# Quantum Computation

## The phase kick-back effect: Bernstein-Varziani and Deutsch-Joza algorithms

Luís Soares Barbosa & Renato Neves

Universidade do Minho

HASLab HIGH-ASSURANCE SOFTWARE LABORATORY

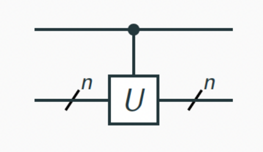
INL INTERNATIONAL IBERIAN NANOTECHNOLOGY LABORATORY

**MSc Physics Engineering**

Universidade do Minho, 2024-25

# The phase kick-back pattern

Recall that every quantum operation gives rise to a controlled quantum operation:




Let $v$ be an eigenvector of $U$ (*i.e.* $Uv = e^{i\theta}v$) and calculate

$$
\begin{aligned}
& cU\Big((\alpha|0\rangle + \beta|1\rangle) \otimes v\Big) \\
&= cU(\alpha|0\rangle \otimes v + \beta|1\rangle \otimes v) \\
&= \alpha|0\rangle \otimes v + \beta|1\rangle \otimes Uv \\
&= \alpha|0\rangle \otimes v + \beta|1\rangle \otimes e^{i\theta}v \\
&= (\alpha|0\rangle + e^{i\theta}\beta|1\rangle) \otimes v
\end{aligned}
$$

# The phase kick-back pattern

What just happened?

- Global phase $e^{i\theta}$ (introduced to $v$) was 'kicked-back' as a relative phase in the control qubit
- Some information of $U$ is now encoded in the control qubit

In general kicking-back such phases causes interference patterns that give away information about $U$

# A parenthesis on global/local phase

$\Big( \qquad \ldots$

# Global phase factor

## Definition

Let $v, u \in \mathbb{C}^{2^n}$ be vectors. If $u = e^{i\theta} v$ we say that it is equal to $v$ up to global phase factor $e^{i\theta}$

## Theorem

*$e^{i\theta} v$ and $v$ are indistinguishable in the world of quantum mechanics*

## Proof sketch

Show that equality up to global phase is preserved by operators and normalisation + show that probability outcomes associated with $v$ and $e^{i\theta} v$ are the same

# Relative phase factor

## Definition

We say that vectors $\sum_{x \in 2^n} \alpha_x |x\rangle$ and $\sum_{x \in 2^n} \beta_x |x\rangle$ differ by a relative phase factor if for all $x \in 2^n$

$$\alpha_x = e^{i\theta_x} \beta_x \qquad \text{(for some angle } \theta_x\text{)}$$

## Example

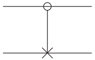
Vectors $|0\rangle + |1\rangle$ and $|0\rangle - |1\rangle$ differ by a relative phase factor.

Vectors that differ by a relative phase factor are distinguishable

# End of parenthesis

... $\color{red}{)}$

# Basic example: $U = cX$



$\overbrace{\begin{bmatrix} I & 0 \\ 0 & X \end{bmatrix}}^{cX}$ corresponds to the oracle: $|xy\rangle \mapsto |x, x \oplus y\rangle$

$$\begin{aligned} cX|0\rangle|\varphi\rangle &= |0\rangle I|\varphi\rangle \\ cX|1\rangle|\varphi\rangle &= |1\rangle X|\varphi\rangle \end{aligned}$$

Thus, e.g.

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

The phase jumps, or is kicked back, from the second to the first qubit.

# Basic example: $U = cX$

Actually, this happens because $\frac{|0\rangle - |1\rangle}{\sqrt{2}}$ is an eigenvector of

- $X$ (with $\lambda = -1$) and of $I$ (with $\lambda = 1$)
- and, thus, $X\frac{|0\rangle - |1\rangle}{\sqrt{2}} = -1\frac{|0\rangle - |1\rangle}{\sqrt{2}}$ and $I\frac{|0\rangle - |1\rangle}{\sqrt{2}} = 1\frac{|0\rangle - |1\rangle}{\sqrt{2}}$

Thus,

$$
\begin{aligned}
cX\,|1\rangle\left(\frac{|0\rangle - |1\rangle}{\sqrt{2}}\right) &= |1\rangle\left(X\left(\frac{|0\rangle - |1\rangle}{\sqrt{2}}\right)\right) \\
&= |1\rangle\left((-1)\left(\frac{|0\rangle - |1\rangle}{\sqrt{2}}\right)\right) \\
&= -|1\rangle\left(\frac{|0\rangle - |1\rangle}{\sqrt{2}}\right)
\end{aligned}
$$

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

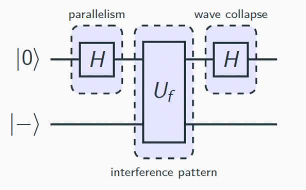
# The phase kick-back pattern

Phase kick-back in $cX$ can be presented as

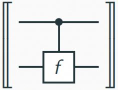
$$cX|b\rangle|-\rangle = (-1)^b|b\rangle|-\rangle$$

with $|b\rangle$ an element of the computational basis.

