Introduction to quantum computing

Yassine Hamoudi

Slides: https://yassine-hamoudi.github.io/intro-qc.pdf/

Image credit: https://commons.wikimedia.org/

Part 1

What is a quantum computer

and how to define it in

mathematical terms

Part 2

What are some

envisioned applications

of quantum computers

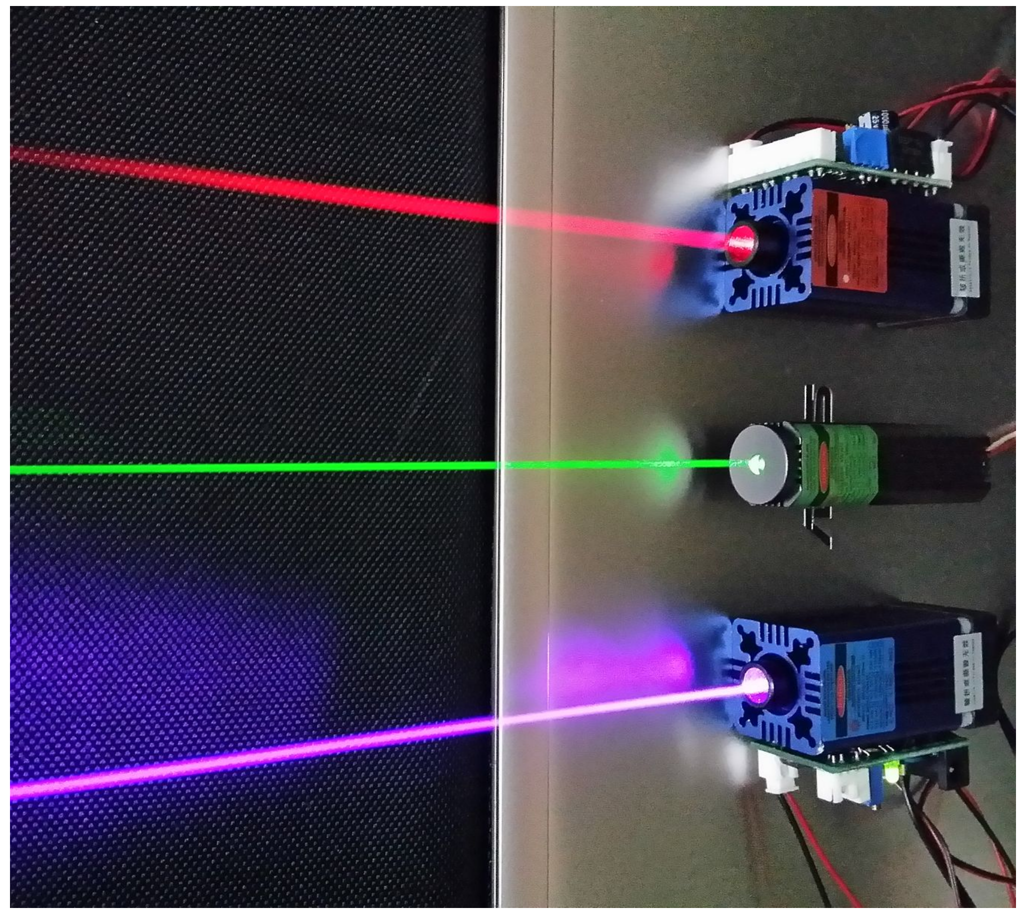
PART 1

What is a quantum computer?

Quantum devices

Stimulated emission

- Laser
- Atomic clock, GPS



Tunnelling

- Flash memory
- Scanning tunneling microscope

Magnetic resonance

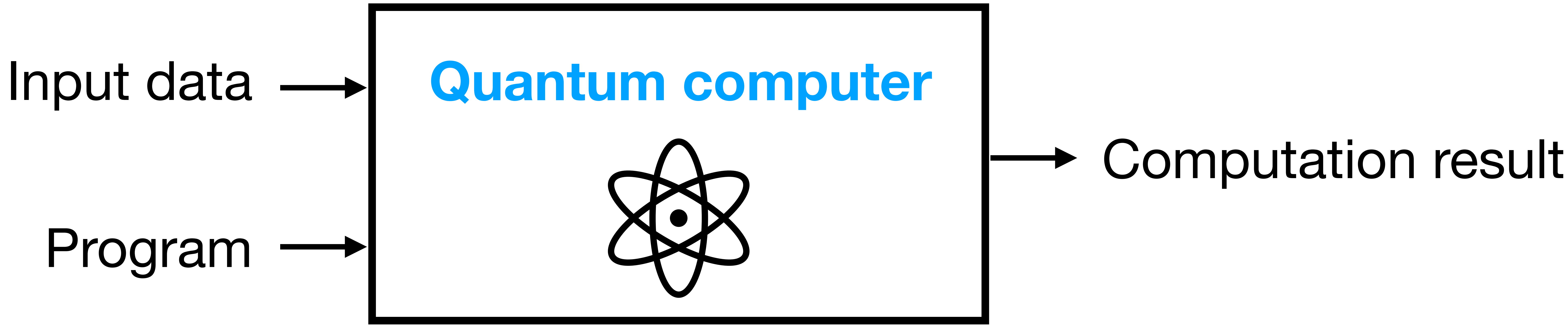
- Magnetic Resonance Imaging
- NMR spectroscopy

Photoelectric & Photovoltaic effect

- Solar panel
- CCD sensor

What is a quantum computer?

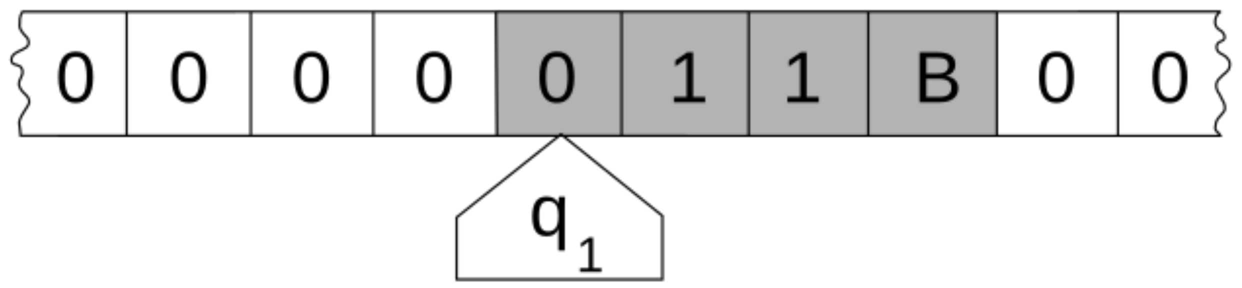
A physical device that exploits the laws of quantum mechanics to perform computational tasks



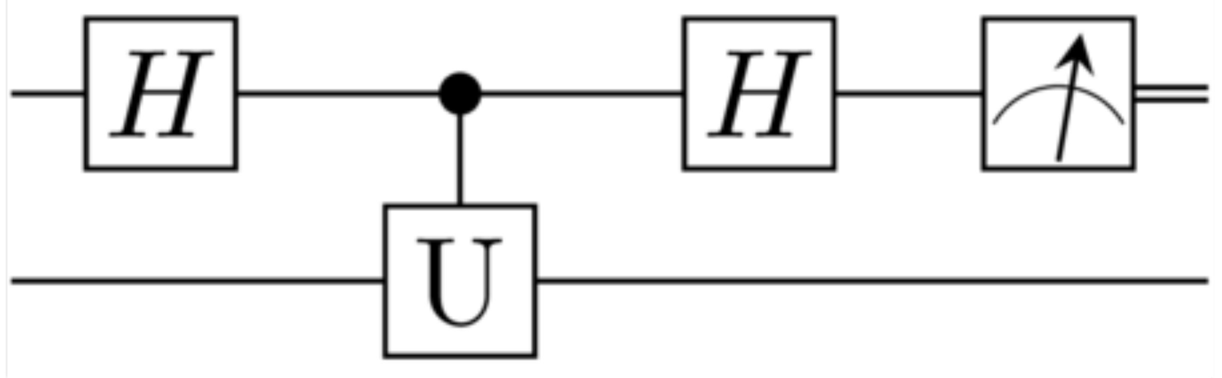
Example of tasks: find integer solutions to $x^2 - 511y^2 = 1$, simulate the FeMoco molecule, find the prime decomposition of $2^{1550019073} - 1$

Do we have quantum computers yet?

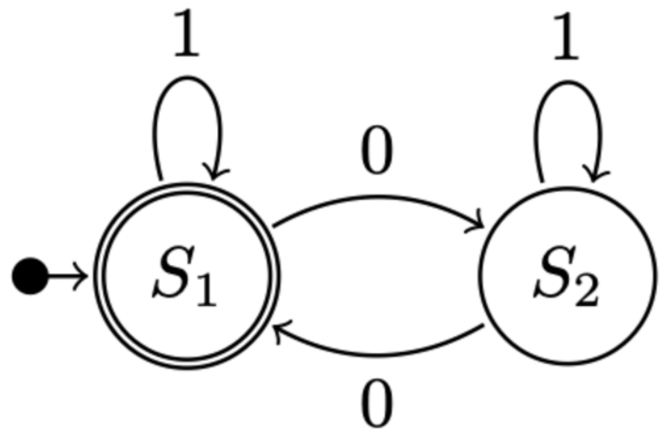
Very neat mathematical models of quantum computation



Q. Turing machine
(~1980-85)



Q. circuit
(~1989-93)



Q. finite automaton
(~1997-2000)

…

… but no scalable, physical realizations of these models yet.

Lots of errors / noise in practice, due to quantum effects

Why do we want quantum computers?

Properties predicted by mathematical models:

1. Faster at solving certain problems
2. New cryptographic tasks that are impossible to achieve with classical computers (q. key distribution, unforgeable q. money…)
3. Computer networks with enhanced properties (more secure communications, better distributed algorithms…)

More details in part 2 of the lecture

What quantum computers **will not do**?

