SAND REPORT
SAND2003-4688
Unlimited Release
Printed December 2003

Quantum Computing Accelerator I/O LDRD 52750 Final Report

Chris P. Tigges, Norman A. Modine, Lyndon G. Pierson,
Anand Ganti, Richard C. Schroeppel

Prepared by
Sandia National Laboratories
Albuquerque, New Mexico 87185 and Livermore, California 94550

Sandia is a multiprogram laboratory operated by Sandia Corporation,
a Lockheed Martin Company, for the United States Department of Energy's
National Nuclear Security Administration under Contract DE-AC04-94-AL85000.

Sandia National Laboratories

Issued by Sandia National Laboratories, operated for the United States Department of Energy by Sandia Corporation.

Available to DOE and DOE contractors from
U.S. Department of Energy
Office of Scientific and Technical Information
P.O. Box 62
Oak Ridge, TN 37831

Telephone: (865)576-8401
Facsimile: (865)576-5728
E-Mail: reports@adonis.osti.gov
Online ordering: http://www.doe.gov/bridge

Available to the public from
U.S. Department of Commerce
National Technical Information Service
5285 Port Royal Rd
Springfield, VA 22161

Telephone: (800)553-6847
Facsimile: (703)605-6900
E-Mail: orders@ntis.fedworld.gov
Online order: http://www.ntis.gov/help/ordermethods.asp?loc=7-4-0#online

SAND2003-4688
Unlimited Release
Printed December 2003

Quantum Computing Accelerator I/O
LDRD 52750 Final Report

Chris P. Tigges
RF Microsystems Technologies

Norman A. Modine
Nanostructure & Semiconductor Physics

Lyndon G. Pierson and Anand Ganti
Advanced Networking Integration

Richard C. Schroeppel
Cryptography and Information Systems Surety

Sandia National Laboratories
P.O. Box 5800
Albuquerque, NM 87185-0603

Abstract

In a superposition of quantum states, a bit can be in both the states "0" and "1" at the same time. This feature of the quantum bit or qubit has no parallel in classical systems. Currently, quantum computers consisting of 4 to 7 qubits in a "quantum computing register" have been built. Innovative algorithms suited to quantum computing are now beginning to emerge, applicable to sorting and cryptanalysis, and other applications. A framework for overcoming slightly inaccurate quantum gate interactions and for causing quantum states to survive interactions with surrounding environment is emerging, called quantum error correction. Thus there is the potential for rapid advances in this field.

Although quantum information processing can be applied to secure communication links (quantum cryptography) and to crack conventional cryptosystems, the first few computing applications will likely involve a "quantum computing accelerator" similar to a "floating point arithmetic accelerator" interfaced to a conventional Von Neumann computer architecture. This research is to develop a roadmap for applying Sandia's capabilities to the solution of some of the problems associated with maintaining quantum information, and with getting data into and out of such a "quantum computing accelerator".

We propose to focus this work on "quantum I/O technologies" by applying quantum optics on semiconductor nanostructures to leverage Sandia's expertise in

semiconductor microelectronic/photonic fabrication techniques, as well as its expertise in information theory, processing, and algorithms. The work will be guided by understanding of practical requirements of computing and communication architectures. This effort will incorporate ongoing collaboration between 9000, 6000 and 1000 and between junior and senior personnel. Follow-on work to fabricate and evaluate appropriate experimental nano/microstructures will be proposed as a result of this work.

Contents

Tables

Summary

Quantum Information Processing is expected to revolutionize the field of computing and communications. This field has demonstrated communications using "quantum entanglement", promising "entirely secure communications" (because interception of the communications will cause obvious changes in the message). Classically encrypted communications (which will continue to be required for any application, that must store ciphertext) will remain vulnerable to cryptanalysis. One of the first impacts in computing will start with certain classes of algorithms that include cryptanalytic key searches. In particular, quantum computing techniques may make the searching of much larger key spaces feasible, calling into question how long conventional cryptography will provide adequate protection for "strategic secrets".

Current work in quantum techniques for securing communications involves detecting whether any but the intended receiver has intercepted the encoded photons. The act of detecting the data also decrypts it. This quantum "photonic encryption" can be applied to a communication line (at low data rates over distances not requiring regeneration) but not easily to data in storage or for applications that may access protected data multiple times. Classical encryption techniques will continue to be used for many applications because of the limitations of photonic encryption. Because of these limitations, improved protection techniques against quantum cryptanalysis will require longer key lengths and/or quantum encryption computations on data that can be transformed (in encrypted form) back into the electrical and/or optical domain for compatibility with storage and retrieval systems. This will require the development of efficient means of transferring data into and out of a quantum computing engine. This work is to perform a detailed assessment of these developments and to plot an appropriate direction for further Sandia work in this area.

Utilizing quantum superposition, a quantum bit can be prepared so that it can be considered (in some contexts) to be simultaneously in both the "0" and "1" states. This feature of the quantum bit or qubit, has no parallel in classical systems whose bits have a definite state either "0" or "1". Currently, quantum computers consisting of 4 to 7 qubits in a "quantum computing register" have been built. Innovative algorithms suited to quantum computing are now beginning to emerge, applicable to sorting and cryptanalysis, and other applications. A framework for overcoming slightly inaccurate quantum gate interactions and for causing quantum states to survive interactions with surrounding environment is emerging, called quantum error correction. Thus, things are poised for rapid developments in this area.

Quantum information processing can be applied to secure communication (quantum cryptography) and to crack conventional cryptosystems. The first few computing applications will likely involve a "quantum computing accelerator" similar to a "floating point arithmetic accelerator" interfaced to a conventional Von Neumann architecture. This research is to address some of the problems associated with maintaining quantum information, and getting data into and out of such a "quantum computing engine".

We focussed this work on "quantum I/O technologies" by examining how to apply quantum optics on semiconductor nanostructures to leverage Sandia's expertise in semiconductor microelectronic/photonic fabrication techniques, as well as its expertise in

information theory, processing, and algorithms. The study was guided by understanding of practical requirements of computing and communication architectures. This effort incorporated ongoing collaboration between 9000, 6000 and 1000 and between junior and senior personnel. Follow-on work to fabricate and evaluate appropriate experimental nano/microstructures will be proposed as a result of this work.

Quantum photonic encryption is not expected to fully replace conventional cryptography because of the low throughput associated with engineering solutions to photon losses in optical fiber, and because the detection of an encoded photon results in its decryption into plaintext, so that such encrypted data cannot be stored or further processed in conventional systems. Quantum encrypting calculations (rather than photonic encoding) may eliminate this deficiency.

While further research is likely to uncover new technologies that may support Quantum Computing implementations, the current efforts are focused on 1) Trapped Ions, 2) Nuclear Magnetic Resonance, 3) Cavity Quantum Electrodynamics, 4) Quantum Dots, and 5) Quantum Photon Interferometry.

In the near term, progress will be made with Nuclear Magnetic Resonance (NMR), and later with Ion Traps; however, the most practical applications will likely come from approaches that take advantage of solid state technology such as Quantum Dots and Cavity QED. Although the solid state approaches are extremely challenging and progress has been slow, nevertheless the synergy with mature microelectronics technologies outweighs these liabilities.

The initial stages of this LDRD surveyed recent developments in all of these technologies. Even though the solution of practical problems (that are at or beyond the state of the art for solution with conventional computing technology) will require Quantum Computing registers consisting of about 25 qubits, current efforts have produced Quantum Computing registers of only 4 to 7 qubits. Current thinking is that scaling Nuclear Magnetic Resonance techniques to large numbers of qubits will be difficult. Even though scaling Ion Traps to large numbers of qubits is feasible in principle, maintaining the quantum coherence of large numbers of Trapped Ions will be difficult. Cavity Quantum Electrodynamics is a newer experimental arena for Quantum Computation, and may be especially useful in coupling quantum states stored in different technologies. Since SNL has great expertise in semiconductor nanostructures, the follow-on work proposed will likely focus on evolving the Quantum Dot technology and Cavity Quantum Electrodynamics.

The mathematics behind quantum information processing is relatively mature compared to the maturity of quantum computing devices. Algorithms for specific applications (in particular, for cryptanalysis) are well established, but the gulf between the QIP mathematics/algorithms and the engineering of physical devices that can support these computations is wide. This project developed synergy between information theorists and device physicists so that both Sandia communities can better understand the requirements and constraints associated with practical realizations of these applications. The lack of meaningful collaborations of this sort is a recognized weakness in the U.S. research program. One of Sandia's strengths is the interdisciplinary skills found in a unified environment. This diversity can be easily tapped requiring only the deliberate coordination and focus to break down inherent colloquial barriers. This requirement is important since device engineering is not easily factored from application requirements and theoretical developments. As a result, this area is rich in opportunities for interdisciplinary interactions. From these interdisciplinary interactions, great creativity

and innovation may evolve. Availability of this technology is expected to enable computations that are otherwise impossible with a conventional classical approach. In addition, quantum information processing has the potential to impact areas other than computation and communication. In particular, co-lateral applications of these technologies promises to revolutionize aspects of other technologies such as metrology (due to surprising aspects of parametric photon down conversion), precision range finding, and simulation. No doubt, as quantum information processing is increasingly realized and becomes more widely appreciated, further surprising applications will result. New business and markets will be enabled, providing new components for the advancement of Sandia's missions in nuclear weapons information security, covert communications, and sensor technologies.

Research in this area involves high technical risk and high potential gain. Some estimate a 20 year development time before Quantum Computing techniques become viable, although Quantum Communication and Quantum Photonic Encryption have already been demonstrated. Therefore, there is also the risk of unforeseen rapid developments in this area. Unexpected early advances in this field could enable applications of great concern to national security. For these reasons, a small effort (such as this LDRD) to assess these developments, potential impacts, and to carefully plot a sensible thrust for Sandia research is prudent.

In the near term, progress will be made with NMR, and later with Ion Traps; however, the most practical applications may come from approaches that take advantage of solid state technology such as Quantum Dots and the P/Si electron-nuclear coupled system. Although the solid state approaches are extremely challenging and progress has been slow, nevertheless the synergy with mature technologies outweighs these liabilities.

As described above, the technologies being explored involve great technical risk, as there are issues of scalability and implementability and of maintaining quantum coherence of qubits long enough to perform meaningful processing with each identified technology. This project has taken a interdisciplinary "hard look" at recent work and emerging technologies to determine an appropriate research path with reasonable technical risk for Sandia's efforts and to engage the appropriate collaborative partners.

The research areas most suitable for Sandia research include (1) high speed and high power electronics for control of quantum computing operations, (2) Micro-Electromechanical Machines (MEMS) Cantilever Technology for measuring qubit spin states, (3) Photonic Lattice Technology for controlled qubit couplings (enhancing decoherence times with photonic lattices), (4) GaAs-based quantum devices incorporating high mobility 2-D electron gasses, and (5) Quantum Device technology utilizing coherent electron transport in "quantum wires". The "systems" level studies include (6) efficient quantum error correction, (7) innovative quantum algorithms for a wide variety of applications, (8) decomposition of quantum algorithms so as to operate with fewer qubits (on smaller quantum computers), and (9) the advantages and limitations of computation of such algorithms using Hilbert Space Analog Computing rather than Quantum Computing.

