Quantum Computing notes

jun.mendoza.01@gmail.com

1 qubit

Define:

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

Where:

α **|0>**

0 state

α is the coefficient of **0** state. It is the percentage that the qubit will collapse in the **0** state

β **|1>**

1 state

β is the coefficient of **1** state. It is the percentage that the qubit will collapse in the **1** state

1 qubit - Superposition

The coefficients α and β are percentages that the qubit will collapse in that state

Given a unit vector: $[0.8, 0.6]$.

This can be visualised in the form $\psi = \alpha|0\rangle + \beta|1\rangle$ as $\psi = 0.8|0\rangle + 0.6|1\rangle$

This means that the qubit is in a superposition where it has a 0.8 chance of collapsing in state |0> and a 0.6 chance of collapsing in state |1>

Superposition transforms

Applying the hadamard gate will place the qubit in a superposition state. Multiplying a qubit with a hadamard matrix illustrates this.

The hadamard matrix:

$$\frac{1}{\sqrt{2}}\begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \text{ or } \begin{pmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} \end{pmatrix}$$

2 qubits
How is superposition visualised with 2 qubits

Define:
ψ = a|00> + b|01> + c|10> + d|11>

a|00>
qubit1 at **0** state, qubit2 at **0** state
a: The coefficient that the qubits will be in state 00
b|01>
qubit1 at **0** state, qubit2 at **1** state
b: The coefficient that the qubits will be in state 01
c|10>
qubit1 at **1** state, qubit2 at **0** state
c: The coefficient that the qubits will be in state 10
d|11>
qubit1 at **1** state, qubit2 at **1** state
d: The coefficient that the qubits will be in state 11

Required math

Unitary Matrix
A qubit can only be multiplied to a unitary matrix - A unitary matrix is a matrix that preserves length. Since a qubit can only have a magnitude of 1, then transforming it requires multiplying it by a unitary matrix. This ensures that its magnitude is 1.

Unitary matrix example
0 1
1 0

A unitary matrix is reversible. This means that if a vector is multiplied to a unitary matrix, the result can be multiplied to the same unitary matrix and you end up with the input.

To prove that matrix U is unitary, get the complex conjugate of U and multiply it by U. The result must be the identity matrix.

Given:
$U^{\dagger} U = I$
Where:
$U^{\dagger}$ is the complex conjugate of U

I is the identity matrix

Observing Phase

Placing qubits in superposition are associated with a phase as shown in the following examples:

Positive Phase(+)

The initial state of the qubit is |0>. When the hadamard operation is applied, its phase is (+)

$$|0\rangle - \boxed{H} = \frac{|0\rangle + |1\rangle}{\sqrt{2}}$$

Negative Phase(-)

The initial state of the qubit is |1>. When the hadamard operation is applied, its phase is (-)

$$|1\rangle - \boxed{H} = \frac{|0\rangle - |1\rangle}{\sqrt{2}}$$

Both qubits are in superposition where their only difference is the initial state that they were in prior to the Hadamard operation. The difference is reflected in its phase (+/-).

Quantum circuit

A quantum algorithms are synthesised into quantum circuits. Below is a 4 qubit circuit

```
start state        quantum gates                    end state
|0> -------------------------------------------------------|0>
|0> -------------------------------------------------------|1>
|0> -------------------------------------------------------|0>
|0> -------------------------------------------------------|1>
```

Start state is α|0000>, where α = 1

This is the initial state (computational basis state) of the quantum algorithm

Quantum gates
This is where the gates are inserted to perform quantum computation. cNOT, Hadamard and so on.

End state
This is the measured collapsed (classical bit) result of the quantum computation

Hadamard Gate
The hadamard gate applied to qubits to put them in a superposition state. The hadamard gate is represented by a matrix in the form of $\frac{1}{\sqrt{2}}\begin{bmatrix}1 & 1\\ 1 & -1\end{bmatrix}$.

The following example places a qubit in superposition by applying the hadamard gate . This can be represented by multiplying the qubit in vector form with the hadamard matrix

$$|0\rangle H = \left(\begin{bmatrix}1\\ 0\end{bmatrix}\right)\left(\frac{1}{\sqrt{2}}\begin{bmatrix}1 & 1\\ 1 & -1\end{bmatrix}\right) = \frac{1}{\sqrt{2}}\begin{bmatrix}1\\ 1\end{bmatrix}.$$

Superposition of n qubits
We can apply the hadamard gate to each qubit to place n qubits in superposition. A convenient way to represent this is through the following equation $|x\rangle^{\otimes n}H^{\otimes n}$
Where:

- x is the initial state of n qubits
- H is the hadamard matrix
- n is the number of qubits
- $|x\rangle^{\otimes n}$ is the tensor product $x \otimes x \ldots \otimes x$
- $H^{\otimes n}$ is the tensor product $H \otimes H \ldots \otimes H$

The following example places 2 qubits in superposition $|0\rangle^{\otimes 2}H^{\otimes 2}$

$$|0\rangle^{\otimes 2} = |0\rangle \otimes |0\rangle = \begin{bmatrix}1\\ 0\end{bmatrix} \otimes \begin{bmatrix}1\\ 0\end{bmatrix} = \begin{bmatrix}1\\ 0\\ 0\\ 0\end{bmatrix}$$

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

Quantum Algorithms
A quantum algorithm is a series of finite steps required to compute a specific task that inherently takes advantage of quantum mechanical effects such as superposition and entanglement. The quantum algorithm works by performing 2^n computations in parallel where 'n' is the number of qubits in the quantum processor.

