Quantum Computing: A High-Level Overview

James Walker

CS5431

1. Introduction

Quantum computation is currently one of the most exciting developing fields, combining concepts of quantum physics and computer science with the promise to deliver fundamentally different, radically more powerful computers of the future. However, this field is mired in mystery, with many fundamental questions still unanswered and barely-functional prototypes comprising the state of the art. Furthermore, most of the literature on quantum computation lies buried under esoteric terminology and highly advanced mathematics, making it difficult for laymen to acquire a basic understanding of the field. This survey aims to make the basic concepts of quantum computers accessible to the uninitiated.

The paper is organized as follows. Section 1 provides an overview of why quantum computation is important and gives a brief history of its development. Section 2 is dedicated to quantum "software," explaining basic quantum mechanical properties important to quantum computation and explaining why quantum algorithms are fundamentally different from their classical counterparts. Section 3 is dedicated to quantum "hardware," describing five major categories of quantum computer designs currently under development. Section 4 presents a brief outlook of quantum computing. Finally, Section 5 briefly summarizes the material described in the previous sections.

1.1. Overview

The promise of quantum computation is that quantum computers possess the ability to efficiently execute algorithms which are so difficult for classical computers that executing them becomes impractical even for very small inputs. These are not merely algorithms of theoretical interest only, but problems of great practical import such as the widely-used, integer factorization-based RSA cryptosystem.

Yet despite the great promise of this field, many aspects of quantum computation remain a mystery. For example, it is unknown what class of problems quantum computers can solve more efficiently than their classical counterparts. Quantum computers are not universally superior to classical computers, for there are many algorithms which are difficult for classical computers which have been shown to be equally difficult for quantum computers. Conversely, there are some problems which were previously believed to be solvable in practical time only by quantum computers, but efficient classical algorithms were eventually discovered, disproving quantum superiority for those problems. In fact, it is still not even well-understood *why* quantum computers are more powerful than their classical counterparts! [21]

These mysteries and setbacks notwithstanding, there are some quantum algorithms which *are* provably superior to the best possible classical counterparts, and still more quantum algorithms for which no efficient classical analogue is known to exist despite extensive research. This proven computational superiority, even if its full scope remains uncertain, has been sufficient to spur tremendous interest and research in this field.

1.2. A Brief History

In a keynote speech given in 1981, Feynman [7] made several interesting observations about the relationship between physics and computers. He was interested in the ability of a computer to simulate physical systems without requiring an exponential increase in the size of the computer's resources as compared to the physical system being simulated. He observed that, because classical physics is local, causal, and reversible, there is in principle nothing to prevent a computer from performing extremely accurate simulations of classical physical systems with only a linear increase in computing resources compared to the system being simulated.

Feynman went on to note that quantum systems, as contrasted with classical systems, are probabilistic rather than causal, and that due to this property, there is no way to avoid an exponential increase in computational resources required to accurately simulate a quantum system. He showed that attempting to reduce the nondeterministic nature of quantum mechanics to an equivalent classical probability computation gives invalid results. The conclusion is that, in order to simulate quantum mechanical effects with only a linear increase in computational resources, the computer itself must be constructed from quantum components.

This observation leads to the implication that a computer built from quantum components—a quantum computer—must have superior computational characteristics to classical computers. In particular, a quantum computer may be able to solve some kinds of problems that exhibit exponential growth in the problem state space without exponential growth in the time taken to complete the computation.

This observation can be inverted to state that, in a quantum computer, available parallelism increases exponentially with a linear increase in computing resources. Deutsch and Jozsa [4] called this characteristic *quantum parallelism.* They observed that when a classical computer performs a computation, it in essence computes a function; the same set of inputs will always result in the same output. Quantum computers, however, operate in a fundamentally different way because they can generate many output values from a single input; that is, they can compute all potential solutions to a calculation simultaneously.

For some time, the computational potential of quantum computers remained only theoretical. However, in 1994, Shor [28] kindled interest in quantum computation by publishing an algorithm that can factor integers in polynomial time using a quantum computer. He further expanded on this work three years later [29]. (See Section 2.3 for a high-level overview of how Shor's algorithm works.)

