Architectural Implications of Quantum Computing Technologies

RODNEY VAN METER
Keio University and CREST-JST
and
MARK OSKIN
University of Washington

In this article we present a classification scheme for quantum computing technologies that is based on the characteristics most relevant to computer systems architecture. The engineering trade-offs of execution speed, decoherence of the quantum states, and size of systems are described. Concurrency, storage capacity, and interconnection network topology influence algorithmic efficiency, while quantum error correction and necessary quantum state measurement are the ultimate drivers of logical clock speed. We discuss several proposed technologies. Finally, we use our taxonomy to explore architectural implications for common arithmetic circuits, examine the implementation of quantum error correction, and discuss cluster-state quantum computation.

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

General Terms: Design

Additional Key Words and Phrases: Quantum computing, quantum computer architecture

1. INTRODUCTION

Quantum computing is a rapidly evolving field. Researchers are motivated by the enormous computational potential [Shor 1997; Grover 1996; Deutsch and Jozsa 1992] compared to classical machines. Much of the effort thus far has been focused on the two extremes of research: algorithms and complexity theory at the top end, and quantum gate and qubit storage technology at the bottom [ARDA 2004]. The role that architects play in bridging this gap, designing practical machines that can run quantum algorithms efficiently, is only

This work is supported in part by the DARPA QuIST Program (ARFL-F30602-01-2-0521), NSF Nanoscale Program (CCF-0210373) and NSF CAREER grants (CCF-0133188). Additional support is provided by the A. P. Sloan Foundation.
Authors' address: R. Van Meter, Keio University, 3-14-1 Hiyoshi, Kohoku-ku, Yokohama-shi, Kanagawa 223-8522, Japan; email: rdv@tera.ics.keio.ac.jp. M. Oskin, University of Washington.

beginning to be explored [Oskin et al. 2003; Steane 2003; Van Meter and Itoh 2005].

The architecture of these systems cannot be an afterthought. A well-designed system will be faster, have higher storage capacity, and be more error-resistant than a poorly built one developed from the same underlying technology. It will therefore be capable of attacking larger, more interesting problems. System architects can also help shape the device research agenda, by demonstrating system-level trade-offs and establishing the relative importance and necessary maturity levels of various technological features. Thus, the system architecture of quantum computers is critical to their success and conducting research on it now is important to the field.

In this article, we hope to provide a useful road map in the form of a taxonomy and survey of some of the many promising quantum computing technologies, written from the perspective of an architecture researcher. We also demonstrate in a concrete fashion how to apply the information from such a taxonomy by evaluating quantum arithmetic algorithms, quantum error correction, and the new topic of cluster-state quantum computation.

When examining a proposal for a new quantum computing technology, the architect is going to ask several basic questions. Among them are:

(1) Does it work (or can it be made to work)?
(2) How do you control it (both hardware and software)?
(3) How does it scale?
(4) How fast is it?
(5) What trade-offs can we make?

In this article we will focus on questions 2–5, with some attention paid to the current state of knowledge vis-a-vis question 1. Question 1 is largely a matter for device physicists, however, architects need to be aware of the potential challenges in the manufacture of a large-scale system because those challenges will also ultimately dictate when and if a quantum computer is built. Furthermore, architects can help to mitigate [Isailovic et al. 2004] those challenges by exploiting their design expertise.

Question 2 is important to architects because the classical control of these quantum technologies is a complicated architectural problem in its own right. Some of these technologies will permit the integration of significant classical logic alongside the quantum bits (qubits), while others will require the classical control to be "off chip." The parallelism on the quantum side of the device, the operation speed, and the necessity of fast, accurate qubit measurement and control create a need for a high-bandwidth interface between the control subsystem and the quantum device itself.

Question 3 is paramount. If you cannot build a large-scale system with hundreds to thousands of application-level qubits, then the device is never going to leave the "lab bench" stage. However, the question is extremely complex to answer, depending upon technological limits as well as fundamental physical characteristics. For this reason, scalability is addressed throughout this text.

The speed of these technologies (question 4) is important for two reasons. First, the speed of computation as it relates to the error rate of that computation (called decoherence in the quantum world) determines whether a quantum computer can even be built and what that computer will be capable of doing. Second, despite the advantage in computational complexity that the quantum computing model has compared to classical machines on some problems, computing interesting results using quantum algorithms requires large numbers of gates. Factoring of kilobit-length numbers, for example, is $O(n^3)$. To handle the high error rates of the fragile quantum states, a significant constant (potentially in the hundreds of thousands) is also in front of this term. Thus, the underlying speed of computation is still an important consideration for quantum systems and their ultimate usefulness [Van Meter et al. 2005].

Finally, question 5 is at the heart of architecture research. We will explore features of different technologies that can be traded off against each other. For example, concurrent gate execution is highly desirable for both error correction and application algorithms. However, for some technologies, manufacturing a concurrent architecture is complex or expensive (e.g., requiring large numbers of lasers for ion trap systems), while for other technologies concurrency results in higher error rates or more complex signal handling (e.g., isolation of qubits in an NMR system).

This article does not require a detailed knowledge of the theory of quantum computing to read. We refer the interested reader to popular [Williams and Clearwater 1999] and technical [Nielsen and Chuang 2000; Preskill 1998a; Spiller et al. 2005; Galindo and Martin-Delgado 2002] texts on the subject. We begin in the next section by describing well-known fundamental criteria for quantum computing devices, followed by the framework for our taxonomy. Section 4 uses this framework to place several promising quantum technologies in context. Section 5 shows some of the ways in which applications, architecture, and technology interact. We conclude with a discussion in Section 6.

2. FUNDAMENTAL REQUIREMENTS

DiVincenzo [1995] described five capabilities which a real-world quantum computing device must have. A quantum computer must:

(1) be a scalable physical system with well-defined qubits;
(2) be initializable to a known state prior to computation;
(3) have adequately long decoherence times;
(4) have a universal set of quantum gates; and
(5) permit high efficiency quantum measurements.

Two additional criteria focus on moving quantum information between two different quantum computers. A viable quantum communications technology must:

(6) be able to convert between physical realizations of qubits that are stationary and moving; and

(7) be able to faithfully transmit a physical realization of a qubit between specified locations.

These criteria are fairly straightforward. Item 1 means there must be some physical entity, such as the energy levels of an ion, the polarization of a photon, or the spin of an electron, that is the actual carrier of the qubit; it must meet basic criteria of quantum behavior and support two distinct states which can be treated as zero and one. Item 1 also refers to "scalability," which means different things in different contexts; we will explore its system aspects. Item 2 is pretty obvious: it is just restating the "garbage in—garbage out" principle [Schulman and Vazirani 1999].

The critical issue of decoherence, or loss of the quantum state (item 3), is a function of time, gate errors, and qubit transport. A fundamental, theoretical calculation places the threshold above which a quantum computer will not function at 10^{-4}, or one error per 10^4 gates executed [Aharonov and Ben-Or 1999], but quantum technologies need to achieve error rates well below this critical threshold to avoid undue overhead from error correction processes [Steane 2003]. Technology- and architecture-specific estimates sometimes differ dramatically from this theoretical threshold, as discussed in Section 5.2.

To satisfy criterion 4, a computer must be able to compute a small number of "universal" gates that can be used to synthesize larger, more complex gates. This is equivalent to saying that a classical computing technology should be able to perform at least a NOR or NAND operation. For quantum computers, one such set of universal gates is X, H, T, and CNOT. The most important thing to know about these gates is that the X, H, and T are single bit operations, while CNOT involves two qubits. Of secondary importance is that X, H and CNOT are relatively simple to make fault tolerant, while T requires a more complex circuit.

Item 5 demands that there be a reliable way to read out the state of a qubit. Measurement is far more important than retrieving results at the end of a computation; it occurs almost continuously as part of quantum error correction and the fault-tolerant execution of gates on encoded bits [Shor 1996; Calderbank and Shor 1996; Steane 2003; Gottesman 1999; Steane 2002].

Items 6 and 7 deal specifically with moving quantum information across long distances for purposes of computation. One important caveat on criterion 6 is that it only applies to systems that compute complex quantum algorithms via shared state. It does not apply to other uses of quantum effects, such as quantum cryptography [Bennett and Brassard 1984; Elliott et al. 2003] and basic demonstrations of quantum teleportation [Bennett et al. 1993; Furusawa et al. 1998] (though teleportation may be used in quantum computer architectures [Gottesman and Chuang 1999; Grover 1997]).

These criteria have been used as a basis for the evaluation of quantum computing technologies [Nielsen and Chuang 2000; Spiller et al. 2005; ARDA 2004]. They are a necessary set of capabilities, but not sufficient to understand the difficulty of building a quantum computer or its speed and utility once built.

3. TAXONOMY FRAMEWORK

To provide architecture researchers with a useful guide to evaluating quantum computing technologies, we have developed a set of classification criteria. We will describe each criterion as well as some of its high-level architectural implications. In the next section we will use this taxonomy to evaluate several proposed computing technologies.

3.1 Basic Features

Stationary, flying, and mobile. Quantum computing technologies can be divided into two categories: those in which the qubits are represented by constantly moving phenomena (photons) and those in which qubits are represented by static phenomena (nuclear or electron spins). For phenomena that move, gates are physical devices which affect qubits as they flow through the gate. These qubits are called "flying qubits." Optical implementations generally fall into this category. For "stationary" phenomena, qubits occupy a physical place and gate operations from an application are applied to them. The "stationary" notion applies *only* during gate operation. Some stationary technologies, such as the proposed scalable ion trap [Kielpinski et al. 2002], permit the physical qubit carrier to be moved prior to application of a gate; we will call these "mobile" qubits.

The key reason to make the distinction between stationary and flying implementations is dynamic control. In a flying qubit device, the order and type of gates must typically be fixed in advance, often at device construction time; different program execution is achieved by classical control of switches that route qubits through different portions of the circuit. A stationary qubit device has more flexibility to reconfigure gates. In this sense, stationary devices are like classical programming, while flying qubit designs are more like classical circuit design [Yao 1993].

Single system versus ensemble. A significant distinction in quantum computing technologies is the choice of *ensemble* computing or *singleton* computing. In ensemble computing, generally implemented on static qubit systems, there are many identical quantum computers all receiving the same operators and executing the same program on the same data (except for noise). Singleton systems have the ability to directly control a single physical entity that is used to represent the qubit.

From a technology perspective, ensemble systems are easier to experiment with, as techniques for manipulating and measuring large numbers of atoms or molecules are well understood. Hence, the largest quantum computing system demonstrations to date have all been on bulk-spin NMR [Vandersypen et al. 2001; Boulant et al. 2003], which uses an ensemble of molecules to compute.

Measurement. In order to compute reliably, and to be able to observe the result of computation, computing technologies must support a readout process. This readout, called measurement, observes the state of the quantum bit and produces a classical result. Four features characterize different measurement schemes: (1) can measurement of multiple quantum bits be performed in parallel or must they be serialized? (2) does measurement of a quantum bit require

interaction with another "clean" qubit in order to produce a result? (3) is the speed of measurement about as fast (in the same order of magnitude) as performing an operation? (4) can measurement be performed almost anywhere, or must the physical entities that are used to represent the qubits be moved to specialized measurement sites?

Reliably computing on a quantum system will mean that many, if not most, of the total quantum operations will be measurements [Steane 2002]. From an architectural perspective, if measurements must be performed serially or are inordinately slow, then Amdahl's Law [Amdahl 1967] will apply and measurement will be the bottleneck in computation. Furthermore, if additional "clean" qubits are required for measurement to take place, then we must plan for the initialization process to occur frequently. Similarly, if technologies restrict where measurement can occur, then those restrictions will need to be designed into the architecture.

Error processes and suppression. The basic theory of quantum error correction (QEC) has been known for almost a decade [Shor 1996; Steane 2003], but its application turns out to be technology-specific [Gottesman 1999]. QEC traditionally depends on interleaving measurement and logic gates, and there has been recent experimental progress on this front [Roos et al. 2004]. However, it is possible to perform QEC without measurement, at a cost of a number of ancillae ("temporary" variable qubits) that grows with the number of applications of error correction [Nielsen and Chuang 2000].

