Quantum interferometers as quantum computers

Physica Scripta. Vol. T76, 218–222, 1998

Quantum Interferometers as Quantum Computers

Artur Ekert*

Clarendon Laboratory, University of Oxford, Oxford OX1 3PU, U.K.

Received October 22, 1997; revised version received January 23, 1998; accepted January 23, 1998

Abstract

Quantum computers which use quantum interference of different computational paths to enhance correct outcomes and suppress erroneous outcomes of computations can be viewed as multiparticle interferometers. I discuss this approach to quantum computation and argue that it provides additional insights into the nature of quantum algorithms.

1. From interferometers to computers

Richard Feynman [1] in his talk during the First Conference on the Physics of Computation held at MIT in 1981 observed that it appears to be impossible to simulate a general quantum evolution on a classical probabilistic computer in an *efficient* way. He pointed out that any classical simulation of quantum evolution appears to involve an exponential slowdown in time as compared to the natural evolution since the amount of information required to describe the evolving quantum state in classical terms generally grows exponentially in time. However, instead of viewing this fact as an obstacle, Feynman regarded it as an opportunity. If it requires so much computation to work out what will happen in a complicated multiparticle interference experiment then, he argued, the very act of setting up such an experiment and measuring the outcome is tantamount to performing a complex computation. Indeed, all quantum multiparticle interferometers *are* quantum computers and some interesting computational problems can be based on estimating internal phase shifts in these interferometers. This approach leads to a unified picture of quantum algorithms and has been recently discussed in detail by Cleve *et al.* [2].

Let us start with the textbook example of quantum interference, namely the double-slit experiment, which, in a more modern version, can be rephrased in terms of Mach-Zehnder interferometry (see Fig. 1).

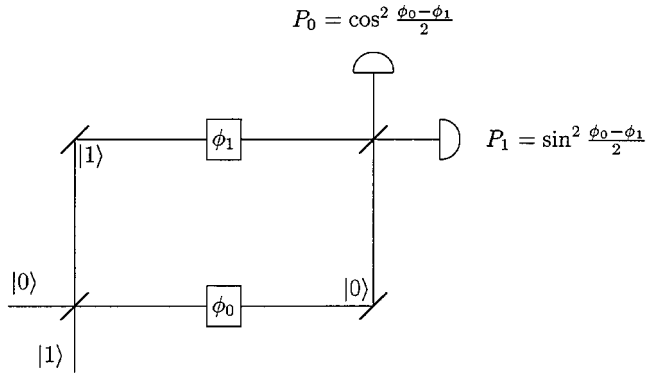

Fig. 1. A Mach-Zehnder interferometer with two phase shifters. The interference pattern depends on the difference between the phase shifts in different arms of the interferometer.

A particle, say a photon, impinges on a beam-splitter (BS1), and, with some probability amplitudes, propagates via two different paths to another beam-splitter (BS2) which directs the particle to one of the two detectors. Along each path between the two beam-splitters, is a phase shifter (PS). If the lower path is labelled as state $|0\rangle$ and the upper one as state $|1\rangle$ then the particle, initially in path $|0\rangle$, undergoes the following sequence of transformations

$$|0\rangle \xrightarrow{\text{BS1}} \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle) \xrightarrow{\text{PS}} \frac{1}{\sqrt{2}}(e^{i\phi_0}|0\rangle + e^{i\phi_1}|1\rangle)$$

$$= e^{i[(\phi_0+\phi_1)/2]} \frac{1}{\sqrt{2}}(e^{i[(\phi_0-\phi_1)/2]}|0\rangle + e^{i[(-\phi_0+\phi_1)/2]}|1\rangle)$$

$$\xrightarrow{\text{BS2}} e^{i[(\phi_1+\phi_2)/2]}(\cos \tfrac{1}{2}(\phi_0 - \phi_1)|0\rangle$$

$$+ i \sin \tfrac{1}{2}(\phi_0 - \phi_1)|1\rangle), \quad (1)$$

where ϕ_0 and ϕ_1 are the settings of the two phase shifters and the action of the beam-splitters is defined as

$$|0\rangle \rightarrow \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle),$$

$$|1\rangle \rightarrow \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle), \quad (2)$$

(we have ignored the phase shift in the reflected beam). The global phase shift $e^{i[(\phi_0+\phi_0)/2]}$ is irrelevant as the interference pattern depends on the difference between the phase shifts in different arms of the interferometer. The phase shifters in the two paths can be tuned to effect any prescribed relative phase shift $\phi = \phi_0 - \phi_1$ and to direct the particle with probabilities $\cos^2(\phi/2)$ and $\sin^2(\phi/2)$ respectively to detectors "0" and "1".

