

8-lecture Mini-course in Quantum Computation

Lecture 1

Lawrence Ioannou

	Topics Covered	References ^{^ % *}
1	Basics I: motivation, interference, qubits, measurement	Mosca: 1.1, 1.2 Textbook: 1.1.1 – 1.3.4
2	Basics II: gates, circuits	Textbook: 4.1-4.4
3	Cool Applications: Bell states, superdense coding, no-cloning theorem, teleportation	Textbook: 1.3.5 - 1.3.7, 2.3
4	Algorithms I: models of computation, universality, relative phases, eigenvalue kick-back, Deutsch's problem	Mosca: 2.1 Textbook: 3.1, 4.5, 4.6, 4.7, 5.1, 5.2
5	Algorithms II: phase estimation, quantum Fourier transform (QFT), eigenvalue estimation, period-finding, order-finding	Mosca: 2.2, 2.3, 2.4, 2.5, 2.8.1, 2.9 Textbook: 5.1, 5.2, 5.4.1, 5.3
6	Cryptographic applications: factoring $N=pq$, breaking the RSA cryptosystem, discrete logarithm problem (El Gamal cryptosystem), quantum key distribution	Mosca: A.5, 2.6 Textbook: A4.3, 5.3, 5.4.2, 12.6.1, 12.6.3
7	Algorithms III: quantum searching, counting	Mosca: 2.7 Textbook: 6
8	Algorithms II*: review, Simon's problem, hidden subgroup problem	quant-ph/9704027 Mosca: 2.8 Textbook: 5.4.3

% Mosca = Michele Mosca's DPhil thesis, available at
<http://www.cacr.math.uwaterloo.ca/~mmosca/moscathesis.ps>

* quant-ph/xxxxxx = www.arxiv.org/quant-ph/xxxxxx

^ Textbook = M. A. Nielsen and I. L. Chuang, "Quantum Computation and Information", Cambridge University Press, 2000.

Motivation

Computation is a physical process

e.g. neurons firing in your brain

e.g. electrons flowing through NAND gates in a computer's circuits

Therefore, laws of physics govern computation

"Classical physics" includes

Newtonian mechanics (e.g. $F=ma$)

Einstein's General Relativity

classical electrodynamics (Maxwell's equations)

c. 1905 experiments showed some of these laws to break down for small-scale systems (e.g. individual photons and electrons)

Motivation

“Quantum physics” replaced “classical physics” as the more correct theory describing the laws of nature

The Turing Machine (i.e. standard computer) can be implemented by physical phenomena that can be described by the laws of “classical physics”

Feynman (1982): But what if we build computers that use phenomena that can only be described by the laws of “quantum physics”?

Quantum computation was born...



Single-photon interferometer

Two paths through space, labelled " $|0\rangle$ " and " $|1\rangle$ "

$|0\rangle$ -----

$|1\rangle$ -----

Single-photon source, placed at start of $|0\rangle$ path

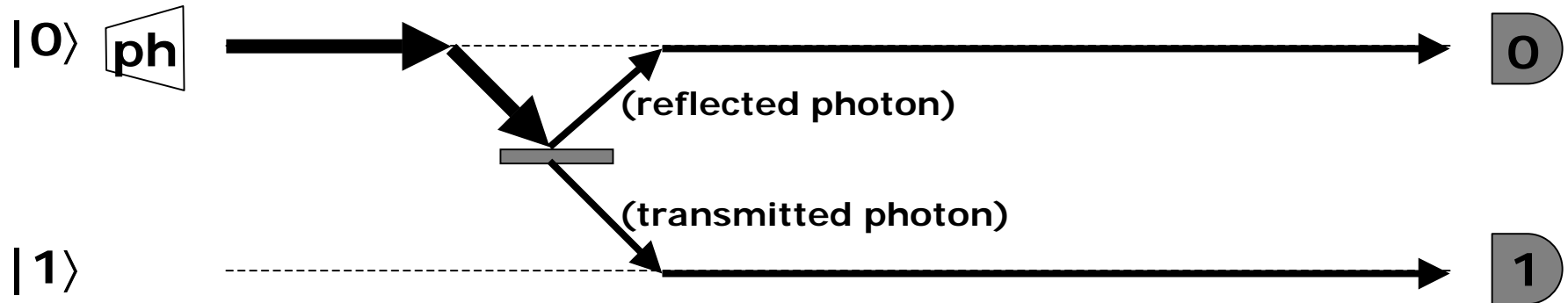
$|0\rangle$   -----

(thick arrow represents trace of path taken by a photon)

$|1\rangle$ -----

Single-photon interferometer (2)

Add in some full-deflection mirrors (not shown), a *beam-splitter** (—), and some (*photon*) detectors (◐)



A detector *clicks* when a photon hits it (no "half-clicks" occur)

Experiment: Have the photon-source emit one photon, and then record which detector clicks; repeat many times

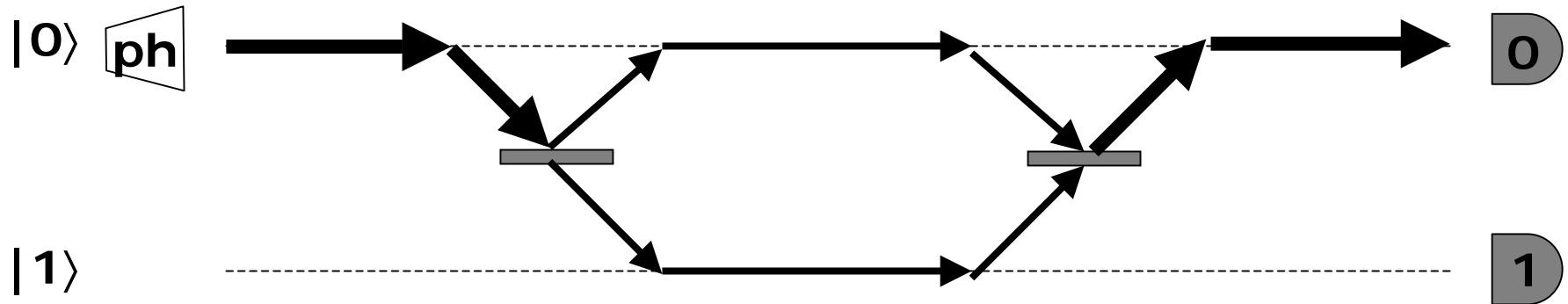
Results: ~50% 0-clicks, ~50% 1-clicks

Reasonable conclusion: beam-splitter deflects photon with probability $\frac{1}{2}$, and transmits photon with probability $\frac{1}{2}$

*we will actually use Hadamard gates (introduced later), which are slightly different from beam-splitters; a beam-splitter can also be referred to as a "half-silvered mirror"

Single-photon interferometer (3)

Now add in some more full-deflection mirrors (not shown) and another beam-splitter



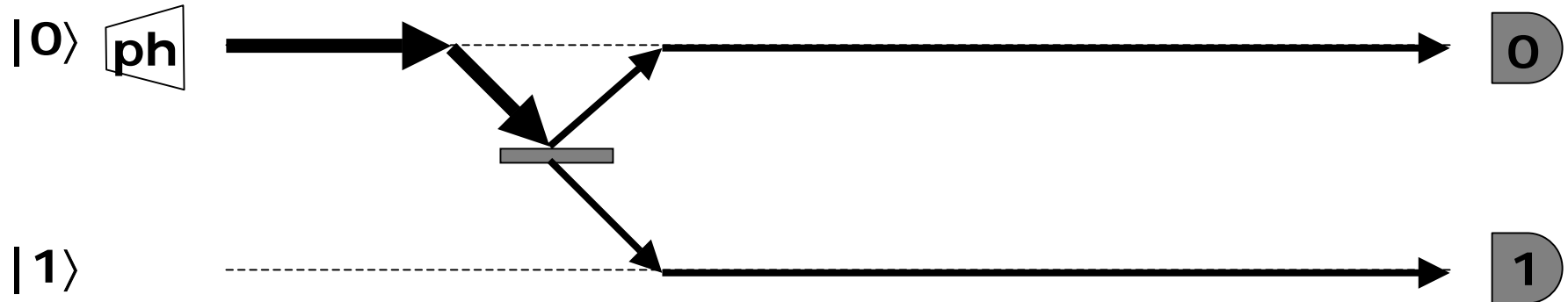
Repeat previous experiment

Surprising results: $\sim 100\%$ 0-clicks, $\sim 0\%$ 1-clicks

The previous conclusion is not complete, as we would have expected the same even distribution...

Single-photon interferometer (4)

Quantum-mechanical explanation of first experiment



The state of the photon can exist in a *superposition* of the two paths

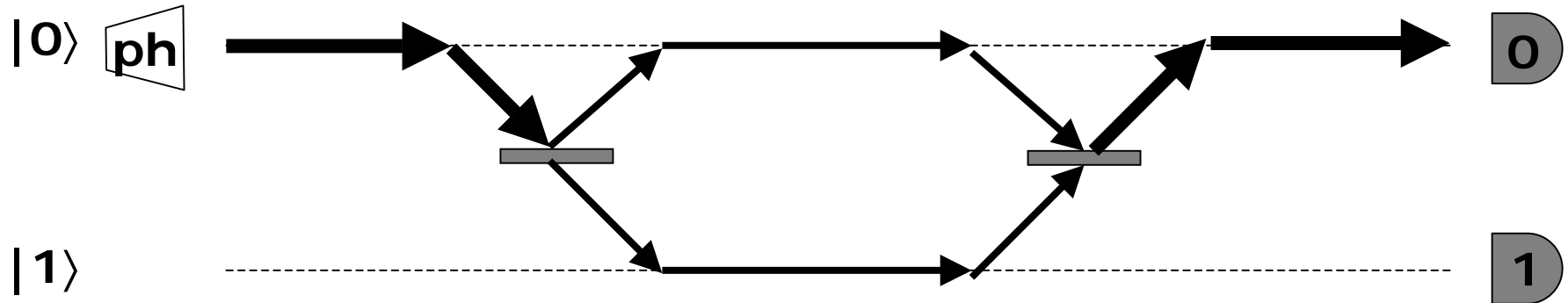
When a photon in the $|0\rangle$ path impinges on the beam-splitter, it exits in the state

$$\frac{1}{\sqrt{2}} |0\rangle + \frac{1}{\sqrt{2}} |1\rangle$$

The probabilities of detecting the photon in the respective paths are obtained by squaring the moduli of the coefficients, which are complex numbers in general

Single-photon interferometer (5)

Quantum-mechanical explanation of second experiment



When a photon in the $|1\rangle$ path impinges on the beam-splitter, it exits in the state

$$\frac{1}{\sqrt{2}}|0\rangle - \frac{1}{\sqrt{2}}|1\rangle$$

We say that the reflected beam picks up a *phase* of -1

The beam-splitter acts independently on each component in the superposition

Thus, the state of the photon after the second beam-splitter is

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

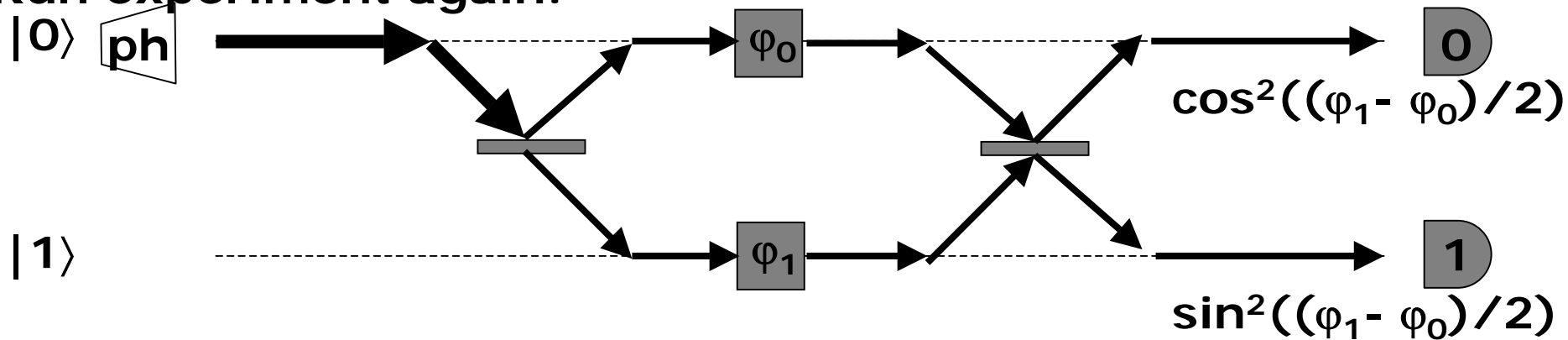
Single-photon interferometer (6)

Inspired by the previous experiment, now add in two *phase-shifters* (φ), which may be pieces of glass of varying thickness characterised by real number φ

For $b \in \{0, 1\}$, when a photon in the $|b\rangle$ path passes through a φ -phase-shifter, it exits as

$$e^{i\varphi} |b\rangle$$

Run experiment again:



Exercise: verify the two probabilities shown in the diagram

Generally, by changing the *relative phase* $(\varphi_1 - \varphi_0)$ between the two paths, we can modify the statistics of the experiment

One Qubit

A (classical) bit is a two-level physical system
The two levels are usually labelled "0" and "1"
Let b be the state of the bit

"Classical physics" dictates that the bit may exist only in either the state 0 or 1:

$$b=0 \text{ or } b=1$$

A quantum bit (qubit) is also a two-level physical system
The two levels are usually labelled " $|0\rangle$ " and " $|1\rangle$ "
Let $|\psi\rangle$ be the (*pure*) *state* of the qubit

"Quantum physics" dictates that the qubit may exist in a *complex superposition* of $|0\rangle$ and $|1\rangle$:

$$|\psi\rangle = \alpha|0\rangle + \beta|1\rangle \quad (\text{superposition principle})$$

where α and β are complex numbers such that $|\alpha|^2 + |\beta|^2 = 1$

One Qubit

A qubit is thus modelled by a two-dimensional complex vector space, \mathbb{C}^2 , with (standard) orthonormal basis $\{|0\rangle, |1\rangle\}$, called the *computational basis*

The state $|\psi\rangle$ of a qubit is modelled by a unit vector in this vector space

Physically, a qubit can be realised by

- the two degrees of freedom of two paths of a single photon (as we saw earlier)
- the two degrees of freedom of a spin-(1/2) particle: spin-up and spin-down)
- many other ways...

Don't Freak Out: vector notation

In quantum mechanics, we use *Dirac bra-ket notation* to denote vectors

	Standard notation	Dirac notation
vector (represented as column-vector)	\vec{v} or v	$ v\rangle$ (ket)
dual vector (represented as row-vector)	v^\dagger	$\langle v $ (bra)
inner product	$\vec{u} \bullet \vec{v}$ or $u^\dagger v$	$\langle u v\rangle$
standard basis of \mathbb{C}^N $N = 2^n$	(elementary basis) $\{e_1, e_2, \dots, e_N\}$ $e_j = \begin{bmatrix} 0 \\ \vdots \\ 0 \\ 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix}$	(computational basis) $\{ 0\rangle, 1\rangle, \dots, N-1\rangle\}$ or $\{ (0)_2\rangle, (1)_2\rangle, \dots, (N-1)_2\rangle\}$ where $(j)_2$ denotes the binary representation of j ; i.e. labels are n -bit strings; when $N=2^n$, each bit corresponds to a qubit

Measurement (2)

Thus, like classical bits, the result of measuring a qubit is binary

Though, unlike classical bits, the result of measuring a qubit is *not deterministic* and *disturbs* its state (e.g. leaves it in $|0\rangle$ or $|1\rangle$ if measuring w.r.t. computational basis)

$$|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$$

The numbers α and β are called (*probability*) *amplitudes*

Qubits

A two-qubit system is a system with 4 levels which correspond to the 4 combinations of the 2 levels in each qubit:

$$|0\rangle|0\rangle, |0\rangle|1\rangle, |1\rangle|0\rangle, |1\rangle|1\rangle$$

Again, the principle of quantum superposition says that the state $|\varphi\rangle$ of the two-qubit system may be

$$|\varphi\rangle = \alpha_0|0\rangle|0\rangle + \alpha_1|0\rangle|1\rangle + \alpha_2|1\rangle|0\rangle + \alpha_3|1\rangle|1\rangle$$

where $|\alpha_0|^2 + |\alpha_1|^2 + |\alpha_2|^2 + |\alpha_3|^2 = 1$ (α_i are complex numbers)

Similar to before, measuring the two-qubit system with respect to the computational basis gives outcome

00 with probability $|\alpha_0|^2$

01 with probability $|\alpha_1|^2$

10 with probability $|\alpha_2|^2$

11 with probability $|\alpha_3|^2$

Qubits (2)

Mathematically, a two-qubit system is modelled by the *tensor product* (\otimes) of two copies of the vector space that models a one-qubit system

E.g. let $|\psi\rangle_A = \alpha|0\rangle_A + \beta|1\rangle_A$ and $|\psi\rangle_B = \gamma|0\rangle_B + \delta|1\rangle_B$ be the states of two qubits; then the state $|\psi\rangle_{AB}$ of the joint two-qubit system (in the order A,B) is

$$\begin{aligned} |\psi\rangle_{AB} &= (\alpha|0\rangle_A + \beta|1\rangle_A) \otimes (\gamma|0\rangle_B + \delta|1\rangle_B) \\ &= \alpha\gamma|0\rangle_A \otimes |0\rangle_B + \alpha\delta|0\rangle_A \otimes |1\rangle_B + \beta\gamma|1\rangle_A \otimes |0\rangle_B + \beta\delta|1\rangle_A \otimes |1\rangle_B \end{aligned}$$

which is usually just written

$$\alpha\gamma|00\rangle + \alpha\delta|01\rangle + \beta\gamma|10\rangle + \beta\delta|11\rangle$$

and the order of systems A and B remembered

Note that the tensor product makes sense in light of the measurement principle and probability theory: e.g. the probability of getting outcome $|01\rangle$ is the probability of getting $|0\rangle$ for qubit A *and* $|1\rangle$ for qubit B (assuming independence of events)

Qubits (3)

In general, an n-qubit system (*register*) is modelled by the *tensor product* (\otimes) of n copies of \mathbb{C}^2

The state $|\psi\rangle$ of an n-qubit register is a complex unit vector in this tensor product space, often written in the computational basis as

$$|\psi\rangle = \sum_{i=0}^{2^n-1} \alpha_i |i\rangle$$

where the index is viewed in its n-bit binary representation when inside the " $| \rangle$ ", and

$$\sum_{i=0}^{2^n-1} |\alpha_i|^2 = 1$$

Measuring the n-qubit register with respect to the computational basis gives outcome

$$|i\rangle \text{ with probability } |\alpha_i|^2$$

More Measurement

More generally, one can *measure with respect to (orthonormal) basis* $B = \{ |b_i\rangle \}$

If B is a basis for the vector space containing $|\psi\rangle$, then the state (expressed in the computational basis)

$$|\psi\rangle = \sum_{i=0}^{2^n-1} \alpha_i |i\rangle$$

may also be written

$$|\psi\rangle = \sum_{i=0}^{2^n-1} \beta_i |b_i\rangle$$

Measurement of the qubit system with respect to basis B gives outcome

$$b_i \text{ with probability } |\beta_i|^2$$

and, immediately after the measurement, the state of the qubit system *is* the basis element labelled by the outcome

More Measurement (2)

E.g. for two-qubit systems, an interesting basis other than the computational (standard) basis is the *Bell basis*, consisting of the four *Bell states*:

$$|00\rangle + |11\rangle$$

$$|00\rangle - |11\rangle$$

$$|01\rangle + |10\rangle$$

$$|01\rangle - |10\rangle$$

where we have omitted the normalisation factor of $\frac{1}{\sqrt{2}}$ in each state.

We'll return to these states later... they are very important for communication protocols and quantum cryptography!

8-lecture Mini-course in Quantum Computation

Lecture 2

Lawrence Ioannou

Quantum Gates

Recall the classical gates (for bits b, c, d):



What should a *quantum gate* be like?

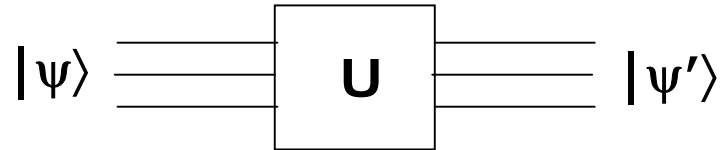


Since $|\psi\rangle$ and $|\psi'\rangle$ are quantum states, the gate should preserve vector norm. Notice that a *unitary operator* has this property....

(Recall: U is unitary means $U^{-1}=U^\dagger$, where \dagger denotes *complex-conjugate transpose* operation, i.e. complex conjugate of each matrix element followed by matrix transpose operation)

Quantum Gates

In general, quantum gates are *unitary operators*



Note that the number of input qubits always equals the number of output qubits

Since all unitary operators have an inverse, all quantum gates are *reversible*



which means no heat need be dissipated during a computation (Landauer's principle)

(Classical computing can also be made reversible)

Matrix Representations

When doing calculations, often it is useful to represent the state of a qubit-register and the operations (gates) applied to it by *matrices with respect to the computational basis*

Technically, we can use the notation “[...]” to distinguish the state or operator from its matrix representation:

e.g. $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$ is represented as $[|\psi\rangle] = \begin{bmatrix} \alpha \\ \beta \end{bmatrix}$

e.g. the one-qubit gate U which maps

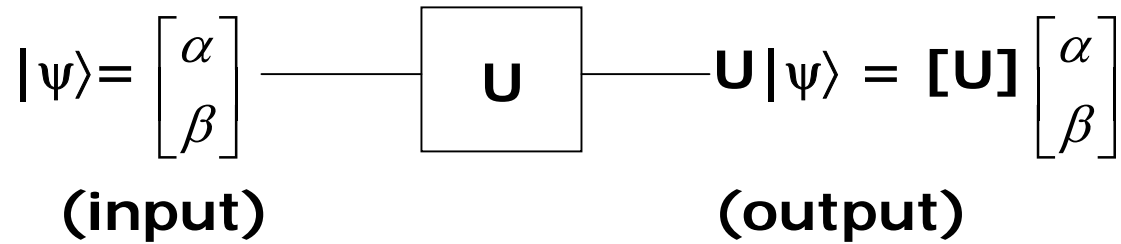
$$\begin{aligned} |0\rangle &\rightarrow a|0\rangle + b|1\rangle \\ |1\rangle &\rightarrow c|0\rangle + d|1\rangle \end{aligned}$$

is represented as $[U] = \begin{bmatrix} a & c \\ b & d \end{bmatrix}$

When the basis is understood to be the computational basis, we may drop the “[...]” notation (especially for states)

e.g. we may write $U|\psi\rangle = [U][|\psi\rangle]$

Some Important One-qubit Gates

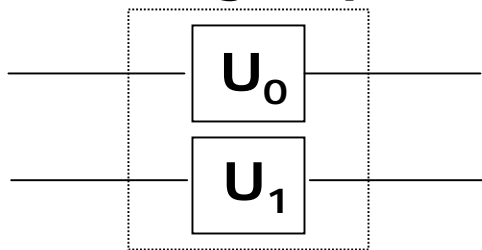


\mathbf{U}	$[\mathbf{U}]$	Gate symbol
Hadamard	$\frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$	
θ -phase	$\begin{bmatrix} 1 & 0 \\ 0 & e^{i\theta} \end{bmatrix}$	
Pauli-X (NOT)	$\begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$	
Pauli-Y	$\begin{bmatrix} 0 & -i \\ i & 0 \end{bmatrix}$	
Pauli-Z	$\begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}$	

Two-Qubit Gates

$$|\psi\rangle = \begin{bmatrix} \alpha_0 \\ \alpha_1 \\ \alpha_2 \\ \alpha_3 \end{bmatrix} \quad \begin{array}{c} \text{---} \\ \text{---} \end{array} \begin{array}{c} \boxed{\text{U}} \\ \boxed{\text{U}} \end{array} \begin{array}{c} \text{---} \\ \text{---} \end{array} \quad \text{U}|\psi\rangle = [\text{U}] \begin{bmatrix} \alpha_0 \\ \alpha_1 \\ \alpha_2 \\ \alpha_3 \end{bmatrix}$$

Two one-qubit gates acting in parallel constitute a two-qubit gate:



If U is the operator corresponding to this 2-qubit gate, how is U represented mathematically in terms of U_0 and U_1 ?

ANSWER: tensor product!

$$U = U_0 \otimes U_1$$

Matrix tensor product: the tensor product $M \otimes N$ of two matrices M and N is the block matrix:

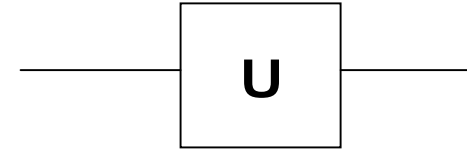
$$[M_{ij}N]$$

where M_{ij} is the element of M in the i th row and j th column

Controlled-Operations (two-qubit gates)

But, in general, a two-qubit gate is not the tensor product of two one-qubit gates!

Suppose U is a one-qubit operation



We can define the operation on qubits A and B that maps

$$|0\rangle_A |\psi\rangle_B \rightarrow |0\rangle_A |\psi\rangle_B$$

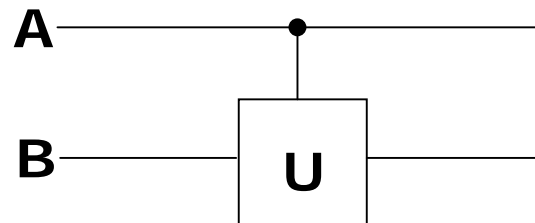
$$|1\rangle_A |\psi\rangle_B \rightarrow |1\rangle_A U|\psi\rangle_B$$

Classically: "If qubit A is in state $|1\rangle$, then apply U to qubit B; otherwise do nothing"

This is the *controlled-U* gate, with *control qubit A* and *target qubit B*.

For general U , controlled- U cannot be written as a tensor product of two one-qubit gates

Gate symbol:



Two Important Two-qubit Gates

$$|\psi\rangle = \begin{bmatrix} \alpha_0 \\ \alpha_1 \\ \alpha_2 \\ \alpha_3 \end{bmatrix} \quad \begin{array}{c} \text{---} \\ \text{---} \end{array} \begin{array}{c} \boxed{\mathbf{U}} \\ \boxed{\mathbf{U}} \end{array} \begin{array}{c} \text{---} \\ \text{---} \end{array} \quad \mathbf{U}|\psi\rangle = [\mathbf{U}] \begin{bmatrix} \alpha_0 \\ \alpha_1 \\ \alpha_2 \\ \alpha_3 \end{bmatrix}$$

All matrices are with respect to the computational basis with the following qubit ordering

$$\{ |0\rangle_A |0\rangle_B, |0\rangle_A |1\rangle_B, |1\rangle_A |0\rangle_B, |1\rangle_A |1\rangle_B \}$$

\mathbf{U}	$[\mathbf{U}]$	Gate symbol
Controlled-NOT ($\text{CNOT}_{A,B}$)	$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix}$	<p>qubit A (control)</p> <p>qubit B (target)</p>
SWAP _{A,B} (maps $ \psi\rangle_A \phi\rangle_B \rightarrow \phi\rangle_A \psi\rangle_B$)	$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$	

Quantum Networks

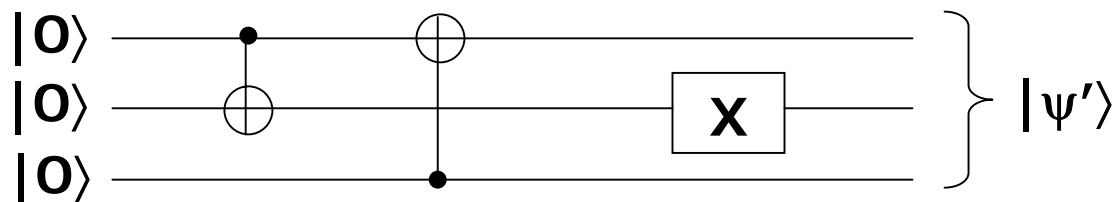
A *quantum network* (diagram) is a tool we use to describe a (complex) quantum operation usually as a sequence of (simpler) operations

It is a concatenation of gate symbols, read from left to right

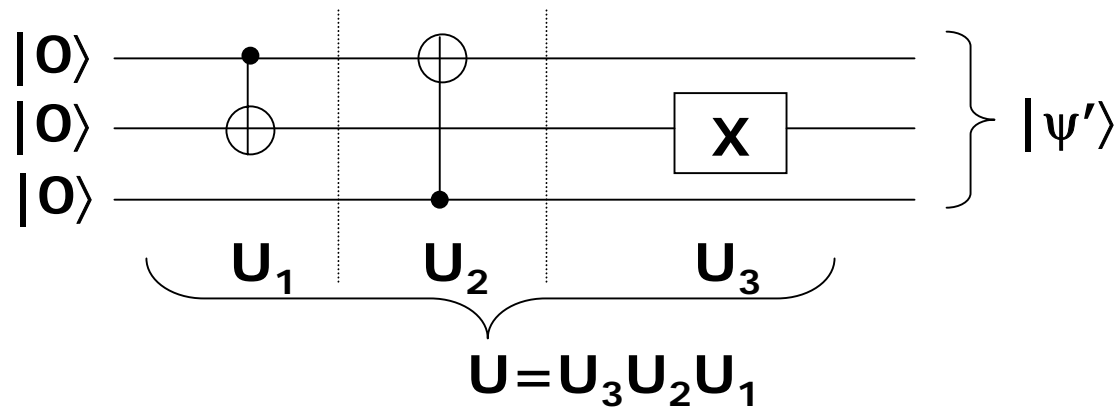
Each horizontal line represents one qubit

To represent a particular computation, the initial (input) state (of each qubit) may be written on the left of the diagram; the final (output) state may be written on the right.

E.g. (a three-qubit quantum network)



Computing U and $|\psi'\rangle$



Label the qubits from top to bottom: A, B, C

Fix a computational basis with the corresponding ordering:

$$\{ |0\rangle_A |0\rangle_B |0\rangle_C, |0\rangle_A |0\rangle_B |1\rangle_C, \dots, |1\rangle_A |1\rangle_B |1\rangle_C \}$$

Reading from left to right, U is made up of three operations U_1 ($\text{CNOT}_{A,B}$), U_2 ($\text{CNOT}_{C,A}$), and U_3 (NOT_B):

$$|\psi'\rangle = U_3 U_2 U_1 |000\rangle$$

Let's calculate the matrix representations of each operation...

