Document type: Proposal, pre-submission.
Agency: NSF/CISE/EIA
Program: 01-95/QuBIC - Quantum and Biological Inspired Computing
Title: *Dawn of a New Field: Quantum Computer Systems Engineering*
PI: Michael Frank (UF/Eng/CISE)
Date: Started 6/22/2001. Due 6/29/2001. LaTeX run of June 26, 2001.

B. Project Summary

The traditional field of *computer systems engineering* (abbreviated CSE in this document) is concerned with the complete, integrated design and analysis of general-purpose (or special-purpose) computing systems. This field of study is highly interdisciplinary, in that can include considerations of technology across all levels from circuit and communication architectures through processor design, chip packaging, multiprocessor system design, programming systems, and software. It can involve a broad range of engineering considerations such as performance, power, cost, and manufacturability.

In our perception, the field of quantum computing is presently lacking a corresponding systems engineering discipline. This is perhaps not surprising, since quantum computing is still in its infancy, and quantum processors of useful capacity have not yet been built. However, the device physics work needed for the implementation of scalable quantum computing has been progressing rapidly [86, 11, 23, 14, 50, 62, 15, 66, 7, 57, 55, 71, 70], and if the historical trends of Moore's law continue, we can reasonably expect that manufacturable bit-devices will reach the scales at which quantum effects are dominant within at most the next 20-50 years, at which point whatever bit-device technology we will be using will necessarily (we argue) have many, if not all, of the characteristics (see [20]) needed for scalable quantum coherent computing. Furthermore, such quantum technology may be available even sooner, if research on macroscopic qubit implementations (*e.g.*, in superconductors [66, 7, 71]) and on error correction algorithms [69] proceeds rapidly.

Once there is a manufacturable, scalable device technology for quantum computing, it will then be necessary to engineer the architecture and software of computers consisting of enormous numbers of these devices. But rather than waiting for the device physicists and manufacturing process engineers to complete their work, we can take a look ahead, and at least get a good head start at working out solutions to the anticipated computer engineering problems. If a mature quantum computer engineering discipline is already well developed by the time quantum bit-devices are practical, then the development of commercializable quantum computer product lines will be greatly accelerated, to the benefit of practical applications throughout society.

To promote this goal, we propose initiating a program of research at UF to build the foundations of a new field of *quantum computer systems engineering*. This will involve work in the following areas:

1. High-level device models that hide the complex details of quantum device physics and manufacturability within a small set of engineering parameters that influence systems design, such as peak frequency, maximum packing density, energy leakage rates, entropy generation rates, decoherence rates, and cost. (See §C.I.a below.)

2. High-performance, distributed simulators for quantum circuit architectures and quantum algorithms, with accompanying visualization tools. (See §C.I.b.)

3. *Asymptotically tight* physically-realistic models of scalable parallel machines composed of quantum devices (previously outlined in [39, 41, 42]). These models would take into account all the relevant fundamental physical constraints, including relativistic limits on communication and thermodynamic limits on entropy removal, and yet offer the highest possible asymptotic performance on all problems. (§C.I.c.)

4. Practical architectures (equal in asymptotic efficiency to the underlying model) that provide a convenient programming model for universal quantum processing elements in a scalable multiprocessing system. (§C.I.d.)

5. Natural high-level programming languages built on our architecture. (See §C.I.e.)

6. Efficient parallel quantum computer algorithms for a variety of natural and important problems of interest in computer science and elsewhere. For example, our group at UF has significant expertise in image processing algorithms [76, 74], which would be investigated as a potential class of candidate applications. (§C.I.f.)

7. Concepts for the design of appropriate multi-tasking, multi-user operating systems for parallel quantum computers. (§C.I.g.)

8. The work at all levels will be done in such a way as to permit a parameterized system cost-performance analysis and thorough design optimization (§C.I.h.)

We have already started working on many of these areas, and our group has significant expertise that can be brought to bear towards the completion of these goals.

C. Project Description

We start with a semi-technical discussion of the long-term goals for our quantum computing research program overall. This research program is ambitious (though realistic) and many-faceted. In section C.II, we will indicate which specific aspects of this larger project we plan to emphasize within the scope of the 3-year award being proposed.

C.I. Technical Discussion

We discuss the technical aspects of our planned development of a quantum computing systems engineering discipline, proceeding from the foundational, low-level components of computing systems, to the highest-level, most abstract components (see fig. 1).

C.I.a. Device Abstraction

The first step in building up a high-level computer systems engineering discipline is to provide a usable foundation for systems analysis by setting up an abstraction barrier to insulate the systems engineer from always having to worry about details of the physical operation of the low-level components of the system.

In quantum computing, the now-standard abstraction for representing the state space of small quantum physical systems is the *qubit*. The qubit represents any two-state quantum system, but systems having more states can be represented satisfactorily using collections of qubits.

Similarly, the standard abstraction of low-level *operations* on qubits is the *quantum logic gate*. The theory of quantum gates, pioneered by David Deutsch in [18] and developed over a period of several years [19, 21, 2], culminating in the discovery [3] that 1-bit quantum gates, together with a single classical reversible 2-bit gate, comprise a complete gate set for quantum computing—that is to say, any quantum logic network can be composed out of such gates, when appropriately interconnected, similarly to how any classical logic network can be composed using 2-input NAND gates.

Qubits and quantum gates give a sufficient foundation for quantum logic. But computer systems engineering (CSE) requires more than just a digital logic foundation. Typical concerns in the engineering of real computing systems include such characteristics as speed, power consumption, and error rates, none of which are addressed by digital logic by itself. Additionally, real qubits are associated with a physical location in space, interact only locally, and require explicit mechanisms for their interconnection.

Therefore, for systems engineering purposes, the description of a primitive quantum computer's components as being qubits and quantum gates must be augmented by some additional parameters that summarize the low-level physical characteristics of these components (in a given device technology) that impact these important systems consideration. These parameters include:

1. **Gate transition time.** Every non-null quantum gate operation requires some non-zero minimum time for its completion. The times for different gates in a system may vary, but it will usually suffice for systems-level analysis to take an average or worst-case time among all gates in the system as being characteristic of all gates. In any event, there is not expected to be a large amount of variation between the minimum transition times of different gates in a system.

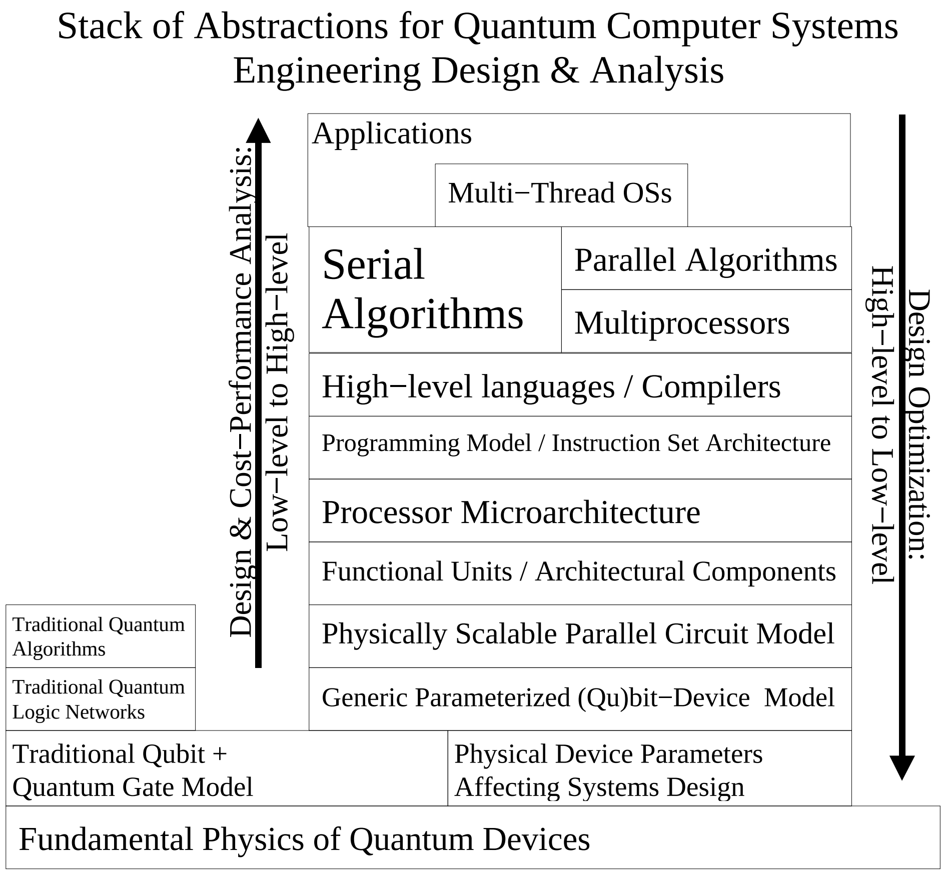

Figure 1: Somewhat analogously to ISO's standard Open Systems Interconnection model for networking systems [68], we can break down the problem of physically-realistic quantum computer systems engineering into a stack consisting of concepts at different layers, from the bottommost foundational layers closest to the physics, up to the higher-level abstractions and artifacts that are the direct subject of systems analysis. The design at each layer depends on the concepts, definitions, and characteristics at the layer below. Traditional qubit and quantum gate models are not suitable for systems design since they lack important physical parameters such as size, cost, power, and error rates. Our extended models allow us to realistically build up engineering designs for higher-level programming structures and algorithms, and then optimize the design all the way back down to the lowest level.

2. **Physical location and velocity.** Qubits must be implemented as real physical objects located in 3-D space. Their relative location is relevant to communication latency concerns. Some qubits in a system (*e.g.*, photons) may be moving relative to the rest of the system ("flying" qubits [20]) in order to enable fast communications; the velocity of such qubits also impacts communications latency and entropy flux density.