The quantum computer applies the algorithm in the following steps:
- Setup the qubits in an initial states
- The QC device controls the qubits by applying quantum gates to the qubits. applying the quantum gates is what allows the QCD to perform computations on the qubits. At a physical level, applying the quantum gates requires the QCD to lower the temperature of the qubits to near absolute 0 degrees. At these low temperatures, microwaves are used to apply the quantum gates to the qubits and place them in desirable states including superposition
- When qubits are in superposition, they can perform multiple operations in parallel. This is the fundamental power of a quantum computer.
- Explain this further in Deutschs algorithm

Typical quantum algorithms
Shor's Algorithm - Searching for factors of large numbers

Grover's algorithm - Finding data in an unordered list

Deutsch's Algorithm
Consider a function whose domain is {0, 1} and range {0, 1}. This means that the following input and output are possible:

f(0) = 0
f(1) = 0
f(0) = 1
f(1) = 1

In other words:

If the input to the function is 0, then the output is 0. This is one function
If the input to the function is 1, then the output is 0. This is one function
If the input to the function is 0, then the output is 1. This is one function
If the input to the function is 1, then the output is 1. This is one function

Basically, there are 4 different functions satisfying the input and output criterias. The implementation of the oracle function is not significant to the algorithm. We are only interested in the inputs and outputs of the oracle function.

Balanced and Constant
A function is constant if the output is the same regardless of the input.

f(0) = f(1) = 0
or
f(0) = f(1) = 1

A function is balanced if either: all inputs match the output, or: all input do not match all outputs.
f(0) = 0 and f(1) = 1
or
f(0) = 1 and f(1) = 0

It helps to think of the classical implementation of the oracle function:

A constant function definition can be
```
int f(int n) {
        return 0;
}
```
or can be
```
int f(int n) {
        return 1;
}
```

A balanced function definition can be
```
int f(int n) {
        return not n
}
```
or can be
```
int f(int n) {
        return n;
}
```

The goal is to determine if the function is balanced or constant.

Classical execution

In order to determine if the function is balanced or constant, we need to evaluate the function twice. Evaluating f(x) once, say f(0) = 0 does not yet say if the function is balanced or constant. You need to evaluate f(1) to be able to determine this.

Given the programmatic statement:

$$iff(0) = 0andf(1) = 1$$

then function is balanced
else function is constant

Programatically, this can be viewed as the following operations:
```
x = 0            // Set input
```

a = f(x)		// Evaluate 1
x = 1		// Set input
b = f(x)		// Evaluate 2
c = a xor b	// Is constant or balanced

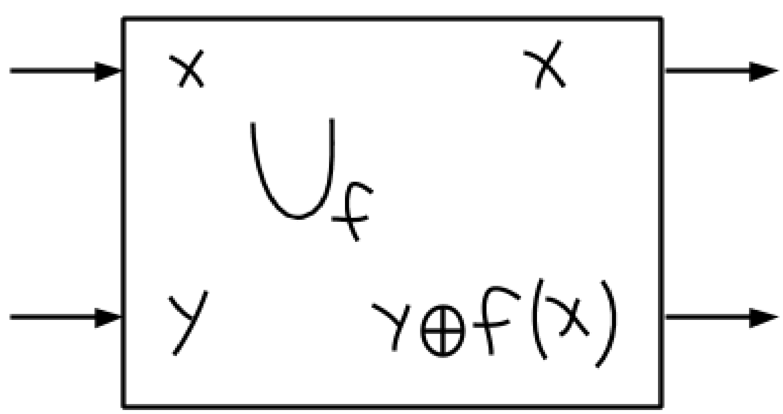
To convert the above into a quantum algorithm:
- Setting x = 0 and x = 1 can be converted to putting x in a superposition using a Hadamard transform
- f(x) can be evaluated in parallel as they are in superposition
- c = a xor b is evaluating the answer of the results of f(x).