Computing U and $|\psi'\rangle$

Computing $[U_2]$ ($\text{CNOT}_{C,A}$)

Either directly from mapping of computational basis:

$$|x\rangle_A |x\rangle_B |0\rangle_C \rightarrow |x\rangle_A |x\rangle_B |0\rangle_C$$

$$|x\rangle_A |x\rangle_B |1\rangle_C \rightarrow |1-x\rangle_A |x\rangle_B |1\rangle_C \quad x \in \{0,1\}$$

or, notice that

$$\text{CNOT}_{C,A} = (\text{SWAP}_{B,C}) (\text{CNOT}_{B,A}) (\text{SWAP}_{B,C})$$

$[U_2] =$

$$\left(\begin{array}{c} \left[\begin{array}{cc} 1 & 0 \\ 0 & 1 \end{array} \right] \otimes \left[\begin{array}{cc} 1 & 0 \\ 0 & 1 \end{array} \right] \\ \left[\begin{array}{ccc} 1 & & \\ & 0 & 1 \\ & 1 & 0 \\ & & & 1 \end{array} \right] \end{array} \right) \left(\begin{array}{c} \left[\begin{array}{ccc} 1 & & \\ & 1 & \\ & & 1 \end{array} \right] \otimes \left[\begin{array}{cc} 1 & 0 \\ 0 & 1 \end{array} \right] \\ \left[\begin{array}{ccc} 1 & & \\ & 0 & 1 \\ & 1 & 0 \\ & & & 1 \end{array} \right] \end{array} \right) \left(\begin{array}{c} \left[\begin{array}{cc} 1 & 0 \\ 0 & 1 \end{array} \right] \otimes \left[\begin{array}{cc} 1 & 0 \\ 0 & 1 \end{array} \right] \\ \left[\begin{array}{ccc} 1 & & \\ & 0 & 1 \\ & 1 & 0 \\ & & & 1 \end{array} \right] \end{array} \right)$$

(zeros elsewhere)

Computing U and $|\psi'\rangle$

Computing $[U_3]$ (NOT_B)

$$U_3 = I \otimes X \otimes I$$

$$[U_3] = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \otimes \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \otimes \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

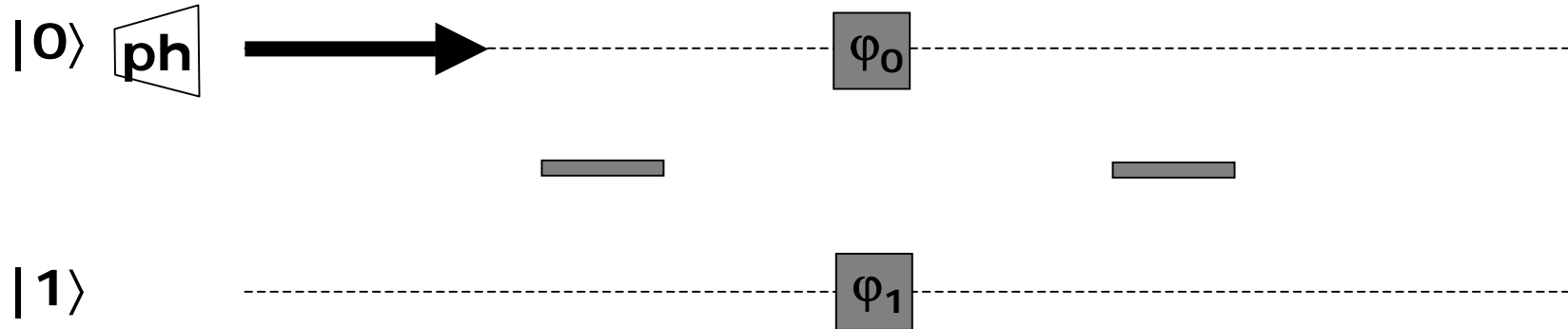
Thus, $[U] = [U_3][U_2][U_1]$

To compute $[|\psi'\rangle]$: $[|\psi'\rangle] = [U][|000\rangle]$

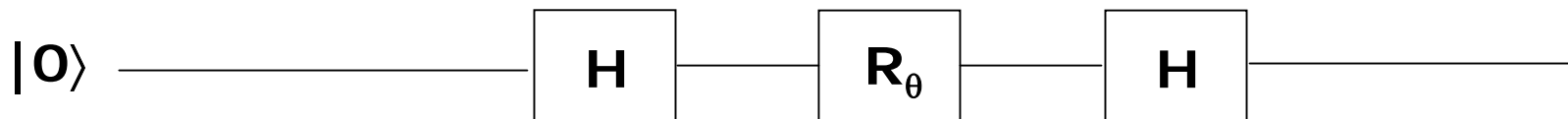
$$= [U_3][U_2][U_1] \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

Single-photon interferometer as a quantum network diagram

The interferometry experiment



is represented by



$$\theta = \varphi_1 - \varphi_0$$

Properties of Tensor Product \otimes

$$\mathbf{A \otimes B \neq B \otimes A}$$

$$\mathbf{A \otimes B \otimes C = (A \otimes B) \otimes C = A \otimes (B \otimes C)}$$

$$\mathbf{A \otimes (B + C) = A \otimes B + A \otimes C}$$

$$\mathbf{(A + B) \otimes C = A \otimes C + B \otimes C}$$

$$\text{for scalar } \alpha: \quad \mathbf{\alpha(A \otimes B) = (\alpha A) \otimes B = A \otimes (\alpha B)}$$

$$\text{for operators } U, V: \quad \mathbf{(U \otimes V)(|\psi\rangle \otimes |\phi\rangle) = (U|\psi\rangle) \otimes (V|\phi\rangle)}$$

$$\text{for matrices } M, N: \quad \mathbf{M \otimes N = [M_{ij}N]} \quad (\text{in block form})$$

where M_{ij} is the element of M in the i th row and j th column

we normally omit the symbol \otimes when it is between states

8-lecture Mini-course in Quantum Computation

Lecture 3

Lawrence Ioannou

Entangled States

Let $|\psi\rangle_A = \alpha|0\rangle_A + \beta|1\rangle_A$ and $|\psi\rangle_B = \gamma|0\rangle_B + \delta|1\rangle_B$ be the states of two qubits; then the state $|\psi\rangle_{AB}$ of the joint two-qubit system is

$$|\psi\rangle_{AB} = (\alpha|0\rangle_A + \beta|1\rangle_A) \otimes (\gamma|0\rangle_B + \delta|1\rangle_B)$$

The state $|\psi\rangle_{AB}$ is a special kind of two-qubit state called a *separable state*, because it is the tensor-product of the states of the two constituent qubits

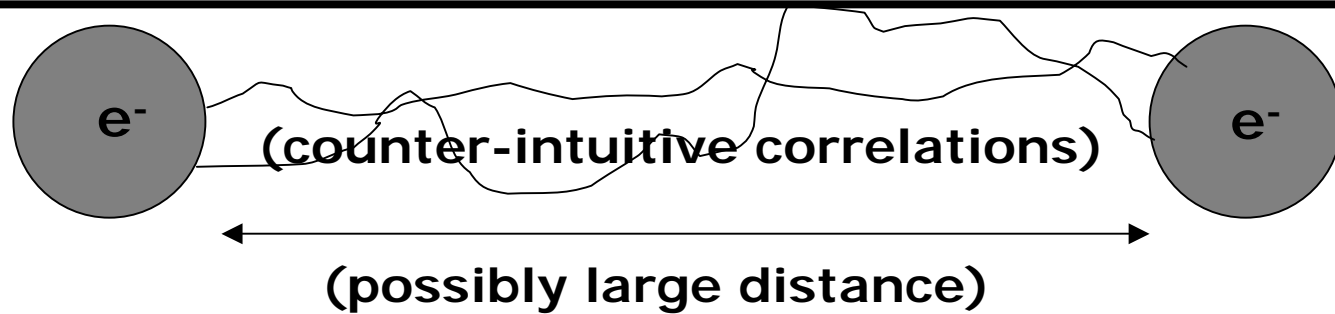
A two-qubit state which is not separable is called an entangled state

e.g. $|00\rangle + |11\rangle$ (omitting normalisation factors)

Exercise: Prove that $|00\rangle + |11\rangle$ is entangled, i.e. show that there do *not* exist $\alpha, \beta, \gamma, \delta$ such that $|00\rangle + |11\rangle$ equals

$$(\alpha|0\rangle + \beta|1\rangle) \otimes (\gamma|0\rangle + \delta|1\rangle)$$

Entangled States (2)



We say that *two qubits are entangled* when their state is an entangled state

Intuitively, entangled qubits can be thought to be “linked” in some elusive, unintuitive way

Even if the two entangled qubits (e.g. two electrons) are physically separated (taken kilometers away from each other), they still remain “linked”

This elusive *nonclassical (quantum) correlation* between the two physically separated entangled qubits can be exploited to transfer information!

Bell States (EPR pairs)

Four special entangled two-qubit states are the *Bell states* or *EPR pairs*:

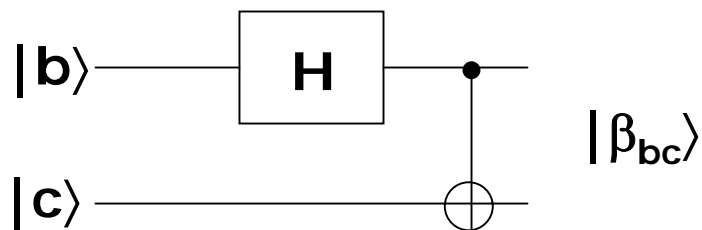
$$|\beta_{00}\rangle = |00\rangle + |11\rangle$$

$$|\beta_{01}\rangle = |01\rangle + |10\rangle$$

$$|\beta_{10}\rangle = |00\rangle - |11\rangle$$

$$|\beta_{11}\rangle = |01\rangle - |10\rangle$$

Note that the following quantum network maps $|bc\rangle$ to $|\beta_{bc}\rangle$



Exercise: show how a similar network can be used to effect a measurement w.r.t. the Bell basis by actually measuring w.r.t. the computational basis

Superdense Coding

Superdense coding is a two-party protocol that utilises a shared EPR pair in order to communicate *two* classical bits from one party (Alice) to the other (Bob) while only communicating *one* of the shared qubits

Suppose Alice and Bob each have one qubit of the state

$$|\beta_{00}\rangle = |00\rangle + |11\rangle$$

(Alice's qubit is on the left, Bob's on the right...)

Alice wishes to send two classical bits b,c to Bob

By operating only on her qubit, Alice can transform $|\beta_{00}\rangle$ into any of the four Bell states:

$$I \otimes I |\beta_{00}\rangle = |\beta_{00}\rangle$$

$$Y \otimes I |\beta_{00}\rangle = |\beta_{01}\rangle$$

$$X \otimes I |\beta_{00}\rangle = |\beta_{10}\rangle$$

$$Z \otimes I |\beta_{00}\rangle = |\beta_{11}\rangle$$

Protocol: Alice transforms $|\beta_{00}\rangle$ into $|\beta_{bc}\rangle$, and then sends her qubit to Bob. Bob measures the two qubits w.r.t. the Bell basis.

No-cloning Theorem

Suppose Alice would like to send a *copy* of her arbitrary qubit to Bob

She cannot!

Suppose that some quantum operation O maps

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

for all states $|\psi\rangle$

Then O must map

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

$$|\phi\rangle \rightarrow |\phi\rangle|\phi\rangle$$

$$|\psi\rangle + |\phi\rangle \rightarrow (|\psi\rangle + |\phi\rangle)(|\psi\rangle + |\phi\rangle)$$

But O is a quantum operation, so it must be linear and thus map

$$|\psi\rangle + |\phi\rangle \rightarrow |\psi\rangle|\psi\rangle + |\phi\rangle|\phi\rangle$$

This is a contradiction, thus O cannot exist

Quantum Teleportation

(Quantum) Teleportation is a two-party protocol that utilises a shared EPR pair in order to send *one qubit* from Alice to Bob while only communicating *two classical bits*

This does not violate the no-cloning theorem because Alice destroys the state of her qubit in the process

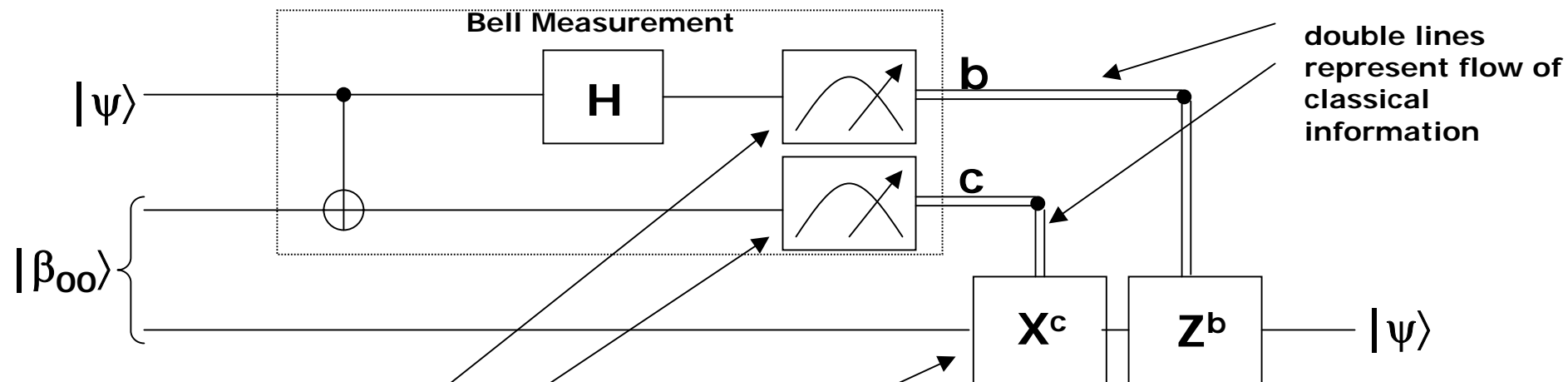
As in superdense coding, Alice and Bob share EPR pair $|\beta_{00}\rangle$ (with Alice's qubit on the left)

Suppose Alice wants to send the qubit $|\psi\rangle$ to Bob

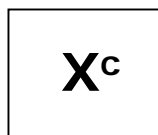
Exercise: verify
$$\begin{aligned} |\psi\rangle \otimes |\beta_{00}\rangle &= |\beta_{00}\rangle \otimes |\psi\rangle \\ &+ |\beta_{01}\rangle \otimes X|\psi\rangle \\ &+ |\beta_{10}\rangle \otimes Z|\psi\rangle \\ &+ |\beta_{11}\rangle \otimes XZ|\psi\rangle \end{aligned}$$

Protocol: Alice measures her two qubits w.r.t. the Bell basis getting outcome β_{bc} , and then sends bits b,c to Bob. Bob applies $Z^b X^c$ to his qubit (note $X^2=Z^2=I$).

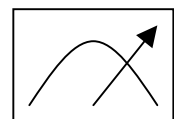
Quantum Network Diagram for Teleportation



double lines represent flow of classical information



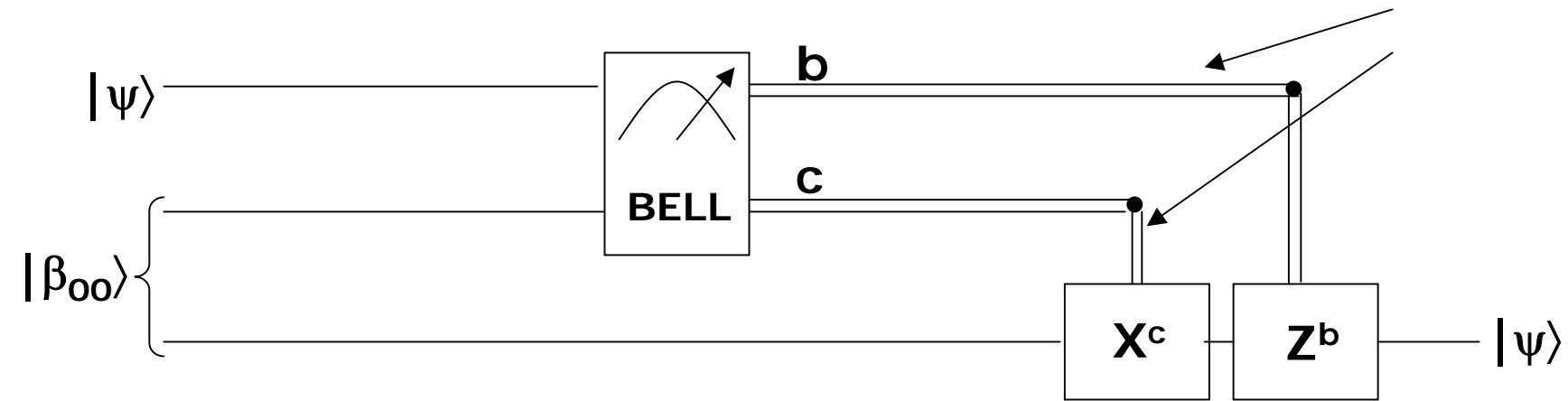
This symbol means "if $c=0$, do nothing, else apply X gate"; it is a *classically-controlled quantum gate*



This symbol represents measurement of a qubit with respect to the computational basis; the output of this "measurement gate" is classical (0 or 1)

In the teleportation protocol, the upper two qubits belong to Alice and the bottom qubit belongs to Bob

Quantum Network Diagram for Teleportation (2)



In this revised diagram, we have replaced the CNOT gate, Hadamard gate, and measurement with respect to the computational basis with a two-qubit “Bell measurement gate”

The physical motivation for this replacement is that, in some physical implementations, performing a measurement w.r.t. the Bell basis *directly* may be easier than performing a CNOT gate...

Teleportation: Not just a pretty face

...based on this assumption, teleportation can be used to fight against errors in the implementation of quantum gates.

It may be that a particular implementation of the CNOT gate is not perfect (fails with some probability)

If a CNOT gate fails midway through a complex computation, then the entire computation might be ruined

Normally, quantum error-correction codes are used to fight against errors

Alternatively, teleportation can be used to reduce the application of an imperfect CNOT gate to the preparation of a particular 4-qubit entangled state

$$|\Psi\rangle = (|0000\rangle + |0011\rangle + |1110\rangle + |1101\rangle)/2$$

Instead of a CNOT gate possibly failing, the preparation of $|\Psi\rangle$ may fail (but this failure does not ruin the computation)

Teleportation: Not just a pretty face (2)

The following sequence of *equivalent* network diagrams illustrates the method

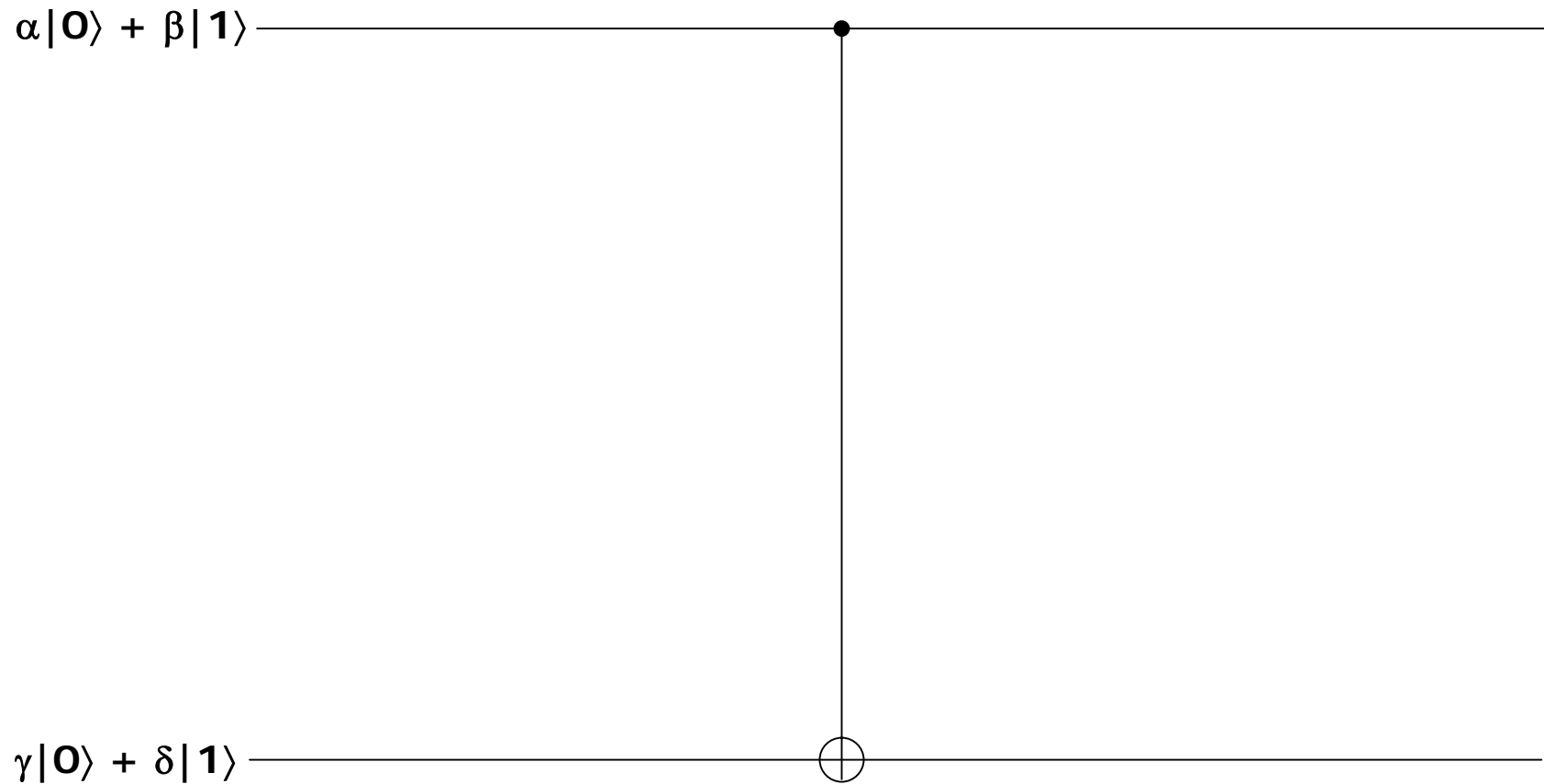


Diagram 1

Teleportation: Not just a pretty face (3)

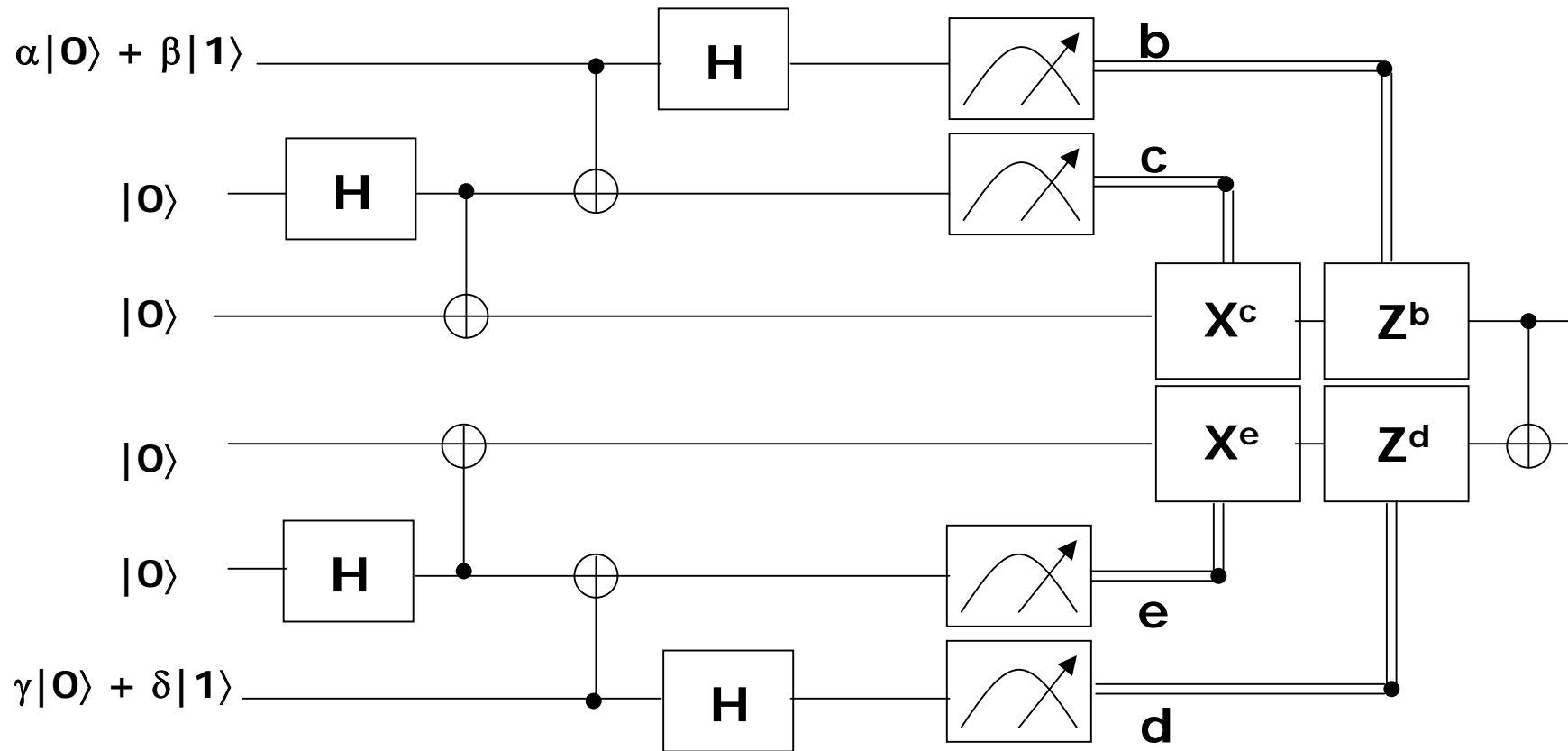


Diagram 2

Added four ancillary qubits in order to teleport the outer two qubits to the inner two qubits, and then apply the CNOT to the inner two qubits

Teleportation: Not just a pretty face (4)

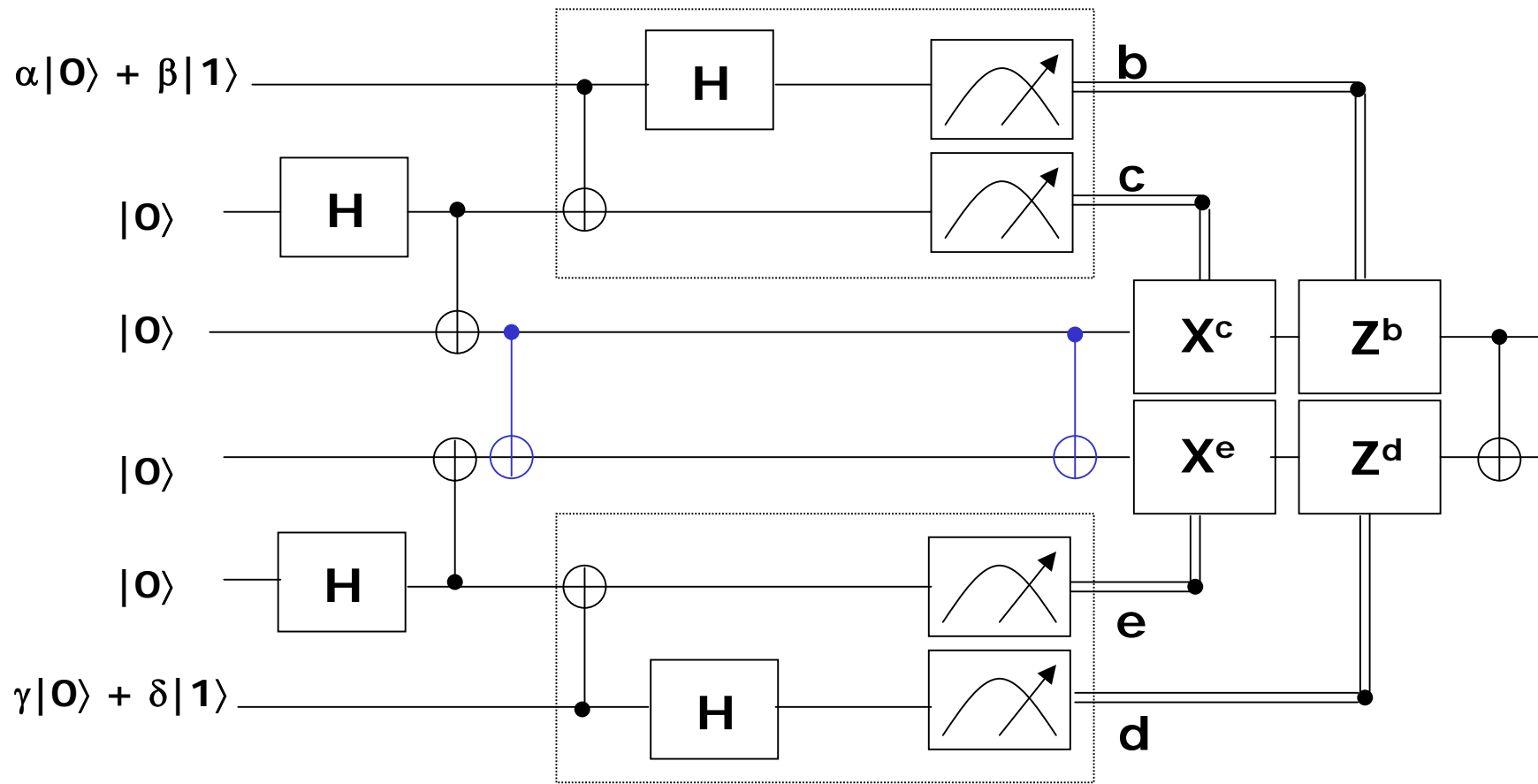


Diagram 3

Added two CNOT gates on the inner two qubits

Teleportation: Not just a pretty face (5)

