Architecting Noisy Intermediate-Scale Trapped Ion Quantum Computers

Prakash Murali
Princeton University

Dripto M. Debroy
Duke University

Kenneth R. Brown
Duke University

Margaret Martonosi
Princeton University

***Abstract*—Trapped ions (TI) are a leading candidate for building Noisy Intermediate-Scale Quantum (NISQ) hardware. TI qubits have fundamental advantages over other technologies such as superconducting qubits, including high qubit quality, coherence and connectivity. However, current TI systems are small in size, with 5-20 qubits and typically use a single trap architecture which has fundamental scalability limitations. To progress towards 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.**

Towards 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 towards practical QC systems.

I. Introduction

Quantum computing (QC) is an emerging paradigm which uses principles of quantum mechanics to manipulate information. In QC, information is represented using *qubits* (quantum bits) and computations are performed using *gates* (operations). Leveraging effects such as superposition, entanglement and interference, QC systems can efficiently explore exponentially large state spaces and compute solutions for certain classically-intractable problems. Practical applications of this paradigm are expected in the near future, particularly in the domains of computational quantum chemistry [1, 2], machine learning [3, 4] and security [5].

QC hardware has progressed rapidly in recent years. Current leading qubit technologies are superconducting qubits [6, 7] and trapped ion (TI) qubits [8–10]; other technologies also being pursued [11–13]. Several superconducting qubit systems having up to 72 qubits have been built [14–16]. TI systems have also been built, with the current largest system having 11 qubits [17]. All these systems have severe resource constraints, including low qubit counts and high operational noise, and therefore are called Noisy Intermediate-Scale Quantum (NISQ) systems. In spite of these limitations, NISQ systems have the potential to demonstrate near-term QC applications [5] especially if they are architected well and used in conjunction with efficient software toolflows.

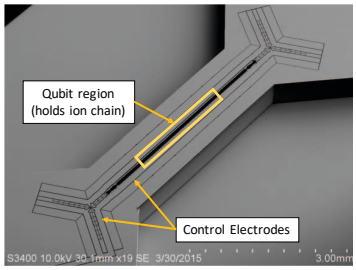
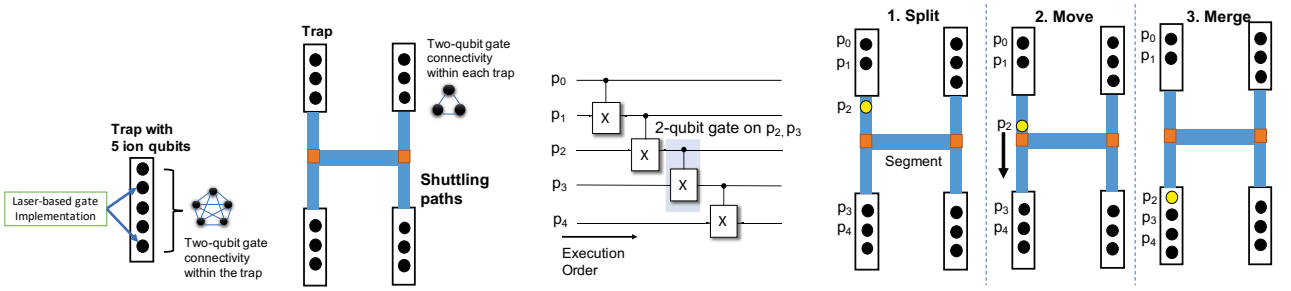


Fig. 1: Scanning electron micrograph of the HOA-2 trap designed and fabricated at Sandia National Laboratories. Figure adapted with permission from [18]. A single trap houses all the ions. Control electrodes are used to load, remove and move ions. This architecture does not scale beyond 50-100 qubits because of gate implementation challenges in long ion chains.

Trapped ion (TI) qubits are one the most promising technology candidates for building NISQ devices. Figure 1 shows a real TI QC system. TI qubits are implemented using the energy states of an atomic ion such as Ca^+ or Yb^+. In a TI system, a set of ions are *trapped* or confined in space using electromagnetic fields. As Figure 2a shows, the ions are arranged in the form of a linear chain, with each ion storing a single qubit. The states of the ions can be manipulated using lasers to implement gate-based computation. Current TI systems with 5-11 qubits have been used to demonstrate near-term applications and quantum error detection [17, 19–21]. Although they are smaller than superconducting systems (pursued by IBM, Google, Rigetti and others), they have fundamental advantages over other technologies, including defect-free identical qubits, very high coherence times [22], and dense qubit connectivity. Indeed, recent comparative studies show ways in which TI systems perform better than superconducting systems of the same size

978-1-7281-4661-4/20/$31.00 ©2020 IEEE
DOI 10.1109/ISCA45697.2020.00051



(a) 5-qubit system with a single trap (b) Modular QCCD system (c) Example program IR (d) Shuttling operation on p_2

Fig. 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.

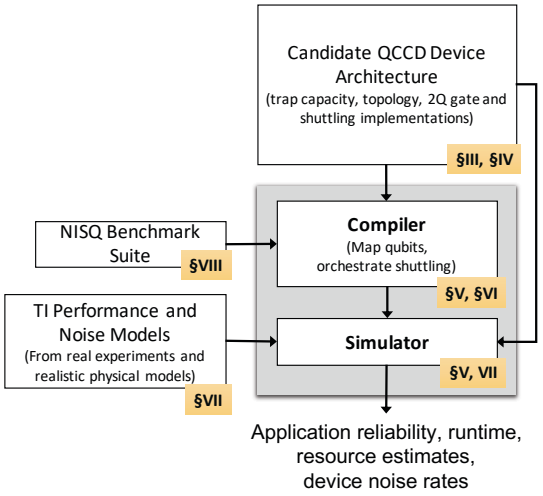


Fig. 3: Our framework for evaluating a candidate QCCD-based TI system. Taking a candidate architecture, a set of NISQ applications, and realistic performance models as input, the toolflow computes application metrics like runtime and reliability (fidelity) and device metrics like heating rates.

[23, 24].

To scale up TI technology for near-term applications, academic and industry efforts are underway to build 50-100 qubit systems [25–27]. Past this scale, architectures based on a single trap are infeasible because of difficulties in qubit control and gate implementation for long ion chains. In light of these difficulties, a modular and scalable architecture called the Quantum Charge Coupled Device (QCCD) was proposed [28]. Figure 2b shows an example. QCCD-based TI systems use multiple traps, with each trap having a small number of ions, allowing reliable gates and full connectivity within each trap. To interconnect traps, QCCD systems use *ion shuttling*, where qubits are physically moved in order to allow communication between traps [29–32]. Figure 2c and 2d show an example shuttling operation. While several other scaling proposals exist in theory [33–36], all basic components required for QCCD systems have been developed and refined over the last decade [17, 32, 37–41], making it a very promising TI scaling path. Recently, Honeywell built the first QCCD system with 4 qubits [42] and shuttling-based systems are also being pursued by other vendors [43].

Although proof-of-concept QCCD systems have been demonstrated, building a large practical system is challenging. On the hardware and architecture side, designers face a wide range of design choices for trap capacity, device topology, gate implementation methods, and shuttling techniques. Currently there is little or no guidance on the performance and reliability tradeoffs of these choices. On the applications end, QC algorithms have widely different qubit and gate counts, error sensitivities, and communication patterns. If hardware is designed in isolation, without considering application characteristics, it will likely result in performance and reliability penalties that are too severe in the NISQ regime. To enable practically useful hardware, computer architecture techniques must be applied to design TI devices that support application requirements well.

To this end, this paper performs an extensive architectural study of modular QCCD-based TI devices targeted for the 50-100 qubit range. Using a suite of NISQ applications, we evaluate a large space of design possibilities including key architectural choices and microarchitectural implementation

methods. Figure 3 shows our design tooflow for evaluating QCCD architectures. Our contributions include:

First, while several works have focused on architecture for superconducting QC systems [44–48], 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.

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 promising applications such as QAOA [49, 50], 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.

II. Background on Quantum Computing

A. 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 $\alpha|0\rangle + \beta|1\rangle$, 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 towards 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 $|1\rangle$.

B. 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 [51, 52]. 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 [8, 53, 54]. 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, captured in Section VII's fidelity 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 [55]. For TI systems, gate fidelities higher than 99% have been achieved in practice [17, 56].

III. Background on QCCD-based TI Systems

A. 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 [57, 58]. 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[1]. Put together, these challenges make it difficult to scale single-trap TI devices beyond tens of qubits.

