Int J Theor Phys (2008) 47: 2158–2164
DOI 10.1007/s10773-007-9580-2

Quantum Interference Computation

David Ritz Finkelstein · Giuseppe Castagnoli

Received: 19 December 2006 / Accepted: 18 September 2007 / Published online: 10 October 2007

Abstract Quantum speed-up has been conjectured but not proven for a general computation. Quantum interference computation (QUIC) provides a general speed-up. It is a form of ground-mode computation that reinforces the ground mode in a beam of mostly non-ground modes by quantum superposition. It solves the general Boolean problem in the square root of the number of operations that a classical computer would need for the same problem. For example a typical 80-bit problem would take about 10^{24} cycles (10^7 years at 1 GHz) of classical computation and about 10^{12} cycles (20 minutes at 1 GHz) of QUIC.

Keywords Quantum computation · Quantum interferometric computation · QUIC

1 The Problem

In several special problems, such as prime-number factorization [12], quantum computation reduces the computation time dramatically, from exponential in problem size to polynomial. It has been conjectured that the general problem of solving a system of Boolean equations can be solved by quantum computation—more precisely, by its adiabatic quantum ground mode form—in a time on the order of the square root of the time required by a classical computer [5] or even in polynomial time [2]. This has not been shown. Indeed, there seems to be no proof of any quantum speed-up whatever for a general computation problem, even allowing any kind of quantum computer whatever. We provide one here, using a concept of quantum interference computation of great general applicability put forward by Shiekh [11] see also [7].

We use QUIC to solve in principle the following problem, which is universal for systems of Boolean equations represented in network form [1, 3, 5]:

To bring the general quantum logical network (defined below) from a standard initial mode of its qubits, independent of the network structure, to a ground mode.

D.R. Finkelstein (✉)
School of Physics, Georgia Institute of Technology, Atlanta, GA 30332, USA
e-mail: df4@mail.gatech.edu

G. Castagnoli
Information Technology Division & Quantum Lab, Elsag spa, 16154 Genova, Italy

We argue heuristically that it is possible to solve this general Boolean problem by quantum interference computation (QUIC) in a time on the order of the square root of the time required by a classical computer, where the unit of time is the basic period of the computer. The problem time is still exponential in the problem size, but with half the exponent. This is the square-root conjecture. It leaves open the possibility of a still faster computation, say in polynomial time.

The problem of Boolean computation in general is to find a root of given Boolean equations. In ground-mode computation, the problem of solving the system is reduced to the problem of putting a quantum network that represents the problem into its ground mode. This can be accomplished in several ways.

Some studies of ground-mode computation use incoherent dissipative processes to relax the system to its ground mode, such as annealing, physical or simulated [2, 8]. Local minima in the energy landscape slow this relaxation to the ground mode.

QUIC has no problem with local minima because its maximally sharp quantum preparations of a system leave the qubits of the network completely undetermined. This uses the inherent kinematical non-locality of quantum mechanics.

One can get the system to its ground mode by adiabatic variation of the Hamiltonian [5]. Adiabatic ground-mode computation must slow down at level crossings. QUIC has no problem of level-crossings either.

By a Boolean variable or bit we mean a classical variable x with values $x = 0, 1$. By a qubit we mean a quantum system with a two-dimensional Hilbert space. Depending on constant we may also mean a fixed Hermitian operator x of that system with eigenvalues $x \doteq 0, 1$, or its physical implementation, such as a spin component. To formulate the most general Boolean system it suffices to consider Boolean relations among bits that are expressed in terms of the following special monadic, dyadic, and triadic relations [1, 3, 5]:

The monadic relations are TRUE and FALSE ($[x = 1]$ and $[x = 0]$), typically implemented by a spin component or electric charge with fixed value. We bracket a relation to designate the projection operator on the subspace for which the relation holds as an eigenvalue relation.

