Qubits: Physical platforms and applications

Maria Longobardi

maria.longobardi@unibas.ch

Swiss National Science Foundation

Universität Basel

NCCR SPIN

Outline

Lesson 1: QC Overview - What is the mathematical language of qubits?

Lesson 2: How do we evaluate physical qubits' performance?

Lesson 3: How do different platforms realize qubits?

Lesson 4: Qubits as a sensor
Overview of the quantum markets and talent development

Outline

Lesson 1: QC Overview - What is the mathematical language of qubits?

Lesson 2: How do we evaluate physical qubits' performance?

Lesson 3: How do different platforms realize qubits?

Lesson 4: Qubits as a sensor

Overview of the quantum markets and talent development

Quantum computing

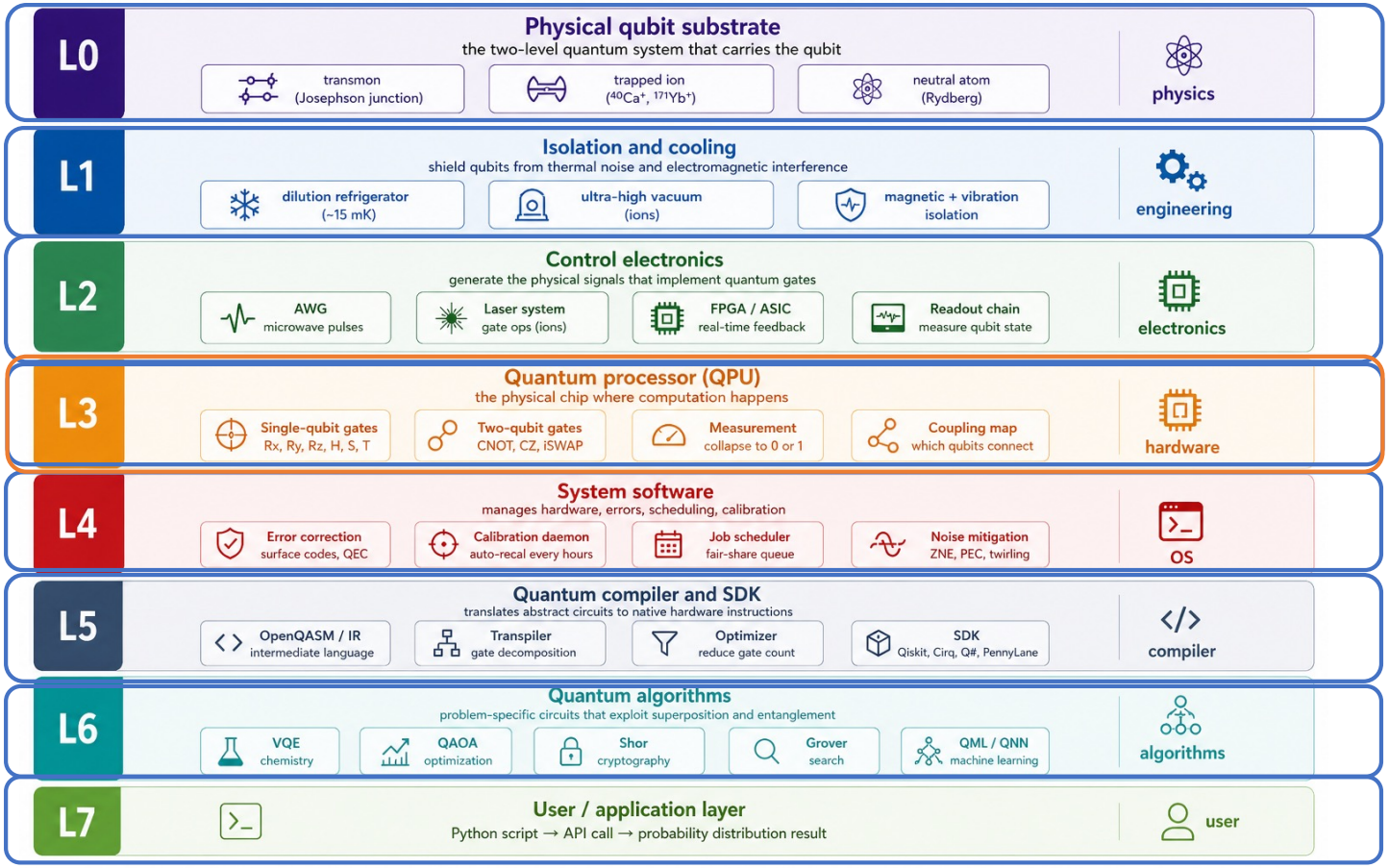
Technologically challenging

It requires expertise across several fields:

physics
materials science
engineering
nanofabrication
cryogenics
computer science
software, compilers, and algorithms

No single discipline is sufficient: it is an interdisciplinary endeavor

Full stack

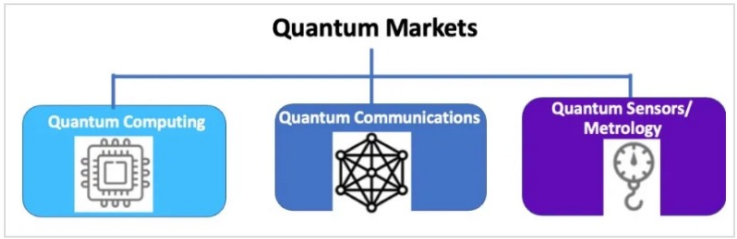

L0
Physical qubit substrate
the two-level quantum system that carries the qubit
transmon (Josephson junction)
trapped ion ($^{40}Ca^+$, $^{171}Yb^+$)
neutral atom (Rydberg)
physics
L1
Isolation and cooling
shield qubits from thermal noise and electromagnetic interference
dilution refrigerator (~15 mK)
ultra-high vacuum (ions)
magnetic + vibration isolation
engineering
L2
Control electronics
generate the physical signals that implement quantum gates
AWG microwave pulses
Laser system gate ops (ions)
FPGA / ASIC real-time feedback
Readout chain measure qubit state
electronics
L3
Quantum processor (QPU)
the physical chip where computation happens
Single-qubit gates Rx, Ry, Rz, H, S, T
Two-qubit gates CNOT, CZ, iSWAP
Measurement collapse to 0 or 1
Coupling map which qubits connect
hardware
L4
System software
manages hardware, errors, scheduling, calibration
Error correction surface codes, QEC
Calibration daemon auto-recal every hours
Job scheduler fair-share queue
Noise mitigation ZNE, PEC, twirling
OS
L5
Quantum compiler and SDK
translates abstract circuits to native hardware instructions
OpenQASM / IR intermediate language
Transpiler gate decomposition
Optimizer reduce gate count
SDK Qiskit, Cirq, Q#, PennyLane
compiler
L6
Quantum algorithms
problem-specific circuits that exploit superposition and entanglement
VQE chemistry
QAOA optimization
Shor cryptography
Grover search
QML / QNN machine learning
algorithms
L7
User / application layer
Python script → API call → probability distribution result
user

Quantum tech

1. **Quantum computing and simulation**: use a controllable many-body quantum system to process information, or simulate quantum matter

2. **Quantum communication**: distribute quantum states or quantum correlations between distant nodes

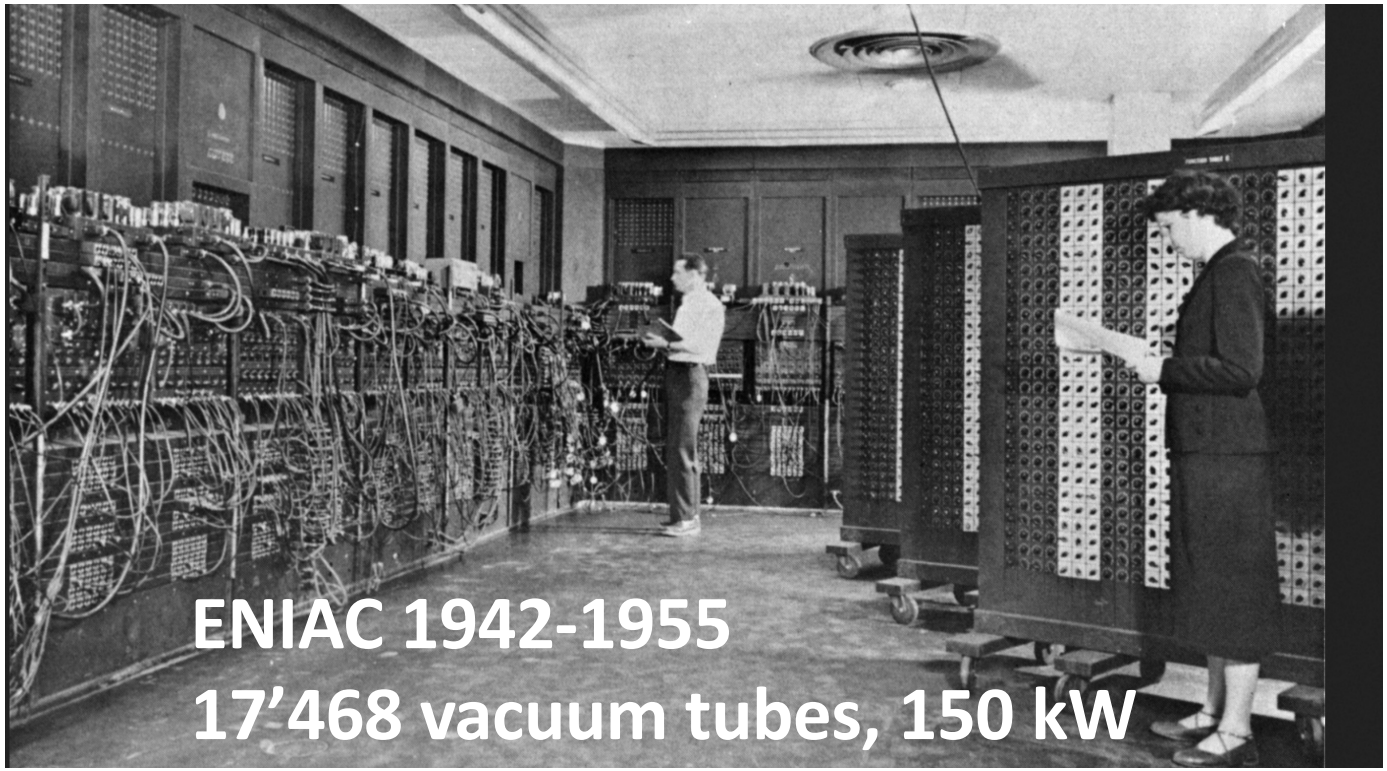
3. **Quantum sensing and metrology**: calibrated probe of an external parameter (magnetic field, acceleration, rotation, time, gravity, electric field, strain, temperature….)



A single well-controlled qubit can be a quantum sensor.

The boundary between the three domains is not always sharp.

Classical computer: First Era



ENIAC

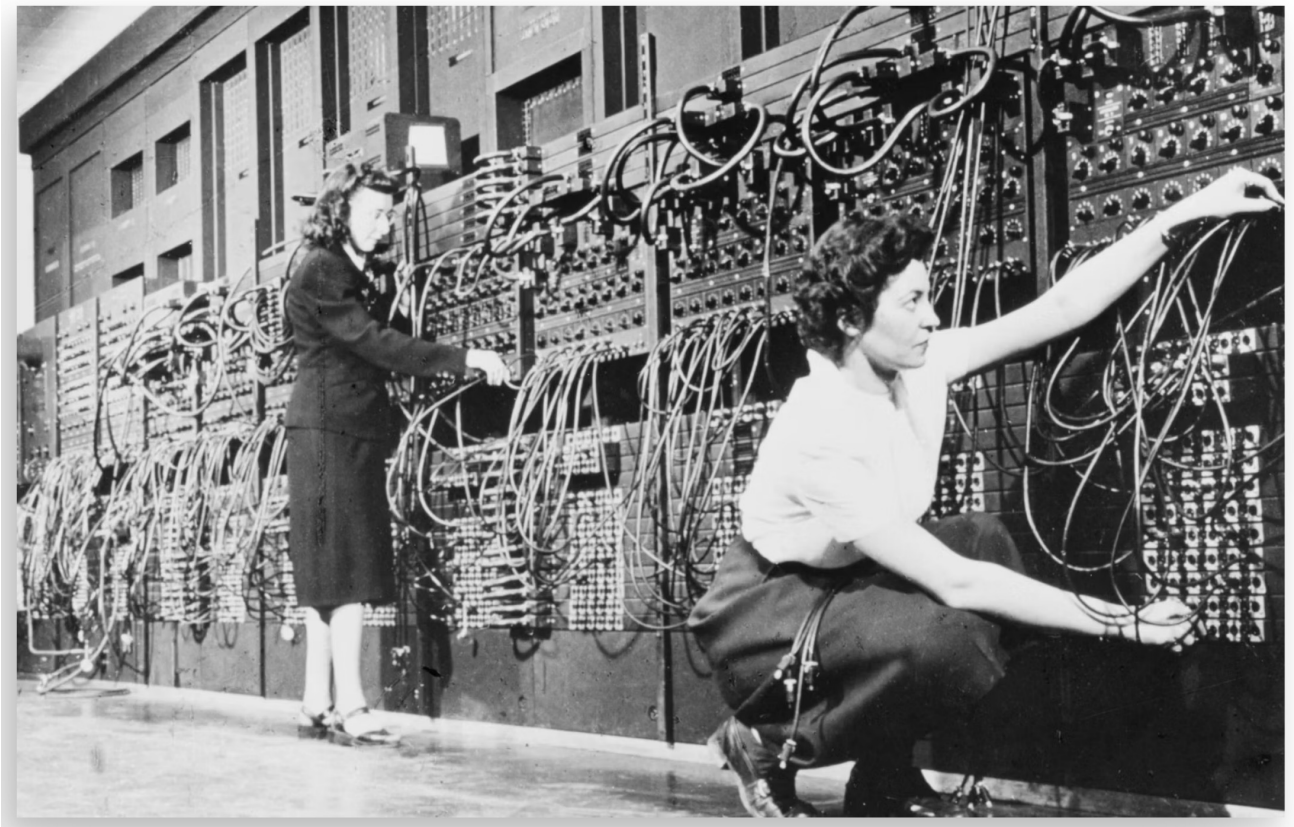
world’s first non-mechanical, general-purpose computer

Room sized
30 Tons

Performing
5000 simple arithmetic operations per second

The Electronic Numerical Integrator and Computer (ENIAC) was unveiled at the **University of Pennsylvania on 15 Feb 1946**

