Available online at www.worldscientificnews.com

World Scientific News

WSN 28 (2016) 41-56 EISSN 2392-2192

Deutsch-Jozsa Algorithm for Three Fermions System

Mezher B. Saleh[1,a], **Loay E. George**[2,b], **Kadhum J. Kadhum**[1,c]

[1]Department of Physics, College of Science, Al- Mustansiriyah University, Baghdad, Iraq

[2]Department of Computer Science, College of Science, Baghdad University, Baghdad, Iraq

[a-c]E-mail address: mezherbaker@yahoo.com , loayedwar57@yahoo.com , kadhum57@yahoo.com

ABSTRACT

One of the most important quantum algorithms is Dutsch-Jozsa algorithm. It is one of the most basic ways to demonstrate the power of quantum computation. In this research a system consists of three electron (fermions) in state |0> and one in state |1> the Hadamard transform was applied on all states. Then, the black box is applied to compute f. Finally it has been applied the Hadamard gate. It was proved this quantum information processing system can solve Deutsch's problem with one evaluation of the function f while the classical system requires ($2^n/2 + 1 = 5$) evaluations.

Keywords: Quantum Bits (Fermions), Quantum Gates, Boolean function, Dutsch-Jozsa algorithm, Hadamard transform, Deutsch's problem

I. INTRODUCTION

One of the most important algorithms is Dutsch-Jozsa algorithm. In earlier published work in the literature this algorithm was used for systems consist of one fermion (electron) and two fermions [1]. In this research project, we will use a quantum system consists of three fermions (three electrons).

- In 2003, Gulde et al. exploited techniques developed for nuclear magnetic resonance to implement the Deutsch-Jozsa algorithm on an ion-trap quantum processor. They considered the use of electronic and motional states of a single calcium ion as qubits. To them ion-based implementation of a full quantum algorithm serves to demonstrate experimental procedures with the quality and precision are required for complex computations, confirming the potential of trapped ions for quantum computation [2].
- In 2008, Batty et al. adapted the Deutsch-Jozsa algorithm to the context of formal language theory. Specifically, they used the algorithm to distinguish between trivial and nontrivial words in groups given by finite presentations, under the promise that a word is of a certain type. This was done by extending the original algorithm to functions of arbitrary length binary output, with the introduction of a more general concept of parity. They provided examples in which properties of the algorithm allow to reduce the number of oracle queries with respect to the deterministic classical case. This has some consequences for the word problem in groups with a particular kind of presentation [3].
- In 2012, Zhang et al. presented a theoretically advance a scheme to implement quantum Deutsch's algorithm in spin-orbital angular momentum space. Their proposed scheme exploits a newly developed optical device called "q-plate", which can couple and manipulate the spin-orbital angular momentum simultaneously. Theoretically, this experimental setup is of high stability and efficiency [4].
- In 2015, Kiktenko et al. investigated a scheme for realization of quantum algorithms using non composite quantum systems (i.e., systems without subsystems). In this framework, n artificially allocated “subsystems” play a role of qubits in n-qubits quantum algorithms. With focus on two-qubit quantum algorithms, they demonstrated a realization of the universal set of gates using an = 5 single qubit state. Manipulation with an ancillary level in the systems allows effective implementation of operators from U(4) group via operators from SU(5) group. Using a possible experimental realization of such systems through harmonic supercond- ducting many-level quantum circuits, they presented a blueprint for a single qubit realization of the Deutsch algorithm [5].

II. THEORY

The Deustch-Jozsa algorithm is one of the most basic ways to demonstrate the power of quantum computation. The Deutsch-Jozsa algorithm can be described as the following game: "Alice, in location 1, selects a number *x* from 0 to 2^n-1, and mail it as a letter to Bob, in location 2. Bob calculates some function f(x) and replies with the result, which is either 0 or 1. Bob use a function *f* which is of one of two types; either *f(x)* is constant for all values of *x*, or *f(x)* is balanced, that is, equal to 1 for exactly half of all possible *x* and 0 for the other half. Alice’s goal is to determine with certainty whether Bob has chosen a constant or a balanced function, corresponding with him as little as possible" [6-8].

A model quantum solution for Deustch-Jozsa algorithm or Deutsch’s problem is briefly presented, as all further analysis will stem from this. This is based on the expression given in which solves Deutsch’s problem with probability 1 using only one call to the quantum black box computing *f* [9]. A traditional classical algorithm would require two calls to a classical black box in order to determine if f is constant or balanced.

