The Deutsch Algorithm – Worksheet 2

Deutsch's algorithm, developed by David Deutsch in 1985, marks a significant milestone in the world of computing. It's the first instance where a quantum algorithm showed a potential advantage over classical algorithms. This discovery opened the door to the idea that computers based on the principles of quantum mechanics could solve certain types of problems more efficiently than classical computers.

In this lesson, you will learn how Deutsch's algorithm works. To fully grasp its significance, you'll first need to understand how the problem the algorithm solves can be approached using classical computing methods.

The Problem

Imagine we have a mystery box represented by a function f. This box is special because when you give it a number (either 0 or 1), it gives you back a number (also either 0 or 1). But there's a secret about how our box works, and we want to find out what it is. The box works in one of two ways:

Constant functions		Balanced functions	
$f_1(0) = 0$ $f_1(1) = 0$	$f_2(0) = 1$ $f_2(1) = 1$	$f_3(0) = 0$ $f_3(1) = 1$	$f_4(0) = 1$ $f_4(1) = 0$

Our challenge is to figure out if our mystery box (function $f(x)$) is *constant* or *balanced* by asking it questions, that is, by giving it numbers (0 or 1) and seeing what it returns.

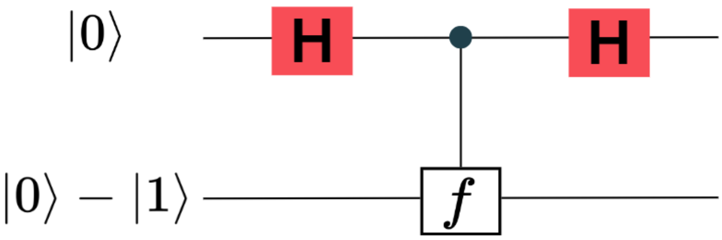
Classical Solution

1. Describe the main difference between a *constant function* and a *balanced function*.

2. Determine the minimum number of questions you need to ask the box to find out whether the function is *constant* or *balanced*. Justify your answer.

Quantum Solution: The Deutsch Algorithm

The Deutsch algorithm tackles our problem using just one query, thanks to the power of quantum computing. It employs the following quantum circuit



that operates in three main steps, utilizing quantum bits (qubits) and quantum gates. Here's how it's set up:

For the first qubit (in our first register), we start with the state $|\mathbf{0}\rangle$**.** For the second register, we use the special combination of states $|\mathbf{0}\rangle - |\mathbf{1}\rangle$. The algorithm unfolds in three key steps:

Step I: Apply a Hadamard gate to $|\mathbf{0}\rangle$ to create an equal superposition of $|0\rangle$ and $|1\rangle$.

Step II: Apply the function f, which modifies the phase of the first register according to $(-1)^{f(x)}$. This step is crucial for quantum interference. If you would like to explore where this phase factor comes from in more detail, see the optional worksheet quantum function evaluation**.**

Step III: Apply a Hadamard gate again to the first register. This step interferes phase-modified states, allowing measurement to determine whether $f(x)$ is constant or balanced.

Your Task: Understanding Step by Step

Now, you will delve into each step of the algorithm to understand how quantum principles enable solving the problem with a single query.

3. Step I: Explain why the input state $|\mathbf{0}\rangle$ transforms into the state $|\mathbf{0}\rangle + |\mathbf{1}\rangle$ after applying the first Hadamard gate.

Hint: The Hadamard operator can be expressed as $H = \frac{1}{\sqrt{2}}\left(|0\rangle\langle 0| + |0\rangle\langle 1| + |1\rangle\langle 0| - |1\rangle\langle 1|\right)$ *. Note that the result is considered up to an irrelevant overall factor.*

4. Step II: In this step, by using quantum function evaluation to introduce phase shifts in the first register, it can be shown (see *WS on quantum function evaluation*) that the state $|\mathbf{0}\rangle + |\mathbf{1}\rangle$ transforms into

$$(-\mathbf{1})^{f(\mathbf{0})}|\mathbf{0}\rangle + (-\mathbf{1})^{f(\mathbf{1})}\,|\mathbf{1}\rangle\,.$$

a) Demonstrate that for a *constant* function f, the state $(-\mathbf{1})^{f(\mathbf{0})}|\mathbf{0}\rangle + (-\mathbf{1})^{f(\mathbf{1})}\,|\mathbf{1}\rangle$ simplifies to $|\mathbf{0}\rangle + \,|\mathbf{1}\rangle$, up to an irrelevant overall factor.

b) Demonstrate that for a *balanced* function f, the state $(-\mathbf{1})^{f(\mathbf{0})}|\mathbf{0}\rangle + (-\mathbf{1})^{f(\mathbf{1})}\,|\mathbf{1}\rangle$ simplifies to $|\mathbf{0}\rangle - |\mathbf{1}\rangle$, up to an irrelevant overall factor.

