Intro to Quantum Computing at CARC

Matthew Fricke

Version 0.1

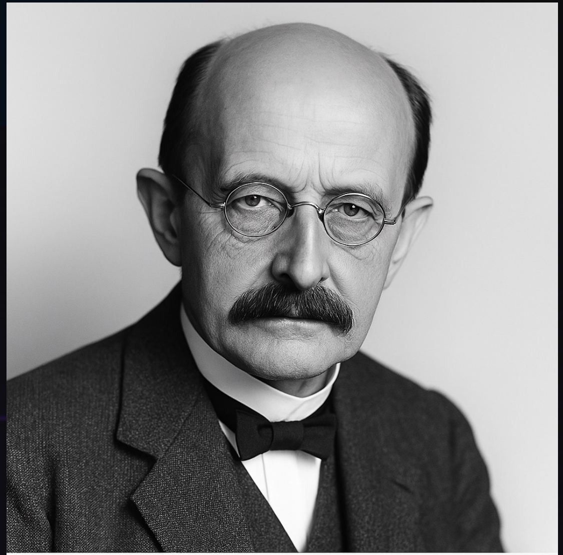
Welcome to the Quantum Frontier

GOALS:

1) Quantum Computation isn’t magic
2) Give a sense of the ideas behind it
3) Show you how to run quantum programs

Not to turn you into a quantum physicist in 2 hours

There will be a variety of levels of presentation – everyone will get lost at some point.



MAX PLANCK

1858–1947

University of Berlin

$E = h\nu$

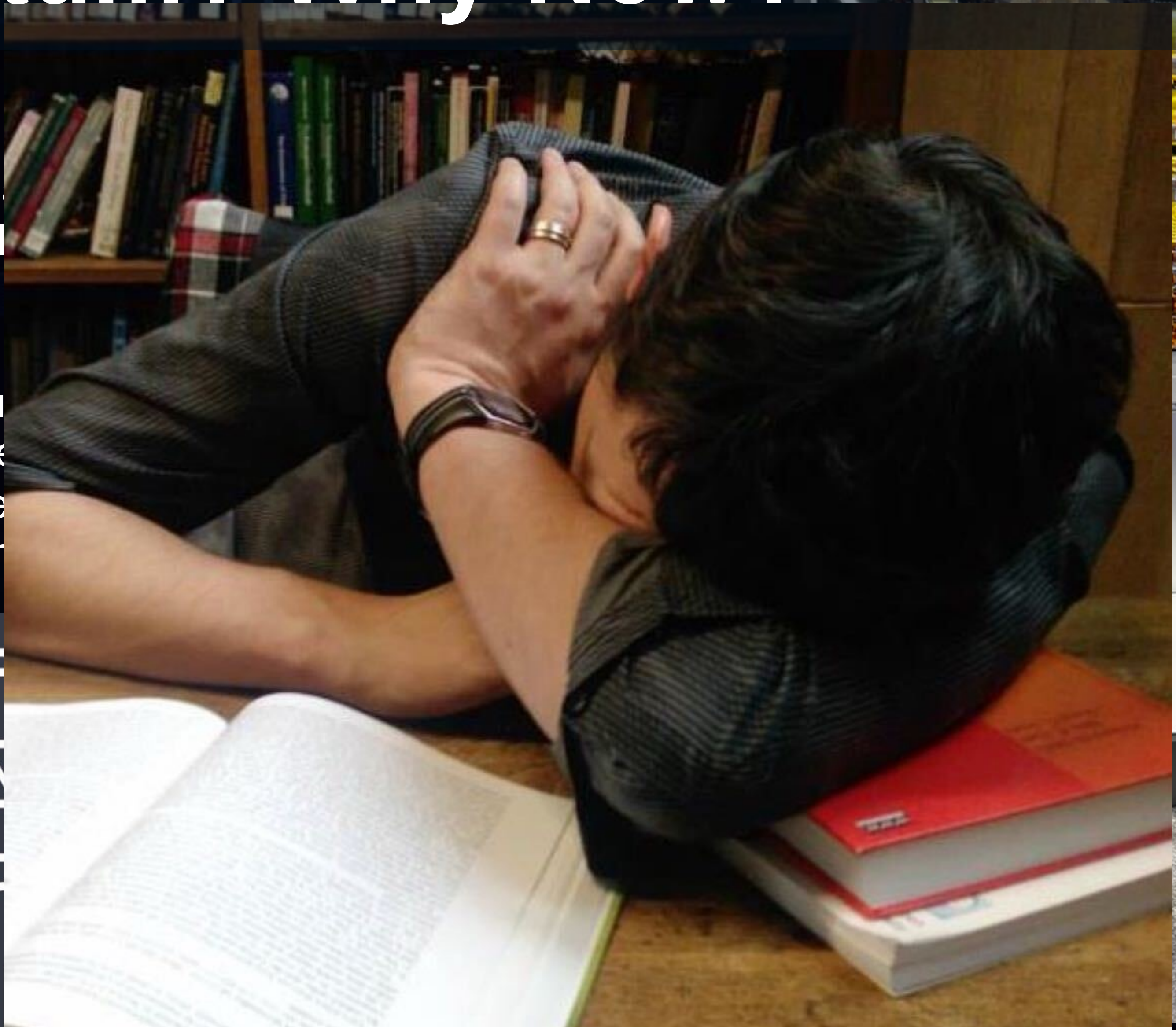
German

Why Quantum? Why Now?

- **Classical limits**
- • Moore’s Law pl
- • Simulation bottl

- **Quantum opportu**
- • Simulating mole
- • Solving hard se
- • Hybrid algorithn

Qis



“Learning” quantum computing in 2012

Why Quantum? Why Now?

•Classical limits

- • Moore’s Law plateau
- • Simulation bottlenecks

•Quantum opportunities

- • Simulating molecules
- • Solving hard search problems
- • Hybrid algorithms for near-term utility

Qiskit

Logging into Hopper

First login to the Linux **workstation** in front of you. Your CARC username is on the sign in sheet.

If you have logged in before use your existing password

This is an “important step” so don’t let me move on until you have logged in

Logging into Hopper

```
ssh vanilla@hopper.alliance.unm.edu
```

Should prompt you for a password…

Don’t let me move on until you are able to login.

Replace vanilla with your name (unless your last name is Ice)

```
[vanilla@hopper ~]$ git clone https://lobogit.unm.edu/CARC/workshops.git
Cloning into 'workshops'...
remote: Enumerating objects: 132, done.
remote: Counting objects: 100% (75/75), done.
remote: Compressing objects: 100% (43/43), done.
remote: Total 132 (delta 33), reused 74 (delta 32), pack-reused 57
Receiving objects: 100% (132/132), 57.58 KiB | 3.60 MiB/s, done.
Resolving deltas: 100% (51/51), done.
```

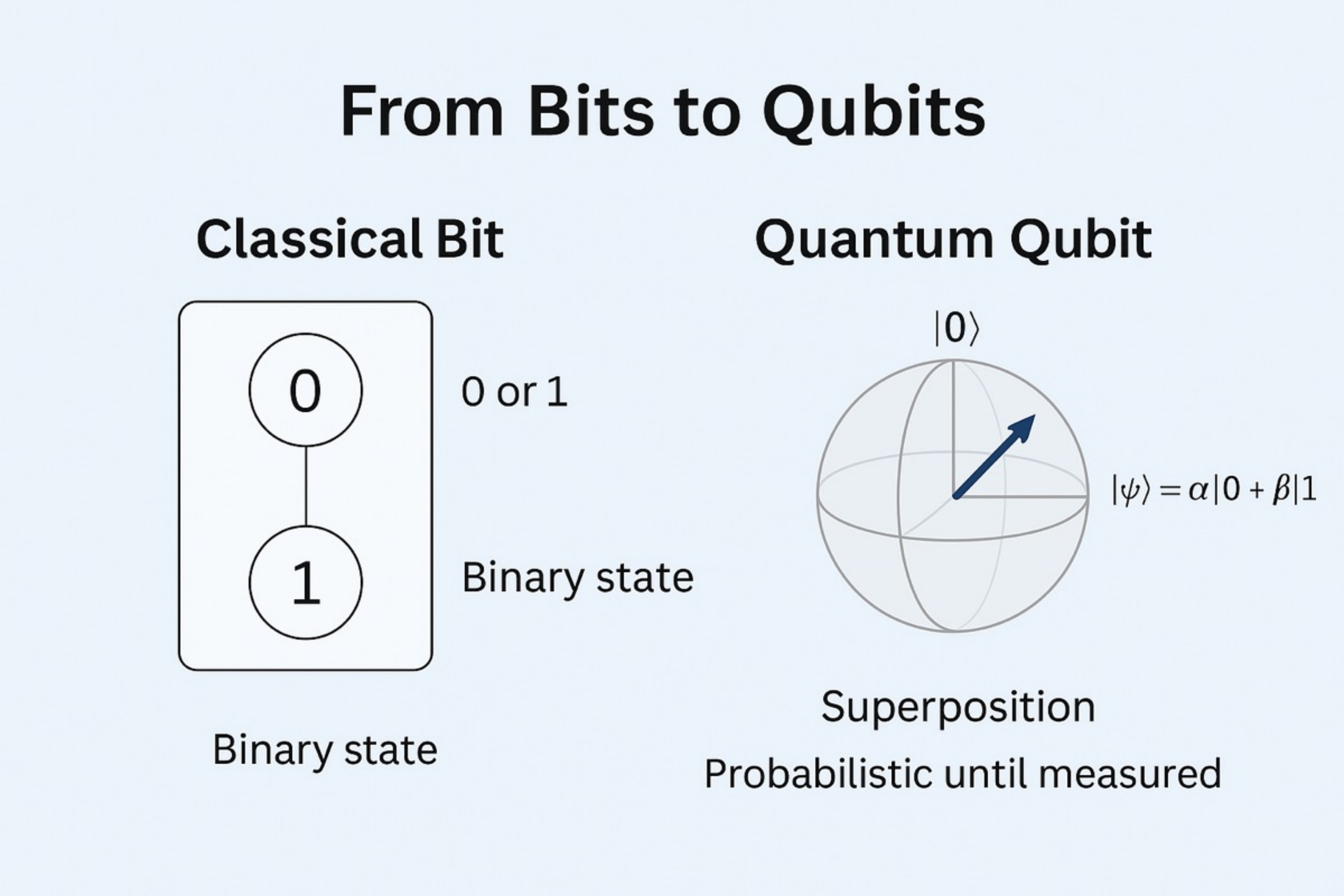
Rather than make you write shell scripts lets just download some we wrote for this workshop…

```
[vanilla@hopper workshops]$ git stash
[vanilla@hopper workshops]$ git pull
```

```
[vanilla@hopper ~]$ cd workshops
[vanilla@hopper workshops]$ module load miniconda3
[vanilla@hopper workshops]$
conda env create --file quantum_computing/conda_envs/quantum-computing.yml

Type “yes” to installing when prompted
```

Let’s setup the quantum packages we need with conda

From Bits to Qubits



From Bits to Qubits

Classical Bit

Quantum Qubit

Binary state

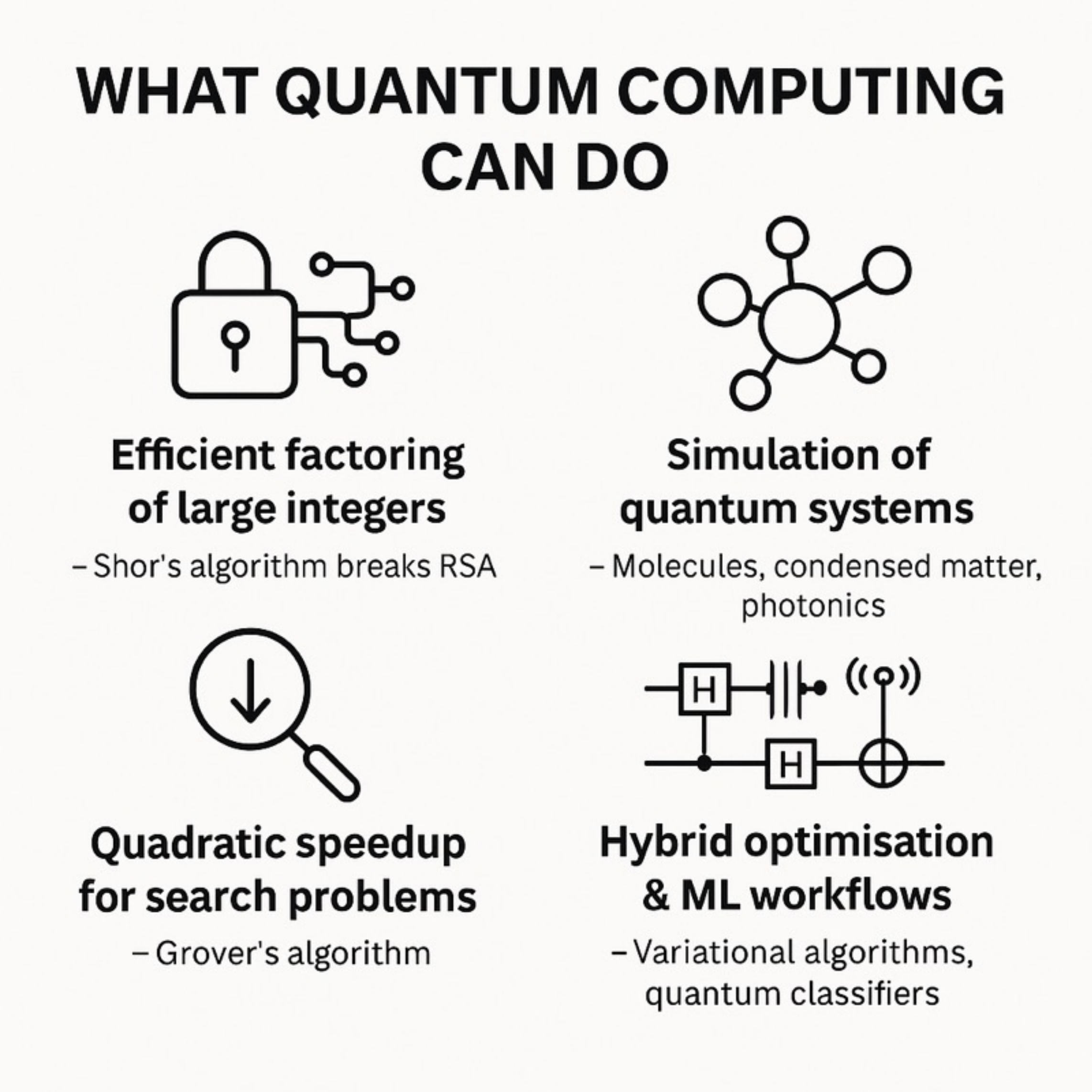
Superposition

Probabilistic until measured

Supremacy vs Advantage vs Utility: What’s the Difference?

Term	Definition	Goal	Usefulness	Example
Quantum Supremacy	A quantum computer solves **at least one** problem faster than any classical computer, regardless of practicality.	Prove quantum capabilities	❌ The problem may be artificial	Google’s Sycamore chip (random circuit sampling)
Quantum Advantage	A quantum system outperforms classical approaches on a specific, structured problem.	Demonstrate task-specific performance	⚠️ Possibly useful	Early VQE on small molecules
Quantum Utility	A quantum device solves a **real-world** problem better than classical methods, justifying its use.	Provide practical value	✅ Clearly useful	🤷‍♂️ As of today: None

What Quantum Computing Can Do



WHAT QUANTUM COMPUTING CAN DO

Efficient factoring of large integers

– Shor's algorithm breaks RSA

Simulation of quantum systems

– Molecules, condensed matter, photonics

Quadratic speedup for search problems

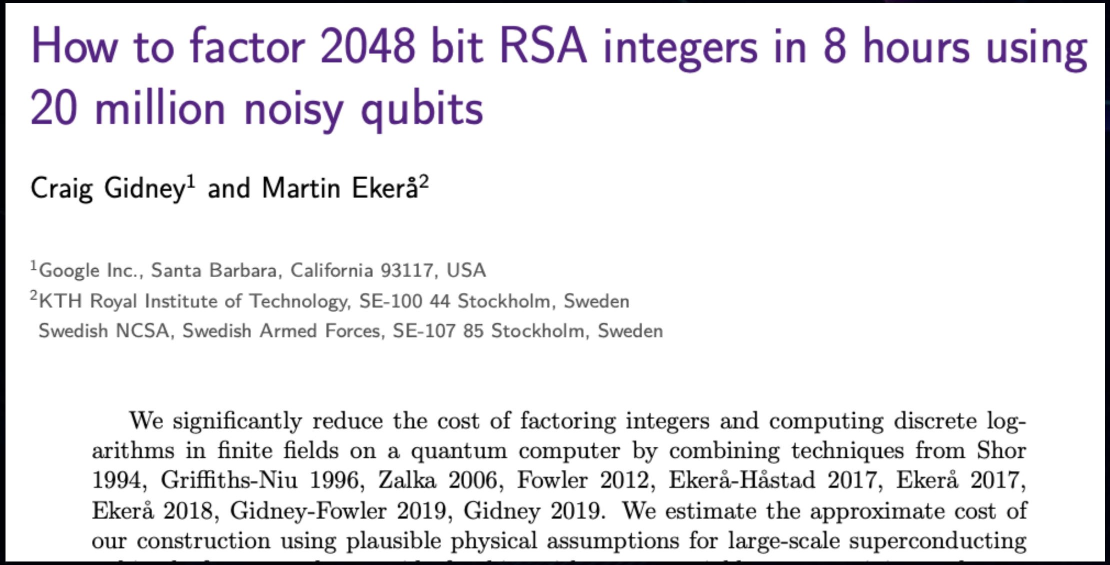
– Grover's algorithm

Hybrid optimisation & ML workflows

– Variational algorithms, quantum classifiers

What Quantum Computing Can't Do Faster

Limitation	Notes
✗ **Not faster at everything**	No speedup for all problems—only specific types
✗ **Not yet fault-tolerant**	NISQ era = noisy hardware, limited depth
✗ **Not scalable (yet)**	Current devices = tens to hundreds of noisy qubits*
✗ **Not a replacement for classical computing**	Best as a **co-processor**, not a universal upgrade



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

Craig Gidney[1] and Martin Ekerå[2]

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

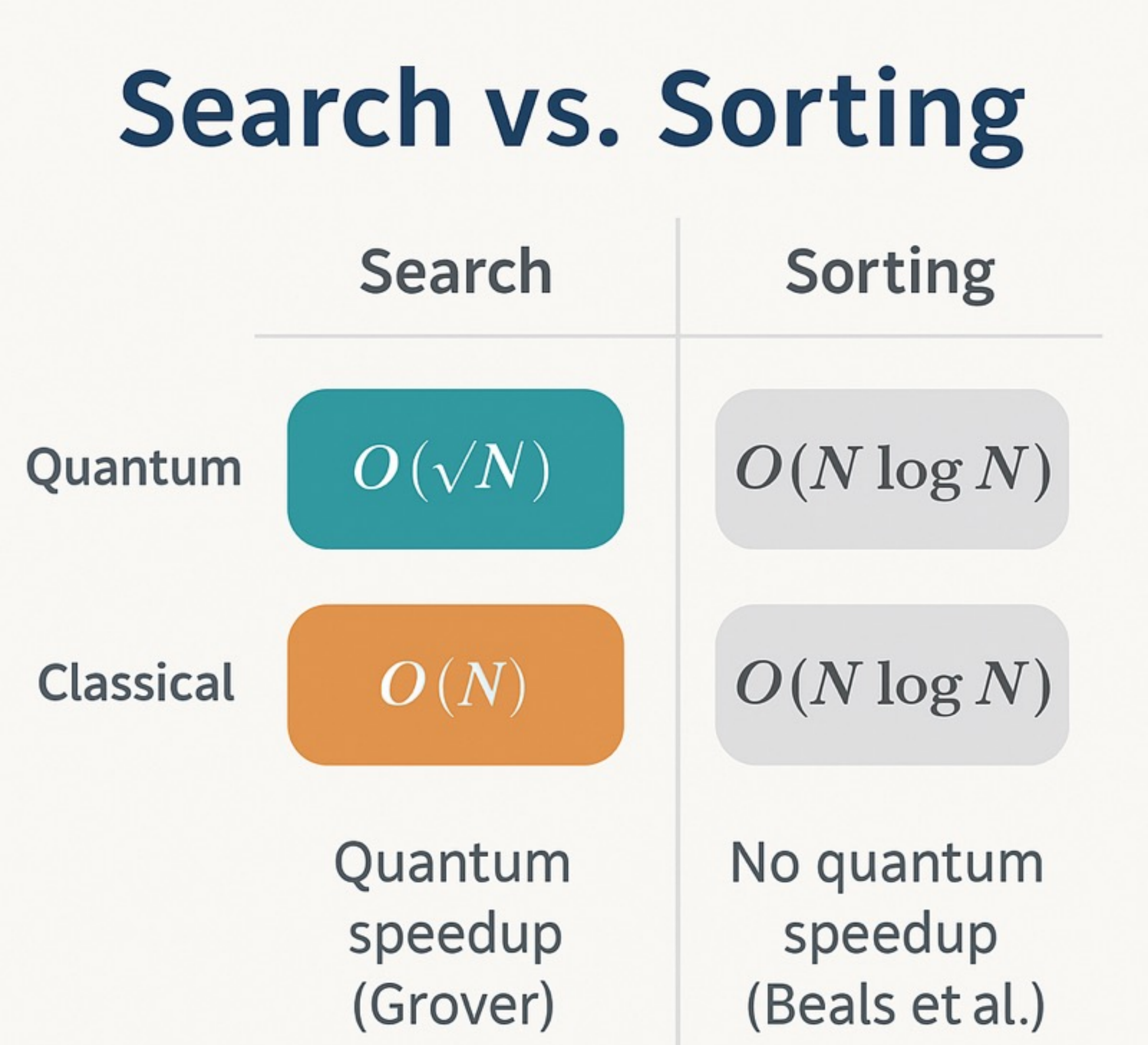
We significantly reduce the cost of factoring integers and computing discrete logarithms in finite fields on a quantum computer by combining techniques from Shor 1994, Griffiths-Niu 1996, Zalka 2006, Fowler 2012, Ekerå-Håstad 2017, Ekerå 2017, Ekerå 2018, Gidney-Fowler 2019, Gidney 2019. We estimate the approximate cost of our construction using plausible physical assumptions for large-scale superconducting

Craig Gidney and **Martin Ekerå**. *How to factor 2048 bit RSA integers in 8 hours using 20 million noisy qubits.* **Quantum**, **5**, 433 (2021). https://doi.org/10.22331/q-2021-04-15-433

Not faster(and adds overhead) for Sorting, matrix multiplication, numerical integration.

What Quantum Computing Can’t Do Faster

Beals, R., Buhrman, H., Cleve, R., Mosca, M., & de Wolf, R. (2001). **Quantum lower bounds by polynomials.** *Journal of the ACM*, 48(4), 778–797. https://doi.org/10.1145/502090.502097



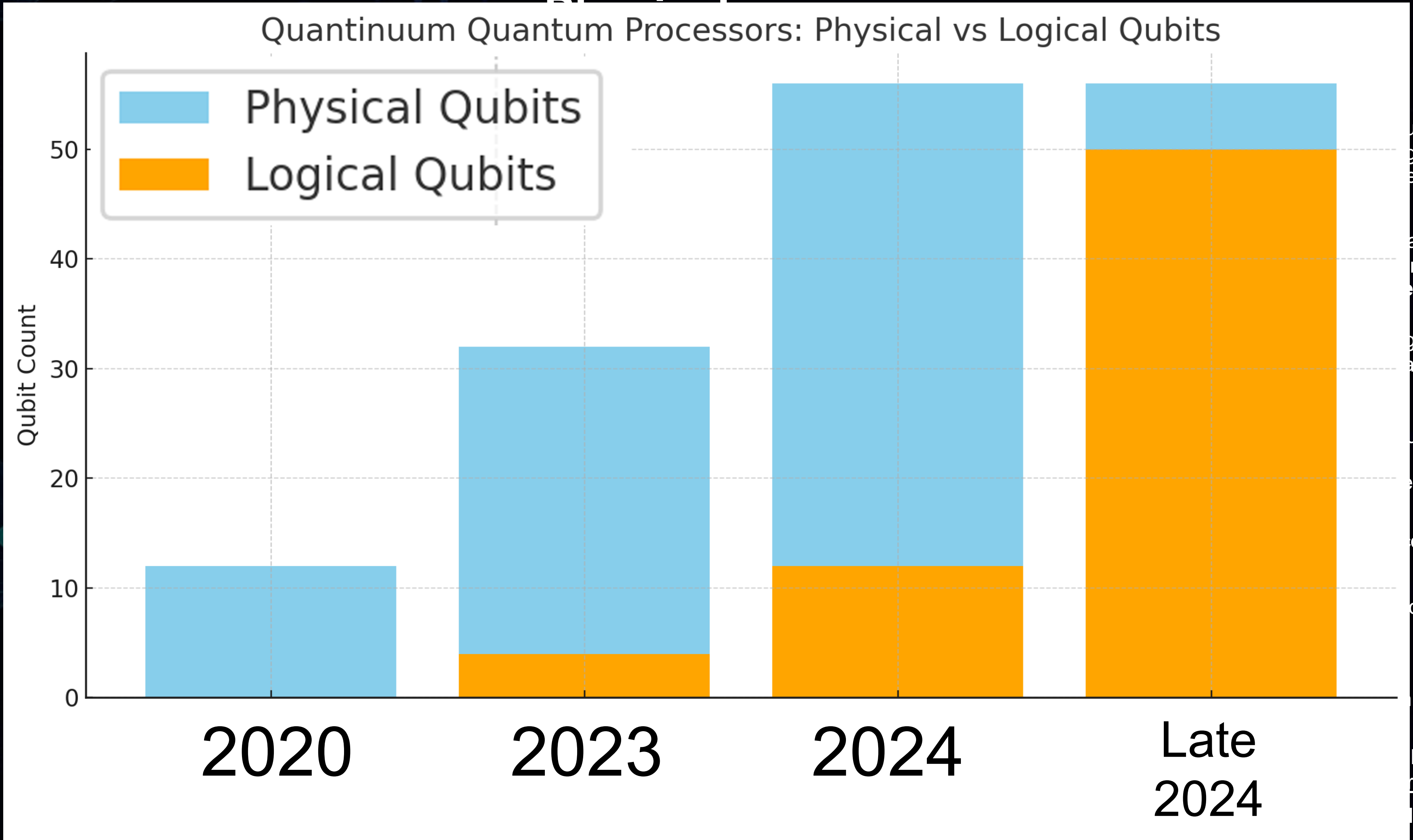
Search vs. Sorting

	Search	Sorting
Quantum	$O(\sqrt{N})$	$O(N \log N)$
Classical	$O(N)$	$O(N \log N)$
	Quantum speedup (Grover)	No quantum speedup (Beals et al.)

