Some way to make memory in quantum computing

Koji Nagata
Department of Physics,
Korea Advanced Institute of Science and Technology,
Daejeon, Korea
E-mail: ko_mi_na@yahoo.co.jp
Phone: +81-90-1933-4796

Tadao Nakamura
Department of Information and Computer Science, Keio University,
3-14-1 Hiyoshi, Kohoku-ku, Yokohama 223-8522, Japan
E-mail: nakamura@pipelining.jp
Phone: +81-90-5849-5848

June 26, 2023

Keywords: Quantum computing; Quantum algorithm; Formalism

Abstract

A novel definition for making memory in quantum computing based on the Bloch sphere is proposed. (1) We discuss memory of functions. We store the property of a function f itself by means of the unitary operation U_f. Therefore, we write function data on the Bloch sphere. In addition, we recall the property of the function f by means of the same unitary operation U_f as the above. That is, we read data from the Bloch sphere. We discuss that we can read/write small amount of data on the Bloch sphere when we consider the function as data. (2) We discuss memory of values. We store an arbitrary natural number relating to a phase ϕ (real number) itself by means of the unitary operation U_f. Therefore, we write value data on the Bloch sphere. Also in addition, we recall the natural number relating to the phase ϕ by means of the same unitary operation U_f as the above. That is, we read data from the Bloch sphere. We discuss that we can read/write infinite data on the Bloch sphere when we consider an arbitrary natural number relating to the phase factor of a quantum state as data. It may be much likely that, from the property of quantum physics, the Bloch sphere is a high-speed memory system in quantum computing at least in an algorithm level.

1 Introduction

Quantum mechanics (cf. [1, 2, 3, 4, 5, 6, 7]) is a physical theory in order to explain the microscopic behaviors of the nature. Articles of research for constructing theoretical quantum algorithms [8] may be mentioned as follows: In 1985, the Deutsch algorithm was introduced and constructed for the function property problem [9, 10, 11]. Recently, Deutsch's algorithm is generalized [12] by determining arbitrary one-variable boolean functions $f : \{0, 1\} \to \{0, 1\}$. In 1993, the Bernstein–Vazirani algorithm was proposed for identifying linear functions [13, 14]. More recently, the Bernstein–Vazirani algorithm generalized is investigated [15, 16]. And solving Bernstein and Vazirani's problem with the 2-bit permutation function is discussed by Chen, Chang, and Hsueh [17]. In 1994, Simon's algorithm [18] and Shor's algorithm [19] were discussed for period finding of periodic functions. In 1996, Grover [20] provided an algorithm for unordered object finding and the motivation for exploring the computational possibilities offered by quantum mechanics.

In 2020, a parallel computation for all of the combinations of values in variables of a logical function was proposed by Nagata and Nakamura [21]. In 2023, Ossorio-Castillo, Pastor-Díaz, and Tornero expressed in the introduction in Ref. [22] the result by Nagata and Nakamura [21] such that Deutsch's problem is generalized by determining arbitrary two-variable boolean functions $f : \{0, 1\}^2 \to \{0, 1\}$. By virtue of the result proposed in Ref. [21], in 2021, the concrete quantum circuits for addition of any two numbers were proposed by Nakamura and Nagata [23]. In preparing all the augmentations for addition, we can expand any operation in mathematics. Therefore, the quantum computer can solve all the four basic operations of arithmetic, addition, subtraction, multiplication, and division. Further it can be said that this quantum computer naturally operates not only arithmetic but also logic in terms of boolean logic. As a result, the theory presented in [21, 23] can build a true quantum-gated computer that is driven and operated by all software (all programs) used on existing electronic computers. We expect its application for the numerical computation. We lead to implementing commercial quantum-gated computers.

The concept of "Memory" is very important for discussing computer science in general. This situation is the same as studying quantum computing. What is the definition of memory in quantum computers? It is different from the one in classical computers because a quantum computer is different from a classical computer. Memory proposed here is like DRAM (Dynamic Random Access Memory) that is a type of semiconductor memory.

In this paper, a novel definition for making memory in quantum computing based on the Bloch sphere is proposed. Ultimately speaking, all of the quantum algorithms rely on a high-speed memory system on the Bloch sphere. Clearly, the Bloch sphere is a place in which quantum states are stored theoretically steadily. Thus, we can steadily read/write data on the Bloch sphere through a quantum state itself. In short, from the property of quantum physics, the Bloch sphere is much likely to be a high-speed memory system.

Why do we use Deutsch's algorithm in order to define memory in quantum computers? Memory is usually a one-variable type and Deutsch's algorithm is suitable for the purpose for defining memory in quantum computers because Deutsch's algorithm treats one-variable functions. So, we use mainly Deutsch's algorithm in this paper.

2 Review of the basic structure of quantum computing

Superposition and phase kickback are fundamental features of many quantum algorithms [7]. They allow quantum computers to evaluate simultaneously the values of a function $f(x)$ for many different x. Suppose

$$f : \{0,1\} \to \{0,1\} \tag{2.1}$$

is a logical function with a one-bit domain and range. Such a function assuming values in the set $\{T, F\}$ is called a logical function. A convenient way of computing the function on a quantum computer is of considering a two-qubit quantum computer that starts with the state $|x, y\rangle$, where x and y are variables for the function f. The abbreviation $|x, y\rangle$ stands for $|x\rangle \otimes |y\rangle$.

We denote a transformation U_f defined by the map

$$U_f|x, y\rangle = |x, y \oplus f(x)\rangle. \tag{2.2}$$

The transformation U_f is called to be the quantum oracle, where $\oplus$ indicates addition modulo 2.

3 Review of Deutsch's algorithm

Deutsch's algorithm is originally deterministic by one query whether the given function $f : \mathbb{Z}_2 \to \mathbb{Z}_2$ is constant or balanced. The function is called to be constant if $f(0) = f(1)$. The function is called to be balanced if $f(0) \neq f(1)$. Let us review Deutsch's formula as follows:

$$\begin{aligned} &U_f|0\rangle(|0\rangle - |1\rangle)/\sqrt{2} = |0\rangle(|0 \oplus f(0)\rangle - |1 \oplus f(0)\rangle)/\sqrt{2} \\ &= \begin{cases} (-1)^{f(0)}|0\rangle(|0\rangle - |1\rangle)/\sqrt{2} & \text{if} \quad f(0) = 0, \\ (-1)^{f(0)}|0\rangle(|0\rangle - |1\rangle)/\sqrt{2} & \text{if} \quad f(0) = 1. \end{cases} \end{aligned} \tag{3.1}$$

$$\begin{aligned} &U_f|1\rangle(|0\rangle - |1\rangle)/\sqrt{2} = |1\rangle(|0 \oplus f(1)\rangle - |1 \oplus f(1)\rangle)/\sqrt{2} \\ &= \begin{cases} (-1)^{f(1)}|1\rangle(|0\rangle - |1\rangle)/\sqrt{2} & \text{if} \quad f(1) = 0, \\ (-1)^{f(1)}|1\rangle(|0\rangle - |1\rangle)/\sqrt{2} & \text{if} \quad f(1) = 1. \end{cases} \end{aligned} \tag{3.2}$$

This is the phase kickback formation.

