Available online at www.sciencedirect.com

ScienceDirect

IERI Procedia 10 (2014) 144 – 152

Procedia IERI

www.elsevier.com/locate/procedia

2014 International Conference on Future Information Engineering

A Comprehensive but not Complicated Survey on Quantum Computing

Pranav Santosh Menon*, Ritwik M.

Department of Computer Science & Engineering, Amrita Vishwa Vidyapeetham, Coimbatore-641112, India

Abstract

This paper discusses the basic components that are required to build a quantum computing environment. Quantum computers have a distinctive advantage over classical computers due to its ability to solve problems with large number of computations faster. To utilize these capabilities to its best, we should ensure that the computers are on par with the quantum computing requirements to float a working environment. The whole thrust on this is to ensure that the quantum computing environment is chained by the laws of quantum mechanics. The issue becomes more complicated when it is to be executed on a classical platform. This paper surveys the basics of Quantum Computing and the existing Quantum Computing simulators. Further, it also points out the basic rules to ensure a proper translation of each capabilities from a classical system to a quantum system and vice versa.

Selection and peer review under responsibility of Information Engineering Research Institute

Keywords:Quantum computer; qubit; superposition; entanglement; parallelism; Bloch sphere;

1. Introduction

Quantum computers have their principles based on quantum physics. Unlike today's computers, Quantum computers have to go a long way before we start using them on a day to day basis. Quantum computers is the next generation of computers that are expected to overcome some of these limitations. They are said to reduce the number of computations to complete a process. Meanwhile the time needed for these computations are said to take longer. Even with such complexities they are said to overcome certain difficulties that we face with the classical computers.

Selection and peer review under responsibility of Information Engineering Research Institute
doi:10.1016/j.ieri.2014.09.069

But today they exist on paper or as experiments at high end laboratories that are part of various premier global institutions, companies and research groups. Though there are a few emulators and simulators that are available to people who do not have access to such facilities. These emulators and simulators have their limitations. Section 2 describes the basics of quantum computing and the various underlying theories. Section 3 is on Quantum gates which form the basis for the hardware that Quantum computers work upon. Section 4 discusses a few programming languages and libraries that are used to simulate a quantum computing environment. Quantum algorithms that enable Quantum computers to produce the same result with lesser computations is discussed in section 5. Advances in this area are elaborated in Section 6. At last we conclude the paper in section 7.

2. Basics of quantum computers

The differences between Quantum computers and its contemporaries arise from its hardware compositions. With proper understanding and better implementation of a quantum computer we can use it to solve various problems that we face in a classical computer.

2.1. Qubit

The elementary component of a Quantum computer is a qubit [16]. Qubits are found in a Quantum computer as photons or nucleuses of certain elements. The minute particles that form the fundamental part of a qubit in the form of a quantum particles make a qubit. They influence various physical properties that a qubit exhibit [16]. These properties are called superposition, entanglement, parallelism. In a Quantum computer a qubit is directed into two distinctive spin directions a spin up for 0 and spin down for 1 [16]. This distinction is necessary to ensure that data is demarcated properly when superposition and entanglement comes into play on the same qubit.

At normal room temperature these photons are in a really unstable state [11]. As they are bouncing around from a lower energy state to a higher energy state and vice versa. Tough at a substantially low room temperature they are at a stable spin down state. If the spin has to be altered into a spin up or down position external energy is need to do this alteration [11].

$$|0\rangle = 0 \qquad |1\rangle = 1$$

This form of notation is called the Dirac notation [16].

Fig. 1.

The first spin state represents |1> and the second represents |0> [16].

2.2. Superposition

The most important ability of a Quantum computer is that it can superimpose various bit with different values into a single qubit this property is called superposition [16]. The distinctive superposition property is exhibited when both the sates co-exist in a single qubit. Here the qubit is considered to be depicting a |0> state as well as a |1> state individually they are called the basis state [2]. Their coexistence is governed by a factor called the probabilistic amplitude [2]. This probabilistic amplitude is important to determine a value for the

superposition state [2]. While measuring a photon at a superposition state only one of its constituent values can be measured. A state can be only measured and referred to as either a |0> or |1>. This is where the probabilistic amplitude comes into play [2]

$$|\psi> = \alpha|0> + \beta|1> \quad (1)$$

From *(1)* represents the general state of a qubit and we can also understand that α and β is called the probabilistic amplitude and its value is denoted according to the superposition [16]. These values represent how much each basis state has influenced a qubit.

$$\alpha^2 + \beta^2 = 1 \quad (2)$$

According to *(2)* the sum of the squares of the probabilistic amplitudes should always be one [2].

