WACQT | Wallenberg Centre for Quantum Technology

Introduction to quantum algorithms

Giulia Ferrini | WACQT

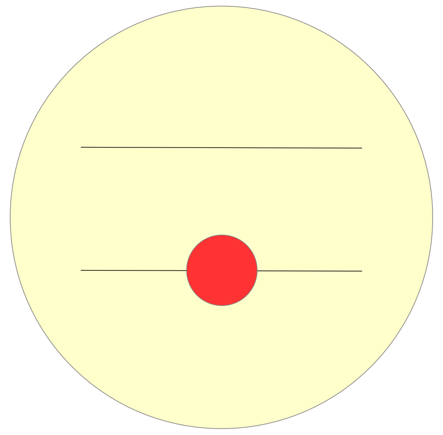
Outline

- Intro: what is a quantum computer
- How do we program a quantum computer? Universal gate sets and notion of universality
- Models of quantum computation
- An example of a quantum algorithm: Deutsch-Jozsa
- What is the status on quantum algorithms? Use cases: quantum algorithms to solve useful problems?
- Quantum algorithms at Chalmers / in WACQT

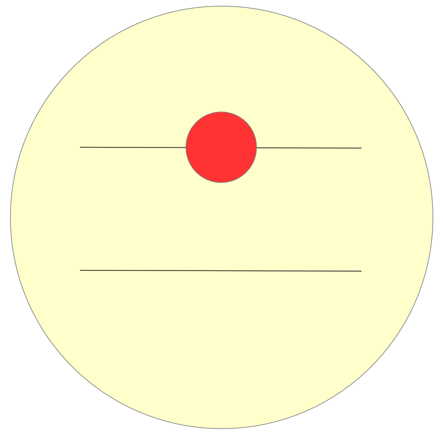
Introduction: what is a quantum computer?

- Quantum system with 2 addressable states (qubit)

State $|0\rangle$



State $|1\rangle$



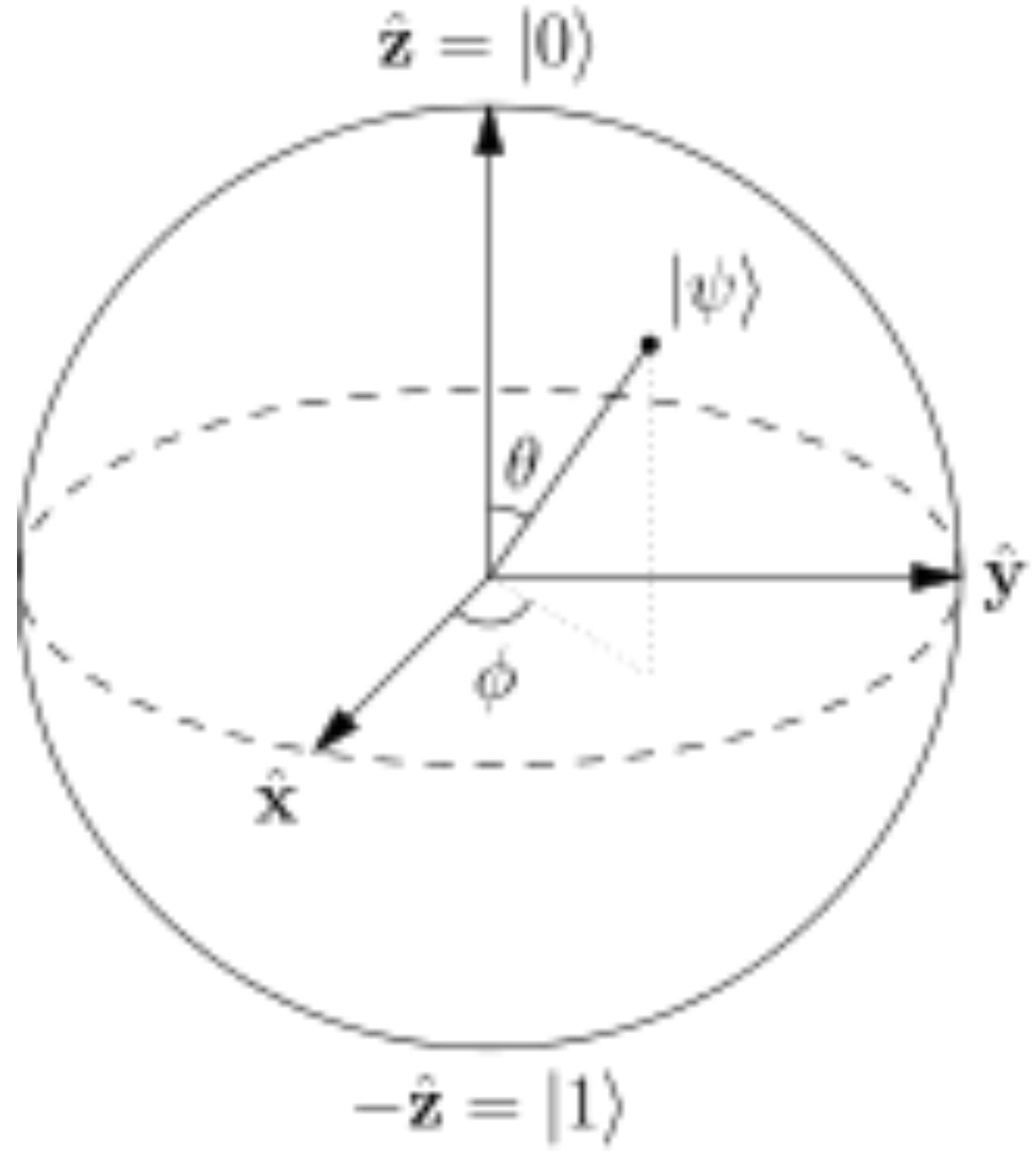
- Arbitrary superpositions are possible

$$|\psi\rangle = \alpha|0\rangle + \beta|1\rangle \qquad |\alpha|^2 + |\beta|^2 = 1$$

- Operations move the state of the qubit around the Bloch sphere

- The state is finally read-out by measurement

Bloch sphere



What is a quantum computer? DiVicenzo criteria

- Constructing a quantum computer requires that the experimental setup meet the following conditions (DiVicenzo, 2000):

1. A scalable physical system with well characterized qubit
2. The ability to initialize the state of the qubits to a simple fiducial state
3. Long relevant decoherence times
4. A "universal" set of quantum gates
5. A qubit-specific measurement capability

Why do we want to build one?

- Theoretical prediction : quantum computers should allow for solving some computational task efficiently, while hard for normal computers !

Why do we want to build one?

- Theoretical prediction : quantum computers should allow for solving some computational task efficiently, while hard for normal computers !

  - Efficient = takes a polynomial time t(L) in the input size L

  - Hard = takes an exponential time t(L) in the input size L

Why do we want to build one?

WACQT | Wallenberg Centre for Quantum Technology

- Theoretical prediction : quantum computers should allow for solving some computational task efficiently, while hard for normal computers !
  - Efficient = takes a polynomial time t(L) in the input size L
  - Hard = takes an exponential time t(L) in the input size L

E.g: Factoring:
15 = 5x3

10433 x 16453 = ? (easy)

? x ? = 171654149 (hard)

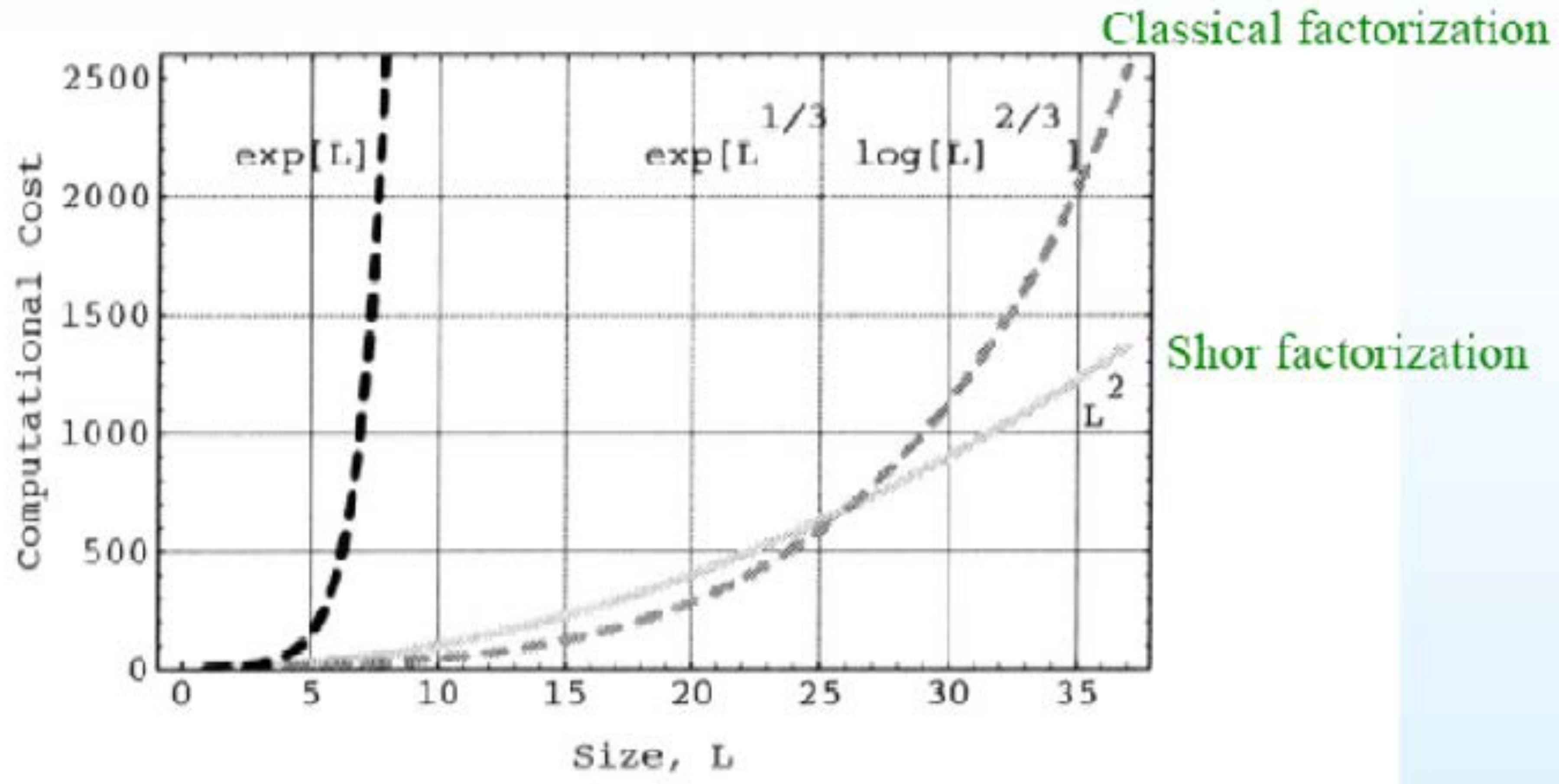


Fig. 2.5 The best factoring algorithms grow subexponentially (but super-polynomially) in *L*, the number of bits needed to specify the number being factored.

Why do we want to build one?

- Theoretical prediction : quantum computers should allow for solving some computational task efficiently, while hard for normal computers !
  - Efficient = takes a polynomial time t(L) in the input size L
  - Hard = takes an exponential time t(L) in the input size L

E.g: Factoring:
15 = 5x3

- Efficient for a quantum computer (Shor)
- Hard for normal computers

10433 x 16453 = ? (easy)

? x ? = 171654149 (hard)

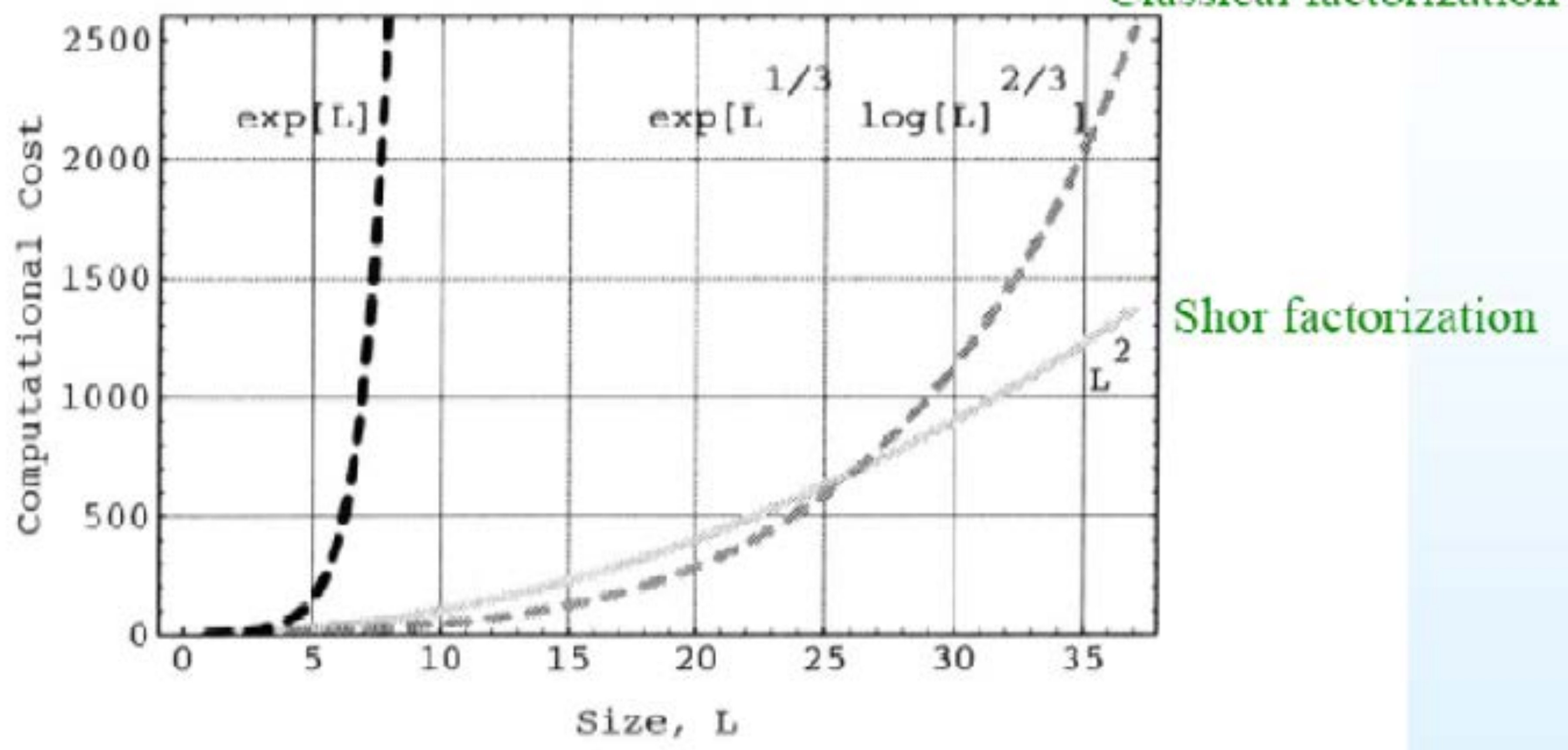


Fig. 2.5 The best factoring algorithms grow subexponentially (but super-polynomially) in *L*, the number of bits needed to specify the number being factored.

How do we program a quantum computer?

What is an algorithm?

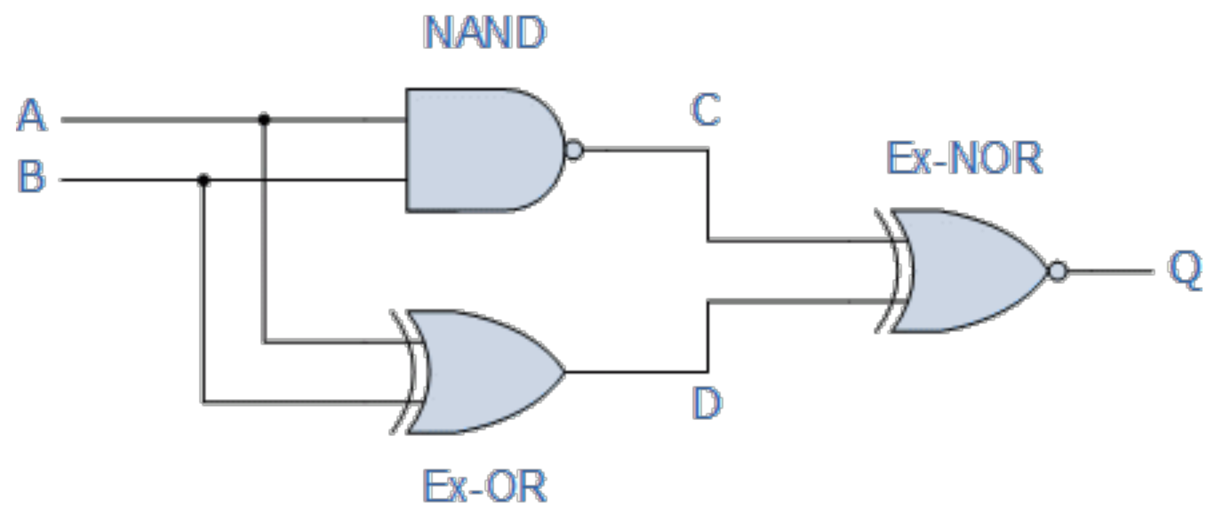
- An algorithm is sequence of operations to solve a specific problem
- It can be broken down into three steps: load, run, and read
- Typical algorithms that run on todays computer are expressed as logical operations on bits of information 0,1
- Example of logical operations:

What is an algorithm?

- An algorithm is sequence of operations to solve a specific problem
- It can be broken down into three steps: load, run, and read
- Typical algorithms that run on todays computer are expressed as logical operations on bits of information 0,1
- Example of logical operations:
  - NOT: $0 \to 1$
    $1 \to 0$ (the only single bit gate)
  - AND: $00 \to 0$
    $01 \to 0$
    $10 \to 0$
    $11 \to 1$
  - XOR: $00 \to 0$
    $01 \to 1$
    $10 \to 1$
    $11 \to 0$
  - NAND: $00 \to 1$
    $01 \to 1$
    $10 \to 1$
    $11 \to 0$

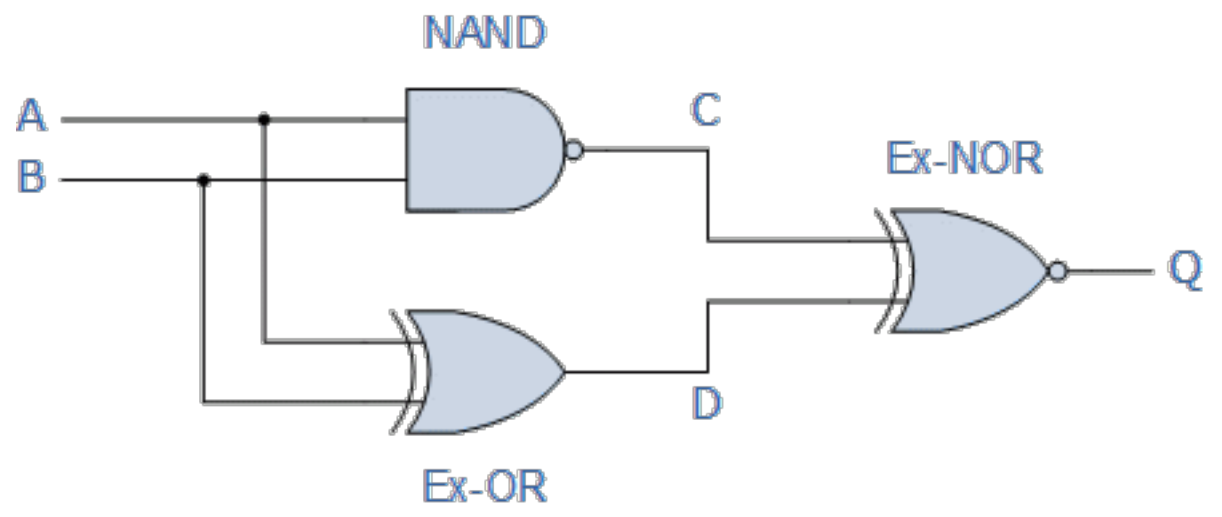
What is an algorithm?

- An algorithm is sequence of operations to solve a specific problem
- It can be broken down into three steps: load, run, and read
- Typical algorithms that run on todays computer are expressed as logical operations on bits of information 0,1
- Example of logical operations:
  - NOT: $0 \rightarrow 1$
    $1 \rightarrow 0$ (the only single bit gate)
  - AND: $00 \rightarrow 0$
    $01 \rightarrow 0$
    $10 \rightarrow 0$
    $11 \rightarrow 1$
  - XOR: $00 \rightarrow 0$
    $01 \rightarrow 1$
    $10 \rightarrow 1$
    $11 \rightarrow 0$
  - NAND: $00 \rightarrow 1$
    $01 \rightarrow 1$
    $10 \rightarrow 1$
    $11 \rightarrow 0$



What is an algorithm?

- An algorithm is sequence of operations to solve a specific problem
- It can be broken down into three steps: load, run, and read
- Typical algorithms that run on todays computer are expressed as logical operations on bits of information 0,1
- Example of logical operations:
  - NOT: $0 \rightarrow 1$
    $1 \rightarrow 0$ (the only single bit gate)
  - AND: $00 \rightarrow 0$
    $01 \rightarrow 0$
    $10 \rightarrow 0$
    $11 \rightarrow 1$
  - XOR: $00 \rightarrow 0$
    $01 \rightarrow 1$
    $10 \rightarrow 1$
    $11 \rightarrow 0$
  - NAND: $00 \rightarrow 1$
    $01 \rightarrow 1$
    $10 \rightarrow 1$
    $11 \rightarrow 0$



The blue and red sets are *universal gate sets* (Nielsen and Chuang, p 133)

Another universal gate set:

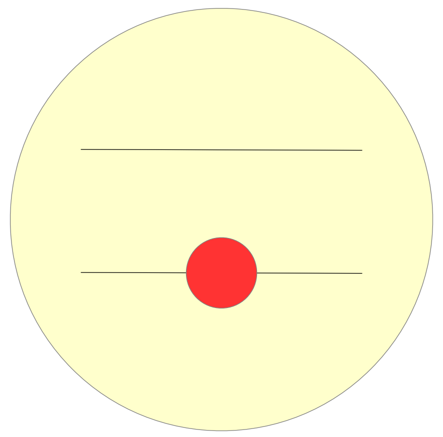
- Toffoli gate
  - Toff:
    $000 \rightarrow 000$
    $001 \rightarrow 001$
    $010 \rightarrow 010$
    ....
    $110 \rightarrow 111$
    $111 \rightarrow 110$

$$Toff = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{pmatrix}$$

  - 3 bit gate
  - Reversible
  - Universal by itself

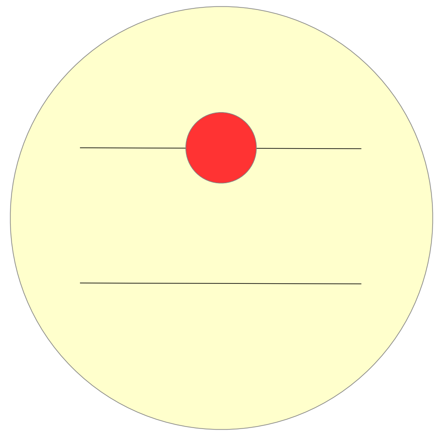
- Unlike the classical bits 0 and 1, it makes sense to consider arbitrary superpositions of the two quantum basis states |0> and |1>

State $|0\rangle$



$$|0\rangle \equiv \begin{pmatrix} 1 \\ 0 \end{pmatrix}$$

State $|1\rangle$



$$|1\rangle \equiv \begin{pmatrix} 0 \\ 1 \end{pmatrix}$$

$$|\psi\rangle = \alpha|0\rangle + \beta|1\rangle \equiv \begin{pmatrix} \alpha \\ \beta \end{pmatrix}$$

$$|\alpha|^2 + |\beta|^2 = 1$$

Quantum algorithms

- A quantum algorithm is sequence of operations to solve a specific problem on a quantum computer
- It can be broken down into three steps: prepare, evolve, and measure
- Single qubit gates

Quantum algorithms

- A quantum algorithm is sequence of operations to solve a specific problem on a quantum computer
- It can be broken down into three steps: prepare, evolve, and measure
- Single qubit gates

Pauli-X	X	$\begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$
Pauli-Z	Z	$\begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}$
$\pi/8$	T	$\begin{bmatrix} 1 & 0 \\ 0 & e^{i\pi/4} \end{bmatrix}$
Hadamard	H	$\frac{1}{\sqrt{2}}\begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$

- $|0\rangle \rightarrow |1\rangle \quad |1\rangle \rightarrow |0\rangle$
- $|0\rangle \rightarrow |0\rangle \quad |1\rangle \rightarrow -|1\rangle$
- $|0\rangle \rightarrow |0\rangle \quad |1\rangle \rightarrow e^{i\frac{\pi}{4}}|1\rangle$
- $|0\rangle \rightarrow |+\rangle \quad |1\rangle \rightarrow |-\rangle$

$$|\pm\rangle = \frac{|0\rangle \pm |1\rangle}{\sqrt{2}}$$

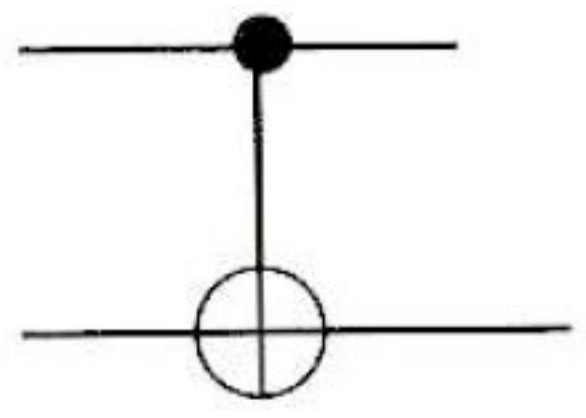
- An example of a 2-qubit gate: controlled NOT (CNOT)

$$|00\rangle \to |00\rangle$$

$$|01\rangle \to |01\rangle$$

$$|10\rangle \to |11\rangle$$

$$|11\rangle \to |10\rangle$$



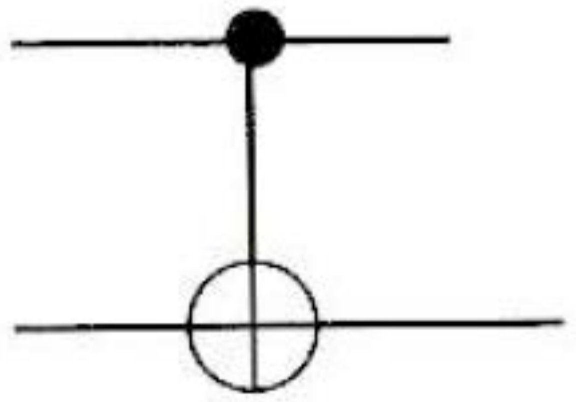
- An example of a 2-qubit gate: controlled NOT (CNOT)

