Distributed Arithmetic on a Quantum Multicomputer

Rodney Van Meter
Keio University and CREST-JST
3-14-1 Hiyoushi, Kohoku-ku
Yokohama-shi, Kanagawa 223-8522, Japan
rdv@tera.ics.keio.ac.jp

W. J. Munro
Hewlett-Packard Laboratories
Filton Road, Stoke Gifford
Bristol BS34 8QZ, United Kingdom
bill.munro@hp.com

Kae Nemoto
National Institute of Informatics
2-1-2 Hitotsubashi, Chiyoda-ku
Tokyo 101-8430, Japan
nemoto@nii.ac.jp

Kohei M. Itoh
Keio University and CREST-JST
3-14-1 Hiyoushi, Kohoku-ku
Yokohama-shi, Kanagawa 223-8522, Japan
kitoh@appi.keio.ac.jp

Abstract

We evaluate the performance of quantum arithmetic algorithms run on a distributed quantum computer (a quantum multicomputer). We vary the node capacity and I/O capabilities, and the network topology. The tradeoff of choosing between gates executed remotely, through "teleported gates" on entangled pairs of qubits (telegate), versus exchanging the relevant qubits via quantum teleportation, then executing the algorithm using local gates (teledata), is examined. We show that the teledata approach performs better, and that carry-ripple adders perform well when the teleportation block is decomposed so that the key quantum operations can be parallelized. A node size of only a few logical qubits performs adequately provided that the nodes have two transceiver qubits. A linear network topology performs acceptably for a broad range of system sizes and performance parameters. We therefore recommend pursuing small, high-I/O bandwidth nodes and a simple network. Such a machine will run Shor's algorithm for factoring large numbers efficiently.

1 Introduction

We are investigating the design of a *quantum multicomputer*, a machine consisting of many small quantum computers connected together to cooperatively solve a single problem. Such a system may overcome the limited capacity of quantum computing technologies expected to be available in the near term, scaling to levels which dramatically outperform classical computers on some problems [42, 44, 23, 16].

The first question in considering a multicomputer is whether the system performance will be acceptable *if* the implementation problems can be solved. We focus on distributed implementation of three types of arithmetic circuits derived from known classical adder circuits [57, 15, 18, 19]. For many algorithms, notably Shor's algorithm for factoring large numbers, arithmetic is an important component, and integer addition is at its core [44, 54]. Our evaluation criterion is the latency to complete the addition. The goal is to achieve "reasonable" performance for Shor's factoring algorithm for numbers up to a thousand bits.

Our distributed quantum computer creates a shared quantum state between the separate nodes of our machine. As we perform our computation, this quantum state evolves and we are dependent on either quantum teleportation of quantum data (called *qubits*), or teleportation-based remote execution of quantum gates [8, 22]; we present teleportation in more detail in section 2.2, and discuss the tradeoff throughout this paper.

The nodes of the machine may be connected in a variety of topologies, which will influence the efficiency of the algorithm. We concentrate on only three topologies (shared bus, line, and fully connected) and two additional variants (2bus, 2fully), constraining our engineering design space and deferring more complex topology analysis for future work. Our analysis is done attempting to minimize the required number of qubits in a node while retaining reasonable performance; we investigate node sizes of one to five logical qubits per node.

In this research we show that:

- teleporting data is better than teleporting gates;

Proceedings of the 33rd International Symposium on Computer Architecture (ISCA'06)
0-7695-2608-X/06 $20.00 © 2006 IEEE

- decomposition of teleportation brings big benefits in performance, making a carry-ripple adder effective even for large problems;
- a linear topology is an adequate network for the foreseeable future; and
- small nodes (only a few logical qubits) perform acceptably, but I/O bandwidth is critical.

A multicomputer built around these principles and based on solid-state qubit technology will perform well on Shor's algorithm. These results collectively represent a large step in the design and performance analysis of distributed quantum computation.

We begin at the foundations, including related work and definitions of some of the terms we have used in this introduction. Next, we discuss our node and interconnect architectures, followed by mapping the arithmetic algorithms to our system. Performance estimates are progressively refined, including showing how decomposing the teleportation operation makes the performance of the CDKM carry-ripple adder competitive with the nominally faster carry-lookahead adder. We conclude with specific recommendations for a medium-term goal of a modest-size quantum multicomputer.

2 Foundations

A quantum computer is a machine that uses quantum mechanical effects to achieve potentially large reductions in the computational complexity of certain tasks [42, 44, 23, 16]. Quantum computers exist, but are slow, very small (consisting of only a few quantum bits, or qubits), not reliable, and have very limited scalability [56, 25]. True architectural research for a large-scale quantum computer can be said to have only just begun [55, 43, 14, 29, 49, 31, 4, 53, 54].

Classically, the best known algorithm for factoring large numbers is $O(e^{(nk\log^2 n)^{1/3}})$, where n is the length of the number, in bits, and $k = \frac{64}{9}\log 2$, whereas Shor's quantum factoring algorithm is polynomial ($O(n^3)$ or better) [32, 44, 54]. These gains are achieved by taking advantage of *superposition* (a quantum being in a weighted combination of states, rather than the single state that is possible classically), *entanglement* (loosely speaking, the state of two quanta not being independent), and *interference* of the quantum wave functions (analogous to interference in classical wave mechanics). Of these, only entanglement of pairs of qubits, as the core of quantum teleportation, is directly relevant to this paper. Otherwise, only a limited familiarity with quantum computing is required to understand this paper, and we introduce the necessary terminology and background in this section. Readers interested in more depth are referred to popular [41, 59] and technical [42, 46] texts on the subject.

Teleportation of quantum states (qubits, or quantum data) has been known for more than a decade [8]. It has been demonstrated experimentally [21, 9], and has been suggested as being necessary for moving data long distances within a single quantum computer [43]. Teleportation consumes Einstein-Podolsky-Rosen pairs, or *EPR pairs*. EPR pairs are pairs of particles or qubits which are *entangled* so that actions on one affect the state of the other. EPR pairs can be created in a variety of ways, including reactions that simultaneously emit pairs of photons whose characteristics are related and many quantum gates on two qubits. Entanglement is a continuous, not discrete, phenomenon, and several weakly entangled pairs can be used to make one strongly entangled pair using a process known as *purification* [12].

Quantum bits, or qubits, have two basis states, corresponding to the zero and one states of a classical bit. These two states are written using Dirac's *ket* notation in the form $|0\rangle$ and $|1\rangle$.

2.1 Qubus Entanglement Protocol

Our approach to creating EPR pairs contains no direct qubit-qubit interactions and does not require the use of single photons, instead using laser or microwave pulses as a probe beam [40, 38]. Two qubits are entangled indirectly through the interaction of the qubits with a common quantum field mode created by the probe beam – a continuous quantum variable – which can be thought of as a quantum communication bus, or "qubus" [47]. We call this process the qubus entanglement protocol, or *QEP*.