We conclude that even though mathematical descriptions of "computationally complete" sets of quantum gates are fairly mature, full understanding of these mathematical models yet remain counter-intuitive to most practitioners. Further, progress in this area is limited (1) by lack of physical devices with which to realize Quantum Computing, (2) by lack of control structures through which to supervise

quantum operations, and (3) by lack of algorithms for which great gain in efficiency over classical algorithms can be demonstrated.

As more mathematicians, cryptographers, systems engineers, and device physicists and engineers interact regarding these issues, the strange interworkings of "quantum information processing" will become more intuitive, and progress will be made on algorithms and on quantum gate devices. It is recommended that an ongoing seminar series on advances in Quantum Computing be conducted to keep Sandia's device physicists and information theorists abreast of multi-disciplinary developments in this area.

In particular, Hilbert Space Analog Computing may be an area of rich productivity. Computing in a Hilbert Space is a superset of Quantum Computing, and is realizable (to the level of a few "Hilbert bits") in conventional microelectronics and/or in current programmable logic devices. By attempting to implement quantum-like algorithms in a Hilbert Space Computer, great insights may be gained into the architectures suitable for Quantum Computing and into the design of algorithms that may prove more efficient than classical algorithms.

Nomenclature

3DES	Triple DES. Design in 1999 by NIST to replace the vulnerable DES as computers advanced in development.
AES	Advanced Encryption Standard. Based on Rijndael algorithm. Is a Federal Information Processing Standard FIPS-197.
AND	AND binary boolean operation. The result is true if and only if both arguments are true.
bit	Leo Szilard invented the concept of a bit of information 1929. Smallest measure of classical information.
CNOT	Controlled-NOT gate. An important (perhaps single most useful) 2-qubit gate. Allows for reversible quantum NOT operation.
CSS	Calderbank-Shor-Steane quantum codes for correcting large qubit errors
DES	Data Encryption Standard. Federal Information Processing Standard FIPS-46-3. Originally developed as Lucifer by IBM in early 1970s.
dissipative	a process having a loss of energy in the form of heat and severe constraints associated with its recovery
entanglement	nonlocal quantum information distinct from classical information
GHz	Giga-Hertz, 109 Hz
Gray Code	encoding of numbers where adjacent numbers differ by 1 in a single digit
Hermitian	Type of operator defined by equating the operator with its adjoint (see Unitary)
HSC	Hilbert Space Computing
I/O	Input/Output
LFSR	Linear Feedback Shift Register. Important method in digital pseudorandom bit sequence generation.
MEMS	MicroElectroMechanical Systems
NMR	Nuclear Magnetic Resonance
MHz	Mega-Hertz, 106 Hz
Nonlinear-FSR	Nonlinear Feedback Shift Register. See LFSR
NOP	No OPeration. A computer operation usually taking a single cycle without further effect.
NP-complete	A problem that is NP (verifiable in nondeterministic polynomial time) and NP-hard (other problems can be translated to the referred problem)
QC	Quantum Computer
QECC	Quantum Error Correction Code
qubit	QUantum BIT, a two-state quantum mechanical system that encodes the basic information unit of a quantum computer. (see bit)
reversible	different operational definitions for logic and thermodynamics but intimately related
RSA	Encryption algorithm invented in 1978 by Ron Rivest, Adi Shamir, and Leonard Aldeman
Serpent	128-bit block cipher candidate for AES developed by Ross Anderson, Eli Biham, and Lars Knudsen.

syndrome	In the context of errors, the result of an operation on data indicating errors if any exist
THz	Tera-Hertz, 1012 Hz
Unitary	Type of operator defined by equating the operator inverse with its adjoint (see Hermitian)
XOR	Exclusive OR binary boolean operation. The result is true if and only if one of its arguments is true.

Quantum Computing Accelerator I/O

Introduction

Generally quantum information processing[1] and specifically quantum computing[2] have enjoyed a substantial increase in attention starting in the early 1980s that extends to the present time. Although much has happened in this recent period, it should not be surprising that the seeds of this new discipline predate this period. In fact, one might be surprised at how long it has taken for the present level of interest to occur. For the interested reader, a brief set of historical highlights is given in Appendix B.

At a fundamental level, it has become clear that an information theory based on quantum principles extends and completes classical information theory. The present theory includes quantum generalizations of classical notions regarding information sources, channels, and codes, but, more importantly, the theory distinguishes between two complimentary and quantifiable kinds of information: classical information and quantum entanglement. Classical information can be copied freely, but can only be transmitted forward in time and strictly within the forward light cone. Entanglement, by contrast, cannot be copied and is thought to be nonlocal.

A quantum state of qubits is represented by a complex unit vector in a 2^N-dimensional Hilbert space. This Hilbert space is defined as a tensor product with factors representing each of the N qubits composed of 2-dimensional Hilbert spaces describing individual qubits. The exponential dimensionality of this space distinguishes Hilbert Space Computing (HSC, a superset of QC) from classical computers, whose state is described by a number of parameters that grows only linearly with the size of the system. This is because classical systems, whether digital or analog, can be completely described by separately describing the state of each part. By contrast to both classical and classical HSC, the vast majority of quantum states are entangled, admitting no such similar description. The ability to preserve and manipulate entangled states is the distinguishing feature of quantum computers, and this ability is responsible for the power and the difficult synthesis of QCs.

Another distinguishing feature of an isolated quantum system is its evolution characteristics. These systems evolve so as to preserve superpositions and distinguishability. Mathematically, these transformations are unitary—that is , linear and inner-product-conserving. Such transformations are the Hilbert-space equivalent of rigid rotation in Euclidean space. Unitary evolution and superposition are the central principles of quantum mechanics that have significant consequences. One important consequence comes from the fact that unitary transformations are inherently reversible. It has been concluded that most computation operations can be done reversibly; however, there are important exceptions such as erasure. As a result, QC schema often resort to the incorporation of ancilla bits and to running backwards so as to "uncompute."

Concept of Quantum Computing I/O

The architecture of a quantum computer will be very different from a conventional computer. Even though some similarities will exist, i.e., multi-qubit registers will be required to input data and measurement of similar registers will be required to output the result of computations, the evolution of data contained in these registers will be under the control of quantum logical gate operations that bear little resemblance to classical logic gates. In addition, as these qubits interact with the environment, their quantum states will become "noisy", requiring extensive quantum error correction "circuitry" to maintain the fidelity of the computation. The evolution of quantum states that represent a quantum calculation, the error correction thereof, and the introduction of "ancilla" bits required to enable quantum operations, all will require the supervision of "quantum gate control hardware" that will be programmed and sequenced by conventional computer systems. The technology with which the qubits are implemented must be able to easily initialize the state of the qubits, to measure the resultant quantum states, and to convert to conventional binary digital representation, as well as to "entangle" the quantum states of other qubits.

The Church-Turing conjecture states that Quantum Computing algorithms perform at least as well as classical algorithms (and for some problems far better).[3,4,5] Is this provably true? Time will tell, but most quantum computing experts expect quantum computing machines to eventually perform the equivalent of any "turing" algorithm efficiently, and some quantum algorithms represent considerable gain in efficiency for certain problems over classical techniques. Even though any classical application could theoretically be adapted for computation on a quantum computing platform, algorithms for quantum computation with high gains in efficiency have been developed for only a few applications.

Grover's search algorithm represents a gain from O(N) to $O(\sqrt{N})$.[6] Shor's factoring algorithm represents even greater gain in efficiency over classical algorithms.[7] In spite of this, the first quantum computing machines will be very specialized, taking advantage of the problem solving efficiency gains for certain applications only. These specialized processors will be operated by interfacing to classical computers for input and output of problems, and for changing of parameters pertaining to the quantum computations. How will this Input/Output between classical and quantum computers be accomplished? The function of these specialized engines will resemble the function of specialized "hardware accelerators" for general purpose computers such as for Fast Fourier Transforms and other math "co-processors," but will utilize a technology that does not resemble our current digital electronics. For these reasons, the first few quantum computing engines will be very specialized, adapted for the solution of narrow classes of problems. These quantum computing engines will be interfaced as an "ancillary processor" to a classical computer (and will have its quantum gate computations controlled/supervised by a classical computer), much like specialized electronic hardware for acceleration of Fast Fourier Transforms or Floating Point calculations have been interfaced to general purpose computers.

There are several technologies that can conceivably be used for quantum computation. Some of these may lend themselves to electronic interface more easily than

others. Regardless of technology and interface technique, there are certain parameters to be communicated into a quantum engine in order to process a given algorithm, and there are quantum superpositions to be prepared and ultimately measured in order to identify the solution or solution space.

What are the operations that must be provided to perform input to a quantum computer, take output from a quantum computer, and/or to "steer," "adapt," or "select" quantum operations or algorithms? The following may be considered necessary if not sufficient: (1) a robust representation of quantum information, (2) preparation a fiducial initial state register, (3) addition of a set of states to a qubit superposition, (4) subtraction of a set of states from a qubit superposition, (5) the ability to perform a universal family of unitary transformations, (6) configuration of quantum gates to control "quantum evolution" of qubits for specific operations, and (7) "measurement" of a q-register and output of its contents to a classical computer.[8]

Fundamentals of Quantum Computation

The four fundamentals of Quantum Computing are (a) State, (b) Operations, (c) Measurement, and (d) Composite States.[8]

State

In classical computing, information is stored in binary strings. In the case of a classical computer with n bit registers, the contents of each register would be one of the possible 2^n strings. So, information storage in classical computing is discrete. However, in quantum computing information is stored in a continuum. An n bit quantum register would store a unit vector in a Hilbert space of dimension 2^n. Abstractly, the standard basis for this Hilbert space is denoted as $|00...0\rangle, |00...1\rangle, \ldots, |11...1\rangle$.

The state of a n-bit quantum register is an arbitrary superposition (linear combination) of the basis vectors. For example, a one bit quantum register would contain a unit vector represented as $a|0\rangle + b|1\rangle$, where a,b are complex numbers such that $a^2 + b^2 = 1$. So, in order to describe the contents of a n bit quantum register one would require 2^n complex numbers. This exponential storage capacity of quantum systems is the root of the potential power of a quantum computing. In classical computing one can manipulate only n bits at a time with a single operation on a n bit register. However, a single quantum operation can simultaneously manipulate the 2^n complex numbers specifying the state of a n qubit register.

Operations

The allowable operations on a quantum state are unitary transformations. An interesting consequence of this constraint is that QC is reversible. That is to say, in

principle all the information required to run the computation forward is sufficient to also run the computation backward. It should also be noted that operations are not allowed to be a function of the state. The laws of quantum mechanics impose these constraints. At the risk of redundancy, in quantum mechanics all processes are reversible, and an operation on a quantum state must lead to another valid state. So operations in quantum computing require Linear Algebra[9] as opposed to Boolean Algebra required in classical computing.

If $|\psi_1\rangle$ is the state at time t_1, then one could generate a state $|\psi_2\rangle$ by applying a linear operator U_{12} at time t_1. Since we have the constraint that U_{12} is a unitary transformation $\langle\psi_2\|\psi_2\rangle = \langle\psi_1| U^{\dagger}_{12}U_{12}|\psi_1\rangle = 1$. Herein, we use $\langle\psi|$ to denote the adjoint of $|\psi\rangle$ and $U^{\dagger}$ to denote the adjoint of U.