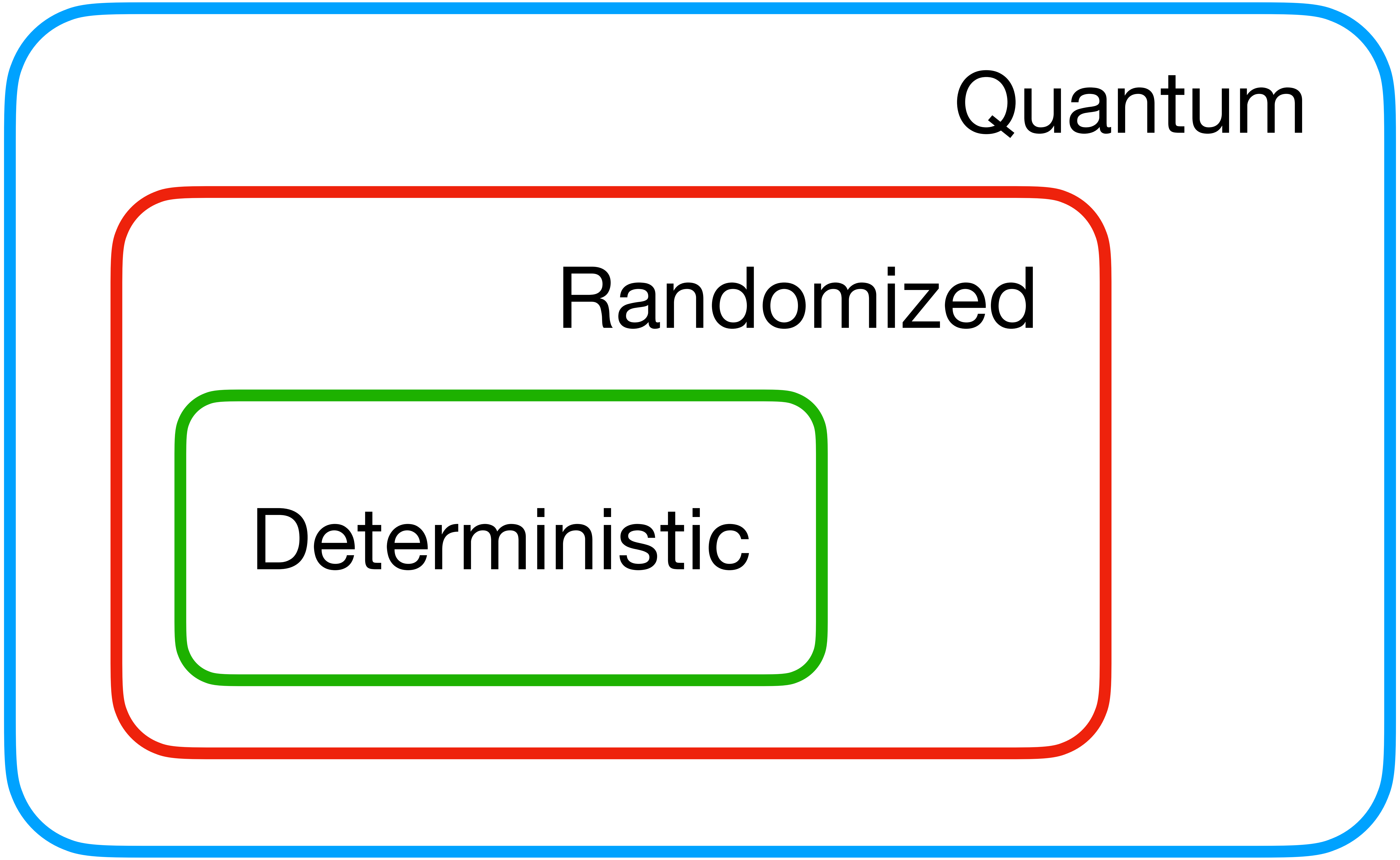
- Speedup every task done by today's computers
  - Relatively few domains of applications in which QC are known to be superior
  - Forecasted as industry/research devices (same as supercomputers, GPU architectures…)
  - Overhead in implementation cost (error correction…) will cancel certain advantages of QC

- Try all solutions to a problem at once / in parallel
  - Properties of quantum mechanics (superposition, interferences,…) are more subtle than that
  - NP-hard problems are believed to remain hard for QC (in complexity-theory terms: $NP \not\subseteq BQP$)

- Break all existing encryption protocols
  - Post-quantum cryptography: study of quantum-safe protocols (lattice-based crypto…)
  - Current attacks (e.g. breaking RSA with Shor's factoring) are out of reach of near term QC

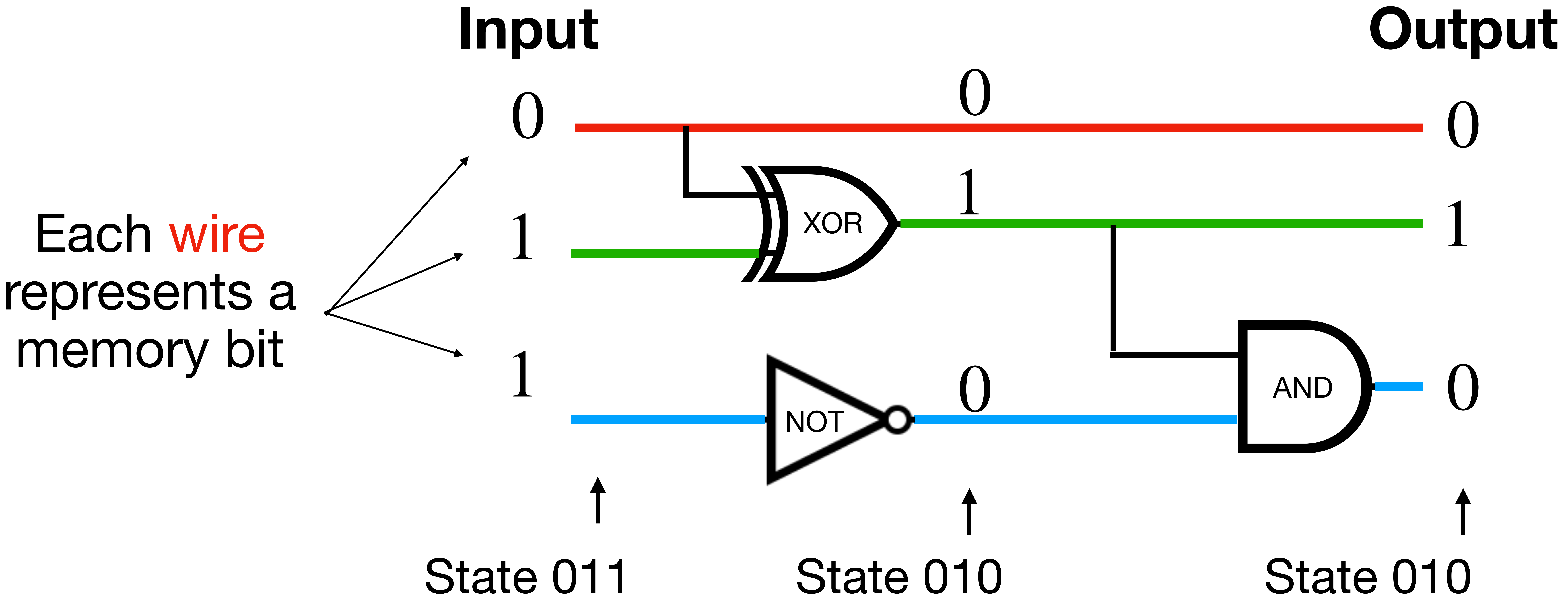
How to (mathematically) construct a quantum computer?

Models of computation



Deterministic computation

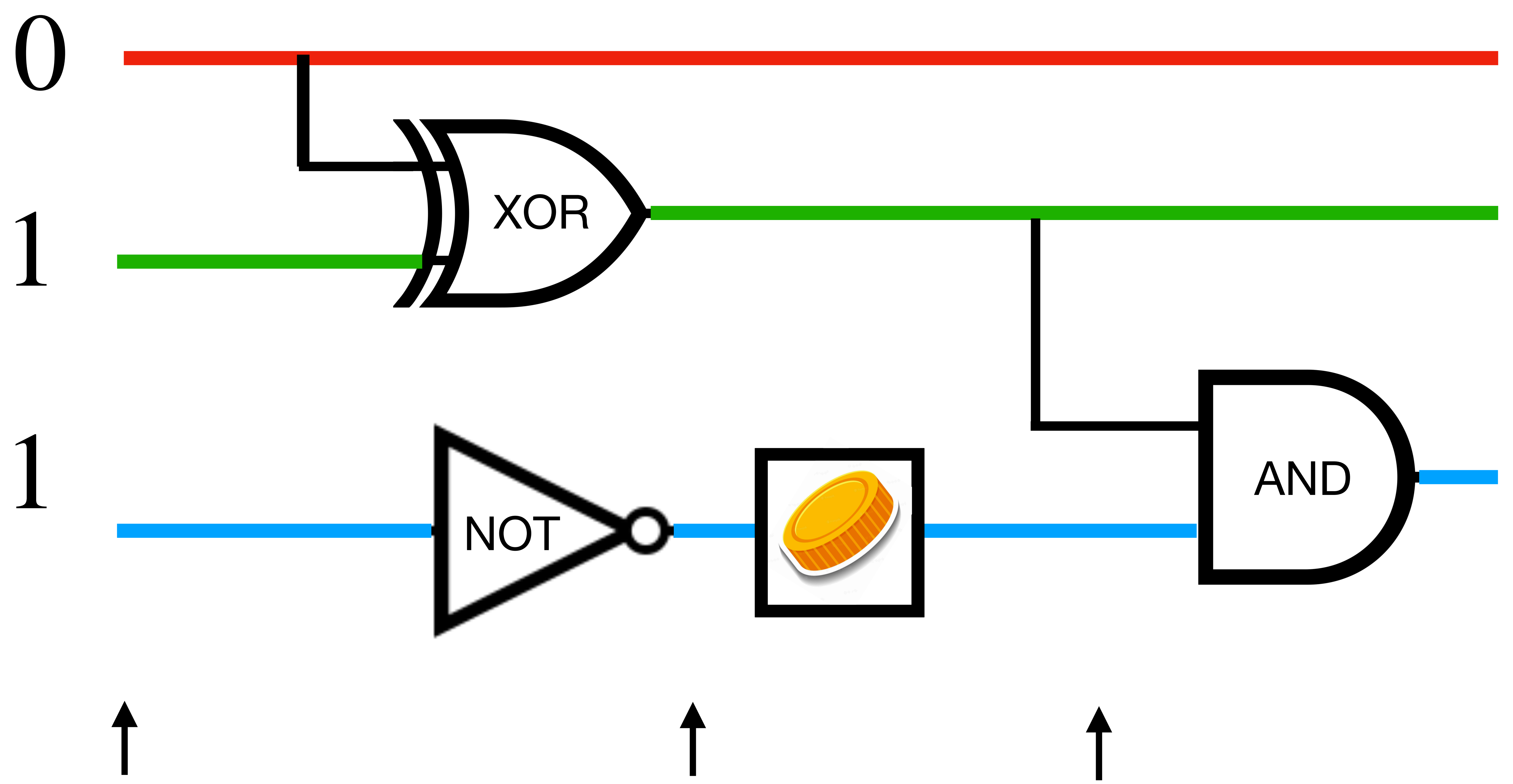
A classical (deterministic) computer can be modeled as a sequence of logic gates operating on a memory of binary cells