The roles of the three key ingredients in this experiment are clear. The first beam-splitter prepares a superposition of possible paths, the phase shifters modify quantum phases in different paths and the second beam-splitter combines all the paths together erasing all information about which path was actually taken by the particle between the two beam-splitters. As we shall see in the following sections quantum algorithms are not much different. A superposition of computational paths is prepared by the Hadamard (or the Fourier) transform, followed by a quantum function evaluation which effectively introduces phase shifts into different computational paths, followed by the Hadamard or the Fourier transform which acts somewhat in reverse to the first Hadamard/Fourier transform and combines the computational paths together. To see this let us start with rephrasing the Mach-Zehnder interferometry in terms of quantum networks.

* e-mail: ekert@physics.ox.ac.uk

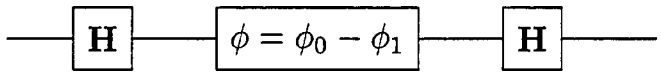

Fig. 2. A quantum networks composed out of three single qubits gates. This network provides a hardware-independent description of any single-particle interference, including the Mach-Zehnder interferometry.

2. Quantum gates and networks

In order to avoid references to experimental details (hardware) let us now describe our Mach-Zehnder interference experiment in more general terms. It is very convenient to view this experiment as a quantum network with three quantum logic gates (elementary unitary transformations) operating on a qubit (a generic two-state system with a prescribed computational basis $\{|0\rangle, |1\rangle\}$). The beam-splitters will now be called the Hadamard gates and the phase shifters the phase shift gate (see Fig. 2).

The Hadamard gate is the single qubit gate **H** performing the unitary transformation known as the Hadamard transform given by (eq. (2))

$$\mathbf{H} = \frac{1}{\sqrt{2}}\begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}, \quad |x\rangle - \boxed{\mathrm{H}} - (-1)^x |x\rangle + |1-x\rangle. \tag{3}$$

The matrix is written in the basis $\{|0\rangle, |1\rangle\}$ and the diagram on the right provides a schematic representation of the gate **H** acting on a qubit in state $|x\rangle$ with $x = 0,1$. Using the same notation we define the phase shift gate ϕ as a single qubit gate such that $|0\rangle \mapsto |0\rangle$ and $|1\rangle \mapsto \mathrm{e}^{\mathrm{i}\phi}|1\rangle$,

$$\phi = \frac{1}{\sqrt{2}}\begin{pmatrix} 1 & 0 \\ 0 & \mathrm{e}^{\mathrm{i}\phi} \end{pmatrix}, \quad |x\rangle - \boxed{\phi} - \mathrm{e}^{\mathrm{i}x\phi}|x\rangle. \tag{4}$$

From now on we will use the language of quantum networks to describe any quantum interference, however, in order to cover the multiparticle case we have to extend the reportoire of our gates to quantum gates which operate on two (or more) qubits.

The conditional phase shift is the two-qubit gate $\mathbf{B}(\phi)$ defined as

$$\mathbf{B}(\phi) = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & \mathrm{e}^{\mathrm{i}\phi} \end{pmatrix}, \quad \begin{matrix} |x\rangle \\ |y\rangle \end{matrix} \!\!\left.\begin{matrix} \\ \\ \end{matrix}\right\} \mathrm{e}^{\mathrm{i}xy\phi}|x\rangle|y\rangle. \tag{5}$$

The matrix is written in the basis $\{|0\rangle|0\rangle, |0\rangle|1\rangle, |1\rangle|0\rangle, |1\rangle|1\rangle\}$ (the diagram on the right shows the structure of the gate). Another important two-qubit gate is the quantum controlled-NOT (or XOR) operation defined as

$$C = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix}, \quad \begin{matrix} |x\rangle \\ |y]\end{matrix} \!\!\left.\begin{matrix} \\ \\ \end{matrix}\right\} |x\rangle \;\; |x \oplus y\rangle \tag{6}$$

where $x, y = 0$ or 1 and $\oplus$ denotes XOR or addition modulo 2. In fact, we do not need to consider any more complicated quantum gates. It has been shown that the controlled-NOT together with the single qubit gates **H** and ϕ are sufficient to construct any quantum network i.e. any unitory transformation operating on n qubits [4, 5].

