RUTGERS

THE STATE UNIVERSITY OF NEW JERSEY

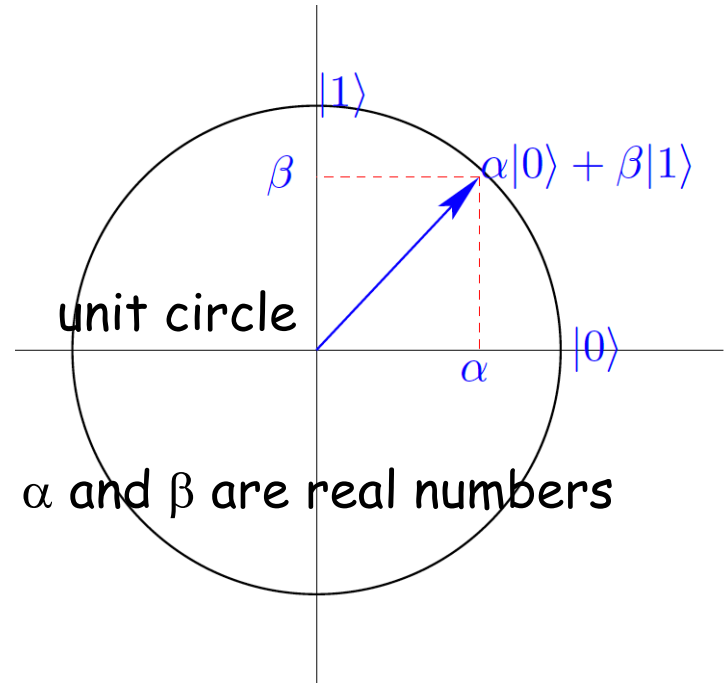
CS 516 Compilers and Programming Languages II

Quantum Computing-3

Review

- cbits are just a special case of qbits:

$$\begin{pmatrix}\alpha\\ \beta\end{pmatrix} \quad \text{or} \quad \alpha|0\rangle + \beta|1\rangle \quad \text{with} \quad \|\alpha\|^2 + \|\beta\|^2 = 1$$

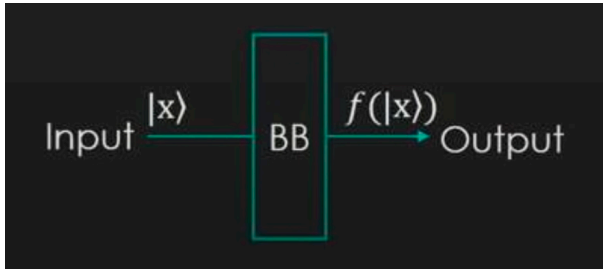


- qbits can be in superposition, and are probabilistically collapsed to cbits by measurement
- Multi-qbit systems are tensor products of single-qbit systems
- Matrices represent operations on qbits and need to be reversible
- The Hadamard gate takes 0-bits and 1-bits to equal superposition, and back
- qbits and their operations are forming a state machine on the unit circle if limited to real numbers (unit sphere if complex numbers)

The Deutsch Oracle

Can a quantum computer "beat" a classical computer?

- Imagine someone gives you a "black box" (BB) containing a function on one bit. There are four possibilities:
  - constant 0
  - constant 1
  - identity
  - negation



$|x\rangle = |0\rangle$
or
$|x\rangle = |1\rangle$

- Question: How many queries (try an input and observe the output) to distinguish constant from non-constant (variable) functions (const 0 & 1 vs. identity & negation are the two categories) assuming a reversible implementation of the "black box"?

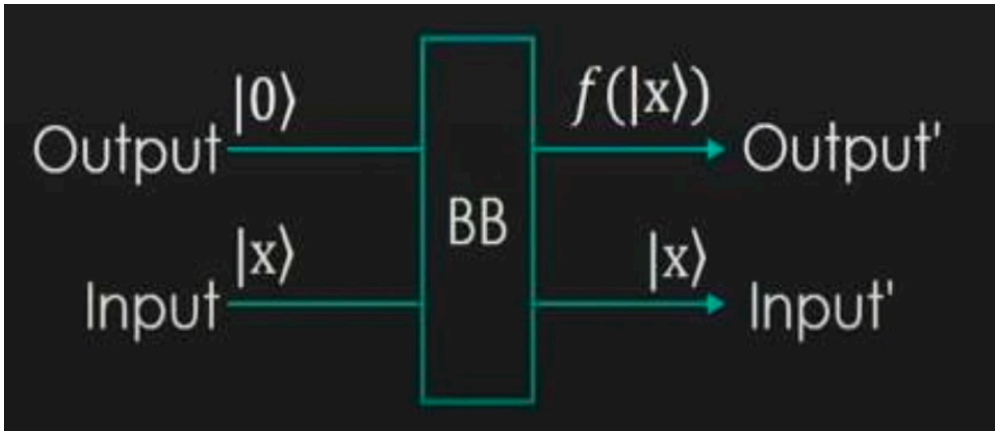
  Classical computer? and Quantum computer?

The Deutsch Oracle

Can a quantum computer "beat" a classical computer?

Need to "rewire" the black box to make it reversible.
Well, we know the "trick": add additional bits!

$|x\rangle = |0\rangle$
or
$|x\rangle = |1\rangle$



- Additional input through **Output** qbit assumed to be constant $|0\rangle$
- Additional output through **Input'** qbit which "passes through" **Input** qbit
- Function is mapping from **Input** to **Output'** qbits

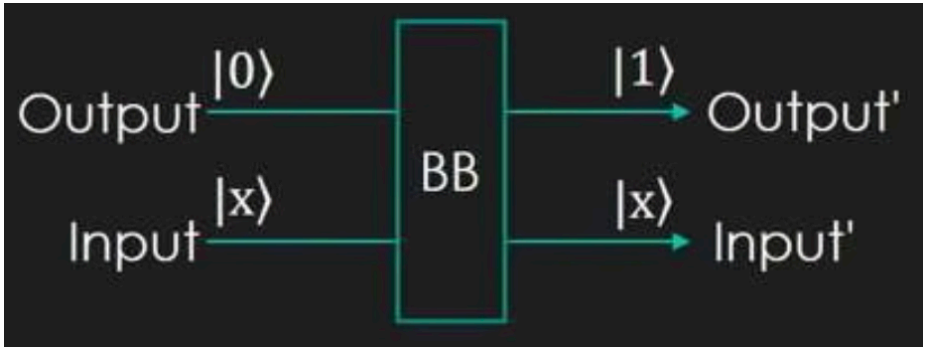
The Deutsch Oracle

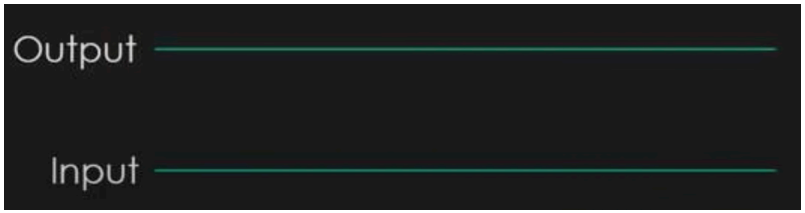
Can a quantum computer "beat" a classical computer?

Constant-0 and Constant-1 implementations

Constant-0

Constant-1





|0⟩

|1⟩

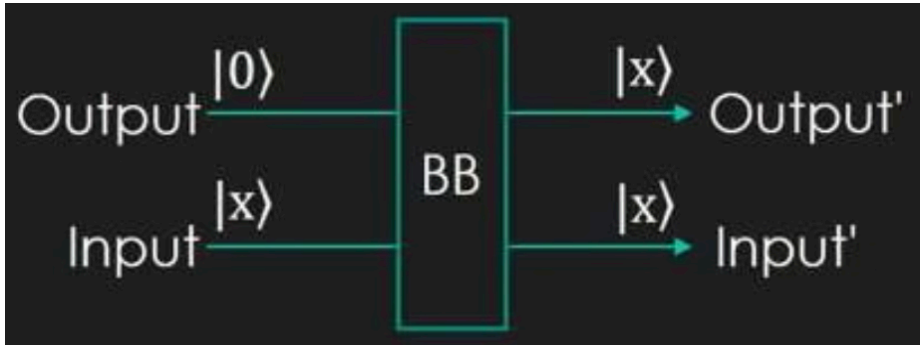
X-gate: Negation

The Deutsch Oracle

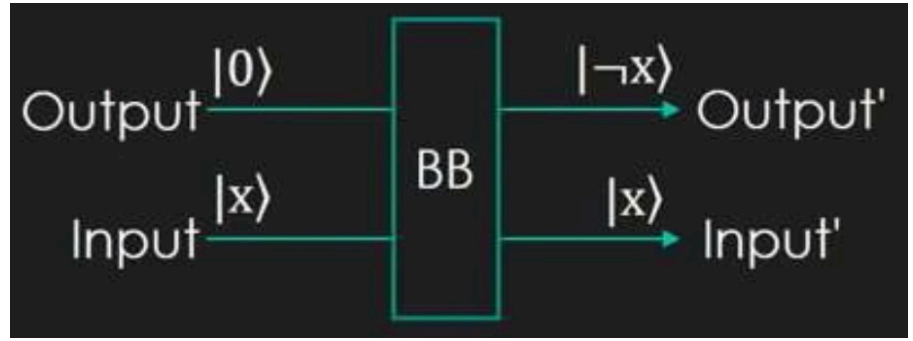
Can a quantum computer "beat" a classical computer?

