EE5340

INTRODUCTION TO QUANTUM COMPUTING AND PHYSICAL BASICS OF COMPUTING

Quantum Algorithms

Ulya Karpuzcu

Classic Computing on Quantum Computers

- Replace all logic by an equivalent, consisting of reversible elements
  - CCN(Toffoli) gate
    - Can mimic NAND and FANOUT
  - Quantum CCN?

Toffoli Gate (CCN)

- (a,b,c) → (a,b,c ⊕ ab)
- Smallest universal reversible (classic) operation
  - Functionally complete (can mimic a NAND gate)

a	b	c	a'	b'	c'
0	0	0	0	0	0
0	0	1	0	0	1
0	1	0	0	1	0
0	1	1	0	1	1
1	0	0	1	0	0
1	0	1	1	0	1
1	1	0	1	1	1
1	1	1	1	1	0

Toffoli Gate (CCN)

- (a,b,c) → (a,b,c ⊕ ab)
- Smallest universal reversible (classic) operation
  - NAND: c =1, read c'

a	b	c	a'	b'	c'
0	0	0	0	0	0
0	0	1	0	0	1
0	1	0	0	1	0
0	1	1	0	1	1
1	0	0	1	0	0
1	0	1	1	0	1
1	1	0	1	1	1
1	1	1	1	1	0

Toffoli Gate (CCN)

- (a,b,c) → (a,b,c ⊕ ab)
- Smallest universal reversible (classic) operation
  - FANOUT: a=1, c=0, b'=c'=b (a'=1)

a	b	c	a'	b'	c'
0	0	0	0	0	0
0	0	1	0	0	1
0	1	0	0	1	0
0	1	1	0	1	1
1	0	0	1	0	0
1	0	1	1	0	1
1	1	0	1	1	1
1	1	1	1	1	0

Toffoli Gate (CCN)

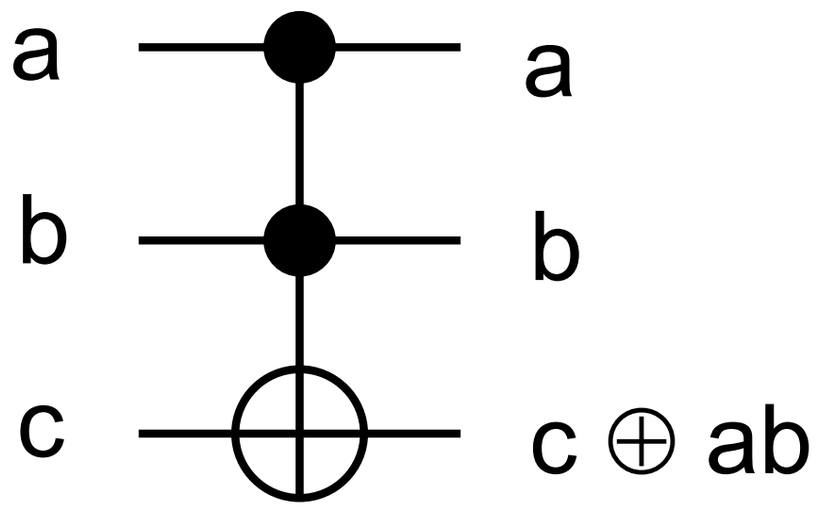
- (a,b,c) → (a,b,c ⊕ ab)
- Smallest universal reversible (classic) operation
  - Functionally complete (can mimic a NAND gate)

a	b	c	a'	b'	c'
0	0	0	0	0	0
0	0	1	0	0	1
0	1	0	0	1	0
0	1	1	0	1	1
1	0	0	1	0	0
1	0	1	1	0	1
1	1	0	1	1	1
1	1	1	1	1	0

$$|000\rangle \rightarrow |000\rangle; \quad |001\rangle \rightarrow |001\rangle; \quad |010\rangle \rightarrow |010\rangle; \quad |011\rangle \rightarrow |011\rangle$$
$$|100\rangle \rightarrow |100\rangle; \quad |101\rangle \rightarrow |101\rangle; \quad |110\rangle \rightarrow |111\rangle; \quad |111\rangle \rightarrow |110\rangle$$

Toffoli Gate (CCN)

- (a,b,c) → (a,b,c ⊕ ab)
- Smallest universal reversible (classic) operation
  - Functionally complete (can mimic a NAND gate)

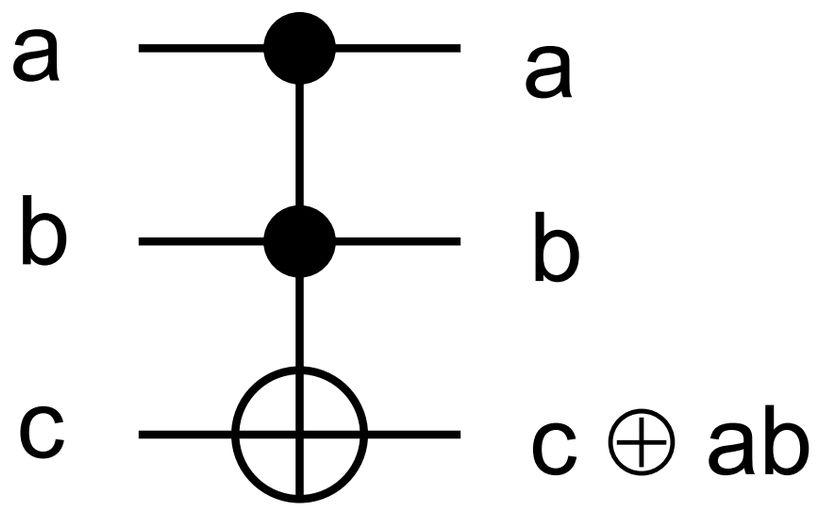


$$U_{toffoli} = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{pmatrix}$$

$$|000\rangle \rightarrow |000\rangle; \quad |001\rangle \rightarrow |001\rangle; \quad |010\rangle \rightarrow |010\rangle; \quad |011\rangle \rightarrow |011\rangle$$
$$|100\rangle \rightarrow |100\rangle; \quad |101\rangle \rightarrow |101\rangle; \quad |110\rangle \rightarrow |111\rangle; \quad |111\rangle \rightarrow |110\rangle$$

Toffoli Gate (CCN)

- (a,b,c) → (a,b,c ⊕ ab)
- Smallest universal reversible (classic) operation
  - Functionally complete (can mimic a NAND gate)



$$U_{toffoli} = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{pmatrix}$$

$|000\rangle \rightarrow |000\rangle$; $|001\rangle \rightarrow |001\rangle$; $|010\rangle \rightarrow |010\rangle$; $|011\rangle \rightarrow |011\rangle$
$|100\rangle \rightarrow |100\rangle$; $|101\rangle \rightarrow |101\rangle$; $|110\rangle \rightarrow |111\rangle$; $|111\rangle \rightarrow |110\rangle$

Classic Computing on Quantum Computers

- Replace all logic by an equivalent, consisting of reversible elements
  - CCN(Toffoli) gate
- Any deterministic classic computation can be mapped
- Non-deterministic classic computation?
  - I.e., random bits are generated to be used in computation
  - How to simulate the outcome of a random fair coin toss?
    - Prepare a qbit in state |0>
    - Send it through a Hadamard gate to produce

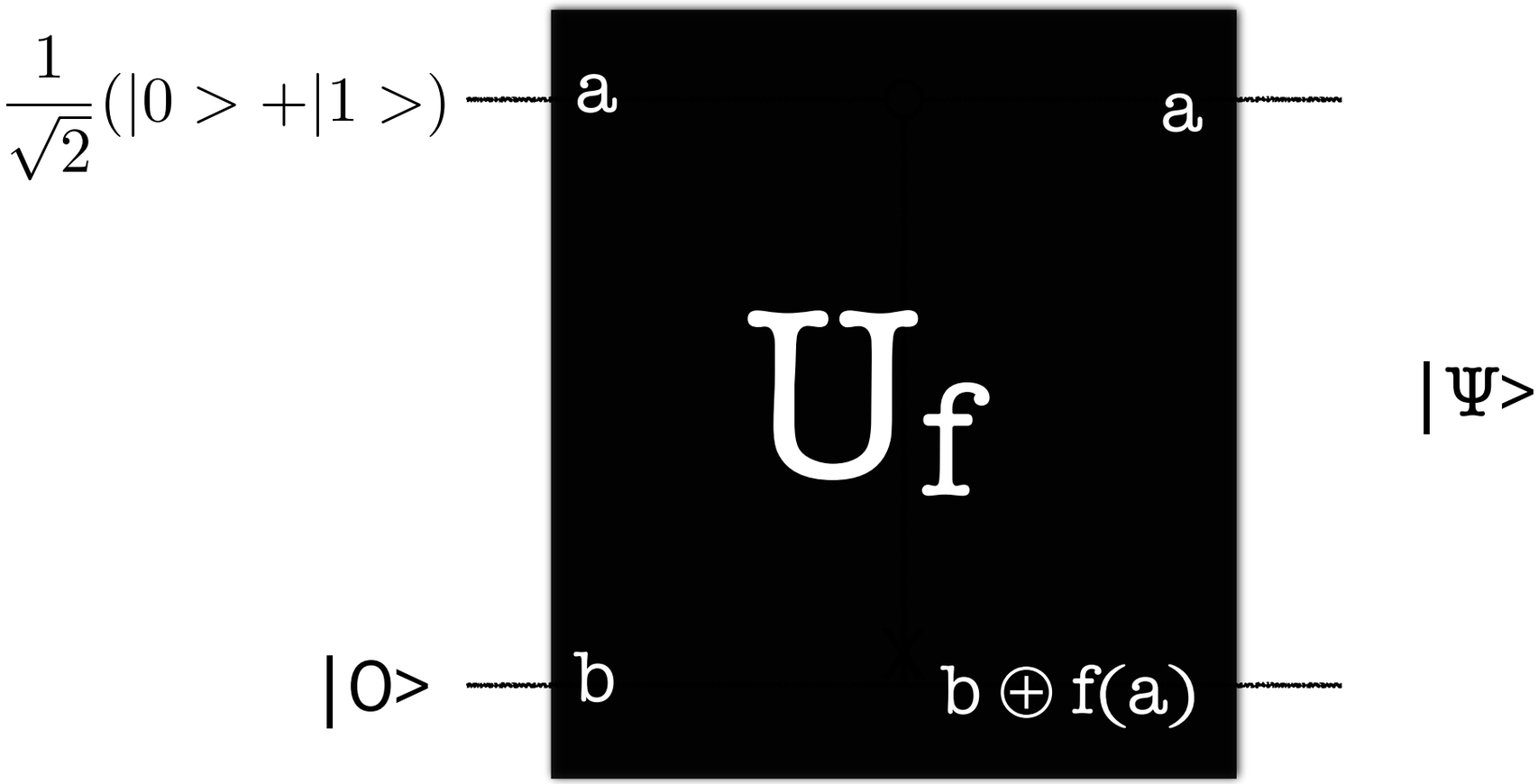
$$H|0>=\frac{1}{\sqrt{2}}|0>+\frac{1}{\sqrt{2}}|1>=\frac{1}{\sqrt{2}}(|0>+|1>)$$

