jet INTL JOURNAL OF ELECTRONICS AND TELECOMMUNICATIONS, 2022, VOL. 68, NO. 1, PP. 77-82
Manuscript received October 20, 2021; revised January, 2022. DOI: 10.24425/ijet.2022.139851

Design of Grover's Algorithm over 2, 3 and 4-Qubit Systems in Quantum Programming Studio

Diana Jingle, Shylu Sam, Mano Paul, Ananth Jude, and Daniel Selvaraj

***Abstract*—In this paper, we design and analyse the Circuit for Grover's Quantum Search Algorithm on 2, 3 and 4-qubit systems, in terms of number of gates, representation of state vectors and measurement probability for the state vectors. We designed, examined and simulated the quantum circuit on IBM Q platform using Quantum Programming Studio. We present the theoretical implementation of the search algorithm on different qubit systems. We observe that our circuit design for 2 and 4-qubit systems are precise and do not introduce any error while experiencing a small error to our design of 3-qubit quantum system.**

***Keywords*—quantum search; Oracle; Qubit; Hadamard transform; phase shift**

I. INTRODUCTION

QUANTUM Search [9], [10] is a technique that can be used to speed up many classical search algorithms. It performs a generic search for a potential solution to a very wide range of problems. For example, given a large integer of size N, the problem is to determine whether an numeral p is a non-trivial factor of N with no prior knowledge about the structure of the information. A simple strategy for finding the non-trivial factors is to search through the set until found. Classically, this problem requires approximately N operations, but a Grover's search algorithm requires approximately $\sqrt{N}$ operations with probability > 1/2.

Grover's Algorithm can be used to searching an actual database but no research has been published on this area. Researchers consider Oracle as Virtual Database to perform searching. The searching time on the database depends upon the size of the database and the quantum hardware. Therefore it is necessary to provide an analysis on designing the quantum circuit using simulation tools. In this paper, we used quantum programming studio for precisely designing the quantum circuit for Grover's search algorithm on 2, 3 and 4-qubit system so as to reduce the errors in the design. We have also presented the step-wise procedure of Grover's algorithm theoretically for better knowledge on the algorithm.

The paper is organized as follows: Section 1 presents the significance of Quantum Search Algorithm and its applications. Section 2 provides analysis of existing approaches and their observations on Grover's search algorithm. Section 3 presents a brief overview on the working principle of Grover's algorithm, Section 4 presents the implementation and performance results of Grover's Algorithm for 2, 3, and 4-qubit systems and finally we conclude in Section 5.

Diana Jingle is with Christ University, Bangalore India (e-mail: diana.jebajingle@christuniversity.in).
Shylu Sam is with Karunya Institute of Technology and Sciences, Coimbatore, (e-mail: shylusam@karunya.edu).
Mano Paul is with Alliance University, Bangalore (e-mail: mano.paul@alliance.edu.in).
Ananth Jude and Daniel Selvaraj are with Sri Krishna College of Engineering and Technology, Coimbatore (e-mail: jude.ananth@gmail.com, danielfrancisselvaraj@gmail.com).

II. LITERATURE SURVEY

In [2], a 4-qubit quantum algorithm was implemented on ibmqx5 architecture which was largely published and currently possible and the performance accuracy of their implementation results were compared with simulation results. Their implementation performance requires more hardware support as they faced a complexity in circuit design. In [4], a classical-quantum search algorithm was proposed to find the shortest path of a given graph. Grover's algorithm cannot be applied to search an item in an actual database. In [5], the theoretical operation of quantum error correction process was demonstrated using QCAD simulator with possible results. Grover's algorithm can be used to search a single object in an unsorted database. In [6], a generalized Grover's Algorithm was framed to support multi-object search in a large and unsorted database and they have provided consistent results with Grover's algorithm for single-object search. The authors in [8] implemented Grover's algorithm on a real ibmqx4 5-qubit quantum computer. A 4-qubit implementation of Grover's Search Algorithm was proposed in [11] with test results in which it was observed that quantum computers can accurately solve only simple problems with small amounts of data. In [14], a two-transmon-qubit interaction was designed in a circuit QED to improve the implementation time of Grover's search algorithm in nanosecond-scale. The Grover's algorithm was implemented using two trapped atomic ion qubits in order to improve the searching speed. In [15], the Grover's algorithm was tried in a scalable system to improve the success probability of the search item. Implementation of 2-Qubit quantum algorithm was performed in [12] using a linear optical chip and a 3-qubit quantum circuit was designed and implemented in [13].

