MICRO-435 Quantum and Nanocomputing

Edoardo Charbon
Mariagrazia Graziano

September 16, 2025

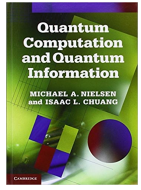
Quantum Computing Syllabus (Week 1-7)

- **Fundamentals of quantum computing**
- Qubit realization & control
- Cryo-CMOS components
- Scalable quantum computers
- Quantum communication, sensing, and metrology

Fundamentals of Quantum Computing (2)

- Basic concepts
- Qubits & superposition
- 1-qubit measurement
- 1-qubit gates
- 2-qubit systems
- 2-qubit measurement
- Bloch sphere, again (last week – Ian Glendinning/Vladimir)
- Unitary transforms
- 2-qubit gates
- Higher-qubit gates
- Encoding functions into unitaries
- Quantum algorithms: introduction
- Quantum arithmetic
- Quantum Fourier transform

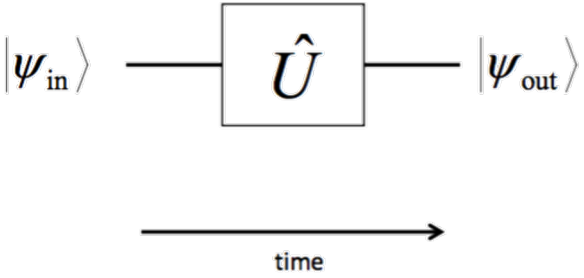
Chapter 4-5



Unitary Transforms

Review from Week1: Unitary Transform

- A unitary transformation is a specific ***rotation*** on the Bloch sphere where condition (1) is satisfied
- Note that a transform requires ***time*** to be executed



(1) $\hat{U}$ is a unitary operator when: $\hat{U}\hat{U}^{\dagger} = \hat{U}^{\dagger}\hat{U} = \hat{I}$

† = conjugate and transpose

Consequences

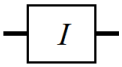
- For U the innerproduct is preserved:
- Then

$$|\Psi\rangle \xrightarrow[U]{} |\Psi'\rangle \qquad |\Phi\rangle \xrightarrow[U]{} |\Phi'\rangle$$

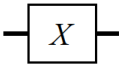
$$\langle \Phi' | \Psi' \rangle = \left\langle \Phi \middle| U^\dagger U \middle| \Psi \right\rangle = \langle \Phi | \Psi \rangle$$

- The reverse is also true, so if → iff.
- NB: unitary transformations preserve orthogonality, i.e. two orthogonal inputs produce orthogonal outputs.
- NB1: If two outputs are orthogonal, then also the inputs must be so.

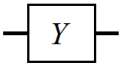
1-Qubit Gates Sofar

Identity 

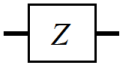
$$\hat{I} \doteq \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$$

Pauli X 

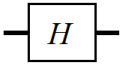
$$\hat{X} \doteq \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$$

Pauli Y 

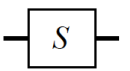
$$\hat{Y} \doteq \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix}$$

Pauli Z 

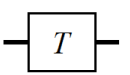
$$\hat{Z} \doteq \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}$$

Hadamard 

$$\hat{H} \doteq \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}$$

S 

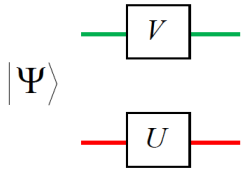
$$\hat{S} \doteq \begin{pmatrix} 1 & 0 \\ 0 & e^{i\pi/2} \end{pmatrix}$$

T 

$$\hat{T} \doteq \begin{pmatrix} 1 & 0 \\ 0 & e^{i\pi/4} \end{pmatrix}$$

Source:
Leo DiCarlo

Products of 1-qubit Unitaries



$|\Psi\rangle$

$V \otimes U |\Psi\rangle$

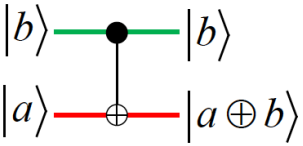
$$V \doteq \begin{pmatrix} a & b \\ c & d \end{pmatrix} \qquad U \doteq \begin{pmatrix} e & f \\ g & h \end{pmatrix}$$

$$V \otimes U \doteq \begin{pmatrix} a\begin{pmatrix} e & f \\ g & h \end{pmatrix} & b\begin{pmatrix} e & f \\ g & h \end{pmatrix} \\ c\begin{pmatrix} e & f \\ g & h \end{pmatrix} & d\begin{pmatrix} e & f \\ g & h \end{pmatrix} \end{pmatrix} \doteq \begin{pmatrix} ae & af & be & bf \\ ag & ah & bg & bh \\ ce & cf & de & df \\ cg & ch & dg & dh \end{pmatrix}$$

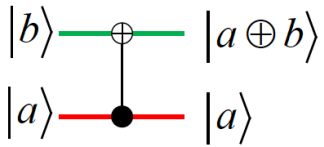
Source:
Leo DiCarlo

2-qubit Gates

Controlled-NOT Gate (C-NOT)



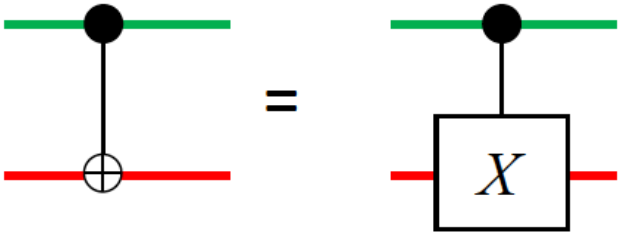
$$\text{C-NOT}_{gr} \doteq \begin{matrix} \\ |00\rangle \\ |01\rangle \\ |10\rangle \\ |11\rangle \end{matrix} \begin{matrix} \begin{matrix} |00\rangle & & & |11\rangle \end{matrix} \\ \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix} \end{matrix}$$



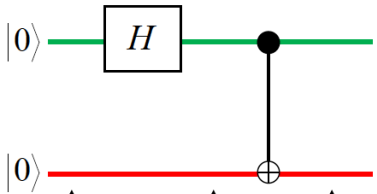
$$\text{C-NOT}_{rg} \doteq \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \end{pmatrix}$$

Source:
Leo DiCarlo

Controlled-NOT Gate (C-NOT)



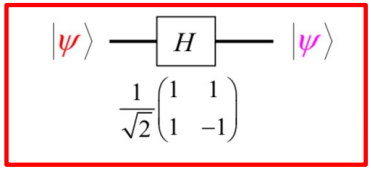
Generate a Bell State with a C-NOT



$|00\rangle$

$\frac{1}{\sqrt{2}}|00\rangle + \frac{1}{\sqrt{2}}|11\rangle$

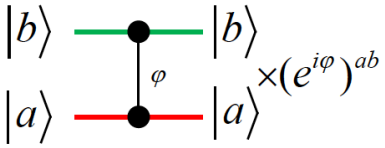
$\left(\frac{1}{\sqrt{2}}|0\rangle + \frac{1}{\sqrt{2}}|1\rangle\right)|0\rangle = \frac{1}{\sqrt{2}}|00\rangle + \frac{1}{\sqrt{2}}|10\rangle$



$$\text{C-NOT}_{gr} \doteq \begin{pmatrix}1 & 0 & 0 & 0\\ 0 & 1 & 0 & 0\\ 0 & 0 & 0 & 1\\ 0 & 0 & 1 & 0\end{pmatrix}$$

Recall: Bell states are states that form an orthonormal basis for the 4-dimensional Hilbert space of 2 qubits.

