Quantum Algorithms
part III

Ian Glendinning

January 8, 2002

Outline

- Introduction
- Quantum Bits and Registers
- Quantum Circuits
- Quantum Algorithms
- Summary and Outlook

Quantum Algorithms

- Deutsch (1985): two function evaluations for the price of one
- Deutsch-Jozsa (1992): $O(2^n)$ evaluations for the price of one
- Simon (1994): simple problem with polynomial time quantum algorithm but no polynomial time randomized algorithm
- Shor (1994): factorizing large numbers
- Grover (1996): unstructured search
- Categories of Quantum Algorithms
- *Experimental realization of Shor's algorithm (Dec. 2001)*

Shor's Algorithm

Inspired by Simon, Shor combined some old ideas:

- Factorization of integers can be reduced to the problem of finding the period of a function
- A Fourier transform puts the period of any periodic function into multiples of the reciprocal of the period

with a clever use of the Quantum Fourier Transform (QFT) and new methods to extract the exact period, and he was able to show that it could be done efficiently.

The Quantum Fourier Transform

The discrete Fourier transform (DFT) transforms a set $x_0, \ldots, x_{N-1}$ of N complex numbers into a set of complex numbers $y_0, \ldots, y_{N-1}$

$$y_k \equiv \frac{1}{\sqrt{N}} \sum_{j=0}^{N-1} x_j e^{2\pi i j k/N}$$

The *quantum Fourier transform* (QFT) is exactly the same transformation, though the notation is somewhat different:

$$|j\rangle \longrightarrow \frac{1}{\sqrt{N}} \sum_{k=0}^{N-1} e^{2\pi i j k/N} |k\rangle$$

where $N = 2^n$. It can be shown that the QFT is unitary.

The Quantum Fourier Transform

Equivalently, the action on an arbitrary superpositon state is:

$$\begin{aligned}
\sum_{j=0}^{N-1} x_j |j\rangle \quad \longrightarrow \quad & \frac{1}{\sqrt{N}} \sum_{j=0}^{N-1} x_j \left[\sum_{k=0}^{N-1} e^{2\pi ijk/N} |k\rangle \right] \\
= \quad & \frac{1}{\sqrt{N}} \sum_{k=0}^{N-1} \left[\sum_{j=0}^{N-1} e^{2\pi ijk/N} x_j \right] |k\rangle \\
= \quad & \frac{1}{\sqrt{N}} \sum_{k=0}^{N-1} y_k |k\rangle
\end{aligned}$$

and we can see that the QFT corresponds to a vector notation for the DFT, for the case $N = 2^n$.

The Quantum Fourier Transform

- The QFT can be implemented with $O(n^2)$ gates
- Whereas the best classical algorithms for the DFT, such as the FFT, require $O(n2^n)$ gates, exponentially more than the QFT!
- *But:* measurement only gives one value of $|k\rangle$ and there is no way to determine the Fourier amplitudes y_k
- However, it still turns out to be possible to find the period of a periodic function

Quantum Period-Finding

If f is a periodic integer function such that $f(x+r) = f(x)$ for unknown period $r < n$ and $f(x)$ is in the range $0 \ldots n-1$

1. Find q, the power of 2 with $n^2 \leq q < 2n^2$
2. Initialize two quantum registers, each with $\log_2 q$ bits, to $|0\rangle|0\rangle$
3. Apply a Hadamard transform to the first register

$$\rightarrow \frac{1}{\sqrt{q}} \sum_{x=0}^{q-1} |x\rangle|0\rangle$$

Quantum Period-Finding

3. Apply the operator U_f, defined as $U_f|x\rangle|y\rangle = |x\rangle|y \oplus f(x)\rangle$

$$\rightarrow \frac{1}{\sqrt{q}} \sum_{x=0}^{q-1} |x\rangle|f(x)\rangle$$

4. Measure the second register, giving the value u

$$\rightarrow \frac{1}{\sqrt{M}} \sum_{j=0}^{M-1} |l + jr\rangle|u\rangle$$

where $l + jr$ are the values of x for which $f(x) = u$, for $j = 0, 1, 2 \ldots M - 1$, and the x register is now in a superposition of M states