III. GROVER'S QUANTUM SEARCH ALGORITHM

Lov Grover in 1996 [1] invented the Grover Search algorithm to perform the fastest possible quantum search for an unsorted database which takes $O(\sqrt{n})$ time and $O(\log N)$ storage space. This mean that, for a list of 108 elements, the number of iterations required to search a desired item is order of 10000 and the obtained result will be accurate with a probability close to 1. Although Grover's algorithm provides only a quadratic speedup, it is considerable when the size of N is large. TABLE 1 shows the Grover's algorithm and Fig. 1 shows its circuit representation.

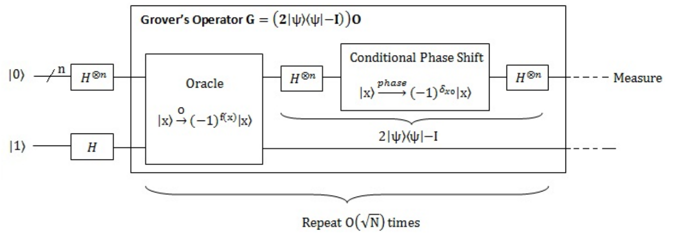


Figure 1. Circuit Representation of Grover's search algorithm

Consider an unsorted database with N entries forming a set $X = \{x_0, x_1, \ldots, x_{N-1}\}$ and given a boolean function $f: X \rightarrow \{0,1\}$ the goal is to find an element x in X such that $f(x) = 1$. Unstructured search is often alternatively formulated as a database search problem in which we are given a database and we want to find an item that meets some specification. For example, given a database of N names, we might want to find where your name is located in the database.

The search is called "unstructured" because we are given no guarantees as to how the database is ordered. If we were given a sorted database, for instance, then we could perform binary search to find an element in logarithmic time. Instead, we have no prior knowledge about the contents of the database. With classical circuits, we cannot do better than performing a linear number of queries to find the target element. The N elements are mapped to 0 to $N-1$ index numbers with $N = 2^n$.The search problem can be appropriately denoted by a function f, which takes an integer x as input from the indexes 0 to N −1. By definition, $f(x) = 1$ if x is found when searched, and $f(x) = 0$ if x is not found when searched. This can be expressed as,

$$f(x) = \begin{cases} 0 & \text{if } x \neq u \\ 1 & \text{if } x = u \end{cases}$$

where u is the element to be searched.

TABLE I:
GROVER'S QUANTUM SEARCH ALGORITHM

Input:
i) An Oracle O that transforms $|x\rangle|q\rangle \xrightarrow{O} |x\rangle|q \oplus f(x)\rangle$ where $f(x) = 0$ for all $0 \leq x < N-1$ and $f(x) = 1$ for $x = u$.
ii) State $|0\rangle$ with $n+1$ qubits.
Output: The matched item found. i.e. u
Runtime: Requires $O(\sqrt{N})$ operations and succeeds with a probability of $O(1)$.
Procedure:
Step 1: Start the algorithm with initial state $|0\rangle^{\otimes n}$.
Step 2: Apply Hadamard transform on the first n qubits and Hadamard and X gate on the last qubit as, $\frac{1}{\sqrt{N}}\sum_{x=0}^{N-1}|x\rangle\left[\frac{|0\rangle-|1\rangle}{\sqrt{2}}\right]$
Step 3: Perform Grover's Operation for $R \leq \left\lceil \frac{\pi}{4}\sqrt{\frac{2N}{M}} \right\rceil$ rounds.
Step 3.1: Apply the Oracle O such that, $|x\rangle \xrightarrow{O} (-1)^{f(x)}|x\rangle$
Step 3.2: Apply the Hadamard transform $H^{\otimes n}$ to the state obtained in step 3.1
Step 3.3: Perform Conditional Phase Shift operation on every state except the $|0\rangle$ state receiving a phase shift of −1, based on the equation, $|x\rangle \xrightarrow{phase} (-1)^{\delta_{x0}}|x\rangle$.
Step 3.4: Apply the Hadamard transform $H^{\otimes n}$. The final state obtained from Grover operation will be,
$[(2|\psi\rangle\langle\psi| - I)O]^R \frac{1}{\sqrt{N}}\sum_{x=0}^{N-1}|x\rangle\left[\frac{|0\rangle-|1\rangle}{\sqrt{2}}\right] \approx u\left[\frac{|0\rangle-|1\rangle}{\sqrt{2}}\right]$
Step 4: Measure the first n qubits. The result will be the matched item u.

