Quantum computing

Davide Rossini

UNIVERSITÀ DI PISA

INFN
Istituto Nazionale
Fisica Nucleare
Sezione di Pisa

QC workshop, Pisa – 17 January 2020

“Information is physical”

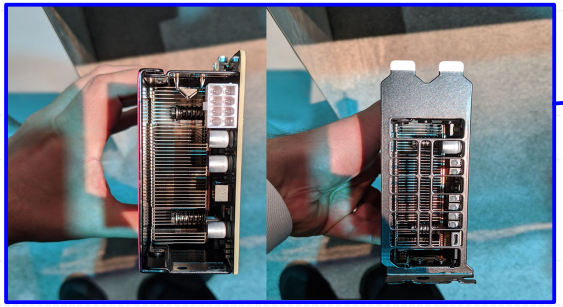
Rolf Landauer

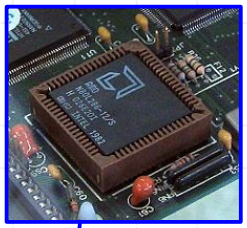
IBM Thomas J. Watson Research Center, NY, USA (1991)

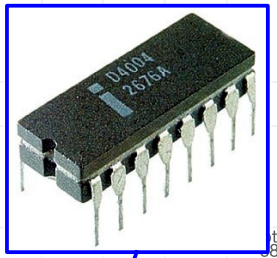
Moore's Law – The number of transistors on integrated circuit chips (1971-2018)

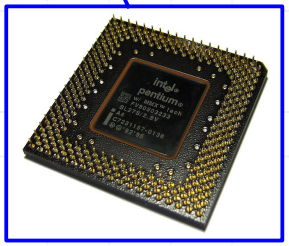
Moore's law describes the empirical regularity that the number of transistors on integrated circuits doubles approximately every two years. This advancement is important as other aspects of technological progress – such as processing speed or the price of electronic products – are linked to Moore's law.

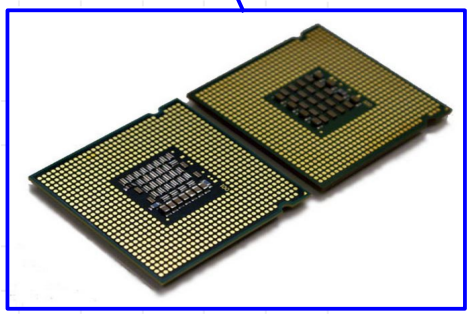
Our World in Data











Transistor count

50,000,000,000
10,000,000,000
5,000,000,000
1,000,000,000
500,000,000
100,000,000
50,000,000
10,000,000
5,000,000
1,000,000
500,000
100,000
50,000
10,000
5,000
1,000

1970 1972 1974 1976 1978 1980 1982 1984 1986 1988 1990 1992 1994 1996 1998 2000 2002 2004 2006 2008 2010 2012 2014 2016 2018

Intel 4004
Intel 8008
TMS 1000
Intel 8080
RCA 1802
Motorola 6800
Zilog Z80
Intel 8085
MOS Technology 6502
Motorola 6809
Intel 8086
Intel 8088
WDC 65C02
WDC 65C816
Novix NC4016
ARM 1
ARM 2
ARM 6
ARM 9TDMI
Intel 80186
Intel 80286
Intel 80386
Motorola 68020
Intel i960
ARM 3
DEC WRL MultiTitan
TI Explorer's 32-bit Lisp machine chip
Intel 80486
R4000
ARM700
Pentium
Pentium Pro
Pentium II Klamath
Pentium II Deschutes
Pentium III Katmai
AMD K5
AMD K6
SA-110
AMD K6-III
AMD K7
Pentium III Coppermine
Pentium III Tualatin
Pentium II Mobile Dixon
Pentium 4 Willamette
Pentium 4 Northwood
Barton
Atom
ARM Cortex-A9
AMD K8
Pentium 4 Prescott
Pentium 4 Cedar Mill
Pentium 4 Prescott-2M
Core 2 Duo Allendale
Itanium 2 McKinley
Cell
Core 2 Duo Wolfdale 3M
Pentium D Smithfield
Itanium 2 Madison 6M
Core 2 Duo Wolfdale
Core 2 Duo Conroe
Itanium 2 with 9 MB cache
AMD K10 quad-core 2M L3
Core i7 (Quad)
POWER6
Pentium D Presler
Dual-core Itanium 2
Six-core Xeon 7400
8-core Xeon Nehalem-EX
12-core POWER8
61-core Xeon Phi
Xbox One main SoC
18-core Xeon Haswell-E5
IBM z13 Storage Controller
72-core Xeon Phi Centriq 2400
SPARC M7
GC2 IPU
32-core AMD Epyc
Apple A12X Bionic
Tegra Xavier SoC
Qualcomm Snapdragon 8cx/SCX8180
HiSilicon Kirin 980 + Apple A12 Bionic
HiSilicon Kirin 710
10-core Core i7 Broadwell-E
Qualcomm Snapdragon 835
Dual-core + GPU Iris Core i7 Broadwell-U
Quad-core + GPU GT2 Core i7 Skylake K
Quad-core + GPU Core i7 Haswell
Apple A7 (dual-core ARM64 "mobile SoC")

Data source: Wikipedia (https://en.wikipedia.org/wiki/Transistor_count)
The data visualization is available at OurWorldinData.org. There you find more visualizations and research on this topic.

Classical vs. Quantum computers

Classical computer:
A computer that uses voltages flowing through circuits and gates which can be manipulated entirely through ***classical mechanics***

→ Moore's law: no. of on-chip transistors doubles every 18 months

→ emergence of quantum phenomena such as electron tunneling through barriers between wires, due to downscaling of circuit boards

→ ***serial processing***: one operation at a time

Classical vs. Quantum computers

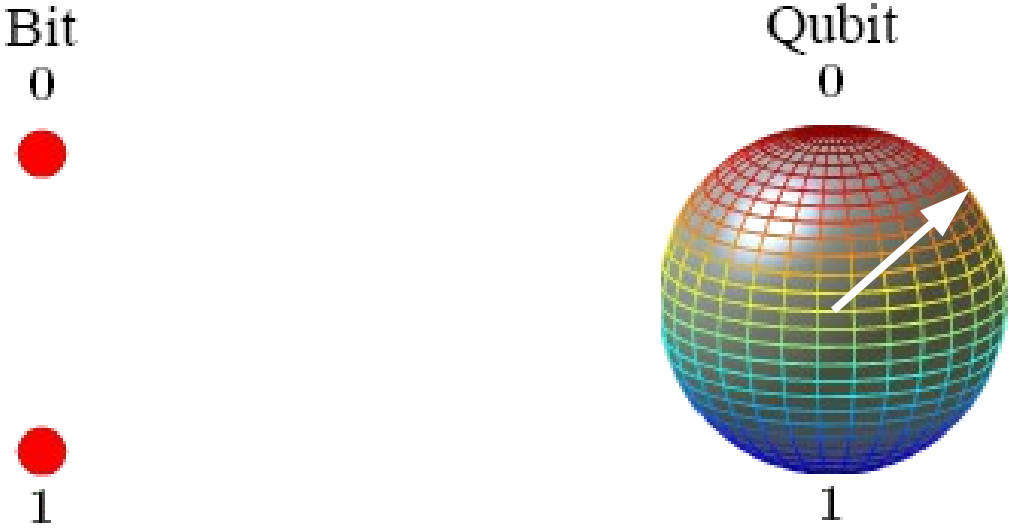
Classical computer:
A computer that uses voltages flowing through circuits and gates which can be manipulated entirely through ***classical mechanics***

→ Moore's law: no. of on-chip transistors doubles every 18 months
→ emergence of quantum phenomena such as electron tunneling through barriers between wires, due to downscaling of circuit boards
→ ***serial processing***: one operation at a time

Quantum computer:
A computer that exploits ***quantum mechanical phenomena*** to perform operations on data through suitable devices

→ harnesses the power of atoms/molecules to perform memory & processing tasks
→ ***parallel processing*** due to quantum superposition and entanglement

