Introduction to Quantum Computing

Wim Lavrijsen* (LBNL)

(With material from tutorials of Costin Iancu and Bert de Jong)

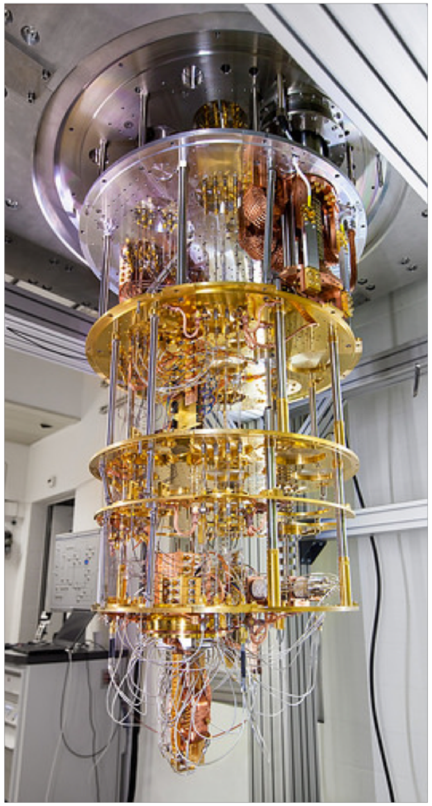
University of Tokyo
Feb 8, 2019

*(WLavrijsen@lbl.gov)

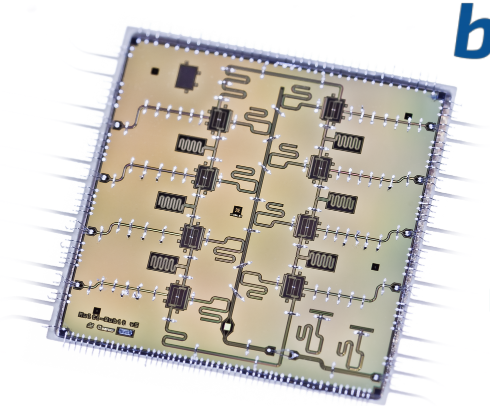
Quantum Computing @ Berkeley

Involvement across the full range:

- Qubit, chip, control hardware design
- Software stack from compilers to control
- Algorithms development
- Domain science applications
- Many partners in academia and industry:



CENTER for QUANTUM COHERENT SCIENCE
BIQC
bleximo
IBM Q™
SIMONS INSTITUTE for the Theory of Computing
CAMBRIDGE QUANTUM COMPUTING LIMITED
MIT LINCOLN LABORATORY
rigetti
Google
Lawrence Livermore National Laboratory



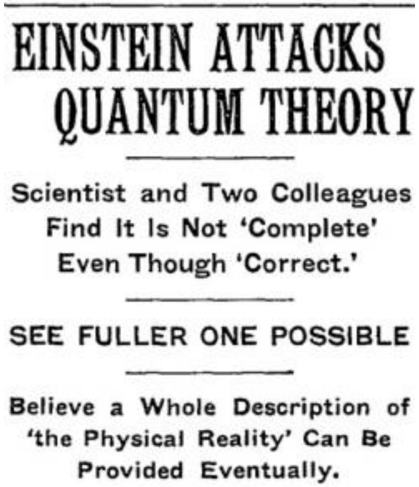
https://berkeleyquantum.org
https://cs.lbl.gov/what-we-do/quantum-information-sciences/

Why Quantum Computing?

- Computational speedups over classical
  - Black-box query (Deutsch-Jozsa 1992; exponential)
  - Factoring of numbers (Shor 1994; exponential)
    - Breaks RSA; note that Shor's algorithm was known before the internet (and online encryption) became a thing!
  - Database search (Grover 1996; quadratic)
- "Unbreakable" encryption protocols
  - Ability to set probability of decryption by adversary
- Quantum simulation
  - Efficient (polynomial) simulation of quantum systems such as molecules (Feynman 1982, Lloyd 1996; exponential)
- Efficient optimization algorithms

Quantum Computing Ingredients

- A different unit of information: qubits
  - Represents arbitrary combination of two states: |0⟩ and |1⟩
  - Together with operations “gates” to control its state
- Different computational resources
  - Superposition
  - Entanglement ⇐ *this is the crucial one*
  - Interference
- Information conserving
  - Unitary operations, no energy loss
- Fundamental constraints on algorithms
  - Essentially analog computing with digital I/O
  - Results are probabilistic



EINSTEIN ATTACKS QUANTUM THEORY

Scientist and Two Colleagues Find It Is Not ‘Complete’ Even Though ‘Correct.’

SEE FULLER ONE POSSIBLE

Believe a Whole Description of ‘the Physical Reality’ Can Be Provided Eventually.