Randomized computation

Randomized gate

Flip the bit with probability 1/2



The output is random

Hard to simulate by a deterministic computer when number of grows

Randomized computation

Linear algebra perspective

Basis: $\{|000\rangle, |001\rangle, \ldots, |111\rangle\}$

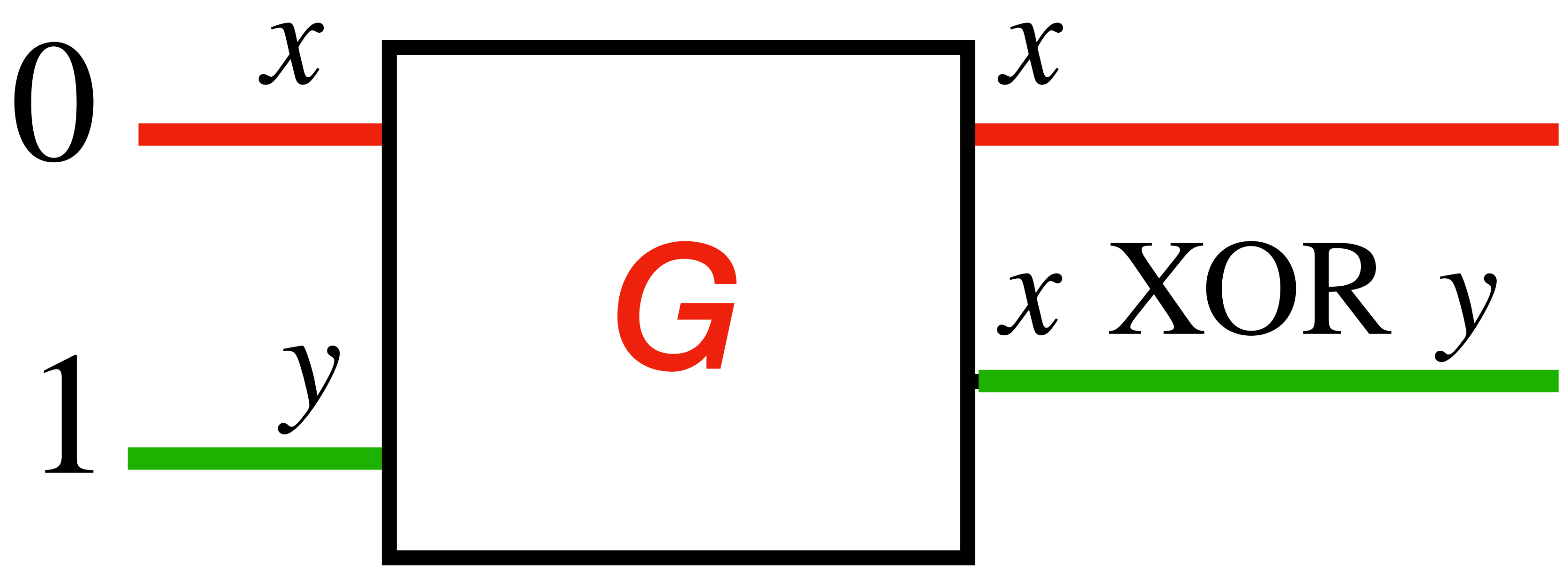
State: probability vector

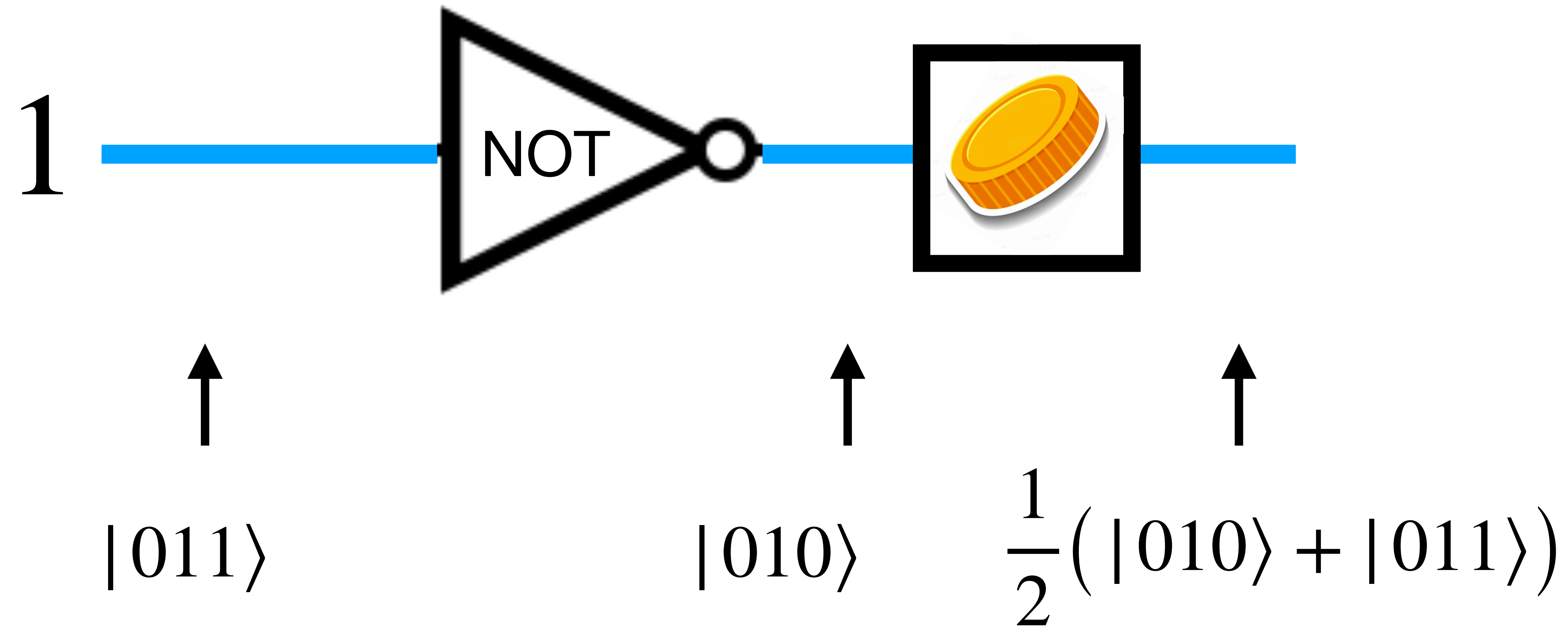
$$\{\,(1/2,\ 010),\ (1/2,\ 011)\,\} \longrightarrow \frac{1}{2}\,|010\rangle + \frac{1}{2}\,|011\rangle$$

Gate: stochastic matrix

$$G = \begin{array}{c} \begin{array}{cccc} |00\rangle & |01\rangle & |10\rangle & |11\rangle \end{array} \\ \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix} \end{array} \begin{array}{c} x\,y \\ |00\rangle \\ |01\rangle \\ |10\rangle \\ |11\rangle \end{array} \longrightarrow$$

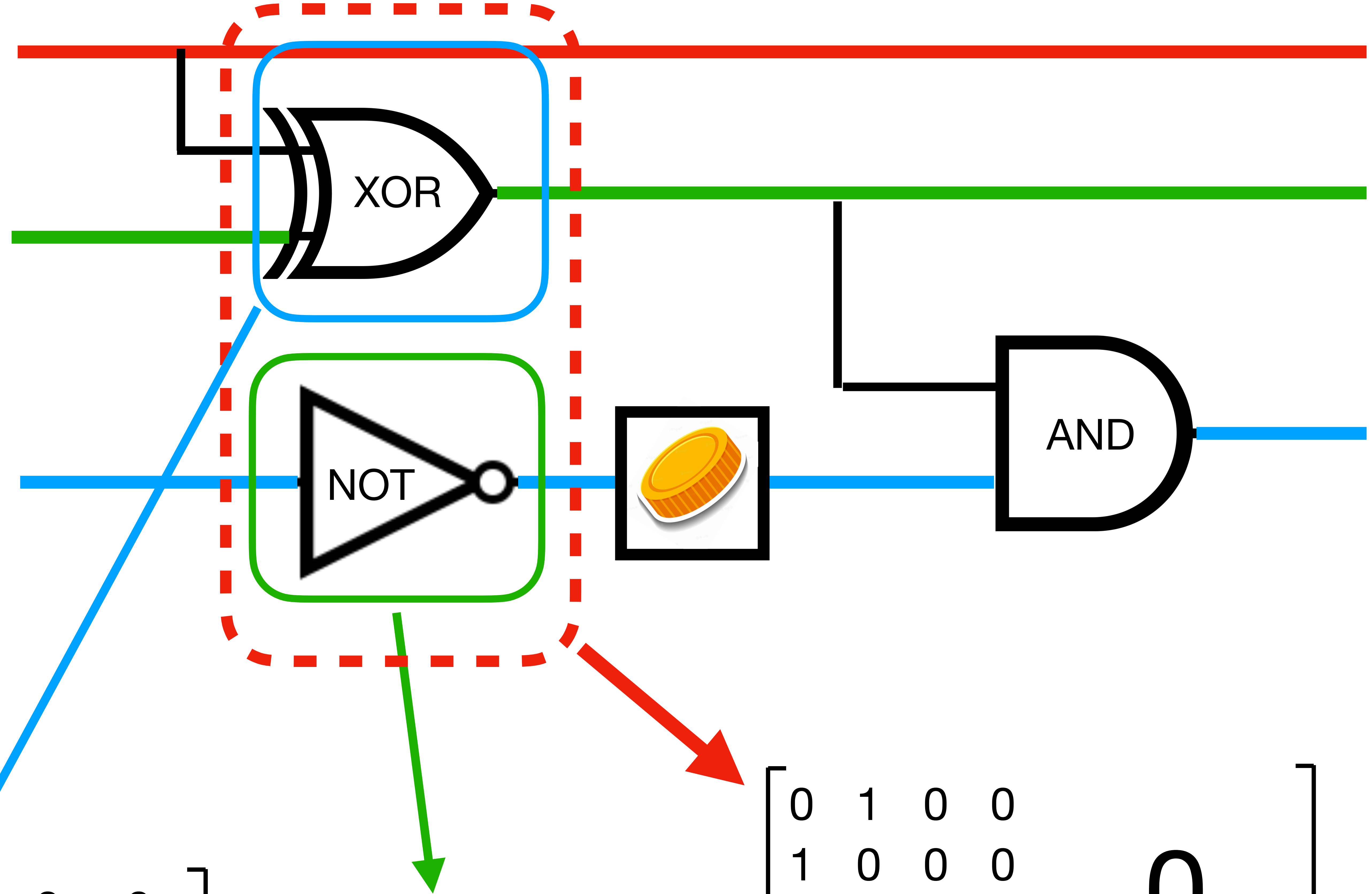
$$G\left(\frac{1}{3}|00\rangle + \frac{2}{3}|10\rangle\right)$$
$$= \frac{1}{3}G|00\rangle + \frac{2}{3}G|10\rangle$$
$$= \frac{1}{3}|00\rangle + \frac{2}{3}|11\rangle$$





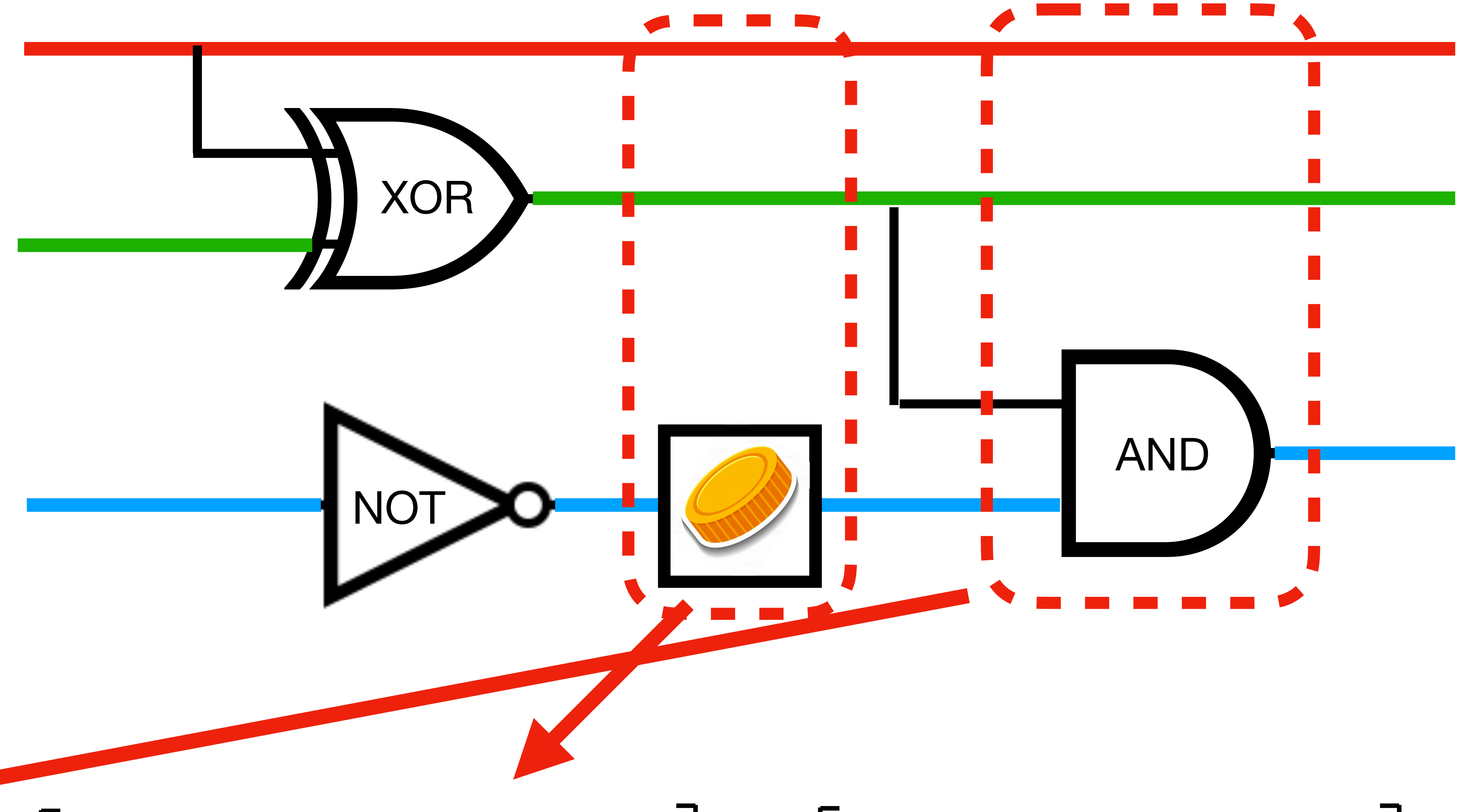
$$= \begin{array}{c} \begin{array}{cc} |0\rangle & |1\rangle \end{array} \\ \begin{bmatrix} 1/2 & 1/2 \\ 1/2 & 1/2 \end{bmatrix} \end{array} \begin{array}{c} \\ |0\rangle \\ |1\rangle \end{array}$$

