Supplementary material, Kolkata 2010

From Interferometers to Quantum Computers

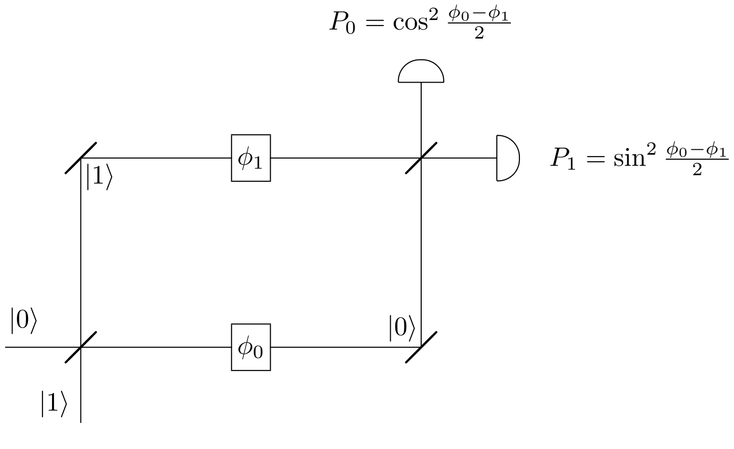
Artur Ekert

Mathematical Institute, University of Oxford
Centre for Quantum Technologies, National University of Singapore

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 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 phases in these interferometers [2].

1 Interferometers

Let us start with the textbook example of quantum interference, namely a Mach-Zehnder interferometer.


$P_0 = \cos^2 \frac{\phi_0-\phi_1}{2}$
$P_1 = \sin^2 \frac{\phi_0-\phi_1}{2}$
ϕ_1
$|1\rangle$
$|0\rangle$
ϕ_0
$|0\rangle$
$|1\rangle$

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

$$\begin{aligned}|\,0\rangle \quad &\overset{\text{BS1}}{\longrightarrow} \quad \frac{1}{\sqrt{2}}(|\,0\rangle + |\,1\rangle) \\ &\overset{\text{PS}}{\longrightarrow} \quad \frac{1}{\sqrt{2}}(e^{i\phi_0}\,|\,0\rangle + e^{i\phi_1}\,|\,1\rangle) = e^{i\frac{\phi_0+\phi_1}{2}}\frac{1}{\sqrt{2}}(e^{i\frac{\phi_0-\phi_1}{2}}\,|\,0\rangle + e^{-i\frac{\phi_0-\phi_1}{2}}\,|\,1\rangle) \\ &\overset{\text{BS2}}{\longrightarrow} \quad e^{i\frac{\phi_1+\phi_2}{2}}(\cos\tfrac{1}{2}(\phi_0-\phi_1)\,|\,0\rangle + i\sin\tfrac{1}{2}(\phi_0-\phi_1)\,|\,1\rangle), \end{aligned} \tag{1}$$

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

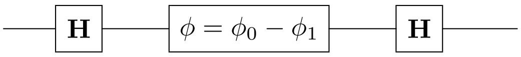
$$\begin{aligned}|\,0\rangle \quad &\longrightarrow \quad \tfrac{1}{\sqrt{2}}(|\,0\rangle + |\,1\rangle) \\ |\,1\rangle \quad &\longrightarrow \quad \tfrac{1}{\sqrt{2}}(|\,0\rangle - |\,1\rangle)\end{aligned} \tag{2}$$

(and extends by linearity to states of the form $\alpha\,|\,0\rangle + \beta\,|\,1\rangle$). Here, we have ignored the $e^{i\frac{\phi_0+\phi_0}{2}}$ phase shift in the reflected beam, which is irrelevant because the interference pattern depends only 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 $\cos^2\left(\frac{\phi}{2}\right)$ and $\sin^2\left(\frac{\phi}{2}\right)$ 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. As we shall see, most quantum algorithms follow this interferometry paradigm: 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 by rephrasing Mach-Zehnder interferometry in terms of quantum networks.

2 Quantum gates & networks

In order to avoid references to specific technological choices (hardware), let us now describe our Mach-Zehnder interference experiment in more abstract terms. It is 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 be now called the Hadamard gates and the phase shifters the phase shift gates.