Let us introduce the Bloch sphere. We consider a quantum state lying in the x-axis and a quantum state lying in the z-axis. Deutsch's formula does not use a quantum state lying in the y-axis. $f(0)$ and $f(1)$ appear in the global phase factor, but we cannot obtain both of them at the same time.

We define the following notations:

$$|+\rangle_x = \frac{|0\rangle + |1\rangle}{\sqrt{2}}, |-\rangle_x = \frac{|0\rangle - |1\rangle}{\sqrt{2}}. \tag{3.3}$$

We may define the initial state $|\psi_0\rangle_d$ as follows. The subscript "d" means Deutsch's algorithm.

$$|\psi_0\rangle_d = \frac{1}{\sqrt{2}}|0\rangle|-\rangle_x + \frac{1}{\sqrt{2}}|1\rangle|-\rangle_x = |+\rangle_x|-\rangle_x, \quad {}_d\langle\psi_0|\psi_0\rangle_d = 1. \tag{3.4}$$

Let us introduce a parameter $j(= 0, 1, 2, 3)$ that distinguishes the logical function one another. Applying $U_{f_j}(j = 0, 1, 2, 3)$ to $|\psi_0\rangle_d$, $U_{f_j}|\psi_0\rangle_d = |\psi_1\rangle_{jd}$, therefore leaves us with one of four cases:

$$\begin{aligned} &U_{f_0}|\psi_0\rangle_d = |\psi_1\rangle_{0d} = \frac{1}{\sqrt{2}}|0\rangle|-\rangle_x + \frac{1}{\sqrt{2}}|1\rangle|-\rangle_x = |+\rangle_x|-\rangle_x \\ &\text{iff} \quad f_0(0) = 0, f_0(1) = 0. \end{aligned} \tag{3.5}$$

$$\begin{aligned} &U_{f_1}|\psi_0\rangle_d = |\psi_1\rangle_{1d} = \frac{1}{\sqrt{2}}|0\rangle|-\rangle_x - \frac{1}{\sqrt{2}}|1\rangle|-\rangle_x = |-\rangle_x|-\rangle_x \\ &\text{iff} \quad f_1(0) = 0, f_1(1) = 1. \end{aligned} \tag{3.6}$$

$$\begin{aligned} &U_{f_2}|\psi_0\rangle_d = |\psi_1\rangle_{2d} = -\frac{1}{\sqrt{2}}|0\rangle|-\rangle_x + \frac{1}{\sqrt{2}}|1\rangle|-\rangle_x = -|-\rangle_x|-\rangle_x \\ &\text{iff} \quad f_2(0) = 1, f_2(1) = 0. \end{aligned} \tag{3.7}$$

$$\begin{aligned} &U_{f_3}|\psi_0\rangle_d = |\psi_1\rangle_{3d} = -\frac{1}{\sqrt{2}}|0\rangle|-\rangle_x - \frac{1}{\sqrt{2}}|1\rangle|-\rangle_x = -|+\rangle_x|-\rangle_x \\ &\text{iff} \quad f_3(0) = 1, f_3(1) = 1. \end{aligned} \tag{3.8}$$

Even though, we have (3.5)–(3.8), we do not obtain simultaneously both $f(0)$ and $f(1)$ by measuring the resulting state.

By measuring $|\psi_1\rangle_{jd}$, we cannot determine simultaneously all the two values of $f_j(x)$ for all x. But, we can determine if the given function is constant or balanced. This is very interesting indeed: the quantum algorithm gives us the ability to determine a property of $f_j(x)$. This is faster than that of its classical apparatus which would require at least two evaluations.

4 Review of Deutsch's algorithm generalized

The discussion is based on Nagata and Nakamura [12]. We generalize Deutsch's algorithm using a quantum state lying in the xy-plane. Deutsch's algorithm generalized determines simultaneously all the mappings of the given function by one query. In what follows, we consider the Bloch sphere, especially, we consider a pure state lying on the surface of the Bloch sphere. From Deutsch's formula and the mapping U_f, we arrive at the following formulas:

$$U_f|0\rangle(\cos\frac{\theta}{2}|0\rangle + e^{i\phi}\sin\frac{\theta}{2}|1\rangle) = |0\rangle(\cos\frac{\theta}{2}|0\oplus f(0)\rangle + e^{i\phi}\sin\frac{\theta}{2}|1\oplus f(0)\rangle)$$
$$= \begin{cases} |0\rangle(\cos\frac{\theta}{2}|0\rangle + e^{i\phi}\sin\frac{\theta}{2}|1\rangle) & \text{if} \quad f(0)=0, \\ |0\rangle(\cos\frac{\theta}{2}|1\rangle + e^{i\phi}\sin\frac{\theta}{2}|0\rangle) & \text{if} \quad f(0)=1. \end{cases} \tag{4.1}$$

$$U_f|1\rangle(\cos\frac{\theta'}{2}|0\rangle + e^{i\phi'}\sin\frac{\theta'}{2}|1\rangle) = |1\rangle(\cos\frac{\theta'}{2}|0\oplus f(1)\rangle + e^{i\phi'}\sin\frac{\theta'}{2}|1\oplus f(1)\rangle)$$
$$= \begin{cases} |1\rangle(\cos\frac{\theta'}{2}|0\rangle + e^{i\phi'}\sin\frac{\theta'}{2}|1\rangle) & \text{if} \quad f(1)=0, \\ |1\rangle(\cos\frac{\theta'}{2}|1\rangle + e^{i\phi'}\sin\frac{\theta'}{2}|0\rangle) & \text{if} \quad f(1)=1. \end{cases} \tag{4.2}$$

To simplify, we suppose the quantum state lying in just the y-axis. Thus let (θ, ϕ) be $(\pi/2, \pi/2)$ and let (θ', ϕ') be $(\pi/2, \pi/2)$ in giving

$$U_f|0\rangle(|0\rangle + i|1\rangle)/\sqrt{2} = \begin{cases} (i)^{f(0)}|0\rangle(|0\rangle + i|1\rangle)/\sqrt{2} & \text{if} \quad f(0)=0, \\ (i)^{f(0)}|0\rangle(|0\rangle - i|1\rangle)/\sqrt{2} & \text{if} \quad f(0)=1. \end{cases} \tag{4.3}$$

$$U_f|1\rangle(|0\rangle + i|1\rangle)/\sqrt{2} = \begin{cases} (i)^{f(1)}|1\rangle(|0\rangle + i|1\rangle)/\sqrt{2} & \text{if} \quad f(1)=0, \\ (i)^{f(1)}|1\rangle(|0\rangle - i|1\rangle)/\sqrt{2} & \text{if} \quad f(1)=1. \end{cases} \tag{4.4}$$

We define the following notations:

$$|+\rangle_y = \frac{|0\rangle + i|1\rangle}{\sqrt{2}}, |-\rangle_y = \frac{|0\rangle - i|1\rangle}{\sqrt{2}}. \tag{4.5}$$

We define the initial state $|\psi_0\rangle$ as follows, using an imaginary number i. Here, we use a quantum phase effect, which is a quantum phenomenon.

$$|\psi_0\rangle = \frac{1}{\sqrt{2}}|0\rangle|+\rangle_y + \frac{1}{\sqrt{2}}|1\rangle|+\rangle_y = |+\rangle_x|+\rangle_y, \quad \langle\psi_0|\psi_0\rangle = 1. \tag{4.6}$$

