubit gates use these joint oscillatory motions, also known as motional modes, as a bus to allow communication between internal states of distant ions.[21] The canonical two-qubit gate is the Mølmer-Sørensen gate (MS), an entangling gate represented by a time evolution under an Ising-type Hamiltonian; it is insensitive to the motional state of the ions. This motional state can cause issues with laser addressing of the ions and is captured in our error models.

Fidelity. In real QC systems, errors occur due to imperfect qubit control, errors in pulse implementation, and external interference. *Gate fidelity* refers to the quality of a gate measured using methods such as randomized benchmarking.

3. BACKGROUND ON QCCD-BASED TI SYSTEMS

3.1. Challenges in single trap architectures

To motivate the design of QCCD-based systems, we consider the challenges in scaling single trap systems to 50–100 qubits. First, within a single trap, the inter-ion spacing is determined by the balance between the trapping field and the Coulomb repulsion between the ions. When the ion count increases, the inter-ion spacing reduces, making it difficult to selectively pulse a qubit using laser controllers. Second, two-qubit gate implementation is also challenging. Within a trap, the ion–ion coupling strength for a pair of ions at distance d scales in proportion to $1/d^{\alpha}$ with α ranging from 1 to 3.[12] This increases the time required to perform an entangling gate on an arbitrary pair of qubits. Furthermore, the collective motional modes (vibrational modes) of the ion chain are used to mediate the two-qubit interaction. The density of modes increases with ion count, worsening the chance of crosstalk among modes and reducing gate fidelity. Put together, these challenges make it difficult to scale single-trap TI devices beyond tens of qubits.

3.2. Components of the QCCD architecture

QCCD devices overcome the challenges of single-trap systems using a modular design having a set of small ion chains, each in an individual trap. In Figure 2b, the system has 12 ions, separated into 4 traps of size 3 each. By restricting capacity, this design achieves fast and high-fidelity two-qubit operations within each trap. To enable two-qubit gates across traps, QCCD uses ion shuttling to physically move ions from one trap to another prior to the entangling operation.

Figure 2d illustrates three steps involved in shuttling. First, the desired ion is *split* from the source chain. To move this ion, shuttling paths are implemented as a set of segments connected by junctions. In Figure 2b, the system has 5 segments (blue), connected using 2 junctions (orange). The split ion is moved from the trap through the segments and junctions to the desired trap. These *move* operations also include any turns required at the junctions. Finally, the shuttled ion is *merged* into the destination chain. Experimentally, these operations are implemented using time-varying waveforms on the control electrodes attached to the trap segments.[3]

4. DESIGN TRADE-OFFS IN QCCD-BASED TI SYSTEMS

4.1. Trap capacity choices

Individual traps within a QCCD architecture are identical to a single-trap TI system; hence, they face the same qubit addressing and gate implementation challenges if the number of ions in a single chain is too high. Therefore, having low trap capacity is beneficial to applications because it enables fast and reliable two-qubit gates within a trap. However, having low capacity is harmful because it sacrifices qubit connectivity, which is a key advantage of TI systems over other technologies. Satisfying an algorithm's two-qubit gate requirements with low trap capacity necessitates more shuttling, including more splits, moves, and merges. These operations increase execution time and reduce reliability. Further, shuttling operations introduce qubit motion via the trapping potentials and induce heating of the vibrational modes of the ion chain. This impacts qubit addressability using lasers and reduces the gate fidelities.

Our work studies: How does trap sizing affect QCCD-based TI systems with 50–100 qubits? What sizes work well for NISQ applications and to what extent do application characteristics such as two-qubit gate patterns affect sizing?

4.2. Communication topology choices

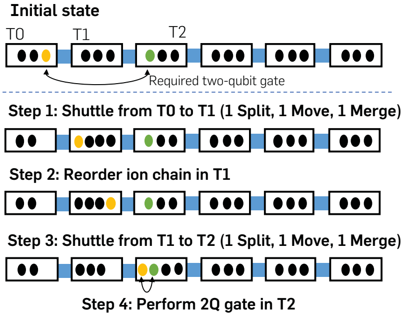
QCCD systems have different topology options for orchestrating shuttling operations. To understand the trade-offs, consider the linear topology shown in Figure 4. This topology is the easiest to build and imposes the minimum requirements on the number of required segments. Since there are no junctions, move operations are simplified. However, the linear topology restricts distant communication paths. To move an ion to a nonadjacent trap, several split and merge operations are required at intermediate traps. Splits and merges are more difficult compared to moves and can potentially impact applications. Additionally, split and merge operations require that the ion is positioned at the correct end of the chain. In our example, after the yellow ion is merged at the second trap, it needs to be repositioned at the right end of the second trap using a *chain reordering* operation. These operations can also impact application metrics. In contrast, grid topologies, such as Figure 2b, offer better communication paths at the expense of more hardware. In this particular 2 × 2 topology, shuttles do not encounter intermediate traps, and hence avoid the extra split, merge operations of the linear topology. However, grids require 3- and 4-way junction turns, which are nontrivial compared to simple move operations through straight segments.

