

Scheduling Physical Operations in Quantum Circuits Using a Greedy Algorithm

M. Raeisi

Computer Engineering Department, Arak Branch Islamic Azad University, Arak, Iran
Email: mhb.raeisi@gmail.com

N. Mohammadzadeh

Computer Engineering Department, Shahed University, Tehran, Iran
Email: mohammadzadeh@shahed.ac.ir

Abstract—During the quantum physical design process, the second process of the quantum circuit design flow, we can use some optimization methods after layout generation to make the metrics better. Following this idea, this paper proposes a polynomial time design heuristic method to improve the latency of quantum circuits. It is composed of five steps that merge gate locations and then exchanges them to improve the latency. Experimental results show that the proposed method decreases the average latency of quantum circuits by about 15.95% for the attempted benchmarks.

Index Terms—quantum physical design, scheduling, latency, ion trap technology

I. INTRODUCTION

Silicon chips have become twice as fast every two years while the dimensions of the structures on them have become twice as small. The issue follows the laws of quantum mechanics on the atomic scale. Hence the behavior of computer circuits will have to be investigated based on quantum mechanical laws rather than classical physics [1] if the scaling continues at this rate. Although these quantum effects are great barriers in the classical CMOS progress, they can be used to develop a radically different form of computation [2].

Quantum computing helps us to solve certain problems thought to be intractable on a classical machine. For example, quantum algorithms solve classically hard problems like: factorization [3], unsorted database search [4], and simulation of quantum mechanical systems [5]. For example, in quantum cryptography, the non-cloning property of quantum states [6] and the phenomenon of entanglement [7] have been utilized to help in the exchange of secret keys between various parties, thus ensuring the security of cryptosystems using public key [8]. MagiQ Technologies [9] and IdQuantique [10] have built such cryptographic systems based on the single-photon communication.

A quantum algorithm needs a quantum circuit for a successful implementation. In a large picture view, the

quantum circuit design flow consists of two main tasks: synthesis and physical design. Optimization techniques might be useful to improve results of two main parts of quantum circuit design flow. In the recent works, techniques for physical synthesis [11], [12] were proposed to improve the objectives by manipulating layout or netlist locally considering layout information. In addition, an optimization technique [13] was proposed for the optimization of quantum circuits in the physical design stage. Following the optimization concept, in this paper a new optimization approach is proposed to improve the latency of quantum circuits. The proposed technique takes an initial netlist and a layout, and tries to merge and exchange gate locations. The goal of the optimization technique is to reach a circuit with a lower latency. Ion trap technology [14] is used as the underlying technology. Ion trap technology has been physically realized using universal elements for quantum computation with a clear scalable model [15].

The rest of this paper is organized as follows: an overview of the prior work is presented in Section 2, followed by an introduction to the ion trap technology in Section 3. Section 4 includes the details of the proposed optimization approach. Section 5 shows the experimental results, and Section 6 concludes the paper.

II. RELATED WORK

Besides significant work done on optimization in the quantum synthesis stage [16]–[21], a number of studies have been done on optimization in the quantum physical design stage.

Svore *et al.* [22], [23] suggested a design flow which takes a quantum program and generates its corresponding physical operations. The proposed design flow converts a high-level program into a low-level set of machine instructions scheduled on a fixed H-tree-based layout [23].

In a similar manner, Balensiefer *et al.* [24], [25] proposed a design flow that starts with a quantum description in QCL [26] for generating a technology-

¹QCL (Quantum Computation Language) defined by B. Omer [26] utilizes a syntax derived from C and provides a quantum simulator for code development and testing on a classical computing platform.

dependent netlist. The resulted netlist is scheduled on a fixed layout by a list-scheduling algorithm in the physical design phase [27].

Metodi *et al.* [28] proposed a tool for scheduling physical operations automatically, given a quantum circuit and a fixed grid-based layout structure. The same group proposed a uniform QLA² architecture [29] and extended it later in [30]. Also, hand-optimized layouts have been proposed in the literature [31].

