



THE UNIVERSITY of EDINBURGH
informatics

Introduction to Quantum Computing

Lecture 8: Bernstein–Vazirani Algorithm

Raul Garcia-Patron Sanchez

Balanced or constant function Game

$$f : \{0, 1\}^n \rightarrow \{0, 1\}$$

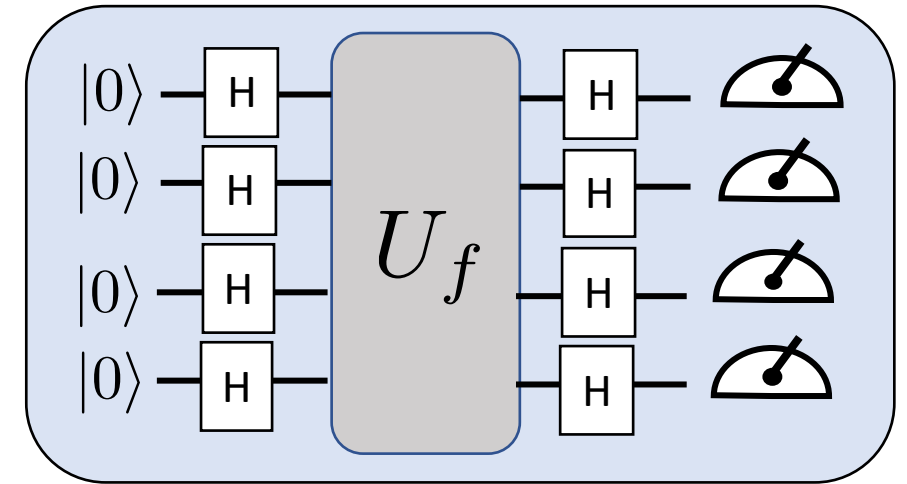
Constant: $f(x) = c \ \forall x$

Balanced: $f(x) = 1$ on half of the inputs

Phase oracle

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

U_f



Constant: $P(0^n) = 1$

Balanced: $P(0^n) = 0$

Bernstein–Vazirani Algorithm [BV93]

$$f : \{0, 1\}^n \rightarrow \{0, 1\}$$

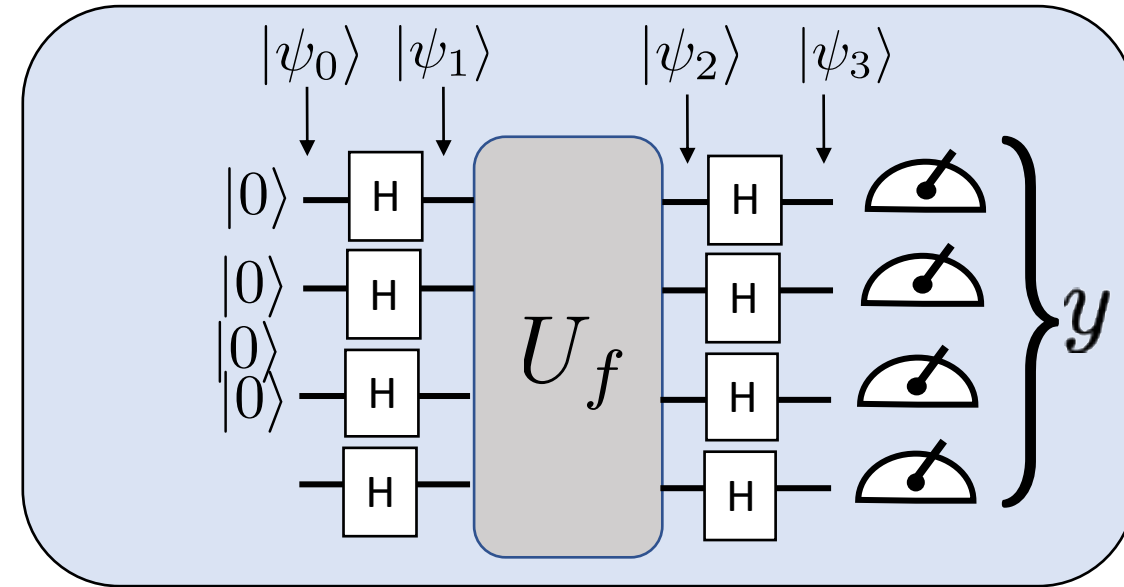
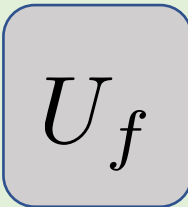
$$\text{Promise: } f(x) = a \cdot x = \sum_i a_i x_i \bmod 2$$

Problem: Find a

Same circuit as Deutsch–Jozsa
different promise

Phase oracle

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





THE UNIVERSITY of EDINBURGH
informatics

Oracles and Phase kickback

Raul Garcia-Patron Sanchez



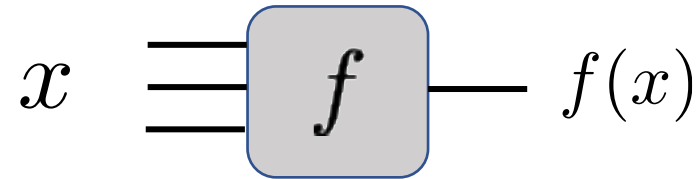
From classical to quantum oracles

Every classical oracle has a quantum analogue

Construction via reversible classical oracle



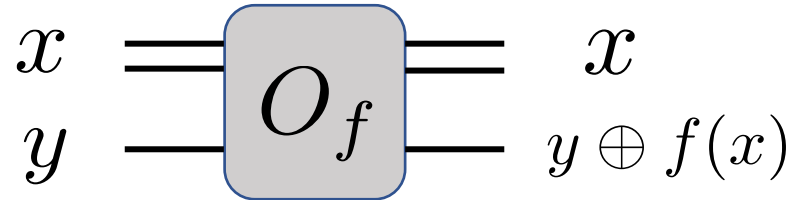
Classical oracle



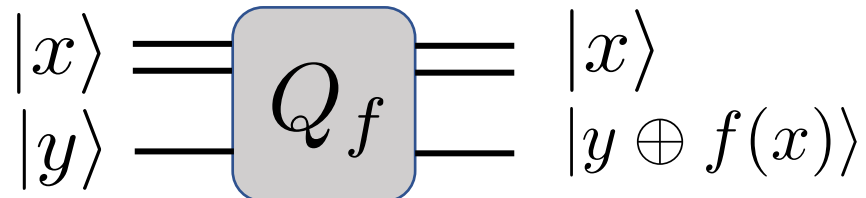
Reversible oracle

Address bits

Target bit



Quantum oracle



Allow superposition of inputs

Quantum oracle

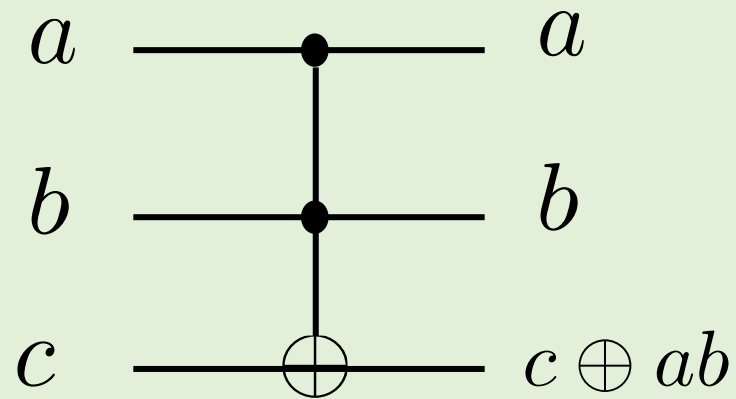
$$\begin{array}{ccc} |x\rangle & \text{---} & |x\rangle \\ |y\rangle & \text{---} & |y \oplus f(x)\rangle \end{array} \quad \boxed{Q_f}$$