As the oracle function f is execution in superposition, its result can also be queried while in superposition

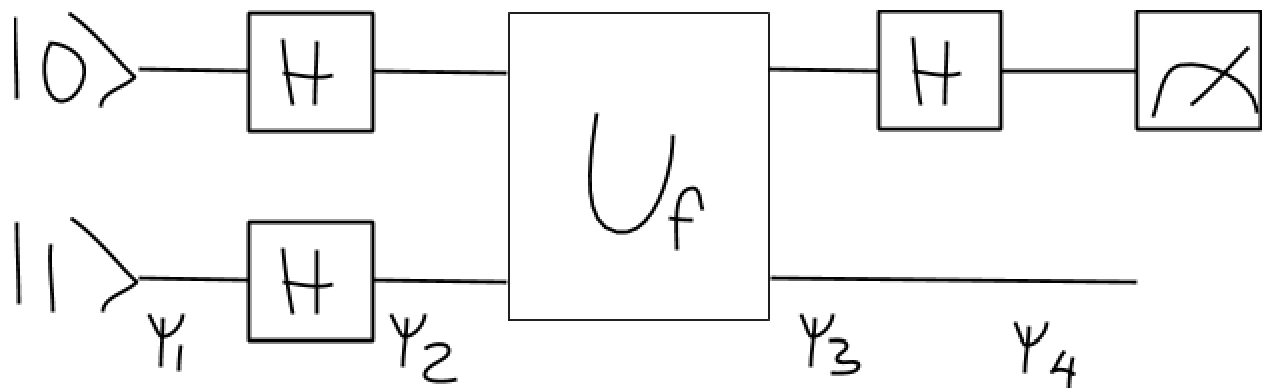
Quantum execution
For quantum execution, can we determine if function f is balanced or constant with only one execution step?

The goal is to arrive at the condition:
f(0) == f(1) -> f is constant
f(0) != f(1) -> f is balanced

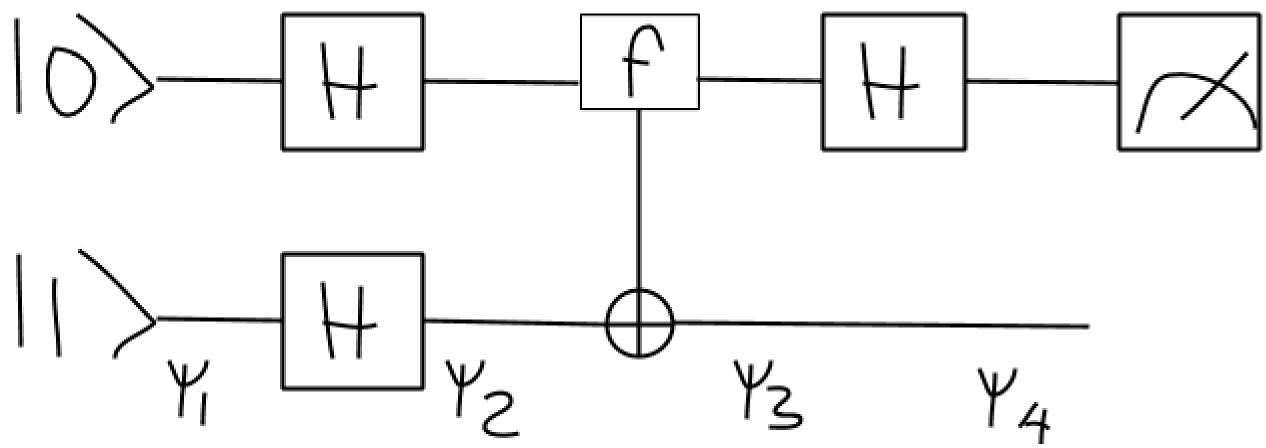
Function Evaluation
Deutschs algorithms requires evaluating a function using the top qubit ‘x’ as the parameter to the function: f(x). The bottom qubit is xor’d to the result of the calling f(x)



The quantum circuit for Deutsch’s Algorithm



The oracle function can be viewed as taking q0 as parameter and flipping q1 if q0=1. The actual body of the function is implementation specific.



NOTE: *The XOR operation will be used throughout the description of the algorithm. The XOR between q0 and q1 is similar to the CNOT gate. The CNOT gate works the by flipping the value of q1 only if q0=1.*

At the start of the computation $\psi 1$, top qubit in set to state $|0\rangle$ and the bottom qubit is set to state $|1\rangle$

$\psi 1 = |01\rangle$

At $\psi 2$, the Hadamard gate is applied to both qubits to put them in superposition.

$$\psi_2 = \left(\frac{|0\rangle + |1\rangle}{\sqrt{2}}\right)\left(\frac{|0\rangle - |1\rangle}{\sqrt{2}}\right)$$

At $\psi 3$, the qubits are passed to the unitary operations Uf. The unitary function takes in the top qubit as the parameter and the result is xor'd to the bottom qubit. To simplify, we first evaluate U_f with qubit q0 at $|x\rangle$ and the bottom qubit q1 in superposition which gives

$$\psi = \left(\frac{|x\rangle|0 \oplus f(x)\rangle - |x\rangle|1 \oplus f(x)\rangle}{\sqrt{2}}\right)$$

Xoring a number with 0 just gives back the number. While xoring a number with 1, gives the inverse or NOT of the number. So the term $0 \oplus f(x)$ just gives back $f(x)$ and $1 \oplus f(x)$ gives the inverse or the NOT of $f(x)$ which can be denoted as $|\overline{f(x)}\rangle$. The state of then becomes $\psi = |x\rangle \left(\frac{|f(x)\rangle - |\overline{f(x)}\rangle}{\sqrt{2}}\right)$

If $f(x) = 0$, then q2 evaluates to $|x\rangle \left(\frac{|0\rangle - |1\rangle}{\sqrt{2}}\right)$

If $f(x) = 1$, then q2 evaluates to $|x\rangle \left(\frac{|1\rangle - |0\rangle}{\sqrt{2}}\right)$

This generates a -1 phase factor $(-1)^{f(x)} \left(\frac{|0\rangle - |1\rangle}{\sqrt{2}}\right)$.

