Your objective is to submit a writeup as well as a quantum circuit code for the implementation of Deutzch's Algorithm

Deutzch's Algorithm

Deutzch's algorithm is considered to be one of the first quantum computing algorithms to clearly showcase the advantage that a quantum computer has over classical computers.
It can determine the nature of a secret function.A function is constant if the output of the function is the same for all the inputs provided and a function is balanced if they output 0 for half the inputs and 0 for half the outputs.The intriguing part is that quantum computers can achieve this in only **one iteration.**
This writeup will provide a walkthrough from the basics of quantum computing to the Deutzch's problem and its solution along with its quantum circuit code and is divided into 5 chapters

Chapter I-Quantum Computing Fundamentals

Quantum computing refers to the branch of computation which uses the concepts of quantum physics.With the principles of quantum physics,it can compute differently than a classical computer.The basic unit of measurement in a quantum computer is called a qubit.This is analogous to a classical bit in a regular compute,but exists in superposition:The phenomenon of being in more than 1 state at a time.A qubit can be expressed in the form:

$$|\psi\rangle = \alpha\,|0\rangle + \beta\,|1\rangle$$

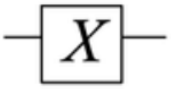
Here |> refers to the dirac notation also know as ket
|0> is called ket 0 and has a matrix form of(1 0) {2x1 matrix}
|1> is called ket 1 and has a matrix form of (0 1)
Multi-qubit states can be expressed as the tensor product of the single qubit states eg.|01>=|0>*tensor product symbol* |1>

Alpha beta are the probability amplitudes
Psi refers to the current state of the qubit
It's just a label for the quantum state.
Upon observing the qubit,it collapses into one of the given states,either 1 or 0.The probability of getting a 0 is (alpha)^2 and (beta)^2 for 1.
Another fundamental property of qubit is entanglement.When two or more qubits are entangled cannot possess 2 independent states.Upon a change in one qubit,the subsequent qubits are bound to change,irrespective of their local coordinates.
We cannot write the two-qubit state as a tensor product of two single-qubit states.
I.e if the 2 qubits were not entangled,then the 2-system qubit could be written as

$$|\psi\rangle = |\phi_1\rangle \otimes |\phi_2\rangle$$

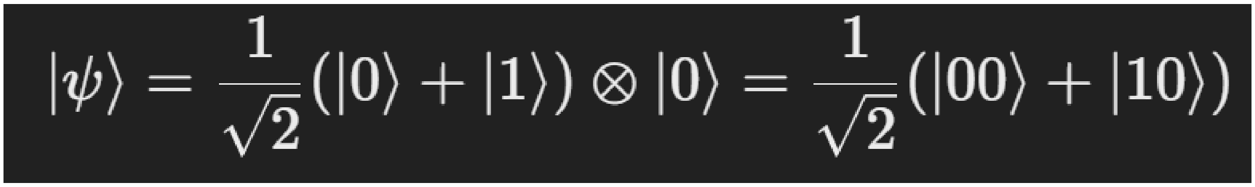
Chapter 2:Quantum Logic Gates

They are analogous to the logic gates in classical circuits and refer to the operations which change the state of a qubit.
The fundamental gates used in Deutsch's algorithm ia
1.Pauli X gate : They are analogous to the NOT operation in classical circuits and flips the state of a qubit.For ex.if the initial state of a qubit is |1> then after applying the X gate,it will change to |1> and vice versa.In matrix format it is represented as



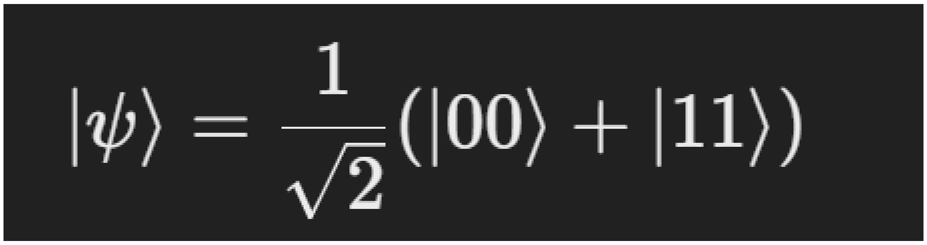
$$\begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$$