3. **Maximum packing density.** The maximum density at which devices (transistors or quantum gates) and state-bearing components (bits or qubits) can be packed together in two and three dimensions is actually an important consideration for computing systems design, because it affects the minimum achievable distances between the components, which (due to the speed-of-light limit) determines the minimum communication latencies in a design. Latency is not so important for uniprocessor machines, but in large-scale parallel systems it is a key concern. This remains true in quantum computers; quantum mechanics does *not* allow faster-than-light communication.

4. **Energy leakage rates.** Any real physical object in a nonequilibrium system dissipates energy (in some form) into its surroundings at *some* nonzero rate, though some objects dissipate much faster than others. Energy leakage is an especially significant concern in nano-scale systems, because at these scales, the potential-energy barriers that usually prevent leakage are much narrower and also lower (due to energy density constraints) than in larger-scale systems. The decreased height of the barriers increases rates of leakage due to thermal fluctuations. Thermal leakage can be suppressed exponentially in the inverse of the absolute temperature, and therefore can be effectively eliminated by sufficient cooling. However, even if thermal leakage is eliminated, quantum tunneling presents another leakage mechanism which not only is not suppressed by cooling, but further is exacerbated by the decreased width (as well as height) of the potential barriers in nanometer scale systems. Still, if states with sufficiently high energy barriers are utilized, it may remain possible to achieve reasonably low leakage rates in even atomic-scale systems—certainly this is the case if state information is encoded in interatomic bond configurations, as is well illustrated by the frequent need in chemistry to use catalysts to lower energy barriers and facilitate chemical reactions that would otherwise occur only extremely slowly.

   Energy leakage is an important systems concern, because (a) it generates entropy which must be removed from the system (entropy cannot be "un-generated"; this is the second law of thermodynamics), and the generation of entropy implies (b) a corresponding loss of free energy which must be compensated for by introducing new free energy from outside (or else the system will run down). The transport of entropy and free energy, in turn, affects overall system design, because there are various technological and fundamental limits to the flux density at which entropy and energy can be transported in a system with bounded temperatures and energy densities, and we do not expect that any future technology will ever allow *unbounded* temperatures and energy densities, since all known material structures break down (melt, vaporize) at high temperatures. Therefore, in designing a computing system, one must explicitly make room for pathways to transport entropy out of the system, and this affects inter-component distances, and thus communications latencies. Already today, the task of providing sufficient cooling and power to VLSI circuits significantly impacts the present-day design of high-performance microprocessors, their packaging, and computer enclosures.

   Therefore, a device model useful for systems engineering will need to include leakage parameters. Bit (or qubit) storage elements in a given technology (at a given temperature) can be well characterized by a constant leakage rate. Active devices can be characterized by a constant leakage rate, and a per-operation energy dissipation (in case there is a transient increase in dissipation rate during device operation). *Adibatic* operations, by definition, have a per-operation dissipation that scales inversely with transition time t, which itself may be a system design variable, implying that there is an important device parameter $k_E = Et$ (the "energy coefficient") that gives the constant energy-time product that characterizes this dissipation term E. (Incidentally, this parameter has the same physical dimensions as angular momentum, or Planck's constant, but we do not yet know if this connection has any physical significance. Planck's constant has been shown *not* to be a lower bound on k_E [60].)

5. **Entropy generation rates.** As we mentioned in the previous item, entropy removal is a significant concern in computer systems engineering. The dissipation of a small amount of energy E causes an amount E/T of entropy to be generated, where T is the temperature of the component's immediate surroundings. This entropy increase can be minimized by maximizing T, but we assume any technology has temperature limits, so entropy generation can only be minimized by raising the cooling system's temperature up to a point—one reason is that heat transfer to the cooling system becomes more difficult. Note also that keeping a high temperature to minimize entropy generation from energy leakage is somewhat at odds with keeping a low temperature to minimize thermally-induced leakage. (However, if thermal leakage is the only source of dissipation, then low temperatures are favored, because thermal leakage goes down exponentially with inverse temperature.) Another important source of energy dissipation is irreversible bit operations [58] which dissipate at least $kT \ln 2$ energy (k being Boltzmann's constant) and therefore generate at least $k \ln 2$ entropy—a constant regardless of operating temperature, so adjusting temperature does not help eliminate this source of entropy generation.

   However, *reversible* operations do avoid this particular source of entropy generation [58, 6], and quantum gate operations are one type of reversible operation. A major focus of our past and future research [41, 42] is in characterizing the system-level design tradeoffs between the entropy generation caused by irreversible operations, and the overhead of the extra operations and space-time storage requirements required for reversible operation. In quantum computers, the requirement for reversibility in order to maintain quantum coherence significantly affects the shape of this tradeoff in quantum algorithms.

6. **Error rates / decoherence rates.** In any computer, the rate at which individual bit-devices experience random errors in their intended logical state is an important system-level consideration, because it impacts the need for digital error-correcting mechanisms and robust architectures and algorithms. Error rates are actually very closely related to rates of energy dissipation and entropy generation. The occurrence of a logical error is a form of energy leakage. In order for a device to mistakenly change states, energy (in some form, such as electrons), must "leak" from the intended region of state space to an unintended region. So, rates of energy leakage between normal states and error states impacts error rates. Also, independently of how much was energy was transferred to cause the error, the unintended change in the machine's logical state represents an increase in *logical* entropy, which is really just a special case of physical entropy. So, error rates impact entropy generation rates. Finally, entropy generation itself is the critical event that causes *quantum decoherence*, which is essentially the dissipation of a quantum probability mass (wave packet) beyond the bounds of a restricted state space in which portions of the wave packet could have been brought together in a controlled way to cause desired interference effects.

   Decoherence rates can be characterized by a rate of decoherence events per unit time for a static data component (qubit), and by a rate of decoherence events per operation for an active device (quantum gate). These rates can be further decomposed into rates for logic errors (bit flips) and phase errors (sign errors in the amplitude) [9]. These separate error rates influence the design of quantum error correction algorithms, an important systems-level concern for applications that require quantum coherence.

7. **Per-device cost** under a given manufacturing process is a key consideration in systems-level design. In particular, the relative cost of manufactured devices and free energy will significantly affect the values of architectural parameters that should be chosen in order to minimize the total (system manufacture plus operation) cost for a given application. For example, some applications will be more energy-limited than others, and so, depending on the device cost and the expected operational lifetime of the system, it may make sense to use more parallel devices, each individually operating more slowly (and so adiabatically dissipating less energy) than in other applications that are less energy limited.

C.I.b. Quantum Logic Emulation and Visualization

After one has a high-level model of logic devices including systems engineering parameters, the next step in computer systems engineering is to determine the number of such devices required to build various intermediate-level structures (laid out in space and/or time), such as functional units needed for common medium-level operations such as integer arithmetic. A realistic count of the device operations required to perform higher-level operations is required in order to derive an aggregate estimate of the amount of system resources (space, time, power, cooling) that are required.

In order to derive accurate estimates of these device counts, our group is already actively carrying out work to investigate these medium-level computing structures. This summer, using existing funding [30, 56], we have hired a bright and highly motivated undergraduate research assistant (Nathan Farrington, who was at the top of the Discrete Mathematics class at UF this Spring) to implement a quantum logic simulator in Java. This work, which has already begun, is planned to eventually include the following features:

1. A spacetime-efficient sparse representation of quantum superpositions, based on eliminating explicit representation of all states having zero amplitude. (Such states dominate whenever classical subroutines are used in quantum algorithms.)

2. A straightforward simulation technique for arbitrary quantum logic networks.

3. Classes implementing all the important varieties of quantum logic gates discussed in the literature , all in terms of primitive gates, as described in [3].

4. Classes implementing higher-level quantum operations, such as reversible arithmetic [80], the quantum Fourier transform [78, 12, 51], and quantum error-correction algorithms [9, 69], in terms of quantum circuit algorithms found in the literature.

5. Classes implementing high-level quantum algorithms, such as the Shor algorithm [cite], as well as a number of classical reversible algorithms of interest. Classical reversible computing is interesting because it can reduce entropy generation [6] and can be implemented even in present-day VLSI technology [88, 87, 48]; it does not require the low decoherence rates needed for true quantum computing.

6. A linear-space (but potentially less time-efficient) simulation technique we have developed that is analogous to the Feynman path integral, or to Cramer's transactional interpretation [17]. This option may be preferable (more efficient overall) for large simulations that would otherwise exceed available computer memories and therefore be I/O-limited.

7. A parallel/distributed simulation technique we have developed that is based on mapping different portions of the quantum state space to different machines on the network. Portions of a quantum wave packet are communicated (using internet packets) to the machines representing the part of state space to which that wave packet is traveling. Essentially, the entire quantum computation is reduced to a data-flow on the network. Synchronization between alternate transmission paths is not necessary due to the linearity of quantum mechanics. The technique can be easily pipelined, leading to high resource utilization. It is also compatible with the path-integral approach in case the memory available on the grid is a limiting factor.

8. Execution of the distributed quantum computer simulator on the OCEAN [33], the market-based distributed computer infrastructure we are developing. A separate proposal for OCEAN funding [26] was submitted by us and other colleagues to the NSF/CISE/ANIR/STI program last week.

9. A refined version of a technique we developed in 1996 for dynamic visualization of the dynamic evolution of quantum computations (see fig. 2). This technique allows the user to watch the movement of amplitude through the machine's configuration space as the computation proceeds. The state spaces of selected multi-qubit machine registers are translated into coordinate axes in a 1, 2, or 3-dimensional

space, with a color-wheel representation of amplitudes. The resulting visualization is rendered onto a graphics display and animated in real time as the simulation proceeds, gate by gate. This visualization tool is both a helpful pedagogical instrument, and a useful tool for gaining scientific insight into the workings of quantum algorithms.