Controlled-Phase (C-PHASE)



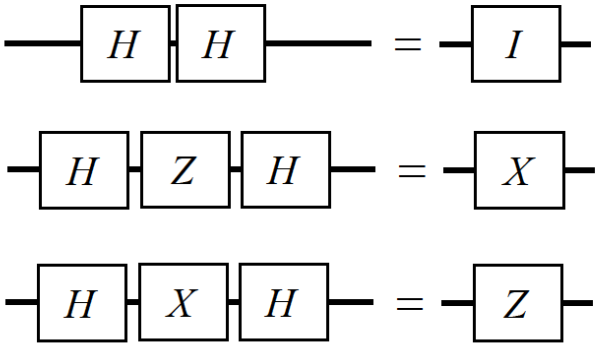
$$\mathrm{C-PHASE}_{\varphi} \doteq \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & e^{i\varphi} \end{pmatrix}$$

NB: when a phase is not specified, it is assumed to be π

Source:
Leo DiCarlo

Simplifying a Quantum Circuit

- Simplifying a quantum circuit is necessary to minimize the gates required to realize it.
- One can use these equalities:



Simplifying a Quantum Circuit (2)

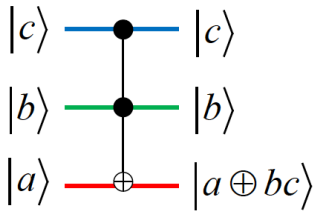
- The simplification process often requires adding Hadamard gates in strategic locations
- Transformation from C-NOT to C-PHASE:
  - The simplification process often re
    strategic locations
  - Transformation from C-NOT to C-P

- NB: an arbitrary operation with n qubits can be implemented using a circuit with $O(n^2 4^n)$ single-qubit and C-not gates (see Section 4.5.2).

Higher-qubit Gates

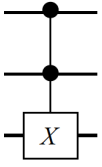
Toffoli Gate

- Also known as Controlled-Controlled-X (C-C-X) and Controlled-Controlled-NOT (C-C-NOT).

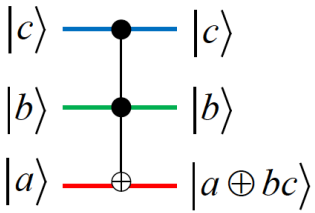


$$\text{TOFFOLI} \doteq \begin{array}{c} \begin{matrix} |000\rangle & & & & & & & |111\rangle \end{matrix} \\ \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{pmatrix} \end{array} \begin{matrix} |000\rangle \\ \\ \\ \\ \vdots \\ \\ \\ |111\rangle \end{matrix}$$

- Another symbol:

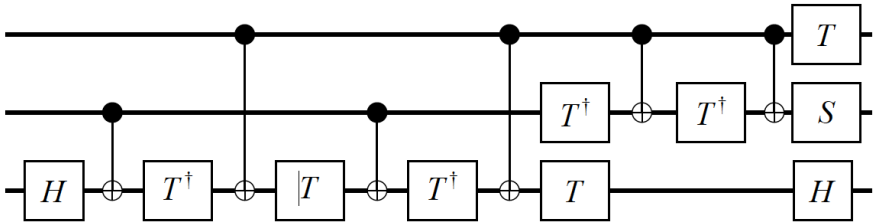


Toffoli Gate Decomposition



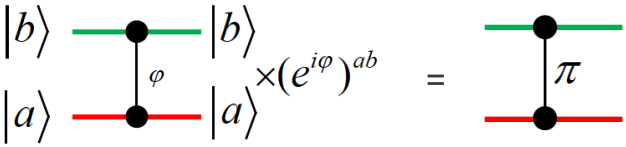
S — S $\quad \hat{S} \doteq \begin{pmatrix} 1 & 0 \\ 0 & e^{i\pi/2} \end{pmatrix}$

T — T $\quad \hat{T} \doteq \begin{pmatrix} 1 & 0 \\ 0 & e^{i\pi/4} \end{pmatrix}$

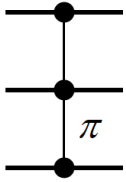


C-C-PHASE Gate

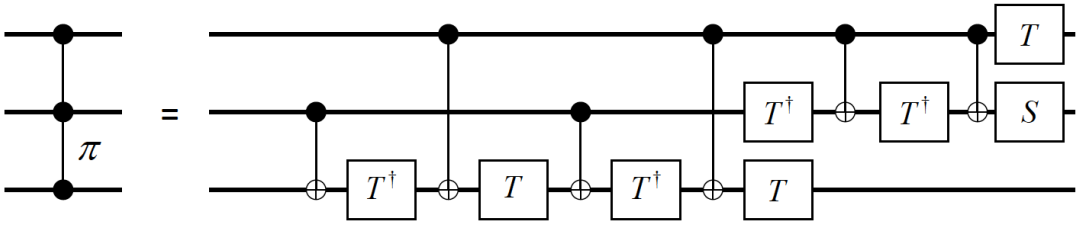
- C-PHASE$_\pi$ was defined as



- The C-C-PHASE$_\pi$ is defined as follows



C-C-PHASE Gate Decomposition



S — S — $\hat{S} \doteq \begin{pmatrix} 1 & 0 \\ 0 & e^{i\pi/2} \end{pmatrix}$

T — T — $\hat{T} \doteq \begin{pmatrix} 1 & 0 \\ 0 & e^{i\pi/4} \end{pmatrix}$

Encoding Functions into Unitaries

Can we Encode Boolean Functions into Unitaries?

- Let $f(x)$ be a m-dimensional mapping of an n-dimensional Boolean variable x.

$$|x\rangle \rightarrow |f(x)\rangle$$

$$|x'\rangle \rightarrow |f(x')\rangle$$

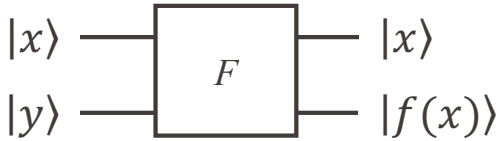
- Therefore:

$$\langle x|x'\rangle = 0 \overset{?}{\Longleftrightarrow} \langle f(x)|f(x')\rangle = 0$$

Not necessarily, since x, x' (where $x \neq x'$) may give $f(x) = f(x')$.

How about Maintaining a Copy of the Inputs?

- Let $f(x)$ be a m-dimensional mapping of an n-dimensional Boolean variable x.



$$|x, y\rangle \rightarrow |x, f(x)\rangle$$

$$|x', y'\rangle \rightarrow |x', f(x')\rangle$$

- Therefore:

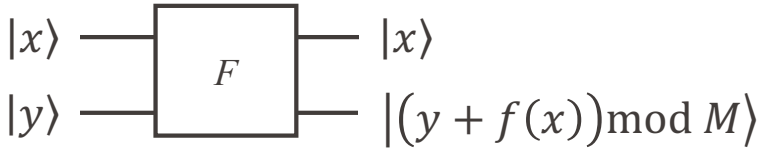
$$\langle x, y|x', y'\rangle = 0 \overset{?}{\Leftrightarrow} \langle x, f(x)|x', f(x')\rangle = 0$$

Not necessarily, since $x = x'$ and it is possible that $y \neq y'$.

Thus, F not unitary!

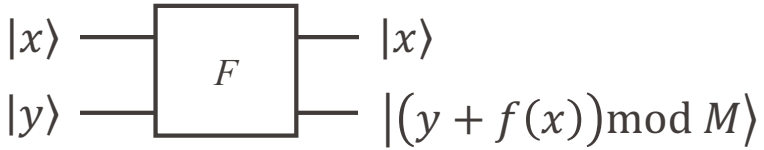
How about This?

- Let $f(x)$ be a m-dimensional mapping of an n-dimensional Boolean variable x.



F is unitary!

- Let $f(x)$ be a m-dimensional mapping of an n-dimensional Boolean variable x.

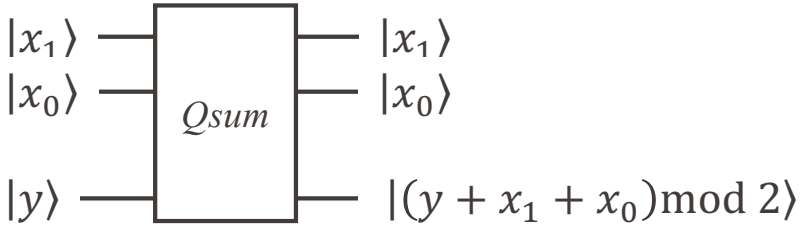


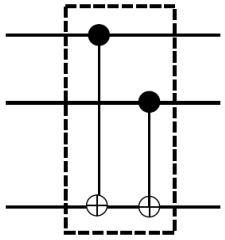
- Therefore:

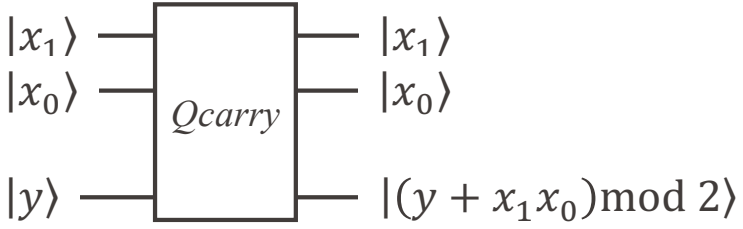
$$\langle x, y | x', y' \rangle = 0 \Leftrightarrow \langle x', (y + f(x)) \bmod M | x', (y' + f(x')) \bmod M \rangle = 0$$

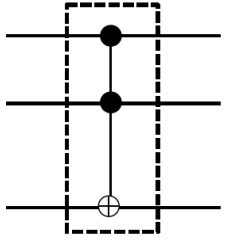
Thus, F unitary!