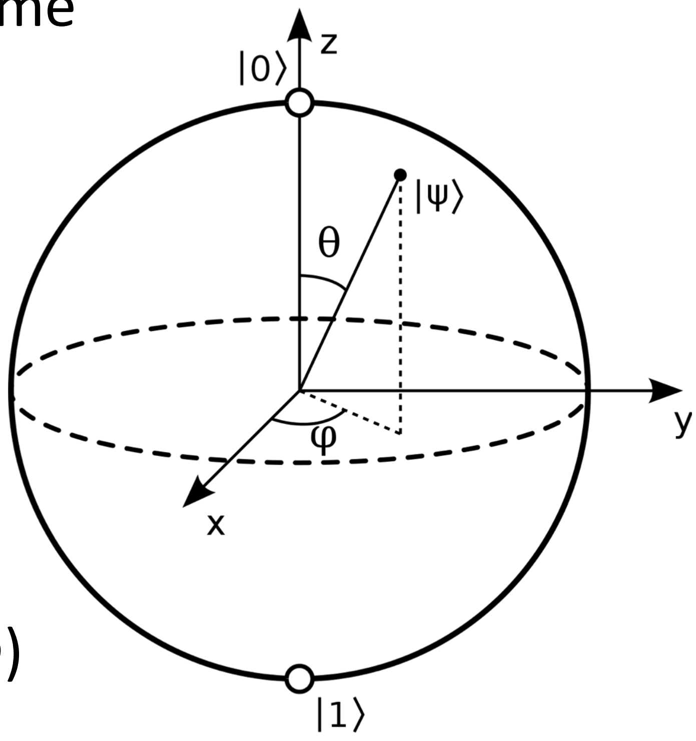
(New York Times)

The Qubit

- Quantum unit of computation
  - Physics: a 2-state quantum system (electron spin, levels of harmonic oscillator, single photon polarization, etc.)
  - Popular: "both 0 and 1 at the same time"
  - Mathematically: a state vector in $\mathbb{C}^2$
- Bloch sphere representation

$$\boxed{|\psi\rangle = e^{i\gamma}\,(\cos(\theta/2)|0\rangle + e^{i\varphi}\sin(\theta/2)|1\rangle)}$$

  - Amplitude is normalized
  - Global phase (γ) is not physical
  - Single-qubit gates change the state
  - Gates are *rotations*: $R_X(\theta)$, $R_Y(\theta)$, $R_Z(\varphi)$
    - E.g. NOT (bit-flip) is a $|1\rangle \rightarrow |0\rangle$ is $R_X(\pi)$



(Smite-Meister - CC BY-SA 3.0)

Superposition

Classical

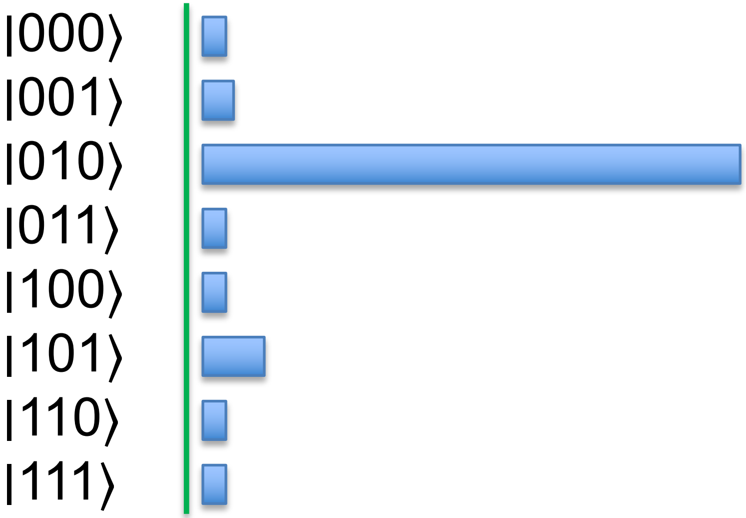
Bits represents a single value, out of 2^N possible bit strings, e.g. for 3 classical bits:

010 == 2

Measurement reduces the ensemble to a single classical bit string, with probability equal to its amplitude squared. Sampling the whole space takes exponential time.

Quantum

Bits represent an ensemble of all 2^N possible bit strings, from which you can sample, e.g. for 3 qubits:



Increases "working memory" up to an exponential factor.

Entanglement

- Unifies multiple qubits into a single state

Example (*maximum entanglement*):

$$|\psi\rangle = (|00\rangle + |11\rangle)/\sqrt{2}$$

$\Longrightarrow$ measuring one qubit determines state of the other

- A "physical" resource
  - Can be "created", used, and "destroyed"
  - States have a quantifiable amount of entanglement

Example: $|\psi\rangle = \alpha|00\rangle + \beta|01\rangle + \gamma|11\rangle$

- Allows "instantaneous" operation on all qubits
  - Physics: ?????
  - Popular: with superposition, "try all solutions in parallel"
  - Mathematically: off-diagonal elements in $2^N \times 2^N$ state matrix

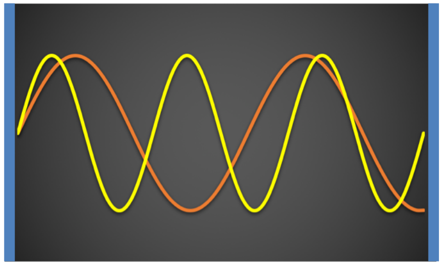
Increases information density up to an exponential factor.