Quantum Period-Finding

Since M is the largest integer such that $l + (M-1)r \leq q$, we have $l \leq r < n$ we have $M \approx \frac{q}{r}$

We will make the simplifying assumption that r divides q exactly, so $M = q/r$. This restriction does not affect the essential ideas, but without it some extra complications must be taken into account.

We are now only interested in the first register, which we can write:

$$|\phi_l\rangle = \sqrt{\frac{r}{q}} \sum_{j=0}^{\frac{q}{r}-1} |l + jr\rangle$$

Quantum Period-Finding

5. Apply a Quantum Fourier transform to the first register

$$
\begin{aligned}
QFT|\phi_l\rangle &= \frac{1}{\sqrt{q}} \sum_{c=0}^{q-1} \sqrt{\frac{r}{q}} \sum_{j=0}^{\frac{q}{r}-1} e^{2\pi i c(l+jr)/q} |c\rangle \\
&= \frac{\sqrt{r}}{q} \sum_{c=0}^{q-1} e^{2\pi i c l/q} \left(\sum_{j=0}^{\frac{q}{r}-1} e^{2\pi i c j r/q} \right) |c\rangle \\
&= \sum_{c=0}^{q-1} \alpha_c |c\rangle
\end{aligned}
$$

Quantum Period-Finding

If c is a multiple of q/r then $e^{2\pi icjr/q} = 1$ and if c is not a multiple of q/r then

$$\sum_{j=0}^{\frac{q}{r}-1} = 0$$

because the sum is over a set of q/r roots of unity equally spaced around the unit circle. Thus

$$\alpha_c = \begin{cases} \frac{1}{\sqrt{r}} e^{2\pi icl/q} & \text{if } c \text{ is a multiple of } q/r \\ 0 & \text{otherwise} \end{cases}$$

Quantum Period-Finding

and therefore

$$|\phi_{out}\rangle = QFT|\phi_l\rangle = \frac{1}{\sqrt{r}}\sum_{j=0}^{r-1} e^{2\pi icl/q}|j\frac{q}{r}\rangle$$

6. Measure $|\phi_{out}\rangle$ giving $c = \lambda q/r$, where λ is a random integer in the range 0 to $r-1$
7. Therefore $\lambda/r = c/q$ and if λ and r have no common factors ($\gcd(\lambda, r) = 1$) then we can cancel c/q down to an irreducible fraction, and obtain $\lambda = c/\gcd(c, q)$ and $r = q/\gcd(c, q)$

Quantum Period-Finding

8. If λ and r have a common factor then the algorithm fails, and must be repeated from the start, but if it is repeated $O(\log\log r)$ times then the probability of success can be as close to 1 as desired and therefore r can be determined efficiently

In the general case when $M \neq q/r$ the calculation of the probabilities is more sophisticated, and a more sophisticated (classical) algorithm is needed to determine r in step 7 (using continued fractions), but no new quantum computing ideas are involved.

Shor's Factorization Algorithm

1. If N is even, return factor 2
2. Determine whether $N = n^m$ for integers $n > 1$ and $m \geq 2$ and if so return the factor n
3. Randomly choose $2 \leq a \leq N - 1$
4. If $\gcd(a, N) > 1$ then return the factor $\gcd(a, N)$
5. Find the period r of $f(x) = a^x \bmod N$
6. If r is even and $a^{r/2} \neq -1(\bmod N)$ compute $\gcd(a^{r/2} - 1, N)$ and $\gcd(a^{r/2} + 1, N)$, and test to see if one of these is a non-trivial factor ($\neq 1$), returning that factor if so. Otherwise go to step 3.

