High-Level Interconnect Model for the Quantum Logic Array Architecture

TZVETAN S. METODI and DARSHAN D. THAKER
University of California, Davis
ANDREW W. CROSS and ISAAC L. CHUANG
Massachusetts Institute of Technology
and
FREDERIC T. CHONG
University of California, Santa Barbara

We summarize the main characteristics of the quantum logic array (QLA) architecture with a careful look at the key issues not described in the original conference publications: primarily, the teleportation-based logical interconnect. The design goal of the the quantum logic array architecture is to illustrate a model for a large-scale quantum architecture that solves the primary challenges of system-level reliability and data distribution over large distances. The QLA's logical interconnect design, which employs the quantum repeater protocol, is in principle capable of supporting the communication requirements for applications as large as the factoring of a 2048-bit number using Shor's quantum factoring algorithm. Our physical-level assumptions and architectural component validations are based on the trapped ion technology for implementing quantum computing.

Categories and Subject Descriptors: C.1.m [**Processor Architectures**]: Miscellaneous

General Terms: Design, Performance, Reliability

Additional Key Words and Phrases: Quantum computer architecture design, quantum, QLA, large scale, fault tolerance, teleportation

ACM Reference Format:

Metodi, T. S., Thaker, D. D., Cross, A. W., Chuang, I. L., and Chong, F. T. 2008. High-Level interconnect model for the quantum logic array architecture. ACM J. Emerg. Technol. Comput. Syst. 4, 1, Article 1 (March 2008), 28 pages. DOI = 10.1145/1330521.1330522 http://doi.acm.org/10.1145/1330521.1330522

Author's addresses: T. S. Metodi (corresponding author), D. D. Thaker, Computer Science Department, University of California at Davis, 2239 Kempber Hall, One Shields Ave., Davis, CA 995616; email: tsmetodiev@ucdavis.edu; A. W. Cross, I. L. Chuang, Department of Electrical Engineering and Computer Science, Massachusetts Institute of Technology, 77 Massachusetts Ave., Cambridge, MA 02139-4307; F. T. Chong, Department of Computer Science, University of California at Santa Barbara, Santa Barbara, CA 93106.

1. INTRODUCTION

The quantum logic array architecture (QLA) [Metodi et al. 2005; Thaker et al. 2006] is a homogeneous, tiled model for large-scale quantum computation, whose basic structure is an interconnected grid of computational tiles that describe a single fault-tolerant quantum bit, or *logical qubit*. A logical qubit is encoded into a number of physical qubits such that its underlying geometrical construction is intended to provide the necessary resources for quantum error correction: by far the most dominant operation for a quantum computer [Oskin et al. 2002]. The logical qubit tiles are connected through teleportation-based [Bennett et al. 1993] communication channels utilizing the concept of quantum repeaters [Dur et al. 1999] to overcome the long-distance data distribution constraints.

The progression of the QLA model, the compressed QLA (CQLA) architecture [Thaker et al. 2006], is a scalable quantum architecture design that employs software-based specialization of the QLA system into memory and computational regions. Each region is individually optimized to match hardware support to the available parallelism. One of the most interesting concepts in a future quantum-memory hierarchy exposed by the CQLA model is that the hierarchy does not have a technology-dependent memory wall that separates the memory and computational regions, but rather an *encoding wall* that separates specialized architectural regions based on the different error correction codes employed.

The design of the QLA architecture is based on the principle of *system-level balance* between fault-tolerant, logical qubit structures and communication mechanisms that protect the quantum data while in transmission. Logical qubit structures include the number of ancillary qubits necessary for the required rate of error correction. The bandwidth of the interconnect channels is balanced with the size and speed of the computational blocks that work on these logical qubits. The distribution of the quantum computational resources is matched to the application's support for gate teleportation or data teleportation, and thus allows for the creation of logical teleportation resources. The amount of quantum resource usage for different error correcting codes or levels of encoding is matched to the size of application and the needed reliability to complete application execution with the desirable success rate. While the performance of the QLA architecture is estimated at the high level of execution through Shor's quantum factoring algorithm [Shor 1995], the focus of our work is not on the specific mapping of applications, but rather on a flexible organization model that allows the mapping of arbitrary circuit-based quantum subroutines.

In this article we provide a better description of the logical qubit interconnect than our conference papers that introduce the QLA architecture [Metodi et al. 2005; Thaker et al. 2006]. We also show that the exponential slowdown due to error correction does not overcome the exponential speedup gained from quantum algorithms such as Shor's. By designing the QLA architecture such that it scales favorably within realistic technology parameters, we have not only provided a valid model for a future large-scale quantum machine, but also hope to have created the software infrastructure necessary to design and test quantum computing systems such that better device targets can be provided. Our simulation infrastructure consists of accurate low-level quantum simulation tools based on the stabilizer formalism [Gottesman 1998] combined with our own low-level circuit scheduling heuristics [Metodi et al. 2006]. This infrastructure allows us to reliably validate the performance of the error correcting codes involved, and hence to accurately study greater than 90% of the low-level computational resources.

We have not, however, focused on a detailed description of how to program the architecture. By making the design decision that each logical qubit is a self-contained processing tile providing the necessary quantum and classical resources for logical gates and error correction, we have completely *decoupled* computation from communication at the application level. This makes programming the architecture significantly easier, and is done entirely through classical compilation routines that translate the necessary quantum circuits into a timed sequence of the pulses executed over each logical qubit. We briefly discuss the programming model in Section 2.2.

The article is organized as follows: In Section 2, we describe the main scalability issues that we needed to consider when designing the QLA architecture. In Section 3 we provide a brief overview of the logical qubit structure and briefly estimate the overhead of the error correction code used, including our simulation results of the logical qubit reliability. Section 4 provides an analysis of the teleportation-based logical interconnect, and finally we conclude with a discussion and future work in Section 5.

2. QUANTUM LOGIC ARRAY ARCHITECTURE: HIGH-LEVEL OVERVIEW

One of the important lessons learned from designing a large-scale system is that there is a significant difference between physically implementing the components needed for a quantum computer, and designing a complete, large-scale quantum architecture that is intended to execute arbitrary computationally relevant programs. Based on the circuit model of computation [Deutsch 1985], we identified three main scalability issues for building a large-scale quantum architecture, as next described.

(1) *Availability of reliable and realistic implementation technology that adheres to the DiVincenzo requirements [DiVincenzo 2000] for implementing quantum computation.* This can be summarized as: (1) a quantum register described as a collection of well-defined single-qubit states, appropriately initialized; (2) a "universal" set of quantum operations, including reliable measurements; and (3) the ability to transmit quantum information.

(2) *Implementation of robust, fault-tolerant structures encoded using efficient error correction algorithms.* This requirement provides system-level fault tolerance that will allow the execution of an arbitrarily large sequence of universal quantum logic operations within the architecture decoherence time. In the circuit model of computation, arbitrary operation and state reliability is achieved through recursively encoding a number of lower-level qubits into a higher-level logical qubit in a fault-tolerant manner. The

geometrical design of a large-scale quantum architecture must be able to efficiently integrate the exponential cost of recursive error correction with the speedup offered by quantum algorithms over classical ones.

(3) *Efficient quantum resource distribution at both application level and physical qubit level that allows maximum overlap of computation and error correction.* Resource distribution is particularly important because the destructive nature of measurement makes the copying of a quantum state impossible [Wootters and Zurek 1982]. The implication is that if the data is needed elsewhere on the chip, the qubit or quantum information, contained in the physical implementation of a qubit, must be physically transported to the new destination while destroying it at the source.

2.1 Technology Assumptions

A number of physical experiments [Cirac and Zoller 1995; Cabrillo et al. 1999; Wineland et al. 1998; Knill et al. 2001; Kane 1998 Makhlin et al. 1999; Platzman and Dykman 1999; Hollenberg et al. 2003; Hime et al. 2007; Niskanen et al. 2007] have recently demonstrated small-scale quantum systems that show promise for scalable quantum information processing. However, to scale these systems to computational relevance, a large-scale quantum architecture must be able to orchestrate the quantum and classical control of tens of millions of qubits for the duration required by the program being executed. Although existing experiments have shown that the realization of practically useful quantum computers is no longer science fiction, the difficulty of deterministically controlling quantum data over long periods of time and the enormous complexity of the classical control mechanisms required for quantum control have so far been prohibitive in realizing large-scale systems.

To provide validated device targets, we model the QLA architecture with ion-trap technology assumptions at the lowest design level. The trapped-ion scheme proposed by Cirac and Zoller [1995] was the first work that described a clear model for physically implementing a complete quantum computer in the laboratory. Subsequently, recent experimental successes with ion-traps [Barrett et al. 2004; Riebe et al. 2004; Langer et al. 2005; Reichle et al. 2006] have physically demonstrated every architectural component required to implement the circuit model on a scalable physical system. In addition, 100-qubit ion-trap chips are being designed [Kim et al. 2005] whose underlying geometrical construction has been improved greatly in recent years [Seidelin et al. 2006; Pearson et al. 2006; Britton et al. 2006]. Figure 1 illustrates the abstraction of the physical ion-trap layout used by the QLA design scheme. The layout can be represented as a collection of trapping regions connected through shared junctions. A fundamental time-step, or clock cycle, in an ion-trap computer can be defined as any physical operation (one-bit or two-bit) on a single ion-qubit, a basic move operation from one trapping region to another, and measurement.

Table I summarizes current experimental parameters and corresponding optimistic parameters for ion traps. All baseline parameters shown in the table are followed by their projected parameters in parentheses, extrapolated following recent literature [Wineland and Heinrichs 2004; Steane 2004; Ozeri et al.

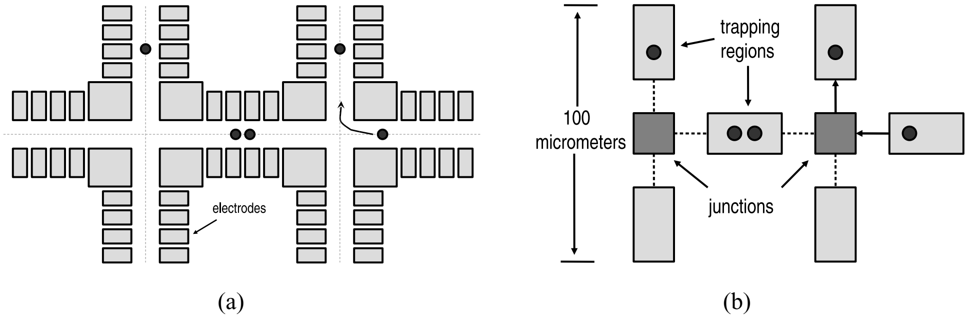


Fig. 1. (a) A simple schematic of the basic elements of a planar ion-trap for quantum computing. Ions are trapped in any of the trapping regions shown and ballistically shuttled from one trapping region to another. When two ions are together, a two-qubit gate can be performed; (b) our abstraction of the ion-trap layout. Each trapping region can hold up to two ions for two-qubit gates. The trapping regions are interconnected with the crossing junctions, which are treated as a shared resource.