By interacting the probe beam with the qubit, the probe beam picks up a θ phase shift if it is in one basis state (e.g., $|0\rangle$) and a $-\theta$ phase shift if it is in the other (e.g., $|1\rangle$). If the same probe beam interacts with two qubits, it is straightforward to see that the probe beam acting on the two-qubit states $|0\rangle|1\rangle$ and $|1\rangle|0\rangle$ picks up no net phase shift because the opposite-sign shifts cancel, while the probe beam acting on the states $|0\rangle|0\rangle$ and $|1\rangle|1\rangle$ picks up phase shift $\pm 2\theta$. An appropiate measurement determines whether the probe beam has been phase shifted (in effect taking the absolute value of the shift), projecting the qubits into either an even parity state or an odd parity state. The measurement shows only the parity of the qubits, not the actual values, leaving them in an entangled state. This state can be then used as our EPR pair.

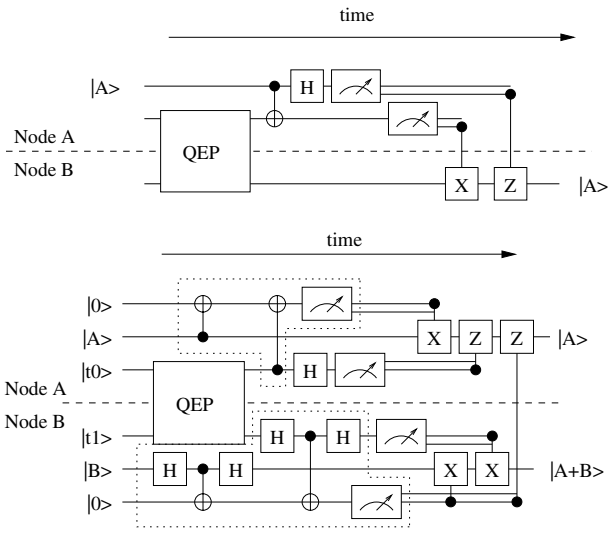

Figure 1. A teleportation circuit (top) and teleported control-NOT (CNOT) gate (bottom). Time flows left to right, each horizontal line represents a qubit, and each vertical line segment with terminals is a quantum gate. A segment with a $\oplus$ terminal is a control-NOT (CNOT) gate. The "meter" box is measurement of a qubit's state, and the double line extending from it is the classical result of that measurement. The boxes with H, X, and Z in them are various qubit gates. The large box labeled QEP is the qubus EPR pair generator. (See Nielsen and Chuang for more details on the notation [42].)

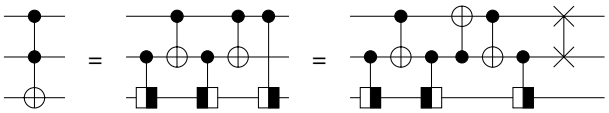

Figure 2. CCNOT (control-control-NOT, or Toffoli) gate constructions for our architectures.The leftmost object is the canonical representation of this three-qubit gate. The rightmost construction we use for the line topology; the middle construction we use for all other topologies. The box with the bar on the right represents the square root of X (NOT) (half a NOT gate, effectively), and the box with the bar on the left its adjoint. The last gate in the rightmost construction is a SWAP gate, which exchanges the state of two qubits.

2.2 Teleporting Gates and Teleporting Data

To teleport a qubit, one member of the EPR pair is held by the teleportation sender, and the other by the teleportation receiver. The qubit to be teleported is entangled with the local EPR member, then both of those are *measured*, which will return a classical 0 or 1 for each qubit and destroy the qubit. The results of this measurement are transmitted to the receiver, which then executes gates locally on its member of the EPR pair, conditional on the measurement results, recreating the (now destroyed) original state at the destination. The circuit for teleportation is shown in figure 1.

Gottesman and Chuang showed that teleportation can be used to construct a control-NOT (CNOT) gate [22]. Their original teleported gate requires two EPR pairs. We use an approach based on parity gates that consumes only one EPR pair, as shown in the bottom half of figure 1 [38]. Locally, the parity gates can be implemented with two CNOT gates and a measurement (outlined with dotted lines in the figure). Double lines are classical values that are the output of the measurements; when used as a control line, we decide classically whether or not to execute the quantum gate, based on the measurement value. The last gate involves classical communication of the measurement result between nodes (where the vertical line segment crosses the dashed line between nodes A and B). As shown, this construction is not fault tolerant; it must be built over fault-tolerant gates. Alternatively, the qubus approach can be used as the node-internal interconnect. Its natural gate is the parity gate, and is fault tolerant.

In designing algorithms for our quantum multicomputer, therefore, we have a choice: when two qubits in different nodes of our multicomputer are required to interact, we can either move data (qubits) from one node to another, then perform the shared gate, or we can use a teleported gate directly on the qubits, without moving them. We will call the data-moving approach *teledata* and the teleportation-based gate approach *telegate*.

For some algorithms, we can use a simple, visual approach to counting the number of remote operations necessary to execute the algorithm using either the teledata or telegate approach (see section 4). For most quantum computing technologies, the three-qubit Toffoli (control-control-NOT) gate must be constructed from two-qubit gates, as shown in figure 2. For the telegate approach, we assign a cost of three to each two-node Toffoli gate, and each three-node Toffoli gate we count as five [54]. The three-node Toffoli gate should cost more, but pipelining of operations across multiple nodes hides the additional latency. We assign two-node CNOT gates a cost of one.[1]

[1]There are other possible compositions of the Toffoli gate, but the dif-

2.3 Distributed Quantum Computation

Early suggestions of distributed quantum computation include Grover [24], Cirac *et al.* [12], and Steane and Lucas [50]. A recent paper has proposed combining the cluster state model with distributed computation [33]. Such a distributed system generally requires the capability of transferring qubit state from one physical representation to another, such as nuclear spin ↔ electron spin ↔ photon [37, 26, 11].

Yepez distinguished between distributed computation using entanglement between nodes, which he called type I, and without inter-node entanglement (i.e., classical communication only), which he called type II [61]. Our quantum multicomputer is a type I quantum computer. Jozsa and Linden showed that Shor's algorithm requires entanglement across the full set of qubits, so a type II quantum computer cannot achieve exponential speedup [28, 34]. Much of the work on our multicomputer involves creation and management of that shared entanglement.

Yimsiriwattana and Lomonaco have discussed a distributed version of Shor's algorithm [62], based on one form of the Beckman-Chari-Devabhaktuni-Preskill modular exponentiation algorithm [6]. The form they use depends on complex individual gates, with many control variables, inducing a large performance penalty compared to using only two- and three-qubit gates. Their approach is similar to our telegate (sec. 2.2), which we show to be slower than teledata. They do not consider differences in network topology, and analyze only circuit complexity, not depth (time performance), whereas our focus is on circuit depth.

3 Node and Interconnect Architecture

A multicomputer [3] is a constrained form of distributed system. All parts of the system are geographically colocated. Short travel distances (up to a few tens of meters) between nodes reduce latency, simplify coordinated control of the system, and increase signal fidelity. We assume a regular network topology, a dedicated network environment, and scalability to thousands of nodes. We concentrate on a homogeneous node technology based on solid-state qubits, with a qubus interconnect, though our results apply to essentially any choice of node and interconnect technologies, such as ion trap nodes and single photon-based qubit transfer [29, 53, 13, 58, 36].