B. 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

[1]Ref. [58] develops entangling gates on chains of 50 ions, but they see a considerable slowdown in two-qubit gate times.

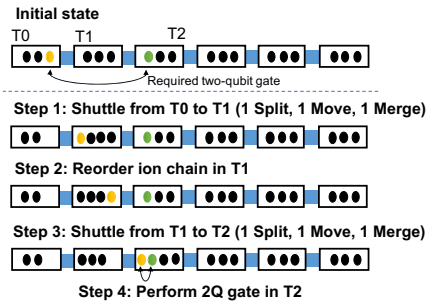


Fig. 4: Shuttling in a QCCD-system which has linear device topology. Extra split and merge operations are required while moving ions through intermediate traps.

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 [32].

IV. Design Tradeoffs in QCCD-based TI Systems

A. 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?*

B. Communcation Topology Choices

QCCD systems have different topology options for orchestrating shuttling operations. To understand the tradeoffs, 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 non-adjacent 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 2x2 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 non-trivial 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?*

C. Gate and Shuttling Implementation Choices

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) [59–61] and frequency modulation (FM) [40, 58] of the laser control pulses. We also consider a recent proposal based on phase modulation (PM) [62].

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, i.e. 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, i.e. for any qubit pair inside the trap, the gate time is constant, but as the gate times get longer as the chain does. These tradeoffs 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 tradeoffs 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

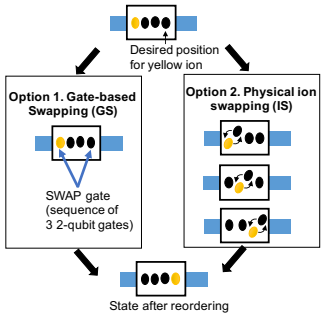


Fig. 5: Choices for chain reordering. GS uses a SWAP gate (implemented with 3 MS two-qubit gates) to exchange quantum state of any arbitrary pair of ions within the trap. IS requires hop-by-hop physical swaps.

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 microarchitectural 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 [63]. 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 (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?*

V. Design Toolflow: Overview

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.

A. 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 [24, 64–66] 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 QC compilers like IBM Qiskit [64, 67] or ScaffCC [68, 69]. 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.

B. 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 (Section VII). In TI systems, gates within a single trap typically execute serially [17, 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 [70], 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 systems [17, 23], on superconducting systems [5], and used in prior works [71, 72].

VI. Optimizations in Our Compiler

The compiler's input IR compiler has a set of single and two-qubit gates, as well as measurement operations. Unlike classical compilation, QC IR does not have control dependencies. It is standard practice to fully unroll all loops and inline all functions; the full instruction sequence is known at compile time [24, 64, 65, 71, 73].

Our compiler first maps the program qubits in the IR onto hardware qubits. For example, for the program in Figure 2c, $\{p_0, p_1, p_2\}$ can be mapped to the first trap, and $\{p_3, p_4\}$ can be mapped to the second trap. The choice of mapping influences the amount of communication. To reduce communication, we use a greedy mapping heuristic adapted from prior work [71, 74]. Our heuristic orders the program qubits according to the sequence in which they are used by the application. It maps each qubit to a trap, co-locating qubits according to trap capacity constraints specified by the architecture. To leave enough buffer space for incoming shuttles, the heuristic ensures that traps are not completely filled (in our experiments, we leave room for 2 incoming ions per trap).

Next, to schedule gates, our compiler uses the earliest-ready-gate-first heuristic [75]. Single-qubit gates on an ion do not need communication and can be scheduled on the trap which holds the ion. For two-qubit gates, the compiler inserts a series of split, move, and merge operations to co-locate the ions if required. To minimize communication, the compiler determines the shortest shuttling path using the device topology. Chain reordering operations are inserted automatically according to the method supported by the target device. For parallel shuttle routing, the compiler avoids deadlocks and manages congestion by leveraging full knowledge of the program instructions and device topology to allocate resources and break dependency cycles. When two shuttles need to access the same segments, we use heuristics to schedule the individual split, move, and merge operations such that no two ion shuttles occupy the same segment at the same time and prioritize earlier gates. To manage congestion at junctions, wait operations are inserted to delay ion shuttles at the intersection when another shuttle is passing or turning through the junction.

VII. Simulation Framework: Performance and Fidelity Models

To effectively study architectural and microarchitectural design, we need an accurate model of the physical behavior of QCCD systems. First, we present performance models for two-qubit gate implementations. Next, we model shuttling performance and trap heating rates. Finally, the gate and shuttling models are combined according to a well-known model to compute the gate fidelity.

A. Gate Time Model

The entangling gate we consider is the Mølmer-Sørensen gate, which is the canonical two-qubit gate on TI systems. This gate creates entanglement between distant ions in a chain by mediating the interaction through the motional mode of the chain [53, 54]. Other popular QC gates such as Controlled NOT are implemented using the MS gate as a low-level primitive [24, 76]. Our work considers four implementation options for the MS gate. The options differ by the laser parameter that is modulated to ensure robust performance on all motional modes. These include gates based on Amplitude Modulation (AM) [59, 61], Phase Modulation [62] and Frequency Modulation (FM) [40, 58]. These gates are standard and implemented in current TI systems [17, 77, 78].

AM Gates: For the AM gates, the operation time of the gate is linearly proportional to the distance between the involved qubits. For clarity we refer to the gate implementation in [59] as AM1, and the gate implementation from [61] as AM2. AM1 gates are slightly more robust to some sources of noise outside of the scope of this study, but as a result are slightly slower overall. Their gate times are well described by the function

$$\tau_{AM1}(d) = 100 * d - 22,$$

where d is the number of ions between the two ions that are being entangled (all times are reported in μs). For the less robust but faster gates AM2 gates [61], we use

$$\tau_{AM2}(d) = 38 * d + 10.$$

PM Gates: Similar to AM gates, the operation time of PM gates can be approximated as being linearly related to the distance between the involved qubits. From [62] we get a scaling of

$$\tau_{PM}(d) = 5 * d + 160.$$

These gates have a much weaker dependence on distance than their AM counterparts, but are slower for nearby qubits.

FM Gates: FM gates have an operation time which is independent of the distance between the two ions, but are instead proportional to the total number of ions in the chain. We assume a gate time of $100\mu s$ for all chains below 12 ions, as extremely fast gates are somewhat sensitive to noise, and then increase linearly from there according to the times given in [40]. From this we get an equation for gate time of

$$\tau_{FM}(N) = max(13.33 * N - 54, 100),$$

where N is the number of ions in the chain.

Operation	Time
Move ion through one segment	$5\mu s$
Splitting operation on a chain	$80\mu s$
Merging an ion with a chain	$80\mu s$
Crossing Y-junction	$100\mu s$
Crossing X-junction	$120\mu s$

TABLE I: Operation times for each shuttling operation, obtained from experimental demonstrations summarized in [79].

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

TABLE II: Applications used in our study.

B. Shuttling Model

Shuttling Time: During shuttling, the trap's electrode voltages are varied appropriately to split the chain and move the ion of interest slowly to the next segment. This slow motion is essential for minimizing the amount of heating present during the operation, but some heating is still unavoidable. Table I gives the times for the various shuttling operations, obtained from real characterization experiments [79]. These operations allow ions to move between chains as needed in order to generate more complex entangled states, while still honoring trap capacity restrictions.

Heating Model: When shuttling a qubit, motion is being introduced to the system via the trapping potentials, and this can cause additional heating of the motional modes of the chain. While our entangling gates do not need the chain to be in a particular motional mode, higher energy states have more vibration, making ideal laser addressing difficult. This leads to a penalty in gate fidelity as energy is added to the system.

In our heating model, every chain is thought of as a quantum oscillator with discrete quantized energy levels. We initialize all chains in the zero energy state of this system and add energy to the system in fractions of the energy difference between these energy levels, known as a quanta. When a chain is split, the energy of the chain is split proportionally to the number of ions in each sub-chain, such that conservation of energy is obeyed. Each sub-chain then gains k_1 quanta of motional energy. Similarly when two chains are merged, the resulting chain has energy equal to the sum of the two chains which are being merged, along with an additional k_1 quanta to account for the energy needed to stop the chains and prevent collisions. Lastly, when an ion is being shuttled, it picks up k_2 quanta of energy per segment it shuttles over, to account for slight imperfections in the shuttling potential, along with fact that the very act of shuttling requires the ion to increase in energy. This model comes from the intuition that the heating is strongest at points where the ion is experiencing higher accelerations, and that adiabatic shuttling has been shown to have high fidelities in practice [37, 80, 81].

