Solid state quantum computers: A nanoscopic solution to the Moore's Law problem

Joseph Ng[a] and Derek Abbott[a]

[a]Centre of Biomedical Engineering (CBME),
Dept. of Electrical and Electronic Engineering,
University of Adelaide,
Adelaide, SA 5005, AUSTRALIA

ABSTRACT

The computer industry has followed Moore's Law closely and faithfully over the past few decades. However, transistors cannot continue to shrink at their current rate forever, and new methods of computation must be explored. Quantum computation is one such method that has received much attention over the past few years and will heavily rely on technological advances in the smart electronics and nanotechnology arena. In this review, we will present some of the problems facing classical computers and why quantum computers may be a viable alternative. We will briefly describe some of the "killer" quantum applications, such as Deutsch's,[1] Shor's[2] and Grover's[3] algorithms that demonstrate the computational powers of quantum computation. Kane's solid state quantum computer in silicon[4,5] promises to demonstrate some of these applications. However there remain many significant technological difficulties which will need to be overcome if we are to see a useful quantum computer. The main technological challenges, for Kane's solid-state computer, of interest to the smart materials and structures community, will be highlighted.

Keywords: Smart electronics, Smart structures and materials, Solid-state quantum computation

1. INTRODUCTION

Very few people will dispute that Moore's Law - the rough rule that says the processing power of computers doubles every 18 months - will break down sometime in the future. A rough projected date is somewhere between 2010 and 2020. What then? Will the computer industry hit a barrier and then stop? Nobody knows, but unless our method of computing changes fundamentally, computers will not be able to keep up with the ever increasing demand of computing power. This paper outlines one approach that has received a lot of attention over the past few years, Quantum Computers (QCs).

Section two discusses the current problems that classical computers are facing. Traditionally, increasing the computational power is primarily achieved by shrinking the devices smaller and smaller. There are obviously practical limitations to the manufacturing aspects of computers as well as physical and functional limits as to how small these devices can get. For many of these limitations, there are no known solutions yet. There are also fundamental limits to classical computing. These fundamental limits come about from the way we design and operate our computers, and thus, it is widely accepted that there are no classical solutions, practically or theoretically, to these limits.

The third section briefly explains a few basic quantum mechanics concepts and how quantum computers can address many of these issues. By exploiting the strange properties of the quantum world, many conventional problems can be solved much more efficiently on a quantum computer. Algorithms employing superposition and entanglement of qubits (quantum bits) have been shown to dramatically speed up many problems, such as Deutsch's problem[1] and non-polynomial(NP) problems such as finding the two prime factors of a particular number[2] - the key to cracking RSA cryptography. Shor's algorithm employs the technique of quantum Fourier transforms. Another application is Grover's search algorithm,[3] which square-roots the conventional processing complexity when searching for a particular item in an unsorted database.

In section four, we will describe a particular proposal by Kane[4] to realise a quantum computer in silicon. It is widely believed, and with good reason, that quantum computers will ultimately be realised in solid state. However,

Further author information: (Send correspondence to Joseph Ng)
Joseph Ng: E-mail: jng@eleceng.adelaide.edu.au
Derek Abbott: E-mail: dabbott@eleceng.adelaide.edu.au

Smart Electronics and MEMS II, Derek Abbott, Vijay K. Varadan, Karl F. Boehringer, Editors,
Proceedings of SPIE Vol. 4236 (2001) © 2001 SPIE · 0277-786X/01/$15.00

there are some very formidable technological challenges that must be overcome before we see a working solid state QC in action; problems involving the building of a QC as well as the operation of the quantum computer.

2. LIMITATIONS OF CLASSICAL COMPUTATION

As devices shrink smaller, not only do they become harder to make, they become increasingly difficult to manufacture precisely. Even small uncertainties can result in large variations in device characteristics. Larger percentage margins will need to be given in designs. More error correction and control logic will also be required.

Classical computers require conducting interconnects to carry the voltage and current signals from one device to another. The capacitances of these wires is a major limiting factor to the speed in which signals can travel, thus limiting the speed of computation. Much effort has been put into routing interconnects to reduce the length of these wires. As feature sizes continue to decrease, the wires are getting thinner and closer together, thus increasing the capacitance between the wires. Increased capacitance will require more current to maintain the speed of communication. However, the decreased cross-sectional area of the wires places a limit on the maximum current that it can carry.

As more and more circuitry is packed into smaller and smaller areas, the heat generated needs to be removed to avoid destroying the fragile devices. The ability to dissipate power plays a major role in determining the maximum component density of devices. It has been roughly calculated that the maximum gate density using present technology is approximately $1.7\text{x}10^7/\text{cm}^2$.[6]

The solubility limit of dopant atoms is also a growing concern when attempting to increase the charge concentration of current devices. As the dopant concentration increases, the dopant atoms interact with each other to form clusters.[7] These clusters increase the impedance of the doped regions, and thus increase the power dissipation of the devices.