Classical vs. Quantum computers



M. A. Nielsen and I. L. Chuang, *Quantum computation and quantum information* (II ed., Cambridge 2010)

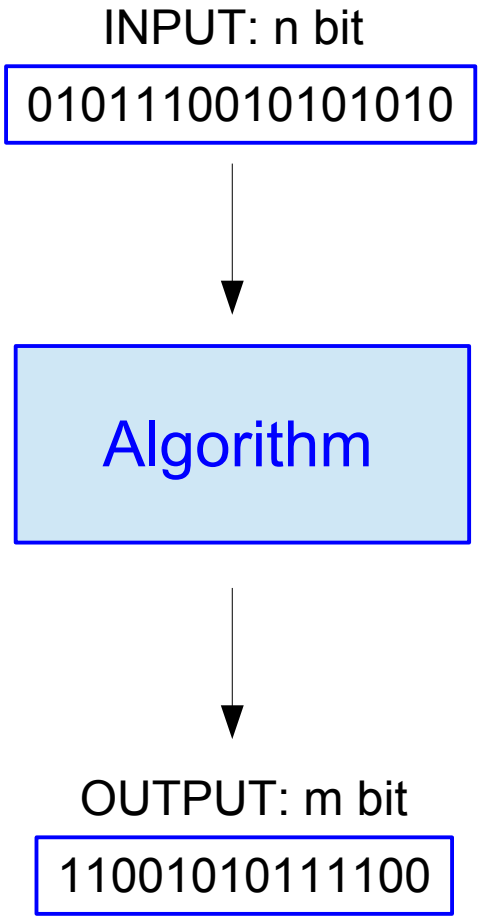
J. Preskill, *Lecture notes for Physics 229: Quantum information and computation* (CalTech 1998) available at: `http://www.theory.caltech.edu/people/preskill/ph219/`

G. Benenti, G. Casati, D. Rossini, G. Strini, *Principles of quantum computation and information: a comprehensive textbook* (World Scientific 2019)

Classical logic: the bit

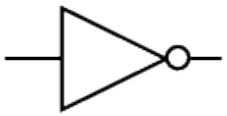
bit: 2 basic states

(0 ≡ off 1 ≡ on)
mutually exclusive



Classical logic

bit: 2 basic states (0 ≡ off 1 ≡ on)
mutually exclusive

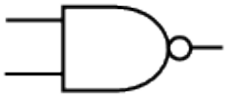


NOT

Input	Output
I	F
0	1
1	0

AND

Inputs		Output
A	B	F
0	0	0
1	0	0
0	1	0
1	1	1



NAND

Inputs		Output
A	B	F
0	0	1
1	0	1
0	1	1
1	1	0

OR

Inputs		Output
A	B	F
0	0	0
1	0	1
0	1	1
1	1	1

NOR

Inputs		Output
A	B	F
0	0	1
1	0	0
0	1	0
1	1	0

NAND
+
COPY

Universal set
of (***irreversible***)
classical gates

EXCLUSIVE OR

Inputs		Output
A	B	F
0	0	0
0	1	1
1	0	1
1	1	0

EXCLUSIVE NOR

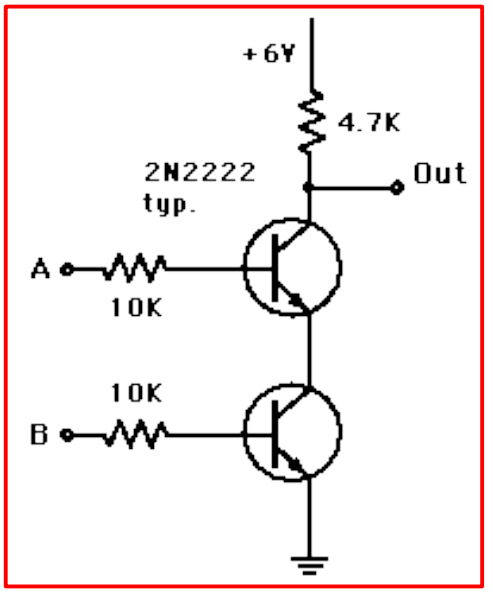
Inputs		Output
A	B	F
0	0	1
0	1	0
1	0	0
1	1	1

Classical logic

In computers the NAND gate is usually implemented via transistors.
A bit is set to 1 if the voltage is positive and to 0 if the voltage is zero.

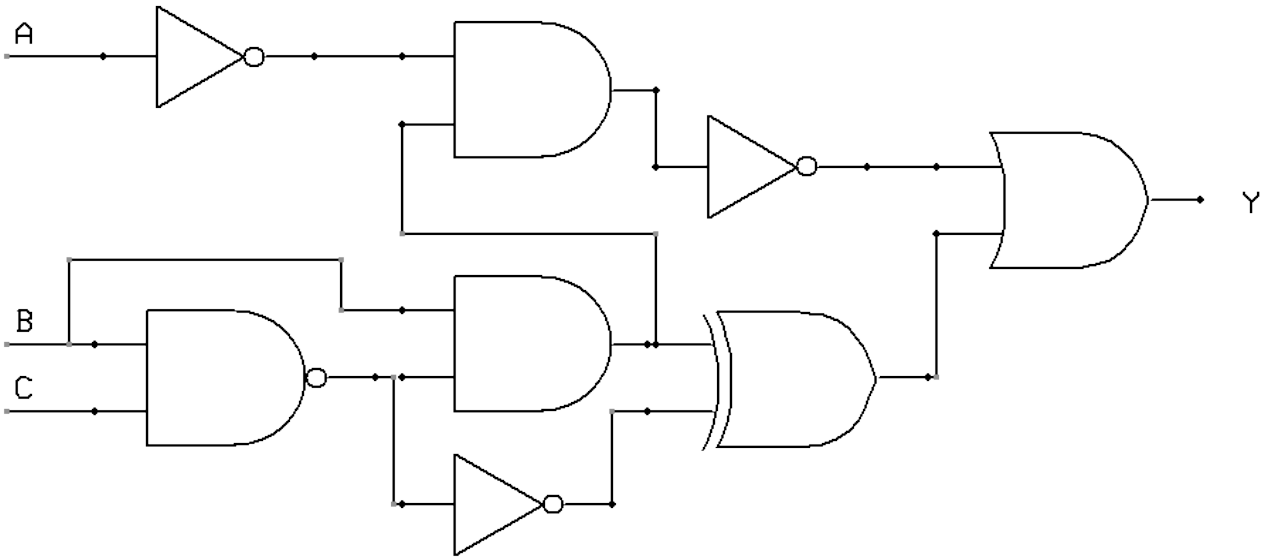
Here the current flows through the transistors if and only if both inputs have positive voltage. In this case, the output has zero voltage.

If at least one of the inputs has zero voltage, there is no current flow and therefore the output has positive voltage.



