Satisfiability Modulo Theories-Based Qubit Mapping for Trapped-Ion Quantum Computing Systems

Wei-Hsiang Tseng
Graduate Institute of Electronics Engineering, National Taiwan University
Taipei, Taiwan
whtseng@eda.ee.ntu.edu.tw

Yao-Wen Chang
Graduate Institute of Electronics Engineering and Department of Electrical Engineering, National Taiwan University
Taipei, Taiwan
ywchang@ntu.edu.tw

Jie-Hong Roland Jiang
Graduate Institute of Electronics Engineering and Department of Electrical Engineering, National Taiwan University
Taipei, Taiwan
jhjiang@ntu.edu.tw

ABSTRACT

Qubit mapping is crucial in optimizing the performance of quantum algorithms for physical executions on quantum computing architectures. Many qubit mapping algorithms have been proposed for superconducting systems recently. However, due to their limitations on the physical qubit connectivity, costly SWAP gates are often required to swap logical qubits for proper quantum operations. Trapped-ion systems have emerged as an alternative quantum computing architecture and have gained much recent attention due to their relatively long coherence time, high-fidelity gates, and good scalability for multi-qubit coupling. However, the qubit mapping of the new trapped-ion systems remains a relatively untouched research problem. This paper proposes a new coupling constraint graph with multi-pin nets to model the unique constraints and connectivity patterns in one-dimensional trapped-ion systems. To minimize the time steps for quantum circuit execution satisfying the coupling constraints for trapped-ion systems, we devise a divide-and-conquer solution using Satisfiability Modulo Theories for efficient qubit mapping on trapped-ion quantum computing architectures. Experimental results demonstrate the superiority of our approach in scalability and effectiveness compared to the previous work.

CCS CONCEPTS

• **Computer systems organization → Quantum computing**; • **Hardware → Physical design (EDA)**.

KEYWORDS

Quantum computing, Trapped-ion system, Qubit mapping, Satisfiability Modulo Theories, Divide-and-conquer, Coupling constraint graph

ACM Reference Format:
Wei-Hsiang Tseng, Yao-Wen Chang, and Jie-Hong Roland Jiang. 2024. Satisfiability Modulo Theories-Based Qubit Mapping for Trapped-Ion Quantum Computing Systems. In *Proceedings of the 2024 International Symposium on Physical Design (ISPD '24), March 12–15, 2024, Taipei, Taiwan.* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3626184.3633329

1 INTRODUCTION

Quantum computing has emerged as a crucial research field recently due to its great potential to solve complex problems (e.g., factorization [25] and unstructured database search [11]) much faster than classical computers. To implement quantum computing on hardware, various physical architectures with different characteristics and advantages have been proposed to enhance fidelity and increase coherence times in recent years. Superconducting systems offer promising capabilities for realizing large-scale quantum processors with improved coherence and gate fidelity, making them a popular candidate for quantum computing realization. Besides, companies such as IBM, Google, and Intel have made significant hardware advancements in superconducting systems [1, 6, 13]. However, these hardware architectures have limited connectivity among physical qubits, necessitating the costly use of SWAP gates to swap two qubits to overcome the connectivity bottleneck on these hardware architectures.

As a result, another promising architecture known as trapped-ion quantum computers have garnered much attention because trapped ions have relatively long coherence time and can interact with neighbors easily. In trapped-ion systems, the ion qubits in an ion channel are fully coupled, thus significantly reducing the need for SWAP gates for multiple-qubit operations. However, research on qubit mapping techniques to map logical qubits to their physical ones in a trapped-ion system remains an important open task. Developing a new qubit mapping method to perform a quantum algorithm in the trapped-ion architecture is desirable.

Qubit mapping is a crucial step in quantum computing that involves assigning logical qubits to physical ones in a hardware platform. Qubit mapping aims to optimize the mapping strategy to minimize the errors and cost of executing quantum algorithms.

To our best knowledge, no research focused on qubit mapping in single-channel (one-dimensional array) trapped-ion systems, except one work similar to the mapping problem in a two-dimensional array. Durandau *et al.* proposed a heuristic algorithm to generate qubit assignments and shuttling schedules in a shuttling-based trapped-ion system [10]. However, all solutions in this approach

This work was partially supported by AnaGlobe, ASUS, Delta Electronics, Google, Maxeda Technology, TSMC, NSTC of Taiwan under Grant NSTC 110-2221-E-002-177-MY3, NSTC 112-2218-E-002-033, NSTC 112-2223-E-002-020, and NSTC 112-2119-M-002-017.

ISPD '24, March 12–15, 2024, Taipei, Taiwan
© 2024 Copyright held by the owner/author(s). Publication rights licensed to ACM.
ACM ISBN 979-8-4007-0417-8/24/03…$15.00
https://doi.org/10.1145/3626184.3633329

depended on the heuristic algorithm, which could not guarantee the determinism of the solutions. Furthermore, this shuttling-based trapped-ion architecture required rearranging a qubit ordering for the two-qubit gate operations. As a result, using SWAP gates to facilitate the required qubit operations became unavoidable. We will detail the 1-D and 2-D arrays in Section 2.3.

Existing work primarily focused on developing efficient qubit mapping methods for superconducting systems. In superconducting systems, because quantum hardware architectures have limited connectivity among qubits, an effective qubit mapping strategy is essential to ensure proper, especially swap-minimized, execution of quantum circuits. However, finding the optimal qubit mapping solution is intractable due to its NP-complete complexity [26].

To find an optimal qubit-mapping solution, much research applied exact algorithms to satisfy all constraints with minimum costs. Wille *et al.* formulated the mapping task as a symbolic optimization problem that was solved using Boolean satisfiability solvers to map quantum circuits to IBM's QX architectures with a minimum number of SWAP and H operations [29]. Itoko *et al.* employed an exact algorithm to consider the transformation and commutation rules to find better circuit mappings than merely inserting the SWAP gates [14]. However, this exact algorithm was slow, required a large amount of memory, and could not easily scale to more than five qubits.

To tackle large-scale problems, much research applied heuristics to replace exact algorithms. Li *et al.* proposed a SWAP-based search heuristic to reduce the search space [17]. Their proposed algorithm updated an initial mapping solution using reversible quantum characteristics repeatedly because the initial mapping could significantly affect the layout synthesis performance. A weak point of this work is that the number of additional gates (SWAP gates) was not gradually reduced over iterations. Niu *et al.* proposed a hardware-aware mapping transition algorithm with simulated annealing and considered the calibration data to improve the overall fidelity of the circuit [22]. However, the mapping solutions generated by its mapping heuristic heavily depended on the quality of the initial solution, leading to high unpredictability.

