ISSN 2278-3091
Volume 5, No.4, July – August 2016
International Journal of Advanced Trends in Computer Science and Engineering
Available Online at http://www.warse.org/IJATCSE/static/pdf/file/ijatcse06542016.pdf

Quantum Computers – Algorithms

Ravi Katukam
CEO
Ennovator
Hyderabad
kkravikiran@gmail.com

Nishank Kumar Pandey
2nd Yr, B.E, CSE
SIT, Tumkur
Karnataka – 572103
Ph: 8904748916
nishank127@gmail.com

Archana Sathish Kumar
2nd Yr, B.E, CSE
SIT, Tumkur
Karnataka – 572103
Ph: 9620488852
archana13696@gmail.com

Kushal Nagaraj
2nd Yr, B.E, CSE
SIT, Tumkur
Karnataka – 572103
Ph: 9632197231
kushal.nagaraj3101996@gmail.com

ABSTRACT
In this paper, we describe the algorithms related to Quantum Computers available and under research.

General Terms
Algorithms, Documentation, Performance, Design, Reliability, Experimentation, Theory.

Keywords
Quantum computing, Algorithms, IBM Quantum computers, Qubits.

1. INTRODUCTION
In quantum computing, a quantum algorithm is an algorithm which runs on a realistic model of quantum computation, the most commonly used model being the quantum circuit model of computation. A classical (or non-quantum) algorithm is a finite sequence of instructions, or a step-by-step procedure for solving a problem, where each step or instruction can be performed on a classical computer. Similarly, a quantum algorithm is a step-by-step procedure, where each of the steps can be performed on a quantum computer. Although all classical algorithms can also be performed on a quantum computer, the term quantum algorithm is usually used for those algorithms which seem inherently quantum, or use some essential feature of quantum computation such as Quantum Superposition or quantum entanglement. Problems which are unsolvable using classical computers remain unsolvable using quantum computers. What makes quantum algorithms interesting is that they might be able to solve some problems faster than classical algorithms.

2. Quantum Algorithms
2A.Changing the direction of a CNOT gate

In this algorithm, we are reversing the actions of target and control qubits keeping the CNOT gate as it is. The CNOT gate will be having Q1 as the control qubit and Q2 as the target qubit. By using Hadamard gate, we can get reverse results. The following equations are enough to convince the reversible property.

$$(H \otimes H)CNOT_{12}(H \otimes H) = CNOT_{21}.$$

The Composer View

The Quantum Source File

```
x q[2];
h q[1];
h q[2];
cx q[1], q[2];
h q[1];
h q[2];
measure q[1];
measure q[2];
```

The Distribution File

The CSV

Shots	Value	Probability	Qubits Measured
1	00	0	1,2
1	01	0	1,2
1	10	0	1,2
1	11	1	1,2

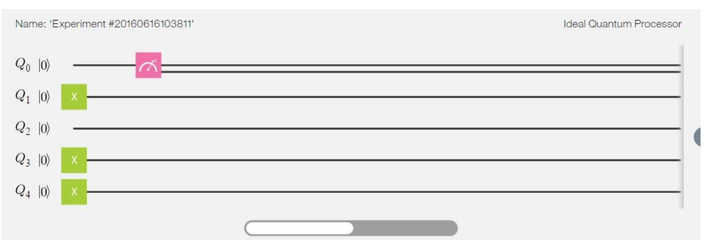
The Result Analysis

The result obtained is 11. According to the truth table of CNOT gate, 10 should be the input to get 11. As we are implementing reverse CNOT gate, 01 is giving us 11.

2B.Even Odd Detector

In this algorithm, we are trying to detect the nature of the 5 Qubit number. If the given number is even, the output will be 0 else the output will be 1. The logic here is, the least significant bit is itself the output.

The Composer View



The Quantum Source File

```
x q[1];
x q[3];
x q[4];
measure q[0];
```

The Distribution File

The CSV

Shots	Value	Probability	Qubits Measured
1	0	1	0
1	1	0	0

The Result Analysis

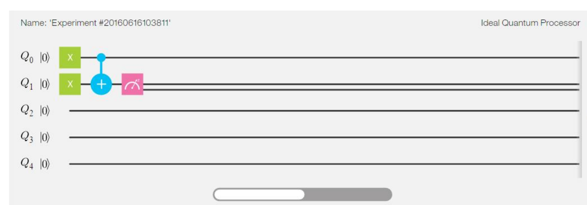
Here Q0 is considered as the least significant Qubit. The input number is 11010, which is a even number. Hence the output is 0. This can also be checked for odd numbers.

