Advanced Parallel School 2022
Quantum Computing – Day 2
Quantum Algorithms

Mengoni Riccardo, PhD

15 Feb 2022

Quantum Computing @ CINECA

CINECA: Italian HPC center

CINECA Quantum Computing Lab:

- Research with Universities, Industries and QC startups

- Internship programs, Courses and Conference (HPCQC)

https://www.quantumcomputinglab.cineca.it

r.mengoni@cineca.it

Recap of Quantum Computing

Vectors

Ket: $$|\psi\rangle = \begin{pmatrix} \psi_1 \\ \psi_2 \\ \vdots \\ \psi_n \end{pmatrix}$$

$\psi_i \in \mathbb{C}$

Complex Number

Bra: $$\langle\psi| = \begin{pmatrix} \psi_1^* & \psi_2^* & \dots & \psi_n^* \end{pmatrix}$$

ψ_i^* Complex Conjugate

Scalar Product

$$\langle \phi | \psi \rangle = \begin{pmatrix} \phi_1^* & \phi_2^* & \cdots & \phi_n^* \end{pmatrix} \begin{pmatrix} \psi_1 \\ \psi_2 \\ \vdots \\ \psi_n \end{pmatrix}$$

$$\langle \phi | \psi \rangle \in \mathbb{C}$$

Complex Number

Scalar Product

$$\langle \phi | \psi \rangle = \begin{pmatrix} \phi_1^* & \phi_2^* & \dots & \phi_n^* \end{pmatrix} \begin{pmatrix} \psi_1 \\ \psi_2 \\ \vdots \\ \psi_n \end{pmatrix}$$

$$\langle \phi | \psi \rangle \in \mathbb{C}$$

Complex Number

The scalar product induces a **norm**

$$\| |\psi\rangle \| = \sqrt{\langle \psi | \psi \rangle}$$

Outer Product

$$|\psi\rangle\langle\phi| = \begin{pmatrix} \psi_1 \\ \psi_2 \\ \vdots \\ \psi_n \end{pmatrix} \begin{pmatrix} \phi_1^* & \phi_2^* & \dots & \phi_n^* \end{pmatrix} = \begin{pmatrix} \psi_1\phi_1^* & \psi_1\phi_2^* & \dots & \psi_1\phi_n^* \\ \psi_2\phi_1^* & \psi_2\phi_2^* & \dots & \psi_2\phi_n^* \\ \vdots & \vdots & & \vdots \\ \psi_n\phi_1^* & \psi_n\phi_2^* & \dots & \psi_n\phi_n^* \end{pmatrix}$$

Dimension = n x n

Tensor Product

$$|\phi\rangle \otimes |\psi\rangle = \begin{pmatrix} \phi_1 \begin{pmatrix} \psi_1 \\ \psi_2 \\ \vdots \\ \psi_n \end{pmatrix} \\ \phi_2 \begin{pmatrix} \psi_1 \\ \psi_2 \\ \vdots \\ \psi_n \end{pmatrix} \\ \vdots \\ \phi_n \begin{pmatrix} \psi_1 \\ \psi_2 \\ \vdots \\ \psi_n \end{pmatrix} \end{pmatrix}$$

Dimension = n^2

Tensor Product

$$|\phi\rangle \otimes |\psi\rangle = \begin{pmatrix} \phi_1 \begin{pmatrix} \psi_1 \\ \psi_2 \\ \vdots \\ \psi_n \end{pmatrix} \\ \phi_2 \begin{pmatrix} \psi_1 \\ \psi_2 \\ \vdots \\ \psi_n \end{pmatrix} \\ \vdots \\ \phi_n \begin{pmatrix} \psi_1 \\ \psi_2 \\ \vdots \\ \psi_n \end{pmatrix} \end{pmatrix}$$

Dimension = n^2

Compact form:

$$|\psi\rangle \otimes |\phi\rangle = |\psi\rangle\,|\phi\rangle = |\psi\,\phi\rangle$$

1. Unit of Information

Classically

Unit of classical information is the bit

State of a bit:

$$|0\rangle = \begin{pmatrix} 1 \\ 0 \end{pmatrix} \qquad |1\rangle = \begin{pmatrix} 0 \\ 1 \end{pmatrix}$$

Quantumly

To a closed quantum system is associated a space of states *H* which is a Hilbert space. The pure state of the system is then represented by a unit norm vector on such Hilbert space.

The unit of quantum information is the quantum bit a.k.a. Qubit

State of a qubit:

$$|\psi\rangle = \alpha|0\rangle + \beta|1\rangle = \begin{pmatrix} \alpha \\ \beta \end{pmatrix}$$

Postulates of Quantum Computing (1)

Space of states: $\mathcal{H} \simeq \mathbb{C}^2$

State of a qubit:

$$|\psi\rangle = \alpha|0\rangle + \beta|1\rangle = \begin{pmatrix} \alpha \\ \beta \end{pmatrix}$$

$$\alpha, \beta \in \mathbb{C} \qquad |\alpha|^2 + |\beta|^2 = 1$$

Postulates of Quantum Computing (1)

Space of states: $\mathcal{H} \simeq \mathbb{C}^2$

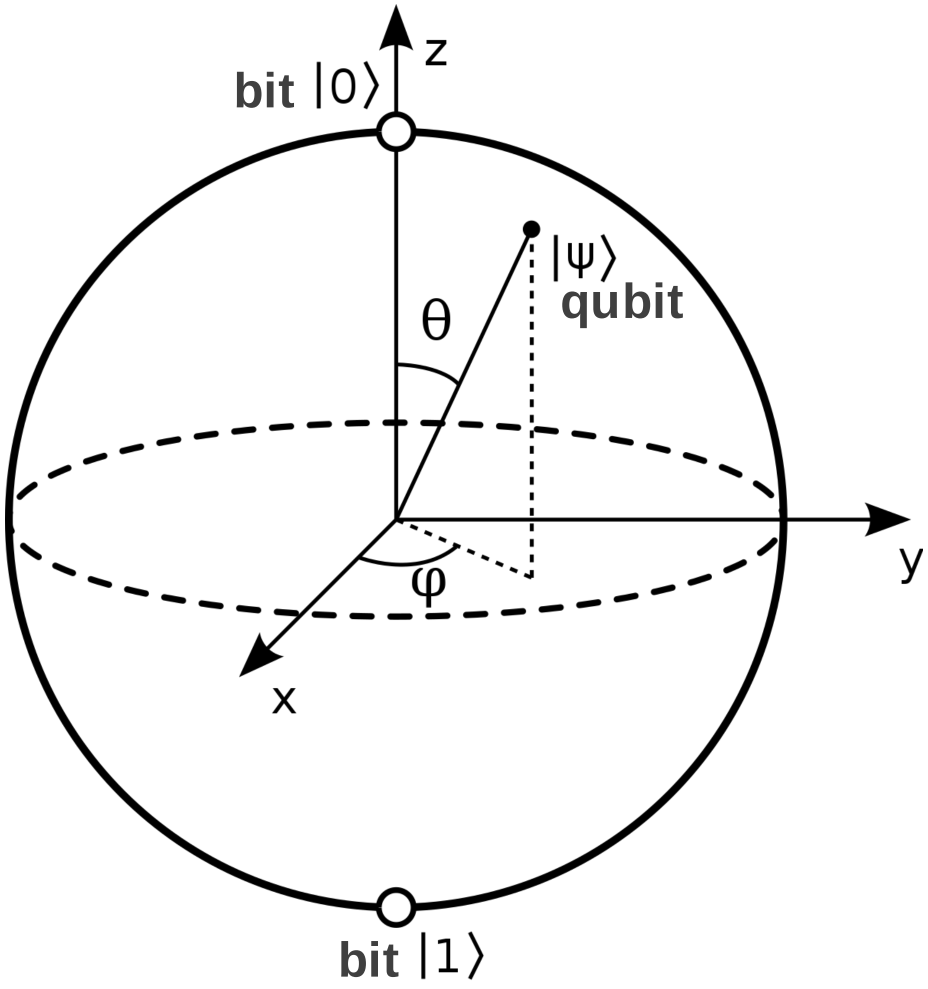
State of a qubit:

$$|\psi\rangle = \alpha|0\rangle + \beta|1\rangle = \begin{pmatrix}\alpha \\ \beta\end{pmatrix}$$

$$\alpha, \beta \in \mathbb{C} \qquad |\alpha|^2 + |\beta|^2 = 1$$

Can be parametrized as:

$$\boxed{|\psi\rangle = \cos\left(\frac{\theta}{2}\right)|0\rangle + e^{i\phi}\sin\left(\frac{\theta}{2}\right)|1\rangle \qquad \theta \in [0,\pi] \quad \phi \in [0, 2\pi]}$$



2. Composite systems

Classically

State of N bits:

$$|000\ldots0\rangle,\ |100\ldots0\rangle,\ |010\ldots0\rangle\ \ldots\ |111\ldots1\rangle$$

Postulates of Quantum Computing (2)

Quantumly

The space of states of a composite system is the tensor product of the spaces of the subsystems

$$\mathbb{C}^2 \otimes \mathbb{C}^2 \otimes \ldots \otimes \mathbb{C}^2$$

State of N qubits:

$$\alpha_1 |000..0\rangle + \alpha_2 |100..0\rangle + \alpha_3 |010..0\rangle + \ldots \quad \alpha_n |111..1\rangle$$

$$\alpha_i \in \mathbb{C} \qquad \sum_i |\alpha_i|^2 = 1$$

Quantum Entanglement

States that can be written as tensor product

$$|\psi\rangle = |\psi_1\rangle \otimes |\psi_2\rangle \otimes \dots \otimes |\psi_N\rangle$$

are called **factorable or product states**

Quantum Entanglement

States that **can NOT** be written as tensor product

$$|\psi\rangle \neq |\psi_1\rangle \otimes |\psi_2\rangle \otimes \dots \otimes |\psi_N\rangle$$

are called **entangled states**

Postulates of Quantum Computing (2)

Quantum Entangled

Bell's states

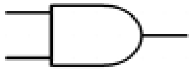
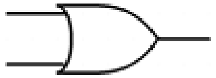
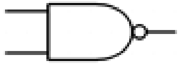
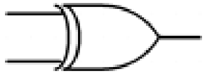
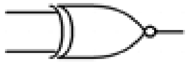
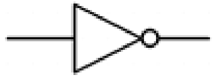
$$\frac{1}{\sqrt{2}}\left(|00\rangle + |11\rangle\right) \qquad \frac{1}{\sqrt{2}}\left(|01\rangle + |10\rangle\right)$$

$$\frac{1}{\sqrt{2}}\left(|00\rangle - |11\rangle\right) \qquad \frac{1}{\sqrt{2}}\left(|01\rangle - |10\rangle\right)$$

3. State Change

Postulates of Quantum Computing (3)

Classically: logic gates

Logic Gate	Symbol	Description	Boolean
AND		Output is at logic 1 when, and only when all its inputs are at logic 1,otherwise the output is at logic 0.	$X = A \cdot B$
OR		Output is at logic 1 when one or more are at logic 1.If all inputs are at logic 0,output is at logic 0.	$X = A+B$
NAND		Output is at logic 0 when,and only when all its inputs are at logic 1,otherwise the output is at logic 1	$X = \overline{A \cdot B}$
NOR		Output is at logic 0 when one or more of its inputs are at logic 1.If all the inputs are at logic 0,the output is at logic 1.	$X = \overline{A+B}$
XOR		Output is at logic 1 when one and Only one of its inputs is at logic 1. Otherwise is it logic 0.	$X = A \oplus B$
XNOR		Output is at logic 0 when one and only one of its inputs is at logic1.Otherwise it is logic 1. Similar to XOR but inverted.	$X = \overline{A \oplus B}$
NOT		Output is at logic 0 when its only input is at logic 1, and at logic 1 when its only input is at logic 0.That's why it is called and INVERTER	$X = \overline{A}$

Quantumly

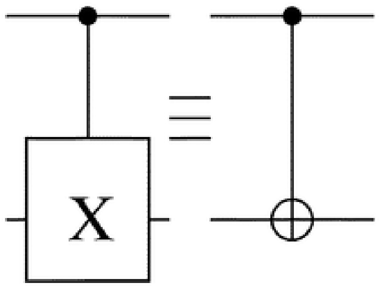
The state change of a closed quantum system is described by a unitary operator

$$i \frac{d|\psi\rangle}{dt} = H|\psi\rangle \Rightarrow |\psi(t)\rangle = e^{-iHt}|\psi(0)\rangle$$

$$U = e^{-iHt}$$

Schrodinger Equation

Quantumly: Quantum Gates

X Gate Bit–flip, Not	X	$\equiv \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} \alpha \\ \beta \end{bmatrix} = \beta\lvert 0\rangle + \alpha\lvert 1\rangle$
Z Gate Phase–flip	Z	$\equiv \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} \begin{bmatrix} \alpha \\ \beta \end{bmatrix} = \alpha\lvert 0\rangle - \beta\lvert 1\rangle$
H Gate Hadamard	H	$\equiv \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \begin{bmatrix} \alpha \\ \beta \end{bmatrix} = \frac{\alpha+\beta\lvert 0\rangle + \alpha-\beta\lvert 1\rangle}{\sqrt{2}}$
T Gate	T	$\equiv \begin{bmatrix} 1 & 0 \\ 0 & e^{i\pi/4} \end{bmatrix} \begin{bmatrix} \alpha \\ \beta \end{bmatrix} = \alpha\lvert 0\rangle + e^{i\pi/4}\beta\lvert 1\rangle$
Controlled Not Controlled X CNot		$\equiv \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} a \\ b \\ c \\ d \end{bmatrix} = a\lvert 00\rangle + b\lvert 01\rangle + d\lvert 10\rangle + c\lvert 11\rangle$

Postulates of Quantum Computing (4)

4. Measurement

Postulates of Quantum Computing (4)

Classically

Measuring returns the state of a bit with certainty

$$|0\rangle \xrightarrow{\text{Measure}} \overset{\text{Outcome}}{|0\rangle} \qquad |1\rangle \xrightarrow{\text{Measure}} \overset{\text{Outcome}}{|1\rangle}$$

Measurements do not affect the state of a bit

Quantumly

Measuring returns the bit state with some probability

Outcome

$$|\psi\rangle = \alpha|0\rangle + \beta|1\rangle \xrightarrow{\text{Measure}} \begin{cases} |0\rangle \text{ with } P_r = |\alpha|^2 \\ |1\rangle \text{ with } P_r = |\beta|^2 \end{cases}$$

Measurement affects the state of a qubit

Postulates of Quantum Computing (4)

Quantumly

- To **any observable** physical quantity is associated an **hermitian operator** *O*

$$O|o_i\rangle = o_i|o_i\rangle$$

- A **measurement** outcomes are the **possibile eigenvalues** $\{o_i\}$.

- The **probability of obtaining** o_i as a result of the measurement is

$$P_r(o_i) = |\langle\psi|o_i\rangle|^2$$

- The effect of the **measure is to change the state** $|\psi\rangle$ **into the eigenvector of** *O*

$$|\psi\rangle \rightarrow |o_i\rangle$$

Quantum Algorithms

Quantum Algorithm = Quantum Circuit

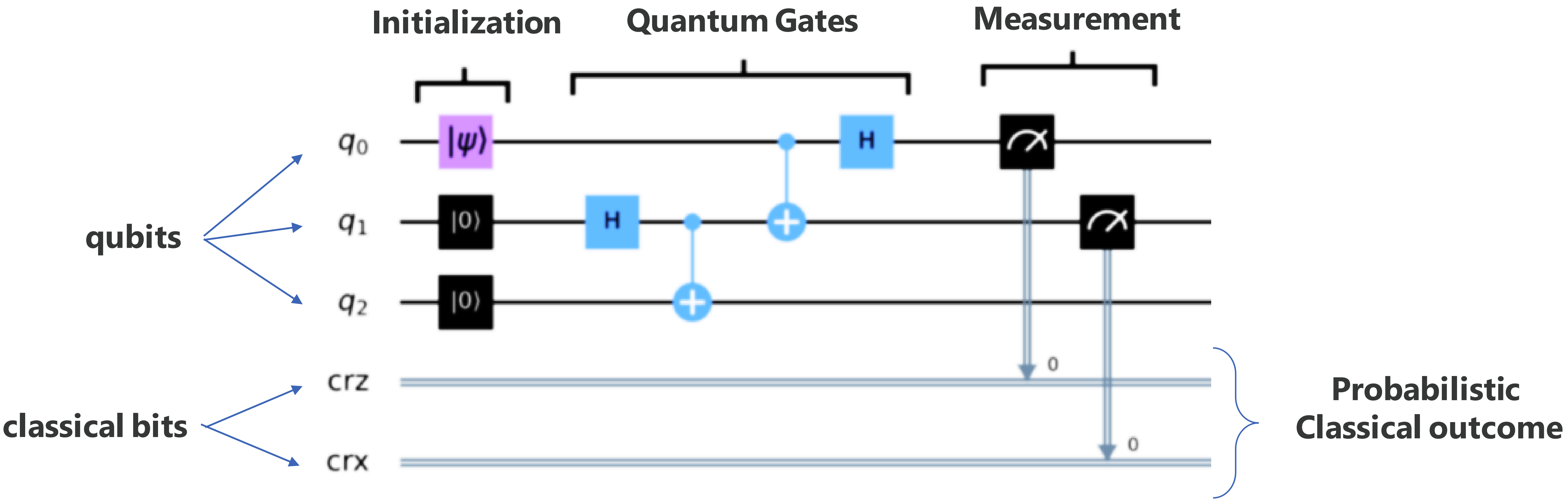
A quantum circuit with *n* input qubits and *n* output qubits is defined by a unitary transformation

$$U \in U(2^n)$$

$$U^\dagger U = UU^\dagger = I$$

$$U^{-1} = U^\dagger$$

Quantum Algorithms



Quantum Algorithms: Gates

Single Qubit Gates

Generic single qubit rotation:

$$R_{\vec{n}}(\theta) = \cos\left(\frac{\theta}{2}\right) I - i \sin\left(\frac{\theta}{2}\right) \vec{n} \cdot \vec{\sigma}$$

Pauli matrices:

$$\sigma_x = X = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \qquad \sigma_y = Y = \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix} \qquad \sigma_z = Z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}$$

Identity: $I = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$

Single Qubit Gates: Hadamard

$$H = \frac{1}{\sqrt{2}}\begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} \qquad H = -\boxed{H}-$$

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

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

Single Qubit Gates: Phase

$$U_\phi = \begin{pmatrix} 1 & 0 \\ 0 & e^{i\phi} \end{pmatrix}$$

$$U_\phi |0\rangle = |0\rangle$$

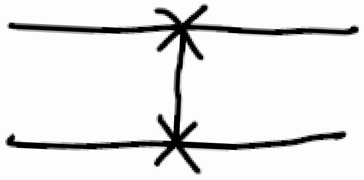
$$U_\phi |1\rangle = e^{i\phi} |1\rangle$$

Two Qubit Gates: SWAP

$$U_{SWAP}\,|z_1\rangle|z_2\rangle = |z_2\rangle|z_1\rangle \qquad z_1, z_2 \in \{0,1\}$$

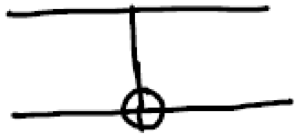
$$U_{SWAP} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