Let us explain now how the phase shift ϕ can be "computed" with the help of any auxiliary qubit (or a set of qubits) in a prescribed state $|u\rangle$ and some controlled-U transformation where $U|u\rangle = \mathrm{e}^{\mathrm{i}\phi}|u\rangle$ (see Fig. 3).

Here the controlled-U means that the form of U depends on the logical value of the control qubit, for example we can apply the identity transformation to the auxiliary qubits (i.e. do nothing) when the control qubit is in state $|0\rangle$ and apply a prescribed U when the control qubit is in state $|1\rangle$. In our example, shown in Fig. 3, we obtain the following sequence of transformations on the two qubits

$$\begin{aligned} |0\rangle|u\rangle &\xrightarrow{H} \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)|u\rangle \\ &\xrightarrow{c-U} \frac{1}{\sqrt{2}}(|0\rangle + \mathrm{e}^{\mathrm{i}\phi}|1\rangle)|u\rangle \\ &\xrightarrow{H} \left(\cos\frac{\phi}{2}|0\rangle + \mathrm{i}\sin\frac{\phi}{2}|1\rangle\right)|u\rangle. \end{aligned} \tag{7}$$

We note that the state of the auxiliary register $|u\rangle$, being an eigenstate of U, is not altered along this network, but its eigenvalue $\mathrm{e}^{\mathrm{i}\phi}$ is "kicked back" in front of the $|1\rangle$ component in the first qubit. The sequence (7) is the exact simulation of the Mach-Zehnder interferometer and, as it was shown in [2], the kernel of quantum algorithms.

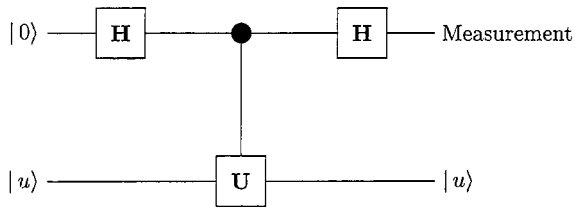

Fig. 3. Phase factors can be introduced into different computational paths via the controlled-U operations. The controlled-U means that the form of U depends on the logical value of the control qubit (the upper qubit). Here, we apply the identity transformation to the auxiliary (lower) qubits (i.e. do nothing) when the control qubit is in state $|0\rangle$ and apply a prescribed U when the control qubit is in state $|1\rangle$. The auxiliary or the target qubit is initially prepared in state $|u\rangle$ which is one of the eigenstates of U.

3. The first quantum algorithm

Since quantum phases in the interferometers can be introduced by some controlled-U operations, it is natural to ask whether effecting these operations can be described as an interesting computational problem.

Suppose an experimentalist, Alice, who runs the Mach-Zehnder interferometer delegates the control of the phase shifters to her colleague, Bob. Bob is allowed to set up any value ϕ and Alice's task is to estimate ϕ. Clearly for general ϕ this involves running the device several times until Alice accumulates enough data to estimate probabilities P_0 and P_1, however, if Bob's promise is that ϕ is set up either at 0 or at π then a single-shot experiment can deliver the conclusive outcome (click in detector "0"/"1" corresponds to $\phi = 0/\phi = \pi$). The first quantum algorithm proposed by David Deutsch in 1985 [3] is related to this effect.

Consider the Boolean functions f that map $\{0, 1\}$ to $\{0, 1\}$. There are exactly four such functions: two constant functions ($f(0) = f(1) = 0$ and $f(0) = f(1) = 1$) and two "balanced" functions ($f(0)$, $f(1) = 1$ and $f(0) = 1$, $f(1) = 0$). Suppose one is allowed to evaluate the function f *only once* and required to deduce from the result whether f is constant or balanced (in other words, whether $f(0)$ and $f(1)$ are the

same or different). Note that we are not asking for the particular values $f(0)$ and $f(1)$ but for the global property of f. Classical intuition tells us that to determine this global property of f, we have to evaluate both $f(0)$ and $f(1)$ anyway, which involves evaluating f twice. We shall see that this is not so in the setting of quantum information, where we can solve Deutsch's problem with a single function evaluation, by employing an algorithm that has the same mathematical structure as the Mach-Zehnder interferometer.

