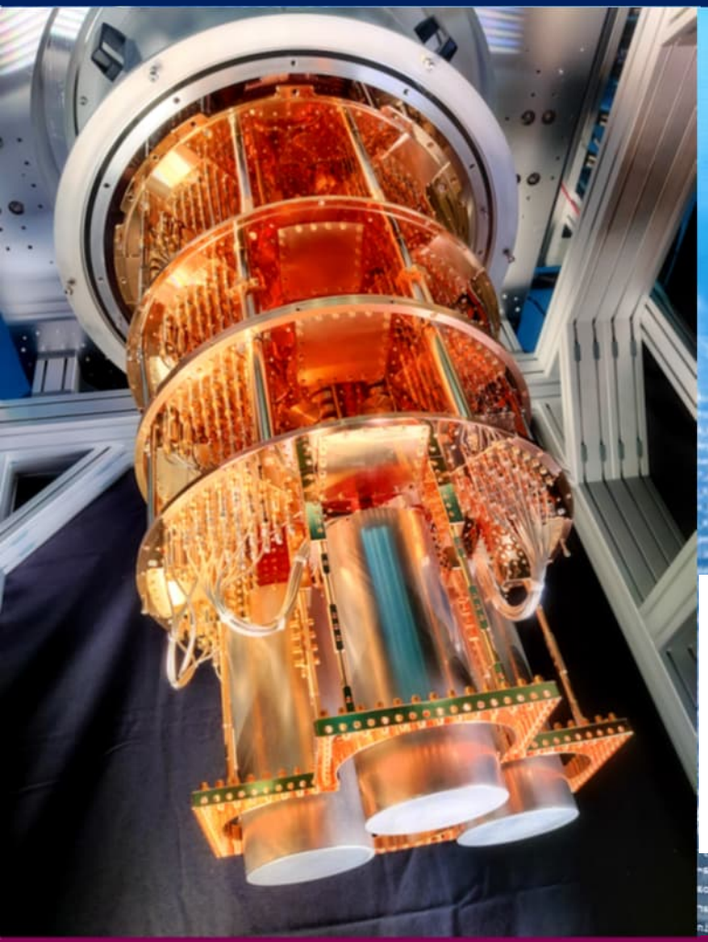
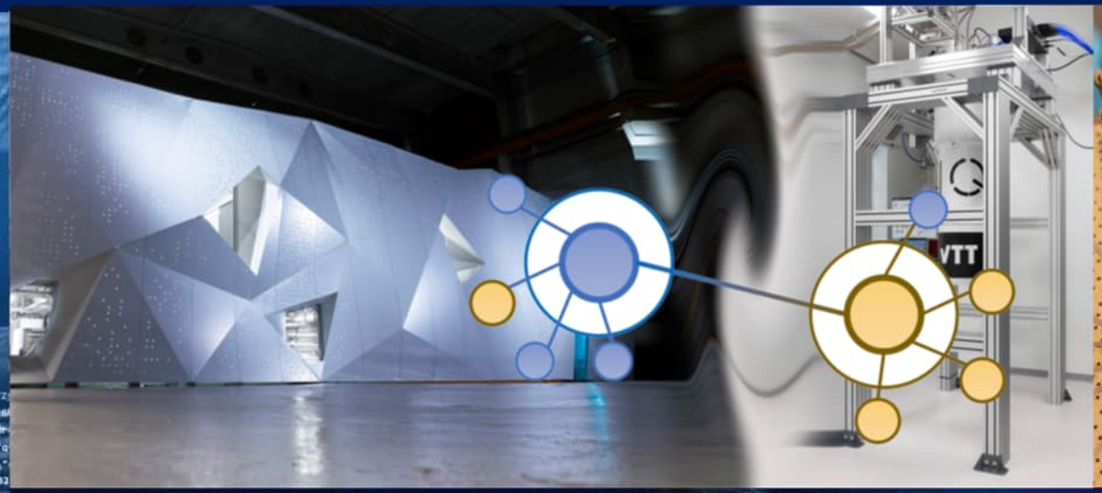
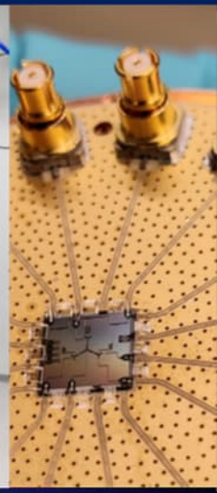
Introduction to quantum computing and FiQCI 2nd Ed.

25–26 November 2024 / Mikael Johansson CSC







Suomen Akatemia
Finlands Akademi
Research Council of Finland

Funded by the European Union
NextGenerationEU

EURO²

Day 1 Tentative Schedule

9:00 *Morning session, includes 2x15 minute breaks as suitable*
Introduction to quantum computing and programming
Hands-on 0: Superposition and measurement

12:00 *Lunch*

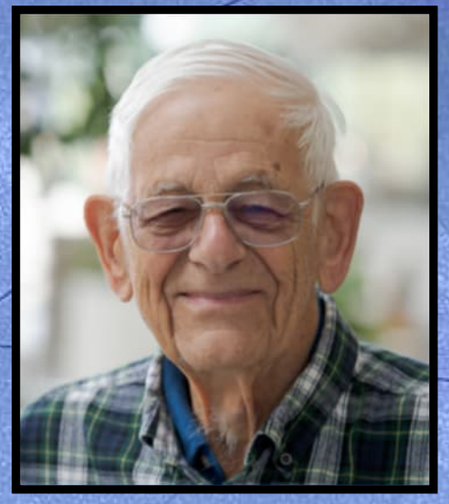
13:15 *Afternoon session, includes 2x15 minute breaks as suitable*
Hands-on 0: Superposition and measurement
Hands-on 1: EPR pairs
Hands-on 2: Parameterized gates

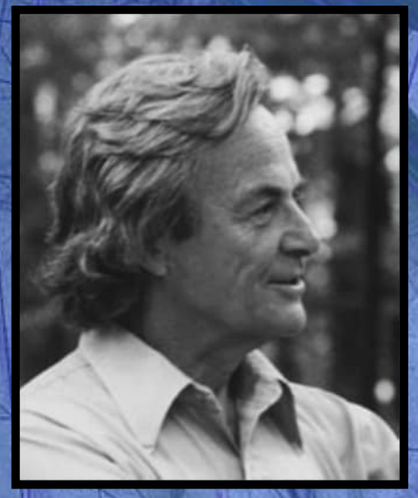
16:00 *End of Day 1*

Day 2 Tentative Schedule

9:00 *Morning session, includes 2x15 minute breaks as suitable*
Physical quantum computers
Hands-on 3: Transpilation of quantum circuits
The Deutsch algorithm
12:00 *Lunch*
13:15 *Afternoon session, includes 2x15 minute breaks as suitable*
Hands-on 4: The Deutsch algorithm
The Quantum Approximate Optimization Algorithm (QAOA)
Hands-on 5: QAOA
Wrapping up
16:00 *End of Day 2*

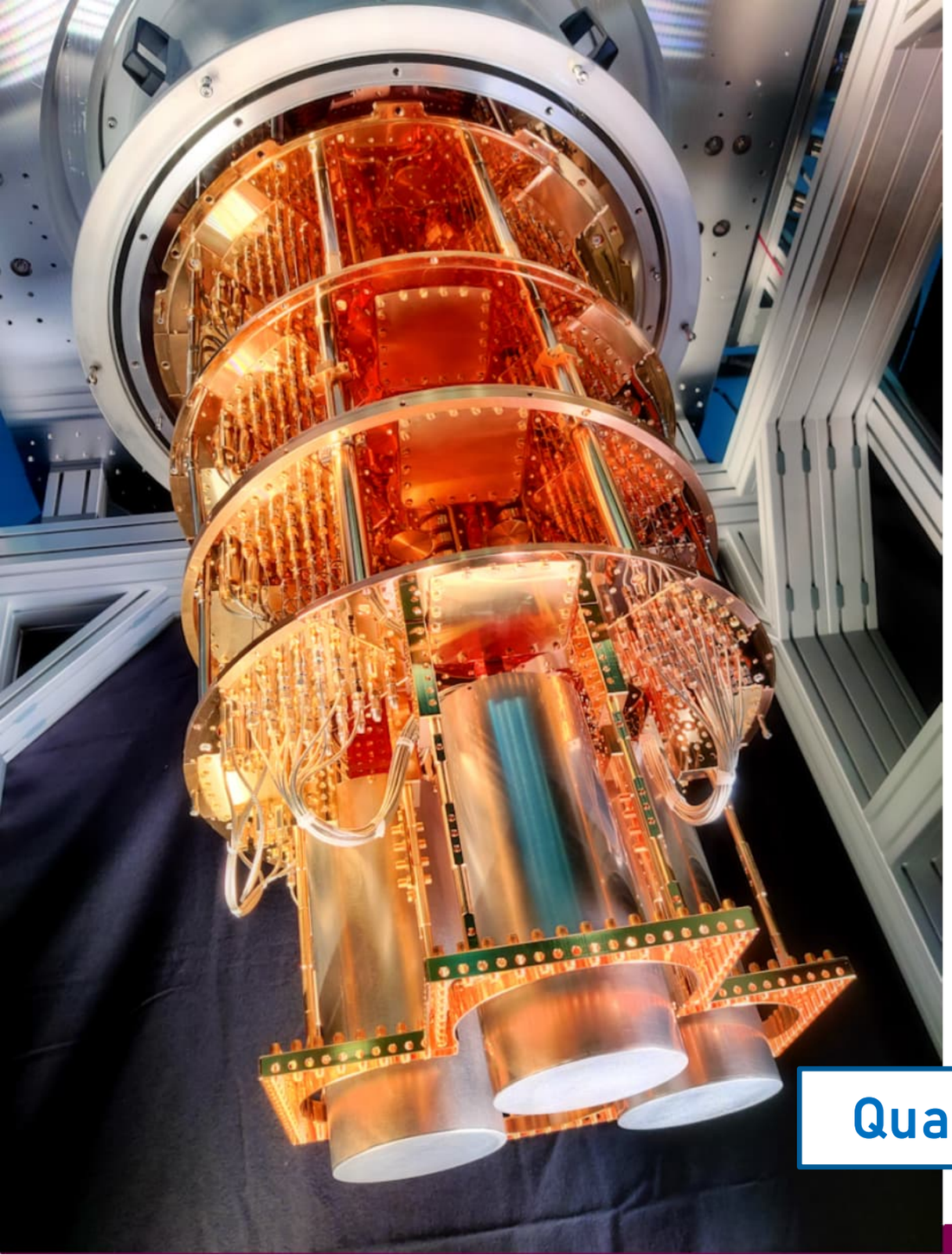
Quantum computing






Photos by:
Justinhsb
Paul Halmos
Tamiko Thiel
Lulie Tanett

- Concrete ideas at the break of 1970-1980
- **Paul Benioff, Yuri Manin, Richard Feynman, David Deutsch**:
  As the physical world is inherently quantum mechanical, simulating it with classical computers is inefficient – and will always be so!
- Also simulating the brain with quantum computers an early idea



What is a quantum computer?

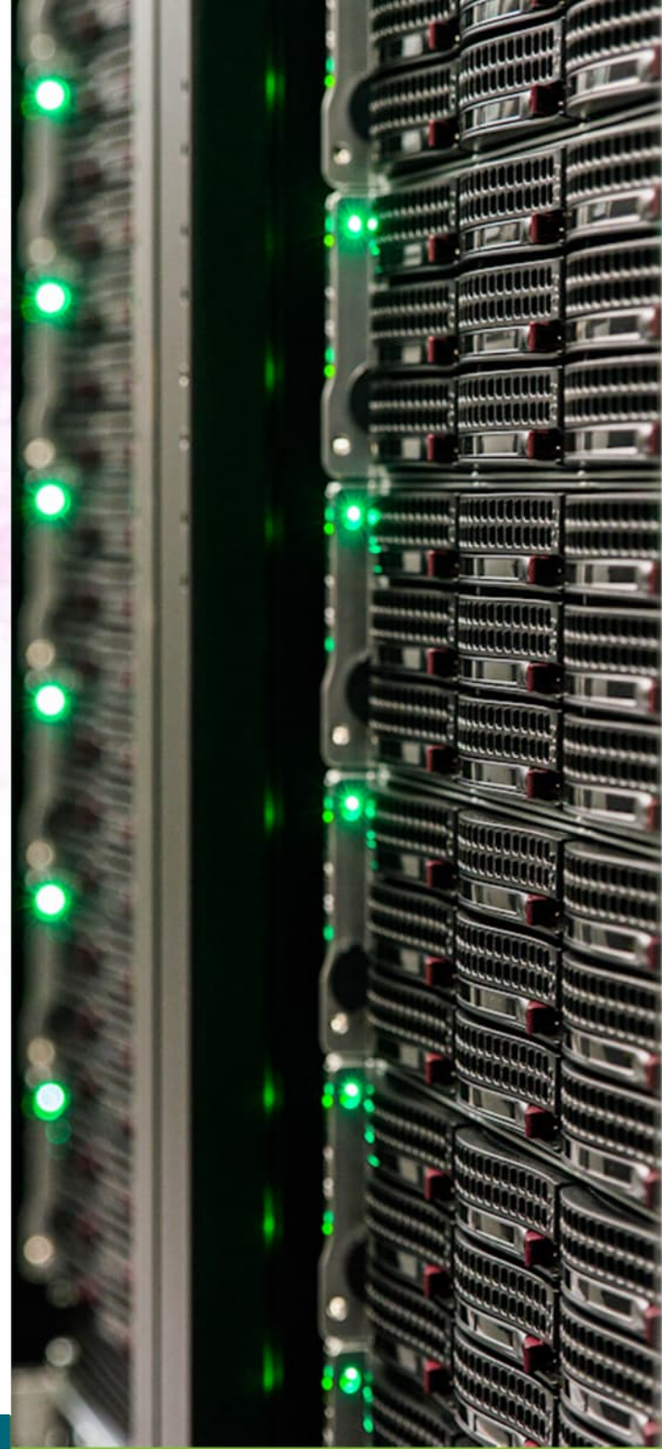
- A quantum computer is a device that *directly* exploits quantum mechanical phenomena to perform a calculation
  - Superposition, entanglement, interference, …

- This enables **extremely powerful solutions** to *certain* types of problems and **enables completely new science**

Quantum computer by IQM

What is a quantum computer *not*?

- A quantum computer is *not* a superfast version of a standard computer – **it is different**
  - For example programming a quantum computer is radically different from programming an ordinary computer
- In a classical computer, all information is represented using **bits** that are either one or zero: `11110111000`
  - When a computer does something, the values change: `1` → `0`; `0` → `1`, …
- A computer program is "just" a recipe for *which* bits to flip and *when*



How could bits be improved?

To be and not to be, that is the anſwer!

'The harlots cheek beautified with plaiſting art,
'Is not more ugly to the thing that helps it,
'Than is my deed to my moſt painted word:
'O heavy burden!"

[*Enter* Hamlet.

Pol. I hear him coming, withdraw my Lord.