SWAP =



Two Qubit Gates: Control Not

$$U_{cx} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix}$$

$$U_{cx} =$$ 

$$U_{cx}\,|z_1\rangle|z_2\rangle = |z_1\rangle\, X^{z_1}\,|z_2\rangle$$

$$U_{cx}\,|00\rangle = |00\rangle$$

$$U_{cx}\,|10\rangle = |11\rangle$$

$$U_{cx}\,|01\rangle = |01\rangle$$

$$U_{cx}\,|11\rangle = |10\rangle$$

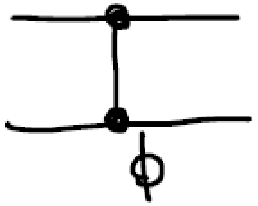
Two Qubit Gates: Control Unitary

$$CU|z_1\rangle|z_2\rangle = |z_1\rangle U^{z_1}|z_2\rangle$$

Control Phase

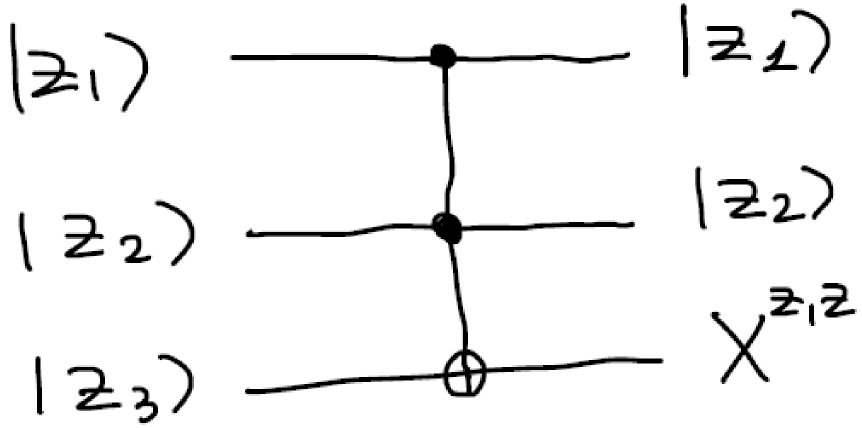
$$CU_\phi|z_1\rangle|z_2\rangle = |z_1\rangle U_\phi^{z_1}|z_2\rangle$$

$$CU_\phi = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & e^{i\phi} \end{pmatrix} \qquad CU_\phi =$$



Three Qubit Gates: Toffoli

$$U_{C_2X}\,|z_1\, z_2\, z_3\rangle = |z_1\, z_2\rangle\, X^{z_1 z_2}\, |z_3\rangle$$



$$X^{z_1 z_2}\, |z_3\rangle = |z_3 \oplus z_1 \cdot z_2\rangle$$

Quantum Algorithms: Universality

Universal set of Quantum Gates

We can exactly build any unitary $U \in U(2^n)$ on n qubits

by means of single qubit gates and Control-Not

$$\mathcal{G}_{ex} = \{ U \in \mathcal{U}(2) \; ; \; U_{cx} \}$$

Universal set of Quantum Gates

We can exactly build any unitary $U \in U(2^n)$ on n qubits

by means of single qubit gates and Control-Not

$$\mathcal{G}_{ex} = \{ U \in \mathcal{U}(2) \; ; \; U_{cx} \}$$

$$R_{\vec{n}}(\theta) = \cos\left(\frac{\theta}{2}\right) I - i \sin\left(\frac{\theta}{2}\right) \vec{n} \cdot \vec{\sigma}$$

$$U_{cx} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix}$$

Universal set of Quantum Gates

Given $U, U' \in \mathcal{U}(2^n)$, U' approximates U within ε $(\varepsilon > 0)$ if $d(U, U') < \varepsilon$

Universal set of Quantum Gates

Given $U, U' \in \mathcal{U}(2^n)$, U' **approximates** U **within** ε $(\varepsilon > 0)$ **if** $d(U, U') < \varepsilon$

where $$d(U, U') = \max_{|\psi\rangle} \| (U - U')|\psi\rangle \|$$

and $$\| |\psi\rangle \| = \sqrt{\langle\psi|\psi\rangle}$$

Universal set of Quantum Gates

We can approximate any unitary $U \in U(2^n)$ on n qubits

by means of the following gates

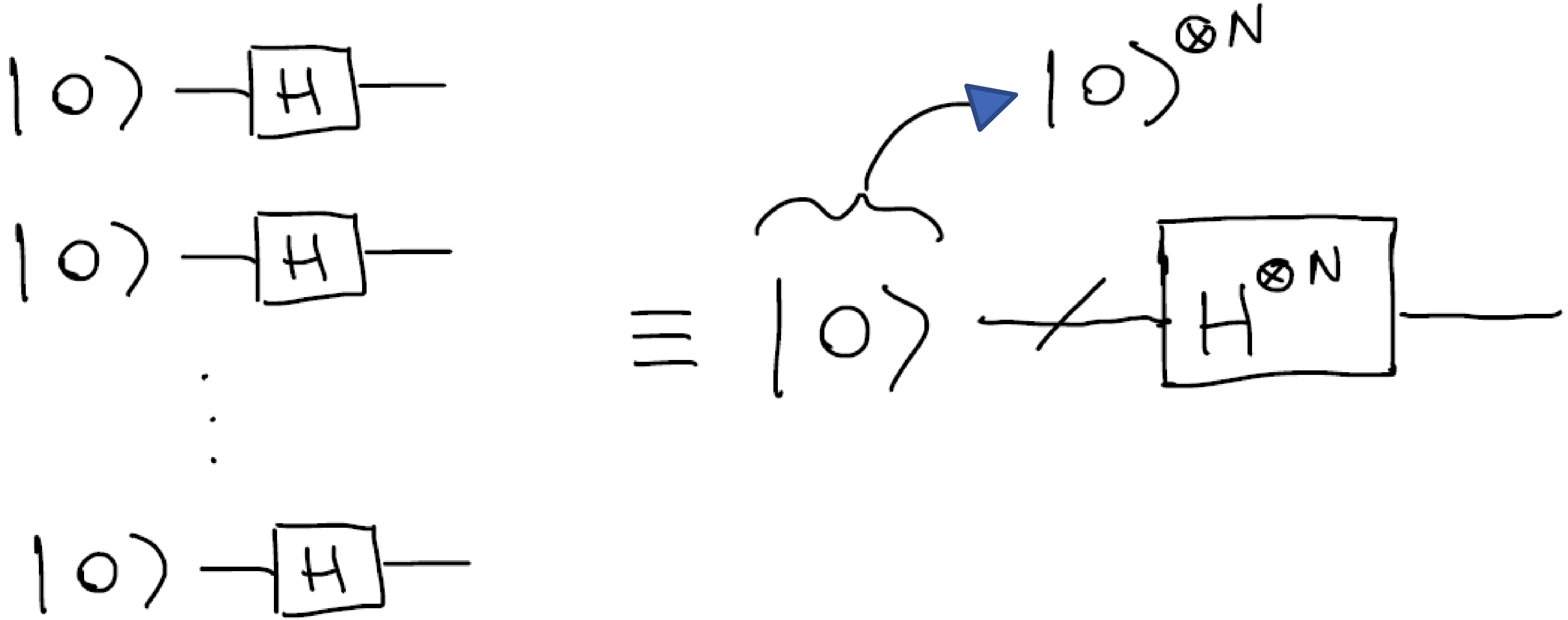
$$\{H, S, T, U_{cx}\}$$

$$H = \frac{1}{\sqrt{2}}\begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} \quad S = \begin{pmatrix} 1 & 0 \\ 0 & i \end{pmatrix} \quad T = \begin{pmatrix} 1 & 0 \\ 0 & e^{i\pi/4} \end{pmatrix}$$

Quantum Algorithms: basics

CINECA QUANTUM COMPUTING LAB

Multiple Hadamard gates



Single Qubit Gates: Hadamard

$$H = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} \qquad H = -\boxed{H}-$$

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

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

Multiple Hadamard gates

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

$$H^{\otimes N} = \frac{1}{\sqrt{2^N}} \sum_{x,y\in\{0,1\}^N} (-1)^{x\cdot y}\, |x\rangle\langle y|$$

Multiple Hadamard gates

$$|0\rangle^{\otimes N} \xrightarrow{H^{\otimes N}} \frac{1}{\sqrt{2^N}} \sum_{x,y\in\{0,1\}^N} (-1)^{x\cdot y}\, |x\rangle\langle y|0\rangle =$$

Multiple Hadamard gates

$$-/-\boxed{H^{\otimes N}}-$$

$$|0\rangle^{\otimes N} \xrightarrow{H^{\otimes N}} \overbrace{\frac{1}{\sqrt{2^N}} \sum_{x,y\in\{0,1\}^N} (-1)^{x\cdot y} |x\rangle \underbrace{\langle y|0\rangle}_{\delta_{0y}}} =$$

Kronecker delta

$$\delta_{i,j} \stackrel{\text{def}}{=} \begin{cases} 1 & \text{if} \quad i = j \\ 0 & \text{otherwise} \end{cases}$$

Multiple Hadamard gates

$$|0\rangle^{\otimes N} \xrightarrow{H^{\otimes N}} \frac{1}{\sqrt{2^N}} \sum_{x,y\in\{0,1\}^N} (-1)^{x\cdot y} |x\rangle \underbrace{\langle y|0\rangle}_{\delta_{0y}} =$$

$$= \boxed{\frac{1}{\sqrt{2^N}} \sum_{x\in\{0,1\}^N} |x\rangle}$$

Kronecker delta

$$\delta_{i,j} \stackrel{\text{def}}{=} \begin{cases} 1 & \text{if} \quad i = j \\ 0 & \text{otherwise} \end{cases}$$

Oracle: Function evaluation

Given a function $f:\{0,1\}^N \to \{0,1\}^M$, an algorithm to evaluate such function is given by the unitary U_f

$$|x\rangle|y\rangle \xrightarrow{U_f} |x\rangle|y \oplus f(x)\rangle$$

where $x \in \{0,1\}^N$ $y \in \{0,1\}^M$

Deutsch Jozsa Algorithm

D-J Problem

Consider a function $f : \{0,1\}^N \to \{0,1\}$ with the premise that it is either constant (returns 0 on all inputs or 1 on all inputs) or balanced (returns 1 for half of the inputs and 0 for the other half).

$$A_0 = \{x \in \{0,1\}^N \mid f(x) = 0\}$$

$$A_1 = \{x \in \{0,1\}^N \mid f(x) = 1\}$$

$$\Rightarrow \begin{cases} |A_0| = 2^N \text{ OR } |A_1| = 2^N, & \text{constant} \\ |A_0| = |A_1| = 2^{N-1}, & \text{balanced} \end{cases}$$

Deutsch Jozsa Algorithm

How many evaluations («queries») of the function are needed to determine with certainty if such function is balanced or constant?

Deutsch Jozsa Algorithm

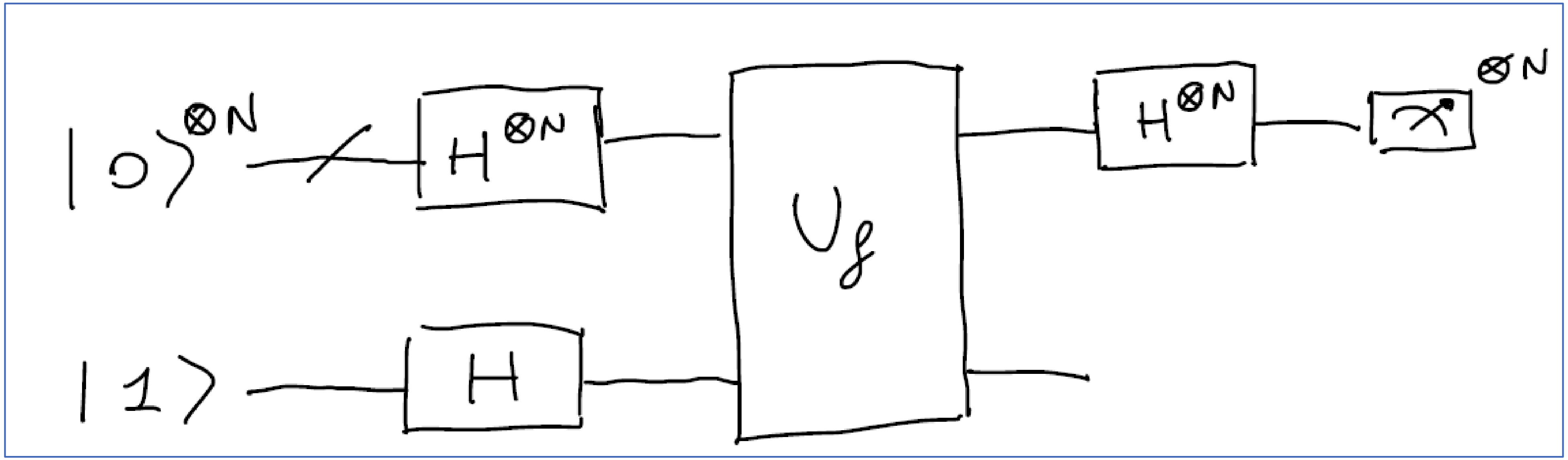
How many evaluations («queries») of the function are needed to determine with certainty if such function is balanced or constant?

Classically

Since the possible input strings are 2^N, we need to check in the worst case (half +1) strings, i.e. $2^{N-1} + 1$ strings

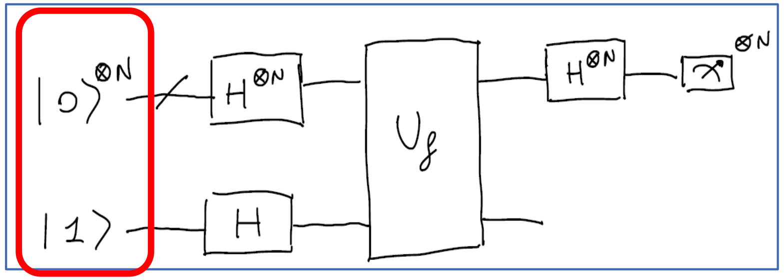
Classical Query Complexity $\sim 2^{N-1} + 1$

Quantum Solution



$f : \{0,1\}^N \to \{0,1\}$ and $|x\rangle|y\rangle \xrightarrow{U_f} |x\rangle|y \oplus f(x)\rangle$

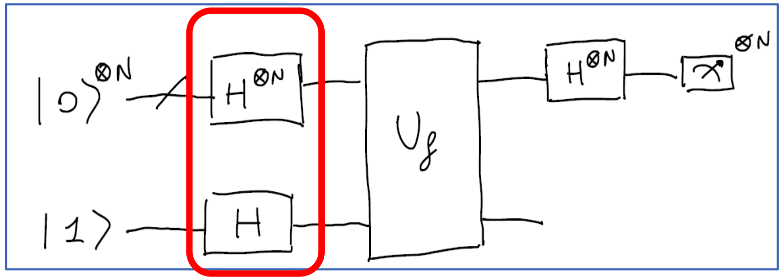
Deutsch Jozsa Algorithm



Step by step analysis

$$|0\rangle^{\otimes N}|1\rangle$$

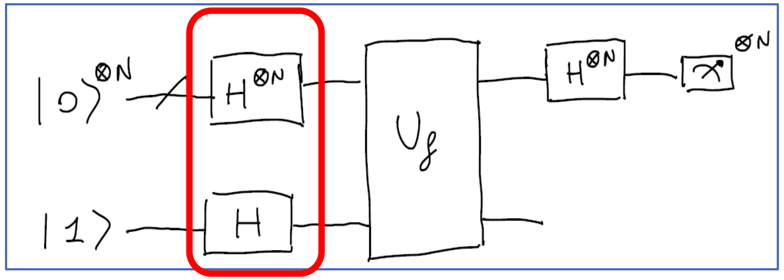
Deutsch Jozsa Algorithm



Step by step analysis

$$|0\rangle^{\otimes N}|1\rangle \xrightarrow{H^{\otimes N} H} H^{\otimes N}|0\rangle^{\otimes N} \, H|1\rangle$$

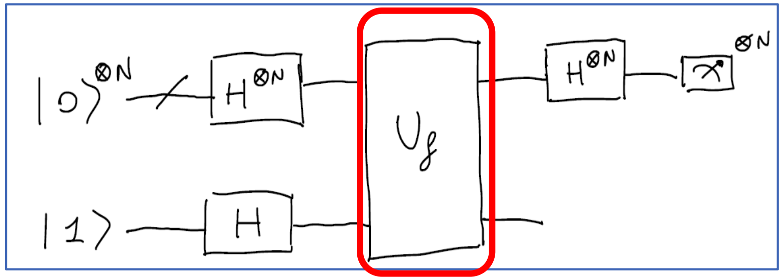
Deutsch Jozsa Algorithm



Step by step analysis

$$|0\rangle^{\otimes N}|1\rangle \xrightarrow{H^{\otimes N}H} H^{\otimes N}|0\rangle^{\otimes N} H|1\rangle = \frac{1}{\sqrt{2^N}}\sum_x |x\rangle\left(\frac{|0\rangle-|1\rangle}{\sqrt{2}}\right)$$

Deutsch Jozsa Algorithm

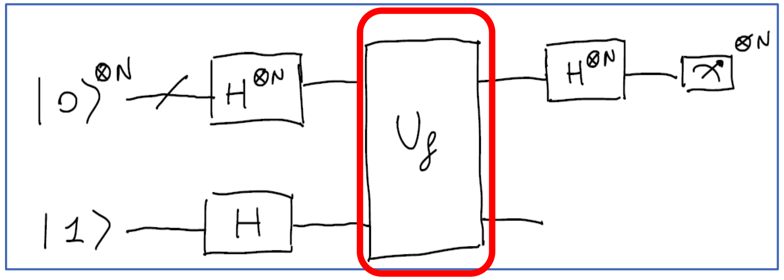


Step by step analysis

$$|0\rangle^{\otimes N}|1\rangle \xrightarrow{H^{\otimes N} H} H^{\otimes N}|0\rangle^{\otimes N} H|1\rangle = \frac{1}{\sqrt{2^N}} \sum_x |x\rangle \left(\frac{|0\rangle - |1\rangle}{\sqrt{2}}\right)$$

$$\frac{1}{\sqrt{2^N}} \sum_x |x\rangle \left(\frac{|0\rangle - |1\rangle}{\sqrt{2}}\right) \xrightarrow{U_f}$$

Deutsch Jozsa Algorithm

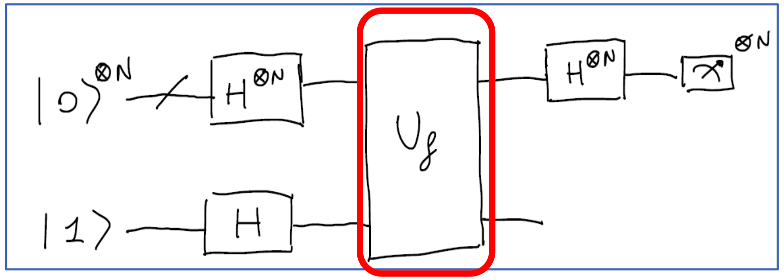


Step by step analysis

$$|0\rangle^{\otimes N}|1\rangle \xrightarrow{H^{\otimes N}H} H^{\otimes N}|0\rangle^{\otimes N} H|1\rangle = \frac{1}{\sqrt{2^N}}\sum_x |x\rangle\left(\frac{|0\rangle-|1\rangle}{\sqrt{2}}\right)$$

$$\frac{1}{\sqrt{2^N}}\sum_x |x\rangle\left(\frac{|0\rangle-|1\rangle}{\sqrt{2}}\right) \xrightarrow{U_f} \frac{1}{\sqrt{2^N}}\sum_x |x\rangle\frac{|0\oplus f(x)\rangle}{\sqrt{2}} - \frac{1}{\sqrt{2^N}}\sum_x |x\rangle\frac{|1\oplus f(x)\rangle}{\sqrt{2}} =$$