10. Using the simulator, carry out a careful counting and characterization of the number of quantum gate-operations required to implement various higher-level architectures and algorithms, and an accounting of the total energy, etc. This data will be an important input to our higher-level systems design work.

Polynomial-Space Simulation of Quantum Computers A widespread misconception is that classically simulating a quantum computer having n qubits requires, in the worst-case, $\Omega(2^n)$ bits of classical storage, in order to track the amplitudes of all 2^n basis states. Actually, storing all states simultaneously isn't necessary. In a technique inspired by Feynman's path integral formalism, only $O(nt)$ bits of classical state are required to simulate t primitive gate operations in an n-qubit quantum computer. The basic idea is to iterate through possible final states, determining their amplitudes via accumulation of a sum of amplitudes over all paths leading from the initial state. The space savings arises because only one such path need be represented at any given time. However, this more space-efficient algorithm does still require exponential time (bounded by $e^{O(t)}$). Nevertheless, the elimination of the large space requirement means that in an academic research setting, where long computer runs may be easier to obtain than massive amounts of storage, the use of our algorithm may slightly increase the size of quantum computations that we can feasibly simulate.

C.I.c. Physically Realistic Models of Parallel Quantum Computation

Our quantum circuit simulator will allow as to build and test quantum circuits for simple functional units within a quantum computer. But they cannot directly tell us about the behavior of more complex quantum computations, due to the fundamental intractability of simulating quantum systems on classical machines [24]. So, in order to plan larger-scale quantum computations, we need a well-structured theoretical model of scalable, parallel universal quantum computation. We already know how to define such a model, based on insights gained from our earlier work on scalable, parallel universal classical reversible computation [48, 41]. The essence of such a model is simply a 3-D universal reversible cellular automaton, with the added feature that certain automata states cause the transformation of a local cellular neighborhood to proceed via a unitary transformation chosen from a universal set, rather than via only a classical reversible computation. This simple universal parallel quantum computer model can then serve as a foundation for the design and scaling analysis of more easily programmable scalable, parallel, quantum computer architectures.

Furthermore, in order to ensure that our models are truly physically realistic, we plan to incorporate within them the constraints imposed by consideration of the engineering parameters mentioned in §}C.I.a above—rates of leakage, decoherence, entropy generation, *etc.* In addition to necessitating error correction, these parameters have the additional affect of forcing processing elements to be spread out somewhat in comparison to their most compact possible configuration in order to provide sufficient area for the removal of entropy generated within the machine, despite constraints on entropy flux. These issues were explored in more detail in [41] in the context of classical reversible computing—but in quantum computers these thermodynamic issues are essentially unchanged.

The quantum circuit simulator discussed in the previous section can be used to test computations in our 3-D spatial model with large numbers of qubits—but only in cases where the degree of quantum parallelism at any time is kept reasonably small.

Classical Plus Quantum Parallelism. We now illustrate the need for models of quantum computation that incorporate both quantum and classical parallelism by analyzing the minimum asymptotic time complexity of algorithms for a certain class of problems, under models incorporating various combinations of quantum and classical parallelism.

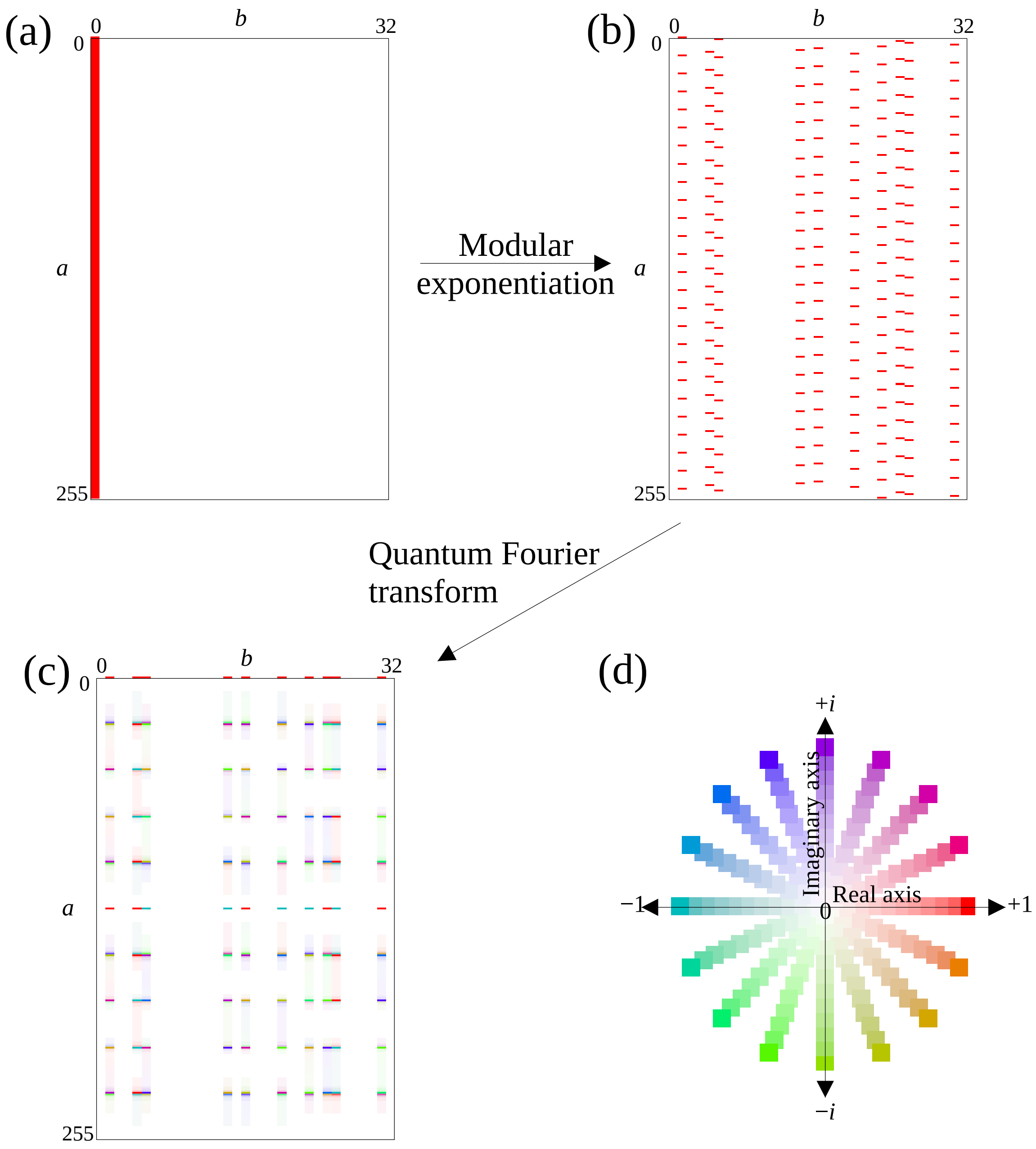

Figure 2: Our visualization technique for quantum computations, applied to the Shor factoring algorithm in two dimensions. Example for factoring $n = 33$. Machine states are projected onto a space defined by the values of two machine registers, a and b. Our new simulator will animate this visualization continuously as individual quantum gate operations are applied. (a) The machine state after generation of a uniform superposition over possible values of a. (b) The state after the modular exponentiation step of Shor's algorithm, $b = x^a \bmod n$; here $x = 5$. (c) The state after performing a quantum Fourier transform over register a. At this point, a can be measured to obtain information about n's factors. (d) A 256-point color-wheel representation for complex-valued state amplitudes.

We choose as our example problem class the *unstructured search problem*, in which a black-box function f is provided, and one is required to find an input value x of the function that produces a given output $y = f(x)$, assuming for simplicity that there is exactly one such input x. We assume that the space of possible inputs x has N elements. To simplify the analysis, we assume that each function evaluation requires $\Theta(1)$ time, and that function evaluations are carried out using a $\Theta(1)$-size mechanism which can be reproduced *en masse* as needed, but which cannot be feasibly transformed to a mechanism for computing the pre-images of given output values y. Note that this black-box model is a reasonable approximation to the actual situation with a typical NP-complete problem, where candidate solutions x of a given size can be checked rapidly using a short algorithm, but there is no known algorithm that can quickly find solutions (and thus the algorithm for computing f might as well be a black box).

An aside: Contrary to a widespread misunderstanding, this particlar black-box model is *not* a very good model of looking up entries in a database, since a database (even a quantum one) having N *arbitrary* (as opposed to correlated) entries will be of size $\Theta(N)$ (not $\Theta(1)$ as in our model), due to the bounded entropy densities of physical systems [4]. In fact, the entire field of quantum statistical mechanics, a cornerstone of modern physics, revolves around the observation that the entropy content of physical systems is bounded by the log-volume of the system's *classical* phase space (measured in Planck units). This log-volume scales only *linearly* with the number of particles. So, for example, n qubits can only store $O(n)$ bits of random classical information; they *cannot*, for example, store the $\Theta(2^n)$ independent bits which would be needed to encode a database of 2^n arbitrary entries. For this and other reasons, we view Grover's characterization of his algorithm as addressing "database search" as being a somewhat misleading picture of its potential real-world applications—another reason being that in an ordinary commercial database, a simple index sufficies to reduce lookup time to $\Theta(1)$ (modulo communications requirements), thereby obviating the whole problem. We therefore choose the more abstract and neutral term "unstructured search" for this problem instead.

Let us now consider various combinations of classical and quantum parallelism. (Some of the below results were first developed by Dr. Frank in 1997 shortly after Grover presented his algorithm.)

1. **No classical or quantum parallelism.** Requires $\Theta(N)$ expected time, since one can do no better than to try possible inputs in an arbitrary sequence until a solution is found.