Ham. To be or not to be, that is the queſtion,
Whether 'tis nobler in the mind to ſuffer
The ſlings and arrows of outragious fortune,
Or to take arms againſt a Sea of troubles,

CSC

Qubits

- A quantum computer exploits the laws of quantum mechanics:

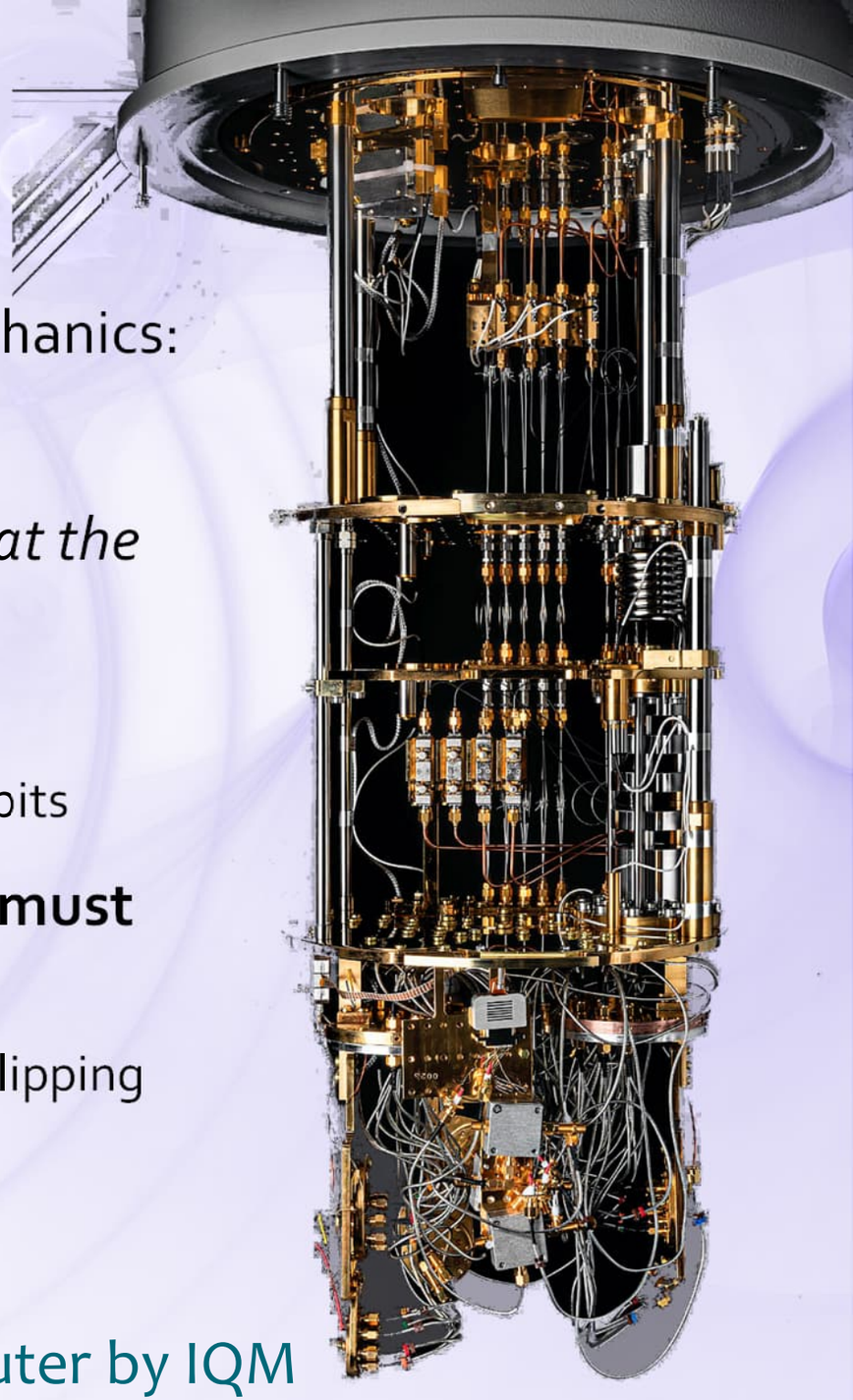
  uses quantum bits, qubits

- Qubits can be one and zero and everything in-between *at the same time* (superposition)

  $|0\rangle$, $|1\rangle$, $\alpha|0\rangle+\beta|1\rangle$, …

  - Thus, **significantly more versatile and powerful** than classical bits

- When programming a quantum computer, one can and **must** take advantage of this flexibility!
  - Makes programming much more difficult, compared to simply flipping bits **$1 \rightarrow 0 \rightarrow 1 \rightarrow \cdots$**
  - In addition, we also need to utilise entanglement and wave function phases



Quantum computer by IQM

Superposition

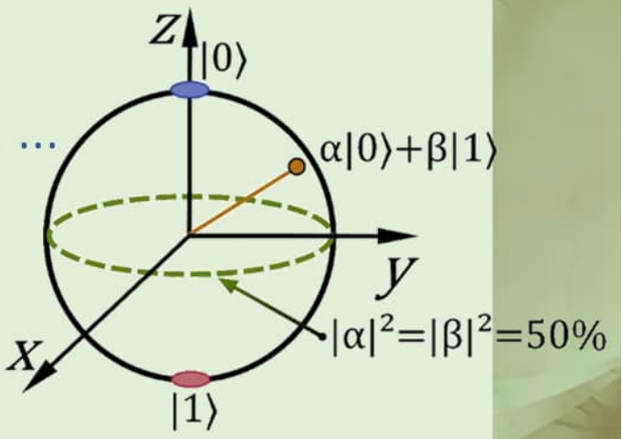
Qubits can be in a quantum mechanical **superposition** of all values simultaneously

The difference between [qu]bits grows more pronounced with increasing count:

2 bits can describe **4** different states: **`00, 01, 10, 11`**
2 qubits can describe all **4** states *at the same time*

3 bits can describe $2^3 = 2\times2\times2 =$ **8** different states: **`000, 001, 010, 011,`** …
3 qubits can describe all **8** states *at the same time*

20 qubits can describe a ***million*** states, *etc…*



The different states can represent different inputs, on which the computer performs a calculation

Quantum Peculiarities in Two Slides

At the scale of the very small, quantum effects instead of “common sense”

- Molecules and atoms *are* small:

Wave/particle duality

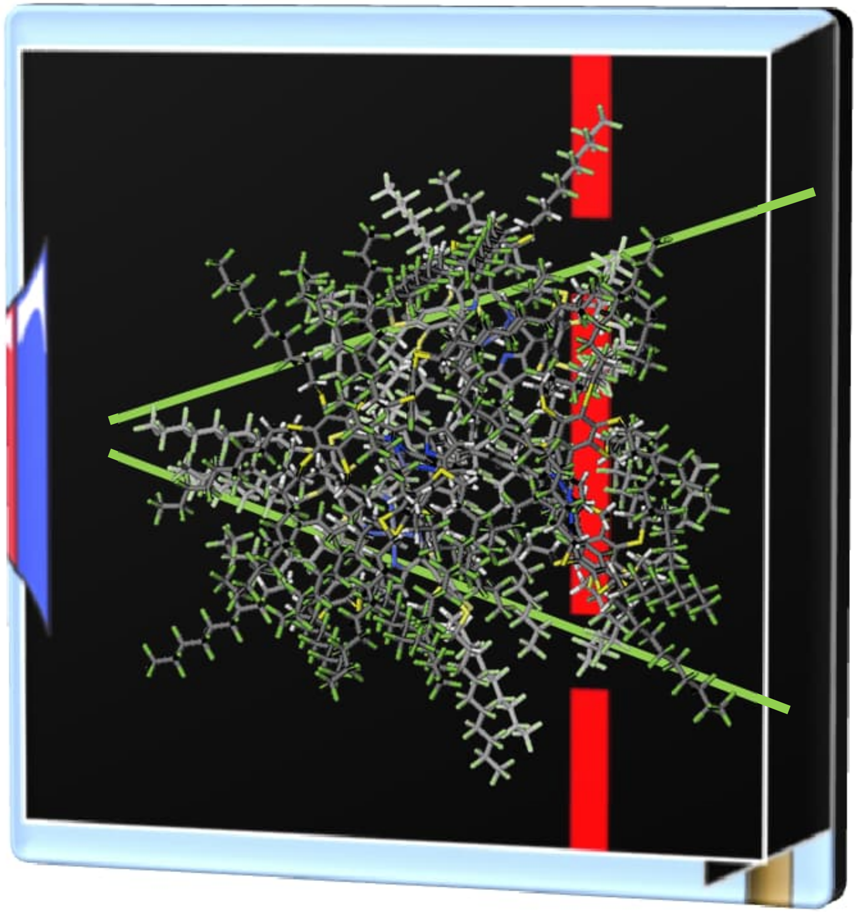
- light (photons) and particles (electrons, atoms, molecules, …) behave both like waves and individual particles

Quantum Peculiarities in Two Slides

- The famous **double-slit experiment** demonstrates three of the quantum phenomena important for quantum computing:

1. **Superposition**
   - Particles go through *both slits at the same time*!

     Photons, electrons, fullerenes, macromolecules, …
2. **Effect of measurement** (observation)
   - When measuring which slit they go through, they *stop* going through both at the same time!
3. **Interference**
   - Constructive and destructive



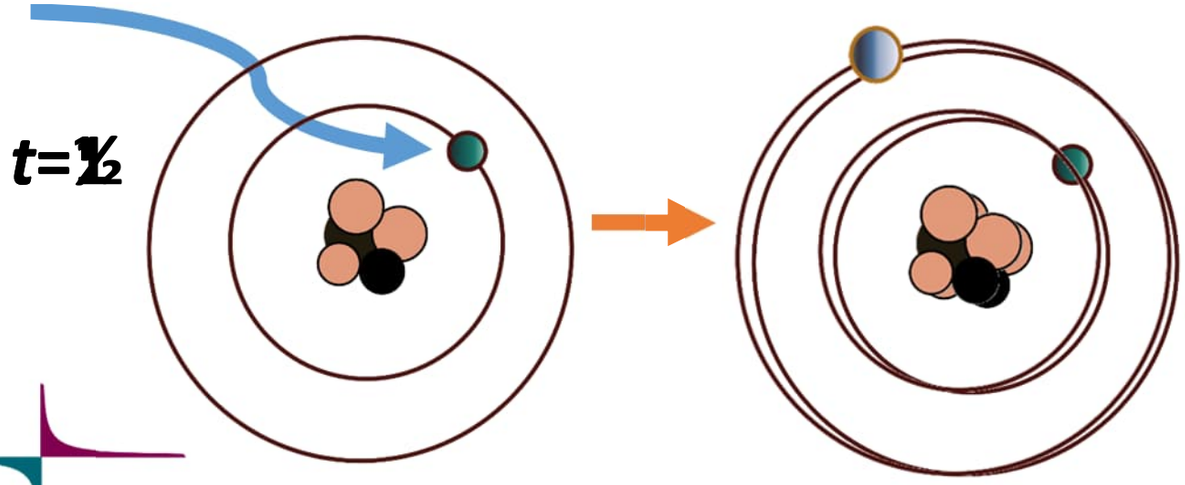
PhET Interactive Simulations
University of Colorado Boulder
https://phet.colorado.edu

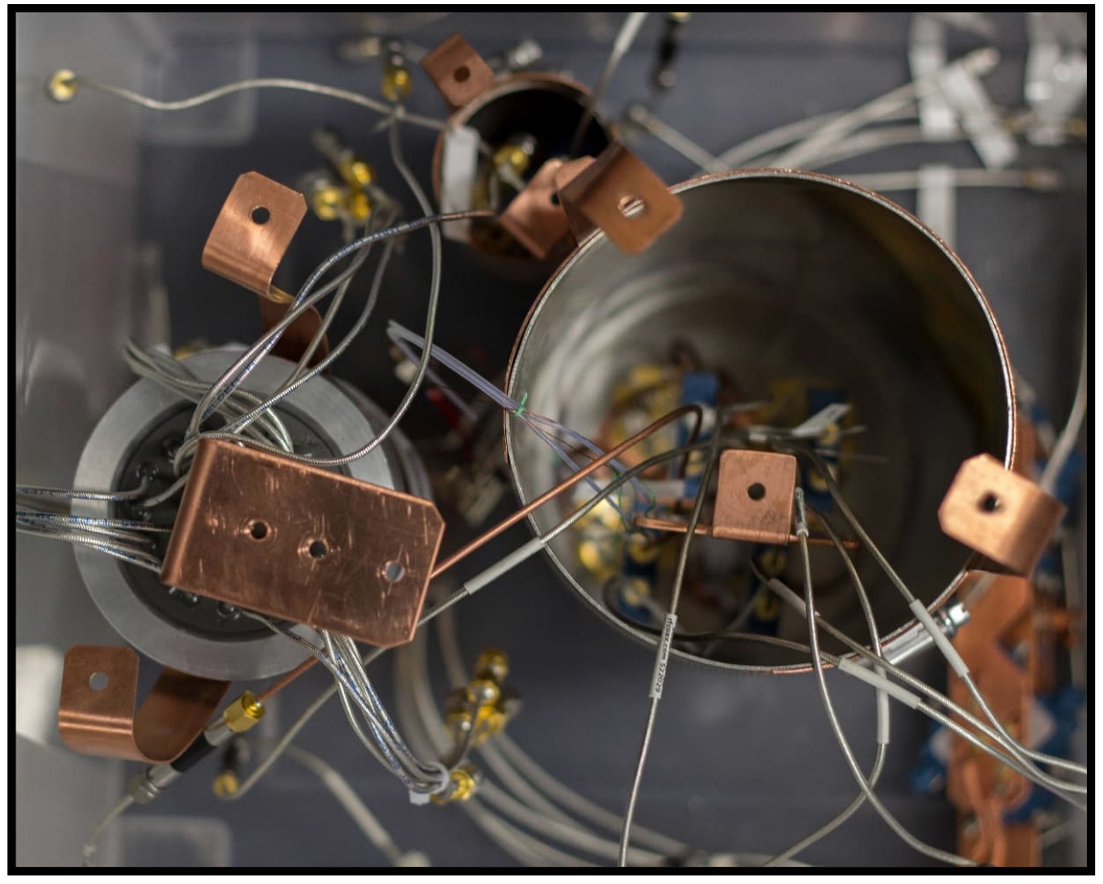
Content provided by
Yaakov Fein
University of Vienna

What does a qubit look like?

A qubit is ***any system*** that can be in a super-position of two states, denoted $|0\rangle$ and $|1\rangle$

For example, ground and excited state of an atom, ion, or molecule





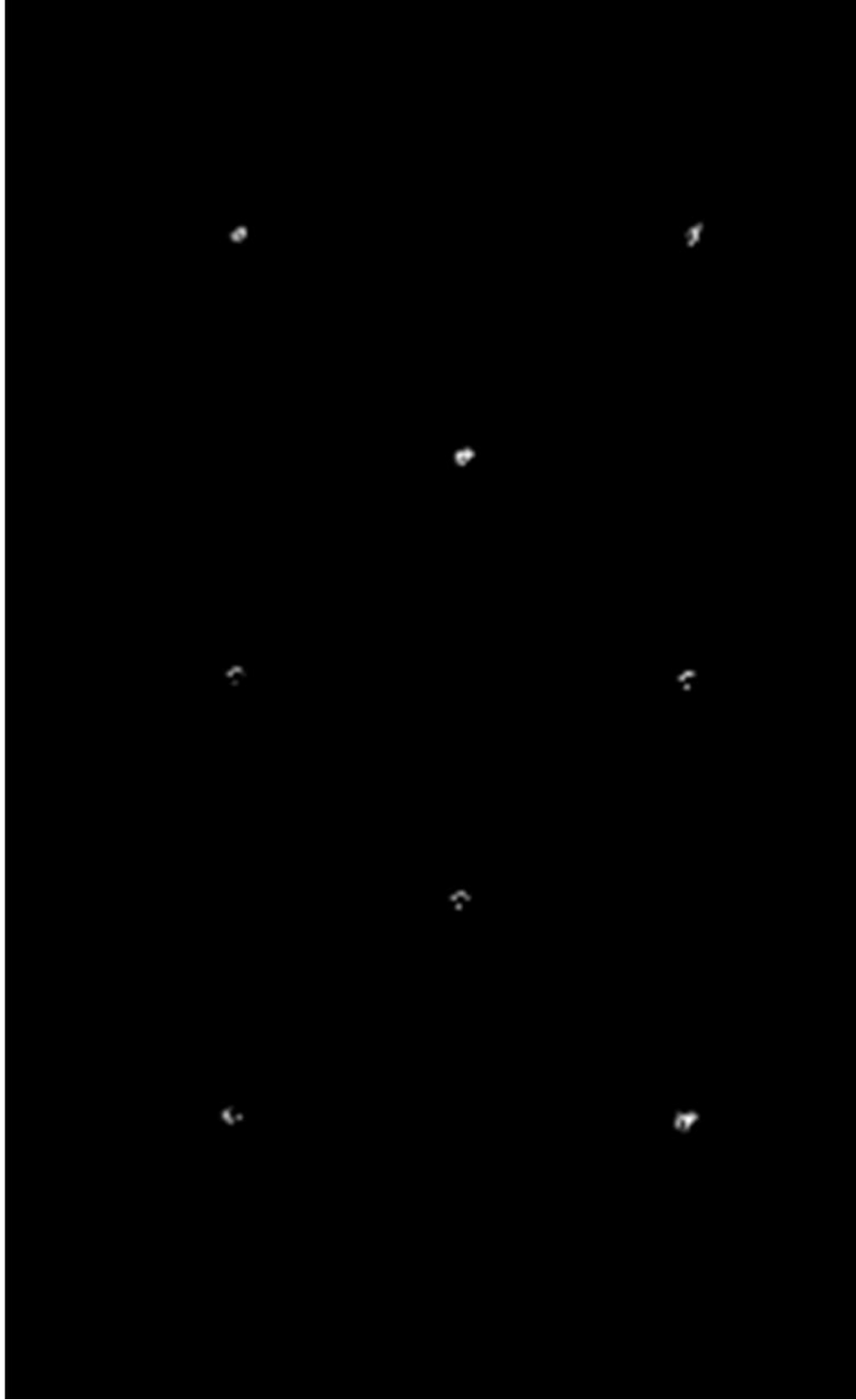
Microwave input lines. Courtesy of IBM
CC BY-ND 2.0

CSC

What does a qubit look like?