- Measure the state: The result is 0 or 1 with 50/50 probability

Quantum ||ism

- f(x): {0,1} → {0,1}
- How to compute f(x) on a quantum computer?
  - Two-qbit computer, initial state: $|\Psi\rangle = |ab\rangle$
  - Sequence of unitary transformations: U_f
    - $|ab\rangle \rightarrow |a, b \oplus f(a)\rangle$
    - if b=0, second qbit state becomes $|f(a)\rangle$

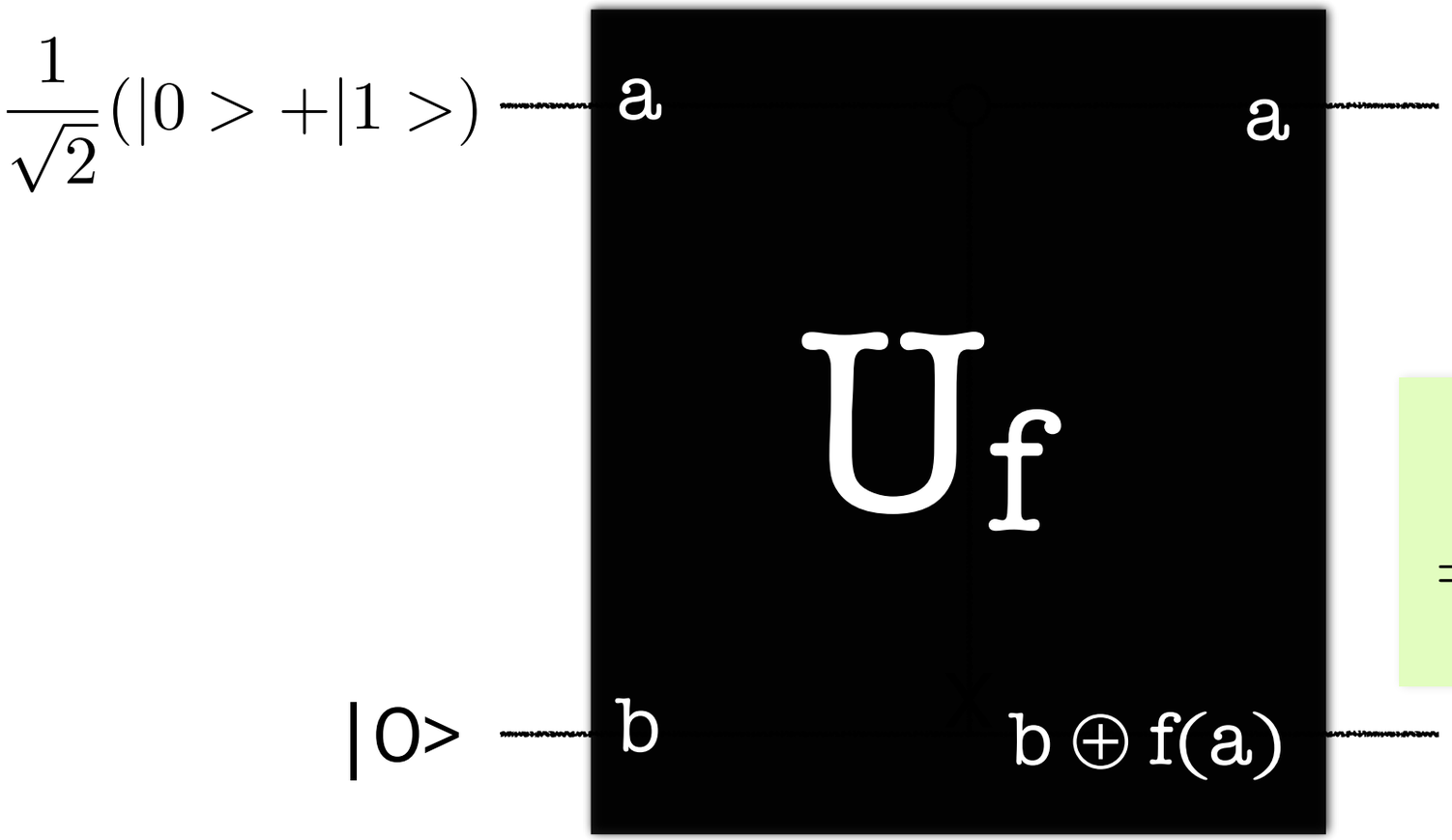
a	b	a	b ⊕ f(a)
0	0	0	f(a)=f(0)
0	1	0	not(f(a))
1	0	1	f(a)=f(1)
1	1	1	not(f(a))



Quantum ||ism

- f(x): {0,1} → {0,1}
- How to compute f(x) on a quantum computer?
  - Two-qbit computer, initial state: |Ψ> = |ab>
  - Sequence of unitary transformations: U_f
    - |ab> → |a, b ⊕ f(a)>
    - if b=0, second qbit state becomes |f(a) >

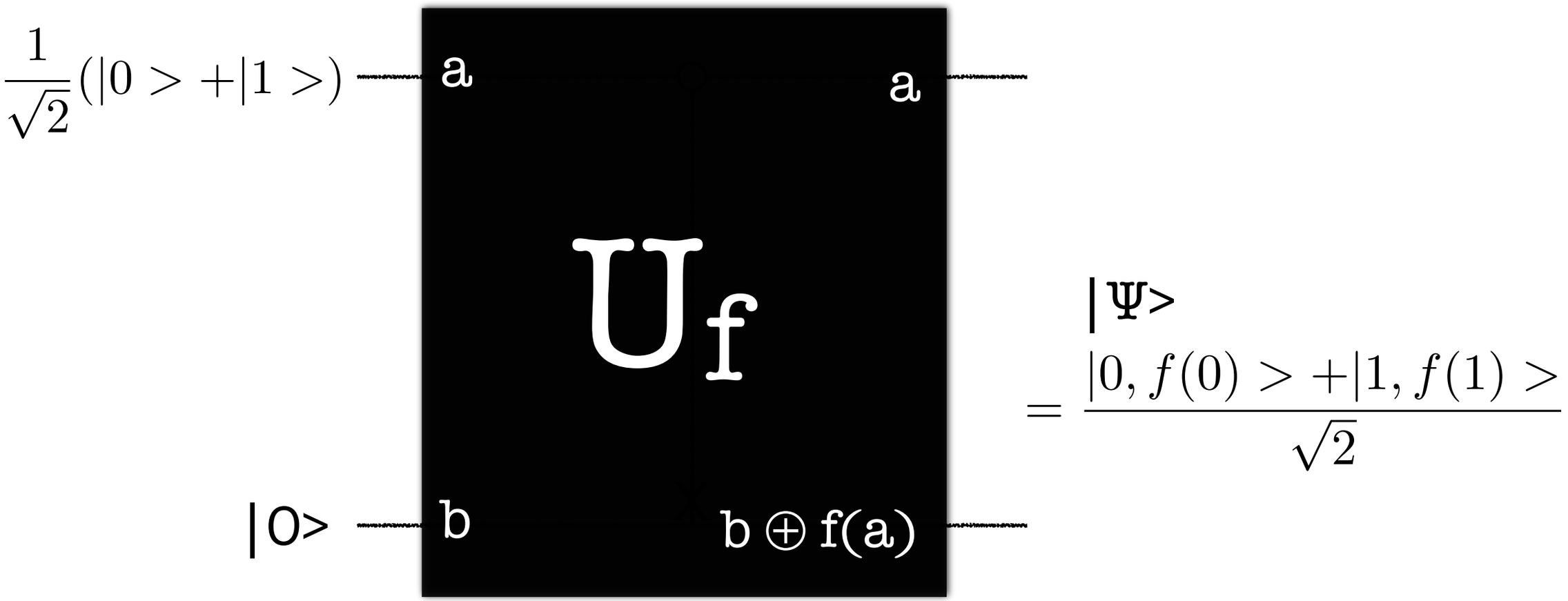
a	b	a	b ⊕ f(a)
0	0	0	f(a)=f(0)
0	1	0	not(f(a))
1	0	1	f(a)=f(1)
1	1	1	not(f(a))



$$= \frac{|\Psi>}{\frac{|0, f(0)> + |1, f(1)>}{\sqrt{2}}}$$

Quantum ||ism

- Classic ||ism
  - **Multiple** f(x) circuits evaluate f(x) for a **single** value of x simultaneously
- Quantum ||ism
  - **Single** f(x) circuit evaluates f(x) for **multiple** values of x simultaneously



Quantum ||ism: Multi-bit generalization

- Function f of arbitrary number of bits?
- Hadamard transform
  - n Hadamard gates acting simultaneously on n qbits
  - n = 2 → $H^{\otimes 2}$

$$\frac{|0>+|1>}{\sqrt{2}}\frac{|0>+|1>}{\sqrt{2}} = \frac{|00>+|01>+|10>+|11>}{2}$$