$|00\rangle \rightarrow |00\rangle$

$|01\rangle \rightarrow |01\rangle$

$|10\rangle \rightarrow |11\rangle$

$|11\rangle \rightarrow |10\rangle$



$$|\Psi\rangle = c_{00}|0\rangle|0\rangle + c_{01}|0\rangle|1\rangle + c_{10}|1\rangle|0\rangle + c_{11}|1\rangle|1\rangle$$

$$\begin{bmatrix} c_{00}^f \\ c_{01}^f \\ c_{10}^f \\ c_{11}^f \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix} \cdot \begin{bmatrix} c_{00}^i \\ c_{01}^i \\ c_{10}^i \\ c_{11}^i \end{bmatrix}$$

Two-qubits gates

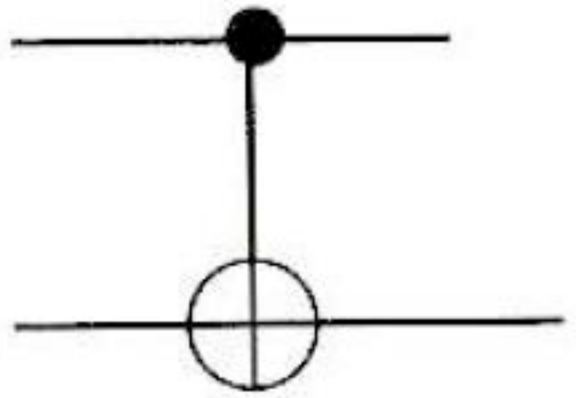
- An example of a 2-qubit gate: controlled NOT (CNOT)

$|00\rangle \rightarrow |00\rangle$

$|01\rangle \rightarrow |01\rangle$

$|10\rangle \rightarrow |11\rangle$

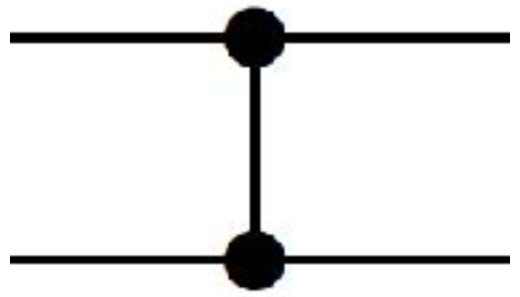
$|11\rangle \rightarrow |10\rangle$



$$|\Psi\rangle = c_{00}|0\rangle|0\rangle + c_{01}|0\rangle|1\rangle + c_{10}|1\rangle|0\rangle + c_{11}|1\rangle|1\rangle$$

$$\begin{bmatrix} c_{00}^f \\ c_{01}^f \\ c_{10}^f \\ c_{11}^f \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix} \cdot \begin{bmatrix} c_{00}^i \\ c_{01}^i \\ c_{10}^i \\ c_{11}^i \end{bmatrix}$$

- Analogously: controlled-Z



$$C_Z = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & -1 \end{pmatrix}$$

Toffoli gate

- Toff: $|000\rangle \rightarrow |000\rangle$
  $|001\rangle \rightarrow |001\rangle$
  $|010\rangle \rightarrow |010\rangle$
  ....
  $|110\rangle \rightarrow |111\rangle$
  $|111\rangle \rightarrow |110\rangle$

$$Toff = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{pmatrix}$$

- 3-qubit gate
- Same matrix representation as the classical Toffoli

Universal gate sets

Universal gate sets

- One possible universal gate set: $\{T = \begin{pmatrix} 1 & 0 \\ 0 & e^{i\frac{\pi}{4}} \end{pmatrix}, H = \frac{1}{\sqrt{2}}\begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}, C_Z = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & -1 \end{pmatrix}\}$

- Another possible universal gate set: $\{Toff = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{pmatrix}, H = \frac{1}{\sqrt{2}}\begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}\}$

Universal gate sets

- One possible universal gate set: $\{T = \begin{pmatrix} 1 & 0 \\ 0 & e^{i\frac{\pi}{4}} \end{pmatrix}, H = \frac{1}{\sqrt{2}}\begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}, C_Z = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & -1 \end{pmatrix}\}$

- Another possible universal gate set: $\{Toff = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{pmatrix}, H = \frac{1}{\sqrt{2}}\begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}\}$

- We can approximating an arbitrary $2^N \times 2^N$ unitary matrix using sequence of 2×2 matrices or 4×4 matrices (6×6 with the 2nd universal gate set)

Universal gate sets

- One possible universal gate set: $\{T = \begin{pmatrix} 1 & 0 \\ 0 & e^{i\frac{\pi}{4}} \end{pmatrix}, H = \frac{1}{\sqrt{2}}\begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}, C_Z = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & -1 \end{pmatrix}\}$

- Another possible universal gate set: $\{Toff = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{pmatrix}, H = \frac{1}{\sqrt{2}}\begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}\}$

- We can approximating an arbitrary $2^N \times 2^N$ unitary matrix using sequence of 2×2 matrices or 4×4 matrices (6×6 with the 2nd universal gate set)

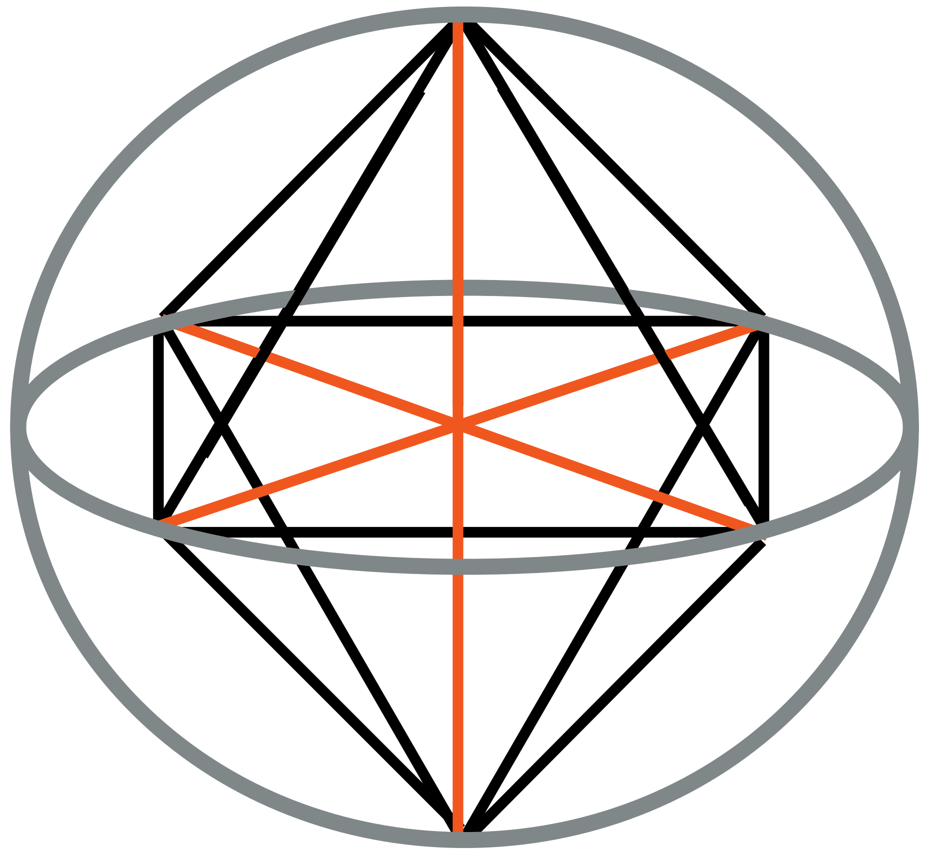
- From the second set we see that classical computing is a subset of quantum computing and that classical computing misses coherence

Gottesman-Knill theorem

A QC based only on:

(i) qubits initialised in a X,Y,Z eigenstate (= stabiliser state)
(ii) Clifford group operations
(iii) X,Y,Z measurements

can be simulated efficiently with a classical computer



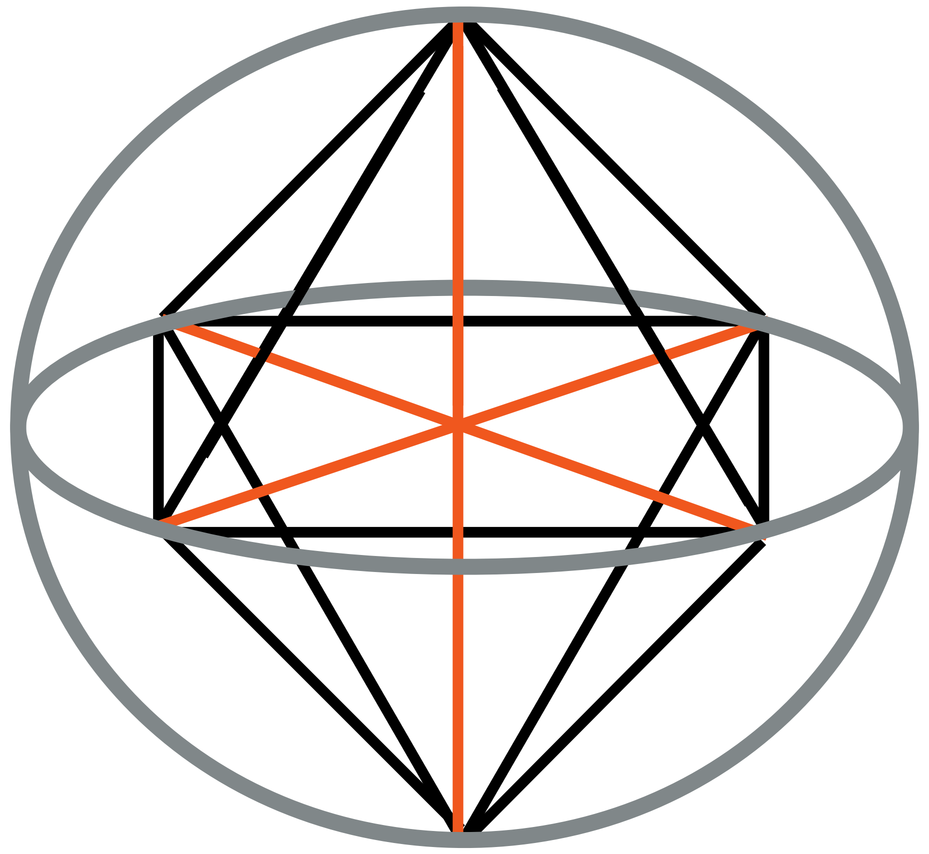
$$\mathcal{C}_2^n = \langle H, S, \text{CNOT} \rangle \qquad H = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} \qquad S = \begin{pmatrix} 1 & 0 \\ 0 & i \end{pmatrix}$$

Includes Pauli matrices X, Y, Z

A QC based only on:

(i) qubits initialised in a X,Y,Z eigenstate (= stabiliser state)
(ii) Clifford group operations
(iii) X,Y,Z measurements

can be simulated efficiently with a classical computer

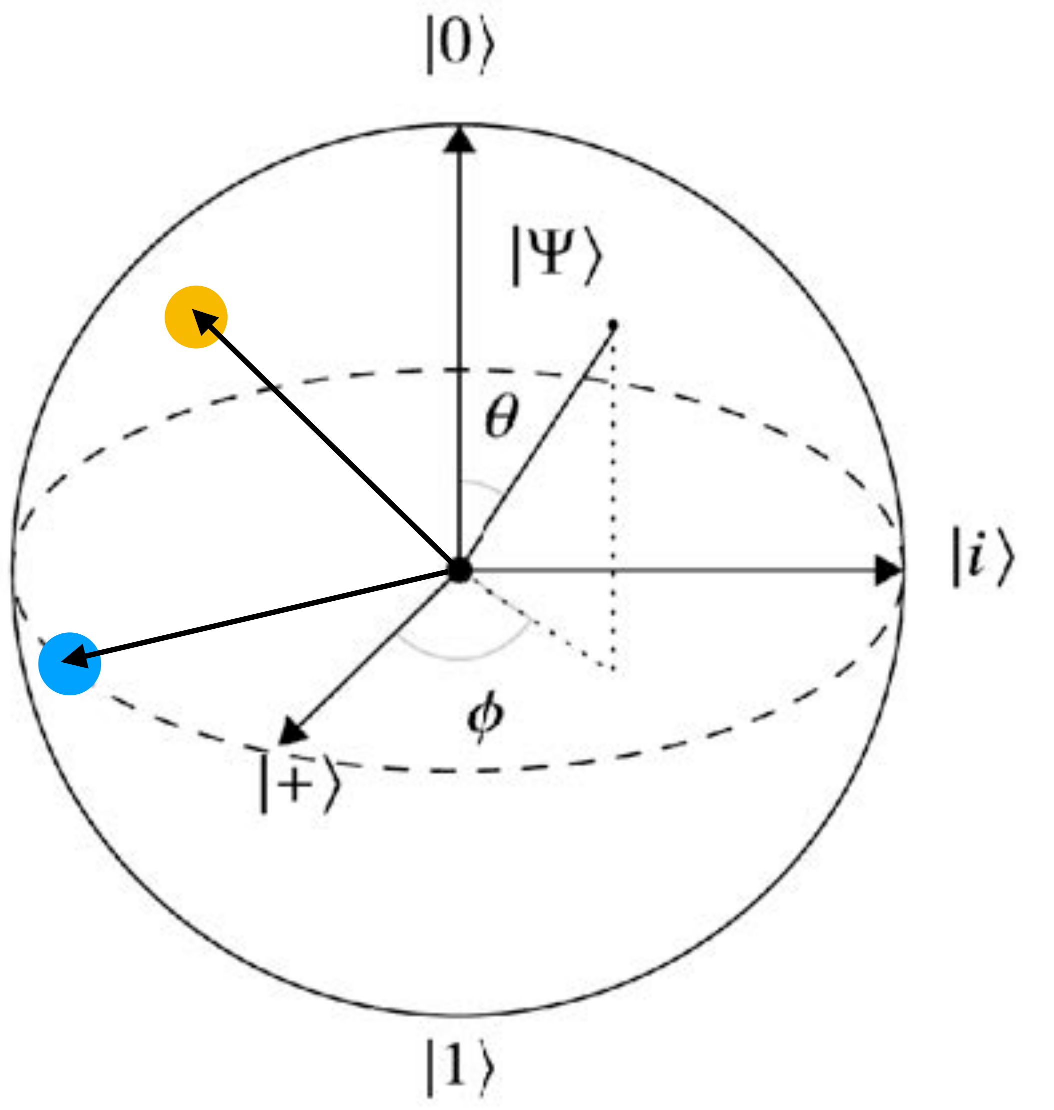


$$\mathcal{C}_2^n = \langle H, S, \text{CNOT} \rangle \qquad H = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} \qquad S = \begin{pmatrix} 1 & 0 \\ 0 & i \end{pmatrix}$$

Includes Pauli matrices X, Y, Z

- No exponential quantum advantage with these ingredients only!

Magic states + Clifford circuits are universal



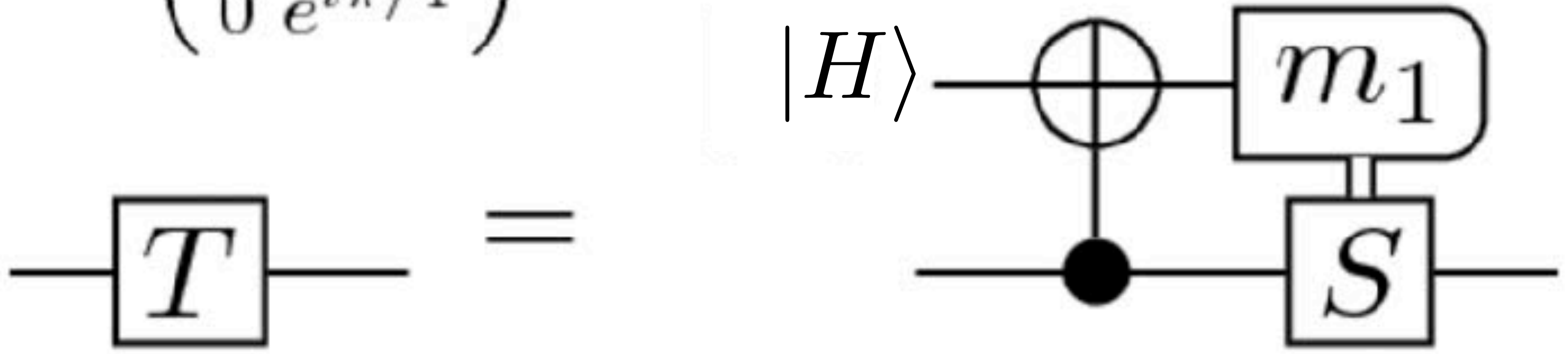
- T-state and H-state:

$$|T\rangle = \cos\frac{\theta}{2}|0\rangle + \sin\frac{\theta}{2}e^{i\frac{\pi}{4}}|1\rangle \quad \text{with} \quad \theta = \arccos\left(\frac{1}{\sqrt{3}}\right)$$

$$|H\rangle = \frac{1}{\sqrt{2}}\left(|0\rangle + e^{i\frac{\pi}{4}}|1\rangle\right),$$

- From magic states to the T-gate:

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

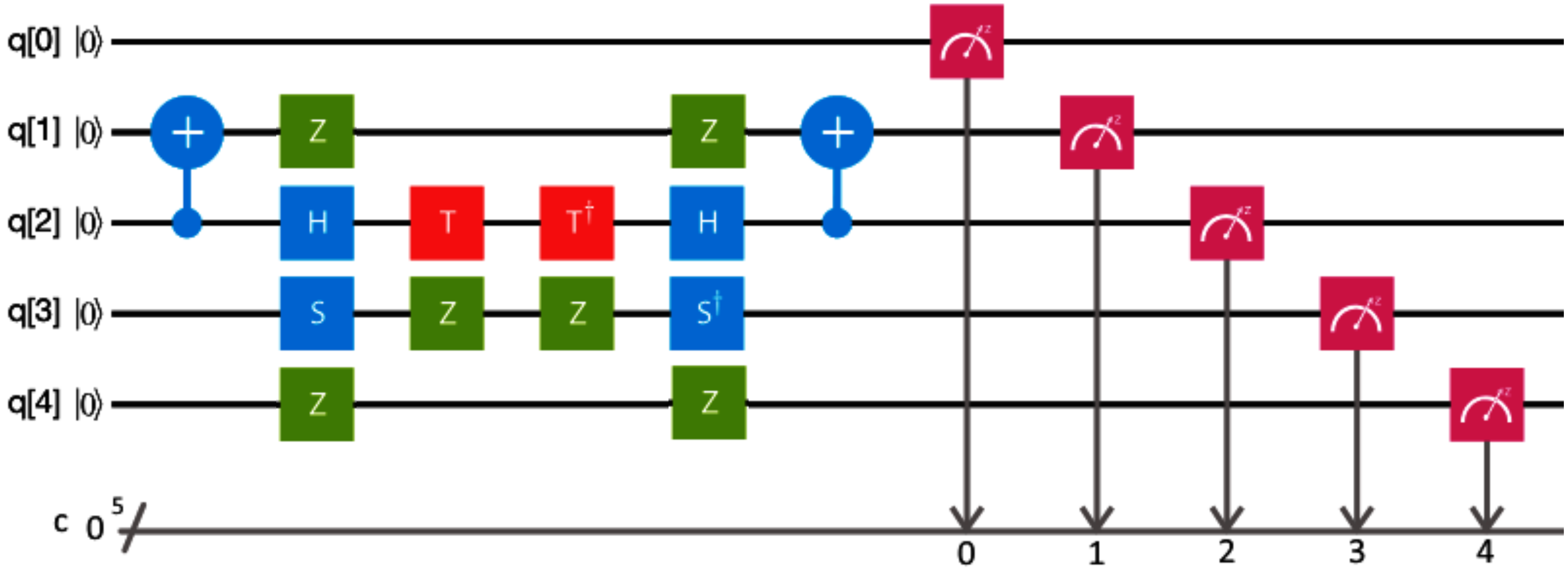


$|H\rangle$ states $(+$Cliffords$)$ enable T gates

Sergey Bravyi and Alexei Kitaev, PRA 71 022316 (2005)

Models of quantum computation

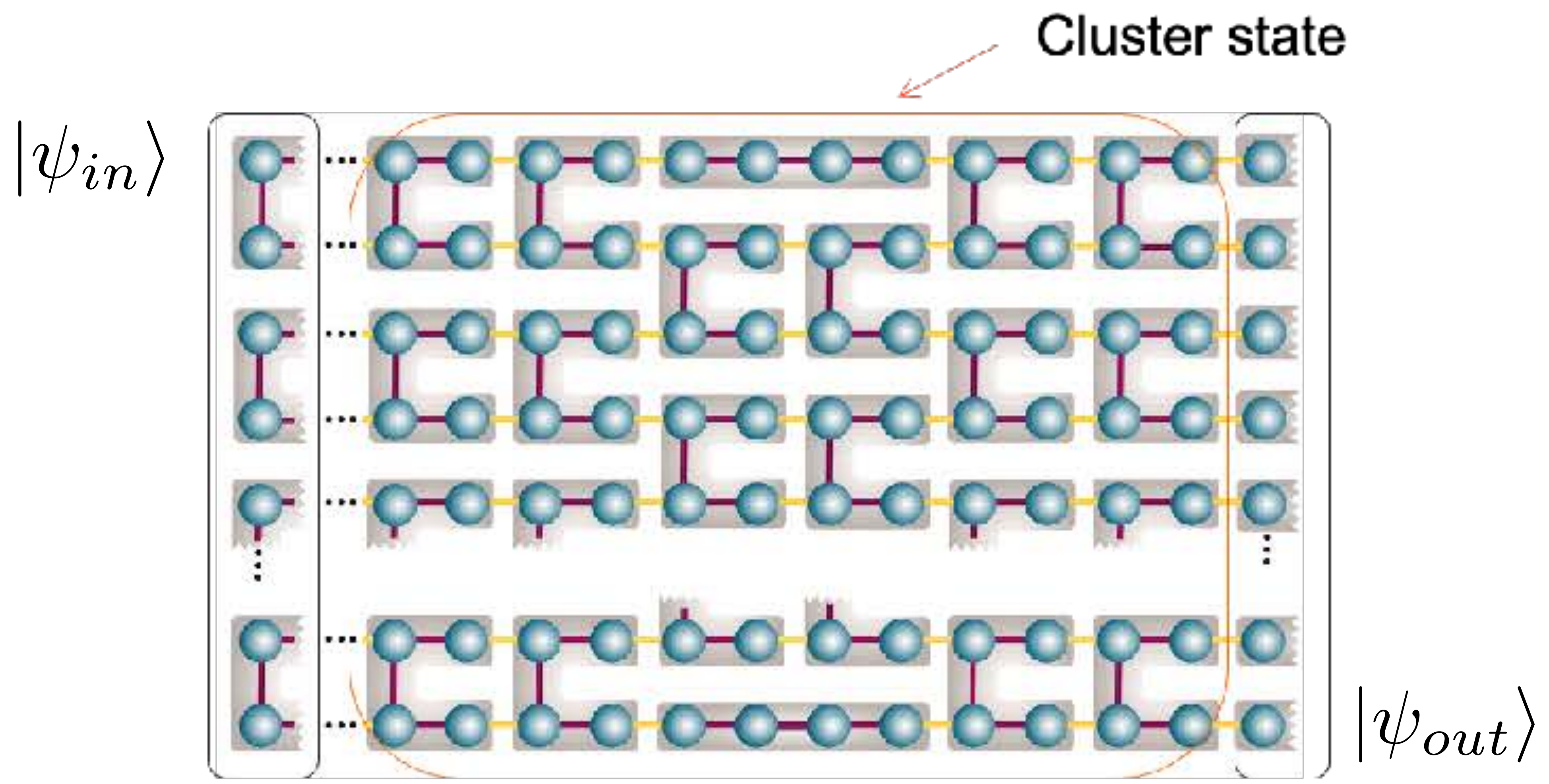
1) Circuit model



$$\{T = \begin{pmatrix} 1 & 0 \\ 0 & e^{i\frac{\pi}{4}} \end{pmatrix}, H = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}, C_Z = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & -1 \end{pmatrix}\}$$ Universal gate set

- We are going to see an example of an algorithm executed in this model (Deutsche-Jozsa)

