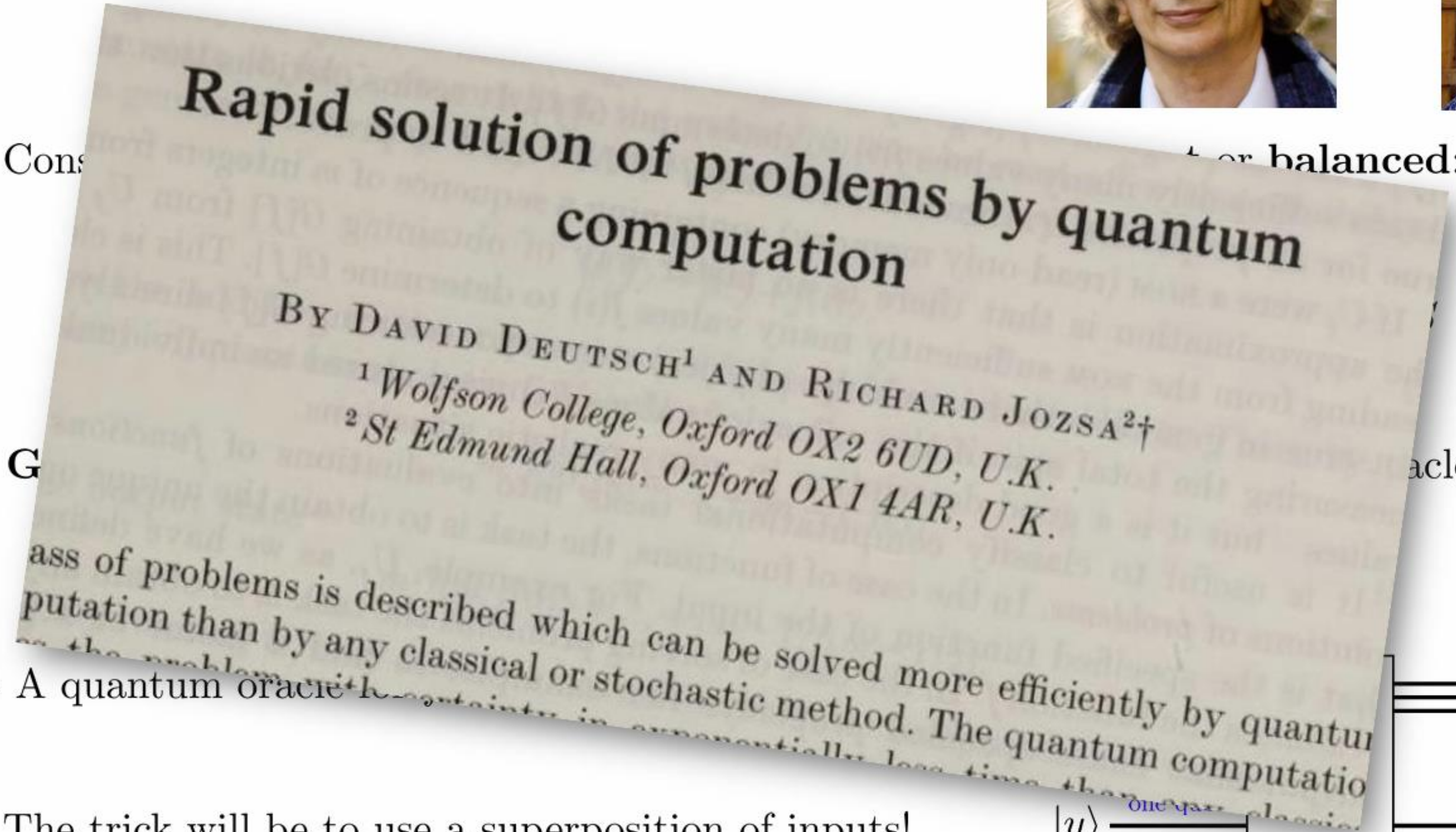
Department of Information Engineering
MSc in Computer Engineering (a.y. 2024/2025)
University of Pisa

Quantum Computing and Quantum Internet

Luciano Lenzini
Full Professor
Department of Information Engineering
School of Engineering
University of Pisa, Italy
e-mail: lenzini44@gmail.com
http://www.iet.unipi.it/~lenzini/
http://www.originiinternetitalia.it/it/

Deutsch-Jozsa Problem

- Cons… …t or **balanced**:



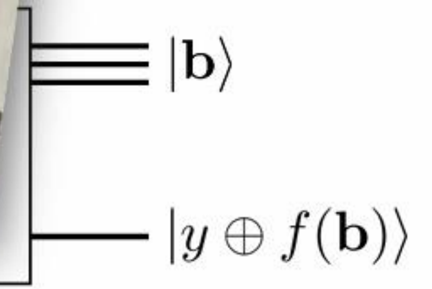
Rapid solution of problems by quantum computation

BY DAVID DEUTSCH[1] AND RICHARD JOZSA[2]†
[1] *Wolfson College, Oxford OX2 6UD, U.K.*
[2] *St Edmund Hall, Oxford OX1 4AR, U.K.*

…ass of problems is described which can be solved more efficiently by quantu…
…putation than by any classical or stochastic method. The quantum computatio…

- **G**… …acle.

- A quantum oracle…



- The trick will be to use a superposition of inputs!

Deutsch Quantum Algorithm

Deutsch’s Problem

- At this stage, we have acquired the concepts and tools needed to decide whether a given function has a certain property using the *Deutsch quantum algorithm*
- This computation cannot be solved as efficiently using any classical computer
- It's not a particularly useful calculation, mind you, because it's pretty contrived
- Nevertheless, it highlights three key concepts of quantum computation: *quantum parallelism, quantum interference,* and the *phase kick-back*

The Problem: Is *f* (*x*) Constant or Balanced?

- Suppose we are given a *black box* that computes some *function*, $f(x)$, even one that may be initially unknown to us

$$x \;—\!\!—\; \boxed{\;f\;} \;—\!\!—\; f(x)$$

- The term *black box* suggests that we don't know what's on the inside or how it works

- Suppose we are dealing with *classical function* which takes a single classical bit in (one bit domain) and produces a single classical bit out (one bit range)

$$f:\{0,1\} \to \{0,1\}$$

The Problem: Is $f(x)$ Constant or Balanced?

- There are only a small number of functions that can act on the set $x \in \{0,1\}$ and give a single bit as output (i.e. one bit domain and range)
- For example, we could have the *identity* function

$$f(x) = \begin{cases} 0 & \text{If} \quad x=0 \\ 1 & \text{If} \quad x=1 \end{cases}$$

- Two more examples are the *constant* functions

$$f(0) = 0,\ f(1) = 0$$

$$f(0) = 1,\ f(1) = 1$$

The Problem: Is $f\,(x)$ Constant or Balanced?

- The final example is the *bit flip* function

$$f(x) = \begin{cases} 1 & \text{If} \quad x=0 \\ 0 & \text{If} \quad x=1 \end{cases}$$

- The *identity* and *bit flip* functions are called *balanced* because the outputs are opposite for half the inputs
- So, a **function on a single bit** can be *constant* or *balanced*

The Problem: Is $f\,(x)$ Constant or Balanced?

$f\,(x)$

- *balanced*
  - $f(0)=0$, $f(1)=1$
  - $f(0)=1$, $f(1)=0$
- *Constant*
  - $f(0)=0$, $f(1)=0$
  - $f(0)=1$, $f(1)=1$

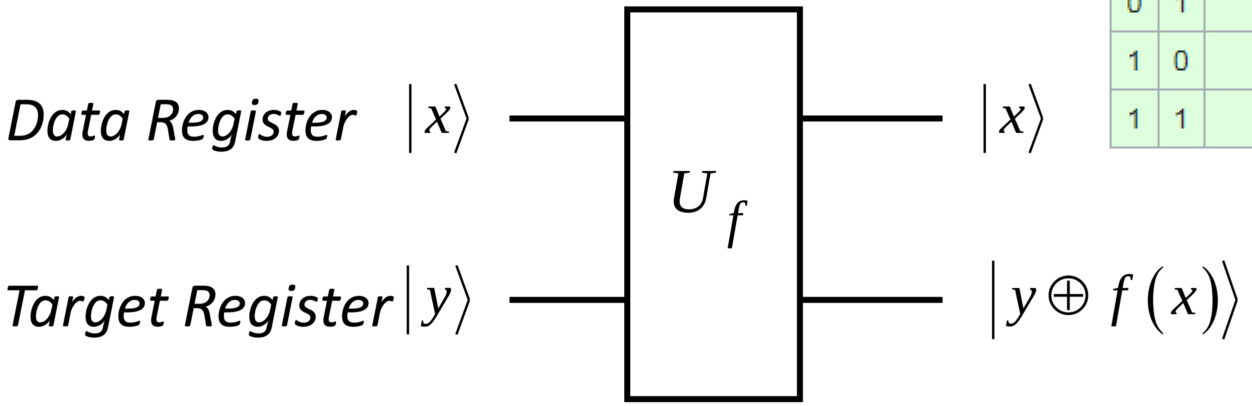
Embedding $f(x)$ in a Quantum Black-Box

- A convenient way of *computing this function* on a **quantum computer** is to consider a two-qubit quantum computer which starts in the state $|x\rangle|y\rangle$

