International Journal of Modern Physics C, Vol. 12, No. 9 (2001) 1273-1284

World Scientific Publishing Company

Type-II Quantum Computers

Jeffrey Yepez*
Air Force Research Laboratory
Hanscom Field, Massachusetts 01731
Jeffrey.Yepez@hanscom.af.mil
http://qubit.plh.af.mil/

September 27, 1999
Received 20 December 2000
Revised 4 April 2001

Abstract

This paper discusses a computing architecture that uses both classical parallelism and quantum parallelism. We consider a large parallel array of small quantum computers, connected together by classical communication channels. This kind of computer is called a *type-II quantum computer*, to differentiate it from a globally phase-coherent quantum computer, which is the first type of quantum computer that has received nearly exclusive attention in the literature. Although a hybrid, a type-II quantum computer retains the crucial advantage allowed by quantum mechanical superposition that its computational power grows exponentially in the number of phase-coherent qubits per node. Only short-range and short time phase-coherence is needed, which significantly reduces the level of engineering facility required to achieve its construction. Therefore, the primary factor limiting its computational power is an economic one and not a technological one, since the volume of its computational medium can in principle scale indefinitely.

Keywords: Quantum Computation; hybrid quantum architecture; quantum network; quantum lattice gas.

*This work was supported by the Air Force Office of Scientific Research Directorate of Mathematics and Space Sciences task No. 2304TD and was presented at the Quantum Computing Colloquium, Nuclear Engineering Dept., MIT on September 27, 1999.

1 Introduction

In 1982 towards the end of his life, the famous American physicist Richard Feynman conjectured that it would be possible to simulate many-body quantum mechanical systems using a new type of computer, which he called a quantum computer [1]. After nearly two decades of remarkable theoretical and experimental progress, a few simple quantum computers have finally been constructed and a few simple quantum algorithms tested.

At the crux of quantum computing is the principle of the quantum mechanical superposition of states. Quantum mechanical superposition of states does not naturally play much of a role in our universe at macroscopic scales of the chip size in a conventional desktop digital computer or greater. This is because at these relatively large scales, the quantum state of the astronomical number of particles comprising any desktop-scale computer completely loses all phase-coherence.[1] Nevertheless it is argued that with precise and meticulous engineering, quantum mechanical superposition of persistent qubits, sustained by quantum error correction techniques [4], can be exploited to solve problems that are intractable by any known classical means (for example, factoring large composite numbers [5] or simulating large many-body quantum systems [6, 7]). This is now widely believed because of potency of quantum error codes to correct bit-flip errors and phase errors [8, 9] even though the hypothetical quantum computing device would necessarily be of macroscopic size given the large number of phase-coherent qubits (on the order of a million qubits) required for any practical algorithm. Yet we know that quantum algorithms that require a large number of globally entangled qubits are difficult to experimentally implement and, in the case of NMR quantum computers at least, the use of additional qubits for error correction causes a substantial loss of signal strength [10]. Therefore, globally phase-coherent quantum algorithms are mainly of academic interest only (at least, until a phase-coherent quantum computer embodying millions of entangled qubits is constructed in a way that can exploit many millions more ancillae for quantum error correction).

In contrast, there exists a class of quantum algorithms that also requires many qubits (at least millions), but which only requires that the qubits be entangled over short ranges (the width of a single molecule) and for only short times (less than the natural T_2 spin-spin decoherence time) [11, 12]. Such quantum algorithms are suited for implementation on large parallel arrays of small quantum computers, with each quantum-computing node embodying only a few qubits, and where a classical communication network connects the nodes

[1] To date, a macroscopic scale computer called an nuclear magnetic resonance (NMR) quantum computer, with a computational medium comprising an astronomical number of particles (about as many as contained in a digital chip's wafer), has performed simple quantum computations [2, 3]. However, the liquid-state NMR quantum computer has embodied only a few quantum bits, and is ultimately limited to embodying perhaps no more than a dozen. Effectively, quantum phase-coherence is retained for only a short time (for example, the spin-spin decoherence times is about 700 milliseconds for the alanine molecule) over an extremely short distance (the width of a single alanine molecule). A brief review of the use of NMR spectroscopy for quantum computation is given below in §4.

together. This is a different type of quantum computing architecture than is usually considered in the literature and deserves its own definition and place. The usual type of quantum computing architecture and this second type are both defined in §2.