Let us formally define the operation of "evaluating" f in terms of the *f-controlled-NOT* operation on two bits: the first contains the input value and the second contains the output value. If the second bit is initialized to 0, the f-controlled-NOT maps $(x, 0)$ to $(x, f(x))$. This is clearly just a formalization of the operation of computing f. In order to make the operation reversible, the mapping is defined for *all* initial settings of the two bits, taking (x, y) to $(x, y \oplus f(x))$. Note that this operation is similar to the controlled-NOT except that the second bit is negated when $f(x) = 1$, rather than when $x = 1$.

If one is only allowed to perform classically the f-controlled-NOT operation once then it is *impossible* to distinguish between balanced and constant functions in the following sense. Whatever the outcome, both possibilities (balanced and constant) remain for f. However, if quantum mechanical superpositions are allowed then a single evaluation of the f-controlled-NOT suffices to classify f. Our quantum algorithm that accomplishes this is best represented as the quantum network shown in Fig. 4, where the middle operation is the f-controlled-NOT, which can be defined as:

$$|x\rangle|y\rangle \xrightarrow{f-c-N} |x\rangle|y \oplus f(x)\rangle. \tag{8}$$

The initial state of the qubits in the quantum network is $|0\rangle(|0\rangle - |1\rangle)$ (apart from a normalization factor, which will be omitted in the following). After the first Hadamard transform, the state of the two qubits has the form $(|0\rangle + |1\rangle)(|0\rangle - |1\rangle)$. To determine the effect of the f-controlled-NOT on this state, first note that, for each $x \in \{0, 1\}$.

$$\begin{aligned}|x\rangle(|0\rangle - |1\rangle) &\xrightarrow{f-c-N} |x\rangle(|0 \oplus f(x)\rangle - |1 \oplus f(x)\rangle) \\ &= (-1)^{f(x)}|x\rangle(|0\rangle - |1\rangle).\end{aligned} \tag{9}$$

Therefore, the state after the f-controlled-NOT is

$$((-1)^{f(0)}|0\rangle + (-1)^{f(1)}|1\rangle)(|0\rangle - |1\rangle). \tag{10}$$

That is, for each x, the $|x\rangle$ term acquires a phase factor of $(-1)^{f(x)}$, which corresponds to the eigenvalue of the state of the auxiliary qubit under the action of the operator that sends $|y\rangle$ to $|y \oplus f(x)\rangle$.

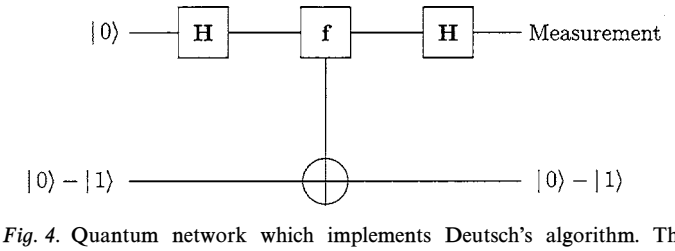

Fig. 4. Quantum network which implements Deutsch's algorithm. The middle gate is the f-controlled-NOT which evaluates one of the four functions f: $\{0, 1\} \mapsto \{0, 1\}$.

This state can also be written as

$$(-1)^{f(0)}(|0\rangle + (-1)^{f(0) \oplus f(1)}|1\rangle)(|0\rangle - |1\rangle), \tag{11}$$

which, after applying the second Hadamard transform to the first qubit, becomes

$$(-1)^{f(0)}|f(0) \oplus f(1)\rangle(|0\rangle - |1\rangle). \tag{12}$$

Therefore, the first qubit is finally in state $|0\rangle$ if the function f is constant and in state $|1\rangle$ if the function is balanced, and a measurement of this qubit distinguishes these cases with certainty.

