Exact Mapping of Quantum Circuit Partitions to Building Blocks of the SAQIP Architecture

Amirmohammad Biuki*, Naser Mohammadzadeh*, Robert Wille†‡, and Sahar Sargaran*
*Department of Computer Engineering, Shahed University, Tehran, Iran
†Technical University of Munich, Munich, Germany
‡Software Competence Center Hagenberg GmbH (SCCH), Austria
a.biuki@shahed.ac.ir, mohammadzadeh@shahed.ac.ir, robert.wille@tum.de, s.sargeran@shahed.ac.ir

***Abstract*—Quantum computing benefits from collective features of quantum states, such as superposition and entanglement, to efficiently address problems that are very hard to be solved on classical systems. The *Scalable Architecture for Quantum Information Processor* (SAQIP) architecture is a promising technology that is based on ion-traps and realizes a hybrid composed of a large number of full-custom building blocks (supposed to realize so-called *Elementary Logic Units*; ELUs) connected by a reconfigurable optical switch network. As with every architecture, corresponding design methods are required in order to properly map a given quantum functionality onto the respective device. However, since the corresponding complexity frequently made exact solutions for this task infeasible for past architectures, most of the existing mapping methods rely on heuristics and, hence, do not provide exact/optimal results. Considering the SAQIP architecture, however, this problem can be prevented. In fact, due to the building blocks of this architecture, any circuit to be mapped has to be partitioned into ELUs anyway. Since those are usually of moderate size, exact/optimal solutions for them are possible. In this work, we sketch an exact mapping method that can generate such optimal results. To this end, we propose a corresponding formulation in mixed-integer linear programming (MILP) that allows to cope with the (smaller, but still not non-trivial) complexity.**

I. Introduction

All of DiVincenzo's requirements for a universal quantum computer and quantum algorithms have been demonstrated with the ion-trap technology [1]: Monroe et al. [2] demonstrated a hybrid distributed architecture for ion-trap quantum processors to conquer the serious scalability issues of *Quantum Charged-Coupled Device* architecture [3]. It is composed of *Elementary Logic Units* (ELUs) including a string of 50–100 ion-qubits connected through an photonic network. Ahsan et al. [4] improved this architecture by replacing the string structure of ELUs with a *Quantum Logic Array* (QLA, [5]). Recently, [6] proposed the *Scalable Architecture for Quantum Information Processors* (SAQIP) architecture to address the issues of Ahsan's architecture. They used the same hybrid system but utilized ion-trap technology capabilities and designed ELUs in the full-custom style to alleviate the issues. Its performance evaluation showed that, thus far, it is the best and most promising architecture.

However, all those architectures can only succeed for practically-relevant applications, if corresponding design methods exist that map a given quantum functionality onto the respective device. Accordingly, several methods have been proposed for this purpose (see, e.g., [4–8]). But, due to the complexity of the problem (which is NP-hard [5]), most of these methods are heuristic in nature, i.e., they cannot provide exact/optimal results. Exceptions are, e.g., the work proposed in [8, 9] which, in turn, are not very scalable and, hence, only applicable for rather small quantum functionality.

Because of this, it remained hard to obtain optimal results for ion-trap architectures. Considering SAQIP, this problem could be prevented. In fact, having the full-custom structure, accommodating more than two qubits in each gate location, and adding a three-qubit Toffoli gate to the gate library cause that SAQIP requires substantially smaller ELUs compared to other architectures [6] – providing the potential to actually address the design task in an exact/optimal fashion at least for ELUs.

In this work, we are aiming to use this potential. We propose a method that maps quantum circuit partitions to SAQIP while, at the same time, keeps the costs, i.e., the number of required time steps, minimal for each ELU. In order to cope with the resulting complexity, we are proposing a corresponding formulation in mixed-integer linear programming (MILP) and use corresponding solvers for the following reasoning. This allows to *efficiently* generate mappings for the considered ELUs, while guaranteeing minimality. The resulting (minimal) ELUs can then be used by the existing SAQIP design flow to realize the entire functionality.