Applying $U_{f_j}(j = 0,1,2,3)$ to $|\psi_0\rangle$, $U_{f_j}|\psi_0\rangle = |\psi_1\rangle_j$, therefore leaves us with one of four cases:

$$U_{f_0}|\psi_0\rangle = |\psi_1\rangle_0 = \frac{1}{\sqrt{2}}|0\rangle|+\rangle_y + \frac{1}{\sqrt{2}}|1\rangle|+\rangle_y$$
$$\text{iff} \quad f_0(0) = 0, f_0(1) = 0. \tag{4.7}$$

$$U_{f_1}|\psi_0\rangle = |\psi_1\rangle_1 = \frac{1}{\sqrt{2}}|0\rangle|+\rangle_y + i\frac{1}{\sqrt{2}}|1\rangle|-\rangle_y$$
$$\text{iff} \quad f_1(0) = 0, f_1(1) = 1. \tag{4.8}$$

$$U_{f_2}|\psi_0\rangle = |\psi_1\rangle_2 = i\frac{1}{\sqrt{2}}|0\rangle|-\rangle_y + \frac{1}{\sqrt{2}}|1\rangle|+\rangle_y$$
$$\text{iff} \quad f_2(0) = 1, f_2(1) = 0. \tag{4.9}$$

$$U_{f_3}|\psi_0\rangle = |\psi_1\rangle_3 = i\frac{1}{\sqrt{2}}|0\rangle|-\rangle_y + i\frac{1}{\sqrt{2}}|1\rangle|-\rangle_y$$
$$\text{iff} \quad f_3(0) = 1, f_3(1) = 1. \tag{4.10}$$

If we have (4.7)–(4.10), we have simultaneously both $f(0)$ and $f(1)$ by measuring the resulting state.

By measuring $|\psi_1\rangle_j$, we may determine simultaneously all the two mappings of $f_j(x)$ for all x. This is very interesting indeed: the quantum algorithm gives us the ability to determine a perfect property of $f_j(x)$, namely, $f_j(x)$ itself. This is faster than that of its classical apparatus which would require at least 2^2 evaluations. However, the four states are not completely orthogonal to one another. Therefore, we have some error probability when we distinguish the four states one another [24, 25]. Nevertheless, we are able to eliminate the error probability into zero as shown below.

5 Deutsch's algorithm generalized and based on orthogonal states

We present Deutsch's algorithm generalized and based on orthogonal states. We propose the initial state as follows:

$$|\psi_0\rangle_d \otimes |\psi_0\rangle = |+\rangle_x|-\rangle_x \otimes |+\rangle_x|+\rangle_y. \tag{5.1}$$

Applying $U_{f_j} \otimes U_{f_j} (j = 0, 1, 2, 3)$ to $|\psi_0\rangle_d \otimes |\psi_0\rangle$, $U_{f_j} \otimes U_{f_j}|\psi_0\rangle_d \otimes |\psi_0\rangle = |\psi_1\rangle_{jd} \otimes |\psi_1\rangle_j$, therefore leaves us with one of four cases:

$$U_{f_0} \otimes U_{f_0}|\psi_0\rangle_d \otimes |\psi_0\rangle = |\psi_1\rangle_{0d} \otimes |\psi_1\rangle_0 = |+\rangle_x|-\rangle_x \otimes (\frac{1}{\sqrt{2}}|0\rangle|+\rangle_y + \frac{1}{\sqrt{2}}|1\rangle|+\rangle_y)$$
$$\text{iff} \quad f_0(0) = 0, f_0(1) = 0. \tag{5.2}$$

$$U_{f_1} \otimes U_{f_1}|\psi_0\rangle_d \otimes |\psi_0\rangle = |\psi_1\rangle_{1d} \otimes |\psi_1\rangle_1 = |-\rangle_x|-\rangle_x \otimes (\frac{1}{\sqrt{2}}|0\rangle|+\rangle_y + i\frac{1}{\sqrt{2}}|1\rangle|-\rangle_y)$$
$$\text{iff} \quad f_1(0) = 0, f_1(1) = 1. \tag{5.3}$$

$$U_{f_2} \otimes U_{f_2}|\psi_0\rangle_d \otimes |\psi_0\rangle = |\psi_1\rangle_{2d} \otimes |\psi_1\rangle_2 = -|-\rangle_x|-\rangle_x \otimes (i\frac{1}{\sqrt{2}}|0\rangle|-\rangle_y + \frac{1}{\sqrt{2}}|1\rangle|+\rangle_y)$$
$$\text{iff} \quad f_2(0) = 1, f_2(1) = 0. \tag{5.4}$$

$$U_{f_3} \otimes U_{f_3}|\psi_0\rangle_d \otimes |\psi_0\rangle = |\psi_1\rangle_{3d} \otimes |\psi_1\rangle_3 = -|+\rangle_x|-\rangle_x \otimes (i\frac{1}{\sqrt{2}}|0\rangle|-\rangle_y + i\frac{1}{\sqrt{2}}|1\rangle|-\rangle_y)$$
$$\text{iff} \quad f_3(0) = 1, f_3(1) = 1. \tag{5.5}$$

If we have the relations above (5.2)–(5.5), we have simultaneously both $f(0)$ and $f(1)$ by measuring the resulting state. The four states are completely orthogonal to one another. Therefore, we have zero error probability when we distinguish the four states one another.

By measuring $|\psi_1\rangle_{jd} \otimes |\psi_1\rangle_j$, we may determine simultaneously all the two mappings of $f_j(x)$ for all x. This is very interesting indeed: the quantum algorithm gives us the ability to determine a perfect property of $f_j(x)$, namely, $f_j(x)$ itself. This is faster than that of its classical apparatus which would require at least 2^2 evaluations.

The algorithm is as follows:

1. Select a function f_j and do not know any mappings of it, that is,

$$f_j(0) = ?, f_j(1) = ?. \tag{5.6}$$

2. Operate $U_{f_j} \otimes U_{f_j}$ to $|\psi_0\rangle_d \otimes |\psi_0\rangle$ in giving $|\psi_1\rangle_{jd} \otimes |\psi_1\rangle_j$.

3. Measure the resulting state $|\psi_1\rangle_{jd} \otimes |\psi_1\rangle_j$ and obtain the values of all the mappings concerning the function f_j.

4. This is faster than that of its classical apparatus which would require at least 2^2 evaluations.

6 Definition for making memory in quantum computing

In order to operate quantum computers, the concept of Memory which stores states of qubits, must be defined here. One unit of memory corresponds to one memory location where one state is stored. The address of the unit is specified by its own variable name that is used as the state enclosed by "Cket sign". For example, $|x\rangle$ means some state x, and x is also the name of the variable here. So, the unit is addressed for variable x, but the value of this variable x is $|x\rangle$ using the format "$|\ \ \rangle$" based on a classical computer's programming sense. However, there is something wondering how the x is still like a variable. This is no choice because so long as handling states in quantum computing and this is enough! In case of registers that exist at the "into" and "out of" sides of quantum oracles, the first, the second, etc. qubits are regarded as the variables.

Here, we propose a novel definition for making memory in all of the quantum algorithms based on the Bloch sphere. Ultimately speaking, all of the quantum algorithms rely on a high-speed memory system on the Bloch sphere. Clearly, the Bloch sphere is a place in which quantum states are stored theoretically steadily. Thus, we can steadily read/write data on the Bloch sphere through a quantum state itself. In short, from the property of quantum physics, the Bloch sphere is a high-speed memory system for any quantum algorithm. Figure 1 represents the overview of memory in quantum computers based on the Bloch sphere.