Table I.

Operation	Timemicrosec. now (possible)	FailureRatesnow (expected)
SingleGate	1 (1)	10^{-4} (10^{-8})
DoubleGate	10 (10)	0.03 (10^{-7})
Measure	200 (10)	0.01 (10^{-8})
Movement	20 (10)	0.005 (5×10^{-8})/*micrometer*
Split	200 (0.1)	
Cooling	200 (0.1)	
Memorytime	10 to 100 *seconds*	
TrapSize	$\sim$ 200 $(1-5)$ *micrometers*	

Column 1 gives estimates for execution times for basic physical operations used in the QLA model. Currently achieved component failure rates are based on experimental measurements at NIST with $^9Be^+$ ions, and using $^{24}Mg^+$ ions for sympathetic cooling.

2004]; these estimates are used in modeling the performance of our architecture. In our subsequent analysis we will assume that each *clock cycle* for a fundamental time-step has a duration of 10 μs (the experimental duration of a two-qubit gate [Reichle et al. 2006]), failure rates of 10^{-8} for single-qubit operations and measurement, 10^{-7} for CNOT gates [Ozeri et al. 2004], and 10^{-6} per fundamental move operation. The movement failure rate is expected to improve from what it is now as trap sizes shrink and electrode surface integrity continues to improve. We assume trap sizes of 5 μm each [Wineland et al. 2005], and on the order of 10 electrodes per trapping region [Hensinger et al. 2005], which gives us a trapping region dimension (including the junction) of 50 μm. The parameters chosen for the example are optimistic compared to Balensiefer et al. [2005] and Meter and Oskin [2006]. Both of those papers assume more pessimistic nearterm parameters which are useful for building a 100 bit prototype, but probably not for a scalable quantum computer that can factor 1024-bit numbers using Shor's algorithm. Based on the quantum computing ARDA roadmap [Wineland and Heinrichs 2004], we feel justified in using aggressive parameters when looking 10 to 15 years into the future.

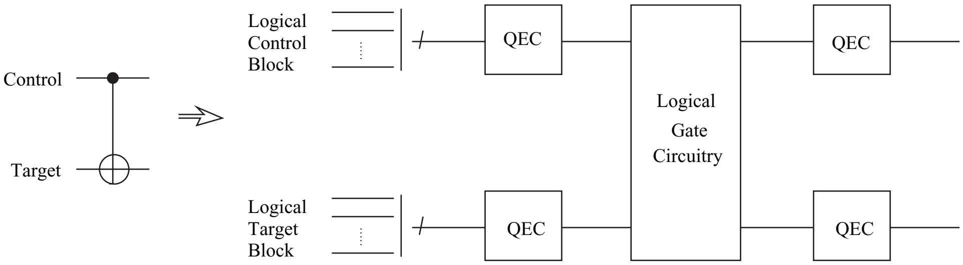


Fig. 2. A schematic for a physical-gate-to-logical-gate sequence for a CNOT operation. Quantum error correction (QEC) procedures precede and follow the logical gate procedure.

2.2 The Need for Error Correction

The circuit model of quantum computation provides the most straightforward interface between algorithms and hardware that is analogous to the operation of current classical architectures. In the circuit model, quantum algorithms are described as a sequence of operations applied on a number of qubits, where each qubit is a quantum system with the two basis states $|0\rangle$ and $|1\rangle$. In our QLA design, we restrict ourselves to a small set of universal quantum gates composed of arbitrary single-qubit operations, the two-qubit *controlled*-NOT (CNOT) gate, and measurement [Barenco et al. 1995]. In particular, our universal gate set consists of the single-qubit Hadamard gate, the three single-qubit Pauli operators (bit-flip, phase-flip, and their product), the two-qubit CNOT gate, measurement, and the single-qubit phase rotations of $\pi/4$ radians known as the T gate and $\pi/8$ radians denoted as the S gate. An arbitrary N-qubit gate can be built using this universal gate set, and is described by a $2^N \times 2^N$ complex-valued unitary matrix that acts on a 2^N-element vector that describes the state of the N-qubit register.

Fault-tolerant quantum error correction codes are necessary to ensure that the underlying assumptions behind the circuit model, such as unitary logic gates and quantum qubit states decoupled from the environment, are preserved with minimal loss in reliability, given faulty device mechanisms. To accommodate error correction, single qubit states at the application level in the QLA architecture are stored as *logical* qubit blocks, whose state is encoded into the state of a number of lower-level physical qubits. Quantum logic gates become *logical gates*, which consist of a number of lower-level physical operations whose overall effect on the logical qubit is equivalent to the corresponding gate action on a physical qubit. Each logical gate must be followed by a network of operations that corresponds to the error correcting procedure of a code that corrects some number of t errors on the physical qubits. Thus, a sequence of operations in a quantum algorithm that requires K time-steps can be abstracted as

$$EC \times U_1 \times EC \times U_2 \times EC \times \cdots \times U_K \times EC, \tag{1}$$

where each logical gate U_i at time-step i is followed by error correction. Figure 2 is a coarse schematic of the controlled-NOT (i.e., CNOT) gate, where the logical gate circuitry is preceded and followed by quantum error correction procedures.

The idea of *fault tolerance* [Shor 1995; Steane 1996; Knill and Laflamme 1997; Aharonov and Ben-Or 1997; Kitaev 1997] is of integral importance to successful quantum error correction. A fault-tolerant implementation of a logical gate for a t-error correcting code means that less than t errors during the gate and error correction procedure will not cause greater than t errors before the next error correction step. Much of the theory of fault tolerance relies on both faulttolerant lower-level network construction and technological capabilities, where the reliability of logical qubits will be improved only if the physical component failure rate is below some threshold value. The threshold value is related to the inverse of the number of physical gates in the logical gate network.

In general, any quantum subroutine, such as the quantum Fourier transform (described in detail in Nielsen and Chuang [2000]) for example, can be represented as a single unitary operator acting on an N-qubit quantum register. Through quantum circuit synthesis methods [Shende et al. 2004, 2003; Bullock and Markov 2004] derived from linear algebra theory, any N-qubit quantum subroutine can be decomposed into any universal set of operations. The QLA architecture is designed such that the compilation of a quantum application is done entirely through classical processing, where each quantum subroutine is: (1) decomposed into a high-level sequence of logical quantum operations that are members of the universal set of operations; (2) the resulting program is made fault tolerant by inserting the necessary error correction networks after each logical gate and decomposing each logical gate into a corresponding fault-tolerant network of lower-level operations; and (3) finally, decomposing each lower-level operation into the necessary sequence of technology pulses and scheduling it for execution on the quantum hardware.

The necessary application-level data distribution mechanisms are calculated during the second compilation step. In the QLA architecture, we have chosen to integrate the concept of quantum teleportation [Bennett et al. 1993], and therefore, the fault-tolerant breakdown of the low-level teleportation mechanisms involving the communication of two logical qubits is inserted within the two-qubit gate circuitry abstracted by Figure 2.

Given the aforesaid scalability considerations, the QLA architecture is designed as a large-scale model consisting of a number of qubit structures. The letter exploit the principle of locality to compute and protect as many qubits as possible by limiting the physical transmission distance of the quantum data. The qubit structures can be connected by a carefully designed teleportation-based interconnect that allows information to be preserved over significantly large distances. A schematic of the QLA model is shown in Figure 3, where we see a homogeneous array of logical qubits implemented as *self-contained* computational tiles, and connected through teleportation-based communication channels that utilize the concept of quantum repeaters. The logical qubit tiles are self-contained to ensure that the necessary quantum physical resources exist when the compilation step creates the low-level fault-tolerance subroutines for each logical operation on each logical qubit.

Each logical qubit in the architecture model of Figure 3 can be thought as a separate processing element, designed to execute a *localized* piece of the larger

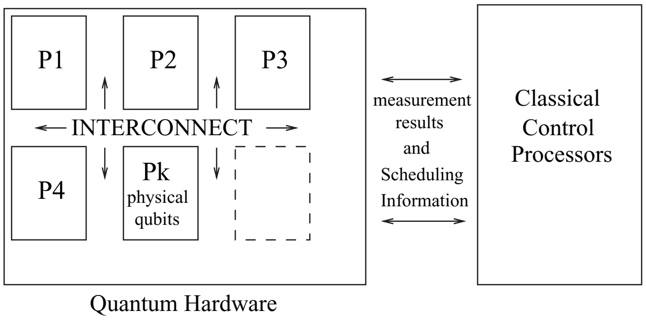

P1
P2
P3
INTERCONNECT
P4
Pk
physical qubits
measurement results and Scheduling Information
Classical Control Processors
Quantum Hardware

Fig. 3. High-level schematic for a quantum computer architecture.

application: a single logical gate followed by error correction. Communication between processing elements is implemented through the teleportation-based interconnect, while communication within is implemented through physical qubit movement as allowed by the underlying technology. Classical control processors orchestrate the scheduling of quantum operations, where the only means of communication between the classical and quantum hardware is through measurement results. By completely decoupling communication and computation, and by using deterministic error correction routines to simulate the performance of each logical qubit tile, we have created an architecture that is in essence a collection of ASIC units, where each ASIC is the logical qubit tile. Creating the timed physical-pulse sequences for a quantum application subroutine, therefore, follows the classical GALS model: globally asynchronous (application level) and locally synchronous (within each logical qubit tile, the error correction procedures can be the same).

Another logical communication alternative is analogous to the physical-level nearest-neighbor communication protocol, which is employed by a large number of promising technology implementations where the qubit containers are fixed in space [Kane 1998; Skinner et al. 2003; Hollenberg et al. 2003; Niskanen et al. 2007; Hime et al. 2007]. Additionally, recent work by Fowler et al. [2007] presents a scalable architecture model for flux qubit quantum computers [Hime et al. 2007; Niskanen et al. 2007], which at the low-level is dependent on nearest-neighbor interactions. Nearest-neighbor communication is analogous to the ion-trap ballistic movement model if each movement "hop" of an ion from one trapping region to the next is replaced with a "swap" operation of the data qubit with another qubit along the direction of movement. Recent threshold studies have proven that the nearest-neighbor physical communication mechanism is fault tolerant [Svore et al. 2006], and there is no fundamental reason for the physical implementation of the QLA architecture not to be realized by some nearest-neighbor interaction technology. Moreover, circuit synthesis research [Shende et al. 2006] has shown that nearest neighbor communication preserves the asymptotic depth of quantum circuits.

