EURO

Fundamental Quantum Computing Algorithms and Their Implementation in Qiskit

Michal Belina

IT4Innovations
VSB - Technical University of Ostrava

18. - 19.3. 2025

Aims of the training

1. Familiarize you with the quantum computing
2. Show the possible advantage of quantum computers on specific tasks

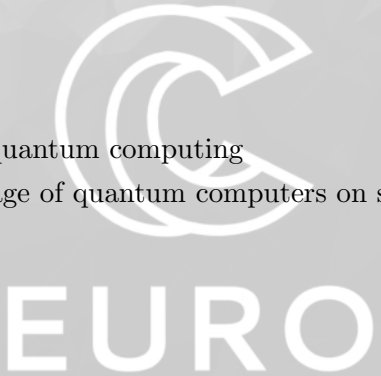


Table of Contents

Timetable of day 1

9:00-10:30	Introduction to Quantum Computers and Quantum Computing
10:30-10:45	Break
10:45-12:00	Quantum entanglement
12:00-13:00	Lunch Break
13:00-14:00	Quantum teleportation
14:00-15:00	Bernstein-Vazirani algorithm
15:00-15:15	Break
15:15-16:15	Simon's algorithm
16:15	Q&A and Closing the day

Questions welcome!!!

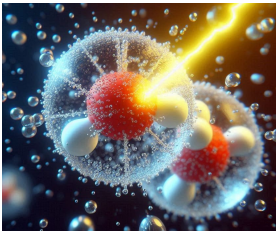
Michal Belina

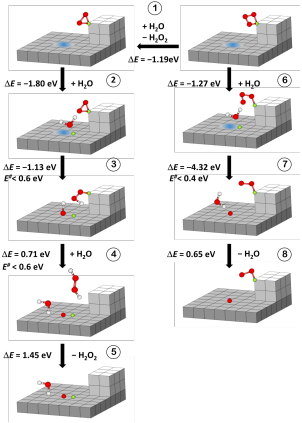
University of Chemistry and Technology Prague (2014-2025):

- Physical chemistry
- Computational chemistry

VŠB-Technical University Ostrava (2024 - present):

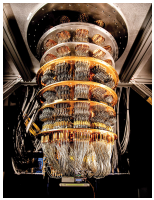
- Quantum Computing Lab
-





IT4Innovations

- Institute of VSB – Technical University of Ostrava
- Supercomputing and research center
- The most powerful supercomputing systems in the Czech Republic.
- Five research labs
- High-Performance Computing, Data Analysis, Quantum Computing, and Artificial Intelligence



First task

Please, introduce yourself.

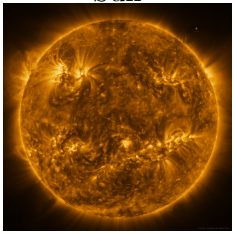
Preferably in chat:
Something like name, institute, field of main expertise and knowledge of quantum mechanics

Questions and even comments welcome at any time and in any way!

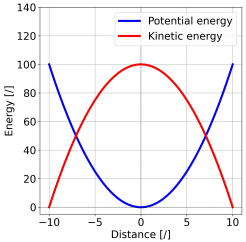
Table of Contents

Classical Mechanics

Sun



Chevrolet Impala 1967

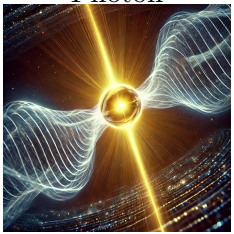


What would happen with the decreasing size?

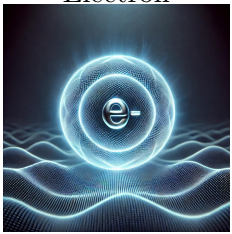
- Weght: kg $\rightarrow 10^{-31} \times$ kg
- Size: m $\rightarrow 10^{-10} \times$ m

Quantum Mechanics

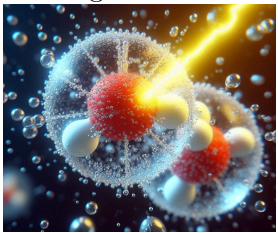
Photon

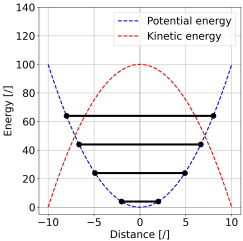


Electron



Light atoms

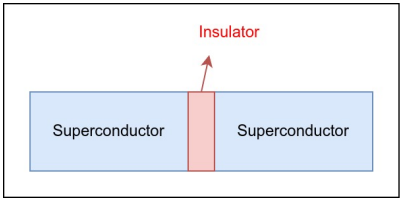


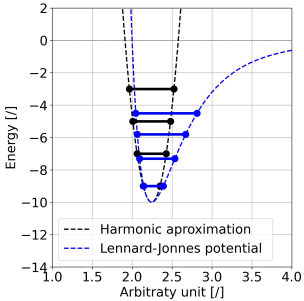


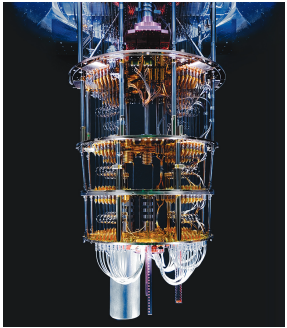
- Stern-Gerlach experiment
- Black-body radiation
- Zeeman effect