We ask: How much does QCCD device topology affect application reliability and performance? Are the overheads of extra split and merge operations in linear topologies prohibitive? What communication topologies can best support NISQ applications with 50–100 qubits?

4.3. Gate and shuttling implementation choice

Two-qubit gates within a trap. To implement two-qubit gates, the shared motion of the ion chain can be harnessed in different ways. The two leading gate methods are based on amplitude modulation (AM)[4, 22, 25] and frequency modulation (FM)[12, 13] of the laser control pulses. We also consider a recent proposal based on phase modulation (PM).[16]

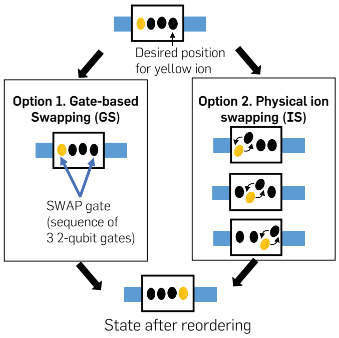
Figure 4. Shuttling in a QCCD-system, which has linear device topology. Extra split and merge operations are required while moving ions through intermediate traps.


Initial state
T0
T1
T2
Required two-qubit gate
Step 1: Shuttle from T0 to T1 (1 Split, 1 Move, 1 Merge)
Step 2: Reorder ion chain in T1
Step 3: Shuttle from T1 to T2 (1 Split, 1 Move, 1 Merge)
Step 4: Perform 2Q gate in T2

To understand the impact of gate choices, consider a trap with n ions, and say we wish to perform a gate between two ions that are separated by d positions inside the trap. In Figure 2a, $n = 5$ and $d = 3$. With AM and PM gates, gate time linearly increases with d, that is, gates between nearby ion pairs are faster than distant pairs assuming constant laser strength. This is a direct consequence of the weaker interaction strength between distant qubit pairs. On the other hand, for FM gates, duration is independent of d, but it increases linearly with n, that is, for any qubit pair inside the trap, the gate time is constant, but as the gate times get longer as the chain does. These trade-offs are not just in gate duration. Gate reliability worsens linearly with higher gate time and differs for AM, PM, and FM methods. Gate reliability also depends on heating rates, which are a function of the trap capacity and communication topology. Most importantly, since QC applications have diverse gate patterns, these trade-offs are likely to play out differently across applications. It should be noted that none of these trends pose fundamental limits though. While there are methods to remove distance dependence for gate time and implementations with different scaling behavior, we consider the most commonly used pulse modulation techniques and base our studies on well-accepted experimental observations in the field.

Chain reordering within a trap. Another important micro-architectural choice is the method of chain reconfiguration. These operations position the ion at the correct end of the chain before a split operation (see Figure 4). The two standard ways of performing reconfiguration, gate-based swapping and physical ion swapping, are shown in Figure 5. In gate-based swapping (GS), a SWAP gate (implemented using 3 MS gates and some single-qubit gates) is used to swap the quantum states of the desired ions. Hence, the performance and reliability of GS is directly influenced by the method for two-qubit gate implementation. The second method, ion swapping (IS), physically swaps adjacent ions and was recently demonstrated.[9] Each 1-hop IS exchange requires a split operation to isolate the two swapping ions, followed by the physical rotation of the two ions by 180 degrees (shown in Figure 5), followed by a merge to reconstruct the chain

Figure 5. Choices for chain reordering. GS uses a SWAP gate (implemented with 3 MS two-qubit gates) to exchange quantum state of any arbitrary pair of ions within the trap. IS requires hop-by-hop physical swaps.



(split and merge not shown). Similar to communication, split and merge operations for IS operations have performance and reliability overheads.

We ask, What is the best method to implement two-qubit MS gates and chain reordering in near-term QCCD devices? Is the most reliable implementation different across applications? How can application characteristics be used to inform microarchitectural choices?

5. OUR DESIGN TOOLFLOW

To evaluate these design questions, we built the toolflow shown in Figure 3. Our framework takes a QCCD-based TI system design configuration as input, including trap sizes, connectivity, two-qubit gate implementation, and chain reordering method. It uses a set of NISQ application benchmarks to evaluate the candidate architecture. For accurate evaluation, our toolflow uses realistic performance models for individual components of the QCCD architecture, including real-system measurements reported in experimental works and known physical models. Our simulator uses these models to compute application-level metrics such as execution time, reliability, and operation counts along with device-level metrics such as trap heating rates.

5.1. Compiler for QCCD-based TI systems

To evaluate a range of architectures, we require application executions that are optimized for each target architecture, ideally through an automated compiler toolflow. Current QC compilers such as IBM Qiskit or Rigetti Quilc do not support QCCD-based TI systems, so we built a backend compiler which maps and optimizes applications for QCCD systems. The input to the compiler is an application intermediate representation (IR) consisting of a gate sequence with data (qubit) dependencies among gates. Such IR can be obtained from the language frontends of common QC compilers. Using the IR, our compiler first maps the program qubits onto distinct hardware qubits using heuristic techniques, which aim to reduce communication. Next, we route shuttling operations through the shortest paths in the hardware and automatically insert the necessary chain reordering operations. Since multiple shuttles are allowed to execute in parallel on QCCD devices, we implement strategies to avoid congestion at junctions and avoid deadlocks while routing parallel shuttles. The output of our compiler is an executable with primitive QCCD instructions. More details about our compiler and the optimization passes can be found in the full paper.[18]

5.2. Simulator using realistic performance models

Next, we built a simulator to run the applications on the candidate architecture. The inputs to the simulator are the compiled executable, the target QCCD device architecture, and physical performance models for QCCD hardware. The goal of the simulator is to estimate application run time, reliability, and device-level metrics such as trap heating rates.