This work was groundbreaking because there is no known classical algorithm that can efficiently perform integer factorization in the general case. Shor's algorithm provided compelling evidence that quantum computers may be inherently more powerful than their classical counterparts. In addition, his algorithm is of great practical interest because the widely-used RSA cryptosystem is dependent on the presumed difficulty of factoring integers. Should a functional quantum computer fall into malicious hands, a global security crisis would result.

Since the publication of Shor's algorithm, many more quantum algorithms have been published. One example is Grover's search algorithm [10]. In classical computation, searching an unsorted list cannot be done in less than O(n) time; however, Grover's algorithm can perform the search in time $O(n^{1/2})$. Two other examples include a quantum algorithm for efficiently estimating Gauss sums [30] and for efficient Fourier fishing and Fourier checking [1]. Many other examples exist; these are chosen simply to give the flavor of problems that can be handled more efficiently with quantum computation.

2. The Method

Quantum computation promises the ability to efficiently execute certain kinds of algorithms which are so difficult for classical computers that executing them becomes impractical even for small inputs. However, significant challenges stand in the way of practical implementations of quantum computing. These challenges include the exotic hardware required to construct quantum computers and the difficulty of designing and executing quantum algorithms in the first place. We will first consider the latter of these two challenges.

2.1. Important Quantum Mechanical Concepts

The first obstacle that must be dealt with in quantum computing is extracting the result of the computation. One of the fundamental characteristics of quantum mechanics that makes quantum computation efficient is the fact that a property of physical magnitude (such as angular momentum) can be in a *superposition;* that is, it can be in all of its possible states simultaneously. In the case of a quantum bit, or *qubit,* this means that rather than having a value of either 0 or 1, it may be both a 0 and a 1 at the same time (by expressing the difference between 0 and 1 in terms of some physical property). It is due to this characteristic that quantum computers can calculate an exponential combination of possible values in the problem state space simultaneously.

However, this condition of superposition is fragile. Interaction with external forces can easily cause a qubit to *decohere* such that it loses its simultaneous state and settles into a 0 or 1. One way to describe this effect is that while a qubit is in a state of superposition, there is a given probability when it interacts with outside forces that it will decohere into a 0 or 1. It is impossible to know with certainty ahead of time what its value will be. However, once it is read, its state collapses such that from that point forward, whenever the qubit is read, it always returns the same value with a probability of 1 (until it is reset to begin a new computation). Because decoherence occurs easily, one of the principle challenges of constructing a functional quantum computer is keeping the qubits isolated until it is time to read their value. This challenge is algorithmic as well as physical.

Furthermore, simply isolating the qubits until the user is ready to read them is not sufficient. If the user reads the qubits in a naïve manner, they will return a random result from all of their possible combinations of values. Clearly, obtaining a random output is not a useful outcome when trying to efficiently compute valid answers to difficult problems. It is necessary to manipulate the qubits in such a way that they return the desired result when they are read. This further increases the difficulty of designing quantum computational algorithms.

Because of these characteristics, many existing quantum algorithms are not deterministic, but rather return the correct answer with a high (but less than 1) probability. Although this is not an entirely desirable state of affairs, most answers can be checked for correctness quickly, and if the quantum algorithm returned an incorrect result, it can simply be run again. If the quantum algorithm is exponentially faster than its classical counterpart, as is often the case, this process is still tremendously faster than running a classical algorithm.

A clever designer may formulate a possible workaround that involves copying qubits; e.g., by copying a qubit and taking two separate readings of it. Unfortunately, such an approach cannot work because quanta in superposition cannot be cloned. [32] This is an important and significant limitation since it means quantum data cannot be directly copied until after it has decohered, which is no different

from copying classical data.

The situation is particularly difficult because even the best present quantum computer designs cannot fully prevent some amount of decoherence from entering the system. This necessitates the inclusion of quantum error correction. In principle, quantum error correction is similar to classical error correction in that it relies on multiple redundant bits to restore a corrupted state, but it is complicated by the fact that state cannot be copied directly. It is possible to apply modified classical error correction techniques to the quantum case using special quantum state transformations. [16][25]

Another important principle of quantum mechanics is *entanglement.* Entanglement is the phenomenon whereby the states of two quanta—for our purposes, two qubits—are not mutually independent. In other words, if the state of one entangled qubit is manipulated, it has an effect on the state of another entangled qubit. For example, there may be a case where reading a 1 from the first qubit guarantees that a 1 will be read from the second qubit. Mathematically, this property can be expressed by saying that entangled states cannot be written as a tensor product of their individual component states. Entangling and disentangling qubits is a critical operation in quantum computers. [2] [25]

2.2. Reversible Computing & Quantum Gates

Another concept important to quantum computation is reversible computing. Classical computing is non-reversible, meaning that inputs cannot be computed based on the outputs. For example, if the output of a two-bit AND gate is 0, it is impossible to know whether the inputs were 00, 01, or 10. The only classical logic gates which are reversible are NOT and XOR.