7 Memory of functions in Deutsch's algorithm

Deutsch's algorithm is a process to store resulting states, where resulting states are data concerning the kind (constant or balanced) of a logical function.

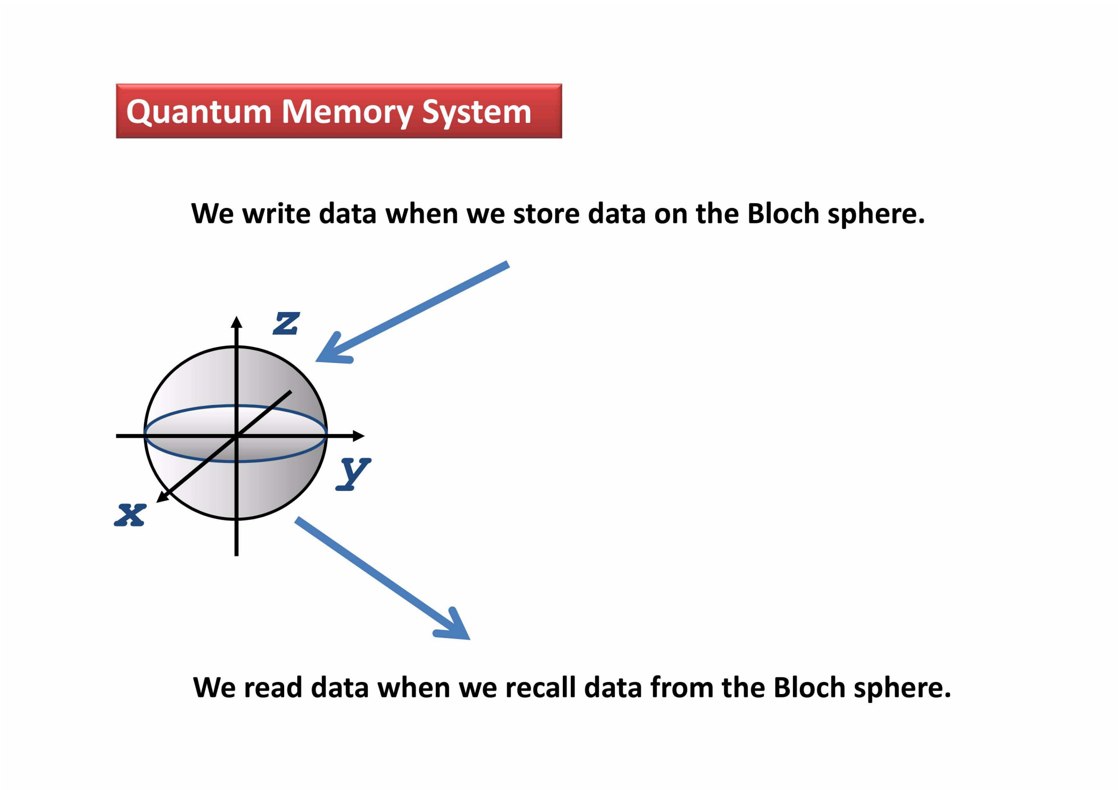

Figure 1: Overview of memory in quantum computers based on the Bloch sphere

- We write data when we store data on the Bloch sphere.
- We read data when we recall data from the Bloch sphere.

We operate some kind of the unitary operation and store the resulting state as data. We write the kind of a logical function by using the unitary operation on the Bloch sphere. That is, we write data on the Bloch sphere in the x-axis. In addition, we operate the same unitary operation as the above and recall the resulting state. We read the kind of the logical function by using the same unitary operation from the Bloch sphere. That is, we read data from the Bloch sphere in the x-axis again. The goal is that we read/write the kind of the logical function which we select first. It may be much likely that, from the property of quantum physics, the Bloch sphere is a high-speed memory system in Deutsch's algorithm.

Suppose that

$$f : \{0, 1\} \to \{0, 1\} \tag{7.1}$$

is a logical function. Our aim is of storing the kind of the logical function, e.g., $f(0) \neq f(1)$. We can select one of the four possible functions because of the combinations of the values. We introduce a parameter $j(= 0, 1, 2, 3)$ for distinguishing among these functions one another.

Let us discuss memory of functions in Deutsch's algorithm. We introduce the transformation U_f defined by the map

$$U_f|x\rangle|k\rangle = |x\rangle|k \oplus f(x)\rangle. \tag{7.2}$$

In fact, from the map U_f, we can define the following formulas:

$$\begin{aligned}
&U_f|0\rangle(|0\rangle - |1\rangle)/\sqrt{2}\\
&= +|0\rangle(|0 \oplus f(0)\rangle - |1 \oplus f(0)\rangle)/\sqrt{2}\\
&= \begin{cases} (-1)^{f(0)}|0\rangle(|0\rangle - |1\rangle)/\sqrt{2} & \text{if } \ f(0) = 0,\\ (-1)^{f(0)}|0\rangle(|0\rangle - |1\rangle)/\sqrt{2} & \text{if } \ f(0) = 1. \end{cases}
\end{aligned} \tag{7.3}$$

$$\begin{aligned}
&U_f|1\rangle(|0\rangle - |1\rangle)/\sqrt{2}\\
&= +|1\rangle(|0 \oplus f(1)\rangle - |1 \oplus f(1)\rangle)/\sqrt{2}\\
&= \begin{cases} (-1)^{f(1)}|1\rangle(|0\rangle - |1\rangle)/\sqrt{2} & \text{if } \ f(1) = 0,\\ (-1)^{f(1)}|1\rangle(|0\rangle - |1\rangle)/\sqrt{2} & \text{if } \ f(1) = 1. \end{cases}
\end{aligned} \tag{7.4}$$

Observe that

$$(U_f)^2|x\rangle|k\rangle = |x\rangle|k \oplus 2f(x)\rangle = |x\rangle|k\rangle. \tag{7.5}$$

Therefore, the map U_f is a unitary operation. This is the key of memory of functions in Deutsch's algorithm. Here, we define the initial state as follows:

$$|\psi_0\rangle_d = \frac{1}{\sqrt{2}}|0\rangle|-\rangle_x + \frac{1}{\sqrt{2}}|1\rangle|-\rangle_x = |+\rangle_x|-\rangle_x, \quad {}_d\langle\psi_0|\psi_0\rangle_d = 1. \tag{7.6}$$