The dyadic relation is EQUALS(x, x'), the relation $[x = x']$ between two q bits x, x'. We call it or its implementation a wire. Each wire contributes a frustration energy term $\omega_1[x \neq x']$ to the triode network Hamiltonian. $[x \neq x']$ is a projection operator on a two-dimensional subspace of the four-dimensional (x, x') Hilbert space of the wire. We eliminate the irrelevant zero-point or rest energy for each wire from consideration by measuring all wire energies relative to the ground-mode energy.

The triadic relation is POR(x, y, z), the relation $[x + y + z = 2]$ among the three bits x, y, z, where $+$ is the ordinary addition of operators. We write this also as a partial operation $z = x \dot{+} y$, with the understanding that this is undefined when $x = y = 0$. Such possibly-undefined expressions enrich the language, since there are now three possibilities instead of two for any expression. In addition to being TRUE (1) or FALSE (0), an expression can be UNDEFINED, ∞ in the notation of Peirce, ω or OM in the notation of SETL [10].

We represent the triadic relations by Kochen–Specker triodes [9], the dyadic relations by wires, and the bits by qubits. Such networks are universal for Boolean problems [3]. QUIC can be implemented in many other ways.

We implement the POR relation in concept with the Kochen–Specker triode, a spin-1 system. The three q bits x, y, z of the triode are the squares of the spin operators S_x, S_y, S_z of the spin-1 system:

$$x := (S_x)^2, \quad y := (S_y)^2; \quad z = (S_z)^2; \quad (S_x)^2 + (S_y)^2 + (S_z)^2 \equiv 2. \tag{1}$$

In any mode of the computational basis, exactly one of these three x, y, z vanishes and the other two are 1. Only three of the four possibilities for any two input bits occur; these form a spin triplet. The singlet input mode $(0, 0)$ is excluded. As a result the spin-1 triode implements a partial function, not a function.

A Kochen–Specker triode takes essentially no intrinsic computation time and does not contribute to the network Hamiltonian, which is made up entirely of wire-frustration energies. It is identically satisfied in virtue of angular-momentum kinematics.

The logical problem is then transformed into a physical problem of constructing a triode network and bringing it to a ground mode. One then reads the qubit variables x^i simultaneously to solve the problem.

Networks of such wires and triodes are universal for Boolean problems [4, 5]. That is, any Boolean relation is expressible in terms of the given monadic, dyadic, and triadic relations.

Programming QUIC consists of stating the problem in the logical language of these relations. We assume that the engineering problem of automatically assembling networks that implement such statements, using wires for EQUALS and Kochen–Specker triodes for SUM2, can be solved, and that this implementation will take a time that is linear in the number of logic elements, hence negligible. The resulting network states the problem. We solve it by a transition to a ground mode as follows.

Let N_2 and N_3 be the number of wires and triodes in a triode network. The number of independent computer modes is $D = 3^{N_3}$.

The total energy or (radian) frequency is $H = f\omega_1$ where $f = 0, 1, \ldots, N_2$ is the number of frustrated wires. To find a solution classically one may probe the D modes at random, repeatedly measuring H until the value 0 appears. If there is just one probe per trial, the average number of trials needed to find a mode with frequency $f = 0$ is $\mathrm{O}(D)$. One expects a classical computer with operation time τ_0 to find a root of the system in time $T = \mathrm{O}(D)\tau_0$, exponential in N_2. D is a rough measure of classical problem time. The information $S = \log_2 D$ is a rough measure of problem space, linear in the number of bits, spins, triodes or similar variable elements of the network.

Networks in the ground mode have the minimum possible frequency/energy. If an input beam $|\mathrm{I}\rangle$ of such computers includes a solution, then one can extract the solution by brute-force spectroscopy, sorting the systems according to energy. However the relative probability of that energy is generally exponentially small like $\mathrm{O}(2^{-N_3})$ in a random initial mode, and so the required to be reasonably sure of reaching a solution would still be exponentially large, like $\mathrm{O}(2^{N_2})$. The problem is to build the ground-mode amplitude up to a significantly larger value.