It is argued here that this second type of quantum computing architecture offers a practical solution for general purpose computing for three reasons. First, as argued in §2, only a limited degree (in both space and time) of quantum phase-coherence is required. Second, as argued in §3, its computational power[2] grows exponentially in the number of phase-coherent qubits per node. And third, as argued in §4, the number of quantum bits per unit volume is naturally commensurate with the density of matter in the liquid or solid state (that is, on the scale of Avagadro's number of qubits per cubic centimeter) over arbitrarily large volumes. Therefore, the achievable computational power of this second architectural type is not unattainable because of overriding technological difficulties but is mainly limited by the cost of the technological implemenation, as is characteristic of massively parallel classical computing architectures.

In short, we envision a quantum computing architecture with many quantum-computing nodes spatially arranged in a regular periodic lattice where small groups of neighboring qubits within each node are homogeneously updated by local quantum gate operations applied simultaneously across the lattice. This type of quantum computer is characterized by both classical parallelism (there being many small quantum computer nodes) and quantum parallelism (there being, in general, quantum superposition and entanglement within each node for some short time span). Therefore, as mentioned above and discussed in more detail in §3, its computational power is the product of the number of nodes in the array times the number of quantum states per node that can be simultaneously superposed. This is a central point of this paper.

2 Definition of Two Types of Quantum Computers

In this section, we define two types of quantum computing architectures: type-I quantum computers have global phase-coherence, and type-II quantum computers have local phase-coherence, limited both in space and time. Fundamental to both architectures, the computing algorithm is represented as a sequence of quantum gate operations [13]. Preliminary quantum gate sequences necessary for modeling a few notoriously difficult physical systems have been "analytically" worked out, but have not yet been tested thoroughly using either numerical simulations or physical experiments. Furthermore, quantum error correction techniques, which are not discussed in this paper, are essential to type-I architectures and not essential, but potentially applicable, to type-II architectures.

[2] I have not formally defined what is meant by the "computational power" of a quantum computer, and instead defer to the usual usage of this term as in the context of classical computation.

2.1 Type-I

In a type-I computer, each qubit may be entangled with any or all other qubits and the system wave function must remain phase-coherent for the duration of the entire quantum gate sequence needed to implement a particular algorithm. The outcome may be determined by measuring one or a few qubits. Type-I quantum computers may be used, for example, for Shor's factoring algorithm [5] or a globally phase-coherent quantum lattice-gas algorithm of the kind developed by Meyer [6], Boghosian and Taylor [7], or Yepez [14].

Developing a quantum algorithm for a type-I quantum computer amounts to unfolding the quantum mechanical evolution operator into an ordered sequence of basic (or fundamental) 2-qubit quantum gate operations. Let $\hat{\Upsilon}$ denote a universal 2-qubit quantum gate. If $\hat{H}$ is the Hamiltonian for a quantum spin system that represents the quantum computer with Q qubits, in general $\hat{H}$ is represented by a dense $2^Q \times 2^Q$ matrix. We must be able to write the quantum mechanical evolution operator that evolves the state of the system wave function for time span τ as the following product

$$e^{-i\hat{H}\tau/\hbar} = \hat{\Upsilon}_N \cdots \hat{\Upsilon}_2 \hat{\Upsilon}_1 \hat{\Upsilon}_0. \tag{1}$$

Here each universal gate operation, $\hat{\Upsilon}_i$, is represented by sparse $2^Q \times 2^Q$ matrix, for $0 \leq i \leq N$. The quantum mechanical evolution equation in Heisenberg form is

$$|\Psi(\vec{x}_1, \ldots, \vec{x}_V; t+\tau)\rangle = \hat{\Upsilon}_N \cdots \hat{\Upsilon}_2 \hat{\Upsilon}_1 \hat{\Upsilon}_0 |\Psi(\vec{x}_1, \ldots, \vec{x}_V; t)\rangle. \tag{2}$$

Equation (2) can be decomposed into a set of equations, one per each gate operation, using N intermediate states $|\Psi_i\rangle$, for $0 \leq i \leq N$

$$\begin{aligned} |\Psi_1\rangle &= \hat{\Upsilon}_0 |\Psi(\vec{x}_1, \ldots, \vec{x}_V; t)\rangle \\ |\Psi_2\rangle &= \hat{\Upsilon}_1 |\Psi_1\rangle \\ |\Psi_3\rangle &= \hat{\Upsilon}_2 |\Psi_2\rangle \\ \vdots \quad &\vdots \quad \vdots \\ |\Psi(\vec{x}_1, \ldots, \vec{x}_V; t+\tau)\rangle &= \hat{\Upsilon}_N |\Psi_N\rangle. \end{aligned} \tag{3}$$