2.Controlled-NOT gate(CNOT):this is a 2-qubit system gate where the first qubit is called the control qubit and the other is called as the target qubit.The CNOT flips the qubit only if control qubit is in ket 1 state.
|00> will result in |00>
|01> will result in |01>
|10> will result in |11>
|11> will result in |10>
Generally since the qubit is in superposition,the control qubit will thus also be in superposition which is given by



$$|\psi\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle) \otimes |0\rangle = \frac{1}{\sqrt{2}}(|00\rangle + |10\rangle)$$

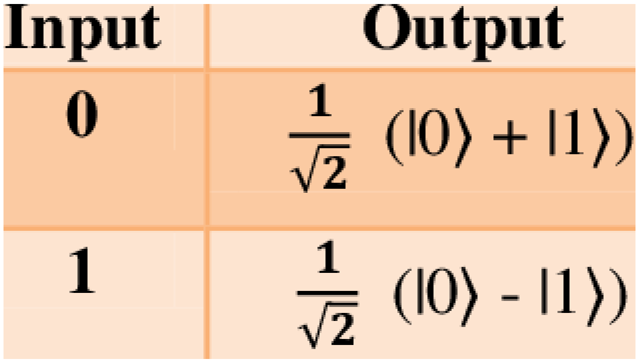
After applying CNOT gate,the control qubit which has ket 1 present will have their target qubit changed



$$|\psi\rangle = \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle)$$

Now this results in quantum entanglement because we cannot separate these into 2 different qubit systems.

3.Hadamard Gate: This is considered to be the most important quantum logical gate used in Deutzch's algorithm.It is used to create superposition,acts upon a single qubit and has the ability to make it into an equal superposition of |0> and |1>



Input	Output
0	$\frac{1}{\sqrt{2}}(\vert 0\rangle + \vert 1\rangle)$
1	$\frac{1}{\sqrt{2}}(\vert 0\rangle - \vert 1\rangle)$

H|0> is called as the |+> state
H|1> is called as the |-> state

The matrix definition of the Hadamard Gate is given by

$$H = \frac{1}{\sqrt{2}}\begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$$

The matrix form of the quantum logic gates cannot just be any 2x2 matrix.It needs to satisfy 2 conditions

1. **They should be Hermitian Matrices:**the given matrix should be equal to its adjoint.Adjoint is the transpose of each element of the matrix.This ensures that the eigenvalues,also known as the observables are real valued.Observables are measurable quantities so they should be real in order to be measured.
2. **They should be Unitary:**The matrix multiplication of the regular matrix and the adjoint should be the identity matrix.This ensures the conservation of probability

Chapter 3:Oracles and the Deutzch's problem

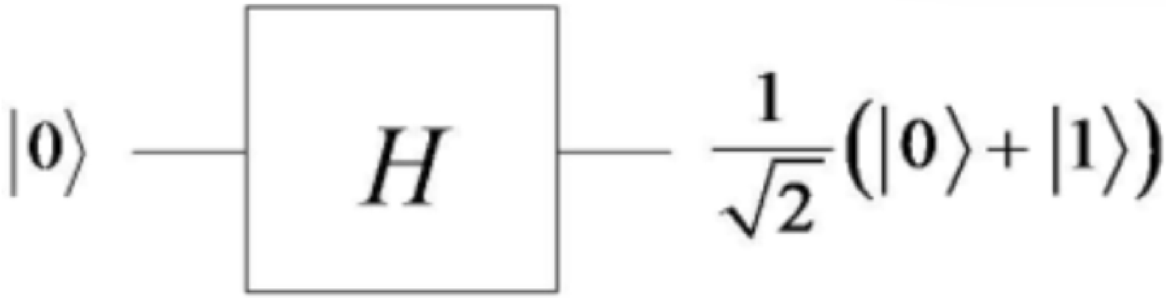
Oracles is basically a special black box which stores the function insides.The input values are fed and the output is generated,however the methodology and the working is not known. Oracle is like a mystery machine which can compute a function which is fed into it without actually knowing what the function is and what it does.