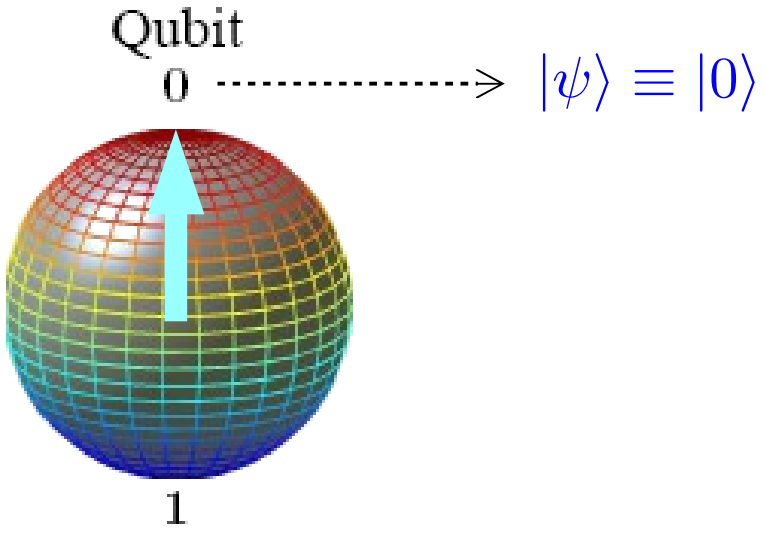
The NAND gate

Network of classical gates



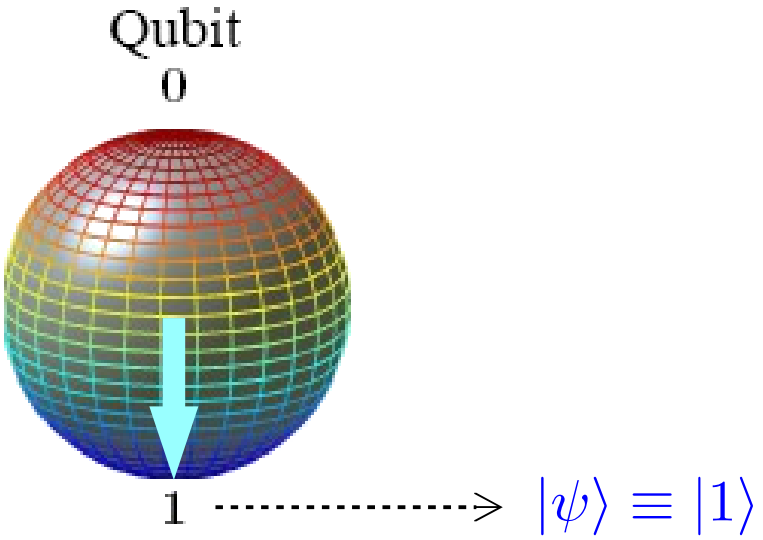
Quantum logic

Qubit (quantum bit): 2 basic states



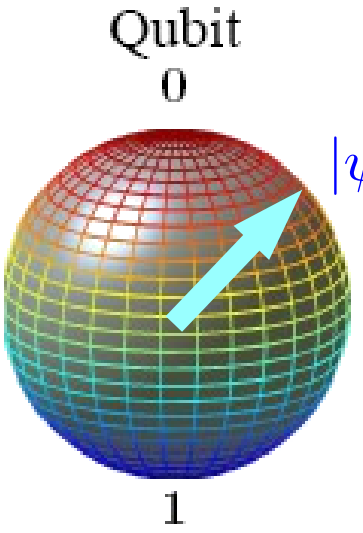
Quantum logic

Qubit (quantum bit): 2 basic states

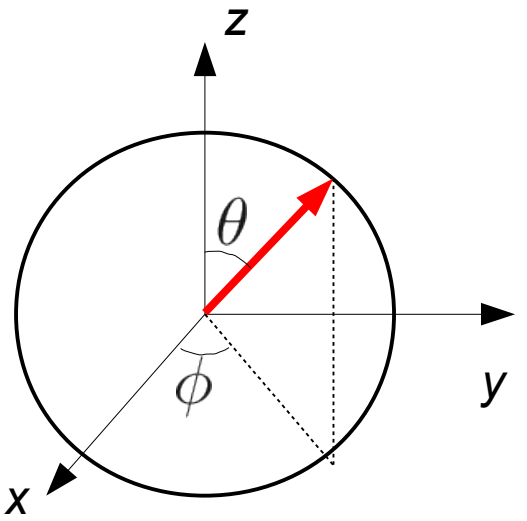


Quantum logic

Qubit (quantum bit): 2 basic states **+** ***superpositions***



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



Bloch sphere

In general the state of a qubit can be a ***linear superposition*** of |0> and |1>

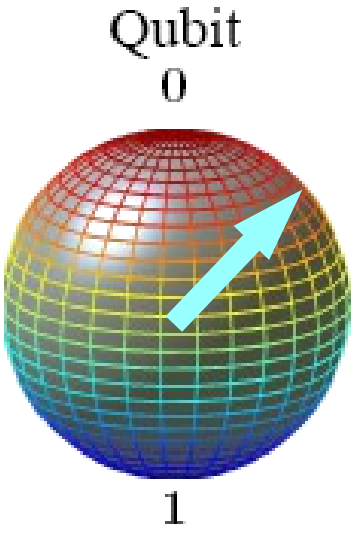
$$|\psi\rangle = \cos\frac{\theta}{2}|0\rangle + e^{i\phi}\sin\frac{\theta}{2}|1\rangle$$

$$0 \le \theta \le \pi$$

$$0 \le \phi < 2\pi$$

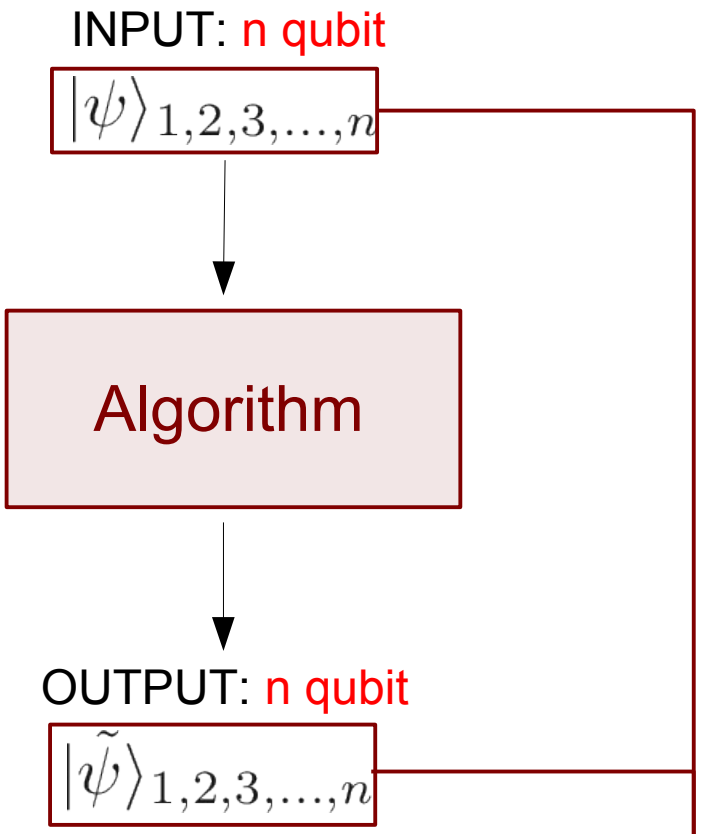
Quantum logic

Qubit (quantum bit): 2 basic states **+ *superpositions***



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

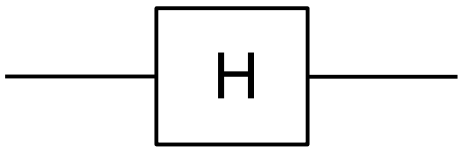
α and β arbitrary complex numbers,
such that $|\alpha|^2 + |\beta|^2 = 1$



$$\alpha_1|0\ldots00\rangle + \alpha_2|0\ldots01\rangle + \ldots + \alpha_{2^n}|1\ldots11\rangle$$

Quantum gates

Set of **universal quantum gates** (***reversible***):



Single-qubit gates

e.g. Hadamard

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

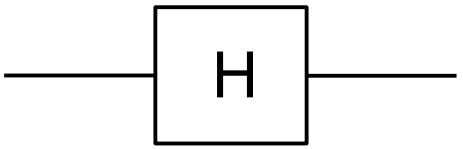
$$|0\rangle \rightarrow \frac{1}{\sqrt{2}}\left(|0\rangle + |1\rangle\right)$$

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

transforms $|0\rangle$ and $|1\rangle$ in superposition states

Quantum gates

Set of **universal quantum gates** (***reversible***):



Single-qubit gates

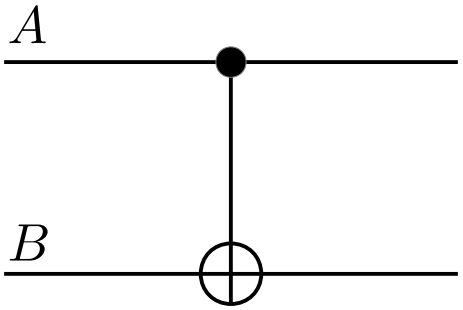
e.g. Hadamard

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

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

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

transforms $|0\rangle$ and $|1\rangle$ in superposition states



Two-qubit gate:

control-NOT

$$U_{CN} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

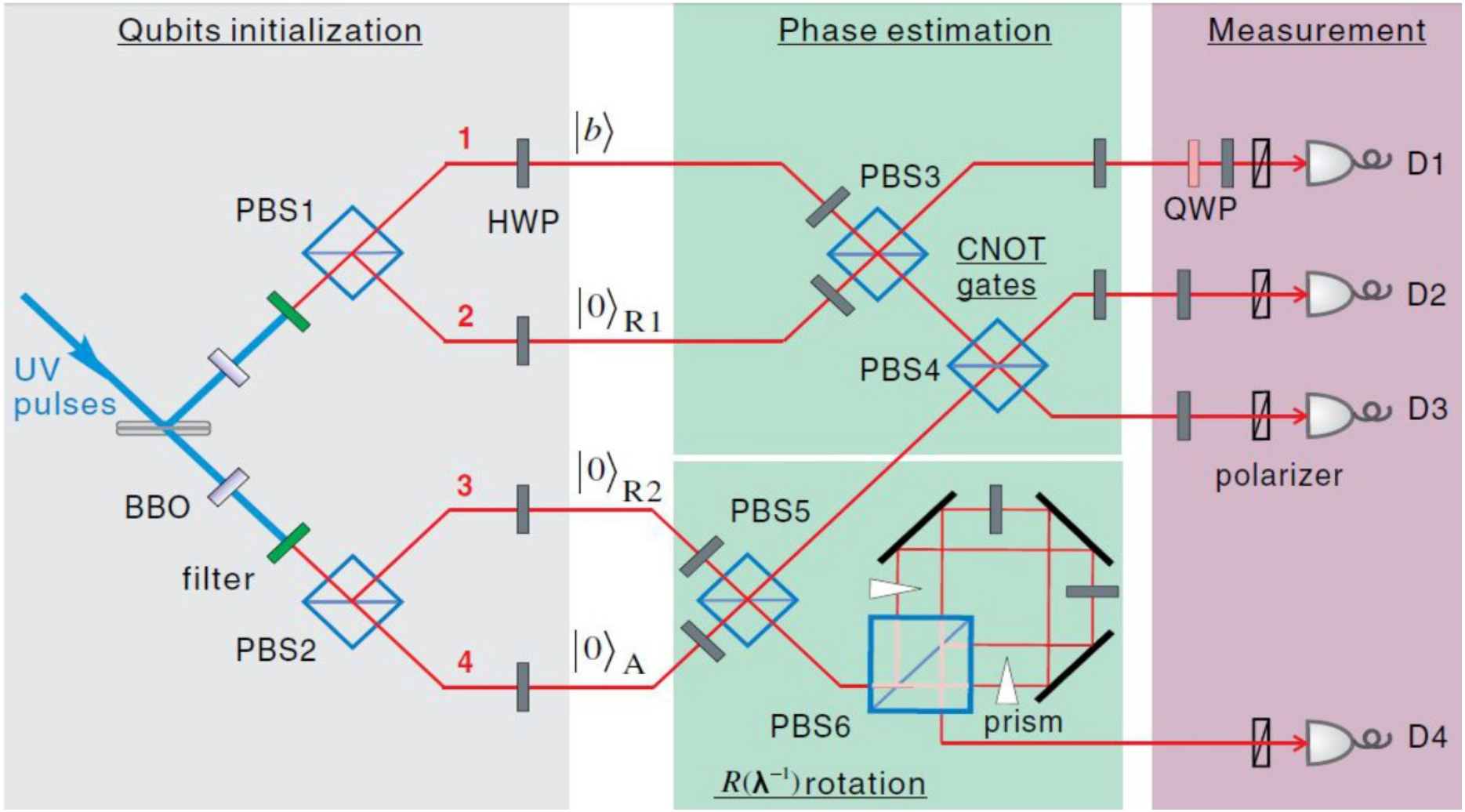
$$|0\rangle_A \otimes |0\rangle_B \rightarrow |0\rangle_A \otimes |0\rangle_B$$