At the logical level, however, a nearest-neighbor-based interconnect will require a number of logical "swap" operations equivalent to the number of logical qubits that separate the source qubit and the destination qubit. To preserve

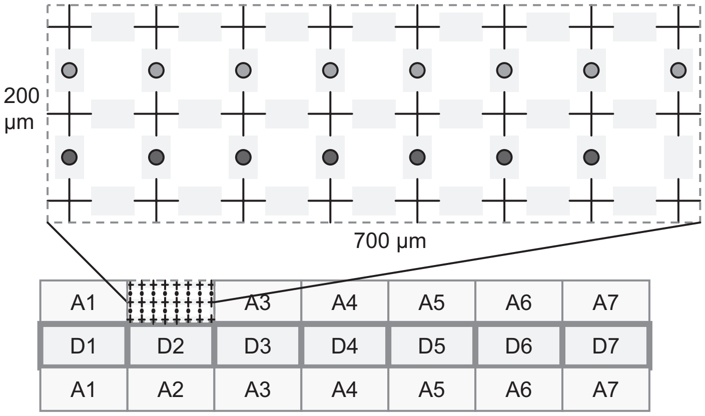


Fig. 4. The logical qubit: seven groups of three level-1 blocks make a single level-2 logical qubit (middle). The two identical conglomerations on the sides are ancillary blocks used for error correction. The shaded boxes of the level-2 qubit are the encoded data level-1 blocks, which are supported by their respective level-1 ancilla blocks.

fault tolerance, each of these "swap" operations must be treated as a transversal logical gate and thus followed by error correction. Given that the duration of a single time-step in the execution of a logical circuit with QLA architecture is defined by the time to error correct, the nearest-neighbor logical interconnect introduces a linear-factor slowdown for the application execution, equivalent to the average communication distance. For factoring a 1024-bit Shor's algorithm, for example, the largest distance traveled by logical qubits will require 256 swap operations during the modular exponentiation component of the algorithm, and thus 256 additional error correction steps. Choosing teleportation-based interconnect for the QLA architecture allows us to "hide" the temporal cost of communication and to decouple it from any logical qubit subroutines. In Section 4, we describe how the overlap of communication with computation is possible, thus eliminating the overhead of additional error correction during movement of logical qubits, even if the technology implementation is such that only nearest-neighbor physical qubit interactions are permitted.

3. LOGICAL QUBIT DESIGN AND COST OF ERROR CORRECTION

The structure of the logical qubit in the QLA is driven by the ion-trap characteristics shown in Table I, which place us significantly below the accuracy threshold value required by the threshold theorem. These parameters are optimistic, but physically achievable in theory [Wineland and Heinrichs 2004; Steane 2004; Ozeri et al. 2004]. Particularly important is the fact that the lifetime of an ion is currently much larger than quantum operations, which are on the order of tens of microseconds. The relatively low memory-error rates allow us to significantly reduce the area of a logical qubit, by reducing the parallelism within a single error correction cycle as well as the ancillary qubits required by the chosen error correction algorithm.

Figure 4 shows the full implementation of a level-2 qubit tile. To reduce communication and complexity, we chose to make each logical qubit a self-contained

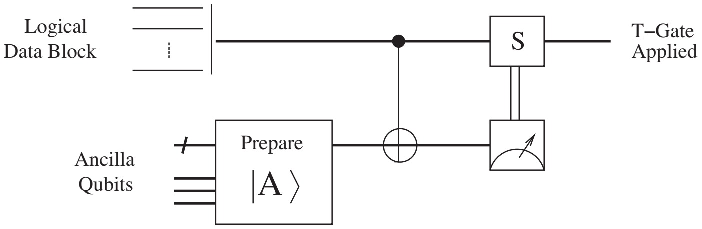


Fig. 5. Logical circuit for the T gate. Some of the logical qubit tiles in the architecture are designated specifically for the preparation and storage of the ancillary data used for the T gate implementation. The S gate is transversal.

unit that requires no additional quantum resources to perform logical gates and error correction. This allows the error correction for each logical qubit to be performed in parallel with error correction on any other logical qubit. There are two high-level ancilla blocks in a single level-2 qubit, a design decision which allows the simultaneous error correction of two level-2 qubits inside a single qubit tile. The two sets of high-level ancilla are necessary in computational tiles to ensure that both of the logical data qubits are error corrected in place immediately after the execution of a two-qubit gate, without stalling the application execution.

In the QLA, a single data logical qubit at level 2 is built by encoding seven level-1 qubit blocks with the Steane $[\![7, 1, 3]\!]$ code. A level-1 qubit block is shown at the top of Figure 4. Seven level-1 blocks interact to encode a logical qubit block at level 2. In Figure 4, level-1 blocks $A1$ through $A7$ form two level-2 ancilla and the data qubit is encoded in blocks $D1$ through $D7$. There is nothing fundamental about the grid-like structure at the lowest level; it simply lends itself more readily to the planar-trap design of ion-trap technology [Hensinger et al. 2005]. It also allows any two-dimensional quantum computing layout to be similarly modeled.

We have chosen the $[\![7, 1, 3]\!]$ error correcting code because it allows a *transversal* implementation of the universal set of logical gates assumed by the QLA architecture, with the exception of the phase gate with angle of rotation $\pi/4$ radians, also known as the T gate [Nielsen and Chuang 2000]. The T gate is the only nontransversal logical gate in the universal set of gates for quantum computation. The T gate is a de facto two-qubit gate, where a specially prepared encoded ancilla state is required to implement the logical T gate through a single-bit teleportation mechanism [Gottesman and Chuang 1999]. The circuit for the logical T gate is shown in Figure 5, and corresponds to the circuit for the logical CNOT gate shown in Figure 2. The preparation procedure for the logical ancilla used by the T gate circuit is described in detail in Aliferis et al. [2005].

A transversal logical quantum bit-flip gate on a logical qubit can be implemented by applying 49 physical bit-flip gates on the physical ion-qubits in parallel. Similarly, a transversal logical CNOT gate is implemented by bringing 49 ions from some qubit-tile A in the same trap as the 49 ions in qubit-tile B. After 49 CNOT gates on the joined ions, the two sets are error corrected by the ancilla on both sides of the data region in a level-2 tile. The ancilla preparation network at level 2 does not require specially designated verification blocks, as the errors are detected during lower-level syndrome extractions [Reichardt 2004].

Considering communication, the level-1 error correction circuit takes 154 cycles, where each cycle is on the order of 10 microseconds, and can be as large as 0.003 seconds per error correction procedure at level 1. In our time estimates we choose to provide a single laser per level-1 block, which means that only one operation can be executed at any given clock cycle. The latency introduced by serializing the level-1 circuit does not qualitatively change the large-scale evaluation estimates, since a maximally parallelized circuit would take approximately 127 cycles per error correction procedure. Clearly, the number of cycles depends on the quality of the scheduling algorithm, and most of all on the two-dimensional layout that provides communication channels for the qubits. To extract the physical communication resources for low-level networks such as level-1 error correction, we have employed the QPOS scheduling tool [Metodi et al. 2006], which is based on traditional compiler scheduling heuristics [Chekuriet al. 1996; Deitrich and Hwu 1996; Shobaki and Wilken 2004], and has demonstrated better circuit schedules than carefully hand-optimized layouts. In this manner, we can extrapolate that fully serialized error correction at level 2 will last approximately 0.3 seconds, which is two orders of magnitude longer than the time to error correct at level 1.

We have made the following assumptions when extracting the error syndromes for both level-1 and Level-2 qubit blocks: (1) Two syndromes are extracted in *serial* for both X and Z errors; and (2) in the case of a nontrivial syndrome, the next extracted syndrome will match it, thus we can proceed with the error correction step. Since our logical qubit at level 2 is equipped with parallel syndrome extraction, assumption (a) makes Eqs. (2) and (3) an overestimate of the final latency $T_{L,ecc}$. We have

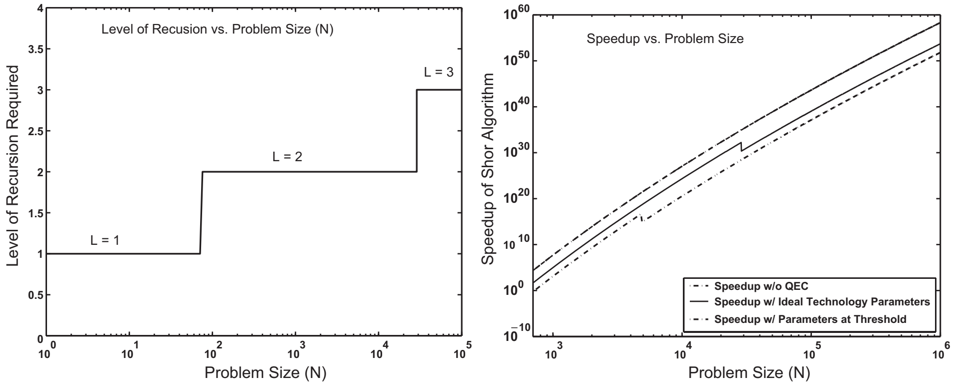
$$T_{L,ecc} = 2 \times T_{L,synd}, \tag{2}$$

for a trivial syndrome indicating no errors in the data, and

$$T_{L,ecc} = 2(2T_{L,synd} + T_1 + T_{L-1,ecc}), \tag{3}$$

for a nontrivial syndrome indicating the presence of errors, and where $T_{L,synd}$ is the time to extract a syndrome at level L, which is a function of the time to prepare the logical ancilla block. Moreover, T_1 denotes the time of a logical one-qubit gate, and $T_{L-1,ecc}$ is the time for a lower-level error correction step that follows each level-L logical gate.