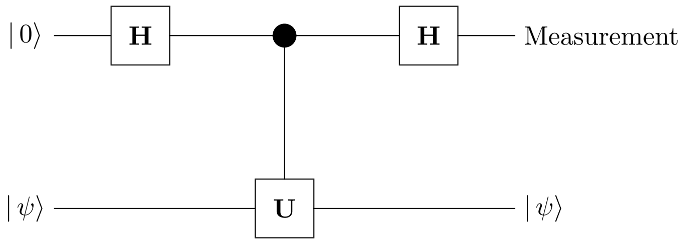
The Hadamard gate is the single qubit gate **H** performing the unitary transformation known as the Hadamard transform given by (Eq. 2)

$$\mathbf{H} = \frac{1}{\sqrt{2}}\begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} \qquad\qquad |\,x\rangle \;-\!\boxed{\mathbf{H}}\!-\; |\,0\rangle + (-1)^x\,|\,1\rangle \tag{3}$$

The matrix is written in the basis $\{|\,0\rangle\,,|\,1\rangle\}$ and the diagram on the right provides a schematic representation of the gate $\mathbf{H}$ 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 $|\,1\rangle \mapsto e^{i\phi}\,|\,1\rangle$,

$$\phi = \begin{pmatrix} 1 & 0 \\ 0 & e^{i\phi} \end{pmatrix} \qquad\qquad |\,x\rangle \;-\!\boxed{\phi}\!-\; e^{ix\phi}\,|\,x\rangle\,\cdot \tag{4}$$

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



Here the controlled-U is a transformation involving two qubits, where the form of U applied to the auxiliary or target qubit 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 above, we obtain the following sequence of transformations on the two qubits

$$\begin{aligned} |\,0\rangle\,|\,\psi\rangle \xrightarrow{H} \tfrac{1}{\sqrt{2}}(|\,0\rangle + |\,1\rangle)\,|\,\psi\rangle \;\; &\overset{c-U}{\longrightarrow} \;\; \tfrac{1}{\sqrt{2}}(|\,0\rangle + e^{i\phi}\,|\,1\rangle)\,|\,\psi\rangle \\ &\overset{H}{\longrightarrow} \;\; e^{(i\frac{\phi}{2})}(\cos\tfrac{\phi}{2}\,|\,0\rangle + i\sin\tfrac{\phi}{2}\,|\,1\rangle)\,|\,\psi\rangle\,. \end{aligned} \tag{5}$$

We note that the state of the auxiliary register $|\,\psi\rangle$, being an eigenstate of U, is not altered along this network, but its eigenvalue $e^{i\phi}$ is "kicked back" in front of the $|\,1\rangle$ component in the first qubit. The sequence (5) is equivalent to the steps of the Mach-Zehnder interferometer (1) and, as was shown in [2], the kernel of most known quantum algorithms.

3 Phases via function evaluations

Since quantum phases in 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 $\phi = \phi_0 - \phi_1$ 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 promises to set up ϕ either at 0 or at π then a single-shot experiment can deliver the conclusive outcome (click in detector "0" corresponds to $\phi = 0$ and in detector "1" corresponds to $\phi = \pi$). The first quantum algorithm proposed by David Deutsch in 1985 [3] is related to this effect.

We have seen in the previous section that a controlled-U transformation can be used to produce a particular phase shift on the control qubit corresponding to its eigenvalue on the auxiliary qubit. If two eigenvalues of the controlled-U transformation lead to different orthogonal states in the control qubit, a single measurement on this qubit will suffice to distinguish the two cases. For example, 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 ($f(0) = 0, f(1) = 1$ and $f(0) = 1, f(1) = 0$). It turns out that it is possible to construct a controlled function evaluation such that two possible eigenvalues are produced which may be used to determine whether the function is constant or balanced. This is done in the following way.

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 initialised 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 $(x, y \oplus f(x))$, where $\oplus$ denotes addition modulo two.

A single evaluation of the f-controlled-NOT on quantum superpositions suffices to classify f as constant or balanced. This is the real advantage of the quantum method over the classical. Classically if the f-controlled-NOT operation may be performed only once then it is *impossible* to distinguish between balanced and constant functions. Whatever the outcome, both possibilities (balanced and constant) remain for f. This corresponds to our classical intuition about the problem since it involves determining not particular values of $f(0)$ and $f(1)$, but a global property of f. Classically to determine this global property of f, we have to evaluate both $f(0)$ and $f(1)$, which involves evaluating f twice.