Measurement

Measurements in quantum computing are fundamentally different from measurements in classical computing. The differences are that (a) the outcome of a measurement is intrinsically probabilistic and (b) the outcome of the measurement affects the state of the system. If the state of a quantum register is unknown and a non-trivial superposition of the basis states, then one can never discover the superposition using a single quantum measurement.

There are two equivalent ways of describing quantum measurements. One is through a set of measurement operators $\{M_m\}$, and the other is through projections onto the eigenspace of a Hermitian operator called the observable. We now describe the first method and for simplicity, assume that the system is non-degenerate.

Suppose we are measuring a quantum state $|\psi\rangle$ using $\{M_m\}$. The set $\{M_m\}$ is required to satisfy the *completeness* equation

$$\sum_m M_m^{\dagger} M_m = I.$$

The measurement outcome can be any of the m values, with the probability of the m th outcome being $\langle\psi|M_m^{\dagger}M_m|\psi\rangle$. If the outcome of the measurement is m, then the state after the measurement

$$\frac{M_m|\psi\rangle}{\sqrt{\langle\psi|M_m^{\dagger}M_m|\psi\rangle}}$$

Projective Measurements

Another, equivalent, method of describing measurements is through projections onto the eigenspace of a Hermitian operator H. Since H is Hermitian, it is diagonalizable with its eigenvectors being orthogonal. Suppose

$$H = \sum_m \lambda_m V_m$$

where V_m are the eigenmatrices corresponding to eigenvalues λ_m of H. The eignematrices satisfy the following properties: $V_n V_m = \delta_{nm}$, and $V_n^{\dagger} = V_n$.

The measurement of $|\psi\rangle$ with H could lead to one of the eigenvalues with the probability of measuring λ_m being $\|V_m|\psi\rangle\|^2 = \langle\psi|V_m|\psi\rangle$. If the outcome of the measurement is λ_m, then the state after the measurement is

$$|\psi_m\rangle = \frac{V_m|\psi\rangle}{\sqrt{\langle\psi|V_m|\psi\rangle}}$$

It can be shown using the above framework that one can only distinguish orthogonal quantum states using simultaneous measurement.

Supposing that λ_m is the outcome of a measurement for the observable H, further repeated measurements result in the same outcome λ_m. Therefore, the initial state has been transformed into $|\psi_m\rangle$ by the first measurement. This phenomenon is called the collapse of the wave function. Although, one could in principle be in a state that is a superposition of $|In\_the\_room\rangle$ and $|Outside\_the\_room\rangle$, it is the continuous interaction with the environment that results in effectively measuring and collapsing the state into one or the other possibility.

Composite States

If $|\psi_1\rangle$ is the state of one quantum register and $|\psi_2\rangle$ is the state of another, then the state of the two registers is given by their tensor product $|\psi_1\rangle \otimes |\psi_2\rangle$. In general several notations are in vogue to represent tensor products such as $|\psi_1\rangle|\psi_2\rangle, |\psi_1\psi_2\rangle$.

Quantum Algorithms

In principle, a quantum computer can efficiently simulate a classical computer. In practice, however, if the classical and quantum algorithms have the same time complexity, one would prefer the classical algorithm. The reason for this preference is that classical computers by their very nature are more robust and have faster operation execution times. The question remains whether there are problems for which a quantum computer is exponentially faster than a classical computer. Currently, there are a few examples where a quantum computer is faster than a classical computer. The algorithms all have the property that if implemented they would need to interface with a classical computer. We list the examples below.

Deutsch Algorithm

Problem: We are given a function $f:\{0,1\}\mapsto\{0,1\}$ and we would like to find out if f is a constant function. On a classical computer we would need to evaluate f at 0 and 1, whereas the Deutsch algorithm solves this problem with a single call to f. The algorithm assumes that there is a quantum black box Q_f that can evaluate the function f. In general f might be a complex function implemented on a classical computer and Q_f would need to interface with it.[10]

Grover's Search Algorithm

Problem[11]: We are given a set $S=\{s_1,s_2,...s_N\}$, and a function $f:S\mapsto\{0,1\}$. The function f is such that there is a unique element in S, which is mapped to 1. The object is to find the element that is mapped to 1. Classically this requires $O(N)$ operations in the worst case, whereas Grover's Search Algorithm achieves this with $O(\sqrt{N})$ operations.

The algorithm assumes that there is a quantum black box Q_f for computing f. The black box essentially implements the mapping $|s_i,b\rangle\mapsto|s_i,b\oplus f(s_i)\rangle$ where b is a single qubit. If this search algorithm were to be implemented for practical applications such as database searches, then clearly Q_f would need to interface with a classical computer.

Shor's Factoring Algorithm

Problem[7]: We are required to find all the factors of a given number N. The fastest classical algorithm currently is the number field sieve[12], which takes $O(\exp(2(\log N)^{1/3}(\log\log N)^{2/3})$ operations.

Shor's algorithm divides the problem of factorization into two phases. The first phase runs on a quantum computer and computes a fraction $\frac{c}{r}$ where r is the order of a random number mod N and c is an arbitrary constant. The second phase involves computing r from $\frac{c}{r}$ using a partial fraction expansion on a classical computer and using

the order r to find a factor of N. The algorithm in total takes $O((\log N)^{1/3})$ operations to factor N.

Shor's Discrete Log Algorithm

Although the first two algorithms are provably faster than classical algorithms, the last two algorithms aren't. But it is generally believed that no polynomial time algorithm exists for factoring or for finding discrete logarithms. The Quantum algorithms, on the other hand, are provably polynomial time algorithms.[7]

Quantum Error Correction

A quantum computer will invariably interact with the environment in unpredictable and technologically unavoidable ways. These interactions lead to what is known as decoherence. Essentially qubits will accumulate errors caused by environmental interactions. In addition to decoherence, the inevitability and importance of imperfect quantum gate implementation has been recognized. In order for the quantum computer to work successfully, we need to combat both the storage and gate errors. For some time it was thought that the no cloning constraint prevented the possibility of quantum error correction theorem.[13] However, in 1995 Peter Shor proposed a scheme that cleverly avoided the cloning issue, and, subsequently, quantum error correction as discipline matured at a rapid pace.

There is an essential difference between classical algebraic coding and quantum coding with respect to the class of errors. In classical algebraic coding, errors are discrete (alphabet flips), whereas in quantum coding errors are continuous in nature. The theory of Quantum Error Correcting coding describes how qubits can be efficiently encoded, so that the quantum information can be recovered even after the encoded bits accumulate errors. The theory of fault-tolerant computation[14] describes how equivalent gate implementations on encoded bits can be made so that a computation can be successfully executed, even with gate errors. In the next few sections, we will describe a framework for an error correction and describe a few classes of quantum error correcting codes and bounds for error correcting codes.

Framework for Error Correction

Consider a single qubit $|\psi\rangle = a|0\rangle + b|1\rangle$ that interacts with the environment. The interaction transforms $|\psi\rangle$ to one of the following possibilities

$$I|\psi\rangle = a|0\rangle + b|1\rangle,$$

$$X|\psi\rangle = a|1\rangle + b|1\rangle,$$

$$Y|\psi\rangle = a|0\rangle - b|1\rangle,$$

and

$$Z|\psi\rangle = a|1\rangle - b|0\rangle.$$

where $\{I, X, Y, Z\}$ are the single qubit Pauli operators. In the above list the first item corresponds to no error, the second item corresponds to a bit error, the third item corresponds to a phase error and the last item corresponds to a bit and a phase error.

The goal of error correction is to guard against the last three items and consists of three steps: encoding, measuring, and correcting. The first step is to encode the single qubit into multiple qubits. This leads to the information being stored across multiple qubits. This nonlocal storage caused by entanglement is recognized as one of the most important and distinguishing features of quantum information processing, and error correction is a good example of its power. As to the nature of decoherence error, it is often assumed that these errors are local – that is they impact each bit in the collection in an independent manner. The second step involves performing a collective measurement on the multiple bits to determine the *nature* of the error without actually discovering the *exact* error. The key observation, made by Shor, is that the *exact* error does not need to be discovered in order for useful error correction to be applied. In fact, knowledge of *exact* error must be avoided. Once the *nature* of the error is determined, a corrective step using a set of operations specifically chosen and applied to reverse the suspected error can be accomplished.

Let Σ be the set of error operators we wish to correct using a Quantum Error Correcting Code (QECC) C. Then C must have the following property

$$jE_b^{\dagger}E_a i = \Gamma_{ab}\delta_{ij} \qquad \textbf{(1)}$$

for all error operators E_b, E_a in Σ and for all code words i, j in C. Here Γ_{ab} is a real number dependent on E_a, E_b and δ_{ij} is the Kronecker delta function. It can also be shown[15] that the above condition is sufficient for error correction.

We will try to explain why eq. (1) is necessary and sufficient for quantum error correction. Since we can only distinguish orthogonal states using quantum measurement, we require that error operators acting on different code words map them into orthogonal vectors. This explains the presence of δ_{ij}. The constant Γ_{ab} is more subtle to explain. Our initial reaction would be that it must be δ_{ab}. In this case, different operators acting on the same code word map it into orthogonal vectors. Although this condition is certainly sufficient for error correction, it is not necessary since we are only interested in correcting the error.

Quantum Error Correcting Codes

Preliminaries

In this document we will restrict ourselves to Quantum codes that are binary. A (n, k, d) binary code, where $k \leq n$, is a 2^k element subset of a 2^n dimensional space. The 2^k elements that make the code are called code words. The code can be represented using bits and hence is called a binary code. The parameter n represents the number of bits (code length) in each code word and k represents the number of data bits being

encoded. The so called distance parameter d determines the robustness of the code to errors.

We will first explain the (n,k,d) terminology by means of a simple classical binary code. Consider the classical binary repetition code $\{00,11\}$ that is a subset of $\{00,01,10,11\}$. In this case $n=2$ and $k=1$, since there are 2^2 two bit words and 2^1 of them are code words. In classical binary codes d represents the minimum number of bit flips one needs to make on an arbitrary codeword to generate another codeword, which in the example is 2. If our (n,k,d) code is required to correct t bit flip errors, then we need $t \le d-t-1$, i.e. the distance from the corrupted codeword to the actual codeword must be at least one less than the distance from the corrupted codeword to another codeword. It follows that an (n,k,d) code is capable of correcting $\left\lfloor \frac{d-1}{2} \right\rfloor$ bit flip errors. In our example the (2,1,2) code can correct $\left\lfloor \frac{2-1}{2} \right\rfloor = 0$ errors, which is obvious by looking at the code. So in fact we have shown that we need at least three bits to correct a bit flip error!

Classical coding theory unlike Quantum coding theory is quite old and there are several references varying in breadth and depth. The *Handbook of Coding Theory*[14] is an exhausting tome on this subject.