Future, larger quantum computers will be built on technologies that are inherently limited in the number of qubits that can be incorporated into a single device [42, 46, 1]. The causes of these limitations vary with the specific technology, and in most cases are poorly understood, but may range from the low tens to perhaps thousands; integration of the densities we are accustomed to in the classical world is not even being seriously discussed for most technologies. Quantum error correction naturally reduces the number of available logical (application-level) qubits by a large factor [45, 10, 48]. As with classical error correction codes, multiple levels of error correction are possible, and often required. Two levels of the Steane 7-qubit code, for example, which encodes a single logical qubit in seven lower-layer qubits, would impose a 49:1 encoding and storage penalty. Therefore, it makes sense to examine the utility of a device that can hold only a few logical qubits.

We choose a node technology based on solid-state qubits, such as Josephson-junction superconducting qubits [39, 58, 27] or quantum dots [20], which will require a microwave qubus. Each node has many qubits which are private to the node, and a few *transceiver qubits* that can communicate with the outside world. Node size is limited by the number of elements that can practically be built into a single device, including control structures, external signalling, packaging, cooling, and shielding constraints.

Throughout this paper, qubits and operations on them are understood to be logical; whether the physical interconnect links and transceiver qubits are parallel or serial remains to be determined, in part by the demands of quantum error correction. Although the QEP protocol in theory supports EPR pair creation over many kilometers, our design goal is a scalable quantum computer in one location (such as a single lab). We consider a 10nsec classical communication latency, corresponding roughly to 2 meters between nodes. The performance figures we find are insensitive to this number.

We consider five interconnect networks: shared bus, line of nodes, fully connected, two-transceiver bus (2bus), and two-transceiver fully connected (2fully) as in figure 3. For the shared bus, all nodes are connected to a single bus. Any two nodes may use the bus to communicate, but it supports only a single transaction at a time. In the line topology, each node uses two transceiver qubits, one to connect to its left-hand neighbor and one to connect to its right-hand neighbor. Each link operates independently, and all links can be utilized at the same time, depending on the algorithm; multi-hop transfers are accomplished via store and forward. For the fully-connected network, each node has a single transceiver qubit which can connect to any other node without penalty via some form of classical switched network, though of course each transceiver qubit can be involved in only one transaction at a time. 2bus and 2fully utilize two transceiver qubits per node for concurrent transfers.

The effective topology may be different from the physical topology, depending on the details of a bus transaction. For example, even if the physical topology is a bus, the sys-

ferences are less than a factor of two, and which approach is best will depend on the choice of quantum error correction (QEC), as some are more difficult to implement on encoded qubits [5, 17].

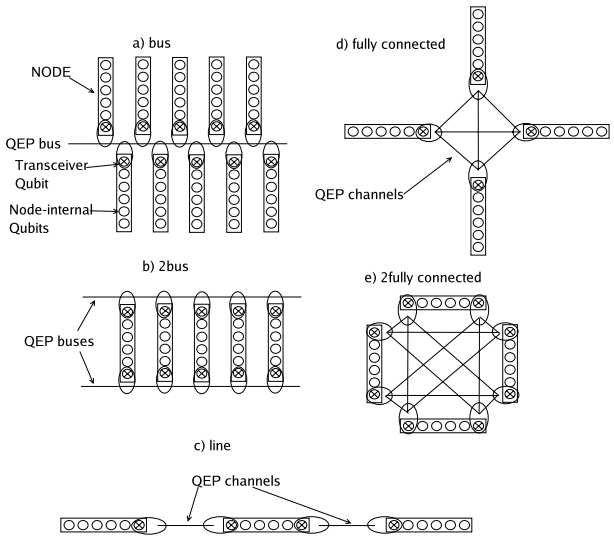

Figure 3. The five physical topologies analyzed in this paper.

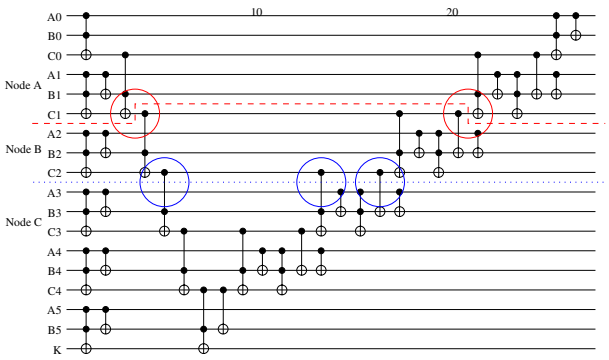

Figure 5. Visual approach to determining relative cost of teleporting data versus teleporting gates for a VBE adder. The upper, dashed (red) line shows the division between two nodes (A and B) using data teleportation. The circles show where the algorithm will need to teleport data. The lower, dotted line (blue) shows the division using gate teleportation (nodes B and C). The circles show where teleported gates must occur. Note that two of these three are CCNOT gates, which may entail multiple two-qubit gates in actual implementation. The numbers at the top are clock cycles.

tem may behave as if it is fully connected if the actions *internal* to a node to complete a bus transaction are much longer than the activities on the bus itself, allowing the bus to be reallocated quickly to another transaction. Some technologies may support frequency division multiplexing on the bus, allowing multiple concurrent transactions.

4 Algorithm

We evaluate three different addition algorithms: the Vedral-Barenco-Ekert (VBE) style of carry-ripple adder, which was the first type of quantum adder described [57], the faster, smaller Cuccaro-Draper-Kutin-Moulton (CDKM) carry-ripple adder [15], and the carry-lookahead adder [18]. Both carry-ripple and carry-lookahead adders are $O(n)$ in complexity to add two n-bit numbers, but they differ in their circuit depth, or latency to complete the addition. Carry-ripple is $O(n)$ latency, while carry-lookahead is $O(\log n)$. In this section we discuss the adders without regard to the network topology; the following section presents numeric values for different topologies and gate timings.

4.1 Carry-Ripple Adders

Figure 4 shows a two-qubit VBE carry-ripple adder [57] in its monolithic (left) and distributed (right) forms. The QEP block creates an EPR pair. The dashed boxes delineate the teleportation circuit, which is assumed to be perfect. The first teleportation moves the qubit $c0$ from node A to node B. $c0$ is used in computation at node B, then moved back to node A via a similar teleportation to complete the computation. The two qubits $t0$ and $t1$ are used as transceiver qubits, and are reinitialized as part of the QEP subcircuit.

Figure 5 shows a larger VBE adder circuit and illustrates a visual method for comparing telegate and teledata. For telegate, we can draw a line across the circuit, with the number of gates (vertical line segments) crossed showing our cost. For teledata, the line must *not* cross gates, instead crossing the qubit lines. The number of such crossings is the number of teleportations required. This approach works well for analyzing the VBE and CDKM adders, but care must be taken with the carry-lookahead adder, because it uses long-distance gates that may be between e.g. nodes 1 and 3.

The VBE adder latency to add two numbers on an m-node machine using the teledata method is $2m - 2$ teleportations plus the circuit cost. For the telegate approach, we must use a five-gate breakdown for CCNOT, *requiring three teleported two-qubit gates to form a CCNOT*. Therefore, implementing telegate, the latency is $7m - 7$ gate teleportations, or 3.5x the cost.