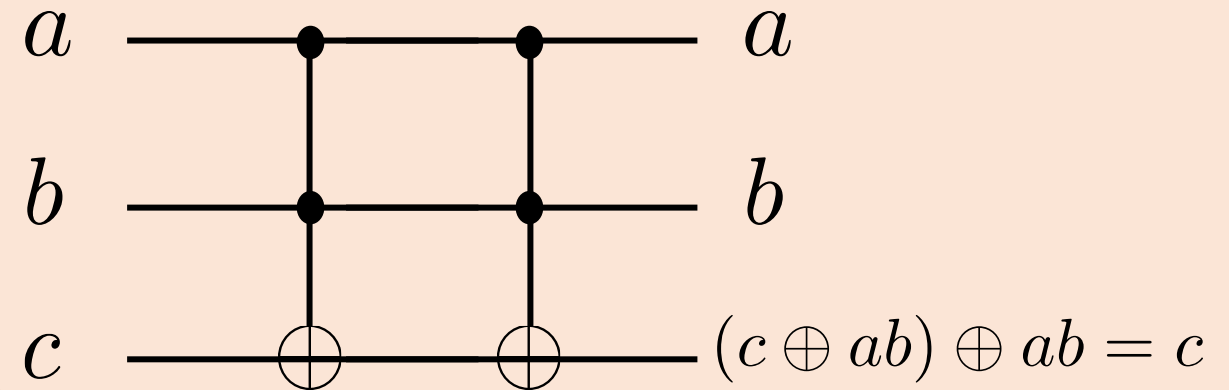
$$\left. \begin{array}{ccc} \sum_{x \in \{0,1\}^n} \psi_x |x\rangle & \text{---} & \\ & \text{---} & \\ & \text{---} & \\ |y\rangle & \text{---} & \end{array} \right\} \boxed{Q_f} \left. \begin{array}{ccc} & \text{---} & \\ & \text{---} & \\ & \text{---} & \\ & \text{---} & \end{array} \right\} \sum_{x \in \{0,1\}^n} \psi_x |x\rangle \otimes |y \oplus f(x)\rangle$$

Reversible classical circuits: a single Toffoli gate

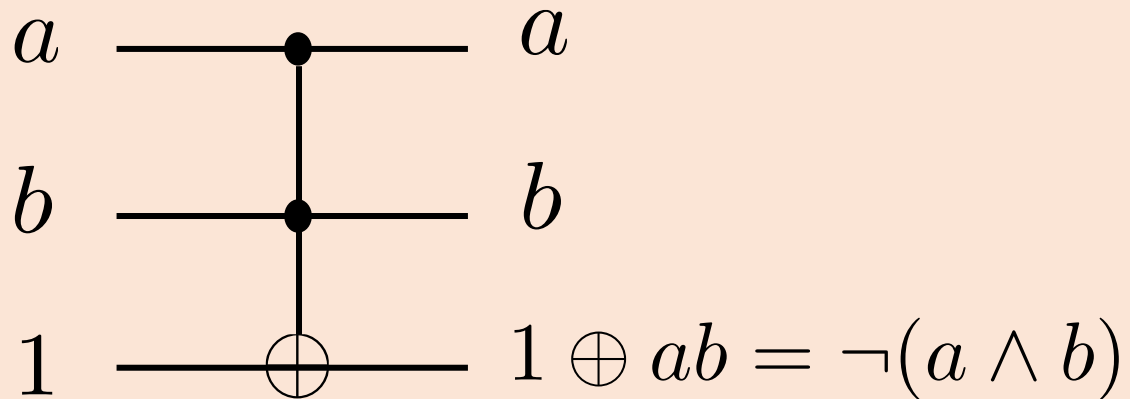
Toffoli gate



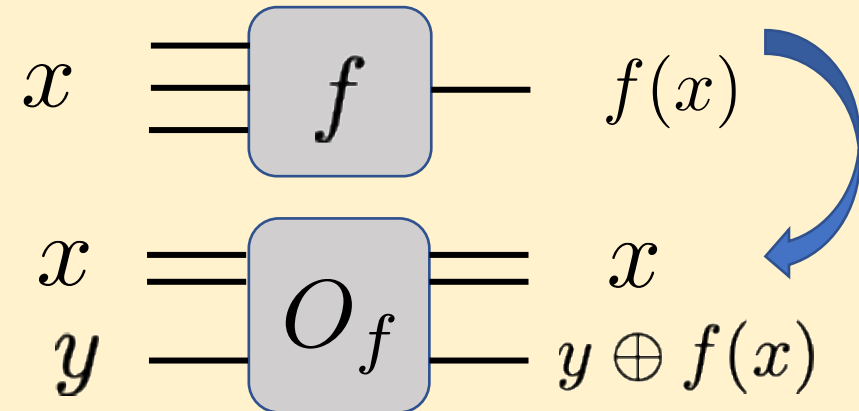
Reversible as self-inverse



Simulates NAND



Replace every NAND by a TOFFOLI



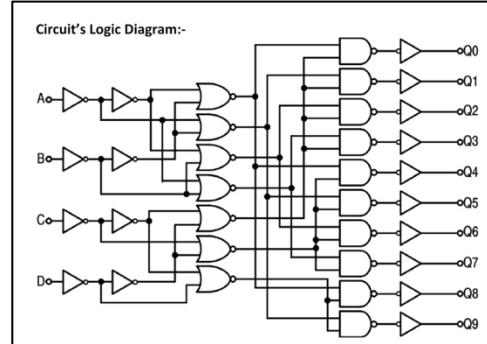
Reversible classical circuits: a full circuit