First Era



Untold History

ENIAC SIX

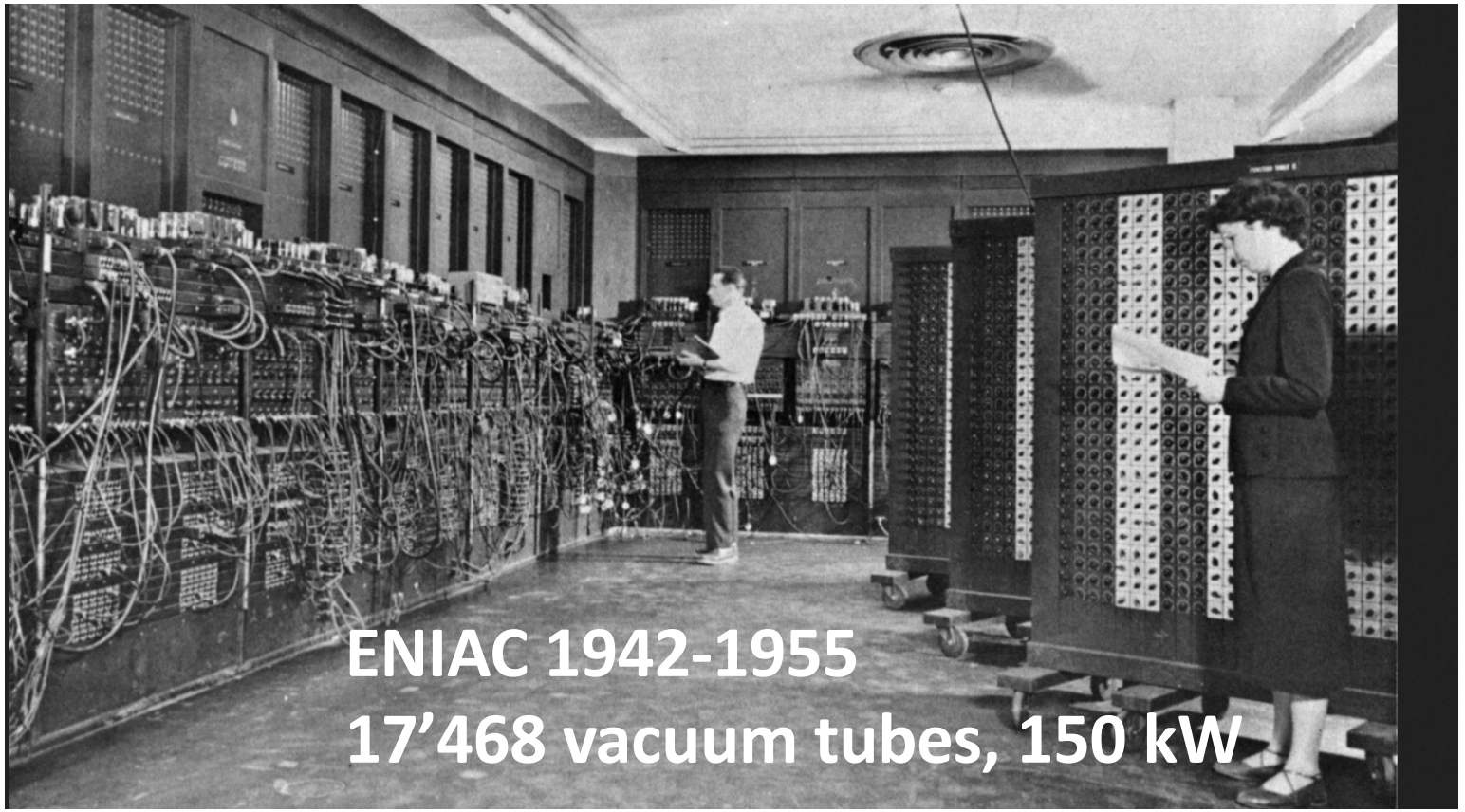
Betty Holberton
Gene Bartik
Kay McNulty
Frances Spence
Marilyn Meltzer
Ruth Teitelbaum

Invisible Women Programmed America's First Electronic Computer

The “human computers” who operated ENIAC

First demonstration: Marlyn Wescoff (left) and Ruth Lichterman were two of the ENIAC SIX first programmers.

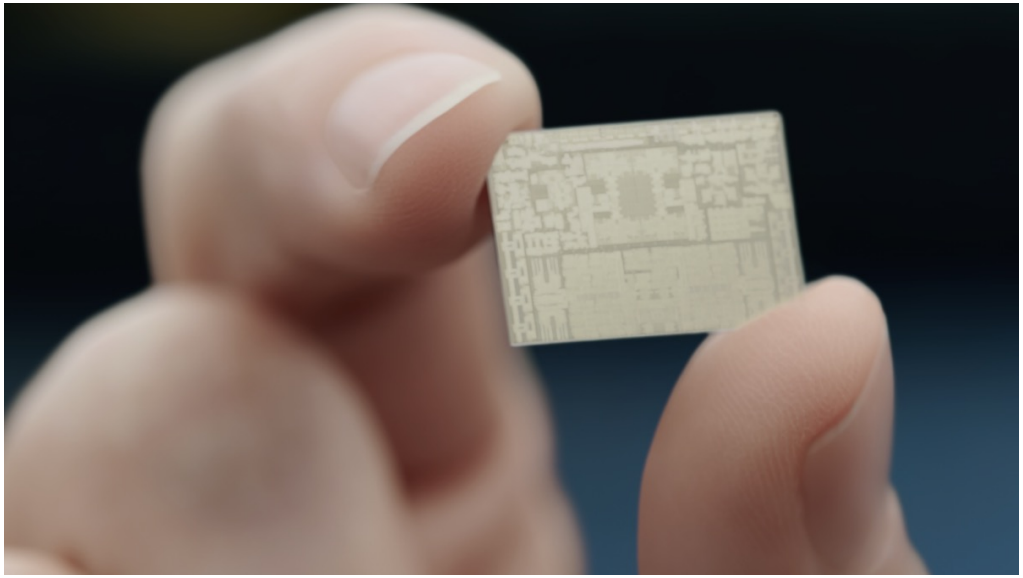
First Era



Room sized
30 Tons

5000 simple arithmetic operations per second

Classical computer



Credit: Apple

2023
Apple M3 Chip

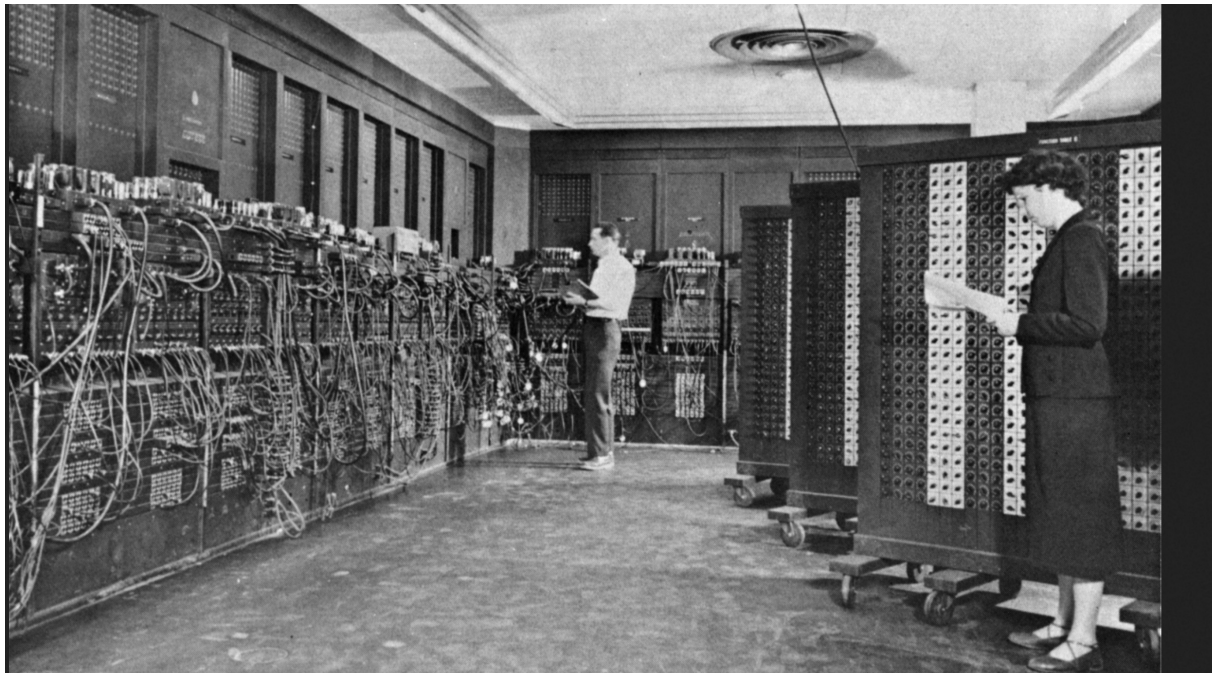
25 billion transistors
30 W

18 trillion operations per second

System	**ENIAC**	**Apple M3**
Year	1940s	2023
Technology	Vacuum tubes	Silicon transistors
Switching elements	**17,468 vacuum tubes**	**25 billion transistors**
Power	~150 kW	tens of watts
Size	room-sized	single chip inside a laptop
Main message	early electronic computer	compact integrated silicon chip
Operations per second	5000 simple arithmetic operations	18 trillion operations/s = 1.8×10^{13}

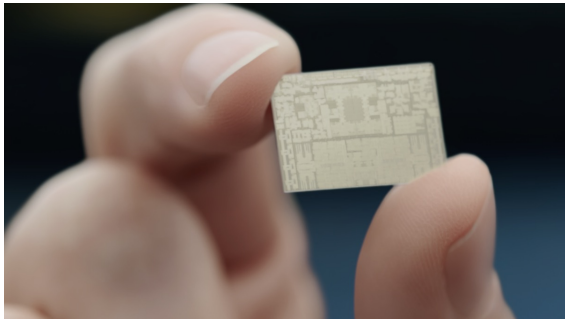
M3 is roughly 10^{13} i.e. about ten orders of magnitude higher than ENIAC

1946



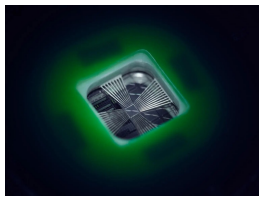
80 years

2024



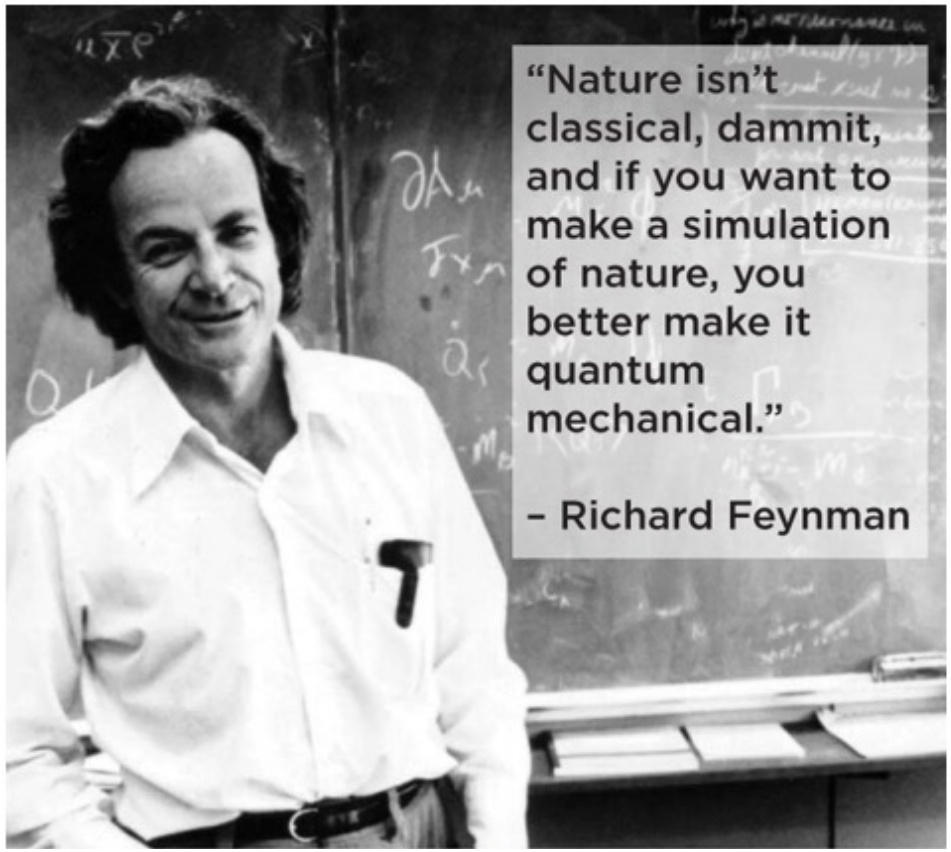
18 trillion operations/s = 1.8×10^{13}

Operations per second 5000 simple arithmetic operations



SemiQon

First Cryogenic CMOS transistor (2024)



Simulating Physics with Computers

Richard P. Feynman

Department of Physics, California Institute of Technology, Pasadena, California 91107

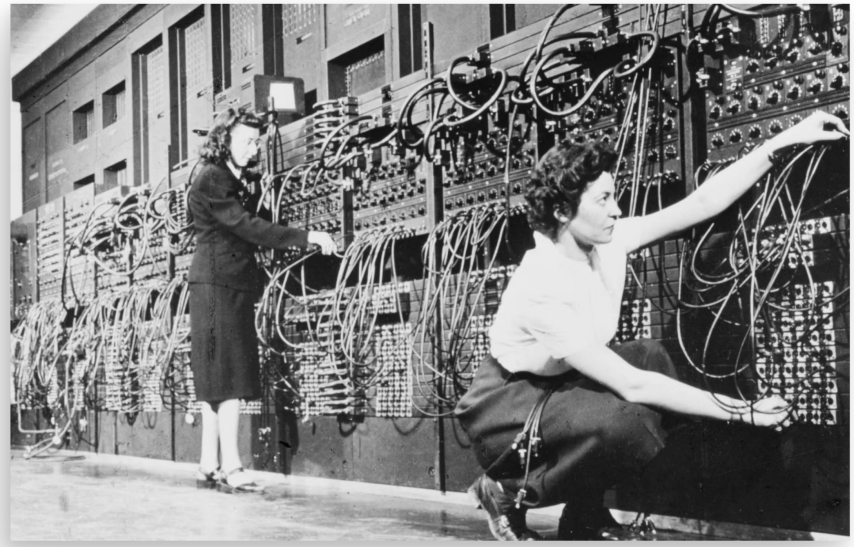
Received May 7, 1981

1. INTRODUCTION

On the program it says this is a keynote speech—and I don't know what a keynote speech is. I do not intend in any way to suggest what should be in this meeting as a keynote of the subjects or anything like that. I have my own things to say and to talk about and there's no implication that anybody needs to talk about the same thing or anything like it. So what I want to talk about is what Mike Dertouzos suggested that nobody would talk about. I want to talk about the problem of simulating physics with computers and I mean that in a specific way which I am going to explain.

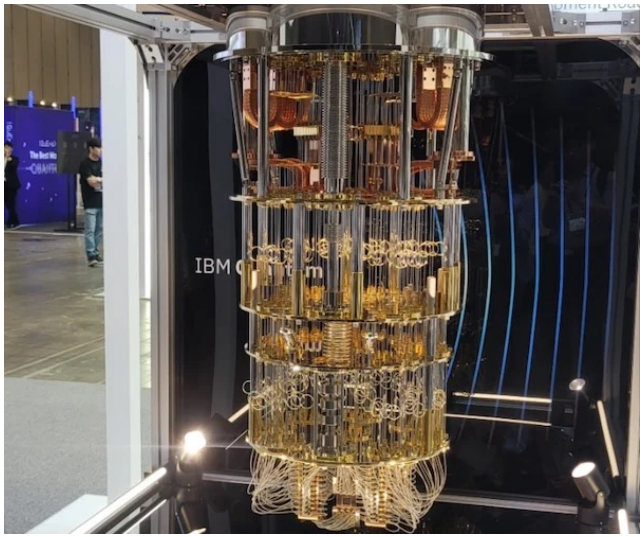
International Journal of Theoretical Physics, VoL 21, Nos. 6/7, 1982

Quantum computing



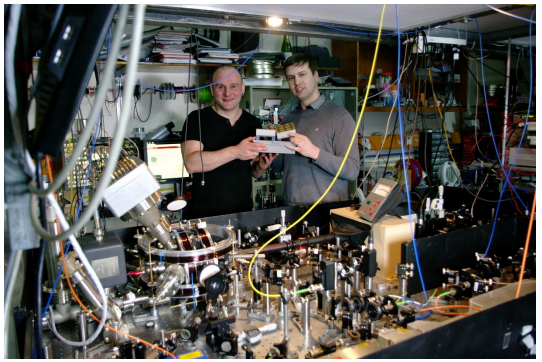
A typical quantum computer would resemble the classical computer in its earlier development time (middle of 19th century). The complexity and size of quantum computers' hardware are enormous.

