Scheduling Physical Operations in Quantum Circuits Using a Greedy Algorithm

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

N. Mohammadzadeh
Computer Engineering Department, Shahed University, Arak, 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 QLA\(^2\) 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](image)

**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 place to apply quantum gates [37].

Fig. 2 shows the library defined in [36]. Each macroblock consists of a 3x3 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...
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.

**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.

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

![Figure 2. Basic macroblocks [36].](image)

![Figure 3. The general idea](image)

![Figure 4. Optimization procedure](image)
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. The gate number is incremented by 1 and we repeat these steps until last gate location is reached. Finally, in step (VI) if the distance between the one and we repeat these steps until last gate location is reached. If the intruder qubit had arrived to merger location, moving 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. The gate number is incremented by 1 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 3x3 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>
<thead>
<tr>
<th>Table II. Latency of Benchmark Circuits Achieved by the Proposed Technique Compared with Withney’s Method</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Benchmark name</strong></td>
</tr>
<tr>
<td>--------------------</td>
</tr>
<tr>
<td>Rent</td>
</tr>
<tr>
<td>P107</td>
</tr>
<tr>
<td>P190</td>
</tr>
<tr>
<td>P35</td>
</tr>
<tr>
<td>S7(2)</td>
</tr>
<tr>
<td>Grover</td>
</tr>
<tr>
<td>[[5,1,3]]</td>
</tr>
<tr>
<td>Ex4</td>
</tr>
<tr>
<td>P25</td>
</tr>
<tr>
<td>[[7,1,3]]</td>
</tr>
<tr>
<td>[[9,1,3]]</td>
</tr>
<tr>
<td>[[11,1,3]]</td>
</tr>
<tr>
<td>[[12,1,4]]</td>
</tr>
<tr>
<td>[[19,4,7]]</td>
</tr>
<tr>
<td>[[20,1,6]]</td>
</tr>
<tr>
<td>[[28,2,8]]</td>
</tr>
<tr>
<td>[[29,1,11]]</td>
</tr>
<tr>
<td>[[32,2,8]]</td>
</tr>
<tr>
<td>[[35,1,10]]</td>
</tr>
<tr>
<td>[[40,3,10]]</td>
</tr>
</tbody>
</table>

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 terms of the latency.

![Latency Improvement Graph](Image)

**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

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.