

Introduction to Quantum Information Processing

Lecture 6

Richard Cleve

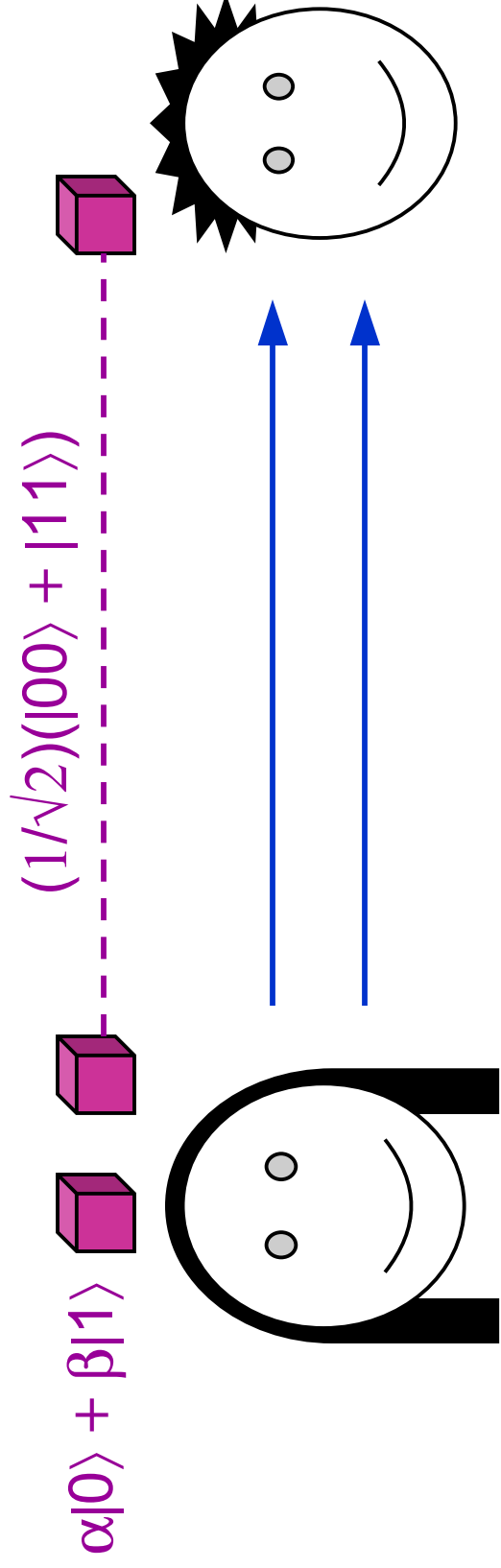
Overview of Lecture 6

- Continuation of teleportation
- Computation and some basic complexity classes
- Simple quantum algorithms in the query scenario (Deutsch)

Recap of teleportation scenario

Goal: for Alice to convey her qubit to Bob

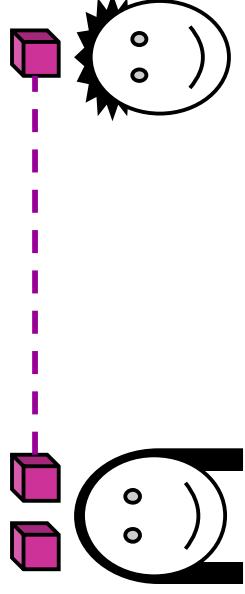
Resources: an entangled state and two bits communication



Note that the initial state of the three qubit system is:

$$(1/\sqrt{2})(\alpha|0\rangle + \beta|1\rangle)(|00\rangle + |11\rangle) \\ = (1/\sqrt{2})(\alpha|000\rangle + \alpha|011\rangle + \beta|100\rangle + \beta|111\rangle)$$