Deutsch's quantum algorithm has the same mathematical structure as the Mach-Zehnder interferometer, with the two phase settings $\phi = 0, \pi$. It is best represented as the quantum network shown in Fig. 1, where the middle operation is the f-controlled-NOT, which can be defined as:

$$|\,x\rangle\,|\,y\rangle \xrightarrow{f-c-N} |\,x\rangle\,|\,y \oplus f(x)\rangle \ . \tag{6}$$

The initial state of the qubits in the quantum network is $|\,0\rangle\,(|\,0\rangle - |\,1\rangle)$ (apart from a normalization

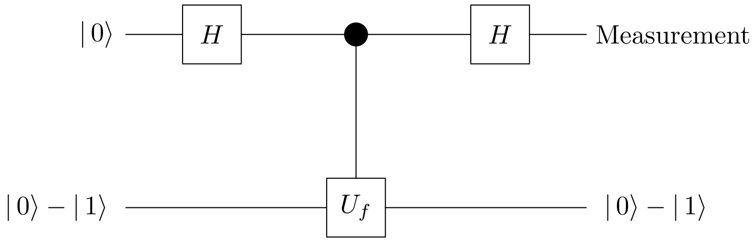


Figure 1: Quantum network which implements Deutsch's algorithm. The middle gate is the f-controlled-NOT which evaluates one of the four functions $f: \{0,1\} \mapsto \{0,1\}$. If the first qubit is measured to be $|\,0\rangle$, then the function is constant, and if $|\,1\rangle$, the function is balanced.

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 $x \in \{0,1\}$,

$$|\,x\rangle\,(|\,0\rangle - |\,1\rangle) \overset{f-c-N}{\longrightarrow} |\,x\rangle\,(|\,0 \oplus f(x)\rangle - |\,1 \oplus f(x)\rangle) = (-1)^{f(x)}\,|\,x\rangle\,(|\,0\rangle - |\,1\rangle)\,. \tag{7}$$

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

$$((-1)^{f(0)}\,|\,0\rangle + (-1)^{f(1)}\,|\,1\rangle)(|\,0\rangle - |\,1\rangle)\,. \tag{8}$$

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 $|\,y \oplus f(x)\rangle$.

This state can also be written as

$$(-1)^{f(0)}(|\,0\rangle + (-1)^{f(0)\oplus f(1)}\,|\,1\rangle)(|\,0\rangle - |\,1\rangle)\,, \tag{9}$$

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

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

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.

The Mach-Zehnder interferometer with phases ϕ_0 and ϕ_1 each set to either 0 or π can be regarded as an implementation of the above algorithm. In this case, ϕ_0 and ϕ_1 respectively encode $f(0)$ and $f(1)$ (with π representing 1), and a single photon can query both phase shifters (i.e. $f(0)$ and $f(1)$) in superposition.

More general algorithms may operate not just on single qubits, as in Deutsch's case, but on sets of qubits or 'registers'. The second qubit becomes an auxiliary register $|\,\psi\rangle$ prepared in a superposition of basis states, each weighted by a different phase factor,

$$|\,\psi\rangle = \sum_{y=0}^{2^m-1} e^{-2\pi i y/2^m}\,|\,y\rangle\,. \tag{11}$$

In general, the middle gate which produces the phase shift is some controlled function evaluation. A controlled function evaluation operates on its second input, the 'target', according to the state of the first input, the 'control'. A controlled function f applied to a control state $|\,x\rangle$, and a target state $|\,\psi\rangle$ gives

$$|\,x\rangle\,|\,\psi\rangle \longrightarrow |\,x\rangle\,|\,\psi + f(x)\rangle\,. \tag{12}$$

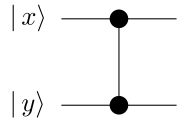
where the addition is mod 2^m or bit by bit. Hence for the register in state (11)

$$|\,x\rangle \sum_{y=0}^{2^m-1} e^{-2\pi i y/2^m}\,|\,y\rangle \longrightarrow e^{2\pi i f(x)/2^m}\,|\,x\rangle \sum_{y=0}^{2^m-1} e^{-2\pi i (y+f(x))/2^m}\,|\,y+f(x)\rangle = e^{2\pi i f(x)/2^m}\,|\,x\rangle\,|\,\psi\rangle\,. \tag{13}$$

Effectively a phase shift proportional to the value of $f(x)$ is produced on the first input.