Deutsch Jozsa Algorithm



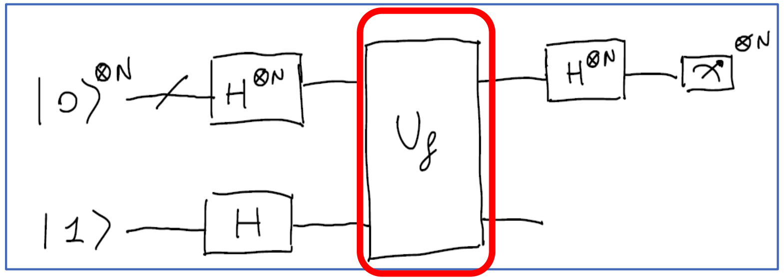
Step by step analysis

$$|0\rangle^{\otimes N}|1\rangle \xrightarrow{H^{\otimes N}H} H^{\otimes N}|0\rangle^{\otimes N}\, H|1\rangle = \frac{1}{\sqrt{2^N}}\sum_{x}|x\rangle\left(\frac{|0\rangle-|1\rangle}{\sqrt{2}}\right)$$

$$\frac{1}{\sqrt{2^N}}\sum_{x}|x\rangle\left(\frac{|0\rangle-|1\rangle}{\sqrt{2}}\right) \xrightarrow{U_f} \frac{1}{\sqrt{2^N}}\sum_{x}|x\rangle\frac{|0\oplus f(x)\rangle}{\sqrt{2}} - \frac{1}{\sqrt{2^N}}\sum_{x}|x\rangle\frac{|1\oplus f(x)\rangle}{\sqrt{2}} =$$

$$= \frac{1}{\sqrt{2^N}}\sum_{x}|x\rangle\left(\frac{|f(x)\rangle-|1\oplus f(x)\rangle}{\sqrt{2}}\right)$$

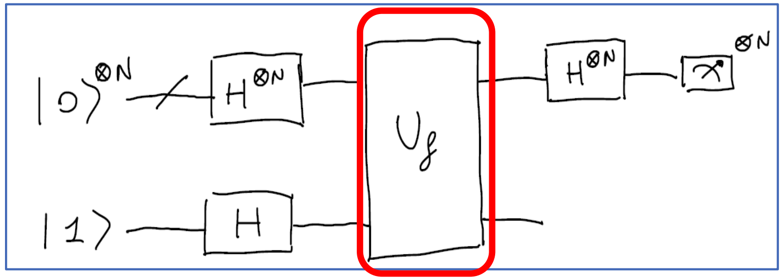
Deutsch Jozsa Algorithm



Step by step analysis

$$\frac{1}{\sqrt{2^N}} \sum_x |x\rangle \left(\frac{|f(x)\rangle - |1 \oplus f(x)\rangle}{\sqrt{2}} \right)$$

Deutsch Jozsa Algorithm

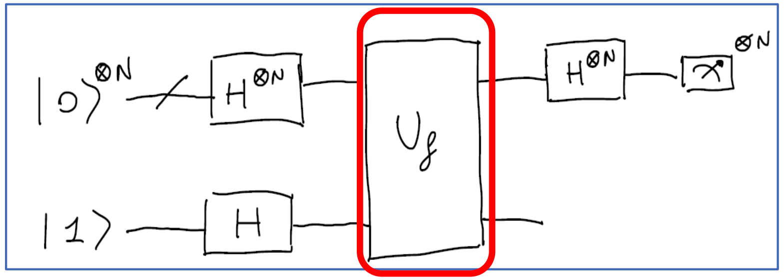


Step by step analysis

$$\frac{1}{\sqrt{2^N}} \sum_x |x\rangle \boxed{\left(\frac{|f(x)\rangle - |1 \oplus f(x)\rangle}{\sqrt{2}}\right)}$$

$$\left(f(x) \in \{0,1\} \rightarrow \begin{cases} f(x) = 0 & \boxed{\dfrac{|0\rangle - |1\rangle}{\sqrt{2}}} \\ f(x) = 1 & \boxed{\dfrac{|1\rangle - |0\rangle}{\sqrt{2}}} \end{cases} \right)$$

Deutsch Jozsa Algorithm



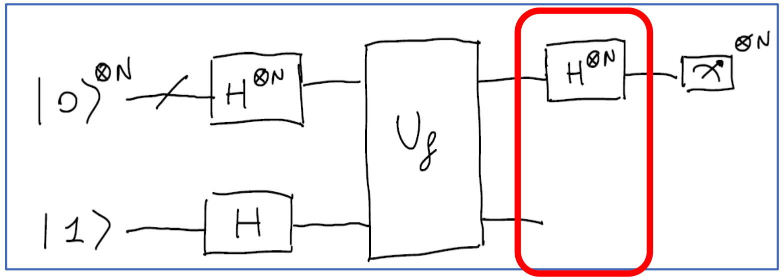
Step by step analysis

$$\frac{1}{\sqrt{2^N}}\sum_x |x\rangle \boxed{\left(\frac{|f(x)\rangle - |1\oplus f(x)\rangle}{\sqrt{2}}\right)}$$

$$f(x) \in \{0,1\} \rightarrow \begin{cases} f(x)=0 & \boxed{\dfrac{|0\rangle - |1\rangle}{\sqrt{2}}} \\ f(x)=1 & \boxed{\dfrac{|1\rangle - |0\rangle}{\sqrt{2}}} \end{cases}$$

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

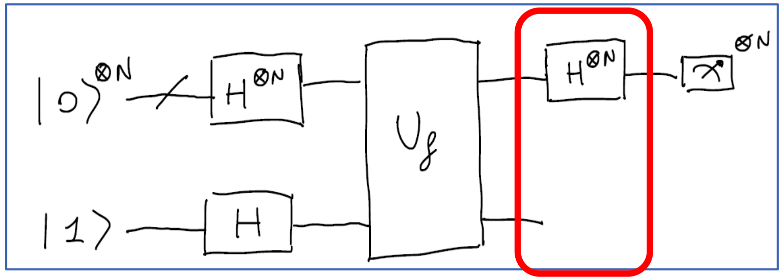
Deutsch Jozsa Algorithm



Step by step analysis

$$\frac{1}{\sqrt{2^N}} \sum_x (-1)^{f(x)} |x\rangle \left(\frac{|0\rangle - |1\rangle}{\sqrt{2}} \right)$$

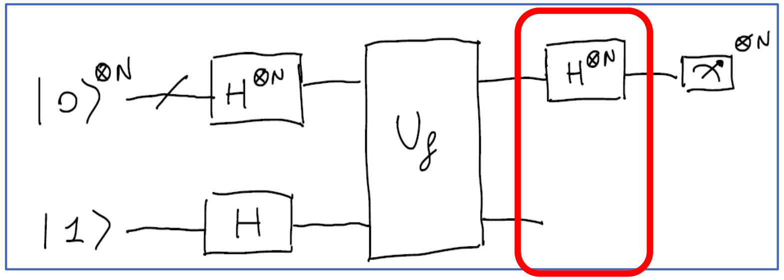
Deutsch Jozsa Algorithm



Step by step analysis

$$\frac{1}{\sqrt{2^N}} \sum_x (-1)^{f(x)} |x\rangle \cancel{\left(\frac{|0\rangle - |1\rangle}{\sqrt{2}} \right)}$$

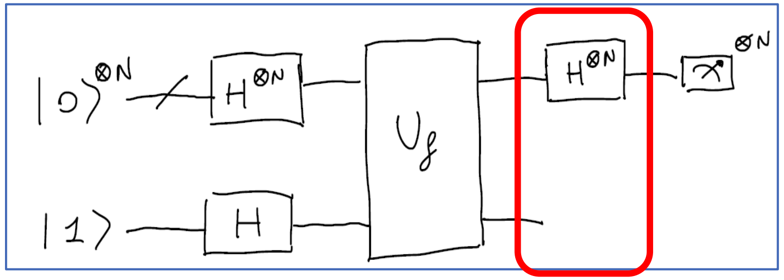
Deutsch Jozsa Algorithm



Step by step analysis

$$\frac{1}{\sqrt{2^N}}\sum_{x}(-1)^{f(x)}|x\rangle\cancel{\left(\frac{|0\rangle-|1\rangle}{\sqrt{2}}\right)}\xrightarrow{H^{\otimes N}}\overbrace{\frac{1}{\sqrt{2^N}}\sum_{y,z}(-1)^{y\cdot z}|y\rangle\langle z|}^{H^{\otimes N}}\frac{1}{\sqrt{2^N}}\sum_{x}(-1)^{f(x)}|x\rangle =$$

Deutsch Jozsa Algorithm

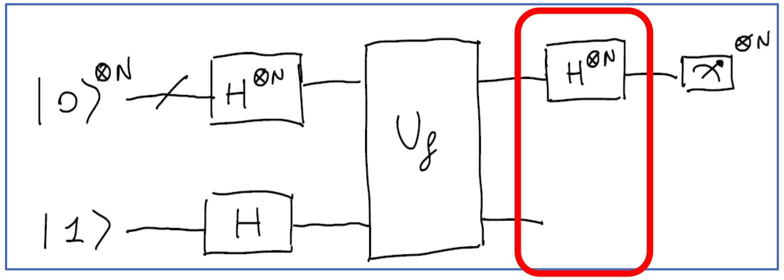


Step by step analysis

$$\frac{1}{\sqrt{2^N}}\sum_{x}(-1)^{f(x)}|x\rangle\cancel{\left(\frac{|0\rangle-|1\rangle}{\sqrt{2}}\right)}\xrightarrow{H^{\otimes N}}\overbrace{\frac{1}{\sqrt{2^N}}\sum_{y,z}(-1)^{y\cdot z}|y\rangle\boxed{\langle z|}}^{H^{\otimes N}}\frac{1}{\sqrt{2^N}}\sum_{x}(-1)^{f(x)}\boxed{|x\rangle}=$$

δ_{zx}

Deutsch Jozsa Algorithm



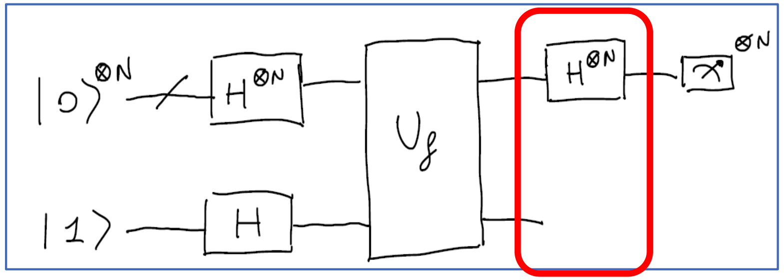
Step by step analysis

$$\frac{1}{\sqrt{2^N}} \sum_x (-1)^{f(x)} |x\rangle \cancel{\left(\frac{|0\rangle - |1\rangle}{\sqrt{2}} \right)} \xrightarrow{H^{\otimes N}} \overbrace{\frac{1}{\sqrt{2^N}} \sum_{y,z} (-1)^{y \cdot z} |y\rangle \boxed{\langle z|}}^{H^{\otimes N}} \frac{1}{\sqrt{2^N}} \sum_x (-1)^{f(x)} \boxed{|x\rangle} =$$

δ_{zx}

$$= \frac{1}{2^N} \sum_{x,y} (-1)^{y \cdot x \oplus f(x)} |y\rangle$$

Deutsch Jozsa Algorithm



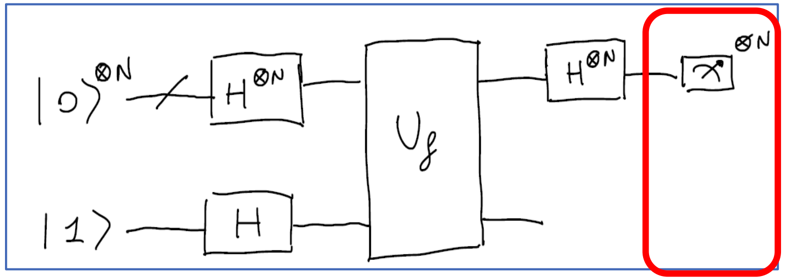
Step by step analysis

$$\frac{1}{\sqrt{2^N}}\sum_x (-1)^{f(x)}|x\rangle \cancel{\left(\frac{|0\rangle-|1\rangle}{\sqrt{2}}\right)} \xrightarrow{H^{\otimes N}} \overbrace{\frac{1}{\sqrt{2^N}}\sum_{y,z}(-1)^{y\cdot z}|y\rangle\boxed{\langle z|}}^{H^{\otimes N}}\ \frac{1}{\sqrt{2^N}}\sum_x(-1)^{f(x)}\boxed{|x\rangle} =$$

δ_{zx}

$$= \frac{1}{2^N}\sum_{x,y}(-1)^{y\cdot x\oplus f(x)}|y\rangle = \sum_y\left[\frac{1}{2^N}\sum_x(-1)^{y\cdot x\oplus f(x)}\right]|y\rangle$$

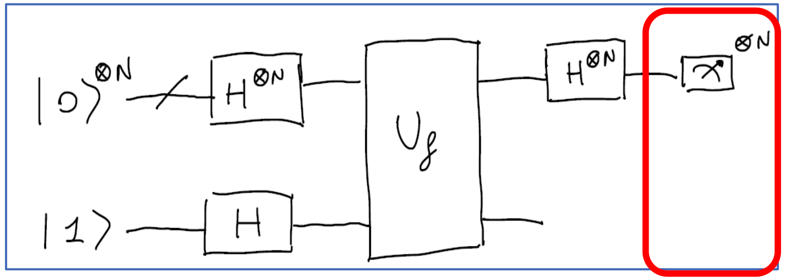
Deutsch Jozsa Algorithm



Step by step analysis

$$\sum_y \left[\frac{1}{2^N} \sum_x (-1)^{y \cdot x \oplus f(x)} \right] |y\rangle$$

Deutsch Jozsa Algorithm

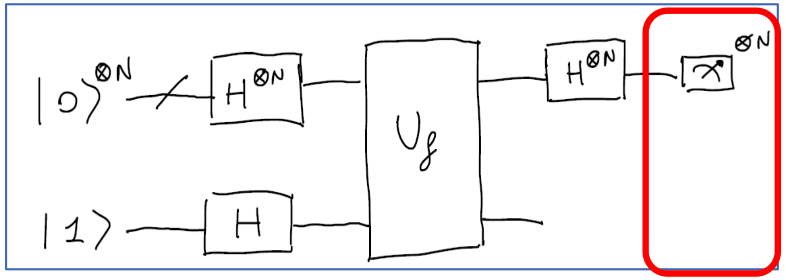


Step by step analysis

$$\sum_y \left[\frac{1}{2^N}\sum_x (-1)^{y\cdot x \oplus f(x)}\right]|y\rangle$$

➡ **Outcome** $y \in \{0,1\}^N$ with $P_r(y) = \left[\frac{1}{2^N}\sum_x (-1)^{y\cdot x \oplus f(x)}\right]^2$

Deutsch Jozsa Algorithm



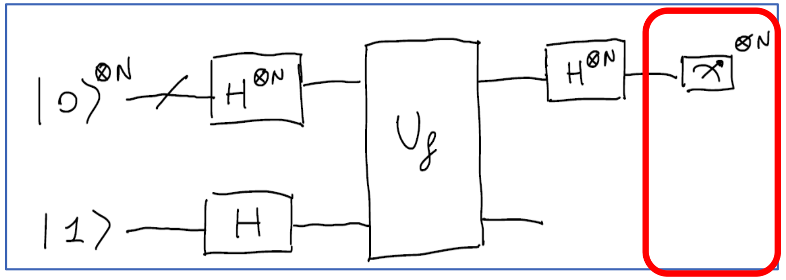
Step by step analysis

$$\sum_y \left[\frac{1}{2^N} \sum_x (-1)^{y \cdot x \oplus f(x)} \right] |y\rangle$$ ➡ **Outcome** $y \in \{0,1\}^N$ with $P_r(y) = \left[\frac{1}{2^N} \sum_x (-1)^{y \cdot x \oplus f(x)} \right]^2$

f constant ➡

(returns 0 on all inputs or 1 on all inputs)

Deutsch Jozsa Algorithm



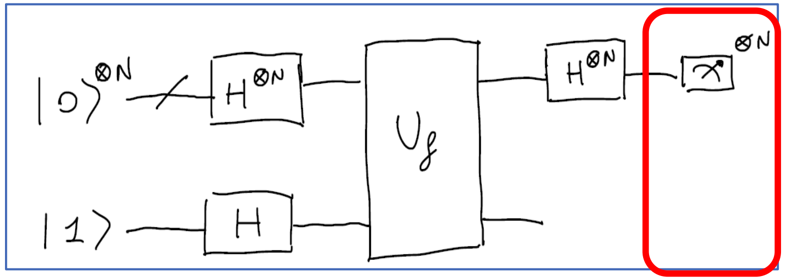
Step by step analysis

$$\sum_y \left[\frac{1}{2^N} \sum_x (-1)^{y \cdot x \oplus f(x)} \right] |y\rangle$$ ➡ **Outcome** $y \in \{0,1\}^N$ with $P_r(y) = \left[\frac{1}{2^N} \sum_x (-1)^{y \cdot x \oplus f(x)} \right]^2$

f constant ➡ $y = (0,0,0\ldots0)$ $\boxed{P_r(y) = \left[\frac{1}{2^N} \sum_x (-1)^{f(x)} \right]^2 = 1}$

(returns 0 on all inputs or 1 on all inputs)

Deutsch Jozsa Algorithm



Step by step analysis

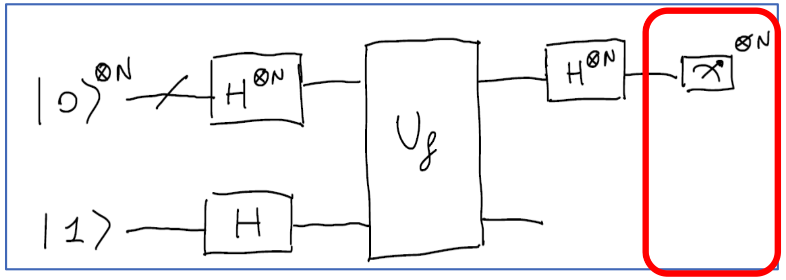
$$\sum_y \left[\frac{1}{2^N}\sum_x (-1)^{y\cdot x \oplus f(x)}\right]|y\rangle$$

➡ **Outcome** $y \in \{0,1\}^N$ with $P_r(y) = \left[\frac{1}{2^N}\sum_x (-1)^{y\cdot x\oplus f(x)}\right]^2$

f balanced ➡

(returns 1 for half of the inputs and 0 for the other half)

Deutsch Jozsa Algorithm



Step by step analysis

$$\sum_y \left[\frac{1}{2^N} \sum_x (-1)^{y \cdot x \oplus f(x)} \right] |y\rangle$$