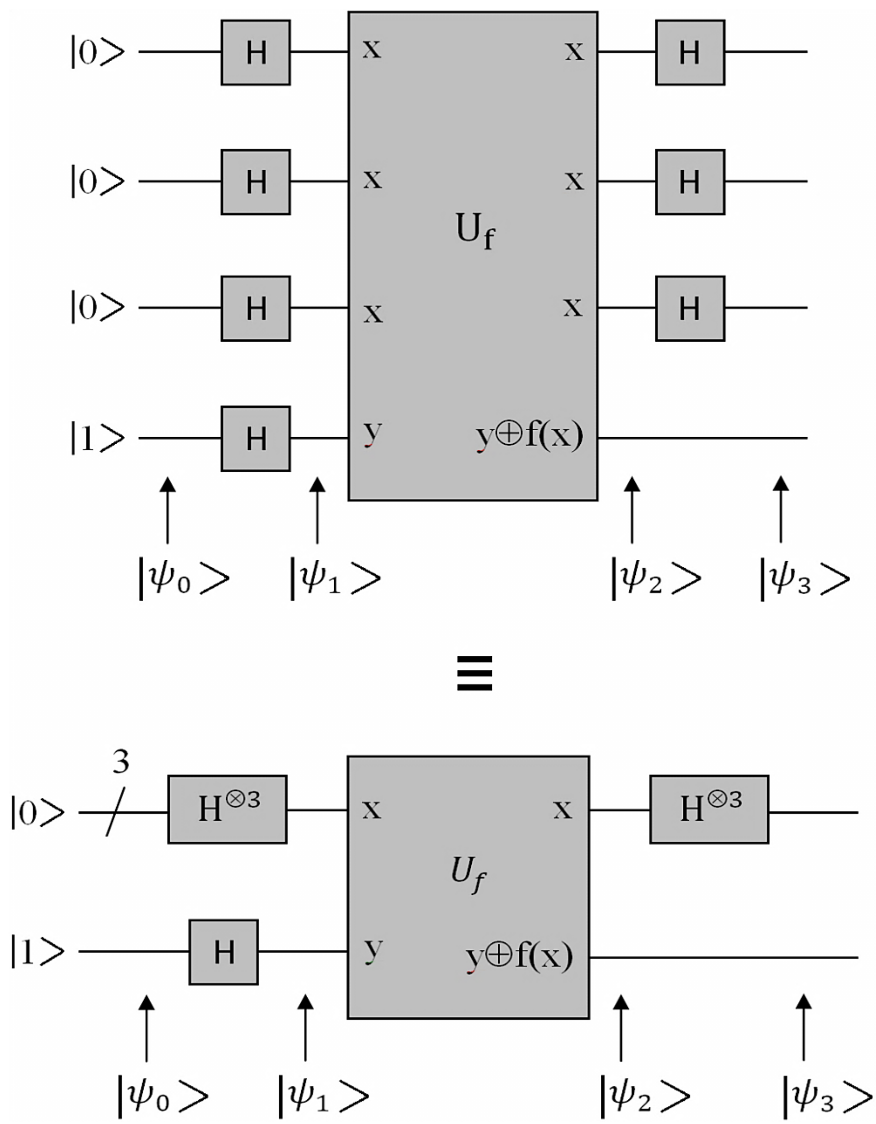

Fig. 1. Quantum circuit implementing the Deutsch-Jozsa algorithm.

The quantum black box extends the classical black box to operate on superposition principle of basis states. The quantum black box can be described by the unitary operator U_f representing an f – controlled - NOT
(f -CNOT) gate such that

$$U_f|x>|y>=|x>|y\oplus f(x)> \quad (1)$$

Noting that

$$\begin{aligned} U_f(U_f|x>|y>) &= U_f|x>|y\oplus f(x)> \\ &= |x>|y\oplus f(x)\oplus f(x)> \\ &= |x>|y> \end{aligned} \quad (2)$$

III. CALCULATION THE DEUTSCH-JOZSA ALGORITHM FOR THREE FERMIONS

The specific steps of the algorithm are depicted in Figure (1). Let us follow the states through this circuit. The input state:

$$|\psi_0>=|0>|0>|0>|1>=|0>^{\otimes 3}|1> \qquad (3)$$

Taking the initial state |0>0>0>1> and operating on it with a 4- qubit Hadamard gate $H^{\otimes 4}$:

$$H^{\otimes 4}|0>^{\otimes 3}|1>=H|0>H|0>H|0>H|1>$$
$$=\frac{1}{\sqrt{2}}\begin{bmatrix}1 & 1\\1 & -1\end{bmatrix}.\begin{bmatrix}1\\0\end{bmatrix}.\frac{1}{\sqrt{2}}\begin{bmatrix}1 & 1\\1 & -1\end{bmatrix}.\begin{bmatrix}1\\0\end{bmatrix}.\frac{1}{\sqrt{2}}\begin{bmatrix}1 & 1\\1 & -1\end{bmatrix}.$$
$$\begin{bmatrix}1\\0\end{bmatrix}.\frac{1}{\sqrt{2}}\begin{bmatrix}1 & 1\\1 & -1\end{bmatrix}\begin{bmatrix}0\\1\end{bmatrix}$$
$$=\frac{1}{\sqrt{2}}\begin{bmatrix}1\\1\end{bmatrix}.\frac{1}{\sqrt{2}}\begin{bmatrix}1\\1\end{bmatrix}.\frac{1}{\sqrt{2}}\begin{bmatrix}1\\1\end{bmatrix}.\frac{1}{\sqrt{2}}\begin{bmatrix}1\\-1\end{bmatrix}$$
$$=\frac{1}{\sqrt{2}}(|0>+|1>).\frac{1}{\sqrt{2}}(|0>+|1>).\frac{1}{\sqrt{2}}(|0>$$
$$+|1\quad>).\frac{1}{\sqrt{2}}(|0>-|1>)$$