To simultaneously maintain solution stability and tackle large-scale problems, some studies combined exact algorithms with a divide-and-conquer approach. Bhattacharjee *et al.* proposed an integer linear programming (ILP) formulation and multi-tier approach to reduce logical depth, minimize gate count, consider the topology, and satisfy gate fidelity constraints [2]. However, this work cannot guarantee the solution quality because of its randomly generated initial mapping. Ding *et al.* first proposed a Satisfiability (SAT) formulation to find the smallest depth mapping solution in a 2-D architecture, then used a Satisfiability Modulo Theories (SMT) formulation to satisfy the nearest neighbor constraints to ensure generating a mapping solution with the smallest number of quantum gates [9]. Besides, they proposed the division of circuits into subcircuits to cope with large-scale circuits because an exhaustive search of the SMT formulation made their algorithm hardly scale. Even with the division approach, the high complexity of the formulation still resulted in significant runtime for small-scale circuits.

Chen *et al.* also proposed an SAT-based algorithm to encode the problem into a linear matrix representation, and then used a SAT solver to check whether a circuit exists using a specified depth [5]. After getting a solution, this work used the ROWCOL-Hybrid algorithm to decode the matrix to a circuit with CNOT gates. At this stage, the circuit does not satisfy the coupling constraint because each subcircuit requires a different coupling graph. To bridge each coupling graph, this work applied the LR-Synth algorithm to add SWAP gates to ensure that the entire circuit satisfied the coupling constraints. Although this work claimed that their algorithm can apply to arbitrary coupling constraint graphs, it might perform poorly when encountering more complex coupling constraint graphs. Furthermore, this approach cannot apply to coupling constraint graphs with a multiple-qubit net in trapped-ion systems.

Based on the SMT formulation, Tan *et al.* proposed two synthesizers: an optimal layout synthesizer for quantum computing (OLSQ) and a transition-based OLSQ (TB-OLSQ) [28]. To accelerate OLSQ, this study proposed TB-OLSQ, which reduced variables and relaxed constraints for better efficiency while sacrificing some solution quality. However, TB-OLSQ was still constrained by its higher complexity, resulting in excessive runtime for larger-scale problems.

To reduce the complexity of the SMT-based mapping algorithm, Wu *et al.* proposed a low-complexity SMT-based qubit mapping checker to determine whether an input quantum algorithm mapping to a hardware architecture had a mapping solution without any SWAP gate [30]. To our best knowledge, their work achieved the best performance and the lowest complexity among the existing SMT-based mapping methods because it simplified the variables and constraints in the SMT formulation by removing the step of SWAP gate insertion. Although this work demonstrated excellent performance on coupling constraint graphs with the two-qubit nets in superconducting systems, their approach cannot be directly extended to the coupling constraint graphs with a multiple-qubit net of the trapped-ion hardware architecture. Besides, the constraints induced in their SMT formulation were insufficient to find the optimal solution on trapped-ion systems because the formulation performed on different coupling constraint graphs with the multiple-qubit nets, as shown in Figure 1.

Figure 1 compares our proposed algorithm with the state-of-the-art superconducting qubit-mapping work [30] using an example with three Mølmer-Sørensen gates. Figures 1(a) and (b) show the algorithm's input data, consisting of a quantum circuit and a coupling constraint graph. Figure 1(a) depicts a quantum circuit with three Mølmer-Sørensen gates. The Mølmer-Sørensen gates are native gates that can be directly operated on trapped-ion systems, to be detailed in Section 2.1. Figure 1(b) represents a coupling constraint graph designed to satisfy the hardware constraints for trapped-ion systems, and we will detail the coupling constraints graph in Section 2.3. Furthermore, Figures 1(c) and (d) represent the mapping solutions of the algorithm from the previous work [30] and our work, respectively. Figure 1(c) shows a legalized solution generated by previous work [30], but it is not an optimal solution due to the design limitations of the SMT formulation.

To remedy the aforementioned limitations, this paper first presents an SMT-based qubit mapping algorithm to find an optimal qubit-mapping solution for small or medium-scale problems on the trapped-ion systems, as shown in Figure 1(d). In this example, our proposed algorithm outperforms the previous work [30], a state-of-the-art superconducting qubit mapping algorithm, with the number of total

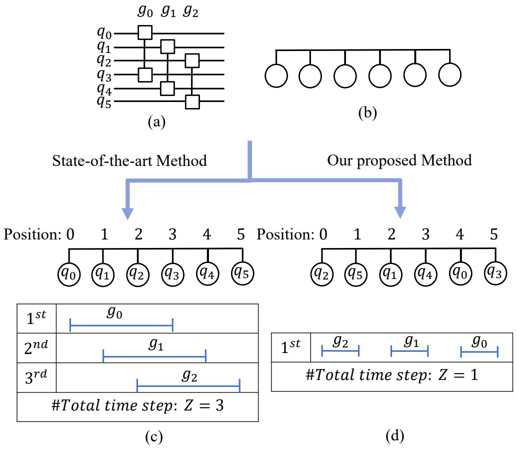


Figure 1: An example of a quantum circuit with three two-qubit gates mapping to the trapped-ion hardware architecture. (a) The quantum circuit with three two-qubit gates. (b) The coupling constraint of the trapped-ion hardware architecture. (c) The mapping solution of previous work [30], without taking into account the characteristics of trapped-ion systems. (d) The mapping solution of our proposed algorithm.

time steps reduction from 3 to 1, due to previous work's improper qubit coupling constraints under the trapped-ion setting. The total time steps represent the cost of executing quantum algorithms on physical architecture in trapped-ion systems, to be detailed in Section 2.4.

To apply our SMT-based qubit mapping algorithm to large-scale problems, we employ a divide-and-conquer approach to alleviate the high complexity of the SMT approach. The divide-and-conquer strategy scales up our algorithm while maintaining the high quality of the solutions provided by the SMT-based qubit mapping algorithm.

We summarize our main contributions as follows:

- We propose a new coupling constraint graph with a multiple-qubit net (multi-pin net) based on the Mølmer-Sørensen gate and its related hardware architecture, which can maintain the fidelity of the quantum circuit and mitigate the occurrence of crosstalk effects.
- We present an SMT-based qubit mapping algorithm to find an optimal qubit-mapping solution on the coupling constraint graphs with a multiple-qubit net for small or medium-scale problems on the trapped-ion systems.
- We prove the optimality of our proposed SMT-based qubit mapping algorithm.
- We present an effective divide-and-conquer method to scale our algorithm and maintain the quality of the SMT solutions for large-scale problems. Based on experimental results, our divide-and-conquer method achieves a 30.56x speedup with only a 3% average loss in solution quality.
- Experimental results show that our algorithm can averagely achieve a 42% total time steps reduction for medium-scale benchmarks and a 45% total time steps reduction for large-scale benchmarks, compared with the state-of-the-art superconducting qubit-mapping work in [30].