- With an appropriate sequence of gates, it is possible to transform $|x\rangle|y\rangle$ into $|x\rangle|y \oplus f(x)\rangle$, where $\oplus$ indicates addition modulo 2 (*XOR*)

Input		Output
A	B	A XOR B
0	0	0
0	1	1
1	0	1
1	1	0

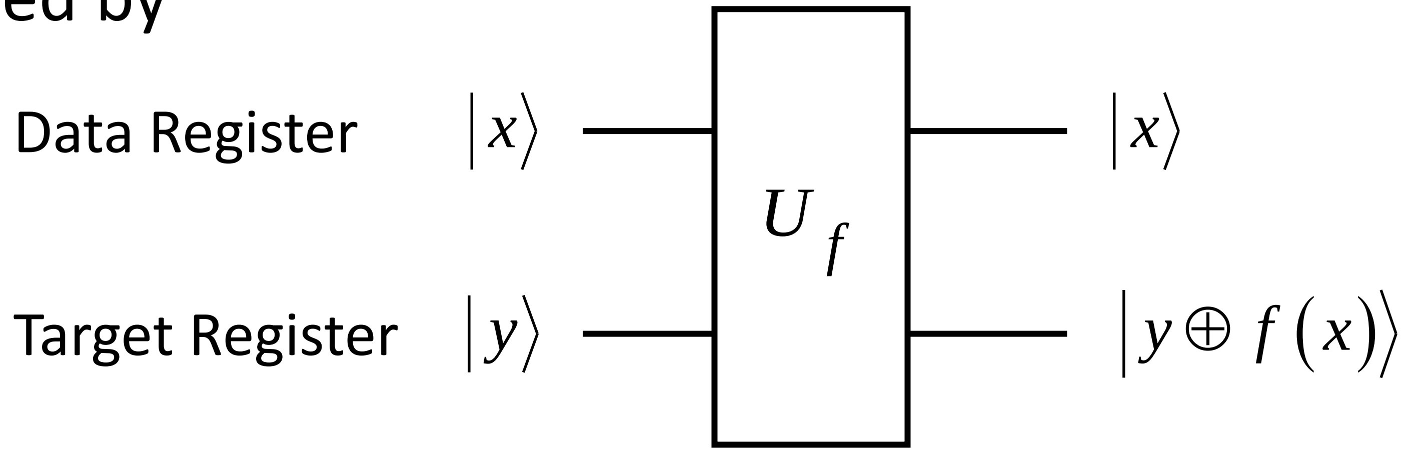
- The first register is called the *'data' register*, and the second register the *'target' register*



Quantum Oracle for $f(x)$

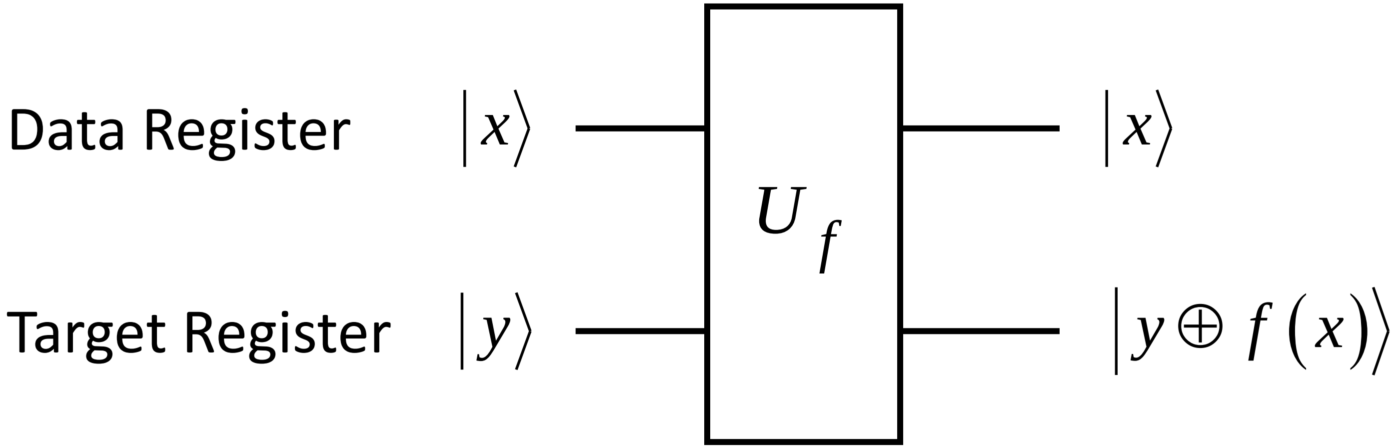
- We won't describe how this works but, instead, take it as a given and call the new, larger circuit “U_f“ the quantum oracle for f

- Its action on *computation basis states* (*CBS*) and its circuit diagram are defined by



Quantum Oracle for *f*(*x*)

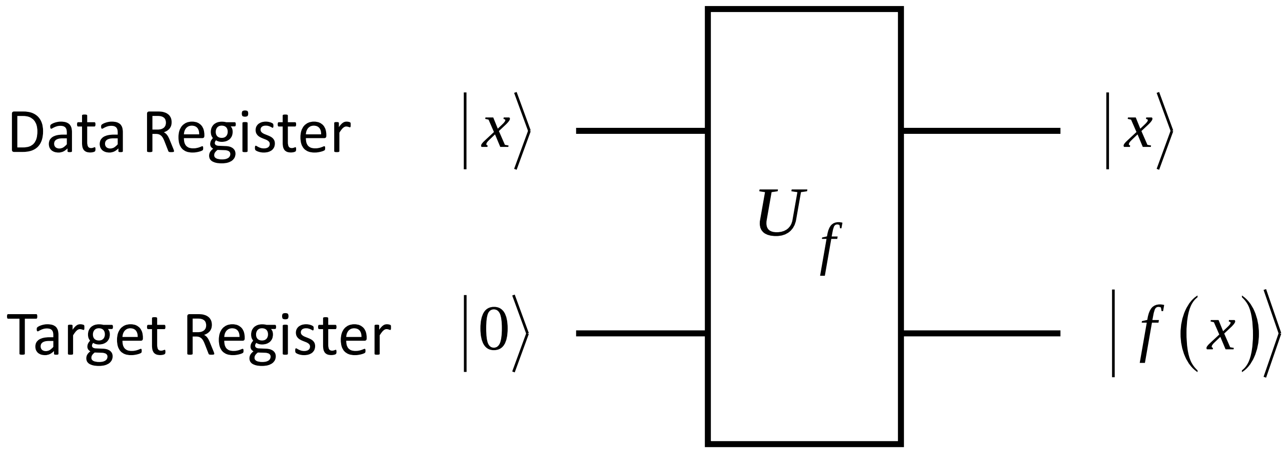
- First, notice that the output of the target register is a CBS; inside the ket we are *XOR*-ing $(\oplus)$ two classical binary values *y* and *f*(*x*), producing another binary value which, in turn, defines a CBS ket: either $|0\rangle$ or $|1\rangle$



Notice that output of U_f for a CBS is always a separable state $|y \oplus f(x)\rangle$; since $y \oplus f(x)$ is always either 0 or 1

Quantum Oracle for $f(x)$

- U_f computes $f(x)$
- This is a consequence of the circuit definition, because if we plug, $0 \to y$ we get



Deutsch's Problem

- Given an *unknown function* with one-bit domain and range, that we are told is either *balanced* or *constant*, determine in *one query* of the quantum oracle, whether f is *constant*

- Notice that we are not asking to determine the exact function, just which *category* it belongs to

- **Even so, we cannot do it classically in a single query**

Deutsch's Algorithm

- The algorithm consists of building a circuit and measuring the *Data Register* - *once*