2. **No classical, but quantum parallelism.** Grover's algorithm takes $\Theta(\sqrt{N})$ time, and this has furthermore been proven [89] to be asymptotically optimal. However, one should keep in mind that the runtime analysis does not apply if there is a constant lower bound on error rates (errors per operation or per unit time) within the machine, since in that case there is an asymptotically divergent overhead factor that is incurred by known fault-tolerant error-correction algorithms [69]. One thing that we plan to do in our proposed work is to accurately analyze the performance of specific quantum algorithms in a realistic model based on known algorithms for quantum error correction.

3. **Classical but no quantum parallelism.** Assume there are M classical serial processors that can simultaneously explore different portions of the search space. Processors are $\Theta(1)$ size, so due to the speed-of-light limit it will take $\Theta\left(M^{1/3}\right)$ time merely to communicate the input to the computation (the desired value) to all M processors from whatever point it originates, when the processors are arranged in a spherical configuration around the origin that minimizes communication time. Each processor takes time $\Theta(N/M)$ to search its portion of the possible inputs, so the total time will be $\Theta\left(M^{1/3} + N/M\right)$. Since $M^{1/3}$ is increasing in M while N/M is decreasing in M, this sum is of least order for fixed N when $M^{1/3} \asymp N/M$ (notation meaning, when the two quantities are of the same asymptotic order). Rearranging, we have $M^{4/3} \asymp N$ or $M \asymp N^{3/4}$. With $M = \Theta\left(N^{3/4}\right)$, the total time is then $\Theta\left(N^{1/4} + N^{1/4}\right) = \Theta\left(N^{1/4}\right)$. Note that this is a factor of $N^{1/4}$ times faster than with quantum parallelism! However, this analysis ignores thermodynamic constraints. We can also show that including a constant temperature bound gives time order $N^{5/16}$, an improvement of $N^{3/16}$ over case 2. Furthermore, including a constant minimum error rate gives time order $N^{1/2}$, which coincidentally is the same as in the quantum case.

4. **Classical and quantum parallelism.** Now each processor can use Grover's algorithm on its part of the search space, and take only $\Theta\left(\sqrt{N/M}\right)$ time for its search. Minimizing $\Theta\left(M^{1/3} + \sqrt{N/M}\right)$ gives $M = N^{3/5}$ and total time order $N^{1/5}$. This is better than with either quantum or classical parallelism by itself. We can also show that adding a temperature constraint gives time $N^{5/24}$ (still better), and adding a constant minimum error rate gives time $N^{1/4}$, which happens to be the same as in the classical case with no bounds on temperatures or error rates. However, this analysis still does not take into account the algorithmic overheads for quantum error correction, only the classical thermodynamic overheads for error-entropy removal. The proposed work would detail and complete the full analysis.

We observe that classical and quantum parallelism taken together seem to offer greater asymptotic performance in realistic models than either type of parallelism taken alone. This observation motivates our study of quantum models that also utilize classical parallelism.

Although the above analyses do not yet take quite all of the relevant physical constraints into account, the results obtained so far lead us to believe that a fuller analysis will continue to show an advantage (though a polynomially smaller one) for both types of parallelism taken together.

C.I.d. Quantum Instruction Set Architectures

Once our model of parallel computation is fleshed out in more detail, we plan to leverage our expertise in designing classical reversible computer architectures [cites] to design a comparable quantum instruction set architecture and a corresponding microarchitectural implementation.

In addition to providing all the ordinary classical reversible instructions that can be found in an architecture such as our PISA architecture [84, 81, 25, 40, 82, 41, 42], the quantum instruction set will also provide special quantum instructions to, for example, perform a bitwise 1-qubit quantum gate operation in parallel across all qubits in a given machine register, or on a selectable subset of the bits. We would also like to incorporate standard DSP (digital signal processing) operations such as FFTs as primitives supported by our instruction set. The quantum Fourier transform used in Shor's algorithm would be another important primitive, as would quantum error correction operations.

We would also like the architecture to include such powerful emerging architectural features as an FPGA-type capability for programmable custom circuits [13], programmable lookup tables for SIMD operations [64, 65], and the capability to program custom data flows among functional units [73].

After specifying the architecture, we would then create a microarchitectural implementation of it by embedding the interconnection structure of all our functional units into the parallel 3-D spatial quantum cellular automaton simulator which we will already have developed.

Of course, our design will be intractable to simulate on a classical computer when executing programs that utilize more than a small amount of quantum parallelism, since it is generally intractable to simulate quantum systems on classical machines. However, in test programs where the amount of quantum parallelism at any time is kept reasonably small, our simulator will be fully capable of simulating all of the functions of the machine, despite the very large number of qubits used in our architecture.

C.I.e. High-Level Quantum Programming Languages

Naturally, one would also like to have high-level quantum programming languages so that one could specify complex quantum algorithms using a textual representation that is more concise and easily readable than assembly language programs for a low-level machine instruction set. We have already accomplished this for classical reversible computation [40, 41, 42]. We would design a quantum version in a natural way using traditional high-level programming language concepts, write a compiler, and run test programs on our simulated quantum architecture. As always, the degree of quantum parallelism that can be feasibly simulated in these tests should be small, but despite this, it should satisfactorily validate the design.

C.I.f. Parallel Quantum Algorithms

Finally, the above-described architecture would be conceived as a single-processor architecture to be used within a scalable 3-D mesh of identical processing nodes (as described in [my stuff]). As discussed above, the most efficient algorithms for many problems would generally be algorithms that used both classical and quantum parallelism. We would develop such algorithms for various applications, such as Grover's unstructured search problem [52], and for image processing applications of interest at UF [75, 76, 74].

C.I.g. Quantum Computer Operating Systems

It is reasonable to ask what kind of operating system a quantum computer should have. One interesting point is that it would be possible to define a type of multithreading that uses quantum rather than classical parallism. Different threads can execute simultaneously on the same hardware resources performing similar (or very different) computations. However, the practicality of this approach may be limited, for the following reasons:

1. The various threads would not be able to communicate with each other at all,
2. none could emit an output in the outside (classical) world without rendering all threads with differing outputs forever inaccessible, and
3. the only way the threads could conspire to produce a result would be for them to all eventually reversibly migrate most of their quantum probability mass to a common set of states, any of which, if measured, would constitute a useful result. The fact that the convergence to the result set has to be done reversibly constitutes a significant constraint on *how* the different threads can arrive at the result set—there must be enough information in the result set to reconstitute the entire history of all the threads. This is what makes quantum algorithms, in general, difficult to design, and the design of a program having multiple quantum threads will generally be just as difficult. Indeed, there is really no difference between the two cases—the operating system does not need to specifically support quantum multithreading, since quantum computing already supports it, by its very nature.

Despite the difficulties of quantum multithreading, the use of *classical* multithreading on a quantum computer is a reasonable and meaningful thing. Multiple threads can perform independent quantum computations. They can be multiplexed in space or in time to run on the same machine. Their results can be separately measured. Communication between threads is possible, but it means that each thread in general will then become entangled with the others, so that afterwards they cannot, for example, independently converge to their result sets—an entire set of intercommunicating threads must be designed to return to their global result set together.

In the design of a quantum OS, not only multithreading but also I/O presents a thorny issue. Input to a reversible or quantum computer from the outside world can be accomplished straightforwardly by performing, from outside a reversible manipulation of previously-empty qubits residing in, for example, a special input register reserved for this purpose. The input register could then be used by the machine in a read-only fashion to affect its internal computations. The input register could be cleared to make room for more data without disturbing the state of the computation.

However, output is a slightly tricky issue. Some cases are easy: A classical thread that is not using quantum parallelism can produce output at any time with no difficulty. Even a thread of a quantum computation can send outputs through a quantum communication network to another quantum computer that is part of a single large distributed quantum computation—this is no different in essence than what happens when information moves in a quantum circuit.

However, if different quantum-parallel threads that are part of a quantum computation wish to produce outputs that can have an effect in the outside (classical) world, then the different threads must produce only the *exact same* output at the *exact same time and place.* If there is any difference between the outputs of the different threads, then this constitutes a measurement by the outside world of the quantum-parallel state (with a probability distribution determined by the relative amplitudes of the different threads), which will forever isolate the threads that produced different outputs from each other. (This is often characterized as the *effective* "collapse" of the probability mass to the subset of the state space that is consistent with the output value that measured by the outside world state that we find ourselves in.)

This difficulty in producing outputs from quantum-parallel threads applies also to the removal of unwanted temporary results from the machine. Again, the only temporary data that can be removed without disrupting the quantum superposition is data that is identical across all the threads and is removed at the same time. Other bits can still be erased, but with the effect of collapsing the state space down to subsets partitioned according to the value of the erased bit. Therefore computations using quantum parallelism will have a need for reversibility that is much stronger than in computations without quantum parallelism, which use it merely to reduce energy dissipation. Thus, the machine will need to support pure reversible operation at all levels throughout its design, from circuits through high-level languages. Fortunately, our extensive earlier work (*cf.* [41]) has already shown that this is straightforward to accomplish.

Depending on the frequency with which it is needed, quantum error correction might also be an appropriate function for an operating system to manage. The OS may periodically interrupt user computations to make a sweep through all machine registers and memory, applying quantum error-correction operations to restore all qubits to the desired region of state space. (This causes some entropy generation, but due to the clever design of quantum error correction techniques, does not collapse the quantum-parallel threads.) The fault-tolerant algorithms reviewed in [69] can be applied recursively to the error-correction mechanism itself (unfortunately with a resulting increase in the asymptotic complexity of quantum computations). However, depending on the decoherence rate of the precise technology, error correction might need to be applied at the hardware level in order to be done frequently enough.