Where, $|->=\frac{1}{\sqrt{2}}(|0>-|1>)$ so that:

$$H^{\otimes 4}|0>^{\otimes 3}|1>=\frac{1}{\sqrt{2^3}}(|0>+|1>).(|0>+|1>).(|0>+|1>).|->$$
$$|\psi_1>=H^{\otimes 4}|0>^{\otimes 3}|1>=\frac{1}{\sqrt{2^3}}[|000>+|001>+|010>+|011>+$$
$$|100>+|101>+|110>+|111>].|-> \qquad (4)$$

For a Boolean function $f:\{0,1\}^n \rightarrow \{0,1\}$ and suppose a black box to compute$f$. The Deutsch-Jozsa problem is to determine if f is constant [i.e., $f(x)$ = const, $\forall x \epsilon \{0,1\}^n$] or if f is balanced [i.e., $f(x)$=0 for exactly half the possible input strings $x \epsilon \{0,1\}^n$] using as few calls to the black box computation f as is possible, assuming f is guaranteed to be constant or balanced [10,11].

For the case "number Qbits = 3":
No. of States = 8
(000, 001, 010, 011, 100, 101, 110, 111)

1. **Constants:** The number of Constants is 2

f_{000}	f_{001}	f_{010}	f_{011}	f_{100}	f_{101}	f_{110}	f_{111}
0	0	0	0	0	0	0	0
1	1	1	1	1	1	1	1

2. Balanced States: The number Balanced states is 70.

f_{000}	f_{001}	f_{010}	f_{011}	f_{100}	f_{101}	f_{110}	f_{111}
1	1	1	1	0	0	0	0
1	1	0	1	1	0	0	0
0	1	1	1	1	0	0	0
1	1	0	1	0	1	0	0
0	1	1	1	0	1	0	0
1	0	1	0	1	1	0	0
1	0	0	1	1	1	0	0
0	0	1	1	1	1	0	0
1	1	0	1	0	0	1	0
0	1	1	1	0	0	1	0
1	0	1	0	1	0	1	0
1	0	0	1	1	0	1	0
0	0	1	1	1	0	1	0
1	0	1	0	0	1	1	0
1	0	0	1	0	1	1	0
0	0	1	1	0	1	1	0
0	1	0	0	1	1	1	0
0	0	0	1	1	1	1	0
1	1	0	1	0	0	0	1
0	1	1	1	0	0	0	1
1	0	1	0	1	0	0	1
1	0	0	1	1	0	0	1
0	0	1	1	1	0	0	1

f_{000}	f_{001}	f_{010}	f_{011}	f_{100}	f_{101}	f_{110}	f_{111}
1	1	1	0	1	0	0	0
1	0	1	1	1	0	0	0
1	1	1	0	0	1	0	0
1	0	1	1	0	1	0	0
1	1	0	0	1	1	0	0
0	1	1	0	1	1	0	0
0	1	0	1	1	1	0	0
1	1	1	0	0	0	1	0
1	0	1	1	0	0	1	0
1	1	0	0	1	0	1	0
0	1	1	0	1	0	1	0
0	1	0	1	1	0	1	0
1	1	0	0	0	1	1	0
0	1	1	0	0	1	1	0
0	1	0	1	0	1	1	0
1	0	0	0	1	1	1	0
0	0	1	0	1	1	1	0
1	1	1	0	0	0	0	1
1	0	1	1	0	0	0	1
1	1	0	0	1	0	0	1
0	1	1	0	1	0	0	1
0	1	0	1	1	0	0	1
1	1	0	0	0	1	0	1

1	0	1	0	0	1	0	1
1	0	0	1	0	1	0	1
0	0	1	1	0	1	0	1
0	1	0	0	1	1	0	1
0	0	0	1	1	1	0	1
1	0	1	0	0	0	1	1
1	0	0	1	0	0	1	1
0	0	1	1	0	0	1	1
0	1	0	0	1	0	1	1
0	0	0	1	1	0	1	1
0	1	0	0	0	1	1	1
0	0	0	1	0	1	1	1

0	1	1	0	0	1	0	1
0	1	0	1	0	1	0	1
1	0	0	0	1	1	0	1
0	0	1	0	1	1	0	1
1	1	0	0	0	0	1	1
0	1	1	0	0	0	1	1
0	1	0	1	0	0	1	1
1	0	0	0	1	0	1	1
0	0	1	0	1	0	1	1
1	0	0	0	0	1	1	1
0	0	1	0	0	1	1	1
0	0	0	0	1	1	1	1

