Quantum Computations for Computer Systems Computer Scientists

Michael Kirkedal Thomsen
DIKU, CompSys, Dec 18 2024

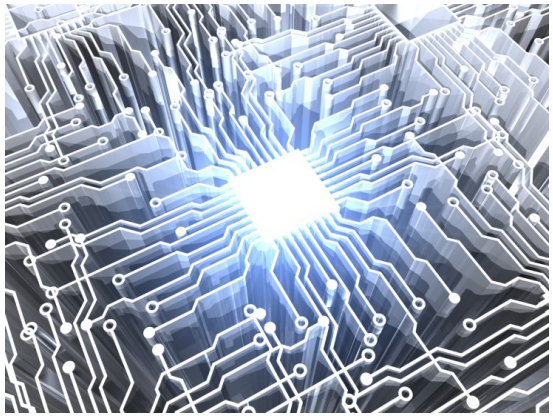
With material from:

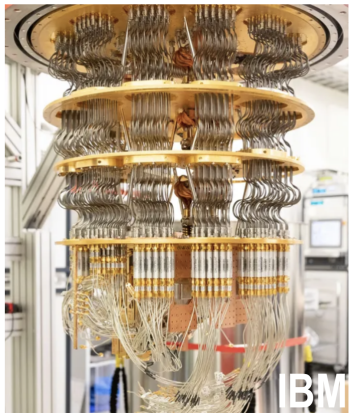
Boris Düdder, DIKU, Univ. Cph
Anders Søndgaard Sørensen, NBI, Univ. Cph
Laura Mancinska, QMath, Univ. Cph
Michael Kastoryano, DIKU, Cph

The hype of Quantum computing

Promise: Build and program an entirely new kind of computers based on quantum mechanics.

- more secure crypto (breaking)
- faster algorithms
- improved communication





Kvantemekanik er svært

Det kan da ikke passe

Albert Einstein

Jo det kan!

Niels Bohr

Fra filosofi til teknologi

- Bohr og Einstein: Mange diskussioner om betydning
  "Gud spiller ikke med terninger"

  Filosofi/religion

- I dag: kan lave eksperimenter med et atom Verden er bare mærkelig

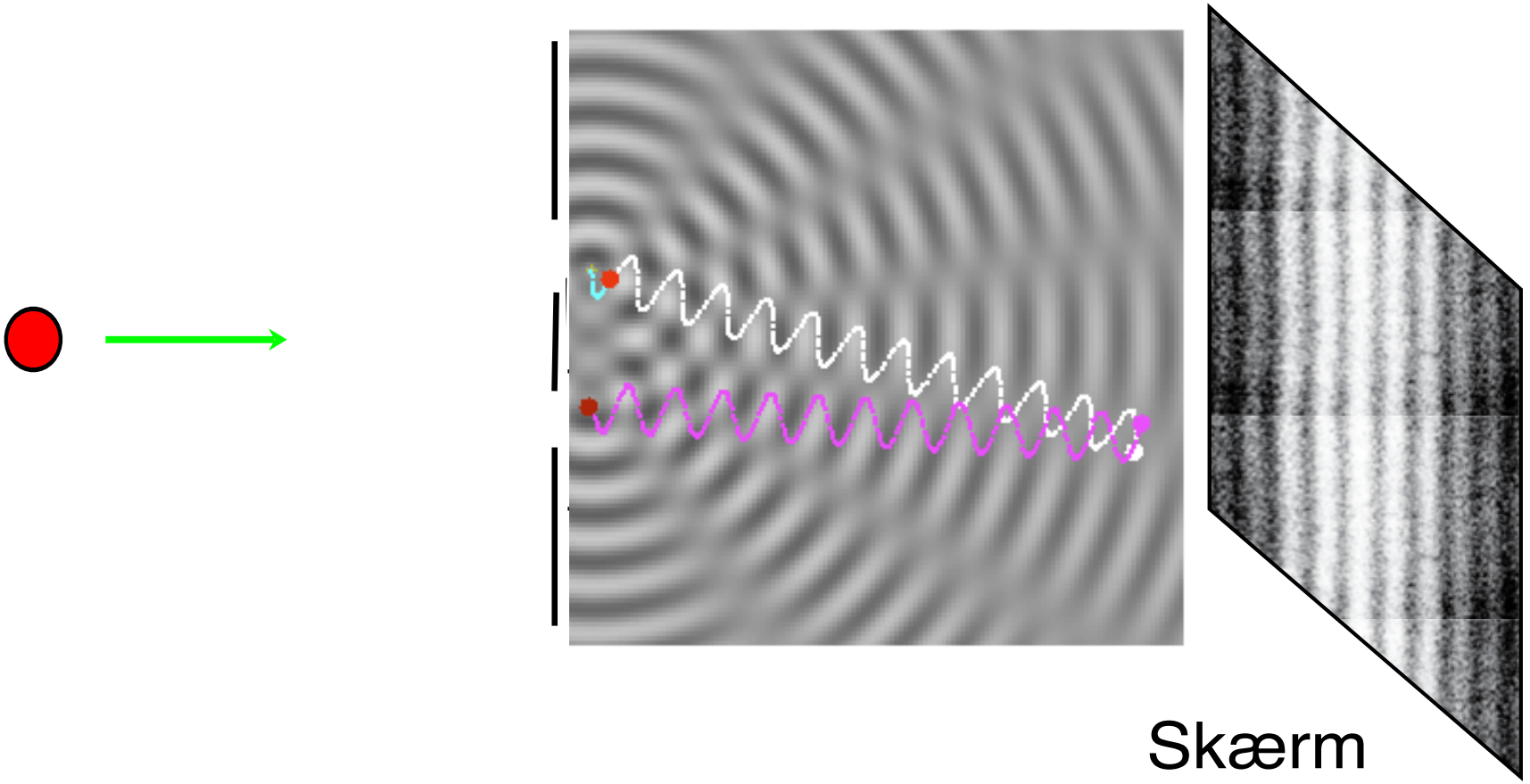
  Fysik

- Kan det mærkelig bruges til noget?