Despite all the practical problems of classical computers, there are some fundamental limits of the way we perform computations. In particular, NP problems are hard because the computational resources required, such as time and memory, increase dramatically with the size of the problem. If we increase exponential problems by five fold, we will require almost 150 times the processing resources to solve the problem! Problems such as the travelling salesman problem and factoring of large numbers into its prime factors fall into this category. However, quantum computers promise to solve some of these problems by reducing the complexity to polynomial rate of increase rather than an exponential one.

3. QUANTUM COMPUTERS

3.1. Basic quantum mechanics

The power of quantum computation over conventional classical computation comes from the ability to place the "bits" into a superposition of states and the ability to entangle the bits. These bits are thus called qubits (quantum bits). A qubit has two distinct states which we can arbitrarily label 0 and 1 for the purpose of computation. These are orthogonal states in Hilbert space. When a qubit is in a superposition, we can think of it as being both 0 and 1 at the same time. However, when we measure the qubit, the superposition will collapse into one of the two states with the probability defined by the nature of the superposition.

The standard notation for expressing these quantum states is the Dirac Bra-Ket notation. Each state is written as $|\psi\rangle$. So, the 0 state is $|0\rangle$, called the 0 ket. A ket is a complex vector in Hilbert space. Superpositions are expressed as vector sums of state kets. In the case of qubits, it is $a|0\rangle + b|1\rangle$, where a and b are, in general, complex probability amplitudes of the respective kets. A measurement is a projection of this superposition onto the basis kets, with each of the magnitudes being the probability that we will find the qubit in a particular state. In other words, $|a|^2$ and $|b|^2$ is the probability that when we measure the qubit, we will find 0 and 1 respectively. From this, we can also conclude that $|a|^2 + |b|^2 = 1$.

One of the easiest ways to picture a qubit is by considering photon polarisations (Fig. 1). We can define vertical polarisation of the photon as $|0\rangle$ and horizontal polarisation as $|1\rangle$. Now imagine if we have a single photon of 45° polarisation. What happens when this photon arrives at a vertically polarised filter? This is a measurement of the photon, and thus the superposition will collapse. The photon will collapse into either a vertically polarised or a horizontally polarised state with 50/50 probability. This is an even superposition of $|0\rangle$ and $|1\rangle$. i.e. $\frac{1}{\sqrt{2}}|0\rangle + \frac{1}{\sqrt{2}}|1\rangle$.

