Quantum Information and Computing Simple Quantum Algorithms- Deutsch Algorithm and Deutsch - Jozsa Algorithms

Dipan Kumar Ghosh
Physics Department,
Indian Institute of Technology Powai, Mumbai 400076

March 16, 2017

1 Introduction

In the lectures given so far, we have talked about the basic postulates of quantum mechanics, elements of linear algebra and have also discussed essential components of a quantum computer such as quantum registers, gates and circuits. What we will do in this lecture is to introduce some simple algorithms. They are simple because though the algorithm take very simple problems which do not need much resource even in a classical computer, they help us in understanding the principles of quantum algorithms while bringing out the fact that quantum computation has some advantages over traditional (or classical) computers. In order to effectively use the quantum principles to solve a problem using a quantum computer, we need to focus on issues connected with the designing of a quantum algorithm. The basic ingredients for such a design are as follows:

1. **State of a quantum register:**
   In a classical Turing machine, the state of a register is fixed at any instant of time. For instance, if we consider a 3 bit register, it can be in any one of the states: 000, 001, 010, 011, 100, 101, 110 and 111. By analogy, an n-bit classical register can at any instant be in a state defined by one of the 2^n values.
   By contrast, an n-bit quantum register is **simultaneously** in all these states. The state of the quantum register is a vector in 2^n dimensional complex vector space. The general state of a single qubit is written in terms of the basis states $\{0, 1\}$ as

$$| \psi_1\rangle = \alpha \mid 0\rangle + \beta \mid 1\rangle \tag{1}$$

with $\mid \alpha \mid^2 + \mid \beta \mid^2 = 1$. One can regard this as a vector in two dimensional complex Hilbert space. The generalization to n- qubits is straight forward. Denoting the basis of the 2^n dimensional complex vector space by the ket $\{\mid i_k\rangle\}$, an arbitrary state $\mid \psi\rangle$ can be expressed as

$$\mid \psi\rangle = \sum_{k=0}^{2^n-1} w_k \mid i_k\rangle \tag{2}$$

where w_k is the complex weight factor for the $k-$ th basis state, with

$$\sum_{k=0}^{2^n-1} \mid w_k \mid^2 = 1 \tag{3}$$

Thus, a quantum register with n-qubits requires specification of 2^n complex numbers. The possibility of storing states of superposition of 2^n states at the same time implies that an exponentially large amount of information can be stored in the memory of a quantum computer compared to that of a classical computer using the same size of the registers.

2. **Quantum Parallelism:**
A computation involves taking in an input x and producing an output $f(x)$. In a quantum computer, these operations are done by unitary operators. If the input has n qubits and the output is of m qubits,

$$f(x) : \{0,1\}^n \rightarrow \{0,1\}^m$$

Though the actual computation involves use of many auxiliary registers, we will represent the process of computation by an $n-$ qubit input register and an $m-$ qubit output register. The process of computation is represented by

$$U\{\mid x\rangle_n \mid 0\rangle_m\} = \mid x\rangle_n \mid f(x)\rangle_m \tag{4}$$

where $\mid 0\rangle_m$ indicates the initial empty state of the output register.
When an operation U is performed on an input register containing superposition of 2^n states, the computation of $f(x)$ gets done in a single computational step and one gets $f(x)$ value corresponding to *all* the input values, i.e.

$$U\left(\sum_x \mid x\rangle \mid 0\rangle\right) \rightarrow \sum \mid x\rangle \mid f(x)\rangle \tag{5}$$

which contains information about all values of $f(x)$..
This is different from what happens in a classical parallel computer where the operations are done by partitioning the problem into different parts whose computation may be performed independently of one another. The result of such parallel computation by different processors are then integrated to arrive at the final solution

of the problem. In quantum machines, parallelism is inherent and is like having parallel computation in a serial machine. The evaluation of the functions is done by an **Oracle** which is essentially a blackbox which hides from view the details of computation much like a classical subroutine.

3. **Collapse of wave function and the process of measurement:**
   After a computation of the function $f(x)$ has been made, one would like to know the value of $f(\mid x\rangle)$ corresponding to a given $\mid x\rangle$. In a classical computer, we could simply read the values or print them. The process of reading the value does not disturb the output register in any way.
   In a quantum computer, on the other hand, there is no apparent way to do it, for, the process of measurement makes the system collapse to one of the eigenkets of the observable that is being measured. As the input state $\mid x\rangle$ is a linear combination of various states $\mid i_k\rangle$ with probability $\mid \omega_k \mid^2$, when a measurement is made, one gets the value of $f(\mid i_k\rangle)$ with the probability $\mid \omega_k \mid^2$. As there is no way of predicting what particular state the state would collapse to as a consequence of the measurement, the situation is no different from doing the calculation in a classical computer with a random input.