$$|0\rangle_A \otimes |1\rangle_B \rightarrow |0\rangle_A \otimes |1\rangle_B$$

$$|1\rangle_A \otimes |0\rangle_B \rightarrow |1\rangle_A \otimes |1\rangle_B$$

$$|1\rangle_A \otimes |1\rangle_B \rightarrow |1\rangle_A \otimes |0\rangle_B$$

Quantum circuits



X.-D. Cai et al., *Experimental quantum computing to solve systems of linear equations*, *Phys. Rev. Lett.* **110**, 230501 (2013)

A simple example

Given a one-bit function *f*, **determine whether it is *costant* or *balanced***

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

if $f(0) = f(1)$ → constant

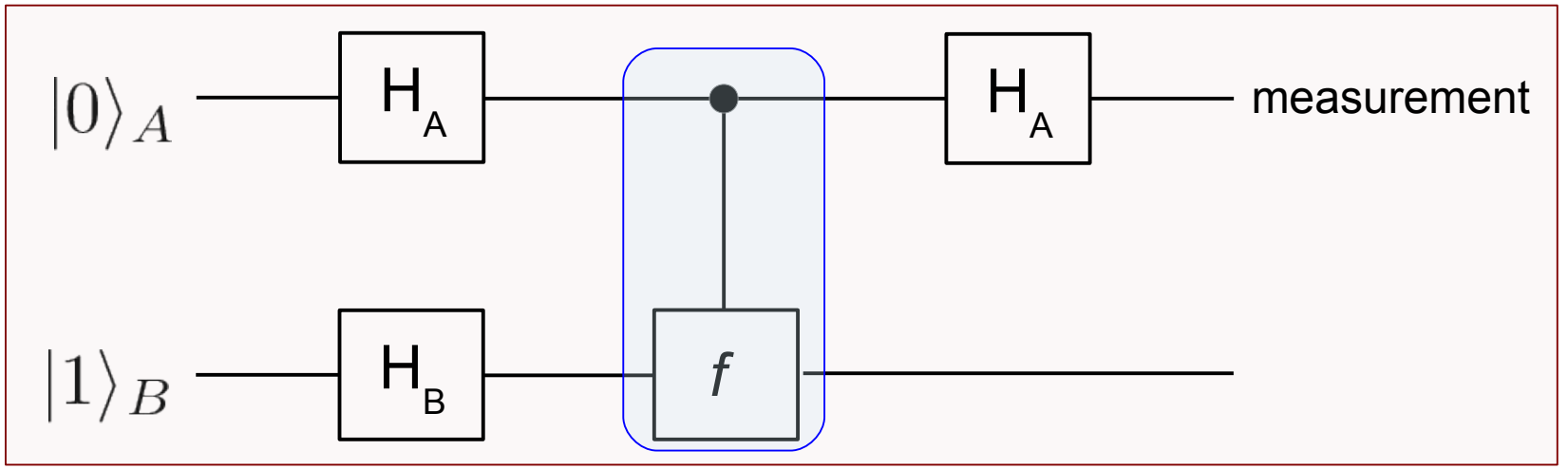
if $f(0) \neq f(1)$ → balanced

➔ With **classical logic** one needs to evaluate *f* **twice**

➔ With **quantum logic** it is sufficient to evaluate *f* **once**

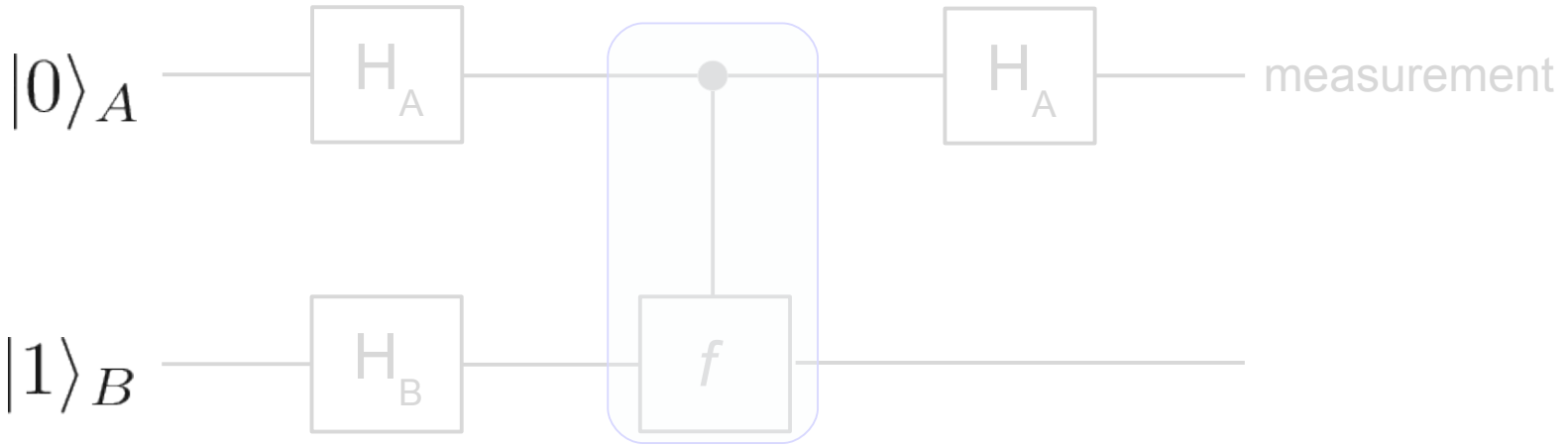
D. Deutsch, *Proc. Roy. Soc. London A* **400**, 97 (1985)
D. Deutsch and R. Jozsa, *Proc. Roy. Soc. London A* **439**, 553 (1992)

Deutsch's algorithm



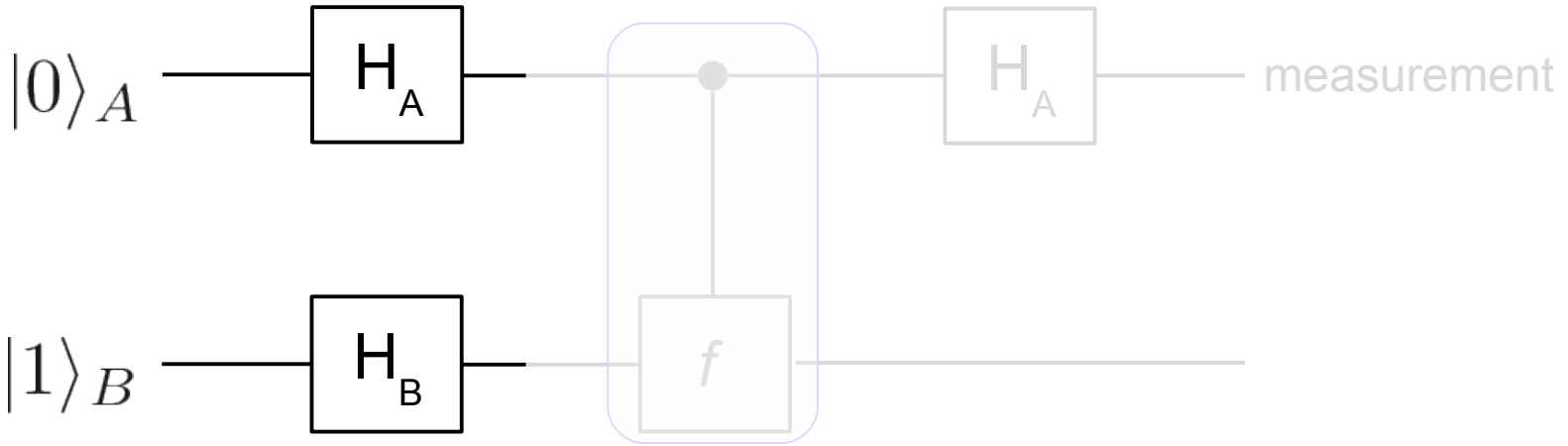
$$|x\rangle_A|y\rangle_B \rightarrow |x\rangle_A|y \oplus f(x)\rangle_B$$