Deutsch's result laid the foundation for the new field of quantum computation, and was followed by several other quantum algorithms for various problems. They all can be viewed as the phase estimation in some multiparticle interferometer. Let us illustrate this taking as an example the Shor quantum algorithm for efficient factorization (for a comprehensive discussion of Shor's factoring algorithm see [6, 7]). This time "the interferometer" is much more complicated and the phase estimation is performed by the quantum Fourier transform (rather than the Hadamard transform).

4. Quantum Fourier transform

The discrete Fourier transform is a unitary transformation of a s-dimensional vector $\{f(0), f(1), f(2), \ldots, f(s-1)\}$ defined by:

$$\tilde{f}(y) = \frac{1}{\sqrt{s}} \sum_{x=0}^{s-1} e^{2\pi i x y/s} f(x), \tag{13}$$

where $f(x)$ and $\tilde{f}(y)$ are in general complex numbers. It can also be represented as a unitary matrix

$$\frac{1}{\sqrt{s}} \begin{pmatrix} 1 & 1 & 1 & \cdots & 1 \\ 1 & \omega & \omega^2 & \cdots & \omega^{(s-1)} \\ 1 & \omega^2 & \omega^4 & \cdots & \omega^{2(s-1)} \\ 1 & \omega^3 & \omega^6 & \cdots & \omega^{3(s-1)} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & \omega^{(s-1)} & \omega^{2(s-1)} & \cdots & \omega^{(s-1)^2} \end{pmatrix}, \tag{14}$$

where $\omega = \exp(2\pi i/s)$ is the sth root of unity. In the following we assume that s is a power of 2, i.e., $s = 2^n$ for some n; this is a natural choice when binary coding is used.

The quantum version of the discrete Fourier transform is a unitary transformation which can be written in a chosen computational basis $\{|0\rangle, |1\rangle, \ldots, |2^n - 1\rangle\}$ as [8, 9],

$$\mathrm{QFT}: |x\rangle \mapsto \frac{1}{\sqrt{s}} \sum_{y=0}^{s-1} \exp(2\pi i x y/s)|y\rangle. \tag{15}$$

More generally, QFT effects the discrete Fourier transform of the input amplitudes. If

$$\mathrm{QFT}: \sum_x f(x)|x\rangle \mapsto \sum_y \tilde{f}(y)|y\rangle, \tag{16}$$

then the coefficients $\tilde{f}(y)$ are the discrete Fourier transforms of $f(x)$'s.

Let x be represented in binary as $x_0 \ldots x_n \in \{0, 1\}^n$, where $x = \sum_{i=1}^{n-1} x_i 2^i$ (and similarly for y). For the purpose of the phase estimation it is interesting to note that the state $\sum_{y=0}^{s-1} \exp(2\pi i x y/s)|y\rangle$ is unentangled, and can in fact be factorized as

$$(|0\rangle + e^{i\phi_x}|1\rangle)(|0\rangle + e^{i2\phi_x}|1\rangle) \cdots (|0\rangle + e^{i2^{n-1}\phi_x}|1\rangle), \tag{17}$$

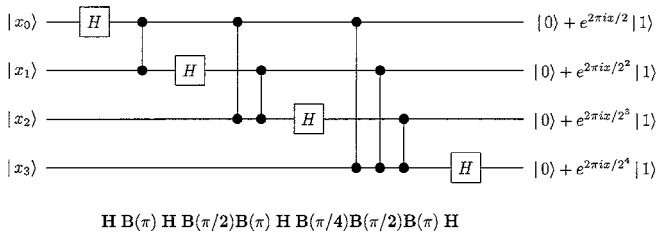

Fig. 5. The quantum Fourier transform (QFT) network operating on four qubits. If the input state represents number $x = \Sigma_k 2^k x_k$ the output state of each qubit is of the form $|0\rangle + e^{i2k\phi_x}|1\rangle$, where $\phi_x = 2\pi/2^n$ and $k = 0, 1, 2 \ldots n-1$. N.B. there are three different types of the $B(\phi)$ gate in the network above: $B(\pi)$, $B(\pi/2)$ and $B(\pi/4)$.

where $\phi_x = 2\pi x/2^n$. Thus if we can prepare a quantum state of the form (17) then by applying the reverse of the QFT to this state we obtain x and consequently the value of ϕ_x. This can be easily seen from the QFT network in Fig. 5. The network is constructed using only two types of quantum gates, these are: the Hadamard gate **H** and the conditional phase shift $\mathbf{B}(\phi)$. The input qubits are initially in some state $|x\rangle = |x_0\rangle|x_1\rangle|x_2\rangle|x_3\rangle$ where $x_0 x_1 x_1 x_3$ is the binary representation of x; $x = \sum_{i=0}^{3} x_i 2^i$.