- Kvanteinformation: Gem en bit i et atom => nye muligheder

  Teknologi

Particle wave duality



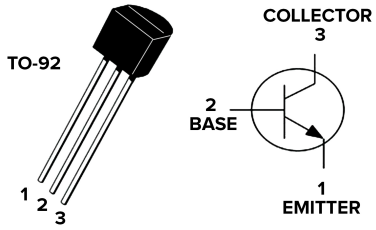
What is a Quantum Computing

Computer using quantum physical principles for computation

- Specialized algorithms for quantum computers
- Quantum Computer Programming languages and systems are different from traditional computer systems

Classical computing

Symbol	Truth Table		
 2-input NAND Gate	A	B	Q
	0	0	1
	0	1	1
	1	0	1
	1	1	0
Boolean Expression Q = A NAND B			



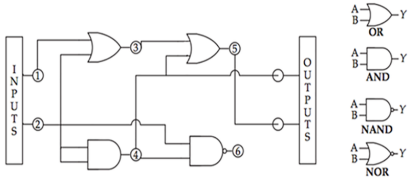
Formally: Input $\text{`}0\text{`} = \begin{pmatrix}1\\0\end{pmatrix}$ or $\text{`}1\text{`} = \begin{pmatrix}0\\1\end{pmatrix}$

Composition

$\text{`}01\text{`} = \text{`}0\text{`} \otimes \text{`}1\text{`}$

Operation NAND $\begin{pmatrix}1 & 1 & 1 & 0\\0 & 0 & 0 & 1\end{pmatrix}$

Larger computation

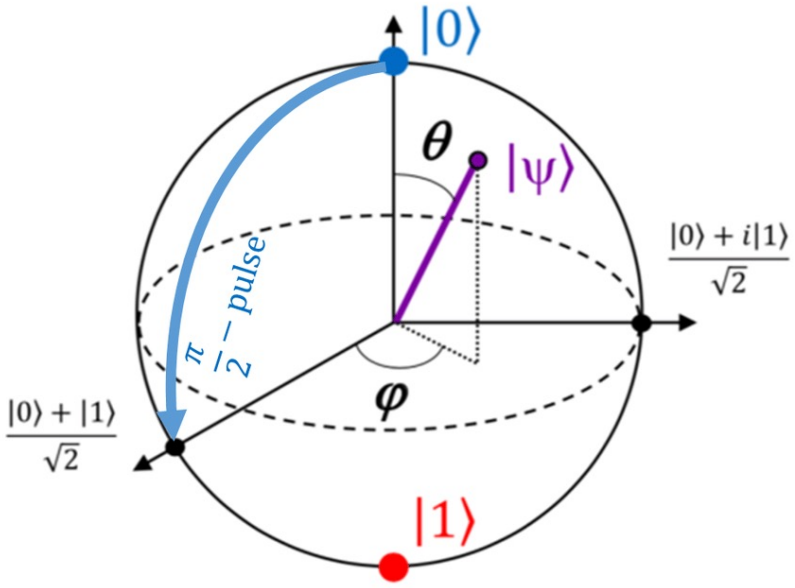


Deterministic input, deterministic output.

Operation errors $\mathcal{O}(10^{-15})$

Quantum computing

Input $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$ $|\alpha|^2 + |\beta|^2 = 1$ $|0\rangle = \begin{pmatrix}1\\0\end{pmatrix}, \ |1\rangle = \begin{pmatrix}0\\1\end{pmatrix}$



$$|\psi\rangle = \cos\frac{\theta}{2}|0\rangle + e^{i\varphi}\sin\frac{\theta}{2}|1\rangle$$

Quantum computing

Input $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$ $|\alpha|^2 + |\beta|^2 = 1$ $|0\rangle = \begin{pmatrix}1\\0\end{pmatrix},\ |1\rangle = \begin{pmatrix}0\\1\end{pmatrix}$

Composition

$|01\rangle = |0\rangle \otimes |1\rangle$

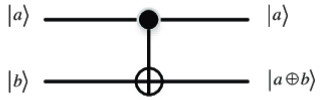
But also $|\phi\rangle = \begin{pmatrix}\phi_0\\\phi_1\\\phi_2\\\phi_3\end{pmatrix}$ $\sum_j |\phi_j|^2 = 1$

Legitimate input state

Like the square root of a probability vector.

Operation

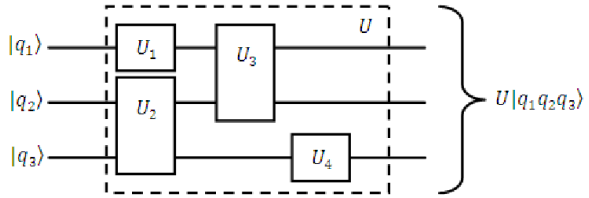
$$CNOT = \begin{bmatrix}1&0&0&0\\0&1&0&0\\0&0&0&1\\0&0&1&0\end{bmatrix}$$



Deterministic input, deterministic output, but with quantum superposition and entanglement.

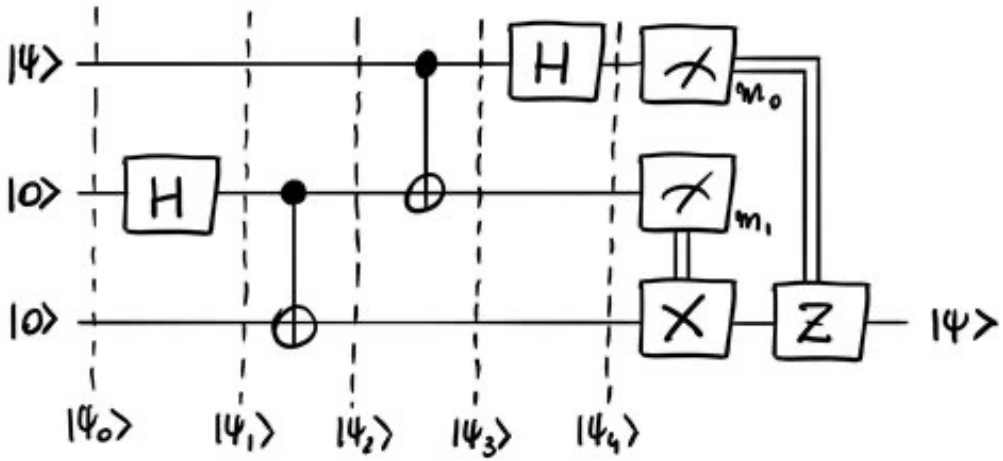
General: $U|\psi\rangle$ $UU^\dagger = \mathrm{id}$