$$f : \{0, 1\}^n \rightarrow \{0, 1\}$$

Circuit with NAND gates

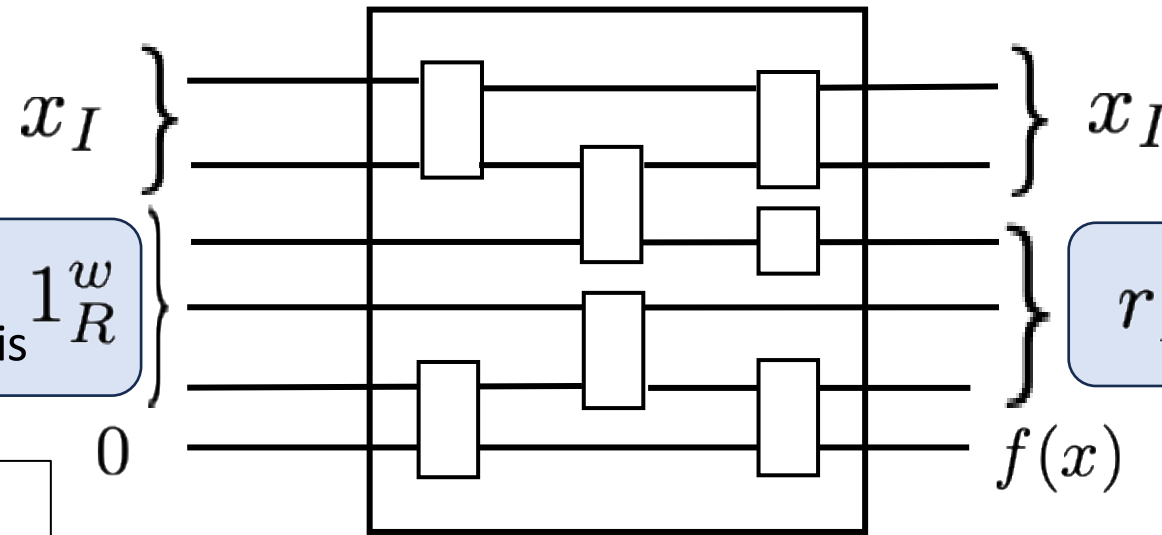
Input n bits

$$x = (x_1, \dots, x_n)$$



Output 1 bit

$$y = f(x)$$



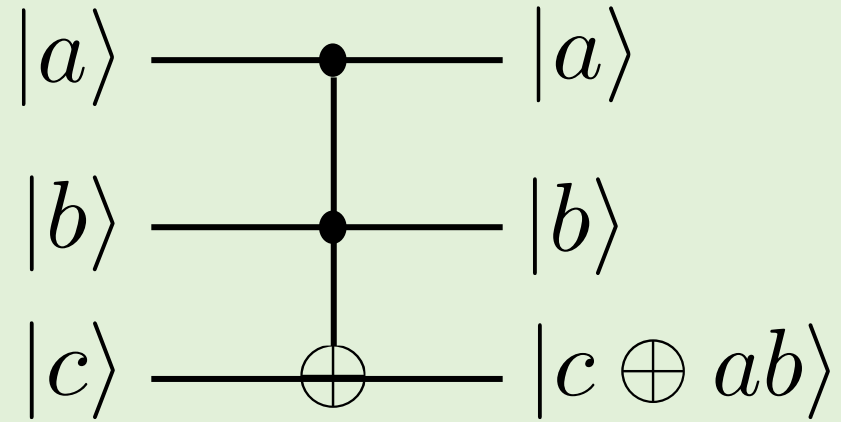
Extra memory needed
to make NANDs into Toffolis 1^w_R

$r_R(x)$ Memory of intermediate
NAND-gate outputs

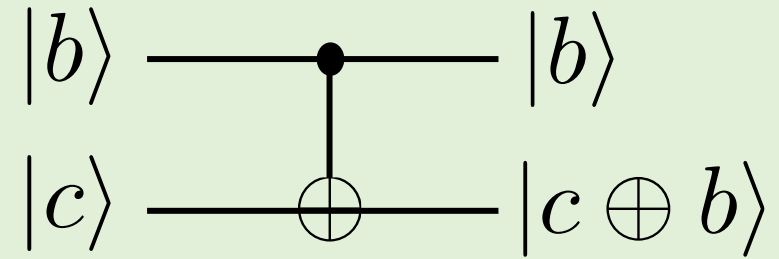
Input register I has n bits
Register R has w bits
Output register O has 1 bit