The the phase factor is generated to ensure that the 2nd qubit stays in the state $|-\rangle$.

It is important to note that the phase factor is general to the entire quantum system. It is however only applied to the first qubit to make any meaningful measurements. This is reflected by rearranging the term to $\psi = (-1)^{f(x)}|x\rangle|-\rangle$. Applying the -1 phase factor to the second qubit does not yield anything useful.

The complete state at $\psi 3$ becomes

$$\psi_3 = \frac{1}{\sqrt{2}}\left((-1)^{f(0)}|0\rangle\left(\frac{|0\rangle - |1\rangle}{\sqrt{2}}\right)\right) + \left((-1)^{f(1)}|1\rangle\left(\frac{|0\rangle - |1\rangle}{\sqrt{2}}\right)\right)$$

The term $\left(\frac{|0\rangle - |1\rangle}{\sqrt{2}}\right)$ can be replaced by $|-\rangle$, thereby simplifying the state to

$$\psi_3 = \frac{(-1)^{f(0)}|0\rangle|-\rangle + (-1)^{f(1)}|1\rangle\ |-\rangle}{\sqrt{2}}$$

Alternatively, this can also be written as $\psi_3 = \frac{(-1)^{f(0)}|0\rangle + (-1)^{f(1)}|1\rangle}{\sqrt{2}}|-\rangle$

The final step is applying a Hadamard gate to the top qubit and measuring its value at $\psi 4$. Let us try the different results of $f(0)$ and $f(1)$, apply the Hadamard gate and measure the final result. Remember that:

$(-1)^0 = 1$

$(-1)^1 = -1$

If $f(0) = f(1) = 0$ then the top qubit state evaluates to

$$\psi \;=\; \left(\frac{(-1)^0|0\rangle + (-1)^0|1\rangle}{\sqrt{2}}\right) \;=\; \left(\frac{(1)|0\rangle + (1)|1\rangle}{\sqrt{2}}\right) \;=\; \left(\frac{|0\rangle + |1\rangle}{\sqrt{2}}\right) \;=\; \begin{bmatrix} \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} \end{bmatrix}$$

Applying the Hadamard gate to the top qubit gives

$$\psi = \begin{bmatrix} \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} \end{bmatrix} \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \end{bmatrix} = |0\rangle$$

The final state becomes $\psi_4 = |0\rangle \otimes |-\rangle$. This means that the function f is constant.

If $f(0) = f(1) = 1$, then the top qubit state evaluates to

$$\psi = \left(\frac{(-1)^1|0\rangle + (-1)^1|1\rangle}{\sqrt{2}}\right) = \left(\frac{(-1)|0\rangle + (-1)|1\rangle}{\sqrt{2}}\right) = \left(\frac{-|0\rangle - |1\rangle}{\sqrt{2}}\right) = \begin{bmatrix} -\frac{1}{\sqrt{2}} \\ -\frac{1}{\sqrt{2}} \end{bmatrix}$$

The result of the function $f(x) = 1$ has caused a -1 phase change in the coefficients of the qubit. This still means that the qubit is in superposition but only with a -1 phase. i.e. it has changed orientation.

Applying the Hadamard gate to the top qubit gives

$$\psi = \begin{bmatrix} -\frac{1}{\sqrt{2}} \\ -\frac{1}{\sqrt{2}} \end{bmatrix} \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} = \begin{bmatrix} -1 \\ 0 \end{bmatrix} = |0\rangle$$

Recall that the -1 coefficient means a phase change in the top qubit. This implies the top qubit is still in the $|0\rangle$ state. The final state becomes $\psi_4 = |1\rangle \otimes |-\rangle$. This means that the function f is constant.

If $f(0) = 0$ and $f(1) = 1$ then the top qubit state evaluates to

$$\psi = \left(\frac{(-1)^0|0\rangle + (-1)^1|1\rangle}{\sqrt{2}}\right) = \left(\frac{(1)|0\rangle + (-1)|1\rangle}{\sqrt{2}}\right) = \left(\frac{|0\rangle - |1\rangle}{\sqrt{2}}\right) = \begin{bmatrix} \frac{1}{\sqrt{2}} \\ -\frac{1}{\sqrt{2}} \end{bmatrix}$$

Applying the Hadamard gate to the top qubit gives

$$\psi = \begin{bmatrix} \frac{1}{\sqrt{2}} \\ -\frac{1}{\sqrt{2}} \end{bmatrix} \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \end{bmatrix} = |1\rangle$$

The final state becomes $\psi_4 = |1\rangle \otimes |-\rangle$. This means that the function f is balanced.