Larger computation



Computations are large unitary matrices, decomposed as a sequence of smaller ones

Quantum circuit and running



https://quantum-computing.ibm.com/

Operator	Gate(s)	Matrix
Pauli-X (X)	X	$\begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$
Pauli-Y (Y)	Y	$\begin{bmatrix} 0 & -i \\ i & 0 \end{bmatrix}$
Pauli-Z (Z)	Z	$\begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}$
Hadamard (H)	H	$\frac{1}{\sqrt{2}}\begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$
Phase (S, P)	S	$\begin{bmatrix} 1 & 0 \\ 0 & i \end{bmatrix}$
$\pi/8$ (T)	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)	Z	$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & -1 \end{bmatrix}$
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)		$\begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix}$

What is a Quantum Computer

Computer using quantum physical principles for computation

- Specialized algorithms for quantum computers
- Quantum Computer Programming languages and systems are different from traditional computer systems

Principles:

- **Superposition** principle
- Quantum **entanglement**

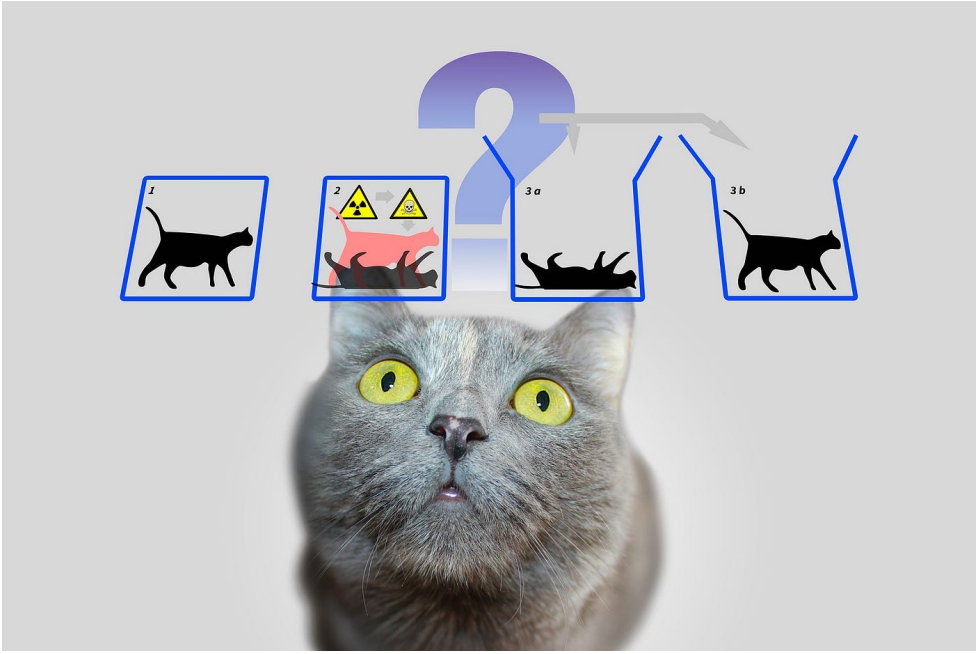
Superposition principle

Superposition principle: A quantum system consists of a superposition of equal physical quantities (these do **not** interfere with each other)

- Example: | 0> + | 1> = State is superposition of the two classic states "Bit is 0" and "Bit is 1". (Qubit = quantum bit)

Problem: **Decoherence**

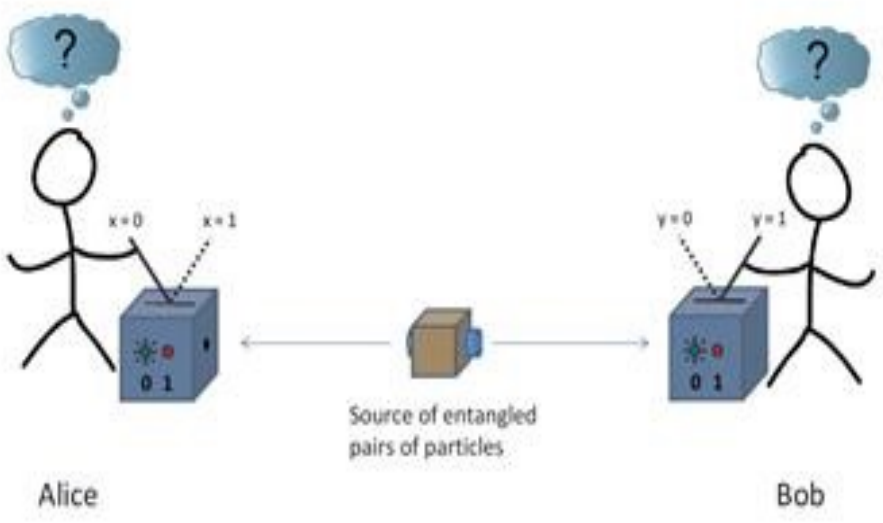
- Decoherence is agent of classicality: It describes the **loss** of quantum properties when a quantum system interacts with its surroundings (classical world).



Quantum entanglement

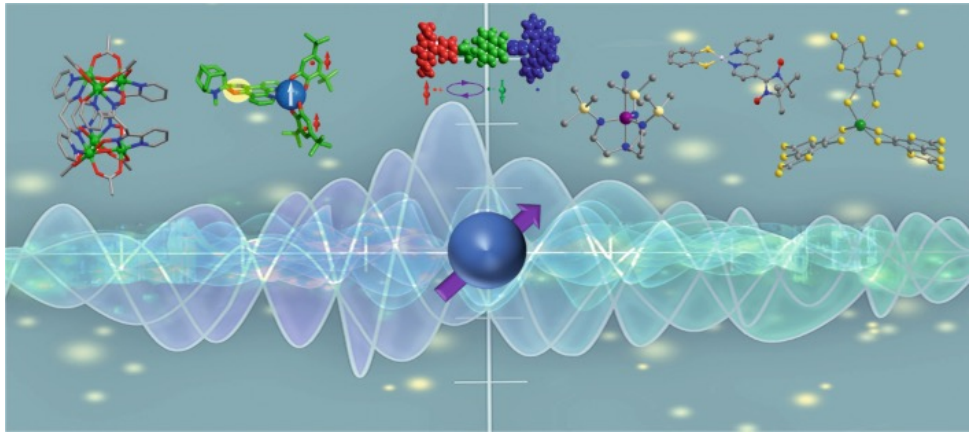
Quantum entanglement: The system state of two or more particles can not be described as a combination of independent one-particle states, but only by a **common** state

For example: a system of **two** entangled **spins** adding up to **zero** can be composed of a **superposition of pairs of spins** with opposite directions while it is absolutely undetermined in which direction the individual spin points.