3. Unbalanced States: The number of Unbalanced States is 184

f_{000}	f_{001}	f_{010}	f_{011}	f_{100}	f_{101}	f_{110}	f_{111}
1	0	0	0	0	0	0	0
1	1	0	0	0	0	0	0
1	0	1	0	0	0	0	0
1	1	1	0	0	0	0	0
1	0	0	1	0	0	0	0
1	1	0	1	0	0	0	0
1	0	1	1	0	0	0	0
0	0	0	0	1	0	0	0
0	1	0	0	1	0	0	0
0	0	1	0	1	0	0	0
0	1	1	0	1	0	0	0
1	0	0	1	1	0	0	0
0	0	1	1	1	0	0	0

f_{000}	f_{001}	f_{010}	f_{011}	f_{100}	f_{101}	f_{110}	f_{111}
0	1	0	0	0	0	0	0
0	0	1	0	0	0	0	0
0	1	1	0	0	0	0	0
0	0	0	1	0	0	0	0
0	1	0	1	0	0	0	0
0	0	1	1	0	0	0	0
0	1	1	1	0	0	0	0
1	0	0	0	1	0	0	0
1	1	0	0	1	0	0	0
1	0	1	0	1	0	0	0
0	0	0	1	1	0	0	0
0	1	0	1	1	0	0	0
1	1	1	1	1	0	0	0

0	0	0	0	0	1	0	0
0	1	0	0	0	1	0	0
0	0	1	0	0	1	0	0
0	1	1	0	0	1	0	0
1	0	0	1	0	1	0	0
0	0	1	1	0	1	0	0
0	0	0	0	1	1	0	0
0	1	0	0	1	1	0	0
1	1	1	0	1	1	0	0
1	1	0	1	1	1	0	0
0	1	1	1	1	1	0	0
0	0	0	0	0	0	1	0
0	1	0	0	0	0	1	0
0	0	1	0	0	0	1	0
0	1	1	0	0	0	1	0
1	0	0	1	0	0	1	0
0	0	1	1	0	0	1	0
0	0	0	0	1	0	1	0
0	1	0	0	1	0	1	0
1	1	1	0	1	0	1	0
1	1	0	1	1	0	1	0
0	1	1	1	1	0	1	0
0	0	0	0	0	1	1	0
0	1	0	0	0	1	1	0
1	1	1	0	0	1	1	0
1	1	0	1	0	1	1	0
0	1	1	1	0	1	1	0

1	0	0	0	0	1	0	0
1	1	0	0	0	1	0	0
1	0	1	0	0	1	0	0
0	0	0	1	0	1	0	0
0	1	0	1	0	1	0	0
1	1	1	1	0	1	0	0
1	0	0	0	1	1	0	0
0	0	1	0	1	1	0	0
0	0	0	1	1	1	0	0
1	0	1	1	1	1	0	0
1	1	1	1	1	1	0	0
1	0	0	0	0	0	1	0
1	1	0	0	0	0	1	0
1	0	1	0	0	0	1	0
0	0	0	1	0	0	1	0
0	1	0	1	0	0	1	0
1	1	1	1	0	0	1	0
1	0	0	0	1	0	1	0
0	0	1	0	1	0	1	0
0	0	0	1	1	0	1	0
1	0	1	1	1	0	1	0
1	1	1	1	1	0	1	0
1	0	0	0	0	1	1	0
0	0	1	0	0	1	1	0
0	0	0	1	0	1	1	0
1	0	1	1	0	1	1	0
1	1	1	1	0	1	1	0

0	0	0	0	1	1	1	0
1	0	1	0	1	1	1	0
1	1	1	0	1	1	1	0
0	1	0	1	1	1	1	0
0	0	1	1	1	1	1	0
0	1	1	1	1	1	1	0
0	0	0	0	0	0	0	1
0	1	0	0	0	0	0	1
0	0	1	0	0	0	0	1
0	1	1	0	0	0	0	1
1	0	0	1	0	0	0	1
0	0	1	1	0	0	0	1
0	0	0	0	1	0	0	1
0	1	0	0	1	0	0	1
1	1	1	0	1	0	0	1
1	1	0	1	1	0	0	1
0	1	1	1	1	0	0	1
0	0	0	0	0	1	0	1
0	1	0	0	0	1	0	1
1	1	1	0	0	1	0	1
1	1	0	1	0	1	0	1
0	1	1	1	0	1	0	1
0	0	0	0	1	1	0	1
1	0	1	0	1	1	0	1
1	1	1	0	1	1	0	1
0	1	0	1	1	1	0	1
0	0	1	1	1	1	0	1

1	1	0	0	1	1	1	0
0	1	1	0	1	1	1	0
1	0	0	1	1	1	1	0
1	1	0	1	1	1	1	0
1	0	1	1	1	1	1	0
1	1	1	1	1	1	1	0
1	0	0	0	0	0	0	1
1	1	0	0	0	0	0	1
1	0	1	0	0	0	0	1
0	0	0	1	0	0	0	1
0	1	0	1	0	0	0	1
1	1	1	1	0	0	0	1
1	0	0	0	1	0	0	1
0	0	1	0	1	0	0	1
0	0	0	1	1	0	0	1
1	0	1	1	1	0	0	1
1	1	1	1	1	0	0	1
1	0	0	0	0	1	0	1
0	0	1	0	0	1	0	1
0	0	0	1	0	1	0	1
1	0	1	1	0	1	0	1
1	1	1	1	0	1	0	1
1	1	0	0	1	1	0	1
0	1	1	0	1	1	0	1
1	0	0	1	1	1	0	1
1	1	0	1	1	1	0	1
1	0	1	1	1	1	0	1