Qubits in some technologies suffer from independent errors, making them amenable to correction via QEC. Additional techniques known as *decoherence free subspaces* (DFSes) [Lidar et al. 1998; Lidar and Whaley 2003] are especially useful when nearby qubits are subject to collective error processes.

In optical systems, the principal source of error is loss of photons. In this case, *erasure codes* (in contrast to *error correcting codes*) work well [Knill et al. 2000], because it is easy to determine which qubit has been lost, much like parity is used in a RAID array [Patterson et al. 1988].

3.2 Manufacturing and Operating Environment

At the moment, all scalable quantum computing technologies are proposals and significant advances in manufacturing will be required to bring them to reality. Nevertheless, some proposals have less onerous technological hurdles in front of them than others. Furthermore, certain proposed technologies integrate better with existing classical silicon-based computing.

Fabrication challenges. To what extent do the proposed technologies rely on difficult-to-achieve advances in manufacturing? For example, early silicon-based NMR relied upon the ability to dope silicon with precisely placed individual phosphorus atoms, and to align these with overlaid structures created using standard VLSI lithography [Kane 1998]. All of the solid-state circuit techniques require classical control lines (e.g., [Fujisawa et al. 1998; Nakamura et al. 1999]), which may benefit from expected improvement in VLSI feature sizes following Moore's Law [Moore 1965; ESIA et al. 2003]. In our taxonomy we will highlight the major technological challenges facing each

quantum computing proposal and discuss the latest advances in overcoming them.

Control parallelism. Despite the algorithmic advantages quantum computing promises, it is still important to extract parallelism from quantum algorithms. On some proposed technologies, such as silicon NMR [Skinner et al. 2003], serial execution of algorithms will be time-consuming. For example, Kunihiro [2005] has estimated the sequential running time of Shor's algorithm factoring a 530–bit number at 1.18 years for a 1kHz device (approximately NMR speeds), 10 hours for a 1MHz device, or 37 seconds for a 1GHz device.

Fortunately, there is significant parallelism available [Moore and Nilsson 2001] in quantum software (error correction [Steane 2003] and factoring [Van Meter and Itoh 2005]). The ability to exploit this parallelism, however, requires technologies with parallel control. This parallel control will require significant classical support circuitry. If this circuitry cannot be located "on chip" near the qubits, then a high-bandwidth interface between a classical device generating control pulses and a quantum device containing the actual qubits will be required.

Operating temperature. In order to control noise, most proposals call for extremely low temperatures achievable only with liquid helium. Others require still colder *millikelvin* temperatures achieved through a dilution refrigerator. These low temperatures are not only operationally challenging, but also affect the ability of classical circuits to operate, complicating the design of the control process [Oskin et al. 2003].

Supporting equipment. Some technologies require complex supporting equipment, notably high-frequency microwave and voltage signal generators. One or more of these per qubit may be needed; as systems scale, switching or sharing of this equipment or direct integration into on-chip systems is likely to be required.

3.3 Algorithmic Efficiency Features

Many features of the various quantum computing proposals will have profound implications for the execution of quantum algorithms on realistic architectures.

Total available qubits. The feature with the single largest impact on the scalability, usefulness, and reliability of the computer is the actual number of physical qubits available. Beyond the minimum requirement for a given algorithm, additional qubits can be utilized to increase reliability via error correction or for algorithmic parallelism, as we show in Section 5.

All entries in Table II are followed by question marks because of very high uncertainty; in some cases, even which factors will prove to be the practical limits are not yet clear. As most researchers are still focusing on very small numbers of qubits, they have not yet attempted to circumscribe this upper limit.

Addressability. In some systems, addressing specific qubits is difficult, because it is hard to localize the classical control (e.g., microwave-frequency electromagnetic field) required to just the small region that a single qubit occupies. One solution, the original Lloyd model, proposes forming small groups of qubits

into cellular automata [Lloyd 1993]. Each qubit position in the automaton can be addressed via a specific electromagnetic frequency. Each automaton follows the same program, effected by electromagnetic radiation blanketing the whole device, which is, in effect, a fully concurrent SIMD machine. One technique for turning a cellular automata into a more easily controlled serial machine is to include in the cellular automata a token that is passed from automaton to automaton; only the automaton holding the token performs the indicated action. We expect that designing architectures and software systems for technologies without the ability to address and operate on specific qubits will be difficult.

Wiring. Optimization of the architecture to support the data movement of a useful class of algorithms is one of the key areas to which computer architects can contribute. In many proposed technologies, only neighboring qubits are allowed to perform two-qubit gates. Either the physical entities representing the qubits (using a control process [Kielpinski et al. 2002]) or just the state (using quantum wires [Oskin et al. 2003]) must be moved around within the machine to support computation. In some cases, technological constraints limit the interconnection topology to a one-dimensional line; in others, a loose two-dimensional lattice, full 2-D mesh, or even 3-D structure has been proposed [Lloyd 1993]. A few proposals support long-distance gates with various trade-offs, such as limited concurrency [You et al. 2002].

3.4 Time and Gate Characteristics

Natural gates. Various sets of gates have been shown to form elementary basis sets [Barenco et al. 1995; DiVincenzo 1994]. The standard set of universal gates (X, H, T, CNOT) is just one example, and all serious proposals for quantum computing technologies include enough operations to provide this or an equivalent universal set. Beyond universality, however, are three important characteristics. (1) Does the technology provide an arbitrary single qubit rotation, or must it be synthesized from X, H and T [Nielsen and Chuang 2000]; (2) How complex is the synthesis for a three qubit controlled-controlled-not (called a Toffoli gate, for its inventor), which is commonly used in quantum algorithms [Barenco et al. 1995]; (3) Do specific gates have unwanted effects on qubits that are *not* the intended operands (that is, are other qubits being implicitly manipulated)? We will discuss these in more detail below.

Arbitrary single qubit rotation and controlled-arbitrary rotation are extremely useful primitives. Technologies that support arbitrary single qubit rotations do not need to utilize any of the synthesis techniques developed to approximate them, saving a polynomial time overhead. Arbitrary controlled rotation (a two qubit operation) is useful, particularly for the quantum Fourier transform (QFT) at the heart of Shor's algorithm [Shor 1994; Hales and Hallgren 2000; Barenco et al. 1996].

The TOFFOLI gate is a three qubit gate that can be used to synthesize conventional logic from quantum bits (NAND, NOR, etc.), and it appears prominently in Shor's algorithm, which relies heavily on arithmetic circuits. We will take a closer look at this in Section 5.1. Efficient constructions of TOFFOLI are possible if

the underlying technology supports certain single qubit operations or arbitrary single qubit rotations.

In static qubit devices such as ion traps or NMR systems, several electromagnetic pulses are generally required to implement each gate. A typical number is five or six, though the exact number and timing are dependent on the gate to be executed. One side effect in NMR systems is that nearby qubits are affected by these pulses and are implicitly operated on by them. To overcome this, additional control sequences called *decoupling pulses* are required [Beckman et al. 1996; Leung et al. 2000].

Coherence versus operation time. The upside to good isolation from environmental effects is long *coherence time*, or the time which a qubit can be "kept." As a broad generalization, those technologies relying upon electrons to maintain quantum state have short coherence times because electrons are fairly mobile and tend to interact with their surrounding environment. Technologies that utilize nuclear effects are more stable. However, the downside to good isolation from environmental effects is relatively slow operation times for two-qubit gates. Across the technologies we examine, the gate speed and decoherence time vary over eight orders of magnitude or more [Ladd et al. 2003]. Coherence time is an especially important research area and will be subject to potentially large advances as QC technology progresses. Gate operation time, however, is often tied directly to physical processes with limited flexibility in engineering parameters.

3.5 Other Features

Logical encoding. Quantum algorithms are written to manipulate abstract, logical qubits. Logical qubits, however, are not always represented by a single physical phenomenon such as a single ion or photon. We call the entities that software manipulates "logical qubits" (or "encoded qubits" when quantum error correction is involved) and the entities that technologies use to implement them "elementary qubits." This is not the same as the ensemble/singleton distinction outlined above.

In some technologies, such as electron count (charge) in quantum dots, a "dual rail" encoding is used. Similarly, a single photon may take either the left or right path through a circuit, corresponding to different logical quantum states (i.e., 0 or 1). In both of these technologies, it is possible to talk about a single quantum dot (or path) as a single qubit, but we arrange computation and measurement to take place on the encoded pair.

Scalability limits. Scaling to large numbers of qubits is, for most architectures, a function of all of the above factors and more. Other factors not yet described are technology specific. For example, in lithography-based systems, they include I/O pads on the chip, the supporting infrastructure such as rack-mount microwave generators, and the practical challenge of simply providing enough control wires to such a small device. Few of the proposals suggest that an actual numerical upper bound exists because of any of these factors, yet they are critical to the success of building systems. In the next section, we will highlight what the primary scalability limit is perceived to be for each technology.

Table I. Qubit Technology Basic Characteristics
(Question marks under QIO indicate that experimental verification has not yet been shown. JJ: Josephson junction, LOQC: linear optics quantum computing.)

Technology	Stationary/ Flying/Mobile	Single/ Ensemble	QIO?	Measurement	References
Si NMR	stationary	ensemble	N	mechanical vibration, concurrent, frequency analysis	[Ladd et al. 2002]
solution NMR	stationary	ensemble	N	concurrent, frequency analysis	[Vandersypen et al. 2001]
quantum dot charge	stationary	single	Y?	concurrent, on-chip auxiliary structures, similar to quantum dots in size and structure	[Loss and DiVincenzo 1998]
scalable ion trap	mobile	single	Y?	limited concurrent, optically induced flourescence	[Cirac and Zoller 1995; Kielpinski et al. 2002]
JJ charge	stationary	single	Y?	concurrent, on-chip charge probe	[Pashkin et al. 2003; You et al. 2003]
Kane model	stationary	single	N?	concurrent, single-electron spin measurement	[Kane 1998]
LOQC	flying	single	Y	single qubit polarization via single photon number resolving optical detectors	[Knill et al. 2001]

Quantum I/O. We may want to move quantum state from one device to another. There are a variety of reasons this may be important: we may simply be aggregating multiple devices into a larger device, or the far node may provide different computational capabilities (e.g., long-term storage) or have access to different data. In some cases, we may wish to move quantum data between devices of different technologies.

Quantum I/O (QIO) is a very error-prone process. Therefore, it is done by first using QIO on "empty" qubits, which we will call QIO sites, creating an entangled state between a pair of devices. Once the existence of the entangled state is confirmed through a process called purification, it can be used to transfer any desired quantum state by using quantum teleportation [Bennett et al. 1993; Furusawa et al. 1998; Lloyd et al. 2000; Matsukevich and Kuzmich 2004].

Question marks appear in the QIO entries in Table I because experimental demonstration in structures similar to those expected to be used in quantum computers has not yet been done, or because adequate fidelity has not been shown. In some cases, basic experimental confirmation or proposals backed by relatively solid analysis exist; in others, only a few sentences in a longer article.

4. QUANTUM TECHNOLOGIES

In this section we survey a variety of proposed quantum computing technologies using the taxonomy framework described in the last section. We have chosen

Table II. Features Affecting Algorithm Efficiency on Specific Qubit Technologies
(The maximum number of qubits in all technologies remains undetermined with any reliability. Question marks in topologies indicate that the natural area for layout is 2-D, but practical engineering constraints may limit full 2-D layout.)

Technology	Concurrency	Max Qubits	Wiring Topologies	Addressability	ops on All Qubits?
Si NMR	limited by ability to suppress activity of uninvolved qubits	hundreds?	linear nearest neighbor	by frequency, all independent	Y
solution NMR	limited by ability to suppress activity of uninvolved qubits	low tens?	linear nearest neighbor, limited non-neighbor	by frequency, all independent	Y
quantum dot charge	limited by control mechanism	large?	linear nearest neighbor	localized, independent control via on-chip systems	Y
scalable ion trap	limited by # of action sites with lasers	large?	open, irregular, up to 2-D?	individual ions and chains moved from addressable storage to action sites	N
JJ charge	limited by coupling mechanism	large?	1-D, 2-D?, long-distance possible?	localized, independent control via on-chip systems	Y
Kane model	limited by control mechanism	large?	1-D or 2-D?	localized, independent control via on-chip systems	Y
LOQC	unlimited?	large?	physical routing, essentially unlimited	physical position	Y

to focus on seven technologies: Si-NMR, P-NMR, solution NMR, quantum dot charge, scalable ion traps, Josephson junction charge, and linear optics-based systems. This selection should by no means be interpreted as exhaustive; many other viable proposals exist [Brennen et al. 1999; Folman et al. 2000; Shahriar et al. 2002; Pellizzari et al. 1995; Childress et al. 2005]. These systems were chosen for their near and long term implementability, and/or scalability and/or pedagogical interest. The information is summarized in Tables I–V. Below we will briefly discuss each technology and its architectural implications.