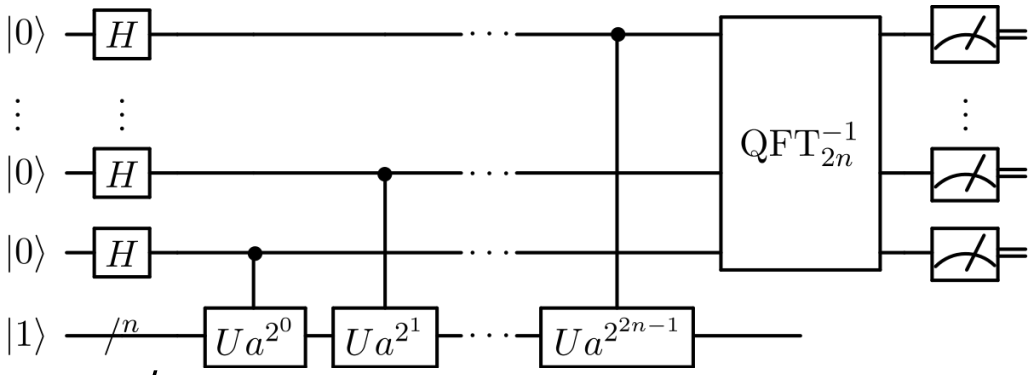


What can quantum computers do

- Quantum Fourier Transform
- Quantum simulations
  - Quantum dynamics
  - Quantum chemistry