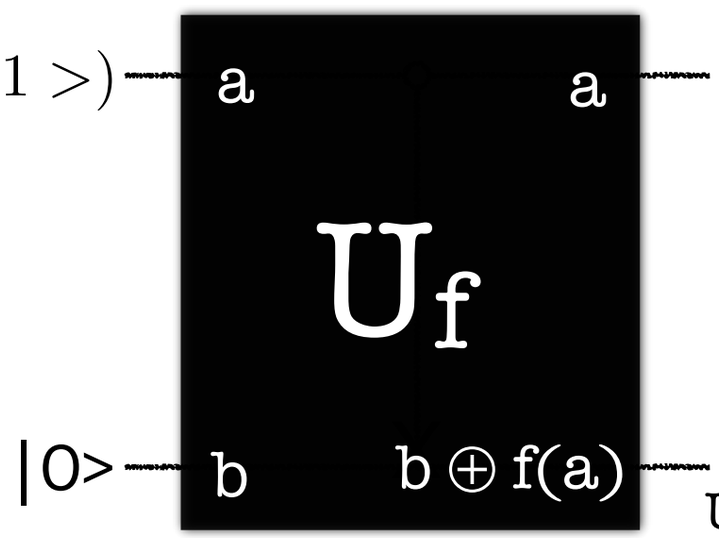
- Hadamard transform $H^{\otimes n}$ on n qbits all initialized to |0>: $\frac{1}{\sqrt{2^n}}\sum_x |x>$
  - Sum over all possible values of x
  - Equal superposition of all computational basis states
    - Superposition of 2^n states using n gates

Quantum ||ism: Multi-bit generalization

- Hadamard transform $H^{\oplus n}$ on n qbits all initialized to |0>: $\frac{1}{\sqrt{2^n}} \sum_x |x>$
  - Sum over all possible values of x
  - Equal superposition of all computational basis states
    - Superposition of 2^n states using n gates

- Quantum || evaluation of f(x) with n-bit input, single-bit output:
  - Prepare the n+1 qbit state $|0>^{\otimes n} |0>$
  - Apply Hadamard transformation on the first n qbits
  - Apply (multi-qbit) U_f on the resulting n qbits
  - |Ψ> becomes $\frac{1}{\sqrt{2^n}} \sum_x |x> |f(x)>$

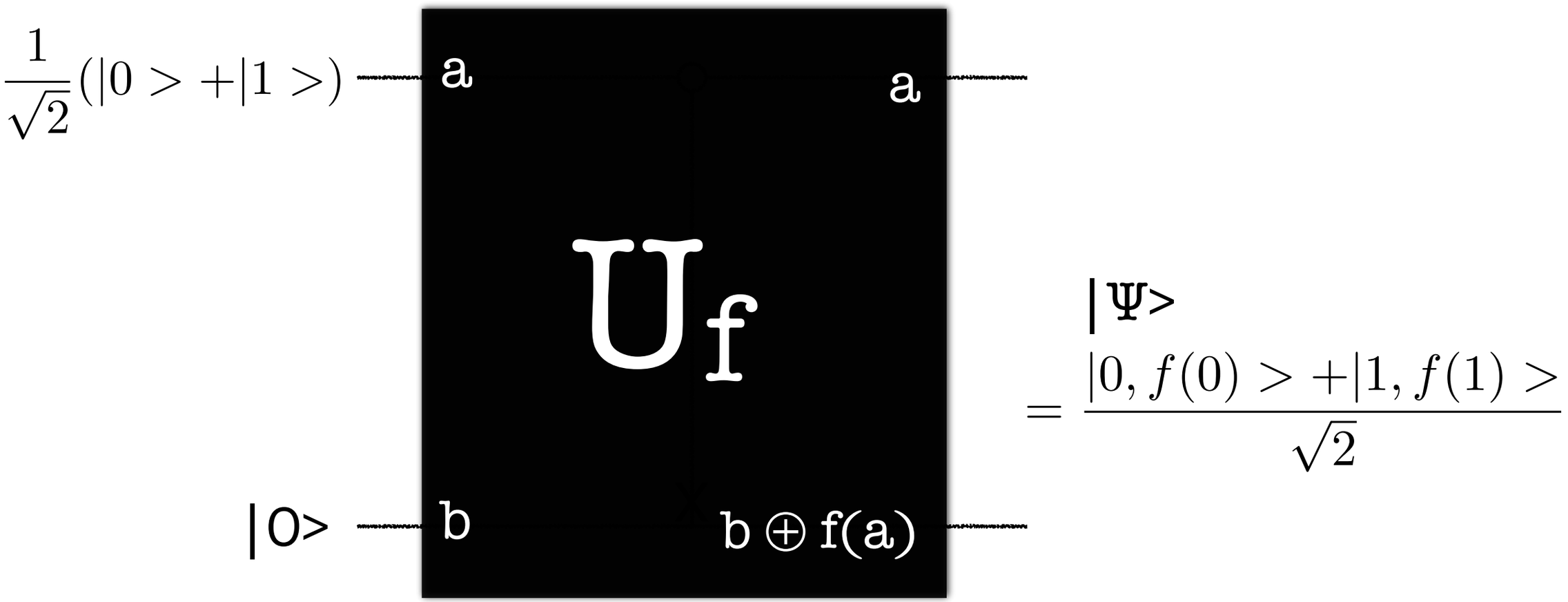
Single-bit: $\frac{1}{\sqrt{2}}(|0> + |1>)$



$$|\Psi> = \frac{|0, f(0)> + |1, f(1)>}{\sqrt{2}}$$

Quantum ||ism

- Classic ||ism
  - **Multiple** f(x) circuits evaluate f(x) for a **single** value of x simultaneously
- Quantum ||ism
  - **Single** f(x) circuit evaluates f(x) for **multiple** values of x simultaneously



Quantum ||ism

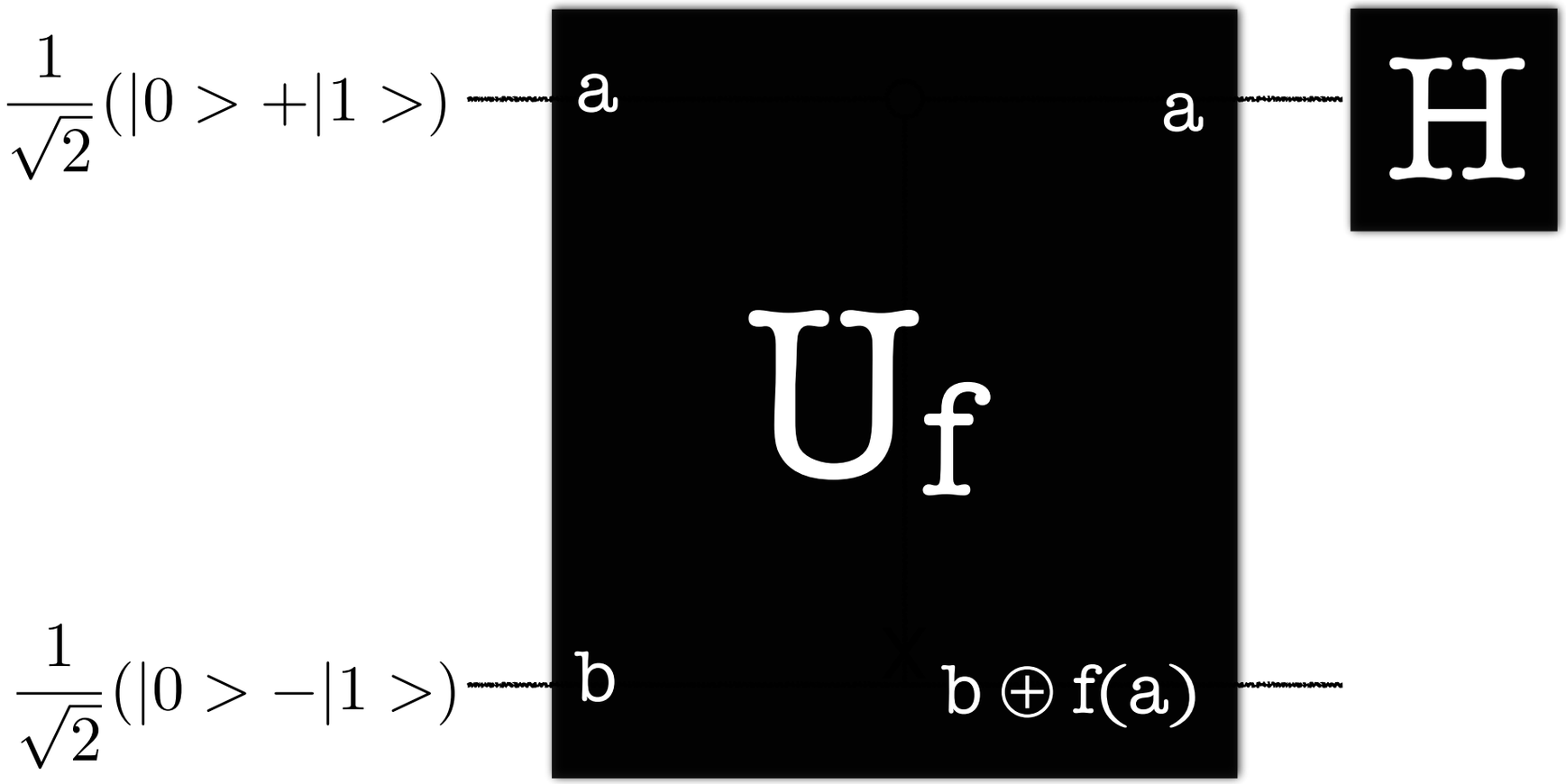
$$\frac{|0, f(0)> + |1, f(1)>}{\sqrt{2}}$$