4.1 Solution NMR

Probably the most complete demonstrations of quantum computation to date are the solution NMR experiments [Vandersypen et al. 2001; Boulant et al. 2003]. In an NMR system, the qubit is represented by the spin of the nucleus of an atom. When placed in a magnetic field, that spin precesses, and the spin

Table III. Clock Speed and Gate Characteristics

Technology	Decoherence Time	Measurement Time	Single-Qubit Gate Clock Speed	Two-Qubit Gate Clock Speed	Natural Two-Qubit Gate
Si NMR	25s	long	40kHz	400Hz	J coupling
solution NMR	seconds	long	50kHz	50Hz	J coupling
quantum dot charge	a few ns	10–100ns [Fujisawa et al. 1998; Loss and DiVincenzo 1998]	10GHz	10GHz	exchange [Loss and DiVincenzo 1998]
scalable ion trap	1ms	100μs [Metodiev et al. 2003] to 10msec [Schmidt-Kaler et al. 2003]	14kHz to 100kHz (speed v. fidelity); $\sim$ 20kHz (ion movement)	$\sim 10\mu$sec	conditional phase shift
JJ charge	a few ns	10ns	10GHz	10GHz	conditional phase shift
Kane model	long?	long	75kHz	75kHz	J coupling
LOQC	limited by scattering and absorption	5–10ns	<1ns	limited by detector time	several possibilities, including conditional phase shift

Table IV. Other Features

Technology	Logical:elementary Encoding	Gate-Level Timing Control	Scalability Limit
Si NMR	1:1	slow gates make precise timing feasible	quality of initialization (no more than $1/n$ copies may be mis-polarized for large n, to achieve adequate SNR), precision of placement in static magnetic field, area of high-quality magnetic field
solution NMR	1:1	slow gates make precise timing feasible	SNR falls exponentially in n
quantum dot charge	1:3	gates must be precise, but jitter is not a problem	external wiring/control
scalable ion trap	1:1	use of decoherence-free subspaces recommended to reduce jitter	probably ability to accurately track large numbers of individual ions, and their movement times
JJ charge	1:1	active control of phases	cross-qubit interference; inductance of Josephson junctions; large numbers of rack-mount microwave generators and getting wires into the dilution refrigerator
Kane model	1:1		manufacturing complexity
LOQC	1:1	"stopped" light [Fleischhauer and Lukin 2000]	skew and jitter in both input generation and gates; single-photon photodetector efficiencies of $\sim$0.9 will scale poorly when used for large numbers of independent qubits; deep circuits subject to loss

can be manipulated via microwave radiation. In solution NMR, a carefully designed molecule is used. Some of the atoms in the molecule have nuclear spins, and the frequency of radiation to which they are susceptible varies depending on their position in the molecule, so that different qubits are addressed by frequency. Many copies of the molecule are held in a liquid solution; each molecule is a separate quantum computer, run independently, with the large numbers providing adequate signal strength for readout. This is the canonical ensemble system. Solution NMR has been used to factor the number 15 using Shor's algorithm, which required 720 milliseconds [Vandersypen et al. 2001]. The largest demonstration to date is 12 qubits [Cory 2004].

Table V. Manufacturing and Operating Environment. K, degrees Kelvin; mK, milliKelvin.

Technology	Manufacturing	Operating Environment	Supporting Equipment
Si NMR	Si micromachining	4 K, 7 T magnetic field	r.f. signal generator
solution NMR	chemical	room temperature, 11 T magnetic field	r.f. signal generator
1-D quantum dot charge	GaAs lithography	20 mK	GHz voltage pulse generator (per qubit?)
scalable ion trap	macroscopic electromechanical assembly	supercooled ions in room temperature vacuum	multiple lasers (gates and measurement), electronic signal generators (ion movement control), CCD cameras (state detection)
JJ charge	Si lithography	30 mK	GHz voltage pulse generator (per qubit?)
1-D Kane model	P implanted in Si lithography	1.5 K, 2 T magnetic field	
LOQC	macroscopic electromechanical assembly	dependent on optical detectors; liquid helium to room temperature	high speed optical switches, atomic clocks

No special cooling apparatus is required for this ensemble system. However, its scalability is believed to be quite limited due to falling signal/noise ratio as the number of qubits increases.

—*strengths.* good decoherence time, room temperature operation, advanced experimental verification.

—*weaknesses.* slow gates, poor scalability, difficult concurrent operations.

4.2 Josephson Junction

Josephson junction-based quantum computing devices are superconducting systems [Shnirman et al. 1997]. They come in three flavors: those that represent qubits using charge (such as the device shown in Figure 1) [Nakamura et al. 1999; Pashkin et al. 2003], those that use flux [Mooij et al. 1999; Chiorescu et al. 2004], and those that use phase [Yu et al. 2002; Martinis et al. 2002]; most of the information in the tables applies to all three. Fabrication is done using conventional electron-beam lithography and shadow evaporation of Al onto an SiN_x insulating substrate. In the JJ charge qubit, a sub-micron size superconducting box (essentially, a small capacitor) is coupled to a larger superconducting reservoir. In a superconductor, electrons move in pairs known as Cooper pairs. The qubit representation is the number of Cooper pairs in the box, controlled to be either zero or one, or a superposition of both. Similarly, for the flux qubit, Cooper pairs are introduced into a superconducting ring, where they circulate and induce a quantized magnetic flux. Because the flux qubit has slower gate times but a relatively even longer coherence time, experimental efforts appear to be shifting toward the flux qubit approach.

In one proposed scalable form of the charge qubit it is possible to address any two qubits and couple them [You et al. 2002]. This is done through a shared

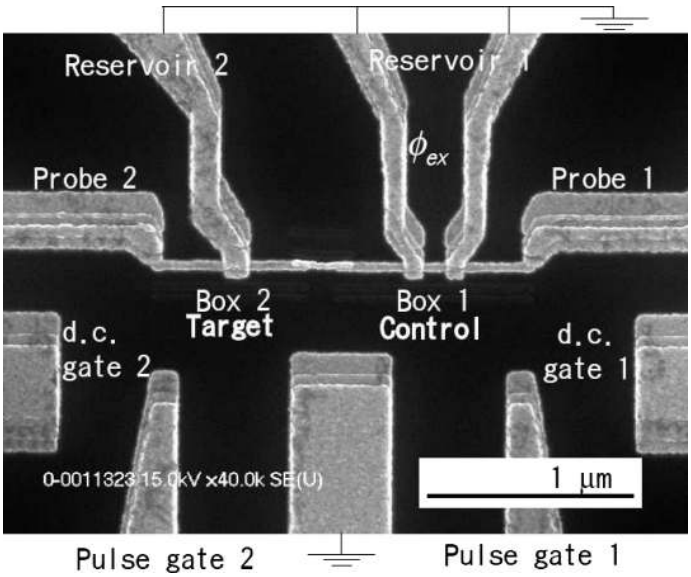

Fig. 1. A pair of coupled Josephson-junction charge qubits (labeled Box 1 and Box 2). This device is designed to execute a two-qubit gate between the qubit labeled "Control" and the one labeled "Target." The coupling between the two qubits is fixed in hardware in this device. Image courtesy of Y. Nakamura and T. Yamamoto, NEC.

inductance. In this case, the restriction of operations involving only neighboring qubits in a linear array is removed, but execution is limited to one gate at a time. A different proposal links neighboring qubits in a one-dimensional structure with nearest-neighbor-only gates, but potentially may allow concurrent gates on independent qubits [Lantz et al. 2004].

—*strengths.* very fast gates, advanced experimental demonstration, straightforward fabrication.

—*weaknesses.* low coherence time relative to measurement time, sensitivity to background charge fluctuations and local magnetic fields.

4.3 All-Silicon NMR

Ladd et al. have proposed an all-silicon NMR-based quantum computer which stores qubits in the nuclear spin of ^{29}Si (spin 1/2 nucleus) laid down in a line across a micromechanical bridge of spin 0 nuclei (^{28}Si and ^{30}Si) [2002], as shown in Figure 2. This is an ensemble system; 10^5 copies are required to get adequate signal for measurement. Readout is done via magnetic resonance force microscopy (MRFM), reading oscillations of the bridge. Initialization is done via electrons whose spins are set with polarized light (optical pumping). Operations are done via microwave radiation directed at the device. A micromagnet provides a high field gradient, allowing individual atoms to be addressed by frequency.

—*strengths.* longest known decoherence time;

—*weaknesses.* slow gates, no QIO, measurement still being designed.

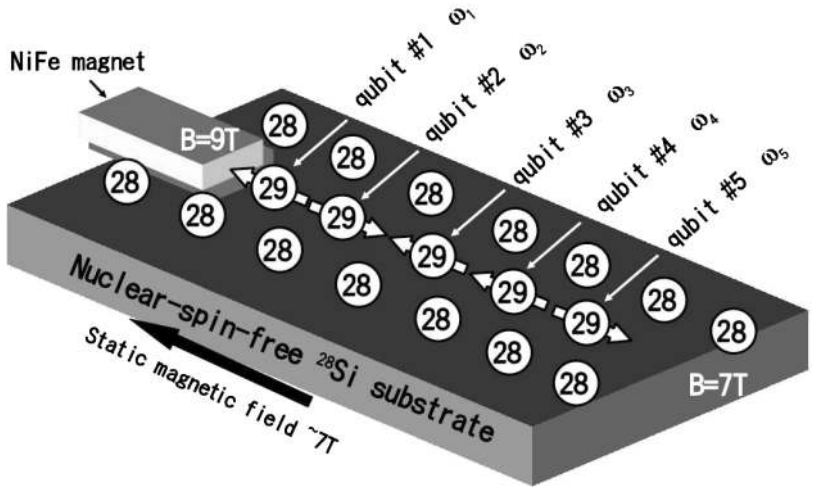

Fig. 2. Schematic of the all-silicon NMR computer. Qubits are the spin of ^{29}Si nuclei on a spin-free base of ^{28}Si. Distance from the micromagnet determines oscillation frequency and provides individual qubit addressability. Image courtesy of K. M. Itoh, Keio University.

4.4 Scalable Ion Trap

One of the few systems which explicitly separates storage areas from interaction areas is the scalable ion trap [Kielpinski et al. 2002; Wineland et al. 2005; Kim et al. 2005; Metodiev et al. 2003]. Initially designed and built at NIST, this is a proposal to scale up an ion trap quantum computer [Cirac and Zoller 1995; Steane 1997; Sørensen and Mølmer 2000; Schmidt-Kaler et al. 2003]. In ion trap systems, qubits are usually stored in the energy levels of individual ions. Early ion trap experiments featuring small numbers of ions held in a single trap have given way to a large system of interconnected, individually controllable traps. In the scalable trap system, the ions are literally moved around using magnetic fields until they reach locations in the system designated for operations, as shown in Figure 3. Small numbers of ions are brought together and formed into chains to execute multi-qubit gates. Gates are effected by laser pulses, and readout is also accomplished by laser pulses creating flourescence (interpreted as a 1) or not (0), depending on the state of the atom. Gate times are moderate; speed can be traded off against fidelity in the range of 14-100kHz. Overall system performance will likely be driven by ion movement times (which naturally depend on distance and topology), times for creating and splitting chains of atoms, time to cool atoms heated by the movement process, and multiplexing of gate operations. The movement operations are unlikely to allow a gate rate in excess of 20kHz.

—*strengths.* scalability of storage;
—*weaknesses.* slow gates [Steane et al. 2000]; limitations on concurrent operations and measurements.

