Quantum Computer Report

Hatem Gamal Ahmed

August 2019

1 Introduction

1.1 Motivation

Quantum computers offers the possibility of exponential growth in computational power. Problems that take exponential time to be solved on classical computers can be solved in polynomial time when run on quantum computers with a specific algorithm. Therefore, some problems that needs years to be solved on classical computers can take only hours using its quantum counterpart. However, there are many limitations and challenges that Quantum computers must overcome in order to be effective. First of all Quantum Computers depend on Quantum properties that exist on particles with very small scale (such as atoms). These properties include, but not limited to, superposition and quantum entanglement.

1.2 Aim of internship

The aim of this internship is to dive into the rising field of quantum computers, their advantages, how they work and finally create a simulator for a quantum computer that runs locally on a classical computer using java, and providing a library for performing gates on qubits and measurement.

2 Literature

2.1 State

An undisturbed motion that is restricted by as many conditions or data as are theoretically possible without mutual interference or contradiction. [2]

2.2 Quantum superposition

Any two or more states may be superposed to give a new state. The procedure of expressing a state as the result of superposition of a number of other states is a mathematical procedure that is always permissible, independent of any reference to physical conditions. [2]

2.3 Quantum entanglement

Consider a source which emits a pair of particles such that one particle emerges to the left and the other one to the right. The source is such that the particles are emitted with opposite momenta. If the particle emerging to the left, which we call particle 1, is found in the upper beam, then particle 2 travelling to the right is always found in the lower beam. Conversely, if particle 1 is found in the lower beam, then particle 2 is always found in the upper beam. In our qubit language we would say that the two particles carry different bit values. Either particle 1 carries ”0” and then particle 2 definitely carries ”1”, or vice versa. Quantum mechanically this is a two-particle superposition state of the form [1]

$$\frac{1}{\sqrt{2}}\left(|0\rangle_1|1\rangle_2 + e^{\imath\chi}|1\rangle_1|0\rangle_2\right) .$$

2.4 Qubit Definition

Qubit: the basic unit of quantum information. A two – state system where the two states are simply called $|0\rangle$ and $|1\rangle$. Basically any quantum system which has at least two states can serve as a qubit. The most essential property of quantum states when used to encode bits is the possibility of coherence and superposition, the general state being $|Q\rangle = \alpha|0\rangle + \beta|1\rangle$.

Which is represented as the vector: $\begin{bmatrix} \alpha \\ \beta \end{bmatrix}$ *with* $|\alpha|^2 + |\beta|^2 = 1$.

What this means is not that the value of a qubit is somewhere between "0" and "1", but rather that the qubit is in a superpositionof both states and, if we measure the qubit we will find it with probability $|\alpha|^2$ to carry the value "0" and with probability $|\beta|^2$ to carry the value "1".

[1]

2.5 Kronecker product

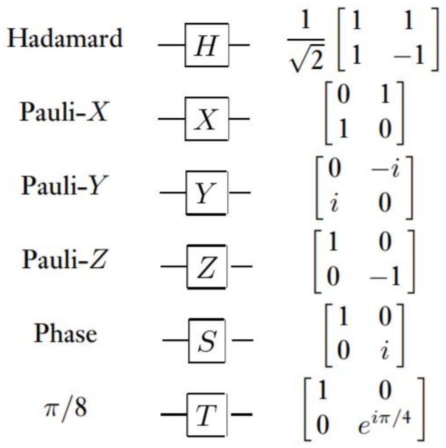
Now since a qubit can be represented by a 2x1 matrix, the representation of quantum gates can also be represented in the form of matrices. Therefore, to apply a gate on a qubit, the gate matrix is multiplied with the qubit matrix.

$$\begin{bmatrix} a & b \\ c & d \end{bmatrix} \otimes \begin{bmatrix} a^* & c^* \\ b^* & d^* \end{bmatrix} = \begin{bmatrix} a\begin{bmatrix} a^* & c^* \\ b^* & d^* \end{bmatrix} & b\begin{bmatrix} a^* & c^* \\ b^* & d^* \end{bmatrix} \\ c\begin{bmatrix} a^* & c^* \\ b^* & d^* \end{bmatrix} & d\begin{bmatrix} a^* & c^* \\ b^* & d^* \end{bmatrix} \end{bmatrix}$$

$$= \begin{bmatrix} aa^* & ac^* & ba^* & bc^* \\ ab^* & ad^* & bb^* & bd^* \\ ca^* & cc^* & da^* & dc^* \\ cb^* & cd^* & db^* & dd^* \end{bmatrix}$$