Interference

- Total probability across all bit strings sums to 1
  - Combined effect of superposition and entanglement

⇒ *As one solution becomes more likely (larger amplitude), others have to become less likely (lower amplitude).*

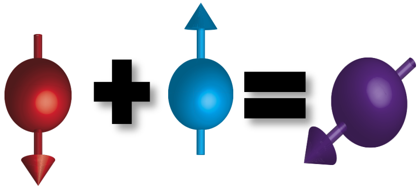
- Amplify right solution, suppress others
  - Physics: wave mechanics
  - Popular: music/orchestra
  - Mathematics: complex (ℂ) math



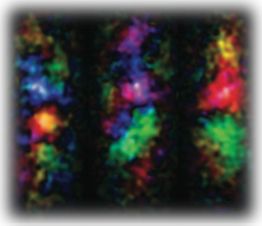
Example, Shor factorization: 3 and 5 fit a whole number of times in 15 ⇒ "standing waves", others interfere destructively.

Interference is how quantum algorithms are designed to work.

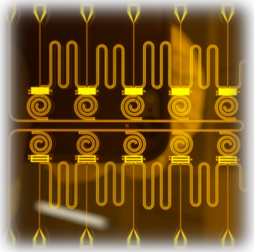
Quantum Computing Hardware



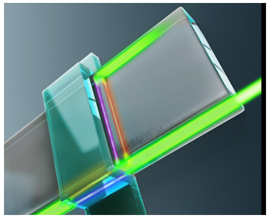
ELECTRONS
SPIN UP + SPIN DOWN



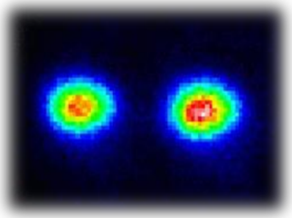
ATOMS



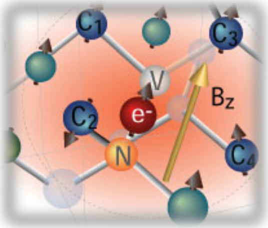
SUPER-CONDUCTING



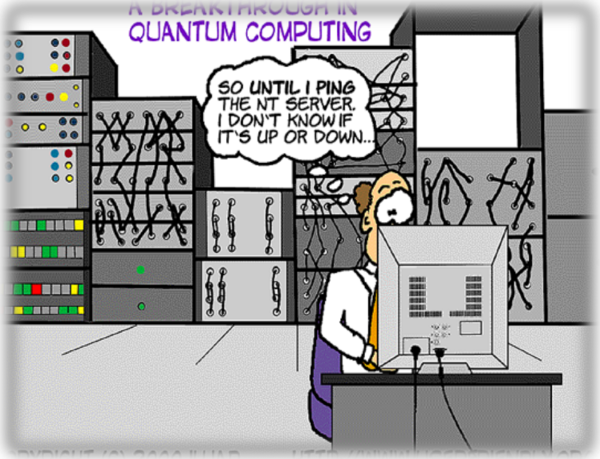
MAJORANA QUASI-PARTICLE



IONS



SOLID STATE (spins)



D-Wave
The Quantum Computing Company™

IBM

Google

rigetti

IONQ

Microsoft

intel

Qubit Errors
Relaxation and Decoherence

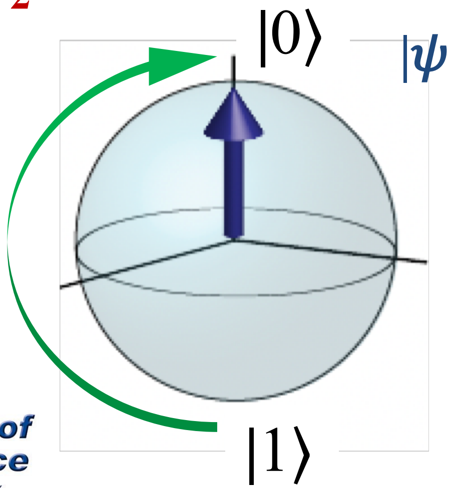
T_1: relaxation, dampening

- Environment exchanges energy with the qubit, mixing the two states by stimulated emission or absorption.
- Perturbation orthogonal to quantization axis, fast fluctuations causing transitions on x, y.
- Important during read-out.
- **Intuitively time to decay from $|1\rangle$ to $|0\rangle$**