Identity and Negation implementations

Identity

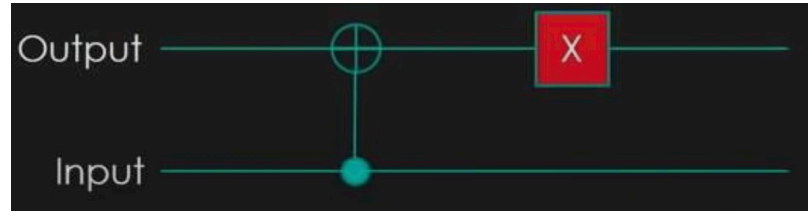


Negation



|x⟩

CNOT-gate: controlled NOT



|¬x⟩

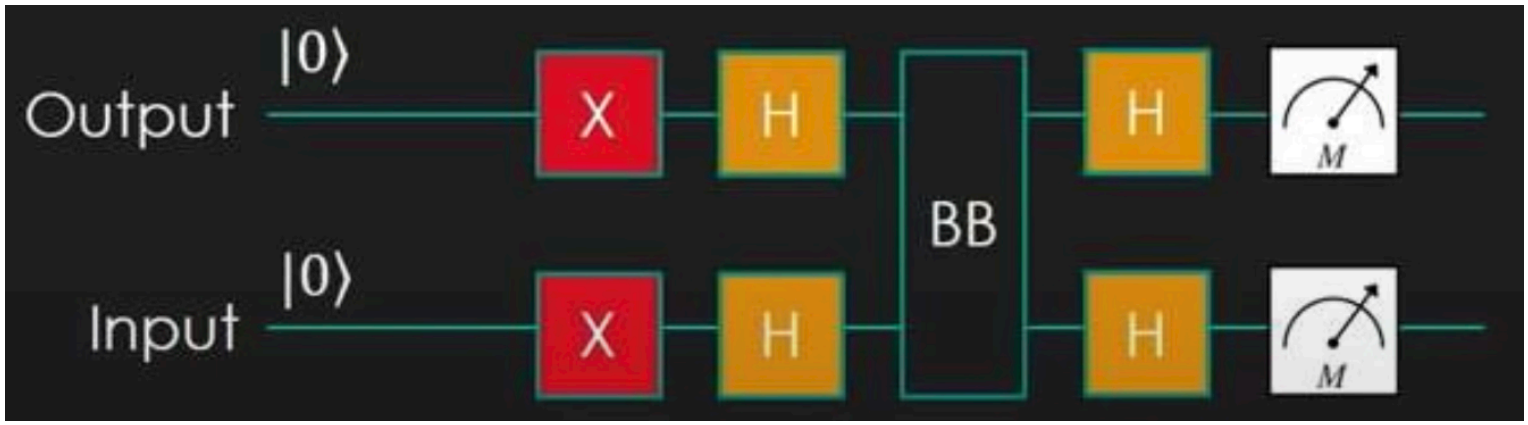
CNOT-gate: controlled NOT

X-gate: Negation

The Deutsch Oracle

Can a quantum computer "beat" a classical computer?

Can we now come up with **a single query** to identify the two categories?



Least significant bit

Most significant bit

H-gate: Hadamard X-gate: Negation. M: measurement (collapse)

- If BB is constant function, system will be in measured state $|11\rangle$
- If BB is variable function, system will be in measured state $|01\rangle$

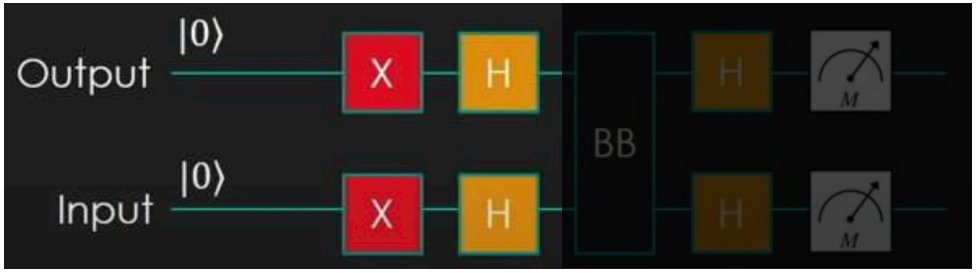
  $\Rightarrow$ quantum computer is "faster" than any classical computer

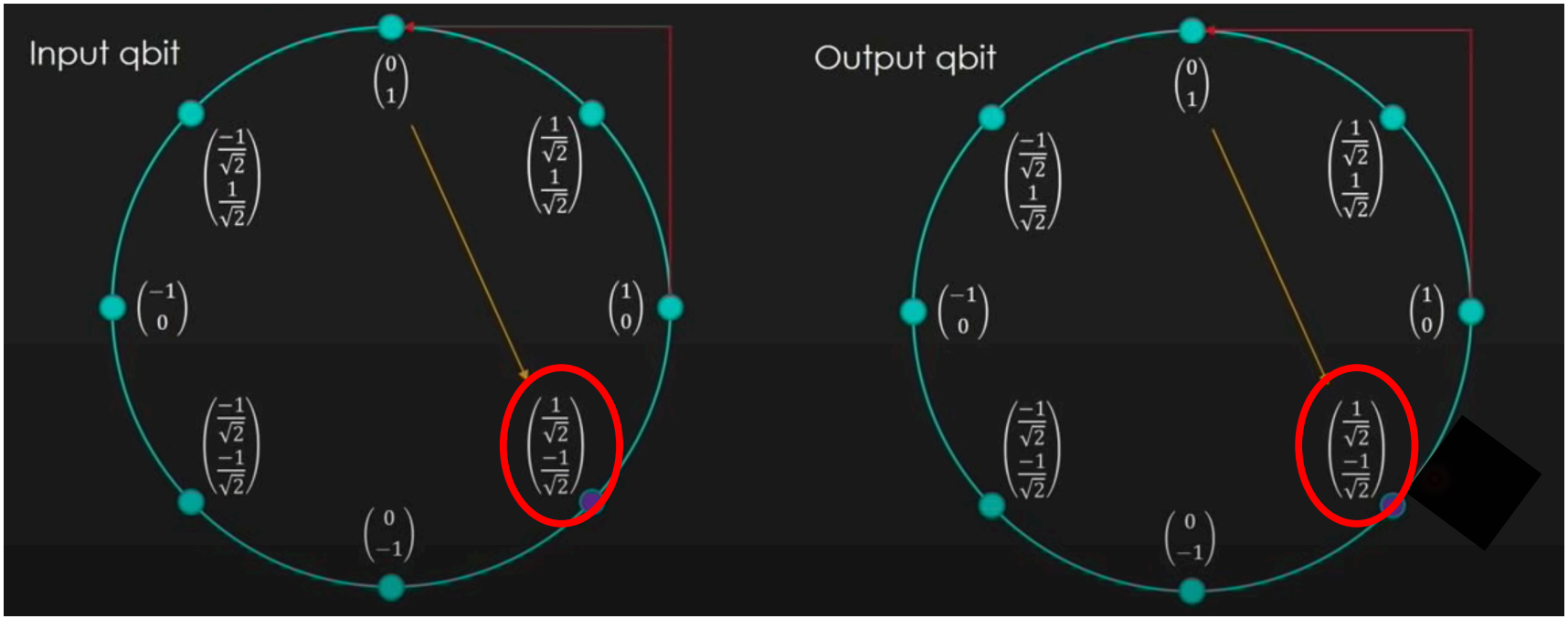
NEED TO VERIFY THIS!