If $f(0) = 1$ and $f(1) = 0$ then the top qubit state evaluates to

$$\psi = \left(\frac{(-1)^1|0\rangle + (-1)^0|1\rangle}{\sqrt{2}}\right) = \left(\frac{(-1)|0\rangle + (1)|1\rangle}{\sqrt{2}}\right) = \left(\frac{-|0\rangle + |1\rangle}{\sqrt{2}}\right) = \begin{bmatrix} -\frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} \end{bmatrix}$$

Applying the Hadamard gate to the top qubit gives

$$\psi = \begin{bmatrix} -\frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} \end{bmatrix} \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} = \begin{bmatrix} 0 \\ -1 \end{bmatrix} = |1\rangle$$

The final state becomes $\psi_4 = |1\rangle \otimes |-\rangle$. This means that function f is balanced.

Summary

- Constant f(0) = f(1) = 0

$$\psi 3 = \left(\frac{(-1)^0|0\rangle + (-1)^0|1\rangle}{\sqrt{2}}\right)|-\rangle = \left(\frac{|0\rangle + |1\rangle}{\sqrt{2}}\right)|-\rangle = |+\rangle|-\rangle$$

At $\psi 3$, the sum of the amplitude of the coefficients is 2. Where N is the number of input states

$$\sum_{x=0}^{2}(-1)^{f(x)} = 2$$

More generally

$$\sum_{x=0}^{N}(-1)^{f(x)} = N$$

- Constant f(0) = f(1) = 1

$$\psi 3 = \left(\frac{(-1)^1|0\rangle + (-1)^1|1\rangle}{\sqrt{2}}\right)|-\rangle = \left(\frac{-|0\rangle - |1\rangle}{\sqrt{2}}\right)|-\rangle = -|+\rangle|-\rangle$$

At $\psi 3$, the sum of the amplitude of the coefficients is 2. Where N is the number of input states

$$\sum_{x=0}^{2}(-1)^f(x) = -2$$

- Balanced f(0) = 0, f(1) = 1

$$\psi 3 = \left(\frac{(-1)^0|0\rangle + (-1)^1|1\rangle}{\sqrt{2}}\right)|-\rangle = \left(\frac{|0\rangle - |1\rangle}{\sqrt{2}}\right)|-\rangle = |-\rangle|-\rangle$$

At $\psi 3$, the sum of the amplitude of the coefficients is 0. Where N is the number of input states:

$$\sum_{x=0}^{2}(-1)^f(x) = 0$$

- Balanced $f(0) = 1, f(1) = 0$

$$\psi 3 = \left(\frac{(-1)^1|0\rangle + (-1)^0|1\rangle}{\sqrt{2}}\right)|-\rangle = \left(\frac{-|0\rangle + |1\rangle}{\sqrt{2}}\right)|-\rangle = -|-\rangle|-\rangle$$

At $\psi 3$, the sum of the amplitude of each of the coefficients is 2. Where N is the number of input states

$$\sum_{x=0}^{2}(-1)^f(x) = 0$$

Deutsch-Jozsa Algorithm

The Deutsch Jozsa algorithm is a generalisation of the Deutsch's algorithm. The goal is still to determine if a function is balanced or constant. The difference is that there can be more than 1 input to the oracle function. Note that the inputs to the function is guaranteed to be constant or balanced. To classically determine if the function is constant or balanced, we would need N/2 calls to the function for the best case and N/2 + 1 calls for the worst case. N is the total number of parameters or qubits of the function. To illustrate, if N = 4:

Given 3 calls where the function is constant

call 1: f(00) = 0
call 2: f(00) = 0
call 3: f(00) = 0

The function is determined to be constant after N/2 + 1 or 3 calls. Calling f for the 4th time is redundant and calling f only twice is not enough because there is still the possibility that the 3rd call is 1, in which case the function is actually balanced.

Given 2 calls where the function is balanced

call 1: f(00) = 0
call 2: f(01) = 1

Since there are only 2 possible calls left and the function is guaranteed to be constant or balanced then the function is already determined to be balanced after only N/2 calls.

General form

The following describes the general form of each state in the Deutsch-Jozsa algorithm.