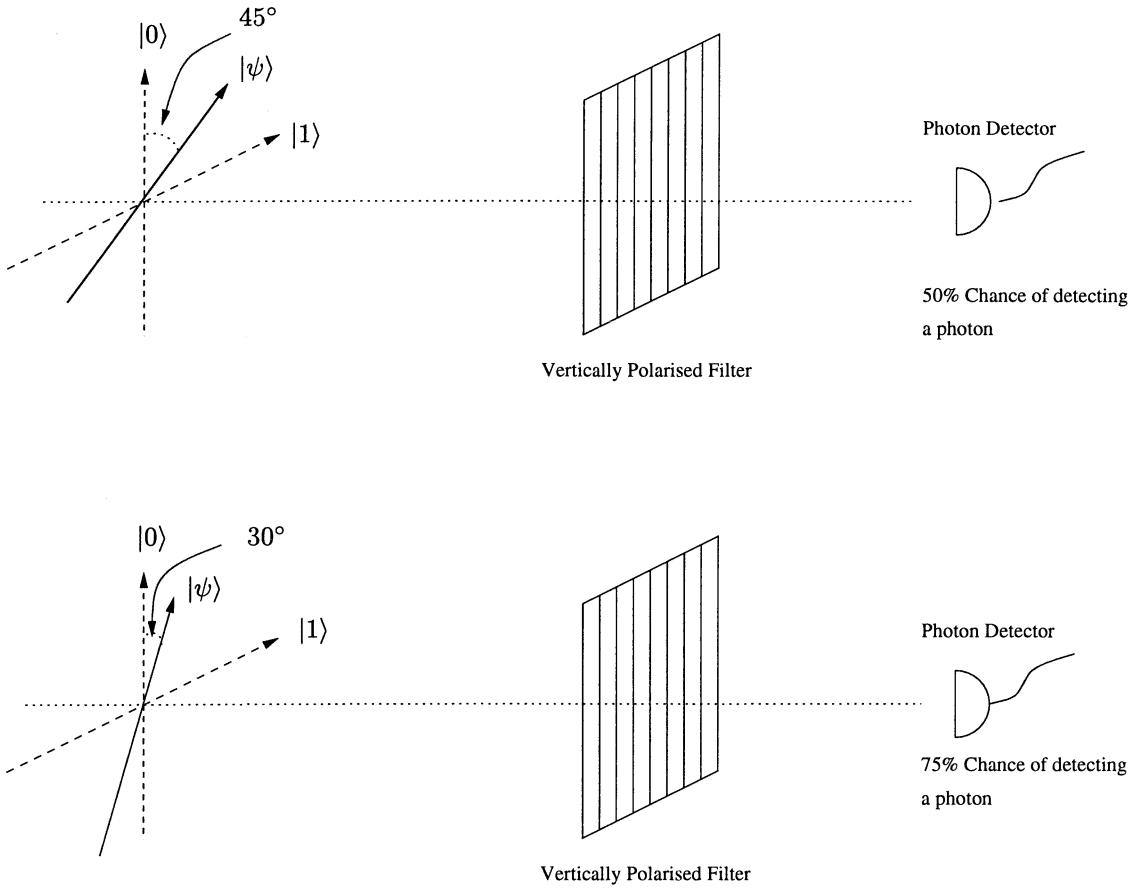

Figure 1. Using photon polarisation as a qubit. For the 45° polarised photon, the photon detector behind the filter has a 50% chance of detecting that the photon has passed through the filter. For the 30° polarised photon, the chance is increased the 75%.

Obviously if the photon is vertically polarised, it will pass through, otherwise, it will not. Now if the photon is 30° polarised, then we can see that this is $\frac{\sqrt{3}}{2}|0\rangle + \frac{1}{2}|1\rangle$, and so this means that we have a $|\frac{\sqrt{3}}{2}|^2 = \frac{3}{4}$ chance of detecting that the photon has passed through the filter.

3.2. Requirements

For any computer, there must be hardware and software. In classical computers, the hardware is the silicon chip, where the bits are represented by different voltages. Classical software a form of higher abstraction level that manipulates the hardware to perform tasks.

In a quantum computer on the other hand, the hardware is the qubits. At the moment, some of the qubits in use could be polarisation of photons (See Section 3.1), physical presence of photons (beam splitters), energy levels of atomic particles,[8] the quantum spin of atomic particles,[9,4] or recently, the charge states of superconducting materials.[10] In this paper, we will consider Kane's solid state quantum computer proposal, which employs nuclear spins of phosphorous atoms as the qubits.

As with classical software, quantum software need not deal with exactly how the hardware is realised, but rather, what to do with the bits/qubits if and when they are realised. This is where the power of quantum computers becomes apparent. Quantum algorithms such as Shor's factoring algorithm,[2] have shown that by exploiting the fuzziness of quantum mechanics we can speed up and solve certain types of problems that no classical computer can ever hope to solve in a reasonable amount of time.

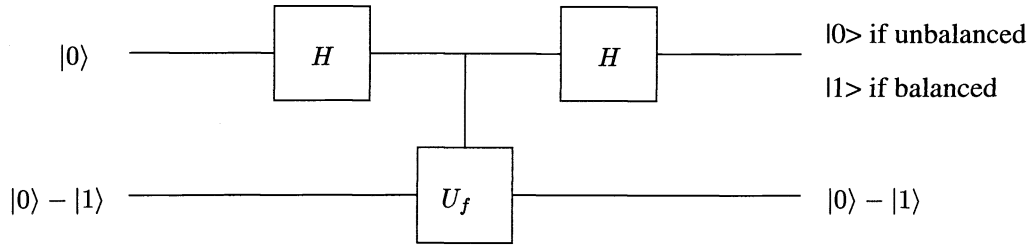

Figure 2. Quantum circuit for solving Deutsch's problem

3.3. Quantum Software

Broadly speaking, there are two major categories of quantum algorithms: Quantum Phase Estimation/Fourier Transform and Quantum Search. This section briefly demonstrates the computational power of a quantum computer. These are not meant to be vigorous derivation of the algorithms, but rather descriptions of how they work and the improvements that they provide over classical methods. For detailed mathematical analysis, see Ref. [11].

3.3.1. Deutsch's Algorithm

Deutsch's problem[1] is perhaps one of the simplest examples where quantum computers can outperform classical computers. The question is: given $f(x)$ does $f(0) = f(1)$?

Classically, we must calculate $f(0)$ and $f(1)$ then compare the two results. This obviously requires two calculations. Using a quantum network, we can do that in one step (Fig. 2).

What we have in Fig. 2 are three quantum gates that manipulate qubits. The Hadamard, H, gate puts a qubit in the zero state into an even superposition of zero and one, $|0\rangle + |1\rangle$. While U_f is a unitary controlled-function gate which performs the function $f(x)$ on the second qubit if the first qubit is in the $|1\rangle$ state, otherwise, the gate leaves the qubit alone. At the end of this simple quantum circuit, if we measure the first qubit in the $|1\rangle$ state, then the function is balanced (i.e. $f(0) = f(1)$), otherwise, if it is in the $|0\rangle$ state, then we have an unbalanced function. Despite this simplicity, we can see that we only need to perform the calculation once in the quantum regime, whereas we need to do it twice classically.