2.3. Entanglement

In a quantum environment a qubit is susceptible to any kind of change with regards to all other qubits in the system [17]. Unlike any other computers that are presently in use a Quantum computer provides a working condition where all the qubits are under the influence of each other. There is no observable interactions between the particles in the quantum environment [17].

In a completely entangled state a quantum computer with n qubits can entangle and superimpose 2^n classical states into n qubits [8]. This is because qubits are subjected to the Bell phenomenon which defines the state that they are confined to in quantum system [9].

Let A and B be two independent systems then the corresponding Hilbert space is represented as in *(3)* [10].

$$H_A \otimes H_B \quad (3)$$

The first system is in state $|\psi>_A$ similarly the second is in state $|\psi>_B$ thereby the equation is as in *(IV)*. [10]

$$|\psi>_A \otimes |\psi>_B \quad (4)$$

Their basis vectors as $|0>_A$ and $|1>_B$ for system A similarly vectors $|0>_B$ and $|1>_B$ are the basis states for system B. The following is the entangled state as in *(V)* [17].

$$|\psi>_{AB} = \frac{1}{\sqrt{2}}(|0>_A \otimes |1>_B - |1>_A \otimes |0>_B) \quad (5)$$

2.4 Parallelism

In reality a classical operation that has 2^n inputs would require 2^n gates or have to wait for 2^n turns to produce a truth table as an output. Whereas Quantum gate with 2^n inputs would only take a single operation to produce a truth table as an output [15]. Parallelism is the property that allows a Quantum computer to perform multiple operation at a time without waiting for another processes to complete or there is no need for more hardware to perform the same processes [6]. However, in these gates processing an input and producing an output takes considerably a longer time. This is due to the reason that there are a lot of intermediate changes

happening due to entangled qubits in the quantum environment. This takes a longer computational time but exponentially reduce the number of computations needed for completing a single process [6].

2.5. Bloch Sphere

Bloch sphere is used to represent a single qubit in a three dimensional space.

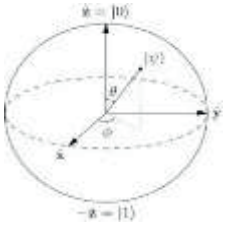

Fig.2. [16]

Fig. 2 represents a Bloch Sphere, here $|\psi\rangle$ represents the qubit while φ and θ is used to represent the polarization and superposition of the qubit [12].

The quantum environment is constantly under pressure to collapse and this leads to decoherence of the system. Thus we can conclude that until there is a higher coherence the system will collapse and resulting in an improper depiction of the qubits in the Bloch sphere [11].

$$\cos(\theta/2)|0\rangle + e^{i\varphi}\sin(\theta/2) = |\psi\rangle \quad (6)$$

Equation *(6)* expresses the state of a single qubit in polar form [10].

$$(x, y, z) = (\sin\theta\cos\varphi, \sin\theta\sin\varphi, \cos\theta) \quad (7)$$

Equation *(7)* expresses the translation of a qubit from Cartesian to polar form [10].

3. Quantum logical gates

All quantum gates can be represented as unitary matrices. Major portion of those representation can be done using 2x2 or 4x4 matrices. In general a quantum logical gate which acts on k qubits is represented with $2^k x 2^k$ matrix [5].

3.1 Hadamard gate

This gate operates about a single qubit. It maps the basis state $|0\rangle$ into $\frac{|0\rangle + |1\rangle}{\sqrt{2}}$ and $|1\rangle$ correspondingly into $\frac{|0\rangle - |1\rangle}{\sqrt{2}}$. It is represented using a Hadamard matrix in fig. 2 [12].

$$H = \frac{1}{\sqrt{2}}\begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$$

Fig. 3

It is one of the unitary matrix as H*H=I where I is an identity matrix [8].