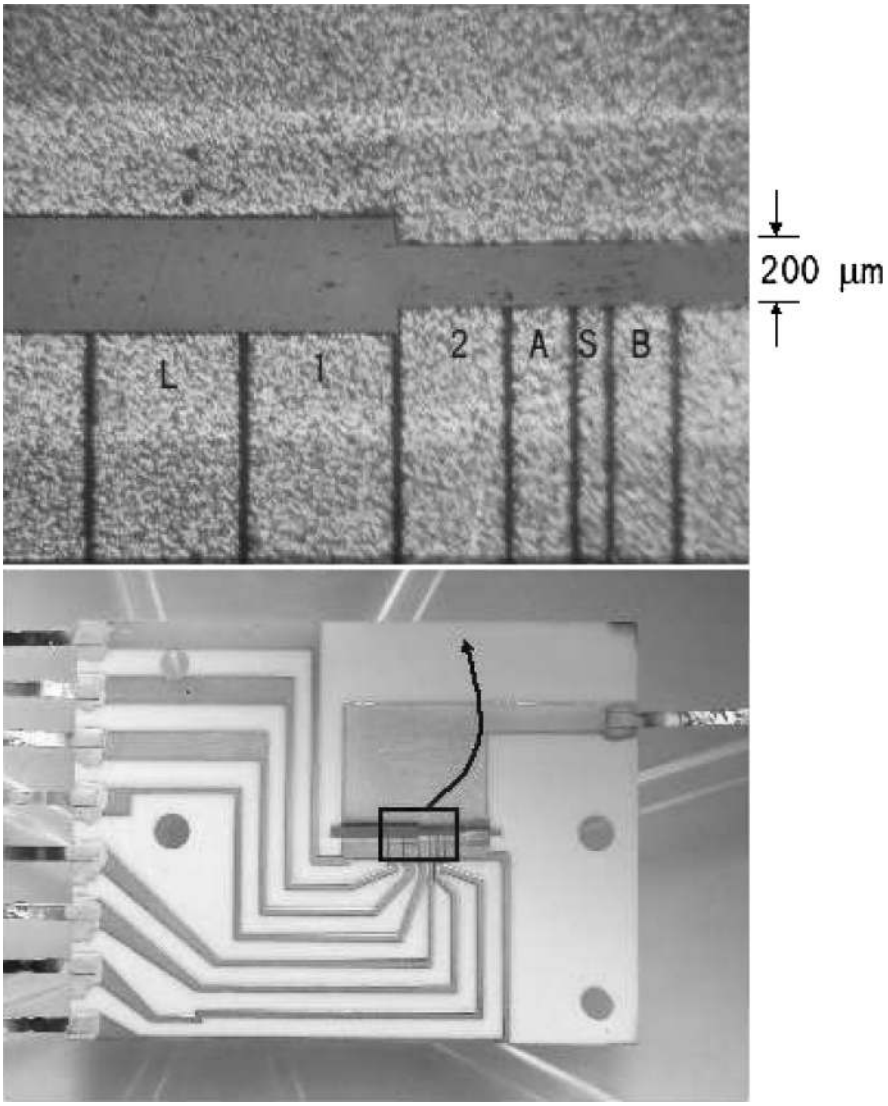

Fig. 3. A six-zone ion trap capable of moving individual ions. Ions are inserted in the landing zone L, and manipulated in the zones A, S, and B. Image courtesy of D. Wineland, NIST.

4.5 All-Optical

All-optical systems come in two flavors: those that depend on nonlinear effects to execute gates, and those in which the only necessary nonlinearity is measurement, known as *LOQC* (linear optics quantum computation) [Knill et al. 2001]. Research on all-optical systems has focused on photon sources capable of generating precise numbers of photons with the necessary timing precision [Santori et al. 2002], gates based on measurement [Knill et al. 2001;

Scheel et al. 2003; Knill 2003; Browne and Rudolph 2005; Yoran and Reznik 2003], and high-quality single-photon detectors [Miller et al. 2003; Waks et al. 2003; James and Kwiat 2002].

Measurement-based gates are inherently probabilistic in nature, though it has been shown that these gates can be built into a scalable feed-forward network [Knill et al. 2001; Ralph et al. 2005]. Much of the current experimental work is focusing on this approach, and individual gates have been shown to work [Pittman et al. 2002; O'Brien et al. 2003; Pittman et al. 2003; Gasparoni et al. 2004; Sanaka et al. 2004].

Jitter and skew are likely to be managed by "stopped light," created by electromagnetically-induced transparency [Fleischhauer and Lukin 2000; Harris 1997].

—*strengths.* well-understood physics and easy fabrication;

—*weaknesses.* photon losses; for nonlinear systems, weak nonlinear effects give poor gate quality; high resource requirements for probabilistic gates.

4.6 Quantum Dot

A "quantum dot," as used in quantum information processing, is a lithographically-defined structure that confines electrons at the boundary layer between two materials, creating a two-dimensional electron gas (2DEG). By varying the surrounding electrical potential, individual electrons can be positioned in a small area, called the quantum dot. A qubit can be defined based on the number of electrons in a quantum dot or the spin or energy levels of a single electron held in a quantum dot.

Several quantum dot devices are under development; one experimentally advanced approach uses a pair of quantum dots as a dual-rail encoded logical qubit, with a single electron in the left dot representing a logical 0, and the electron in the right dot representing a logical 1 [Fujisawa et al. 1998]. Another approach uses a linear array of single-electron quantum dots, and encodes the qubit in the spin of the excess electron [Loss and DiVincenzo 1998].

In a third approach, DiVincenzo et al. [2000] proposed that the only operation needed is an exchange between two neighboring qubits, accomplished by lowering the electrical potential and allowing the electrons to tunnel [DiVincenzo et al. 2000; Loss and DiVincenzo 1998; Myrgren and Whaley 2003]. This is easier to accomplish than precise control of a magnetic field, which would be required in order to effect other gates on specifically addressable bits.

Perhaps the biggest drawback of this approach is that exchange-only computation requires encoding a single logical qubit onto multiple physical qubits. A CNOT, for example, requires each logical qubit to be encoded in three physical qubits, and the exchange times must be controlled fairly precisely. The CNOT on neighboring logical qubits requires 19 exchange operations [DiVincenzo et al. 2000], though Myrgren and Whaley [2003] have found interesting optimizations that allow non-neighbor operations to be effected in 28% fewer total operations than the obvious formulation of repeated use of the 19-exchange CNOT.

Continued compiler work may reduce the encoded execution time penalty further, though the important storage penalty remains.

—*strengths*. advanced fabrication;
—*weaknesses*. low coherence time.

4.7 Kane Solid-State NMR

Kane has proposed a solid-state NMR system with excellent scalability, built on VLSI techniques for control [Kane 1998]; Oskin, Copsey et al. have followed that work with engineering studies, suggesting that teleportation may be required to move qubits long distances even for error correction [Oskin et al. 2003; Copsey et al. 2003], and progress in fabrication has been made [Clark et al. 2003]. In this system, individual phosphorus atoms are embedded in a silicon substrate, and standard photolithography techniques are used to build control structures on the surface. The qubit is held in the spin of the phosphorus nucleus, and interactions between neighboring qubits are mediated by electrons coupled to the nuclei via hyperfine interactions. The shape of the electron wave function is controlled via the control structures built on the Si surface; the distance between neighboring P atoms and the accuracy of aligning the control gates to the P impurities will determine the quality of qubit interactions.

—*strengths*. long coherence time;
—*weaknesses*. difficult fabrication, creating adequate overlap in electron wave functions.

5. ARCHITECTURE AND APPLICATIONS

In this section, we show some of the influences of technology on architecture by examining quantum arithmetic, the various forms of error management, and the new approach of cluster state computing.

5.1 Arithmetic

The current world record for factoring is 576 bits [RSA Security Inc. 2004]. The previous world record, 530 bits, was accomplished in one month using 104 PCs and workstations manufactured in 2003. Based on Moore's Law for CPU speed alone (ignoring communications, memory, and I/O), the largest number factorable by the number field sieve (NFS) should be growing at about 18 bits per year in the current range of 500–600 bits, and this is indeed what the RSA Challenge numbers show [Knuth 1998]. Although earlier estimates place the factoring of a 1024-bit number as close as the year 2018, we speculate, based on recent progress, that it may remain out of reach for another 25 years [Cavallar et al. 2000; Lenstra et al. 2003].

Using a quantum computer, factoring a 576-bit number in one month using Shor's algorithm requires, first of all, that a coherent state be maintained for a month across several thousand logical qubits, through the use of quantum error correction. Second, a logical clock speed of 0.3Hz to 27Hz is required [Van

Meter et al. 2005]. In this section, we use arithmetic as an example of how the architecture of a quantum computer affects both the asymptotic ($O(\cdot)$) performance of an algorithm and its constant factors [Ercegovac and Lang 2004]. Some algorithms are fixed in their resource requirements and execution schedule, while others can utilize increased space and concurrency, enabling greater parallelism and reducing the circuit depth.

Consider, for example, the modular exponentiation that is the most computationally intensive part of Shor's factoring algorithm. Researchers have worked on the overall exponentiation algorithm [Vedral et al. 1996; Beckman et al. 1996; Zalka 1998], and on arithmetic building blocks [Gossett 1998; Draper 2000; Cuccaro et al. 2004]. Comparatively little, however, has been done on mapping those algorithms to realistic architectures, with or without error correction [Devitt et al. 2004; Fowler et al. 2004; Yimsiriwattana and Lomonaco Jr. 2004; Van Meter and Itoh 2005; Vartiainen et al. 2004].

It has been shown that, although the computational complexity is $O(n^3)$, the circuit depth is $O(\log^3 n)$ to exponentiate an n-bit number when three important criteria are met: the architecture supports concurrent gates on all qubits, $O(n^3)$ application-level qubits are available, and any two qubits anywhere in the system can interact without penalty [Cleve and Watrous 2000; Van Meter and Itoh 2005].

When these criteria cannot be met, performance suffers. When only $O(n^2)$ qubits are available, algorithmic parallelism is limited, and circuit depth increases to $O(n \log^2 n)$. A commonly proposed architecture is a linear chain of qubits with nearest-neighbor-only interaction, which increases the depth to $O(n^2 \log n)$. Further reduction in the amount of space available again increases the asymptotic depth. The slow 0.3Hz quantum computer suggested above assumes the highest possible parallelism and support for long-distance gates, whereas the the 27Hz computer uses high parallelism, but nearest-neighbor-only interactions.

Figures 4 and 5 show two types of quantum adder circuits, the Vedral-Barenco-Ekert (VBE) carry-ripple adder [Vedral et al. 1996] and the Draper-Kutin-Rains-Svore carry-lookahead adder [Draper et al. 2004]. The first, most obvious difference between the two is how "busy" the diagrams appear. The carry-ripple adder shows that most of the qubits sit idle during most of the computation, waiting for the carry to ripple across the circuit (and back, as a cleanup operation). The carry-lookahead adder is much denser, accomplishing its work in fewer timesteps by executing more gates in parallel.

The second most prominent visual difference is the span of the gates (vertical line segments). Carry-ripple adders operate only on qubits that are nearby, while the carry-lookahead adder leapfrogs long distances. This gives the carry-ripple adder $O(n)$ latency, compared to $O(\log n)$ for the carry-lookahead—if long-distance gates are supported. This format of circuit diagram abstracts away the physical layout of qubits, and for any layout other than linear nearest neighbor, gives the wrong impression of "nearby." Therefore, we have begun animating the action of some circuits for more complex topologies [Van Meter 2005].