Hardware improvement in Qubits

Year	Name	Physical Qubits	Logical Qubits	Notes
1999	D-Wave founded	**N/A**	N/A	First company to pursue commercial quantum computing via **quantum annealing**, not gate-based methods
2011	D-Wave One	**~128**	N/A	First commercially available **quantum annealer**; **not a general-purpose quantum computer**
2015	D-Wave 2X	**~1,000**	N/A	Quantum annealing system; not capable of universal quantum algorithms
2017	D-Wave 2000Q	**~2,048**	N/A	Expanded qubit count; annealing only
2020	Quantinuum H1-1	**12**	N/A	Gate-based, trapped-ion quantum computer
2021	IBM Eagle	**127**	N/A	Gate-based superconducting processor
2022	IBM Osprey	**433**	N/A	Increased scale and coherence
2023	IBM Condor	**1,121**	N/A	First IBM chip over 1,000 qubits
2023	Quantinuum H2	**32**	**4**	First demonstration of 4 logical qubits (source)
2024	Quantinuum H2-1	**56**	**12**	12 logical qubits in hybrid workflow with Microsoft (source)
2024	Quantinuum H2-1	**56**	**50**	Achieved 50 logical qubits with >98% fidelity (source)
2025	D-Wave Advantage	**5,000+**	N/A	Latest-generation quantum annealer; **still not gate-based**

Hardware improvement in Qubits


Quantinuum Quantum Processors: Physical vs Logical Qubits

Year				
1999				ursue commercial g via **quantum** e-based methods
2011				available **quantum** **neral-purpose** **er**
2015				g system; not al quantum
2017				unt; annealing only
2020				ed-ion quantum
2021				onducting
2022				d coherence
2023				1,000 qubits
2023				of 4 logical qubits
2024				hybrid workflow rce)
2024				qubits with >98% fidelity (source)
2025	D-Wave Advantage	**5,000+**	N/A	Latest-generation quantum annealer; **still not gate-based**

Hardware improvement in Qubits

Year

1999

2011

2015

2017

2020

2021

2022

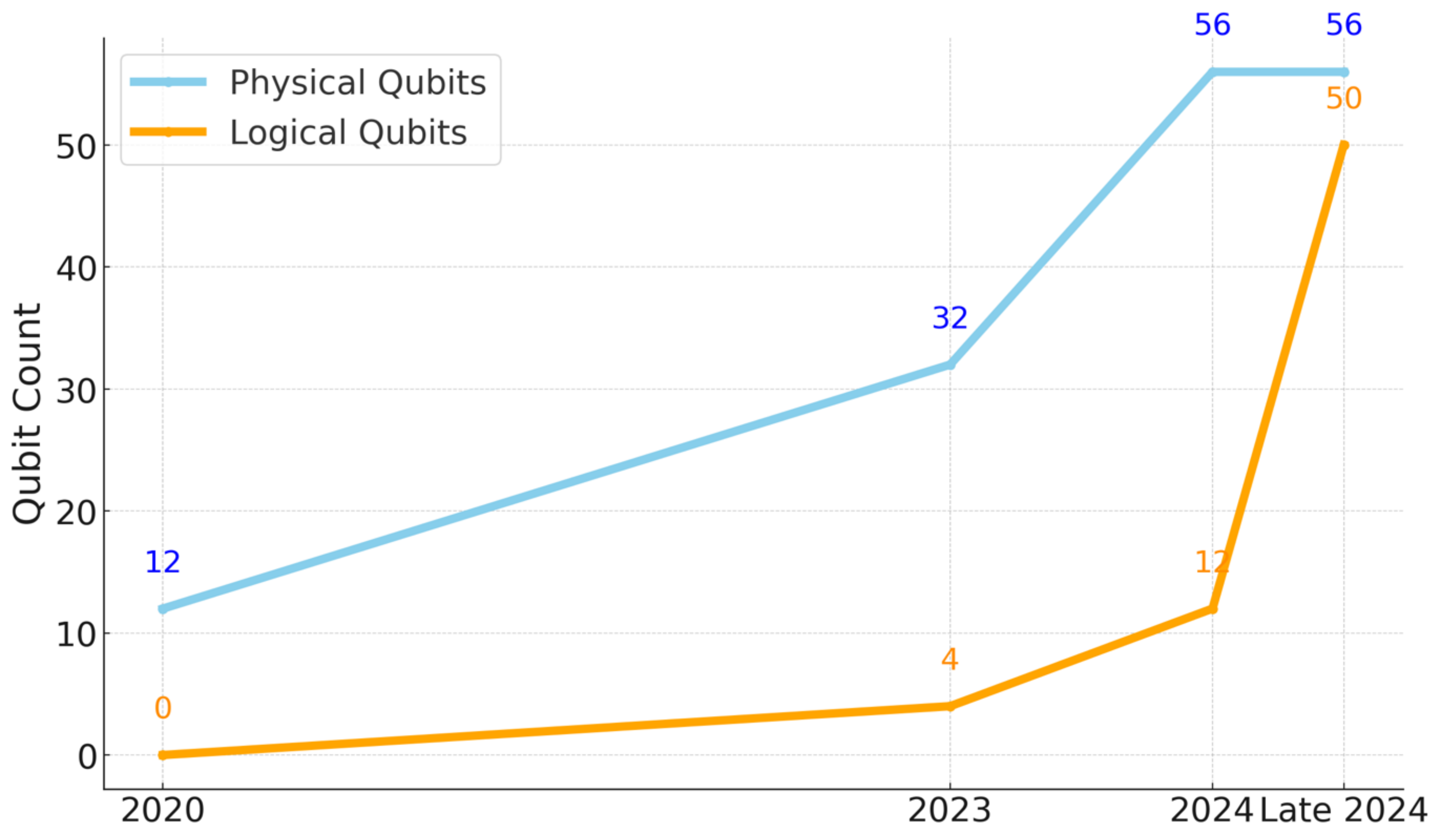
2023

2023

2024

2024

2025 D-Wave Advantage 5,000+ N/A



mmercial
antum
methods

e quantum
urpose

; not
um

ealing only

uantum

ng

ence

ubits

ical qubits

orkflow

with >98%

annealer;
still not gate-based

Why HPC Cares About Quantum and Quantum Needs HPC

Category	High Performance Computing (HPC)	Quantum Computing (QC)
💰 Cost	~$1–10M (e.g. Easley: $1.5M)*	~$100M+ per system
🛰️ Access	Remote job scheduler	Remote job scheduler
👥 Users	Shared systems managed by scheduler	Shared systems managed by scheduler
⚙️ Hardware	Large remote scarce systems	Large remote scarce systems
🧑‍🏫 Training	Needed	Needed
🧪 Use Today	Primarily researchers	Still experimental
🔁 Link	Runs quantum simulators	HPC for simulations

*doesn’t fit on a desk

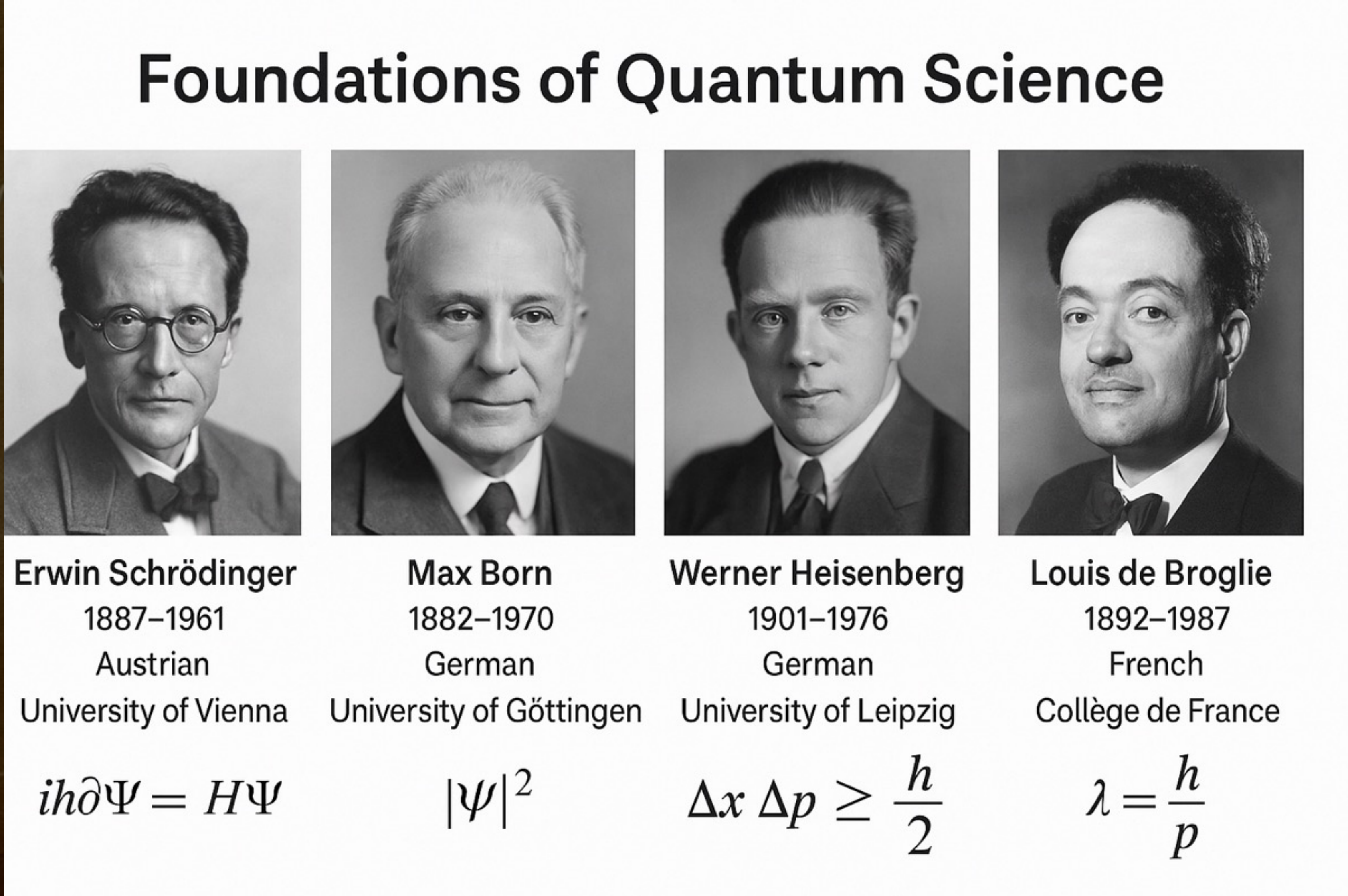
Both HPC and Quantum are expensive, specialised, and require expert support. Until large fault-tolerant QCs exist, HPC is where most quantum computing algorithms actually run.

Classical vs Quantum Thinking

Classical Thinking	**Quantum Thinking**
Bits: 0 or 1	Qubits: 0 *and* 1 (superposition)
Logic is deterministic	Logic is probabilistic*
Measurement reveals truth	Measurement changes the system
Copying is easy	No-cloning: copying is forbidden
State is local and separable	State can be entangled and nonlocal
Errors are binary and correctable	Errors are subtle, continuous, and must be detected differently

*Interfering **wave functions** shape the probability of outcomes. $P(x) = |\psi(x)|^2$

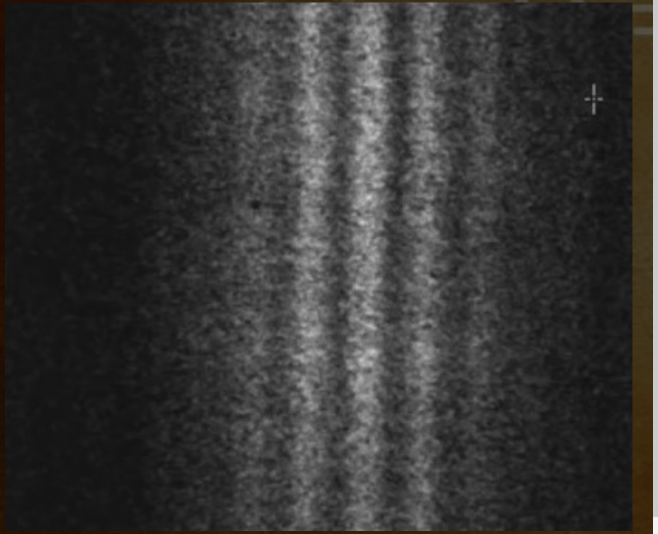
The Quantum Revolution

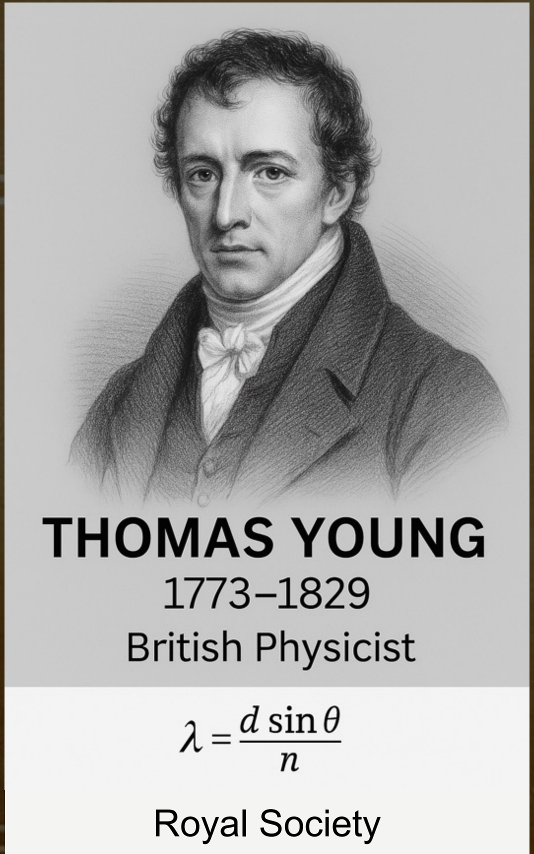


Foundations of Quantum Science

Erwin Schrödinger	Max Born	Werner Heisenberg	Louis de Broglie
1887–1961	1882–1970	1901–1976	1892–1987
Austrian	German	German	French
University of Vienna	University of Gőttingen	University of Leipzig	Collège de France
$ih\partial\Psi = H\Psi$	$\lvert\psi\rvert^2$	$\Delta x \, \Delta p \geq \frac{h}{2}$	$\lambda = \frac{h}{p}$

Photons interfere with one another like waves do





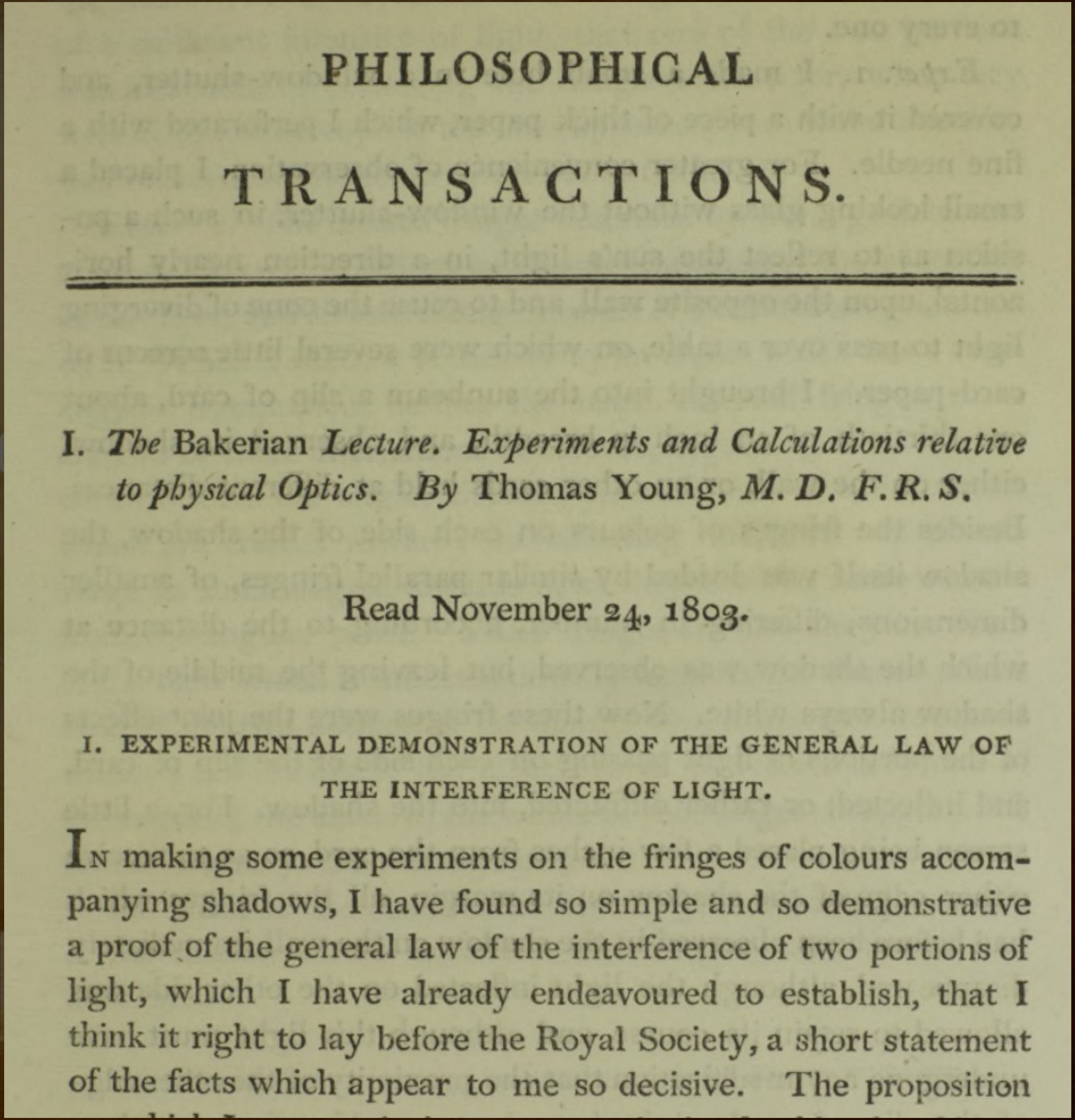
THOMAS YOUNG

1773–1829

British Physicist

$$\lambda = \frac{d \sin\theta}{n}$$

Royal Society



PHILOSOPHICAL TRANSACTIONS.

I. *The* Bakerian *Lecture. Experiments and Calculations relative to physical Optics. By* Thomas Young, *M. D. F. R. S.*

Read November 24, 1803.

I. EXPERIMENTAL DEMONSTRATION OF THE GENERAL LAW OF THE INTERFERENCE OF LIGHT.

In making some experiments on the fringes of colours accompanying shadows, I have found so simple and so demonstrative a proof of the general law of the interference of two portions of light, which I have already endeavoured to establish, that I think it right to lay before the Royal Society, a short statement of the facts which appear to me so decisive. The proposition

Thomas Young's interference experiments confirmed the wave nature of light, laying crucial groundwork for the later discovery of wave–particle duality.

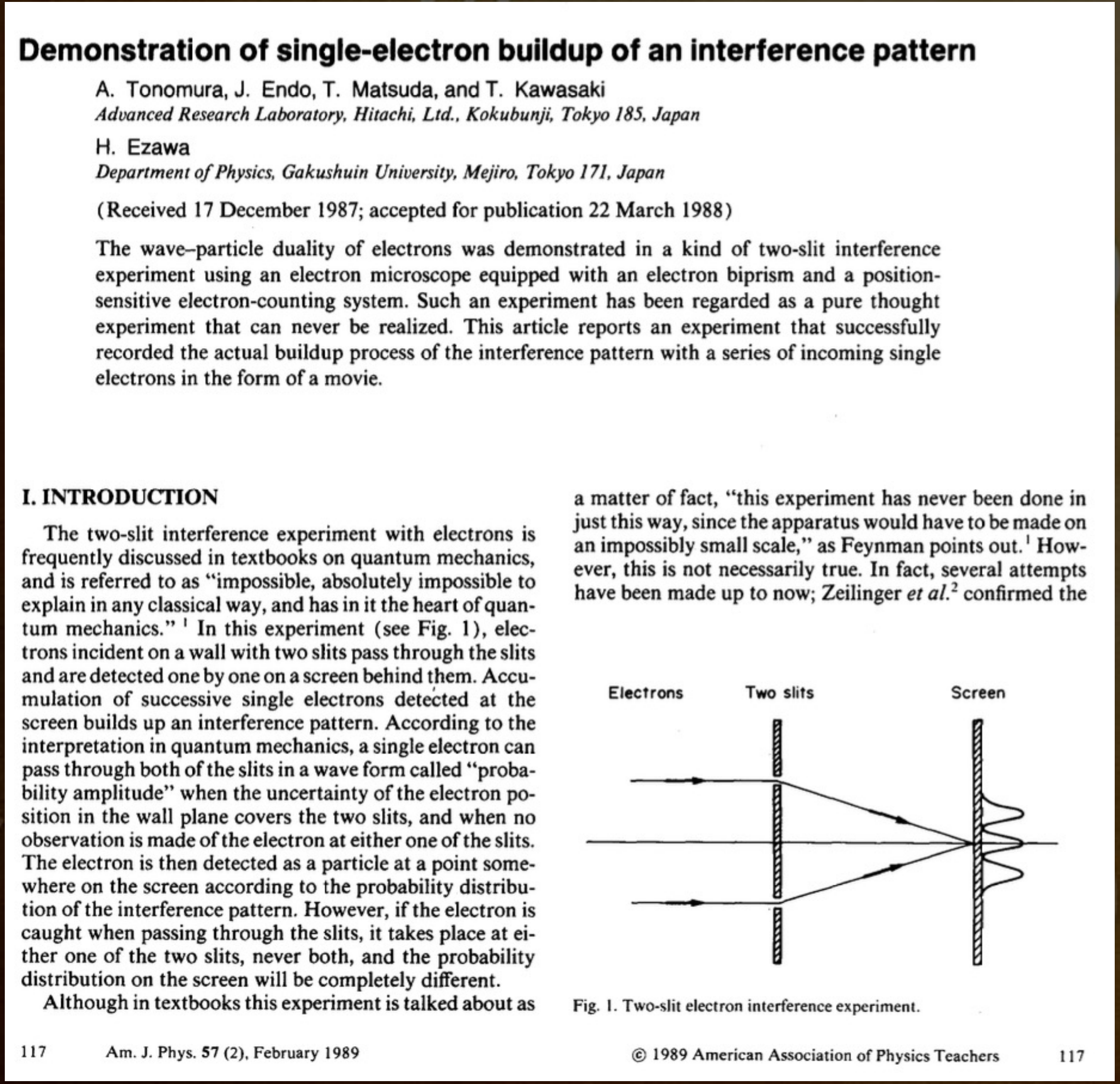
Wave-Particle Duality

Einstein (1905): Light quanta (photons) to explain the **photoelectric effect**

•**de Broglie (1924)**: Hypothesised that **matter (like electrons)** also has wave properties

Davisson–Germer (1927): Confirmed **electron diffraction**, validating de Broglie

Bohr, Heisenberg, Schrödinger (1920s): Formalised the **quantum framework**



Demonstration of single-electron buildup of an interference pattern

A. Tonomura, J. Endo, T. Matsuda, and T. Kawasaki
Advanced Research Laboratory, Hitachi, Ltd., Kokubunji, Tokyo 185, Japan

H. Ezawa
Department of Physics, Gakushuin University, Mejiro, Tokyo 171, Japan

(Received 17 December 1987; accepted for publication 22 March 1988)

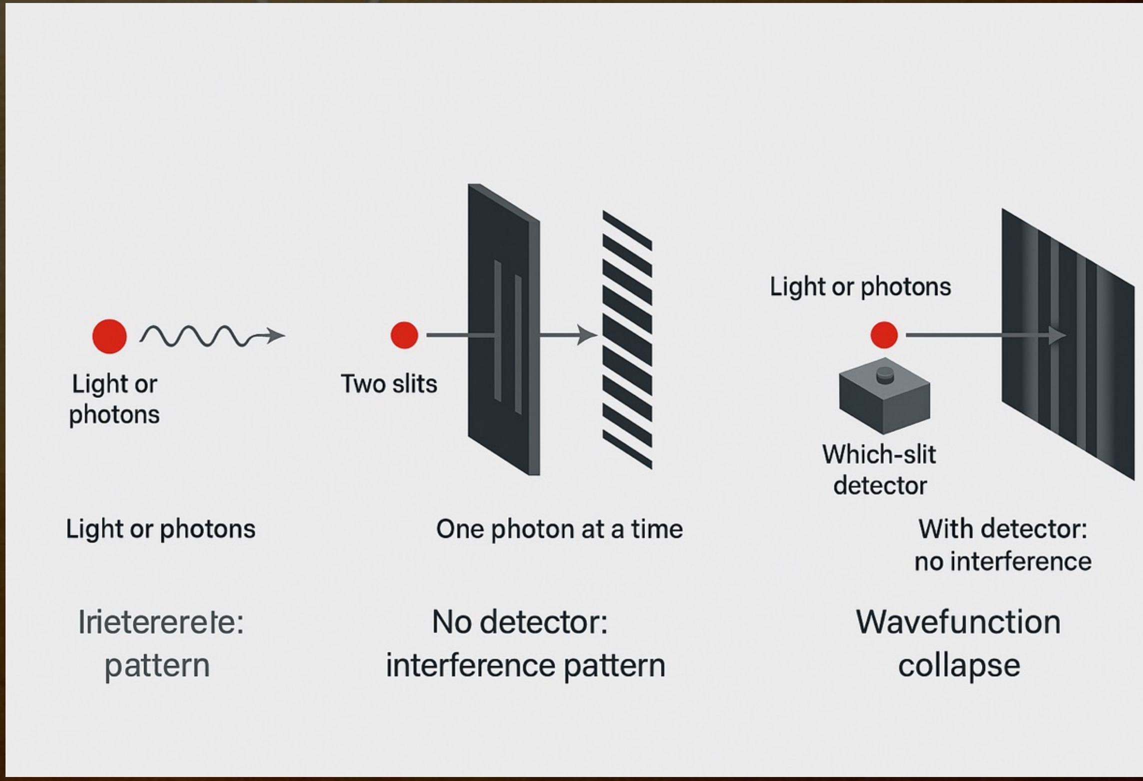
The wave–particle duality of electrons was demonstrated in a kind of two-slit interference experiment using an electron microscope equipped with an electron biprism and a position-sensitive electron-counting system. Such an experiment has been regarded as a pure thought experiment that can never be realized. This article reports an experiment that successfully recorded the actual buildup process of the interference pattern with a series of incoming single electrons in the form of a movie.

I. INTRODUCTION

The two-slit interference experiment with electrons is frequently discussed in textbooks on quantum mechanics, and is referred to as "impossible, absolutely impossible to explain in any classical way, and has in it the heart of quantum mechanics."[1] In this experiment (see Fig. 1), electrons incident on a wall with two slits pass through the slits and are detected one by one on a screen behind them. Accumulation of successive single electrons detected at the screen builds up an interference pattern. According to the interpretation in quantum mechanics, a single electron can pass through both of the slits in a wave form called "probability amplitude" when the uncertainty of the electron position in the wall plane covers the two slits, and when no observation is made of the electron at either one of the slits. The electron is then detected as a particle at a point somewhere on the screen according to the probability distribution of the interference pattern. However, if the electron is caught when passing through the slits, it takes place at either one of the two slits, never both, and the probability distribution on the screen will be completely different.

Although in textbooks this experiment is talked about as a matter of fact, "this experiment has never been done in just this way, since the apparatus would have to be made on an impossibly small scale," as Feynman points out.[1] However, this is not necessarily true. In fact, several attempts have been made up to now; Zeilinger *et al.*[2] confirmed the

Fig. 1. Two-slit electron interference experiment.



Light or photons

Two slits

Light or photons

Which-slit detector

Light or photons

One photon at a time

With detector: no interference

Irietererete: pattern

No detector: interference pattern

Wavefunction collapse

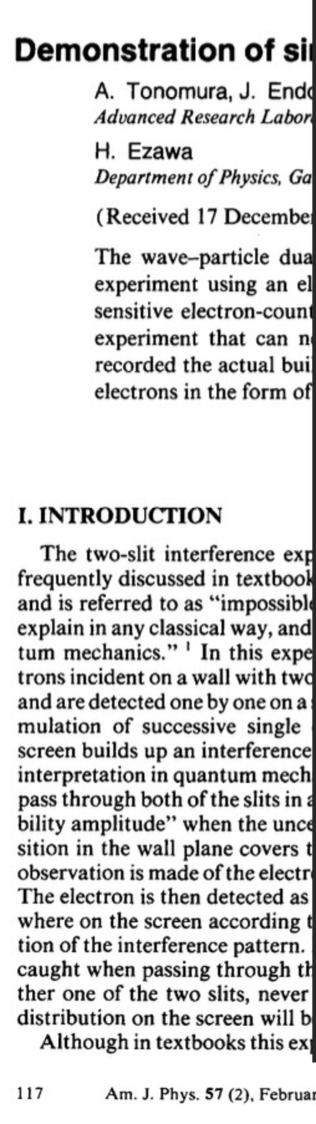
Wave-Particle Duality

Einstein (1905): Light quanta (photons) to explain the **photoelectric effect**

- **de Broglie (1924)**: Hypothesised that **matter (like electrons)**

ectron diffraction,

): Formalised the



Demonstration of si

A. Tonomura, J. Endo
Advanced Research Labor

H. Ezawa
Department of Physics, Ga

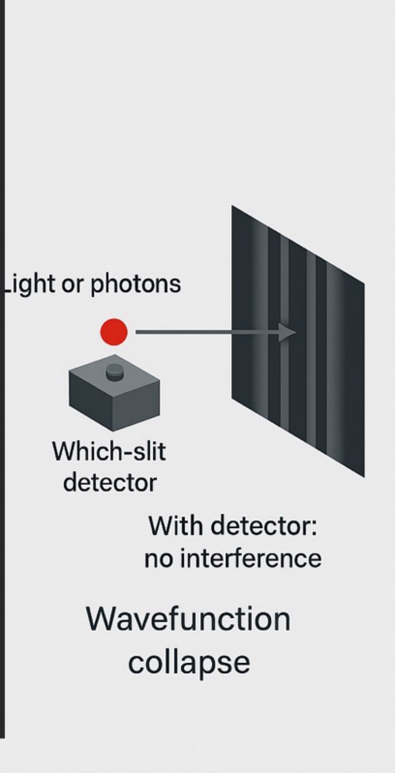
(Received 17 Decembe

The wave–particle dua
experiment using an el
sensitive electron-count
experiment that can n
recorded the actual bui
electrons in the form of

I. INTRODUCTION

The two-slit interference exp
frequently discussed in textbook
and is referred to as "impossibl
explain in any classical way, and
tum mechanics." [1] In this expe
trons incident on a wall with two
and are detected one by one on a
mulation of successive single
screen builds up an interference
interpretation in quantum mech
pass through both of the slits in a
bility amplitude" when the unce
sition in the wall plane covers t
observation is made of the electr
The electron is then detected as
where on the screen according t
tion of the interference pattern.
caught when passing through th
ther one of the two slits, never
distribution on the screen will b
Although in textbooks this ex

117 Am. J. Phys. 57 (2), February 1989 © 1989 American Association of Physics Teachers 117



Wavefunction collapse

Wave-Particle Duality

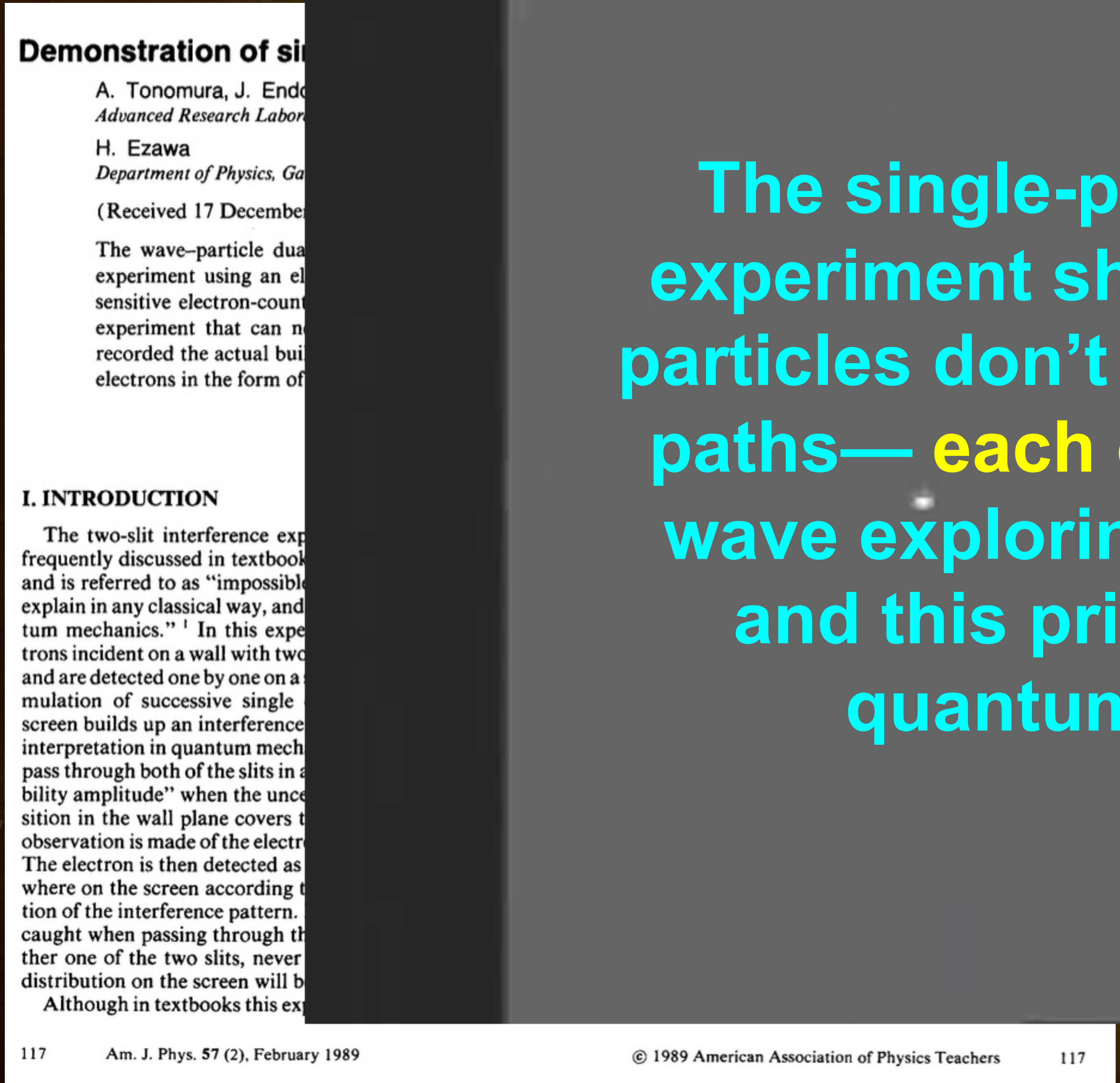
Einstein (1905): Light quanta (photons) to explain the **photoelectric effect**

•**de Broglie (1924)**: Hypothesised that **matter (like electrons)**

ectron diffraction,

): Formalised the

The single-photon double-slit experiment shows that quantum particles don’t follow independent paths— each one behaves like a wave exploring all possibilities, and this principle underlies quantum computing.



