CAB402 Programming Paradigms
Quantum Computing

Thanat Chokwijitkul n9234900

Contents

1. Introduction

The theory of computation has been extensively developed during the last few decades. Computers have provided reliable solutions for a myriad of community's seemingly unsolvable problems. Notwithstanding, various complicated problems have been continuously introduced to society as it never stops growing and becomes more complex. Even though technology nowadays has been steady advancing to approach the demands of society, whereas such steady progress has its limitation since many of those problems are intricate to model or appear to require time-intensive solutions. This phenomenon implies to the necessity of a new computing revolution since classical computation no longer has an ability to reach the increased demand.

According to Moore's law, the computational power of computers would dramatically increase approximately every two years (Thompson & Parthasarathy, 2006). The theory can be proven real since the size of transistors has rapidly become smaller to a few nanometres (Nielsen & Chuang, 2000). This results in a higher number of transistors mounted on an integrated circuit. The laws of classical physics do not function with objects with such small size. Thus, quantum computing becomes a next solution to deal with these complex problems.

Quantum computing is the revolution. Although it is a relatively new field of research, this technology has the potential to introduce the world of computing to a new stage where certain computationally intense problems can be solved with a shorter amount of processing time. One can consider quantum computing as the art of utilising all the possibilities that the laws of quantum physics contribute to solving computational problems while classical computers merely use a minuscule subset of these possibilities. However, quantum computers are not a replacement for conventional computers as quantum mechanics only improve the computing efficiency for certain types of computation.

This research report delivers a brief glimpse into the world of quantum computers and the laws of quantum mechanics applied to this entirely new type of computer. It will also introduce the fundamental concepts of the field along with exploring quantum computing in practice using Language-Integrated Quantum Operations (LIQUi$|\rangle$). This report concludes with experience evaluation in the quantum computing research in relation to the content of the unit (Programming Paradigms).

2. Early History

In the early 1980s, physicist Richard Freyman recognised that it was not possible for phenomena associated with entangled particles (quantum phenomena) to be efficiently simulated on classical computers (Deutsch, 1982). Freyman was the first who suggested that quantum-mechanical systems might have higher computational power than conventional computers.

In the same decade, Paul Benioff proved that quantum-mechanical systems were at least as powerful as classical computers because Turing machines could be modelled with such system (Yanofsky & Mannucci, 2008).

In 1985, David Deutsch, of Oxford, published the paper ”Quantum theory, the Church-Turing principle and the universal quantum computer” (Deutsch, 1985). Deutsch claimed that the universal Turing machine has superior abilities compared with Turing machine, including random number generation, parallelism and physical system simulation with finite-dimensional state spaces.

Another of Deutsch's paper ”Quantum computational networks” (Deutsch, 1989) was published in 1989. The article proved that quantum circuits are as powerful as the universal Turing machine. Deutsch also introduced the first truly quantum algorithm in 1990, namely Deutsch's algorithm which later was generalised to the Deutsch-Jozsa algorithm.

In 1993, Andrew Chi-Chih Yao extended Deutsch's paper ”Quantum computational networks” (Yao, 1993) in the paper ”Quantum circuit complexity” by addressing the complexity of quantum computation according to Deutsch's work. Yao's findings indicated quantum computing researchers to concentrate on quantum circuits instead of quantum Turing machine.

Peter Shor proposed another quantum algorithm in 1994. This algorithm uses the concept of qubit entanglement and superposition for integer factorisation (Bhunia, 2010). In principle, executing the algorithm on a quantum computer would far surpass the efficiency of all classical computers.

The University of California, Harvard University, Massachusetts Institute of Technology and IBM researchers conducted an experiment using nuclear magnetic resonance (NMR) to manipulate quantum data in liquids. The team also developed a 2-bit quantum computer with radio frequency as its input. Afterwards, a new quantum algorithm that executes on quantum computers was introduced by Lov Grover of Bell Laboratories in 1996 by the name Grover's quantum algorithm (Bhunia, 2010).

In 1998, researchers at the University Innsbruck in Austria put the idea of quantum teleportation (IBM, 2014), proposed in 1993, into practice. The theorem demonstrates the concept of entanglement and teleportation. This research is an implication for data transfer and network in the quantum system.