II. Background and Considered Problem

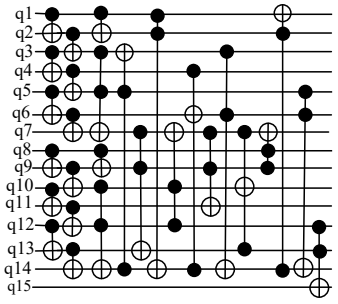
A. Quantum Circuits

The quantum circuit model is the most widely-used and developed model for quantum computation. In a quantum circuit, times goes from left to right and each line represents the evolution of each qubit through time.

Example 1. *Fig. 1 provides an example of a quantum circuit composed of fifteen qubits q1 to q15, fourteen two-qubit gates, and fourteen three-qubit gates.*

B. SAQIP Architecture

The SAQIP architecture [6] (as sketched in Fig. 2) relaxed some assumptions based on the recent advancements in ion-trap technology [1]. It is composed of a large number of full-custom



G1: CX q1, q2
G2: CX q3, q4
G3: CX q5, q6
G4: CX q8, q9
G5: CX q10, q11
G6: CX q12, q13
G7: CX q2, q3
G8: CX q4, q5
G9: CX q6, q7
G10: CX q9, q10
G11: CX q11, q12
G12: CX q13, q14
G13: CX q1, q2
G14: Toffoli q3, q5, q7
G15: CX q8, q9
G16: Toffoli q10, q12, q14
G17: Toffoli q14, q5, q3
G18: Toffoli q7, q9, q13
G19: Toffoli q1, q2, q14
G20: Toffoli q12, q10, q7
G21: Toffoli q14, q4, q6
G22: Toffoli q7, q9, q11
G23: Toffoli q3, q6, q14
G24: Toffoli q7, q12, q10
G25: Toffoli q9, q8, q7
G26: Toffoli q14, q2, q1
G27: Toffoli q5, q6, q14
G28: Toffoli q12, q13, q15

Fig. 1: A sample circuit and its netlist including fifteen qubits, fourteen two-qubit gates, and fourteen three-qubit gates.

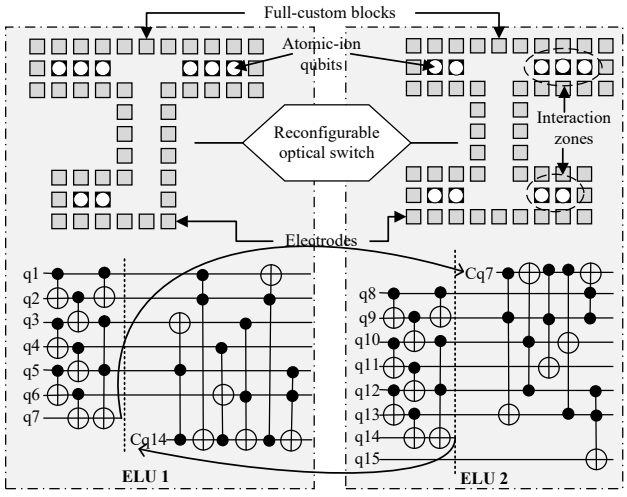


Fig. 2: Overview of the SAQIP architecture [6]. Cq7 and Cq14 are the commination qubits taking the states of qubits q7 and q14 respectively.

blocks connected by a reconfigurable optical switch network. Its blocks are designed in the full-custom style based on the multiplexed ion-trap structure [10]. Each block is composed of some interaction zones and each zone confines a few ions. Quantum data is transferred between zones of a block by ballistically shuttling ions and between blocks by photons via the optical switch network.