T_2: dephasing

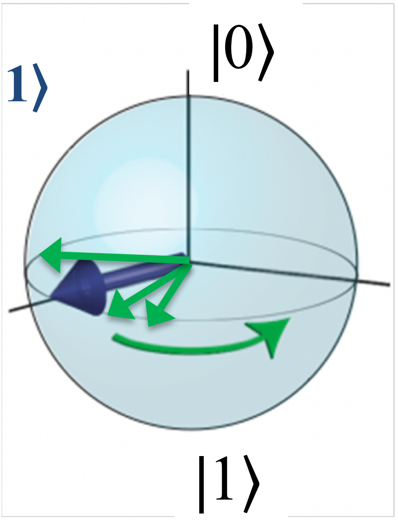
- Environment creates loss of phase memory by smearing energy levels (coupling w/o energy exchange), changing phase velocity.
- Slow perturbation along quantization axis (z); e.g. magnetic flux noise causing phase randomization.
- Important during “computation”, bounds circuit depth (number of consecutive gates)
- **Intuitively time for φ to get imprecise**

$T_1 > T_2$



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

These are not cut-off times, but “half-lives.” Decay is continuous.



Strongest Contenders

Superconducting Qubits

(transmon, flux, phase)

- Qubit – superconducting isles from Josephson junctions + capacitors
- Information encoded by charge on superconductor
- Controlled by microwave
- Dilution fridge required
- Gates: rotations, CNOT, CZ

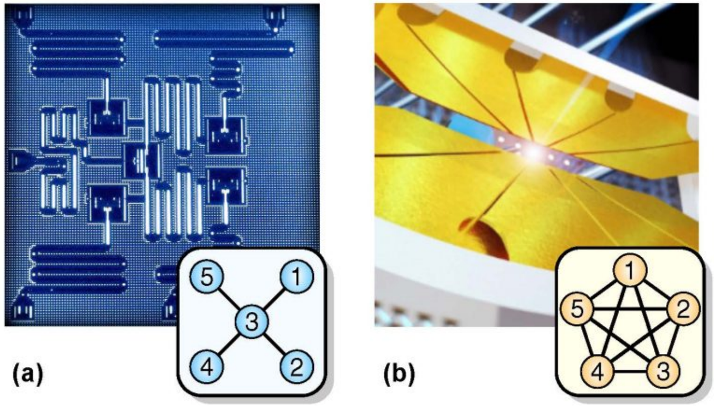
T2: ~100μs
Gate: ~ns

Trapped Ions Qbits/Qudits

(hyperfine, optical)

- Qubit – ion (Ca, Yb) trapped in vacuum
- Information encoded in energy levels
- Controlled by laser
- Room temperature
- Gates: Alltoall, ising, phase shift