A reversible computer, by contrast, is composed exclusively of reversible logic gates. It is possible to construct a variety of reversible gates which are collectively capable of performing the same kinds of computations as non-reversible gates.

This concept is vital to quantum computing because all quantum transformations are unitary, and therefore reversible. Thus, all quantum gates themselves must be reversible. This further complicates the design of quantum algorithms, since users only familiar with classical programming encounter a steep learning curve when they must design algorithms that work exclusively with reversible computations.

Note that the use of the term “gates” when describing quantum gates should be taken conceptually. As we will see, transformations on qubits are not necessarily applied with gates in the conventional sense. Note also that because of the superposition phenomenon, qubit states are expressed not as bits but as matrices of bits. Therefore, quantum gates actually perform transformations on matrices.

The simplest non-trivial quantum logic gate is a controlled-NOT gate. The quantum cNOT gate can be used as a basis to create more general quantum gates. Quantum logic gates can be used to apply unitary transformations to the state of qubits (their probability matrices) without causing them to decohere, and even to entangle and disentangle qubits. Quantum parallelism arises from the fact that such transformations are applied to all basis vectors in the qubit's superposition simultaneously.

For many years, it was believed that being able to perform these kinds of operations was critical to perform useful quantum computations and read meaningful results afterwards. [2] However, it has recently been discovered these these superposition-preserving unitary transformations are not always necessary. The newest developments in quantum algorithms have discovered that it is possible to

extract useful results through measurement alone, by reading the correct qubits in the correct order with the correct measurement techniques. [21]

Even with these latest simplifying techniques, however, a solid understanding of the quantum concepts just discussed is necessary to design working algorithms. The details of these algorithms are highly complex, drawing on multiple fields of advanced mathematics, and as such lie outside the scope of a brief survey paper.

2.3. Shor's Algorithm

We conclude this section with a high-level overview of Shor's algorithm (see Section 1.2) to give the flavor of quantum algorithm designs. [28][29] Shor's algorithm presents an efficient general method for factoring integers. The algorithm consists of two parts. The first part transforms the problem of factoring to the problem of finding the period of a function. This part can be done with a classical computer and so is not treated here.

The second part of the algorithm proceeds to actually find the period of the function, and this is where the power of quantum computation is utilized. Note that the following summary uses bra-ket notation, which is a commonly-used shorthand for expressing vectors in quantum mechanics:

1. First, initialize the registers to a superposition $Q^{-1/2}\Sigma(x=0, Q-1)\ |x\rangle\ |0\rangle$ where, given N (the integer to be factored), $Q = 2^q$ such that $N^2 <= Q < 2N^2$.

2. Construct a function $f(x) = a^x \bmod N$ where a is a randomly chosen number < N and apply it to the state from step 1 to obtain $Q^{-1/2}\Sigma(x)\ |x\rangle\ |\ f(x)\rangle$.

3. Apply a quantum Fourier transform to obtain the state $Q^{-1}\Sigma(x)\Sigma(y)\omega^{xy}\ |y\rangle\ |f(x)\rangle$.

4. Perform a quantum measurement to obtain a value y and perform continued fraction expansion on y/Q to produce some c/r' that satisfies r' < N and $|y/Q - c/r'| < 1/2Q$, with the result that r' is the correct period r with high probability.

5. Check if r' satisfies the problem. If not, try additional candidates with values near y or multiples of r'.

6. If the problem is still not satisfied, return to step 1 and repeat.

3. Implementations

3.1. Considerations

There are many engineering challenges to building quantum computers. As section 2.1 discussed, qubits must be prevented from decohering prematurely. In part, this is accomplished by minimizing qubits' interaction with the external world. However, even the best-designed system cannot completely

prevent some entropy from entering the system, just as wireless communications are always subject to some amount of thermal noise. Following initialization by cooling the qubits to a low-entropy state, quantum computers should be able to perform their computations quickly enough to output an answer before thermal noise or other factors cause premature decoherence. This cannot always be prevented, so error-correcting mechanisms must be present in the system; e.g., a set of redundant qubits whose sole purpose is to restore state if it becomes corrupted before computation is complete.