Example 2. *Fig. 2 shows a sample SAQIP architecture generated for the circuit depicted in Fig. 1 including two ELUs, ELU 1 and ELU 2. As can be seen, the state of q7 located firstly in ELU 1 is transferred to a communication qubit, namely Cq7, in ELU 2. Correspondingly, the state of q14 located firstly in ELU 2 is transferred to a communication qbit, namely Cq14 in ELU 1. The arrows between the sub-circuits show the qubit state transfers between ELUs.*

C. Considered Design Problem

Fig. 3 shows the *Computer-Aided Design* flow proposed for SAQIP as well as the detailed steps for mapping one partition onto physical layout. In the first step, the k-way hMETIS partitioning algorithm [11] breaks up qubits among blocks such that the number of communications between blocks is minimized. To map a resulting quantum circuit partition onto an ELU, three separate main processes are performed, i.e.,

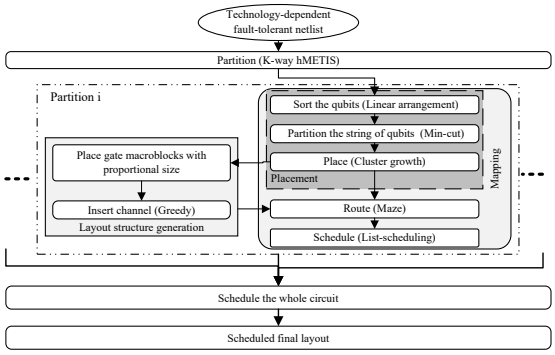


Fig. 3: The SAQIP architecture design flow [6].

placement, routing, and scheduling. To place one partition, first the linear arrangement method sorts the qubits of each ELU as a single sorted string. This string subsequently is partitioned by the min-cut algorithm into substrings. Then, the cluster growth placement approach determines the initial position of qubits (substrings). Meanwhile, a modified version of the algorithm proposed in [12] generates the layout structure with placing gate macroblocks with the proportional size to substrings size and inserting channels to connect the gate locations. After the layout is generated, the Maze routing method determines the movement paths of qubits. Afterwards, the list scheduling method step determines the instruction execution sequence as well as the order of qubit movements across channels. Finally, a simple greedy scheduling method schedules the whole circuit and generates the final layout.

Within this flow, mapping partitions onto the ELUs apparently is a key step that is the main focus of this paper.

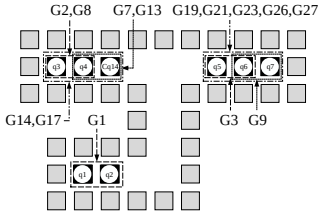
Example 3. *Fig. 4 sketches the optimal mapping of the partition of the considered quantum circuit assigned to ELU 1 depicted in Fig. 2 onto the generated layout. Fig. 4a sketches the list assigned to each gate location that determines the initial location of each qubit and the gates that are to be performed in that gate location. Fig. 4b sketches the schedule for the gates running on the layout. This circuit needs 5657 time steps to be run on the layout. Fig. 4c sketches the schedule for the qubits.*

III. Proposed Solution

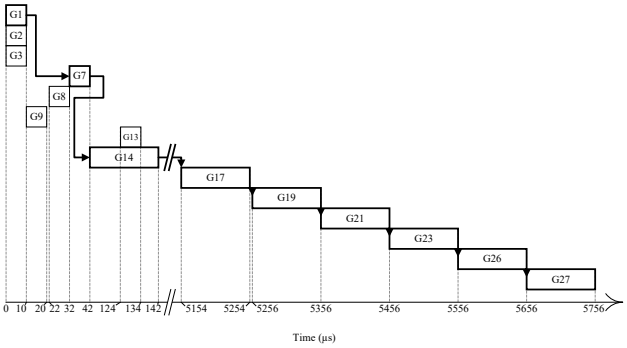
A. General Idea

A substantial amount of work has been tried to develop methods that map a quantum circuit onto corresponding ion-trap architectures. For SAQIP, similar solutions than those are needed, too, which might tempt researchers and engineers to simply reuse them. However, most of the existing approaches either have been developed for grid ion-trap layouts confining only two qubits in each gate location or have been developed for linear-trap ion-trap architectures that are not scalable. Moreover, most of the existing solutions are not exact, thus cannot guarantee minimality. Some few of them can guarantee minimal solutions, but then, they are hardly scalable.