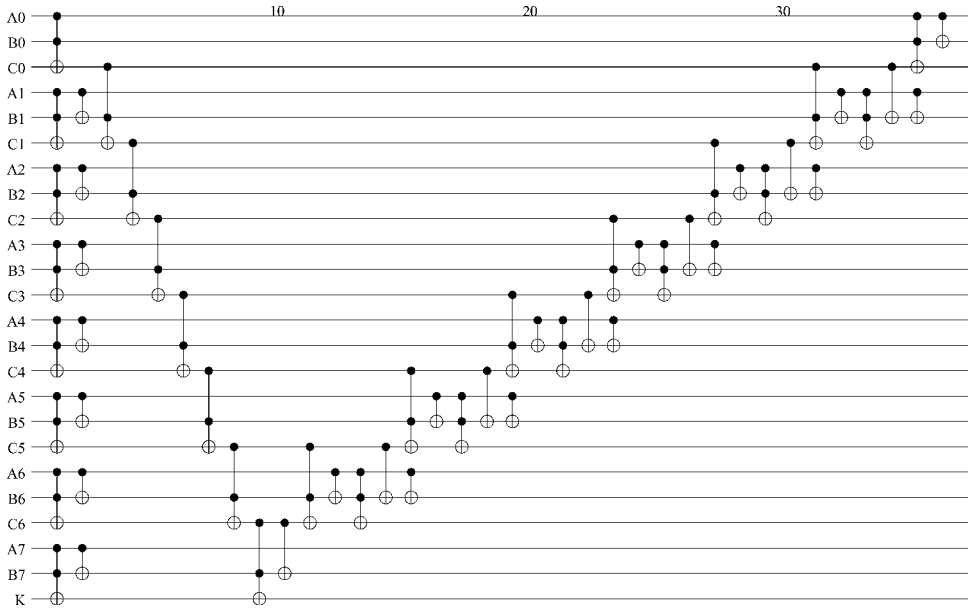

Fig. 4. Eight-bit quantum VBE carry-ripple adder. Each horizontal line represents a qubit; the horizontal axis is time (indicated at the top of the figure). Each vertical line segment or circle represents a gate on one, two, or three qubits. The "V" shape of the circuit shows the ripple of the carry down and back (necessary to clean temporary variables).

In Table VI we list recommendations for adders that match various technologies. For example, the Fourier adder [Draper 2000] uses only $2n$ space, compared to the $3n$ of standard carry-ripple adders [Vedral et al. 1996; Beckman et al. 1996], but requires n concurrent gates to achieve the $O(n)$ time bound when performing the quantum Fourier transform (QFT) required to move numbers into and out of the Fourier representation, compared to concurrency of 2 for carry-ripple. The Fourier adder also requires precise rotations similar to those in the QFT, which may be hard to implement accurately. A newly designed carry-ripple adder uses only $2n$ space and small concurrency, making it now the preferred choice in many cases [Cuccaro et al. 2004].

Likewise, some entries recommend both the conditional-sum and carry-lookahead adders, which have almost identical $O(\log n)$ latencies. A conditional-sum adder requires more space and concurrency than carry-lookahead. However, it has different locality characteristics which might make it map better to an irregular architecture. In particular, the scalable ion trap has limited concurrency, but the distance an ion must move may have a factor of two or more performance impact, making locality desirable; the design of such a system is not yet advanced enough to definitively choose between the two proposed types of adders.

For the two-dimensional layout of the Kane lattice, an ideal $O(\log n)$ adder can reach latency of only $O(\sqrt{n})$ due to the communications cost of moving qubits.

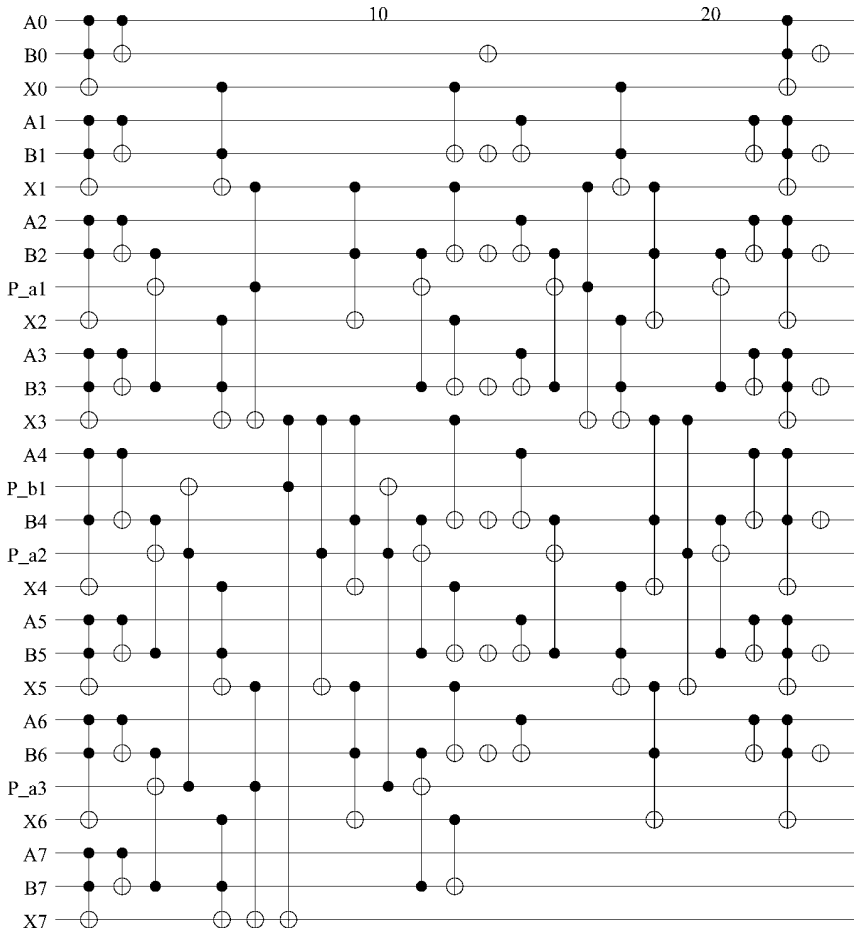

Fig. 5. Eight-bit quantum carry-lookahead adder. Each horizontal line represents a qubit; the horizontal axis is time (indicated at the top of the figure). Each vertical line segment or circle represents a gate on one, two, or three qubits.

For the Josephson-junction qubits, we recommend using long-distance inductive or capacitive transfer structures only if concurrent operations can be preserved for at least some qubits. Alternating cycles of a single long-distance interaction and many nearest-neighbor interactions would be adequate. Designs in which only some of the qubits can transfer long distances while others execute concurrent nearest-neighbor operations seem physically plausible, and would result in intermediate performance, possibly using a carry-select or conditional-sum adder. Concrete performance analysis will depend on the details of such a heterogeneous architecture.

The issue of concurrency highlights an important factor in system design: depending on the quantum memory error rate compared to the fault-tolerant gate error rate, and the execution times of FT gates and QEC procedures, a large portion of the computation time may be spent in preventing “idle” qubits from developing uncorrectable errors [Steane 2003; Devitt et al. 2004].

Table VI. Qubit Technologies and Recommended Choice of Adder (conc., required application-level concurrency.)

Technology	Adder	Conc.	Latency
Si NMR	carry-ripple	2	$O(n)$
solution NMR	carry-ripple	2	$O(n)$
1-D quantum dot	carry-ripple, Fourier	2 or n	$O(n)$
1-D JJ charge	carry-ripple, Fourier	2 or n	$O(n)$
1-D Kane model	carry-ripple, Fourier	2 or n	$O(n)$
scalable ion trap	carry-lookahead, conditional-sum	n or $2n$	$O(\log n)$
Oskin lattice	carry-lookahead, conditional-sum	n or $2n$	$O(\sqrt{n})$
all-optical	carry-lookahead, conditional-sum	n or $2n$	$O(\log n)$

5.2 Error Management

The field of error management actually consists of several types of techniques, each of which protects against different error processes. Quantum error correction (QEC) and the associated fault-tolerant (FT) methods are based on classical error-correction codes. The first and most important class of these is the Calderbank-Shor-Steane (CSS) codes [Calderbank and Shor 1996; Shor 1996; Steane 1996; Preskill 1998b]. The theory and practice (including both experimental demonstrations [Chiaverini et al. 2004; Pittman et al. 2005; Roos et al. 2004] and system design [Svore et al. 2005; Steane 2002; Copsey et al. 2003; Burkard et al. 1999; Devitt et al. 2004; Metodiev et al. 2003; Szkopek et al. 2004]) of QEC and FT operation are vast; we will not cover them in any depth here. Nevertheless, a basic understanding of the pressures that QEC and FT place on architecture is critical. QEC and FT demand the continuous preparation and measurement of a set of ancillae (temporary work) qubits, and raise the overall cost of quantum computation by as much as four orders of magnitude for *each* level of QEC built into the system—and it appears that two or more levels may be necessary. The logical clock speed of the system will correspond roughly to the QEC cycle time, and is correspondingly slower than the physical clock speed, though the exact ratio will depend on both technology- and machine-dependent details.

These QEC codes encode one or more qubits into a code word. The error syndromes on this code word are continuously calculated and measured, and corrective actions applied to the code word. The measurement of the syndrome actually effects a key portion of the error control process; it forces ("projects") the state either back into a good state (with high probability) and returns a zero (no error) syndrome, or an error state (with low probability) and returns a non-zero syndrome. When the syndrome is non-zero, one or two corrective gates are indicated and applied. Unfortunately, this syndrome calculation and measurement process may also introduce errors. Technologies that support nearest-neighbor-only interactions require swapping of qubits in order to calculate the error syndrome, with the swap gates possibly introducing errors themselves, making the threshold requirements for effective error correction more stringent; in some studies, as much as 175 times worse [Svore et al.

2005; Steane 2003; Aharonov and Ben-Or 1999; Fowler et al. 2004; Szkopek et al. 2004]. Applying two-qubit gates can result in the propagation of an error from one qubit to another, even from the target of the gate to its control. The parity calculations necessary to retrieve the error syndrome, then, cannot be carried out directly, but must operate indirectly on an entangled state prepared especially to defend against this propagation. That state preparation requires as many qubits as the code word itself, and may be the driver of the cycle time for QEC. Measurement of qubit state on some technologies is slow compared to the gate time, so this also figures prominently into the cycle time.

The overhead of QEC is very high compared to classical systems. A single qubit may be encoded in five to nine qubits just to protect against a single error. The five-qubit encoding is difficult to manipulate, so the seven-bit Steane code is generally used in the literature. Larger encodings are of course possible, borrowing from classical systems; the storage ratio of those generally remains in the 3:1 area even for moderately large blocks. Logical operations on logical qubits stored in the same block become difficult, so qubits are swapped out of the block, logical gates are performed, then the qubits are swapped back [Steane 2003; Steane and Ibinson 2003].

QEC generally deals with independent single-qubit errors. Some error processes (e.g., stray magnetic fields) may affect nearby qubits similarly. Techniques known as *decoherence free subspaces* (DFSes) help defend against these by, effectively, encoding a logical qubit in the *difference*, or delta, between two or more qubits [Lidar et al. 1998; Lidar and Whaley 2003]. Techniques derived from classical erasure codes, such as those typically used in RAID arrays, can be helpful in optical systems where photon loss is a key error process [Knill et al. 2000]. QEC, via the discretization of states forced by the error syndrome measurement, helps to control against small errors in the analog state of qubits (small over- or under-rotations in gates)[1]. Other techniques based on NMR combine multiple sub-gate rotations into a single gate and suppress rotation errors directly during single-qubit operations [Vandersypen and Chuang 2004].

As qubits are subject to error processes when idle as well as while being used, the total amount of error correction in the system is dependent on the size of the machine as well as the number of logical gates being executed. If each qubit must be "refreshed" at one-tenth the QEC cycle rate, for example, then we must build a system in which one-tenth of the qubits can all be undergoing QEC at the same time. Longer waits for correction increase the probability of error; this must be balanced against the number of levels of QEC and the engineering difficulties of initialization and measurement. Quantum dots and superconducting qubits require additional on-chip structures to perform measurement [Petta et al. 2005]. This will limit layout flexibility and consume die space. If possible, it will be desirable to perform entire QEC sequences on-chip; however, in the short run, it may be necessary to use off-chip signal generators and control circuitry, requiring a wide, high-bandwidth I/O interface from the chip itself. For scalable ion trap systems, many laser beams must excite

[1]This is a key factor in the "quantum computation is not analog computation" argument.

many ions. Complex optics and photon detectors may be required to read out the state of many qubits at once; CCD cameras involve a direct tradeoff of speed versus noise, while avalanche photodiodes are difficult to integrate and photon counters require cryogenic operation [Kim et al. 2005].