Whitney *et al.* [32] proposed a quantum design flow which takes a description and generates its layout in ion trap technology. They suggested new heuristics for layout generation and scheduling. The proposed technique merges some gate locations during layout generation to improve latency. This approach can be considered an optimization technique in the physical design stage.

Dousti *et al.* [33] focus on minimizing the total latency of the circuit to minimize the error in the circuit. A CAD tool, called Quantum mapper based on Scheduling, Placement, and Routing or QSPR, was developed to perform this task automatically. More precisely, the destination qubit is fixed in one trap while the source qubit is moved to reach the destination. Quantum physical operations scheduler (QPOS) distinguishes between the source and destination operands of a two-qubit instruction during the routing step [34]. QPOS extracts a routing path for each of the ready-to-issue instructions. If there are any overlaps among these paths, QPOS selects an instruction to execute based on the following criteria: 1) highest initial priority, 2) lowest among of congestion that is going to be introduced by using the path, and 3) shortest path length. Finally, QPOS maps these paths to the quantum circuit fabric and uses a deadlock prevention algorithm to prohibit qubits to locate in a position that further movement is impossible.

Mohammadzadeh *et al.* [11], [12] introduced the physical synthesis concept for a quantum design flow to mitigate the effects of separate synthesis and physical design processes on the optimality of results. They proposed [11] a technique for physical synthesis in quantum circuits using gate-exchanging heuristic to improve the latency of quantum circuits. They also suggested [12] a new technique for physical synthesis using auxiliary qubit selection to improve the latency of quantum circuits. Recently, the same group proposed [13] a new optimization technique, called gate location changing, for the optimization of quantum circuits in the physical design stage. The proposed technique takes an initial netlist and a layout, and tries to change locations of the gates that are on the critical path, considering the scheduling information.

We consider a new approach that creates an optimum layout in acceptable perform time for larger circuits with more qubits. The goal of the proposed approach is to minimize the total latency.

III. ION TRAP TECHNOLOGY FOR QUANTUM COMPUTING

Ion-trap technology is the most promising technology for implementing quantum circuits to date [33]. Therefore, it is selected as the underlying technology. In ion trap technology, a physical qubit is an ion, and a gate is a location where a trapped ion may be operated upon by a modulated laser. Pulse sequences applied to discrete electrodes on the edges of the ion traps cause the ions to be trapped or ballistically moved between traps. Fig. 1a shows a layout that was experimentally demonstrated for a three-way intersection [35].

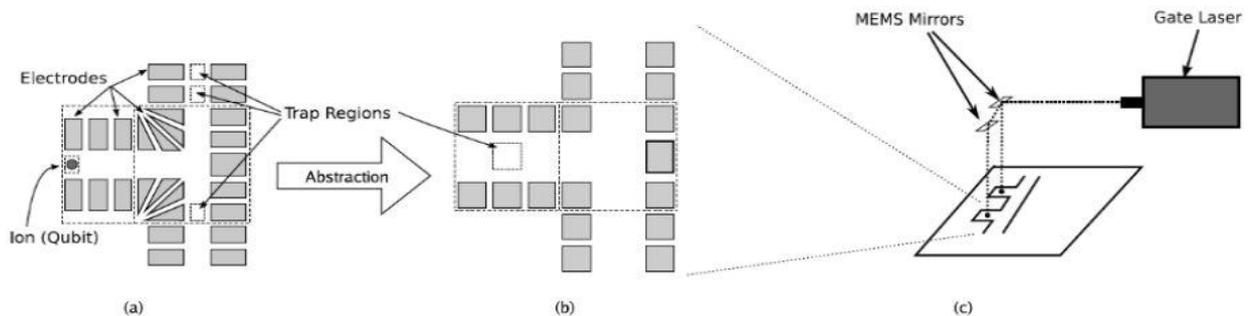


Figure 1. (a) Physical layout demonstrated for a T-junction (three-way intersection). (b) Abstraction of the circuit in (a), built using the straight channel and three way intersection macro blocks shown in Fig. 2. (c) MEMS mirrors placed above the ion traps plane guide the laser beams to gate locations [36].