3.4. Quantum Fourier Transform

Fourier transforms is one of the most important and useful tools in engineering. However, calculating the Fourier transform is not an easy task computationally. Classically, the discrete Fourier transform (as opposed to continuous FT since we are dealing with digital systems) is defined as $y_k = \frac{1}{\sqrt{N}} \sum_{j=0}^{N-1} x_j e^{2\pi ijk/N}$.

A fast and widely used classical algorithm for computing the DFT is the Fast Fourier Transform (FFT), which is of order $O(n2^n)$ (where the number of elements $N = 2^n$).

On the other hand, quantum Fourier transform (QFT), defined as $|a\rangle \rightarrow \frac{1}{\sqrt{N}} \sum_{b=0}^{N-1} e^{2\pi iab/N}|b\rangle$ can provide significant speedup *. So how fast is the speedup?

For computation purposes, the above equation is usually written in a more convenient tensor product form.

$|a_1, a_2, ..., a_n\rangle \rightarrow \frac{(|0\rangle + e^{2\pi 0.a_n}|1\rangle)(|0\rangle + e^{2\pi 0.a_{n-1}a_n}|1\rangle)...(|0\rangle + e^{2\pi 0.a_1 a_2 \cdots a_n}|1\rangle)}{2^{n/2}}$

This leads to the circuit in Fig. 3.

As can be seen in Fig. 3 for 2^n elements, there are $n(n+1)/2$ gates involved in performing a quantum Fourier transform. Thus, the complexity of the QFT circuit is $O(n^2)$ which is polynomial! So by doing the transformation in the quantum regime, we have an exponential speedup over the classical Fourier transform method.

Unfortunately, this method cannot be used to simply replace FFTs in conventional applications. In the QFT, the results are probability amplitudes which means that we can not directly measure them. To make use of the QFT, we need applications which are also in the quantum regime which makes use of and manipulates quantum probability amplitudes. One such application is Shor's factoring algorithm.

*QFT is in fact an example of a more general algorithm; that of Kitaev's phase estimation algorithm. See Ref. 11 for an extensive discussion on Kitaev's algorithm

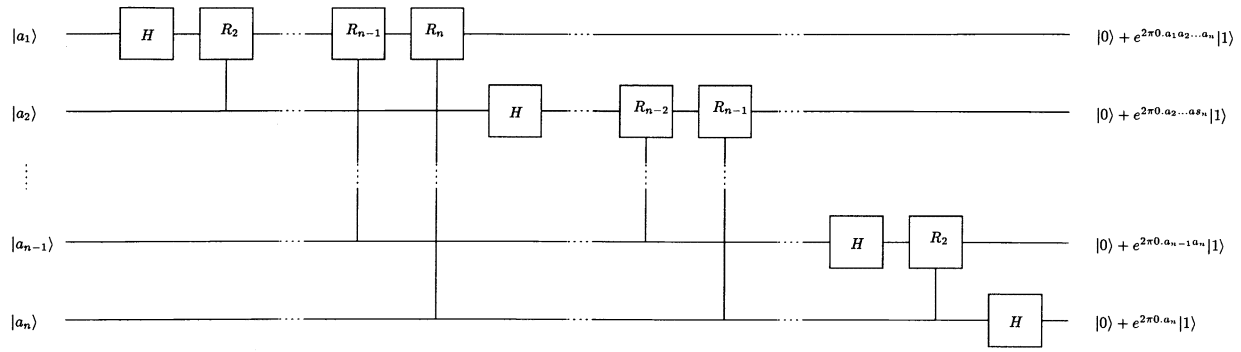

Figure 3. QFT quantum circuit representation

3.4.1. Shor's Algorithm

Shor's factoring algorithm[2] is one of the spectacular applications of quantum computation.

If we know two prime numbers, x and y, then finding the product $N = x * y$ is easy. However, if we only know N, finding the factors x and y is an NP problem. This difficulty in factoring large numbers is the basis of many public key encryption schemes, such as RSA encryption.

To solve this, given N, the naive method of factoring it into its prime factors is to simply starting from 1, and then working upwards until we have found one of the prime factors, which will give us the second one very easily. This requires $\sqrt{N}$ tries for the worst case scenario.