To manage errors effectively, then, we can say that a technology must support large numbers of concurrent qubit state preparations, gates, and measurements. As the required operations are much more complex than a DRAM refresh cycle and are close to the universal gate set, a large-scale difference in structure akin to the CPU/RAM dichotomy of classical systems is unlikely. However, at the small scale, systems which store qubits in nuclear spins while idle and shift to electron spins for active gates have been proposed [Steane and Lucas 2000; Kane 1998; Mehring et al. 2003; Jelezko et al. 2004; Childress et al. 2005].

5.3 Cluster-State Computing

Perhaps the most exciting theoretical advance in recent years in quantum computing is the development of *cluster-state computing*, or *one-way computation* [Raussendorf et al. 2003; Nielsen 2005; Walther et al. 2005]. In theory, cluster-state computing offers a broader mathematical palette from which to design computations; in practice, the best-understood use is as a substrate on which the circuit model described above is implemented. Cluster-state computing uses quantum computational capabilities rather differently from the basic circuit model. Prior to the beginning of the computation proper, a very large entangled state called the *cluster state* is created. Computation itself is executed via single-qubit measurements only, conducted in specific sequences and along axes sometimes chosen dynamically, based on prior measurement outcomes. The growth of the cluster state can be probabilistic but *heralded*; that is, it is known whether or not the operation succeeded, allowing the operation to be retried if necessary. This meshes well with the capabilities of linear optics quantum computation (LOQC), so, although cluster-state computation is an abstraction, much current research excitement exists over the possibility of realizable quantum computation based on these technologies [Knill et al. 2001; Nielsen 2004].

Cluster-state computation, like all forms of quantum computation, is subject to error processes that require management. Recent research has proposed using the cluster state as the "bottom" architectural layer, with fault tolerance as a middle layer and the application algorithm on top [Nielsen and Dawson 2004; Dawson et al. 2005; Varnava et al. 2005]. Cluster-state-like techniques for fault tolerance have also been developed for the circuit model for optics [Ralph et al. 2005].

In its most naive form, cluster-state computation requires enormous physical resources. The circuit diagrams shown in section 5.1 have a vertical dimension that corresponds to physical resources, and a horizontal dimension that corresponds to time. In cluster-state computation, the physical resources assumed correspond to the entire area of such a diagram, multiplied by a factor to account for the resources of a single cluster state. The time dimension

"emerges anew" [Raussendorf et al. 2003]; for some circuits, once the cluster state construction is complete the entire algorithm, composed of single-qubit measurements, can be executed in a single time step. In practice, there are two obstacles. The first is the resources required; for example, a VBE adder uses one hundred times as many qubits in the cluster-state model as it does in the circuit model.[2] The second is that most "interesting" algorithms, including any that use the Toffoli (control-control-NOT) gate, cannot be executed concurrently because the choice of measurement axis for the qubits is dependent on prior results. The execution time for a VBE adder, for example, is $O(n)$, the same as in the more conventional circuit model. The first issue is mitigated somewhat because cluster-state creation, in linear optics, is far more efficient than direct gate execution. Both issues are alleviated by using a sliding window in the computation and creating the cluster state in a just-in-time fashion. This allows for fewer resources, and relaxes error management constraints as well. The amount of cluster state that must be buffered depends on the details of the success probability in growing the cluster state. The physical system must either support a logical topology that can be configured as a cylinder, wrapping the right (leading) edge of the cluster back around onto the beginning, or the entire cluster state must shift through the machine as resources are consumed on the left and demanded on the right.

The basic cluster state is logically a two-dimensional mesh connecting nearest Manhattan neighbors (although other topologies have been proposed for specific purposes). It can be constructed on any physically realizable topology.

Photons naturally travel long distances well; it would thus seem that cluster-state computation is not taking full advantage of this capability. Where gate operands are a long logical distance apart, as in Figure 5, a "wire" is laid out across the cluster state. This consumes spatial resources, but the entire wire can be executed in a single timestep, moving a state from one part of the cluster to another quickly.

Finally, a problem with photon-based quantum computation is storage. One recent proposal suggests using individual solid-state qubits and an optical interconnect to build cluster states [Lim et al. 2005]. With a suitably rich interconnect, this should work well.

6. CONCLUSIONS

This article, so far as we are aware, is the first attempt to organize information about quantum computers in a way that specifically focuses on scalability, implementability, and architectural implications. We have broken down quantum technologies into stationary and flying qubits, and within the stationary have seen a subclass of mobile qubits. Stationary qubit technologies include both single and ensemble systems. The evaluation criteria we have laid out should make it possible to compare technologies and determine which will be useful in

[2]A naive mapping of the VBE adder would grow the size of the cluster state quadratically in the length of the numbers to be added; Raussendorf et al. show how to perform the mapping with linear growth in resources, and only a constant factor larger than the circuit implementation.

different roles of a system, and how application algorithms can be mapped to and compiled for various architectures.

Each of the technologies discussed here has its own particular set of technological hurdles to overcome before it can be considered practical. NMR-based systems have slow gate times, but have good coherence times; if a QIO mechanism can be designed [Wallraff et al. 2004], they will make excellent storage devices, but pure NMR systems are unlikely to make adequate factoring machines. Josephson-junction devices and quantum dots have extremely fast gate times, but have poor coherence times. Both of these systems have yet to demonstrate scalability in implementation and addressing of qubits, though both have been designed. Pure optical systems need more efficient single-photon detectors. Ion traps have many desirable features that make them scalable architecturally.

The complex tradeoffs in controlling a quantum computer include trading speed for coherence time. The quantum wiring and classical control are under investigation in both technology-dependent and -independent fashions, but many scaling questions remain. Work on both programming language design to support quantum computation and backend optimization for specific architectural characteristics has just begun [Ömer 2002; Aho and Svore 2003; Nakajima et al. 2005; Kawano et al. 2005]. Finally, the mapping of algorithms to these architectures will determine the performance and practicality of particular architectures.

The most prominent proposed use of quantum computers is Shor's algorithm for factoring large numbers, which has the potential to make the widely used RSA public-key cryptosystem and Diffie-Hellman key exchange protocol insecure. The encrypting operations and the execution of Shor's algorithm are, not coincidentally, both $O(n^3)$ for n-bit keys. Both manufacturing and operating costs for qubits and quantum gates will remain many orders of magnitude more expensive than classical bits and gates for the foreseeable future. Classical systems can therefore afford to go to larger key lengths far more easily than a quantum system, staying ahead in the cryptographic arms race (although this cost must be borne by all users, not those breaking the codes). However, the known existence (or even imminent delivery) of even a single large quantum computer may prompt a shift away from cryptosystems perceived to be vulnerable.[3] Thus, Shor's algorithm alone is unlikely to be adequate economic incentive for the development and purchase of more than a handful of large quantum computers. We look forward to the continued development of important quantum computing algorithms.

The field of quantum computer architecture can be said to be in its infancy. Researchers who have focused on the fundamental technologies are now beginning to examine how to build complete quantum computing systems. In practice, more than one of these technologies may come to fruition, providing system architects with a suite of capabilities from which to fashion complete systems. This is a scenario to be hoped for as quantum computing devices develop into systems that can solve real-world problems.

[3]We wish to point out here that quantum key distribution [Bennett and Brassard 1984; Elliott et al. 2003] does not solve the problems that Shor's algorithm creates [Paterson et al. 2004].

ACKNOWLEDGMENTS

The authors thank K. M. Itoh, T. D. Ladd, K. Nemoto and W. J. Munro for reviewing earlier drafts of this paper. We thank Y. Nakamura, T. Yamamoto, D. Wineland, and K. M. Itoh for the figures.

REFERENCES

AHARONOV, D. AND BEN-OR, M. 1999. Fault-tolerant quantum computation with constant error rate. http://arXiv.org/quant-ph/9906129. (Extended version of STOC 1997 paper.)

AHO, A. V. AND SVORE, K. M. 2003. Compiling quantum circuits using the palindrome transform. http://arXiv.org/quant-ph/0311008.

AMDAHL, G. 1967. Validity of the single processor approach to achieving large-scale computing capabilities. In *AFIPS Conference Proceedings*. 483–485.

ARDA. 2004. *A Quantum Information Science and Technology Roadmap*, v2.0 ed. ARDA.

BARENCO, A., BENNETT, C. H., CLEVE, R., DIVINCENZO, D. P., MARGOLUS, N., SHOR, P., SLEATOR, T., SMOLIN, J., AND WEINFURTER, H. 1995. Elementary gates for quantum computation. *Phys. Rev. A 52*, 3457.

BARENCO, A., EKERT, A., SUOMINEN, K.-A., AND TÖRMA, P. 1996. Approximate quantum Fourier transform and decoherence. *Phy. Rev. A 54*, 139–146.

BECKMAN, D., CHARI, A. N., DEVABHAKTUNI, S., AND PRESKILL, J. 1996. Efficient networks for quantum factoring. *Phys. Rev. A 54*, 1034–1063. http://arXiv.org/quant-ph/9602016.

BENNETT, C. H. AND BRASSARD, G. 1984. Quantum cryptography: Public key distribution and coin tossing. In *Proceedings of the IEEE International Conference on Computers, Systems, and Signal Processing*. IEEE. 175–179.

BENNETT, C. H., BRASSARD, G., CRÉPEAU, C., JOSZA, R., PERES, A., AND WOOTTERS, W. 1993. Teleporting an unknown quantum state via dual classical and EPR channels. *Phy. Rev. Lett. 70*, 1895–1899.

BOULANT, N., EDMONDS, K., YANG, J., PRAVIA, M. A., AND CORY, D. G. 2003. Experimental demonstration of an entanglement swapping operation and improved control in NMR quantum-information processing. *Phy. Rev. A 68*, 032305.

BRENNEN, G. K., CAVES, C. M., JESSEN, P. S., AND DEUTSCH, I. H. 1999. Quantum logic gates in optical lattices. *Phys. Rev. Lett. 82*, 5 (Feb.), 1060–1063.

BROWNE, D. E. AND RUDOLPH, T. 2005. Resource-efficient linear optical quantum computation. *Phys. Rev. Lett. 95*, 010501.

BURKARD, G., LOSS, D., DIVINCENZO, D. P., AND SMOLIN, J. A. 1999. Physical optimization of quantum error correction circuits. *Phys. Rev. B 60*, 16, 11404–11416.

CALDERBANK, A. R. AND SHOR, P. W. 1996. Good quantum error-correcting codes exist. *Phys. Rev. A 54*, 1098–1105.

CAVALLAR, S., DODSON, B., LENSTRA, A. K., LIOEN, W., MONTGOMERY, P. L., MURPHY, B., TE RIELE, H., AARDAL, K., GILCHRIST, J., GUILLERM, G., LEYLAND, P., MARCHAND, J., MORAIN, F., MUFFETT, A., PUTNAM, C., PUTNAM, C., AND ZIMMERMANN, P. 2000. Factorization of a 512-bit RSA modulus. In *Advances in Cryptology—EUROCRYPT 2000: International Conference on the Theory and Application of Cryptographic Techniques*. Lecture Notes in Computer Science, vol. 1807. 1. Springer-Verleg, New York, 2000.

CHIAVERINI, J., LEIBFRIED, D., SCHAETZ, T., BARRETT, M. D., BLAKESTAD, R. B., BRITTON, J., ITANO, W. M., JOST, J. D., KNILL, E., LANGER, C., OZERI, R., AND WINELAND, D. J. 2004. Realization of quantum error correction. *Nature 432*, 602–605.

CHILDRESS, L., TAYLOR, J. M., SØRENSEN, A. S., AND LUKIN, M. 2005. Fault-tolerant quantum repeaters with minimal physical resources, and implementations based on single-photon emitters. http://arXiv.org/quant-ph/0502112.

CHIORESCU, I., BERTET, P., SEMBA, K., NAKAMURA, Y., HARMANS, C. J. P. M., AND MOOIJ, J. E. 2004. Coherent dynamics of a flux qubit coupled to a harmonic oscillator. *Nature 431*, 159–162.

CIRAC, J. I. AND ZOLLER, P. 1995. Quantum computations with cold trapped ions. *Phys. Rev. Lett. 74*, 4091–4094.