2) Measurement-based model



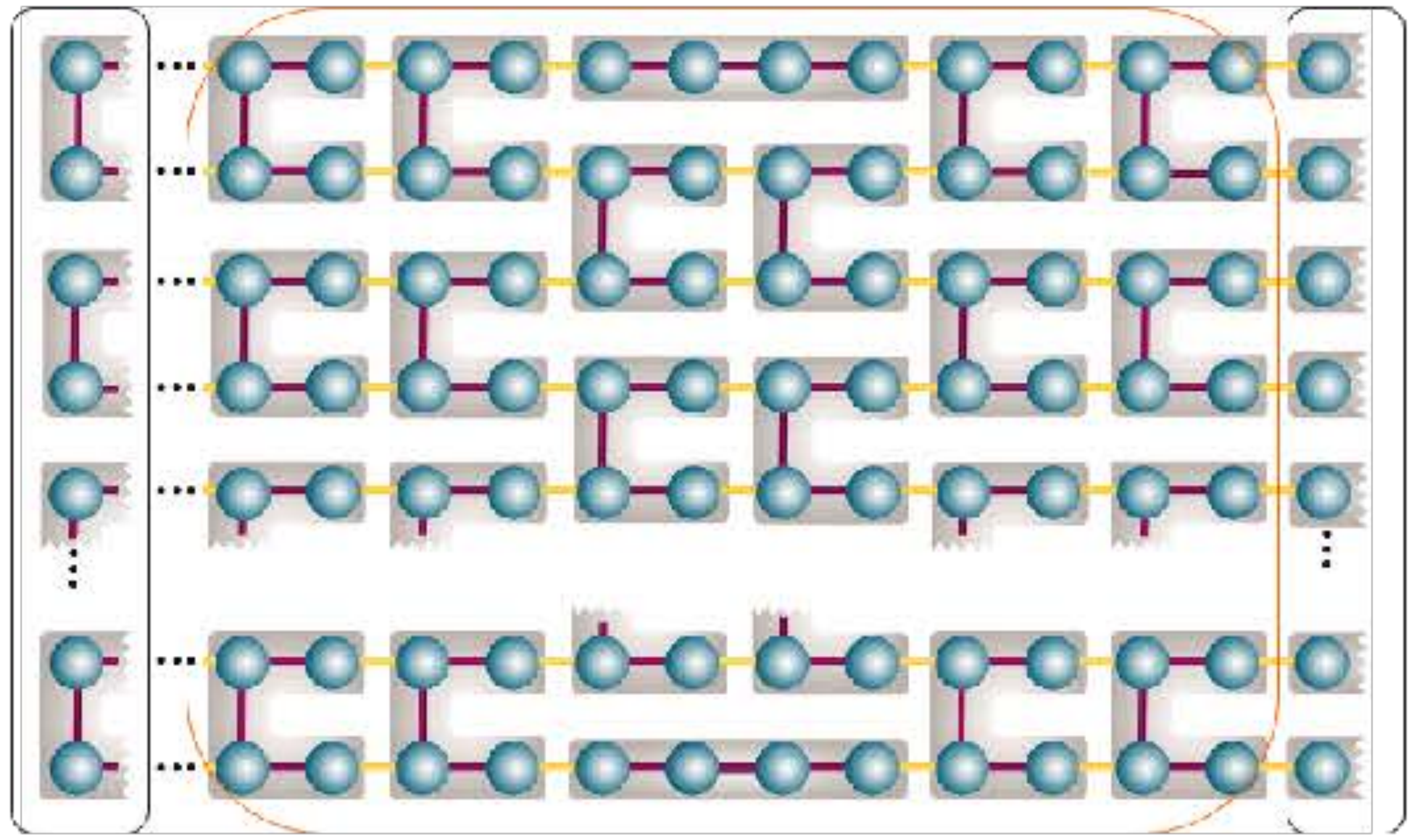
Manipulation of the input state achieved by entangling it with a cluster state and by performing suitable local measurements on its nodes

$\longrightarrow$ Unmeasured nodes projected on $|\psi_{out}\rangle = U|\psi_{in}\rangle$

2) Measurement-based model

Cluster state

$|\psi_{in}\rangle$



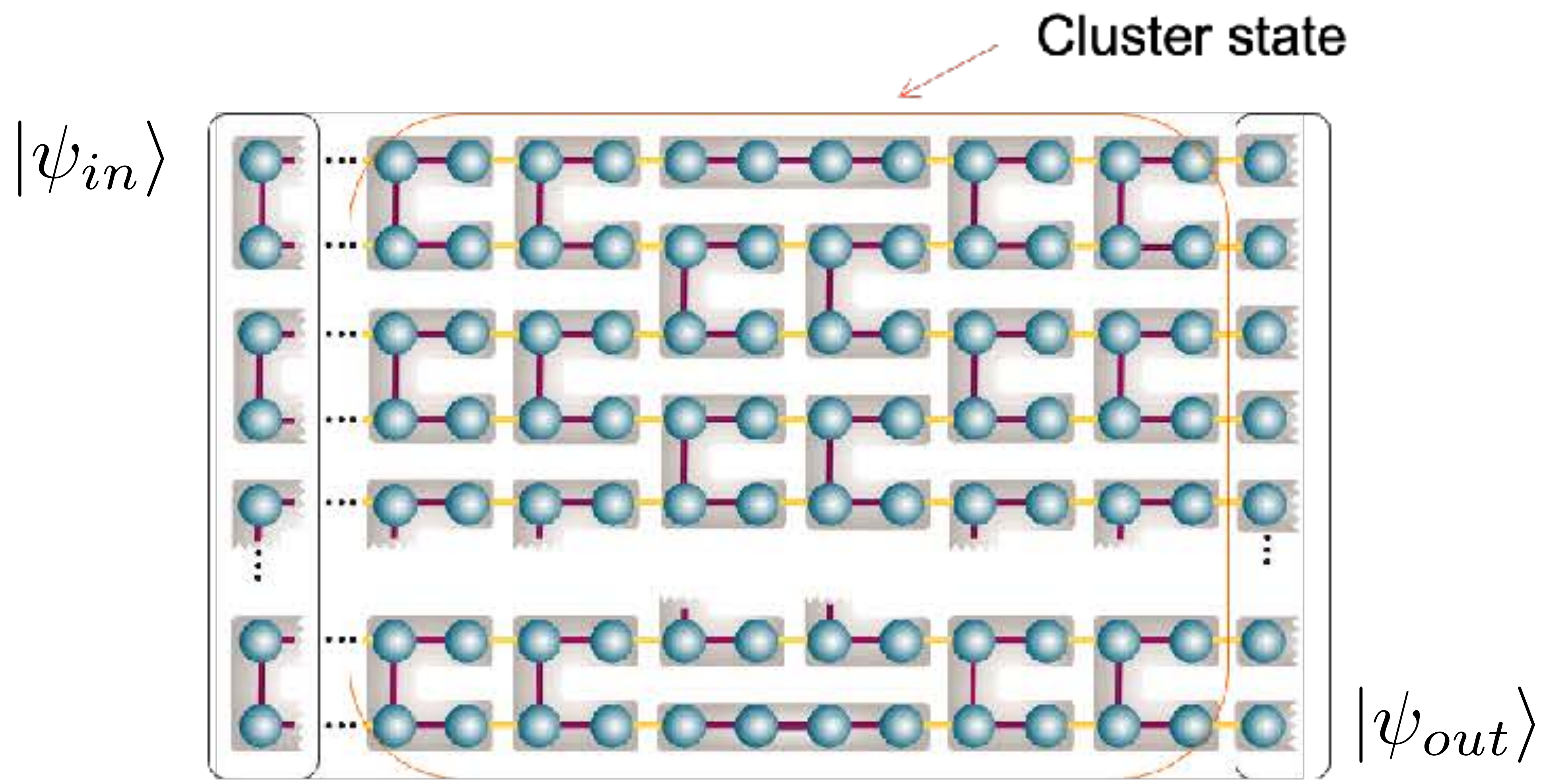
$|\psi_{out}\rangle$

Manipulation of the input state achieved by entangling it with a cluster state and by performing suitable local measurements on its nodes

$\longrightarrow$ Unmeasured nodes projected on $|\psi_{out}\rangle = U|\psi_{in}\rangle$

- Cluster state: state associated to a graph, operationally defined as:

  - start with as many |+> states as the nodes of the graph
  - apply CZ gate if two nodes are related by an edge



Manipulation of the input state achieved by entangling it with a cluster state and by performing suitable local measurements on its nodes

$\longrightarrow$ Unmeasured nodes projected on $|\psi_{out}\rangle = U|\psi_{in}\rangle$

- Cluster state: state associated to a graph, operationally defined as:

  - start with as many |+> states as the nodes of the graph
  - apply CZ gate if two nodes are related by an edge

- Example: linear cluster state

$$|\psi_V\rangle = C_Z^{1,2} C_Z^{2,3} |+\rangle|+\rangle|+\rangle$$

3) Adiabatic quantum computation

N spins ↑ or ↓ (resp 1 or –1), connected by wires, J < 0 (ferromagnetic) or J > 0 (antiferromagnetic). External magnetic field h

$$H(\lambda) = \lambda H_1 + (1-\lambda) H_0$$
$$= \lambda\left(\sum_{ij} J_{ij} S_i^z S_j^z + \sum_i h_i S_i^z\right) - (1-\lambda)\sum_i S_i^x$$

final Hamiltonian, ground state encodes the solution of the problem

initial Hamiltonian, ground state easy to prepare

T. Albash and D. Lidar, Rev. Mod. Phys. 90, 015002 (2018)

- Circuit model, MBQC and Adiabatic Quantum Computing are
  - equivalent
  - universal
  - only some of the universal models for QC

- Circuit model, MBQC and Adiabatic Quantum Computing are
  - equivalent
  - universal
  - only some of the universal models for QC

- Let’s see how to used them for a simple algorithm: Deutsch-Jozsa!

- Circuit model, MBQC and Adiabatic Quantum Computing are
  - equivalent
  - universal
  - only some of the universal models for QC

- Let's see how to used them for a simple algorithm: Deutsch-Jozsa!

- We focus on the circuit model, but constructions exist to convert quantum algorithms within different models

An example of quantum algorithm: Deutsch-Jozsa

Quantum parallelism

Quantum parallelism

- By definition, with a universal quantum computer we can implement any unitary (e.g. on 2 qubits)
- Consider the unitary *Uf* associated to the classical single bit function *f*: $|x\rangle|y\rangle \rightarrow |x\rangle|y \oplus f(x)\rangle$

Quantum parallelism

- By definition, with a universal quantum computer we can implement any unitary (e.g. on 2 qubits)
- Consider the unitary *Uf* associated to the classical single bit function *f*: $|x\rangle|y\rangle \rightarrow |x\rangle|y \oplus f(x)\rangle$

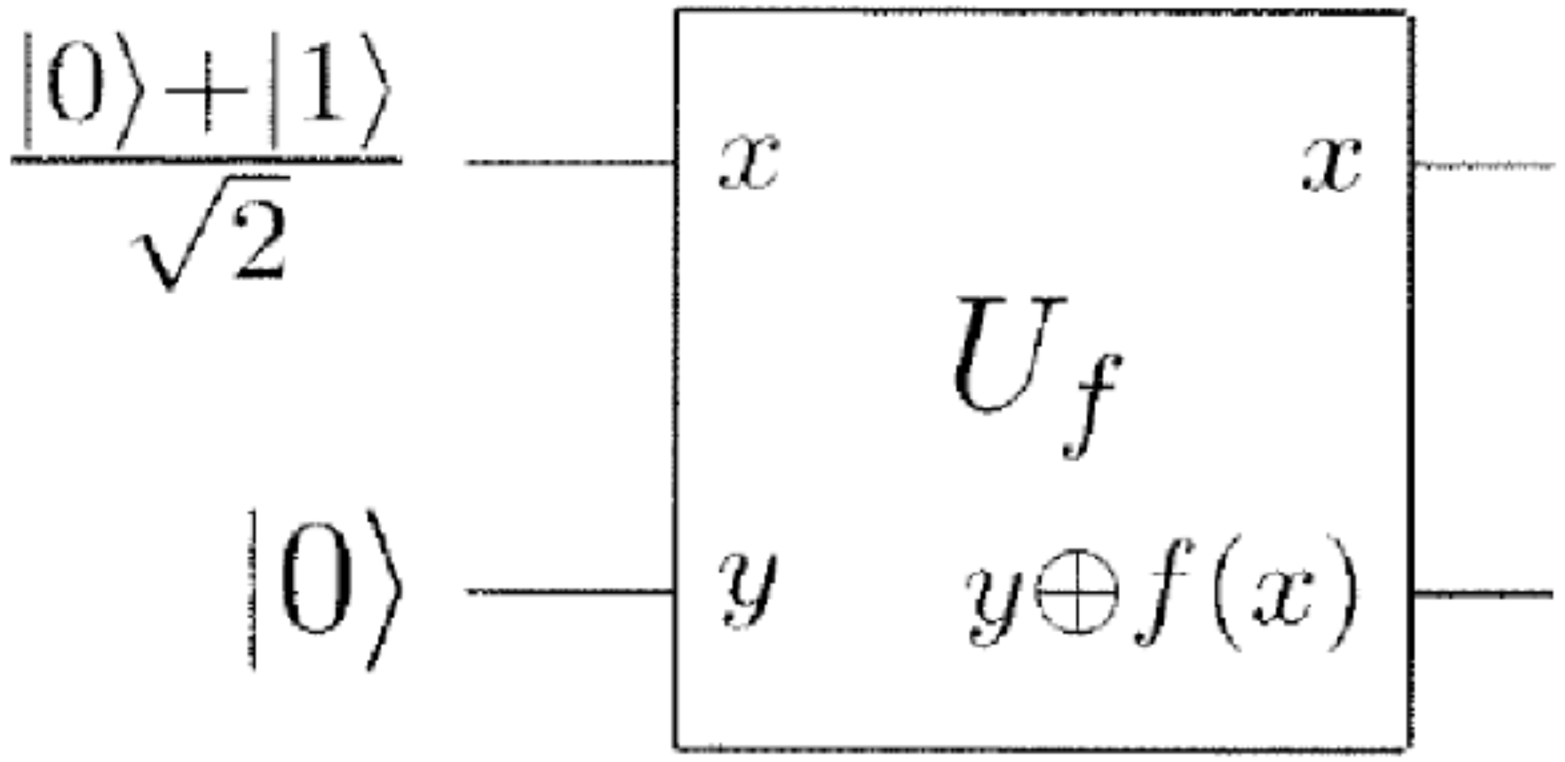
  if |y>=|0> we have: $|x\rangle|0\rangle \rightarrow |x\rangle|f(x)\rangle$

Quantum parallelism

WACQT Wallenberg Centre for Quantum Technology

- By definition, with a universal quantum computer we can implement any unitary (e.g. on 2 qubits)
- Consider the unitary *Uf* associated to the classical single bit function *f*: $|x\rangle|y\rangle \rightarrow |x\rangle|y \oplus f(x)\rangle$

  if |y>=|0> we have: $|x\rangle|0\rangle \rightarrow |x\rangle|f(x)\rangle$



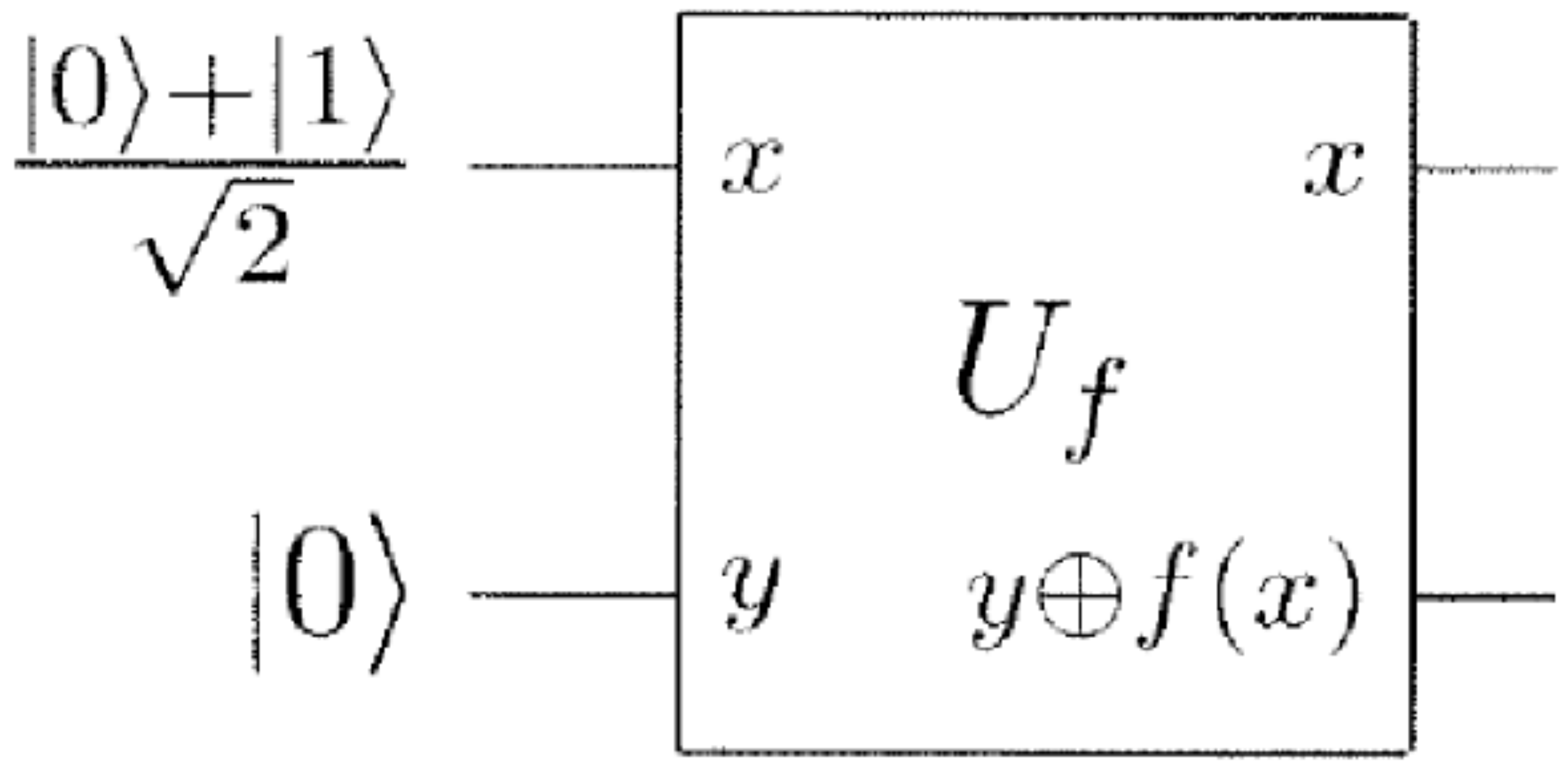
$$|\Psi\rangle = \frac{|0\rangle|f(0)\rangle + |1\rangle|f(1)\rangle}{\sqrt{2}}$$

Quantum parallelism

WACQT | Wallenberg Centre for Quantum Technology

- By definition, with a universal quantum computer we can implement any unitary (e.g. on 2 qubits)
- Consider the unitary *Uf* associated to the classical single bit function *f*: $|x\rangle|y\rangle \rightarrow |x\rangle|y \oplus f(x)\rangle$

if |y>=|0> we have: $|x\rangle|0\rangle \rightarrow |x\rangle|f(x)\rangle$



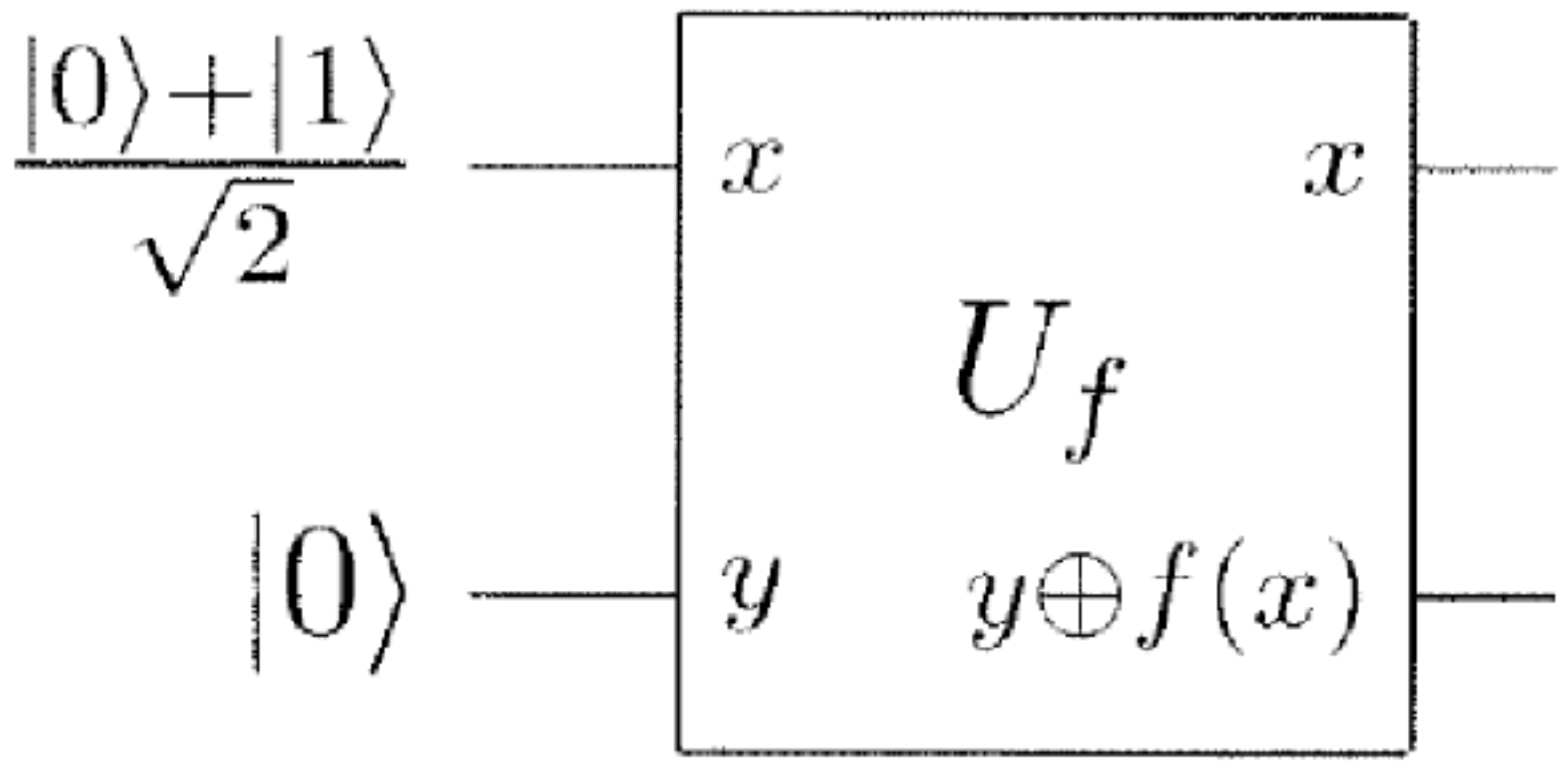
$$|\Psi\rangle = \frac{|0\rangle|f(0)\rangle + |1\rangle|f(1)\rangle}{\sqrt{2}}$$

- The output state contains information about both values of the function *f(0)* and *f(1)*!

Quantum parallelism

- By definition, with a universal quantum computer we can implement any unitary (e.g. on 2 qubits)
- Consider the unitary *Uf* associated to the classical single bit function *f*: $|x\rangle|y\rangle \rightarrow |x\rangle|y \oplus f(x)\rangle$

  if |y>=|0> we have: $|x\rangle|0\rangle \rightarrow |x\rangle|f(x)\rangle$



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

- The output state contains information about both values of the function *f(0)* and *f(1)*!
- But reading out the state, we get either one or the other…

Deutsch's algorithm

- Can we exploit quantum parallelism to extract a global property of the function?

Deutsch's algorithm

- Can we exploit quantum parallelism to extract a global property of the function?

- Given a single-bit function f(x), that takes 0-> f(0) and 1-> f(1),

  we are interested in the property: is f(x) constant, f(0)=f(1), or balanced, f(0) different from f(1)?

Deutsch's algorithm

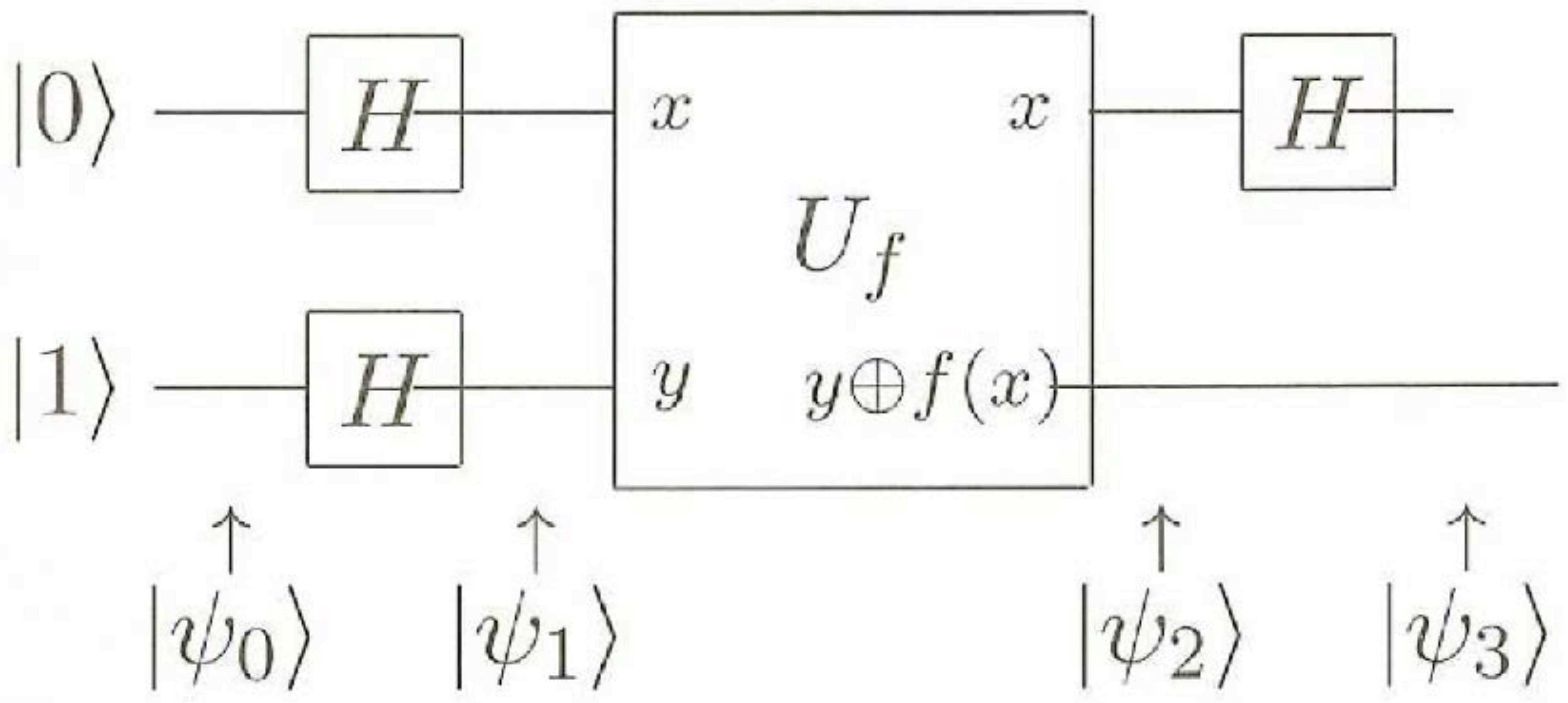
- Can we exploit quantum parallelism to extract a global property of the function?