The Deutsch Oracle

Can a quantum computer "beat" a classical computer?

Preprocessing

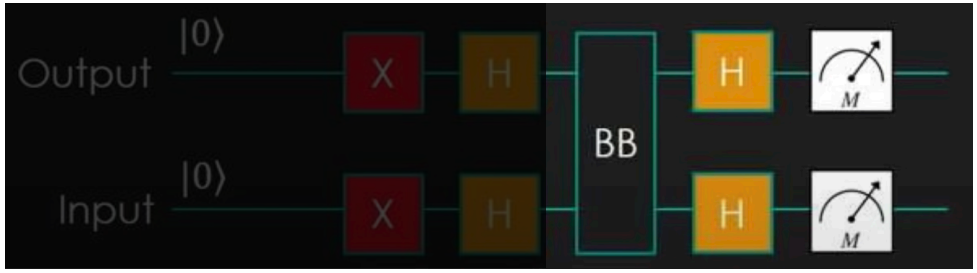




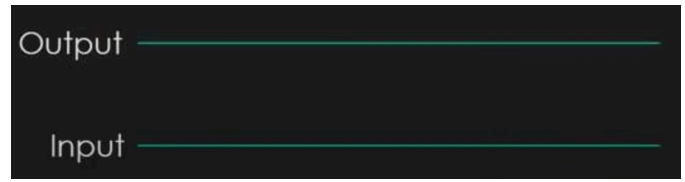
The Deutsch Oracle

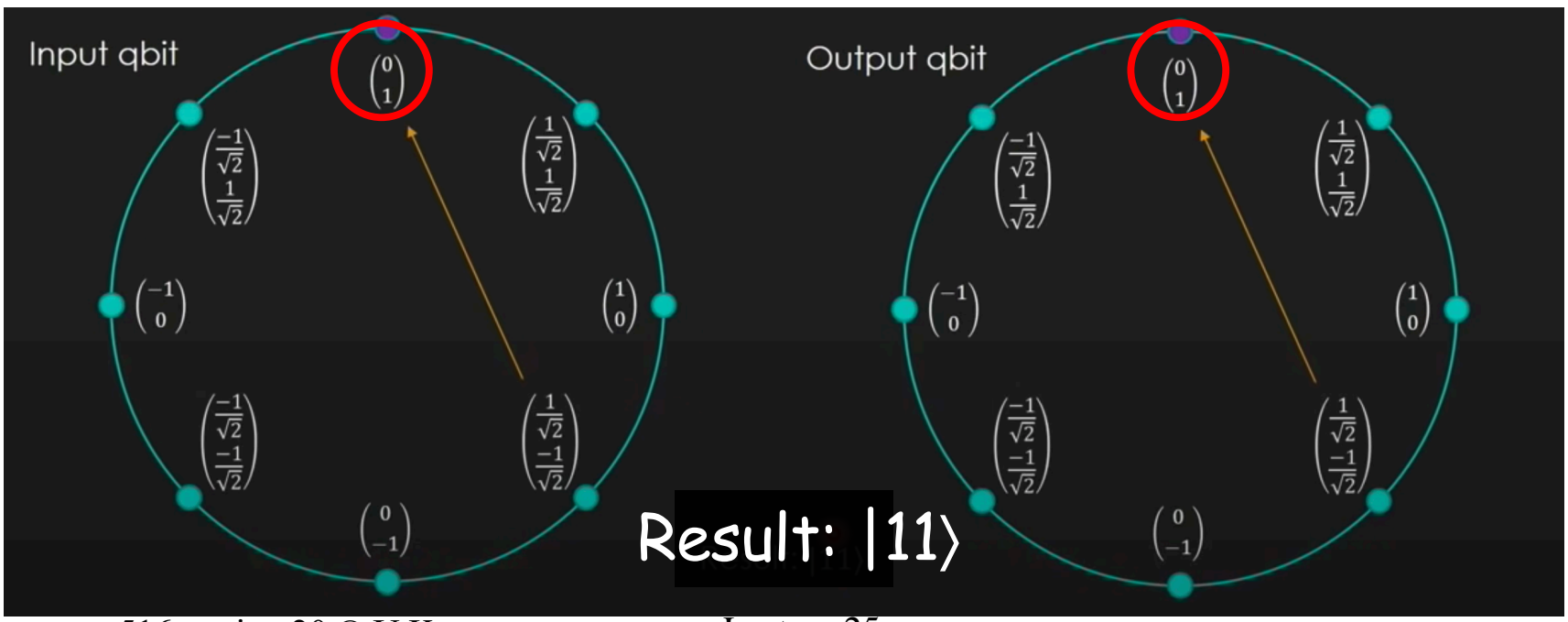
Can a quantum computer "beat" a classical computer?

Constant-0



BB implementation

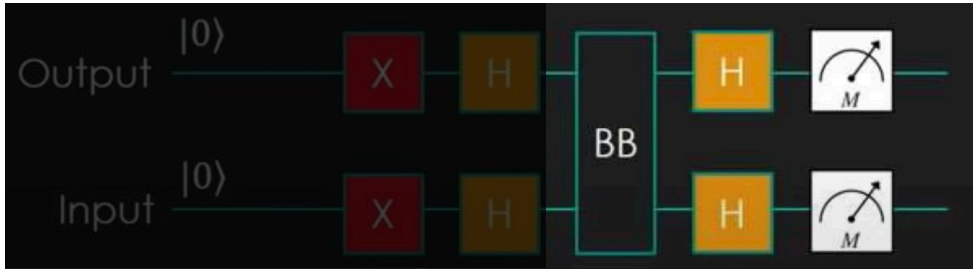




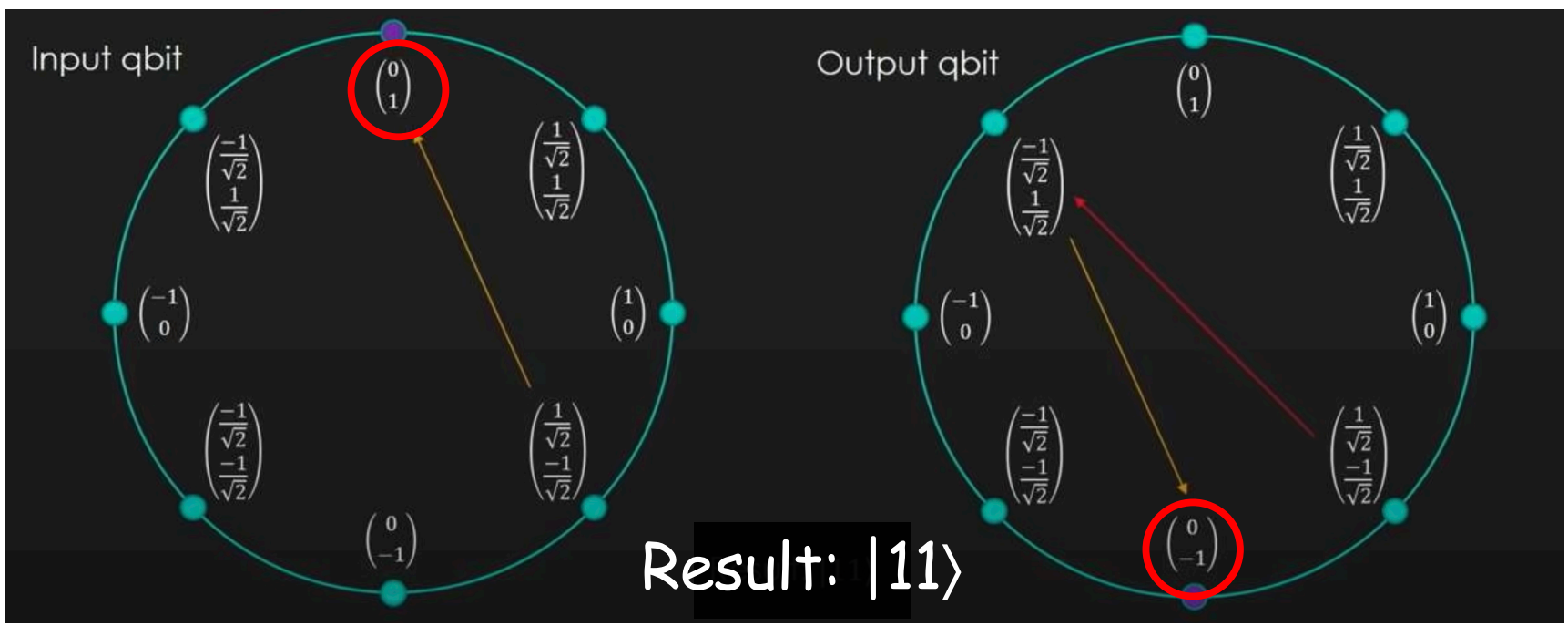
The Deutsch Oracle

Can a quantum computer "beat" a classical computer?