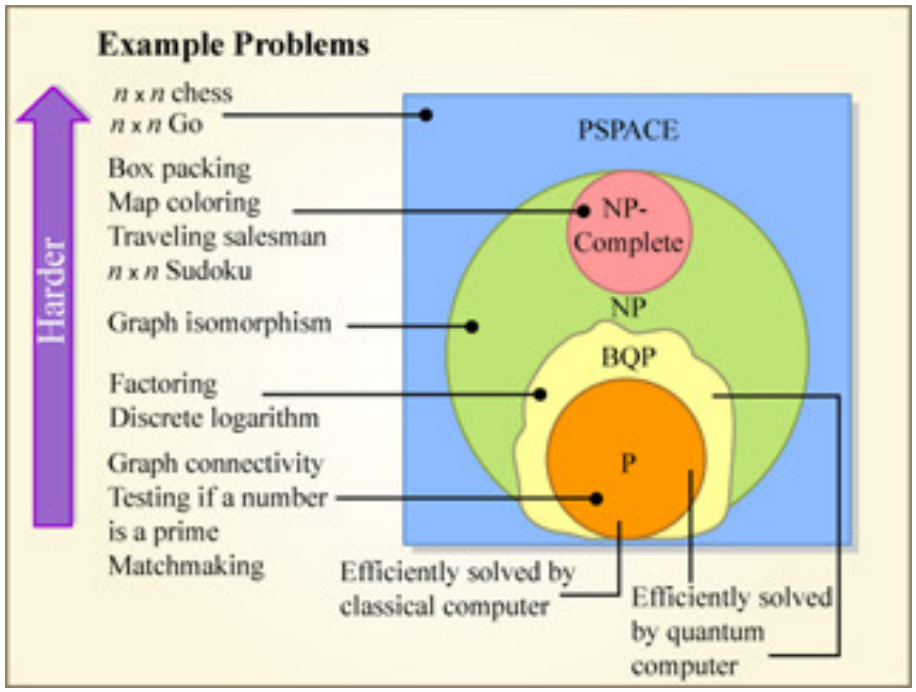


- Quantum algorithms
  - Shor's algorithm
  - Groover's search



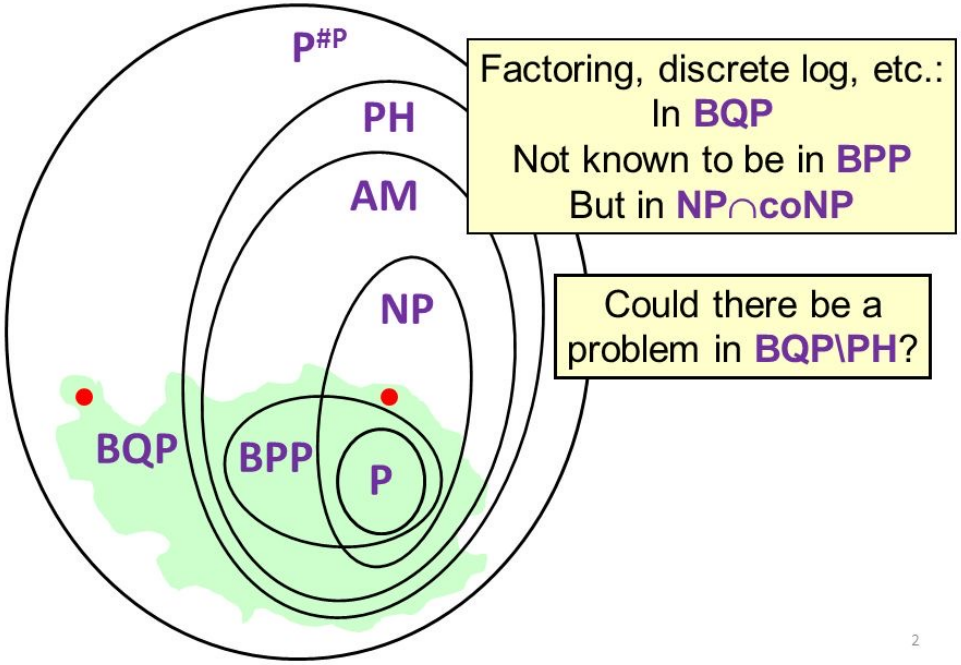
- https://quantumalgorithmzoo.org/

What can quantum computers do

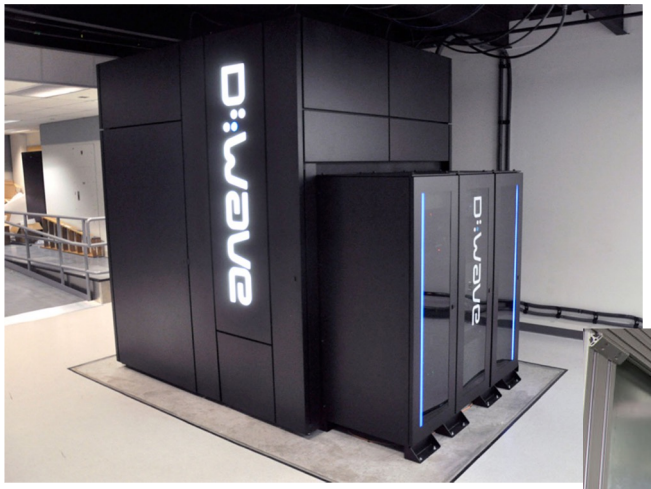


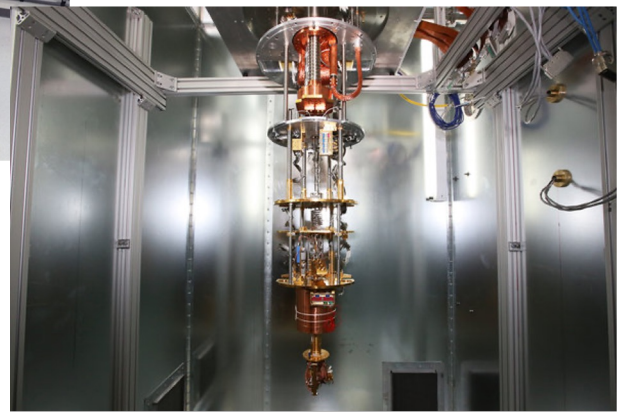
- Quantum machine learning?
- Quantum TSP?
- Quantum Finances

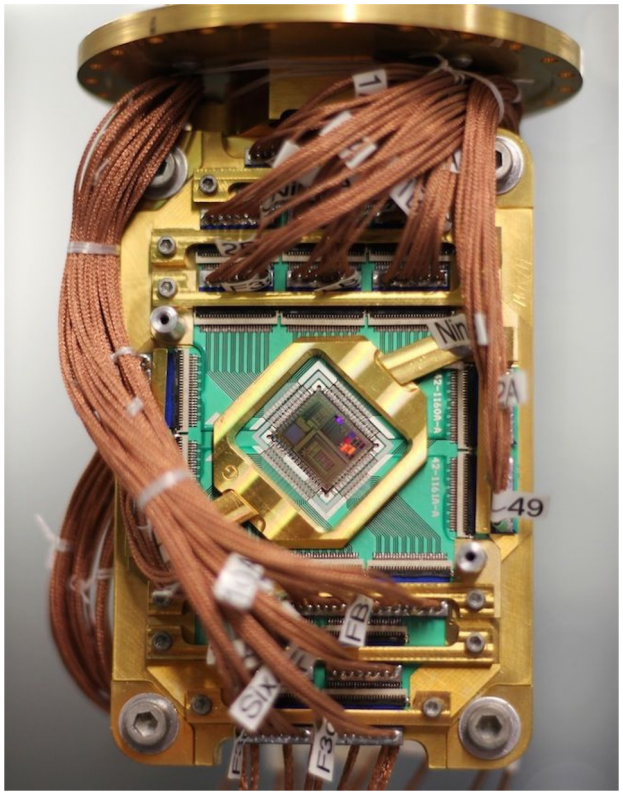
Quantum Computing: Where Does It Fit?



Quantum Computer



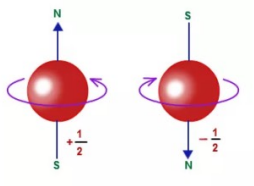


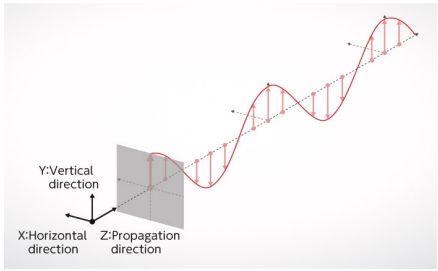


How do you build qubits?

Microscopic systems have mostly continuous degrees of freedom

Particle spin and light polarisation are discrete





Because quantum systems are very fragile, operations are very noisy

Qubit Type	Pros/Cons	Select Players
Superconducting	***Pros***: High gate speeds and fidelities. Can leverage standard lithographic processes. Among first qubit modalities so has a head start. ***Cons***: Requires cryogenic cooling; short coherence times; microwave interconnect frequencies still not well understood.	rigetti, Google, IBM Q, QuTech, OQC, IQM, qci Quantum Circuits, Inc, 本源量子 Origin Quantum
Trapped Ions	***Pros***: Extremely high gate fidelities and long coherence times. Extreme cryogenic cooling not required. Ions are perfect and consistent. ***Cons***: Slow gate times/ operations and low connectivity between qubits. Lasers hard to align and scale. Ultra-high vacuum required. Ion charges may restrict scalability.	IONQ, AQT, QUANTINUUM, oxford ionics, Universal Quantum
Photonics	***Pros***: Extremely fast gate speeds and promising fidelities. No cryogenics or vacuums required. Small overall footprint. Can leverage existing CMOS fabs. ***Cons***: Noise from photon loss; each program requires its own chip. Photons don't naturally interact so 2Q gate challenges.	PsiQuantum, XANADU, QUIX QUANTUM, ORCA Computing
Neutral Atoms	***Pros***: Long coherence times. Atoms are perfect and consistent. Strong connectivity, including more than 2Q. External cryogenics not required. ***Cons***: Requires ultra-high vacuums. Laser scaling challenging.	ColdQuanta, QuEra Computing Inc., atom computing, PASQAL
Silicon Spin/Quantum Dots	***Pros***: Leverages existing semiconductor technology. Strong gate fidelities and speeds. ***Cons***: Requires cryogenics. Only a few entangled gates to-date with low coherence times. Interference/cross-talk challenges.	intel, Silicon Quantum Computing, diraq, QUANTUM MOTION, QUANTUM BRILLIANCE