- Given a single-bit function f(x), that takes 0-> f(0) and 1-> f(1),

  we are interested in the property: is f(x) constant, f(0)=f(1), or balanced, f(0) different from f(1)?

- Like before, we have a quantum computer which implements *Uf* $|x\rangle|y\rangle \to |x\rangle|y \oplus f(x)\rangle$

Deutsch algorithm

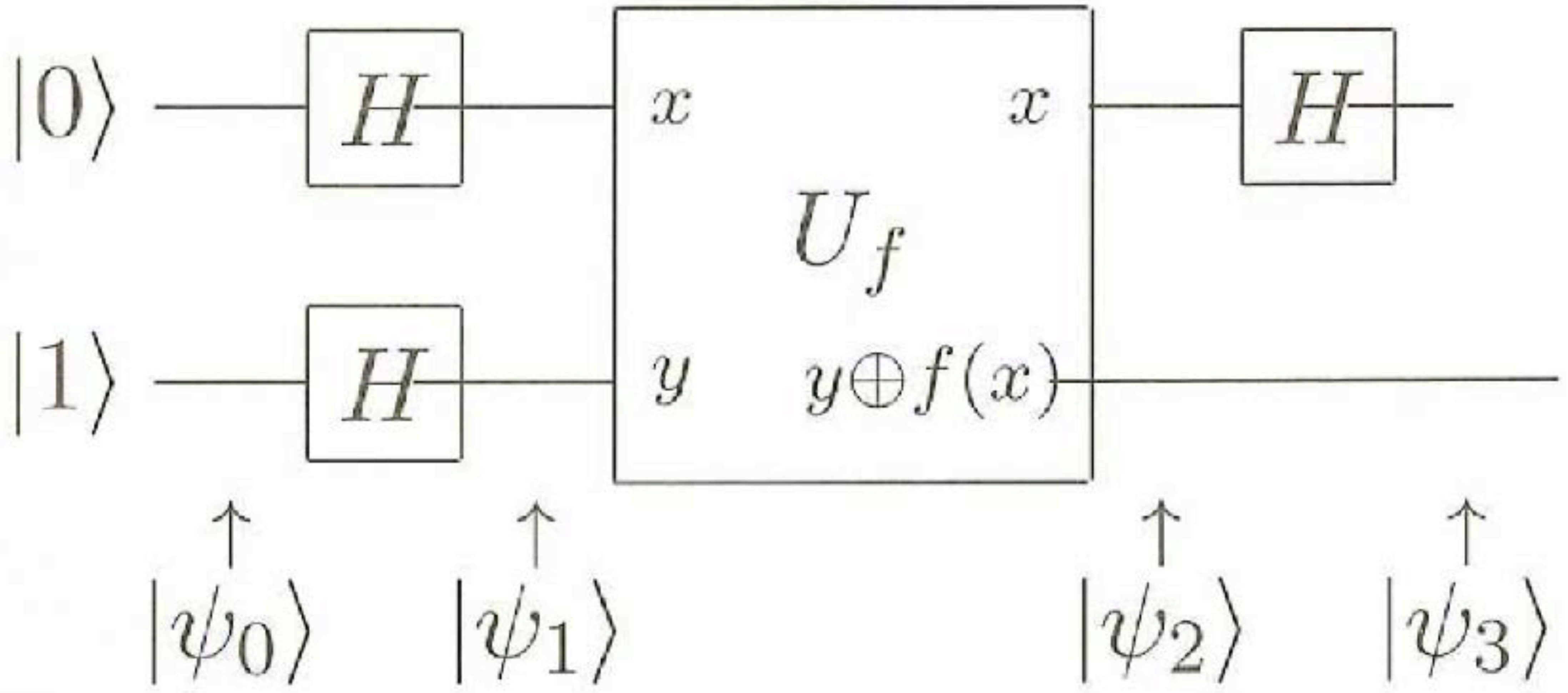


$$|\Psi_1\rangle = \left(\frac{|0\rangle + |1\rangle}{\sqrt{2}}\right)\left(\frac{|0\rangle - |1\rangle}{\sqrt{2}}\right)$$

- *Uf* acts on $|x\rangle(|0\rangle - |1\rangle)/\sqrt{2}$ as

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

From Nielsen Chuang



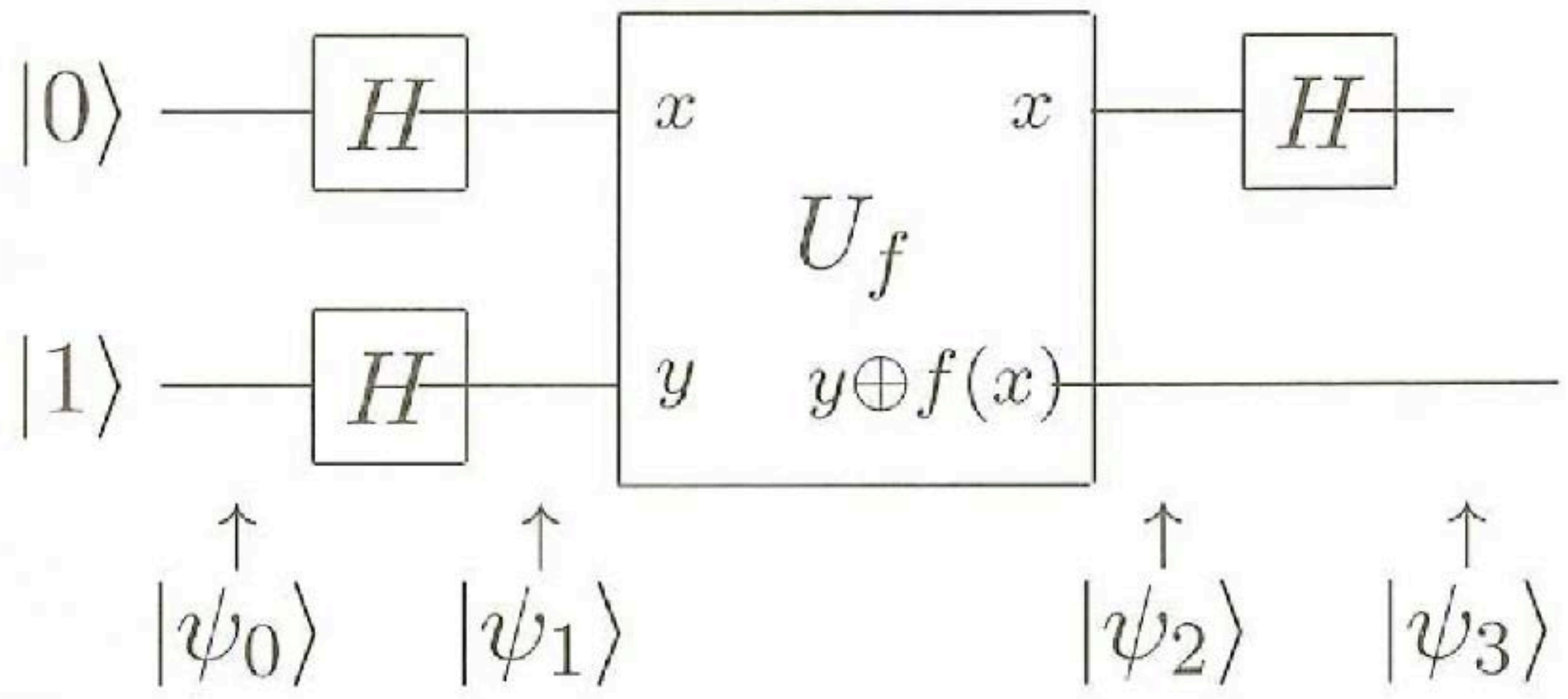
- *Uf* acts on $|x\rangle(|0\rangle - |1\rangle)/\sqrt{2}$ as

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

$$|\Psi_1\rangle = \left(\frac{|0\rangle + |1\rangle}{\sqrt{2}}\right)\left(\frac{|0\rangle - |1\rangle}{\sqrt{2}}\right)$$

$$|\Psi_2\rangle = \begin{cases} \pm\left(\frac{|0\rangle+|1\rangle}{\sqrt{2}}\right)\left(\frac{|0\rangle-|1\rangle}{\sqrt{2}}\right), & f(0) = f(1) \\ \pm\left(\frac{|0\rangle-|1\rangle}{\sqrt{2}}\right)\left(\frac{|0\rangle-|1\rangle}{\sqrt{2}}\right), & f(0) \neq f(1) \end{cases}$$

From Nielsen Chuang



- *Uf* acts on $|x\rangle(|0\rangle - |1\rangle)/\sqrt{2}$ as

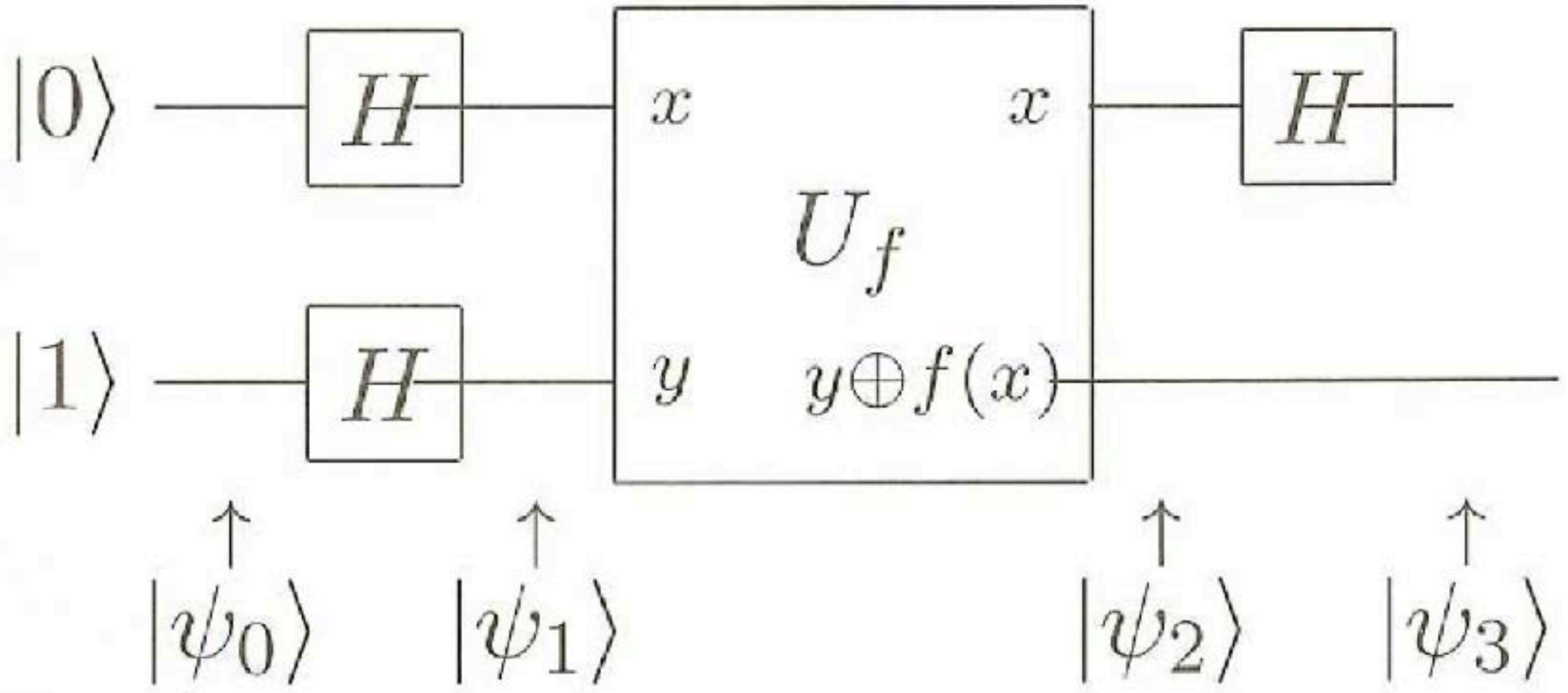
$$|x\rangle \frac{(|f(x)\rangle - |1 \oplus f(x)\rangle)}{\sqrt{2}} = (-1)^{f(x)} \frac{|x\rangle(|0\rangle - |1\rangle)}{\sqrt{2}}$$

$$|\Psi_1\rangle = \left(\frac{|0\rangle + |1\rangle}{\sqrt{2}}\right)\left(\frac{|0\rangle - |1\rangle}{\sqrt{2}}\right)$$

$$|\Psi_2\rangle = \begin{cases} \pm\left(\frac{|0\rangle+|1\rangle}{\sqrt{2}}\right)\left(\frac{|0\rangle-|1\rangle}{\sqrt{2}}\right), & f(0) = f(1) \\ \pm\left(\frac{|0\rangle-|1\rangle}{\sqrt{2}}\right)\left(\frac{|0\rangle-|1\rangle}{\sqrt{2}}\right), & f(0) \neq f(1) \end{cases}$$

$$|\Psi_3\rangle = \begin{cases} \pm|0\rangle\left(\frac{|0\rangle-|1\rangle}{\sqrt{2}}\right), & f(0) = f(1) \\ \pm|1\rangle\left(\frac{|0\rangle-|1\rangle}{\sqrt{2}}\right), & f(0) \neq f(1) \end{cases}$$

From Nielsen Chuang



- *Uf* acts on $|x\rangle(|0\rangle - |1\rangle)/\sqrt{2}$ as

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

$$|\Psi_1\rangle = \left(\frac{|0\rangle + |1\rangle}{\sqrt{2}}\right)\left(\frac{|0\rangle - |1\rangle}{\sqrt{2}}\right)$$

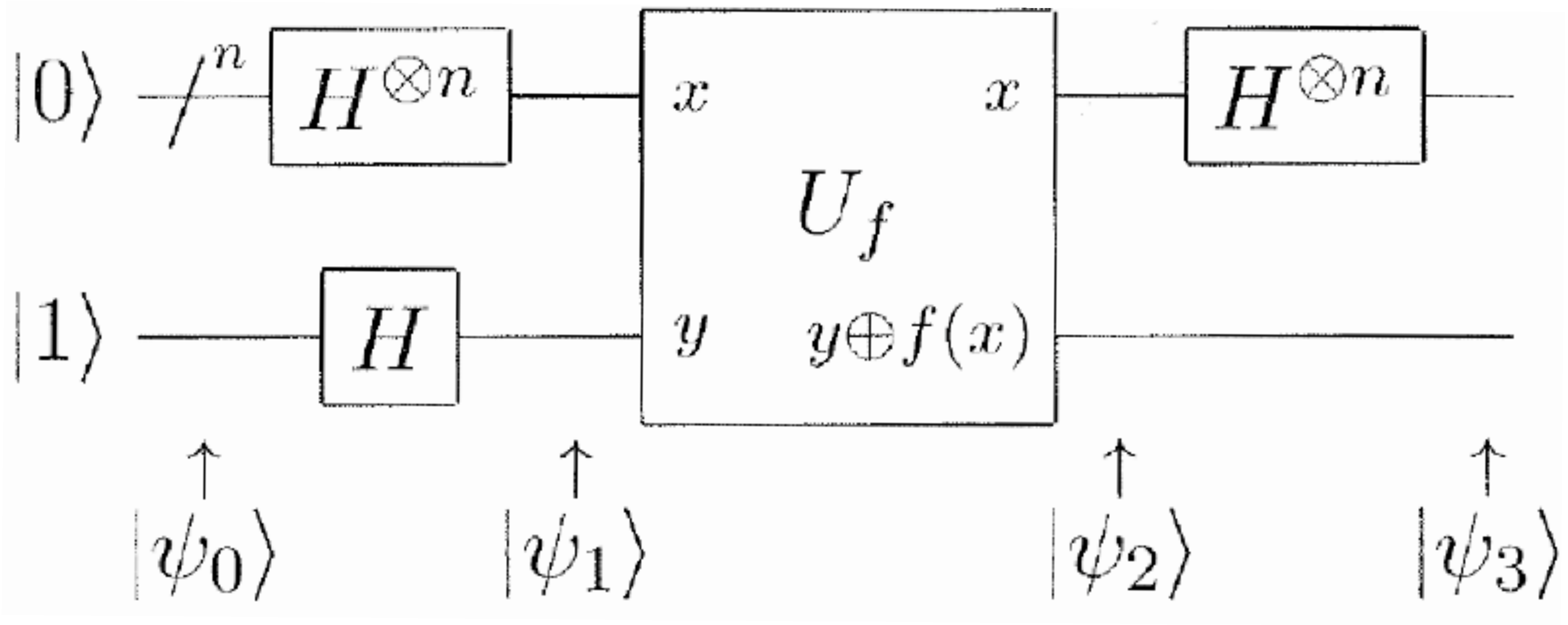
$$|\Psi_2\rangle = \begin{cases} \pm\left(\frac{|0\rangle+|1\rangle}{\sqrt{2}}\right)\left(\frac{|0\rangle-|1\rangle}{\sqrt{2}}\right), & f(0) = f(1) \\ \pm\left(\frac{|0\rangle-|1\rangle}{\sqrt{2}}\right)\left(\frac{|0\rangle-|1\rangle}{\sqrt{2}}\right), & f(0) \neq f(1) \end{cases}$$

$$|\Psi_3\rangle = \begin{cases} \pm|0\rangle\left(\frac{|0\rangle-|1\rangle}{\sqrt{2}}\right), & f(0) = f(1) \\ \pm|1\rangle\left(\frac{|0\rangle-|1\rangle}{\sqrt{2}}\right), & f(0) \neq f(1) \end{cases}$$

A single run of the algorithm allows for determining if f is constant or balanced, while 2 runs are needed classically

From Nielsen Chuang

- f(x) is a n-bit function: $x \in \{0,1\}^n \rightarrow f(x) \in \{0,1\}$; is f(x) constant or balanced?



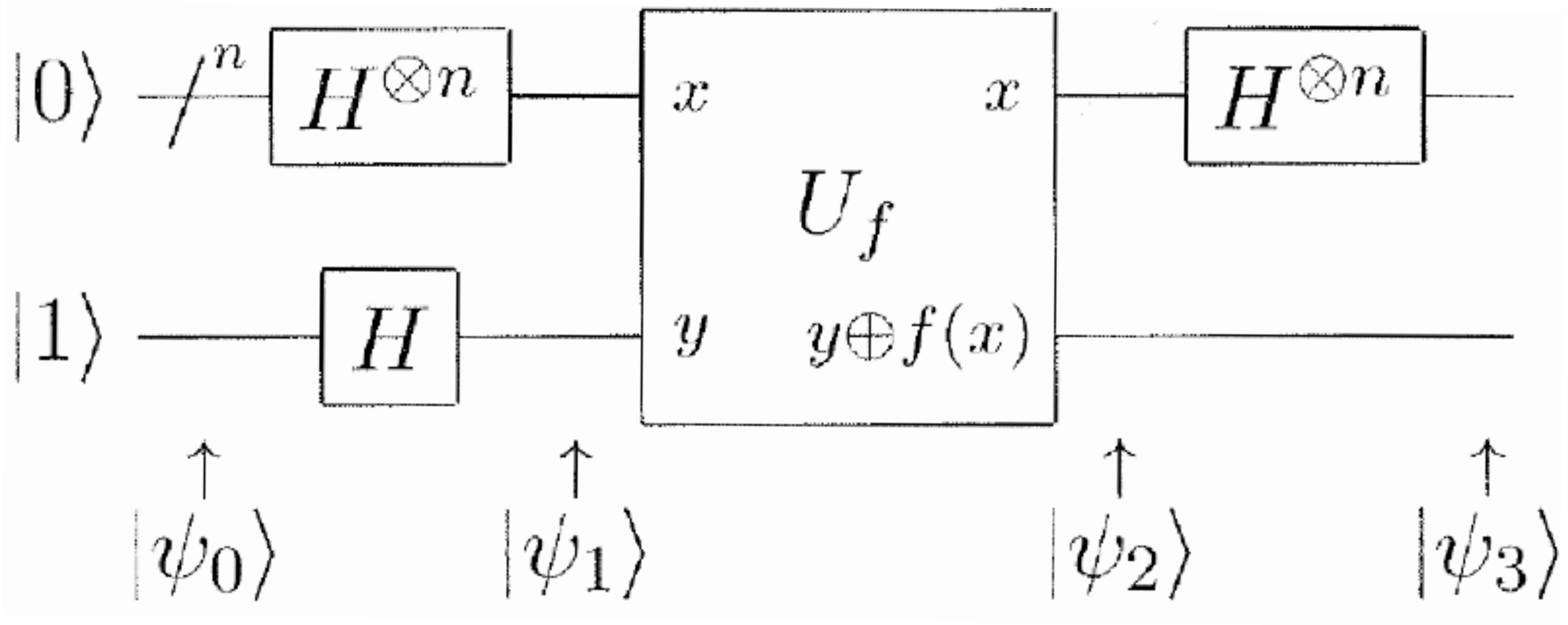
$$|z\rangle = |z_1 z_2 ... z_n\rangle$$

$$|\Psi_3\rangle = \sum_z \sum_x \frac{(-1)^{x\cdot z + f(x)}|z\rangle}{2^n} \left(\frac{|0\rangle - |1\rangle}{\sqrt{2}}\right)$$

From Nielsen Chuang

Deutsch-Jozsa algorithm

- f(x) is a n-bit function: $x \in \{0,1\}^n \rightarrow f(x) \in \{0,1\}$; is f(x) constant or balanced?



$$|z\rangle = |z_1 z_2 ... z_n\rangle$$

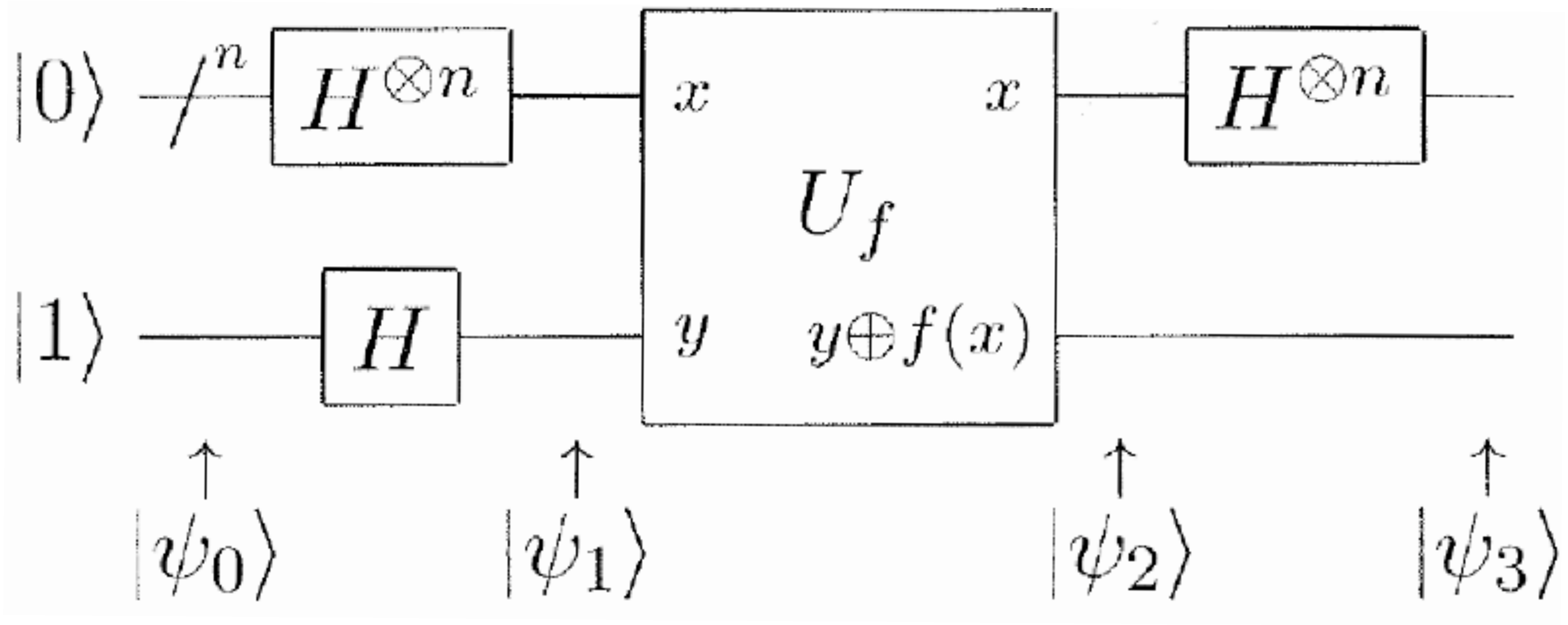
$$|\Psi_3\rangle = \sum_z \sum_x \frac{(-1)^{x \cdot z + f(x)}|z\rangle}{2^n}\left(\frac{|0\rangle - |1\rangle}{\sqrt{2}}\right)$$