Quantum Computers

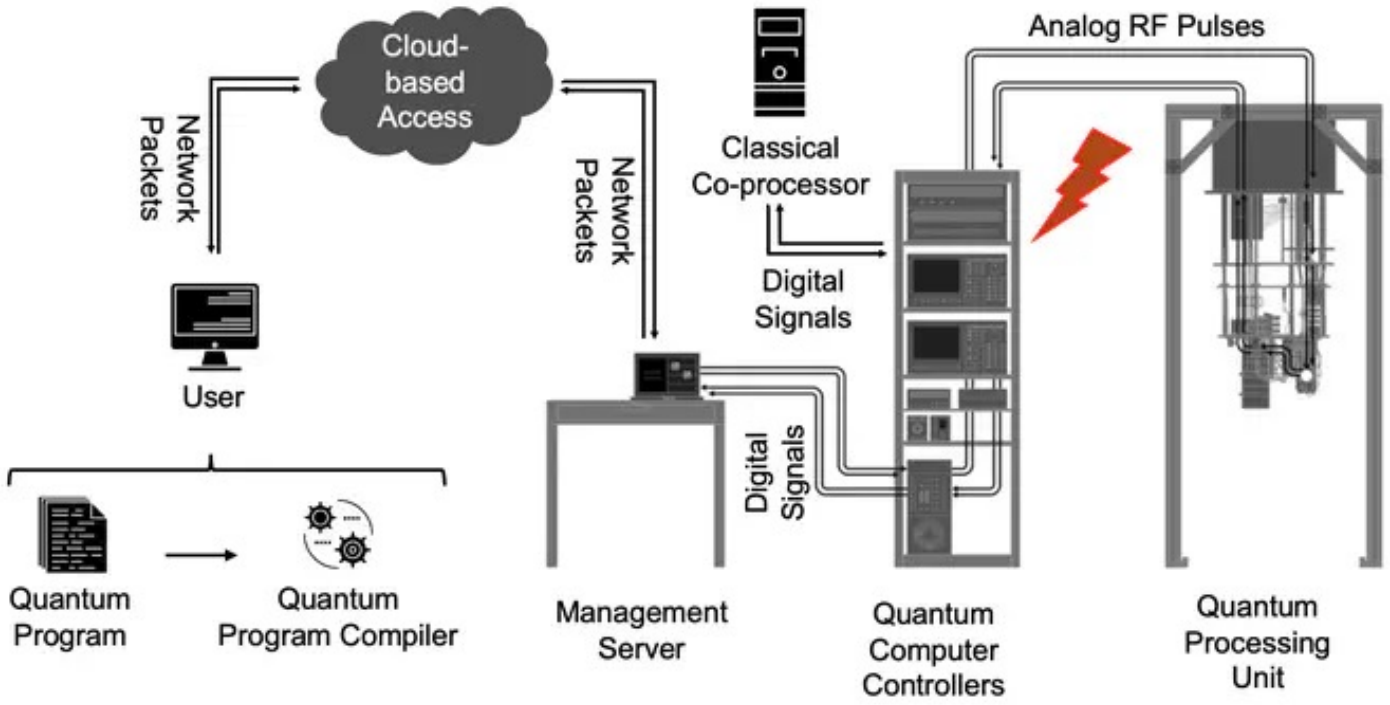


IBM

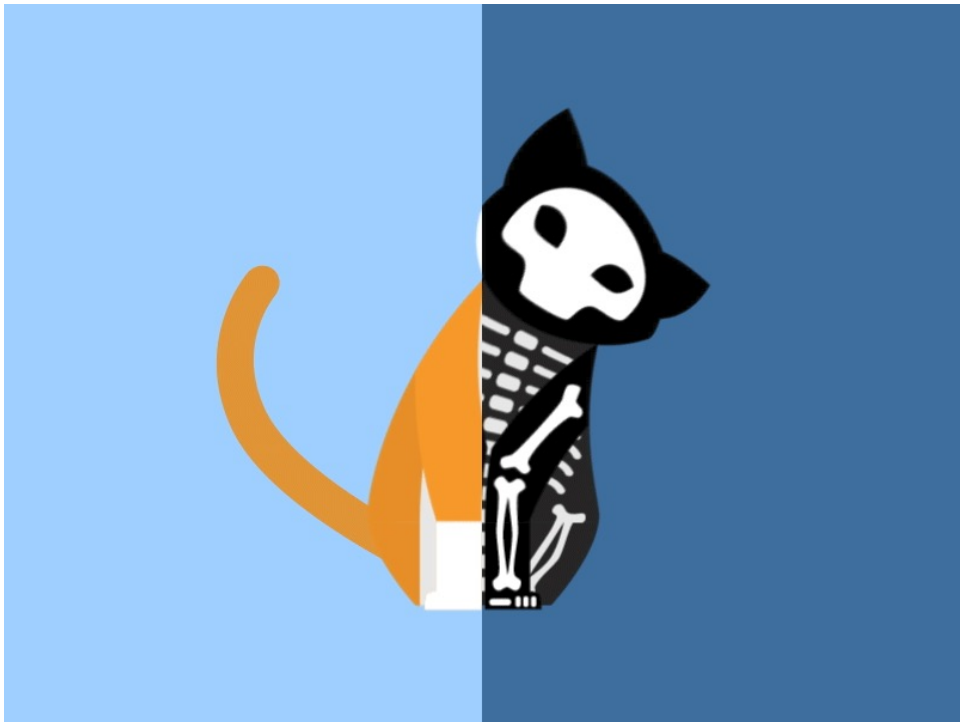
Nighthawk
120 Qubits
5000 Gates



Cloud based Quantum Computers

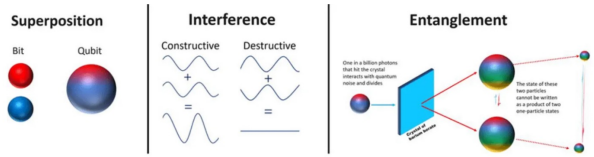


What is a qubit?



Credits: Kayla Mahoney

1. Superposition
2. Entanglement
3. Interference
4. Decoherence and noise



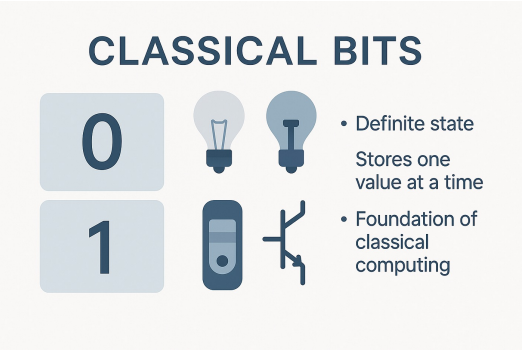
Properties of Quantum Computers (Source: SN Computer Science)

Bits

The most familiar information carrier is the classical bit:

$$b \in \{0, 1\}$$

A bit has two possible logical values.
At a given time, an ideal classical bit is either in state 0 or in state 1.



A classical bit is a two-state classical system.
It has two possible values, but it occupies one definite value at a time.

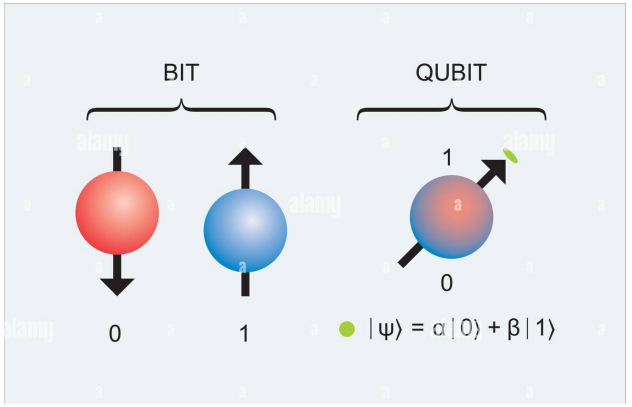
A qubit is also based on two distinguishable states,
but its state space is not simply the set {0, 1}.

Bits vs Qubits

Classical bit: one of two states, 0 or 1.

Qubit: quantum two-level system with basis states $|0\rangle \quad \text{and} \quad |1\rangle$

and superpositions $\alpha|0\rangle + \beta|1\rangle$.

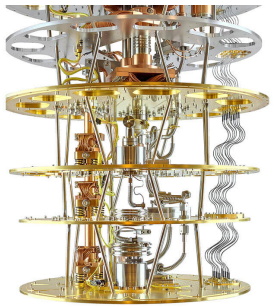


Qubits

A qubit is not only an abstract mathematical object.

It can be implemented using two controllable quantum states of a physical system.

These states may be energy levels, spin states, charge states, polarization states, or other well-isolated degrees of freedom.



Typical example is:

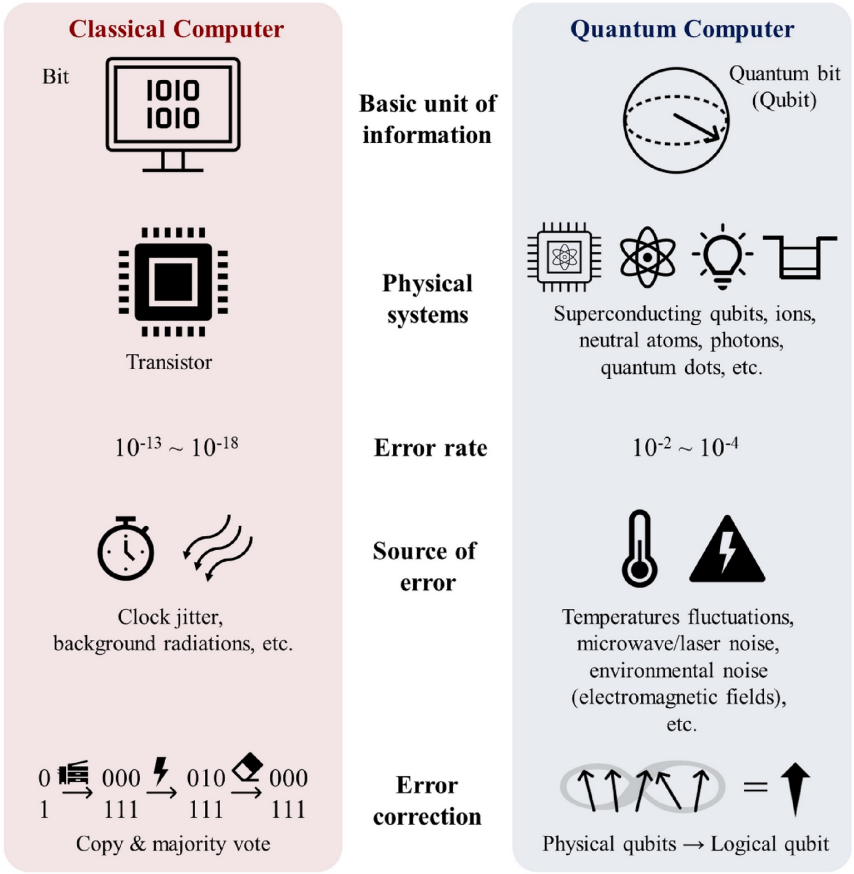
$$|\uparrow\rangle , |\downarrow\rangle$$

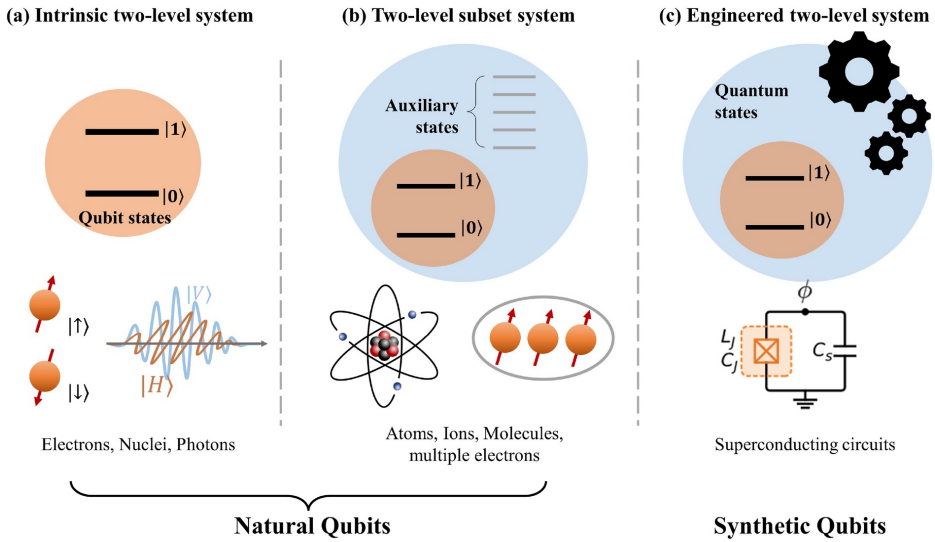
for a spin-1/2 degree of freedom. In quantum information, these are relabelled as

$$|0\rangle , |1\rangle$$

A qubit is a physical two-level quantum system whose states can be prepared, controlled, coupled, and measured.

Bits vs Qubits





Chae, E., et al..
Nano Convergence **11**, 11 (2024)

Qubit count is not enough

Qubit count is necessary, but incomplete.

It tells us the size of the quantum register, not whether the processor can perform a useful computation

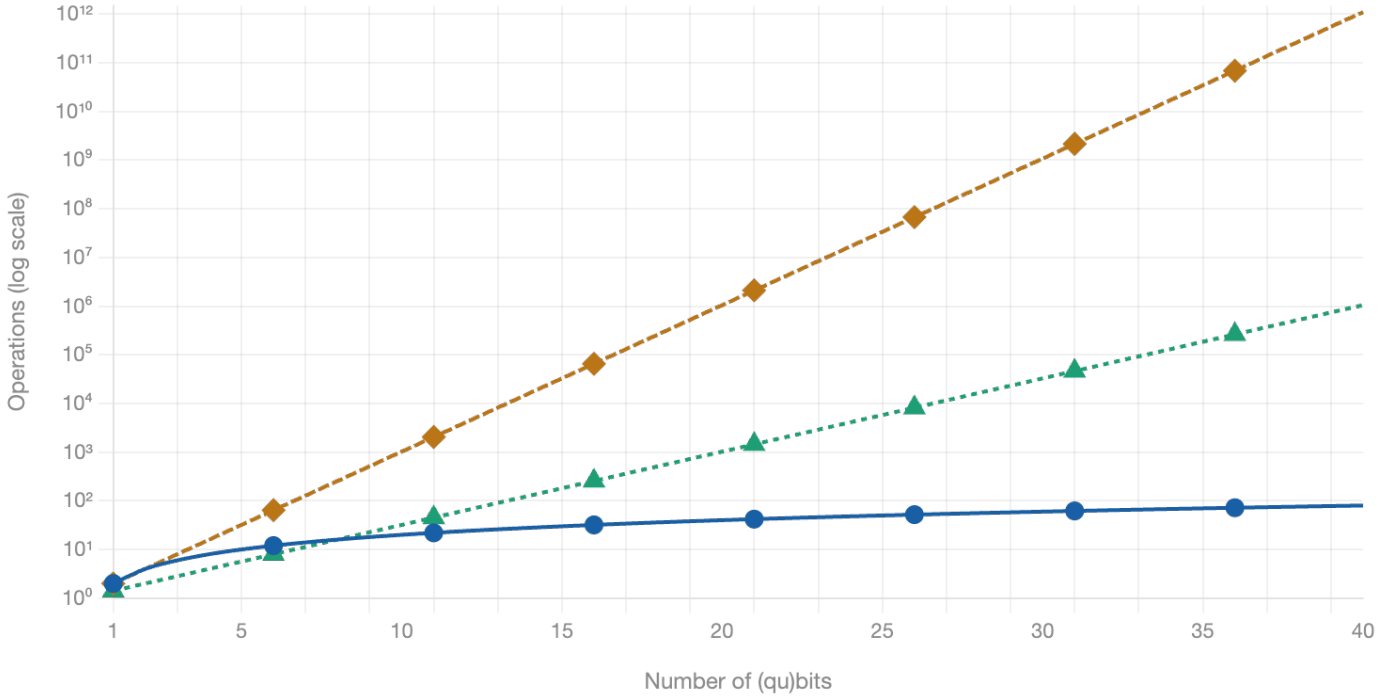
Counting qubits is like counting musicians in an orchestra.