Demonstration of si

A. Tonomura, J. Endo

Advanced Research Labor

H. Ezawa

Department of Physics, Ga

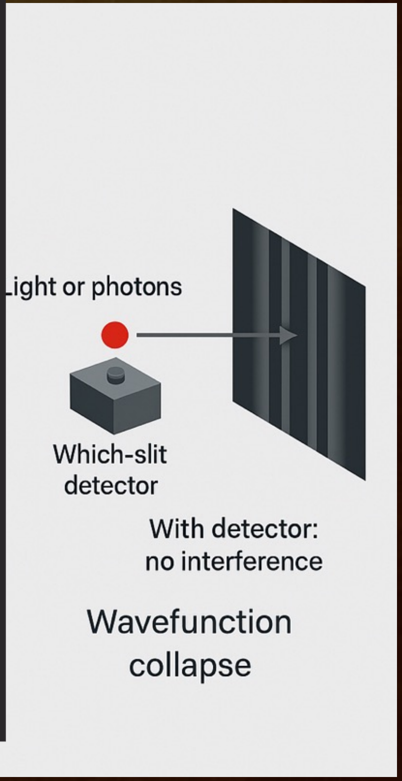
(Received 17 December

The wave–particle dua
experiment using an el
sensitive electron-count
experiment that can n
recorded the actual bui
electrons in the form of

I. INTRODUCTION

The two-slit interference exp
frequently discussed in textbook
and is referred to as “impossibl
explain in any classical way, and
tum mechanics.”[1] In this expe
trons incident on a wall with two
and are detected one by one on a
mulation of successive single
screen builds up an interference
interpretation in quantum mech
pass through both of the slits in a
bility amplitude” when the unce
sition in the wall plane covers t
observation is made of the electr
The electron is then detected as
where on the screen according t
tion of the interference pattern.
caught when passing through th
ther one of the two slits, never
distribution on the screen will b
Although in textbooks this exp

117 Am. J. Phys. 57 (2), February 1989 © 1989 American Association of Physics Teachers 117



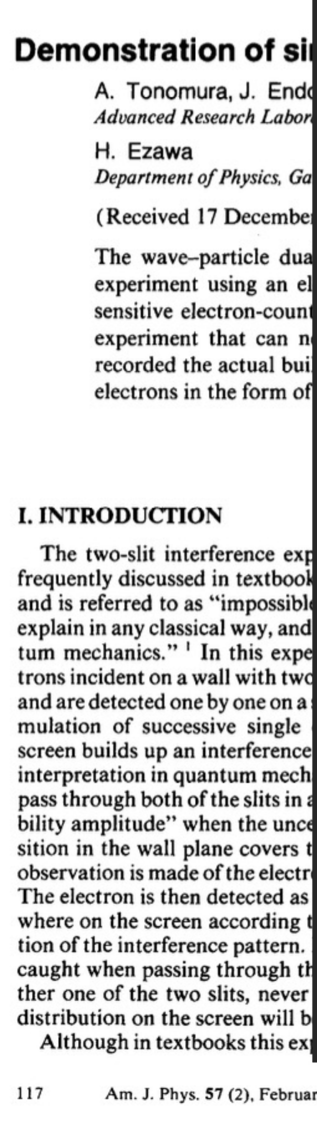
Wave-Particle Duality

Einstein (1905): Light quanta (photons) to explain the **photoelectric effect**

•**de Broglie (1924)**: Hypothesised that **matter (like electrons)**

ectron diffraction,

): Formalised the



Demonstration of si

A. Tonomura, J. End

Advanced Research Labor

H. Ezawa

Department of Physics, Ga

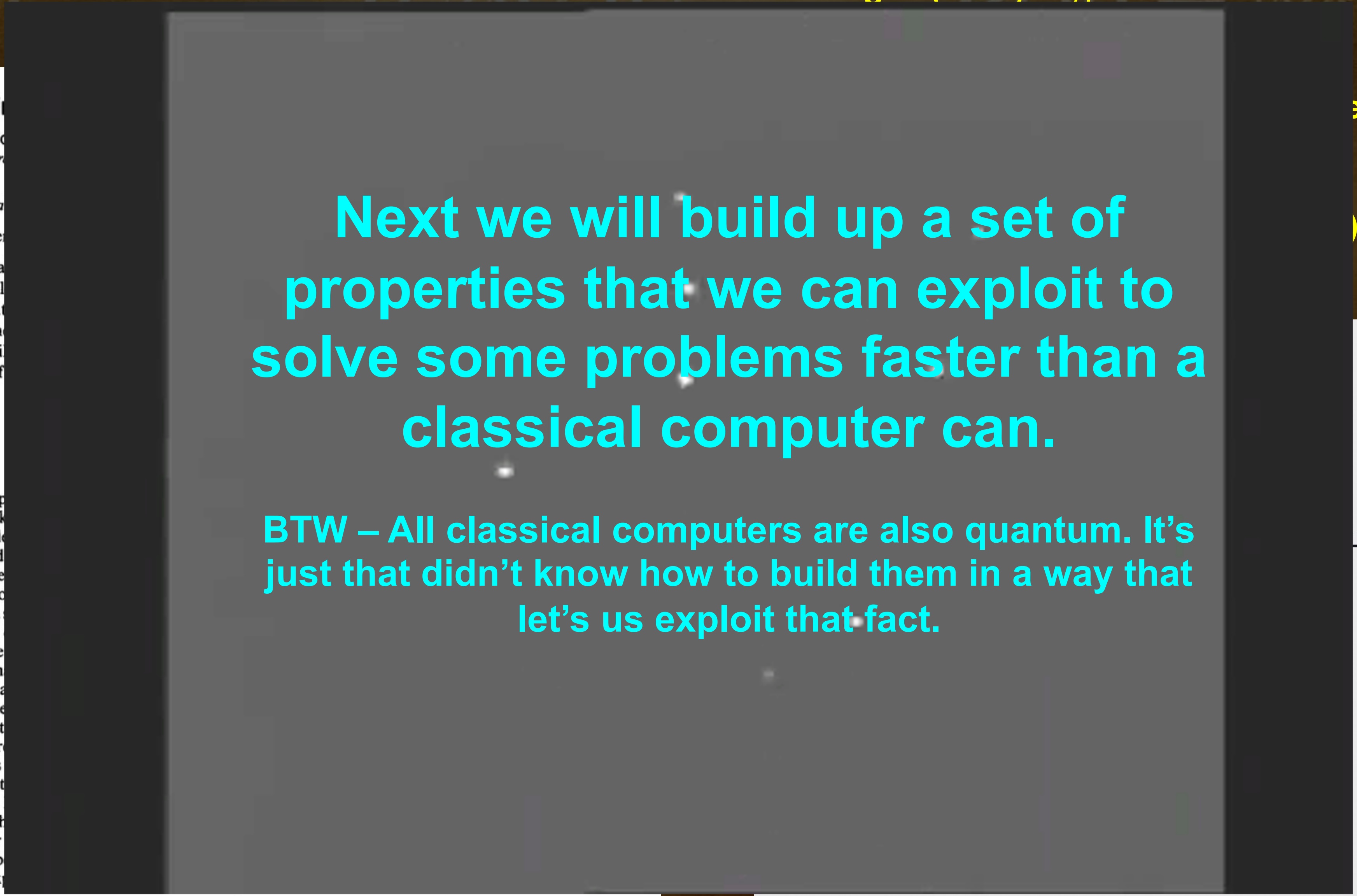
(Received 17 Decembe

The wave–particle dua
experiment using an el
sensitive electron-coun
experiment that can n
recorded the actual bui
electrons in the form of

I. INTRODUCTION

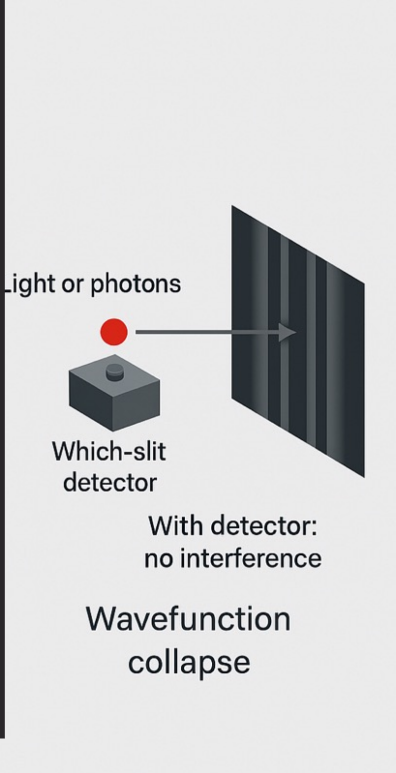
The two-slit interference exp
frequently discussed in textbook
and is referred to as "impossibl
explain in any classical way, and
tum mechanics."[1] In this expe
trons incident on a wall with two
and are detected one by one on a
mulation of successive single
screen builds up an interference
interpretation in quantum mech
pass through both of the slits in
bility amplitude" when the unce
sition in the wall plane covers t
observation is made of the electr
The electron is then detected as
where on the screen according t
tion of the interference pattern.
caught when passing through th
ther one of the two slits, never
distribution on the screen will b
Although in textbooks this ex

117 Am. J. Phys. 57 (2), February 1989 © 1989 American Association of Physics Teachers 117



Next we will build up a set of properties that we can exploit to solve some problems faster than a classical computer can.

BTW – All classical computers are also quantum. It's just that didn't know how to build them in a way that let's us exploit that fact.



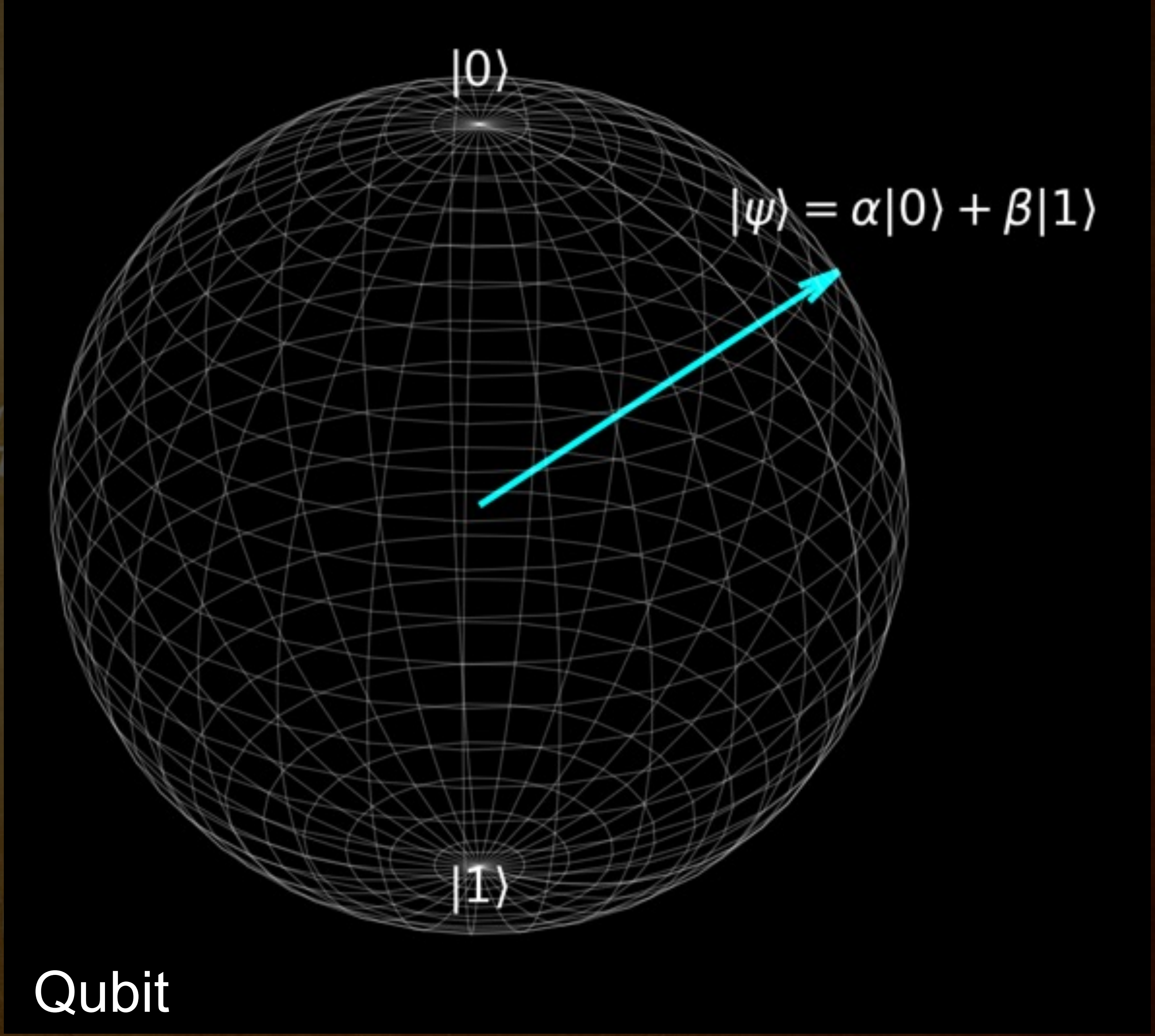
Superposition

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

This expression is the quantum equivalent of a wave function: it describes a qubit's state as a superposition of basis states, not a single bit value. Psi the quantum state vector and is a linear combination of 0 and 1 the "ket" here just means a column vector.

$$|0\rangle = \begin{bmatrix} 1 \\ 0 \end{bmatrix} \quad \text{and} \quad |1\rangle = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$$

A qubit isn't just 0 or 1—it's like a wave that can be both at once, with weights that tell us how likely each outcome is when we measure it.



Superposition

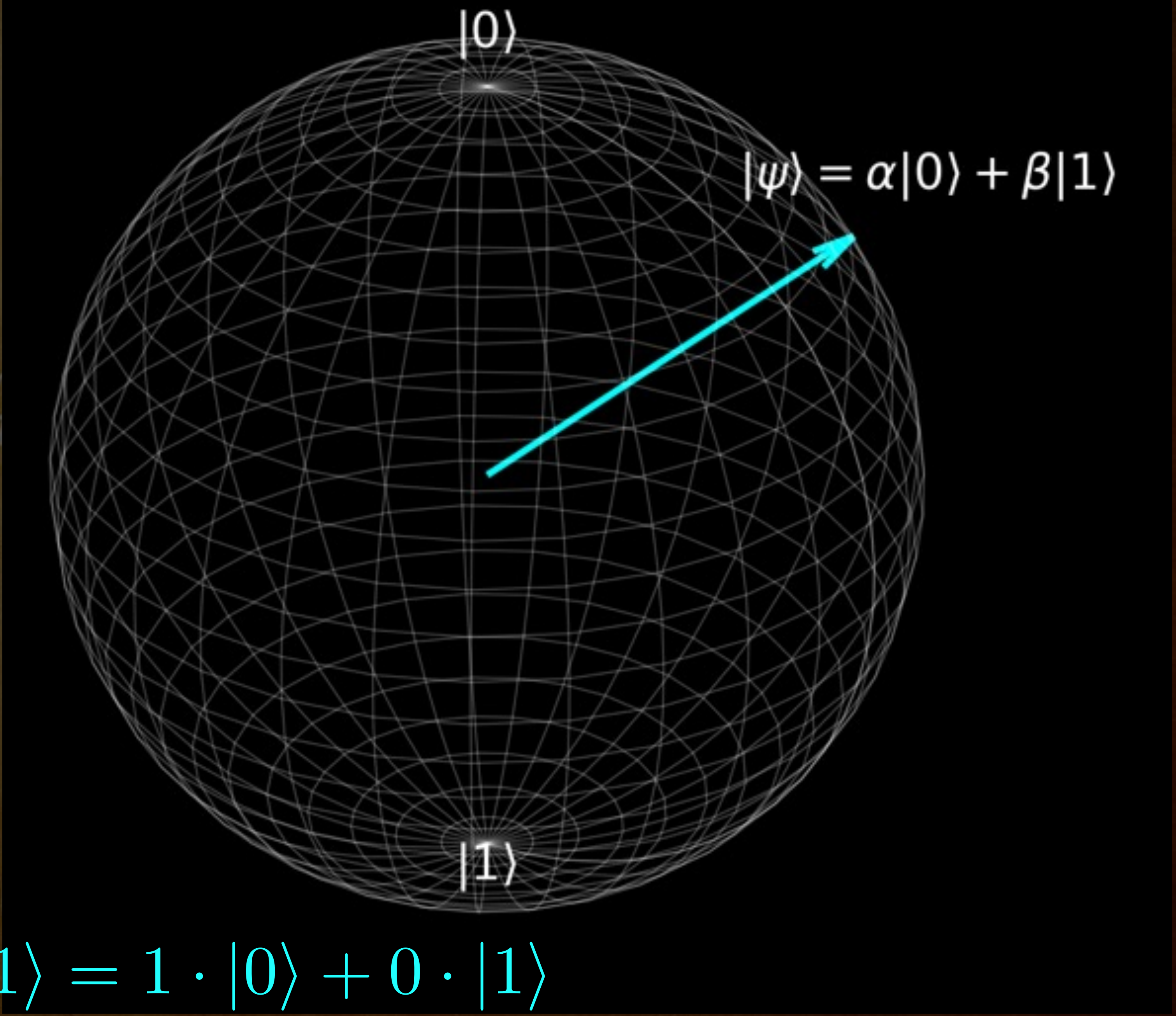
$$|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$$

$$|0\rangle = \begin{bmatrix}1\\0\end{bmatrix} \quad \text{and} \quad |1\rangle = \begin{bmatrix}0\\1\end{bmatrix}$$

We map the position on the sphere with the Bloch formula:

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

E.g for Theta = 0:



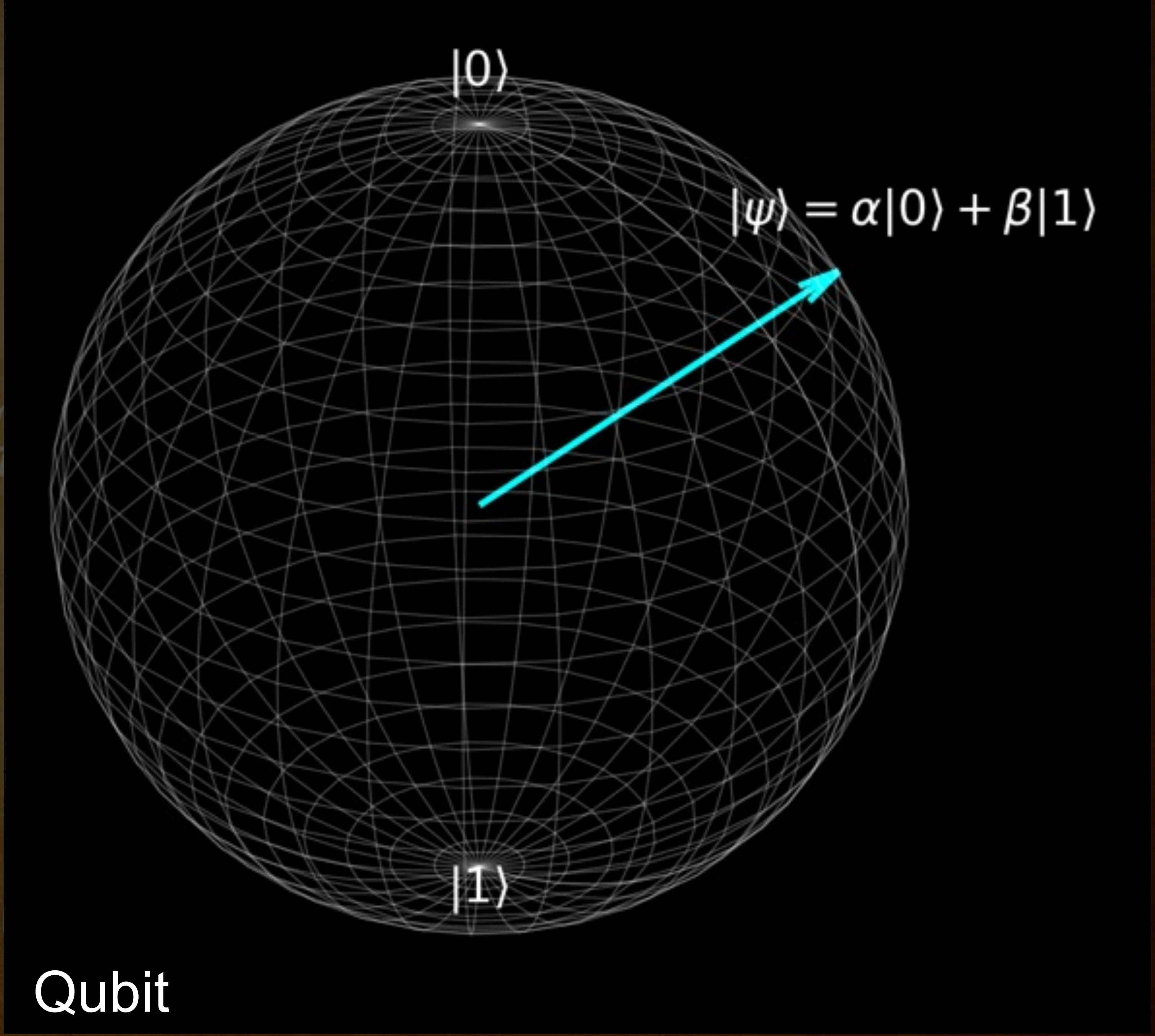
$$|\psi\rangle = \cos(0)|0\rangle + \sin(0)|1\rangle = 1\cdot|0\rangle + 0\cdot|1\rangle$$

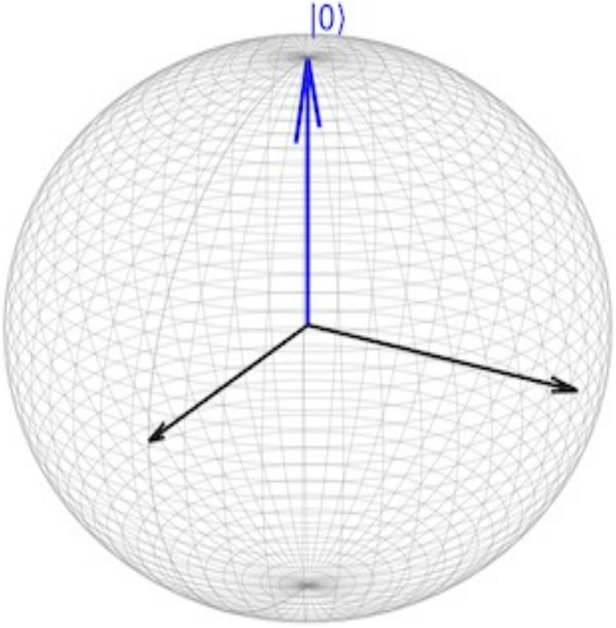
Superposition

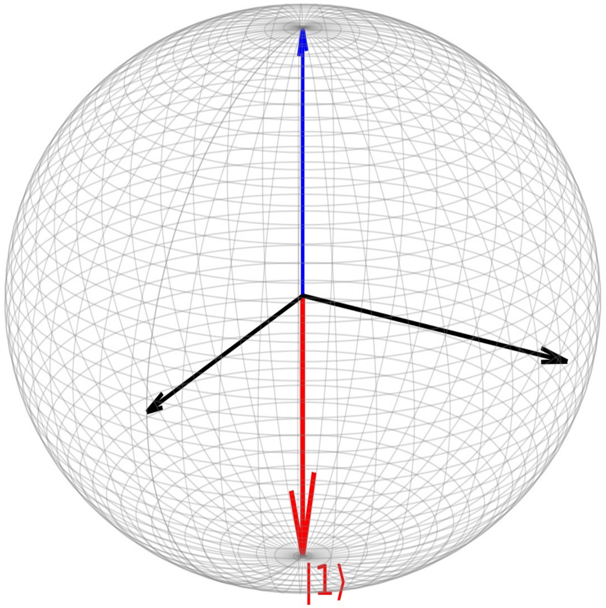
$$|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$$

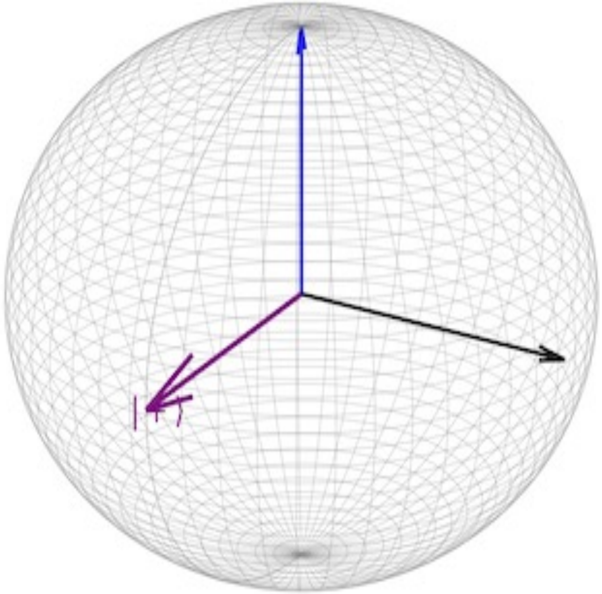
A qubit isn't just 0 or 1—it's like a wave that can be both at once, with weights that tell us how likely each outcome is when we measure it.

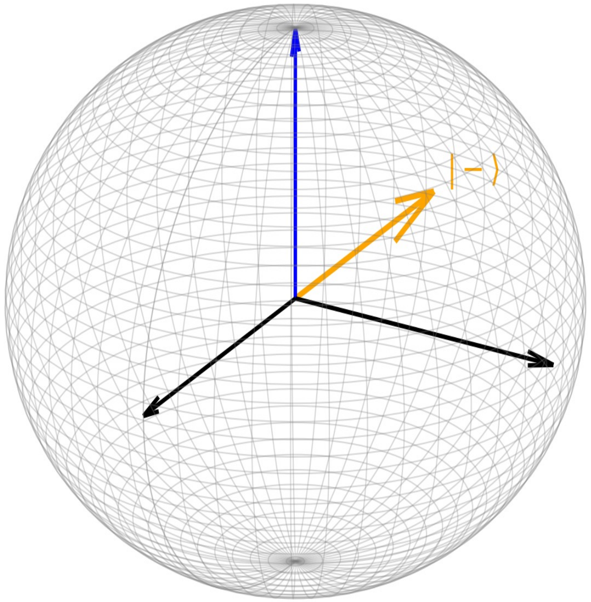
The formula is Dirac notation for mapping the superposition to a sphere.











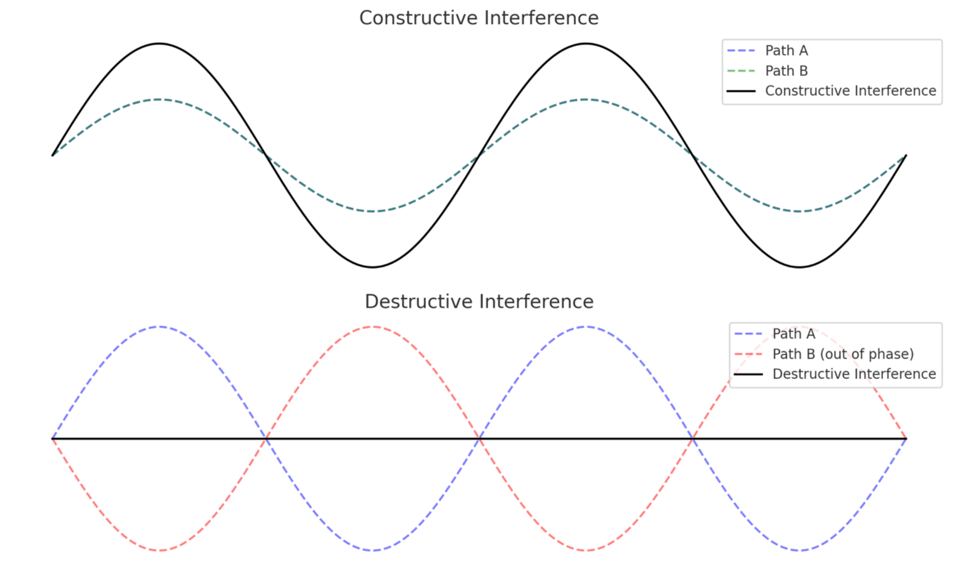
$$|0\rangle = \begin{bmatrix} 1 \\ 0 \end{bmatrix} \qquad |1\rangle = \begin{bmatrix} 0 \\ 1 \end{bmatrix} \qquad |+\rangle = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 \\ 1 \end{bmatrix} \quad |-\rangle = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 \\ -1 \end{bmatrix}$$

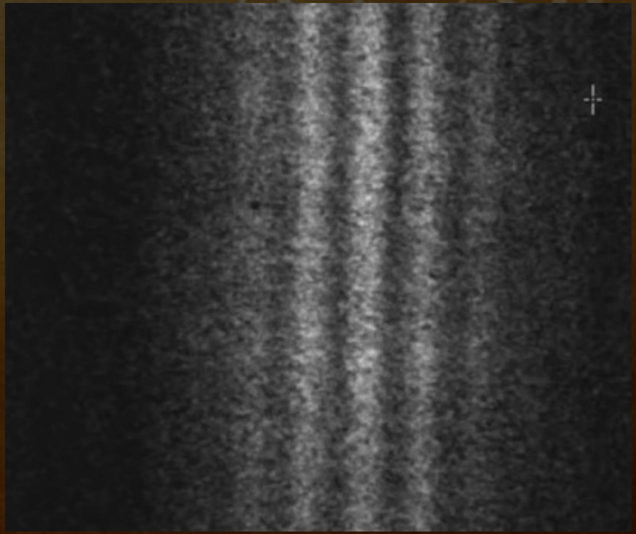
Standard Basis

Hadamard Basis

Quantum Interference

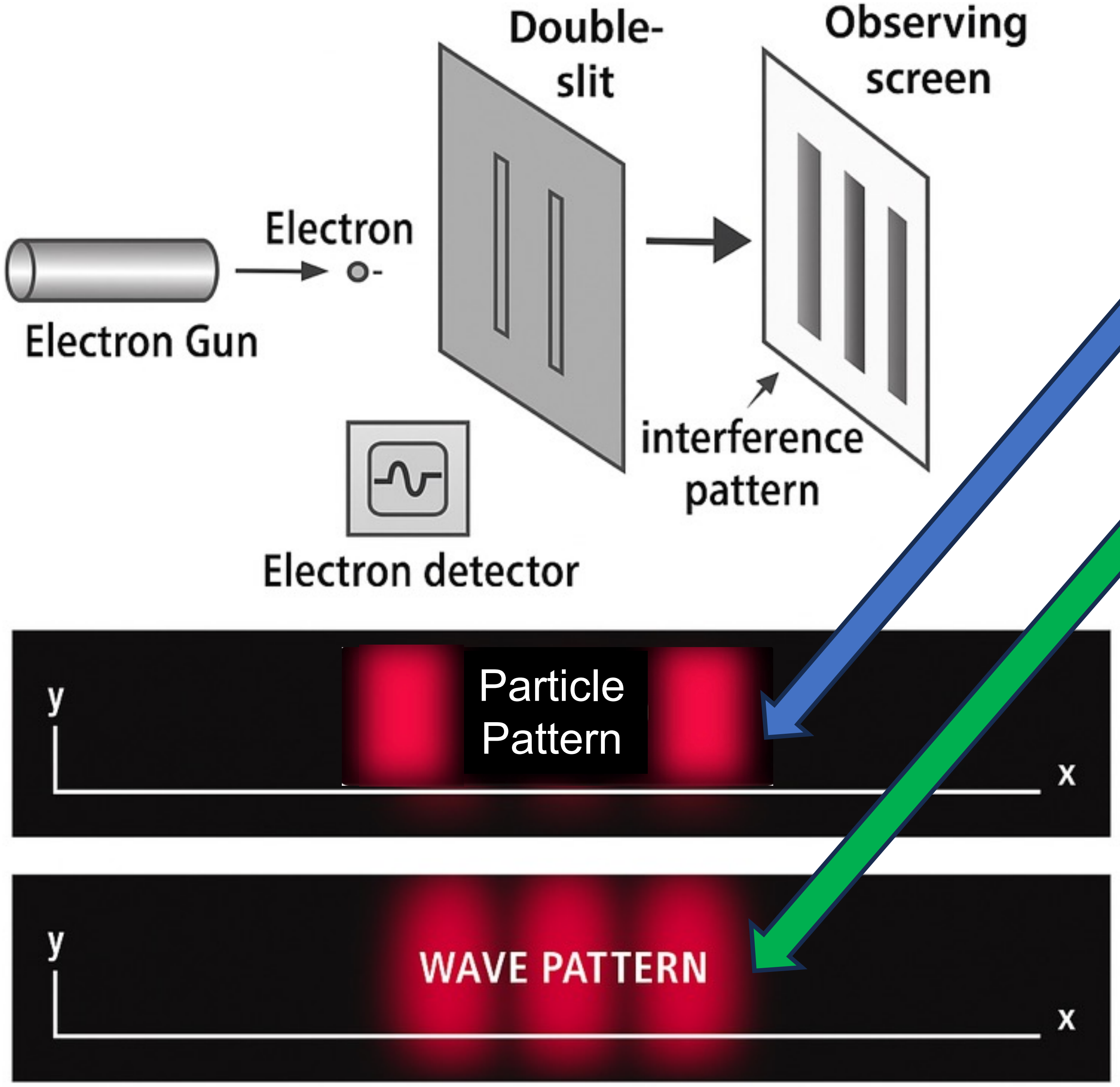
Concept	**Quantum Interference**
Like in waves	Quantum amplitudes can add or cancel
Constructive	Amplitudes **reinforce**, increasing outcome probability
Destructive	Amplitudes **cancel**, suppressing outcomes
Why it matters	Used in quantum algorithms to find correct answers faster





Unitary Evolution & Reversibility

Classical Systems	**Quantum Systems**
Many processes are **irreversible** (e.g. XOR, erasure)	All quantum operations must be **reversible**
Bit states collapse or overwrite	Qubit states evolve smoothly under unitary gates
Logic gates may lose information	Quantum gates preserve information (unitary)
Example: NAND cannot be undone	Example: Hadamard, CNOT can be reversed



With measurement (Electron detector let’s us know whether an electron went through a particular slit): interference disappears

Without measurement: wave interference pattern appears

Mirrors **quantum computing**: coherence enables interference

Measurement = collapse

Decoherence = unintended collapse

Measurement Collapses the Wavefunction

- **Measurement collapses** the quantum state into $|0\rangle or |1\rangle$*
  Collapse is **probabilistic** and **irreversible**

- **Decoherence** is unintentional measurement by the environment

- Quantum computation must **preserve coherence** until final measurement

- **Unitary operations preserve coherence**, which allows quantum superpositions to evolve and interfere. **Measurement and decoherence both break coherence**, collapsing or scrambling those superpositions.

