

A General Purpose Architectural Layout for Arbitrary Quantum Computations

Tzvetan S. Metodiev[†], Darshan Thaker[†], Andrew W. Cross[‡], Frederic T. Chong[†] and Isaac L. Chuang[‡]

[†]University of California at Davis, One Shields Avenue, Davis, CA 95616, USA

[‡]Massachusetts Institute of Technology, 77 Massachusetts Avenue, Cambridge, MA 02139

ABSTRACT

Exploiting recent advances in quantum trapped-ion technologies, we propose a scalable, fault-tolerant quantum computing architecture that overcomes the fundamental challenges of building a full-scale quantum computer and leaves the fabrication a daunting but primarily an engineering concern. Using a hierarchical array-based design and a quantum teleportation communication protocol, we are able to overcome the primary scalability challenges of reliability, communication, and quantum resource distribution.

In particular, we present a reconfigurable quantum circuit substrate, or "quantum FPGA" (qFPGA) which allows efficient implementation of universal quantum gates and error correction. We use this qFPGA as a basic building block for an array structure that scalably provides communication channels and quantum resource distribution. We exploit a hierarchical combination of ballistic transport of data ions and quantum teleportation to reduce the cost of reliable communication from exponential to polynomial in distance.

By using a set of simulation tools we are able to evaluate a hypothetical design of a future general purpose quantum computer and describe the execution of a fault-tolerant Toffoli gate construction. Without considering classical control constraints and assuming best-possible ion-trap parameters our computer consists of level 2 encoded qubits with the Steane $[[7, 1, 3]]$ code tightly connected by the teleportation interconnect, and capable of executing a fault-tolerant Toffoli gate in roughly 2.3 seconds. This translates to factoring a 128-bit number in slightly over 40 hours in circuits dominated by Toffoli gates.

1. INTRODUCTION

In recent years, trapped ions^{1–3} have become a leading implementation candidate for the realization of quantum computers. CCD-style ion-trap quantum information processing architectures have been proposed by Kielpinski et al,² and recent experimental progress has dramatically increased the potential scalability of this technology.

The existence of a scalable technology brings forth the importance of investigating elementary architectural concepts for large-scale machines. Previous studies⁴ have proposed a high-level, reliable architecture that can be programmed for arbitrary quantum computations. In fact, a reliable system demands a dynamically programmable architecture, where the quantum bits can be reprogrammed and reinitialized in place, in order to achieve the desired reliability.^{4, 5}

We present a reprogrammable quantum circuit layout, a *quantum FPGA* or qFPGA, analogous to classical Field-Programmable Gate Arrays. Our qFPGA is capable of implementing any quantum circuit and optimized for the necessary requirements for a reliable machine: maximum parallelism, initialization and state preparation on demand and in place, intermediate error correction for traveling logical qubits, and efficient error correction. The qFPGA consists of tiling blocks of encoded physical ion-qubits in a grid. The structure of each block is optimized and specific to the base level-one error-correction algorithm used and each block represents a single level-one logical qubit. The physical ion-qubits can communicate entirely in parallel with the surrounding blocks in all four cardinal directions. Individual blocks can further be used for encoding data qubits at higher recursion levels, ancilla qubits for error correction, teleportation, or simply extra physical ion-qubits. A typical circuit execution will consist of determining the role of each block, then combining this information with quantum computation primitives and classical computation and control flow into a single instruction stream.

Additional author information: (Contact T.S.M. for correspondence)

T.S.M.: E-mail: metodiev@cs.ucdavis.edu, Telephone: 1 530 752 1417

We have developed an extensive set of simulation tools that will allow us to investigate and design a large-scale architecture based on this layout. Our tool chain provides individual system failure rates, the latency of a given quantum circuit, and the specific geometric characteristics and constraints of the system. In addition we have extended the qFPGA physical layout to describe a reconfigurable high-level quantum computer composed of an array of logical qubits interconnected with tightly integrated programmable teleportation channels.

The paper is organized as follows: in Section 2 we introduce the underlying technology, ion-traps, used to build the qFPGA structure. The following Section 3 gives an overview of the programmable quantum computer at the low and high-level. The construction of encoded logical qubits, the main building blocks of the computer, is described in Section 4 along with the time for a single cycle of error correction. Section 5 describes in more detail the logical interconnect between the encoded qubits. Finally we demonstrate a simple execution cycle of our computer by implementing a fault-tolerant construction of the Toffoli gate in Section 6, and we conclude our discussion in Section 7.

2. ION-TRAP TECHNOLOGY OVERVIEW

The technology used to implement a quantum information processor must adhere to three important requirements⁶: 1) It must allow the initialization of the n -qubit quantum system to the known state of $|000\dots0\rangle$. 2) A universal set of unitary quantum operations must be available to manipulate the initialized system and bring it to a desired entangled state. 3) It must have the ability to reliably measure the quantum system. The second requirement encompasses multi-qubit operations, thus it implies that a quantum architecture must allow for sufficient and reliable communication between physical qubits.

These seemingly contradicting requirements rule out most known technologies. On one hand qubits are asked to behave as quantum memories for long-term storage and reliable operations, and on another hand they must behave and transmitters of quantum information. All this without interacting with the environment to avoid decoherence. Right away silicon transistors are ruled out, since the stability of information in silicon is controlled through the constant monitoring by the environment. Quantum optics and atomic physics offer the most attractive candidates for the realization of quantum information processors.⁷

The most promising candidate technology are laser cooled trapped atomic ions,^{1-3, 7} with recent experimental progress demonstrating all necessary requirements for a quantum architecture.^{2, 8-16} Initially proposed by Cirac and Zoller¹ the qubits are formed from the electronic states of the ions, and were confined in a string of ions within a single trap. Even though the three major requirements for quantum computation have been experimentally demonstrated on such a trap, manipulating large number of ions in a single trap is not possible, thus limiting the scalability of this proposal.⁸ By limiting the ions to less than 10 per trap it is possible to achieve arbitrary scaling by either shuttling the ions in a Quantum Charge-Coupled Device (*QCCD*) style multiplexed traps,² or connecting remote ions through common photon interaction.⁶ We will focus on the interconnected *QCCD* trap array as the underlying layout for our computer.