Outcome $y \in \{0,1\}^N$ with $P_r(y) = \left[\frac{1}{2^N} \sum_x (-1)^{y \cdot x \oplus f(x)} \right]^2$

f balanced

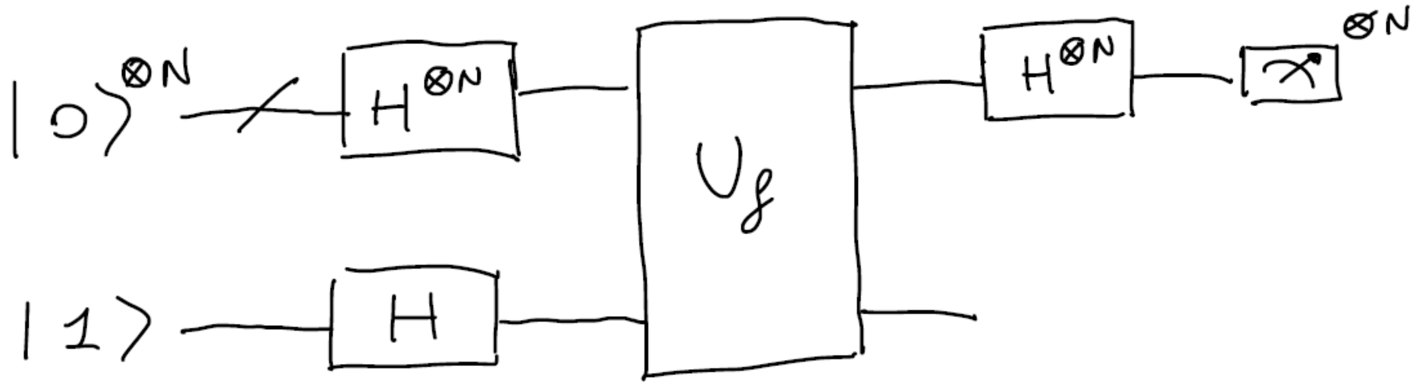
(returns 1 for half of the inputs and 0 for the other half)

$y = (0, 0, 0 \ldots 0)$

$$\boxed{P_r(y) = \left[\frac{1}{2^N} \sum_x (-1)^{f(x)} \right]^2 = 0}$$

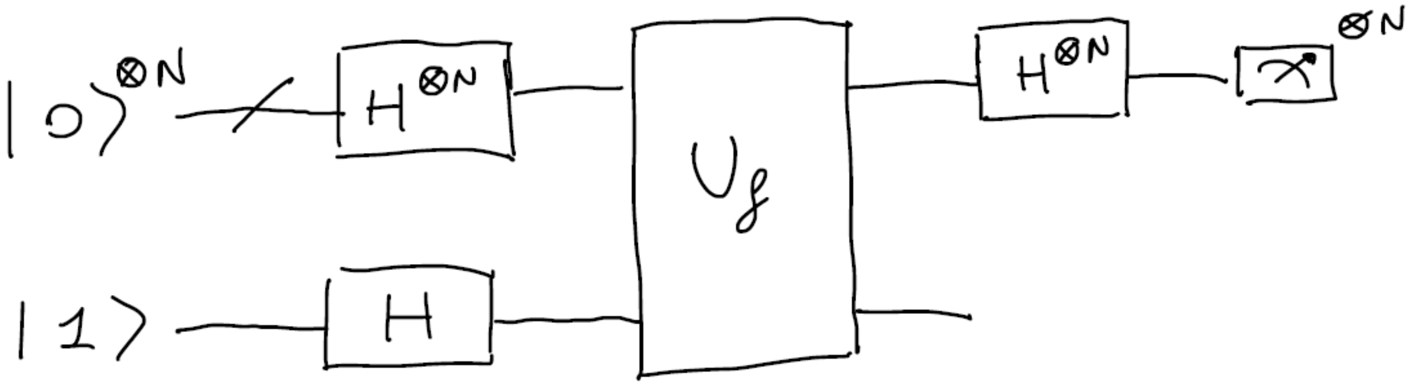
Deutsch Jozsa Algorithm

How many evaluations («queries») of the function are needed to determine with certainty if the function is balanced or constant?



Deutsch Jozsa Algorithm

How many evaluations («queries») of the function are needed to determine with certainty if the function is balanced or constant?



$$\boxed{\text{Quantum Query Complexity} = 1}$$

Classical Query Complexity $\sim 2^{N-1} + 1$

Bernstein Vazirani Algorithm

B-V Problem

Consider a function $f : \{0,1\}^N \to \{0,1\}$ such that

$$f(x) = w \cdot x = (w_1, w_2 \dots w_N) \cdot (x_1, x_2 \dots x_N)$$

The task is to find the string w

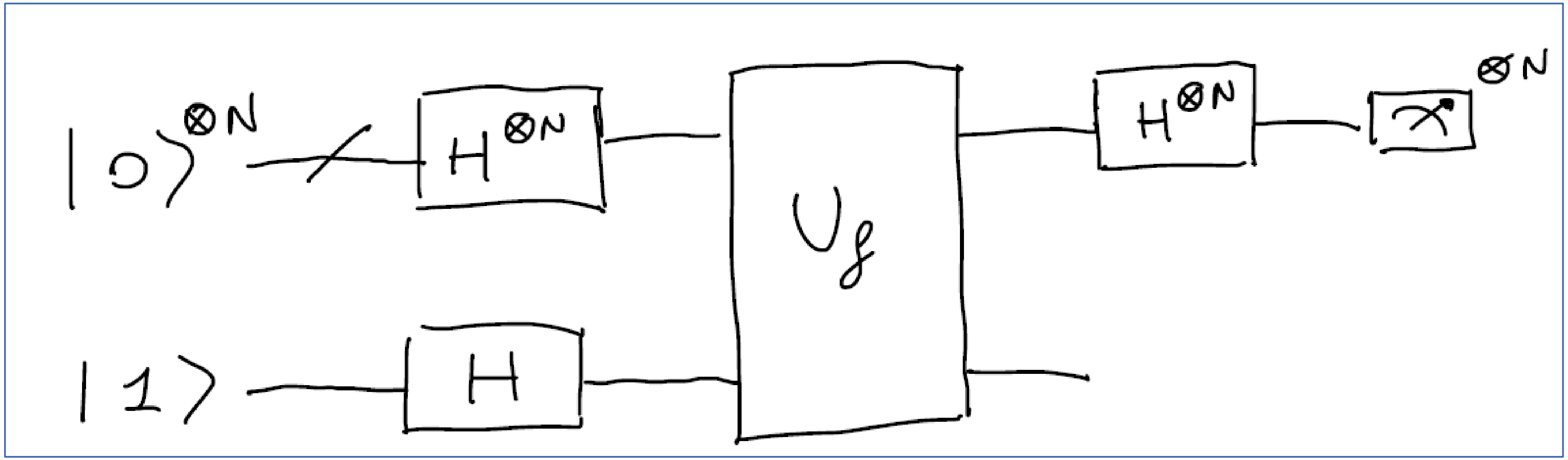
Classical Solution

$$f(x) = w \cdot x = (w_1\, w_2 \ldots w_N) \cdot (x_1\, x_2 \ldots x_N)$$

$$\left.\begin{array}{l}
(w_1\ w_2\ \ldots\ w_N) \cdot (1\ 0\ 0\ \ldots\ 0) \\
(w_1\ w_2\ \ldots\ w_N) \cdot (0\ 1\ 0\ \ldots\ 0) \\
\qquad\qquad\ldots \\
(w_1\ w_2\ \ldots\ w_N) \cdot (0\ 0\ 0\ \ldots\ 1)
\end{array}\right\}$$

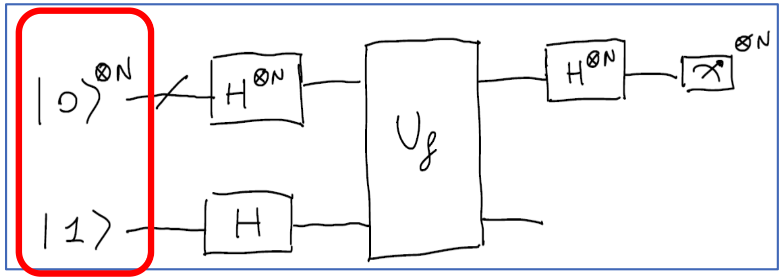
Classically we need **N evaluations** of the function **to recover** w

Quantum Solution (same circuit)



$$f : \{0,1\}^N \to \{0,1\} \quad \text{and} \quad |x\rangle|y\rangle \xrightarrow{U_f} |x\rangle|y \oplus f(x)\rangle$$

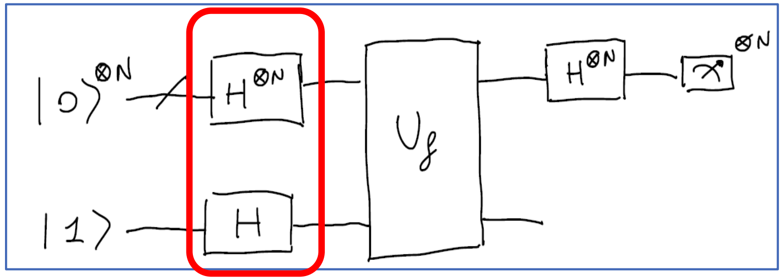
Bernstein Vazirani Algorithm



Step by step analysis

$$|0\rangle^{\otimes N}|1\rangle$$

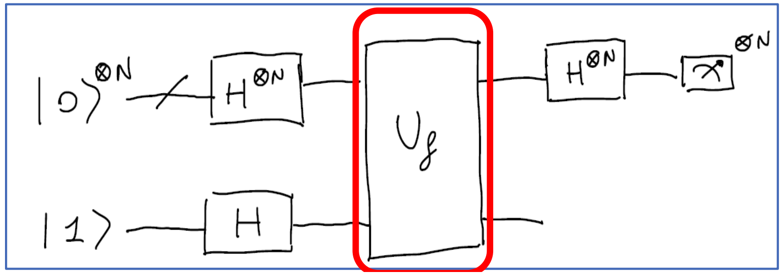
Bernstein Vazirani Algorithm



Step by step analysis

$$|0\rangle^{\otimes N}|1\rangle \xrightarrow{H^{\otimes N} H} \frac{1}{\sqrt{2^N}}\sum_x |x\rangle\left(\frac{|0\rangle-|1\rangle}{\sqrt{2}}\right)$$

Bernstein Vazirani Algorithm

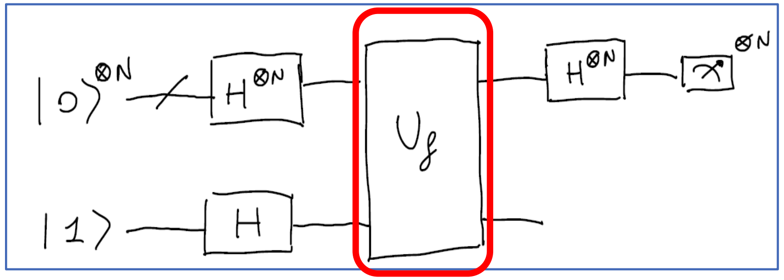


Step by step analysis

$$|0\rangle^{\otimes N}|1\rangle \xrightarrow{H^{\otimes N} H} \frac{1}{\sqrt{2^N}}\sum_x |x\rangle\left(\frac{|0\rangle-|1\rangle}{\sqrt{2}}\right)$$

$$\frac{1}{\sqrt{2^N}}\sum_x |x\rangle\left(\frac{|0\rangle-|1\rangle}{\sqrt{2}}\right) \xrightarrow{U_f} \frac{1}{\sqrt{2^N}}\sum_x |x\rangle\left(\frac{|w\cdot x\rangle-|1\oplus w\cdot x\rangle}{\sqrt{2}}\right)$$

Bernstein Vazirani Algorithm



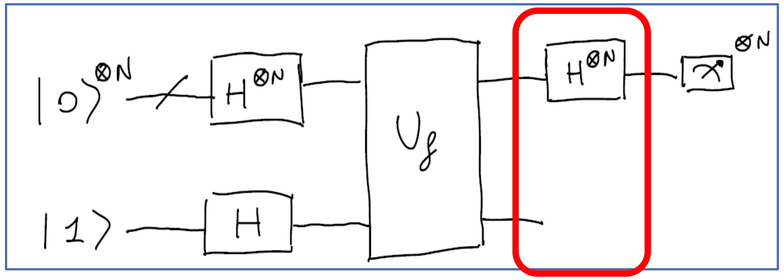
Step by step analysis

$$|0\rangle^{\otimes N}|1\rangle \xrightarrow{H^{\otimes N} H} \frac{1}{\sqrt{2^N}}\sum_x |x\rangle\left(\frac{|0\rangle-|1\rangle}{\sqrt{2}}\right)$$

$$\frac{1}{\sqrt{2^N}}\sum_x |x\rangle\left(\frac{|0\rangle-|1\rangle}{\sqrt{2}}\right) \xrightarrow{U_f} \frac{1}{\sqrt{2^N}}\sum_x |x\rangle\left(\frac{|w\cdot x\rangle-|1\oplus w\cdot x\rangle}{\sqrt{2}}\right)$$

$$= \frac{1}{\sqrt{2^N}}\sum_x (-1)^{w\cdot x}|x\rangle\left(\frac{|0\rangle-|1\rangle}{\sqrt{2}}\right)$$

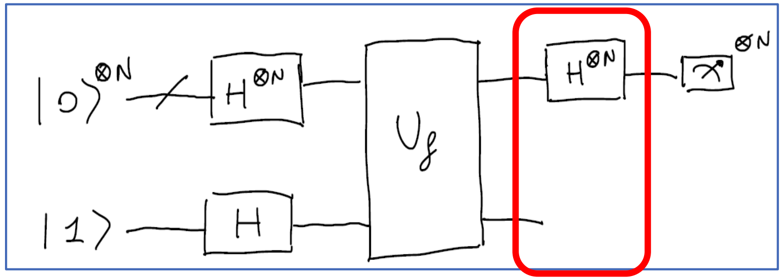
Bernstein Vazirani Algorithm



Step by step analysis

$$\frac{1}{\sqrt{2^N}} \sum_x (-1)^{\omega \cdot x} |x\rangle \left(\frac{|0\rangle - |1\rangle}{\sqrt{2}} \right)$$

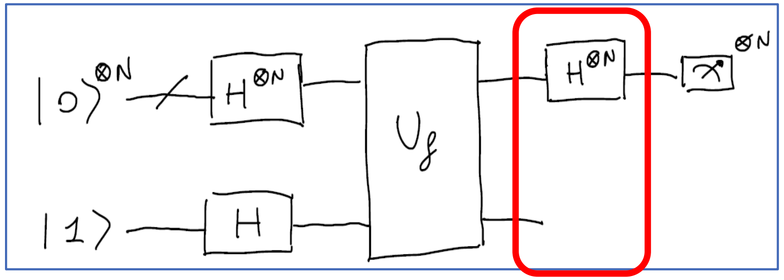
Bernstein Vazirani Algorithm



Step by step analysis

$$\frac{1}{\sqrt{2^N}}\sum_x (-1)^{\omega\cdot x}|x\rangle \cancel{\left(\frac{|0\rangle-|1\rangle}{\sqrt{2}}\right)} \xrightarrow{H^{\otimes N}} \overbrace{\frac{1}{\sqrt{2^N}}\sum_{y,z}(-1)^{y\cdot z}|y\rangle\langle z|}^{H^{\otimes N}} \frac{1}{\sqrt{2^N}}\sum_x(-1)^{\omega\cdot x}|x\rangle =$$

Bernstein Vazirani Algorithm

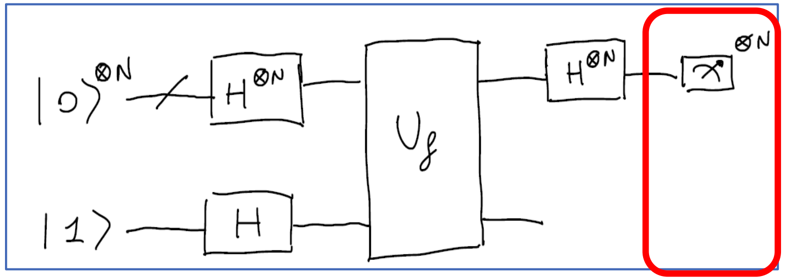


Step by step analysis

$$\frac{1}{\sqrt{2^N}}\sum_x (-1)^{w\cdot x}|x\rangle \cancel{\left(\frac{|0\rangle-|1\rangle}{\sqrt{2}}\right)} \xrightarrow{H^{\otimes N}} \overbrace{\frac{1}{\sqrt{2^N}}\sum_{y,z}(-1)^{y\cdot z}|y\rangle\langle z|}^{H^{\otimes N}}\ \frac{1}{\sqrt{2^N}}\sum_x(-1)^{w\cdot x}|x\rangle =$$

$$= \frac{1}{2^N}\sum_{y,x}(-1)^{y\cdot x\oplus w\cdot x}|y\rangle = \sum_y\left[\frac{1}{2^N}\sum_x(-1)^{y\cdot x\oplus w\cdot x}\right]|y\rangle$$

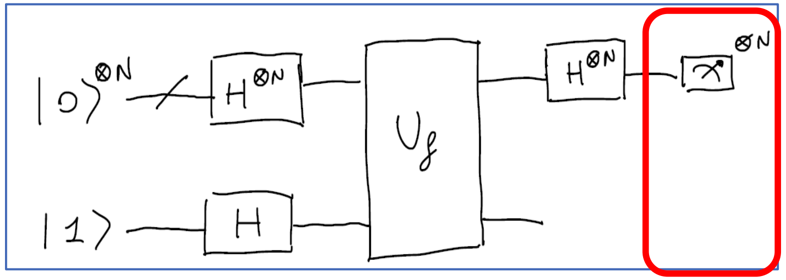
Bernstein Vazirani Algorithm



Step by step analysis

$$\sum_y \left[\frac{1}{2^N} \sum_x (-1)^{y \cdot x \oplus w \cdot x} \right] |y\rangle$$

Bernstein Vazirani Algorithm



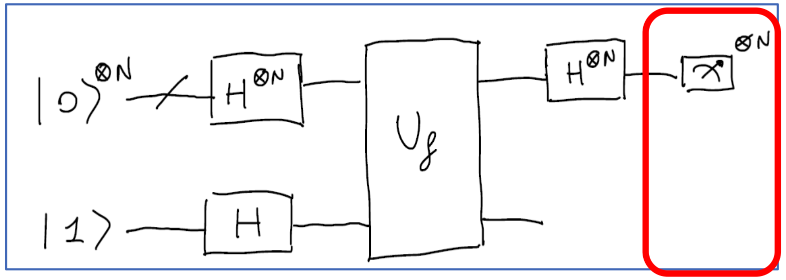
Step by step analysis

$$\sum_y \left[\frac{1}{2^N}\sum_x (-1)^{y\cdot x \oplus w\cdot x}\right]|y\rangle$$

Outcome $|y\rangle = |w\rangle$ with probability

$$P_r(w) = \left(\frac{1}{2^N}\sum_x (-1)^{(w\oplus w)x}\right)^2 = 1$$

Bernstein Vazirani Algorithm



Step by step analysis

$$\sum_y \left[\frac{1}{2^N} \sum_x (-1)^{y\cdot x \oplus w \cdot x} \right] |y\rangle \Rightarrow$$

Outcome $|y\rangle = |w\rangle$ with probability

$$\boxed{P_r(w) = \left(\frac{1}{2^N} \sum_x (-1)^{(w \oplus w) x} \right)^2 = 1}$$

Quantumly we need **1 evaluation** of the function **to recover** *w*

(classically it was N)

Simon Algorithm

Simon Problem

Consider a function $f:\{0,1\}^N \longrightarrow \{0,1\}^N$ such that

$$\exists\, p \in \{0,1\}^N \Rightarrow f(x \oplus p) = f(x) \qquad \forall\, x \in \{0,1\}^N$$

The task is to find the string p

Simon Problem

x	$f(x)$
000	101
001	010
010	000
011	110
100	000
101	110
110	101
111	010

$$p = ?$$

Simon Problem

x	$f(x)$
000	101
001	010
010	000
011	110
100	000
101	110
110	101
111	010

$$p = 110$$

Classical Solution

Consider *M* strings $x^{(1)}, x^{(2)} \dots x^{(M)}$ with $x^{(i)} \in \{0,1\}^N$ and check if

$$f(x^{(i)}) = f(x^{(j)})$$, if so $$x^{(i)} = x^{(j)} \oplus p \rightarrow p = x^{(i)} \oplus x^{(j)}$$