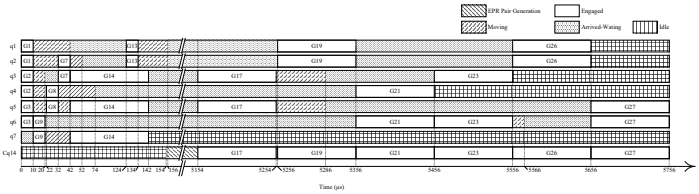
However, considering the SAQIP architecture, an alternative approach becomes possible that allows for a solution that can guarantee minimality (at least for the quantum circuit partitions to be mapped to ELUs) while still remains applicable. This is,



(a) Qubit initial placement and gate placement. The list assigned to each gate location determines the initial location of each qubit and the gates that are to be performed in that.



(b) Gate scheduling.



(c) Qubit scheduling.

Fig. 4: Optimal mapping of the circuit assigned to ELU 1 onto the layout generated by the SAQIPSim toolset that is obtained by our approach.

because the SAQIP architecture provides building blocks (i.e., entities for ELUs) that will not exceed a certain size. Because of this, scalability might be less an issue.

Nevertheless, determining exact/optimal results still requires an efficient solution. We propose to address this by providing a MILP formulation of the problem.

B. Sketch of the Resulting Formulation

In this section, the proposed formulation is described, which precisely takes into account the SAQIP architecture features introduced in Section II-B. The MILP model is presented for the quantum mapping problem which takes two inputs: the partition of a quantum circuit assigned to an ELU in the partitioning step and a corresponding layout to be mapped to. Then, the formulation is supposed to describe the problem of mapping the circuit partition onto the layout. The formulation is defined to determine the best scheduling, initial qubit placement and gate placement, as well as qubit routing for each partition in terms of number of time steps.

The proposed model is summarized as follows: The critical path delay l is considered as the main objective in Formula (1):

$$\text{Minimize} \quad l \tag{1}$$

Formula (2) defines the critical path as the maximum completion time of gates and Formula (3) calculates the end time of gates, i.e.,

$$l \geq e_g \qquad \forall g \in G, \tag{2}$$

$$s_g + o_g = e_g \qquad \forall g \in G, \tag{3}$$

where s_g and e_g define the start time variable and the end time variable of performing gate g, respectively and o_g is the delay parameter of performing gate g. G is the set of gates.

Formula (4) is defined to keep the order of execution of dependent gates, i.e.,

$$e_{g'} + d_{tt'} + (p_{g'}^{qt'} + p_g^{qt} - 2)M \leq s_g \qquad \forall (g, g', q) \in F, \; (t, t') \in P, \tag{4}$$

where $d_{tt'}$ is the time that a qubit needs to traverse from trap t to t' (pre-computed by the Dijkstra algorithm) and p_g^{qt} is a variable that is set to 1 if qubit q from gate g is placed in trap t, otherwise, it is set to 0. Set F includes qubit dependencies between gates. The set P is the set of all possible pairs of traps in the layout. Finally, M is a very big number.

To handle gate placement, Formulas (5)–(9) are defined to map the qubits of gates to traps according to the clusters that the gates belong to. A cluster is defined as a group of adjacent traps in an interaction zone that may consist of one trap up to three traps. The set C indicates the set of clusters. More precisely: The mapping of the qubit of each one-qubit gate to a trap is constrained in Formula (5), i.e.,

$$p_g^c \leq p_g^{qt} \qquad \forall g \in G_1, (g, q) \in R, (c, t) \in S_1, \tag{5}$$

where p_g^c is a variable that is set to 1 if gate g is assigned to cluster $c \in C$, and otherwise is set to 0. G_1 is the set of one-qubit gates ($G_1 \subset G$). R is the set of (g, q) that gate $g \in G$ needs qubit q for execution. S_1 is the set of (c, t) that the cluster $c \in C_1$ contains trap t, where C_1 is the set of clusters containing only one trap ($C_1 \subset C$).