3. Basic Concepts

3.1 Qubits

A bit is the most fundamental building block of the classical model of computer, which has a single logical value, either false or true or simply 0 or 1. In a quantum computer, the quantum bit or qubit also has two computational basis states; 0 and 1, represented by $|0\rangle$ and $|1\rangle$ respectively. However, it can be in a superposition of quantum mechanical two-state systems, meaning the qubit is both in state 0 and 1 simultaneously. A superposition of the qubit can be represented by $\alpha|0\rangle + \beta|1\rangle$ for some α and β such that $|\alpha|^2 + |\beta|^2 = 1$.

In computer system, information is represented in binary form since it is stored in the registers. For example, the non-negative numbers can be represented in binary form as

$$0, 1, 10, 11, 100, 101, 110, 111...$$

A number of bits can determine how many configurations that binary string can represent since $2^n = y$ where n is a number of bits and y is a number of different configurations. For example, a three-bit binary string can represent $2^3 = 8$ numbers including 0 to 7. On the other hand, in quantum computers, the non-negative numbers can be represented in binary form as

$$|0\rangle, |1\rangle, |1\rangle \otimes |0\rangle, |1\rangle \otimes |1\rangle, |1\rangle \otimes |0\rangle \otimes |0\rangle ...$$

In this case, an integer can be written in the form of $|x_{n-1}\rangle \otimes |x_{n-2}\rangle \otimes |x_{n-3}\rangle \otimes ... \otimes |x_1\rangle \otimes |x_0\rangle$ where $|x\rangle$ is a single qubit and $x \in \{0, 1\}$. Therefore, a quantum register of size three must be able to represent positive integers from 0 to 7 as the following:

$$|0\rangle \otimes |0\rangle \otimes |0\rangle \equiv |000\rangle \equiv |0\rangle \ ... \ |1\rangle \otimes |1\rangle \otimes |1\rangle \equiv |111\rangle \equiv |7\rangle$$

However, each qubit can be in both states simultaneously. A superposition of a single qubit can be denoted by $1/\sqrt{2}(|0\rangle + |1\rangle)$. Therefore, a superposition of a quantum register of size three will be

$$\frac{1}{\sqrt{2}}(|0\rangle + |1\rangle) \otimes \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle) \otimes \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$$

This can be represented in binary and decimal forms without the constant respectively as

$$\begin{aligned} &|000\rangle + |001\rangle + |010\rangle + |011\rangle + |100\rangle + |101\rangle + |110\rangle + |111\rangle \\ &\equiv |0\rangle + |1\rangle + |2\rangle + |3\rangle + |4\rangle + |5\rangle + |6\rangle + |7\rangle \\ &\equiv \sum_{x=0}^{7} |x\rangle \end{aligned}$$

3.2 Quantum Gates

In order to explain the concept of quantum gates, the basis states must be represented differently from the previous section. Since matrix transformations will be used to demonstrate the concept of each gate, each qubit state will be represented by any two orthogonal unit column vectors as follows:

$$|0\rangle = \begin{bmatrix} 1 \\ 0 \end{bmatrix}, |1\rangle = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$$

In this case, the state $|0\rangle$ and $|1\rangle$ are the representations of logical zero and logical one respectively. The qubit's actual state $|\Psi\rangle$ or its superposition can be represented by

$$|\Psi\rangle = \alpha|0\rangle + \beta|1\rangle = \begin{bmatrix} \alpha \\ \beta \end{bmatrix}$$

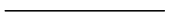
3.2.1 Quantum Gates

Each quantum gate can be represented by a square matrix. It also needs to be unitary since being unitary will preserve the unit length of the state vector $|\Psi\rangle$ after matrix multiplication and the new state must meet the normalisation criteria $|\alpha|^2 + |\beta|^2 = 1$. Thus, given the 2×2 identity matrix I, the output state will remain in the same state.

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

$$I|\Psi\rangle = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \alpha \\ \beta \end{bmatrix} = \begin{bmatrix} \alpha \\ \beta \end{bmatrix} = |\Psi\rangle$$

The matrix multiplication yields a new qubit state which is identical to the input state. This kind of quantum gate can be represented by a single wire as the following:



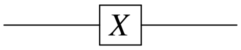
3.2.2 The NOT gate

The NOT gate, represented by the negation matrix, flips its input into the opposite value. This rule indicates that if the initial state of the input is 0, and result will be 1 and vice versa. Thus, given the negation matrix X, each state will be inverted into its opposite value after the multiplication process.