0	1	1	1	1	1	0	1
0	0	0	0	0	0	1	1
0	1	0	0	0	0	1	1
1	1	1	0	0	0	1	1
1	1	0	1	0	0	1	1
0	1	1	1	0	0	1	1
0	0	0	0	1	0	1	1
1	0	1	0	1	0	1	1
1	1	1	0	1	0	1	1
0	1	0	1	1	0	1	1
0	0	1	1	1	0	1	1
0	1	1	1	1	0	1	1
0	0	0	0	0	1	1	1
1	0	1	0	0	1	1	1
1	1	1	0	0	1	1	1
0	1	0	1	0	1	1	1
0	0	1	1	0	1	1	1
0	1	1	1	0	1	1	1
1	0	0	0	1	1	1	1
1	1	0	0	1	1	1	1
1	0	1	0	1	1	1	1
1	1	1	0	1	1	1	1
1	0	0	1	1	1	1	1
1	1	0	1	1	1	1	1
1	0	1	1	1	1	1	1

1	1	1	1	1	1	0	1
1	0	0	0	0	0	1	1
0	0	1	0	0	0	1	1
0	0	0	1	0	0	1	1
1	0	1	1	0	0	1	1
1	1	1	1	0	0	1	1
1	1	0	0	1	0	1	1
0	1	1	0	1	0	1	1
1	0	0	1	1	0	1	1
1	1	0	1	1	0	1	1
1	0	1	1	1	0	1	1
1	1	1	1	1	0	1	1
1	1	0	0	0	1	1	1
0	1	1	0	0	1	1	1
1	0	0	1	0	1	1	1
1	1	0	1	0	1	1	1
1	0	1	1	0	1	1	1
1	1	1	1	0	1	1	1
0	1	0	0	1	1	1	1
0	0	1	0	1	1	1	1
0	1	1	0	1	1	1	1
0	0	0	1	1	1	1	1
0	1	0	1	1	1	1	1
0	0	1	1	1	1	1	1
0	1	1	1	1	1	1	1

Before we analyze the Deustch-Jozsa algorithm, it will be helpful to think more about Hadamard transforms. For $x \in \{0,1\}$, we have

$$H|x> = \frac{1}{\sqrt{2}}|0> + \frac{1}{\sqrt{2}}(-1)^x|1> \tag{5}$$

In general, the above equation could be expressed as follows:

$$H|x> = \frac{1}{\sqrt{2}}\sum_{y\in\{0,1\}}(-1)^{xy}\,|y> \tag{6}$$

For a system contains three qubits (three fermions), the Boolean function: $\{0,1\}^3 \rightarrow \{0,1\}$, and the states will be:

$$|y> = |y_1y_2y_3> \in \{0,1\}^3 \rightarrow |000>, |001>, |010>, |011>, |100>, |101>, |110>, |111> \tag{7}$$

For three qubits |000> the state can be denoted as

$$|x> = x_1x_2x_3 = |000> \in \{0,1\}^3 \tag{8}$$

So that equation (4) can be re-written as the following mathematical relationship:

$$H^{\otimes 3}|0>^{\otimes 3} H|1 \geq H^{\otimes 3}|0>| \rightarrow [\frac{1}{\sqrt{2^3}}\sum_{y\in\{0,1\}^3}(-1)^{x.y}]|-> \tag{9}$$

Where,

$$x.y = \sum_{i=1}^{3} x_iy_i\,(mod\ 2) \tag{10}$$

According to above equation, the term *x.y* is a modulo-2 expression. Also, the above equations can be used to the realization of equation (4) from (9). Substituting equations (7) & (8) in (9) leads to:

$$|\psi_1> = H^{\otimes 3}|0> H|1> = H^{\otimes 3}|0>|-> = \frac{1}{\sqrt{2^3}}[(-1)^{0.0+0.0+0.0}|000> + (-1)^{0.0+0.0+0.1}|001> + (-1)^{0.0+0.1+0.0}|010> + (-1)^{0.0+0.1+0.1}|011> + (-1)^{0.1+0.0+0.0}|100> + (-1)^{0.1+0.0+0.1}|101> + (-1)^{0.1+0.1+0.0}|110> + (-1)^{0.1+0.1+0.1}|111>]|->$$

$$|\psi_1> = \frac{1}{\sqrt{2^3}}[(-1)^0|000> + (-1)^0|001> + (-1)^0|010> + (-1)^0|011> + (-1)^0|100> + (-1)^0|101> + (-1)^0|110> + (-1)^0|111>]|->$$

$$|\psi_1\rangle = \frac{1}{\sqrt{2^3}}[|000\rangle + |001\rangle + |010\rangle + |011\rangle + |100\rangle + |101\rangle + |110\rangle + |111\rangle]|-\rangle \qquad (11)$$

Equation (11) is similar to equation (4).

