Quantum Computing

2. Algorithms

C S Vijay and Vishal Gupta

C S Vijay is in the final year of a dual degree program in microelectronics at the Department of Electrical Engineering at IIT Mumbai. His primary interest is in developing algorithms for performance driven optimization of VLSI circuits.

Vishal Gupta was an undergraduate student at the Department of Electrical Engineering, IIT Mumbai. His interests include quantum computing, physics, telephony, and information theory. He is currently involved with a start-up company at IIT Mumbai.

Part 1. Building Blocks of Quantum Computers, *Resonance*, Vol.5, No.9, pp.69-81, 2000.

Introduction

In the first part of this article, we had looked at how quantum physics can be harnessed to make the building blocks of a quantum computer. In this concluding part, we look at algorithms which can exploit the power of this computational device, and some practical difficulties in building such a device.

Quantum Algorithms

We had commented earlier that the state space of quantum systems grows exponentially with a linear increase in physical size, i.e., an n qubit system generates a 2^n dimensional space. Quantum algorithms exploit this fact by making a function act on an n qubit superposition. We first take an n qubit pre-set system (usually a collection of n $|0\rangle$s) and apply the Walsh–Hadamard transformation to each of them. Starting with n $|0\rangle$ s, we obtain

$$H|0\rangle \otimes H|0\rangle .....H|0\rangle \otimes H|0\rangle =$$

$$|00...000\rangle + |00...01\rangle + ... = \frac{1}{\sqrt{2^n}} \sum_{1}^{2^n} |x\rangle,$$

which represents all the numbers from 1 to 2^n. Any function f, when applied to this superposition, along with the customary additional $|0\rangle$ qubit (this field is the output location), would generate an equally weighted superposition of the values the function takes with each input separately. For example:

$$H|0\rangle \otimes H|0\rangle \otimes |0\rangle = \frac{1}{2}(|000\rangle + |010\rangle + |100\rangle + |110\rangle)$$

$$T(|000\rangle+|010\rangle+|100\rangle+|110\rangle) = |000\rangle+|010\rangle+|100\rangle+|111\rangle$$

In effect we get all the entries of the truth table of the function at one stroke. A measurement of the final superposition gives us one of those entries. Thus, making a measurement at this point is thoroughly useless, as you only get a random measurement. Quantum algorithms apply transformations on this superposition that increase the probability of a desired result when a measurement is finally made. These transformations usually hope to achieve one of the following two objectives. They could, as in the case of Shor's factoring algorithm, try to extract some common features of the elements of the superposition, such as the periodicity of a function on them. The other possibility is of amplifying the coefficients of certain values of interest, e.g. those values which satisfy a certain search condition.

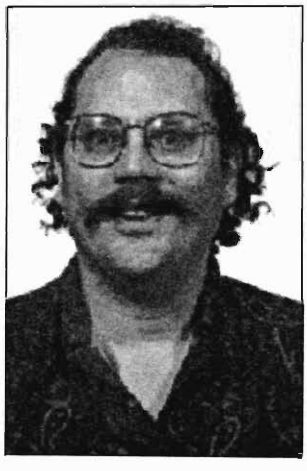
Shor's Factoring Algorithm

Consider the remainders left when non-negative integer powers of 2 are divided by 15. They form the periodic sequence 1, 2, 4, 8 repeated over and over again. This kind of a periodic sequence is obtained for any function of the form

$$f(x) = a^x \mathrm{mod} N,$$

where a and N are relatively prime. Let r be the period of the function. We notice that $f(0) = 1$. Hence $f(r) = 1$, which implies that N divides $a^r - 1$. For even numbers r, if the condition $a^{r/2} \mathrm{mod} N \neq \pm 1$ is satisfied then a factor of N can be obtained as the GCD of $a^{r/2} \pm 1$, and N (since $a^r - 1 = (a^{r/2} - 1)(a^{r/2} + 1)$).