* Recall this notation just means $|0\rangle = \begin{bmatrix} 1 \\ 0 \end{bmatrix} \quad \text{and} \quad |1\rangle = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$ which is a definite point on the surface of the Bloch sphere representing the qubit

Decoherence: When Quantum Becomes Classical

- **Wavefunction collapse** is what we **observe** when we measure a quantum system.

- **Decoherence** is a **physical explanation** for why quantum states lose superposition **without measurement** due to entanglement with the environment.

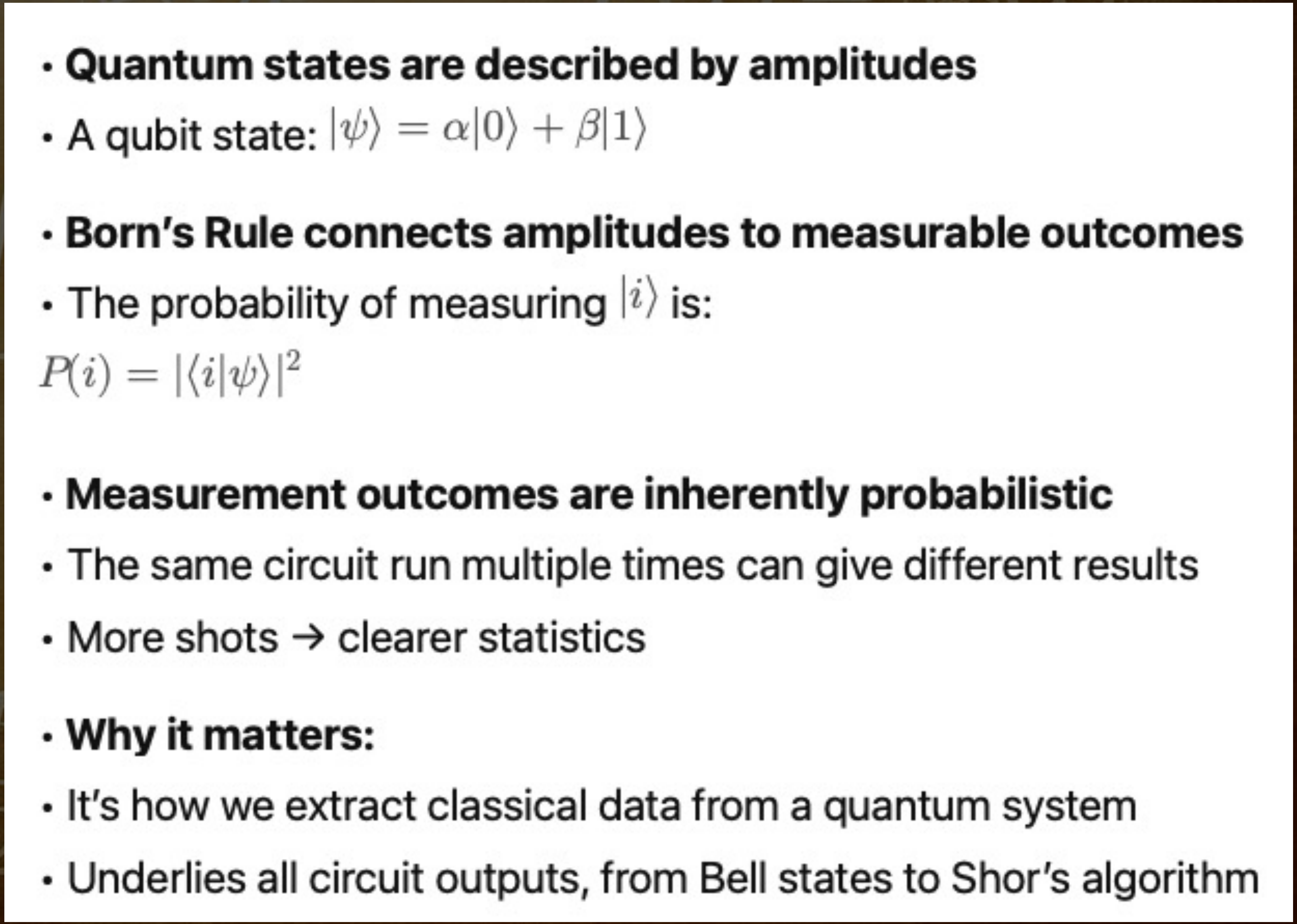
- BTW this is why sticking a cat into a box with cyanide and a cesium atom trigger wouldn’t really put the cat in a superposition of live and dead, unless the box it is in isolates the contents from the rest of the universe at a quantum level.

Born’s Rule

A **shot** is one run of the circuit from initial state preparation to final measurement.

We need multiple shots to converge on a solution.

Alex?



- **Quantum states are described by amplitudes**
- A qubit state: $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$
- **Born’s Rule connects amplitudes to measurable outcomes**
- The probability of measuring $|i\rangle$ is:

$P(i) = |\langle i|\psi\rangle|^2$

- **Measurement outcomes are inherently probabilistic**
- The same circuit run multiple times can give different results
- More shots → clearer statistics
- **Why it matters:**
- It’s how we extract classical data from a quantum system
- Underlies all circuit outputs, from Bell states to Shor’s algorithm

Summary: What We Know

We have seen that the double-slit experiment revealed a fundamental property of the universe.

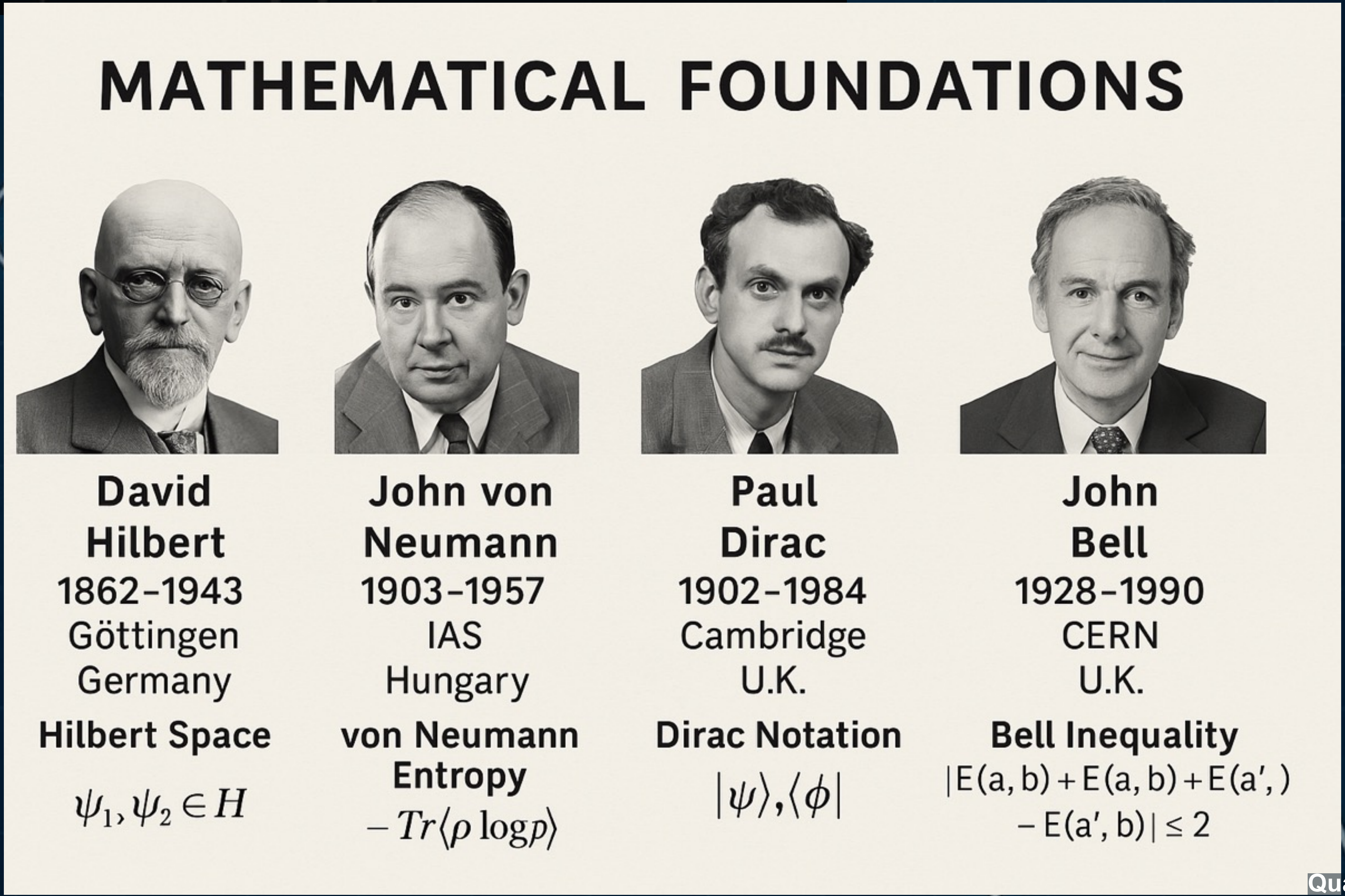
Until measurement or decoherence objects* exist as a wave function governed by Schrodinger's equation. This means they exist in all possible states simultaneously.

When unintended interactions with the environment occur, or an intended measurement occurs, the wave function collapses into just one of the states.

Quantum computers maintain that Wave function state of qubits in order to perform computations faster.

📖 e.g. Bucky Ball made of 240 Carbon atoms. Fein, Y., et al. (2019). *“Quantum superposition of molecules beyond 25 kDa.”* **Nature Physics**, **15**, 1242-1245. **DOI:** 10.1038/s41567-019-0663-9

From Bits to Qubits



MATHEMATICAL FOUNDATIONS

David Hilbert
1862–1943
Göttingen
Germany
Hilbert Space
$\psi_1, \psi_2 \in H$

John von Neumann
1903–1957
IAS
Hungary
von Neumann Entropy
$-Tr\langle\rho \log p\rangle$

Paul Dirac
1902–1984
Cambridge
U.K.
Dirac Notation
$|\psi\rangle, \langle\phi|$

John Bell
1928–1990
CERN
U.K.
Bell Inequality
$|E(a, b) + E(a, b) + E(a',) - E(a', b)| \leq 2$

Quantum Gates: Basic Building Blocks

- Gates must be reversable

Logical “not” is reversable. If you see not(x) = 1, you know that x must be 0

Logical “or” is not reversable. or(x,y)=1 doesn’t fix whether x was 1 or y was 1.

Nothing irreversible can happen in Schrodinger’s world.

Quantum Gates: Basic Building Blocks

Operating on bits involve flipping some on/off value: A low high voltage on a wire, magnetizing a small region on an iron coated disk, or laser writing pits into a surface. These encode 0s and 1s.

We manipulate the 0s and 1s with binary operators following an deterministic algorithm,

In quantum computing: Qubits are a superposition of 0 and 1 that defines the surface of a sphere.

We modify the qubit by changing the probability amplitudes that define where the qubit is likely to be on the Bloch sphere when we measure it.

So quantum operations are usually not writing a 0 or 1 it is applying a rotation on the vector that defines the qubit.

So rather than writing 01+01=10 as we would adding binary numbers, we might rotate the qbit about the x axis by angle theta with the RX(theta) operator.

Quantum Gates: Basic Building Blocks

A lot of quantum operations (gates) are analogous to classical gates.

But to take advantage of superposition and coherence we have to make reversable versions.

For example, exclusive or (xor) is not reversable …

xor(0,0) = 0,   xor(0,1) = 1
xor(1,0) = 1,   xor(1,1) = 0

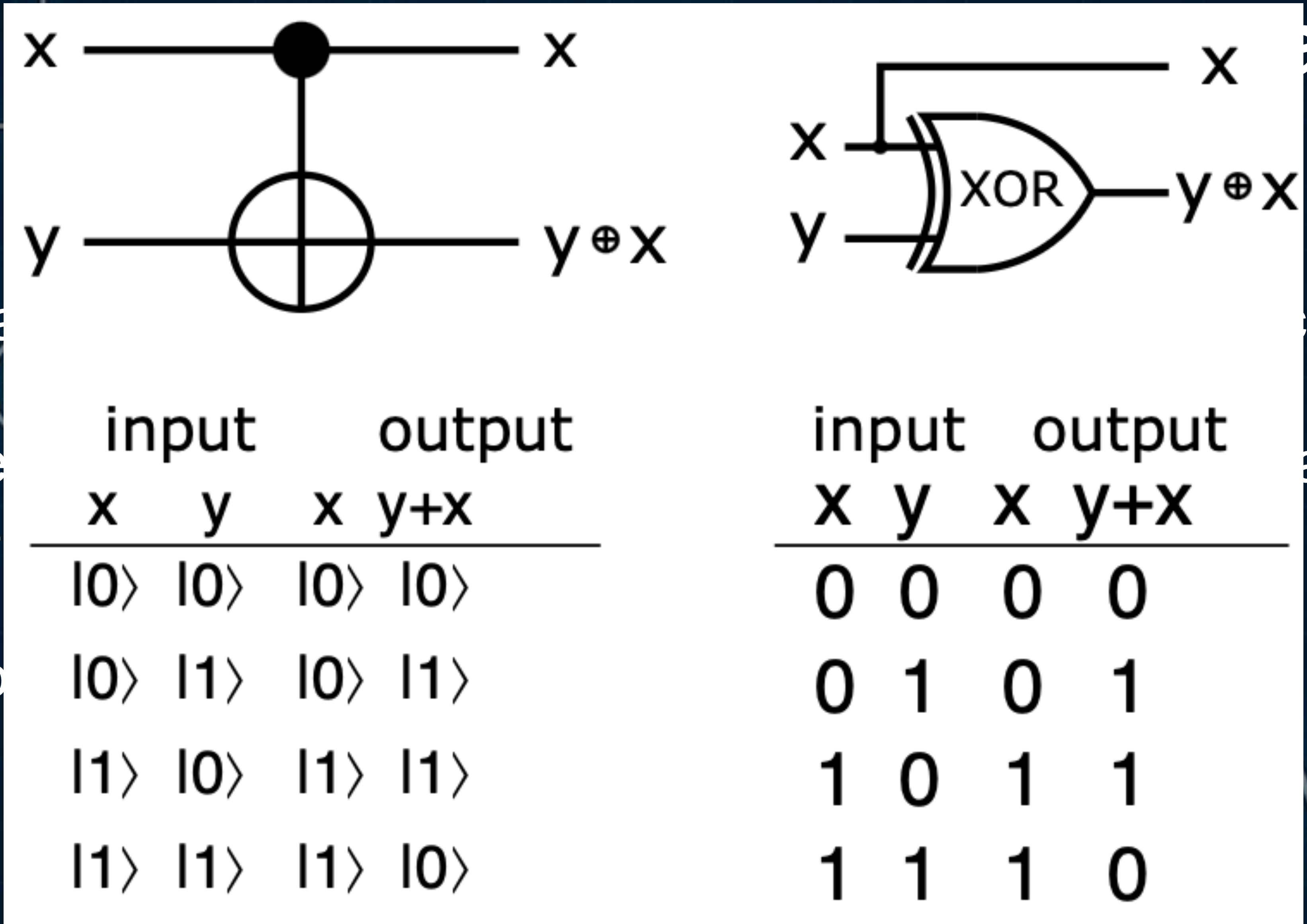
If you get a 1 you don’t know which of the inputs was 1

asic

A lot of qua al gates.

But to take ave to make reversable

For examp



input		output	
x	y	x	y+x
|0⟩	|0⟩	|0⟩	|0⟩
|0⟩	|1⟩	|0⟩	|1⟩
|1⟩	|0⟩	|1⟩	|1⟩
|1⟩	|1⟩	|1⟩	|0⟩

input		output	
x	y	x	y+x
0	0	0	0
0	1	0	1
1	0	1	1
1	1	1	0

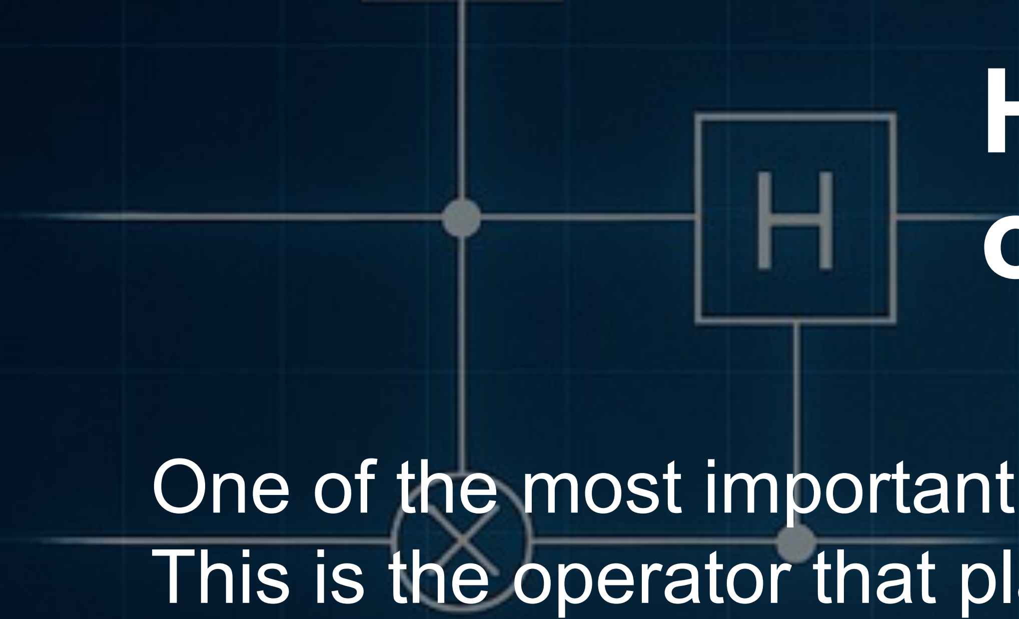
But control xor is – we output the value of one of the inputs to make it reversable

$$|0\rangle = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$$

$$|1\rangle = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$$

Symbol	Name	Description	Matrix Representation
I	Identity	Does nothing to the qubit	$\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$
X	Pauli-X	Bit-flip gate (like NOT)	$\begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$
Y	Pauli-Y	Bit & phase flip	$\begin{bmatrix} 0 & -i \\ i & 0 \end{bmatrix}$
Z	Pauli-Z	Phase flip gate	$\begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}$
H	Hadamard	Creates superposition	$\frac{1}{\sqrt{2}}\begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$
S	Phase Gate	Applies phase of i	$\begin{bmatrix} 1 & 0 \\ 0 & i \end{bmatrix}$
T	$\pi/8$ Gate	Applies phase of $e^{i\pi/4}$	$\begin{bmatrix} 1 & 0 \\ 0 & e^{i\pi/4} \end{bmatrix}$
RX(θ)	X-Rotation	Rotates around X-axis by θ	$\cos\left(\frac{\theta}{2}\right) I - i \sin\left(\frac{\theta}{2}\right) X$
RY(θ)	Y-Rotation	Rotates around Y-axis by θ	$\cos\left(\frac{\theta}{2}\right) I - i \sin\left(\frac{\theta}{2}\right) Y$
RZ(θ)	Z-Rotation	Rotates around Z-axis by θ	$\begin{bmatrix} 1 & 0 \\ 0 & e^{i\theta} \end{bmatrix}$
CX	CNOT	Controlled-X gate	Acts on 2 qubits: flips target if control is $\vert 1\rangle$
CZ	Controlled-Z	Applies Z if control is $\vert 1\rangle$	Diagonal matrix with -1 at bottom right
SWAP	Swap	Swaps two qubits	Interchanges the two states

Table 1: Common Unitary Quantum Gates

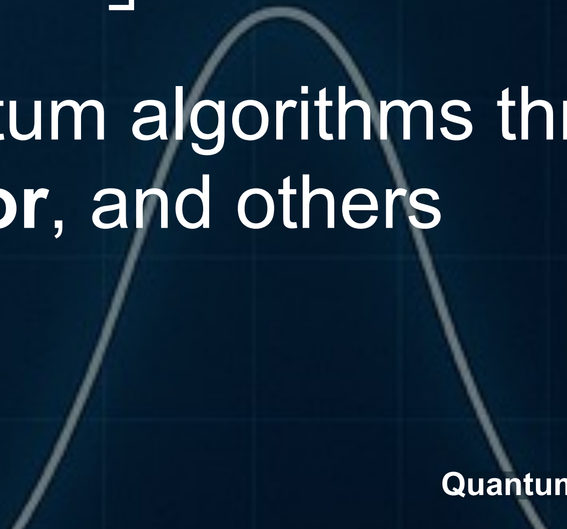


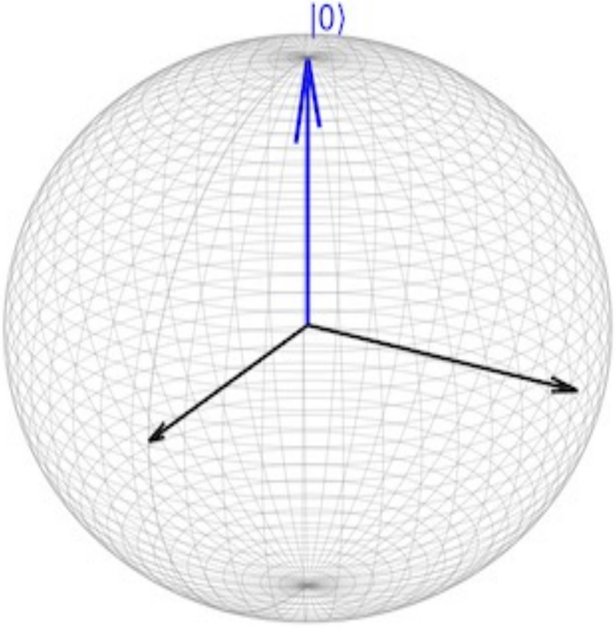
Hadamard and the Birth of Superposition

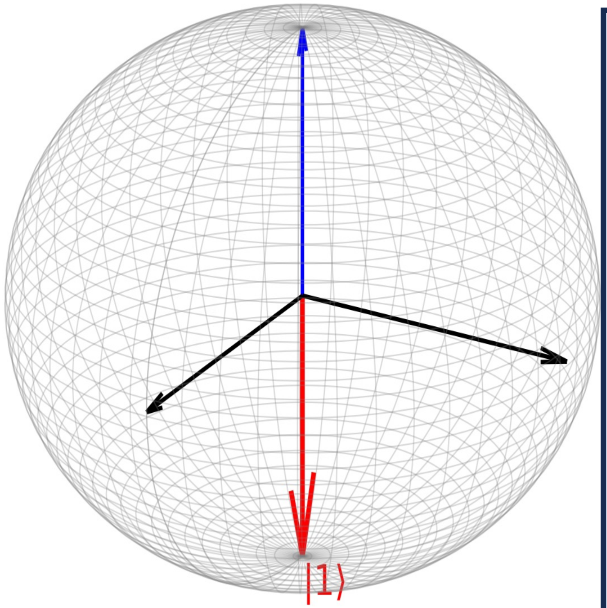
One of the most important operators is H, the Hadamard operator. This is the operator that places qbits into superposition so they obey Schrodinger’s equation.

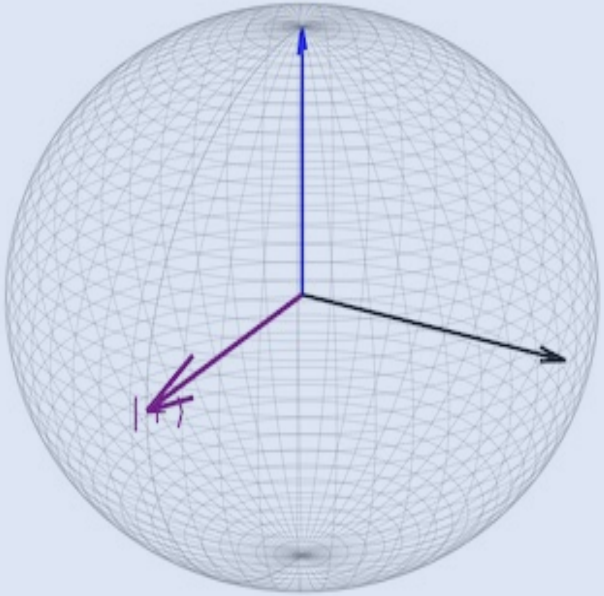
$$H = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$$

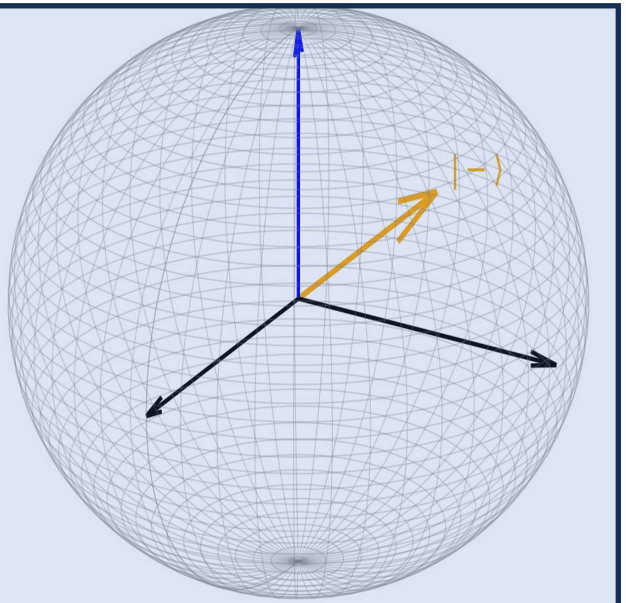
•The gate itself became standardised in quantum algorithms through the work of **David Deutsch** (1985), **Peter Shor**, and others developing quantum logic circuits.