2C.Addition of Two Qubits

In this algorithm, we are adding 2 qubits according to the boolean algebra. The truth table of Boolean addition is as follows.

Q1	Q2	SUM
0	0	0
0	1	1
1	0	1
1	1	0

The Composer View



The Quantum Source File

```
x q[0];
x q[1];
cx q[0], q[1];
measure q[1];
```

The Distribution File

The CSV

Shots	Value	Probability	Qubits Measured
1	0	1	1
1	1	0	1

The Result Analysis

In the above example, the input given is 11. According to the truth table of boolean addition, the output should be 0. We are getting the same using composer also. This can be checked for other inputs also.

2D.Grover's Algorithms

Grover's algorithm is a quantum algorithm that finds with high probability the unique input to a black box function that produces a particular output value, using just O(*N*1/2)evaluations of the function, where *N* is the size of the function's domain. It was originated by Lov Grover in 1996.

Application

The purpose of Grover's algorithm is usually described as "searching a database". More accurately, its described as “inverting a function”. Roughly speaking, if we have a function y = f(x) that can be evaluated on a quantum computer, Grover's algorithm allows us to calculate x when given y. Inverting a function is related to the searching of a database because we could come up with a function that produces one particular value of y("true" for instance) if x matches a desired entry in a database, and another value of y ("false") for other values of x.

Grover's algorithm can also be used for estimating the mean and median of a set of numbers, and for solving the Collision problem. The algorithm can be further optimized if there is more than one matching entry and the number of matches is known beforehand.

Problem Statement

Consider an unsorted database with N entries. The algorithm requires an N-dimensional state space H, which can be supplied by n=log2 N qubits. Consider the problem of determining the index of the database entry which satisfies some search criterion. Let f be the function which maps database entries to 0 or 1, where f(ω)=1 if and only if ωsatisfies the search criterion. We are provided with (quantum black box) access to a subroutine in the form of a unitary operator, Uω, which acts as follows (for the ω for whichf(ω)=1):

$$U_\omega|\omega\rangle = -|\omega\rangle$$
$$U_\omega|x\rangle = |x\rangle \qquad \text{for all basis } x \neq \omega$$

Our goal is to find w.

Algorithm

Let s denote uniform super imposition over all state.

$$|s\rangle = \frac{1}{\sqrt{N}}\sum_{x=0}^{N-1}|x\rangle.$$

Then the operator,

$$U_s = 2\,|s\rangle\,\langle s| - I$$

is known as “Grover Diffusion Operator”.

The algorithm is as follows,

1. Initialize system to the state,

$$|s\rangle = \frac{1}{\sqrt{N}}\sum_{x=0}^{N-1}|x\rangle.$$

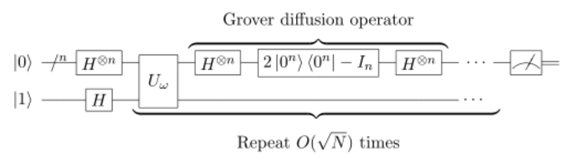
2. Perform the following Grover Itiration r(N) times. The function r(N) is described as,

1. Apply the operator Uw
2. Apply the operator Us.

3. Perform the measurement Ω. The measurement result will be eigenvalue λω with probability

approaching 1 for N≫1. From λω, ω may be obtained.

Quantum Circuit Representation

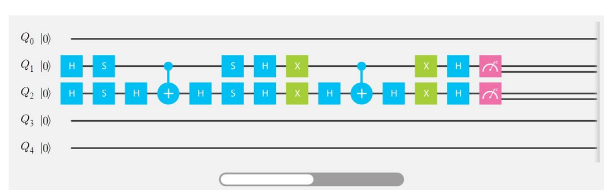


Simulation

For the simulation of Grover's Algorithm on a composer, we are taking 2 qubits and placing the required item in all the possible positions(00,10,01,11).