4. **Entanglement:**
   If a measurement of the output register is made, it will return m qubits of the value of $\mid f(x)\rangle$ corresponding to only one of the values of the input x, say $x = r$. The n qubit input register will now have a superposition of all those values of x which give the same output r. (In case the function is one-one, it will now hold only that value of x which results in the output r.
   The above is a consequence of the fact that the n bit input register and the m bit output registers are strongly correlated, i.e., they are entangled. When a measurement of the content of either register is made, the content of the other register would collapse to the corresponding value.

5. **Quantum No-Cloning Theorem**
   It may be thought that if we could make several identical copies of the two registers before attempting a measurement of either of the registers, we may be able to get the value of $f(x)$ for the state $\mid x\rangle$ in which we are interested in by repeating the measurement process on the copies. However, according to the quantum no-cloning theorem, an arbitrary quantum state cannot be copied.

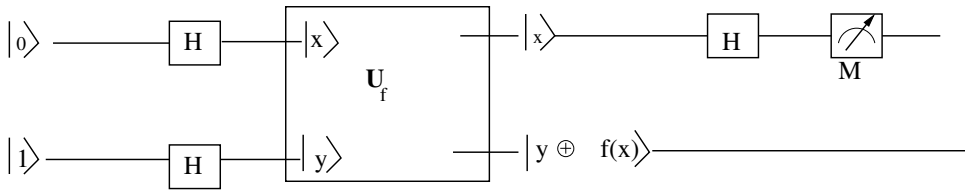
2 Deutsch Problem

Deutsch problem is an example of a quantum algorithm which seeks to distinguish between two classes of function that can be computed $f : \{0, 1\} \rightarrow \{0, 1\}$. One class of function is **constant**, i.e., $f(0) = f(1) = 0$ or $f(0) = f(1) = 1$ while in the other class of

function (called **a balanced function**)there are equal number of zero and one, i.e. either $f(0) = 0, f(1) = 1$ or $f(0) = 1, f(1) = 0$. Though this is a rather trivial computation in a classical computer, it establishes an important fact that quantum computer performs a task more efficiently than a classical computer. An oracle can compute the functions using a unitary transformation:

$$U_f :\mid x\rangle \mid y\rangle \rightarrow \mid x\rangle \mid y \oplus f(x)\rangle \tag{6}$$

where the state $\mid x\rangle$ is a single quit input register and $\mid y\rangle$ is the target register into which $f(x)$ is to be copied. The symbol $\oplus$ denotes an exclusive OR operation, which is addition modulo 2. Note that a classical computer requires two queries before deciding which of the class of functions $f(x)$ belongs to. In a quantum computer, it is possible to decide the query in a single query, though, unlike in the case of classical computer, the query does not at the same time give us the value of the function $f(x)$ itself.
Let the input register be prepared to contain the state $\frac{\mid 0\rangle + \mid 1\rangle}{\sqrt{2}}$, which is obtained by passing the quit $\mid 0\rangle$ through a Hadamard gate. Similarly, the target register $\mid y\rangle$ is prepared in the state $\frac{\mid 0\rangle - \mid 1\rangle}{\sqrt{2}}$, which can be obtained by passing the quit $\mid 1\rangle$ through a Hadamard gate. The schematic diagram of the quantum circuit along with the oracle is shown below.



Thus the input states in the two registers are given by the expression

$$\frac{1}{\sqrt{2}}(\mid 0\rangle + \mid 1\rangle) \otimes \frac{1}{\sqrt{2}}(\mid 0\rangle - \mid 1\rangle) = \frac{1}{2}\left[\mid 00\rangle + \mid 10\rangle - \mid 01\rangle - \mid 11\rangle\right]$$

On application of U_f, the second bit is transformed to $y \oplus f(x)$ giving,

$$\frac{1}{2}\left[\mid 0, f(0)\rangle + \mid 1, f(1)\rangle - \mid 0, \bar{f}(0)\rangle - \mid 1, \bar{f}(1)\rangle\right]$$

where we have used $0 \oplus f(x) = f(x)$ and $1 \oplus f(x) = \bar{f}(x)$, the bar over f stands for the complement of f. Thus if $f(0) = f(1)$, we get the output to be

$$\frac{1}{2}\left[\mid 0, f(0)\rangle + \mid 1, f(0)\rangle - \mid 0, \bar{f}(0)\rangle - \mid 1, \bar{f}(0)\rangle\right] = \frac{1}{2}(\mid 0\rangle + \mid 1\rangle)(\mid f(0)\rangle - \mid \bar{f}(0)\rangle) \tag{7}$$