C.I.h. System Scaling Analysis & Design Optimization

In the above sections, we have described device models, circuit simulators, and design work that can feasibly be accomplished for architectures, programming languages, and application algorithms, all before the first significant quantum computer is built. This design work sets the stage for a meaningful, detailed, and physically-realistic analysis of the asymptotic performance characteristics of parallel quantum algorithms, one that takes all of the important real-world engineering considerations into account, such as communications delays, power and cooling requirements, and overheads from error correction and reversible computation.

This analysis, in turn, can be fed back through the entire system design process, to optimize the choices of system design parameters at all levels so as to maximize performance-per-cost for a given (special-purpose or general-purpose) application. Optimizing each level's parameters given the optimizations already taking place at higher levels allows one to propagate the system-level performance considerations all the way down to the design parameters at the lowest level. This will allow us to derive a utility function (reflecting performance-per-cost in high-level applications) on the original parameter space that we started with for our lowest-level devices—individual qubits and quantum gates.

This utility function will then be available as a principled basis for the design optimization of the low-level components, telling the device physicists and the manufacturing process engineers how best to trade off, for example, manufacturing cost against decoherence rates in order to achieve the highest overall utility for overall system applications, given the rational design optimizations that will be made at all of the higher levels.

The utility function on the device parameter space will have the earliest impact on real-world quantum computer design, as it guides the device physicists towards the best device designs. As device technologies improve, examination of our higher-level designs and analysis will then inform manufacturers whether the system performance/cost attainable using a new device is high enough to launch a competitive product line. If so, the manufacturer can proceed to use our optimized systems design and software tools (such as compilers), or more generally our design techniques and methods, to begin the manufacture of a high-quality line of complete quantum computing systems almost immediately after the low-level device technology is available.

That, ultimately, is the impact of this proposed line of research: To allow quantum device physicists and process engineers to quickly focus on the right design goals, and to allow manufacturers to quickly design products. This combination of effects could potentially speed the development of practical, well-designed quantum computers by a number of years, yielding possibly billions of dollars in economic benefit.

We are not so bold as to expect that we will be the only ones to pursue this sort of engineering path: indeed we view it as somewhat obvious, and feel that quantum computer engineering will in subsequent decades become a major field of study. However, we feel that our experience in designing classical reversible computing systems at all levels, and our well-grounded understanding of quantum computing, uniquely qualifies us to initiate this line of research. We intend to devote much effort towards guiding large numbers of people within the device physics, process design, and computer engineering communities to work towards the rational ends proposed in this research. Our role as individual researchers is merely to get the ball rolling early, flesh out the technical details of the above concepts and publish them, and thereby educate the wider community about the above-described, natural outline for the new field of quantum computer systems engineering which we wish to help found.

C.II. Statement of Work to be Undertaken

The previous section describes the long-term goals of our research program. Although it is likely infeasible for our small group to accomplish all of the above-described goals in a 3-year period, we do feel that a significant subset of this work is achievable, and will serve as a useful foundation for follow-on work by ourselves and other researchers. In this section, we describe the work to be undertaken with this award.

C.II.a. Objectives and Significance

In this section, we indicate our plan for the objectives to be pursued under the scope of this award, organizing them chronologically by quarter-year. After each year's objectives we state the significance of that year's work. Items are annotated with a tentative selection of the workers primarily responsible for each objective.

- **Year 0** (before award period)
  - **~3rd quarter 2001** (2 months, Jul.-Aug.):
    - *(Frank, Vieri)* Publish earlier results on classical reversible architectures, languages, and systems analysis with several conference and journal articles.
    - *(Frank, Motter)* Publish earlier results with Motter on numerical stability of reversible simulations of quantum mechanics (which has possible relevance to the new simulator work).
    - *(Frank, Schmalz)* Finish characterization of quantum bit-device abstraction suitable for systems analysis. Determine & tabulate parameter values for various qubit implementations proposed in the literature. Publish results.
    - *(Frank, Farrington)* Implement first version of Java-based quantum circuit simulator & visualization tool (naive algorithm, not distributed). Test using Shor & Grover algorithms and quantum mechanics simulations. Publish.
- **Year 1** (4th qtr. 2001 - 3rd qtr. 2002)
  - **~4th qtr. 2001** (4 months, Sep.-Dec.)
    - *(Frank)* Complete and publish modeling and analysis of classical plus quantum parallelism, outlined in section C.I.c above. Publish.

    - *(Frank, student 1)* Improve quantum circuit simulator: Implement distributed version and path-integral option. Publish.
    - *(Schmalz, student 2)* As a test application, develop reversible and quantum versions of previously-developed image & signal processing primitives such as FFTs, convolutions, etc.; test on circuit simulator; publish.
  - **1st qtr. 2002** (Jan.-Mar.)
    - *(Frank)* Re-teach course on "Physical Limits of Computing" [43] (last taught Spr. 2000), which covers reversible computing, quantum computing, and their impact on systems engineering. Assign student projects to experiment with reversible & quantum machine simulators. Publish course materials via the web.
    - *(Frank, student 1)* Generalize the PISA reversible architecture [84, 81, 25, 38, 82, 41, 42] to support quantum primitive operations. Investigate quantum versions of emerging architectural features such as FPGAs, cellular automata machines, and reprogrammable systolic arrays [64, 65]. Publish.
    - *(Frank, Schmalz, student 2)* Design & implement quantum microarchitecture incorporating both general-purpose and signal-processing capabilities. Test on simulator. Publish.
  - **2nd qtr. 2002** (Apr.-Jun.)
    - *(Frank)* Continue teaching course (through early May).
    - *(Frank, student 1)* Generalize R reversible high-level language compiler [40] to a new language (to be dubbed "Q") that supports quantum primitives. Publish.
    - *(Frank, Schmalz, student 2)* Reimplement test algorithms (Shor, Grover, quantum mechanics simulation, image processing) in new Q programming language. Test under simulator. Publish.
  - **3rd qtr. 2002** (Jul.-Sep.)
    - (*Frank, Schmalz, students)* Develop parallelized versions of test algorithms to run on a mesh of the processors we have designed. Test on simulator.
- **Year 2** (4th qtr. 2002 - 3rd qtr. 2003)
  - **4th qtr. 2002** (Oct.-Dec.)
    - (*Frank, Schmalz, students)* Integrate a quantum error correction scheme into our architecture. Publish.
  - **1st qtr. 2003** (Jan.-Mar.)
    - *(Frank, Schmalz, students)* Develop realistic asymptotic scaling analysis for cost-performance of our sample applications on our parallel architectural model, in terms of device and system-design parameters. Publish.
  - **2nd qtr. 2003** (Apr.-Jun.)
    - (*Frank, Schmalz, students)* Refine above scaling analysis to include numerical values for all constant factors. Publish.
  - **3rd qtr. 2003** (Jul.-Sep.)
    - (*Frank, Schmalz, students)* Back-propagate the system-level cost-performance scaling analysis to determine how to optimize the system design parameters at all levels, given the cost-performance optimizations to be applied at the levels above. Publish.
- **Year 3** (4th qtr. 2003 - 3rd qtr. 2004)

- **4th qtr. 2003** (Oct.-Dec.)
  * *(Frank, Schmalz, students)* Given the above work, derive the utility function over the quantum bit-device parameter space that maximizes system-level cost-performance on the types of test applications we have explored. Publish this result and a description of the methodology used, for the benefit of the quantum device physics community.
- **1st qtr. 2004** (Jan.-Mar.)
  * *(Frank)* Again teach course integrating these research results, like in year 1. This time, student projects can utilize the new parallel machine model (not just the serial one).
- **2nd qtr. 2004** (Apr.-Jun.)
  * *(Frank)* Continue teaching course.
  * *(Frank, Schmalz, students)* Develop concepts for multi-user, multi-tasking operating systems for our quantum computer architecture.
- ~**3rd qtr. 2004** (2 months, Jul.-Aug.)
  * *(Frank, Schmalz, students)* Develop additional test applications for our architecture; characterize how different types of applications impact the utility function over the device design space.

C.II.b. Relation to Longer-Term Goals

Beyond the above-listed objectives, our longer-term goal is of course to establish quantum computer engineering as a major field of study. The above work will lay much of the basic foundation for this field. The most important missing pieces are the following:

1. The device physics, engineering, and manufacturing process design work needed to build the quantum bit-devices that optimize the system-level cost-performance function revealed by our research.

2. Other systems engineering work, relating to thermal and materials engineering of cooling systems, optimization of interconnection systems, packaging and structural integrity, component accessibility for maintenance or replacement, *etc.* Our work only lays the foundation for the design of the bit devices and the computing structures; it doesn't address these other important system design areas, except in the sense that our models ensure that our computation structures will scale realistically with respect to thermodynamic and communications constraints. (These constraints are built into our models and analysis.)

3. Systems design work for additional special-purpose applications.

4. Analysis of processor architecture concepts lying outside our design space. For example, while our architecture might include parameterizable tradeoffs between the amount of space devoted to certain known programming models, such as traditional instruction sets and reconfigurable logic, it will not anticipate alterative architectures that may support programming models that have not been conceived. However, our general system engineering methodology should be adaptable to any possible programming architecture.

C.II.c. Relation to State of Field and Work in Progress Elsewhere

Of course, there are by now many major groups working on the field of quantum computation in general. A sampling of a few of the most well-known ones includes: Oxford's Centre for Quantum Computation Buhrman which includes early pioneer David Deutsch [16]; the DARPA-funded Quantum Information and

Computation collaboration among Caltech, MIT, and USC [72]; the DARPA-funded NMR quantum computing collaboration involving Stanford, Berkeley, MIT, and IBM [79]; Buhrman, Tromp, Vitanyi *et al.*'s group at CWI in Amsterdam [22]; and the Laboratory for Theoretical and Quantum Computing at the University of Montreal [63]. However, essentially all of the existing groups focus either on the fundamental theory of quantum computing & quantum information, or on the experimental physical realization of low-level (few-qubit) quantum devices, rather than on the higher-level systems design & engineering focus that we are proposing. So we seem to be filling a mostly-empty niche.