Fig. 1 shows a possible mapping of a demonstrated layout (Fig. 1a) to macroblock abstractions (Fig. 1b). As Fig. 1c shows, the laser pulses are guided to the gate locations by an array of MEMS mirrors located above the ion trap plane to apply quantum gates [37].

Fig. 2 shows the library defined in [36]. Each macroblock consists of a 3×3 structure of trap regions and electrodes with some ports to allow qubit movement. The black squares are gate locations which may not be

performed at intersections or turns. We can use Different orientations of these macroblocks in a layout.

Some key characteristics of ion trap technology can be summarized as follows:

Rectangular channels lined with electrodes make “wires” in ion traps. Atomic ions (qubits) can be suspended above the channel regions and moved ballistically by application of voltages on the channel electrodes [38].

Any operation available in the ion trap technology can be performed at each gate location. This makes it possible

²Quantum Logic Array

to reuse gate locations for different operations within a quantum circuit.

Fabrication and control of ion traps in the third dimension is difficult. Therefore, scalable ion trap systems are two dimensional [35]. Thus, routing channels should have T-junction(s) or cross-junction(s) to allow ions to move from one channel to another.

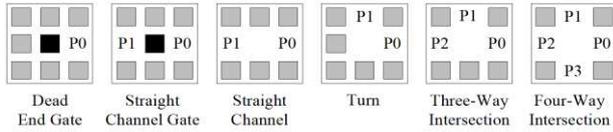


Figure 2. Basic macroblocks [36].

IV. THE PROPOSED APPROACH

The primary goal of the proposed approach is to decrease the latency of quantum circuits. To reach this goal, we proposed the flow presented in Fig. 3 that optimizes the initial layout.