The step-wise explanation of the Grover's algorithm is described as follows: The algorithm begins in the state$|0\rangle^{\otimes n}$. Now apply Hadamard transform $H^{\otimes n}$ on the first n qubits and Pauli X transform and Hadamard transform (HX) to the last qubit to obtain equal superposition state; the operation is described as,

$$H^{\otimes n}|0\rangle^{\otimes n} = |\psi\rangle = \frac{1}{\sqrt{N}}\sum_{x=0}^{N-1}|x\rangle \quad (2)$$

$$HX|1\rangle = \left[\frac{|0\rangle-|1\rangle}{\sqrt{2}}\right] \quad (3)$$

where $|\psi\rangle = \frac{1}{\sqrt{N}}\sum_i |i\rangle$ is the superposition of all basis states. Now apply the Grover's operator. The quantum circuit of Grover iteration is broken up into four steps namely, i) Oracle function, ii) Hadamard transform, iii) Conditional phase shift operation on every computational basis state except |0> receiving a phase shift of −1 and iv) Hadamard transform. The oracle is a unitary operator, O, which performs the following operation:

$$|x\rangle|q\rangle \xrightarrow{O} |x\rangle|q \oplus f(x)\rangle \quad (4)$$

where $|x\rangle$ is the input register that stores the index of all elements from 0 to N-1, $\oplus$ denotes addition modulo 2, $|q\rangle$ is the oracle qubit which is flipped if $f(x) = 1$, and is unchanged otherwise. The term ancilla or ancillary qubit can be used to refer the oracle qubit $|q\rangle$ as it adds some extra qubits for the algorithm. If x is not found, applying the oracle to the state does not change the state. But if x is found, then the final state is changed to a new state. The oracle used for Grover's algorithm is very simple. Applying the oracle to inputs $|x\rangle|q\rangle$ will produce output as $|x\rangle|q \oplus f(x)\rangle$. The first two qubits are unmodified (in case when $x = \{0,1\}$). The third qubit is flipped if the first two qubits are 1. Thus we get,

$$|x\rangle\left(\frac{|0\rangle-|1\rangle}{\sqrt{2}}\right) \xrightarrow{O} (-1)^{f(x)}|x\rangle\left(\frac{|0\rangle-|1\rangle}{\sqrt{2}}\right) \quad (5)$$

Equation (5) can be written as,

$$|x\rangle \xrightarrow{O} (-1)^{f(x)}|x\rangle \quad (6)$$

Steps 2, 3 and 4 can be represented as, $H^{\otimes n}(2|0\rangle\langle 0|)\, H^{\otimes n} = 2|\psi\rangle\langle\psi| - I$ and the Grover's operator can be denoted as, $G = (2|\psi\rangle\langle\psi| - I)O$. The Grover's algorithm is regarded as a rotation of R times within an angle $\theta/2 \leq \pi/4$ of $|\beta\rangle$ and θ is a real number ranging from 0 to π/2 and it can be expressed in superposition state as,

$$|\psi\rangle \equiv \sqrt{\frac{N-M}{N}}\,|\alpha\rangle + \sqrt{\frac{1}{N}}\,|\beta\rangle \quad (7)$$

where $|\alpha\rangle \equiv \frac{1}{\sqrt{N-M}}\sum_{x\neq u}|x\rangle$ and $|\beta\rangle \equiv |u\rangle$. The rotation can be expressed as, $R \leq \left\lceil \frac{\pi}{4}\sqrt{\frac{2N}{M}} \right\rceil$ and therefore, $R = O\left(\sqrt{N/M}\right)$. If $M = 1$, ie., if the solution is found in one iteration of Grover's algorithm, then $R = O(\sqrt{N})$.

IV. IMPLEMENTATION AND PERFORMANCE ANALYSIS

In this section, the system design, implementation and performance results are discussed for 2-qubit, 3-qubit and 4-

qubit system. We implemented the algorithm on Quantum Programming Studio, a web-based IDE and simulator for designing, inspecting and simulating/running quantum algorithms on platforms like IBM Q, Rigetti and Google Cirq. Quantum-circuit is open source simulator implemented in JavaScript. A Quantum-circuit runs 20+ qubit simulations at server (node.js) or in browser. Quantum Circuit can be imported from OpenQASM and Quil. After designing the Quantum circuit in IDE, it can also be exported to OpenQASM, pyQuil, Quil, Qiskit, Cirq, TensorFlowQuantum, QSharp, and QuEST, so it can be used for conversion between quantum programming languages. It is observed that, each quantum gate when applied, creates a small error to the quantum state. This is due to the circuit design and the interaction of each qubit with other qubits and gates.