Basic idea: To facilitate the discrete energetic levels

Superconducting technology - Josephson junction

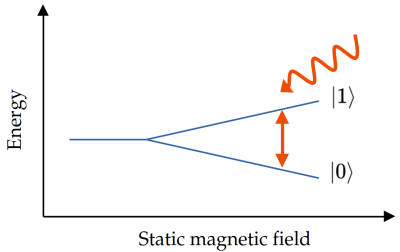


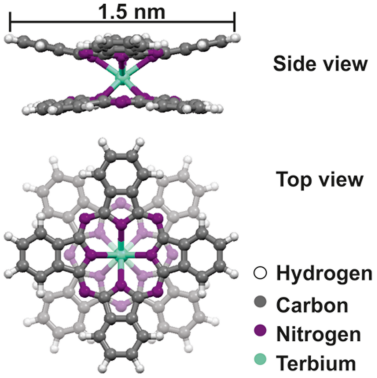




QC based on molecular spin

- Single-ion Magnets
- f-block ion in the middle – 1 “unpaired electron”
- Highly adjustable ligands

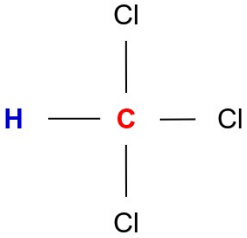


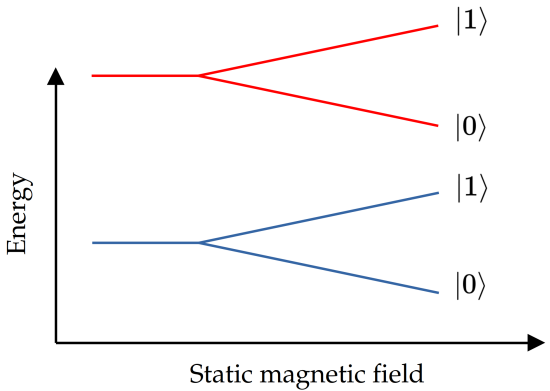


https://pubs.acs.org/doi/pdf/10.1021/acs.jpcc.6b03676

QC based on NMR

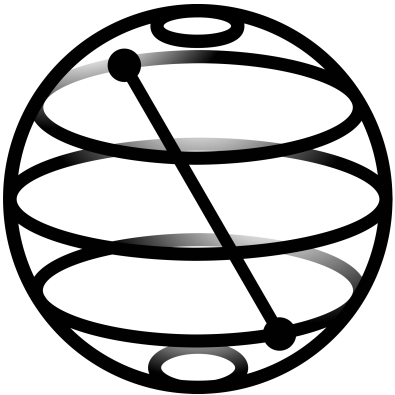
- Atoms with odd number of protons and/or neutrons.
- ^{1}H, ^{13}C...^{19}F,^{31}P





Qiskit

- Qiskit = Quantum Information Software Kit
- open-source software development kit (SDK) for working with quantum Computers



Quantum bit (qubit) is a two-dimensional quantum mechanical system that is in the state:

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

where α and β are complex numbers (the amplitudes of the quantum states $|0\rangle$ and $|1\rangle$, respectively), and they satisfy the normalization condition:

$$|\alpha|^2 + |\beta|^2 = 1 \tag{2}$$

This definition uses the **standard bra-ket notation**, where quantum states are represented as vectors:

$$|0\rangle = \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \quad |1\rangle = \begin{bmatrix} 0 \\ 1 \end{bmatrix} \tag{3}$$

Quantum Registers

- Each bit configuration in quantum superposition is represented as the tensor product of individual qubits.
- For example, the state $|10\rangle$, which represents the number 2 in a bit string, is written as:

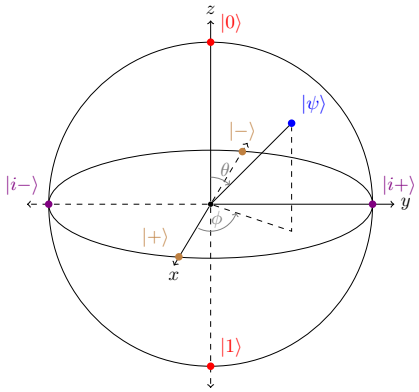
$$|10\rangle = |1\rangle \otimes |0\rangle = \begin{bmatrix} 0 \\ 1 \end{bmatrix} \otimes \begin{bmatrix} 1 \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \end{bmatrix} \tag{4}$$

Bloch sphere

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

$$\alpha = cos(\theta/2)$$
$$\beta = e^{i\phi}sin(\theta/2)$$

$$P(|0\rangle) = |\alpha| = cos^2(\theta/2)$$
$$P(|1\rangle) = |\beta| = |e^{i\phi}sin(\theta/2)|^2 = 1sin^2(\theta/2)$$
$$P(|0\rangle) + P(|1\rangle) = cos^2(\theta/2) + sin^2(\theta/2) = 1$$