For the CDKM carry-ripple adder [15], which more aggressively reuses data space, teledata requires a minimum of six movements, whereas telegate requires two CCNOTs and three CNOTs, or a total of nine two-qubit gates, as shown

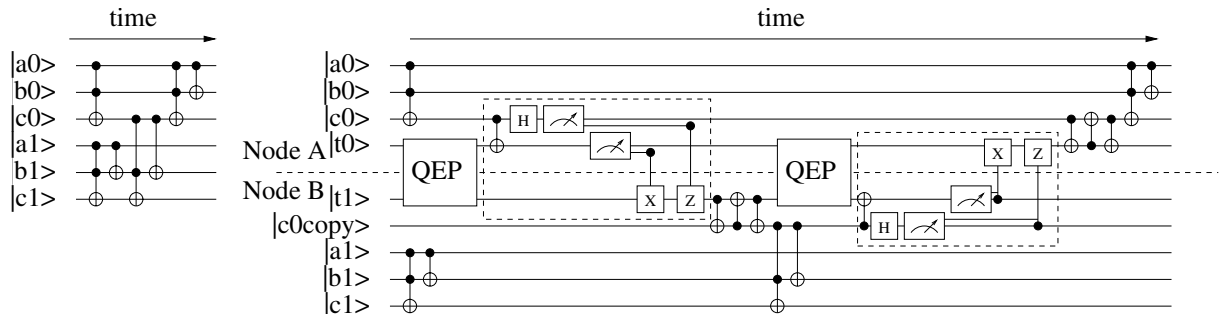

Figure 4. Details of a distributed 2-qubit VBE adder. The right-hand circuit is the distributed form using the teledata method; the left-hand circuit is the monolithic equivalent. The solid box (QEP) is the qubus EPR pair generator; the circuits in dashed boxes are standard quantum teleportation circuits.

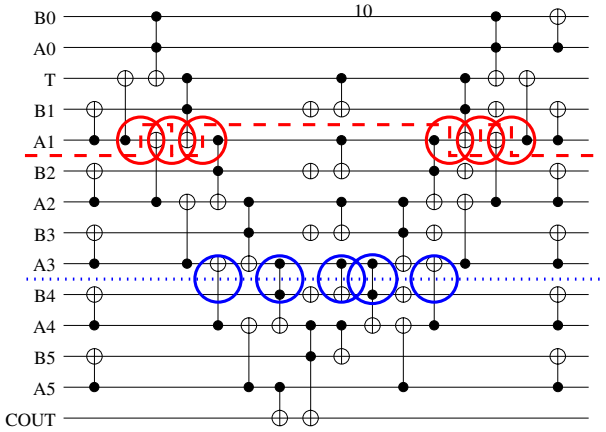

Figure 6. Visual approach to determining relative cost of teleporting data versus teleporting gates for a CDKM adder. Note that two of the five teleported gates are CCNOT gates, which may entail multiple two-qubit gates in actual implementation.

in figure 6. The CDKM adder pipelines extremely well, so the actual latency penalty for more than two nodes is only $2m + 2$ data teleportations, or $6m$ gate teleportations, when there is no contention for the inter-node links, as in our line and fully-connected topologies. The bus topology performance is limited by contention for access to the interconnect.

4.2 Carry Lookahead

Analyzing the carry-lookahead adder is more complex, as its structure is not regular, but grows more intertwined toward the middle bits. Gate scheduling is also variable, and the required concurrency level is high. The latency is $O(\log n)$, making it one of the fastest forms of adder for large numbers [18, 54, 19].

Let us look at the performance in a monolithic quantum computer, for n a power of two. Based on table 1 from Draper *et al.* [18], for $n = 2^k$, the circuit depth of $4k + 3$ Toffoli gates is 19, 31, and 43 Toffoli gates, for 16, 128, and 1,024 bits, respectively. We assume a straightforward mapping of the circuit to the distributed architecture. We assign most nodes four logical qubits (A_i, B_i, C_i, and one temporary qubit used as part of the carry propagation). In the next section, we see that the transceiver qubits are the bottleneck; we cannot actually achieve this $4k + 3$ latency.

5 Performance

The modular exponentiation to run Shor's factoring algorithm on a 1,024-bit number requires approximately 2.8 million calls to the integer adder [54]. With a 100 μsec adder, that will require about five minutes; with a 1 msec adder, it will take under an hour. Even a system two to three orders of magnitude slower than this will have attractive performance, provided that error correction can sustain the system state for that long, and that the system can be built and operated economically. This section presents numerical estimates of performance which show that this criterion is easily met by a quantum multicomputer under a variety of assumptions about logical operation times, providing plenty of headroom for quantum error correction.

5.1 Initial Estimate

Our initial results are shown in table 1. Units are in number of complete teleportations, treating teleportation and EPR pair generation as a single block, and assuming zero cost for local gates. In the following subsections

algo.	size	Baseline			Telegate					Teledata				
		bus	line	fully	bus	2bus	line	fully	2F	bus	2bus	line	fully	2F
VBE	16	360	305	182	105	105	105	105	105	30	30	30	30	30
	128	3048	2545	1526	889	889	889	889	889	254	254	254	254	254
	1024	24552	20465	12278	7161	7161	7161	7161	7161	2046	2046	2046	2046	2046
CDKM	16	232	160	160	138	96	96	97	96	90	60	34	90	34
	128	1912	1280	1280	1146	768	768	768	768	762	508	258	762	258
	1024	15352	10240	10240	9210	6144	6144	6145	6144	6138	4092	2050	6138	2050
Carry-	16	644	N/A	99	444	222	N/A	136	135	260	178	N/A	96	56
look-	128	6557	N/A	159	4901	2451	N/A	256	255	3176	2028	N/A	192	104
ahead	1024	54806	N/A	219	41502	20751	N/A	376	375	27260	17206	N/A	288	152

Table 1. Estimate of latency necessary to execute various adder circuits on different topologies of quantum multicomputer, assuming monolithic teleportation blocks (Sec. 5.1). Units are in number of teleportation blocks, including EPR pair creation (bus transaction), local gates and classical communication. Size, length of the numbers to be added, in bits. Lower numbers are faster (better).

these assumptions are revisited. We show three approaches (baseline, telegate, and teledata) and three adder algorithms (VBE, CDKM, and carry-lookahead) for five networks (bus, 2bus, line, fully, 2fully) and three problem sizes (16, 128, and 1024 bits). In the baseline case, each node contains only a single logical qubit; gates are therefore executed using the telegate approach. For each algorithm we need one node per qubit. VBE requires $3n$ nodes and CDKM $2n + 2$ nodes. Carry-lookahead requires 59, 504, and 4,085 nodes, respectively, for 16, 128, and 1,024-bit adders. For the telegate and teledata columns, we chose node sizes and number of nodes to suit the algorithms. For telegate, VBE uses n three-qubit nodes; CDKM uses $n + 1$ two-qubit nodes; and carry-lookahead uses n four-qubit nodes. For teledata, the same number of nodes is required, but each node must hold one more logical qubit.

The VBE adder, although larger and slower than CDKM on a monolithic computer, is faster in a distributed environment. The VBE adder exhibits a large (3.5x) performance gain by using the teledata method instead of telegate. For teledata, the performance is independent of the network topology, because only a single operation is required at a time, moving a qubit to a neighboring node. The CDKM adder also communicates only with nearest neighbors, but performs more transfers. The single bus configuration is almost 3x slower than the line topology. On a line, in most time slots, three concurrent transfers are conducted (e.g., between nodes $1 \rightarrow 2$, $3 \rightarrow 2$, and $3 \rightarrow 4$).