Example: Quantum Arithmetic

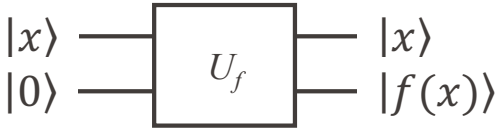




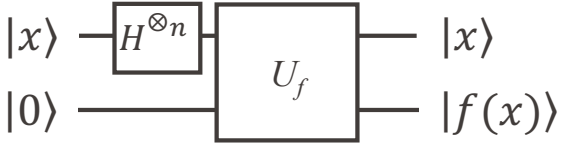




Quantum Parallelism



- Prepare a superposition of all possible x inputs:



$$|0\rangle|0\rangle \rightarrow \frac{1}{\sqrt{N}} \sum_{x=0}^{N-1} |x\rangle|0\rangle \rightarrow \frac{1}{\sqrt{N}} \sum_{x=0}^{N-1} |x\rangle\,|f(x)\rangle$$

- Evaluate function for all inputs in one go

Quantum Algorithms: Introduction

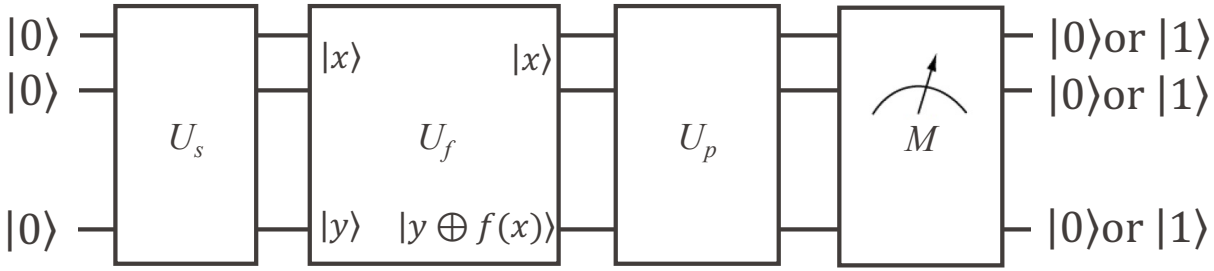
Deutsch Algorithm

Grover Algorithm

Recall: DiVincenzo Criteria for Quantum Computing

1. A scalable physical system with well characterized qubits.
2. The ability to initialize the state of the qubits to a simple fiducial state.
3. Long relevant decoherence times.
4. A “universal” set of quantum gates.
5. A qubit-specific measurement capability.
6. The ability to interconvert stationary and flying qubits.
7. The ability to faithfully transmit flying qubits between specified locations.

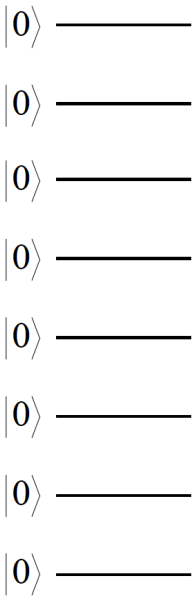
Essence of a Quantum Algorithm



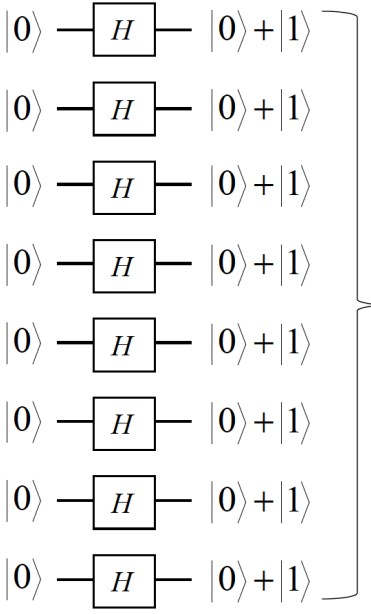
Maintain quantum coherence

- Initialize qubits
- Create superposition
- Encode function in unitary
- Process
- Measure

Initialize Qubits



Create Maximum Superposition



$$|\Psi\rangle = \frac{1}{\sqrt{N}} \sum_{x=0}^{N-1} |x\rangle$$

$$= \frac{1}{\sqrt{N}} \left(|00000000\rangle + \ldots + |11111111\rangle \right)$$

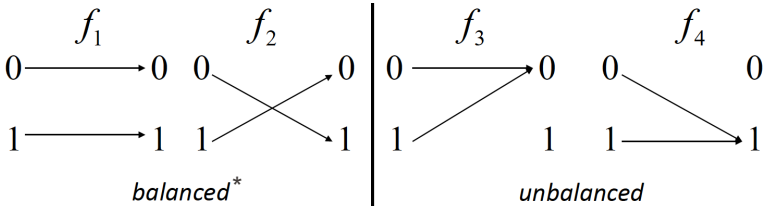
Deutsch's Problem

- Classical problem:

You are given a black box with one of the four 1-bit to 1-bit Boolean functions f_i. Determine if the function is balanced or unbalanced.

- Quantum version of the problem:

You are given a quantum black box with one of functions f_i encoded into the unitary U_{f_i}. Determine if the function is balanced or unbalanced.



*) Balanced means that the function gives '1' for exactly half of the possible inputs and '0' otherwise.

Deutsch's Problem (2)

- Classical problem:

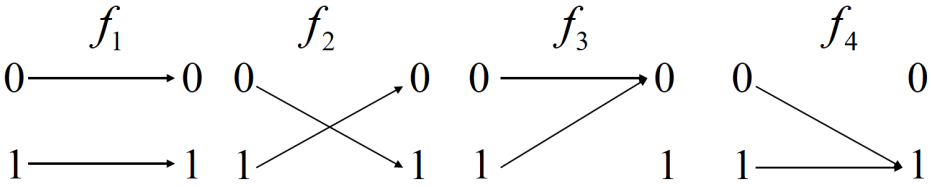
You must make two calls to the black box to determine if the function is balanced or not. You need to evaluate both '0' and '1'.

In the worst case, you will need to perform $\frac{2^n}{2} + 1$ queries.

- Quantum version of the problem:

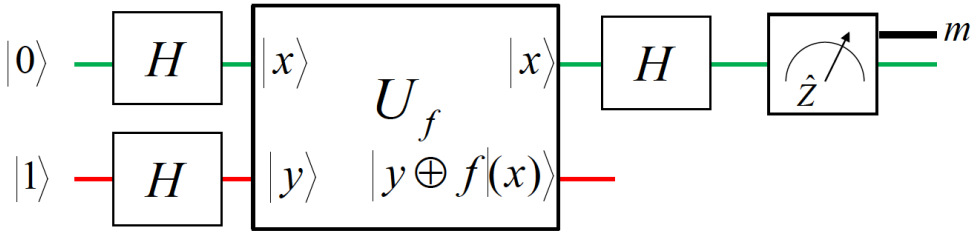
Only one call gives you the answer.