Constant-1



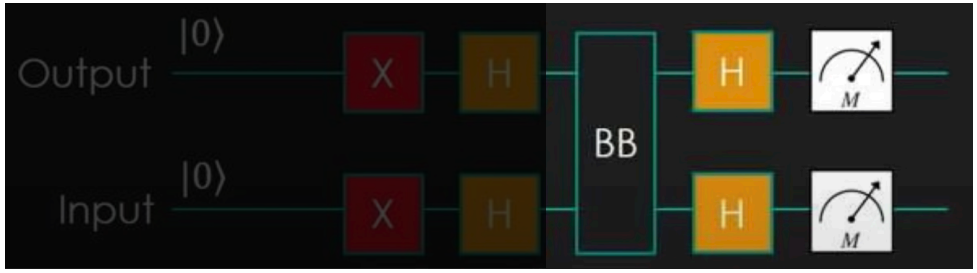
BB implementation



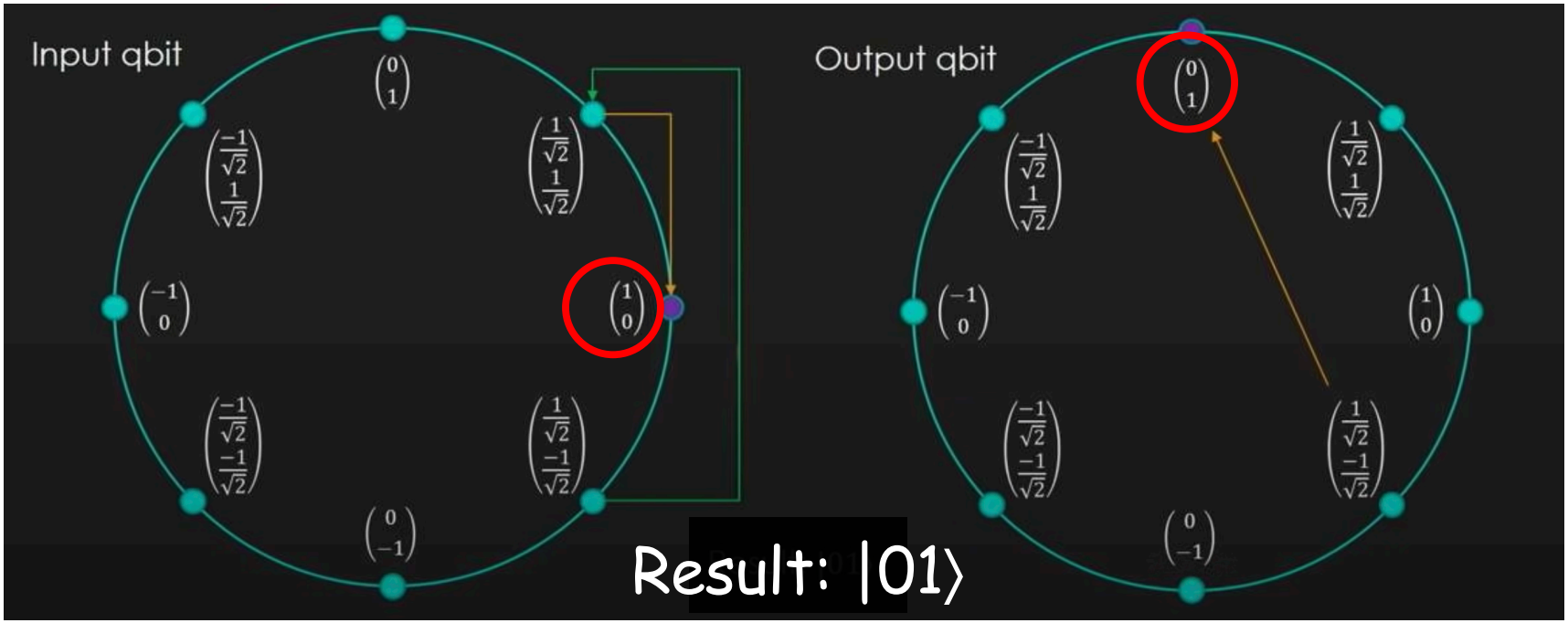
The Deutsch Oracle

Can a quantum computer "beat" a classical computer?

Identity



BB implementation



The Deutsch Oracle

Can a quantum computer "beat" a classical computer?

CNOT

BB implementation

Transition of CNOT gate is not intuitive. Rely on the math!

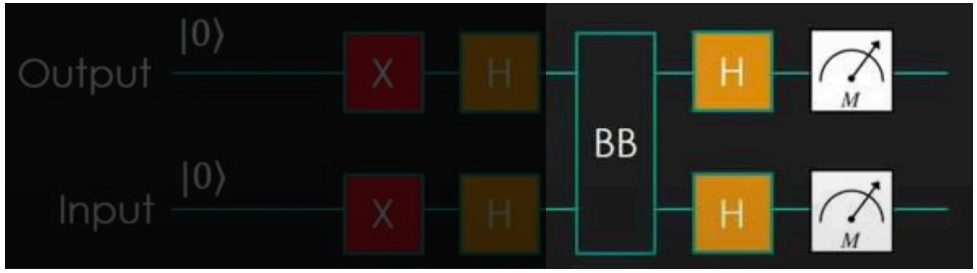
$$C\left(\begin{pmatrix}\frac{1}{\sqrt{2}}\\ \frac{-1}{\sqrt{2}}\end{pmatrix}\otimes\begin{pmatrix}\frac{1}{\sqrt{2}}\\ \frac{-1}{\sqrt{2}}\end{pmatrix}\right)=C\begin{pmatrix}\frac{1}{2}\\ \frac{-1}{2}\\ \frac{-1}{2}\\ \frac{1}{2}\end{pmatrix}=\frac{1}{2}\begin{pmatrix}1&0&0&0\\0&1&0&0\\0&0&0&1\\0&0&1&0\end{pmatrix}\begin{pmatrix}1\\-1\\-1\\1\end{pmatrix}=\frac{1}{2}\begin{pmatrix}1\\-1\\1\\-1\end{pmatrix}=\begin{pmatrix}\frac{1}{\sqrt{2}}\\ \frac{1}{\sqrt{2}}\end{pmatrix}\otimes\begin{pmatrix}\frac{1}{\sqrt{2}}\\ \frac{-1}{\sqrt{2}}\end{pmatrix}$$