An unanticipated but intuitive result is that the performance of the carry-lookahead adder is better in the baseline case than the telegate case, for the fully-connected network. This is due to the limitation of having a single transceiver qubit per node. Putting more qubits in a node increases contention for the transceiver qubit, and reduces performance even though the absolute number of gates that must be executed via teleportation has been reduced. The carry-lookahead adder is easily seen to be inappropriate for the line architecture, since the carry-lookahead requires the use of long-distance gates in order to propagate carry information quickly. Our numbers also show that the carry-lookahead adder is not a good match for a bus architecture, despite the favorable long-distance transport, again because of excessive contention for the bus.

For the telegate carry-lookahead, performing some adjustments to eliminate intra-node gates, we find $8n - 9k - 8$ total Toffoli gates that need arguments that are originally stored on three separate nodes, plus $n-2$ two-node CNOTs. For the bus case, which allows no concurrency, this is our final cost. For the fully-connected network, we find a depth of $8k - 10$ three-node CCNOTs, 8 two-node CCNOTs, and 1 CNOT. These must be multiplied by the appropriate CCNOT breakdown. The fully and 2fully perform similarly, with the algorithm unable to take advantage of the availability of extra interconnect bandwidth when using the telegate method.

For the teledata, carry-lookahead, fully-connected case, each three-node Toffoli gate requires four teleportations (in and out for each of two variables). For the 2fully network, the latency of the three-node Toffolis is halved, but the two-node Toffolis do not benefit, giving us a final cost of slightly over half the fully network cost.

5.2 Improved Performance

The analysis in section 5.1 assumed that a teleportation operation is a monolithic unit. However, figure 4 makes it clear that a teleportation actually consists of several phases. The first portion is the creation of the entangled EPR pair. The second portion is local computation and measurement

algo.	size	Baseline			Telegate					Teledata				
		bus	line	fully	bus	2bus	line	fully	2F	bus	2bus	line	fully	2F
VBE	16	360	16	16	105	53	7	14	7	30	15	2	4	2
	128	3048	16	16	889	445	7	14	7	254	127	2	4	2
	1024	24552	16	16	7161	3581	7	14	7	2046	1023	2	4	2
CDKM	16	232	21	19	135	68	11	18	9	90	60	6	12	6
	128	1912	21	19	1146	573	11	18	9	762	508	6	12	6
	1024	15352	21	19	9210	4605	11	18	9	6138	4092	6	12	6
Carry-	16	644	N/A	99	444	222	N/A	89	45	260	178	N/A	96	56
look-	128	6557	N/A	159	4901	2451	N/A	149	75	3176	2028	N/A	192	104
ahead	1024	54806	N/A	219	41502	20751	N/A	209	105	27260	17206	N/A	288	152

Table 2. Estimated latency to execute various adders on different topologies, for decomposed teleportation blocks (sec. 5.2), assuming classical communication and local gates have zero cost. Units are in EPR pair creation times. 2F, 2fully.

at the sending node, followed by classical communication between nodes, then local operations at the receiving node. The EPR pair creation is not data-dependent; it can be done in advance, as resources (bus time slots, qubits) become available, for both telegate and teledata.

Our initial execution time model treats local gates and classical communication as free, assuming that EPR pair creation is the most expensive portion of the computation. For example, for the teledata VBE adder on a linear topology, all of the EPR pairs needed can be created in two time steps at the beginning of the computation. The execution time would therefore be 2, constant for all n. Table 2 shows the performance under this assumption. The performance of the carry-lookahead adder does not change, as the bottleneck link is busy full-time creating EPR pairs.

This model gives a misleading picture of performance once EPR pair creation is decoupled from the teleportation sequence. When the cost of the teleportation itself or of local gates exceeds $\sim 1/n$ of the cost of the EPR pair generation, the simplistic model breaks down; in the next subsection, we examine the performance with a more realistic model.

5.3 Detailed Estimate

To create figures 7-9, we make assumptions about the execution time of various operations. Classical communication between nodes is 10nsec. A CCNOT (Toffoli) gate on encoded qubits takes 50nsec, CNOT 10nsec, and NOT 1nsec. These numbers can be considered realistic but optimistic for a technology with physical gate times in the low nanoseconds; for quantum error correction-encoded solid-state systems, the bottleneck is likely to be the time for qubit initialization or reliable single-shot measurement, which is still being designed (see the references in [55]).

Figures 7 and 8 show, top to bottom, the 2fully and line networks for the telegate and teledata methods. We plot adder time against EPR pair creation time and the length of the numbers to be added. The left hand plot shows the shape of the surfaces, with the z axis being latency to complete the addition. The right hand plot, with the same x and y axes, shows the regions in which each type of adder is the fastest. The hatched red area indicates areas where carry-lookahead is the fastest, the diagonally lined green area indicates CDKM carry-ripple, and solid blue indicates VBE carry-ripple.

We vary the EPR pair creation time from 10nsec to 1280nsec. This creation process is influenced by the choice of parallel or serial bus and the cycle time of an optical homodyne detector, repeated as necessary for entanglement purification [12, 7]. Photodetectors may be inherently fast, but their performance is limited by surrounding electronics [2, 51]. Final performance may be faster or slower than our model, but the range of values we have analyzed is broad enough to demonstrate clearly the important trends.

These figures show that the teledata method is faster than telegate. They also show that the carry-lookahead adder is very dependent on EPR pair creation time, while neither carry-ripple adder is. If EPR pair creation time is low, the carry-lookahead adder is very fast; if creation time is high, the adder is slow. The execution time grows only logarithmically in the length of the numbers to be added, but that time is dominated by the actual EPR pair creation time, whereas carry-ripple adders require only a small, constant number of EPR pairs to be created per node. We also find that, because the carry-ripple adder times are now dominated by the classical communication and local gates, carry-ripple adder time is not strongly influenced by topology. In figure 9 we show this in more detail. For fast (10nsec) EPR pair creation, the carry-lookahead adder is faster for

all problem sizes. For slow (1280nsec) EPR pair creation time, carry-lookahead is not faster until we reach 512 bits. The times for the fully and 2fully networks are both almost identical to the times for the linear network.

Although we do not include graphs, we have also varied the time for classical communication and the other types of gates. The performance of an adder is fairly insensitive to these changes; it is dominated by the relationship between CCNOT and EPR pair creation times.

6 Conclusion

We have evaluated the performance of arithmetic circuits on a quantum multicomputer, a system composed of multiple nodes, for different problem sizes, interconnect topologies, and gate timings. Although we have assumed that the interconnect is based on the qubus entanglement protocol creation of EPR pairs, our analysis, especially table 1, applies equally well to any two-level structure with low-latency local operations and high-latency long-distance operations. The details of the cost depend on the interconnect topology, number of transceiver qubits, and the chosen breakdown for CCNOT. More important than actual gate times for this analysis is gate time ratios. The time values presented here are reasonable for solid-state qubits under optimistic assumptions about advances in the underlying technology. Applying our results to slower technologies (or the same technology using more layers of quantum error correction) is a simple matter of scaling by the appropriate clock speed and storage requirements.