Later, we see that the kind of the logical function is stored into the resulting state. This stage means we write data on the Bloch sphere. At the beginning of memory of functions in Deutsch's algorithm, we apply $U_{f_j}(j = 0, 1, 2, 3)$ to the initial state $|\psi_0\rangle$, $U_{f_j}|\psi_0\rangle = |\psi_1\rangle_{jd}$, then the resulting state is one of four cases:

$$\begin{aligned}
&U_{f_0}|\psi_0\rangle_d = |\psi_1\rangle_{0d} = \frac{1}{\sqrt{2}}|0\rangle|-\rangle_x + \frac{1}{\sqrt{2}}|1\rangle|-\rangle_x = |+\rangle_x|-\rangle_x\\
&\text{iff} \ \ f_0(0) = 0, f_0(1) = 0.
\end{aligned} \tag{7.7}$$

$$\begin{aligned}
&U_{f_1}|\psi_0\rangle_d = |\psi_1\rangle_{1d} = \frac{1}{\sqrt{2}}|0\rangle|-\rangle_x - \frac{1}{\sqrt{2}}|1\rangle|-\rangle_x = |-\rangle_x|-\rangle_x\\
&\text{iff} \ \ f_1(0) = 0, f_1(1) = 1.
\end{aligned} \tag{7.8}$$

$$\begin{aligned}
&U_{f_2}|\psi_0\rangle_d = |\psi_1\rangle_{2d} = -\frac{1}{\sqrt{2}}|0\rangle|-\rangle_x + \frac{1}{\sqrt{2}}|1\rangle|-\rangle_x = -|-\rangle_x|-\rangle_x\\
&\text{iff} \ \ f_2(0) = 1, f_2(1) = 0.
\end{aligned} \tag{7.9}$$

$$\begin{aligned}
&U_{f_3}|\psi_0\rangle_d = |\psi_1\rangle_{3d} = -\frac{1}{\sqrt{2}}|0\rangle|-\rangle_x - \frac{1}{\sqrt{2}}|1\rangle|-\rangle_x = -|+\rangle_x|-\rangle_x\\
&\text{iff} \ \ f_3(0) = 1, f_3(1) = 1.
\end{aligned} \tag{7.10}$$

We apply the same unitary operation U_{f_j} as the above to the resulting state $|\psi_1\rangle_{jd}$ and recall the kind of the function f_j by going back to the initial state $|\psi_0\rangle$ from the resulting state $|\psi_1\rangle_{jd}$.

By recalling the resulting state $|\psi_1\rangle_{jd}$ from the Bloch sphere, we can read the kind of the logical function depending on the parameter j. Interestingly, memory of functions in Deutsch's algorithm gives us the ability to recall a property of $f_j(x)$, namely, $f_j(x)$ is constant or balanced. This stage means we read data from the Bloch sphere.

With the above, memory of functions in Deutsch's algorithm is as follows:

1. Select the kind of a function f_j which is constant $(f_j(0) = f_j(1))$ or balanced $(f_j(0) \neq f_j(1))$.
2. Select the unitary operator U_{f_j} along with the selected function f_j.
3. Apply U_{f_j} to the initial state $|\psi_0\rangle$ and store the kind of the function into the resulting state $|\psi_1\rangle_{jd}$.
4. This stage means we write data on the Bloch sphere.
5. Select the same unitary operator U_{f_j} as the above.
6. Apply U_{f_j} to the resulting state $|\psi_1\rangle_{jd}$ and recall the kind of the function f_j by going back to the initial state $|\psi_0\rangle$ from the resulting state $|\psi_1\rangle_{jd}$.
7. This stage means we read data from the Bloch sphere.

This is faster than that of its classical apparatus which would require at least two evaluations. It may be much likely that, from the property of quantum physics, the Bloch sphere is a high-speed memory system for Deutsch's algorithm.

8 Memory of functions in Deutsch's algorithm generalized

Deutsch's algorithm generalized is a process to store resulting states, where resulting states are data concerning all of the combinations of values in variables of a logical function. We operate one of four kinds of the unitary operations and store the resulting state. We write one of four kinds of a logical function by using the unitary operation on the Bloch sphere. That is, we write data on the Bloch sphere in the xy-plane. In addition, we operate the same unitary operation as the above and recall the resulting state. We read the same kind of the logical function by using the same unitary operation on the Bloch sphere. That is, we read data from the Bloch sphere in the xy-plane again. The goal is that we read/write one of four kinds of the logical function which we select first. It may be much likely that, from the property of quantum physics, the Bloch sphere is a high-speed memory system in Deutsch's algorithm generalized.

Suppose that

$$f : \{0, 1\} \rightarrow \{0, 1\} \tag{8.1}$$

is a logical function. Our aim is of storing both the two values of the logical function, e.g.,

$$f(0) = 0, f(1) = 1, \tag{8.2}$$

that is, $f(x)$ itself.

Let us discuss memory of functions in Deutsch's algorithm generalized. We introduce the transformation U_f defined by the map

$$U_f|x\rangle|k\rangle = |x\rangle|k \oplus f(x)\rangle. \tag{8.3}$$

In fact, from the map U_f, we can define the following formulas:

$$U_f|0\rangle(|0\rangle + i|1\rangle)/\sqrt{2} = \begin{cases} (i)^{f(0)}|0\rangle(|0\rangle + i|1\rangle)/\sqrt{2} & \text{if } f(0) = 0, \\ (i)^{f(0)}|0\rangle(|0\rangle - i|1\rangle)/\sqrt{2} & \text{if } f(0) = 1. \end{cases} \tag{8.4}$$

$$U_f|1\rangle(|0\rangle + i|1\rangle)/\sqrt{2} = \begin{cases} (i)^{f(1)}|1\rangle(|0\rangle + i|1\rangle)/\sqrt{2} & \text{if } f(1) = 0, \\ (i)^{f(1)}|1\rangle(|0\rangle - i|1\rangle)/\sqrt{2} & \text{if } f(1) = 1. \end{cases} \tag{8.5}$$

Observe that

$$(U_f)^2|x\rangle|k\rangle = |x\rangle|k \oplus 2f(x)\rangle = |x\rangle|k\rangle. \tag{8.6}$$

Therefore, the map U_f is a unitary operation. This is the key of memory of functions in Deutsch's algorithm generalized. Here, we define the initial state as follows:

$$|\psi_0\rangle = \frac{1}{\sqrt{2}}|0\rangle|+\rangle_y + \frac{1}{\sqrt{2}}|1\rangle|+\rangle_y = |+\rangle_x|+\rangle_y, \quad \langle\psi_0|\psi_0\rangle = 1. \tag{8.7}$$

Later, we see that both the values for f_j is stored into the resulting state. This stage means we write data on the Bloch sphere. At the beginning of memory of functions in Deutsch's algorithm generalized, we apply $U_{f_j}(j = 0, 1, 2, 3)$ to the initial state $|\psi_0\rangle$, $U_{f_j}|\psi_0\rangle = |\psi_1\rangle_j$, then the resulting state is one of four cases:

$$\begin{aligned} &U_{f_0}|\psi_0\rangle = |\psi_1\rangle_0 = \frac{1}{\sqrt{2}}|0\rangle|+\rangle_y + \frac{1}{\sqrt{2}}|1\rangle|+\rangle_y \\ &\text{iff } \ f_0(0) = 0, f_0(1) = 0. \end{aligned} \tag{8.8}$$

$$\begin{aligned} &U_{f_1}|\psi_0\rangle = |\psi_1\rangle_1 = \frac{1}{\sqrt{2}}|0\rangle|+\rangle_y + i\frac{1}{\sqrt{2}}|1\rangle|-\rangle_y \\ &\text{iff } \ f_1(0) = 0, f_1(1) = 1. \end{aligned} \tag{8.9}$$

$$\begin{aligned} &U_{f_2}|\psi_0\rangle = |\psi_1\rangle_2 = i\frac{1}{\sqrt{2}}|0\rangle|-\rangle_y + \frac{1}{\sqrt{2}}|1\rangle|+\rangle_y \\ &\text{iff } \ f_2(0) = 1, f_2(1) = 0. \end{aligned} \tag{8.10}$$

$$U_{f_3}|\psi_0\rangle = |\psi_1\rangle_3 = i\frac{1}{\sqrt{2}}|0\rangle|-\rangle_y + i\frac{1}{\sqrt{2}}|1\rangle|-\rangle_y$$
$$\text{iff} \quad f_3(0) = 1, f_3(1) = 1. \tag{8.11}$$

In (8.8)–(8.11), the operations on the mapping look fine to us because the process here is based upon the phase that was obtained from the kickback formation. Therefore, the issue of orthogonality is not so essential here as we consider the phase of each state to be guaranteed.