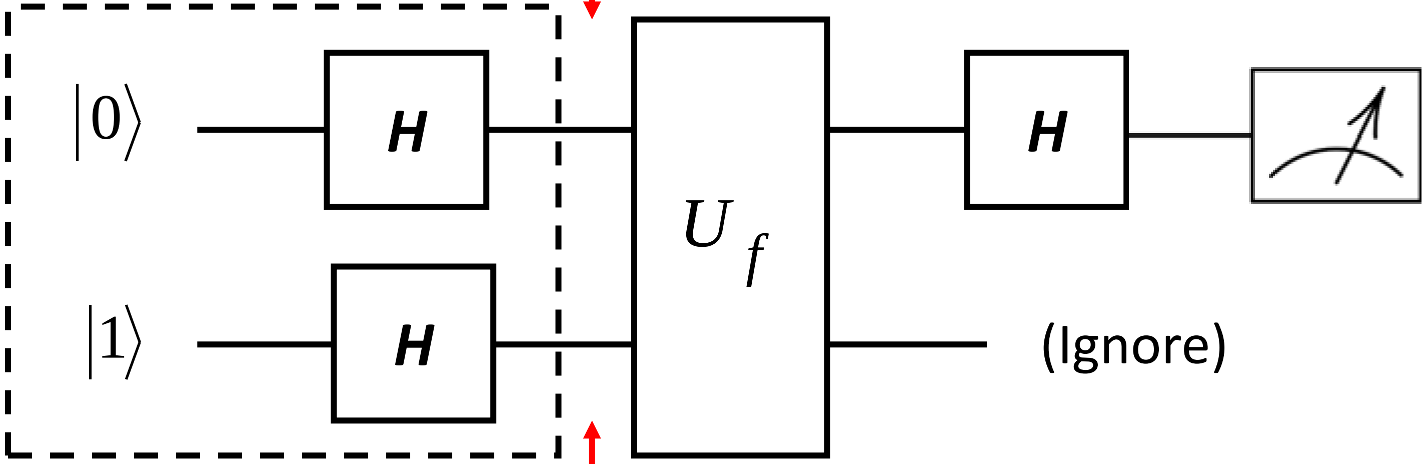
- Our conclusion about *f* is determined by the result of the measurement

- If we get a 0 the function is *constant*, if we get 1 the function is *balanced*

Preparing the Oracle's Input

- We combine the quantum oracle for f with a few *Hadamard* gates in a very small circuit

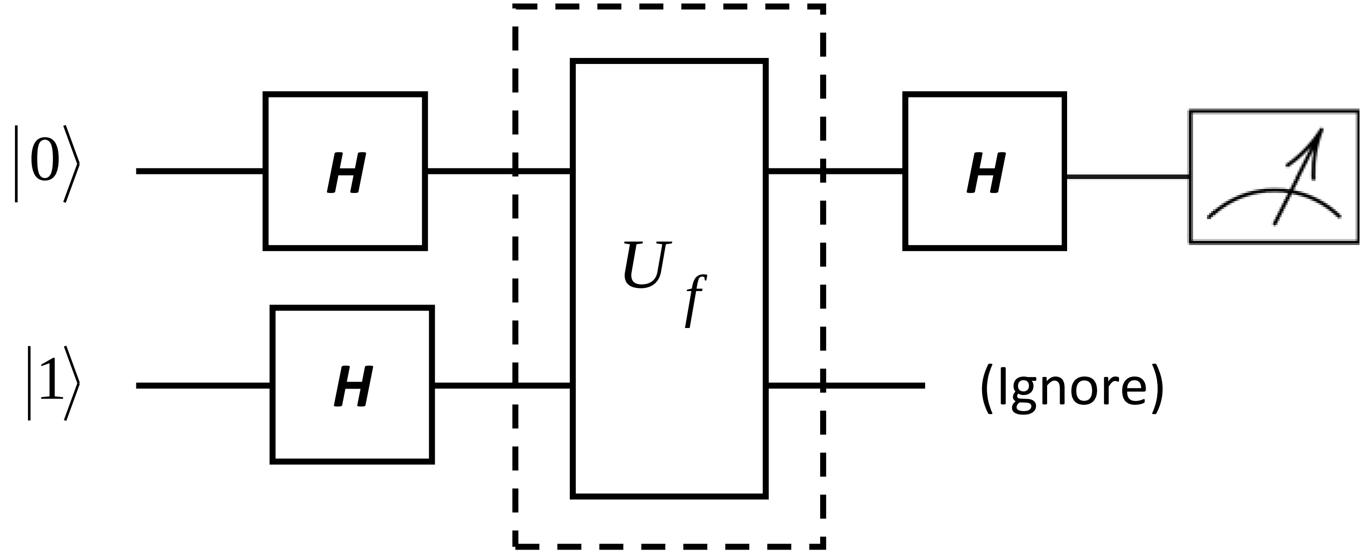
$$H|0\rangle = \frac{1}{\sqrt{2}}\left(|0\rangle + |1\rangle\right) = |+\rangle$$



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

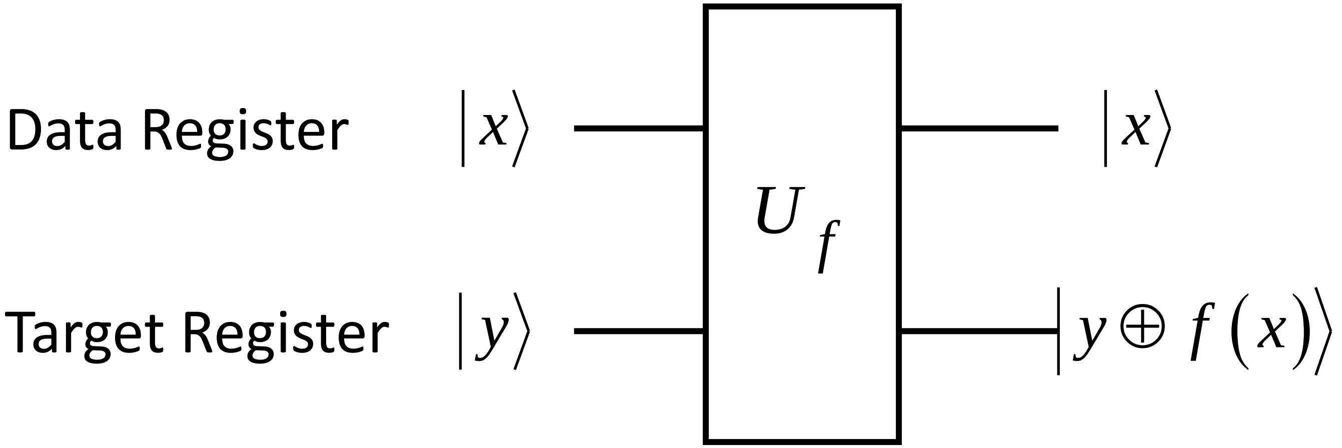
Analyzing the Oracle

- The real understanding of how the algorithm works comes by analyzing the kernel of the circuit, the oracle (in the dashed-box),



Analyzing the Oracle

- Step 1. CBS Into Both Channels



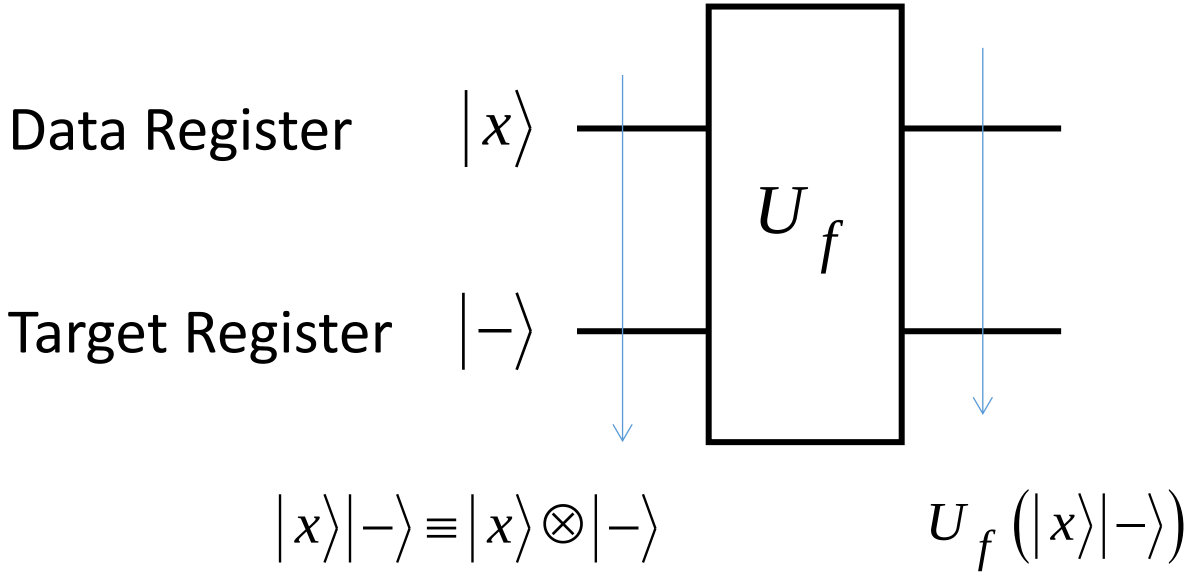
- Algebraically

$$|x\rangle|y\rangle \rightarrow U_f\left(|x\rangle|y\rangle\right) = |x\rangle|y \oplus f(x)\rangle \quad \longrightarrow$$

Both, input and output of the quantum oracle are separable

Analyzing the Oracle

- Step 2. CBS Into Data and Superposition into Target
- We stick with a CBS $|x\rangle$ going into the data register, but now allow the superposition $|-\rangle$ go into the target register