There are a few exceptions: Kevin Obenland's Ph.D. dissertation under Al Despain at USC [67] made significant strides in the area of quantum circuit simulation, including error modeling and error correction. We intend to leverage some of this work, updating it in the new context of our spatial physical device model, and construct and test higher-level architectural components using methodologies similar to Obenland's.

John Hayes and Igor Markov at the University of Michigan recently started a new DARPA-funded project [54] to develop logic synthesis tools and testing methodologies for quantum computers. We don't intend to focus on these areas, per se, but this work could be a useful complement to our own. We have started a dialogue with this group to discuss collaboration possibilities. In fact, our undergraduate student DoRon Motter who worked with us on quantum mechanics simulations is now a graduate student at Michigan and is considering joining that group.

Benjamin and Johnson at the University of Oxford have a paper [5] in which they introduce a cellular architecture for computing computing that is similar in spirit to the kind of parallel spatial model we proposed to develop in section C.I.c above. However, the Benjamin-Johnson model does not include all of the parameters that need to be included for systems engineering purposes, and it is two-dimensional rather than three-dimensional. We also believe that a much denser and more compact cellular architecture can be developed.

On the topic of applications, there has been almost no work yet on developing image processing or computer vision algorithms for quantum computers. In fact, so far we have found only two papers. In [85], Vlasov begins to explore some interesting quantum image recongition algorithms, and Hattori *et al.* follow up on this work in [53]. But these two papers are just a start, and, we believe, only scratch the surface of the potential applications of quantum computing in image and signal processing and in computer vision that we wish to investigate.

C.III. General Plan of Work

We have already detailed our planned activities in section C.I, and our planned work schedule in C.II.a. Here are some additional points.

C.III.a. Activities to be Undertaken

The work can be roughly divided into architecture work, applications work, and analytical work. We describe this breakdown and our qualifications to do the described work.

Architectural work. Dr. Frank will have primary responsibility for supervising the architectural work, described in more detail in sections C.I and C.II.a above. This will include:

1. Design and implementation of quantum circuit simulators.

2. Development of machine models and architectures to support systems analysis.

3. Design and testing of programming systems: architectural components & languages.

Dr. Frank has significant background relevant to this work. During his graduate work at MIT, Dr. Frank learned computer architecture and systems engineering principles in courses and talks by Dr. William

J. Dally (now at Stanford). He went on to join the MIT Reversible Computing Project [32, 83], where he studied quantum computing [37]; built the world's first reversible instruction set implementation [27] and the first scalable, universal, parallel, fully-reversible processor [48]; helped design the first asymptotically efficient RISC-style reversible instruction set [38, 41]; proposed scalable, physically realistic models of computing [39]; proved new separations and lower bounds in complexity theory for reversible computing [44, 45]; designed & implemented the first C-like reversible programming language [40, 41]; performed a systems-level scaling analysis of reversible machines, which established for the first time that they can outperform conventional machines [47, 46]; and tied all this material together into the first comprehensive Ph.D. thesis on reversible computing [41]. Since his graduation in 1999, Dr. Frank has been working at the University of Florida to initiate collaborations in (classical reversible) adiabatic circuits and quantum computing with faculty in the CISE, ECE, Physics, and Math departments [31, 34], while teaching courses in discrete math [29], computer organization & architecture [28], and the physical limits of computing [43]. Meanwhile, he has also been making informal notes on more optimized reversible circuits and more sophisticated systems analysis, work which would written up and subsumed within the effort proposed here.

Applications work. Dr. Schmalz will have primary responsibility for supervising the applications work, particularly aspects relating to our new image processing applications, with support from Dr. Frank in the area of more traditional quantum computing applications. Our suite of test applications will include:

1. Shor's factoring algorithm [78].
2. Grover's algorithm for unstructured search [52].
3. Algorithms for simulation of quantum physical systems [24, 8, 1].
4. Algorithms derived from UF's work on image algebra and image processing [75, 76, 74].

Dr. Mark Schmalz (co-PI) specializes in image and data compression, image and signal processing (ISP), and computer vision (CV). In this proposed project, Dr. Schmalz would apply his expertise in ISP/CV and performance analysis to (a) translate basic ISP/CV algorithms to the proposed reversible/quantum architecture, and (b) measure and analyze performance of a wide variety of image-related tools and algorithms within this framework. Such algorithms are of key importance to rapidly expanding areas of technology including Internet video, telemedicine, and distance learning. The proposed effort leverages a decade of epxerience at UF in porting image algebra to a wide variety of sequential and parallel workstations, massively parallel processors, digital signal processors, and adaptive or reconfigurable computing systems [74, 76]. Performance of these image algorithms would be analyzed in terms of computational error, as well as space and time complexity [75, 76], and fed back into the overall system optimization analysis.

Analytical work. Drs. Frank and Schmalz will work together on analytical aspects of this project. This includes:

1. Deriving analytical expressions for system-level cost/performance of our architectures, in terms of low-level device parameters and architectural parameters.
2. Deriving analytical expressions for the optimal values of system-level architectural parameters in terms of lower-level parameters.
3. Back-propagating the optimization of the system-level cost/performance to determine a utility function over the parameter space at all lower levels, including the device level.
4. Incorporating considerations of data compression, error analysis, and error correction at all levels in the architecture and applications. This includes issues from quantum error correction at the low-level, to image error correction and lossy compression at the applications level.

C.III.b. Experimental Methods and Procedures

Experiments in our project center around simulation of our architectures using our quantum circuit simulator. We will develop distributed versions of our simulator for faster simulation of large circuits. Such distributed simulations may be written to use the OCEAN distribution computation market project which is simultaneously being proposed [26]. The basic experimental methodology will be to code up a quantum algorithm, and run a given simulation of the algorithm using a range of input sizes and settings of architectural parameter values. The resulting data points will be graphed and compared with theoretical graphs predicted by our analytical models of our architectures. Checking that the two are in correspondence will allow us to validate both the designs and the models. The simulator itself will be validated by checking that test programs run as predicted and produce correct outputs.

C.III.c. Plans for Preservation, Documentation, and Sharing of Research Products

All software, design files, and publications produced by the project will be made permanently publicly available through the UF CISE department's web server. In addition, all major results will be archived via journal publications.

C.IV. Broader Impacts of Proposed Activity

C.IV.a. Integration of Research and Education

This research will be highly integrated with our educational work, due to:

1. PI Frank's bi-yearly course on *Physical Limits of Computation* [43], which was very popular among both CISE and ECE students (last year it filled its 50-student enrollment limit almost instantly) and among both graduate and undergraduate students. The course received extremely high evaluations from both groups. The course covers both reversible and quantum computing, and includes student group projects that experiment with these concepts. (For example, last year one group simulated Grover's algorithm.) These projects will be able to make use of the simulator tools, architectural components, and language compilers that we will be developing.

2. Both Frank and Schmalz are heavily involved in the CISE department's Senior Project course [77]—the ABET-approved capstone senior design course required of all engineering students in our department. Dr. Schmalz organizes the course, and offers project topics, and Dr. Frank supervises several dozen individual projects annually. Many of Dr. Frank's students' senior projects already have been in the area of reversible and/or quantum computing [34]. The results of this research would directly feed in to improve the resources available to these students, and the quality of the resulting projects. Additionally, many senior projects would in turn feed back to contribute to this research.

3. Dr. Frank is in the process of writing a textbook *Reversible Computing: From Theory to Engineering* to be published by MIT Press [49], based on Dr. Frank's Ph.D. thesis and the results of his recent systems engineering research. The book is already planned to include major portions on reversible computer systems engineering, and can easily be modified to include the quantum computing aspects of this work.

4. Dr. Frank and Dr. Schmalz both occasionally teach the undergraduate computer organization course at UF [28], and Dr. Frank will also teach the graduate computer architecture course starting this Fall. Dr. Frank's reversible computing work is already feeding into these courses, in the form of a reversible MIPS simulator written by one of his Masters' students, which will be used in the undegraduate course. Dr. Frank also plans to incorporate a significant treatment of systems engineering principles, and a few lectures on reversible and quantum architectures, into his graduate architecture course this Fall. As

the quantum architectures proposed in this project are developed, they will help improve the maturity of the quantum computing coverage in the graduate architecture course.

C.IV.b. Participation by Underrepresented Groups

Dr. Frank has always encouraged participation by women, minorities, and other underrepresented groups on his research projects. He also serves in UF's University Minority Mentoring Program, and he works to help ensure diversity in the student body through his role on the CISE department's graduate committee. This committee strives to ensure, for example, that women and students from underrepresented nations (unfortunately, among graduate students, the U.S. is underrepresented!) are especially encouraged to attend the UF CISE graduate program. Dr. Frank has supervised a wide diversity of students on his own projects, and will continue to do so in the future. For example, last year he supervised a senior highest-honors project by an African-American student, DoRon Motter, who made a major theoretical contribution (not yet published) to the study of reversible simulations of quantum mechanics.

C.IV.c. Enhancements to Infrastructure for Research and Education

The quantum computer simulators and architectural components we develop will be made freely, publicly available as they are developed through our group's web site [35], and publicized at conferences, so that other researchers, instructors, and students at other universities who are interested in quantum computing can benefit from these tools for use in their own projects.

C.IV.d. Dissemination of Results

As outlined in the objectives list above, we are planning to regularly produce papers on all stages of this work. We will balance conference publications (to educate the community) with journal articles (to archive the resutls). We intend to strongly encourage all graduate students involved in this project to contribute to these publication efforts.