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

$$X|\Psi\rangle = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} \alpha \\ \beta \end{bmatrix} = \begin{bmatrix} \beta \\ \alpha \end{bmatrix}$$

The following diagram represents the NOT gate in the quantum circuit:



3.2.3 The Hadamard Gate

This quantum gate is very important since the actual state of each qubit can be in a superposition state. Given the Hadamard matrix H, if the input is 0, its output will be the normalised sum of both basis states (0 and 1). In contrast, if the input is 1, the output will be the normalised difference of both basis states.

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

$$H\Psi = \frac{1}{\sqrt{2}}\begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}\begin{bmatrix} \beta \\ \alpha \end{bmatrix} = \frac{1}{\sqrt{2}}\begin{bmatrix} \alpha+\beta \\ \alpha-\beta \end{bmatrix}$$

The following diagram represents the Hadamard gate in the quantum circuit:

3.2.4 Entanglement

Each qubit has its own quantum state. However, two or more qubits can act on one another which leads to the formation of an entangled system. When qubit states are entangled, it needs to be treated as the entire system or overall state, instead of individual quantum state. For example, given a two-bit system, it can be any integer between the range 0 and 3 inclusive as the following:

$$|00\rangle + |01\rangle + |10\rangle + |11\rangle$$

Thus, its normalised superposition can be expressed as

$$|\Psi\rangle = \alpha|00\rangle + \beta|01\rangle + \gamma|10\rangle + \delta|11\rangle$$

This rule also needs to meet the condition $|\alpha|^2 + |\beta|^2 + |\gamma|^2 + |\delta|^2 = 1$. Thus, given an arbitrary set of orthogonal column vectors, each vector represents a possible quantum state, it should yield a new column vector with multi-qubit states.

$$|00\rangle = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix}, |01\rangle = \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \end{bmatrix}, |10\rangle = \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \end{bmatrix}, |11\rangle = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} \quad thus, |\Psi\rangle = \begin{bmatrix} \alpha \\ \beta \\ \gamma \\ \delta \end{bmatrix}$$

3.2.5 The Controlled-NOT Gate

Unlike single-qubit gates such as the Hadamard gate or the NOT gate, the controlled-NOT or CNOT gate operates on two qubits by flipping the value of the second bit if the first bit is 1, but the value of the second bit remains unchanged if the first bit is 0. The following matrices illustrate how the CNOT gate operate on two qubits.

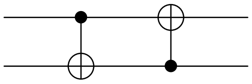
$$CNOT = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

$$CNOT\,|\Psi\rangle = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} \alpha \\ \beta \\ \gamma \\ \delta \end{bmatrix} = \begin{bmatrix} \alpha \\ \beta \\ \delta \\ \gamma \end{bmatrix}$$

In order to make the demonstration more concrete, the state can be represented with a 4×2 matrix representing state values of both qubits.

$$CNOT\,|\Psi\rangle = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} 0 & 0 \\ 0 & 1 \\ 1 & 0 \\ 1 & 1 \end{bmatrix} = \begin{bmatrix} 0 & 0 \\ 0 & 1 \\ 1 & 1 \\ 1 & 0 \end{bmatrix}$$

The following diagram illustrates the CNOT gate in the quantum circuit:



3.3 Unitary Transformation

As demonstrated in the quantum gates section, unitary transformations can be justified as matrix operations on vectors. This kind of quantum state manipulation can be represented as the following:

$$|\Psi\rangle \mapsto M \cdot |\Psi\rangle$$

In this case $|\Psi\rangle$ is a quantum state and M is a matrix. M is unitary if $M' \cdot M = I$ such that I is the identity matrix. Unitary transformations are reversible and opposed to being information destructive.

3.4 Measurements

Contrary to the concept of unitary transformations, measurements are not reversible and therefore information destructive. However, this kind of operation is effective in retrieving classical information back from quantum information along with interfering or destroying the quantum state.

For example, given the quantum state $|\Psi\rangle = \alpha|0\rangle + \beta|1\rangle$, the state $|0\rangle$ with probability α^2 and the state $|1\rangle$ with probability β^2 will be used to measure the actual state $|\Psi\rangle$. The qubit will be determined to be in one of both states until a new transformation occurs.

4. Applications

Since quantum computing is relatively a new area of study, many of its real-world applications are not yet to be seen. Nevertheless, this young field has been being explored by researchers who acknowledge its possibility in advancing the area of high-performance computing. This section delineates potential applications of quantum computing in cryptography along with its classical quantum algorithms, including Shor and Grover's algorithms.