In Honeywell's 4-qubit QCCD system, the average heating rate per shuttling operation was measured to be less than 2 quanta per second [42]. Since further improvement will be necessary for realizing 50-100 qubit systems, we assume an order of magnitude lower heating rates and use $k_1 = 0.1$ and $k_2 = 0.01$.

C. Gate Fidelity Model

Assuming ideal addressing, the MS gate's fidelity is independent of the motional mode of the qubits. In practice however, thermal motions from higher motional modes reduce the fidelity of the gate. Additionally, if background heating from the electric fields in the trap occurs during the gate, that gate will fail as the MS gate relies on a constant motional mode during application. These two effects lead to a gate fidelity F defined as:

$$F = 1 - \Gamma\tau - A\left(2\bar{n} + 1\right). \tag{1}$$

Here, Γ is the background heating rate of the trap, τ is the gate duration defined in Section VII-A, $A \propto \frac{N}{ln(N)}$ is a scaling factor on the second term which represents the thermal laser beam instabilities (thermal motion of the laser beams perpendicular to the ion chain), and $\bar{n}$ is the motional mode of the chain (vibrational energy), in units of motional quanta [59]. In other words, fidelity decreases at higher gate durations because of background heating. Fidelity also decreases when the chain's motional energy increases from shuttling operations.

VIII. Experimental Setup

A. Applications

Table II lists the six applications used in our study. Google's recent supremacy demonstration used a circuit with 53 qubits and 430 two-qubit gates on real superconducting hardware [5]. Using this as a baseline capability for 50-100 qubit NISQ systems, we selected applications with 60-80 qubits and 500-4000 two-qubit gates.

The quantum supremacy benchmark is designed to demonstrate a classically-intractable computation on a near-term QC system [5, 82]. Quantum Approximate Optimization Algorithm (QAOA) is an important optimization algorithm with near-term applications [49, 50, 83, 84]. We use the hardware-efficient ansatz for QAOA described in [85]. SquareRoot is an implementation of Grover's search algorithm [86]. Quantum Fourier Transform (QFT) [87] and Adder are important QC kernels. Bernstein-Vazirani (BV) algorithm has been used to characterize current trapped ion systems [17].

We obtained the IR for SquareRoot and QFT from ScaffCC [68, 69], Supremacy from Google Cirq [65], and QAOA, BV and Adder from the toolflow in [88]. Our backend compiler supports an OpenQASM interface [89] which allows us to easily interface with high-level language frontends like Cirq and ScaffCC.

B. 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 [33, 34]. 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 [42]), and **G2X3** a grid device similar to Figure 2b with 6 traps arranged in two rows and three columns [28]. To test gate implementations, we consider 4 variants of the MS gate: **AM1**, **AM2**, **PM**, and **FM**. We also test two variants of chain reordering: **GS** and **IS**.

All compilation and simulation experiments are run on an Intel Skylake processor (2.6GHz, 12GB RAM) using Python 3.7.

IX. Architectural Design Exploration

A. 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 (see FM gate scaling in Section VII-A). 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 [39, 90], our work shows that computation and communication performance depend on application characteristics as well as device architecture.

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 $15\times$ 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 6g analyses the contribution of background heating and motional mode energy towards two-qubit gate errors for Supremacy (see Equation 1). 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 $3\times$ for a capacity of 35 ions, compared to 20 ions. This is for two reasons: First, thermal laser beam instabilities increase with trap capacity (captured by the second term in Equation 1). This increases the contribution of motional mode error by $1.5\times$ 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.

B. 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 $7000\times$ higher fidelity than the linear topology. For QFT, the linear topology offers up to $4\times$ 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)

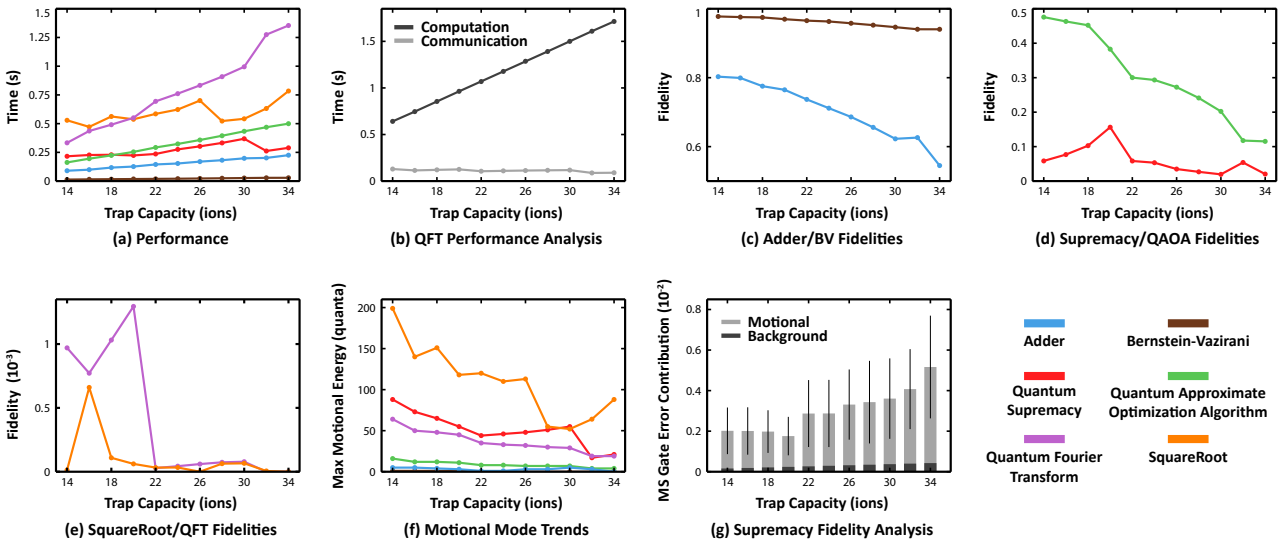


Fig. 6: **Trap Sizing Choices:** Experiments use L6 device, with FM two-qubit gates and GS chain reordering. Capacity denotes the maximum number of ions in an individual trap. (a) Application runtime (lower is better). Runtime depends on trap capacity, but is also influenced by application characteristics. (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 works 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 IX-A.

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.

X. Microarchitectural Design Exploration

Figure 8 shows the performance and fidelity for the six applications 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.

A. Two-qubit Gate Choice

The right half of Figure 8 show the performance of the gate implementations. Application performance depends on the gate implementation, with up to $5\times$ 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.

The left side of Figure 8 shows that application fidelity depends significantly on the two-qubit gate implementation choices (with the GS chain reordering method). Fidelity varies by up to $7\times$ across implementations (not including IS). FM gates obtain up to $9\times$ improvement over AM1 and $1.5\times$ over AM2. In most cases, FM and PM gates have comparable fidelities. For SquareRoot, FM and AM2 gates are comparable and obtain up to $2\times$ improvement over AM1. For QAOA, AM2 gates work well, for Supremacy FM, AM2 and PM gates are comparable and better than AM1. Similar to the performance variations, fidelity varies due to different application requirements. QAOA, Supremacy and Adder benefit from fast gate times at short-range, hence AM2 gates work well. QFT, SquareRoot, BV have short and long-range interactions which are reliably provided by the FM or PM implementations.