Now,for the deutsch's problem we take a single bit function f(x) and can have only 2 possible input values,0 and 1.The function is supposed to be either constant or balanced(explained in the introduction).
A classical computer will solve a question like this by checking each input to see if the function is constant or balanced.
For example,if the input is 0 and the output is also 0,we cannot determine if the function is constant(the output will be 0 in both the input cases(0 and 1)) or balanced(the output is 0 for input 0 and output is 1 for input 1)
So in order to find the type of function,we would require **(2^n-1)+1** iterations.

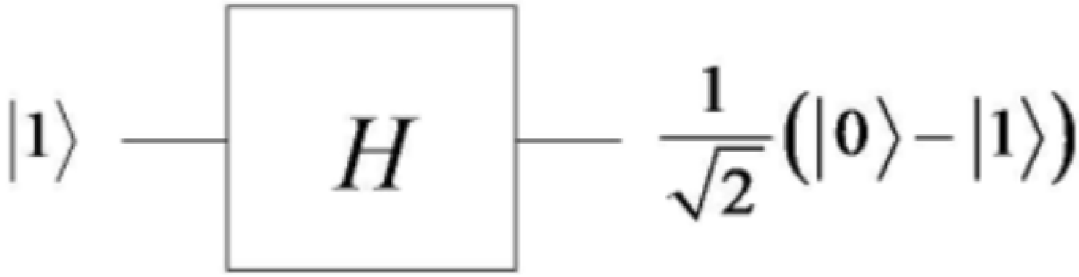
Now,Deutzch’s algorithm can solve this problem with just one iteration in the oracle by the help of superposition and interference.

Chapter 4:Deutsch’s Algorithm

Before using the Algorithm,a two qubit system is prepared in which the first qubit(used as input) is in a superposition of 0 and 1.The second qubit will be used by the oracle to output the function value[without actually knowing what the function is or what is does]
Initially,we will prepare the qubits in a known state.the input qubit is taken as |0> and the output qubit is taken as |1>
The reason for this is:
1.This results in a deterministic outcome and we get the correct values for a constant function and balanced function(0 and 1 resp.)
2.We need the output as 1 in order to perform phase kickback after applying the Hadamard gate.
Now,the Hadamard gate is applied to both the qubits-thereby creating superposition
The input qubit was |0>.After Hadamard,the input qubit becomes



The output qubit is |1>.After Hadamard,the output qubit becomes



Also known as |->,this helps to imprint a phase kickback on the input qubit's state.This means that the oracle will not disturb the output qubit at all.

Now,we apply the oracle,represented by Uf,which maps |x,y> to x and the tensor product of y and f(x)

$$|x, y\rangle \to |x,\, y \oplus f(x)\rangle$$

(tensor product is the symbol of circle with x inside it and XOR is symbol of circle with + inside it-got confused a bit)

Now we know |y> is taken as 1

After applying Hadamard gate,it has become

$$\frac{1}{\sqrt{2}}(|0\rangle - |1\rangle)$$

which is also called as |->

If f(x)=0 then

$$y \oplus f(x)\rangle$$

Which is equal to y[nothing changes]