Setup all input qubits to 0 and the output qubit to 1

$$\psi_1 = |0...n\rangle|1\rangle$$

Place all qubits in superposition.

$$\psi_2 = \frac{1}{\sqrt{2^n}} \sum_{x=0}^{2^{n-1}} |x\rangle|-\rangle$$

Where:

n = input qubits

x = bit string representation of qubits in superposition e.g.

if n = 2 then 0 = 00, 1 = 01, 2 = 01, 3 = 11

if n = 3 then 0 = 000, 1 = 001, 2 = 010, 3 = 011, 4 = 100 and so on

Apply the oracle function Uf

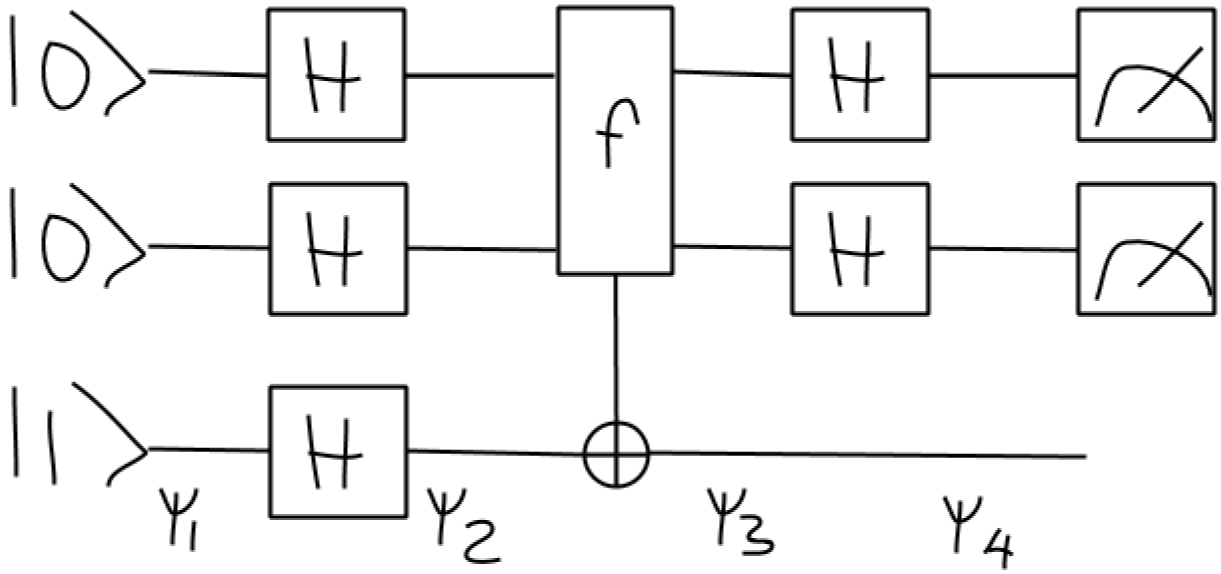
$$\psi_3 = \frac{1}{\sqrt{2^n}} \sum_{x=0}^{2^{n-1}} (-1)^{f(x)} |x\rangle|-\rangle$$

Multiply each n input qubits of the state ψ_3 to a hadamard gate. In other words, multiply the result of the input qubits in superposition at ψ_3 by the tensor product of n hadamard gates $H\otimes^n$.

$$\psi_4 = (H^{\otimes n}) \left(\frac{1}{\sqrt{2^n}} \sum_{x=0}^{2^{n-1}} (-1)^{f(x)} |x\rangle|-\rangle \right)$$

2 input qubits

Quantum circuit



$$\psi_1 = |001\rangle$$

Apply the hadamard gates to all qubits. This

$$\psi_2 = \left(\frac{|0\rangle + |1\rangle}{\sqrt{2}}\right)\left(\frac{|0\rangle + |1\rangle}{\sqrt{2}}\right)\left(\frac{|0\rangle - |1\rangle}{\sqrt{2}}\right)$$

$$\psi_2 = \frac{|00\rangle|-\rangle + |01\rangle|-\rangle + |10\rangle|-\rangle + |11\rangle|-\rangle}{2}$$

$$\psi_2 = \frac{|00\rangle + |01\rangle + |10\rangle + |11\rangle}{2}|-\rangle$$

Applying the oracle function at $\psi 3$

$$\psi 3 = \frac{|00\rangle(|0 \oplus f(00)\rangle - |1 \oplus f(00)\rangle) + |01\rangle(|0 \oplus f(01)\rangle - |1 \oplus f(01)\rangle) + |10\rangle(|0 \oplus f(10)\rangle - |1 \oplus f(10)\rangle) + |11\rangle(|0 \oplus f(11)\rangle - |1 \oplus f(11)\rangle)}{2}$$

$$\psi 3 = \frac{|00\rangle(|f(00)\rangle - |\overline{f(00)}\rangle) + |01\rangle(|f(01)\rangle - |\overline{f(01)}\rangle) + |10\rangle(|f(10)\rangle - |\overline{f(10)}\rangle) + |11\rangle(|f(11)\rangle - |\overline{f(11)}\rangle)}{2}$$

Example 1

Given

$$f(00) = f(01) = f(10) = f(11) = 0$$

Then

$$\psi 3 = \frac{|00\rangle + |01\rangle + |10\rangle + |11\rangle}{2} |-\rangle$$

At this point, we're only interested in the input qubits so the output qubit is ignored

$$\psi 3 = \frac{|00\rangle + |01\rangle + |10\rangle + |11\rangle}{2}$$

or

$$\psi 3 = \frac{1}{2}\begin{bmatrix} 1|00\rangle \\ 1|01\rangle \\ 1|10\rangle \\ 1|11\rangle \end{bmatrix} = \frac{1}{2}\begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \end{bmatrix}$$