Here, all $N+1$ gate operations are applied within the time period τ. Equation (3) formally expresses a general quantum algorithm for a type-I quantum computer. For any quantum algorithm that would give a solution to a practical problem that could not be obtained efficiently by classical means, very many qubits would have to remain phase-coherent and globally entangled over the entire course of the quantum computation, and this in turn requires the use of extra qubits to correct for bit-flip and phase errors [4, 8]. The use of quantum error correction codes is an intrinsic part of any practical type-I quantum computing algorithm.

Let us consider, as an example, a quantum lattice-gas algorithm [14]. In a quantum lattice gas, the quantum algorithm is quite simple and can be expressed in a form like Equation (3), but with only a single intermediate state[3] $|\Psi'\rangle$

$$|\Psi'\rangle = \hat{C}|\Psi(\vec{x}_1, \ldots, \vec{x}_V; t)\rangle \quad (4)$$

$$|\Psi(\vec{x}_1, \ldots, \vec{x}_V; t+\tau)\rangle = \hat{S}|\Psi'\rangle. \quad (5)$$

Each qubit, $|q\rangle = \alpha|0\rangle + \beta|1\rangle$, in a quantum lattice-gas system is located at a particular node of the lattice with coordinate $\vec{x}$. Each qubit is also associated with a particular lattice vector, $\hat{v}$. The amplitude squared of the "one" state, $|\beta|^2$, by definition equals the probability of finding a particles at location $\vec{x}$ moving with unit speed along direction $\hat{v}$. That is, the *occupancy probability* of a local position-momentum state is defined to be $|\beta|^2$.

The unitary operator $\hat{C}$ changes the occupancy probabilities on each node independently. It causes local collisional scattering of particles at each node in the system independently, and therefore, can be written as a V-fold tensor product over the lattice nodes, $\hat{C} = \bigotimes_{x=1}^{V} \hat{U}$. In general U is a block diagonal unitary matrix, where each block mixes local on-site configuration that have the same additive conserved quantities. For the simplest one dimensional quantum lattice-gas model [15], each $\hat{U}$ is implemented by a single quantum gate operation. For more complex models, $\hat{U}$ would be implemented by a sequence of quantum gates. With N qubits per node, $\hat{U}$ is represented by a $2^N \times 2^N$ size matrix. All the gate operations needed to implement $\hat{U}$ are homogeneously applied on a node-by-node basis causing local superposition and entanglement within each node.

The unitary operator $\hat{S}$ shifts the occupancy probabilities between nodes, but otherwise does not change them. It causes particle movement and is represented by an orthogonal permutation matrix, and would otherwise cause strictly classical data movement if there were not superposed or entangled on-site qubits. In a type-I quantum computer, the application of the streaming operator $\hat{S}$ spreads quantum correlations across the lattice. A detailed treatment of a globally phase-coherent quantum lattice has shown that quantum correlations can significantly alter the macroscopic scale transport properties of the system [14]. The type-I approach has received almost exclusive attention. Its development may take decades, if achievable at all. At the Air Force Research Laboratory[4], we are also focusing our attention on the type-II approach defined in the next section.

[3] Actually, this is an oversimplification of the quantum lattice-gas algorithm because an additional intermediate state is needed to implement the streaming operator $\hat{S}$. Two "checkerboard" lattice partitions must be used to correctly move the particles. So, the streaming operation is done in two steps, $\hat{S} = \hat{S}_1\hat{S}_2$. Furthermore, intermediate states are needed in general to implement the collision operator $\hat{C}$ in two and three dimensional quantum lattice-gas systems.

[4] The quantum computing project, Air Force Office of Scientific Research Task No. 2304TD, is conducted at AFRL/VSBL, Hanscom AFB in Massachusetts (see http://qubit.plh.af.mil/).

