ISSN No. 0976-5697

Volume 3, No. 3, May-June 2012

International Journal of Advanced Research in Computer Science

RESEARCH PAPER

Available Online at www.ijarcs.info

Quantum Computing: A Future Trends in Computing

Amit V.Pandhare
Computer Science & Engineering Department
PRMIT&R, Badnera
Badnera-Amravati, India.
amitvpandhare@gmail.com

Syed Tanzeem
Computer Science & Engineering Department
PRMIT&R, Badnera
Badnera-Amravati, India.
tanzeem321@gmail.com

Sunil Gupta
Computer Science & Engineering Department
PRMIT&R, Badnera
Badnera-Amravati, India.
sunilguptacse@gmail.com

Abstract: If the bits of computers are someday scaled down to the size of individual atoms, quantum mechanical effects may profoundly change the nature of computation itself. The wave function of such a quantum computer could consist of a superposition of many computations carried out simultaneously; this kind of parallelism could be exploited to make some important computational problems, like the prime factoring of large integers, tractable. However, building such a quantum computer would place undreamed of demands on the experimental realization of highly quantum-coherent systems; present-day experimental capabilities in atomic physics and other fields permit only the most rudimentary implementation of quantum computation.

Keywords: Quantum Computing, Qubit, Bloch Spher, Quantum Circuits, Quantum Gates, Shor's algorithm.

I. INTRODUCTION

A quantum computer is a device for computation that makes direct use of quantum mechanical phenomena, such as superposition and entanglement, to perform operations on data. Quantum computers are different from digital computers based on transistors. Whereas digital computers require data to be encoded into binary digits (bits), quantum computation utilizes quantum properties to represent data and perform operations on these data.

Civilization has advanced as people discovered new ways of exploiting various physical resources such as materials, forces and energies. In the twentieth century information was added to the list when the invention of computers allowed complex information processing to be performed outside human brains. The history of computer technology has involved a sequence of changes from one type of physical realization to another — from gears to relays to valves to transistors to integrated circuits and so on. Today's advanced lithographic techniques can squeeze fraction of micron wide logic gates and wires onto the surface of silicon chips. Soon they will yield even smaller parts and inevitably reach a point where logic gates are so small that they are made out of only a handful of atoms. On the atomic scale matter obeys the rules of quantum mechanics, which are quite different from the classical rules that determine the properties of conventional logic gates. So if computers are to become smaller in the future, new, quantum technology must replace or supplement what we have now. The point is, however, that quantum technology can offer much more than cramming more and more bits to silicon and multiplying the clock-speed of microprocessors. It can support entirely new kind of computation with qualitatively new algorithms based on quantum principles!

Consider a register composed of three physical bits. Any classical register of that type can store in a given moment of time only one out of eight different numbers i.e. the register can be in only one out of eight possible configurations such as 000, 001, 010, ... 111. A quantum register composed of three qubits can store in a given moment of time all eight numbers in a quantum superposition. This is quite remarkable that all eight numbers are physically present in the register but it should be no more surprising than a qubit being both in state 0 and 1 at the same time. If we keep adding qubits to the register we increase its storage capacity exponentially i.e. three qubits can store 8 different numbers at once, four qubits can store 16 different numbers at once, and so on; in general L qubits can store 2L numbers at once. Once the register is prepared in a superposition of different numbers we can perform operations on all of them. For example, if qubits are atoms then suitably tuned laser pulses affect atomic electronic states and evolve initial superpositions of encoded numbers into different superpositions. During such evolution each number in the superposition is affected and as the result we generate a massive parallel computation albeit in one piece of quantum hardware. This means that a quantum computer can in only one computational step perform the same mathematical operation on 2L different input numbers encoded in coherent superpositions of L qubits. In order to accomplish the same

© 2010, IJARCS All Rights Reserved

task any classical computer has to repeat the same computation 2L times or one has to use 2L different processors working in parallel. In other words a quantum computer offers an enormous gain in the use of computational resources such as time and memory.

The paper is organized as, in the section 1 introduces quantum computing and in section 2 basics of quantum computing is given. In section 3 discuss about the quantum computing algorithms. In section 4 enlist advantages of quantum computing and finally concluding in section 5..