2.1. Physical Ion-Trap Architecture Model

The *QCCD* model of an ion-trap quantum computing architecture consists of interconnected arrays of linear r.f. Paul traps, and can be abstracted as a 2-dimensional grid of plaquette cells, where a cell can contain an ion, a trapping electrode, or is empty and will serve the purpose of a channel for ballistic ion movement. The hyperfine electronic levels of a single ion (normally Be^+ , or Ca^+) are used to represent a qubit. These ion-qubits are confined in a trap whose *QCCD* abstraction model is shown in Figure 1, where a simpler version of such a trap has been initially demonstrated in Ref. [8] and most recently in References [15] and [16]. Ions are trapped by a sequence of control electrodes, where the physical location of each ion are defined by the trapping electrodes. By applying potentials and changing the electrode voltages ions are split from one-another and ballistically shuttled from trap to trap.

A high-level schematic of our ion-trap quantum information processor is shown in Figure 2(b) of Section 3. The figure shows a sea of logical qubits separated by long range communication channels as described in.¹⁷ The quantum ion-trap processor is surrounded by *classical control processors*. Each logical qubit is a sea of physical ions controlled by sequences of *laser pulses*. Our simulation tool has modules that generate pulse sequence files which are then executed on the general quantum architecture simulator machine. For scalability, an actual ion-trap system could manipulate qubits by focusing a small number of lasers through a MEMS mirror array as used in optical routers.¹⁸ The optimization of our algorithms to use the smallest number of lasers, essentially making them more SIMD, is a subject of future research. A tool chain to generate such optimized schedules is also an open area. Our focus is the design of the microarchitecture and its evaluation through hand-optimized applications.

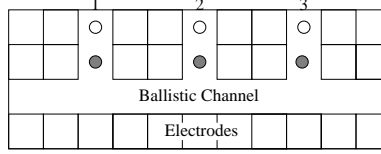


Figure 1. A schematic of the proposed *QCCD* model for trapped ions (circles). Each ion is trapped between voltage electrodes (cells), which define the ion’s physical location. A laser is used to physically address and manipulate the quantum information stored in the trapped ions. Ions are moved by changing the voltages from cell to cell within the ballistic channel. The cooling ions (clear circles) are used to sympathetically cool unwanted motional heating in the data ions (solid circles).

2.1.1. Gates and Measurement

Laser beams are used to implement arbitrary single qubit gates of any rotation^{13,19} $R(\theta, \phi) = (\cos(\theta/2)I + i \sin(\theta/2) \cos(\phi)\sigma_x + i \sin(\theta/2) \sin(\phi)\sigma_y)$, where I is the identity operator, $\sigma_x, \sigma_y, \sigma_z$ denote the Pauli spin matrices, θ is the angle or rotation and is proportional to the pulse duration, and ϕ denotes the relative phase between the lasers and the ion position. For simplicity we assume single-qubit gates perform with the same failure rate and gate duration. The experimental failure rate is $p_f \approx 10^{-4}$ and the execution time is $1\mu s$.

Universal quantum computation is completed with the ability to perform two-qubit gates, which are physically implemented with several laser pulses on each of the two ion-qubits. For example a controlled-phase gate that changes the sign of the qubit’s phase if the control qubit is $|1\rangle$ consists of three laser pulses.¹³ In reality two-qubit gates in ion-traps are composed of controlled- Z gates. The implementation of *CNOT* gate consists of two Hadamard gates and a controlled- Z . As with the one qubit gate, we have made the simplification that all two qubit gates have the same duration and failure rate. For two qubit gates the execution time is on the order of $10\mu s$ and the failure probability is about 0.03.

Individual ions are measured through state-dependent resonance fluorescent readout, where $|1\rangle$ fluoresces weakly and $|0\rangle$ very strongly.²⁰ A readout laser is used to perform the measurement. Measurement is a rather lengthy process that takes around $100\mu s$ and fails about 1% of the time.

2.1.2. Quantum Communication via Ballistic Transport

The physical displacement of the ion-qubit through changing voltage potentials across an empty channel is termed as *ballistic transport*. In one experiment,⁸ a single ion was transferred 10^6 times between two trapping electrodes separated by $1.2mm$ while preserving coherence. Each transfer process took $\approx 50\mu s$ with a $\approx 50\mu s$ wait between transfers. We model the rate at which the qubit dephases during transport as a failure probability of ~ 0.005 per micron moved. The main cause of decoherence during movement is not movement itself, but the exposure of the ion-qubit to fluctuating background fields as it moves.

Another cause of decoherence is the *heat* acquired by the ions as they move (i.e. their vibrational motion intensifies), thus ions must be periodically cooled as they travel. To avoid changing the state of the qubit contained in the ion by direct application of a cooling laser, the ions must be cooled *sympathetically*^{21,22} through the use of another ion designated for sympathetic cooling and normally a different atomic species (e.g. Mg^+). If specific individual ions in a large linear ion chain need to be moved, a potential wedge must be created to separate those ions from the rest of the chain. Chain splitting heats all of the ions involved and takes a comparably large amount of time (see Table 1).

Recall, the abstraction of an ion channel as a sequence of empty cells surrounded by electrode cells. Previous work¹⁷ has studied in detail quantum channels which consist of swapping the information from qubit to qubit. The ion-trap case is equivalent if we think of the information being moved cell by cell. The latency is proportional to the number of cells traversed. If D is the number of cells and T is the time to go from cell to cell, then the total time of the trip is $(\tau + (T \times D))$, where τ is the initial cost of starting a move instruction. The initial cost τ is due to the split (10μ seconds) required from the chain the ion is a part of. Considering a trap of $20\mu m$ as suggested in Ref. 23 a single trap can be traversed with a time cost of $T = 0.01\mu s$. The independence of the voltages cells from one another allows the ions to move in parallel, thus pipelining a single channel, we get a bandwidth of $\approx 100M$ qbps (qubits per second).

Operation	Time	Exp. Failure	Ideal Failure
Single Gate	$1\mu s$	0.0001	10^{-8}
Double Gate	$10\mu s$	0.03	10^{-7}
Measure	$100\mu s$	0.01	10^{-8}
Movement	$10ns/\mu m$	$0.005/\mu m$	$10^{-6}/cell$
Split	$10\mu s$		
Cooling	$1\mu s$		
Memory time	10 – 100 sec		

Table 1. Summary of the experimentally determined physical parameters for ion-traps. The current gate parameters are taken from¹⁴ and¹³ and are based on ${}^9Be^+$ ions cooled with ${}^{24}Mg^+$ ions. The times are the ideal ion-trap times based on 'best-possible' experimental results for the technology as are the ideal failure rates motivated by References^{7, 23}

2.1.3. Expected Ion-Trap Parameters

The current challenges with the ion-trap technology are technical,⁷ such as electrode surface integrity, and the structure of the substrate. The fourth column of Table 1 shows the expected ideal future parameters for the technology. Movement errors can be substantially reduced by improving the trap electrode surface integrity.⁸ The quality of the trap surface also directly affects movement and gate speed, since its improvement would substantially reduce motional heating. Using semiconductor materials for the trap implementation has been proposed^{2, 24} which will significantly improve the electrode surfaces.