The remainder of this paper is organized as follows. Section 2 first gives some preliminaries and then formulates the qubit mapping problem. Section 3 details our proposed algorithm. Section 4 reports the experimental results, and Section 5 concludes this paper.

2 PRELIMINARIES

This section introduces the trapped-ion architecture, the coupling constraint graph, the time step definition, and the SWAP gates. Finally, we formulate the qubit mapping problem.

2.1 Trapped-ion Architecture

Trapped-ion systems are promising for practical quantum computing because trapped ions have relatively long coherence times and can interact with neighbors [4] easily. To implement a quantum circuit, it is crucial to determine how to perform single-qubit and two-qubit gate operations on qubits of trapped-ion systems. However, performing two-qubit gate operations presents significantly greater challenges than single-qubit ones. Therefore, we shall explore existing techniques for operating two-qubit gates on qubits. Currently, there are three mainstream approaches to generating two-qubit gates.

Monroe *et al.* proposed the Cirac-Zoller gate, which first entangled two ions. Furthermore, the CNOT operations could be implemented using a Cirac-Zoller gate and single-qubit rotations, implying that many circuits could be implemented using CZ gates [19]. However, the original Cirac-Zoller proposal had significant limitations because it required the ions to remain in the motional ground state. Additionally, even if the ions had been cooled to the motional ground state, they were still susceptible to heating due to electric-field noise.

To avoid the need for ions to remain in the motional ground state, Leibfried introduced the geometric-phase gate, which used detuned laser beams to generate state-dependent forces [7]. The state-dependent force traced a closed path in phase space to apply specific phase transformations to the quantum states of ions. Although the geometric-phase gate also used the coupling induced by the ion's motion, it was insensitive to the initial ion motional state because the geometric-phase gate was achieved by applying specific phase transformations. However, the geometric-phase gate cannot be applied to FOFI qubits, which is a drawback [16].

A third type of gate could be applied to all kinds of qubits. Mølmer and Sørensen proposed the Mølmer-Sørensen (MS) gate, which was a controlled-phase gate that could be implemented without requiring the ions to be in the motional ground state [27]. Therefore, the MS gate could be applied to ions not cooled to the motional ground state. Additionally, global control lasers could entangle multiple ions and generate MS gates, eliminating the need to focus independently on each ion. However, the internal and motional states had to be disentangled for all values after the gate. To disentangle the internal and motional states, we must consider the crosstalk effect, to be detailed in Section 2.2.

In this paper, we choose the Mølmer-Sørensen gate and its related framework to derive our coupling constraint.

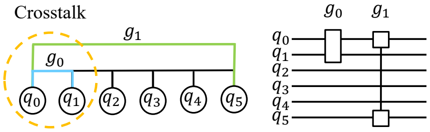


Figure 2: A crosstalk illustration.

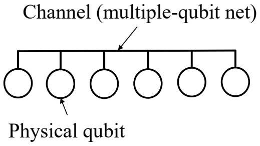


Figure 3: Illustration of the coupling constraint graph.

2.2 Crosstalk

Crosstalk in the trapped-ion system comes from the unpredicted interaction or interference between qubits that partially occurs during gate operations. Besides, crosstalk leads to errors in computations and compromises the reliability and accuracy of quantum algorithms. To avoid the crosstalk effect, we can control the gate timing to minimize the cross-interactions between qubits.

Figure 2 illustrates the crosstalk. If we simultaneously operate gates g_0 and g_1, it will result in crosstalk, and it will be challenging to disentangle the internal and motional states. Therefore, our algorithm ensures that gates are operated at non-overlapping positions in the same time step, improving fidelity and facilitating the disentanglement of internal and motional states after gates.

2.3 Coupling Constraint Graph

In this subsection, we propose a new coupling constraint graph based on the Mølmer-Sørensen gate and its related hardware architecture, which can maintain fidelity and facilitate the internal and motional disentanglement. Figure 3 shows our proposed coupling constraint graph. To ensure non-overlapping operations of gates at the same time step, we employ a multi-pin net (single channel) to connect all physical qubits. To operate a two-qubit gate on two physical qubits, we require a portion of the channel to connect these two physical qubits. When a gate occupies a portion of the channel, other gates at the same time step cannot pass through the already occupied portion. The design of a single channel helps avoid operating multiple gates at overlapping physical positions. Additionally, connecting a single channel to all physical qubits (forming a multiple-qubit net) is an essential feature of a trapped-ion system. Since the physical qubits are interconnected, there is no need to use SWAP gates.

In contrast, in previous superconducting systems, the edges are mainly composed of two-qubit nets, and the number of qubit connections is limited. Therefore, the mapping problem of trapped-ion systems requires new resolution mapping methods.

In this work, we focus on single-channel trapped-ion systems because current ion-based quantum computing demonstrations are conducted using ions trapped in 1D linear arrays. With an increasing number of ions, many advantages of using a 1D array, such as avoiding unnecessary SWAP gates, are preserved. Therefore, building physical trapped-ion systems has largely focused on manufacturing larger 1D arrays. Monz *et al.* presented a qubit quantum processor with 5 quantum bits (qubits) and demonstrated the implementation of several quantum algorithms on this processor [21]. Besides, Yunseong *et al.* proposed the demonstration of a chemical simulation extending 5 up to 11 ions. Monz *et al.* successfully generated entangled states of 14 ions in their experiment [20]. In addition, Zhang *et al.* involved up to 53 ions held in a 1D array to perform quantum simulations [31].

Another research used multiple segments of 1D arrays to form a 2D architecture to replace the expansion of one-dimensional arrays. Therefore, the challenge of such methods lies in how to best move quantum information between 1D arrays. Kaushal *et al.* proposed the architecture of shuttling-based trapped-ion quantum computing involves the storage of multiple trapped-ion qubit sets (2D arrays) in segmented microchip traps [15]. Segmented microchip traps connected to each other by shuttling of ions. In addition, several 2D arrays, such as T-shaped [12] and X-shaped [3] arrays, have been proposed, differing in their connectivity between the 1D arrays. These 2D architectures are extensions of linear architectures, allowing our framework to be potentially extended to different two-dimensional systems with good scalability. As a result, we develop a coupling constraint graph based on the most mature linear architecture.