5. Step III: In this step, you will apply the Hadamard gate to the states obtained in Task 4.

a) If the function is constant, apply the Hadamard gate to the state $|\mathbf{0}\rangle + |\mathbf{1}\rangle$ and show that you obtain the state $|\mathbf{0}\rangle$.

d) If the function is balanced, apply the Hadamard gate to the state $|\mathbf{0}\rangle - |\mathbf{1}\rangle$ and show that you obtain the state $|\mathbf{1}\rangle$.

6. Measurement

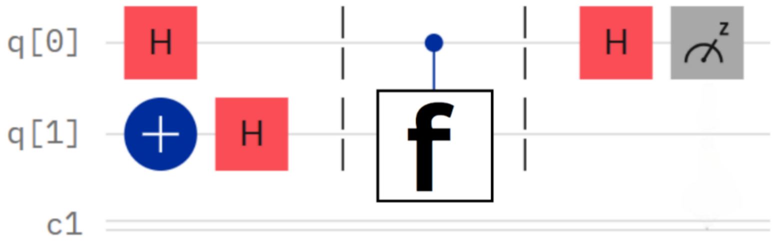
a) Which state do you measure after Step III if the function is constant? ☐ $|0\rangle$ ☐ $|1\rangle$

b) Which state do you measure after Step III if the function is balanced? ☐ $|0\rangle$ ☐ $|1\rangle$

c) Explain how your answers in a) and b) effectively solve the problem.

Implementation with the Quantum Composer

The Deutsch Algorithm can be easily implemented using the Quantum Composer with the following circuit:

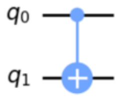


In the following tasks, you will model and test the algorithm with the quantum composer.

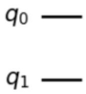
7. First, you need to model the constant and balanced functions using the quantum composer. These functions can be represented using specific configurations of quantum gates.

Consider q_0 as the input qubit (the x-value of $f(x)$) and observe how the value of the second register changes after applying the gates specified. The final state of the second register represents the output of $f(x)$.

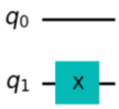
Below are the gate configurations for the functions f_1, f_2, f_3 and f_4. Each configuration corresponds to one of these functions:



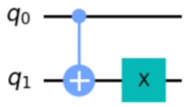
a)



b)



c)



d)

Match each of the functions f_1, f_2, f_3 and f_4 with the correct circuit.

8. Explain why the following part of the circuit

results in the state $|\mathbf{0}\rangle - |\mathbf{1}\rangle$.

9. Now, you're ready to experiment with the quantum composer. Choose one of the representations of the functions f_1, f_2, f_3 or f_4 , implement the circuit in the quantum composer, and verify if the algorithm works as expected.

Summary of quantum results

By using superposition, the quantum program can determine whether the function is constant or balanced with just one evaluation of the function. A classical deterministic method needs two evaluations, because it has to check both input cases separately.

What makes this especially remarkable is that distinguishing between "constant" and "balanced" is a global property of the function. That means: you cannot recognize it by looking at a single input-output pair. Only by comparing both values, f(0) and f(1), can you decide. A classical algorithm must therefore query both values separately.

In contrast, a quantum algorithm like the Deutsch algorithm uses superposition and interference to evaluate both inputs simultaneously. This allows it to analyze the global structure of the function in a single computational step. The measurement process then yields exactly one bit of information, which directly reveals the sought-after global property.

The Deutsch algorithm also illustrates two important points:

- Quantum computers are not superior in every respect.
  The quantum algorithm only decides whether a function is constant or balanced—it does not identify the exact function in the black box. Whether a given problem is suitable for quantum computing must be analyzed on a case-by-case basis.

- The solution lies in the input qubit.
  In the end, it is not the output qubit but the input qubit that is measured. During the algorithm, the relevant information is “moved” into this qubit, while the other part of the system loses information. Such information shifts are typical in many quantum algorithms. Often, the solution lies in places we would not intuitively expect.

Although the Deutsch problem is a rather simple and theoretical construct, it was the first to demonstrate fundamental principles of quantum computing. It paved the way for more powerful quantum algorithms that will be explored in the following lessons.