How teleportation works

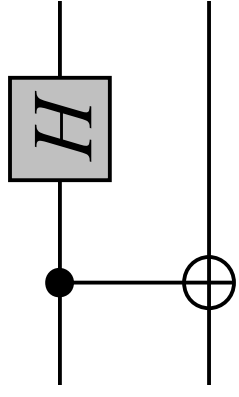


Initial state: $(\alpha|0\rangle + \beta|1\rangle)(|00\rangle + |11\rangle)$ (omitting the $1/\sqrt{2}$ factor)

$$= \alpha|000\rangle + \alpha|011\rangle + \beta|100\rangle + \beta|111\rangle$$
$$= \frac{1}{2}(|00\rangle + |11\rangle)(\alpha|0\rangle + \beta|1\rangle)$$
$$+ \frac{1}{2}(|00\rangle - |11\rangle)(\alpha|1\rangle + \beta|0\rangle)$$
$$+ \frac{1}{2}(|01\rangle + |10\rangle)(\alpha|0\rangle - \beta|1\rangle)$$
$$+ \frac{1}{2}(|01\rangle - |10\rangle)(\alpha|1\rangle - \beta|0\rangle)$$

Protocol: Alice measures her two qubits *in the Bell basis* and sends the result to Bob (who then “corrects” his state) ⁴

What Alice does specifically



Alice applies

to her two qubits, yielding:

$$\left\{ \begin{array}{l} \frac{1}{2}|00\rangle(\alpha|0\rangle + \beta|1\rangle) \\ + \frac{1}{2}|01\rangle(\alpha|1\rangle + \beta|0\rangle) \\ + \frac{1}{2}|10\rangle(\alpha|0\rangle - \beta|1\rangle) \\ + \frac{1}{2}|11\rangle(\alpha|1\rangle - \beta|0\rangle) \end{array} \right.$$

$$\left\{ \begin{array}{l} (00, \alpha|0\rangle + \beta|1\rangle) \text{ with prob. } \frac{1}{4} \\ (01, \alpha|1\rangle + \beta|0\rangle) \text{ with prob. } \frac{1}{4} \\ (10, \alpha|0\rangle - \beta|1\rangle) \text{ with prob. } \frac{1}{4} \\ (11, \alpha|1\rangle - \beta|0\rangle) \text{ with prob. } \frac{1}{4} \end{array} \right.$$

Then Alice sends her two classical bits to Bob, who then adjusts his qubit to be $\alpha|0\rangle + \beta|1\rangle$ whatever case occurs

Bob's adjustment procedure

Bob receives two classical bits a , b from Alice, and:

if $b = 1$ he applies X to qubit

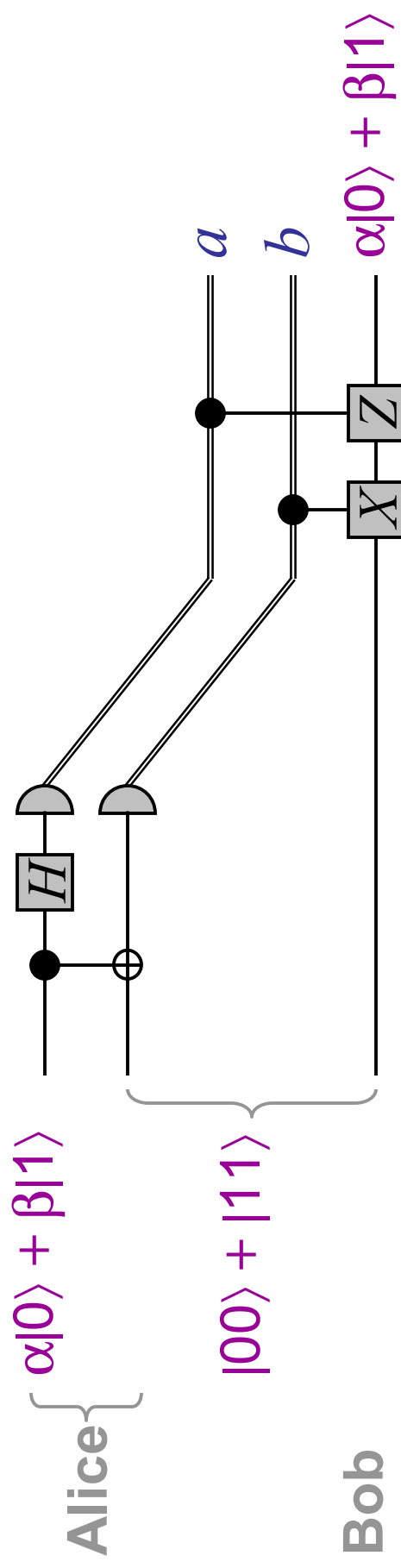
if $a = 1$ he applies Z to qubit

$$X = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \quad Z = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}$$