Deutsch's algorithm



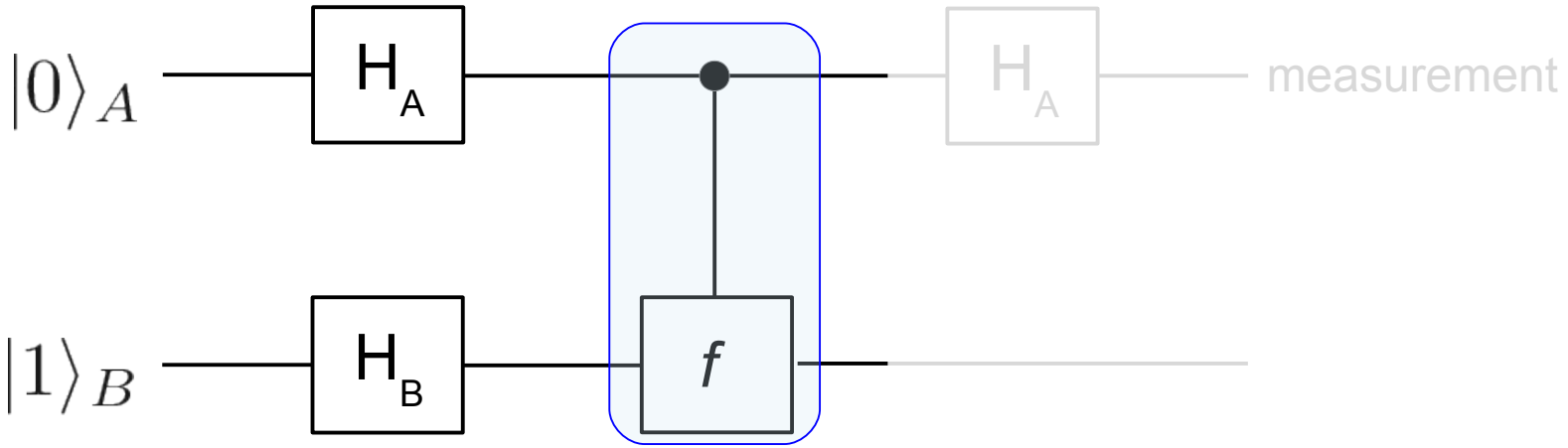
$|0\rangle_A|1\rangle_B$

Deutsch's algorithm



$$|0\rangle_A|1\rangle_B \xrightarrow{H_A \times H_B} \frac{|0\rangle_A + |1\rangle_A}{\sqrt{2}} \cdot \frac{|0\rangle_B - |1\rangle_B}{\sqrt{2}}$$

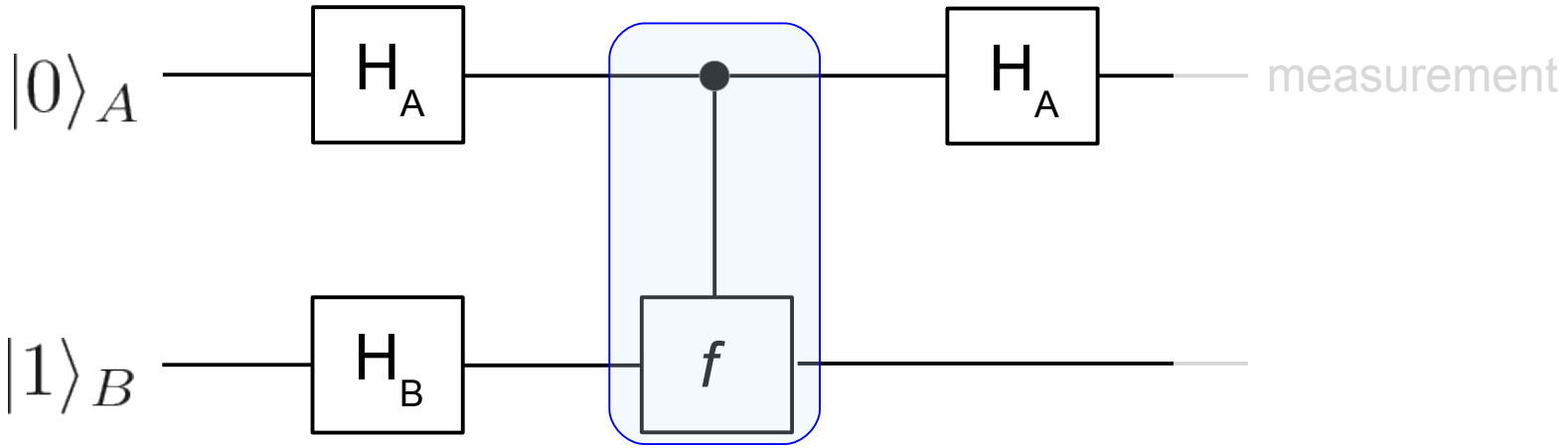
Deutsch's algorithm



$$|x\rangle_A|y\rangle_B \rightarrow |x\rangle_A|y \oplus f(x)\rangle_B$$

$$|0\rangle_A|1\rangle_B \xrightarrow{H_A \times H_B} \frac{|0\rangle_A + |1\rangle_A}{\sqrt{2}} \cdot \frac{|0\rangle_B - |1\rangle_B}{\sqrt{2}} \xrightarrow{f} \left[\frac{(-1)^{f(0)}|0\rangle_A + (-1)^{f(1)}|1\rangle_A}{\sqrt{2}}\right] \frac{|0\rangle_B - |1\rangle_B}{\sqrt{2}}$$

Deutsch's algorithm

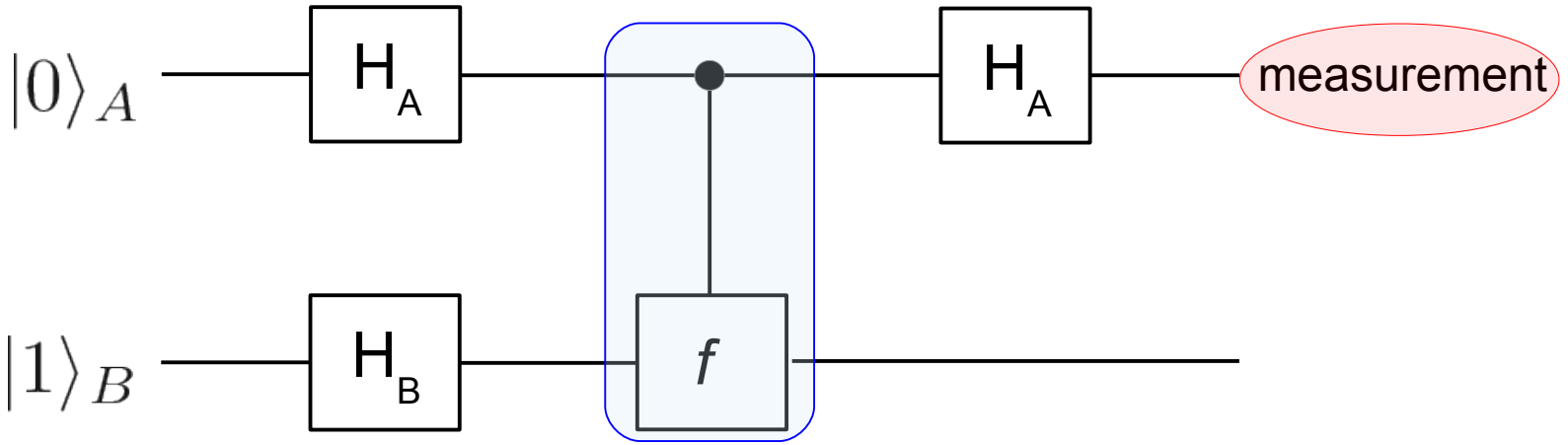


$$|x\rangle_A|y\rangle_B \rightarrow |x\rangle_A|y \oplus f(x)\rangle_B$$

$$|0\rangle_A|1\rangle_B \xrightarrow{H_A \times H_B} \frac{|0\rangle_A + |1\rangle_A}{\sqrt{2}} \cdot \frac{|0\rangle_B - |1\rangle_B}{\sqrt{2}} \xrightarrow{f} \left[\frac{(-1)^{f(0)}|0\rangle_A + (-1)^{f(1)}|1\rangle_A}{\sqrt{2}}\right] \frac{|0\rangle_B - |1\rangle_B}{\sqrt{2}}$$

$$\xrightarrow{H_A} \frac{1}{2}\left[(-1)^{f(0)} + (-1)^{f(1)}\right]|0\rangle_A + \left[(-1)^{f(0)} - (-1)^{f(1)}\right]|1\rangle_A$$