Therefore, QCCD systems should support multiple imple-

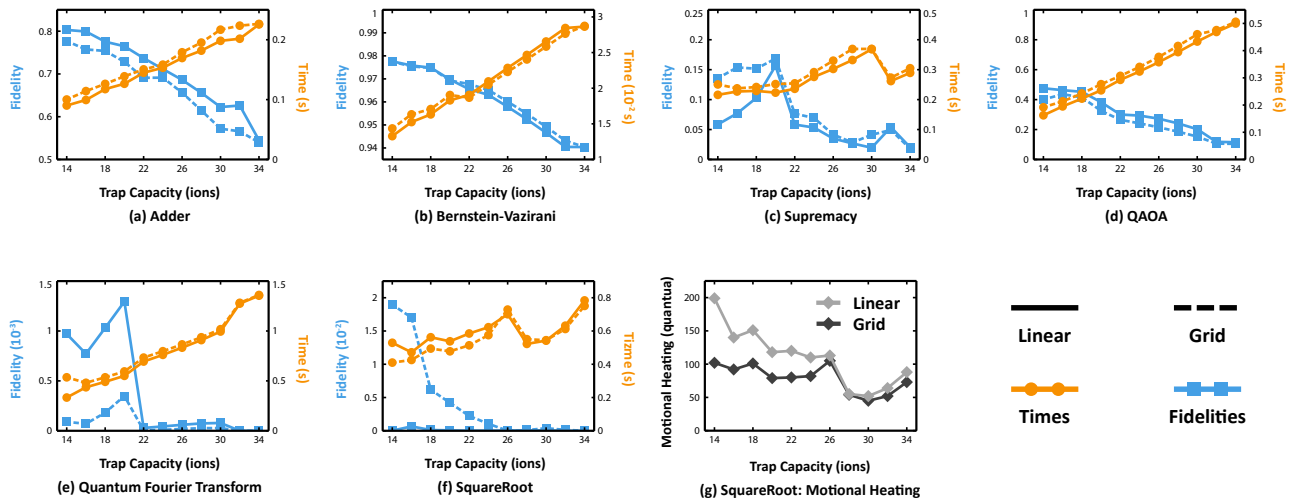


Fig. 7: **Communication topology choices:** Figure compares two topologies: L6 and G2x3. Experiments used FM two-qubit gates with GS reordering. (a)-(f) Application runtime (lower is better) and fidelity (higher is better). Topology affects performance, depending on application characteristics. **Application 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) shows 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.

mentations 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 [77].*

B. Chain Reconfiguration Choices

Figure 8 shows that GS chain reordering has superior fidelity to IS (for QAOA, GS and IS curves match exactly because chain reordering is not required). Although fast methods have been developed for IS [63], 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 used 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.

XI. Related Work

Several works have developed software tools and architecture for superconducting systems. [44–46, 91] developed the QuMA microarchitecture and simulation tools. [47, 48, 71, 92] are other recent simulation and compilation works on superconducting systems. IBM's Qiskit [64], Google's Cirq [65] and Rigetti's Quil [66] platforms are software toolflows for their respective superconducting devices. Overall, superconducting systems have received significant attention recently, in part because several industry players have provided access to real devices. Our work brings renewed attention to TI systems which may offer comparable or higher reliability [23, 24].

Prior works have evaluated real implementations of TI systems to understand architecture design choices. [23] compared the performance of a 5-qubit TI system from University of Maryland (UMD) with a 5-qubit superconducting system from IBM. [24] conducted a larger study, comparing superconducting systems from IBM and Rigetti to the UMD 5-qubit TI system. From [23, 24], the full connectivity of TI systems and powerful primitive operations offer high application success rates compared to other platforms. These real-system studies provide ample evidence to show that TI systems are very promising for NISQ applications. While these works focus on existing devices with less than 20 qubits, our work focuses on the 50-100 qubit range using an automated design and simulation toolflow.

Prior works have also considered very large or fault-tolerant TI devices. [93–95] developed simulation tools for systems with million qubits. [33, 34] developed the MUSIQC architecture which uses optical interconnects to scale to thousands of qubits, [36] proposed a scalable architecture for TI systems based on a reconfigurable optical interconnect. All these works focus on very large or fault tolerant systems which are unlikely to be realized in the next ten years [96].

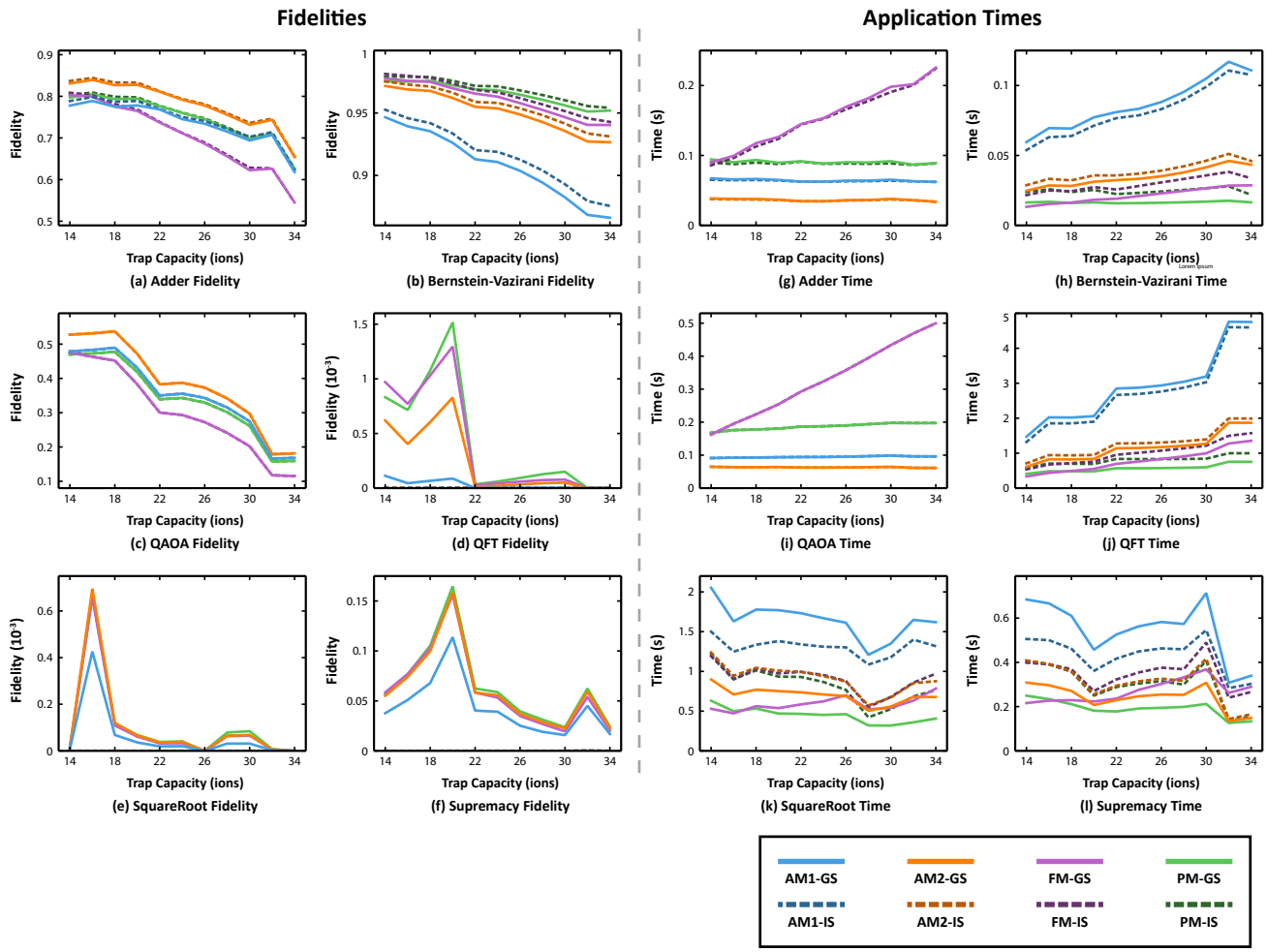


Fig. 8: **Effect of microarchitecture choices on application fidelity and performance.** Experiments use L6 topology. Comparison of 8 combinations with 4 gate choices: AM1, AM2, PM, and FM, and two chain reordering methods: GS, and IS. Application fidelity varies across gate implementations. FM and PM have no or weak distance dependence respectively, so they work well for QFT and SquareRoot which requires several long-range gates. PM or AM2 gates work well for other applications because they offer fast and high-fidelity short-range interactions. For QAOA, GS and IS curves match exactly because chain reordering is not required. For SquareRoot, Supremacy and QFT, IS fidelities are close to 0. Across applications, gate-based swapping is superior to physical swapping of ions in terms of fidelity because the latter requires many more split and merge operations.

XII. Conclusions

Current TI systems use a single-trap architecture where all qubits reside in the same ion chain. Realizing the scaling issues of this design, the Quantum Charge Coupled Device (QCCD) architecture was proposed as a path towards modular TI systems [28]. Over the last two decades, all components required for QCCD systems have been experimentally developed and honed. Recently, Honeywell integrated these techniques to demonstrate the first QCCD system having two traps and four qubits [42]. However, building a practically useful QCCD system is challenging due to the wide range of possible hardware choices and the need to support an evolving mix of QC applications. While performance trends are known in isolation for individual components, there is little guidance available on their system-level performance or impact on applications.