In quantum coding, although the notions of n and k extend in a straightforward manner from classical coding, the notion of distance d is slightly different. Whereas there are only bit flip errors in classical coding, there are bit flip, phase flip and bit and phase flip errors in quantum coding. The distance d, in quantum coding is defined in terms of the *weight* of Pauli operators. The *weight* of a Pauli operator is the number of bits of a codeword that it affects in a non-trivial manner. For example the Pauli operator Z_1X_2 has a weight of two. The distance d of a QECC is defined as the minimum weight of a Pauli operator that can convert one codeword to another.

Now we give an example of a (9,1,3) QECC which was discovered by Peter Shor. The code has two code words

$$|\bar{0}\rangle = \frac{1}{\sqrt{2}}(|000\rangle + |111\rangle)^{\otimes 3}$$

$$|\bar{1}\rangle = \frac{1}{\sqrt{2}}(|000\rangle - |111\rangle)^{\otimes 3}$$

In order to show the distance is 3, note that the code words are tensor product of three identical clusters and one requires a weight 1 linear operator to convert the cluster $\frac{1}{\sqrt{2}}(|000\rangle + |111\rangle)$ to $\frac{1}{\sqrt{2}}(|000\rangle - |111\rangle)$ or vice versa.

When does coding help?

In general, consider a (n,k,d) code. It takes k data qubits and maps into n code qubits. Let ε be the probability that one qubit undergoes an error within one coherence time and assume that errors in different bits are independent. If there were no coding

then the block of k qubits would be corrupted with probability $k\varepsilon + O(\varepsilon^2)$. With error coding the probability that the block would be corrupted is

$$\binom{n}{d+1}\varepsilon^{d+1}(1-\varepsilon)^{n-d-1} + O(\varepsilon^{d+2}).$$

So, error coding is useful only if

$$\binom{n}{d+1}\varepsilon^{d+1}(1-\varepsilon)^{n-d-1} + O(\varepsilon^{d+2}) < k\varepsilon + O(\varepsilon^2).$$

This imposes an upper bound on ε. For example, if one evaluates the robustness of the (7,1,3) Steane code[16] along the above lines, then one obtains the constraint

$$\binom{7}{2}\varepsilon^2 \sim< 1 \Leftrightarrow \varepsilon \sim< \frac{1}{\sqrt{2}}.$$

So, for coding to be useful, we require that the probability of a qubit error to be below a threshold.

CSS Codes

There is a subclass of classical binary codes called classical binary linear codes that have the property that modulo 2 addition of any two words in the code results in another codeword. Binary linear codes have several useful properties that can be exploited to design fast encoding and decoding algorithms. The Calderbank-Shor-Steane (CSS)[16,17] codes are built from classical binary linear codes. The construction is given as follows.

Let $i \in \{1,2\}$ and C_i be a (n,k,d_i) code with a $n-k_i \times n$ parity matrix H_i. We also assume that C_2 is a proper subcode of C_1. The subcode C_2 defines a equivalence relation on C_1, given by the following. Two code words $u,v \in C_1$ are considered to be equivalent if there exists a $w \in C_2$, such that $u+w=v$. The subcode C_2 divides C_1 into $2^{k_1-k_2}$ equivalence classes. The CSS code is constructed by selecting a codeword for each equivalence class using $\frac{1}{\sqrt{2^{k_2}}}\sum_{v\in C_2}|v+w\rangle$.

The process of error correction is done in the following manner. We first consider bit flip errors. Note that each CSS codeword is a superposition of code words in C_1. Hence, we can perform a parity check using the parity check matrix H_1 of C_1 to correct up to $\left\lfloor \frac{d-1}{2} \right\rfloor$ bit flip errors. To correct phase errors we make the following key observation: phase errors in the standard basis are transformed into bit flip errors in the Hadamard basis. The following illustrates this point.

Let $|\psi\rangle$ be an n qubit standard basis state and let H be the Hadamard rotation. Let e be an arbitrary binary vector and let $E_p^{\,e}$ denote phase flips at all bits where e is one. Similarly let $E_f^{\,e}$ denote bit flips at all bits where e is one. The Hadamard rotation of a phase flipped standard basis vector $|\psi\rangle$ is given by

$$H\left(E_p^{\,e}|\psi\rangle\right) = H\left(-1^{\psi\cdot e}|\psi\rangle\right) = \frac{1}{\sqrt{2^n}}\sum_j -1^{\psi\cdot(e+j)}|j\rangle \ . \tag{2}$$

The bit flip of a Hadamard rotated standard basis vector $|\psi\rangle$ is given by

$$E_f^{\,e}\left(H|\psi\rangle\right) = E_f^{\,e}\left(\frac{1}{\sqrt{2^n}}\sum_j -1^{\psi\cdot j}|j\rangle\right) = \frac{1}{\sqrt{2^n}}\sum_j -1^{\psi\cdot j}|j+e\rangle = \frac{1}{\sqrt{2^n}}\sum_i -1^{\psi\cdot(i+e)}|i\rangle \tag{3}$$

The last equation is obtained by changing the summation index and using $2e = 0(\mathrm{mod}\,2)$. From equations (2) and (3) we obtain that phase flip errors in the standard basis are transformed into bit errors in the Hadamard basis.

When we apply the Hadamard transformation to the CSS code words we obtain a superposition of code words in the dual code to C_2. We can then use the generator matrix G_2 for C_2 to detect bit flip errors in the rotated basis. So, in effect, we can correct $\left\lfloor \frac{d_2' - 1}{2} \right\rfloor$ phase errors, where d_2' is the distance of the dual code to C_2.

The 7-qubit Steane Code

The simplest of the CSS codes is the 7-qubit (7,1,3) code discovered by Andrew Steane.[16] In this case C_1 is the (7,4,3) Hamming code and C_2 is the (7,3,4) subcode containing only the even code words of C_1. It so happens that the dual code to C_2 is C_1. So we can use the parity check matrix of C_1 to correct both bit and phase errors. The code is described as follows

$$|\bar{0}\rangle = \frac{1}{\sqrt{8}}\left(|0000000\rangle + |0001111\rangle + |0110011\rangle + |0111100\rangle + |1010101\rangle + |1011010\rangle + |1100110\rangle + |1101001\rangle\right)$$

$$|\bar{1}\rangle = \frac{1}{\sqrt{8}}\left(|1111111\rangle + |1110000\rangle + |1001100\rangle + |1000011\rangle + |0101010\rangle + |0100101\rangle + |0011001\rangle + |0010110\rangle\right)$$

To perform the error correction we augment the 7 qubit code with 6 ancilla bits, 3 of which are for bit errors and the remaining 3 for phase errors. That is we perform the operation

$$|\psi\rangle \otimes |0\rangle \otimes |0\rangle \mapsto |\psi\rangle \otimes |H\psi\rangle \otimes |H(\mathrm{H}|\psi\rangle)\rangle$$

where H denotes the Hamming matrix and **H** denotes the Hadamard operator. We then measure the ancilla bits to correct bit flip or phase errors, if there are any. This code will correct a single bit flip or a phase flip or a single bit flip and a phase flip error.

Stabilizer Codes

Stabilizer codes are constructed using the group property of Pauli Operators. Recall that the Pauli operators are I, X, Y, Z. The n-fold tensor product of Pauli operators also forms a group Γ_n of order $2 \cdot 4^n$. Γ_n has the following properties: each $M \in \Gamma_n$ is unitary, i.e., $M^{-1} = M^{\dagger}$. $M^2 = \pm I \ \forall \ M \in \Gamma_n$. Furthermore, $M^2 = I$, if the number of "Ys" in M is even and $M^2 = -I$, if the number of "Ys" in M is odd. $MN = \pm NM$ for all $M, N \in \Gamma_n$.

Let S denote an Abelian subgroup (a commutating subgroup) of Γ_n. Then the stabilizer code C_S associated with S is the simultaneous eigenspace of all elements in S with eigenvalue 1. Mathematically $|\psi\rangle \in C_S$ iff $M|\psi\rangle = |\psi\rangle$. The group S is called the stabilizer of the code, since it acts like the identity transformation on the code C_S.

The group S is characterized by a set of independent matrices called generators. The generators have the property that every element in S can be expressed as a product of the generators and no generator can be expressed as a product of other elements of the group. It can be shown[15] that if the number of generators is $n-k$, then the number of code words in C_S is 2^k. In other words, the number of generators of S determines the number of bits encoded by C_S.

Error correction is done by measuring the generators on the possibly corrupted code words. Let $G_1, \ldots, G_{n-k}$ be the generators of S. Suppose a code word $|\psi\rangle$ is acted on by an error operator E_a; then it is detectable if it anti-commutes with some generator G_i. In this case one can detect the error by measuring G_i, since $G_i E_a |\psi\rangle = -E_a G_i |\psi\rangle = -E_a |\psi\rangle$. To perform error correction, one first measures the received code word with all the generators. Let $\lambda_1, \ldots, \lambda_{n-k}$ be the collection of the measurements called the syndrome. Assume the error operator satisfies the necessary conditions for error correction, i.e., it belongs to the set Σ that satisfies eq. (1).

If an error operator E_a results in a unique syndrome, then one can correct it by applying $E^{\dagger}_a$ on the received word. Suppose the syndrome is not unique, i.e., E_a and E_b result in the same syndrome, then $E_b^{\dagger} E_a \in S$. So one can perform error correction by applying $E^{\dagger}_b$ on the received word.

Concatenated Codes

Concatenated coding is the process of increasing code distances by repeating the encoding process multiple times. Although this process leads to code distances increasing geometrically, it also results in code lengths increasing geometrically. We now illustrate

this process. Consider a (n,k,d) QECC. It takes k qubits and encodes it into n code qubits. Suppose we take each of the n code qubits and re-encode them using the (n,k,d) QECC. We now have a code that takes k qubits and encodes it into n^2 code qubits. But we have also increased the distance to d^2, since we now require d error operators of weight d to convert one code word to another. If we repeat this process m times, then we obtain a (n^m,k,d^m) code.

Quantum Error Code Bounds

The field of error code bounds involves finding relationships between n,k,d, i.e., the length of the code, the number of code words and distance between code words. Intuitively if one wants greater number of code words in a given code length, then one has to trade off the distance of the code. In the following subsections we quantify this notion.

Quantum Hamming Bound

Consider a (n,k,d) QECC that can correct up to t errors (bit flips, phase flips, both) which is non-degenerate. A non-degenerate code satisfies equation eq. (2) with $\Gamma_{ab} = \delta_{ab}$, i.e., distinct errors result in orthogonal vectors.

Note that a qubit error can be a bit flip, phase flip or a combination of both, and so, if a codeword has $j < t$ errors, then it can result in 3^j possible orthogonal vectors. So a single codeword with j errors can generate $\binom{n}{j}3^j$ orthogonal error vectors. Therefore, the total number of orthogonal error vectors in the code is $2^k\sum_{j=1}^{t}\binom{n}{j}3^j$. Now the error vectors as well as the code words must be accommodated in the original subspace of dimension 2^n, from which follows the Hamming bound.

$2^k\sum_{j=1}^{t}\binom{n}{j}+2^k \le 2^n \Leftrightarrow \sum_{j=0}^{t}\binom{n}{j}3^j \le 2^{n-k}$ where $t=\left\lfloor\frac{d-1}{2}\right\rfloor$.