Deutsch's algorithm



$$|x\rangle_A|y\rangle_B \rightarrow |x\rangle_A|y \oplus f(x)\rangle_B$$

$$|0\rangle_A|1\rangle_B \xrightarrow{H_A \times H_B} \frac{|0\rangle_A + |1\rangle_A}{\sqrt{2}} \cdot \frac{|0\rangle_B - |1\rangle_B}{\sqrt{2}} \xrightarrow{f} \left[\frac{(-1)^{f(0)}|0\rangle_A + (-1)^{f(1)}|1\rangle_A}{\sqrt{2}}\right] \frac{|0\rangle_B - |1\rangle_B}{\sqrt{2}}$$

$$\xrightarrow{H_A} \frac{1}{2}\left[(-1)^{f(0)} + (-1)^{f(1)}\right]|0\rangle_A + \left[(-1)^{f(0)} - (-1)^{f(1)}\right]|1\rangle_A$$

if one measures $|0\rangle_A$ then *f* is **constant**

if one measues $|1\rangle_A$ then *f* is **balanced**

Quantum algorithms

Exploiting ***quantum parallelism***, it is possible to devise algorithms which are *exponentially (or substantially) faster* than any known classical algorithm for the same purpose

Quantum algorithms

Exploiting ***quantum parallelism***, it is possible to devise algorithms which are ***exponentially (or substantially) faster*** than any known classical algorithm for the same purpose

SIAM J. COMPUT.
Vol. 26, No. 5, pp. 1484–1509, October 1997

009

POLYNOMIAL-TIME ALGORITHMS FOR PRIME FACTORIZATION AND DISCRETE LOGARITHMS ON A QUANTUM COMPUTER*

PETER W. SHOR†

Abstract. A digital computer is generally believed to be an efficient universal computing device; that is, it is believed able to simulate any physical computing device with an increase in computation time by at most a polynomial factor. This may not be true when quantum mechanics is taken into consideration. This paper considers factoring integers and finding discrete logarithms, two problems which are generally thought to be hard on a classical computer and which have been used as the basis of several proposed cryptosystems. Efficient randomized algorithms are given for these two problems on a hypothetical quantum computer. These algorithms take a number of steps polynomial in the input size, e.g., the number of digits of the integer to be factored.

$$O(e^{\sqrt[3]{n}}) \rightarrow O(n^2 \log n)$$

Quantum algorithms

Exploiting ***quantum parallelism***, it is possible to devise algorithms which are ***exponentially (or substantially) faster*** than any known classical algorithm for the same purpose

VOLUME 79, NUMBER 2 PHYSICAL REVIEW LETTERS 14 JULY 1997

Quantum Mechanics Helps in Searching for a Needle in a Haystack

Lov K. Grover*

3C-404A Bell Labs, 600 Mountain Avenue, Murray Hill, New Jersey 07974

(Received 4 December 1996)

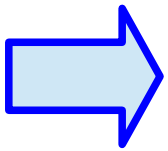
Quantum mechanics can speed up a range of search applications over unsorted data. For example, imagine a phone directory containing N names arranged in completely random order. To find someone's phone number with a probability of 50%, any classical algorithm (whether deterministic or probabilistic) will need to access the database a minimum of $0.5N$ times. Quantum mechanical systems can be in a superposition of states and simultaneously examine multiple names. By properly adjusting the phases of various operations, successful computations reinforce each other while others interfere randomly. As a result, the desired phone number can be obtained in only $O(\sqrt{N})$ accesses to the database.

[S0031-9007(97)03564-3]

$$O(e^n) \rightarrow O(e^{n/2})$$

Quantum algorithms

Exploiting ***quantum parallelism***, it is possible to devise algorithms which are *exponentially (or substantially) faster* than any known classical algorithm for the same purpose



Unfortunately such algorithms require
a **huge** amount of resources (# quantum gates)

e.g. factorizing a 4096-bit number would require $O(10^{12})$ gates...

Unfeasible with present-day technology :-(

Look for possible alternatives.....

Adiabatic Quantum Computation

A class of procedures for solving *optimization problems* with quantum computers.

E. Fahri, J. Goldstone, S. Gutmann, J. Lapan, A. Lundgren, D. Preda, *Science* ***292****, 472 (2001)*
G. E. Santoro, R. Martonak, E. Tosatti, R. Car, *Science* ***295****, 2427 (2002)*

Adiabatic Quantum Computation

A class of procedures for solving *optimization problems* with quantum computers.

Basic strategy:

- **design** a *problem Hamiltonian* H_P whose ground state encodes the solution of an optimization problem

E. Fahri, J. Goldstone, S. Gutmann, J. Lapan, A. Lundgren, D. Preda, *Science* ***292****, 472 (2001)*
G. E. Santoro, R. Martonak, E. Tosatti, R. Car, *Science* ***295****, 2427 (2002)*

Adiabatic Quantum Computation

A class of procedures for solving *optimization problems* with quantum computers.

Basic strategy:

- **design** a *problem Hamiltonian* H_P whose ground state encodes the solution of an optimization problem
- **prepare** the known ground state of a simple Hamiltonian H_0

E. Fahri, J. Goldstone, S. Gutmann, J. Lapan, A. Lundgren, D. Preda, *Science* ***292****, 472 (2001)*
G. E. Santoro, R. Martonak, E. Tosatti, R. Car, *Science* ***295****, 2427 (2002)*

Adiabatic Quantum Computation

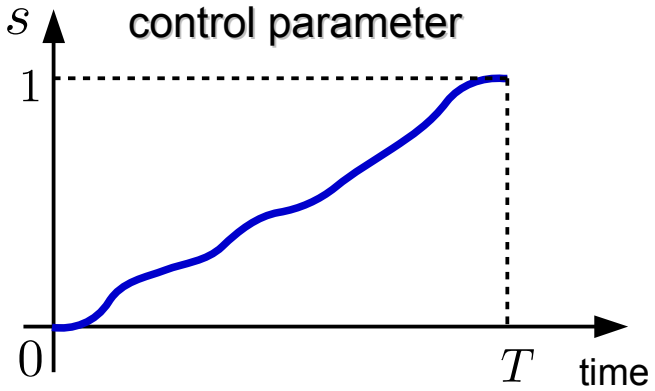
A class of procedures for solving *optimization problems* with quantum computers.

Basic strategy:

- **design** a *problem Hamiltonian* H_P whose ground state encodes the solution of an optimization problem
- **prepare** the known ground state of a simple Hamiltonian H_0
- **interpolate** slowly

$$H(s) = \big[1 - s\big] H_0 + s\, H_P$$

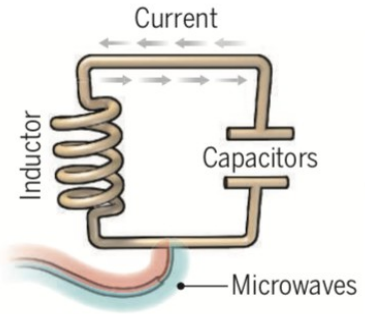
(*adiabatic theorem*)

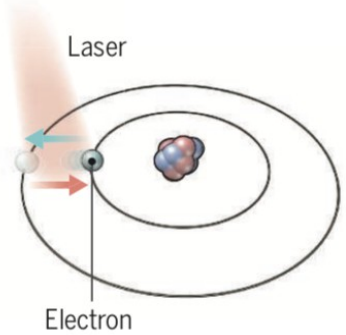


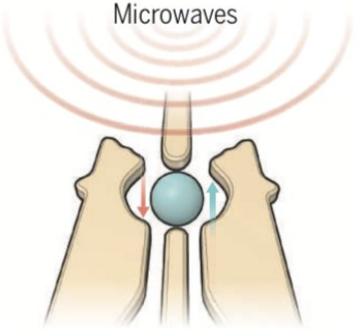
E. Fahri, J. Goldstone, S. Gutmann, J. Lapan, A. Lundgren, D. Preda, *Science* ***292****, 472 (2001)*
G. E. Santoro, R. Martonak, E. Tosatti, R. Car, *Science* ***295****, 2427 (2002)*

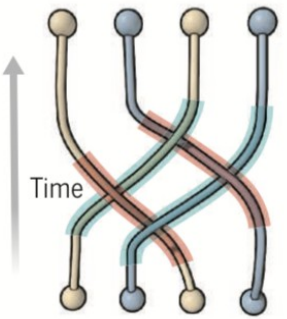
Where are we now?