We measure the result of the input qubits by multiplying it with the tensor product of the hadamard gates $H^{\otimes 2}$

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

Multiply the input qubits in vector form by the tensor product matrix

$$\psi 4 = \frac{1}{\sqrt{2}}\begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \end{bmatrix} \frac{1}{\sqrt{2}}\begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \end{bmatrix} = \begin{bmatrix} 0.25 + 0.25 + 0.25 + 0.25 \\ 0.25 - 0.25 + 0.25 - 0.25 \\ 0.25 + 0.25 - 0.25 - 0.25 \\ 0.25 - 0.25 - 0.25 + 0.25 \end{bmatrix}$$

$$\psi 4=\begin{bmatrix}1\\0\\0\\0\end{bmatrix}=\begin{bmatrix}1|00\rangle\\0|01\rangle\\0|10\rangle\\0|11\rangle\end{bmatrix}$$

$$\psi 4=|00\rangle$$

The final measurement of the input qubits yield $|00\rangle$. All qubits are 0 so the function is constant.

Example 2

Given

$$f(00)=f(01)=f(10)=f(11)=1$$

Then

$$\psi 3=\frac{|00\rangle-|01\rangle-|10\rangle-|11\rangle}{2}=\frac{1}{2}\begin{bmatrix}-1\\-1\\-1\\-1\end{bmatrix}$$

Multiply the input qubits in vector form by the tensor product matrix

$$\psi 4=\frac{1}{\sqrt{2}}\begin{bmatrix}-1\\-1\\-1\\-1\end{bmatrix}\frac{1}{\sqrt{2}}\begin{bmatrix}1&1&1&1\\1&-1&1&-1\\1&1&-1&-1\\1&-1&-1&1\end{bmatrix}=\begin{bmatrix}-0.25-0.25-0.25-0.25\\-0.25+0.25-0.25+0.25\\-0.25-0.25+0.25+0.25\\-0.25+0.25+0.25-0.25\end{bmatrix}$$

$$\psi 4=\begin{bmatrix}-1\\0\\0\\0\end{bmatrix}=\begin{bmatrix}-1|00\rangle\\0|01\rangle\\0|10\rangle\\0|11\rangle\end{bmatrix}$$

$$\psi 4=-|00\rangle$$

All qubits are 0 regardless of phase +/- so the function is constant.

Example 3

Given

$$f(00) = f(01) = 0, f(10) = f(11) = 1$$

Then

$$\psi 3 = \frac{|00\rangle + |01\rangle - |10\rangle - |11\rangle}{2} = \frac{1}{2}\begin{bmatrix} 1 \\ 1 \\ -1 \\ -1 \end{bmatrix}$$

$$\psi 4 = \frac{1}{\sqrt{2}}\begin{bmatrix} 1 \\ 1 \\ -1 \\ -1 \end{bmatrix} \frac{1}{\sqrt{2}}\begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \end{bmatrix}$$

$$\psi 4 = |01\rangle$$

If at least 1 qubit is 1, then the function is balanced. The 2nd qubit is 1 so the function is balanced.

Example 4

Given

$$f(00) = f(01) = 1, f(10) = f(11) = 0$$

Then

$$\psi 3 = \frac{-|00\rangle - |01\rangle + |10\rangle + |11\rangle}{2} = \frac{1}{2}\begin{bmatrix} -1 \\ -1 \\ 1 \\ 1 \end{bmatrix}$$

$$\psi 4 = \frac{1}{\sqrt{2}}\begin{bmatrix} -1 \\ -1 \\ 1 \\ 1 \end{bmatrix} \frac{1}{\sqrt{2}}\begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ -1 \\ 0 \end{bmatrix}$$

$$\psi 4 = -|01\rangle$$

If at least 1 qubit is 1 regardless of phase +/-, then the function is balanced. The 2nd qubit is 1 so the function is balanced.

Concluding statements

Deutsch's Algorithm determines if a function is constant if c or $-|+\rangle$

Deutsch's Algorithm determines if a function is balanced if $\psi 3 = |-\rangle$ or $-|-\rangle$

Deutsch-Jozsa Algorithm determines if a function is constant if all qubits at $\psi 4$ are $|0\rangle$ regardless of phase (+/-)

Deutsch-Jozsa Algorithm determines if a function is balanced if at least 1 qubit at $\psi 4$ is $|1\rangle$ regardless of phase (+/-)

Grover Algorithm

Grover's search algorithm involves searching for a particular result of an oracle function given n inputs. The domain of the function is $x \in \{0, 1\}^n$ or 2^n possible inputs. The range is $y \in \{0, 1\}$. In a classical algorithm, in order to find the function that returns 1, we would need to call a function n times in the worst case. In C, the oracle function f and the search algorithm can be implemented as follows:

```
size_t f(size_t x) {
      if (x == 10) {
            return 1;
      }
      return 0;
}
bool groversSearch(size_t n) {
      for (size_t x = 0; x < n; ++n) {
            if (f(x) == 1) {
                  printf("Function found at x = %d: \n", x);
                  return true;
            }
      }
      printf("Function not found. \n", x);
      return false;
}
```