We find that the teledata method is faster than the telegate method, and that separating the actual data teleportation from the necessary EPR pair creation allows a carry-ripple adder to be efficient for large problems. Each of the adder algorithms has natural groupings of small numbers (2-5) of qubits; when groupings are mapped to nodes, a linear network topology is adequate for up to a hundred nodes or more, depending on the cost ratio of EPR pair creation to local gates. For very large systems, switching interconnects, which are well understood in the optical domain [30, 35, 52], may become necessary, though we recommend deferring the addition of switching due to the complexity and the inherent signal loss; switching time in such systems also must be considered.

Our results show that node size, interconnect topology, distributed gate approach (teledata v. telegate), and choice of adder affect overall performance in sometimes unexpected ways. Increasing the number of logical qubits per node, for example, reduces the total number of interconnect transfers but concentrates them in fewer places, causing contention for access. Therefore, increasing node size is not favorable *unless node I/O bandwidth increases proportionally*; we recommend keeping the node size small and fixed for the foreseeable future.

This research is part of an overall effort to design a scalable quantum multicomputer. We are currently investigating distributed quantum error correction, which will determine whether each link in the interconnect, as presented here, must be parallel or may be serial [60]. Many mappings of qubits to nodes (and gates to bus timeslots) are possible; we do not claim the arrangements presented here are optimal. We are investigating further layouts using evolutionary algorithms, and expect to report those results at a future date.

Our data presents a clear path forward. We recommend pursuing a node architecture consisting of only a few logical qubits and initially two transceiver (quantum I/O) qubits. This will allow construction of a linear network, which will perform adequately with a carry-ripple adder up to moderately large systems. Engineering emphasis should be placed on supporting more transceiver qubits in each node, which can be used to parallelize transfers, decrease the network diameter, and provide fault tolerance. Significant effort is warranted on minimizing the key parameter of EPR pair creation time. Only once these avenues have been exhausted should the node size be increased and a switched optical network introduced. This approach should lead to the design of a viable quantum multicomputer.

Acknowledgments

The authors thanks Eisuke Abe for useful discussions, and Thaddeus Ladd for both discussions and writing advice. We thank the anonymous referees for their valuable input. We thank Darshan Thaker for helping to ensure consistency of presentation in the 2006 ISCA quantum computing papers.

References

[1] ARDA. *A quantum information science and technology roadmap*, v2.0 edition, Apr. 2004.

[2] M. A. Armen, J. K. Au, J. K. Stockton, A. C. Doherty, and H. Mabuchi. Adaptive homodyne measurement of optical phase. *Physical Review Letters*, 89:133602, 2002.

[3] W. C. Athas and C. L. Seitz. Multicomputers: message-passing concurrent computers. *IEEE Computer*, 21:9–24, Aug. 1988.

[4] S. Balensiefer, L. Kregor-Stickles, and M. Oskin. An evaluation framework and instruction set architecture for ion-trap based quantum micro-architectures. In *Proc. 32nd Annual International Symposium on Computer Architecture*, June 2005.

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

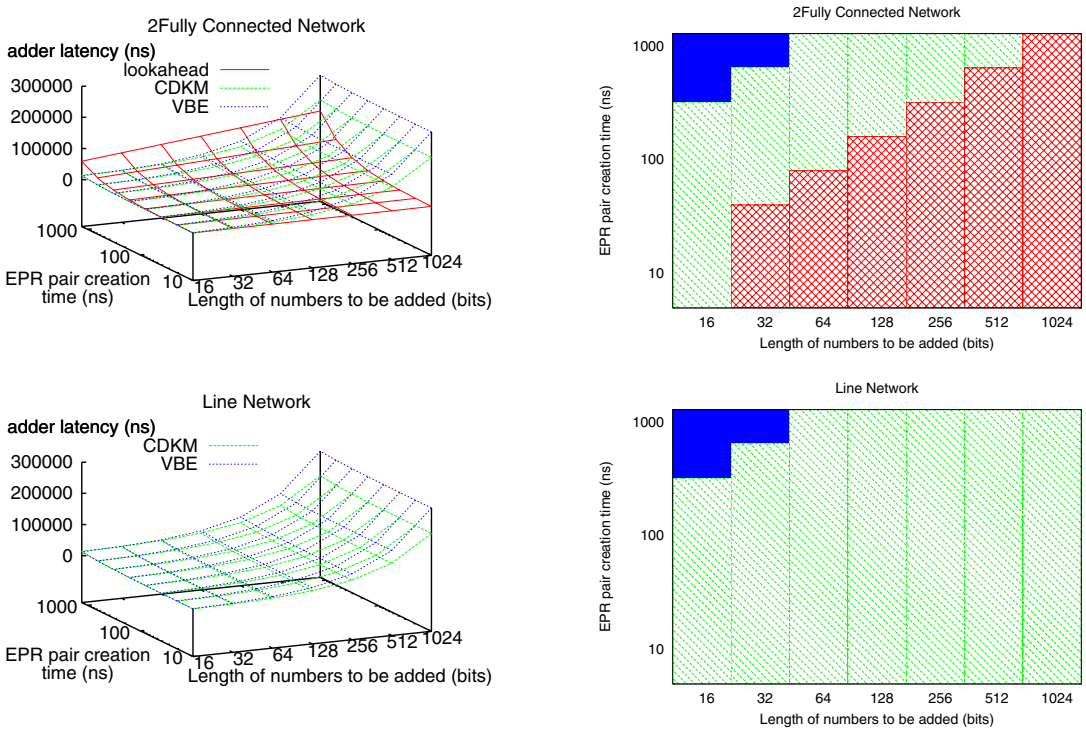

Figure 7. (Telegate) Performance of different adders on two different networks, one with two links per node (2fully) and one line configuration. See section 5.3.