Randomized computation



$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix} \otimes \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} = \left[\begin{array}{cc} \begin{matrix} 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{matrix} & 0 \\ 0 & \begin{matrix} 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{matrix} \end{array}\right]$$

Randomized computation

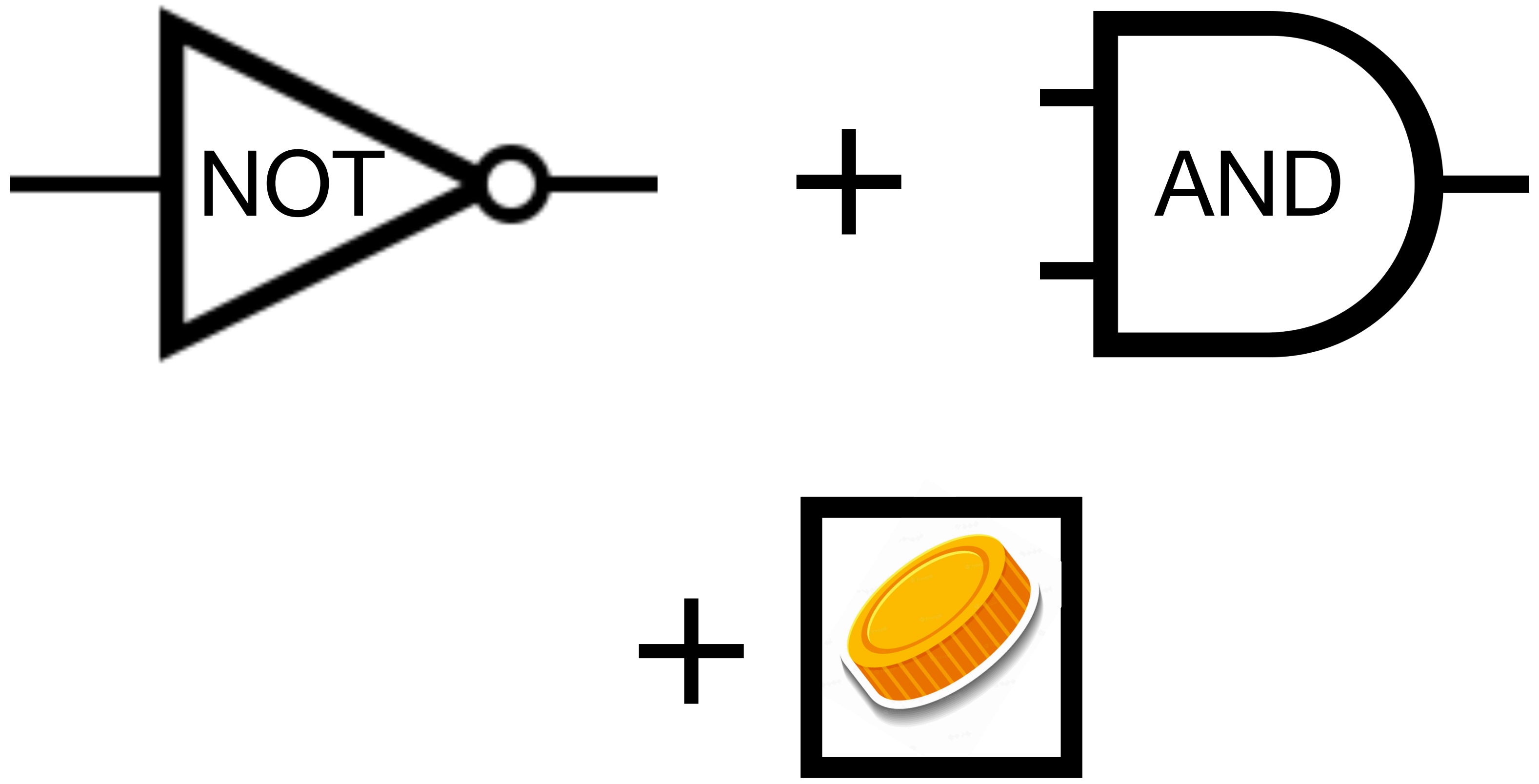


$$
\begin{bmatrix} \begin{matrix}1&0&0&0\\1&0&0&0\\0&0&1&0\\0&0&0&1\end{matrix} & 0 \\ 0 & \begin{matrix}1&0&0&0\\1&0&0&0\\0&0&1&0\\0&0&0&1\end{matrix} \end{bmatrix}
\times
\begin{bmatrix} \begin{matrix}1/2&1/2\\1/2&1/2\end{matrix} & & & \\ & \begin{matrix}1/2&1/2\\1/2&1/2\end{matrix} & & 0 \\ & & \begin{matrix}1/2&1/2\\1/2&1/2\end{matrix} & \\ & 0 & & \begin{matrix}1/2&1/2\\1/2&1/2\end{matrix} \end{bmatrix}
\times
\begin{bmatrix} \begin{matrix}0&1&0&0\\1&0&0&0\\0&0&0&1\\0&0&1&0\end{matrix} & 0 \\ 0 & \begin{matrix}0&0&0&1\\0&0&1&0\\0&1&0&0\\1&0&0&0\end{matrix} \end{bmatrix}
=
\begin{bmatrix} \begin{matrix}1/2&1/2\\1/2&1/2\end{matrix} & 0 & & \\ 0 & \begin{matrix}1/2&1/2\\1/2&1/2\end{matrix} & & 0 \\ & & 0 & \begin{matrix}1/2&1/2\\1/2&1/2\end{matrix} \\ & 0 & \begin{matrix}1/2&1/2\\1/2&1/2\end{matrix} & 0 \end{bmatrix}
$$

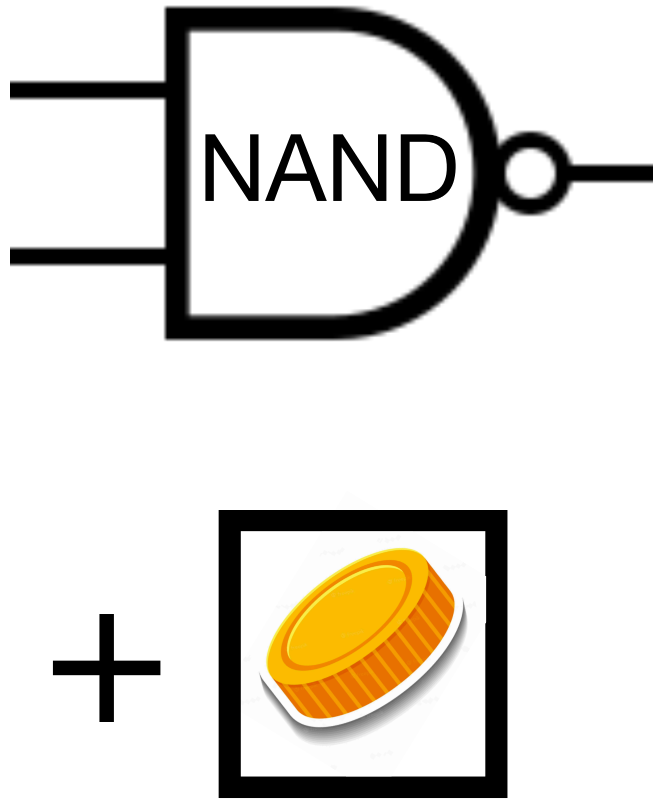
Randomized computation

<u>Any</u> stochastic transformation can be achieved using a universal gate set

Set 1



Set 2



…

Challenge: find circuits of small complexity (depth,size,…) that implement the desired stochastic transformation

Quantum computation

Principle 1: A quantum state is a vector of length 1 in Euclidean norm

Example: $$\frac{1}{\sqrt{3}}\,|0110\rangle - \sqrt{\frac{2}{3}}\,|0101\rangle$$

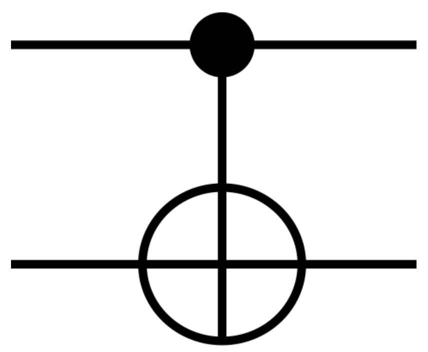
“superposition of 0110 and 0101”

“amplitudes $1/\sqrt{3}$ and $-\sqrt{2/3}$”

Principle 2: A quantum gate is a transformation represented by a unitary matrix

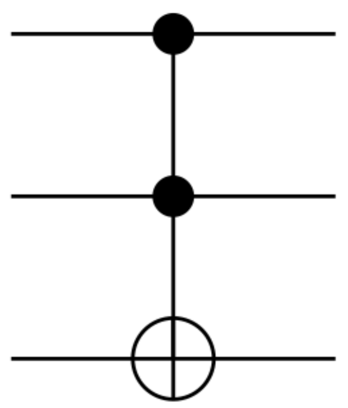
Examples:

All reversible gates



$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix}.$$

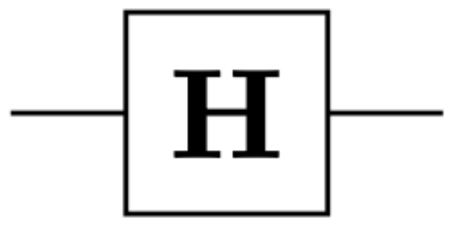
CNOT



$$\begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix}$$

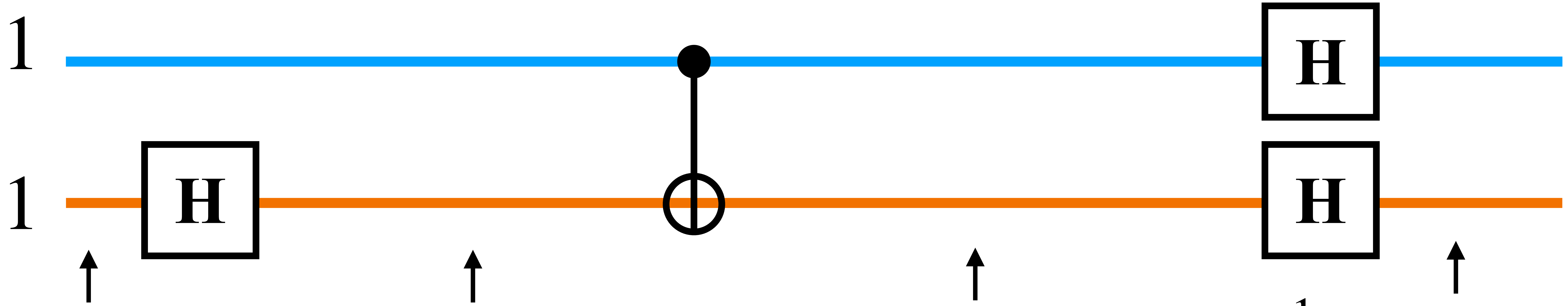
Toffoli

Hadamard gate



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

Quantum computation



Each wire represents a quantum bit (qubit)

$$|11\rangle$$

$$\frac{1}{\sqrt{2}}(|10\rangle - |11\rangle)$$

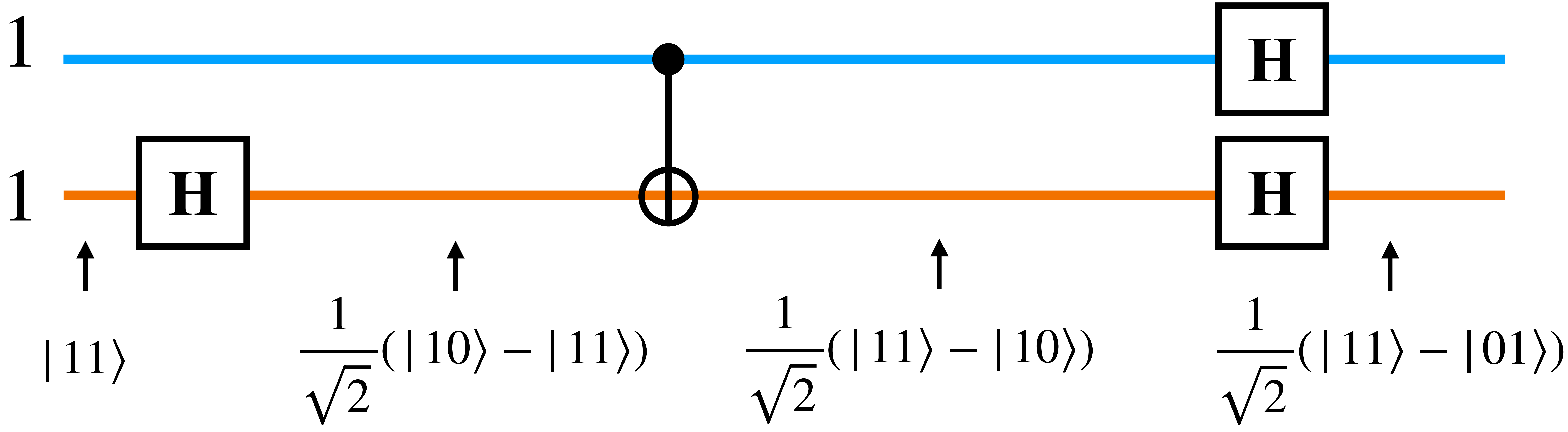
$$\frac{1}{\sqrt{2}}(|11\rangle - |10\rangle)$$

$$\frac{1}{2\sqrt{2}}\Big(\cancel{|00\rangle} - \cancel{|10\rangle} - |01\rangle + |11\rangle - \cancel{|00\rangle} + \cancel{|10\rangle} - |01\rangle + |11\rangle\Big)$$

Destructive interference

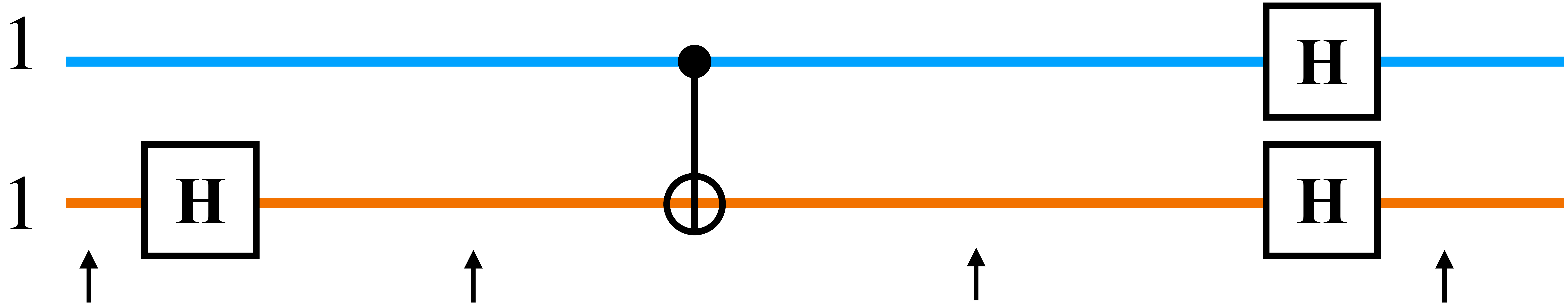
$$= \frac{1}{\sqrt{2}}(|11\rangle - |01\rangle)$$

Quantum computation



$$\begin{bmatrix} \frac{1}{2} & \frac{1}{2} & \frac{1}{2} & \frac{1}{2} \\ \frac{1}{2} & -\frac{1}{2} & \frac{1}{2} & -\frac{1}{2} \\ \frac{1}{2} & \frac{1}{2} & -\frac{1}{2} & -\frac{1}{2} \\ \frac{1}{2} & -\frac{1}{2} & -\frac{1}{2} & \frac{1}{2} \end{bmatrix} \times \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix} \times \begin{bmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & 0 & 0 \\ \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} & 0 & 0 \\ 0 & 0 & \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ 0 & 0 & \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} \end{bmatrix} = \begin{bmatrix} \frac{1}{\sqrt{2}} & 0 & \frac{1}{\sqrt{2}} & 0 \\ 0 & \frac{1}{\sqrt{2}} & 0 & -\frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} & 0 & -\frac{1}{\sqrt{2}} & 0 \\ 0 & \frac{1}{\sqrt{2}} & 0 & \frac{1}{\sqrt{2}} \end{bmatrix}$$

Quantum computation



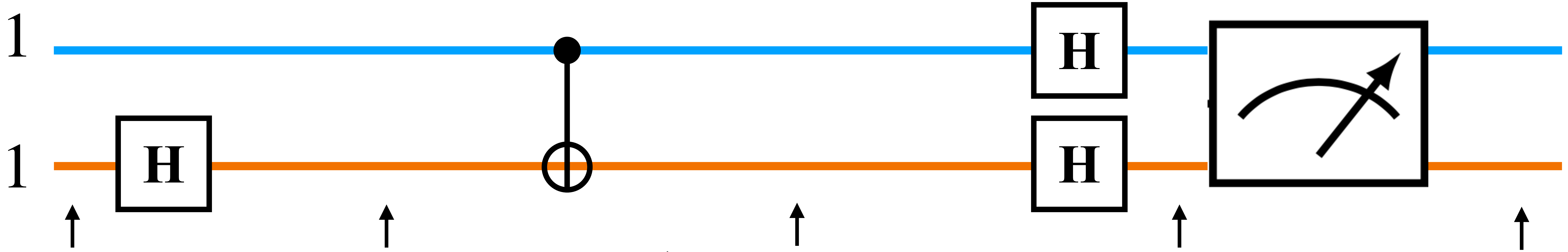
How to interpret that state?

$$|11\rangle \qquad \frac{1}{\sqrt{2}}(|10\rangle - |11\rangle) \qquad \frac{1}{\sqrt{2}}(|11\rangle - |10\rangle) \qquad \frac{1}{\sqrt{2}}(|11\rangle - |01\rangle)$$

Principle 3: A quantum state can be downgraded into a classical (random) state by doing a measurement (= observation). The probabilities are given by the amplitudes squared.

$$\sqrt{\frac{2}{3}}|00\rangle - \frac{i}{\sqrt{6}}|01\rangle + \frac{1}{\sqrt{6}}|11\rangle \longrightarrow \{ (2/3, 00), (1/6, 01), (1/6, 11) \}$$

Quantum computation



$|11\rangle$ $\quad \frac{1}{\sqrt{2}}(|10\rangle - |11\rangle)$ $\quad \frac{1}{\sqrt{2}}(|11\rangle - |10\rangle)$ $\quad \frac{1}{\sqrt{2}}(|11\rangle - |01\rangle)$ $\quad$ { (1/2, 11), (1/2, 01) }

Principle 3: A quantum state can be downgraded into a classical (random) state by doing a measurement (= observation). The probabilities are given by the amplitudes squared.

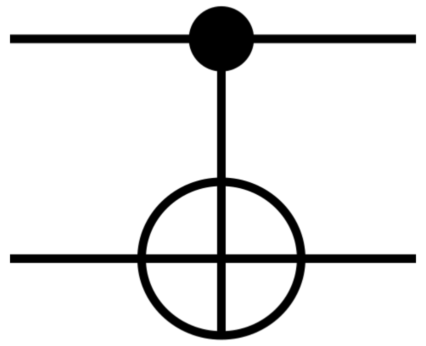
$$\sqrt{\frac{2}{3}}|00\rangle - \frac{i}{\sqrt{6}}|01\rangle + \frac{1}{\sqrt{6}}|11\rangle \longrightarrow \{ (2/3, 00), (1/6, 01), (1/6, 11) \}$$

Quantum computation

<u>Any</u> unitary can be achieved using a universal quantum gate set

Set 1

CNOT



$+$

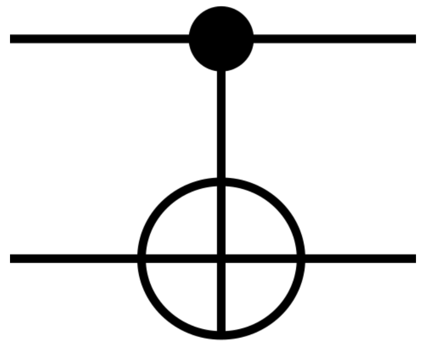
All unitaries on 1 qubit

$$\begin{bmatrix} a & b \\ -e^{i\varphi}\bar{b} & e^{i\varphi}\bar{a} \end{bmatrix}$$

$$|a|^2 + |b|^2 = 1$$

Set 2 (Solovay-Kitaev theorem) ...