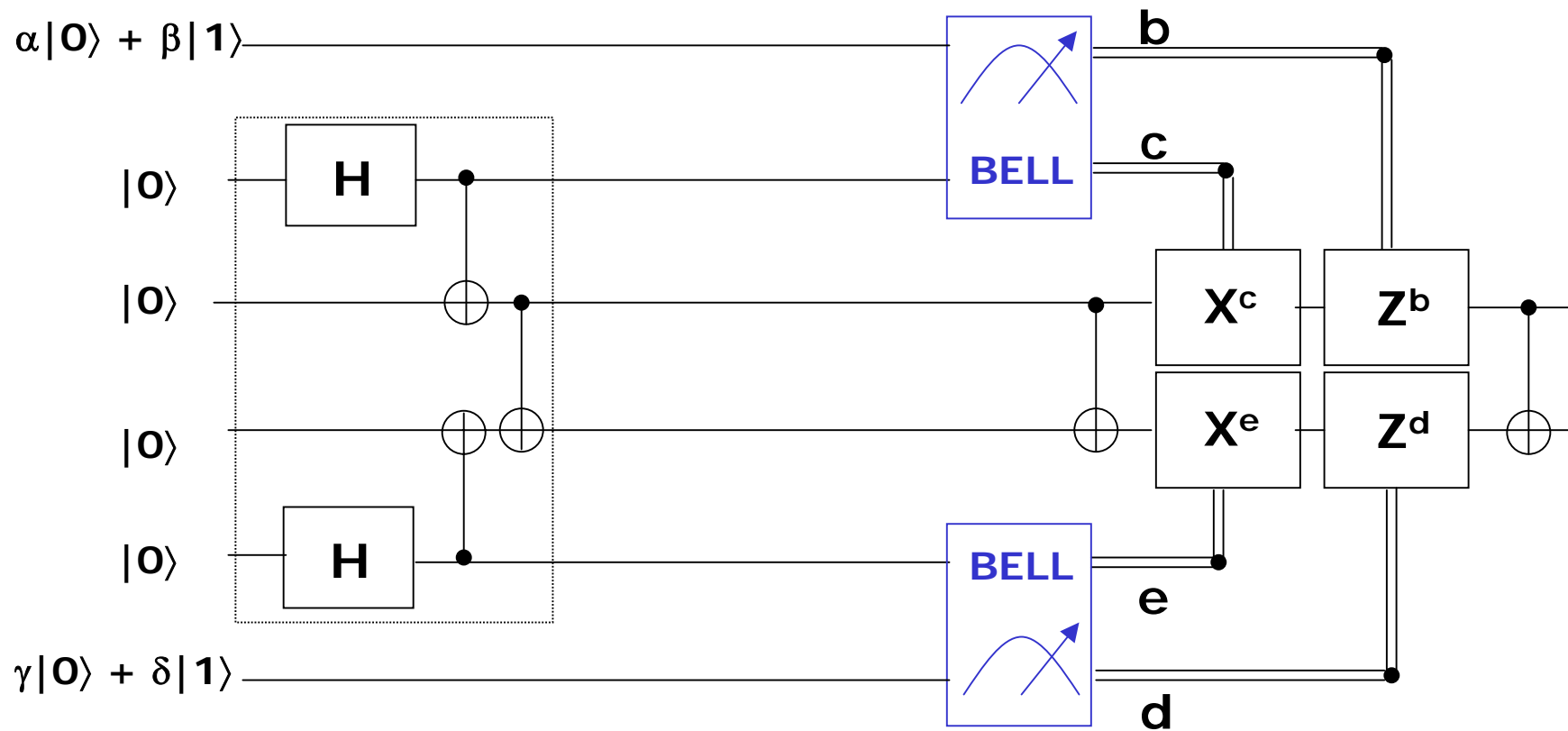


Diagram 4

Replaced two instances of CNOT, Hadamard, and measurement w.r.t. computational basis with "Bell measurement gate" (based on assumption that a direct Bell measurement may be easier to implement than a CNOT gate)

Teleportation: Not just a pretty face (6)

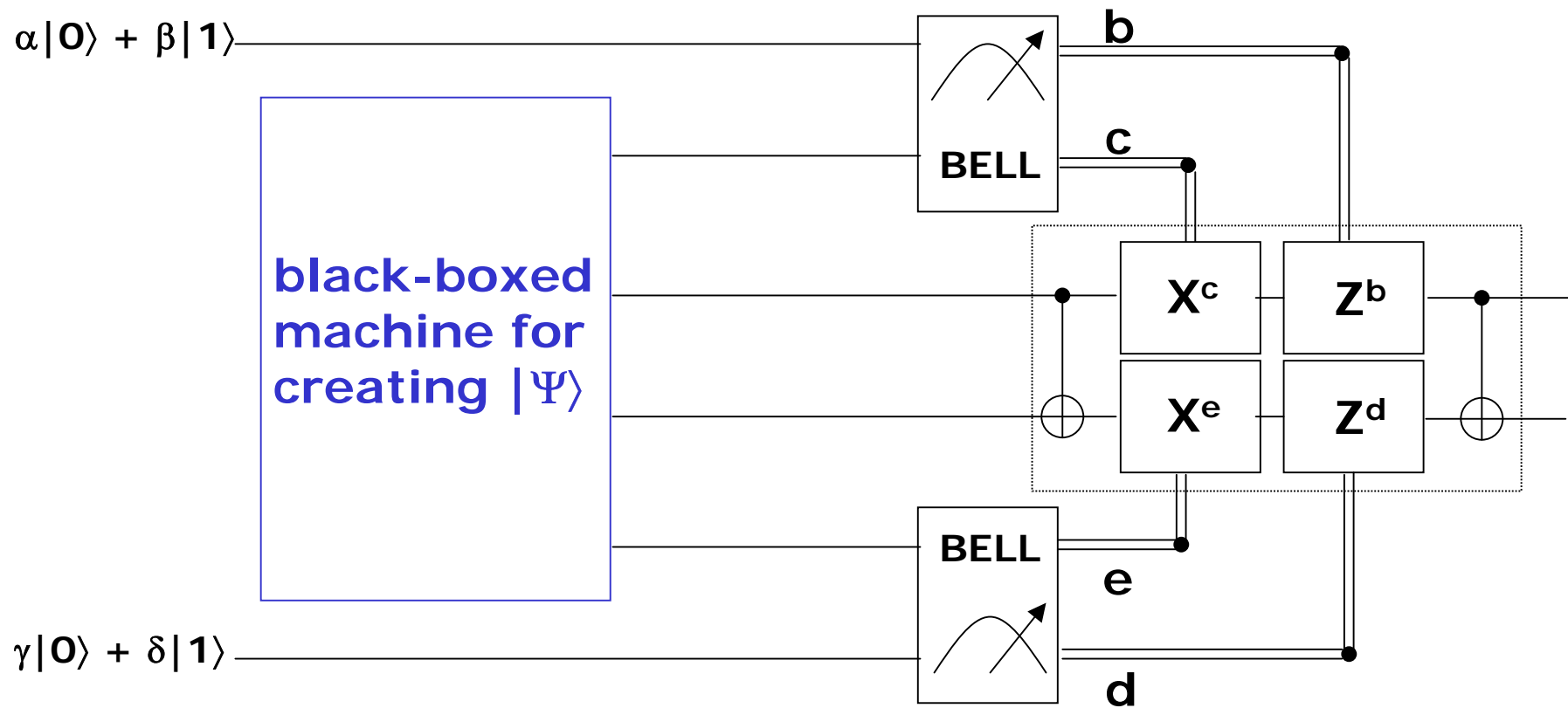


Diagram 5

Replaced dashed box on previous slide with a machine for producing the state

$$|\Psi\rangle = (|0000\rangle + |0011\rangle + |1110\rangle + |1101\rangle)/2$$

Main idea is that failed attempts to produce $|\Psi\rangle$ occur "off line" (inside the black box) and hence do not affect the computation on the outer qubits

Teleportation: Not just a pretty face (7)

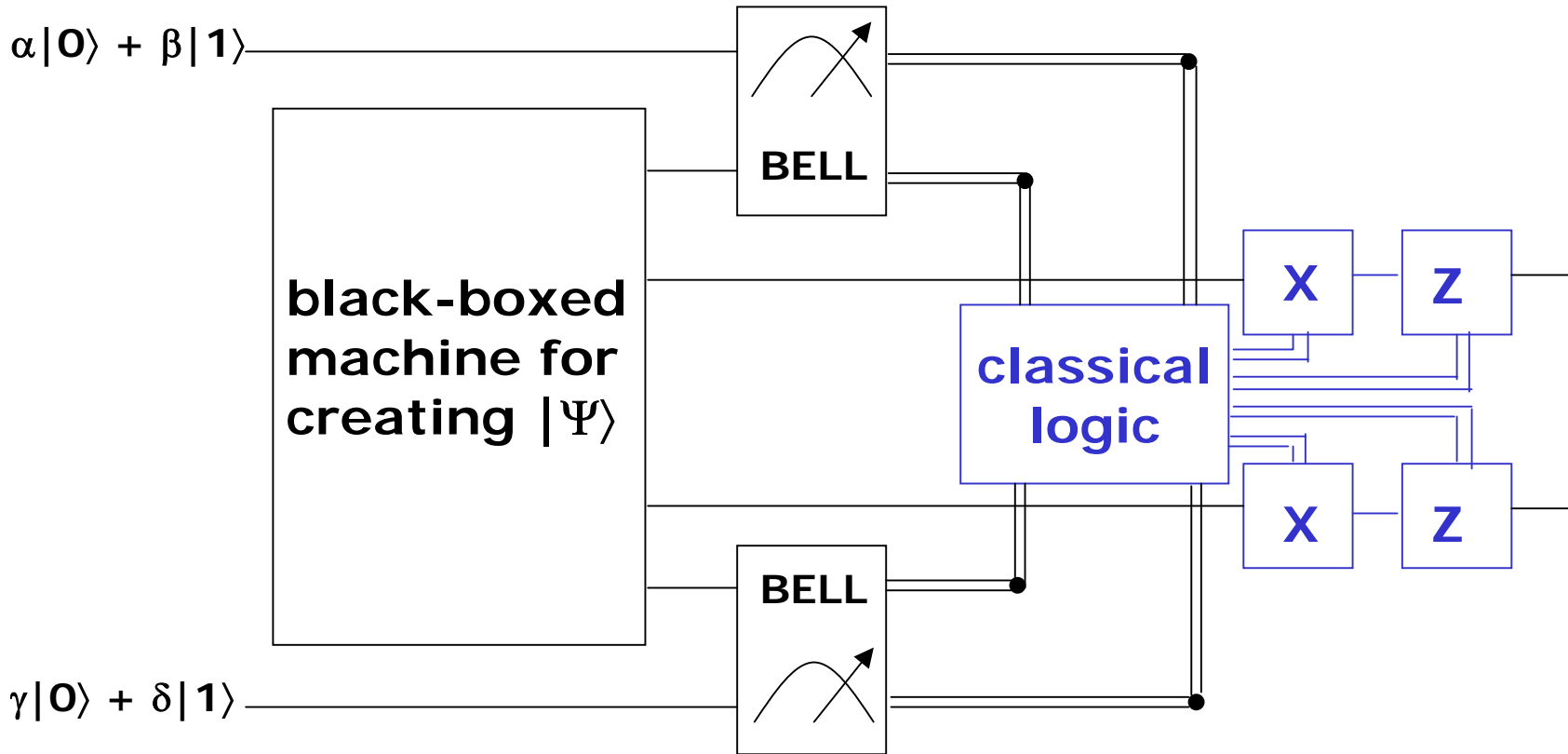


Diagram 6

Exercise: Show that this final replacement is valid; i.e. show that the two CNOT gates in the dashed box on the previous slide can be removed as long as the classical logic controlling the X and Z gates is appropriately modified

8-lecture Mini-course in Quantum Computation

Lecture 4

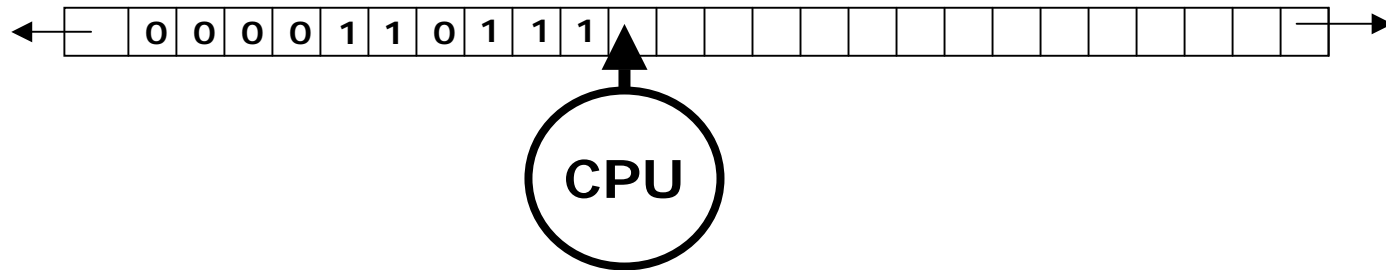
Lawrence Ioannou

Models of Computation

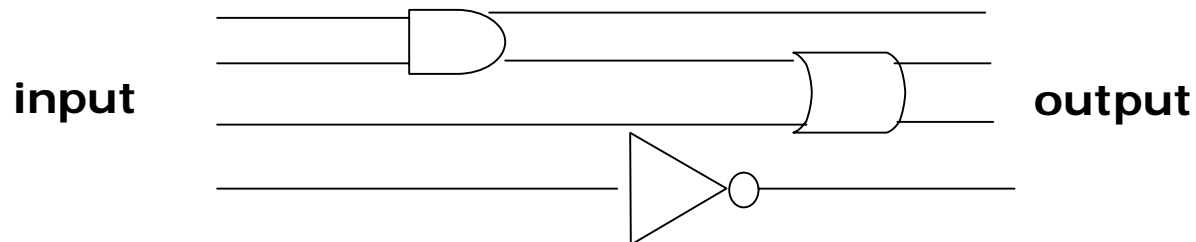
Recall the polytime-equivalent classical models:

Turing Machine

- Idealised computer
- Two-way infinite tape, mobile read/write head, CPU (transfer function)



Reversible acyclic gate array (reversible circuit model)



Model of Computation (2)

Although one can define a *quantum Turing Machine*, we will use the quantum analogue of the classical reversible circuit model, known as the *quantum circuit model*

More specifically, we restrict to *uniform families* of quantum networks (or *quantum gate arrays*, or *quantum circuits*):

For a given *problem*, there exists a classical Turing Machine that, given the *input size*, n , of an *instance* of the problem, generates a quantum network diagram to solve the instance in *poly*(n) steps

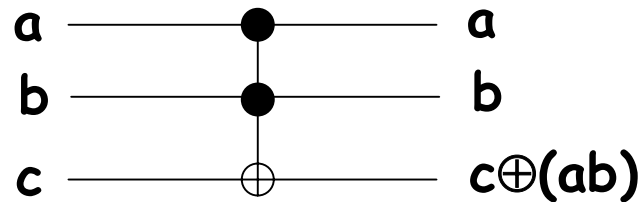
This ensures that the proposed quantum solution of the problem is (considered) *efficient*

Such a uniform family of quantum networks is an *efficient algorithm*

Universality

Recall that the gates *AND*, *OR*, and *NOT* (or just the *NAND*-gate) are *universal* for classical (non-reversible) computation

For reversible classical computation, the three-bit *Toffoli gate* (or controlled-controlled-*NOT*) is universal



Toffoli gate

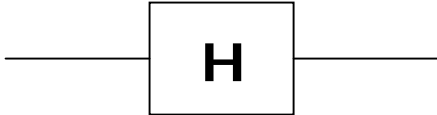


Universality (2)

A fixed set G of quantum gates is *universal* for quantum computation if any unitary operator can be approximated to arbitrary accuracy by a quantum network using only gates from G

Some universal sets of gates for quantum computation:

$$G = \{\text{CNOT}\} \cup \{\text{all one-qubit gates}\}$$

$$G = \{\text{CNOT}\} \cup \{\text{H, S, T}\}$$

U	[U]	Gate symbol
Hadamard	$\frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$	
"phase"	$\begin{bmatrix} 1 & 0 \\ 0 & i \end{bmatrix}$	
" π -by-8"	$\begin{bmatrix} 1 & 0 \\ 0 & e^{i\pi/4} \end{bmatrix}$	

What's so great about quantum algorithms

Efficient quantum algorithms exist for problems such as **INTEGER FACTORING** and **DISCRETE LOGARITHM**

- **INTEGER FACTORING:** Given integer N , such that $N=pq$ for odd prime numbers p and q ; find p

- **DISCRETE LOGARITHM:** Given integers N, b, y ; find integer x such that $y = b^x \bmod N$ (i.e. N divides b^x with remainder y)

- Efficient classical algorithms are not known to exist for these problems which form the basis for computational secure cryptosystems like RSA and El Gamal

Quantum algorithms can efficiently simulate quantum physics (see 4.7 in textbook)

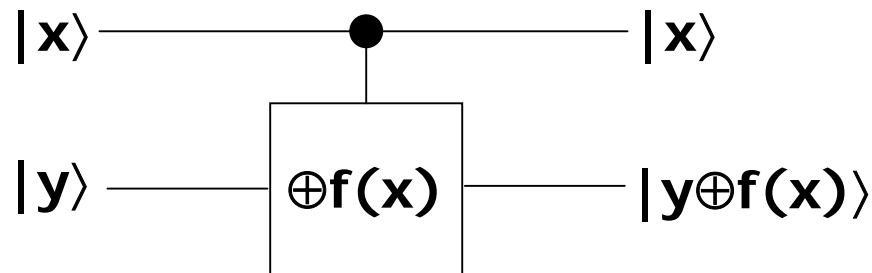
Problems as black box functions

Many computational problems can be expressed in terms of a *black box function* f

$$f: G \rightarrow \{0, 1, \dots, N-1\}$$

where the “answer” to the problem is some (hidden) property of f (G is some set)

We assume that we can query the black box, i.e. give it an input x and get out $f(x)$ (reversibly)



here, “ \oplus ” means bit-wise addition modulo 2

Problems as black box functions (2)

Some examples:

Searching problem:

Given black box for $f: \{0,1\}^n \rightarrow \{0,1\}$

Find an x such that $f(x)=1$

($\{0,1\}^n$ is the “database”)

Period-finding problem:

Given black box for $f: \{0,1,\dots\} \rightarrow \{0,1,\dots,N-1\}$, where f is periodic, $f(x+r)=f(x)$

Find the period r

e.g. $f(x)=a^x \bmod N$ (related to INTEGER FACTORING)

Classical v. Quantum Computers

The state of a classical reversible computer is confined to being one of the computational basis states at any time (queries to the black box for f can only be made one at a time)

Quantum computers can branch out over exponentially many computational basis states, like

$$\sum_{x=0}^{2^n-1} \frac{1}{\sqrt{2^n}} |x\rangle$$

Using the black box for f *only once*, one can then evaluate $f(x)$ for exponentially many x in superposition:

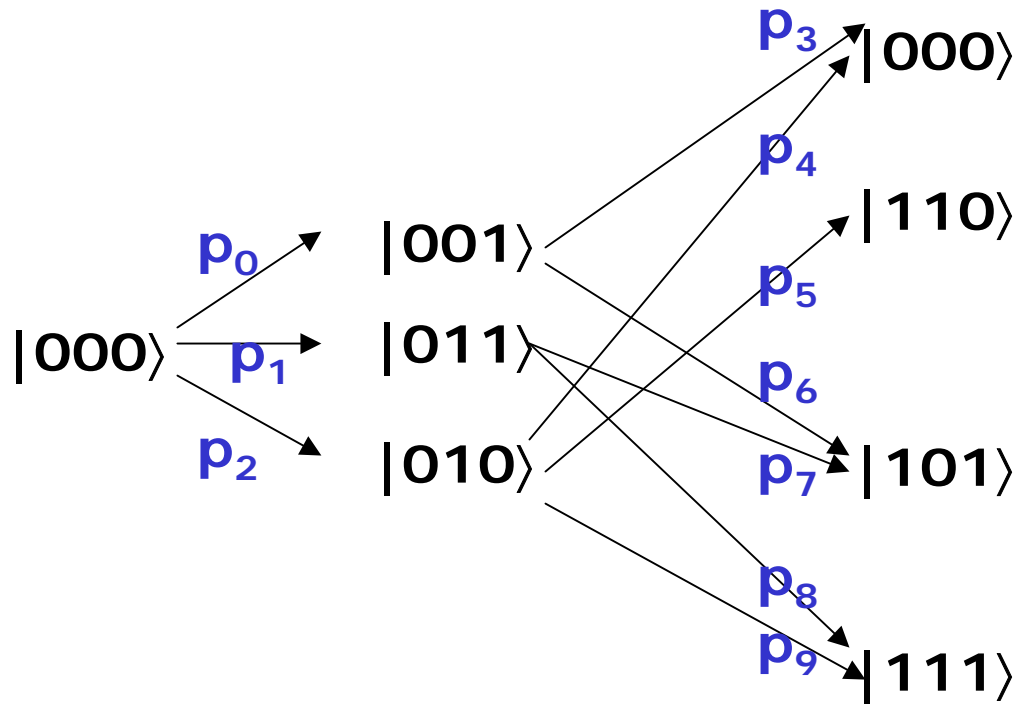
$$\sum_{x=0}^{2^n-1} \frac{1}{\sqrt{2^n}} |x\rangle |f(x)\rangle$$

Such states can be further processed (quantumly) to extract hidden properties of f

Randomised Classical v. Quantum Computers (1)

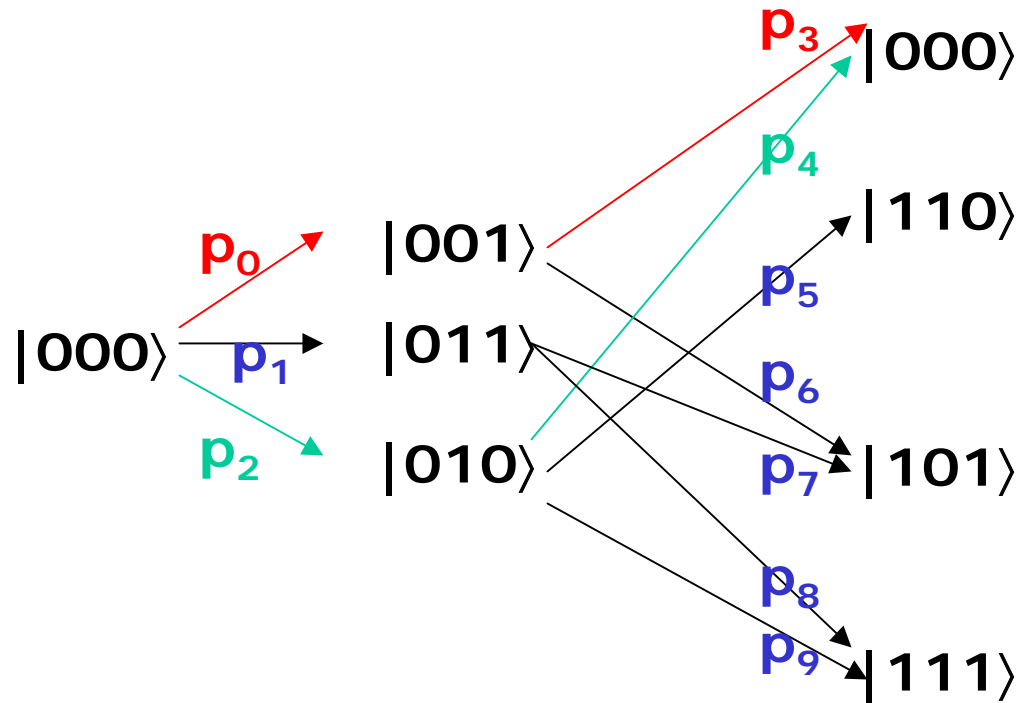
Recall that a deterministic computation can be regarded as a path through “configuration space” of all configurations of a Turing machine (each configuration corresponds to an element of the computational basis)

A randomised computation can be regarded as a tree



where each branch has a probability p_i associated with it

Randomised Classical v. Quantum Computers (2)



The outcome $|000\rangle$ in this computation can be reached by two paths (red and green)

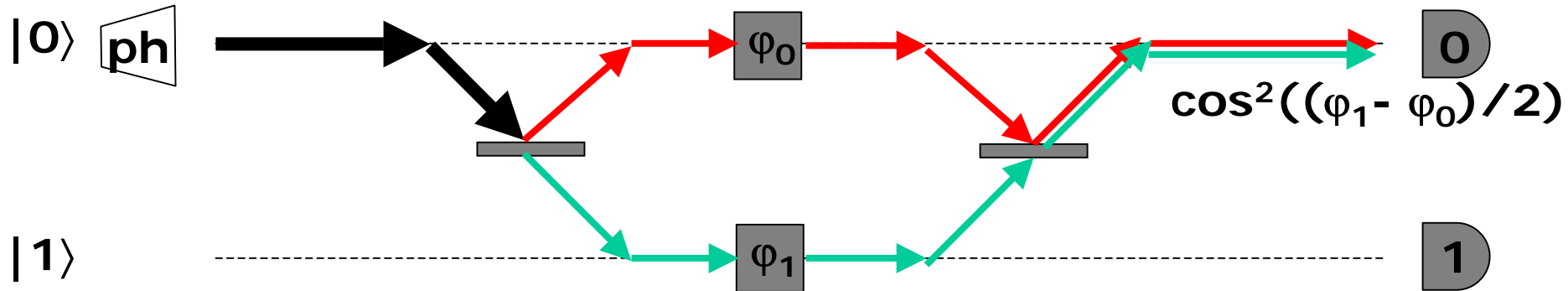
Probability of reaching $|000\rangle$ by the red path is $|a_{00}|^2 = p_0 p_3$

Probability of reaching $|000\rangle$ by the green path is $|a_{10}|^2 = p_2 p_4$

The *total* probability of reaching $|000\rangle$ is thus $|a_{00}|^2 + |a_{10}|^2$

Randomised Classical v. Quantum Computers (3)

In our interferometry experiment, recall that there are two “computational paths” that lead to the outcome 0 (red path and green path):



The probability *amplitude* of reaching 0 by the red path is

$$a_{00} = \exp(i\varphi_0)/2$$

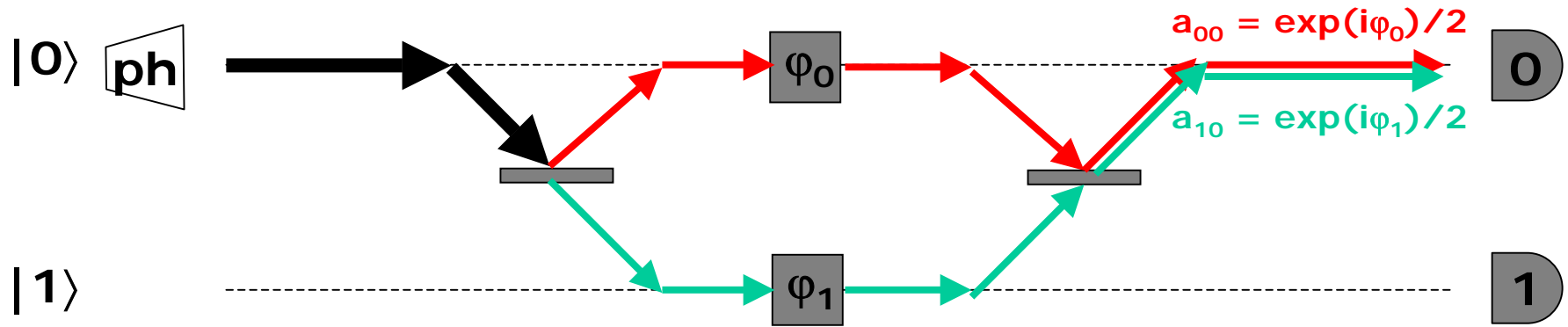
The probability *amplitude* of reaching 0 by the green path is

$$a_{10} = \exp(i\varphi_1)/2$$

The *total* probability of reaching 0 is

$$|a_{00} + a_{10}|^2 = \cos^2((\varphi_1 - \varphi_0)/2)$$

Randomised Classical v. Quantum Computers (4)



The *total* probability of reaching 0 is $|a_{00} + a_{10}|^2 = \cos^2((\phi_1 - \phi_0)/2)$

The *relative phase* between the probability amplitudes of the two paths matters (no such concept in the classical case), and can result in *constructive* or *destructive* interference

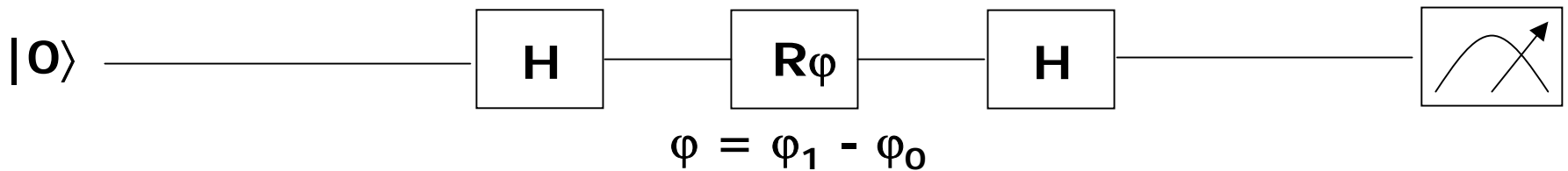
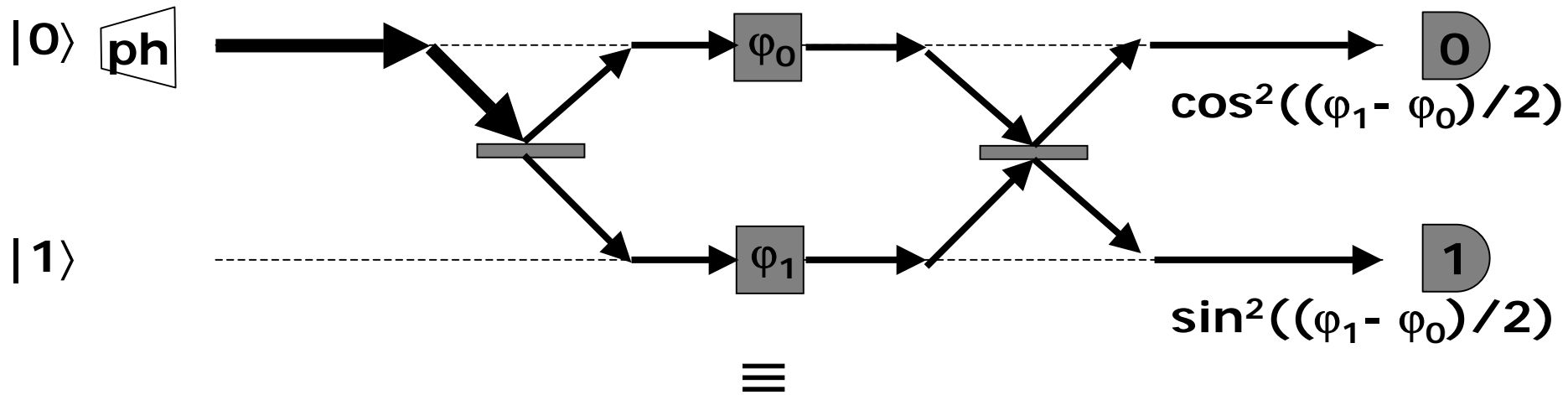
e.g. destructive interference occurs when $a_{00} = -a_{10}$,

e.g. constructive interference occurs when $a_{00} = a_{10}$

One goal of quantum algorithms is to induce constructive interference on *good* outcomes and destructive interference on *bad* outcomes