yielding:
$$\left\{ \begin{array}{l} 00, \quad \alpha|0\rangle + \beta|1\rangle \\ 01, \quad X(\alpha|1\rangle + \beta|0\rangle) = \alpha|0\rangle + \beta|1\rangle \\ 10, \quad Z(\alpha|0\rangle - \beta|1\rangle) = \alpha|0\rangle + \beta|1\rangle \\ 11, \quad ZX(\alpha|1\rangle - \beta|0\rangle) = \alpha|0\rangle + \beta|1\rangle \end{array} \right.$$

Note that Bob acquires the correct state in each case

Summary of teleportation

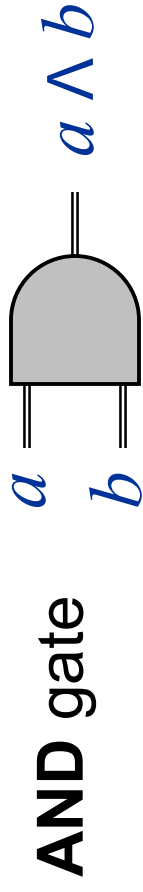


Suggested exercise: try to work through the analysis of the teleportation protocol on your own

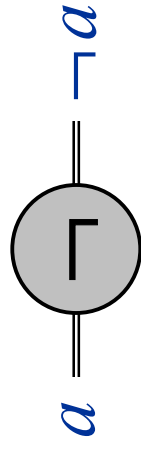
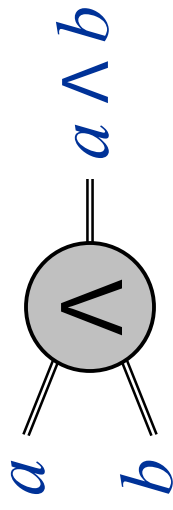
computation and basic complexity classes