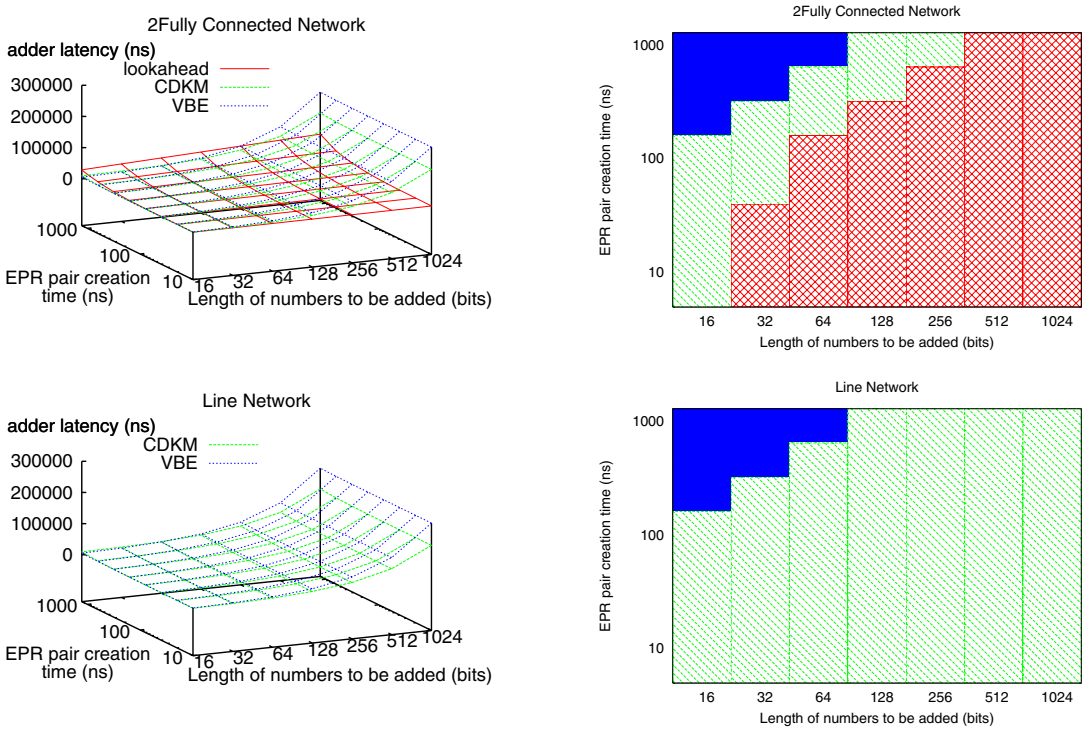

Figure 8. (Teledata) The setup is the same as the previous graph.

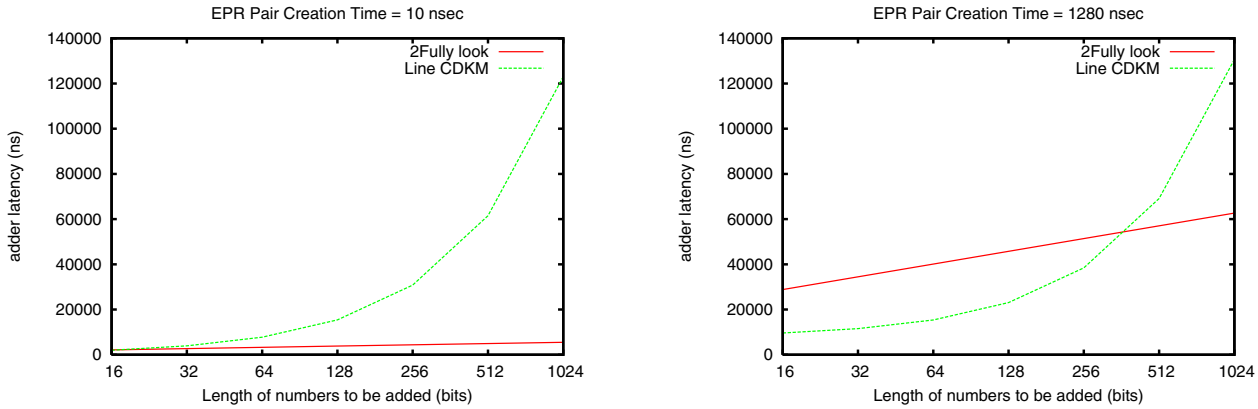

Figure 9. (Teledata) Comparison of CDKM on a line network with carry-lookahead on a 2fully network. These are the "front" and "back" cross-sections of figure 8.

[6] D. Beckman, A. N. Chari, S. Devabhaktuni, and J. Preskill. Efficient networks for quantum factoring. *Phys. Rev. A*, 54:1034–1063, 1996. http://arXiv.org/quant-ph/9602016.

[7] C. H. Bennett, H. J. Bernstein, S. Popescu, and B. Schumacher. Concentrating partial entanglement by local operations. *Physical Review A*, 53:2046, 1996.

[8] C. H. Bennett, G. Brassard, C. Crépeau, R. Josza, A. Peres, and W. Wootters. Teleporting an unknown quantum state via dual classical and EPR channels. *Physical Review Letters*, 70:1895–1899, 1993.

[9] D. Bouwmeester, J.-W. Pan, K. Mattle, M. Eibl, H. Weinfurter, and A. Zeilinger. Experimental quantum teleportation. *Nature*, 390:575–579, Dec. 1997.

[10] A. R. Calderbank and P. W. Shor. Good quantum error-correcting codes exist. *Physical Review A*, 54:1098–1105, 1996.

[11] L. Childress, J. M. Taylor, A. S. Sørensen, and M. Lukin. Fault-tolerant quantum repeaters with minimal physical resources, and implementations based on single-photon emitters. http://arXiv.org/quant-ph/0502112, Feb. 2005.

[12] J. Cirac, A. Ekert, S. Huelga, and C. Macchiavello. Distributed quantum computation over noisy channels. *Physical Review A*, 59:4249, 1999.

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

[14] D. Copsey, M. Oskin, T. Metodiev, F. T. Chong, I. Chuang, and J. Kubiatowicz. The effect of communication costs in solid-state quantum computing architectures. In *Proceedings of the fifteenth annual ACM Symposium on Parallel Algorithms and Architectures*, pages 65–74, 2003.

[15] S. A. Cuccaro, T. G. Draper, S. A. Kutin, and D. P. Moulton. A new quantum ripple-carry addition circuit. http://arXiv.org/quant-ph/0410184, Oct. 2004.

[16] D. Deutsch and R. Jozsa. Rapid solution of problems by quantum computation. *Proc. R. Soc. London*, Ser. A, 439:553, 1992.

[17] D. P. DiVincenzo. Quantum gates and circuits. *Proc. Royal Soc. London A*, 1998.

[18] T. G. Draper, S. A. Kutin, E. M. Rains, and K. M. Svore. A logarithmic-depth quantum carry-lookahead adder. http://arXiv.org/quant-ph/0406142, June 2004.

[19] M. D. Ercegovac and T. Lang. *Digital Arithmetic*. Morgan Kaufmann, San Francisco, CA, 2004.

[20] T. Fujisawa, T. H. Oosterkamp, W. G. van der Wiel, B. W. Broer, R. Aguado, S. Tarucha, and L. P. Kouwenhoven. Spontaneous emission spectrum in double quantum dot devices. *Science*, 282:932–935, 1998.

[21] A. Furusawa, J. L. Sørensen, S. L. Braunstein, C. A. Fuchs, H. J. Kimble, and E. S. Polzik. Unconditional Quantum Teleportation. *Science*, 282(5389):706–709, 1998.

[22] D. Gottesman and I. L. Chuang. Demonstrating the viability of universal quantum computation using teleportation and single-qubit operations. *Nature*, 402:390–393, 1999.

[23] L. Grover. A fast quantum-mechanical algorithm for database search. In *Proc. 28th Annual ACM Symposium on the Theory of Computation*, pages 212–219, 1996. http://arXiv.org/quant-ph/9605043.

[24] L. K. Grover. Quantum telecomputation. http://arXiv.org/quant-ph/9704012, Apr. 1997.

[25] S. Gulde, M. Riebe, G. P. T. Lancaster, C. Becher, J. Eschner, H. Haffner, F. Schmidt-Kaler, I. L. Chuang, and R. Blatt. Implementation of the Deutsch-Jozsa algorithm on an ion-trap quantum computer. *Nature*, 421:48–50, 2003.