Numerical simulations of a level-2 qubit showed that a nontrivial syndrome was measured for level 1 with a rate of $3.35 \times 10^{-4} \pm 0.41 \times 10^{-4}$, and for level 2 at a rate of $7.92 \times 10^{-4} \pm 0.81 \times 10^{-4}$. Our simulations did not yield more than two repetitions of a syndrome before the error correction step at optimistic error rates for ion traps. Thus, it is a reasonable assumption that in the case of a nontrivial syndrome we require at most one more syndrome extraction before we are ready to apply the correcting gate. Taking a weighted average of the two cases in Eqs. (2) and (3) we determine a level-2 error correction time



(a) (b)

Fig. 6. (a) Required level of recursion for Shor's algorithm as a function of the problem size N, defined in the context of an N-bit number that is being factored; (b) speedup of Shor's algorithm as a function of the problem size N. The topmost line shows the speedup without error correction, and the middle line shows the speedup with error correction,and but at error parameters approximately 3 orders of magnitude below the accuracy threshold for the Steane $[\![7,1,3]\!]$ code. In the bottom line the error parameters are at threshold value of the $[\![7,1,3]\!]$ code. Each "glitch" in the two lower lines is an increase in the level of recursion.

of approximately 0.3 seconds. As shown in Figure 6(a), using level-2 recursion with this qubit tile design is sufficient for factoring numbers as large as 2048-bit modulus.

We used QASM-TOOLS, formerly known as ARQ, to empirically compute p_{th} at level 2 for the QLA logical qubit. Our results show that the failure probability of a single one-qubit logical gate rapidly drops to zero at component failure rates lower than $p_{th} = (2.1 \pm 1.8) \times 10^{-3}$. Above this value, the rapid decrease in the reliability of our system as recursion increases can be attributed to the additional resource overhead of recursion.

The estimated threshold failure probability is much higher than the theoretical estimate of 7.5×10^{-5} computed in Svore et al. [2004], for several reasons: (1) The structure of the qubit is optimized for the error correction circuit and may vary for different codes; (2) the high reliability of ion-trap memory has allowed us to significantly reduce the overall area and ancillary resources required; and (3) the fixed, low movement error probability, as well as the fact that we made the design decision to never physically move the data, pushed our qubit's threshold closer to the 9×10^{-3} threshold value estimated by Reichardt Reichardt [2004]. We observed no failure at level-2 recursion as the physical component errors approached the expected ion-trap parameters from Table I, which was anticipated. Using the methodology described in Gottesman [2000] for calculating the reliability of an encoded operation, given the effect of communication with our empirical threshold value, we find an estimated level-2 logical qubit reliability approaching 10^{-21}.

From a first look, it seems that the exponential slowdown due to error correction, even with qubit tiles of only a few levels of recursion, is prohibitive when the system size S becomes very large. For some applications, however, an exponential slowdown from error correction is balanced by the exponential speedup offered by the quantum algorithm structure versus its classical counterpart. One such application is Shor's quantum factoring algorithm, which is designed to break the widely used RSA public-key cryptosystem. RSA's security lies in the assumption that factoring large integers is very hard, and as the RSA system and cryptography in general have attracted much attention, so has the factoring problem. The efforts of many researchers have made factoring easier for numbers of any size, irrespective of hardware speed. However, factoring is still a very difficult problem. The best classical algorithm known today [Buhler et al. 1994] has complexity of

$$\exp\left((1.923 + o(1))(log\ N)^{1/3}(log\ log\ N)^{2/3}\right)$$

for an N-bit integer. As a basis of comparison we use the most recent success at factoring a 663-bit number [Bahr et al. 2005] classically for an estimated 121,000 MIPS years ($\approx 4 \times 10^{18}$ instructions). This is equivalent to a little over one year on a 100 GHz PC with a perfectly parallelized and distributed factoring implementation.

A plot of the required level of recursion versus the problem size N for factoring an N-bit integer using Shor's algorithm is shown in Figure 6(a). The system parameters used are the Steane $[\![7, 1, 3]\!]$ code with the optimistic ion-trap technology assumptions shown in Table I. Slowdown due to error correction can be seen in the logarithmic scale plot shown in Figure 6(b), where the $\hat{y}$-axis marks the speedup of the quantum algorithm over its classical counterpart. The speedup is calculated as the number of days classically divided, by the number of days quantum mechanically. The top line shows speedup without error correction. The middle line shows the speedup with optimistic ion-trap parameters, while the bottom line shows the speedup with technology error rates at the threshold value of approximately 10^{-5}. Each "blip" on the speedup lines with error correction corresponds to increasing the level of recursion by one unit. The smallest problem size shown is $N = 700$, which requires level-2 encoding. The same problem size requires level-3 encoding if the technology parameters are at threshold value.

As we can see, even with error correction, the exponential speedup is preserved over classical computation. The slight curvature of the speedup lines indicates that the speedup achieved from Shor's quantum algorithm for factoring is not truly exponential. This is because the classical cost of factoring a number is in fact not exponential, but superpolynomial. A physical operation in an ion-trap quantum computer is on the order of 10 μs; thus at the physical level, the speedup calculated is based on a KHz quantum computer.

4. LOGICAL QUBIT INTERCONNECT

A logical two-qubit gate between level-2 qubits $Q1$ and $Q2$ is executed by moving all 49 physical ion-qubits that encode qubit $Q1$ to the computational tile where qubit $Q2$ resides. If the application being executed is the factoring of a 1024-bit number using Shor's factoring algorithm, $Q1$ could be moving as much as 0.5 meters (or 256 logical qubits) across the ion-trap chip. Even if it may be possible to ballistically transport ions between *adjacent* logical qubits, greater

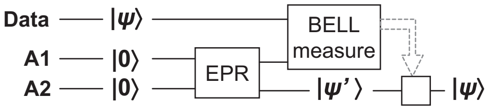


Fig. 7. Illustration of the stages of teleportation.

distances are physically impossible for direct ion transport, and thus require the dedicated teleportation-based logical interconnect architecture.

The teleportation protocol [Bennett et al. 1993] requires three qubits: the source qubit whose quantum state $|\Psi\rangle$ is being transported, and two auxiliary qubits that facilitate the data transfer. A circuit schematic that illustrates the stages of teleportation is shown in Figure 7. In the figure no actual quantum gates are shown, but it is an accurate schematic of the general circuit representation of quantum algorithms, where time goes from left to right and each line represents the evolution of each qubit through time. The first stage is to prepare the two auxiliary qubits initialized to $|0\rangle$ into a maximally entangled EPR state: $\frac{1}{\sqrt{2}}(|00\rangle + |11\rangle)$. The second stage is to physically transport one of the two EPR qubits to the destination and the other, A1, to the source qubit. Once A1 has arrived at the locale of the source qubit, a *Bell measurement* is applied on A1 and the source qubit. The result of a Bell measurement is to collapse the states of the two qubits in any of the four possible states known as *Bell states* $\{|\Psi_+\rangle,\ |\Psi_-\rangle,\ |\Phi_+\rangle,\ |\Phi_-\rangle\}$ [Bell 1964].

$$\begin{aligned}
|\Psi_+\rangle &= \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle) &&\rightarrow \text{.............. no errors}\\
|\Psi_-\rangle &= \frac{1}{\sqrt{2}}(|00\rangle - |11\rangle) &&\rightarrow \textit{Z error on } q1 \textit{ or } q2\\
|\Phi_+\rangle &= \frac{1}{\sqrt{2}}(|01\rangle + |10\rangle) &&\rightarrow \textit{X error on } q1 \textit{ or } q2\\
|\Phi_-\rangle &= \frac{1}{\sqrt{2}}(|01\rangle - |10\rangle) &&\rightarrow \textit{both X and Z errors} \qquad (4)
\end{aligned}$$

The result of the Bell measurement step is to force qubit A2 into the original state of the data qubit with some error labeled $|\Psi\rangle'$, as in Figure 7. The error can be corrected based on the classical information available as a result of the Bell measurement and we can recreate the true original state of the source qubit over A2 at the desired location. Note that the original data has been destroyed, thus teleportation does not copy the quantum data and prevents direct movement of the data, but requires physical movement of auxiliary EPR qubits.

At first glance, it is not clear that teleportation is a more reliable method for communication than direct physical movement. The combined physical distance that the two EPR qubits move is equal to the distance which the data qubit would have moved, thus the errors accumulated due to movement are not eliminated. EPR pairs, however, are replaceable and can be prepared offline. Hence, so long as we have a single faithful EPR pair that connects the source qubit to its destination location, we can recreate the source at the destination

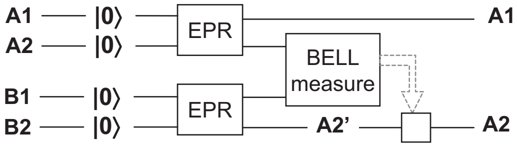


Fig. 8. When A2 is teleported over B2, the (A1, A2) EPR pair spans the distance between A1 and B2.

with nothing more than the transmission of classical information followed by a few single-qubit operations to correct the error.

Damaged EPR pairs can be repaired through a process called *entanglement purification* [Bennett et al. 1996; Deutschet al. 1996], which uses ancillary EPR pairs to distill good from bad pairs. The caveat to purification is that the number of resources increases exponentially with EPR separation distance until purification becomes impossible. We assume that a teleportation step is successful if the source qubit is recreated at the destination location with probability of failure less than the threshold value of the error correction code that is used to encode the logical qubits of which each teleported data qubit is a part. For our purposes we use the empirical threshold estimate of 7.5×10^{-5} used in Svore et al. [2004].

To allow the successful teleportation of physical qubits between any two logical qubits in the architecture at arbitrary distance, we use the fact that entanglement is preserved through teleportation. For example, a logical qubit encoded into the state of 49 physical qubits remains unchanged after some of the physical qubits have been teleported to another region of the architecture. Similarly, teleporting one of the two qubits in an EPR pair does not break the maximum entanglement between the two qubits. The logical qubit interconnect in the QLA architecture is designed such that EPR pairs can be created between source and destination through teleporting the EPR qubits themselves, rather than physically moving them apart, as shown in Figure 8. EPR pairs (A1, A2) and (B1, B2) are created and then can be separated such that after A2 is teleported over B2, the distance that (A1, A2) spans is almost twice as large as it was originally. The protocol in Figure 8 is known as *entanglement swapping*, where entanglement of EPR pairs is transferred through a channel divided by a number of smaller segments known as *quantum repeaters* [Dur et al. 1999]. The result is a single EPR pair that spans the entire channel.