To measure application run time, our simulator considers known gate performance models, shuttling time models, and parallelism constraints in QCCD systems. The gate and shuttling performance models are derived from real device characterization studies and allow us to accurately model the performance of all primitive operations in the QCCD architecture. In TI systems, gates within a single trap typically execute serially.[19, 24] But, independent ion shuttles can run in parallel with each other, and in parallel with gates in other traps. Considering these constraints, the simulator walks through the instructions in the compiled executable and schedules their execution on the device. The simulation begins with each qubit laid out according to the initial qubit layout specified by the executable. For shuttling operations, the simulator moves ion from one trap to another as specified by the executable. For each instruction, the simulator tracks start and finish times, allowing it to estimate total application runtime at the end of the program.

To measure application reliability, we ideally require a quantum noise simulator. While such noise simulators have been developed, their compute requirements scale exponentially with qubit count and are intractable beyond 50–60 qubits. Moreover, current simulators are specific to superconducting qubits and do not include QCCD system models. Hence, we build a custom simulator for QCCD systems. Our simulator uses known physical models and estimates from real-system experiments to model gate fidelity and trap heating rates from operational and background noise sources.

The simulation starts with each chain in a zero motional mode energy state. When shuttling operations are executed, the motional energy of the ion chains increase (the ions vibrate more because energy is added to the system to move them). The simulator tracks these energy changes using estimates from a physical model. For each gate, the simulator computes the fidelity using a model, which includes errors from chain temperature and background heating. To measure application reliability (fidelity), the simulator computes the product of fidelities for each operation in the program. This model closely approximates real executions and has been experimentally validated on current TI and superconducting systems.

6. EXPERIMENTAL SETUP

6.1. Applications

Table 1 lists the six applications used in our study. This includes near-term applications such as Quantum Approximate Optimization Algorithm (QAOA), classical applications such as Grover's search (SquareRoot), and important kernels like Quantum Fourier Transform (QFT). Google's recent supremacy demonstration used a circuit with 53 qubits and 430 two-qubit gates on real superconducting hardware.[1] Using this as a baseline capability for 50–100 qubit NISQ systems, we selected application instances with 60–80 qubits and 500–4000 two-qubit gates. More details about the application instances can be found in the full version.[18]

6.2. Device configurations

QCCD systems are designed to operate in the regime of 50–200 qubits. Beyond that optical interconnects and other scaling techniques are required to build very large systems with thousands of qubits.[17] We evaluate architectures with 50–200 qubits and consider individual trap capacities in the range of 15–35 ions per trap. To explore communication topologies, we use two device topologies: L6, a device similar to Figure 4 with 6 traps connected in a linear fashion (this is the topology of Honeywell's QCCD system[20]), and G2X3, a grid device similar to Figure 2b with 6 traps arranged in two rows and three columns.[11] To test gate implementations, we consider 4 variants of the MS gate: AM1,[25] AM2,[22] PM,[16] and FM.[13] We also test two variants of chain reordering: GS and IS.

All compilations and simulations are run on an Intel Sky-lake processor (2.6GHz, 12GB RAM) using Python 3.7.