# Revisiting Deutsch's problem



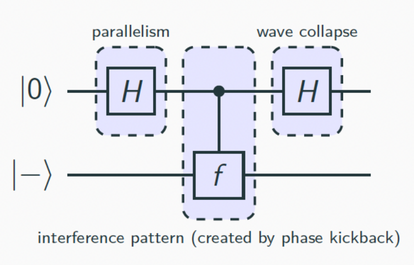

Oracle $U_f$ can be seen as a generalised controlled not-operation




$$= |x\rangle\,|y\rangle \mapsto \begin{cases} |x\rangle\,|y\rangle & \text{if } f(x) = 0 \\ |x\rangle\,\neg\,|y\rangle & \text{if } f(x) = 1 \end{cases}$$

# Revisiting Deutsch's problem

Thus,




Analogously to the $cX$ case, phase kick-back can be represented as

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

# The Bernstein-Vazirani algorithm

Let $2^n = \{0,1\}^n = \{0,1,2,\cdots 2^n - 1\}$ be the set of non-negative integers represented as bit strings up to $n$ bits). Then, consider the following problem:

## The problem

Let $s$ be an unknown non-negative integer less than $2^n$, encoded as a bit string, and consider a function $f : \{0,1\}^n \longrightarrow \{0,1\}$ which hides secret $s$ as follows: $f(x) = x \cdot s$, for some fixed bit-string $s$, where

$$x \cdot s \;=\; x_1 s_1 \oplus x_2 s_2 \oplus \cdots \oplus x_n s_n$$

i.e. the bitwise product of $x$ and $s$, modulo 2.

Note that juxtaposition abbreviates conjunction, i.e. $x_1 s_1 \;=\; x_1 \wedge s_1$

# Setting the stage

## Lemma

*(1) For* $a, b \in \{0, 1\}$ *the equation* $(-1)^a(-1)^b = (-1)^{a \oplus b}$ *holds*

## Proof sketch

Build a truth table for each case and compare the corresponding contents

## Lemma

*(2) For any three binary strings* $x, a, b \in \{0, 1\}^n$ *the equation* $(x \cdot a) \oplus (x \cdot b) = x \cdot (a \oplus b)$ *holds*

## Proof sketch

Follows from the fact that for any three bits $a, b, c \in \{0, 1\}$ the equation $(a \wedge b) \oplus (a \wedge c) = a \wedge (b \oplus c)$ holds

# Setting the stage

## Lemma

*(3)* *For any element $|b\rangle$ in the computational basis of $\mathbb{C}^2$,*

$$H|b\rangle = \frac{1}{\sqrt{2}} \sum_{z \in 2} (-1)^{b \wedge z} |z\rangle$$

## Proof sketch

Build a truth table and compare the corresponding contents

## Theorem

*(1)* *For any element $|b\rangle$ in the computational basis of $\mathbb{C}^{2^n}$,*

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

## Proof sketch

Follows by induction on the size of $n$

# The Bernstein-Vazirani algorithm

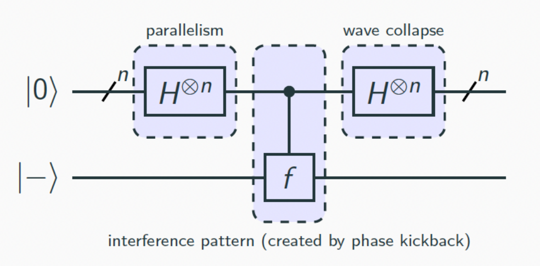
$\boxed{\text{How many times one has to call } f \text{ to determine } s\text{?}}$

- Classically, we run $f$ $n$-times by computing

$$\begin{aligned} f(1\dots 0) &= (s_1 \wedge 1) \oplus \cdots \oplus (s_n \wedge 0) = s_1 \\ &\vdots \\ f(0\dots 1) &= (s_1 \wedge 0) \oplus \cdots \oplus (s_n \wedge 1) = s_n \end{aligned}$$

- With a quantum algorithm, we may discover $s$ by running $f$ only once

# The circuit


parallelism
wave collapse
$|0\rangle$
$|-\rangle$
$n$
$H^{\otimes n}$
$f$
$H^{\otimes n}$
$n$
interference pattern (created by phase kickback)