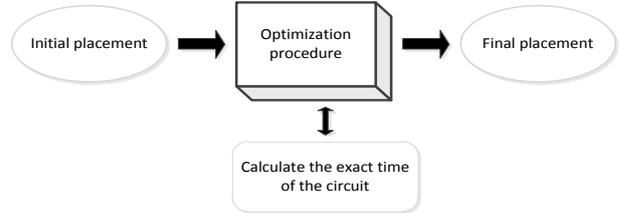


Figure 3. The general idea

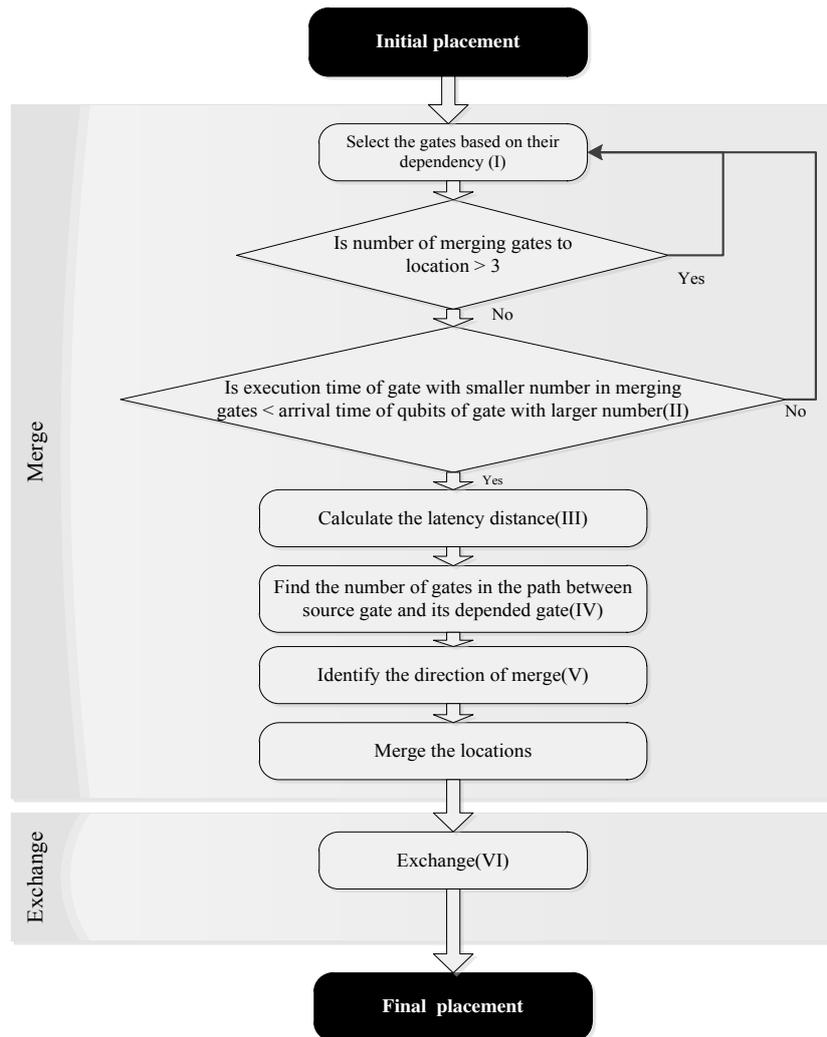


Figure 4. Optimization procedure

The optimization procedure takes the initial layout and generates the final layout. The procedure includes two stages: 1) merging that is based on gates dependency and consists of five parts, 2) exchanging. The optimization procedure is shown in Fig. 4.

The procedure starts with selecting the gates based on their dependency. We assume that the gate number 1 is at (i,j) . In step (I) we check the dependency of (i,j) with gate locations of $(i-1,j)$, $(i-1,j+1)$, $(i+1,j)$ and $(i+1,j+1)$.

In this way, three different situations may occur; if there is no dependency among this location and another four locations, we check next gate. If there is one adjacent gate, merge process starts. Otherwise, there are two adjacent gates; both of them enter into the merge process. We compare these two locations to select one of them. After step (I), number of merging gates in the location is checked. If the number of gates after merging is greater than three, we return to step (I). Else, we start

step (II). In step (II) if qubits of gate with larger number in merging gates arrive at the common location earlier than expected time they will not be allowed to enter in location. It causes to enter required qubit for the first gate; as a result, congestion and latency decrease in that place. We compute the arrival time of each qubit to merge location. If the intruder qubit had arrived to merger location, merging wouldn't be possible and we return to step (I), otherwise we start step (III). Direction of merge can be identified by step (III), (IV). The selected gate is the gate that its location is changed by the merge process. Distance of this gate with all its input and output gates is calculated. This distance before and after merge is named d_1 and d_2 respectively. The difference between d_1 and d_2 is called credit. In step (IV) the number of gates in the path between source gate and its depended gates is calculated. For each gate the specific weight is added to the credit that was calculated in the previous step. In step (V) the higher credit indicates the higher priority for direction of merge. The gate number is incremented by one and we repeat these steps until last gate location is reached. Finally, in step (VI) if the distance between the 1-qubit gates and their depended gates reduces after exchange, exchanging 1-qubit gates with empty spaces is possible. Following this idea, we find the adjacent node(s) of first 1-qubit gate and then a 3×3 square around the node(s) is considered. We place the 1-qubit gate in each empty space and calculate the distance between the 1-qubit gate and its depended gates. The minimum distance

represents the best location for the 1-qubit gate.

The second process of the proposed flow continues until there is no unprocessed 1-qubit gate. At the end of this step, the final layout is provided. In this way, we confront some problems such as, manner of leaving qubits from gate location, movement qubits in channel, blockage and congestion. We proposed a solution for each of these problems.

V. EXPERIMENTAL RESULTS

We experimented with a number of quantum circuit benchmarks from [39]-[45]. Physical latencies shown in Table I are used for the gates and for the two types of move operations in ion trap technology [39]. Table II shows the experimental results. The proposed algorithm is implemented in C#.