7. ARCHITECTURAL DESIGN EXPLORATION

7.1. Trap capacity choices

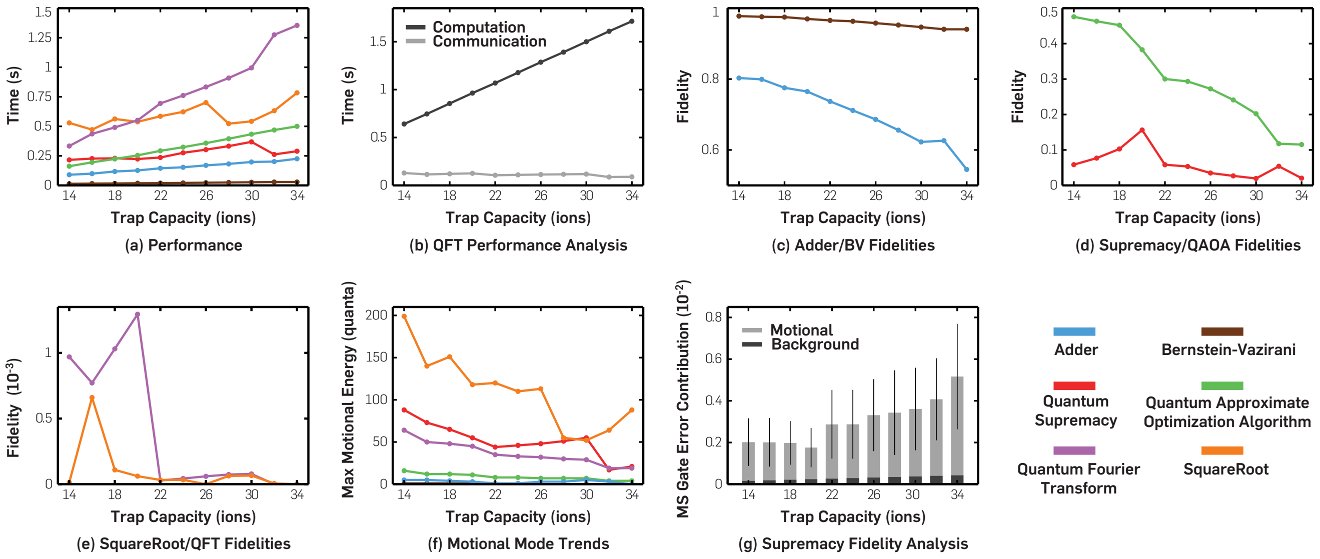
Figure 6 shows the effect of trap sizing on application and device-level metrics. Figure 6a shows the execution time (performance) for the six applications (lower is better). For SquareRoot, Supremacy, and BV, the performance is relatively stable with increasing capacity. This arises because of relative amounts of compute and communication and the different scaling trends for these components. As trap capacity increases, the amount of communication drops. However, the gate time increases because longer duration is necessary to perform entangling gates in large traps. Hence, the overall time remains relatively constant irrespective of trap size. Figure 6b analyses the computation and communication performance for QFT. In this case, computation time is the dominant factor and the total time increases with trap size. Therefore, while it is generally believed that the shuttling time will be a major performance bottleneck for QCCD systems, our work shows that computation and communication performance depend on application characteristics as well as device architecture.

Table 1. Applications used in our study.

Application	Qubits	Two-qubit gates	Communication pattern
Supremacy	64	560	Nearest neighbor gates
QAOA	64	1260	Nearest neighbor gates
SquareRoot	78	1028	Short- and long-range gates
QFT	64	4032	All distances (64*63 gates)
Adder	64	545	Short-range gates
BV	64	64	Short- and long-range gates

Figure 6c–6e show the fidelity of six applications (higher is better). For BV, Adder, and QAOA, fidelity is high even at very low trap capacity because of their low communication requirements. For Supremacy, SquareRoot, and QFT, fidelity is low at small trap capacity (<15 ions), attains a maximum thereafter and drops significantly when the trap capacity is 30 or more. For Supremacy, the best fidelity is 15× higher than the worst, showing the importance of optimizing trap sizing. To analyze the trend, Figure 6f shows the maximum motional mode across the traps in the device (the motional mode quantifies unwanted energy accumulated in an ion chain, higher is worse). The motional energy is high at small capacity because more communication operations are required. Each shuttling operation adds energy to the ion chains, increasing heating, worsening qubit addressability and gate fidelity. Since heating rates reduce with increasing trap capacity, why does gate fidelity worsen at higher capacity?

Figure 6. Trap sizing choices: Experiments use L6 device, with FM two-qubit gates and GS chain reordering. Capacity denotes the maximum number of ions in an individual trap. (a) Application runtime (lower is better). Runtime depends on trap capacity but is also influenced by application characteristics. (b) Trends of computation and communication time for QFT. Communication time decreases with high trap capacity, while computation time increases because of higher gate time in large traps. (c-e) Application fidelity (product of gate fidelities, higher is better). Application fidelity varies dramatically based on individual trap capacity. 15–25 ions per trap work well across applications, with severe fidelity degradation beyond 35 ions. (f) Maximum motional mode energy across the device (unwanted vibrational energy in ion chains, lower is better). Motional mode energy decreases at higher capacity because of reduced communication. (g) Contribution of background heating and motional mode energy to two-qubit gate error rate (error rate is 1—gate fidelity, lower is better). Motional mode energy is the major contributor to heating error. The trend is explained in Section 7.1.



(a) Performance

(b) QFT Performance Analysis

(c) Adder/BV Fidelities

(d) Supremacy/QAOA Fidelities

(e) SquareRoot/QFT Fidelities

(f) Motional Mode Trends

(g) Supremacy Fidelity Analysis

Figure 6g analyses the contribution of background heating and motional mode energy toward two-qubit gate errors for Supremacy. Gate error is dominated by the motional mode error, with only a negligible contribution from background heating. Surprisingly, even though the motional mode energies reduce at larger trap capacity, the thermal contribution to gate error increases with capacity—the error rate increases by 3× for a capacity of 35 ions, compared to 20 ions. This is for two reasons: First, thermal laser beam instabilities increase with trap capacity. This increases the contribution of motional mode error by 1.5× as the trap capacity increases to 35 ions. Second, heating of a long ion chain causes a large motional energy hot spot, worsening all gates in that trap. With small trap capacities, heating effects can effectively be localized to small regions of the device.

Therefore, for maximizing the reliability of QCCD systems, there is a trap capacity sweet spot of 15–25 ions, depending on the application. This capacity minimizes the impact of heating from communication, thermal motion of the laser-beams, and large hot spots on the device. Moreover, this trap sizing also offers very good runtime performance across applications.