The total number of checks using *M* strings is

$$\frac{M(M-1)}{2}$$

Classical Solution

The probability of finding *p* using *M* strings is hence

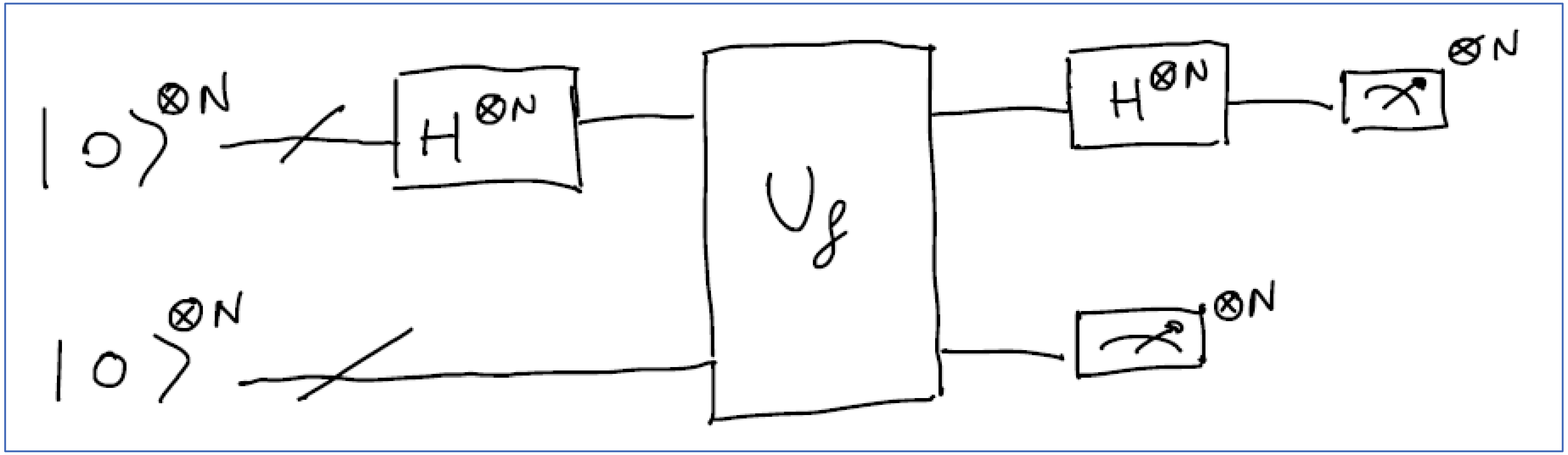
$$Pr(p) = \frac{M(M-1)}{2} / 2^N$$

If we want at least $Pr(p) > \frac{1}{2}$ **this means that**

$$\frac{\frac{M(M-1)}{2}}{2^N} > \frac{1}{2} \sim M > 2^{N/2}$$

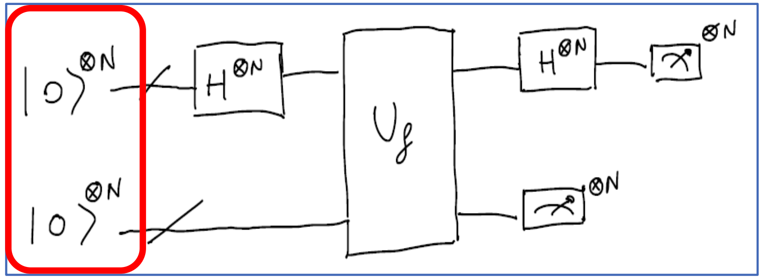
M scales exponentially

Quantum Solution (not the same circuit)



$f : \{0,1\}^N \to \{0,1\}$ and $|x\rangle|y\rangle \xrightarrow{U_f} |x\rangle|y \oplus f(x)\rangle$

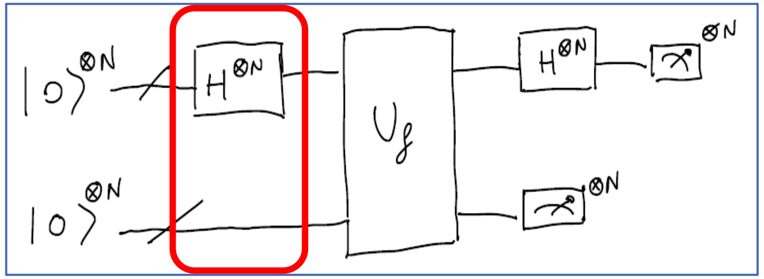
Simon Algorithm



Step by step analysis

$$|0\rangle^{\otimes N}|0\rangle^{\otimes N}$$

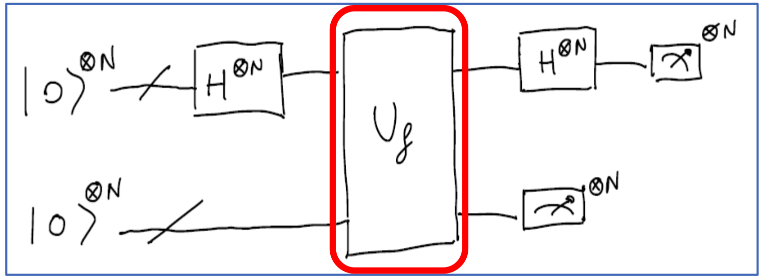
Simon Algorithm



Step by step analysis

$$|0\rangle^{\otimes N}|0\rangle^{\otimes N} \xrightarrow{H^{\otimes N}} \frac{1}{\sqrt{2^N}} \sum_x |x\rangle |0\rangle^{\otimes N}$$

Simon Algorithm

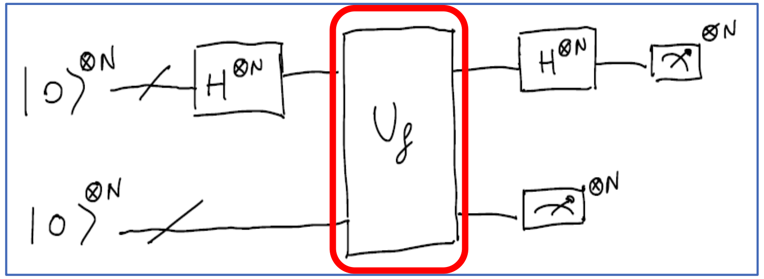


Step by step analysis

$$|0\rangle^{\otimes N}|0\rangle^{\otimes N} \xrightarrow{H^{\otimes N}} \frac{1}{\sqrt{2^N}}\sum_x |x\rangle|0\rangle^{\otimes N}$$

$$\frac{1}{\sqrt{2^N}}\sum_x |x\rangle|0\rangle^{\otimes N} \xrightarrow{U_f} \frac{1}{\sqrt{2^N}}\sum_x |x\rangle|f(x)\rangle$$

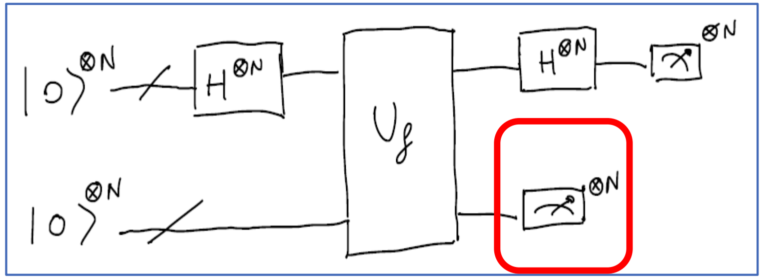
Simon Algorithm



Step by step analysis

$$\frac{1}{\sqrt{2^N}} \sum_x |x\rangle |f(x)\rangle$$

Simon Algorithm



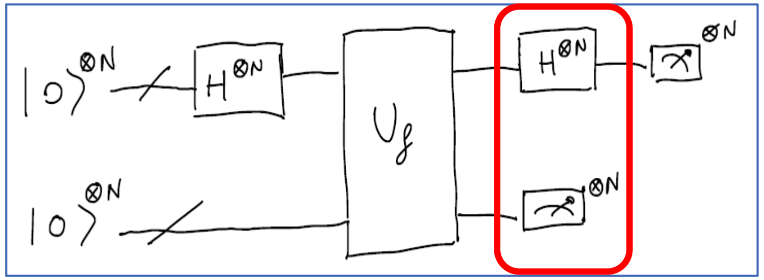
Step by step analysis

$$\frac{1}{\sqrt{2^N}} \sum_x |x\rangle |f(x)\rangle$$ and **measure** the **second register**

Suppose we measure $|f(\tilde{x})\rangle$, the **state after the measurement is**

$$\frac{1}{\sqrt{2}} \left(|\tilde{x}\rangle + |\tilde{x} \oplus p\rangle \right) |f(\tilde{x})\rangle$$

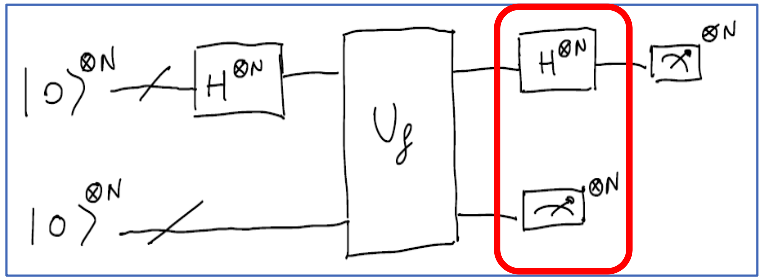
Simon Algorithm



Step by step analysis

$$\frac{1}{\sqrt{2}}\left(|\tilde{x}\rangle + |\tilde{x}\oplus p\rangle\right)\cancel{|f(\tilde{x})\rangle}$$

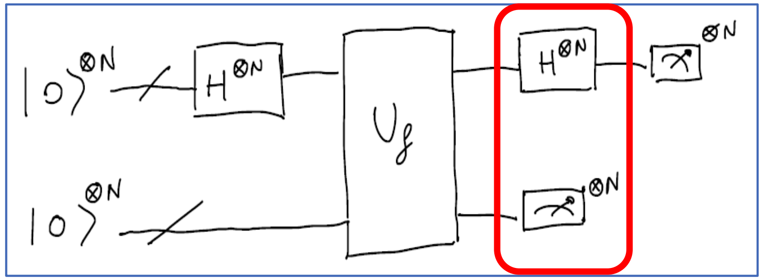
Simon Algorithm



Step by step analysis

$$\frac{1}{\sqrt{2}}\left(|\tilde{x}\rangle + |\tilde{x}\oplus p\rangle\right)\cancel{|f(\tilde{x})\rangle} \xrightarrow{H^{\otimes N}} \overbrace{\frac{1}{\sqrt{2^N}}\sum_{y,z}(-1)^{y\cdot z}|y\rangle\langle z|}^{H^{\otimes N}}\left(\frac{|\tilde{x}\rangle + |\tilde{x}\oplus p\rangle}{\sqrt{2}}\right)$$

Simon Algorithm

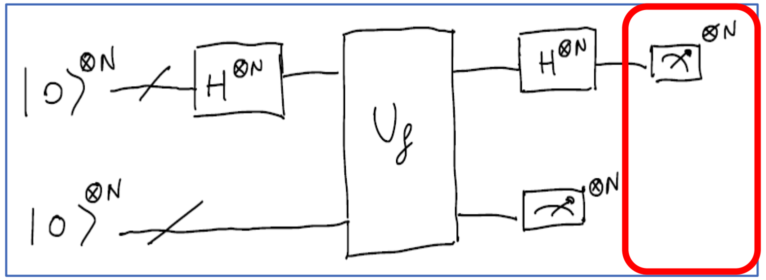


Step by step analysis

$$\frac{1}{\sqrt{2}}\left(|\tilde{x}\rangle + |\tilde{x}\oplus p\rangle\right)\cancel{|f(\tilde{x})\rangle} \xrightarrow{H^{\otimes N}} \overbrace{\frac{1}{\sqrt{2^N}}\sum_{y,z}(-1)^{y\cdot z}|y\rangle\langle z|}^{H^{\otimes N}}\left(\frac{|\tilde{x}\rangle + |\tilde{x}\oplus p\rangle}{\sqrt{2}}\right)$$

$$= \sum_{y}\frac{1}{\sqrt{2^{N+1}}}\left[(-1)^{y\cdot\tilde{x}} + (-1)^{y\cdot(\tilde{x}\oplus p)}\right]|y\rangle$$

Simon Algorithm



Step by step analysis

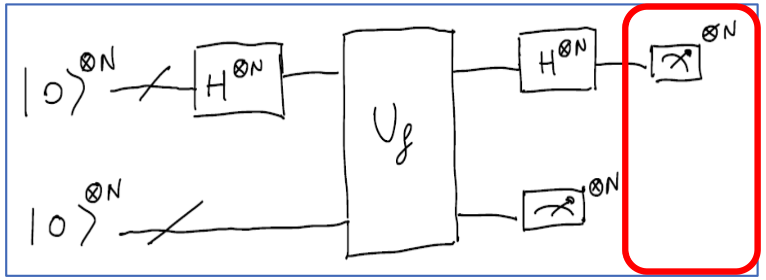
$$\frac{1}{\sqrt{2}}\left(|\tilde{x}\rangle + |\tilde{x}\oplus p\rangle\right)\cancel{|f(\tilde{x})\rangle} \xrightarrow{H^{\otimes N}} \overbrace{\frac{1}{\sqrt{2^N}}\sum_{y,z}(-1)^{y\cdot z}|y\rangle\langle z|}^{H^{\otimes N}}\left(\frac{|\tilde{x}\rangle + |\tilde{x}\oplus p\rangle}{\sqrt{2}}\right)$$

$$= \sum_{y}\frac{1}{\sqrt{2^{N+1}}}\left[(-1)^{y\cdot\tilde{x}} + (-1)^{y\cdot(\tilde{x}\oplus p)}\right]|y\rangle$$

Outcome string *y* with probability

$$\boxed{P_r(y) = \frac{1}{2^{N+1}}\left[(-1)^{y\cdot\tilde{x}} + (-1)^{y\cdot(\tilde{x}\oplus p)}\right]^2}$$

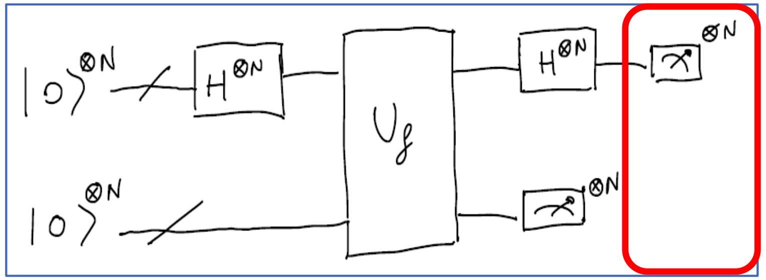
Simon Algorithm



Step by step analysis

$$P_r(y) = \frac{1}{2^{N+1}}\left[(-1)^{y\cdot\tilde{x}} + (-1)^{y\cdot(\tilde{x}\oplus p)}\right]^2$$

Simon Algorithm



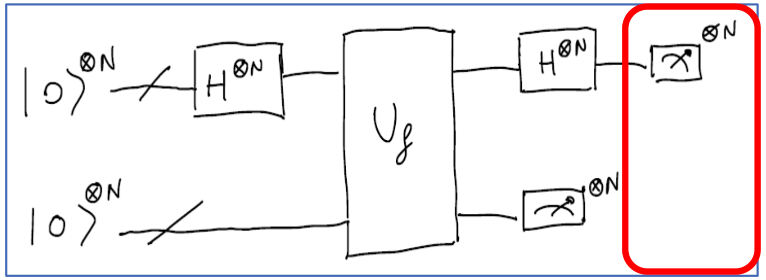
Step by step analysis

$$P_r(y) = \frac{1}{2^{N+1}}\left[(-1)^{y\cdot\tilde{x}} + (-1)^{y\cdot(\tilde{x}\oplus p)}\right]^2$$

If $p\cdot y = 1$ we get

$$P_r(y) = \frac{1}{2^{N+1}}\left[(-1)^{y\cdot\tilde{x}} - (-1)^{y\cdot\tilde{x}}\right]^2 = 0$$

Simon Algorithm



Step by step analysis

$$P_r(y) = \frac{1}{2^{N+1}} \left[(-1)^{y\cdot\tilde{x}} + (-1)^{y\cdot(\tilde{x}\oplus p)} \right]^2$$

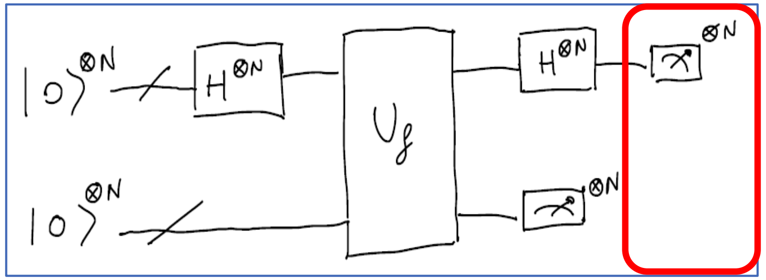
If $p\cdot y = 1$ we get

$$P_r(y) = \frac{1}{2^{N+1}} \left[(-1)^{y\cdot\tilde{x}} - (-1)^{y\cdot\tilde{x}} \right]^2 = 0$$

We always find a string s.t.

$$p\cdot y = 0$$

Simon Algorithm



Step by step analysis

$$P_r(y) = \frac{1}{2^{N+1}}\left[(-1)^{y\cdot\tilde{x}} + (-1)^{y\cdot(\tilde{x}\oplus p)}\right]^2$$

If $p\cdot y = 1$ we get

$$P_r(y) = \frac{1}{2^{N+1}}\left[(-1)^{y\cdot\tilde{x}} - (-1)^{y\cdot\tilde{x}}\right]^2 = 0$$

We always find a string s.t.

$$p\cdot y = 0$$

To recover *p* we need to solve this linear system

$$\begin{cases} p\cdot y^{(1)} = 0 \\ p\cdot y^{(2)} = 0 \\ \vdots \\ p\cdot y^{(N)} = 0 \end{cases}$$

Step by step analysis

$$\begin{cases} p \cdot y^{(1)} = 0 \\ p \cdot y^{(2)} = 0 \\ \vdots \\ p \cdot y^{(N)} = 0 \end{cases}$$

⇨ The **probability** of having $y^{(1)} \; y^{(2)} \; \dots \; y^{(m)}$ **linearly independent** is: $Pr(L.i.) = 1 - \frac{2^m}{2^N}$ with $m < N$

Step by step analysis

$$\begin{cases} p \cdot y^{(1)} = 0 \\ p \cdot y^{(2)} = 0 \\ \vdots \\ p \cdot y^{(N)} = 0 \end{cases}$$

The **probability** of having $y^{(1)}\ y^{(2)} \dots y^{(m)}$ **linearly independent** is: $Pr(L.i.) = 1 - \frac{2^m}{2^N}$ with $m < N$

In order to be sure to find a L.i. set, we have to **repeat the algorithm a number of times equal to**

$$1 < \frac{1}{1 - \frac{2^m}{2^N}} \leq 2$$

Step by step analysis

$$\begin{cases} \varphi \cdot y^{(1)} = 0 \\ \varphi \cdot y^{(2)} = 0 \\ \vdots \\ \varphi \cdot y^{(N)} = 0 \end{cases}$$

The **probability** of having $y^{(1)}\ y^{(2)} \dots y^{(m)}$ **linearly independent** is: $P_{re}(L.i.) = 1 - \frac{2^m}{2^N}$ with $m < N$

In order to be sure to find a L.i. set, we have to **repeat the algorithm a number of times equal to**

$$1 < \frac{1}{1 - \frac{2^m}{2^N}} \leq 2$$

Complexity of the Simon Algorithm scales like 2N (classically it was $2^{N/2}$)

Quantum Fourier Transform

Discrete Fourier Transform

Given a function $f : G \to \mathbb{C}$, the DFT is defined as

$$\boxed{\tilde{f}(g_K) = \frac{1}{\sqrt{N}} \sum_{J=0}^{N-1} \chi_K(g_J)\, f(g_J)}$$

where $\chi_K(g_J) = e^{2\pi i \frac{KJ}{N}}$

Quantum Fourier Transform

Given a basis state $|g_J\rangle$, the QFT is defined as

$$\boxed{|g_J\rangle \xrightarrow{QFT} \frac{1}{\sqrt{N}} \sum_{K=0}^{N-1} \chi_K(g_J)\,|g_K\rangle}$$

where $\chi_K(g_J) = e^{2\pi i \frac{KJ}{N}}$

Quantum Fourier Transform

Given a state $|\psi\rangle = \sum_{J=0}^{N-1} f(g_J)|g_J\rangle$, the QFT is defined as

$$\boxed{|\psi\rangle = \sum_{J=0}^{N-1} f(g_J)|g_J\rangle \xrightarrow{QFT} \sum_{J=0}^{N-1} f(g_J) \frac{1}{\sqrt{N}} \sum_{K=0}^{N-1} \chi_K(g_J)|g_K\rangle}$$

where $\chi_K(g_J) = e^{2\pi i \frac{KJ}{N}}$

Quantum Fourier Transform

Suppose $J \in \{0 \dots 2^N - 1\}$ i.e. the dimension of the space is 2^N

The QFT in this case becomes

$$|J\rangle \xrightarrow{QFT} \frac{1}{\sqrt{2^N}} \sum_{K=0}^{2^N-1} e^{2\pi i \frac{KJ}{2^N}} |K\rangle$$

Is it possible to realize such transformation efficiently on a Quantum Computer?