2.6 Single Quantum Gates

[3]



Hadamard	$-\boxed{H}-$	$\frac{1}{\sqrt{2}}\begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$
Pauli-X	$-\boxed{X}-$	$\begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$
Pauli-Y	$-\boxed{Y}-$	$\begin{bmatrix} 0 & -i \\ i & 0 \end{bmatrix}$
Pauli-Z	$-\boxed{Z}-$	$\begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}$
Phase	$-\boxed{S}-$	$\begin{bmatrix} 1 & 0 \\ 0 & i \end{bmatrix}$
$\pi/8$	$-\boxed{T}-$	$\begin{bmatrix} 1 & 0 \\ 0 & e^{i\pi/4} \end{bmatrix}$

2.7 Qubit Group

Since for a quantum gate that is applied on two qubits will have a 4x4 matrix, it should be multiplied with 4x1 qubit matrix. This 4x1 matrix represents two qubits grouped together.[3]

$$|\psi_0\rangle = \alpha_0|0\rangle + \alpha_1|1\rangle, \quad |\psi_1\rangle = \beta_0|0\rangle + \beta_1|1\rangle$$

$$\underbrace{|\psi_0\rangle|\psi_1\rangle}_{\text{2 qubits}} = \alpha_0\beta_0|00\rangle + \alpha_0\beta_1|01\rangle + \alpha_1\beta_0|10\rangle + \alpha_1\beta_1|11\rangle$$

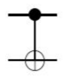
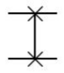
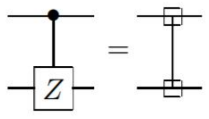
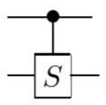
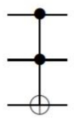
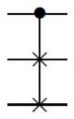
$$\textit{This can also be represented as:} \begin{bmatrix} \alpha_0\beta_0 \\ \alpha_0\beta_1 \\ \alpha_1\beta_0 \\ \alpha_1\beta_1 \end{bmatrix}$$

2.8 Measurement

The operation of collapsing a qubit or a qubit cluster to a single value.

2.9 Some Multi Quantum Gates

[3]

controlled-NOT		$\begin{bmatrix}1&0&0&0\\0&1&0&0\\0&0&0&1\\0&0&1&0\end{bmatrix}$
swap		$\begin{bmatrix}1&0&0&0\\0&0&1&0\\0&1&0&0\\0&0&0&1\end{bmatrix}$
controlled-Z		$\begin{bmatrix}1&0&0&0\\0&1&0&0\\0&0&1&0\\0&0&0&-1\end{bmatrix}$
controlled-phase		$\begin{bmatrix}1&0&0&0\\0&1&0&0\\0&0&1&0\\0&0&0&i\end{bmatrix}$
Toffoli		$\begin{bmatrix}1&0&0&0&0&0&0&0\\0&1&0&0&0&0&0&0\\0&0&1&0&0&0&0&0\\0&0&0&1&0&0&0&0\\0&0&0&0&1&0&0&0\\0&0&0&0&0&1&0&0\\0&0&0&0&0&0&0&1\\0&0&0&0&0&0&1&0\end{bmatrix}$
Fredkin (controlled-swap)		$\begin{bmatrix}1&0&0&0&0&0&0&0\\0&1&0&0&0&0&0&0\\0&0&1&0&0&0&0&0\\0&0&0&1&0&0&0&0\\0&0&0&0&1&0&0&0\\0&0&0&0&0&0&1&0\\0&0&0&0&0&1&0&0\\0&0&0&0&0&0&0&1\end{bmatrix}$

3 Methodology

3.1 Addressing Challenges

First of all, the problem of the inverse Kronecker product, for grouped qubits (more than 1 qubit) represented by a Nx1 matrix (where N is a power of 2), it is often not possible to get 2x1 matrices representing the separate single qubits that form this qubit group. Therefore for code simplicity, checking for separability and separation is not implemented in the code. If more than one qubit are put together in a group, operations can be performed on the qubits forming the qubit group, but the measurement is done on the group as a whole.