Measure upper register

From Nielsen Chuang

Deutsch-Jozsa algorithm

- f(x) is a n-bit function: $x \in \{0,1\}^n \to f(x) \in \{0,1\}$; is f(x) constant or balanced?



$$|z\rangle = |z_1 z_2 ... z_n\rangle$$

$$|\Psi_3\rangle = \sum_z \sum_x \frac{(-1)^{x\cdot z + f(x)}|z\rangle}{2^n} \left(\frac{|0\rangle - |1\rangle}{\sqrt{2}}\right)$$

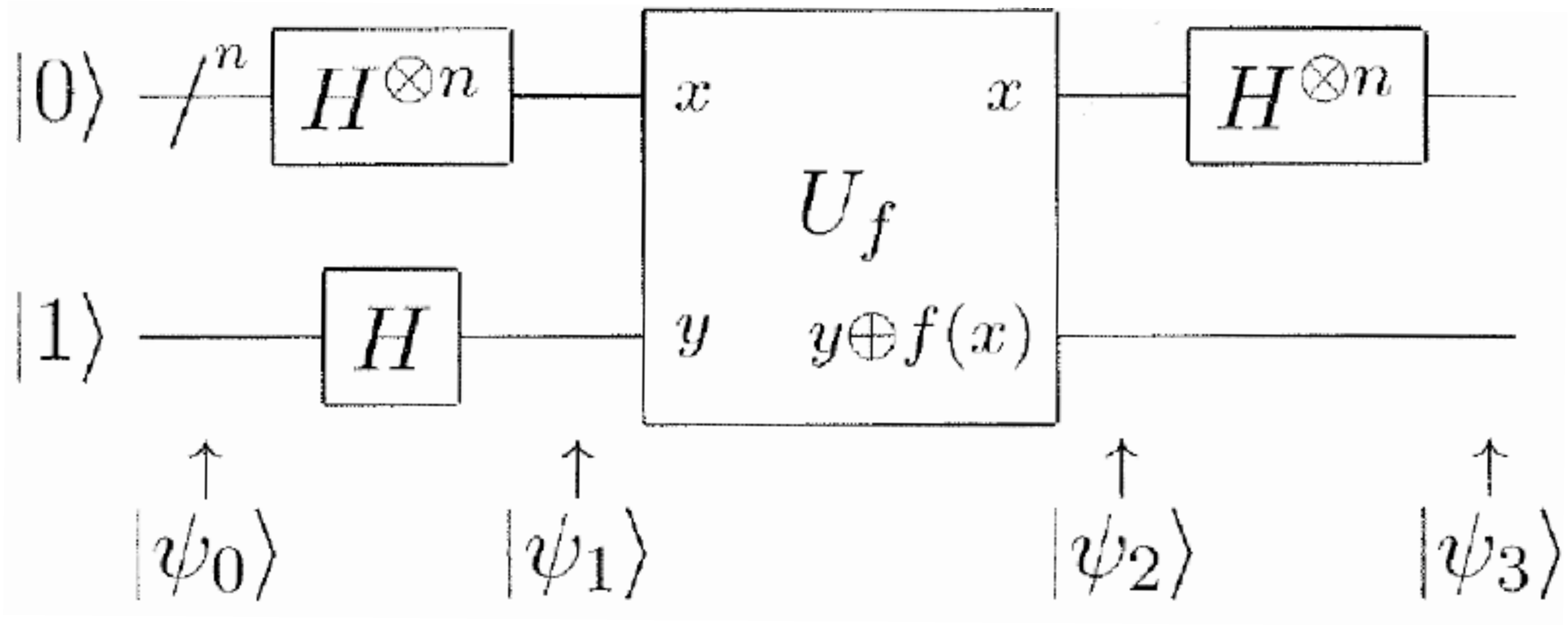
Measure upper register

- The amplitude of |z=00…0> is equal to 1 if f is constant

From Nielsen Chuang

Deutsch-Jozsa algorithm

- f(x) is a n-bit function: $x \in \{0,1\}^n \rightarrow f(x) \in \{0,1\}$; is f(x) constant or balanced?



$$|z\rangle = |z_1 z_2 ... z_n\rangle$$

$$|\Psi_3\rangle = \sum_z \sum_x \frac{(-1)^{x \cdot z + f(x)}|z\rangle}{2^n} \left(\frac{|0\rangle - |1\rangle}{\sqrt{2}}\right)$$

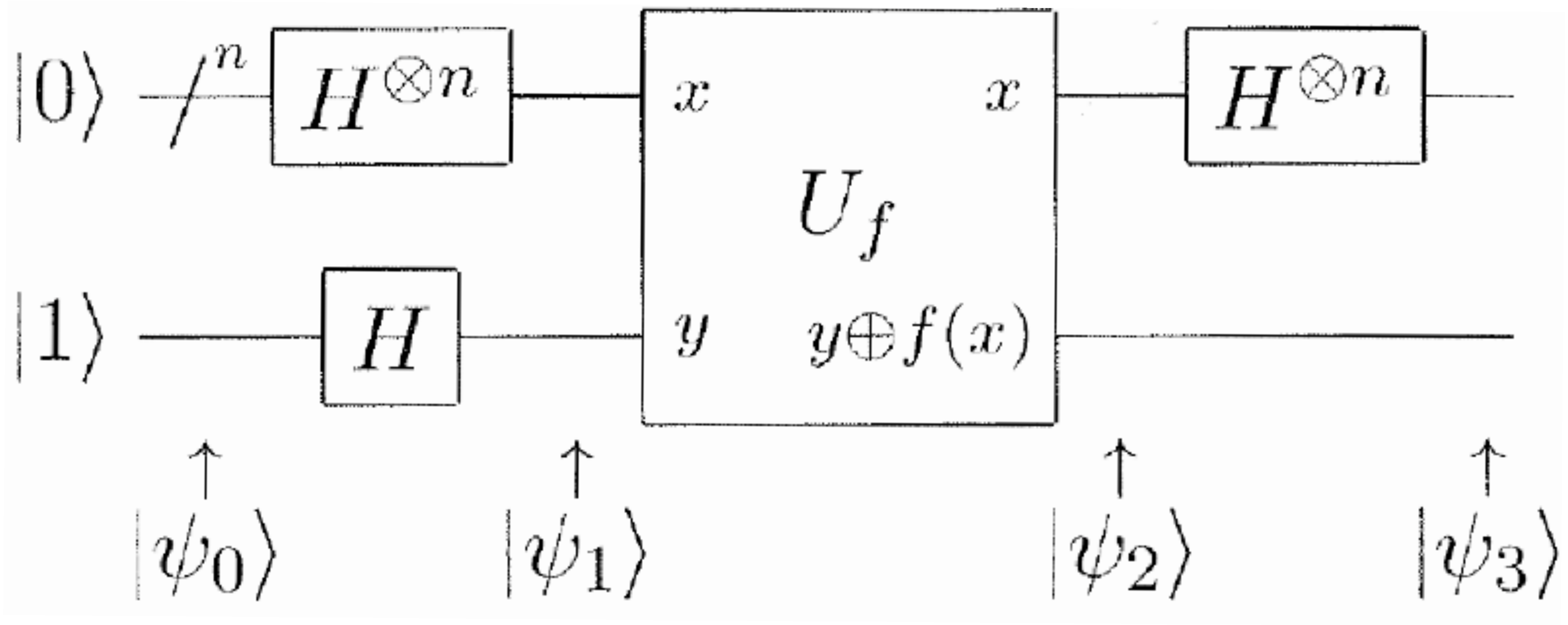
Measure upper register

- The amplitude of |z=00…0> is equal to 1 if f is constant
- The amplitude of |z=00…0> is equal to 0 if f is balanced

From Nielsen Chuang

Deutsch-Jozsa algorithm

- f(x) is a n-bit function: $x \in \{0,1\}^n \to f(x) \in \{0,1\}$; is f(x) constant or balanced?



$$|z\rangle = |z_1 z_2 \ldots z_n\rangle$$

$$|\Psi_3\rangle = \sum_z \sum_x \frac{(-1)^{x\cdot z + f(x)}|z\rangle}{2^n} \left(\frac{|0\rangle - |1\rangle}{\sqrt{2}}\right)$$

Measure upper register

- The amplitude of |z=00…0> is equal to 1 if f is constant
- The amplitude of |z=00…0> is equal to 0 if f is balanced
- Only one run of the quantum algorithm is necessary, vs 2^n/2 +1 classically with probability =1
- However a probabilistic classical algorithm can determine the property efficiently

From Nielsen Chuang

What is the status on quantum algorithms?

One challenge: build a reliable QC

- Environment affects quantum computers by inducing decoherence

Picture from Wikipedia

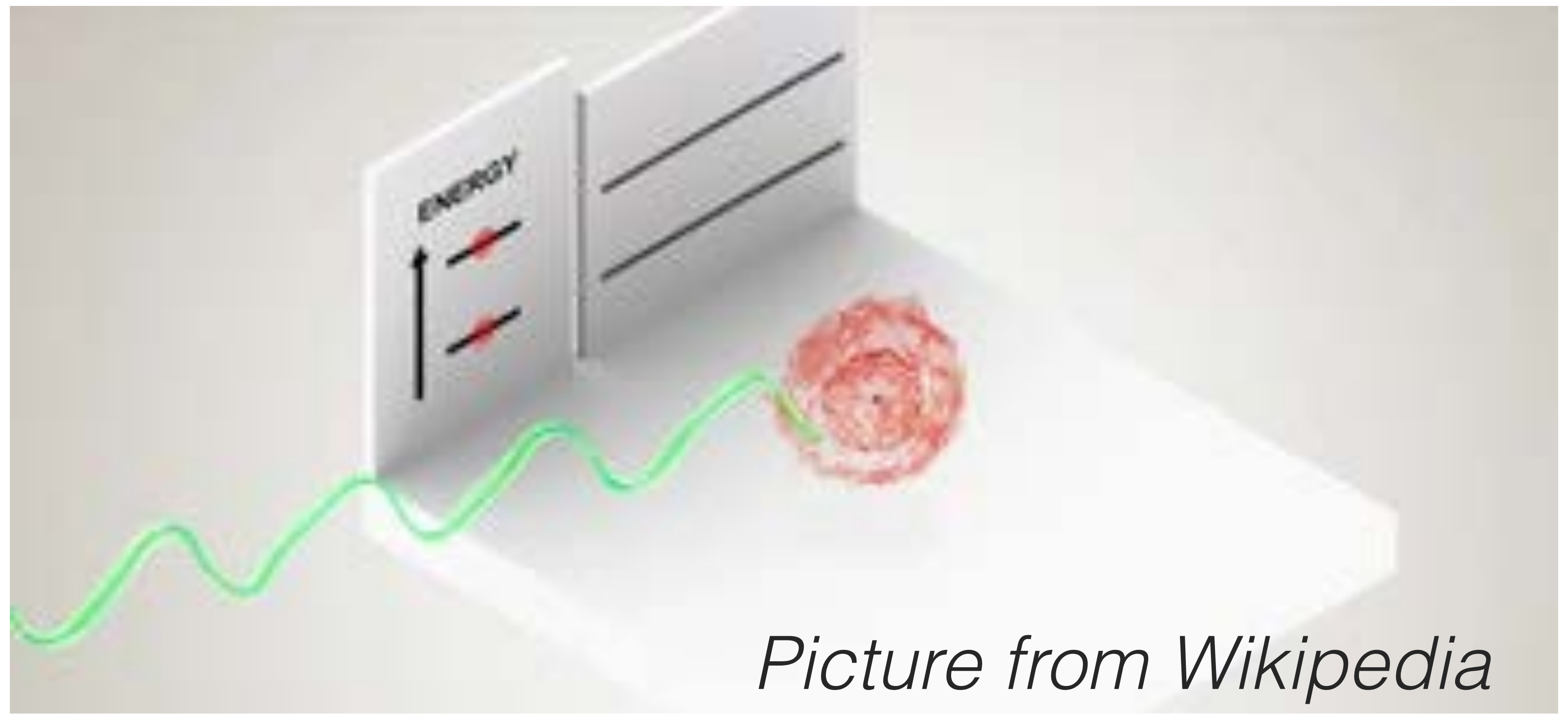
- Redundancy is needed in order to restore quantum information via Quantum Error correction

Repetition code

$$|\Psi\rangle = \alpha|00000\rangle + \beta|11111\rangle$$

One challenge: build a reliable QC

- Environment affects quantum computers by inducing decoherence



Picture from Wikipedia

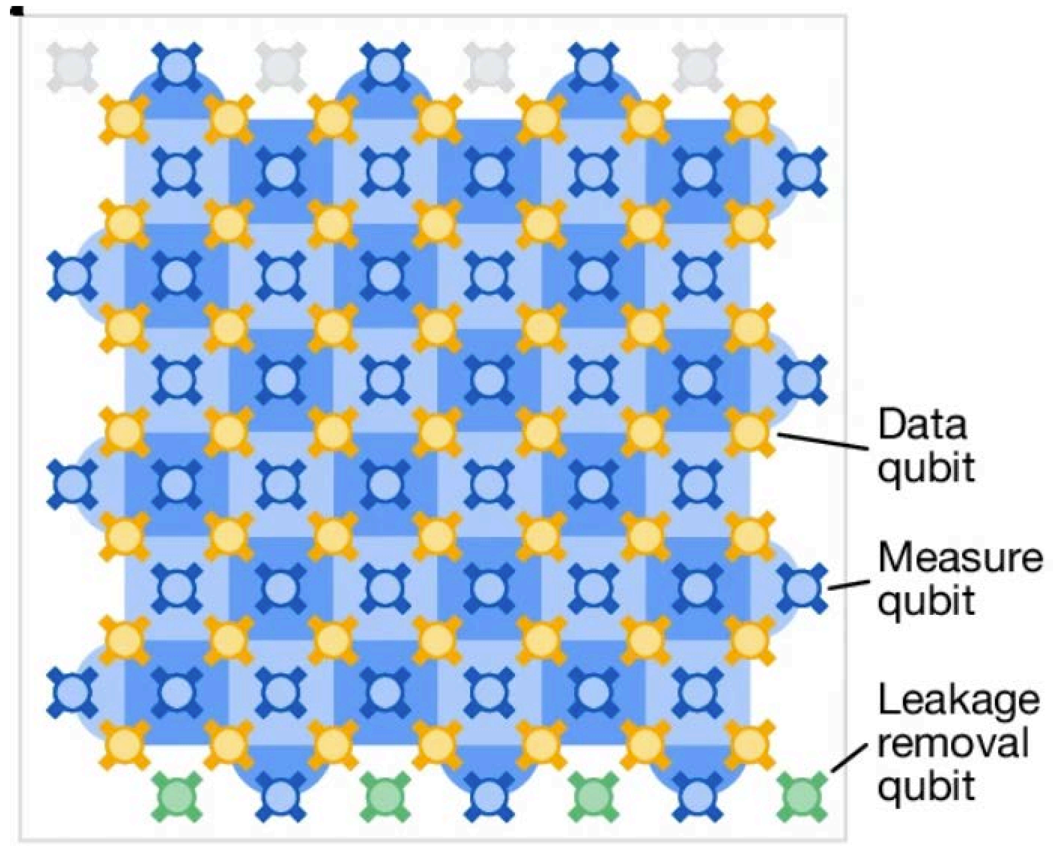
- Redundancy is needed in order to restore quantum information via Quantum Error correction

Repetition code

$$|\Psi\rangle = \alpha|00000\rangle + \beta|11111\rangle$$

Surface code
(d = 7, 105 qubits)

Google AI, Nature 2025



- One million qubits needed to factor a meaningful integer (due to the need for quantum error-correction)

How to factor 2048 bit RSA integers in 8 hours using 20 million noisy qubits

Craig Gidney[1] and Martin Ekerå[2,3]

[1]Google Inc., Santa Barbara, California 93117, USA
[2]KTH Royal Institute of Technology, SE-100 44 Stockholm, Sweden
[3]Swedish NCSA, Swedish Armed Forces, SE-107 85 Stockholm, Sweden

- One million qubits needed to factor a meaningful integer (due to the need for quantum error-correction)

How to factor 2048 bit RSA integers in 8 hours using 20 million noisy qubits

Craig Gidney[1] and Martin Ekerå[2,3]

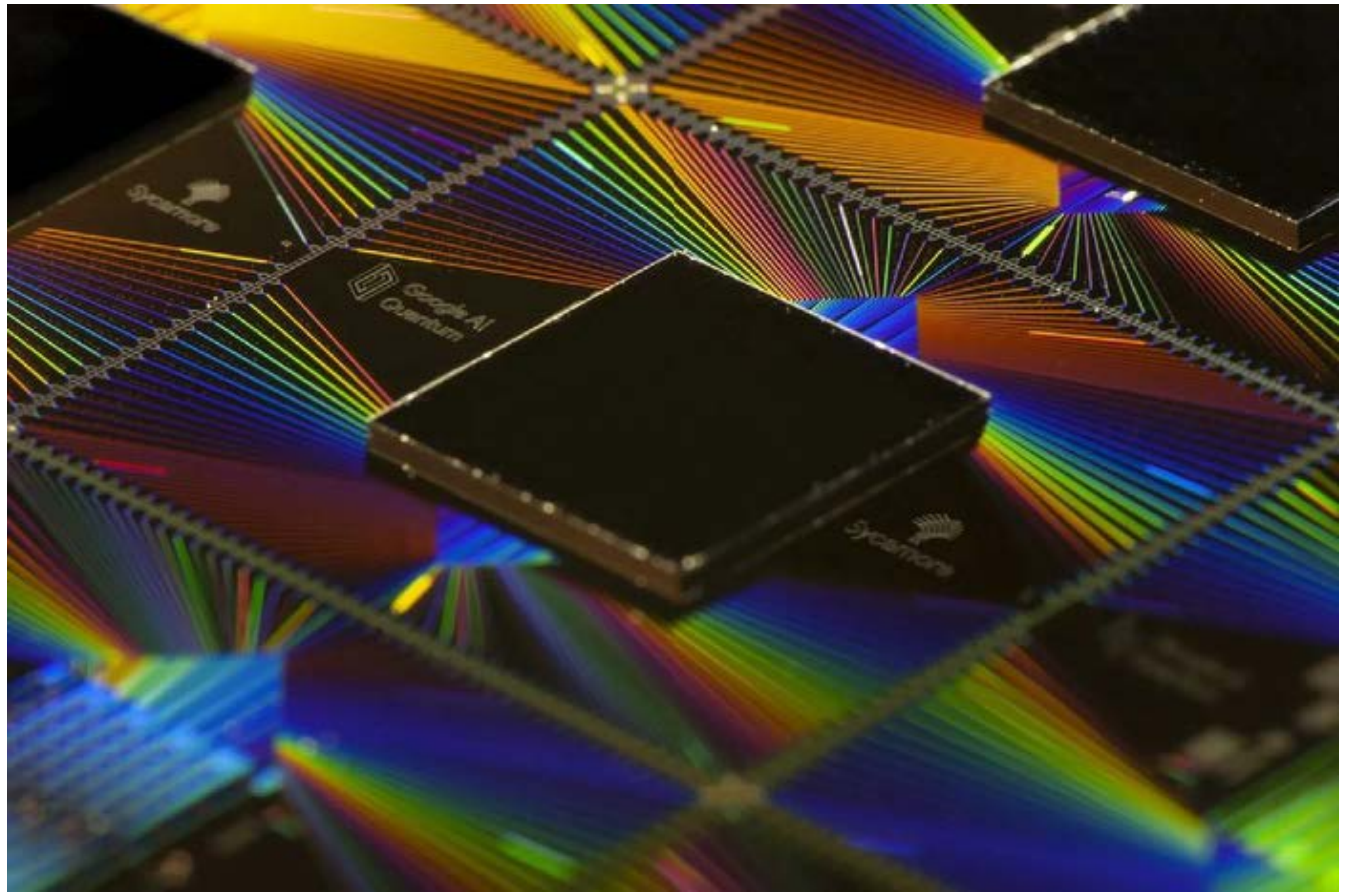
[1]Google Inc., Santa Barbara, California 93117, USA
[2]KTH Royal Institute of Technology, SE-100 44 Stockholm, Sweden
[3]Swedish NCSA, Swedish Armed Forces, SE-107 85 Stockholm, Sweden

- So far there is no "proper" experimental implementation of Shor's algorithm

See: Craig Gidney blow "Why haven't quantum computers factored 21 yet?"
https://algassert.com/post/2500

Quantum primacy models

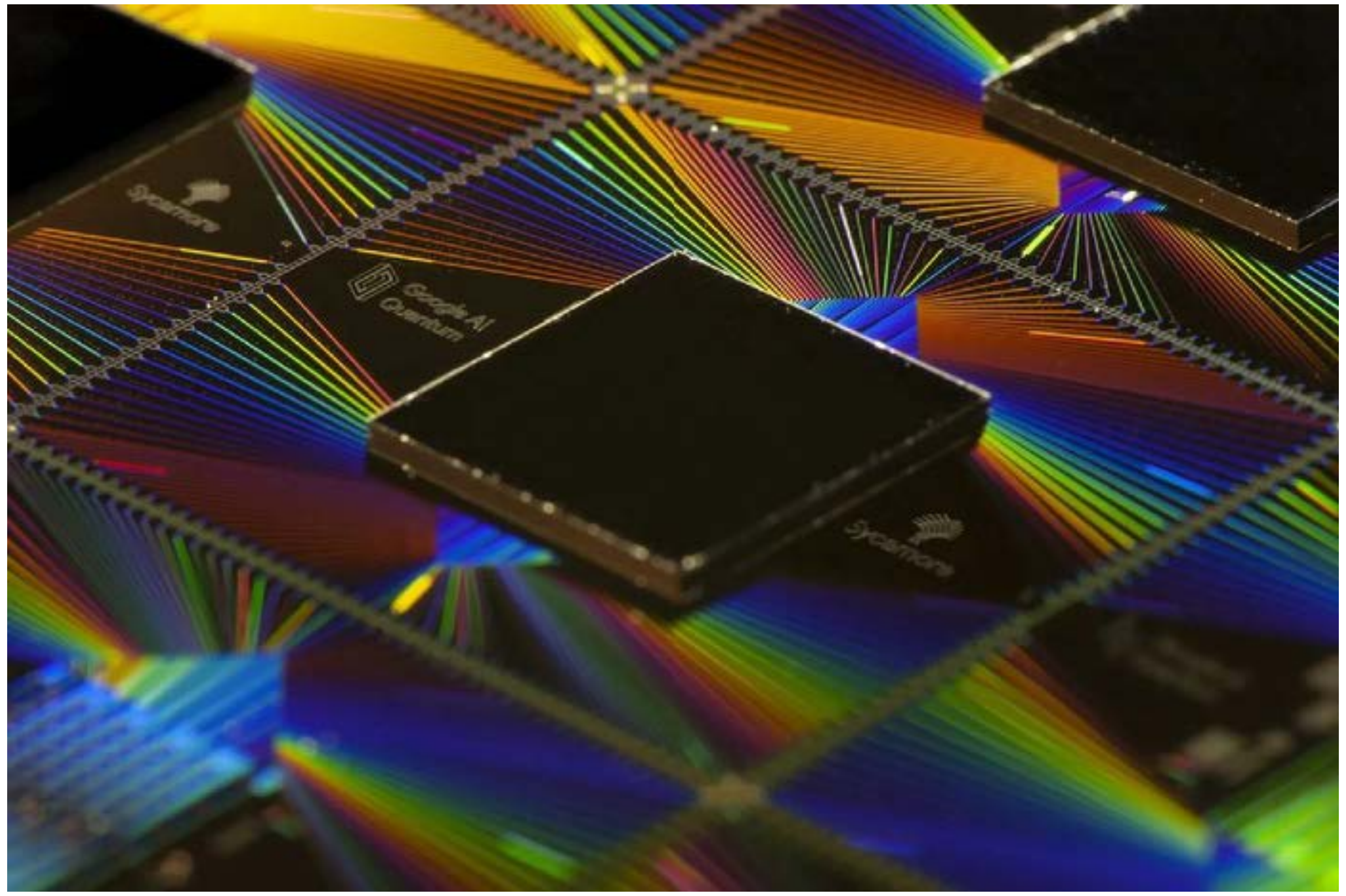
- We have programmable quantum processors of 50-100 qubits
  -> “Quantum primacy” experiments by Google (53 qubits, 2020) & Pann (56 qubits, 2021), Google Willow chip (105 qubits, 2025, see press-release)



Quantum primacy models

- We have programmable quantum processors of 50-100 qubits -> “Quantum primacy” experiments by Google (53 qubits, 2020) & Pann (56 qubits, 2021), Google Willow chip (105 qubits, 2025, see press-release)

