



Quantum circuit physical design flow for the multiplexed trap architecture



Naser Mohammadzadeh*, Elaheh Taqavi

Department of Computer Engineering, Shahed University, Tehran, Iran

ARTICLE INFO

Article history:

Received 20 June 2015

Accepted 29 February 2016

Available online 3 March 2016

Keywords:

Quantum circuit

Multiplexed trap architecture

Physical design

ABSTRACT

Physical design is the second process in the design flow of quantum circuits that receives a netlist as input and generates a layout at a target technology. Quantum physical design problem is intractable. This process tackles the operation scheduling, placement, and qubit routing problems. Some approaches have been proposed for the physical design in ion trap technology that is currently one of the most advanced quantum technologies. These methods do not use all capabilities of the technology. Focusing on this issue, in this paper, a physical design flow is proposed for the multiplexed trap architecture utilizing the capabilities provided by the technology not considered by other approaches to improve the latency and area metrics. Experimental results show that the proposed approach decreases the average latency of quantum circuits by about 39.5% for the attempted benchmarks.

© 2016 Elsevier B.V. All rights reserved.

1. Introduction

Quantum effects have been a major concern in classical computers, more specifically in metal-oxide-semiconductor technology (CMOS), as the feature size shrinks into tens of nanometers [1]. Quantum effects such as entanglement and superposition are amplified in quantum computers. They operate on the entangled superposition states, where the power of quantum algorithms comes from. Theoretically, quantum computers, computers using quantum effects, could outperform their classical counterparts when solving certain problems. Factorization [2], unsorted database search [3], and the simulation of quantum-mechanical systems [4] are some classic hard problems that benefit from quantum algorithms. For example, the successful large-scale implementation of Shor's integer factorization [2] can have a deep effect on the RSA cryptosystem used within the domain of electronic commerce.

A quantum algorithm requires a quantum circuit for successful implementation. Generally, the quantum circuit design flow can be divided into two main processes: synthesis and physical design. The synthesis process takes a description and generates a technology-dependent netlist. On the other hand, the physical design process creates a specific layout of a circuit based on the target technology. The physical design consists of scheduling, placement, and qubit routing processes.

Several candidate technologies have been proposed for realization of a quantum computer to date [5]. Among these technologies, ion trap technology [6, 7] is currently one of the most advanced [8]. Trapped ions have a long decoherence time and have shown good potential for scalability [9]. As a result, it is considered as one of the most promising technologies for implementing a scalable universal quantum computer. Therefore, in this paper, it is selected as the substrate technology.

The prior algorithms proposed for physical design in ion trap technology have two assumptions. First, they assume that the maximum number of qubits placed in each gate location is two [8]. Second, they restrict quantum operations to only one- and two-qubit gates. With recent advancements in ion trap technology, these assumptions can be relaxed [8, 10].

Abiding to the first assumption has two main disadvantages. First, it increases communication in the layout. Due to the high cost in time and fidelity of communication [11], algorithms and designs proposed for physical design must utilize locality and minimize communication to achieve high reliability and low latency. Communication decreases the fidelity of the circuit. For many designs, nearly 50% of systematic failures occur due to errors in communication [12], and moving and turning ions remain the most error-prone operations by far, transforming the relatively few instances of these operations into major sources of systemic error. Communication also increases the latency. The cooling, splitting, and recombining ions are by far the two slowest operations. Layouts must be designed to avoid unnecessarily cooling, splitting, and recombining ion chains. Cooling is needed because ions collect about one-tenth of a quanta of heat for each micron they travel.

* Corresponding author at: Persian-Gulf Highway, Shahed University, Tehran, Iran, Tel.: +98 21 51212098; fax: +98 21 51212021.

E-mail addresses: mohammadzadeh@shahed.ac.ir (N. Mohammadzadeh), e.taqavi@shahed.ac.ir (E. Taqavi).

Heating degrades some types of two-qubit gates, and thus ions must be re-cooled at their destinations. Laser cooling is a comparatively slow process that takes about 10 ms to cool to the lowest motional state, but entire chains of ions can be cooled to less than one quanta of heat. To preserve the logical state of a qubit in practice, ions must be cooled via *sympathetic cooling* [13]. Sympathetic cooling requires an extra ion of a different species to be placed in a linear chain to be cooled. This extra ion is periodically cooled by laser, allowing the computational ions to cool without damaging their internal electronic state. The second disadvantage is that it results in a large number of gate locations. In the context of ion-traps, a physical limitation is the significant overhead from switching lasers from one trap to the next over the entire layout at random order. Therefore, physical design must minimize the locations in the physical layout where lasers are applied.

Although the aforementioned statements imply that increasing the numbers of qubits in each gate location decreases the error and the latency, but it seems impractical to construct a device for a large number of qubits by trapping all ions in the same gate location because as the length of the ion chain increases, the vibrational modes become progressively harder to identify [14], decreasing the gate fidelities. These modes also couple more strongly to ambient fields, increase the heating rate and, hence, the dephasing rate. Up to date there are a couple of routes known, which potentially ease these technological challenges. Almost all of them are based on distributing the ions across different traps and interconnect these traps. Experiments show that placing up to ~ 10 qubits in each gate location gives a good fidelity for gates [8]. Therefore, the physical design flow proposed in this paper restricts the number of qubits in each location up to 10 qubits. Placing more than two qubits in each gate location also has another important advantage. Two gates having the same control qubit can be potentially performed at the same time. This capability can decrease the circuit latency.