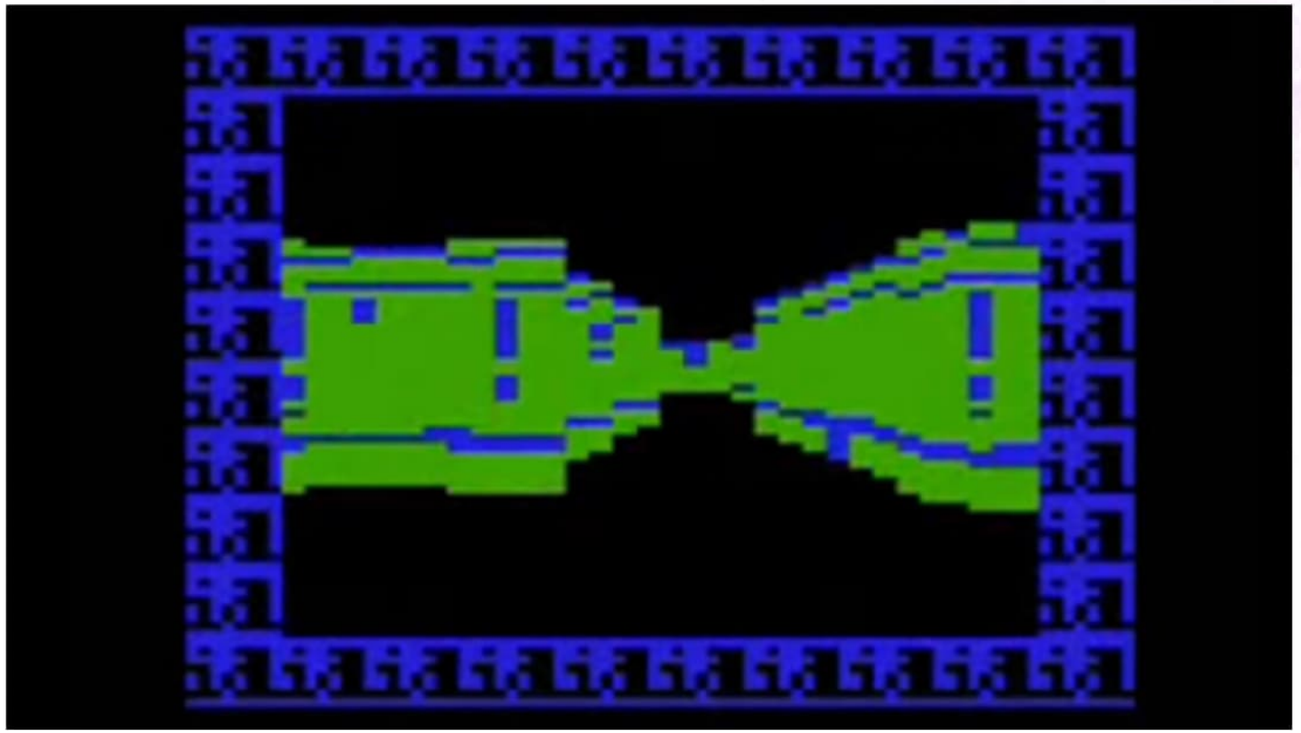
Stop-motion ion transport video showing a chosen sorting operation implemented on an 8-site 2D grid trap with the swap-or-stay primitive. The sort is implemented by discrete choices of swaps or stays between neighboring sites. The numbers shown (indicated by dashed circles) at the beginning and end of the video show the initial and final location of the ions after the sort, e.g. the ion that starts at the top left site ends at the bottom right site. The stop-motion video was collected by segmenting the primitive operation and pausing mid-operation such that Yb fluorescence could be detected with a CMOS camera exposure.

Quantinuum 2024
https://www.quantinuum.com/news/quantinuum-extends-its-significant-lead-in-quantum-computing-achieving-historic-milestones-for-hardware-fidelity-and-quantum-volume



What can one do with a quantum computer?

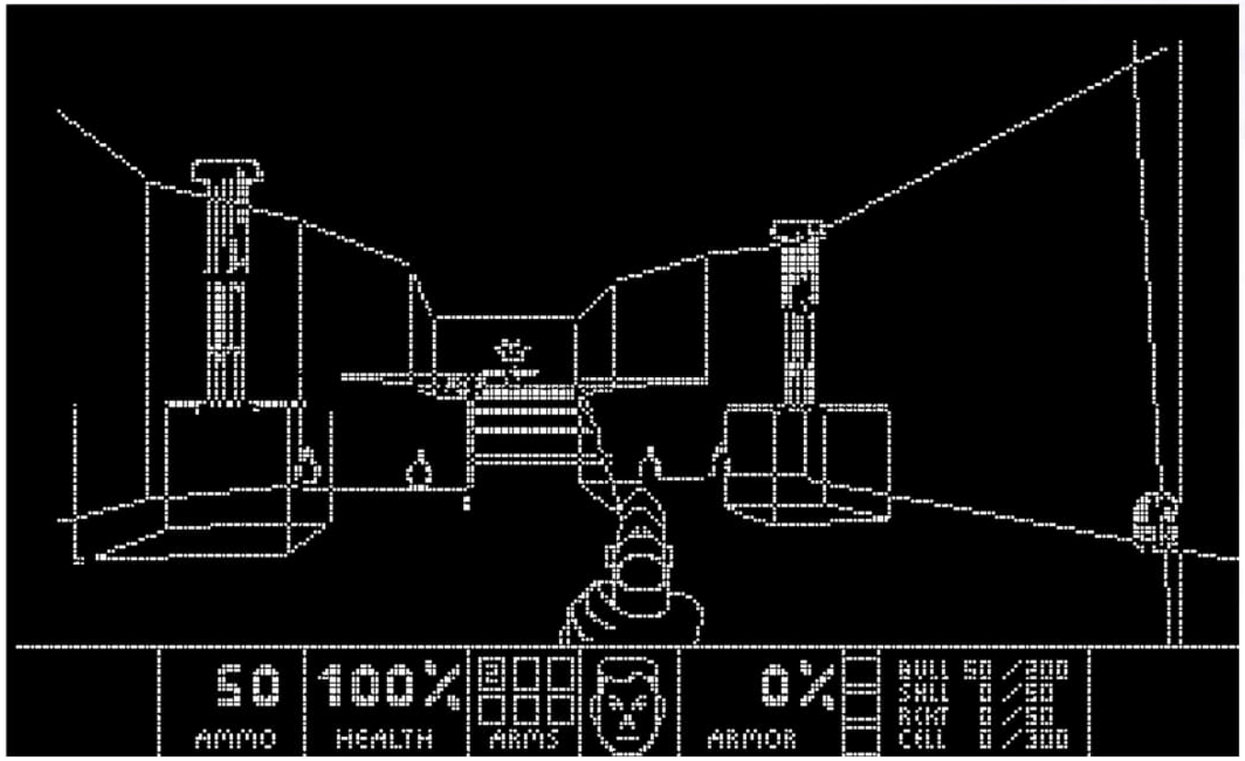
Will it run DOOM ?



No, this is VIC-20 from the 1980'ies

(answer on the spring edition of this course)

Will it run DOOM ?



Well, it could, but not efficiently!

https://github.com/Lumorti/Quandoom

The power of superposition

With qubits in superposition, all inputs can be processed in one go!

In a classical computer, the inputs need to be computed one by one

BUT! A quantum computer only gives *one* answer

In general suitable for problems where one is interested in the ”best” answer

Porvoo playing cards
by Niko Laurila

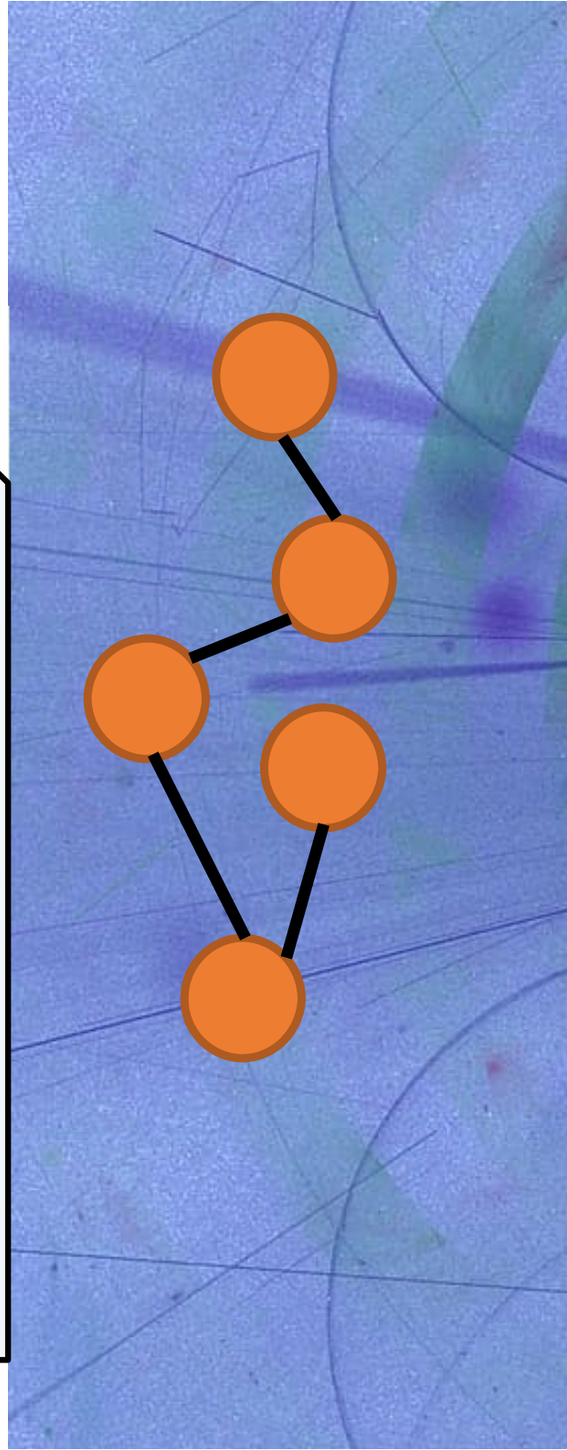
The travelling salesperson problem

- Find an optimal route connecting a set of *n* points

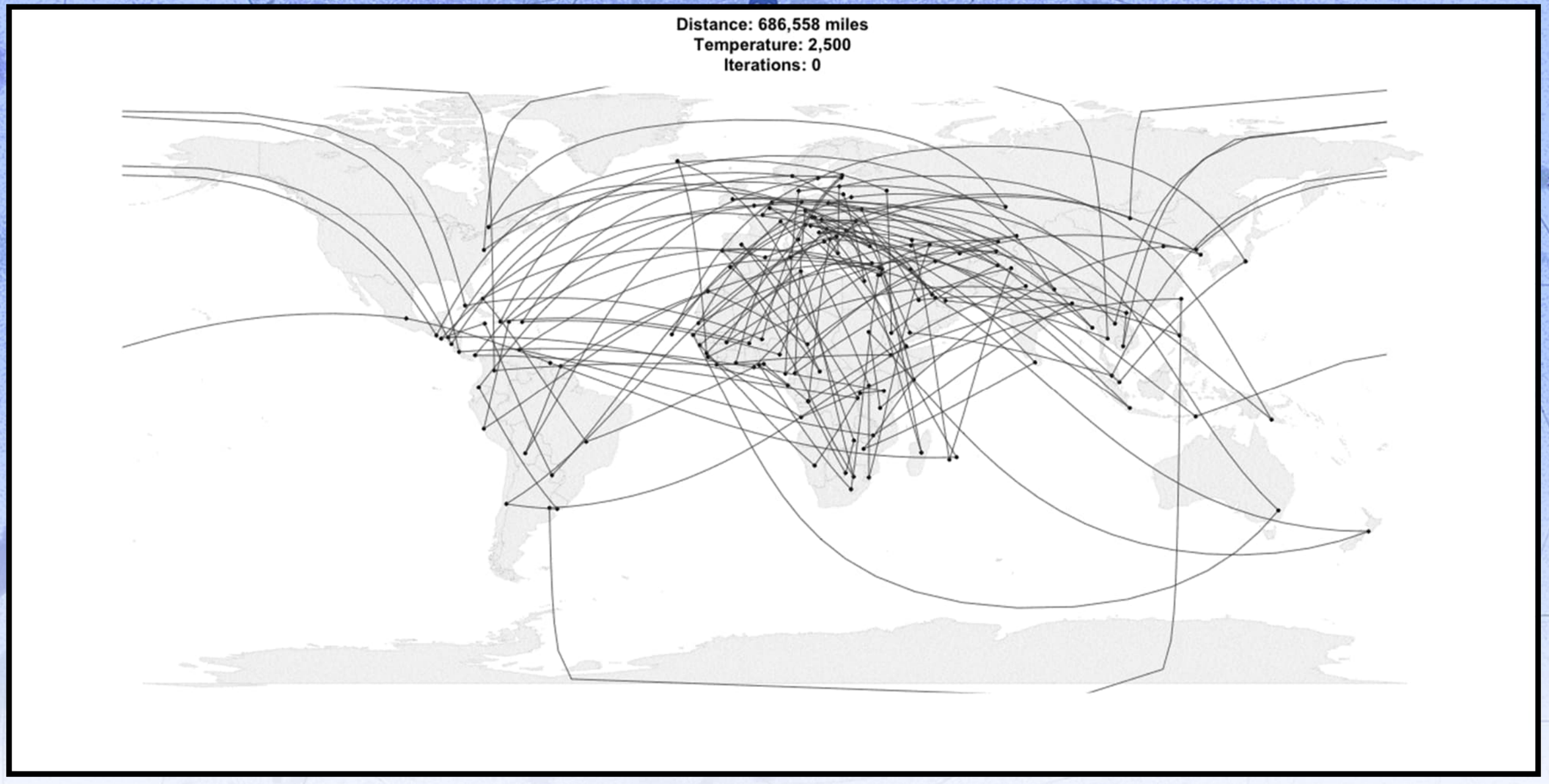
- Brute force: (*n*-1)! different options
  - **4** cities -> 3! = 1×2×3 = **6** options
  - **10** cities-> 9! = **362,880** options
  - **20** cities-> **121,645,100,408,832,000** options…

    121 quadrillion

- Only the (almost) optimal answer is of interest!
- Of course, classical algorithms do not go through all routes either



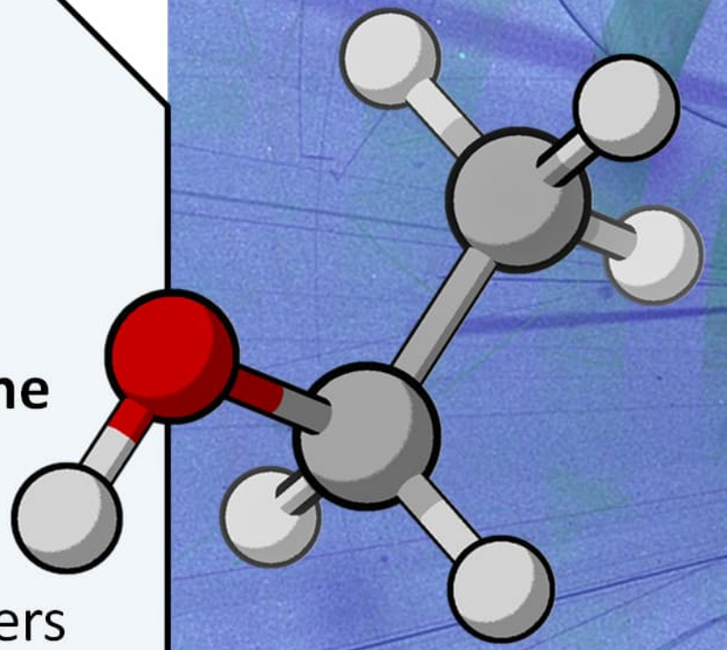
CSC



Simulation by Todd Schneider, reproduced with permission
https://toddwschneider.com/posts/traveling-salesman-with-simulated-annealing-r-and-shiny/

Not the travelling salesperson problem

- There are **several optimisation problems** that, unlike the travelling salesperson problem, are **very difficult for classical computers**
- **The electronic structure of molecules and materials is one of them!**
- Also here, we need to keep in mind that classical computers are *often* very good at solving electronic structure with useful accuracy



CSC

Threats

Breaking “bad” cryptography

A quantum computer can factorise integers into their prime number constituents (Peter Shor 1994)

- Endangers presently used public-key cryptography
- Still some time before quantum computers are there!