From classical reversible to quantum circuit

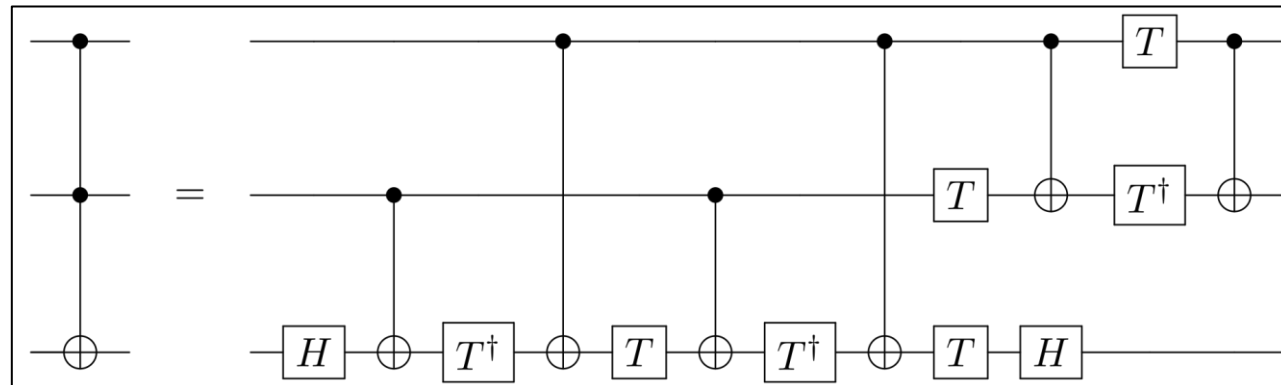
Toffoli gate



CNOT gate

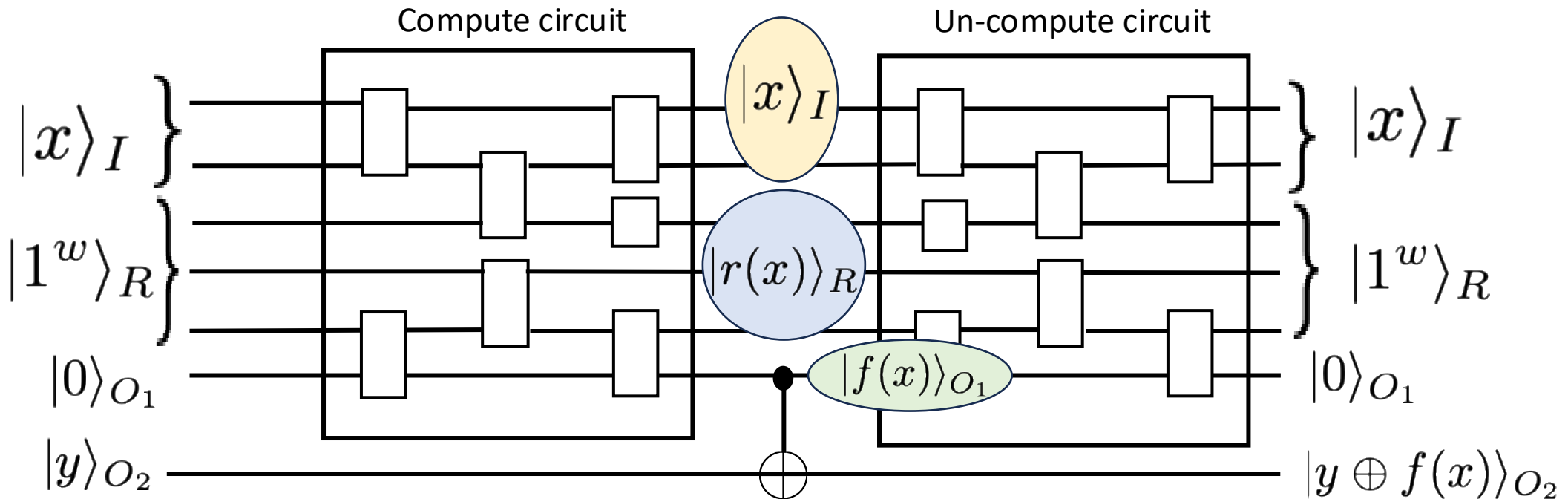


- Now allow for superpositions!
- Toffoli gate $\equiv$ Controlled-controlled-NOT
- A 2-qubit gate circuit



Reversible quantum circuits: a full circuit

$$f : \{0, 1\}^n \rightarrow \{0, 1\}$$



Input register I has n **qubits**
 Register R has w **qubits** (intermediate outputs)
 Output register O₁ has 1 **qubit** – later un-computed
 Output register O₂ has 1 **qubit** – quantum output

We un-compute to remove all memory of intermediate steps. We just want input – output

$$\begin{aligned}
 |x\rangle_I \otimes |1^w\rangle_R \otimes |0\rangle_{O_1} \otimes |y\rangle_{O_2} &\xrightarrow{C_{IRO_1} \otimes I_{O_2}} |x\rangle_I \otimes |r(x)\rangle_R \otimes |f(x)\rangle_{O_1} \otimes |y\rangle_{O_2} \\
 &\xrightarrow{I_{IR} \otimes \text{CNOT}_{O_1 \rightarrow O_2}} |x\rangle_I \otimes |r(x)\rangle_R \otimes |f(x)\rangle_{O_1} \otimes |y \oplus f(x)\rangle_{O_2} \\
 &\xrightarrow{C_{IRO_1}^\dagger \otimes I_{O_2}} |x\rangle_I \otimes |1^w\rangle_R \otimes |0\rangle_{O_1} \otimes |y \oplus f(x)\rangle_{O_2}
 \end{aligned}$$

From the quantum *function oracle* to the *phase oracle*

Quantum function oracle

$$\begin{array}{c} |x\rangle \\ |y\rangle \end{array} \Rightarrow \boxed{Q_f} \Rightarrow \begin{array}{c} |x\rangle \\ |y \oplus f(x)\rangle \end{array}$$

$$\left. \begin{array}{c} \sum_{x \in \{0,1\}^n} \psi_x |x\rangle \\ |y\rangle \end{array} \right\} \Rightarrow \boxed{Q_f} \Rightarrow \sum_{x \in \{0,1\}^n} \psi_x |x\rangle \otimes |y \oplus f(x)\rangle$$

Phase oracle

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

$$\boxed{U_f}$$



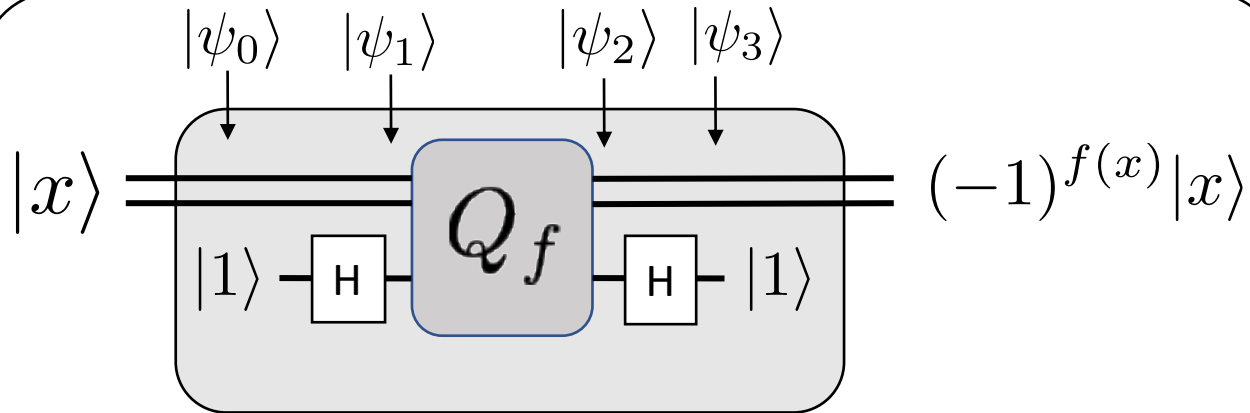
Phase Kickback

$$f : \{0, 1\}^n \rightarrow \{0, 1\}$$

Phase oracle

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

U_f



$$\begin{array}{c} |x\rangle \\ |y\rangle \end{array} \begin{array}{c} \text{---} \\ \text{---} \end{array} \begin{array}{c} \boxed{Q_f} \end{array} \begin{array}{c} \text{---} \\ \text{---} \end{array} \begin{array}{c} |x\rangle \\ |y \oplus f(x)\rangle \end{array}$$