On the other hand, if $f(0) \neq f(1)$, we get, using $f(1) = \bar{f}(0)$ and $\bar{f}(1) = f(0)$,

$$\frac{1}{2}\left[\mid 0, f(0)\rangle + \mid 1, \bar{f}(0)\rangle - \mid 0, \bar{f}(0)\rangle - \mid 1, \bar{f}(0)\rangle\right] = \frac{1}{2}(\mid 0\rangle - \mid 1\rangle)(\mid f(0)\rangle - \mid \bar{f}(0)\rangle) \tag{8}$$

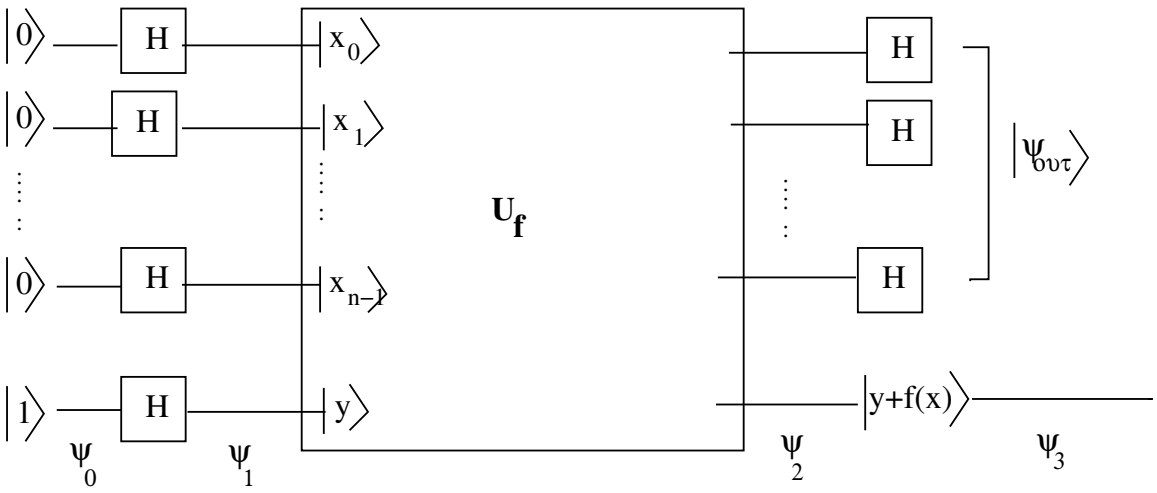
Note that the states in the two registers are now entangled. We pass the first register through a Hadamard gate. If $f(0) = f(1)$, the state of the first register becomes $\mid 0\rangle$, and if $f(0) \neq f(1)$, the content of the first register is $\mid 1\rangle$. Thus a measurement of the first register will tell us what type of function $f(x)$ is. Note that in both cases, the content of the second register is the same, viz. $\mid f(0)\rangle - \mid \bar{f}(0\rangle$, unto an overall phase factor, which has no effect on a measurement. By measuring the second register, one cannot get information on the value of $f(0)$. In Deutsch algorithm, discussed above, we identify one of the two one qubit functions $f : (0, 1) \rightarrow 0, 1$ whose range and domain are both single qubits. The functions were either constant or what we called a balanced function, the latter being defined as a function for which $f(0) \neq f(1)$. We saw that while a classical computer would require two queries to determine which type of function has been computed by the oracle, a quantum computer can identify it in a single query. It must be emphasized that the classical algorithm had an advantage of yielding the value of the function as well which the quantum computer evaluation did not do.
In this following we extend Deutsch algorithm to the case when the input is an n-qubit string while the output still remains a single qubit, i.e. $f : \{0, 1\}^{\otimes n} \rightarrow 0, 1$.