Without going into too much detail and numerical verification, Shor's algorithm is as follows:

1) Find random co-prime number $a < N$

2) Compute $f(x) = a^x \bmod(N)$

3) Find period, r, of $f(x)$. i.e. $f(x + r) = f(x)$

4) x and y are the greatest common denominators of N and $a^{r/2} \pm 1$

For example, Let $N = 15$

1) Choose $a = 7$

2) Compute $f(x) = 7^x$ then $\bmod(15) = 1, 7, 4, 13, 1, 7, 4, 13, ...$

3) Period, $r = 4$

4) $x = GCD(15, 7^{4/2} + 1) = 5$, $y = GCD(15, 7^{4/2} - 1) = 3$

The hard part is step 3, finding r. Using the binary form of $N = 2^n$ It is an order-finding problem of complexity $O(N) = O(2^n)$.

However, through application of quantum Fourier transforms, we can speed up this difficult step exponentially.

1) We start off with initial state $|0\rangle|0\rangle$

2) Create the superposition $\frac{1}{\sqrt{2^t}} \sum_{x=0}^{2^t-1} |x\rangle|0\rangle$

3) Apply an unitary operator such that $|0\rangle \rightarrow |f(x)\rangle$ resulting in $\frac{1}{\sqrt{2^t}} \sum_{x=0}^{2^t-1} |x\rangle|f(x)\rangle$

4) Apply QFT to $|f(x)\rangle$ resulting in $\frac{1}{\sqrt{2^t}\sqrt{r}} \sum_{x=0}^{2^t-1} |x\rangle \sum_{l=0}^{r-1} e^{2\pi i l x/r}|F(l)\rangle$

5) Apply inverse QFT resulting in $\frac{1}{\sqrt{r}} \sum_{l=0}^{r-1} |l/r\rangle|F(l)\rangle$

6) Measure $|l/r\rangle$ to get l/r

7) Obtain r

Steps 4 and 5 are both $O(n^2)$, so factoring has become a polynomial increase problem!

Now, our initial choice of a is very important, and it is easy to see that not all a's give the right result. This is not a major issue, since verification is very easy. For factoring large numbers, this method will find x and y with a probability very close to 1.

3.4.2. Grover's Algorithm

Grover's algorithm[3] provides a less significant speed up to the everyday problem of searching an unsorted database than Shor's algorithm did to factoring large numbers. However, the fact that there is still an improvement demonstrates that quantum computers are not limited to simply one class of problems.

The problem we have here is simple. Given an unsorted database, how are we to find a particular item that we want? On a classical computer, the only method to perform this search is to start from the beginning and check each element one at a time. For $N = 2^n$ elements, this obviously has complexity $O(2^n)$. Through Grover's quantum search algorithm, this complexity is reduced to $O(\sqrt{2^n})$. This is only a quadratic improvement, but is an improvement nonetheless.

This algorithm is an iterative process, with each application of the "Grover iteration" yielding a more refined and better result. Another key feature of the algorithm is the "oracle" which flips an oracle qubit if we have found the correct element, otherwise, it leaves the bit alone. This means that we require two sets of n qubits: one for the actual data and computation, the other as the oracle workspace. The rest of the algorithm is outlined below. See Fig. 4 and Fig. 5:

1) Apply Hadamard to n qubits

2) Apply Grover's Iteration for $\sqrt{N}$ iterations:

2a) Apply Oracle

2b) Apply Hadamard transform

2c) Perform phase shift on the qubits such that $|0\rangle \rightarrow |0\rangle$, $|x\rangle \rightarrow -|x\rangle$ for $x > 0$.

2d) Apply Hadamard transform

This algorithm works by assuming that there is only one correct item in the database. Other techniques have evolved from this base to count the number of correct items in the database[12] and finding matches between two databases.[13]

4. QUANTUM HARDWARE: KANE'S SOLID STATE COMPUTER

The previous section demonstrated that quantum computers can spectacularly improve the efficiency of real-life problems. This section describes one proposal of realising a quantum computer. Ref. [14] gives descriptions of other solid state QC proposals.

4.1. How it works

The Kane proposal[4,5] is a silicon based solid state computer. The qubits are nuclear spins of phosphorus (^{31}P) donor atoms embedded in the silicon lattice (Fig. 6). ^{31}P was chosen because it is very well isolated from the environment with electron relaxation time of thousands of seconds and nuclear spin relaxation times of greater than 10 hours at low temperatures. It is estimated that phonon limited spin relaxation time is of the order of 10^{18} seconds in the milliKelvin range.[15] Also, ^{31}P being approximately the same size as a silicon atom, it can readily replace a silicon atom in the lattice.