CNOT



$+$

Hadamard

$+$

S gate

$$\begin{bmatrix} 1 & 0 \\ 0 & i \end{bmatrix}$$

$+$

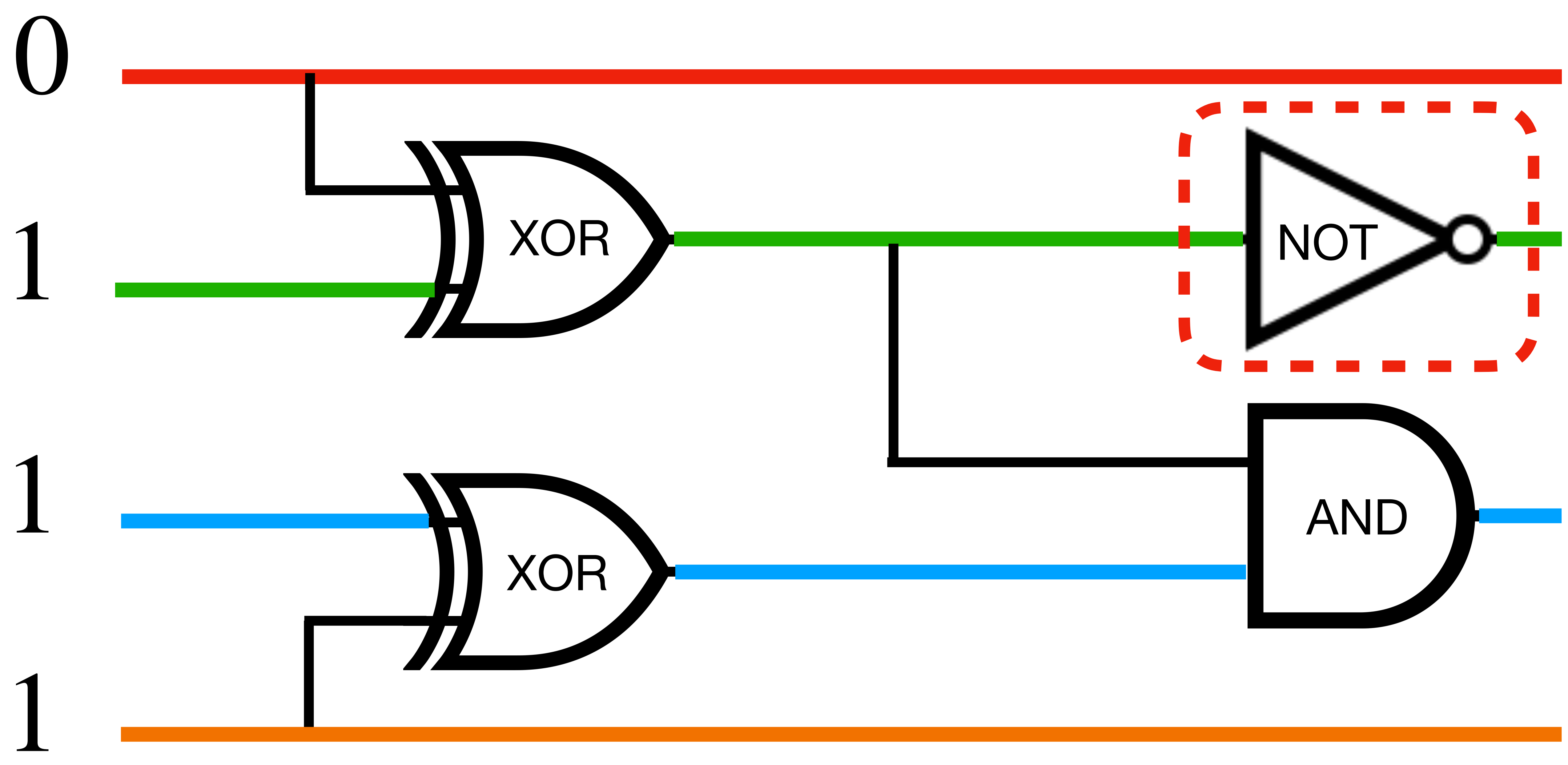
T gate

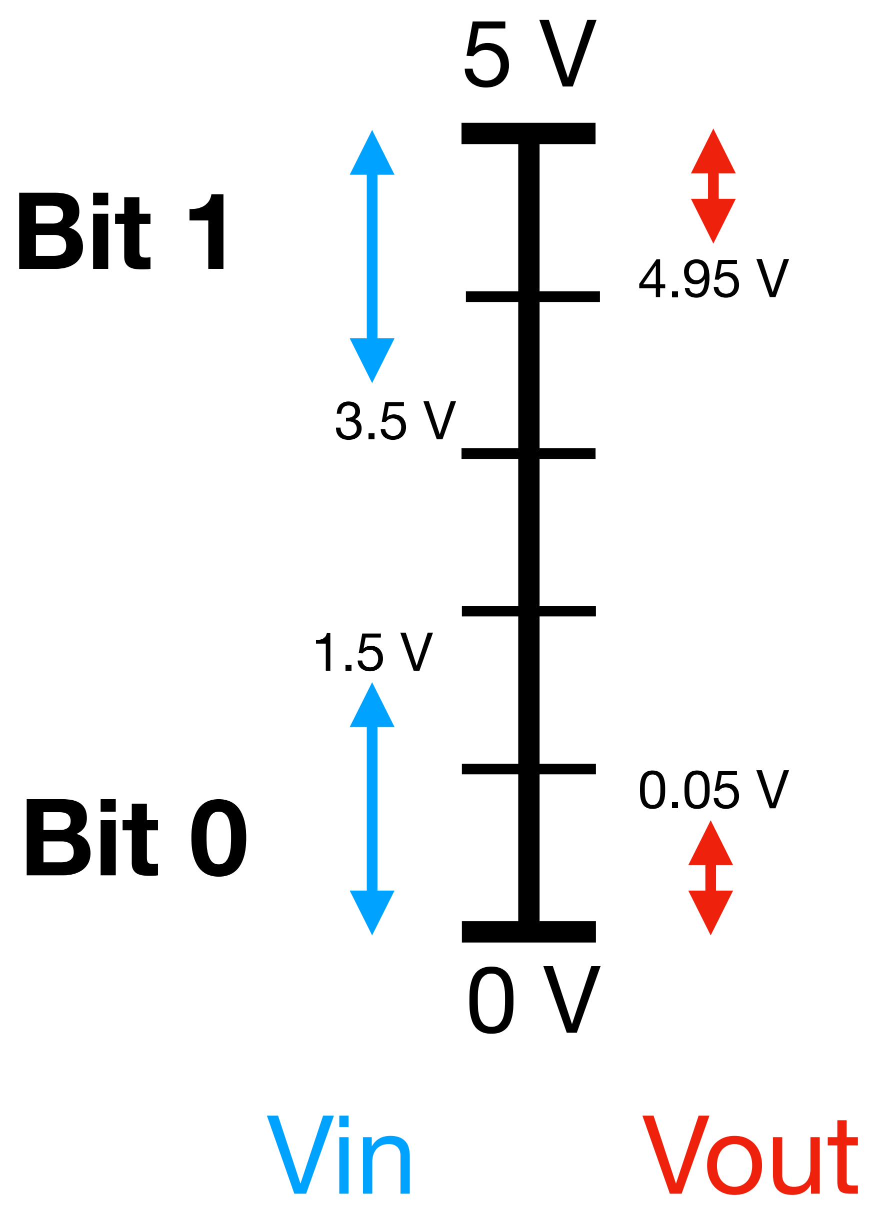
$$\begin{bmatrix} 1 & 0 \\ 0 & e^{i\frac{\pi}{4}} \end{bmatrix}$$

How to (physically) construct a quantum computer?

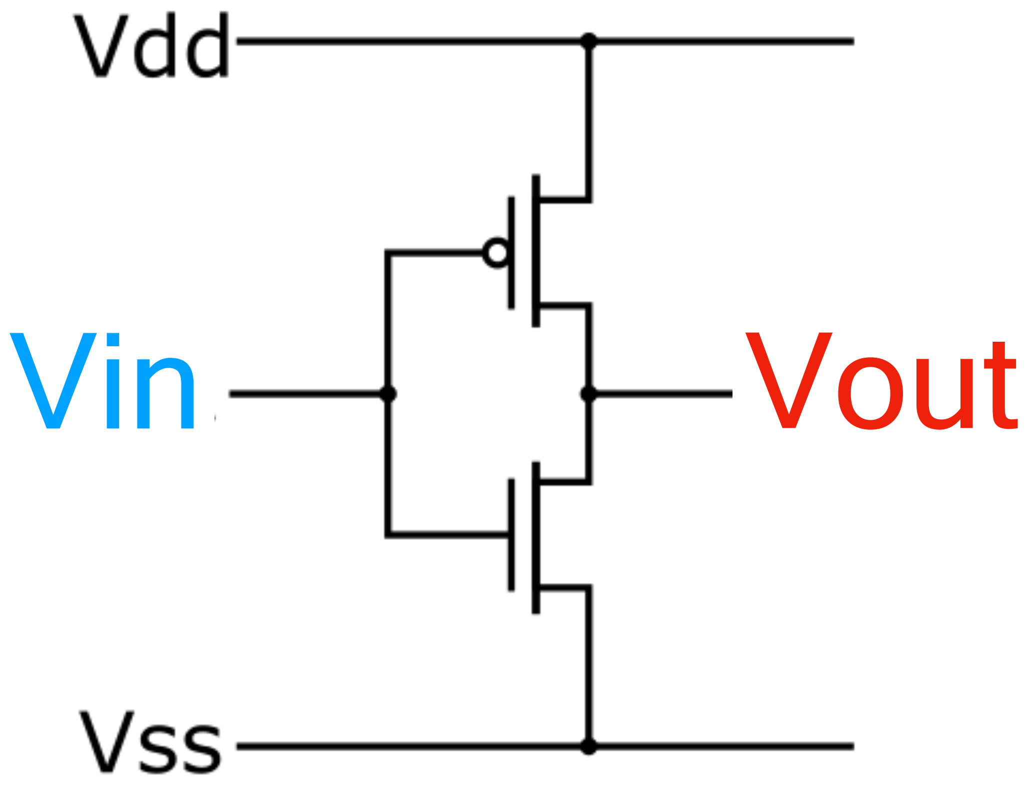
Digital computer

Lots of possible technologies (e.g. transistors)
that match very closely the mathematical model





NOT gate



CMOS inverter

Quantum computation

We don’t have yet the technologies to
construct large-scale quantum computers

Some major challenges:

- imperfections in qubits/gates implementations (noise accumulation)
- decoherence effects (uncontrolled transition from quantum to classical state)

→ Both theoretical and engineering questions

(finding efficient quantum error correcting codes, constructing qubits and gates of good quality, …)

Candidates technologies for physical qubits

Superconductors Google IBM amazon ALICE & BOB

Trapped ions QUANTINUUM IONQ

Photons XANADU PsiQuantum

Neutral atoms PASQAL |QuEra⟩

...

PART 2

Some applications of quantum computing

Area	Example
Simulation of quantum systems	Hamiltonian simulation
Cryptographic attacks	Factoring
Cryptographic protocols	Key distribution
Optimization	Semidefinite programming
Learning	State tomography
...	...

Simulation of quantum systems

Simulation of quantum systems

Simulating a system that evolves according to the laws of quantum mechanics and predicting its properties

The grand motivation for constructing a quantum computer:

"If you want to make a simulation of Nature, you'd better make it quantum mechanical."