# The computation

$$
\begin{aligned}
& H^{\otimes n}|0\rangle|-\rangle \\
&= \tfrac{1}{\sqrt{2^n}} \textstyle\sum_{z\in 2^n} |z\rangle|-\rangle && \{\text{Theorem (1)}\} \\
&\overset{U_f}{\mapsto} \quad \tfrac{1}{\sqrt{2^n}} \textstyle\sum_{z\in 2^n} (-1)^{f(z)}|z\rangle|-\rangle && \{\text{Definition}\} \\
&\overset{H^{\otimes n}\otimes I}{\mapsto} \quad \tfrac{1}{2^n} \textstyle\sum_{z\in 2^n} (-1)^{f(z)} \left( \textstyle\sum_{z'\in 2^n} (-1)^{z\cdot z'} |z'\rangle \right) |-\rangle && \{\text{Theorem (1)}\} \\
&= \tfrac{1}{2^n} \textstyle\sum_{z\in 2^n} \textstyle\sum_{z'\in 2^n} (-1)^{(z\cdot s)\oplus(z\cdot z')} |z'\rangle|-\rangle && \{\text{Lemma (1)}\} \\
&= \tfrac{1}{2^n} \textstyle\sum_{z\in 2^n} \textstyle\sum_{z'\in 2^n} (-1)^{z\cdot(s\oplus z')} |z'\rangle|-\rangle && \{\text{Lemma (2)}\} \\
&= |s\rangle|-\rangle && \{\text{Why?}\}
\end{aligned}
$$

# Why?

$$\cdots = \frac{1}{2^n} \sum_{z \in 2^n} \sum_{z' \in 2^n} (-1)^{z \cdot (s \oplus z')} |z'\rangle |-\rangle = \cdots$$

For each $z'$, $\frac{1}{2^n} \sum_{z=0}^{2^n-1} (-1)^{z \cdot (s \oplus z')}$ is $1$ iff $(s \oplus z') = 0$, which happens only if $s = z'$ In all other cases $\frac{1}{2^n} \sum_{z=0}^{2^n-1} (-1)^{z \cdot (s \oplus z')}$ is $0$.

The reason is easy to guess:

- for $s \oplus z' = 0$, $\frac{1}{2^n} \sum_{z=0}^{2^n-1} (-1)^{z \cdot (s \oplus z')} = \frac{1}{2^n} \sum_{z=0}^{2^n-1} 1 = 1$.
- for $s \oplus z' \neq 0$, as $z$ spans all numbers from 0 to $2^n - 1$, half of the $2^n$ factors in the sum will be $-1$ and the other half 1, thus summing up to 0.

Thus, the only non zero amplitude is the one associated to $s$.

# Why?

Alternatively, consider the probability of measuring $s$ at the end of the computation:

$$\left|\frac{1}{2^n}\sum_{z\in 2^n}(-1)^{z\cdot(s\oplus s)}\right|^2$$
$$=\left|\frac{1}{2^n}\sum_{z\in 2^n}(-1)^{z\cdot 0}\right|^2$$
$$=\left|\frac{1}{2^n}\sum_{z\in 2^n}1\right|^2$$
$$=\left|\frac{2^n}{2^n}\right|^2$$
$$=1$$

This means that somehow all values yielding wrong answers were completely cancelled.

# Deutsch-Josza

## The Problem

Take a function $f : \{0, 1\}^n \longrightarrow \{0, 1\}$, which is known to be either constant or balanced.

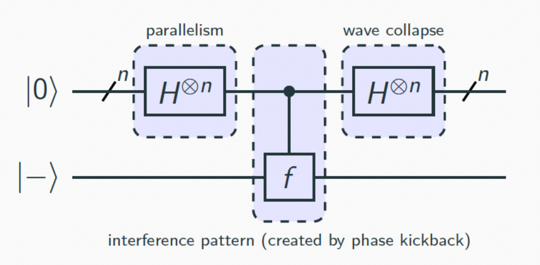
Find out which case holds.

Classically, we evaluate half of the inputs ($\frac{2^n}{2} = 2^{n-1}$), evaluate one more and run the decision procedure,

- output always the same $\Longrightarrow$ constant
- otherwise $\Longrightarrow$ balanced

which requires running $f$ $2^{n-1} + 1$ times.
A quantum algorithm replies by running $f$ only once.