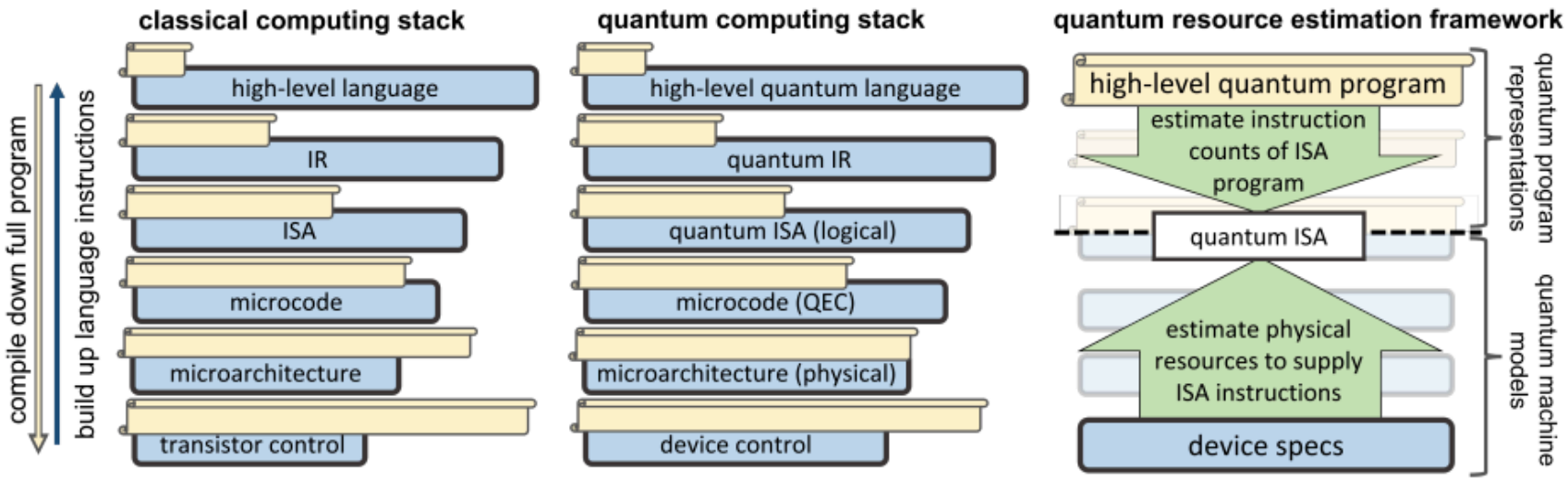
State of the art

Modality	Super-conducting	Trapped Ion	Photonic	Neutral Atom	Silicon Spin
# Qubits	127Q	32Q	20 Photons/ 216 Qumode	100Q	2Q
T2 Lifetime	Short 15μs-256μs	Long 0.2s-50s	Short 150μs	Long 0.2s-10s	Mixed 1μs-0.5s
2Q Gate Fidelity	High 99%-99.7%	High 98.5%-99.92%	Promising 98%	Promising 97.4%	Promising 90%-98%
Gate Speed	Fast 10ns-196ns	Mixed 1μs-3ms	Very Fast 1ns	Medium 1μs	Fast 0.8ns-80ns

Further considerations:

- How well can they be scaled up?
- How much energy do they consume?
- How large does the machine have to be?

Quantum Stack

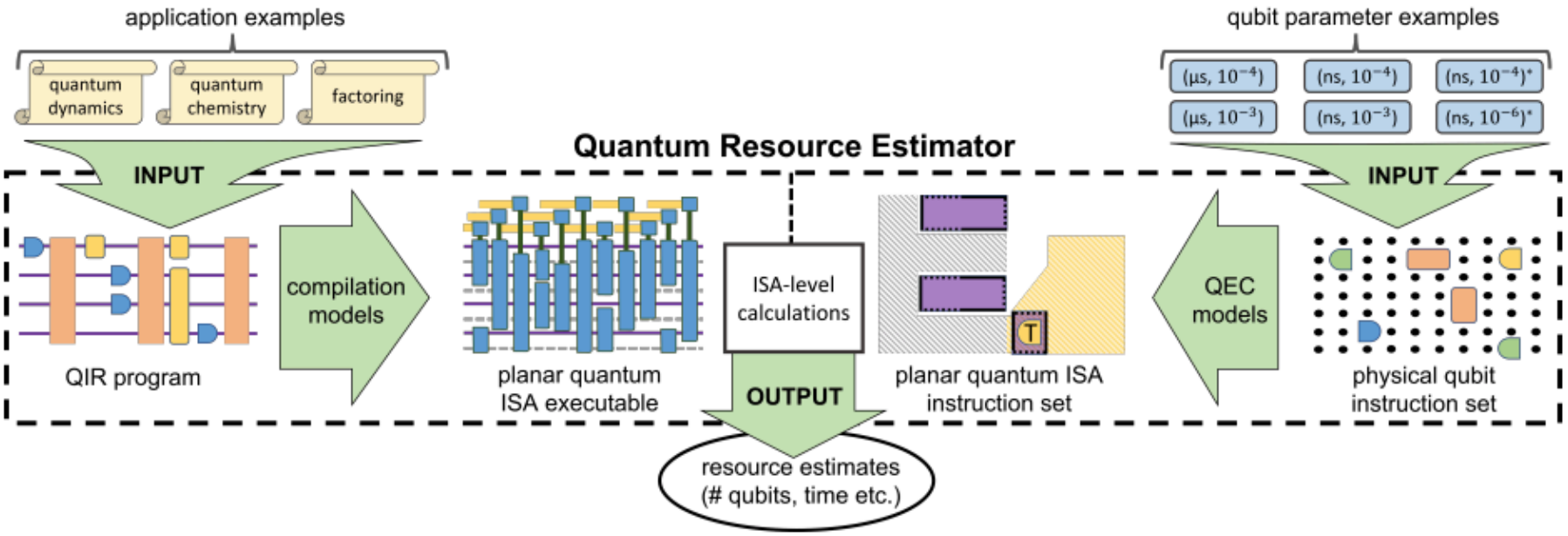


M. Beverland et. al, arXiv:2211.07629

Major differences:

- Multiple platforms, each with advantages and disadvantages
- Microarchitecture -> microcode very costly (because of quantum error correction)
- Applications are scarce, very specialised