A general case of n qubits requires a trivial extension of the network following the same sequence pattern of gates **H** and **B**.

Let us describe now how quantum the states of the form (17) can be generated as the result of some quantum computations.

5. Computing phase shifts

Suppose that U is any unitary transformation on m qubits and $|\Phi\rangle$ is an eigenvector of U with eigenvalue $e^{i\phi}$ and consider the following scenario. We do not explicitly know U or $|\Phi\rangle$ or $e^{i\phi}$, but instead are given devices that perform controlled-U, controlled-U^{2^1}, controlled-U^{2^2} and so on until we reach controlled-$U^{2^{n-1}}$. Also, assume that we are given a single preparation of the state $|\Phi\rangle$. From this, our goal is to obtain an n-bit estimator of ϕ.

This can be solved as follows. First, apply the network of Fig. 6. This network produces the state

$$(|0\rangle + e^{i2^{n-1}\phi}|1\rangle)(|0\rangle + e^{i2^{n-2}\phi}|1\rangle)\cdots(|0\rangle + e^{i\phi}|1\rangle). \quad (18)$$

As noted in the last section, in the special case where $\phi = 2\pi x/2^n$, the state $|x_0 \ldots x_{n-1}\rangle$ (and hence ϕ) can be obtained by just applying the inverse of the QFT (which is the network of Fig. 5 in the backwards direction). If x is an n-bit number this will produce the state $|x_0 \ldots x_{n-1}\rangle$ exactly (and hence the exact value ϕ).

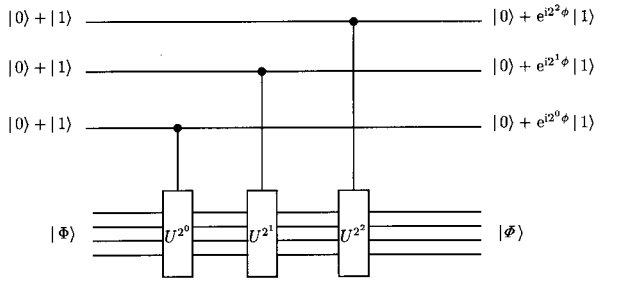

Fig. 6. The network which computes phase shifts in Shor's algorithms; it also implements the modular exponentiation function via repeated squarings.

However, ϕ is not in general a fraction of a power of two (and may not even be a rational number). For such a ϕ, it turns out that applying the inverse of the QFT produces the best n-bit approximation of ϕ with probability at least $4/\pi^2 \approx 0.405$ [2].

6. Quantum factoring

Shor's quantum factoring of an integer N is based on calculating the period of the function $a^x \bmod N$ for a randomly selected integer a between 0 and N. It turns out that for increasing powers of a, the remainders form a repeating sequence with a period which we denote r. Once r is known the factors of N are obtained by calculating the greatest common divisor of N and $a^{r/2} \pm 1$.

Suppose we want to factor 15 using this method. Let $a = 11$. For increasing x the function $11^x \bmod 15$ forms a repeating sequence 1, 11, 1, 11, 1, 11, ... The period is $r = 2$, and $a^{r/2} \bmod 15 = 11$. Then we take the greatest common divisor of 10 and 15, and of 12 and 15 which gives us respectively 5 and 3, the two factors of 15. Classically calculating r is at least as difficult as trying to factor N; the execution time of calculations grows exponentially with the number of digits in N. Quantum computers can find r very efficiently.

In order to formulate Shor's algorithm in terms of the phase estimation let us apply the construction from the last section taking

$$|\Phi\rangle = \sum_{j=0}^{r-1} e^{(-2\pi i j)/r}|a^j \bmod N\rangle. \quad (19)$$

Such a state is not at all trivial to fabricate; we shall see how this difficulty is circumvented later. Consider the unitary transformation U that maps $|x\rangle$ to $|ax \bmod N\rangle$. Note that $|\Phi\rangle$ is an eigenvector of U with eigenvalue $e^{2\pi i(1/r)}$. Also, for any j, it is possible to implement efficiently a controlled-U^{2^j} gate. Thus, using the state $|\Phi\rangle$ and the implementation of controlled-U^{2^j} gates, we can directly apply the method of the last section to efficiently obtain an estimator of $1/r$.