$$\frac{1}{\sqrt{2^n}} \sum_x |x> |f(x)>$$

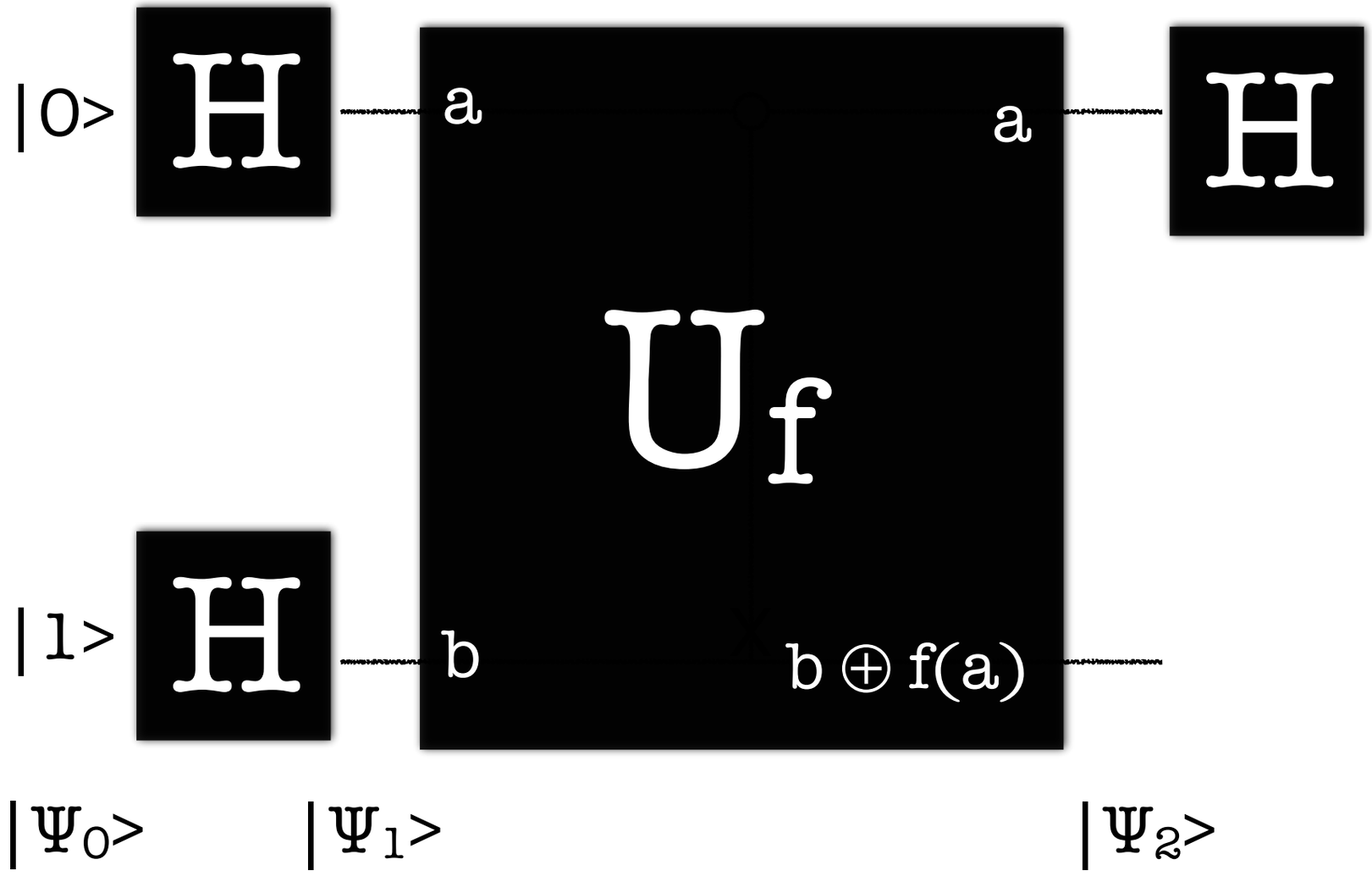
- Single qbit
  - Measurement of the output state renders either |0, f(0)> or |1, f(1)>, not both.
- Multi-qbit
  - Measurement of the output state renders a single value of f(x), not all.

Not useful unless we can extract information about more than one value of f(x) from superposition states.