Deutsch's Problem Quantum Solution



$$U_{f_1} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix} \quad U_{f_2} = \begin{pmatrix} 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad U_{f_3} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad U_{f_4} = \begin{pmatrix} 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix}$$

Deutsch's Problem Quantum Solution (2)

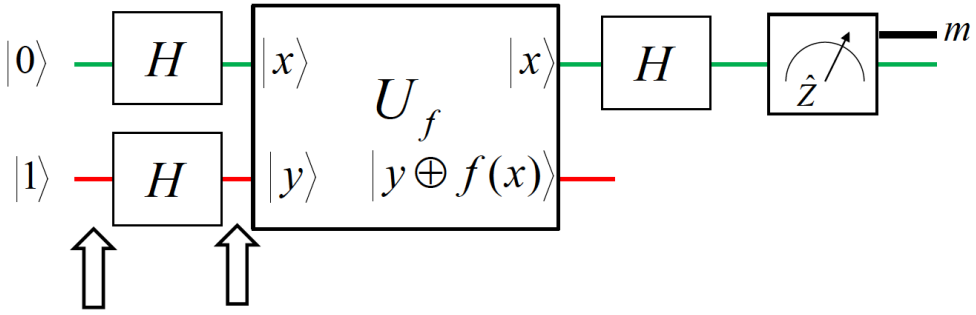


$m = +1$ ⟶ function is unbalanced

$m = -1$ ⟶ function is balanced

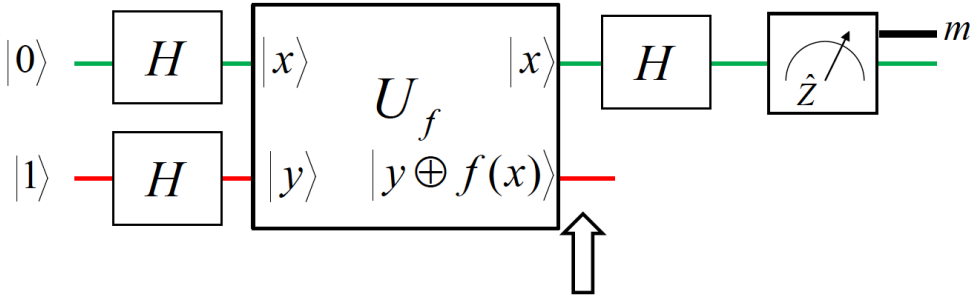
Source:
Leo DiCarlo

Deutsch's Problem Quantum Solution (2)



Source:
Leo DiCarlo

Deutsch's Problem Quantum Solution (2)



Case *f(x)=0:*

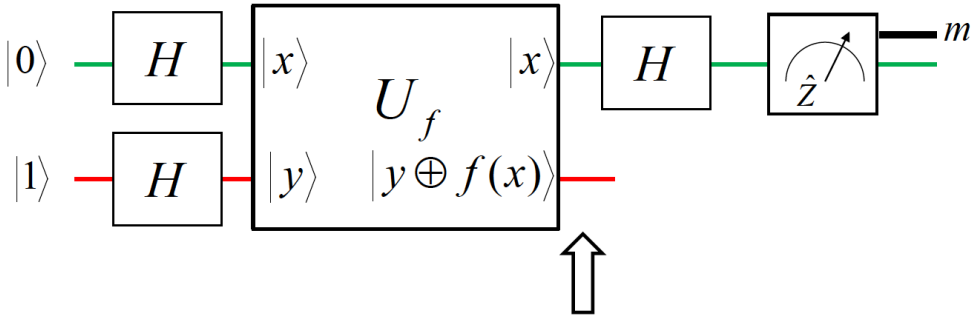
$$|x\rangle \otimes (|0\rangle - |1\rangle) \rightarrow |x\rangle \otimes (|0 \oplus 0\rangle - |1 \oplus 0\rangle)$$
$$= |x\rangle \otimes (|0\rangle - |1\rangle)$$
$$= (-1)^{f(x)} |x\rangle \otimes (|0\rangle - |1\rangle)$$

Case *f(x)=1:*

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

Source:
Leo DiCarlo

Deutsch's Problem Quantum Solution (2)



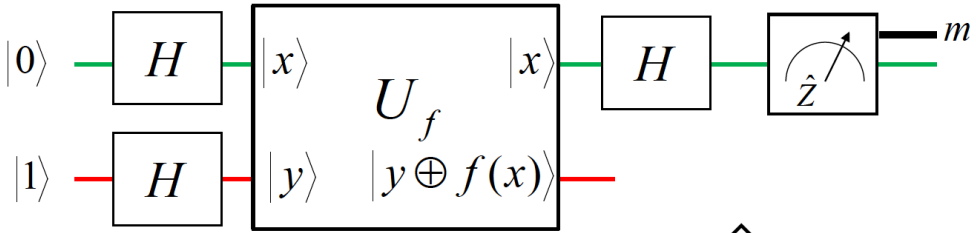
$$(-1)^{f(0)}|0\rangle\otimes\left(|0\rangle-|1\rangle\right)+(-1)^{f(1)}|1\rangle\otimes\left(|0\rangle-|1\rangle\right)$$

$$(-1)^{f(0)}\left(|0\rangle+(-1)^{f(1)-f(0)}|1\rangle\right)\otimes\left(|0\rangle-|1\rangle\right)$$

State of LSB is not changed by U_f.
State of the MSB is: quantum amplitude $|x\rangle$ multiplied by $(-1)^{f(x)}$
This trick is called *quantum kickback*.

Source:
Leo DiCarlo

Deutsch's Problem Quantum Solution (2)



$$|0\rangle \otimes (|0\rangle - |1\rangle) \text{ if } f(0) = f(1)$$

$$|1\rangle \otimes (|0\rangle - |1\rangle) \text{ if } f(0) \neq f(1)$$

$m = +1 \longrightarrow$ function is unbalanced

$m = -1 \longrightarrow$ function is balanced

Source:
Leo DiCarlo

The Search Problem

- Classical problem:

  You are given a database (e.g. a series or a vector or a function)

$$f(x) = \begin{cases} 1 & \text{for } x = x^* \\ 0 & \text{for } x \neq x^* \end{cases}$$

- Problem: find x^*

- How many calls to $f(x)$ are needed, on average?

Exercise

The Search Problem

- Classical problem:

  You are given a database (e.g. a series or a vector or a function)

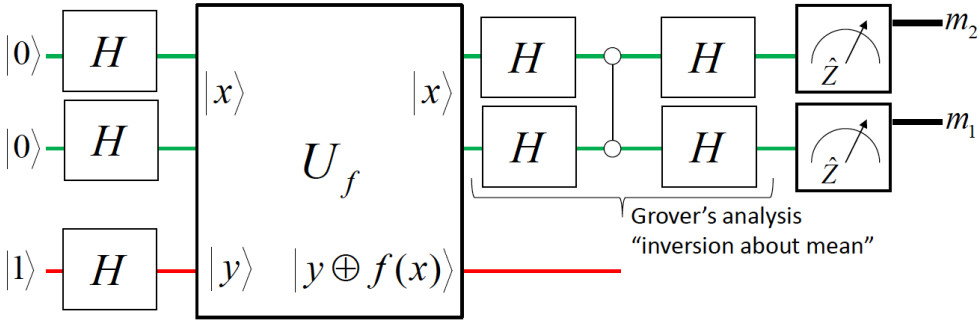
$$f(x) = \begin{cases} 1 & \text{for } x = x^* \\ 0 & \text{for } x \neq x^* \end{cases}$$

- Problem: find x^*

- Quantum mechanically, 1 readout is enough.

Grover's Search Algorithm

- Execute this sequence, calling the quantum box only once!

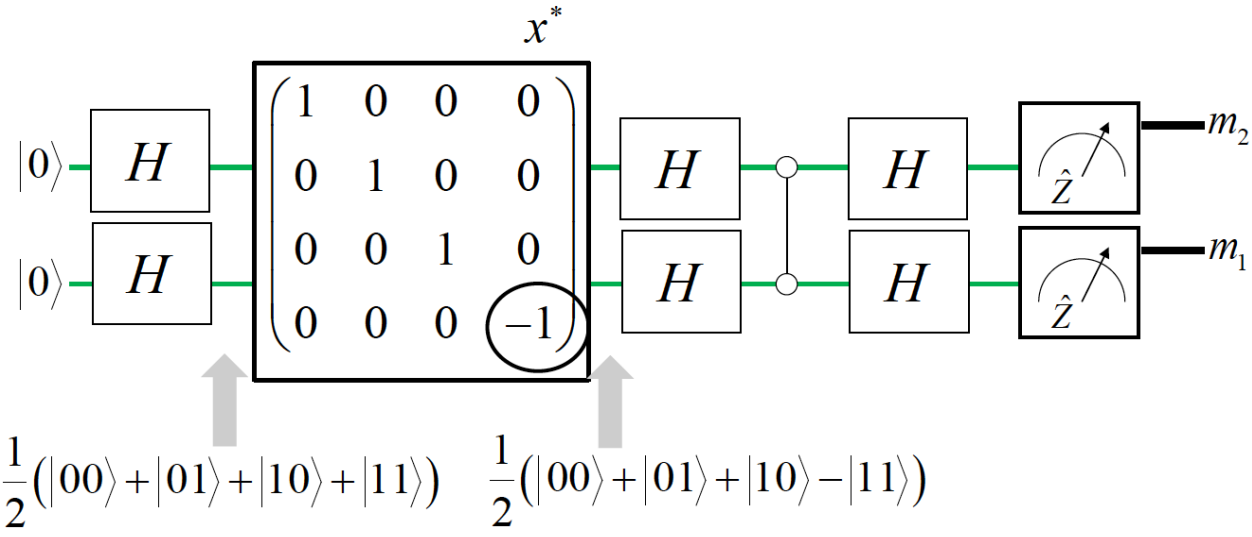


- Answer:

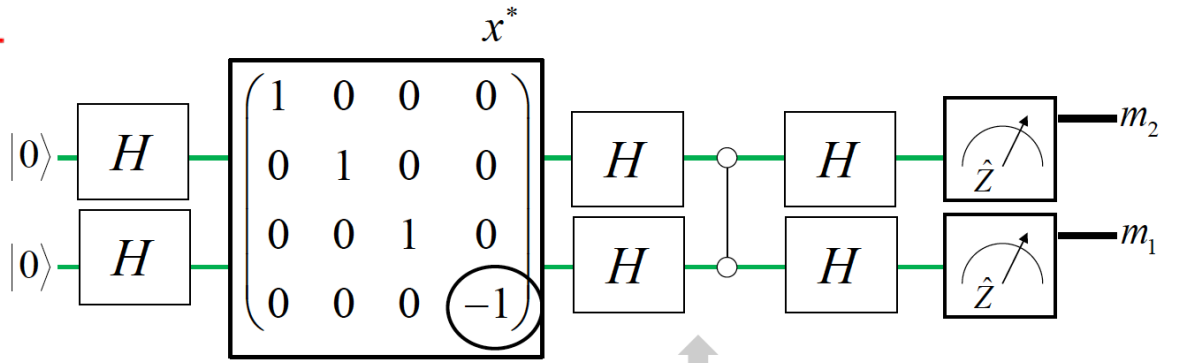
$$(m_2, m_1) = \begin{cases} (+1,+1) & x^* = 00 \\ (+1,-1) & x^* = 01 \\ (-1,+1) & x^* = 10 \\ (-1,-1) & x^* = 11 \end{cases}$$

Source:
Leo DiCarlo

Example

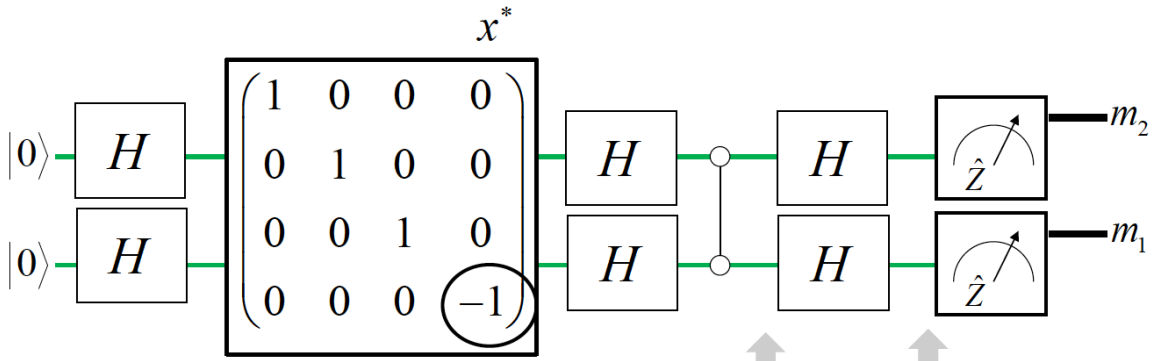


Source:
Leo DiCarlo



$$\frac{1}{4}\begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \end{pmatrix}\begin{pmatrix} 1 \\ 1 \\ 1 \\ -1 \end{pmatrix} = \frac{1}{2}\begin{pmatrix} 1 \\ 1 \\ 1 \\ -1 \end{pmatrix}$$

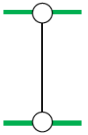
$$\frac{1}{2}\left(|00\rangle + |01\rangle + |10\rangle - |11\rangle\right)$$



$|11\rangle$

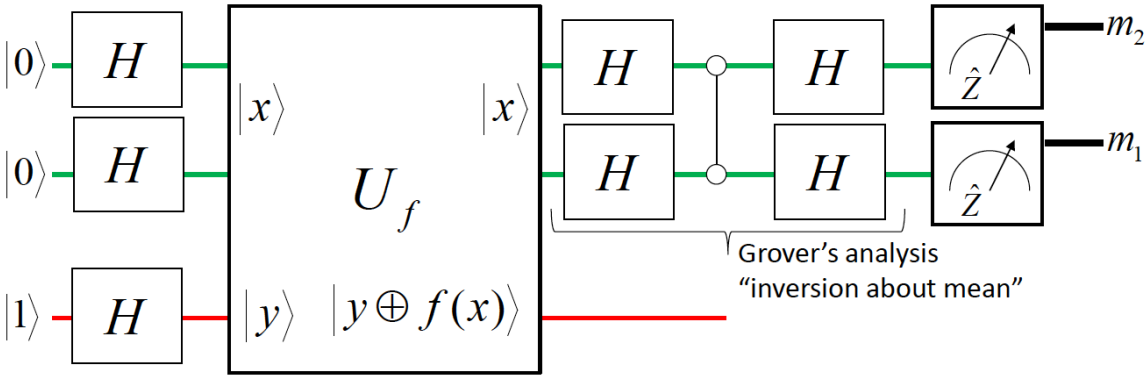
$$\frac{1}{2}\left(-|00\rangle+|01\rangle+|10\rangle-|11\rangle\right)$$

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



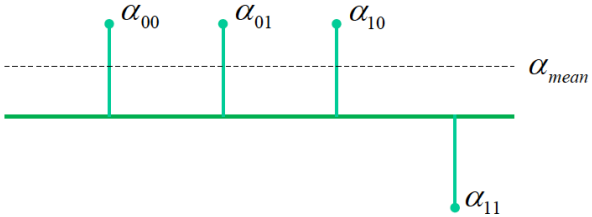
$$\doteq\begin{pmatrix} -1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

Grover's Analysis Step



Source:
Leo DiCarlo

Inversion about the Mean



$$\alpha_{mean} = \frac{1}{N}\sum \alpha_i$$

$$\alpha_i \rightarrow \alpha_i' = \alpha_{mean} - (\alpha_i - \alpha_{mean}) = 2\alpha_{mean} - \alpha_i$$

Case N=4:

$$M_{inv.\ about\ mean} = \frac{1}{2}\begin{pmatrix} -1 & 1 & 1 & 1 \\ 1 & -1 & 1 & 1 \\ 1 & 1 & -1 & 1 \\ 1 & 1 & 1 & -1 \end{pmatrix}$$

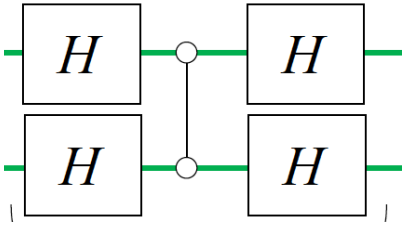
Source:
Leo DiCarlo

Verify: Inversion about the Mean

$$U_{Grover\ Analysis} = \frac{1}{4}\begin{pmatrix}1&1&1&1\\1&-1&1&-1\\1&1&-1&-1\\1&-1&-1&1\end{pmatrix}\begin{pmatrix}-1&0&0&0\\0&1&0&0\\0&0&1&0\\0&0&0&1\end{pmatrix}\begin{pmatrix}1&1&1&1\\1&-1&1&-1\\1&1&-1&-1\\1&-1&-1&1\end{pmatrix}$$

$$= \frac{1}{4}\begin{pmatrix}1&1&1&1\\1&-1&1&-1\\1&1&-1&-1\\1&-1&-1&1\end{pmatrix}\begin{pmatrix}-1&-1&-1&-1\\1&-1&1&-1\\1&1&-1&-1\\1&-1&-1&1\end{pmatrix}$$