When do secrets expire? Need to act now!

- New post-quantum cryptography standards by NIST proposed in 2024

=5901233864018024937004363556106
2742586661904093308985703901346 8
8215730412521263433401869628323 2
3759983735090855684385601340135 7
8989956154839413146385685585661 86
3304527591821571487873258704431 2
8751720834603630783866242514056 6
8772458012842549101450588357324 6
7910865156193407997484857589733 9
8321573111017791512651050884578 6
1523247307886222911623825721897 7
0212701404360919961795008142148 7
3049660880589373192523251984236 4
9513973137219109800511051143912 1
9733671146287552056089353839106 8
3921391869950825069550937343069 6
0947183297452340372019584675407 1
1110855641976739054364618321231 2
5175867864973058503850517489619 6
0527380619206057515171061273234 7
340416039

Star Wreck: In the Pirkinning

CSC

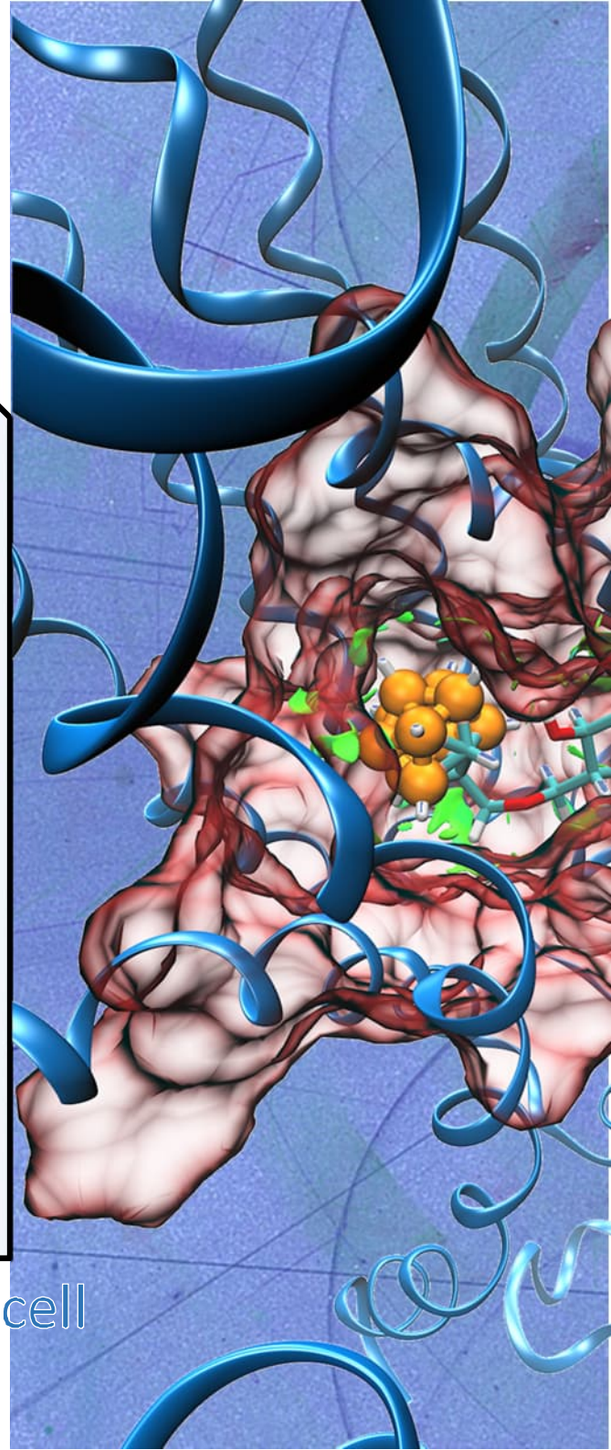
Types of problems for QC

- QC is good with problems with a *moderate* amount of both input and output variables
- The **relation** between input and output should be a **highly complex equation** that can be solved efficiently by some quantum algorithm
  - QC typically excels at problems with a **large potential solution *space***, but only a small set or even **a single solution**
  - Additionally, the input parameters need to be of the same order of magnitude as the number of qubits of the quantum computer

CSC

Typical HPC+QC applications in the near-term

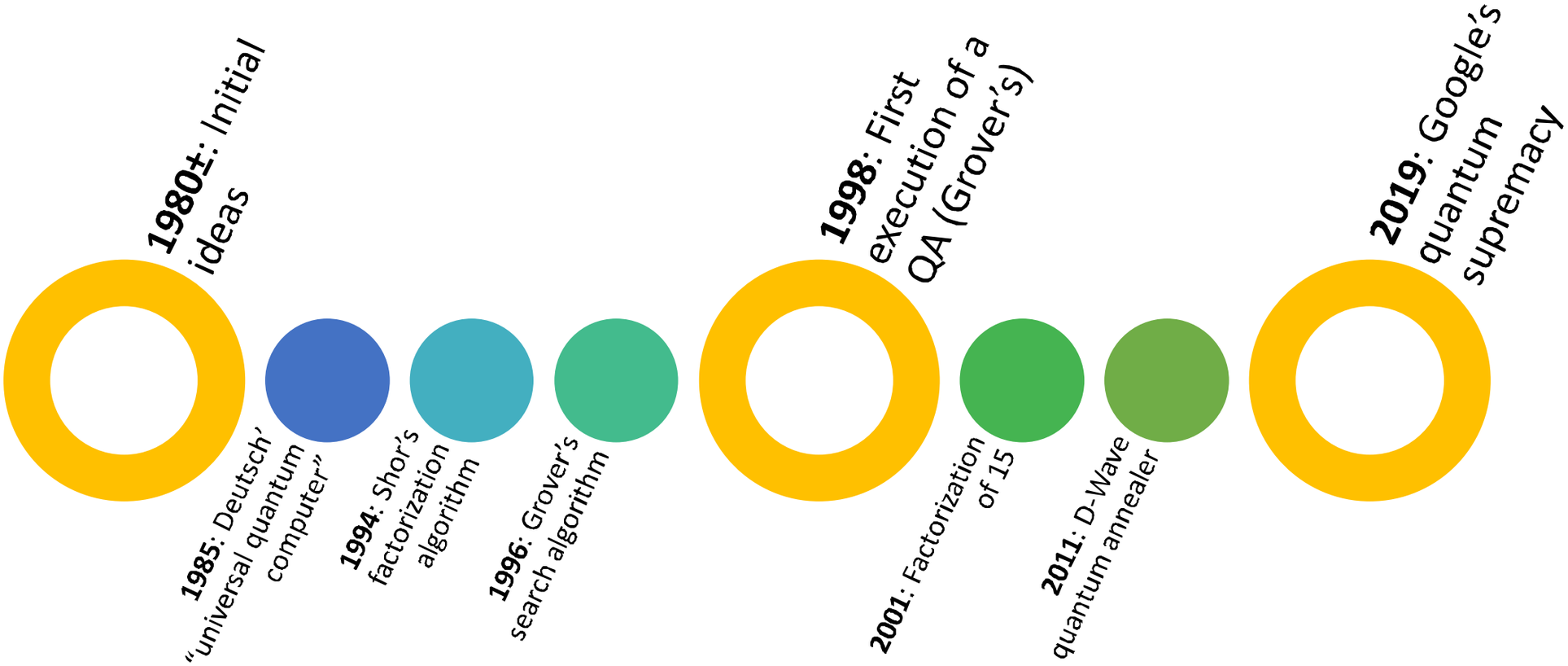
- **Electronic structure problems**
  - Do the best initial guess HPC is capable of, then refine further with quantum computing
- **Optimisation problems**
  - Compare solutions: if QC found a better one, great!
- **Machine learning**
  - Quantum computers can do ML *differently*
- **Fintech**
  - Even small improvement in risk analysis or portfolio optimisation = €$£¥



Drug molecule attacking a cancer cell

CSC

Timeline of quantum computing



NEWS AND VIEWS · 23 OCTOBER 2019

Quantum computing takes flight

A programmable quantum computer has been reported to outperform the most powerful conventional computers in a specific task — a milestone in computing comparable in importance to the Wright brothers' first flights.

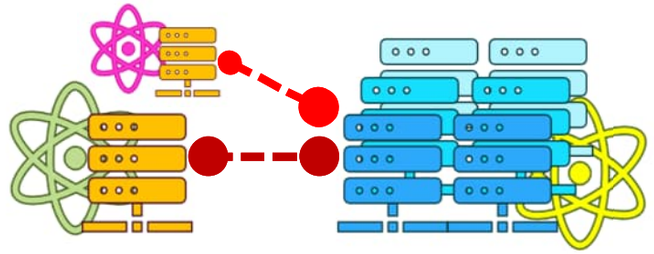
William D. Oliver ✉

Quantum computers are already faster!

- Quantum computers can already do calculations that supercomputers cannot!
- The problem is, that due to **noise** and **imprecise control**, they “crash” very quickly: **There is not time to do any *useful* calculation faster than a laptop!**
- **Car analogy:** Quantum computers are like the fastest racing cars ever built, but their batteries run empty after a few metres!
  - **During these few metres, they run faster than anything ever seen, however**
  - Now it’s “just” a matter of improving the batteries and installing better steering
  - More on this during tomorrow’s lecture!

CSC

Quantum computing now



Quantum computers can excel at *certain* computational tasks

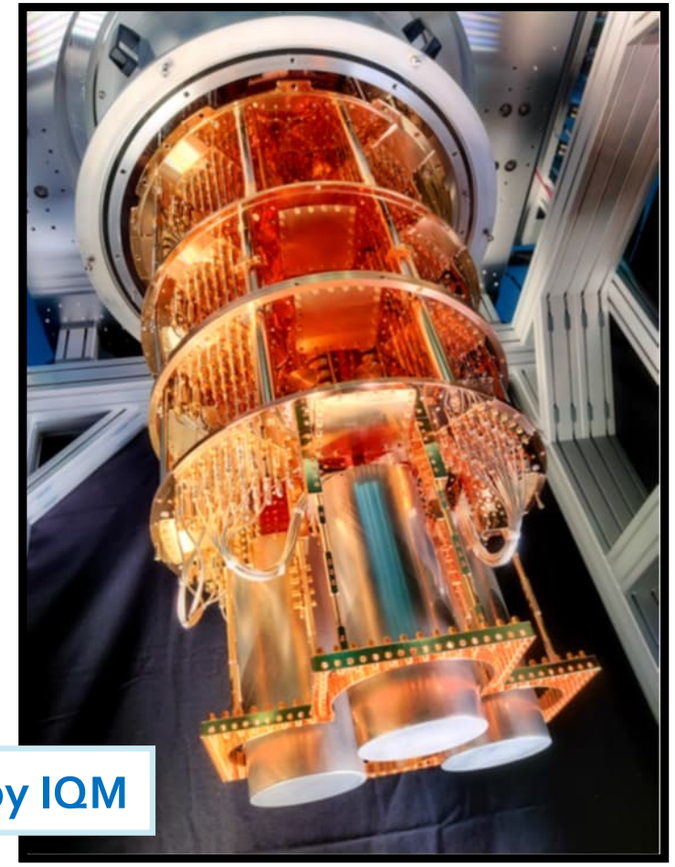
- Not for all task: **will always need co-operation with a classical computer**

QCs are *already* more powerful than supercomputers for *some* computational tasks

- Not for anything *useful*, though (except physics experiments!)
- Need more and higher quality qubits for **quantum advantage**

All major nations and large ICT companies involved in quantum computing

- IBM, Google, Intel, Ali Baba, Amazon, Microsoft, …
- **In Europe (and Finland) almost purely start-up driven**
- Largest quantum computer so far: **IBM Condor with 1,121 qubits** (retired)
- Both IBM and Google estimate **industrially relevant QC by 2030**
- McKinsey estimates **$620 - $1,270 billion potential value in industrial use cases by 2035**



Quantum computer by IQM

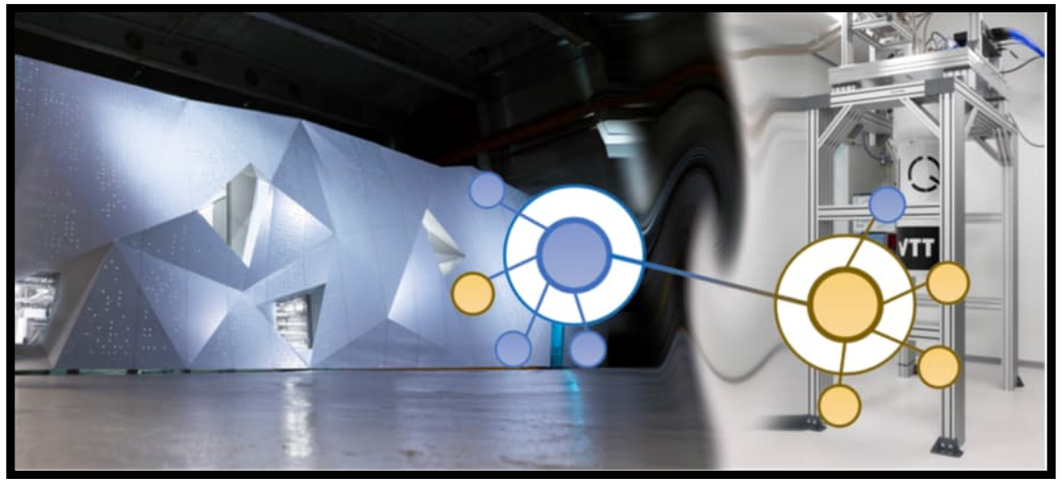
CSC

Connecting supercomputers with quantum computers

High-Performance Computing + Quantum Computing (HPC+QC)

Quantum acceleration

- Quantum computers are performing computations in a completely different manner
- **For some types of calculations**, QC can potentially speed up calculations significantly
- **For most types of calculations**, QC provides *no* advantage over classical computing



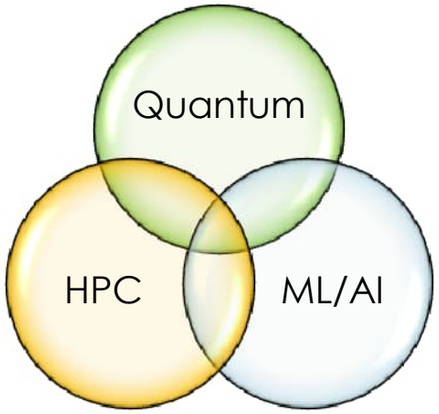
LUMI and Helmi

→ **Hybrid HPC+QC is the future!**

- Quantum computers will merge with supercomputers, both need each other

- **Finland is at the global forefront in developing HPC+QC**
- LUMI and Helmi are connected
  - Academic use (contact CSC)
  - Commercial use (contact VTT)