# The circuit


parallelism
wave collapse
$|0\rangle$
$n$
$H^{\otimes n}$
$f$
$H^{\otimes n}$
$n$
$|-\rangle$
interference pattern (created by phase kickback)

# The computation

$$
\begin{aligned}
& H^{\otimes n}|0\rangle|-\rangle \\
&= \frac{1}{\sqrt{2^n}} \sum_{z\in 2^n} |z\rangle|-\rangle && \{\text{Theorem 1}\} \\
&\overset{U_f}{\longmapsto} \quad \frac{1}{\sqrt{2^n}} \sum_{z\in 2^n} (-1)^{f(z)}|z\rangle|-\rangle && \{\text{Definition }\} \\
&\overset{H^{\otimes n}\otimes I}{\longmapsto} \quad \underbrace{\frac{1}{2^n} \sum_{z\in 2^n} (-1)^{f(z)} \left( \sum_{z'\in 2^n} (-1)^{z\cdot z'}|z'\rangle \right)}_{\square \text{ upper qubits}} |-\rangle && \{\text{Theorem 1}\}
\end{aligned}
$$

# Developing $\square$ by case distinction

## $f$ is constant

$$\frac{1}{2^n} \sum_{z \in 2^n} (-1)^{f(z)} \left( \sum_{z' \in 2^n} (-1)^{z \cdot z'} |z'\rangle \right)$$
$$= \frac{1}{2^n} (\pm 1) \sum_{z \in 2^n} \left( \sum_{z' \in 2^n} (-1)^{z \cdot z'} |z'\rangle \right)$$

Therefore, the amplitude at state $|0\rangle$ is

$\boxed{f \text{ is constant at } 1} \quad \rightsquigarrow \quad \frac{-(2^n)|\mathbf{0}\rangle}{2^n} = -|\mathbf{0}\rangle$

$\boxed{f \text{ is constant at } 0} \quad \rightsquigarrow \quad \frac{(2^n)|\mathbf{0}\rangle}{2^n} = |\mathbf{0}\rangle$

# Developing $\square$ by case distinction

Actually the probability of measuring $|0\rangle$ at the end given by

$$\left|\frac{1}{2^n}(\pm 1)\sum_{z\in 2^n}(-1)^{z\cdot 0}\right|^2$$
$$=\left|\frac{1}{2^n}(\pm 1)\sum_{z\in 2^n}1\right|^2$$
$$=\left|\frac{2^n}{2^n}\right|^2$$
$$=1$$

So if $f$ is constant we measure $|0\rangle$ with probability 1.

# Developing $\square$ by case distinction

## $f$ is balanced

$$\frac{1}{2^n}\sum_{z\in 2^n}(-1)^{f(z)}\Big(\sum_{z'\in 2^n}(-1)^{z\cdot z'}|z'\rangle\Big)$$

$$= \frac{1}{2^n}\Big(\sum_{z\in 2^n, f(z)=0}(-1)^{f(z)}\Big(\sum_{z'\in 2^n}(-1)^{z\cdot z'}|z'\rangle\Big) + \sum_{z\in 2^n, f(z)=1}(-1)^{f(z)}\Big(\sum_{z'\in 2^n}(-1)^{z\cdot z'}|z'\rangle\Big)\Big)$$

$$= \frac{1}{2^n}\Big(\sum_{z\in 2^n, f(z)=0}\Big(\sum_{z'\in 2^n}(-1)^{z\cdot z'}|z'\rangle\Big) + \sum_{z\in 2^n, f(z)=1}(-1)\Big(\sum_{z'\in 2^n}(-1)^{z\cdot z'}|z'\rangle\Big)\Big)$$

# Developing $\square$ by case distinction

Probability of measuring $|0\rangle$ at the end given by

$$
\begin{aligned}
&\left|\frac{1}{2^n}\left(\sum_{z\in 2^n, f(z)=0}(-1)^{z\cdot 0} + \sum_{z\in 2^n, f(z)=1}(-1)(-1)^{z\cdot 0}\right)\right|^2 \\
&= \left|\frac{1}{2^n}\left(\sum_{z\in 2^n, f(z)=0} 1 + \sum_{z\in 2^n, f(z)=1}(-1)\right)\right|^2 \\
&= \left|\frac{1}{2^n}\left(\sum_{z\in 2^n, f(z)=0} 1 - \sum_{z\in 2^n, f(z)=1} 1\right)\right|^2 \\
&= 0
\end{aligned}
$$

So if $f$ is balanced we measure $|0\rangle$ with probability 0

# Concluding

## Deutsch problem

Classically, need to run $f$ twice. With a quantum algorithm once is enough.

## Berstein-Varziani problem

Classically, need to run $f$ $n$ times. With a quantum algorithm once is enough.

## Deutsch-Joza problem

Classically, need to evaluate half of the inputs ($\frac{2^n}{2} = 2^{n-1}$), evaluate one more and run the decision procedure,

- output always the same $\implies$ constant
- otherwise $\implies$ balanced

With a quantum algorithm once is enough.