CLARK, R. G. ET AL. 2003. Progress in silicon-based quantum computing. *Phil. Trans. R. Soc. London A 361*, 1451–1471.

CLEVE, R. AND WATROUS, J. 2000. Fast parallel circuits for the quantum Fourier transform. In *Proceedings of the 41st Annual Symposium on Foundations of Computer Science*. ACM, New York, 526–536.

COPSEY, D., OSKIN, M., METODIEV, T., CHONG, F. T., CHUANG, I., AND KUBIATOWICZ, J. 2003. The effect of communication costs in solid-state quantum computing architectures. In *Proceedings of the 15th Annual ACM Symposium on Parallel Algorithms and Architectures*. 65–74.

CORY, D. G. 2004. Private communication.

CUCCARO, S. A., DRAPER, T. G., KUTIN, S. A., AND MOULTON, D. P. 2004. A new quantum ripple-carry addition circuit. http://arXiv.org/quant-ph/0410184.

DAWSON, C. M., HASELGROVE, H. L., AND NIELSEN, M. A. 2005. Noise thresholds for optical quantum computers. http://arXiv.org/quant-ph/0509060.

DEUTSCH, D. AND JOZSA, R. 1992. Rapid solution of problems by quantum computation. *Proc. R. Soc. London Ser. A, 439*, 553.

DEVITT, S. J., FOWLER, A. G., AND HOLLENBERG, L. C. 2004. Simulations of Shor's algorithm with implications to scaling and quantum error correction. http://arXiv.org/quant-ph/0408081.

DIVINCENZO, D. P. 1994. Two-bit gates are universal for quantum computation. *Phys. Rev. A*. http://arXiv.org/cond-mat/9407022.

DIVINCENZO, D. P. 1995. Quantum computation. *Science 270*, 5234, 255–261.

DIVINCENZO, D. P., BACON, D., KEMPE, J., BURKARD, G., AND WHALEY, K. B. 2000. Universal quantum computation with the exchange interaction. *Nature 408*, 339–342.

DRAPER, T. G. 2000. Addition on a quantum computer. http://arXiv.org/quant-ph/0008033. (First draft dated Sept. 1998.)

DRAPER, T. G., KUTIN, S. A., RAINS, E. M., AND SVORE, K. M. 2004. A logarithmic-depth quantum carry-lookahead adder. http://arXiv.org/quant-ph/0406142.

ELLIOTT, C., PEARSON, D., AND TROXEL, G. 2003. Quantum cryptography in practice. In *Proceedings of the SIGCOMM 2003*. ACM, New York. http://arXiv.org/quant-ph/0307049.

ERCEGOVAC, M. D. AND LANG, T. 2004. *Digital Arithmetic*. Morgan Kaufmann, San Francisco, CA.

ESIA, JEITIA, KSIA, TSIA, AND SIA. 2003. International technology roadmap for semiconductors. Tech. Rep., ESIA and JEITIA and KSIA and TSIA and SIA. http://public.itrs.net/Files/2003ITRS/Home2003.htm.

FLEISCHHAUER, M. AND LUKIN, M. D. 2000. Dark-state polaritons in electromagnetically induced transparency. *Phys. Rev. Lett. 84*, 5094–5097.

FOLMAN, R., KRÜGER, P., CASSETTARI, D., HESSMO, B., MAIER, T., AND SCHMIEDMAYER, J. 2000. Controlling cold atoms using nanofabricated surfaces: Atom chips. *Phys. Rev. Lett. 84*, 4749–4752.

FOWLER, A. G., DEVITT, S. J., AND HOLLENBERG, L. C. 2004. Implementation of Shor's algorithm on a linear nearest neighbor qubit array. *Quantum Inf. Comput. 4*, 4, 237. http://arXiv.org/quant-ph/0402196.

FOWLER, A. G., HILL, C. D., AND HOLLENBERG, L. C. L. 2004. Quantum error correction on linear nearest neighbor qubit arrays. *Phys. Rev. A 69*, 042314.

FUJISAWA, T., OOSTERKAMP, T. H., VAN DER WIEL, W. G., BROER, B. W., AGUADO, R., TARUCHA, S., AND KOUWENHOVEN, L. P. 1998. Spontaneous emission spectrum in double quantum dot devices. *Science 282*, 932–935.

FURUSAWA, A., SØRENSEN, J. L., BRAUNSTEIN, S. L., FUCHS, C. A., KIMBLE, H. J., AND POLZIK, E. S. 1998. Unconditional quantum teleportation. *Science 282*, 5389, 706–709.

GALINDO, A. AND MARTIN-DELGADO, M. A. 2002. Information and computation: Classical and quantum aspects. *Rev. Modern Phys. 74*, 347–423.

GASPARONI, S., PAN, J., WALTHER, P., RUDOLPH, T., AND ZEILINGER, A. 2004. Realization of a photonic controlled-NOT gate sufficient for quantum computation. *Phys. Rev. Lett. 93*, 020504.

GOSSETT, P. 1998. Quantum carry-save arithmetic. http://arXiv.org/quant-ph/9808061.

GOTTESMAN, D. 1999. Fault tolerant quantum computation with local gates. http://arXiv.org/quant-ph/9903099.

GOTTESMAN, D. AND CHUANG, I. L. 1999. Quantum teleportation is a universal computational primitive. *Nature 402*, 390–393.

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

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

HALES, L. AND HALLGREN, S. 2000. An improved quantum Fourier transform algorithm and applications. In *Proceedings of the 41st Annual Symposium on Foundations of Computer Science*. ACM, New York.

HARRIS, S. E. 1997. Electromagnetically induced transparency. *Phys. Today 50*, 7 (July), 36–42.

ISAILOVIC, N., WHITNEY, M., PATEL, Y., KUBIATOWICZ, J., COPSEY, D., CHONG, F. T., CHUANG, I. L., AND OSKIN, M. 2004. Datapath and control for quantum wires. *ACM Trans. Architecture and Code Optimization 1*, 1 (Mar.).

JAMES, D. F. V. AND KWIAT, P. G. 2002. Atomic-vapor-based high efficiency optical detectors with photon number resolution. *Phys. Rev. Lett. 89*, 183601.

JELEZKO, F., GAEBEL, T., POPA, I., DOMHAN, M., GRUBER, A., AND WRATCHTRUP, J. 2004. Observation of coherence oscillation of a single nuclear spin and realization of a two-qubit conditional quantum gate. *Phys. Rev. Lett. 93*, 130501.

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

KAWANO, Y., YAMASHITA, S., AND KITAGAWA, M. 2005. Explicit implementation of quantum circuits on a unidirectional periodic structure. *Phys. Rev. A 72*, 012301.

KIELPINSKI, D., MONROE, C., AND WINELAND, D. J. 2002. Architecture for a large-scale ion-trap quantum computer. *Nature 417*, 709–711.

KIM, J. ET AL. 2005. System design for large-scale ion trap quantum information processor. *Quantum Inf. and Comput. 5*, 7, 515–537.

KNILL, E. 2003. Bounds on the probability of success of postselected nonlinear sign shifts implemented with linear optics. *Phys. Rev. A 68*, 064303.

KNILL, E., LAFLAMME, R., AND MILBURN, G. J. 2000. Thresholds for linear optics quantum computation. http://arXiv.org/quant-ph/0006120.

KNILL, E., LAFLAMME, R., AND MILBURN, G. J. 2001. A scheme for efficient quantum computation with linear optics. *Nature 409*, 46–52.

KNUTH, D. E. 1998. *The Art of Computer Programming, vol. 2 / Seminumerical Algorithms*, 3rd ed. Addison-Wesley, Reading, MA.

KUNIHIRO, N. 2005. Exact analyses of computational time for factoring in quantum computers. *IEICE Trans. Fundamentals E88-A*, 1, 105–111.

LADD, T. D., GOLDMAN, J. R., YAMAGUCHI, F., YAMAMOTO, Y., ABE, E., AND ITOH, K. M. 2002. All-silicon quantum computer. *Phys. Rev. Lett. 89*, 1 (July), 17901.

LADD, T. D., MARYENKO, D., YAMAMOTO, Y., ABE, E., AND ITOH, K. M. 2003. Coherence time of a solid-state nuclear qubit. http://arXiv.org/quant-ph/0309164.

LANTZ, J., WALLQUIST, M., SHUMEIKO, V. S., AND WENDIN, G. 2004. Josephson junction qubit network with current-controlled interaction. *Phys. Rev. B 70*, 140507.

LENSTRA, A., TROMER, E., SHAMIR, A., KORTSMIT, W., DODSON, B., HUGHES, J., AND LEYLAND, P. 2003. Factoring estimates for a 1024-bit RSA modulus. In *AsiaCrypt 2003*. Lecture Notes in Computer Science. Springer-Verlag, New York.

LEUNG, D. W., CHUANG, I. L., YAMAGUCHI, F., AND YAMAMOTO, Y. 2000. Efficient implementation of selective recoupling in heteronuclear spin systems using Hadamard matrices. *Phys. Rev. A 61*. http://arXiv.org/quant-ph/9904100.

LIDAR, D. A., CHUANG, I. L., AND WHALEY, K. B. 1998. Decoherence-free subspaces for quantum computation. *Phys. Rev. Lett. 81*, 12 (Sept.), 2594–2597.

LIDAR, D. A. AND WHALEY, K. B. 2003. *Irreversible Quantum Dynamics*. Chapter Decoherence-Free Subspaces and Subsystems. Lecture Notes in Phys., vol. 622. Springer-Verlag, New York.

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. http://arXiv.org/quant-ph/0508218.

LLOYD, S. 1993. A potentially realizable quantum computer. *Science 261*, 1569–1571.

LLOYD, S., SHAHRIAR, M. S., AND HEMMER, P. 2000. Teleportation and the quantum internet. http://arXiv.org/quant-ph/0003147.

LOSS, D. AND DIVINCENZO, D. P. 1998. Quantum computation with quantum dots. *Phys. Rev. A 57*, 120.

MARTINIS, J. M., NAM, S., AUMENTADO, J., AND URBINA, C. 2002. Rabi oscillations in a large Josephson-junction qubit. *Phys. Rev. Lett. 89*, 117901.

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

MEHRING, M., MENDE, J., AND SCHERER, W. 2003. Entanglement between an electron and a nuclear spin 1/2. *Phys. Rev. Lett. 90*, 153001.

METODIEV, T., CROSS, A., THAKER, D., BROWN, K., COPSEY, D., CHONG, F. T., AND CHUANG, I. L. 2003. Preliminary results on simulating a scalable fault tolerant ion-trap system for quantum computation. In *Proceedings of the 3rd Workshop on Non-Silicon Computation* (NSC-3).

MILLER, A. J., NAM, S. W., MARTINIS, J. M., AND SERGIENKO, A. V. 2003. Demonstration of a low-noise near-infrared photon counter with multiphoton discrimination. *Appl. Phys. Lett. 83*, 791–793.

MOOIJ, J. E., ORLANDO, T. P., LEVITOV, L., TIAN, L., VAN DER WAL, C. H., AND LLOYD, S. 1999. Josephson persistent-current qubit. *Science 285*, 1036–1039.

MOORE, C. AND NILSSON, M. 2001. Parallel quantum computation and quantum codes. *SIAM J. Comput. 31*, 3, 799–815. http://arxiv.org/abs/quant-ph/9808027.

MOORE, G. E. 1965. Cramming more components onto integrated circuits. *Electronics 38*, 8 (Apr.).

MYRGREN, E. S. AND WHALEY, K. B. 2003. Implementing a quantum algorithm with exchange-coupled quantum dots: A feasibility study. *Quantum Inf. Proces*. to appear: http://arXiv.org/quant-ph/0309051.

NAKAJIMA, Y., KAWANO, Y., AND SEKIGAWA, H. 2005. A new algorithm for producing quantum circuits using KAK decompositions. http://arXiv.org/quant-ph/0509196.

NAKAMURA, Y., PASHKIN, Y. A., AND TSAI, J. S. 1999. Coherent control of macroscopic quantum states in a single-cooper-pair box. *Nature 398*, 786–788.

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

NIELSEN, M. A. 2005. Cluster-state quantum computation. http://arxiv.org/abs/quant-ph/0504097.