FiQCI



CSC

FiQCI – The Finnish Quantum-Computing Infrastructure

FiQCI

VTT Aalto University CSC

https://fiqci.fi

Goal

Provide a stable and competitive quantum computing environment and service

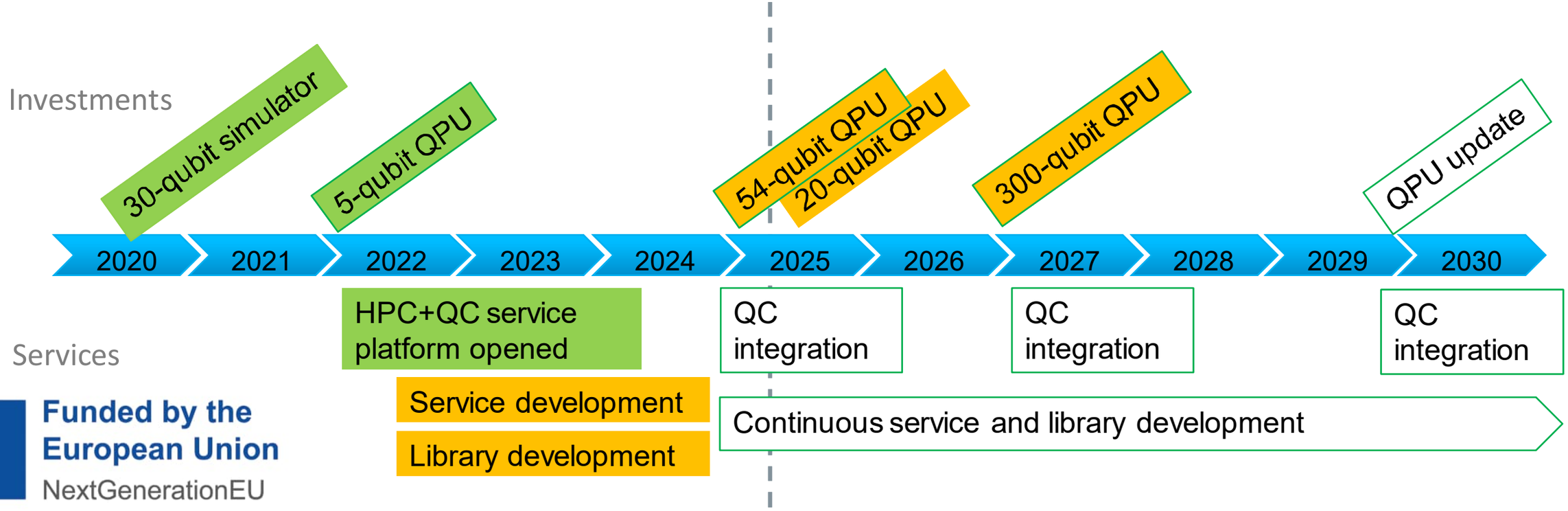
- Manage, maintain, and upgrade a hybrid high-performance computing + quantum computing (HPC+QC) infrastructure
- Support the introduction and development of quantum computing in Finland
- Support the teaching of quantum computing

Funded by the European Union
NextGenerationEU

Roadmap

On-going activities & existing HW

- The Eviden Qaptiva QLM 30 qubit classical simulator/emulator Kvasi (CSC)
- Building Finnish quantum computers with TEM funding (VTT together with IQM)
  - 5 qubits (now) → 54 qubits (2025) → up to 300 qubits (2027)
- 20-qubit quantum computer also for physical experiments (Aalto)
- HPC & service platform integration



Funded by the European Union
NextGenerationEU

Wider connection

FiQCI collaborates actively with other initiatives

- Synergies from several European QT projects
  - **NordIQuEst** Nordic-Estonian Quantum e-Infrastructure Quest (NeIC)
  - **QuTI** Quantum Technologies Industrial (Business Finland)
  - **OpenSuperQ+100** (EU Quantum Flagship)
  - **LUMI-Q** (EuroHPC)
  - **EuroQHPC-Integration** (EuroHPC)

International networking

- Global collaboration is crucial for driving quantum tech forward!
- FiQCI increases international visibility

> "***FiQCI is thus one of the most mature hybrid HPC-QC platforms in the world.***"
>
> Strategic Research and Industry Agenda, EU Quantum Flagship (2024)

Funded by the European Union
NextGenerationEU

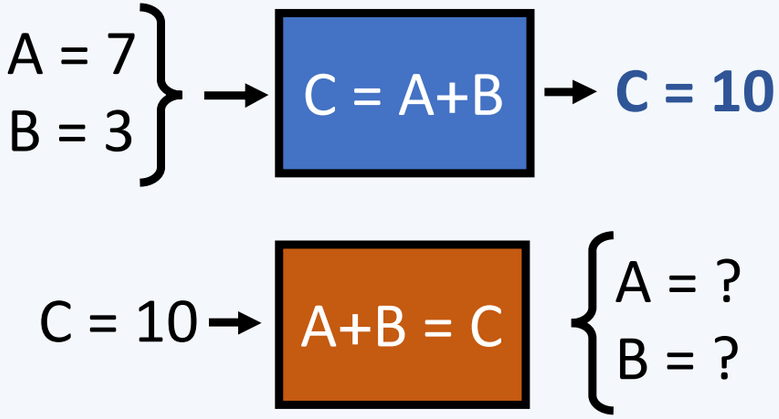
Programming quantum computers

Irreversible and reversible computing

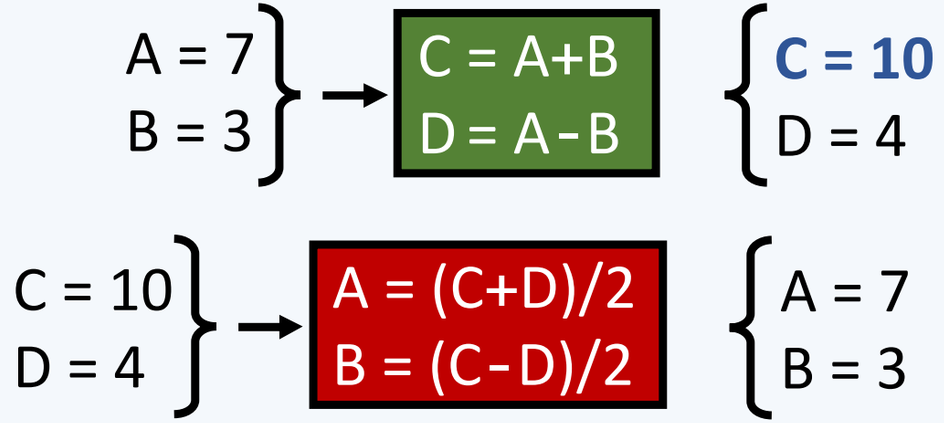
Classical computer programs are normally *irreversible:*
the input cannot be inferred from the output, *information is lost during computing*

- **Example**: Adding two numbers, A + B → C

Irreversible



Reversible



CSC

Irreversible and reversible operations

Classical computing uses *irreversible* operations/computations on the basic information units, the bits

- Example, **OR**:
- From result “1” we do not know the input

X	Y	X ∨ Y
0	0	0
0	1	1
1	0	1
1	1	1

Some operations are *reversible*, for example **NOT**: 0 -> 1
1 -> 0

CSC

Quantum operations are reversible

In quantum computing, **all** logical operations must be reversible!

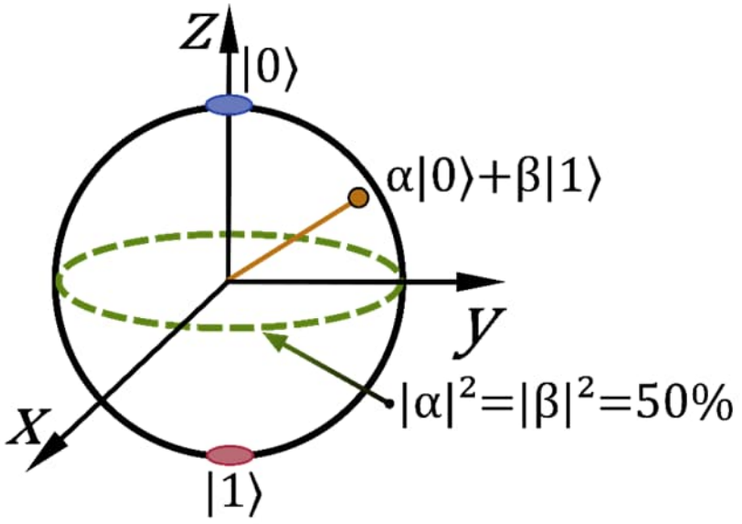
- Follows from the rules of quantum mechanics
  - All operators must be *unitary*, **UU* = I**
  - Keeps the probability $|\alpha|^2 + |\beta|^2$ **= 100%** for our qubit $\alpha|0\rangle + \beta|1\rangle$
- Using only reversible operations requires rethinking of algorithms
  - It *is* possible to do even classical computation with only reversible gates

CSC

Bit / qubit manipulations

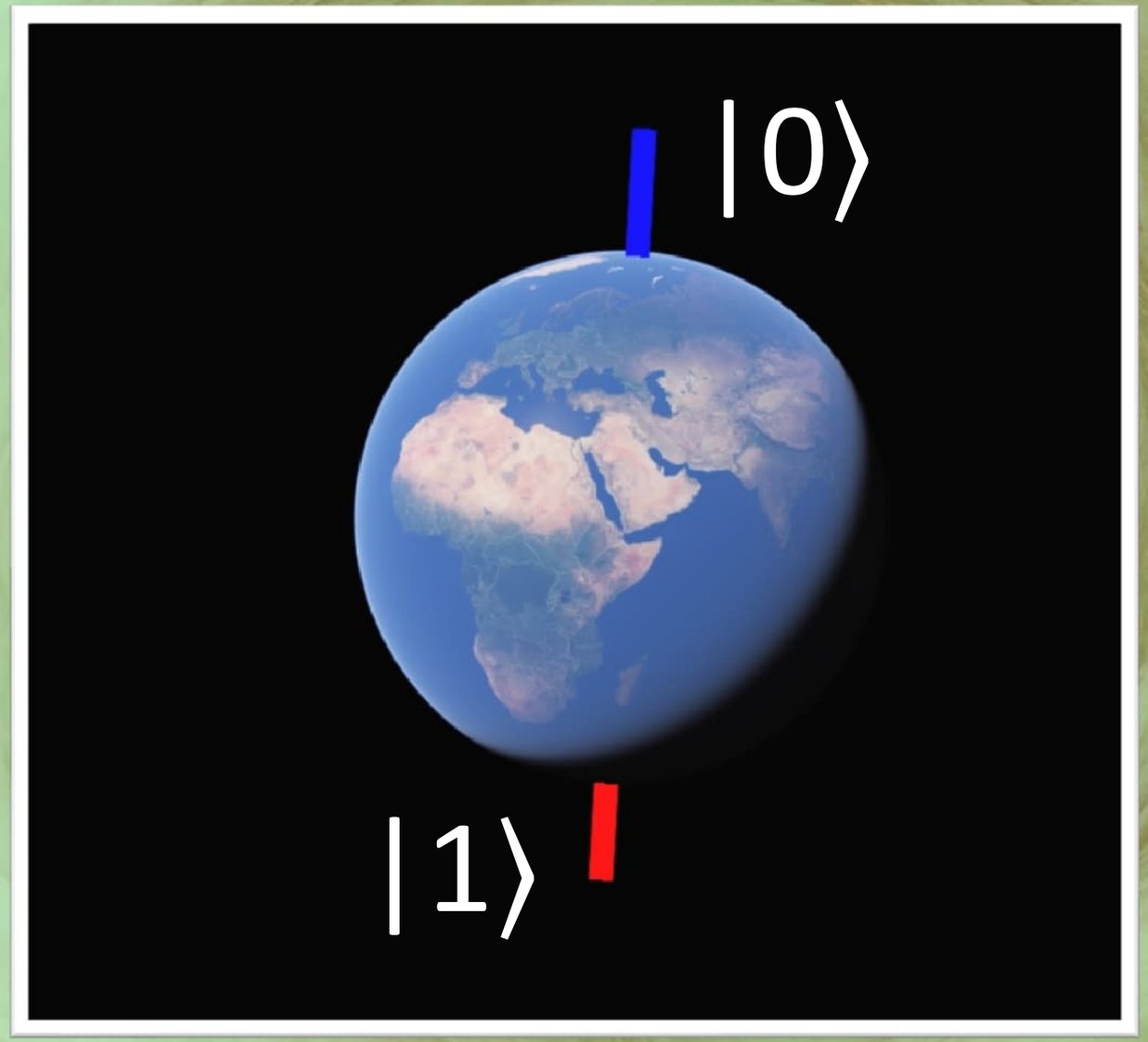
- A classical computer simply flips the states of the bits: 0 -> 1 -> 0 -> 1 …
- The quantum bits, qubits, are **much more versatile**: they can be something *in-between* $|0\rangle$ and $|1\rangle$:
  $\alpha|0\rangle+\beta|1\rangle$
- α and β are **complex numbers:** real and imaginary part
- Can be represented as **points on a sphere**
- In a quantum computer, the operations on the qubits can and *have to* take advantage of this flexibility

The Bloch Sphere



CSC

Bloch Sphere




Google Earth
Google FI, Data SIO, NOAA

Looking inside the box

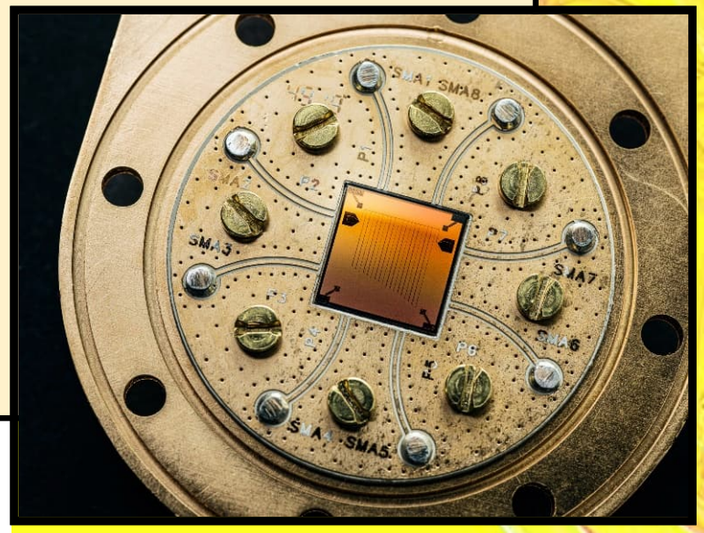
- Maintaining superposition of the qubits is central to quantum computing
  - **When superposition is lost, the calculation is over**
- Superposition lost if the state of the qubit is **measured**, that is, **observed**

Image: Adobe Stock

- Measuring gives **|0⟩** ***or*** **|1⟩** also with the qubit in superposition of **|0⟩** ***and*** **|1⟩**
  → After measuring |0⟩ *or* |1⟩ there is no way of going back, **the value sticks**
  → The values of the qubits can only be “printed out” once! At the end