QFT Circuit

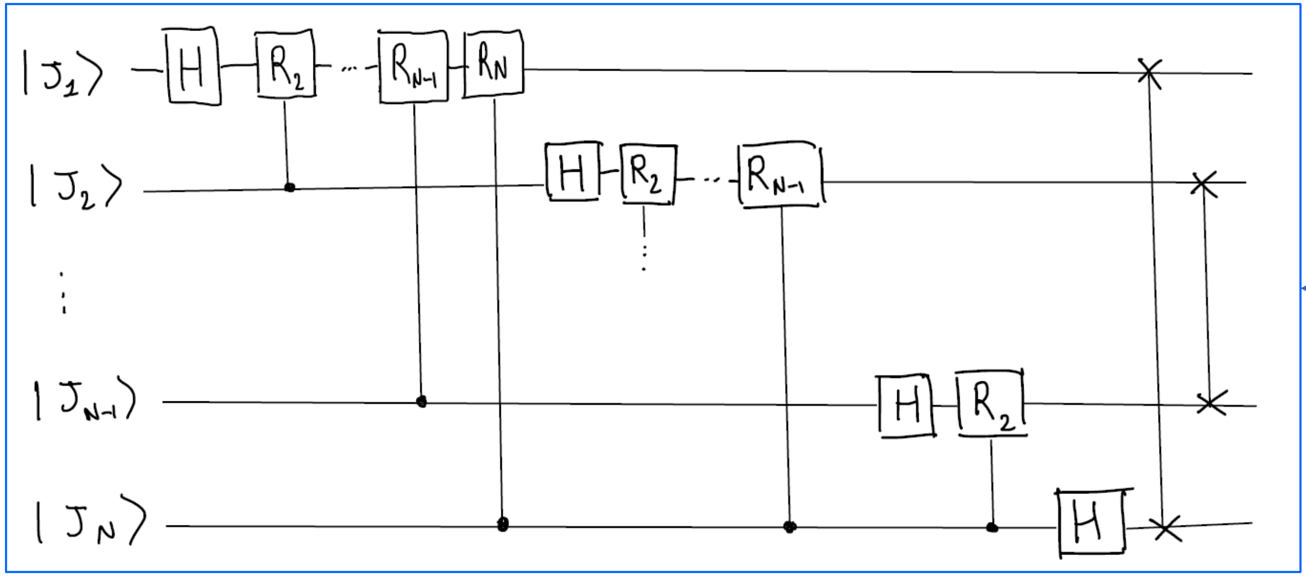
It is possible to rewrite the previus equation as follows

$$|J\rangle \xrightarrow{QFT} \frac{1}{\sqrt{2^N}} \sum_{K=0}^{2^N-1} e^{2\pi i \frac{KJ}{2^N}} |K\rangle = \boxed{\frac{1}{\sqrt{2^N}} \bigotimes_{L=1}^{N} \left(|0\rangle + e^{\frac{2\pi i J}{2^L}} |1\rangle \right)}$$

QFT Circuit

It is possible to rewrite the previus equation as follows

$$|J\rangle \xrightarrow{QFT} \frac{1}{\sqrt{2^N}} \sum_{K=0}^{2^N-1} e^{2\pi i \frac{KJ}{2^N}} |K\rangle = \boxed{\frac{1}{\sqrt{2^N}} \bigotimes_{L=1}^{N} \left(|0\rangle + e^{\frac{2\pi i J}{2^L}} |1\rangle\right)}$$



QFT Circuit Proof

Recall that we can write in **binary form** as follows

$$J \in \{0,1 \dots 2^N-1\} \rightarrow J = \sum_{L=1}^{N} J_L \, 2^{N-L} \,, \quad \boxed{K \in \{0,1 \dots 2^N-1\} \rightarrow K = \sum_{L=1}^{N} K_L \, 2^{N-L}}$$

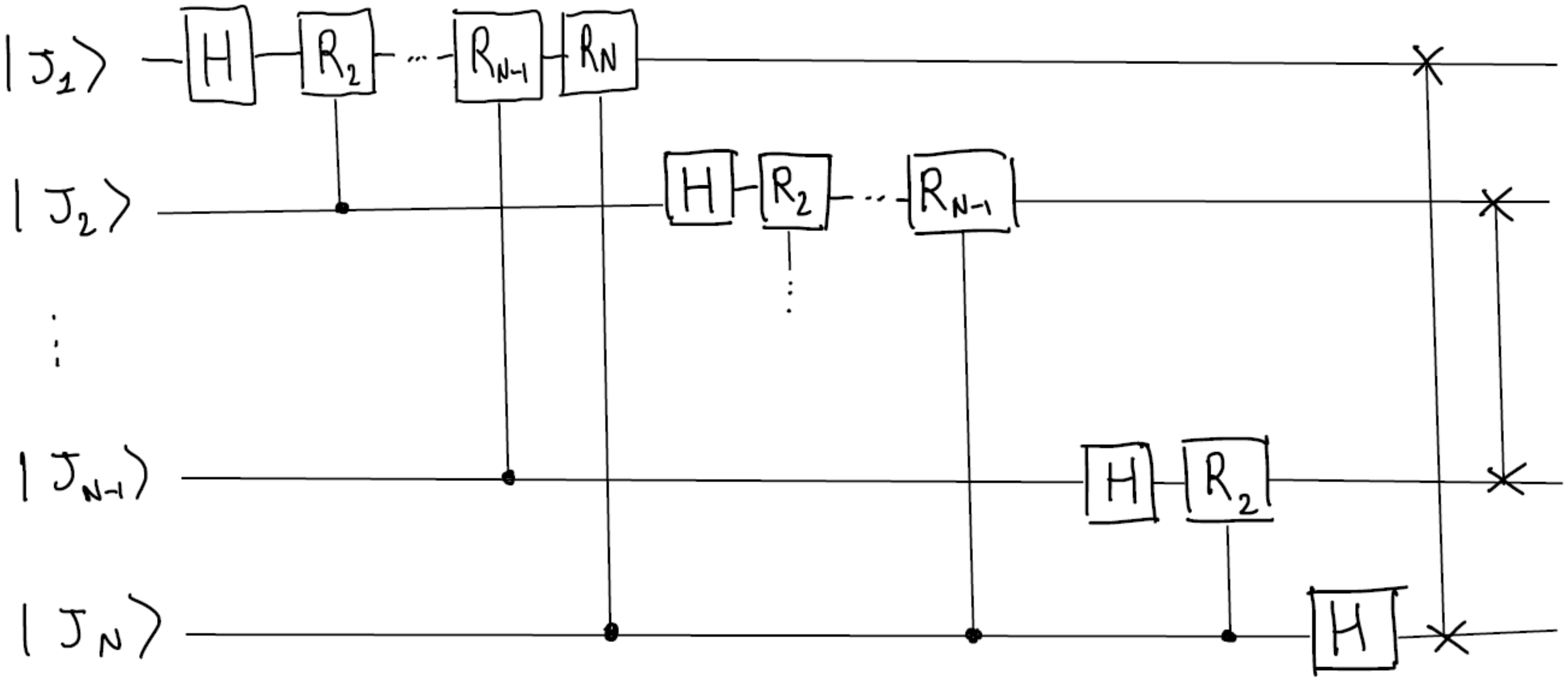
$$|J\rangle \xrightarrow{QFT} \frac{1}{\sqrt{2^N}} \sum_{K=0}^{2^N-1} e^{2\pi i \frac{KJ}{2^N}} |K\rangle = \frac{1}{\sqrt{2^N}} \sum_{K_1=0}^{1} \dots \sum_{K_N=0}^{1} e^{2\pi i J \frac{\sum_{L=1}^{N} K_L 2^{N-L}}{2^N}} |K_1 K_2 \dots K_N\rangle =$$

$$= \frac{1}{\sqrt{2^N}} \sum_{K_1=0}^{1} \dots \sum_{K_N=0}^{1} \bigotimes_{L=1}^{N} e^{2\pi i J \frac{K_L}{2^L}} |K_L\rangle = \frac{1}{\sqrt{2^N}} \bigotimes_{L=1}^{N} \sum_{K_L=0}^{1} e^{2\pi i J \frac{K_L}{2^L}} |K_L\rangle =$$

$$\boxed{= \frac{1}{\sqrt{2^N}} \bigotimes_{L=1}^{N} \left(|0\rangle + e^{\frac{2\pi i J}{2^L}} |1\rangle \right)}$$

Quantum Fourier Transform

QFT Circuit



$$J \in \{0, 1 \dots 2^N - 1\}$$

$$\downarrow$$

$$J = \sum_{L=1}^{N} J_L \, 2^{N-L}$$

$$\boxed{R_K} = U_{\phi = \frac{2\pi}{2^K}} = \begin{pmatrix} 1 & 0 \\ 0 & e^{i 2\pi / 2^K} \end{pmatrix}$$

$\boxed{H}$ $\boxed{SWAP}$

Complexity: $N \boxed{H} + \frac{N}{2} \boxed{SWAP} + \frac{N(N-1)}{2} \boxed{R_K} \longrightarrow O(N^2)$

Quantum Phase Estimation

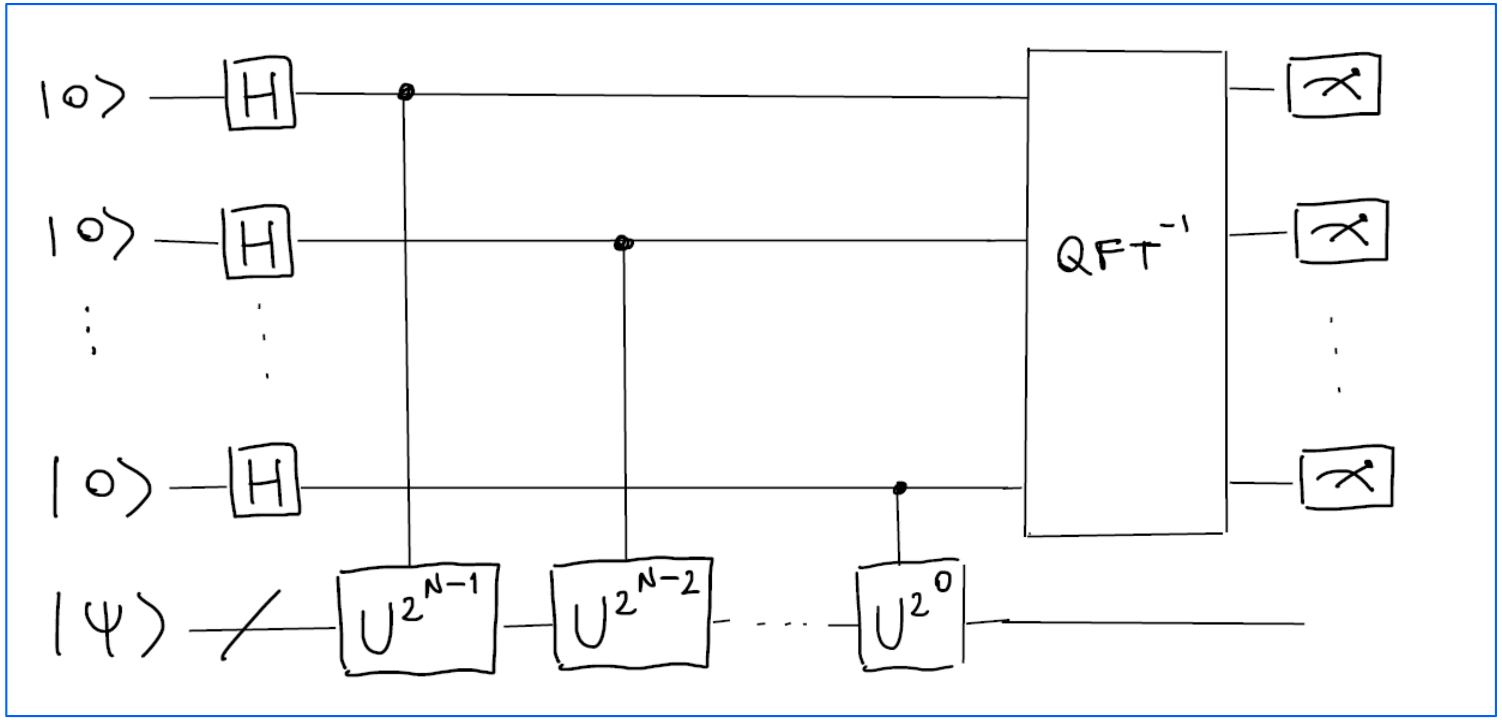
QPE problem

Given a Unitary U and a quantum state $|\psi\rangle$ such that

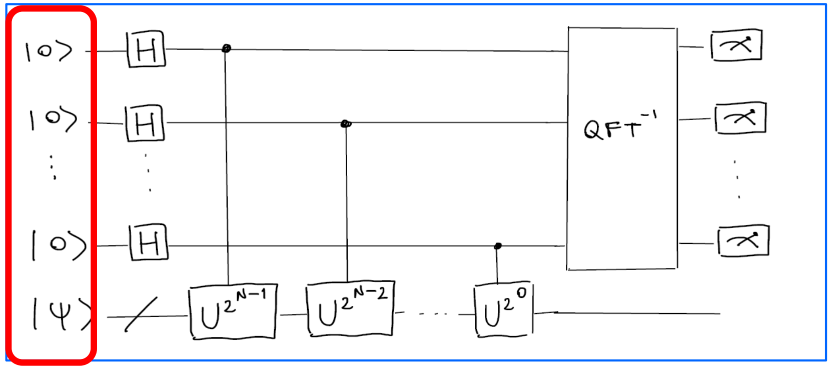
$$\boxed{U|\psi\rangle = e^{2\pi i \theta}|\psi\rangle}$$

The task is to estimate θ

QPE circuit



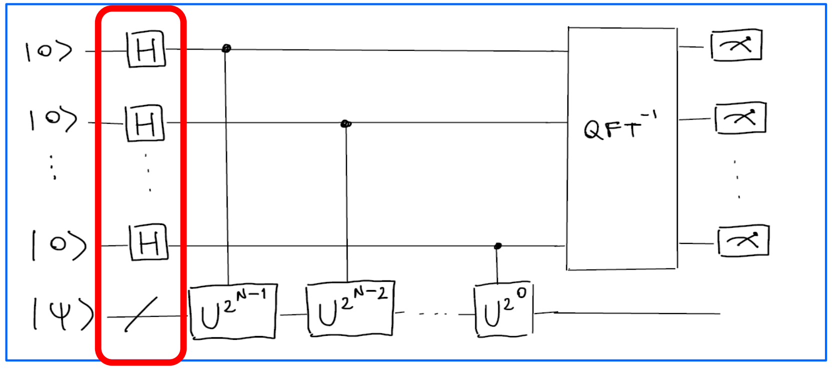
Quantum Phase Estimation



QPE circuit analysis

$$|\psi_0\rangle = |0\rangle^{\otimes N} |\psi\rangle$$

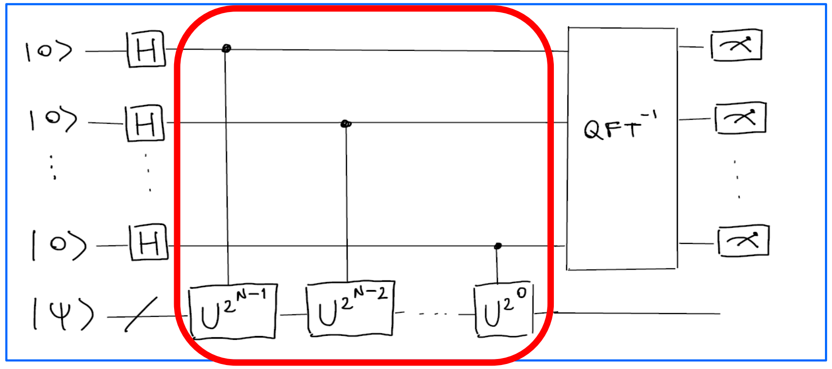
Quantum Phase Estimation



QPE circuit analysis

$$|\psi_0\rangle = |0\rangle^{\otimes N} |\psi\rangle \longrightarrow |\psi_1\rangle = \frac{1}{\sqrt{2^N}}\left(|0\rangle + |1\rangle\right)^{\otimes N} |\psi\rangle$$

Quantum Phase Estimation

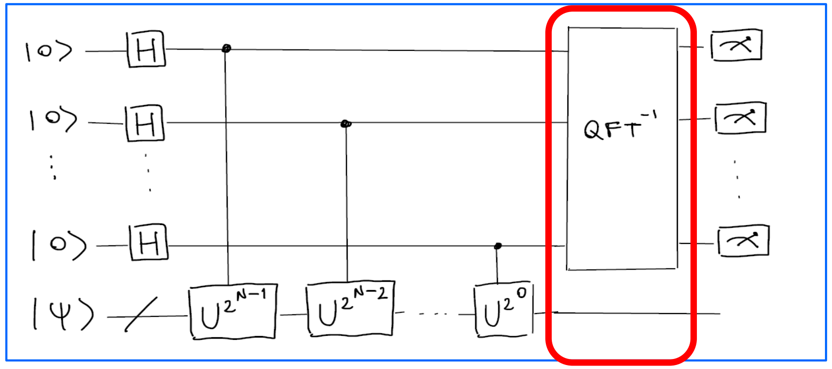


QPE circuit analysis

$$|\psi_0\rangle = |0\rangle^{\otimes N} |\psi\rangle \longrightarrow |\psi_1\rangle = \frac{1}{\sqrt{2^N}} \left(|0\rangle + |1\rangle\right)^{\otimes N} |\psi\rangle$$

$$|\psi_2\rangle = \frac{1}{\sqrt{2^N}} \sum_{k=0}^{2^N-1} e^{2\pi i k \theta} |k\rangle |\psi\rangle$$