Bloch sphere

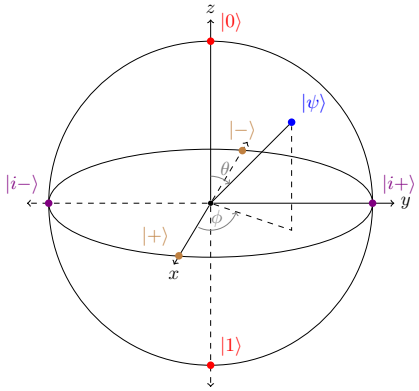
Other interesting point on Bloch sphere:

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

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

$$|i+\rangle = \frac{1}{\sqrt{2}}(|0\rangle + i\,|1\rangle)$$

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



One-Qubit Quantum Gates

Definition: A one-qubit gate is represented by a 2×2 unitary matrix U acting on a single qubit:

$$U|\psi\rangle = U \begin{bmatrix} \alpha \\ \beta \end{bmatrix}$$

Important One-Qubit Gates:

- **Pauli Gates:**

$$X = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}, \quad Y = \begin{bmatrix} 0 & -i \\ i & 0 \end{bmatrix}, \quad Z = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}$$

- **Hadamard Gate:**

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

- **Phase Gates:**

$$S = \begin{bmatrix} 1 & 0 \\ 0 & i \end{bmatrix}, \quad T = \begin{bmatrix} 1 & 0 \\ 0 & e^{i\pi/4} \end{bmatrix}$$

Actions of One-Qubit Quantum Gates

Example:

- Pauli-X (NOT gate): $X|0\rangle = |1\rangle\,, \quad X|1\rangle = |0\rangle$
- Hadamard on $|0\rangle$:

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

- Hadamard on $|1\rangle$:

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

- Z-gate on superposition state:

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

Actions of One-Qubit Quantum Gates II

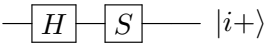
Initial State: The superposition state

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

Action of the S Gate:

$$S\,|+\rangle = \frac{1}{\sqrt{2}}(|0\rangle + i\,|1\rangle) = |i+\rangle$$

Quantum Circuit:



The S gate introduces a phase factor i to $|1\rangle$, transforming $|+\rangle$ into $|i+\rangle$, an eigenstate of the Y operator.

Phase Transformations

$$P(\lambda)\left|\psi\right\rangle = \begin{bmatrix}1 & 0\\ 0 & e^{i\lambda}\end{bmatrix}\begin{bmatrix}\alpha\\ \beta\end{bmatrix} = \begin{bmatrix}\alpha\\ e^{i\lambda}\beta\end{bmatrix}$$

$$Z\left|\psi\right\rangle = \begin{bmatrix}1 & 0\\ 0 & e^{i\pi}\end{bmatrix}\begin{bmatrix}\alpha\\ \beta\end{bmatrix} = \begin{bmatrix}1 & 0\\ 0 & -1\end{bmatrix}\begin{bmatrix}\alpha\\ \beta\end{bmatrix} = \begin{bmatrix}\alpha\\ -\beta\end{bmatrix}$$

$$S\left|\psi\right\rangle = \begin{bmatrix}1 & 0\\ 0 & e^{i\frac{\pi}{2}}\end{bmatrix}\begin{bmatrix}\alpha\\ \beta\end{bmatrix} = \begin{bmatrix}1 & 0\\ 0 & i\end{bmatrix}\begin{bmatrix}\alpha\\ \beta\end{bmatrix} = \begin{bmatrix}\alpha\\ i\beta\end{bmatrix}$$

$$T\left|\psi\right\rangle = \begin{bmatrix}1 & 0\\ 0 & e^{i\frac{\pi}{4}}\end{bmatrix}\begin{bmatrix}\alpha\\ \beta\end{bmatrix} = \begin{bmatrix}\alpha\\ e^{i\frac{\pi}{4}}\beta\end{bmatrix} = \begin{bmatrix}\alpha\\ \frac{1}{\sqrt{2}}(1+i)\beta\end{bmatrix}$$

Action of Gates on Special States:

$$Z\left|+\right\rangle = \left|-\right\rangle, \quad Z\left|-\right\rangle = \left|+\right\rangle, \quad S\left|+\right\rangle = \left|i+\right\rangle$$

$$Z\left|i-\right\rangle = SS\left|i-\right\rangle = TTTT\left|i-\right\rangle = \left|i+\right\rangle$$

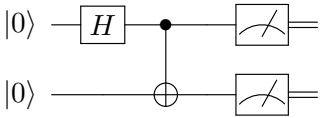
Quantum Circuits

Definition: A quantum circuit is a computational model in which quantum gates are applied to qubits to perform quantum computations.