"input" qbits CNOT "measured" qbits

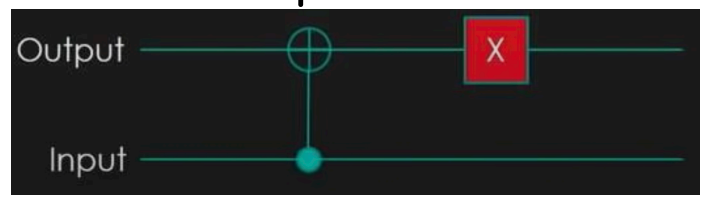
The Deutsch Oracle

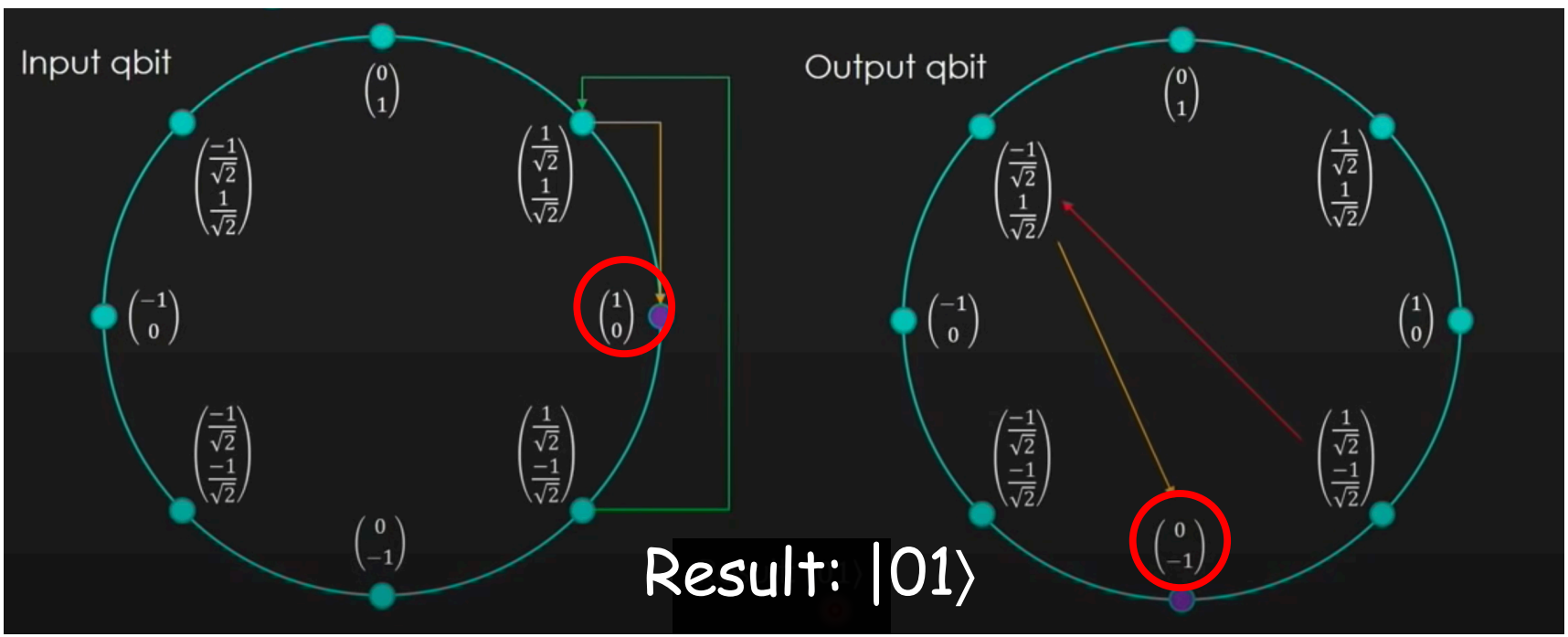
Can a quantum computer "beat" a classical computer?

Negation



BB implementation





Summary

Can a quantum computer "beat" a classical computer?

ANSWER: YES – here even in a deterministic way!

What are the practical implications of the Deutsch Oracle?

- Quantum computing can recognize patterns/properties in the input
- There is a generalization of the presented problem to an n-bit black box with two categories, constant and balanced, where balanced maps half of the inputs to 1, and half to 0 (Deutsch-Josza problem)

  Example: $n = 50 \Rightarrow$ around 10^{15} different input values
  classical computer: 10^{15} steps, since each input needs to be tested
  quantum computer: 1 single step $\Rightarrow$ exponential speedup

- Shor's algorithm also detects a pattern (periods) in the input which will allow factorization

Power of Quantum computing?

How does quantum work?

Your program describes interference patterns between qbits in superposition. Wrong answers should "cancel out", while correct answers should be "amplified".

Quantum computers are good at "period finding" in the "computed state". Think of it as "waves" (peaks and valleys), and the distance between the "waves" as the period.

My intuition (warning: I am not an expert on quantum computing!)

- Waves have non-local behaviors, while particles have local behaviors. Classical example: two-slit experiment
- Your "circuit" generates a "uniform" wave as "input" by putting qbits into superposition using H-gates and |0> input
- Formulate your problem to be solved as an interference with this input wave; think of it as an obstacle the waves have to travel around
- Measure periods in observed interference pattern as the answer

Entanglement

Two qbits are entangled if their product state cannot be factored

$$\begin{pmatrix} \frac{1}{\sqrt{2}} \\ 0 \\ 0 \\ \frac{1}{\sqrt{2}} \end{pmatrix} = \begin{pmatrix} a \\ b \end{pmatrix} \otimes \begin{pmatrix} c \\ d \end{pmatrix} \qquad \begin{aligned} ac &= \frac{1}{\sqrt{2}} \\ ad &= 0 \\ bc &= 0 \\ bd &= \frac{1}{\sqrt{2}} \end{aligned}$$

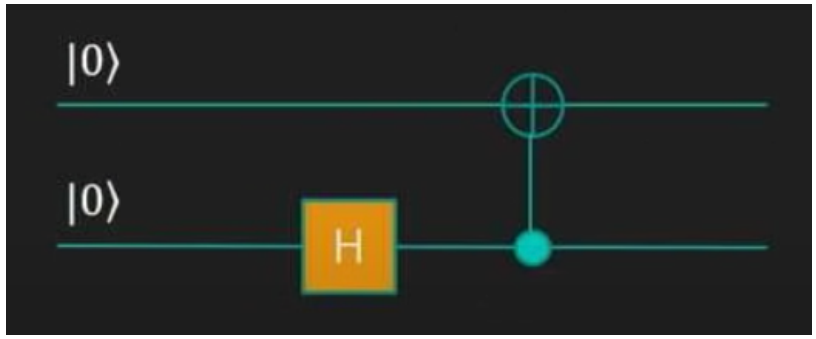
Values of qbits cannot be separated, i.e., only make sense together

- The system of equations does not have a solution, so we cannot factor this quantum state.
- This has a 50% chance of collapsing to $|00\rangle$ and 50% chance of collapsing to $|11\rangle$
- Einstein: Spooky action at a distance

Entanglement

How to entangle qbits?

This does the trick!



$$CH_1\left(\begin{pmatrix}1\\0\end{pmatrix}\otimes\begin{pmatrix}1\\0\end{pmatrix}\right)=C\left(\begin{pmatrix}\frac{1}{\sqrt{2}}\\1\\\frac{1}{\sqrt{2}}\end{pmatrix}\otimes\begin{pmatrix}1\\0\end{pmatrix}\right)=\begin{pmatrix}1&0&0&0\\0&1&0&0\\0&0&0&1\\0&0&1&0\end{pmatrix}\begin{pmatrix}\frac{1}{\sqrt{2}}\\0\\1\\\frac{1}{\sqrt{2}}\\0\end{pmatrix}=\begin{pmatrix}\frac{1}{\sqrt{2}}\\0\\0\\1\\\frac{1}{\sqrt{2}}\end{pmatrix}$$

Entanglement

- The qbits seem to be coordinating in some way
  $\Rightarrow$ measuring one qbit also collapses the other in a correlated state (spooky action at a distance)

- Coordination can happen even across large spatial distances and is instantaneous (faster than speed of light); this has been experimentally verified

- Conjecture: qbits "decided" at the time of entanglement to what value to collapse to ("hidden variable" theory); has been disproved by John Bell in 1964

- Note: Coordination is faster than light; however, no information can be communicated (Einstein's theory of relativity still works!)

IBM Quantum experience

Let's use IBM's Q machine to write a Quantum program to find out how / whether entanglement works.

Create free IBM quantum account (educational)
https://www.ibm.com/quantum-computing/technology/experience