Classical (boolean logic) gates

“old” notation



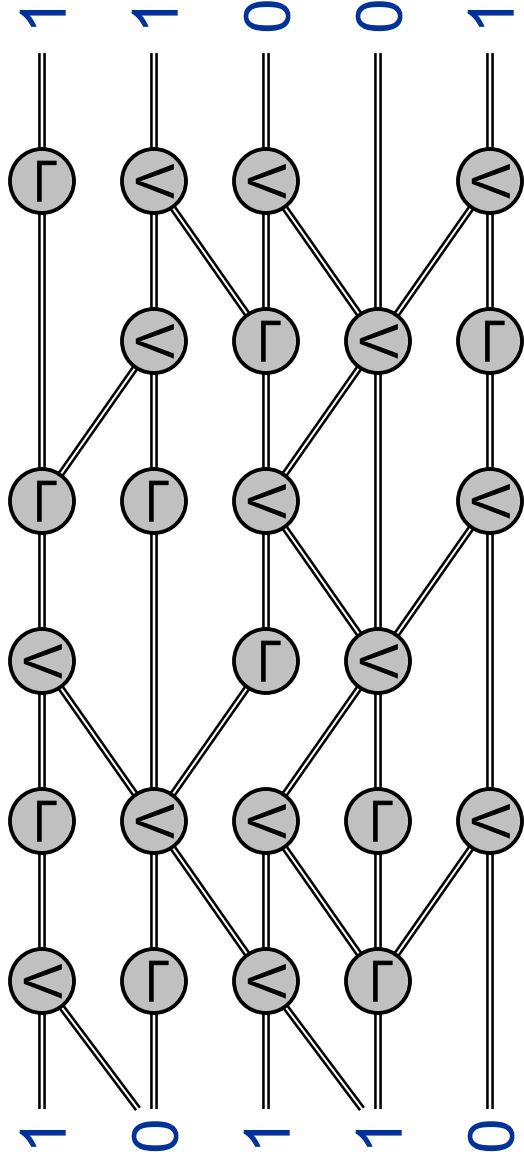
“new” notation



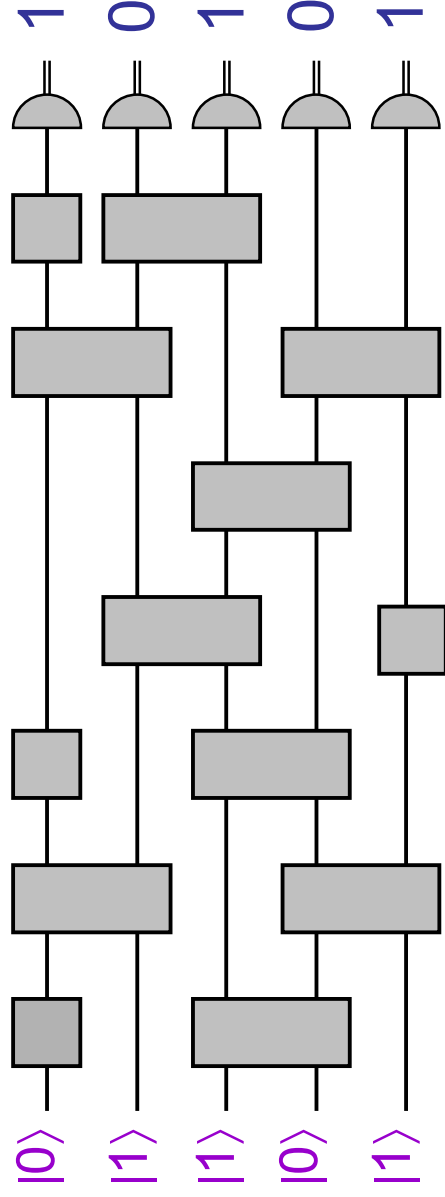
Note: an **OR** gate can be simulated by one **AND** gate and three **NOT** gates

Models of computation

Classical circuits:



Quantum circuits:



Multiplication problem

Input: two n -bit numbers (e.g. 101 and 111)

Output: their product (e.g. 100011)

- “Grade school” algorithm costs $O(n^2)$
- Best currently-known **classical** algorithm costs $O(n \log n \log \log n)$
- Best currently-known **quantum** method: same

Factoring problem

Input: an n -bit number (e.g. 100011)

Output: their product (e.g. 101, 111)

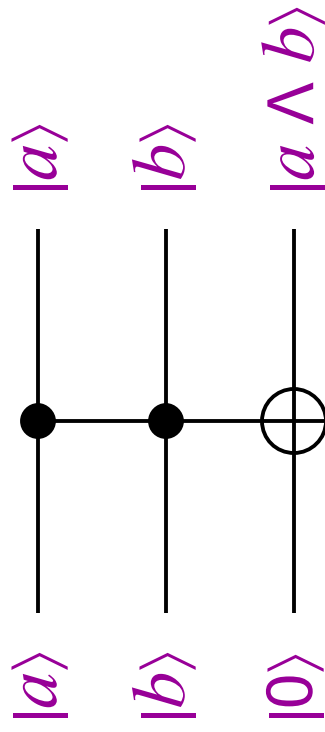
- Trial division costs $\approx 2^{n/2}$
- Best currently-known **classical** algorithm costs $\approx 2^{n^{1/3}}$
- Hardness of factoring is the basis of the security of many cryptosystems (e.g. RSA)
- Shor's **quantum** algorithm costs $\approx n^2$
- Implementation would break RSA and many other cryptosystems

Quantum vs. classical circuits

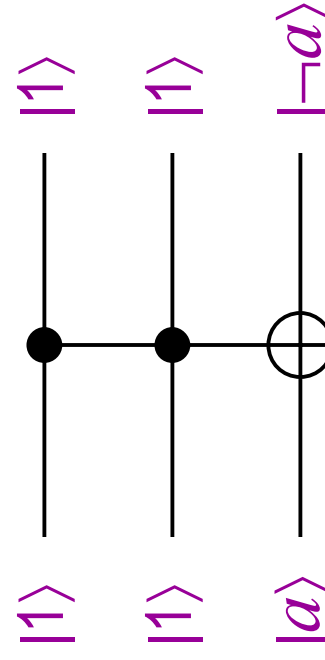
Theorem: a classical circuit of size s can be simulated by a quantum circuit of size $O(s)$

Idea: using Toffoli gates, one can simulate:

AND gates



NOT gates



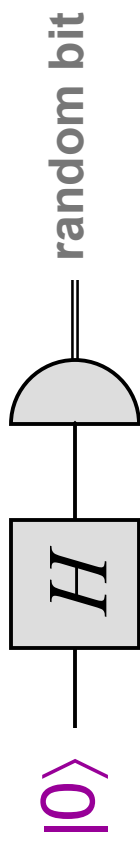
Some complexity classes

- **P (polynomial time):** problems solved by $O(n^c)$ -size classical circuits (decision problems and uniform circuit families)
- **BPP (bounded error probabilistic polynomial time):** problems solved by $O(n^c)$ -size *probabilistic* circuits that err with probability $\leq 1/4$
- **BQP (bounded error quantum polynomial time):** problems solved by $O(n^c)$ -size *probabilistic* circuits that err with probability $\leq 1/4$
- **EXP (exponential time):** problems solved by $O(2^{n^c})$ -size circuits.

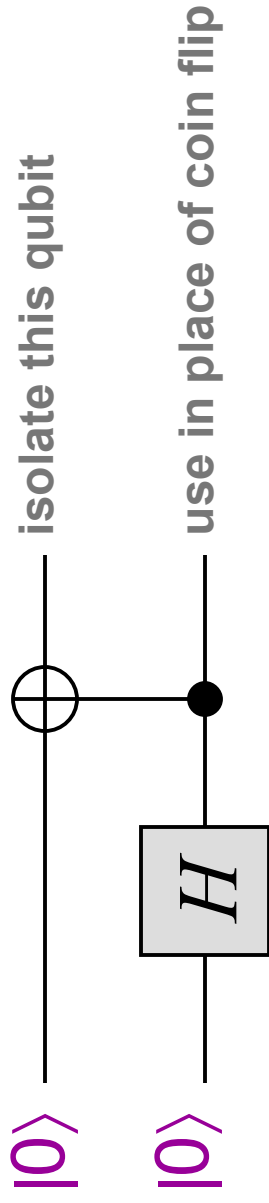
BPP \subseteq BQP

Since quantum gates can simulate classical gates, the only outstanding issue is how to simulate randomness

To simulate “coin flips”, one can use the circuit:



It can also be done without intermediate measurements:



BQP \subseteq EXP

Theorem: a quantum circuit of size s can be simulated by a classical circuit of size $O(n^{cs})$

Idea: to simulate an n -qubit state, use an array of size 2^n containing values of all 2^n amplitudes within precision 2^{-n}

α_{000}
α_{001}
α_{010}
α_{011}
...
α_{111}

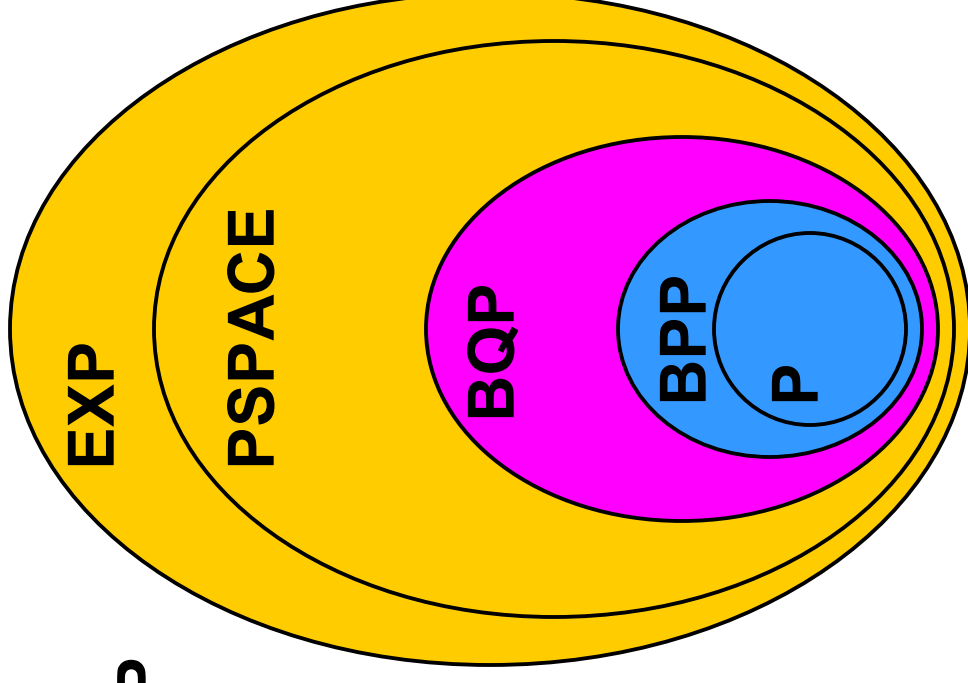
Can adjust this state vector whenever a unitary operation is performed