Components of a Quantum Circuit:

- Qubits
- Quantum Gates (e.g., H, X, Z, $CNOT$, T)
- Measurements (collapsing qubits into classical bits)

Example: Bell State Circuit



Common Two-Qubit Quantum Gates

- CNOT (Controlled-NOT) Gate:

$$\text{CNOT}\,|11\rangle = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix} |11\rangle = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \end{bmatrix} = |10\rangle$$

- SWAP Gate:

$$\text{SWAP} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

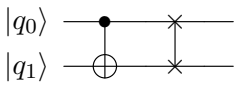
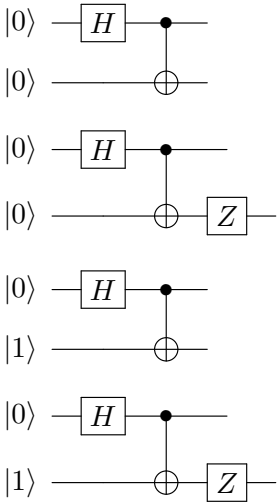


Table of Contents

Bell States

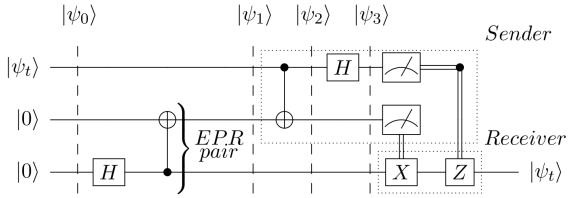


$$CX|H\,|00\rangle = CX\left(\frac{1}{\sqrt{2}}\,|00\rangle + \frac{1}{\sqrt{2}}\,|01\rangle\right) = \frac{1}{\sqrt{2}}CX\,|00\rangle + \frac{1}{\sqrt{2}}CX\,|01\rangle =$$
$$= \frac{1}{\sqrt{2}}\,|00\rangle + \frac{1}{\sqrt{2}}\,|11\rangle = \left|\Phi^+\right\rangle$$

$$|\Phi^+\rangle = \tfrac{1}{\sqrt{2}}(|00\rangle + |11\rangle)$$
$$|\Phi^-\rangle = \tfrac{1}{\sqrt{2}}(|00\rangle - |11\rangle)$$
$$|\Psi^+\rangle = \tfrac{1}{\sqrt{2}}(|10\rangle + |01\rangle)$$
$$|\Psi^-\rangle = \tfrac{1}{\sqrt{2}}(|10\rangle - |01\rangle)$$

Table of Contents

Quantum teleportation



$$|\psi_t\rangle = \alpha_t |0\rangle + \beta_t |1\rangle \quad |\psi_0\rangle = |\psi_t\rangle \otimes |00\rangle = \alpha_t |000\rangle + \beta_t |100\rangle$$

$$|\psi_1\rangle = \frac{\alpha_t}{\sqrt{2}} |000\rangle + \frac{\alpha_t}{\sqrt{2}} |011\rangle + \frac{\beta_t}{\sqrt{2}} |100\rangle + \frac{\beta_t}{\sqrt{2}} |111\rangle$$

$$|\psi_2\rangle = \frac{\alpha_t}{\sqrt{2}} |000\rangle + \frac{\alpha_t}{\sqrt{2}} |011\rangle + \frac{\beta_t}{\sqrt{2}} |110\rangle + \frac{\beta_t}{\sqrt{2}} |101\rangle$$

$$|\psi_3\rangle = \frac{1}{2} |00\rangle \otimes (\alpha_t |0\rangle + \beta_t |1\rangle) + \frac{1}{2} |01\rangle \otimes (\alpha_t |1\rangle + \beta_t |0\rangle) +$$

$$+ \frac{1}{2} |10\rangle \otimes (\alpha_t |0\rangle - \beta_t |1\rangle) + \frac{1}{2} |11\rangle \otimes (\alpha_t |1\rangle - \beta_t |0\rangle) =$$

$$= \frac{1}{2} |00\rangle \otimes |\psi_t\rangle + \frac{1}{2} |01\rangle \otimes |\overline{\psi_t}\rangle + \frac{1}{2} |10\rangle \otimes \left|\psi_t^\dagger\right\rangle + \frac{1}{2} |11\rangle \otimes \left|\overline{\psi_t^\dagger}\right\rangle$$

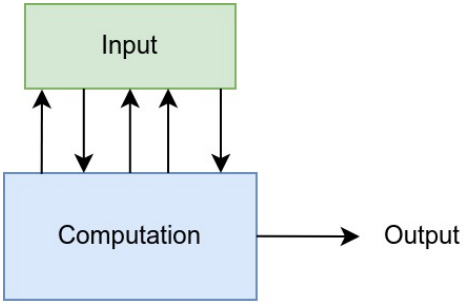
Table of Contents

Usual calculations

Keypoint

The entire input is provided to the computation — nothing being hidden from the computation.

The query model of computation



Input is in a form of function, which is accessed from computation via **queries**.
The input can be provided by:

- **Oracle**: who knows everything, but answers only questions
- **Blackbox**: we dont know the analytical expression of function, but can evaluate arguments

The query model of computation

Functions:

$$f : \Sigma^n \to \Sigma^m$$

where n and m are positive integers and $\Sigma = \{0, 1\}$

Queries

The computation does a query when the function is evaluated once and $f(x)$ can be used in the following calculations.
Efficiency is measured by counting the number of qeries.

Examples of query problems

Or

Input: $f : \Sigma^n \to \Sigma$

Output: 1 if there exists a string $x \in \Sigma^n$ for which $f(x) = 1$,
0 if there is no such string.