C.IV.e. Potential Benefits to Society at Large

Since scalable quantum computing is still probably at least one or two decades away, the benefits to society will be long-term. However, it is hoped that by providing guidance to device physicists and quantum computer engineers, this research will significantly speed the advance of quantum computing, perhaps accelerating its practical implementation by several years. By speeding the arrival of the new applications (particularly quantum physical simulations for drug design, materials engineering, and molecular nanotechnology) that will be enabled by practical quantum computing, this research is projected to have enormous societal benefits, for example by potentially saving the lives of millions who might die of some disease if not for the quick arrival of some future drug to be designed by quantum molecular modeling on a efficient quantum computer, one that will be optimized using the principles introduced by our work, on *quantum computer systems engineering*. We estimate that the potential value to society of speeding the eventual development of quantum computers by several years is certainly in the billions of dollars.

C.V. Results from Prior NSF Support

Neither PI has had NSF support in the last 5 years. However, Dr. Frank held an NSF graduate fellowship from 1993-1995, the first 2 years of which he used for his Masters' work on decision-theoretic AI [], and the last year of which he used while working briefly on DNA computing at MIT. It happened that the technique for universal computing with DNA which Frank developed during that period was forced to be reversible due to fundamental thermochemical reasons, and so, during the course of this work, Frank became familiar with the field of reversible computing, which led him to pursue the more practical (non-biological) line of reversible computer engineering research that he conducted during his Ph.D. and is continuing now (discussed in §C.II.a above). Thus, this graduate NSF support led, if indirectly, to a Ph.D. thesis and to many interesting published results [36, 46, 47, 48, 41] and available web materials, and to the concept for the quantum computing work which we are now proposing.

D. References Cited

References

[1] Daniel S. Abrams and Seth Lloyd. Simulation of many-body fermi systems on a universal quantum computer. *Phys.Rev.Lett.*, 79:2586–2589, 1997. `http://arxiv.org/abs/quant-ph/9703054`.

[2] Adriano Barenco. A universal two-bit gate for quantum computation. *Proceedings of the Royal Society of London Ser. A*, 449:679–683, 1995. Preprint at Los Alamos Physics Preprint Archive, `http://xxx.-lanl.gov/abs/quant-ph/9505016`.

[3] Adriano Barenco, Charles H. Bennett, Richard Cleve, David P. DiVincenzo, Norman H. Margolus, Peter W. Shor, Tycho Sleator, John A. Smolin, and Harald Weinfurter. Elementary gates for quantum computation. *Physical Review A*, 52(5):3457–3467, 1995. Preprint at Los Alamos Physics Preprint Archive, `http://xxx.lanl.gov/abs/quant-ph/9503016` and at `http://vesta.physics.ucla.edu/cgi-bin/uncompress_ps_cgi?torgats1.ps`.

[4] Jacob D. Bekenstein. Universal upper bound on entropy-to-energy ratio for bounded systems. *Phys. Rev. D*, 23(2):287–298, January 1981.

[5] Simon C. Benjamin and N. F. Johnson. Cellular structures for computation in the quantum regime. Preprint, 1998. `http://arXiv.org/abs/cond-mat/9808243`.

[6] C. H. Bennett. Logical reversibility of computation. *IBM J. Research and Development*, 17(6):525–532, 1973.

[7] G. Blatter, V. B. Geshkenbein, and L. Ioffe. Engineering superconducting phase qubits. preprint, December 1999. `http://arXiv.org/abs/cond-mat/9912163`.

[8] Bruce M. Boghosian and Washington Taylor IV. Simulating quantum mechanics on a quantum computer. In Tommaso Toffoli, Michael Biafore, and João Leão, editors, *PhysComp96*, pages 32–35. New England Complex Systems Institute, 1996. `http://xxx.lanl.gov/abs/quant-ph/9701019`.

[9] A. R. Calderbank and Peter W. Shor. Good quantum error-correcting codes exist. Los Alamos Physics Preprint Archive, `http://xxx.lanl.gov/abs/quant-ph/9512032`, December 1995.

[10] C. S. Calude, J. Casti, and M. J. Dinneen, editors. *Unconventional Models of Computation*. Springer, 1998. (Proc. of the First International Conference on Unconventional Models of Computation (UMC'98), held at the University of Auckland, January 5–9, 1998).

[11] J. I. Cirac and P. Zoller. Quantum computations with cold trapped ions. *Physical Review Letters*, 74:4091–4094, 1995.

[12] D. Coppersmith. An approximate Fourier transform useful in quantum factoring. Research Report RC 19642, IBM, 1994.

[13] Xilinx Corporation. Xilinx: Programmable logic devices, FPGA & CPLD. corporate web site. `http://www.xilinx.com`.

[14] David G. Cory, Amr F. Fahmy, and Timothy F. Havel. Ensemble quantum computing by nuclear magnetic resonance spectroscopy. *Proceedings of the National Academy of Sciences of the United States of America*, 94(5):1634–1639, 4 March 1997. See `ftp://deas.ftp.harvard.edu/techreports/tr-10-96.ps.gz`.

[15] David G. Cory, M. D. Price, and T. F. Havel. Nuclear magnetic resonance spectroscopy: An experimentally accessible paradigm for quantum computing. *Physica D*, 120(1–2):82–101, 1 September 1998.

[16] Centre for quantum computation. Web site. At Oxford University. `http://www.qubit.org`.

[17] John G. Cramer. The transactional interpretation of quantum mechanics. *Reviews of Modern Physics*, 58:647–688, July 1986.

[18] David Deutsch. Quantum computational networks. *Proceedings of the Royal Society of London Ser. A*, A425:73–90, 1989.

[19] David Deutsch, Adriano Barenco, and Artur K. Ekert. Universality in quantum computation. *Proceedings of the Royal Society of London Ser. A*, 449:669–677, 1995. Shows that almost all 2-bit gates are universal. Preprint available at Los Alamos Physics Preprint Archive, `http://xxx.lanl.gov/abs/quant-ph/9508012`.

[20] David DiVincenzo. The physical implementation of quantum computation. preprint for *Fortschritte der Physik* special issue, Experimental Proposals for Quantum Computation, February 2000. `http://arXiv.org/abs/quant-ph/0002077`.

[21] David P. DiVincenzo. Two-bit gates are universal for quantum computation. *Physical Review A*, 51(2):1015–1022, February 1995. Also at Los Alamos Physics Preprint Archive, `http://xxx.lanl.gov/abs/cond-mat/9407022`.

[22] H. M. Buhrman *et al.* Quantum computing. Web site. `http://dbs.cwi.nl:8080/cwwwi/owa/cwwwi.print_projects?ID=34`.

[23] T. Pellizzari *et al.* Decoherence, continuous observation and quantum computing: a cavity QED model. Preprint, June 1995. University of Innsbruck.

[24] Richard Feynman. Simulating physics with computers. *International Journal of Theoretical Physics*, 21(6&7):467–488, 1982.

[25] Michael Frank. Time-symmetric control-flow instructions for less garbage in reversible programs. MIT Reversible Computing Project Internal Memo #M1, February 1996. `http://www.ai.mit.edu/~mpf/rc/memos/M01_symmarch.html`.

[26] Michael Frank, J. Hammer, A. Helal, J. Peters, and M. Schmalz. OCEAN: A highly liquid market for distributed computation. proposal to NSF STI program (01-90), June 2001. `http://www.cise.ufl.edu/research/ocean/index.html#sti`.

[27] Michael Frank and Scott Rixner. Tick: A simple reversible processor (6.371 project report). Online term paper, May 1996. `http://www.ai.mit.edu/~mpf/rc/tick-report.ps`.

[28] Michael P. Frank. Cda 3101, fall 2000: Introduction to computer organization. course web site. `http://www.cise.ufl.edu/class/cda3101fa00`.

[29] Michael P. Frank. Cot 3100 spr.'01: Applications of discrete structures. course web site. `http://www.cise.ufl.edu/class/cot3100sp01`.

[30] Michael P. Frank. Design & implement a prototype for a dsl model that supports multiple simultaneous voice phones over a single dsl line. online IPPD [56] project description. http://www.cise.ufl.edu/~mpf/projects.html#VoDSL.

[31] Michael P. Frank. Mike frank's funding proposals. web page. http://www.cise.ufl.edu/~mpf/proposals.html.

[32] Michael P. Frank. "mike's reversible computing page". web page. http://www.ai.mit.edu/~mpf/rc/, now redirected to http://www.cise.ufl.edu/~mpf/rc/.

[33] Michael P. Frank. OCEAN project home page. web page. http://www.cise.ufl.edu/research/ocean.

[34] Michael P. Frank. Reversible computing group members. web page. http://www.cise.ufl.edu/research/revcomp/people.html.

[35] Michael P. Frank. The reversible computing (revcomp) group. research group web site. http://www.-cise.ufl.edu/research/revcomp.

[36] Michael P. Frank. Advances in decision-theoretic AI: Limited rationality and abstract search. Master's thesis, Massachusetts Institute of Technology, Cambridge, Massachusetts, May 1994. http://www.ai.-mit.edu/~mpf/papers/Frank-94/Frank-94.html.

[37] Michael P. Frank. Quantum computation primitives. Area exam paper, http://www.ai.mit.edu/~mpf/qcpaper.html, February 1996.

[38] Michael P. Frank. Modifications to PISA architecture to support guaranteed reveribility and other features. MIT Reversible Computing Project Internal Memo #M7, February 1997. http://www.ai.-mit.edu/~mpf/rc/memos/M07/M07_revarch.html.

[39] Michael P. Frank. Physically-motivated models of computation for complexity theory. MIT Reversible Computing Project Internal Memo #M5, March 1997. http://www.ai.mit.edu/~mpf/rc/memos/M05_physical.html.

[40] Michael P. Frank. The r programming language and compiler. MIT Reversible Computing Project Internal Memo #M8, July 1997. http://www.ai.mit.edu/~mpf/rc/memos/M08/M08_rdoc.html.