Phase Kickback

$f : \{0, 1\}^n \rightarrow \{0, 1\}$

Phase oracle

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

U_f

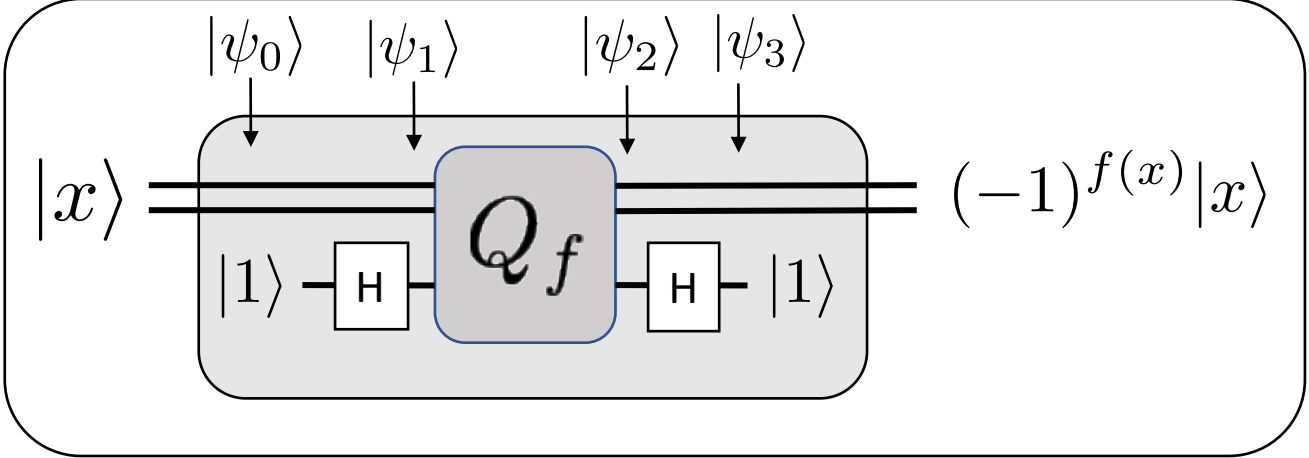
$|x\rangle =$

Q_f

 $=$
 $|x\rangle$
 $|y\rangle =$

Q_f

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



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

$|\psi_0\rangle = |x\rangle \otimes |1\rangle$
 $|\psi_1\rangle = |x\rangle \otimes \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle)$
 $|\psi_2\rangle = |x\rangle \otimes \frac{1}{\sqrt{2}}(|f(x)\rangle - |f(x) \oplus 1\rangle)$
 $= (-1)^{f(x)} |x\rangle \otimes \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle)$
 $|\psi_3\rangle = (-1)^{f(x)} |x\rangle \otimes |1\rangle$

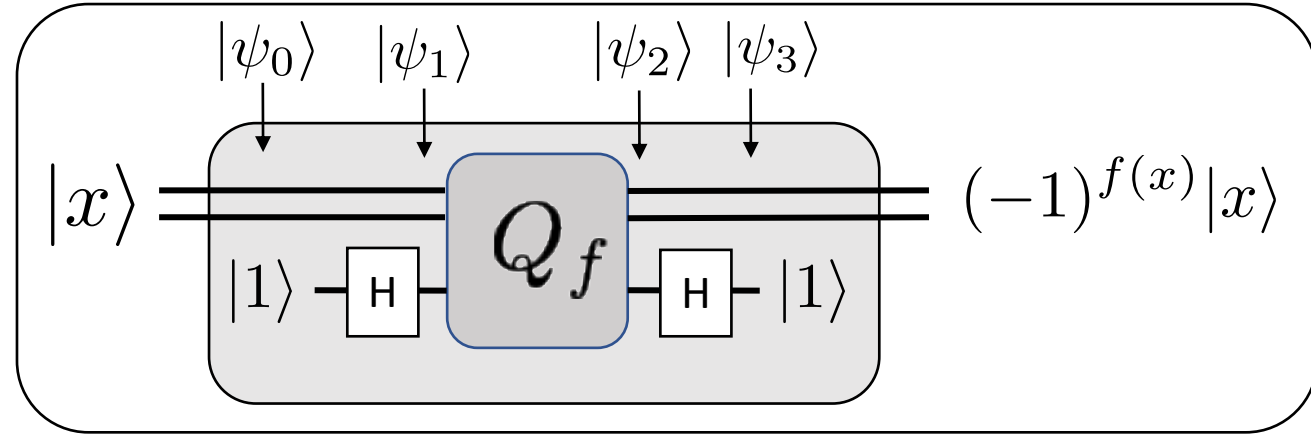
We can forget the ancillary qubit output

Phase oracle

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

U_f

Input: $|\phi\rangle = \sum_{x \in \{0,1\}^n} \phi_x |x\rangle$



$|\psi_0\rangle$

$$|\phi\rangle \otimes |1\rangle = \left(\sum_{x \in \{0,1\}^n} \phi_x |x\rangle \right) \otimes |1\rangle = \sum_{x \in \{0,1\}^n} \phi_x |x\rangle \otimes |1\rangle$$

$|\psi_1\rangle$

$$\xrightarrow{I_{2^n} \otimes H} \sum_{x \in \{0,1\}^n} \phi_x |x\rangle \otimes \frac{1}{\sqrt{2}} (|0\rangle - |1\rangle)$$

$|\psi_2\rangle$

$$\xrightarrow{Q_f} \sum_{x \in \{0,1\}^n} \phi_x (-1)^{f(x)} |x\rangle \otimes \frac{1}{\sqrt{2}} (|0\rangle - |1\rangle)$$

$|\psi_3\rangle$

$$\xrightarrow{I_{2^n} \otimes H} \sum_{x \in \{0,1\}^n} \phi_x (-1)^{f(x)} |x\rangle \otimes |1\rangle = \left(\sum_{x \in \{0,1\}^n} \phi_x (-1)^{f(x)} |x\rangle \right) \otimes |1\rangle$$

Phase Kickback

$$f : \{0, 1\}^n \rightarrow \{0, 1\}$$

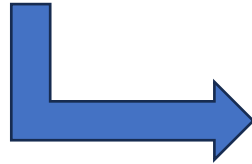
$$\begin{array}{c} |x\rangle \\ |y\rangle \end{array} \xrightarrow{Q_f} \begin{array}{c} |x\rangle \\ |y \oplus f(x)\rangle \end{array}$$

n+1 qubits



$$\begin{array}{c} |\psi_0\rangle \quad |\psi_1\rangle \quad |\psi_2\rangle \quad |\psi_3\rangle \\ |x\rangle \xrightarrow{\begin{array}{c} \text{---} \text{H} \text{---} \end{array}} \begin{array}{c} \text{---} Q_f \text{---} \end{array} \begin{array}{c} \text{---} \text{H} \text{---} \end{array} |1\rangle \end{array} \rightarrow (-1)^{f(x)} |x\rangle$$