TI devices can be easily reconfigured to support fewer ions than the trap maximum capacity, simply by loading fewer ions. Hence, we recommend that QCCD systems should be designed to support up to 20–25 ions per trap. The actual used capacity can be reduced for applications that need only small trap sizes.

7.2. Communication topology choices

Figure 7 compares the execution time and fidelity of linear (L6) and grid (G2X3) communication topologies across applications. For Adder, QFT, Supremacy, and QAOA, the linear topology offers slightly better performance than grid. For SquareRoot, the grid topology offers better performance than linear. Comparing QFT and SquareRoot, SquareRoot has fewer two-qubit operations than QFT, but its communication pattern is more irregular. QFT has a very regular communication pattern where every ion communicates with every other ion in sequence. Hence, QFT maps well onto the linear topology and SquareRoot maps well onto the grid topology. Therefore, for a given architecture, application gate patterns significantly influence runtime performance.

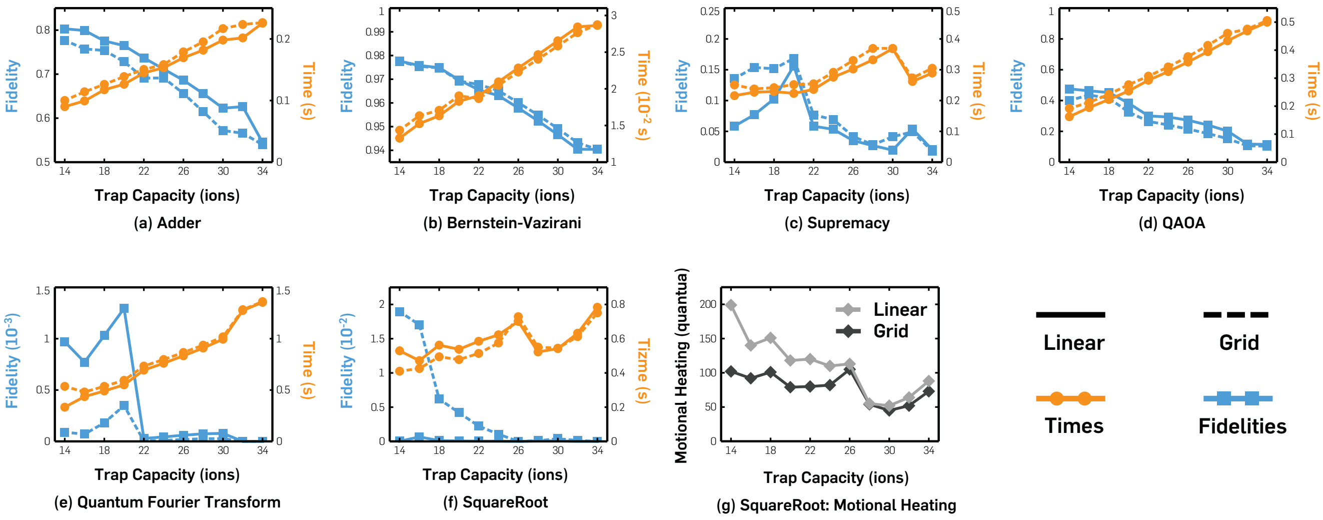
Comparing fidelities, topology has a significant impact on the fidelity of SquareRoot and QFT. For SquareRoot, the grid topology offers up to 7000× higher fidelity than the linear topology. For QFT, the linear topology offers up to 4× higher fidelity than grid. Figure 7g shows the motional mode energies for SquareRoot. The grid topology offers benefits for SquareRoot because it reduces the number of split and merge operations at intermediate traps and therefore accrues less motional heating. The grid topology also allows shorter shuttling paths for the irregular communication pattern of this application, further minimizing unwanted motional energy. For Adder, BV, Supremacy, and QAOA, the impact of topology is less because they are not communication-intensive. In particular, Supremacy and QAOA (we use the hardware-efficient ansatz) are designed for nearest-neighbor connectivity and work well on QCCD systems with linear topology.

Thus, device topology must be co-designed for needs of applications. For NISQ systems, fidelity losses from application-device topology mismatch can be very severe. For nearest-neighbor applications such as QAOA and Supremacy, linear QCCD topologies work well.

8. MICROARCHITECTURAL DESIGN EXPLORATION

Our work also explored application performance and fidelity under eight microarchitecture combinations: four two-qubit gate implementation methods (AM1, AM2, PM, FM) and two chain reordering methods (GS, IS). For this simulation, we used a linear device topology with 6 trapping zones.

Figure 7. Communication topology choices: Figure compares two topologies: L6 and G2x3. Experiments used FM two-qubit gates with GS reordering. (a)–(f) Application runtime (lower is better) and fidelity (higher is better). Topology affects performance, depending on application characteristics. Application fidelity is significantly impacted by communication topology. When application and device topology are well matched, fidelity is boosted by up to 3 orders of magnitude. (g) Motional mode energy for SquareRoot (lower is better, common legend not applicable for this figure). Grid topology offers high fidelity for this application because it reduces communication operations and hence has lower motional mode energy.