Parity

Input: $f : \Sigma^n \to \Sigma$

Output: 0 if $f(x) = 1$ for an even number of strings $x \in \Sigma^n$,
1 if $f(x) = 1$ for an odd number of strings $x \in \Sigma^n$.

Minimum

Input: $f : \Sigma^n \to \Sigma^m$

Output: The string $y \in \{f(x) : x \in \Sigma^n\}$ that comes first in the lexicographic ordering of Σ^m.

Examples of query problems

Sometimes we also consider query problems where we have a **promise** on the input. Inputs that don't satisfy the promise are considered as "don't care" inputs.

Unique Search

Unique Search

Input: $f : \Sigma^n \to \Sigma$
Promise: There is exactly one string $z \in \Sigma^n$ for which $f(z) = 1$, with $f(x) = 0$ for all strings $x \neq z$
Output: The string z

We sometimes consider very complicated and highly contrived problems to look for extremes that reveal potential advantages of quantum computing.

Query Gates

Unitary implementation the problems (e.g. OR, Parity check ...), which allows the usage with quantum circuits.

Definition

The query gate U_f for any function $f : \Sigma^n \to \Sigma^m$ is defined as:

$$U_f(|y\rangle|x\rangle) = |y \oplus f(x))|x\rangle$$

for all $x \in \Sigma^n$ and $y \in \Sigma^m$.

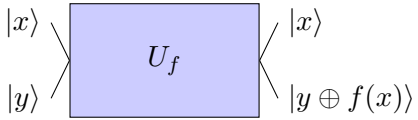
Notation

The string $y \oplus f(x)$ is the bitwise XOR of y and $f(x)$. For example:

$$\texttt{001} \oplus \texttt{101} = \texttt{100}$$

Query Gates

In circuit diagrammatic form U_f operates like this:



This gate is always unitary, for any choice of the function f.

Deutsch's Problem

Deutsch's problem is the **Parity** problem for functions of the form $f : \Sigma \to \Sigma$.

There are four functions of the form $f : \Sigma \to \Sigma$:

a	$f_1(a)$	a	$f_2(a)$	a	$f_3(a)$	a	$f_4(a)$
0	0	0	0	0	1	0	1
1	0	1	1	1	0	1	1

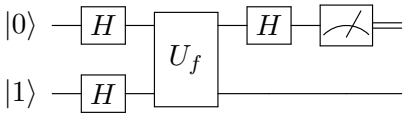
The functions f_1 and f_4 are **constant** while f_2 and f_3 are **balanced**.

Deutsch's Problem

Input: $f : \Sigma \to \Sigma$
Output: 0 if f is constant, 1 if f is balanced.

Deutsch's Algorithm

Deutsch's algorithm solves **Deutsch's problem** using a single query.



Output Interpretation

- 0 if f is constant
- 1 if f is balanced

Mathematical Representation

$$|\pi_1\rangle = |-\rangle|+\rangle = \frac{1}{2}(|0\rangle - |1\rangle)|0\rangle + \frac{1}{2}(|0\rangle - |1\rangle)|1\rangle$$

$$|\pi_2\rangle = \frac{1}{2}(|0 \oplus f(0)\rangle - |1 \oplus f(0)\rangle)|0\rangle + \frac{1}{2}(|0 \oplus f(1)\rangle - |1 \oplus f(1)\rangle)|1\rangle$$

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

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

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

$$= \begin{cases} (-1)^{f(0)}|-\rangle|+\rangle & f(0) \oplus f(1) = 0 \\ (-1)^{f(0)}|-\rangle|-\rangle & f(0) \oplus f(1) = 1 \end{cases}$$

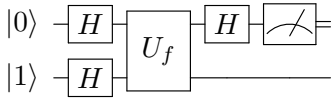
Mathematical Representation II

$$|\pi_2\rangle = \begin{cases} (-1)^{f(0)}|-\rangle|+\rangle & f(0) \oplus f(1) = 0 \\ (-1)^{f(0)}|-\rangle|-\rangle & f(0) \oplus f(1) = 1 \end{cases}$$

$$|\pi_3\rangle = \begin{cases} (-1)^{f(0)}|-\rangle|0\rangle & f(0) \oplus f(1) = 0 \\ (-1)^{f(0)}|-\rangle|1\rangle & f(0) \oplus f(1) = 1 \end{cases}$$

$$= (-1)^{f(0)}|-\rangle|f(0) \oplus f(1)\rangle$$

Phase Kickback



Mathematical Representation

$$|b \oplus c\rangle = X^c|b\rangle$$

$$U_f(|b\rangle|a\rangle) = |b \oplus f(a)\rangle|a\rangle = (X^{f(a)}|b\rangle)|a\rangle$$

$$U_f(|-\rangle|a\rangle) = (X^{f(a)}|-\rangle)|a\rangle = (-1)^{f(a)}|-\rangle|a\rangle$$

$$U_f(|-\rangle|a\rangle) = (-1)^{f(a)}|-\rangle|a\rangle \quad \leftarrow \text{phase kickback}$$

The Deutsch-Jozsa Problem

The Deutsch-Jozsa problem generalizes Deutsch's problem: for an input function $f : \Sigma^n \to \Sigma$, the task is to output 0 if f is constant and 1 if f is balanced.