The logical qubit interconnect in the QLA is based on the quantum repeater protocol, a schematic of which is shown in Figure 9. EPR pairs only travel to two nearby repeater islands (shaded boxes), where they can be efficiently purified using the purification protocols with some additional ancillary EPR pairs. To create a single entangled EPR pair between the source and destination spanning over the entire channel, we use a logarithmic algorithm similar to computing the transitive closure. Each stage of the protocol shown in Figure 9 reduces the number of connecting EPR pairs by half, without destroying the connection between source and destination. Finally, we teleport the source qubit to its desired location when a single EPR pair spans the connection channel. An EPR pair distributed in such a way between the logical qubits $Q1$ and $Q2$

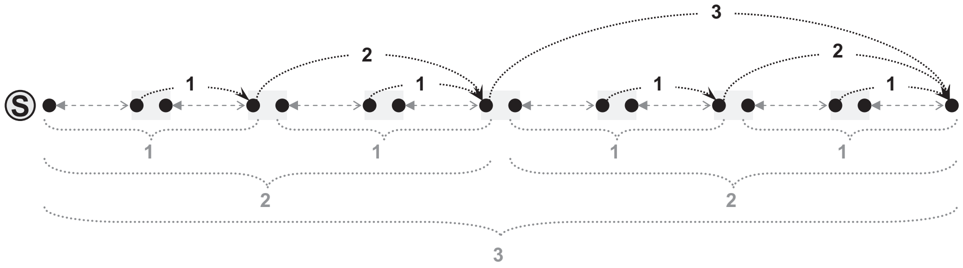


Fig. 9. Illustration of entanglement swapping protocol for teleporting the source qubit marked with "S". The bottom portion of the figure shows the scope of the resulting EPR pairs after each teleportation step.

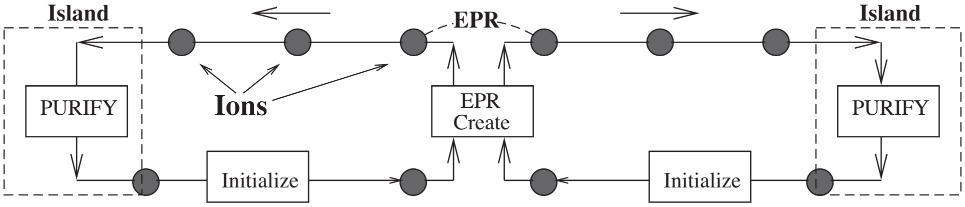


Fig. 10. Detail of a channel between two repeater stations. The channel is a two-way ballistic transport region, where EPR pairs are created in the middle and distributed in pipeline fashion to the two island/reapeater stations.

is required for each of the 49 ion qubits of qubit $Q1$ such that each of the 49 qubits can be teleported to the computational tile of qubit $Q2$.

EPR pairs are created in between all adjacent repeater islands and separated by the two opposing ends. To minimize accumulated errors from EPR generation and movement, the EPR pairs are purified [Bennett et al. 1996] using a number of additional EPR pairs. To optimize space as well as performance, and to allow enough EPR resources for purification, we strategically model the physical channels between each island as a two-way ballistic transport region, as shown in Figure 10. The figure also illustrates the pipeline purification protocol employed by the QLA architecture for purifying a single EPR pair. The basic idea of purification is to use several copies of lower-fidelity EPR pairs to *distill* a single high-fidelity EPR state that can be used for teleportation. In Figure 11 a detailed schematic of the pipeline network is shown when repeater stations are placed between every logical qubit. The source ion is being teleported in the direction shown by "DESTINATION" after the channel is prepared.

If the initial preparation fidelity is high enough, by applying successive purification steps an EPR pair can be purified to an arbitrarily high fidelity. The pipeline purification sequence works by designating one EPR pair as the data pair which is continually purified in round-robin pipeline fashion by the additional ancillary EPR pairs. We assume sufficient ion resources in the pipeline to handle the maximum amount of required purification steps, without having to wait for the creation of new EPR pairs before each successive purification step. The original purification protocol was formulated by Bennett et al. [1996],

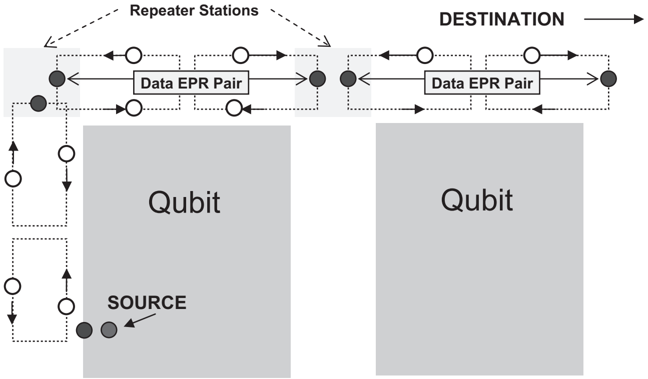


Fig. 11. A slightly higher level of detail of the communication channel as repeater stations are placed at the corner of each logical qubit. Ions are not to-scale compared to the size of the logical qubit or the pipeline-based interconnect.

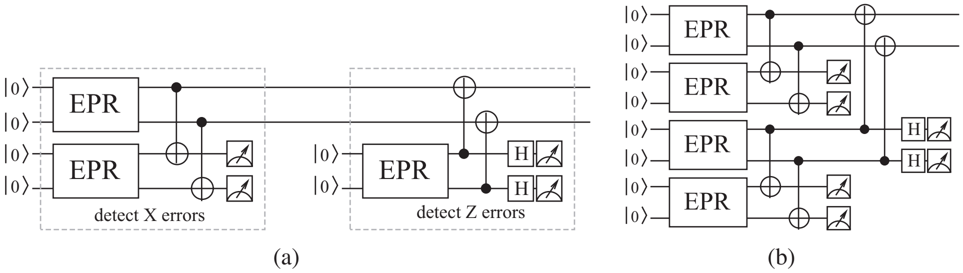


Fig. 12. (a) The data EPR pair (top) is created in parallel with an ancillary EPR pair, which is first used to detect bit-flip errors. Phase-flip errors are detected with a third EPR pair or with the previous ancillary pair reinitialized; (b) four EPR pairs are created, two of which are used to check the other two for bit-flip errors. This is followed by the detection of phase-flip errors on the two EPR pairs remaining.

where the efficiency of purification depends highly on the reliability of the physical gates that make up the protocol (i.e., Hadamard and CNOT gates) and on the inital fidelity of the EPR pair [Dur et al. 1999]. We use the recursive fidelity equations in Dur et al. [1999] (i.e., Eqs. (9) and (19), where the fidelity of ancillary EPR pairs remains constant) to study the implementation employed by our architecture for purification protocols. The efficiency of these protocols depends as much on gate reliability as on the type of errors that occur and how the errors accumulate in EPR states before and during purification.

The purification circuit is shown in Figure 12, where two example purification network choices are shown. After its creation or even during the purification procedure, the data EPR pair accumulates the bit-flip of phase-flip errors that can place it in any of the four possible Bell states. The purification circuit shown in Figure 12(a) uses one ancillary EPR pair to check the state $|\Psi_+\rangle = (|00\rangle + |11\rangle)/\sqrt{2}$ first for bit-flip errors and then another ancillary EPR pair to check the state for phase-flip (i.e., sign) errors. The schematic in Figure 12(a) corresponds to the purification protocol intended for a sequential pipeline design. After interaction with the data EPR pair through the CNOT gates, the two ancillary qubits are measured, where odd parity for either X or Z error checks will indicate an error in the data EPR pair. In the case of an error, the data EPR qubits are recycled in the pipeline and the next ancillary EPR pair resumes the function of the data, which is then purified. Each successful purification step increases the likelihood that the data EPR pair is free of error, thus increases its fidelity. The principle is the same as throwing a weighted coin with unknown weight for obtaining either heads or tails. Each time the coin lands heads, given that it has landed heads in the previous throw, the probability that the coin is truly weighted towards heads increases.

An alternate purification procedure is shown in Figure 12(b), where four EPR pairs are prepared in parallel at the beginning. The data EPR pair is at the top and is checked in parallel with an additional EPR pair for X errors. If both pass, the data EPR is checked for Z errors. Although we haven't studied this protocol, it may have the potential to offer better purification efficiency by ensuring that the ancillary EPR pair used in Z error detection is checked against X errors. The interaction between the two EPR states when checking for Z errors will cause an X error in the ancilla to propagate to the data through the CNOT gates. This X error would remain undetected when the teleportation procedure is executed. The implementation of the network in Figure 12(b) would require different EPR generation and island structure, where each island would need to hold more than one data EPR pair at each node.

In reality, any of the four Bell states can be used for teleportation; thus, the purification efficiency can be further improved if we allow the X or Z errors on the EPR qubits to remain, and use subsequent purification steps to ensure that, indeed, the X and Z errors detected in the previous step are present. In such cases we know which of four Bell states our EPR qubit is in, and modify the teleportation protocol accordingly. The modification consists of a different interpretation of the 2-bit bit-string that indicates how to apply correcting gates on the destination qubit to recreate the source qubit (shown in Figure 7) at the end of the teleportation protocol.

The first tradeoff when considering microarchitecture design for the communication network arises as we speculate on the number of ion-qubit resources required to distill a single high-fidelity EPR pair spanning any distance. Clearly, the minimum resource required consists of four ion-qubits when considering the network in Figure 12(a): two for the data EPR pair and two for the ancillary pair used for purification. The ancillary pair is continuously reprepared for each purification step. Alternately, the maximum number of resources used can be reached by creating all EPR pairs required for j purification steps, which would require $\eta(2 \times 3)^j$ ion-qubits, where η is some constant that takes into account the possibility of failure at some stage in the purification.

Both resource extremes are shown in Figure 13, where the protocol that uses the minimum resources is shown on the lefthand side. Ion-qubits A and B are continuously reprepared and interact with the data EPR pair at each step of purification. In the scheme on the righthand side, a two-step purification tree is shown, where 18 ion-qubits are prepared into three groups of three EPR pairs used for the first purification step. After the first step, three purified

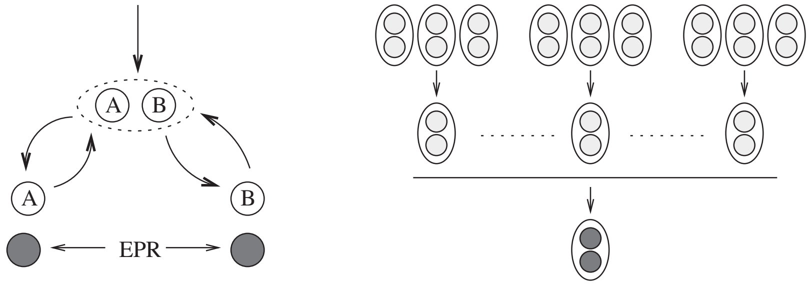


Fig. 13. Minimum and maximum number of resources needed to purify a single EPR pair.