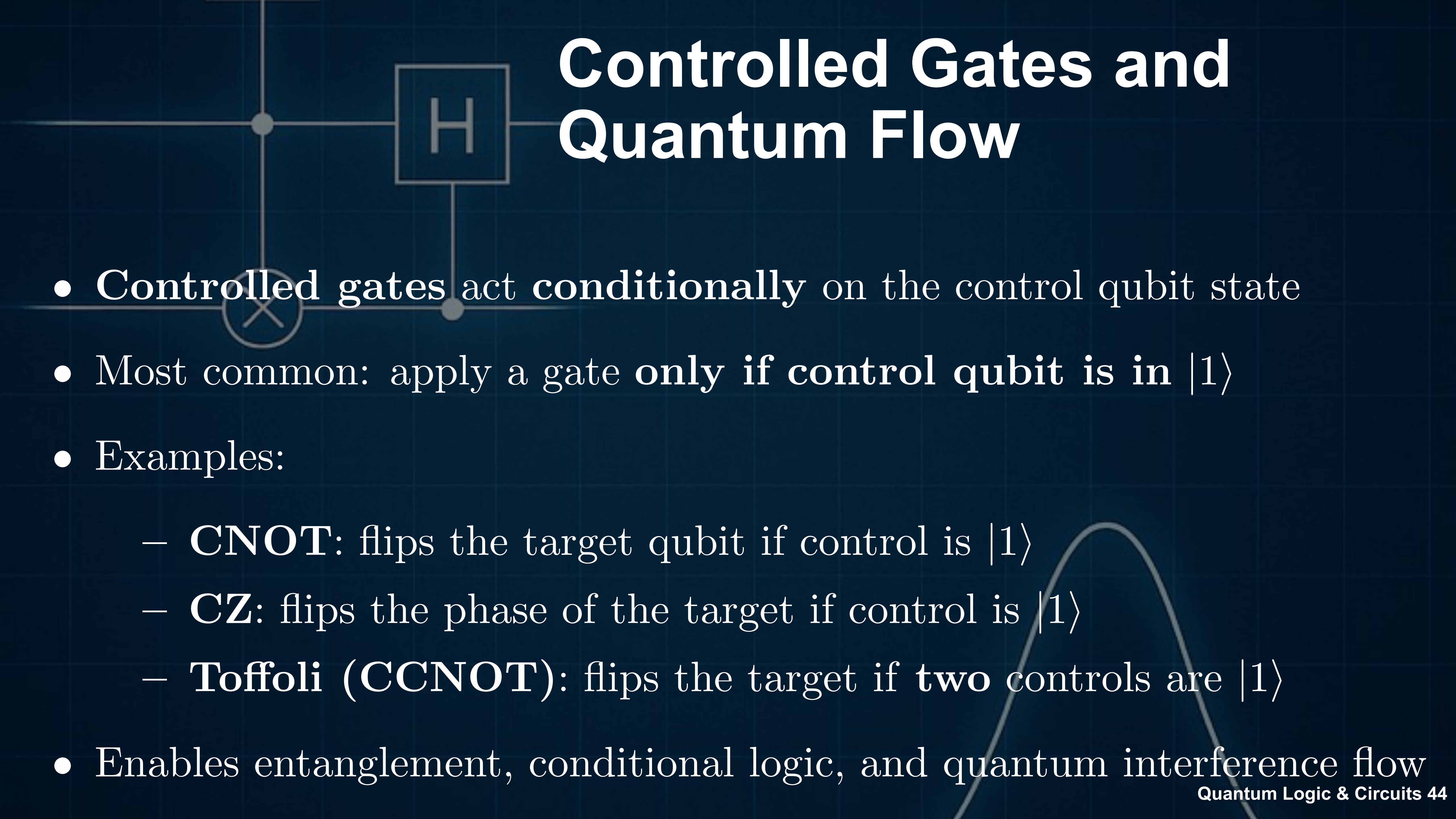
$$|0\rangle = \begin{bmatrix} 1 \\ 0 \end{bmatrix} \qquad |1\rangle = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$$

Standard Basis

$$|+\rangle = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 \\ 1 \end{bmatrix} \quad |-\rangle = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 \\ -1 \end{bmatrix}$$

Hadamard Basis

This is how we put the qubits into superposition



Controlled Gates and Quantum Flow

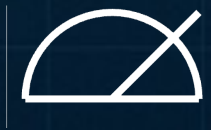
- **Controlled gates** act **conditionally** on the control qubit state
- Most common: apply a gate **only if control qubit is in** $|1\rangle$
- Examples:
  - **CNOT**: flips the target qubit if control is $|1\rangle$
  - **CZ**: flips the phase of the target if control is $|1\rangle$
  - **Toffoli (CCNOT)**: flips the target if **two** controls are $|1\rangle$
- Enables entanglement, conditional logic, and quantum interference flow

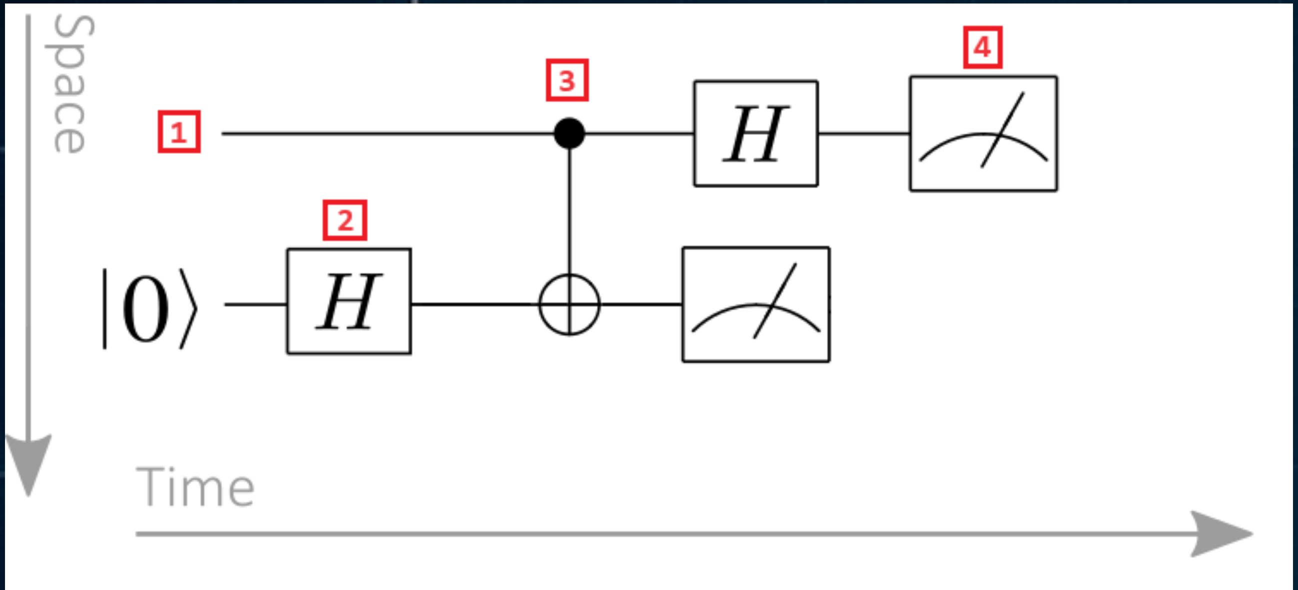
Quantum Circuit Diagrams 101

- **Qubit wires**: horizontal lines (top to bottom = different qubits)
- **Time flows left to right**

- **Boxes** = gates (e.g., H, X, Z, etc.)

- **Dots and lines** = control and target (e.g., CNOT)

- **Measurement** shown with  or [M]



1. **Qubit Registers** – Horizontal lines represent qubits. Top to bottom = Qubit 0, Qubit 1, etc.

2. **Quantum Gate** – Boxes apply quantum operations. The `H` gate here puts the qubit into superposition.

3. **Controlled Gate** – The CNOT gate uses a control (filled circle) and a target (plus in a circle). It flips the target qubit if the control is $|1\rangle$.

4. **Measurement** – Meter symbols measure qubits, collapsing them to classical bits (0 or 1).

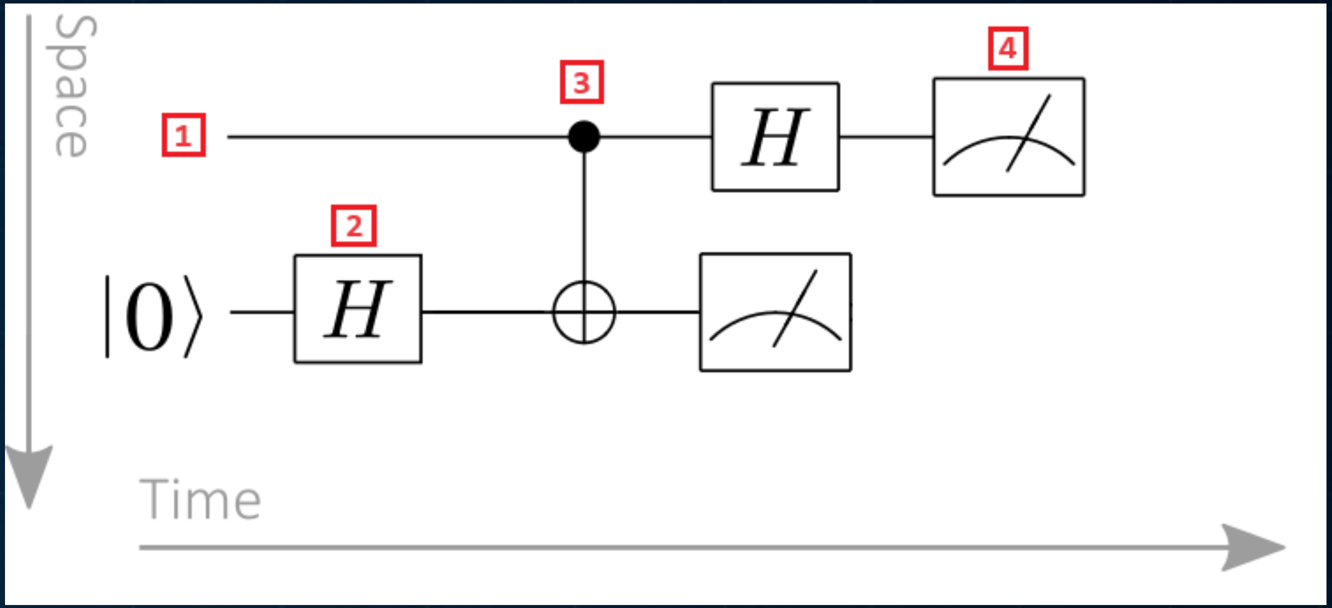
https://learn.microsoft.com/en-us/azure/quantum/concepts-circuits

1. Hadamard gate on qubit 0:

$|0\rangle \xrightarrow{H} \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$

$\rightarrow$ Now the state is:

$\frac{1}{\sqrt{2}}(|0\rangle + |1\rangle) \otimes |0\rangle$



2. CNOT gate, with:

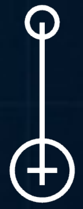
Control: qubit 0

Target: qubit 1

$\rightarrow$ This flips the second qubit only when the first is $|1\rangle$

Final state: $\frac{1}{\sqrt{2}}(|00\rangle + |11\rangle)$

Controlled not gate:



https://learn.microsoft.com/en-us/azure/quantum/concepts-circuits

No-Cloning

No-Cloning Theorem

It is **not** possible to copy an **arbitrary unknown** quantum state

This contrasts with classical information, which can be duplicated freely

Implications

Reinforces quantum information’s fragility

Enables secure communication (e.g., quantum key distribution)

Forces new strategies:

entanglement, teleportation, error correction

Schrödinger vs Classical Logic Flow

- **Classical logic:** follows a branching flow — conditionals like if/else determine discrete paths through a program.
- **Quantum logic:** evolves smoothly and reversibly via Schrödinger's equation; all possibilities evolve in superposition.
- **Measurement:** collapses the superposition into a definite classical state — the only irreversible step.

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

Quantum evolution is continuous, reversible, and governed by unitary transformations.

Building a Superposition Circuit

Let’s walk through how we build a quantum state that explores many possibilities in parallel.

We always begin with qubits in the $|0\rangle$ state. These are like classical bits set to 0.

By applying a Hadamard gate to each qubit, we turn each $|0\rangle$ into $\frac{1}{\sqrt{2}}(|0\rangle + |1\rangle).$

If we do this to *n* qubits, we get a superposition over all 2^n possible combinations.

Mathematically, this is written as: $H^{\otimes n}|0\rangle^{\otimes n} = \frac{1}{\sqrt{2^n}} \sum_{x=0}^{2^n-1} |x\rangle$ Each $|x\rangle$ represents a possible bitstring.

Summary: Thinking in Quantum Logic

Qubits can be in superpositions: not just 0 or 1, but both

Operations are reversible and unitary (no information is lost)

Entanglement creates strong correlations beyond classical limits

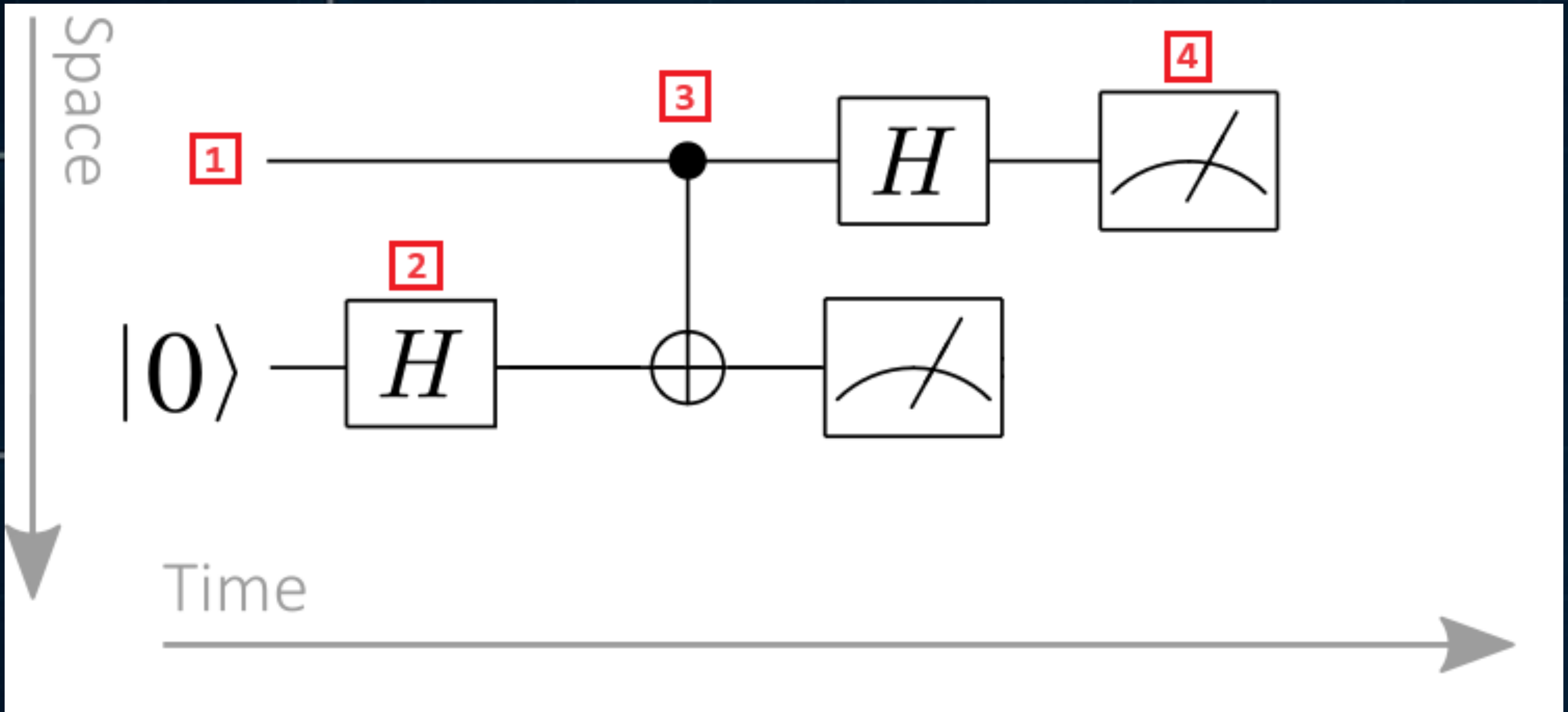
Measurement collapses a quantum state to a classical outcome

No cloning: we can’t copy unknown quantum states

Quantum computation uses interference, not branching, to find answers

What Is Entanglement?

- Entanglement links qubits so their states are no longer independent
- Measuring one immediately affects the other, no matter the distance
- Entangled states cannot be described as separate parts
- Enables key quantum protocols: teleportation, superdense coding, QKD



JOHN STEWART BELL
1928–1990
Northern Irish Physicist

$$\left|E(a,b)+E(a,b')+E(a',b)\right| \leq 2$$

CERN

This simple circuit creates a Bell state—an entangled pair of qubits. Measuring one qubit immediately determines the result of the other, no matter how far apart they are. It's the foundation of quantum teleportation, superdense coding, and quantum key distribution.

Meet the Bell States

Meet the Bell States

The four maximally entangled two-qubit states:

$$\Phi^+ = \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle)$$

$$\Phi^- = \frac{1}{\sqrt{2}}(|00\rangle - |11\rangle)$$

$$\Psi^+ = \frac{1}{\sqrt{2}}(|01\rangle + |10\rangle)$$

$$\Psi^- = \frac{1}{\sqrt{2}}(|01\rangle - |10\rangle)$$

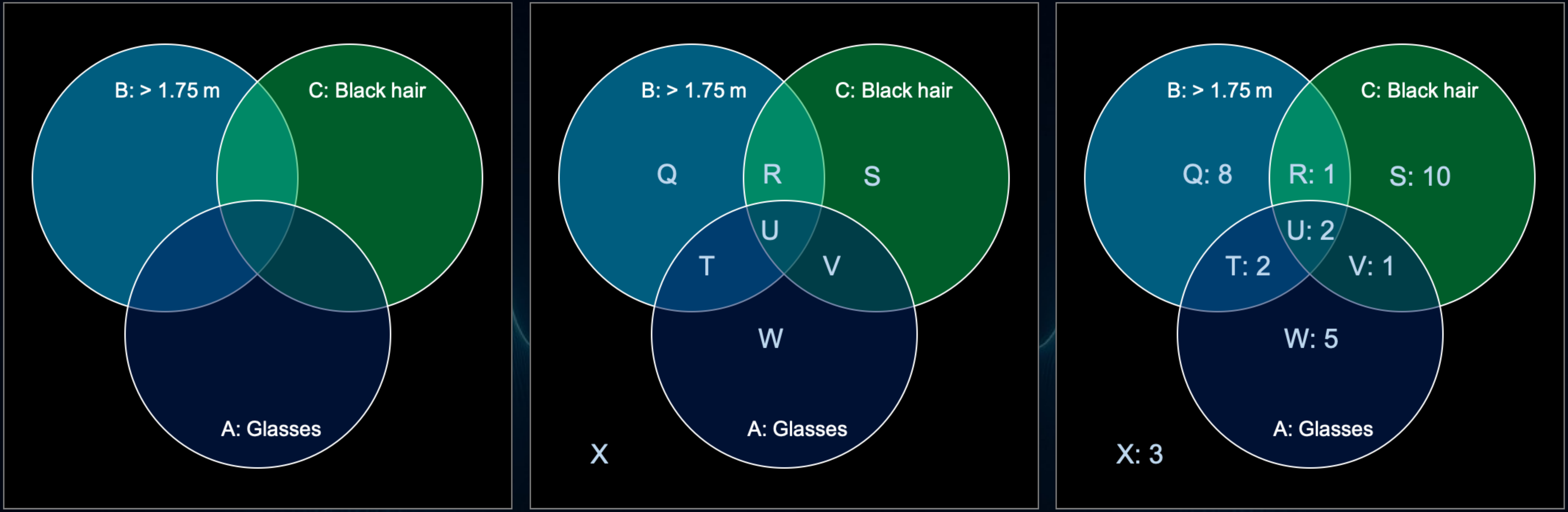
Now we are dealing with two qubits

$$|10\rangle$$

Means one qubit is in state [0,1] and the other is in state [1,0]

- Perfect correlation or anti-correlation when measured
- Basis for quantum communication, teleportation, and error correction

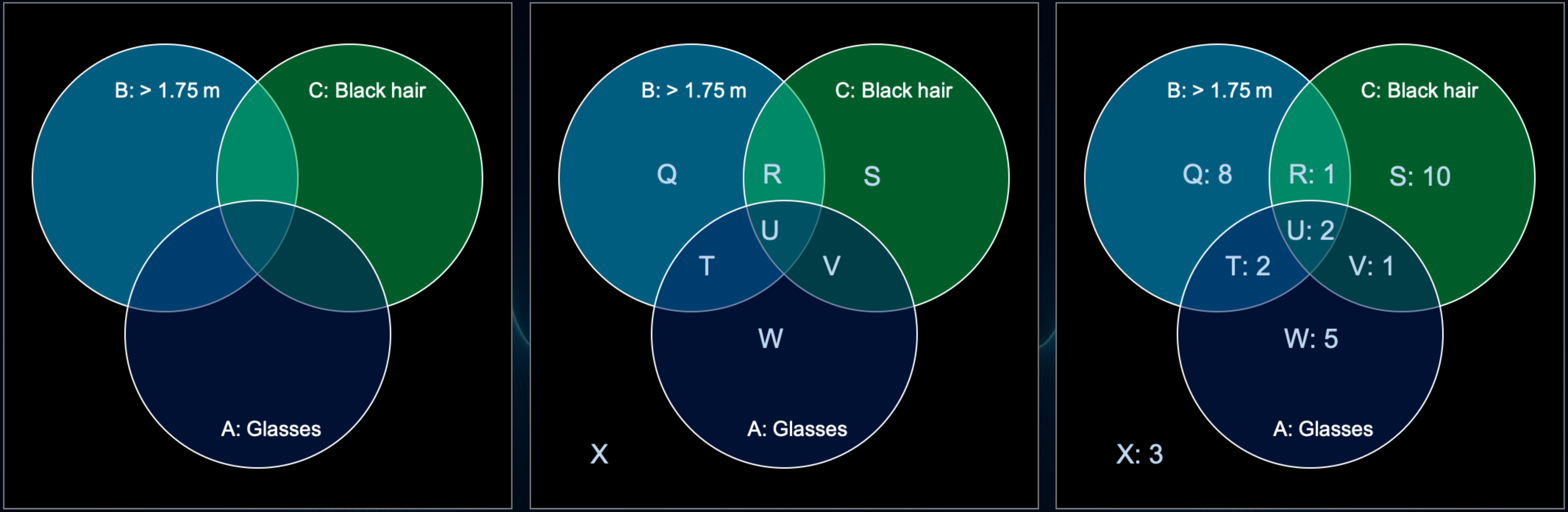
Bell’s Theorem



The number of people who are in A but not B, plus the number who are in B but not C, must be greater than or equal to the number who are in A but not C.

https://blog.qutech.nl/2020/11/27/entanglement-and-bell-inequalities/

Bell’s Theorem



The number of people who **wear glasses but are shorter than 1.75 m**, plus those who are **taller than 1.75 m but don’t have black hair**, should be **at least as many** as those who **wear glasses and don’t have black hair**.

https://blog.qutech.nl/2020/11/27/entanglement-and-bell-inequalities/

Bell's Theorem: Spooky Action at a Distance

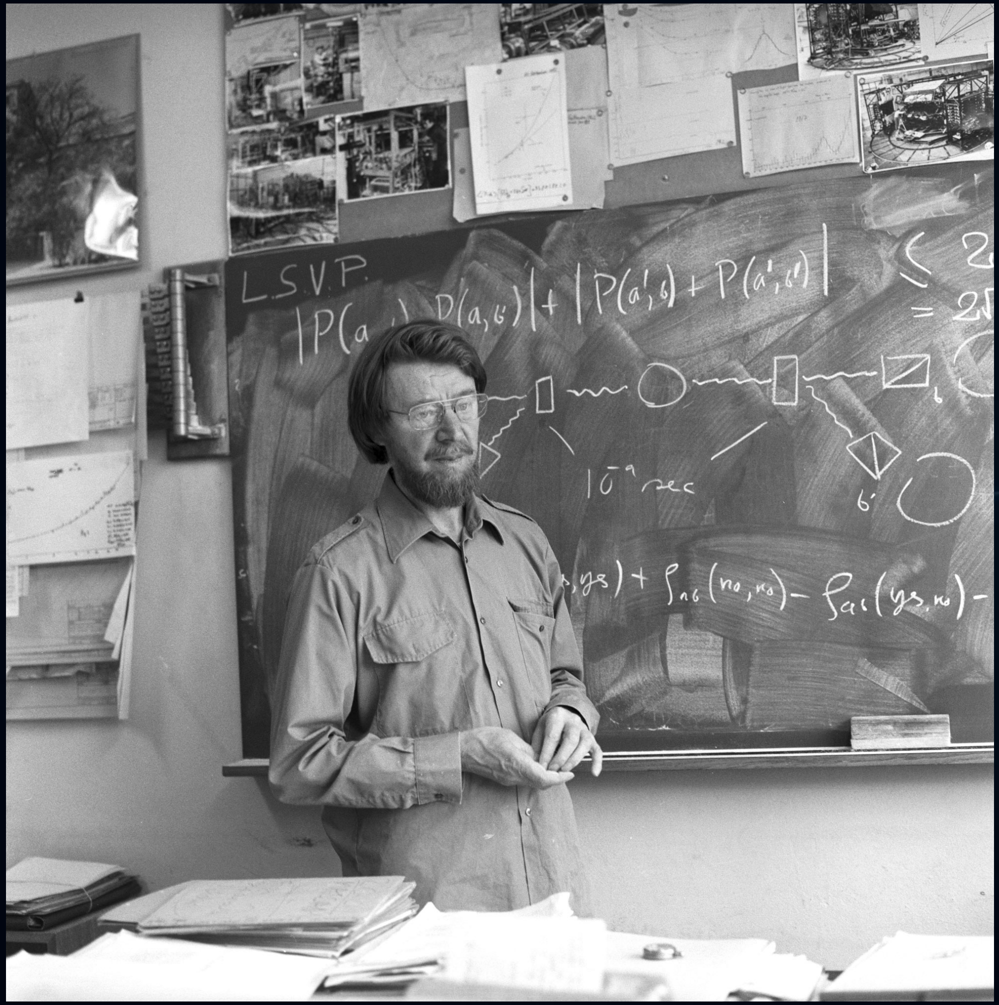
- Classical physics assumes:
  - **Locality**: no influence travels faster than light
  - **Realism**: properties exist before measurement
- Bell's theorem shows that quantum predictions can violate inequalities based on these assumptions
- **Experiments confirm:** entangled particles do not follow classical rules
- Quantum mechanics is either **nonlocal**, **non-real**, or both

$E(a,b) = \langle A(a) \cdot B(b) \rangle$ in the classical world where E ranges from -1 (anticorrelated) to +1 (perfect correlation).

$$\text{Bell's Inequality:} \quad |E(a,b) - E(a,b')| + |E(a',b) + E(a',b')| \leq 2$$

Quantum mechanics can produce results up to $2\sqrt{2}$

Bell Inequality in Action



John Clauser, b 1942
Lawrence BerkleyLabs and UC Berkley,
American

- Entangled photon pairs sent to Alice and Bob
- Each chooses random detector angles: a, a', b, b'
- Compute correlations:

$$E(a,b), \quad E(a,b'), \quad E(a',b), \quad E(a',b')$$

- CHSH*inequality:

$$S = |E(a,b) - E(a,b')| + |E(a',b) + E(a',b')| \leq 2$$

- **Classical physics:** $S \leq 2$
- **Quantum mechanics:** $S \leq 2\sqrt{2} \approx 2.828$
- **Experimental results:** $S \approx 2.4$–2.8
- Confirms violation of classical realism and locality

*1969, John **Clauser**, Michael **Horne**, Abner **Shimony**, and Richard **Holt** developed a more **experimentally testable** version inequality.

One or more of the following assumptions is wrong…

- **Reject Locality**
→ Maybe particles communicate faster than light.
- **Reject Reality**
→ Particles don’t have definite properties until measured.
- **Reject Free Will (Superdeterminism)**
→ Everything—including our measurement choices—was predetermined; the universe is fully scripted.

Giustina, M. et al. Significant-loophole-free test of Bell’s theorem with entangled photons. Phys. Rev. Lett. 2015, Vol. 115, 25040.38.

Entanglement vs Classical Correlation

	Classical Correlation	**Quantum Entanglement**
Origin	Shared cause or common information	Joint quantum preparation (e.g. Bell state)
Description	Probabilistic — based on known values	No classical description — shared quantum state
Measurement Effect	Measuring one doesn’t affect the other	Measuring one collapses the state of the other
Copyable	Yes, information can be cloned	No — copying violates the no-cloning theorem
Limitations	Can’t violate Bell’s inequality	Violates Bell’s inequality (experimentally confirmed)
Example	Two people given matching socks either blue or red. Knowing the colour or one sock implies the colour of the other one	$\frac{1}{\sqrt{2}}(\lvert 00\rangle + \lvert 11\rangle)$

The Quantum Edge

•**Superposition**
→ A quantum system can explore many states at once

•**Entanglement**
→ Correlations stronger than anything classically possible

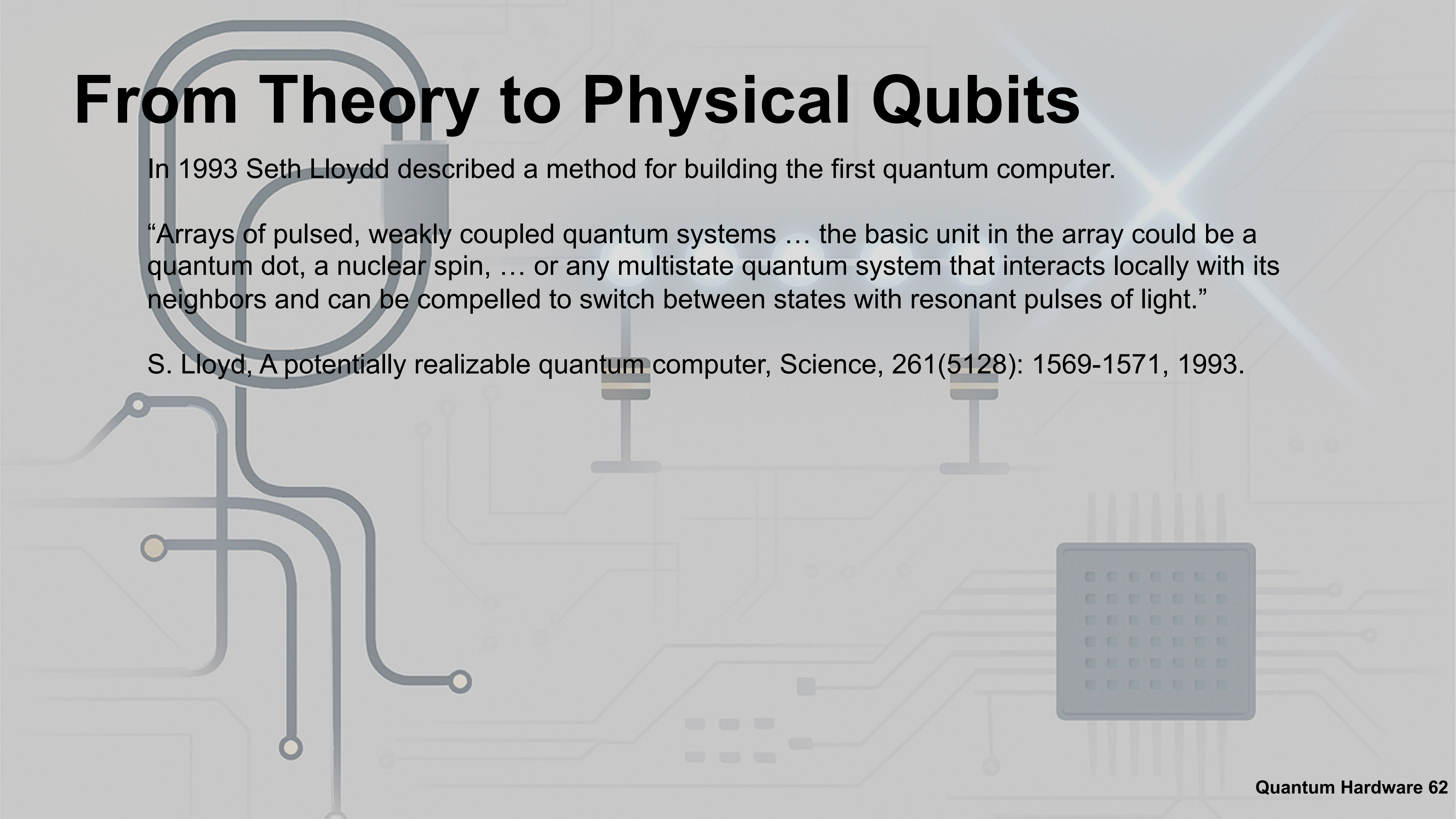
•**Interference**
→ Amplifies correct paths, cancels the wrong ones

From Theory to Physical Qubits

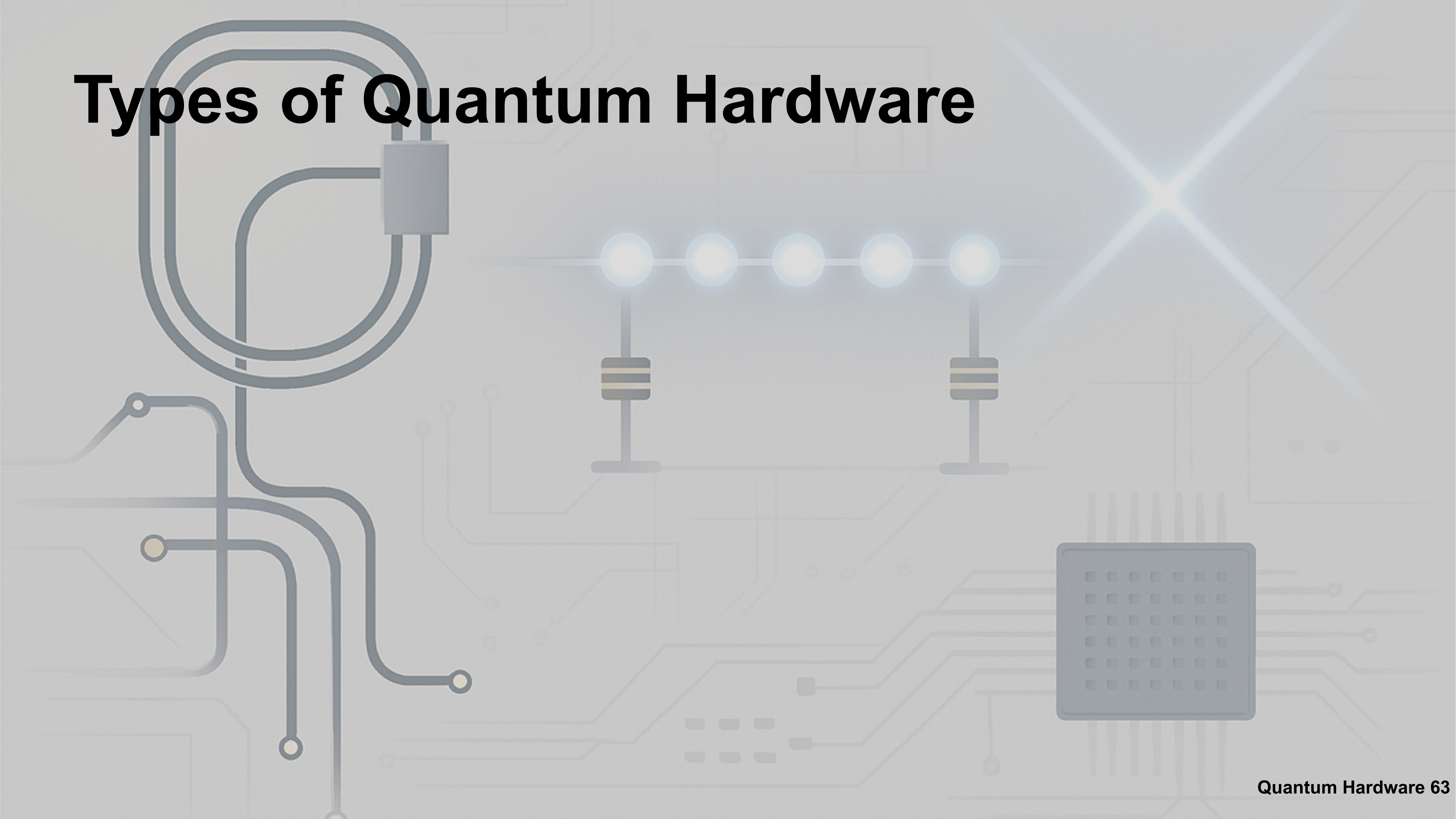
In 1993 Seth Lloydd described a method for building the first quantum computer.