TABLE I. THE LATENCY VALUES FOR VARIOUS PHYSICAL OPERATIONS IN ION TRAP TECHNOLOGY [38]

Physical Operation	Latency (μs)
One-Qubit Gate	1
Two-Qubit Gate	10
Measurement	50
Zero Prepare	51
Straight Move	1
Turn	10
Idle (per μs)	N/A

TABLE II. LATENCY OF BENCHMARK CIRCUITS ACHIEVED BY THE PROPOSED TECHNIQUE COMPARED WITH WITHNEY'S METHOD

Benchmark name	Qubit Count	Gate Count	Latency (μs)		Improvement (%)
			Before applying the proposed flow	After applying the proposed flow	
Rent	6	9	110	92	16
P107	5	5	56	34	39
P190	7	12	93	84	9
P35	7	12	164	115	29
S7(2)	7	12	114	88	22
Grover	5	9	133	114	14
[[5,1,3]]	5	13	225	168	25
Ex4	3	7	139	139	0
P25	4	10	190	105	44
[[7,1,3]]	7	14	226	140	38
[[9,1,3]]	9	28	375	370	1
[[11,1,3]]	11	40	547	436	20
[[12.1.4]]	12	51	667	577	13
[[19.1.7]]	19	118	1211	1163	3
[[20,1,6]]	20	134	1204	1141	5
[[28.2.8]]	28	255	1754	1528	12
[[29.1.11]]	29	275	1735	1663	4
[[32.2.8]]	32	264	1934	1889	2
[[35.1.10]]	35	389	4612	3973	13
[[40.3.10]]	40	483	6191	5571	10
Average : 15.95%					

Table II shows the latency of the benchmark circuits achieved by the proposed technique compared with Withney's Method [32]. The latency of circuits before and after applying the proposed technique are shown in

the third and the forth columns, respectively. The results reported in the column "Before Applying the proposed flow" is obtained by the best prior physical design flow in terms of latency. The column "Improvement" shows the

latency improvement resulted from the proposed technique in this paper. As can be seen, an average improvement of 15.95% is achieved in the latency of the benchmarks. The results of Table II are summarized in Fig. 5 in term of the latency.

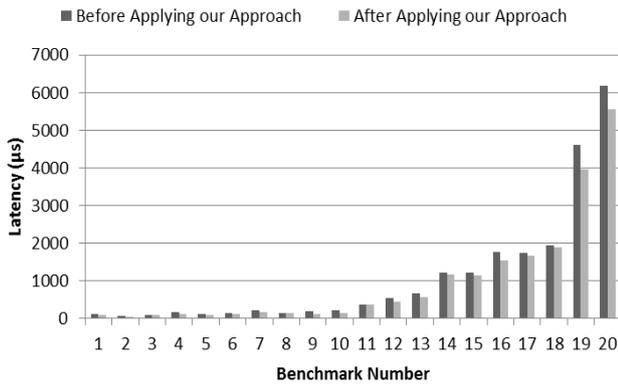


Figure 5. The latency reduction achieved by the proposed approach

VI. CONCLUSION

In this paper, an optimization technique was proposed which modifies the layout to improve the latency of quantum circuit execution. It is composed of merge and exchange processes. In this way, layout and scheduling information is used to find better gate locations for merge and exchange decreasing the overall latency. The proposed technique was applied to a set of benchmarks. Experimental results show that the proposed technique improves the latency of quantum circuits by about 15.95% for the attempted benchmarks.