T2: ~1s
Gate: ~μs

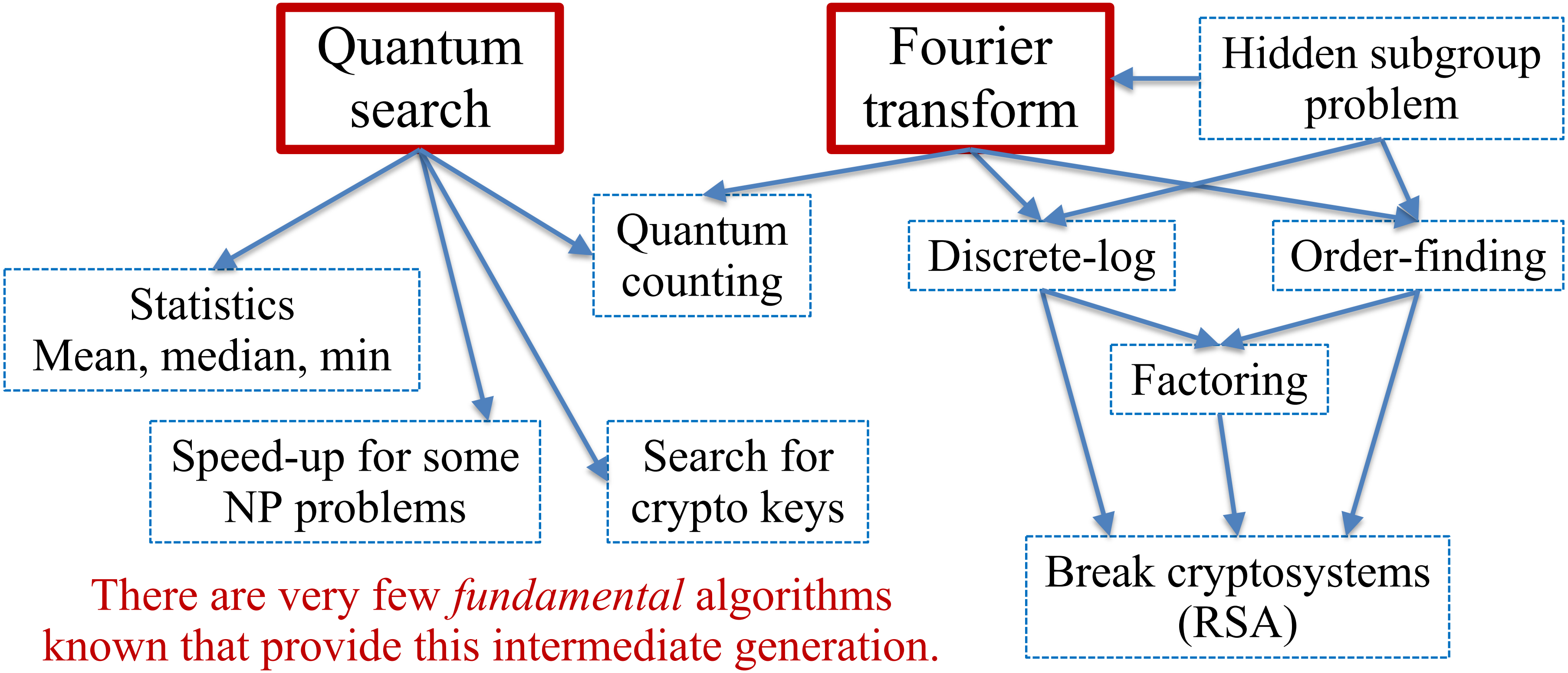


Commercially viable technologies, fully explored

- Superconducting deemed as scalable
- Ions deemed less noisy (T_2), room temp

Algorithms

I/O is classical (N), whereas intermediate is quantum (2^N) . To take advantage of the large state, it should be generated from the classical input (e.g. combinatorics, Fourier transform (QFT), linear solver (HLL), etc.) for quantum to offer speedup.



There are very few *fundamental* algorithms known that provide this intermediate generation.

Nielsen & Chuang

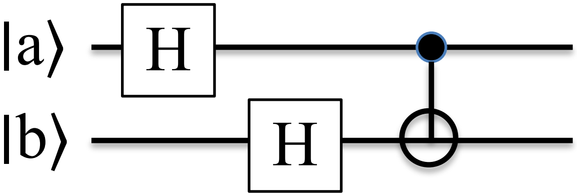
Programming

Circuit Model

- Diagrams of wires and gates.
- Write by hand or generated with science domain software (e.g. OpenFermion for chemistry)
- Hard to generate optimally.

Unitary Linear Algebra

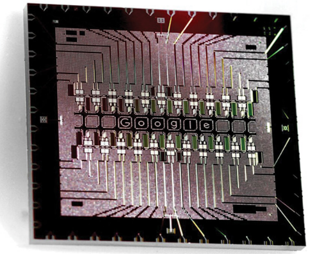
- Matrices and vectors.
- Often more natural to science domain (e.g. coupling strengths in an energy equation).
- Hard to decompose.

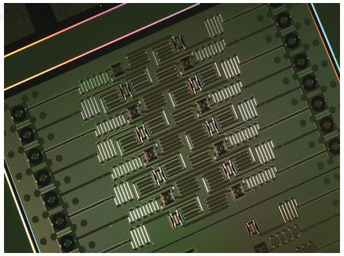


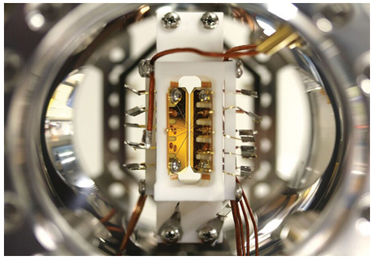
$$\Longleftrightarrow \quad \frac{1}{2} \times \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 \\ 1 & -1 & -1 & 1 \\ 1 & 1 & -1 & -1 \end{bmatrix} \times \begin{bmatrix} a_1 b_1 \\ a_1 b_2 \\ a_2 b_1 \\ a_2 b_2 \end{bmatrix}$$

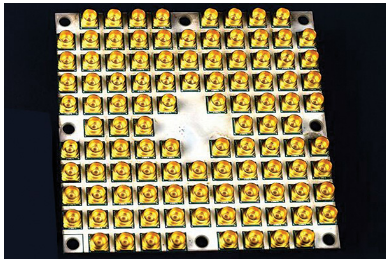
Representations are equivalent, can go back and forth, and even mix.

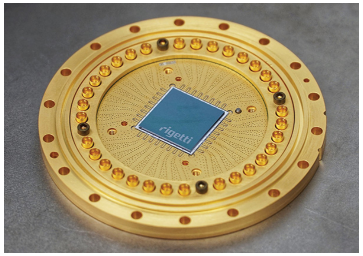
Gate Sets in Superconducting Chips


Google

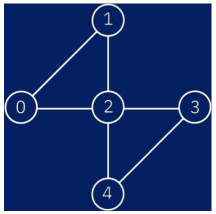

IBM


Rigetti


Intel


IonQ

- Each chip has own native gate set
  - Single qubit, usually rotations, and Hadamard
  - Two-qubit, usually CNOT, CZ (Google), SWAP
  - Hardware level gates, are de facto unique for each chip
- Each chip has a constrained topology
  - Ring, array, mesh, bow-tie