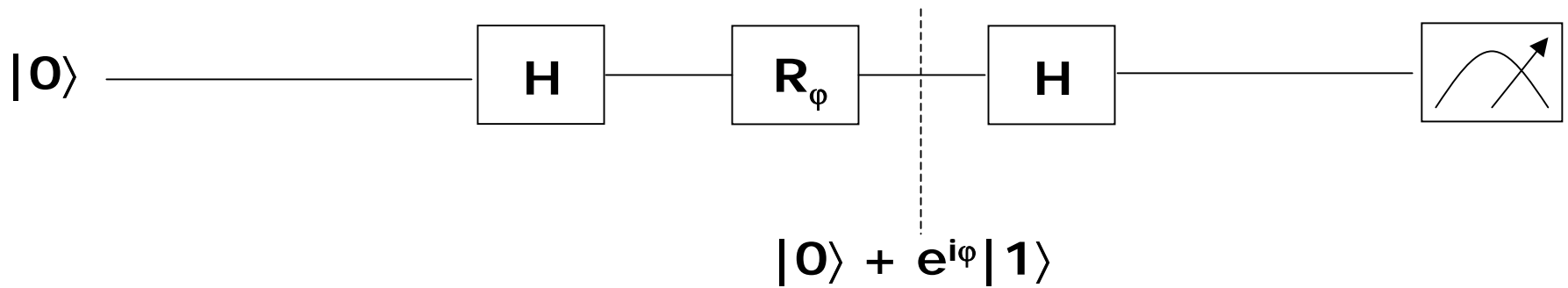
Quantum Algorithms as Interferometry

Most quantum algorithms can be viewed as big interferometry experiments

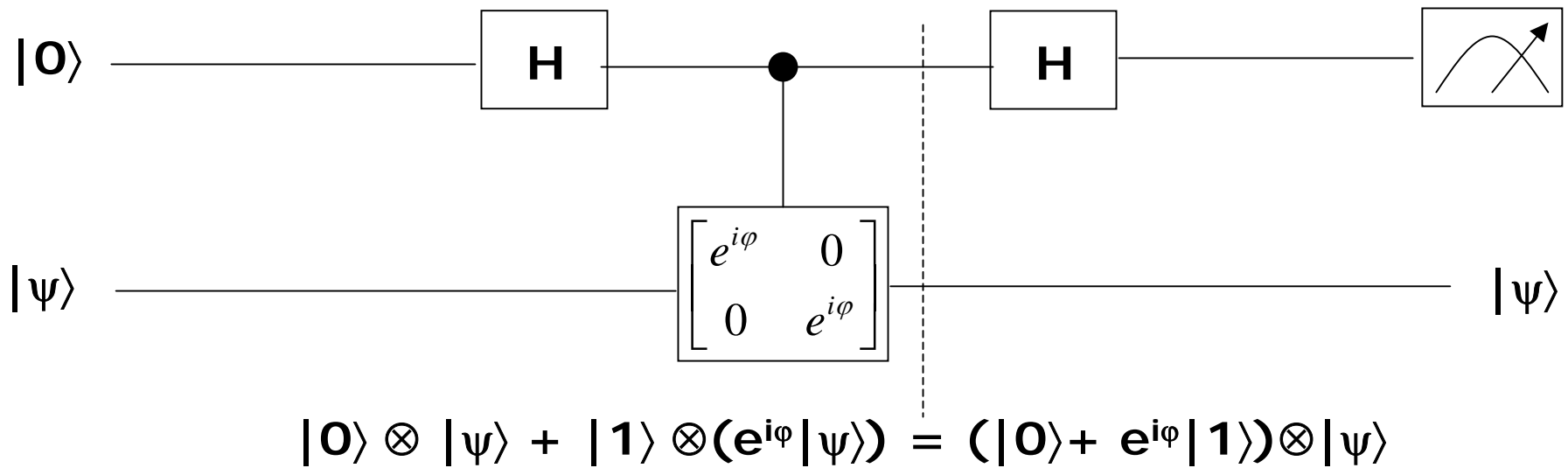


Basic idea: the measurement can distinguish the two cases $\varphi=0$ and $\varphi=\pi$

Other ways to introduce a relative phase



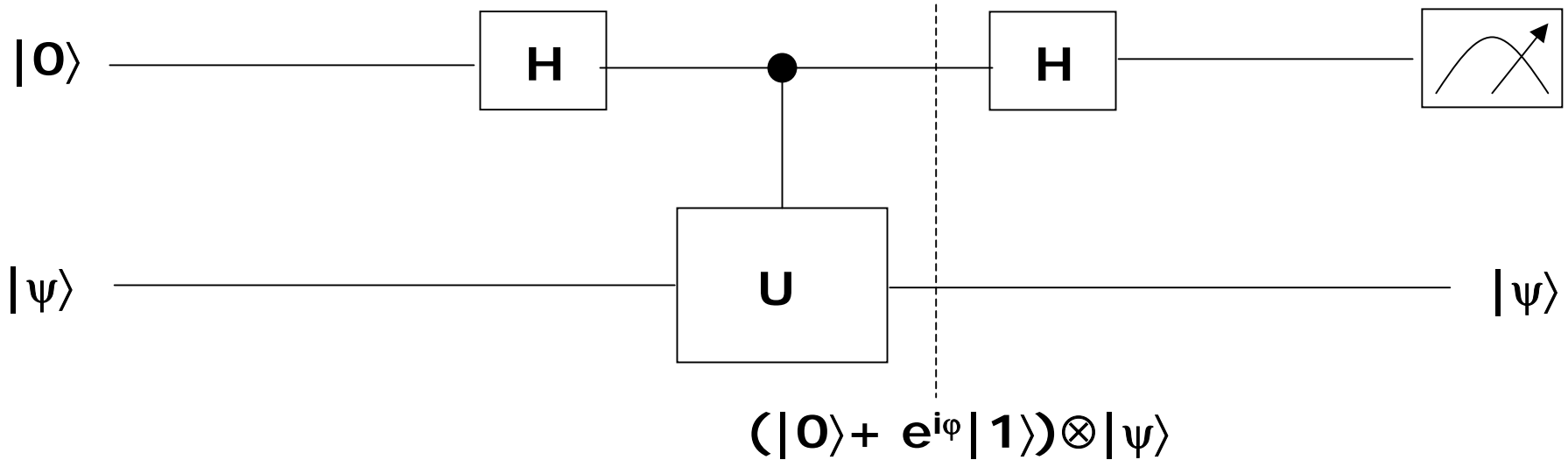
is equivalent to



with respect to the top qubit; bottom qubit was unchanged...

Other ways to introduce a relative phase (2)

... more generally, the bottom qubit will “kick back” a *relative phase (eigenvalue)* in the top qubit if the bottom qubit is in an *eigenstate* of U :



where

$$U|\psi\rangle = e^{i\phi}|\psi\rangle$$

This so-called “eigenvalue kick-back” is a useful mechanism by which to *analyse* (though, not necessarily *implement*) quantum algorithms

Deutsch's problem

Given a black box which computes the function

$f: \{0,1\} \rightarrow \{0,1\}$ such that

$$f(0) = f(1)$$

or

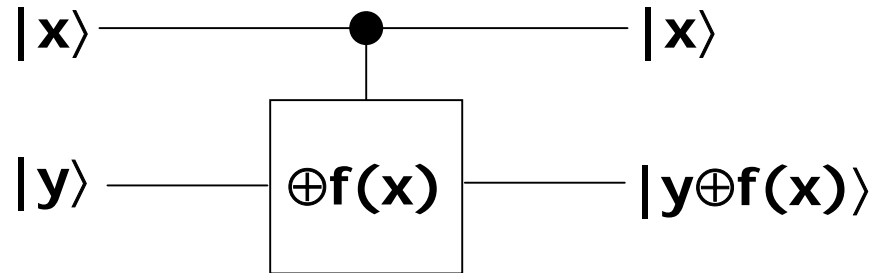
$$f(0) \neq f(1)$$

distinguish the two cases i.e. compute $f(0) \oplus f(1)$ (with *as few evaluations as possible* of the black box)

If restricted to classical computation (classical black box), then the number of evaluations of the black box is 2; evaluating the black box only once can *never* solve the problem (this is easily verified)

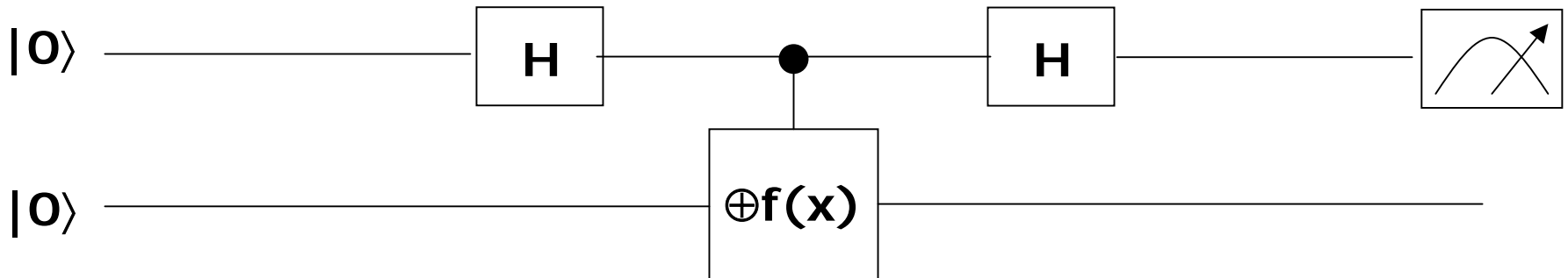
Deutsch's problem (2)

However, suppose one has a quantum gate that *computes* f :

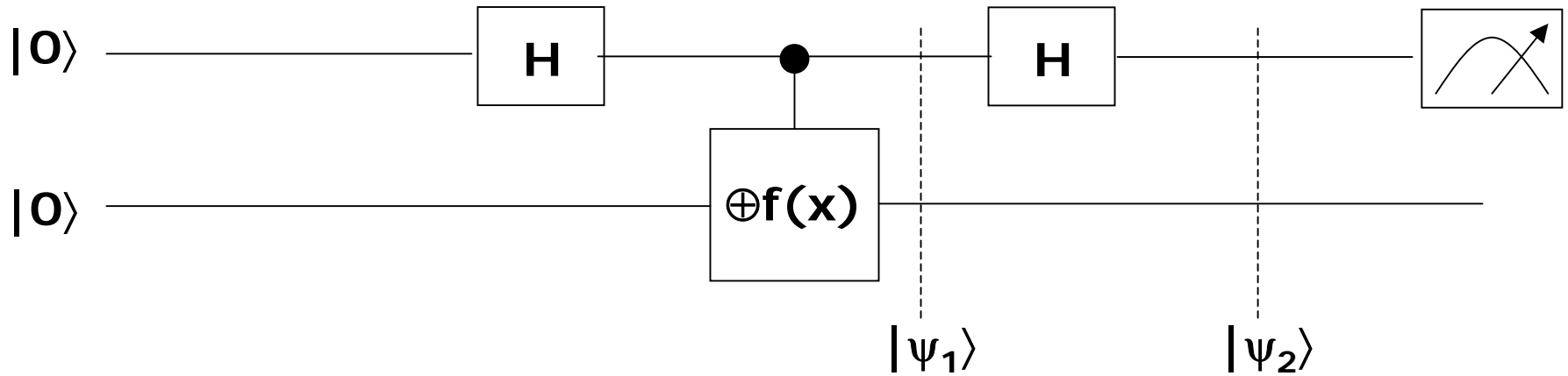


for $x, y \in \{0, 1\}$

Then the following quantum network solves Deutsch's problem with probability $\frac{1}{2}$



Deutsch's problem (3)



$$|\psi_1\rangle = |0\rangle|f(0)\rangle + |1\rangle|f(1)\rangle$$

$$|\psi_2\rangle = (|0\rangle + |1\rangle)|f(0)\rangle + (|0\rangle - |1\rangle)|f(1)\rangle$$

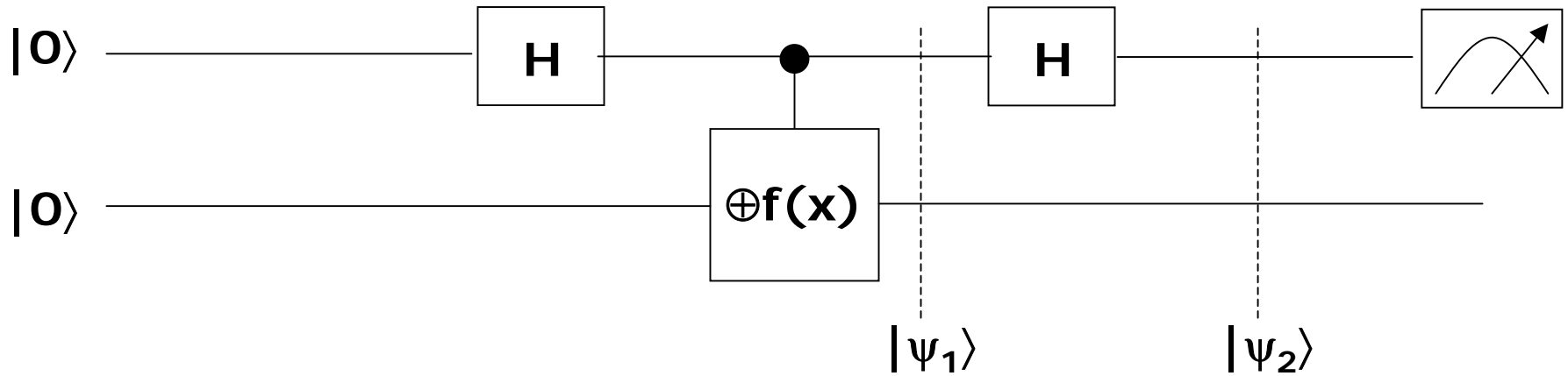
$$= |0\rangle(|f(0)\rangle + |f(1)\rangle) + |1\rangle(|f(0)\rangle - |f(1)\rangle)$$

If $f(0) = f(1)$, then cannot measure outcome 1 in upper qubit

Thus, if one does get outcome 1, then one may conclude that $f(0) \neq f(1)$ (if one gets outcome 0, then either case may hold)

This solution to Deutsch's problem is already better (on average) than any classical solution; but we can do better...

Deutsch's problem (4)



Note $|\psi_1\rangle = |0\rangle|f(0)\rangle + |1\rangle|f(1)\rangle$

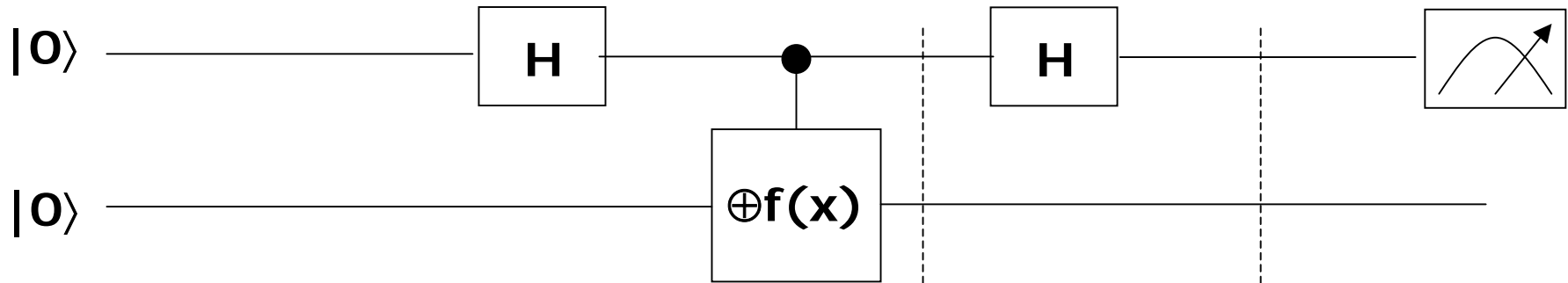
$$= [|0\rangle + |1\rangle] (|0\rangle + |1\rangle) + [(-1)^{f(0)}|0\rangle + (-1)^{f(1)}|1\rangle] (|0\rangle - |1\rangle)$$

$$= [|0\rangle + |1\rangle] (|0\rangle + |1\rangle) + (-1)^{f(0)} [|0\rangle + (-1)^{f(0)+f(1)}|1\rangle] (|0\rangle - |1\rangle)$$

Now $|\psi_2\rangle = |0\rangle (|0\rangle + |1\rangle) + (-1)^{f(0)} |f(0) \oplus f(1)\rangle (|0\rangle - |1\rangle)$

The measurement now clearly gives the answer $f(0) \oplus f(1)$ with probability 50% (the other half of the time, the measurement gives 0, which is inconclusive)

Deutsch's problem (5)



$$\begin{aligned}
 |\psi_1\rangle &= |0\rangle|f(0)\rangle + |1\rangle|f(1)\rangle \\
 &= [|0\rangle + |1\rangle] (|0\rangle + |1\rangle) + [(-1)^{f(0)} |0\rangle + (-1)^{f(1)} |1\rangle] (|0\rangle - |1\rangle)
 \end{aligned}$$

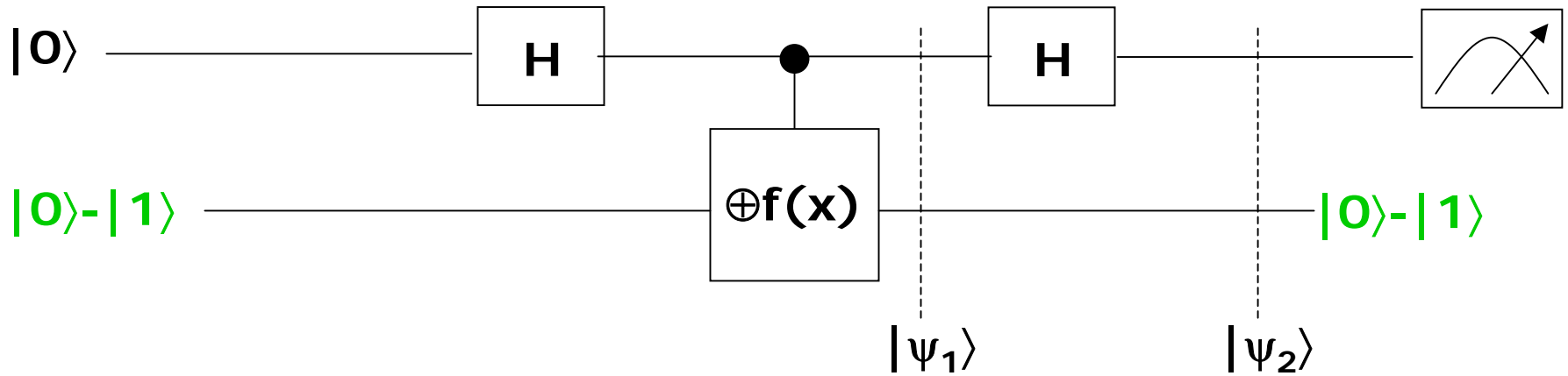
$$|\psi_2\rangle = |0\rangle (|0\rangle + |1\rangle) + (-1)^{f(0)} |f(0) \oplus f(1)\rangle (|0\rangle - |1\rangle)$$

BIG IDEA: $|x\rangle (|0\rangle + |1\rangle)$ and $|x\rangle (|0\rangle - |1\rangle)$ are *eigenvectors* of the operation $|x\rangle|b\rangle \rightarrow |x\rangle|b \oplus f(x)\rangle$, with eigenvalues **1** and $(-1)^{f(x)}$

Thus the controlled- $(\oplus f(x))$ gate actually induces a relative phase in the top qubit, via an eigenvalue kick-back

Since the $|0\rangle - |1\rangle$ eigenvector is the one that produces the **desired** relative phase in the top qubit...

Deutsch's problem (6)

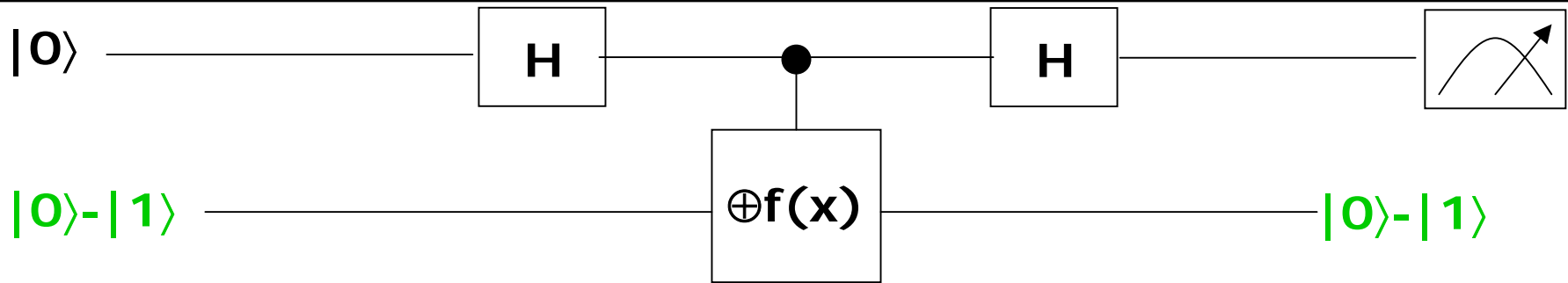


$$|\psi_1\rangle = [(-1)^{f(0)} |0\rangle + (-1)^{f(1)} |1\rangle] (|0\rangle - |1\rangle)$$

$$|\psi_2\rangle = (-1)^{f(0)} |f(0) \oplus f(1)\rangle (|0\rangle - |1\rangle)$$

This revised algorithm succeeds with probability 1

Deutsch's problem (7)



Deutsch's problem seems to be special: because of its simplicity, the operation $|b\rangle \rightarrow |b \oplus f(x)\rangle$ can be analysed in a useful eigenbasis, namely $\{ |0\rangle + |1\rangle, |0\rangle - |1\rangle \}$

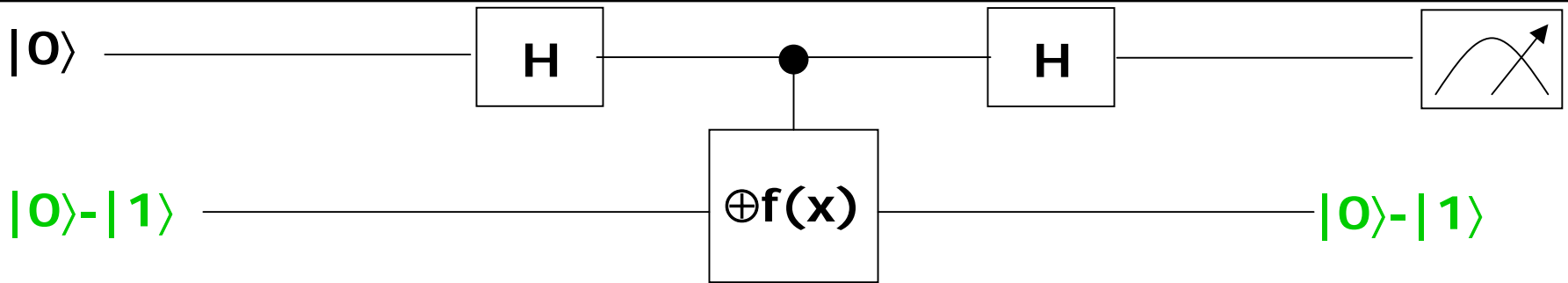
But for more general problems, like period-finding (and the general *hidden subgroup problem*), we introduce the *shift operation*, $U_{\text{sh}(f)}$:

$$U_{\text{sh}(f)}: |f(x)\rangle \rightarrow |f(x+1)\rangle$$

For general f , $U_{\text{sh}(f)}$ may not be implementable, because f is not necessarily one-to-one

However, $U_{\text{sh}(f)}$ is a powerful analysis tool...

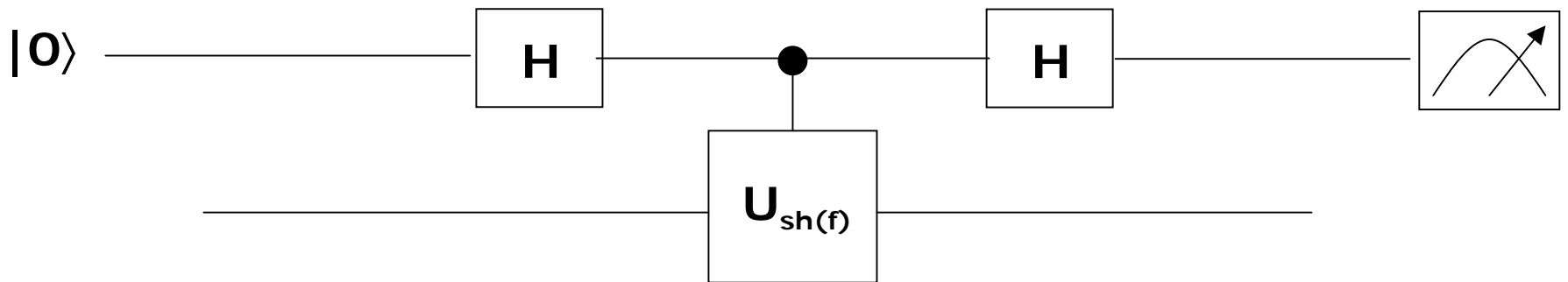
Deutsch's problem (8)



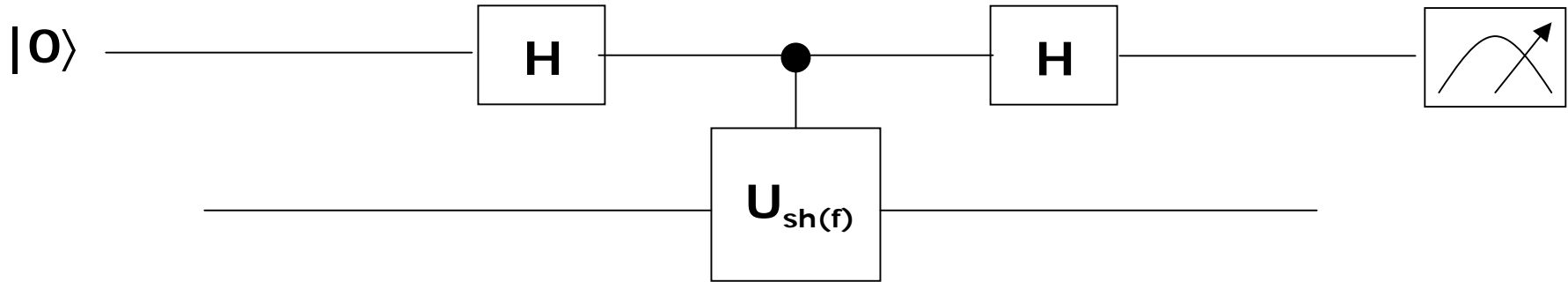
Instead of the controlled- $\oplus f(x)$ gate (above), assume we have a *controlled- $U_{sh(f)}$* gate which maps

$$|0\rangle|f(x)\rangle \rightarrow |0\rangle|f(x)\rangle$$

$$|1\rangle|f(x)\rangle \rightarrow |0\rangle|f(x+1)\rangle$$



Deutsch's problem (9)



Note: $f(0)=f(1)$ implies $U_{sh(f)} = I$

$f(0)\neq f(1)$ implies $U_{sh(f)} = X$

Both I and X have eigenvectors $\{ |0\rangle + |1\rangle, |0\rangle - |1\rangle \}$, but I has eigenvalues $\{ 1, 1 \}$ whereas X has eigenvalues $\{ 1, -1 \}$

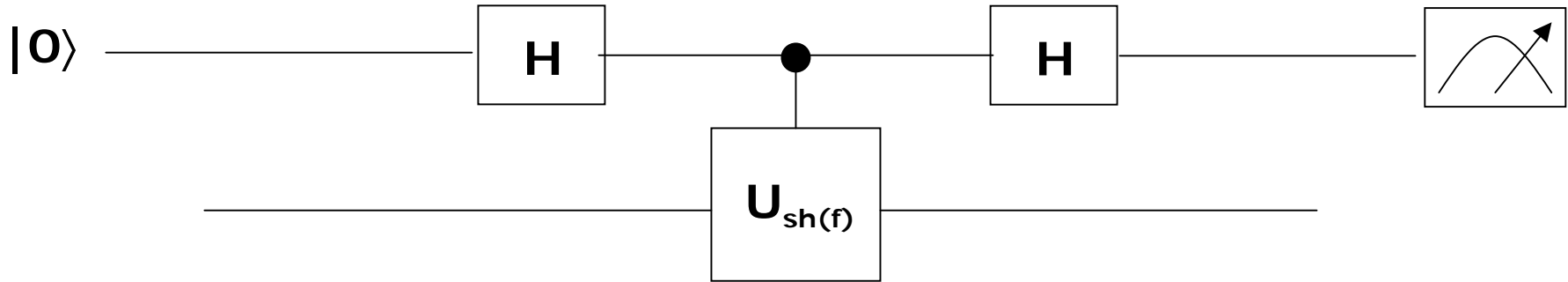
So, $U_{sh(f)}$ has eigenvectors $\{ |0\rangle + |1\rangle, |0\rangle - |1\rangle \}$ with eigenvalues

$$\{ 1, (-1)^{f(0)\oplus f(1)} \}$$

or, writing the eigenvalues another way,

$$\{ 1, e^{i\pi f(0)\oplus f(1)} \}$$

Deutsch's problem (10)



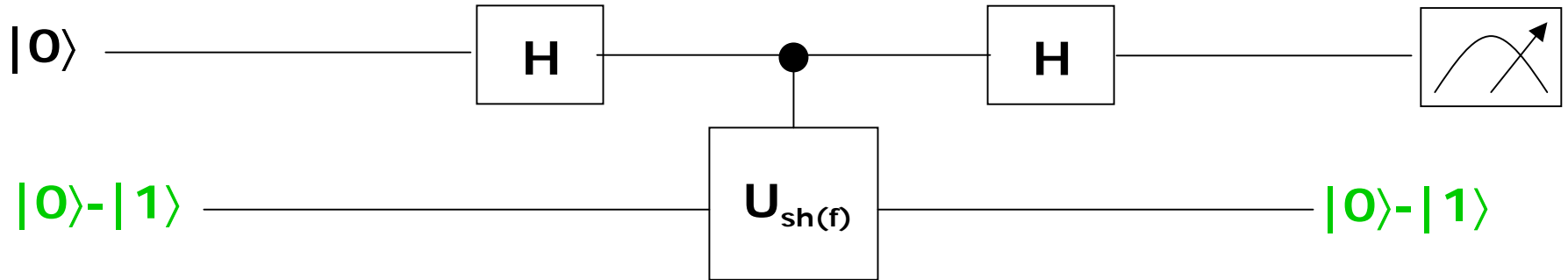
Eigenvalues of $U_{sh(f)}$ are $\{1, e^{i\pi f(0)\oplus f(1)}\}$

Thus, the answer $f(0)\oplus f(1)$ is encoded in the "phase" of an eigenvalue of the shift operator!