1) Implementation with 2-Qubits

Assume $n = 2$, the possible states are $N = 2^2 = 4$ and suppose the solution is found at index 2 i.e., state $|10\rangle$, then we get,

$$|\psi\rangle \xrightarrow{O} \frac{1}{2}(|00\rangle + |01\rangle - |10\rangle + |11\rangle) \tag{8}$$

Thus the action of oracle operator on the state $|x\rangle$, changes the amplitude of that state. Now apply the Hadamard transform $H^{\otimes n}$ on these basis states and the result is shown as,

$$|\psi\rangle \xrightarrow{H^{\otimes 2}} \frac{1}{2}(|00\rangle - |01\rangle + |10\rangle + |11\rangle) \tag{9}$$

Now perform a conditional phase shift on every computational basis state except $|0\rangle$. Thus we get,

$$|\psi\rangle \xrightarrow{phase} \frac{1}{2}(|00\rangle + |01\rangle - |10\rangle - |11\rangle) \tag{10}$$

Again apply the Hadamard transform $H^{\otimes n}$ to the obtained superposition states, we get the matched element on an index and for our example it is given as,

$$|\psi\rangle \rightarrow |10\rangle \tag{11}$$

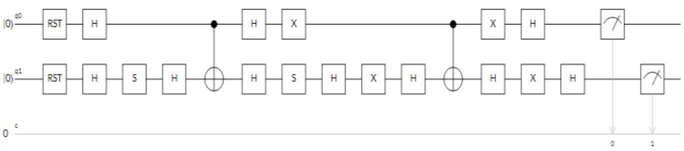


Fig. 2. Quantum Circuit Diagram for the state $|10\rangle$

State vector

00	0.00000000+0.00000000i	0.00000%
01	0.00000000+0.00000000i	0.00000%
10	-1.00000000+0.00000000i	100.00000%
11	0.00000000+0.00000000i	0.00000%

Fig. 3. Vector Representation for the state $|10\rangle$

Thus when $n = 2$ for the state $|10\rangle$, we got the desired search with just one iteration of the Grover's algorithm. Figure 2 shows the quantum circuit diagram for 2-qubit system with 22 gates. The circuit of 2-qubit system produces no error and thus it is free from random noise in the system. The vector representation and measurement probability for the 2-qubit state are shown in Figure 3 and Figure 4 respectively.

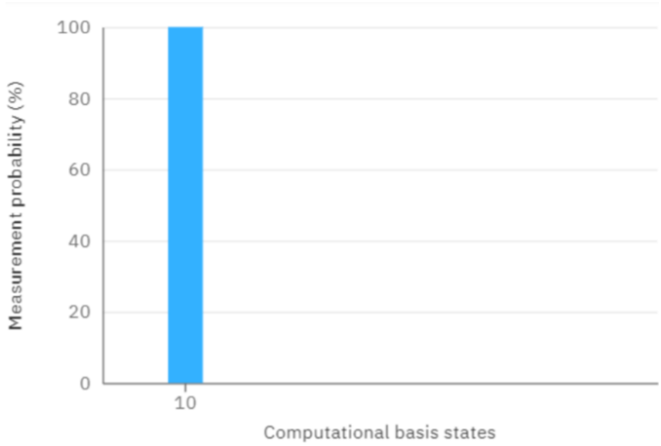


Fig. 4. Measurement Probability of 2-qubit system for the state $|10\rangle$

2) Implementation with 3-Qubits

Figure 5 shows the quantum circuit diagram for 3-qubit system with 39 gates. The circuit of 3-qubit system produces high error rate and thus introduces random noise in the system. The vector representation and measurement probability for the 3-qubit state are shown in Figure 6 and Figure 7 respectively. When $n = 3$, the possible states are $N = 2^3 = 8$, and suppose the solution is found at index 4 in state $|100\rangle$, then we get,

$$|\psi\rangle \xrightarrow{O} \frac{1}{2}(|000\rangle + |001\rangle + |010\rangle + |011\rangle + |100\rangle + |101\rangle + |110\rangle + |111\rangle) \tag{12}$$