Feynman, 1981

Lots of use cases: chemistry (designing new drugs or battery materials…), condensed matter physics, high-energy physics…

Schrödinger equation

The state $|\psi(t)\rangle$ of a quantum system evolving under the dynamic described by a Hamiltonian H is governed by the Schrödinger equation:

$$i\frac{d|\psi(t)\rangle}{dt} = H|\psi(t)\rangle$$

solution

$$|\psi(t)\rangle = \boxed{e^{-iHt}}|\psi(0)\rangle$$

State after t time steps

Unitary operator that we want to simulate

Initial state of the system

Task: Hamiltonian simulation

Given the description of a Hamiltonian H, construct a quantum circuit that takes as input $|\psi(0)\rangle$ and that outputs $|\psi(t)\rangle = e^{-iHt}|\psi(0)\rangle$

Example: $H = -gJ\sum_{j=1}^{n} \sigma_j^x - J\sum_{j=1}^{n} \sigma_j^z \sigma_{j+1}^z$ *(1D transverse field Ising model)*

$2^n \times 2^n$ Hermitian matrix

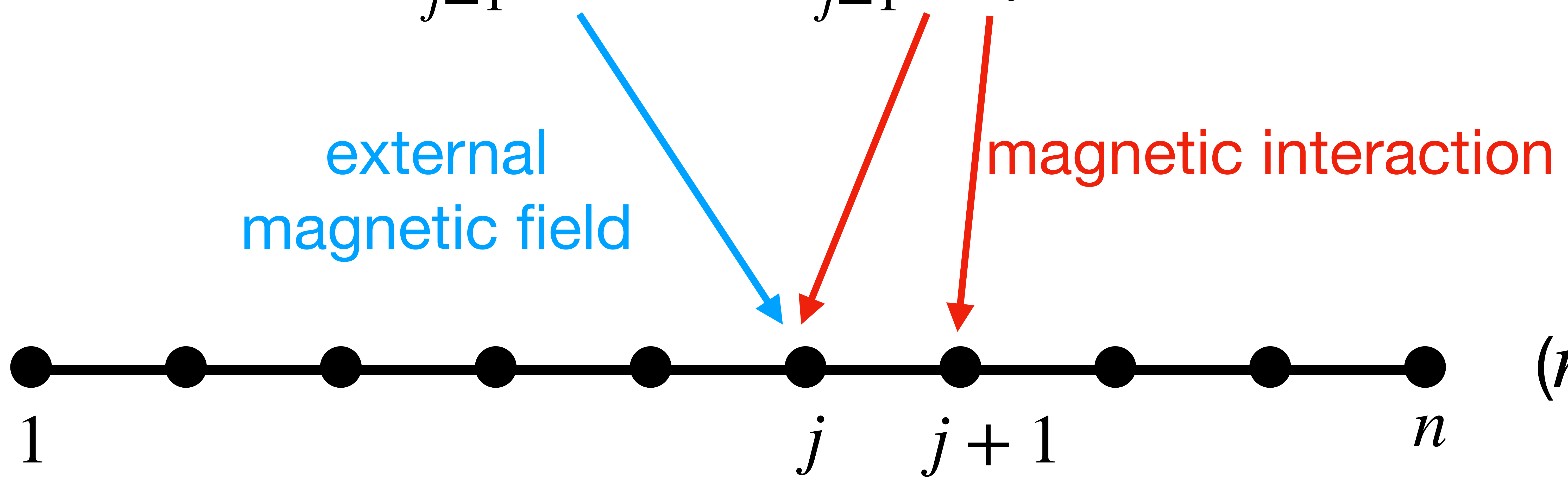
constant numbers

Pauli matrices

Task: Hamiltonian simulation

Given the description of a Hamiltonian H, construct a quantum circuit that takes as input $|\psi(0)\rangle$ and that outputs $|\psi(t)\rangle = e^{-iHt}|\psi(0)\rangle$

Example: $H = -gJ\sum_{j=1}^{n}\sigma_j^x - J\sum_{j=1}^{n}\sigma_j^z\sigma_{j+1}^z$ *(1D transverse field Ising model)*



(n qubits on a line)

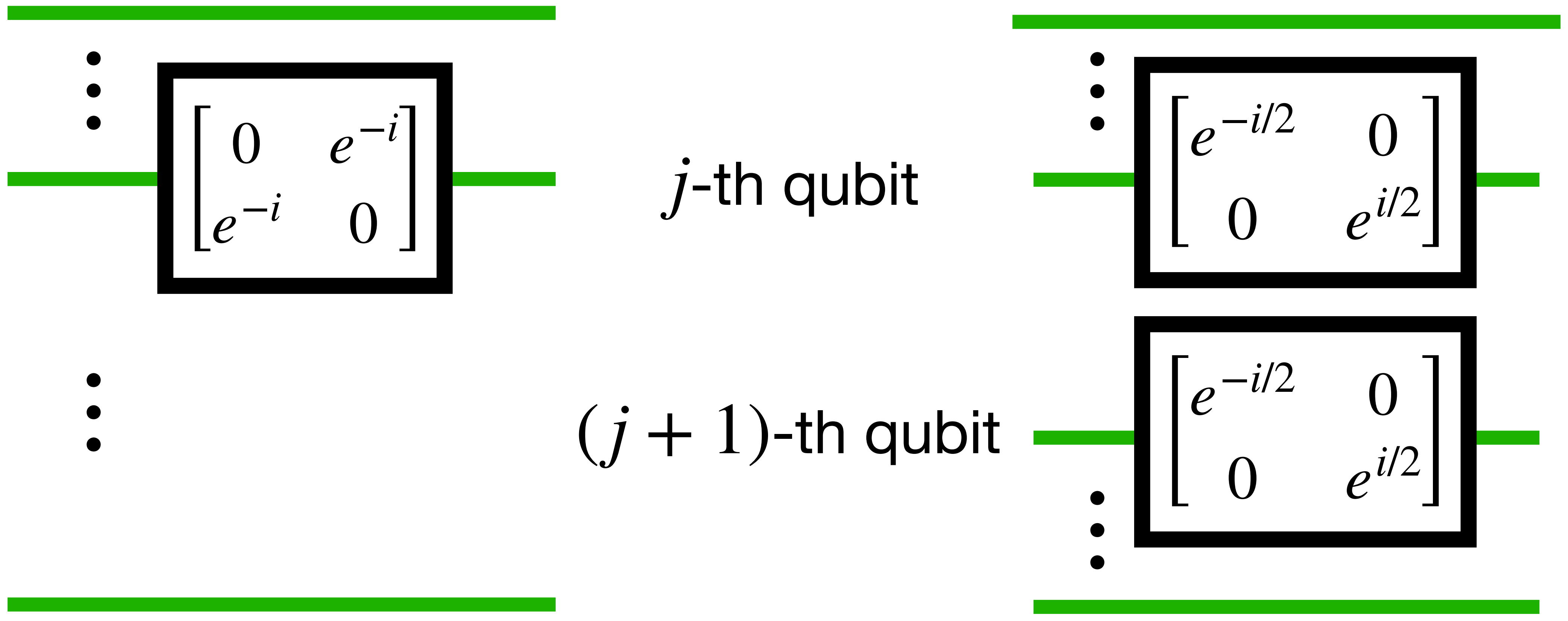
Task: Hamiltonian simulation

Given the description of a Hamiltonian H, construct a quantum circuit that takes as input $|\psi(0)\rangle$ and that outputs $|\psi(t)\rangle = e^{-iHt}|\psi(0)\rangle$

Example: $H = -gJ\sum_{j=1}^{n}\sigma_j^x - J\sum_{j=1}^{n}\sigma_j^z\sigma_{j+1}^z$ *(1D transverse field Ising model)*

How to simulate $e^{-i\sigma_j^x}$

How to simulate $e^{-i\sigma_j^z\sigma_{j+1}^z}$



σ_j^x **and** $\sigma_j^z\sigma_{j+1}^z$ **don't commute**

$$e^{-iHt} \neq \prod_{j=1}^{n} e^{-i\sigma_j^x gJt} \prod_{j=1}^{n} e^{-i\sigma_j^z\sigma_{j+1}^z Jt}$$

Trotter-Suzuki product formulas

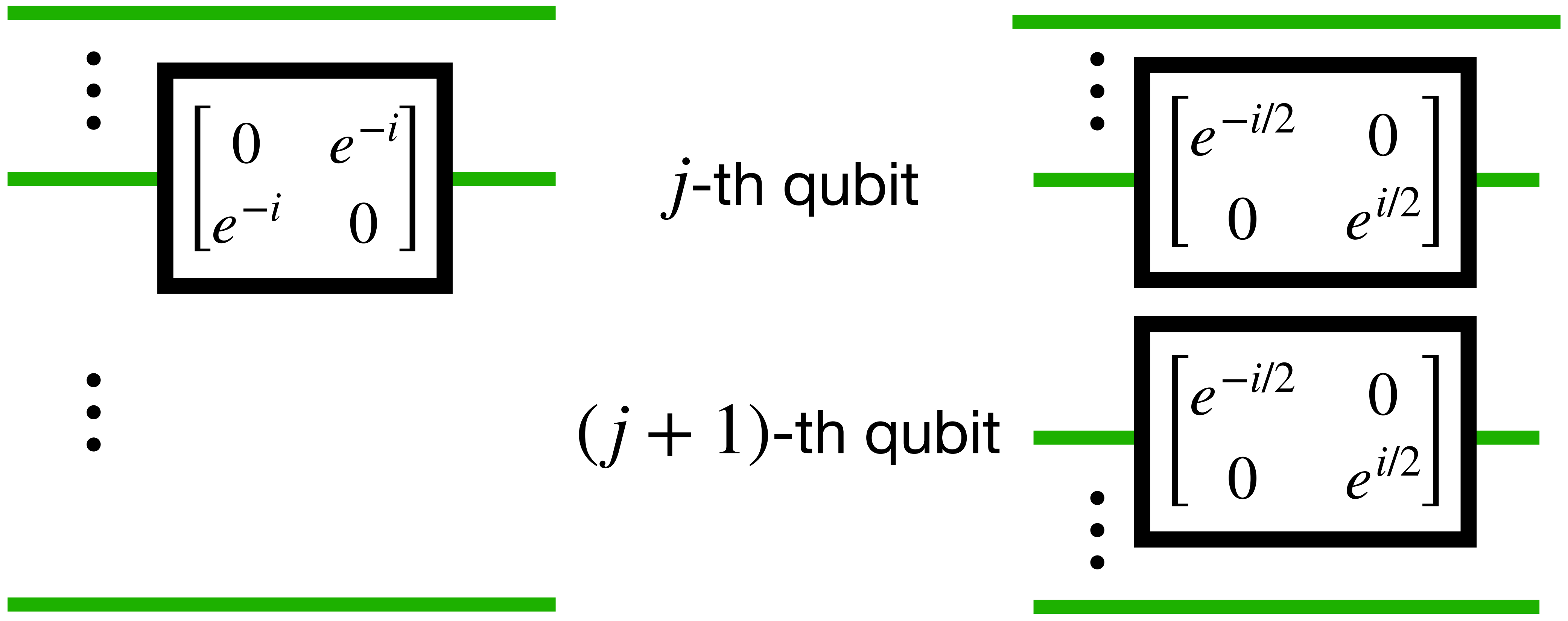
$$e^{-iHt} \underset{\Delta \to 0}{=} \left(\prod_{j=1}^{n} e^{-i\sigma_j^x gJ\Delta} \prod_{j=1}^{n} e^{-i\sigma_j^z\sigma_{j+1}^z J\Delta}\right)^{t/\Delta}$$

Task: Hamiltonian simulation

Given the description of a Hamiltonian H, construct a quantum circuit that takes as input $|\psi(0)\rangle$ and that outputs $|\psi(t)\rangle = e^{-iHt}|\psi(0)\rangle$

Example: $H = -gJ\sum_{j=1}^{n}\sigma_j^x - J\sum_{j=1}^{n}\sigma_j^z\sigma_{j+1}^z$ *(1D transverse field Ising model)*

How to simulate $e^{-i\sigma_j^x}$ *How to simulate* $e^{-i\sigma_j^z\sigma_{j+1}^z}$



Simulation method

Alternate between the Pauli's with tiny Δ-steps

Trotter-Suzuki product formulas

$$e^{-iHt} \underset{\Delta\to 0}{=} \left(\prod_{j=1}^{n} e^{-i\sigma_j^x gJ\Delta}\prod_{j=1}^{n} e^{-i\sigma_j^z\sigma_{j+1}^z J\Delta}\right)^{t/\Delta}$$

Task: Hamiltonian simulation

The product formulas method can simulate e^{-iHt} on n qubits with accuracy ε (in op. norm) at a cost proportional to nt^2/ε

(exponential speedup over best known classical algos)

More advanced methods with even better cost:

- Quantum Walks
- Linear Combination of Unitaries
- Quantum Singular Value Transformation
- ...