In addition, we apply U_{f_j} to the resulting state $|\psi_1\rangle_j$ and recall both the two values of the function f_j by going back to the initial state $|\psi_0\rangle$ from the resulting state $|\psi_1\rangle_j$.

By recalling the resulting state $|\psi_1\rangle_j$, we can evaluate simultaneously both the two values of $f_j(x)$ for all $x(= 0, 1)$. Interestingly, memory of functions in Deutsch's algorithm generalized gives us the ability to recall a perfect property of $f_j(x)$, namely, $f_j(x)$ itself. This stage means we read data from the Bloch sphere.

With the above, memory of functions in Deutsch's algorithm generalized is as follows:

1. Select the kind of a function f_j which is a logical function.
2. Select the unitary operator U_{f_j} along with the selected function f_j.
3. Apply U_{f_j} to $|\psi_0\rangle$ and store both the two values of the function f_j into the resulting state $|\psi_1\rangle_j$.
4. This stage means we write data on the Bloch sphere.
5. Select the same unitary operator U_{f_j} as the above.
6. Apply U_{f_j} to the resulting state $|\psi_1\rangle_j$ and recall both the two values of the function f_j by going back to the initial state $|\psi_0\rangle$ from the resulting state $|\psi_1\rangle_j$.
7. This stage means we read data from the Bloch sphere.

This is faster than that of its classical apparatus which would require at least 2^2 evaluations. It may be much likely that the Bloch sphere is a high-speed memory system for Deutsch's algorithm generalized.

9 Memory of functions in Deutsch's algorithm generalized and based on orthogonal states

We discuss memory of functions in Deutsch's algorithm generalized and based on orthogonal states. We propose the following initial state:

$$|\psi_0\rangle_d \otimes |\psi_0\rangle = |+\rangle_x|-\rangle_x \otimes |+\rangle_x|+\rangle_y. \tag{9.1}$$

Applying $U_{f_j} \otimes U_{f_j} (j = 0, 1, 2, 3)$ to $|\psi_0\rangle_d \otimes |\psi_0\rangle$, $U_{f_j} \otimes U_{f_j} |\psi_0\rangle_d \otimes |\psi_0\rangle = |\psi_1\rangle_{jd} \otimes |\psi_1\rangle_j$, therefore leaves us with one of four cases:

$$U_{f_0} \otimes U_{f_0} |\psi_0\rangle_d \otimes |\psi_0\rangle = |\psi_1\rangle_{0d} \otimes |\psi_1\rangle_0 = |+\rangle_x |-\rangle_x \otimes (\frac{1}{\sqrt{2}} |0\rangle |+\rangle_y + \frac{1}{\sqrt{2}} |1\rangle |+\rangle_y)$$
$$\text{iff} \quad f_0(0) = 0, f_0(1) = 0. \tag{9.2}$$

$$U_{f_1} \otimes U_{f_1} |\psi_0\rangle_d \otimes |\psi_0\rangle = |\psi_1\rangle_{1d} \otimes |\psi_1\rangle_1 = |-\rangle_x |-\rangle_x \otimes (\frac{1}{\sqrt{2}} |0\rangle |+\rangle_y + i \frac{1}{\sqrt{2}} |1\rangle |-\rangle_y)$$
$$\text{iff} \quad f_1(0) = 0, f_1(1) = 1. \tag{9.3}$$

$$U_{f_2} \otimes U_{f_2} |\psi_0\rangle_d \otimes |\psi_0\rangle = |\psi_1\rangle_{2d} \otimes |\psi_1\rangle_2 = -|-\rangle_x |-\rangle_x \otimes (i \frac{1}{\sqrt{2}} |0\rangle |-\rangle_y + \frac{1}{\sqrt{2}} |1\rangle |+\rangle_y)$$
$$\text{iff} \quad f_2(0) = 1, f_2(1) = 0. \tag{9.4}$$

$$U_{f_3} \otimes U_{f_3} |\psi_0\rangle_d \otimes |\psi_0\rangle = |\psi_1\rangle_{3d} \otimes |\psi_1\rangle_3 = -|+\rangle_x |-\rangle_x \otimes (i \frac{1}{\sqrt{2}} |0\rangle |-\rangle_y + i \frac{1}{\sqrt{2}} |1\rangle |-\rangle_y)$$
$$\text{iff} \quad f_3(0) = 1, f_3(1) = 1. \tag{9.5}$$

If we have the relations above (9.2)–(9.5), we have simultaneously both $f_j(0)$ and $f_j(1)$ by measuring the resulting state. The four states are completely orthogonal to one another. Therefore, we have zero error probability when we distinguish the four states one another. This stage means we write data on the Bloch sphere.

In addition, we apply $U_{f_j} \otimes U_{f_j}$ to the resulting state $|\psi_1\rangle_{jd} \otimes |\psi_1\rangle_j$ and recall both the two values of the function f_j by going back to the initial state $|\psi_0\rangle_d \otimes |\psi_0\rangle$ from the resulting state $|\psi_1\rangle_{jd} \otimes |\psi_1\rangle_j$. This stage means we read data from the Bloch sphere.

With the above, memory of functions in Deutsch's algorithm generalized and based on orthogonal states is as follows:

1. Select the kind of a function f_j which is a logical function.
2. Select the unitary operator $U_{f_j} \otimes U_{f_j}$ along with the selected function f_j.
3. Apply $U_{f_j} \otimes U_{f_j}$ to $|\psi_0\rangle_d \otimes |\psi_0\rangle$ and store both the two values of the function f_j into the resulting state $|\psi_1\rangle_{jd} \otimes |\psi_1\rangle_j$.
4. This stage means we write data on the Bloch sphere.
5. Select the same unitary operator $U_{f_j} \otimes U_{f_j}$ as the above.
6. Apply $U_{f_j} \otimes U_{f_j}$ to the resulting state $|\psi_1\rangle_{jd} \otimes |\psi_1\rangle_j$ and recall both the two values of the function f_j by going back to the initial state $|\psi_0\rangle_d \otimes |\psi_0\rangle$ from the resulting state $|\psi_1\rangle_{jd} \otimes |\psi_1\rangle_j$.
7. This stage means we read data from the Bloch sphere.

This is faster than that of its classical apparatus which would require at least 2^2 evaluations. It may be much likely that the Bloch sphere is a high-speed memory system for Deutsch's algorithm generalized and based on orthogonal states.