2.4 Time Step Definition

This section clarifies the definition of time steps employed in this paper. Specifically, to avoid the crosstalk effect, we define that the operations of two gates in the same time step should not overlap (as mentioned in Section 2.3). Besides, a physical qubit can only be operated by one gate in the same time step.

2.5 SWAP Gates

This section explains why SWAP gates are essentially unnecessary in trapped-ion systems, focusing on higher execution costs.

To begin, it is worth noting that a SWAP gate can be decomposed into three CNOT gates. Consequently, the execution of a SWAP gate between two qubits can be viewed as a sequential implementation of three CNOT gates between the two qubits.

To illustrate the impact of SWAP gate insertion, Figure 4 shows the associated cost increases. Initially, Figure 4(c) depicts the mapping solution for the quantum circuit shown in Figure 4(a) before any SWAP gates are included. In this mapping, the gates g_3 and g_4 are assigned to the second- and third-time steps, respectively, according to the definition of time steps.

To allow for concurrent execution of gates g_3 and g_4 in the same time step, we introduce a SWAP gate, as depicted in Figure 4(b), which swaps the positions of the logical qubits q_1 and q_2 corresponding to the physical qubits. Consequently, the resulting mapping solution is illustrated in Figure 4(d). Here, although g_3 and g_4 can be executed simultaneously in the fifth time step, the insertion of the SWAP gate necessitates three additional time steps (specifically, the second, third, and fourth time steps) for its operation. As a result, after inserting a SWAP gate, two more time steps are required. Besides, further decomposition of a CNOT gate involves an MS gate

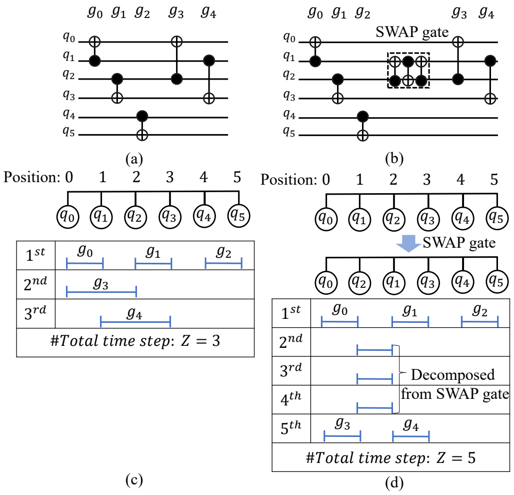

g_0 g_1 g_2 g_3 g_4
q_0 q_1 q_2 q_3 q_4 q_5
(a)
SWAP gate
(b)
Position: 0 1 2 3 4 5
q_0 q_1 q_2 q_3 q_4 q_5
1^{st} 2^{nd} 3^{rd}
#Total time step: $Z = 3$
(c)
SWAP gate
4^{th} 5^{th}
Decomposed from SWAP gate
#Total time step: $Z = 5$
(d)

Figure 4: An example of the cost increases when SWAP gates are added in the trapped-ion systems. (a) The quantum circuit before any SWAP gates are included. (b) The quantum circuit after a SWAP gate is included. (c) The mapping solution before inserting the SWAP gate. (d) The mapping solution after inserting the SWAP gate.

and four single-qubit gates (as discussed in Section 3.5). Therefore, inserting a SWAP gate will increase the total time step by 10 when we convert its three CNOT gates to native gates.

This example demonstrates that incorporating SWAP gates increases the required number of time steps for circuit execution, consequently leading to higher execution costs in nearly all single-track trapped-ion systems. In this paper, we avoid employing any SWAP gates to interchange the positions of logical qubits with respect to physical ones.

2.6 Problem Formulation

The qubit mapping problem can be defined as follows:

Problem 1 (Qubit Mapping Problem). *Given a circuit synthesized from a quantum algorithm and a coupling constraint graph, we minimize the number of total time steps to generate a mapping solution satisfying the coupling constraints.*

3 PROPOSED ALGORITHMS

For a large-scale quantum circuit, qubit mapping optimization is a challenging problem. Our algorithm tries to optimize the total time steps and satisfy the coupling constraints in five major stages: (1) preprocessing, (2) circuit division, (3) optimal mapping for each subcircuit, (4) conquering (subcircuits to the total circuit) and (5) decomposition into native gates.

Figure 5 shows the overall flow of our proposed SMT-based algorithm with a divide-and-conquer approach, where the five stages are detailed in the following subsections.

3.1 Preprocessing

In the preprocessing stage, we note that all quantum circuits can be transformed into the ICM representation [24], which consists of qubit (I)nitialization, (C)NOT gates, and (M)easurements. This paper adopts the technique described in [23] to decompose an arbitrary quantum circuit into the ICM representation. We input the circuit with the ICM representation to the next stage.

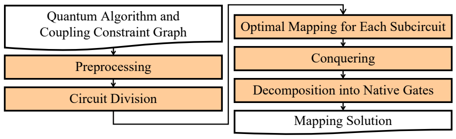

Quantum Algorithm and Coupling Constraint Graph
Preprocessing
Circuit Division
Optimal Mapping for Each Subcircuit
Conquering
Decomposition into Native Gates
Mapping Solution

Figure 5: Overview of our algorithm.

To consider the commutativity of CNOT gates on the control side in the following stages, we decompose the circuit into an ICM representation instead of a native gate circuit in this stage. As shown in Figure 6, the gate arrangement of CNOT gates $g1$ and $g2$ does not affect the circuit's functionality because the input and output values of CNOT gates on the control side are the same. Therefore, a circuit with fixed functionality may have many different gate arrangements. It is essential to consider the gate arrangements of CNOT gates so that the optimal mapping solution can be obtained in the following stages.

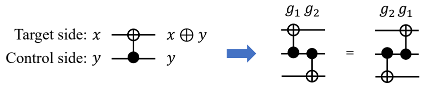


Figure 6: Illustration of positions of control sides does not affect circuit functionality.

3.2 Circuit Division

After decomposing the input quantum circuit into the decomposed one, this stage aims to enhance the scalability of handling large-scale quantum circuits by dividing them into subcircuits. We introduce the parameter n_h to determine the necessity of circuit division. Specifically, if the number of two-qubit gates n_g in the decomposed circuit exceeds n_h, the circuit is divided into $\lceil n_g/n_h \rceil$ subcircuits. Subsequently, the gates are evenly distributed among the subcircuits in a sequential manner. Conversely, if n_g is smaller than n_h, circuit division is unnecessary.

Choosing an appropriate value for n_h involves a trade-off between solution quality and computation time. If n_h is set too small, the excessive subdivision of subcircuits may occur, reducing the solution quality. In contrast, if n_h is set too large, the subcircuits become overly large, leading to prolonged computation time or even making the SMT-based qubit mapping algorithm unable to find a solution. Hence, the selection of n_h necessitates a careful balance between solution quality and computation time considerations.