Thus the action of oracle operator on the $|x\rangle$, changes the amplitude of that state. Now apply the Hadamard transform $H^{\otimes n}$ on these basis states and the result is:

$$|\psi\rangle \xrightarrow{H^{\otimes 2}} \frac{1}{2}(|000\rangle + |001\rangle + |010\rangle + |011\rangle - |100\rangle + |101\rangle + |110\rangle + |111\rangle) \tag{13}$$

After performing a conditional phase shift, we get 2^3 different states in superposition as,

$$|\psi\rangle \xrightarrow{phase} \frac{1}{2}(|000\rangle - |001\rangle - |010\rangle - |011\rangle + |100\rangle - |101\rangle - |110\rangle - |111\rangle) \tag{14}$$

Applying the Hadamard transform $H^{\otimes n}$ to the obtained states, we get the matched element on an index ande it is given as,

$$|\psi\rangle \rightarrow |100\rangle \tag{15}$$

Thus for $n = 3$, the desired state is obtained in 2 iterations of the Grover's algorithm.

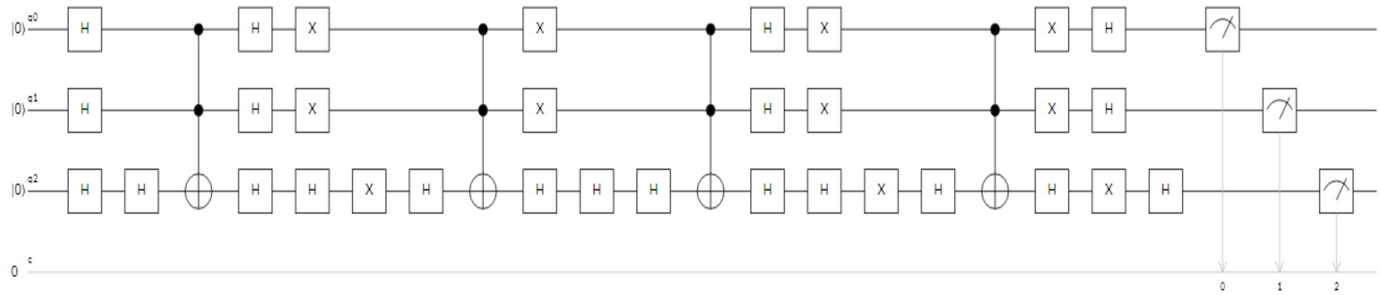
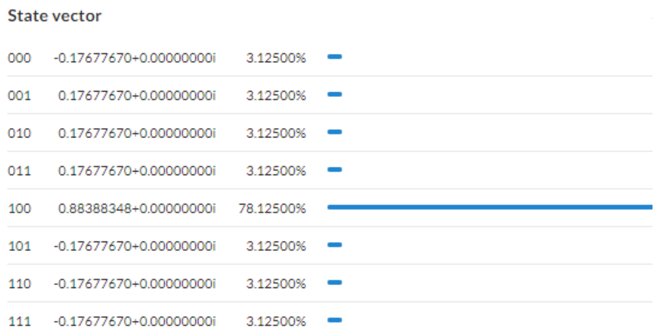


Fig. 5. Quantum Circuit Diagram for the state |100⟩



State vector

000	-0.17677670+0.00000000i	3.12500%	
001	0.17677670+0.00000000i	3.12500%	
010	0.17677670+0.00000000i	3.12500%	
011	0.17677670+0.00000000i	3.12500%	
100	0.88388348+0.00000000i	78.12500%	
101	-0.17677670+0.00000000i	3.12500%	
110	-0.17677670+0.00000000i	3.12500%	
111	-0.17677670+0.00000000i	3.12500%	