2.2 Type-II

A type-II quantum computer represents a network or array of small quantum computers interconnected by classical communication channels. Each qubit may be entangled with only nearby qubits at a particular node of the quantum computer and for only for a short time. The system wave function is always factorized into a tensor product state over the nodes of the lattice

$$|\Psi(\vec{x}_1, \ldots, \vec{x}_V; t)\rangle = |\psi(\vec{x}_1, t)\rangle \otimes |\psi(\vec{x}_2, t)\rangle \otimes \cdots \otimes |\psi(\vec{x}_V, t)\rangle. \tag{6}$$

With N qubits per lattice node, the *on-site ket* $|\psi(\vec{x}, t)\rangle$ resides in a Hilbert space of size $2^N \times 2^N$. The outcome of a computation is determined by measuring the probability of qubit occupancies on all nodes of the array using either an ensemble or coarse-grain averaging technique. That is, the probability of occupancy of a qubit $|q\rangle$ located at coordinate $\vec{x}$ and corresponding to lattice vector $\vec{v}$ is determined by computing the following trace

$$f(\vec{x}, \vec{v}, t) = \mathrm{Tr}\left(|\Psi(\vec{x}_1, \ldots, \vec{x}_V; t)\rangle\langle\Psi(\vec{x}_1, \ldots, \vec{x}_V; t)|\hat{n}_q\right), \tag{7}$$

where $\hat{n}_q$ is the number operator associated with qubit $|q\rangle$.

A type-II quantum computer can be used to run a "factorized" quantum lattice-gas algorithm. An example application of the factorized quantum lattice-gas model is the simulation of of diffusion [15] and a viscous Navier-Stokes fluid [11, 12]. In a factorized quantum lattice gas, the quantum algorithm is also expressed in a form like Equation (3), but with two intermediate states $|\Psi'\rangle$ and $|\Psi''\rangle$

$$\begin{aligned} |\Psi'\rangle &= \hat{C}|\Psi(\vec{x}_1, \ldots, \vec{x}_V; t)\rangle & (8)\\ |\Psi''\rangle &= \hat{\Gamma}|\Psi'\rangle & (9)\\ |\Psi(\vec{x}_1, \ldots, \vec{x}_V; t+\tau)\rangle &= \hat{S}|\Psi''\rangle. & (10) \end{aligned}$$

The additional operation, denoted by $\hat{\Gamma}$, is required to control the system wave function so that it remains a tensor product over the on-site submanifolds after application of the streaming operator $\hat{S}$. The operator $\hat{\Gamma}$ is a *projection operator* and is applied homogeneously across the nodes of the lattice and acts on all the qubits at each node. Therefore, it can be expressed as a tensor product, $\hat{\Gamma} = \bigotimes_{x=1}^{V} \hat{\Gamma}_x$. That is, $\hat{\Gamma}_x$ acts on the on-site ket $|\psi'(\vec{x}, t)\rangle$ independently. It corresponds to a measurement of the occupancy probabilities of each qubit as specified in (7) and causes a collapse of the system wave function and hence is non-unitary

$$|\psi''(\vec{x}, t)\rangle = \hat{\Gamma}_x|\psi'(\vec{x}, t)\rangle = \bigotimes_{a=1}^{n} |q_a\rangle. \tag{11}$$

The measurement procedure specified by the application of $\hat{\Gamma}$ keeps all the occupancy probabilities conserved, keeps each on-site ket (and in turn, the system wave function) unit normal, and in the context of the lattice-gas method obeys the principle of detailed-balance. That is, after the application of $\hat{\Gamma}$, we require that

$$f''(\vec{x}, \vec{v}, t) = f'(\vec{x}, \vec{v}, t), \tag{12}$$

for all $\vec{x}$ and $\vec{v}$. A treatment of a locally phase-coherent quantum lattice has shown there exists a nontrivial projection operator that satisfies the requirement given by Equation (12), and which obeys the principle of detailed-balance [12].

The repeated measurement procedure specified by the application of the projection operator $\hat{\Gamma}$ during each and every time step evolution of the quantum computer wave function is a distinguishing characteristic of a type-II quantum computer. This measurement step alleviates uncontrolled bit-flip and phase errors for all the qubits in the system. Therefore, ancillea for quantum error correction are not needed in a type-II quantum computer. Using ^{13}C labeled molecules such as chloroform (2 qubits), dibromopropionic acid (3 qubits), alanine (3-4 qubits), or transcrotonic acid (5 qubits) in an NMR-based type-II quantum computer gives us only a few qubits per node. Aside from the fact that allocating any of these qubits for the purpose error correction substantially reduces the signal strength [10], practically speaking, at present we do not have any qubits to spare for error correction and all those available are used directly for data encoding and information processing. If in the future it were possible to build a type-II quantum computer with many phase-coherent qubits per node, then quantum error correction could potentially be used for improving the efficiency and durability of the type-II quantum computer.