REFERENCES

- [1] S. Lloyd, "Quantum-Mechanical computers," *Scientific American*, vol. 273, pp. 140-145, 1995.
- [2] R. P. Feynman, "Quantum mechanical computers," *Foundations of Physics*, vol. 16, pp. 507-531, 1986.
- [3] P. Shor, "Polynomial time algorithms for prime factorization and discrete logarithms on a quantum computer," *SIAM Journal on Computing*, vol. 26, no. 5, pp. 1484-1509, 1997.
- [4] L. Grover, "A fast quantum mechanical algorithm for database search," in *Proc. ACM Symposium on Theory of Computing*, 1996, pp. 212-219.
- [5] C. Zalka, "Simulating quantum systems on a quantum computer," *Proceedings: Mathematical, Physical and Engineering Sciences*, vol. 454, pp. 313-322, 1998.
- [6] W. K. Wootters and W. H. Zurek, "A single quantum cannot be cloned," *Nature*, vol. 299, pp. 802-803, 1982.
- [7] E. Schrödinger, "The present situation in quantum mechanics," *Naturewiss*, vol. 48, pp. 807-812, 1935.
- [8] D. Welsh, *Codes and Cryptography*, Oxford University Press, 1988.
- [9] (2010). Q. Magi Technologies, Inc. [Online]. Available: <http://www.magiqtech.com/MagiQ/Home.html>
- [10] (2010). Our business: Network encryption-photon counting-randomness. [Online]. Available: <http://www.idquantique.com>
- [11] N. Mohammadzadeh, M. Sedighi, and M. S. Zamani, "Quantum physical synthesis: improving physical design by netlist modifications," *Elsevier Microelectronics Journal*, vol. 41, pp. 219-230, 2010.
- [12] N. Mohammadzadeh, M. S. Zamani, and M. Sedighi, "Auxiliary qubit selection: A physical synthesis technique for quantum circuits," *Springer Quantum Information Processing Journal*, 2011.
- [13] N. Mohammadzadeh, M. Sedighi, and M. S. Zamani, "Gate Location changing: An optimization technique for quantum circuits," *World Scientific International Journal of Quantum Information (IJQI)*, vol. 10, no. 3, 2012.
- [14] H. Häffner, C. F. Roos, and R. Blatt, "Quantum computing with trapped ions," *Physics Reports*, vol. 469, pp. 155-203, 2008.
- [15] 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, Jun. 2002.
- [16] D. Maslov, et al., "Quantum circuit simplification and level compaction," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 27, no. 3, pp. 436-444, 2008.
- [17] D. Maslov, et al., "Quantum circuit simplification using templates," in *Proc. Conference on Design, Automation and Test in Europe*, Mar. 2005, pp. 1208-1213.
- [18] D. Maslov, G. W. Dueck, and D. M. Miller, "Simplification of toffoli networks via templates," in *Proc. 16th Symposium on Integrated Circuits and System Design*, 2003, pp. 53-58.
- [19] V. V. Shende, et al., "Scalable simplification of reversible circuits," in *Proc. IWLS*, 2003.
- [20] M. Sedlak and M. Plesch, "Towards optimization of quantum circuits," *Central European Journal of Physics*, vol. 6, no. 1, pp. 128-134, 2008.
- [21] A. Aho and K. Svore, "The design and optimization of quantum circuits using the palindrome transform," in *Proc. ERATO Conference on Quantum Information Sciences (EQIS)*, 2003.
- [22] K. Svore, et al., "A layered software architecture for quantum computing design tools," *Computer*, vol. 39, no. 1, pp. 74-83, 2006.
- [23] K. Svore, et al., "Toward a software architecture for quantum computing design tools," in *Proc. 2nd International Workshop on Quantum Programming Languages (QPL)*, 2004, pp. 145-162.
- [24] S. Balensiefer, L. Kreger-Stickles, and M. Oskin, "QUALE: quantum architecture layout evaluator," *Proceedings of SPIE the international society for optical engineering*, Vol. 5815, pp. 103-114, 2005.
- [25] S. Balensiefer, L. Kreger-Stickles, and M. Oskin, "An evaluation framework and instruction set architecture for ion-trap based quantum micro-architectures," in *Proc. International Symposium on Computer Architecture (ISCA)*, 2005, pp. 186-196.
- [26] B. Omer, "Quantum programming in QCL," Master thesis, Technical University of Vienna, 2000.
- [27] T. Yang and A. Gerasoulis, "List scheduling with and without communication delays," *Journal of Parallel Computing*, vol. 19, no. 12, pp. 1321-1344, 1993.
- [28] T. S. Metodi, D. D. Thaker, A. W. Cross, F. T. Chong, and I. L. Chuang, "Scheduling physical operations in a quantum information processor," in *Proc. SPIE*, 2006.
- [29] T. Metodi, et al., "A quantum logic array microarchitecture: Scalable quantum data movement and computation," in *Proc. 38th International Symposium on Microarchitecture (MICRO)*, 2005, pp. 305-318.
- [30] D. Thaker, et al., "Quantum memory hierarchies: efficient designs to match available parallelism in quantum computing," in *Proc. 33rd International Symposium on Computer Architecture (ISCA)*, 2006, pp. 378-390.
- [31] D. E. Copesey, "Designing scalable quantum computer architectures: Layout and initialization," PH.D. Thesis, University Of California, 2005.
- [32] M. Whitney, N. Isailovic, Y. Patel, and J. Kubiawicz, "Automated generation of layout and control for quantum circuits," *Proceedings of Computing Frontiers*, pp. 83 - 94, 2007.
- [33] M. J. Dousti and M. Pedram, "Minimizing the latency of quantum circuits during mapping to the ion-trap circuit fabric," in *Proc. Design Automation and Test in Europe*, Mar. 2012.
- [34] P. Aliferis, D. Gottesman, and J. Preskill, "Quantum accuracy threshold for concatenated distance-3 codes," *Quant. Inf. Comput.*, pp. 97-165, 2005.
- [35] D. Hucul, et al., "On the transport of atomic ions in linear and multidimensional ion trap arrays," *Journal of Quantum Information and Computation*, vol. 8, no. 6, pp. 0501-0578, 2008.
- [36] M. Whitney, "Practical fault tolerance for quantum circuits," PH. D. Thesis, University of California, Berkeley, 2009.
- [37] J. Kim, et al., "System design for large-scale ion trap quantum information processor," *Journal of Quantum Information and Computation*, vol. 5, no. 7, pp. 515-537, 2005.