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

NIELSEN, M. A. AND DAWSON, C. M. 2004. Fault-tolerant quantum computation with cluster states. http://arXiv.org/quant-ph/0405134.

O'BRIEN, J. L., PRYDE, G. J., WHITE, A. G., RALPH, T. C., AND BRANNING, D. 2003. Demonstration of an all-optical quantum controlled-NOT gate. *Nature 426*, 264–267.

ÖMER, B. 2002. Classical concepts in quantum programming. In *Proc. Quantum Structures*.

OSKIN, M., CHONG, F. T., CHUANG, I. L., AND KUBIATOWICZ, J. 2003. Building quantum wires: The long and short of it. In *Computer Architecture News, Proceedings of the 30th Annual International Symposium on Computer Architecture*. ACM, New York.

PASHKIN, Y. A., YAMAMOTO, T., ASTAFIEV, O., NAKAMURA, Y., AVERIN, D. V., AND TSAI, J. S. 2003. Quantum oscillations in two coupled charge qubits. *Nature 421*, 823–826.

PATERSON, K. G., PIPER, F., AND SCHACK, R. 2004. Why quantum cryptography? http://arxiv.org/quant-ph/0406147.

PATTERSON, D. A., GIBSON, G., AND KATZ, R. H. 1988. A case for redundant arrays of inexpensive disks (RAID). In *Proceedings of the 1998 ACM SIGMOD Conference*. ACM, New York, 109–116.

PELLIZZARI, T., GARDINER, S. A., CIRAC, J. I., AND ZOLLER, P. 1995. Decoherence, continuous observation, and quantum computing: A cavity QED model. *Phys. Rev. Lett. 75*, 3788–3791.

PETTA, J. R., JOHNSON, A. C., TAYLOR, J. M., LAIRD, E. A., YACOBY, A., LUKIN, M. D., MARCUS, C. M., HANSON, M. P., AND GOSSARD, A. C. 2005. Coherent manipulation of coupled electron spins in semiconductor quantum dots. *Science 309*, 2180–2184.

PITTMAN, T., JACOBS, B., AND FRANSON, J. 2005. Demonstration of quantum error correction using linear optics. *Phys. Rev. A*, 052332.

PITTMAN, T. B., FITCH, M. J., JACOBS, B. C., AND FRANSON, J. D. 2003. Experimental controlled-NOT logic gate for single photons in the coincidence basis. *Phys. Rev. A 68*, 032316.

PITTMAN, T. B., JACOBS, B. C., AND FRANSON, J. D. 2002. Demonstration of nondeterministic quantum logic operations using linear optical elements. *Phys. Rev. Lett. 88*, 257902.

PRESKILL, J. 1998a. Lectures notes on quantum computation. http://www.theory.caltech.edu/~preskill/ph219/index.html.

PRESKILL, J. 1998b. Reliable quantum computers. *Proc. Roy. Soc. Lond. A 454*, 385–410.

RALPH, T. C., HAYES, A. J. F., AND GILCHRIST, A. 2005. Loss-tolerant optical qubits. *Phys. Rev. Lett. 95*, 100501.

RAUSSENDORF, R., BROWNE, D. E., AND BRIEGEL, H. J. 2003. Measurement-based quantum computation on cluster states. *Phys. Rev. A 68*, 022312.

ROOS, C. F., RIEBE, M., HÄFFNER, H., HÄNSEL, W., BENHELM, J., LANCASTER, G. P., BECHER, C., SCHMIDT-KALER, F., AND BLATT, R. 2004. Control and measurement of three-qubit entangled states. *Science 304*, 1478–1480.

RSA SECURITY INC. 2004. web page. http://www.rsasecurity.com/rsalabs/node.asp?id=2096.

SANAKA, K., JENNEWEIN, T., PAN, J., RESCH, K., AND ZEILINGER, A. 2004. Experimental nonlinear sign shift for linear optics quantum computation. *Phys. Rev. Lett. 92*, 017902.

SANTORI, C., FATTAL, D., VUCKOVIC, J., SOLOMON, G. S., AND YAMAMOTO, Y. 2002. Indistinguishable photons from a single-photon device. *Nature 419*, 594–597.

SCHEEL, S., NEMOTO, K., MUNRO, W. J., AND KNIGHT, P. L. 2003. Measurement-induced nonlinearity in linear optics. *Phys. Rev. A 68*, 032310.

SCHMIDT-KALER, F., HAFFNER, H., RIEBE, M., GULDE, S., LANCASTER, G. P. T., DEUSCHLE, T., BECHER, C., ROOS, C. F., ESCHNER, J., AND BLATT, R. 2003. Realization of the Cirac-Zoller controlled-NOT quantum gate. *Nature 422*, 408.

SCHULMAN, L. J. AND VAZIRANI, U. V. 1999. Molecular scale heat engines and scalable quantum computers. In *Proceedings of the 31st ACM Symposium on the Theory of Computing*. ACM, New York, p. 322.

SHAHRIAR, M. S., HEMMER, P. R., LLOYD, S., BHATIA, P. S., AND CRAIG, A. E. 2002. Solid-state quantum computing using spectral holes. *Phys. Rev. A 66*, 032301.

SHNIRMAN, A., SCHÖN, G., AND HERMON, Z. 1997. Quantum manipulations of small Josephson junctions. *Phys. Rev. Lett. 79*, 2371–2374.

SHOR, P. W. 1994. Algorithms for quantum computation: Discrete logarithms and factoring. In *Proceedings of the 35th Symposium on Foundations of Computer Science*. IEEE Computer Society Press, Los Alamitos, CA, 124–134.

SHOR, P. W. 1996. Fault-tolerant quantum computation. In *Proceedings of the 37th Symposium on Foundations of Computer Science*. IEEE Computer Society Press, Los Alamitos, CA, 56–65.

SHOR, P. W. 1997. Polynomial time algorithms for prime factorization and discrete logarithms on a quantum computer. *SIAM J. Compt. 26*, 5, 1484–1509. http://arXiv.org/quant-ph/9508027.

SKINNER, A. J., DAVENPORT, M. E., AND KANE, B. E. 2003. Hydrogenic spin quantum computing in silicon: A digital approach. *Phys. Rev. Lett. 90*, 087901. http://arXiv.org/quant-ph/0206159.

SØRENSEN, A. AND MØLMER, K. 2000. Entanglement and quantum computation with ions in thermal motion. *Phys. Rev. A 62*.

SPILLER, T. P., MUNRO, W. J., BARRETT, S. D., AND KOK, P. 2005. An introduction to quantum information processing: Applications and realisations. Tech. Rep. HPL-2005-192. Oct.

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

STEANE, A. 1997. The ion trap quantum information processor. *Appl.Phys. B 64*, 623–642.

STEANE, A. ET AL. 2000. Speed of ion trap quantum information processors. *Phys. Rev. A 62*. http://arXiv.org/quant-ph/0003087.

STEANE, A. M. 2002. Quantum computer architecture for fast entropy extraction. *Quantum Inf. Comput. 2*, 4, 297–306. http://arxiv.org/quant-ph/0203047.

STEANE, A. M. 2003. Overhead and noise threshold of fault-tolerant quantum error correction. *Phys. Rev. A 68*, 042322. http://arXiv.org/quant-ph/0207119.

STEANE, A. M. AND IBINSON, B. 2003. Fault-tolerant logical gate networks for CSS codes. http://arXiv.org/quant-ph/0311014.

STEANE, A. M. AND LUCAS, D. M. 2000. Quantum computing with trapped ions, atoms, and light. *Fortschritte der Physik*. http://arXiv.org/quant-ph/0004053.

SVORE, K. M., TERHAL, B. M., AND DIVINCENZO, D. P. 2005. Local fault-tolerant quantum computation. *Phys. Rev. A 72*, 022317.

SZKOPEK, T., BOYKIN, P., FAN, H., ROYCHOWDHURY, V., YABLONOVITCH, E., SIMMS, G., GYURE, M., AND FONG, B. 2004. Threshold error penalty for fault tolerant computation with nearest neighbour communication. http://arxiv.org/abs/quant-ph/0411111.

VAN METER, R. 2005. http://www.tera.ics.keio.ac.jp/person/rdv/quantum/arithmetic.html.

VAN METER, R. AND ITOH, K. M. 2005. Fast quantum modular exponentiation. *Phys. Rev. A 71*, 5 (May), 052320. http://arXiv.org/quant-ph/0408006.

VAN METER, R., ITOH, K. M., AND LADD, T. D. 2005. Architecture-dependent execution time of Shor's algorithm. http://arXiv.org/quant-ph/0507023.

VANDERSYPEN, L. M. AND CHUANG, I. 2004. NMR techniques for quantum computation and control. *Rev. Modern Phys. 76*, 1037.

VANDERSYPEN, L. M. K., STEFFEN, M., BREYTA, G., YANNONI, C. S., SHERWOOD, M. H., AND CHUANG, I. L. 2001. Experimental realization of Shor's quantum factoring algorithm using nuclear magnetic resonance. *Nature 414*, 883–887.

VARNAVA, M., BROWNE, D. E., AND RUDOLPH, T. 2005. Loss tolerant one-way quantum computation—a horticultural approach.

VARTIAINEN, J. J., NISKANEN, A. O., NAKAHARA, M., AND SALOMAA, M. M. 2004. Implementing Shor's algorithm on Josephson charge qubits. *Phys. Rev. A 70*, 012319.

VEDRAL, V., BARENCO, A., AND EKERT, A. 1996. Quantum networks for elementary arithmetic operations. *Phys. Rev. A 54*, 147–153. http://arXiv.org/quant-ph/9511018.

WAKS, E., INOUE, K., OLIVER, W., DIAMANTI, E., AND YAMAMOTO, Y. 2003. High-efficiency photon-number detection for quantum information processing. *IEEE J. Selected Topics Quantum Electronics 9*, 1502–1511.

WALLRAFF, A., SCHUSTER, D. I., BLAIS, A., FRUNZIO, L., HUANG, R.-S., MAJER, J., KUMAR, S., GIRVIN, S. M., AND SCHOELKOPF, R. J. 2004. Strong coupling of a single photon to a superconducting qubit using circuit quantum electrodynamics. *Nature 431*, 162–167.

WALTHER, P., RESCH, K. J., RUDOLPH, T., SCHENCK, E., WEINFURTER, H., VEDRAL, V., ASPELMEYER, M., AND ZEILINGER, A. 2005. Experimental one-way quantum computing. *Nature 434*, 169–176.

WILLIAMS, C. P. AND CLEARWATER, S. H. 1999. *Ultimate Zero and One: Computing at the Quantum Frontier*. Copernicus Books.

WINELAND, D. J. ET AL. 2005. Quantum control, quantum information processing, and quantum-limited metrology with trapped ions. In *Proceedings of the International Conference on Laser Spectroscopy (ICOLS)*. http://arxiv.org/quant-ph/0508025.

YAO, A. 1993. Quantum circuit complexity. In *Proceedings of the 34th Annual Symposium on Foundations of Computer Science*. IEEE Computer Society Press, Los Alamitos, CA, 352–361.

YIMSIRIWATTANA, A. AND LOMONACO JR., S. J. 2004. Distributed quantum computing: A distributed Shor algorithm. http://arxiv.org/quant-ph/0403146.

YORAN, N. AND REZNIK, B. 2003. Deterministic linear optics quantum computation with single photon qubits. *Phys. Rev. Lett. 91*, 037903.

YOU, J. Q., TSAI, J. S., AND NORI, F. 2002. Scalable quantum computing with Josephson charge qubits. *Phys. Rev. Lett. 89*. http://arXiv.org/cond-mat/0306209.

YOU, J. Q., TSAI, J. S., AND NORI, F. 2003. Quantum computing with many superconducting qubits. http://arXiv.org/cond-mat/0306208.

YU, Y., HAN, S., CHU, X., CHU, S.-I., AND WANG, Z. 2002. Coherent temporal oscillations of macroscopic quantum states in a Josephson junction. *Science 296*, 5569, 889–892.

ZALKA, C. 1998. Fast versions of Shor's quantum factoring algorithm. http://arXiv.org/quant-ph/9806084.

Received July 2005; revised December 2005; accepted January 2006