“Arrays of pulsed, weakly coupled quantum systems … the basic unit in the array could be a quantum dot, a nuclear spin, … or any multistate quantum system that interacts locally with its neighbors and can be compelled to switch between states with resonant pulses of light.”

S. Lloyd, A potentially realizable quantum computer, Science, 261(5128): 1569-1571, 1993.



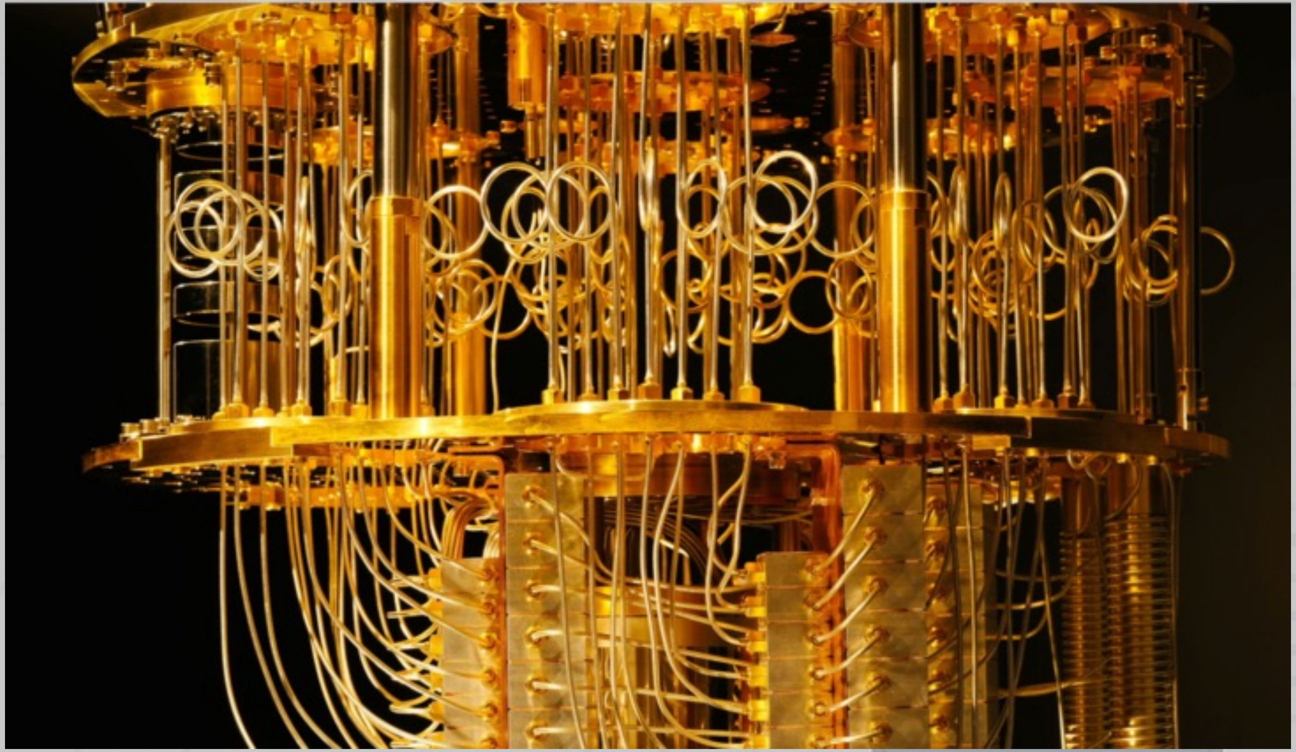
Types of Quantum Hardware



Superconducting Qubits

- **Description:** Utilize superconducting circuits at cryogenic temperatures to perform quantum operations.

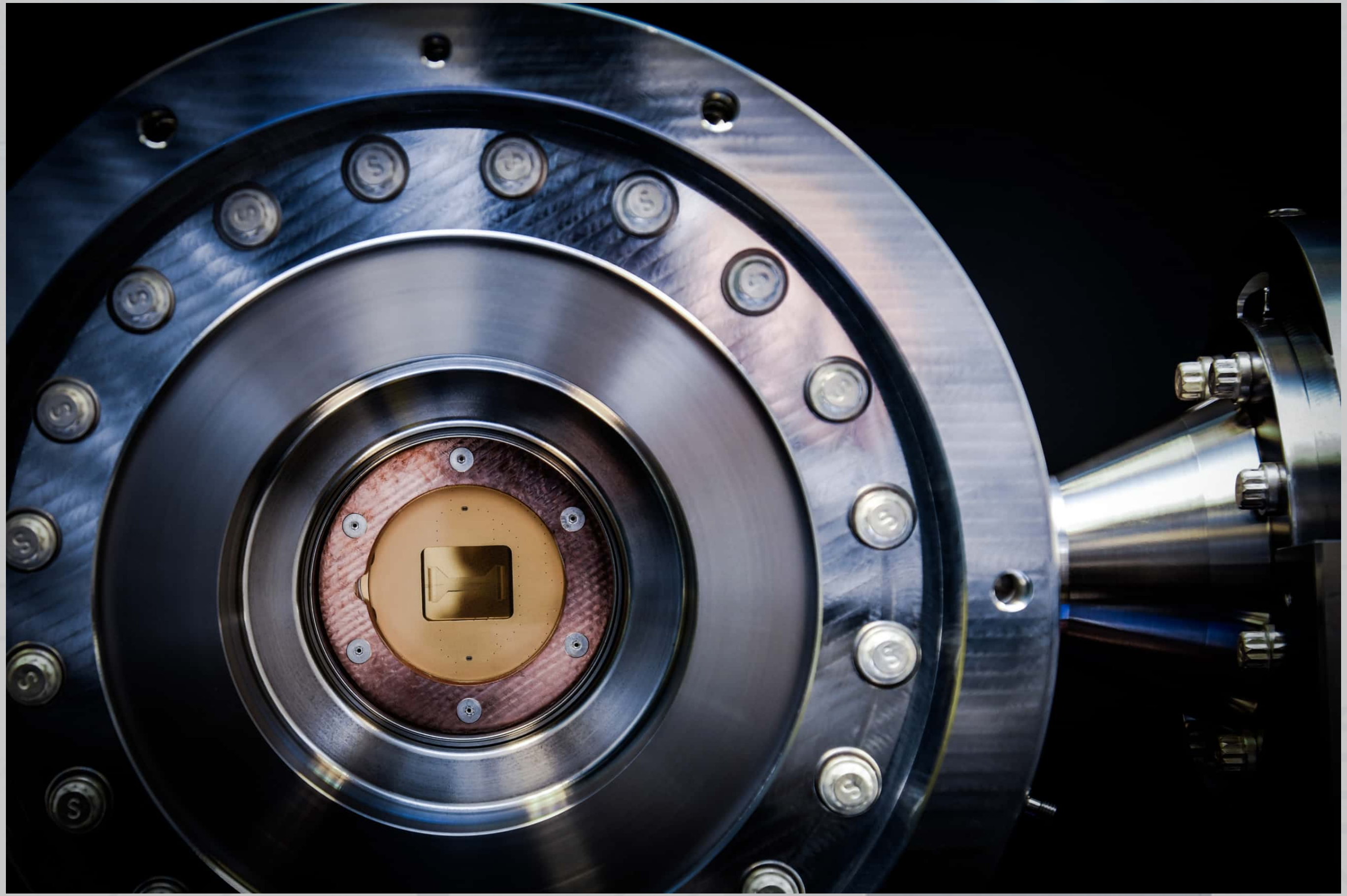
- **Notable Companies:**
- **IBM**
- **Google**
- **Rigetti Computing**
- **Intel**



Superconducting quantum processor suspended in a cryostat for ultra-cold operation and precise qubit measurement.

Trapped Ion Qubits

- **Description:** Employ ions confined and manipulated using electromagnetic fields.
- **Notable Companies:**
- **IonQ**
- **Quantinuum**
- **Honeywell**

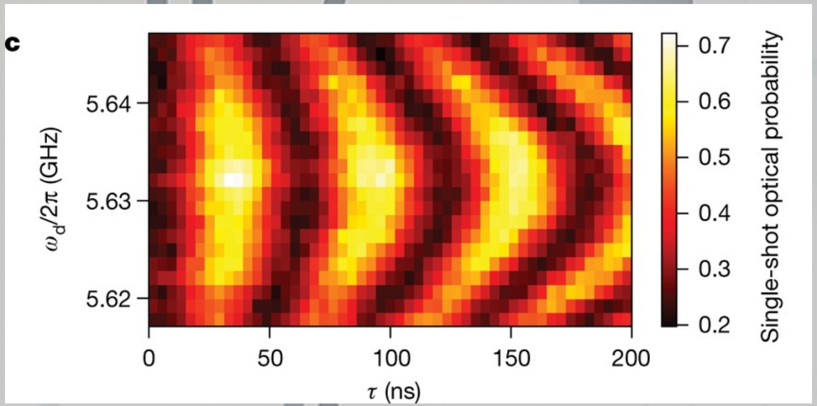


Ion trap quantum processor housed in an ultra-high vacuum chamber, used to isolate and control individual trapped ion qubits.

Measurement in Hardware

Superconducting qubits

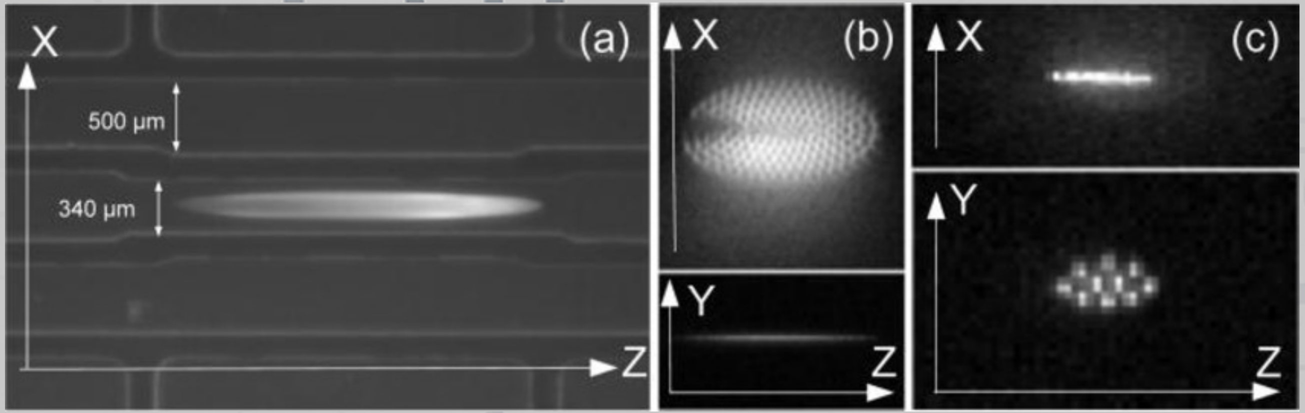
- Readout via resonator frequency shift (microwave cavity QED)



Delaney, R. D., et al. "Superconducting-qubit readout via low-backaction electro-optic transduction." *Nature* 606.7914 (2022): 489-493.

Trapped ions

- Fluorescence detection (ions emit or don’t under laser illumination)



Szymanski, B., et al. "Large 2D Coulomb crystals in a radio frequency surface ion trap." *arXiv preprint arXiv:1201.2584* (2012).

Other Quantum Computer Implementation

Photonic Qubits

Use photons for quantum computations, often at room temperature.

- **PsiQuantum**
- **Xanadu**
- **ORCA Computing**

Neutral Atom Qubits

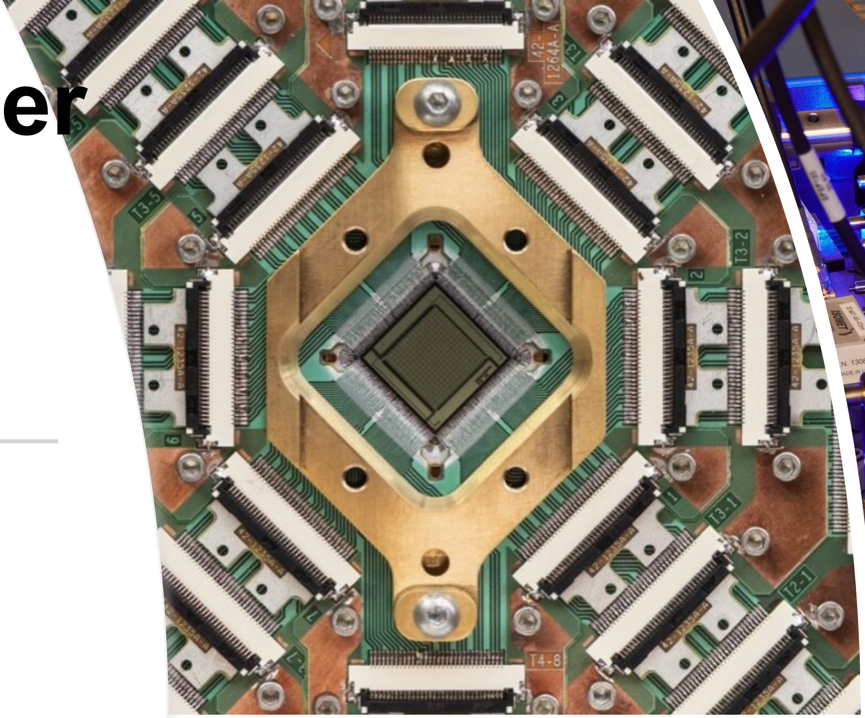
Neutral atoms trapped in optical lattices or tweezers.

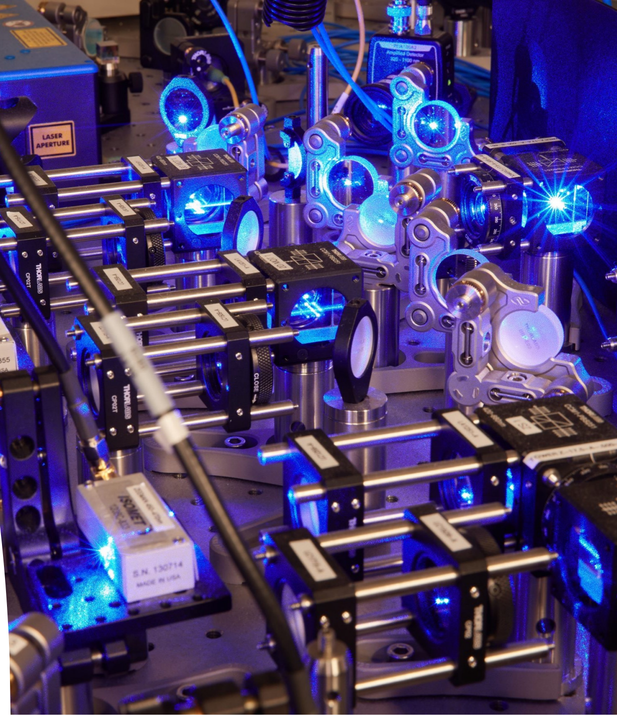
- **QuEra Computing**
- **ColdQuanta**
- **Atom Computing**

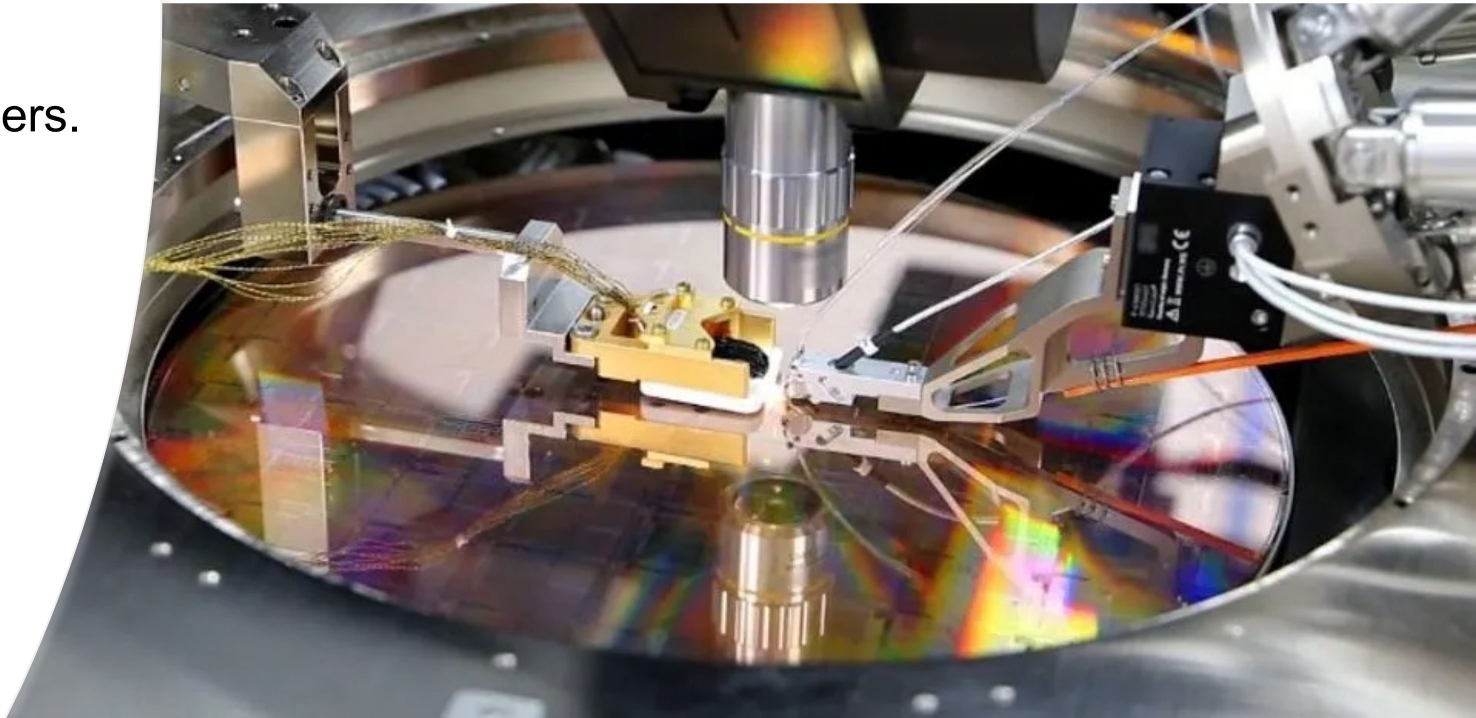
Quantum Annealers

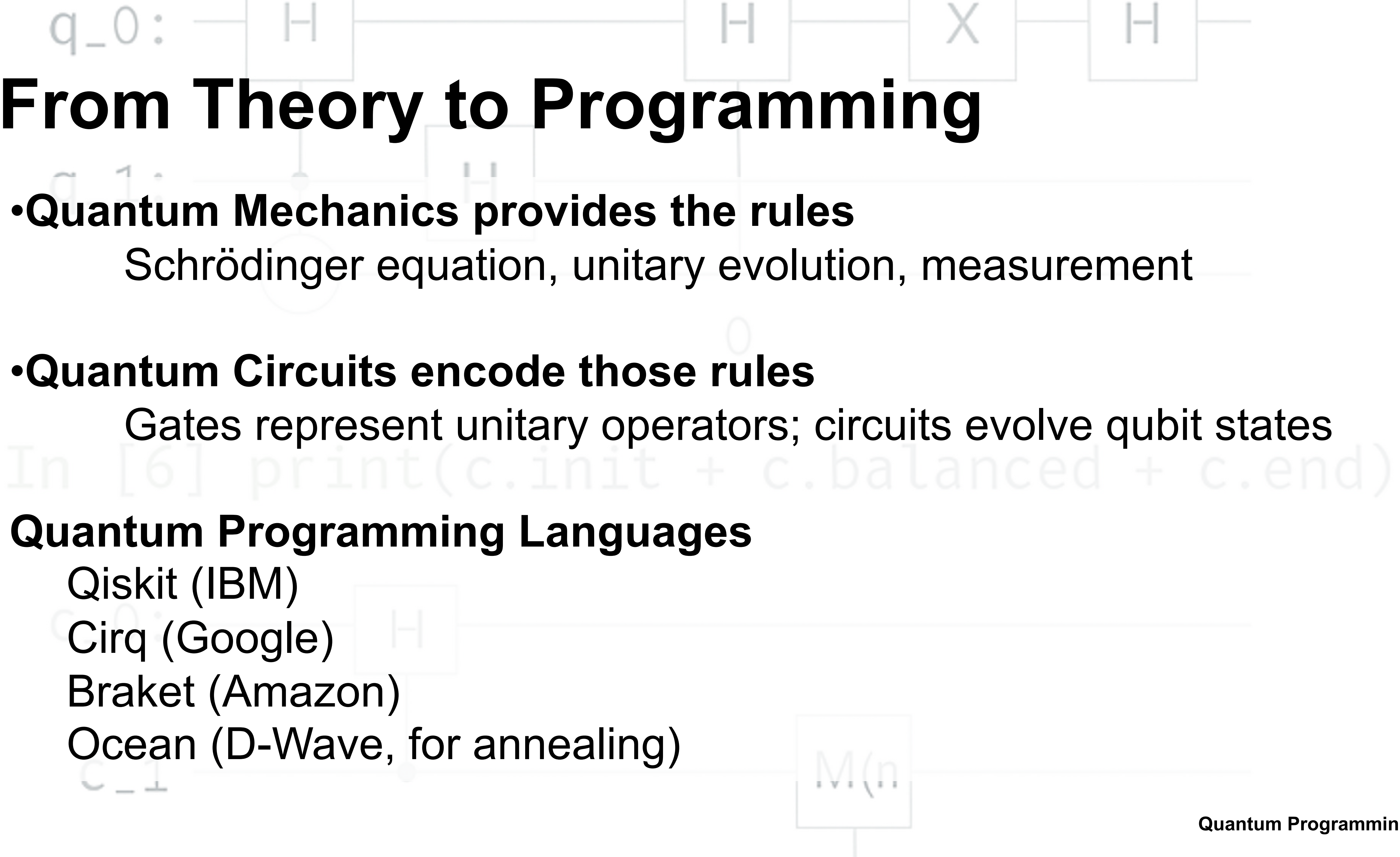
Specialized devices for optimization problems, leveraging quantum tunneling.

- **D-Wave Systems**









From Theory to Programming

- **Quantum Mechanics provides the rules**

  Schrödinger equation, unitary evolution, measurement

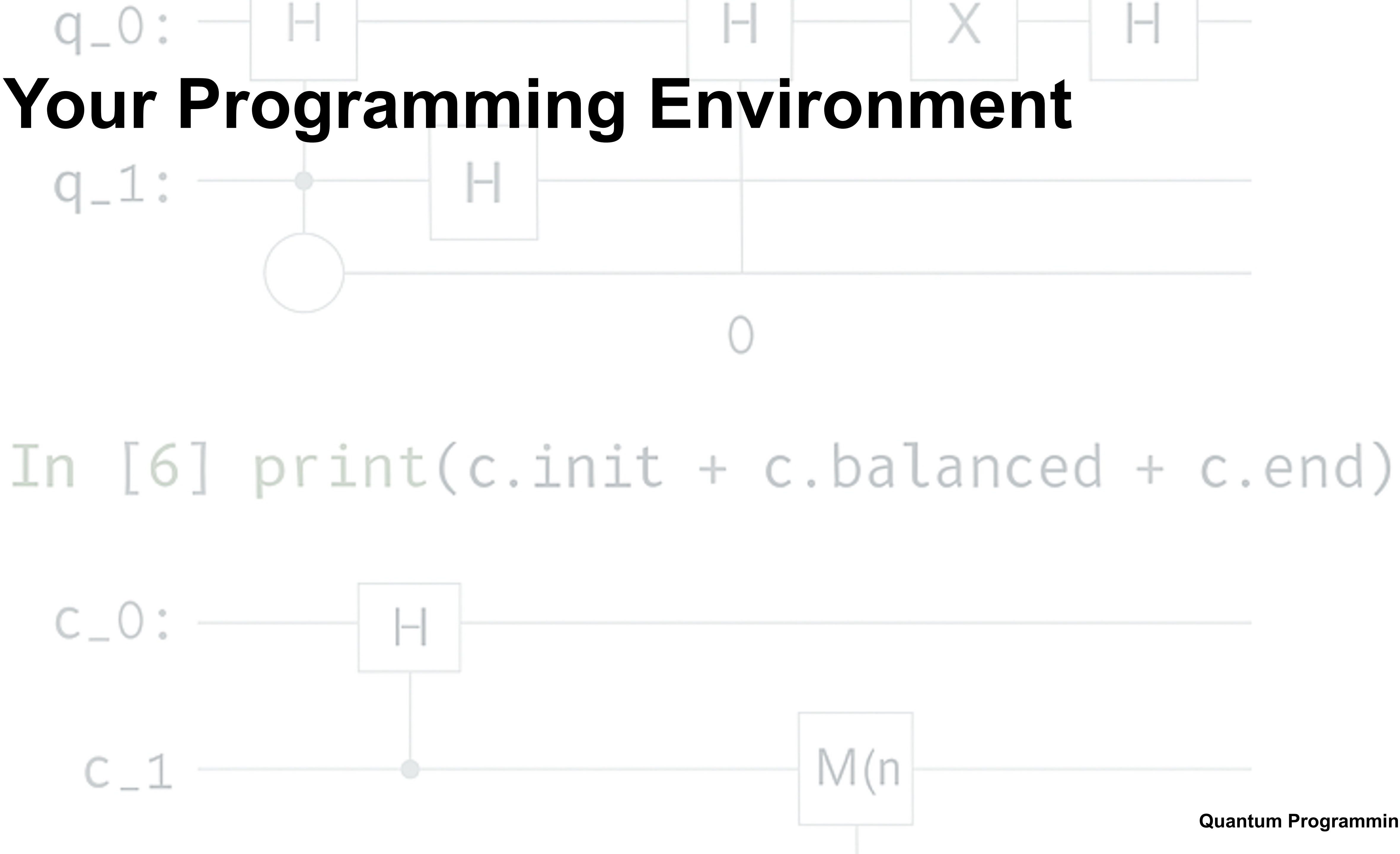
- **Quantum Circuits encode those rules**

  Gates represent unitary operators; circuits evolve qubit states

Quantum Programming Languages

Qiskit (IBM)
Cirq (Google)
Braket (Amazon)
Ocean (D-Wave, for annealing)