We know that if we input the eigenvector $|0\rangle - |1\rangle$ in the bottom register, the controlled-shift gate will kick back this relative phase into the top qubit; the phase $\varphi = \pi f(0)\oplus f(1)$ is either 0 or π

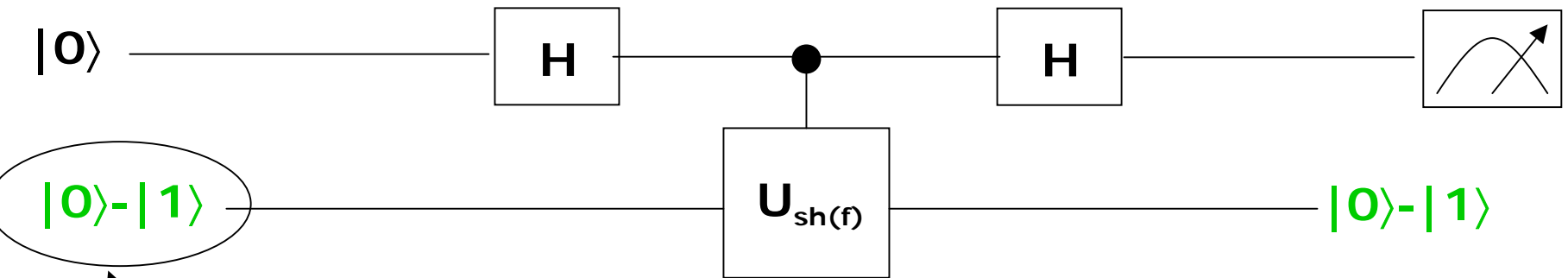
From our interferometry experiment, we know we can distinguish the two cases $\varphi = 0$ or $\varphi = \pi$...

Deutsch's problem (11)



The above network thus solves Deutsch's problem with probability 1

Controlled- $\oplus f(x)$ v. Controlled- $U_{sh(f)}$



In most cases, the desired eigenvector is not known

However

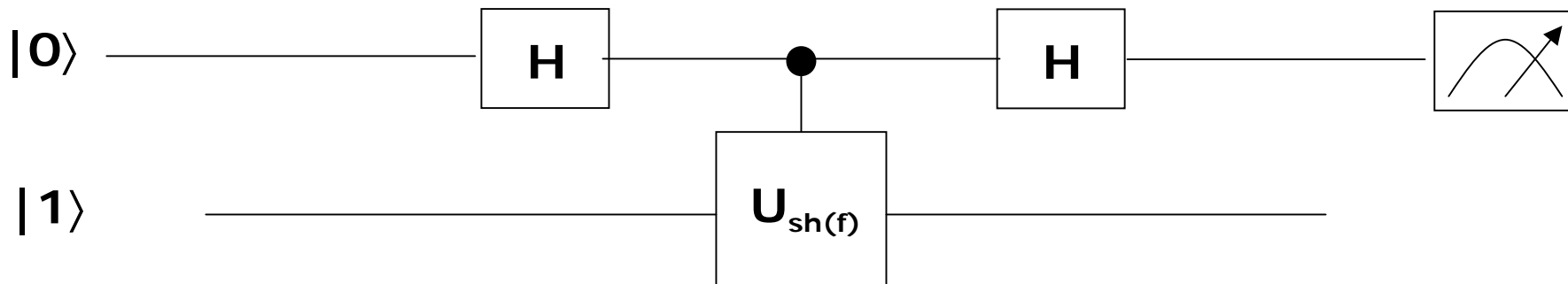
$$|0\rangle = (|0\rangle + |1\rangle) + (|0\rangle - |1\rangle)$$
$$|1\rangle = (|0\rangle + |1\rangle) - (|0\rangle - |1\rangle)$$

It turns out we can *always* resort to inputting an equal superposition of eigenvectors of the shift operator, which will give the desired eigenvalue kick back in the top qubit with some reasonable probability (in this case $\frac{1}{2}$)

(We actually already saw this effect in the controlled- $\oplus f(x)$ solution to Deutsch's problem)

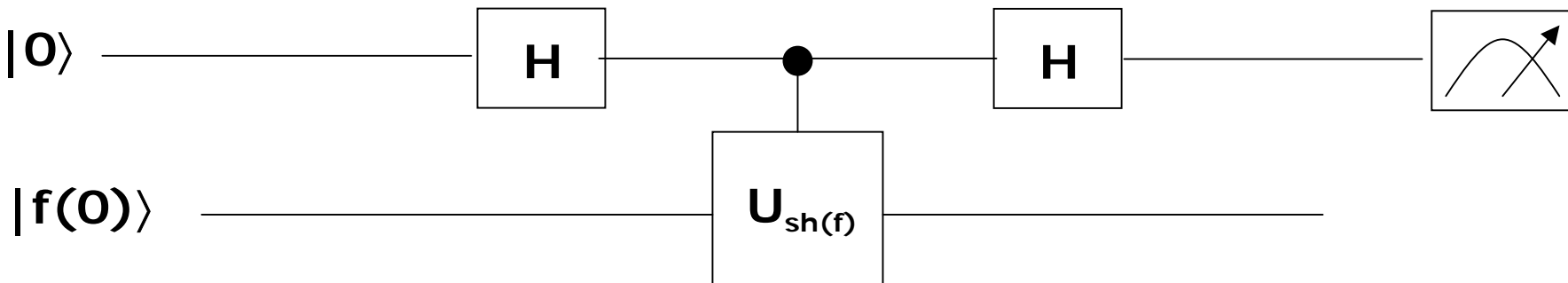
Controlled- $\oplus f(x)$ v. Controlled- $U_{sh(f)}$ (2)

Thus, the following network solves Deutsch's problem with probability 1/2



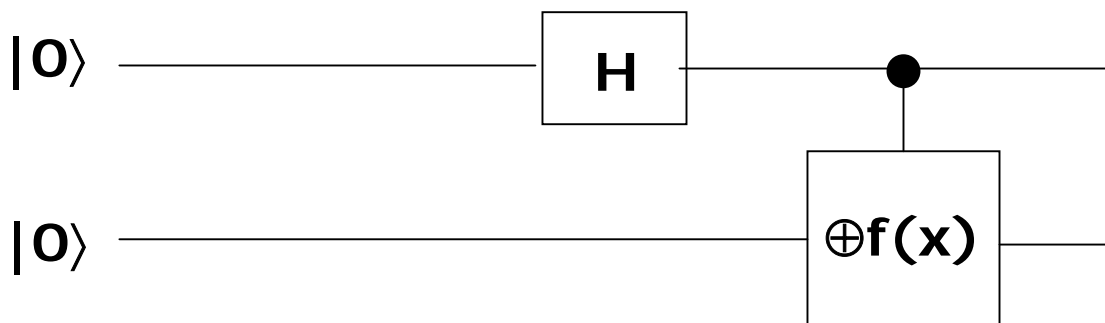
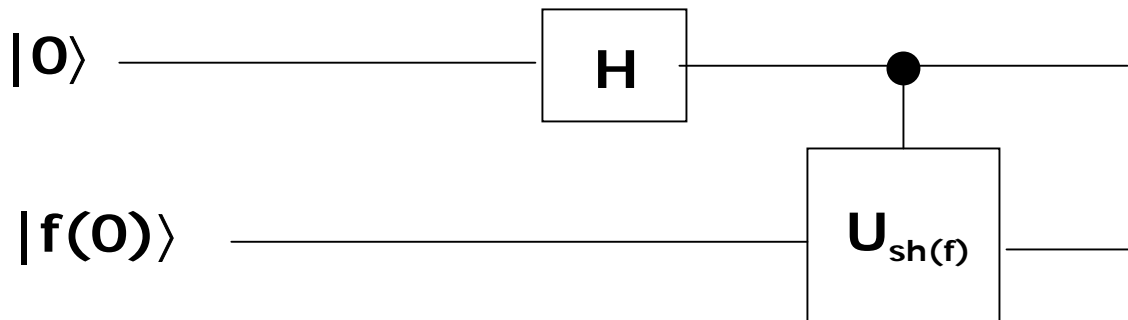
Suppose we were given the state $|f(0)\rangle$, which is a uniform superposition of the eigenvectors of $U_{sh(f)}$ (in general, as we'll see later!)

Then, the following network solves Deutsch's problem with probability 1/2



Controlled- $\oplus f(x)$ v. Controlled- $U_{sh(f)}$ (3)

Exercise: Compare the states produced by the following networks



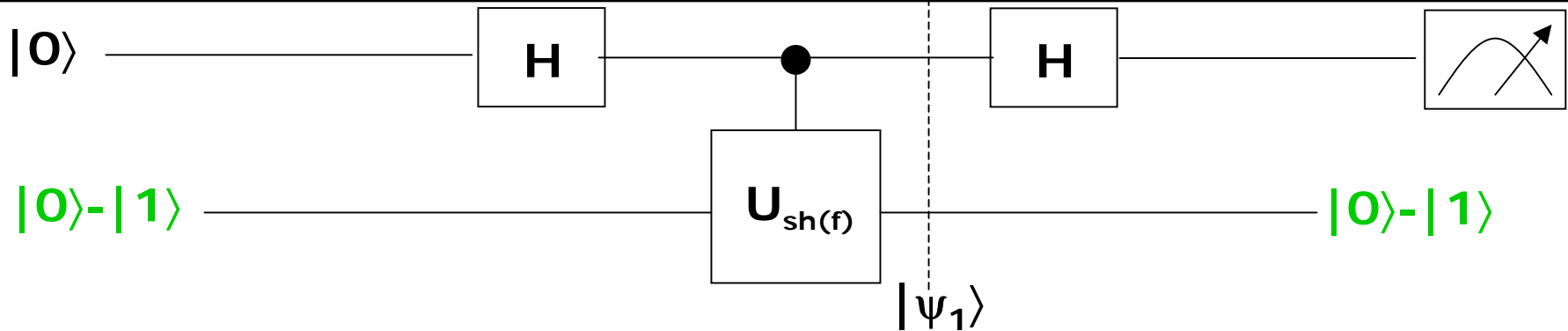
The equivalence of these two states is the fundamental link between the shift operator (eigenvalue-estimation approach to quantum algorithms) and the controlled- $\oplus f(x)$ operator (standard approach)

8-lecture Mini-course in Quantum Computation

Lecture 5

Lawrence Ioannou

Phase Estimation



$$|\psi_1\rangle = [|0\rangle + (-1)^{f(0)+f(1)} |1\rangle] (|0\rangle - |1\rangle)$$

Let's rewrite $|\psi_1\rangle$ in a more standardised phase notation:

$$|\psi_1\rangle = [|0\rangle + e^{2\pi i \omega} |1\rangle] (|0\rangle - |1\rangle)$$

where $\omega = f(0) \oplus f(1) / 2$

Applying the Hadamard gate to $|0\rangle + e^{2\pi i \omega} |1\rangle$ gives us $|f(0) \oplus f(1)\rangle$ which we will now think of as the most significant bit of the binary expansion of ω :

$$\omega = 0.a_1 a_2 a_3 \dots \equiv a_1 / 2 + a_2 / 4 + a_3 / 8 + \dots \quad (a_i \in \{0, 1\})$$

$$a_1 = f(0) \oplus f(1), \quad a_2 = 0, \quad a_3 = 0, \dots$$

Phase Estimation (2)

$$\omega = 0.a_1a_2a_3 \equiv a_1/2 + a_2/4 + a_3/8 \quad (a_i \in \{0,1\})$$

This procedure can be generalised to the case where ω has lower-order (nonzero) bits

E.g. suppose $\omega = 0.a_1a_2a_3$ i.e. $\omega = j/8$ for some integer j in $[0,1,\dots,7]$

Suppose we have prepared the tensor-product state

$$(|0\rangle + e^{2\pi i(4\omega)} |1\rangle) (|0\rangle + e^{2\pi i(2\omega)} |1\rangle) (|0\rangle + e^{2\pi i\omega} |1\rangle)$$

which, noting $e^{2\pi i} = 1$, we can also express as

$$(|0\rangle + e^{2\pi i(0.a_3)} |1\rangle) (|0\rangle + e^{2\pi i(0.a_2a_3)} |1\rangle) (|0\rangle + e^{2\pi i(0.a_1a_2a_3)} |1\rangle)$$

Our goal is to find a network that will map this state to

$$|a_3\rangle |a_2\rangle |a_1\rangle$$

so that measuring the qubits w.r.t. the computational basis will determine ω

Phase Estimation (3)

$$\omega = 0.a_1a_2a_3 \equiv a_1/2 + a_2/4 + a_3/8 \quad (a_i \in \{0,1\})$$

$$(|0\rangle + e^{2\pi i(0.a_3)}|1\rangle)(|0\rangle + e^{2\pi i(0.a_2a_3)}|1\rangle)(|0\rangle + e^{2\pi i(0.a_1a_2a_3)}|1\rangle) \rightarrow |a_3\rangle|a_2\rangle|a_1\rangle$$

(we'll use the facts that $H=H^{-1}$ and H maps

$$|b\rangle \rightarrow |0\rangle + (-1)^b|1\rangle = |0\rangle + e^{2\pi i(0.b)}|1\rangle)$$

We can map the first qubit to $|a_3\rangle$ by just applying the Hadamard gate, as we did in Deutsch's problem, to get

$$|a_3\rangle(|0\rangle + e^{2\pi i(0.a_2a_3)}|1\rangle)(|0\rangle + e^{2\pi i(0.a_1a_2a_3)}|1\rangle)$$

If the second qubit were in the state $|0\rangle + e^{2\pi i(0.a_2)}|1\rangle$, then we could just apply the Hadamard gate

Note

$$\begin{aligned} e^{2\pi i(0.a_2a_3)} &= e^{2\pi i(0.0a_3)}e^{2\pi i(0.a_2)} \\ &= e^{2\pi i(a_3/4)}e^{2\pi i(0.a_2)} \\ &= e^{i(a_3\pi/2)}e^{2\pi i(0.a_2)} \end{aligned}$$

If $a_3=0$, then the second qubit is in the desired state; else, if $a_3=1$, then applying a $(-\pi/2)$ -phase gate $R_{-\pi/2}$ to the second qubit will give the desired state

Phase Estimation (4)

$$\omega = 0.a_1a_2a_3 \equiv a_1/2 + a_2/4 + a_3/8 \quad (a_i \in \{0,1\})$$

$$(|0\rangle + e^{2\pi i(0.a_3)}|1\rangle)(|0\rangle + e^{2\pi i(0.a_2a_3)}|1\rangle)(|0\rangle + e^{2\pi i(0.a_1a_2a_3)}|1\rangle) \rightarrow |a_3\rangle|a_2\rangle|a_1\rangle$$

(we use the facts that $H=H^{-1}$ and H maps

$$|b\rangle \rightarrow |0\rangle + (-1)^b|1\rangle = |0\rangle + e^{2\pi i(0.b)}|1\rangle)$$

Because the first qubit is already in the state $|a_3\rangle$, we can apply a controlled- $R_{-\pi/2}$ gate, to effect the conditional phase-change

Applying this gate and then the Hadamard gate gives

$$|a_3\rangle|a_2\rangle(|0\rangle + e^{2\pi i(0.a_1a_2a_3)}|1\rangle)$$

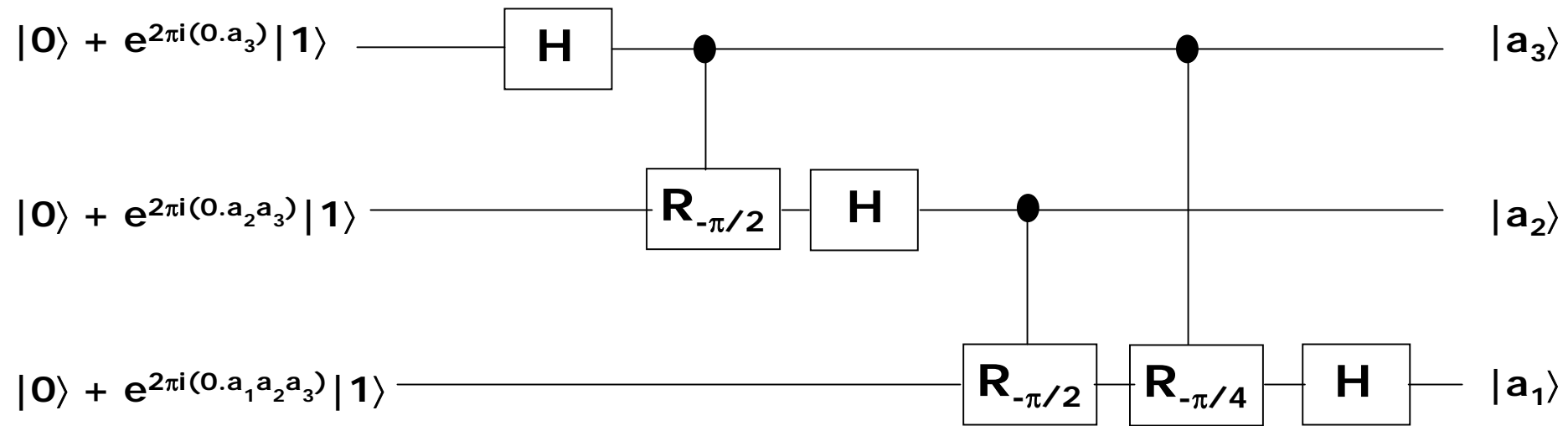
Similarly (exercise), applying the controlled- $R_{-\pi/2}$ gate (to get rid of the $e^{2\pi i(0.0a_2^0)}$) and then the controlled- $R_{-\pi/4}$ gate (to get rid of the $e^{2\pi i(0.00a_3)}$) gives $|a_3\rangle|a_2\rangle(|0\rangle + e^{2\pi i(0.a_1)}|1\rangle)$, which after a final Hadamard gate gives

$$|a_3\rangle|a_2\rangle|a_1\rangle$$

Phase Estimation (5)

$$\omega = 0.a_1a_2a_3\dots a_n \equiv a_1/2 + a_2/4 + \dots + a_n/2^n \quad (a_i \in \{0,1\})$$

The complete network looks like this



If $\omega = 0.a_1a_2a_3 \dots a_n$, then the obvious generalisation of this network maps

$$(|0\rangle + e^{2\pi i(2^{n-1}\omega)}|1\rangle)(|0\rangle + e^{2\pi i(2^{n-2}\omega)}|1\rangle)\dots(|0\rangle + e^{2\pi i(2\omega)}|1\rangle)(|0\rangle + e^{2\pi i\omega}|1\rangle) = \sum_{x=0}^{2^n-1} e^{2\pi i x \omega} |x\rangle$$

to

$$|a_n\rangle |a_{n-1}\rangle \dots |a_2\rangle |a_1\rangle$$

Phase Estimation (6)

(Definition of Quantum Fourier Transform)

$$\omega = 0.a_1a_2a_3\dots a_n \equiv a_1/2 + a_2/4 + \dots + a_n/2^n \quad (a_i \in \{0,1\})$$
$$(|0\rangle + e^{2\pi i(2^{n-1}\omega)}|1\rangle)(|0\rangle + e^{2\pi i(2^{n-2}\omega)}|1\rangle)\dots(|0\rangle + e^{2\pi i(2\omega)}|1\rangle)(|0\rangle + e^{2\pi i\omega}|1\rangle) = \sum_{x=0}^{2^n-1} e^{2\pi i x \omega} |x\rangle$$

For $\omega = 0.a_1a_2a_3\dots a_n = j/2^n$, for some j in $\{0,1,\dots,2^n-1\}$, the above state is an element of the *Fourier basis*

Apart from a final reversal of the order of the qubits, the network we described implements the *inverse quantum Fourier transform (QFT⁻¹)*

For any integer $M > 1$, the quantum Fourier transform, QFT(M), acts on the vector space generated by $|0\rangle, |1\rangle, \dots, |M-1\rangle$, and maps

$$|j\rangle \mapsto \sum_{x=0}^{M-1} e^{2\pi i \frac{j}{M} x} |x\rangle$$

Phase Estimation (7)

$$\text{QFT}^{-1}: \sum_{x=0}^{M-1} e^{2\pi i \omega x} |x\rangle \mapsto |j\rangle \quad \text{if } \omega = j/M \text{ for } j \text{ in } \{0, 1, \dots, M-1\}$$

What about when ω is not j/M for some integer j ?

It turns out that applying QFT^{-1} to

$$\sum_{x=0}^{M-1} e^{2\pi i \omega x} |x\rangle$$

and measuring w.r.t. the computational basis gives j such that $\text{Prob}(|j - M\omega| \leq 1) \geq 8/\pi^2$

That is, with high probability, QFT^{-1} gives the best estimator of ω

Note that increasing M increases the accuracy of the estimate

Eigenvalue Estimation

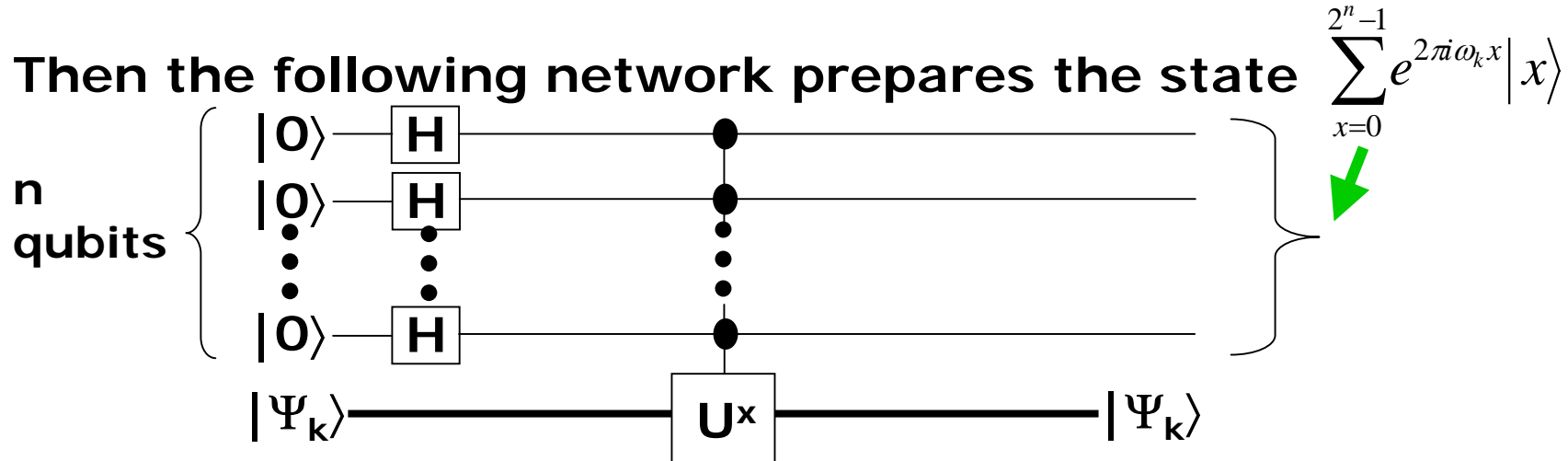
We can use eigenvalue kick-back and phase estimation (like we did in Deutsch's problem) to estimate an eigenvalue of a unitary operator U (in Deutsch's problem we estimated an eigenvalue of the shift operator)

Suppose U is such that

$$U|\Psi_k\rangle = e^{2\pi i\omega_k}|\Psi_k\rangle$$

Assume we are given $|\Psi_k\rangle$ and we can construct a controlled- U^x gate

Then the following network prepares the state



Applying $\text{QFT}(2^n)^{-1}$ to the upper n qubits and measuring w.r.t. the computational basis (and dividing result by 2^n) gives the best n -bit estimate of ω_k with high probability

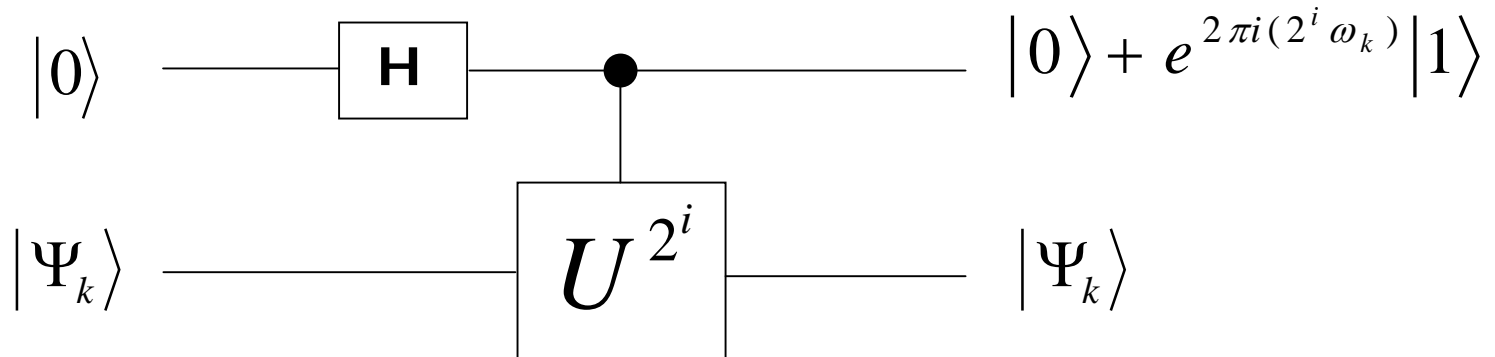
Eigenvalue Estimation (2)

Here is a way to effect the controlled- U^x gate, assuming we can construct the controlled- U^s gate for $s=2^i$ for any i

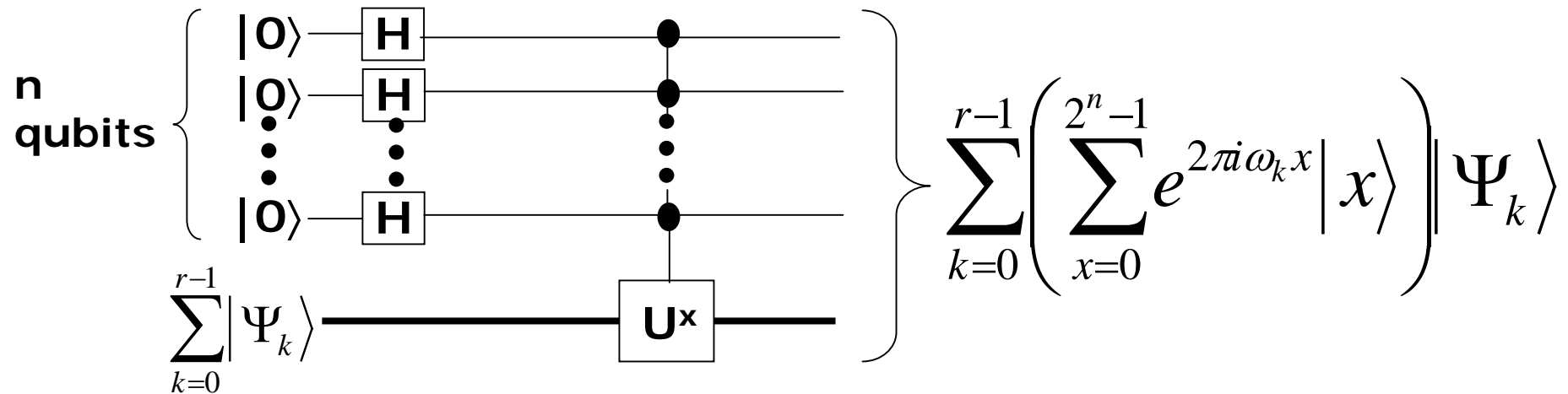
Recall

$$(|0\rangle + e^{2\pi i(2^{n-1}\omega)}|1\rangle)(|0\rangle + e^{2\pi i(2^{n-2}\omega)}|1\rangle)\cdots(|0\rangle + e^{2\pi i(2\omega)}|1\rangle)(|0\rangle + e^{2\pi i\omega}|1\rangle) = \sum_{x=0}^{2^n-1} e^{2\pi i x \omega} |x\rangle$$

We can build this state one qubit at a time like this



Eigenvalue Estimation (3)



Of course, if we input an equally-weighted superposition of r eigenvectors in the bottom register, then applying the inverse-QFT to the top n qubits and measuring will give an estimate of a particular ω_k with probability $1/r$