Case1: Item at 00

Composer View



The Quantum Source File

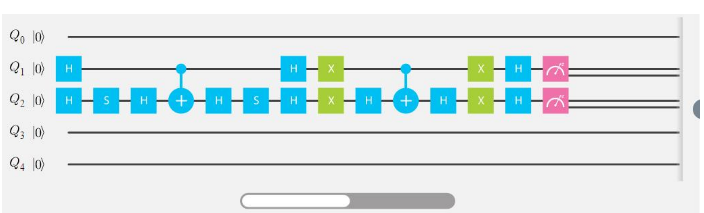
```
h q[1];
h q[2];
s q[1];
s q[2];
h q[2];
cx q[1], q[2];
h q[2];
s q[1];
s q[2];
h q[1];
h q[2];
x q[1];
x q[2];
h q[2];
cx q[1], q[2];
h q[2];
x q[1];
x q[2];
h q[1];
h q[2];
measure q[1];
measure q[2];
```

The Distribution Table

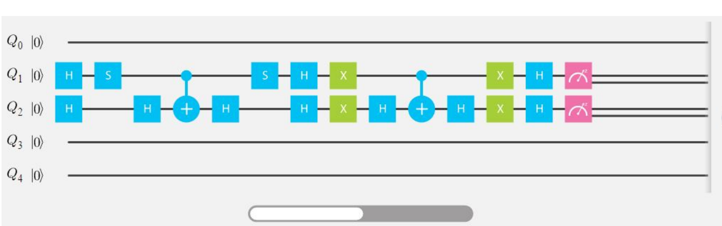
The Result Analysis

In the above exapmle, the item we want is in position 00. For checking the efficience of Grover's algorithm we have initialized the item to 00. After the items are processed using the circuit for Grover's algorithm, we get the output 00. Hence Grover's algorithm is implemented.

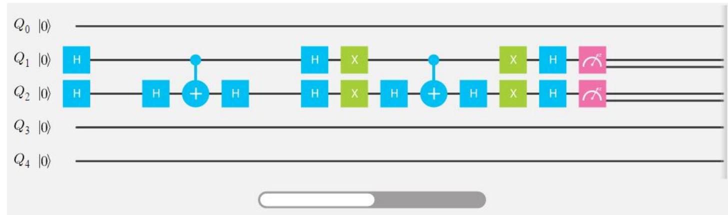
Case2 : Item at 01



Case3 : Item at 10



Case4 : Item at 11



2E.Swapping the states of qubits

"Swap," demonstrates a building block that allows you to permute the information stored in a set of qubits. Suppose we want to exchange the states of a pair of qubits by implementing a SWAP gate on the pair. There is no SWAP gate in our basis, but we can construct one from three CNOT gates -

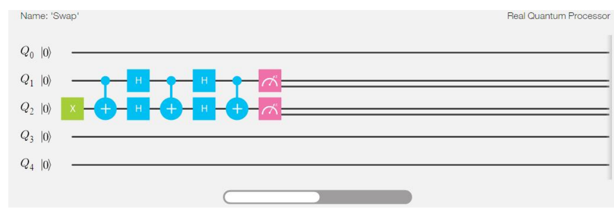
SWAP12 = CNOT12 CNOT21 CNOT12

For example: consider classical state 01.

The first gate CNOT12 does nothing since the control is 0.

The second gate CNOT21 flips the first qubit, so we have 11. Finally, the last CNOT12 flips the second qubit and we get 10.

The Composer View



The Quantum Source File

```
x q[2];
cx q[1], q[2];
h q[1];
h q[2];
cx q[1], q[2];
h q[1];
h q[2];
cx q[1], q[2];
measure q[1];
measure q[2];
```

The Distribution Table

The CSV

Shots	Value	Probability	Qubits Measured
1	0	0	1,2
1	1	0	1,2
1	10	1	1,2
1	11	0	1,2

The Result Analysis

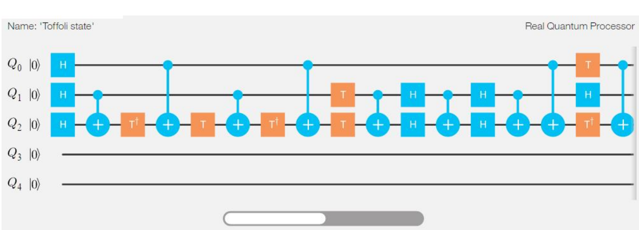
In the circuit shown above, the input given is 01. After the inputs are processed, the obtained output will be 10. Hence it the circuit works as a swapping machine.