This will give you access to different tools to write programs

- Circuit composer
- OpenQasm

These programs can be

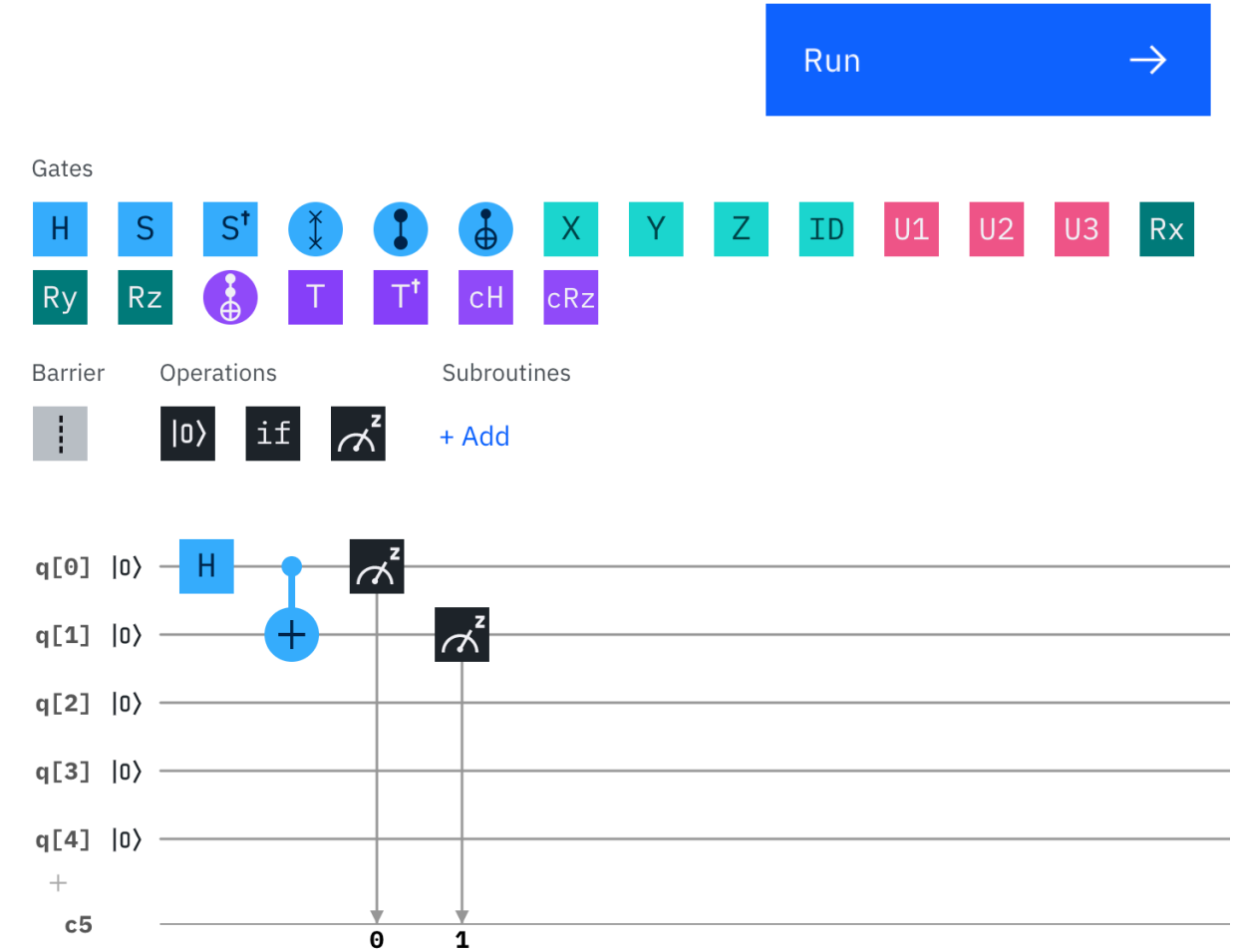
- Simulated
- Executed on an actual quantum computer

IBM Quantum experience

Let's use IBM's Q machine to write a Quantum program to find out how / whether entanglement works.

Step 1: Drag/drop gates to write your circuit

Step 2: Select run
- simulation
- actual machine

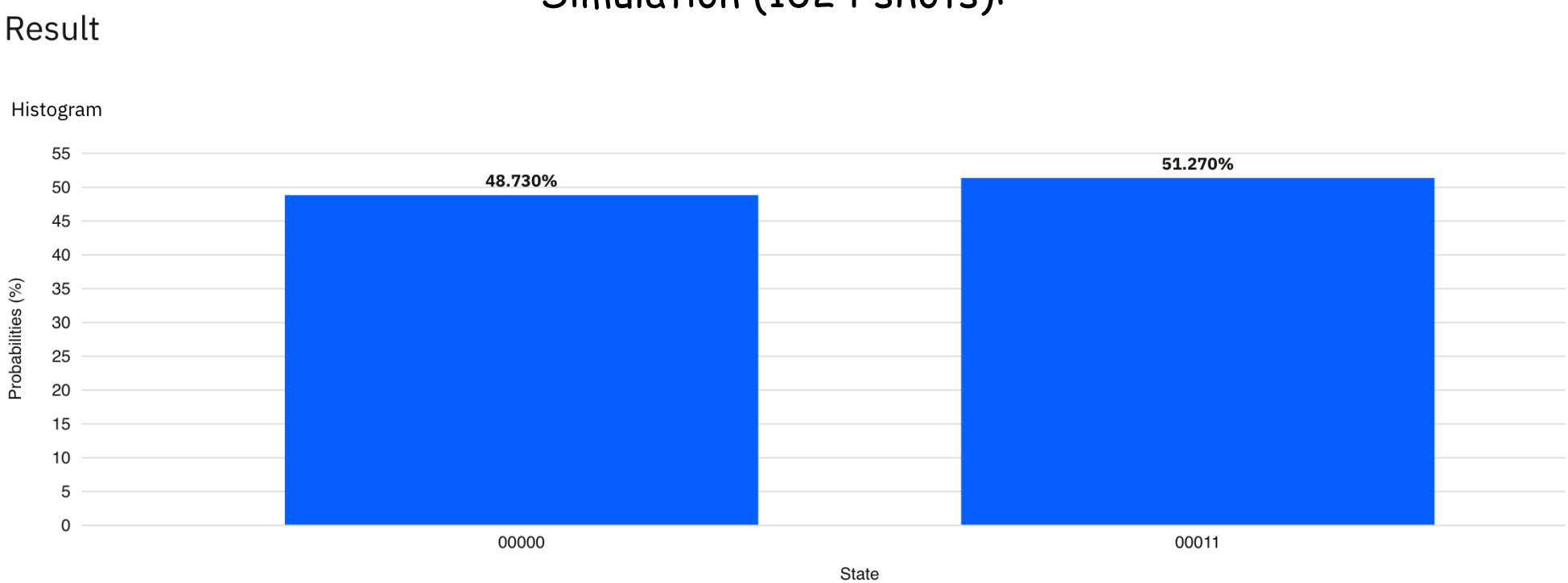


IBM Quantum experience

Let's use IBM's Q machine to write a Quantum program to find out how / whether entanglement works.

Shots specify the number of experiments measured.

Simulation (1024 shots):

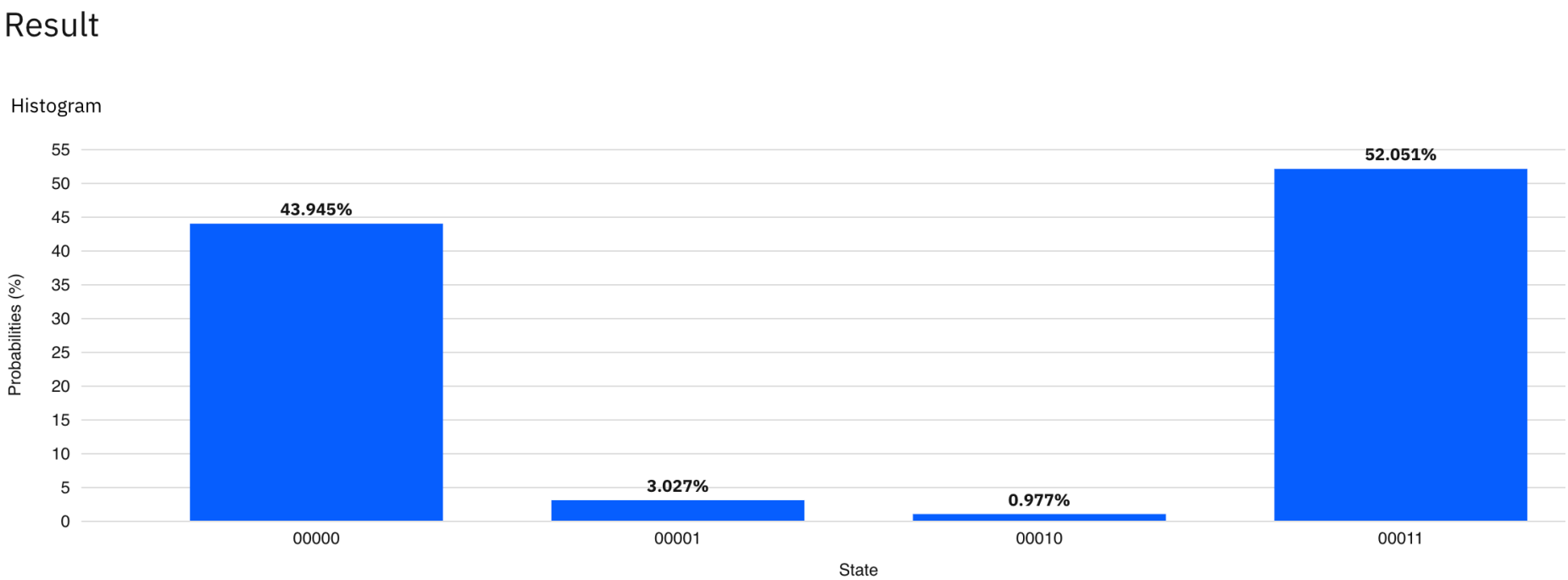


IBM Quantum experience

Let's use IBM's Q machine to write a Quantum program to find out how / whether entanglement works.