Fig. 6. Vector Representation of 3-qubit system for the state |100⟩

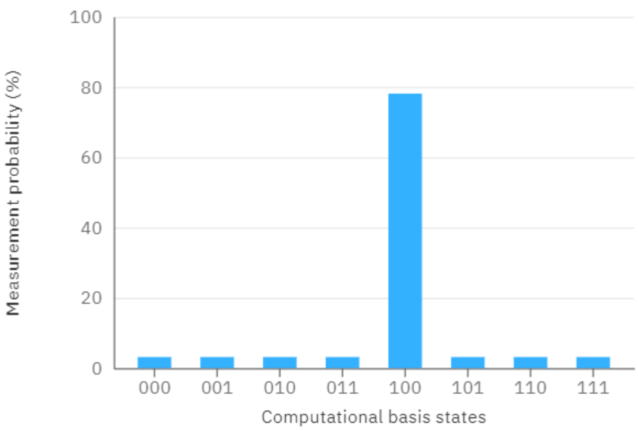


Fig. 7. Measurement Probability of 3-qubit system for the state |100⟩

3) Implementation with 4-Qubits

Figure 9 shows the quantum circuit diagram for 4-qubit system with 34 gates and it does not produce any noise and error in the design of the system. The vector representation and measurement probability for the 2-qubit state are shown in Figure 10 and Figure 11 respectively. When $n = 4$, the possible states are $N = 2^4 = 16$, assume the solution is found at index 4 i.e., $|0100\rangle$, we get,

$$|\psi\rangle \xrightarrow{O} 1/2(|0000\rangle + |0001\rangle + |0010\rangle + |0011\rangle + |0100\rangle + |0101\rangle + |0110\rangle + |0111\rangle + |1000\rangle + |1001\rangle + |1010\rangle + |1011\rangle + |1100\rangle + |1101\rangle + |1110\rangle + |1111\rangle) \quad (16)$$

The state of the system after applying Hadamard transform $H^{\otimes n}$ on $|x\rangle$, is shown as:

$$|\psi\rangle \xrightarrow{H^{\otimes 2}} 1/2(|0000\rangle + |0001\rangle + |0010\rangle + |0011\rangle - |0100\rangle + |0101\rangle + |0110\rangle + |0111\rangle + |1000\rangle + |1001\rangle + |1010\rangle + |1011\rangle + |1100\rangle + |1101\rangle + |1110\rangle + |1111\rangle) \quad (17)$$

The result after performing a Conditional Phase Shift, on all the basis state except $|0000\rangle$ is shown as,

$$|\Psi\rangle \xrightarrow{phase} ½(|0000\rangle - |0001\rangle - |0010\rangle - |0011\rangle + |0100\rangle - |0101\rangle - |0110\rangle - |0111\rangle - |1000\rangle - |1001\rangle - |1010\rangle - |1011\rangle - |1100\rangle - |1101\rangle - |1110\rangle - |1111\rangle \quad (18)$$

$$|\psi\rangle \rightarrow |0100\rangle \quad (19)$$

After applying $H^{\otimes n}$ to the states, the matching is found. Thus for $n = 4$, the desired state is obtained in 3 iterations of the Grover's algorithm.