A QECC satisfying the Hamming bound with equality is called a perfect code. Consider the case of a QECC that can correct a single error. Substituting $t=1$ in the Hamming bound we obtain that $3n+1 \le 2^{n-k}$. If, furthermore, we require that the code contain only two code words, i.e., $k=1$, then we have $3n+1 \le 2^n/2$. The smallest n that satisfies the above inequality is 5, implying that we can do better than the 7-qubit Steane code described earlier.

Note that the quantum Hamming bound is only valid for non-degenerate codes. However, as of yet, no degenerate code that violates the Quantum Hamming bound has been discovered.

A Perfect Code

Recall that a perfect code is one that satisfies the Quantum Hamming bound with equality. We now describe a (5,1,3) perfect stabilizer code. The stabilizers of the code are

$$M_1 = XZZXI$$

$$M_2 = IXZZX$$

$$M_3 = XIXZZ$$

$$M_4 = ZXIXZ$$

One can check that each Pauli operator of weight 1 or 2 anti-commutes with at least one of the generators. So the distance of the code is at least 3. In other words one can correct a Pauli operator of weight 1, i.e., a single error.

Fault Tolerant Computation

So far we have outlined how to use error correcting codes to make bits more robust to decoherence. However, this is only a partial solution to the problem of operating a quantum computer in an error prone environment. Since our data bits are encoded now, we must have a method of processing encoded bits in a manner as to be equivalent to the processing of data bits. In other words, we need to determine how to implement the fundamental gates on encoded bits, so that the output is consistent. For example, suppose $|\psi\rangle$ is a data qubit and a certain computation requires implementing $X|\psi\rangle$. If we have encoded $|\psi\rangle$ as $|\overline{\psi}\rangle$ using a QECC, then we would like to produce an operation $\overline{X}$ such that $\overline{X}\,|\overline{\psi}\rangle$ is the QECC encoding of $X|\psi\rangle$. In order to perform a generic computation in a robust manner, one must have equivalent robust implementations of a set of universal gates. Note that even the gates might have inherent errors and one must have a mechanism to combat both the gate errors as well as storage errors. The process of encoding clearly involves gate operations, and the process of error correcting involves measurement that might be error prone as well. This process of building a robust universal set of gates is fault-tolerant computation.

At first glance, this problem might seem to be insurmountable. We need to encode bits to protect them from errors, but the process of encoding might introduce errors. Furthermore, we need to measure and correct errors based on the measurements, but the process of measurement in itself is error prone. So, as a result, one might perform an erroneous correction due to an error in the measurement process. However, it can be shown that if the storage, gate, measurement errors of the underlying technology implementing the QC are statistically independent and below a certain threshold, then one can perform arbitrarily long computations with arbitrary small probability of error. Now classical computers do not encounter this issue of fault-tolerant computation, since the underlying technology is extremely reliable. However, Von Neumann[18] considered the same problem for classical computers and, under the assumption that transfer of bits from one gate to another was reliable, came up with a fault-tolerant computer. Gac[19] has further improved the solution by removing the assumption of perfect transfer of bits.

Clearly, the fault-tolerant implementation of the fundamental gates is a function of the QECC used in encoding the data bits. Now, the QECC selected to perform the

encoding is a function of the computational complexity of the algorithm being implemented, i.e. the size of the problem being processed on the QC. So a generic QC may achieve fault tolerance in two ways. One might estimate the largest problem that can be computed on it and select a QECC and a fault-tolerant implementation of a universal set of gates in hardware. A second approach might be to have a mechanism to select a QECC and produce corresponding fault-tolerant implementations of a universal set of gates as a function of the size of the problem. While the former approach is a lot simpler, the latter has the advantage of being more efficient.

Fault-tolerant computation also introduces a tradeoff into the process of selecting a QECC. It might be the case that there are two different QECCs C_1, C_2 with the same distance. However C_1 uses fewer bits to perform the encoding than C_2. If it were just the matter of storing bits clearly, C_1 would be preferred over C_2. But consider the scenario where the fault-tolerant implementation of a universal set of gates for C_2 involves fewer gates than for C_1. In this case one might choose C_2 over C_1.

Device Technology

Quantum computing hardware development is clearly in its infancy. However, significant advances have occurred. Arguably the most difficult part of designing and building quantum computing hardware is getting two qubits to interact with one another. In this regard, Table 1 itemizes the state of the art in the development of a few prototypical systems for quantum computation. Further advances are expected, but problems and even significant limitations have been anticipated. The scale of existing hardware most likely will have to increase by many orders of magnitude before truly interesting calculations can be attempted. This scale increase must be achieved for both qubit storage and gate applications. It is likely, if quantum computers are to be practical, new ideas about the construction of quantum hardware will be required.

Many possible physical implementations of a quantum computer have been proposed. The essential ingredients of such an implementation are a series of quantum mechanical two-state systems that encode the qubits of the quantum computer, a means of applying field pulses that control the couplings between the two states of each qubit and between neighboring qubits, and the ability to perform a quantum mechanical measurement of the state of a qubit. Leading proposals for the physical implementation of a quantum computer are based on nuclear magnetic resonance (NMR), ion traps, Josephson junctions, optical cavities, quantum dots, quantum wires, and impurities in semiconductors.

An important property of a physical implementation of a quantum computer is the difference in energy between the two states of a qubit. This determines a fundamental frequency for the qubit representation. Typical fundamental frequencies range from values measured in MHz for a representation based on nuclear spins, to GHz for electronic spins, to optical frequencies for electronic transitions. This fundamental frequency gives the carrier frequency of the field pulses used to manipulate the qubits,

and it typically gives an upper bound on the rate at which gates can be performed on the quantum computer. The energy difference associated with the fundamental frequency can also be converted to a characteristic temperature. The temperature of the qubits must be less than this value in order to prepare a pure state for use in quantum computation. Typical characteristic temperatures range from a few milli-Kelvin (for nuclear spins) to a few Kelvin (for electronic spins) to thousands of Kelvin (for electronic transitions).

SYSTEM	ACHIEVEMENT	REFERENCES
Trapped ions	4-qubit entanglement	20, 21
Cavity QED	2-qubit entanglement	22, 23
NMR	7-qubit operation	24, 25

Table 1. State of the art of some prototypical quantum computing systems.

Another important property of a physical representation of a quantum computer is the coherence time. In general terms, the coherence time is the time over which an arbitrary superposition of the states of the quantum computer can be maintained. At minimum, sustained computation requires that one cycle of the error correction algorithm can be run within the coherence time. A detailed definition of the coherence time tends to be complicated and system dependent since there can be several different mechanisms that can degrade different aspects of coherence. However, upper bounds on the coherence time can be obtained by considering specific mechanisms. For example, spontaneous emission of electromagnetic radiation is one mechanism that destroys coherence. In 3-dimensional free space, the spontaneous emission rate scales as the cube of the fundamental frequency due to the density of states of photon modes, and typical spontaneous emission lifetime ranges from microseconds for optical transitions to millions of years for nuclear spins. In practice, mechanisms other than spontaneous emission dominate decoherence for systems with small fundamental frequencies, and as a result real coherence times for nuclear spin systems are typically measured in seconds.

Nevertheless, quantum computer implementations based on nuclear spins (NMR and ion traps) naturally have relatively long coherence times, and, largely as a result, the greatest progress toward quantum computation has been reported within these paradigms. However, both of these approaches are believed to have serious limitations that may prevent scaling of the number of qubits needed to perform practical calculations. These limitations arise because the characteristic temperature associated with a nuclear spin representation is small compared to temperatures that practically can be maintained in the laboratory. NMR and ion trap based quantum computers have taken two different approaches to overcoming this difficulty. In the case of NMR, researchers have worked with an ensemble of systems in a mixed state and used the "effective pure state" approach to separate the signal from a single component of the mixed state. This approach has been successful for small numbers of qubits, but the resulting output signal strength decays exponentially with the number of qubits used in the computation. In the case of ion traps, special laser cooling techniques have been used to initially cool the qubits to

below their characteristic temperature. This demanding technique depends on the extreme thermal isolation achieved in an ion trap, cannot be maintained during the computation, and becomes increasingly difficult as the number of trapped ions is increased.

These limitations of NMR and ion trap based approaches suggest that quantum computer implementations where the characteristic temperature can be maintained practically in the laboratory may ultimately be more fruitful. In addition, Sandia's institutional expertise overlaps most strongly with implementations of a quantum computer based on quantum dots or wires and impurities in semiconductors, that principally represent qubits as electronic spins or electronic states. Since these approaches are based on solid-state technology, it is hoped that the same technologies responsible for the remarkable scaling of integrated circuits over the last thirty years can be applied to scale a quantum computer to an adequate number of qubits. However, the fundamental frequencies of these solid-state implementations are high and their natural coherence times are quite short (see Table 2) representing a significant control/interface challenge. It needs to be emphasized that the values presented in this table are not intrinsic, can vary considerably even within a system class, and are subject to technological context *(a constantly moving target)*. These short coherence times are largely compensated by the correspondingly high speeds (GHz to THz) at which quantum gate operations can, in principle, be performed. However, applying well-controlled field pulses at these high speeds is likely to put extreme demands on the input electronics controlling the pulses. Furthermore, depending on how well losses can be controlled in the implementation, a considerable amount of power will likely be needed to create the strong fields that enable these rapid gate operations. Sandia's expertise in high speed, high power electronics may allow us to make a major contribution to the specialized input electronics of any future solid-state implementation of a quantum computer.

System	τ_s [s]	τ_c [s]	ratio
Electrons in GaAs	10^{-13}	10^{-10}	10^{3}
Electrons in Au	10^{-14}	10^{-8}	10^{6}
Trapped ions	10^{-14}	10^{-1}	10^{13}
Optical microcavity	10^{-14}	10^{-5}	10^{9}
Electron spin	10^{-7}	10^{-3}	10^{4}
Electron quantum dot	10^{-6}	10^{-3}	10^{3}
Nuclear spin	10^{-3}	10^{4}	10^{7}

Table 2. Comparison of characteristics times for several quantum systems.[8,26] The minimum time to execute a gate operation is $\tau_s = \frac{\hbar}{\Delta E}$. The coherence time, τ_c, is subject to improvement with technological advance.

Suggestions for Future Work

The following Sandia technologies are likely to have applications in quantum computing, and these potential applications should be considered when planning further research involving these technologies.

High Speed Electronics for Quantum Computer Input

The above discussion of device technology suggested that Sandia should focus on quantum computer technologies with large fundamental frequencies. However, applying well-controlled field pulses at these high speeds is likely to put extreme demands on the input electronics controlling the pulses. Furthermore, depending on how well losses can be controlled in the implementation, a considerable amount of power may be needed to create the strong fields that enable these rapid gate operations. Furthermore, it will be necessary to interface the resulting electronic system to a classical digital computer that will generate the series of gate operations needed in order to perform quantum calculations. Sandia's expertise in high speed, high power electronics may allow us to make a major contribution to the specialized input electronics of any future solid-state implementation of a quantum computer.