Shots specify the number of experiments measured.

IBMqx2 machine (1024 shots):



Technical Challenges / Results

- **Decoherence problem**: Hard to keep qbits in a superposition state, or avoid interacting with their environment in a undesirable fashion (currently in micro-second range) [fault tolerant quantum computing]

- 1-qbit or 2-qbit gates (i.e. local operations) are sufficient for universal computing [Barenco et al. 1995]

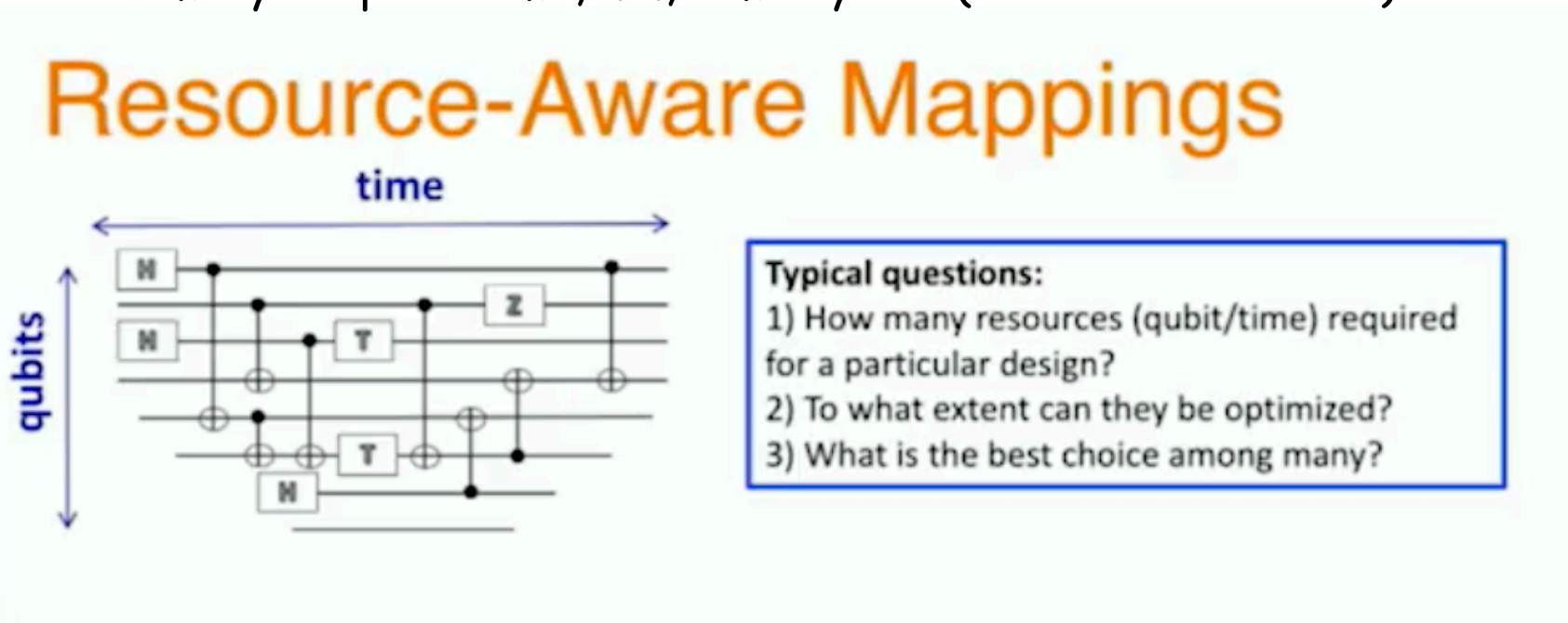
- What are the implications for programming tools?
  - Programming abstractions (languages)
  - Debuggers
  - Simulation environments

How to write Quantum programs?

Two resource dimensions: #qbits and time

Circuit width (top-to-bottom):
- how many qbits

Circuit depth (left-to-right):
- how many steps in time, i.e., time cycles (decoherence issue)



Algorithm choice will depend on resource availabilities.
How you compile may actually change the answer due to robustness issues.

How to write Quantum programs?

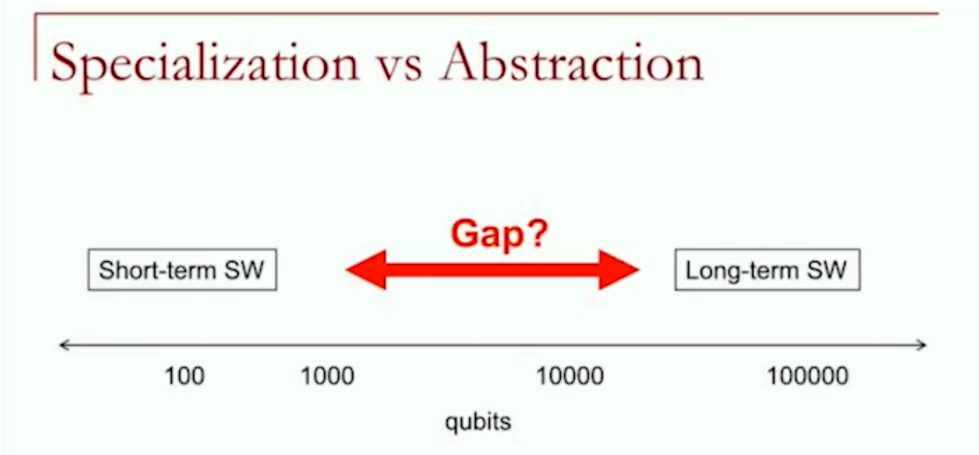
EPiQC: Enabling Practical-scale Quantum Computation

https://www.epiqc.cs.uchicago.edu/

Practical consideration

Margaret Martonosi: Error correction is about 10x to 50x overhead (you need that many more qbits – so need to work with noisy computation)

Best way right now is to look at specialization rather than general strategies due to limited quantum volume:



Compiler challenges

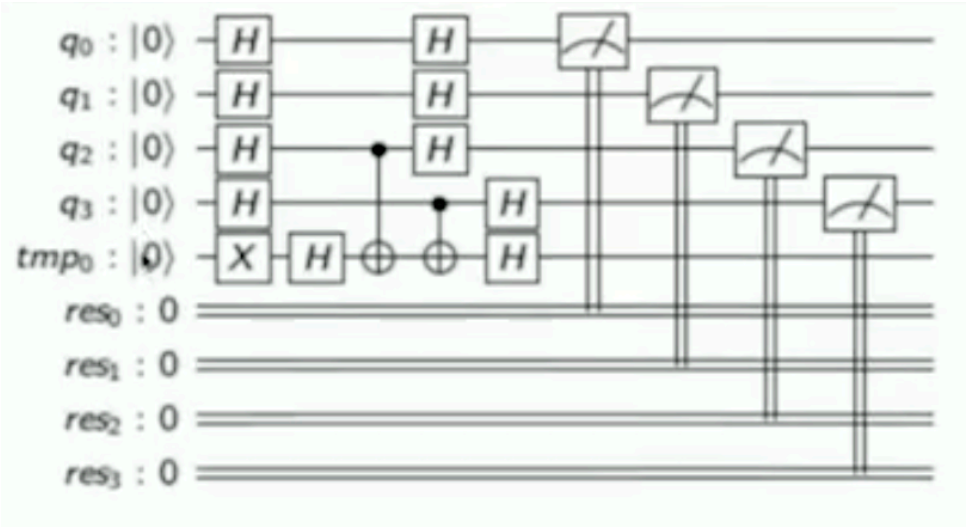
Actual device implementations may have connectivity restrictions;
Compiler must satisfy a machine's qbit connectivity constraints (connectivity map):

Q_a => Q_b means that Q_a may be the CNOT control bit for target qbit Q_b

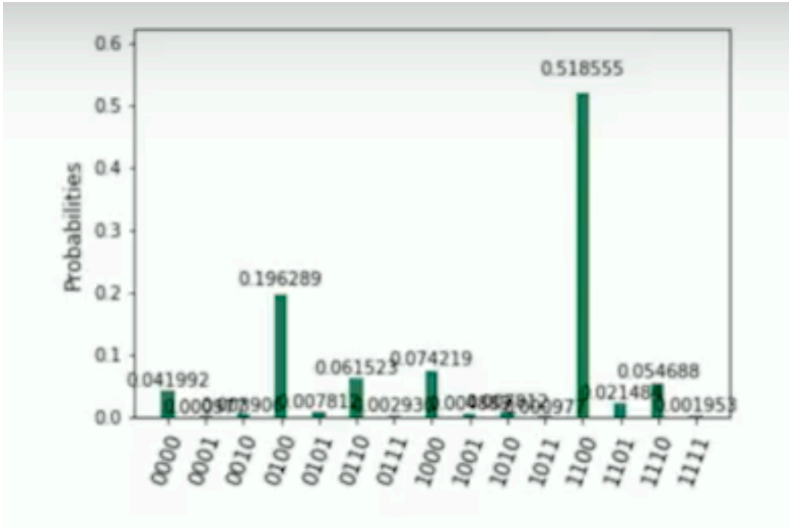
This is a challenging optimization problem. In addition, different mappings may have different circuit depth and width, and/or different numbers of CNOT gates which are rather noisy. Therefore, there is a significant robustness issue here.