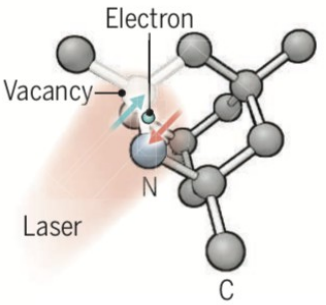
Physical systems for qubits







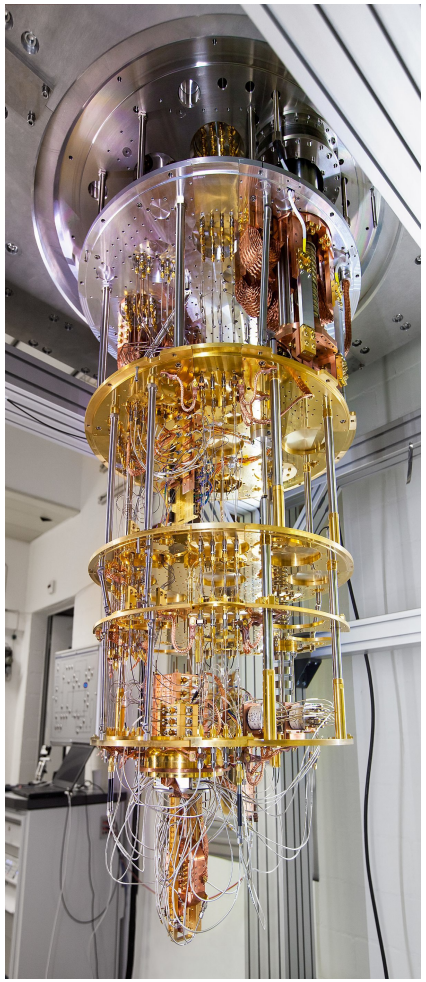




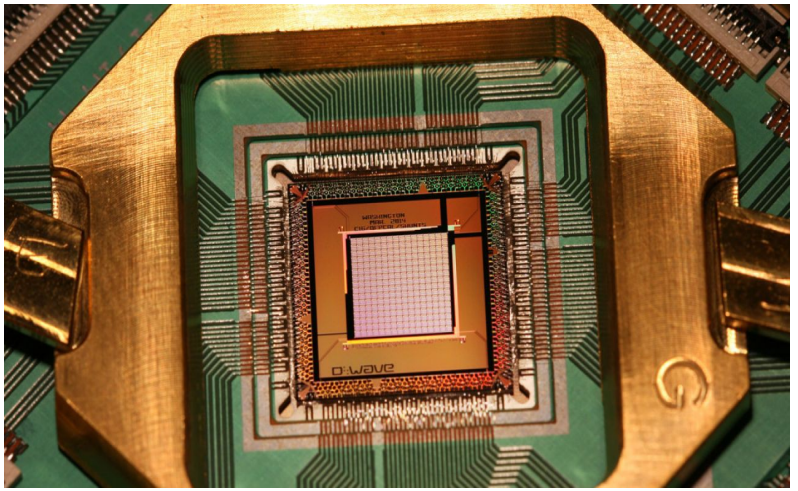
Superconducting loops	**Trapped ions**	**Silicon quantum dots**	**Topological qubits**	**Diamond vacancies**
A resistance-free current oscillates back and forth around a circuit loop. An injected microwave signal excites the current into super-position states.	Electrically charged atoms, or ions, have quantum energies that depend on the location of electrons. Tuned lasers cool and trap the ions, and put them in superposition states.	These "artificial atoms" are made by adding an electron to a small piece of pure silicon. Microwaves control the electron's quantum state.	Quasiparticles can be seen in the behavior of electrons channeled through semi-conductor structures.Their braided paths can encode quantum information.	A nitrogen atom and a vacancy add an electron to a diamond lattice. Its quantum spin state, along with those of nearby carbon nuclei, can be controlled with light.
Longevity (seconds) 0.00005	>1000	0.03	N/A	10
Logic success rate 99.4%	99.9%	~99%	N/A	99.2%
Number entangled 9	14	2	N/A	6
Company support Google, IBM, Quantum Circuits	ionQ	Intel	Microsoft, Bell Labs	Quantum Diamond Technologies
Pros Fast working. Build on existing semiconductor industry.	Very stable. Highest achieved gate fidelities.	Stable. Build on existing semiconductor industry.	Greatly reduce errors.	Can operate at room temperature.
Cons Collapse easily and must be kept cold.	Slow operation. Many lasers are needed.	Only a few entangled. Must be kept cold.	Existence not yet confirmed.	Difficult to entangle.

G. Popkin, Science **354**, 1091 (2016)

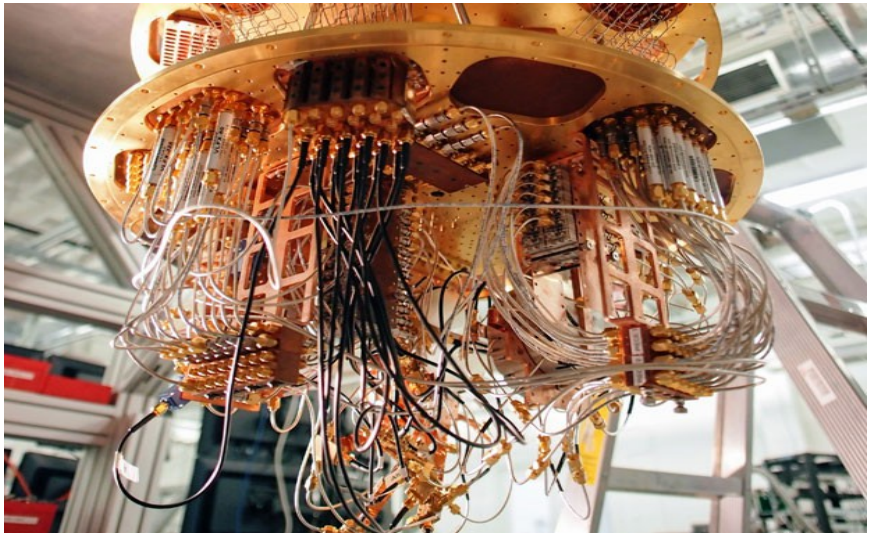
Quantum computers



zurich.ibm.com



dwavesys.com



research.google

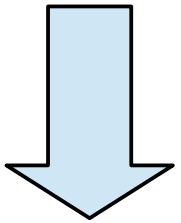
Quantum computers: status

Current technology

≈ 50-qubit operating machines needed to rival current classical equivalents

Errors

- Imperfections in realizing quantum gates
- Decoherence: tendency to decay from a given quantum state into an incoherent state, due to interactions with environment



≈ $O(10^2)$ gates
until decoherence sets in

breakdown of information
stored in the quantum computer

Error rates typically proportional to the ratio of operating time to decoherence time:
operations must be completed *much quicker than the decoherence time*.

Quantum supremacy tests

REVIEW

doi:10.1038/nature23458

Quantum computational supremacy

Aram W. Harrow[1] & Ashley Montanaro[2]

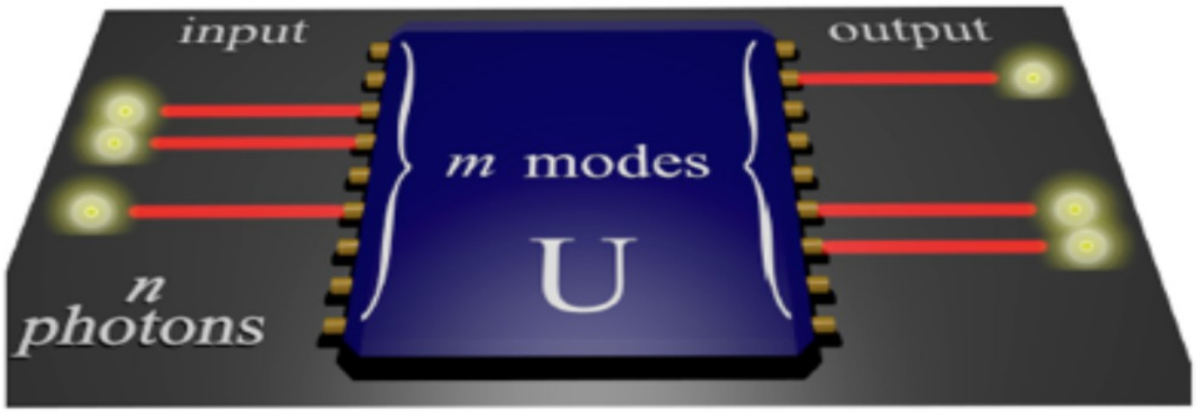
The field of quantum algorithms aims to find ways to speed up the solution of computational problems by using a quantum computer. A key milestone in this field will be when a universal quantum computer performs a computational task that is beyond the capability of any classical computer, an event known as quantum supremacy. This would be easier to achieve experimentally than full-scale quantum computing, but involves new theoretical challenges. Here we present the leading proposals to achieve quantum supremacy, and discuss how we can reliably compare the power of a classical computer to the power of a quantum computer.