Quantum Phase Estimation

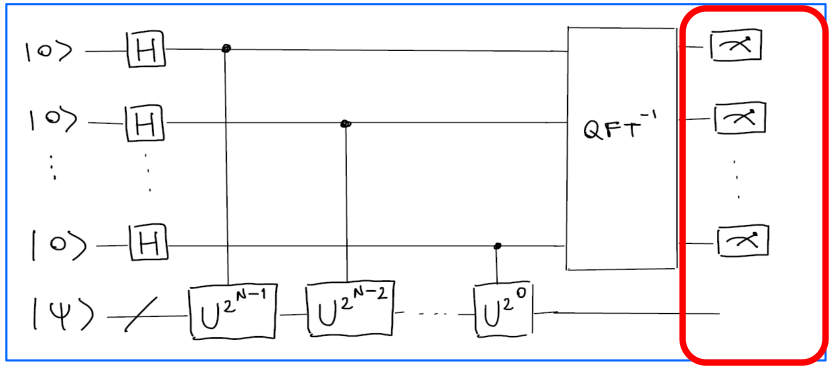


QPE circuit analysis

$$|\Psi_0\rangle = |0\rangle^{\otimes N}|\psi\rangle \longrightarrow |\Psi_1\rangle = \frac{1}{\sqrt{2^N}}\left(|0\rangle + |1\rangle\right)^{\otimes N}|\psi\rangle$$

$$|\Psi_2\rangle = \frac{1}{\sqrt{2^N}}\sum_{k=0}^{2^N-1} e^{2\pi i k\theta}|k\rangle|\psi\rangle$$

$$\boxed{|\Psi_3\rangle = \frac{1}{2^N}\sum_{j=0}^{2^N-1}\sum_{k=0}^{2^N-1} e^{\frac{2\pi i k}{2^N}\left(2^N\theta - j\right)}|j\rangle|\psi\rangle}$$

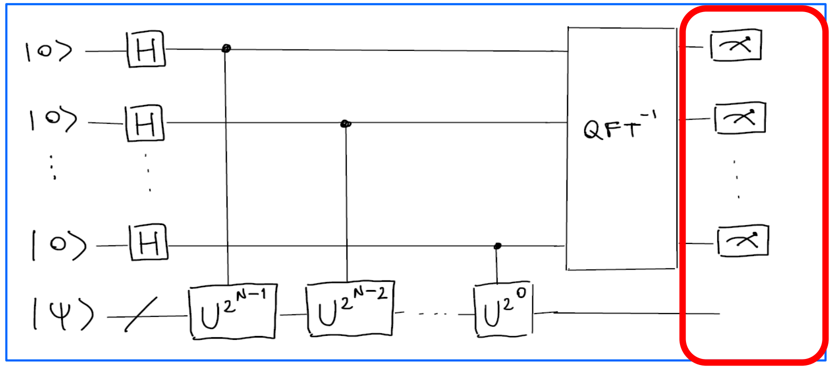


QPE circuit analysis

The probability of measuring *j*

$$|\psi_3\rangle = \frac{1}{2^N}\sum_{j=0}^{2^N-1}\sum_{k=0}^{2^N-1} e^{\frac{2\pi i k}{2^N}(2^N\theta - j)} |j\rangle|\psi\rangle \Rightarrow Pr(j) = \left[\frac{1}{2^N}\sum_{k=0}^{2^N-1} e^{\frac{2\pi i k}{2^N}(2^N\theta - j)}\right]^2$$

Quantum Phase Estimation



QPE circuit analysis

The probability of measuring *j*

$$|\psi_3\rangle = \frac{1}{2^N}\sum_{j=0}^{2^N-1}\sum_{k=0}^{2^N-1} e^{\frac{2\pi i k}{2^N}(2^N\theta - j)}|j\rangle|\psi\rangle \Rightarrow P_r(j) = \left[\frac{1}{2^N}\sum_{k=0}^{2^N-1} e^{\frac{2\pi i k}{2^N}(2^N\theta - j)}\right]^2$$

If $j = 2^N\theta$ the probability becomes $P_r(j = 2^N\theta) = 1$

State after measurement: $|\psi_4\rangle = |2^N\theta\rangle|\psi\rangle$

Shor Algorithm

Facorization Problem

Given N , find the two prime numbers such that

$$\boxed{N = p \times q}$$

Facorization Problem

Given N , find the two prime numbers such that

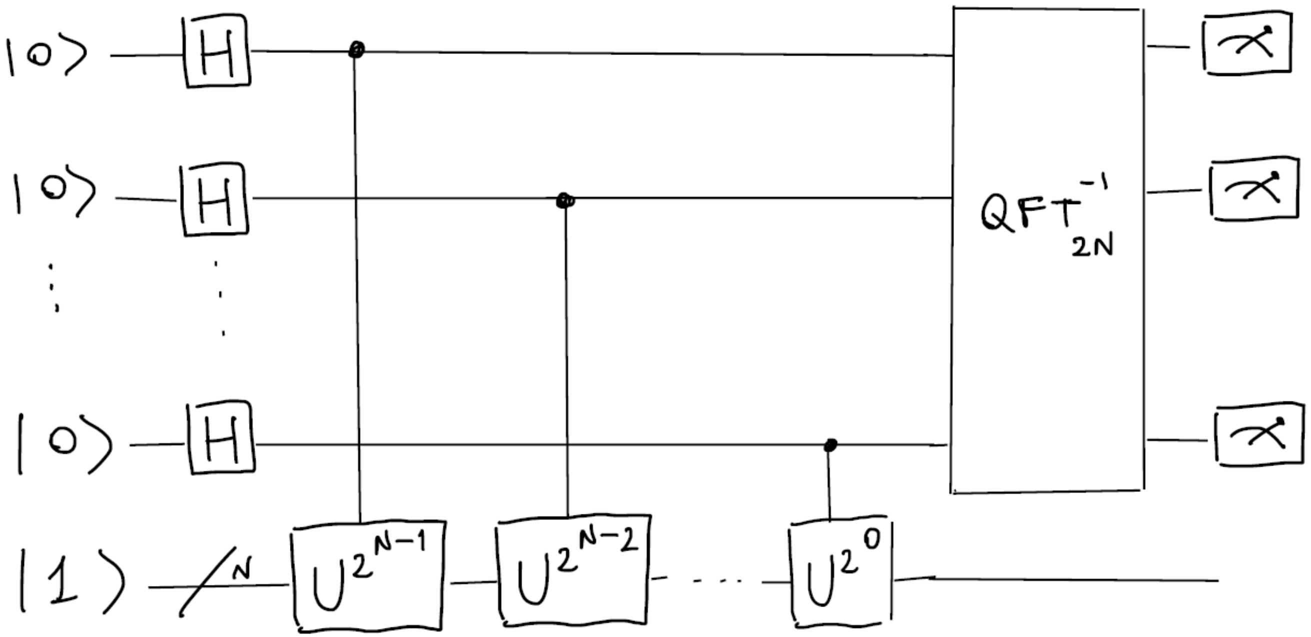
$$\boxed{N = p \times q}$$

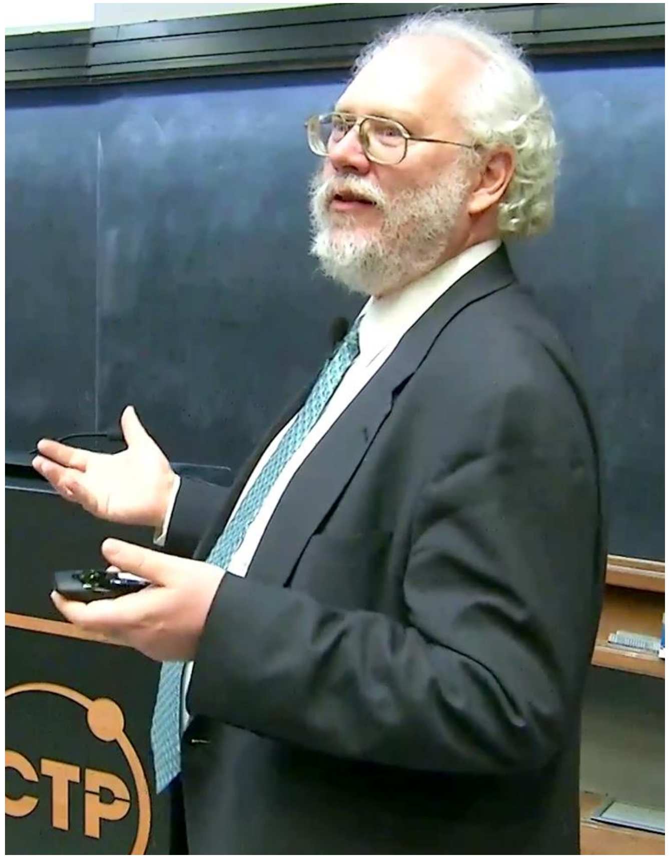
Classically: Finding solution requires **exponential time**

Used in the RSA crypto system

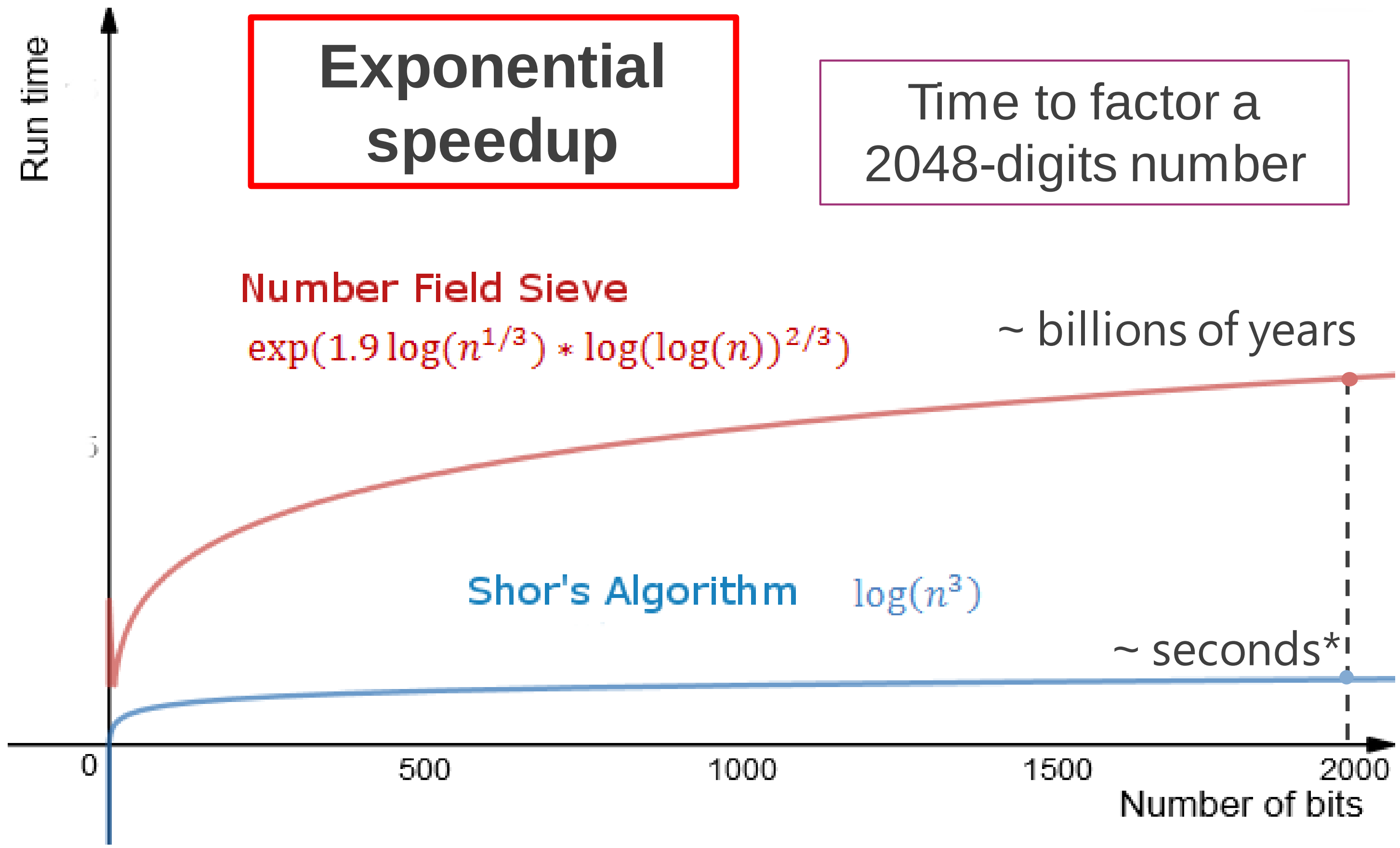
Shor Algorithm

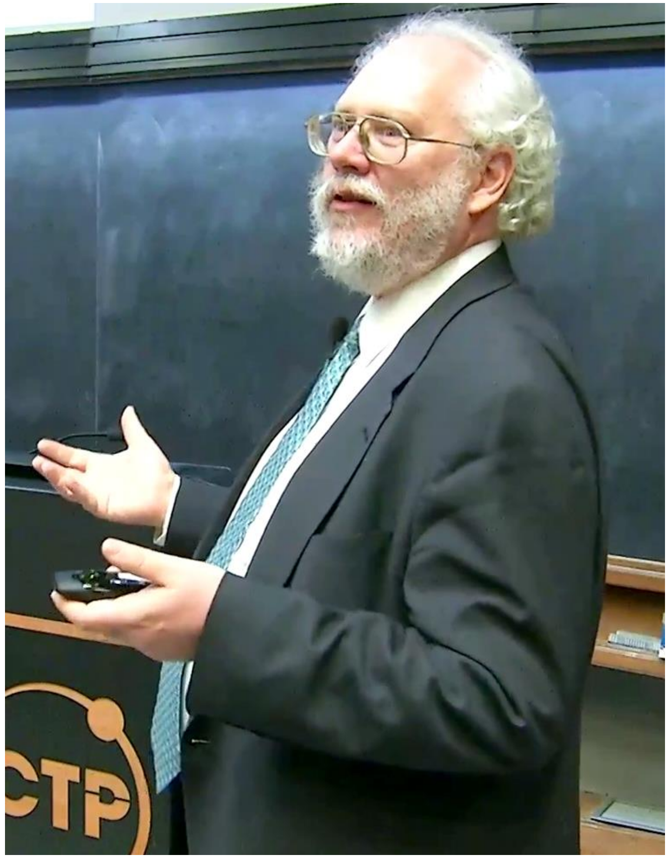
Modified version of QPE to solve factorization in polynomial time





Shor Algorithm





* Assuming we have a fault-tolerant quantum computer capable of executing Shor's algorithm by applying gates at the speed of current quantum computers based on superconducting circuits

Grover Search

Searching Problem

We have access to an unstructured database of 2^N elements, the task is to find the $\tilde{x}$ element

Assume to have a function $f : \{0,1\}^N \rightarrow \{0,1\}$ such that

$$\boxed{f(x) = \begin{cases} 1 & \text{IF } x = \tilde{x} \\ 0 & \text{IF } x \neq \tilde{x} \end{cases}}$$

Searching Problem

We have access to an unstructured database of 2^N elements, the task is to find the $\tilde{x}$ element

Assume to have a function $f : \{0,1\}^N \to \{0,1\}$ such that

$$f(x) = \begin{cases} 1 & \text{IF } x = \tilde{x} \\ 0 & \text{IF } x \neq \tilde{x} \end{cases}$$

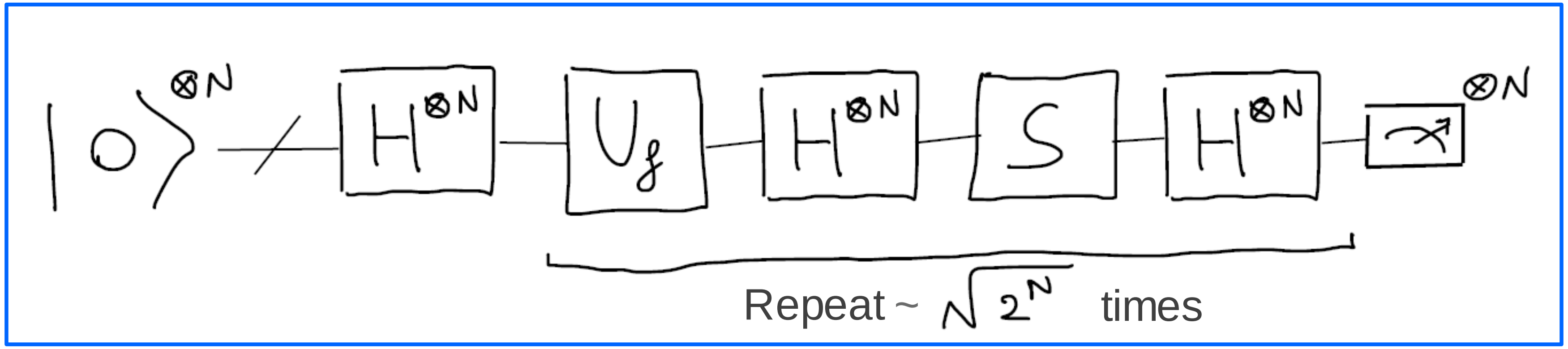
➡ **Classically,** in order to find the searched element**,** we have to evaluate this function on 2^{N-1} inputs (on average)