Another major challenge to a large scale processor is the ability to multiplex ion-traps (e.g. crossing and turning corners).² This has not been experimentally verified, but conceptually turning a corner is not so different than splitting two adjacent ions by controlling the voltage in the traps, which has been done in the experiments. Once again the challenges are technical. We assume that the trap separation is $\approx 20\mu m$, with a turning speed equivalent to splitting of $10\mu s$. Our qFPGA structure is designed in such a way that no single gate at any level of recursion will require more than two turns when we are using direct ballistic communication, and no turns at all when we are using teleportation. Our analysis of the quantum system throughout this paper will assume the ideal expected ion-trap parameters in the fourth column of Table 1.

3. THE QUANTUM FPGA: A RECONFIGURABLE QUANTUM COMPUTER DESIGN

In this section we describe our proposed design for a configurable general purpose quantum computer based on the ion-trap technology. The need to physically transport quantum information from one location to another for multi-qubit gates together with the highly volatile data and high gate failure rates, forces quantum architecture designers to consider only highly optimal datapath oriented implementations. For this reason we propose a quantum FPGA (qFPGA) structure that allows us to leverage the QCCD technology and build a truly scalable system that supports efficient communication and error correction. An FPGA-like structure is a natural choice since it will allow us to explore the fine-grained parallelism existing in quantum applications and the *QCCD* layout.

Much as traditional FPGAs invest area in reconfigurability and interconnect, the qFPGA invests area in communication channels to allow movement of ions without impeding the rest of the computation and hindering the parallelism inherent in the ion-trap technology. We structured the qFPGA in such a way that the physical ion-qubits can communicate entirely in parallel with the surrounding blocks in all four cardinal directions without interfering with other qubits.

The qFPGA is further optimized for error correction, by far the most dominant and basic operation in a quantum machine. The qFPGA fits naturally to quantum error correction because the structure of the building blocks reflects the base level-one error-correction algorithm used, where each block represents a single level-one logical qubit. For simplicity, Figure 2(a) is drawn for a 3-bit error correcting code, but the structure is easily extended to 7-bit and larger codes (by 7-bit we mean an encoding of 7 lower level qubits into one logical qubit). A good example is the integration of the 7-bit circuit in Ref. [25] with the compact structure of the level 2 logical qubit (Figure 3) supported by its level 2 ancilla. Note also that the investment in communication channels also avoids excess heating of ions (causing errors) by limiting the number of turns an ion must take. Any two qubit gate at level k encoding requires at most 2 turns per physical ion in each direction. The configurability and adaptability of an FPGA design to the application allows logical qubits to be structured such that they fully comply with the fault-tolerant error correction requirements in References [5] and [26].

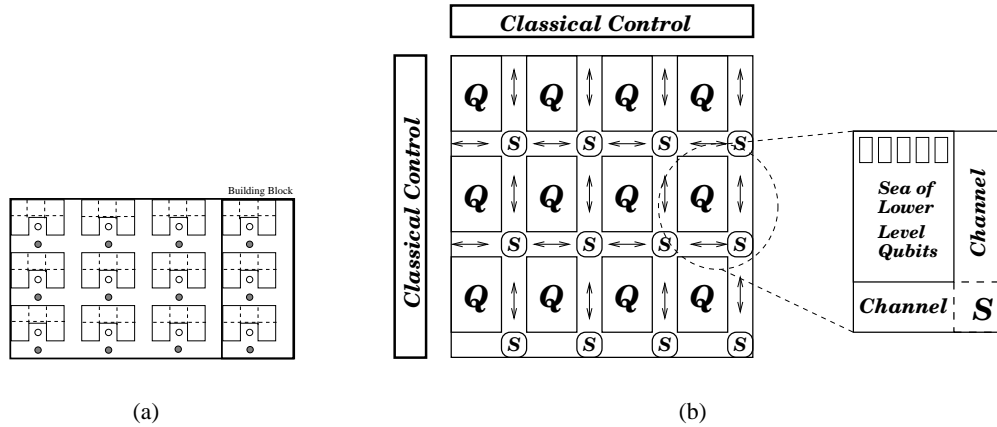


Figure 2. (a) The building blocks of the qFPGA layout. The figure consists of 4 level 1 building blocks, where the far right side outlines a single block. The circles are ions with data (solid) and cooling (clear). (b) High-Level quantum computer structure, where a hypothetical computer might be an interconnected logical qubits with fully reconfigurable communication mechanism. The letters S denote an integrated switch islands for redirecting quantum data.

3.1. High-Level Computer Architecture

At the application level a working quantum computer will exhibit much coarser granularity than physical ion-qubits. The single data bits will be encoded logical qubits comprising of many physical ion-qubits used for recursive error correction. In fact, a quantum computer will spend the majority of its computation within the logical qubits performing error correction. Still, the complexity and size of error-coded logical qubits together with the space required for qubit movement will result in communication between logical qubits as a primary challenge in building scalable systems. Once again a configurable data oriented logical level quantum computer design is desirable. A schematic of such a high-level quantum computer is shown in Figure 2(b). The computer is comprised of an interconnected array of logical qubits denoted with the letter Q , where each qubit is encoded with a sea of lower level qubits. Unrolling the levels of encoding we can see that each logical qubit is nothing more than a enormous sea of level 1 qFPGA building blocks each with a function designated by the corresponding error correcting code.

The logical qubits are connected with a tightly integrated switched (S) interconnect as shown in Figure 2(b). This makes the logical design very similar to the classical *RAW* architecture developed at MIT, however the key differences are that the communication paths are highly faulty and composed of the same qFPGA building blocks as the logical qubits. The integrated switches denoted with the letter S will be examined in more detail in Section 5, where we will call them *teleportation islands* that redirect traffic in the 4 cardinal directions. The communication channels between each logical qubit are driven by the concept of teleportation to avoid the high movement costs associated with transporting physical data ions directly.