2f.Toffoli gate

Classical approach	Quantum approach
We have AND gate that is important for computation of non-linear transformation. But it is irreversible ie, if we get an output 0, then we have three input combinations 00,01,10 so we can't predict the input for which the output is 0.	This drawback of classical approach is overcome by Toffoli gate, which implements AND gate reversibly using 3 wires.

TOF | A,B,C> = |a,b,(a AND b) XOR c>

The Composer View



The Quantum Source File

```
h q[0];
h q[1];
h q[2];
cx q[1], q[2];
tdg q[2];
cx q[0], q[2];
t q[2];
cx q[1], q[2];
tdg q[2];
cx q[0], q[2];
t q[1];
t q[2];
cx q[1], q[2];
h q[1];
h q[2];
cx q[1], q[2];
h q[1];
h q[2];
cx q[1], q[2];
cx q[0], q[2];
t q[0];
h q[1];
tdg q[2];
cx q[0], q[2];
measure q[0];
measure q[1];
measure q[2];
```

The Distribution Table

The CSV

Shots	Value	Probability	Qubits Measured
1	0	0.25	0,1,2
1	1	0.25	0,1,2
1	10	0	0,1,2
1	11	0	0,1,2
1	100	0.25	0,1,2
1	101	0	0,1,2
1	110	0	0,1,2
1	111	0.25	0,1,2

The Result Analysis

According to the input given in the composer, the output is equally distributed between 000,001,100,111.

2G.Quantum Phase Estimation

In quantum computing, the **quantum phase estimation algorithm** is a quantum algorithm that finds many applications as a subroutine in other algorithms. The quantum phase estimation algorithm allows one to estimate the eigenphase of an eigenvector of a unitary gate, given access to a quantum state proportional to the eigenvector and a procedure to implement the unitary conditionally.

The Problem Statement

Let U be the unitary operator that operates on m qubits. Then all of the eigenvalues of *U* have absolute value 1. Thus the spectrum of a unitary operator consistes of phases $e^{i\theta}$.

Given an eigenvector $|\psi\rangle$,such that $U|\psi\rangle = e^{i\theta}|\psi\rangle$. The objective is to find θ.

The phase estimation algorithm solves this problem.

Algorithm

Suppose we wish to compute the phases to an accuracy of n bits. We achieve this by subjecting our eigenvector $|\psi\rangle$ of U to a succession of n controlled operators, followed by the inverse of the QuantumFourier Transform. The controlled operators are the powers of U grom U to controlled $U^{2^{n-1}}$

After putting the control lines into the Hadamard state, we have

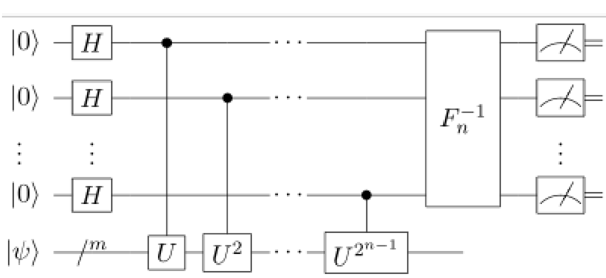
$$\frac{1}{\sqrt{2^n}}\sum_x |x\rangle \otimes |\psi\rangle.$$

After the controlled application of U we have,

$$\frac{1}{\sqrt{2^n}}\sum_x e^{ix\theta}|x\rangle \otimes |\psi\rangle.$$

Applying the inverse of the QuantumFourier Transform upon the *n* qubits yields

$$\frac{1}{2^n}\sum_y\sum_x e^{-2\pi i x\cdot y/2^n} e^{ix\theta}|y\rangle \otimes |\psi\rangle = \frac{1}{2^n}\sum_y \frac{e^{i2^n\theta}-1}{e^{i(\theta-2\pi y/2^n)}-1}|y\rangle \otimes |\psi\rangle.$$