Resource estimation



Resource estimation

Minimal applications

application	algorithm execution accuracy $1-\epsilon$	quantum executable parameters			quality requirements	
		Q	$C_{\min}$	M	max P	max P_T
quantum dynamics	0.999	230	$1.5 \cdot 10^5$	$2.4 \cdot 10^6$	$9.7 \cdot 10^{-12}$	$1.4 \cdot 10^{-10}$
quantum chemistry	0.99	2740	$4.1 \cdot 10^{11}$	$5.4 \cdot 10^{11}$	$3.0 \cdot 10^{-17}$	$6.1 \cdot 10^{-15}$
factoring	0.667	25481	$1.2 \cdot 10^{10}$	$1.5 \cdot 10^{10}$	$3.5 \cdot 10^{-16}$	$7.4 \cdot 10^{-12}$

 # qubits

 # sequential operations

 # required qubit quality

Cost of error correction

qubit parameter examples	QEC code selected	logical qubit parameters for $P(d) \leq 3.5 \cdot 10^{-16}$			T factory parameters for $P_T(\mathcal{D}) \leq 7.4 \cdot 10^{-12}$	
		distance d	# qubits $n(d)$	logical time step $\tau(d)$	# qubits $n(\mathcal{D})$	duration $\tau(\mathcal{D})$
(μs, 10^{-3}) qubit	surface	27	1458	16 ms	17640	163 ms
(μs, 10^{-4}) qubit	surface	13	338	7 ms	4840	85 ms
(ns, 10^{-3}) qubit	surface	27	1458	10 μs	33320	128 μs
(ns, 10^{-4}) qubit	surface	13	338	5 μs	5760	72 μs

Error correction trades off qubit quality for qubit number and slowdown (and a lot of infrastructure)

 # qubit overhead

 # clock speed (slowdown)

Quantum error correction

Classical error correction:

Example: repetition code

$00000 \rightarrow$ ‘0‘

$11111 \rightarrow$ ‘1‘

Decode errors by majority vote

$$\epsilon_L \sim e^{-\alpha(\epsilon)d}$$

$$d \propto N$$

`d` is the distance of the code.
Relation only holds below threshold

$\sim 50\%$

Quantum error correction:

Need to protect against `flip` and `phase` errors.

Flip $|0\rangle \leftrightarrow |1\rangle$

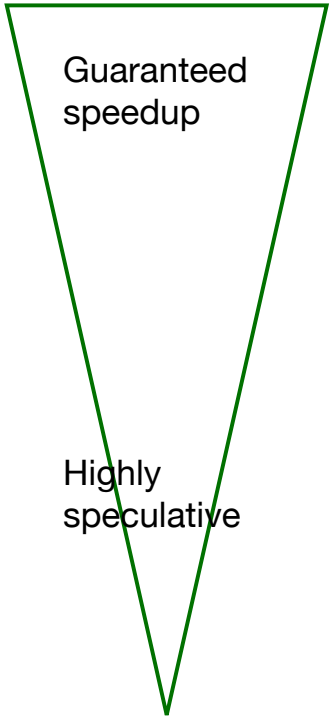
Phase $|1\rangle \leftrightarrow -|1\rangle$

Natural error rates are very high

Suppression of errors: $\epsilon_L \sim e^{-\alpha(\epsilon)d}$ $d \propto \sqrt{N}$

Threshold: $\sim 1\%$

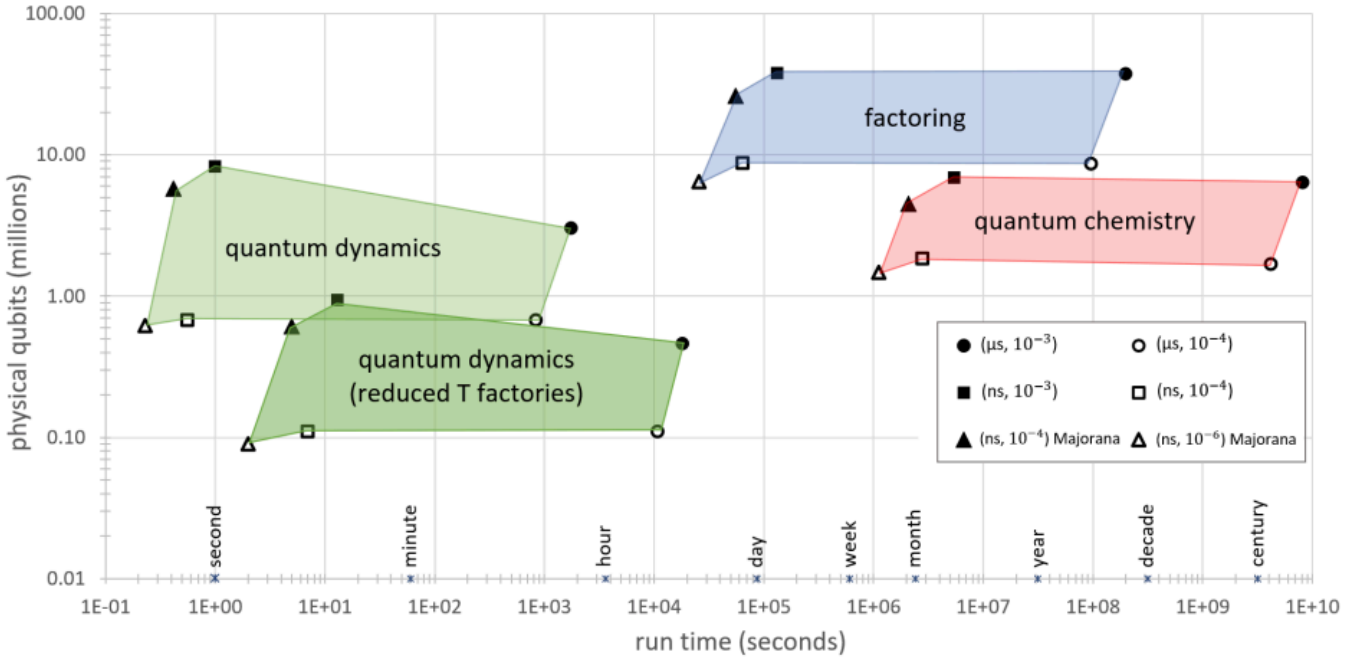
Applications



Factoring, discrete log	$\sim 10^7$qubits,	$\sim 10^{10}$gates
Materials simulation	$\sim 10^6$qubits,	$\sim 10^9$gates
Molecules simulations	$\sim 5 \times 10^6$qubits,	$\sim 10^{12}$gates
Everything else	$\sim 10^{4-}$ qubits,	$\sim 10^{13-}$ gates