Peter Shor



To start the quantum algorithm we prepare a superposition of numbers from 0 to $M - 1$ using the Walsh–Hadamard transformation on a set of $|0\rangle$ qubits. A good value for M is a power of two between N^2 and $2N^2$. Then we apply the unitary transform corresponding to the function f:

$$\frac{1}{\sqrt{M}}\left(\sum_{x=0}^{M-1} |x\rangle\right)|0\rangle \stackrel{U_f}{\rightarrow} \frac{1}{\sqrt{M}} \sum_{x=0}^{M-1} |x\rangle|f(x)\rangle. \qquad (1)$$

To extract the period, the tool we use is the quantum analogue of the Fast Fourier Transform (called the Quantum Discrete Fourier Transform), which acts on the amplitudes associated with the components of the input state. Hence, we create a state in which the amplitudes have the same period as f. This we do by measuring the second register which contains a superposition of values of f. Let the measured value be $f(l)$ where l is the smallest x for which $f(x) = f(l)$, i.e.

$$f(l) = f(jr + l) \quad \text{for} \quad j = 0, 1, 2, ... \tag{2}$$

Hence the post measurement state is:

$$|\phi_{in}\rangle = \frac{1}{\sqrt{A+1}} \sum_{j=0}^{A} |jr + l\rangle |f(l)\rangle. \tag{3}$$

In the above state the first register contains a uniform superposition of labeled states, the labels having a period r; all the information about the periodicity of f is now in the first register. The Quantum Discrete Fourier Transform acts as:

$$|x\rangle \rightarrow \frac{1}{\sqrt{M}} \sum_{y=0}^{M-1} e^{(\frac{2\pi\iota xy}{M})} |y\rangle, \tag{4}$$

where x and y represent the integer equivalent of $|x\rangle$ and $|y\rangle$ in their binary representation.

The fact that makes this algorithm efficient is that there exists an efficient quantum algorithm for the quantum DFT, and that $f(x)$ can be computed efficiently. Let us first consider the simple case where r divides M exactly. Thus $A = (\frac{m}{r} - 1)$ and hence the first register in (3) can be written as:

$$|\phi_{\text{in}} = \sum_{x=0}^{M-1} (\sqrt{\frac{r}{m}} \delta_{x,jr+l} \, |x\rangle \quad \text{for} \quad j = 0, 1, ...A.$$

We notice that the amplitude function of this state is a periodic function of x. Performing DFT on this superposition, we get:

$$|\phi_{\text{out}}\rangle = \sum_{y=0}^{M-1} \left(\frac{\sqrt{r}}{M}\left[\sum_{j=0}^{A} e^{\left(\frac{2\pi\iota jy(jr)}{M}\right)}\right]e^{\left(\frac{2\pi\iota jy}{M}\right)}\right)|y\rangle. \qquad (5)$$

The term in the square brackets evaluates to $\frac{M}{r}$ or 0 depending on whether or not y is a multiple of $\frac{M}{r}$. Therefore the result of the DFT on $|\phi_{\text{in}}\rangle$ is:

$$|\phi_{\text{out}}\rangle = \frac{1}{\sqrt{r}} \sum_{j=0}^{r-1} e^{\left(\frac{2\pi\iota jy}{M}\right)}|j\frac{M}{r}\rangle, \qquad (6)$$

which is a state whose amplitudes have period $\frac{M}{r}$. Notice that we have inverted the periodicity using the quantum DFT. We now make a measurement on the quantum register containing the superposition of $|j\frac{M}{r}\rangle$. If the result of this measurement be y, we can write:

$$\frac{y}{M} = \frac{\lambda}{r}, \qquad (7)$$

where λ is one of the values of j. We further make the assumption that λ and r do not have any common factors. Then r is the denominator of the reduced fraction $\frac{M}{r}$. We may have to repeat the algorithm if

1. r does not divide M exactly, in which case the measured y would not be a multiple of $\frac{M}{r}$ but could be somewhat away from it.

2. The period r is odd.

3. λ and r have a common factor, in which case we obtain a factor of r only.

4. We get N as N's factor.

Repeating the algorithm log (N) number of times guarantees that the probability of success is greater than 0.5. The success probability can thus be made arbitrarily close to 1, using $O(\log(N))$ resources.

Grover's algorithm, which has been proved optimal for a quantum computer, runs in time proportional to the square root of the size of the search space.

Grover's Search Algorithm

In this section we provide a brief overview of a search algorithm developed by Lov Grover of Bell Labs. Unstructured search is a linear time algorithm on a classical computer. Grover's algorithm, which has been proved optimal for a quantum computer, runs in time proportional to the square root of the size of the search space. Other more efficient algorithms have been found for the cases where the search space has some structure.