In classical computing, architectural simulations are a key enabler of technology progress, allowing us to predict the performance of the next generation of machines before building them. Our work uses computer architecture and simulation techniques to study scaling TI quantum systems to the next major milestone of 50-100 qubits. We build a design toolflow for the QCCD architecture, including an optimizing compiler and simulator. Using real performance models and device characterization data as inputs to our toolflow, we evaluate application runtime and reliability across several design possibilities. We show that trap sizing, communication topology and microarchitectural

gate choices can impact application reliability by up to four orders of magnitude. As a result, we provide design insights and recommendations for choosing trap sizes, topology, and gate implementations to maximize reliability and performance. With several efforts underway to build large QCCD systems, our work has the potential to guide QC hardware design in the near future.

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.

REFERENCES

[1] A. Peruzzo, J. McClean, P. Shadbolt, M.-H. Yung, X.-Q. Zhou, P. J. Love, A. Aspuru-Guzik, and J. L. O'Brien, "A variational eigenvalue solver on a photonic quantum processor," *Nature Communications*, vol. 5, Jul 2014.

[2] A. Kandala, A. Mezzacapo, K. Temme, M. Takita, M. Brink, J. M. Chow, and J. M. Gambetta, "Hardware-efficient variational quantum eigensolver for small molecules and quantum magnets," *Nature*, vol. 549, Sep 2017.

[3] I. Cong, S. Choi, and M. D. Lukin, "Quantum convolutional neural networks," *Nature Physics*, 2019.

[4] J. Biamonte, P. Wittek, N. Pancotti, P. Rebentrost, N. Wiebe, and S. Lloyd, "Quantum machine learning," *Nature*, vol. 549, Sep 2017.

[5] F. Arute, K. Arya, R. Babbush, D. Bacon, J. C. Bardin, R. Barends, R. Biswas, S. Boixo, F. G. S. L. 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, T. S. Humble, S. V. Isakov, E. Jeffrey, Z. Jiang, D. Kafri, K. Kechedzhi, J. Kelly, P. V. Klimov, S. Knysh, A. Korotkov, F. Kostritsa, D. Landhuis, M. Lindmark, E. Lucero, D. Lyakh, S. Mandrà, J. R. McClean, M. McEwen, A. Megrant, X. Mi, K. Michielsen, M. Mohseni, J. Mutus, O. Naaman, M. Neeley, C. Neill, M. Y. Niu, E. Ostby, A. Petukhov, J. C. Platt, C. Quintana, E. G. Rieffel, P. Roushan, N. C. Rubin, D. Sank, K. J. Satzinger, V. Smelyanskiy, K. J. Sung, M. D. Trevithick, A. Vainsencher, B. Villalonga, T. White, Z. J. Yao, P. Yeh, A. Zalcman, H. Neven, and J. M. Martinis, "Quantum supremacy using a programmable superconducting processor," *Nature*, vol. 574, no. 7779, pp. 505–510, 2019.

[6] C. Rigetti, J. M. Gambetta, S. Poletto, B. L. T. Plourde, J. M. Chow, A. D. Córcoles, J. A. Smolin, S. T. Merkel, J. R. Rozen, G. A. Keefe, M. B. Rothwell, M. B. Ketchen, and M. Steffen, "Superconducting qubit in a waveguide cavity with a coherence time approaching 0.1 ms," *Phys. Rev. B*, vol. 86, p. 100506, Sep 2012.

[7] J. Majer, J. M. Chow, J. M. Gambetta, J. Koch, B. R. Johnson, J. A. Schreier, L. Frunzio, D. I. Schuster, A. A. Houck, A. Wallraff, A. Blais, M. H. Devoret, S. M. Girvin, and R. J. Schoelkopf, "Coupling superconducting qubits via a cavity bus," *Nature*, vol. 449, Sep 2007.

[8] J. I. Cirac and P. Zoller, "Quantum computations with cold trapped ions," *Phys. Rev. Lett.*, vol. 74, pp. 4091–4094, May 1995.

[9] T. P. Harty, D. T. C. Allcock, C. J. Ballance, L. Guidoni, H. A. Janacek, N. M. Linke, D. N. Stacey, and D. M. Lucas, "High-fidelity preparation, gates, memory, and readout of a trapped-ion quantum bit," *Phys. Rev. Lett.*, vol. 113, p. 220501, Nov 2014.

[10] S. Debnath, N. M. Linke, C. Figgatt, K. A. Landsman, K. Wright, and C. Monroe, "Demonstration of a small programmable quantum computer with atomic qubits," *Nature*, vol. 536, Aug 2016.

[11] A. Kitaev, "Fault-tolerant quantum computation by anyons," *Annals of Physics*, vol. 303, no. 1, pp. 2 – 30, 2003.

[12] T. Karzig, C. Knapp, R. M. Lutchyn, P. Bonderson, M. B. Hastings, C. Nayak, J. Alicea, K. Flensberg, S. Plugge, Y. Oreg, C. M. Marcus, and M. H. Freedman, "Scalable designs for quasiparticle-poisoning-protected topological quantum computation with Majorana zero modes," *Phys. Rev. B*, vol. 95, p. 235305, Jun 2017.

[13] J. J. Pla, K. Y. Tan, J. P. Dehollain, W. H. Lim, J. J. L. Morton, D. N. Jamieson, A. S. Dzurak, and A. Morello, "A single-atom electron spin qubit in silicon," *Nature*, vol. 489, Sep 2012.

[14] Google, "A Preview of Bristlecone, Google's New Quantum Processor," https://ai.googleblog.com/2018/03/a-preview-of-bristlecone-googles-new.html, 2018, accessed: 2018-08-05.

[15] IBM, "IBM Announces Advances to IBM Quantum Systems and Ecosystem," https://www-03.ibm.com/press/us/en/pressrelease/53374.wss, 2018, accessed: 2018-08-05.

[16] S. A. Caldwell, N. Didier, C. A. Ryan, E. A. Sete, A. Hudson, P. Karalekas, R. Manenti, M. P. da Silva, R. Sinclair, E. Acala, N. Alidoust, J. Angeles, A. Bestwick, M. Block, B. Bloom, A. Bradley, C. Bui, L. Capelluto, R. Chilcott, J. Cordova, G. Crossman, M. Curtis, S. Deshpande, T. E. Bouayadi, D. Girshovich, S. Hong, K. Kuang, M. Lenihan, T. Manning, A. Marchenkov, J. Marshall, R. Maydra, Y. Mohan, W. O'Brien, C. Osborn, J. Otterbach, A. Papageorge, J.-P. Paquette, M. Pelstring, A. Polloreno, G. Prawiroatmodjo, V. Rawat, M. Reagor, R. Renzas, N. Rubin, D. Russell, M. Rust, D. Scarabelli, M. Scheer, M. Selvanayagam, R. Smith, A. Staley, M. Suska, N. Tezak, D. C. Thompson, T.-W. To, M. Vahidpour, N. Vodrahalli, T. Whyland, K. Yadav, W. Zeng, and C. Rigetti, "Parametrically Activated Entangling Gates Using Transmon Qubits," *Phys. Rev. Applied*, vol. 10, p. 034050, Sep 2018.

[17] K. Wright, K. M. Beck, S. Debnath, J. M. Amini, Y. Nam, N. Grzesiak, J.-S. Chen, N. C. Pisenti, M. Chmielewski, C. Collins, K. M. Hudek, J. Mizrahi, J. D. Wong-Campos, S. Allen, J. Apisdorf, P. Solomon, M. Williams, A. M. Ducore, A. Blinov, S. M. Kreikemeier, V. Chaplin, M. Keesan, C. Monroe, and J. Kim, "Benchmarking an 11-qubit quantum computer," *Nature Communications*, vol. 10, no. 1, p. 5464, Nov 2019.

[18] P. Maunz, "High Optical Access Trap 2.0." Sandia National Lab, SAND-2016-0796R, Tech. Rep., 2016. [Online]. Available: https://www.osti.gov/servlets/purl/1237003

[19] Y. Nam, J.-S. Chen, N. C. Pisenti, K. Wright, C. Delaney, D. Maslov, K. R. Brown, S. Allen, J. M. Amini, J. Apisdorf, K. M. Beck, A. Blinov, V. Chaplin, M. Chmielewski, C. Collins, S. Debnath, K. M. Hudek, A. M. Ducore, M. Keesan, S. M. Kreikemeier, J. Mizrahi, P. Solomon, M. Williams, J. D. Wong-Campos, D. Moehring, C. Monroe, and J. Kim, "Ground-state energy estimation of the water molecule on a trapped-ion quantum computer," *npj Quantum Information*, vol. 6, no. 1, p. 33, Apr 2020.

