Scheme for Entering Binary Data Into a Quantum Computer

This could be an important step toward making quantum computing practical.

NASA's Jet Propulsion Laboratory, Pasadena, California

A quantum algorithm provides for the encoding of an exponentially large number of classical data bits by use of a smaller (polynomially large) number of quantum bits (qubits). The development of this algorithm was prompted by the need, heretofore not satisfied, for a means of entering real-world binary data into a quantum computer. The data format provided by this algorithm is suitable for subsequent ultrafast quantum processing of the entered data. Potential applications lie in disciplines (e.g., genomics) in which one needs to search for matches between parts of very long sequences of data. For example, the algorithm could be used to encode the N-bit-long human genome in only $\log_2 N$ qubits. The resulting $\log_2 N$-qubit state could then be used for subsequent quantum data processing — for example, to perform rapid comparisons of sequences.

Below are the steps of the algorithm, illustrated with the example of the four-bit string 0111:

1. Specify a correspondence between (a) each classical bit in a string of 2^n such bits and (b) a unique n-bit eigenstate in a set of 2^n such eigenstates. For example, if a classical 2^2-bit string is 0111, then the corresponding four 2-bit eigenstates could be $|00\rangle$, $|01\rangle$, $|10\rangle$, and $|11\rangle$.
2. Construct a superposition, $|\psi\rangle$, of equally weighted quantum states that is peaked at only those eigenstates that correspond to 1s in the classical bit string. In the example of the bit string 0111, the corresponding 2-qubit state would be $|\psi\rangle = 3^{-1/2}(|01\rangle+|10\rangle+|11\rangle)$. In the general case, the superposition would be an entangled state of n qubits that encodes a specific sequence of 2^n classical bits.
3. Compute the unitary transformation needed to obtain the superposition starting from an easy-to-make state (for example, $|00\rangle$). Equivalently, compute a unitary matrix that maps the chosen state (e.g., $|00\rangle$) into the state $|\psi\rangle$. For the classical bit string 0111, the unitary matrix would be

$$\begin{pmatrix} 0 & -1/\sqrt{3} & -1/\sqrt{3} & -1/\sqrt{3} \\ 1/\sqrt{3} & 2/3 & -1/3 & -1/3 \\ 1/\sqrt{3} & -1/3 & 2/3 & -1/3 \\ 1/\sqrt{3} & -1/3 & -1/3 & 2/3 \end{pmatrix}$$

To compute the matrix, first compute $|\psi\rangle\langle\psi|$ (which gives one column of the matrix), then generate the remaining orthonormal vectors for the other columns.
4. By use of software developed previously for this purpose, compute the form of a feasible quantum circuit equivalent to the unitary matrix. The quantum circuit could be implemented in one of several physical embodiments: for example, spin-based, charge-based, optical, or superconducting quantum computer hardware.

This work was done by Colin Williams of Caltech for **NASA's Jet Propulsion Laboratory**. *Further information is contained in a TSP (see page 1).*

In accordance with Public Law 96-517, the contractor has elected to retain title to this invention. Inquiries concerning rights for its commercial use should be addressed to:

Innovative Technology Assets Management
JPL
Mail Stop 202-233
4800 Oak Grove Drive
Pasadena, CA 91109-8099
(818) 354-2240
E-mail: iaoffice@jpl.nasa.gov

Refer to NPO-30209, volume and number of this NASA Tech Briefs *issue, and the page number.*

Encryption for Remote Control via Internet or Intranet

This protocol provides security against control by unauthorized users.

John F. Kennedy Space Center, Florida

A data-communication protocol has been devised to enable secure, reliable remote control of processes and equipment via a collision-based network, while using minimal bandwidth and computation. The network could be the Internet or an intranet. Control is made secure by use of both a password and a dynamic key, which is sent transparently to a remote user by the controlled computer (that is, the computer, located at the site of the equipment or process to be controlled, that exerts direct control over the process). The protocol functions in the presence of network latency, overcomes errors caused by missed dynamic keys, and defeats attempts by unauthorized remote users to gain control. The protocol is not suitable for real-time control, but is well suited for applications in which control latencies up to about 0.5 second are acceptable.

The encryption scheme involves the use of both a dynamic and a private key, without any additional overhead that would degrade performance. The dynamic key is embedded in the equipment- or process-monitor data packets sent out by the controlled computer: in other words, the dynamic key is a subset of the data in each such data packet. The controlled computer maintains a history of the last 3 to 5 data packets for use in decrypting incoming control commands. In addition, the controlled computer records a private key (password) that is given to the remote computer. The encrypted