Secondly, after the implementation of the Dual gates and grouping up to 3 qubits, the application of a dual gate on 2 qubits of the 3 qubit cluster came to rise as a problem, which qubits to choose for the dual gate to be applied on? It was decided to always apply the dual gate on the last 2 qubits from the right, and to implement two new Tri gates (matrix wise that were not implemented physically) which are:

Circular Shift Left Gate: removes the left-most qubit and appends it to the right.

Circular Shift Right Gate: removes the right-most qubit and appends it to the left.

$$Circular\ Shift\ Left: \begin{bmatrix} 1&0&0&0&0&0&0&0\\ 0&0&1&0&0&0&0&0\\ 0&0&0&0&1&0&0&0\\ 0&0&0&0&0&0&1&0\\ 0&1&0&0&0&0&0&0\\ 0&0&0&1&0&0&0&0\\ 0&0&0&0&0&1&0&0\\ 0&0&0&0&0&0&0&1 \end{bmatrix}$$

$$Circular\ Shift\ Right: \begin{bmatrix} 1&0&0&0&0&0&0&0\\ 0&0&0&0&1&0&0&0\\ 0&1&0&0&0&0&0&0\\ 0&0&0&0&0&1&0&0\\ 0&0&1&0&0&0&0&0\\ 0&0&0&0&0&0&1&0\\ 0&0&0&1&0&0&0&0\\ 0&0&0&0&0&0&0&1 \end{bmatrix}$$

3.2 Implementation

3.2.1 Algebra package

A complex number java class was used in the implementation of the quantum computer, url: `https://github.com/abdulfatir/jcomplexnumber/blob/master/com/abdulfatir/jcomplexnumber/ComplexNumber.java`

and Matrix multiplication was done with the help of an answer from stack-overflow url: `https://stackoverflow.com/questions/17623876/matrix-multiplication-using-arrays`

The previous two references were used to create the package where the classes (Complex number and Matrix) exist, some modifications were added to the complex number class as the ability to create a complex number with the magnitude and phase angle. Moreover, the Kronecker Product was implemented to the Matrix Class.

3.2.2 Quantum package

Qubit : A java class that extends MultiQubit class as it is a special case of it with number of qubits=1.

MultiQubit: A java class that simulates the behaviour of a group of qubits with a Ax1 matrix where A is (n power 2), and n is the number of qubits the instance represents.

3.2.3 Gates package

Gate: a generic Abstract class for the basis of a Gate class, it contains a matrix and number of rows and columns of the matrix.

Package: **gates.single** contains the Classes that implement the gates (H,I,RootNot,S,T,X,Y,Z) representing their matrices respectively

Package: **gates.dual** contains the Classes that implement the gates (Swap,RootSwap and the coupling of XX,YY,ZZ) representing their matrices respectively. Moreover a class is made representing (Controlled Single gate) that takes a parameter of the type gate (represented as an enum) that is to determine the gate to be controlled.

Package: **gates.tri** contains the classes that implement the gates (Controlled CNOT, Controlled Swap, Circular Shift Right and Circular shift left) representing their matrices respectively.

SingleGates: a collection class that apply single gates on qubits.

DualGates: a collection class that apply dual gates on qubits.

TriGates: a collection class that apply tri gates on qubits.

3.2.4 Exceptions package

Contains the necessary exception classes to describe correctly what happened wrong when an error occures.

References

[1] Zeilinger A. Bouwmeester D. *The Physics of Quantum Information: Basic Concepts*. Springer, Berlin, Heidelberg, 2000.

[2] Paul Dirac. *The Principles of Quantum Mechanics*. Oxford University Press, 1930.

[3] Isaac Nielsen, Michael A.; Chuang. *Quantum Computation and Quantum Information*. Cambridge University Press, 2000.