When $n \geq 2$, some functions $f : \Sigma^n \to \Sigma$ are neither constant nor balanced.

Example

This function is neither constant nor balanced:

x	f(x)
00	0
01	0
10	0
11	1

Input functions that are neither constant nor balanced are "don't care" inputs.

The Deutsch-Jozsa Problem

The Deutsch-Jozsa problem generalizes Deutsch's problem: for an input function $f : \Sigma^n \to \Sigma$, the task is to output 0 if f is constant and 1 if f is balanced.

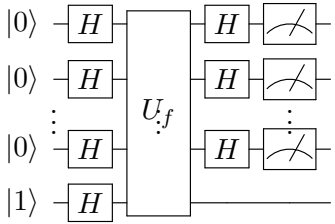
Deutsch-Jozsa problem

Input: $f : \Sigma^n \to \Sigma$
Promise: f is either constant or balanced
Output: 0 if f is constant, 1 if f is balanced

The Deutsch-Jozsa Problem

Deutsch-Jozsa problem

Input: $f : \Sigma^n \to \Sigma$
Promise: f is either constant or balanced
Output: 0 if f is constant, 1 if f is balanced



Deutsch-Jozsa Analysis

The Hadamard operation works like this on standard basis states:

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

We can express these two equations as one:

$$H|a\rangle = \frac{1}{\sqrt{2}}|0\rangle + \frac{1}{\sqrt{2}}(-1)^a|1\rangle = \frac{1}{\sqrt{2}}\sum_{b\in\{0,1\}}(-1)^{ab}|b\rangle$$

The Hadamard operation works like this on standard basis states:

$$H|a\rangle = \frac{1}{\sqrt{2}} \sum_{b \in \{0,1\}} (-1)^{ab} |b\rangle$$

Now suppose we perform a Hadamard operation on each of n qubits:

$$H^{\otimes n} |x_{n-1} \cdots x_1 x_0\rangle$$

$$= (H|x_{n-1}\rangle) \otimes \cdots \otimes (H|x_0\rangle)$$

$$= \left(\frac{1}{\sqrt{2}} \sum_{y_{n-1} \in \Sigma} (-1)^{x_{n-1} y_{n-1}} |y_{n-1}\rangle \right) \otimes \cdots \otimes \left(\frac{1}{\sqrt{2}} \sum_{y_0 \in \Sigma} (-1)^{x_0 y_0} |y_0\rangle \right)$$

$$= \frac{1}{\sqrt{2^n}} \sum_{y_{n-1} \cdots y_0 \in \Sigma^n} (-1)^{x_{n-1} y_{n-1} + \cdots + x_0 y_0} |y_{n-1} \cdots y_0\rangle$$

$$H^{\otimes n}|x_{n-1}\cdots x_1x_0\rangle = \frac{1}{\sqrt{2^n}}\sum_{y\in\Sigma^n}(-1)^{x\cdot y}|y\rangle$$

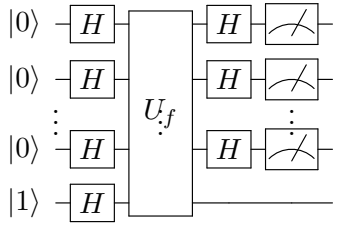
For binary strings $x = x_{n-1}\cdots x_0$ and $y = y_{n-1}\cdots y_0$ we define:

$$x\cdot y = x_{n-1}y_{n-1}\oplus\cdots\oplus x_0y_0$$

$$= \begin{cases} 1 & \text{if } x_{n-1}y_{n-1}+\cdots+x_0y_0 \text{ is odd} \\ 0 & \text{if } x_{n-1}y_{n-1}+\cdots+x_0y_0 \text{ is even} \end{cases}$$

Deutsch-Jozsa Analysis

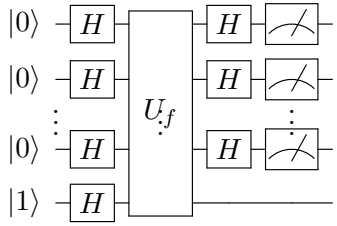
$$H^{\otimes n}|x\rangle = \frac{1}{\sqrt{2^n}} \sum_{y \in \Sigma^n} (-1)^{x \cdot y} |y\rangle$$



$$|\pi_1\rangle = |-\rangle \otimes \frac{1}{\sqrt{2^n}} \sum_{x \in \Sigma^n} |x\rangle$$

Deutsch-Jozsa Analysis

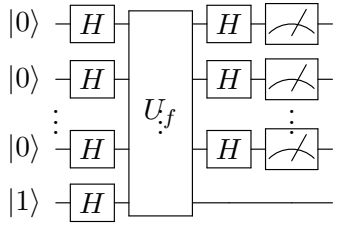
$$H^{\otimes n}|x\rangle = \frac{1}{\sqrt{2^n}} \sum_{y \in \Sigma^n} (-1)^{x \cdot y} |y\rangle$$



$$|\pi_2\rangle = |-\rangle \otimes \frac{1}{\sqrt{2^n}} \sum_{x \in \Sigma^n} (-1)^{f(x)} |x\rangle$$