II. BASICS Of QUANTUM COMPUTING

In this section we present the basic paradigm for quantum algorithms, namely the quantum circuit model, which is composed of the basic quantum units of information (qubits) and the basic logical manipulations thereof (quantum gates).

A. *Quantum bit:*

The Quantum bit is smallest unit of information in a quantum computer. Unlike bits in classical systems, which are in one of two possible states labeled 1 and 0, a quantum bit exists in a superposition of these two states, settling on one or the other only when a measurement of the state is made, also called qubit.

The qubit is the quantum analogue of the bit, the classical fundamental unit of information. It is a mathematical object with specific properties that can be realized physically in many different ways as an actual physical system. Just as the classical bit has a state (either 0 or 1), a qubit also has a state. Yet contrary to the classical bit, 0 and 1 are but two possible states of the qubit, and any linear combination (superposition) thereof is also physically possible. In general, thus, the physical state of a qubit is the superposition $\psi = \alpha 0 + \beta 1$ (where α and β are complex numbers). The state of a qubit can be described as a vector in a two-dimensional Hilbert space, a complex vector space (see the entry on quantum mechanics). The special states 0 and 1 are known as the computational basis states, and form an orthonormal basis for this vector space. According to quantum theory, when we try to measure the qubit in this basis in order to determine its state, we get either 0 with probability α^2 or 1 with probability β^2. Since $\alpha^2 + \beta^2 = 1$ (i.e., the qubit is a unit vector in the aforementioned two-dimensional Hilbert state), we may (ignoring the overall phase factor) effectively write its state as $\psi = \cos(\theta)\, 0 + e i\varphi \sin(\theta) 1$, where the numbers θ and φ define a point on the unit three-dimensional sphere, as shown here. This sphere is often called the Bloch sphere, and it provides a useful means to visualize the state of a single qubit.

B. *Representation of Qubit:*

The two states in which a qubit may be measured are known as basis states (or basis vectors). As is the tradition with any sort of quantum states, Dirac, or bra-ket notation, is used to represent them. This means that the two computational basis states are conventionally written as and (pronounced "ket 0" and "ket 1").

Theoretically, a single qubit can store an infinite amount of information, yet when measured it yields only the classical result (0 or 1) with certain probabilities that are specified by the quantum state. In other words, the measurement changes the state of the qubit, "collapsing" it from the superposition to one of its terms. The crucial point is that unless the qubit is measured, the amount of "hidden" information it stores is conserved under the dynamic evolution (namely, Schrödinger's equation). This feature of quantum mechanics allows one to manipulate the information stored in unmeasured qubits with quantum gates, and is one of the sources for the putative power of quantum computers.

To see why, let us suppose we have two qubits at our disposal. If these were classical bits, then they could be in four possible states (00, 01, 10, 11). Correspondingly, a pair of qubits has four computational basis states ($|00\rangle$, $|01\rangle$, $|10\rangle$, $|11\rangle$). But while a single classical two-bit register can store these numbers only one at a time, a pair of qubits can also exist in a superposition of these four basis states, each of which with its own complex coefficient (whose mod square, being interpreted as probability, is normalized). As long as the quantum system evolves unitarily and is unmeasured, all four possible states are simultaneously "stored" in a single two-qubit quantum register. More generally, the amount of information that can be stored in a system of n unmeasured qubits grows exponentially in n. The difficult task, however, is to retrieve this information efficiently.

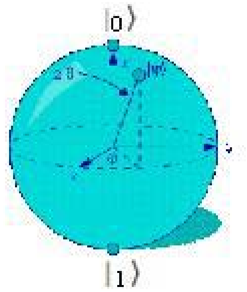

Figure 1. The Bloch Sphere.