The state after the first layer of performing Hadamard transform is:

$$|\psi_1\rangle = \frac{1}{\sqrt{2^3}} \sum_{x\in\{0,1\}^3} |x\rangle \left(\frac{1}{\sqrt{2}}\Big|0\rangle - \frac{1}{\sqrt{2}}\Big|1\rangle\right) \qquad (12)$$

After getting the state ($|\psi_1\rangle$), then transformation U_f is applied on $|\psi_1\rangle$:

$$\begin{aligned}
|\psi_2\rangle &= U_f|\psi_1\rangle \\
|\psi_2\rangle &= U_f \tfrac{1}{\sqrt{2^3}}[|000\rangle + |001\rangle + |010\rangle + |011\rangle + |100\rangle + \\
&\quad |101\rangle\, |110\rangle + |111\rangle](\tfrac{1}{\sqrt{2}}|0\rangle - \tfrac{1}{\sqrt{2}}|1\rangle) \\
|\psi_2\rangle &= \tfrac{1}{\sqrt{2^4}}\{|000\rangle[|f(000)\oplus 0\rangle - |f(000)\oplus 1\rangle] + \\
&\quad |001\rangle[|f(001)\oplus 0\rangle - |f(001)\oplus 1\rangle] + \\
&\quad |010\rangle[|f(010)\oplus 0\rangle - |f(010)\oplus 1\rangle] + \\
&\quad |011\rangle[|f(011)\oplus 0\rangle - |f(011)\oplus 1\rangle] + \\
&\quad |100\rangle[|f(100)\oplus 0\rangle - |f(100)\oplus 1\rangle] + \\
&\quad |101\rangle[|f(101)\oplus 0\rangle - |f(101)\oplus 1\rangle] + \\
&\quad |110\rangle[|f(110)\oplus 0\rangle - |f(110)\oplus 1\rangle] + \\
&\quad |111\rangle[|f(111)\oplus 0\rangle - |f(111)\oplus 1\rangle]\}
\end{aligned} \qquad (13)$$

There are three cases

1. If $f(000)= f(001)= f(010)= f(011)= f(100)= f(101)= f(110)= f(111)=0$.
2. If $f(000)= f(001)= f(010)= f(011)= f(100)= f(101)= f(110)=f(111)=1$.
3. If half of $f(000)$, $f(001)$, $f(010)$, $f(011)$, $f(100)$, $f(101)$, $f(110)$, $f(111)$, equal to 0 and another half equal to1.

Apply case (1) on equation (13):

$$\begin{aligned}
|\psi_2\rangle &= \frac{1}{\sqrt{2^4}}\{|000\rangle[|0\oplus 0\rangle - |0\oplus 1\rangle] + |001\rangle[|0\oplus 0\rangle - \\
&\quad |0\oplus 1\rangle] + |010\rangle[|0\oplus 0\rangle - |0\oplus 1\rangle] + |011\rangle[|0\oplus 0\rangle - \\
&\quad |0\oplus 1\rangle] + |100\rangle[|0\oplus 0\rangle - |0\oplus 1\rangle] + |101\rangle[|0\oplus 0\rangle - \\
&\quad |0\oplus 1\rangle] + |110\rangle[|0\oplus 0\rangle - |0\oplus 1\rangle] + |111\rangle[|0\oplus 0\rangle - \\
&\quad |0\oplus 1\rangle]\} \\
|\psi_2\rangle &= \frac{1}{\sqrt{2^4}}\{|000\rangle[|0\rangle - |1\rangle] + |001\rangle[|0\rangle - |1\rangle] + \\
&\quad |010\rangle[|0\rangle - |1\rangle] + |011\rangle[|0\rangle - |1\rangle] + \\
&\quad |100\rangle[|0\rangle - |1\rangle] + |101\rangle[|0\rangle - |1\rangle] +
\end{aligned}$$

$$|110>[|0>-|1>]+|111>[|0>-|1>]\}$$

$$|\psi_2> = \frac{1}{\sqrt{2^3}}[|000>+|001>+|010>+|011>+|100>+|101>+|110>+|111>]\left(\frac{1}{\sqrt{2}}\Big|0>-\frac{1}{\sqrt{2}}\Big|1>\right) \qquad (14)$$

For case (2): all functions are 1the wave function ψ_2 in this state becomes

$$|\psi_2> = \frac{1}{\sqrt{2^4}}\{|000>[|1\oplus0>-|1\oplus1>]+|001>[|1\oplus0>-|1\oplus1>]+|010>[|1\oplus0>-|1\oplus1>]+|011>[|1\oplus0>-|1\oplus1>]+|100>[|1\oplus0>-|1\oplus1>]+|101>[|1\oplus0>-|1\oplus1>]+|110>[|1\oplus0>-|1\oplus1>]+|111>[|1\oplus0>-|1\oplus1>]\}$$

$$|\psi_2> = \frac{1}{\sqrt{2^4}}\{|000>[|1>-|0>]+|001>[|1>-|0>]+|010>[|1>-|0>]+|011>[|1>-|0>]+|100>[|1>-|0>]+|101>[|1>-|0>]+|110>[|1>-|0>]+|111>[|1>-|0>]\}$$

$$|\psi_2 = \frac{1}{\sqrt{2^3}}\{-|000>-|001>-|010>-|011>-|100>-|101>-|110>-|111>\}\left(\frac{1}{\sqrt{2}}\Big|0>-\frac{1}{\sqrt{2}}\Big|1>\right) \qquad (15)$$