[1]Center for Theoretical Physics, Massachusetts Institute of Technology, Cambridge, Massachusetts 02139, USA. [2]School of Mathematics, University of Bristol, Bristol BS8 1TW, UK.

Quantum supremacy tests

Boson sampling

Sampling the output distribution of noninteracting bosons evolving through an arbitrary (random) linear network



Transition amplitudes are related to the permanent of a square matrix (classically hard to compute)

$$\mathrm{per}(U) = \sum_{\sigma \in S_n} \prod_{i=1}^{n} u_{i,\sigma(i)}$$

"Small-scale quantum computers made from an array of interconnected waveguides on a glass chip can now perform a task that is considered hard to undertake on a large scale by classical means."

T. Ralph, *News & Views, Nature Photonics* **7**, 514 (2013)

Quantum supremacy tests

Random circuit sampling

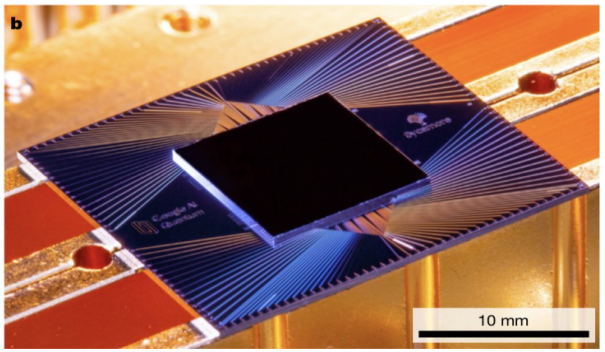
More in general, use quantum computers for their natural tasks: execute an arbitrary (random) circuit.

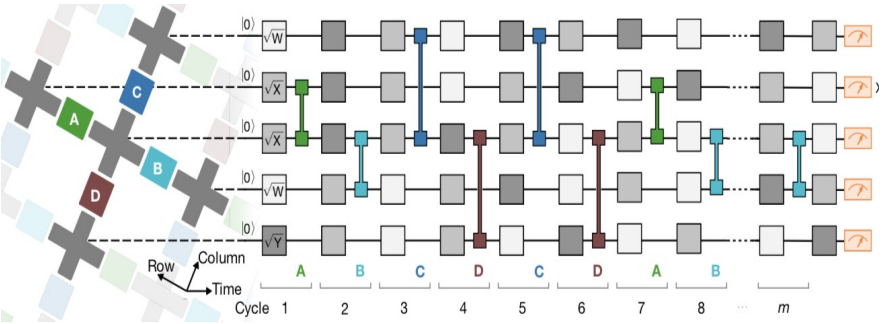
Article | Published: 23 October 2019

Quantum supremacy using a programmable superconducting processor

Frank Arute, Kunal Arya, [...] John M. Martinis

Nature **574**, 505–510(2019) | Cite this article



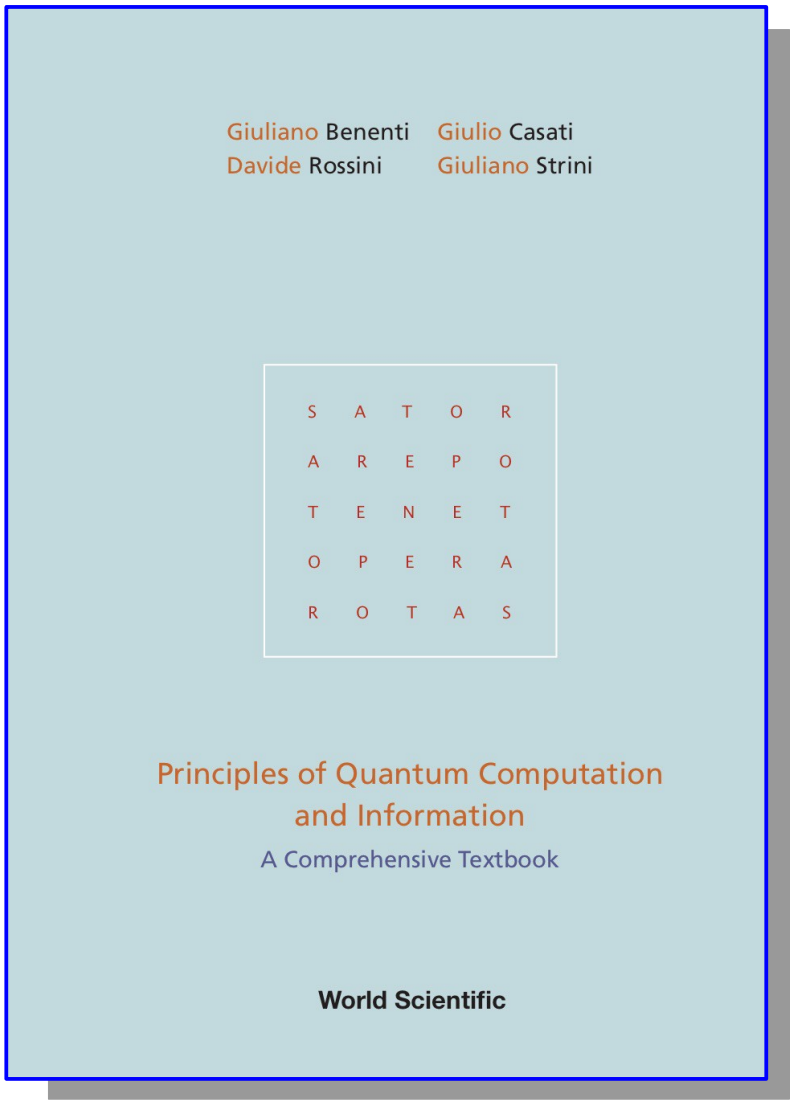


Abstract

The promise of quantum computers is that certain computational tasks might be executed exponentially faster on a quantum processor than on a classical processor[1]. A fundamental challenge is to build a high-fidelity processor capable of running quantum algorithms in an exponentially large computational space. Here we report the use of a processor with programmable superconducting qubits[2–7] to create quantum states on 53 qubits, corresponding to a computational state-space of dimension 2^{53} (about 10^{16}). Measurements from repeated experiments sample the resulting probability distribution, which we verify using classical simulations. Our Sycamore processor takes about 200 seconds to sample one instance of a quantum circuit a million times—our benchmarks currently indicate that the equivalent task for a state-of-the-art classical supercomputer would take approximately 10,000 years. This dramatic increase in speed compared to all known classical algorithms is an experimental realization of quantum supremacy[8–14] for this specific computational task, heralding a much-anticipated computing paradigm.

Principles of Quantum Computation and Information:
A Comprehensive Textbook

G. Benenti, G. Casati, D. Rossini, G. Strini



Contents

World Scientific, Singapore, 2019

Adiabatic Quantum Computation

$$H(s) = \big[1 - s\big] H_0 + s\, H_P$$

$$s(t) \in [0, 1]$$

$$s(0) = 0;\ s(T) = 1$$

The interpolation has to be done *slowly*.

According to the **adiabatic theorem**, the time *T* has to be:

$$T \gg \Gamma^2 / \Delta^2_{\min}$$

$$\text{with}\ \ \Gamma^2 = \max_{s \in [0,1]} \left\| \big[\dot{H}(s)\big]^2 \right\|$$

Adiabatic Quantum Computation

$$H(s) = \big[1 - s\big] H_0 + s\, H_P$$

$$s(t) \in [0,1]$$

$$s(0) = 0;\ s(T) = 1$$

The interpolation has to be done *slowly*.

According to the **adiabatic theorem**, the time *T* has to be:

$$T \gg \Gamma^2 / \Delta_{\min}^2 \qquad \text{with} \quad \Gamma^2 = \max_{s \in [0,1]} \left\| \left[\dot{H}(s) \right]^2 \right\|$$

How big is $\Delta_{\min}$?

$\Delta_{\min} \geq 1/\mathrm{poly}(n)$ ⟶ efficient quantum algorithm

$\Delta_{\min} \sim 1/\exp(n)$ ⟶ inefficient quantum algorithm

Hard problems (NPC) are equivalent to finding the gs of *Ising-like spin-glass* Hamiltonians. F. Barahona, *J. Phys. A* ***15****, 3241 (1982)*