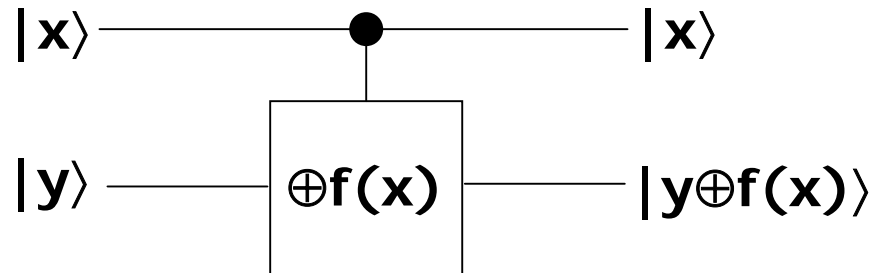
(Recall that we did this in Deutsch's problem)

Period-Finding

We can use the QFT to find the period of a function f such that
 $f: \mathbb{Z} \rightarrow \{0, 1, \dots, N-1\}$

$$f(x) = f(y) \Leftrightarrow x - y \in r\mathbb{Z}$$

if given the controlled- $\oplus f(x)$ gate



Assume a bound on r : $M > 2r^2$ (let $M = 2^n$, for convenience)

Period-finding algorithm (basic version):

1. Using controlled- $\oplus f(x)$ (or otherwise!) prepare $\sum_{x=0}^{2^n-1} |x\rangle |f(x)\rangle$
2. Apply $\text{QFT}(2^n)^{-1}$ to first register
3. Measure first register; result is j
4. (classical) Run continued-fractions algorithm on $j/2^n$ to get coprime integers k' and r' such that $k'/r' = k/r$ for some k in $\{0, 1, \dots, r-1\}$; test if r' is period of f ; if not, repeat

Period-Finding (2)

There are at least two ways to analyse the algorithm:

- (1) phase-estimation capability of QFT⁻¹ (due to Kitaev, Mosca et al.)
- (2) “classic” properties of QFT (due to Shor)

We’ll cover (1) first, since phase-estimation is fresh in our minds

Consider the shift operator $U_{sh(f)}$ that (perhaps magically) maps

$$|f(x)\rangle \rightarrow |f(x+1)\rangle$$

Exercise: verify that

$$U_{sh(f)}|\Psi_k\rangle = e^{2\pi i \frac{k}{r}}|\Psi_k\rangle$$

where

$$|\Psi_k\rangle = \sum_{j=0}^{r-1} e^{-2\pi i j \frac{k}{r}} |f(j)\rangle$$

for $k=0,1,\dots,r-1$

Period-Finding (3)

$$U_{sh(f)}|\Psi_k\rangle = e^{2\pi i \frac{k}{r}}|\Psi_k\rangle \quad |\Psi_k\rangle = \sum_{j=0}^{r-1} e^{-2\pi i j \frac{k}{r}}|f(j)\rangle \quad \text{for } k=0,1,\dots,r-1$$

Estimating a random eigenvalue of $U_{sh(f)}$ allows us to find r :

once a sufficiently accurate estimation of k/r (for some k) is in hand, we give it to the continued-fractions algorithm (described later)

To estimate a k/r , we know that in principle we would need:

1. to know how to construct controlled- $(U_{sh(f)})^x$
2. an equally-weighted superposition of $|\Psi_k\rangle$

However, amazingly, $|f(0)\rangle = \sum_{k=0}^{r-1} |\Psi_k\rangle$

So, *if* we could construct controlled- $(U_{sh(f)})^x$, then we *could* use it and $|f(0)\rangle$ in our eigenvalue estimation procedure, in which we would produce the state

$$\sum_{k=0}^{r-1} \left(\sum_{x=0}^{2^n-1} e^{2\pi i \frac{k}{r} x} |x\rangle \right) |\Psi_k\rangle \quad \dots$$

Period-Finding (4)

$$U_{sh(f)}|\Psi_k\rangle = e^{2\pi i \frac{k}{r}}|\Psi_k\rangle \quad |\Psi_k\rangle = \sum_{j=0}^{r-1} e^{-2\pi i j \frac{k}{r}}|f(j)\rangle \quad \text{for } k=0,1,\dots,r-1$$

$$|f(0)\rangle = \sum_{k=0}^{r-1} |\Psi_k\rangle$$

But clearly, applying the controlled- $(U_{sh(f)})^x$ to

$$\sum_{x=0}^{2^n-1} |x\rangle |f(0)\rangle$$

gives

$$\sum_{x=0}^{2^n-1} |x\rangle |f(x)\rangle = \sum_{k=0}^{r-1} \left(\sum_{x=0}^{2^n-1} e^{2\pi i \frac{k}{r} x} |x\rangle \right) |\Psi_k\rangle$$

Thus, we can make the “eigenvalue-estimation state”

$$\sum_{k=0}^{r-1} \left(\sum_{x=0}^{2^n-1} e^{2\pi i \frac{k}{r} x} |x\rangle \right) |\Psi_k\rangle$$

by just using the given controlled- $\oplus f(x)$ gate

Measuring the first register of the above state will give us an estimate $x/2^n$ of k/r

Period-Finding (5)

we have an estimate x/M of k/r , where $M > 2r^2$ and $M = 2^n$

Every real number y has a sequence of rationals, called *convergents*, that approximate it

Lemma: Given the integers x and M , if

$$\left| \frac{k}{r} - \frac{x}{M} \right| \leq \frac{1}{M} < \frac{1}{2r^2}$$

then the fraction k/r is a convergent of x/M

With high probability, we know our x/M satisfies the first inequality; the second inequality is satisfied by assumption

The *continued fractions* algorithm (see next slide) is used to compute the convergents of x/M , and finds coprime integers k' and r' such that $k'/r' = k/r$; we test r' to see if it is r

k and r are coprime with reasonably high probability, so the given period-finding algorithm terminates quickly (there are more sophisticated ways to increase success probability)

Period-Finding (6)

Define $[a_0, \dots, a_L] \equiv a_0 + \frac{1}{a_1 + \frac{1}{a_2 + \frac{1}{\ddots + \frac{1}{a_L}}}}$

where the a_i are positive integers (except a_0 may be 0)

Define the m^{th} convergent ($0 \leq m \leq L$) to this continued fraction to be $[a_0, \dots, a_m]$

We give x/M to the continued fractions algorithm, and it computes the convergents of x/M

This is an efficient computation (terminates quickly)

Period-Finding (7)

Applying the continued fractions algorithm to 31/13

$$\frac{31}{13} = 2 + \frac{5}{13} = 2 + \frac{1}{\frac{13}{5}}$$

Notice the “split-and-invert”
nature of the algorithm

$$= 2 + \frac{1}{2 + \frac{3}{5}} = 2 + \frac{1}{2 + \frac{1}{\frac{5}{3}}}$$

(Example taken from Nielsen
and Chuang)

$$= 2 + \frac{1}{2 + \frac{1}{1 + \frac{2}{3}}} = 2 + \frac{1}{2 + \frac{1}{1 + \frac{1}{\frac{3}{2}}}}$$

$$= 2 + \frac{1}{2 + \frac{1}{1 + \frac{1}{1 + \frac{1}{2}}}}$$

 **STOP** when you get a 1 in the
numerator after splitting

Period-Finding (8) (a la Shor)

Let us briefly look at Shor's original analysis of the period-finding algorithm

First let's modify the algorithm to be like Shor's original algorithm, using $\text{QFT}(2^n)$ instead of $\text{QFT}(2^n)^{-1}$:

Quantum part of period-finding algorithm (basic version):

1. Using controlled- $\oplus f(x)$ (or otherwise!) prepare $\sum_{x=0}^{2^n-1} |x\rangle |f(x)\rangle$
2. Apply $\text{QFT}(2^n)$ to first register
3. Measure first register; result is j

So as not to panic, note that $\text{QFT} = \text{QFT}^{-1} \text{QFT}^2$, so we can think of this revised period-finding algorithm as the previous one, but with an extra QFT^2 after the QFT^{-1}

Exercise: show that $\text{QFT}(M)^2$ is a fairly benign operation mapping $|j\rangle \rightarrow |M-j \bmod M\rangle$

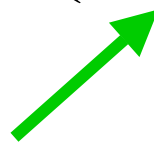
Period-Finding (9) (a la Shor)

The Fourier transform is well known in mathematics and physics to pick out the frequency of a periodic function

Similarly, the QFT would map a periodic superposition of computational basis states to a state that had most probability amplitude in basis states corresponding to the period

For convenience, assume $2^n = tr$ for some integer t (obviously this would rarely be the case; when it's not, as long as $2^n/r$ is sufficiently big, the end result will be approximately the same...)

$$\sum_{x=0}^{2^n-1} |x\rangle |f(x)\rangle = \sum_{l=0}^{r-1} \left(\sum_{j=0}^{t-1} |l + jr\rangle \right) |f(l)\rangle$$



periodic superposition!

Period-Finding (10) (a la Shor)

For convenience, assume $2^n = tr$ for some integer t

To simplify notation, assume we've measured the second register, thus leaving the first register in the state

$$\sum_{j=0}^{t-1} |l + jr\rangle \quad (\text{for some } l)$$

(note this measurement is optional; it does not change the algorithm)

Applying QFT(2^n) now gives

$$\sum_{x=0}^{2^n-1} e^{2\pi i l x / 2^n} \left(\sum_{j=0}^{t-1} e^{2\pi i j r \frac{x}{2^n}} \right) |x\rangle$$

When $x/2^n = k/r$, the sum inside the brackets is equal to t ; otherwise, the sum is 0 (if $2^n \neq tr$, the sums are approximately t and 0)

Thus measuring this state w.r.t. the computational basis gives an estimate of k/r for some k (just like the eigenvalue-estimation analysis did)

Order-Finding

For positive coprime integers N and y , the *order* of y modulo N is defined to be the least positive integer r such that

$$y^r \equiv 1 \pmod{N}$$

The INTEGER FACTORING problem reduces to the order-finding problem

We can use the period-finding algorithm to solve the order-finding problem, noting that the order of y is the period of the function

$$f(x) = y^x$$

Note that for this particular $f(x)$ we *do* know how to implement the shift operator: just multiply $f(x)$ by y to get $f(x+1)$

8-lecture Mini-course in Quantum Computation

Lecture 6

Lawrence Ioannou

Reduction of Factoring to Order-Finding

The (probabilistic) reduction-algorithm is as follows:

FindFactor(N)

1. If N is even, return factor 2
2. Determine if $N=c^b$; if so, return factor c
3. Randomly choose y from Z_N^* ; if $\gcd(y,N) > 1$, return factor $\gcd(y,N)$
4. Find the order r of y modulo N
5. If r is even, compute $d=\gcd(y^{r/2}-1,N)$, return d if $d \neq 1$; else go to step 3

Reduction of Factoring to Order-Finding (2)

FindFactor(N)

3. Randomly choose y from Z_N^* ; if $\gcd(y, N) > 1$, return factor $\gcd(y, N)$

4. Find the order r of y modulo N

5. If r is even, compute $d = \gcd(y^{r/2} - 1, N)$, return d if $d \neq 1$; else go to step 3

We give a proof for when $N = pq$, for p and q prime:

Let y_1 and y_2 be generators of Z_p^* and Z_q^* respectively

Note that randomly choosing y from Z_N^* is *equivalent* to randomly choosing x_1 from Z_p^* and x_2 from Z_q^* , and then computing y as the (unique) solution to the congruence system

$$y \equiv y_1^{x_1} \pmod{p}$$

$$y \equiv y_2^{x_2} \pmod{q}$$

(using the Chinese Remainder Theorem)

The order of y modulo p is $r_1 = (p-1)/\gcd(x_1, p-1)$

The order of y modulo q is $r_2 = (q-1)/\gcd(x_2, q-1)$

The order of y modulo N is $r = \text{lcm}(r_1, r_2) = r_1 r_2 / \gcd(r_1, r_2)$

Reduction of Factoring to Order-Finding (3)

$$d = \gcd(y^{r/2} - 1, N), \quad r \text{ even, } r = \text{order of } y \text{ mod } N$$

$y^{r/2} \not\equiv 1 \pmod{N}$ (or else r is not the order of y)

But since $y^r = 1 \pmod{N} \Leftrightarrow (y^{r/2} - 1)(y^{r/2} + 1) = 0 \pmod{N}$

Then *either* $y^{r/2} \equiv -1 \pmod{N}$ *or* d is a nontrivial factor of N

Let $r_1 = 2^{c_1}(\text{odd}_1)$
 $r_2 = 2^{c_2}(\text{odd}_2)$

If $c_1 \neq c_2$, then $r = \text{lcm}(r_1, r_2) = r_1 r_2 / \gcd(r_1, r_2)$

$$= \begin{cases} 2r_2(\text{int}) & \text{if } c_1 > c_2 \\ 2r_1(\text{int}) & \text{if } c_1 < c_2 \end{cases}$$

$$\Rightarrow \begin{cases} d = p \\ d = q \end{cases}$$

If $c_1 = c_2$, then $r = r_1(\text{odd}_1) = r_2(\text{odd}_2)$

$$\Rightarrow y^{r/2} \equiv -1 \pmod{p} \text{ and } y^{r/2} \equiv -1 \pmod{q}$$

$$\Rightarrow y^{r/2} \equiv -1 \pmod{N}$$

Reduction of Factoring to Order-Finding (4)

$$r_1 = 2^{c_1}(\text{odd}_1)$$

$$r_2 = 2^{c_2}(\text{odd}_2)$$

y_1 and y_2 are generators of Z_p^* and Z_q^* respectively

Thus $c_1 \neq c_2$ is success and $c_1 = c_2$ is failure

What's the probability that $c_1 \neq c_2$?

Suppose $p-1 = 2^{k_1}s_1$ and $q-1 = 2^{k_2}s_2$

We only explain the worst case $k_1 = k_2 = 1$ (y_1, y_2 have order $2s$):

The even powers $y_i^{x_i}$ of y_i have odd order r_i

Whereas the odd powers have order $2(\text{odd}_i)$

Thus if x_i are chosen randomly, $\text{Prob}(c_1 \neq c_2) = 1/2$ (because there are just as many even x_i as there are odd x_i)

Breaking the RSA Cryptosystem

Old-school cryptosystems used a *symmetric* or *secret key*

Two parties (Alice and Bob) wishing to communicate secretly would first (somehow) decide on a random binary string (the secret key)

Before *quantum key distribution*, this was very impractical

In the 1970s, people had the idea of using complexity-theoretic *assumptions* to build *computationally-secure* cryptosystems (like the RSA cryptosystem)

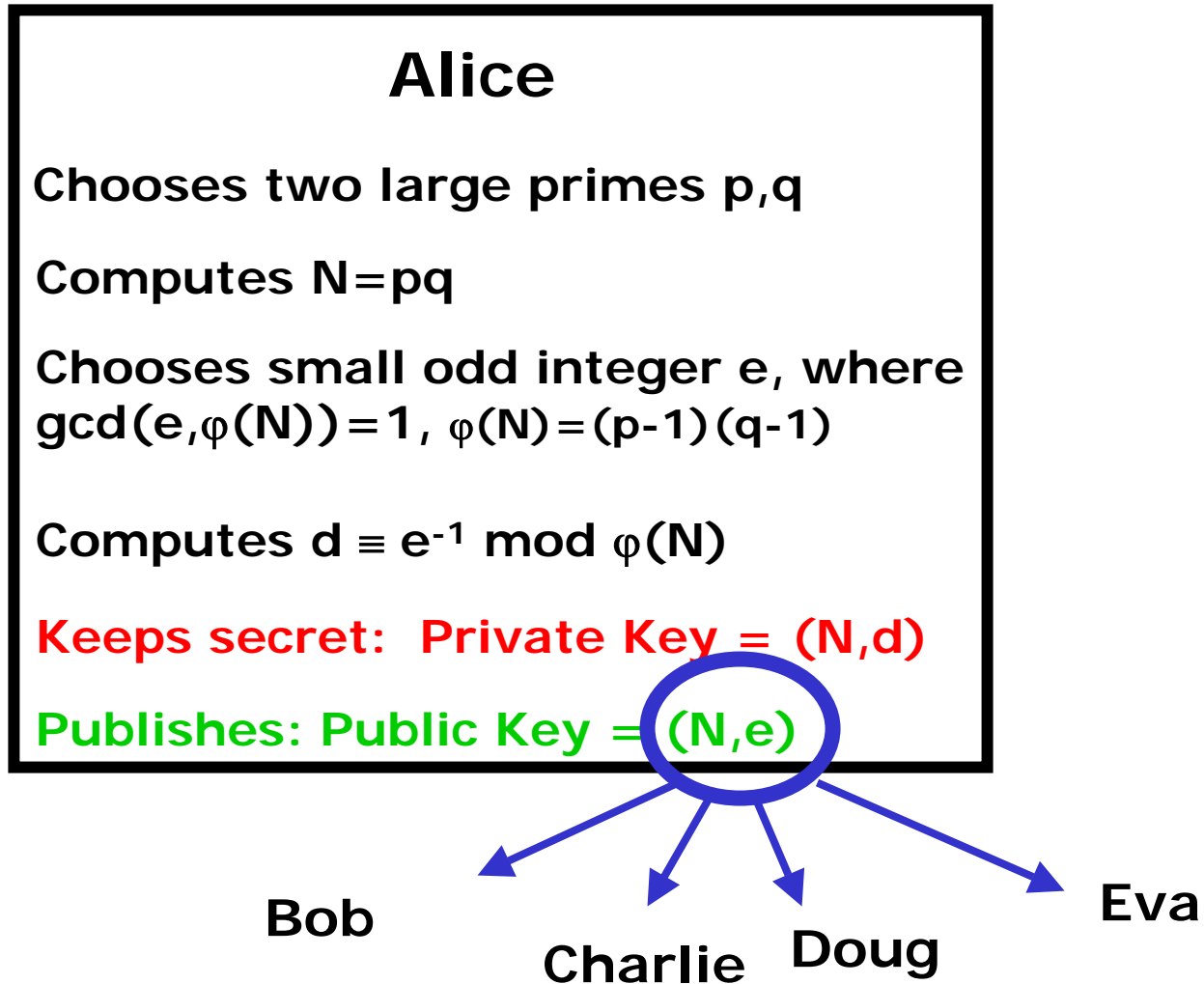
These cryptosystems use a *public key* (chosen by Alice), allowing *anyone* to send Alice a secret message without ever sharing a secret in the first place – *the public key is good for many messages*

Of course, given enough time, the public key can be used to break the cryptosystem

The assumption of the RSA cryptosystem is that the integer-factorisation problem is very difficult (not in BPP)

Breaking the RSA Cryptosystem (2)

The RSA cryptosystem (for anyone to send secret messages to Alice):



Clearly, knowing the factors p, q of N allows anyone to compute the private key d

Breaking the RSA Cryptosystem (3)

For completeness, we give the encryption and decryption algorithms:

To encrypt and send a message $m \in \{0, 1, \dots, N-1\}$, Bob computes and sends the number $c = m^e \pmod{N}$

To decrypt the received ciphertext c , Alice computes

$$\begin{aligned} c^d \pmod{N} &= m^{ed} \pmod{N} \\ &= m^{1+k\phi(N)} \pmod{N} \end{aligned}$$

If $\gcd(m, N) = 1$, then Euler's generalisation of Fermat's Little Theorem gives $m^{k\phi(N)} \pmod{N} = 1$, and thus the decryption is successful

Exercise: When m and $N = pq$ have a common factor, then $c^d \pmod{N}$ still equals m (use the Chinese Remainder Theorem)

Exercise: Show how to compute m from c using an order-finding algorithm *without factoring* N

Discrete Logarithm Problem

Not all public-key cryptosystems are based on the classical difficulty of factoring

The *El Gamal* public-key cryptosystem is based on the classical difficulty of the discrete logarithm problem (DLP):

Let p be a prime, $\alpha \in \mathbb{Z}_p^*$ a generator, and $\beta \in \mathbb{Z}_p^*$.

Find unique integer s such that $\beta \equiv \alpha^s \pmod{p}$

Classically, this problem is thought to be hard for well-chosen p (p should have at least 150 digits and $p-1$ should have at least one large prime)

To set up the El Gamal cryptosystem, Alice chooses well the above parameters, publishes the **public key** (p, α, β) but keeps the **private key** s secret

If Bob wants to send a message m , he chooses a random $k \in \mathbb{Z}_{p-1}$, and sends to Alice $(y_1, y_2) = (\alpha^k \pmod{p}, m\beta^k \pmod{p})$

To decrypt, Alice computes $m = y_2(y_1^s)^{-1} \pmod{p}$ (Exercise: verify)

Discrete Logarithm Problem (2)

We can define the DLP more generally, in any group:

Given elements α and $\beta = \alpha^s$, $s \in \{0, 1, \dots, r-1\}$
(r is the order of α), from the group H ,
find s

Note that we can reduce the DLP to period-finding:

The function

$$f(x_1, x_2) = \alpha^{sx_1 + x_2} = \alpha^{x_2} \beta^{x_1}$$

is periodic, with period $(1, -s)$

From the period, we can calculate the discrete logarithm s ,
and thus we can break the El Gamal cryptosystem with a
quantum computer

To do this, we use a straightforward modification of the
period-finding algorithm, noting that the domain of f is now 2-
dimensional...

Discrete Logarithm Problem (3)

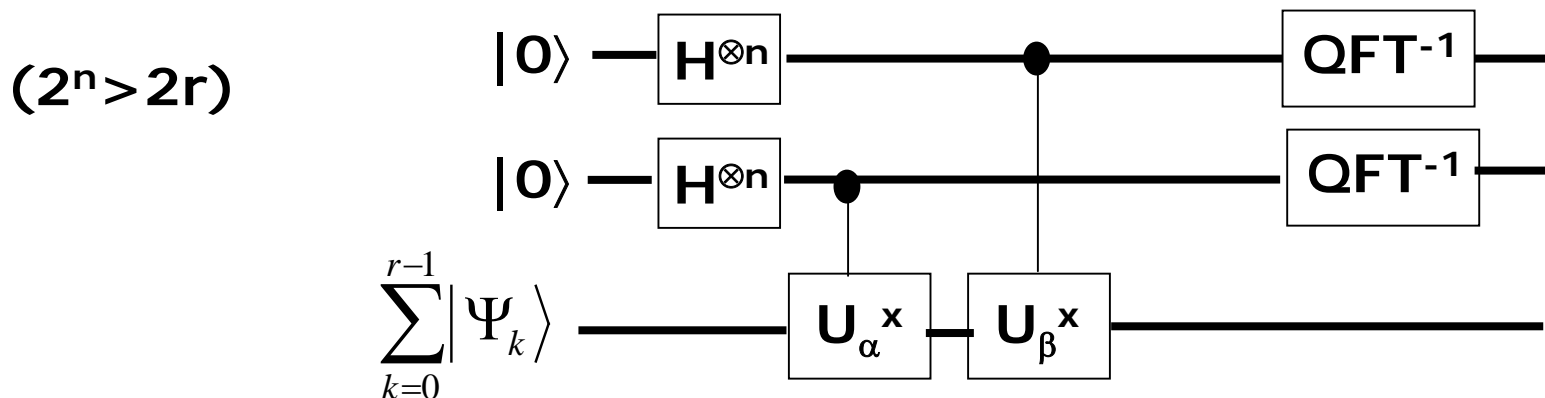
Let p be a prime, $\alpha \in \mathbb{Z}_p^*$ a generator, and $\beta \in \mathbb{Z}_p^*$.

Find unique integer s such that $\beta \equiv \alpha^s \pmod{p}$

The algorithm can be thought of as estimating eigenvalues $e^{2\pi i k/r}$ and $e^{2\pi i ks/r}$ of the "shift" operators $U_\alpha: |y\rangle \rightarrow |\alpha y\rangle$ and $U_\beta: |y\rangle \rightarrow |\beta y\rangle$, where r is the order modulo p of α (which we can compute using the order-finding algorithm)

Since we know r , we just need to estimate the eigenvalues with an error of at most $1/2r$ to find the numerators k and ks (we don't need the continued-fractions algorithm)

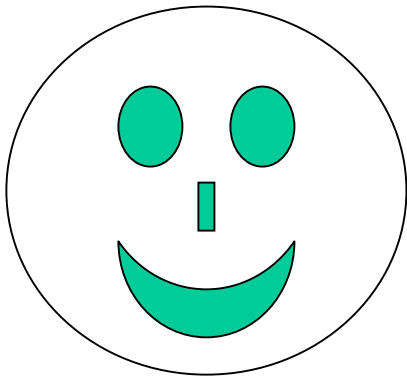
Finally, from k and ks , s is calculated as $s = k^{-1}(ks) \pmod{r}$



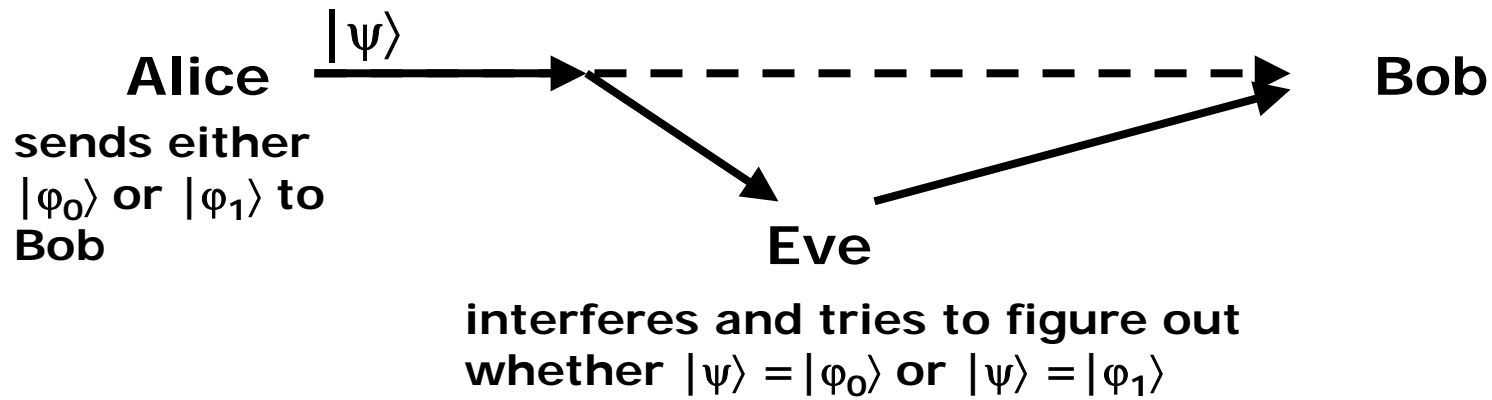
Quantum Cryptography

We've seen that exploiting quantum mechanics allows us to crack some (but not all!) classical **public-key** cryptosystems

But it also allows us to build new, *unconditionally-secure* cryptographic protocols



Information-disturbance tradeoff



Suppose eavesdropper Eve is trying to distinguish the two *non-orthogonal* states $|\varphi_0\rangle$ and $|\varphi_1\rangle$

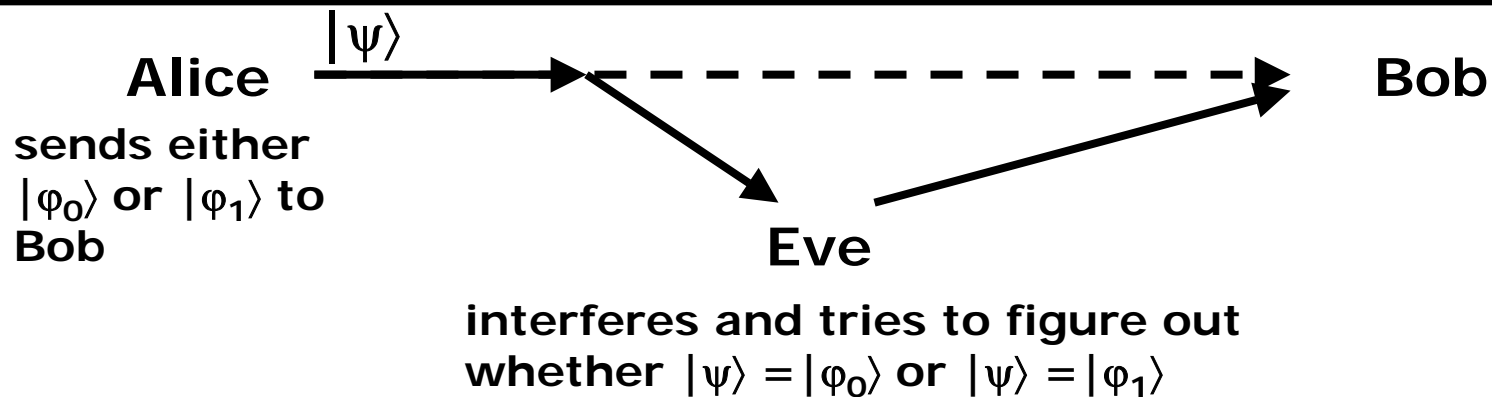
Without loss of generality, Eve uses a unitary operator U acting on the state $|\psi\rangle$ and an ancillary system prepared in a standard state $|0\rangle$

By way of contradiction, assume Eve's actions do not disturb the state (sent by Alice):

$$U: |\varphi_0\rangle|0\rangle \rightarrow |\varphi_0\rangle|v\rangle$$

$$|\varphi_1\rangle|0\rangle \rightarrow |\varphi_1\rangle|v'\rangle$$

Information-disturbance tradeoff (2)



$$\begin{aligned} U: |\varphi_0\rangle|0\rangle &\rightarrow |\varphi_0\rangle|v\rangle \\ |\varphi_1\rangle|0\rangle &\rightarrow |\varphi_1\rangle|v'\rangle \end{aligned}$$

Eve would like to measure the second register, in order to identify (or at least get some information about) the state in the first register

Thus she needs $|v\rangle$ and $|v'\rangle$ to be different

But the unitarity of U (along with the non-orthogonality of $|\varphi_0\rangle$ and $|\varphi_1\rangle$) implies that $|v\rangle$ and $|v'\rangle$ are identical