The number matters, but it does not tell us whether they are in tune, coordinated, following the same score, or able to play without mistakes.

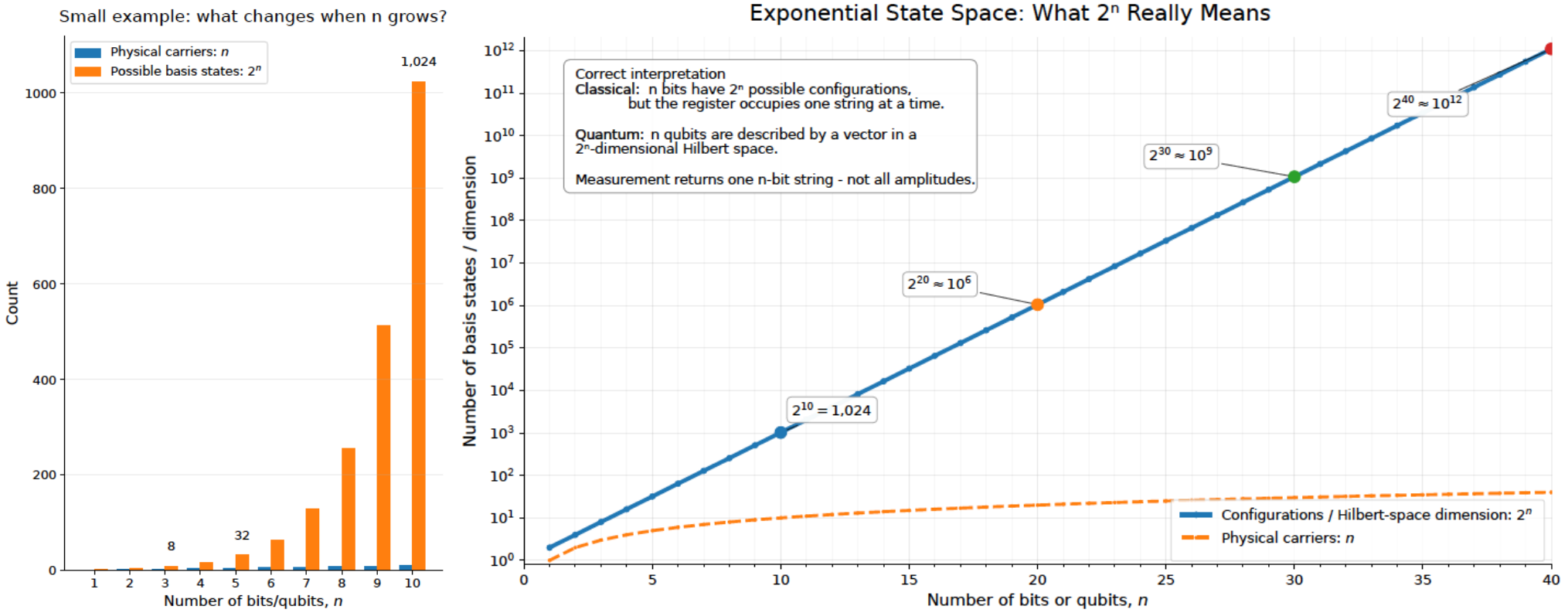
Classical vs Quantum

Classical register: one state among 2^n states

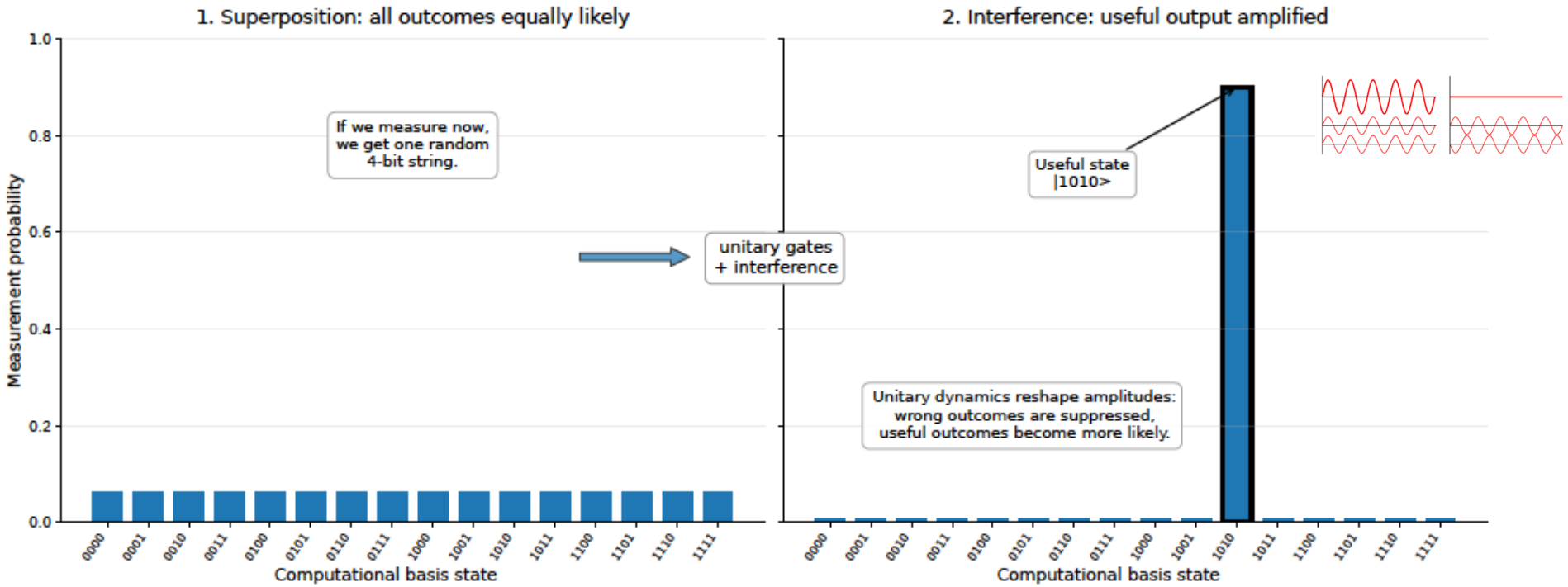
Quantum state: superposition over 2^n basis states



Classical vs Quantum



Qubits



Toy 4-qubit example: 16 basis states. Measurement returns one string, but the circuit can redistribute amplitudes before

Superposition, interference and entanglement

Superposition and interference are necessary, but usually not sufficient, for powerful quantum computation.

Entanglement allows qubits to behave as one correlated quantum system, rather than as many independent single-qubit systems.

Without entanglement, a many-qubit circuit remains close to a collection of independent single-qubit evolutions.

Such a circuit can still show superposition and interference, but it does not generally access the full many-body structure of Hilbert space and is often efficiently simulable on a classical computer.

Entanglement is what makes quantum computation genuinely multi-qubit.

Superposition, interference and entanglement

Superposition

The state can contain amplitudes over many basis states.

$$|\psi\rangle = \sum_x c_x |x\rangle$$

Interference

Gates reshape amplitudes to enhance useful outcomes and suppress others.

$$P(x) = |c_x|^2$$

Entanglement

The state may be genuinely multi-qubit, not a product of independent qubits.

$$|\psi\rangle \neq |\psi_1\rangle \otimes \cdots \otimes |\psi_n\rangle$$

Measurement

Only one classical bit string is observed.

$$x \in \{0,1\}^n$$

Noise and decoherence

Noise and decoherence destroy the quantum resources needed for computation.

They randomize phases, so interference becomes unreliable.

They reduce coherence, so amplitudes no longer evolve predictably.

They degrade entanglement, so multi-qubit correlations disappear or become effectively classical.

They create gate and readout errors, so the measured output no longer reflects the intended computation.

Quantum computation requires not only many qubits, but qubits that remain coherent, controllable, and accurately measurable.

NISQ = Noisy Intermediate-Scale Quantum

Today's quantum processors have physical qubits and can run quantum circuits.

However, they are **noisy**:
imperfect gates
finite coherence times
readout errors
limited circuit depth…

NISQ devices are useful for experiments, benchmarking, quantum simulation, and developing error-correction strategies.

Towards Quantum Utility

NISQ devices are moving from proof-of-principle demonstrations toward application-relevant experiments.

Selected demonstrations suggest quantum utility may be possible before full fault tolerance.

The field is now focused on improving qubit quality, error mitigation, and early error correction.

IBM has claimed evidence of quantum utility in selected experiments, not broad practical quantum advantage.

Towards Fault-Tolerant QC

Fault-tolerant quantum computing means that errors are continuously detected and corrected during the computation.

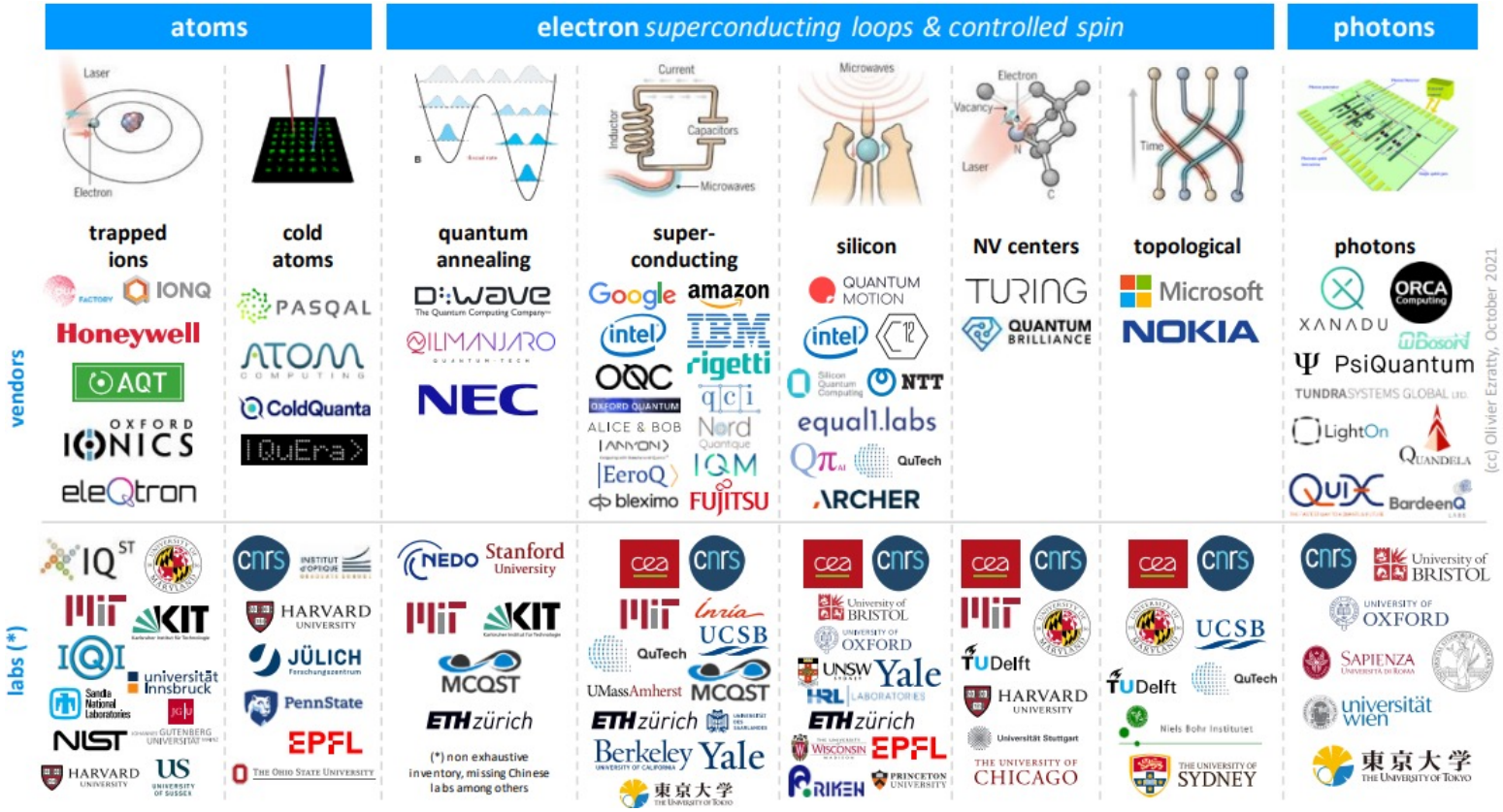
The goal is to build logical qubits from many imperfect physical qubits.

A logical qubit should fail much less often than the physical qubits used to encode it.

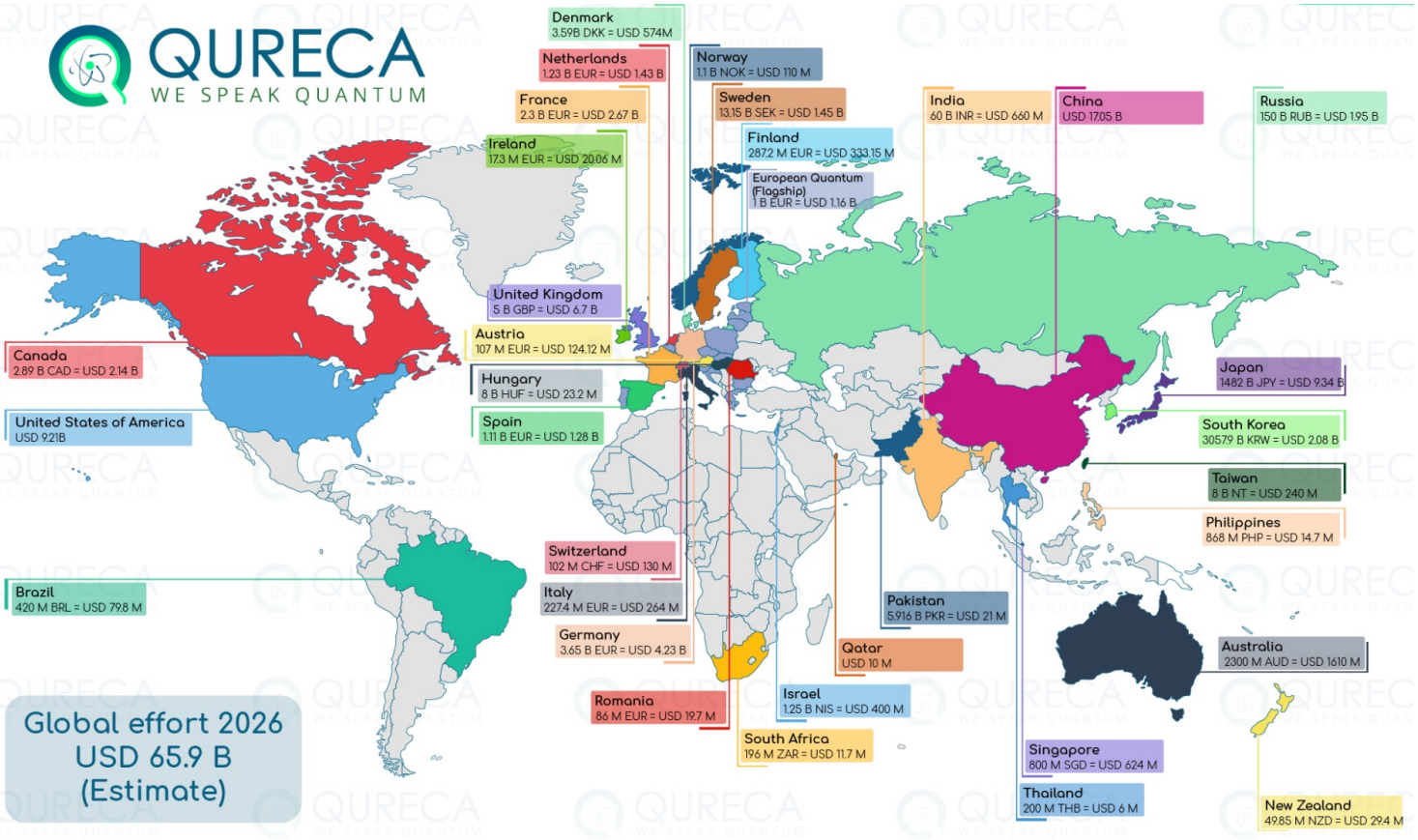
<u>Key requirements:</u>

high-fidelity physical gates
long coherence times
fast and reliable readout
quantum error correction
real-time decoding and feedback
scalable control electronics
..