(a) Adder

(b) Bernstein-Vazirani

(c) Supremacy

(d) QAOA

(e) Quantum Fourier Transform

(f) SquareRoot

(g) SquareRoot: Motional Heating

We describe the key insights in this section and refer the reader to the full version for details.[18]

Application performance depends on the gate implementation, with up to 5× performance variation across implementations. Thus the best choice of gate differs according to the application. For QAOA where all the two-qubit gates are short range, AM gates perform better than the FM gate. This is because FM gates have high execution times, which increase linearly with the number of ions in the chain. However, FM gate time is independent of the ion separation for a particular two-qubit gate and PM gates only have a weak distance dependence and, therefore, they are suitable for SquareRoot and QFT, which have long range two-qubit operations. Similarly, application fidelity also depends significantly on the two-qubit gate implementation choices. Fidelity varies by up to 9× across implementations, due to different application requirements. QAOA, Supremacy, and Adder benefit from fast and highly-reliable gates at short range; hence, AM2 gates work well. QFT, SquareRoot, and BV have short- and long-range interactions, which are reliably provided by the FM or PM implementations.

Therefore, QCCD systems should support multiple implementations for two-qubit gates to allow applications to be matched to the most suitable implementation. The right choice of gate can improve fidelity by up to 9×. However, this will not require extra hardware; current TI systems already include all the hardware necessary to allow experiments with different gate implementations.[2]

Our studies show that GS chain reordering has superior fidelity to IS. Although fast methods have been developed for IS,[9] our simulations indicate that this method has severe fidelity overheads. With current protocols for reordering, each pair of adjacent ions requires an additional split and merge operation. Applications such as SquareRoot require several reordering operations, especially at small trap sizes, increasing the overheads of IS. GS works well across applications, across FM and AM2 gates, and across different trap sizes, providing vastly superior fidelity compared to IS.

Thus, we recommend that QCCD-based TI systems use gate-based swapping for chain reordering. This method also has the advantage that it can leverage one or more two-qubit gate implementations available for the trap.

9. FUTURE OUTLOOK AND CONCLUSION

With a major thrust to develop QC hardware, superconducting qubits (IBM, Google, Rigetti, and others) and trapped ion (TI) qubits have emerged as strong candidates for large scale QC. Although TI systems have shown considerable promise for application executions, current computer science and systems research largely focuses on superconducting systems. Our work brings the attention of the community to TI-based QC technology and lays out important architectural foundations and opportunities in this space.

TI systems have reached an inflection point in terms of qubit counts, reliability, and compute capabilities. Early TI systems were typically small, having less than 5–10 qubits, but in the past two years, efforts from industry vendors and academic groups have pushed the boundary to 32 qubits in a single ion chain (IonQ). However, experiments with long ion chains (from IonQ) show the difficulties of adding more qubits and demonstrate the need for scaling using modular architectures like QCCD. The first QCCD system was recently demonstrated by Honeywell[20] and several groups are working toward scaling the technology.[7, 10, 23] Our work explores foundational architectural issues such as trap capacity, shuttling topology, and gate implementations for the next generation of devices with 50–100 qubits that are likely to be realized in the coming decade.

Looking beyond TI systems, one of the central insights from our work is the value of architectural design approaches for scaling up QC devices. Current QC devices are largely designed in a “bottom up” fashion, based on physical hardware constraints and low-level physical simulations. While such approaches have been acceptable for small systems, our work shows that QC systems suffer severe reliability penalties if algorithmic success is not also accounted for during design. While classical processors are designed based on application considerations, high-level simulations, and architectural approaches, such approaches are not yet employed in QC.

Our work brings such systematic simulation-driven approaches to designing the next generation of QC systems. By co-designing hardware and applications, we show how to gain over four orders of magnitude (i.e., 10,000×) improvement in application reliability. In the current technology landscape, massive engineering efforts are required to add a few qubits or slightly improve gate error rates. The reliability gains from approaches like ours will therefore be indispensable for future QC systems.

To conclude, our work underscores the important role that computer architects and systems researchers have to play in shaping the future of quantum computing. By leveraging proven architectural techniques and expertise drawn from several decades of optimizing classical processors, we are poised to close large gaps in reliability and performance and significantly accelerate the progress toward practically useful QC.

Acknowledgments

DD would like to thank Pak Hong Leung and Ye Wang for helpful discussions regarding ion trap gates and errors. This work is funded in part by Enabling Practical-scale Quantum Computation (EPiQC), an NSF Expedition in Computing, grants 1730082, 1730104 and Software-Tailored Architecture for Quantum co-design (STAQ) under NSF grant 1717523. PM was also supported by an IBM Ph.D. Fellowship.