Your Programming Environment

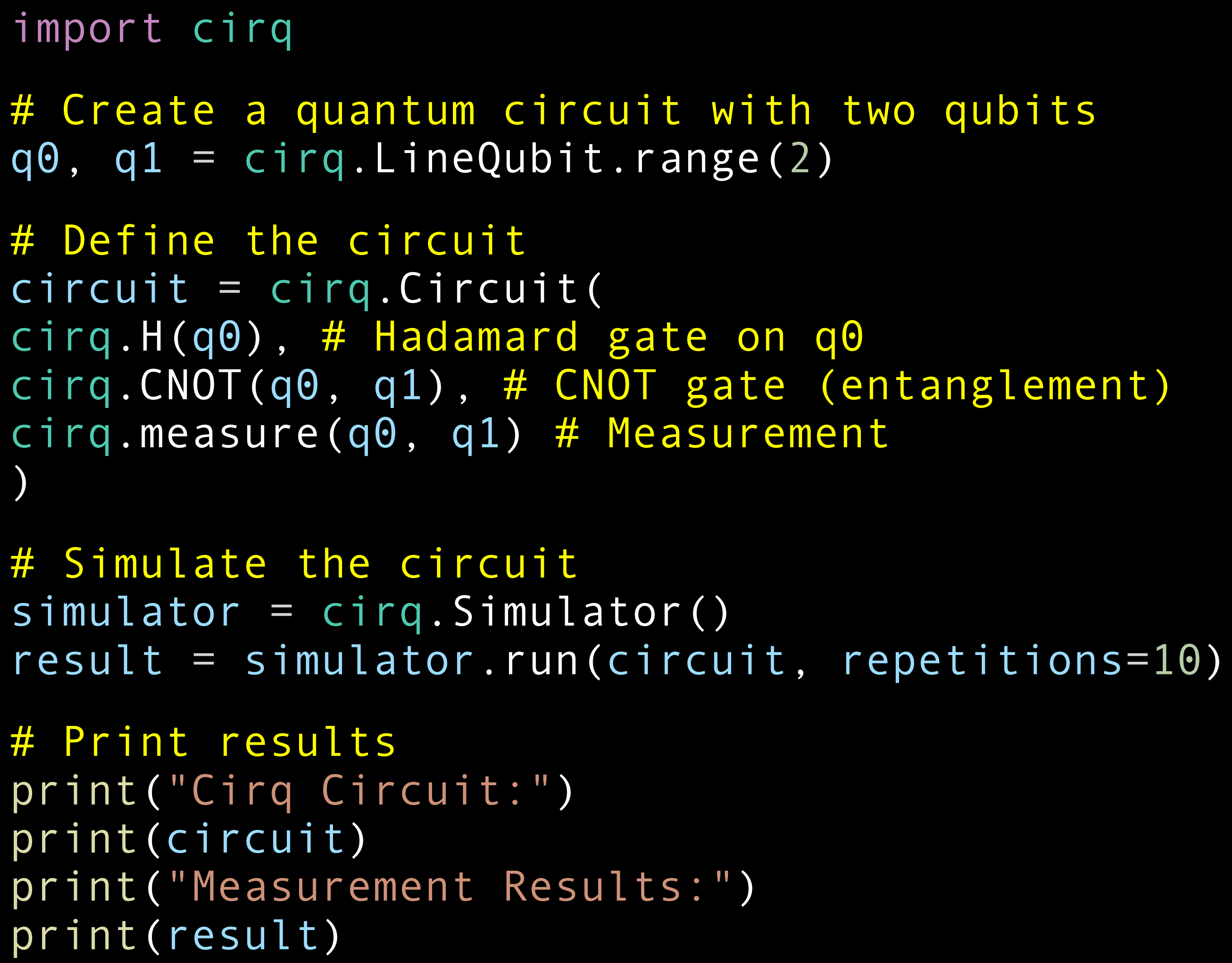


```
In [6] print(c.init + c.balanced + c.end)
```

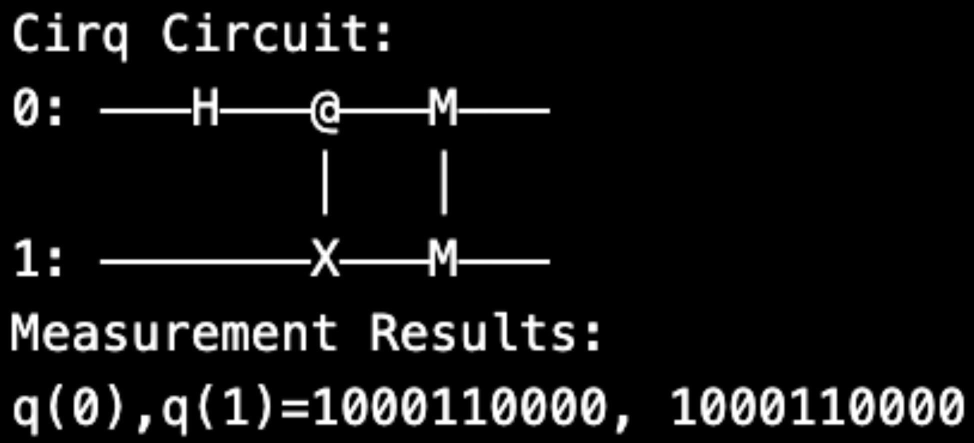
Qubits, Gates, and Circuits in Code (Cirq)

```python
import cirq

# Create a quantum circuit with two qubits
q0, q1 = cirq.LineQubit.range(2)

# Define the circuit
circuit = cirq.Circuit(
cirq.H(q0), # Hadamard gate on q0
cirq.CNOT(q0, q1), # CNOT gate (entanglement)
cirq.measure(q0, q1) # Measurement
)

# Simulate the circuit
simulator = cirq.Simulator()
result = simulator.run(circuit, repetitions=10)

# Print results
print("Cirq Circuit:")
print(circuit)
print("Measurement Results:")
print(result)
```

Qubits, Gates, and Circuits in Code (Cirq)



```python
import cirq

# Create a quantum circuit with two qubits
q0, q1 = cirq.LineQubit.range(2)

# Define the circuit
circuit = cirq.Circuit(
cirq.H(q0), # Hadamard gate on q0
cirq.CNOT(q0, q1), # CNOT gate (entanglement)
cirq.measure(q0, q1) # Measurement
)

# Simulate the circuit
simulator = cirq.Simulator()
result = simulator.run(circuit, repetitions=10)

# Print results
print("Cirq Circuit:")
print(circuit)
print("Measurement Results:")
print(result)
```



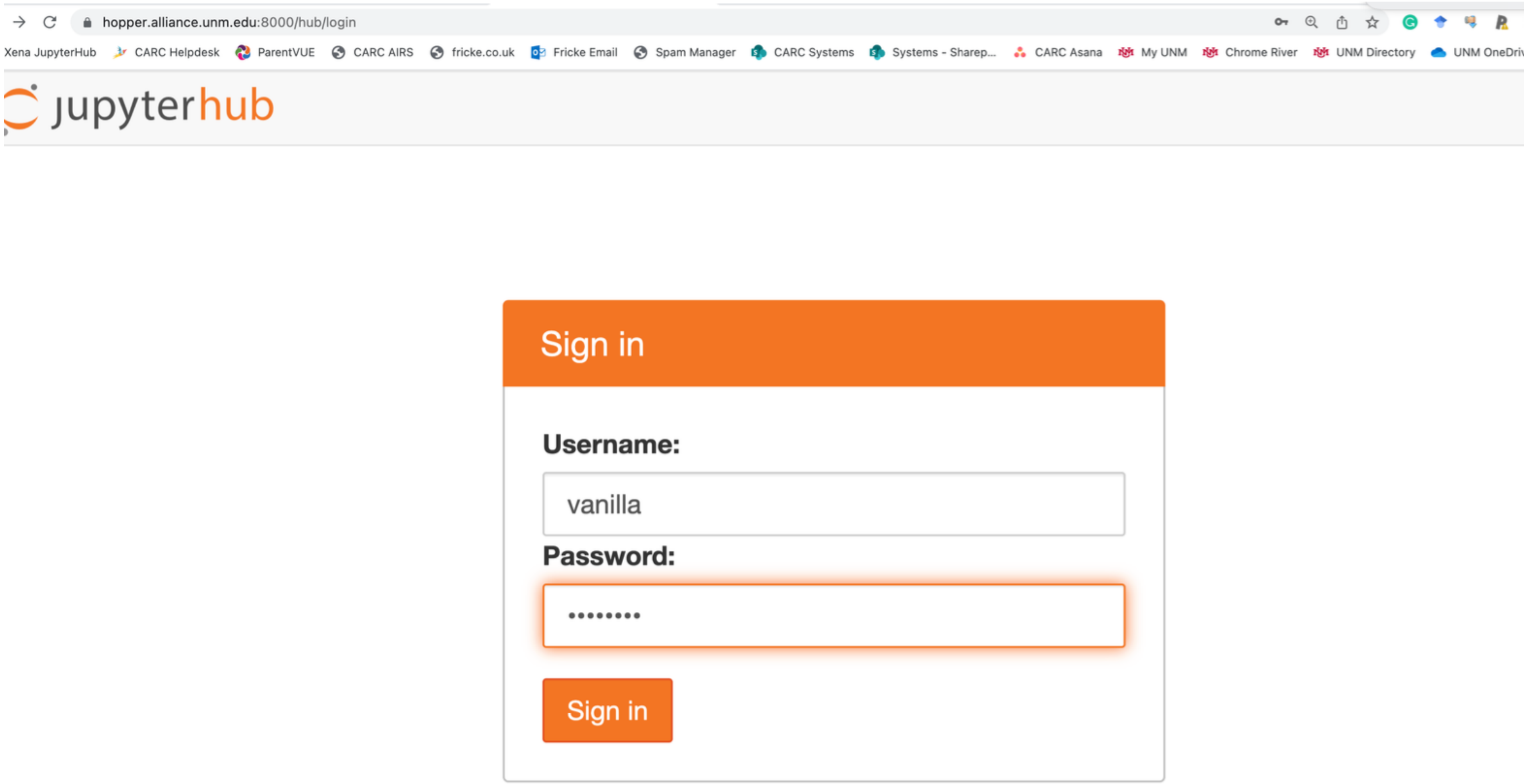
```
Cirq Circuit:
0: ───H───@───M───
          │   │
1: ───────X───M───
Measurement Results:
q(0),q(1)=1000110000, 1000110000
```

The outputs are the same for both qubits.

They are not communicating through wires or any other “real” connection.

Recall Bell’s three hypotheses…

Now it's your turn
Go to https://hopper.alliance.unm.edu:8000



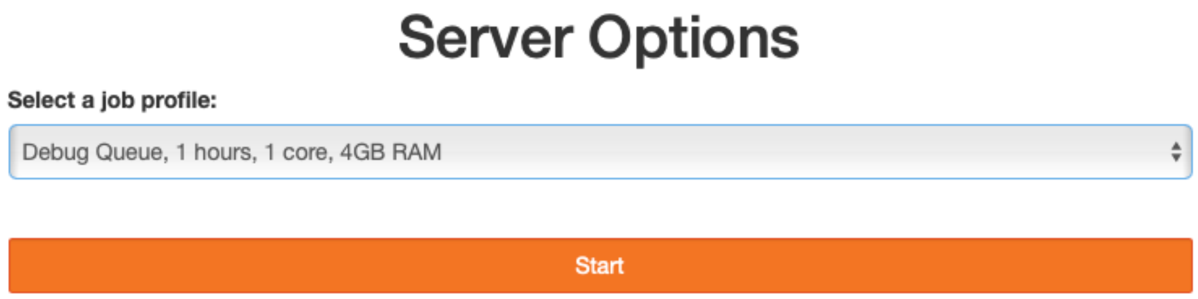
hopper.alliance.unm.edu:8000/hub/login

Sign in

Username: vanilla

Password: •••••••••

Now it’s your turn
Go to https://hopper.alliance.unm.edu:8000



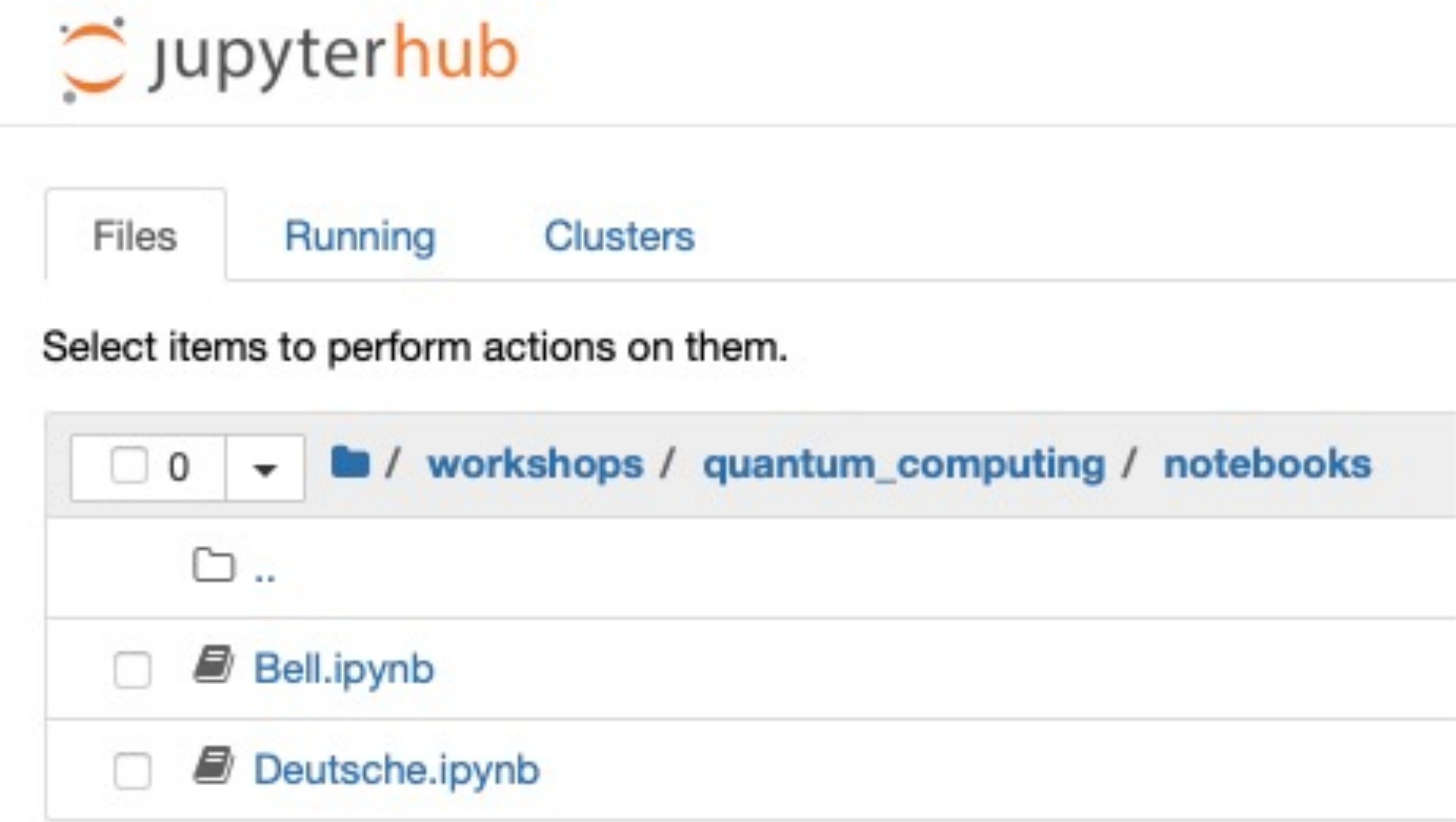
Server Options

Select a job profile:

Debug Queue, 1 hours, 1 core, 4GB RAM

Now it’s your turn

Go to https://hopper.alliance.unm.edu:8000



Select items to perform actions on them.

/ workshops / quantum_computing / notebooks

..

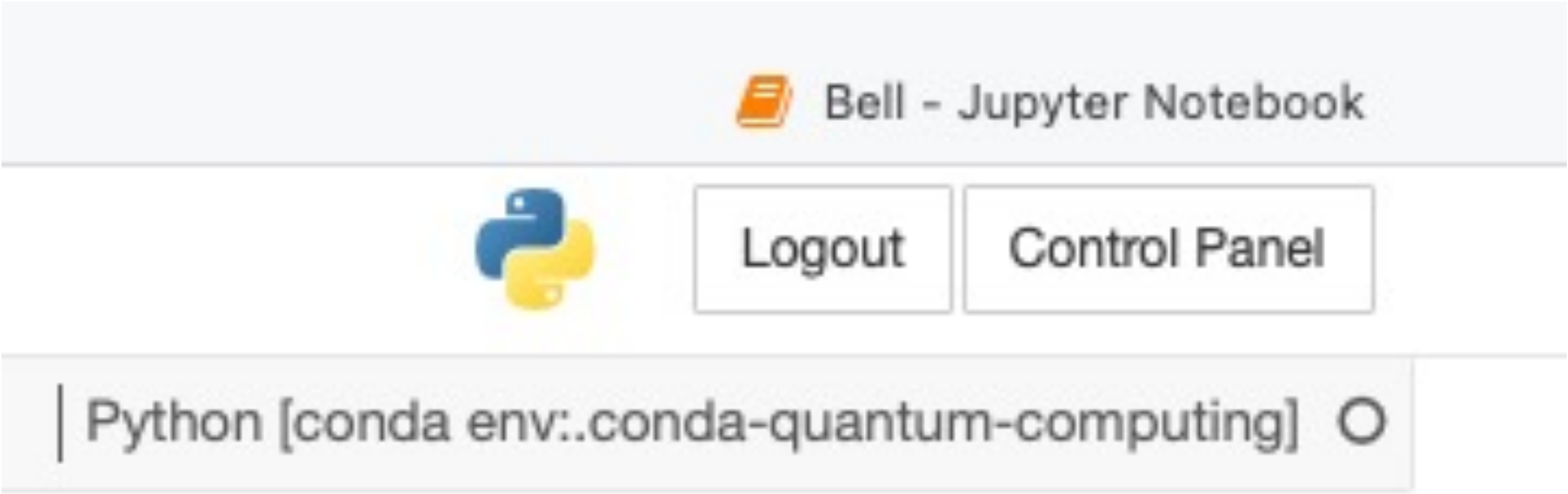
□ Bell.ipynb

□ Deutsche.ipynb

Click on Bell.ipynb

Now it’s your turn

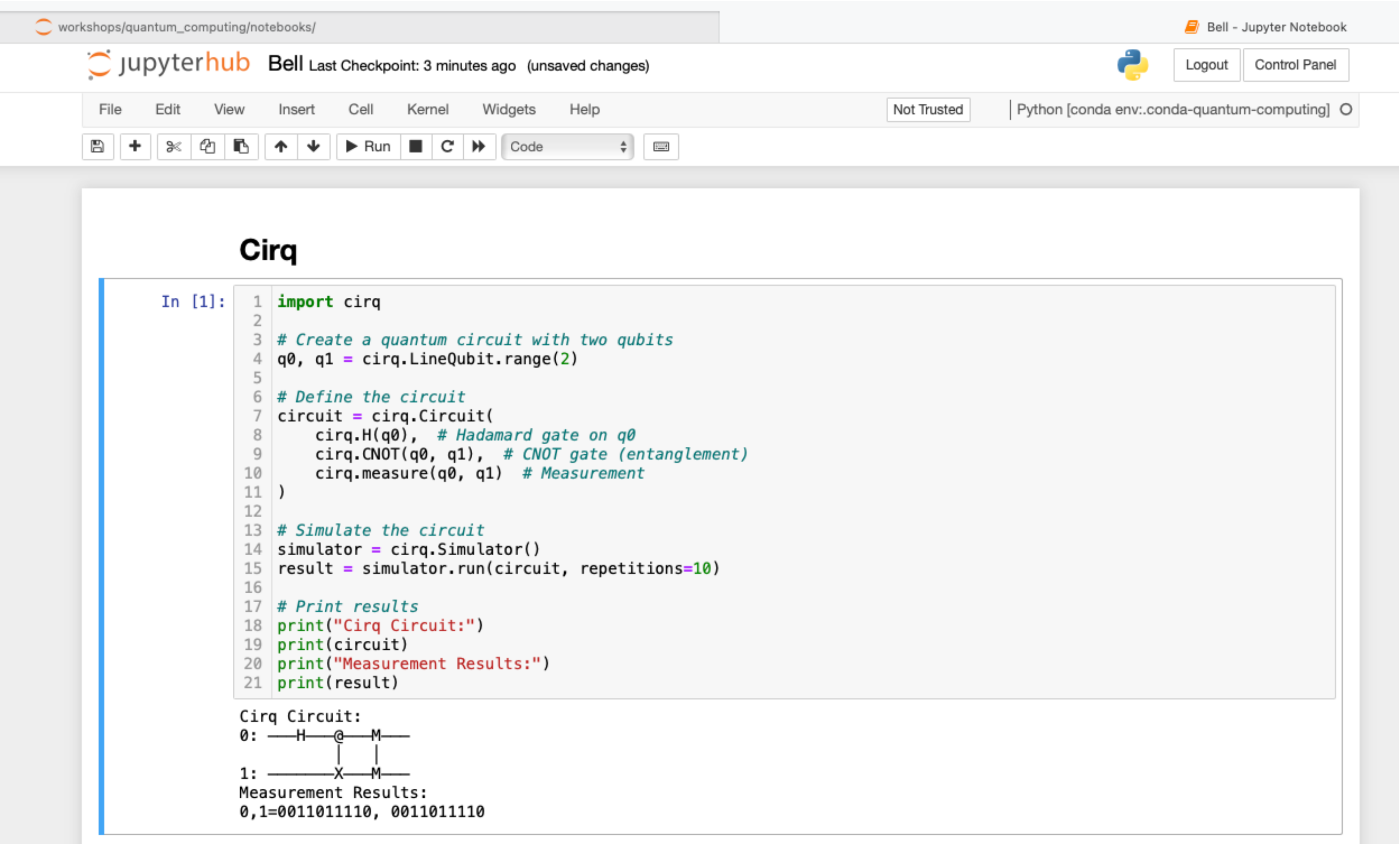
Go to https://hopper.alliance.unm.edu:8000



Python [conda env:.conda-quantum-computing] ○

Double check you have this kernel loaded

Now it’s your turn
Go to https://hopper.alliance.unm.edu:8000


workshops/quantum_computing/notebooks/
Bell - Jupyter Notebook
jupyterhub
Bell Last Checkpoint: 3 minutes ago (unsaved changes)
Logout
Control Panel
File
Edit
View
Insert
Cell
Kernel
Widgets
Help
Not Trusted
Python [conda env:.conda-quantum-computing]
Run
Code

Cirq

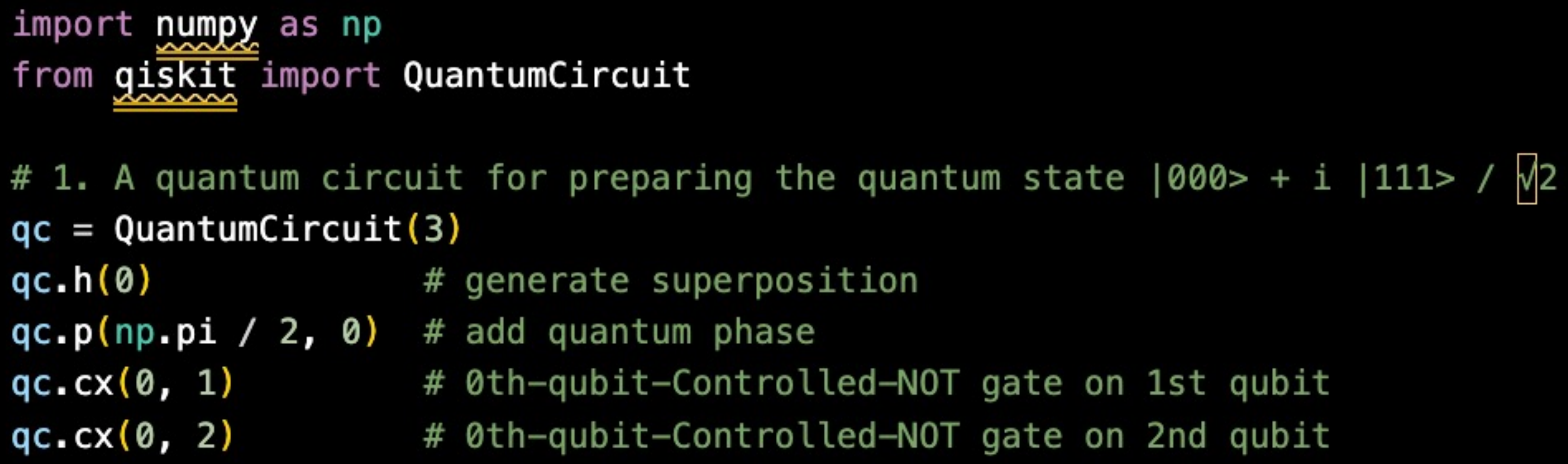
In [1]:

```python
import cirq

# Create a quantum circuit with two qubits
q0, q1 = cirq.LineQubit.range(2)

# Define the circuit
circuit = cirq.Circuit(
    cirq.H(q0),  # Hadamard gate on q0
    cirq.CNOT(q0, q1),  # CNOT gate (entanglement)
    cirq.measure(q0, q1)  # Measurement
)

# Simulate the circuit
simulator = cirq.Simulator()
result = simulator.run(circuit, repetitions=10)

# Print results
print("Cirq Circuit:")
print(circuit)
print("Measurement Results:")
print(result)
```

```
Cirq Circuit:
0: ───H───@───M───
          │   │
1: ───────X───M───
Measurement Results:
0,1=0011011110, 0011011110
```

Qubits, Gates, and Circuits in Code (qiskit)

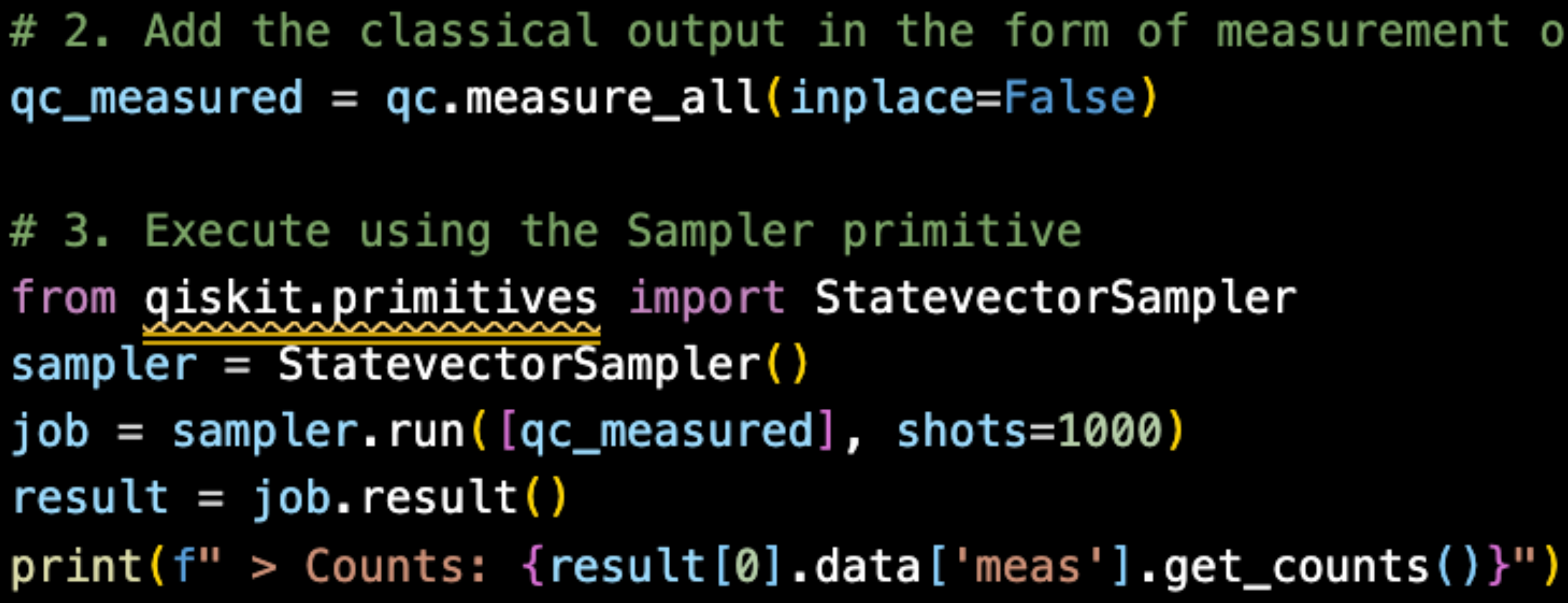
Now we will entangle three qubits with qiskit and then measure their correlation.



```
import numpy as np
from qiskit import QuantumCircuit

# 1. A quantum circuit for preparing the quantum state |000> + i |111> / √2
qc = QuantumCircuit(3)
qc.h(0)             # generate superposition
qc.p(np.pi / 2, 0)  # add quantum phase
qc.cx(0, 1)         # 0th-qubit-Controlled-NOT gate on 1st qubit
qc.cx(0, 2)         # 0th-qubit-Controlled-NOT gate on 2nd qubit
```

```
<qiskit.circuit.instructionset.InstructionSet at 0x147a396db910>
```

Qubits, Gates, and Circuits in Code (qiskit)

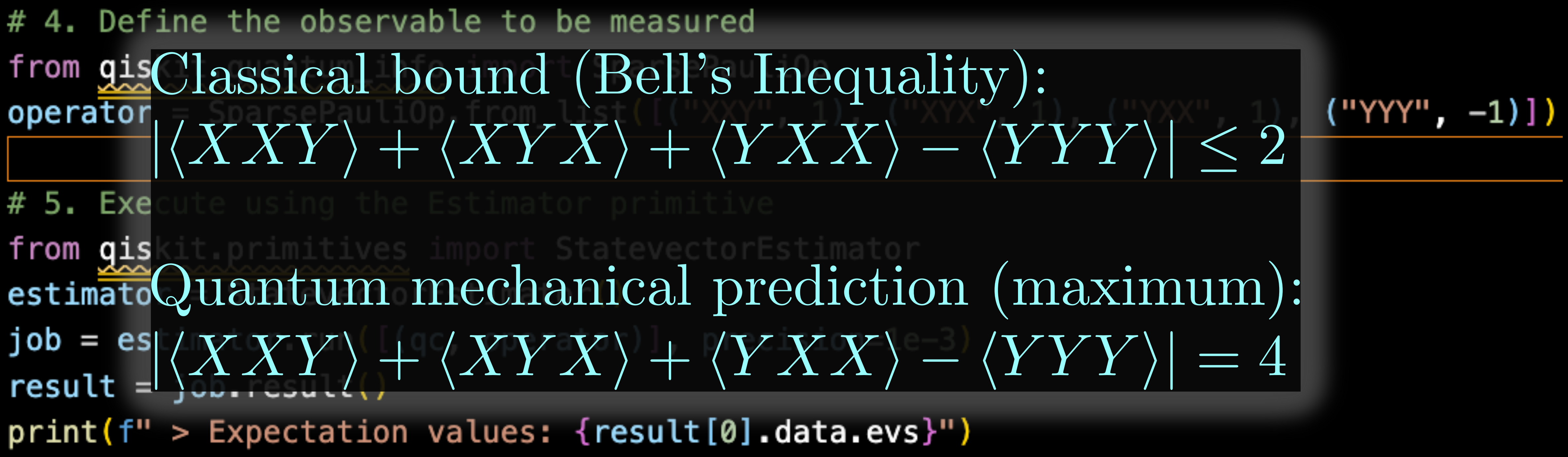


```
# 2. Add the classical output in the form of measurement of all qubits
qc_measured = qc.measure_all(inplace=False)

# 3. Execute using the Sampler primitive
from qiskit.primitives import StatevectorSampler
sampler = StatevectorSampler()
job = sampler.run([qc_measured], shots=1000)
result = job.result()
print(f" > Counts: {result[0].data['meas'].get_counts()}")
```

```
> Counts: {'111': 508, '000': 492}
```

The **StatevectorEstimator** computes properties **analytically** from the full quantum state, not just from sampling.