Deutsch's Algorithm

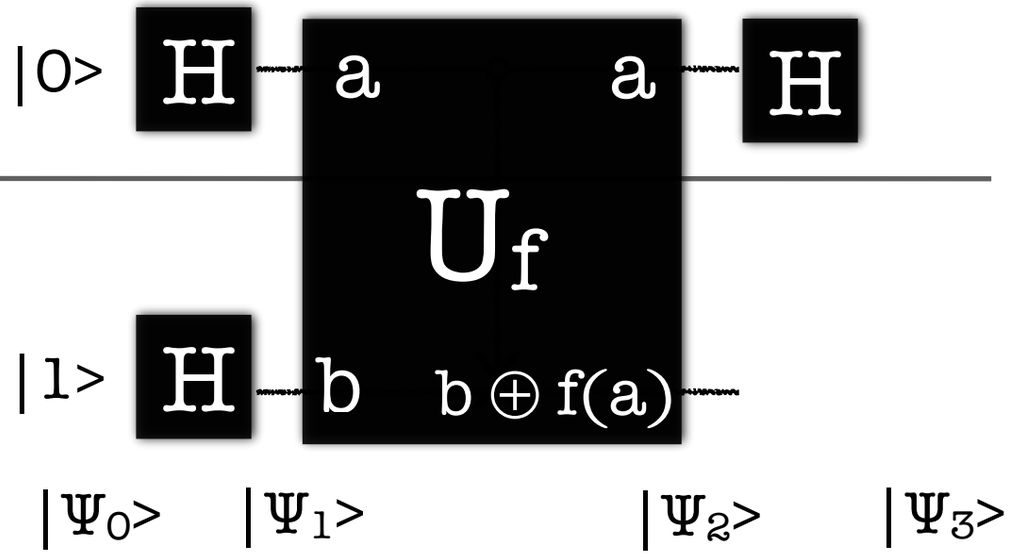


Deutsch's Algorithm



Deutsch's Algorithm

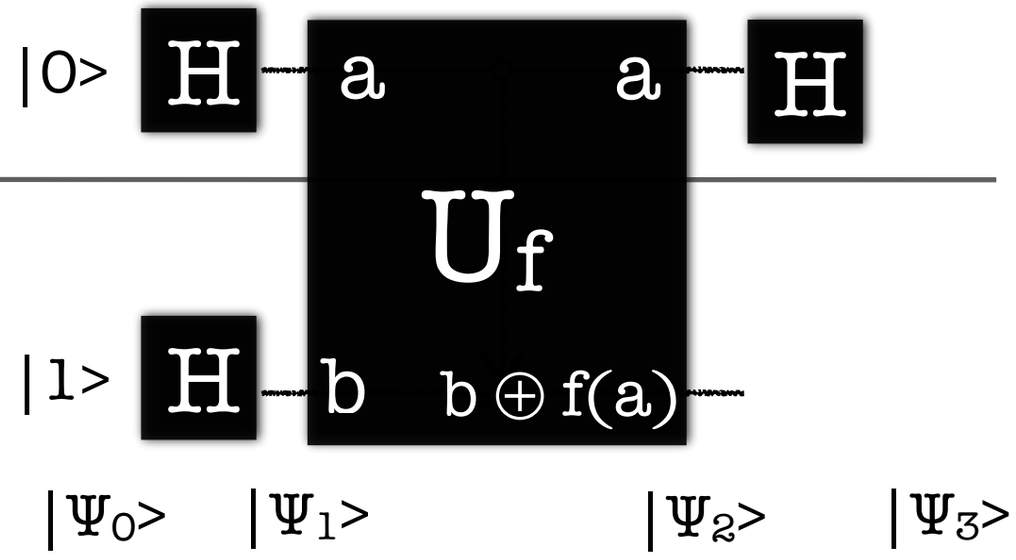
$$|\Psi_0 >= |01 >$$



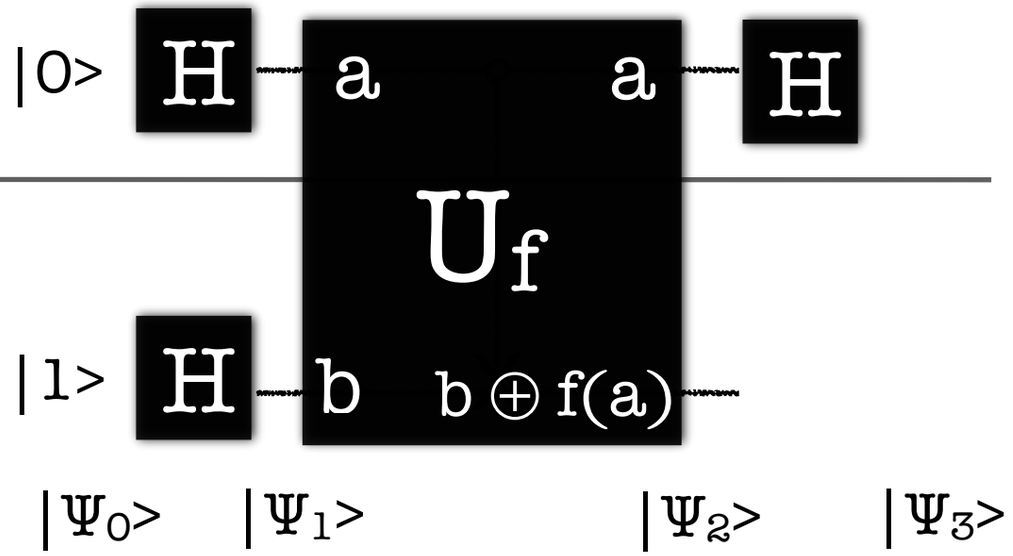
Deutsch's Algorithm

$$|\Psi_0> = |01>$$

$$|\Psi_1> = [\frac{|0> + |1>}{\sqrt{2}}][\frac{|0> - |1>}{\sqrt{2}}]$$



Deutsch's Algorithm



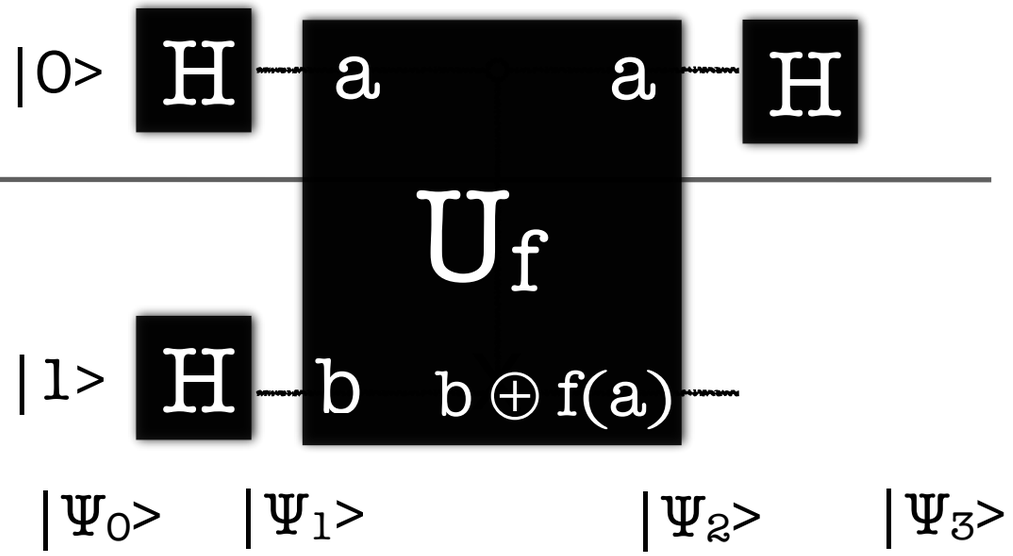
$$|\Psi_0>=|01>$$

$$|\Psi_1>=[\frac{|0>+|1>}{\sqrt{2}}][\frac{|0>-|1>}{\sqrt{2}}]$$

U_f on state $|a>[\frac{|0>-|1>}{\sqrt{2}}]$?

a	b	a	b ⊕ f(a)
0	0	0	f(a)
0	1	0	not(f(a))
1	0	1	f(a)
1	1	1	not(f(a))

Deutsch's Algorithm



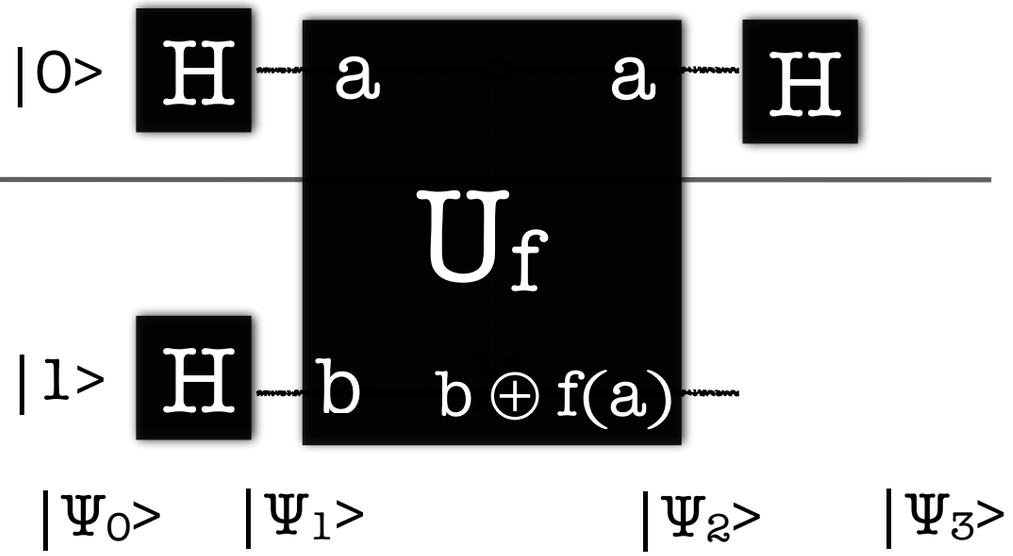
$$|\Psi_0>=|01>$$

$$|\Psi_1>=[\frac{|0>+|1>}{\sqrt{2}}][\frac{|0>-|1>}{\sqrt{2}}]$$

U_f on state $|a>[\frac{|0>-|1>}{\sqrt{2}}]$? $(-1)^{f(a)}|a>[\frac{|0>-|1>}{\sqrt{2}}]$

a	b	a	b ⊕ f(a)
0	0	0	f(a)
0	1	0	not(f(a))
1	0	1	f(a)
1	1	1	not(f(a))

Deutsch's Algorithm



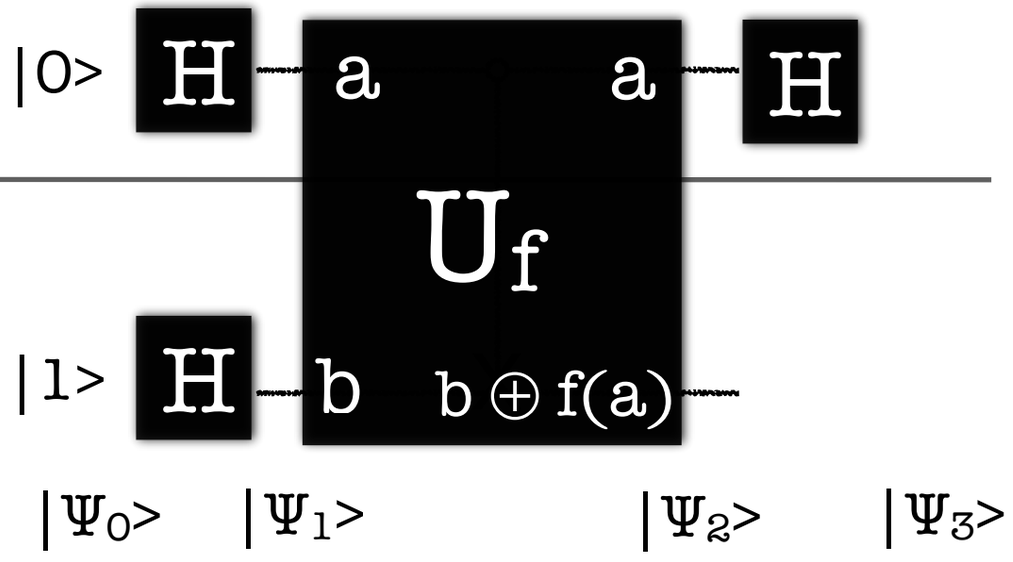
$$|\Psi_0>=|01>$$

$$|\Psi_1>=[\frac{|0>+|1>}{\sqrt{2}}][\frac{|0>-|1>}{\sqrt{2}}]$$

U_f on state $|a>[\frac{|0>-|1>}{\sqrt{2}}]$? $(-1)^{f(a)}|a>[\frac{|0>-|1>}{\sqrt{2}}]$

$$|\Psi_2>=\begin{cases}\pm[\frac{|0>+|1>}{\sqrt{2}}][\frac{|0>-|1>}{\sqrt{2}}] & if \quad f(0)=f(1)\\ \pm[\frac{|0>-|1>}{\sqrt{2}}][\frac{|0>-|1>}{\sqrt{2}}] & if \quad f(0)\neq f(1)\end{cases}$$

Deutsch's Algorithm



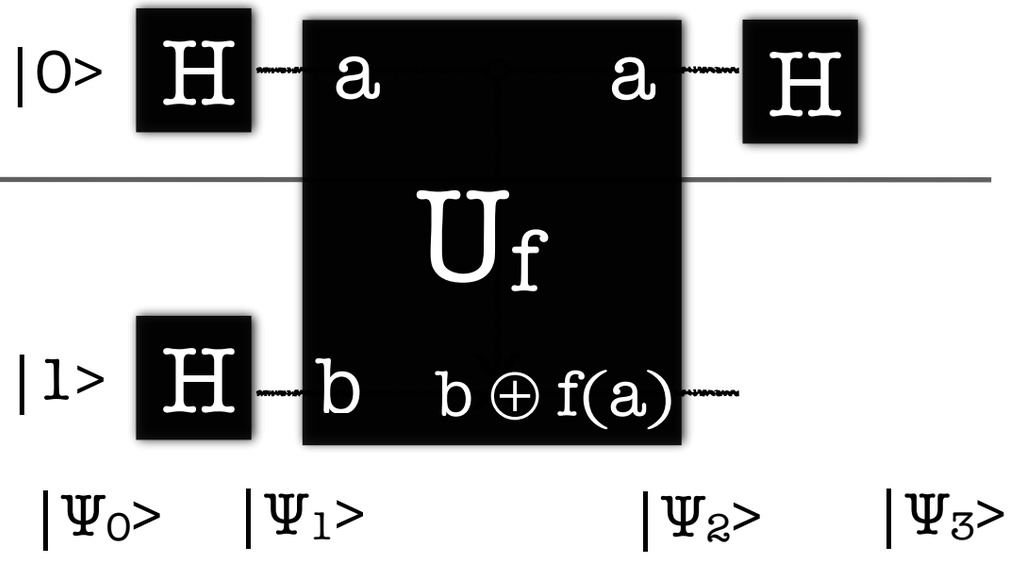
$$|\Psi_0>=|01>$$

$$|\Psi_1>=[\frac{|0>+|1>}{\sqrt{2}}][\frac{|0>-|1>}{\sqrt{2}}]$$

$$|\Psi_2>=\begin{cases}\pm[\frac{|0>+|1>}{\sqrt{2}}][\frac{|0>-|1>}{\sqrt{2}}] & if \quad f(0)=f(1)\\ \pm[\frac{|0>-|1>}{\sqrt{2}}][\frac{|0>-|1>}{\sqrt{2}}] & if \quad f(0)\neq f(1)\end{cases}$$

$$|\Psi_3>=\begin{cases}\pm|0>[\frac{|0>-|1>}{\sqrt{2}}] & if \quad f(0)=f(1)\\ \pm|1>[\frac{|0>-|1>}{\sqrt{2}}] & if \quad f(0)\neq f(1)\end{cases}$$