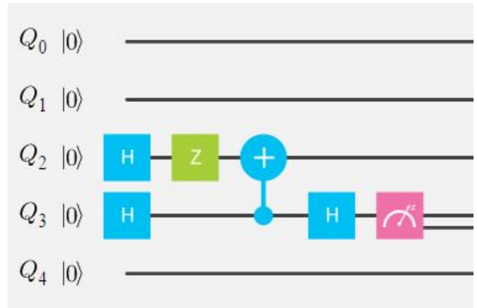


Quantum Circuit representation

Simulation

1. Phase Estimation Circuit(-)

The Composer View

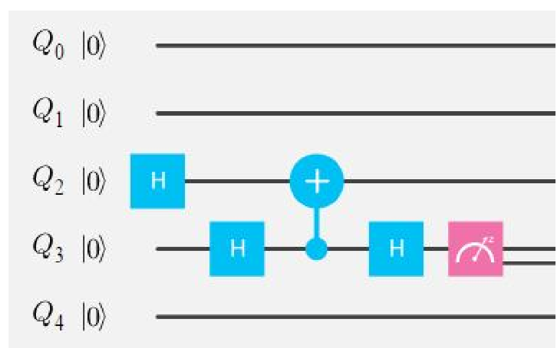


The Quantum Source File

```
h q[2];
h q[3];
z q[2];
cx q[3], q[2];
h q[3];
measure q[3];
```

2. Phase Estimation Circuit(+)

The Composer View



The Quantum Source File

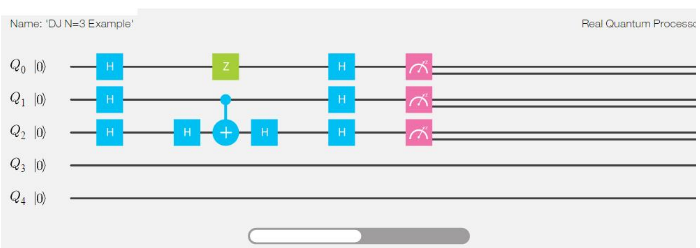
```
h q[2];
h q[3];
cx q[3], q[2];
h q[3];
measure q[3];
```

2H.Deutschmark-Jozsa Algorithm

In this algorithm, we find whether a function is a constant or a balanced function and it should be promised to be either of the cases. For the function f(x) with n-bits, if f(x)=1 then it is a constant or else its a balanced function.

Classical approach	Quantum approach
*In this approach, the probability of getting 1 is more than 0, if f(x) is constant then output is true for many cases and in case f(x) is balanced then output may be false. *for this linear search classical computers may take time complexity O(n).	*In this approach, the black-box problem can be solved efficiently with no errors and with less queries. *for this linear search quantum computers may solve the problem by exponentially faster than classical computers.

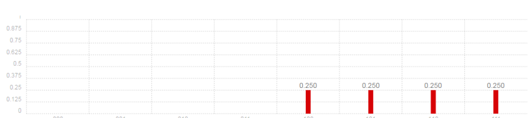
The Composer View



The Quantum Source File

```
h q[0];
h q[1];
h q[2];
h q[2];
z q[0];
cx q[1], q[2];
h q[2];
h q[0];
h q[1];
h q[2];
measure q[0];
measure q[1];
measure q[2];
```

The Distribution Table



The CSV

Shots	Value	Probability	Qubits Measured
1	0	0	0,1,2
1	1	0	0,1,2
1	10	0	0,1,2
1	11	0	0,1,2
1	100	0.25	0,1,2
1	101	0.25	0,1,2
1	110	0.25	0,1,2
1	111	0.25	0,1,2

3. ACKNOWLEDGMENTS

Our thanks to **Mr. RAVI KATUKAM,** Director, Disruptive Innovative Group, ValueLabs, Hyderabad for guiding us through this research work.

4. CONCLUSION

The area of implementing algorithms is vast. Day by day the number of algorithms successfully implemented on quantum computers is increasing. It's a tough task to understand every bit of the quantum working. But the research is going on. Hope to see the quantum computers in our hands one day.

REFERENCES

[1] http://www.sciencealert.com/scientists-have-discovered-a-material-that-could-create-quantum-optical-computers

[2] http://phys.org/news/2015-12-quantum-standard-semiconductor-materials.html#jCp

[3] http://www.popsci.com/scitech/article/2009-06/possible-silicon-replacement

[4] http://www.dwavesys.com/tutorials/background-reading-series/introduction-d-wave-quantum-hardware

[5] **https://quantumexperience.ng.bluemix.net**