MEMS Cantilever Technology for Quantum Computer Output

One of the biggest challenges in implementing a practical quantum computer is performing quantum measurements on the qubits in order to determine the output of the device. This is especially challenging for several proposed implementations, which otherwise seem promising, where the qubit is encoded using spin. Techniques to measure the state of a single spin reliably have not previously been developed. One very interesting proposed approach to single spin measurement is the further development of Magnetic Resonance Force Microscopy (MFRM) technique. In this approach, a resonant cantilever is coupled to the spin via the interaction between the magnetic moment of the spin and a magnetic field created by a small magnetic particle mounted on the cantilever. A series of pi-pulses are used to flip the spin at the resonant frequency driving the oscillations of the cantilever. These oscillations can be detected optically, and the initial state of the spin can be determined from the phase of the oscillations. It is believed that over the next several years this approach can be refined to the point where a single electron spin can be measured.

Sandia's expertise in the production of cantilever devices with integrated optical readout using MEMS technology makes this a particularly suitable area of research for Sandia. The proposed work would differ from previous Sandia cantilever designs in that it would require the development of very high compliance cantilevers.[27] Sandia's expertise in the production of nanoscale magnetic clusters might also be applicable to this endeavor.

Photonic Lattice Technology for Controlled Qubit Couplings

It may be possible to suppress spontaneous emission from qubits represented using impurities or quantum dots by enclosing the qubit within a photonic lattice with a photonic bandgap at the fundamental frequency of the representation. This could significantly enhance the coherence times of such a qubit representation and greatly simplify the development of a quantum computer based on such technology. Furthermore, photonic lattices with engineered defects might allow efficient and controlled coupling between a qubit and a propagating photon modes. Difficulties in obtaining such a coupling are currently the chief bottleneck in quantum computer implementations based on cavity quantum electrodynamics.

GaAs-based Quantum Computer Technology

Sandia is a world leader in producing GaAs-based devices incorporating ultrahigh mobility 2-D electron gasses. Gates can be added to these structures to create quantum wires, quantum dots, and quantum point contacts from the original 2-D electron gas. The extreme mobility of these systems allows electrons to propagate long distances without loosing coherence. In principle, this should allow an electronic analog of an optical quantum computer based on the two-rail representation of qubits. However, the Coulomb blockade effect allows a strong interaction between single electrons, and thus the main drawback of an optical approach (weak interactions between photons) is avoided.

Coherent Electron Transport in Quantum Wires

The possibility of realizing a universal set of quantum logic gates using solid-state coherent electron transport in quantum wires has been reported.[28] The basic technique couples two quantum wires with a carefully designed potential barrier allowing for controlled interactions. Numerical analysis has demonstrated the possibility of implementing a one-qubit rotation operation using a coupling barrier and a two-qubit CNOT gate using coulomb interaction. What is remarkable about these possibilities is that they can be realized with a relatively mature technology that is inherently integrable with conventional electronics. This method of quantum computing may be particularly of interest to Sandia since the required development would largely leverage leading capabilities already in place at these labs.

Next Steps for Quantum Computing Input Output

It looks like real-time error-correction will be necessary for quantum computing to work. Each error-bit corrected must begin life as a known value (say 0), and, through a series of state manipulations, be mapped into an error bit. The error bit must be read, so as to remove it from the system, without disturbing the rest of the computation.

It's important to do physical experiments to confirm that these steps are possible. There isn't a lot of doubt about the possibility of adding new bits to a computation. This is one natural way to begin a quantum computation, introducing new bits into the entanglement one at a time. We need to confirm that this can be done in the middle of a computation.

It should also be possible to read out a qubit, collapsing just that portion of the entanglement, leaving the remaining state unmolested. But this deserves experimental confirmation.

Quantum operations above the gate level must be conducted with reversible computation. This is not a physical requirement, but a practical one: each bit computed must be written into a new quantum place, another dimension in the state space. This will exhaust our limited "memory" available. The fix is to "uncompute" the bit when we are done with it, avoiding the cost of erasure.

It's conceivable that temporary bits can be supplied as extra 0s in the state space. They would be used for a while, then uncomputed back to 0, and read out to re-fix their 0-ness for the next use.

The next obvious algorithm steps are small binary or gray-code counters, and a short LFSR-style shift register. Subsequent to these steps, a nonlinear-FSR could be developed offering a wider choice of periods. These are one-to-one devices with no information loss. The challenge is to get them to run as many steps as possible before the states decay, and augmenting the number of steps with error correction. Following this, a very simple 2- or 3-bit adder, with the sum copied elsewhere and then uncomputed in the original bits, will confirm the basic ideas of reversible computation being usable in this environment. Doing actual quantum arithmetic will require many instances of this addition system.

What can we expect to do with small numbers of quantum bits?

We need to explore whether the search algorithms can be subdivided into smaller problems that can be tackled with smaller quantum computers. There's no problem dividing (say) a DES key search into pieces on classical computers, but it's not obvious that the same approach will work in the quantum arena. We can divide up the key space by fixing some of the key bits and letting others be entangled state pairs but then we must carry out an encryption, which, by the nature of the encryption algorithm, will entangle all the bits of the plaintext state. It appears that a key-search engine will need a minimum of bits equal to the block-size, 64 bits for DES, 128 for AES, plus the number of key-bits being searched (as many as possible, up to 56 for DES and 128-256 for AES), plus a

small number of temporary bits. The win-factor, the amount of gain over a conventional computer key search, would be 1/2 the number of quantum key bits.

Appendix A contains the details of a DES encryption with a quantum computer[a]. A DES encryption can be done with 123 qubits, and 42000 gate operations. One gate operation does the C language equivalent of Z ^= (X&Y), i.e. Z is complemented if both X and Y are 1. This algorithm incorporates no overhead for error-correction. It also includes 56 qubits of key for searching.

For the factoring and discrete log problems, its not known if the problem can be subdivided at all. These problems depend on finding periodicities in a virtual array of numbers. Possibly some kind of heterodyning could be used to shift the frequencies of the virtual period. Again, quantum arithmetic on full sized numbers is required, even if the number of quantum parallelism bits is small. It's worth noting that an algorithm for solving the problem "Does N have a divisor between A and B?" is thought to be NP-complete[b,29], so the benefits of solving this problem would stretch beyond cryptography.

Hilbert Space Analog Computing

As a final note, there have been speculative reports of possible advantages of Hilbert Space Computing (HSC) over QC.[30] Both QC and HSC store information physically in a way that can be represented abstractly by a complex unit vector (or, more properly, a ray since an overall phase factor does not have physical meaning) in a 2^N-dimensional Hilbert space. This Hilbert space is defined as a tensor product with factors representing each of the N qubits composed of 2-dimensional Hilbert spaces describing individual qubits. The exponential dimensionality of this space distinguishes HSC and QC (as a subset of HSC) from classical analog computers, whose state is described by a number of parameters that grows only linearly with the size of the system. QC contrasts with classical HSC due to the important quantum attribute often referred to as entanglement. HSC advocates suggest that *classical* physical examples requiring Hilbert space description exist and hence have this exponential property inherent to these spaces.

Suggested QC characteristics such as gate number scaling with qubit number, serial and statistical output porting, no-cloning constraint, decoherence, and low-temperature constraints are often touted as significant road blocks that might be avoided with some examples of HSC. Claims have been made that HSC has advantages regarding the size, complexity and speed of hardware. Also touted are the advantages in parallel output, copying of data, the irrelevance of decoherence, and the potential to operate at elevated temperatures.

It is not clear to us at present whether any or all of these claims are valid. Also not clear are the specific ramifications to computing with nonlocal entanglement inherent in QC and presumably not in HSC. Nevertheless, it is tempting to speculate that decoherence is a major roadblock in QC and that there may be some advantage in developing HSC.

---

[a] Developed by Rich Schroeppel

Conclusion

We conclude that even though mathematical descriptions of "computationally complete" sets of quantum gates are fairly mature, full understanding of these mathematical models yet remains counter-intuitive to most practitioners. Further, progress in this area is limited (1) by lack of physical devices with which to realize Quantum Computing, (2) by lack of control structures through which to supervise quantum operations, and (3) by lack of algorithms for which great gain in efficiency over classical algorithms can be demonstrated.

As more mathematicians, cryptographers, systems engineers, and device physicists and engineers interact regarding these issues, the strange inter-workings of "quantum information processing" will become more intuitive, and progress will be made on algorithms and on quantum gate devices. It is recommended that an ongoing seminar series on advances in Quantum Computing be conducted to keep Sandia's device physicists and information theorists abreast of multi-disciplinary developments in this area.

In particular, Hilbert Space Analog Computing may be an area of rich productivity. Computing in a Hilbert Space is a superset of Quantum Computing, and is realizable (to the level of a few "Hilbert bits") in conventional microelectronics and/or in current programmable logic devices. By attempting to implement quantum-like algorithms in a Hilbert Space Computer, great insights may be gained into the architectures suitable for Quantum Computing and into the design of algorithms that may prove more efficient than classical algorithms.

Appendix A

Computing the DES Block Cipher with a Quantum Computer

Abstract

We describe how to compute the DES block cipher with a 123-qubit quantum computer, using 27264 gate operations.

Introduction

The DES cipher was released in 1975. It was a great achievement for its time, cramming a lot of functionality into one chip. The original specification required hardware implementation explicitly forbidding a software implementation.

DES is a block cipher. The block size is 64 bits, and the key size is 56 bits. The user supplies his 56-bit key, and a 64-bit (8 byte) block of data (the plaintext). The DES function returns another 64-bit block, of encrypted data (the ciphertext). The decryption function takes the same 56-bit key, and the 64-bit ciphertext, and returns the original 64-bit plaintext data.

We assume all operations are perfect and that no errors occur. The algorithm presented below uses two standard types of quantum gates. The two kinds of gates used are represented in the C language. For the first gate, X ^= Y, the bit Y is XORed into the bit X. X is complemented if Y is 1. For the second gate, X ^= (Y&Z), the bit X is complemented if Y and Z are both 1. (Also included: Y&~Z, and ~Y&~Z.) Both of these operations are reversible: doing either a second time undoes the effect of the first time. Both operations have been demonstrated in NMR systems.[24,25]

The algorithm implements DES with 123 qubits and 27264 gate operations. Presumably this would be a subroutine in a Grover's Algorithm search for a DES key. 64 of the qubits are used to represent the plaintext and its intermediate values (middletext) as it is transformed into the ciphertext. (Usually, one known plaintext-ciphertext pair is required to determine a DES key.) 56 of the qubits specify the key, and are read-only, not changed during the algorithm execution. 3 of the qubits are temporary values, and they do the bulk of the computing.

Smaller implementations, which use fewer qubits, can search portions of the keyspace. In this case, some of the key bits are fixed, and no longer need qubits. The minimum number of qubits is 69, for searching a 2-bit portion of the keyspace.

The DES Algorithm

The DES encryption function consists of an initial permutation, 16 key-controlled rounds, and a final permutation. The initial and final permutations simply rearrange the bits of the plaintext and ciphertext. They seem to exist for historical reasons related to clocking the data onto the chip. Cryptographically speaking, they are NOPs, so we'll assume that whatever control computer is operating the quantum machine also applies the initial and final permutations to our plaintext and ciphertext.