For each two-qubit and three-qubit gates, all the involving qubits in that gate should be placed in adjacent traps included in the cluster. In order to place and execute a two-qubit or three-qubit gates, the following constraints should be satisfied:

1) Each qubit must be assigned to only one trap of the cluster assigned to a gate.
2) Each trap of a cluster assigned to a gate must belong to only one qubit.

These conditions are checked for two-qubit gate in Formulas (6) and (7), and for three-qubit gates in Formula (8) and (9), i.e.,

$$(1 - p_g^c)m \leq 1 - (p_g^{qt} + p_g^{qt'}) \leq (1 - p_g^c)M \quad \forall g \in G_2, (g, q) \in R, (c, t, t') \in S_2 \tag{6}$$

$$(1 - p_g^c)m \leq 1 - (p_g^{qt} + p_g^{q't}) \leq (1 - p_g^c)M \quad \forall (g, q, q') \in R_2, c \in C_2, (c, t) \in S \tag{7}$$

$$(1 - p_g^c)m \leq 1 - (p_g^{qt} + p_g^{qt'} + p_g^{qt''}) \leq (1 - p_g^c)M \quad \forall g \in G_3, (g, q) \in R, (c, t, t', t'') \in S_3 \tag{8}$$

$$(1 - p_g^c)m \leq 1 - (p_g^{qt} + p_g^{q't} + p_g^{q''t}) \leq (1 - p_g^c)M \quad \forall (g, q, q', q'') \in R_3, c \in C_3, (c, t) \in S \tag{9}$$

where G_2 and G_3 are the sets of two- and three-qubit gates, respectively. S_2 is the set of (c, t, t'), where cluster $c \in C_2$ contains adjacent traps t and t'. C_2 is the set of clusters that contain two adjacent traps ($C_2 \subset C$). S_3 is the set of (c, t, t', t''), where cluster $c \in C_3$ contains adjacent traps t, t' and t''. C_3 is the set of clusters that contain three adjacent traps ($C_3 \subset C$). R_2 is the set of (g, q, q'), where gate $g \in G_2$ needs qubits q and q' for execution. R_3 is the set of (g, q, q', q''), where gate $g \in G_3$ needs qubits q, q', and q'' for execution. M and m are a sufficiently large number and a sufficiently small number respectively. S is the set of (c, t) that cluster $c \in C$ contains trap t.

The initial qubit placement in each ELU is enforced in Formulas (10) and (11), i.e.,

$$\sum_{(q,g)\in D} p_g^{qt} = 1 \qquad \forall t \in T, \tag{10}$$

$$\sum_{t\in T} p_g^{qt} = 1 \qquad \forall (q, g) \in D, \tag{11}$$

where D is the set of (q, g) that qubit q debuts in gate g in the netlist and T is the set of traps.

Rules for assigning gates to clusters are satisfied in Formulas (12) to (17). Since, in any type of gates, the number of gates could normally exceed the number of relevant clusters available due to the SAQIP architecture, each cluster might hold several gates. Although we did not consider congestion in the model, we try to fairly distribute gates over the layout. Therefore, we apply lower bounds in Formulas (13), (15), and (17) which are related to one-, two- and three-qubit gates, respectively. These bounds are achieved by distributing gates over possible clusters aiming to give equal shares to clusters. More precisely:

$$\sum_{c\in C_1} p_g^c = 1 \qquad \forall g \in G_1 \tag{12}$$

$$\left\lfloor \frac{|G_1|}{|C_1|} \right\rfloor \leq \sum_{g\in G_1} p_g^c \qquad \forall c \in C_1 \tag{13}$$

$$\sum_{c\in C_2} p_g^c = 1 \qquad \forall g \in G_2 \tag{14}$$

$$\left\lfloor \frac{|G_2|}{|C_2|} \right\rfloor \leq \sum_{g\in G_2} p_g^c \qquad \forall c \in C_2 \tag{15}$$

$$\sum_{c\in C_3} p_g^c = 1 \qquad \forall g \in G_3 \tag{16}$$

$$\left\lfloor \frac{|G_3|}{|C_3|} \right\rfloor \leq \sum_{g\in G_3} p_g^c \qquad \forall c \in C_3 \tag{17}$$