3 Deutsch-Jozsa Algorithm

We may extend Deutsch algorithm to the case where the input is an n bit string. The oracle U_f evaluates a function $f : \{0, 1\}^n \rightarrow \{0, 1\}$ which, once again belongs to one of the two classes, viz. either $f(x) =$ constant or $f(x) =$ a balanced function, i.e., the number of zeros is equal to the number of ones.
Like in the previous case, we pass each bit of the initial input strings of $\mid 0\rangle$ through Hadamard gates. This gives isa uniform linear combination of the n quit basis functions:

$$\begin{aligned} \mid x\rangle =& \mid x_{n-1} \cdots x_1 x_0\rangle \\ &= \prod_{j=0}^{2^n-1} \frac{\mid 0\rangle + \mid 1\rangle}{\sqrt{2}} \\ &= \frac{1}{2^{n/2}} \sum_{j=0}^{2^n-1} \mid j\rangle \end{aligned} \tag{9}$$

where $\mid j\rangle$ is n quit state $\mid j_{n-1} \cdots j_1 j_0\rangle$ and the sum is over all possible combination of the values of $j_0, j_1, \ldots j_{n-1}$ each of which can be either 0 or 1.



The oracle computes $f(x)$ for the input $\mid x\rangle$ and, by linearity, it is computed simultaneously for all values of $\mid j\rangle$. Thus, if the target state is $\mid y\rangle$, the action of the oracle is given by

$$U_f :\mid x\rangle \mid y\rangle \rightarrow \mid x\rangle \mid y \oplus f(x)\rangle$$

i.e.

$$\frac{1}{2^{n/2}} \sum_{j=0}^{2^n-1} \mid j\rangle \mid y\rangle \rightarrow \frac{1}{2^{n/2}} \sum_{j=0}^{2^n-1} \mid j\rangle \mid y \oplus f(j)\rangle$$

If y were zero, $\mid y \oplus f(j)\rangle$ would be $\mid f(j)\rangle$. On measuring the n quit output lines from 1 to n, the state in the (n+1)-th line (target bit) would then have collapsed to the value of $f(j)$ corresponding to the measured value of j.

Like in Deutsch problem, we prepare the target bit as $\frac{1}{\sqrt{2}}(\mid 0\rangle - \mid 1\rangle)$ before subjecting the (n+1)- th qubit to the oracle. The output bit can be expressed as follows:

$$\begin{aligned} &\frac{1}{2^{(n+1)/2}} \sum_{j=0}^{2^n-1} [\mid j, 0 \oplus f(j)\rangle - \mid j, 1 \oplus f(j)\rangle \\ &= \frac{1}{2^{(n+1)/2}} \sum_{j=0}^{2^n-1} [\mid j, f(j)\rangle - \mid j, \bar{f}(j)\rangle] \\ &= \frac{1}{2^{(n+1)/2}} \sum_{j=0}^{2^n-1} \mid j, f(j) - \bar{f}(j)\rangle \end{aligned} \tag{10}$$

Since $f(j)$ can take only two values, 0 and 1, $\mid f(j) - \bar{f}(j)\rangle = \mid 0\rangle - \mid 1\rangle$ if $f(j) = 0$ and is equal to $\mid 1\rangle - \mid 0\rangle$ if $f(j) = 1$. One can combine the two alternatives by writing the output as

$$\frac{1}{2^n} \sum_{j=0}^{2^n-1} \mid j\rangle (-1)^{f(j)} \frac{\mid 0\rangle - \mid 1\rangle}{\sqrt{2}}$$

One can associate the sign with the first n qubits and write the output as

$$\left(\frac{1}{2^n}\sum_{j=0}^{2^n-1}(-1)^{f(j)} \mid j\rangle\right)\frac{\mid 0\rangle - \mid 1\rangle}{\sqrt{2}} \tag{11}$$

When the first n qubits of the output are parallel passed through Hadamard gates, the resulting state, represented by ψ_3 in the figure, can be expressed as follows. Using the fact that a qubit $\mid j_i\rangle$ becomes $\frac{\mid 0\rangle + (-1)^{j_i} \mid 1\rangle}{\sqrt{2}} = \frac{1}{\sqrt{2}}\sum_{k_i=0,1}(-1)^{j_i k_i} \mid k_i\rangle$, we have on writing $\mid j\rangle$ in the bit notation,

$$\begin{aligned}
\mid j\rangle =& \mid j_{n-1}j_{n-2}\cdots j_1 j_0\rangle \\
&= \frac{1}{2^{n/2}}\sum_{k_{n-1}=(0,1)\ldots k_1=(0,1)k_0=(0,1)} (-1)^{j_{n-1}k_{n-1}+\ldots+j_1k_1+j_0k_0} \mid k_{n-1}\ldots k_1k_0\rangle \\
&= \frac{1}{2^{n/2}}\sum_{k_{n-1}=(0,1)\ldots k_1=(0,1)k_0=(0,1)} (-1)^{\sum_i j_i k_i} \mid k_{n-1}\ldots k_1k_0\rangle \\
&= \frac{1}{2^{n/2}}\sum_{k=0}^{2^n-1}(-1)^{k\cdot j} \mid k\rangle
\end{aligned} \tag{12}$$