3.2 Pauli-X gate

This gate has only one qubit as its input and it is equivalent to a classical NOT gate. It rotates a qubit by π radians along the X axis. It maps |0> to |1> and |1> to |0> [8]. The fig. 3 represents the matrix that is used for computing the output matrix.

$$X = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$$

Fig. 4.

3.3 Pauli- Y gate

It is similar to Pauli-X gate. It rotates the qubit in the Bloch sphere by π radian along the Y axis. Here the gate maps a |0> to i|1> and |1> to –i|0> [8]. The output is computed using the matrix from fig. 4.

$$Y = \begin{bmatrix} 0 & -i \\ i & 0 \end{bmatrix}$$

Fig. 5.

3.4 Pauli-Z gate

Operation of this gate is similar to Pauli-X and Pauli-Y gates. But the gate rotates the qubit by π radian about the Z axis. Here it leaves |0> state unchanged but it maps |1> to -|1> state [8]. Computation with the matrix in fig. 5 we get the corresponding output.

$$Z = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}$$

Fig. 6.

3.5 Phase Shift gate

This gate shifts the phase of a qubit but the probability of finding the qubit as a |0> or |1> remains the same. It leaves a |0> basis state unchanged and |1> to $e^{i\theta}|1>$ [12]. The phase shift is computed from the matrix in fig. 6.

$$R_\theta = \begin{bmatrix} 1 & 0 \\ 0 & e^{i\theta} \end{bmatrix}$$

Fig. 7.

Here θ is the phase shift. Pauli Z gate is a special of phase shift gate when θ=π [8].

3.6 Swap gate

This gate swaps the values of two qubits [12]. The matrix that allows the swap is represented in fig. 7.

$$SWAP = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Fig. 8.

3.7 Controlled gate

This gate works on two or more qubits. Here one or more qubits act as a control for an operation. A more general example is where U a gate that operates on a single qubit is applied to the controlled gate as follows [3]. Matrix represented in fig.8 is a universal representation of a controlled gate.

$$U = \begin{bmatrix} x_{00} & x_{01} \\ x_{10} & x_{11} \end{bmatrix}$$

Fig. 9.

Then the Controlled-U gate works with the first qubit as a control and the second qubit is the input for the operation. The Controlled-U gate maps the input as the following *(8)* , *(9)* , *(10* and *(11)* [4].

$$|00\rangle \longrightarrow |00\rangle \quad (8)$$

$$|01\rangle \longrightarrow |01\rangle \quad (9)$$

$$|10\rangle \longrightarrow |1\rangle U |0\rangle = |1\rangle (x_{00}|0\rangle + x_{10}|1\rangle) \quad (10)$$

$$|11\rangle \longrightarrow |1\rangle U |0\rangle = |1\rangle (x_{01}|0\rangle + x_{11}|1\rangle) \quad (11)$$

The matrix representation of the Controlled-U gate is as represented in fig. 9 [12].

$$C(U) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & x_{00} & x_{01} \\ 0 & 0 & x_{10} & x_{11} \end{bmatrix}$$

Fig. 10.

3.8 Toffoli gate

It is similar to the controlled controlled gate with 3 qubits as input. Here it only applies the operation on the third qubit if the first two is |1> otherwise it leaves the qubit undisturbed [5].

3.9 Fredkin gate

It is a quantum gate with three qubits as the inputs that performs a controlled swap. It has a useful property where the total number of |0> and |1> are preserved after the operation [5].

- Quantum Fourier Transform

It is the linear transformation on qubits. In a quantum computer we can perform Quantum Fourier transform with decomposition of simpler unitary matrices. Here it is achieved using combinations of Hadamard and phase shift quantum logical gates. In a Quantum computer Fourier transform can be implemented using $O(n^2)$ quantum logical gates where n is the number of qubits [13]. Where as in a classical system we need $O(n2^n)$ gates. This gives a quantum computer an exponential speed up over the classical computer. Quantum Fourier transform algorithms known today can achieve this in $O(n \log n)$ gates to achieve this approximation [13].