Additionally, quantum computers need resources for manipulating their state with power equivalent to a Turing machine in order to be considered universal computers. This may be accomplished with quantum analogues to logic gates, which as we will see, can take many forms. However, quantum computers need not be built with gates. They can also be constructed as *adiabatic* or *cluster-state* machines. In adiabatic systems, the answer to a computation is defined as the ground state of a network of qubit interactions, then evolves the qubits into the ground state by activating these interactions in sequence. In this case, it must be shown that the available interactions are sufficiently complex to allow universal computation, and various support systems (such as cooling) must be scalable as described previously. In cluster-state systems, a particular quantum state is created by manipulating qubits with a small set of non-universal gates, then universality is achieved by altering how the ensuring measurements are taken. Properly built adiabatic and cluster-state systems are equivalent in power to gate-based quantum computers, and are simpler to implement with certain kinds of technologies. [16]

Lastly, any quantum computer design must be scalable. The entire point of quantum computers is that, for certain kinds of problems, exponential speedup can be achieved with a linear increase in resources. In order to preserve the value of quantum computation, any physical implementation must respect this restriction. While qubits inherently possess the quality of quantum parallelism, they are not the only component of a physical system. As just discussed, the system also requires a way to isolate qubits from the external world to prevent decoherence, to restore corrupted state, to cool the qubits to a low-entropy state in order to initialize them for computation, and resources to manipulate and measure qubits. These resources are frequently larger by many orders of magnitude than the ultra-miniaturized components of classical computers; thus if the system is to be scalable, all of these resources must grow linearly with the number of qubits.

Having established these restrictions, we now turn to five major categories of physical quantum computers that have been implemented to date.

3.2. Atom Trap Computers

Atom trap designs use individual atoms as qubits. These employ particularly exotic hardware designs, using electric fields to keep atoms suspended with nanometer precision in a vacuum at near absolute zero temperatures. As one might expect, these extreme conditions are effective in isolating the atoms from external influences, and the qubits generally maintain coherence far longer than is needed to complete quantum calculations—a major strength of this technology. [25][16]

A common subdesign in this category is the ion trap computer. In this design, lasers act as logic gates. By applying lasers to the ions, transformations of each qubit's quantum state can be accomplished, and qubits can also be entangled with each other. As of March 2011, 14 qubits have been successfully entangled using this technology [20]. Initialization is accomplished via optical pumping, a process wherein light is used to raise the energy level of electrons, coupling the target ion to excited states which eventually decay to a single state. The ions are measured by applying a laser, which will

either cause the ion to emit photons if the ion has collapsed to a 1 state, or to emit nothing if it has collapsed to 0. These designs suffer from scalability issues because susceptibility to decoherence increases as more ions are added [17] and crosstalk can decrease the effectiveness of using the lasers as logic gates [31], but recent research has sought to address this problem by physically shuttling ions around within their containing electric field [13].

An alternative to ion traps is to use neutral atoms. In this design, arrays of atoms are confined using an “optical lattice” of crossed laser beams [5][19]. Qubits can be made to interact by bringing adjacent atoms together and causing entanglement through contact interactions. The primary challenge with this design is controlling the initialization, interaction, and measurement of the qubits [16].

3.3. Nuclear Magnetic Resonance Computers

Nuclear magnetic resonance (NMR) is a phenomenon wherein nuclei in a magnetic field absorb and emit radiation. NMR is utilized for studying quantum mechanical effects and molecular physics, and is also utilized in magnetic resonance imaging (MRI). NMR is already a relatively mature technology, so in 1996, methods were proposed to construct quantum computers using NMR technology as a baseline. [16][25][3][9]

NMR quantum computers can be broadly divided into two categories, solid-state and liquid-state. In both cases, NMR computers use entire molecules as qubits, utilizing their overall molecular spin to differentiate state. Unfortunately, a poor signal-to-noise ratio inhibits the scalability of NMR designs, and liquid-state NMR designs have failed to exhibit quantum entanglement, preventing true quantum computation. Ironically, despite using the most mature technology of all current quantum computer designs, NMR computers are more likely to be of use for advancing other quantum technologies rather than developing large-scale NMR computers. [16]