4. DESIGNING THE LOGICAL QUBIT

Although, every component of the system is necessary, the most important and complex part of the architecture is the quantum bit itself. Due to the high error rates on single physical qubits, quantum logic will be dictated by encoding many physical qubits into a single logical qubit, which can further be encoded with other logical qubits to form a final large qubit. DAVE's bits of data can be considered separate processing units which are exactly such logical qubits. Each logical qubit must be equipped with the tools to undergo constant state stabilization in an efficient fault-tolerant manner without interfering with the rest of the computation. Choosing the correct error correction algorithm and its implementation directly impacts the performance of our entire system.

For the description of the logical qubit we chose the Steane $[[7, 1, 3]]$ code²⁷ as the error correction algorithm. The reason for this choice is that this is the smallest encoding which has known and relatively easy fault-tolerant logical operations. Ideally the qFPGA will allow us to dynamically generate a logical qubit layout based on the input set of

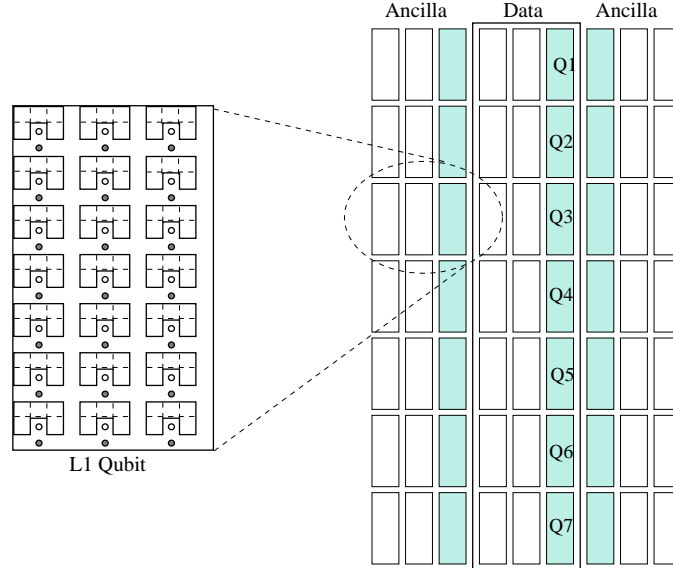


Figure 3. The Logical Qubit: 7 groups of 3 level 1 blocks make a single level 2 logical qubit (middle). The two identical conglomerations on the sides are ancillary blocks used for error correction. The shaded boxes are the data level 1 blocks. A zoomed in view of a level 1 qubit is shown on the left. Only 7 ions for a level 1 encoded qubit, however it must have two additional 7 ion blocks for error correction.

operations, however here we demonstrate a hand optimized design using the flexibility of the qFPGA. We design the qubit with the ideal ion-trap parameters from Table 1. This provides us with the benefit of being far below a certain threshold of failure required to sustain arbitrary long computations,^{5, 28} which is on the order of 10^{-5} for the $[[7, 1, 3]]$ code.²⁶ Another key assumption is that the ion-trap lifetime is much higher (≈ 10 sec) than any operation we can make, so we do not take lifetime into our design, and thus able to explore smaller system size versus maximum parallelism. A detailed analysis of the $[[7, 1, 3]]$ error correction algorithm can be found in.^{25, 29, 30}

We find that the qFPGA offers an efficient implementation of some of the best known theoretical fault-tolerant, recursive constructions for error correction.⁵ Figure 3 shows the full implementation of a level 2 qubit. Our choice of the Steane $[[7, 1, 3]]$ code means that a single data logical qubit is built by stacking 7 level 1 blocks. However, each level 1 block must be error corrected at level 1, so to each one we attach two more blocks used as ancilla. To add level 2 error correction we add two more identical ancilla structures at level 2 on both sides of the data logical block. Level 1 blocks labeled as $Q1 - Q7$ denote the data blocks that are used as a single logical qubit. Everything else is ancilla used for error correction.

The error correction algorithm consists of extracting a syndrome for X and Z errors and applying a correction operation based on the syndrome. The syndrome extraction process consists encoding a block of ancilla at the same level as the data and interacting the ancilla and the data. The number of ancilla blocks we have affects the parallelism we can explore when extracting syndromes for the two types of error. For example, the L1 qubit shown on the left of 3 uses 7 ions as data and 7 ions as ancilla, the other 7 are used as verification bits for encoding. Thus it must extract the two syndromes one after the other. At L2 however we have ancilla conglomerations on both sides of the data block (See Figure) and we can prepare the ancilla blocks in parallel and extract the syndromes in parallel.

4.1. Level 2 Error Correction Latency

Let us first estimate the time required for each error correction step assuming the ideal ion-trap parameters from table 1. The error correction cycle divides into three main portions:

1. Syndrome extraction: a process which includes the preparation of the ancilla qubits to the required logical initial state denoted as $|0\rangle_L$. Two sets of ancilla are needed to extract an error syndrome for both X and Z errors in parallel. If the first syndrome extraction shows a non-trivial syndrome (e.g. error exists) we repeat the process until we reach two successive agreeing syndromes.

2. Correcting the error with the appropriate gate followed by a level 1 error correction cycle.
3. Lower level error correction at each logical gate. Single qubit gates are transversal for the Steane $[[7, 1, 3]]$ code, thus we assume that the correcting process takes only the time for one single qubit gate.

$$T_{L,ecc} = \begin{cases} 2 \times T_{L,synd}, & \text{Trivial syndrome} \\ 2(2T_{L,synd} + T_1 + T_{L-1,ecc}), & \text{Non-trivial} \end{cases} \quad (1)$$

Where $T_{L,synd} = (T_{L,prep} + 4T_{L-1,ecc} + T_1 + T_2 + T_m + 40(D_L) + 2\tau)$ is the time to extract the syndrome at level L , and $T_{L-1,ecc}$ is the time to error correct at level $(L - 1)$. $T_{L,prep}$, D_L , T_2 , T_1 , T_m , and τ are respectively the times for preparing level k ancilla, inter-block distance at level L , two-qubit gates, single-qubit gates, measurement, and split times. The inter qubit distance scales exponentially with level L as $O(D_1^{(\frac{L}{2}+1)})$, where D_1 is maximum 21 cells and denotes movement from one level 1 block to the next.