Deutsch-Jozsa Analysis

$$H^{\otimes n}|x\rangle = \frac{1}{\sqrt{2^n}} \sum_{y \in \Sigma^n} (-1)^{x \cdot y} |y\rangle$$



$$|\pi_3\rangle = |-\rangle \otimes \frac{1}{2^n} \sum_{y \in \Sigma^n} \sum_{x \in \Sigma^n} (-1)^{f(x)+x \cdot y} |y\rangle$$

Deutsch-Jozsa Analysis

The probability for the measurements to give $y = 0^n$ is:

$$p(0^n) = \left| \frac{1}{2^n} \sum_{x \in \Sigma^n} (-1)^{f(x)} \right|^2 = \begin{cases} 1 & \text{if } f \text{ is constant} \\ 0 & \text{if } f \text{ is balanced} \end{cases}$$

The Deutsch-Jozsa algorithm therefore solves the Deutsch-Jozsa problem without error with a single query.

Any **deterministic** algorithm for the Deutsch-Jozsa problem must at least $2^{n-1} + 1$ queries.

A **probabilistic** algorithm can, however, solve the Deutsch-Jozsa problem using just a few queries:

1. Choose k input strings $x^1, \ldots, x^k \in \Sigma^n$ uniformly at random.
2. If $f(x^1) = \cdots = f(x^k)$, then answer 0 (constant), else answer 1 (balanced).

If f is constant, this algorithm is correct with probability 1.

If f is balanced, this algorithm is correct with probability $1 - 2^{-k+1}$.

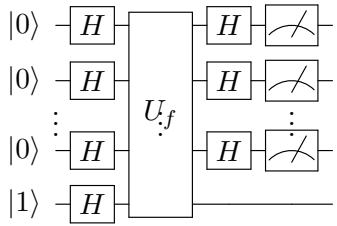
The Bernstein-Vazirani Problem

Bernstein-Vazirani problem

Input: $f : \Sigma^n \to \Sigma$

Promise: there exists a binary string $s = s_{n-1} \cdots s_0$ for which $f(x) = s \cdot x$ for all $x \in \Sigma^n$

Output: the string s



The Bernstein-Vazirani Problem

$$|\pi_3\rangle = |-\rangle \otimes \frac{1}{2^n} \sum_{y \in \Sigma^n} \sum_{x \in \Sigma^n} (-1)^{f(x)+x \cdot y} |y\rangle$$

$$= |-\rangle \otimes \frac{1}{2^n} \sum_{y \in \Sigma^n} \sum_{x \in \Sigma^n} (-1)^{s \cdot x + y \cdot x} |y\rangle$$

$$= |-\rangle \otimes \frac{1}{2^n} \sum_{y \in \Sigma^n} \sum_{x \in \Sigma^n} (-1)^{(s \oplus y) \cdot x} |y\rangle$$

$$= |-\rangle \otimes |s\rangle$$

The Deutsch-Jozsa circuit therefore solves the Bernstein-Vazirani problem with a single query.

Any probabilistic algorithm must make at least n queries to find s.

Table of Contents

Simon's Problem

Simon's problem

Input: A function $f : \Sigma^n \to \Sigma^m$

Promise: There exists a string $s \in \Sigma^n$ such that

$$f(x) = f(y) \iff (x = y) \text{ or } (x \oplus s = y)$$

for all $x, y \in \Sigma^n$

Output: The string s

Case 1: $s = 0^n$

The condition in the promise simplifies to:

$$f(x) = f(y) \iff x = y$$

This is equivalent to f being *one-to-one*.

Case 2: $s \neq 0^n$

The function f must be *two-to-one* to satisfy the promise:

$$f(x) = f(x \oplus s)$$

with *distinct output strings* for each pair.

x	$f(x)$
000	10011
001	00101
010	00101
011	10011
100	11010
101	00001
110	00001
111	11010

$$s = 011$$

$$f(000) = f(011) = 10011$$

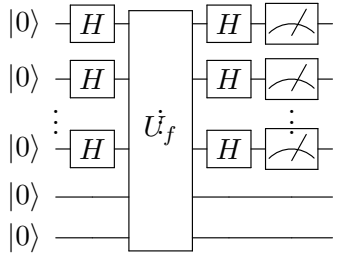
$$f(001) = f(010) = 00101$$

$$f(100) = f(111) = 11010$$

$$f(101) = f(110) = 00001$$

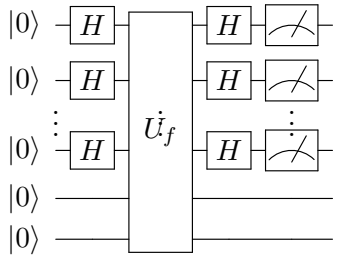
Simon's Algorithm

Simon's algorithm consists of running the following circuit several times, followed by a post-processing step.



Measurements yield results $y \in \Sigma^n$.