Grover Algorithm

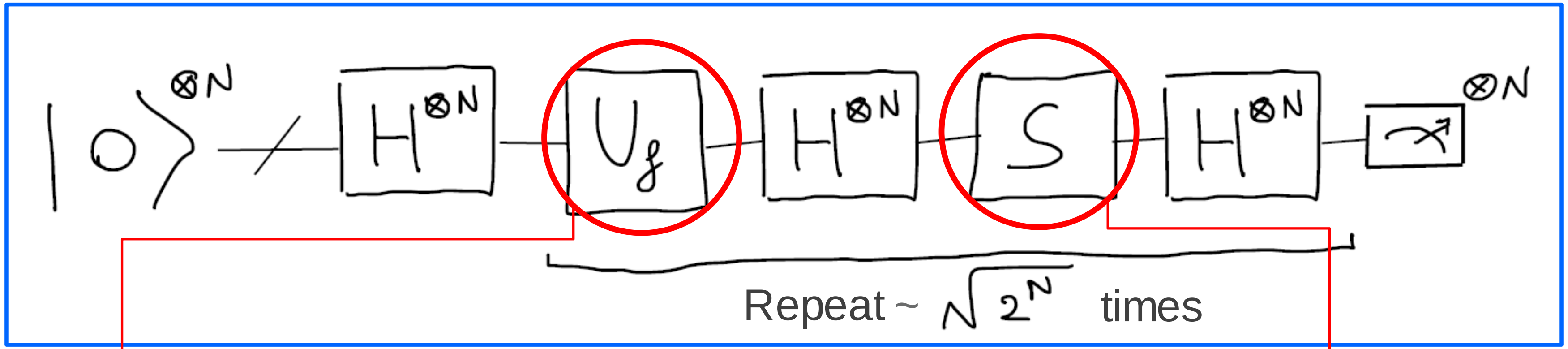
$$f(x) = \begin{cases} 1 & \text{IF } x = \tilde{x} \\ 0 & \text{IF } x \neq \tilde{x} \end{cases} \xrightarrow[\text{the unitary}]{\text{Obtained via}} U_f |x\rangle = \begin{cases} -|x\rangle & \text{IF } x = \tilde{x} \\ |x\rangle & \text{IF } x \neq \tilde{x} \end{cases}$$

$$\boxed{U_f |x\rangle = (-1)^{f(x)} |x\rangle}$$

Grover Algorithm



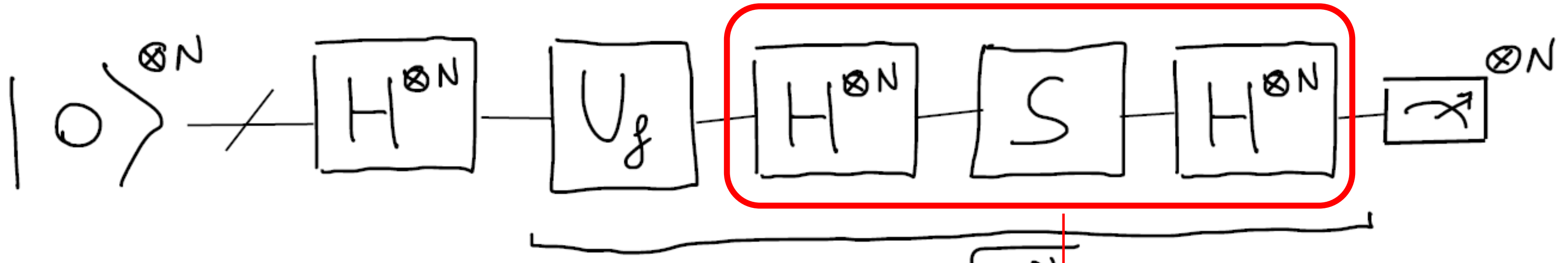
Grover Algorithm



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

$$S = 2|0\rangle^{\otimes N}\langle 0|^{\otimes N} - I$$

Grover Algorithm



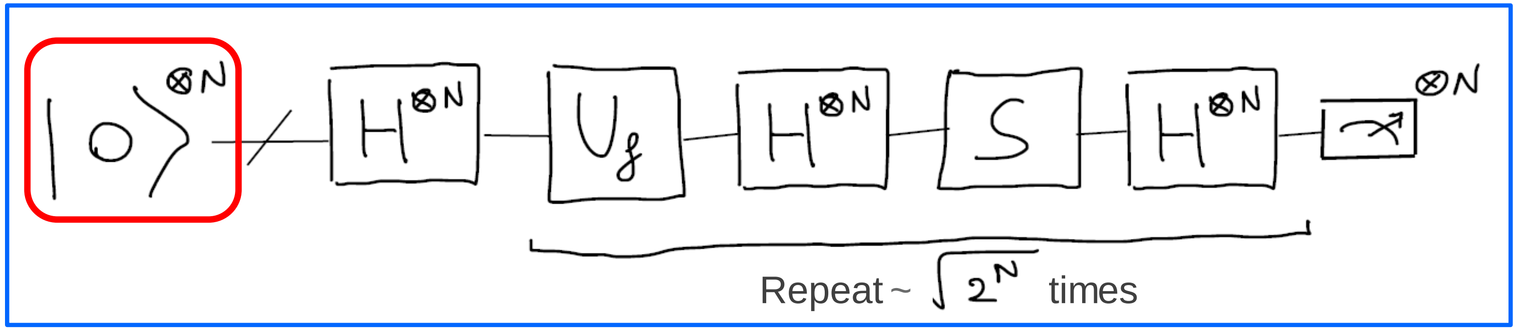
Repeat ~ $\sqrt{2^N}$ times

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

$$S = 2|0\rangle^{\otimes N}\langle 0|^{\otimes N} - I$$

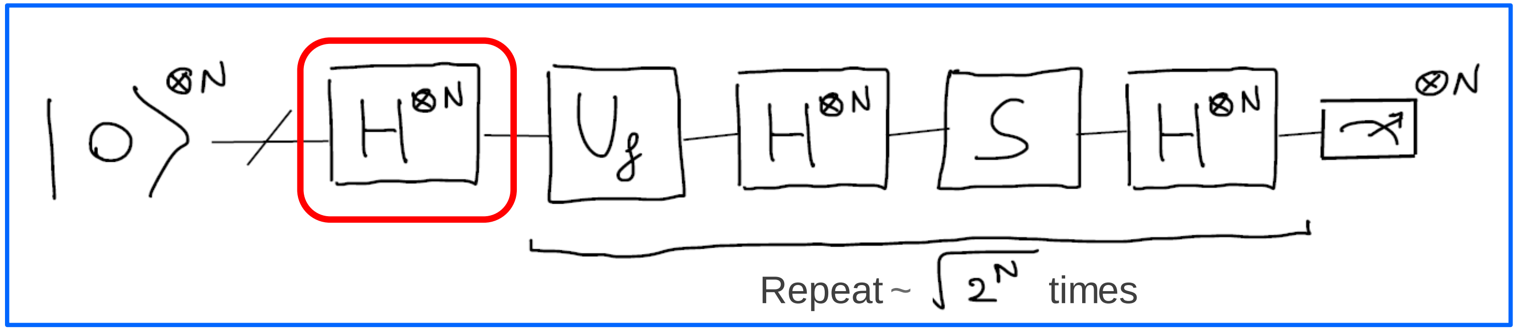
$$\boxed{U_s = H^{\otimes N}\left(2|0\rangle^{\otimes N}\langle 0|^{\otimes N} - I\right)H^{\otimes N} = 2|s\rangle\langle s| - I}$$

Grover Search

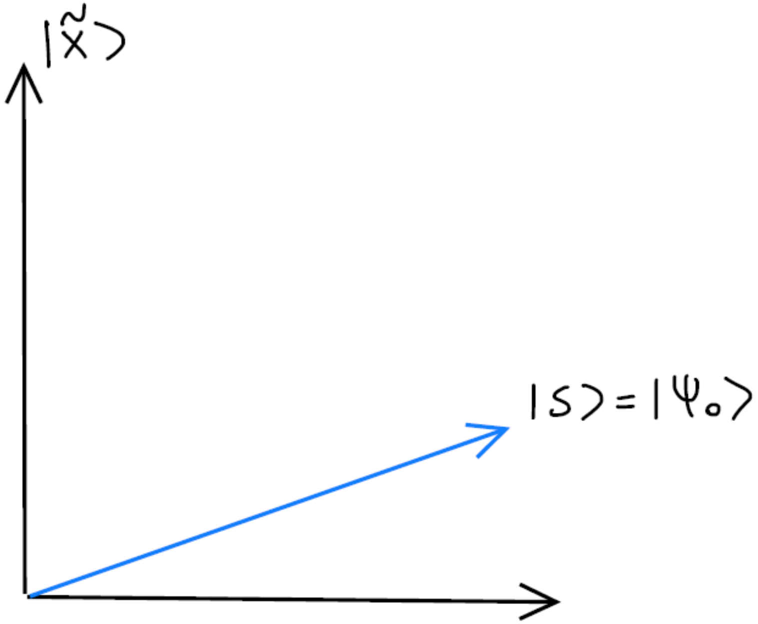


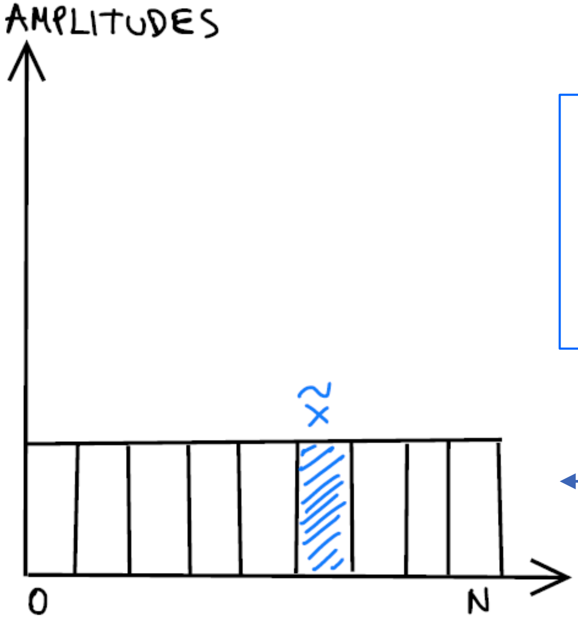
Grover Algorithm: geometrical analysis

Grover Search



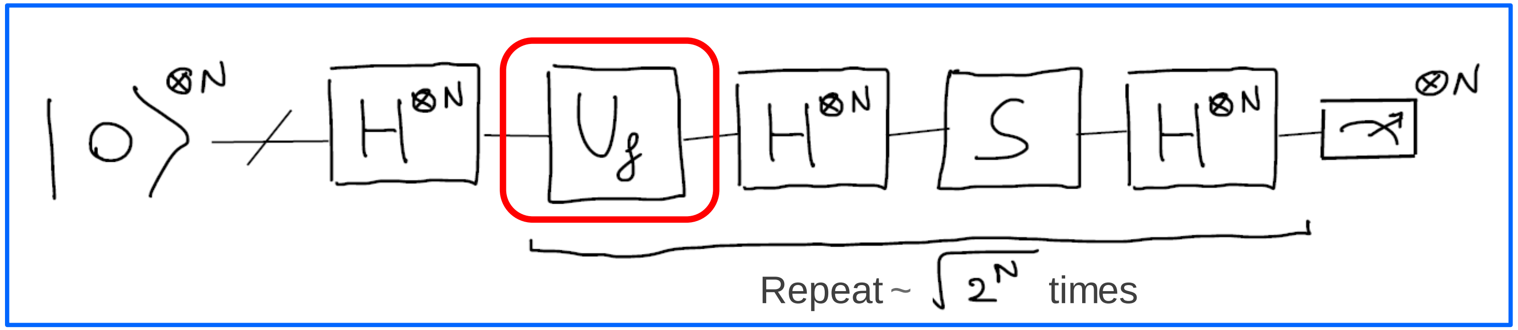
Grover Algorithm: geometrical analysis



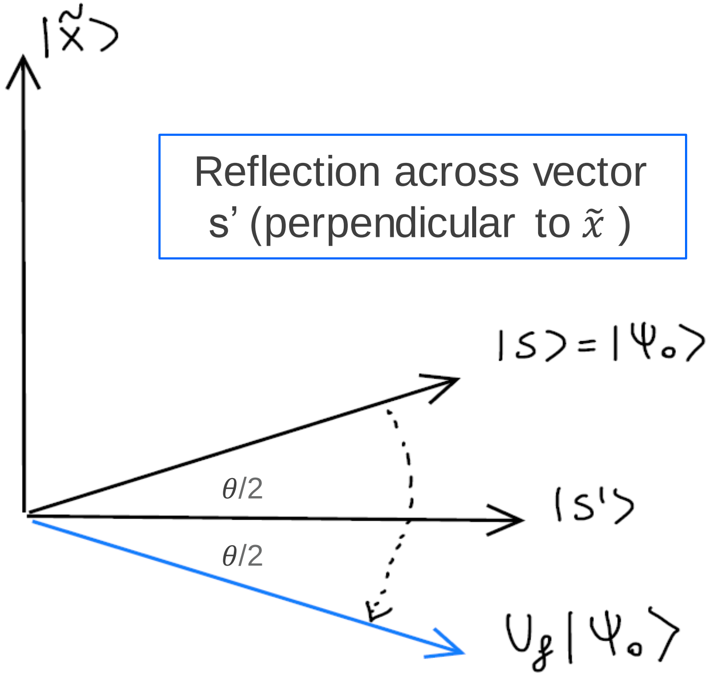


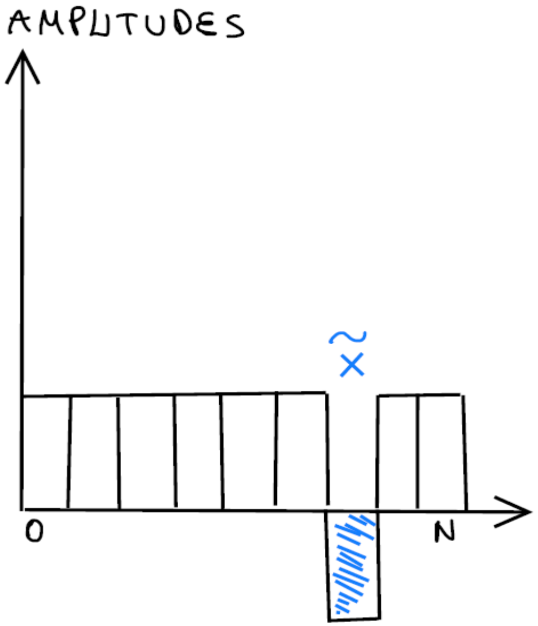
Equal superposition of all possible elements

Grover Search



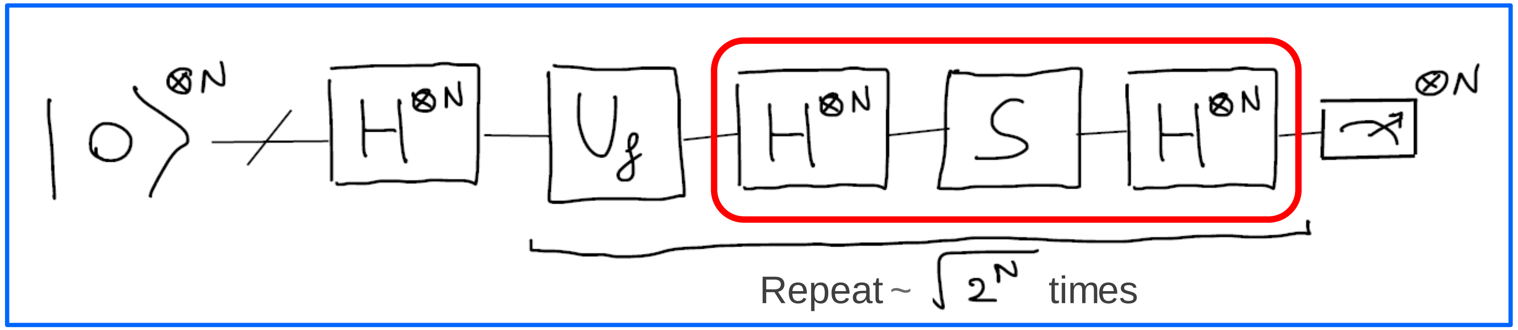
Grover Algorithm: geometrical analysis



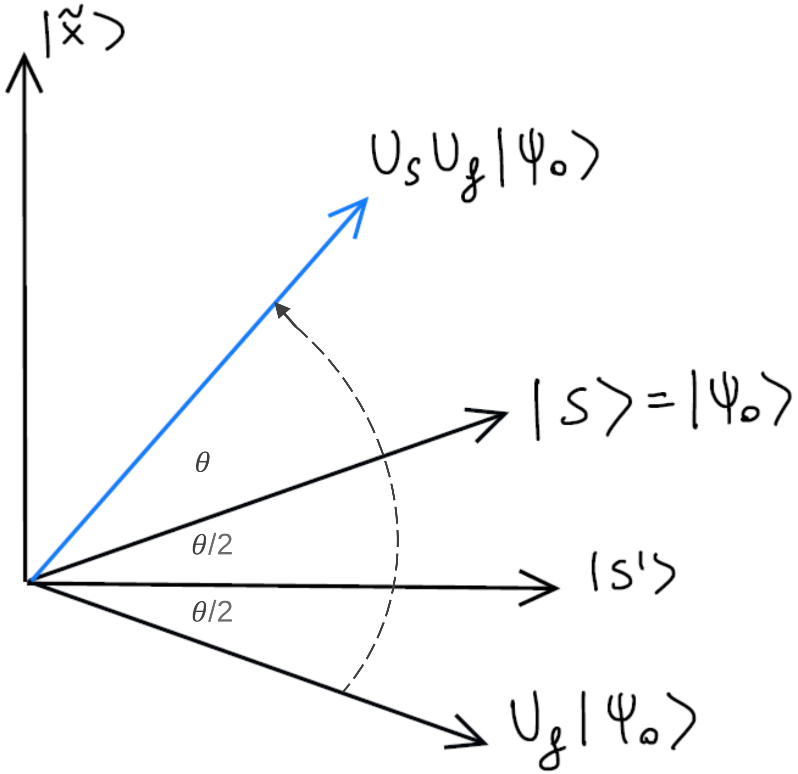


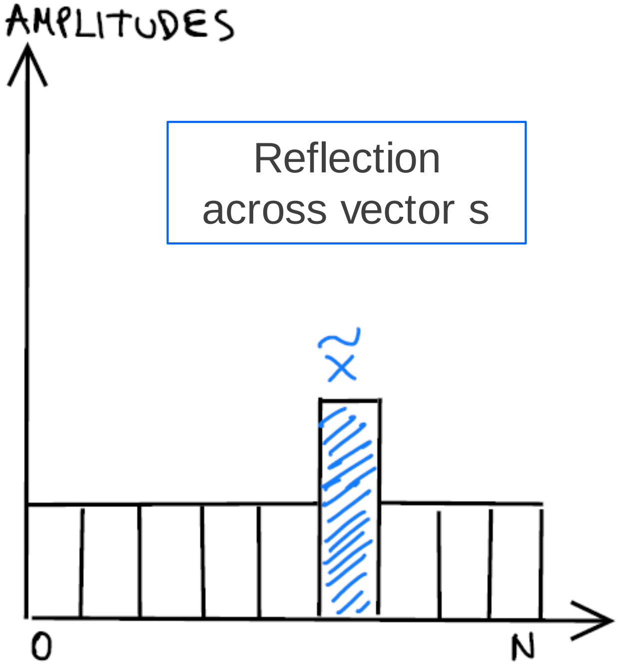
Amplitude of the searched element becomes negative

Grover Search



Grover Algorithm: geometrical analysis

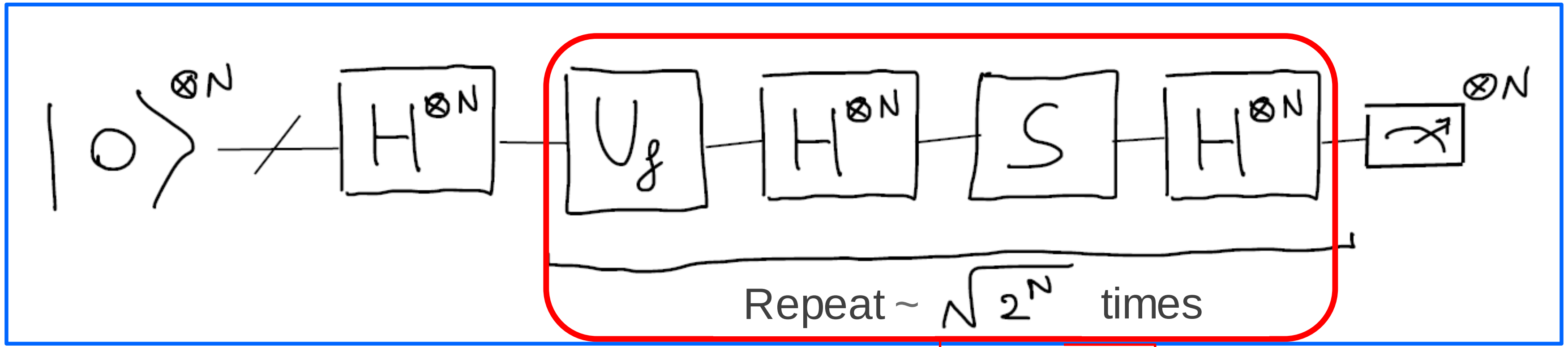




Reflection across vector s

Amplitude amplification of searched element

Grover Algorithm



Optimally $\frac{\pi\sqrt{2^N}}{4}$

Quadratic speedup wrt the classical case, where we have to evaluate this function 2^{N-1} times

Quantum Computing @ CINECA

CINECA: Italian HPC center

CINECA Quantum Computing Lab:

- Research with Universities, Industries and QC startups

- Internship programs, Courses and Conference (HPCQC)

https://www.quantumcomputinglab.cineca.it

r.mengoni@cineca.it

QUANTUM COMPUTING LAB