Changing one element at a time gives S a network topology, and a relevant concept of (connected) path. and of descending path. From a fixed origin point there is in general no descending path to the solution, which may be completely surrounded by high-frustration barriers. A state with $f = 1$ may be as many steps from a solution $f = 0$ as a state with $f \gg 1$. One cannot determine that a network is topologically near a solution by measuring any variable of the network, unless we know the solution. Our problem is not to be solved as a labyrinth problem, by a succession of local operations. It must be solved globally, if it is to be solved in a practical time.

2 The QUIC Solution

We measure energy in frequency units so that $\hbar = 1$ and define (radian) periods $T_f = 2\pi/\omega_f = T_1/f$. The highest quantum frequency that can occur is $N_2\omega_1$, when all wires

are frustrated, and this sets the basic cycle period

$$\tau := T_1/N_2$$

for QUIC.

Each wire contributes either 0 to the total energy, when it is satisfied, or ω_1, when it is frustrated. As a result the frequency/energy spectrum of the network relative to the ground-mode energy consists of multiples of one basic frequency ω_1 by integers in a relatively small finite interval $\{0, 1, \ldots, N_2\}$ with $N_2 + 1$ possible energies. The energy 0 characterizes solutions. Thus all the periods in QUIC are multiples of the one basic period τ. QUIC makes use of this harmonic spectrum and the fact that a sum of harmonics is a delta function:

$$\sum_{n=0}^{N-1} \mathrm{e}^{2\pi inm} = N\delta_0^m. \tag{2}$$

When we speak of a computer beam we refer to the ket or mode-vector of a computer, defining a coherent input channel. There may be only one computer in a computer beam. Quantum interference requires only one computer in principle, since a quantum system can interfere with itself.

The QUIC operation has five stages: input, analysis, phase-shift, superposition, and output.

2.1 Input

We will form $B = N_2 + 1$ coherent input computer beams where B is the number of possible energy values of the network. These still need contain only one computer among them. Instead of separating these wave-functions in space, as the term "beam" usually implies, it is convenient to adjoin a beam label b to the q bits of the network. We distinguish the beams by the value of the label b. The beam label b may be encoded in about $\log_2 B$ additional q bits of the network. We still call the terms in the mode-vector with different b-values "beams." By a spectrometer we mean a coherent process with a unitary operator U_S that changes the value of the beam label b from 0 to $b = 0, 1, \ldots, N_2$ according to whether the network energy is $0, \omega_1, \ldots, N_2\omega_1$. This can be done by a fixed unitary operator. In a b basis, we require U_S to have a given unit vector for one of its columns. There is no problem in finding a unitary matrix U_S meeting that condition.

We also write b for the Hermitian operator that multiplies each mode-vector with sharp beam label b by b.

Before the time $t = 0$, QUIC prepares one input beam of network(s) with $b = 0$ in a coherent sum of all 2^N orthogonal computational modes with non-zero amplitudes, such as

$$|\mathrm{I}, t = 0\rangle = 2^{-N/2} \prod_{n\in N} (|b = 0, x^n = 0\rangle + |b = 0, x^n = 1\rangle). \tag{3}$$

Naturally this is an irreversible process. If the q bits x^n are the vertical components of spins, one might apply a strong off-axis magnetic field and let the spins relax to the ground mode in this field through spontaneous photon emission, in a time of order O(1) independent of N_2. This mode vector gives each computational mode $\prod_n |x^n\rangle$ about the same probability $1/D$. The main thing is that the initial mode is independent of the problem, requires no knowledge of the solution, and includes the solution as one of its components, though with small amplitude.