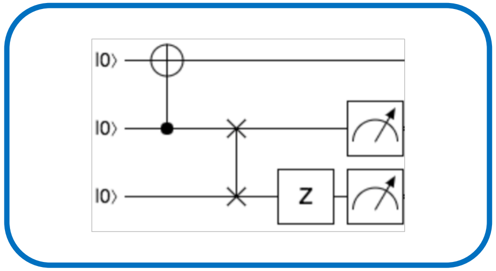


- Compilers to translate gate sets, do mapping
  - *Ideal*: target the actual chip, not the advertised gate set

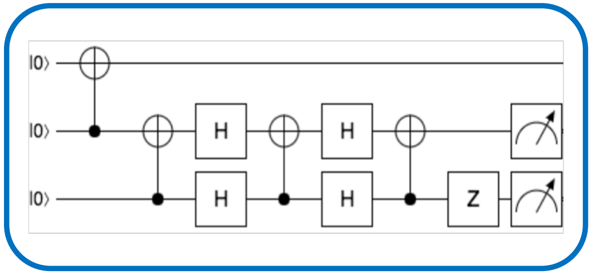
Compilation Toolchain

Scientist ⟷ *Hardware*

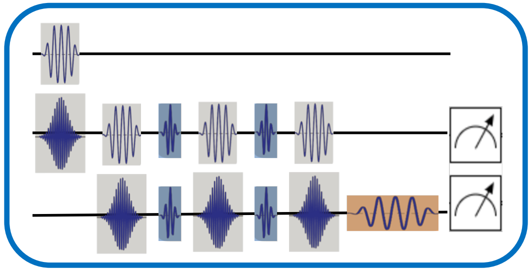
Initial Quantum Algorithm



Compiled Quantum Algorithm



Pulses output by AWG



High level interface

- Arbitrary gates, qubit reset, feedback, measurement
- Algorithm specified in any gate set

Translate to processor

- Arbitrary gates compiled into available gate set
- Processor connectivity and timing constraints enforced

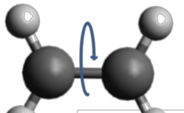
Translation to hardware

- Gates define pulse parameters (shape, phase, sequence)
- Reset/feedback code applied by FPGAs

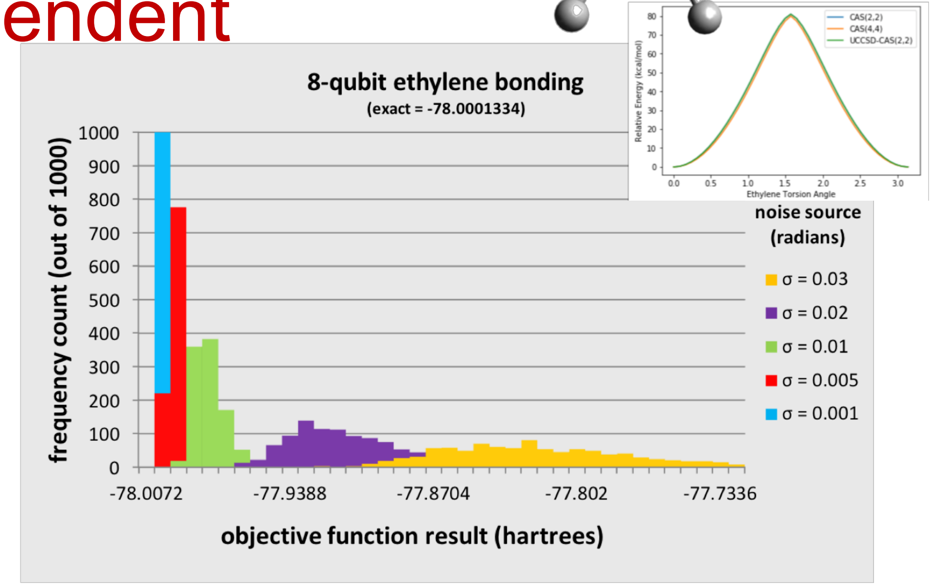
Courtesy of Irfan Siddiqi.

Noise

- Quantum computing is a *physics experiment*
  - Noise is everywhere
    - Circuit-level (inexact gate operations, calibration)
    - Control hardware (electronic cross talk)
    - Quantum cross-talk (extra terms in Hamiltonian; not understood)
  - Measurement errors
- Impact is algorithm-dependent



Case Study: simulation of ethylene, using the Variational Quantum Eigensolver (a hybrid quantum-classical minimizer), shows that noise acts as a *random walk away* from the global minimum, not a Gaussian spread around it.