Qubits

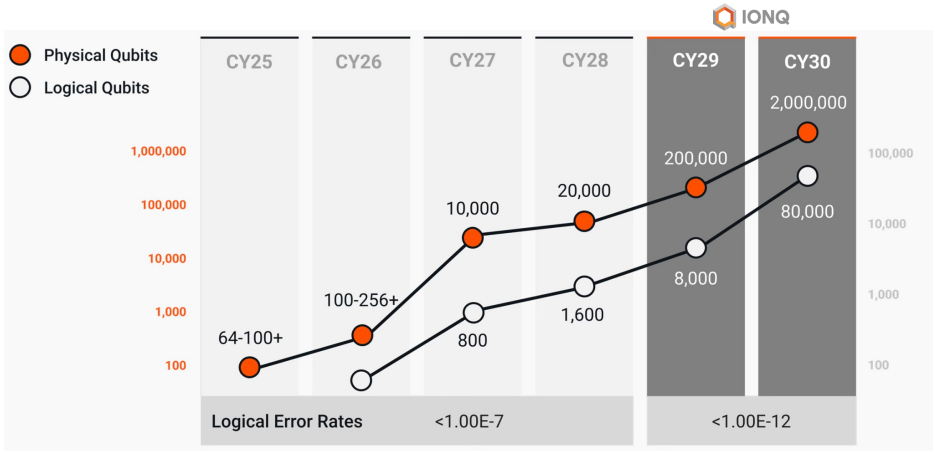

atoms
electron superconducting loops & controlled spin
photons
vendors
labs (*)
trapped ions
cold atoms
quantum annealing
super-conducting
silicon
NV centers
topological
photons
Honeywell
IONQ
AQT
OXFORD IONICS
eleQtron
PASQAL
ATOM COMPUTING
ColdQuanta
QuEra
D-Wave
ILMANJARO
NEC
Google
amazon
intel
IBM
rigetti
OQC
OXFORD QUANTUM
ALICE & BOB
IARYON
EeroQ
bleximo
FUJITSU
QUANTUM MOTION
intel
NTT
equal1.labs
QuTech
ARCHER
TURING
QUANTUM BRILLIANCE
Microsoft
NOKIA
XANADU
ORCA Computing
PsiQuantum
TUNDRA SYSTEMS GLOBAL
LightOn
QUANDELA
QuiX
BardeenQ
(cc) Olivier Ezratty, October 2021
IQ ST
MIT
KIT
IQI
universität Innsbruck
Sandia National Laboratories
NIST
HARVARD UNIVERSITY
US
cnrs
HARVARD UNIVERSITY
JÜLICH
PennState
EPFL
The Ohio State University
NEDO
Stanford University
MIT
KIT
MCQST
ETH zürich
(*) non exhaustive inventory, missing Chinese labs among others
cea
cnrs
MIT
Inria
QuTech
UCSB
UMassAmherst
MCQST
ETH zürich
Berkeley
Yale
東京大学
University of BRISTOL
UNIVERSITY OF OXFORD
UNSW
Yale
HRL LABORATORIES
ETH zürich
WISCONSIN
EPFL
RIKEN
PRINCETON UNIVERSITY
TU Delft
HARVARD UNIVERSITY
Universität Stuttgart
THE UNIVERSITY OF CHICAGO
UCSB
TU Delft
QuTech
Niels Bohr Institutet
THE UNIVERSITY OF SYDNEY
University of BRISTOL
UNIVERSITY OF OXFORD
SAPIENZA
universität wien
東京大学

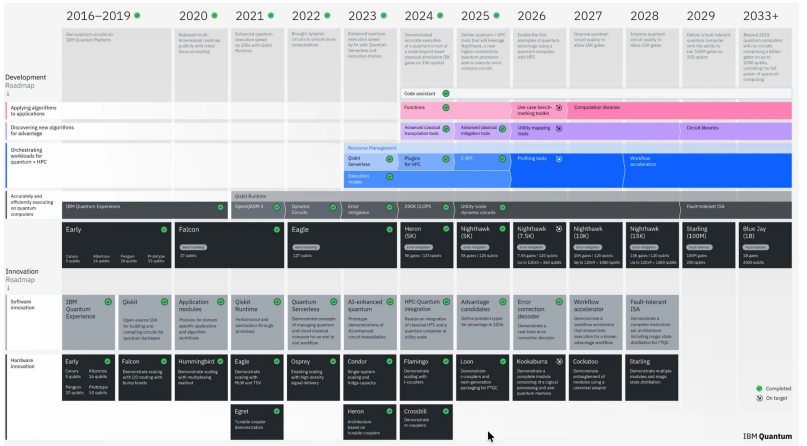
Global Quantum Investments



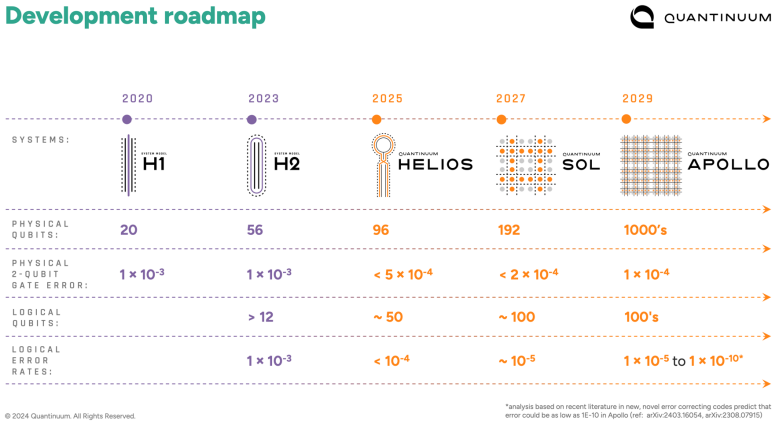
Italy has invested **227.4 million** € during 2021-2024 to bolster quantum research, industrial applications, and workforce development

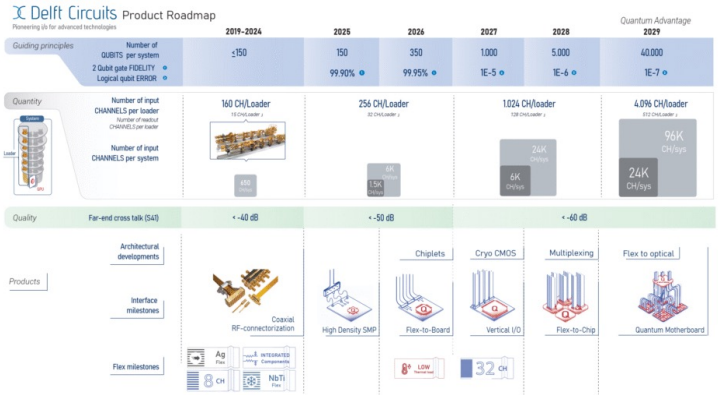
QC Roadmaps



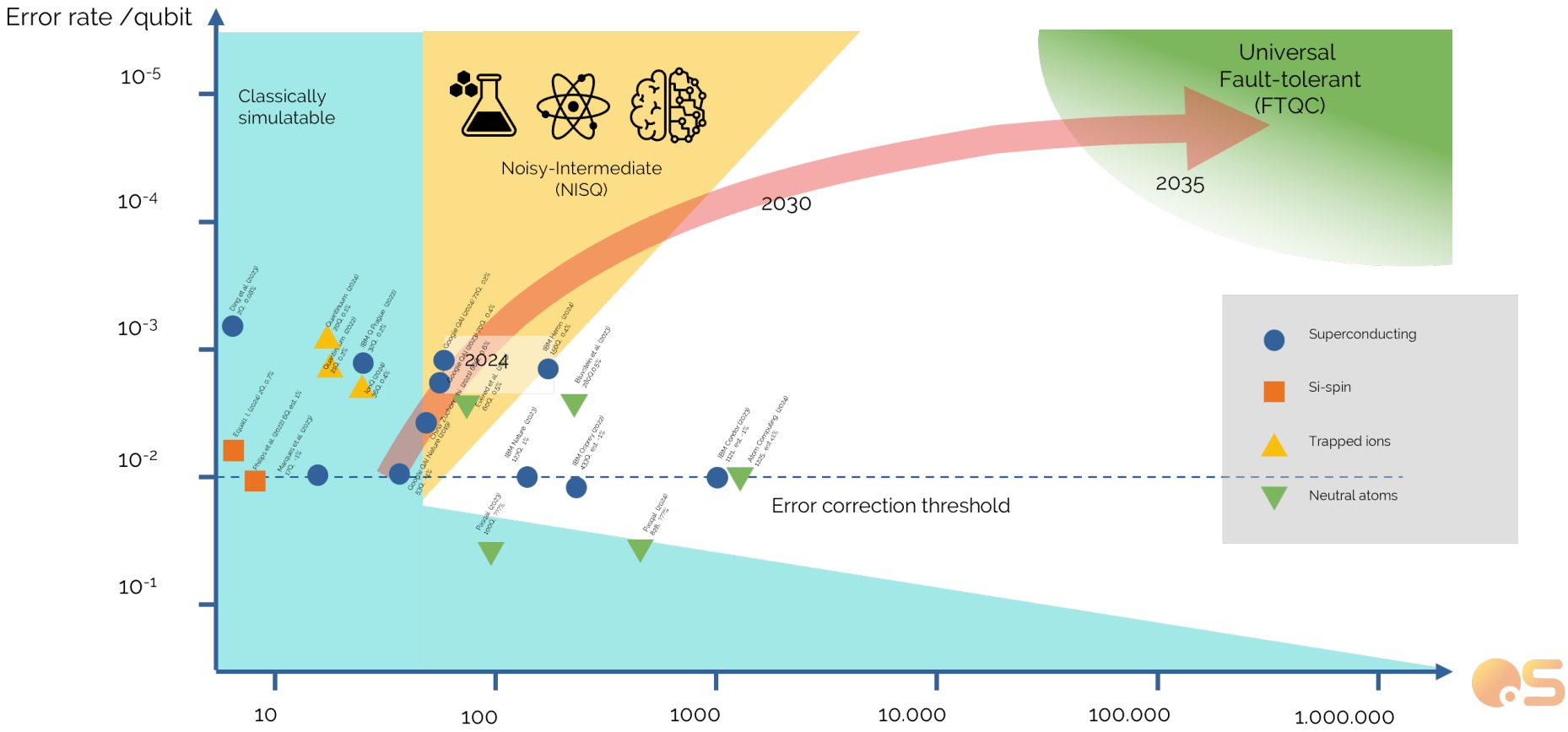


IBM





QC Roadmaps


Error rate /qubit
10^{-5}
10^{-4}
10^{-3}
10^{-2}
10^{-1}
Classically simulatable
Noisy-Intermediate (NISQ)
Universal Fault-tolerant (FTQC)
2024
2030
2035
Error correction threshold
Superconducting
Si-spin
Trapped ions
Neutral atoms
10
100
1000
10.000
100.000
1.000.000
# of qubits (log)

DiVincenzo Criteria (2000)

David DiVincenzo

1	**Scalable physical system** well-characterized qubits	Can you add more qubits without losing control?	partial
2	**Initialization** reset to |000...⟩	Can you reliably start every computation from a clean state?	solved
3	**Long coherence times** $T_2 \gg t_{gate}$	Do qubits stay quantum long enough to finish the algorithm?	improving
4	**Universal gate set** single + two-qubit gates	Can you implement any quantum algorithm with available gates?	solved
5	**Qubit-specific measurement** readout fidelity > 99%	Can you read one qubit without disturbing the others?	partial