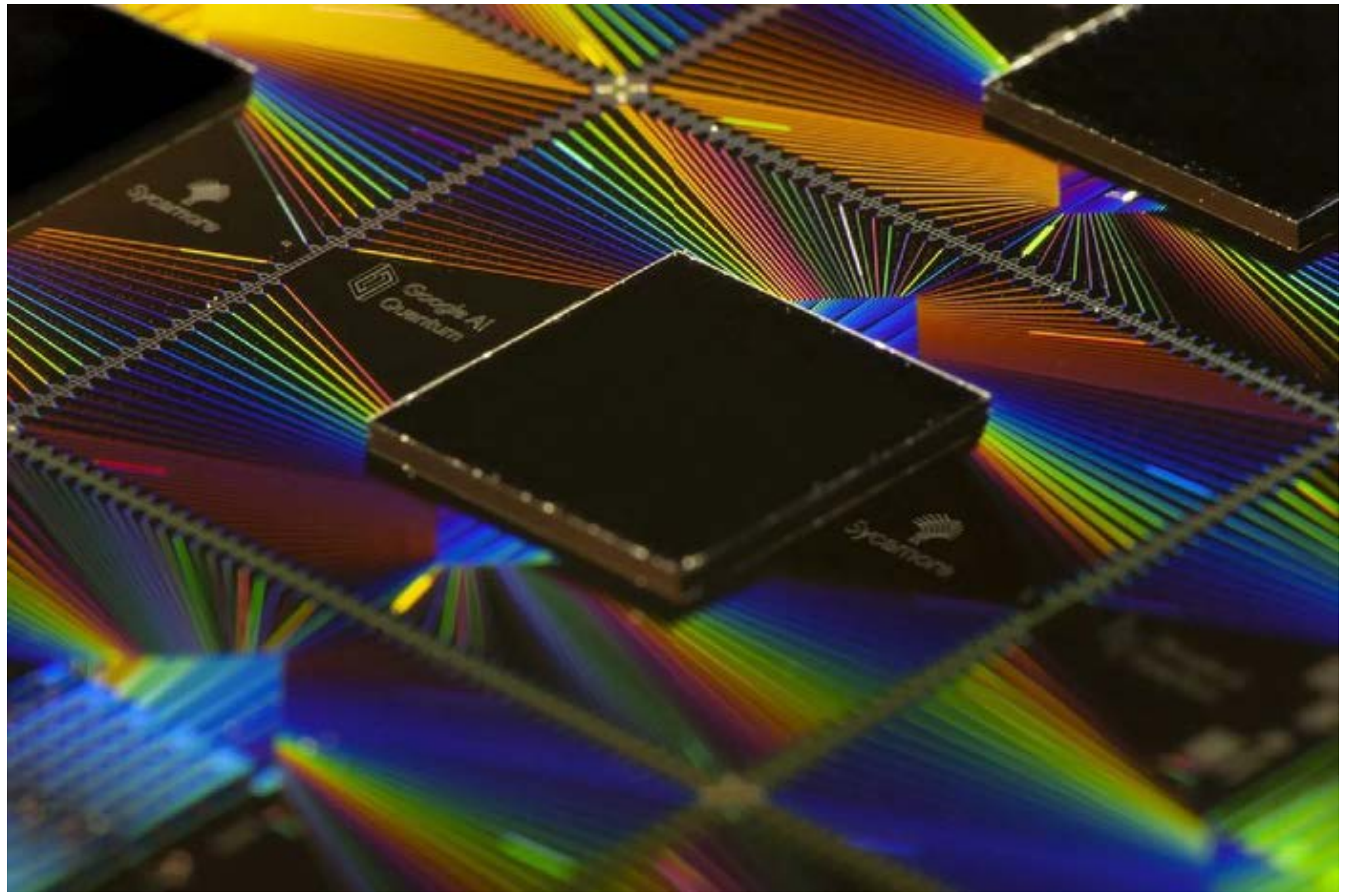
- Capable of solving a task faster than classical computers



Quantum primacy models

WACQT Wallenberg Centre for Quantum Technology

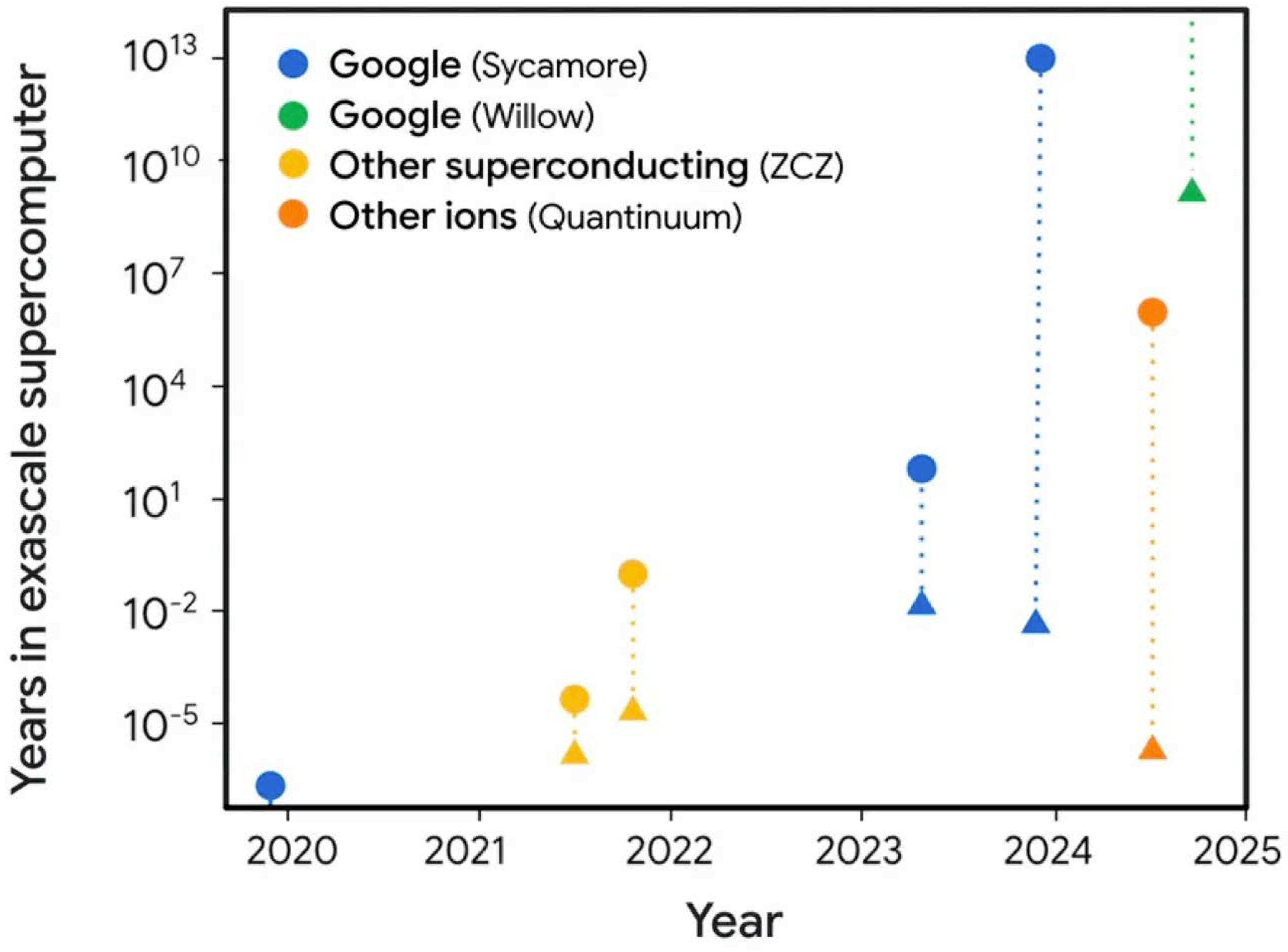
- We have programmable quantum processors of 50-100 qubits
-> “Quantum primacy” experiments by Google (53 qubits, 2020) & Pann (56 qubits, 2021), Google Willow chip (105 qubits, 2025, see press-release)



- Capable of solving a task faster than classical computers

- The task solved faster (sampling) is useless

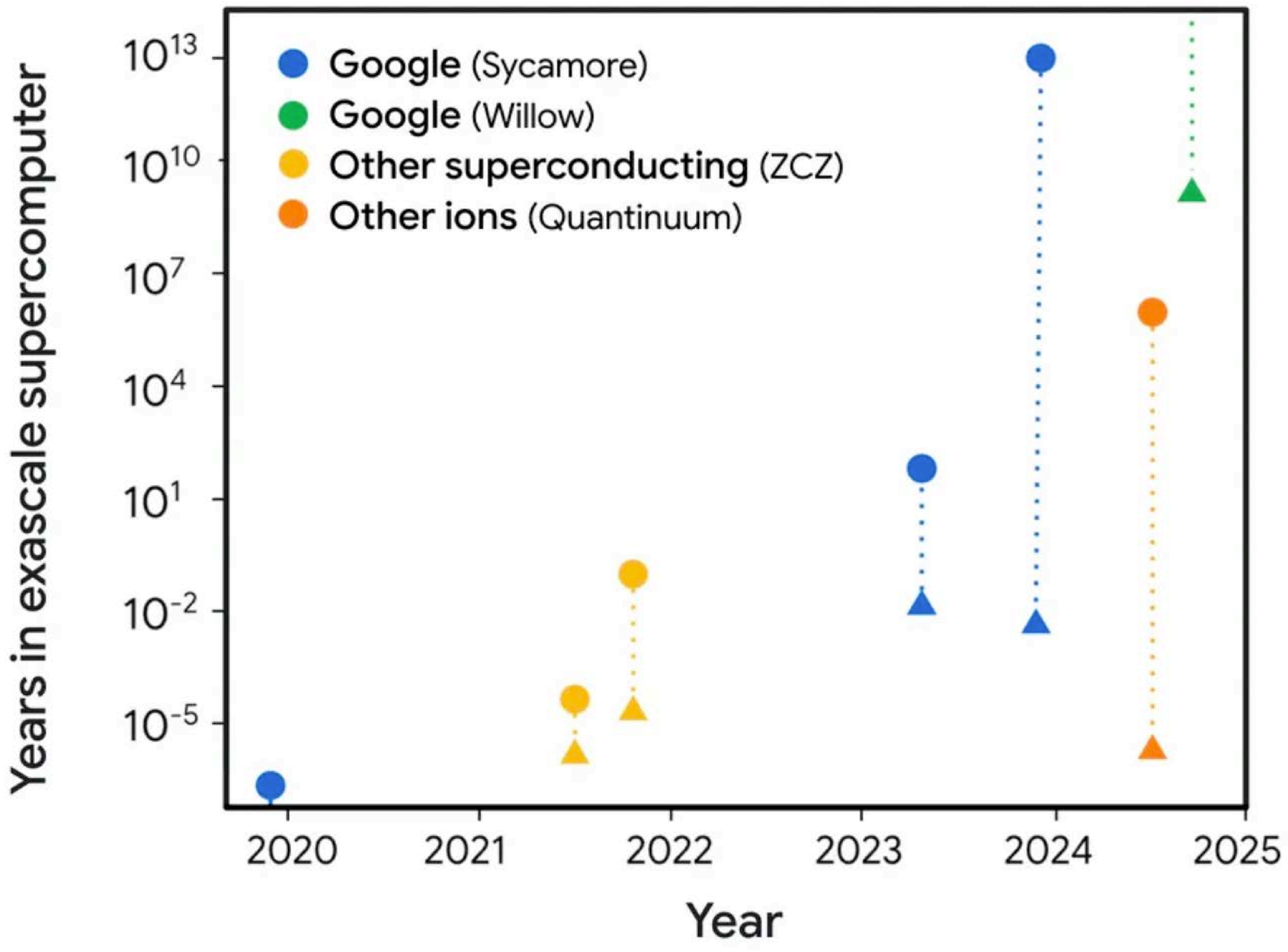
How much faster?



Computational costs are heavily influenced by available memory. Our estimates therefore consider a range of scenarios, from an ideal situation with unlimited memory (▲) to a more practical, embarrassingly parallelizable implementation on GPUs (●).

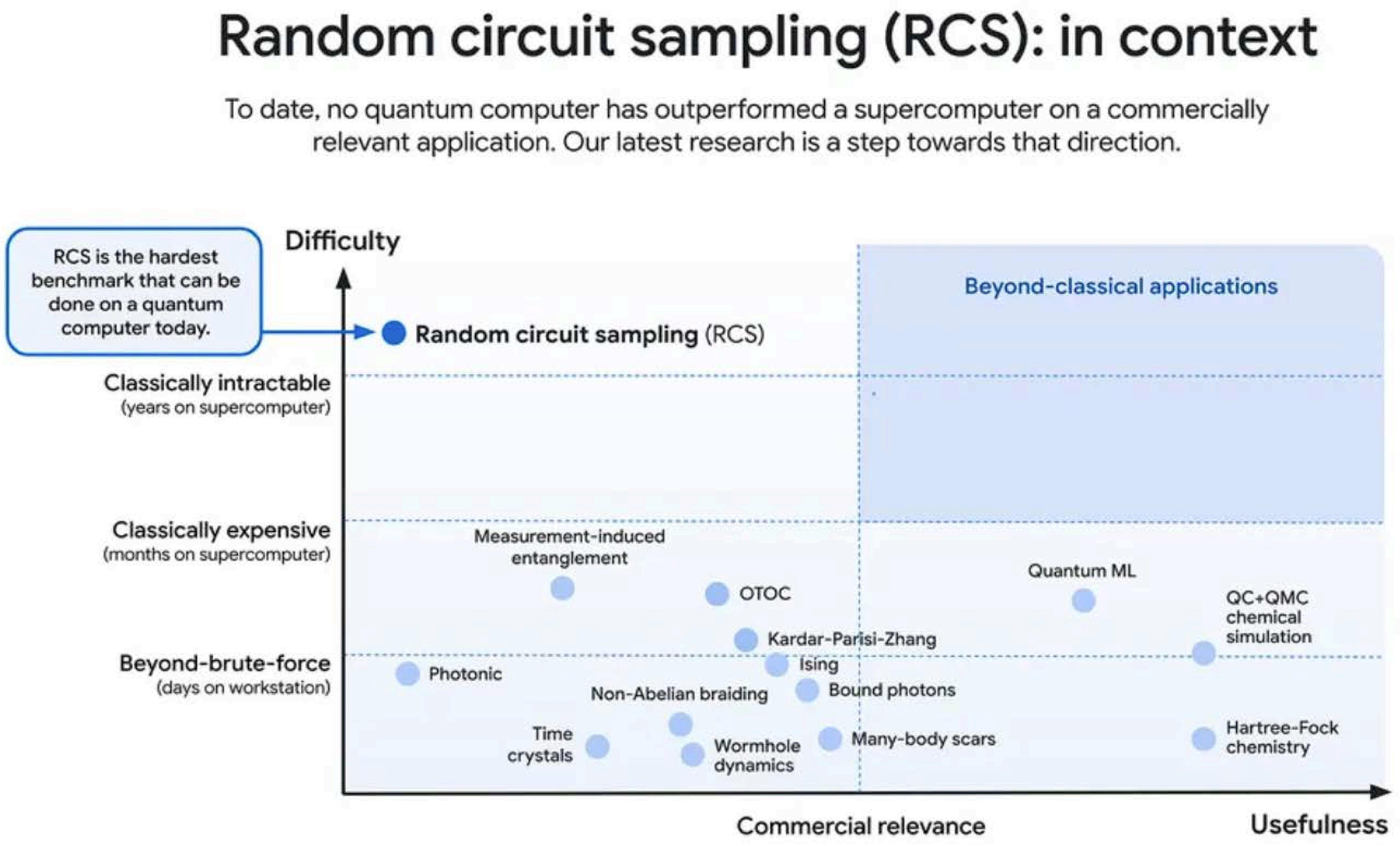
How much faster?

- Took Willow chip 5 minutes, would require 10^{25} years for a normal computer



Computational costs are heavily influenced by available memory. Our estimates therefore consider a range of scenarios, from an ideal situation with unlimited memory (▲) to a more practical, embarrassingly parallelizable implementation on GPUs (●).

How useless?

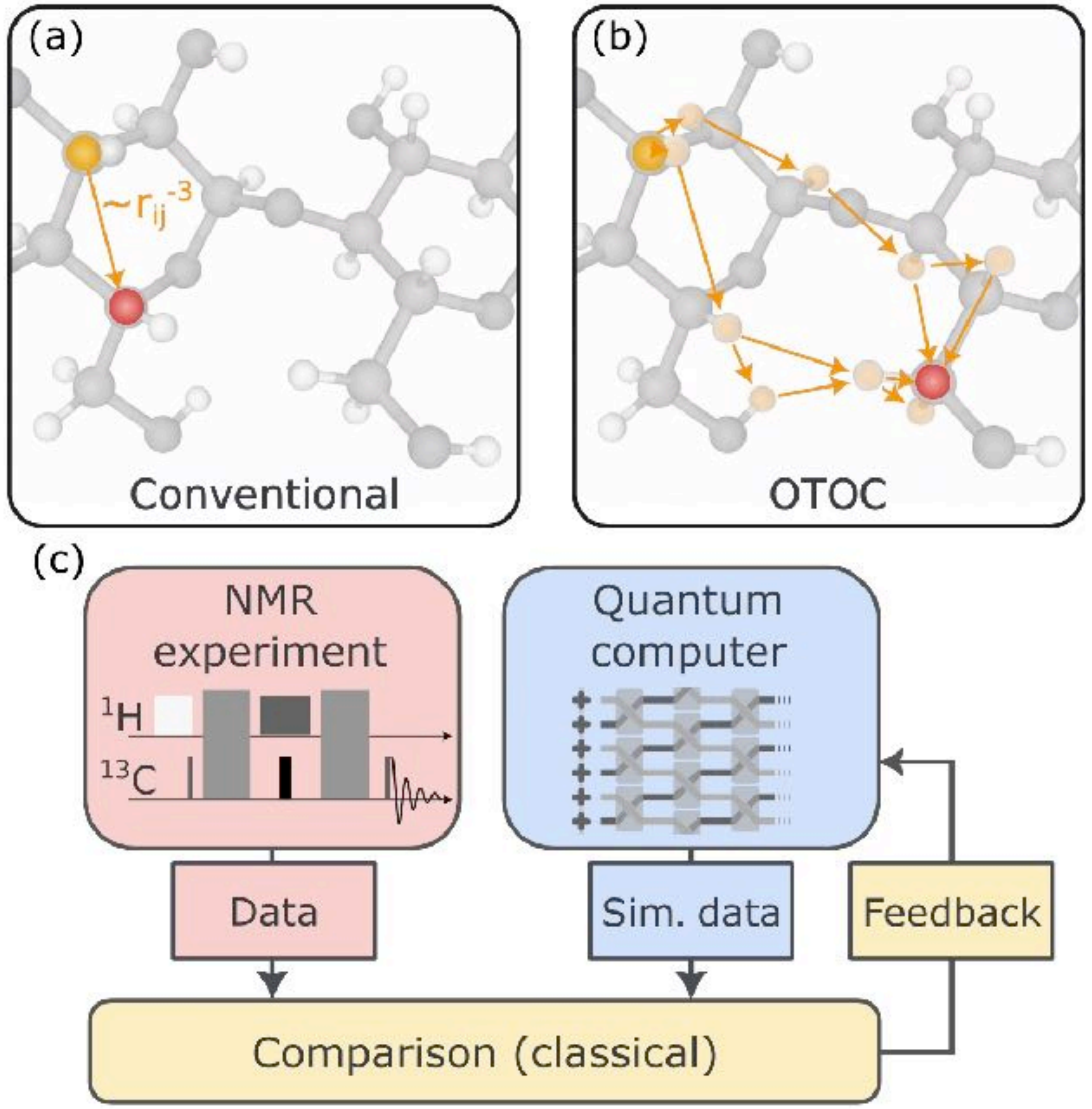


Random circuit sampling (RCS), while extremely challenging for classical computers, has yet to demonstrate practical commercial applications.

Source: Google Quantum AI

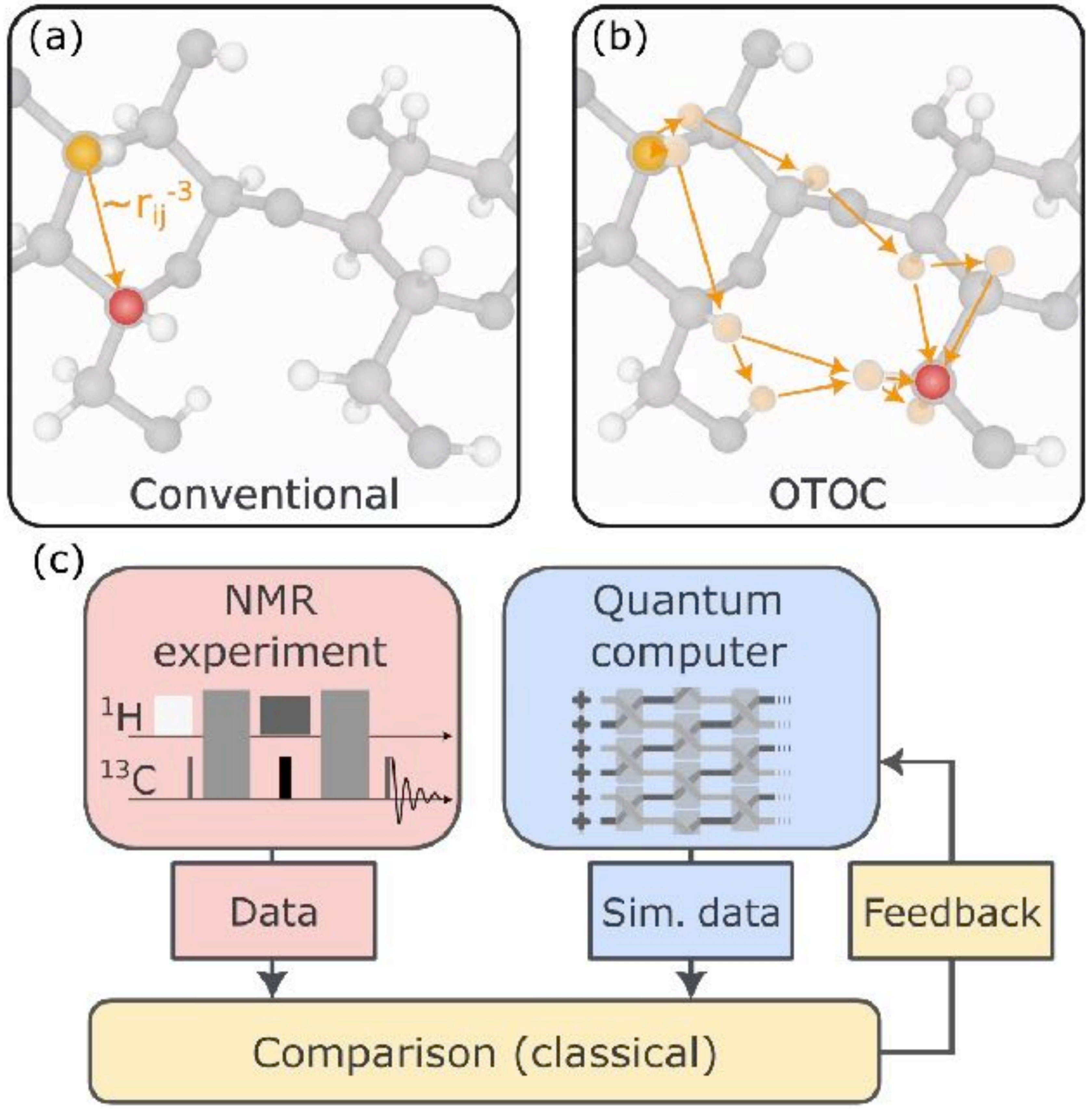
A useful quantum algorithm?

Google claims they now have a first-ever algorithm to achieve verifiable quantum advantage on hardware: "Quantum Echoes"



Google Quantum AI, arXiv:2510.19550, see also arXiv:2510.19751

Google claims they now have a first-ever algorithm to achieve verifiable quantum advantage on hardware: “Quantum Echoes”



Computes out of order time-correlators, and can be used a “molecular ruler” — can measure longer distances than today’s methods, using data from Nuclear Magnetic Resonance (NMR) to gain more information about chemical structure.