Error Correction

- Quantum computing is analog
  - Sensitive to noise: no projection to 0 or 1 as in digital
  - All states are valid: can not detect noisy results
- Use group theory: algebra over *logical* qubits
  - Use multiple qubits to represent states
  - Errors fall outside the group and can be detected
  - Stabilizer codes map errors back onto the group
  - Will require 1000s of qubits: not near-term

Example (3-bit flip code):

$|0\rangle \rightarrow |0_L\rangle \equiv |000\rangle$

$|1\rangle \rightarrow |1_L\rangle \equiv |111\rangle$

Single bit-flip leads to a detectable (and correctable) state:

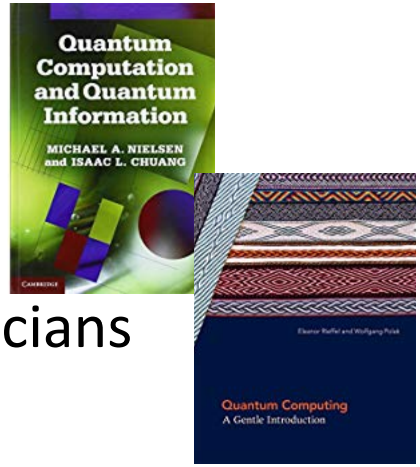
$|101\rangle \rightarrow |111\rangle$

Getting Started

- All quantum computing are physics experiments
    - Calibration issues, noise, experimental uncertainty, etc.
    - Lots of statistics and data analysis required
    - Physicists (still) dominate the field
- Virtually all software is Python-based (with C++)
    - Some domain specific languages (Q#, Python based)
- Math is important but not crucial at current scales
    - Unitary decomposition, compilation, group theory, solvers
    - Brute force approaches still okay up to ~16 qubits

If you come from a High Energy Physics background, you already have the right skill set to get started.

Study Resources



- Nielsen & Chuang
  - Complete, lots of material, better for physicists
- Rieffel & Polak, “A Gentle Introduction”
  - Targeted at computer scientists and mathematicians
- John’s Preskill’s lecture notes

  http://www.theory.caltech.edu/~preskill/ph219/ph219_2017
- Todd Brun’s lecture notes (insightful)

  https://www-bcf.usc.edu/~tbrun/Course/
- Interactive circuit simulator

  http://algassert.com/quirk

Conferences: http://quantum.info/conf/2019.html
Papers: https://arxiv.org

Software

- Frameworks from most chip providers
  - Rapid development, no clear overall front-runner

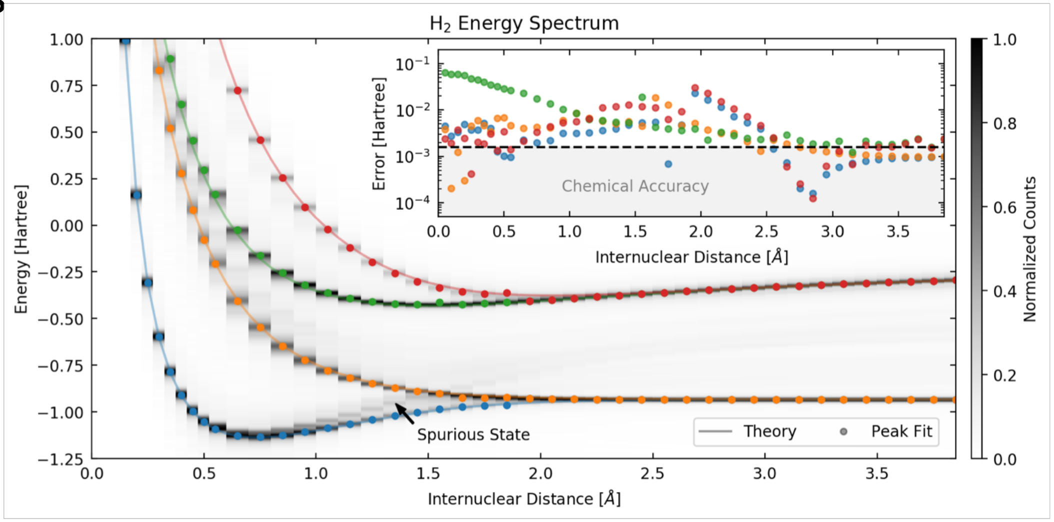
Provider	*Framework*	*License*	*Cloud*
IBM	QisKit	Minor restrictions	IBM Q-Experience
Google	Cirq	Open	
Rigetti	Forest / PyQuil	Restrictive	Rigetti QCS (beta)
Microsoft	LiQUi> / Q#	Minor restrictions	
D-Wave	qbsolv	Minor restrictions	D-Wave Leap

- Academia & startups target the above
  - E.g. PyTKET (Cambridge Quantum), ProjectQ (ETH Zürich)
  - QuTiP (Academia, also RIKEN; http://qutip.org)

https://github.com/qosf/os_quantum_software

Noisy Intermediate-Scale Quantum Devices (NISQ)

- Near-term devices will be noisy and limited
  - Likely tailored to specific applications
    - Gate set, connectivity, calibration, etc.
  - Early algorithms will be hybrid classical-quantum
- Most likely early success stories:
  - Material science, pharmaceuticals
    - Simulation of molecules
  - Optimization problems
    - AI training
    - Graph problems
  - Pure science
    - Black hole simulation