3 Computational Scaling with a Type-II Quantum Computer

With a type-II array with V nodes, and n qubits per node, the computational power of the machine scales as $V2^{2n}$, which is exponential in n. The factor of V arises from the classical parallelism of having multiple quantum computing nodes. The exponential factor arises from quantum parallelism since a square matrix of size $2^n \times 2^n$ is needed to represent the evolution operator of the n phase-coherent qubits per node. The combined computational power of a type-II quantum computer is just the product of these two factors. Therefore, if a quantum computer array is constructed with enough nodes that its computational power is equivalent to a conventional supercomputer, then a redesign of the machine using $n \to n+1$ qubits per node doubles its performance. For example, if a hypothetical NMR array using chloroform (a n=2 molecule) is replaced with one using dibromoproprionic acid (a n=3 molecule, considering only the carbons), its accessible memory doubles while the total volume of the computational medium remains constant (a fixed volume of liquid in a small test tube). This type of scaling law does not apply to classical computers, where twice as much circuitry (double the volume) is needed to double its accessible memory.

Figure 1 is a log-linear plot of the computational scaling law of a type-II

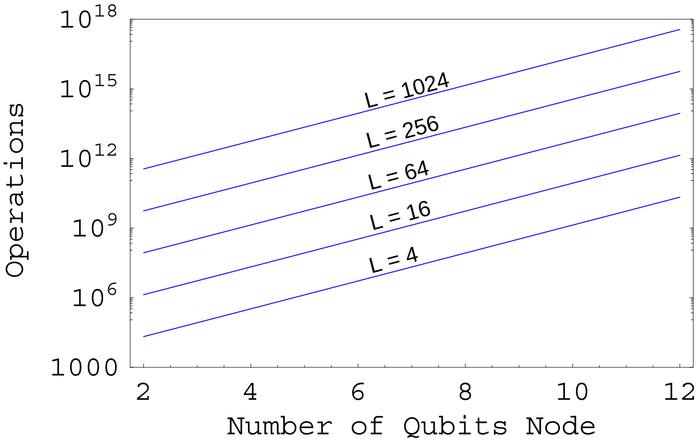

Figure 1: Estimation of the number of equivalent floating-point operations needed to simulate, on a conventional computer, the collision process for a factorized quantum lattice gas that occurs in a single time step. When implemented on a classical computer, the collision process involves the following steps: (1) tensor product operation for projection; (2) matrix multiplication of the on-site ket by a unitary collision matrix of size $2^n \times 2^n$; and, (3) measurement of the qubit occupations.

quantum computer. The number of nodes in the lattice is the cube of the lattice size, $V = L^3$, for the three dimensional lattice pictured below in §4 in Fiqure 2.

4 Quantum Computing using Nuclear Magnetic Resonance Spectroscopy

Among the approaches taken to date, natural quantum mechanical interactions in Carbon-13 have been controlled by nuclear magnetic resonance (NMR) to demonstrate simple computations [2, 3]. The spin-1/2 nucleus of a single Carbon-13 isotope in an external magnetic field (~ 10 Tesla) is a good example of a two-level quantum system. The two quantum states of the nucleus, the spin-up (aligned) and spin-down (anti-aligned), are used to embody a qubit. Several Carbon-13 isotopes, such as the three consecutive carbon nuclei in the organic amino acid called “labeled alanine,” have recently been used as proof-of-concept. Naturally occurring quantum mechanical interactions among the isotopes in the alanine molecule have been coaxed using NMR spectroscopy to do simple computations as a sequence of quantum gates (for example, the Deutsch-Jozsa algorithm and the discrete Fourier transform).