Analyzing the Oracle

- Extend the above linearly,

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

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

$$=\frac{1}{\sqrt{2}}\left(|x\rangle|f(x)\rangle-|x\rangle|\overline{f(x)}\rangle\right)$$

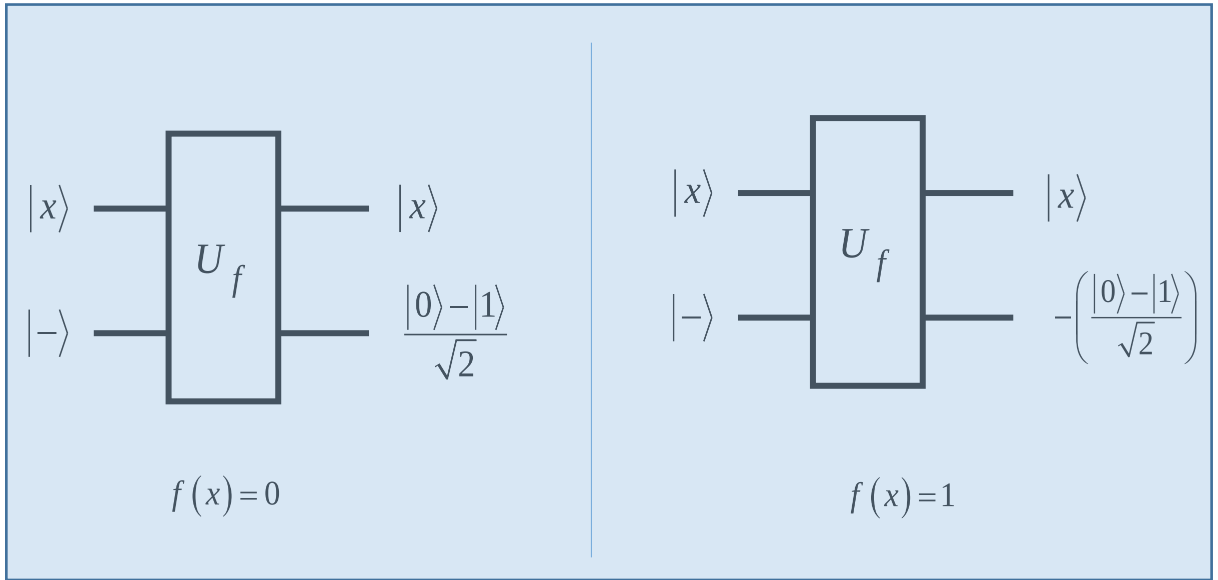
$$=|x\rangle\left(\frac{|f(x)\rangle-|\overline{f(x)}\rangle}{\sqrt{2}}\right)$$

Analyzing the Oracle

- This amounts to

$$U_f\left(|x\rangle|-\rangle\right)=|x\rangle\begin{cases}\dfrac{|0\rangle-|1\rangle}{\sqrt{2}} & \text{when} \quad f(x)=0\\ -\left(\dfrac{|0\rangle-|1\rangle}{\sqrt{2}}\right) & \text{when} \quad f(x)=1\end{cases}$$

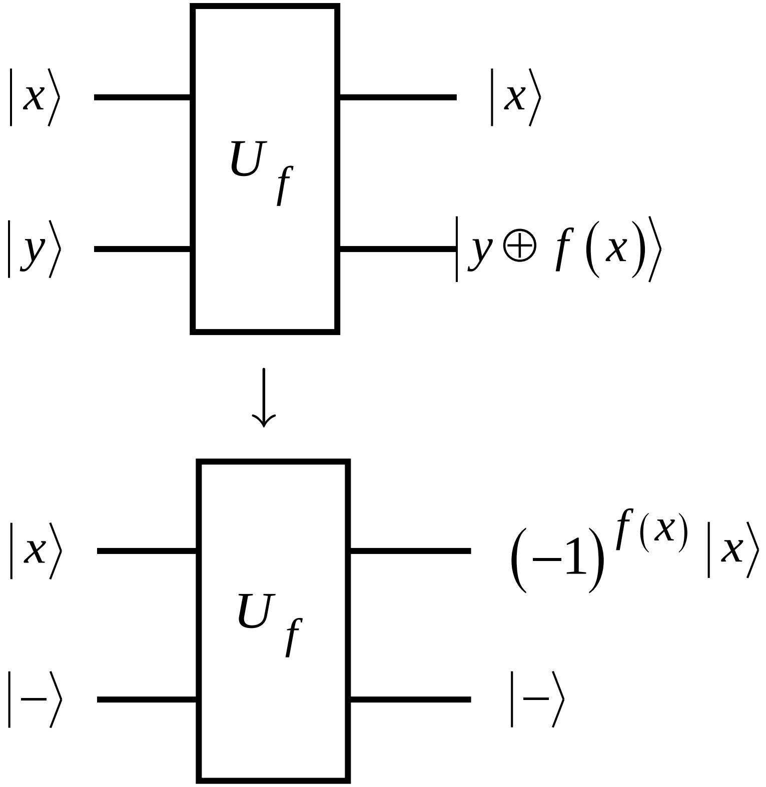
$$=|x\rangle(-1)^{f(x)}\left(\frac{|0\rangle-|1\rangle}{\sqrt{2}}\right)$$



Analyzing the Oracle

Since it's a scalar, $(-1)^{f(x)}$ can be moved to the left and be attached to the Data Register's $|x\rangle$, a mere rearrangement of the terms

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



and we have successfully *(like magic)* moved all of the information about *f*(x) from the Target Register to the Data Register, where it appears as an *overall phase factor* in the scalar's exponent

Analyzing the Oracle

Let's pause to summarize what we have accomplished so far

- We have proven that $|x\rangle|-\rangle$ is an eigenvector of U_f with eigenvalue $(-1)^{f(x)}$ for *x* = 0, 1

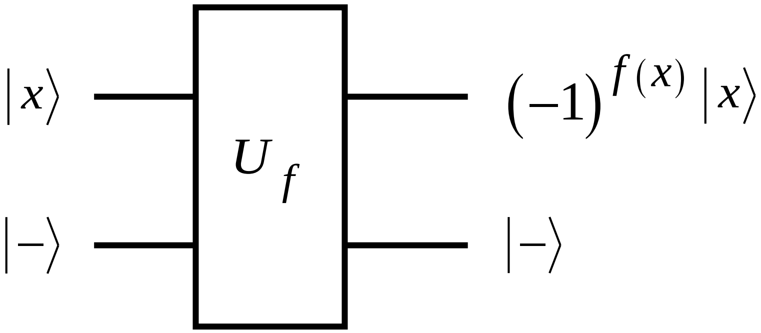
$$U_f\left(|x\rangle|-\rangle\right)=\left((-1)^{f(x)}|x\rangle\right)|-\rangle=(-1)^{f(x)}\left(|x\rangle|-\rangle\right)$$

Analyzing the Oracle

- The information about $f(x)$ is encoded *"kicked-back"* in the Data Register's output

- That's where we plan to look for it in the coming step

- Viewed this way, the **Target Register** retains *no useful* information

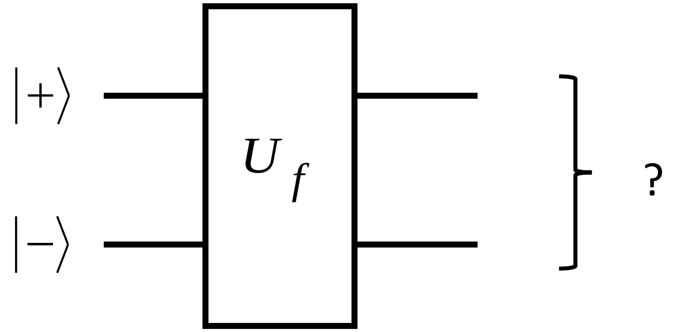


Analyzing the Oracle

Step 3. Superpositions into Both Registers

- Finally, we want the state $|+\rangle$ to go into the data register so we can process both *f*(0) and *f*(1) in a single pass

- The effect is to present the separable $|+\rangle \otimes |-\rangle$ to the oracle and see what comes out



Analyzing the Oracle

Applying linearity to the last result we get

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

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

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

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

Analyzing the Oracle

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

This is a remarkable state! The **different terms** contain information about both $f(0)$ and $f(1)$; it is almost as if we have evaluated $f(x)$ for two values of x *simultaneously*, a feature known as **QUANTUM PARALLELISM**

Thus, by combining the *phase kick-back* with *quantum parallelism*, we've managed to get an expression containing both $f(0)$ and $f(1)$ in the Data Register

Analyzing the Oracle

- Unlike *classical parallelism*, where *multiple* circuits each built to compute $f(x)$ are executed simultaneously, here a *single* $f(x)$ circuit is employed to evaluate the function for *multiple values of x simultaneously*, by exploiting the ability of a qubit to be in **superpositions of different states**

Analyzing the Oracle

- Question: what is the difference between the *balanced case* $\left(f(0) \neq f(1)\right)$ and the *constant case* $\left(f(0) = f(1)\right)$?