Need to be extremely cautious with promises

Resource estimation

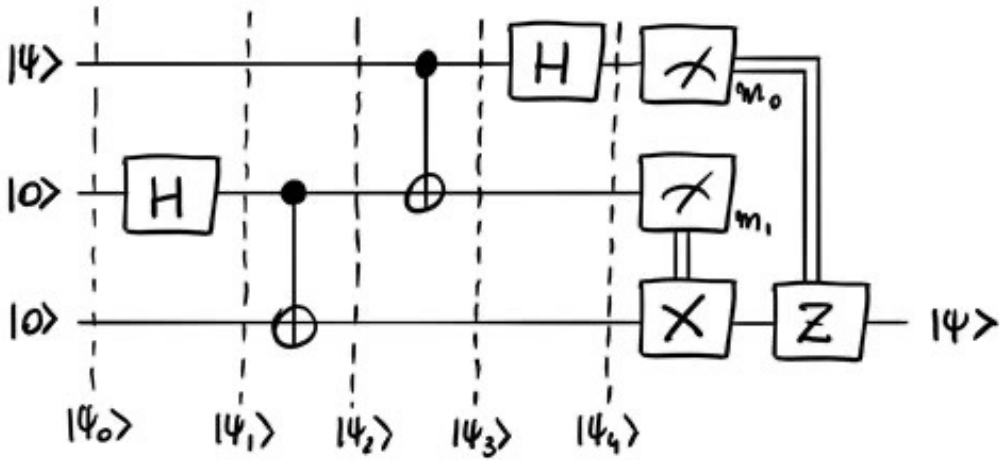


Three most promising applications of quantum computing

Boxes indicate minimal instances where we are confident that they are challenging for classical computers

Note, energy/cost is not represented here

Quantum circuit and running



https://quantum-computing.ibm.com/

Operator	Gate(s)	Matrix
Pauli-X (X)	X	$\begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$
Pauli-Y (Y)	Y	$\begin{bmatrix} 0 & -i \\ i & 0 \end{bmatrix}$
Pauli-Z (Z)	Z	$\begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}$
Hadamard (H)	H	$\frac{1}{\sqrt{2}}\begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$
Phase (S, P)	S	$\begin{bmatrix} 1 & 0 \\ 0 & i \end{bmatrix}$
$\pi/8$ (T)	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)	Z	$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & -1 \end{bmatrix}$
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)		$\begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix}$

Quantum programming languages

- Python (Actual Quantum Programming Language)
- Qiskit (Open-source Programming Tool)
- Ocean™ (Quantum Computing Programming Suite)
- Q# (Quantum Computing Programming Algorithm)
- Cirq (Google AI Programming Language)
- Silq
- Qrisp

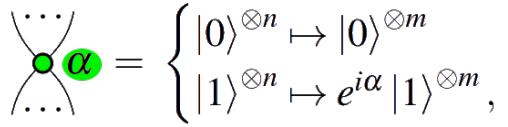
- https://qrisp.eu/general/tutorial/tutorial.html

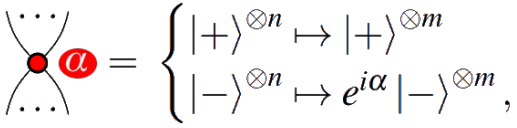
```
def solve[n:!N](bits:!𝔹^n){
  // prepare superposition between 0 and 1
  x:=H(0:𝔹);
  // prepare superposition between bits and 0
  qs := if x then bits else (0:int[n]) as 𝔹^n;
  // uncompute x
  forget(x=qs[0]); // valid because `bits[0]==1`
  return qs;
}

// EXAMPLE CALL

def main(){
  // example usage for bits=1, n=2
  x := 1:!int[2];
  y := x as !𝔹^2;
  return solve(y);
}
```

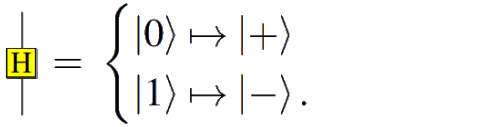
```
fun teleport (q1 : qubit<P>) : qubit<P> = (* pure type *)
  let q23 : (qubit & qubit)<P> = bell_pair () in
  let (q2 : qubit<M>, q3 : qubit<M>) = q23 in
  let (q1 : qubit<M>, q2 : qubit<M>) = CNOT (q1, q2) in
  let q1 : qubit<M> = H (q1) in
  let (q2 : qubit<M>, q3 : qubit<M>) = CNOT (q2, q3) in
  let (q1 : qubit<M>, q3 : qubit<M>) = CZ (q1, q3) in
  let q123 : ((qubit & qubit) & qubit)<M> = ((q1, q2), q3) in
  (* assert ((q1, q2), q3) is pure; check statically *)
  let q123 : ((qubit & qubit) & qubit)<P> = cast<P>(q123) in
  (* assert q3 is separable from (q1, q2); check dynamically *)
  let (q12 : (qubit & qubit)<P>, q3 : qubit<P>) = split<P>(q123) in
  let _ : bool * bool = measure (q12) in q3
```

Quantum models and algebra!

- Quantum Turing machines
  - Quantum Theory, the Church-Turing Principle and the Universal Quantum Computer*;* David Deutsch, 1985
- ZX calculus
  - Coecke, Duncan
  - https://zxcalculus.com/
- Programming the quantum future; Valiron, Ross, Selinger, Alexander, Smith
  - https://dl.acm.org/doi/10.1145/2699415
- Reversible quantum combinators: UΠ; Heunen, Kaarsgaard
  - https://arxiv.org/abs/2107.12144







How much hype is there in QC?

- Error correction (fault tolerance) is necessary
- Real quantum computers will still take a long time
- There are only few very promising applications at present
- Classical control hardware is also a big part of the game
- No algebraic model for QC

- There are many pitfalls, but the upsides seems to be real

- Related topics (not QC) so not to forget
  - Quantum key exchange
  - Post-Quantum encryption