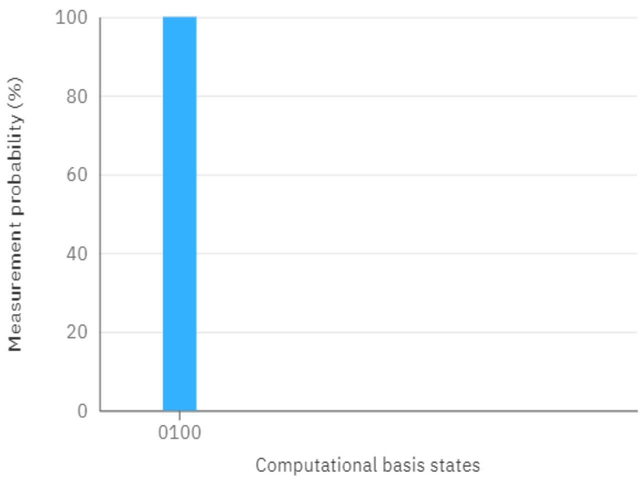


Fig. 8. Measurement Probability of 3-qubit system for the state |100⟩

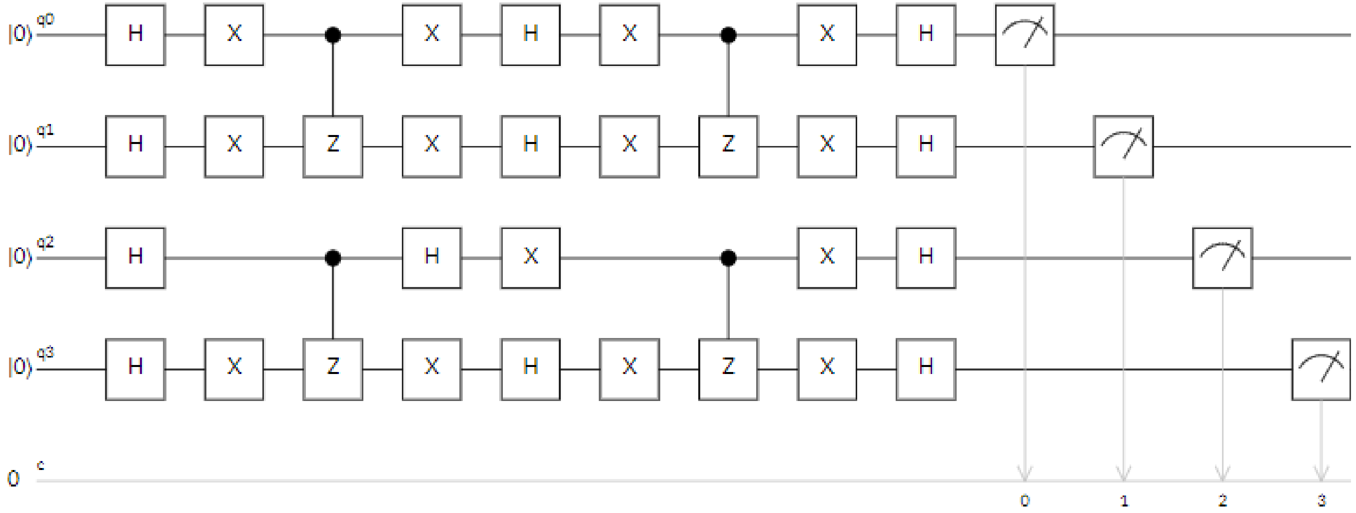


Fig. 9. Quantum Circuit Diagram for the state |0100⟩

State vector

0000	0.00000000+0.00000000i	0.00000%
0001	0.00000000+0.00000000i	0.00000%
0010	0.00000000+0.00000000i	0.00000%
0011	0.00000000+0.00000000i	0.00000%
0100	1.00000000+0.00000000i	100.00000%
0101	0.00000000+0.00000000i	0.00000%
0110	0.00000000+0.00000000i	0.00000%
0111	0.00000000+0.00000000i	0.00000%
1000	0.00000000+0.00000000i	0.00000%
1001	0.00000000+0.00000000i	0.00000%
1010	0.00000000+0.00000000i	0.00000%
1011	0.00000000+0.00000000i	0.00000%
1100	0.00000000+0.00000000i	0.00000%
1101	0.00000000+0.00000000i	0.00000%
1110	0.00000000+0.00000000i	0.00000%
1111	0.00000000+0.00000000i	0.00000%

Fig. 10. Vector Representation of 3-qubit system for the state |0100⟩

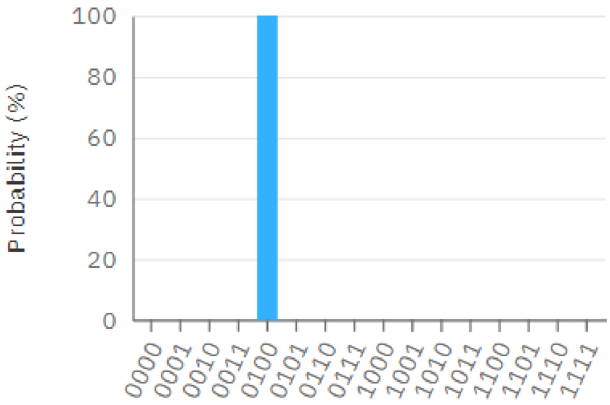


Fig. 11. Measurement Probability of 3-qubit system for the state |100⟩

Conclusion

Quantum Search Algorithms can be potentially applied i) for efficiently counting the number of solutions for a given search problem; ii) speedup the solution of NP-complete problems, iii) speed up the search for keys to cryptosystems [3], iv) finding the shortest path between two cities [7], and v) searching problems or extracting statistics in unstructured databases [6] more quickly than using classical computing. We have designed and implemented the Grover's quantum search algorithm on 2, 3 and 4-qubit systems and presented our analysis in terms of number of gates used, vector representation and measurement probability. We used Quantum Programming Studio tool for designing the circuit diagram precisely on IBM Q platform. We have also provided theoretical knowledge on the algorithm for 2, 3 and 4-qubit system. In future, we will plan to use the search algorithm for real-time scenario.