We will now see how phase estimation on registers may be carried out by networks consisting of only two types of quantum gates: the Hadamard gate $\mathbf{H}$ and the conditional phase shift $\mathbf{R}(\phi)$. The conditional phase shift is the two-qubit gate $\mathbf{R}(\phi)$ defined as

$$\mathbf{R}(\phi) = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & e^{i\phi} \end{pmatrix}$$



$$\Bigg\}\; e^{ixy\phi}\,|\,x\rangle\,|\,y\rangle\,. \tag{14}$$

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). For some of the known quantum algorithms, when working with registers, the Hadamard transformation, corresponding to the beamsplitters in the interferometer, is generalised to a quantum Fourier transform.

4 Quantum Fourier transform and computing phase shifts

The discrete Fourier transform is a unitary transformation of a s–dimensional vector

$$(f(0), f(1), f(2), \ldots, f(s-1)) \to (\tilde{f}(0), \tilde{f}(1), \tilde{f}(2), \ldots, \tilde{f}(s-1)) \tag{15}$$

defined by:

$$\tilde{f}(y) = \frac{1}{\sqrt{s}} \sum_{x=0}^{s-1} e^{2\pi i x y/s} f(x), \tag{16}$$

where $f(x)$ and $\tilde{f}(y)$ are in general complex numbers. 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 (QFT) is a unitary transformation which can be written in a chosen computational basis $\{|0\rangle, |1\rangle, \ldots, |2^n - 1\rangle\}$ as,

$$|x\rangle \longmapsto \frac{1}{\sqrt{s}} \sum_{y=0}^{s-1} \exp(2\pi i x y/s)\; |y\rangle. \tag{17}$$

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

$$\mathrm{QFT} : \sum_x f(x)|x\rangle \longmapsto \sum_y \tilde{f}(y)|y\rangle, \tag{18}$$

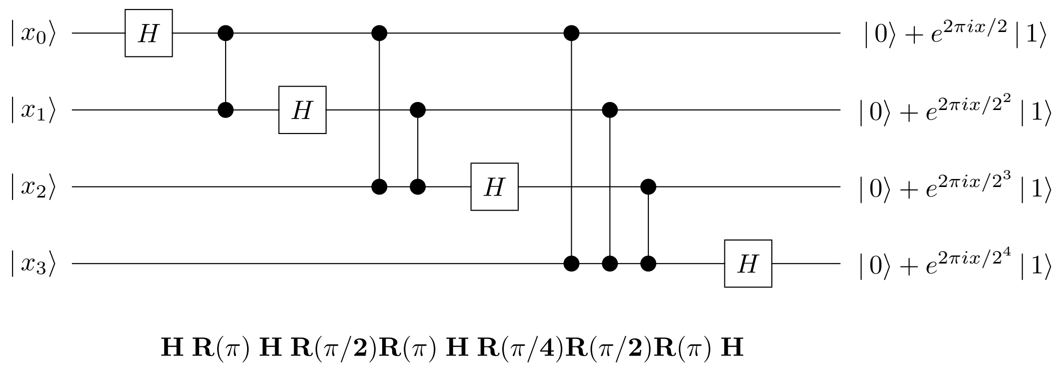

$|x_0\rangle$
$|x_1\rangle$
$|x_2\rangle$
$|x_3\rangle$
H
H
H
H
$|0\rangle + e^{2\pi i x/2}|1\rangle$
$|0\rangle + e^{2\pi i x/2^2}|1\rangle$
$|0\rangle + e^{2\pi i x/2^3}|1\rangle$
$|0\rangle + e^{2\pi i x/2^4}|1\rangle$
$\mathbf{H}\ \mathbf{R}(\pi)\ \mathbf{H}\ \mathbf{R}(\pi/\mathbf{2})\mathbf{R}(\pi)\ \mathbf{H}\ \mathbf{R}(\pi/\mathbf{4})\mathbf{R}(\pi/\mathbf{2})\mathbf{R}(\pi)\ \mathbf{H}$

Figure 2: The quantum Fourier transform (QFT) network operating on four qubits. If the input state represents number $x = \sum_k 2^k x_k$ the output state of each qubit is of the form $|0\rangle + e^{i2^{n-1-k}\phi_x}|1\rangle$, where $\phi_x = 2\pi x/2^n$ and $k = 0, 1, 2 \ldots n-1$. N.B. there are three different types of the $R(\phi)$ gate in the network above: $R(\pi)$, $R(\pi/2)$ and $R(\pi/4)$. The size of the rotation is indicated by the distance between the 'wires'.