The second assumption can be relaxed because advancements in the ion trap technology made it possible to implement multi-qubit gates such as Toffoli [10].

To the best of our knowledge, no flow has proposed that relaxes one or both assumptions. Focusing on this issue, in this paper, a physical design flow is presented that utilizes these capabilities to minimize the latency and area. Furthermore, a detailed analysis is conducted on the effect of the number of qubits in each gate location on latency.

This paper is organized as follows: an overview of the prior works is presented in Section 2, followed by the description of the multiplexed trap architecture in Section 3. Section 4 contains the details of the proposed physical design flow and then Section 5 shows the experimental results and discusses on the result. Finally, Section 6 concludes the paper.

2. Related work

In this section, a short review on related papers in the field of quantum physical design is presented to illustrate the position of the proposed approach in the literature.

Balensiefer et al. [15, 16] proposed QUALE, which is a set of tools for designing microarchitectures for ion trapped quantum computers. They used as late as possible method (ALAP) to schedule instructions of a quantum circuit. In this work, a greedy algorithm was used for initial placement of qubits.

Metodi et al. [17] presented an instruction scheduler called QPOS. This tool employs a combination of list scheduling and as soon as possible scheduling methods (ASAP) for quantum instruction scheduling and a priority-based routing algorithm for routing.

Whitney et al. [18] proposed a computer-aided design flow for the layout, scheduling, and control of ion trap-based quantum cir-

cuits. They proposed two heuristics for layout design. The first heuristic is a greedy algorithm that is appropriate for small circuits. The dataflow-graph-based algorithm was proposed for larger circuits aiming at placing and routing of a circuit.

Mohammadzadeh et al. [19] proposed an optimization technique applying gate location changing (GLC) to reduce the latency of quantum circuits. The proposed technique finds critical paths by using layout and scheduling information, and reduces their latency by modifying locations of the gates on the critical path. Moreover, in [20–23] they introduced the physical synthesis concept. The physical synthesis includes techniques that change the netlist using layout information to reach better circuits in terms of latency and/or area.

Yazdani et al. [24] presented a physical design flow for quantum circuits in ion trap technology, which consists of two parts. First, a scheduler takes a description of a circuit and finds the best order for the execution of its quantum gates using ILP. Then a layout generator receives the schedule produced by the scheduler and generates a layout for the circuit using a graph-drawing algorithm [25]. Moghdam et al. [26] proposed a scalable physical design method for scheduling and layout generation problems. This method uses a hierarchical approach for generating layout.

Dousti et al. [27] introduced a multi-core quantum processor architecture and a scalable quantum mapper which contributes an effective method for sharing ancilla qubits. The quantum mapper divides a quantum circuit into a couple of quantum kernels. Each kernel is composed of k parts such that each part will run on exactly one of k available cores.

Goudarzi et al. [28] proposed quantum physical mapper which considers the effect of the routing time on quantum instruction scheduling and placement. They formulated the scheduling and placement problem and presented solution algorithms for these problems. A mapping tool for ion trap technology, called QSPR was proposed in [29]. It uses an iterative approach for placement of qubits and it improves the routing solution by simultaneously moving the source and destination qubits toward a designated site.

3. Multiplexed trap architecture

Ion traps present the most promising technology to build a large-scale quantum computer. They are fabricated using existing silicon-manufacturing facilities at scales that are relatively easy for the experiments. Nonetheless, only extremely small-scale experimental systems have been built and many fundamental technology parameters, such as fidelity and operation timing, remain in flux. Despite this degree of technological uncertainty, computer architectures [30] have been successful in helping to guide the device physics research [31].

Trapped ions as a processor for quantum information were first proposed by Cirac and Zoller in 1995 [7]. The architecture consists of one string of ions stored in a linear quadruple trap. Two long-lived electronic levels of each ion are used to implement one qubit, well protected against environmental disturbances. The necessary interactions for gate manipulations can be precisely switched on and off by focused laser beams addressing the qubits individually.

In 1998 the group at NIST proposed a multiplexed trap architecture [9, 32] that alleviated the problems of the Cirac/Zoller proposal and is modular, so scaling to higher qubit numbers seems to be feasible. The basic idea is to expand the original architecture to an array of many independently controllable subtraps (Fig. 1). Qubits that do not partake in a given step of the algorithm are stored in “memory” regions. To execute a gate on certain qubits, they are separated out of the memory regions and shifted into one of the “processor” regions. Moving ion qubits around does not lead to decoherence in the computational Hilbert space spanned by the qubits, since the motion is only used for coupling the

Download English Version:

<https://daneshyari.com/en/article/461201>

Download Persian Version:

<https://daneshyari.com/article/461201>

[Daneshyari.com](https://daneshyari.com)