4.1 Cryptography

Public key cryptography has a long history in information security since it is an effectively practical cryptosystem used for data transmission. This approach primarily relies on the complexity of how to crack the communication. This cryptosystem is still considered unconditionally secure since there does not exist a mathematical theorem that prevents eavesdroppers from creating sophisticated revolutionary algorithms to interfere the communication.

Quantum cryptography provides an elegant solution to the dilemma using Quantum Key Distribution (QKD) to exchange a symmetric key over a quantum channel. The security of data transmission can be ensured since quantum error correction codes allow all parties of the communication channel to be able to detect the presence of potential eavesdroppers.

4.2 Algorithms

It is wildly acknowledged that potential quantum-mechanic applications are primarily based on quantum algorithms, computational algorithms run on a quantum computer with highly efficiency improvement over any classical algorithm. This section intends to deliver an overview of quantum algorithmics, focusing on Shor and Grover's algorithms.

4.2.1 Shor's Algorithm

On a classical computer, factoring a large integer is considered a highly time-intensive task. However, Shor indicates that factoring a large integer in polynomial relies on the capacity to find the order of that integer *mod N*. Shor's algorithm follows the processes outlined below:

1. Select a random x, such that $x < N$
2. Compute $f = gcd(x,N)$. This may be done using the Euclidean algorithm. If $f \neq 1$, then return f since it is a prime factor.

3. Find the least r such that $x^r \equiv 1 \; mod \; N$, where r is a repetition period.

4. If either $gcd(x^{r/2}-1,N) \neq 1$ or $gcd(x^{r/2}+1,N) \neq 1$, then return it as a prime factor.

5. Otherwise, repeat the process from the first step.

This algorithm consists of two parts, including the classical and quantum parts. According to the algorithm, computing x^r in the third step is necessary. However, this cannot be implemented classically. In this case, Shor utilised the quantum Fourier transform which relies on quantum parallelism to solve the problem.

4.2.2 Grover's Algorithm

Grover's algorithm is one of the most well-known quantum algorithms. It is extremely beneficial to any large technology-based organisation which has its data stored in a database where searching is required. This process of searching is incredibly time-efficient compared with any other classical algorithms. For a search over a set of unordered data, the best classical algorithm requires $O(N)$ time, while Grover's algorithm performs a search in only $O(\sqrt{N})$ operations to find the unique element that satisfies a particular condition, which is a quadratic speedup. Grover's algorithm is as follows:

1. Prepare a quantum register of n qubits, where n is a number of qubits necessary to represent the search space of size $2^n = N$. All qubits need to be initialised to the state $|0\rangle$.

2. Compute $P(x_i)$ where the superposition is calculated.

$$\frac{1}{\sqrt{2^n}} \sum_{x=0}^{n-1} |x, P(x)\rangle$$

3. Invert amplitude of a_j to $-a_j$ such that $P(x_j) = 1$. Then apply inversion to increase amplitude of x_j with $P(x_j) = 1$.

4. Repeat steps 2 to 3 $\frac{\pi}{4}\sqrt{2^n}$ times.

5. Measure the result.

In order to achieve such performance, Grover's algorithm relies on the superposition of qubit states. Additionally, it also utilises the amplitude amplification algorithms which are unique to quantum computing to solve the problem.

References

Bhunia, C. T. (2010). *Introduction to quantum computing*. New Age International.

Deutsch, D. (1982). Quantum computation. *Physics World*.

Deutsch, D. (1985). Quantum theory, the church-turing principle and the universal quantum computer. *Proceedings of the Royal Society of London, Series A*, *400*(1818), 97-117.

Deutsch, D. (1989). Quantum computational networks. *Proceedings of the Royal Society of London, Series A*, *425*(1868), 73-90.

IBM. (2014). *Quantum teleportation.* `http://researcher.watson.ibm.com/researcher/view_group.php?id=2862`.

Nielsen, M. A., & Chuang, I. L. (2000). *Quantum computation and quantum information*. Cambridge University Press.

Thompson, S. E., & Parthasarathy, S. (2006). Moore's law: the future of si microelectronics. *Materials Today*, *9*(6), 20-25.

Yanofsky, N. S., & Mannucci, M. A. (2008). *Quantum computing for computer scientists*. Cambridge University Press.

Yao, A. C.-C. (1993). Quantum circuit complexity. *Proceedings of 34th Annual IEEE Symposium on Foundations of Computer Science*, 352-361.