Simon's Algorithm Analysis



Measurements yield results $y \in \Sigma^n$.

$$|\pi_1\rangle = \frac{1}{\sqrt{2^n}} \sum_{x \in \Sigma^n} |0^m\rangle|x\rangle \qquad |\pi_2\rangle = \frac{1}{\sqrt{2^n}} \sum_{x \in \Sigma^n} |f(x)\rangle|x\rangle$$

$$|\pi_3\rangle = \frac{1}{\sqrt{2^n}} \sum_{x \in \Sigma^n} |f(x)\rangle \otimes \left(\frac{1}{\sqrt{2^n}} \sum_{y \in \Sigma^n} (-1)^{x \cdot y} |y\rangle \right) = \frac{1}{2^n} \sum_{y \in \Sigma^n} \sum_{x \in \Sigma^n} (-1)^{x \cdot y} |f(x)\rangle|y\rangle$$

Simon's Algorithm Analysis

$$\frac{1}{2^n}\sum_{y\in\Sigma^n}\sum_{x\in\Sigma^n}(-1)^{x\cdot y}|f(x)\rangle|y\rangle$$

$$p(y)=\left\|\frac{1}{2^n}\sum_{x\in\Sigma^n}(-1)^{x\cdot y}|f(x)\rangle\right\|^2=\left\|\frac{1}{2^n}\sum_{z\in\text{range}(f)}\left(\sum_{x\in f^{-1}(z)}(-1)^{x\cdot y}\right)|z\rangle\right\|^2=$$

$$=\frac{1}{2^{2n}}\sum_{z\in\text{range}(f)}\left|\sum_{x\in f^{-1}(z)}(-1)^{x\cdot y}\right|^2$$

$$\text{range}(f)=\{f(x)\mid x\in\Sigma^n\}$$

$$f^{-1}(\{z\})=\{x\in\Sigma^n\mid f(x)=z\}$$

Simon's Algorithm Analysis

$$p(y) = \frac{1}{2^{2n}} \sum_{z \in \text{range}(f)} \left| \sum_{x \in f^{-1}(z)} (-1)^{x \cdot y} \right|^2$$

Case 1: $s = 0^n$ Because f is one-to-one, there is a single element $x \in f^{-1}(\{z\})$ for every $z \in \text{range}(f)$:

$$\left| \sum_{x \in f^{-1}(\{z\})} (-1)^{x \cdot y} \right|^2 = 1$$

There are 2^n elements in $\text{range}(f)$, so

$$p(y) = \frac{1}{2^{2n}} \cdot 2^n = \frac{1}{2^n}, \text{ for every } y \in \Sigma^n$$

Simon's Algorithm Analysis II

Case 2: $s \neq 0^n$ There are two strings in the set $f^{-1}(\{z\})$ for each $z \in \text{range}(f)$; if $w \in f^{-1}(\{z\})$ either one of them, then $w \oplus s$ is the other.

$$\left| \sum_{x \in f^{-1}(\{z\})} (-1)^{x \cdot y} \right|^2 = \left| (-1)^{w \cdot y} + (-1)^{(w \oplus s) \cdot y} \right|^2$$

$$= |1 + (-1)^{s \cdot y}|^2 = \begin{cases} 4 & s \cdot y = 0 \\ 0 & s \cdot y = 1 \end{cases}$$

There are 2^{n-1} elements in $\text{range}(f)$, so

$$p(y) = \frac{1}{2^{2n}} \sum_{z \in \text{range}(f)} \left| \sum_{x \in f^{-1}(z)} (-1)^{x \cdot y} \right|^2 = \begin{cases} \frac{1}{2^{n-1}} & s \cdot y = 0 \\ 0 & s \cdot y = 1 \end{cases}$$

Classical Post-Processing

Running the circuit from Simon's algorithm one time gives us a random string $y \in \Sigma^n$.

Case 1: $s = 0^n$

$$p(y) = \frac{1}{2^n}$$

Case 2: $s \neq 0^n$

$$p(y) = \begin{cases} \frac{1}{2^{n-1}} & s \cdot y = 0 \\ 0 & s \cdot y = 1 \end{cases}$$

Suppose we run the circuit independently $k = n + r$ times, obtaining strings $y^1, \ldots, y^k$.

Classical Post-Processing II

$$\mathbf{y^1} = y^1_{n-1} \cdots y^1_0$$
$$\mathbf{y^2} = y^2_{n-1} \cdots y^2_0$$
$$\vdots$$
$$\mathbf{y^k} = y^k_{n-1} \cdots y^k_0$$

$$\mathbf{M} = \begin{pmatrix} y^1_{n-1} & \cdots & y^1_0 \\ y^2_{n-1} & \cdots & y^2_0 \\ \vdots & \ddots & \vdots \\ y^k_{n-1} & \cdots & y^k_0 \end{pmatrix}$$

$$\mathbf{M} \begin{pmatrix} s_{n-1} \\ \vdots \\ s_0 \end{pmatrix} = \begin{pmatrix} 0 \\ \vdots \\ 0 \end{pmatrix}$$

Using *Gaussian elimination*, we can efficiently compute the *null space* (modulo 2) of M. With probability greater than $1 - 2^{-r}$, it will be $\{0^n, s\}$.

Classical Difficulty

Any probabilistic algorithm making fewer than $2^{n/2-1} - 1$ queries will fail to solve Simon's problem with probability at least $1/2$.

- Simon's algorithm solves Simon's problem with a *linear* number of queries.
- Every classical algorithm for Simon's problem requires an *exponential* number of queries.

Table of Contents

Q&A and Closing the day