[20] N. M. Linke, M. Gutierrez, K. A. Landsman, C. Figgatt, S. Debnath, K. R. Brown, and C. Monroe, "Fault-tolerant quantum error detection," *Science advances*, vol. 3, no. 10, p. e1701074, 2017.

[21] J. Chiaverini, D. Leibfried, T. Schaetz, M. D. Barrett, R. Blakestad, J. Britton, W. M. Itano, J. D. Jost, E. Knill, C. Langer *et al.*, "Realization of quantum error correction," *Nature*, vol. 432, no. 7017, p. 602, 2004.

[22] Y. Wang, M. Um, J. Zhang, S. An, M. Lyu, J.-N. Zhang, L.-M. Duan, D. Yum, and K. Kim, "Single-qubit quantum memory exceeding ten-minute coherence time," *Nature Photonics*, vol. 11, no. 10, p. 646, 2017.

[23] N. M. Linke, D. Maslov, M. Roetteler, S. Debnath, C. Figgatt, K. A. Landsman, K. Wright, and C. Monroe, "Experimental comparison of two quantum computing architectures," *Proceedings of the National Academy of Sciences*, vol. 114, no. 13, pp. 3305–3310, 2017.

[24] P. Murali, N. M. Linke, M. Martonosi, A. Javadi-Abhari, N. H. Nguyen, and C. H. Alderete, "Full-stack, Real-system Quantum Computer Studies: Architectural Comparisons and Design Insights," in *Proceedings of the 46th International Symposium on Computer Architecture*, ser. ISCA '19. New York, NY, USA: ACM, 2019, pp. 527–540.

[25] STAQ, "STAQ: Software-Tailored Architectures for Quantum Codesign," http://staq.pratt.duke.edu/, 2019, Research group effort. Accessed: 2019-08-01.

[26] Honeywell, "Honeywell: Our First-Generation Ion Traps Emerge," https://www.honeywell.com/en-us/newsroom/news/2018/11/our-first-generation-ion-traps-emerge, 2019, accessed: 2019-08-01.

[27] IonQ, "IonQ," https://ionq.com, 2019, accessed: 2019-08-01.

[28] D. Kielpinski, C. Monroe, and D. J. Wineland, "Architecture for a large-scale ion-trap quantum computer," *Nature*, vol. 417, no. 6890, pp. 709–711, 2002.

[29] M. Rowe, A. Ben-Kish, B. Demarco, D. Leibfried, V. Meyer, J. Beall, J. Britton, J. Hughes, W. Itano, B. Jelenkovic, C. Langer, T. Rosenband, and D. Wineland, "Transport of quantum states and separation of ions in a dual rf ion trap," *Quantum Information and Computation*, vol. 2, pp. 257–271, 07 2002.

[30] W. Hensinger, S. Olmschenk, D. Stick, D. Hucul, M. Yeo, M. Acton, L. Deslauriers, C. Monroe, and J. Rabchuk, "T-junction ion trap array for two-dimensional ion shuttling, storage, and manipulation," *Applied Physics Letters*, vol. 88, no. 3, p. 034101, 2006.

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

[32] R. Bowler, J. Gaebler, Y. Lin, T. R. Tan, D. Hanneke, J. D. Jost, J. P. Home, D. Leibfried, and D. J. Wineland, "Coherent diabatic ion transport and separation in a multizone trap array," *Phys. Rev. Lett.*, vol. 109, p. 080502, Aug 2012.

[33] C. Monroe and J. Kim, "Scaling the ion trap quantum processor," *Science*, vol. 339, no. 6124, pp. 1164–1169, 2013.

[34] C. Monroe, R. Raussendorf, A. Ruthven, K. R. Brown, P. Maunz, L.-M. Duan, and J. Kim, "Large-scale modular quantum-computer architecture with atomic memory and photonic interconnects," *Phys. Rev. A*, vol. 89, p. 022317, Feb 2014.

[35] T. S. Metodi, D. D. Thaker, A. W. Cross, F. T. Chong, and I. L. Chuang, "A quantum logic array microarchitecture: Scalable quantum data movement and computation," in *38th Annual IEEE/ACM International Symposium on Microarchitecture (MICRO'05)*, Nov 2005, pp. 12 pp.–318.

[36] S. Sargaran and N. Mohammadzadeh, "Saqip: A scalable architecture for quantum information processors," *ACM Trans. Archit. Code Optim.*, vol. 16, no. 2, pp. 12:1–12:21, Apr. 2019.

[37] R. B. Blakestad, C. Ospelkaus, A. P. VanDevender, J. H. Wesenberg, M. J. Biercuk, D. Leibfried, and D. J. Wineland, "Near-ground-state transport of trapped-ion qubits through a multidimensional array," *Phys. Rev. A*, vol. 84, p. 032314, Sep 2011. [Online]. Available: https://link.aps.org/doi/10.1103/PhysRevA.84.032314

[38] K. Wright, J. M. Amini, D. L. Faircloth, C. Volin, S. C. Doret, H. Hayden, C.-S. Pai, D. W. Landgren, D. Denison, T. Killian, R. E. Slusher, and A. W. Harter, "Reliable transport through a microfabricatedX-junction surface-electrode ion trap," *New Journal of Physics*, vol. 15, no. 3, p. 033004, mar 2013.

[39] A. K. Ratcliffe, R. L. Taylor, J. J. Hope, and A. R. R. Carvalho, "Scaling trapped ion quantum computers using fast gates and microtraps," *Phys. Rev. Lett.*, vol. 120, p. 220501, May 2018.

[40] P. H. Leung, K. A. Landsman, C. Figgatt, N. M. Linke, C. Monroe, and K. R. Brown, "Robust 2-qubit gates in a linear ion crystal using a frequency-modulated driving force," *Physical review letters*, vol. 120, no. 2, p. 020501, 2018.

[41] Y. Wan, R. Jördens, S. D. Erickson, J. J. Wu, R. Bowler, T. R. Tan, P. Y. Hou, D. J. Wineland, A. C. Wilson, and D. Leibfried, "Ion transport and reordering in a two-dimensional trap array," *arXiv preprint arXiv:2003.03520*, 2020.

[42] J. M. Pino, J. M. Dreiling, C. Figgatt, J. P. Gaebler, S. A. Moses, C. H. Baldwin, M. Foss-Feig, D. Hayes, K. Mayer, C. Ryan-Anderson, and B. Neyenhuis, "Demonstration of the qccd trapped-ion quantum computer architecture," *arXiv preprint arXiv:2003.01293*, 2020.

[43] V. Kaushal, B. Lekitsch, A. Stahl, J. Hilder, D. Pijn, C. Schmiegelow, A. Bermudez, M. Müller, F. Schmidt-Kaler, and U. Poschinger, "Shuttling-based trapped-ion quantum information processing," *arXiv preprint arXiv:1912.04712*, 2019.

[44] X. Fu, M. A. Rol, C. C. Bultink, J. van Someren, N. Khammassi, I. Ashraf, R. F. L. Vermeulen, J. C. de Sterke, W. J. Vlothuizen, R. N. Schouten, C. G. Almudever, L. DiCarlo, and K. Bertels, "A microarchitecture for a superconducting quantum processor," *IEEE Micro*, vol. 38, no. 3, pp. 40–47, May 2018.

[45] X. Fu, R. Schouten, C. Almudever, L. DiCarlo, K. Bertels, M. Rol, C. Bultink, H. van Someren, N. Khammassi, I. Ashraf, R. Vermeulen, J. Sterke, and W. Vlothuizen, "An experimental microarchitecture for a superconducting quantum processor," in *Proceedings of the 50th Annual IEEE/ACM International Symposium on Microarchitecture*, ser. MICRO-50 '17, 2017, p. 813–825.

[46] X. Fu, L. Riesebos, M. A. Rol, J. van Straten, J. van Someren, N. Khammassi, I. Ashraf, R. F. L. Vermeulen, V. Newsum, K. K. L. Loh, J. C. de Sterke, W. J. Vlothuizen, R. N. Schouten, C. G. Almudever, L. DiCarlo, and K. Bertels, "eQASM: An Executable Quantum Instruction Set Architecture," 2018.