where $k\cdot j = \sum_i k_i j_i$ is the bitwise product of k with j. Thus, the output,given by (11), after passing through the Hadamard gates becomes

$$\frac{1}{2^n}\sum_{j=0}^{2^n-1}\sum_{k=0}^{2^n-1}(-1)^{j\cdot k - f(j)} \mid k\rangle\frac{1}{\sqrt{2}}(\mid 0\rangle - \mid 1\rangle) \tag{13}$$

where $j,k \in \{0,1\}^n$.
If the function $f(j) = f_0 =$ constant, we may take it out of the sum, in which case, $\sum_j(-1)^{j\cdot k}$ will have as many positive term as the negative terms and will give zero, except in one case where $\mid k\rangle =\mid 0\rangle^{\otimes n}$ when the phase factor becomes 1 for each j. In this case the output becomes

$$(-1)^{f_0} \mid 0\rangle^{\otimes n}\frac{1}{\sqrt{2}}(\mid 0\rangle - \mid 1\rangle) \tag{14}$$

If, on the other hand, the function is balanced, the coefficient of $\mid k\rangle =\mid 0\rangle^{\otimes n}$ is $\frac{1}{2^n}\sum_{j=0}^{2^n-1}(-1)^{f(j)} = 0$, i.e., the probability of the state $\mid 0\rangle^{\otimes n}$ is zero. Thus a single query is sufficient to determine whether the function is balanced or is constant.
It would seem that the number of queries required by a classical computer to determine whether the function is balanced or is constant could be $(2^{n-1}+1)$ and hence the solution is not possible in polynomial time. Quantum algorithm appears to speed up the query exponentially. However, in practice, since a classical search is done without replacement, the probability of success in polynomial time is quite high.

4 Bernstein-Vazirani Problem

Deutsch Jozsa algorithm can be used to solve the following problem known as Bernstein Vazirani problem. We are given an oracle which can compute a function $x \in \{0,1\}^{\otimes n}$, given by

$$f(x) = a \cdot x$$

where $a \cdot x$ is the bitwise product of the n qubit string x with an unknown string a,

$$a \cdot x = a_0 x_0 + a_1 x_1 + \cdots + a_{n-1} x_{n-1} \pmod 2$$

The function calculates only a single bit output.

The problem is to determine the unknown string a. Since the addition is modulo 2, one can determine a by a classical subroutine by n queries. This is because the m-th bit of the above sum for $0 \leq m \leq n$ is $2^{m-1} a_{m-1} x_{m-1}$. Thus, given x_{m-1}, we can determine a_{m-1}. In particular,

$$\begin{aligned} f(100\ldots0) &= a_{n-1} \\ f(010\ldots0) &= a_{n-2} \\ &\cdots\cdots \\ &\cdots\cdots \\ f(000\ldots1) &= a_0 \end{aligned}$$

In a quantum computer, it is possible to determine it with a single query of the oracle. As in Deutsch-Jozsa problem, we prepare the n qubit as a uniform linear combination of all n bit computational basis : $\mid x\rangle = \frac{1}{\sqrt{2^n}} \sum_j \mid j\rangle$. When the target bit is $\frac{(\mid 0\rangle - \mid 1\rangle)}{\sqrt{2}}$, the oracle gives, using $f(j) = a \cdot j$

$$\frac{1}{2^n} \sum_{j=0}^{2^n-1} \sum_k (-1)^{j\cdot k + a\cdot j} \mid k\rangle \frac{1}{\sqrt{2}} (\mid 0\rangle - \mid 1\rangle)$$

We can perform the sum over j

$$\sum_{j=0}^{2^n-1} (-1)^{j\cdot(k+a)} = \prod_{i=1}^{n} \sum_{j_i=0}^{1} (-1)^{j_i(a_i+k_i)} = \prod_{i=1}^{n} \left(1 + (-1)^{a_i+k_i}\right)$$

Now, $(-1)^{a_i+k_i}$ takes the value $+1$ only when a_i and k_i are equal, i.e. with both are zero or both 1. Thus, except where the i -th bit of $\mid a\rangle$ is identical to the i-th bit of k for every i, at least one term in the product will be zero. The output then becomes

$$\mid a\rangle^{\otimes n} \frac{\mid 0\rangle - \mid 1\rangle}{\sqrt{2}}$$

A measurement of the first n bits of the output will determine a with certainty.