Grover's Algorithm

- The problem: In an unsorted list of N items, one item satisfies a given condition - find it
- Classically this would require on average $N/2$ steps, but Grover's algorithm can search a quantum database in $O(\sqrt{N})$ steps
- Concentrate on the *index* to the elements, a number in the range 0 to $N-1$, where for convenience $N = 2^n$
- The condition can be represented by a function f of an integer in the range 0 to $N-1$, such that $f(x) = 1$ if x is the index of the item we're seeking, and $f(x) = 0$ otherwise.

Grover's Algorithm

1. Initialize an n-bit quantum register to $|0\rangle^{\otimes n}$
2. Apply a Hadamard transform $\rightarrow \frac{1}{\sqrt{2^n}} \sum_{x=0}^{2^n-1} |x\rangle$
3. Apply the *oracle* operator O defined by $O|x\rangle = (-1)^{f(x)}|x\rangle$
4. Apply the 'Diffusion' operator D with matrix elements: $D_{ij} = \frac{2}{N}$ if $i \neq j$ and $D_{ii} = -1 + \frac{2}{N}$
5. Iterate steps 3 and 4 $\lceil \frac{\pi}{4}\sqrt{N} \rceil$ times
6. Measure the register to get x_0. If $f(x_0) \neq 1$, go to step 1

Unitarity

The operator O is clearly unitary. To see that D is unitary, write $D = A - I$ where every element of A is $\frac{2}{N}$, i.e. $A_{ij} = \frac{2}{N}$. Therefore

$$D^\dagger D = (A - I)^\dagger (A - I) = (A - I)^2 = A^2 - 2A + I$$

and

$$[A^2]_{ij} = \sum_{k=0}^{N-1} A_{ik} A_{kj} = N \left(\frac{2}{N} \right)^2 = \frac{4}{N} = 2A_{ij}$$

so $A^2 = 2A$ and

$$D^\dagger D = I$$

as required for unitarity.

The Diffusion Operator

To understand what D does, consider its action on a general state

$$D \sum_k \alpha_k |k\rangle = (A - I) \sum_k \alpha_k |k\rangle = \sum_k \alpha_k A |k\rangle - \sum_k \alpha_k |k\rangle$$

To evaluate $A|k\rangle$ consider the matrix-vector product $y = Ax$

$$y_i = \sum_j A_{ij} x_j = \sum_j \frac{2}{N} x_j$$

$|k\rangle$ has kth element equal to 1, all others 0, so $A|k\rangle = \sum_i \frac{2}{N} |i\rangle$ and

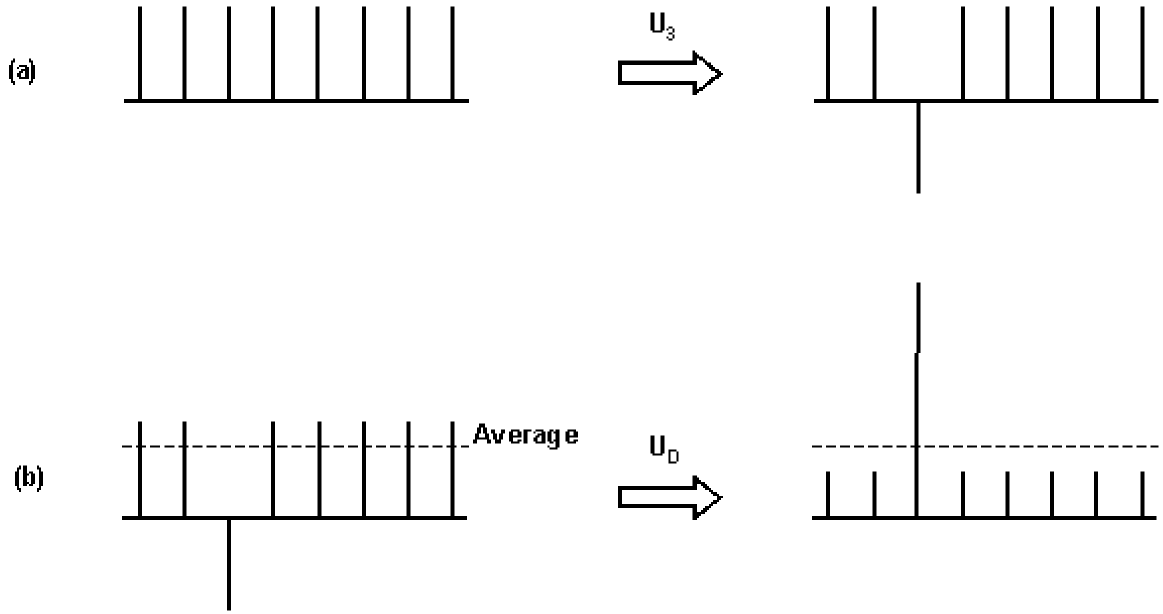
$$D \sum_k \alpha_k |k\rangle = \sum_k \alpha_k \sum_i \frac{2}{N} |i\rangle - \sum_k \alpha_k |k\rangle$$