Let us notice in passing that the sequence of controlled-U^{2^j} operations is equivalent to the implementation (via repeated squarings) of the modular exponentiation function in Shor's algorithm,

$$a^x = a^{2^0 x_0} \cdot a^{2^1 x_1} \cdot \cdots a^{2^{n-1} x_{n-1}}, \quad (20)$$

where $x_0, x_1 \ldots$ are the binary digits of x [10].

The problem with the above method is that we are aware of no straightforward efficient method to prepare state $|\Phi\rangle$, however, let us notice that any state $|\Phi_k\rangle$ of the form

$$|\Phi_k\rangle = \sum_{j=0}^{r-1} e^{-[(2\pi i k j)/r]}|a^j \bmod N\rangle, \quad (21)$$

where k is from $\{1, \ldots, r\}$ would also do the job. For each $k \in \{1, \ldots, r\}$, the eigenvalue of state $|\Phi_k\rangle$ is $e^{2\pi i(k/r)}$, and we can again use the technique from the last section to efficiently determine k/r and if k and r happen to be coprime then this yields r. Now the key observation is that

$$|1\rangle = \sum_{k=1}^{r} |\Phi_k\rangle, \quad (22)$$

and $|1\rangle$ *is* an easy state to prepare.

If we substituted $|1\rangle$ in place of $|\Phi\rangle$ in the last section

then effectively we would be estimating one of the r, randomly chosen, eigenvalues $e^{2\pi i(k/r)}$. This demonstrates that Shor's algorithm, in effect, estimates the eigenvalue corresponding to an eigenstate of the operation U that maps $|x\rangle$ to $|ax \bmod N\rangle$.

7. Conditional quantum dynamics

Quantum gates and quantum networks provide a very convenient language for building any quantum computer or (which is basically the same) quantum multiparticle interferometer. But can we build quantum logic gates?

Single qubit quantum gates are regarded as relatively easy to implement. For example, a typical quantum optical realization uses atoms as qubits and controls their states with laser light pulses of carefully selected frequency, intensity and duration; any prescribed superposition of two selected atomic states can be prepared this way. Two-qubit gates are much more difficult to build.

In order to implement two-qubit quantum logic gates it is sufficient, from the experimental point of view, to induce a conditional dynamics of physical bits, i.e. to perform a unitary transformation on one physical subsystem conditioned upon the quantum state of another subsystem,

$$U = |0\rangle\langle 0| \oplus U_0 + |1\rangle\langle 1| \otimes U_1 + \cdots + |k\rangle\langle k| \otimes U_k, \tag{23}$$

where the projectors refer to quantum states of the control subsystem and the unitary operations U_i are performed on the target subsystem [4]. The simplest non-trivial operation of this sort is probably a conditional phase shift such as $\mathbf{B}(\phi)$ which we used to implement the quantum Fourier transform and the quantum controlled-NOT (or XOR) gate.

Let us illustrate the notion of the conditional quantum dynamics with a simple example (see Fig. 7). Consider two qubits, e.g. two spins, atoms, single-electron quantum dots, which are coupled via $\sigma_z^{(1)}\sigma_z^{(2)}$ interaction (e.g. a dipole–dipole interaction). The first qubit with the resonant frequency ω_1 will act as the control qubit and the second one, with the resonant frequency ω_2, as the target qubit. Due to the coupling V the resonant frequency for transitions between the states $|0\rangle$ and $|1\rangle$ of one qubit *depends on the neighbour's state*. The resonant frequency for the first qubit becomes $\omega_1 \pm \Omega$ depending on whether the second qubit is in state $|0\rangle$ or $|1\rangle$. Similarly the second qubit's resonant frequency becomes $\omega_2 \pm \Omega$, depending on the state of the first qubit. Thus a π-pulse at frequency $\omega_2 + \Omega$ causes the transition $|0\rangle \leftrightarrow |1\rangle$ in the second qubit only if the first qubit is in $|1\rangle$ state. This way we can implement the quantum controlled-NOT gate.