Numerical simulations of a level 2 qubit showed that a non-trivial syndrome was measured for level one with a rate of $3.35 \times 10^{-4} \pm 4.1 \times 10^{-5}$, and for level two at a rate of $7.917 \times 10^{-4} \pm 8.1 \times 10^{-5}$. Taking a weighted average of the two cases in Equation 1 we determine a level 2 error correction time of approximately 0.043 seconds, where 0.08 seconds is spent in preparation of the logical ancilla. This estimate is somewhat conservative since Equation 1 assumes that the X and Z syndromes are extracted in serial instead of parallel.

4.2. Logical Qubit Size and Recursion Level

The level of recursion for each logical qubit is the most crucial assumption for both the performance and the size of our system. Since we perform error correction at each gate of the application the total time for each gate rises exponentially with the level. The size of our logical qubit also rises exponentially since at each level we increase the qubits by a factor equal to the encoding size.

A quantum computer of size $S = KQ$, where K is the number of time-steps and Q is the number of logical qubits, will need to have a component failure rate of at most $P_f = 1/KQ$. To evaluate the expected component failure rate at some level or recursion we use Gottesman's estimate for local architectures⁵ shown in Equation 2 below.

$$P_f = \frac{1}{cr^2r^k} (cr^2p_0)^{2^k} = \frac{p_{th}}{r^k} (p_{th}^{-1}p_0)^{2^k} \quad (2)$$

The value for r is the communication distance between level 1 blocks which are aligned in the qFPGA in a way to allow $r = 12$ cells on average. The threshold failure rate, p_{th} , for the Steane $[[7, 1, 3]]$ circuit accounting for movement, gates, and waiting time errors was computed in²⁶ to be 7.5×10^{-5} . Taking as p_0 the average of the ideal failure probabilities given in Table 1, and plugging these numbers into Equation 2, we get an estimated level 2 failure rate of 1.02×10^{-16} . This gives a computer of size $S = KQ = 9.89 \times 10^{15}$. As a simple example, we can consider Shor's factoring algorithm for a 1024-bit number. Borrowing a circuit description optimized for latency from Reference 31, we will need a computer of roughly $S = 4.36 \times 10^{12}$, which is a few orders of magnitude below the size attainable with level 2 recursion.

5. INTERCONNECT DESIGN

Along with low logical gate error rates, the execution of any algorithm will depend of the efficient interaction among logical qubits. The dimensions of a single logical qubit are 147 by 36 cells, and thus too far for each data ion to be physically transported especially from one end of the chip to the other. The concept of *quantum teleportation*³² is utilized within each processor to provide high-speed, low-latency, fault-tolerant network interconnection between the logical qubits.

Teleportation begins by preparing two maximally entangled qubits A and B in an Einstein-Podolsky-Rosen (EPR) state³³: $|\Psi\rangle = |0_A0_B\rangle + |1_A1_B\rangle$ using a Hadamard and a controlled-*NOT* gate. Qubit A is sent to the location of the source qubit C and qubit B to the destination of location C . Entangling A and C and measuring them allows us to recreate the state of C over the destination qubit B . We have effectively teleported C 's state over a very large distance without having to move it directly. The drawback is that we are still physically moving the entangled qubits A and B , however EPR pairs are replaceable and with enough resources we can establish entanglement between the source and the destination just

in time for the communication to be completed. Experimental demonstration of the teleportation protocol with ion-traps has been done most recently in References [16] and [15].

The assumptions for our interconnect design decisions are: 1) The transport protocol must be powerful enough to allow reliable communication of logical ions at arbitrary distances from one another without significant resource increase, and within the error correction time of a single qubit. Precommunication becomes critical. 2) We assume ideal ion-trap parameters. Trap size is $20\mu\text{m}$ with minimal ion heating due to motion, and state of the art surfaces.^{23,24} Time to move from trap to trap is $0.01\mu\text{s}$ per micron. 3) Due to many unknowns about gate fidelity as a function of the motional ion-heating we are assuming robust cooling with time of $10\mu\text{s}$ and performed every 10 cells. And finally, 4) Although there are many optimized entanglement purification schemes we have purposely chosen to study the original protocol proposed by Bennet in.³⁴ We show that it is not only sufficient for our needs, but it demonstrates that the system can only be improved from here on.

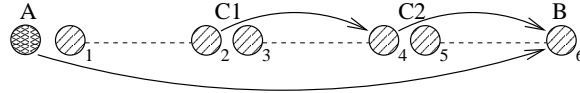


Figure 4. A schematic of a single bandwidth channel between the logical qubits of our ion-trap processor. Rather than distributing an EPR pair of qubits directly from the source to the destination, the channel is divided into islands (here denoted as $C1$, and $C2$), that can be used to expand the entanglement over the entire channel.

The EPR transport problem can be solved by combining the concepts of *quantum repeaters*³⁵ and *entanglement purification*.³⁴⁻³⁸ The quantum repeaters are islands that are strategically placed in the channels between the logical qubits to limit the distance traveled by each EPR pair. EPR pairs only travel to two neighboring islands, whose entanglement can then be efficiently purified using the purification protocols with some additional ancillary EPR pairs. Figure 4 demonstrates the concept of quantum repeaters by showing a simple schematic of the channel between the logical qubits. The channel shown has a bandwidth of 1 physical bit at a time, and the ancillary EPR qubits are pipelined to the two opposing islands as they purify the data EPR pairs. The arrows in Figure 4 show how teleporting each individual pair, entanglement can be extended to be between the source and the destination before the final teleportation stage. The following steps outline the logical qubit communication protocol in more detail:

- The channel is divided into N segments represented by a sequence of EPR pairs. In Figure 4 $N = 3$ and the segments are described by the EPR pairs (1,2), (3,4), and (5,6). Each member of the EPR pairs is ballistically transported to the opposite ends of their segments to a Teleportation Island (C_i).
- After the EPR pairs are purified we are ready to begin teleporting. Recall that the idea is to create a single EPR pair between locations A and B .
- First we teleport qubit 2 to qubit 4. This action has now created an EPR pair between 1 and 4 by transferring 2's entanglement with 1 to 4.
- Then we further teleport 4 to 6, transferring the entanglement to 1 and 6. Now we have created a single EPR pair (1,6) between connecting locations A and B .
- We then proceed to teleport qubit A to location B using the new EPR pair.