EPR pairs remain and are used to further distill a single EPR pair. While the first protocol uses far fewer resources, the final fidelity of the data EPR pair is severely limited by the fact that the ancillary EPR pair is continuously reprepared and retains the same level of noise throughout the purification process. In the second protocol, on the other hand, the data and ancillary EPR pairs are equally purified at each step, and a much higher fidelity is achieved for the final EPR pair. This, however, is at the expense of using a high number ion-qubit resources, as well as a complex microarchitecture that supports the movement of all EPR pairs at each purification step. The pipeline approach we use as shown in Figure 10 allows sequential purification without memory-cycle delay between each purification step. By avoiding recursive purification and pipelining the ancillary EPR qubits, the QLA is able to utilize significantly reduced bandwidth requirements for each distillation of EPR pairs between two adjacent repeater stations.

A second important tradeoff arises when deciding the separation between two adjacent repeater stations and thus the distance at which EPR pairs are purified. There are three possible ways to connect a source and destination separated by a number of repeater islands, such that the final teleportation step of the data qubit between source and destination is teleported with the desired threshold fidelity required for error correction. We list these three methods next.

(1) A *purely linear* approach distills high-fidelity EPR pairs only between adjacent islands to some fidelity F, which will allow $O(\log K)$ teleportation hops (see Figure 9) to be performed such that the final fidelity of teleported data is within the threshold value. The total time to achieve a given relatively large distance varies as the separation between repeater islands is changed. As the separation decreases, purification will be followed by a greater number of teleportation hops between source and destination; thus more purification is needed to achieve a higher EPR starting fidelity. Alternately, as the separation increases, there are less teleportation hops, but data and ancillary EPR pairs travel longer in the pipeline, thus more purification is needed to reduce the fidelity. This is an interesting tradeoff for a system designer to explore, and offers an opportunity to design a reconfigurable dynamic interconnect.

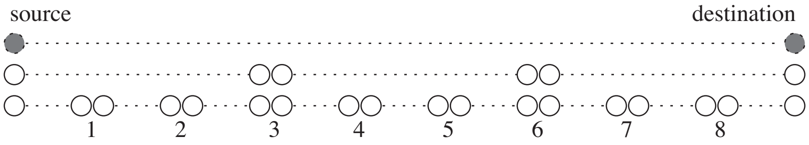


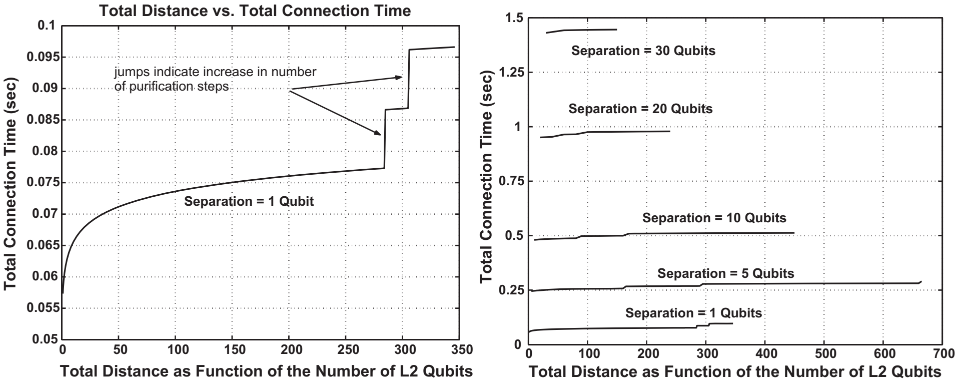
Fig. 14. Three nesting levels are shown for the nested purification protocol.

(2) A *nested, semi-linear* approach distills EPR pairs at different nesting levels with increasing scope per level. This method was analyzed in detail in Dur et al. [1999]. At the lowest nesting level, EPR pairs are created with a scope of m junctions, which are used to purify an EPR pair with the same scope at the second nesting level. The freshly purified scope-m EPR pairs are connected to create an EPR pair with scope km for some other constant k, which is then used to distill a single EPR pair of scope km at the third nesting level. This process is repeated until we have a single EPR pair connecting source and destination, as shown in Figure 14.

(3) Finally, we can create EPR pairs directly between source and destination without purifying at any intermediate scope. The purification is performed for an EPR pair that spans the source and destination until a desired fidelity is reached.

The QLA architecture utilizes the first approach, where we find that at the optimistic technology parameters for ion-traps, the distances required for communication when factoring a 2048-bit number (maximum across 512 logical qubits) are attainable when the separation between two adjacent repeater islands is 5 logical qubits.

The third approach was studied in detail in Isailovic et al. [2006], where the creation of hundreds of EPR pairs is required between the source and destination to purify EPR pairs that span the entire channel needed. Intuitevely, both the second and third approaches offer much longer final distance than the linear approach employed by the QLA architecture. For example, when the movement failure rate is reduced by an order of magnitude to O(10^{-5}), the QLA interconnect can send qubits across only 10 logical qubits, the third approach allows qubits to be sent across as many as 221 logical qubits while still attaining final fidelity within the threshold fidelity for computation. This approach, however, is very expensive when one considers the necessary EPR resources. Our simulations indicate that four purification steps are required to bring the fidelity to the threshold of $1 - 7.5 \times 10^{-5}$. This means that 16 EPR pairs must be sent across the entire channel to distill a single good EPR pair, which must be doubled to accommodate both bit-flip and phase-flip errors. In addition, this does not take into consideration the additional EPR pairs required within each two adjacent repeater stations. The greater achievable distance can be contributed to the fact that at each purification step, the fidelity of the data EPR pair and that of the ancillary EPR pair decrease equivalently, while in the purely linear approach, the fidelity of the ancillary EPR pairs remains both constant and as a function of the separation between adjacent repeater stations.



(a) (b)

Fig. 15. (a) Total time to teleport 49 qubits sequentially as a function of total communication distance for a repeater separation of 1 logical qubit; (b) same as (a), but with varying repeater separation. In each figure we performed enough purification steps such that the failure rate of the teleported data qubit is below the threshold estimate of 7.5×10^{-5}.

The pipelined linear approach offers comparatively smaller bandwidth by providing only a single pipeline-based channel from source to destination. The tradeoff is in the serialization of the purification process, however: Because the purification is pipelined, the only temporal cost suffered is an initial purification cost, which increases linearly with the separation of the repeater stations. The initial cost is the time of creating the first ancillary EPR pair and the time it takes to transport the pair to the two adjacent repeater stations. We offer estimates of the interconnect performance in Figure 15. On the lefthand side in Figure 15(a), the total connection time for teleporting 49 physical ions sequentially is plotted as a function of the desired communication distance. We see that a repeater separation of one logical qubit is enough to allow us to communicate all 49 physical ions sequentially across a distance sufficient for applications as large as one factoring a 1024-bit number.

One important observation in the plot of Figure 15(a) is that total connection time is dominated by the number of purification steps, and thus by the separation of the repeater stations. For a total distance just below 300 qubits, the total communication time behaves logarithmically as expected, but jumps suddenly each time additional purification steps are required. For factoring a 1024-bit number with maximum communication distance across 256 logical qubits and repeater separation of 1 logical qubit, we need at most 3 purification steps.

The plot shown in Figure 15(b) shows how total communication time increases as the separation between repeater islands increases. After a maximum-distance peak, the total possible distance decreases as the separation increases, simply because the EPR qubits cannot be purified enough. The increase in total connection time is linear with the increase in island separation.

The latency cost of communication between logical qubits is critical for the success of the entire architecture during the execution of an application. We have made the 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 in order to keep the failure rate due to movement below threshold. 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.

The result of Figure 15 indicate that 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? In other words, given a mapping of the high-level circuitry for some quantum subroutine such as the quantum Fourier transform, the central question is whether there is enough logical interconnect channels that a free to allow us to preprepare them for teleportation (i.e., distribute the needed spanning EPR pairs) while the source qubit is undergoing error correction.

To answer this question, we used a tool to schedule the movement of EPR pairs in QLA [Metodi et al. 2005]. We assigned one channel to carry the created EPR pairs to their destinations and another channel to return used EPR pairs. Within each channel, the EPR pairs are pipelined. We define the bandwidth of the QLA's communication channels as the number of physical channels in each direction; for example, the channel shown in Figure 10 has a bandwidth of 2. The goal of the scheduler 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 a heuristic, greedy, and 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 endpoints. A simple approach to doing a two-qubit gate between logical qubits Q1 and Q2 would be as follows: Teleport Q1 to Q2's physical location, perform the gate, and teleport it back. An optimization that the scheduler incorporates is that it only moves logical qubit Q1 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 to which the qubits are subjected. With all of the aforesaid considerations in the scheduler, we found that, given a single pipeline for each channel as shown in Figure 11, we can schedule communication such that it always overlaps with error correction of the logical qubits. The end result is reliable movement over sufficiently large distances with minimal overhead.

To test that computation and communication can be overlapped and as an example of QLA performance, we used the aforementioned methodology to schedule onto the QLA architecture a high-level execution of both quantum subroutines of Shor's factoring algorithm: the modular exponentiation routine and quantum Fourier transform. The former divides into a series of modular multiplications, which in turn divide into a series of quantum adders. We divides each subroutine into the universal gate set given in Section 2, where each Toffoli gate (the dominant gate in quantum adders) divides into 14 basic operations [Nielsen and Chuang 2000], including some T gates. As the building

Table II.

	$N = 128$	$N = 512$	$N = 1024$	$N = 2048$
Logical Qubits	37,971	150,771	301,251	602,259
Toffoli Gates	63,729	397,910	964,919	2,301,767
Total Gates	115,033	1,016,295	3,270,582	11,148,214
Area(m^2)	0.11	0.45	0.90	1.80
Time(days)	0.9	5.5	13.4	32.1

Column 1 gives estimates for execution times for basic physical operations used in the QLA model. Currently achieved component failure rates are based on experimental measurements at NIST with $^{9}Be^{+}$ ions, and using $^{24}Mg^{+}$ ions for sympathetic cooling.

block for modular exponentiation, we used the circuit-depth-optimal Draper quantum carry lookahead adder [Draper et al. 2004] which was further optimized in Meter and Itoh [2004]. Table II summarizes the performance of the homogeneous QLA architecture for Shor's algorithm when considering its two quantum subroutines.