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

A given phase $\phi_x = 2\pi x/2^n$ can be encoded by a QFT. In this process the information about ϕ_x is distributed between states of a register. Let x be represented in binary as $x_0 \ldots x_{n-1} \in \{0,1\}^n$, where $x = \sum_{i=0}^{n-1} x_i 2^i$ (and similarly for y). An important observation is that the QFT of x, $\sum_{y=0}^{s-1} \exp(2\pi i x y/s)\,|y\rangle$, is unentangled, and can in fact be factorised as

$$(|0\rangle + e^{i\phi_x}|1\rangle)(|0\rangle + e^{i2\phi_x}|1\rangle)\cdots(|0\rangle + e^{i2^{n-1}\phi_x}|1\rangle)\,. \tag{19}$$

The network for performing the QFT is shown in Fig. 2. The input qubits are initially in some state $|x\rangle = |x_0\rangle|x_1\rangle|x_2\rangle|x_3\rangle$ where $x_0x_1x_2x_3$ is the binary representation of x, that is, $x = \sum_{i=0}^{3} x_i 2^i$. As the number of qubits becomes large, the rotations $R(\pi/2^n)$ will require exponential precision, which is impractical. Fortunately, the algorithm will work even if we omit the small rotations, [4, 5]. The general case of n qubits requires a simple extension of the network following the same pattern of $\mathbf{H}$ and $\mathbf{R}$ gates.

States of the form (19) are produced by function evaluation in a quantum computer. Suppose that U is any unitary transformation on m qubits and $|\psi\rangle$ is an eigenvector of U with eigenvalue $e^{i\phi}$. The scenario is that we do not explicitly know U or $|\psi\rangle$ 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 $|\psi\rangle$. From this, our goal is to obtain an n-bit estimator of ϕ.

In a quantum algorithm a quantum state of the form

$$(|0\rangle + e^{i2^{n-1}\phi}|1\rangle)(|0\rangle + e^{i2^{n-2}\phi}|1\rangle)\cdots(|0\rangle + e^{i\phi}|1\rangle) \tag{20}$$

is created by applying the network of Fig. 3. Then, in the special case where $\phi = 2\pi x/2^n$, the state $|x_0 \cdots x_{n-1}\rangle$ (and hence ϕ) can be obtained by just applying the inverse of the QFT (which is the

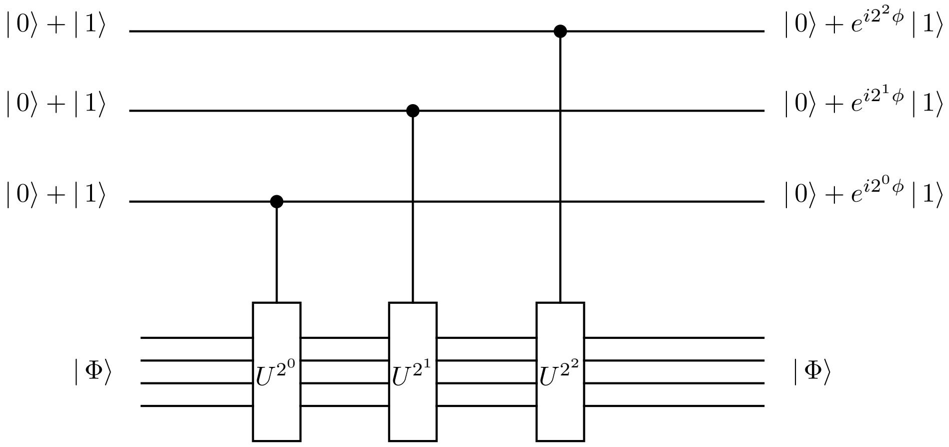


Figure 3: The network which computes phase shifts in Shor's algorithms; it also implements the modular exponentiation function via repeated squarings.

network of Fig. 2 in the backwards direction and with the qubits in reverse order). If x is an n-bit number this will produce the state $|x_0 \cdots x_{n-1}\rangle$ exactly (and hence the exact value ϕ).