We can now describe in better detail the logical qubit interconnect shown in Figure 2(b), the regions between the qubits will be defined by the *communication channels*, namely the teleportation segments described above. Each single communication channel will connect two teleportation islands which lie at the corners of four logical qubits, denoted with the letter S in the figure. The size of each logical qubit is 147×36 cells, the width of the channels is 11 and 6 cells for the \hat{y} and the \hat{x} channel respectively.

The teleportation islands are equipped with the capability of being used or not being used. This allows a communication scheduler to pick the optimal inter-island separation for the total distance traveled. Borrowing and adapting the recursive fidelity equations (9,19) given in³⁵ for the Bennett purification protocol,³⁴ and limiting purification to be only between two adjacent islands we determine optimal separation between two islands to be about 100 cells at distances less than 7000 cells and about 350 cells at greater distances. In the \hat{x} direction this amounts to an island at every third and tenth logical qubit respectively. In the \hat{y} direction, however, we place an island at every logical qubit due to the fact that a logical qubit is 147 cells in this direction. The total connection time can be determined by analyzing enough purification steps to avoid purification of the final EPR pair between the source and the destination.

Let us now explain the dimensions of the inter-island segments in Figure 2(b). To optimize space and performance we modeled the channels between each island as a two-way ballistic transport region. Figure 5 shows a very stripped down schematic of the horizontal channel between two islands. Each EPR pair is created in the middle and separated to the two opposing ends. One pair is designated as the data EPR and is continually purified in round-robin pipeline fashion. We assume to have enough ions to handle the maximum amount of needed purification steps without having to wait for the creation of new EPR pairs.

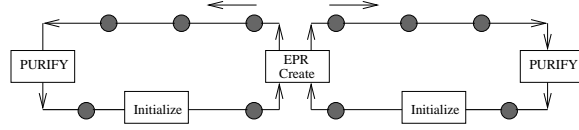


Figure 5. A very stripped down schematic of the horizontal channels between two islands.

Since each horizontal channel can purify one EPR pair at a time, the bandwidth of the entire communication is defined by the number of horizontal (\hat{x}) channels involved, each having a width of 6 cells. The construction of the vertical channels is equivalent but they have the advantage of being in a parallel line with the logical qubits and can thus purify at once as many as the entire total number of 49 physical ion-qubits that make up each logical qubit. This added advantage considerably reduces our system size as the area only grows for the \hat{x} channels as we increase bandwidth.

6. EXAMPLE: FAULT-TOLERANT TOFFOLI GATE

Having described the general structure of the quantum computer we will demonstrate a simple example: namely the 3-qubit Toffoli gate, which is a *controlled-controlled-NOT* gate. For three qubits X , Y , and Z by choosing X and Y to be the control qubits and Z the target, the Toffoli gate will output $\{X, Y, (Z \oplus XY)\}$. It appears that the Toffoli gate would require only three logical qubits, however fault-tolerant design requires that a single error cannot spread to more than one additional qubit in any quantum operation. Thus the Toffoli gate is not fault-tolerant. The execution of a fault-tolerant Toffoli gate can be performed with a set of smaller one and two-qubit gates with the addition of 6 ancilla qubits. The circuit for a single fault-tolerant Toffoli is shown in Figure 6, where the top portion is the preparation of the ancilla qubits A_0 , A_1 , and A_2 with the use of the three other ancilla qubits C_0 , C_1 , and C_2 . The circuit can be constructed by the guidelines of References [39], and [40]. We will omit the preparation of each individual logical qubit from our analysis.

We choose the construction of a fault-tolerant Toffoli gate because it's circuit offers large enough sequence of quantum operations to encompass an interesting execution cycle and small enough to be followed easily by the reader. Moreover, the Toffoli gate is central in many quantum algorithms including Shor's algorithm for factoring large numbers.

The critical component for the success of the whole design is the cost of communication between logical qubits. We have made a design decision that ballistic transport must be used for moving ions within a logical qubit, and teleportation will be preferred when moving across larger distances. The teleportation protocol analyzed maintains constant cost in the face of increasing distance and hence is a critical weapon in our armory. Since EPR pairs are required for teleportation, we can reduce communication costs to a minimum if we have the required number of EPR pairs available at a logical qubit at the same time that it is ready to move. Fortunately, this is possible because of the high cost of error correcting the logical qubits. We can create, purify and transport the required EPR pairs to their respective qubits while they are undergoing error correction. But can this be done at a large scale?

To answer this question, we developed a tool to schedule the movement of EPR pairs in the qFPGA. We assigned one channel to carry the created EPR pairs to their destinations and another channel to return the used EPR pairs. Within each channel, the EPR pairs are pipelined. We define the bandwidth of the qFPGA as the number of channels in each direction. The distance between each Teleportation Island was fixed at 100 qFPGA cells. The goal of our scheduler then, is to find paths between logical qubits to transport all the required EPR pairs within the time it takes to perform a level 2 error correction.

The scheduler is heuristic greedy scheduler that scalably achieves an average of $\sim 23\%$ aggregate bandwidth utilization (see Figure 7) on our implementation of the Toffoli gate. It works by grabbing all available bandwidth whenever it can. However, if this means that the scheduler cannot find the necessary paths, it will back off and retry with a different set of start and end points. A simple approach to doing a two qubit gate between logical qubits A and B would be as follows:

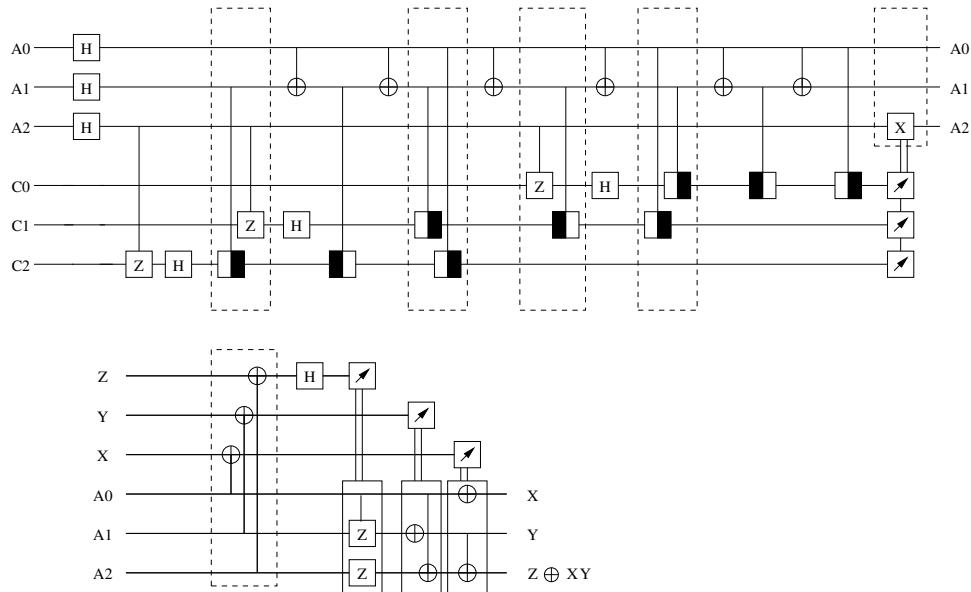


Figure 6. A map from a standard circuit to a fault tolerant layout for a single toffoli gate. The logical qubits X , Y , and Z are the participants in the gate and the rest of the qubits are the ancilla required for the fault tolerant construction. The top portion of the figure shows the ancilla preparation procedure. Each dashed box represents a time-cycle (i.e. a set of gates that can be performed in parallel).

teleport A to B's physical location, perform the gate and teleport it back. An optimization that the scheduler incorporates is that it only moves logical qubit A back if necessary. As a result, the logical qubits *drift* from one location to another. This adds a level of complexity to the scheduler, but at the same time reduces the amount of movement that the qubits are subjected to.

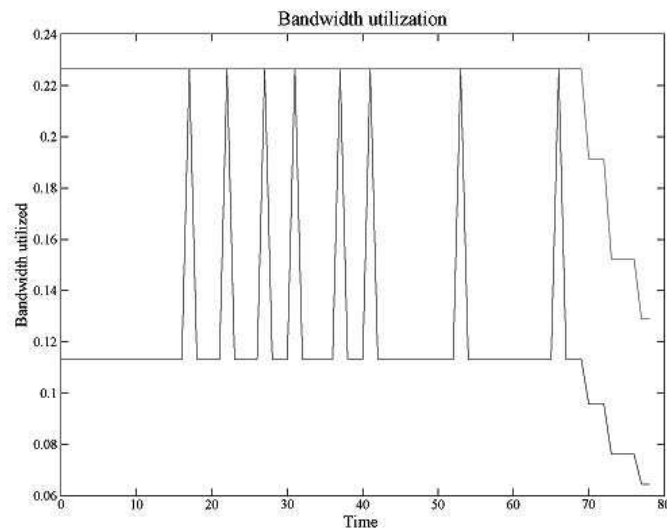


Figure 7. The bandwidth utilization while executing the Toffoli gate.

With all the above considerations in the scheduler, we found that given two channels in each directions (bandwidth of 2), we could schedule communication such that it always overlapped with error correction of the logical qubits. The end result being reliable movement over arbitrary distances with minimal overhead, where the cost of the circuit depends on the

frequency of error correction cycles for the logical qubits. Each Toffoli will contribute approximately 16 error correction steps for the ancilla preparation repeated three times, and 5 error correction cycles to finish the gate (bottom portion of Figure 6). A single time-step is defined by an error correction cycle since the qubits involved at each logical gate must be error corrected. Recall that it took roughly 0.043 seconds for an error correction steps, thus the latency of a single fault-tolerant Toffoli of 53 error correction cycles is roughly 2.3 seconds. Extending this a bit further: Factoring a 128-bit number requires 63730 Toffoli gates when we use the optimized logarithmic-depth quantum carry lookahead adder in Ref. 31, and.⁴¹ Without considering overlapping ancilla preparation from one Toffoli to the next, we can see that factoring a 128-bit number will take at least 40 hours.

7. CONCLUSION

By presenting a carefully constructed reconfigurable quantum circuit substrate tailored to ion-trap technology, we have managed to create an architecture that can implement any quantum algorithm and yet keep its geometry intact. This not only simplifies manufacturing, but vastly increases the possible applications of such a processor. In our analysis, we have incorporated the qFPGA as the basic building block for a system that considers not only fault-tolerant quantum error correction, but breaks down each component into a fault-tolerant counter part implementation.

Our qFPGA study is only a proof-of-concept, but it demonstrates the core mechanisms necessary to build a scalable system. Given the ideal ion-trap parameters from Section 2, our error-correction and communications infrastructure scales to more than 3×10^8 physical ions - enough to factor a 2048-bit number.³¹ Although the actual engineering of such a system will require physicists and engineers to overcome many technical (but not fundamental) challenges, we have provided a blue-print for the structure and microarchitecture of the system. The most important assumptions we have made in our time estimations are the complete availability of control lasers for the execution of the qubit gates in a parallelizable form, the ability to implement the control circuitry of the trap electrodes, and the omission of classical computation from the calculations. A natural extension to the proposed design is to insert these assumptions in our simulation tools and present a design that takes into account the many possible sources of engineering difficulties.

REFERENCES

1. J. I. Cirac and P. Zoller, "Quantum computations with cold trapped ions," *Phys. Rev. Lett.* **74**, pp. 4091–4094, 1995.
2. D. K. and C. Monroe and D. J. Wineland, "Architecture for a large-scale ion-trap quantum computer," *Nature* **417**, pp. 709–711, 2002.
3. J. V. P. et. al., "Quantum information with neutral atoms as qubits," *Phil. Trans. R. Soc. Lond.* **A361**, pp. 1417–1427, 2003.
4. M. Oskin, F. Chong, and I. Chuang, "A practical architecture for reliable quantum computers," *IEEE Computer* **January**, 2002.
5. D. Gottesman, "Fault-tolerant quantum computation with local gates," *arXiv e-print quant-ph/9903099*, 1999.
6. C. Monroe, "Quantum information processing with atoms and photons," *Nature* **416**, p. 238, 2002.
7. D. Wineland and T. Heinrichs, "Ion trap approaches to quantum information processing and quantum computing," *A Quantum Information Science and Technology Roadmap*, p. <http://quist.lanl.gov>, 2004.
8. M. A. Rowe and et. al., "Transport of quantum states and separation of ions in a dual rf ion trap," *Quant. Inf. Comp.* **2**, pp. 257–271, 2002. ePrint [quant-ph/0205094](http://arxiv.org/abs/quant-ph/0205094).
9. D. Kielpinski, "Entanglement and decoherence in a trapped-ion quantum register," *Thesis, Univ. Colorado*, 2001.
10. C. M. et. al., "Scalable entanglement of trapped ions," *AIP Conf.* **551**, pp. 173–186, 2001.
11. C. Monroe, D. Meekhof, B. King, W. Itano, and D. Wineland, "Demonstration of a fundamental quantum logic gate," *Phys. Rev. Lett.* **75**, p. 4714, 1995.
12. D. Meekhof, C. Monroe, B. King, W. Itano, and D. Wineland, "Generation of nonclassical motional states of a trapped atom," *Phys. Rev. Lett.* **76**, p. 1796, 1996.
13. D. L. et al., "Experimental demonstration of a robust, high-fidelity geometric two ion-qubit phase gate," *Nature* **422**, pp. 412–415, 2003.
14. D. J. W. et al., "Experimental issues in coherent quantum-state manipulation of trapped atomic ions," *Journal of Research of the National Institute of Standards and Technology* **103**, pp. 259–328, 1998.