[47] A. Javadi-Abhari, P. Gokhale, A. Holmes, D. Franklin, K. R. Brown, M. Martonosi, and F. T. Chong, "Optimized surface code communication in superconducting quantum computers," in *Proceedings of the 50th Annual IEEE/ACM International Symposium on Microarchitecture*, ser. MICRO-50 '17. New York, NY, USA: ACM, 2017, pp. 692–705.

[48] G. Li, Y. Ding, and Y. Xie, "SANQ: A Simulation Framework for Architecting Noisy Intermediate-Scale Quantum Computing System," *arXiv preprint arXiv:1904.11590*, 2019.

[49] E. Farhi and A. W. Harrow, "Quantum supremacy through the quantum approximate optimization algorithm," *arXiv preprint arXiv:1602.07674*, 2016.

[50] E. Farhi, J. Goldstone, and S. Gutmann, "A quantum approximate optimization algorithm," *arXiv preprint arXiv:1411.4028*, 2014.

[51] R. Ozeri, "The trapped-ion qubit tool box," *Contemporary Physics*, vol. 52, no. 6, pp. 531–550, 2011.

[52] D. J. Wineland, C. Monroe, W. M. Itano, D. Leibfried, B. E. King, and D. M. Meekhof, "Experimental issues in coherent quantum-state manipulation of trapped atomic ions," *Journal of Research of the National Institute of Standards and Technology*, vol. 103, no. 3, p. 259, 1998.

[53] K. Mølmer and A. Sørensen, "Multiparticle entanglement of hot trapped ions," *Phys. Rev. Lett.*, vol. 82, pp. 1835–1838, Mar 1999.

[54] A. Sørensen and K. Mølmer, "Quantum computation with ions in thermal motion," *Phys. Rev. Lett.*, vol. 82, pp. 1971–1974, Mar 1999.

[55] E. Magesan, J. M. Gambetta, and J. Emerson, "Characterizing quantum gates via randomized benchmarking," *Phys. Rev. A*, vol. 85, p. 042311, Apr 2012.

[56] C. J. Ballance, T. P. Harty, N. M. Linke, M. A. Sepiol, and D. M. Lucas, "High-fidelity quantum logic gates using trapped-ion hyperfine qubits," *Phys. Rev. Lett.*, vol. 117, p. 060504, Aug 2016.

[57] J. Zhang, G. Pagano, P. W. Hess, A. Kyprianidis, P. Becker, H. Kaplan, A. V. Gorshkov, Z.-X. Gong, and C. Monroe, "Observation of a many-body dynamical phase transition with a 53-qubit quantum simulator," *Nature*, vol. 551, no. 7682, pp. 601–604, 2017.

[58] P. H. Leung and K. R. Brown, "Entangling an arbitrary pair of qubits in a long ion crystal," *Physical Review A*, vol. 98, no. 3, p. 032318, 2018.

[59] Y. Wu, S.-T. Wang, and L.-M. Duan, "Noise analysis for high-fidelity quantum entangling gates in an anharmonic linear paul trap," *Physical Review A*, vol. 97, no. 6, p. 062325, 2018.

[60] T. Choi, S. Debnath, T. A. Manning, C. Figgatt, Z.-X. Gong, L.-M. Duan, and C. Monroe, "Optimal quantum control of multimode couplings between trapped ion qubits for scalable entanglement," *Phys. Rev. Lett.*, vol. 112, p. 190502, May 2014.

[61] C. J. Trout, M. Li, M. Gutiérrez, Y. Wu, S.-T. Wang, L. Duan, and K. R. Brown, "Simulating the performance of a distance-3 surface code in a linear ion trap," *New Journal of Physics*, vol. 20, no. 4, p. 043038, 2018.

[62] A. R. Milne, C. L. Edmunds, C. Hempel, F. Roy, S. Mavadia, and M. J. Biercuk, "Phase-modulated entangling gates robust to static and time-varying errors," *arXiv preprint arXiv:1808.10462*, 2018.

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

[64] H. Abraham, I. Y. Akhalwaya, G. Aleksandrowicz, T. Alexander, G. Alexandrowics, E. Arbel, A. Asfaw, C. Azaustre, AzizNgoueya, P. Barkoutsos, G. Barron, L. Bello, Y. Ben-Haim, D. Bevenius, L. S. Bishop, S. Bosch, S. Bravyi, D. Bucher, F. Cabrera, P. Calpin, L. Capelluto, J. Carballo, G. Carrascal, A. Chen, C.-F. Chen, R. Chen, J. M. Chow, C. Claus, C. Clauss, A. J. Cross, A. W. Cross, S. Cross, J. Cruz-Benito, C. Culver, A. D. Córcoles-Gonzales, S. Dague, T. E. Dandachi, M. Dartiailh, DavideFrr, A. R. Davila, D. Ding, J. Doi, E. Drechsler, Drew, E. Dumitrescu, K. Dumon, I. Duran, K. EL-Safty, E. Eastman, P. Eendebak, D. Egger, M. Everitt, P. M. Fernández, A. H. Ferrera, A. Frisch, A. Fuhrer, J. Gacon, Gadi, B. G. Gago, J. M. Gambetta, D. Greenberg, D. Grinko, W. Guan, J. A. Gunnels, I. Haide, I. Hamamura, V. Havlicek, J. Hellmers, Ł. Herok, S. Hillmich, H. Horii, C. Howington, S. Hu, W. Hu, H. Imai, T. Imamichi, K. Ishizaki, R. Iten, T. Itoko, A. Javadi-Abhari, Jessica, K. Johns, T. Kachmann, N. Kanazawa, Kang-Bae, A. Karazeev, P. Kassebaum, S. King, Knabberjoe, A. Kovyrshin, V. Krishnan, K. Krsulich, G. Kus, R. LaRose, R. Lambert, J. Latone, S. Lawrence, D. Liu, P. Liu, Y. Maeng, A. Malyshev, J. Marecek, M. Marques, D. Mathews, A. Matsuo, D. T. McClure, C. McGarry, D. McKay, D. McPherson, S. Meesala, M. Mevissen, A. Mezzacapo, R. Midha, Z. Minev, A. Mitchell, N. Moll, M. D. Mooring, R. Morales, N. Moran, P. Murali, J. Müggenburg, A. Phan, M. Pistoia, A. Pozas-iKerstjens, V. Prutyanov, D. Puzzuoli, M. Reuter, J. Rice, D. M. Rodríguez, M. Rossmannek, M. Ryu, N. Sathaye, B. Schmitt, C. Schnabel, Z. Schoenfeld, T. L. Scholten, E. Schoute, J. Schwarm, I. F. Sertage,

K. Setia, N. Shammah, Y. Shi, A. Silva, A. Simonetto, N. Singstock, Y. Siraichi, I. Sitdikov, S. Sivarajah, M. B. Sletfjerding, J. A. Smolin, M. Soeken, I. O. Sokolov, SooluThomas, D. Steenken, M. Stypulkoski, J. Suen, H. Takahashi, I. Tavernelli, C. Taylor, P. Taylour, S. Thomas, M. Tillet, M. Tod, E. de la Torre, K. Trabing, M. Treinish, TrishaPe, W. Turner, Y. Vaknin, C. R. Valcarce, F. Varchon, A. C. Vazquez, D. Vogt-Lee, C. Vuillot, J. Weaver, R. Wieczorek, J. A. Wildstrom, R. Wille, E. Winston, J. J. Woehr, S. Woerner, R. Woo, C. J. Wood, R. Wood, S. Wood, J. Wootton, D. Yeralin, R. Young, J. Yu, C. Zachow, and L. Zdanski, "Qiskit: An open-source framework for quantum computing," 2019.

[65] Google, "Cirq," https://github.com/quantumlib/Cirq, 2018, accessed: 2018-11-29.

[66] Rigetti, "PyQuil," https://github.com/rigetti/quilc, 2019, accessed: 2019-08-06.

[67] IBM, "Qiskit Terra," https://github.com/Qiskit/qiskit-terra, 2017, accessed: 2019-08-01.

[68] A. Javadi-Abhari, S. Patil, D. Kudrow, J. Heckey, A. Lvov, F. T. Chong, and M. Martonosi, "ScaffCC: A Framework for Compilation and Analysis of Quantum Computing Programs," in *Proceedings of the 11th ACM Conference on Computing Frontiers*, ser. CF '14. ACM, 2014, pp. 1:1–1:10.

[69] ScaffCC, "ScaffCC Compiler," https://github.com/epiqc/ScaffCC, 2018, accessed: 2018-05-16.

