

# QLA: Quantum Logic Array Microarchitecture, a Brief Overview

Tzvetan Metodi<sup>†</sup>, Darshan Thaker<sup>†</sup>, Andrew Cross<sup>‡</sup>, Frederic T. Chong<sup>†</sup>, and Isaac L. Chuang<sup>‡</sup>

<sup>†</sup> *University of California at Davis, {tmetodie, ddthaker, fchong}@ucdavis.edu*

<sup>‡</sup> *Massachusetts Institute of Technology {awcross, ichuang}@mit.edu*

## ABSTRACT

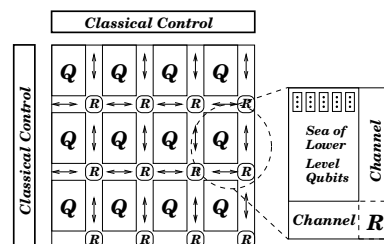
Recent experimental advances have demonstrated technologies capable of supporting scalable quantum computation. A critical next step is how to put those technologies together into a scalable, fault-tolerant system that is also feasible. We propose a Quantum Logic Array (QLA) microarchitecture that forms the foundation of such a system. QLA is a quantum computer architecture designed for efficient error-correction and error-free communication, regardless of distance. Our designs are validated by analytical reasoning and also by simulation. We emphasize the importance of a datapath oriented large-scale quantum architecture for solving realistic problems, and show how the QLA achieves such design goals. In addition, we introduce ARQ, a tool used to map quantum applications to fault-tolerant architectures.

## 1. INTRODUCTION

Quantum computation draws its power from the fact that a single quantum bit, *qubit*, which can be implemented by the polarization states of a photon or the spin of a single atom, is always in a probabilistic superposition of the binary “0” and “1” states. With  $N$  qubits a quantum computer can be in  $2^N$  unique states at any given time. These states can be inter-correlated such that a single logic gate, expressed as a unitary operator, can act on all possible  $2^N$  states. This exponential speedup inherent in quantum computing combined with the ability to harness the information through gate manipulation [4] has led to several quantum algorithms with substantial advantages over traditional computation. The most significant is Shor’s algorithm for factoring the product of two large primes in polynomial time [9]. Additional algorithms include Grover’s fast database search; adiabatic solution of optimization problems; precise clock synchronization; quantum key distribution; and recently, Gauss sums and Pell’s equation.

Quantum data is inherently very unstable, which leads to a lack of reliable operations that can be performed on it. Also if left idle, this quantum data will interact with its environment and lose state, a process called *decoherence*. Finally, there is the difficulty of transmitting quantum data between computational units without losing state. This implies that the greatest challenge towards a large, practically useful and relevant quantum computer, is designing an architecture that incorporates the required amount of fault-tolerance while minimizing overhead.

Previous work in large-scale quantum architecture [8, 3] has led to the consideration of several main scalability issues that must be taken into account: **a)** Reliable and realistic implementation technology that allows the initialization of a quantum system to a known state and reliable physical measurement and gate operations. **b)** Robust error correction and fault-tolerant structures that minimize the overhead of error correcting codes while maintaining high system reliability. **c)** Efficient quantum resource distribution



**Figure 1: High-Level quantum computer structure, where a full-size computer consists of interconnected logical qubits connected with programmable communication network. The letters  $R$  denote an integrated switch islands for redirecting quantum data coming from nearby logical qubits or other repeater islands.**

that allows each quantum bit (qubit) to be a reliable physical transmitter of information.

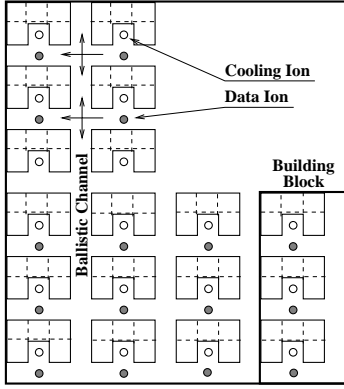
In this letter we introduce a Quantum Logic Array (QLA) microarchitecture, which enables substantial performance improvements critical to supporting full-scale applications such as Shor’s factorization algorithm. In the next section we describe the QLA’s approach to leveraging the three architecture requirements described in the previous paragraph, which is followed by final thoughts and discussion.

## 2. ARCHITECTURE OVERVIEW

At the lowest level QLA is based on the trapped ion-technology [2, 7], which uses a single trapped atomic ion as a storage for a single unit of quantum data. In particular QLA is based on the highly scalable model of (CCD) style ion-trap quantum information processing architecture proposed by Kielpinski et al [7]. This model consists of ions trapped in interconnected trap arrays and moved from trap to trap to interact [1].

A high-level schematic of our ion-trap quantum information processor is shown in Figure 1. The figure shows a number of logical computational units (denoted by the letter  $Q$ ) separated by long range teleportation based communication channels. Each computational unit is a sea of physical atomic ions as shown in Figure 2. The quantum ion-trap processor is surrounded by classical processors, which are used to control the execution of almost everything, from processing quantum measurement information to scheduling of the laser pulses that operate on the ions. What follows in the next paragraphs is a high-level description of the QLA ion-trap quantum computing processor.