3.4. Photonic Computers

This class of quantum computers uses photons as its constituent state units; e.g., by streaming photons across a chip using double refraction. A strength of photonic designs is that photons are relatively resistant to decoherence; conversely, achieving interactions that enable universal logic is relatively difficult using this technology. [16]

In 2001, Knill et al. [15] showed photon-based quantum computers to be scalable by demonstrating that it can be done with single-photon sources and detectors and linear optical circuits. However, present designs employ nondeterministic interactions, reducing their usefulness; research is ongoing to enable deterministic interactions. [26][6] Using a cluster-state design as described in section 3.1, simple quantum algorithms have been demonstrated with photon systems [24], and theoretical reductions in resource overhead have been proposed [23]. Current circuits employ logic gates about one centimeter in size—many orders of magnitude larger than classical counterparts, yet because quantum computers gain exponential computational power with a linear increase in physical resources, this is small enough for practical use. [24][18]

3.5. Quantum Dot Computers

Quantum dots are very small crystals that act as semiconductors with electric characteristics closely tied to the size and shape of the crystal. Quantum dot computing explores the use of quantum dots as qubits. The flow of electrons through quantum dots can be precisely controlled, enabling accurate measurements of spin and other properties. As with other quantum computer technologies, various subcategories exist; these include electrostatically defined dots as well as self-assembling dots, the latter of which exhibit random construction characteristics. [16] Additionally, various methods for achieving universal computation using quantum dots have been proposed; for example, Loss and DiVincenzo proposed quantum dots each containing a single electron. In this design, the electrons themselves act as qubits, with their current state defined by their spin. Logical transformations are achieved by changing voltages on electrostatic gates, activating and deactivating interactions by moving electrons closer and farther apart. [11]

Electrostatically defined and self-assembling quantum dots exhibit distinct weaknesses. Electrostatic quantum dots are hindered by extremely short-range exchange interaction, which is a major constraint on implementing fault-tolerant quantum error correction. [16] In the case of self-assembling quantum dots, their main problem is their randomness, as they form in random locations and do not possess uniform optical characteristics. Advanced fabrication techniques are being explored to apply limited control to the dots' behavior [14][8], or even to enable deterministic placement of dots [27], which may alleviate this problem. Conversely, quantum dot computers exhibit the ability to be controlled in a matter of picoseconds per operation, which shows a potential for extremely fast computation.

3.6. Superconductor Computers

Classical integrated circuits suffer from high power leakage. Because of this characteristic, if an attempt were made to use them as quantum circuits, decoherence would occur far too quickly to allow any useful computation. However, this decoherence is much less severe in superconductors at low temperatures. Therefore, attempts have been made to construct quantum circuits using this technology, and they have the advantage that they can be fabricated using existing methods. Of all quantum computer designs, superconductor qubits have the closest physical resemblance to classical bits. They are built from circuits with a Josephson junction, a thin insulating layer separating sections of a superconductor, as the critical component. The flow of electrons across the Josephson junction gives rise to physical properties that make the circuit suitable for use as a qubit. [16]

In this scheme, basic quantum logic gates are created by having adjacent qubits couple either capacitively or inductively. However, this mechanism is not very adjustable. There has been research into activating and deactivating interactions via adjustable couplers [22], and the possibility has been explored of using this technique to achieve adiabatic quantum computation with superconductors [12].

Initially, it was believed that the macroscopic nature of superconducting qubits, which utilize about 10^{10} conduction electrons, would lead to impractically swift decoherence times. Indeed, in early experiments, quantum superconductors experienced decoherence times measured in nanoseconds. More recently, however, decoherence times have been extended to several microseconds, which is longer than a superconductor's initialization, logic, and measurement times by one to two orders of magnitude. Nonetheless, fighting rapid decoherence is currently the greatest obstacle in implementing

practical superconductor quantum computers, and microscopic materials engineering will likely be required to further reduce decohering noise. [16]

4. Outlook

At this point, there are many unknowns in the field of quantum computation, which makes future prediction difficult. The limitations of quantum computation have not yet been demonstrated. However, it has been shown that quantum computers are demonstrably more efficient than classical computers for some problems, so it is reasonable to assume that more such problems will continue to be discovered.