References

1. Arute, F., Arya, K., Babbush, R., Bacon, D., Bardin, J.C., Barends, R., Biswas, R., Boixo, S., Brandao, F.G.S.L., Buell, D.A., Burkett, B., Chen, Y., Chen, Z., Chiaro, B., Collins, R., Courtney, W., Dunsworth, A., Farhi, E., Foxen, B., Fowler, A., Gidney, C., Giustina, M., Graff, R., Guerin, K., Habegger, S., Harrigan, M.P., Hartmann, M.J., Ho, A., Hoffmann, M., Huang, T., Humble, T.S., Isakov, S.V., Jeffrey, E., Jiang, Z., Kafri, D., Kechedzhi, K., Kelly, J., Klimov, P.V., Knysh, S., Korotkov, A., Kostritsa, F., Landhuis, D., Lindmark, M., Lucero, E., Lyakh, D., Mandrà, S., McClean, J.R., McEwen, M., Megrant, A., Mi, X., Michielsen, K., Mohseni, M., Mutus, J., Naaman, O., Neeley, M., Neill, C., Niu, M.Y., Ostby, E., Petukhov, A., Platt, J.C., Quintana, C., Rieffel, E.G., Roushan, P., Rubin, N.C., Sank, D., Satzinger, K.J., Smelyanskiy, V., Sung, K.J., Trevithick, M.D., Vainsencher, A., Villalonga, B., White, T., Yao, Z.J., Yeh, P., Zalcman, A., Neven, H., Martinis, J.M. Quantum supremacy using a programmable superconducting processor. *Nature 574*, 7779 (2019), 505–510.
2. Blumel, R., Grzesiak, N., Nam, Y. Power-optimal, stabilized entangling gate between trapped-ion qubits. arXiv:1905.09292 (2019).
3. Bowler, R., Gaebler, J., Lin, Y., Tan, T.R., Hanneke, D., Jost, J.D., Home, J.P.,

Leibfried, D., Wineland, D.J. Coherent diabatic ion transport and separation in a multizone trap array. *Phys. Rev. Lett. 109* (2012), 080502.