**Block Structure for Error Correction:** The QLA structure fits



**Figure 2: The building blocks of the QLA microarchitecture. The figure consists of 6 level 1 building blocks, where the far right side outlines a single block. The circles are ions with data (solid) and cooling (clear).**

naturally to quantum error correction because the structure of the building blocks reflects the error-correction algorithm used. Each basic building block represents a single level-one logical qubit formed by encoding a number of *physical* ion-qubits as shown in Figure 2. Each logical qubit preserves the property of being in a superposition of “0” and “1” much like a single physical qubit. Encoding once more we can create a *logical qubit* at level 2 recursion with  $N^2$  physical ion-qubits. With each level,  $L$ , of encoding the probability of failure of the system scales as  $p_0^{2^L}$ , where  $p_0$  is the failure rate of each individual physical components. For simplicity, Figure 2 is drawn to show the level 1 blocks of a 3-bit error correcting code, but the structure is easily extended to 7-bit and larger codes. As the figure shows, each building block consists of data ions supported by their cooling ions and trapped between the electrode cells. The investment in communication channels for ballistic ion movement around the physical qubits allows us to limit the high costs of turning. Any two qubit gate at any level of encoding requires at most 2 turns per physical ion in each direction. Furthermore, the adaptability of the QLA design to the application being executed allows us to structure the logical qubits such that they fully comply with the fault-tolerant error correction requirements [6].

**Logical Interconnect:** The computational units denoted by the letter  $Q$  in Figure 1 are in fact encoded logical qubits that represent a single qubit of information. Each logical qubit is a regular structure of physical ions as shown in Figure 2 controlled by sequences of *laser pulses*. The logical qubits are positioned on the substrate in a regular array fashion, connected with a tightly integrated repeater-based [5], where qubits are *teleported* from one repeater to the next (see Figure 1). This makes the high-level design very similar to classical tile based architectures. The key difference is that the communication paths must account for data errors in addition to latency. The communication paths are composed of similar physical building blocks as the logical qubits. The integrated repeaters denoted with the letter  $R$  in Figure 1 are called *teleportation islands* that redirect traffic in the 4 cardinal directions by teleporting data from one repeater to the next. This interconnect design is one of the key innovative features of our quantum architecture, as it allows us to completely overlap communication and computation, thus eliminating communication latency at the application level of the program.

**Programming The Architecture:** All scheduling and physical

control is performed by the classical processors surrounding the quantum machine. Since physical quantum operations have a latency several orders of magnitude larger than classical operations, a sophisticated classical processor will easily be able to schedule the operations at run-time throughout the execution of the algorithm. Our general purpose quantum simulator ARQ is the closest we have come to modeling a full scale machine. ARQ takes a description of a general quantum circuit with a sequence of quantum gates as an input, maps them onto a specified physical layout, and generates pulse sequence files, which are then executed on the general quantum architecture simulator.

### 3. CONCLUSION

The QLA architecture leverages current quantum architectures research, however its development must also leverage the vast amount of knowledge and research from classical architectures. Although early results suggest that the QLA architecture can be used to factor numbers as large as module 2048, several critical issues quickly come to mind for the advancement of the design: 1) Relaxing the technology restrictions within our design will lead to quicker realization of the QLA microarchitecture. 2) Classical resources must be optimized both in quantity and usage complexity through clever scheduling and representation of quantum operations. 3) The area of a QLA ion-trap chip for even the factoring of a 128-bit number is estimated to be roughly 0.45 square meters. This amounts to a chip size of 33 centimeters at each edge if we assume a square chip. This is a substantial fabrication and yield challenge. QLA offers an inherent redundancy within itself, which we can explore to raise the yield by diagnosing and masking out defects. The fabrication challenges, however, suggest that a multi-chip solution for solving such large problems is desirable.

### 4. REFERENCES

- [1] M. Barrett, J. Chiaverini, T. Schaetz, J. Britton, W. Itano, J. Jost, E. Knill, C. Langer, D. Leibfried, R. Ozeri, and D. Wineland. Deterministic quantum teleportation of atomic qubits. *Nature*, 429, 2004.
- [2] J. I. Cirac and P. Zoller. Quantum computations with cold trapped ions. *Phys. Rev. Lett.*, 74:4091–4094, 1995.
- [3] D. Copley and et. al. The effect of communication costs in solid-state quantum architectures. *Symposium on Parallel Architectures and Applications (SPAA)*, 2003.
- [4] D. Deutsch. Quantum theory, the church-turing principle and the universal quantum computer. *Proc. R. Soc. Lond.*, A 400:97117, 1985.
- [5] W. Dur, H. J. Briegel, J. I. Cirac, and P. Zoller. Quantum repeaters based on entanglement purification. *arXiv:quantu-ph/9808065*, 1998.
- [6] D. Gottesman. Fault-tolerant quantum computation with local gates. *arXiv e-print quant-ph/9903099*, 1999.
- [7] D. Kielpinski, C. Monroe, and D. Wineland. Architecture for a large-scale ion-trap quantum computer. *Nature*, 417:709–711, 2002.
- [8] M. Oskin, F. Chong, and I. Chuang. A practical architecture for reliable quantum computers. *IEEE Computer*, January, 2002.
- [9] P. Shor. Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. *35<sup>th</sup> Annual Symposium on Foundations of Computer Science, IEEE Computer Society Press, Los Alamitos, CA*, pages 124–134, 1994.