^{31}P atoms have five valence electrons. With four of these electrons involved in covalent bonds with the neighbouring silicon atoms, at low temperatures, it has effectively one bounded electron like a hydrogen atom. However, for ^{31}P atoms, there is a two fold electron spin degeneracy at the ground state, and so, using perturbation theory, an external magnetic field, B, needs to be applied to break this degeneracy. The phosphorus nuclear spins are coupled to their electron spins by the hyperfine interaction. The A-Gates above the donor atoms control the strength of the hyperfine interaction, which alters the resonant frequency of the ^{31}P nucleus. This system is a voltage controlled

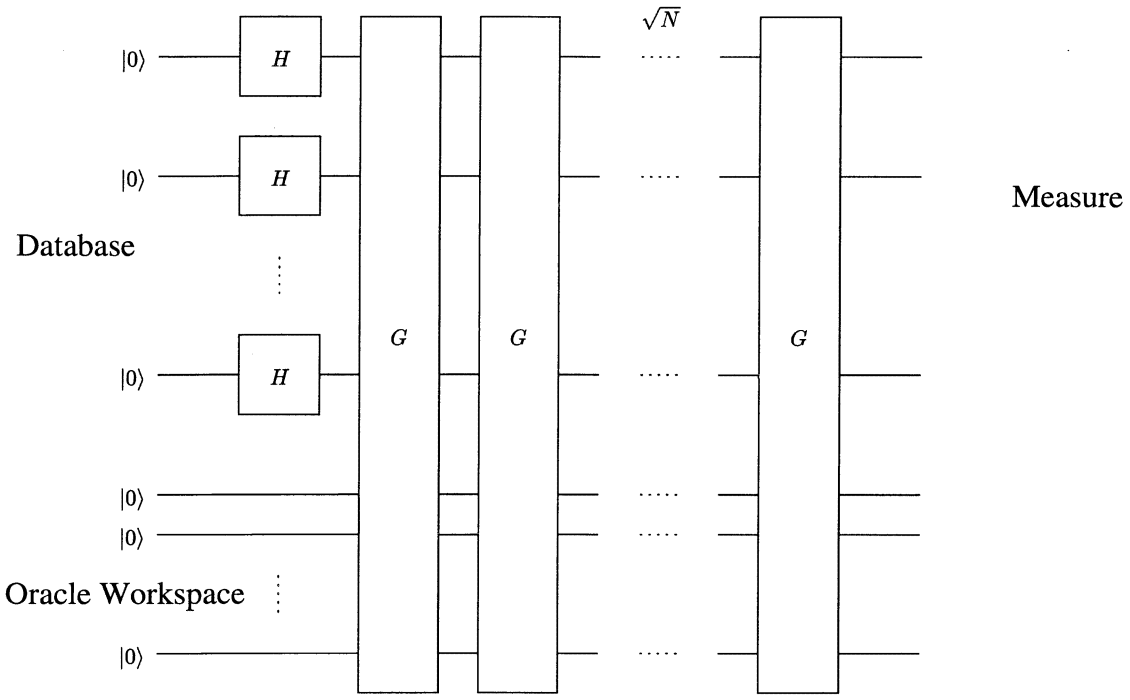

Figure 4. Quantum Circuit for Grover's Algorithm

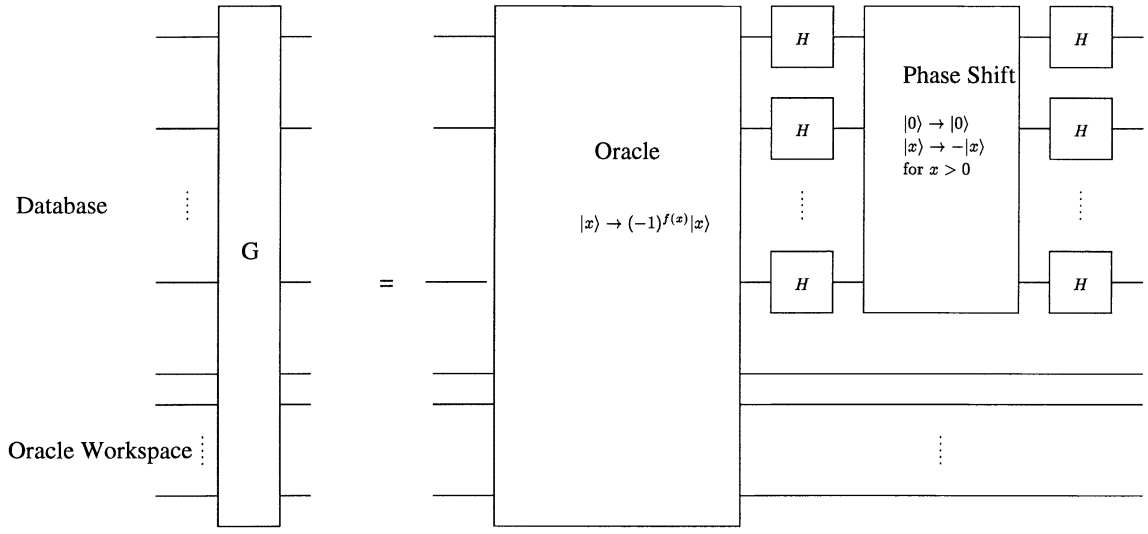

Figure 5. Grover's Iteration

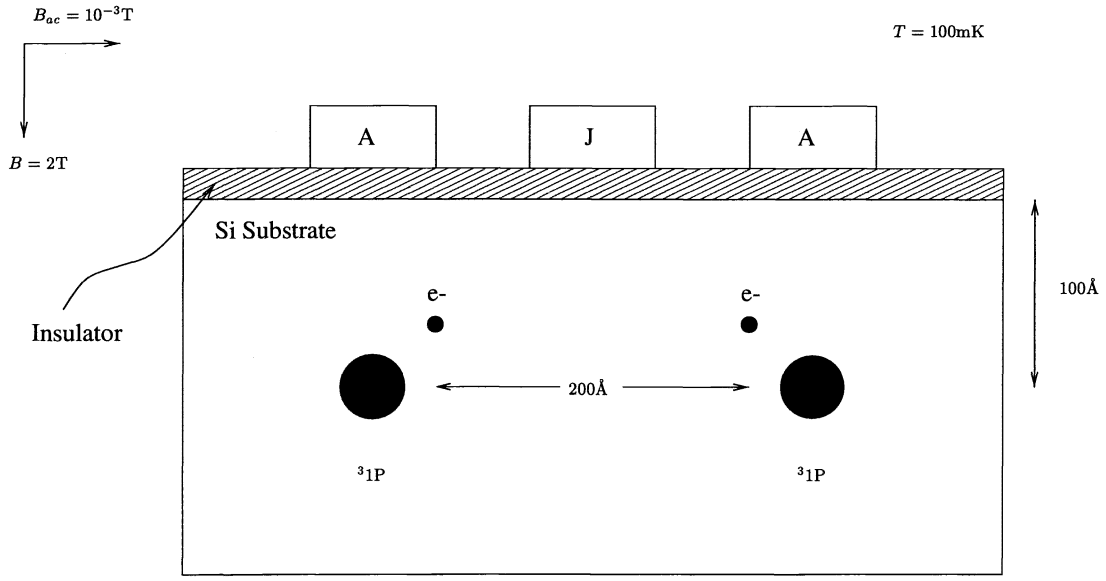

Figure 6. Schematic of Kane's solid-state Si:P quantum computer

oscillator. When a positive voltage is applied, it shifts the electron wave function away from the ^{31}P nucleus towards the gate, thus lowering the resonant frequency, while a negative voltage increases the resonant frequency. Now by applying a global AC magnetic field, B_{ac}, we can arbitrarily rotate the nuclear spin at resonance. Two qubit operations are performed by turning the electron-mediated coupling between two donor atoms on and off using the J-gates positioned between the A-gates.

One proposal for nuclear spin readouts is done by applying different voltages on the A-gates and detecting the resulting electron movement.[4] Pauli exclusion principle dictates that only electrons in opposite spins can occupy the same orbital space, and thus if the electrons are indeed in opposite spins, there should be a detectable electron current from the donor under the negatively biased gate to the positively biased gate. Single Electron Transistors (SET) have been proposed to detect this electron current.[16] The minimum error due to external noise for this proposal is of the order of 10^{-6} per second for typical values of noise.[17]

4.2. Technical Difficulties

So far, we have avoided discussing the practical difficulties in realising the above proposal. The technological challenges involved are currently beyond our capabilities today, but not inconceivably so.