The Diffusion Operator

$$D \sum_k \alpha_k |k\rangle = 2\langle\alpha\rangle \sum_i |i\rangle - \sum_k \alpha_k |k\rangle$$
$$= \sum_k \left[2\langle\alpha\rangle - \alpha_k\right] |k\rangle$$

- The diffusion operator corresponds to a reflection of all the amplitudes about their mean: $\alpha_k - 2(\alpha_k - \langle\alpha\rangle) = 2\langle\alpha\rangle - \alpha_k$
- Because the sign of the amplitude we want has been reversed, it is amplified at the expense of the others

Amplitude Amplification



Categories of Quantum Algorithms

- Fourier transform
  - Deutsch, Period-finding, Factoring, Discrete logarithm, etc.
  - All special cases of the *Hidden subgroup problem*
  - Encompases all known ‘exponentially fast’ QFT applications
- Search
  - Grover, Minimum, Quantum counting, speeding up solution of NP-complete problems
- Simulation of quantum systems

Experimental realization of Shor's algorithm

- Nature 20/27 December 2001: Nature **414**, 883-887 (2001)
- Implementation of the simplest instance of Shor's algorithm: factorization of $N = 15$ (whose prime factors are 3 and 5)
- Seven spin-1/2 nuclei in a molecule were used as qubits
- Manipulated with room temperature liquid-state nuclear magnetic resonance (NMR) techniques

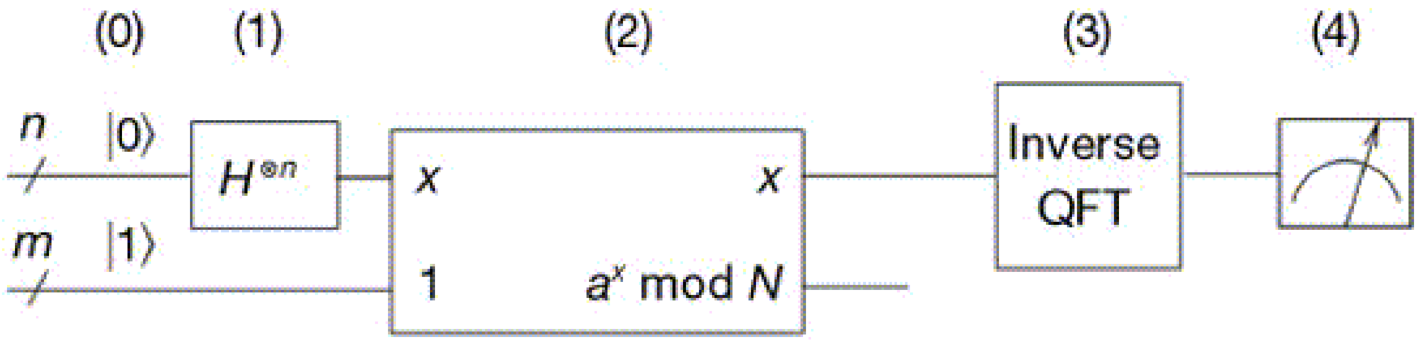
Experimental realization of Shor's algorithm