10 Memory of values in Deutsch's algorithm generalized

We propose memory of values in Deutsch's algorithm generalized by using all quantum states lying in the xy-plane. In what follows, we consider the Bloch sphere, especially, we consider a pure state lying on the surface of the Bloch sphere. From Deutsch's formula and the mapping U_f, we arrive at the following formulas:

$$U_f|0\rangle(\cos\frac{\theta}{2}|0\rangle + e^{i\phi}\sin\frac{\theta}{2}|1\rangle) = |0\rangle(\cos\frac{\theta}{2}|0\oplus f(0)\rangle + e^{i\phi}\sin\frac{\theta}{2}|1\oplus f(0)\rangle)$$
$$= \begin{cases} |0\rangle(\cos\frac{\theta}{2}|0\rangle + e^{i\phi}\sin\frac{\theta}{2}|1\rangle) & \text{if} \quad f(0)=0, \\ |0\rangle(\cos\frac{\theta}{2}|1\rangle + e^{i\phi}\sin\frac{\theta}{2}|0\rangle) & \text{if} \quad f(0)=1. \end{cases} \tag{10.1}$$

$$U_f|1\rangle(\cos\frac{\theta'}{2}|0\rangle + e^{i\phi'}\sin\frac{\theta'}{2}|1\rangle) = |1\rangle(\cos\frac{\theta'}{2}|0\oplus f(1)\rangle + e^{i\phi'}\sin\frac{\theta'}{2}|1\oplus f(1)\rangle)$$
$$= \begin{cases} |1\rangle(\cos\frac{\theta'}{2}|0\rangle + e^{i\phi'}\sin\frac{\theta'}{2}|1\rangle) & \text{if} \quad f(1)=0, \\ |1\rangle(\cos\frac{\theta'}{2}|1\rangle + e^{i\phi'}\sin\frac{\theta'}{2}|0\rangle) & \text{if} \quad f(1)=1. \end{cases} \tag{10.2}$$

To simplify, we suppose a quantum state lying in the xy-plain. So let (θ, ϕ) be $(\pi/2, \phi)$ and let (θ', ϕ') be $(\pi/2, \phi)$ in giving

$$U_f|0\rangle(|0\rangle + e^{i\phi}|1\rangle)/\sqrt{2} = \begin{cases} (e^{i\phi})^{f(0)}|0\rangle(|0\rangle + e^{i\phi}|1\rangle)/\sqrt{2} & \text{if} \quad f(0)=0, \\ (e^{i\phi})^{f(0)}|0\rangle(|0\rangle + e^{-i\phi}|1\rangle)/\sqrt{2} & \text{if} \quad f(0)=1. \end{cases} \tag{10.3}$$

$$U_f|1\rangle(|0\rangle + e^{i\phi}|1\rangle)/\sqrt{2} = \begin{cases} (e^{i\phi})^{f(1)}|1\rangle(|0\rangle + e^{i\phi}|1\rangle)/\sqrt{2} & \text{if} \quad f(1)=0, \\ (e^{i\phi})^{f(1)}|1\rangle(|0\rangle + e^{-i\phi}|1\rangle)/\sqrt{2} & \text{if} \quad f(1)=1. \end{cases} \tag{10.4}$$

We define the following notation:

$$|+\rangle_\phi = \frac{|0\rangle + e^{i\phi}|1\rangle}{\sqrt{2}}. \tag{10.5}$$

We define the initial state $|\psi_0\rangle$ as follows, using an imaginary number $e^{i\phi}$. Here, we use a quantum phase effect, which is a quantum phenomenon.

$$|\psi_0\rangle = \frac{1}{\sqrt{2}}|0\rangle|+\rangle_\phi + \frac{1}{\sqrt{2}}|1\rangle|+\rangle_\phi = |+\rangle_x|+\rangle_\phi, \quad \langle\psi_0|\psi_0\rangle = 1. \tag{10.6}$$

Applying $U_{f_j}(j = 0, 1, 2, 3)$ to $|\psi_0\rangle$, $U_{f_j}|\psi_0\rangle = |\psi_1\rangle_j$, therefore leaves us with one of four cases:

$$U_{f_0}|\psi_0\rangle = |\psi_1\rangle_0 = \frac{1}{\sqrt{2}}|0\rangle|+\rangle_\phi + \frac{1}{\sqrt{2}}|1\rangle|+\rangle_\phi$$
$$\text{iff} \quad f_0(0) = 0, f_0(1) = 0. \tag{10.7}$$

$$U_{f_1}|\psi_0\rangle = |\psi_1\rangle_1 = \frac{1}{\sqrt{2}}|0\rangle|+\rangle_\phi + e^{i\phi}\frac{1}{\sqrt{2}}|1\rangle|+\rangle_{-\phi}$$
$$\text{iff} \quad f_1(0) = 0, f_1(1) = 1. \tag{10.8}$$

$$U_{f_2}|\psi_0\rangle = |\psi_1\rangle_2 = e^{i\phi}\frac{1}{\sqrt{2}}|0\rangle|+\rangle_{-\phi} + \frac{1}{\sqrt{2}}|1\rangle|+\rangle_{\phi}$$
$$\text{iff} \quad f_2(0) = 1, f_2(1) = 0. \tag{10.9}$$

$$U_{f_3}|\psi_0\rangle = |\psi_1\rangle_3 = e^{i\phi}\frac{1}{\sqrt{2}}|0\rangle|+\rangle_{-\phi} + e^{i\phi}\frac{1}{\sqrt{2}}|1\rangle|+\rangle_{-\phi}$$
$$\text{iff} \quad f_3(0) = 1, f_3(1) = 1. \tag{10.10}$$

In (10.7)–(10.10), the operations on the mapping look fine to us because the process here is based upon the phase that was obtained from the kickback formation. Therefore, the issue of orthogonality is not so essential here as we consider the phase of each state to be guaranteed.

In addition, we apply U_{f_j} to the resulting state $|\psi_1\rangle_j$ and recall both the two values of the function f_j by going back to the initial state $|\psi_0\rangle$ from the resulting state $|\psi_1\rangle_j$. This stage means we read data from the Bloch sphere. By storing and recalling resulting states (10.7)–(10.10), we can read/write an arbitrary natural number in $\{0, 1, 2, 3, 4, 5, ..., +\infty\}$ as data as shown below.

Suppose we are using $\phi \in [0, 2\pi)$ constructing both the initial state and the resulting state. The possible value of $\phi/2\pi$ is in $[0, 1)$. Let us depicture $\phi/2\pi$ in the decimal system as follows:

$$\phi/2\pi = 0.a_1a_2...a_M = a_1/10 + a_2/10^2 + ... + a_M/10^M, \tag{10.11}$$

where $a_1, a_2, a_3, ..., a_M$ are natural numbers and $0 \le a_1 \le 9$, $0 \le a_2 \le 9$,...,$0 \le a_M \le 9$. There may exist a bit-string $b = (b_1...b_N)$ such that

$$a_1/10 + a_2/10^2 + ... + a_M/10^M = b_1/2 + b_2/2^2 + ... + b_N/2^N, \tag{10.12}$$

where M, N are natural numbers. This is a transformation into the binary system from the decimal system. Introduce a function $g(b)$ that transforms a bit-string into a natural number. Choose the function g such that, for the bit-string $b = (b_1...b_N)$,

$$g(b) = b_N 2^{N-1} + ... + b_2 2^1 + b_1. \tag{10.13}$$

We let M, N be $+\infty$. Then, we define a mapping

$$[0, 1) \to \{0, 1, 2, 3, 4, 5, ..., +\infty\}. \tag{10.14}$$

For example, if we are using $0.625(= \phi/2\pi)$ in the decimal system, then $a_1 = 6, a_2 = 2, a_3 = 5$. We have $b_1 = 1, b_2 = 0, b_3 = 1$ by the transformation into the binary system from the decimal system. Thus, the bit-string is $b = (101)$ and we obtain $g(b) = 5$. In this example, we can read/write five, in the decimal system, as data. So, we can read/write infinite data on the Bloch sphere by using the phase factor of a quantum state.