Essentially error-free NMR bulk computation allows for accurate and nondestructive qubit measurement using a large ensemble ($\sim 10^{18}$) of identical quantum computers. Yet, the present day NMR spectroscopic technique is not likely to scale to more than a dozen qubits per quantum computer.[5]

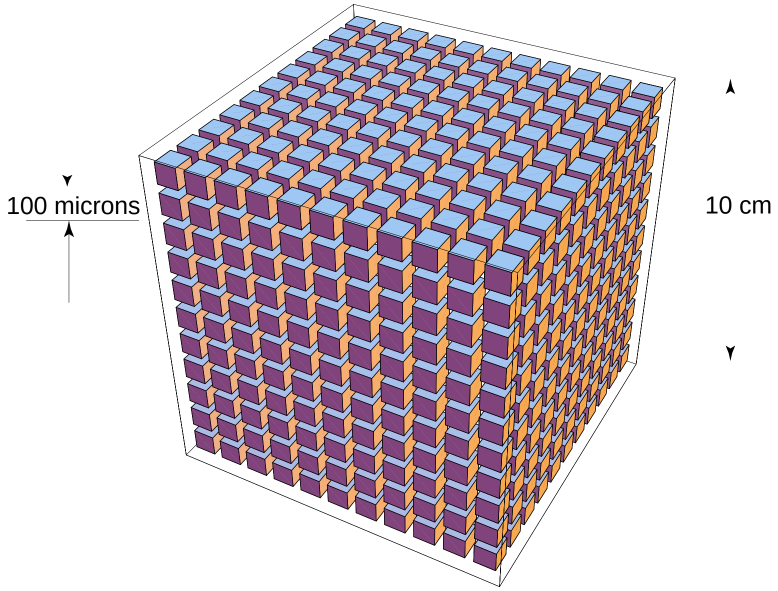

Figure 2: A depiction of a hypothetical cubical array of small quantum computers. Each "node" in the cubical lattice is a parcel of liquid or solid (i a liquid each node is effectively isolated from neighboring nodes because of the relatively slow rate of thermal diffusion of molecules in the sample). Computation, corresponding to the lattice-gas collision operator, is performed in classical parallel fashion independently at each node containing "incoming" qubits. This is accomplished by applying radio frequency electromagnetic pulses homogeneously across the entire sample. The computations occurring within each node is quantum mechanical in nature since the superposition of states is exploited locally. Prior to a loss of signal, on the order of the T_2 spin-spin decoherence time, all "outgoing" qubit occupancies are subsequently measured. Then, the quantum state of the entire system (which has been collapsed into separable tensor product form) is refreshed as the qubit occupancies are rewritten in a permutated order corresponding to the lattice-gas streaming operator. This completes one computational cycle.

In the NMR machine, an ensemble of molecules ($\sim 10^{18}$) embodies a single "mesoscopic n-qubit quantum gate," which is mathematically represented by a unitary matrix in the special unitary group SU(n) (where n is a small number likely to be ≤ 12 by present technology) that acts on n-qubits in a 2^n submanifold of the full Hilbert space. Since a laboratory liquid sample comprises Avagodro's number of molecules ($\sim 6 \times 10^{23}$) there are millions of quantum gates per mole. Hence, the sample can be partitioned into a "Bravais lattice" with a gate at each node of the lattice (see Figure 2).[6]

[5] Other approaches to quantum computing are based on quantum dots, Josephson junctions, SQUIDs, spin electronics, and optical lattices.

[6] In a liter of liquid, a 512^3 array of 6-qubit quantum gates may be accommodated in the

The computational medium is at room temperature (although surrounded by a supercooled superconducting magnet) and the experimental apparatus is nearly "off-the-shelf." The large number of "identical" molecules per quantum gate gives a high signal-to-noise ratio when "reading" the quantum state of a gate (causing phase decoherence on the order of a part per million). Since the atomic nuclei resonate at a unique Bloch frequency fixed by an externally applied magnetic field (like a top processing in a uniform gravity field), a uniform gradient in the field allows for spatial localization within the network. Individual qubits are differentiated by energy level splitting arising from magnetic dipole-dipole coupling within the molecule. This provides a means of "addressing" data within an individual gate.

Since each node (with a ~ 1 millimeter gate-size clocked at ~ 1 kHz) is effectively isolated from every other node (thermal diffusion of molecules in the liquid causes them to move about a micron every millisecond), quantum computation can be done independently and simultaneously across the entire network. This in principle allows for homogeneous parallel computation of the particle collisions occurring at the nodes in the lattice. Furthermore, because of superposition of classical electromagnetic fields, multiple quantum computer nodes within the network can be simultaneously addressed. This allows for parallel computation of particle motion between the node-pairs where particle occupancy information is transfered by auxilary classical means.