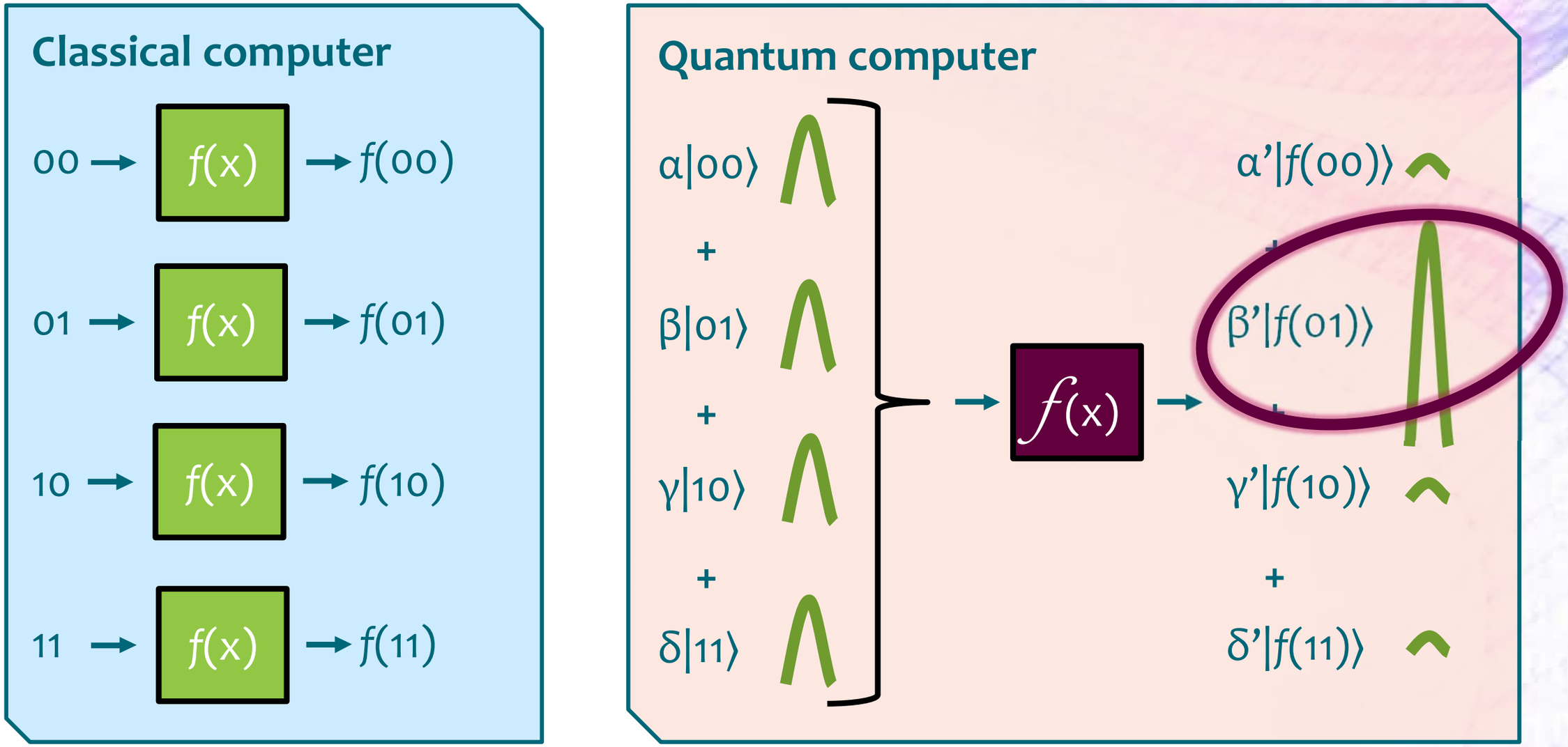
Measurement

- Even if **several inputs** can be processed at once, **only one answer** will emerge from the computer when you **measure** the result
  - $\alpha|0\rangle + \beta|1\rangle \rightarrow$ QPU $\rightarrow \boldsymbol{\alpha'}|f(0)\rangle + \boldsymbol{\beta'}|f(1)\rangle \rightarrow$  → ***f*(0)** *or* ***f*(1)**
  - The answer depends on the **amplitudes** **α'**, **β'**
  - $|\text{amplitude}|^2$ = **probability**; $|\alpha|^2 + |\beta|^2$ **= 100%**
- ***A quantum computer is not deterministic***
  - In general, *different answers for the same input*
  - This really is a **feature, not a bug!**

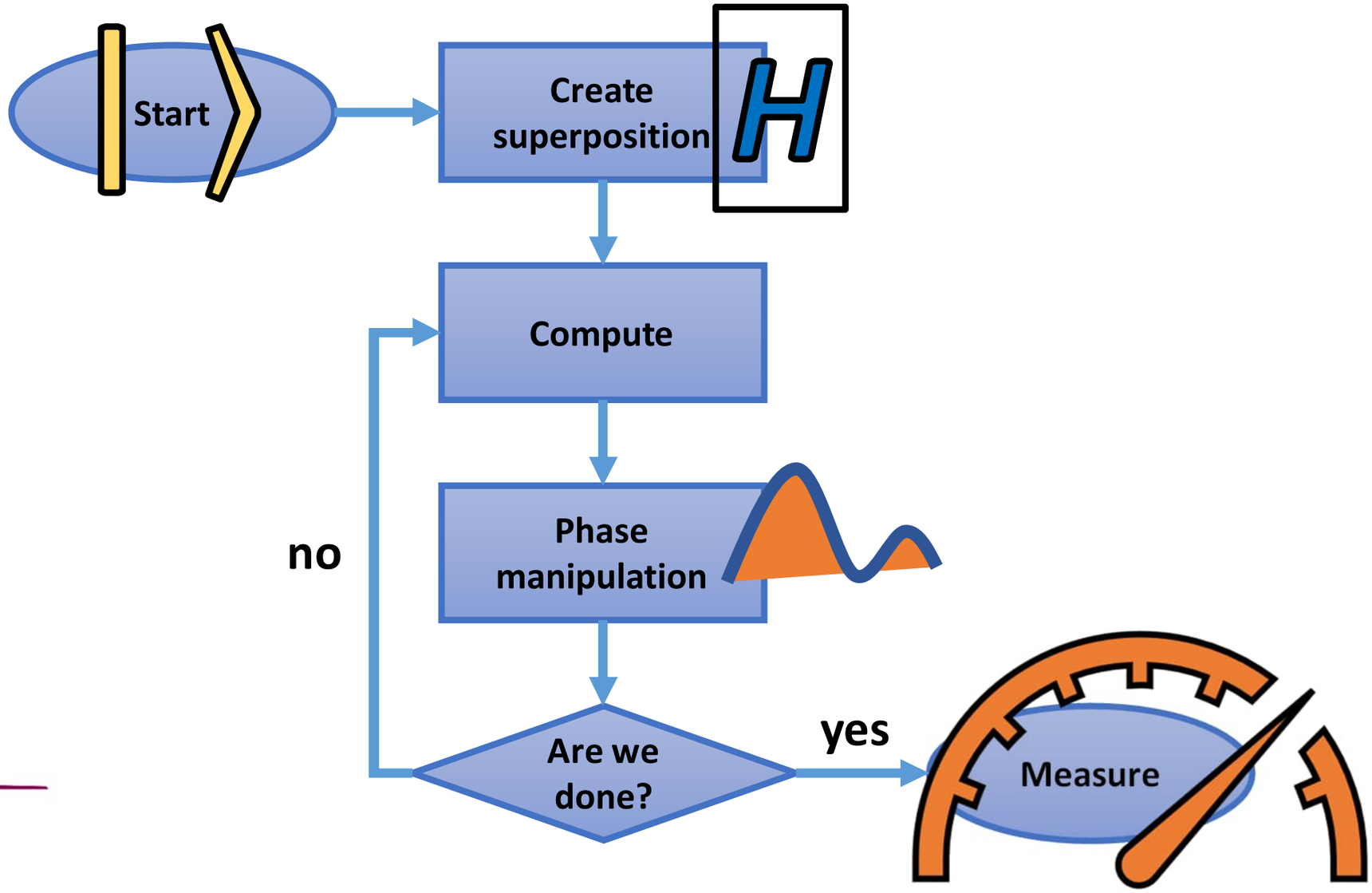


Quantum chip by IQM Quantum Computers

Parallelism from superposition



General quantum algorithm example



CSC

Disturbing noise

Qubits are **extremely delicate**: their states change due to outside influence, **noise** (like temperature): **superposition lost!**

- Affects the way algorithms need to be constructed in order to be efficient
- In **classical computing**, an inefficient program just runs for longer than necessary
- In **quantum computing**, **a longer run-time leads to larger errors as well!**

Kvanttisopuli: ”Kohina” saa kvanttitietokoneen tuottamaan hölynpölyvastauksia, Johansson & Åström, *Tekniikan Maailma* (online) 2020

CSC

Flavours of quantum computers

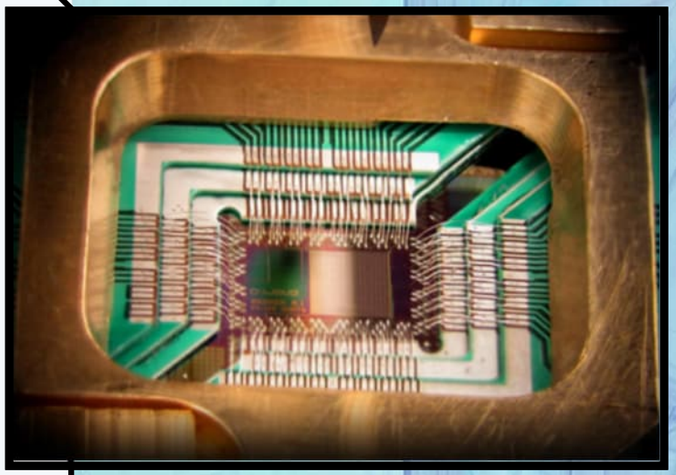
Quantum computers are programmed in different ways, depending on how the “quantum” is implemented

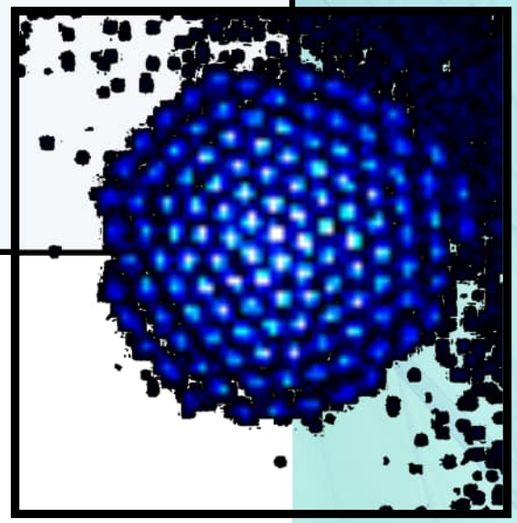
Arguably, the main classes are

- Quantum annealers
- Quantum simulators

- Continuous variable quantum computers
- **Gate-based quantum computers**

Photo: D-Wave
CC BY 3.0





Trapped beryllium ions
Image: NIST

CSC

Quantum operations on qubits

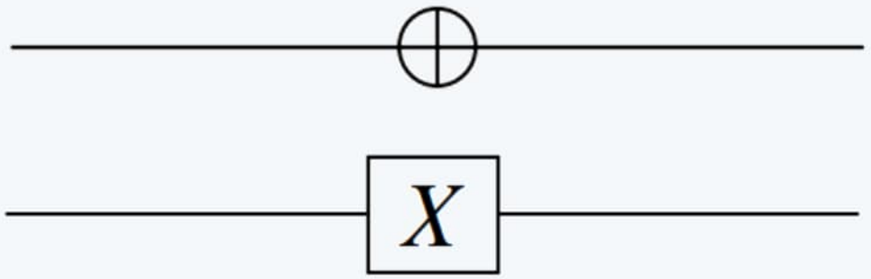
In computing using **quantum gates**, **operations** are represented using **linear algebra**

- States, qubits, are vectors $|0\rangle = \begin{pmatrix} 1 \\ 0 \end{pmatrix}, \quad |1\rangle = \begin{pmatrix} 0 \\ 1 \end{pmatrix}$
- Operators are matrices, for example NOT $X := \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$
- Applying a quantum operation on a state means performing the linear algebra
  - For example NOT$|0\rangle$, that is, X$|0\rangle$ $\begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \begin{pmatrix} 1 \\ 0 \end{pmatrix} = \begin{pmatrix} 0 + 0 \\ 1 + 0 \end{pmatrix} = \begin{pmatrix} 0 \\ 1 \end{pmatrix} = |1\rangle$

NOT $|0\rangle$

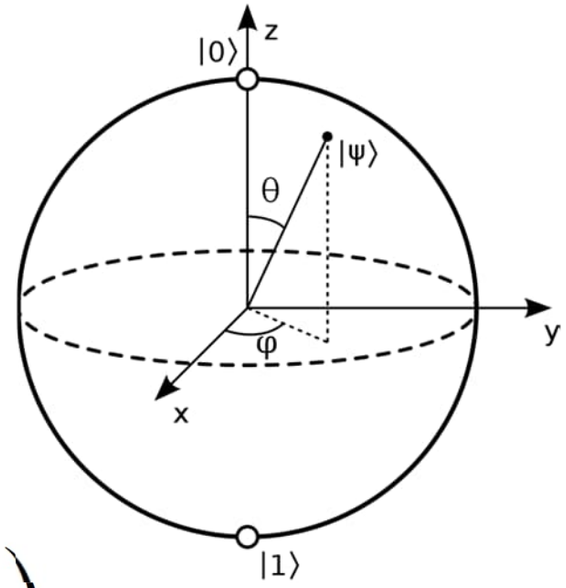
Quantum circuit diagrams

- Quantum algorithms are often shown as circuit diagrams
- One can use either symbols or names; for example for X

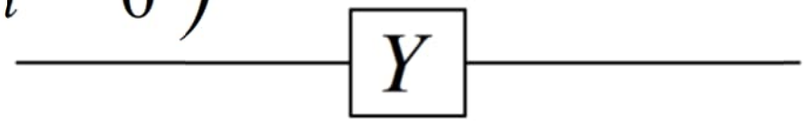


- A circuit diagram for NOT on $|0\rangle$ would then look like:

The Pauli operators

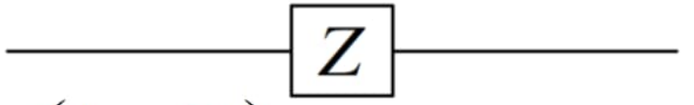


- Operators on single qubits are called **unary**
- **The Pauli operators X, Y, Z**
- X, “bit flip”, NOT $X := \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$
- Y, rotation of the state vector around the *y*-axis $Y = \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix}$
  - Example, Y|1⟩



$$\begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix} \begin{pmatrix} 0 \\ 1 \end{pmatrix} = \begin{pmatrix} 0 - i \\ 0 + 0 \end{pmatrix} = \begin{pmatrix} -i \\ 0 \end{pmatrix} = -i\,|0\rangle$$

Phase shift operators



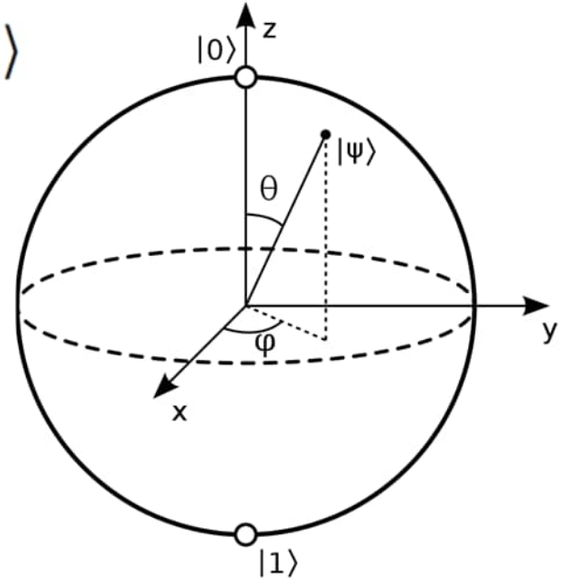
- The Z operator, rotation around the *z*-axis
  - Also called *phase flip*, as it flips the phase by π radians (180°)
  - Examples:

$$Z := \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}$$

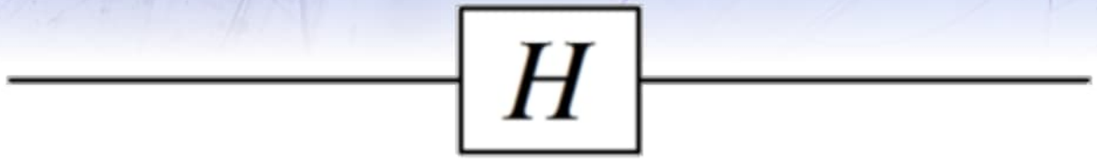
$$\begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix} \begin{pmatrix} 1 \\ 0 \end{pmatrix} = \begin{pmatrix} 1+0 \\ 0+0 \end{pmatrix} = \begin{pmatrix} 1 \\ 0 \end{pmatrix} = (-1)^0 |0\rangle = |0\rangle$$

$$\begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix} \begin{pmatrix} 0 \\ 1 \end{pmatrix} = \begin{pmatrix} 0+0 \\ 0-1 \end{pmatrix} = \begin{pmatrix} 0 \\ -1 \end{pmatrix} = (-1)^1 |1\rangle = -|1\rangle$$