Finally, Formulas (18) to (21) restrict variables to valid values.

$$p_g^{qt} \in \{0, 1\} \qquad \forall g \in G, q \in Q, t \in T \tag{18}$$

$$p_g^c \in \{0, 1\} \qquad \forall g \in G, c \in C \tag{19}$$

$$s_g \geq 0 \qquad \forall g \in G \tag{20}$$

$$e_g \geq 0 \qquad \forall g \in G \tag{21}$$

Passing the formulation from above to a reasoning engine, assignments to all used variables are obtained out of which an optimal mapping of a quantum circuit partition for an ELU can be derived. By this, optimal results for ELUs are obtained which, afterwards, can be combined to realize a mapping for the entire circuit (note that the combination is not necessarily minimal anymore, but still can improve a heuristic solution).

IV. Conclusions and Future Work

In this work, we proposed an exact approach for mapping partitions of a quantum circuit onto building blocks of the SAQIP architecture. This became possible, since the small size of ELUs in the SAQIP architecture allowed to generate optimal results for its ELUs without scalability issues. To address the (still non-trivial) complexity, a MILP formulation has been proposed which can efficiently been solved using a reasoning engine. In future work, detailed evaluations will be conducted to evaluate the performance of the proposed approach.

Acknowledgments

This work received funding from the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme (grant agreement No. 101001318), was part of the Munich Quantum Valley, which is supported by the Bavarian state government with funds from the Hightech Agenda Bayern Plus, and has been supported by the BMK, BMDW, and the State of Upper Austria in the frame of the COMET program (managed by the FFG).

References

[1] 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.

[2] C. Monroe, R. Raussendorf, A. Ruthven, K. Brown, P. Maunz, L.-M. Duan, and J. Kim, "Large-scale modular quantum-computer architecture with atomic memory and photonic interconnects," *Physical Review A*, vol. 89, no. 2, p. 022317, 2014.

[3] 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.

[4] M. Ahsan, R. V. Meter, and J. Kim, "Designing a million-qubit quantum computer using a resource performance simulator," *ACM Journal on Emerging Technologies in Computing Systems (JETC)*, vol. 12, no. 4, pp. 1–25, 2015.

[5] N. Mohammadzadeh, "Physical design of quantum circuits in ion trap technology–a survey," *Microelectronics journal*, vol. 55, pp. 116–133, 2016.

[6] S. Sargaran and N. Mohammadzadeh, "Saqip: A scalable architecture for quantum information processors," *ACM Transactions on Architecture and Code Optimization (TACO)*, vol. 16, no. 2, pp. 1–21, 2019.

[7] M. Webber, S. Herbert, S. Weidt, and W. K. Hensinger, "Efficient qubit routing for a globally connected trapped ion quantum computer," *Advanced Quantum Technologies*, vol. 3, no. 8, p. 2000027, 2020.

[8] X.-C. Wu, Y. Ding, Y. Shi, Y. Alexeev, H. Finkel, K. Kim, and F. T. Chong, "Ilp-based scheduling for linear-tape model trapped-ion quantum computers," in *the Proceedings of The International Conference for High Performance Computing, Networking, Storage, and Analysis, Denver Co*, 2019.

[9] O. Keszocze, N. Mohammadzadeh, and R. Wille, "Exact physical design of quantum circuits for ion-trap-based quantum architectures," in *2021 Design, Automation & Test in Europe Conference & Exhibition (DATE)*. IEEE, 2021, pp. 344–349.

[10] N. Akerman, N. Navon, S. Kotler, Y. Glickman, and R. Ozeri, "Universal gate-set for trapped-ion qubits using a narrow linewidth diode laser," *New Journal of Physics*, vol. 17, no. 11, p. 113060, 2015.

[11] G. Karypis and V. Kumar, "Multilevel k-way hypergraph partitioning," *VLSI design*, vol. 11, no. 3, pp. 285–300, 2000.

[12] M. G. Whitney, *Practical fault tolerance for quantum circuits*. University of California, Berkeley, 2009.