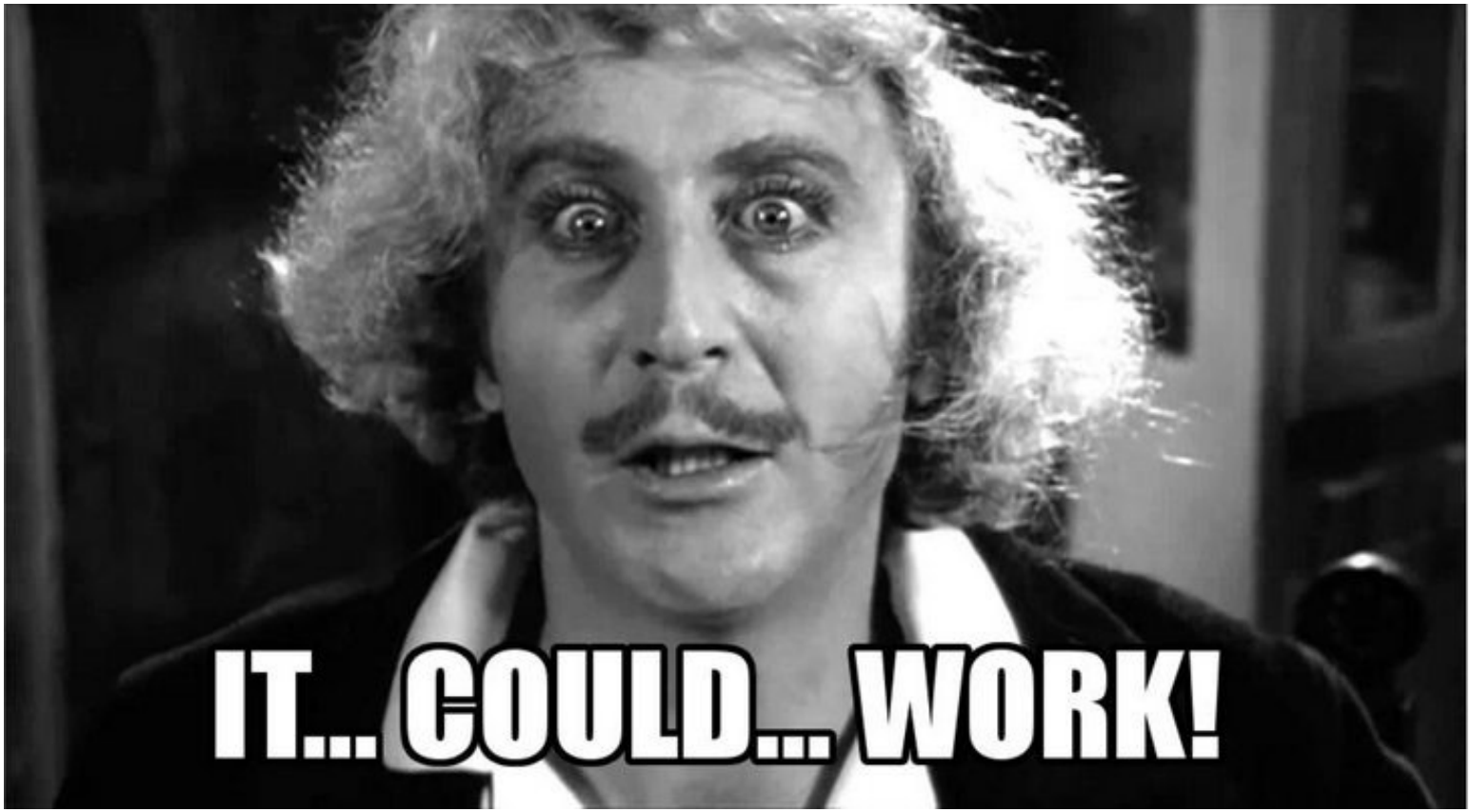
FOR QUANTUM COMMUNICATION

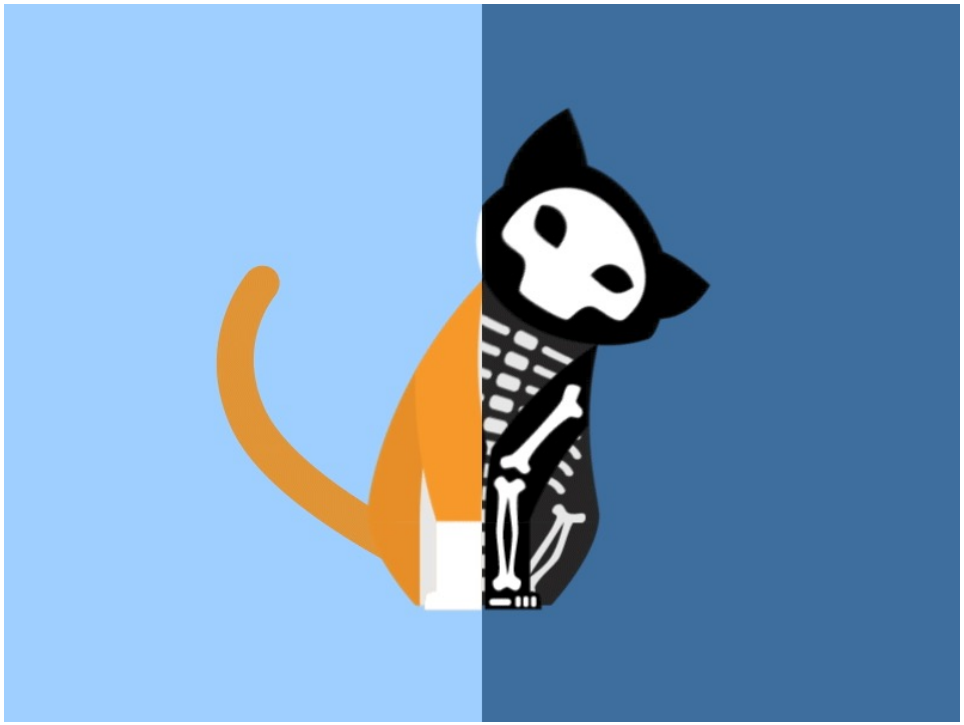
6	**Stationary ↔ flying qubit** processor ↔ photon interface	Can you transfer a qubit from a chip onto a photon to send it?	open
7	**Faithful transmission** quantum networks	Can you send qubits between distant locations without errors?	open

- **solved** — achieved on all major platforms
- **partial / improving** — works but not at scale
- **open** — no platform has achieved this yet

Scalable, stable and measurable qubits

that can be used to send information

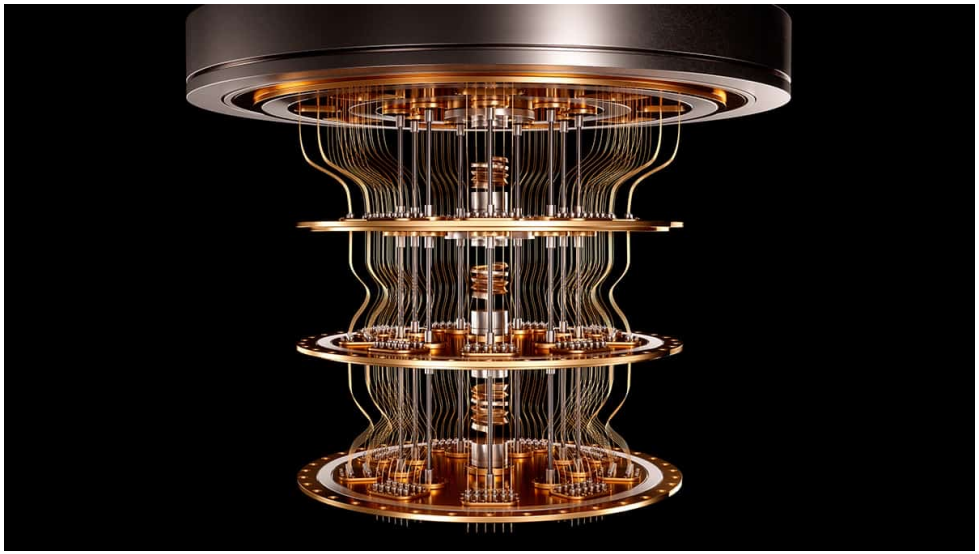

IT... COULD... WORK!



Credits: Kayla Mahoney

1. Qubits and Bloch Sphere
2. Pauli Matrices
3. Multi-qubits

How to describe a qubit mathematically?



Quantum computing is built on

linear algebra
in complex Hilbert spaces

Basic concepts

Linear transformations

Algebra

Hilbert Space

Basic concepts: vectors, algebra and linear transformations

Quantum is built on linear algebra

Algebra

Algebra is a set of rules for combining mathematical objects.

In linear algebra, the main objects **are vectors and linear transformations**.

We can add vectors, multiply them by scalars, and represent transformations as matrices.

This is the mathematical language used to describe quantum states, gates, and measurements.

Basic concepts: vectors, algebra and linear transformations

Linear transformations

A linear operation preserves superpositions:
scaling the input scales the output, adding two inputs gives the sum of the two outputs

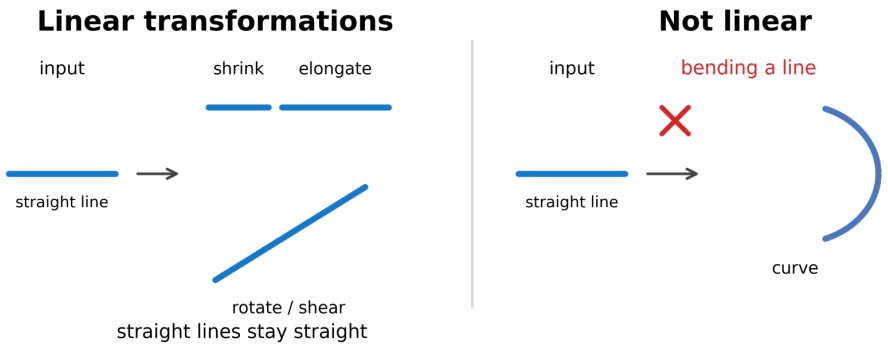
Geometrically, linear transformations map straight lines to straight lines.

- $T(u+v) = T(u) + T(v)$ **additivity**
  $T(cu) = cT(u)$ **homogeneity**

Together:

$$T(au + bv) = aT(u) + bT(v)$$

Preservation of linear combinations.



This is why linear transformations are compatible with superposition.

Quantum states are vectors in the Hilbert Space

The key point is that **Hilbert Space is not just any vector space**.

It has an inner product, so we can define:

$$\langle \psi | \phi \rangle$$

and therefore probabilities, orthogonality, normalization, projections, and expectation values.

Quantum mechanics is formulated using linear algebra in complex Hilbert spaces.

States are represented by **vectors**.
Observables are represented by **self-adjoint operators**.
Physical evolution is represented by **unitary transformations**.

Quantum State Space: Hilbert Space

The state space of a quantum system is represented by a complex Hilbert space.

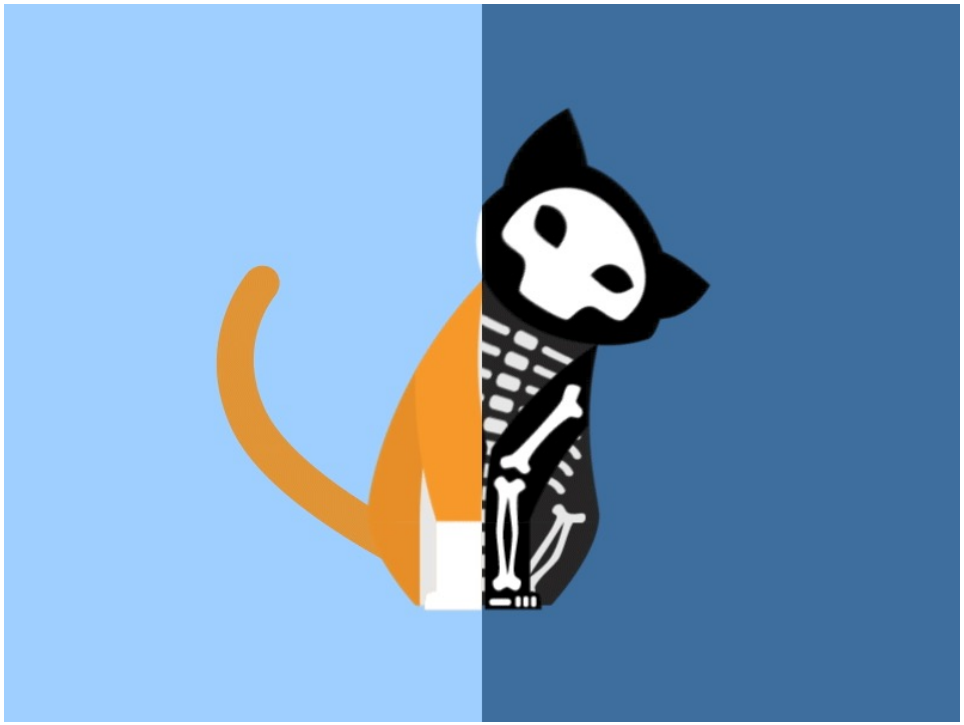
A Hilbert space is:
a complex vector space
equipped with an inner product
complete with respect to the norm induced by that inner product

Vector space:
we can add vectors and multiply them by complex scalars, following the rules of linear algebra

Inner product:
defines lengths, angles, overlaps, and measurement probabilities.

Completeness:
every Cauchy sequence of vectors converges to a vector inside the space.

Intuition: the space has no missing limit points, or no “holes.”



Credits: Kayla Mahoney

1. Qubits and Bloch Sphere
2. Pauli Matrices
3. Multi-qubits

Pure quantum states

Pure quantum states are represented by normalized vectors in a complex Hilbert space.

$$|\psi\rangle \in \mathcal{H}$$

$$\langle\psi|\psi\rangle = 1$$

Two vectors that differ only by a global phase (rotation in the complex space) represent the same physical state:

$$|\psi\rangle \sim e^{i\alpha}|\psi\rangle$$

A **pure state** means that the quantum state is known if it can be fully described by a single normalized state vector $|\psi\rangle$.

In a finite−dimensional (d) system

$$|\psi\rangle = \sum_{j=0}^{d-1} c_j |j\rangle ,$$

$|j\rangle$ is the j-th basis state

$$\sum_j |c_j|^2 = 1.$$

c_j are complex amplitudes.

Single qubit

A qubit $|\psi\rangle$ is the special case d=2: $\mathcal{H} \simeq \mathbb{C}^2$,

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

computational basis
$|0\rangle$ $|1\rangle$

The coefficients $\boldsymbol{\alpha}$ **and** $\boldsymbol{\beta}$ **are complex amplitudes**.

A physical state must be **normalized**:

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

Measurement in the computational basis gives:

$$P(0) = |\alpha|^2, \qquad P(1) = |\beta|^2$$

The quantum state contains amplitudes.
The measurement returns one classical outcome.

From amplitudes to the Bloch sphere

Every pure qubit $|\psi\rangle$ can be written as:

After removing the physically irrelevant global phase

(spherical coordinates)

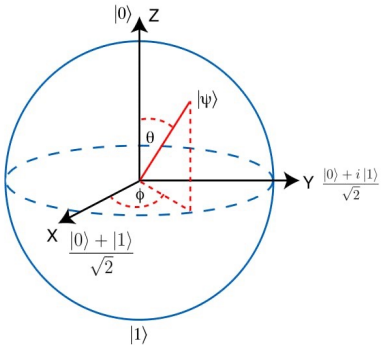
$$|\psi(\theta,\phi)\rangle = \cos\left(\frac{\theta}{2}\right)|0\rangle + e^{i\phi}\sin\left(\frac{\theta}{2}\right)|1\rangle \qquad 0 \le \theta \le \pi, \quad 0 \le \phi < 2\pi$$

The corresponding Bloch vector is:

$$\boldsymbol{r} = (\sin\theta\, cos\,\phi\,, si\,n\,\theta\, si\,n\,\phi\,, cos\,\theta), \qquad |\mathbf{r}| = 1$$

The angles θ and ϕ define a point on the Bloch sphere.

θ controls the population balance between $|0\rangle$ and $|1\rangle$.
ϕ controls the relative phase.



Bloch Sphere

The Bloch sphere is a geometric representation of a single pure qubit state.

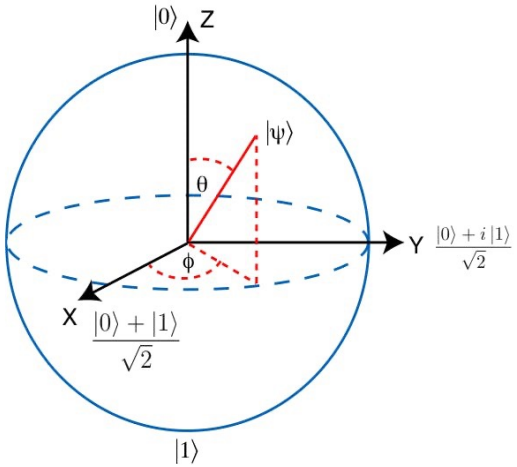
Bloch sphere

The surface of the Bloch sphere represents pure single-qubit states.

$|0\rangle$ is the **North pole**
$|1\rangle$ is the **South pole**.

Equatorial states ($\theta = \frac{\pi}{2}$) have equal populations and differ by relative phase.

$$|\psi\rangle = \frac{1}{\sqrt{2}}|0\rangle + \frac{e^{i\phi}}{\sqrt{2}}|1\rangle.$$



The Bloch sphere does not directly represent:

generic two-qubit states
entangled states
many-qubit Hilbert spaces

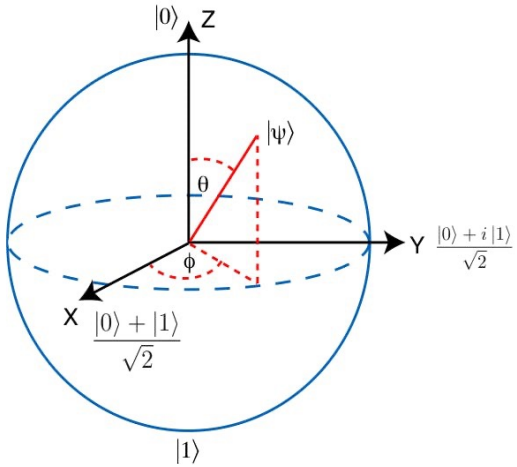
relative phase ϕ *and* **relative weight** θ are physically meaningful

Mixed single-qubit states live inside the Bloch ball.

Interference

The **surface** represents **pure states**.
That includes: $|0\rangle$ *and* $|1\rangle$ and all superpositions such as $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$.

Mixed single-qubit states live inside the Bloch ball.



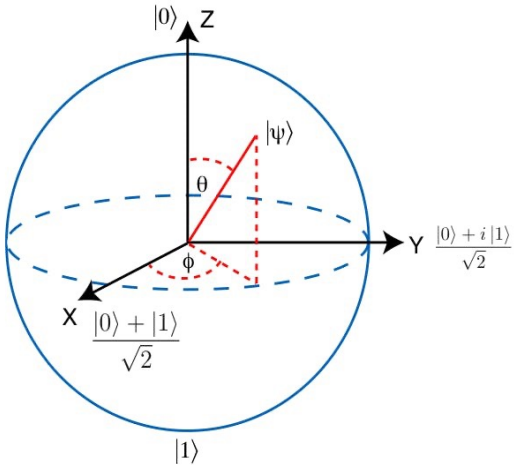
Interference requires coherence.