Cryptographic attacks

Task 1: Factoring

Find the prime factors of an integer

Large fraction of crypto built on the assumption that Factoring is hard

Breakthrough in 1994 by Peter Shor: an efficient quantum algorithm

→ Factoring-based protocols (e.g. RSA) are not safe against quantum computers

→ Triggered a lot of research on quantum computing and cryptography

Part of a larger family of quantum attacks for Hidden Subgroup Problems

(discrete log, Simon's problem, Dihedral Coset Problem…)

Task 2: Simon's problem

A toy problem invented in 1994 that displays an exponential quantum speedup and inspired Shor's algorithm

Find the secret $s \in \{0,1\}^n$ hidden into a function $f : \{0,1\}^n \to \{0,1\}^n$ promised to be $f(x) = f(y)$ if and only if $y = x \oplus s$.

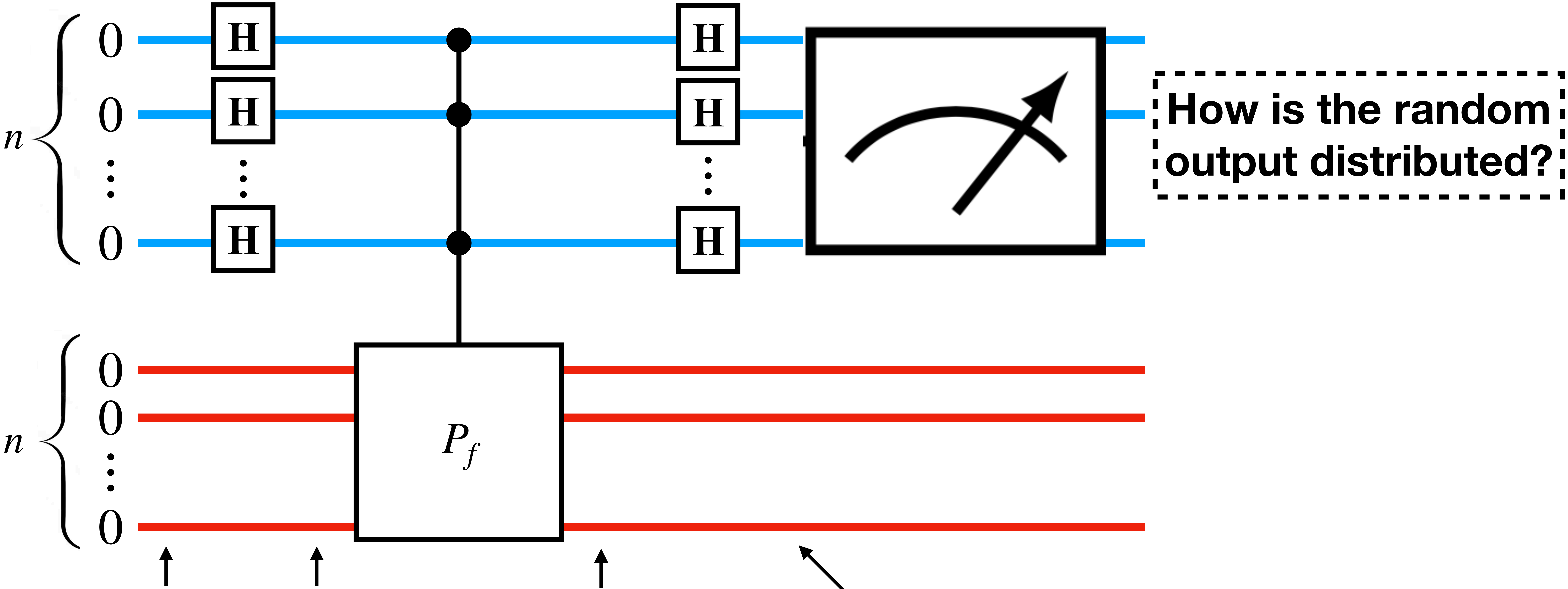
(**Scenario:** s is obfuscated into a program P_f that evaluates f)

Classical algorithm: 1/ evaluate P_f on random $x_1, x_2, x_3\dots$ until finding $f(x_i) = f(x_j)$
2/ output $s = x_i \oplus x_j$

Birthday paradox: $\approx 2^{n/2}$ evaluations before it succeeds

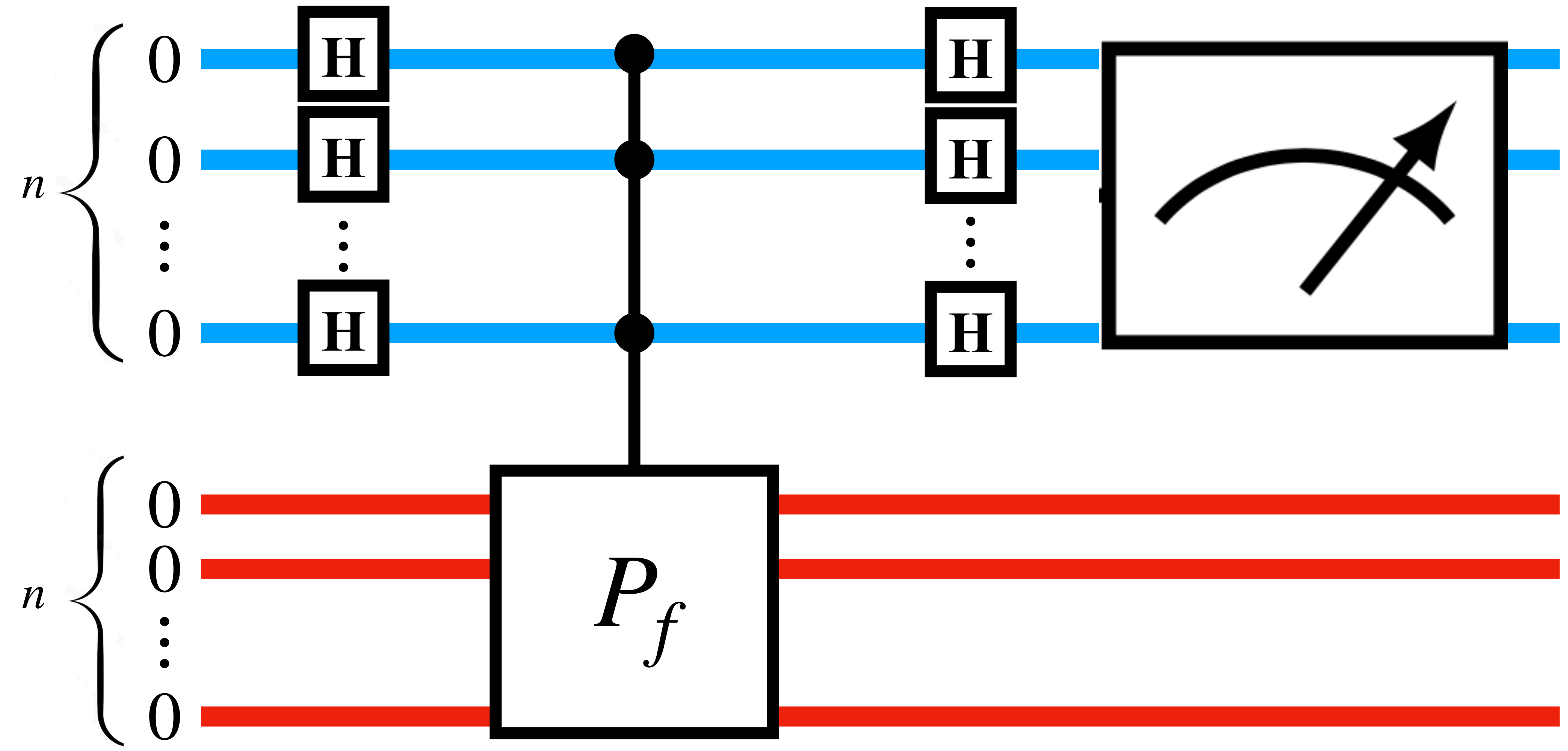
Quantum algorithm: only $\approx n$ evaluations (in superposition)

Task 2: Simon's problem



$|0\ldots0, 0\ldots0\rangle$ $\quad \frac{1}{2^{n/2}}\sum_{x=0}^{2^n-1}|x, 0\ldots0\rangle$ $\quad \frac{1}{2^{n/2}}\sum_{x=0}^{2^n-1}|x, f(x)\rangle$ $\quad \frac{1}{2^n}\sum_{x,y=0}^{2^n-1}(-1)^{\langle x,y\rangle}|y, f(x)\rangle$

Task 2: Simon's problem



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

Key property

The output follows the uniform distribution over the set

$$s^{\perp} = \{y \in \{0,1\}^n : \langle y, s\rangle = 0\}$$

Linear equation in s

Overall algorithm

Repeat the procedure $\approx n$ times to obtain a system of n linear independent equations, and solve it by Gaussian elimination

Further readings

Quantum computing 40 years later

John Preskill

Forty years ago, Richard Feynman proposed harnessing quantum physics to build a more powerful kind of computer. Realizing Feynman's vision is one of the grand challenges facing 21st century science and technology. In this article, we'll recall Feynman's contribution that launched the quest for a quantum computer, and assess where the field stands 40 years later.

https://arxiv.org/abs/2106.10522

Quantum Computing: Lecture Notes

Ronald de Wolf (QuSoft, CWI and University of Amsterdam)

This is a set of lecture notes suitable for a Master's course on quantum computation and information from the perspective of theoretical computer science. The first version was written in 2011, with many extensions and improvements in subsequent years. The first 10 chapters cover the circuit model and the main quantum algorithms (Deutsch-Jozsa, Simon, Shor, Hidden Subgroup Problem, Grover, quantum walks, Hamiltonian simulation and HHL). They are followed by 4 chapters about complexity, 4 chapters about distributed ("Alice and Bob") settings, a chapter about quantum machine learning, and a final chapter about quantum error correction. Appendices A and B give a brief introduction to the required linear algebra and some other mathematical and computer science background. All chapters come with exercises, with some hints provided in Appendix C.

https://arxiv.org/abs/1907.09415

Quantum algorithms: A survey of applications and end-to-end complexities

Alexander M. Dalzell, Sam McArdle, Mario Berta, Przemyslaw Bienias, Chi-Fang Chen, András Gilyén, Connor T. Hann, Michael J. Kastoryano, Emil T. Khabiboulline, Aleksander Kubica, Grant Salton, Samson Wang, Fernando G. S. L. Brandão

The anticipated applications of quantum computers span across science and industry, ranging from quantum chemistry and many-body physics to optimization, finance, and machine learning. Proposed quantum solutions in these areas typically combine multiple quantum algorithmic primitives into an overall quantum algorithm, which must then incorporate the methods of quantum error correction and fault tolerance to be implemented correctly on quantum hardware. As such, it can be difficult to assess how much a particular application benefits from quantum computing, as the various approaches are often sensitive to intricate technical details about the underlying primitives and their complexities. Here we present a survey of several potential application areas of quantum algorithms and their underlying algorithmic primitives, carefully considering technical caveats and subtleties. We outline the challenges and opportunities in each area in an "end-to-end" fashion by clearly defining the problem being solved alongside the input-output model, instantiating all "oracles," and spelling out all hidden costs. We also compare quantum solutions against state-of-the-art classical methods and complexity-theoretic limitations to evaluate possible quantum speedups.

https://arxiv.org/abs/2310.03011

Slides: https://yassine-hamoudi.github.io/intro-qc.pdf/