- Answer: when **constant**, the two terms in the numerator have the same sign and when **balanced**, they have different signs, to wit

$$\frac{(-1)^{f(0)}|0\rangle + (-1)^{f(1)}|1\rangle}{\sqrt{2}}|-\rangle \quad \longrightarrow \quad U_f\left(|+\rangle|-\rangle\right) = \begin{cases} (\pm 1)\dfrac{|0\rangle + |1\rangle}{\sqrt{2}}|-\rangle & \text{if} \quad f(0) = f(1) \\ (\pm 1)\dfrac{|0\rangle - |1\rangle}{\sqrt{2}}|-\rangle & \text{if} \quad f(0) \neq f(1) \end{cases}$$

Analyzing the Oracle

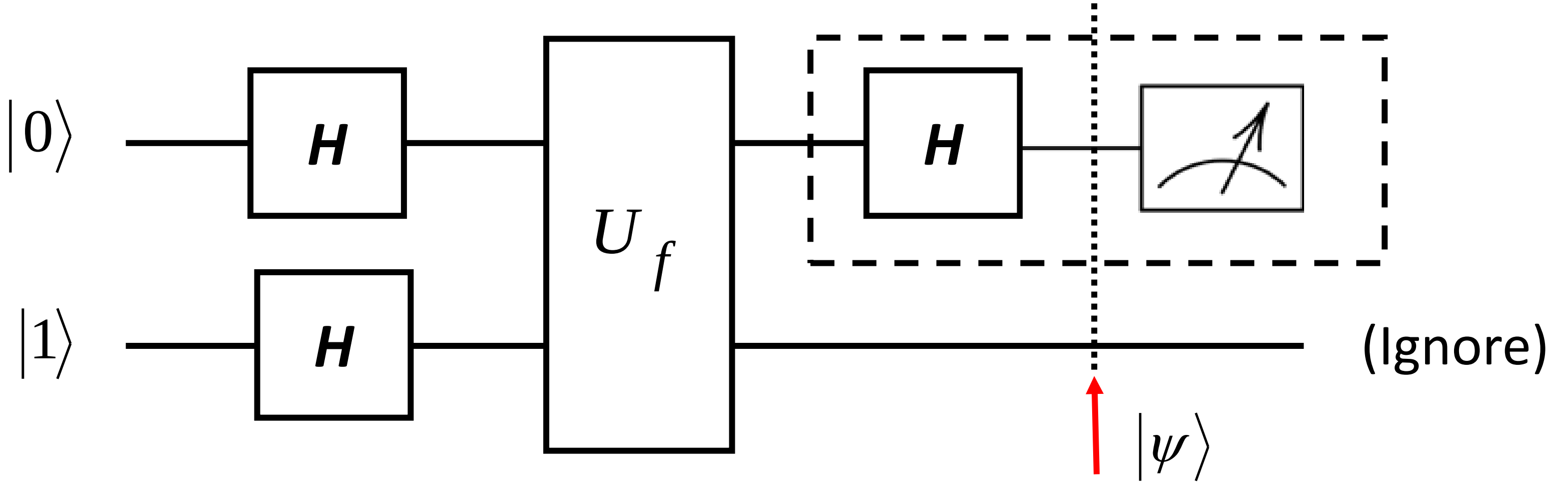
- We don't care about a possible overall phase factor or (-1) in front of all this since it's a unit scalar in a state space

- Thus, in the following we dump it

$$\frac{(-1)^{f(0)}|0\rangle + (-1)^{f(1)}|1\rangle}{\sqrt{2}}|-\rangle \quad \longrightarrow \quad U_f\left(|+\rangle|-\rangle\right) = \begin{cases} \frac{|0\rangle + |1\rangle}{\sqrt{2}}|-\rangle & \text{if } \; f(0) = f(1) \\ \frac{|0\rangle - |1\rangle}{\sqrt{2}}|-\rangle & \text{if } \; f(0) \neq f(1) \end{cases}$$

Measurement

- The final Hadamard gate on the Target Register thus gives



$$\frac{|0\rangle+|1\rangle}{\sqrt{2}}|-\rangle \to |\psi\rangle = (H\otimes I)\left(\frac{|0\rangle+|1\rangle}{\sqrt{2}}|-\rangle\right) = \left(H\left(\frac{|0\rangle+|1\rangle}{\sqrt{2}}\right)\right)\otimes\left(I|-\rangle\right) = |0\rangle|-\rangle \qquad \text{if} \quad f(0)=f(1)$$

$$\frac{|0\rangle-|1\rangle}{\sqrt{2}}|-\rangle \to |\psi\rangle = (H\otimes I)\left(\frac{|0\rangle-|1\rangle}{\sqrt{2}}|-\rangle\right) = \left(H\left(\frac{|0\rangle-|1\rangle}{\sqrt{2}}\right)\right)\otimes\left(I|-\rangle\right) = |1\rangle|-\rangle \qquad \text{if} \quad f(0)\neq f(1)$$

Measurement

$$|\psi\rangle = |0\rangle|-\rangle \quad \text{if} \quad f(0) = f(1)$$
$$|\psi\rangle = |1\rangle|-\rangle \quad \text{if} \quad f(0) \neq f(1)$$

- Realizing that

$$f(0) \oplus f(1) = \begin{cases} 0 & \text{if} \quad f(0) = f(1) \\ 1 & \text{if} \quad f(0) \neq f(1) \end{cases}$$

we can rewrite the above result

$$|\psi\rangle = |f(0) \oplus f(1)\rangle \left[\frac{|0\rangle - |1\rangle}{\sqrt{2}}\right] = |f(0) \oplus f(1)\rangle|-\rangle$$

so, by measuring the *first qubit* we may determine

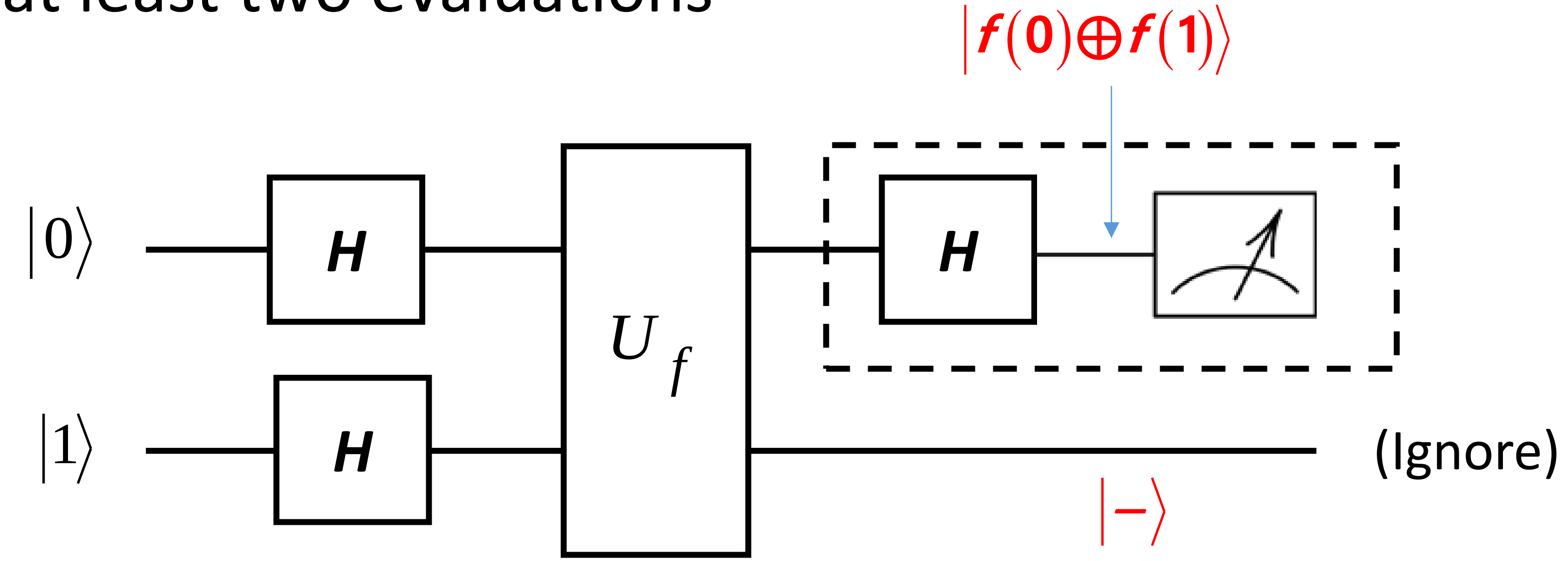
$$f(0) \oplus f(1)$$

We only care about the first qubit, since the second qubit will always collapse to $|-\rangle$

Measurement

- This is very interesting indeed: the quantum circuit has given us the ability to determine a *global property* of *f*(*x*), namely $f(0) \oplus f(1)$, using only *one* evaluation of *f*(*x*)!