Table 1: Three Different Pictures: A Single Point of the System

NMR	QC	Lattice-Gas QC
spin-$\frac{1}{2}$ nuclei (*i.e.* hydrogen or carbon 13-isotope)	qubit $\lvert q\rangle = \alpha\lvert 0\rangle + \beta\lvert 1\rangle$	particle's local state $\hat{n}_\alpha$, $\hat{e}_a$
Wigner-Seitz cell or a molecule (*i.e.* chloroform or alanine)	microscopic quantum computer	site of lattice $\vec{x}$
$\sim 10^{18}$ nuclei (*i.e.* parcel of liquid or solid)	mesoscopic quantum computer many quantum computers	site of superlattice $\vec{X}$ at the mesoscopic scale
RF pulse sequence free induction decay	gate $\hat{U}$	on-site collision operator microscopic $\Omega_a(\vec{x}, t)$
entanged spins	ket $\lvert\psi\rangle$	many outgoing collision possibilities
bulk absorption of RF pulse	ensemble measurement	mesoscopic quantity occupation probability $f_a(\vec{x}, t) = \mathrm{Tr}[\varrho(t)\hat{n}_\alpha]$
gradient fields	Von Neumann measurement Pure State	tensor product on-site ket $\lvert\psi(\vec{x}, t)\rangle \to \bigotimes_{a=1}^{B} \lvert q_a(\vec{x}, t)\rangle$

Quantum computing theory is often pictured with a high level of mathemat-

future (massive classical parallelism).

Table 2: Three Different Pictures: The Entire System

NMR Array	QC	Lattice-Gas QC
magnetic resonance imaging	array of mesoscopic quantum computers	discrete lattice
molecular independence	tensor product wave function $\vert\Psi\rangle = \bigotimes_{x=1}^{V} \vert\psi(x)\rangle$	mesoscopic ensemble of states (only local entanglement)
homogeneously applied RF	tensor product operator $\hat{C} = \bigotimes_{x=1}^{V} \hat{U}$	parallel computation on-site collisions
relaxation state re-preparation	classical communications	particle motion $f_a(\vec{x}, t) \to f_a(\vec{x} + \ell \hat{e}_a, t + \tau)$

ical abstraction whereas quantum-computing experimentation is often pictured using a broad range of physical concepts and practices. The connection between these two pictures has been established in the quantum computing literature [2, 3]. However, to picture lattice-gas quantum computation, one necessarily draws upon many concepts from kinetic particle theory and statistical mechanics as well as concepts from the field of computer science relating to systolic processing and massively parallel single-instruction multiple data processing. Making the connection between the lattice-gas quantum computation picture and the usual theoretical or experimental quantum computation pictures may be difficult to those new to this subject matter. Therefore, to help make the connection between these three different pictures (which are three equally valid ways of representing the same thing), two tables are provided as a learning aid (see Figures 1 and 2). For example, in the first row of Table 1 the concept of a qubit is rendered from three different viewpoints. To an NMR spectroscopist, a qubit is just the quantum number I_z of the state of a spin-$\frac{1}{2}$ nuclei of say a carbon 13 isotope oscillating in an external magnetic field along the $\hat{z}$-axis, as an example. To a computer scientist, a qubit is an abstract ket in a two dimensional Hilbert space, $|q\rangle = \alpha|0\rangle + \beta|1\rangle$. And to a computational physicist, a qubit encodes a particle's occupation of a single local state, which is associated with a particular number operator, $\hat{n}_\alpha$. Each of the rows of tables 1 and 2 illustrates the connection between a particular quantity or concept from the three different viewpoints.

5 Conclusion

Presented in this paper was the simple idea of a quantum computing architecture that is a hybrid between classical massively parallel architecture and an architecture using quantum gates that operate on a set of phase-coherent qubits. A computer built using such hybrid architecture is herein termed a type-II quantum computer. Its salient features are that the number of phase-coherent qubits needed is quite small (less than a dozen for many practical applications such as computational fluid dynamics), its computational power grows exponentially in the number of phase-coherent qubits, and its computational density (number of qubits per unit volume) can be very high while its total computational volume (the volume of matter directly embodying the actual computation) can be arbitrarily large. Quantum lattice-gas algorithms were used as an example to define and illustrate certain details of type-I and type-II quantum computers. The technological application of NMR spectroscopy to type-II quantum computing was also presented as a means of shedding light on this second type of quantum computing architecture, but the type-II quantum computing architecture can be implemented using other technologies as well.