Google Quantum AI, arXiv:2510.19550, see also arXiv:2510.19751

Status of quantum algorithms, summary

- Shor’s algorithm requires million of qubits to factor a non-trivial number (with error correction)
- “Useless” quantum advantage has been demonstrated for sampling (Google, Pan, Xanadu)
- It is an open question which kind of problems can be solved on NISQs prototypes (early claims of utility with “Quantum Echoes”)

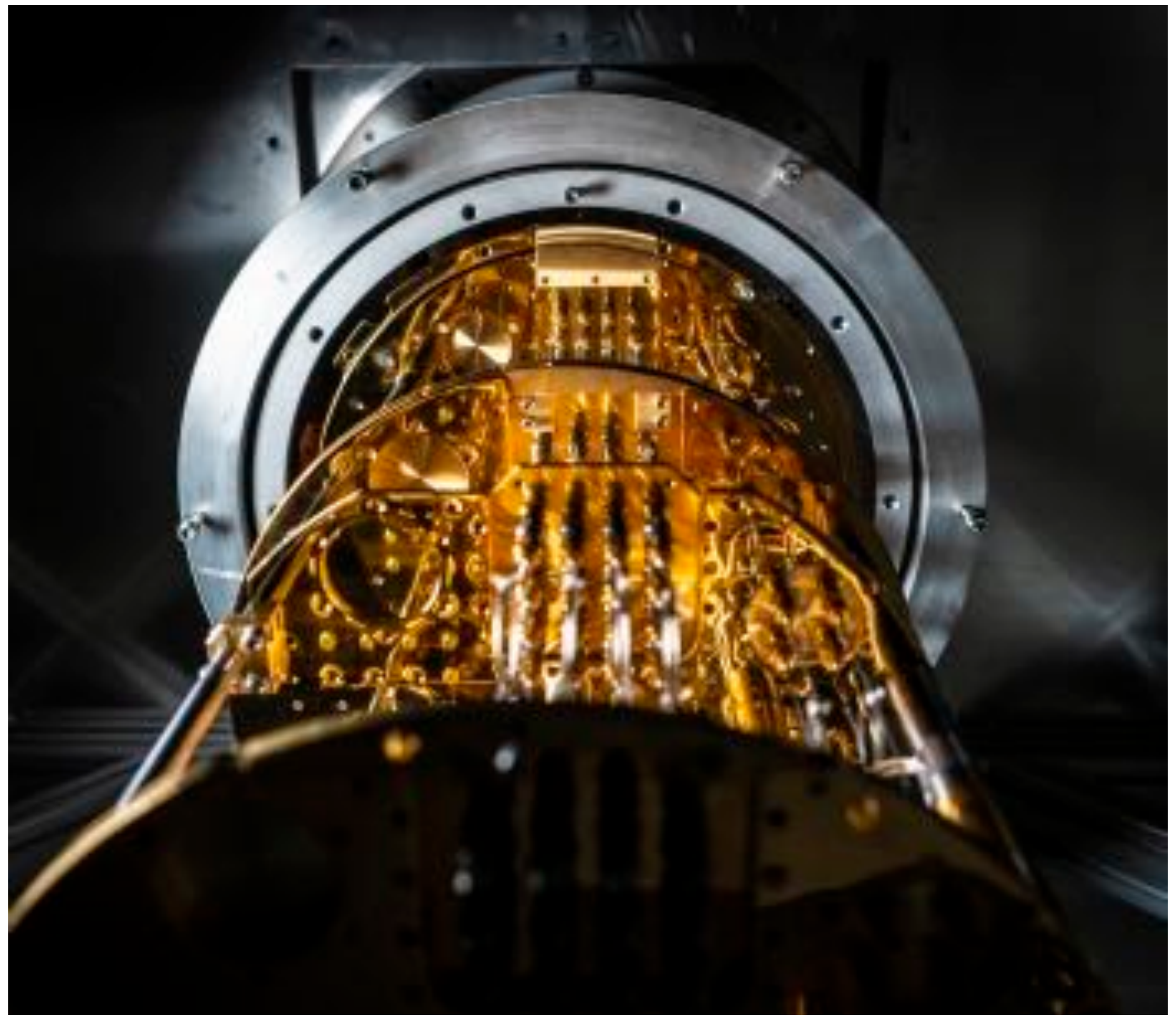
For a comprehensive list of quantum algorithms and their advantage see the quantum algorithm zoo: https://quantumalgorithmzoo.org

Quantum algorithms at Chalmers / in WACQT

Sweden: Wallenberg Center for Quantum Technology

WACQT | Wallenberg Centre for Quantum Technology

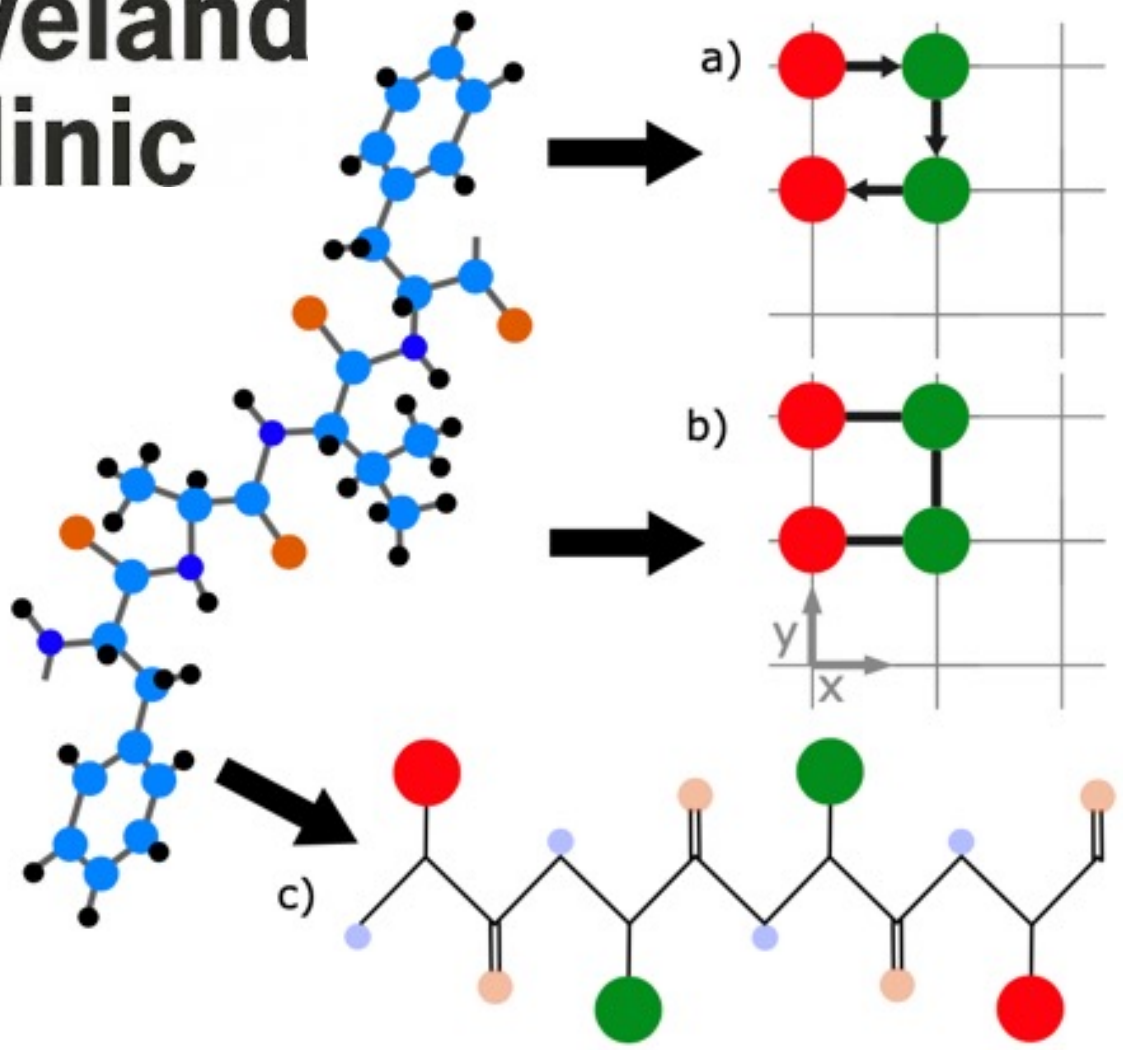
- Main goals:
  (1) To build the Swedish Quantum computer (core project);
  (2) To develop quantum technology know-how in Sweden (excellence project)
- Located (mainly) in: Gothenburg, Stockholm and Lund
- 12 years, (2018-2030)
- Involving industry
- Funding: >150 millions euros (KAW, industry, univ.)
- 200+ researchers (about 100 at Chalmers)



Quantum Computing Applications

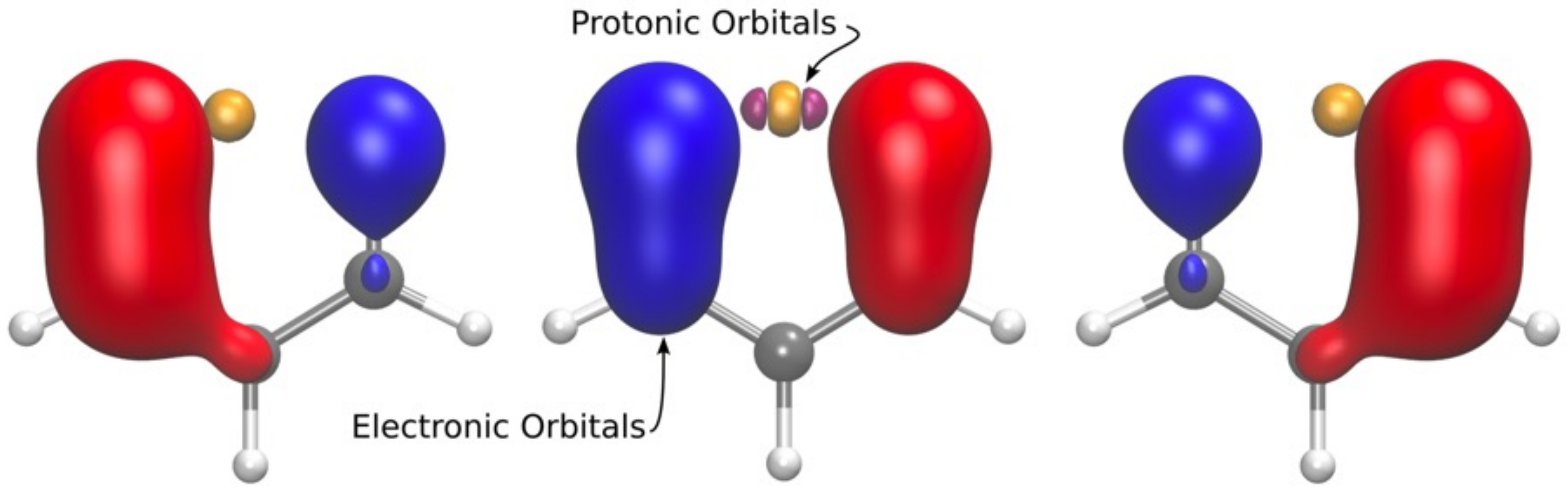
REGION VÄSTRA GÖTALAND
SAHLGRENSKA UNIVERSITY HOSPITAL

Cleveland Clinic



Model **protein folding**: important for biological functionality and illness

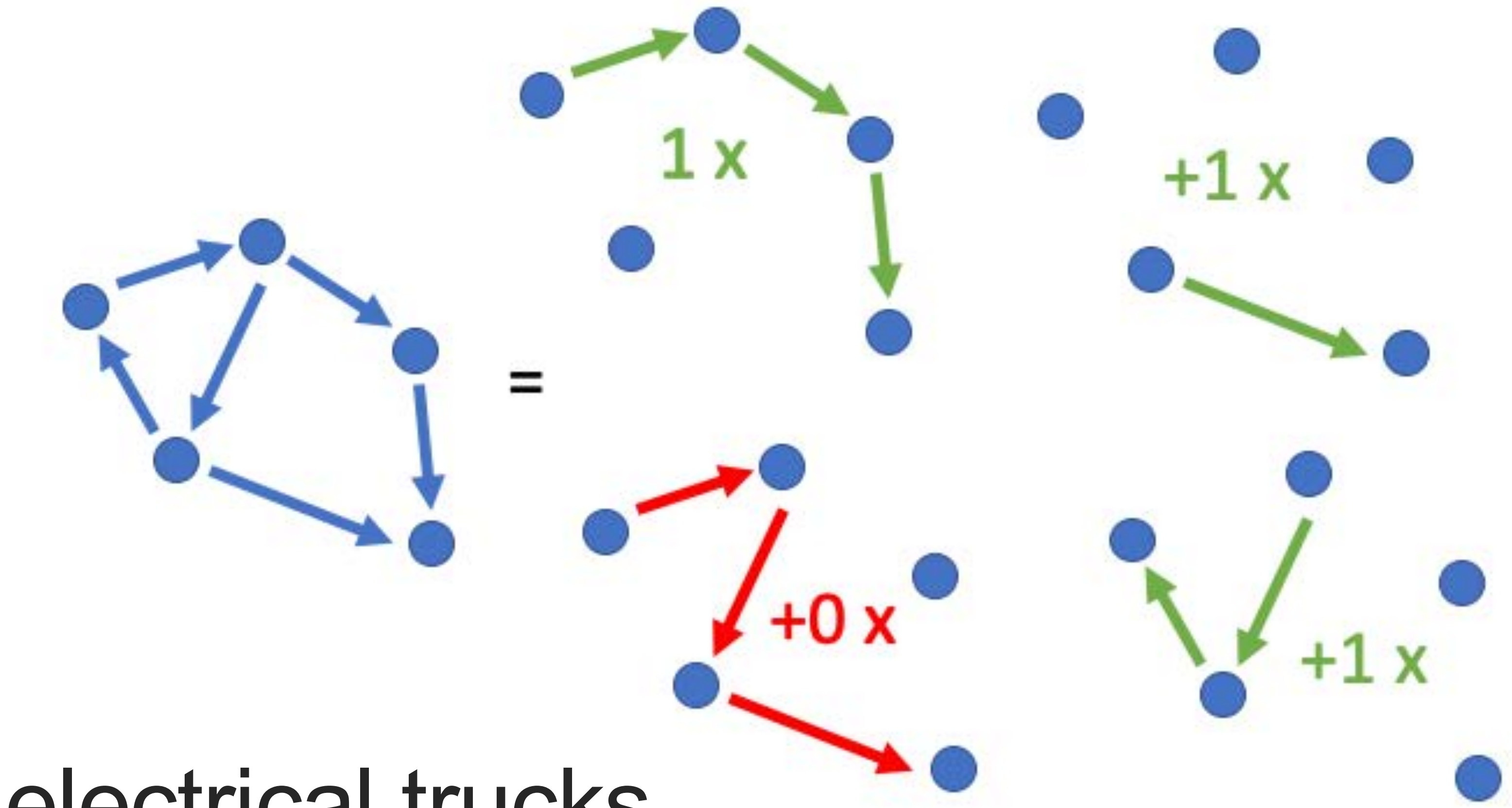
AstraZeneca



Quantum Chemistry –
New possibilities in modeling molecules and materials

JEPPESEN
A BOEING COMPANY

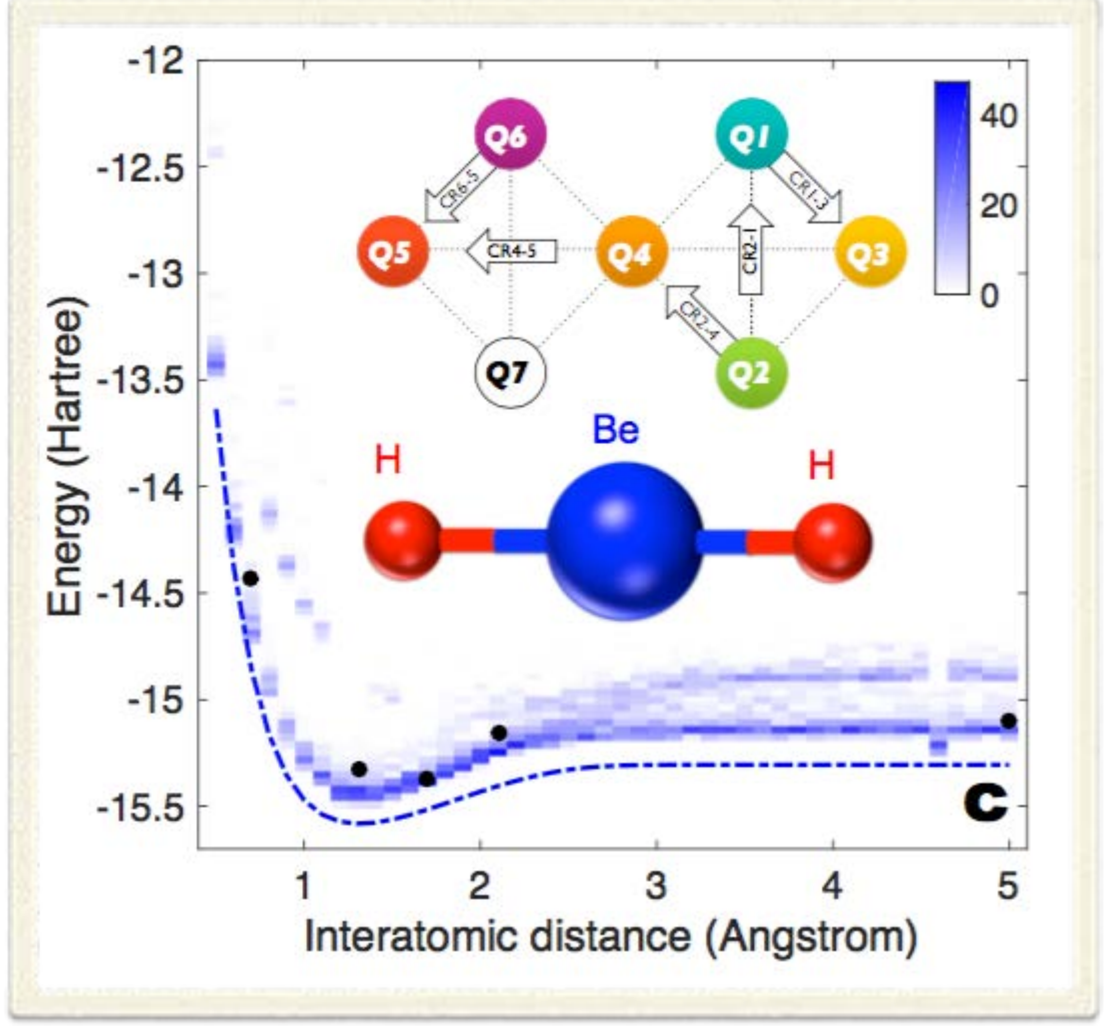
FOI VOLVO



Logistics Optimization –
Find better solutions for e.g. airlines and electrical trucks

Does not replace classical computers. “Combinatorial co-processor.”

- Quantum chemistry -> Design of new drugs and fertilizers



Benchmarking the variational quantum eigensolver through simulation of the ground state energy of prebiotic molecules on high-performance computers

AIP Conference Proceedings **2362**, 030005 (2021); https://doi.org/10.1063/5.0054915

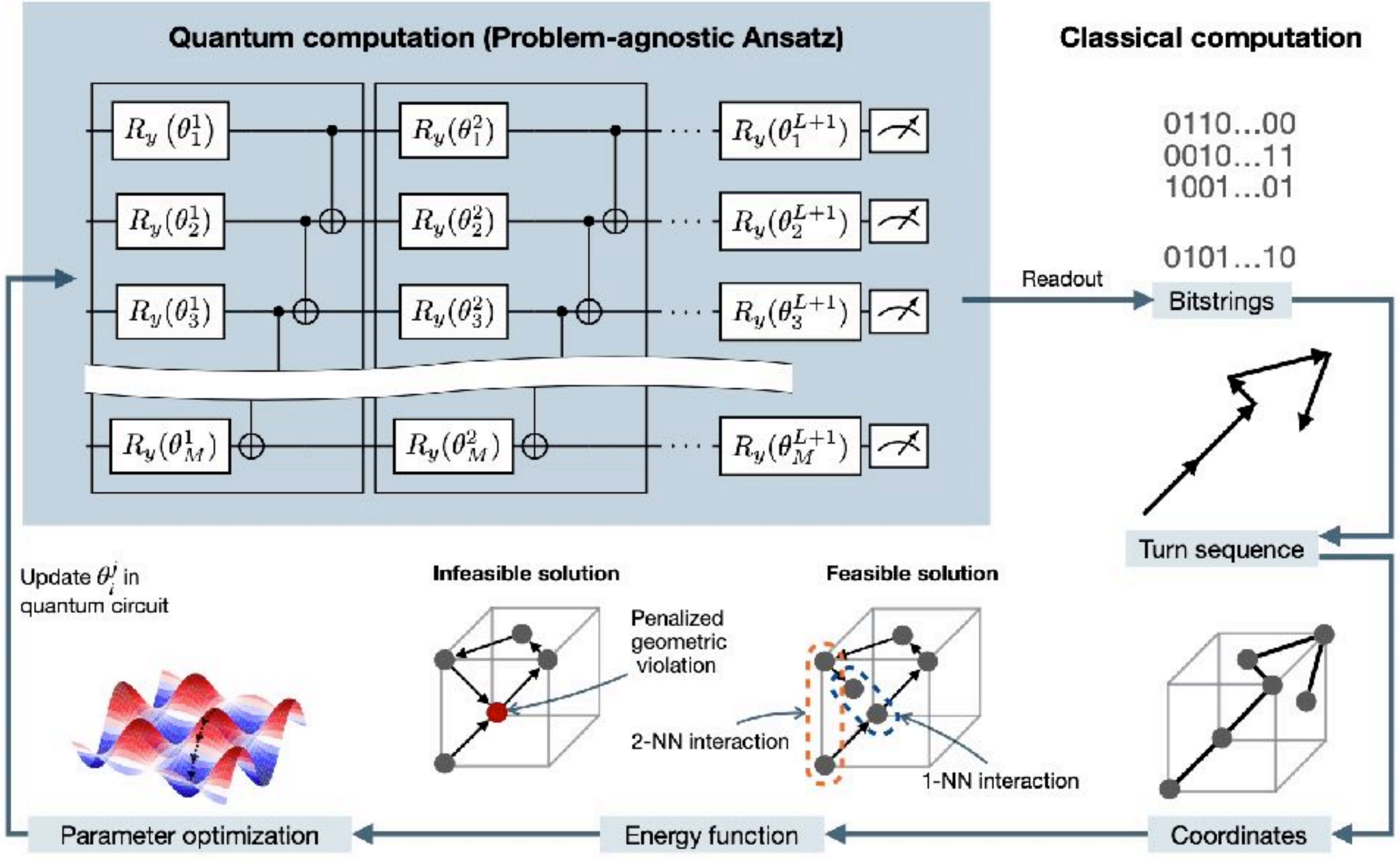
P. Lolur[1,a)], M. Rahm[1,b)], M. Skogh[1,2,c)], L. García-Álvarez[3,d)], *and* G. Wendin[4,e)]

Hide Affiliations

[1)]Department of Chemistry and Chemical Engineering, Chalmers University of Technology, SE-412 96 Gothenburg, Sweden
[2)]Data Science & Modelling, Pharmaceutical Science, R&D, AstraZeneca, Gothenburg, Sweden
[3)]Applied Quantum Physics Laboratory, Department of Microtechnology and Nanoscience-MC2, Chalmers University of Technology, SE-412 96 Gothenburg, Sweden
[4)]Quantum Technology Laboratory, Department of Microtechnology and Nanoscience-MC2, Chalmers University of Technology, SE-412 96 Gothenburg, Sweden

See works by Martin Rahm's group

Hardware efficient ansatz (HEA) for protein folding



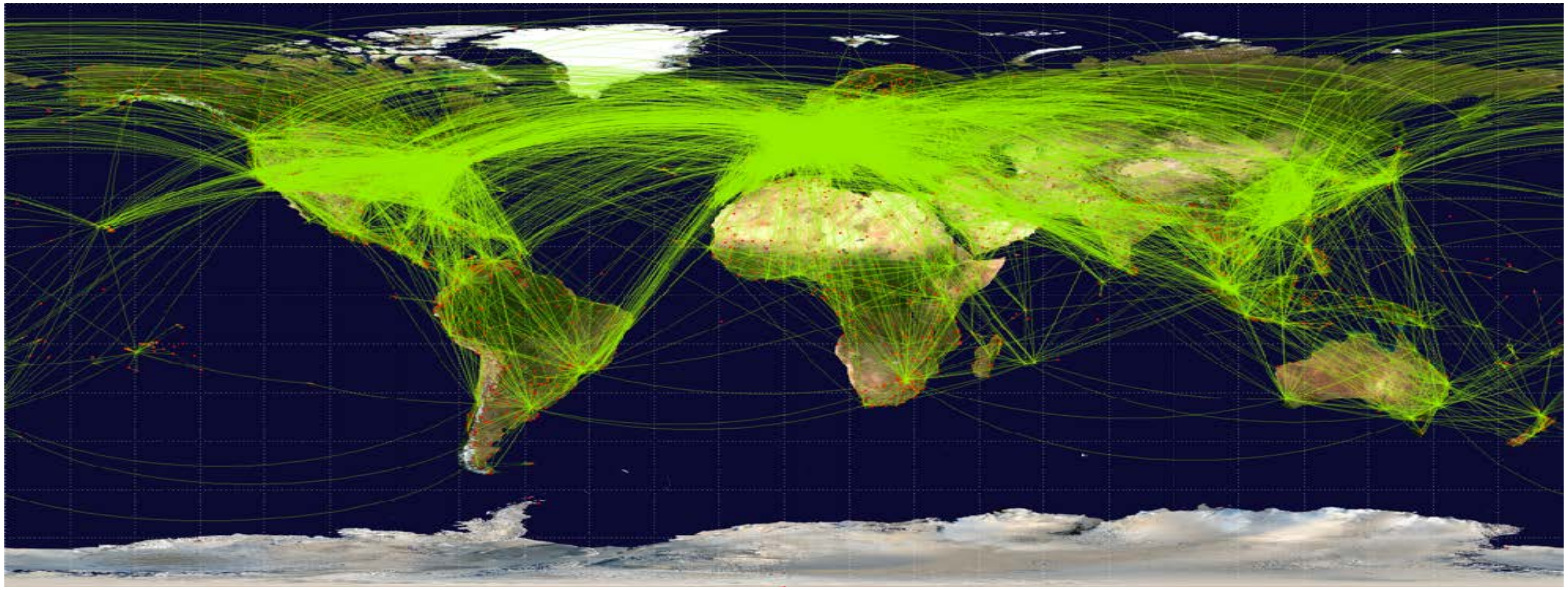
Efficient Quantum Protein Structure Prediction with Problem-Agnostic Ansatzes

arXiv:2509.18263

See works by Laura García-Álvarez's and G. Johansson's group

E.g., optimize aircraft (= tail) assignment: assigning aircraft to routes

Assign 100 guests to 100 chairs = 100*99*98* … *3*2*1 ≈ 10^{157} configurations



Each trial route maps to a qubit

$$H_C = \sum_{i<j} J_{ij}\sigma_i^z\sigma_j^z + \sum_{i=1}^{n} h_i\sigma_i^z$$

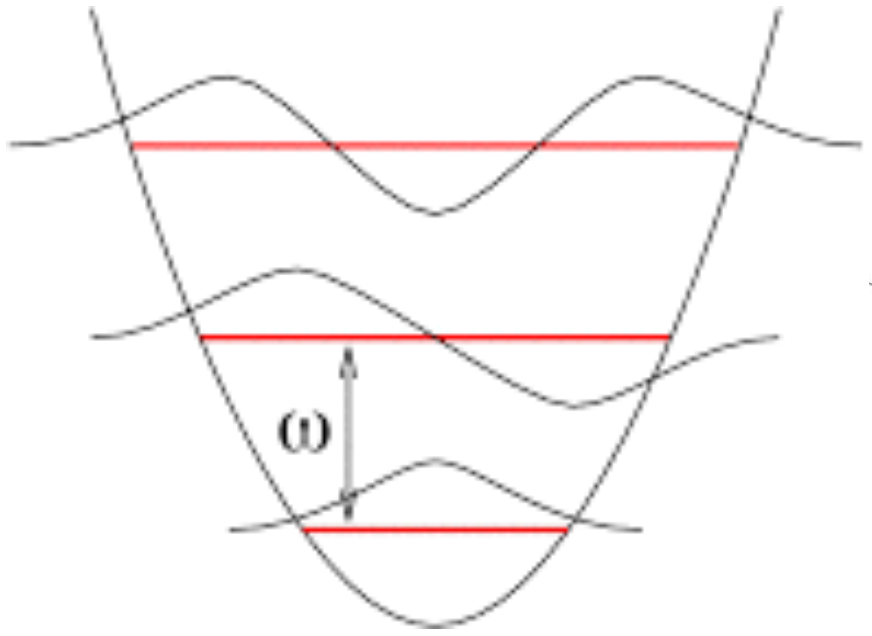
Map optimization to the ground state of an Ising hamiltonian.