Thus Eve can't distinguish the states, unless she disturbs at least one of them

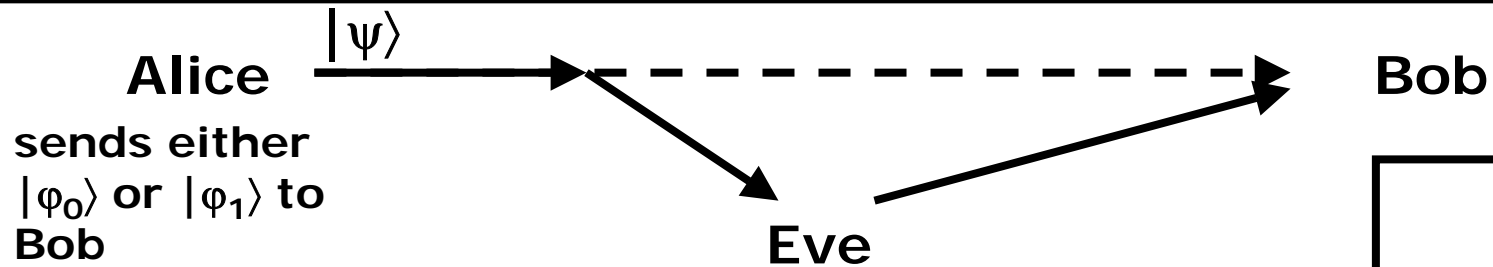
Information-disturbance tradeoff (3)

Proposition [12.18 in textbook]: In any attempt to distinguish between two non-orthogonal quantum states, information gain is only possible at the expense of introducing disturbance to the signal

The above theorem is at the heart of quantum key distribution (QKD)

Alice can send quantum states to Bob, and an eavesdropper will not learn much about the states unless she disturbs them – in which case (one can show that) Alice and Bob could detect Eve's meddling

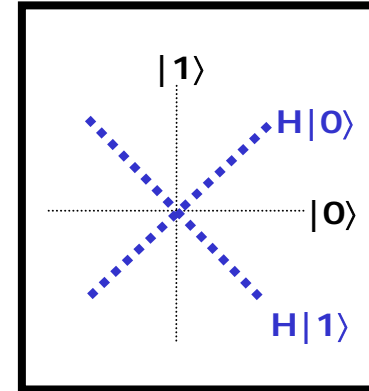
Quantum Key Distribution (QKD)



Define

$$|\varphi_0\rangle := |0\rangle$$

$$|\varphi_1\rangle := (|0\rangle + |1\rangle) / \sqrt{2} = H|0\rangle$$



For each secret bit, $a \in \{0, 1\}$, chosen by Alice:

1. Alice sends $|\psi\rangle = |\varphi_a\rangle$ to Bob

2. Bob chooses random bit, $a' \in \{0, 1\}$, and measures $|\psi\rangle$ w.r.t. $\{H^{a'}|0\rangle, H^{a'}|1\rangle\}$, obtaining result $b \in \{0, 1\}$ (w.r.t. basis ordering)

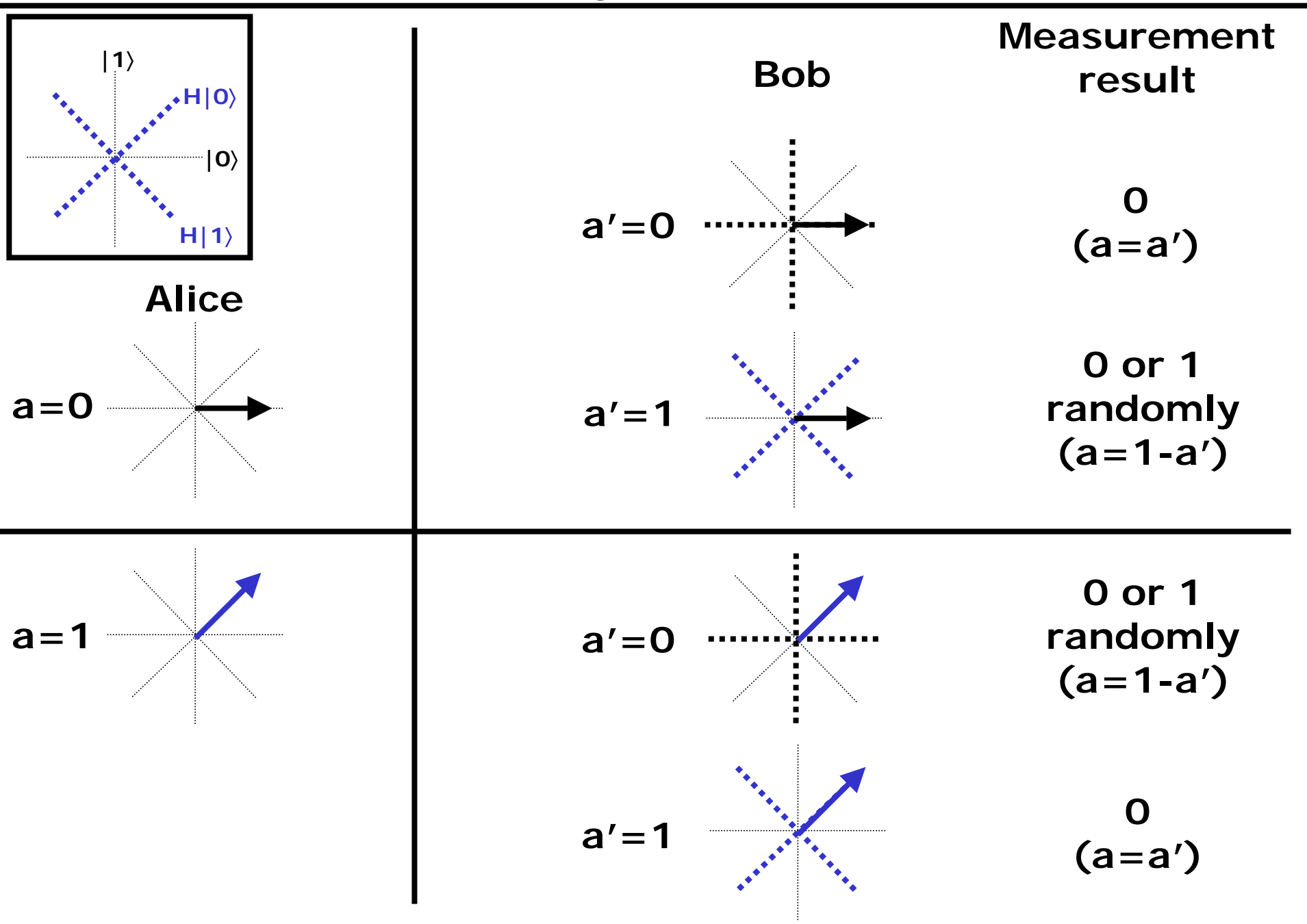
3. Bob publicly announces b , but keeps a' secret; Alice and Bob keep a and a' only if $b=1$ (else discard this a and a')

Note: If $a=a'$, then $b=0$; else if $a=1-a'$, then b is randomly 0 or 1. Thus about $3/4$ of the $\{a, a'\}$ are discarded

4. The final shared key bit is: a (for Alice) and $1-a'$ (for Bob)

Alice and Bob then (publicly) choose a sufficiently large proportion of the bits to check to see if they are the same; they abort if too many bits are different.

Quantum Key Distribution (2)



Quantum Key Distribution (3)

The protocol based on the previous two slides is known as *B92*, after Bennett, who discovered this streamlining of his original protocol known as *BB84* (after Bennett and Brassard)

Note we have not shown that the protocol is secure (or even described the whole protocol), but it should be clear that Eve learns nothing useful from the public discussion alone. A security proof would involve showing that if Eve did try to measure the states $|\psi\rangle = |\varphi_a\rangle$, she would disturb them such that Alice's and Bob's final shared key would fail the check at the end

There is another, fundamentally different type of protocol, called E91, discovered by Ekert; it uses EPR pairs

$$|\beta_{00}\rangle = |00\rangle + |11\rangle$$

Quantum Key Distribution (4)

Define

$$\begin{aligned} |+\rangle &= H|0\rangle \\ |-\rangle &= H|1\rangle \end{aligned}$$

Note (ignoring normalisation factors)

$$|0\rangle|0\rangle + |1\rangle|1\rangle = |+\rangle|+\rangle + |-\rangle|-\rangle$$

If Alice and Bob share this state (Alice has the left qubit and Bob has the right qubit), and they both measure their qubits with respect to the same basis, either $\{|0\rangle, |1\rangle\}$ or $\{|+\rangle, |-\rangle\}$, then they must get the same result (0 or 1 as in the B92 protocol description)

Exercise: Derive the E91 protocol (to the same level of detail that we described B92; assuming the “shared EPR pairs” are perfect and perfectly shared, you don’t need to have a test of the shared key bits at the end; in the full E91 protocol, Alice and Bob can test (using the public channel) the fidelity of their EPR pairs at the beginning of the protocol to ensure they are sharing actual EPR pairs to a sufficiently high fidelity)

Note that all the QKD protocols rely on the existence of an *authenticated public channel*

8-lecture Mini-course in Quantum Computation

Lecture 7

Lawrence Ioannou

Unordered-Search Problem

Given black box for $f: \{0,1\}^n \rightarrow \{0,1\}$

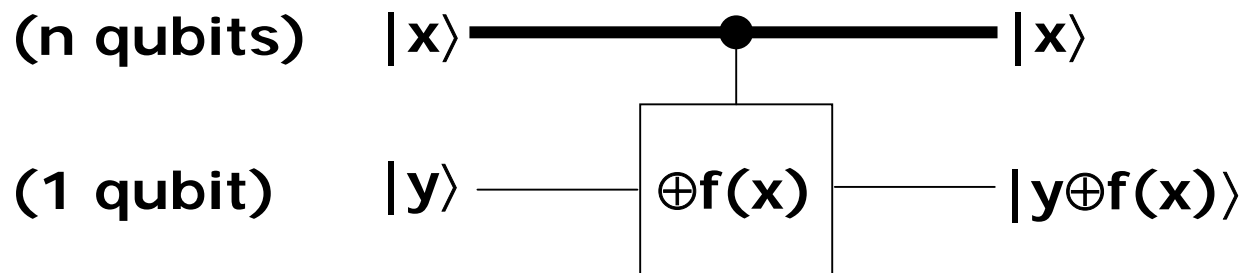
Find an x such that $f(x)=1$ (we are promised such x exists)

(think of $\{0,1\}^n$ as a “database”, perhaps...)

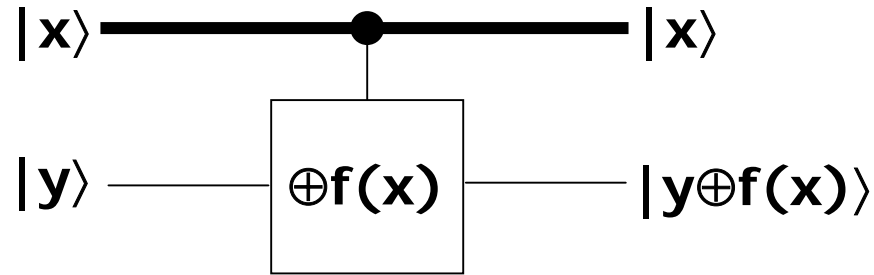
The best classical algorithm for unordered search requires roughly $N=2^n$ queries of f

The best quantum algorithm (due to Lov Grover) for unordered search requires roughly $N^{1/2}=2^{n/2}$ queries of f (we won't show optimality today)

We assume we are given a quantum black box

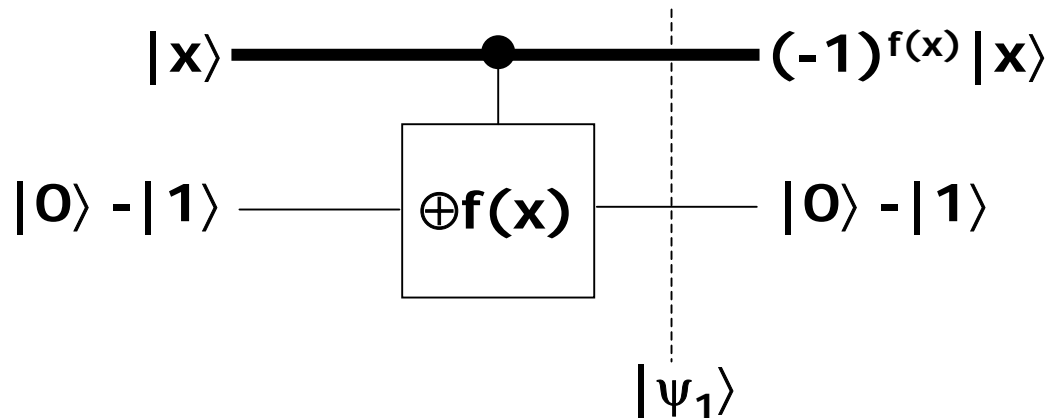


Unordered-Search Problem (2)



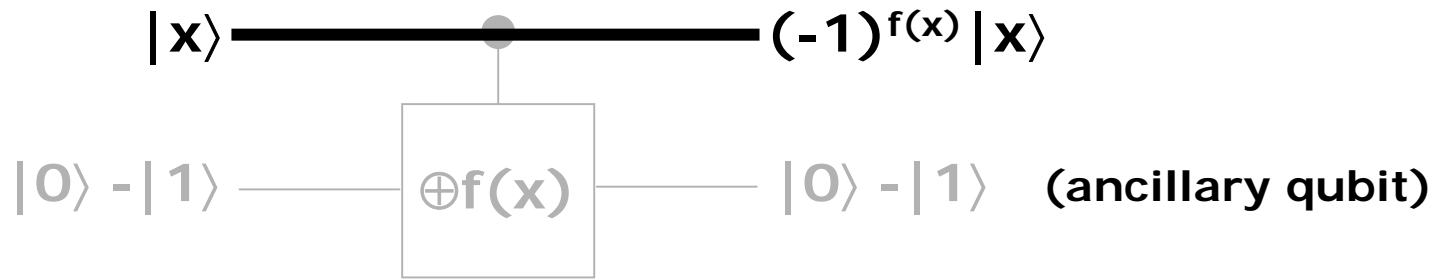
What shall we do with this black box?

Inspired by Deutsch's algorithm, we notice that



$$\begin{aligned}
 |\psi_1\rangle &= |x\rangle \otimes (|f(x)\rangle - |1 \oplus f(x)\rangle) \\
 &= \begin{cases} |x\rangle \otimes (|0\rangle - |1\rangle) & \text{if } f(x) = 0 \\ |x\rangle \otimes (|1\rangle - |0\rangle) & \text{if } f(x) = 1 \end{cases} \\
 &= (-1)^{f(x)} |x\rangle \otimes (|0\rangle - |1\rangle)
 \end{aligned}$$

Unordered-Search Problem (3)



So we use the black box to implement an operator U_f mapping

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

Recall we can use the n -qubit Hadamard gate, $H^{\otimes n}$, to make the equally-weighted superposition of all computational basis states:

$$\begin{array}{l}
 \left. \begin{array}{l}
 |0\rangle \\
 |0\rangle \\
 \vdots \\
 |0\rangle
 \end{array} \right\} \begin{array}{l}
 \text{n} \\
 \text{qubits}
 \end{array}
 \end{array}
 \left\{ \begin{array}{l}
 \begin{array}{c}
 \boxed{H} \\
 \boxed{H} \\
 \vdots \\
 \boxed{H}
 \end{array}
 \end{array} \right.
 \left. \begin{array}{l}
 |0\rangle + |1\rangle \\
 |0\rangle + |1\rangle \\
 \vdots \\
 |0\rangle + |1\rangle
 \end{array} \right\}
 \left(|0\rangle + |1\rangle \right)^{\otimes n} = \sum_{x \in \{0,1\}^n} |x\rangle$$

Let $|Y\rangle$ be this state:

$$|Y\rangle \equiv H^{\otimes n} |0\rangle = \sum_{x \in \{0,1\}^n} |x\rangle \equiv \sum_{i=0}^{2^n-1} |i\rangle$$

Recall there are two summation conventions: we can label the computational basis states by binary strings (x) or the corresponding integers (i)

Unordered-Search Problem (4)

$$|Y\rangle \equiv H^{\otimes n}|0\rangle = \sum_{i=0}^{2^n-1} |i\rangle$$

Note $|Y\rangle$ is just a basis state of another basis called the *Hadamard basis* which is

$$\{H^{\otimes n}|x\rangle : x \in \{0,1\}^n\}$$

(i.e. the Hadamard basis is the basis derived from applying the Hadamard gate to the elements of the computational basis)

Let $|x^*\rangle \equiv H^{\otimes n}|x\rangle$ for $x \in \{0,1\}^n$ denote an element in the Hadamard basis (similarly $|i^*\rangle \equiv H^{\otimes n}|i\rangle$ for $i \in \{0,1,\dots,2^n-1\}$)

$$\text{Computational basis} = \{ |x\rangle : x \in \{0,1\}^n \}$$

$$\text{Hadamard basis} = \{ |x^*\rangle : x \in \{0,1\}^n \}$$

Note $|Y\rangle = |0^*\rangle$

Unordered-Search Problem (5)

$$|Y\rangle \equiv H^{\otimes n}|0\rangle = \sum_{i=0}^{2^n-1} |i\rangle = |0^*\rangle$$

Computational basis = $\{|x\rangle : x \in \{0,1\}^n\} = \{|i\rangle : i=0,1,\dots,2^n-1\}$

Hadamard basis = $\{|x^*\rangle : x \in \{0,1\}^n\} = \{|i^*\rangle : i=0,1,\dots,2^n-1\}$

Introduce the operator U_0 that maps

$$U_0: |0\rangle \rightarrow -|0\rangle$$

$$|i\rangle \rightarrow |i\rangle \text{ for } i \neq 0$$

Consider the operator $U_Y \equiv H^{\otimes n}U_0H^{\otimes n}$, which does the analogous operation in the Hadamard basis:

$$U_Y|Y\rangle = -|Y\rangle$$

$$U_Y|i^*\rangle = |i^*\rangle \text{ for } i \neq 0$$

since $U_Y|i^*\rangle = H^{\otimes n}U_0H^{\otimes n}H^{\otimes n}|i\rangle$
 $= H^{\otimes n}U_0|i\rangle$, because $H^{\otimes n}$ is its own inverse

Unordered-Search Problem (6)

But what does the operator $U_Y \equiv H^{\otimes n} U_0 H^{\otimes n}$ do in the computational basis?

Recall that the (vector) *projection* of a unit-vector $|v\rangle$ onto a unit-vector $|u\rangle$ is given by

$$(|u\rangle, |v\rangle) |u\rangle$$

where $(|u\rangle, |v\rangle)$ is the inner product of the two vectors (like dot-product in a Euclidean space)

We will write $(|u\rangle, |v\rangle)$ as $\langle u|v\rangle$ (this is actually quantum-mechanical shorthand for the product $\langle u| |v\rangle$ where $\langle u|$ is the *dual-vector* of $|u\rangle$)

Let $P_{|u\rangle}$ be the (linear) projection operator, projecting onto $|u\rangle$

$$P_{|u\rangle} |v\rangle = \langle u|v\rangle |u\rangle$$

Unordered-Search Problem (7)

Action of $U_Y \equiv H^{\otimes n} U_0 H^{\otimes n}$ in the computational basis

$$P_{|u\rangle} |v\rangle = \langle u|v\rangle |u\rangle$$

$$\begin{aligned} U_Y &= H^{\otimes n} U_0 H^{\otimes n} = H^{\otimes n} \begin{bmatrix} -1 & 0 & \cdots & 0 \\ 0 & 1 & 0 & \vdots \\ \vdots & 0 & \ddots & 0 \\ 0 & \cdots & 0 & 1 \end{bmatrix} H^{\otimes n} \\ &= H^{\otimes n} \left(\begin{bmatrix} 1 & 0 & \cdots & 0 \\ 0 & 1 & 0 & \vdots \\ \vdots & 0 & \ddots & 0 \\ 0 & \cdots & 0 & 1 \end{bmatrix} - 2 \begin{bmatrix} 1 & 0 & \cdots & 0 \\ 0 & 0 & 0 & \vdots \\ \vdots & 0 & \ddots & 0 \\ 0 & \cdots & 0 & 0 \end{bmatrix} \right) H^{\otimes n} \\ &= H^{\otimes n} (I - 2P_{|0\rangle}) H^{\otimes n} \\ &= I - 2P_{|Y\rangle} \end{aligned}$$

Unordered-Search Problem (8)

Action of $U_Y \equiv H^{\otimes n} U_0 H^{\otimes n}$ in the computational basis

$$P_{|u\rangle} |v\rangle = \langle u|v\rangle |u\rangle$$

$$N = 2^n$$

Thus $-U_Y = 2P_{|Y\rangle} - I$

Consider an arbitrary state of an n-qubit register $\sum_{k=0}^{N-1} \alpha_k |k\rangle$
where $N = 2^n$

$$\begin{aligned} P_{|Y\rangle} \sum_{k=0}^{N-1} \alpha_k |k\rangle &= \sum_{k=0}^{N-1} \alpha_k P_{|Y\rangle} |k\rangle = \sum_{k=0}^{N-1} \alpha_k \langle Y|k\rangle |Y\rangle \\ &= \sum_{k=0}^{N-1} \alpha_k \langle Y|k\rangle \left(\frac{1}{\sqrt{N}} \sum_{i=0}^{N-1} |i\rangle \right) = \sum_{i=0}^{N-1} \left(\sum_{k=0}^{N-1} \frac{\alpha_k \langle Y|k\rangle}{\sqrt{N}} \right) |i\rangle \\ &= \sum_{i=0}^{N-1} \left(\sum_{k=0}^{N-1} \frac{\alpha_k}{N} \right) |i\rangle = \sum_{k=0}^{N-1} \langle \alpha \rangle |k\rangle \end{aligned}$$

where $\langle \alpha \rangle$ is the mean average of all the α_k

Unordered-Search Problem (9)

Action of $U_Y \equiv H^{\otimes n} U_0 H^{\otimes n}$ in the computational basis

$$P_{|u\rangle} |v\rangle = \langle u|v\rangle |u\rangle \quad N=2^n$$

Thus $-U_Y = 2P_{|Y\rangle} - I$ maps the arbitrary state

$$\sum_{k=0}^{N-1} \alpha_k |k\rangle$$

to

$$\sum_{k=0}^{N-1} [-\alpha_k + 2\langle \alpha \rangle] |k\rangle$$

$-U_Y$ is sometimes referred to as the *inversion about the mean*:

$$\langle \alpha \rangle = \frac{1}{N} \sum_{k=0}^{N-1} \alpha_k \quad (\text{the mean average})$$

Unordered-Search Problem (10)

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

$$-U_Y : \sum_{k=0}^{N-1} \alpha_k |k\rangle \rightarrow \sum_{k=0}^{N-1} [-\alpha_k + 2\langle \alpha \rangle] |k\rangle$$

The quantum searching algorithm is

1. Prepare an n-qubit register in the state $|Y\rangle = \sum_{i=0}^{2^n-1} |i\rangle$
2. Apply the operator $G \equiv -U_Y U_f$ roughly $N^{1/2}$ times
3. Measure the register; output is x such that $f(x) = 1$ with high probability

We'll give a high-level analysis first, using the inversion-about-the-mean interpretation, then we'll give a more rigorous analysis using analysis

Unordered-Search Problem (11)

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

$$-U_Y : \sum_{k=0}^{N-1} \alpha_k |k\rangle \rightarrow \sum_{k=0}^{N-1} [-\alpha_k + 2\langle\alpha\rangle] |k\rangle$$

$$G \equiv -U_Y U_f$$

Suppose, for simplicity, $f(x) = 1$ for only one element x

It suffices to see what the effect of G is on the starting state (it's clear that continuing to apply G will produce the desired effect):

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

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

approx. mean avg.

Thus the amplitude of the solution state *increased* to roughly $3/N^{1/2}$, whereas the amplitude of the other states slightly *decreased*

Unordered-Search Problem (12)

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

$$G \equiv -U_Y U_f$$

Let's proceed with a more rigorous analysis of the algorithm

We can assume that $f(x)=1$ has more than one solution

Let X_1 be the set of solutions to $f(x)=1$ (the *good x's*)

Let X_0 be the set of solutions to $f(x)=0$ (the *bad x's*)

Define the equally-weighted superpositions of the "good" and "bad" states:

$$|X_1\rangle = \frac{1}{\sqrt{|X_1|}} \sum_{x \in X_1} |x\rangle$$

$$|X_0\rangle = \frac{1}{\sqrt{|X_0|}} \sum_{x \in X_0} |x\rangle$$

Unordered-Search Problem (13)

$$\mathbf{U}_f : |\mathbf{x}\rangle \rightarrow (-1)^{f(\mathbf{x})} |\mathbf{x}\rangle \quad |\mathbf{X}_1\rangle = \frac{1}{\sqrt{|\mathbf{X}_1|}} \sum_{x \in X_1} |x\rangle \quad |\mathbf{X}_0\rangle = \frac{1}{\sqrt{|\mathbf{X}_0|}} \sum_{x \in X_0} |x\rangle$$
$$\mathbf{G} \equiv -\mathbf{U}_Y \mathbf{U}_f$$

Write $|\mathbf{Y}\rangle$ as a linear combination of $|\mathbf{X}_0\rangle$ and $|\mathbf{X}_1\rangle$:

$$|\mathbf{Y}\rangle = \sqrt{\frac{|\mathbf{X}_0|}{N}} |\mathbf{X}_0\rangle + \sqrt{\frac{|\mathbf{X}_1|}{N}} |\mathbf{X}_1\rangle \equiv \sqrt{p_0} |\mathbf{X}_0\rangle + \sqrt{p_1} |\mathbf{X}_1\rangle$$

Let $\omega \in (0, 1/2)$ be the real number such that

$$p_1 = \sin^2(\pi\omega) \quad \text{and} \quad p_0 = \cos^2(\pi\omega)$$

Thus $|\mathbf{Y}\rangle = \cos(\pi\omega) |\mathbf{X}_0\rangle + \sin(\pi\omega) |\mathbf{X}_1\rangle$

Let $|\mathbf{Y}'\rangle = \sin(\pi\omega) |\mathbf{X}_0\rangle - \cos(\pi\omega) |\mathbf{X}_1\rangle$

Unordered-Search Problem (14)

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

$$G \equiv -U_Y U_f$$

$$|Y\rangle = \cos(\pi\omega) |X_0\rangle + \sin(\pi\omega) |X_1\rangle$$

$$|Y'\rangle = \sin(\pi\omega) |X_0\rangle - \cos(\pi\omega) |X_1\rangle$$

We now have *two (orthonormal) bases* for the same 2-dimensional subspace:

$$\{ |X_0\rangle, |X_1\rangle \} \quad \text{and} \quad \{ |Y\rangle, |Y'\rangle \}$$

(Note U_f and $-U_Y$ are reflections in this plane, changing the sign of one coordinate)

We analyse the action of $G = -U_Y U_f$ using these two bases

Let $|\Psi\rangle = \cos(\varphi) |X_0\rangle + \sin(\varphi) |X_1\rangle$ be any state

$$U_f |\Psi\rangle = \cos(\varphi) |X_0\rangle - \sin(\varphi) |X_1\rangle$$

$$= \cos(\varphi + \pi\omega) |Y\rangle + \sin(\varphi + \pi\omega) |Y'\rangle \quad \text{(switch to basis } \{ |Y\rangle, |Y'\rangle \} \text{ to apply } -U_Y)$$

$$G |\Psi\rangle = \cos(\varphi + \pi\omega) |Y\rangle - \sin(\varphi + \pi\omega) |Y'\rangle$$

$$= \cos(\varphi + 2\pi\omega) |X_0\rangle + \sin(\varphi + 2\pi\omega) |X_1\rangle$$

Thus G (the product of two reflections) gives us a rotation of $2\pi\omega$ in the plane spanned by $|X_0\rangle$ and $|X_1\rangle$

Unordered-Search Problem (15)

$$G: \cos(\varphi) |X_0\rangle + \sin(\varphi) |X_1\rangle \rightarrow \cos(\varphi + 2\pi\omega) |X_0\rangle - \sin(\varphi + 2\pi\omega) |X_1\rangle$$

Since the search algorithm starts in the state with $\varphi = \pi\omega$, we see that after k applications of G , the state of the n -qubit register is

$$\cos((2k+1)\pi\omega) |X_0\rangle + \sin((2k+1)\pi\omega) |X_1\rangle$$

To measure a good state, we want $\sin((2k+1)\pi\omega) \sim 1$

This is true when $k \sim 1/(4\omega) - 1/2$

$$\sim \pi/[4(p_1)^{1/2}]$$

$$= \pi/[4(|X_1|/N)^{1/2}]$$

$$= \pi N^{1/2}/[4(|X_1|)^{1/2}]$$

Thus applying G $\pi N^{1/2}/[4(|X_1|)^{1/2}]$ times gives a state with large probability amplitude in the state $|X_1\rangle$

Unordered-Search Problem (16)

The operator G , also called the *Grover iterate*, has eigenvectors in the 2-dimensional space

$$|\Psi_+\rangle = \frac{1}{\sqrt{2}}|X_0\rangle + \frac{i}{\sqrt{2}}|X_1\rangle$$

$$|\Psi_-\rangle = \frac{1}{\sqrt{2}}|X_0\rangle - \frac{i}{\sqrt{2}}|X_1\rangle$$

with respective eigenvalues $e^{2\pi i\omega}$ and $e^{-2\pi i\omega}$

The initial state $|Y\rangle = \cos(\pi\omega)|X_0\rangle + \sin(\pi\omega)|X_1\rangle$ is expressed in this eigenbasis as

$$\frac{e^{\pi i\omega}}{\sqrt{2}}|\Psi_+\rangle + \frac{e^{-\pi i\omega}}{\sqrt{2}}|\Psi_-\rangle$$

Thus applying G^k to this state gives

$$\frac{e^{\pi i(2k+1)\omega}}{\sqrt{2}}|\Psi_+\rangle + \frac{e^{-\pi i(2k+1)\omega}}{\sqrt{2}}|\Psi_-\rangle$$

This gives an alternative derivation of the number k of required applications of G in the search algorithm

Counting

$G = -U_y U_f$ has eigenvalues $e^{2\pi i \omega}$ and $e^{-2\pi i \omega}$

Suppose we want to count the number of solutions to $f(x) = 1$

This number is just $|X_1| = Np_1$, where $p_1 = \sin^2(\pi\omega)$

Thus estimating ω using the eigenvalue estimation algorithm will give us an estimate of $|X_1|$