From the final amplitudes, can determine how to set each output bit

Exercise: show how to do the simulation using only a polynomial amount of **space** (memory)

Summary of basic containments

$$P \subseteq BPP \subseteq BQP \subseteq PSPACE \subseteq EXP$$

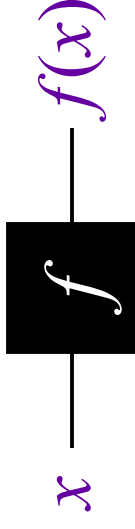


This picture will be fleshed out further in due course

simple quantum algorithms in the query scenario

Query scenario

Input: a function f , given as a black box (a.k.a. oracle)



Goal: determine some information about f making as few queries to f (and other operations) as possible

Example: polynomial interpolation

Let: $f(x) = c_0 + c_1x + c_2x^2 + \dots + c_dx^d$

Goal: determine $c_0, c_1, c_2, \dots, c_d$

Question: How many f -queries does this take?

Deutsch's problem



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

There are **four** possibilities:

x	$f_1(x)$	x	$f_2(x)$	x	$f_3(x)$	x	$f_4(x)$
0	0	0	1	0	0	0	1
1	0	1	1	1	1	1	0

Goal: determine whether or not $f(0) = f(1)$ (i.e. $f(0) \oplus f(1)$)

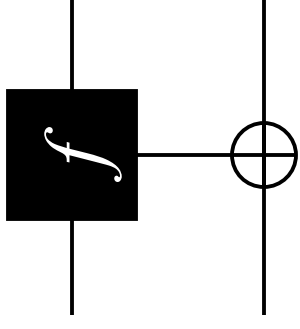
Any classical method requires **two** queries

What about a quantum method?

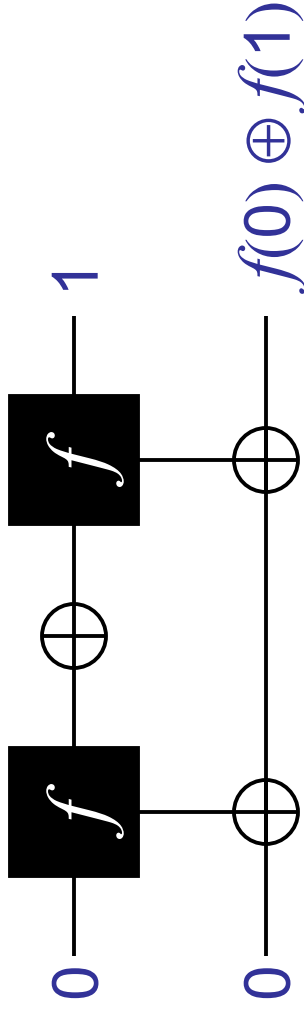
Reversible black box for f



alternate
notation:

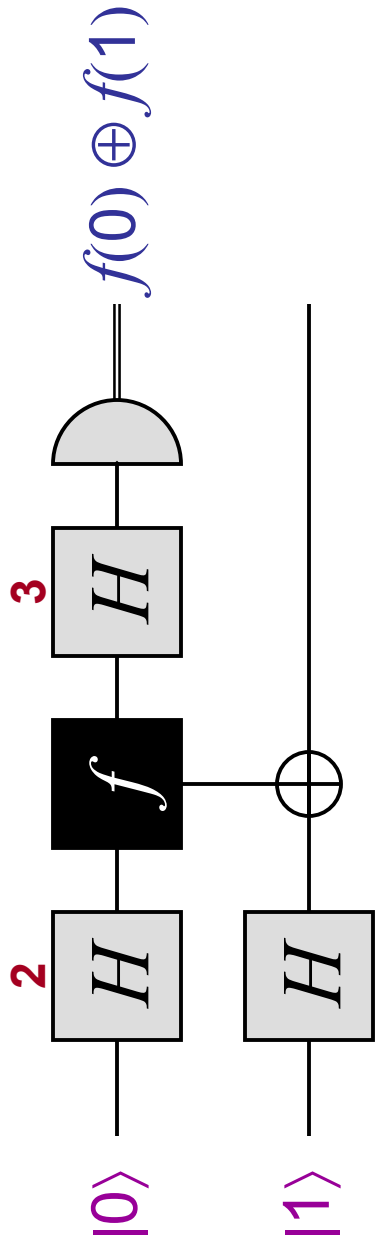


A classical algorithm:
(still requires 2 queries)



2 queries + 1 auxiliary operation

Quantum algorithm for Deutsch



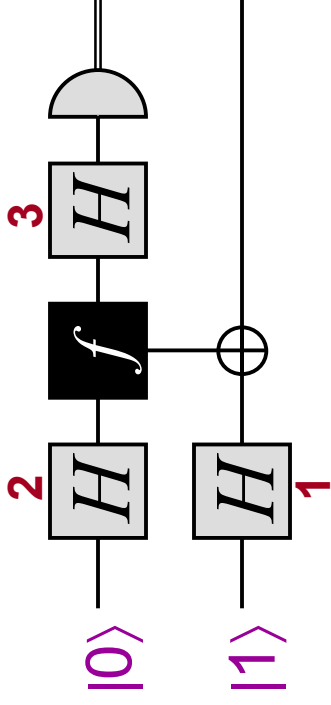
$$H = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$$

1 query + 4 auxiliary operations

How does this algorithm work?

Each of the three H operations plays a different role ...

Quantum algorithm (1)



1. Creates the state $|0\rangle - |1\rangle$, which is an eigenvector of

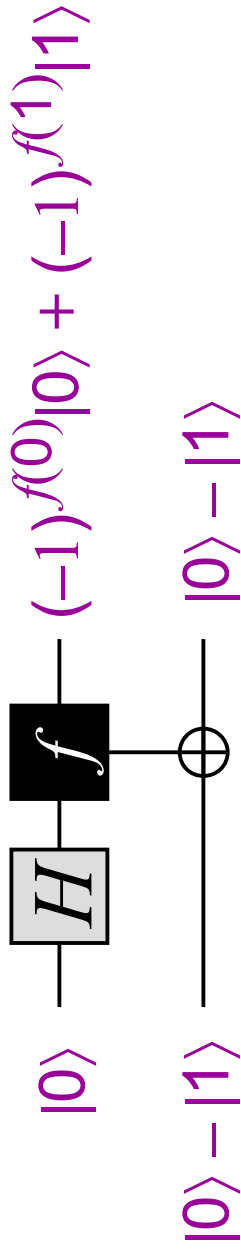
$$\begin{cases} \text{NOT with eigenvalue } -1 \\ I \text{ with eigenvalue } +1 \end{cases}$$

This causes f to induce a **phase shift** of $(-1)^{f(x)}$ to $|x\rangle$

$$\begin{array}{c} |x\rangle \text{ --- } \boxed{f} \text{ --- } (-1)^{f(x)}|x\rangle \\ |0\rangle - |1\rangle \text{ --- } \oplus \text{ --- } |0\rangle - |1\rangle \end{array}$$

Quantum algorithm (2)

2. Causes f to be queried *in superposition* (at $|0\rangle + |1\rangle$)



x	$f_1(x)$	x	$f_2(x)$	x	$f_3(x)$	x	$f_4(x)$
0	0	0	1	0	0	0	1
1	0	1	1	1	1	1	0

$\underbrace{\hspace{15em}}_{\pm(|0\rangle + |1\rangle)}$
 $\underbrace{\hspace{15em}}_{\pm(|0\rangle - |1\rangle)}$

Quantum algorithm (3)