The algorithm begins, as in the case of Shor's algorithm, by preparing a register containing all the possible values between 0 and $2^n - 1$, where n is a number such that 2^n is greater than N (the size of the search space). We label the elements of the search space with numbers 1 to N Let f be the function whose value is 1 for the arguments which satisfy the search condition and 0 otherwise. We apply this function to the superposition obtained earlier. Now we have a superposition where all the amplitudes are equal. Grover's algorithm increases the amplitudes of the states satisfying the search condition by repeatedly applying two operations: *selective inversion* (changing the amplitude of states which satisfy the search condition to a negative value of the same magnitude) and *inversion about average* (if any amplitude is a certain amount above the average of all amplitudes, then change it to the same amount below the average, and vice versa). Since, after selective inversion, the amplitude of the required states will be much below the average, and the amplitudes of the other states will be just above the average, inversion about average raises the amplitude of the required states. Thus, making a measurement now yields the right answer with a higher degree of success. For large N, the failure rate is less than $O(\frac{1}{N})$, after $\frac{\pi}{4}\sqrt{N}$ steps.

The *inversion about the average* operation is simply the operation $-HI_0H$, where H is the Hadamard operation,

and I_0 is represented by the matrix:

$$I_0 = \begin{pmatrix} -1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

for $N = 4$.

The *selective inversion* operation is a combination of U_f, the gate which implements f, and the C_{NOT} gate. The input is passed through U_f and the result acts as the controlling qubit of a C_{NOT} gate whose other input is the constant $\frac{1}{\sqrt{2}}(|0\rangle - |1\rangle)$.

In the case of classical computation it is intuitively clear why the lower bound on an unstructured search should be a linear time algorithm. However, though Grover's algorithm has been proven optimal for quantum computation, the fundamental reason behind it is not obvious. This shows that there is a long way to go before we fully appreciate the essence of quantum computation.

Practical Issues

The fundamental obstacle to building a quantum computer is that of decoherence. The interaction of a quantum system with the environment obstructs the unitary evolution of the system and causes dissipation of information, reducing coherence of information. In simpler words, the interaction with the environment is like making a measurement before the computation has been completed. It has been shown that decoherence can't be efficiently removed by simply repeating the computation many times. Technologists are working to reduce the effects of decoherence by employing techniques that allow more computation steps before the time when the effects of decoherence become significant. On the algorithmic side, more success has been achieved, with attempts to come up with error correction techniques by modeling the errors caused by decoherence.

The interaction of a quantum system with the environment obstructs the unitary evolution of the system and causes dissipation of information, reducing coherence of information.

Suggested Reading

[1] S Lloyd, *Scientific American*, October 1995.

[2] N Gershenfeld and I L Chuang, *Scientific American*, June 1998.

[3] D P Divincenzo, *Science*, Vol. 270, Oct 13, 1995.

[4] Online resources at http://www.qubit.org/

[5] Online resources at http://xxx.lanl.gov/archive/quant-ph/

Attempts at physical realization of a quantum computer have used many different quantum systems. Some of these technologies of realizing qubits are:

1. Atomic ions trapped in cryogenic ion traps with lasers manipulating the ion's state (spin and oscillation).

2. Optical cavity with a trapped atom.

3. Nuclear magnetic resonance spectroscopy with liquid samples, where radio frequency beams manipulate the states of coupled nuclear spins in a molecule.

4. Using the tip of a scanning tunneling microscope to set the spin, inside nano-dots (magnetic dots of metals).

5. Ramsey atomic interferometry.

Upto five qubit quantum computers have been constructed using NMR spectroscopy. Grover's search algorithm has been demonstrated on these systems.

Conclusion

The physical realization of a full fledged quantum computer looks to many a distant dream, but some of the more optimistic researchers claim that it is a matter of a few more years. An important issue is what uses such a computer can be put to. Currently, apart from a very few algorithms, not much is known about the programming of such computers. A lot of work needs to be done not only on the physical side, but also on the algorithmic side, if we are to make the promises of this technology come true.

We would like to stress that quantum systems provide an incredibly strong way of computation – however, to exploit this power, we will have to develop algorithms which are inherently 'quantum' in nature; doing so requires a paradigm shift in our theories of computation.

Address for Correspondence
C S Vijay
Hostel 9, IIT, Powai
Mumbai 400 076, India.
Email:csvijay@ee.iitb.ernet.in

Vishal Gupta
Herald Logic Inc
IT Business Incubator
Department of Physics
IIT Powai,
Mumbai 400 076, India.
Email: gupta_vish@yahoo.com