First of all, the silicon host needs to consist of almost completely pure spin zero, charge neutral isotopes. Secondly, the entire slab of Si:P needs to be cooled to the milliKelvin range during operation. This is generally done in a dilution fridge. At high temperatures, the electron and nuclear spins have too many degrees of freedom. We need to reduce phonon-induced decoherence as much as possible. Also, at high temperatures, the electrons will generally not remain bound to their donor nuclei to participate in the electron-nucleus coupling.

The most difficult technological hurdle, however, is in fabrication. The phosphorus donors must be placed in an ordered array, with exactly one donor per array cell. Neighbouring donors are required to be far enough apart so that, under unbiased conditions, their electron wave functions do not overlap, while still close enough for strong coupling between the nuclei during operation. These conditions limit the separations to be approximately 100-200 Å.[4,18] Atom-optics and ultra-high-vacuum scanning tunnelling microscopy are being investigated as possible methods for creating the donor array.[4] After the donors have been placed, the next task is to then bury the donor arrays in more pure silicon. The current practice is to grow the silicon at temperatures of around 600 K, which is unsuitable, considering the donors were placed onto the silicon at 4 K before hand. The high temperature will cause too much movement of the ^{31}P atoms. Once the silicon is somehow grown and the oxide layer has been deposited, the donors will then have to be found again. It has been proposed that a scanned probe with a SET at the tip could be used to locate the hidden donor,[16] a bit like a "metal" detector. However, the $-e$ electronic signal due to the extra electron

could be too weak to be detected under a few hundred Angstroms of silicon, since there is a $+e$ nucleus nearby. After the donors have been located, the A-gates must be constructed directly above the donor nuclei, while still managing to fit a J-gate between them. Thus, the minimum feature size for the construction of the gate should be no more than 0.1μm. This is on the border of the current state-of-the-art technology.