Pure superpositions have maximal coherence.
Mixed states may have reduced coherence.
Fully incoherent mixtures do not interfere.

Equatorial states

Equatorial states have equal populations:

$$P(0) = P(1) = 1/2$$



Therefore **their information is carried by the relative phase φ:**

$$|\psi\rangle = \frac{1}{\sqrt{2}}|0\rangle + \frac{e^{i\phi}}{\sqrt{2}}|1\rangle.$$

They are central for phase-sensitive experiments.

A π/2 pulse prepares an equatorial state
During free evolution, the state accumulates phase around the equator.
Loss of phase coherence appears as decay of the equatorial Bloch vector.

This is why Ramsey and dephasing measurements are naturally described using equatorial states.

Why phase matters (interference)

Because amplitudes are complex numbers, they have magnitude and phase:

$$\alpha = |\alpha|e^{i\phi_\alpha}, \qquad \beta = |\beta|e^{i\phi_\beta}$$

Magnitudes determine measurement probabilities.
Relative phase determines interference.

A global phase does not change the physical state:

But the relative phase between $|0\rangle$ and $|1\rangle$ is **observable**.

This is why a qubit is more than a probability distribution over 0 and 1.

Note: Bra and Ket notation

A ket represents a state vector:

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

$$|\psi\rangle = \begin{pmatrix}\alpha\\ \beta\end{pmatrix}, \qquad \alpha, \beta \in C$$

A bra is the Hermitian conjugate row vector (transpose and complex conjugate):

$$\langle\psi| = \alpha^{*}\langle 0| + \beta^{*}\langle 1|$$

$$\langle\psi| = (\alpha^{*}, \beta^{*})$$

The bra-ket gives an inner product (number):

$$\langle\phi|\psi\rangle$$

The ket-bra gives an outer product (operator/matrices):

$$|\psi\rangle\langle\phi|$$

Inner products and measurement amplitudes

The inner product between two states is $|\psi\rangle$ and $|\phi\rangle$:

complex amplitude $\langle\phi|\psi\rangle \in \mathbb{C}$

Complex probability amplitude

magnitude → contributes to probability
phase → controls interference with other amplitudes

If the system is in the state $|\psi\rangle$, then $\langle\varphi|\psi\rangle$ is the amplitude for finding it in the state $|\varphi\rangle$.

The measurement probability is the squared modulus:

For measurement in the computational basis:

$$P(0) = |\langle 0|\psi\rangle|^2$$

$$P(1) = |\langle 1|\psi\rangle|^2$$

Bra-ket notation therefore connects state vectors directly to measurement probabilities.

Outer product and projectors

A ket-bra is an outer product.

It gives an operator (act on the states): $|\psi\rangle\langle\phi|$

A special case is the **projector** onto a state:

$$\Pi_\psi = |\psi\rangle\langle\psi|$$

A projector extracts the component of a state along $|\psi\rangle$.

For the computational basis:

$$\Pi_0 = |0\rangle\langle 0|$$
$$\Pi_1 = |1\rangle\langle 1|$$

Π_0: "Is the qubit in $|0\rangle$?"

Π_1: "Is the qubit in $|1\rangle$?"

If $|\chi\rangle = \alpha|0\rangle + \beta|1\rangle$, then:

projectors **separate the two components** of the qubit state.

Projectors are operators

$$\Pi_\psi|\psi\rangle = |\psi\rangle$$
$$\Pi_\psi|\varphi\rangle = 0 \quad if \langle\psi|\varphi\rangle = 0$$

They keep the component parallel to $|\psi\rangle$ and remove the orthogonal component.

From state operators to density matrix

For a pure state, ρ has the same mathematical form as Π_ψ, but a different role.

Π_ψ is a measurement operator $\quad \Pi_\psi = |\psi\rangle\langle\psi|$

ρ is the state description $\quad \rho = |\psi\rangle\langle\psi|$

Here, ρ describes the quantum state, while Π_ψ is used as a measurement projector.

This matrix contains population and coherence terms

For an ideal pure qubit:

$$Tr(\rho^2) = 1$$

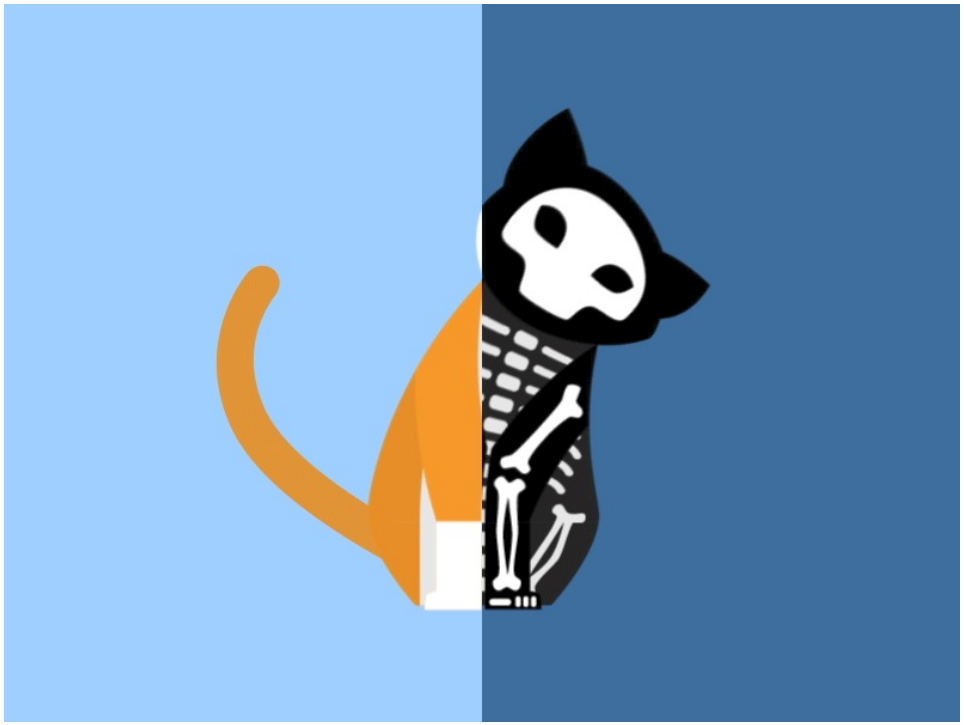
For real noisy qubits, the density matrix will become essential

Noise, decoherence, and imperfect measurement are naturally described using ρ.

Projectors experiments

Projectors are the **bridge between the abstract quantum state and the experimental outcome**.

	What is physically measured	**What the projector represents**
Superconducting qubit dispersive readout	microwave resonator phase/amplitude shift	projection onto $\|0\rangle$ or $\|1\rangle$
Spin-qubit spin-to-charge readout	charge sensor signal or tunneling event	projection onto $\|\downarrow\rangle$ or $\|\uparrow\rangle$
Pauli spin blockade	allowed or blocked charge transition	projection onto singlet-like or triplet-like subspace
Trapped-ion fluorescence readout	many photons or few photons	projection onto bright or dark internal state
Neutral-atom fluorescence imaging	atom present/absent or state-dependent fluorescence	projection onto selected atomic state



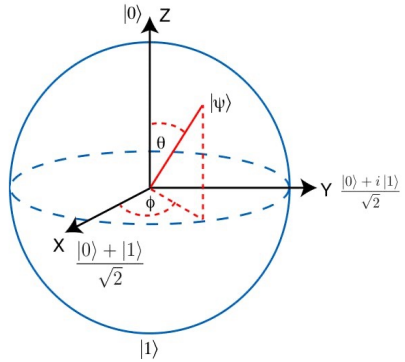
Credits: Kayla Mahoney

1. Qubits and Bloch Sphere
2. Pauli Matrices

Why do Pauli Matrices matter?

What are they? **Basic qubit operators**
→ standard toolbox for describing and manipulating a qubit.

What do they measure? **Qubit axis projections: X, Y, Z**
→ They allow us to reconstruct the Bloch vector and characterize the qubit state.



What do they do as gates? **Flip or phase states**
→ They implement basic single-qubit operations used inside quantum circuits.

How do they generate rotations? **Set rotation axes: X, Y, Z**
→ They define how microwave, laser, or electrical pulses control qubit motion.

How do they appear in Hamiltonians and errors? **Describe fields and noise**
→ They model energy splittings, interactions, dephasing, relaxation, and gate errors.

Pauli Matrices

The three Pauli matrices are:

$$X = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \qquad Y = \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix} \qquad Z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}$$

They are Hermitian (observable):

$$X = X^{\dagger}, \qquad Y = Y^{\dagger}, \qquad Z = Z^{\dagger}$$

They are also unitary (reversible):

$$X^{\dagger} X = I, \qquad Y^{\dagger} Y = I, \qquad Z^{\dagger} Z = I$$

Hermitian (self-adjoint) means they can represent **observables**.

Unitary means they can also act as **reversible quantum gates.**

As observables, they correspond to measurements.

As gates, they implement specific reversible qubit transformations.

From geometry to measurements

The Bloch sphere gives a geometric picture of a qubit.

To make this operational, we need operators and observables.

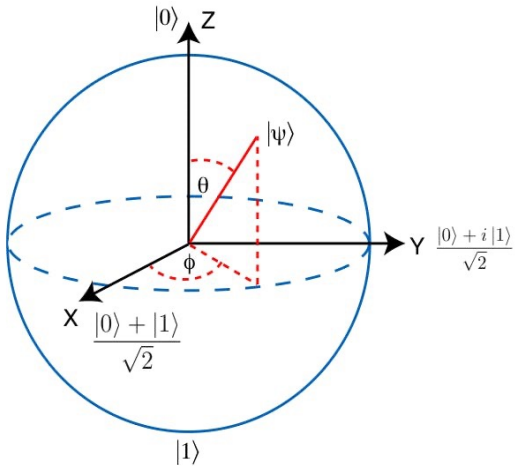
The Pauli matrices define the three fundamental measurement and control axes:

$$X, Y, Z$$

They correspond to the x, y, and z axes of the Bloch sphere.

The components of the Bloch vector are expectation values:

$$r = (\langle X \rangle, \langle Y \rangle, \langle Z \rangle)$$

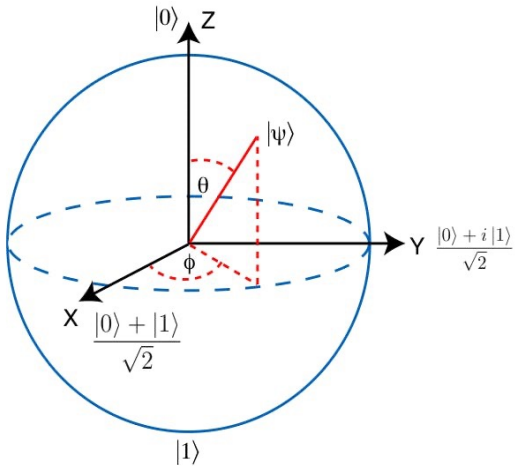


This connects the abstract state to measurable quantities.
Pauli matrices are therefore the bridge between:
abstract qubit states → physical measurements, gates, and dynamics.

Pauli measurements and Bloch vectors

For a single-qubit state, the Bloch vector is defined by Pauli expectation values:

$$\vec{r} = (\langle X \rangle, \langle Y \rangle, \langle Z \rangle)$$



Each component is experimentally measurable.

$\langle X \rangle$ measures the projection along the x-axis.
$\langle Y \rangle$ measures the projection along the y-axis.
$\langle Z \rangle$ measures the projection along the z-axis.

The **density matrix** can be written as:

$$\rho = \frac{1}{2}(I + \langle X \rangle X + \langle Y \rangle Y + \langle Z \rangle Z)$$

This connects the abstract state to measurable quantities

What do measurements X, Y and Z ask?

A Pauli measurement asks a binary question about the qubit state.

Z measurement:
Is the qubit closer to $|0\rangle \; or \; |1\rangle$?

X measurement:
Is the qubit closer to $|+\rangle \; or \; |-\rangle$?

Y measurement:
Is the qubit aligned with the +Y or −Y direction on the Bloch sphere?

The measurement outcomes are always: $+1 \; or \; -1$

But the expectation value is an average over many repetitions:

$$\langle X \rangle, \quad \langle Y \rangle, \quad \langle Z \rangle$$

This is how Bloch-sphere coordinates are measured experimentally.

Pauli Matrices do not commute (order matters)

Different Pauli matrices generally do not commute:

$[X, Y] = 2iZ \qquad [Y, Z] = 2iX \qquad [Z, X] = 2iY$

This means that the order of operations matters.

Applying X and then Y is not the same as applying Y and then X.

Physically, this reflects the fact that measurements and rotations along different axes are incompatible.

For example:

measuring Z gives information about the computational basis
measuring X gives information about a different basis

A qubit cannot have definite values of X, Y, and Z at the same time.

Pauli Matrices as gates

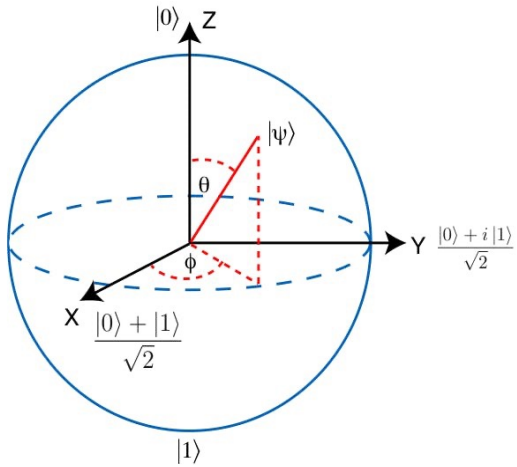
The Pauli matrices can also act as single-qubit gates.

The X gate **flips the computational basis states**: $X|0\rangle = |1\rangle$ $X|1\rangle = |0\rangle$

The Z gate **changes the relative phase**: $Z|0\rangle = |0\rangle$ $Z|1\rangle = -|1\rangle$

The Y gate **combines a bit flip with a phase change**:

$$Y|0\rangle = i|1\rangle, \quad Y|1\rangle = -i|0\rangle$$



Evolution (Unitary operators)

For a time-independent Hamiltonian, the Schrödinger equation gives:

$$|\psi(t)\rangle = U(t)|\psi(0)\rangle$$

$$U(t) = e^{-iHt/\hbar}$$

The operator U(t) is unitary.

This means that time evolution preserves normalization:

$$\langle\psi(t)|\psi(t)\rangle = 1$$

On the Bloch sphere, single-qubit unitary evolution corresponds to a rotation of the state vector.

Pauli Matrices generate rotations

If the Hamiltonian is proportional to a Pauli matrix, the unitary evolution is a rotation:
Pauli matrices generate continuous single-qubit rotations.

A rotation around the x-axis is: $R_x(\theta) = e^{-i\frac{\theta}{2}X}$

A rotation around the y-axis is: $R_y(\theta) = e^{-i\frac{\theta}{2}Y}$

A rotation around the z-axis is: $R_z(\theta) = e^{-i\frac{\theta}{2}Z}$

Pauli Matrices Generate Single-Qubit Rotations

The angle θ determines how far the qubit rotates.

The Pauli operator determines the rotation axis.

This is why Pauli matrices are the basic language of single-qubit control.

Hamiltonians from Pauli operators

Any single-qubit Hamiltonian can be written as a linear combination of the identity and Pauli operators:

$$H = aI + bX + cY + dZ$$

The identity term shifts all energy levels equally.
The Pauli terms determine energy splittings and dynamics.