The impact of varying n_h values on solution quality and computation time is shown in Figure 7 with two benchmarks. Both benchmarks demonstrate that the cost reduction is not effective for $n_h > 60$ in Figure 7(a). Besides, using n_h values much smaller than 60 makes subcircuits too small to maintain solution quality. Furthermore, both benchmarks indicate a significant increase in computation time for n_h values exceeding 60 in Figure 7(b). Therefore, to simultaneously ensure solution quality and reasonable computation time, our algorithm sets the n_h value to a default of 60.

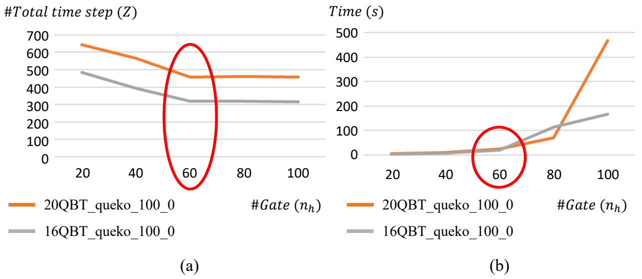
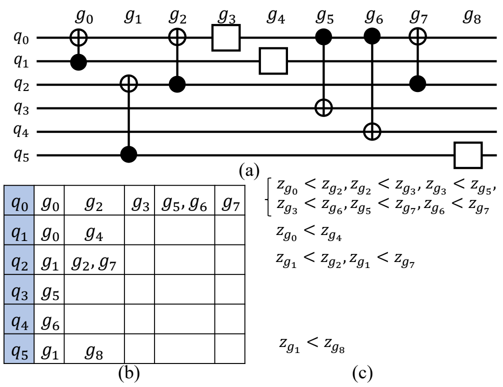


Figure 7: Illustration of the solution quality and computation time with different n_h. (a) The n_h and $\#Total\ time\ steps\ (Z)$ of the line chart. (b) The n_h and $Time\ (s)$ of the line chart.

After dividing the entire circuit into subcircuits, for each of these, we generate a gate dependency table. For a CNOT gate, we focus on the operation location at the target side. Additionally, we must consider the operation location of each single-qubit gate. In Figure 8, we illustrate how to pass gate operation information from a subcircuit to a gate dependency table. As an example in Figure 8(a), the gates g_0, g_2, g_3, g_5, g_6 and g_7 are operated on $q0$. We must execute the g_0, g_2, g_3, and g_7 sequentially on q_0 because the target side of g_0, g_2, and g_7 are operated on q_0 and g_3 is a single-qubit gate. We are not concerned with the operation order between g_5 and g_6 on q_0 because q_0 is operated by the control side of g_5 and g_6. After recording the gate operation information for each qubit, we can construct a dependency table, as shown in Figure 8(b).


g_0 g_1 g_2 g_3 g_4 g_5 g_6 g_7 g_8
q_0
q_1
q_2
q_3
q_4
q_5
(a)
(b)
$z_{g_0} < z_{g_2}, z_{g_2} < z_{g_3}, z_{g_3} < z_{g_5},$
$z_{g_3} < z_{g_6}, z_{g_5} < z_{g_7}, z_{g_6} < z_{g_7}$
$z_{g_0} < z_{g_4}$
$z_{g_1} < z_{g_2}, z_{g_1} < z_{g_7}$
$z_{g_1} < z_{g_8}$
(c)

q_0	g_0	g_2	g_3	g_5, g_6	g_7
q_1	g_0	g_4			
q_2	g_1	g_2, g_7			
q_3	g_5				
q_4	g_6				
q_5	g_1	g_8			

Figure 8: Illustration of elevating gate operation information from a subcircuit to a gate dependency table. (a) A subcircuit. (b) A gate dependency table. (c) The operation order constraints.

3.3 Optimal Mapping for Each Subcircuit

In this stage, we develop an SMT-based qubit mapping algorithm to find an optimal qubit mapping solution for a small/medium-scale circuit or a subcircuit. First, we need to use variables to give the model constraints for ultimately capturing the integral hardware architecture, and then the SMT solver can check whether the model is satisfiable. If the model is satisfiable, the SMT solver will return the variable assignments that are the mapping solution to execute the quantum circuit or subcircuit. If the model is unsatisfiable, we can gradually increase the dimensions of the number of total time steps. Therefore, we first focus on using variables to give the model constraints for capturing the integral hardware architecture. The notations used in the SMT formulation are as follows:

Table 1: Notation table of the SMT formulation.

Notation	Description
Q	a set of logical qubits
G	a set of gates
q_i	the i-th logical qubit in Q
g_i	the i-th gate
n_p	the number of physical qubits
$P(q_i)$	the position of the physical qubit, mapped by q_i
g_i^k	the k-th logical qubit, operated by g_i
z_{g_i}	the time step executing g_i
Z	the total time steps

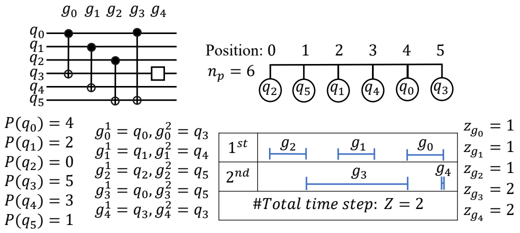


Figure 9: Illustration of the notations.

Figure 9 illustrates the notations corresponding to the quantum circuit and the hardware architecture using an example with four two-qubit gates and a single-qubit gate. To simplify the following SMT constraints, we convert a single-qubit gate to a pseudo-two-qubit gate. In other words, two sides of a pseudo-two-qubit gate operate on the same physical qubit. As shown in Figure 9, the qubit q_3 operated by the single-qubit gate g_4 is thus written in both g_4^1 and g_4^2. Using the above notations and following the coupling constraint, we derive the four SMT constraints as follows:

- One-to-one mapping constraint: This constraint restricts one-to-one mapping between logical and physical qubits. This constraint is as follows:

$$P(q_i) \neq P(q_j), \forall q_i, q_j \in Q, i \neq j, \forall P(q_i) < n_p. \tag{1}$$

- Operation order constraint: This constraint ensures that gates are operated in dependency order, thus maintaining the correctness of the circuit functionality. As shown in Figure 8(c), we extract an operation order constraint from the gate dependency graph (as mentioned in Section 3.2). This constraint is as follows:

$$z_{g_i} < z_{g_j}, \forall g_i \mapsto g_j, \forall g_i, g_j \in D, \tag{2}$$

where D is a gate dependency table, and $g_i \mapsto g_j$ means denotes that g_i has to precede g_j.