n+1 qubits



N qubits

Phase oracle

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

U_f

Balanced or constant function Game

$$f : \{0, 1\}^n \rightarrow \{0, 1\}$$

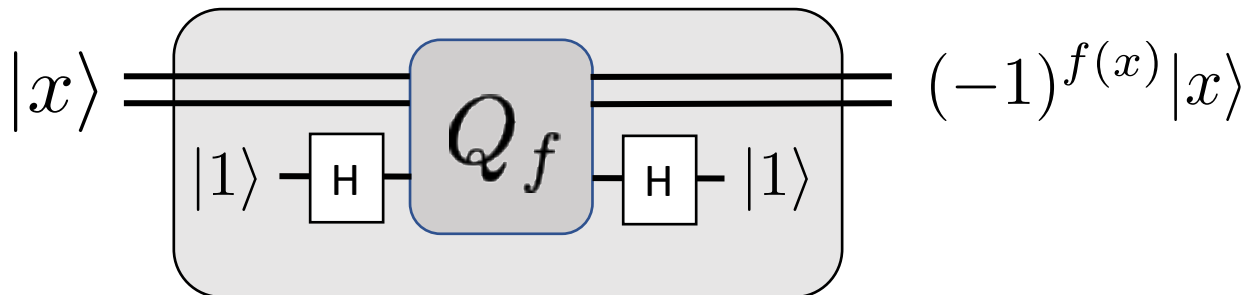
Constant: $f(x) = c \ \forall x$

Balanced: $f(x) = 1$ on half of the inputs

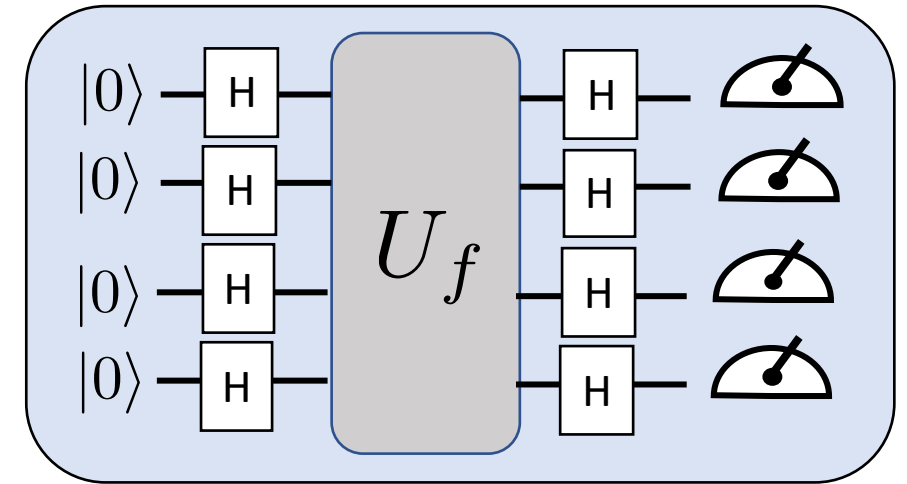
Phase oracle

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

U_f



Phase Kickback circuit



Constant: $P(0^n) = 1$

Balanced: $P(0^n) = 0$



THE UNIVERSITY of EDINBURGH
informatics

Bernstein–Vazirani Algorithm

Raul Garcia-Patron Sanchez

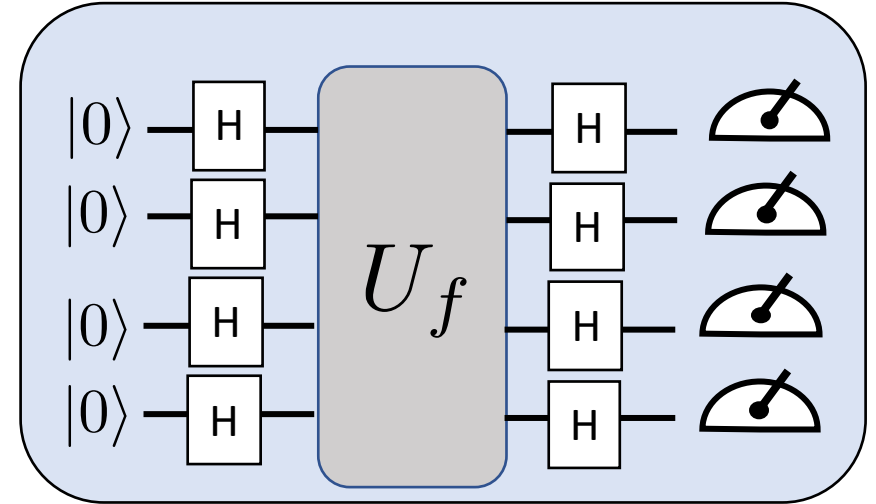


Detecting linear functions

$$f : \{0, 1\}^n \rightarrow \{0, 1\}$$

$$\text{Promise: } f(x) = a \cdot x = \sum_i a_i x_i \bmod 2$$

Problem: Find a

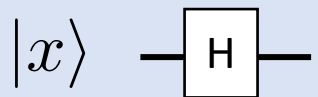


The same circuit as for Deutsch–Jozsa allows us to determine “ a ” with a single query to the quantum oracle whereas classically we need at least n queries.

1 single query!

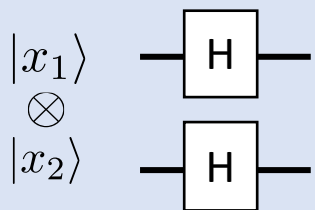


Walsh–Hadamard transform



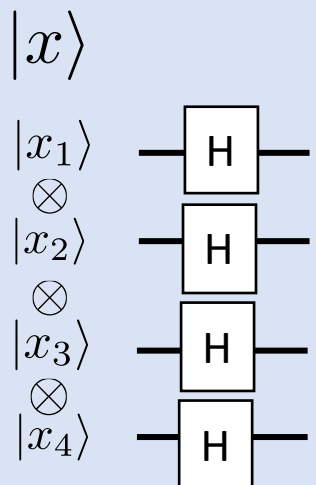
$$|x_1\rangle \xrightarrow{H} \frac{1}{\sqrt{2}} \sum_{y_1 \in \{0,1\}} (-1)^{x_1 y_1} |y_1\rangle$$

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



$$(H \otimes H)|x_1\rangle \otimes |x_2\rangle = \frac{1}{2} \left(\sum_{y_1 \in \{0,1\}} (-1)^{x_1 y_1} |y_1\rangle \right) \otimes \left(\sum_{y_2 \in \{0,1\}} (-1)^{x_2 y_2} |y_2\rangle \right)$$

$$(H \otimes H)|x_1\rangle \otimes |x_2\rangle = \frac{1}{2} \sum_{y_1, y_2 \in \{0,1\}} (-1)^{x_1 y_1 + x_2 y_2} |y_1\rangle \otimes |y_2\rangle$$



$$|x\rangle = |x_1, x_2, \dots, x_n\rangle = |x_1\rangle \otimes |x_2\rangle \otimes \dots \otimes |x_n\rangle$$

$$\text{where } x \cdot y = \sum_{i=1}^n x_i y_i$$

$$|x\rangle \xrightarrow{H^{\otimes n}} \bigotimes_{i=1}^n H|x_i\rangle = \frac{1}{\sqrt{2^n}} \bigotimes_{i=1}^n \left[\sum_{y_i \in \{0,1\}} (-1)^{x_i y_i} |y_i\rangle \right] = \frac{1}{\sqrt{2^n}} \sum_{y \in \{0,1\}^n} (-1)^{x \cdot y} |y\rangle$$