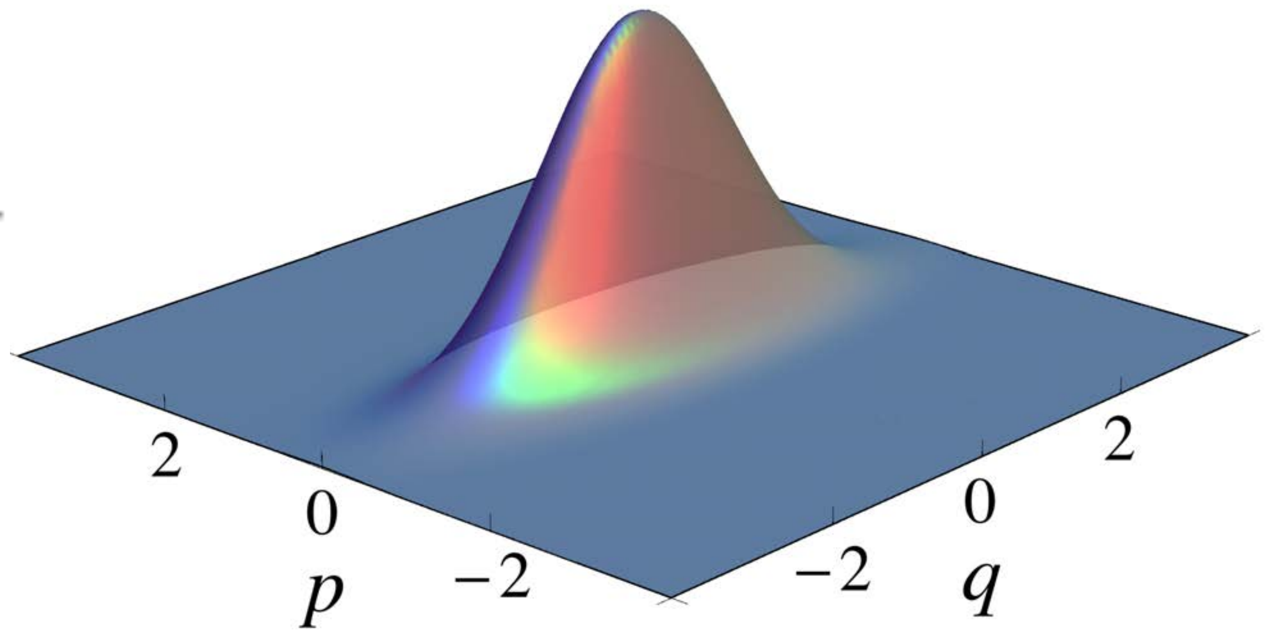
Find the ground state using QAOA.

P. Vikstål, M Grönkvist, M Svensson, M Andersson, G Johansson, G Ferrini, Phys. Rev. Applied 14, 034009 (2020)

JEPPESEN A BOEING COMPANY

What do I do? QC with the quantized harmonic oscillator





Phase space = Complex Plane

Infinite-dimensional Hilbert space

Example of operation:

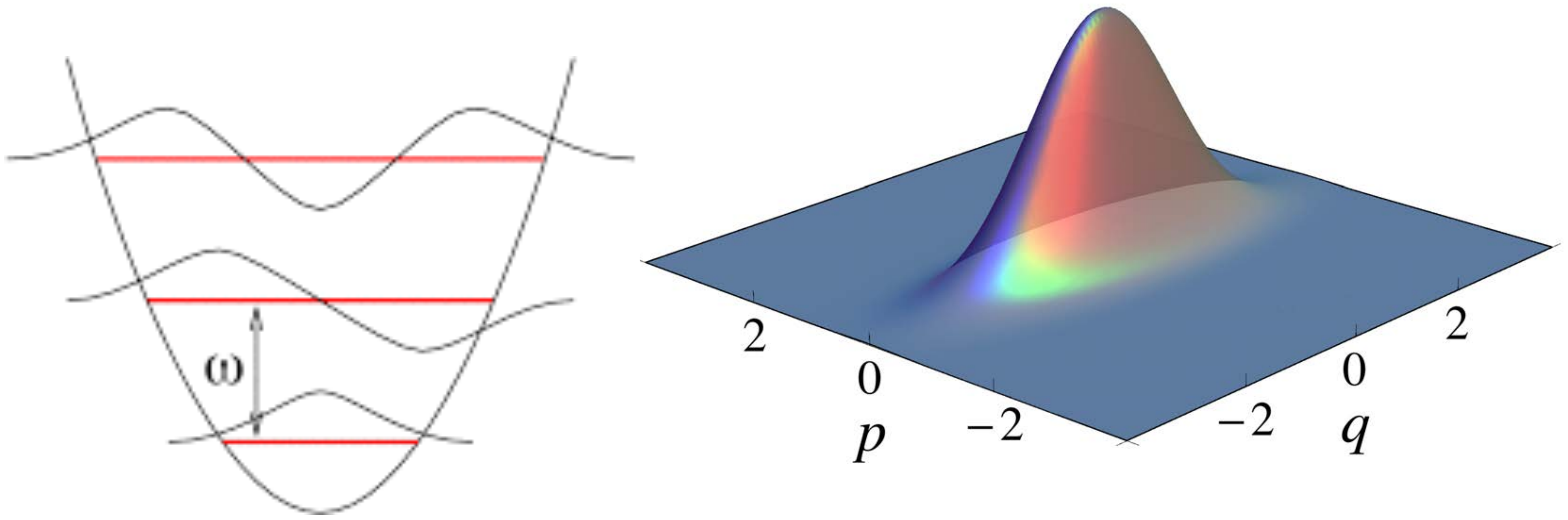
$$X(s) = e^{-is\hat{p}}$$

$$Z(s) = e^{is\hat{q}}$$

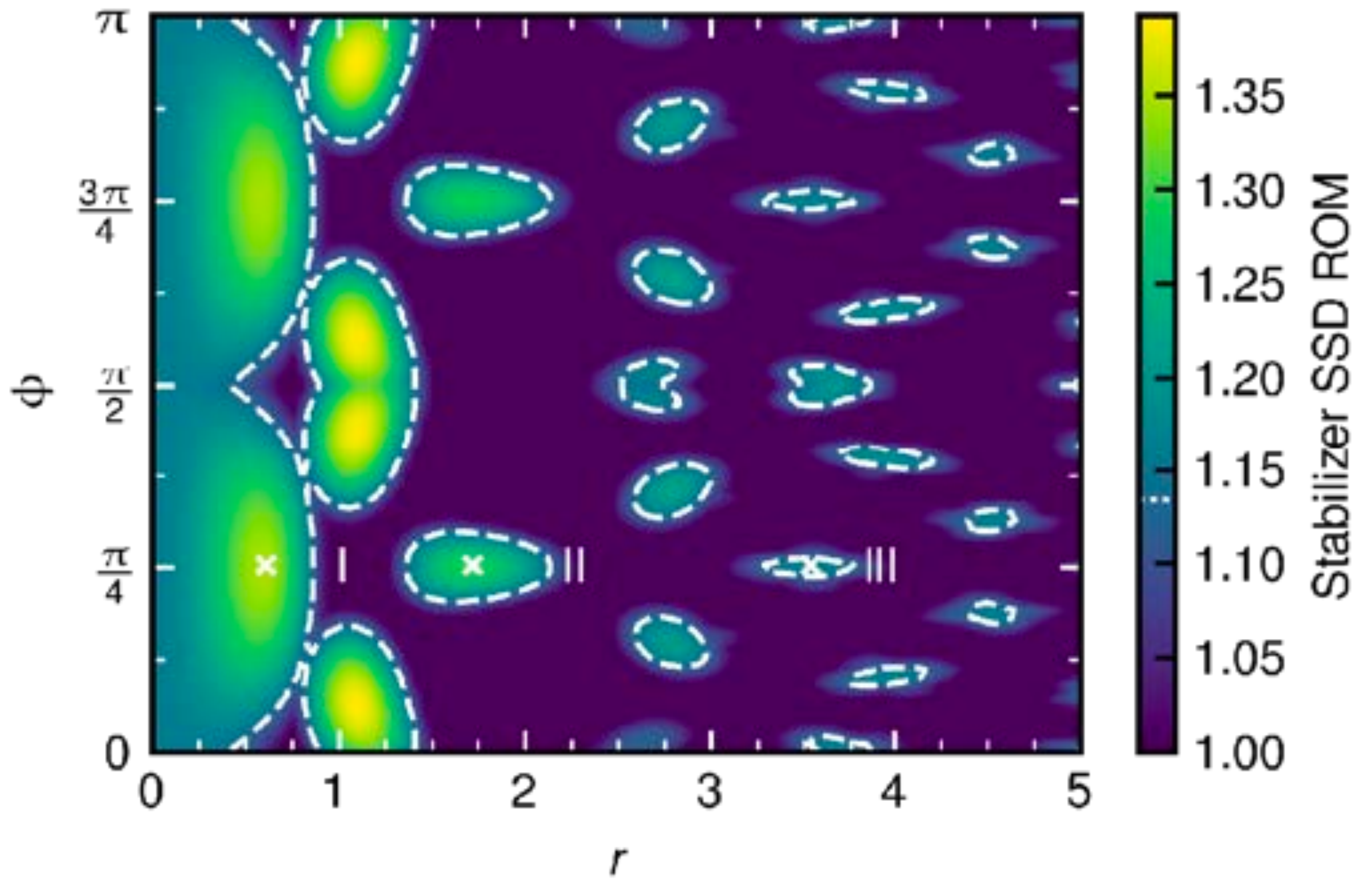
Feel free to ask me!

What do I do? QC with the quantized harmonic oscillator

WACQT Wallenberg Centre for Quantum Technology



Quantum resource theory for quantum computation: how resourceful bosonic states are?



Phase space = Complex Plane

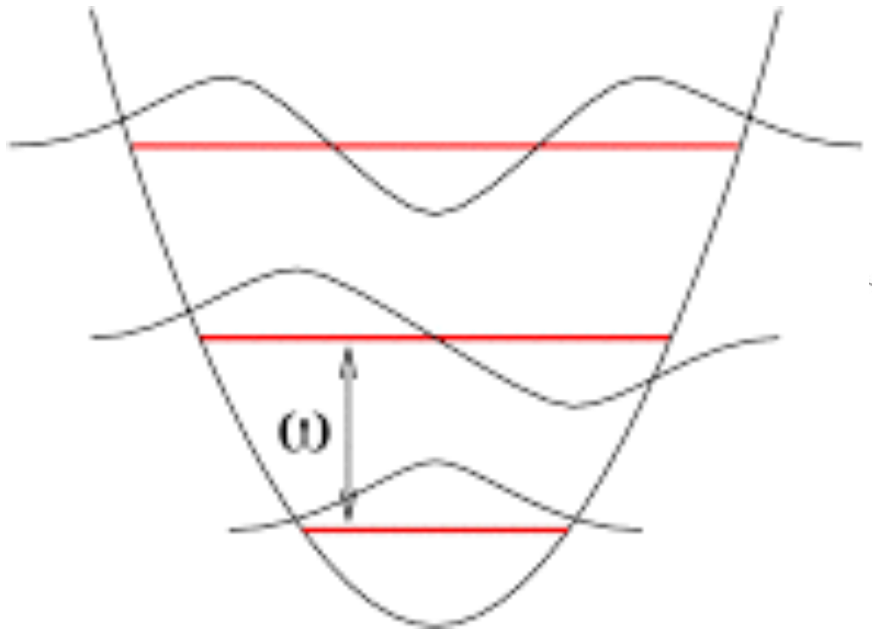
Infinite-dimensional Hilbert space

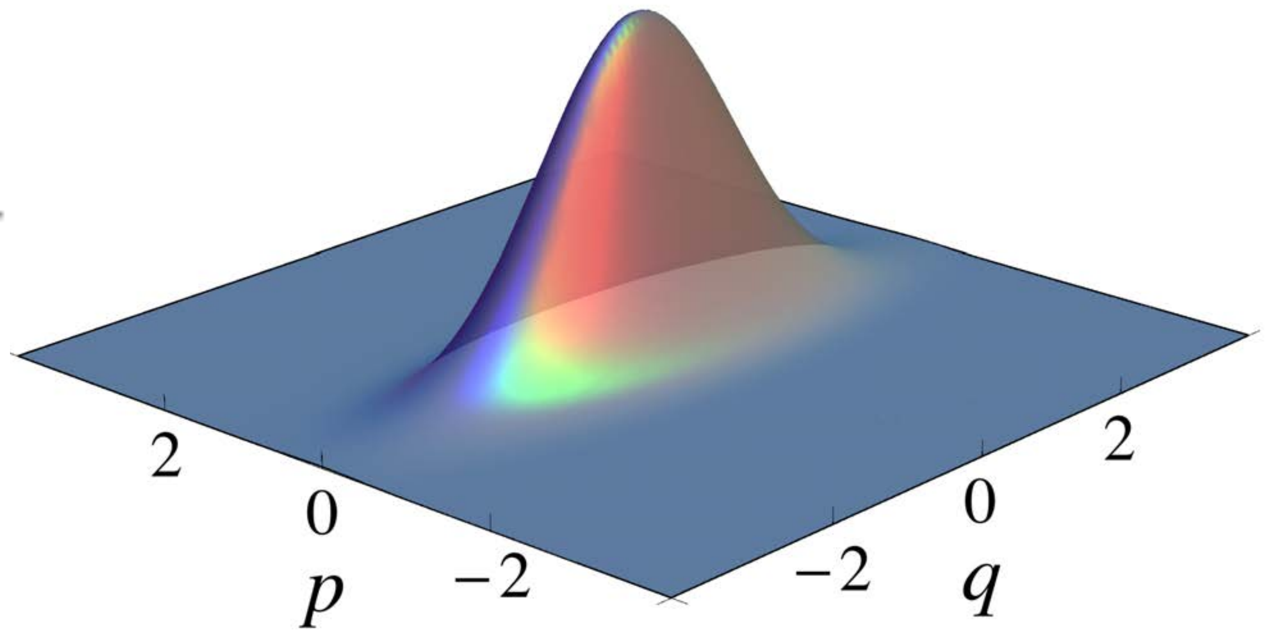
Example of operation:

$$X(s) = e^{-is\hat{p}}$$
$$Z(s) = e^{is\hat{q}}$$

Feel free to ask me!

What do I do? QC with the quantized harmonic oscillator





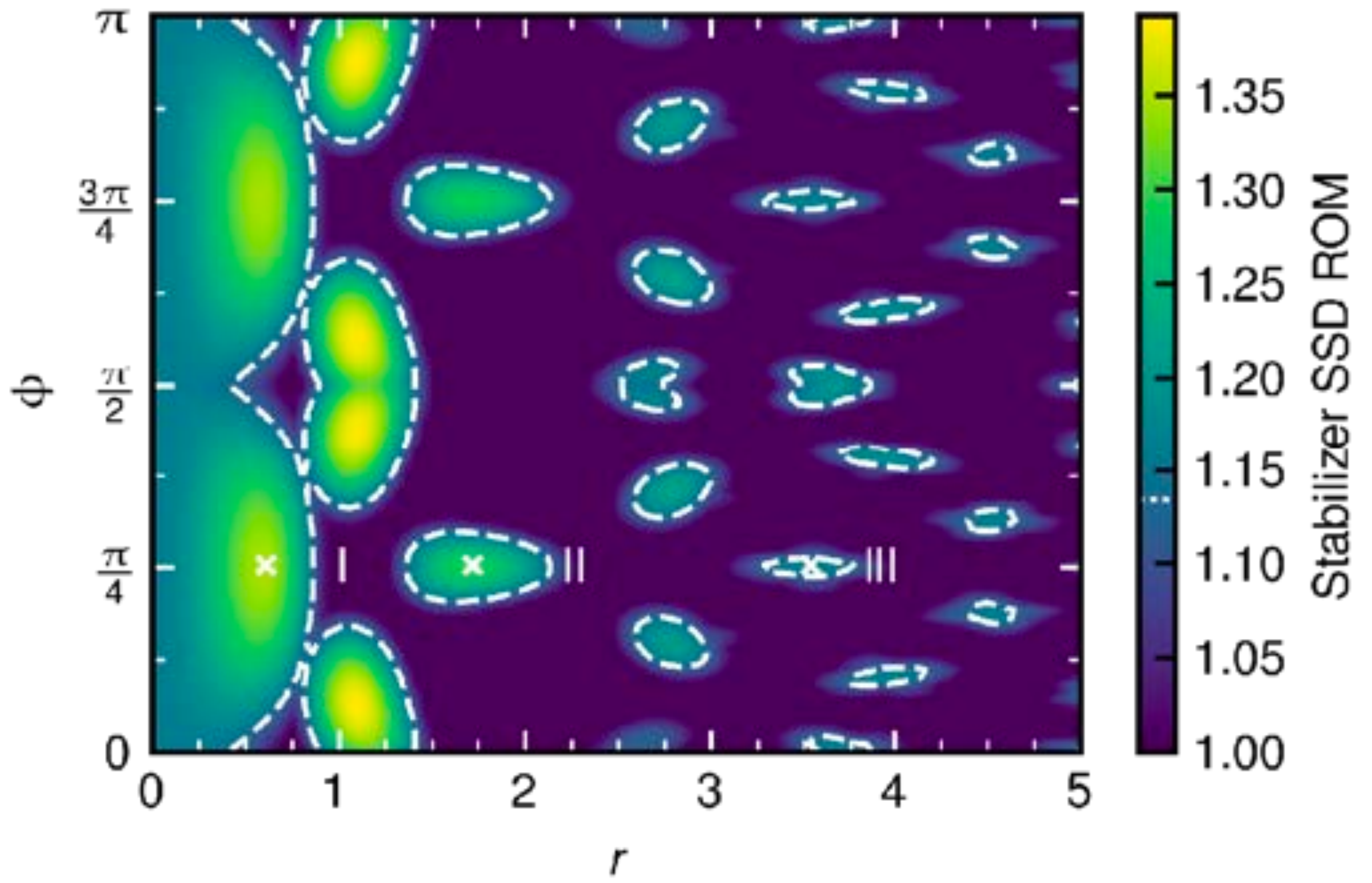
Phase space = Complex Plane

Infinite-dimensional Hilbert space

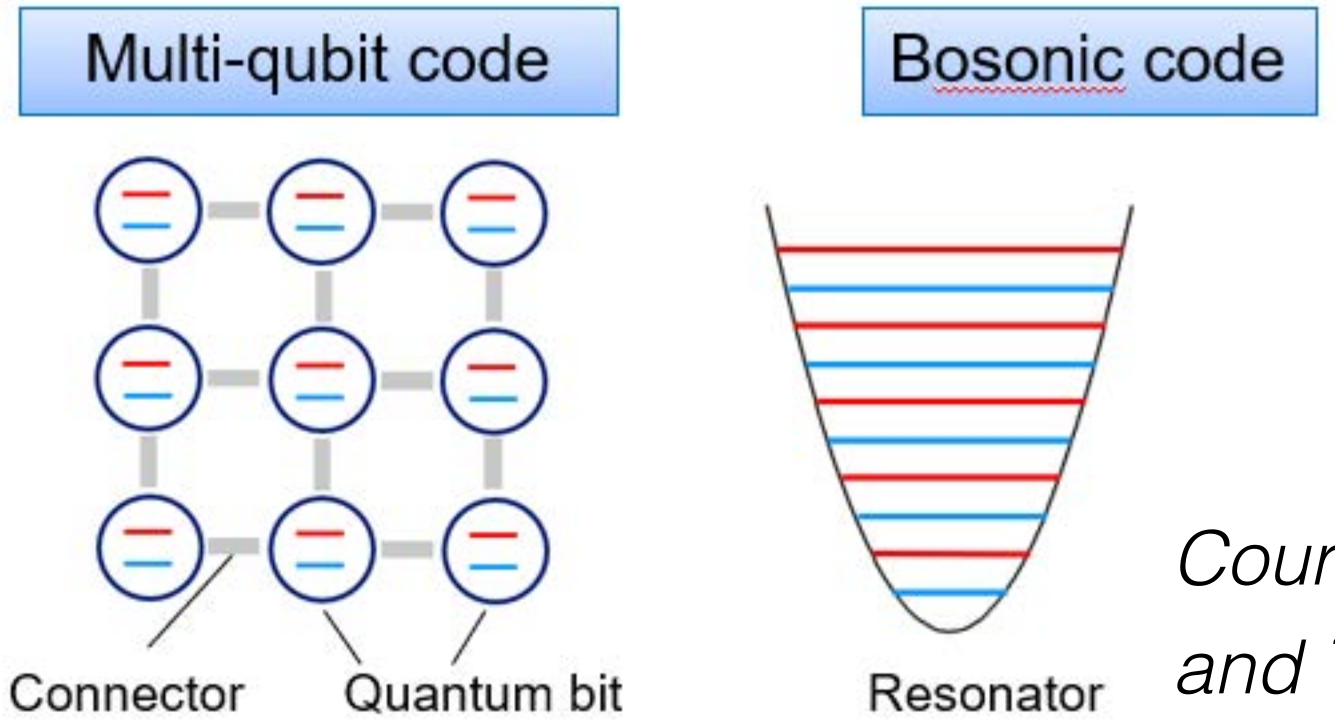
Example of operation:

$$X(s) = e^{-is\hat{p}}$$
$$Z(s) = e^{is\hat{q}}$$

Quantum resource theory for quantum computation: how resourceful bosonic states are?



Bosonic codes for QEC

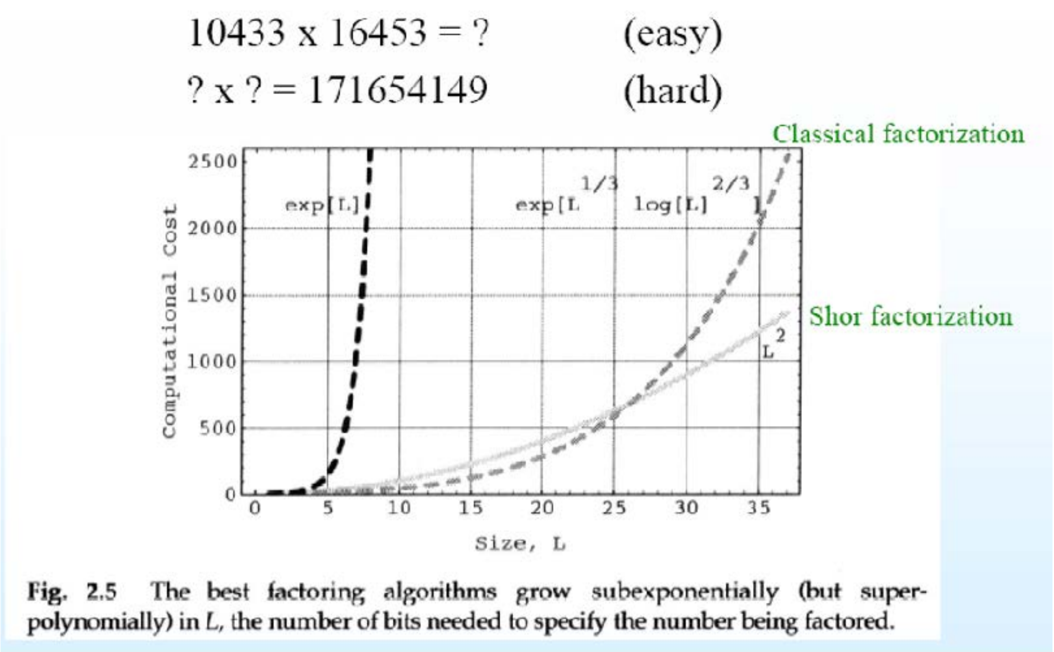


Courtesy of Japan Science and Technology Agency

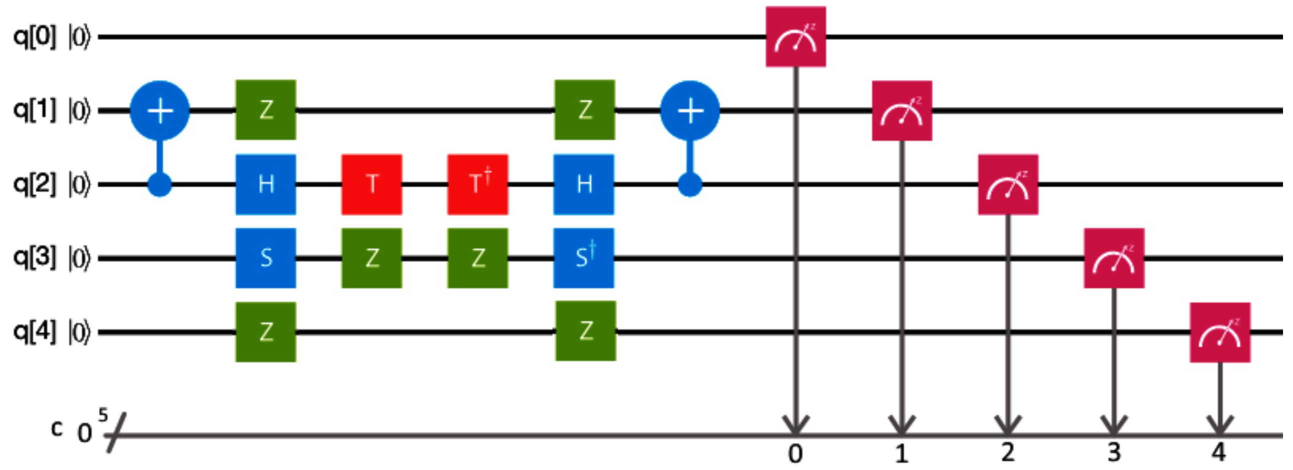
Feel free to ask me!

What have we learnt today?

- Why: A quantum computer would allow for solving problems that are intractable today



- How (software): Quantum algorithms are sequences of quantum gates



- Wanted: useful problems solvable on available quantum processors?

Thank you for your attention!

- Questions?