Steps

$\psi_1 = |0\rangle^{\otimes n}|1\rangle = |0...0\rangle|1\rangle$

$$\psi_2 = (|0\rangle^{\otimes n} H^{\otimes n}) H|1\rangle = \sum_{x=0}^{2^{n-1}} |x\rangle|-\rangle$$

Iteration start - From here, apply $\sqrt{N}$ times

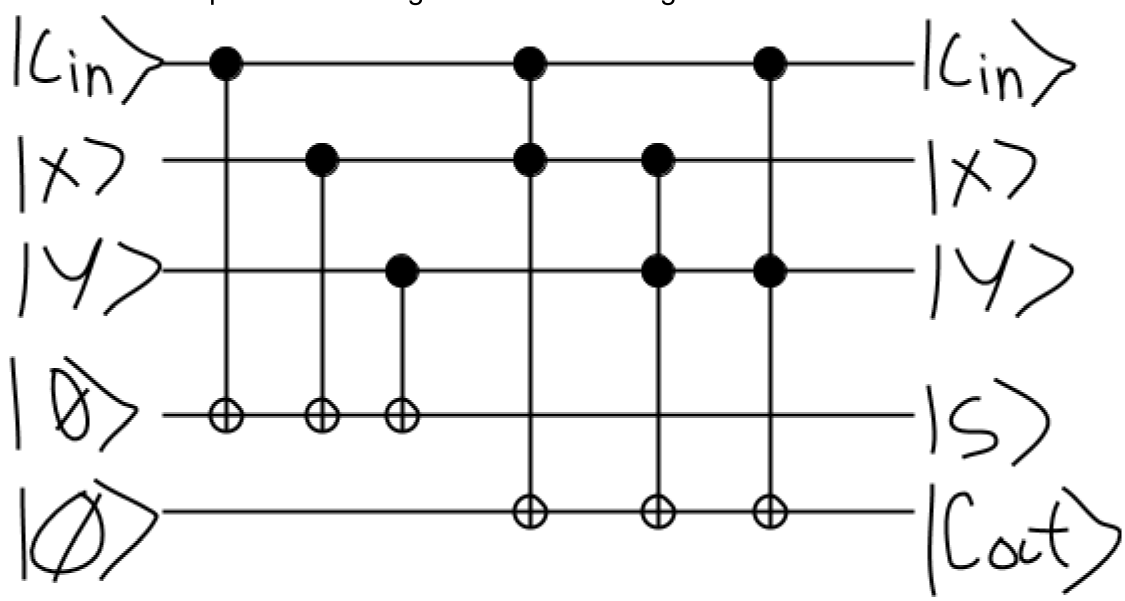
Apply phase inversion

$$\psi_3 = \frac{1}{\sqrt{2^n}} \sum_{x=0}^{2^{n-1}} (-1)^{f(x)} |x\rangle|-\rangle$$

Apply Diffusion Operation
Todo

Simon's Algorithm
Find the unknown period in a periodic function.
Given: $f(x) = f(x \oplus c)$
Where: c is the unknown period

Classical computation on quantum circuits
It is certainly possible to perform classical computations using quantum circuits. The following is a 1-bit adder implemented using a series of cNOT gates $|0\rangle$



Quantum Decoherence

A quantum circuit needs to perform its calculation before decoherence occurs. This means that there is a relative amount of time that the quantum computer must perform its calculation before random environmental observations destroy the superposition states of an ongoing calculation.

Quantum DesignScript

```
// Example 1
// Define 4 qubit registers
qubit qr[4];

// Setup the initial qubit states
reset(qr[0], 0);
reset(qr[1], 0);
reset(qr[2], 1);
reset(qr[3], 0);

update();

hadamard(qr[0]);
hadamard(qr[2]);

update();

cNOT(qr[0], qr[1]);
cNOT(qr[2], qr[3]);

update();

measure(qr[0]);
measure(qr[1]);
measure(qr[2]);
measure(qr[3]);



// Example 2
// Define 2 qubit registers
qubit qr[2];

// Setup the initial qubit states
reset(qr[0], 0);
```

```
reset(qr[1], 0);

update();

hadamard(qr[0]);

update();

cNOT(qr[0], qr[1]);

update();

// Restore qubit1 state
hadamard(qr[0]);

update();

measure(qr[0]);
measure(qr[1]);
```

```
// Example 3
// Define 2 qubit registers
qubit qr[2];

// Setup the initial qubit states
reset(qr[0], 0);
```

```
reset(qr[1], 0);

update();

hadamard(qr[0]);

update();

Uf(qr[0], qr[1]);

update();

// Restore qubit1 state
hadamard(qr[0]);

update();

measure(qr[0]);
measure(qr[1]);
```