Bernstein–Vazirani Algorithm [BV93]

$$f : \{0, 1\}^n \rightarrow \{0, 1\}$$

Promise: $f(x) = a \cdot x = \sum_i a_i x_i \bmod 2$

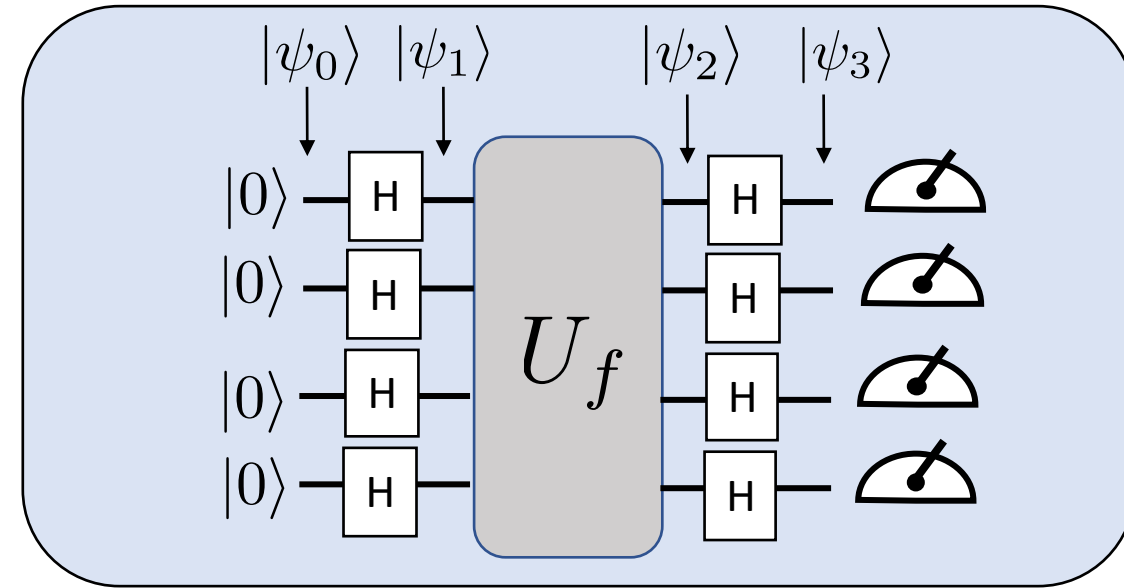
Problem: Find a

$$|0\rangle^{\otimes n} \xrightarrow{H^{\otimes n}} \frac{1}{\sqrt{2^n}} \sum_{x \in \{0,1\}^n} |x\rangle$$

$$\xrightarrow{U_f} \frac{1}{\sqrt{2^n}} \sum_{x \in \{0,1\}^n} (-1)^{f(x)} |x\rangle$$

$$\xrightarrow{H^{\otimes n}} \frac{1}{\sqrt{2^n}} \sum_{x \in \{0,1\}^n} (-1)^{f(x)} \left(\frac{1}{\sqrt{2^n}} \sum_{y \in \{0,1\}^n} (-1)^{x \cdot y} |y\rangle \right)$$

$$= \sum_{y \in \{0,1\}^n} \left(\frac{1}{2^n} \sum_{x \in \{0,1\}^n} (-1)^{f(x) + x \cdot y} \right) |y\rangle$$



Walsh–Hadamard transform

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

Bernstein–Vazirani Algorithm [BV93]

$$f : \{0, 1\}^n \rightarrow \{0, 1\}$$

Promise: $f(x) = a \cdot x = \sum_i a_i x_i \bmod 2$

Problem: Find a

$$|0\rangle^{\otimes n} \xrightarrow{H^{\otimes n}} \frac{1}{\sqrt{2^n}} \sum_{x \in \{0,1\}^n} |x\rangle$$

$$\xrightarrow{U_f} \frac{1}{\sqrt{2^n}} \sum_{x \in \{0,1\}^n} (-1)^{f(x)} |x\rangle$$

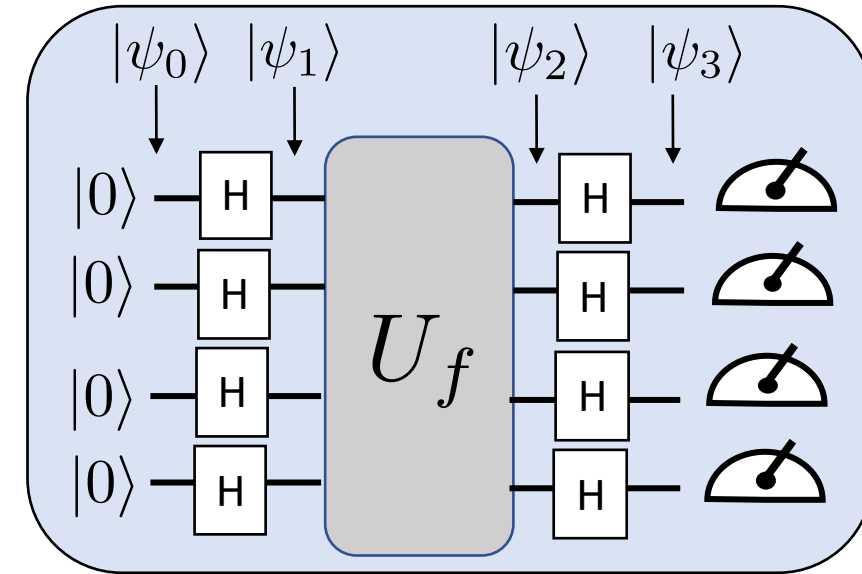
$$\xrightarrow{H^{\otimes n}} \sum_{y \in \{0,1\}^n} \left(\frac{1}{2^n} \sum_{x \in \{0,1\}^n} (-1)^{f(x) + x \cdot y} \right) |y\rangle$$

$$= \sum_{y \in \{0,1\}^n} \left(\frac{1}{2^n} \sum_{x \in \{0,1\}^n} (-1)^{(a \oplus y) \cdot x} \right) |y\rangle$$

$$|\psi_1\rangle$$

$$|\psi_2\rangle$$

$$|\psi_3\rangle$$



Phase oracle

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

Walsh–Hadamard transform

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

$$f(x) = a \cdot x$$

$$(-1)^{f(x) + x \cdot y} = (-1)^{f(x) \oplus x \cdot y} = (-1)^{(a \oplus y) \cdot x}$$