- The simplest meaningful instance of Shor's algorithm is for $N = 15$ because the algorithm fails for N even or a prime power
- Even for such small N, quantum factorization poses a significant experimental challenge, requiring coherent control of seven qubits during a long sequence of controlled interactions
- Even after maximal reduction of the quantum circuit, interactions between almost all pairs of qubits are needed

Experimental realization of Shor's algorithm

- Furthermore, numerically predicting the outcome of the experiments is impractical, due to the enormous size of the state space transformations
- Including decoherence effects, the transformations are described by $\sim 4^7 \times 4^7$ real parameters (1 Gbyte with 4 byte words)!
- Implementation of the algorithm can be broken into four steps, with the most complex being the computation of $f(x) = a^x \bmod N$ for 2^n values of x in parallel

Quantum Circuit for Shor's Algorithm



Implementing Modular Exponentiation

Following standard techniques, the following identity was utilized:

$$a^x = a^{(2^{n-1}x_{n-1}+\cdots+2x_1+x_0)} = a^{2^{n-1}x_{n-1}} \ldots a^{2x_1} a^{x_0}$$

where x_k are the binary digits of x.

Modular exponentiation thus consists of serial multiplication by $a^{2^k} \bmod N$ for all k for which $|x_k\rangle = |1\rangle$.

The powers a^{2^k} can be efficiently pre-computed on a classical machine.