Deutsch's Algorithm



$$|\Psi_0>=|01>$$

$$|\Psi_1>=[\frac{|0>+|1>}{\sqrt{2}}][\frac{|0>-|1>}{\sqrt{2}}]$$

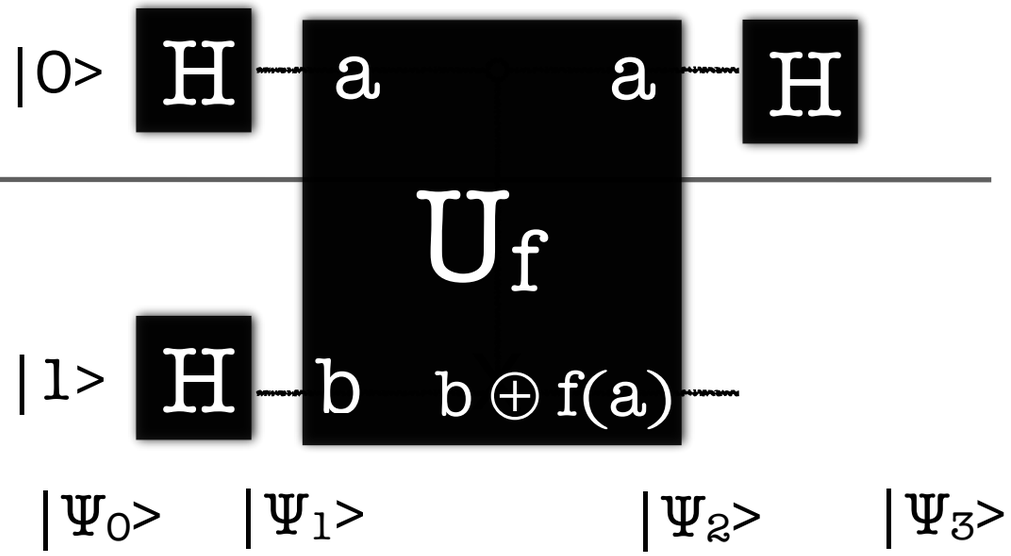
$$|\Psi_2>=\begin{cases}\pm[\frac{|0>+|1>}{\sqrt{2}}][\frac{|0>-|1>}{\sqrt{2}}] & if \quad f(0)=f(1)\\ \pm[\frac{|0>-|1>}{\sqrt{2}}][\frac{|0>-|1>}{\sqrt{2}}] & if \quad f(0)\neq f(1)\end{cases}$$

$$|\Psi_3>=\begin{cases}\pm|0>[\frac{|0>-|1>}{\sqrt{2}}] & if \quad f(0)=f(1)\\ \pm|1>[\frac{|0>-|1>}{\sqrt{2}}] & if \quad f(0)\neq f(1)\end{cases}$$

$$|\Psi_3>=\pm|f(0)\oplus f(1)>[\frac{|0>-|1>}{\sqrt{2}}]$$

Deutsch's Algorithm

$$|\Psi_3 >= \pm|f(0) \oplus f(1) > [\frac{|0> - |1>}{\sqrt{2}}]$$



- By measuring the first qbit, we determine $f(0) \oplus f(1)$
  - A global property of f(x), using only one evaluation of f(x)
  - Classic equivalent requires at least 2 evaluations

Deutsch-Jozsa Algorithm

- Problem statement:
  - A(lice) in Amsterdam selects a number x from 0–2^n-1, and mails it in a letter to B(ob) in Boston
  - B calculates some function on x, which returns 0 or 1, and replies with the result
  - B can only use one of two types of functions
    - Constant for all x
    - Balanced: Equal to 1 for exactly half of the possible x
  - A's task is to determine which type of function B picked with minimal communication

Deutsch-Jozsa Algorithm

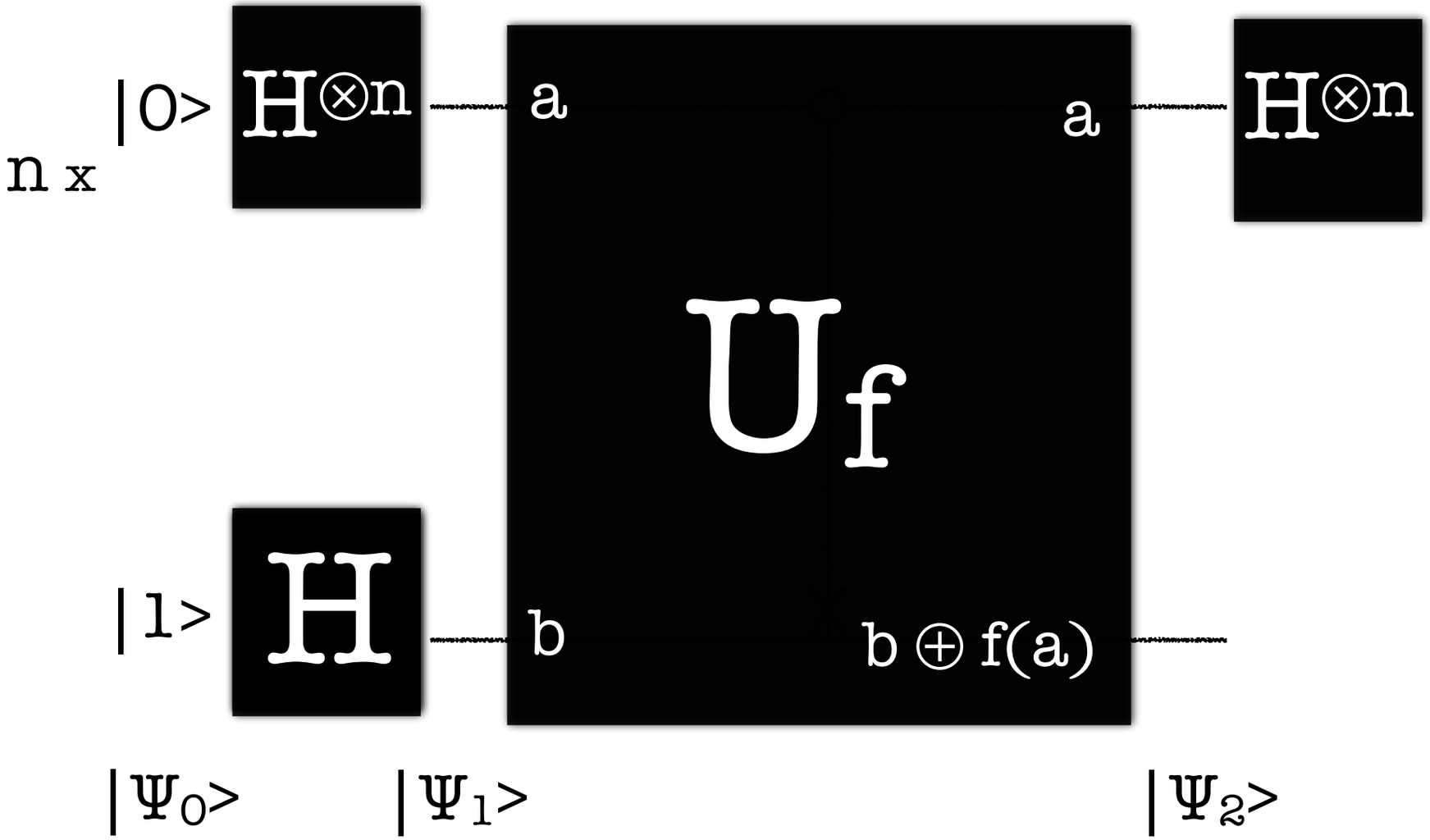
- Problem statement:
  - Alice in Amsterdam selects a number x from $0-2^n-1$, and mails it in a letter to Bob in Boston
  - Bob calculates some function on x, which returns 0 or 1, and replies with the result
  - Bob can only use one of two types of functions
    - Constant for all x
    - Balanced: Equal to 1 for exactly half of the possible x
  - Alice's task is to determine which type of function Bob picked with minimal communication
- Classic case
  - Alice may need to send one value of x per letter
    - At least $2^n/2+1$ queries in the worst case
    - May receive $2^n/2$ 0's before receiving the first 1 if the function was balanced
    - In each query, Alice sends n bits of information

Deutsch-Jozsa Algorithm

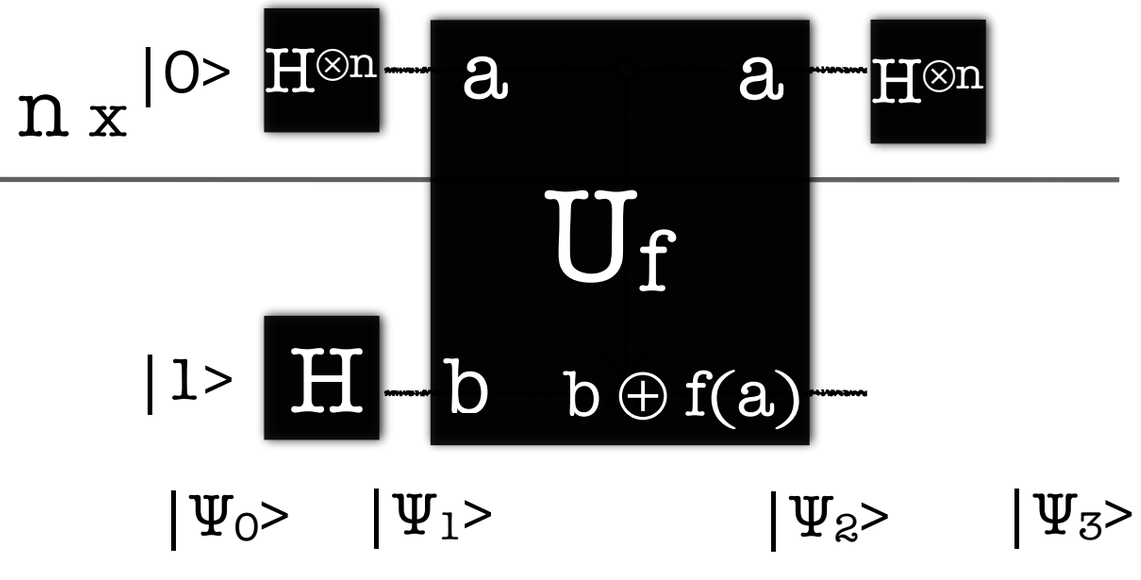
- Problem statement:
  - Alice in Amsterdam selects a number x from 0–2^n-1, and mails it in a letter to Bob in Boston
  - Bob calculates some function on x, which returns 0 or 1, and replies with the result
  - Bob can only use one of two types of functions
    - Constant for all x
    - Balanced: Equal to 1 for exactly half of the possible x
  - Alice's task is to determine which type of function Bob picked with minimal communication
- Classic case
  - Alice may need to send one value of x per letter
    - At least $2^n/2+1$ queries in the worst case
    - May receive $2^n/2$ 0's before receiving the first 1 if the function was balanced
    - In each query, Alice sends n bits of information
- Quantum case: one query may suffice
  - Alice and Bob can exchange quantum bits
  - Bob agrees to calculate the function using a unitary transformation