For case (3): half of the functions are 0 and another half are 1; let *f(000)= f(001)= f(010)= f(011)= 0* and *f(100)= f(101)= f(110)= f(111)=* 1so that the $\psi_2$becomes:

$$|\psi_2> = \frac{1}{\sqrt{2^4}}\{|000>[|0\oplus0>-|0\oplus1>]+|001>[|0\oplus0>-|0\oplus1>]+|010>[|0\oplus0>-|0\oplus1>]+|011>[|0\oplus0>-|0\oplus1>]+|100>[|1\oplus0>-|1\oplus1>]+|101>[|1\oplus0>-|1\oplus1>]+|110>[|1\oplus0>-|1\oplus1>]+|111>[|1\oplus0>-|1\oplus1>]\}$$

$$|\psi_2> = \frac{1}{\sqrt{2^4}}\{|000>[|0>-|1>]+|001>[|0>-|1>]+|010>[|0>-|1>]+|011>[|0>-|1>]+|100>[|1>-|0>]+|101>[|1>-|0>]+|110>[|1>-|0>]+|111>[|1>-|0>]\}$$

$$|\psi_2> = \frac{1}{\sqrt{2^3}}\{|000>+|001>+|010>+|011>-|100>-|101>-|110>-|111>\}\left(\frac{1}{\sqrt{2}}\Big|0>-\frac{1}{\sqrt{2}}\Big|1>\right) \qquad (3.16)$$

Any four functions of the eight functions have value 0 other four take 1; we will get the same result as those in equation (16).

In general, equations (14), (15) and (16) can be rewritten as following:

$$|\psi_2> = \frac{1}{\sqrt{2^3}} \sum_{x\in\{0,1\}^3} (-1)^{f(x)}\,|x>\left(\frac{1}{\sqrt{2}}\Big|0> - \frac{1}{\sqrt{2}}\Big|1>\right) \tag{17}$$

Now, the last qubit is discarded and the 3 Hadamard transforms are applied to get:

$$|\psi_3> = \frac{1}{\sqrt{2^3}} \sum_{x\in\{0,1\}^3} (-1)^{f(x)} \left(\frac{1}{\sqrt{2^3}} \sum_{y\in\{0,1\}^3} (-1)^{x.y}\,|y>\right) \tag{18}$$

Equation (3.18) can be re-written as follows:

$$|\psi_3> = \sum_{y\in\{0,1\}^3} \left(\frac{1}{2^3} \sum_{x\in\{0,1\}^3} (-1)^{f(x)+x.y}\right)|y> \tag{19}$$

Sothe equation turned out to be composed of a number of states and each has amplitude (Amp) which is determined as follows:

$$Amp = \frac{1}{2^3} \sum_{x\in\{0,1\}^3} (-1)^{f(x)} \tag{20}$$

Thus, the probability that the measurements all give outcome 0 is:

$$\left|\frac{1}{2^3} \sum_{x\in\{0,1\}^3} (-1)^{f(x)}\right|^2 = \begin{cases} 1 & if\ f\ is\ constant \\ 0 & if\ f\ is\ balanced \end{cases} \tag{21}$$

The quantum information processing system can solve Deutsch's problem with one evaluation of the function *f* compared to the classical requirement for $(2^n/2+1)$ evaluations.

Table 1. Number of queries for classical and quantum state with Number of Bits and Qubits.

No. of Bits	No. of queries in Classical	No. of Qubits	No. of queries in Quantum
1	2	1	1
2	3	2	1
3	5	3	1
4	9	4	1
5	17	5	1
6	33	6	1