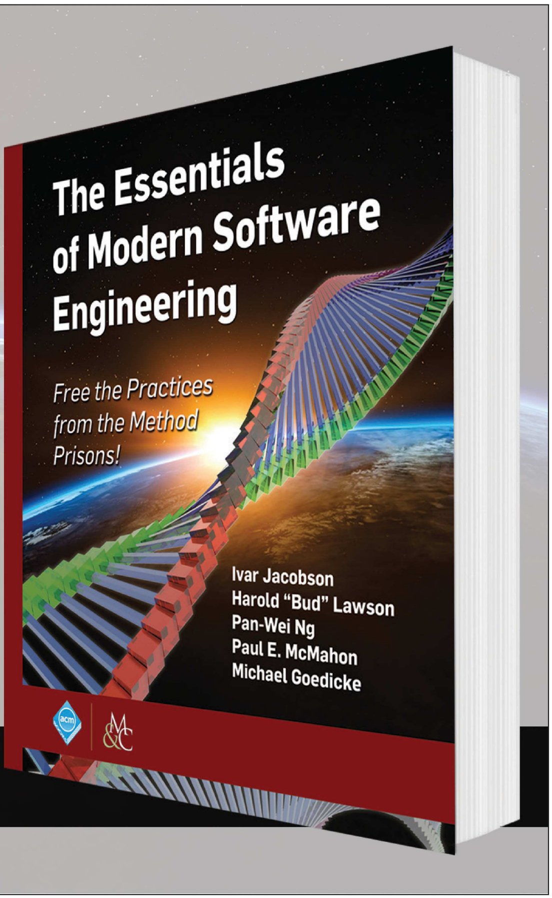
4. Choi, T., Debnath, S., Manning, T.A., Figgatt, C., Gong, Z.-X., Duan, L.-M., Monroe, C. Optimal quantum control of multimode couplings between trapped ion qubits for scalable entanglement. *Phys. Rev. Lett. 112* (2014), 190502.
5. Egan, L., Debroy, D.M., Noel, C., Risinger, A., Zhu, D., Biswas, D., Newman, M., Li, M., Brown, K.R., Cetina, M., Monroe, C. Fault-tolerant operation of a quantum error-correction code. arXiv:2009.11482 (2020).
6. Fu, X., Schouten, R., Almudever, C., DiCarlo, L., Bertels, K., Rol, M., Bultink, C., van Someren, H., Khammassi, N., Ashraf, I., Vermeulen, R., Sterke, J., Vlothuizen, W. An experimental microarchitecture for a superconducting quantum processor. In *Proceedings of the 50th Annual IEEE/ACM International Symposium on Microarchitecture*, MICRO-50'17, Association for Computing Machinery, New York, NY, USA, 2017, 813–825.
7. Holz, P.C., Auchter, S., Stocker, G., Valentini, M., Lakhmanskiy, K., Rössler, C., Stampfer, P., Sgouridis, S., Aschauer, E., Colombe, Y., Blatt, R. 2D Linear trap array for quantum information processing. *Adv. Quantum Technol. 3*, 11 (2020), 2000031.
8. Javadi-Abhari, A., Gokhale, P., Holmes, A., Franklin, D., Brown, K.R., Martonosi, M., Chong, F.T. Optimized surface code communication in superconducting quantum computers. In *Proceedings of the 50th Annual IEEE/ACM International Symposium on Microarchitecture*, MICRO-50'17, ACM, New York, NY, USA, 2017, 692–705.
9. Kaufmann, H., Ruster, T., Schmiegelow, C.T., Luda, M.A., Kaushal, V., Schulz, J., von Lindenfels, D., Schmidt-Kaler, F., Poschinger, U.G. Fast ion swapping for quantum-information processing. *Phys. Rev. A 95* (2017), 052319.
10. Kaushal, V., Lekitsch, B., Stahl, A., Hilder, J., Pijn, D., Schmiegelow, C., Bermudez, A., Müller, M., Schmidt-Kaler, F., Poschinger, U. Shuttling-based trapped-ion quantum information processing. *AVS Quant. Sci. 2*, 1 (2020), 014101.
11. Kielpinski, D., Monroe, C., Wineland, D.J. Architecture for a large-scale ion-trap quantum computer. *Nature 417*, 6890 (2002), 709–711.
12. Leung, P.H., Brown, K.R. Entangling an arbitrary pair of qubits in a long ion crystal. *Phys. Rev. A 98*, 3 (2018), 032318.
13. Leung, P.H., Landsman, K.A., Figgatt, C., Linke, N.M., Monroe, C., Brown, K.R. Robust 2-qubit gates in a linear ion crystal using a frequency-modulated driving force. *Phys. Rev. Lett.* 120, 2 (2018), 020501.
14. Li, G., Ding, Y., Xie, Y. Towards efficient superconducting quantum processor architecture design. In *Proceedings of the Twenty-Fifth International Conference on Architectural Support for Programming Languages and Operating Systems*, ASPLOS'20, Association for Computing Machinery, New York, NY, USA, 2020, 1031–1045.
15. Maunz, P. High Optical Access Trap 2.0. Technical report, Sandia National Lab, SAND-2016-0796R, 2016.
16. Milne, A.R., Edmunds, C.L., Hempel, C., Roy, F., Mavadia, S., Biercuk, M.J. Phase-modulated entangling gates robust to static and time-varying errors. arXiv:1808.10462 (2018).
17. Monroe, C., Kim, J. Scaling the ion trap quantum processor. *Science 339*, 6124 (2013), 1164–1169.
18. Murali, P., Debroy, D.M., Brown, K.R., Martonosi, M. Architecting noisy intermediate-scale trapped ion quantum computers. In *Proceedings of the ACM/IEEE 47th Annual International Symposium on Computer Architecture*, ISCA'20, IEEE Press, Virtual Conference, 2020, 529–542.
19. Murali, P., Linke, N.M., Martonosi, M., Javadi-Abhari, A., Nguyen, N.H., Alderete, C.H. Full-stack, real-system quantum computer studies: architectural comparisons and design insights. In *Proceedings of the 46th International Symposium on Computer Architecture*, ISCA'19, ACM, New York, NY, USA, 2019, 527–540.
20. Pino, J.M., Dreiling, J.M., Figgatt, C., Gaebler, J.P., Moses, S.A., Allman, M.S., Baldwin, C.H., Foss-Feig, M., Hayes, D., Mayer, K., Ryan-Anderson, C., Neyenhuis, B. Demonstration of the trapped-ion quantum CCD computer architecture. *Nature 592*, 7853 (2021), 209–213.
21. Sørensen, A., Mølmer, K. Quantum computation with ions in thermal motion. *Phys. Rev. Lett. 82* (1999), 1971–1974.
22. Trout, C.J., Li, M., Gutiérrez, M., Wu, Y., Wang, S.-T., Duan, L., Brown, K.R. Simulating the performance of a distance-3 surface code in a linear ion trap. *New J. Phys. 20*, 4 (2018), 043038.
23. Wan, Y., Jördens, R., Erickson, S.D., Wu, J.J., Bowler, R., Tan, T.R., Hou, P.-Y., Wineland, D.J., Wilson, A.C., Leibfried, D. Ion transport and reordering in a 2d trap array. *Adv. Quant. Technol. 3*, 11 (2020), 2000028.
24. Wright, K., Beck, K.M., Debnath, S., Amini, J.M., Nam, Y., Grzesiak, N., Chen, J.-S., Pisenti, N.C., Chmielewski, M., Collins, C., Hudek, K.M., Mizrahi, J., Wong-Campos, J.D., Allen, S., Apisdorf, J., Solomon, P., Williams, M., Ducore, A.M., Blinov, A., Kreikemeier, S.M., Chaplin, V., Keesan, M., Monroe, C., Kim, J. Benchmarking an 11-qubit quantum computer. *Nat. Commun. 10*, 1 (2019), 5464.
25. Wu, Y., Wang, S.-T., Duan, L.-M. Noise analysis for high-fidelity quantum entangling gates in an anharmonic linear paul trap. *Phys. Rev. A 97*, 6 (2018), 062325.

Prakash Murali Princeton University, Princeton, NJ, USA.

Dripto M. Debroy Google Quantum AI, Venice, CA, USA.

Kenneth R. Brown Duke University, Durham, NC, USA.

Margaret Martonosi Princeton University, Princeton, NJ, USA.



The Essentials of Modern Software Engineering
Free the Practices from the Method Prisons!
Ivar Jacobson
Harold "Bud" Lawson
Pan-Wei Ng
Paul E. McMahon
Michael Goedicke