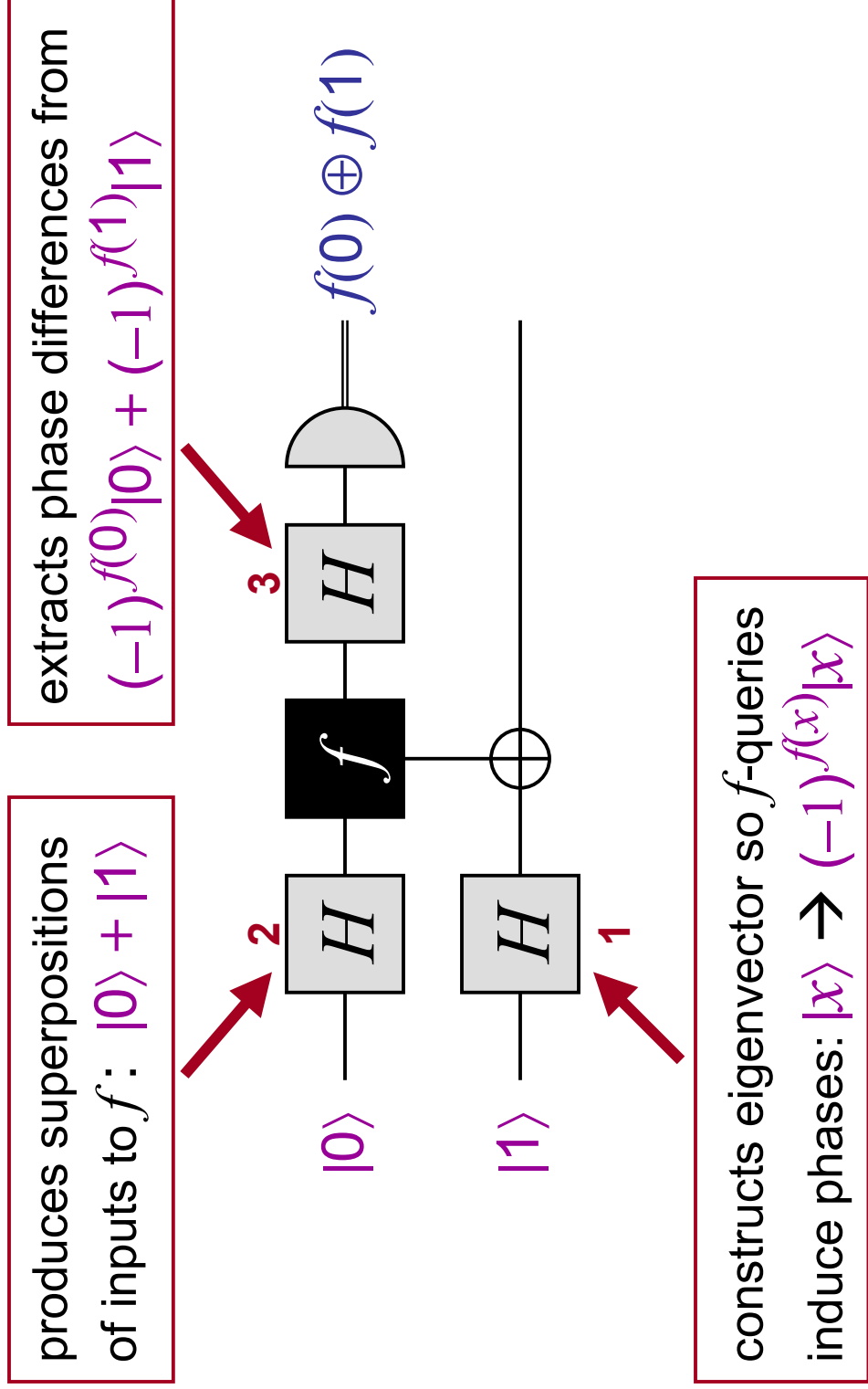
3. Distinguishes between $\pm(|0\rangle + |1\rangle)$ and $\pm(|0\rangle - |1\rangle)$

$$\pm(|0\rangle + |1\rangle) \xleftrightarrow{H} \pm|0\rangle$$

$$\pm(|0\rangle - |1\rangle) \xleftrightarrow{H} \pm|1\rangle$$

Summary of Deutsch's algorithm

Makes only one query, whereas two are needed classically



How to contact Richard Cleve

Office: DC 3524 (also at IQC)

Email: cleve@iqc.ca

Phone: 888-4567 x7762