- [38] J. Chiaverini, *et al.*, "Surface-Electrode architecture for ion-trap quantum information processing," *Journal of Quantum Information and Computation*, vol. 5, no. 5, pp. 419-439, 2005.
- [39] N. Isailovic, "An investigation into the realities of a quantum datapath," Technical Report, University of California at Berkeley, 2010.
- [40] M. Grassl. (2013). Quantum error-correcting codes. [Online]. Available: <http://iaks-www.ira.uka.de/home/grassl/QECC/Cyclic/index.html>
- [41] M. Whitney, *et al.*, "A fault tolerant, area efficient architecture for shor's factoring algorithm," in *Proc. ISCA'09*, 2009.
- [42] R. Van Meter, "Architecture of a quantum multicomputer optimized for shor's factoring algorithm," Technical Report, Keio University, Jul. 2006.
- [43] A. Cross, "Synthesis and evaluation of fault-tolerant quantum computer architectures," Ph.D. Thesis, Massachusetts Institute of Technology, 2005.
- [44] I. Chuang. (2013). Quantum error-correcting codes. [Online]. Available: <http://www.media.mit.edu/quanta/qasm2circ/index.html>
- [45] T. S. Metodi and F. T. Chong, *Quantum Computing for Computer Architects*, Morgan & Claypool, 2006.



Mahbobeh Raeisi received the B.S. degree in hardware computer engineering from Shahid Bahonar University of Kerman in 2007. She received the M.Sc. degree in computer architecture engineering from Arak Branch Islamic Azad University in 2013. She is currently a lecturer at Scientific-Applied University of Medical Science in Shiraz. Quantum computing is her field of study.



Naser Mohammadzadeh received the B.S. and M.Sc. degrees in computer engineering, both from Sharif University of Technology, in 2004 and 2006, respectively. He received the Ph.D. degree in computer architecture engineering from Amirkabir University of Technology in 2010. He is currently an assistant professor at Shahed University. His interests include combinational optimization with applications to the design of integrated circuits, as well as quantum logic circuits.