5. CONCLUSION

In this work we have described the logical qubit interconnect of the QLA architecture. The validation of our architecture rests in our use of realistic technology expectations derived from the ion-trap literature and careful low-level simulations. The architecture is based on the circuit model for quantum computation; however, other examples of computational models exist and include the adiabatic quantum computation model [Farhi et al. 2000; Aharonovet al. 2004; Childs et al. 2002], cluster state quantum computation [Briegel and Raussendorf 2001; Zeng et al. 2003; Nielsen and Dawson 2004; Nielsen 2004], geometric quantum computation [Jones et al. 2000], and the theory of topological quantum computation [Freedman et al. 2003]. Combined, the variety of quantum computation models provides different methods for extending the application space for quantum information processing, and may someday redefine the system design of a large-scale machine. Perhaps in the future, a large-scale quantum architecture will unify several different computational models to fully maximize the possible applications being executed.

Achieving component failure rates at or below the threshold value may allow arbitrary reliability for quantum operations if sufficient error correcting resources are provided. However, the requirements for scalable communication seem much more stringent. When reducing the component failure rate parameters by just a factor of 5 from the expected ion-trap failure rates, the maximum possible distance at which we can reliably teleport an ion is approximately 80 logical qubits, which is not enough for large-scale quantum computation. While this distance can be improved by providing additional EPR resources, it is absolutely imperative that the technology continue to improve if computationally relevant quantum computing is to be feasible.

One method to alleviate the communication constraints is to consider a distributed quantum architecture [Cirac et al. 1999; Grover 1997; Lim et al. 2005]. Given that it has been shown that entanglement between remote atomic qubits (such as trapped ions) can be achieved through photon interactions [Cabrillo et al. 1999; Duan et al. 2004; Matsukevich and Kuzmich 2004; Spiller et al. 2005], one can imagine that the QLA architecture can be divided into a number of processing chips, where each chip resembles a smaller version of the QLA. Specifically for Shor's algorithm, the structure of the algorithm lends itself to distribution with minimal interchip communication and such organization has been explored in Yimsiriwattana and Lomonaco [2004]. In addition, one should not discard the possibility of designing a quantum architecture where communication and computation are not decoupled, and quantum gates are teleported rather than teleporting quantum data [Gottesman and Chuang 1999]. Van Meter et al. have calculated that when overall circuit depth is considered, teleporting data is cheaper then teleporting gates [Meter et al 2006]; however, the authors had neither considered overall reliability nor the cost associated with the teleportation of logical, rather than physical, gates.

Perhaps the most interesting achievement of the second conference paper on QLA architecture [Thaker et al. 2006] is reduction of the area requirements by a factor of nine, significantly reducing the overall distance the logical qubits must travel for each application being executed. We achieved the area reduction with minimal impact on the estimated performance by employing software-based specialization of the QLA system into memory and computational regions. This was done such that each region can be individually optimized to match hardware support to the available parallelism in the quantum circuits. A specialized architecture also offers a path the easing of management of the classical resources for quantum computers. By dividing the architecture into memory and computational regions, we can optimize and focus the resource consumption individually to each region for each application. For ion-trap quantum computation, one of the major bottlenecks is the required density of the classical circuitry to control the trapping electrodes and laser access to every region in the architecture [Kim et al. 2005]. An optimization technique governed by strict SIMD methodology is necessary to optimize the classical control and to allow the orchestration of tens of millions of physical ions.

In conclusion, the QLA architecture offers a design model that may conceivably allow implementation of the tens of millions of physical ions needed for the necessary quantum subroutines for large-scale applications, such as Shors factoring algorithm. Such performance assumes aggressive technology parameters which have not yet been demonstrated, but are believed to be within reach of present experimental techniques. The goal of the QLA architecture is to provide vital insight and motivation, from a systems-level perspective, to the physicists actually involved in building a large-scale quantum computer. For architects, the QLA forms a technically sound base that can be used to confidently study interesting issues in quantum architectures and to work towards more efficient, reliable, and scalable quantum computers.

REFERENCES

Aharonov, D. and Ben-Or, M. 1997. Fault tolerant computation with constant error. In *Proceedings the Annual ACM Symposium on Theory of Computing (STOC)*, 176–188.

Aharonov, D., van Dam, W., Kempe, J., Landau, Z., Lloyd, S., and Regev, O. 2004. Adiabatic quantum computation is equivalent to standard quantum computation. *SIAM J. Compute. 37*, 1, 164–196.

Aliferis, P., Gottesman, D., and Preskill, J. 2005. Quantum accuracy threshold for distance 3 codes. Eprint: quant-ph/0504218.

Bahr, F., Boehm, M., Franke, J., and Kleinjung, T. 2005. Rsa-640 is factored! RSA Lab, Bedford, MA.

Balensiefer, S., Kregor-Stickles, L., and Oskin, M. 2005. An evaluation framework and instruction set architecture for ion-trap based quantum micro-architectures. In *Proceedings the Annual International Symposium on Computer Architecture (ISCA)*, Madison, WI.

Barenco, A., Bennett, C. H., Cleve, R., DiVincenzo, D., Margolus, N., Shor, P., Sleator, T., Smolin, J., and Weinfurter, H. 1995. Elementary gates for quantum computation. *Phys. Rev. A. 52*, 3457.

Barrett, M., Chiaverini, J., Schaetz, T., Britton, J., Itano, W. M., Jost, J. D., Knill, E., Langer, C., Liebfried, D., Ozeri, R., and Wineland, D. J. 2004. Deterministic quantum teleportation of atomic qubits. *Nature 429*.

Bell, J. S. 1964. On the Einstein-Podolsky-Rosen paradox. *Phys. 1*, 195–200.

Bennett, C., Brassard, G., Popescu, S., Schumacher B., Smolin, J. A., and Wooters, W. K. 1996. Purification of noisy entanglement and faithful teleportation via, noisy channels. *Phys. Rev. Lett. 76*, 722.

Bennett, C. H., Barassard, G., Crépeau, C., Jozsa, R., Peres, A., and Wooters, W. K. 1993. Teleporting an unknown quantum state via dual classical and EPR channels. *Phys. Rev. Lett. 70*, 1895–1899.

Briegel, H. and Raussendorf, R. 2001. Persistent entanglement in arrays of interacting particles. *Phys. Rev. Lett 86*, 910–913.

Britton, J., Leibfried, D., Beall J., Blakestad, R. B., Bollinger, J. J., Chiaverini, J., Epstein, R. J., Jost, J. D., Kielpinski, D., Langer, C., Ozeri, R., Reichle, R., Seidelin, S., Shiga, N., Wesenberg, J. H., and Wineland, D. J. 2006. A microfabricated surface-electrode ion trap in silicon. http://arxiv.org/abs/quant=ph/0605170.

Buhler, J., Lenstra, H., and Pomerance, C. 1994. Factoring integers with the number field sieve. *In The Development of the Number Field Sieve.* Lecture Notes: in Mathematics, vol. 1554. Springer, 50–94.

Bullock, S. S. and Markov, I. L. 2004. Asymptotically optimal circuits for arbitrary n-qubit diagonal computations. *Quantum Inf. Comput. 4*, 1, 027–047.

Cabrillo, C., Cirac, J. I., Garcia Fernandez, P., and Roller, P. 1999. Creation of entangled states of distant atoms by interference. *Phys. Rev. A 59*, 1025–1033.

Chekuri, C., Johnson, R., Motwani, R., Natarajan, B., Rau, B., and Schlansker, M. 1996. Profile-Driven instruction level parallel scheduling with applications to superblocks. In *Proceedings of the 29th International Symposium on Microarchitecture 29*, 58–67.

Childs, A. M., Farhi, E., and Preskill, J. 2002. Robustness of adiabatic quantum computation. *Phys. Rev. A 65*.

Cirac, J., Ekert, A., Huelga, S., and Macchiavello, C. 1999. Distributed quantum computation over noisy channels. *Phys. Rev. A 59*, 4249.

Cirac, J. I. and Zoller, P. 1995. Quantum computations with cold trapped ions. *Phys. Rev. Lett. 74*, 4091–4094.

Deitrich, B. L. and Hwu, M. W. 1996. Speculative hedge: Regulating compile-time speculation against profile variations. In *Proceedings of the 29th International Symposium on Microarchitecture*.

Deutsch, D. 1985. Quantum computational networks. *Proc. Royal. Soc. London. A 400*, 97–117.

Deutsch, D., Ekert, A., Jozsa, R., Macchiavello, C., Popescu, S., and Sanpera, A. 1996. Quantum privacy amplification and the security of quantum cryptography over noisy channels. *Phys. Rev. Lett. 77*, 2818–2821.

DiVincenzo, D. P. 2000. The physical implementation of quantum computation. *Fortschr. Phys. 48*, 771–783.

Draper, T., Kutin, S., Rains, E., and Svore, K. 2004. A logarithmic-depth quantum carry-lookahead adder. E-Print: quant-ph/0406142.

DUAN, L., BLINOV, B., MOEHRING, D., AND MONROE, C. 2004. Scalable trapped ion quantum computation with a probabilistic ion-photon mapping. E-Print: quant-ph/0401020.

DUR, W., BRIEGEL, H. J., CIRAC, J. I., AND ZOLLER, P. 1999. Quantum repeaters based on entanglement purification. *Phys. Rev. A59*, 169.

FARHI, E., GOLDSTONE, J., GUTMANN, S., AND SIPSER, M. 2000. Quantum computation by adiabatic evolution. arXiv.org=quant-ph/0001106.

FOWLER, A., THOMPSON, W. F., YAN, Z., STEPHENS, A. M., PLOURDE, B., AND WILHELM, F. K. 2007. Long-Range coupling and scalable architecture for superconducting flux qubits. arXiv:cond-mat/0702620.

FREEDMAN, M., KITAEV, A., LARSEN, M., AND WANG, Z. 2003. Topological quantum computation. *Bull. Amer. Math. Soc. 40*, 3138.

GOTTESMAN, D. 2000. Fault tolerant quantum computation with local gates. *J. Modern Optics 47*, 333–345.

GOTTESMAN, D. 1998. Theory of fault-tolerant quantum computation. *Phys. Rev. A 57*, 127–137.

GOTTESMAN, D. K. AND CHUANG, I. L. 1999. Quantum teleportation is a universal computational primitive. *Nature* 402, 390–392.

GROVER, L. K. 1997. Quantum telecomputation. E-Print: http://arXiv.org/quant-ph/9704012.

HENSINGER, W. K., OLMSCHENK, S., STICK, D., HUCUL, D., YEO, M., ACTON, M., DESLAURIERS, L., RABCHUK, J., AND MONROE, C. 2005. T-Junction ion trap array for two-dimensional ion shuttling, storage and manipulation. E-Arxiv: quant-ph/0508097.