- This is faster than is possible with a classical apparatus, which would require at least two evaluations



Measurement

- In a quantum computer it is possible for the two alternatives to interfere with one another to yield some global property of the function *f*, by using something like the Hadamard gate to recombine the different alternatives, as was done in Deutsch's algorithm

- The essence of the design of many quantum algorithms is that a clever choice of function and final transformation allows efficient determination of useful global information about the function - information which cannot be attained quickly on a classical computer

The Deutsch–Jozsa Algorithm

The Deutsch–Jozsa Algorithm

- The *Deutsch–Jozsa algorithm* solves a problem that is a straightforward generalization of the problem solved by the Deutsch algorithm

- As with the *Deutsch algorithm*, we are given a reversible circuit implementing an unknown function *f*, but this time *f* is a function from *n-bit strings* $x = x_1 x_2 x_3 \cdots x_n$ to *a single bit*. That is,

$$f : \{x_1, x_2, x_3, \cdots, x_n\} \to \{0,1\} \quad \text{where} \quad x_i \in \{0,1\}, \quad i \in \{1, 2, \cdots, n\}$$

The Deutsch–Jozsa Algorithm

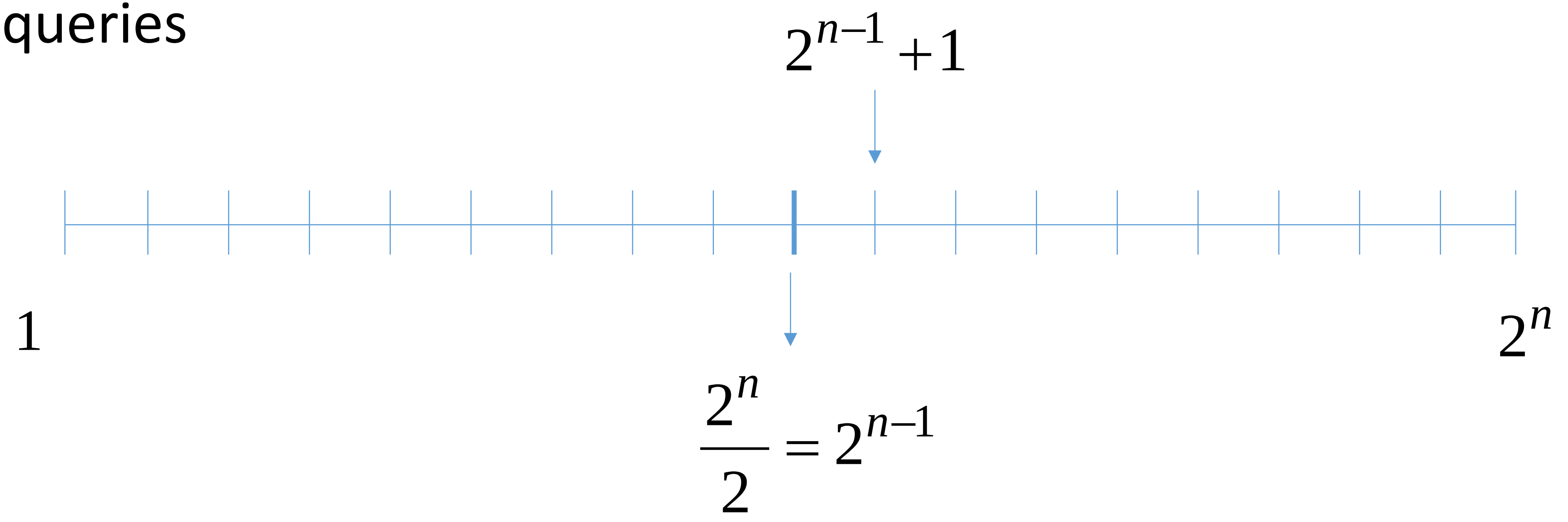
- We are also given the promise that
  - *f* is either *constant* (meaning *f*(*x*) is the same for all *x*), or
  - *f* is *balanced* (meaning *f*(*x*) = 0 for exactly half of the input strings x, and *f*(*x*) = 1 for the other half of the inputs)

- The problem here is to determine whether *f* is ***constant***, or ***balanced***, by making queries to the circuit for *f*

The Deutsch–Jozsa Algorithm

- Consider solving this problem by a classical algorithm

- In the worst case, using a classical algorithm we cannot decide with certainty whether *f* is constant or balanced using any less than 2^{n-1} + 1 queries



The Deutsch–Jozsa Algorithm

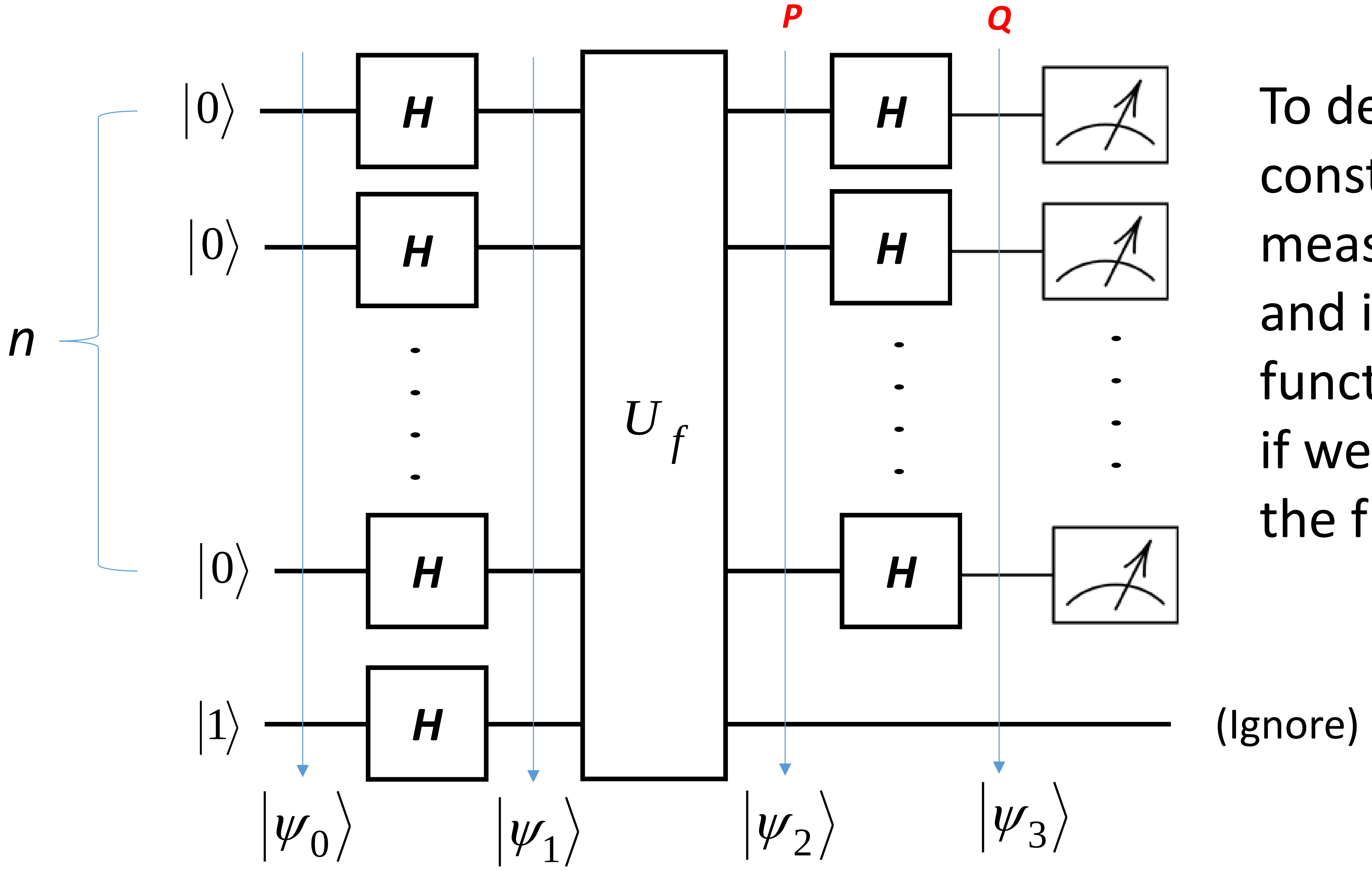
- The Deutsch-Jozsa algorithm is very similar to Deutsch's algorithm, but we now have *n qubits* plus the *target qubit*

- These *n qubits* are initially each in the $|0\rangle$ state, and we apply Hadamards to put them in a superposition of all *n*-bit strings