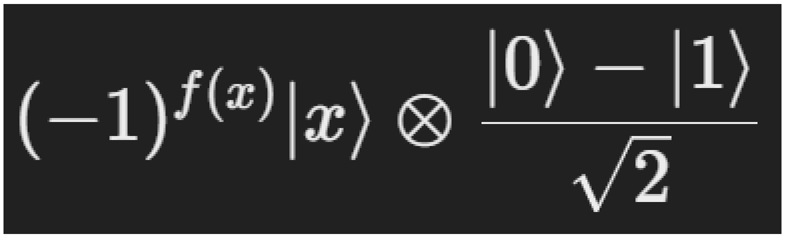
But if f(x)=1

It flips |0> to |1>

So |0>-|1> become |1>-|0>=-[|0>-|1>]

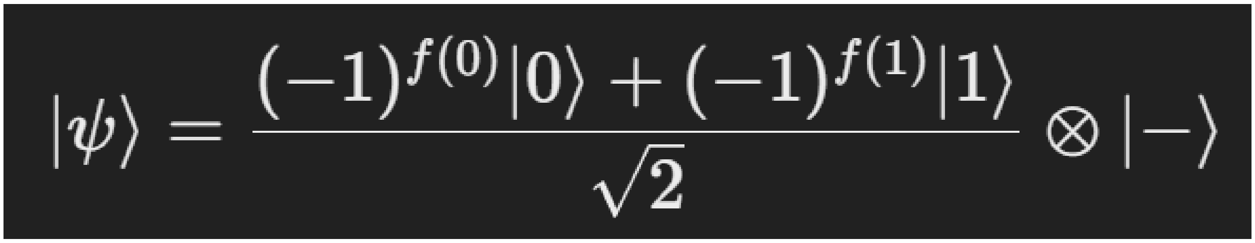
(phase of -1)

Now,the oracle of the tensor product of |x> and|-> gives



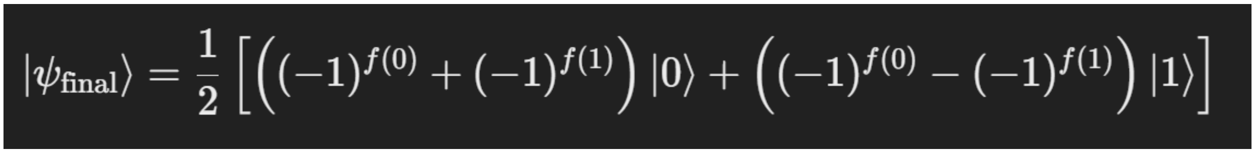
$$(-1)^{f(x)}|x\rangle \otimes \frac{|0\rangle - |1\rangle}{\sqrt{2}}$$

f(x) is solely used as a phase now while keeping the output qubit untouched.
Now the qubits are in a state of



$$|\psi\rangle = \frac{(-1)^{f(0)}|0\rangle + (-1)^{f(1)}|1\rangle}{\sqrt{2}} \otimes |-\rangle$$

Where |-> is considered as output and the initial part is the input qubit state. Applying Hadamard gate on the input qubit state,the final qubit state becomes



$$|\psi_{\text{final}}\rangle = \frac{1}{2}\left[\left((-1)^{f(0)} + (-1)^{f(1)}\right)|0\rangle + \left((-1)^{f(0)} - (-1)^{f(1)}\right)|1\rangle\right]$$

If the function is constant,then the input becomes |0> and if the function is balanced,the input becomes |1>.

Chapter 5: Conclusion

Deutsch Algorithm looks like a very extensive framework for a simple 2 output question,but it is quite significant in the history of quantum computing as it became the first algorithm which could outperform a classical computer with a well enough margin.Moreover,this advantage was able to be achieved due to the properties of qubits such as superposition and parallelism.This principle can be scaled up to much bigger questions where the classical computations may take (2^n-1)+1 iterations to conclude with an answer,meanwhile quantum computers can get the output exponentially faster.

However,in 2025,Deutsch algorithm is nothing but a proof of concept that the quantum computers outperformed the classical computers in a certain task.The task is itself way too small and does not utilise the quantum power well.The best part about Deutsch’s algorithm was that it was able to lay the foundation for other complex algorithms such as Grover’s algorithm,Shor’s algorithm etc.