Likewise, it is difficult to predict which of the current hardware paradigms—if any—will yield a scalable design. A significant problem is that maintaining coherence and manipulating qubits are two necessary components of all designs which are fundamentally at odds with each other. The least promising of the current technologies is nuclear magnetic resonance, so it is unlikely that this path will yield practical quantum computers. Atom trap designs are effective at maintaining coherence and have succeeded in entangling relatively large numbers of simultaneous qubits. Conversely, quantum dot designs have the advantage of extremely fast “cycle times” (measured in picoseconds), while superconductors have the advantage that they can be fabricated using well-understood methods (but they struggle with rapid decoherence). It could be argued that atom trap computers have achieved the most promising results to date.

Because quantum computers are difficult to scale and there are many common applications for which classical computers are just as effective, it is possible that quantum computers might never achieve widespread consumer use. Additionally, designing quantum algorithms requires not only knowledge of computer programming, but a thorough understanding of quantum physics. Average programmers—or indeed, even many very good programmers—are unlikely to have the necessary training, so software engineering for quantum computers will be restricted to programmers who are also accomplished physicists. As the technology matures, quantum computers may become useful as expensive specialty computation machines used mostly by research laboratories or large corporations, similar to mainframes. On the other hand, the research is still immature, so it is still possible that a breakthrough enabling practical consumer use could be discovered.

5. Conclusion

Quantum computation promises the ability to compute solutions to problems that, for all practical purposes, are insoluble by classical computers. Some quantum algorithms have been shown to be more efficient than any possible classical alternative, and some of these algorithms have been implemented on small-scale prototype hardware. However, the quantum promise is still a long way from achieving practical realization. The same properties of quantum mechanics that enable quantum computers' superior performance also make the design of quantum algorithms and the construction of functional hardware extremely difficult.

We then described five broad categories of quantum computer hardware designs. These include photon, atom trap, nuclear magnetic resonance (NMR), quantum dot, and superconductor-based designs. Each of these designs exhibits varying strengths and weaknesses, and while many of them

show promise, none have yet succeeded in implementing large-scale quantum computers. It is difficult to predict which of these designs, if any, will be the first to successfully implement a practical quantum computer. However, with the promise of radically more powerful computation waiting just beyond the horizon, it is safe to say that research will continue apace until the dream of a large-scale quantum computer capable of executing previously impractical algorithms has been realized, or else proven to be unachievable.

References

[1] Aaronson, S. (2009). BQP and the Polynomial Hierarchy. ArXiv preprint. Available: http://arxiv.org/abs/0910.4698

[2] Barenco, A., Deutsch, D., Ekert, A., & Jozsa, R. (1995). Conditional quantum dynamics and logic gates. *Physical Review Letters, 74*(20), 4083-4086.

[3] Cory, D.G., Fahmy, A.F., & Havel, T.F. (1997). Ensemble quantum computing by NMR-spectroscopy. *Proceedings of the National Academy of Sciences USA* 94, 1634-1639.

[4] Deutsch, D. & Jozsa, R. (1992). Rapid solution of problems by quantum computation. In *Proceedings of the Royal Society of London Series A A439,* 553-558.

[5] DiVincenzo, D.P. (2000). The physical implementation of quantum computation. *Fortschritte der Physik* 48, 771-783.

[6] Duan, L. M. & Kimble, H. J. (2004). Scalable photonic quantum computation through cavity-assisted interactions. *Physical Review Letters 92,* 127902.

[7] Feynman, R. (1982). Simulating physics with computers. *International Journal of Theoretical Physics 21,* 6&7, 467-488.

[8] Fushman, I., D., Faraon, A., Stoltz, N., Petroff, P., & Vučković, J. (2008). Controlled phase shifts with a single quantum dot. *Science 320,* 769–772.

[9] Gershenfeld, N.A. & Chuang, I.L. (1997). Bulk spin resonance quantum computing. *Science 275,* 50-356.

[10] Grover, L.K. (1996). A fast quantum mechanical algorithm for database search. In *Proceedings of the Twenty-Eight Annual ACM Symposium on the Theory of Computing* (Philadelphia, Pennsylvania, 22-24 May 1996), 212-219.