We assume we are given a particular 64-bit plaintext value to work with, and need to encrypt it with a 56-bit key. We assume that we have 64 qubits available to hold the plaintext and its successor values as it is modified during the encryption. The plaintext qubits are initialized to the known plaintext value. We assume the key is given as an additional 56 qubits. The key is prepared as 56 entangled bits, so it simultaneously takes on all 2^{56} possible key values. In addition, we need three temporary qubits, called simply A, B, C. They are initialized to 0.

The Round Function

Each round of the encryption uses 48 of the 56 key bits. The details of which bits are used don't matter for this paper, so the lists are omitted. One important point to note is that the values of the key bits are not changed during the algorithm. (This is different in modern ciphers like AES, where key bits are changed during the algorithm; this complicates searching portions of keyspace for these ciphers.)

DES uses the Feistel construction. The 64-bit plaintext is divided into two 32-bit halves, called the left and right halves. In each round, a 32-bit hash is derived from one of the 32-bit halves and 48 of the 56 key bits. The hash is XORed into the other half. The left half is modified in odd rounds, and the right half is modified in even rounds. Since one half is unmodified in each round, the round operation is reversible, which is how the cipher is decrypted.

Sboxes

Computing the hash is the heart of the encryption, and takes most of the work. The 32-bit half is divided into 8 nibbles of 4 bits each. Two other bits are borrowed from adjacent nibbles to make a 6-bit quantity, the extended nibble. This is XORed with 6 key bits, giving an index into a table called an Sbox. The Sbox has 64 entries, each a 4-bit value. There are 8 different Sboxes, used for the 8 extended nibbles. The total output of the 8 Sboxes is 32 bits. Before being XORed into the other half, these 32 output bits are rearranged in a specific pattern called Permutation P, whose purpose is to make sure that the influence of each Sbox output is spread around in the targeted half. Our implementation cost is unaffected by the details of Permutation P.

14	4	13	1	2	15	11	8	3	10	6	12	5	9	0	7
0	15	7	4	14	2	13	1	10	6	12	11	9	5	3	8
4	1	14	8	13	6	2	11	15	12	9	7	3	10	5	0
15	12	8	2	4	9	1	7	5	11	3	14	10	0	6	13

Table 3. Sbox1 values: the 2 borrowed bits select one of the 4 rows, and the 4-bit nibble selects one of the 16 columns. The value is a 4-bit number. The 4 bits are XORed into 4 bits of the target half.

The results of the first extended nibble, the output of Sbox1, are XORed into left half bits L9, L17, L23, and L31.

One property of our scheme is that the Sbox values don't explicitly appear in the quantum state, and no extra qubits are required for them. Instead, the actions of the control computer implicitly define the Sboxes for the quantum engine.

Sbox Lookups

Our plan is to compute each of the 8 Sboxes in turn. The output bits of the Sbox are XORed into the target Half as they are computed. To compute an individual Sbox, our algorithm marches through each cell of the Sbox, asking "is this the active cell?", and, if so, XORing the cell value into the appropriate bits of the target Half.

The marching works by considering each possible combination of values for the six bits of the extended nibble. There are 64 combinations. We start with 111111, and use a gray-code path, modifying one bit position at a time, while covering all the cells. We use a simple trick. The six bits are considered as two groups of three bits, and we walk each group through all 8 possible combinations.

We describe Round 1, Sbox1 as an example. First, six bits of key are XORed into six bits of the right half. We call the result bits HIJKLM. They occupy six qubits in the right half. IJKL correspond to the first nibble; H and M are borrowed bits to make up the extended nibble. These six bits will select a value from Sbox1. We will walk through all possible combinations of the six bits, and for each value we will XOR the appropriate bits (from 0 to 4) into the target half.

Marching Through the Cells of Sbox1

We begin with the temporary bits A, B, C all in the 0 state. We will compute the AND of the bits HIJ and place it in bit C. We complement bit A if bits H and I are 1, A^= H&I. Then we complement bit C if bits A and J are 1, C ^= A&J. After these operations, C = H&I&J. So C will be in the 1 state 1/8 of the time, and in the 0 state 7/8 of the time. (Or, in our quantum entangled world, C is 1/8 1 and 7/8 0.) We need to reset A to 0, so we "uncompute" it by again XORing in H&I. A ^= H&I. Next we similarly compute the AND of KLM in bit B, again using A as a temporary. We don't need to restore A to 0 just yet. If all of HIJKLM are 1, then bits B and C are also 1. The Sbox value is 13 (in the lower right corner of the table), binary 1101. So we should XOR 3 1 bits into particular bit positions in the left half, positions L9, L17, and L31. (We could

also XOR a 0 into bit L23, but there's no need.) We execute the gate operations L11 ^= B&C, L25 ^= B&C, and L14 ^= B&C. This does 1/64 of the work for Sbox1, one cell of the table.

We move on to another cell for the Sbox by adjusting bit B. We execute B ^= K&L. Most of the time this does nothing, but if K and L are both 1, then B is complemented. After the operation, B will be 1 if KLM = 110. This corresponds to row 3 in the last column of the Sbox, with value 0. So we have no bits to XOR into our target for this cell.

We move on to another cell by XORing K&~M into B. Now B is 1 when KLM = 100. This selects the next-to-rightmost entry in row 3 of the Sbox, with value 5, binary 0101. We XOR a 1 into bits L17 and L31 of the left half. L17 ^= B&C, L31 ^=B&C.

We move to another cell of the Sbox by XORing B ^= K&~L. Now B is 1 when KLM = 101, which selects the 6 (binary 0110) in the bottom row of the Sbox; we XOR 1 into L17 and L23. And so it goes. We continue conditionally complementing B, walking through all 8 combinations of the bits KLM.

Changing Qubit C

After the 7th XOR step, B has been through all 8 possible KLM combinations. We execute C ^= H&I to move to a new HIJ value, HIJ=110. We walk B back through the 8 KLM values in reverse, then move to another new HIJ value and so on. (This is just a Gray code for the 64 combinations of HIJKLM.)

Counting the Gate Operations

Overall, we will need 63 steps for the 64 values. Each Sbox cell has an average of 2 bits, so we'll need 128 XORs on left half bits. So our walk through the Sbox needs 191 gate operations, plus 5 to compute initial settings for B and C, and 5 more to un-compute them and restore A, B, C to 0. We need 6 XORs of key bits into right half bits for the Sbox selection, and 6 more to undo the key bit XORs and restore the right half values. So our total cost for one Sbox is 213 gate operations. All 8 Sboxes will cost 1704 gate operations. 16 rounds will come to 27264 gate operations.

If more qubits are available to hold temporary values, the number of gate operations can be reduced somewhat.

Searching Partial Key Spaces

If we are searching a portion of the key space, some of the key bits are assumed to have fixed given values. These bits won't need to have qubits in our "circuit", so we can use fewer than 123 qubits. In the extreme case that we are fixing 54 of the key bits, and only quantum searching 2 key bits, we need only 69 qubits total - 64 for the middletext state (left and right halves), 2 for the quantum portion of the key, and the three temporary bits A, B, C. A small reduction in the number of gate operations is possible.

DISCRIPTION	REQUIREMENT
cipher blocksize	64 bits
key size	56 bits
temporary bits	3 bits
total qubits needed	123 qubits
one Sbox cell	Average 3 gate operations
one Sbox	213 gate operations
one cipher round	1704 gate operations
full 16 round cipher	27264 gate operations

Table 4. Summary of requirements for quantum DES.

3DES

3DES requires three times as many gate operations to do the cipher three times. The two key versions of 3DES will need an additional 56 read-only qubits for the second key. An additional known plaintext-ciphertext pair is required, and another 64 qubits to hold the middletext. Since two plaintexts must be encrypted, the number of gate operations is doubled. For three-key 3DES, a further 56 read-only qubits are needed for the third key, and another 64 qubits for a third known plaintext-ciphertext pair. Since we must encrypt three plaintext blocks, the number of gate operations increases by 50% over the two-key case.

It may be possible to take advantage of a meet-in-the-middle attack to reduce the key search effort. A conventional non-quantum attack can use MITM to reduce the work to 2^{112} encryptions, even though 168 key bits must be found for three-key 3DES. Further research is required.

DISCRIPTION	QUBIT REQUIREMENT	GATE OPERATIONS
two-key version	243	163584
three-key version	363	245376

Table 5. Summary of requirements for 3DES.

AES Prospects

The AES Sbox has 256 values (8 input bits, and 8 output bits), so the average work is 4 XORs x 256 gray code steps. 16 bytes per round gives 16384 gateops/round or 163840 for ten rounds, plus 25% for key-related Sbox operations, 204800. We'll need a few temporary values, maybe 8 or 16, for a total of about 270 qubits. A few more gate operations for the Mix Column step, and the round key xors. (We might do better on the Sbox by using subfields.) One major cost is uncomputing the inputs to the Sbox, which potentially doubles the number of gate operations to roughly 400000. The gray-code walk through the Sbox will need some work, since the number of input variables is 8

rather than 6. We might do it as 5+3, with the bottom level 3-variable walk being the same, and the 5-variable walk being more complicated, but only every eighth step, so the complexity is amortized.

Because AES (and Serpent) change the key during the algorithm execution, searching partial key spaces with some of the key bits fixed doesn't seem to reduce the total number of qubits needed.

Serpent

Serpent is easier, since the Sboxes are smaller, only 4 bits for the input and output. We will still need 128 bits for the middletext, and will need 256 bits for the intermediate key, and a few bits for temporary storage.

Appendix B

QIP History

1926 Born interpretation of the collapse of the wave function
1929 Leo Szilard anticipated Bennet(1982) and invented the concept of a bit of information.[31]
1932-1936 Church-Turing conjecture
1935 Erwin Schrodinger proposes, a now famous, illustration of quantum superposition often referred to as Schrodinger's cat. [31]
1935 Einstein, Podolsky, and N. Rosen, (EPR) publish what they thought might be a fatal flaw of quantum mechanics – nonlocal interactions.[32]
1949 John Tukey introduced the term "bit" [31], see (Szilard,1929) associated entropy $\Delta S = k \ln(2)$ with acquisition of 1 bit.
1950s John Von Neumann CC w/noisy components [31] can use redundancy to work reliably.
1952 G.C. Wick introduces super selection rules.[33]
1955 Anderson, et al., point out that nuclear spins can be used for storing information.[34]
1961 Landauer's principle: Landauer's principle: erasure of information is necessarily a dissipative process [31]
1964 John Bell shows that the predicitions of quantum mechanics cannot be reproduced by any local hidden variable theory [35]
1973 Charles Bennet: any computation can be performed using only reversible steps in principal no dissipation (no power) required
1980 The study of quantum information as a coherent discipline begins to emerge.
1980s That quantum mechanics is computationally difficult leads Feynman to speculate that a quantum computer should be able to perform certain tasks beyond the reach of a conceivable classical computer.
1982 A. Aspect, A. Dailbard, and G. Roger give experimental verification of non-locality in quantum mechanics.[36]
1982 Benioff first to explicitly point out that a quantum system can perform computation [37,38]
1982 Bennet reconciliation of Maxwell's demon with the second law of Thermodynamics.
1982 No cloning principle anticipated by Wooters, Zurek, and Dieks [13,31,37]
1982 Paul Benioff and R Feynman represent two different points of view on QC[31]
1982 Feynman points out the value of quantum computer for quantum simulation emphasizing quantum information storage capacity.[37,39]
1983 RSA algorithm patented by MIT and exclusively licensed to RSA Security Incorporated.
1985 David Deutsch empasized that quantum computers can best realize their computational potential by exploiting massive quantum parallelism [37,5]

David Deutsch coined "quantum parallelism" (Feynman's idea made more explicit)[31]

1991 R. Landauer discusses the implications from information being physical[40]

1993 Bernstein and Vazirani demonstrated that even if NP-complete problems could be solved a classical computer would still not be able to simulate a quantum computer efficiently.[41]

1993 Don Page reports that the average entropy of a subsystem is usually very close to $S \cong N - 2^{-(N+1)}$.[42]

1994 Daniel Simon exhibited the first example of a quantum algorithm that efficiently solves an interesting hard problem. [43]

1994 Peter Shor factoring Algorithm[7]

1995 Peter Shor proposes first example of quantum error correction.[44] No-cloning theorem does not prevent the development of viable quantum error correction schemes.