- *General* phase flip operator Rϕ $R_\varphi := \begin{pmatrix} 1 & 0 \\ 0 & e^{i\varphi} \end{pmatrix}$
- Z is just Rϕ with ϕ= π ($e^{i\pi} = -1$)



The Hadamard gate



- **The arch-quantum gate, operator**
- Transforms a qubit from a specific state to a **superposition** of two states

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

- $H|0\rangle = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} \begin{pmatrix} 1 \\ 0 \end{pmatrix} = \frac{1}{\sqrt{2}} \begin{pmatrix} 1+0 \\ 1+0 \end{pmatrix} = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 \\ 1 \end{pmatrix} = \frac{|0\rangle + |1\rangle}{\sqrt{2}}$

- Note $\sqrt{2}$: The *square* of the amplitude is the probability of the state
  - The sum must be 1 (100%)
  - In general $\alpha|0\rangle + \beta|1\rangle$; $\mathbf{|\alpha|^2+|\beta|^2 = 1}$

Binary operators

- With two qubits, the 4 different states are (by convention)

$$|00\rangle = \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \end{pmatrix}, \quad |01\rangle = \begin{pmatrix} 0 \\ 1 \\ 0 \\ 0 \end{pmatrix}, \quad |10\rangle = \begin{pmatrix} 0 \\ 0 \\ 1 \\ 0 \end{pmatrix}, \quad |11\rangle = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \end{pmatrix}$$

- **Binary operators on 2 qubits are 4x4 matrices**

SWAP gate

- The SWAP gate swaps the state of two qubits

$$|0\rangle = \begin{pmatrix} 1 \\ 0 \end{pmatrix}, \quad |1\rangle = \begin{pmatrix} 0 \\ 1 \end{pmatrix} \qquad |00\rangle = \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \end{pmatrix}, \quad |01\rangle = \begin{pmatrix} 0 \\ 1 \\ 0 \\ 0 \end{pmatrix}, \quad |10\rangle = \begin{pmatrix} 0 \\ 0 \\ 1 \\ 0 \end{pmatrix}, \quad |11\rangle = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \end{pmatrix}$$

$$SWAP := \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

Example: SWAP|01> = $\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 0 \\ 1 \\ 0 \\ 0 \end{pmatrix} = \begin{pmatrix} 0+0+0+0 \\ 0+0+0+0 \\ 0+1+0+0 \\ 0+0+0+0 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 1 \\ 0 \end{pmatrix} = |10\rangle$

Controlled NOT, CNOT, CX

A **controlled** two-qubit gate uses one qubit to *control* the operation and the other as *target*

- If control = $|0\rangle$ then do nothing

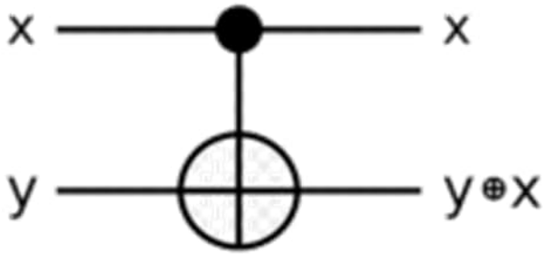
The controlled NOT, **CNOT** performs NOT on *target* if *control* = $|1\rangle$

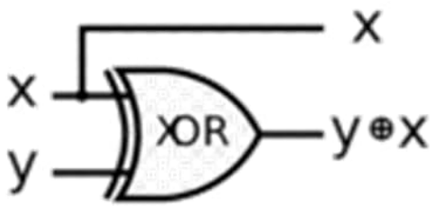
$$CNOT := \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix}$$

$$\text{CNOT}|10\rangle = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} 0 \\ 0 \\ 1 \\ 0 \end{pmatrix} = \begin{pmatrix} 0+0+0+0 \\ 0+0+0+0 \\ 0+0+0+0 \\ 0+0+1+0 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \end{pmatrix} = |11\rangle$$

- Analogous to classical eXclusive OR, **XOR**

CNOT vs XOR





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

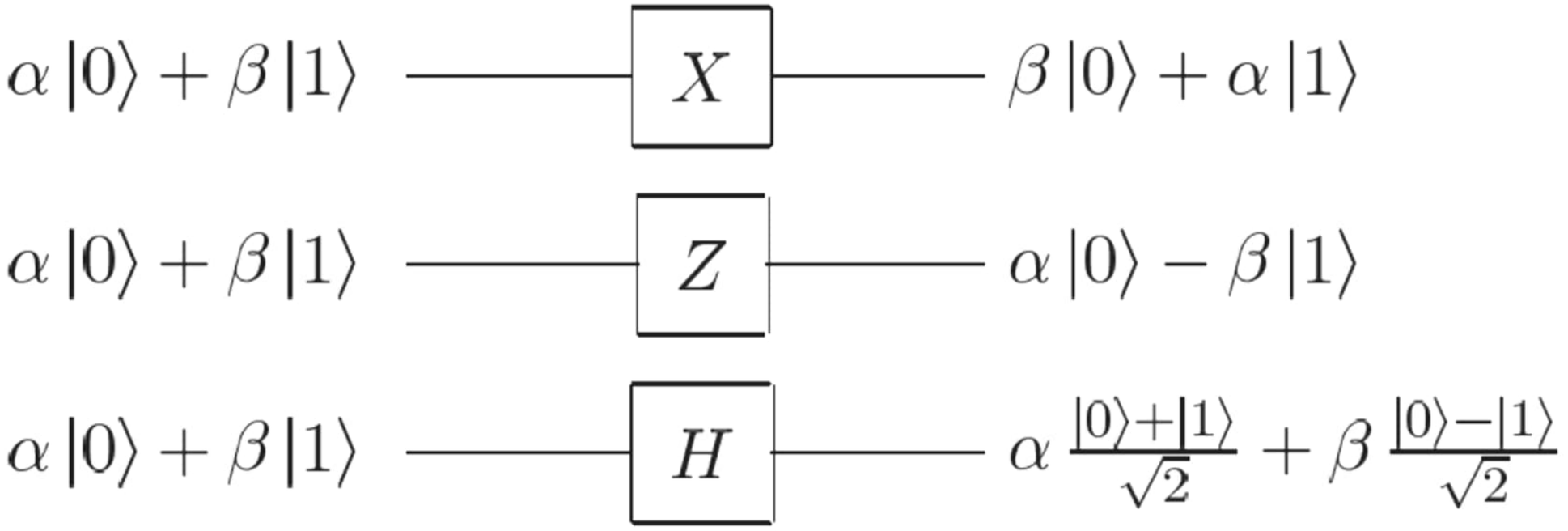
input		output	
x	y	x	y+x
0	0	0	0
0	1	0	1
1	0	1	1
1	1	1	0

All looks quite classical still.

But what if the input states are in superposition?

Operating on superposition states

When we have something else than the **basis states** $|0\rangle$ and $|1\rangle$, the quantum gates operate in a more complex manner



Entanglement

- Special kind of superposition

MAY 15, 1935 PHYSICAL REVIEW VOLUME 47

Can Quantum-Mechanical Description of Physical Reality Be Considered Complete?

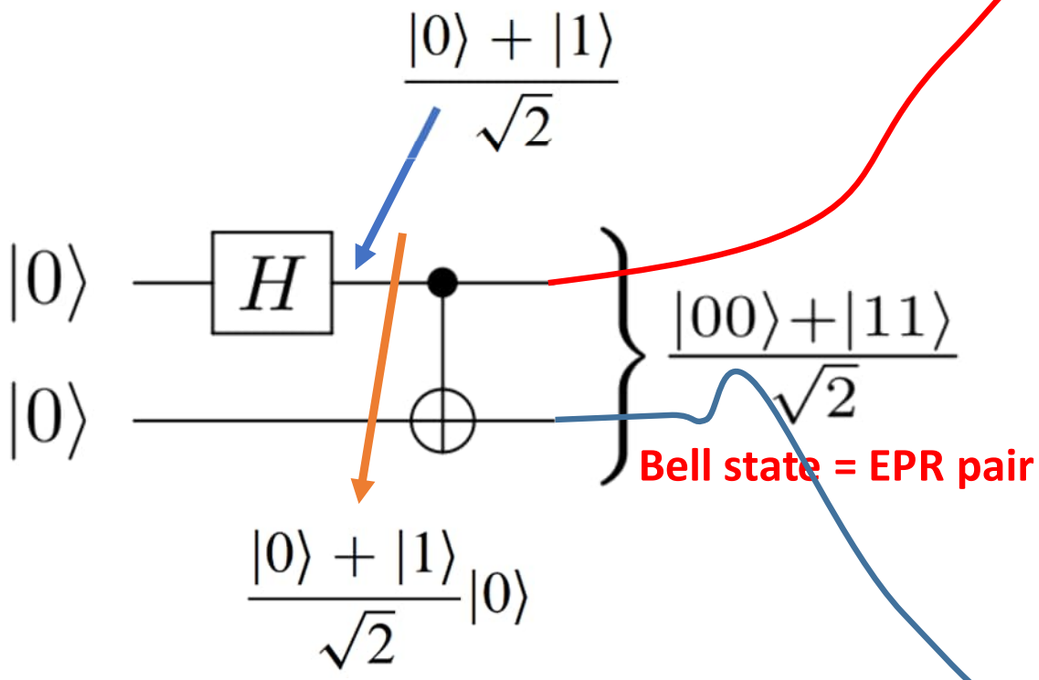
A. EINSTEIN, B. PODOLSKY AND N. ROSEN, *Institute for Advanced Study, Princeton, New Jersey*

(Received March 25, 1935)

- Two quantum states (say qubits) that are entangled experience the famous "*spooky action at a distance*" (orig. "*spukhafte Fernwirkung*")
- In entangled superposition, the states of qubit A and B are unknown before measured
- After measuring *one* of the qubits, the state of the other is immediately set
- The superpositions of the two qubits are inseparable

H+CNOT = Entanglement

What if the *control* qubit for CNOT is in a superposition of $|0\rangle$ and $|1\rangle$?



- Equal superposition of **two out of four** possible two-qubit states
  - If qubit A = $|0\rangle$ also qubit B = $|0\rangle$
  - If qubit A = $|1\rangle$ also qubit B = $|1\rangle$
- **Before measuring**, we do not know either state, however!
- We *do* know they are equal -> **entanglement**

$$CNOT\,\frac{1}{\sqrt{2}}(|00\rangle + |10\rangle) = \frac{1}{\sqrt{2}}(CNOT|00\rangle + CNOT|10\rangle) = \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle$$

THE INSTITUTE FOR ADVANCED STUDY

PRINCETON, NEW JERSEY

July 5th,1952

SCHOOL OF MATHEMATICS

Mr.Daniel M.Lipkin
4925 Rubicam Str.
Philadelphia 44,Pa.

Dear Mr.Lipkin:

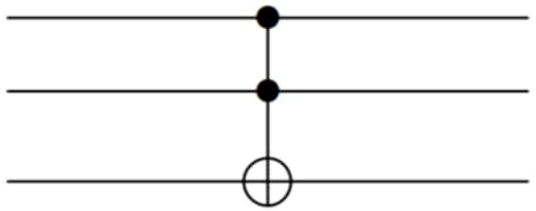
I too have many reasons to believe that the present quantum theory,inspite of its many successes, is far from the truth. This theory reminds me a little of the system of delusion of an exceedingly intelligent paranoiac concocted of incoherent elements of thought. As you also seem to believe I believe it impossible to get a real insight without satisfying from the start the principle of general relativity. I feel,however, by no means sure that my own approach is the right one.

Sincerely yours,

A. Einstein

Albert Einstein.

Toffoli, a ternary operator



- Ternary 3-qubit operators are represented by 8x8 matrices
- **Toffoli** = CCNOT, controlled-controlled-NOT
- If *both* control qubits are in state $|1\rangle$ then flip the third qubit

- Ex: CCNOT$|110\rangle$ =

$$\begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \\ 0 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \end{pmatrix} = |111\rangle$$

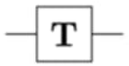
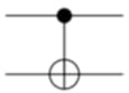
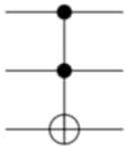
Quantum logic gates

A set of reversible logic operations to use for programming a quantum computer

- There are of course (infinitely) many more; however:

We do not need all of the gates, and most are not actually implemented on real quantum computers anyway!

- Real quantum computers have different gates, which might be conceptually more complex, but easier to implement in the real world
  - XX, CPHASE, CU1, U3, …

Operator	Gate(s)	Matrix
Pauli-X (X)		$\begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$
Pauli-Y (Y)		$\begin{bmatrix} 0 & -i \\ i & 0 \end{bmatrix}$
Pauli-Z (Z)		$\begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}$
Hadamard (H)		$\frac{1}{\sqrt{2}}\begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$
Phase (S, P)		$\begin{bmatrix} 1 & 0 \\ 0 & i \end{bmatrix}$
$\pi/8$ (T)		$\begin{bmatrix} 1 & 0 \\ 0 & e^{i\pi/4} \end{bmatrix}$
Controlled Not (CNOT, CX)		$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix}$
Controlled Z (CZ)		$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & -1 \end{bmatrix}$
SWAP		$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$
Toffoli (CCNOT, CCX, TOFF)		$\begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix}$

Rxtreme / Wikipedia

Universal gate sets: NAND

- A **universal** set of gates can describe all other logic gates
- In **classical, irreversible** computing, the **NAND** gate, NOT-AND is universal (functionally complete)
  - All logical Boolean gates can be implemented by combining just NAND gates
  - Can require an exponential number of gates, though, so not actually efficient
- NAND is irreversible (multiple inputs possible from one output)
  - Thus, does not work for quantum computing

INPUT		OUTPUT
A	B	A NAND B
0	0	1
0	1	1
1	0	1
1	1	0

Universal gate sets: Toffoli

- Toffoli, CCNOT, is the **reversible** equivalent of NAND
- {Toffoli} is universal for classical computing
  - One can construct NAND from Toffolis
- Alone, Toffoli is *not* sufficient for **quantum** computing: There is no way to create a superposition!
- **Universal quantum gate sets** include:

  {Toffoli, H}

  {CNOT, T, H}

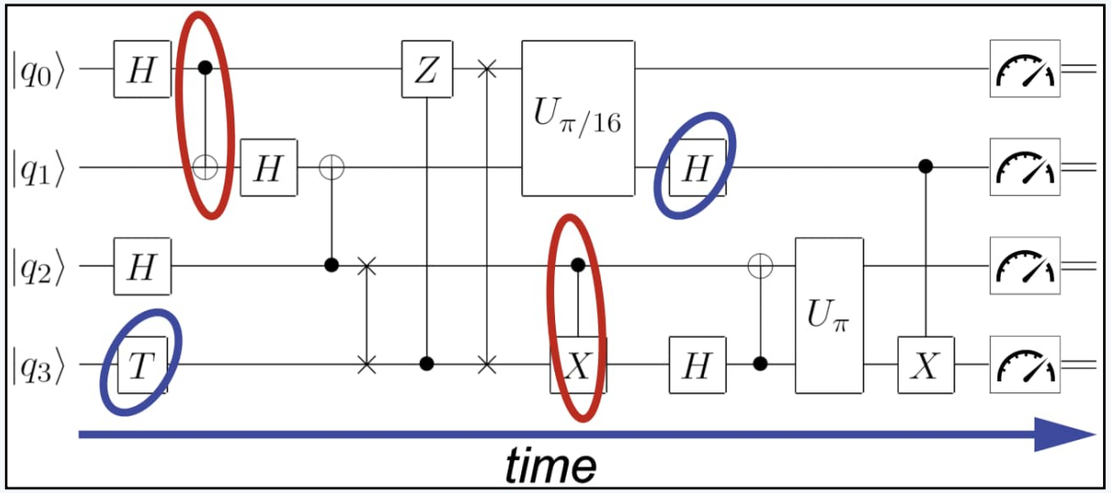
INPUT			OUTPUT		
0	0	0	0	0	0
0	0	1	0	0	1
0	1	0	0	1	0
0	1	1	0	1	1
1	0	0	1	0	0
1	0	1	1	0	1
1	1	0	1	1	1
1	1	1	1	1	0

Combining operations into algorithms

- The quantum gates (H, CNOT, SWAP, …) are quite different from the commonly used classical operations (AND, OR, XOR, …)
- Already this shows that programming quantum computers will be different
- In addition, to actually gain any advantage from a quantum computer, **we have to exploit superposition and entanglement!**
- This means that quantum algorithms have to be developed
- Problems need to be formulated so that they utilise quantum phenomena *explicitly*
- Many problems *cannot* be formulated so that there would be a quantum advantage
- Those that can, can see an **enormous** speed-up!