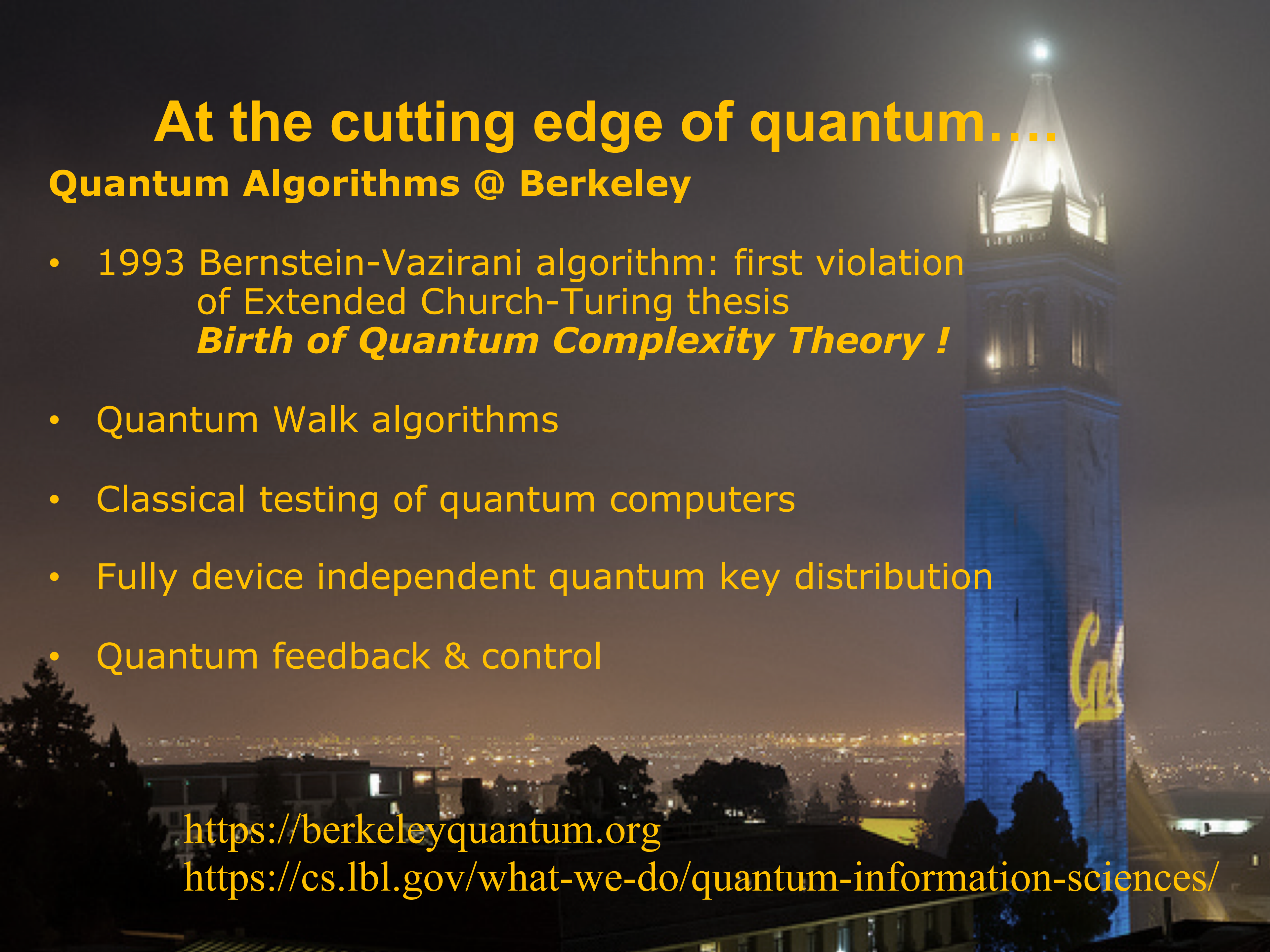


Outlook

- No current technology is destined to succeed
  - How to scale? Error correct? Handle I/O?
- Will there be a "Quantum Winter"?
  - Maybe not: the hype cycle was in the 1990s
- But quantum need not beat classical to be useful
  - A supercomputer is easily 100x as expensive (to build *and* run)
  - When memory limited, supercomputers are well below peak

Regardless where quantum computing will end up, there are many open questions and there is lots of room for new, creative ideas.

Now is an excellent and fun time to join!



At the cutting edge of quantum….

Quantum Algorithms @ Berkeley

- 1993 Bernstein-Vazirani algorithm: first violation of Extended Church-Turing thesis
  Birth of Quantum Complexity Theory !
- Quantum Walk algorithms
- Classical testing of quantum computers
- Fully device independent quantum key distribution
- Quantum feedback & control

https://berkeleyquantum.org
https://cs.lbl.gov/what-we-do/quantum-information-sciences/