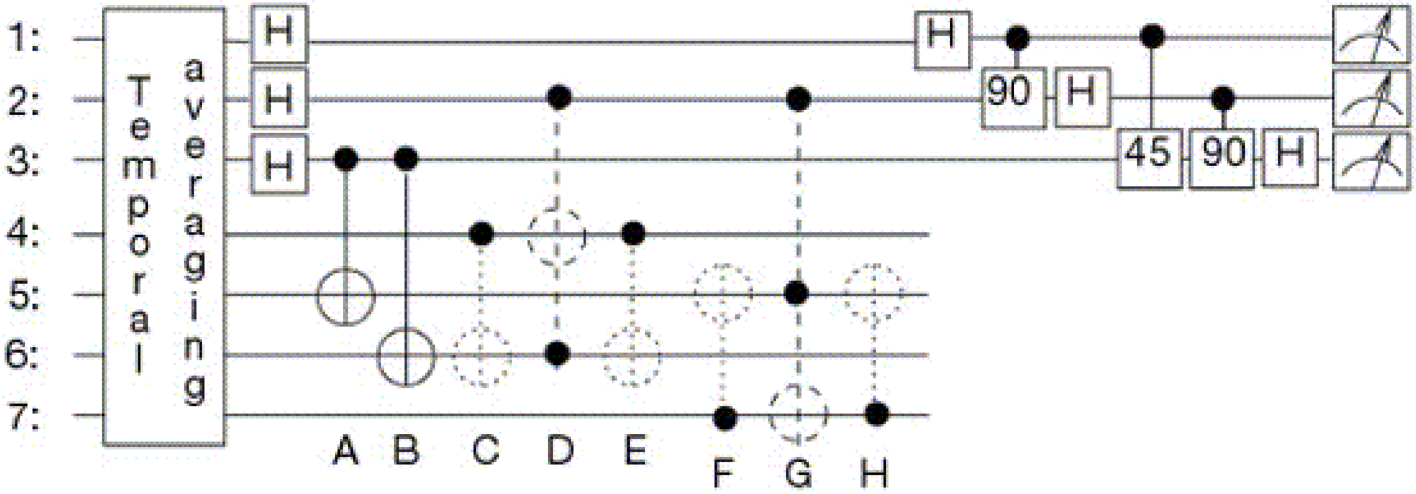
Implementing Modular Exponentiation

- Shor's algorithm requires $\gcd(a, N) = 1$, so for $N = 15$, $a = 2$, 4, 7, 8, 11, 13 or 14
- If $a = 2, 7, 8$ or 13 then $a^4 \bmod 15 = 1$ so $a^{2^k} \bmod N = 1$ if $k \geq 2$
  - Because $ab \bmod N = (a \bmod N)(b \bmod N) \bmod N$
  - E.g. $a^8 \bmod 15 = a^4 a^4 \bmod 15 =$ $(a^4 \bmod 15)(a^4 \bmod 15) \bmod 15 = 1$
  - So the two bits x_0 and x_1 determine the product
- If $a = 4, 11$ or 14, we find that $a^2 \bmod 15 = 1$
  - So only x_0 is needed to determine the product

Implementing Modular Exponentiation

- Thus the first register can be as small as two qubits ($n = 2$)
- But three qubits allow more periods to be detected
- Together with the four qubits needed to hold $f(x)$, seven qubits are therefore needed in total

Detailed Quantum Circuit for $a = 7$



Peephole Compiler Optimization

1. The control qubit of gate C is $|0\rangle$, so C was suppressed
2. Similarly gate F was replaced by X_5
3. Gates H and E are inconsequential for the final state of the first register, so they were omitted
4. The targets of the doubly controlled NOT gates D and G are in basis states, so the operation of the gate can be simplified
5. To keep ‘refocusing’ simple, A was carried out after E

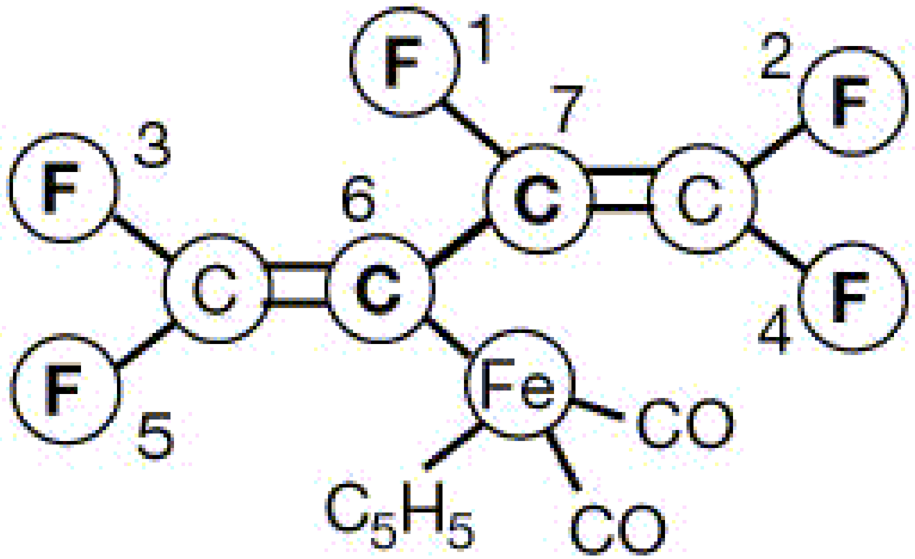
Run Time

- After all these simplifications, the pulse sequence for $7^x \bmod 15$ was ~ 400 ms long
- State preparation took ~ 200 ms, and the inverse QFT took ~ 120 ms
- The duration of the complete sequence for the Shor algorithm was thus up to ~ 720 ms

Experimental Details

- The custom-synthesized molecule used as the quantum computer contained five ^{19}F and two ^{13}C spin-1/2 nuclei as qubits
- In a magnetic field, each spin has two discrete energy states $|0\rangle$ (spin-up) and $|1\rangle$ (spin-down)
- The desired initial state is $|0000001\rangle$, but the experiment started from thermal equilibrium
- 7-spin effective pure state obtained by temporal averaging over 36 experiments preceded by different CN_{ij} and X_i gate sequences