HIME, T., REICHARDT, P., PLOURDE, B., ROBERTSON, T., WU, C.-E., USTINOV, A., AND CLARKE, J. 2007. Solid-State qubits with current-controlled coupling. *Sci. 314*, 5804, 1427–1429.

HOLLENBERG, L. C. L., DZURAK, A. S., WELLARD, C., HAMILTON, A. R., REILLY, D. J., MILBURN, G. J., AND CLARK, R. 2003. Charge-Based quantum computing using single donors in semiconductors. *Phys. Rev. B 69*, 113301.

ISAILOVIC, N., PATEL, Y., WHITNEY, M., AND KUBIATOWICZ, J. 2006. Interconnection networks for scalable quantum computers. In *Proceedings of the 33rd Annual ACM International Symposium on Computer Architecture (ISCA)*, Boston, MA.

JONES, J., VEDRAL, V., EKERT, A., AND CASTAGNOLI, G. 2000. Geometric quantum computation using nuclear magnetic resonance. *Nature 403*, 869–871.

KANE, B. 1998. A silicon-based nuclear spin quantum computer. *Nature 393*, 133–137.

KIM, J., PAU, S., MA, Z., MCLELLAN, H., GATES, J., KORNBLIT, A., AND SLUSHER, R. 2005. System design for a large-scale ion-trap quantum information processor. *Quantum Inf. Comput. 5,* 7, 515.

KITAEV, A. Y. 1997. Quantum error correction with imperfect gates. In *Proceedings of the 3rd International Conference on Quantum Communication and Measurement*, 181–188.

KNILL, E. AND LAFLAMME, R. 1997. A theory of quantum error-correcting codes. *Phys. Rev. A 55*, 900–911.

KNILL, E., LAFLAMME, R., AND MILBURN, G. 2001. A scheme for efficient quantum computation with linear optics. *Nature 409*, 4652.

LANGER, C., OZERI, R., JOST, J. D., CHIAVERINI, J., DEMARCO, B., BEN-KISH, A., BLAKESTAD, R. B., BRITTON, J., HUME, D. B., ITANO, W. M., LIEBFRIED, D., REICHLE, R., ROSENBAND, T., SCHAETZ, T., SCHMIDT, P. O., AND WINELAND, D. J. 2005. Long-Lived qubit memory using atomic ions. E-Print: quant-ph/0504076.

LIM, Y. L., BARRETT, S. D., BEIGE, A., KOK, P., AND KWEK, L. C. 2005. Repeat-Until-Success quantum computing using stationary and flying qubits. E-Print: http://arXiv.org/quant-ph/0508218.

MAKHLIN, Y., SCHOEN, G., AND SHNIRMAN, A. 1999. Josephson-Junction qubits with controlled couplings. *Nature 398*, 305.

MATSUKEVICH, D. AND KUZMICH, A. 2004. Quantum state transfer between matter and light. *Sci. 306*, 5696, 663–666.

METODI, T. S., THAKER, D. D., CROSS, A. W., CHONG, F. T., AND CHUANG, I. L. 2005. A quantum logic array microarchitecture: Scalable quantum data movement and computation. In *Proceedings of the 38th International Symposium on Microarchitecture (MICRO)*.

METODI, T. S., THAKER, D. D., CROSS, A. W., CHONG, F. T., AND CHUANG, I. L. 2006. Physical operations scheduler in a quantum information processor. In *Proceedings of the SPIE Defense and Security Symposium,* Orlando, FL.

NIELSEN, M. 2004. Optical quantum computation using cluster states. *Phys. Rev. Lett. 93* (040503).

NIELSEN, M. A. AND CHUANG, I. L. 2000. *Quantum Computation and Quantum Information*. Cambridge University Press, Cambridge, UK.

NIELSEN, M. A. AND DAWSON, C. M. 2004. Fault-Tolerant quantum computation with cluster states.

NISKANEN, A., HARRABI, K., YOSHIHARA, F., NAKAMURA, Y., LLOYD, S., AND TSAI, J. 2007. Quantum coherent tunable coupling of superconducting qubits. *Sci. 316*, 5825, 723–726.

OSKIN, M., CHONG, F., AND CHUANG, I. 2002. A practical architecture for reliable quantum computers. *IEEE Comput.*

OZERI, R., LANGER, C., JOST, J. D., DE MARCO, B., BEN-KISH, A., BLAKESTAD, B. R., BRITTON, J., CHIAVERINI, J., ITAND, W. M., HUME, D. B., LEIBFRIED, D., ROSENBAND, T., SCHMIDT, P. O., AND WINELAND, D. J. 2004. Hyperfine coherence in the presence of spontaneous photon scattering. arXiv:quant-ph/0502063.

PEARSON, C. E., LEIBRANDT, D. R., BAKR, W. S., MALLARD, W. J., BROWN, K. R., AND CHUANG, I. L. 2006. Experimental investigation of planar ion traps. *Phys. Rev. A 73*, 032307.

PLATZMAN, P. M. AND DYKMAN, M. I. 1999. Computing with electrons floating on liquid helium. *Sci. 284*, 1967–1969.

REICHARDT, B. W. 2004. Improved ancilla preparation scheme increases fault-tolearant threshold. E-Print: quant-ph/0406025.

REICHLE R., LEIBFRIED, D., KNILL, E., BRITTON, J., BLAKESTAD, R., JOST, J., LANGER, C., OZERI, R., SEIDELIN, S., AND WINELAND, D. 2006. Experimental purification of two-atom entanglement. *Nature 443*, 19, 838–841.

RIEBE, M., HÄFFNER, H., ROOS, C., HÄNSEL, W., BENHELN J., KÖRBER, T. W., BECHER, C., SCHMID-KALER, F., AND BLATT, R. 2004. Deterministic quantum teleportation with atoms. *Nature 429*, 6993, 734–737.

SEIDELIN, S., CHIAVERINI, J., REICHLE, R., BOLLINGER, J., LEIBFRIED, D., BRITTON, J., WESENBERG, J. H., BLAKESTAD, R. B., EPSTEIN, R. J., HUME, D. , ITANO, W. M., JOST, J. D., LANGER, C., OZERI, R., SHIGA, N., AND WINELAND, D. J. 2006. A microfabricated surface-electrode ion trap for scalable quantum information processing. ArXiv Quantum Physics e-prints.

SHENDE, V., BULLOCK, S., AND MARKOV, I. 2006. Synthesis of quantum logic circuits. *IEEE Trans. Comput.-Aided Des. 25*, 6, 1000–1010.

SHENDE, V. V., MARKOV, I. L., AND BULLOCK, S. S. 2003. Minimal universal two-qubit quantum circuits. *Phys. Rev. A 69*, 062321, 1–7.

SHENDE, V. V., MARKOV, I. L., AND BULLOCK, S. S. 2004. Finding small two-qubit circuits. In *Proceedings of the SPIE*, vol. 5436, 348–359.

SHOBAKI, G. AND WILKEN, K. 2004. Optimal superblock scheduling using enumeration. In *Proceedings of the 37th International Symposium on Microarchitecture (MICRO)*.

SHOR, P. W. 1995. Scheme for reducing decoherence in quantum computer memory. *Phys. Rev. A 54*, 2493.

SKINNER, A., DAVENPORT, M., AND KANE, B. 2003. Hydrogenic spin quantum computing in silicon: A digital approach. *Phys. Rev. L 90*, 087901 (Feb.).

SPILLER, T., NEMOTO, K., BRAUNSTEIN, S., MUNRO, W., VAN LOOCK, P., AND MILBURN, G. 2005. Quantum computation by communication. http://arxiv.org/abs/quant-ph/0509202.

STEANE, A. 1996. Error correcting codes in quantum theory. *Phys. Rev. Lett. 77*, 793–797.

STEANE, A. 2004. How to build a 300 bit, 1 GOP quantum computer. arXiv:quant-ph/0412165.

SVORE, K., TERHAL, B., AND DIVINCENZO, D. 2004. Local fault-tolerant quantum computation. E-Print: quant-ph/0410047.

SVORE, K. M., DIVINCENZO, D. P., AND TERHAL, B. M. 2006. Noise threshold for a fault-tolerant two-dimensional lattice architecture. E-Print (Arxiv.org): quant-ph/0604090.

THAKER, D. D., METODI, T. S., CROSS, A. W., CHONG, F. T., AND CHUANG, I. L. 2006. Quantum memory hierarchies: Efficient designs to match available parallelism in quantum computing. In *Proceedings of the 33rd Annual ACM International Symposium of Computer Architecture (ISCA),* Boston, MA.

VAN METER, R., NEMOTO, K., MUNRO, W. J., AND ITOH, K. M. 2006. Distributed arithmetic on a quantum multi-computer. In *Proceedings of the 33rd Annual ACM International Symposium of Computer Architecture (ISCA)*, Boston, MA.

Van Meter, R. and Itoh, K. M. 2004. Fast quantum modular exponentiation. E-Print: quant-ph/0408006.

Van Meter, R. and Oskin, M. 2006. Architectural implications of quantum computing technologies. *ACM J. VAN Emerging Technol. Comput. Syst. 2*, 1.

Wineland, D., Monroe, C., Itano, W., Leibfried, D., King, B., and Meekhof, D. 1998. Experimental issues in coherent quantum-state manipulation of trapped atomic ions. *J. Res. NIST 103*, 259–328.

Wineland, D. and Heinrichs, T. 2004. Ion trap approaches to quantum information processing and quantum computing. *A Quantum Information Science and Technology Roadmap*. url: http://quist.lanl.gov.

Wineland, D., Leibfried, D., Barrett, M., Ben-Kish, A., Bergquist, J. C., Blakestad, R. B., Bollinger, J. J., Britton, J., Chaverini, B., DeMarco, B., Hume, D., Itano, W. M., Jensen, M., Jost, J. D., Knill, E., Koelemeij, J., Langer, C., Oskay, W., Ozeri, R., Reichle, R., Roseband, T., Schaetz, T., Schmidt, P. O., and Seideling, S. 2005. Quantum control, quantum information processing, and quantum-limited metrology with trapped ions. In *Proceedings of the International Conference on Laser Spectroscopy (ICOLS)*.

Wootters, W. and Zurek, W. 1982. A single quantum cannot be cloned. *Nature 299*, 802–803.

Yimsiriwattana, A. and Lomonaco, S. J. 2004. Distributed quantum computing: A distributed Shor algorithm. E-Print: arXiv.org:quant-ph/0403146.

Zeng, B., Zhou, D., Xu, Z., and Sun, C. 2003. Quantum teleportation using cluster states. ArXiv Quantum Physics e-prints.

Received November 2007; accepted November 2007 by Vijay Krishnan Narayanan