- Then we query the oracle on this superposition

- Finally, we apply Hadamards to all the qubits to create a state that we measure, and whose measurement outcome allows us to distinguish whether the function is constant or balanced

The Deutsch–Jozsa Algorithm



To determine if *f* is constant or balanced, we measure the *n qubits*, and if we get $|000\cdots0\rangle$, the function is **constant**, and if we get anything else, the function is **balanced**

The Deutsch–Jozsa Algorithm

- As we did for the Deutsch algorithm, we follow the state through the circuit. Initially the state is

$$|\psi_0\rangle = |0\rangle^{\otimes n}|1\rangle$$

- After the Hadamard transform on the query register and the Hadamard gate on the answer register we have

$$|\psi_1\rangle = H^{\otimes n+1}\left(|0\rangle^{\otimes n}|1\rangle\right) = \left(H^{\otimes n}\otimes H\right)\left(|0\rangle^{\otimes n}\otimes|1\rangle\right) = \left(H^{\otimes n}|0\rangle^{\otimes n}\right)\otimes\left(H|1\rangle\right)$$

The Deutsch–Jozsa Algorithm

- Consider the action of an n-qubit Hadamard transformation on the state $|0\rangle^{\otimes n}$

$$H^{\otimes n}|0\rangle^{\otimes n} = \underbrace{(H \otimes H \otimes \cdots \otimes H)}_{n}\underbrace{(|0\rangle \otimes |0\rangle \otimes \cdots \otimes |0\rangle)}_{n} = \underbrace{(H|0\rangle) \otimes (H|0\rangle) \otimes \cdots \otimes (H|0\rangle)}_{n}$$

$$= \left(\frac{1}{\sqrt{2}}\right)^n \underbrace{(|0\rangle + |1\rangle) \otimes (|0\rangle + |1\rangle) \otimes \cdots \otimes (|0\rangle + |1\rangle)}_{n}$$

$$\{0,1\}^n = \underbrace{\{0,1\} \times \{0,1\} \times \cdots \times \{0,1\}}_{n}$$

$$= \left(\frac{1}{\sqrt{2}}\right)^n \underbrace{|00\cdots00\rangle \otimes |00\cdots01\rangle \otimes |00\cdots10\rangle \otimes \cdots \otimes |11\cdots10\rangle |11\cdots11\rangle}_{n} = \left(\frac{1}{\sqrt{2}}\right)^n \sum_{x \in \{0,1\}^n} |x\rangle$$

The Deutsch–Jozsa Algorithm

- This is a very common and useful way of writing this state; the *n*-qubit Hadamard gate acting on the *n*-qubit state of all zeros gives a superposition of all *n*-qubit basis states, all with the same amplitude (called an 'equally weighted superposition') $1/\sqrt{2^n}$

- So, the state immediately after the first $H^{\otimes n}$ in the Deutsch–Jozsa algorithm is

$$|\psi_1\rangle = \left(\frac{1}{\sqrt{2}}\right)^n \sum_{x\in\{0,1\}^n} |x\rangle \left(\frac{|0\rangle - |1\rangle}{\sqrt{2}}\right)$$

The Deutsch–Jozsa Algorithm

- Now consider the state immediately after the U_f gate. The state is

$$\left|\psi_2\right\rangle = \left(\frac{1}{\sqrt{2}}\right)^n U_f\left(\sum_{x\in\{0,1\}^n}|x\rangle\left(\frac{|0\rangle-|1\rangle}{\sqrt{2}}\right)\right) = \left(\frac{1}{\sqrt{2}}\right)^n \sum_{x\in\{0,1\}^n} U_f\left(|x\rangle\left(\frac{|0\rangle-|1\rangle}{\sqrt{2}}\right)\right)$$

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

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

The Deutsch–Jozsa Algorithm

- To facilitate our analysis of the state after the interference is completed by the second Hadamard gate, consider the action of the *n*-qubit Hadamard gate on an *n*-qubit basis state $|x\rangle$

The Deutsch–Jozsa Algorithm

- It is easy to verify that the effect of the **1-qubit Hadamard gate** on a 1-qubit basis state $|x\rangle$ can be written as

$$H|x\rangle = \frac{1}{\sqrt{2}}\left(|0\rangle + (-1)^x |1\rangle\right) = \frac{1}{\sqrt{2}} \sum_{z\in\{0,1\}} (-1)^{xz} |z\rangle$$

- Then we can see that the action of the Hadamard transformation on an *n*-qubit basis state $|x\rangle = |x_1\rangle|x_2\rangle|x_3\rangle\cdots|x_n\rangle$ is given by the action of the *n*-qubit Hadamard gate on an *n*-qubit basis state

The Deutsch–Jozsa Algorithm

$$H^{\otimes n}|x\rangle = \underbrace{(H \otimes H \otimes \cdots \otimes H)}_{n} \underbrace{\left(|x_1\rangle \otimes |x_2\rangle \otimes \cdots \otimes |x_n\rangle\right)}_{n} = \underbrace{\left(H|x_1\rangle\right) \otimes \left(H|x_2\rangle\right) \otimes \cdots \otimes \left(H|x_n\rangle\right)}_{n}$$

$$= \left(\frac{1}{\sqrt{2}}\right)^n \underbrace{\left(|0\rangle + (-1)^{x_1}|1\rangle\right) \otimes \left(|0\rangle + (-1)^{x_2}|1\rangle\right) \otimes \cdots \otimes \left(|0\rangle + (-1)^{x_n}|1\rangle\right)}_{n}$$

$$= \left(\frac{1}{\sqrt{2}}\right)^n \left(\sum_{z_1 \in \{0,1\}} (-1)^{x_1 z_1} |z_1\rangle\right) \left(\sum_{z_2 \in \{0,1\}} (-1)^{x_2 z_2} |z_2\rangle\right) \cdots \left(\sum_{z_n \in \{0,1\}} (-1)^{x_n z_n} |z_n\rangle\right)$$

$$= \left(\frac{1}{\sqrt{2}}\right)^n \sum_{z_1 z_2 \cdots z_n \in \{0,1\}} (-1)^{x_1 z_1 + x_2 z_2 + \cdots + x_n z_n} |z_1\rangle |z_2\rangle \cdots |z_n\rangle$$

The Deutsch–Jozsa Algorithm

- We are now in a position to derive $|\psi_3\rangle$ by applying the separable transformation $H^{\otimes n} \otimes I$ to $|\psi_2\rangle$

- We only care about the Hadamard part, since it is the output of the data register we will measure. It produces the output

$$|\psi_3\rangle = \left(H^{\otimes n} \otimes I\right)|\psi_2\rangle = \left(\frac{1}{\sqrt{2}}\right)^n \sum_{x\in\{0,1\}^n} (-1)^{f(x)} H^{\otimes n}|x\rangle \otimes I\left(\frac{|0\rangle - |1\rangle}{\sqrt{2}}\right)$$

$$= \underbrace{\left(\frac{1}{\sqrt{2}}\right)^n \sum_{x\in\{0,1\}^n} (-1)^{f(x)} H^{\otimes n}|x\rangle}_{\textit{We are interested in this part}} \otimes \left(\frac{|0\rangle - |1\rangle}{\sqrt{2}}\right)$$

The Deutsch–Jozsa Algorithm

- By exploiting the previous result

$$H^{\otimes n}|x\rangle = \left(\frac{1}{\sqrt{2}}\right)^n \sum_{z_1 z_2 \cdots \in \{0,1\}} (-1)^{x_1 z_1 + x_2 z_2 + \cdots + x_n z_n} |z_1\rangle |z_2\rangle \cdots |z_n\rangle$$

$$|\psi_3\rangle = \left(\frac{1}{\sqrt{2}}\right)^n \sum_{x \in \{0,1\}^n} (-1)^{f(x)} H^{\otimes n}|x\rangle \otimes \left(\frac{|0\rangle - |1\rangle}{\sqrt{2}}\right)$$

$$|\psi_3\rangle = \left[\left(\frac{1}{\sqrt{2}}\right)^n \sum_{x \in \{0,1\}^n} (-1)^{f(x)} \left(\frac{1}{\sqrt{2}}\right)^n \sum_{z_1 z_2 \cdots \in \{0,1\}} (-1)^{x_1 z_1 + x_2 z_2 + \cdots + x_n z_n} |z_1\rangle |z_2\rangle \cdots |z_n\rangle\right] \otimes \left(\frac{|0\rangle - |1\rangle}{\sqrt{2}}\right)$$