[11] Hanson, R., Kouwenhoven, L. P., Petta, J. R., Tarucha, S. & Vandersypen, L. M. K. (2007). Spins in few-electron quantum dots. *Reviews of Modern Physics 79*, 1217–1265.

[12] Harris, R. et al. (2009). Experimental demonstration of a robust and scalable flux qubit. ArXiv preprint. Available: http://arxiv.org/abs/0909.4321

[13] Home, J. P., Hanneke, D., Jost, J. D., Amini, J. M., Leibfried, D., & Wineland, D. J. (2009). Complete methods set for scalable ion trap quantum information processing. *Science* 325, 1227-1230.

[14] Kistner, C., Heindel, T., Schneider, C., Rahimi-Iman, A., Reitzenstein, S., & Forchel, A. (2008). Demonstration of strong coupling via electro-optical tuning in high-quality QD-micropillar systems. *Optics Express* 16, 15006–15012.

[15] Knill, E., Laflamme, R. & Milburn, G. J. (2001). A scheme for efficient quantum computation with linear optics. *Nature* 409, 46–52.

[16] Ladd, T. D., Jelezko, F., Laflamme, R., Nakamura, Y., Monroe, C., & O'Brien, J. L. (2010). Quantum computers. *Nature*, *464*(7285), 45-53.

[17] Leibfried, D., Blatt, R., Monroe, C., & Wineland, D. (2003). Quantum dynamics of single trapped ions. *Reviews of Modern Physics* 75, 281-324.

[18] Matthews, J. C. F., Politi, A., Stefanov, A. & O'Brien, J. L. (2009). Manipulation of multiphoton entanglement in waveguide quantum circuits. *Nature Photonics 3,* 346–350.

[19] Mizel, A. Lidar, D. A. & Mitchell, M. (2007). Simple proof of equivalence between adiabatic quantum computation and the circuit model. *Physical Review Letters* 99, 070502.

[20] Monz, Thomas. (2011). 14-Qubit Entanglement: Creation and Coherence. *Physical Review Letters* 106, 130506.

[21] Nielsen, M.A. & Chuang, I.L. (2010). *Quantum Computation and Quantum Information 10th Anniversary Edition.* Cambridge University Press.

[22] Niskanen, A. O., Harrabi, K., Yoshihara, F., Nakamura, Y., Lloyd, S., & Tsai, J. S. (2007). Quantum coherent tunable coupling of superconducting qubits. *Science* 316, 723-726.

[23] O'Brien, J. L. (2007). Optical quantum computing. *Science 318,* 1567–1570.

[24] Politi, A., Matthews, J. C. F. & O'Brien, J. L. (2009). Shor's quantum factoring algorithm on a photonic chip. *Science 325,* 1221.

[25] Rieffel, E., & Polak, W. (2000). An introduction to quantum computing for non-physicists. *ACM Computing Surveys*, *32*(3), 300-335.

[26] Schmidt, H. & Imamoglu, A. (1996). Giant Kerr nonlinearities obtained by electromagnetically induced transparency. *Optics Letters 21,* 1936–1938.

[27] Schneider, C., Straub, M., Sunner, T., Huggenberger, A., Wiener, D., Reitzenstein, S., ... & Forchel, A. (2008). Lithographic alignment to site-controlled quantum dots for device integration. *Applied Physics Letters* 92, 183101.

[28] Shor, P.W. (1994). Algorithms for quantum computation: Discrete log and factoring. In *Proceedings of the 35th Annual Symposium on Foundations of Computer Science* (Nov. 1994), pp. 124-134.

[29] Shor, P.W. (1997). Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. *Society for Industrial and Applied Mathematics Journal on Computing 26,* 5, 1484-1509. Expanded version of [28].

[30] van Dam, W. & Seroussi, G. (2002). Efficient Quantum Algorithms for Estimating Gauss Sums. ArXiv preprint. Available: http://arxiv.org/abs/quant-ph/0207131

[31] Wineland, D. J., Monroe, C., Itano, W. M., Leibfried, D., King, B. E., & Meekhof, D. M. (1998). Experimental issues in coherent quantum-state manipulation of trapped atomic ions. *Journal of Research of National Institute of Standards and Technology* 103, 259-328.

[32] Wootters, W.K. & Zurek, W.H. (1982). A single quantum cannot be cloned. *Nature 299,* 802.