For example:

$$H = \frac{\hbar\Omega}{2}X$$

generates rotations around the x-axis.

The corresponding time evolution is:

$$U(t) = e^{-i\frac{Ht}{\hbar}}$$

Substituting the Hamiltonian gives

$$U(t) = e^{-i\frac{\Omega t}{2}X} = R_x(\Omega t)$$

Controlling the Hamiltonian means controlling the qubit rotation

Ω Rabi frequency

Pauli errors

Pauli operators also provide a simple language for quantum errors.

An X error is a bit-flip error: $|0\rangle \leftrightarrow |1\rangle$

A Z error is a phase-flip error: $|1\rangle \rightarrow -|1\rangle$

A Y error combines bit flip and phase flip.

Many realistic errors are more complicated than a single Pauli error.
However, Pauli errors are extremely useful because they form a basis for describing and analyzing noise.

This is why Pauli operators appear throughout quantum computing:

Measurements, gates, rotations, Hamiltonians, noise, error correction…

A qubit is not only a two-level quantum system.

A useful physical qubit must be controllable:

we must be able to prepare it, preserve coherence, apply gates, and measure it.

In Hamiltonian language, control means being able to engineer rotations around the Bloch sphere.

Without controllable rotations, a two-level system cannot function as a practical qubit for quantum computing.

Measurements

Measurement converts quantum information into classical information.

For a single qubit:

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

a measurement in the computational basis gives:
0 with probability $|\alpha|^2$, 1 with probability $|\beta|^2$

After measurement, the result is classical.

The measurement outcome is probabilistic, but the probabilities are determined by the amplitudes.

Measurements in Bloch coordinates

Measurement is basis-dependent.

In many physical platforms, **the natural readout is effectively a Z-basis measurement**.

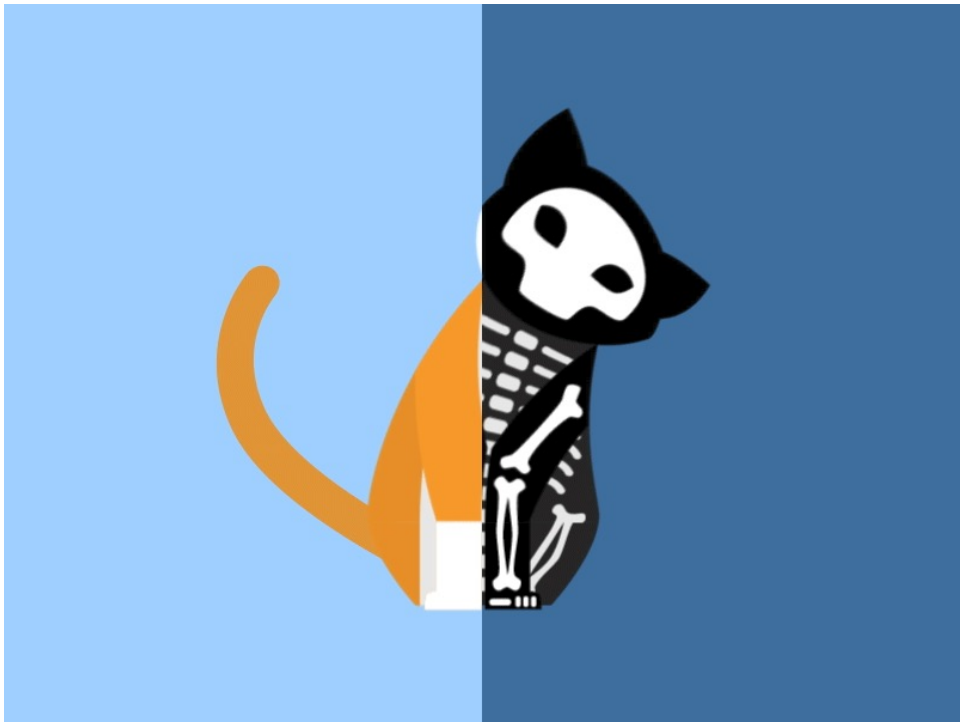
To measure on a different basis, we first rotate the qubit and then perform a fixed-Z readout.

Example: to measure in the X basis, apply a Hadamard gate and then measure in the computational basis.

Hadamard gate H maps the X basis to the Z basis:

$$H|+\rangle = |0\rangle, \qquad H|-\rangle = |1\rangle$$

Hadamard converts the X-basis questions (and measurements) into a Z-basis measurement



Credits: Kayla Mahoney

1. Qubits and Bloch Sphere
2. Pauli Matrices
3. Multi-qubits

Tensor product

The tensor product combines quantum systems into one joint Hilbert space.

For **two qubits**:

$$\mathcal{H}_{\mathcal{AB}} = \mathcal{H}_{\mathcal{A}} \otimes \mathcal{H}_{\mathcal{B}}$$

A product state is written as: $|\psi\rangle_A \otimes |\phi\rangle_B$

Examples:

$$|0\rangle \otimes |0\rangle = |00\rangle, \qquad |0\rangle \otimes |1\rangle = |01\rangle$$

The dimension multiplies:

$$2 \times 2 = 4$$

Therefore, two qubits have four basis states:

$$|00\rangle, \qquad |01\rangle, \qquad |10\rangle, \qquad |11\rangle$$

This is the starting point for multi-qubit states and multi-qubit operators.

The tensor product is the mathematical operation used to combine quantum systems.

For two qubits, each qubit has a two-dimensional Hilbert space.

The joint system is not a sum, but a tensor product

The tensor product is what makes the joint state space larger than the state space of each individual qubit.

Two Qubits

For two qubits, the computational basis has four states:

$$|00\rangle, \quad |01\rangle, \quad |10\rangle, \quad |11\rangle$$

A generic two-qubit pure state is:

$$|\psi\rangle = c_{00}|00\rangle + c_{01}|01\rangle + c_{10}|10\rangle + c_{11}|11\rangle$$

with normalization:

$$|c_{00}|^2 + |c_{01}|^2 + |c_{10}|^2 + |c_{11}|^2 = 1$$

The coefficients c_{ij} are complex amplitudes.

Measurement returns one classical bit string, with probabilities given by the squared amplitudes.

n Qubits

For n qubits, the computational basis contains 2^n bit strings.

$$|\psi\rangle = \sum_{x=0}^{2^n-1} c_x|x\rangle, \qquad \sum_x |c_x|^2 = 1$$

The state space grows exponentially with n.
This exponential state space is the kinematic resource behind quantum computation.

But useful computation requires more than a large state space: controlled dynamics, interference, entanglement and measurement.

The Hilbert space is a tensor product of n qubit spaces:

$$\mathcal{H}_{2^n} = \mathcal{H}_2 \otimes \cdots \otimes \mathcal{H}_2$$

This tensor-product structure defines local operations, subsystems, and correlations.

Multi-qubit Pauli operators

Multi-qubit operators are built using tensor products.

For two qubits:

$$X_1 = X \otimes I, \qquad Z_2 = I \otimes Z$$

$$Z_1 Z_2 = Z \otimes Z, \qquad X_1 X_2 = X \otimes X$$

The subscript indicates which qubit each Pauli operator acts on.

Examples:
X_1 acts on qubit 1 only.

Z_2 acts on qubit 2 only.

$Z_1 Z_2$. acts on both qubits and represents a two-qubit correlation or interaction.

Pauli operators in many-qubits systems

For many qubits, Pauli operators act on specific qubits.

Examples:

X_1 acts with X on qubit 1
Z_2 acts with Z on qubit 2

Z_1Z_2 describes a two-qubit interaction
X_1X_2 describes another type of two-qubit coupling

This is the language used to describe coupling, entangling gates, noise, and quantum simulation.

Many-qubit Hamiltonians are often built from tensor products of Pauli operators.

Examples:

$$H = JZ_1Z_2$$
$$H = JX_1X_2$$
$$H = h_1Z_1 + h_2Z_2 + JZ_1Z_2$$

Tensor products as many qubit operators

A typical two-qubit Hamiltonian may contain local and interaction terms:

$$H = JZ_1Z_2 + hX_1$$

The term JZ_1Z_2 represents an interaction between the two qubits.

The term hX_1 represents a local drive on qubit 1.

For n qubits, tensor products of I, X, Y, Z form Pauli strings:

$$P = P_1 \otimes P_2 \otimes \cdots \otimes P_n, \qquad P_i \in \{I, X, Y, Z\}$$

There are 4^n Pauli strings.

Hamiltonians, error models, stabilizers, and observables are often written in this language.

Entanglement

Entanglement means that the correlations belong to the joint state, not to pre-existing independent states of the subsystems.

A two-qubit state is separable if it can be written as a product of two single-qubit states:

$$|\psi\rangle = |\psi_1\rangle \otimes |\psi_2\rangle$$

A state is entangled if this factorization is not possible.

Entanglement means that the joint state contains correlations that cannot be reduced to independent states of the individual qubits.

In other words: the two-qubit system must be described as a whole.

Bell state (entanglement)

A standard entangled state is

Bell state: $|\Phi^+\rangle = \frac{|00\rangle+|11\rangle}{\sqrt{2}}$

This state cannot be written as: $|\psi_1\rangle \otimes |\psi_2\rangle$

If both qubits are measured in the computational basis, their outcomes are perfectly correlated:

if the first outcome is 0, the second is also 0
if the first outcome is 1, the second is also 1

But before measurement, the state is not “00 or 11”; it is a coherent superposition of both.

Two basic gates: Hadamard and CNOT

The Hadamard gate is a **single-qubit gate**

It maps computational-basis states on the Z axis to superposition states on the X axis:

$$H|0\rangle = |+\rangle \qquad H|1\rangle = |-\rangle$$

Z basis: $|0\rangle$, $|1\rangle$
X basis: $|+\rangle$, $|-\rangle$

$$H|0\rangle = \frac{|0\rangle + |1\rangle}{\sqrt{2}}, \qquad H|1\rangle = \frac{|0\rangle - |1\rangle}{\sqrt{2}}$$

Hadamard alone does not create entanglement.
It creates superposition on individual qubits.

To create entanglement, you need a **two-qubit gate** or interaction, such as CNOT.

Two basic gates: Hadamard and CNOT

The CNOT gate is a **two-qubit gate**

The first qubit is the control.
The second qubit is the target.

If the control qubit is $|0\rangle$, the target is unchanged.
If the control qubit is $|1\rangle$, the target is flipped.

CNOT does **not always** create entanglement.

It can create entanglement **when the control qubit is in superposition**.

The CNOT maps:

$|00\rangle \rightarrow |00\rangle$
$|10\rangle \rightarrow |11\rangle$

Therefore:

$$\frac{|00\rangle + |10\rangle}{\sqrt{2}} \rightarrow \frac{|00\rangle + |11\rangle}{\sqrt{2}}$$

The CNOT maps:
the superposition
of $|00\rangle$ and $|10\rangle$
into
a superposition of $|00\rangle$ and $|11\rangle$.

Preparing a Bell state (entanglement)

Start from two qubits in the computational basis state: $|00\rangle$

- First, apply a Hadamard gate to qubit 1.

H+CNOT=Bell State

This **creates a superposition of the first qubit**:

$$\frac{|00\rangle + |10\rangle}{\sqrt{2}}$$

- Then apply a CNOT gate, with qubit 1 as control and qubit 2 as target.

H creates
branch 1: $|00\rangle$
branch 2: $|10\rangle$

CNOT correlates
$|00\rangle \rightarrow |00\rangle$
$|10\rangle \rightarrow |11\rangle$

The **Hadamard creates the two coherent branches, and CNOT correlates them.**
The entanglement (Bell State) is the result.

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

Key ingredients:
superposition on one qubit
two-qubit entangling gate

Quantum circuits

A quantum circuit represents a quantum computation as a sequence of controlled operations on qubits.

It starts from an initialized state, applies gates, and ends with a measurement.

Example:

$$|00\rangle \rightarrow H^1 \rightarrow CNOT^{1,2} \rightarrow measurement \rightarrow output \quad 00 \ \ or \ \ 11$$

Before measurement, ideal gates are unitary: they transform the state coherently and reversibly while preserving total probability.

The gates create the quantum state.
The measurement converts it into classical bits.

Quantum circuits

A quantum circuit represents computation as a sequence of controlled operations.

$$|0\rangle^{\otimes n} \rightarrow \text{unitary gates} \rightarrow \text{measurement}t \rightarrow cla\text{ssical output}t$$

$|0\rangle^{\otimes n}$ means n qubits, all initialized in the ground state $|0\rangle$

Main ingredients:

initialization/state preparation
single-qubit gates
two-qubit gates
measurement
classical post-processing

In the circuit model, gates are unitary operations; in the laboratory, they are implemented by calibrated physical controls.

Quantum circuits

A quantum circuit represents a quantum computation as a sequence of controlled operations on qubits.

It starts from an initialized state, applies single-qubit and two-qubit gates, and ends with measurement.

Before measurement, ideal gates are unitary operations transforming the quantum state coherently and reversibly while preserving total probability (no collapse).

X flips $|0\rangle$ and $|1\rangle$ Y flips $|0\rangle$ and $|1\rangle$ and adds a phase
Z changes the relative phase between $|0\rangle$ and $|1\rangle$.

R_x, R_y, R_z rotate a qubit around the x, y, or z axis.

H creates equal coherent superpositions
CNOT couples two qubits and can create entanglement

Measurement converts the final quantum state into a classical bit string.
Unlike unitary gates, measurement generally destroys the original superposition.

Physical implementation

To build a quantum computer, one must implement the full operational stack:

initialize
control
couple
measure
reset
scale

At the hardware level, this requires:

reliable two-level systems
high-fidelity gates
isolation from noise
error characterization
eventually, error correction

DiVincenzo Criteria

The DiVincenzo criteria summarize what a physical quantum computer must provide:

1. scalable, well-characterized qubits

2. initialization to a simple fiducial state

3. coherence times long compared with gate times

4. a universal set of quantum gates

5. qubit-specific measurement

The central hardware challenge is to preserve quantum coherence while still allowing control, coupling, and measurement.

DiVincenzo Criteria - Communication

For quantum communication, two additional criteria are added:

6. interconversion between stationary and flying qubits

7. faithful transmission of flying qubits

Once the formalism is clear, the next question is experimental:

How do we measure T_1, T_2, gate fidelity, readout fidelity, leakage, and crosstalk in a real device?

These metrics tell us whether an abstract qubit can become a useful physical qubit.

Qubits: Summary

A qubit is a controllable physical two-level quantum system.

Its state is described by complex amplitudes, whose magnitudes and phases determine measurement outcomes.

The **Bloch sphere** gives a geometric picture of a single qubit.

Pauli matrices provide the operator language for measurement axes, control axes, and rotations.
Hamiltonians built from Pauli operators generate unitary time evolution.

Gates are calibrated physical operations that implement these rotations.
Measurement converts the final quantum state into classical information.

Many-qubit systems are built with tensor products and can exhibit entanglement.
Quantum circuits organize these elements into computations.

The circuit model becomes useful only when it can be implemented in physical hardware.

Quantum Computing

Quantum computing is scientifically interesting because it combines quantum many-body physics, control, measurement, information theory, and engineering

A quantum processor is a **controllable quantum many-body system**.

Its power does not come from reading out all amplitudes,
but from engineering dynamics so that
interference amplifies useful outcomes and suppresses unwanted ones.

References

M. A. Nielsen and I. L. Chuang, Quantum Computation and Quantum Information, Cambridge University Press.

J. Preskill, Lecture Notes on Quantum Computation, Caltech.

M. M. Wilde, Quantum Information Theory, Cambridge University Press.

IBM Quantum Learning, modules on linear algebra and single-qubit gates.

D. P. DiVincenzo, “The Physical Implementation of Quantum Computation,” Fortschritte der Physik 48, 771–783 (2000).

Thank you
for your attention

Maria Longobardi

Maria.longobardi@unibas.ch