The Deutsch–Jozsa Algorithm

Therefore, the state after the final n-qubit Hadamard gate in the Deutsch-Jozsa algorithm is

$$\left|\psi_3\right\rangle = \left(\frac{1}{2}\right)^n \sum_{z_1 z_2 \cdots z_n \in \{0,1\}} \underbrace{\left(\sum_{x_1 x_2 \cdots x_n \in \{0,1\} \in \{0,1\}} (-1)^{f\left(x_1, x_2, \cdots, x_n\right) + x_1 z_1 + x_2 z_2 + \cdots + x_n z_n} \right)}_{G\left(z_1, z_2, \cdots, z_n\right)} \left|z_1\right\rangle \left|z_2\right\rangle \cdots \left|z_n\right\rangle \otimes \left(\frac{\left|0\right\rangle - \left|1\right\rangle}{\sqrt{2}} \right)$$

where we have regrouped the sum and defined a scalar function $G\left(z_1, z_2, \cdots, z_n\right)$ of the summation indices $z_1, z_2, \cdots, z_n$

The Deutsch–Jozsa Algorithm

So, the final output is an expansion along the z-basis,

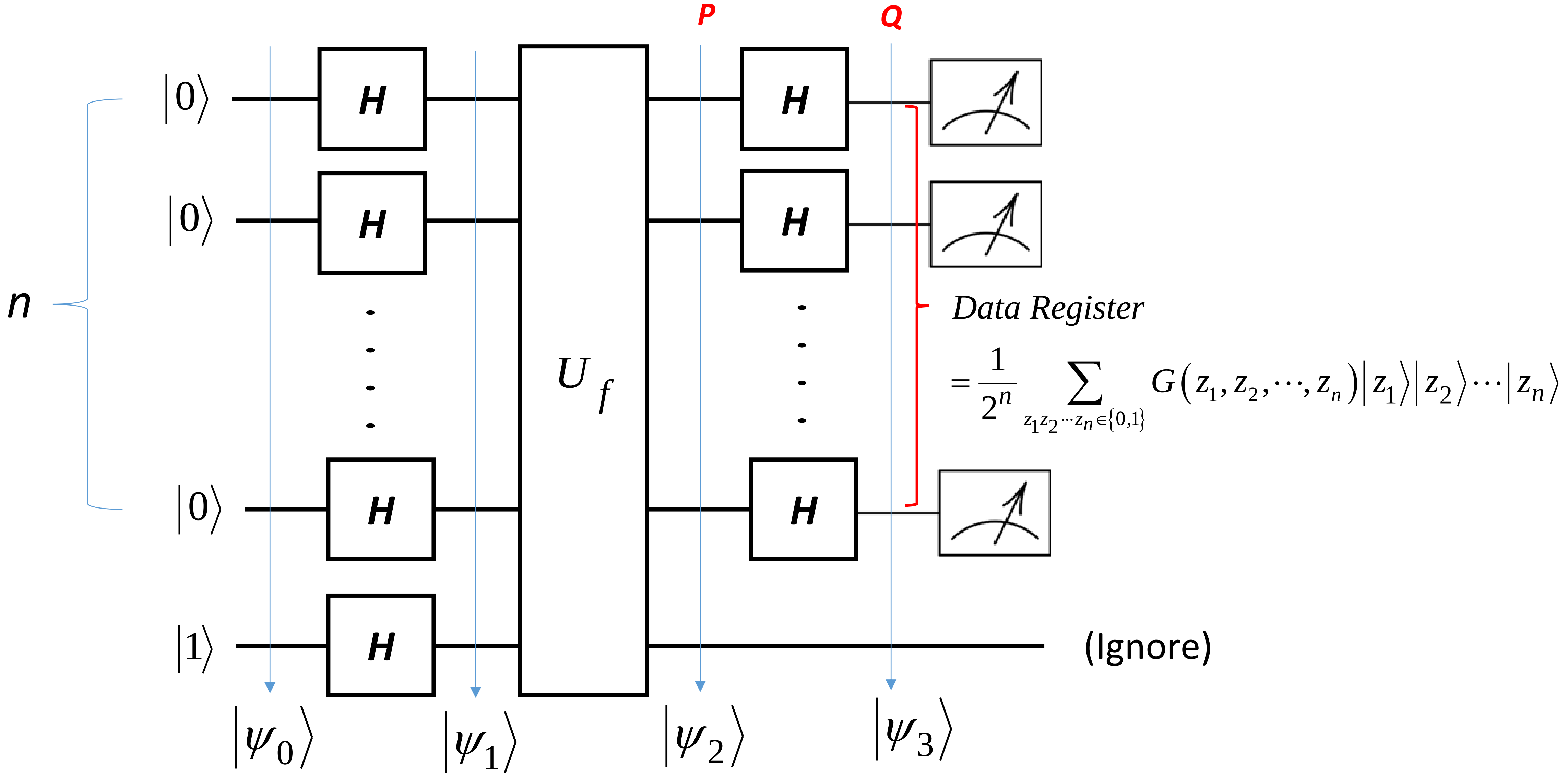
$$|\psi_3\rangle = \left(\frac{1}{2}\right)^n \sum_{z_1 z_2 \cdots z_n \in \{0,1\}} G(z_1, z_2, \cdots, z_n)|z_1\rangle|z_2\rangle\cdots|z_n\rangle \otimes \left(\frac{|0\rangle - |1\rangle}{\sqrt{2}}\right)$$

Therefore

Data register at access point Q $= \dfrac{1}{2^n} \displaystyle\sum_{z_1 z_2 \cdots z_n \in \{0,1\}} G(z_1, z_2, \cdots, z_n)|z_1\rangle|z_2\rangle\cdots|z_n\rangle$

At the end of the algorithm a measurement of the data register is made in the computational basis (just as was done for the Deutsch algorithm)

The Deutsch–Jozsa Algorithm



The Deutsch–Jozsa Algorithm

- To see what happens, consider the total amplitude (coefficient) *G*(0) of

$$|z\rangle = |0\rangle|0\rangle\cdots|0\rangle \equiv |0\rangle^{\otimes n}$$

- This will tell us something about the other $2^n - 1$ CBS amplitudes, $G(z_1, z_2, \cdots, z_n)$; for $z_i > 0, \forall i \in \{1, 2, \ldots, n\}$. This amplitude is

$$\frac{G(0,0,\ldots\ldots,0)}{2^n} = \frac{1}{2^n} \sum_{x_1 x_2 \cdots x_n \in \{0,1\}} (-1)^{f(x_1, x_2, \cdots, x_n)}$$

- Consider this amplitude in the two cases: *f* constant and *f* balanced

The Deutsch–Jozsa Algorithm

- If *f* is constant, the amplitude of $|0\rangle^{\otimes n}$ is either +1 or −1 (depending on what value *f*(*x*) takes, i.e. 0 or 1)

$$\frac{G(0,0,\ldots\ldots,0)}{2^n} = \frac{1}{2^n}\sum_{x_1x_2\cdots x_n\in\{0,1\}}(-1)^0 = \frac{1}{2^n}\sum_{x_1x_2\cdots x_n\in\{0,1\}}1 = \frac{2^n}{2^n} = 1 \quad for \quad f(x_1,x_2,\cdots,x_n)=0$$

$$\frac{G(0,0,\ldots\ldots,0)}{2^n} = \frac{1}{2^n}\sum_{x_1x_2\cdots x_n\in\{0,1\}}(-1)^1 = \frac{1}{2^n}\sum_{x_1x_2\cdots x_n\in\{0,1\}}(-1) = -\frac{2^n}{2^n} = -1 \quad for \quad f(x_1,x_2,\cdots,x_n)=1$$

thereby forcing the **amplitudes** of all other *z*-basis kets in the expansion to be 0

The Deutsch–Jozsa Algorithm

- Thus, in *the constant* case

  Data register at access point Q $= \pm|0\rangle|0\rangle\cdots|0\rangle$

- It turns out that if f is constant, a measurement of the data register is *certain to return all 0s* (by ‘all 0s’ we mean the binary string $00 \cdots 0$)

The Deutsch–Jozsa Algorithm

If *f* is balanced, then the amplitude of $|0\rangle^{\otimes n}$ in the expansion is

$$\frac{G(0,0,\ldots\ldots,0)}{2^n} = \frac{1}{2^n} \sum_{x_1 x_2 \cdots x_n \in \{0,1\} \in \{0,1\}} (-1)^{f(x_1, x_2, \cdots, x_n)}$$

but a balanced *f* promises an equal number $f(x_1, x_2, \cdots, x_n) = 0$ and $f(x_1, x_2, \cdots, x_n) = 1$, so the sum has an equal number of +1s and -1s, forcing it to be 0

The Deutsch–Jozsa Algorithm

- Therefore the probability of a measurement causing a collapse to the state $|0\rangle^{\otimes n}$ is 0 (the amplitude-squared of the CBS state $|0\rangle^{\otimes n}$)

- We are guaranteed, for a balanced function, to never get a reading of "0" when we measure the data register at access point Q

The Deutsch–Jozsa Algorithm in Summary

- We've explained the purpose of all the components in the circuit and how each plays a role in leveraging *quantum parallelism*, *interference* and *phase kick-back*

- We run the circuit **one time only** and measure the data register output in the standard basis

- If we read "0" then ***f* is constant**

- Otherwise, ***f* is balanced**