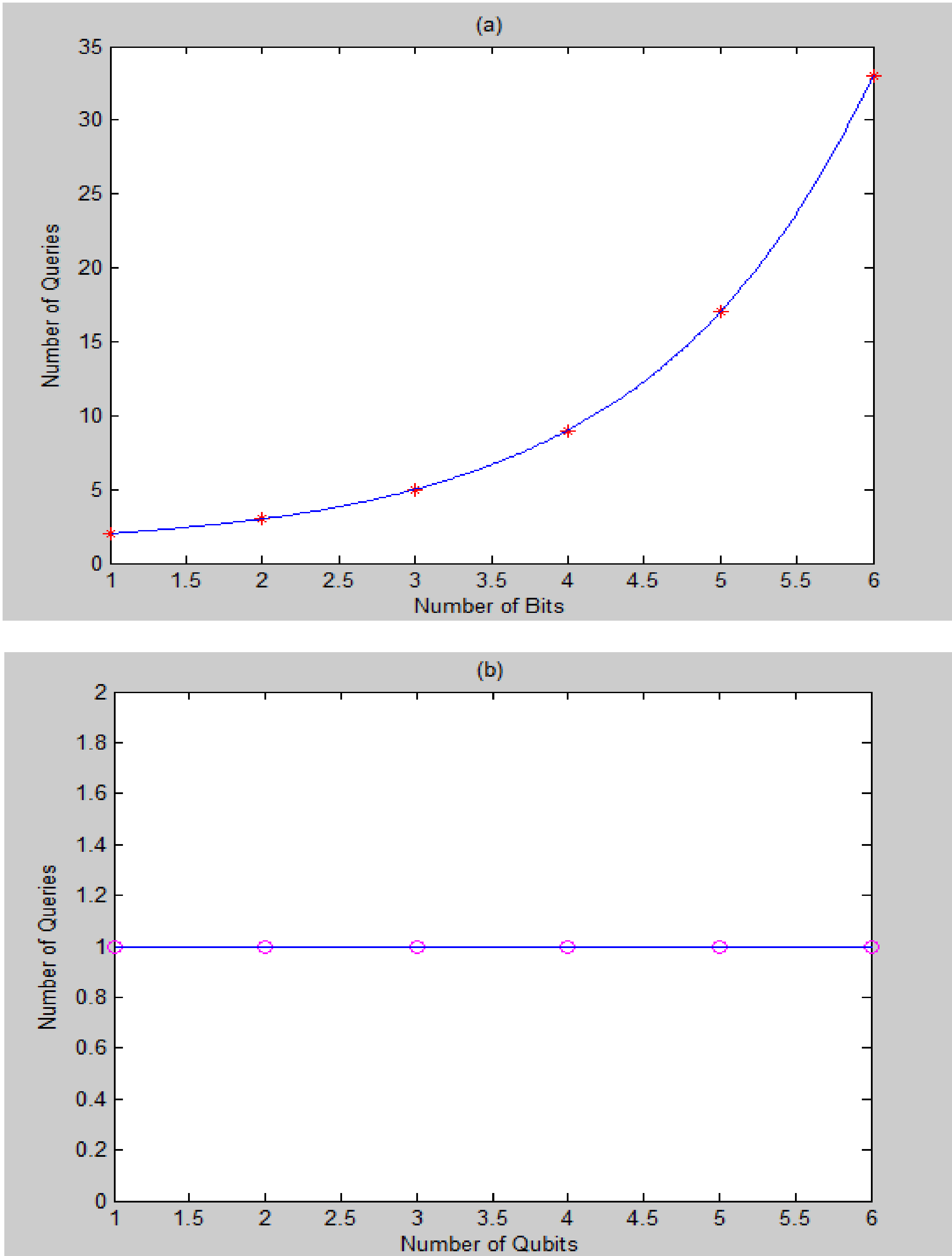

Figure 2. Relationship between number of bits and number of queries: (a) Classical approach, (b) Quantum Approach.

It can be noticed that the increase in the number of query in the classical case be increased exponentially, while in Quantum mechanics the number of queries remains same (= 1) with the increase of number of qubits as shown in the Figure (2).

IV. CONCLUSION

Through the study of Deustch-Jozsa algorithm with three fermions we have that a quantum computer (Q.C.) can solve Deutsch's problem with one estimate of the function f compared to the classical requirement for $2^3/2 + 1 = 5$ estimates. In general for classical computer the relation between number of bits and number of queries are exponentially but for quantum computer the number of queries always one with different number of qubits. This shown the power of quantum computer

References

[1] Alastair, A. A.; The Deutsch-Jozsa Problem: De-quantization and Entanglement. Centre for Discrete Mathematics and Theoretical Computer Science University of Auckland, New Zealand, CDMTCS-371, 2009.

[2] Gulde, S., Riebe, M., Lancaster, G.P., Becher, C., Eschner, J., Haffner, H., Kaler, F.S., Chuang, I.L., and Blatt, R.; Implementation of the Deutsch-Jozsa Algorithm on an Ion-Trap Quantum Computer. *Nature Publishing Group*, Vol. 421, No. 2, Pp. 48-50, 2003.

[3] Batty, M.; Casaccino, A.; Duncan, A.J.; Rees, S. and Severini, S. An Application of the Deutsch-Jozsa Algorithm to Formal Languages and the Word Problem in Groups. Springer-Verlag Berlin Heidelberg, Pp. 57-59, 2008.

[4] Zhang, P., Jiang, Y., Liu, R.F., Gao, H., Li, H.R., and Lia, F.; Implementing the Deutsch's algorithm with Spin-Orbital Angular Momentum of Photon without Interferometer. *Optics Communications*, 285, Pp. 838-841, 2012.

[5] Kiktenko, E.O., Fedorov, A.K. Strakhovd A.A.; and Man'ko, V.I; Single Qudit Realization of the Deutsch Algorithm using Superconducting Many-Level Quantum Circuits. *Physics Letters A*, Vol. 379, No. 22-23, Pp. 1409-1413, 2015.

[6] Marinescu, D. C. and Marinescu, G. M.; Classical and Quantum, Information. Elsevier, 2012.

[7] Nielsen, M.A. and Chuang, I.L.; Quantum Computation and Quantum Information. Cambridge University Press, Cambridge, England, 2010.

[8] Mermin, N.D.; Quantum Computer Science: An Introduction. United States of America by Cambridge University, 2007.

[9] Cleve R., Ekert A., Macchiavello C. and Mosca M.; Quantum Algorithms Revisited. *Proceedings of the Royal Society of London* Vol. 454, Pp. 339-354, 1998.

[10] Alina, V. Quantum Algorithms for Computing the Boolean Function and Verifying Repetition Code. *Computer Science and Information Technologies*, Vol. 756, Pp. 227-247, 2010.

[11] Nisan M. S. On the Degree of Boolean Functions as Real Polynomials. *Computational Complexity*, Vol. 4, No. 4, Pp. 301-313, 1994.

(Received 16 October 2015; accepted 02 November 2015)