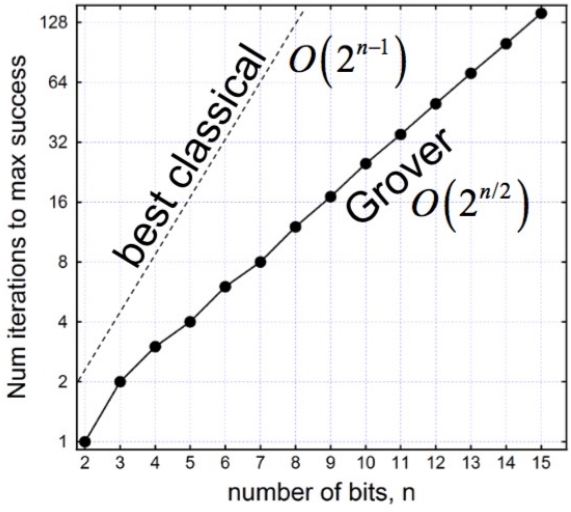
$$= \frac{1}{2}\begin{pmatrix}1&-1&-1&-1\\-1&1&-1&-1\\-1&-1&1&-1\\-1&-1&-1&1\end{pmatrix}$$



Source:
Leo DiCarlo

Grover's Algorithm

- Grover's algorithm grows (number of calls) with the square root of the number of bits
- The best classical algorithm still has a higher complexity



The Bernstein-Vazirani Problem

- Consider the 8-to-1 bit function:

$$y = f(a \bullet x)$$

$$a \bullet x \equiv a_8 x_8 \oplus a_7 x_7 \oplus a_6 x_6 \oplus a_5 x_5 \oplus a_4 x_4 \oplus a_3 x_3 \oplus a_2 x_2 \oplus a_1 x_1$$

$$a = a_8 a_7 a_6 a_5 a_4 a_3 a_2 a_1$$

$$x \equiv x_8 x_7 x_6 x_5 x_4 x_3 x_2 x_1$$

- Example

a = 1 1 0 1 0 0 1 1

x = 0 1 1 1 0 0 1 0

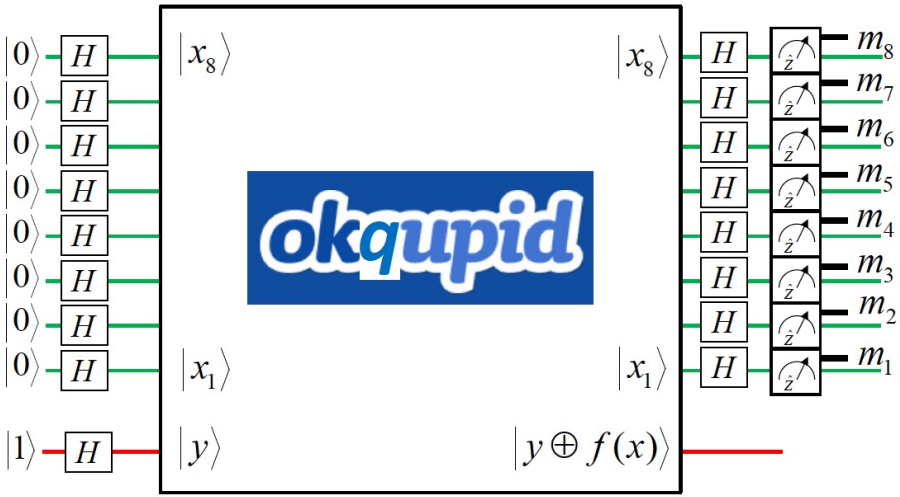
$f(x)=x \bullet a$ = (0+1+0+1+0+0+1+0) mod 2 =1

- You're given the **classical** and **quantum black boxes**.
- **Challenge:** Find ***a***

Source:
Leo DiCarlo

The Bernstein-Vazirani Problem

- Execute this sequence calling the quantum black box **once.**

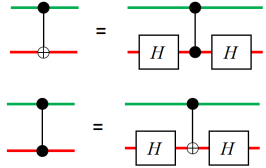


- Answer:

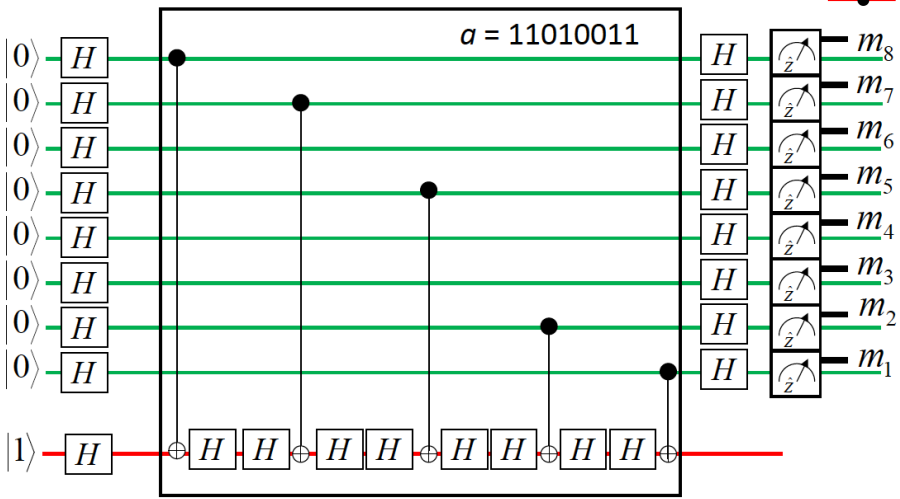
$$m_i = +1 \Rightarrow a_i = 0, \qquad m_i = -1 \Rightarrow a_i = 1$$

Source:
Leo DiCarlo

The Bernstein-Vazirani Problem



- Execute this sequence calling the quantum black box **once.**

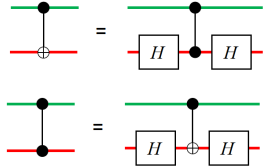


- Answer:

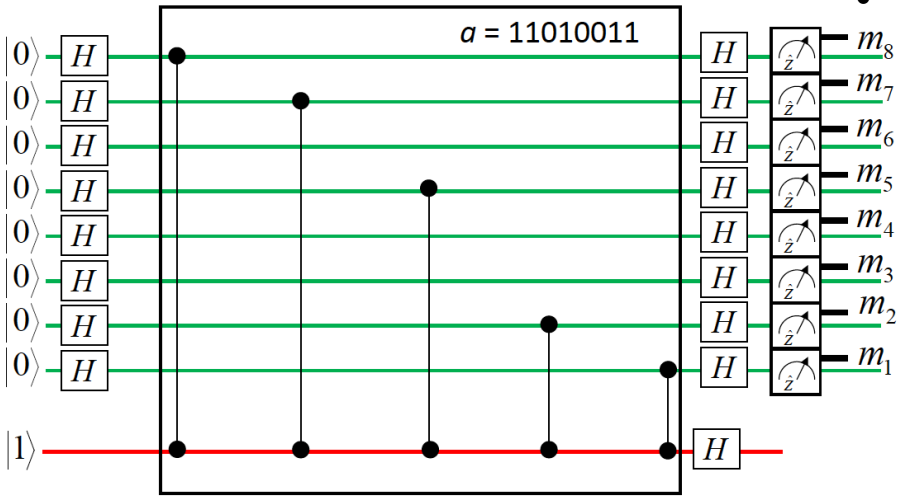
$$m_i = +1 \Rightarrow a_i = 0, \qquad m_i = -1 \Rightarrow a_i = 1$$

Source:
Leo DiCarlo

The Bernstein-Vazirani Problem



- Execute this sequence calling the quantum black box **once.**



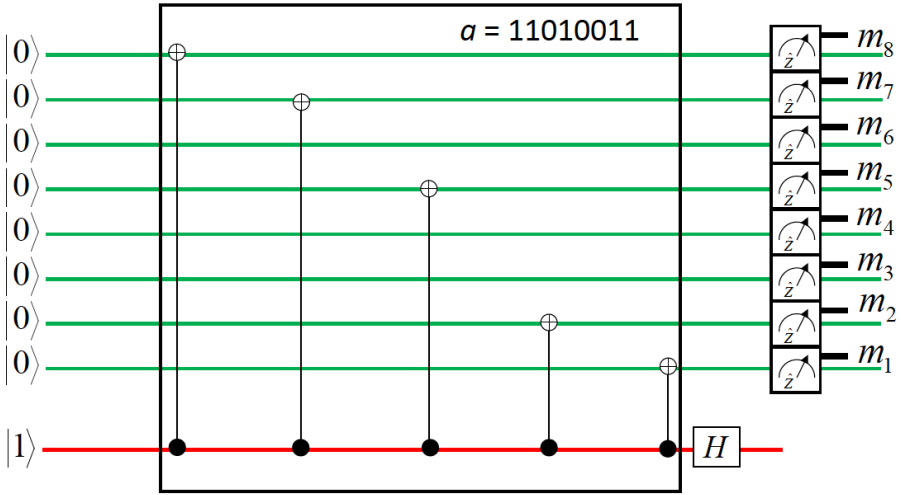
- Answer:

$$m_i = +1 \Rightarrow a_i = 0, \qquad m_i = -1 \Rightarrow a_i = 1$$

Source:
Leo DiCarlo

The Bernstein-Vazirani Problem

- Execute this sequence calling the quantum black box **once.**



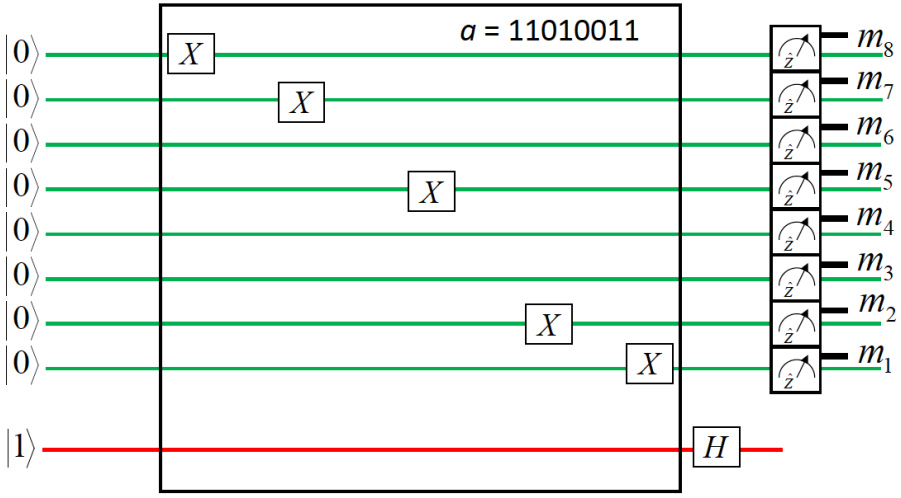
- Answer:

$$m_i = +1 \Rightarrow a_i = 0, \qquad m_i = -1 \Rightarrow a_i = 1$$

Source:
Leo DiCarlo

The Bernstein-Vazirani Problem

- Execute this sequence calling the quantum black box **once.**



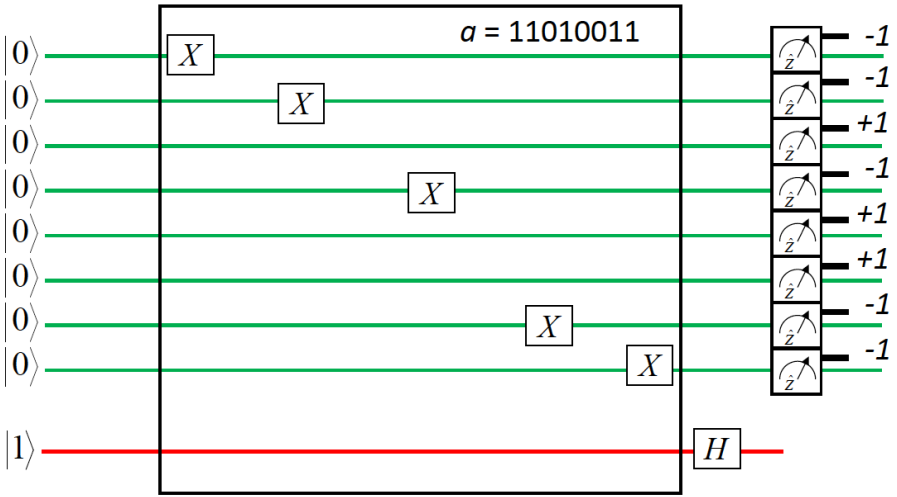
- Answer:

$$m_i = +1 \Rightarrow a_i = 0, \qquad m_i = -1 \Rightarrow a_i = 1$$

Source:
Leo DiCarlo

The Bernstein-Vazirani Problem

- Execute this sequence calling the quantum black box **once.**



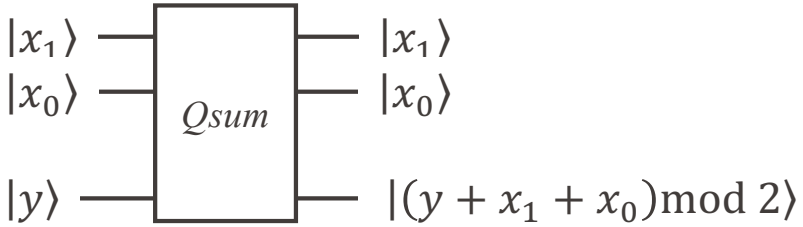
- Answer:

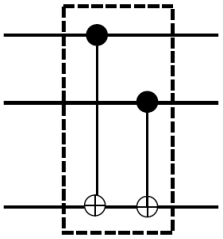
$$m_i = +1 \Rightarrow a_i = 0, \qquad m_i = -1 \Rightarrow a_i = 1$$

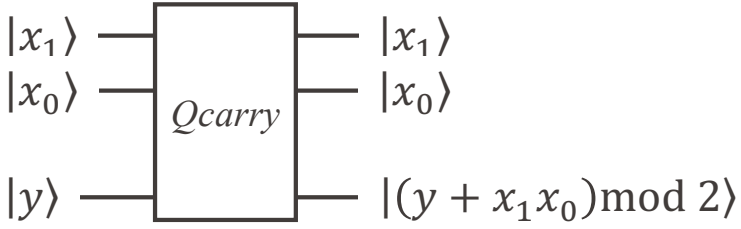
Source:
Leo DiCarlo

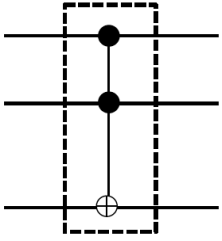
Quantum Arithmetic

Quantum Arithmetic







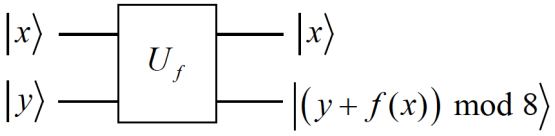


Example

$f(x) = x^2 \mod 8,$

with

$x = x_1 x_0$ a 2-bit number



Is M_f unitary?

y \ x	0	1	2	3
0	0	1	4	1
1	1	2	5	2
2	2	3	6	3
3	3	4	7	4
4	4	5	0	5
5	5	6	1	6
6	6	7	2	7
7	7	0	3	0

$(y + f(x)) \mod 8$

Yes! Orthogonal in means orthogonal out.

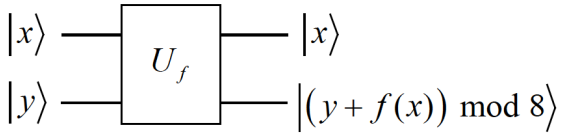
Source: Leo DiCarlo

Example (Cont)

$f(x) = x^2 \mod 8,$

with

$x = x_1 x_0$ a 2-bit number



$x_1 x_0$

		x_1	x_0
$\times$		x_1	x_0
		$x_1 x_0$	x_0
+	x_1	$x_1 x_0$	0
$x_1 x_0 \oplus x_1$		0	x_0
	$x_1 \overline{x}_0$	0	x_0

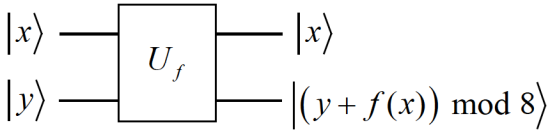
Source:
Leo DiCarlo

Example (Cont)

$f(x) = x^2 \mod 8,$

with

$x = x_1 x_0$ a 2-bit number



	$x_0 y_0 y_1$	$x_0 y_0$	
	$x_1 \overline{x}_0$	0	x_0
$+$	y_2	y_1	y_0
	$x_0 y_0 y_1 \oplus x_1 \overline{x}_0 \oplus y_2$	$x_0 y_0 \oplus y_1$	$x_0 \oplus y_0$

So we need a circuit that implements the unitary transformation:

$$
\begin{aligned}
x_1 &\longrightarrow x_1 \\
x_0 &\longrightarrow x_0 \\
y_2 &\longrightarrow x_0 y_0 y_1 \oplus x_1 \overline{x}_0 \oplus y_2 \\
y_1 &\longrightarrow x_0 y_0 \oplus y_1 \\
y_0 &\longrightarrow x_0 \oplus y_0
\end{aligned}
$$

