

## AN OVERVIEW AND EXPLANATION OF THE DEUTSCH-JOZSA QUANTUM ALGORITHM

**Bhairaba Kumar Majhi**

Department of Mathematics

Centurion University of Technology and Management, Bolangir, Odisha, India

**Satyabrata Sadangi**

Department of Mathematics

Centurion University of Technology and Management, Bolangir, Odisha, India

### ABSTRACT

*In this article we discussed a thorough guide to the Deutsch-Jozsa Algorithm in practice, including a overview and explanation of the mathematics, a complete Circuit, and an example implementation in Code using IBM's Qiskit to build and run on Quantum Computers. we tried covering everything from the logic of the Problem, Classical & Quantum solutions, to the Qiskit implementation*

*Keywords: Deutsch-Jozsa problem, Oracles, classical solutions, Quantum Advantage and speed-ups, superposition, Phase Kickback.*

### INTRODUCTION

#### THE DEUTSCH-JOZSA PROBLEM

Imagine the problem like a game show. You're a contestant, and the host lays out the rules of the game: "Here is a list of numbers. You can choose any of them, and hand it to me. In return, I'll tell you '0' or '1'.

"I may have sorted these numbers into two balanced halves, and assigned each half to '0' or '1'. Depending on what you tell me, I would tell you '0' or '1' based on which half your number is in. Or, I might not. Instead, I could always say '0', or always say '1'. You don't know."

"Now give me a number, and I'll tell you an answer. To win, all you have to do is tell me how I got that '0' or '1'. Did I split the list of numbers? Or did I say '0' or '1' the whole time? Who knows?"

#### THE CLASSICAL ALGORITHM

**Step One** - Here we will model this problematic situation into an algorithm.

The input component is ones question — the number you give the host. We can call this a '**Query**'. Your query is a number from a list, or an *input domain* — essentially a list of all possible queries. The host's job is to perform *some* unknown function on this query, and give you either 0 or 1. Exactly what that function is, we simply do not know — only that it will always be zero, always be one, or be balanced halves. Unknown functions like these are referred to as 'black boxes', and in the context of a problem like Deutsch-Jozsa, we'll call them an '**Oracle**'. You can give input, and receive output, but *cannot* see the inner workings of the Oracle. The output component is the host's answer of '0' or '1'. We'll just call this '**Output**'.

This is what the algorithm should look like. Pass a query to the Oracle, and receive an output



The Classical Algorithm has one input and output.

#### THE CLASSICAL SOLUTION

As you can imagine, this problem could be solved fairly well for a lucky query. If the Oracle returns '0', but then '1' for your next question, you *instantly* know that it's returning a split list, balanced between 0 and 1. He's not always saying 0 or 1, after all.