How to write Quantum programs?

Results of a Quantum computation

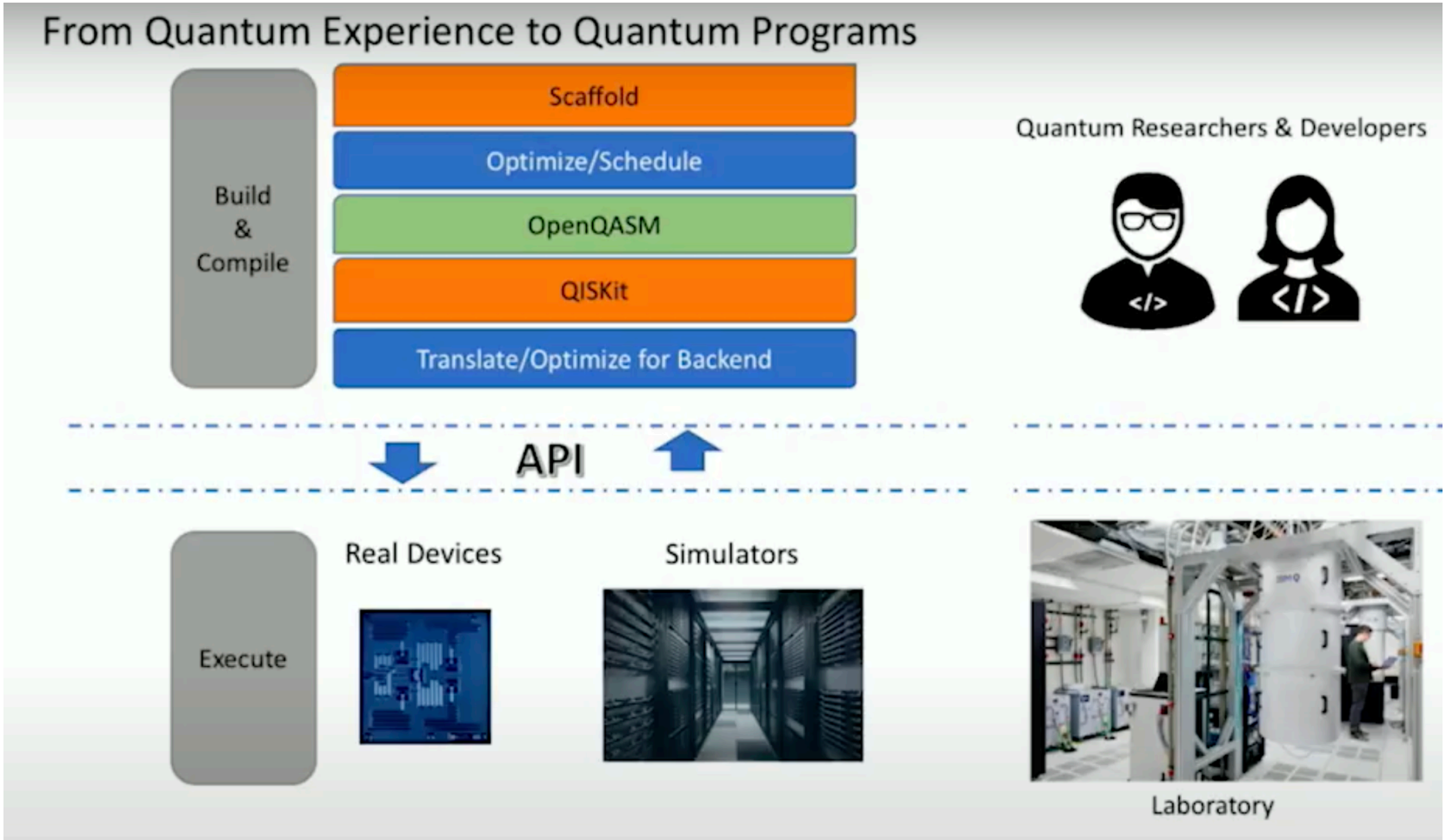


Answers are always probabilistic. Therefore, need to build histogram over 100s or 1000s of runs.



How to write Quantum programs?

IBM Q quantum experience



How to write Quantum programs?

Promising (Near) Future Development: Get the foot in the door, i.e., show that Quantum Computing is practical
→ will allow hands-on experimentation and new designs that nobody has thought of

Quantum Approximate Optimization Algorithms (QAOA)

- Rely on smaller systems (10s of qbits) and shallow depth
- Try to solve existing optimization problems approximately
- Seems promising from a theoretical computer science point of view

Specialized software/hardware

- Instead of going for general purpose strategies (languages / compilers), try to exploit practical structures and devices as much as possible
- Somewhat similar to accelerator designs and FPGA compilation where many parameters are known at compile time

How to write Quantum programs?

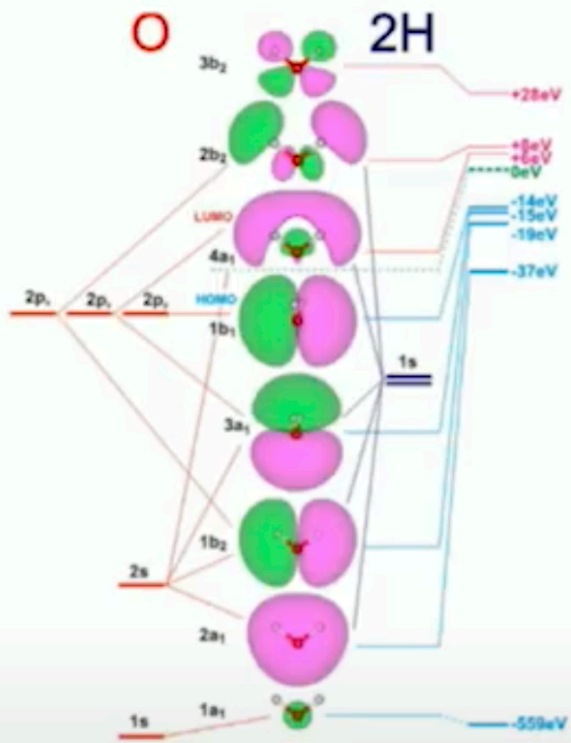
Example: Quantum programs to solve chemistry problems

Issue: Current methods are not predictive, i.e., can analyze existing structures but not constructively come up with new ones

Key question: Where should the electrons go? This will determine some desired property of the molecule.

Problem: There are many possible states / configurations a water molecule can be in

- Water H_2O
- 2+6+2 electrons
- How many addresses?
  - **Simplest model: 14**
  - 3003 states
  - **cc-pVTZ model: 116**
  - 8×10^{23} states



Resources

NSF Expedition in Computing
EPiQC: Enabling Practical-scale Quantum Computation
https://www.epiqc.cs.uchicago.edu/

Research Challenges in Quantum Computing (Fred Chong, Tutorial @ISCA 2018)
https://www.youtube.com/watch?v=ZQ_NE-E91aU&list=PLfOgkuiMs5qD6BkS7Bk2gLGdHSxjMz5MY

IBM Quantum Experience
- create a free IBM account (educational)

https://www.ibm.com/quantum-computing/technology/experience

Quantum Computing architecture and tool chains
- Margaret Martonosi (Princeton)
  https://quantumarchitectureprinceton.github.io/
- Yipeng Huang (soon Rutgers)
  https://yipenghuang.com/