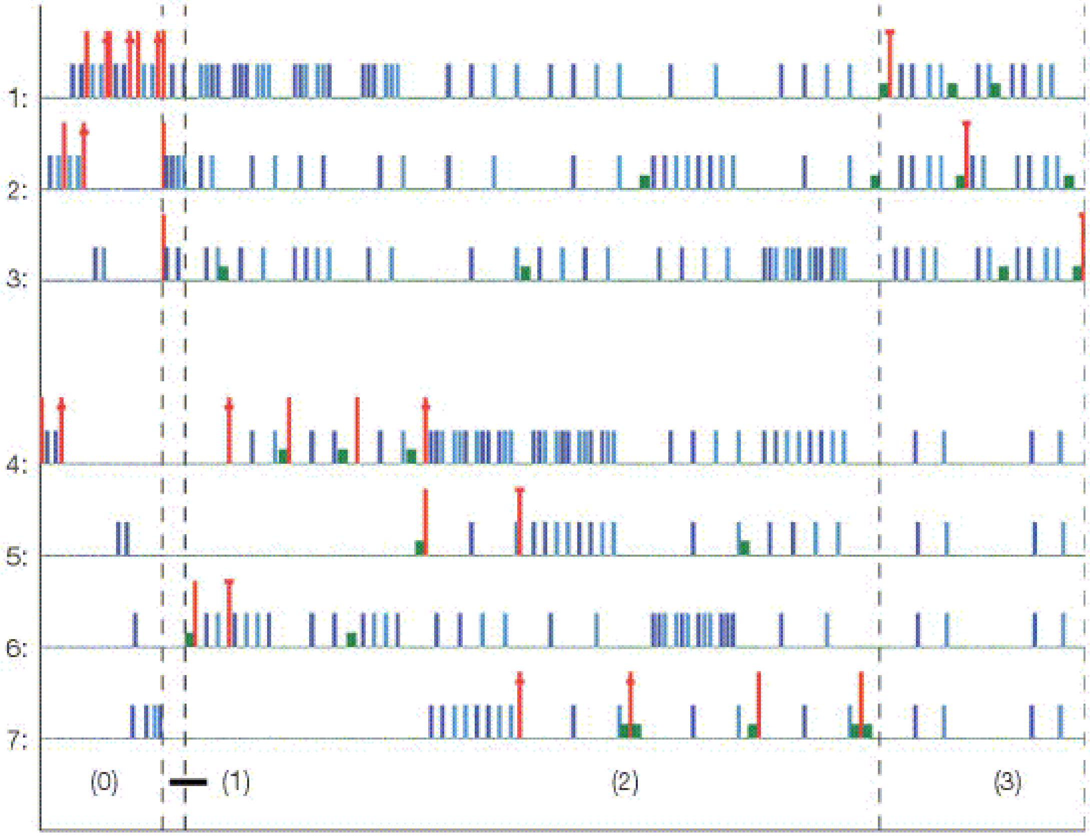
The Quantum Computer Molecule



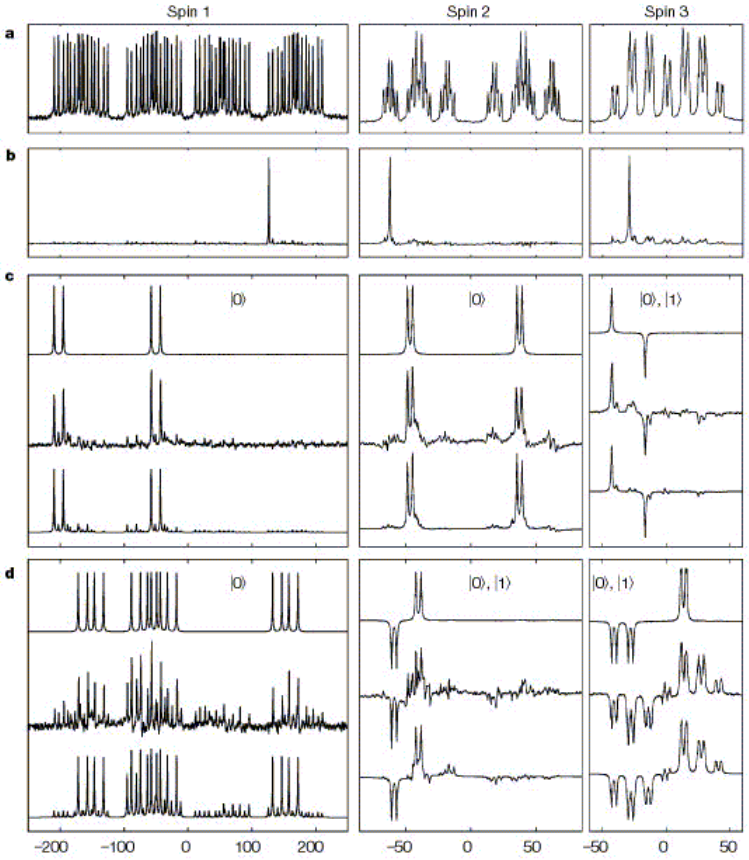
Experimental Details

- The algorithm was tested on two representative parameter choices: $a = 11$ ('easy') and $a = 7$ ('difficult')
- The quantum circuit was realised with a sequence of ~ 300 ($a = 7$) spin-selective radio-frequency (r.f.) pulses, separated by time intervals of free evolution
- After the sequence of r.f. pulses, the state of the first three qubits was estimated by NMR spectroscopy

Pulse Sequence for $a = 7$



NMR Spectra


Spin 1
Spin 2
Spin 3
a
b
c
d
|0⟩
|0⟩
|0⟩, |1⟩
|0⟩
|0⟩, |1⟩
|0⟩, |1⟩
−200
−100
0
100
200
−50
0
50
−50
0
50

Results

- In this experiment an ensemble of independent quantum computers was used, so the measurement gives the bit-wise average of $8\lambda/r$ instead of $8\lambda/r$ for a random λ
- For the case $a = 11$, qubits 1 and 2 are $|0\rangle$ (spectral lines up) and qubit 3 is an equal mixture of $|0\rangle$ and $|1\rangle$ (lines up and down)
- The first register is thus in a mixture of $|000\rangle$ and $|100\rangle$, or $|0\rangle$ and $|4\rangle$, and the periodicity is 4
- Therefore $8/r = 4$, so $r = 2$
- $\gcd(11^{2/2} \pm 1, 15) = 3$ and 5, the prime factors of 15!

Results

- For the case $a = 7$, qubit 1 is in $|0\rangle$ and qubits 2 and 3 are in a mixture of $|0\rangle$ and $|1\rangle$
- The first register is thus in a mixture of $|000\rangle$, $|010\rangle$, $|100\rangle$ and $|110\rangle$, or $|0\rangle$, $|2\rangle$, $|4\rangle$, and $|6\rangle$, and the periodicity is 2
- Therefore $8/r = 2$, so $r = 4$
- $\gcd(7^{4/2} \pm 1, 15) = 3$ and 5
- Thus, even after the long and complex pulse sequence of the difficult case, the experimental data conclusively indicate the successful execution of Shor's algorithm to factor 15

Summary and Outlook

- Quantum Algorithms can solve hard problems
- However, the class of problems for which quantum computers have a big advantage remains to be characterized
- There is almost daily progress in developing new quantum algorithms and technologies for quantum computation
- Distribution of e-prints via the Internet has contributed to this:
  - http://arXiv.org/archive/quant-ph
- **Further Information:** http://www.qubit.org/