[41] Michael P. Frank. *Reversibility for Efficient Computing.* PhD thesis, Massachusetts Institute of Technology, Cambridge, MA, May 1999. http://www.ai.mit.edu/~mpf/rc/thesis/phdthesis.html.

[42] Michael P. Frank. Reversibility for efficient computing. Course manuscript for the *Physical Limits of Computing* course at the University of Florida, Spring 2000[43], December 1999. http://www.cise.-ufl.edu/~mpf/manuscript.

[43] Michael P. Frank. Physical limits of computing. Course web site, April 2000. http://www.cise.ufl.-edu/~mpf/physlim.

[44] Michael P. Frank and M. Josephine Ammer. Separations of reversible and irreversible space-time complexity classes. MIT Reversible Computing Project Memo #M6, http://www.ai.mit.edu/~mpf/rc/memos/M06_oracle.html, February 1997.

[45] Michael P. Frank and M. Josephine Ammer. Relativized separation of reversible and irreversible space-time complexity classes. Submitted to *Information and Computation.*, May 2001. Revised and expanded version of [44].

[46] Michael P. Frank and Thomas F. Knight, Jr. Ultimate theoretical models of nanocomputers. *Nanotechnology*, 9(3):162–176, 1998. Presented at the Fifth Foresight Conference on Molecular Nanotechnology, Palo Alto, CA, November 1997. http://www.ai.mit.edu/~mpf/Nano97/paper.html.

[47] Michael P. Frank, Thomas F. Knight, Jr., and Norman H. Margolus. Reversibility in optimally scalable computer architectures. In Calude et al. [10], pages 165–182. http://www.ai.mit.edu/~mpf/rc/scaling_paper/scaling.html.

[48] Michael P. Frank, Carlin Vieri, M. Josephine Ammer, Nicole Love, Norman H. Margolus, and Thomas F. Knight, Jr. A scalable reversible computer in silicon. In Calude et al. [10], pages 183–200. http://www.ai.mit.edu/~mpf/rc/flattop/ft.html.

[49] Michael P. Frank and Carlin J. Vieri. *Reversible Computing: From Theory to Engineering.* MIT Press, 2002. Manuscript [42] currently under development.

[50] Neil A. Gershenfeld and Isaac L. Chuang. Bulk spin resonance quantum computation. *Science*, 275(350), jan 1997.

[51] Robert B. Griffiths and Chi-Sheng Niu. Semiclassical Fourier transform for quantum computation. Los Alamos Physics Preprint Archive, http://xxx.lanl.gov/abs/quant-ph/9511007, November 1995.

[52] Lov. K. Grover. A fast quantum mechanical algorithm for database search. In *Proc. , 28th Annual ACM Symposium on the Theory of Computing (STOC)*, pages 212–219, May 1996. http://xxx.lanl.-gov/abs/quant-ph/9605043.

[53] Tetsuo Hattori, Katsunori Takeda, and Toshinori Yamasaki. Recognition of image data in quantum computer. In Eric Donkor and Andrew R. Pirich, editors, *Quantum Computing*, volume 4047 of *Proceedings of SPIE*, pages 27–34, 2000.

[54] John Hayes and Igor Markov. The Quantum Circuits Group at the Univ. of Michigan. Web page, 2001. http://vlsicad.eecs.umich.edu/Quantum/.

[55] Xuedong Hu and S. Das Sarma. Theoretical issues in spin-based quantum dot quantum computation. preprint, submitted to the IQC01 conference proceedings, February 2001. http://arXiv.org/abs/cond-mat/0102019.

[56] IPPD: Integrated product & process design. educational program web site. http://www.ise.ufl.edu/ippd.

[57] B. E. Kane. Silicon-based quantum computation. preprint, submitted to Fortschritte der Physik Special Issue on Experimental Proposals for Quantum Computation, March 2000. http://arXiv.org/abs/quant-ph/0003031.

[58] Rolf Landauer. Irreversibility and heat generation in the computing process. *IBM J. Research and Development*, 5:183–191, 1961. Reprinted in [59], ch. 4, pp. 188–196.

[59] Harvey S. Leff and Andrew F. Rex, editors. *Maxwell's Demon: Entropy, Information, Computing.* Princeton series in physics. Princeton University Press, Princeton, NJ, 1990. May be ordered through http://www.ioppublishing.com/Books/Catalogue/020/__26/0750300566.

[60] K. K. Likharev. Classical and quantum limitations on energy consumption in computation. *International Journal of Theoretical Physics*, 21(3/4):311–326, 1982.

[61] Hoi-Kwong Lo, Tim Spiller, and Sandu Popescu, editors. *Introduction to Quantum Computation and Information.* World Scientific Pub. Co., 1998. ISBN 981023399X.

[62] Daniel Loss and David P. DiVincenzo. Quantum computation with quantum dots. *Phys. Rev. A*, 57(120), 1998. Preprint available at `http://arXiv.org/abs/cond-mat/9701055`.

[63] Laboratory for theoretical and quantum computing. web site. `http://www.iro.umontreal.ca/labs/theorique/index.html.en`.

[64] Norman Margolus. An FPGA architecture for DRAM-based systolic computations. In Arnold *et al.*, editor, *Proceedings of the IEEE Workshop on FPGAs for Custom Computing Machines*, pages 2–11. IEEE Computer Society Press, 1997.

[65] Norman Margolus. An embedded DRAM architecture for large-scale spatial-lattice computations. In *International Symposium on Computer Architecture*, Vancouver, jun 2000.

[66] J. E. Mooij, T. P. Orlando, L. Levitov, Lin Tian, Caspar H. van der Wal, and Seth Lloyd. Josephson persistent-current qubit. *Science*, 285:1036–1039, 13 August 1999.

[67] Kevin Mark Obenland. *Using Simulation to Assess the Feasibility of Quantum Computing*. PhD thesis, University of Southern California, aug 1998. `http://www.isi.edu/acal/quantum/quantum_intro.-html`.

[68] International Standards Organization. Open systems interconnection. ISO International Classification for Standards field number 35.100. `http://www.iso.ch/iso/en/CatalogueListPage.-CatalogueList?ICS1=35&ICS2=100&ICS3=`.

[69] John Preskill. Fault-tolerant quantum computation. In Lo et al. [61]. `arxiv.org/abs/quant-ph/9712048`.

[70] Vladimir Privman and Dima Mozyrsky. Semiconductor quantum computer design with 100 nm separation of nuclear-spin qubits. preprint, March 2001. `http://arXiv.org/abs/quant-ph/0103116`.

[71] Vladimir Privman, Dima Mozyrsky, and Israel D. Vagner. Quantum computing with spin qubits in semiconductor structures. preprint, February 2001. `http://arXiv.org/abs/cond-mat/0102308`.

[72] Quantum information and computation. web site. `http://theory.caltech.edu/~quic`.

[73] raw architecture workstation project. project web site. `http://www.cag.lcs.mit.edu/raw`.

[74] G. X. Ritter and J. N. Wilson. *Handbook of Computer Vision Algorithms in Image Algebra*. CRC Press, second edition, 2001.

[75] M. S. Schmalz. Automated analysis and prediction of accuracy and performance in ATR algorithms. *Proceedings SPIE*, 3079:249–272, 1997.

[76] M. S. Schmalz, G. X. Ritter, and F. M. Caimi. Space, time, error, and power optimization of image compressions transforms. *Proceedings SPIE*, 4122:176–189, 2000.

[77] Mark Schmalz. Cis4914: Senior project – main web page. course web site. `http://www.cise.ufl.-edu/~mssz/SeniorProject/senior-C01.html`.

[78] Peter W. Shor. Algorithms for quantum computation: Discrete log and factoring. In *Foundations of Computer Science, Proc. 35th Ann. Symp.*, pages 124–134. IEEE Computer Society Press, 1994.

[79] The stanford-berkeley-MIT-IBM NMR quantum computation project. web site. At Stanford University. `http://squint.stanford.edu`.

[80] Vlatko Vedral, Adriano Barenco, and Artur K. Ekert. Quantum networks for elementary arithmetic operations. Submitted to Physical Review A, November 1995. Los Alamos Physics Preprint Archive, http://xxx.lanl.gov/abs/quant-ph/9511018.

[81] Carlin Vieri. The pendulum instruction set architecture (PISA). MIT Reversible Computing Project internal memo. http://www.ai.mit.edu/~cvieri/pisa.ps, May 1996.

[82] Carlin Vieri. *Reversible Computer Engineering and Architecture*. PhD thesis, Massachusetts Institute of Technology, 1999.

[83] Carlin J. Vieri. "MIT AI lab reversible computing home page". web page. http://www.ai.mit.edu/~cvieri/reversible.html.

[84] Carlin J. Vieri. Pendulum: A reversible computer architecture. Master's thesis, MIT Artificial Intelligence Laboratory, 1995.

[85] Alexander Yu. Vlasov. Quantum computations and images recognition. Talk given at QCM'96 (Sep. 1996, Japan), November 1996. Preprint at http://arXiv.org/abs/quant-ph/9703010.

[86] D. J. Wineland, J. J. Bollinger, W. M. Itano, and D. J. Heinzen. Squeezed atomic states and projection noise in spectroscopy. *Physical Review A*, 50(1):67–88, July 1994.

[87] S. G. Younis. *Asymptotically Zero Energy Computing Using Split-Level Charge Recovery Logic*. PhD thesis, MIT Artificial Intelligence Laboratory, 1994.

[88] S. G. Younis and T. F. Knight, Jr. Asymptotically zero energy split-level charge recovery logic. In *International Workshop on Low Power Design*, pages 177–182, 1994.

[89] Christof Zalka. Grover's quantum searching algorithm is optimal. *Phys. Rev. A*, 60:2746–2751, 1999. http://arxiv.org/abs/quant-ph/9711070.