4.3. Future Prospects

The Kane proposal has many advantages over other proposals. It is scalable to large numbers of qubits. It's rate of decoherence is slow enough to allow error correction techniques to be used.

Fabrication technological challenges aside, it can thus be considered a second generation quantum "computer". One that can be used to demonstrate and experiment with more advanced quantum computing methods and algorithms.

ACKNOWLEDGMENTS

We would like to thank Gerard Milburn, Bill Munro, Michael Nielsen of the SRC for Quantum Computer Technology, University of Queensland for their incredible patience and invaluable help in the preparation of the material for this paper. Joseph would also like to thank Geraldine Yam for all the emotional support during this sometimes rather lonely work.

REFERENCES

1. D. Deutsch, "Quantum theory, the Church-Turing principle and the universal quantum computer," *Proc. Roy. Soc. London A* **400**, pp. 97–117, 1985.
2. P. Shor, "Algorithms for quantum computation: Discrete logarithms and factoring," in *Proc. 35th Annual Symposium on the Foundations of Computer Science*, S.Goldwasser, ed., pp. 124–134, IEEE Computer Society Press, (Los Alamitos, California), Nov 1994. Updated version see LANL preprint quant-ph/9508027.
3. L. Grover, "Quantum mechanics helps in searching for a needle in a haystack," *Phys. Rev. Lett.* **79**(2), pp. 325–328, 1997. Updated version see LANL preprint quant-ph/9706033.
4. B. Kane, "A silicon-based nuclear spin quantum computer," *Nature* **393**, pp. 133–137, 1998.
5. B. Kane, "Silicon-based quantum computation," 2000. LANL Preprint quant-ph/0003031.
6. S. Bandyopadhyay, V.P.Roychowdhury, and D.B.Janes, "Chemically self-assembled nanoelectronic computing networks," *Int. Journal of High Speed Elec. and Sys.* **9**(1), pp. 1–35, 1998.
7. P. Packan, "Pushing the limits," *Science* **285**, pp. 2079–2081, Sep 1999.
8. J. Cirac and P.Zoller, "Quantum computation with cold trapped ions," *Phys. Rev. Lett.* **74**(4091), 1995.
9. N. Gershenfeld and I.L.Chuang, "Bulk spin-resonance quantum computation," *Science* **275**(5298), pp. 350–356, 1997.
10. Y. Nakamura, Y.A.Pashkin, and J.S.Tsai, "Coherent control of macroscopic quantum states in a single-Cooper-pair box," *Nature* **398**, p. 786, April 1999. See also LANL Preprint cond-mat/9904003.
11. M. Nielsen and I.L.Chuang, *Quantum Computation and Quantum Information*, Cambridge University Press, 2000.
12. G.Brassard, P.Hoyer, and A.Tapp, "Quantum counting," in *25th Int. Coll. on Automata, Languages and Programming*, pp. 820–831, 1998. See also LANL Preprint quant-ph/9805082.
13. M. Heiligman, "Finding matches between two databases on a quantum computer," 2000. LANL preprint quant-ph/0006136.
14. G.P.Berman, G.D.Doolen, and V.I.Tsifrinovich, "Solid-state quantum computation - a new direction for nanotechnology," *Superlattices and Microstructures* **27**(2), 2000.
15. J.S.Waugh and C.P.Slichter, "Mechanism of nuclear spin-lattice relaxation in insulators at very low temperatures," *Phys. Rev. B* **37**, pp. 1068–1083, 1988.
16. B.E.Kane, N.S.McAlpine, A.S.Dzurak, R.G.Clark, G.J.Milburn, H.B.Sun, and H.Wiseman, "Single spin measurement using single electron transistors to probe two electron systems," *Phys. Rev. B* **61**(4), pp. 2961–2972, 2000. See also LANL Preprint cond-mat/9903371.

17. G. Berman, D.K.Campbell, G.D.Doolen, and K.E.Nagaev, "Dynamics of the measurement of nuclear spins in a solid-state quantum computer," *Journal of Physics - Condensed Matter* **12**(13), pp. 2945–2952, 2000. See also LANL preprint cond-mat/9905200.
18. H.-S. Goan and G. Milburn, "Silicon-based electron-mediated nuclear spin quantum computer." Unpublished. See http://www.physics.uq.edu.au/quant_comp_tech/qct-pubs.htm, 2000.