Acknowledgements

We are very grateful to experts for their appropriate and constructive suggestions to improve this template.

References

[1] Luan L., Wang Z., Liu S., "Progress of Grover Quantum Search Algorithm", *Energy Procedia*, vol. 16, pp. 1701 – 1706, 2012. https://doi.org/10.1016/j.egypro.2012.01.263

[2] Karlsson V. B., Stromberg P., "4-qubit Grover's alg. impl. for the ibmqx5 archit.", *Degree Project in Computer Science*, First Cycle, 15 Credits Stockholm, Sweden, 2018.

[3] Zalka C., "Could Grover's Quantum Algorithm Help in Searching an Actual Database?", *Quantum Physics*, 1999, pp. 1-7. https://doi.org/10.1103/PhysRevA.62.052305

[4] Aghaei M. R. S., Zukarnain Z. A., Mamat A., Zainuddin H., "A Hybrid Algorithm For Finding Shortest Path In Network Routing", *Journal Of Theoretical And Applied Information Technology*, 2009, pp. 360-365.

[5] Priya R. P., Baradeswaran A., "An efficient simulation of quantum error correction Codes", *Alexandria Engineering Journal*, Vol. 57, 2018, pp. 2167–2175. https://doi.org/10.1016/j.aej.2017.06.013

[6] Chen G., Fulling S. A., and Scully M. O., "Grover's Algorithm for Multiobject Search in Quantum Computing", *Article in Lecture Notes in Physics*, 1999, pp. 1-12.

[7] Hahanov V., Miz V., "Quantum computing approach for shortest route finding", *East-West Design & Test Symposium (EWDTS 2013)*, Rostov-on-Don, Russia, 2013, pp. 27-30. https://doi.org/10.1109/EWDTS.2013.6673095

[8] Abhijith J., Adedoyin A., Ambrosiano J., Anisimov P., Bärtschi A., Casper W., Chennupati G., Coffrin C., Djidjev H., Gunter D., Karra S., Lemons N., Lin S., Malyzhenkov A., Mascarenas D., Mniszewski S., Nadiga B., O'malley D., Oyen D., Pakin S., Prasad L., Roberts R., Romero P., Santhi N., Sinitsyn N., Swart P. J., Wendelberger J. G., Yoon B., Zamora R., Zhu W., Eidenbenz S., Coles P. J., Vuffray M. and Lokhov A. Y., "Quantum Algorithm Implementations For Beginners", *Computer Science Emerging Technologies*, 2020, pp. 1-94.

[9] Kaye P., Laflamme R., Mosca M., "An Introduction to Quantum Computing", Oxford University Press Inc., New York, 2007, pp. 1-276.

[10] Nielsen M. A. and Chuang I. L., "Quantum Computation and Quantum Information", *Cambridge University Press*, New York, 2010, pp. 1-676.

[11] Mandviwalla A., Ohshiro K., Ji B., "Implementing Grover's Algorithm on the IBM Quantum Computers", in Proc. *2018 IEEE Int. Conference on Big Data*, 2018,pp. 2531-2537. https://doi.org/10.1109/BigData.2018.8622457

[12] Samsonov E., Kiselev F., Shmelev Y., Egorov V., Goncharov R., Santev A., Pervushin B. and Gleim A., "Modeling two-qubit Grover's algorithm implementation in a linear optical chip", *Physica Scripta*, vol. 95, no. 4, 2020. https://doi.org/10.1088/1402-4896/ab6523

[13] Figgatt C., Maslov D., Landsman K. A., Linke N. M., Debnath S. and Monroe C., "Complete 3-Qubit Grover search on a programmable quantum computer", *Nature Communications*, vol. 8, no. 1918, 2017, pp. 1-9. https://doi.org/10.1038/s41467-017-01904-7

[14] Said T., Chouikh A., Essammouni K. and Bennai M., "Implementation of Grover quantum search algorithm with two transmon qubits via circuit QED", *Quant. Phys. Lett.*, vol. 6, no. 1, 2017, pp. 29-35. https://doi.org/10.18576/qpl/060105

[15] Brickman K. A., Haljan P. C., Lee P. J., Acton M., Deslauriers L. and Monroe C., "Implementation of Grover's Quantum Search Algorithm in a Scalable System", *Physical Review A*, vol. 72, no. 5, 2005, , pp. 1-4. https://doi.org/ 10.1103/PhysRevA.72.050306