[26] F. Jelezko, T. Gaebel, I. Popa, M. Domhan, A. Gruber, and J. Wratchtrup. Observation of coherence oscillation of a single nuclear spin and realization of a two-qubit conditional quantum gate. *Physical Review Letters*, 93:130501, Sept. 2004.

[27] J. Johansson et al. Vacuum Rabi oscillations in a macroscopic superconducting qubit LC oscillator system. http://arXiv.org/cond-mat/0510457, 2005.

[28] R. Jozsa and N. Linden. On the role of entanglement in quantum computational speedup. *Proc. Royal Soc. London A*, 459:2011–2032, 2003. http://arXiv.org/quant-ph/0201143.

[29] D. Kielpinski, C. Monroe, and D. J. Wineland. Architecture for a large-scale ion-trap quantum computer. *Nature*, 417:709–711, 2002.

[30] J. Kim et al. 1100x1100 port MEMS-based optical crossconnect with 4-dB maximum loss. *IEEE Photonics Technology Letters*, 15(11):1537–1539, 2003.

[31] J. Kim et al. System design for large-scale ion trap quantum information processor. *Quantum Information and Computation*, 5(7):515–537, 2005.

[32] D. E. Knuth. *The Art of Computer Programming, volume 2 / Seminumerical Algorithms*. Addison-Wesley, Reading, MA, 3rd edition, 1998.

[33] Y. L. Lim, S. D. Barrett, A. Beige, P. Kok, and L. C. Kwek. Repeat-Until-Success quantum computing using stationary and flying qubits. http://arXiv.org/quant-ph/0508218, Aug. 2005.

[34] P. J. Love and B. M. Boghosian. Type-II quantum algorithms. *Physica A*, 2005, to appear.

[35] P. J. Marchand, A. V. Krishnamoorthy, G. I. Yayla, S. C. Esener, and U. Efron. Optically augmented 3-d computer: System technology and architecture. *J. Parallel and Distributed Computing*, 41(1):20–35, Feb. 1997.

[36] D. N. Matsukevich and A. Kuzmich. Quantum State Transfer Between Matter and Light. *Science*, 306(5696):663–666, 2004.

[37] M. Mehring, J. Mende, and W. Scherer. Entanglement between and electron and a nuclear spin 1/2. *Physical Review Letters*, 90:153001, Apr. 2003.

[38] W. Munro, K. Nemoto, and T. Spiller. Weak nonlinearities: a new route to optical quantum computation. *New Journal of Physics*, 7:137, May 2005.

[39] Y. Nakamura, Y. A. Pashkin, and J. S. Tsai. Coherent control of macroscopic quantum states in a single-cooper-pair box. *Nature*, 398:786–788, Apr. 1999.

[40] K. Nemoto and W. J. Munro. Nearly deterministic linear optical controlled-NOT gate. *Physical Review Letters*, 93:250502, 2004.

[41] M. A. Nielsen. Simple rules for a complex quantum world. In *The Edge of Physics*. Scientific American, 2003.

[42] M. A. Nielsen and I. L. Chuang. *Quantum Computation and Quantum Information*. Cambridge University Press, 2000.

[43] M. Oskin, F. T. Chong, I. L. Chuang, and J. Kubiatowicz. Building quantum wires: The long and short of it. In *Computer Architecture News, Proc. 30th Annual International Symposium on Computer Architecture*. ACM, June 2003.

[44] P. W. Shor. Algorithms for quantum computation: Discrete logarithms and factoring. In *Proc. 35th Symposium on Foundations of Computer Science*, pages 124–134, Los Alamitos, CA, 1994. IEEE Computer Society Press.

[45] P. W. Shor. Fault-tolerant quantum computation. In *Proc. 37th Symposium on Foundations of Computer Science*, pages 56–65, Los Alamitos, CA, 1996. IEEE Computer Society Press.

[46] T. P. Spiller, W. J. Munro, S. D. Barrett, and P. Kok. An introduction to quantum information processing: applications and realisations. Technical Report HPL-2005-192, Oct. 2005.

[47] T. P. Spiller, K. Nemoto, S. L. Braunstein, W. J. Munro, P. van Loock, and G. J. Milburn. Quantum computation by communication. http://arxiv.org/abs/quant-ph/0509202, Sept. 2005.

[48] A. M. Steane. Overhead and noise threshold of fault-tolerant quantum error correction. *Physical Review A*, 68:042322, 2003.

[49] A. M. Steane. How to build a 300 bit, 1 Gop quantum computer. http://arxiv.org/abs/quant-ph/0412165, Dec. 2004.

[50] A. M. Steane and D. M. Lucas. Quantum computing with trapped ions, atoms, and light. *Fortschritte der Physik*, Apr. 2000. http://arXiv.org/quant-ph/0004053.

[51] J. Stockton, M. Armen, and H. Mabuchi. Programmable logic devices in experimental quantum optics. *J. Opt. Soc. Am. B*, 19:3019, 2002.

[52] T. Szymanski and H. Hinton. Design of a terabit free-space photonic backplane for parallel computing. In *Proc. Second Workshop on Massively Parallel Processing Using Optical Interconnections*. IEEE, 1995.

[53] D. D. Thaker, T. Metodi, A. Cross, I. Chuang, and F. T. Chong. CQLA: Matching density to exploitable parallelism in quantum computing. In *Computer Architecture News, Proc. 33rd Annual International Symposium on Computer Architecture*. ACM, June 2006.

[54] R. Van Meter and K. M. Itoh. Fast quantum modular exponentiation. *Physical Review A*, 71(5):052320, May 2005.

[55] R. Van Meter and M. Oskin. Architectural implications of quantum computing technologies. *J. Emerging Tech. in Comp. Sys.*, 2(1), Jan. 2006. to appear.

[56] L. M. K. Vandersypen, M. Steffen, G. Breyta, C. S. Yannoni, M. H. Sherwood, and I. L. Chuang. Experimental realization of Shor's quantum factoring algorithm using nuclear magnetic resonance. *Nature*, 414:883–887, Dec. 2001.

[57] V. Vedral, A. Barenco, and A. Ekert. Quantum networks for elementary arithmetic operations. *Phys. Rev. A*, 54:147–153, 1996. http://arXiv.org/quant-ph/9511018.

[58] A. Wallraff, D. I. Schuster, A. Blais, L. Frunzio, R.-S. Huang, J. Majer, S. Kumar, S. M. Girvin, and R. J. Schoelkopf. Strong coupling of a single photon to a superconducting qubit using circuit quantum electrodynamics. *Nature*, 431:162–167, Sept. 2004.

[59] C. P. Williams and S. H. Clearwater. *Ultimate Zero and One: Computing at the Quantum Frontier*. Copernicus Books, 1999.

[60] F. Yamaguchi, K. Nemoto, and W. J. Munro. Quantum error correction via robust probe modes. http://arxiv.org/abs/quant-ph/0511098, Nov. 2005.

[61] J. Yepez. Type-II quantum computers. *International Journal of Modern Physics C*, 12(9):1273–1284, 2001.

[62] A. Yimsiriwattana and S. J. Lomonaco Jr. Distributed quantum computing: A distributed Shor algorithm. http://arxiv.org/quant-ph/0403146, Mar. 2004.