C. *Quantum Gates:*

Classical computational gates are Boolean logic gates that perform manipulations of the information stored in the bits. In quantum computing these gates are represented by matrices, and can be visualized as rotations of the quantum state on the Bloch sphere. This visualization represents the fact that quantum gates are unitary operators, i.e., they preserve the norm of the quantum state (if U is a matrix describing a single qubit gate, then $U^\dagger U=I$, where $U^\dagger$ is the ad joint of U, obtained by transposing and then complex-conjugating U). As in the case of classical computing, where there exists a universal gate (the combinations of which can be used to compute any computable function), namely, the NAND gate which results from performing an AND gate and then a NOT gate, in quantum computing it was shown [9] that any multiple qubit logic gate may be composed from a quantum CNOT gate (which operates on a multiple qubit by flipping or preserving the target bit given the state of the control bit, an operation analogous to the classical XOR, i.e., the exclusive OR gate) and single qubit gates. One feature of quantum gates that distinguishes them from classical gates is that they are reversible: the inverse of a unitary matrix is also a unitary

© 2010, IJARCS All Rights Reserved

matrix, and thus a quantum gate can always be inverted by another quantum gate.

$$\mathrm{CNOT} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix}.$$

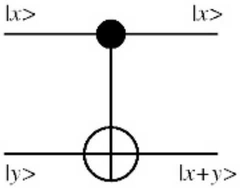

Figure 2. The CNOT Gate.

Unitary gates manipulate the information stored in the quantum register, and in this sense ordinary (unitary) quantum evolution can be regarded as computation [10] showed how a small set of single-qubit gates and a two-qubit gate is universal, in the sense that a circuit combined from this set can approximate to arbitrary accuracy any unitary transformation of nqubits). In order to read the result of this computation, however, the quantum register must be measured. The measurement gate is a non-unitary gate that “collapses” the quantum superposition in the register onto one of its terms with the corresponding probability. Usually this measurement is done in the computational basis, but since quantum mechanics allows one to express an arbitrary state as a linear combination of basis states, provided that the states are orthonormal (a condition that ensures normalization) one can in principle measure the register in any arbitrary orthonormal basis. This, however, doesn't mean that measurements in different bases are efficiently equivalent. Indeed, one of the difficulties in constructing efficient quantum algorithms stems exactly from the fact that measurement collapses the state, and some measurements are much more complicated than others.

D. Quantum Circuits:

Quantum circuits are similar to classical computer circuits in that they consist of wires and logical gates. The wires are used to carry the information, while the gates manipulate it (note that the wires do not correspond to physical wires; they may correspond to a physical particle, a photon, moving from one location to another in space, or even to time-evolution). Conventionally, the input of the quantum circuit is assumed to be a computational basis state, usually the state consisting of all 0 . The output state of the circuit is then measured in the computational basis, or in any other arbitrary orthonormal basis. The first quantum algorithms (i.e. Deutsch-Jozsa, Simon, Shor and Grover) were constructed in this paradigm.

E. Model quantum computer and quantum code:

In this section we describe a simple model for a quantum computer based on a classical computer instructing a machine to manipulate a set of spins. This model has some intrinsic limitations which make designing algorithms in a high-level language somewhat tricky. We discuss some of the rules for writing such quantum computer code as a high-level language and give an example.

Consider the following model for the operation of a quantum computer: Several thousand spin- particles (or two-level systems) are initially in some well defined state, such as spin-down. A classical machine takes single spins or pairs of spins and entangles them (performing an elementary one-bit operation or the two-bit XOR gate); see Figure 3 a, b and c. These stages are repeated on different pairs of spins according to the instructions of a conventional computer program. Since the spins are entangled, we must not look at the spins at intermediate stages: We must keep the quantum superposition intact. Furthermore, nothing else may interfere with the spins which could destroy their orientation or interrupt their unitary evolution. Once this well-defined cycle of manipulation is complete the orientations of the spins are measured (Fig. 3d). This set of measured orientations is the output of the computation.

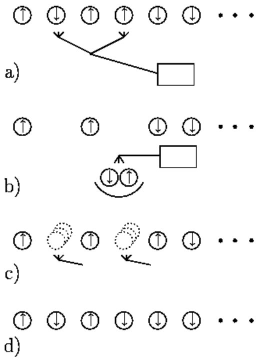

Figure 3. Model quantum computer

Figure 3 Model quantum computers as pictured by Shor [10]. Initially all particles are spin-down. Stage a) a classical machine takes a single or pair of spins and in stage b) it performs a selected one-bit or two-bit operation; in stage c) the ``entangled" particles are returned to their original locations. These three stages are repeated many times in accord with the instructions given by an ordinary classical computer. When this cycle is complete stage d) consists of measuring the state of the particles (leaving them in some particular bit-string); this bit-string is the result of the computation.