However, ϕ is not in general a fraction of a power of two (and may not even be a rational number). For such a $\phi = 2\pi\omega$, it turns out that applying the inverse of the QFT produces the best n-bit approximation of ω with probability at least $4/\pi^2 \approx 0.41$ [2].

5 Quantum factoring

Let me now illustrate the general framework described in the preceding section with Shor's quantum algorithm for efficient factorisation (for a comprehensive discussion of quantum factoring see [2, 6, 7]).

Shor's quantum factoring of an integer N is based on calculating the period of the function $f(x) = a^x \bmod N$ for a randomly selected integer a between 1 and N. For any positive integer y, we define $y \bmod N$ to be the remainder (between 0 and $N-1$) when we divide N by y. More generally, $y \bmod N$ is the unique positive integer $\overline{y}$ between 0 and $N-1$ such that N evenly divides $y - \overline{y}$. For example, $2 \bmod 35 = 2$, $107 \bmod 35 = 2$, and $-3 \bmod 35 = 32$. We can test if a is relatively prime to N using the Euclidean algorithm. If it is not, we can compute the greatest common divisor of a and N using the extended Euclidean algorithm. This will factor N into two factors N_1 and N_2 (this is called *splitting* N). We can then test if N_1 and N_2 are powers of primes, and otherwise proceed to split them if they are composite. We will require at most $\log_2(N)$ splittings before we factor N into its prime factors.

It turns out that for increasing powers of a, the remainders form a repeating sequence with a period r. We can also call r the *order* of a since $a^r = 1 \bmod N$. Once r is known, factors of N are obtained by calculating the greatest common divisor of N and $a^{r/2} \pm 1$.Suppose we want to factor 35 using this method. Let $a = 4$. For increasing x the function $4^x \bmod 35$ forms a repeating sequence $4, 16, 32, 29, 9, 1, 4, 16, 29, 32, 9, 1, \ldots$. The period is $r = 6$, and $a^{r/2} \bmod 35 = 29$. Then we take the greatest common divisor of 28 and 35, and of 30 and 35, which gives us 7 and 5, respectively, the two factors of 35. Classically, calculating r is at least as difficult as trying to factor N; the execution time of the best currently-known algorithms grows exponentially with the number of digits in N. Quantum computers can find r very efficiently.

Consider the unitary transformation U_a that maps $|\,x\rangle$ to $|\,ax \bmod N\rangle$. Such a transformation is realised by simply implementing the reversible classical network for multiplication by a modulo N using quantum gates. The transformation U_a, like the element a, has order r, that is, $U_a^r = I$, the identity operator. Such an operator has eigenvalues of the form $e^{\frac{2\pi i k}{r}}$ for $k = 0, 1, 2, \ldots, r-1$. In order to formulate Shor's algorithm in terms of phase estimation let us apply the construction from the last section taking

$$|\,\psi\rangle = \sum_{j=0}^{r-1} e^{\frac{-2\pi i j}{r}} \left|\,a^j \bmod N\right\rangle \ . \tag{21}$$

Note that $|\,\psi\rangle$ is an eigenvector of U_a with eigenvalue $e^{2\pi i(\frac{1}{r})}$. Also, for any j, it is possible to implement efficiently a controlled-$U_a^{2^j}$ gate by a sequence of squaring (since $U_a^{2^j} = U_{a^{2^j}}$). Thus, using the state $|\,\psi\rangle$ and the implementation of controlled-$U_a^{2^j}$ gates, we can directly apply the method of the last section to efficiently obtain an estimator of $\frac{1}{r}$.