Moore's law, accurately obeyed by the computing industry now for over five decades, points towards information storage and processing at such a small spatial scale that classical electrical circuit theory will be insufficient for continued engineering progress. Quantum mechanical techniques will likely be an essential tool for commercial-grade engineering of computational devices within two decades. Practical type-II quantum computing technology should have application to type-I quantum computers in the longer-term, being an incremental step in that direction.

6 Acknowledgements

I would like to acknowledge David Cory, Richard Nelson, and Bruce Boghosian for helpful discussions.

References

[1] Richard P. Feynman. Simulating physics with computers. *International Journal of Theoretical Physics*, 21(6/7):467–488, 1982. pages 2

[2] Neil Gershenfeld and Isaac L. Chuang. Bulk spin resonance quantum computation. *Science*, 275:350, 1997. January issue. pages 2, 8, 11

[3] David G. Cory, Amr F. Fahmy, and Timothy F. Havel. Ensemble quantum computing by nuclear magnetic resonance spectroscopy. In Tommaso Toffoli, Michael Biafore, and J. Le ao, editors, *PhysComp 96*, pages 87–91. Proceedings on the Fourth Workshop on Physics and Computation, New England Complex Systems Institute, 1996. pages 2, 8, 11

[4] Peter W. Shor. Scheme for reducing decoherence in quantum computer memory. *Physical Review A*, 52:R2493–R2496, 1995. pages 2, 4

[5] Peter W. Shor. Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. In *Proceedings of the 35th Annual Symposium on Foundations of Computer Science*, pages 124–134. Santa Fe, NM, IEEE Computer Society Press, 1994. pages 2, 4

[6] David A. Meyer. Quantum mechanics of lattice gas automata: One-particle plane waves and potentials. *Physical Review E*, 55(5):5261–5269, 1997. pages 2, 4

[7] Bruce M. Boghosian and Washington Taylor IV. A quantum lattice-gas model for the many-particle schrodinger equation in d dimensions. *Physical Review E*, 57:54–66, 1998. pages 2, 4

[8] Emanuel Knill, Raymond Laflamme, and Wojciech H. Zurek. Resilient quantum computation. *Science*, 279:342–345, 1998. pages 2, 4

[9] Emanuel Knill, Raymond Laflamme, and Lorenza Viola. Theory of quantum error correction for general noise. *Los Alamos National Laboratory Archive*, page 6 pages, 1999. quant-ph/9908066. pages 2

[10] David G. Cory, M. Price, W. Mass, E. Knill, Raymond Laflamme, Wojciech H. Zurek, Timothy F. Havel, and S.S. Somaroo. Experimental quantum error correction. *Physical Review Letters*, 81(10):2152–2155, 1998. quant-ph/9802018. pages 2, 7

[11] Jeffrey Yepez. Lattice-gas quantum computation. *International Journal of Modern Physics C*, 9(8):1587–1596, 1998. Proceeding of the 7th International Conference on the Discrete Simulation of Fluids, University of Oxford. pages 2, 6

[12] Jeffrey Yepez. Quantum computation of fluid dynamics. In Collin P. Williams, editor, *Quantum Computing and Quantum Communications*, page 480pp. Lecture Notes in Computer Science, Springer-Verlag, 1999. First NASA International Conference, QCQC'98, Palm Springs, California, USA, February 17-20, 1998, Selected Papers. pages 2, 6, 7

[13] Adriano Barenco, Charles H. Bennett, Richard Cleve, David P. DeVincenzo, Norman H. Margolus, Peter W. Shor, Tycho Sleator, John Smolin, and Harald Weinfurter. Elementary gates for quantum computation. *Physical Review A*, 52(5):3457–3467, 1995. pages 3

[14] Jeffrey Yepez. A quantum lattice-gas model for computational fluid dynamics. *Physical Review E*, (046702):046702–1 to 046702–18, 2001. APS E-Print: aps1999Oct22_002. pages 4, 5

[15] Jeffrey Yepez. Quantum lattice-gas model for the diffusion equation. *International Journal of Modern Physics C*, 12(9):1–19, 2001. Presented at the 9th International Conference on Discrete Simulation of Fluid Dynamics, Santa Fe, NM, August 22, 2000. pages 5, 6