Combining operations into algorithms

- For now, quantum programming means writing rather **low-level code**
- **Quantum algorithms are built up from basic gate operations**



- **In the future/in the making:** libraries of usable quantum algorithms, more advanced compilers, FortranQ, *etc.*
- **Need to be developed so that they are ready when the big QPUs come!**

CSC

Ada King, Countess of Lovelace

First computer algorithms also written before the hardware

- Lovelace's Bernoulli-algorithm **published in 1843**
- Programmed for Charles Babbage's *Analytical Engine*

Watercolour: Alfred Edward Chalon

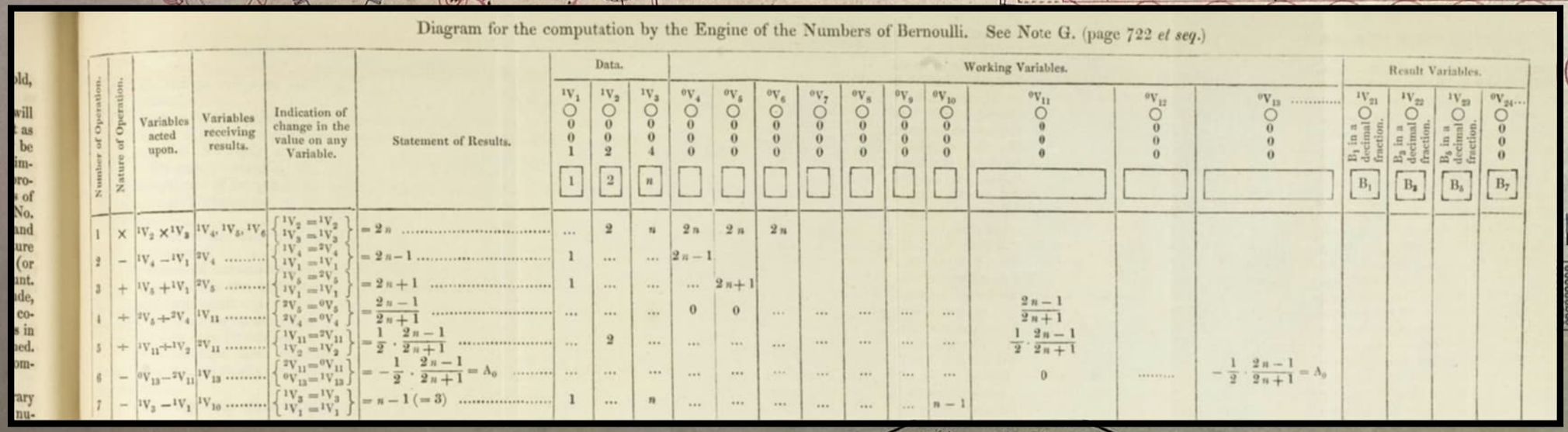


Diagram for the computation by the Engine of the Numbers of Bernoulli. See Note G. (page 722 *et seq.*)

Number of Operation.	Nature of Operation.	Variables acted upon.	Variables receiving results.	Indication of change in the value on any Variable.	Statement of Results.	Data.			Working Variables.										Result Variables.			
						$^1V_1$ 0 0 0 1 [1]	$^1V_2$ 0 0 0 2 [2]	$^1V_3$ 0 0 0 4 [n]	$^0V_4$ 0 0 0 0 [ ]	$^0V_5$ 0 0 0 0 [ ]	$^0V_6$ 0 0 0 0 [ ]	$^0V_7$ 0 0 0 0 [ ]	$^0V_8$ 0 0 0 0 [ ]	$^0V_9$ 0 0 0 0 [ ]	$^0V_{10}$ 0 0 0 0 [ ]	$^0V_{11}$ 0 0 0 0 [ ]	$^0V_{12}$ 0 0 0 0 [ ]	$^0V_{13}$ ... 0 0 0 0 [ ]	$^1V_{21}$ B_1 in a decimal fraction. [B_1]	$^1V_{22}$ B_3 in a decimal fraction. [B_3]	$^1V_{23}$ B_5 in a decimal fraction. [B_5]	$^0V_{24}$... 0 0 0 0 [B_7]
1	×	$^1V_2 \times {}^1V_3$	$^1V_4, {}^1V_5, {}^1V_6$	$\left\{\begin{matrix}{}^1V_2 = {}^1V_2 \\ {}^1V_3 = {}^1V_3\end{matrix}\right\}$	$= 2n$	...	2	n	$2n$	$2n$	$2n$												
2	−	$^1V_4 - {}^1V_1$	$^2V_4$	$\left\{\begin{matrix}{}^1V_4 = {}^2V_4 \\ {}^1V_1 = {}^1V_1\end{matrix}\right\}$	$= 2n-1$	1	...	...	$2n-1$														
3	+	$^1V_5 + {}^1V_1$	$^2V_5$	$\left\{\begin{matrix}{}^1V_5 = {}^2V_5 \\ {}^1V_1 = {}^1V_1\end{matrix}\right\}$	$= 2n+1$	1	...	...	...	$2n+1$													
4	+	$^2V_5 + {}^2V_4$	$^1V_{11}$	$\left\{\begin{matrix}{}^2V_5 = {}^0V_5 \\ {}^2V_4 = {}^0V_4\end{matrix}\right\}$	$= \frac{2n-1}{2n+1}$	...	...	...	0	0	...	...	...	...	...	$\frac{2n-1}{2n+1}$							
5	+	$^1V_{11} + {}^1V_2$	$^2V_{11}$	$\left\{\begin{matrix}{}^1V_{11} = {}^2V_{11} \\ {}^1V_2 = {}^1V_2\end{matrix}\right\}$	$= \frac{1}{2} \cdot \frac{2n-1}{2n+1}$	...	2	...	...	...	...	...	...	...	...	$\frac{1}{2} \cdot \frac{2n-1}{2n+1}$							
6	−	$^0V_{13} - {}^2V_{11}$	$^1V_{13}$	$\left\{\begin{matrix}{}^2V_{11} = {}^0V_{11} \\ {}^0V_{13} = {}^1V_{13}\end{matrix}\right\}$	$= -\frac{1}{2} \cdot \frac{2n-1}{2n+1} = A_0$	...	...	...	...	...	...	...	...	...	...	0	.....	$-\frac{1}{2} \cdot \frac{2n-1}{2n+1} = A_0$					
7	−	$^1V_3 - {}^1V_1$	$^1V_{10}$	$\left\{\begin{matrix}{}^1V_3 = {}^1V_3 \\ {}^1V_1 = {}^1V_1\end{matrix}\right\}$	$= n-1\,(=3)$	1	...	n	...	...	...	...	...	...	$n-1$								

The General Plan of Mr. Babbages Great Calculating Engine.

Background: ArnoldReinhold / Wikipedia

Levels of quantum programming

3. Ready-made libraries of common mathematical subroutines

- To be called from within a standard programming language
- “QBLAS”, Qiskit Aqua, Q# Libraries, …
- New variable type `qubit` for existing languages?

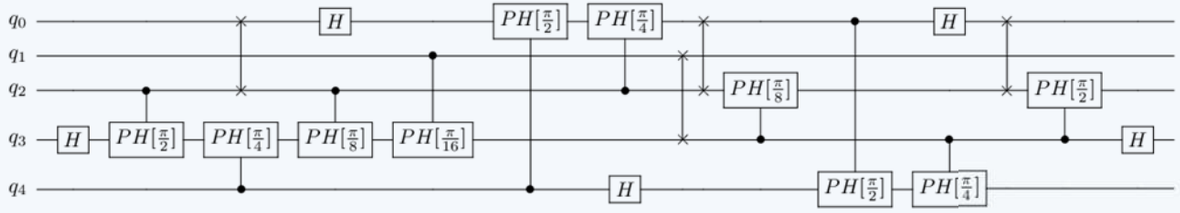
```
logical function qrand()
  implicit none
  qubit :: q0
  q0 = (1.0,0.0,0.0,0.0)
  q0 = q_H(q0)
  qrand = q_measure(q0)
end function qrand
```

2. Higher-level languages/SDKs for quantum programming

- Devise new quantum algorithms and apply existing algorithms to new problems
- Qiskit, PennyLane, Q#, Cirq, t|ket⟩, Silq, pyAQASM, Strawberry Fields, …

1. Circuit-level assembly

- QIR, OpenQASM, AQASM, …



0. Hardware-level coding

- Firmware, microcode, hardware-specific compilers, …
- Not for general programming

CSC

Image: Adobe Stock

Quantum programming challenges

1. **Problem identification**
   - *What exactly can/should QPUs compute?*
2. **Extracting advantage**
   - *How to exploit superposition, entanglement, wave-function phases?*
3. **Working around limitations**
   - ***Noise**, qubit connectivity, decoherence, …*
4. **Hybrid HPC+QC computing**
   - *Getting the best of both worlds*
5. **Very different programming paradigm**
   - *All basic operations have to be **reversible!***
   - ***Only one measurement**, usually at the end*
   - *Probabilistic, not deterministic*
   - ***Uncomputing** necessary*

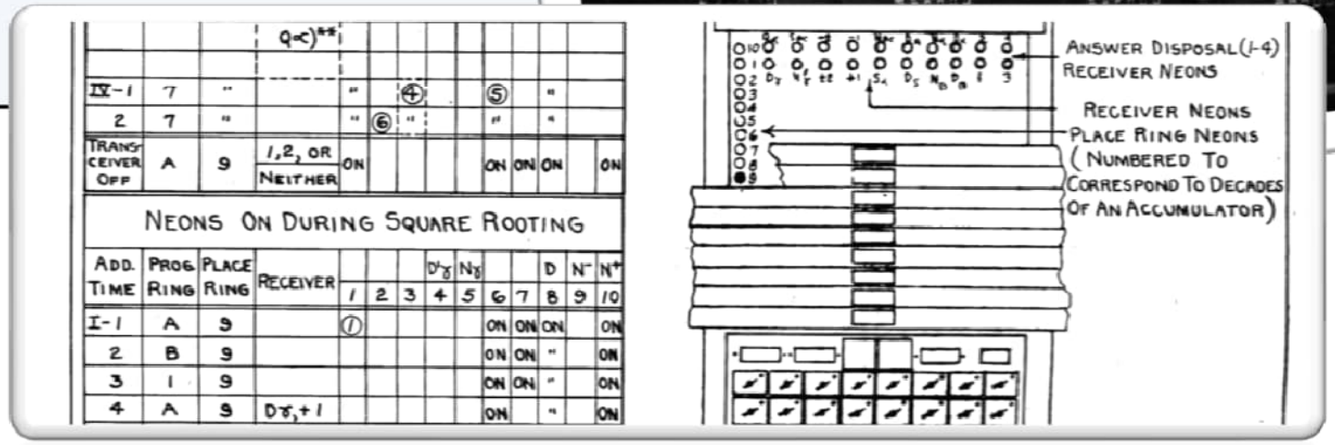
Programming challenges 75 years ago

- **ENIAC (Electronic Numerical Integrator and Computer) 1946:**

7. Do not open d-c fuse cabinet with the d-c power turned on. This not only exposes a person to voltage differences of around 1500 volts but the person may be burned by flying pieces of molten fuse wire in case a fuse should blow.

We have perhaps become a bit spoilt!

Picture: U.S. Army



			(Q-C)**												
IX-1	7	"		"		④			⑤		"				
2	7	"		"	⑥	"			"		"				
TRANSCEIVER OFF	A	9	1, 2, OR NEITHER	ON					ON	ON	ON		ON		

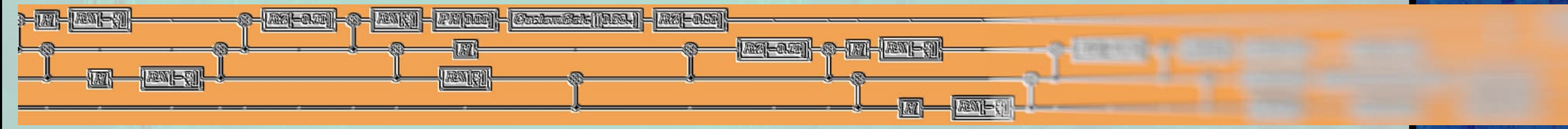
NEONS ON DURING SQUARE ROOTING

ADD. TIME	PROG. RING	PLACE RING	RECEIVER	1	2	3	D'9	4	N9	5	6	7	D	8	N⁻	9	N⁺	10
I-1	A	9		①							ON	ON	ON					ON
2	B	9									ON	ON	"					ON
3	1	9									ON	ON	"					ON
4	A	9	D5, +1								ON		"					ON

CSC

Conclusions

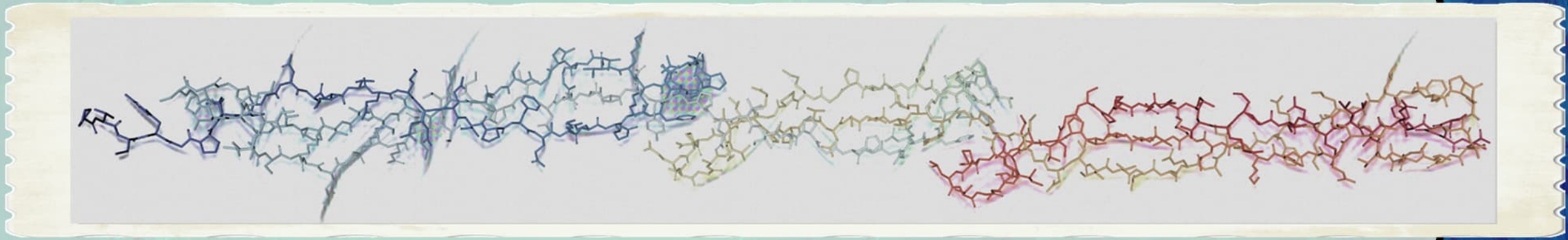
- We already have quantum computers that can outperform the best supercomputers, *but not really for useful* ***computations***
- Need to develop **algorithms that are short enough, so they complete before the quantum computer crashes!**



- **A truly multidisciplinary problem:** combine knowledge of mathematics, quantum physics, and computer science **with the application field**!
- Only with **both the software algorithms and hardware in place**, can the quantum revolution for computational modelling begin!

Conclusions

- Presently, **quantum computing is something of a dream come true for method developers**
  - Plenty of new areas to explore and use your creativity for **high-impact science**!



- **At some point**, quantum computers and quantum computing will **transform from being a subject *of* research to being a tool *for* research**
  - First incremental, then disruptive change in computational power
- **Let's start hacking!**

Dr. Mikael Johansson

Manager, Quantum Technologies
mikael.johansson@csc.fi

linkedin.com/in/mikael-p-johansson

facebook.com/CSCfi

twitter.com/CSCfi

youtube.com/CSCfi

linkedin.com/company/csc---it-center-for-science

github.com/CSCfi