Qubits, Gates, and Circuits in Code (qiskit)



```
# 4. Define the observable to be measured
from qiskit.quantum_info import SparsePauliOp
operator = SparsePauliOp.from_list([("XXY", 1), ("XYX", 1), ("YXX", 1), ("YYY", -1)])

# 5. Execute using the Estimator primitive
from qiskit.primitives import StatevectorEstimator
estimator = StatevectorEstimator()
job = estimator.run([(qc, operator)], precision=1e-3)
result = job.result()
print(f" > Expectation values: {result[0].data.evs}")
```

Classical bound (Bell's Inequality):

$$|\langle XXY\rangle + \langle XYX\rangle + \langle YXX\rangle - \langle YYY\rangle| \leq 2$$

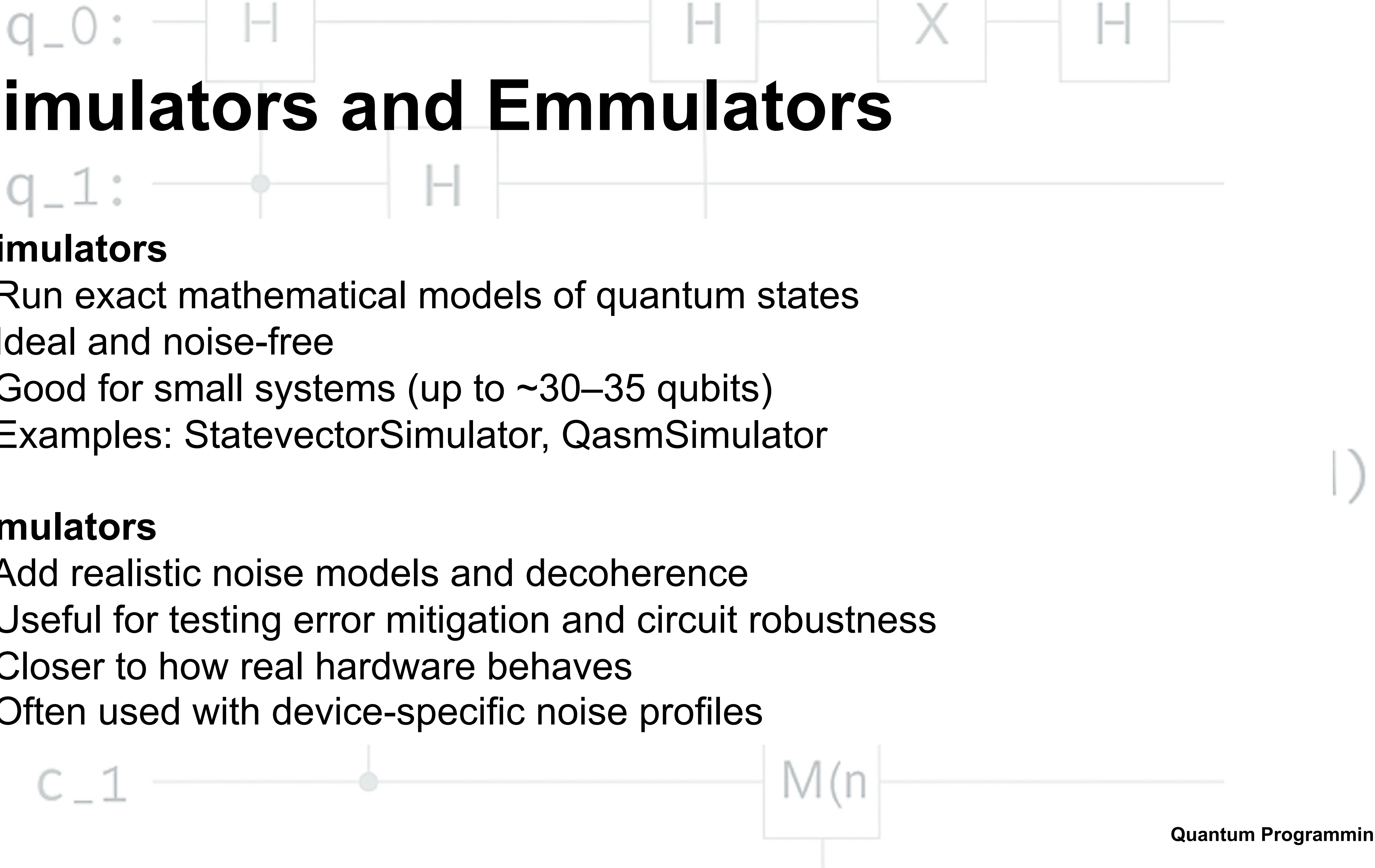
Quantum mechanical prediction (maximum):

$$|\langle XXY\rangle + \langle XYX\rangle + \langle YXX\rangle - \langle YYY\rangle| = 4$$

```
> Expectation values: 4.000376620708558
```

Slightly more than 4 due to numerical instability. This tells us it was run on a simulator.
On a real quantum computer, you would get slightly less than 4.

Simulators and Emmulators



•Simulators
•• Run exact mathematical models of quantum states
•• Ideal and noise-free
•• Good for small systems (up to ~30–35 qubits)
•• Examples: StatevectorSimulator, QasmSimulator

•Emulators
•• Add realistic noise models and decoherence
•• Useful for testing error mitigation and circuit robustness
•• Closer to how real hardware behaves
•• Often used with device-specific noise profiles

How to Access Real Quantum Devices

IBM Quantum (Qiskit)
Sign up at quantum-computing.ibm.com
Free tier includes several 5–7 qubit devices
Use IBMProvider to submit jobs via Qiskit

Amazon Braket (AWS)
Access devices from IonQ, OQC, Rigetti
Requires AWS account and setup of a Braket notebook instance
Pay-as-you-go pricing

Google (Cirq)
Access to real hardware is limited
Most public Cirq usage is simulator-based
Quantum Computing Service access is via partnerships or Google collaborators

Quantinuum, Xanadu, QuEra, etc.
Access often through API keys or cloud portals
Some offer free trials, most are enterprise or research-facing

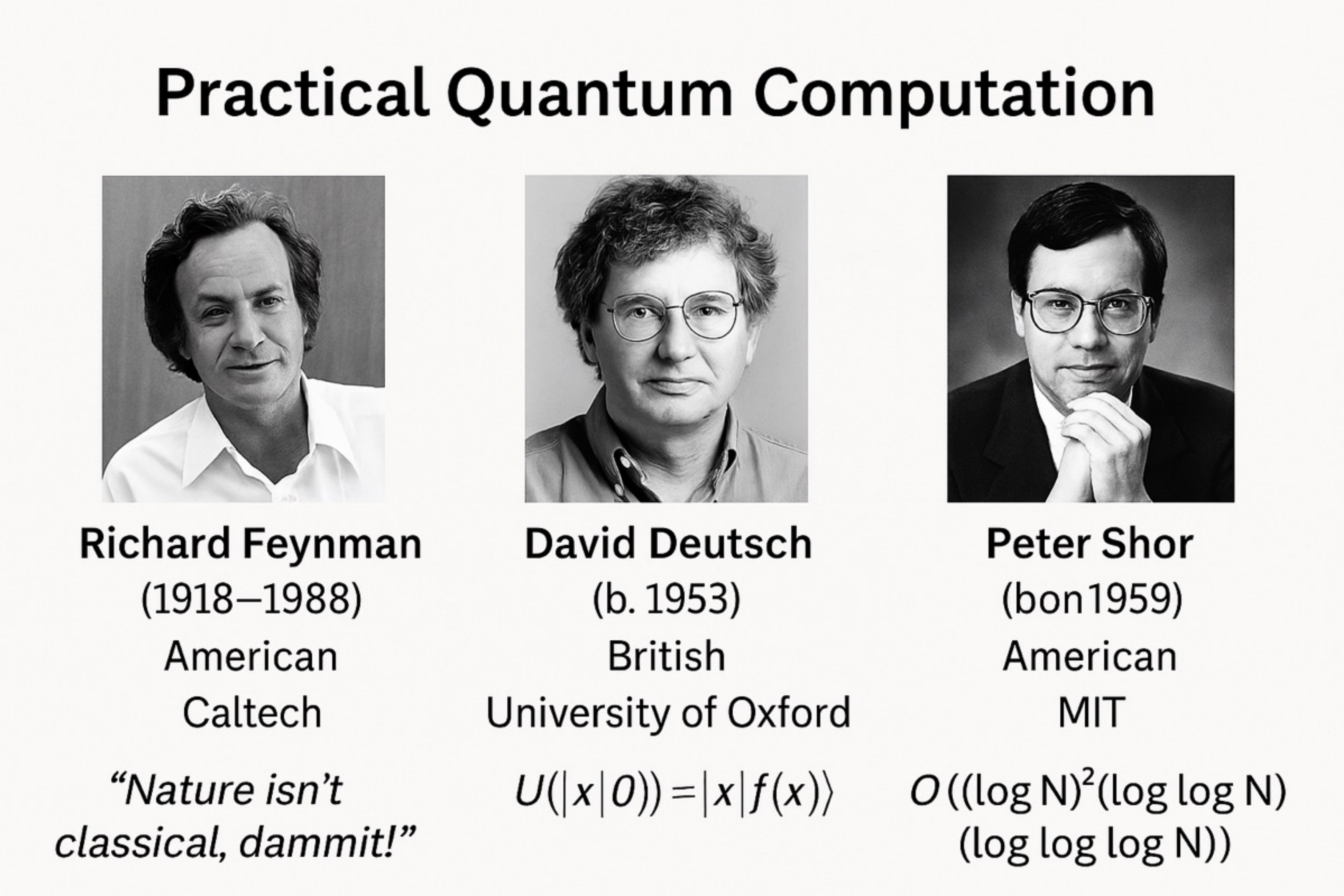
Quantum Algorithms

•**Quantum algorithms use quantum phenomena**
→ Superposition, interference, entanglement, and measurement

•**Different structure than classical algorithms**
→ Operate on qubit states, not deterministic bit states
→ Exploit probability amplitudes to steer computation

•**Goal: amplify correct answers, cancel out wrong ones**

Quantum Algorithms



Practical Quantum Computation

Richard Feynman
(1918–1988)
American
Caltech

“Nature isn’t classical, dammit!”

David Deutsch
(b. 1953)
British
University of Oxford

$U(|x|0))=|x|f(x)\rangle$

Peter Shor
(bon1959)
American
MIT

$O((\log N)^2(\log\log N)(\log\log\log N))$

Deutsch’s Algorithm

Problem:
Given a function $f : \{0, 1\} \to \{0, 1\}$, determine whether it is **constant** or **balanced**.

Classical approach: Requires 2 evaluations of f.
Quantum advantage: Requires only 1 quantum evaluation using superposition and interference.

Key Steps:

- Apply Hadamard gates to create superposition.
- Use a quantum oracle to evaluate $f(0)$ and $f(1)$ simultaneously.
- Apply Hadamard to interfere the outputs.
- Measure to determine whether f is constant or balanced.

Deutsch’s Algorithm

Qiskit-style pseudocode:

```
from qiskit import QuantumCircuit
qc = QuantumCircuit(2, 1)
qc.h([0, 1])          # Superposition
qc.cx(0, 1)           # Oracle (e.g., balanced case)
qc.h(0)               # Interference
qc.measure(0, 0)      # Measure first qubit
```

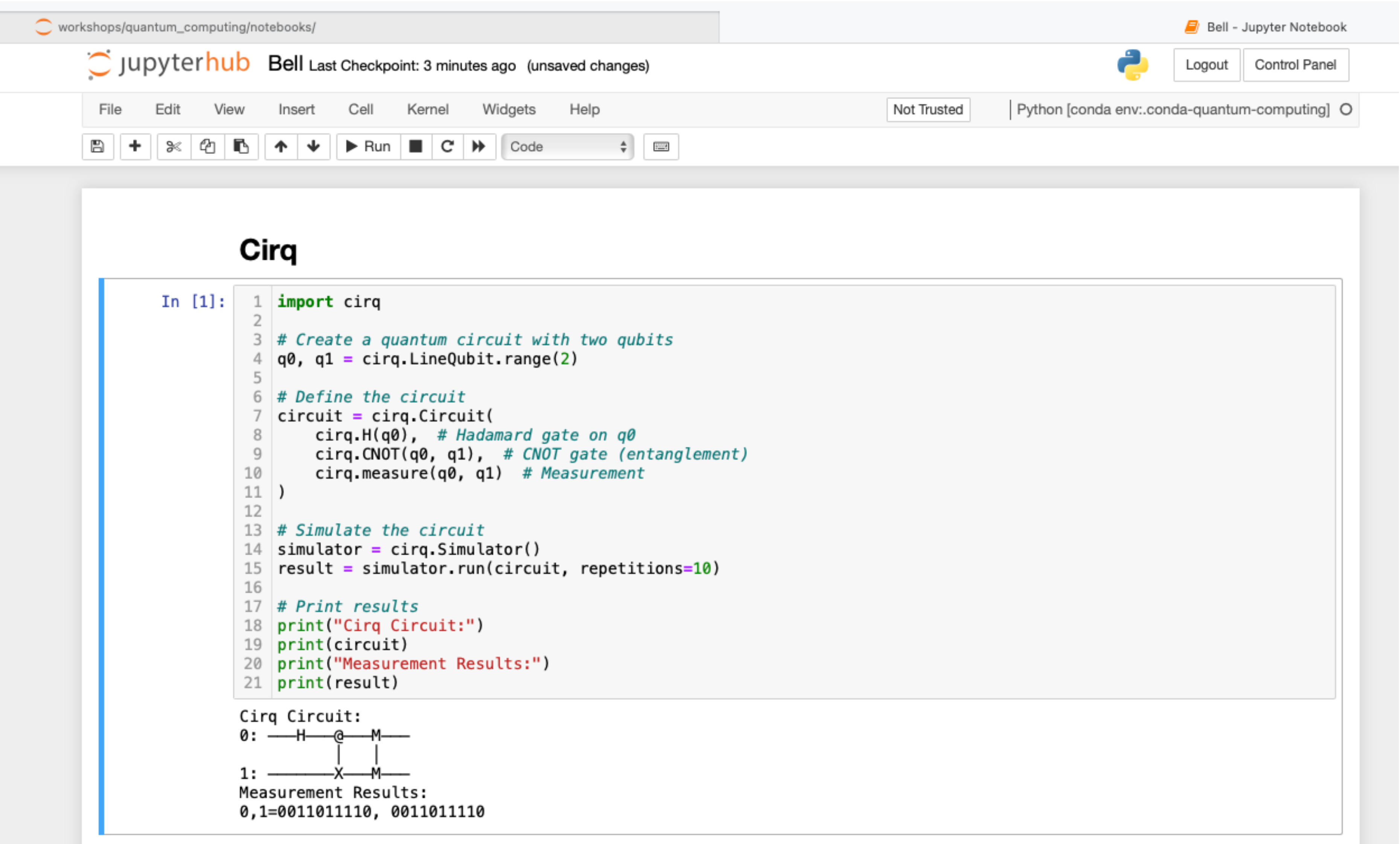
Interpretation:

- Measure 0 → f is **constant**
- Measure 1 → f is **balanced**

Classical Computer takes needs to test the function twice.

Quantum Computer only needs to test it once.

Now it’s your turn
Go to https://hopper.alliance.unm.edu:8000


workshops/quantum_computing/notebooks/
Bell - Jupyter Notebook
jupyterhub
Bell Last Checkpoint: 3 minutes ago (unsaved changes)
Logout
Control Panel
File
Edit
View
Insert
Cell
Kernel
Widgets
Help
Not Trusted
Python [conda env:.conda-quantum-computing]
Run
Code

Cirq

In [1]:

```python
import cirq

# Create a quantum circuit with two qubits
q0, q1 = cirq.LineQubit.range(2)

# Define the circuit
circuit = cirq.Circuit(
    cirq.H(q0),  # Hadamard gate on q0
    cirq.CNOT(q0, q1),  # CNOT gate (entanglement)
    cirq.measure(q0, q1)  # Measurement
)

# Simulate the circuit
simulator = cirq.Simulator()
result = simulator.run(circuit, repetitions=10)

# Print results
print("Cirq Circuit:")
print(circuit)
print("Measurement Results:")
print(result)
```

```
Cirq Circuit:
0: ───H───@───M───
          │   │
1: ───────X───M───
Measurement Results:
0,1=0011011110, 0011011110
```

Deutsch-Jozsa Algorithm

Given a function

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

guaranteed to be either:

- **Constant:** the output is the same for all 2^n inputs, or
- **Balanced:** the output is 0 for exactly half of the inputs, and 1 for the other half,

determine which case applies using as few evaluations of f as possible.

In other words, we generalize to the case where the problem size is n instead of 2.

Deutsch-Jozsa Algorithm

In the worst case, a classical algorithm needs to evaluate up to: $2^{n-1} + 1$ times to be sure—because you could see 2^{n-1} zeros before hitting a one.

The Deutsch–Jozsa algorithm requires exactly 1 call to the quantum oracle. All 2^n inputs are explored in superposition,
and quantum interference reveals the answer after a single evaluation.

Input Size n	**Classical (Worst Case)**	**Quantum**
1	2	1
2	3	1
3	5	1
n	$2^{n-1} + 1$	1
100	$\approx 6.3 \times 10^{29}$	**1**

Quantum Fourier Transform (QFT)

- Quantum analogue of the Discrete Fourier Transform (DFT)

Maps amplitudes to frequency components in the quantum state

- Operates on a quantum register

Transforms basis states $|x\rangle$ into a superposition of phase-encoded states

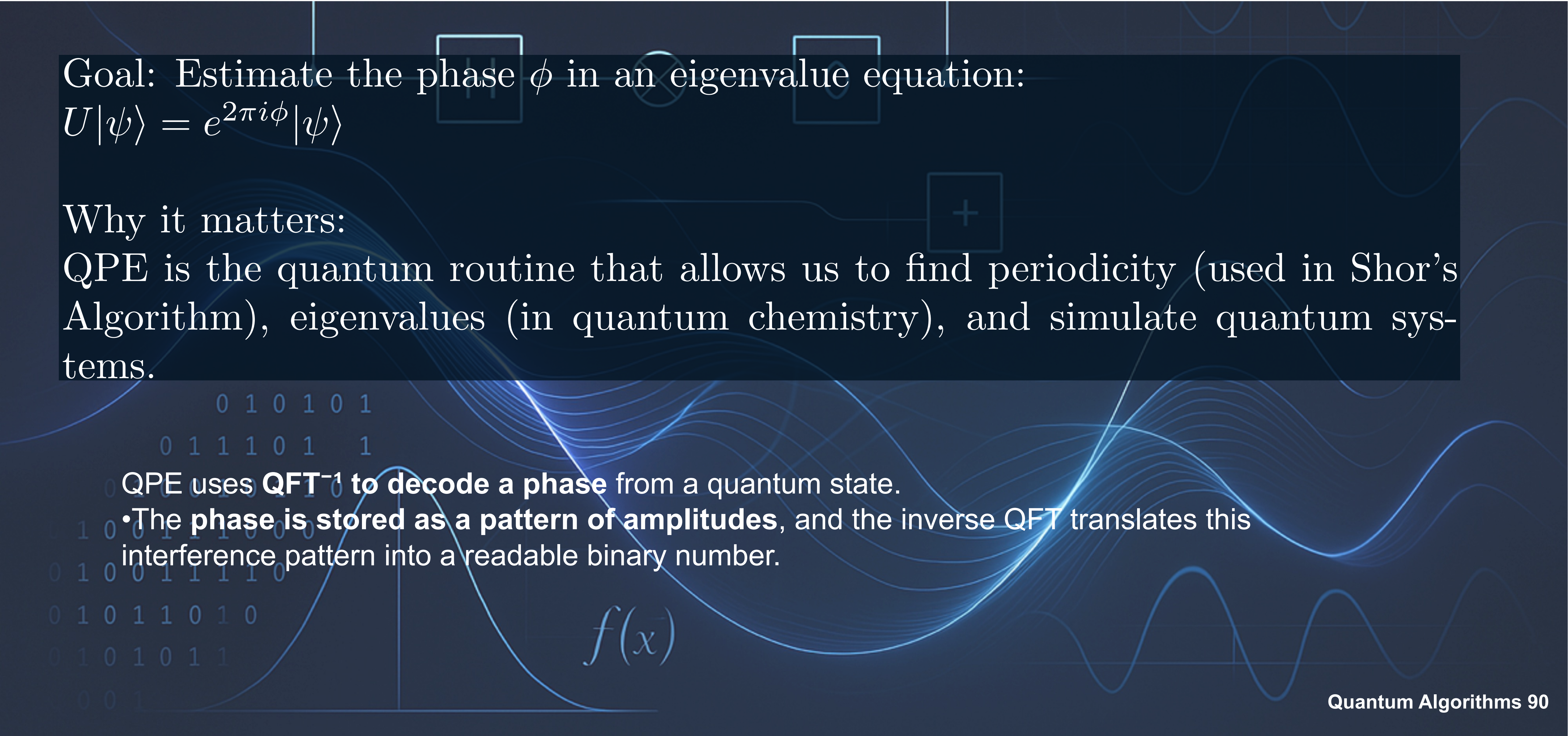
Core tool in algorithms like:

- Shor's Algorithm (factoring)
- Quantum Phase Estimation
- Hidden Subgroup Problems

Exponential speedup:

QFT uses $O(n^2)$ gates vs. classical DFT's $O(n2^n)$ operations

Quantum Phase Estimation (QPE)



Goal: Estimate the phase ϕ in an eigenvalue equation:
$U|\psi\rangle = e^{2\pi i\phi}|\psi\rangle$

Why it matters:
QPE is the quantum routine that allows us to find periodicity (used in Shor's Algorithm), eigenvalues (in quantum chemistry), and simulate quantum systems.

QPE uses **QFT^{-1} to decode a phase** from a quantum state.
•The **phase is stored as a pattern of amplitudes**, and the inverse QFT translates this interference pattern into a readable binary number.

Shor’s Algorithm

Goal: Factor a large integer N efficiently

- **Classical complexity:** Exponential time
- **Quantum complexity:** Polynomial time (with high probability)

- **Key insight:**
- Reduces factoring to a **period-finding** problem

- **Quantum subroutine:**

Uses **Quantum Phase Estimation (QPE)** and the **Quantum Fourier Transform (QFT)**

- **Breaks RSA encryption**
- (Because RSA’s security depends on factoring being hard)

Variational Quantum Eigensolver

Goal: Approximate the ground state energy of a quantum system (e.g., molecules, spin chains, Hamiltonians)

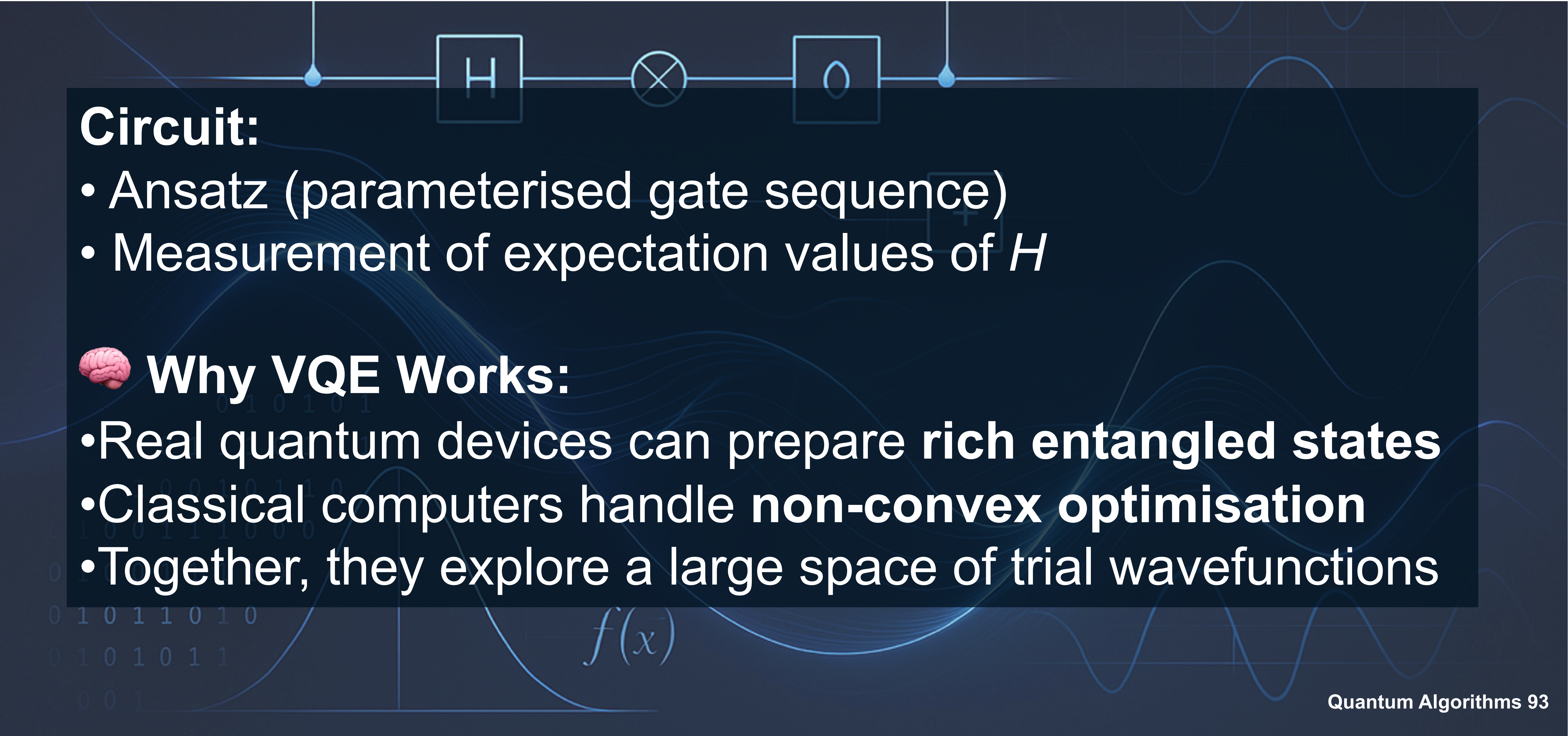
Hybrid approach:

- Quantum computer prepares parameterised quantum states
- Classical optimizer adjusts parameters to minimise energy
- Uses the variational principle:

$E(\theta) = \langle \psi(\theta) | H | \psi(\theta) \rangle \geq E_0$

→ The lowest measured energy gives an upper bound on the true ground state E_0

Variational Quantum Eigensolver



Circuit:

- Ansatz (parameterised gate sequence)
- Measurement of expectation values of *H*

🧠 **Why VQE Works:**

- Real quantum devices can prepare **rich entangled states**
- Classical computers handle **non-convex optimisation**
- Together, they explore a large space of trial wavefunctions

Why Quantum Error Correction Is Necessary

Qubits are fragile

- Easily disturbed by noise, temperature, or stray electromagnetic fields
- Suffer from **decoherence**, **bit-flip**, and **phase-flip** errors

No-cloning theorem

- We can’t make backup copies of quantum states like in classical RAM

Gate operations and measurements are imperfect

- Error rates per operation are often 0.1–1%, and these errors accumulate

Quantum algorithms require thousands or millions of gates

→ Without correction, noise overwhelms computation

QEC uses redundancy across multiple qubits

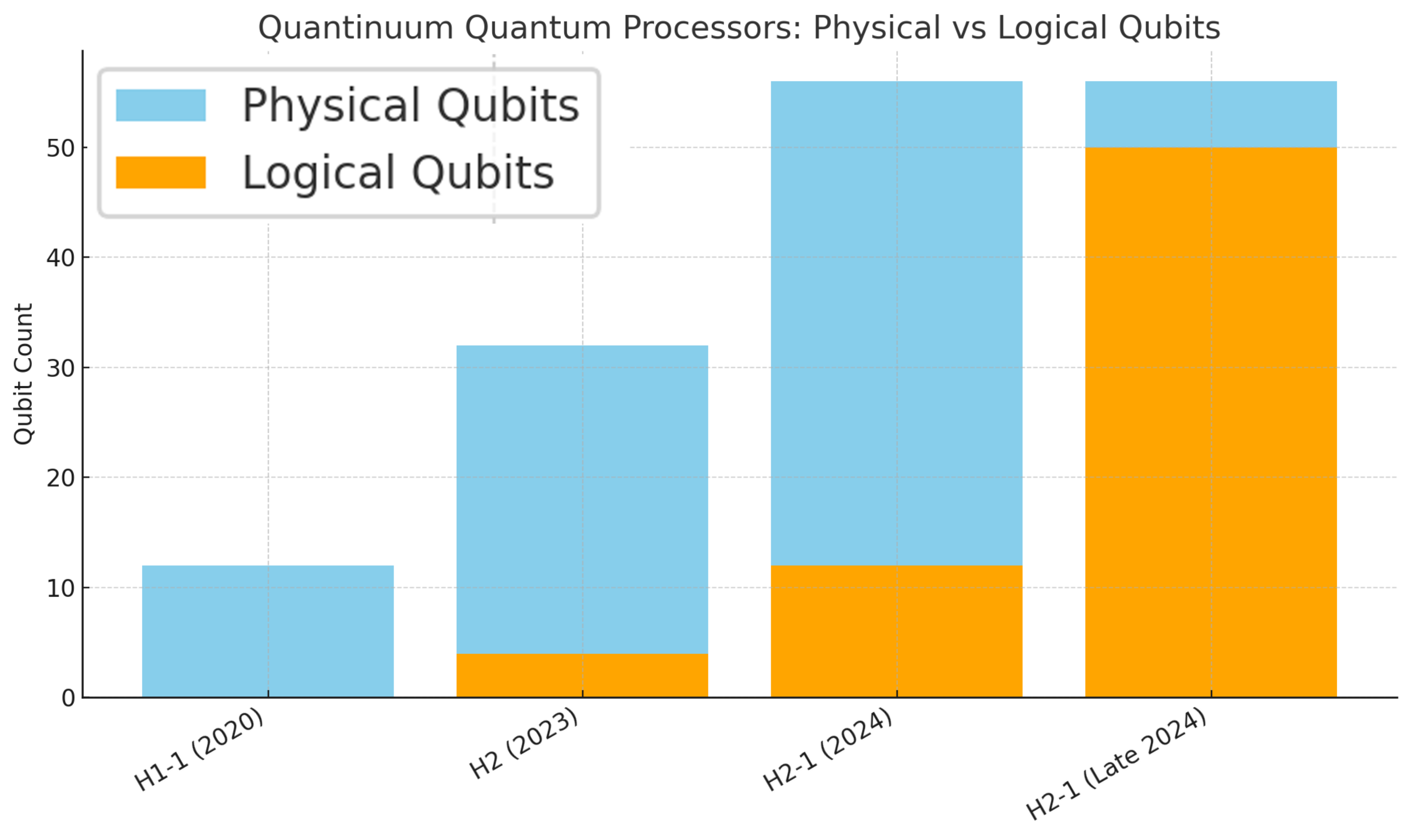
- One logical qubit = many physical qubits
- Errors are detected and corrected without collapsing the state

How We Protect Quantum Information

Encode 1 logical qubit into many physical qubits

Redundancy allows detection and correction of errors.

E.g. **Steane code** – uses 7 physical qubits per logical qubit



Summary: Quantum computing is real, and rapidly evolving

- Grounded in principles like superposition, entanglement, and interference
- Promises exponential speedups for select problems

- **We explored core concepts:**
  - Qubits, circuits, gates, and measurement
  - Quantum logic, algorithms, and hardware models
  - Key applications: factoring, search, simulation, optimization

- **Quantum advantage is still emerging**
  - Today’s machines are noisy and small
  - Error correction and scalability are the next frontier

- **HPC and quantum are deeply connected**
  - Both require specialised hardware, remote access, and hybrid workflows
  - CARC provides the tools to explore both worlds