- Crosstalk avoidance constraint: This constraint prevents the occurrence of severe crosstalk effects during gate operations. This constraint is as follows:

$$\bigwedge_{k,l\in[1,2]} P(g_i^k) > P(g_j^l) || \bigwedge_{k,l\in[1,2]} P(g_i^k) < P(g_j^l) || z_{g_i} \neq z_{g_j}, \qquad (3)$$
$$\forall g_i, g_j \in G.$$

  This constraint is divided into three major parts using logical OR operations. The first and second parts determine if the operation positions of g_i and g_j overlap on the physical architecture. When the physical positions $(P(g_i^1), P(g_i^2))$ of both qubits (g_i^1, g_i^2) being operated by g_i are simultaneously greater than the physical positions $(P(g_j^1), P(g_j^2))$ of the qubits $((g_j^1, g_j^2))$ being operated by g_j, it implies no overlap (expressed by the first part of the constraint).
  Similarly, if the physical positions $(P(g_i^1), P(g_i^2))$ are simultaneously smaller than the physical positions $(P(g_j^1), P(g_j^2))$, it implies no overlap (expressed by the second part of the constraint).
  Non-overlapping g_i and g_j can be executed in the same time step. However, if the operation positions of g_i and g_j overlap, they must be operated in different time steps (represented by the third part of the constraint).
- Total time step constraint: This constraint determines the total time steps required for the mapping solution. We compute the minimum possible total time step as the starting value for Z. If the model is satisfied, it represents the desired mapping solution; otherwise, we gradually increase Z until a satisfying model is obtained. Therefore, we can obtain the optimal mapping solution for a medium-scale circuit or a subcircuit. This constraint is as follows:

$$\forall z_{g_i} \leq Z, \forall g_i \in G. \qquad (4)$$

3.4 Conquering

After performing the SMT-based qubit mapping algorithm, we must conquer all results to reconstruct the entire circuit for the large-scale circuits. Figure 10 illustrates the conquering method. In this example, the circuit is divided into four subcircuits, and the SMT-based qubit mapping algorithm is applied to each subcircuit to obtain their respective optimal mapping solutions. Subsequently, the subcircuits are conquered with adjacent subcircuits, forming larger conquered subcircuits, and superior solutions are selected from the conquered subcircuits. Specifically, the first subcircuit is conquered with the second subcircuit, resulting in a larger conquered subcircuit. The optimal solution from the second subcircuit exhibits superior performance on the conquered subcircuit compared to the solution from the first subcircuit, thereby becoming the optimal solution for the first conquered circuit.

At the second level, two conquered subcircuits are combined to form the complete circuit. The solution from the first conquered subcircuit (derived from the second subcircuit) achieves better results on the complete circuit than the second one. Consequently, the solution from the second subcircuit serves as the mapping solution for the complete circuit.

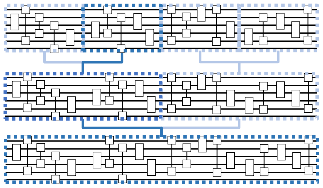


Figure 10: Illustration of the conquering method.

3.5 Decomposition into Native Gates

After we obtain the mapping solution with single-qubit gates and CNOT gates for the complete circuit, we only need to convert the CNOT gates into native gates (MS gates) to enable the execution of the circuit composed of single-qubit gates and MS gates by the trapped-ion hardware architecture. To convert CNOT gates into native gates, we employ the technique described in [18] to decompose a CNOT gate into one MS gate and four single-qubit gates, as shown in Figure 11.

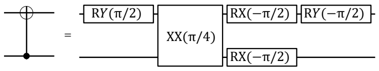


Figure 11: Illustration of decomposing a CNOT gate to native gates.

As an example in Figure 12, we convert four CNOT gates into native gates to get a final solution with a mapping solution from our SMT-based algorithm. Therefore, the total time step (Z) for the final solution in this example is 8.

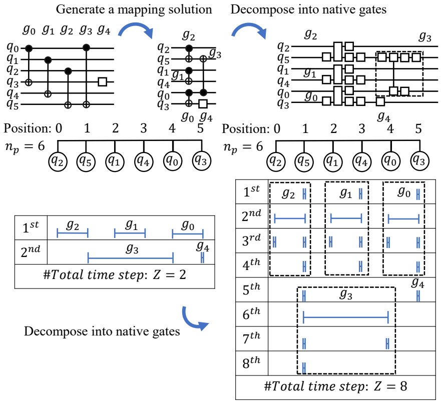

Generate a mapping solution
Decompose into native gates
g_0 g_1 g_2 g_3 g_4
q_0 q_1 q_2 q_3 q_4 q_5
Position: 0 1 2 3 4 5
$n_p = 6$
q_2 q_5 q_1 q_4 q_0 q_3
1^{st} 2^{nd} 3^{rd} 4^{th} 5^{th} 6^{th} 7^{th} 8^{th}
#Total time step: Z = 2
#Total time step: Z = 8
Decompose into native gates

Figure 12: Illustration of converting the CNOT gates into native gates for a mapping solution.

4 EXPERIMENTAL RESULTS

Our proposed algorithm was implemented in the C++ programming language with the Z3 SMT solver 4.12.1 [8]. We performed all experiments on a Linux workstation with 64-core 2.9 GHz AMD Ryzen CPU with 125 GB memory. We used the benchmarks [30] for our experiments, widely used circuits in qubit mapping research.

Table 2: Comparison of the time steps between the previous work [30] and ours for medium-scale benchmarks.

Benchmark	[30]			Ours		
	Z	Ratio	Runtime (s)	Z	Ratio	Runtime (s)
or	24	1.14	0.17	21	1.00	0.07
adder	28	1.08	0.30	26	1.00	0.14
qaoa5	35	2.19	0.37	16	1.00	0.11
Mod5mils_65	53	1.33	0.99	40	1.00	6.17
queko_05_0	32	1.78	0.25	18	1.00	0.20
queko_10_3	68	2.83	1.60	24	1.00	4.09
tof_4	104	1.79	5.01	58	1.00	1.52
Avg. Ratio		1.73			1.00	

Table 3: Comparison of the time steps between the previous work [30] and ours for large-scale benchmarks.