2.2 Analysis

Use a unitary spectrometer operator U_S to split the initial beam with mode-vector $|\mathrm{I}, t=0\rangle$ into a coherent sum of B coherent parts—also called "beams"—

$$S|\mathrm{I}, t=0\rangle = \sum_{b\in B} |b, \mathrm{I}, t=0\rangle \tag{4}$$

in such a way that networks with energy $f\omega_1$ go from $b=0$ to $b=f$, without changing energy. This takes a time that is almost independent of B, hence negligible.

2.3 Phase Shift

Introduce the time-delay τ into each beam b while its mode vector is developing with frequency $b\omega_1$. If this is done by the time t, it carries out the unitary transformation

$$|\mathrm{II}, t\rangle = \mathrm{e}^{-2\pi i b}|\mathrm{I}, t\rangle. \tag{5}$$

In optics such variations in delay time usually result from variations in the optical path length for the different beams.

2.4 Superposition

Combine the B beams coherently into one, with an inverted spectrometer. This process is to change all values of the beam label b from their initial values $0, \ldots, N_2$ to (say) $b=0$ without changing the remaining computational bits x^n. This can be accomplished by a fixed unitary transformation U_b acting in the b factor-space alone, the same for all problems with a given spectrum size B. In a b basis, the matrix for U_b has a top row consisting of the number $B^{-1/2}$ repeated B times, and the rest of U_b is irrelevant to this process and is freely adjustable to satisfy unitarity. The important thing is that one knows exactly which bits to change, from what value and to what value, and need know nothing about the solution to do this. The resulting mode vector is

$$|\mathrm{III}, t\rangle = U_b|\mathrm{II}, t\rangle = \sum_b |\widehat{b}=0\rangle\langle\widehat{b}=b|\mathrm{e}^{-i\Omega\tau b}|\mathrm{I}, t\rangle. \tag{6}$$

There is no loss of beam intensity in this process.

For input with $\Omega \doteq 0$, the ground mode $|0\rangle$, the sum over b is $B|0\rangle$.

For input with $\Omega \doteq \omega_1, \ldots, B\omega_1$, the excited modes, the sum consists of B unit vectors in the complex plane distributed uniformly about a circle. This sum is 0.

Thus the single emerging beam is produced with a pure ground mode.

2.5 Output

Finally we measure the computer q bits at any time $t \equiv 0 \bmod \tau$. The result of the measurement is certainly a solution, with $f=0$, as required.

3 Computation Time and Energy

3.1 Uncertainty Principle

The time-energy uncertainty principle sets basic limits on QUIC. At the beginning the network must be produced with accurate energies, and this takes time. At the end the bits of the network must be measured at an accurate time, and this takes energy. These resources vary, however, not in proportion to 2^{N_2} as in classical computation but in proportion to $\sqrt{2^{N_2}}$. We see this as follows.

If the detection of the computer in the output beam occurs within a time interval ΔT, the frequency Ω of the computer mode-vector is uncertain, with uncertainty $\Delta\Omega \gtrsim 4\pi/\Delta T$. This creates an uncertainty in the phases of the mode-vectors.

QUIC requires strict control of the quantum phase, however. The number of non-solution modes superposed in the initial mode is 2^B, exponentially large in N_2. Nevertheless they must cancel sufficiently in the interference pattern not to overwhelm the ground mode amplitude.

If the solution is unique, the worst possible case for this limitation, the probability of solving the problem by choosing a mode at random is only $p = 1/D \ll 1$. Classically we would have to make $\mathrm{O}(D)$ trials to have a useful probability of solving the problem.

Quantally, suppose random phase errors with standard deviation $\delta\phi$ occur before the final measurement of the QUIC process. Then each of the amplitudes $\mathrm{e}^{-ib\Omega\tau}$ in (6) has an out-of-phase random contribution with a norm whose standard deviation is

$$|\delta \mathrm{e}^{i\phi}|^2 \sim |\mathrm{e}^{i\phi}\delta\phi|^2 = (\delta\phi)^2. \tag{7}$$

Classical error probabilities are linear in the parametric errors, while quantum error probabilities may be merely quadratic in the parameter errors. This familiar non-classical quadratic dependence leads to the QUIC speed-up.