15. M. Riebe, H. Haffner, C. Roos, W. Hansel, J. Benhelm, G. Lancaster, T. Korber, C. Becher, F. Schmidt-Kaler, D. James, and R. Blatt, "Deterministic quantum teleportation with atoms," *Nature* **429**(6993), pp. 734–737, 2004.
16. M. B. 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.
17. M. Oskin and et al., "Building quantum wires: The long and the short of it," *In the 30th Annual International Symposium on Computer Architecture*, 2003.
18. D. J. Bishop, C. R. Giles, and G. P. Austin, "Lucent LambdaRouter: MEMS technology of the future here today," *IEEE Commun. Mag.* **40**, pp. 75–79, March 2002.
19. D. Wineland and et al., "Quantum information processing with trapped ions," *Phil. Trans. R. Soc. Lond. A* **361**, pp. 1349–1361, 2003.
20. E. Hahn, "Spin echoes," *Phys. Rev.* **80**, pp. 580–594, 1950.
21. M. Barrett, B. DeMarco, T. Schaetz, D. Leibfried, J. Britton, J. Chiaverini, W. Itano, B. Jelenkovic, J. Jost, C. Langer, T. Rosenband, and D. Wineland, "Sympathetic cooling of $^9\text{Be}^+$ and $^{24}\text{Mg}^+$ for quantum logic," *Phys. Rev. A.* **68**(4), p. 42302, 2003. ePrint quant-ph/0307088.
22. D. Kielpinski, B. King, C. Myatt, C. Sackett, Q. Turchette, W. Itano, and D. Wineland, "Sympathetic cooling of trapped ions for quantum logic," *Phys. Rev. A.* **61**(3), p. 32310, 2000.
23. A. M. Steane, "How to build a 300 bit, 1 gop quantum computer," *quant-ph/0412165*, 2004.
24. M. Madsen, W. Hensinger, D. Stick, J.A.Rabchuk, and C. Monroe, "Planar ion-trap geometry for microfabrication," *Applied Physics B-Lasers and Optics* **78**, pp. 639–651, 2004.
25. A. M. Steane, "Space, time, parallelism and noise requirements for reliable quantum computing," *Fortsch.Phys.* **46**, pp. 443–458, 1998.
26. K. Svore, B. Terhal, and D. DiVincenzo, "Local fault-tolerant quantum computation," *preprint quant-ph/0410047*, 2004.
27. A. Steane, "Error correcting codes in quantum theory," *Phys. Rev. Lett.* **77**, pp. 793–797, 1996.
28. D. Aharonov and M. Ben-Or, "Fault tolerant computation with constant error," in *Proceedings of the Twenty-Ninth Annual ACM Symposium on the Theory of Computing*, pp. 176–188, 1997.
29. A. M. Steane, "Overhead and noise threshold of fault-tolerant quantum error correction," *quant-ph/0207119*, 2002.
30. B. W. Reichardt, "Improved ancilla preparation scheme increases fault-tolerant threshold," *quant-ph/0406025*, 2004.
31. R. V. Meter and K. M. Itoh, "Fast quantum modular exponentiation," *E-Print: quant-ph/0408006*, 2004.
32. C. H. Bennett, G. Brassard, C. Crépeau, R. Jozsa, A. Peres, and W. Wootters, "Teleporting an unknown quantum state via dual classical and EPR channels," *Phys. Rev. Lett.* **70**, pp. 1895–1899, 1993.
33. J. S. Bell, "On the Einstein-Podolsky-Rosen paradox," *Physics* **1**, pp. 195–200, 1964. Reprinted in J. S. Bell, *Speakable and Unspeakable in Quantum Mechanics*, Cambridge University Press, Cambridge, 1987.
34. C. H. Bennett, G. Brassard, S. Popescu, B. Schumacher, J. A. Smolin, and W. K. Wootters, "Purification of noisy entanglement and faithful teleportation via noisy channels," *Phys. Rev. Lett.* **76**, p. 722, 1996. arXive e-print quant-ph/9511027.
35. W. Dur, H. J. Briegel, J. I. Cirac, and P. Zoller, "Quantum repeaters based on entanglement purification," *arXiv:quantu-ph/9808065*, 1998.
36. C. H. Bennett, D. P. DiVincenzo, J. A. Smolin, and W. K. Wootters, "Mixed state entanglement and quantum error correction," *Phys. Rev. A* **54**, p. 3824, 1996. arXive e-print quant-ph/9604024.
37. C. H. Bennett, D. P. DiVincenzo, W. K. Smolin, and W. K. Wootters, "Mixed-state entanglement and quantum error correction," *Phys. Rev. A* **54**, pp. 3824–3853, 1996.
38. D. Deutsch, A. Ekert, R. Jozsa, C. Macchiavello, S. Popescu, and A. Sanpera, "Quantum privacy amplification and the security of quantum cryptography over noisy channels," *Phys. Rev. Lett.* **77**, pp. 2818–2821, 1996.
39. M. A. Nielsen and I. L. Chuang, *Quantum Computation and Quantum Information*, Cambridge University Press, Cambridge, UK, 2000.
40. A. M. Steane, "Efficient fault-tolerant quantum computing," *Phys. Rev. Lett.* **78**, pp. 2252–2255, 1997.
41. T. Draper, S. Kutin, E. Rains, and K. Svore, "A logarithmic-depth quantum carry-lookahead adder," *E-Print: quant-ph/0406142*, 2004.