Benchmark	[30]			Ours		
	Z	Ratio	RT (s)	Z	Ratio	RT (s)
queko_15_1	99	2.11	1.50	47	1.00	1.68
barenco_tof_4	157	1.80	3.61	87	1.00	0.55
tof_5	141	2.35	3.15	60	1.00	0.33
barenco_tof_5	221	2.33	9.43	95	1.00	2.20
mod_mult_55	132	1.63	3.36	81	1.00	2.35
vbe_adder_3	143	1.59	4.28	90	1.00	2.08
4gt13_92	88	1.22	1.65	72	1.00	47.97
re_adder_6	184	1.67	6.30	110	1.00	4.65
16QBT_queko_100_0	712	2.24	11.79	318	1.00	18.54
16QBT_queko_100_1	708	2.07	11.90	342	1.00	16.39
16QBT_queko_900_0	6156	2.10	107.35	2926	1.00	166.76
16QBT_queko_900_1	5350	1.81	106.53	2953	1.00	150.46
20QBT_queko_100_0	769	1.68	12.51	457	1.00	27.89
20QBT_queko_100_1	856	1.72	12.88	499	1.00	27.30
20QBT_queko_500_0	3681	1.53	62.83	2403	1.00	225.30
20QBT_queko_500_1	3883	1.63	62.50	2377	1.00	259.04
54QBT_queko_05_0	99	1.98	1.03	50	1.00	9.81
54QBT_queko_05_1	116	1.93	1.02	60	1.00	9.19
54QBT_queko_900_0	18195	1.47	227.92	12375	1.00	1966.96
54QBT_queko_900_1	18035	1.56	227.01	11582	1.00	1931.58
Avg. Ratio		1.82			1.00	

Table 4: Comparison of the time steps and runtime between w/o and w/ divide-and-conquer approach.

Benchmark	w/o				w/			
	Z	Ratio	RT (s)	Ratio	Z	Ratio	RT (s)	Ratio
queko_15_1	45	0.96	48.95	29.14	47	1.00	1.68	1.00
barenco_tof_4	87	1.00	13.24	23.99	87	1.00	0.55	1.00
tof_5	58	0.97	3.87	11.72	60	1.00	0.33	1.00
barenco_tof_5	95	1.00	165.17	75.18	95	1.00	2.20	1.00
mod_mult_55	74	0.91	87.69	37.28	81	1.00	2.35	1.00
vbe_adder_3	90	1.00	12.61	6.08	90	1.00	2.08	1.00
Avg. Ratio		0.97		30.56		1.00		1.00

We briefly explain how we must re-implement the algorithms in the previous work to operate on the proposed coupling constraint graph for trapped-ion systems. To execute the original algorithm from the previous work, we must provide a quantum circuit and a coupling constraint graph as inputs. However, we cannot provide the coupling constraint graph because the coupling constraint graphs differ between superconducting and trapped-ion systems. In superconducting systems, the edges in the coupling constraint graph represent two-qubit nets (two-pin nets), which vary from the multiple-qubit nets (multi-pin nets) in our framework (trapped-ion systems). Because the coupling constraint graph on trapped-ion systems cannot be directly used as an input for the program from the previous work, we re-implemented the algorithm presented in the previous work to operate on our proposed coupling constraint graph.

To facilitate fair comparison and accurate analysis of the experimental results, we categorize the benchmarks into medium-scale and large-scale benchmarks. The time-step comparisons of the previous work [30] and our proposed algorithm are summarized in Tables 2 (medium-scale benchmarks) and 3 (large-scale benchmarks). Note that "Z" denotes the number of the total time steps, which has been detailed in Section 2.4.

In the medium-scale benchmarks, our results are the optimal solutions under the problem formulation of this paper because no divide-and-conquer approach to divide the circuits is involved. Applying our SMT-based qubit mapping algorithm guarantees the optimal mapping solution and achieves an average 42% improvement over the previous work [30].

For large-scale benchmarks, because we employ the divide-and-conquer approach to divide the problem into subproblems, our solutions are not always optimal. Although we sacrifice a portion of the solution quality due to the divide-and-conquer approach, our algorithm can achieve an average significant improvement of 45% over the previous work [30], as shown in Table 3.

We select six large-scale benchmarks to show the efficiency of our proposed divide-and-conquer method, as shown in Table 4. For the other fourteen large-scale benchmarks, which are not shown in the Table 4, the results can only be generated by including the divide-and-conquer method (runtime limit of 3600 seconds). As a result, our divide-and-conquer method achieves a 30.56x speedup with only a 3% average loss in solution quality. For further analysis, we use the Penalized Average Runtime PAR-2 score to compare twenty large-scale benchmarks, where the fourteen unsolved cases are given a penalty of 2 times the time-out limit. Our divide-and-conquer method achieves a 245.41x speedup on twenty benchmarks. Therefore, the experimental results based on the medium-scale and large-scale benchmarks well justify the effectiveness of our mapping algorithm.

5 CONCLUSION

We have proposed a novel coupling constraint graph with a multiple-qubit net based on the Mølmer-Sørensen gate and its associated hardware architecture. This graph has been designed to effectively maintain the fidelity of the quantum circuit and alleviate the crosstalk effects. To address the mapping challenges encountered in coupling constraint graphs with a multiple-qubit net in a trapped-ion system, we have proposed an SMT-based qubit mapping algorithm to find an optimal qubit-mapping solution for the small and medium-scale problems. We have proven that our proposed SMT-based qubit mapping algorithm can guarantees the solution optimality. For the larger-scale problems, we have employed the divide-and-conquer method to scale our algorithm effectively. This approach allows us to tackle large-scale and complex problems while preserving the quality of solutions from the SMT-based qubit mapping algorithm. Through extensive experimental evaluations, we have validated the effectiveness of our algorithm. The results have shown that, on average, our algorithm achieves respective 42% and 45% reductions in total time steps for medium-scale and large-scale benchmarks.

REFERENCES

[1] Frank Arute, Kunal Arya, Ryan Babbush, et al. 2020. Hartree-Fock on a superconducting qubit quantum computer. *arXiv preprint arXiv:2004.04174* (2020).

[2] Debjyoti Bhattacharjee, Abdullah Ash Saki, Mahabubul Alam, Anupam Chattopadhyay, and Swaroop Ghosh. 2019. MUQUT: Multi-Constraint Quantum Circuit Mapping on NISQ Computers: Invited Paper. In *Proc. of ICCAD*. Westminster, Colorado, USA, 1–7.

[3] R. B. Blakestad, C. Ospelkaus, A. P. VanDevender, J. M. Amini, J. Britton, D. Leibfried, and D. J. Wineland. 2009. High-Fidelity Transport of Trapped-Ion Qubits through an **X**-Junction Trap Array. *Phys. Rev. Lett.* 102, 15 (2009), 153002–153005.