11 Conclusions

In conclusions, we have reviewed Deutsch's algorithm. We have also reviewed Deutsch's algorithm generalized. Deutsch's algorithm generalized has determined simultaneously all the mappings of the given function by one query. We have discussed Deutsch's algorithm generalized and based on orthogonal states. As a main discussion, we have proposed a novel definition for making memory in all of the quantum algorithms based on the Bloch sphere. Ultimately speaking, all of the quantum algorithms have relied on a high-speed memory system on the Bloch sphere. Clearly, the Bloch sphere has been a place in which quantum states are stored theoretically steadily. Thus, we can steadily have written data on the Bloch sphere through a quantum state itself. We can steadily have read data from the Bloch sphere through a quantum state itself. In short, from the property of quantum physics, the Bloch sphere has been a high-speed memory system for any Deutsch's algorithm at least in an algorithm level. We have discussed the fact that we can read/write infinite data on the Bloch sphere by using the phase factor of a quantum state.

Acknowledgments

We thank Soliman Abdalla, Jaewook Ahn, Josep Batle, Do Ngoc Diep, Mark Behzad Doost, Ahmed Farouk, Han Geurdes, Preston Guynn, Shahrokh Heidari, Wenliang Jin, Hamed Daei Kasmaei, Janusz Milek, Mosayeb Naseri, Santanu Kumar Patro, Germano Resconi, and Renata Wong for their valuable support.

Declarations

Ethical approval

We are in an applicable thought to ethical approval.

Competing interests

The corresponding author states that there is no conflict of interest.

Author contributions

Koji Nagata and Tadao Nakamura wrote and read the manuscript.

Funding

Not applicable.

Data availability

No data associated in the manuscript.

References

[1] J. J. Sakurai, *Modern Quantum Mechanics* (Addison-Wesley Publishing Company, 1995), Revised ed.

[2] A. Peres, *Quantum Theory: Concepts and Methods* (Kluwer Academic, Dordrecht, The Netherlands, 1993).

[3] M. Redhead, *Incompleteness, Nonlocality, and Realism* (Clarendon Press, Oxford, 1989), 2nd ed.

[4] J. von Neumann, *Mathematical Foundations of Quantum Mechanics* (Princeton University Press, Princeton, New Jersey, 1955).

[5] M. A. Nielsen and I. L. Chuang, *Quantum Computation and Quantum Information* (Cambridge University Press, 2000).

[6] A. S. Holevo, *Quantum Systems, Channels, Information, A Mathematical Introduction* (De Gruyter, 2012), https://doi.org/10.1515/9783110273403.

[7] K. Nagata, D. N. Diep, A. Farouk, and T. Nakamura, *Simplified Quantum Computing with Applications* (IOP Publishing, Bristol, UK, 2022), https://doi.org/10.1088/978-0-7503-4700-6.

[8] R. Rennie (Editor), *Oxford dictionary of physics* (Oxford University Press, 2015), Seventh ed.

[9] D. Deutsch, Quantum theory, the Church-Turing principle and the universal quantum computer. Proc. R. Soc. Lond. A **400**, 97 (1985). https://doi.org/10.1098/rspa.1985.0070

[10] D. Deutsch and R. Jozsa, Rapid solution of problems by quantum computation. Proc. R. Soc. Lond. A **439**, 553 (1992). https://doi.org/10.1098/rspa.1992.0167

[11] R. Cleve, A. Ekert, C. Macchiavello, and M. Mosca, Quantum algorithms revisited. Proc. R. Soc. Lond. A **454**, 339 (1998). https://doi.org/10.1098/rspa.1998.0164

[12] K. Nagata and T. Nakamura, Generalization of Deutsch's algorithm. Int. J. Theor. Phys. **59**, 2557 (2020). https://doi.org/10.1007/s10773-020-04522-0

[13] E. Bernstein and U. Vazirani, Quantum complexity theory. Proceedings of 25th Annual ACM Symposium on Theory of Computing (STOC '93), p. 11 (1993). https://doi.org/10.1145/167088.167097

[14] E. Bernstein and U. Vazirani, Quantum Complexity Theory. SIAM J. Comput. **26**, 1411 (1997). https://doi.org/10.1137/S0097539796300921

[15] K. Nagata, G. Resconi, T. Nakamura, J. Batle, S. Abdalla, and A. Farouk, A generalization of the Bernstein-Vazirani algorithm. MOJ Ecol. Environ. Sci. **2**(1), 00010 (2017). https://doi.org/10.15406/mojes.2017.02.00010

[16] K. Nagata, H. Geurdes, S. K. Patro, S. Heidari, A. Farouk, and T. Nakamura, Generalization of the Bernstein-Vazirani algorithm beyond qubit systems. Quantum Stud.: Math. Found. **7**, 17 (2020). https://doi.org/10.1007/s40509-019-00196-4

[17] C.-Y. Chen, C.-Y. Chang, and C.-C. Hsueh, Solving Bernstein and Vazirani's Problem with the 2-bit Permutation Function. Quantum Information Processing, Volume 21, Issue 1 (2022), Article number: 15. https://doi.org/10.1007/s11128-021-03345-0

[18] D. R. Simon, On the power of quantum computation. Proceedings of 35th IEEE Annual Symposium on Foundations of Computer Science, p. 116 (1994). https://doi.org/10.1109/SFCS.1994.365701

[19] P. W. Shor, Algorithms for quantum computation: discrete logarithms and factoring. Proceedings of 35th IEEE Annual Symposium on Foundations of Computer Science, p. 124 (1994). https://doi.org/10.1109/SFCS.1994.365700

[20] L. K. Grover, A fast quantum mechanical algorithm for database search. Proceedings of 28th Annual ACM Symposium on Theory of Computing, p. 212 (1996). https://doi.org/10.1145/237814.237866

[21] K. Nagata and T. Nakamura, Some Theoretically Organized Algorithm for Quantum Computers. Int. J. Theor. Phys. **59**, 611 (2020). https://doi.org/10.1007/s10773-019-04354-7

[22] J. Ossorio-Castillo, U. Pastor-Díaz, and J. M. Tornero, A generalisation of the Phase Kick-Back. Quantum Information Processing, Volume 22, Issue 3 (2023), Article number: 143. https://doi.org/10.1007/s11128-023-03884-8

[23] T. Nakamura and K. Nagata, Physics' Evolution Toward Computing. Int. J. Theor. Phys. **60**, 70 (2021). https://doi.org/10.1007/s10773-020-04661-4

[24] G. Jaeger and A. Shimony, Optimal distinction between two non-orthogonal quantum states. Phys. Lett. A **197**, 83 (1995). https://doi.org/10.1016/0375-9601(94)00919-G

[25] A. Peres and D. R. Terno, Optimal distinction between non-orthogonal quantum states. J. Phys. A: Math. Gen. **31**, 7105 (1998). https://doi.org/10.1088/0305-4470/31/34/013