In a classical system the Discrete Fourier transform function maps (x_1, x_2, …, x_{N-1}) into (y_1, y_2, …, y_{N-1}) according to the *(XII)* [18].

$$y_k = \frac{1}{\sqrt{N}} \sum_{j=0}^{N-1} x_j \omega^{jk} \quad \text{, Where } \omega = e^{\frac{2\pi i}{N}} \quad (12)$$

Similarly in a quantum system quantum state $\sum_{i=0}^{N-1} x_i |i>$ gets mapped into $\sum_{i=0}^{N-1} y_i |i>$ according to the *(13)* [18]

$$y_k = \frac{1}{\sqrt{N}} \sum_{j=0}^{N-1} x_j \omega^{jk}. \tag{13}$$

Equation *(13)* can be mapped into a qubit representation using *(14)*

$$|j\rangle \mapsto \frac{1}{\sqrt{N}} \sum_{k=0}^{N-1} \omega^{jk} |k\rangle. \tag{14}$$

Eventually the matrix F_N that contains the quantum Fourier transform on qubits is as follows in fig. 10 [18].

$$F_N = \frac{1}{\sqrt{N}} \begin{bmatrix} 1 & 1 & 1 & 1 & \cdots & 1 \\ 1 & \omega & \omega^2 & \omega^3 & \cdots & \omega^{N-1} \\ 1 & \omega^2 & \omega^4 & \omega^6 & \cdots & \omega^{2(N-1)} \\ 1 & \omega^3 & \omega^6 & \omega^9 & \cdots & \omega^{3(N-1)} \\ \vdots & \vdots & \vdots & \vdots & & \vdots \\ 1 & \omega^{N-1} & \omega^{2(N-1)} & \omega^{3(N-1)} & \cdots & \omega^{(N-1)(N-1)} \end{bmatrix}.$$

Fig. 11.

4. Quantum languages/libraries

Quantum languages and libraries enable us to create an artificial Quantum computer to simulate a quantum computing environment. These languages and libraries fail to hold the quantum computing principle. Quantum computing environment should be left with minimum interference from any external sources to function properly [16]. The system should not be interfered constantly, as each interaction produces a ripple effect throughout the system. We will get an answer to all these problems if we satisfy the following conditions

- The computer that is being used for running these languages should be working on a digital computer and not an analog computer [1]. This is because an analog computer cannot process any other computation other than its design specification [1]. This very disadvantage that an analog computer poses is the very reason why we should choose a digital computer. There is a need for an ever changing environment in a quantum computer these changes can only be simulated in a digital computer [1].
- The programming language should follow a functional programming structure. By using a functional programming structure we can compute the process as a whole entity with a proper bounded structure [7]. This type of programming will give the process a well-defined mathematical structure to carry about various mathematical calculations needed in a quantum computer. In case of a functional programming approach the quantum environment to work without any interference and the result is only produced when the computations are complete [7]. By this method we are not bothered about the intricacies only the proper execution and correct output is the requisite [7]. Such an approach will ensure that the principles of quantum computing is met in the simulation.

5. Quantum algorithms

First of all we should understand that quantum algorithms can only solve the problems that a classical computer can solve. But it can solve a problem much faster than a classical computer in various instances such as integer factorization, search algorithm, etc.

This exponential speed up is because of the fact that a quantum computers can carry out certain computation faster than a classical computer. Most of the successful quantum algorithms use Quantum Fourier transforms in them this is because of the fact that they require much lesser hardware. In a quantum Fourier transform the gates need for computing n qubits is much lesser than a classical computer [18]. That is the reason why Shor's algorithm for integer factorization is faster. It would only require lesser hardware as it is based on quantum Fourier transform to carry out the repeated operation that the algorithm uses [13]. Another such algorithm is Grover's search algorithm, here faster search is possible as there are multiple processes running in the system at the same time computed using Quantum Fourier transform [14].

6. Advances in quantum computing

Since its conception in the later part of the twentieth century quantum computer have remained as a theoretical concept until the past few years. Today there are quantum computing labs with working models of quantum computer to test these algorithms. DWave a quantum computing company that has come out with actual working model of quantum computers. The 512 qubit chipset system called DWave 2 is to be installed in a NASA installation to allow researchers to work on some tough computer science problems involving machine learning, speech recognition, web search, search for exoplanets, planning and execution and in mission control canters [19].