[4] Colin D. Bruzewicz, John Chiaverini, Robert McConnell, and Jeremy M. Sage. 2019. Trapped-ion quantum computing: Progress and challenges. *arXiv preprint arXiv:1904.04178* (2019).

[5] Cynthia Chen, Bruno Schmitt, Helena Zhang, Lev S. Bishop, and Ali Javadi-Abhar. 2022. Optimizing Quantum Circuit Synthesis for Permutations Using Recursion. In *Proc. of DAC*. San Francisco, California, 7–12.

[6] Jerry Chow, Oliver Dial, and Jay Gambetta. 2021. IBM Quantum breaks the 100-qubit processor barrier. *IBM Research Blog* (2021).

[7] Sean D Conner and Sandra L Schmid. 2003. Regulated portals of entry into the cell. *Nature* 422, 6927 (2003), 37–44.

[8] Leonardo De Moura and Nikolaj Bjørner. 2008. Z3: An efficient SMT solver. In *Proc. of TACAS*. Budapest, Hungary, 337–340.

[9] Jingwen Ding and Shigeru Yamashita. 2020. Exact Synthesis of Nearest Neighbor Compliant Quantum Circuits in 2-D Architecture and Its Application to Large-Scale Circuits. *IEEE Tran. on CAD* 39, 5 (2020), 1045–1058.

[10] Jonathan Durandau, Janis Wagner, Frédéric Mailhot, Charles-Antoine Brunet, Ferdinand Schmidt-Kaler, Ulrich Poschinger, and Yves Bérubé-Lauzière. 2022. Automated Generation of Shuttling Sequences for a Linear Segmented Ion Trap Quantum Computer. *arXiv preprint arXiv:2208.04881* (2022).

[11] Lov K Grover. 1996. A fast quantum mechanical algorithm for database search. In *Proc. of STOC*. Philadelphia, Pennsylvania, USA, 212–219.

[12] WK Hensinger, S Olmschenk, D Stick, D Hucul, M Yeo, Mark Acton, L Deslauriers, C Monroe, and J Rabchuk. 2006. T-junction ion trap array for two-dimensional ion shuttling, storage, and manipulation. *Appl. Phys. Lett.* 88, 3 (2006), 034101.

[13] Jeremy Hsu. 2018. Ces 2018: Intel's 49-qubit chip shoots for quantum supremacy. *IEEE Spectrum Tech Talk* (2018), 1–6.

[14] Toshinari Itoko, Rudy Raymond, Takashi Imamichi, and Atsushi Matsuo. 2020. Optimization of quantum circuit mapping using gate transformation and commutation. *Integration* 70, 1 (2020), 43–50.

[15] V Kaushal, Bjoern Lekitsch, A Stahl, J Hilder, Daniel Pijn, C Schmiegelow, Alejandro Bermudez, M Müller, Ferdinand Schmidt-Kaler, and U Poschinger. 2019. Shuttling-Based Trapped-Ion Quantum Information Processing. *arXiv preprint arXiv:1912.04712* (2019).

[16] Christopher E Langer. 2006. *High fidelity quantum information processing with trapped ions.* Ph. D. Dissertation. University of Colorado at Boulder.

[17] Gushu Li, Yufei Ding, and Yuan Xie. 2019. Tackling the qubit mapping problem for NISQ-era quantum devices. In *Proc. of ASPLOS*. Providence, Rhode Island, 1001–1014.

[18] Dmitri Maslov. 2016. Basic circuit compilation techniques for an ion-trap quantum machine. *arXiv preprint arXiv:1603.07678* (2016).

[19] C. Monroe, D. M. Meekhof, B. E. King, W. M. Itano, and D. J. Wineland. 1995. Demonstration of a Fundamental Quantum Logic Gate. *Phys. Rev. Lett.* 75, 25 (1995), 4714–4717.

[20] Thomas Monz, Philipp Schindler, Julio T. Barreiro, Michael Chwalla, Daniel Nigg, William A. Coish, Maximilian Harlander, Wolfgang Hänsel, Markus Hennrich, and Rainer Blatt. 2011. 14-Qubit Entanglement: Creation and Coherence. *Phys. Rev. Lett.* 106, 13 (2011), 130506–130510.

[21] Yunseong Nam, Jwo-Sy Chen, Neal C. Pisenti, et al. 2019. Ground-state energy estimation of the water molecule on a trapped ion quantum computer. *arXiv preprint arXiv:1902.10171* (2019).

[22] Siyuan Niu, Adrien Suau, Gabriel Staffelbach, and Aida Todri-Sanial. 2020. A Hardware-Aware Heuristic for the Qubit Mapping Problem in the NISQ Era. *arXiv preprint arXiv:2010.03397* (2020).

[23] Alexandru Paler, Simon J Devitt, and Austin G Fowler. 2016. Synthesis of arbitrary quantum circuits to topological assembly. *Sci. Rep.* 6, 1 (2016), 1–16.

[24] Alexandru Paler, Ilia Polian, Kae Nemoto, and Simon J Devitt. 2015. A fully fault-tolerant representation of quantum circuits. In *Proc. of RC*. Grenoble, France, 139–154.

[25] Peter W Shor. 1999. Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. *SIAM Rev.* 41, 2 (1999), 303–332.

[26] Marcos Yukio Siraichi, Vinícius Fernandes dos Santos, Caroline Collange, and Fernando Magno Quintao Pereira. 2018. Qubit allocation. In *Proc. of CGO*. Vienna, Austria, 113–125.

[27] Anders Sørensen and Klaus Mølmer. 1999. Quantum Computation with Ions in Thermal Motion. *Phys. Rev. Lett.* 82, 9 (1999), 1971–1974.

[28] Bochen Tan and Jason Cong. 2020. Optimal layout synthesis for quantum computing. In *Proc. of ICCAD*. Virtual Event, 1–9.

[29] Robert Wille, Lukas Burgholzer, and Alwin Zulehner. 2019. Mapping quantum circuits to IBM QX architectures using the minimal number of SWAP and H operations. In *Proc. of DAC*. Las Vegas, Nevada, USA, 1–6.

[30] Tsou-An Wu, Yun-Jhe Jiang, and Shao-Yun Fang. 2022. A robust quantum layout synthesis algorithm with a qubit mapping checker. In *Proc. of ICCAD*. San Diego, California, USA, 1–9.

[31] Jiehang Zhang, Guido Pagano, Paul W Hess, Antonis Kyprianidis, Patrick Becker, Harvey Kaplan, Alexey V Gorshkov, Z-X Gong, and Christopher Monroe. 2017. Observation of a many-body dynamical phase transition with a 53-qubit quantum simulator. *Nature* 551, 7682 (2017), 601–604.