Bernstein–Vazirani Algorithm [BV93]

$$f : \{0, 1\}^n \rightarrow \{0, 1\}$$

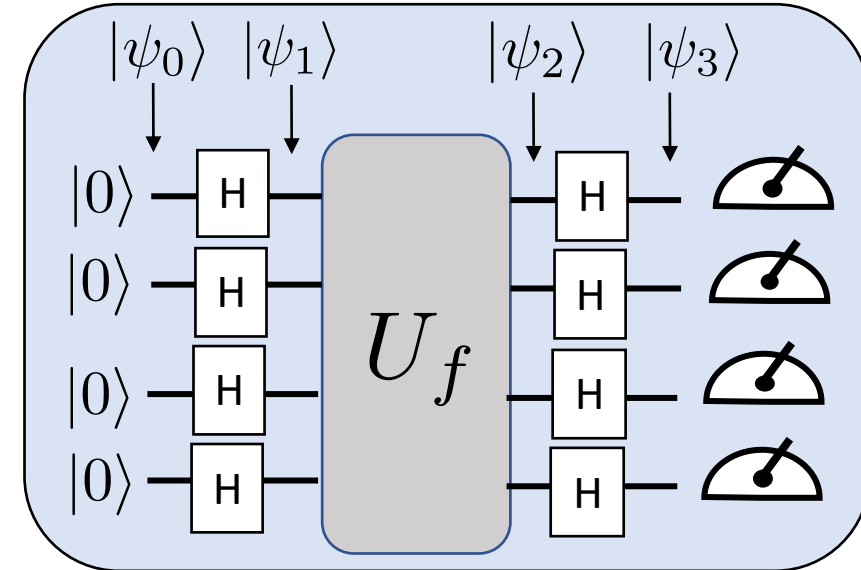
Promise: $f(x) = a \cdot x = \sum_i a_i x_i \bmod 2$

Problem: Find a

$$|0\rangle^{\otimes n} \rightarrow \sum_{y \in \{0,1\}^n} \left(\frac{1}{2^n} \sum_{x \in \{0,1\}^n} (-1)^{f(x) + x \cdot y} \right) |y\rangle$$

$$= \sum_{y \in \{0,1\}^n} \left(\frac{1}{2^n} \sum_{x \in \{0,1\}^n} (-1)^{(a \oplus y) \cdot x} \right) |y\rangle$$

$$P(y) = \left| \frac{1}{2^n} \sum_{x \in \{0,1\}^n} (-1)^{(a \oplus y) \cdot x} \right|^2$$



$$f(x) = a \cdot x$$

$$(-1)^{f(x) + x \cdot y} = (-1)^{f(x) \oplus x \cdot y} = (-1)^{(a \oplus y) \cdot x}$$

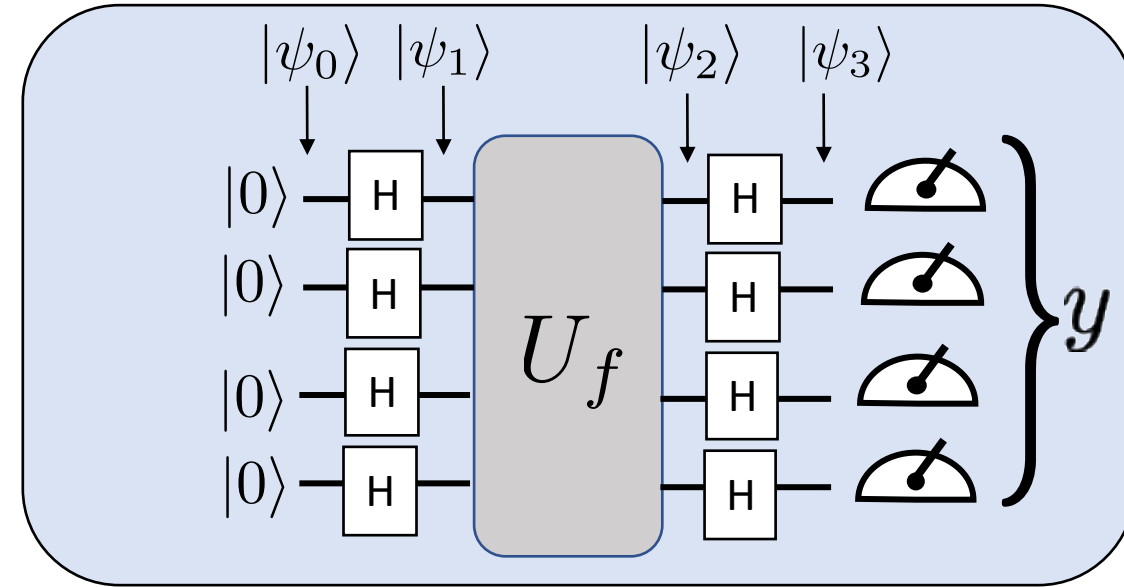
Bernstein–Vazirani Algorithm [BV93]

$$f : \{0, 1\}^n \rightarrow \{0, 1\}$$

$$\text{Promise: } f(x) = a \cdot x = \sum_i a_i x_i \bmod 2$$

Problem: Find a

$$P(y) = \left| \frac{1}{2^n} \sum_{x \in \{0,1\}^n} (-1)^{(a \oplus y) \cdot x} \right|^2$$



○ If $y = a$ we have $(-1)^{(a \oplus y) \cdot x} = 1 \quad \Rightarrow \quad P(a) = 1$

○ As $\sum_i P(i) = 1$ we have that $\forall y \neq a : P(y) = 0$

Classical strategies

$$f : \{0, 1\}^n \rightarrow \{0, 1\}$$

$$\text{Promise: } f(x) = a \cdot x = \sum_i a_i x_i \bmod 2$$

Problem: Find a



O_f



- Each query reveals at most 1 bit of information about a
 - a contains n bits, therefore we need $\Omega(n)$ queries to learn it
 - Linear separation between the quantum and classical complexities.
- Recursive version of problem needs $\text{poly}(n)$ quantum queries.
 - Classical randomized algorithms need $n^{\Omega(\log n)}$ queries.

Phase Kickback is a fundamental subroutine

General phase oracle

$$U_f : |x\rangle \rightarrow e^{i\phi_x} |x\rangle$$

U_f

This lecture $\phi_x = f(x)\pi$

Multiple output oracle

$$f : \{0, 1\}^n \rightarrow \{0, 1\}^m$$

$$\begin{array}{c} |x\rangle \\ |y\rangle \end{array} \begin{array}{c} \text{---} \\ \text{---} \end{array} \boxed{Q_f} \begin{array}{c} \text{---} \\ \text{---} \end{array} \begin{array}{c} |x\rangle \\ |y \oplus f(x)\rangle \end{array}$$

Bitwise (mod 2) addition

We can build oracles with larger output registers (see Simon's algorithm lecture 13).

Other reversible forms exist ($|x\rangle|y\rangle \rightarrow |x\rangle|g_x(y)\rangle$ with g_x a permutation (QPE version of Shor).

References

Reading references

1. Bernstein–Vazirani RdW 2.4.2 and G 7.5
2. Phase kickback RdW 2.4.1 and G 7.1-7.4

NC $\equiv$ Michael Nielsen and Isaac Chuang, Quantum Computation and Quantum Information
Cambridge University Press (2010)

RdW $\equiv$ Quantum Computing Lecture Notes, Ronald de Wolf, <https://arxiv.org/abs/1907.09415>

G $\equiv$ Introduction to Quantum Computation, Sevag Gharibian, [Lecture notes](#)