Deutsch-Jozsa Algorithm

- Alice has an n qbit query register and a single qbit answer register
- Alice prepares the register values in a superposition state
- Bob evaluates the function using quantum parallelism, and stores the result in the answer register
- Alice interferes the states (of the query register) in the super-position using Hadamard transform, and performs a suitable measurement to determine the type of the function



Deutsch-Jozsa



$$\Psi_0 = |0>^{\otimes n} \; |1>$$

$$\Psi_1 = \sum_{x \in \{0,1\}^n} \frac{|x>}{\sqrt{2^n}} [\frac{|0> - |1>}{\sqrt{2}}]$$

Query register keeps the superposition of all possible values

Answer register keeps the 50-50(%) superposition of 0 and 1

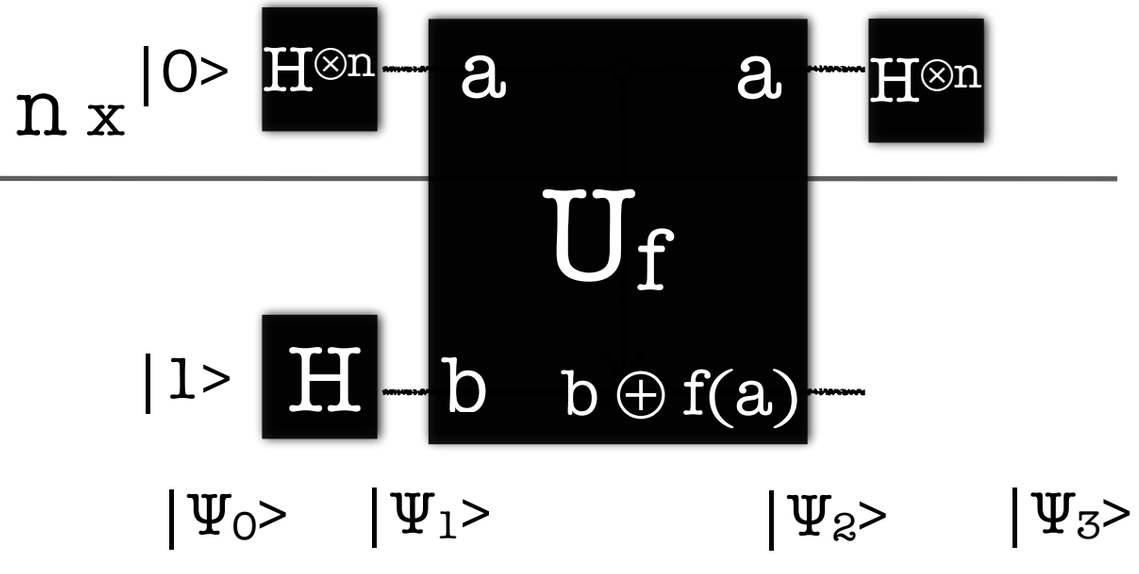
Cheat-Sheet

- Hadamard transform
  - n Hadamard gates acting simultaneously on n qbits
  - $n = 2 \rightarrow H^{\otimes 2}$

$$\frac{|0> + |1>}{\sqrt{2}} \frac{|0> + |1>}{\sqrt{2}} = \frac{|00> + |01> + |10> + |11>}{2}$$

- Hadamard transform $H^{\otimes n}$ on n qbits all initialized to |0>: $\frac{1}{\sqrt{2^n}} \sum_x |x>$
  - Sum over all possible values of x
- Equal superposition of all computational basis states
  - Superposition of 2^n states using n gates

Deutsch-Jozsa



$$\Psi_0 = |0>^{\otimes n} |1>$$

$$\Psi_1 = \sum_{x \in \{0,1\}^n} \frac{|x>}{\sqrt{2^n}} [\frac{|0> - |1>}{\sqrt{2}}]$$

Query register keeps the superposition of all possible values

Answer register keeps the 50-50(%) superposition of 0 and 1

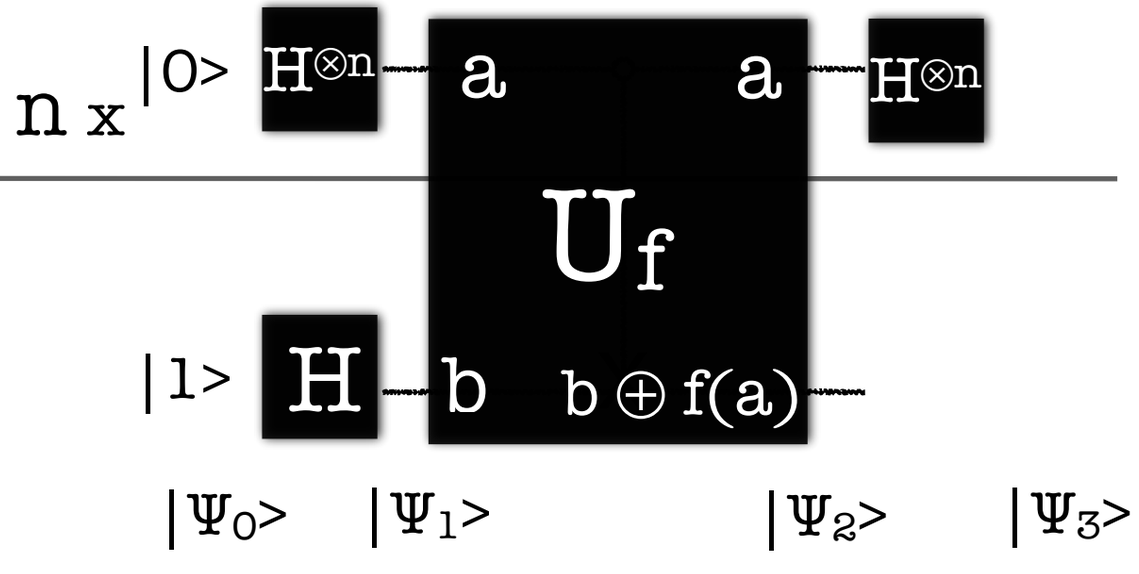
Bob evaluates the function next:

$$\Psi_2 = \sum_x \frac{(-1)^{f(x)}|x>}{\sqrt{2^n}} [\frac{|0> - |1>}{\sqrt{2}}]$$

U_f on state $|a> [\frac{|0> - |1>}{\sqrt{2}}]$?

$(-1)^{f(a)}|a> [\frac{|0> - |1>}{\sqrt{2}}]$

Deutsch-Jozsa



$$\Psi_0 = |0>^{\otimes n} |1>$$

$$\Psi_1 = \sum_{x \in \{0,1\}^n} \frac{|x>}{\sqrt{2^n}} [\frac{|0> - |1>}{\sqrt{2}}]$$

Query register keeps the superposition of all possible values

Answer register keeps the 50-50(%) superposition of 0 and 1

Bob evaluates the function next:

$$\Psi_2 = \sum_x \frac{(-1)^{f(x)}|x>}{\sqrt{2^n}} [\frac{|0> - |1>}{\sqrt{2}}]$$

Alice applies H transform on the query register

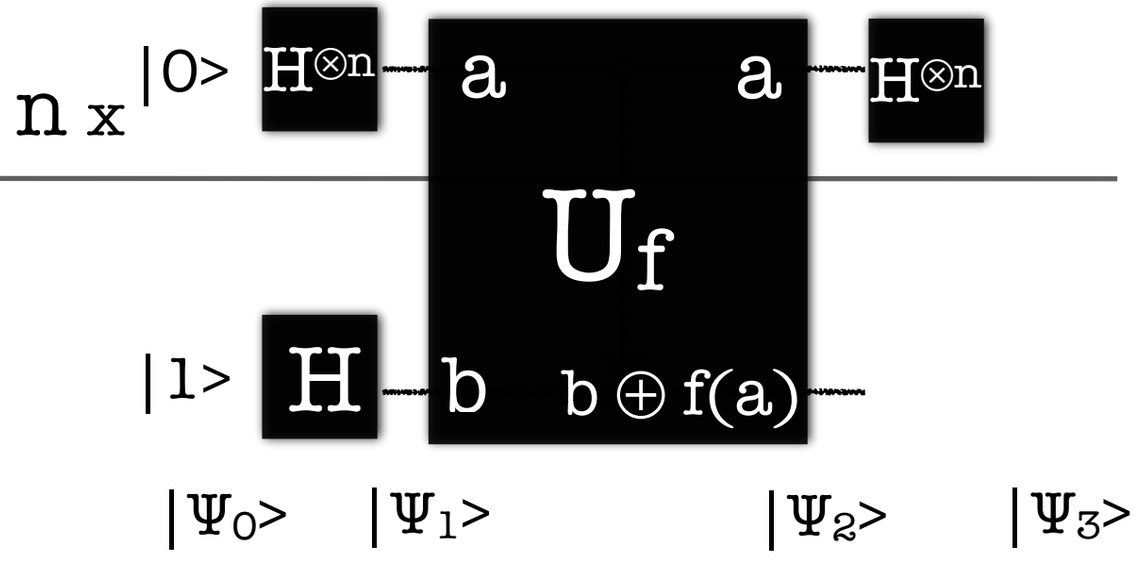
Cheat-Sheet

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

$$H^{\otimes n}|x_1...x_n>=H|x>=\sum_{z_1...z_n}\frac{(-1)^{x_1z_1+...+x_nz_n}|z_1...z_n>}{\sqrt{2^n}}$$

$$H^{\otimes n}|x>=H|x>=\sum_{z}\frac{(-1)^{x\bullet z}|z_1...z_n>}{\sqrt{2^n}}$$