Given this paradigm for a quantum computer, what might its high-level language (its computer code) look like? The most serious difficulty that must be dealt with is that the quantum information is manipulated by a conventional computer in a completely blind manner---without any access to the values of this quantum information. This means that the program cannot utilize ``shortcuts" conditional on the value of a quantum variable (or register or bit). For example, loops must be iterated through exactly the same number of times independent of the values of the quantum variables. Similarly, conditional branches around large pieces of code must be broken down into repeated conditions for each step. In addition, each instruction performed upon the quantum bits must be logically reversible. Thus, ordinary assignments of a value to a variable, such as

$|a\rangle$ = n, are not legal and must instead be performed as increments on an initially zeroed variable, such as $|a\rangle = |a\rangle$ +n.

An example of such code that could run on this machine might look like this :

```
        do 10 k = 1, worstdiv
        |a⟩ = |a⟩ - n
        if ( |a⟩ >= 0 ) |q⟩ = |q⟩ + 1
10 continue
        do 20 k = 1, worstdiv
        if ( k > |q⟩ ) |a⟩ = |a⟩ + n
20 continue
```

This code fragment could be used to calculate the quotient and the remainder, placed in $|q\rangle$ and $|a\rangle$, respectively, for the division of $|a\rangle$ by n; the constant worstdiv is the worst-case number of times the loop must be traversed. Here $|q\rangle$ is initially zero. Each instruction here is either a conventional computer instruction or one involving some quantum variables. The former are direct instructions for the external computer, while the latter must be interpreted as a sequence of manipulations to be performed upon the quantum bits. As it stands, this code is not reversible (neither is it very efficient), e.g., the label 10 gives no specification of which routes might be used to get to it. It can, however, be easily rewritten.

III. QUANTUM ALGORITHM

Algorithm design is a highly complicated task, and in quantum computing it becomes even more complicated due to the attempts to harness quantum mechanical features to reduce the complexity of computational problems and to “speed-up” computation. Before attacking this problem, we should first convince ourselves that quantum computers can be harnessed to perform standard, classical, computation without any “speed-up”

Shor's algorithm [08], [09], [10] exploits the ingenious number theoretic argument that two prime factors p, q of a positive integer N=pq can be found by determining the period of a function f(x) = yx mod N, for any y < N which has no common factors with N other than 1 (Nielsen and Chuang 2000, App. 4). The period r of f(x) depends on y and N. Once one knows the period, one can factor N if r is even and yr/2$\neq$ −1 mod N, which will be jointly the case with probability greater than 1/2 for any y chosen randomly (if not, one chooses another value of y and tries again). The factors of N are the greatest common divisors of yr/2 ± 1 and N, which can be found in polynomial time using the well known Euclidean algorithm. In other words, Shor's remarkable result rests on the discovery that the problem of factoring reduces to the problem of finding the period of a certain periodic function f: Zn→ ZN, where Zn is the additive group of integers mod n (Note that f(x) = yx mod N so that f(x+r) = f(x) if x≤+n . The function is periodic if r divides n exactly, otherwise it is almost periodic). That this problem can be solved efficiently by a quantum computer is demonstrated with Simon's oracle.

Shor's result is the most dramatic example so far of quantum “speed-up” of computation, notwithstanding the fact that factoring is believed to be only in NP and not in NP-complete. To verify whether n is prime takes a number of steps which is a polynomial in log2n (the binary encoding of a natural number n requires log2n resources). But nobody knows how to factor numbers into primes in polynomial time, not even on a probabilistic Turing machine, and the best classical algorithms we have for this problem are sub-exponential. This is yet another open problem in the theory of computational complexity. Modern cryptography and Internet security protocols such public key and electronic signature are based on these facts [10] .It is easy to find large prime numbers fast and it is hard to factor large composite numbers in any reasonable amount of time. The discovery that quantum computers can solve factoring in polynomial time has had, therefore, a dramatic effect. The implementation of the algorithm on a physical machine would have economic, as well as scientific consequences.