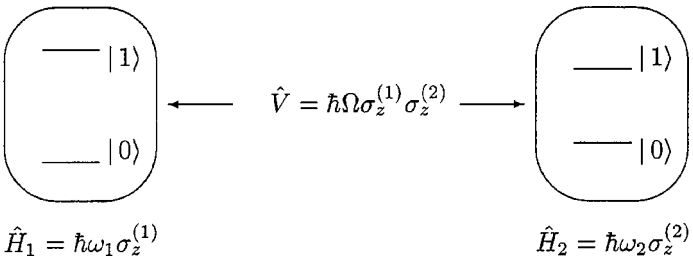

Fig. 7. The control qubit of resonant frequency ω_1 interacts via $\hat{V}$ with the target qubit of resonant frequency ω_2. Due to the interaction the two resonant frequencies are modified and the combined system of the two qubits has four different resonant frequencies $\omega_1 \pm \Omega$ and $\omega_2 \pm \Omega$. A π-pulse at frequency $\omega_2 + \Omega$ causes the transition $|0\rangle \leftrightarrow |1\rangle$ in the second qubit only if the first qubit is in state $|1\rangle$. This is one possible realization of the quantum controlled-NOT gate.

Thus in principle we know how to build a quantum computer; we can start with simple quantum logic gates and try to integrate them together into quantum networks. However if we keep on putting quantum gates together into networks we will quickly run into some serious practical problems. The more interacting qubits are involved the harder it tends to be to engineer the interaction that would display the quantum interference. Apart from the technical difficulties of working at single-atom and single-photon scales, one of the most important problems is that of preventing the surrounding environment from being affected by the interactions with the computer. The more components the more likely it is that quantum computation will spread outside the computational unit and will irreversibly dissipate useful information to the environment. In other words the environment can learm about which computational path was taken in the multiparticle interferometer and this "welcher Weg" information can destroy the interference and the power of quantum computing. However, current developments in the experimental quantum computing together with a set of new tricks to protect quantum interference give some hope that complex multiparticle interferometers will be built in a not too distant future.

8. Concluding remarks

Multiparticle interferometers can be viewed as quantum computers and any quantum algorithm follows the typical multiparticle interferometry sequence of operations. This approach sems to provide an additional insight into the nature of quantum computation and, I believe, will help to unify all quantum algorithms and relate them into different instances of quantum phase estimation.

Acknowledgements

I am greatly indebted to Richard Cleve, David DiVincenzo, Chiara Macchiavello and Michele Mosca for our endless discussions about the connections between quantum interferometry and quantum computation. The author is supported by the Royal Society, London. This work was supported in part by the European TMR Research Network ERP-4061PL95-1412, Hewlett-Packard and Elsag-Bailey.

References

1. Feynman, R., Int. J. Theor. Phys. **21**, 467 (1982).
2. Cleve, R., Ekert, A., Macchiavello, C. and Mosca, M., "Quantum Algorithms Revisited", Proc. R. Soc. Lond. **A454**, 339 (1998).
3. Deutsch, D., Proc. R. Soc. Lond. **A400**, 97 (1985).
4. Barenco, A., Deutsch, D., Ekert, A. and Jozsa, R., Phys. Rev. Lett. **74**, 4083 (1995).
5. Barenco, A. *et al.*, Phys. Rev. **A52**, 3457 (1995).
6. Shor, P. W. (1994) in Proceedings of the 35th Annual Symposium on the Foundations of Computer Science (edited by S. Goldwasser) (IEEE Computer Society Press, Loss Alamitos, CA), p. 124; Expanded version of this paper is available at LANL quant-ph archive.
7. Ekert, A. and Jozsa, R., Rev. Mod. Phys. **68**, 733 (1996).
8. Coppersmith, D., An Approximate Fourier Transform Useful in Quantum Factoring, IBM Research Report No. RC19642 (1994).
9. Barenco, A., Ekert, A., Suominen, K.-A. and Törma P., Phys. Rev. **A54**, 139 (1996).
10. Vedral, V., Barenco, A. and Ekert, A., Phys. Rev. **A54**, 147 (1996).