1995 Ignacio Cirac and Peter Zoller suggest Ion trap QC[20] showed that the quantum XOR (or controlled not) gate $|x,y\rangle \rightarrow |x,y \oplus x\rangle$, can be implemented in an ion trap with altogether 5 laser pulses.

1995 Ion trap XOR demonstrated experimentally by NIST group.[45]

1995 Pellizzari, Gardiner, Cirac, and Zoller suggest Cavity QED QC.[46]

1996 Grover publishes a clever method of searching an unsorted database.[6,47]

1996 Number field sieve developed by Pollard was used to factor RSA-130.[31]

1997 Bennet et al. obtain the result that Grover's algorithm is optimal; no quantum algorithm can solve the database search problem faster than $N^{1/2}$.[48]

1997 Jeff Kimble's group at Caltech persue Cavity QED[49]

1997 Gershenfeld and Chuang and indpendently Cory, Fahmy, and Havel, pointed out that NMR provides a useful implementation of quantum computation.[50]

2000 RSA algorithm, patented by MIT, is released to the public domain.

2001 Ferry et.al. first to propose classical physical systems for HSC[30,51]

2002 O'uchi et. al. first to build a classical physical HSC[30,52]

2003 H.M. Wiseman and J.A. Vaccaro reassess traditional entanglement measures with regard to super selection rules.[53]

2003 F. Verstraete and J.I. Cirac discuss limitations of bipartite operations caused by super selection rules.[54]

References

[1] Charles Bennett and Peter W. Shor, *"Qunatum Information Processing"*, IEEE Transactions on Information theory, vol. **44**, No. 6, p. 2724-2742 (1998).

[2] Andrew Steane, *"Quantum computing"*, Rep. Prog. Phys. **61**, 117–173 (1998).

[3] A. Church, *"A set of Postulates for the Foundation of Logic"*, Annals of Mathematics, second series, 33, 346-366 (1932).

[4] A.M. Turing, *"On Computable Numbers, with an Application to the Entscheidungs problem"*, Proceedings of the London Mathematical Society, Series 2, 42 pp.230-265 (1936-37).

[5] Deutsch, D., *"Quantum Theory, the Church-Turing Principle and the Universal Quantum Computer"*, Proc. R. Soc. Lond. A **400**, 96 (1985).

[6] L.K. Grover, Phys. Rev. Lett. 79, 325, (1997).

[7] P.W. Shor, *"Polynomial-Time Algorithms for Prime Factorization and Discrete Logarithms on a Quantum Computer"*, SIAM J. Computing 26, 1484-1509 (1997).

[8] Michael A. Nielsen and Isaac L. Chuang, *"Quantum Computing and Quantum Information"*, Cambridge University Press, 2000. ISBN 0 521 635039

[9] K. Hoffman and R. Kunze, *"Linear Algebra"*, Prentice Hall 1971.

[10] Issac L. Chuang, *"Quantum Computation: Theory, Practice, and Future Prospects"*, p253-260, editors Averin, Ruggiero, and Silvestrini, *"Macroscopic Quantum Coherence and Quantum Computing"*, Kluwer Academic / Plenum Publishers (2001).

[11] L.K. Grover, *"Quantum Mechanics helps in searching for a needle in a haystack"*.

[12] A.K. Lenstra, and H.W. Lenstra Jr., *"The Development of the Number Field Sieve"*, Berlin: Springer-Verlag, 1993.

[13] Wootters, W. K. & Zurek, W. H., *"A single quantum cannot be cloned"*, *Nature* **299**, 802 (1982).

[14] J. Preskill, ``Fault-Tolerant Quantum Computation", December (1997).

[15] J. Preskill, *Lecture Notes On Quantum Computing.*

[16] A.M. Steane, *"Multiple Particle Interference and Quantum Error Correction"*, Proc. Roy. Soc. Lond. A **452**, 2551-2577 (1996).

[17] A.R. Calderbank, P.W. Shor, *"Good Quantum Error Correcting Codes Exist"*, Phys. Rev. A **54**, 1098-1105 (1996).

[18] J.V. Neumann, *"Probabilistic logics and synthesis of reliable organisms from unreliable components"*, Automata Studies, ed. C.E. Shannon and J. McCarthy, Princeton, Princeton University Press (1956).

[19] P. Gacs, "*Reliable Computation with Cellular Automata*", *Journal of Computer Science*, vol. **32**, no. 15. (1986).

[20] J.I. Cirac and P. Zoller, *"Quantum Computations with Cold Trapped Ions"*, Phys. Rev. Lett. **74**, 4091 (1995).

[21] Sackett, et al., Nature **404**, 256 (2000).

[22] Turchette, et al., Phys. Rev. Lett. **75**, 4710 (1995).

[23] Rauschenbeutel, et al., Phys. Rev. Lett. **83**, 5166 (1999).

[24] Jones, et al., J. Mag. Res. **135**, 353 (1998).

[25] Vandersypen, et al., Nature **414**, 883 (2001).

[26] Michael Brooks [ed.], *"Quantum Computing and Communications"*, Springer-Verlag London Limited (1999).

[27] Thomas Kenny. *"Nanometer-Scale Force Sensing with MEMS Devices"*, IEEE Sensors Journal, Vol. 1, No. 2, 148 (2001).

[28] A. Bertoni, et al, *"Quantum Logic Gates based on Coherent Electron Transport in Quantum Wires"*, Phys. Rev. Lett. **84**, 5912 (2000).

[29] Garey, M. R. & Johnson, D. S., *"Computers and Intractability: a Guide to the Theory of NP-completeness"*, New York: W. H. Freeman & Co. (1979).

[30] Laszlo B. Kish, *"Quantum Computing with Analog Circuits: Hibert Space Computing"*, Smart Structures and Materials 2003: Smart Electronics, MEMS, BioMEMS, and Nanotechnology, Vijay K. Varadanm, Laszlo B. Kish, Editors, Proceedings of SPIE (2003).

[31] Preskill, chapter 1, http://www.theory.caltech.edu/~preskill/ph229

[32] A. Einstein, B. Podolsky and N. Rosen, Phys. Rev. **47**, 777 (1935).

[33] G.C. Wick, et al, Phys. Rev., **88**, 101 (1952).

[34] A.G. Anderson, R.L. Garwin, E.L. Hahn, J.W. Horton. G.L. Tucker and R.M. Walker, J. Appl. Physics, **26**, 1324 (1955).

[35] J.S. Bell, Physics 1, 195 (1964).

[36] A. Aspect, A. Dailbard, and G. Roger, Phys. Rev. Lett. 49, 1804 (1982).

[37] Preskill, J., *"Quantum computing: pro and con", Proc. Royal Soc. London A* **454**, 469-486 (1998).

[38] Benioff, P., *"Quantum mechanical models of Turing machines that dissipate no energy"*, Phys. Rev. Lett. **48**, 1581 (1982).

[39] Feynman, R.P., *"Simulating Physics with Computers", Int. J. Theor. Phys.* **21**, 467 (1982).

[40] R. Landauer, *"Information is Physical"*, Physics Today, Vol. **44**, pp. 23-29 (1991)

[41] Bernstein, E., and Vazirani, U., *"Quantum Complexity Theory", Proc. 25th ACM Symp. on the Theory of Computation,* pp. 11-20. New York: ACM (1993).

[42] Don N. Page, *"Average Entropy of a Subsystem"*, airXiv:gr-qc/9305007 v2, 31 Aug. 1993.

[43] Simon D., *"On the Power of Quantum Computation"*, Proc. 35th Annual Symp. on Foundations of Computer, Science (Los Alamitos, CA: IEEE Computer Society Press) pp 124–34 (1994).

[44] P.W. Shor, *"Scheme for Reducing Decoherence in Quantum Memory"*, Phys. Rev. A **52**, 2493 (1995).

[45]C. Monroe, D.M. Meekhof, B.E. King, W.M. Itano, and D.J. Wineland, *"Demostration of a Fundamental Quantum Logic Gate"*, Phys. Rev. Lett., **75**, 4714 (1995).

[46] T. Pellizari, S.A. Gardiner, J.I. Cirac, and P. Zoller, *"Dechoherence, Continuous Observation, and Quantum Computing: A cavity QED Model"*, Phys. Rev. Lett., **75**, 3788 (1995).

[47] Grover, L.K., *"A Fast Quantum Machanical Algorithm for Database Search", Proc. 28th ACM Symp. on the Theory of Computation,* p. 212 (1996).

[48] Bennett, C. B., Bernstein, E., Brassard, G. and Vazirani, U., *"Strengths and weaknesses of quantum computing"*, Online preprint quant-ph/9701001 (1997).

[49] H.J. Kimble, *"Strong Interactions of Single Atoms and Photons in Cavity QED"*, Physica Scripta. Vol T76, 127-137 (1998).

[50] N.A. Gershenfeld and I.L. Chuang, Science **275**, 350 (1997).

[51] Ferry, D.K., Akis, R, Harris, J., *"Quantum wave processing", Superlattices and Microstructures* **30**, 81 (2001).

[52] O'uchi, S., Fujishima, M., and Hoh, K., *"An 8-qubit Quantum-Circuit Processor", Proc. IEEE Internat. Symp. On Circuits and Systems* (ISCAS, May 26-29, 2002, Pheonix, Arizona) **5**, 209-212 (2002).

[53] H.M. Wiseman and J.A. Vaccaro, Phys. Rev. Lett. **91**, 097902 (2003).

[54] F. Verstraete and J.I. Cirac, Phys. Rev. Lett. **91**, 010404 (2003).

Distribution:

1	MS 9018	Central Technical Files, 8945-1
2	MS 0899	Technical Library, 9616
1	MS 0323	D. Chavez, LDRD Office, 1011
1	MS 0603	Chris Tigges, 1742
1	MS 1415	Normand Modine, 1112
1	MS 0806	Lyndon Pierson, 9336
1	MS 0806	Anand Ganti, 9336
1	MS 0785	Richard Schroeppel, 6514