IV. ADVANTAGES

There are several reasons that researchers are working so hard to develop a practical quantum computer. First, atoms change energy states very quickly -- much more quickly than even the fastest computer processors. Next, given the right type of problem, each qubit can take the place of an entire processor -- meaning that 1,000 ions of say, barium, could take the place of a 1,000-processor computer. The key is finding the sort of problem a quantum computer is able to solve.

If functional quantum computers can be built, they will be valuable in factoring large numbers, and therefore extremely useful for decoding and encoding secret information. If one were to be built today, no information on the Internet would be safe. Our current methods of encryption are simple compared to the complicated methods possible in quantum computers. Quantum computers could also be used to search large databases in a fraction of the time that it would take a conventional computer.

It has been shown in theory that a quantum computer will be able to perform any task that a classical computer can. However, this does not necessarily mean that a quantum computer will outperform a classical computer for all types of task. If we use our classical algorithms on a quantum computer, it will simply perform the calculation in a similar manner to a classical computer. In order for a quantum computer to show its superiority it needs to use new algorithms which can exploit the phenomenon of quantum parallelism.

The implications of the theories involved in quantum computation reach further than just making faster computers. Some of the applications for which they can be used are

A. *Quantum Communication:*

Quantum communication systems allow a sender and receiver to agree on a code without ever meeting in person. The uncertainty principle, an inescapable property of the quantum world, ensures that if an eavesdropper tries to monitor the signal in transit it will be disturbed in such a way that the sender and receiver are alerted.

B. *Quantum Cryptography:*

The expected capabilities of quantum computation promise great improvements in the world of cryptography. Ironically

CONFERENCE PAPER
National Conference on Information and Communication Technology for Development
Organized by PRMITR, Amravati (MS) India
http://mitra.ac.in/forthcoming.html

the same technology also poses current cryptography techniques a world of problems. They will create the ability to break the RSA coding system and this will render almost all current channels of communication insecure.

C. Artificial Intelligence:

The theories of quantum computation suggest that every physical object, even the universe, is in some sense a quantum computer. As Turing's work says that all computers are functionally equivalent, computers should be able to model every physical process. Ultimately this suggests that computers will be capable of simulating conscious rational thought. And a quantum computer will be the key to achieving true artificial intelligence.

V. CONCLUSION

Experimental and theoretical research in quantum computation is accelerating world-wide. New technologies for realizing quantum computers are being proposed, and new types of quantum computation with various advantages over classical computation are continually being discovered and analyzed, and we believe some of them will bear technological fruit. From a fundamental standpoint, however, it does not matter how useful quantum computation turns out to be, nor does it matter whether we build the first quantum computer tomorrow, next year or centuries from now. The quantum theory of computation must in any case be an integral part of the world view of anyone who seeks a fundamental understanding of the quantum theory and the processing of information.

VI. REFERENCES

[1]. R. Feynman, Int. J. Theor. Phys. 21, 467 (1982).

[2]. Deutsch, Proc. R. Soc. London A 400, 97 (1985).

[3]. P.W. Shor, in Proceedings of the 35th Annual Symposium on the Foundations of Computer Science, edited by S. Goldwasser (IEEE Computer Society Press, Los Alamitos, CA), p. 124 (1994).

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

[5]. R. Landauer, Trans. R. Soc. London, Ser. A 353, 367 (1995).

[6]. P. Domokos, J.M. Raymond, M. Brune and S. Haroche, Phys. Rev. A 52, 3554 (1995).

[7]. C. Monroe, D.M. Meekhof, B.E. King, W.M. Itano and D.J. Wineland, Phys. Rev. Lett. 75, 4714 (1995).

[8]. Shor, P. (1994) ‘Algorithms for quantum computation: Discrete logarithms and factoring’, Proceedings of the 35th Annual IEEE Symposium on Foundations of Computer Science, pp. 124–134.

[9]. Shor, P. (1995), ‘Scheme for reducing decoherence in quantum computer memory’, Phys. Rev., A 52: 2493–2496.

[10]. Shor, P. (2004), ‘Progress in quantum computing’, Quantum Information Processing, 3: 5–13.

[11]. Pitowsky, I. (2002), ‘Quantum speed-up of computations’, Philosophy of Science, 69: S168-S177.

© 2010, IJARCS All Rights Reserved