Source:
Leo DiCarlo

Example (Cont)

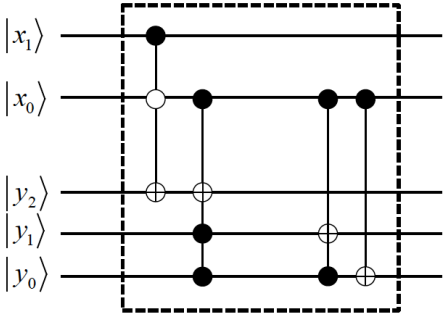
$$x_1 \longrightarrow x_1$$

$$x_0 \longrightarrow x_0$$

$$y_2 \longrightarrow x_0 y_0 y_1 \oplus x_1 \overline{x}_0 \oplus y_2$$

$$y_1 \longrightarrow x_0 y_0 \oplus y_1$$

$$y_0 \longrightarrow x_0 \oplus y_0$$

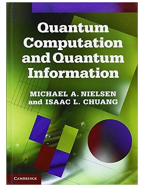


Note:
- ● = active high
- ○ =active low

Source:
Leo DiCarlo

Quantum Fourier Transform

Chapter 5



Conventional Discrete Fourier Transform

- Classical DFT:

  Let input vector of complex numbers $x_0, \ldots, x_{N-1}$, where N is fix.
- DCT will output a vector $y_0, \ldots, y_{N-1}$, defined by

$$y_k = \frac{1}{\sqrt{N}} \sum_{j=0}^{N-1} x_j e^{2\pi i j k / N}$$

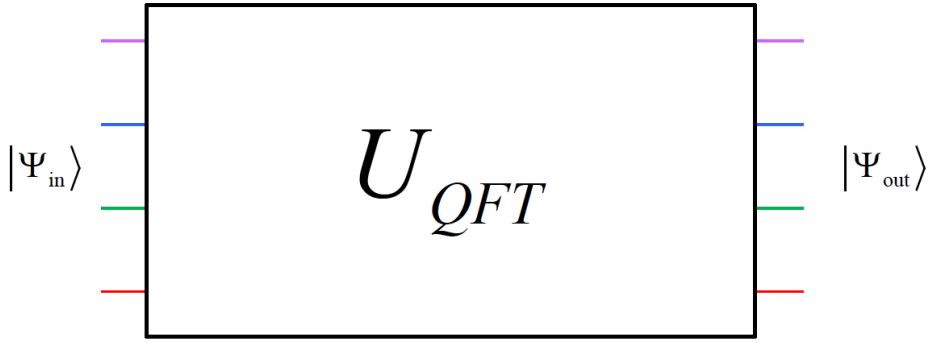
Quantum Fourier Transform

- The quantum Fourier transform is exactly the same, except that it transforms a qubit register k into j, whereas we use the orthonormal basis $|0\rangle, \ldots, |N-1\rangle$, where N is also fix.
- The QFT is defined by

$$|j\rangle \rightarrow \frac{1}{\sqrt{N}} \sum_{k=0}^{N-1} e^{2\pi i j k/N} |k\rangle$$

- The QFT is a unitary transformation.

Quantum Fourier Transform



$$|\Psi_{out}\rangle = \left(\frac{1}{\sqrt{N}} \sum_{l=0}^{N-1} \sum_{k=0}^{N-1} e^{\frac{i2\pi lk}{N}} |l\rangle\langle k| \right) |\Psi_{in}\rangle$$

$$\alpha_l' = \frac{1}{\sqrt{N}} \sum_{k=0}^{N-1} e^{\frac{i2\pi lk}{N}} \alpha_k$$

Source:
Leo DiCarlo

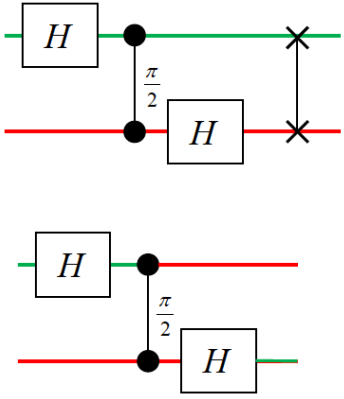
Examples

- $N = 2$

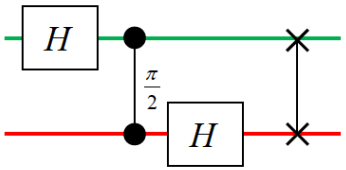
$$U_{QFT}\frac{1}{\sqrt{2}}\begin{pmatrix}1 & 1\\ 1 & -1\end{pmatrix} = H$$

- $N = 4$

$$U_{QFT}\frac{1}{\sqrt{4}}\begin{pmatrix}1 & 1 & 1 & 1\\ 1 & +i & -1 & -i\\ 1 & -1 & 1 & -1\\ 1 & -i & -1 & +i\end{pmatrix}$$



Verification



$$\doteq \frac{1}{2}\begin{pmatrix}1&0&0&0\\0&0&1&0\\0&1&0&0\\0&0&0&1\end{pmatrix}\begin{pmatrix}1&1&0&0\\1&-1&0&0\\0&0&1&1\\0&0&1&-1\end{pmatrix}\begin{pmatrix}1&0&0&0\\0&1&0&0\\0&0&1&0\\0&0&0&i\end{pmatrix}\begin{pmatrix}1&0&1&0\\0&1&0&1\\1&0&-1&0\\0&1&0&-1\end{pmatrix}$$

swap H on LSB C-PHASE H on MSB

$$= \frac{1}{2}\begin{pmatrix}1&0&0&0\\0&0&1&0\\0&1&0&0\\0&0&0&1\end{pmatrix}\begin{pmatrix}1&1&0&0\\1&-1&0&0\\0&0&1&1\\0&0&1&-1\end{pmatrix}\begin{pmatrix}1&0&1&0\\0&1&0&1\\1&0&-1&0\\0&i&0&-i\end{pmatrix}$$

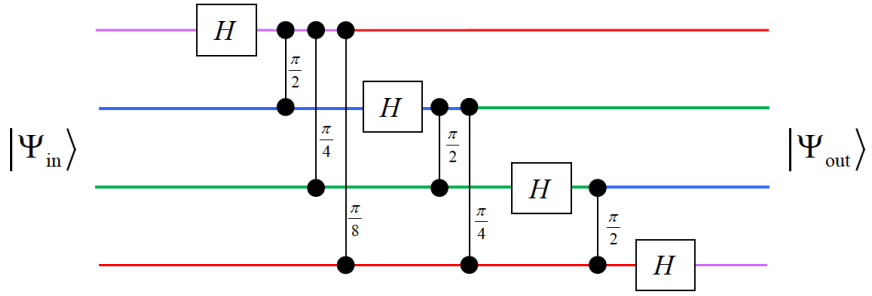
$$= \frac{1}{2}\begin{pmatrix}1&0&0&0\\0&0&1&0\\0&1&0&0\\0&0&0&1\end{pmatrix}\begin{pmatrix}1&1&1&1\\1&-1&1&-1\\1&i&-1&-i\\1&-i&-1&i\end{pmatrix}$$

$$= \frac{1}{2}\begin{pmatrix}1&1&1&1\\1&i&-1&-i\\1&-1&1&-1\\1&-i&-1&i\end{pmatrix} \doteq U_{QFT}$$

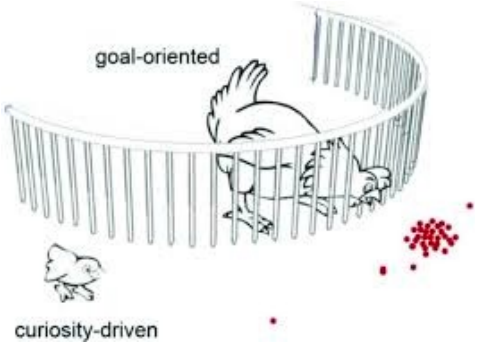
Source:
Leo DiCarlo

Complexity of Quantum Fourier Transform

- QFT requires $1 + 2 + 3 + \cdots + n = n(n+1)/2$ gates, so it is $O(n^2)$.
- If the number of elementary operations necessary to execute a quantum circuit grows polynomially with the number of qubits (n), then it is said to be fast or efficient.



Thank you



T. Haensch