A working Quantum computer would already be in existence if we are to go by former NSA agent Edward Snowden's reports. No one knows how it would turn out to be in the coming years. But one thing is for sure that quantum computers are going to change the face of computing by allowing us to overcome some of the limitations that a classical computer faces.

7. Conclusion

We do believe that after reading this paper one can understand the basics of quantum computer with much less hassle. Quantum computing is one field that has come into existence decades back and only since the last decade more attention is being paid into this field. With proper research on the above said areas, it wouldn't be a long wait to get our hands on a feasible quantum computer that can solve our problems faster.

References

[1] H. Toibman, "A Painless Survey of Quantum Computation", Dec. 2004.
[2] Y. Kanamori, S. M. Yoo, W. D. Pan, and F. T. Sheldon, "A Short Survey on Quantum Computers", International Journal of Computers and Applications, Vol.28, No.3, 2006.
[3] A. Barenco, C. H. Bennett, R. Cleve, D. P. DiVincenzo, N. Margolus, P. Shor, T. Sleator, J. A. Smolin, H. Weinfurter, "Elementary Gates for Quantum Computation", The American Physical Society, Vol.50, No.5, Nov. 1995.
[4] A. Muthukrishnan, "Classical and Quantum Logic Gates: An Introduction to Quantum Computing", Quantum Information Seminar, Rochester Center for Quantum Information, Sep. 1999.
[5] A. Al-Rabadi, L. Casperson, M. Perkowski, and X. Song, "Multiple Valued Quantum Logic", Quantum Vol.10, No.2, 2002.
[6] D. Deutsch, "Quantum theory, the Church-Turing principle and the universal quantum computer", The

Royal Society of London A.400, pp 97-117, 1985.
[7] S. J. Gay, “Quantum Programming Languages Survey and Bibliography”, Math Structure in Computer Science, Vol.14, No.4, 2006.
[8] Shaktikanta Nayak, Sitakanta Nayak, J. P. Singh, “An Introduction to Basic Logic Gates for Quantum Computer”, International Journal of Advanced Research in Computer Science and Software Engineering, Vol.3, Issue.10, Oct. 2013.
[9] R. A. Bertlmann, H. Narnhofer, and W. Thirring,” A Geometric Picture of Entanglement and Bell Inequalities”, The American Physical Society, A.66, Issue 3, Sep 2002.
[10] A. Mandilara, J. W. Clark and M. S. Byrd “Elliptical orbits in the Bloch sphere", Journal of Optics B: Quantum and Semi classical Optics, Volume 7 Number 10, Sep. 2005.
[11] A. C. Elitzur, L. Vaidman, “Quantum Mechanical Interaction-Free Measurements”, Foundations of Physics, Vol. 23, No. 7, 1993.
[12] B. Omer, “Structured Quantum Programming”, Ph.D. thesis, Technical University of Vienna, 2003.
[13] P. Shor, “Polynomial-Time Algorithms for Prime Factorization and Discrete Logarithms on a Quantum Computer”, 35th Annual Symposium on Foundations of Computer Science, 1994.
[14] “A fast quantum mechanical algorithm for database search”, 28th ACM Annual Symposium on theory of computing (STOC), pages 212-219, May 1996.
[15] M. Ziegler, “Computational Power of Infinite Quantum Parallelism”, International Journal of Theoretical Physics, Vol.44, No.11, November2005.
[16] M. D. Purkeypile, “Cove: A Practical Quantum Computer Programming Framework”, PhD thesis, Colorado Technical University, Colorado Springs, USA, Sep. 2009.
[17] W. Tittel, J. Brendel, H. Zbinden and N. Gisin,“Quantum Cryptography using entangled photons in energy – time Bell states”, Physics Review Letters, Vol. 84 Issue 20, pp:4737-4740, 2000.
[18] C. M. Bowden, G. Chen, Z. Diao, A. Klappenecker, “The Universality of the Quantum Fourier Transform in Forming the Basis of Quantum Computing Algorithms”, arXiv preprint quant-ph/0007122, 2000.
[19] (2013), DWave systems website press release, [online], http://www.dwavesys.com/en/pressreleases.html#dwaveus_Google_NASA.