(Also works when $|X_1| = 0$ or $|X_1| = N$)

8-lecture Mini-course in Quantum Computation

Lecture 8

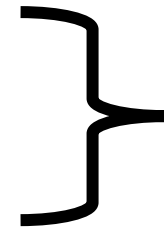
Lawrence Ioannou

Review of n-qubit Hadamard Gate, $H^{\otimes n}$

Recall the one-qubit Hadamard gate, H , maps

$$H: |0\rangle \rightarrow (|0\rangle + |1\rangle)/2^{1/2}$$

$$|1\rangle \rightarrow (|0\rangle - |1\rangle)/2^{1/2}$$



$$\frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$$

the matrix-representation, with respect to the computational basis, of the Hadamard gate

Alternatively

$$H: |b\rangle \rightarrow (|0\rangle + (-1)^b |1\rangle)/2^{1/2}, \quad \text{for } b \in \{0,1\}$$

which can also be written

$$H: |b\rangle \rightarrow (\sum_x (-1)^{b \cdot x} |x\rangle)/2^{1/2}, \quad \text{for } b, x \in \{0,1\}$$

Exercise: show that the n-qubit Hadamard gate, $H^{\otimes n}$, maps

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

where $x \cdot y = \sum_i x_i y_i$, where $x = x_{n-1} x_{n-2} \dots x_1 x_0 \in \{0,1\}^n$ and similarly for y

Measurement as a vector-projection

Recall $P_{|u\rangle}$ is the (linear) projection operator, projecting onto the vector $|u\rangle$:

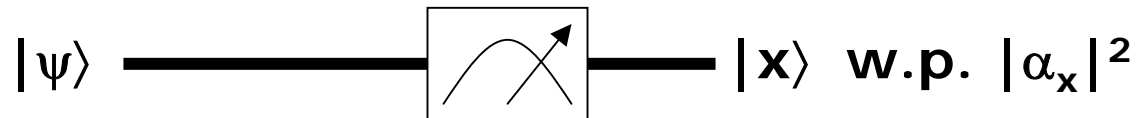
$$P_{|u\rangle} |v\rangle = \langle u|v\rangle |u\rangle$$

inner product of vector $|u\rangle$ and vector $|v\rangle$

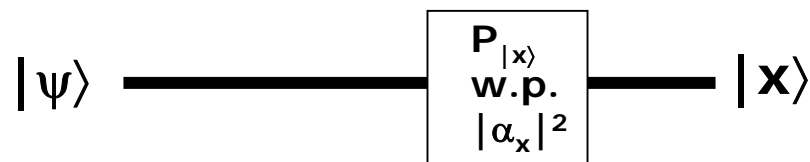
A measurement of the n-qubit state

$$|\psi\rangle = \sum_{x \in \{0,1\}^n} \alpha_x |x\rangle$$

with respect to the computational basis can be thought of as applying (to $|\psi\rangle$) the projection operator $P_{|x\rangle}$ with probability α_x (and then renormalising $P_{|x\rangle}|\psi\rangle$ so it's a unit-vector still)



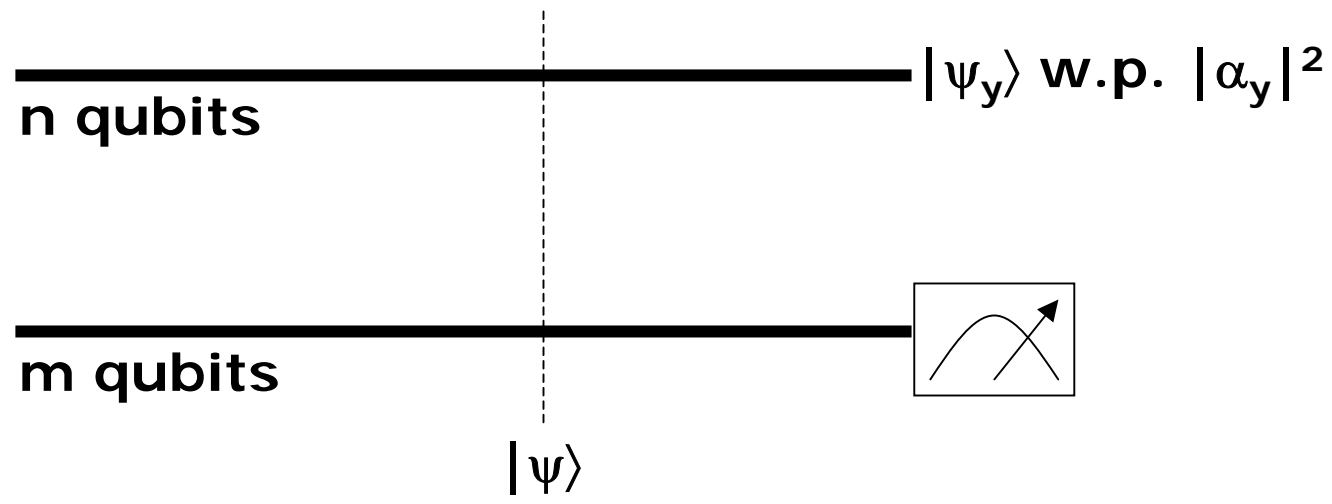
is equivalent to



(Note "w.p." is shorthand for "with probability")

Measurement as a vector-projection (2)

What if we have two registers, an n-qubit register and an m-qubit register, and we measure just the m qubit register?

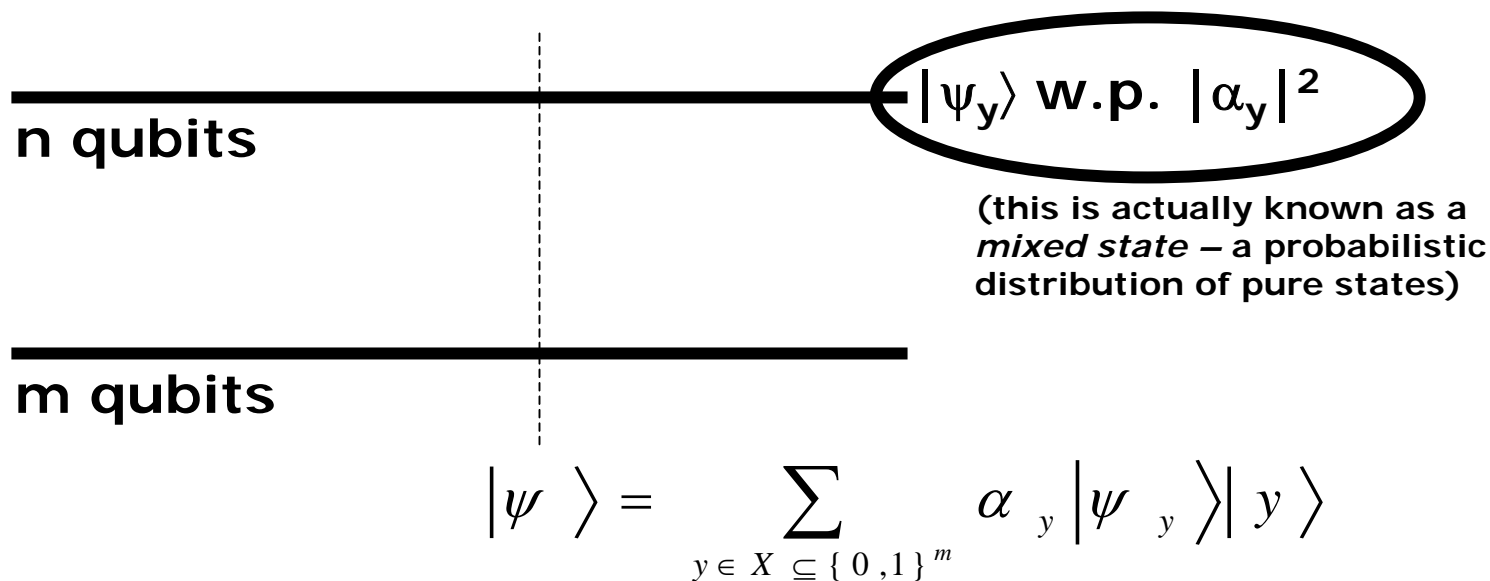


Suppose $|\psi\rangle$ is of the form

$$|\psi\rangle = \sum_{y \in X \subseteq \{0,1\}^m} \alpha_y |\psi_y\rangle |y\rangle$$

Then measuring the second (bottom) register is equivalent to applying (to $|\psi\rangle$) the projector $I \otimes P_{|y\rangle}$ w.p. $|\alpha_y|^2$

Measurement as a vector-projection (3)



Note that if we did not actually do the measurement, but just *ignored* (or *threw away*, or *traced out*) the second (bottom) register, then we can just think of the first (upper) register as in the state $|\psi_y\rangle$ w.p. $|\alpha_y|^2$

Sometimes, to make notation simpler, we ignore the bottom register after some point in the computation or (equivalently) assume that we have measured it

Simon's problem

Given a black box which computes the function

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

$f(x) = f(x \oplus s)$, for some $s \in \{0,1\}^n$

Find the "period" s

Example (taken from Artur Ekert's course notes):

x	$f(x)$	$s = 110$ is the period (in the additive group $(\mathbb{Z}_2)^3$)
000	111	
001	010	$f(x \oplus 110) = f(x)$
010	100	
011	110	
100	100	$f(000) = f(000 \oplus 110) = f(110) = 111$
101	110	$f(001) = f(001 \oplus 110) = f(111) = 010$
110	111	$f(010) = f(010 \oplus 110) = f(100) = 100$
111	010	$f(011) = f(011 \oplus 110) = f(101) = 110$

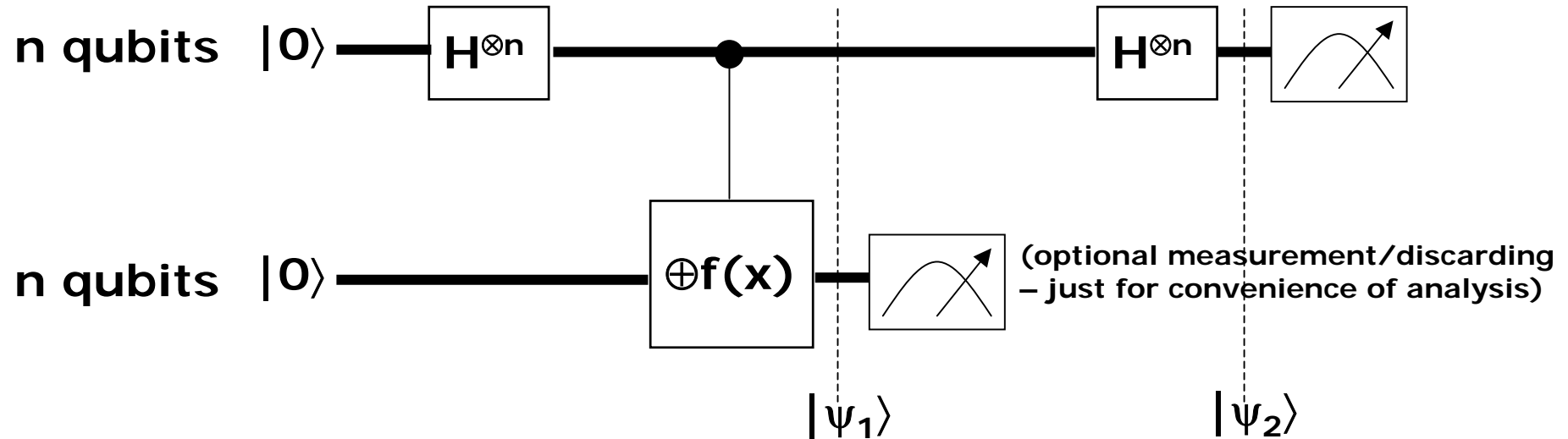
Simon's problem (2)

Given a black box which computes the function

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

$f(x) = f(x \oplus s)$, for some $s \in \{0,1\}^n$

Find the "period" s



$$\begin{aligned}
 |\psi_1\rangle &= \sum_{x \in \{0,1\}^n} |x\rangle |f(x)\rangle = \sum_{f(x_0) \in \text{Range}(f)} \left(\sum_{x: f(x)=f(x_0)} |x\rangle \right) |f(x_0)\rangle \\
 &= \sum_{f(x_0) \in \text{Range}(f)} (|x_0\rangle + |x_0 \oplus s\rangle) |f(x_0)\rangle
 \end{aligned}$$

$$|\psi_2\rangle = H^{\otimes n} (|x_0\rangle + |x_0 \oplus s\rangle) = \sum_y (-1)^{y \cdot x_0} [1 + (-1)^{s \cdot y}] |y\rangle$$

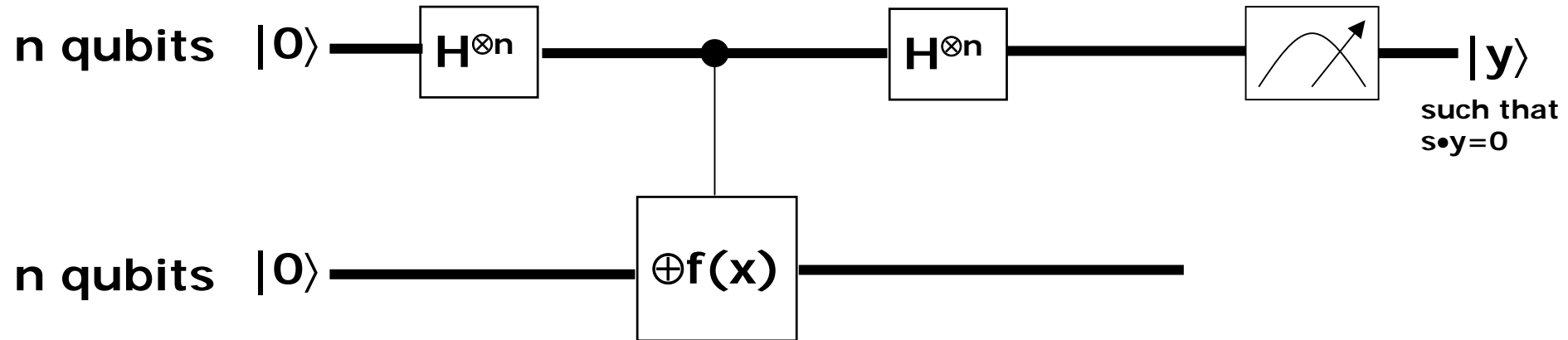
Simon's problem (3)

Given a black box which computes the function

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

$f(x) = f(x \oplus s)$, for some $s \in \{0,1\}^n$

Find the "period" s



Thus we effectively end up measuring the state

$$\sum_y (-1)^{y \cdot x_0} [1 + (-1)^{s \cdot y}] |y\rangle$$

Note that the probability amplitude of $|y\rangle$ such that $s \cdot y = 1$ is 0

Thus the result of the measurement is a particular y such that $s \cdot y = 0$, with probability $2/2^n$

Simon's problem (4)

Given a black box which computes the function

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

$f(x) = f(x \oplus s)$, for some $s \in \{0,1\}^n$

Find the "period" s

Simon's algorithm is to repeat the previous experiment roughly n times to get a linearly independent set $y^{(1)}, y^{(2)}, \dots, y^{(n)}$ such that $y^{(i)} \bullet s = 0$ (the probability of failing to produce a linearly independent set is less than 0.75)

Then we can solve for s by solving the system of n linear equations:

$$\begin{aligned} y^{(1)} \bullet s &= 0 \\ y^{(2)} \bullet s &= 0 \\ &\vdots \\ y^{(n)} \bullet s &= 0 \end{aligned}$$

It may help to think of each equation $y^{(i)} \bullet s = 0$ as

$$y_0^{(i)} s_0 + y_1^{(i)} s_1 + \dots + y_{n-1}^{(i)} s_{n-1} = 0$$

where the s_i are the n "unknowns"

Hidden Subgroup Problem (HSP)

The problems studied so far are special cases of the more general *hidden subgroup problem*:

Let f be a function from a finitely generated group G to a finite set X , such that

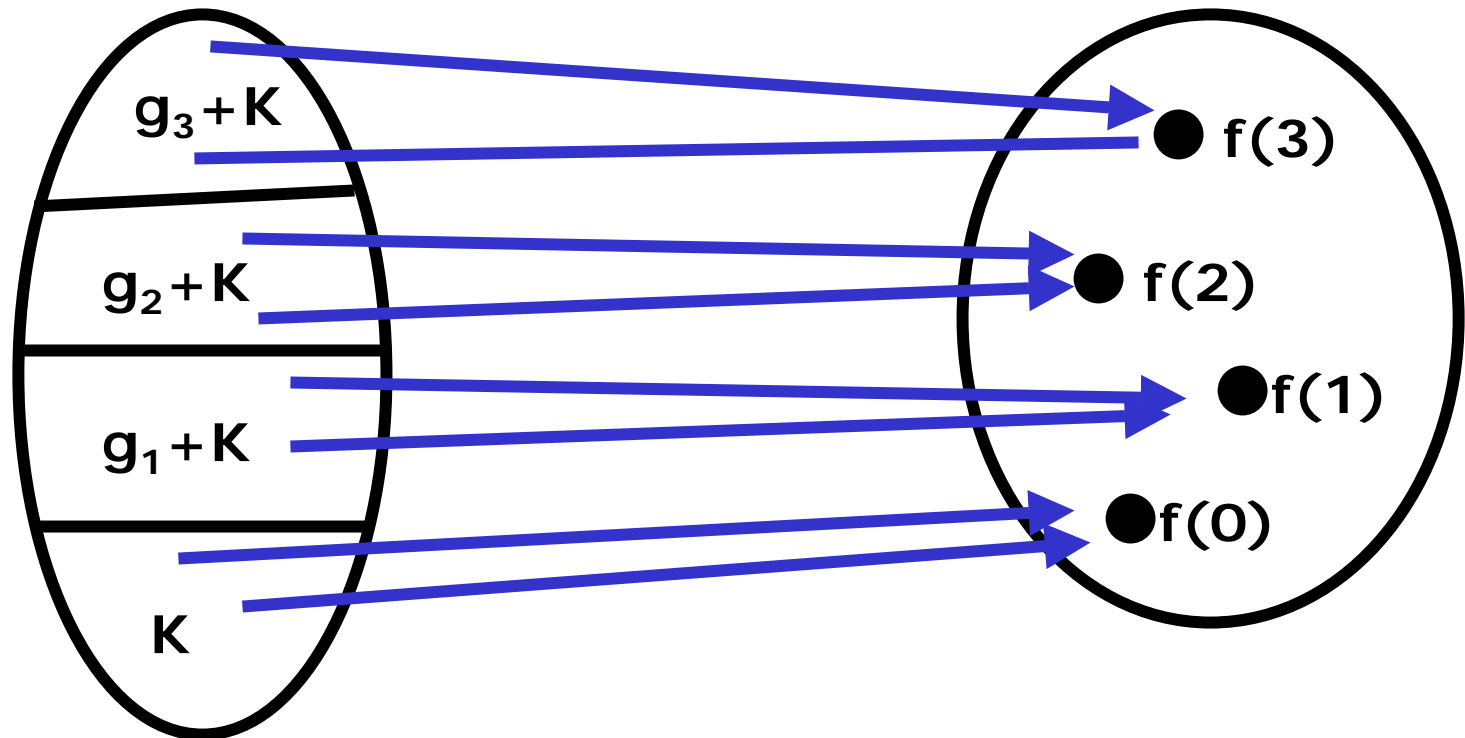
- (i) f is constant on cosets of a subgroup K
- and
- (ii) f is distinct on each coset.

Given the controlled- $\oplus f(x)$ gate, find a generating set for K

The idea is that the function f is “hiding” the subgroup K

HSP (2)

We need the group to be Abelian (so that it is a product of cyclic groups)



g_i+K denotes a coset of K

HSP (3)

Here are the problems we have studied, a la HSP:

Deutch's problem:

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

$$f(x)=f(y) \text{ iff } x-y \in K, \text{ where either}$$

$$K=\{0\} \text{ or } K=\{0,1\}$$

Period-Finding:

$$f: \mathbb{Z} \rightarrow \text{any finite set}$$

$$f(x)=f(y) \text{ iff } x-y \in K, \text{ where}$$

$$K=r\mathbb{Z}$$

Discrete Logarithm Problem:

$$f: \mathbb{Z}_r \times \mathbb{Z}_r \rightarrow H, \text{ where } H \text{ is a group (in which we define the DLP)}$$

$$f(x_1, x_2) = f(y_1, y_2) \text{ iff } (x_1, x_2) - (y_1, y_2) \in K, \text{ where}$$

$$K = \langle (1, -s) \rangle = \{ (k, -ks), k=0, 1, \dots, r-1 \}$$

$$(\text{Recall } f(x_1, x_2) := \alpha^{sx_1 + x_2} = \alpha^{x_2} \beta^{x_1}, \alpha \text{ has order } r \text{ in } H)$$

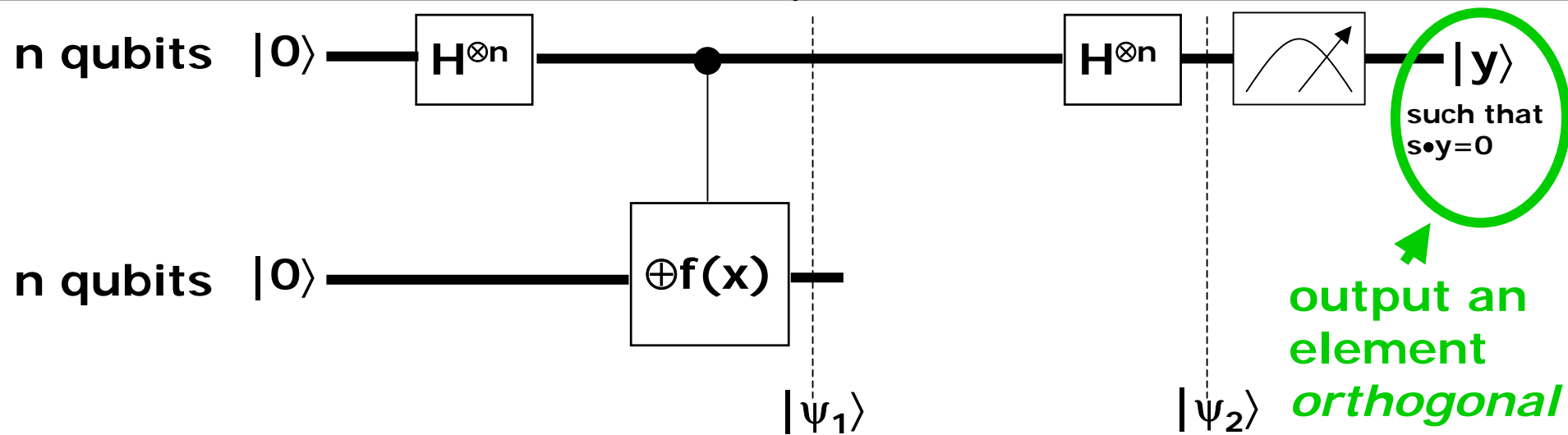
HSP (4): Simon's problem as HSP

Given a black box which computes the function

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

$f(x) = f(x \oplus s)$, for some $s \in \{0,1\}^n$

Find the "period" s



$$|\psi_1\rangle = \sum_{f(x_0) \in \text{Range}(f)} (|x_0\rangle + |x_0 \oplus s\rangle) |f(x_0)\rangle$$

sum over elements of a coset of $K = \{0, s\}$

$$|\psi_2\rangle = H^{\otimes n} (|x_0\rangle + |x_0 \oplus s\rangle) = \sum_y (-1)^{y \cdot x_0} [1 + (-1)^{s \cdot y}] |y\rangle$$

information about *which particular* coset ends up in the phase

HSP (5)

It turns out that we can reduce the case of finitely-generated (Abelian) G to finite G , by using the period-finding algorithm to find the period of f on each of the generators (e.g. $\mathbb{Z} \rightarrow \mathbb{Z}_r$)

As well, the HSP algorithm itself can be used to explicitly decompose finite G into a product of cyclic subgroups

$$G = \mathbb{Z}_{N^{(1)}} \times \mathbb{Z}_{N^{(2)}} \times \mathbb{Z}_{N^{(3)}} \times \dots \times \mathbb{Z}_{N^{(m)}}$$

Further, with the factoring algorithm, we can factor each $N^{(j)}$ into its prime factors, and then efficiently find the isomorphism between $\mathbb{Z}_{N^{(j)}}$ and the product of cyclic groups of prime-power order; thus we can assume $N^{(j)}$ are prime powers

(If G is a group of order $p^k m$ where $\gcd(p^k, m) = 1$, let G_p denote the *Sylow p -subgroup* of G , $|G_p| = p^k$)

Finally, any subgroup K of an Abelian $G = G_{p^{(1)}} \times G_{p^{(2)}} \times \dots \times G_{p^{(k)}}$ is of the form $K_{p^{(1)}} \times K_{p^{(2)}} \times \dots \times K_{p^{(k)}}$ where $K_{p^{(j)}}$ is a subgroup of $G_{p^{(j)}}$. Thus K can be found piecewise: For $j = 1, \dots, k$, find the hidden subgroup $K_{p^{(j)}}$ of the function $f_{p^{(j)}}: G_{p^{(j)}} \rightarrow X$, where $f_{p^{(j)}}(x) = f(0, \dots, 0, x, 0, \dots, 0)$ (x appears in the j th entry)

HSP (6)

If G is a group of order $p^t m$ where $\gcd(p^t, m) = 1$, let G_p denote the *Sylow p -subgroup* of G , $|G_p| = p^t$

Therefore, we can restrict attention to finite groups $G = G_p$

$$G = G_p \approx \mathbb{Z}_{p^{a_1}} \times \mathbb{Z}_{p^{a_2}} \times \mathbb{Z}_{p^{a_3}} \times \dots \times \mathbb{Z}_{p^{a_t}}$$

for some prime p and positive integers a_j

Let $a = \max\{a_j\}$

We'll explain the hidden subgroup algorithm by analogy to the period-finding algorithm, using eigenvalue-estimation analysis...

HSP (7)

$$\mathbf{G} \approx \mathbf{Z}_{p^{a_1}} \times \mathbf{Z}_{p^{a_2}} \times \mathbf{Z}_{p^{a_3}} \times \dots \times \mathbf{Z}_{p^{a_t}}$$

Let $a = \max\{a_j\}$

Recall the eigenvectors and eigenvalues of the shift operator in the period-finding problem

$$U_{sh(f)} |\Psi_k\rangle = e^{2\pi i \frac{k}{r}} |\Psi_k\rangle \quad |\Psi_k\rangle = \sum_{j=0}^{r-1} e^{-2\pi i \frac{jk}{r}} |f(j)\rangle \quad \text{for } k=0,1,\dots,r-1$$

and recall that the hidden subgroup was $r\mathbf{Z} = \{0, r, 2r, \dots\}$

Rewrite the eigenvectors more group-theoretically

$$|\Psi_k\rangle = \sum_{s \in \mathbf{Z}/r\mathbf{Z}} e^{-2\pi i s \frac{k}{r}} |f(s)\rangle$$

where the sum is over a set of coset representatives (recall \mathbf{G}/\mathbf{K} is the family of all cosets of \mathbf{K} in \mathbf{G})

There is an eigenvector corresponding to each k that satisfies

$$2^n h \frac{k}{r} \equiv 0 \pmod{2^n}, \text{ for all } h \in r\mathbf{Z}$$

HSP (8)

$$\mathbf{G} \approx \mathbf{Z}_{p^{a_1}} \times \mathbf{Z}_{p^{a_2}} \times \mathbf{Z}_{p^{a_3}} \times \dots \times \mathbf{Z}_{p^{a_t}}$$

$$\text{Let } a = \max\{a_j\}$$

Analogously, define the eigenstates (of the shift operator) for every $\mathbf{k} = (k_1, \dots, k_t) \in \mathbf{Z}_{p^{a_1}} \times \mathbf{Z}_{p^{a_2}} \times \mathbf{Z}_{p^{a_3}} \times \dots \times \mathbf{Z}_{p^{a_t}}$ satisfying

$$p^a \sum_{j=1}^t h_j \frac{k_j}{p^{a_j}} \equiv 0 \pmod{p^a}, \text{ for all } h \in \mathbf{K}$$

as

$$\left| \Psi_{(k_1, \dots, k_t)} \right\rangle = \sum_{(s_1, \dots, s_t) \in G/K} e^{-2\pi i \sum_{j=1}^t s_j \frac{k_j}{p^{a_j}}} \left| f((s_1, \dots, s_t)) \right\rangle$$

with eigenvalues

$$e^{2\pi i \sum_{j=1}^t \frac{k_j}{p^{a_j}}}$$

HSP (9)

Let T be the set of all $k=(k_1,\dots,k_t)$ that satisfy

$$p^a \sum_{j=1}^t h_j \frac{k_j}{p^{a_j}} \equiv 0 \pmod{p^a}, \text{ for all } h=(h_1,\dots,h_t) \in K$$

The following algorithm determines a uniformly random $k \in T$, and by the above equation we can use enough random k to determine the subgroup K using linear algebra (like in Simon's algorithm)

HSP Algorithm:

1. Using $\text{QFT}(p^{a_j})$ for $j=1,\dots,t$, and the black box for f , create the state

$$\sum_{x \in Z_{p^{a_1}} \times Z_{p^{a_2}} \times \dots \times Z_{p^{a_t}}} |x\rangle |f(x)\rangle = \sum_{k \in T} \left(\sum_{x_1=0}^{p^{a_1}-1} e^{2\pi i \frac{x_1 k_1}{p^{a_1}}} |x_1\rangle \right) \dots \left(\sum_{x_t=0}^{p^{a_t}-1} e^{2\pi i \frac{x_t k_t}{p^{a_t}}} |x_t\rangle \right) |\Psi_k\rangle$$

2. Apply $\text{QFT}(p^{a_1})^{-1} \otimes \dots \otimes \text{QFT}(p^{a_t})^{-1} \otimes I$ to get

$$\sum_{k \in T} |k_1\rangle \dots |k_t\rangle |\Psi_k\rangle$$

3. Measure the control registers, output k_1,\dots,k_t