Deutsch-Jozsa



$$\Psi_0 = |0>^{\otimes n} |1>$$

$$\Psi_1 = \sum_{x \in \{0,1\}^n} \frac{|x>}{\sqrt{2^n}} [\frac{|0> - |1>}{\sqrt{2}}]$$

Query register keeps the superposition of all possible values

Answer register keeps the 50-50(%) superposition of 0 and 1

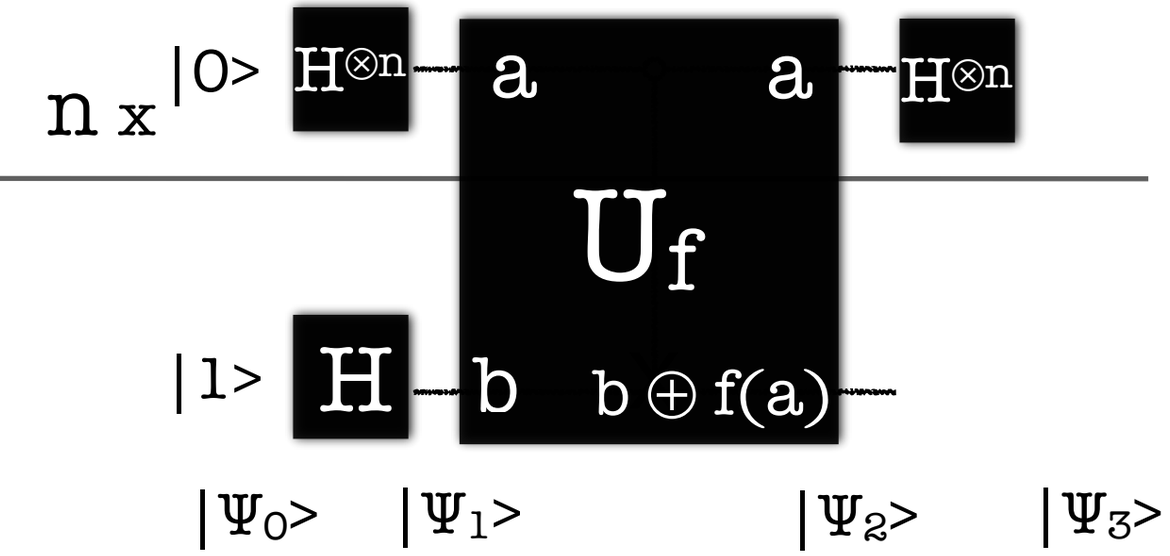
Bob evaluates the function next:

$$\Psi_2 = \sum_x \frac{(-1)^{f(x)}|x>}{\sqrt{2^n}} [\frac{|0> - |1>}{\sqrt{2}}]$$

Alice applies H transform on the query register

$$\Psi_3 = \sum_z \sum_x \frac{(-1)^{x \bullet z + f(x)}|z>}{2^n} [\frac{|0> - |1>}{\sqrt{2}}]$$

Deutsch-Jozsa

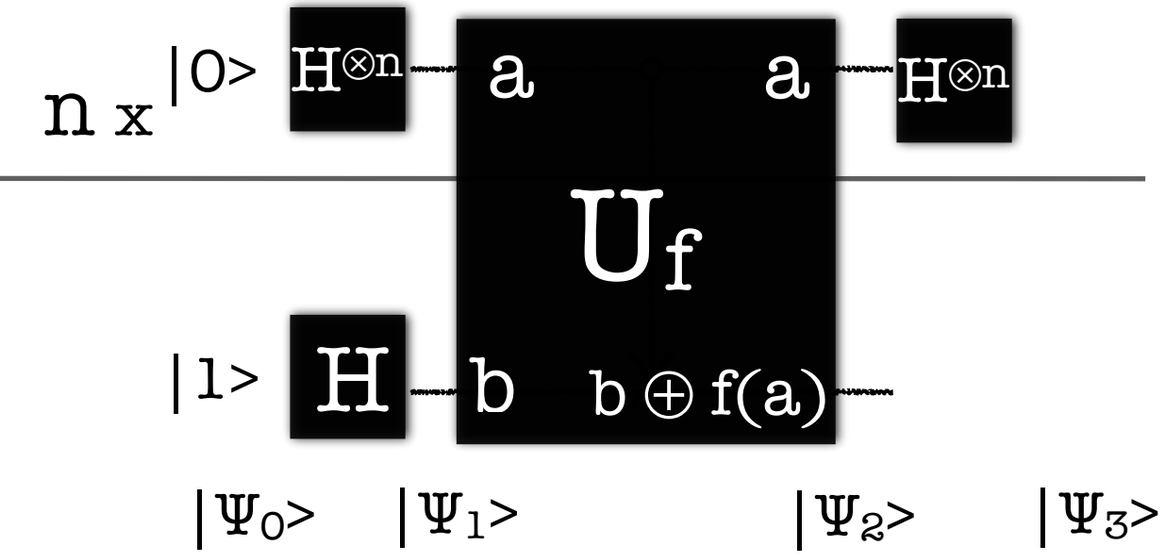


Alice applies H transform on the query register

$$\Psi_3 = \sum_z \sum_x \frac{(-1)^{x \bullet z + f(x)} |z>}{2^n} [\frac{|0> - |1>}{\sqrt{2}}]$$

Alice observes the query register …

Deutsch-Jozsa



Alice applies H transform on the query register

$$\Psi_3 = \sum_z \sum_x \frac{(-1)^{x \bullet z + f(x)} |z>}{2^n} [\frac{|0> - |1>}{\sqrt{2}}]$$

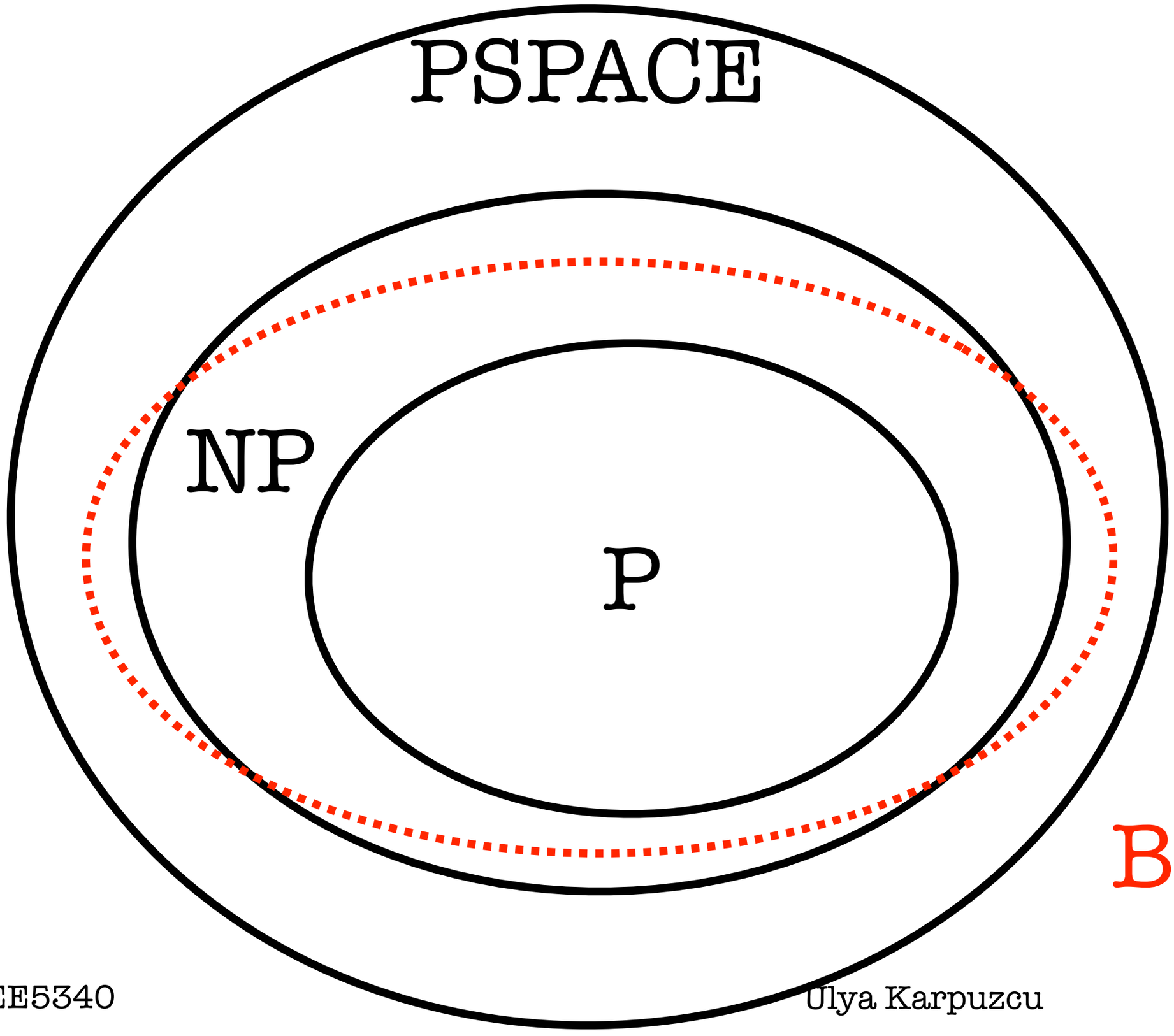
Alice observes the query register ...

The probability amplitude of state $|0>^{\otimes n}$ $[\sum_x \frac{(-1)^{f(x)}}{2^n}]^2$

f(x) constant: prob.(00...0...0) = 1

f(x) balanced: prob.(00...0...0) = 0

How powerful are quantum computers?



Bibliography

- Feynman Lectures on Computation, Chapter V
- Metodi et al., Quantum Computing for Computer Architects
- Nielsen and Chuang, Quantum Computation and Quantum Information

EE5340

INTRODUCTION TO QUANTUM COMPUTING AND PHYSICAL BASICS OF COMPUTING

Quantum Algorithms

Ulya Karpuzcu