[70] IBM, "Qiskit Aer Simulator," https://qiskit.org/aer, 2019, accessed: 2019-08-01.

[71] P. Murali, J. M. Baker, A. Javadi-Abhari, F. T. Chong, and M. Martonosi, "Noise-Adaptive Compiler Mappings for Noisy Intermediate-Scale Quantum Computers," in *Proceedings of the Twenty-Fourth International Conference on Architectural Support for Programming Languages and Operating Systems*, ser. ASPLOS '19. New York, NY, USA: Association for Computing Machinery, 2019, p. 1015–1029.

[72] S. Nishio, Y. Pan, T. Satoh, H. Amano, and R. V. Meter, "Extracting Success from IBM's 20-Qubit Machines Using Error-Aware Compilation," *arXiv preprint arXiv:1903.10963*, 2019.

[73] A. Javadi-Abhari, A. Faruque, M. J. Dousti, L. Svec, O. Catu, A. Chakrabati, C.-F. Chiang, S. Vanderwilt, J. Black, F. Chong, M. Martonosi, M. Suchara, K. Brown, M. Pedram, and T. Brun, "Scaffold: Quantum programming language," Princeton University, Report TR-934-12, 2012.

[74] M. Y. Siraichi, V. F. d. Santos, S. Collange, and F. M. Q. Pereira, "Qubit Allocation," in *Proceedings of the 2018 International Symposium on Code Generation and Optimization*, ser. CGO 2018. ACM, 2018, pp. 113–125.

[75] J. Heckey, S. Patil, A. Javadi-Abhari, A. Holmes, D. Kudrow, K. R. Brown, D. Franklin, F. T. Chong, and M. Martonosi, "Compiler management of communication and parallelism for quantum computation," in *Proceedings of the Twentieth International Conference on Architectural Support for Programming Languages and Operating Systems*, ser. ASPLOS '15. ACM, 2015, pp. 445–456.

[76] D. Maslov, "Basic circuit compilation techniques for an ion-trap quantum machine," *New J. Phys.*, vol. 19, no. 2, p. 023035, 2017.

[77] R. Blumel, N. Grzesiak, and Y. Nam, "Power-optimal, stabilized entangling gate between trapped-ion qubits," *arXiv preprint arXiv:1905.09292*, 2019.

[78] H. Ball, M. J. Biercuk, A. Carvalho, R. Chakravorty, J. Chen, L. A. de Castro, S. Gore, D. Hover, M. Hush, P. J. Liebermann *et al.*, "Software tools for quantum control: Improving quantum computer performance through noise and error suppression," *arXiv preprint arXiv:2001.04060*, 2020.

[79] M. Gutiérrez, M. Müller, and A. Bermúdez, "Transversality and lattice surgery: Exploring realistic routes toward coupled logical qubits with trapped-ion quantum processors," *Phys. Rev. A*, vol. 99, p. 022330, Feb 2019.

[80] G. Huber, T. Deuschle, W. Schnitzler, R. Reichle, K. Singer, and F. Schmidt-Kaler, "Transport of ions in a segmented linear paul trap in printed-circuit-board technology," *New Journal of Physics*, vol. 10, no. 1, p. 013004, 2008.

[81] J. F. Eble, S. Ulm, P. Zahariev, F. Schmidt-Kaler, and K. Singer, "Feedback-optimized operations with linear ion crystals," *J. Opt. Soc. Am. B*, vol. 27, no. 6, pp. A99–A104, Jun 2010.

[82] I. L. Markov, A. Fatima, S. V. Isakov, and S. Boixo, "Quantum supremacy is both closer and farther than it appears," *arXiv preprint arXiv:1807.10749*, 2018.

[83] G. Pagano, A. Bapat, P. Becker, K. S. Collins, A. De, P. W. Hess, H. B. Kaplan, A. Kyprianidis, W. L. Tan, C. Baldwin, L. T. Brady, A. Deshpande, F. Liu, S. Jordan, A. V. Gorshkov, and C. Monroe, "Quantum approximate optimization with a trapped-ion quantum simulator," *arXiv preprint arXiv:1906.02700*, 2019.

[84] F. Arute, K. Arya, R. Babbush, D. Bacon, J. C. Bardin, R. Barends, S. Boixo, M. Broughton, B. B. Buckley, D. A. Buell, B. Burkett, N. Bushnell, Y. Chen, Z. Chen, B. Chiaro, R. Collins, W. Courtney, S. Demura, A. Dunsworth, E. Farhi, A. Fowler, B. Foxen, C. Gidney, M. Giustina, R. Graff, S. Habegger, M. P. Harrigan, A. Ho, S. Hong, T. Huang, L. B. Ioffe, S. V. Isakov, E. Jeffrey, Z. Jiang, C. Jones, D. Kafri, K. Kechedzhi, J. Kelly, S. Kim, P. V. Klimov, A. N. Korotkov, F. Kostritsa, D. Landhuis, P. Laptev, M. Lindmark, M. Leib, E. Lucero, O. Martin, J. M. Martinis, J. R. McClean, M. McEwen, A. Megrant, X. Mi, M. Mohseni, W. Mruczkiewicz, J. Mutus, O. Naaman, M. Neeley, C. Neill, F. Neukart, H. Neven, M. Y. Niu, T. E. O'Brien, B. O'Gorman, E. Ostby, A. Petukhov, H. Putterman, C. Quintana, P. Roushan, N. C. Rubin, D. Sank, K. J. Satzinger, A. Skolik, V. Smelyanskiy, D. Strain, M. Streif, K. J. Sung, M. Szalay, A. Vainsencher, T. White, Z. J. Yao, P. Yeh, A. Zalcman, and L. Zhou, "Quantum approximate optimization of non-planar graph problems on a planar superconducting processor," *arXiv preprint arXiv:2004.04197*, 2020.

[85] 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, and K. Temme, "Quantum optimization using variational algorithms on near-term quantum devices," *Quantum Science and Technology*, vol. 3, no. 3, p. 030503, jun 2018.

[86] L. K. Grover, "A fast quantum mechanical algorithm for database search," in *Proceedings of the Twenty-eighth Annual ACM Symposium on Theory of Computing*, ser. STOC '96. ACM, 1996, pp. 212–219.

[87] E. Bernstein and U. Vazirani, "Quantum complexity theory," in *Proceedings of the Twenty-fifth Annual ACM Symposium on Theory of Computing*, ser. STOC '93. ACM, 1993, pp. 11–20.

[88] Teague Tomesh, "Quantum Circuit Generator," https://github.com/teaguetomesh/quantum_circuit_generator, 2019, accessed: 2019-10-27.

[89] A. W. Cross, L. S. Bishop, J. A. Smolin, and J. M. Gambetta, "Open quantum assembly language," *arXiv preprint arXiv:1707.03429*, 2017.

[90] C. D. Bruzewicz, J. Chiaverini, R. McConnell, and J. M. Sage, "Trapped-ion quantum computing: Progress and challenges," *Applied Physics Reviews*, vol. 6, no. 2, p. 021314, 2019.

[91] M. Zhang, "QuMAsim : A Quantum Architecture Simulation and Verification Platform," Masters dissertation, Delft University of Technology, 2018.

[92] S. S. Tannu and M. K. Qureshi, "Not All Qubits Are Created Equal: A Case for Variability-Aware Policies for NISQ-Era Quantum Computers," in *Proceedings of the Twenty-Fourth International Conference on Architectural Support for Programming Languages and Operating Systems*, ser. ASPLOS '19. Association for Computing Machinery, 2019, p. 987–999.

[93] M. Ahsan, B. Choi, and J. Kim, "Performance simulator based on hardware resources constraints for ion trap quantum computer," in *2013 IEEE 31st International Conference on Computer Design (ICCD)*, Oct 2013, pp. 411–418.

[94] M. Ahsan, R. V. Meter, and J. Kim, "Designing a Million-Qubit Quantum Computer Using a Resource Performance Simulator," *J. Emerg. Technol. Comput. Syst.*, vol. 12, no. 4, pp. 39:1–39:25, Dec. 2015.

[95] M. Ahsan and J. Kim, "Optimization of quantum computer architecture using a resource-performance simulator," in *Proceedings of the 2015 Design, Automation & Test in Europe Conference & Exhibition*, ser. DATE '15, 2015, pp. 1108–1113.

[96] NASEM, "Quantum Computing: Progress and Prospects," 2019. [Online]. Available: https://www.nap.edu/catalog/25196/quantum-computing-progress-and-prospects