Since the errors in phase are random we sum the probabilities, not the amplitudes. The total probability of an error in the output beam, based on (6), is

$$P_e = \frac{1}{2}\sum (\delta\phi)^2 \|\psi(\omega)\| \sim D(\delta\phi)^2. \tag{8}$$

It is enough for the solution probability to be of the same order as the non-solution probability, since we can run QUIC repeatedly until a solution occurs. For this we must have $P_e = \frac{1}{2}(\delta\phi)^2 D \lesssim 1$. By the uncertainty principle this requires a time

$$T \lesssim \sqrt{D}. \tag{9}$$

A similar discussion applies to the preparation time. This square root is the speed-up that QUIC provides relative to classical computation. This time is still exponential in the number of bits, but with about half the classical exponent. It is mostly needed just to prepare the computer with sharp quantum phases.

4 Discussion

QUIC still takes a time exponential in the problem size, but it halves the exponent.

The actual beam-sorting, phase-shifting, and beam-combining take a merely polynomial time, determined by the time required to shift the relative phases of the many computer-beams. The exponential computation time is mainly consumed by preparing the computer q bits before the spectral analysis and measuring them after the spectral synthesis.

The P computer—a universal quantum computer that solves problems in polynomial time—would be much more powerful than QUIC and is widely sought. A P computer must make more ingenious use of entanglement than QUIC. There is no proof that a P computer does not exist. It has been suggested that nature is a quantum computer, to the extent that even the underlying space-time variables are non-commutative [6], and there is no convincing evidence that nature solves any universal problems in polynomial time. For Boolean logic problems with more than a few hundred binary variables the QUIC time is prohibitive too and the P computer would reign. There is a range of problems that QUIC can solve that a classical computer cannot. A problem that takes the age of the known universe—say 10^{30} operations at a 1 GHz operation frequency—by classical computation would still take QUIC only about 10^15 basic QUIC periods, nominally a week at the GHz rate.

References

1. Aharanov, D., van Dam, W., Kempe, J., Landau, Z., Lloyd, S., Regev, O.: Adiabatic quantum computation is equivalent to standard quantum computation. arXiv:quant-ph/0405098, v1, 18 May 2004
2. Castagnoli, G.: Physica D **120**, 48 (1998)
3. Castagnoli, G., Finkelstein, D.R.: Quantum ground-mode computation with static gates. arXiv:quant-ph/0209169, 2002
4. Castagnoli, G., Finkelstein, D.R.: Quantum ground-state computation with kinematical gates. Proc. R. Soc. Lond. A **459**, 3099–3108 (2003)
5. Farhi, E., Goldstone, J., Gutmann, S., Lapan, J., Lundgren, A., Preda, D.: Science **292**, 472 (2001)
6. Finkelstein, D.: Space-time code. Phys. Rev. **184**, 126–1271 (1969)
7. Long, G.-L.: The general quantum interference computer and the duality principle. arXiv:quant-ph/0512120, 15 December 2005
8. Kadowaki, T.: Study of optimization problems by quantum annealing. arXiv:quant-ph/0205020, 2002
9. Kochen, S., Specker, E.: J. Math. Mech. **17**, 59–87 (1967)
10. Schwartz, J.T., Dewar, R.B.K., Dubinsky, E., Schonberg, E.: Programming with Sets: An Introduction to SETL. Springer, New York (1986)
11. Shiekh, A.: The role of quantum interference in quantum computing. Int. J. Theor. Phys. **45**, 1 (2006)
12. Shor, P.W.: Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. In: Proceedings of the 35th Annual Symposium on Foundations of Computer Science, Santa Fe, NM, 20–22 November 1994. SIAM J. Sci. Statist. Comput. **26**, 1484 (1997). arXiv:quant-ph/9508027