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

$$|\,\psi_k\rangle = \sum_{j=0}^{r-1} e^{-\frac{2\pi i k j}{r}} \left|\,a^j \bmod N\right\rangle \ , \tag{22}$$

where k is from $\{0, \ldots, r-1\}$ would also do the job. For each $k \in \{0, 1, \ldots, r-1\}$, the eigenvalue of state $|\,\psi_k\rangle$ is $e^{2\pi i(\frac{k}{r})}$, and we can again use the technique from the last section to efficiently determine $\frac{k}{r}$ and if k and r are coprime then this yields r (see [2] for more detailed analysis). Now the key observation is that

$$|\,1\rangle = \sum_{k=1}^{r} |\,\psi_k\rangle \ , \tag{23}$$

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

If we substituted $|\,1\rangle$ in place of $|\,\psi\rangle$ in the last section then effectively we would be estimating one of the r, randomly chosen, eigenvalues $e^{2\pi i(\frac{k}{r})}$. This demonstrates that Shor's algorithm, in effect, estimates the eigenvalue corresponding to an eigenstate of the operation U_a that maps $|\,x\rangle$ to $|\,ax \bmod N\rangle$. A classical procedure - the continued fractions algorithm - can be employed to estimate r from these results. The value of r is then used to factor the integer.

6 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?

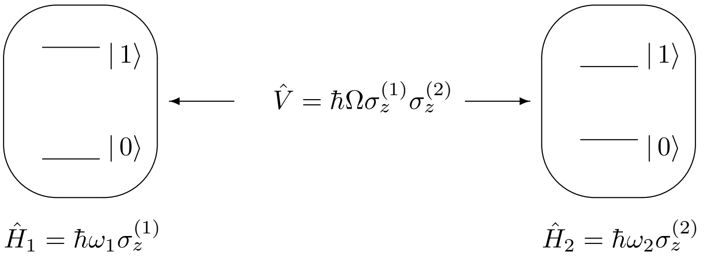

$|\,1\rangle$
$|\,0\rangle$
$\hat{V} = \hbar\Omega\sigma_z^{(1)}\sigma_z^{(2)}$
$|\,1\rangle$
$|\,0\rangle$
$\hat{H}_1 = \hbar\omega_1\sigma_z^{(1)}$
$\hat{H}_2 = \hbar\omega_2\sigma_z^{(2)}$

Figure 4: 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 realisation of the quantum controlled-NOT gate.

Single qubit quantum gates are regarded as relatively easy to implement. For example, a typical quantum optical realisation 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,

$$U = |\,0\rangle\,\langle 0\,| \otimes U_0 + |\,1\rangle\,\langle 1\,| \otimes U_1 + \ldots + |\,k\rangle\,\langle k\,| \otimes U_k, \tag{24}$$

where the projectors refer to quantum states of the control subsystem and the unitary operations U_i are performed on the target subsystem [8]. 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.4). 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.

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 learn about which computational path was taken in the multi-particle 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 trick to protect quantum interference give some hope that complex multi-particle interferometers will be build in a not too distant future.

7 Concluding remarks

Quantum computers use the quantum interference of different computational paths to enhance correct outcomes and suppress erroneous outcomes of computations; they act as (multi-particle) interferometers. This way of thinking about quantum computation provides lots of insights and makes the whole subject less mysterious.

References

[1] R. Feynman, Simulating physics with computers. Int. J. Theor. Phys. **21**, 1982, pp. 467-488.

[2] R. Cleve, A. Ekert, C. Macchiavello, and M. Mosca: Quantum Algorithms Revisited, Proc. R. Soc. Lond. A **454**, 1998, pp. 339–354.

[3] D. Deutsch: Quantum-theory, the Church-Turing principle and the universal quantum computer. Proc. R. Soc. Lond. A **400**,1985, pp. 97-117.

[4] D. Coppersmith: An Approximate Fourier Transform Useful in Quantum Factoring, IBM Research Report No. RC19642, 1994.

[5] A. Barenco, A. Ekert, K. Suominen and P. Törma: Approximate quantum Fourier-transform and decoherence. Phys. Rev. A **54**, 1996, pp. 139-146.

[6] P.Shor: Algorithms for quantum computation: Discrete logarithms and factoring. Proc. 35th Ann. Symp. on Foundations of Comp. Sci., 1994, pp. 124–134.

[7] A. Ekert and R. Jozsa, Quantum computation and Shor’s factoring algorithm, Rev. Mod. Phys. **68**, 733, 1996, pp. 733-753.

[8] Barenco, A., Deutsch, D., Ekert, A., and Jozsa, R. Phys. Rev. Lett. **74**, 4083 (1995).